ME 639: Introduction to Robotics (Fall'18) Mid-term exam report

Shail Jadav

03/10/2018

1 Equation of motion

For a planar 2-DOF, 2R robot (all revolute joints), with link lengths $l_1 \ \ and \ \ l_2$, link masses $m_1 \ \ and \ \ m_2$. Joint positions, joint velocities, and joint accelerations are denoted as $\theta_i, \dot{\theta}_i, and \ddot{\theta}_i$, i=1-2, respectively. Also, τ_i , i=1-2 , denote the joint torque values.

The equation of motion given as:

$$\tau_1 = M_{11}\ddot{\theta}_1 + M_{12}\ddot{\theta}_2 + H_1 + G_1$$
$$\tau_2 = M_{21}\ddot{\theta}_1 + M_{22}\ddot{\theta}_2 + H_2 + G_2$$

where

$$M_{11} = (\tfrac{m_1}{3} + m_2)l_1^2 + \tfrac{m_2}{3}l_2^2 + m_2 l_1 l_2 \cos\theta_2$$

$$M_{12} = M_{21} = m_2(\tfrac{l_2^2}{3} + \tfrac{l_1}{2}l_2\cos\theta_2)$$

$$M_{22} = \tfrac{m_2}{3}l_2^2$$

$$H_1 = -m_2 l_1 l_2 \sin\theta_2 \dot{\theta}_1\dot{\theta}_2 - \tfrac{m_2}{2}l_1 l_2 \sin\theta_2 \dot{\theta}_2^2$$

$$H_2 = \tfrac{m_2}{2}l_1 l_2 \sin\theta_2 \dot{\theta}_1^2$$

$$G_1 = ((\tfrac{m_1}{2} + m_2)l_1\cos\theta_1 + \tfrac{m_2}{2}l_2\cos(\theta_1+\theta_2))g$$

$$G_2 = \tfrac{m_2}{2}l_2\cos(\theta_1+\theta_2)g$$

$$\begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix} = \begin{bmatrix}M_{11} & M_{12}\\ M_{21} & M_{22}\end{bmatrix}\begin{bmatrix}\ddot{\theta}_1\\ \ddot{\theta}_2\end{bmatrix} + \begin{bmatrix}H_1\\ H_2\end{bmatrix} + \begin{bmatrix}G_1\\ G_2\end{bmatrix}$$

2 Problem 1

(Forward Dynamics) For the following initial condition (at time, t, = 0), IC, and manipulator parameters (in SI units),

$IC = [q_1 \quad q_2 \quad q_{d1} \quad q_{d2}] = [\pi/4 \quad \pi/4 \quad 0 \quad 0]; l_1 = l_2 = 1; m_1 = m_2 = 1; I_1 = I_2 = 1/12$; Use a numerical solver, say ode45 function in Matlab , to solve Eq. (1) for a span of time, $[t_i \quad t_f] = [0 \quad 10]$, when

2.1 A

No joint torques are applied, i.e., $\tau_1 = \tau_2 = 0$. Plot the kinetic, potential and total energy of the system with respect to time. Discuss the results.

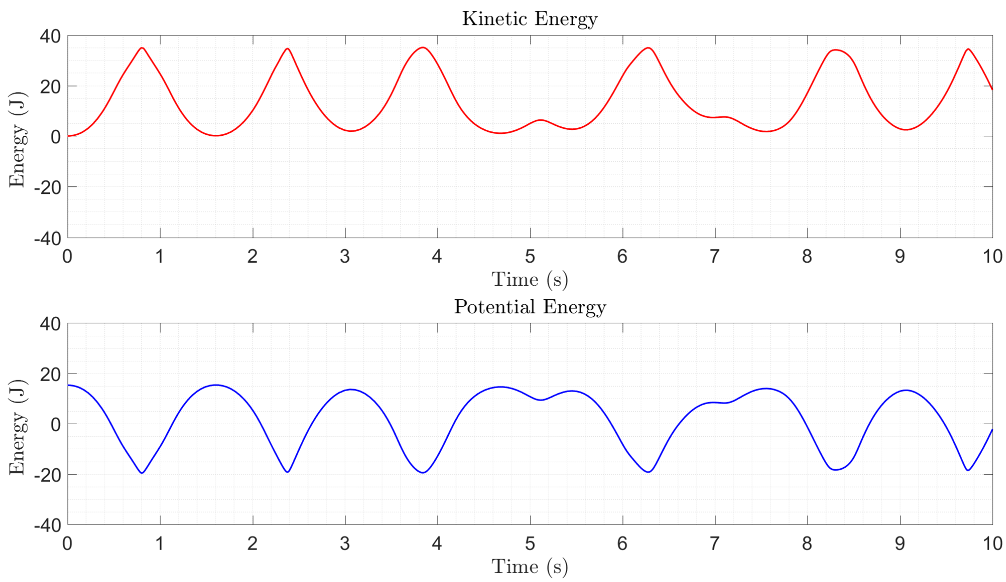

Figure 1: Kinetic energy and potential energy when both joint torques are zero

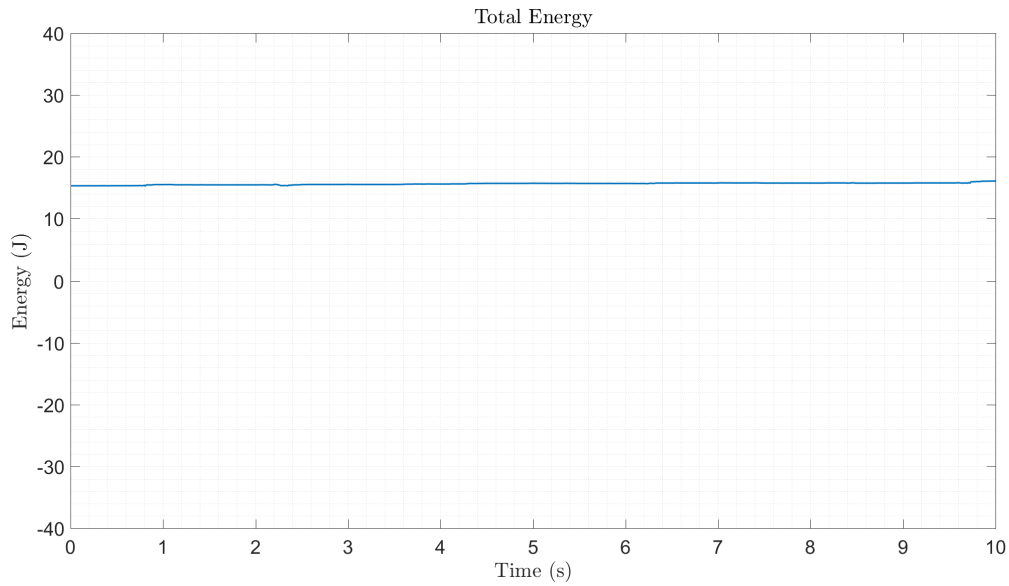

Figure 2: Total energy when both joint torques are zero

As expected initially there is 15.31 J potential energy because we are releasing it from height, we are not applying any torque to the system means no energy input to the system, due to that If the kinetic energy is increasing the potential energy must decrease so we can have constant total energy,this analogy we can see in the figure 1.

The total energy should be constant because we are not applying torque to the links so there is no energy input. In figure two we can observe that the total energy is constant although there are some numerical errors associated with that we can see in figure two, by using ode23 we can have accurate results sown in figure 3.

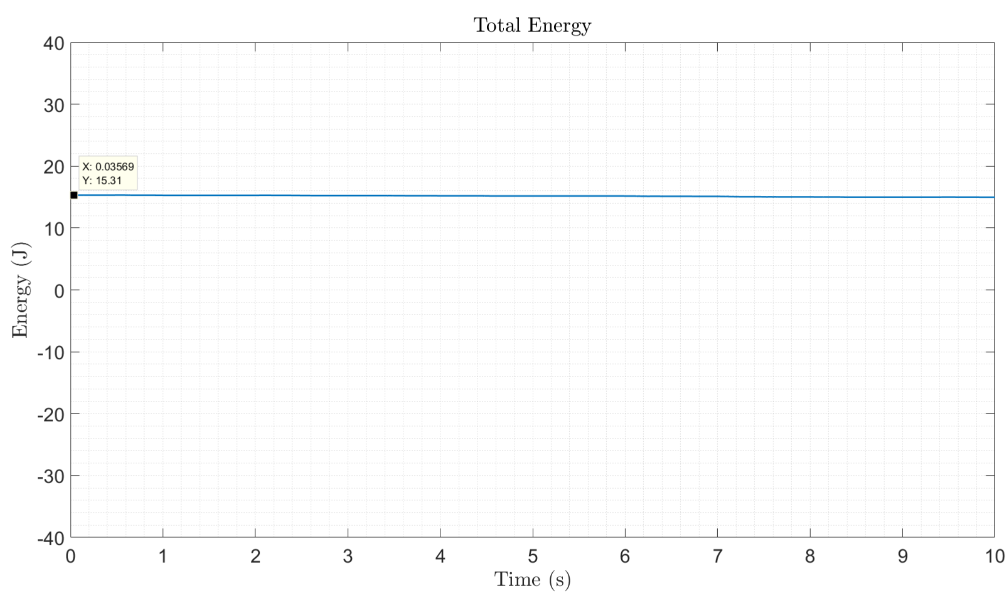

Figure 3: Total energy when both joint torques are zero computed with ODE23

2.1.1 Matlab Program

```
% ME 639: Introduction to robotics
% Midsem exam : Question 1 (a)
%                3 Oct 2018
%a.  No joint torques are applied, i.e., T 1 = T 2 = 0 . Plot the kinetic, potential
%    and total energy of the system with respect to time. Discuss the results.
%
% Author: Shail Jadav 18310039
%% Initialization
clear
close all
clc
%% ODE solver

[t,x]=ode45('ode_solver_script_q1_a',[0,10],[pi/4,0,pi/4,0]);  % Time span 0 to 10 IC=[pi/4 0 pi/4
    0] Theta1=pi/4 Theta2=pi/4

m1=1; m2=1; l1=1; l2=1; g=9.81;

th1=x(:,1); dth1=x(:,2); th2=x(:,3); dth2=x(:,4); %Joint position and velocities

%% Energy Calculation

for i=1:1:length(th1)

% Equation for kinetic energy
KE(i,1) = (0.5* (((m1/3) + m2)*l1*l1*dth1(i)*dth1(i))) + ((1/6)*m2*l2*l2*(dth1(i)^2 + dth2(i)^2 +
    2*dth1(i)*dth2(i))) + (0.5*m2*l1*l2*cos(th2(i))*(dth1(i)^2 + dth1(i) *dth2(i)));

% Equation for potential energy
PE(i,1)=((0.5*m1 + m2)*g*l1*sin(th1(i)))+(0.5*m2*g*l2*sin(th1(i)+th2(i)));
end

%Total energy
TE=KE+PE;

%% Display The Results

%Ploting energies
figure('units','normalized','outerposition',[0 0 1 1])
subplot(211)
plot(t,KE,'r','LineWidth',1.5)
title("Kinetic Energy",'Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Energy (J) ','Interpreter','latex')
ylim([-40 40])
set(gca,'FontSize',18)
grid minor


subplot(212)
plot(t,PE,'b','LineWidth',1.5)
title("Potential Energy",'Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Energy  (J) ','Interpreter','latex')
ylim([-40 40])
set(gca,'FontSize',18)
grid minor
set(gca)
saveas(gcf,'Q1_a_KE_PE1.png')

figure('units','normalized','outerposition',[0 0 1 1])
plot(t,TE,'LineWidth',1.5)
title("Total Energy",'Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Energy  (J)','Interpreter','latex')
ylim([-40 40])
set(gca,'FontSize',18)
grid minor
saveas(gcf,'Q1_a_TE1.png')
```

ODE Solver script

```
function Out= ode_solver_script_q1_a(t,x)

%% Input parameters
m2=1; m1=1; l1=1; l2=1; g=9.81;

tau1= 0; %Input torque is zero
tau2=tau1; %second torque is the same as torque 1 which is zero


%% Equation of motion

%M matrix
M11=((((m1/3) + m2)*l1^2) +((m2/3)*l2^2) + (m2*l1*l2*cos(x(3))));
M12=(m2*(((l2^2)/3) + (0.5*l1*l2*cos(x(3)))));
M21=M12;
M22=((1/3)*m2*l2*l2);

%H matrix
H1 =((-m2*l1*l2*sin(x(3))*x(2)*x(4)) - (0.5*m2*l1*l2*sin(x(3))*x(4)*x(4)));
H2 = (0.5 * m2* l1*l2*sin(x(3))*x(2)*x(2));

%G matrix
G1=( ((((0.5*m1) + m2)*l1*cos(x(1))) + (0.5*m2*l2*cos(x(1)+x(3))))*g);
G2=0.5*m2*l2*cos(x(1)+x(3))*g;

T=[tau1; tau2];

M=[M11 M12;M21 M22];

HG = [H1 + G1; H2 + G2];

%% Equation in terms of acceleration

ddth = (inv(M)) * (T - HG )     ;% Joint  accelaration

OP=zeros(4,1);

%% Output
OP(1)=x(2); %Intergretion of velocity will give the position for theta 1
OP(2)=ddth(1);%Intergretion of acceleration will give the velocity for theta 1
OP(3)=x(4); %Intergretion of velocity will give the position  for theta 2
OP(4)=ddth(2); %Intergretion of acceleration will give the velocity for theta 2


Out=OP; % Output

end
```

2.2 B

Sinusoidal joint torques are applied, i.e., $\tau_1 = \tau_2 = \sin t$. Plot joint positions and angular velocities with respect to time. Discuss the results.

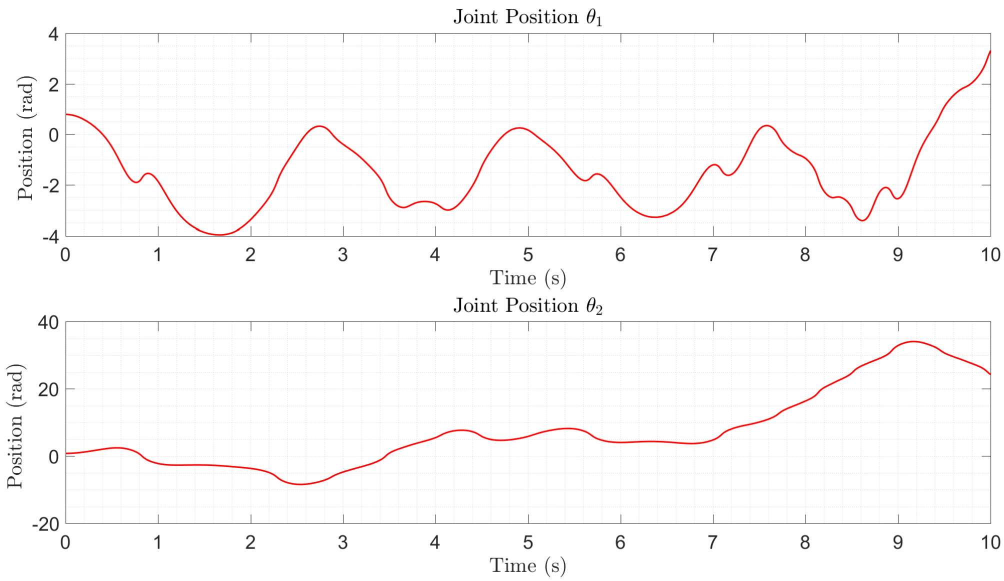

Figure 4: Joint positions when joint torques are sinusoidal

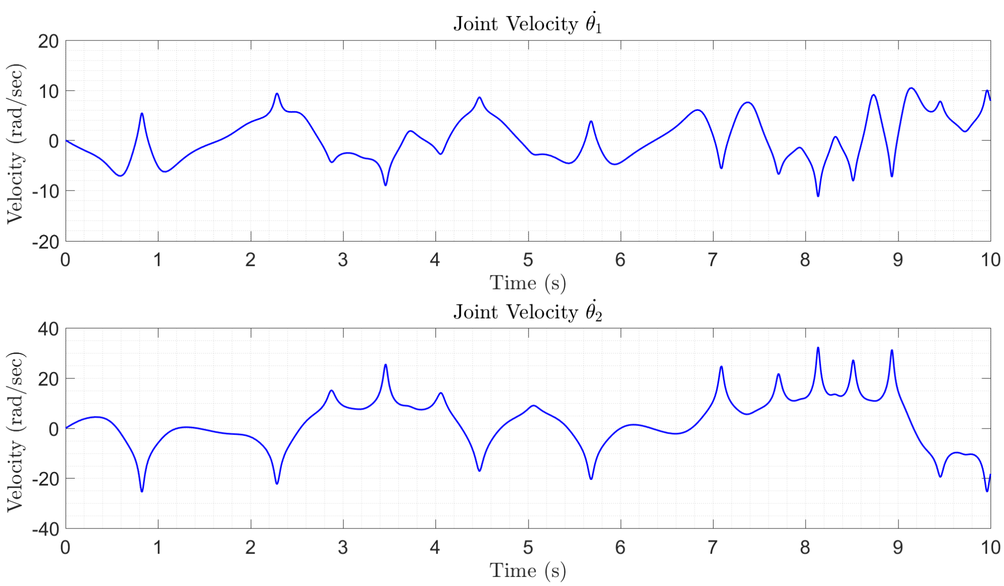

Figure 5: Joint velocities when joint torques are sinusoidal

As the initial conditions of both joints are pi/4 the solution starts with pi/4 initial conditions the joint 1 position is between [-4 4] rad wheres the joint position 2 is having larger magnitude between [-9 34], after 2.55 sec till 9.55 sec the joint 2 is keep rotating in one direction. This mechanical system also chaotic that is represented in the figure 4 and 5.The both joint velocities are rapidly changing the directions due to the chaotic motion produce by the system.

2.2.1 Matlab Program

```matlab
% ME 639: Introduction to robotics
% Midsem exam : Question 1 (b)
%               3 Oct 2018
%b. Sinusoidal joint torques are applied, i.e., T 1 = T 2 = sin(t) . Plot joint positions and angular
%   velocities with respect to time. Discuss the results.
%
% Author: Shail Jadav 18310039
%% Initialization
clear
close all
clc
%% ODE solver

[t,x]=ode45('ode_solver_script_q1_b',[0,10],[pi/4,0,pi/4,0]);  % Time span 0 to 10 IC=[pi/4 0 pi/4 0] Theta1=pi/4 Theta2=pi/4

m1=1; m2=1; l1=1; l2=1; g=9.81;

th1=x(:,1); dth1=x(:,2); th2=x(:,3); dth2=x(:,4);

%% Display The Results

figure('units','normalized','outerposition',[0 0 1 1])
subplot(211)
plot(t,th1,'r','LineWidth',1.5)
title('Joint Position $\theta_1$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Position (rad) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor


subplot(212)
plot(t,th2,'r','LineWidth',1.5)
title('Joint Position $\theta_2$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Position (rad) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor
set(gca)
saveas(gcf,'Q1_b_JP.png')

figure('units','normalized','outerposition',[0 0 1 1])
subplot(211)
plot(t,dth1,'b','LineWidth',1.5)
title('Joint Velocity $\dot{\theta_1}$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Velocity (rad/sec) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor


subplot(212)
plot(t,dth2,'b','LineWidth',1.5)
title('Joint Velocity $\dot{\theta_2}$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Velocity (rad/sec) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor
set(gca)
saveas(gcf,'Q1_b_JV.png')
```

ODE Solver script

```matlab
function Out= ode_solver_script_q1_b(t,x)

%% Input parameters
```

```
m2=1; m1=1; l1=1; l2=1; g=9.81;

tau1=sin(t); %Input torque is sin(t)
tau2=tau1; %second torque is the same as torque 1


%% Equation of motion
M11=((((m1/3) + m2)*l1^2) +((m2/3)*l2^2) + (m2*l1*l2*cos(x(3))));
M12=(m2*(((l2^2)/3) + (0.5*l1*l2*cos(x(3)))));
M21=M12;
M22=((1/3)*m2*l2*l2);


H1 =((-m2*l1*l2*sin(x(3))*x(2)*x(4)) - (0.5*m2*l1*l2*sin(x(3))*x(4)*x(4)));
H2 = (0.5 * m2* l1*l2*sin(x(3))*x(2)*x(2));

G1=( ((((0.5*m1) + m2)*l1*cos(x(1))) + (0.5*m2*l2*cos(x(1)+x(3))))*g);
G2=0.5*m2*l2*cos(x(1)+x(3))*g;

T=[tau1; tau2];

M=[M11 M12;M21 M22];

HG = [H1 + G1; H2 + G2];

%% Equation in terms of acceleration

ddth = (inv(M)) * (T - HG )     ;

OP=zeros(4,1);

%% Output
OP(1)=x(2); %Intergretion of velocity will give the position for theta 1
OP(2)=ddth(1);%Intergretion of acceleration will give the velocity for theta 1
OP(3)=x(4); %Intergretion of velocity will give the position  for theta 2
OP(4)=ddth(2); %Intergretion of acceleration will give the velocity for theta 2


Out=OP; % Output

end
```

2.3 C

From the results of part (b), explain how changing the IC to [0 0 0 0] would affect the solution.

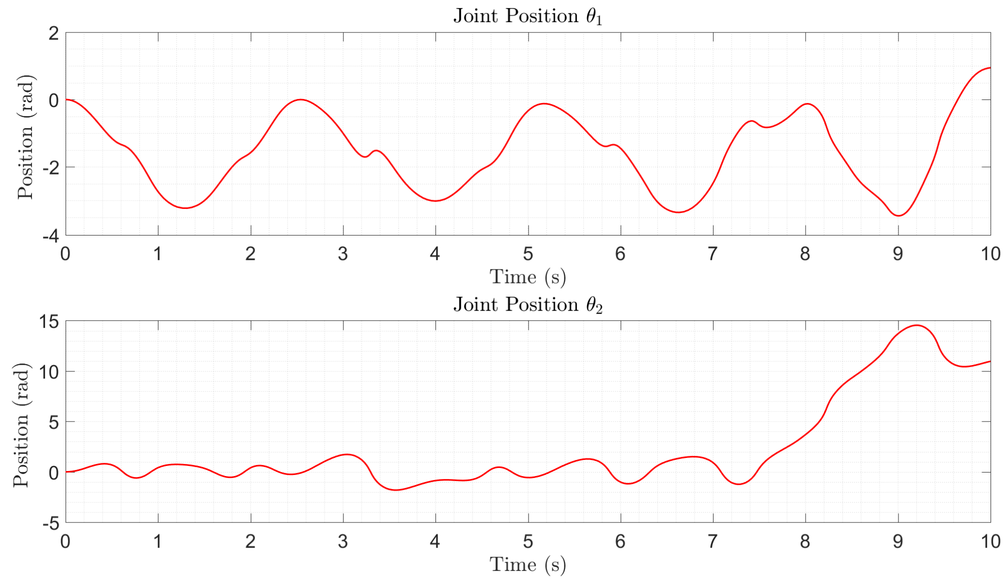

Figure 6: Joint positions when joint torques are sinusoidal

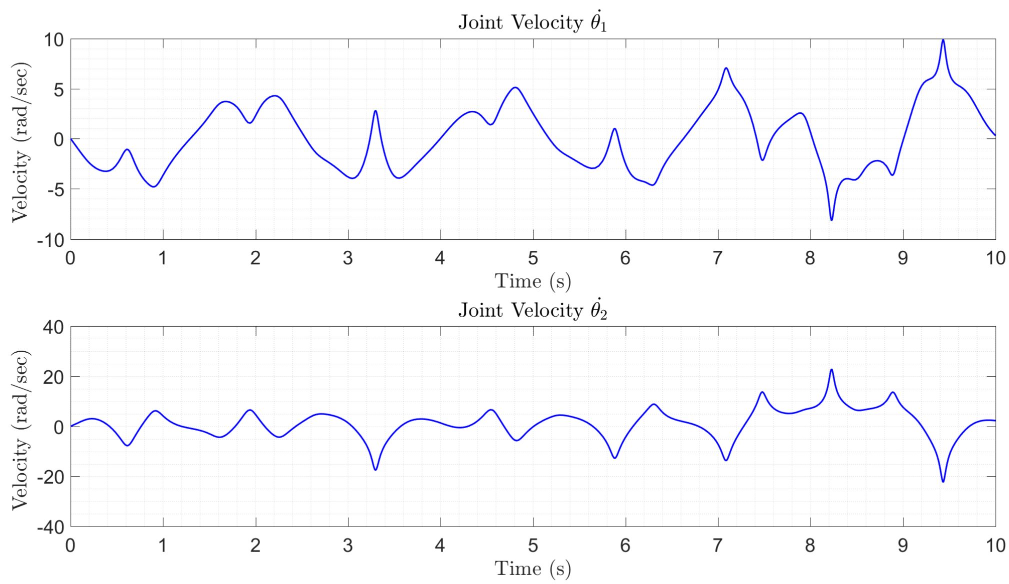

Figure 7: Joint velocities when joint torques are sinusoidal

As the initial conditions of both joints are 0 the solution starts with 0 initial conditions the joint 1 position is between [-4 2] rad wheres the joint position 2 is having larger magnitude between [-1 16], after 7.55 sec till 9 sec the joint 2 is keep rotating in one direction.Initially the joint 1 velocity is significantly low due to less potential energy than the previous initial conditions. As well as the overall magnitude of joint position 2 is less than the values for previous initial conditions.

2.3.1 Matlab Program

```matlab
% ME 639: Introduction to robotics
% Midsem exam : Question 1 (c)
%               3 Oct 2018
%c. From the results of part (b), explain how changing the IC to [0 0 0 0] would affect the
%   solution.
%
% Author: Shail Jadav 18310039
%% Initialization
clear
close all
clc
%% ODE solver

[t,x]=ode45('ode_solver_script_q1_c',[0,10],[0,0,0,0]);  % Time span 0 to 10 IC=[0 0 0 0] Theta1=0
    Theta2=0

m1=1; m2=1; l1=1; l2=1; g=9.81;

th1=x(:,1); dth1=x(:,2); th2=x(:,3); dth2=x(:,4);



%% Display The Results

figure('units','normalized','outerposition',[0 0 1 1])
subplot(211)
plot(t,th1,'r','LineWidth',1.5)
title('Joint Position $\theta_1$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Position (rad) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor


subplot(212)
plot(t,th2,'r','LineWidth',1.5)
title('Joint Position $\theta_2$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Position (rad) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor
set(gca)
saveas(gcf,'Q1_c_JP.png')

figure('units','normalized','outerposition',[0 0 1 1])
subplot(211)
plot(t,dth1,'b','LineWidth',1.5)
title('Joint Velocity $\dot{\theta_1}$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Velocity (rad/sec) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor


subplot(212)
plot(t,dth2,'b','LineWidth',1.5)
title('Joint Velocity $\dot{\theta_2}$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Velocity (rad/sec) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor
set(gca)
saveas(gcf,'Q1_c_JV.png')
```

ODE Solver script

```matlab
function Out= ode_solver_script_q1_c(t,x)

```

```
%% Input parameters
m2=1; m1=1; l1=1; l2=1; g=9.81;

tau1=sin(t); %Input torque is zero
tau2=tau1; %second torque is the same as torque 1


%% Equation of motion
M11=((((m1/3) + m2)*l1^2) +((m2/3)*l2^2) + (m2*l1*l2*cos(x(3))));
M12=(m2*(((l2^2)/3) + (0.5*l1*l2*cos(x(3)))));
M21=M12;
M22=((1/3)*m2*l2*l2);


H1 =((-m2*l1*l2*sin(x(3))*x(2)*x(4)) - (0.5*m2*l1*l2*sin(x(3))*x(4)*x(4)));
H2 = (0.5 * m2* l1*l2*sin(x(3))*x(2)*x(2));

G1=( ((((0.5*m1) + m2)*l1*cos(x(1))) + (0.5*m2*l2*cos(x(1)+x(3))))*g);
G2=0.5*m2*l2*cos(x(1)+x(3))*g;

T=[tau1; tau2];

M=[M11 M12;M21 M22];

HG = [H1 + G1; H2 + G2];

%% Equation in terms of acceleration

ddth = (inv(M)) * (T - HG )    ;

OP=zeros(4,1);

%% Output
OP(1)=x(2); %Intergretion of velocity will give the position for theta 1
OP(2)=ddth(1);%Intergretion of acceleration will give the velocity for theta 1
OP(3)=x(4); %Intergretion of velocity will give the position  for theta 2
OP(4)=ddth(2); %Intergretion of acceleration will give the velocity for theta 2


Out=OP; % Output

end
```

2.4 D. Animate the 2R manipulator for part (a) and (b).

Part A

Please click on following links (Blue color)

Animation of Q1 A

2.4.1 Matlab Program

```
% ME 639: Introduction to robotics
% Midsem exam : Question 1 D
%               3 Oct 2018
% Animate the 2R manipulator for part (a) and (b).
%
% Author: Shail Jadav 18310039
%% Initialization
clear
close all
clc
%%
%syms theta1 theta2 theta3 l1 l2 l3

[t,x]=ode45('ode_solver_script_q1_a',[0,10],[pi/4,0,pi/4,0]);

m1=1; m2=1; l1=1; l2=1; g=9.81;

th1=x(:,1); dth1=x(:,2); th2=x(:,3); dth2=x(:,4);

%%

```

```
for i=1:1:length(th1)

KE(i,1) = (0.5* (((m1/3) + m2)*l1*l1*dth1(i)*dth1(i))) + ((1/6)*m2*l2*l2*(dth1(i)^2 + dth2(i)^2 +
    2*dth1(i)*dth2(i))) + (0.5*m2*l1*l2*cos(th2(i))*(dth1(i)^2 + dth1(i) *dth2(i)));

PE(i,1)=((0.5*m1 + m2)*g*l1*sin(th1(i)))+(0.5*m2*g*l2*sin(th1(i)+th2(i)));
end

TE=KE+PE;


figure('units','normalized','outerposition',[0 0 1 1])
for i=1:1:length(x)

theta1=x(i,1);
dtheta1=x(i,2);
theta2=x(i,3);
dtheta2=x(i,4);

l1=1; %Input the l length
l2=1; %Input the l length



% Homogeneus transformation matrix
H01 = [cos(theta1) -sin(theta1) 0 l1*cos(theta1);sin(theta1) cos(theta1) 0 l1*sin(theta1);0 0 1
    0;0 0 0 1]; %Frame 0 to 1 tranformation
H12 = [cos(theta2) -sin(theta2) 0 l2*cos(theta2);sin(theta2) cos(theta2) 0 l2*sin(theta2);0 0 1
    0;0 0 0 1]; %Frame 1 to 2 tranformation


H02=H01*H12;      %Frame 0 to 2 tranformation


O=[0,0];                    %Joint 1 position
P1=[H01(1,4) H01(2,4)];     %Joint 2 position
P2=[H02(1,4) H02(2,4)];     %Joint 3 position


Orn= atan2(H02(2,1),H02(1,1));  %Orientation of end effector
Orn=(Orn)*(180/pi);
subplot(221)
plot(P1(1),P1(2),'ok','LineWidth',5)
hold on
plot(P2(1),P2(2),'om','LineWidth',5)
plot(0,0,'ok','LineWidth',10)
xlim([-2.5 2.5])
ylim([-2.5 2.5])

grid minor
plot([0 P1(1)], [0 P1(2)],'r','LineWidth',5)
plot([P1(1) P2(1)], [P1(2) P2(2)],'b','LineWidth',5)
hold off
title(strcat('Time = ',num2str(t(i,1))),'Interpreter','latex')
xlabel('X axis (m)','Interpreter','latex')
ylabel('Y axis (m)','Interpreter','latex')
set(gca,'FontSize',18)
pause(0.000000000000000000000000000000000000000001);

subplot(223)
plot(t(i,1),TE(i,1),'ok','LineWidth',1)
hold on
title("Total Energy",'Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Energy  (J)','Interpreter','latex')
ylim([-40 40])
set(gca,'FontSize',18)
grid minor


subplot(222)
plot(t(i,1),KE(i,1),'or','LineWidth',1)
hold on
```

```matlab
title("Kinetic Energy",'Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Energy (J) ','Interpreter','latex')
ylim([-40 40])
set(gca,'FontSize',18)
grid minor

subplot(224)
plot(t(i,1),PE(i,1),'ob','LineWidth',1)
hold on
title("Potential Energy",'Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Energy  (J) ','Interpreter','latex')
ylim([-40 40])
set(gca,'FontSize',18)
grid minor
set(gca)
F(i) = getframe(gcf) ;
end

% create the video writer with 30 fps
  writerObj = VideoWriter('animation_Q1_a.avi');
  writerObj.FrameRate = 30;
 % set the seconds per image
% open the video writer
open(writerObj);
% write the frames to the video
for i=1:length(F)
    % convert the image to a frame
    frame = F(i) ;
    writeVideo(writerObj, frame);
end
% close the writer object
close(writerObj);
```

Part B

Animation of Q1 B

2.4.2 Matlab Program

```matlab
% ME 639: Introduction to robotics
% Midsem exam : Question 1 D
%                3 Oct 2018
% Animate the 2R manipulator for part (a) and (b).
%
% Author: Shail Jadav 18310039
%% Initialization
clear
close all
clc
%%
%syms theta1 theta2 theta3 l1 l2 l3

[t,x]=ode45('ode_solver_script_q1_b',[0,10],[pi/4,0,pi/4,0]);

m1=1; m2=1; l1=1; l2=1; g=9.81;

th1=x(:,1); dth1=x(:,2); th2=x(:,3); dth2=x(:,4);

%%

for i=1:1:length(th1)

    KE(i,1) = (0.5* (((m1/3) + m2)*l1*l1*dth1(i)*dth1(i))) + ((1/6)*m2*l2*l2*(dth1(i)^2 + dth2(i)
    ^2 + 2*dth1(i)*dth2(i))) + (0.5*m2*l1*l2*cos(th2(i))*(dth1(i)^2 + dth1(i) *dth2(i)));

    PE(i,1)=((0.5*m1 + m2)*g*l1*sin(th1(i)))+(0.5*m2*g*l2*sin(th1(i)+th2(i)));
end

TE=KE+PE;


figure('units','normalized','outerposition',[0 0 1 1])
for i=1:1:length(x)
```

```
theta1=x(i,1);
dtheta1=x(i,2);
theta2=x(i,3);
dtheta2=x(i,4);

l1=1; %Input the l length
l2=1; %Input the l length



% Homogeneus transformation matrix
H01 = [cos(theta1) -sin(theta1) 0 l1*cos(theta1);sin(theta1) cos(theta1) 0 l1*sin(theta1);0 0
1 0;0 0 0 1]; %Frame 0 to 1 tranformation
H12 = [cos(theta2) -sin(theta2) 0 l2*cos(theta2);sin(theta2) cos(theta2) 0 l2*sin(theta2);0 0
1 0;0 0 0 1]; %Frame 1 to 2 tranformation


H02=H01*H12;      %Frame 0 to 2 tranformation


O=[0,0];                    %Joint 1 position
P1=[H01(1,4) H01(2,4)];     %Joint 2 position
P2=[H02(1,4) H02(2,4)];     %Joint 3 position


Orn= atan2(H02(2,1),H02(1,1));  %Orientation of end effector
Orn=(Orn)*(180/pi);
subplot(221)
plot(P1(1),P1(2),'ok','LineWidth',5)
hold on
plot(P2(1),P2(2),'om','LineWidth',5)
plot(0,0,'ok','LineWidth',10)
xlim([-2.5 2.5])
ylim([-2.5 2.5])
grid minor
plot([0 P1(1)], [0 P1(2)],'r','LineWidth',5)
plot([P1(1) P2(1)], [P1(2) P2(2)],'b','LineWidth',5)
hold off
title(strcat('Time = ',num2str(t(i,1))),'Interpreter','latex')
xlabel('X axis (m)','Interpreter','latex')
ylabel('Y axis (m)','Interpreter','latex')
set(gca,'FontSize',18)
pause(0.0000000000000000000000000000000000000001);

subplot(223)
plot(t(i,1),TE(i,1),'ok','LineWidth',1)
hold on
title("Total Energy",'Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Energy  (J)','Interpreter','latex')
ylim([-40 40])
set(gca,'FontSize',18)
grid minor


subplot(222)
plot(t(i,1),theta1,'or','LineWidth',1)
hold on
title('$\theta_{1}$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Energy (J) ','Interpreter','latex')
set(gca,'FontSize',18)
grid minor

subplot(224)
plot(t(i,1),theta2,'ob','LineWidth',1)
hold on
title('$\theta_{2}$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Energy  (J) ','Interpreter','latex')
set(gca,'FontSize',18)
grid minor
```

```
    set(gca)
    F(i) = getframe(gcf) ;
end

% create the video writer with 30 fps
writerObj = VideoWriter('animation_Q1_b.avi');
writerObj.FrameRate = 30;
% set the seconds per image
% open the video writer
open(writerObj);
% write the frames to the video
for i=1:length(F)
    % convert the image to a frame
    frame = F(i) ;
    writeVideo(writerObj, frame);
end
% close the writer object
close(writerObj);
```

3 Problem 2

3.1 Part A

Plan a trajectory when states of the system, [q1 q2 qd1 qd2], change from [0 0 0 0] to [pi/6 pi/3 0 0] in 10 sec . Plot the trajectory (q, qd and qdd) with respect to time.

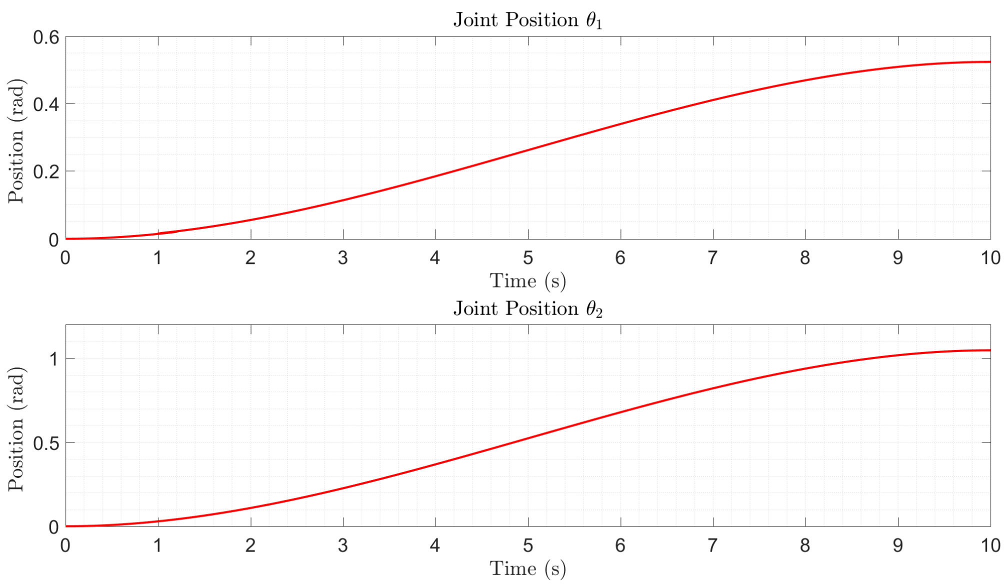

Figure 8: Joint positions for generated trajectory

We can plan the trajectory for point to point motion using polynomial method for trajectory generation where the desired trajectory can be given as the $\theta_i^d = a_0 + a_1 t + a_2 t^2 + a_3 t^3$

where

$a_0 = \theta_i$

$a_1 = \dot{\theta}_i$

$a_2 = \frac{3(\theta_f - \theta_i) - (2\dot{\theta}_i + \dot{\theta}_f)t_f}{t_f^2}$

$a_3 = \frac{2(\theta_i - \theta_f) - (\dot{\theta}_i + \dot{\theta}_f)t_f}{t_f^3}$

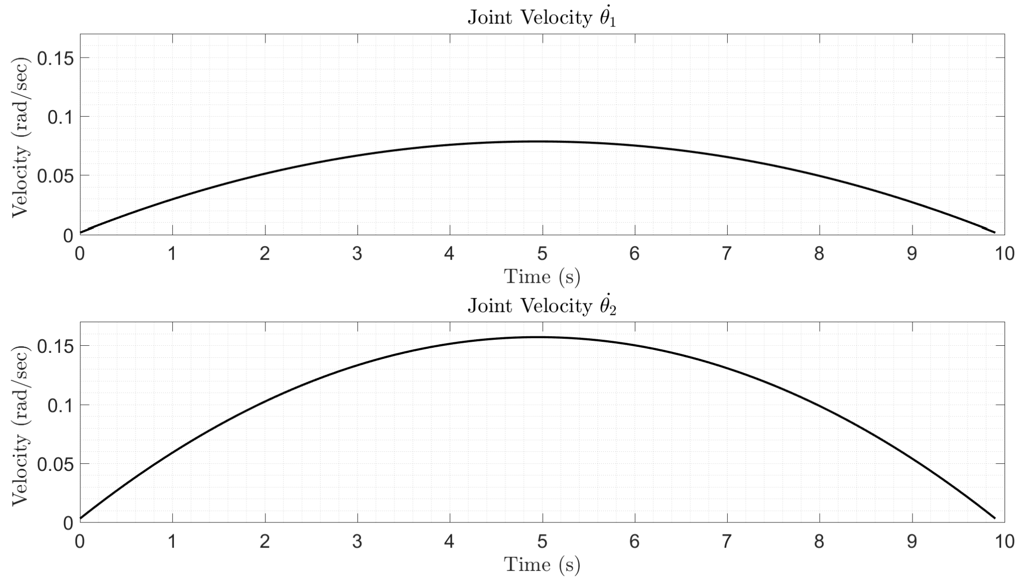

Figure 9: Joint velocities for generated trajectory

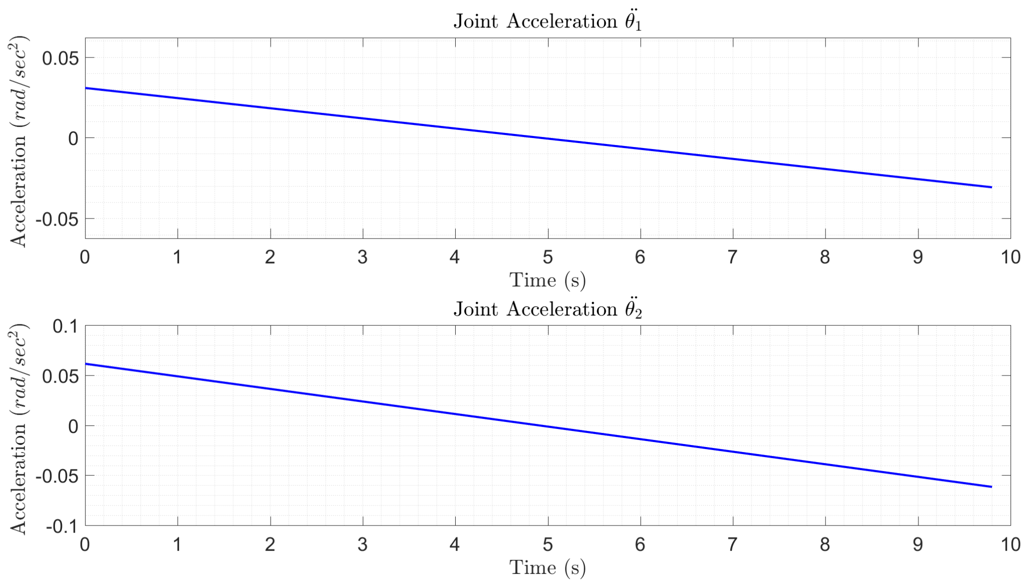

Figure 10: Joint acceleration for generated trajectory

3.1.1 Part B

For the desired joint trajectory from part (a), compute the desired joint torques. Plot the desired joint torque values with respect to time.

The torque can be calculated by just putting the values of joint position, velocities and accelerations in the system equation. $\begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix} = \begin{bmatrix}M_{11} & M_{12}\\ M_{21} & M_{22}\end{bmatrix}\begin{bmatrix}\ddot{\theta}_1\\ \ddot{\theta}_2\end{bmatrix} + \begin{bmatrix}H_1\\ H_2\end{bmatrix} + \begin{bmatrix}G_1\\ G_2\end{bmatrix}$

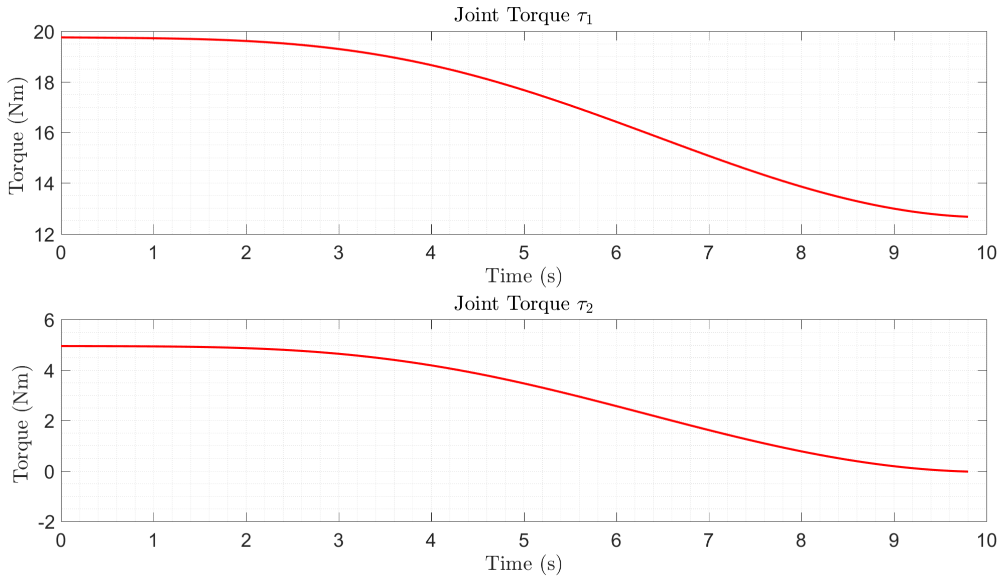

Figure 11: Computed torque for generated trajectory

3.1.2 Matlab Program

```
%% Initialization
clear
close all
clc
%% Trejectory genration
[~,q0,dq0,ddq0]=trecgen(0,1/10,10,0,0,pi/6,0);  %Trejectory generation for theta 1 0 to pi/6
[t,q1,dq1,ddq1]=trecgen(0,1/10,10,0,0,pi/3,0);  %Trejectory generation for theta 2 0 to pi/3


%% Input parameters
m2=1; m1=1; l1=1; l2=1; g=9.81;

tau1= 0; %Input torque is zero
tau2=tau1; %second torque is the same as torque 1



%% Equation of motion

for i=1:1:length(ddq0)
M11=((((m1/3) + m2)*l1^2) +((m2/3)*l2^2) + (m2*l1*l2*cos(q1(1,i))));
M12=(m2*(((l2^2)/3) + (0.5*l1*l2*cos(q1(1,i)))));
M21=M12;
M22=((1/3)*m2*l2*l2);


H1 =((-m2*l1*l2*sin(q1(1,i))*dq0(1,i)*dq1(1,i) - (0.5*m2*l1*l2*sin(q1(1,i))*dq1(1,i)*dq1(1,i))));
H2 = (0.5 * m2* l1*l2*sin(q1(1,i))*dq0(1,i)*dq0(1,i));

G1=( ((((0.5*m1) + m2)*l1*cos(q0(1,i))) + (0.5*m2*l2*cos(q0(1,i)+q1(1,i))))*g);
G2=0.5*m2*l2*cos(q0(1,i)+q1(1,i))*g;

Tau1(1,i)= M11*ddq0(1,i) + M12*ddq1(1,i) + H1 +G1;
Tau2(1,i)= M21*ddq0(1,i) + M22*ddq1(1,i) + H2 +G2;
end

%% Display generated trejectories
figure('units','normalized','outerposition',[0 0 1 1])
subplot(211)
plot(t,q0,'r','LineWidth',2)
title('Joint Position $\theta_1$','Interpreter','latex')
```

```
xlabel('Time (s)','Interpreter','latex')
ylabel('Position (rad) ','Interpreter','latex')
set(gca,'FontSize',18)
grid minor


subplot(212)
plot(t,q1,'r','LineWidth',2)
title('Joint Position $\theta_2$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Position (rad) ','Interpreter','latex')
set(gca,'FontSize',18)
grid minor
ylim([0 1.2])
saveas(gcf,'Q2_a_JP.png')

figure('units','normalized','outerposition',[0 0 1 1])
subplot(211)
plot(t(1,1:end-1),dq0,'k','LineWidth',2)
title('Joint Velocity $\dot{\theta_1}$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Velocity (rad/sec) ','Interpreter','latex')
set(gca,'FontSize',18)
ylim([0 0.17])
grid minor

subplot(212)
plot(t(1,1:end-1),dq1,'k','LineWidth',2)
title('Joint Velocity $\dot{\theta_2}$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Velocity (rad/sec) ','Interpreter','latex')
set(gca,'FontSize',18)
ylim([0 0.17])
grid minor
saveas(gcf,'Q2_a_JV.png')

figure('units','normalized','outerposition',[0 0 1 1])
subplot(211)
plot(t(1,1:end-2),ddq0,'b','LineWidth',2)
title('Joint Acceleration $\ddot{\theta_1}$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Acceleration $(rad/sec^2)$ ','Interpreter','latex')
set(gca,'FontSize',18)
ylim([-0.062 0.062])
grid minor

subplot(212)
plot(t(1,1:end-2),ddq1,'b','LineWidth',2)
title('Joint Acceleration $\ddot{\theta_2}$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Acceleration $(rad/sec^2)$ ','Interpreter','latex')
set(gca,'FontSize',18)
grid minor
saveas(gcf,'Q2_a_JA.png')


%% Display computed torque
figure('units','normalized','outerposition',[0 0 1 1])
subplot(211)
plot(t(1,1:end-2),Tau1,'r','LineWidth',2)
title('Joint Torque $\tau_1$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Torque (Nm) ','Interpreter','latex')
set(gca,'FontSize',18)
grid minor


subplot(212)
plot(t(1,1:end-2),Tau2,'r','LineWidth',2)
title('Joint Torque $\tau_2$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Torque (Nm) ','Interpreter','latex')
set(gca,'FontSize',18)
```

```
grid minor
saveas(gcf,'Q2_b_CT.png')

%% Validation through animation
figure('units','normalized','outerposition',[0 0 1 1])
for i=1:1:length(q0)

theta1=q0(1,i);
theta2=q1(1,i);


l1=1; %Input the l length
l2=1; %Input the l length



% Homogeneus transformation matrix
H01 = [cos(theta1) -sin(theta1) 0 l1*cos(theta1);sin(theta1) cos(theta1) 0 l1*sin(theta1);0 0 1 0;0 0 0 1]; %Frame 0 to 1 tranformation
H12 = [cos(theta2) -sin(theta2) 0 l2*cos(theta2);sin(theta2) cos(theta2) 0 l2*sin(theta2);0 0 1 0;0 0 0 1]; %Frame 1 to 2 tranformation


H02=H01*H12;       %Frame 0 to 2 tranformation


O=[0,0];                    %Joint 1 position
P1=[H01(1,4) H01(2,4)];     %Joint 2 position
P2=[H02(1,4) H02(2,4)];     %Joint 3 position


Orn= atan2(H02(2,1),H02(1,1));  %Orientation of end effector
Orn=(Orn)*(180/pi);
%subplot(221)
plot(P1(1),P1(2),'ok','LineWidth',5)
hold on
plot(P2(1),P2(2),'om','LineWidth',5)
plot(0,0,'ok','LineWidth',10)
xlim([-2.5 2.5])
ylim([-2.5 2.5])
grid minor
plot([0 P1(1)], [0 P1(2)],'r','LineWidth',5)
plot([P1(1) P2(1)], [P1(2) P2(2)],'b','LineWidth',5)
hold off
title(strcat('Time = ',num2str(t(1,i))),'Interpreter','latex')
xlabel('X axis (m)','Interpreter','latex')
ylabel('Y axis (m)','Interpreter','latex')
set(gca,'FontSize',18)
pause(0.0000000000000000001);
F(i) = getframe(gcf) ;
end

% create the video writer with 30 fps
  writerObj = VideoWriter('animation_Q2_a.avi');
  writerObj.FrameRate = 30;
  % set the seconds per image
% open the video writer
open(writerObj);
% write the frames to the video
for i=1:length(F)
    % convert the image to a frame
    frame = F(i) ;
    writeVideo(writerObj, frame);
end
% close the writer object
close(writerObj);
```

Function for trajectory generation

```
function [t,q,dq,ddq] = trecgen(t0,dt,tf,q0,dq0,q1,dq1)
%% Function for trjectory generation using polynomial equation
t=t0:dt:tf;
q= (q0) + (dq0*t) + (((3*(q1 - q0))-(2*dq0 + dq1)*tf)/tf^2)*t.^2 + (((2*(q0 - q1))-(dq0 + dq1)*tf)/tf^3)*t.^3 ; %Position
```

```
dq=diff(q)/dt; %Velocity
ddq=diff(dq)/dt; %Acceleration
end
```

4 Problem 3

(Dynamic Simulation) Use the manipulator parameters as in Q1 and the initial condition (at t = 0), IC = [q 1 q 2 q d1 q d2] = [0 0 0 0]

4.1 Part A

A. Solve Eq. (1) using a numerical solver, say ode45, for a span of time, [t i t f] = [0 10] and for joint torque values computed in Q2 (b). Compare the computed joint positions and velocities with the desired trajectory as planned in Q2 (a). Plot the results and discuss.

To solve the equation as the function of time the curve fitting on computed torque needed, so we can have torque as the function of time.

$\tau_1 = (-3.474e-05)*t^6 + (0.0008649)*t^5 + (-0.004596)*t^4 + (-0.01598)*t^3 + (0.03615)*t^2 + (-0.0571)*t + 19.76$

$\tau_2 = (-2.676e-05)*t^6 + (0.0006413)*t^5 + (-0.003133)*t^4 + (-0.01392)*t^3 + (0.03229)*t^2 + (-0.03644)*t + 4.956$

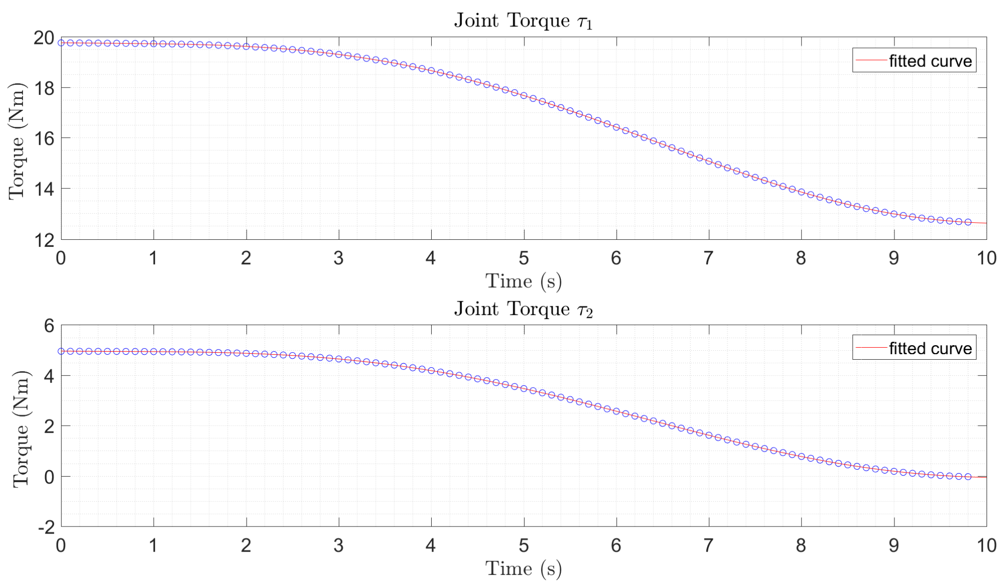

Figure 12: Curve fitting on planned torque

Matlab code for torque computation

```
%% Initialization
clear
close all
clc
%% Trejectory genration
[~,q0,dq0,ddq0]=trecgen(0,1/10,10,0,0,pi/6,0);  %Trejectory generation for theta 1 0 to pi/6
[t,q1,dq1,ddq1]=trecgen(0,1/10,10,0,0,pi/3,0);  %Trejectory generation for theta 2 0 to pi/3


x=[q0 dq0 q1 dq1];

%% Initialization of parameters
m2=1; m1=1; l1=1; l2=1; g=9.81;

%% Equation of motion

for i=1:1:length(ddq0)
M11=((((m1/3) + m2)*l1^2) +((m2/3)*l2^2) + (m2*l1*l2*cos(q1(1,i))));
M12=(m2*(((l2^2)/3) + (0.5*l1*l2*cos(q1(1,i)))));
M21=M12;
M22=((1/3)*m2*l2*l2);

```

```
H1 =((-m2*l1*l2*sin(q1(1,i))*dq0(1,i)*dq1(1,i) - (0.5*m2*l1*l2*sin(q1(1,i))*dq1(1,i)*dq1(1,i))));
H2 = (0.5 * m2* l1*l2*sin(q1(1,i))*dq0(1,i)*dq0(1,i));

G1=( ((((0.5*m1) + m2)*l1*cos(q0(1,i))) + (0.5*m2*l2*cos(q0(1,i)+q1(1,i))))*g);
G2=0.5*m2*l2*cos(q0(1,i)+q1(1,i))*g;

Tau1(1,i)= M11*ddq0(1,i) + M12*ddq1(1,i) + H1 +G1;
Tau2(1,i)= M21*ddq0(1,i) + M22*ddq1(1,i) + H2 +G2;
end

%% Curve fitting and equation
f1= fit(t(1,1:end-2)',Tau1','poly6');
cf1=coeffvalues(f1);
f2= fit(t(1,1:end-2)',Tau2','poly6');
cf2=coeffvalues(f2);

figure('units','normalized','outerposition',[0 0 1 1])
subplot(211)
plot(t(1,1:end-2),Tau1,'ob','LineWidth',1)
hold on
plot(f1)
title('Joint Torque $\tau_1$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Torque (Nm) ','Interpreter','latex')
set(gca,'FontSize',18)
grid minor


subplot(212)
plot(t(1,1:end-2),Tau2,'ob','LineWidth',1)
hold on
plot(f2)
title('Joint Torque $\tau_2$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Torque (Nm) ','Interpreter','latex')
set(gca,'FontSize',18)
grid minor
saveas(gcf,'Q3_a_CT.png')
```

Comparison of the computed joint positions and velocities with the desired trajectory as planned.

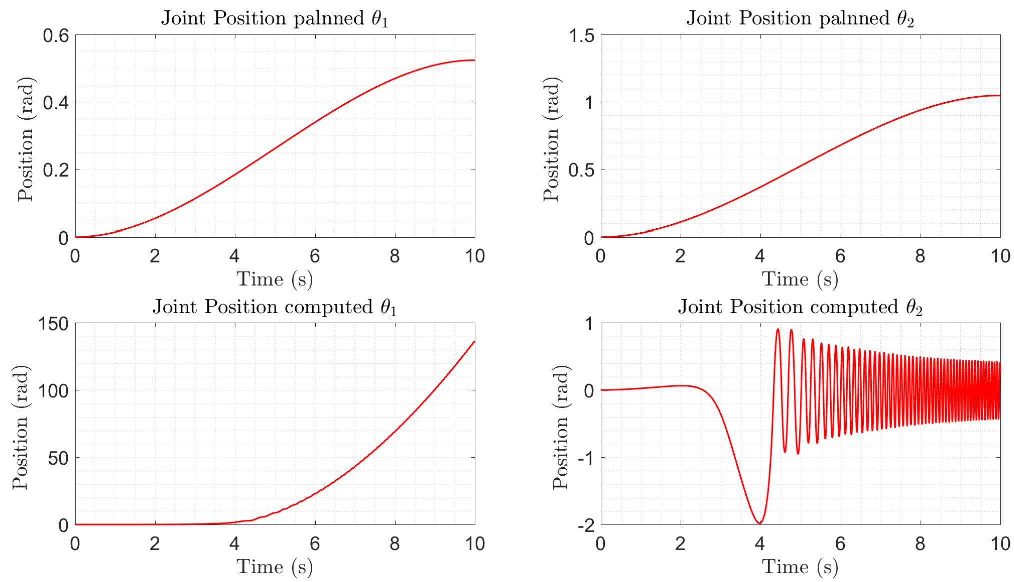

Figure 13: Comparison of the computed joint positions with the desired trajectory as planned

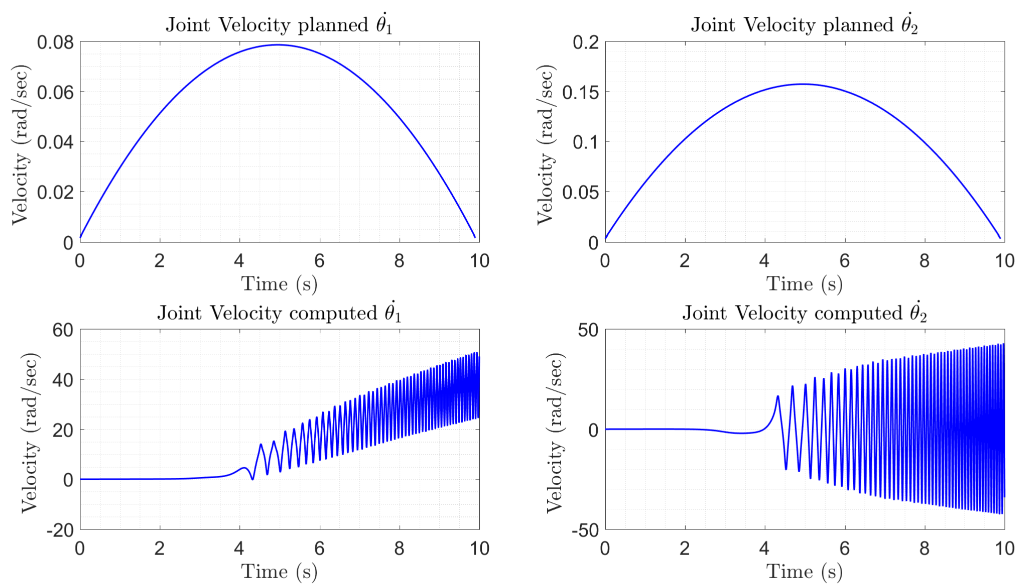

Figure 14: Comparison of the computed joint velocities with the desired trajectory as planned

As we know that this system is chaotic system and exhibits rich dynamic behavior with a strong sensitivity to initial conditions, And we are estimating the torques for the system as a function of time if there is very small error in estimation the entire response will change due to propagation of that error. The system is also coupled and exhibits some non linearity so directed feed forward approach creates the problem due avoiding small estimation errors. Due to the nonlinear behavior and couple between the link for the same torque different trajectories can be generated. The propagation of errors and non linearity as well as the couple system behaviors, we can see that the computed potion and velocities are very different from the planned position and velocities. The joint position 1 rapidly and exponentially increase with the time with some oscillations and the joint position 2 is oscillating after some period of time. The joint velocities are also exhibits the oscillatory behavior with some offsets.

This problem can be solved by using PID or PD controller with the system which will take care of errors. The potential approach can be presented as listed below.

Animation of PID Control

4.1.1 Matlab Program for PID

```
% ME 639: Introduction to robotics
% Midsem exam : PID Control
%                18 Oct 2018
%
% Author: Shail Jadav 18310039
%% Initialization
clear
close all
clc
%%


[t,x]=ode45('ode_solver_script_pidcontrol',[0,10],[0,0,0,0]);

m1=1; m2=1; l1=1; l2=1; g=9.81;

th1=x(:,1); dth1=x(:,2); th2=x(:,3); dth2=x(:,4);



figure('units','normalized','outerposition',[0 0 1 1])

c=1;
for i=1:10:length(x)

    theta1=x(i,1);
    dtheta1=x(i,2);
    theta2=x(i,3);
    dtheta2=x(i,4);

    l1=1; %Input the l length
    l2=1; %Input the l length



    % Homogeneus transformation matrix
    H01 = [cos(theta1) -sin(theta1) 0 l1*cos(theta1);sin(theta1) cos(theta1) 0 l1*sin(theta1);0 0
    1 0;0 0 0 1]; %Frame 0 to 1 tranformation
    H12 = [cos(theta2) -sin(theta2) 0 l2*cos(theta2);sin(theta2) cos(theta2) 0 l2*sin(theta2);0 0
    1 0;0 0 0 1]; %Frame 1 to 2 tranformation


    H02=H01*H12;      %Frame 0 to 2 tranformation


    O=[0,0];                  %Joint 1 position
    P1=[H01(1,4) H01(2,4)];   %Joint 2 position
    P2=[H02(1,4) H02(2,4)];   %Joint 3 position


    Orn= atan2(H02(2,1),H02(1,1));  %Orientation of end effector
    Orn=(Orn)*(180/pi);

    plot(P1(1),P1(2),'ok','LineWidth',5)
    hold on
    plot(P2(1),P2(2),'om','LineWidth',5)
    plot(0,0,'ok','LineWidth',10)
    xlim([-2.5 2.5])
    ylim([-2.5 2.5])
    grid minor
    plot([0 P1(1)], [0 P1(2)],'r','LineWidth',5)
    plot([P1(1) P2(1)], [P1(2) P2(2)],'b','LineWidth',5)
    hold off
    title(strcat('Time = ',num2str(t(i,1))),'Interpreter','latex')
    xlabel('X axis (m)','Interpreter','latex')
    ylabel('Y axis (m)','Interpreter','latex')
```

```
    set(gca,'FontSize',18)
    pause(0.00000000000000000000000000000000000000001);


    F(c) = getframe(gcf) ;
    c=c+1;
end

% create the video writer with 30 fps
writerObj = VideoWriter('PID.avi');
writerObj.FrameRate = 30;
% set the seconds per image
% open the video writer
open(writerObj);
% write the frames to the video
for i=1:length(F)
    % convert the image to a frame
    frame = F(i) ;
    writeVideo(writerObj, frame);
end
% close the writer object
close(writerObj);
```

The solver using PID controller

```
function Out= ode_solver_script_pidcontrol(t,x)

m2=1; m1=1; l1=1; l2=1; g=9.81;

th1f=pi/6;
th2f=pi/3;

tau1=(-3.474e-05)*t.^6 + (0.0008649 )*t.^5 + (-0.004596)*t.^4 + (-0.01598  )*t.^3 + ( 0.03615)*t.^2 + (-0.0571 )*t +19.76; %Input torque is zero
tau2=(-2.676e-05 )*t.^6 + (0.0006413 )*t.^5 + (-0.003133)*t.^4 + (-0.01392  )*t.^3 + ( 0.03229  )*t.^2 + (-0.03644 )*t +4.956; %second torque is the same as torque 1




M11=((((m1/3) + m2)*l1^2) +((m2/3)*l2^2) + (m2*l1*l2*cos(x(3))));
M12=(m2*(((l2^2)/3) + (0.5*l1*l2*cos(x(3)))));
M21=M12;
M22=((1/3)*m2*l2*l2);


H1 =((-m2*l1*l2*sin(x(3))*x(2)*x(4)) - (0.5*m2*l1*l2*sin(x(3))*x(4)*x(4)));
H2 = (0.5 * m2* l1*l2*sin(x(3))*x(2)*x(2));

G1=( ((((0.5*m1) + m2)*l1*cos(x(1))) + (0.5*m2*l2*cos(x(1)+x(3))))*g);
G2=0.5*m2*l2*cos(x(1)+x(3))*g;

T=[tau1; tau2];

M=[M11 M12;M21 M22];

HG = [H1 + G1; H2 + G2];

S=[0 ; 0.001*x(3)];

%PID Controller
P=[853*(th1f - x(1)) - 100*x(2) + 0.5*x(1) ;553*(th2f - x(3)) - 100*x(4)  + 0.5*x(3)];



ddth = (inv(M)) * (T  - HG -S +P)     ;

OP=zeros(4,1);

OP(1)=x(2);
OP(2)=ddth(1);
OP(3)=x(4);
OP(4)=ddth(2);

```

```
Out=OP;
end
```

4.1.2 Matlab Program

```
% ME 639: Introduction to robotics
% Midsem exam : Question 3 (a)
%               3 Oct 2018
%   Solve Eq. (1) using a numerical solver, say ode45, for a span of time, [t i t f ] = [0 10] and
%   for joint torque values computed in Q2 (b). Compare the computed joint positions and
%   velocities with the desired trajectory as planned in Q2 (a). Plot the results and discuss.
%
% Author: Shail Jadav 18310039
%% Initialization
clear
close all
clc
%% ODE solver computed

[t,x]=ode45('ode_solver_script_q3_a',[0,10],[0,0,0,0]);
m1=1; m2=1; l1=1; l2=1; g=9.81;

th1=x(:,1); dth1=x(:,2); th2=x(:,3); dth2=x(:,4);

%% Trejectory genration plannned
[~,q0,dq0,ddq0]=trecgen(0,1/10,10,0,0,pi/6,0);  %Trejectory generation for theta 1 0 to pi/6
[tq,q1,dq1,ddq1]=trecgen(0,1/10,10,0,0,pi/3,0);  %Trejectory generation for theta 2 0 to pi/3



%% Display The Results

figure('units','normalized','outerposition',[0 0 1 1])
subplot(223)
plot(t,th1,'r','LineWidth',1.5)
title('Joint Position computed $\theta_1$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Position (rad) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor


subplot(224)
plot(t,th2,'r','LineWidth',1.5)
title('Joint Position computed $\theta_2$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Position (rad) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor
set(gca)


subplot(221)
plot(tq,q0,'r','LineWidth',1.5)
title('Joint Position palnned $\theta_1$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Position (rad) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor


subplot(222)
plot(tq,q1,'r','LineWidth',1.5)
title('Joint Position palnned $\theta_2$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Position (rad) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor
set(gca)
saveas(gcf,'Q3_a_JP.png')

figure('units','normalized','outerposition',[0 0 1 1])
```

```

subplot(223)
plot(t,dth1,'b','LineWidth',1.5)
title('Joint Velocity computed $\dot{\theta_1}$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Velocity (rad/sec) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor


subplot(224)
plot(t,dth2,'b','LineWidth',1.5)
title('Joint Velocity computed $\dot{\theta_2}$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Velocity (rad/sec) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor
set(gca)




subplot(221)
plot(tq(1,1:end-1),dq0,'b','LineWidth',1.5)
title('Joint Velocity planned $\dot{\theta_1}$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Velocity (rad/sec) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor


subplot(222)
plot(tq(1,1:end-1),dq1,'b','LineWidth',1.5)
title('Joint Velocity planned $\dot{\theta_2}$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Velocity (rad/sec) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor
set(gca)

saveas(gcf,'Q3_a_JV.png')
```

Function of ODE

```
function Out= ode_solver_script_q3_a(t,x)

%% Input parameters
m2=1; m1=1; l1=1; l2=1; g=9.81;

tau1=(-3.474e-05)*t.^6 + (0.0008649 )*t.^5 + (-0.004596)*t.^4 + (-0.01598  )*t.^3 + ( 0.03615)*t.^2 + (-0.0571 )*t +19.76; %Input torque is zero
tau2=(-2.676e-05 )*t.^6 + (0.0006413 )*t.^5 + (-0.003133)*t.^4 + (-0.01392  )*t.^3 + ( 0.03229  )*t.^2 + (-0.03644 )*t +4.956; %second torque is the same as torque 1


%% Equation of motion
M11=((((m1/3) + m2)*l1^2) +((m2/3)*l2^2) + (m2*l1*l2*cos(x(3))));
M12=(m2*(((l2^2)/3) + (0.5*l1*l2*cos(x(3)))));
M21=M12;
M22=((1/3)*m2*l2*l2);


H1 =((-m2*l1*l2*sin(x(3))*x(2)*x(4)) - (0.5*m2*l1*l2*sin(x(3))*x(4)*x(4)));
H2 = (0.5 * m2* l1*l2*sin(x(3))*x(2)*x(2));

G1=( ((((0.5*m1) + m2)*l1*cos(x(1))) + (0.5*m2*l2*cos(x(1)+x(3))))*g);
G2=0.5*m2*l2*cos(x(1)+x(3))*g;

T=[tau1; tau2];

```

```
M=[M11 M12;M21 M22];

HG = [H1 + G1; H2 + G2];

%% Equation in terms of acceleration

ddth = (inv(M)) * (T - HG )     ;

OP=zeros(4,1);

%% Output
OP(1)=x(2); %Intergretion of velocity will give the position for theta 1
OP(2)=ddth(1);%Intergretion of acceleration will give the velocity for theta 1
OP(3)=x(4); %Intergretion of velocity will give the position  for theta 2
OP(4)=ddth(2); %Intergretion of acceleration will give the velocity for theta 2


Out=OP; % Output

end
```

4.2 Part B

Repeat above part (a) when a point mass, m e = 1 , is added at the center of mass of the link 2. Plot the results and discuss.

To solve the equation as the function of time the curve fitting on computed torque needed, so we can have torque as the function of time.

$\tau_1 = (-6.672e{-}05)*t.^6+(0.001649)*t.^5+(-0.00861)*t.^4+(-0.0318)*t.^3+(0.07193)*t.^2+(-0.1118)*t+34.6$

$\tau_2 = (-5.353e{-}05)*t.^6+(0.001283)*t.^5+(-0.006265)*t.^4+(-0.02784)*t.^3+(0.06457)*t.^2+(-0.07288)*t+9.912$

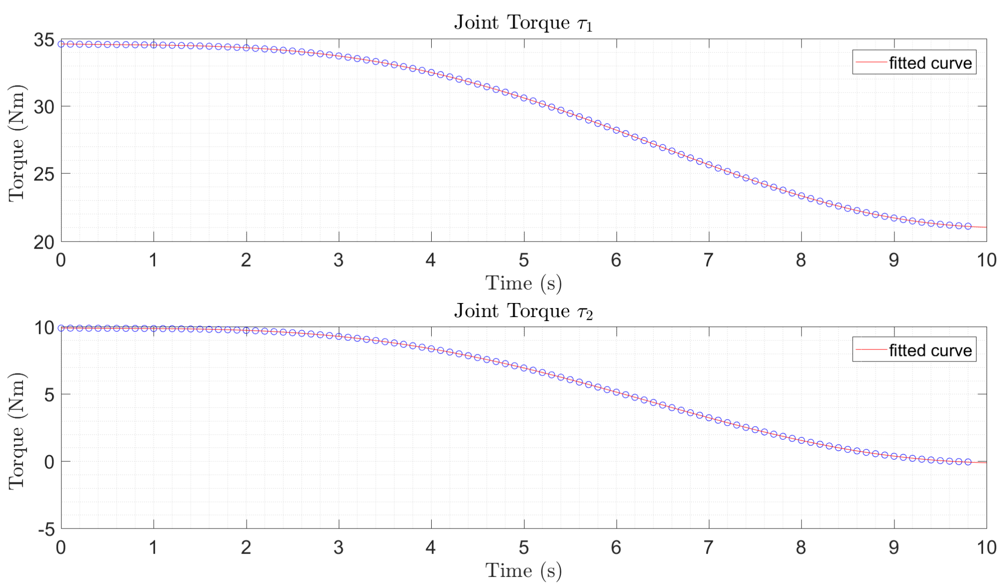

Figure 15: Curve fitting on planned torque

Matlab code

```
%% Initialization
clear
close all
clc
%% Trejectory genration
[~,q0,dq0,ddq0]=trecgen(0,1/10,10,0,0,pi/6,0);  %Trejectory generation for theta 1 0 to pi/6
[t,q1,dq1,ddq1]=trecgen(0,1/10,10,0,0,pi/3,0);  %Trejectory generation for theta 2 0 to pi/3


x=[q0 dq0 q1 dq1];
```

```
%% Initialization of parameters
m2=2; m1=1; l1=1; l2=1; g=9.81;


%% Equation of motion

for i=1:1:length(ddq0)
M11=((((m1/3) + m2)*l1^2) +((m2/3)*l2^2) + (m2*l1*l2*cos(q1(1,i))));
M12=(m2*(((l2^2)/3) + (0.5*l1*l2*cos(q1(1,i)))));
M21=M12;
M22=((1/3)*m2*l2*l2);


H1 =((-m2*l1*l2*sin(q1(1,i))*dq0(1,i)*dq1(1,i) - (0.5*m2*l1*l2*sin(q1(1,i))*dq1(1,i)*dq1(1,i))));
H2 = (0.5 * m2* l1*l2*sin(q1(1,i))*dq0(1,i)*dq0(1,i));

G1=( ((((0.5*m1) + m2)*l1*cos(q0(1,i))) + (0.5*m2*l2*cos(q0(1,i)+q1(1,i))))*g);
G2=0.5*m2*l2*cos(q0(1,i)+q1(1,i))*g;

Tau1(1,i)= M11*ddq0(1,i) + M12*ddq1(1,i) + H1 +G1;
Tau2(1,i)= M21*ddq0(1,i) + M22*ddq1(1,i) + H2 +G2;
end


%% Curve fitting and equation
f1= fit(t(1,1:end-2)',Tau1','poly6');
cf1=coeffvalues(f1);
f2= fit(t(1,1:end-2)',Tau2','poly6');
cf2=coeffvalues(f2);

figure('units','normalized','outerposition',[0 0 1 1])
subplot(211)
plot(t(1,1:end-2),Tau1,'ob','LineWidth',1)
hold on
plot(f1)
title('Joint Torque $\tau_1$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Torque (Nm) ','Interpreter','latex')
set(gca,'FontSize',18)
grid minor


subplot(212)
plot(t(1,1:end-2),Tau2,'ob','LineWidth',1)
hold on
plot(f2)
title('Joint Torque $\tau_2$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Torque (Nm) ','Interpreter','latex')
set(gca,'FontSize',18)
grid minor
saveas(gcf,'Q3_b_CT.png')
```

Comparison of the computed joint positions and velocities with the desired trajectory as planned.

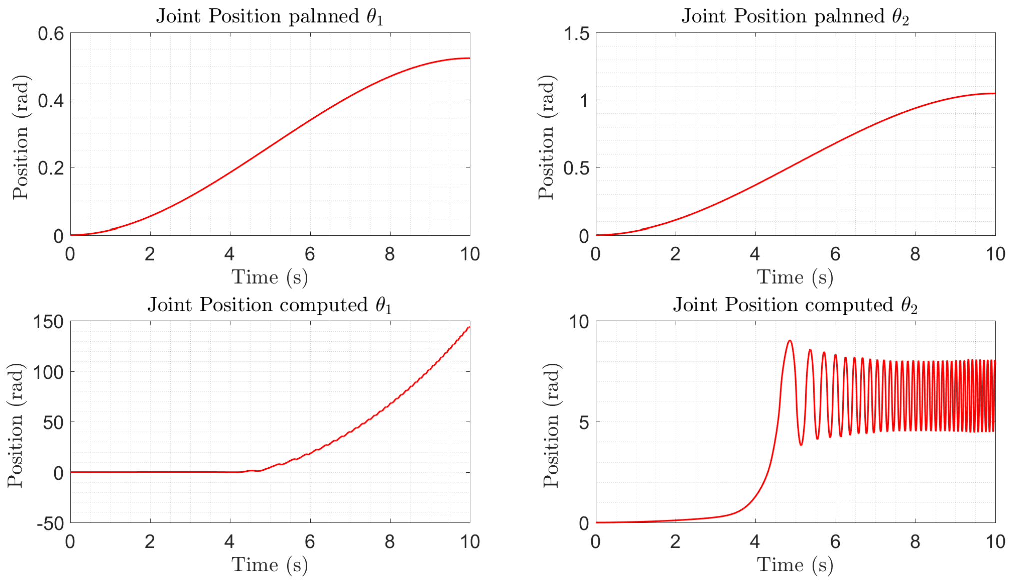

Figure 16: Comparison of the computed joint positions with the desired trajectory as planned

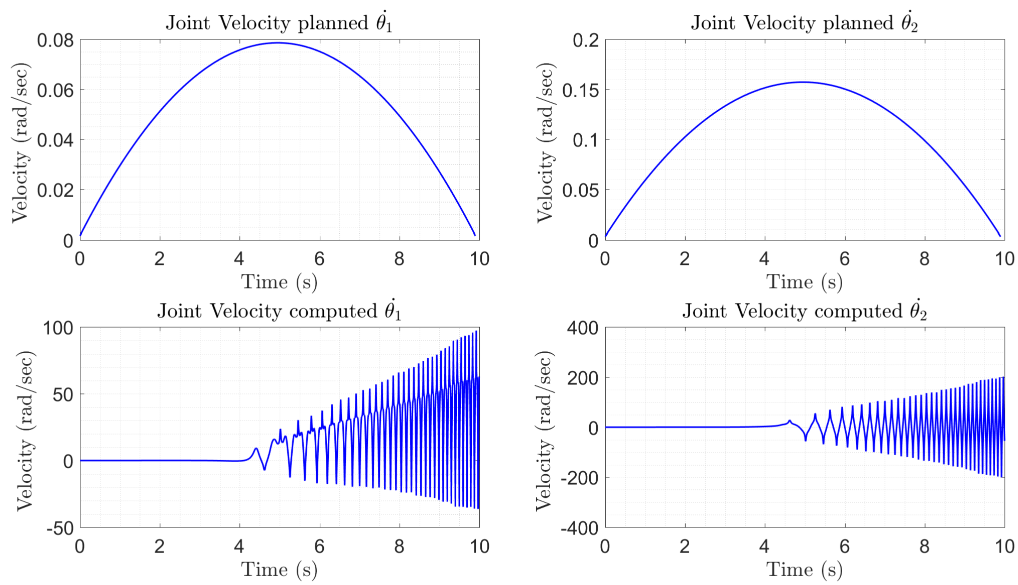

Figure 17: Comparison of the computed joint velocities with the desired trajectory as planned

Due to higher mass than the previous condition the holding torque increase significantly of both joints. Due to propagation of errors as well as the non itineraries the system behave similar to the previous case, we can see that the computed potion and velocities are very different from the planned position and velocities. The joint position 1 rapidly and exponentially increase with the time with some oscillations and the joint position 2 is oscillating after some period of time. The joint velocities are also exhibits the oscillatory behavior with some offsets.

4.2.1 Matlab Program

```
% ME 639: Introduction to robotics
% Midsem exam : Question 3 (b)
%                3 Oct 2018
%Repeat above part (a) when a point mass, m e = 1 , is added at the center of mass of the
%link 2. Plot the results and discuss.
%
% Author: Shail Jadav 18310039
%% Initialization
clear
close all
clc
%% ODE solver computed

[t,x]=ode45('ode_solver_script_q3_b',[0,10],[0,0,0,0]);  % Time span 0 to 10 IC=[pi/4 0 pi/4 0]
    Theta1=pi/4 Theta2=pi/4

m1=1; m2=2; l1=1; l2=1; g=9.81;

th1=x(:,1); dth1=x(:,2); th2=x(:,3); dth2=x(:,4);

%% Trejectory genration plannned
[~,q0,dq0,ddq0]=trecgen(0,1/10,10,0,0,pi/6,0);  %Trejectory generation for theta 1 0 to pi/6
[tq,q1,dq1,ddq1]=trecgen(0,1/10,10,0,0,pi/3,0);  %Trejectory generation for theta 2 0 to pi/3



%% Display The Results

figure('units','normalized','outerposition',[0 0 1 1])
subplot(223)
plot(t,th1,'r','LineWidth',1.5)
title('Joint Position computed $\theta_1$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Position (rad) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor


subplot(224)
plot(t,th2,'r','LineWidth',1.5)
title('Joint Position computed $\theta_2$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Position (rad) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor
set(gca)


subplot(221)
plot(tq,q0,'r','LineWidth',1.5)
title('Joint Position palnned $\theta_1$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Position (rad) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor


subplot(222)
plot(tq,q1,'r','LineWidth',1.5)
title('Joint Position palnned $\theta_2$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Position (rad) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor
set(gca)
saveas(gcf,'Q3_b_JP.png')

```

```
figure('units','normalized','outerposition',[0 0 1 1])

subplot(223)
plot(t,dth1,'b','LineWidth',1.5)
title('Joint Velocity computed $\dot{\theta_1}$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Velocity (rad/sec) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor


subplot(224)
plot(t,dth2,'b','LineWidth',1.5)
title('Joint Velocity computed $\dot{\theta_2}$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Velocity (rad/sec) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor
set(gca)




subplot(221)
plot(tq(1,1:end-1),dq0,'b','LineWidth',1.5)
title('Joint Velocity planned $\dot{\theta_1}$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Velocity (rad/sec) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor


subplot(222)
plot(tq(1,1:end-1),dq1,'b','LineWidth',1.5)
title('Joint Velocity planned $\dot{\theta_2}$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Velocity (rad/sec) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor
set(gca)

saveas(gcf,'Q3_b_JV.png')
```

Function of ODE

```
function Out= ode_solver_script_q3_b(t,x)

%% Input parameters
m2=2; m1=1; l1=1; l2=1; g=9.81;

tau1=(-6.672e-05)*t.^6 + (0.001649 )*t.^5 + (-0.00861 )*t.^4 + ( -0.0318 )*t.^3 + ( 0.07193)*t.^2
    + (-0.1118 )*t + 34.6 ; %Input torque is zero
tau2=(-5.353e-05 )*t.^6 + (0.001283 )*t.^5 + (-0.006265 )*t.^4 + (-0.02784 )*t.^3 + ( 0.06457 )
    *t.^2 + ( -0.07288 )*t +9.912 ; %second torque is the same as torque 1


%% Equation of motion
M11=((((m1/3) + m2)*l1^2) +((m2/3)*l2^2) + (m2*l1*l2*cos(x(3))));
M12=(m2*(((l2^2)/3) + (0.5*l1*l2*cos(x(3)))));
M21=M12;
M22=((1/3)*m2*l2*l2);


H1 =((-m2*l1*l2*sin(x(3))*x(2)*x(4)) - (0.5*m2*l1*l2*sin(x(3))*x(4)*x(4)));
H2 = (0.5 * m2* l1*l2*sin(x(3))*x(2)*x(2));

G1=( ((((0.5*m1) + m2)*l1*cos(x(1))) + (0.5*m2*l2*cos(x(1)+x(3))))*g);
G2=0.5*m2*l2*cos(x(1)+x(3))*g;

T=[tau1; tau2];
```

```
M=[M11 M12;M21 M22];

HG = [H1 + G1; H2 + G2];

%% Equation in terms of acceleration

ddth = (inv(M)) * (T - HG )    ;

OP=zeros(4,1);

%% Output
OP(1)=x(2); %Intergretion of velocity will give the position for theta 1
OP(2)=ddth(1);%Intergretion of acceleration will give the velocity for theta 1
OP(3)=x(4); %Intergretion of velocity will give the position  for theta 2
OP(4)=ddth(2); %Intergretion of acceleration will give the velocity for theta 2


Out=OP; % Output

end
```

4.3 Part C

Animate the 2R manipulator for part (a) and (b)

Part A

Please click on following links (Blue colour)

Animation of Q3 A Planned trajectory

Animation of Q3 A Computed trajectory

4.3.1 Matlab Program

```
% ME 639: Introduction to robotics
% Midsem exam : Question 3 (a) Animation
%                3 Oct 2018
%   Solve Eq. (1) using a numerical solver, say ode45, for a span of time, [t i t f ] = [0 10] and
%   for joint torque values computed in Q2 (b). Compare the computed joint positions and
%   velocities with the desired trajectory as planned in Q2 (a). Plot the results and discuss.
%
% Author: Shail Jadav 18310039
%% Initialization
clear
close all
clc
%% ODE solver computed

[t,x]=ode45('ode_solver_script_q3_a',[0,10],[0,0,0,0]);
m1=1; m2=1; l1=1; l2=1; g=9.81;

th1=x(:,1); dth1=x(:,2); th2=x(:,3); dth2=x(:,4);

%% Trejectory genration plannned
[~,q0,dq0,ddq0]=trecgen(0,1/10,10,0,0,pi/6,0);  %Trejectory generation for theta 1 0 to pi/6
[tq,q1,dq1,ddq1]=trecgen(0,1/10,10,0,0,pi/3,0);  %Trejectory generation for theta 2 0 to pi/3


%% Planned
figure('units','normalized','outerposition',[0 0 1 1])
for i=1:1:length(q0)

theta1=q0(1,i);
theta2=q1(1,i);


l1=1; %Input the l length
l2=1; %Input the l length



% Homogeneus transformation matrix
H01 = [cos(theta1) -sin(theta1) 0 l1*cos(theta1);sin(theta1) cos(theta1) 0 l1*sin(theta1);0 0 1
     0;0 0 0 1]; %Frame 0 to 1 tranformation
```

```
H12 = [cos(theta2) -sin(theta2) 0 l2*cos(theta2);sin(theta2) cos(theta2) 0 l2*sin(theta2);0 0 1
    0;0 0 0 1]; %Frame 1 to 2 tranformation


H02=H01*H12;       %Frame 0 to 2 tranformation


O=[0,0];                    %Joint 1 position
P1=[H01(1,4) H01(2,4)];     %Joint 2 position
P2=[H02(1,4) H02(2,4)];     %Joint 3 position


Orn= atan2(H02(2,1),H02(1,1));  %Orientation of end effector
Orn=(Orn)*(180/pi);
%subplot(221)
plot(P1(1),P1(2),'ok','LineWidth',5)
hold on
plot(P2(1),P2(2),'om','LineWidth',5)
plot(0,0,'ok','LineWidth',10)
xlim([-2.5 2.5])
ylim([-2.5 2.5])
grid minor
plot([0 P1(1)], [0 P1(2)],'r','LineWidth',5)
plot([P1(1) P2(1)], [P1(2) P2(2)],'b','LineWidth',5)
hold off
title(strcat('Time = ',num2str(tq(1,i))),'Interpreter','latex')
xlabel('X axis (m)','Interpreter','latex')
ylabel('Y axis (m)','Interpreter','latex')
set(gca,'FontSize',18)
pause(0.0000000000000000001);
F(i) = getframe(gcf) ;
end

% create the video writer with 30 fps
  writerObj = VideoWriter('Q3_planned_a.avi');
  writerObj.FrameRate = 30;
  % set the seconds per image
% open the video writer
open(writerObj);
% write the frames to the video
for i=1:length(F)
    % convert the image to a frame
    frame = F(i) ;
    writeVideo(writerObj, frame);
end
% close the writer object
close(writerObj);


%% Computed
figure('units','normalized','outerposition',[0 0 1 1])
c=1;
for i=1:2:length(th1)

theta1=th1(i,1);
theta2=th2(i,1);


l1=1; %Input the l length
l2=1; %Input the l length



% Homogeneus transformation matrix
H01 = [cos(theta1) -sin(theta1) 0 l1*cos(theta1);sin(theta1) cos(theta1) 0 l1*sin(theta1);0 0 1
    0;0 0 0 1]; %Frame 0 to 1 tranformation
H12 = [cos(theta2) -sin(theta2) 0 l2*cos(theta2);sin(theta2) cos(theta2) 0 l2*sin(theta2);0 0 1
    0;0 0 0 1]; %Frame 1 to 2 tranformation


H02=H01*H12;       %Frame 0 to 2 tranformation
```

```
O=[0,0];                    %Joint 1 position
P1=[H01(1,4) H01(2,4)];     %Joint 2 position
P2=[H02(1,4) H02(2,4)];     %Joint 3 position


Orn= atan2(H02(2,1),H02(1,1)); %Orientation of end effector
Orn=(Orn)*(180/pi);
%subplot(221)
plot(P1(1),P1(2),'ok','LineWidth',5)
hold on
plot(P2(1),P2(2),'om','LineWidth',5)
plot(0,0,'ok','LineWidth',10)
xlim([-2.5 2.5])
ylim([-2.5 2.5])
grid minor
plot([0 P1(1)], [0 P1(2)],'r','LineWidth',5)
plot([P1(1) P2(1)], [P1(2) P2(2)],'b','LineWidth',5)
hold off
title(strcat('Time = ',num2str(t(i,1))),'Interpreter','latex')
xlabel('X axis (m)','Interpreter','latex')
ylabel('Y axis (m)','Interpreter','latex')
set(gca,'FontSize',18)
pause(0.0000000000000000001);
F(c) = getframe(gcf) ;
c=c+1;
end

% create the video writer with 30 fps
  writerObj = VideoWriter('Q3_computed_a.avi');
  writerObj.FrameRate = 30;
  % set the seconds per image
% open the video writer
open(writerObj);
% write the frames to the video
for i=1:length(F)
    % convert the image to a frame
    frame = F(i) ;
    writeVideo(writerObj, frame);
end
% close the writer object
close(writerObj);
```

Part B

Animation of Q3 B Computed trajectory

```
% ME 639: Introduction to robotics
% Midsem exam : Question 3 (b)
%                3 Oct 2018
%   Repeat above part (a) when a point mass, m e = 1 , is added at the center of mass of the
%   link 2. Plot the results and discuss.
%
% Author: Shail Jadav 18310039
%% Initialization
clear
close all
clc
%% ODE solver computed

[t,x]=ode45('ode_solver_script_q3_b',[0,10],[0,0,0,0]);

th1=x(:,1); dth1=x(:,2); th2=x(:,3); dth2=x(:,4);

%% Trejectory genration plannned
[~,q0,dq0,ddq0]=trecgen(0,1/10,10,0,0,pi/6,0); %Trejectory generation for theta 1 0 to pi/6
[tq,q1,dq1,ddq1]=trecgen(0,1/10,10,0,0,pi/3,0); %Trejectory generation for theta 2 0 to pi/3


%% Computed
figure('units','normalized','outerposition',[0 0 1 1])
c=1;
for i=1:2:length(th1)

theta1=th1(i,1);
theta2=th2(i,1);
```

```
l1=1; %Input the l length
l2=1; %Input the l length



% Homogeneus transformation matrix
H01 = [cos(theta1) -sin(theta1) 0 l1*cos(theta1);sin(theta1) cos(theta1) 0 l1*sin(theta1);0 0 1
    0;0 0 0 1]; %Frame 0 to 1 tranformation
H12 = [cos(theta2) -sin(theta2) 0 l2*cos(theta2);sin(theta2) cos(theta2) 0 l2*sin(theta2);0 0 1
    0;0 0 0 1]; %Frame 1 to 2 tranformation


H02=H01*H12;      %Frame 0 to 2 tranformation


O=[0,0];                    %Joint 1 position
P1=[H01(1,4) H01(2,4)];     %Joint 2 position
P2=[H02(1,4) H02(2,4)];     %Joint 3 position


Orn= atan2(H02(2,1),H02(1,1));  %Orientation of end effector
Orn=(Orn)*(180/pi);
%subplot(221)
plot(P1(1),P1(2),'ok','LineWidth',5)
hold on
plot(P2(1),P2(2),'om','LineWidth',5)
plot(0,0,'ok','LineWidth',10)
xlim([-2.5 2.5])
ylim([-2.5 2.5])
grid minor
plot([0 P1(1)], [0 P1(2)],'r','LineWidth',5)
plot([P1(1) P2(1)], [P1(2) P2(2)],'b','LineWidth',5)
hold off
title(strcat('Time = ',num2str(t(i,1))),'Interpreter','latex')
xlabel('X axis (m)','Interpreter','latex')
ylabel('Y axis (m)','Interpreter','latex')
set(gca,'FontSize',18)
pause(0.0000000000000000001);
F(c) = getframe(gcf) ;
c=c+1;
end

% create the video writer with 30 fps
  writerObj = VideoWriter('Q3_computed_b.avi');
  writerObj.FrameRate = 30;
  % set the seconds per image
% open the video writer
open(writerObj);
% write the frames to the video
for i=1:length(F)
    % convert the image to a frame
    frame = F(i) ;
    writeVideo(writerObj, frame);
end
% close the writer object
close(writerObj);
```

5 Problem 4

(Under-actuated System) Use the manipulator parameters as in Q1 and the initial condition (at t = 0), IC = [q 1 q 2 q d1 q d2] = [0 0 0 0] . Consider that the second joint is not actuated, i.e., T 2 = 0 always.

5.1 Part A

For Torque 1 as computed in Q2(b), solve Eq. (1) using a numerical solver, say ode45, over a span of time, [t i t f] = [0 10] . Compare the computed joint positions and velocities with the desired trajectory as planned in Q2 (a). Plot the results and discuss.

Comparison of the computed joint positions and velocities with the desired trajectory as planned.

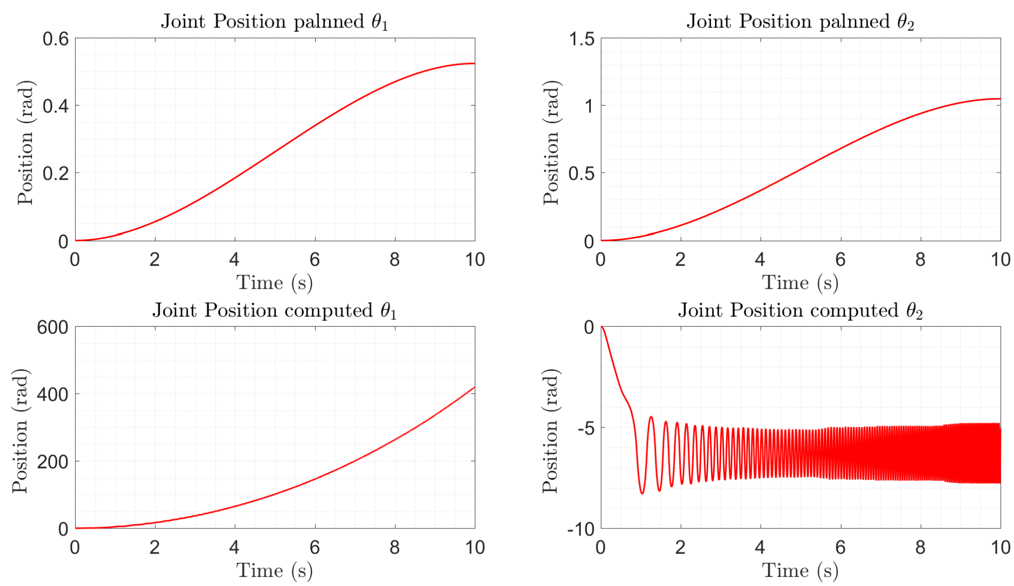

Figure 18: Comparison of the computed joint positions with the desired trajectory as planned

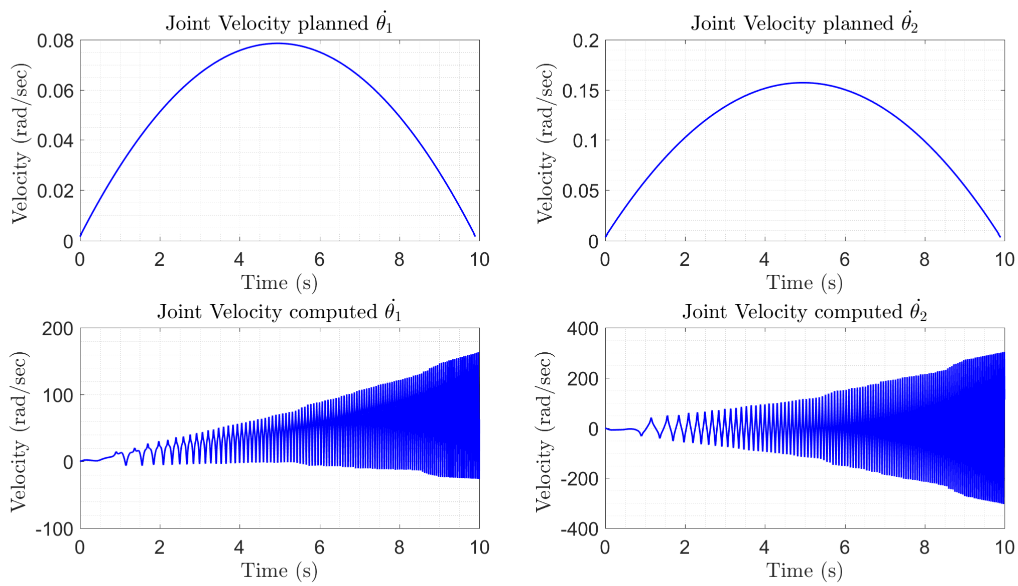

Figure 19: Comparison of the computed joint velocities with the desired trajectory as planned

As the τ_2 is zero the the joint 2 is free to rotate, the τ_1 is present so the joint position 1 rapidly and exponentially increase with the time with some oscillations and the joint position 2 is oscillating after some period of time. The joint velocities are also exhibits the oscillatory behavior with some offsets.

5.1.1 Matlab Program

```
% ME 639: Introduction to robotics
% Midsem exam : Question 4 (a)
%               3 Oct 2018
%  a. For T 1 as computed in Q2(b), solve Eq. (1) using a numerical solver, say ode45, over a
%  span of time, [t i t f ] = [0 10] . Compare the computed joint positions and velocities with
%  the desired trajectory as planned in Q2 (a). Plot the results and discuss.
%
% Author: Shail Jadav 18310039
%% Initialization
clear
close all
clc
%% ODE solver computed

[t,x]=ode45('ode_solver_script_q4_a',[0,10],[0,0,0,0]);
m1=1; m2=1; l1=1; l2=1; g=9.81;

th1=x(:,1); dth1=x(:,2); th2=x(:,3); dth2=x(:,4);

%% Trejectory genration plannned
[~,q0,dq0,ddq0]=trecgen(0,1/10,10,0,0,pi/6,0);  %Trejectory generation for theta 1 0 to pi/6
[tq,q1,dq1,ddq1]=trecgen(0,1/10,10,0,0,pi/3,0);  %Trejectory generation for theta 2 0 to pi/3



%% Display The Results

figure('units','normalized','outerposition',[0 0 1 1])
subplot(223)
plot(t,th1,'r','LineWidth',1.5)
title('Joint Position computed $\theta_1$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Position (rad) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor


subplot(224)
plot(t,th2,'r','LineWidth',1.5)
title('Joint Position computed $\theta_2$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Position (rad) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor
set(gca)


subplot(221)
plot(tq,q0,'r','LineWidth',1.5)
title('Joint Position palnned $\theta_1$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Position (rad) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor


subplot(222)
plot(tq,q1,'r','LineWidth',1.5)
title('Joint Position palnned $\theta_2$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Position (rad) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
```

```
grid minor
set(gca)
saveas(gcf,'Q4_a_JP.png')

figure('units','normalized','outerposition',[0 0 1 1])

subplot(223)
plot(t,dth1,'b','LineWidth',1.5)
title('Joint Velocity computed $\dot{\theta_1}$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Velocity (rad/sec) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor


subplot(224)
plot(t,dth2,'b','LineWidth',1.5)
title('Joint Velocity computed $\dot{\theta_2}$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Velocity (rad/sec) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor
set(gca)




subplot(221)
plot(tq(1,1:end-1),dq0,'b','LineWidth',1.5)
title('Joint Velocity planned $\dot{\theta_1}$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Velocity (rad/sec) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor


subplot(222)
plot(tq(1,1:end-1),dq1,'b','LineWidth',1.5)
title('Joint Velocity planned $\dot{\theta_2}$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Velocity (rad/sec) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor
set(gca)

saveas(gcf,'Q4_a_JV.png')
```

Function of ODE

```
function Out= ode_solver_script_q4_a(t,x)

%% Input parameters
m2=1; m1=1; l1=1; l2=1; g=9.81;

tau1=(-3.474e-05)*t.^6 + (0.0008649 )*t.^5 + (-0.004596)*t.^4 + (-0.01598  )*t.^3 + ( 0.03615)*t
    .^2 + (-0.0571 )*t +19.76; %Input torque for planned trajectory
tau2=0; %Torque 2 is zero

%% Equation of motion
M11=((((m1/3) + m2)*l1^2) +((m2/3)*l2^2) + (m2*l1*l2*cos(x(3))));
M12=(m2*(((l2^2)/3) + (0.5*l1*l2*cos(x(3)))));
M21=M12;
M22=((1/3)*m2*l2*l2);


H1 =((-m2*l1*l2*sin(x(3))*x(2)*x(4)) - (0.5*m2*l1*l2*sin(x(3))*x(4)*x(4)));
H2 = (0.5 * m2* l1*l2*sin(x(3))*x(2)*x(2));

G1=( ((((0.5*m1) + m2)*l1*cos(x(1))) + (0.5*m2*l2*cos(x(1)+x(3))))*g);
G2=0.5*m2*l2*cos(x(1)+x(3))*g;
```

```
T=[tau1; tau2];

M=[M11 M12;M21 M22];

HG = [H1 + G1; H2 + G2];

%% Equation in terms of acceleration

ddth = (inv(M)) * (T - HG )     ;

OP=zeros(4,1);

%% Output
OP(1)=x(2); %Intergretion of velocity will give the position for theta 1
OP(2)=ddth(1);%Intergretion of acceleration will give the velocity for theta 1
OP(3)=x(4); %Intergretion of velocity will give the position  for theta 2
OP(4)=ddth(2); %Intergretion of acceleration will give the velocity for theta 2


Out=OP; % Output

end
```

5.2 Part B

Repeat above part (a) when a rotational spring, k r = 1 , is added at the joint 2 between link 1 and 2. Plot the results and discuss, how does the value of k r change the results.

Comparison of the computed joint positions and velocities with the desired trajectory as planned.

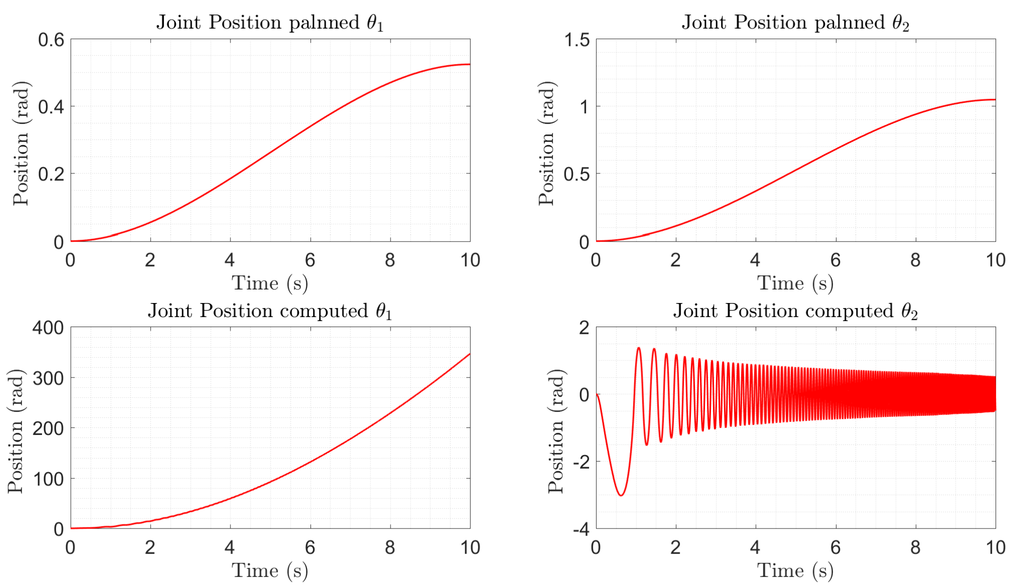

Figure 20: Comparison of the computed joint positions with the desired trajectory as planned

Initially τ_2 was zero now due to spring the torque due to spring $\tau_k = k * \theta_2$ will act on the joint 2 due to that the joint potion and velocities magnitude decrease significantly, we can observe the joint position and velocities in figures.With increasing the spring constant values the magnitude of position two will decrease as presented in animations.

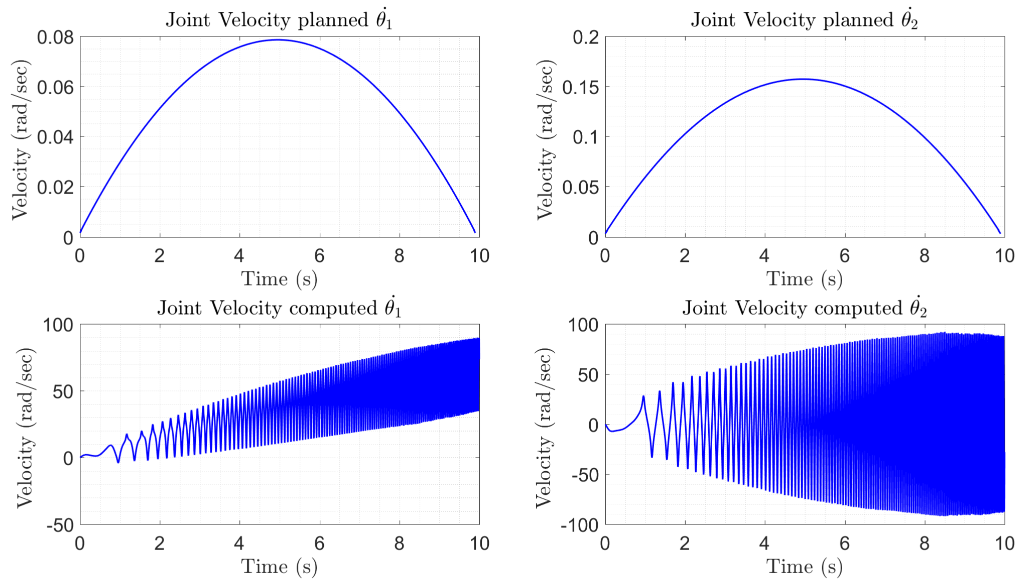

Figure 21: Comparison of the computed joint velocities with the desired trajectory as planned

5.2.1 Matlab Program

```
% ME 639: Introduction to robotics
% Midsem exam : Question 4 (b)
%                3 Oct 2018
%  Repeat above part (a) when a rotational spring , k r = 1 , is added at the joint 2 between
%  link 1 and 2. Plot the results and discuss , how does the value of k r change the results.
%
% Author: Shail Jadav 18310039
%% Initialization
clear
close all
clc
%% ODE solver computed

[t,x]=ode45('ode_solver_script_q4_b',[0,10],[0,0,0,0]);  % Time span 0 to 10 IC=[pi/4 0 pi/4 0]
    Theta1=pi/4 Theta2=pi/4

m1=1; m2=1; l1=1; l2=1; g=9.81;

th1=x(:,1); dth1=x(:,2); th2=x(:,3); dth2=x(:,4);

%% Trejectory genration plannned
[~,q0,dq0,ddq0]=trecgen(0,1/10,10,0,0,pi/6,0);  %Trejectory generation for theta 1 0 to pi/6
[tq,q1,dq1,ddq1]=trecgen(0,1/10,10,0,0,pi/3,0);  %Trejectory generation for theta 2 0 to pi/3



%% Display The Results

figure('units','normalized','outerposition',[0 0 1 1])
subplot(223)
plot(t,th1,'r','LineWidth',1.5)
title('Joint Position computed $\theta_1$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Position (rad) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor


subplot(224)
plot(t,th2,'r','LineWidth',1.5)
```

```
title('Joint Position computed $\theta_2$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Position (rad) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor
set(gca)


subplot(221)
plot(tq,q0,'r','LineWidth',1.5)
title('Joint Position palnned $\theta_1$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Position (rad) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor


subplot(222)
plot(tq,q1,'r','LineWidth',1.5)
title('Joint Position palnned $\theta_2$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Position (rad) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor
set(gca)
saveas(gcf,'Q4_b_JP.png')

figure('units','normalized','outerposition',[0 0 1 1])

subplot(223)
plot(t,dth1,'b','LineWidth',1.5)
title('Joint Velocity computed $\dot{\theta_1}$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Velocity (rad/sec) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor


subplot(224)
plot(t,dth2,'b','LineWidth',1.5)
title('Joint Velocity computed $\dot{\theta_2}$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Velocity (rad/sec) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor
set(gca)




subplot(221)
plot(tq(1,1:end-1),dq0,'b','LineWidth',1.5)
title('Joint Velocity planned $\dot{\theta_1}$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Velocity (rad/sec) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor


subplot(222)
plot(tq(1,1:end-1),dq1,'b','LineWidth',1.5)
title('Joint Velocity planned $\dot{\theta_2}$','Interpreter','latex')
xlabel('Time (s)','Interpreter','latex')
ylabel('Velocity (rad/sec) ','Interpreter','latex')
%ylim([-40 40])
set(gca,'FontSize',18)
grid minor
```

```
set(gca)

saveas(gcf,'Q4_b_JV.png')
```

Function of ODE

```
function Out= ode_solver_script_q4_b(t,x)

%% Input parameters
m2=1; m1=1; l1=1; l2=1; g=9.81; k=1; %k is spring constant

tau1=(-3.474e-05)*t.^6 + (0.0008649 )*t.^5 + (-0.004596)*t.^4 + (-0.01598  )*t.^3 + ( 0.03615)*t
    .^2 + (-0.0571 )*t +19.76; %Input torque is zero
tau2=0;


%% Equation of motion
M11=((((m1/3) + m2)*l1^2) +((m2/3)*l2^2) + (m2*l1*l2*cos(x(3))));
M12=(m2*(((l2^2)/3) + (0.5*l1*l2*cos(x(3)))));
M21=M12;
M22=((1/3)*m2*l2*l2);


H1 =((-m2*l1*l2*sin(x(3))*x(2)*x(4)) - (0.5*m2*l1*l2*sin(x(3))*x(4)*x(4)));
H2 = (0.5 * m2* l1*l2*sin(x(3))*x(2)*x(2));

G1=( ((((0.5*m1) + m2)*l1*cos(x(1))) + (0.5*m2*l2*cos(x(1)+x(3))))*g);
G2=0.5*m2*l2*cos(x(1)+x(3))*g;

T=[tau1; tau2];

M=[M11 M12;M21 M22];

HG = [H1 + G1; H2 + G2];

S=[0;k*x(3)]; %Torque due to spring

%% Equation in terms of acceleration

ddth = (inv(M)) * (T  - HG -S )     ;

OP=zeros(4,1);

%% Output
OP(1)=x(2); %Intergretion of velocity will give the position for theta 1
OP(2)=ddth(1);%Intergretion of acceleration will give the velocity for theta 1
OP(3)=x(4); %Intergretion of velocity will give the position  for theta 2
OP(4)=ddth(2); %Intergretion of acceleration will give the velocity for theta 2


Out=OP; % Output

end
```

5.3 Part C

Animate the 2R manipulator for part (a) and (b)

Part A

Please click on following links (Blue colour)

Animation of Q4 A Computed trajectory

5.3.1 Matlab Program

```
% ME 639: Introduction to robotics
% Midsem exam : Question 4 (a)
%              3 Oct 2018
%  a. For T 1 as computed in Q2(b), solve Eq. (1) using a numerical solver, say ode45, over a
%  span of time, [t i t f ] = [0 10] . Compare the computed joint positions and velocities with
%  the desired trajectory as planned in Q2 (a). Plot the results and discuss.
%
% Author: Shail Jadav 18310039
%% Initialization
```

```
clear
close all
clc
%% ODE solver computed

[t,x]=ode45('ode_solver_script_q4_a',[0,10],[0,0,0,0]);
m1=1; m2=1; l1=1; l2=1; g=9.81;

th1=x(:,1); dth1=x(:,2); th2=x(:,3); dth2=x(:,4);

%% Trejectory genration plannned
[~,q0,dq0,ddq0]=trecgen(0,1/10,10,0,0,pi/6,0); %Trejectory generation for theta 1 0 to pi/6
[tq,q1,dq1,ddq1]=trecgen(0,1/10,10,0,0,pi/3,0); %Trejectory generation for theta 2 0 to pi/3



%% Planned
figure('units','normalized','outerposition',[0 0 1 1])
c=1;
for i=1:10:length(q0)

theta1=q0(1,i);
theta2=q1(1,i);


l1=1; %Input the l length
l2=1; %Input the l length



% Homogeneus transformation matrix
H01 = [cos(theta1) -sin(theta1) 0 l1*cos(theta1);sin(theta1) cos(theta1) 0 l1*sin(theta1);0 0 1
    0;0 0 0 1]; %Frame 0 to 1 tranformation
H12 = [cos(theta2) -sin(theta2) 0 l2*cos(theta2);sin(theta2) cos(theta2) 0 l2*sin(theta2);0 0 1
    0;0 0 0 1]; %Frame 1 to 2 tranformation


H02=H01*H12;      %Frame 0 to 2 tranformation


O=[0,0];                    %Joint 1 position
P1=[H01(1,4) H01(2,4)];     %Joint 2 position
P2=[H02(1,4) H02(2,4)];     %Joint 3 position


Orn= atan2(H02(2,1),H02(1,1)); %Orientation of end effector
Orn=(Orn)*(180/pi);
%subplot(221)
plot(P1(1),P1(2),'ok','LineWidth',5)
hold on
plot(P2(1),P2(2),'om','LineWidth',5)
plot(0,0,'ok','LineWidth',10)
xlim([-2.5 2.5])
ylim([-2.5 2.5])
grid minor
plot([0 P1(1)], [0 P1(2)],'r','LineWidth',5)
plot([P1(1) P2(1)], [P1(2) P2(2)],'b','LineWidth',5)
hold off
title(strcat('Time = ',num2str(tq(1,i))),'Interpreter','latex')
xlabel('X axis (m)','Interpreter','latex')
ylabel('Y axis (m)','Interpreter','latex')
set(gca,'FontSize',18)
pause(0.0000000000000000001);
F(c) = getframe(gcf) ;
c=c+1;
end

% create the video writer with 30 fps
  writerObj = VideoWriter('Q4_planned_a.avi');
  writerObj.FrameRate = 30;
 % set the seconds per image
% open the video writer
open(writerObj);
```

```matlab
% write the frames to the video
for i=1:length(F)
    % convert the image to a frame
    frame = F(i) ;
    writeVideo(writerObj, frame);
end
% close the writer object
close(writerObj);


%% Computed
figure('units','normalized','outerposition',[0 0 1 1])
c=1;
for i=1:10:length(th1)

theta1=th1(i,1);
theta2=th2(i,1);


l1=1; %Input the l length
l2=1; %Input the l length



% Homogeneus transformation matrix
H01 = [cos(theta1) -sin(theta1) 0 l1*cos(theta1);sin(theta1) cos(theta1) 0 l1*sin(theta1);0 0 1
    0;0 0 0 1]; %Frame 0 to 1 tranformation
H12 = [cos(theta2) -sin(theta2) 0 l2*cos(theta2);sin(theta2) cos(theta2) 0 l2*sin(theta2);0 0 1
    0;0 0 0 1]; %Frame 1 to 2 tranformation


H02=H01*H12;      %Frame 0 to 2 tranformation


O=[0,0];                   %Joint 1 position
P1=[H01(1,4) H01(2,4)];    %Joint 2 position
P2=[H02(1,4) H02(2,4)];    %Joint 3 position


Orn= atan2(H02(2,1),H02(1,1));  %Orientation of end effector
Orn=(Orn)*(180/pi);
%subplot(221)
plot(P1(1),P1(2),'ok','LineWidth',5)
hold on
plot(P2(1),P2(2),'om','LineWidth',5)
plot(0,0,'ok','LineWidth',10)
xlim([-2.5 2.5])
ylim([-2.5 2.5])
grid minor
plot([0 P1(1)], [0 P1(2)],'r','LineWidth',5)
plot([P1(1) P2(1)], [P1(2) P2(2)],'b','LineWidth',5)
hold off
title(strcat('Time = ',num2str(t(i,1))),'Interpreter','latex')
xlabel('X axis (m)','Interpreter','latex')
ylabel('Y axis (m)','Interpreter','latex')
set(gca,'FontSize',18)
pause(0.0000000000000000001);
F(c) = getframe(gcf) ;
c=c+1;
end

% create the video writer with 30 fps
  writerObj = VideoWriter('Q4_computed_a.avi');
  writerObj.FrameRate = 30;
  % set the seconds per image
% open the video writer
open(writerObj);
% write the frames to the video
for i=1:length(F)
    % convert the image to a frame
    frame = F(i) ;
    writeVideo(writerObj, frame);
end
```

```
% close the writer object
close(writerObj);
```

Part B

Animation of Q4 B Computed trajectory with k=1

Animation of Q4 B Computed trajectory with k=10

Animation of Q4 B Computed trajectory with k=100

5.3.2 Matlab Program

```
% ME 639: Introduction to robotics
% Midsem exam : Question 4 (b)
%                3 Oct 2018
%  b. Repeat above part (a) when a rotational spring , k r = 1 , is added at the joint 2 between
%  link 1 and 2. Plot the results and discuss , how does the value of k r change the results.
% Author: Shail Jadav 18310039
%% Initialization
clear
close all
clc
%% ODE solver computed

[t,x]=ode45('ode_solver_script_q4_b',[0,10],[0,0,0,0]);  % Time span 0 to 10 IC=[pi/4 0 pi/4 0] Theta1=pi/4 Theta2=pi/4

m1=1; m2=1; l1=1; l2=1; g=9.81;

th1=x(:,1); dth1=x(:,2); th2=x(:,3); dth2=x(:,4);

%% Trejectory genration plannned
[~,q0,dq0,ddq0]=trecgen(0,1/10,10,0,0,pi/6,0);  %Trejectory generation for theta 1 0 to pi/6
[tq,q1,dq1,ddq1]=trecgen(0,1/10,10,0,0,pi/3,0);  %Trejectory generation for theta 2 0 to pi/3




%% Computed
figure('units','normalized','outerposition',[0 0 1 1])
c=1;
for i=1:3:length(th1)

theta1=th1(i,1);
theta2=th2(i,1);


l1=1; %Input the l length
l2=1; %Input the l length



% Homogeneus transformation matrix
H01 = [cos(theta1) -sin(theta1) 0 l1*cos(theta1);sin(theta1) cos(theta1) 0 l1*sin(theta1);0 0 1 0;0 0 0 1]; %Frame 0 to 1 tranformation
H12 = [cos(theta2) -sin(theta2) 0 l2*cos(theta2);sin(theta2) cos(theta2) 0 l2*sin(theta2);0 0 1 0;0 0 0 1]; %Frame 1 to 2 tranformation


H02=H01*H12;       %Frame 0 to 2 tranformation


O=[0,0];                       %Joint 1 position
P1=[H01(1,4) H01(2,4)];     %Joint 2 position
P2=[H02(1,4) H02(2,4)];     %Joint 3 position


Orn= atan2(H02(2,1),H02(1,1));  %Orientation of end effector
Orn=(Orn)*(180/pi);
%subplot(221)
plot(P1(1),P1(2),'ok','LineWidth',5)
hold on
plot(P2(1),P2(2),'om','LineWidth',5)
plot(0,0,'ok','LineWidth',10)
xlim([-2.5 2.5])
```

```
ylim([-2.5 2.5])
grid minor
plot([0 P1(1)], [0 P1(2)],'r','LineWidth',5)
plot([P1(1) P2(1)], [P1(2) P2(2)],'b','LineWidth',5)
hold off
title(strcat('Time = ',num2str(t(i,1))),'Interpreter','latex')
xlabel('X axis (m)','Interpreter','latex')
ylabel('Y axis (m)','Interpreter','latex')
set(gca,'FontSize',18)
pause(0.0000000000000000001);
F(c) = getframe(gcf) ;
c=c+1;
end

% create the video writer with 30 fps
  writerObj = VideoWriter('Q4_computed_b_k_1.avi');
  writerObj.FrameRate = 30;
  % set the seconds per image
% open the video writer
open(writerObj);
% write the frames to the video
for i=1:length(F)
    % convert the image to a frame
    frame = F(i) ;
    writeVideo(writerObj, frame);
end
% close the writer object
close(writerObj);
```

6 Project proposal and feedback

The project proposal and feedback are separate documents in main zip file.