HW2

Aly Khater

(1) cubic $- ax^3 + bx^2 + cx + d = f(x)$

six joints, four via points

$6 \times 4 = \boxed{24 \text{ individual cubic equations}}$

4 coefficients

$24 \times 4 = 96$ coefficients

(2) $\theta(t) = at^3 + bt^2 + ct + d$ $\qquad \theta(0) = -60^\circ \qquad \dot{\theta}(0) = 0$

a) $\dot{\theta}(t) = 3at^2 + 2bt + c$ $\qquad \theta(5) = 60^\circ \qquad \dot{\theta}(5) = 0$

$\theta(0) = d = -60$

$\theta(5) = 125a + 25b + 5c + d = 60$

$\dot{\theta}(0) = c = 0$

$\dot{\theta}(5) = 75a + 10b + c = 0$

$75a + 10b = 0$

$15a + 2b = 0$

$b = -\frac{15}{2}a \rightarrow b = 14.4$

$125a + 25b + 5c + d = 60$

$125a + 25b - 60 = 60$

$125a + 25b = 120$

$25a + 5b = 24$

$25a - \frac{75}{2}a = 24$

$a = -1.92$

$\theta(t) = -1.92t^3 + 14.4t^2 - 60$

(2) b) $\ddot{\theta} \geq \frac{4(\theta_f - \theta_0)}{t^2}$; $\theta_f = 60$, $\theta_f = -60$, $t = 5$

$$\ddot{\theta} \geq \frac{4(60+60)}{5^2} = \frac{480}{25} = \underline{19.2 \ deg/s^2}$$

Since blend acceleration is 85 deg/s², this is okay

$$t_b = \frac{t}{2} - \frac{\sqrt{\ddot{\theta}^2 t^2 - 4\ddot{\theta}(\theta_f - \theta_0)}}{2\ddot{\theta}}, \quad \ddot{\theta} = 85°, t = 5,$$

$$\frac{5}{2} - \frac{\sqrt{85^2(5^2) - 4(85)(120)}}{2(85)} \Rightarrow \frac{5}{2} - 2.2$$

$$t_b = 0.3 \ s$$

$$w = \ddot{\theta} t_b = (85)(0.3) = 25.5 \ deg/sec = w$$

$$T_{linear} = T - 2T_b = 5 - 2(0.3) = 4.4s = T_{linear}$$

(c,d) SEE MATLAB

e) jtraj is the smoothest directory.

f) jtraj uses sinusoidal functions to smooth the curve.

(3) $t_f = 3\,sec$, $t_0 = 0\,sec$

$\theta(t) = 10 + 90t^2 - 60t^3$
$\dot{\theta}(t) = 180t - 180t^2$
$\ddot{\theta}(t) = 180 - 360t$

$\boxed{\begin{matrix} \theta(0) = 10^\circ & \theta(3) = -800^\circ \\ \dot{\theta}(0) = 0 & \dot{\theta}(3) = -1080^\circ/s \\ \ddot{\theta}(0) = 180^\circ/s^2 & \ddot{\theta}(3) = -900^\circ/s^2 \end{matrix}}$

$\theta(3) = 10 + 90(3^2) - 60(3^3) = -800$

$\dot{\theta}(3) = 180(3) - 180(3^2) = -1080$

$\ddot{\theta}(3) = 180 - 360(3) = -900$

(4) a) Joint space,
It affects trajectory planning if the motion would bring the joint to move outside of its angular limit,

Cartesian,
The motion may cause it to fail and not reach the positions

b) The joint rates spike

Might be going into singularities, Very sensitive to changes,

See Matlab to see error,

c) $T3(:,:,21) = \begin{pmatrix} -.7242 & .6895 & 0 & -.9997 \\ -.6895 & -.7242 & 0 & .0245 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

Ikine failed

c) Doesn't consider Joint limitations,

d) - Singularity issues
Joint limit Violations
High Velocities,

e) Cartesian is predictable

(5) $m=1, b=4, k=5$. $\omega = 6$ rad/s

$\omega_N = 3$ rad/sec

$\omega_N = \sqrt{\frac{k'}{m}} = 3 = \sqrt{k'} = k' = 9$

$k_p = k' - k = 9 - 5 = \underline{4 = k_p}$

$b' = 2\sqrt{mk'} = 2\sqrt{9}$; $k_v = b' - b = 6 - 4 = \underline{2 = k_v}$

$= 6$

See Matlabb

(6) $m\ddot{x} + b\dot{x} + q_x x^3 = f$

$\alpha = m = 1$

$\beta = b\dot{x} + kx$ $\beta = 4\dot{x} + 5x$

$\ddot{x} = -k_v\dot{x} - k_p x$, e

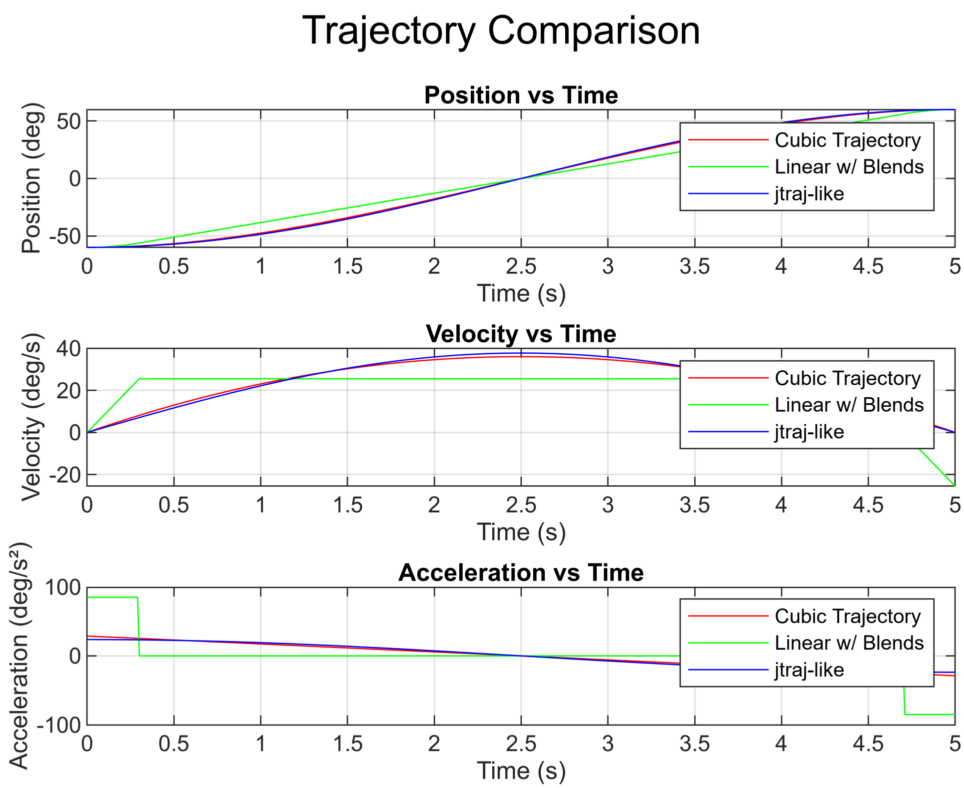
$k_v = 2\sqrt{k_p}$ for critical damping

$\underline{k_p = 4}$ $\underline{k_v = 4}$

See Matlab

```
%Problem 2 Aly Khater

clc;
clear;
close all;

% Time parameters
T = 5;  % total time in seconds
t = linspace(0, T, 500);  % time vector with 500 points

% Cubic trajectory from Part (a)
a_cubic = -1.92;
b_cubic = 14.4;
c_cubic = 0;
d_cubic = -60;

theta_cubic = a_cubic * t.^3 + b_cubic * t.^2 + c_cubic * t + d_cubic;
velocity_cubic = 3 * a_cubic * t.^2 + 2 * b_cubic * t + c_cubic;
acceleration_cubic = 6 * a_cubic * t + 2 * b_cubic;

% Linear trajectory with parabolic blends from Part (b)
a_b = 85;  % blend acceleration in deg/sec^2
T_b = 0.3;  % blend time in seconds
omega_max = a_b * T_b;
T_linear = T - 2 * T_b;

theta_linear = zeros(1, length(t));
velocity_linear = zeros(1, length(t));
acceleration_linear = zeros(1, length(t));

for i = 1:length(t)
    if t(i) < T_b
        theta_linear(i) = -60 + 0.5 * a_b * t(i)^2;
        velocity_linear(i) = a_b * t(i);
        acceleration_linear(i) = a_b;
    elseif t(i) <= T - T_b
        theta_linear(i) = -60 + 0.5 * a_b * T_b^2 + omega_max * (t(i) - T_b);
        velocity_linear(i) = omega_max;
        acceleration_linear(i) = 0;
    else
        theta_linear(i) = 60 - 0.5 * a_b * (T - t(i))^2;
        velocity_linear(i) = -a_b * (t(i) - (T - T_b));
        acceleration_linear(i) = -a_b;
    end
end

% jtraj-like approximation (using a simple sine blend for smooth transitions)
theta_jtraj = -60 + (60 + 60) * (0.5 * (1 - cos(pi * t / T)));
velocity_jtraj = (60 + 60) * (0.5 * pi / T) * sin(pi * t / T);
acceleration_jtraj = (60 + 60) * (0.5 * pi^2 / T^2) * cos(pi * t / T);
```

```
% Plotting
figure;

% Position Plot
subplot(3, 1, 1);
plot(t, theta_cubic, 'r', 'DisplayName', 'Cubic Trajectory');
hold on;
plot(t, theta_linear, 'g', 'DisplayName', 'Linear w/ Blends');
plot(t, theta_jtraj, 'b', 'DisplayName', 'jtraj-like');
title('Position vs Time');
xlabel('Time (s)');
ylabel('Position (deg)');
legend;
grid on;

% Velocity Plot
subplot(3, 1, 2);
plot(t, velocity_cubic, 'r', 'DisplayName', 'Cubic Trajectory');
hold on;
plot(t, velocity_linear, 'g', 'DisplayName', 'Linear w/ Blends');
plot(t, velocity_jtraj, 'b', 'DisplayName', 'jtraj-like');
title('Velocity vs Time');
xlabel('Time (s)');
ylabel('Velocity (deg/s)');
legend;
grid on;

% Acceleration Plot
subplot(3, 1, 3);
plot(t, acceleration_cubic, 'r', 'DisplayName', 'Cubic Trajectory');
hold on;
plot(t, acceleration_linear, 'g', 'DisplayName', 'Linear w/ Blends');
plot(t, acceleration_jtraj, 'b', 'DisplayName', 'jtraj-like');
title('Acceleration vs Time');
xlabel('Time (s)');
ylabel('Acceleration (deg/s²)');
legend;
grid on;

% Adjust layout
sgtitle('Trajectory Comparison');
```




```
%Problem 4
%RR robot - trajectory generation example
hold off
axis
```

```
ans = 1×4
     0     5  -100   100
```

```
clear all
SCURRTWOLINK
%robot definition
L1 = 1;
Link1 = link([0 0 0 0 0], 'modified');
Link2 = link([0 L1 0 0 0], 'modified');
r = robot({Link1 Link2});

plotoption=8;
%establish a time vector
t=[0:.5:10];
%define initial and final robot joint space poses
Qi=[0;pi/4]; %These are the values for the lecture example
Qf=[3*pi/4;pi/2];
%generate a trajectory plan
[q qd qdd]=jtraj(Qi, Qf, t);
%show how angles, velocities, accelerations evolve
if plotoption==1
```

```
subplot(3,1,1), plot(t, q)
subplot(3,1,2), plot(t, qd)
subplot(3,1,3), plot(t, qdd)
end
%show how robot moves in a 'straight line' in joint space
if plotoption==2
plot(q*[1;0],q*[0;1])
end
%show how the motion is not 'straight' in operational space
TJ=fkine(r, q)
```

```
TJ =
TJ(:,:,1) =

    0.7071   -0.7071         0    1.0000
    0.7071    0.7071         0         0
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,2) =

    0.7045   -0.7097         0    1.0000
    0.7097    0.7045         0    0.0027
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,3) =

    0.6878   -0.7259         0    0.9998
    0.7259    0.6878         0    0.0202
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,4) =

    0.6456   -0.7637         0    0.9980
    0.7637    0.6456         0    0.0627
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,5) =

    0.5675   -0.8234         0    0.9907
    0.8234    0.5675         0    0.1360
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,6) =

    0.4441   -0.8960         0    0.9704
    0.8960    0.4441         0    0.2415
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,7) =
```

```
    0.2697   -0.9629         0    0.9271
    0.9629    0.2697         0    0.3749
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,8) =

    0.0466   -0.9989         0    0.8504
    0.9989    0.0466         0    0.5262
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,9) =

   -0.2103   -0.9776         0    0.7331
    0.9776   -0.2103         0    0.6801
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,10) =

   -0.4731   -0.8810         0    0.5746
    0.8810   -0.4731         0    0.8184
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,11) =

   -0.7071   -0.7071         0    0.3827
    0.7071   -0.7071         0    0.9239
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,12) =

   -0.8810   -0.4731         0    0.1724
    0.4731   -0.8810         0    0.9850
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,13) =

   -0.9776   -0.2103         0   -0.0374
    0.2103   -0.9776         0    0.9993
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,14) =

   -0.9989    0.0466         0   -0.2292
   -0.0466   -0.9989         0    0.9734
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,15) =
```

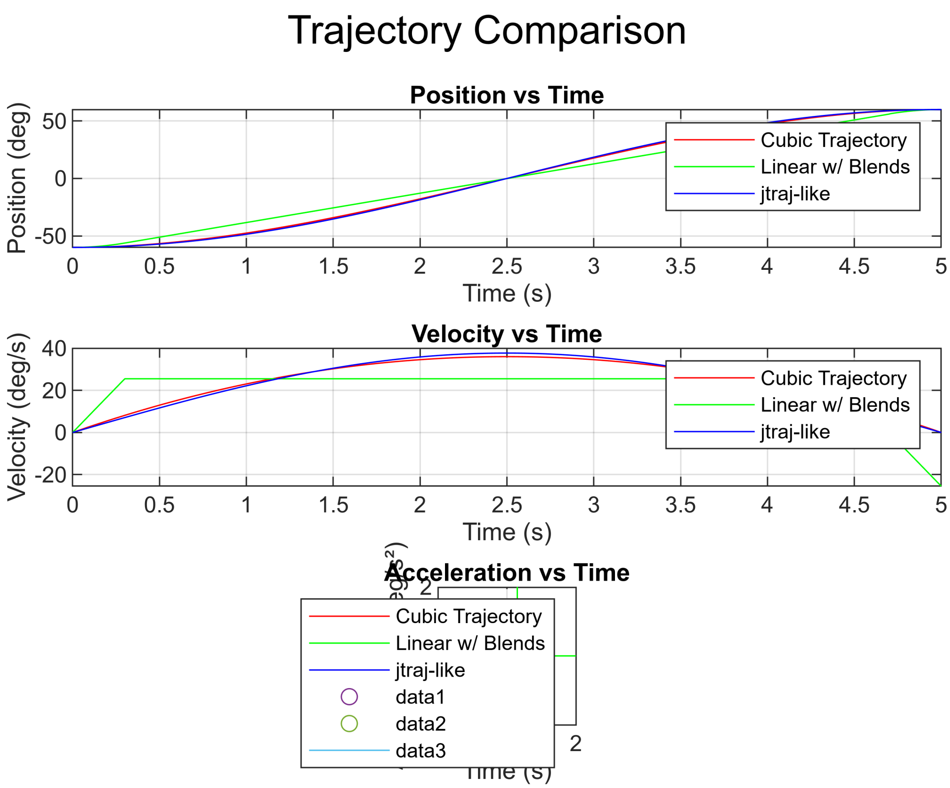
```
   -0.9629    0.2697         0   -0.3905
   -0.2697   -0.9629         0    0.9206
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,16) =

   -0.8960    0.4441         0   -0.5154
   -0.4441   -0.8960         0    0.8569
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,17) =

   -0.8234    0.5675         0   -0.6043
   -0.5675   -0.8234         0    0.7967
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,18) =

   -0.7637    0.6456         0   -0.6614
   -0.6456   -0.7637         0    0.7500
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,19) =

   -0.7259    0.6878         0   -0.6927
   -0.6878   -0.7259         0    0.7212
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,20) =

   -0.7097    0.7045         0   -0.7052
   -0.7045   -0.7097         0    0.7090
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,21) =

   -0.7071    0.7071         0   -0.7071
   -0.7071   -0.7071         0    0.7071
         0         0    1.0000         0
         0         0         0    1.0000
```

```
Ree=transl(TJ);
if plotoption==3
subplot(2,1,1), plot(t,Ree(:,1))
subplot(2,1,2), plot(t,Ree(:,2))
end
if plotoption==4
plot(r, q, 'loop')
end
if plotoption==5
```

```
axis('square'); axis([-2 2 -2 2]); axis manual; hold on;
for z=1:1:length(t)
plot(cos(q(z,1)), sin(q(z,1)), 'o')
plot(Ree(z,1), Ree(z,2), 'o')
plot([0;cos(q(z,1));Ree(z,1)],[0;sin(q(z,1));Ree(z,2)])
end
end
% Now let's try doing the planning in Cartesian Space
Tinit=fkine(r,Qi);
Tfinal=fkine(r, Qf);
rr=jtraj(0,1,t);
TC = ctraj(Tinit, Tfinal, rr);
%Endpoint x and y coordinates over time
if plotoption==6
plot(t, transl(TC)); grid;
end
%Endpoint in Cartesian space
k=transl(TC);
if plotoption==7
axis('square'); axis([-2 2 -2 2]); grid
plot(k(:,1),k(:,2))
end
%The resulting joint angles
Q=ikine(r, TC, [0;pi/2], [1 1 0 0 0 0])
```

```
Q = 21×2
         0    1.5708
    0.0008    1.5700
    0.0061    1.5647
    0.0197    1.5511
    0.0454    1.5254
    0.0887    1.4821
    0.1585    1.4123
    0.2710    1.2998
    0.4556    1.1152
    0.7555    0.8153
      ⋮
```

```
%Which from an overhead view gives
if plotoption==8
axis('square'); axis([-2 2 -2 2]); axis manual; hold on;
for z=1:1:length(t)
plot(cos(Q(z,1)), sin(Q(z,1)), 'o')
plot(k(z,1), k(z,2), 'o')
plot([0;cos(Q(z,1));k(z,1)],[0;sin(Q(z,1));k(z,2)])
pause
end
end
```




```
%Problem 4b
%RR robot - trajectory generation example
hold off
axis
```

```
ans = 1×4
    -2     2    -2     2
```

```
clear all
SCURRTWOLINK
%robot definition
L1 = 1;
Link1 = link([0 0 0 0 0], 'modified');
Link2 = link([0 L1 0 0 0], 'modified');
r = robot({Link1 Link2});

plotoption=8;
%establish a time vector
t=[0:.5:10];
%define initial and final robot joint space poses
Qi=[0;pi/4]; %These are the values for the lecture example
Qf=[15*pi/16;pi/4];
%generate a trajectory plan
[q qd qdd]=jtraj(Qi, Qf, t);
%show how angles, velocities, accelerations evolve
if plotoption==1
```

```
subplot(3,1,1), plot(t, q)
subplot(3,1,2), plot(t, qd)
subplot(3,1,3), plot(t, qdd)
end
%show how robot moves in a 'straight line' in joint space
if plotoption==2
plot(q*[1;0],q*[0;1])
end
%show how the motion is not 'straight' in operational space
TJ=fkine(r, q)
```

```
TJ =
TJ(:,:,1) =

    0.7071   -0.7071         0    1.0000
    0.7071    0.7071         0         0
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,2) =

    0.7047   -0.7095         0    1.0000
    0.7095    0.7047         0    0.0034
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,3) =

    0.6891   -0.7247         0    0.9997
    0.7247    0.6891         0    0.0252
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,4) =

    0.6496   -0.7603         0    0.9969
    0.7603    0.6496         0    0.0783
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,5) =

    0.5768   -0.8169         0    0.9855
    0.8169    0.5768         0    0.1698
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,6) =

    0.4622   -0.8868         0    0.9539
    0.8868    0.4622         0    0.3002
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,7) =
```

```
    0.3004   -0.9538         0    0.8869
    0.9538    0.3004         0    0.4621
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,8) =

    0.0926   -0.9957         0    0.7696
    0.9957    0.0926         0    0.6386
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,9) =

   -0.1490   -0.9888         0    0.5939
    0.9888   -0.1490         0    0.8046
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,10) =

   -0.4013   -0.9159         0    0.3639
    0.9159   -0.4013         0    0.9314
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,11) =

   -0.6344   -0.7730         0    0.0980
    0.7730   -0.6344         0    0.9952
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,12) =

   -0.8201   -0.5723         0   -0.1752
    0.5723   -0.8201         0    0.9845
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,13) =

   -0.9408   -0.3390         0   -0.4255
    0.3390   -0.9408         0    0.9050
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,14) =

   -0.9946   -0.1034         0   -0.6302
    0.1034   -0.9946         0    0.7764
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,15) =
```

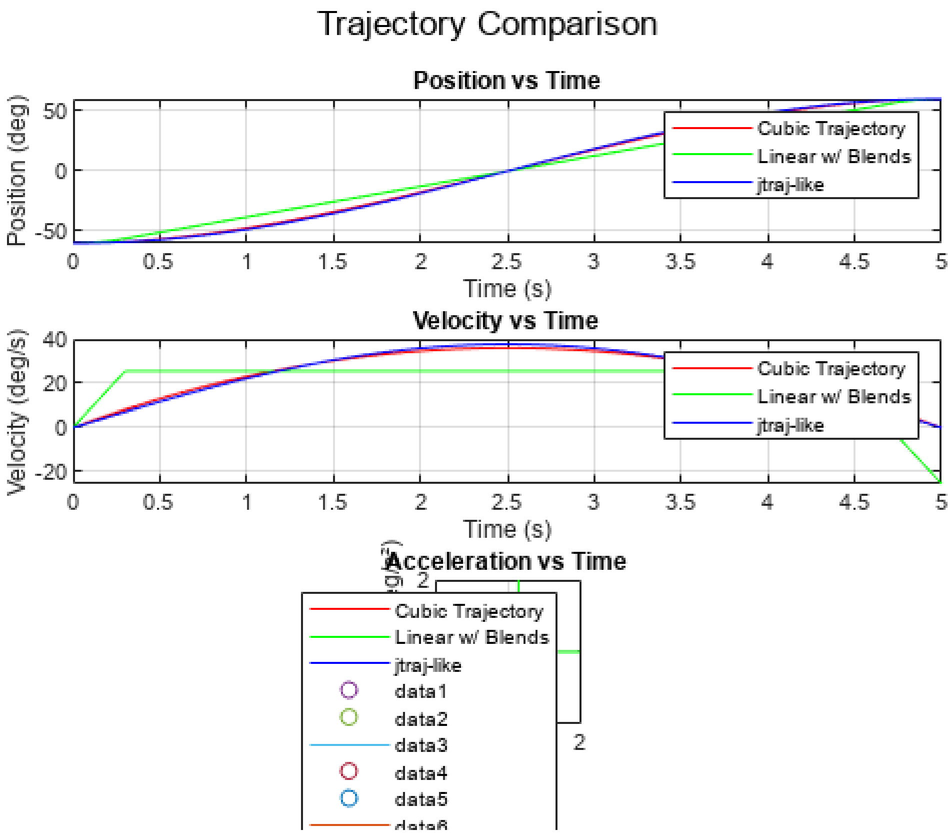
```
   -0.9941    0.1085         0   -0.7797
   -0.1085   -0.9941         0    0.6262
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,16) =

   -0.9599    0.2804         0   -0.8770
   -0.2804   -0.9599         0    0.4805
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,17) =

   -0.9137    0.4064         0   -0.9334
   -0.4064   -0.9137         0    0.3588
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,18) =

   -0.8724    0.4888         0   -0.9625
   -0.4888   -0.8724         0    0.2713
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,19) =

   -0.8452    0.5344         0   -0.9756
   -0.5344   -0.8452         0    0.2198
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,20) =

   -0.8334    0.5527         0   -0.9801
   -0.5527   -0.8334         0    0.1984
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,21) =

   -0.8315    0.5556         0   -0.9808
   -0.5556   -0.8315         0    0.1951
         0         0    1.0000         0
         0         0         0    1.0000
```

```
Ree=transl(TJ);
if plotoption==3
subplot(2,1,1), plot(t,Ree(:,1))
subplot(2,1,2), plot(t,Ree(:,2))
end
if plotoption==4
plot(r, q, 'loop')
end
if plotoption==5
```

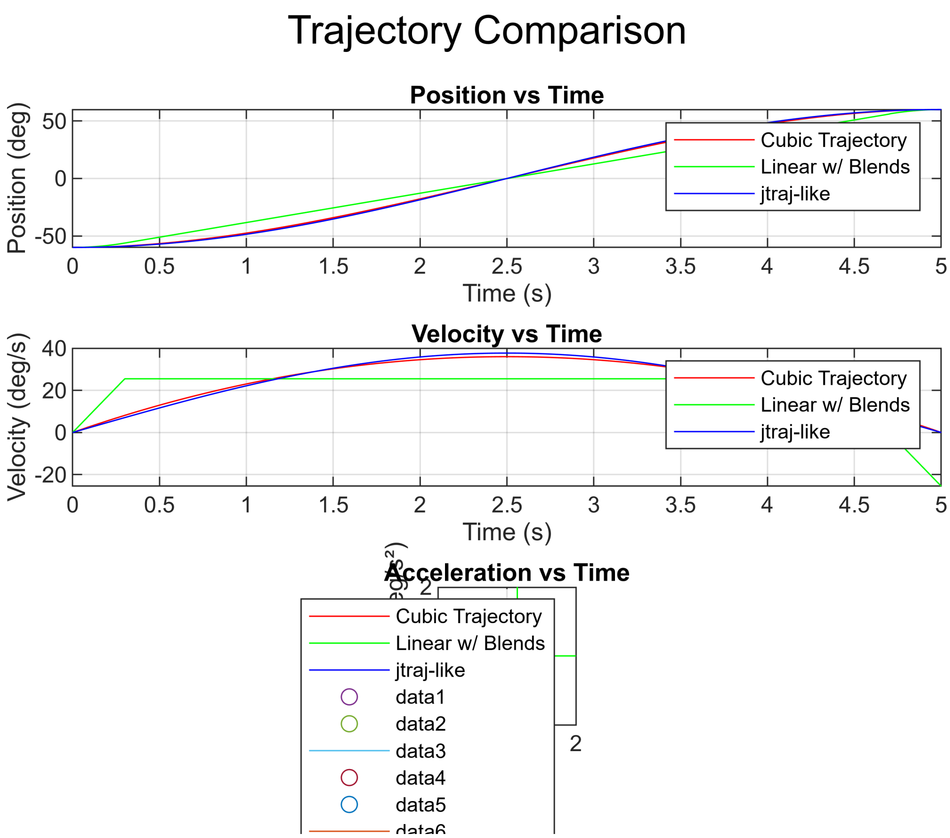
```
axis('square'); axis([-2 2 -2 2]); axis manual; hold on;
for z=1:1:length(t)
plot(cos(q(z,1)), sin(q(z,1)), 'o')
plot(Ree(z,1), Ree(z,2), 'o')
plot([0;cos(q(z,1));Ree(z,1)],[0;sin(q(z,1));Ree(z,2)])
end
end
% Now let's try doing the planning in Cartesian Space
Tinit=fkine(r,Qi);
Tfinal=fkine(r, Qf);
rr=jtraj(0,1,t);
TC = ctraj(Tinit, Tfinal, rr);
%Endpoint x and y coordinates over time
if plotoption==6
plot(t, transl(TC)); grid;
end
%Endpoint in Cartesian space
k=transl(TC);
if plotoption==7
axis('square'); axis([-2 2 -2 2]); grid
plot(k(:,1),k(:,2))
end
%The resulting joint angles
Q=ikine(r, TC, [0;pi/2], [1 1 0 0 0 0])
```

```
Q = 21×2
         0    1.5708
    0.0002    1.5706
    0.0017    1.5691
    0.0055    1.5653
    0.0128    1.5580
    0.0254    1.5454
    0.0470    1.5238
    0.0857    1.4851
    0.1653    1.4055
    0.3882    1.1825
      ⋮
```

```
%Which from an overhead view gives
if plotoption==8
axis('square'); axis([-2 2 -2 2]); axis manual; hold on;
for z=1:1:length(t)
plot(cos(Q(z,1)), sin(Q(z,1)), 'o')
plot(k(z,1), k(z,2), 'o')
plot([0;cos(Q(z,1));k(z,1)],[0;sin(Q(z,1));k(z,2)])
pause
end
end
```




```
%Problem 4b,
%RR robot - trajectory generation example
hold off
```




```
axis
```

```
ans = 1×4
    -2     2    -2     2
```

```
clear all
SCURRTWOLINK
%robot definition
L1 = 1;
Link1 = link([0 0 0 0 0], 'modified');
Link2 = link([0 L1 0 0 0], 'modified');
r = robot({Link1 Link2});

plotoption=8;
%establish a time vector
t=[0:.5:10];
%define initial and final robot joint space poses
Qi=[0;pi/4]; %These are the values for the lecture example
Qf=[127*pi/128;pi/4];
%generate a trajectory plan
[q qd qdd]=jtraj(Qi, Qf, t);
%show how angles, velocities, accelerations evolve
if plotoption==1
subplot(3,1,1), plot(t, q)
subplot(3,1,2), plot(t, qd)
subplot(3,1,3), plot(t, qdd)
end
```

```
%show how robot moves in a 'straight line' in joint space
if plotoption==2
plot(q*[1;0],q*[0;1])
end
%show how the motion is not 'straight' in operational space
TJ=fkine(r, q)
```

```
TJ =
TJ(:,:,1) =

    0.7071   -0.7071         0    1.0000
    0.7071    0.7071         0         0
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,2) =

    0.7045   -0.7097         0    1.0000
    0.7097    0.7045         0    0.0036
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,3) =

    0.6880   -0.7257         0    0.9996
    0.7257    0.6880         0    0.0267
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,4) =

    0.6461   -0.7633         0    0.9966
    0.7633    0.6461         0    0.0829
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,5) =

    0.5686   -0.8226         0    0.9837
    0.8226    0.5686         0    0.1796
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,6) =

    0.4464   -0.8948         0    0.9484
    0.8948    0.4464         0    0.3171
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,7) =

    0.2735   -0.9619         0    0.8736
    0.9619    0.2735         0    0.4867
         0         0    1.0000         0
         0         0         0    1.0000
```

```
TJ(:,:,8) =

    0.0523   -0.9986         0    0.7431
    0.9986    0.0523         0    0.6691
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,9) =

   -0.2027   -0.9792         0    0.5491
    0.9792   -0.2027         0    0.8357
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,10) =

   -0.4643   -0.8857         0    0.2980
    0.8857   -0.4643         0    0.9546
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,11) =

   -0.6984   -0.7157         0    0.0123
    0.7157   -0.6984         0    0.9999
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,12) =

   -0.8740   -0.4859         0   -0.2744
    0.4859   -0.8740         0    0.9616
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,13) =

   -0.9740   -0.2266         0   -0.5285
    0.2266   -0.9740         0    0.8490
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,14) =

   -0.9996    0.0278         0   -0.7265
   -0.0278   -0.9996         0    0.6872
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,15) =

   -0.9683    0.2499         0   -0.8614
   -0.2499   -0.9683         0    0.5080
         0         0    1.0000         0
         0         0         0    1.0000
```

```
TJ(:,:,16) =

   -0.9055    0.4243         0   -0.9403
   -0.4243   -0.9055         0    0.3403
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,17) =

   -0.8363    0.5483         0   -0.9790
   -0.5483   -0.8363         0    0.2036
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,18) =

   -0.7789    0.6272         0   -0.9942
   -0.6272   -0.7789         0    0.1073
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,19) =

   -0.7424    0.6700         0   -0.9987
   -0.6700   -0.7424         0    0.0512
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,20) =

   -0.7267    0.6869         0   -0.9996
   -0.6869   -0.7267         0    0.0281
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,21) =

   -0.7242    0.6895         0   -0.9997
   -0.6895   -0.7242         0    0.0245
         0         0    1.0000         0
         0         0         0    1.0000
```

```
Ree=transl(TJ);
if plotoption==3
subplot(2,1,1), plot(t,Ree(:,1))
subplot(2,1,2), plot(t,Ree(:,2))
end
if plotoption==4
plot(r, q, 'loop')
end
if plotoption==5
axis('square'); axis([-2 2 -2 2]); axis manual; hold on;
for z=1:1:length(t)
plot(cos(q(z,1)), sin(q(z,1)), 'o')
```

```
plot(Ree(z,1), Ree(z,2), 'o')
plot([0;cos(q(z,1));Ree(z,1)],[0;sin(q(z,1));Ree(z,2)])
end
end
% Now let's try doing the planning in Cartesian Space
Tinit=fkine(r,Qi);
Tfinal=fkine(r, Qf);
rr=jtraj(0,1,t);
TC = ctraj(Tinit, Tfinal, rr);
%Endpoint x and y coordinates over time
if plotoption==6
plot(t, transl(TC)); grid;
end
%Endpoint in Cartesian space
k=transl(TC);
if plotoption==7
axis('square'); axis([-2 2 -2 2]); grid
plot(k(:,1),k(:,2))
end
%The resulting joint angles
Q=ikine(r, TC, [0;pi/2], [1 1 0 0 0 0])
```

```
i=11, nm=0.000000
Error using ikine
Solution wouldn't converge
```

```
%Which from an overhead view gives
if plotoption==8
axis('square'); axis([-2 2 -2 2]); axis manual; hold on;
for z=1:1:length(t)
plot(cos(Q(z,1)), sin(Q(z,1)), 'o')
plot(k(z,1), k(z,2), 'o')
plot([0;cos(Q(z,1));k(z,1)],[0;sin(Q(z,1));k(z,2)])
pause
end
end
```

```
%Problem 4b
%RR robot - trajectory generation example
hold off
axis
```

```
ans = 1×4
    -2     2    -2     2
```

```
clear all
SCURRTWOLINK
%robot definition
L1 = 1;
Link1 = link([0 0 0 0 0], 'modified');
Link2 = link([0 L1 0 0 0], 'modified');
```

```
r = robot({Link1 Link2});

plotoption=5;
%establish a time vector
t=[0:.5:10];
%define initial and final robot joint space poses
Qi=[3*pi/8;-pi/2]; %These are the values for the lecture example
Qf=[5*pi/8;pi/2];
%generate a trajectory plan
[q qd qdd]=jtraj(Qi, Qf, t);
%show how angles, velocities, accelerations evolve
if plotoption==1
subplot(3,1,1), plot(t, q)
subplot(3,1,2), plot(t, qd)
subplot(3,1,3), plot(t, qdd)
end
%show how robot moves in a 'straight line' in joint space
if plotoption==2
plot(q*[1;0],q*[0;1])
end
%show how the motion is not 'straight' in operational space
TJ=fkine(r, q)
```

```
TJ =
TJ(:,:,1) =

    0.9239    0.3827         0    0.3827
   -0.3827    0.9239         0    0.9239
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,2) =

    0.9256    0.3785         0    0.3818
   -0.3785    0.9256         0    0.9242
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,3) =

    0.9362    0.3514         0    0.3765
   -0.3514    0.9362         0    0.9264
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,4) =

    0.9588    0.2842         0    0.3633
   -0.2842    0.9588         0    0.9317
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,5) =
```

```
    0.9864    0.1645         0    0.3403
   -0.1645    0.9864         0    0.9403
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,6) =

    0.9999   -0.0138         0    0.3064
    0.0138    0.9999         0    0.9519
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,7) =

    0.9695   -0.2452         0    0.2615
    0.2452    0.9695         0    0.9652
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,8) =

    0.8624   -0.5062         0    0.2065
    0.5062    0.8624         0    0.9784
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,9) =

    0.6571   -0.7538         0    0.1429
    0.7538    0.6571         0    0.9897
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,10) =

    0.3576   -0.9339         0    0.0731
    0.9339    0.3576         0    0.9973
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,11) =

    0.0000   -1.0000         0    0.0000
    1.0000    0.0000         0    1.0000
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,12) =

   -0.3576   -0.9339         0   -0.0731
    0.9339   -0.3576         0    0.9973
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,13) =
```

```
   -0.6571   -0.7538         0   -0.1429
    0.7538   -0.6571         0    0.9897
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,14) =

   -0.8624   -0.5062         0   -0.2065
    0.5062   -0.8624         0    0.9784
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,15) =

   -0.9695   -0.2452         0   -0.2615
    0.2452   -0.9695         0    0.9652
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,16) =

   -0.9999   -0.0138         0   -0.3064
    0.0138   -0.9999         0    0.9519
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,17) =

   -0.9864    0.1645         0   -0.3403
   -0.1645   -0.9864         0    0.9403
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,18) =

   -0.9588    0.2842         0   -0.3633
   -0.2842   -0.9588         0    0.9317
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,19) =

   -0.9362    0.3514         0   -0.3765
   -0.3514   -0.9362         0    0.9264
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,20) =

   -0.9256    0.3785         0   -0.3818
   -0.3785   -0.9256         0    0.9242
         0         0    1.0000         0
         0         0         0    1.0000


TJ(:,:,21) =
```

```
   -0.9239    0.3827         0   -0.3827
   -0.3827   -0.9239         0    0.9239
         0         0    1.0000         0
         0         0         0    1.0000
```

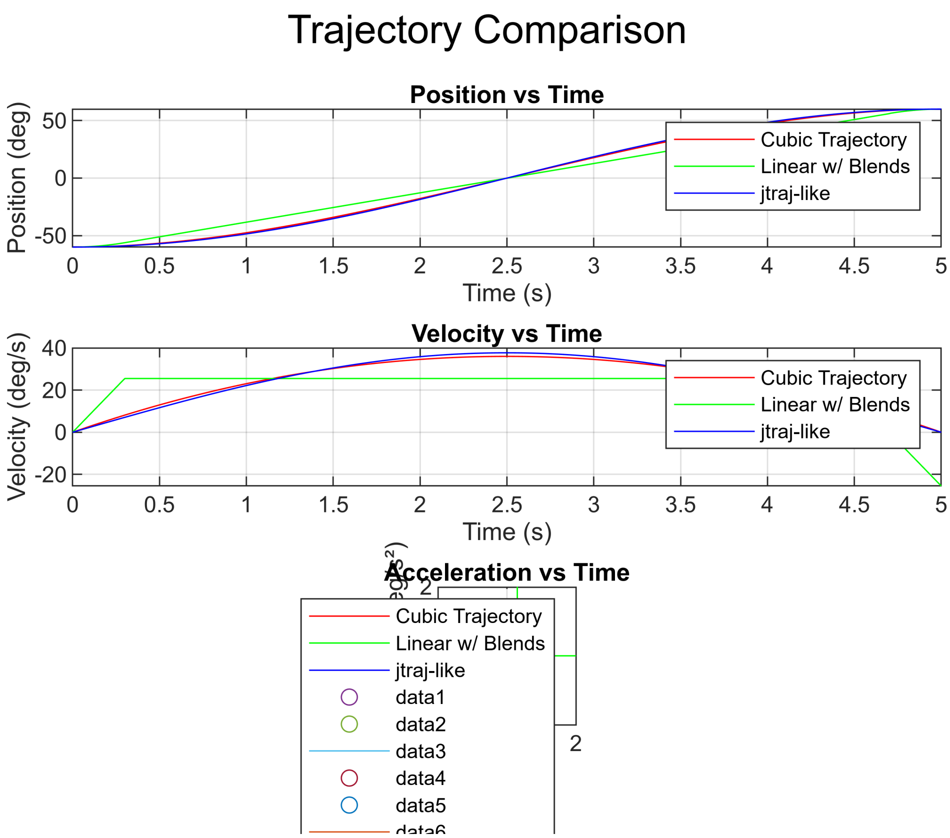
```
Ree=transl(TJ);
if plotoption==3
subplot(2,1,1), plot(t,Ree(:,1))
subplot(2,1,2), plot(t,Ree(:,2))
end
if plotoption==4
plot(r, q, 'loop')
end
if plotoption==5
axis('square'); axis([-2 2 -2 2]); axis manual; hold on;
for z=1:1:length(t)
plot(cos(q(z,1)), sin(q(z,1)), 'o')
plot(Ree(z,1), Ree(z,2), 'o')
plot([0;cos(q(z,1));Ree(z,1)],[0;sin(q(z,1));Ree(z,2)])
end
end
% Now let's try doing the planning in Cartesian Space
Tinit=fkine(r,Qi);
Tfinal=fkine(r, Qf);
rr=jtraj(0,1,t);
TC = ctraj(Tinit, Tfinal, rr);
%Endpoint x and y coordinates over time
if plotoption==6
plot(t, transl(TC)); grid;
end
%Endpoint in Cartesian space
k=transl(TC);
if plotoption==7
axis('square'); axis([-2 2 -2 2]); grid
plot(k(:,1),k(:,2))
end
%The resulting joint angles
Q=ikine(r, TC, [0;pi/2], [1 1 0 0 0 0])
```

```
Q = 21×2
    1.1781    0.3927
    1.1789    0.3919
    1.1842    0.3866
    1.1971    0.3737
    1.2197    0.3511
    1.2534    0.3174
    1.2986    0.2722
    1.3548    0.2160
    1.4207    0.1501
    1.4938    0.0770
      ⋮
```

```
%Which from an overhead view gives
```

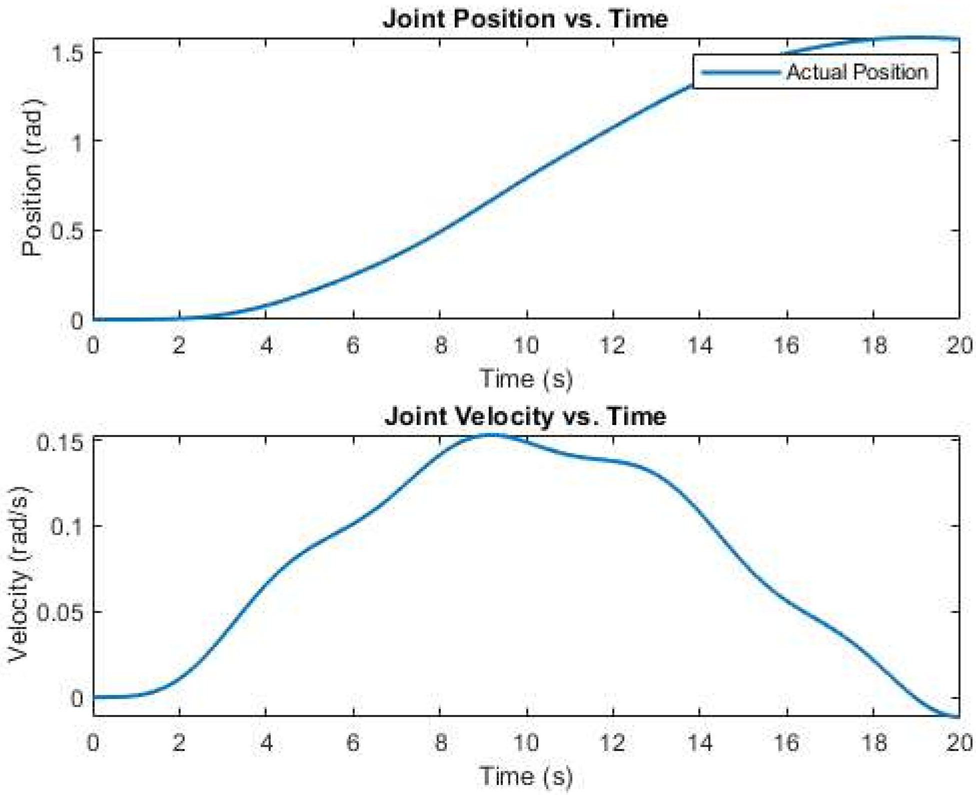
```
plotoption = 8;
if plotoption==8
axis('square'); axis([-2 2 -2 2]); axis manual; hold on;
for z=1:1:length(t)
plot(cos(Q(z,1)), sin(Q(z,1)), 'o')
plot(k(z,1), k(z,2), 'o')
plot([0;cos(Q(z,1));k(z,1)],[0;sin(Q(z,1));k(z,2)])
pause
end
end
```

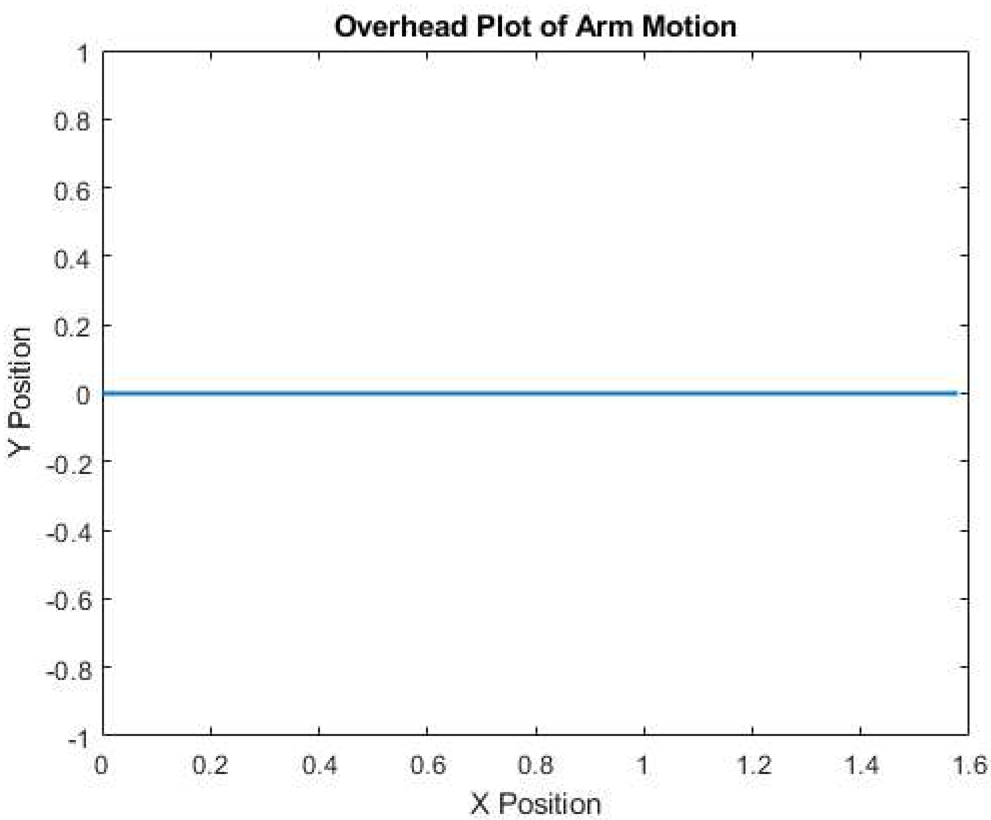
```
Warning: Limiting legend entries to 50. Specify a vector of graphics objects to display more than 50 entries.
```



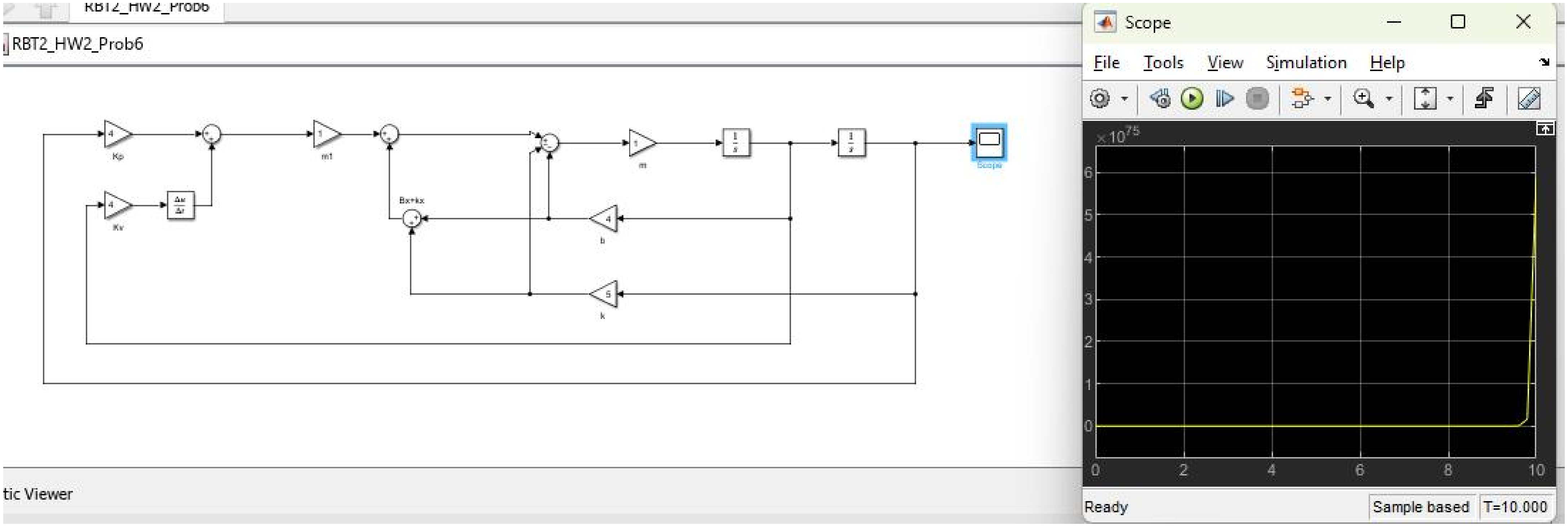
```
clc;
close all;
clear


t_end = 20;
n_steps = 2000;
t = linspace(0, t_end, n_steps);
dt = t(2) - t(1);


q0 = 0;
qf = pi/2;
[qd_traj, qd_dot, qd_ddot] = jtraj(q0, qf, t);

q = 0;                          % Initial position
qd = 0;                         % Initial velocity
kp = 2;                         % Proportional gain
tau = 0;                        % Initial torque
I = 1;                          % Moment of inertia

% Arrays to store results
position = zeros(1, n_steps);
velocity = zeros(1, n_steps);

for i = 1:n_steps
    % Compute the position error
    error = qd_traj(i) - q;

    % Compute the control torque using the P control law
    tau = kp * error;

    % Update the dynamics
    qdd = tau / I;              % Angular acceleration
    qd = qd + qdd * dt;         % Update velocity
    q = q + qd * dt;            % Update position

    position(i) = q;
    velocity(i) = qd;
end

figure;
subplot(2,1,1);
plot(t, position, 'LineWidth', 1.5);
hold on;
xlabel('Time (s)');
ylabel('Position (rad)');
title('Joint Position vs. Time');
legend('Actual Position')

subplot(2,1,2);
plot(t, velocity, 'LineWidth', 1.5);
xlabel('Time (s)');
ylabel('Velocity (rad/s)');
title('Joint Velocity vs. Time');
```

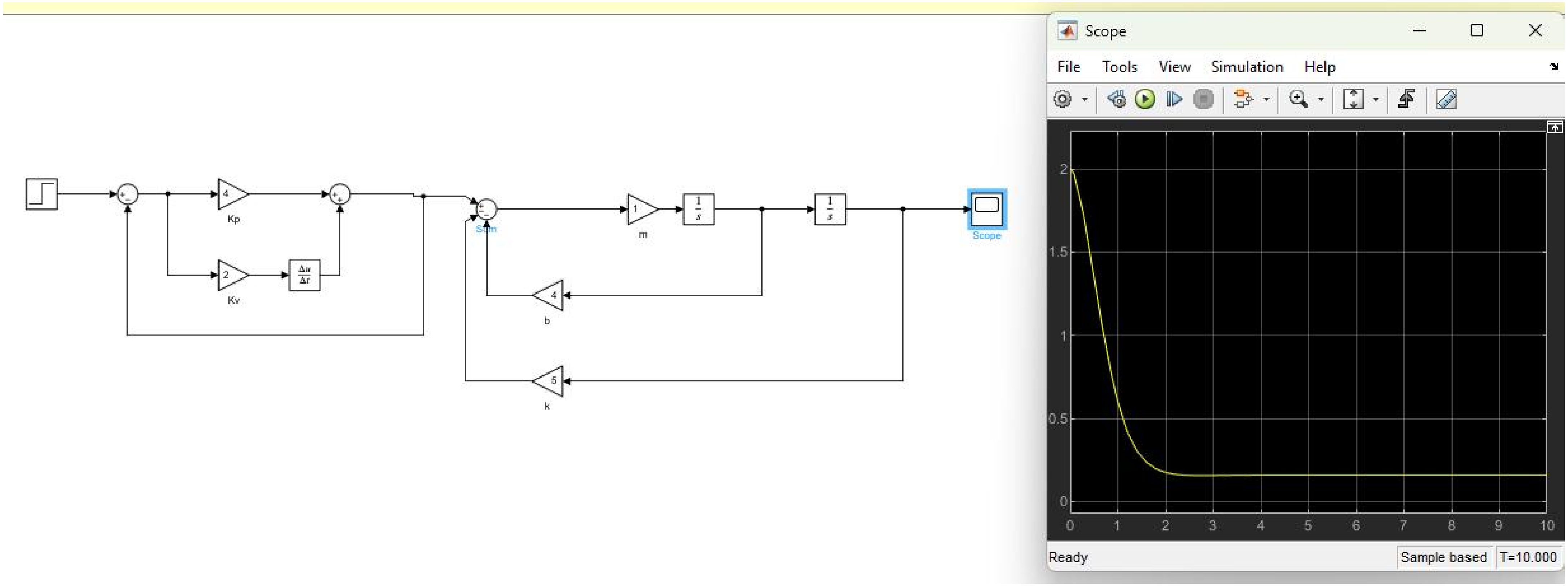
```
% Overhead plot of arm motion
figure;
plot(position, zeros(size(position)), 'LineWidth', 1.5);
xlabel('X Position');
ylabel('Y Position');
title('Overhead Plot of Arm Motion');
```

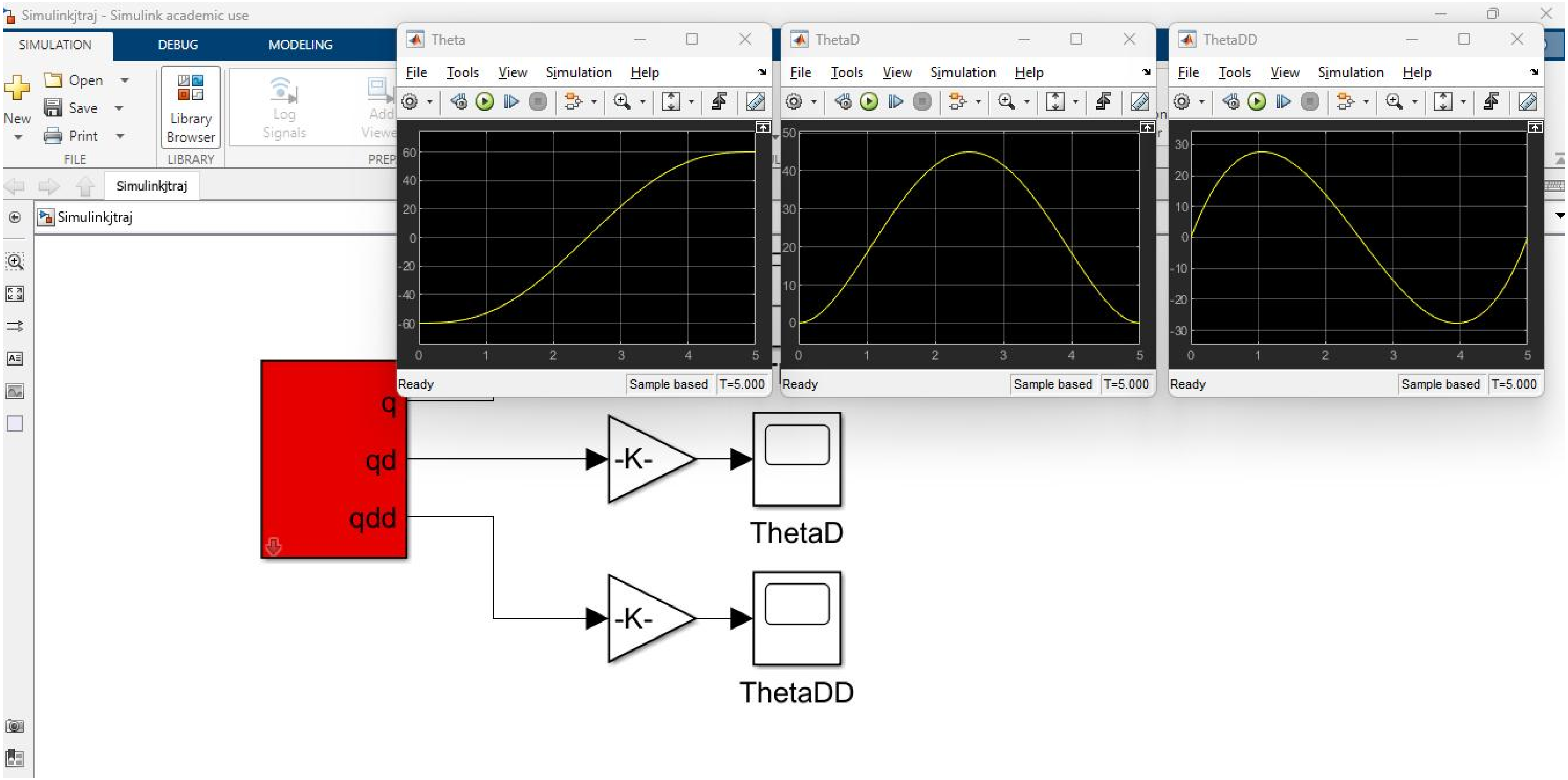




Published with MATLAB® R2023b


RBT2_HW2_Prob6
RBT2_HW2_Prob6
Kp
m1
Kv
Bx+kx
m
b
k
tic Viewer
Scope
File
Tools
View
Simulation
Help
×10^75
0
1
2
3
4
5
6
0
2
4
6
8
10
Ready
Sample based
T=10.000


Kp
Kv
Sum
m
b
k
Scope
Scope
File
Tools
View
Simulation
Help
2
1.5
1
0.5
0
0
1
2
3
4
5
6
7
8
9
10
Ready
Sample based
T=10.000


Simulinkjtraj - Simulink academic use
SIMULATION
DEBUG
MODELING
Open
Save
Print
New
FILE
Library Browser
LIBRARY
Log Signals
Simulinkjtraj
Theta
ThetaD
ThetaDD
File Tools View Simulation Help
Ready
Sample based
T=5.000
q
qd
qdd
-K-
ThetaD
ThetaDD