

# 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,  $\tau_i$ ,  $i=1-2$ , denote the joint torque values.

The equation of motion given as:

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

where

$$M_{11} = \left(\frac{m_1}{3} + m_2\right)l_1^2 + \frac{m_2}{3}l_2^2 + m_2l_1l_2 \cos \theta_2$$

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

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

$$H_1 = -m_2l_1l_2 \sin \theta_2 \dot{\theta}_1 \dot{\theta}_2 - \frac{m_2}{2}l_1l_2 \sin \theta_2 \dot{\theta}_2^2$$

$$H_2 = \frac{m_2}{2}l_1l_2 \sin \theta_2 \dot{\theta}_1^2$$

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

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

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

## 2 Problem 1

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

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

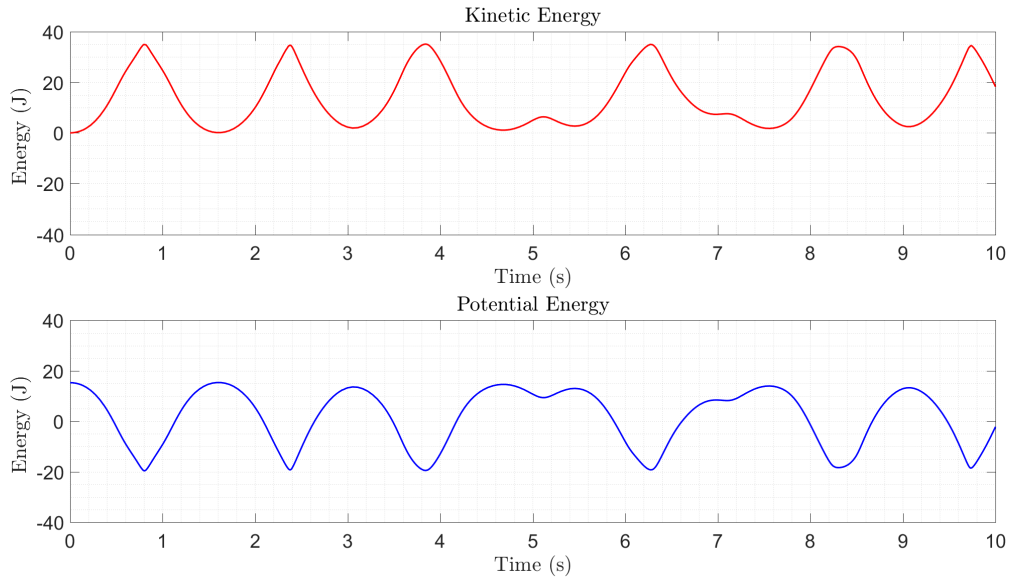


Figure 1: Kinetic energy and potential energy when both joint torques are zero

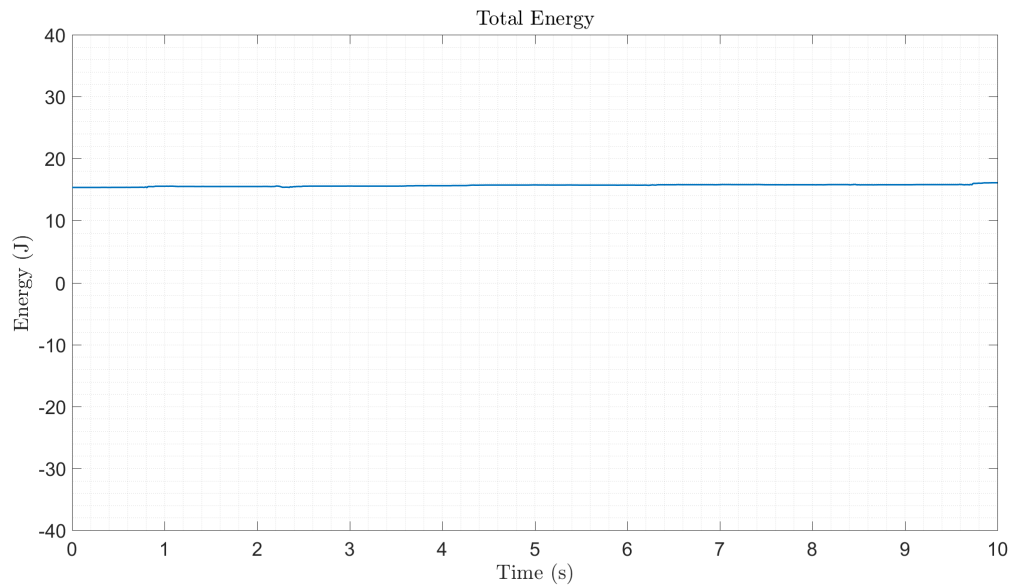


Figure 2: Total energy when both joint torques are zero

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 shown in figure 3.

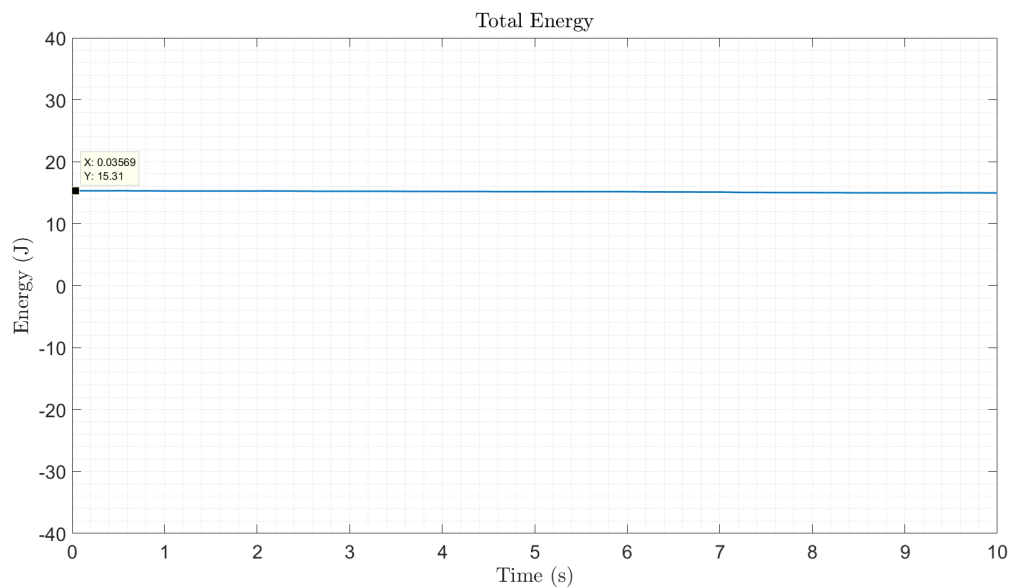


Figure 3: Total energy when both joint torques are zero computed with ODE23

### 2.1.1 Matlab Program

```
1 % ME 639: Introduction to robotics
2 % Midsem exam : Question 1 (a)
3 % 3 Oct 2018
4 %a. No joint torques are applied, i.e.,  $T_1 = T_2 = 0$ . Plot the kinetic, potential
5 % and total energy of the system with respect to time. Discuss the results.
6 %
7 % Author: Shail Jadav 18310039
8 %% Initialization
9 clear
10 close all
11 clc
12 %% ODE solver
13
14 [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
15 % 0] Theta1=pi/4 Theta2=pi/4
16
17 m1=1; m2=1; l1=1; l2=1; g=9.81;
18 th1=x(:,1); dth1=x(:,2); th2=x(:,3); dth2=x(:,4); %Joint position and velocities
19
20 %% Energy Calculation
21
22 for i=1:length(th1)
23
24 % Equation for kinetic energy
25 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 +
26 % 2*dth1(i)*dth2(i))) + (0.5*m2*l1*l2*cos(th2(i))*(dth1(i)^2 + dth1(i)*dth2(i)));
27
28 % Equation for potential energy
29 PE(i,1) = ((0.5*m1 + m2)*g*l1*sin(th1(i)))+(0.5*m2*g*l2*sin(th1(i)+th2(i)));
30
31 %Total energy
32 TE=KE+PE;
33
34 %% Display The Results
35
36 %Plotting energies
37 figure('units','normalized','outerposition',[0 0 1 1])
38 subplot(211)
39 plot(t,KE,'r','LineWidth',1.5)
40 title("Kinetic Energy",'Interpreter','latex')
41 xlabel('Time (s)','Interpreter','latex')
42 ylabel('Energy (J) ','Interpreter','latex')
43 ylim([-40 40])
44 set(gca,'FontSize',18)
45 grid minor
46
47
48 subplot(212)
49 plot(t,PE,'b','LineWidth',1.5)
50 title("Potential Energy",'Interpreter','latex')
51 xlabel('Time (s)','Interpreter','latex')
52 ylabel('Energy (J) ','Interpreter','latex')
53 ylim([-40 40])
54 set(gca,'FontSize',18)
55 grid minor
56 set(gca)
57 saveas(gcf,'Q1_a_KE_PE1.png')
58
59 figure('units','normalized','outerposition',[0 0 1 1])
60 plot(t,TE,'LineWidth',1.5)
61 title("Total Energy",'Interpreter','latex')
62 xlabel('Time (s)','Interpreter','latex')
63 ylabel('Energy (J) ','Interpreter','latex')
64 ylim([-40 40])
65 set(gca,'FontSize',18)
66 grid minor
67 saveas(gcf,'Q1_a_TE1.png')
```

ODE Solver script

```

1 function Out= ode_solver_script_q1_a(t,x)
2
3 %% Input parameters
4 m2=1; m1=1; l1=1; l2=1; g=9.81;
5
6 tau1= 0; %Input torque is zero
7 tau2=tau1; %second torque is the same as torque 1 which is zero
8
9
10 %% Equation of motion
11
12 %% M matrix
13 M11=((((m1/3) + m2)*l1^2) +((m2/3)*l2^2) + (m2*l1*l2*cos(x(3))));
14 M12=(m2*((l2^2)/3) + (0.5*l1*l2*cos(x(3))));
15 M21=M12;
16 M22=((1/3)*m2*l2*l2);
17
18 %% H matrix
19 H1 =((-m2*l1*l2*sin(x(3))*x(2)*x(4)) - (0.5*m2*l1*l2*sin(x(3))*x(4)*x(4)));
20 H2 = (0.5 * m2* l1*l2*sin(x(3))*x(2)*x(2));
21
22 %% G matrix
23 G1=( (((0.5*m1) + m2)*l1*cos(x(1))) + (0.5*m2*l2*cos(x(1)+x(3)))*g);
24 G2=0.5*m2*l2*cos(x(1)+x(3))*g;
25
26 T=[tau1; tau2];
27
28 M=[M11 M12;M21 M22];
29
30 HG = [H1 + G1; H2 + G2];
31
32 %% Equation in terms of acceleration
33
34 ddth = (inv(M)) * (T - HG) ; % Joint acceleration
35
36 OP=zeros(4,1);
37
38 %% Output
39 OP(1)=x(2); %Interpretation of velocity will give the position for theta 1
40 OP(2)=ddth(1); %Interpretation of acceleration will give the velocity for theta 1
41 OP(3)=x(4); %Interpretation of velocity will give the position for theta 2
42 OP(4)=ddth(2); %Interpretation of acceleration will give the velocity for theta 2
43
44
45 Out=OP; % Output
46
47 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.

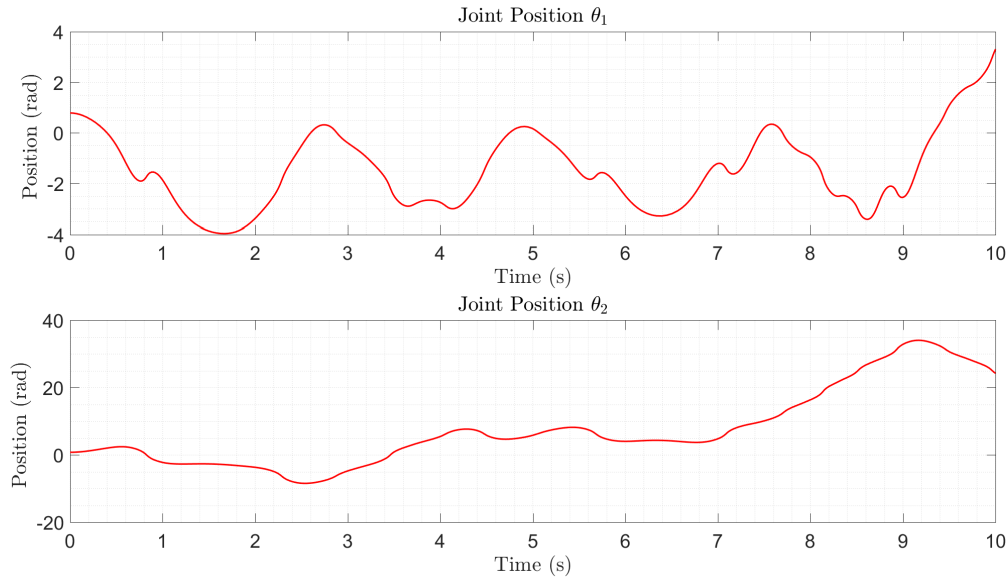


Figure 4: Joint positions when joint torques are sinusoidal

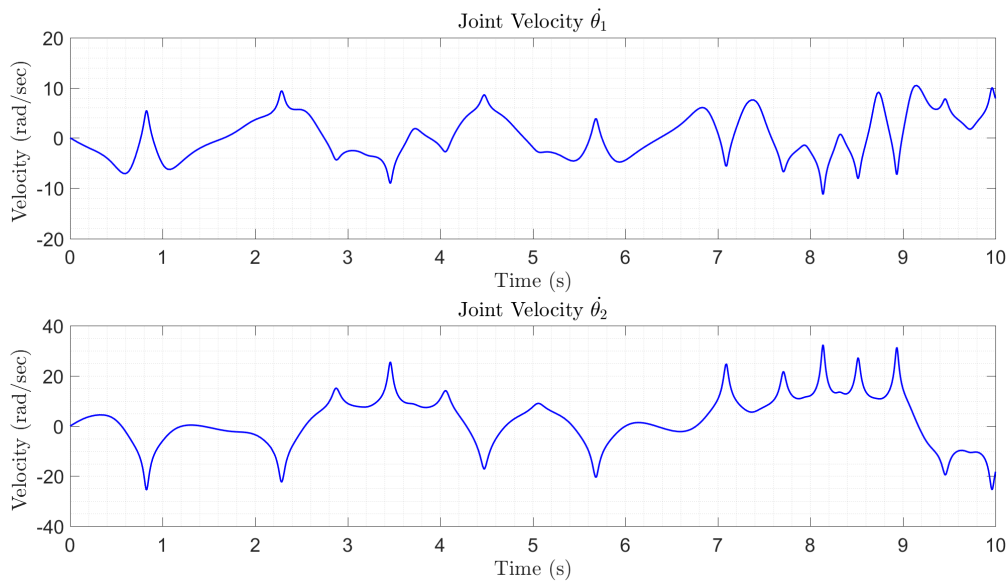


Figure 5: Joint velocities when joint torques are sinusoidal

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 where 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

```
1 % ME 639: Introduction to robotics
2 % Midsem exam : Question 1 (b)
3 % 3 Oct 2018
4 %b. Sinusoidal joint torques are applied, i.e.,  $T_1 = T_2 = \sin(t)$ . Plot joint positions and
   angular
5 % velocities with respect to time. Discuss the results.
6 %
7 % Author: Shail Jadav 18310039
8 %% Initialization
9 clear
10 close all
11 clc
12 %% ODE solver
13
14 [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
15
16 m1=1; m2=1; I1=1; I2=1; g=9.81;
17
18 th1=x(:,1); dth1=x(:,2); th2=x(:,3); dth2=x(:,4);
19
20 %% Display The Results
21
22 figure('units','normalized','outerposition',[0 0 1 1])
23 subplot(211)
24 plot(t,th1,'r','LineWidth',1.5)
25 title('Joint Position  $\theta_1$ ','Interpreter','latex')
26 xlabel('Time (s)','Interpreter','latex')
27 ylabel('Position (rad)','Interpreter','latex')
28 ylim([-40 40])
29 set(gca,'FontSize',18)
30 grid minor
31
32
33 subplot(212)
34 plot(t,th2,'r','LineWidth',1.5)
35 title('Joint Position  $\theta_2$ ','Interpreter','latex')
36 xlabel('Time (s)','Interpreter','latex')
37 ylabel('Position (rad)','Interpreter','latex')
38 ylim([-40 40])
39 set(gca,'FontSize',18)
40 grid minor
41 set(gca)
42 saveas(gcf,'Q1_b_JP.png')
43
44 figure('units','normalized','outerposition',[0 0 1 1])
45 subplot(211)
46 plot(t,dth1,'b','LineWidth',1.5)
47 title('Joint Velocity  $\dot{\theta}_1$ ','Interpreter','latex')
48 xlabel('Time (s)','Interpreter','latex')
49 ylabel('Velocity (rad/sec)','Interpreter','latex')
50 ylim([-40 40])
51 set(gca,'FontSize',18)
52 grid minor
53
54
55 subplot(212)
56 plot(t,dth2,'b','LineWidth',1.5)
57 title('Joint Velocity  $\dot{\theta}_2$ ','Interpreter','latex')
58 xlabel('Time (s)','Interpreter','latex')
59 ylabel('Velocity (rad/sec)','Interpreter','latex')
60 ylim([-40 40])
61 set(gca,'FontSize',18)
62 grid minor
63 set(gca)
64 saveas(gcf,'Q1_b_JV.png')
```

ODE Solver script

```
1 function Out= ode_solver_script_q1_b(t,x)
2
3 %% Input parameters
```

```

4 m2=1; m1=1; l1=1; l2=1; g=9.81;
5
6 tau1=sin(t); %Input torque is sin(t)
7 tau2=tau1; %second torque is the same as torque 1
8
9
10 %% Equation of motion
11 M11=((((m1/3) + m2)*l1^2) + ((m2/3)*l2^2) + (m2*l1*l2*cos(x(3))));
12 M12=(m2*((l2^2)/3) + (0.5*l1*l2*cos(x(3))));
13 M21=M12;
14 M22=((1/3)*m2*l2*l2);
15
16
17 H1 =((-m2*l1*l2*sin(x(3))*x(2)*x(4)) - (0.5*m2*l1*l2*sin(x(3))*x(4)*x(4)));
18 H2 = (0.5 * m2* l1*l2*sin(x(3))*x(2)*x(2));
19
20 G1=( (((0.5*m1) + m2)*l1*cos(x(1))) + (0.5*m2*l2*cos(x(1)+x(3)))*g);
21 G2=0.5*m2*l2*cos(x(1)+x(3))*g;
22
23 T=[tau1; tau2];
24
25 M=[M11 M12;M21 M22];
26
27 HG = [H1 + G1; H2 + G2];
28
29 %% Equation in terms of acceleration
30
31 ddth = (inv(M)) * (T - HG) ;
32
33 OP=zeros(4,1);
34
35 %% Output
36 OP(1)=x(2); %Interpretation of velocity will give the position for theta 1
37 OP(2)=ddth(1);%Interpretation of acceleration will give the velocity for theta 1
38 OP(3)=x(4); %Interpretation of velocity will give the position for theta 2
39 OP(4)=ddth(2); %Interpretation of acceleration will give the velocity for theta 2
40
41
42 Out=OP; % Output
43
44 end

```



### 2.3 C

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

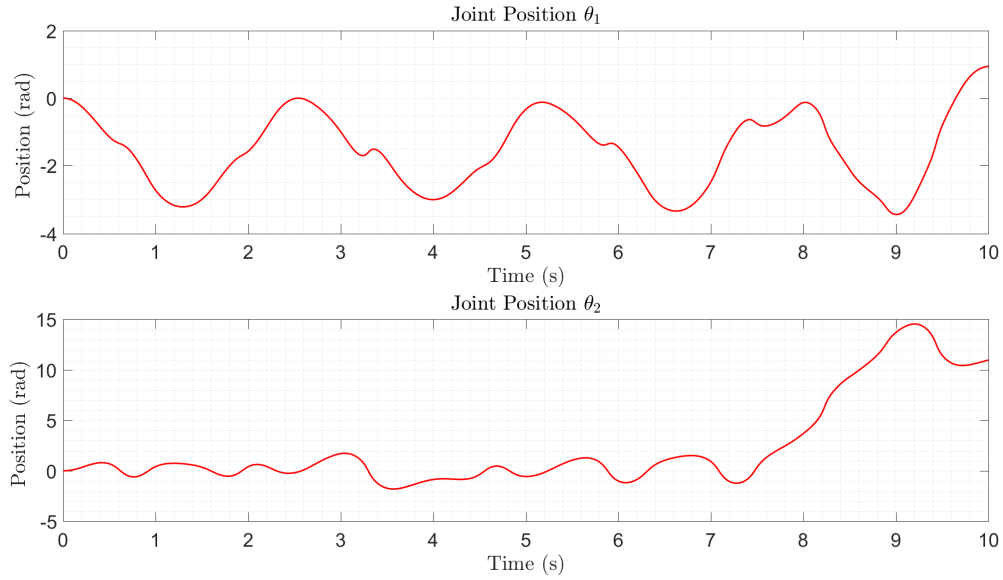


Figure 6: Joint positions when joint torques are sinusoidal

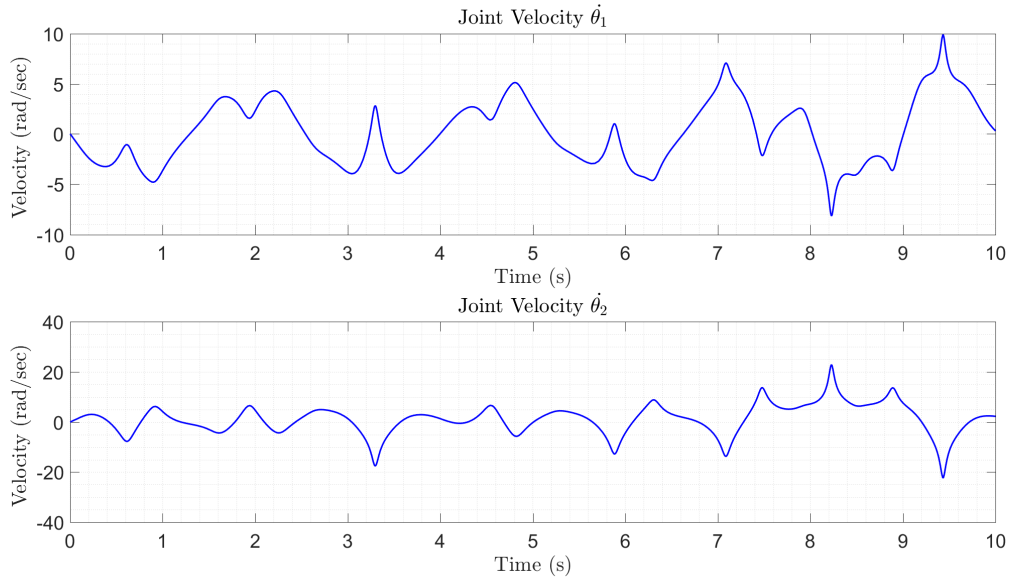


Figure 7: Joint velocities when joint torques are sinusoidal

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 where 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

```
1 % ME 639: Introduction to robotics
2 % Midsem exam : Question 1 (c)
3 % 3 Oct 2018
4 %c. From the results of part (b), explain how changing the IC to [0 0 0 0] would affect the
5 % solution.
6 %
7 % Author: Shail Jadav 18310039
8 %% Initialization
9 clear
10 close all
11 clc
12 %% ODE solver
13
14 [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
15 % Theta2=0
16 m1=1; m2=1; l1=1; l2=1; g=9.81;
17
18 th1=x(:,1); dth1=x(:,2); th2=x(:,3); dth2=x(:,4);
19
20
21
22 %% Display The Results
23
24 figure('units','normalized','outerposition',[0 0 1 1])
25 subplot(211)
26 plot(t,th1,'r','LineWidth',1.5)
27 title('Joint Position  $\theta_1$ ','Interpreter','latex')
28 xlabel('Time (s)','Interpreter','latex')
29 ylabel('Position (rad) ','Interpreter','latex')
30 ylim([-40 40])
31 set(gca,'FontSize',18)
32 grid minor
33
34
35 subplot(212)
36 plot(t,th2,'r','LineWidth',1.5)
37 title('Joint Position  $\theta_2$ ','Interpreter','latex')
38 xlabel('Time (s)','Interpreter','latex')
39 ylabel('Position (rad) ','Interpreter','latex')
40 ylim([-40 40])
41 set(gca,'FontSize',18)
42 grid minor
43 set(gca)
44 saveas(gcf,'Q1_c_JP.png')
45
46 figure('units','normalized','outerposition',[0 0 1 1])
47 subplot(211)
48 plot(t,dth1,'b','LineWidth',1.5)
49 title('Joint Velocity  $\dot{\theta}_1$ ','Interpreter','latex')
50 xlabel('Time (s)','Interpreter','latex')
51 ylabel('Velocity (rad/sec) ','Interpreter','latex')
52 ylim([-40 40])
53 set(gca,'FontSize',18)
54 grid minor
55
56
57 subplot(212)
58 plot(t,dth2,'b','LineWidth',1.5)
59 title('Joint Velocity  $\dot{\theta}_2$ ','Interpreter','latex')
60 xlabel('Time (s)','Interpreter','latex')
61 ylabel('Velocity (rad/sec) ','Interpreter','latex')
62 ylim([-40 40])
63 set(gca,'FontSize',18)
64 grid minor
65 set(gca)
66 saveas(gcf,'Q1_c_JV.png')
```

ODE Solver script

```
1 function Out= ode_solver_script_q1_c(t,x)
2
```

```

3 %% Input parameters
4 m2=1; m1=1; l1=1; l2=1; g=9.81;
5
6 tau1=sin(t); %Input torque is zero
7 tau2=tau1; %second torque is the same as torque 1
8
9
10 %% Equation of motion
11 M11=((((m1/3) + m2)*l1^2) + ((m2/3)*l2^2) + (m2*l1*l2*cos(x(3))));
12 M12=(m2*((l2^2)/3) + (0.5*l1*l2*cos(x(3))));
13 M21=M12;
14 M22=((1/3)*m2*l2*l2);
15
16
17 H1 =((-m2*l1*l2*sin(x(3))*x(2)*x(4)) - (0.5*m2*l1*l2*sin(x(3))*x(4)*x(4)));
18 H2 = (0.5 * m2* l1*l2*sin(x(3))*x(2)*x(2));
19
20 G1=( (((0.5*m1) + m2)*l1*cos(x(1))) + (0.5*m2*l2*cos(x(1)+x(3)))*g);
21 G2=0.5*m2*l2*cos(x(1)+x(3))*g;
22
23 T=[tau1; tau2];
24
25 M=[M11 M12;M21 M22];
26
27 HG = [H1 + G1; H2 + G2];
28
29 %% Equation in terms of acceleration
30
31 ddth = (inv(M)) * (T - HG) ;
32
33 OP=zeros(4,1);
34
35 %% Output
36 OP(1)=x(2); %Intergration of velocity will give the position for theta 1
37 OP(2)=ddth(1); %Intergration of acceleration will give the velocity for theta 1
38 OP(3)=x(4); %Intergration of velocity will give the position for theta 2
39 OP(4)=ddth(2); %Intergration of acceleration will give the velocity for theta 2
40
41
42 Out=OP; % Output
43
44 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

```

1 % ME 639: Introduction to robotics
2 % Midsem exam : Question 1 D
3 % 3 Oct 2018
4 % Animate the 2R manipulator for part (a) and (b).
5 %
6 % Author: Shail Jadav 18310039
7 %% Initialization
8 clear
9 close all
10 clc
11 %%
12 %syms theta1 theta2 theta3 l1 l2 l3
13
14 [t,x]=ode45('ode_solver_script_q1_a',[0,10],[pi/4,0,pi/4,0]);
15
16 m1=1; m2=1; l1=1; l2=1; g=9.81;
17
18 th1=x(:,1); dth1=x(:,2); th2=x(:,3); dth2=x(:,4);
19
20 %%
21

```



```

92 title("Kinetic Energy",'Interpreter','latex')
93 xlabel('Time (s)','Interpreter','latex')
94 ylabel('Energy (J) ','Interpreter','latex')
95 ylim([-40 40])
96 set(gca,'FontSize',18)
97 grid minor
98
99 subplot(224)
100 plot(t(i,1),PE(i,1),'ob','LineWidth',1)
101 hold on
102 title("Potential Energy",'Interpreter','latex')
103 xlabel('Time (s)','Interpreter','latex')
104 ylabel('Energy (J) ','Interpreter','latex')
105 ylim([-40 40])
106 set(gca,'FontSize',18)
107 grid minor
108 set(gca)
109 F(i) = getframe(gcf) ;
110 end
111
112 % create the video writer with 30 fps
113 writerObj = VideoWriter('animation_Q1_a.avi');
114 writerObj.FrameRate = 30;
115 % set the seconds per image
116 % open the video writer
117 open(writerObj);
118 % write the frames to the video
119 for i=1:length(F)
120     % convert the image to a frame
121     frame = F(i) ;
122     writeVideo(writerObj, frame);
123 end
124 % close the writer object
125 close(writerObj);

```

## Part B

### Animation of Q1 B

#### 2.4.2 Matlab Program

```

1 % ME 639: Introduction to robotics
2 % Midsem exam : Question 1 D
3 % 3 Oct 2018
4 % Animate the 2R manipulator for part (a) and (b).
5 %
6 % Author: Shail Jadav 18310039
7 %% Initialization
8 clear
9 close all
10 clc
11 %%
12 %syms theta1 theta2 theta3 l1 l2 l3
13
14 [t,x]=ode45('ode_solver_script_q1_b',[0,10],[pi/4,0,pi/4,0]);
15
16 m1=1; m2=1; l1=1; l2=1; g=9.81;
17
18 th1=x(:,1); dth1=x(:,2); th2=x(:,3); dth2=x(:,4);
19
20 %%
21
22 for i=1:length(th1)
23
24     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)));
25
26     PE(i,1) = ((0.5*m1 + m2)*g*l1*sin(th1(i))) + (0.5*m2*g*l2*sin(th1(i)+th2(i)));
27 end
28
29 TE=KE+PE;
30
31
32 figure('units','normalized','outerposition',[0 0 1 1])
33 for i=1:length(x)

```



```

105     set(gcf)
106     F(i) = getframe(gcf) ;
107 end
108
109 % create the video writer with 30 fps
110 writerObj = VideoWriter('animation_Q1_b.avi');
111 writerObj.FrameRate = 30;
112 % set the seconds per image
113 % open the video writer
114 open(writerObj);
115 % write the frames to the video
116 for i=1:length(F)
117     % convert the image to a frame
118     frame = F(i) ;
119     writeVideo(writerObj, frame);
120 end
121 % close the writer object
122 close(writerObj);

```

### 3 Problem 2

#### 3.1 Part A

Plan a trajectory when states of the system,  $[q_1 \ q_2 \ \dot{q}_1 \ \dot{q}_2]$ , change from  $[0 \ 0 \ 0 \ 0]$  to  $[\pi/6 \ \pi/3 \ 0 \ 0]$  in 10 sec . Plot the trajectory (  $q$ ,  $\dot{q}$  and  $\ddot{q}$  ) with respect to time.

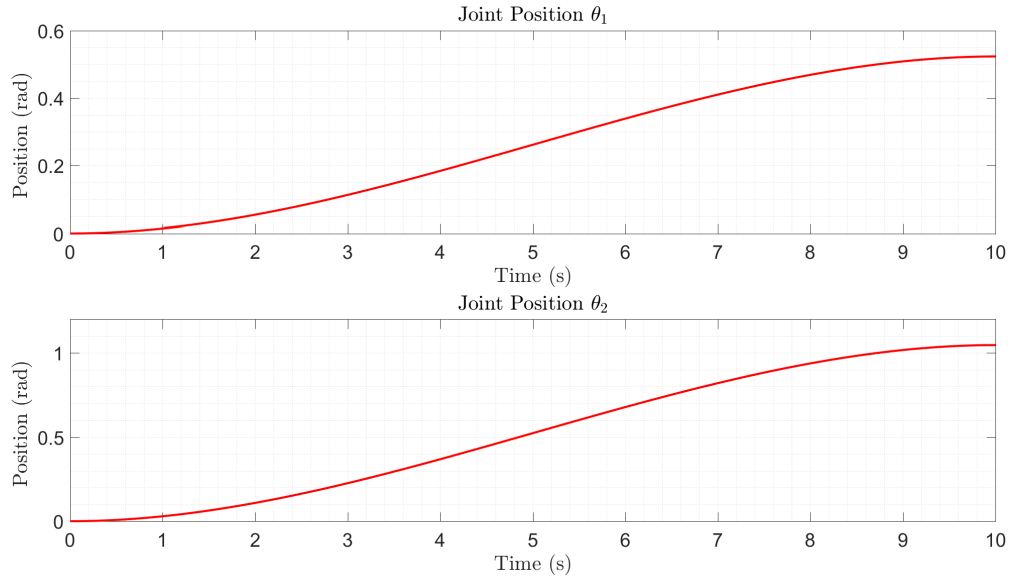


Figure 8: Joint positions for generated trajectory

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

$$a_0 = \theta_i$$

$$a_1 = \dot{\theta}_i$$

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

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

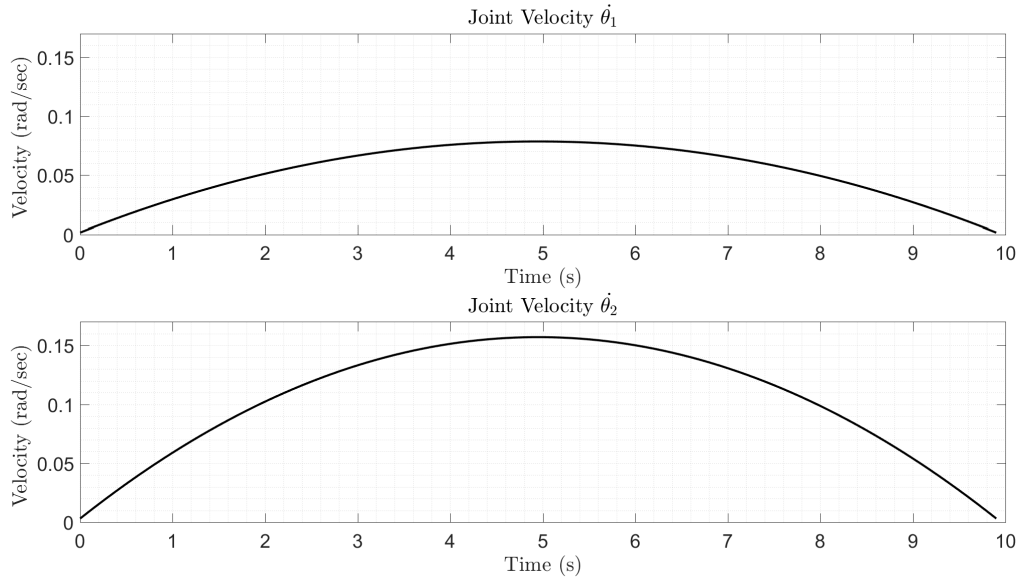


Figure 9: Joint velocities for generated trajectory

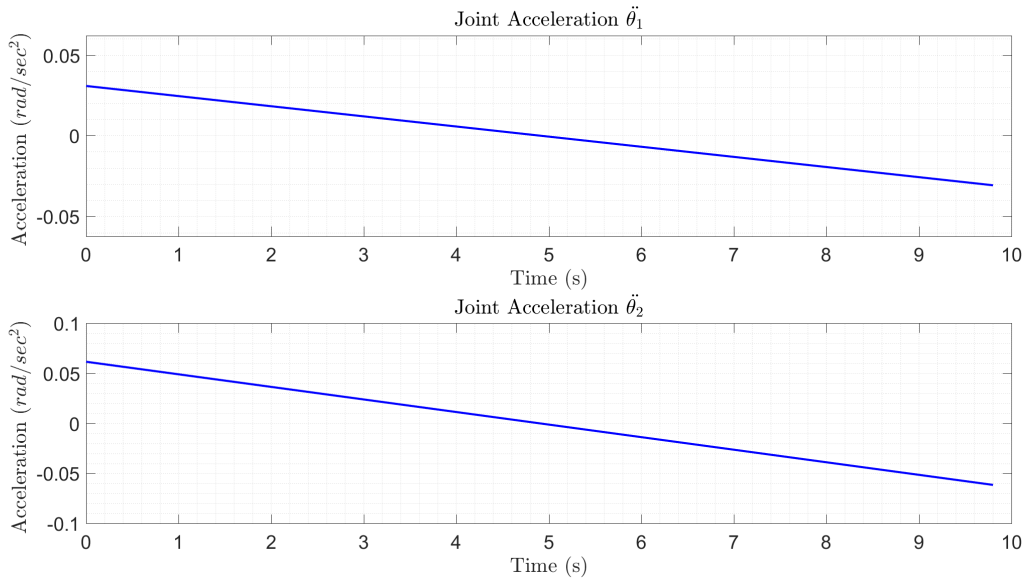


Figure 10: Joint acceleration for generated trajectory

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



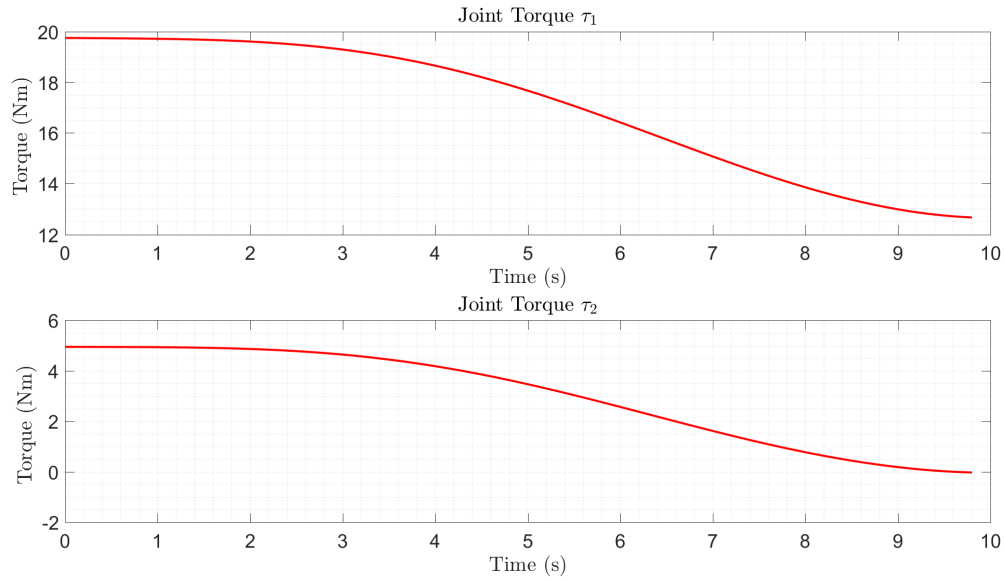


Figure 11: Computed torque for generated trajectory

### 3.1.2 Matlab Program

```

1 %% Initialization
2 clear
3 close all
4 clc
5 %% Trejectory genration
6 [~,q0,dq0,ddq0]=trecegen(0,1/10,10,0,0,pi/6,0); %Trejectory generation for theta 1 0 to pi/6
7 [t,q1,dq1,ddq1]=trecegen(0,1/10,10,0,0,pi/3,0); %Trejectory generation for theta 2 0 to pi/3
8
9
10 %% Input parameters
11 m2=1; m1=1; l1=1; l2=1; g=9.81;
12
13 tau1= 0; %Input torque is zero
14 tau2=tau1; %second torque is the same as torque 1
15
16
17
18 %% Equation of motion
19
20 for i=1:length(ddq0)
21 M11=((((m1/3) + m2)*l1^2) +((m2/3)*l2^2) + (m2*l1*l2*cos(q1(1,i))));
22 M12=(m2*((((l2^2)/3) + (0.5*l1*l2*cos(q1(1,i))))));
23 M21=M12;
24 M22=((1/3)*m2*l2*l2);
25
26
27 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)));
28 H2 = (0.5 * m2* l1*l2*sin(q1(1,i))*dq0(1,i)*dq0(1,i));
29
30 G1=( (((0.5*m1) + m2)*l1*cos(q0(1,i))) + (0.5*m2*l2*cos(q0(1,i)+q1(1,i))))*g);
31 G2=0.5*m2*l2*cos(q0(1,i)+q1(1,i))*g;
32
33 Tau1(1,i)= M11*ddq0(1,i) + M12*ddq1(1,i) + H1 +G1;
34 Tau2(1,i)= M21*ddq0(1,i) + M22*ddq1(1,i) + H2 +G2;
35 end
36
37 %% Display generated trejectories
38 figure('units','normalized','outerposition',[0 0 1 1])
39 subplot(211)
40 plot(t,q0,'r','LineWidth',2)
41 title('Joint Position $\theta_1$', 'Interpreter','latex')

```

```

42 xlabel('Time (s)', 'Interpreter', 'latex')
43 ylabel('Position (rad) ', 'Interpreter', 'latex')
44 set(gca, 'FontSize', 18)
45 grid minor
46
47
48 subplot(212)
49 plot(t, q1, 'r', 'LineWidth', 2)
50 title('Joint Position  $\dot{\theta}_2$ ', 'Interpreter', 'latex')
51 xlabel('Time (s)', 'Interpreter', 'latex')
52 ylabel('Position (rad) ', 'Interpreter', 'latex')
53 set(gca, 'FontSize', 18)
54 grid minor
55 ylim([0 1.2])
56 saveas(gcf, 'Q2_a_JP.png')
57
58 figure('units', 'normalized', 'outerposition', [0 0 1 1])
59 subplot(211)
60 plot(t(1, 1:end-1), dq0, 'k', 'LineWidth', 2)
61 title('Joint Velocity  $\dot{\theta}_1$ ', 'Interpreter', 'latex')
62 xlabel('Time (s)', 'Interpreter', 'latex')
63 ylabel('Velocity (rad/sec) ', 'Interpreter', 'latex')
64 set(gca, 'FontSize', 18)
65 ylim([0 0.17])
66 grid minor
67
68 subplot(212)
69 plot(t(1, 1:end-1), dq1, 'k', 'LineWidth', 2)
70 title('Joint Velocity  $\dot{\theta}_2$ ', 'Interpreter', 'latex')
71 xlabel('Time (s)', 'Interpreter', 'latex')
72 ylabel('Velocity (rad/sec) ', 'Interpreter', 'latex')
73 set(gca, 'FontSize', 18)
74 ylim([0 0.17])
75 grid minor
76 saveas(gcf, 'Q2_a_JV.png')
77
78 figure('units', 'normalized', 'outerposition', [0 0 1 1])
79 subplot(211)
80 plot(t(1, 1:end-2), ddq0, 'b', 'LineWidth', 2)
81 title('Joint Acceleration  $\ddot{\theta}_1$ ', 'Interpreter', 'latex')
82 xlabel('Time (s)', 'Interpreter', 'latex')
83 ylabel('Acceleration  $(\text{rad}/\text{sec}^2)$  ', 'Interpreter', 'latex')
84 set(gca, 'FontSize', 18)
85 ylim([-0.062 0.062])
86 grid minor
87
88 subplot(212)
89 plot(t(1, 1:end-2), ddq1, 'b', 'LineWidth', 2)
90 title('Joint Acceleration  $\ddot{\theta}_2$ ', 'Interpreter', 'latex')
91 xlabel('Time (s)', 'Interpreter', 'latex')
92 ylabel('Acceleration  $(\text{rad}/\text{sec}^2)$  ', 'Interpreter', 'latex')
93 set(gca, 'FontSize', 18)
94 grid minor
95 saveas(gcf, 'Q2_a_JA.png')
96
97
98 %% Display computed torque
99 figure('units', 'normalized', 'outerposition', [0 0 1 1])
100 subplot(211)
101 plot(t(1, 1:end-2), Tau1, 'r', 'LineWidth', 2)
102 title('Joint Torque  $\tau_1$ ', 'Interpreter', 'latex')
103 xlabel('Time (s)', 'Interpreter', 'latex')
104 ylabel('Torque (Nm) ', 'Interpreter', 'latex')
105 set(gca, 'FontSize', 18)
106 grid minor
107
108
109 subplot(212)
110 plot(t(1, 1:end-2), Tau2, 'r', 'LineWidth', 2)
111 title('Joint Torque  $\tau_2$ ', 'Interpreter', 'latex')
112 xlabel('Time (s)', 'Interpreter', 'latex')
113 ylabel('Torque (Nm) ', 'Interpreter', 'latex')
114 set(gca, 'FontSize', 18)

```



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

## 4 Problem 3

(Dynamic Simulation) Use the manipulator parameters as in Q1 and the initial condition (at  $t = 0$ ),  $IC = [q_1 \ q_2 \ \dot{q}_1 \ \dot{q}_2] = [0 \ 0 \ 0 \ 0]$

### 4.1 Part A

A. Solve Eq. (1) using a numerical solver, say ode45, for a span of time,  $[t_{\text{start}} \ t_{\text{end}}] = [0 \ 10]$  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.

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

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

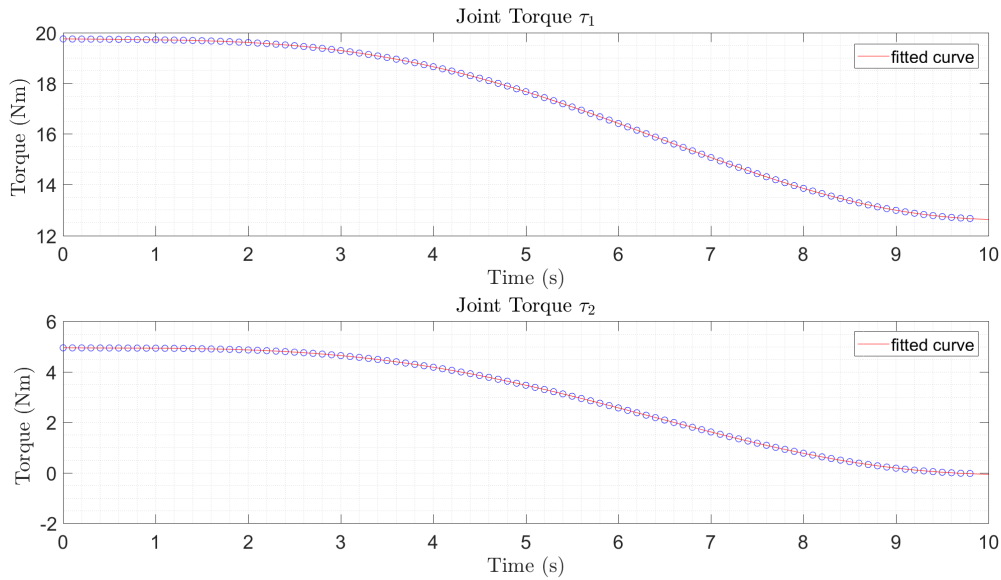


Figure 12: Curve fitting on planned torque

Matlab code for torque computation

```
1 %% Initialization
2 clear
3 close all
4 clc
5 %% Trejectory generation
6 [~,q0,dq0,ddq0]=trecegen(0,1/10,10,0,0,pi/6,0); %Trejectory generation for theta 1 0 to pi/6
7 [t,q1,dq1,ddq1]=trecegen(0,1/10,10,0,0,pi/3,0); %Trejectory generation for theta 2 0 to pi/3
8
9
10 x=[q0 dq0 q1 dq1];
11
12 %% Initialization of parameters
13 m2=1; m1=1; l1=1; l2=1; g=9.81;
14
15 %% Equation of motion
16
17 for i=1:length(ddq0)
18 M11=((((m1/3) + m2)*l1^2) + ((m2/3)*l2^2) + (m2*l1*l2*cos(q1(1,i))));
19 M12=(m2*((l2^2)/3) + (0.5*l1*l2*cos(q1(1,i))));
20 M21=M12;
21 M22=((1/3)*m2*l2*l2);
22
```

```

23
24 H1 =((-m2*m1*m2*sin(q1(1,i))*dq0(1,i)*dq1(1,i) - (0.5*m2*m1*m2*sin(q1(1,i))*dq1(1,i)*dq1(1,i)));
25 H2 = (0.5 * m2* m1*m2*sin(q1(1,i))*dq0(1,i)*dq0(1,i));
26
27 G1=( (((0.5*m1) + m2)*m1*cos(q0(1,i))) + (0.5*m2*m2*cos(q0(1,i)+q1(1,i))))*g);
28 G2=0.5*m2*m2*cos(q0(1,i)+q1(1,i))*g;
29
30 Tau1(1,i)= M11*ddq0(1,i) + M12*ddq1(1,i) + H1 +G1;
31 Tau2(1,i)= M21*ddq0(1,i) + M22*ddq1(1,i) + H2 +G2;
32 end
33
34 %% Curve fitting and equation
35 f1= fit(t(1,1:end-2)',Tau1','poly6');
36 cf1=coeffvalues(f1);
37 f2= fit(t(1,1:end-2)',Tau2','poly6');
38 cf2=coeffvalues(f2);
39
40 figure('units','normalized','outerposition',[0 0 1 1])
41 subplot(211)
42 plot(t(1,1:end-2),Tau1,'ob','LineWidth',1)
43 hold on
44 plot(f1)
45 title('Joint Torque $\tau_1$','Interpreter','latex')
46 xlabel('Time (s)','Interpreter','latex')
47 ylabel('Torque (Nm) ','Interpreter','latex')
48 set(gca,'FontSize',18)
49 grid minor
50
51
52 subplot(212)
53 plot(t(1,1:end-2),Tau2,'ob','LineWidth',1)
54 hold on
55 plot(f2)
56 title('Joint Torque $\tau_2$','Interpreter','latex')
57 xlabel('Time (s)','Interpreter','latex')
58 ylabel('Torque (Nm) ','Interpreter','latex')
59 set(gca,'FontSize',18)
60 grid minor
61 saveas(gcf,'Q3_a_CT.png')

```

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

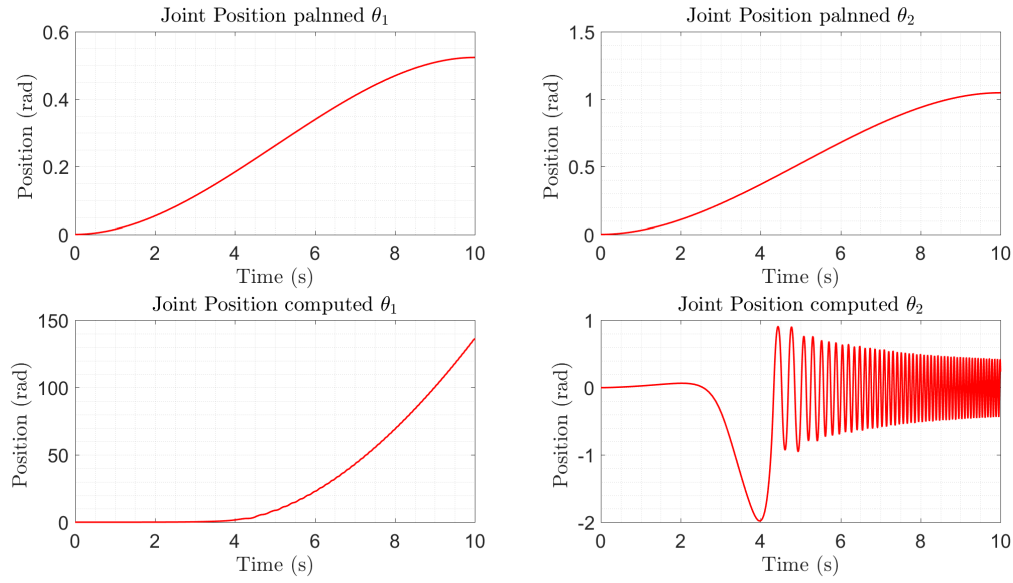


Figure 13: Comparison of the computed joint positions with the desired trajectory as planned

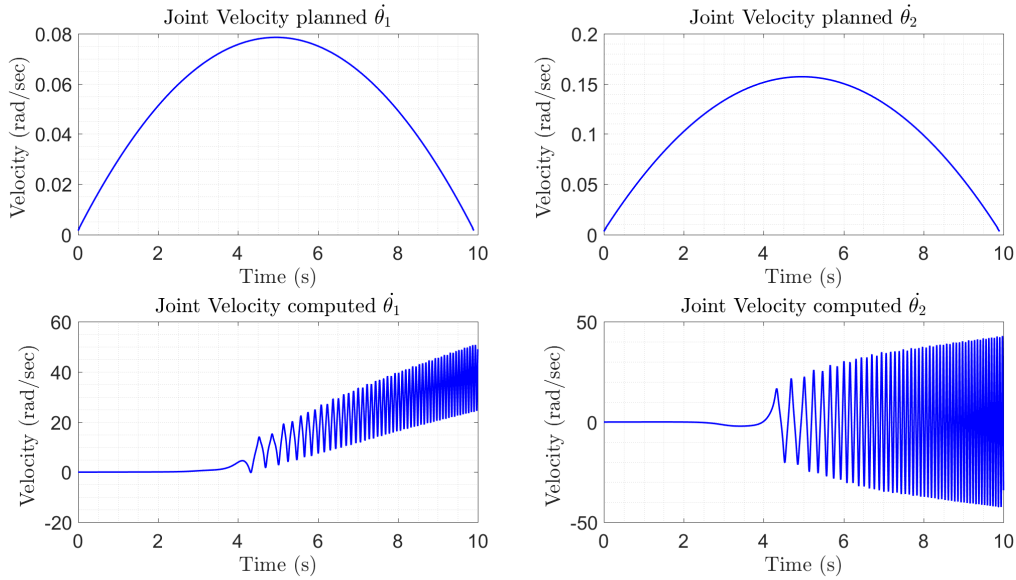


Figure 14: Comparison of the computed joint velocities with the desired trajectory as planned

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 position 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

```

1 % ME 639: Introduction to robotics
2 % Midsem exam : PID Control
3 % 18 Oct 2018
4 %
5 % Author: Shail Jadav 18310039
6 %% Initialization
7 clear
8 close all
9 clc
10 %%
11
12
13 [t,x]=ode45('ode_solver_script_pidcontrol',[0,10],[0,0,0,0]);
14
15 m1=1; m2=1; l1=1; l2=1; g=9.81;
16
17 th1=x(:,1); dth1=x(:,2); th2=x(:,3); dth2=x(:,4);
18
19
20
21 figure('units','normalized','outerposition',[0 0 1 1])
22
23 c=1;
24 for i=1:10:length(x)
25
26     theta1=x(i,1);
27     dtheta1=x(i,2);
28     theta2=x(i,3);
29     dtheta2=x(i,4);
30
31     l1=1; %Input the l length
32     l2=1; %Input the l length
33
34
35
36 % Homogeneous transformation matrix
37 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
38 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
39
40
41 H02=H01*H12; %Frame 0 to 2 tranformation
42
43
44 O=[0,0]; %Joint 1 position
45 P1=[H01(1,4) H01(2,4)]; %Joint 2 position
46 P2=[H02(1,4) H02(2,4)]; %Joint 3 position
47
48
49 Orn= atan2(H02(2,1),H02(1,1)); %Orientation of end effector
50 Orn=(Orn)*(180/pi);
51
52 plot(P1(1),P1(2),'ok','LineWidth',5)
53 hold on
54 plot(P2(1),P2(2),'om','LineWidth',5)
55 plot(0,0,'ok','LineWidth',10)
56 xlim([-2.5 2.5])
57 ylim([-2.5 2.5])
58 grid minor
59 plot([0 P1(1)], [0 P1(2)], 'r', 'LineWidth',5)
60 plot([P1(1) P2(1)], [P1(2) P2(2)], 'b', 'LineWidth',5)
61 hold off
62 title(strcat('Time = ',num2str(t(i,1))), 'Interpreter','latex')
63 xlabel('X axis (m)', 'Interpreter','latex')
64 ylabel('Y axis (m)', 'Interpreter','latex')

```



[illegible]

### The solver using PID controller

```

1 function Out= ode_solver_script_pidcontrol(t,x)
2
3 m2=1; m1=1; l1=1; l2=1; g=9.81;
4
5 th1f=pi/6;
6 th2f=pi/3;
7
8 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
9 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
10
11
12
13
14 M11=((((m1/3) + m2)*l1^2) +((m2/3)*l2^2) + (m2*l1*l2*cos(x(3))));
15 M12=(m2*(((l2^2)/3) + (0.5*l1*l2*cos(x(3)))));
16 M21=M12;
17 M22=((1/3)*m2*l2*l2);
18
19
20 H1 =((-m2*l1*l2*sin(x(3))*x(2)*x(4)) - (0.5*m2*l1*l2*sin(x(3))*x(4)*x(4)));
21 H2 = (0.5 * m2* l1*l2*sin(x(3))*x(2)*x(2));
22
23 G1=( (((0.5*m1) + m2)*l1*cos(x(1))) + (0.5*m2*l2*cos(x(1)+x(3)))*g);
24 G2=0.5*m2*l2*cos(x(1)+x(3))*g;
25
26 T=[tau1; tau2];
27
28 M=[M11 M12;M21 M22];
29
30 HG = [H1 + G1; H2 + G2];
31
32 S=[0 ; 0.001*x(3)];
33
34 %PID Controller
35 P=[853*(th1f - x(1)) - 100*x(2) + 0.5*x(1) ;553*(th2f - x(3)) - 100*x(4) + 0.5*x(3)];
36
37
38
39 ddth = (inv(M)) * (T - HG -S +P) ;
40
41 OP=zeros(4,1);
42
43 OP(1)=x(2);
44 OP(2)=ddth(1);
45 OP(3)=x(4);
46 OP(4)=ddth(2);

```

```
48  
49 Out=OP;  
50 end
```

## 4.1.2 Matlab Program

```
1 % ME 639: Introduction to robotics
2 % Midsem exam : Question 3 (a)
3 % 3 Oct 2018
4 % Solve Eq. (1) using a numerical solver, say ode45, for a span of time, [t i t f] = [0 10] and
5 % for joint torque values computed in Q2 (b). Compare the computed joint positions and
6 % velocities with the desired trajectory as planned in Q2 (a). Plot the results and discuss.
7 %
8 % Author: Shail Jadav 18310039
9 %% Initialization
10 clear
11 close all
12 clc
13 %% ODE solver computed
14
15 [t,x]=ode45('ode_solver_script_q3_a',[0,10],[0,0,0,0]);
16 m1=1; m2=1; l1=1; l2=1; g=9.81;
17
18 th1=x(:,1); dth1=x(:,2); th2=x(:,3); dth2=x(:,4);
19
20 %% Trejectory generation palnned
21 [~,q0,dq0,ddq0]=trecgen(0,1/10,10,0,0,pi/6,0); %Trejectory generation for theta 1 0 to pi/6
22 [tq,q1,dq1,ddq1]=trecgen(0,1/10,10,0,0,pi/3,0); %Trejectory generation for theta 2 0 to pi/3
23
24
25
26 %% Display The Results
27
28 figure('units','normalized','outerposition',[0 0 1 1])
29 subplot(223)
30 plot(t,th1,'r','LineWidth',1.5)
31 title('Joint Position computed  $\theta_1$ ','Interpreter','latex')
32 xlabel('Time (s)','Interpreter','latex')
33 ylabel('Position (rad)','Interpreter','latex')
34 ylim([-40 40])
35 set(gca,'FontSize',18)
36 grid minor
37
38
39 subplot(224)
40 plot(t,th2,'r','LineWidth',1.5)
41 title('Joint Position computed  $\theta_2$ ','Interpreter','latex')
42 xlabel('Time (s)','Interpreter','latex')
43 ylabel('Position (rad)','Interpreter','latex')
44 ylim([-40 40])
45 set(gca,'FontSize',18)
46 grid minor
47 set(gca)
48
49
50 subplot(221)
51 plot(tq,q0,'r','LineWidth',1.5)
52 title('Joint Position palnned  $\theta_1$ ','Interpreter','latex')
53 xlabel('Time (s)','Interpreter','latex')
54 ylabel('Position (rad)','Interpreter','latex')
55 ylim([-40 40])
56 set(gca,'FontSize',18)
57 grid minor
58
59
60 subplot(222)
61 plot(tq,q1,'r','LineWidth',1.5)
62 title('Joint Position palnned  $\theta_2$ ','Interpreter','latex')
63 xlabel('Time (s)','Interpreter','latex')
64 ylabel('Position (rad)','Interpreter','latex')
65 ylim([-40 40])
66 set(gca,'FontSize',18)
67 grid minor
68 set(gca)
69 saveas(gcf,'Q3_a_JP.png')
70
71 figure('units','normalized','outerposition',[0 0 1 1])
```

```

72
73 subplot(223)
74 plot(t,dth1,'b','LineWidth',1.5)
75 title('Joint Velocity computed $\dot{\theta}_1$', 'Interpreter','latex')
76 xlabel('Time (s)', 'Interpreter','latex')
77 ylabel('Velocity (rad/sec)', 'Interpreter','latex')
78 %ylim([-40 40])
79 set(gca,'FontSize',18)
80 grid minor
81
82
83 subplot(224)
84 plot(t,dth2,'b','LineWidth',1.5)
85 title('Joint Velocity computed $\dot{\theta}_2$', 'Interpreter','latex')
86 xlabel('Time (s)', 'Interpreter','latex')
87 ylabel('Velocity (rad/sec)', 'Interpreter','latex')
88 %ylim([-40 40])
89 set(gca,'FontSize',18)
90 grid minor
91 set(gca)
92
93
94
95
96 subplot(221)
97 plot(tq(1,1:end-1),dq0,'b','LineWidth',1.5)
98 title('Joint Velocity planned $\dot{\theta}_1$', 'Interpreter','latex')
99 xlabel('Time (s)', 'Interpreter','latex')
100 ylabel('Velocity (rad/sec)', 'Interpreter','latex')
101 %ylim([-40 40])
102 set(gca,'FontSize',18)
103 grid minor
104
105
106 subplot(222)
107 plot(tq(1,1:end-1),dq1,'b','LineWidth',1.5)
108 title('Joint Velocity planned $\dot{\theta}_2$', 'Interpreter','latex')
109 xlabel('Time (s)', 'Interpreter','latex')
110 ylabel('Velocity (rad/sec)', 'Interpreter','latex')
111 %ylim([-40 40])
112 set(gca,'FontSize',18)
113 grid minor
114 set(gca)
115
116 saveas(gcf,'Q3_a_JV.png')

```

#### Function of ODE

```

1 function Out= ode_solver_script_q3_a(t,x)
2
3 %% Input parameters
4 m2=1; m1=1; l1=1; l2=1; g=9.81;
5
6 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
7 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
8
9
10 %% Equation of motion
11 M11=((((m1/3) + m2)*l1^2) + ((m2/3)*l2^2) + (m2*l1*l2*cos(x(3))));
12 M12=(m2*((l2^2)/3) + (0.5*l1*l2*cos(x(3))));
13 M21=M12;
14 M22=((1/3)*m2*l2^2);
15
16
17 H1 =((-m2*l1*l2*sin(x(3)))*x(2)*x(4)) - (0.5*m2*l1*l2*sin(x(3)))*x(4)*x(4));
18 H2 = (0.5 * m2* l1*l2*sin(x(3)))*x(2)*x(2));
19
20 G1=( (((0.5*m1) + m2)*l1*cos(x(1))) + (0.5*m2*l2*cos(x(1)+x(3))))*g);
21 G2=0.5*m2*l2*cos(x(1)+x(3))*g;
22
23 T=[tau1; tau2];
24

```

```

25 M=[M11 M12;M21 M22];
26
27 HG = [H1 + G1; H2 + G2];
28
29 %% Equation in terms of acceleration
30
31 ddth = (inv(M)) * (T - HG) ;
32
33 OP=zeros(4,1);
34
35 %% Output
36 OP(1)=x(2); %Intergration of velocity will give the position for theta 1
37 OP(2)=ddth(1);%Intergration of acceleration will give the velocity for theta 1
38 OP(3)=x(4); %Intergration of velocity will give the position for theta 2
39 OP(4)=ddth(2); %Intergration of acceleration will give the velocity for theta 2
40
41
42 Out=OP; % Output
43
44 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.

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

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

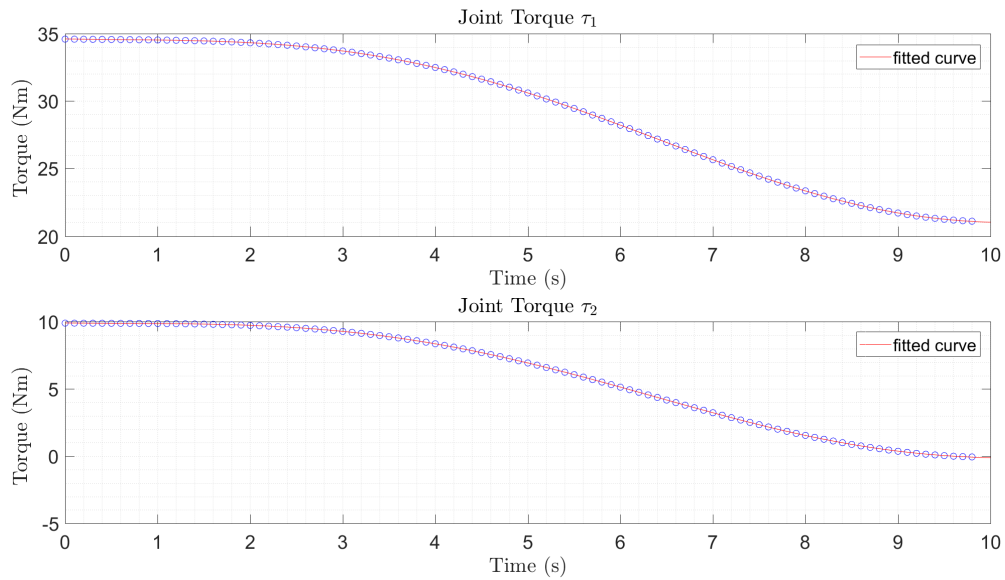


Figure 15: Curve fitting on planned torque

Matlab code

```

1 %% Initialization
2 clear
3 close all
4 clc
5 %% Trajectory generation
6 [~,q0,dq0,ddq0]=trecegen(0,1/10,10,0,0,pi/6,0); %Trajectory generation for theta 1 0 to pi/6
7 [t,q1,dq1,ddq1]=trecegen(0,1/10,10,0,0,pi/3,0); %Trajectory generation for theta 2 0 to pi/3
8
9
10 x=[q0 dq0 q1 dq1];

```

```

11
12 %% Initialization of parameters
13 m2=2; m1=1; l1=1; l2=1; g=9.81;
14
15
16 %% Equation of motion
17
18 for i=1:length(ddq0)
19 M11=(((m1/3) + m2)*l1^2) + ((m2/3)*l2^2) + (m2*l1*l2*cos(q1(1,i)));
20 M12=(m2*(((l2^2)/3) + (0.5*l1*l2*cos(q1(1,i)))));
21 M21=M12;
22 M22=(((1/3)*m2*l2^2);
23
24
25 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)));
26 H2 = (0.5 * m2* l1*l2*sin(q1(1,i))*dq0(1,i)*dq0(1,i));
27
28 G1=( (((0.5*m1) + m2)*l1*cos(q0(1,i))) + (0.5*m2*l2*cos(q0(1,i)+q1(1,i))))*g);
29 G2=0.5*m2*l2*cos(q0(1,i)+q1(1,i))*g;
30
31 Tau1(1,i)= M11*ddq0(1,i) + M12*ddq1(1,i) + H1 +G1;
32 Tau2(1,i)= M21*ddq0(1,i) + M22*ddq1(1,i) + H2 +G2;
33 end
34
35
36 %% Curve fitting and equation
37 f1= fit(t(1,1:end-2)',Tau1','poly6');
38 cf1=coeffvalues(f1);
39 f2= fit(t(1,1:end-2)',Tau2','poly6');
40 cf2=coeffvalues(f2);
41
42 figure('units','normalized','outerposition',[0 0 1 1])
43 subplot(211)
44 plot(t(1,1:end-2),Tau1,'ob','LineWidth',1)
45 hold on
46 plot(f1)
47 title('Joint Torque $\tau_1$', 'Interpreter','latex')
48 xlabel('Time (s)', 'Interpreter','latex')
49 ylabel('Torque (Nm) ', 'Interpreter','latex')
50 set(gca,'FontSize',18)
51 grid minor
52
53
54 subplot(212)
55 plot(t(1,1:end-2),Tau2,'ob','LineWidth',1)
56 hold on
57 plot(f2)
58 title('Joint Torque $\tau_2$', 'Interpreter','latex')
59 xlabel('Time (s)', 'Interpreter','latex')
60 ylabel('Torque (Nm) ', 'Interpreter','latex')
61 set(gca,'FontSize',18)
62 grid minor
63 saveas(gcf,'Q3_b_CT.png')

```

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

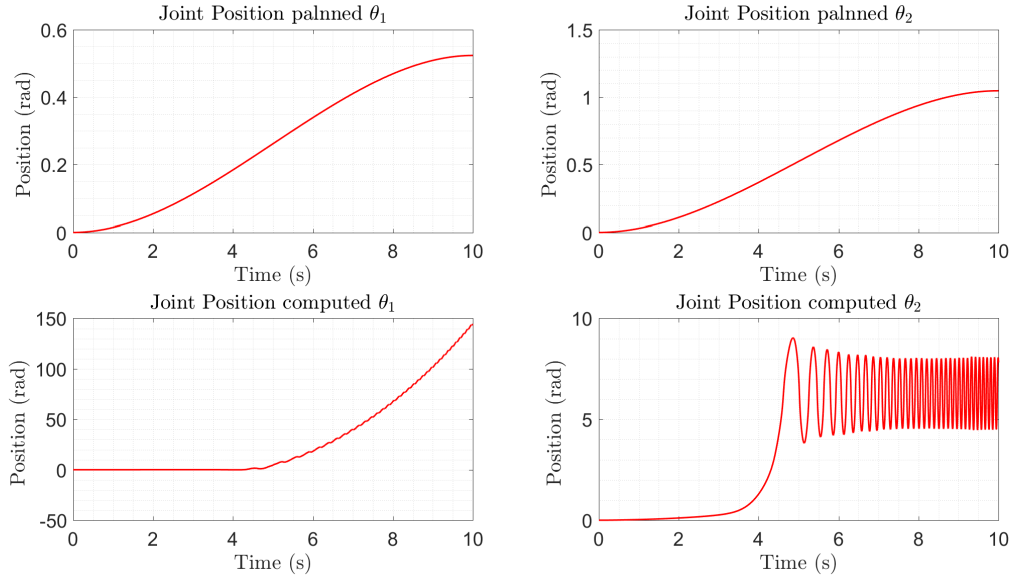


Figure 16: Comparison of the computed joint positions with the desired trajectory as planned

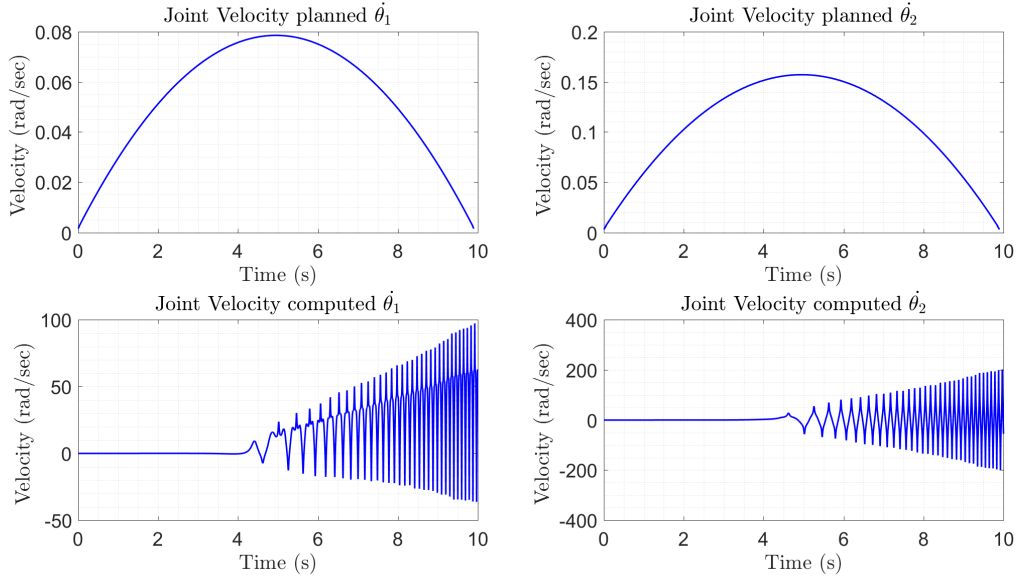


Figure 17: Comparison of the computed joint velocities with the desired trajectory as planned

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

```

1 % ME 639: Introduction to robotics
2 % Midsem exam : Question 3 (b)
3 % 3 Oct 2018
4 %Repeat above part (a) when a point mass, m e = 1 , is added at the center of mass of the
5 %link 2. Plot the results and discuss.
6 %
7 % Author: Shail Jadav 18310039
8 %% Initialization
9 clear
10 close all
11 clc
12 %% ODE solver computed
13
14 [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]
15 % Theta1=pi/4 Theta2=pi/4
16 m1=1; m2=2; l1=1; l2=1; g=9.81;
17
18 th1=x(:,1); dth1=x(:,2); th2=x(:,3); dth2=x(:,4);
19
20 %% Trejectory genration plannned
21 [~,q0,dq0,ddq0]=trecgen(0,1/10,10,0,0,pi/6,0); %Trejectory generation for theta 1 0 to pi/6
22 [tq,q1,dq1,ddq1]=trecgen(0,1/10,10,0,0,pi/3,0); %Trejectory generation for theta 2 0 to pi/3
23
24
25
26 %% Display The Results
27
28 figure('units','normalized','outerposition',[0 0 1 1])
29 subplot(223)
30 plot(t,th1,'r','LineWidth',1.5)
31 title('Joint Position computed $\theta_1$', 'Interpreter','latex')
32 xlabel('Time (s)', 'Interpreter','latex')
33 ylabel('Position (rad) ', 'Interpreter','latex')
34 ylim([-40 40])
35 set(gca,'FontSize',18)
36 grid minor
37
38
39 subplot(224)
40 plot(t,th2,'r','LineWidth',1.5)
41 title('Joint Position computed $\theta_2$', 'Interpreter','latex')
42 xlabel('Time (s)', 'Interpreter','latex')
43 ylabel('Position (rad) ', 'Interpreter','latex')
44 ylim([-40 40])
45 set(gca,'FontSize',18)
46 grid minor
47 set(gca)
48
49
50 subplot(221)
51 plot(tq,q0,'r','LineWidth',1.5)
52 title('Joint Position palnned $\theta_1$', 'Interpreter','latex')
53 xlabel('Time (s)', 'Interpreter','latex')
54 ylabel('Position (rad) ', 'Interpreter','latex')
55 ylim([-40 40])
56 set(gca,'FontSize',18)
57 grid minor
58
59
60 subplot(222)
61 plot(tq,q1,'r','LineWidth',1.5)
62 title('Joint Position palnned $\theta_2$', 'Interpreter','latex')
63 xlabel('Time (s)', 'Interpreter','latex')
64 ylabel('Position (rad) ', 'Interpreter','latex')
65 ylim([-40 40])
66 set(gca,'FontSize',18)
67 grid minor
68 set(gca)
69 saveas(gcf,'Q3_b_JP.png')
70

```



```

71 figure('units','normalized','outerposition',[0 0 1 1])
72
73 subplot(223)
74 plot(t,dth1,'b','LineWidth',1.5)
75 title('Joint Velocity computed $\dot{\theta}_1$', 'Interpreter','latex')
76 xlabel('Time (s)','Interpreter','latex')
77 ylabel('Velocity (rad/sec) ', 'Interpreter','latex')
78 %ylim([-40 40])
79 set(gca,'FontSize',18)
80 grid minor
81
82
83 subplot(224)
84 plot(t,dth2,'b','LineWidth',1.5)
85 title('Joint Velocity computed $\dot{\theta}_2$', 'Interpreter','latex')
86 xlabel('Time (s)','Interpreter','latex')
87 ylabel('Velocity (rad/sec) ', 'Interpreter','latex')
88 %ylim([-40 40])
89 set(gca,'FontSize',18)
90 grid minor
91 set(gca)
92
93
94
95
96 subplot(221)
97 plot(tq(1,1:end-1),dq0,'b','LineWidth',1.5)
98 title('Joint Velocity planned $\dot{\theta}_1$', 'Interpreter','latex')
99 xlabel('Time (s)','Interpreter','latex')
100 ylabel('Velocity (rad/sec) ', 'Interpreter','latex')
101 %ylim([-40 40])
102 set(gca,'FontSize',18)
103 grid minor
104
105
106 subplot(222)
107 plot(tq(1,1:end-1),dq1,'b','LineWidth',1.5)
108 title('Joint Velocity planned $\dot{\theta}_2$', 'Interpreter','latex')
109 xlabel('Time (s)','Interpreter','latex')
110 ylabel('Velocity (rad/sec) ', 'Interpreter','latex')
111 %ylim([-40 40])
112 set(gca,'FontSize',18)
113 grid minor
114 set(gca)
115
116 saveas(gcf,'Q3_b_JV.png')

```

### Function of ODE

```

1 function Out= ode_solver_script_q3_b(t,x)
2
3 %% Input parameters
4 m2=2; m1=1; l1=1; l2=1; g=9.81;
5
6 tau1=(-6.672e-05)*t.^6 + (0.001649 )*t.^5 + (-0.00861 )*t.^4 + ( -0.0318 )*t.^3 + ( 0.07193)*t.^2
7 + (-0.1118 )*t + 34.6 ; %Input torque is zero
8 tau2=(-5.353e-05 )*t.^6 + (0.001283 )*t.^5 + (-0.006265 )*t.^4 + (-0.02784 )*t.^3 + ( 0.06457 )
9 *t.^2 + ( -0.07288 )*t +9.912 ; %second torque is the same as torque 1
10
11 %% Equation of motion
12 M11=(((m1/3) + m2)*l1^2) + ((m2/3)*l2^2) + (m2*l1*l2*cos(x(3)));
13 M12=(m2*(((l2^2)/3) + (0.5*l1*l2*cos(x(3))));
14 M21=M12;
15 M22=((1/3)*m2*l2*l2);
16
17 H1 =((-m2*l1*l2*sin(x(3))*x(2)*x(4)) - (0.5*m2*l1*l2*sin(x(3))*x(4)*x(4)));
18 H2 = (0.5 * m2* l1*l2*sin(x(3))*x(2)*x(2));
19
20 G1= ( (((0.5*m1) + m2)*l1*cos(x(1))) + (0.5*m2*l2*cos(x(1)+x(3))))*g);
21 G2=0.5*m2*l2*cos(x(1)+x(3))*g;
22
23 T=[tau1; tau2];

```

```

24 M=[M11 M12;M21 M22];
25
26
27 HG = [H1 + G1; H2 + G2];
28
29 %% Equation in terms of acceleration
30
31 ddth = (inv(M)) * (T - HG) ;
32
33 OP=zeros(4,1);
34
35 %% Output
36 OP(1)=x(2); %Intergration of velocity will give the position for theta 1
37 OP(2)=ddth(1);%Intergration of acceleration will give the velocity for theta 1
38 OP(3)=x(4); %Intergration of velocity will give the position for theta 2
39 OP(4)=ddth(2); %Intergration of acceleration will give the velocity for theta 2
40
41
42 Out=OP; % Output
43
44 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

```

1 % ME 639: Introduction to robotics
2 % Midsem exam : Question 3 (a) Animation
3 % 3 Oct 2018
4 % Solve Eq. (1) using a numerical solver, say ode45, for a span of time, [t i t f] = [0 10] and
5 % for joint torque values computed in Q2 (b). Compare the computed joint positions and
6 % velocities with the desired trajectory as planned in Q2 (a). Plot the results and discuss.
7 %
8 % Author: Shail Jadav 18310039
9 %% Initialization
10 clear
11 close all
12 clc
13 %% ODE solver computed
14
15 [t,x]=ode45('ode_solver_script_q3_a',[0,10],[0,0,0,0]);
16 m1=1; m2=1; l1=1; l2=1; g=9.81;
17
18 th1=x(:,1); dth1=x(:,2); th2=x(:,3); dth2=x(:,4);
19
20 %% Trejectory genration plannned
21 [~,q0,dq0,ddq0]=trecegen(0,1/10,10,0,0,pi/6,0); %Trejectory generation for theta 1 0 to pi/6
22 [tq,q1,dq1,ddq1]=trecegen(0,1/10,10,0,0,pi/3,0); %Trejectory generation for theta 2 0 to pi/3
23
24
25 %% Planned
26 figure('units','normalized','outerposition',[0 0 1 1])
27 for i=1:length(q0)
28
29 theta1=q0(1,i);
30 theta2=q1(1,i);
31
32
33 l1=1; %Input the l length
34 l2=1; %Input the l length
35
36
37
38 % Homogeneous transformation matrix
39 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

```

```

40 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
41
42
43 H02=H01*H12;          %Frame 0 to 2 tranformation
44
45
46 O=[0,0];              %Joint 1 position
47 P1=[H01(1,4) H01(2,4)]; %Joint 2 position
48 P2=[H02(1,4) H02(2,4)]; %Joint 3 position
49
50
51 Orn= atan2(H02(2,1),H02(1,1)); %Orientation of end effector
52 Orn=(Orn)*(180/pi);
53 %subplot(221)
54 plot(P1(1),P1(2),'ok','LineWidth',5)
55 hold on
56 plot(P2(1),P2(2),'om','LineWidth',5)
57 plot(0,0,'ok','LineWidth',10)
58 xlim([-2.5 2.5])
59 ylim([-2.5 2.5])
60 grid minor
61 plot([0 P1(1)], [0 P1(2)], 'r','LineWidth',5)
62 plot([P1(1) P2(1)], [P1(2) P2(2)], 'b','LineWidth',5)
63 hold off
64 title(strcat('Time = ',num2str(tq(1,i))), 'Interpreter','latex')
65 xlabel('X axis (m)','Interpreter','latex')
66 ylabel('Y axis (m)','Interpreter','latex')
67 set(gca,'FontSize',18)
68 pause(0.00000000000000000001);
69 F(i) = getframe(gcf) ;
70 end
71
72 % create the video writer with 30 fps
73 writerObj = VideoWriter('Q3_planned_a.avi');
74 writerObj.FrameRate = 30;
75 % set the seconds per image
76 % open the video writer
77 open(writerObj);
78 % write the frames to the video
79 for i=1:length(F)
80     % convert the image to a frame
81     frame = F(i) ;
82     writeVideo(writerObj, frame);
83 end
84 % close the writer object
85 close(writerObj);
86
87
88 %% Computed
89 figure('units','normalized','outerposition',[0 0 1 1])
90 c=1;
91 for i=1:2:length(th1)
92
93     theta1=th1(i,1);
94     theta2=th2(i,1);
95
96
97     l1=1; %Input the l length
98     l2=1; %Input the l length
99
100
101
102 % Homogeneous transformation matrix
103 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
104 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
105
106
107 H02=H01*H12;          %Frame 0 to 2 tranformation
108
109

```

```

110 O=[0,0]; %Joint 1 position
111 P1=[H01(1,4) H01(2,4)]; %Joint 2 position
112 P2=[H02(1,4) H02(2,4)]; %Joint 3 position
113
114
115 Orn= atan2(H02(2,1),H02(1,1)); %Orientation of end effector
116 Orn=(Orn)*(180/pi);
117 %subplot(221)
118 plot(P1(1),P1(2),'ok','LineWidth',5)
119 hold on
120 plot(P2(1),P2(2),'om','LineWidth',5)
121 plot(0,0,'ok','LineWidth',10)
122 xlim([-2.5 2.5])
123 ylim([-2.5 2.5])
124 grid minor
125 plot([0 P1(1)], [0 P1(2)], 'r', 'LineWidth', 5)
126 plot([P1(1) P2(1)], [P1(2) P2(2)], 'b', 'LineWidth', 5)
127 hold off
128 title(strcat('Time = ',num2str(t(i,1))), 'Interpreter','latex')
129 xlabel('X axis (m)', 'Interpreter','latex')
130 ylabel('Y axis (m)', 'Interpreter','latex')
131 set(gca, 'FontSize', 18)
132 pause(0.00000000000000000001);
133 F(c) = getframe(gcf) ;
134 c=c+1;
135 end
136
137 % create the video writer with 30 fps
138 writerObj = VideoWriter('Q3_computed_a.avi');
139 writerObj.FrameRate = 30;
140 % set the seconds per image
141 % open the video writer
142 open(writerObj);
143 % write the frames to the video
144 for i=1:length(F)
145     % convert the image to a frame
146     frame = F(i) ;
147     writeVideo(writerObj, frame);
148 end
149 % close the writer object
150 close(writerObj);

```

## Part B

### Animation of Q3 B Computed trajectory

```

1 % ME 639: Introduction to robotics
2 % Midsem exam : Question 3 (b)
3 % 3 Oct 2018
4 % Repeat above part (a) when a point mass, m e = 1 , is added at the center of mass of the
5 % link 2. Plot the results and discuss.
6 %
7 % Author: Shail Jadav 18310039
8 %% Initialization
9 clear
10 close all
11 clc
12 %% ODE solver computed
13
14 [t,x]=ode45('ode_solver_script_q3_b',[0,10],[0,0,0,0]);
15
16 th1=x(:,1); dth1=x(:,2); th2=x(:,3); dth2=x(:,4);
17
18 %% Trejectory genration plannned
19 [~,q0,dq0,ddq0]=trecgen(0,1/10,10,0,0,pi/6,0); %Trejectory generation for theta 1 0 to pi/6
20 [tq,q1,dq1,ddq1]=trecgen(0,1/10,10,0,0,pi/3,0); %Trejectory generation for theta 2 0 to pi/3
21
22
23 %% Computed
24 figure('units','normalized','outerposition',[0 0 1 1])
25 c=1;
26 for i=1:2:length(th1)
27
28     theta1=th1(i,1);
29     theta2=th2(i,1);

```



## 5 Problem 4

(Under-actuated System) Use the manipulator parameters as in Q1 and the initial condition (at  $t = 0$ ),  $IC = [q_1 \ q_2 \ \dot{q}_1 \ \dot{q}_2] = [0 \ 0 \ 0 \ 0]$ . 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,  $[t_{i f}] = [0 \ 10]$ . 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.

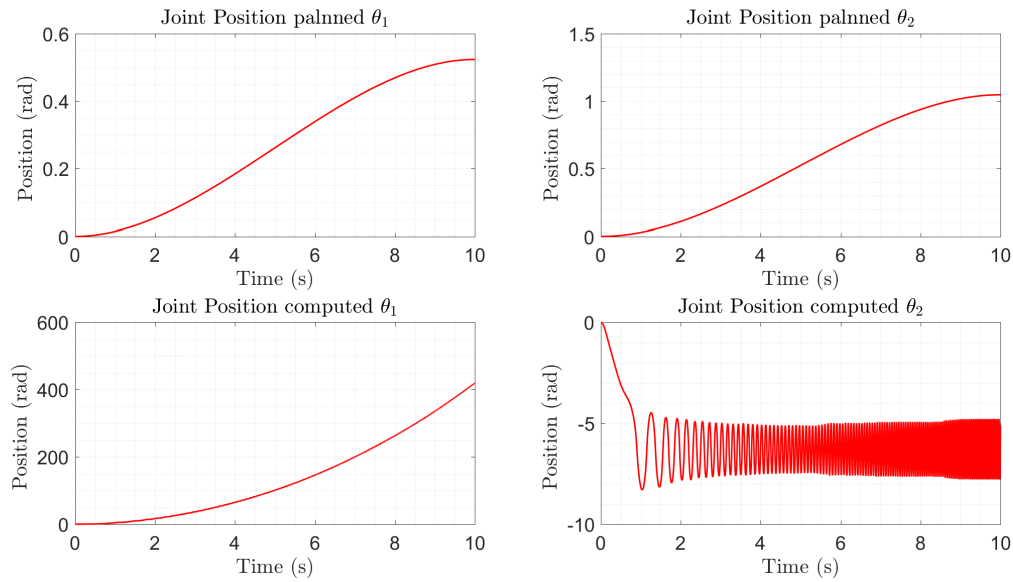


Figure 18: Comparison of the computed joint positions with the desired trajectory as planned

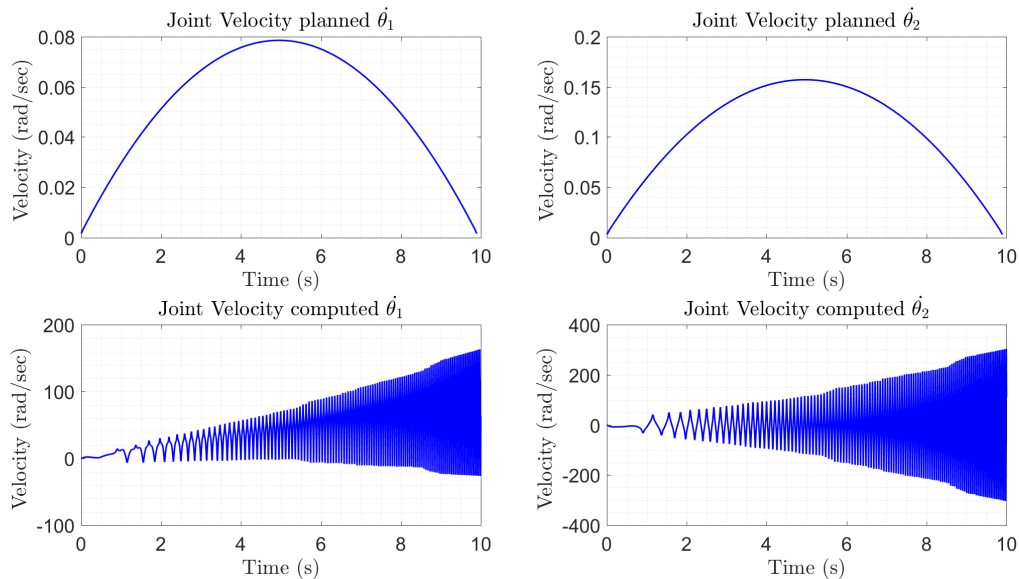


Figure 19: Comparison of the computed joint velocities with the desired trajectory as planned

As the  $\tau_2$  is zero the the joint 2 is free to rotate, the  $\tau_1$  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

```

1 % ME 639: Introduction to robotics
2 % Midsem exam : Question 4 (a)
3 % 3 Oct 2018
4 % a. For T 1 as computed in Q2(b), solve Eq. (1) using a numerical solver, say ode45, over a
5 % span of time, [t i t f] = [0 10] . Compare the computed joint positions and velocities with
6 % the desired trajectory as planned in Q2 (a). Plot the results and discuss.
7 %
8 % Author: Shail Jadav 18310039
9 %% Initialization
10 clear
11 close all
12 clc
13 %% ODE solver computed
14
15 [t,x]=ode45('ode_solver_script_q4_a',[0,10],[0,0,0,0]);
16 m1=1; m2=1; l1=1; l2=1; g=9.81;
17
18 th1=x(:,1); dth1=x(:,2); th2=x(:,3); dth2=x(:,4);
19
20 %% Trejectory generation palnned
21 [~,q0,dq0,ddq0]=trecgen(0,1/10,10,0,0,pi/6,0); %Trejectory generation for theta 1 0 to pi/6
22 [tq,q1,dq1,ddq1]=trecgen(0,1/10,10,0,0,pi/3,0); %Trejectory generation for theta 2 0 to pi/3
23
24
25
26 %% Display The Results
27
28 figure('units','normalized','outerposition',[0 0 1 1])
29 subplot(223)
30 plot(t,th1,'r','LineWidth',1.5)
31 title('Joint Position computed $\theta_1$', 'Interpreter','latex')
32 xlabel('Time (s)', 'Interpreter','latex')
33 ylabel('Position (rad) ', 'Interpreter','latex')
34 ylim([-40 40])
35 set(gca,'FontSize',18)
36 grid minor
37
38
39 subplot(224)
40 plot(t,th2,'r','LineWidth',1.5)
41 title('Joint Position computed $\theta_2$', 'Interpreter','latex')
42 xlabel('Time (s)', 'Interpreter','latex')
43 ylabel('Position (rad) ', 'Interpreter','latex')
44 ylim([-40 40])
45 set(gca,'FontSize',18)
46 grid minor
47 set(gca)
48
49
50 subplot(221)
51 plot(tq,q0,'r','LineWidth',1.5)
52 title('Joint Position palnned $\theta_1$', 'Interpreter','latex')
53 xlabel('Time (s)', 'Interpreter','latex')
54 ylabel('Position (rad) ', 'Interpreter','latex')
55 ylim([-40 40])
56 set(gca,'FontSize',18)
57 grid minor
58
59
60 subplot(222)
61 plot(tq,q1,'r','LineWidth',1.5)
62 title('Joint Position palnned $\theta_2$', 'Interpreter','latex')
63 xlabel('Time (s)', 'Interpreter','latex')
64 ylabel('Position (rad) ', 'Interpreter','latex')
65 ylim([-40 40])
66 set(gca,'FontSize',18)

```

```

67 grid minor
68 set(gca)
69 saveas(gcf,'Q4_a_JP.png')
70
71 figure('units','normalized','outerposition',[0 0 1 1])
72
73 subplot(223)
74 plot(t,dth1,'b','LineWidth',1.5)
75 title('Joint Velocity computed $\dot{\theta}_1$', 'Interpreter','latex')
76 xlabel('Time (s)', 'Interpreter','latex')
77 ylabel('Velocity (rad/sec)', 'Interpreter','latex')
78 %ylim([-40 40])
79 set(gca,'FontSize',18)
80 grid minor
81
82
83 subplot(224)
84 plot(t,dth2,'b','LineWidth',1.5)
85 title('Joint Velocity computed $\dot{\theta}_2$', 'Interpreter','latex')
86 xlabel('Time (s)', 'Interpreter','latex')
87 ylabel('Velocity (rad/sec)', 'Interpreter','latex')
88 %ylim([-40 40])
89 set(gca,'FontSize',18)
90 grid minor
91 set(gca)
92
93
94
95
96 subplot(221)
97 plot(tq(1,1:end-1),dq0,'b','LineWidth',1.5)
98 title('Joint Velocity planned $\dot{\theta}_1$', 'Interpreter','latex')
99 xlabel('Time (s)', 'Interpreter','latex')
100 ylabel('Velocity (rad/sec)', 'Interpreter','latex')
101 %ylim([-40 40])
102 set(gca,'FontSize',18)
103 grid minor
104
105
106 subplot(222)
107 plot(tq(1,1:end-1),dq1,'b','LineWidth',1.5)
108 title('Joint Velocity planned $\dot{\theta}_2$', 'Interpreter','latex')
109 xlabel('Time (s)', 'Interpreter','latex')
110 ylabel('Velocity (rad/sec)', 'Interpreter','latex')
111 %ylim([-40 40])
112 set(gca,'FontSize',18)
113 grid minor
114 set(gca)
115
116 saveas(gcf,'Q4_a_JV.png')

```

#### Function of ODE

```

1 function Out= ode_solver_script_q4_a(t,x)
2
3 %% Input parameters
4 m2=1; m1=1; I1=1; I2=1; g=9.81;
5
6 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
7 tau2=0; %Torque 2 is zero
8
9 %% Equation of motion
10 M11=((((m1/3) + m2)*I1^2) + ((m2/3)*I2^2) + (m2*I1*I2*cos(x(3))));
11 M12=(m2*((I2^2)/3) + (0.5*I1*I2*cos(x(3))));
12 M21=M12;
13 M22=((1/3)*m2*I2*I2);
14
15
16 H1 =((-m2*I1*I2*sin(x(3))*x(2)*x(4)) - (0.5*m2*I1*I2*sin(x(3))*x(4)*x(4)));
17 H2 = (0.5 * m2* I1*I2*sin(x(3))*x(2)*x(2));
18
19 G1=( (((0.5*m1) + m2)*I1*cos(x(1))) + (0.5*m2*I2*cos(x(1)+x(3))))*g);
20 G2=0.5*m2*I2*cos(x(1)+x(3))*g;

```



```

21
22 T=[tau1 ; tau2];
23
24 M=[M11 M12;M21 M22];
25
26 HG = [H1 + G1; H2 + G2];
27
28 %% Equation in terms of acceleration
29
30 ddth = (inv(M)) * (T - HG) ;
31
32 OP=zeros(4,1);
33
34 %% Output
35 OP(1)=x(2); %Intergration of velocity will give the position for theta 1
36 OP(2)=ddth(1);%Intergration of acceleration will give the velocity for theta 1
37 OP(3)=x(4); %Intergration of velocity will give the position for theta 2
38 OP(4)=ddth(2); %Intergration of acceleration will give the velocity for theta 2
39
40
41 Out=OP; % Output
42
43 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.

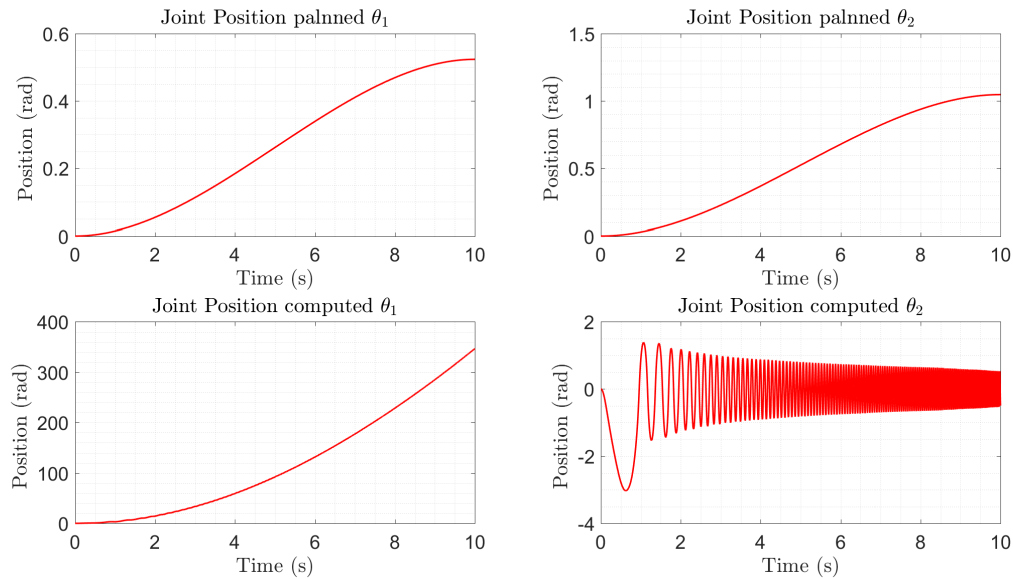


Figure 20: Comparison of the computed joint positions with the desired trajectory as planned

Initially  $\tau_2$  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 position 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.

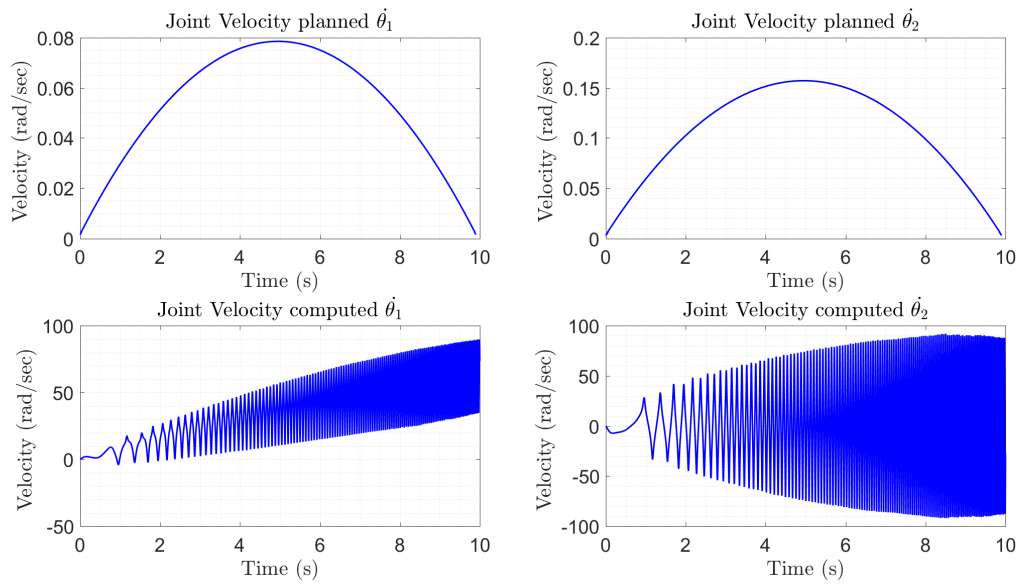


Figure 21: Comparison of the computed joint velocities with the desired trajectory as planned

### 5.2.1 Matlab Program

```

1 % ME 639: Introduction to robotics
2 % Midsem exam : Question 4 (b)
3 % 3 Oct 2018
4 % Repeat above part (a) when a rotational spring,  $k_r = 1$ , is added at the joint 2 between
5 % link 1 and 2. Plot the results and discuss, how does the value of  $k_r$  change the results.
6 %
7 % Author: Shail Jadav 18310039
8 %% Initialization
9 clear
10 close all
11 clc
12 %% ODE solver computed
13
14 [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]
15 % Thetal=pi/4 Theta2=pi/4
16 m1=1; m2=1; l1=1; l2=1; g=9.81;
17 th1=x(:,1); dth1=x(:,2); th2=x(:,3); dth2=x(:,4);
18
19 %% Trejectory genration planned
20 [~,q0,dq0,ddq0]=trecegen(0,1/10,10,0,0,0,pi/6,0); %Trejectory generation for theta 1 0 to pi/6
21 [tq,q1,dq1,ddq1]=trecegen(0,1/10,10,0,0,0,pi/3,0); %Trejectory generation for theta 2 0 to pi/3
22
23
24
25
26 %% Display The Results
27
28 figure('units','normalized','outerposition',[0 0 1 1])
29 subplot(223)
30 plot(t,th1,'r','LineWidth',1.5)
31 title('Joint Position computed  $\theta_1$ ','Interpreter','latex')
32 xlabel('Time (s)','Interpreter','latex')
33 ylabel('Position (rad)','Interpreter','latex')
34 ylim([-40 40])
35 set(gca,'FontSize',18)
36 grid minor
37
38
39 subplot(224)
40 plot(t,th2,'r','LineWidth',1.5)

```

```

41 title('Joint Position computed  $\theta_2$ ','Interpreter','latex')
42 xlabel('Time (s)','Interpreter','latex')
43 ylabel('Position (rad) ','Interpreter','latex')
44 %ylim([-40 40])
45 set(gca,'FontSize',18)
46 grid minor
47 set(gca)
48
49
50 subplot(221)
51 plot(tq,q0,'r','LineWidth',1.5)
52 title('Joint Position planned  $\theta_1$ ','Interpreter','latex')
53 xlabel('Time (s)','Interpreter','latex')
54 ylabel('Position (rad) ','Interpreter','latex')
55 %ylim([-40 40])
56 set(gca,'FontSize',18)
57 grid minor
58
59
60 subplot(222)
61 plot(tq,q1,'r','LineWidth',1.5)
62 title('Joint Position planned  $\theta_2$ ','Interpreter','latex')
63 xlabel('Time (s)','Interpreter','latex')
64 ylabel('Position (rad) ','Interpreter','latex')
65 %ylim([-40 40])
66 set(gca,'FontSize',18)
67 grid minor
68 set(gca)
69 saveas(gcf,'Q4_b_JP.png')
70
71 figure('units','normalized','outerposition',[0 0 1 1])
72
73 subplot(223)
74 plot(t,dth1,'b','LineWidth',1.5)
75 title('Joint Velocity computed  $\dot{\theta}_1$ ','Interpreter','latex')
76 xlabel('Time (s)','Interpreter','latex')
77 ylabel('Velocity (rad/sec) ','Interpreter','latex')
78 %ylim([-40 40])
79 set(gca,'FontSize',18)
80 grid minor
81
82
83 subplot(224)
84 plot(t,dth2,'b','LineWidth',1.5)
85 title('Joint Velocity computed  $\dot{\theta}_2$ ','Interpreter','latex')
86 xlabel('Time (s)','Interpreter','latex')
87 ylabel('Velocity (rad/sec) ','Interpreter','latex')
88 %ylim([-40 40])
89 set(gca,'FontSize',18)
90 grid minor
91 set(gca)
92
93
94
95
96 subplot(221)
97 plot(tq(1,1:end-1),dq0,'b','LineWidth',1.5)
98 title('Joint Velocity planned  $\dot{\theta}_1$ ','Interpreter','latex')
99 xlabel('Time (s)','Interpreter','latex')
100 ylabel('Velocity (rad/sec) ','Interpreter','latex')
101 %ylim([-40 40])
102 set(gca,'FontSize',18)
103 grid minor
104
105
106 subplot(222)
107 plot(tq(1,1:end-1),dq1,'b','LineWidth',1.5)
108 title('Joint Velocity planned  $\dot{\theta}_2$ ','Interpreter','latex')
109 xlabel('Time (s)','Interpreter','latex')
110 ylabel('Velocity (rad/sec) ','Interpreter','latex')
111 %ylim([-40 40])
112 set(gca,'FontSize',18)
113 grid minor

```

```

114 set(gca)
115
116 saveas(gcf, 'Q4_b_JV.png')

```

### Function of ODE

```

1 function Out= ode_solver_script_q4_b(t,x)
2
3 %% Input parameters
4 m2=1; m1=1; l1=1; l2=1; g=9.81; k=1; %k is spring constant
5
6 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
7 tau2=0;
8
9
10 %% Equation of motion
11 M11=((((m1/3) + m2)*l1^2) +((m2/3)*l2^2) + (m2*l1*l2*cos(x(3))));
12 M12=(m2*((l2^2)/3) + (0.5*l1*l2*cos(x(3))));
13 M21=M12;
14 M22=((1/3)*m2*l2*l2);
15
16
17 H1 =((-m2*l1*l2*sin(x(3))*x(2)*x(4)) - (0.5*m2*l1*l2*sin(x(3))*x(4)*x(4)));
18 H2 = (0.5 * m2* l1*l2*sin(x(3))*x(2)*x(2));
19
20 G1=( (((0.5*m1) + m2)*l1*cos(x(1))) + (0.5*m2*l2*cos(x(1)+x(3))))*g);
21 G2=0.5*m2*l2*cos(x(1)+x(3))*g;
22
23 T=[tau1 ; tau2];
24
25 M=[M11 M12;M21 M22];
26
27 HG = [H1 + G1; H2 + G2];
28
29 S=[0;k*x(3)]; %Torque due to spring
30
31 %% Equation in terms of acceleration
32
33 ddth = (inv(M)) * (T - HG -S) ;
34
35 OP=zeros(4,1);
36
37 %% Output
38 OP(1)=x(2); %Interpretation of velocity will give the position for theta 1
39 OP(2)=ddth(1);%Interpretation of acceleration will give the velocity for theta 1
40 OP(3)=x(4); %Interpretation of velocity will give the position for theta 2
41 OP(4)=ddth(2); %Interpretation of acceleration will give the velocity for theta 2
42
43
44 Out=OP; % Output
45
46 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

```

1 % ME 639: Introduction to robotics
2 % Midsem exam : Question 4 (a)
3 % 3 Oct 2018
4 % a. For T1 as computed in Q2(b), solve Eq. (1) using a numerical solver, say ode45, over a
5 % span of time, [t i t f] = [0 10] . Compare the computed joint positions and velocities with
6 % the desired trajectory as planned in Q2 (a). Plot the results and discuss.
7 %
8 % Author: Shail Jadav 18310039
9 %% Initialization

```

```

10 clear
11 close all
12 clc
13 %% ODE solver computed
14
15 [t,x]=ode45('ode_solver_script_q4_a',[0,10],[0,0,0,0]);
16 m1=1; m2=1; l1=1; l2=1; g=9.81;
17
18 th1=x(:,1); dth1=x(:,2); th2=x(:,3); dth2=x(:,4);
19
20 %% Trajectory generation planned
21 [~,q0,dq0,ddq0]=trecegen(0,1/10,10,0,0,pi/6,0); %Trajectory generation for theta 1 0 to pi/6
22 [tq,q1,dq1,ddq1]=trecegen(0,1/10,10,0,0,pi/3,0); %Trajectory generation for theta 2 0 to pi/3
23
24
25
26 %% Planned
27 figure('units','normalized','outerposition',[0 0 1 1])
28 c=1;
29 for i=1:10:length(q0)
30
31     theta1=q0(1,i);
32     theta2=q1(1,i);
33
34
35     l1=1; %Input the l length
36     l2=1; %Input the l length
37
38
39
40 % Homogeneous transformation matrix
41 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
42 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
43
44
45 H02=H01*H12; %Frame 0 to 2 tranformation
46
47
48 O=[0,0]; %Joint 1 position
49 P1=[H01(1,4) H01(2,4)]; %Joint 2 position
50 P2=[H02(1,4) H02(2,4)]; %Joint 3 position
51
52
53 Orn= atan2(H02(2,1),H02(1,1)); %Orientation of end effector
54 Orn=(Orn)*(180/pi);
55 %subplot(221)
56 plot(P1(1),P1(2),'ok','LineWidth',5)
57 hold on
58 plot(P2(1),P2(2),'om','LineWidth',5)
59 plot(0,0,'ok','LineWidth',10)
60 xlim([-2.5 2.5])
61 ylim([-2.5 2.5])
62 grid minor
63 plot([0 P1(1)], [0 P1(2)], 'r','LineWidth',5)
64 plot([P1(1) P2(1)], [P1(2) P2(2)], 'b','LineWidth',5)
65 hold off
66 title(strcat('Time = ',num2str(tq(1,i))),'Interpreter','latex')
67 xlabel('X axis (m)','Interpreter','latex')
68 ylabel('Y axis (m)','Interpreter','latex')
69 set(gca,'FontSize',18)
70 pause(0.00000000000000000001);
71 F(c) = getframe(gcf) ;
72 c=c+1;
73 end
74
75 % create the video writer with 30 fps
76 writerObj = VideoWriter('Q4_planned_a.avi');
77 writerObj.FrameRate = 30;
78 % set the seconds per image
79 % open the video writer
80 open(writerObj);

```



```

152 % close the writer object
153 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

```

1 % ME 639: Introduction to robotics
2 % Midsem exam : Question 4 (b)
3 % 3 Oct 2018
4 % b. Repeat above part (a) when a rotational spring,  $k_r = 1$ , is added at the joint 2 between
5 % link 1 and 2. Plot the results and discuss, how does the value of  $k_r$  change the results.
6 % Author: Shail Jadav 18310039
7 %% Initialization
8 clear
9 close all
10 clc
11 %% ODE solver computed
12
13 [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]
14 % Theta1=pi/4 Theta2=pi/4
15 m1=1; m2=1; l1=1; l2=1; g=9.81;
16
17 th1=x(:,1); dth1=x(:,2); th2=x(:,3); dth2=x(:,4);
18
19 %% Trejectory generation plannned
20 [~,q0,dq0,ddq0]=trecgen(0,1/10,10,0,0,pi/6,0); %Trejectory generation for theta 1 0 to pi/6
21 [tq,q1,dq1,ddq1]=trecgen(0,1/10,10,0,0,pi/3,0); %Trejectory generation for theta 2 0 to pi/3
22
23
24
25
26 %% Computed
27 figure('units','normalized','outerposition',[0 0 1 1])
28 c=1;
29 for i=1:3:length(th1)
30
31 theta1=th1(i,1);
32 theta2=th2(i,1);
33
34
35 l1=1; %Input the l length
36 l2=1; %Input the l length
37
38
39
40 % Homogeneous transformation matrix
41 H01 = [cos(theta1) -sin(theta1) 0 l1*cos(theta1); sin(theta1) cos(theta1) 0 l1*sin(theta1); 0 0 1
42 0;0 0 0 1]; %Frame 0 to 1 tranformation
43
44 H12 = [cos(theta2) -sin(theta2) 0 l2*cos(theta2); sin(theta2) cos(theta2) 0 l2*sin(theta2); 0 0 1
45 0;0 0 0 1]; %Frame 1 to 2 tranformation
46
47 H02=H01*H12; %Frame 0 to 2 tranformation
48
49 O=[0,0]; %Joint 1 position
50 P1=[H01(1,4) H01(2,4)]; %Joint 2 position
51 P2=[H02(1,4) H02(2,4)]; %Joint 3 position
52
53 Orn= atan2(H02(2,1),H02(1,1)); %Orientation of end effector
54 Orn=(Orn)*(180/pi);
55 %subplot(221)
56 plot(P1(1),P1(2),'ok','LineWidth',5)
57 hold on
58 plot(P2(1),P2(2),'om','LineWidth',5)
59 plot(0,0,'ok','LineWidth',10)
60 xlim([-2.5 2.5])

```

```

61 ylim([-2.5 2.5])
62 grid minor
63 plot([0 P1(1)], [0 P1(2)], 'r', 'LineWidth', 5)
64 plot([P1(1) P2(1)], [P1(2) P2(2)], 'b', 'LineWidth', 5)
65 hold off
66 title(strcat('Time = ', num2str(t(i,1))), 'Interpreter', 'latex')
67 xlabel('X axis (m)', 'Interpreter', 'latex')
68 ylabel('Y axis (m)', 'Interpreter', 'latex')
69 set(gca, 'FontSize', 18)
70 pause(0.00000000000000000001);
71 F(c) = getframe(gcf) ;
72 c=c+1;
73 end
74
75 % create the video writer with 30 fps
76 writerObj = VideoWriter('Q4_computed_b_k_1.avi');
77 writerObj.FrameRate = 30;
78 % set the seconds per image
79 % open the video writer
80 open(writerObj);
81 % write the frames to the video
82 for i=1:length(F)
83     % convert the image to a frame
84     frame = F(i) ;
85     writeVideo(writerObj, frame);
86 end
87 % close the writer object
88 close(writerObj);

```

## 6 Project proposal and feedback

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