BIRZEIT UNIVERSITY

Signals and Systems (ENEE 2312)

MATLAB Assignment

Student Name: Mohammad Salem

Student No#: 1200651

Instructor’s Name: Dr. Ashraf Al-Rimawi

Sections: 3

Date: 20 JUN 2023

Question 1:

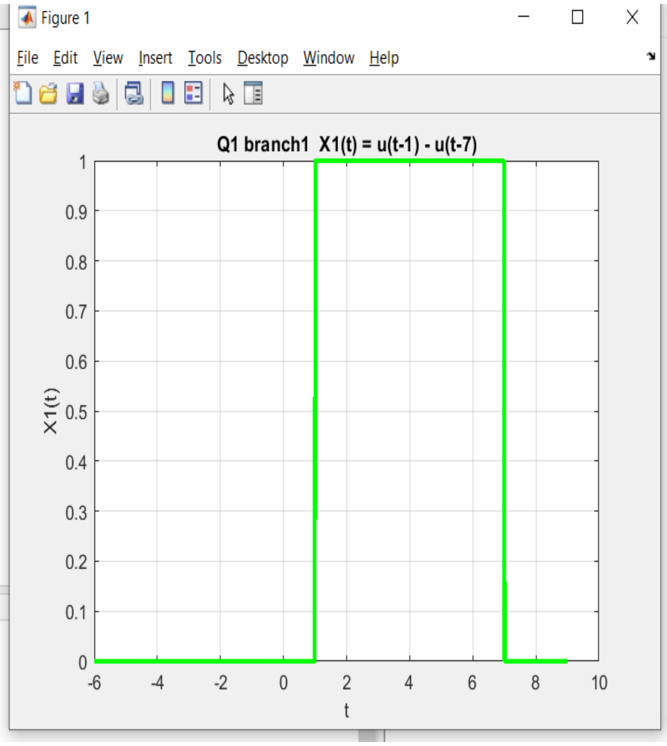
Q1_1:

test.m plot2.m Q1.m

```
t = -6:0.01:9;

X1 = heaviside(t-1) - heaviside(t-7);

% Plotting
%figure;
plot(t, X1, 'g', 'LineWidth', 2.5);
xlabel('t');
ylabel('X1(t)');
title('Q1 branch1  X1(t) = u(t-1) - u(t-7)');
grid on;
```

Command Window

```
>> Q1
>> Q1
>> Q1
>> Q1
```



Q1_2:

```
%Mohammad Salme 1200651

% Q 1 branch 2
t = -10:0.01:10; % Define the time range (adjust as needed)
pulse = 5 * rectpuls(t, 16); % Compute the pulse signal

% Plotting

plot(t, pulse, 'LineWidth', 2);

xlabel('Time');

ylabel('Amplitude');

title('Finite Pulse - n(t)');

grid on;
```

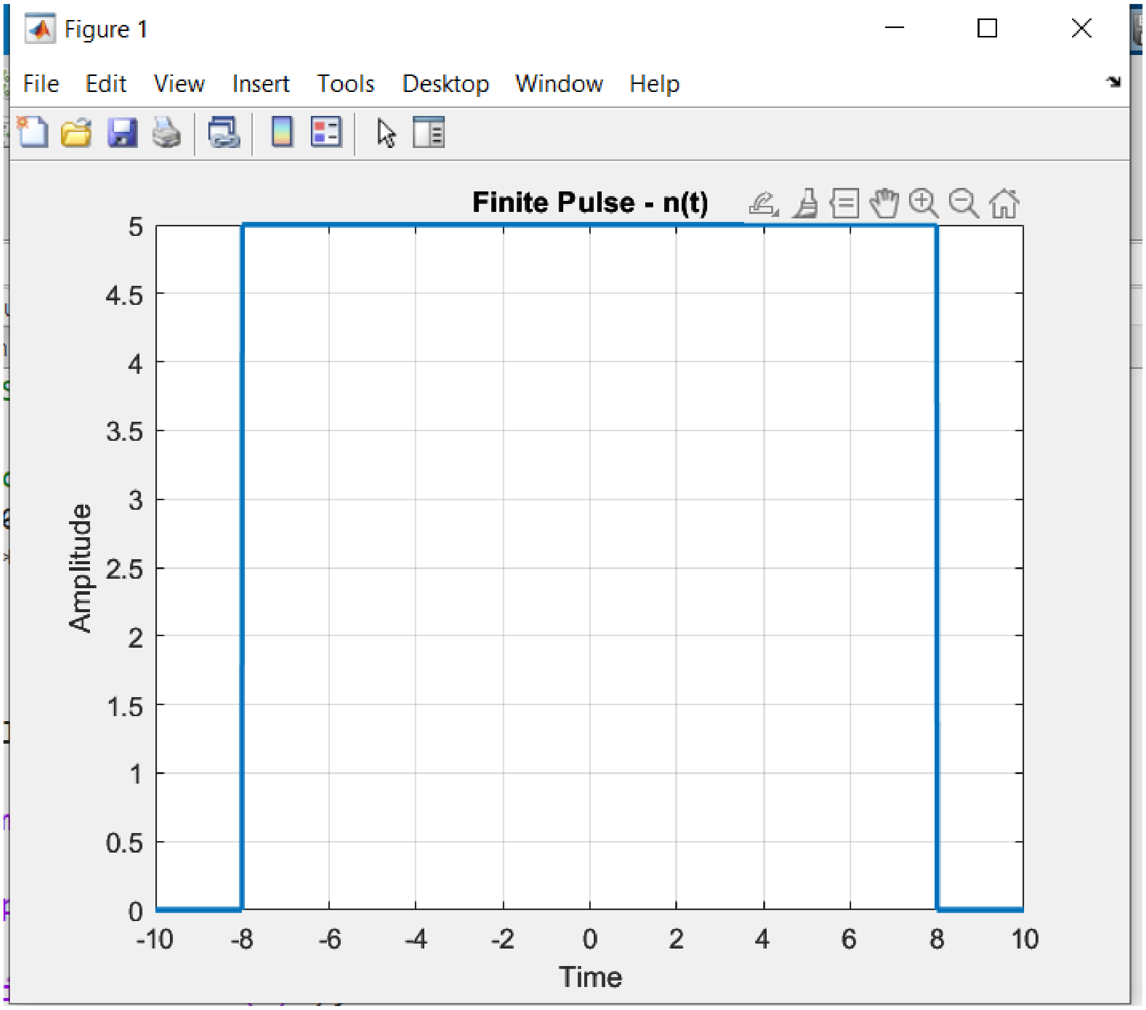
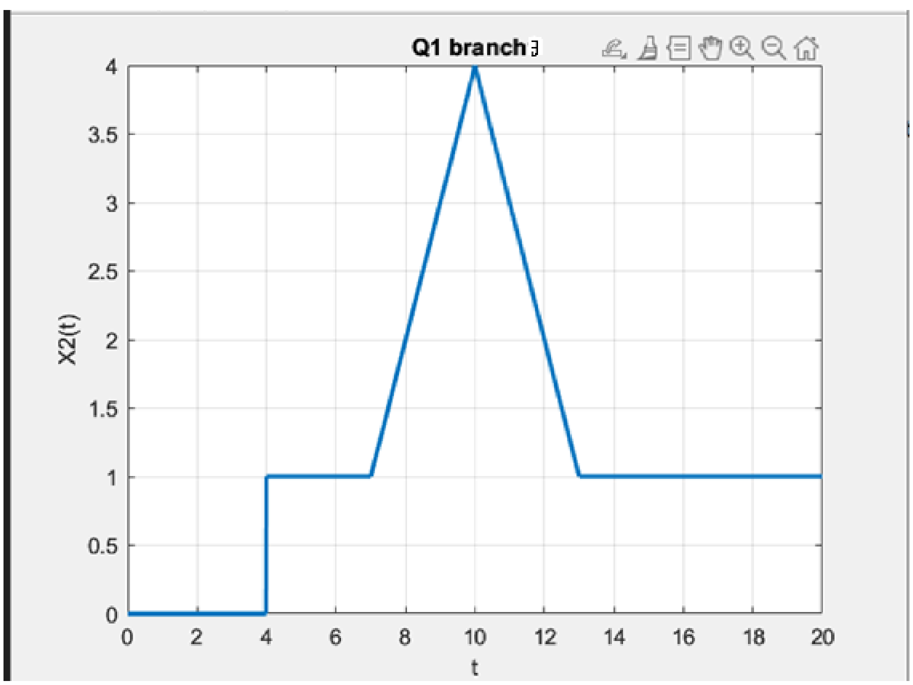


Figure 1
File Edit View Insert Tools Desktop Window Help
Finite Pulse - n(t)
Amplitude
Time
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
-10 -8 -6 -4 -2 0 2 4 6 8 10

Q1_3:

```
% Q1 branch 3
% Mohammad Salem

t = 0:0.01:20;
X2 = heaviside(t-4) + (t-7).*(t>=7) - 2*(t-10).*(t>=10) + (t-13).*(t>=13);

plot(t, X2, 'LineWidth', 2);
xlabel('t');
ylabel('X2(t)');
title('Q1 branch1');
grid on;
%axis([0 20 -1 3]);
```



Question 2:

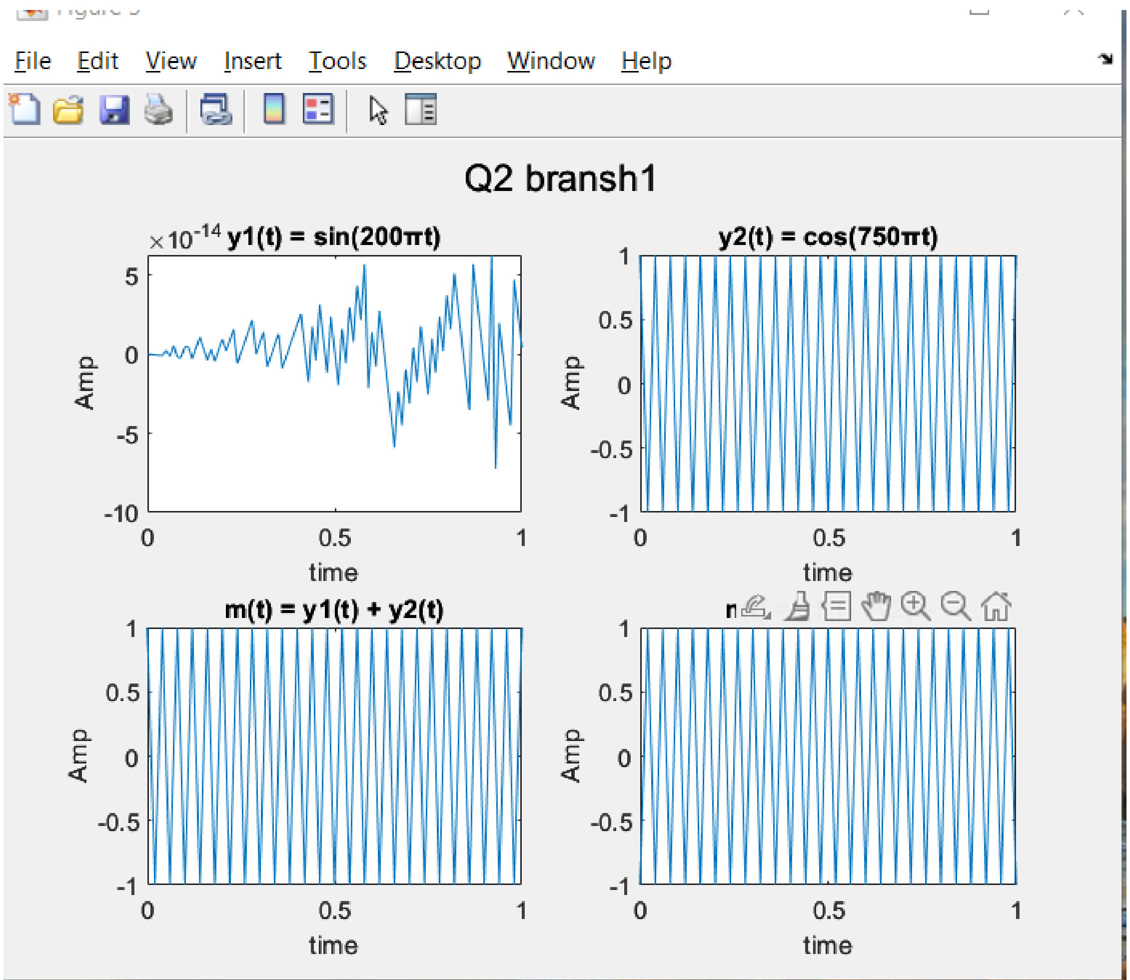
Q2_1:

test.m plot2.m Q1.m Q2.m

```
% Q2_1
% 1. Generate and plot the signals y1(t)= sin 200π(t),
% y2(t)= cos750πt, then determine y1
% and plot the signals m(t)= y1+y2 and n(t)= y1-y2

t = 0:0.01:1; % Time vector with Scale 0.01
y1 = sin(200*pi*t); % Signal y1(t) = sin(200πt)
y2 = cos(750*pi*t); % Signal y2(t) = cos(750πt)

m = y1 + y2; % Signal m(t) = y1(t) + y2(t)
n = y1 - y2; % Signal n(t) = y1(t) - y2(t)

% Plotting y1(t)
subplot(2, 2, 1);
plot(t, y1);
title('y1(t) = sin(200πt)');
xlabel('time');
ylabel('Amp');

% Plotting y2(t)
subplot(2, 2, 2);
plot(t, y2);
```

test.m | plot2.m | Q1.m | Q2.m

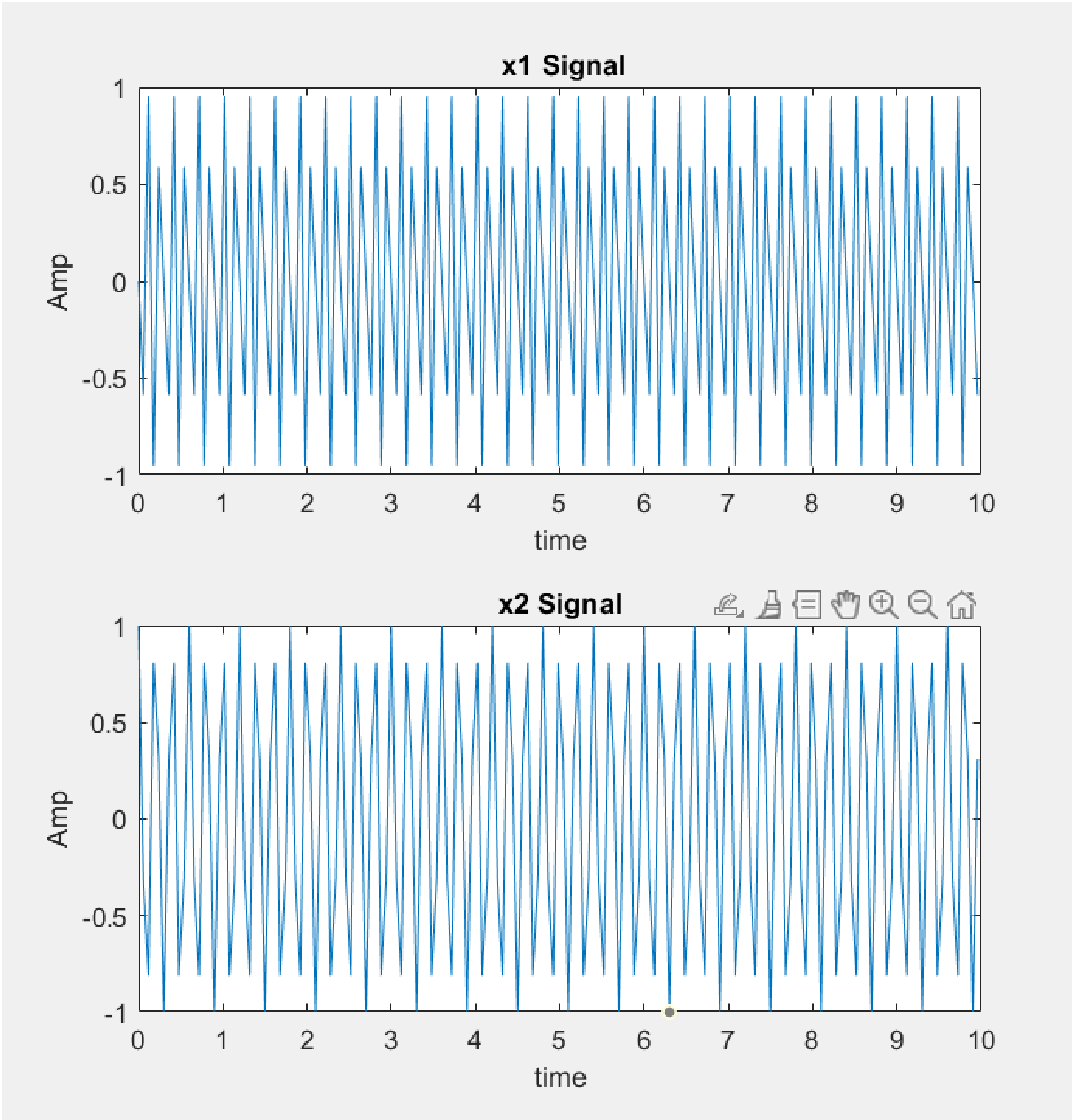
```
xlabel('time');
ylabel('Amp');

% Plotting y2(t)
subplot(2, 2, 2);
plot(t, y2);
title('y2(t) = cos(750πt)');
xlabel('time');
ylabel('Amp');

% Plotting m(t)
subplot(2, 2, 3);
plot(t, m);
title('m(t) = y1(t) + y2(t)');
xlabel('time');
ylabel('Amp');

% Plotting n(t)
subplot(2, 2, 4);
plot(t, n);
title('n(t) = y1(t) - y2(t)');
xlabel('time');
ylabel('Amp');

% Adjusting the subplot layout
sgtitle('Q2 bransh1');
```


File Edit View Insert Tools Desktop Window Help
Q2 bransh1
×10^-14 y1(t) = sin(200πt)
y2(t) = cos(750πt)
m(t) = y1(t) + y2(t)
Amp
time

Q2_2:

test.m Q1.m plot2.m Q2.m

```
% Q2 branch2 analyze the signal's plot and apply mathematical concepts.
t = 0:0.06:10;   % time vector with step size of 0.01
fs = 1 / (t(2) - t(1));   % Calculate the sampling rate
x1 = sin(2*pi*10*t);    % sin with a frequency of 10 Hz
x2 = cos(2*pi*5*t);     % cos wave with a frequency of 5 Hz
figure;
subplot(2,1,1);
plot(t,x1);
title(' x1 Signal');
xlabel('time');
ylabel('Amp');

subplot(2,1,2);
plot(t,x2);
title('x2 Signal');
xlabel('time');
ylabel('Amp');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Assuming the signal is periodic and has one period
% within the plotted time range
[peaks,locs] = findpeaks(x1);   % Find peak values and their locations
period_time = t(locs(2)) - t(locs(1));   % Calculate the time difference between two peaks
fundamental_frequency = 1 / period_time;   % Calculate the fundamental frequency
```


x1 Signal
Amp
time
x2 Signal
Amp
time

Q3_1:

test.m | plot2.m | Q1.m | Q2.m | Q3.m

```
  % Q3 branch 1 5dy(t)dt + 20y(t)=15   , using zero initial conditions
  % Define the differential equation
syms y(t)
equation = 5*diff(y(t), t) + 20*y(t) == 15;

% Solve the differential equation
sol = dsolve(equation, y(0) == 0);

% Display the solution
disp("The solution for the differential equation is:")
disp(sol)

% Plot the solution
fplot(sol, [0, 10])
xlabel('t')
ylabel('y(t)')
title('Q3 branch1')
```

```
>> Q3
The solution for the differential equation is:
3/4 - (3*exp(-4*t))/4

fx >>
```

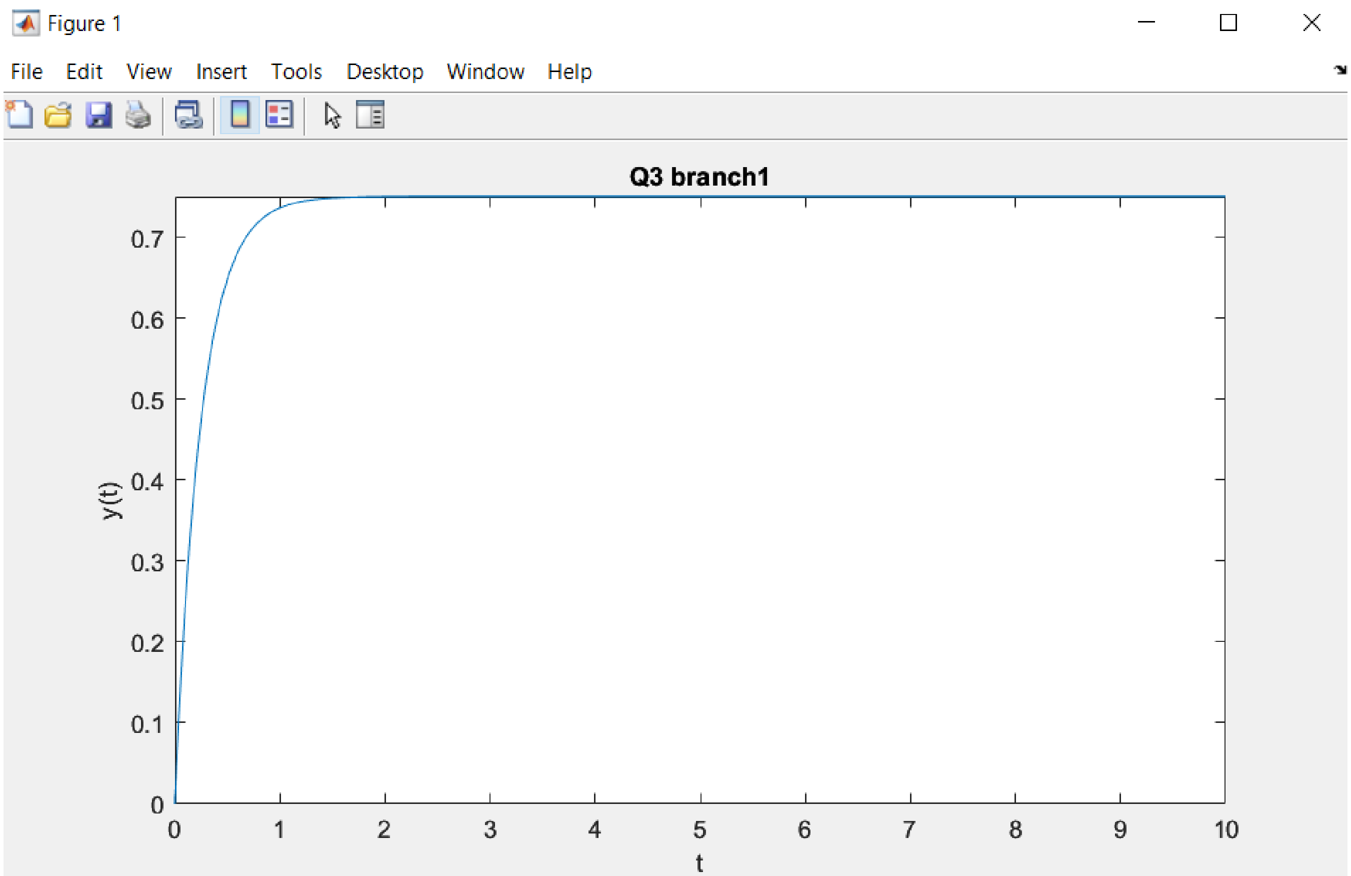


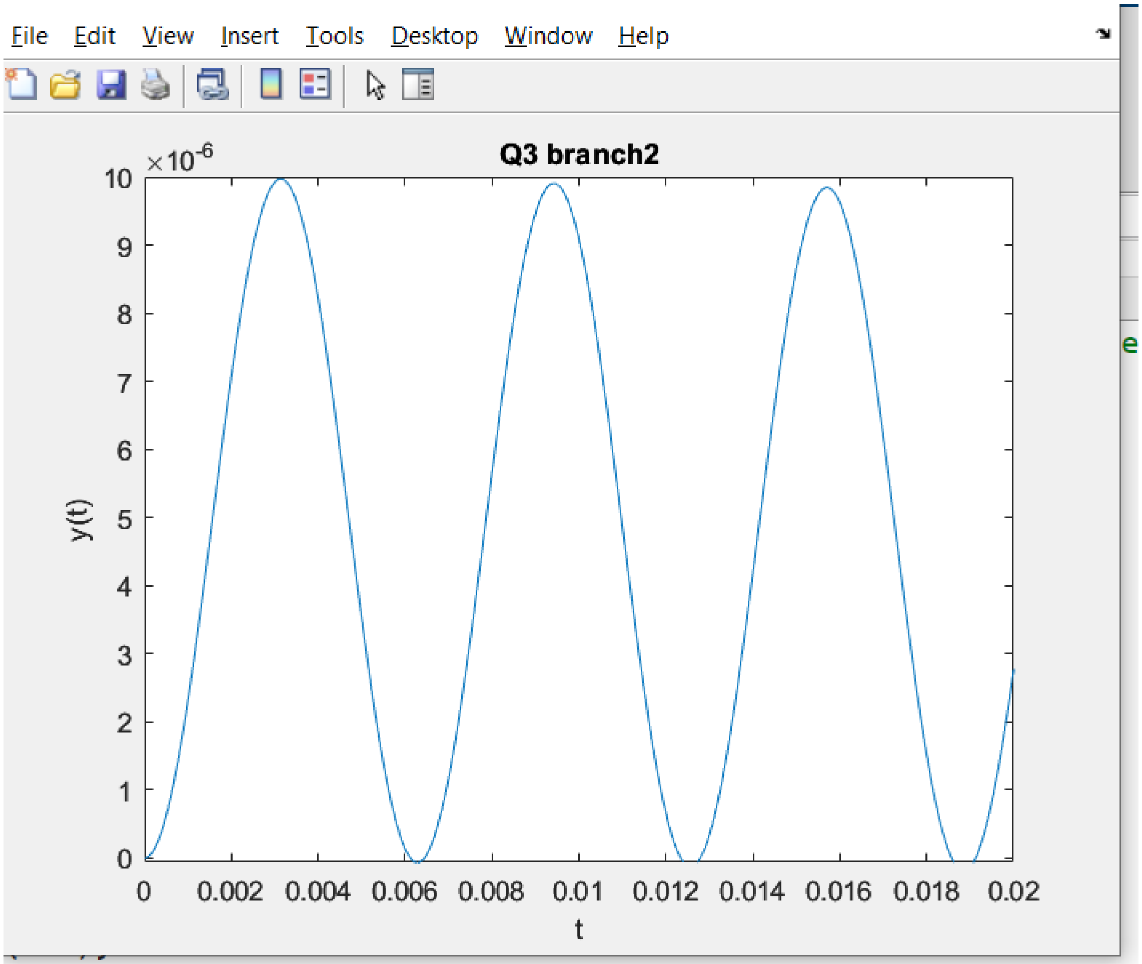
Figure 1
File Edit View Insert Tools Desktop Window Help
Q3 branch1
y(t)
t
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0 1 2 3 4 5 6 7 8 9 10

Q3_2:

test.m plot2.m Q1.m Q2.m Q3.m

```matlab
% Q3 branch 2   , d2y(t)dt2+2dydt +4y(t)=5 cos1000t   , using zero initial conditions

 % Define the differential equation
syms y(t)
Dy = diff(y,t);
DE = diff(y, t, 2) + 2*diff(y, t) + 4*y == 5*cos(1000*t);

con1 = y(0)==0;
con2 = Dy(0)==0;

sol = dsolve(DE,con1,con2);
%simple_sol = simplify(sol);

% Display the solution
disp(sol);

% Plot
fplot(sol, [0, 0.02]);
xlabel('t');
ylabel('y(t)');
title('Q3 branch2');
```

```
% Q3 branch2 the solution for diffrantioal equasion
sin(3^(1/2)*t)*
((625*cos(1000*t - 3^(1/2)*t))/
124999500002
- (625*cos(1000*t + 3^(1/2)*t))/
124999500002 -
(1249995*sin(1000*t + 3^(1/2)*t))/
499998000008 +
 (1249995*sin(1000*t - 3^(1/2)*t))/
499998000008 +
 (1250005*3^(1/2)*cos(1000*t + 3^(1/2)*t))/
1499994000024 + (1250005*3^(1/2)*cos(1000*t - 3^(1/2)*t))/
1499994000024 + (312499375*3^(1/2)*sin(1000*t + 3^(1/2)*t))/
374998500006 + (312499375*3^(1/2)*sin(1000*t - 3^(1/2)*t))/
374998500006) - (5*3^(1/2)*cos(3^(1/2)*t)*((sin(t*(3^(1/2) - 1000)) -
 cos(t*(3^(1/2) - 1000))*(3^(1/2) - 1000))/
((3^(1/2) - 1000)^2 + 1) + (sin(t*(3^(1/2) + 1000)) - cos(t*(3^(1/2) + 1000))*(3^(1/2) + 1000))/
((3^(1/2) + 1000)^2 + 1)))/6 - (1250005*3^(1/2)*exp(-t)*sin(3^(1/2)*t))/
749997000012 - (1249995*exp(-t)*cos(3^(1/2)*t))/
(4*(500*3^(1/2) - 250001)*(500*3^(1/2) + 250001))
```


File Edit View Insert Tools Desktop Window Help
Q3 branch2
×10^-6
10
9
8
7
6
5
4
3
2
1
0
y(t)
0 0.002 0.004 0.006 0.008 0.01 0.012 0.014 0.016 0.018 0.02
t

Q4_1:

```
% Q4 branch 1
% Mohammad Salem

syms t y(t)
dy(t) = diff(y(t),t)
ic = y(0)==2     % ic : initial_condition

func = dy(t)+2*y==7*heaviside(t)

sol = dsolve(func,ic)

simple_sol = simplify(sol)
```

Command Window

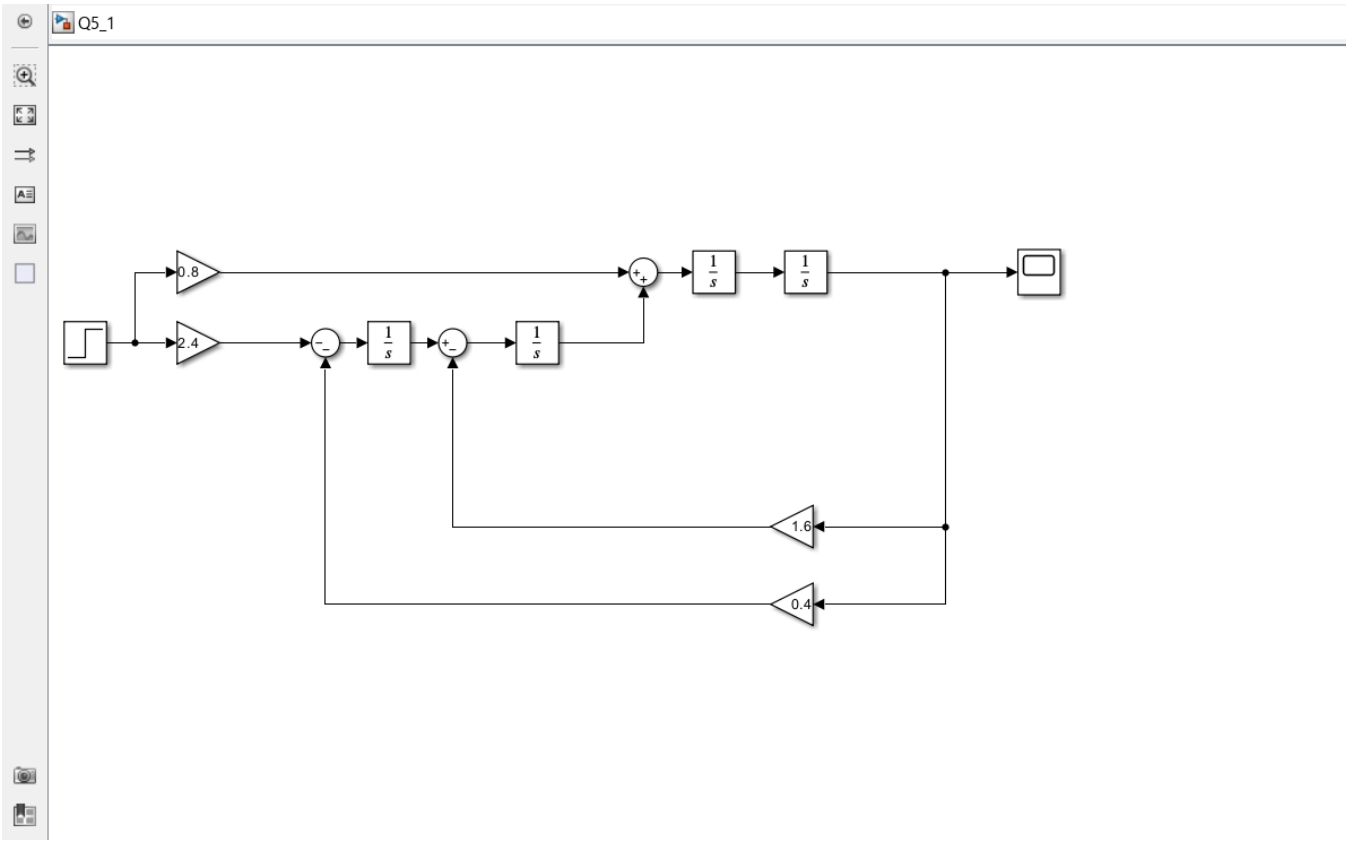
```
2*y(t) + diff(y(t), t) == 7*heaviside(t)


sol =

exp(-2*t)/4 - exp(-2*t)*((7*sign(t))/4 - (7*exp(2*t)*(sign(t) + 1))/4)


simple_sol =

exp(-2*t)/4 + (7*sign(t))/4 - (7*exp(-2*t)*sign(t))/4 + 7/4
```

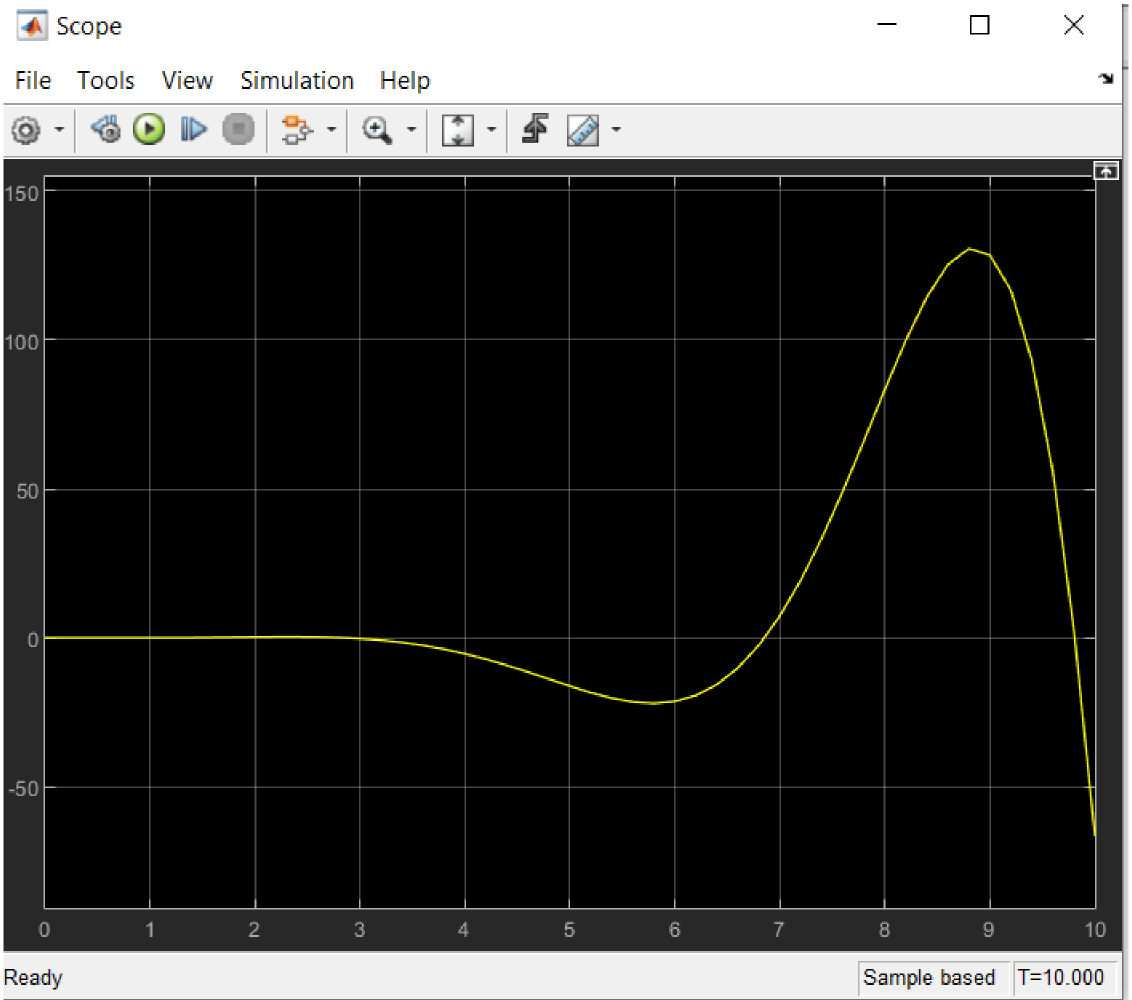
Q4_2:

```
% Q4 branch 2
% Mohammad Salem

 syms t,y(t)
 fun  = diff(y(t),t,2)+4*diff(y(t),t)+5*y(t)==5*cos(2000*t)

 ic1 = y(0) ==1    % ic1 : the first initial_condition
 dy = diff(y,t)

 ic2 =dy(0) ==2   % ic2 : the secand initial_condition
 cond = [ic1,ic2]

 sol = dsolve(fun,cond)

 simple_sol = simplify(sol)
```

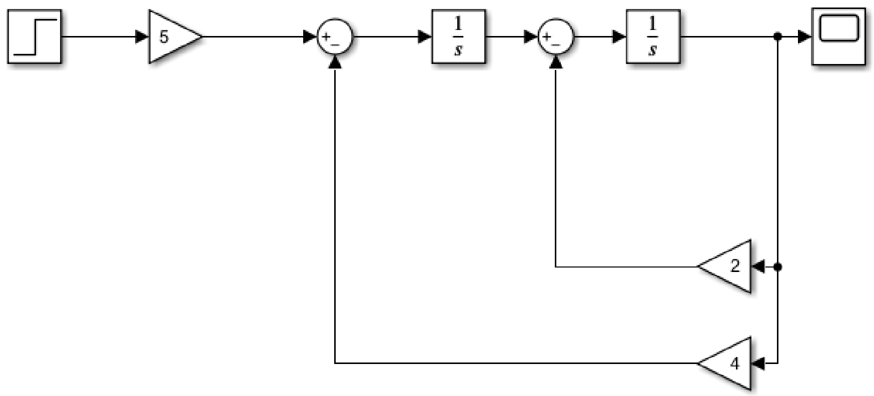
```
simple_sol =

sin(t)*(cos(1999*t)/799201 + cos(2001*t)/800801 + (1999*sin(1999*t))/
1598402 + (2001*sin(2001*t))/1601602) - cos(t)*((1999*cos(1999*t))/
1598402 - (2001*cos(2001*t))/
1601602 - sin(1999*t)/
799201 + sin(2001*t)/800801) + (640001760000*exp(-2*t)*cos(t))/
640000960001 + (2560002240002*exp(-2*t)*sin(t))/640000960001

>>
```

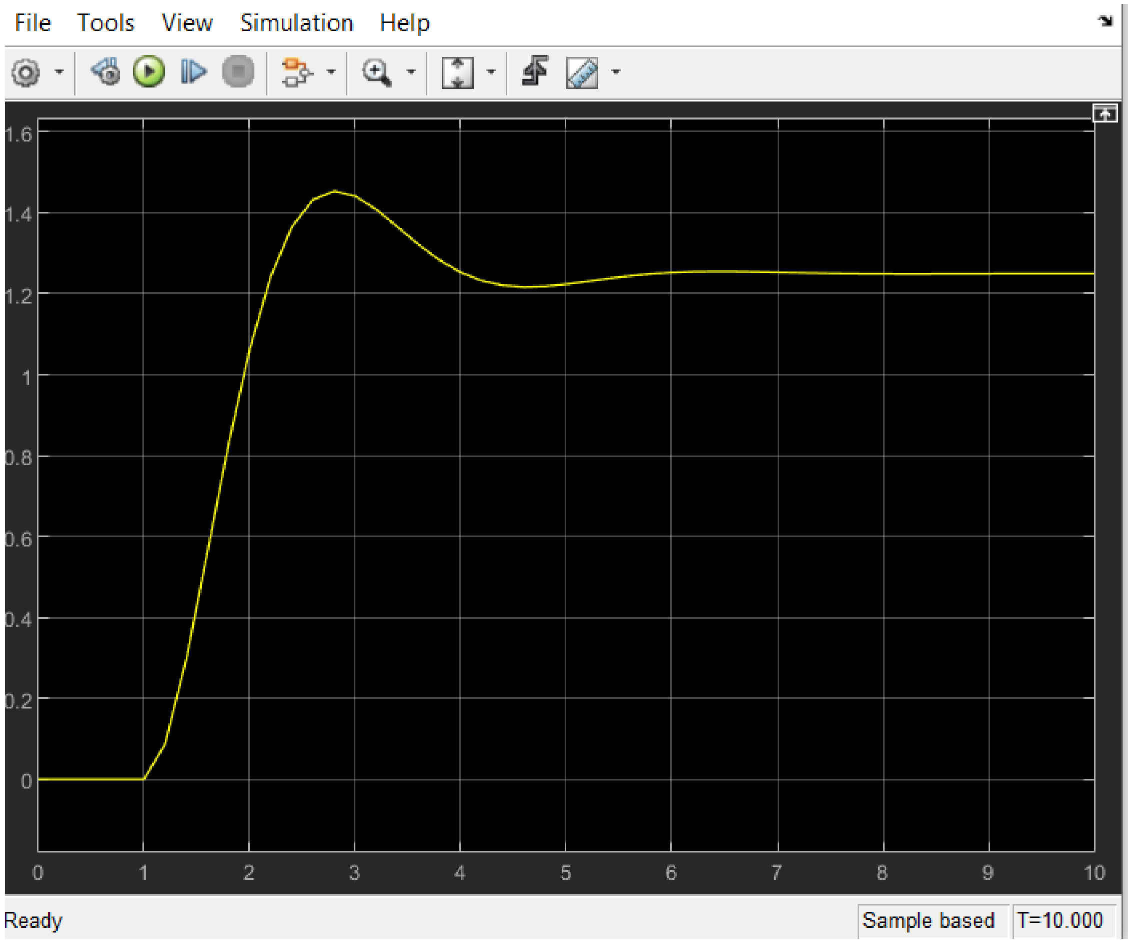
Q5_1:




Scope
File Tools View Simulation Help
150
100
50
0
-50
0 1 2 3 4 5 6 7 8 9 10
Ready
Sample based
T=10.000

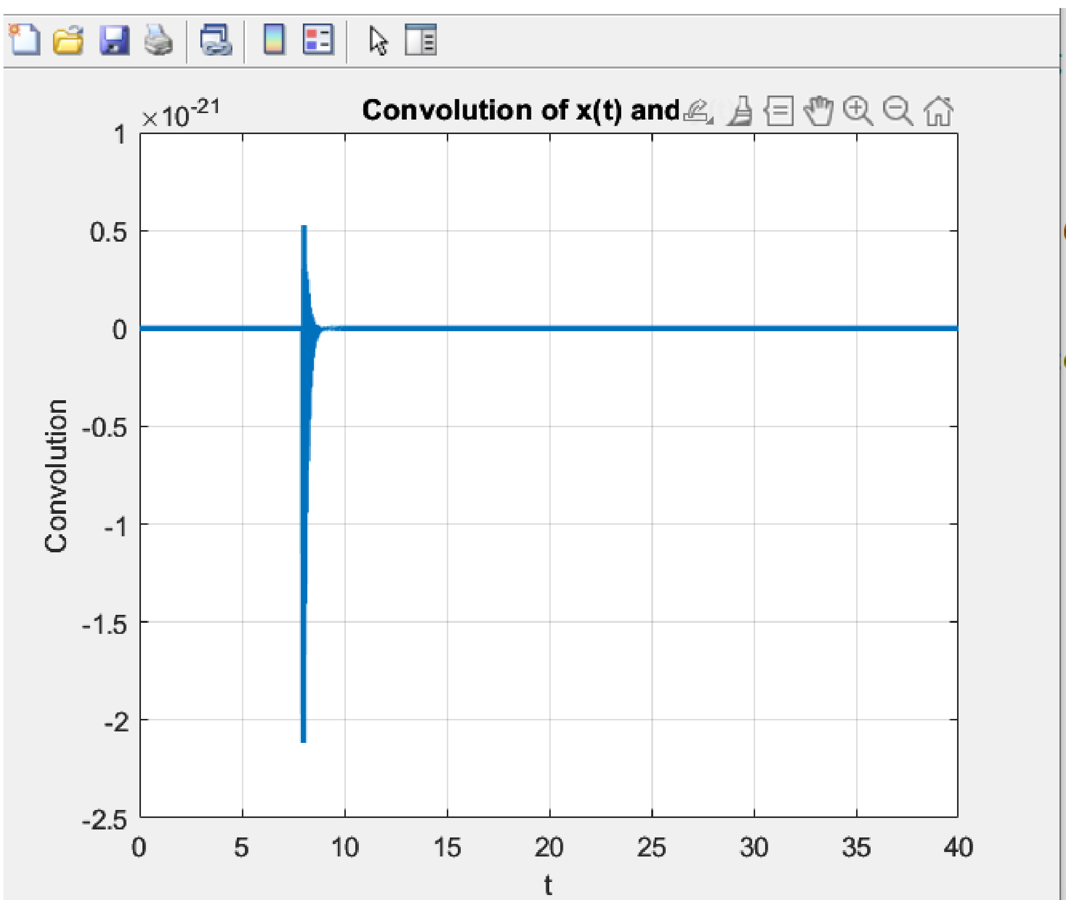
Q5_2:




File Tools View Simulation Help
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0 1 2 3 4 5 6 7 8 9 10
Ready
Sample based
T=10.000

Q6:

```
%Q6
%Mohammad salem
t = 0:0.01:20;   % Define the range of t values for plotting

% Define the functions x(t) and y(t)
x = 10 * exp(-6 * t) .* heaviside((t - 2) / 4);
y = 10 * exp(-6 * t) .* cos(100 * t) .* heaviside((t - 6) / 6);

% Compute the convolution of x(t) and y(t)
convolution = conv(x, y) * 0.01;   % Multiply by the time step to approximate the integral

% Define the range of t values for the convolution result
t_conv = 0:0.01:(length(convolution)-1)*0.01;

% Plot the convolution
plot(t_conv, convolution, 'LineWidth', 2);
xlabel('t');
ylabel('Convolution');
title('Convolution of x(t) and y(t)');
grid on;
```



Q7_1:

```
% Q7 branch 1
% Mohammad Salem

% Define the time range
t = -10:0.01:10;

% Define the function y(t)
y = @(t) (10 * exp(-(3/2) * t)) .* (t >= 0);

% Fourier transform of y(t)
Y = fftshift(fft(y(t)));

%  frequency axis
Fs = 1 / (t(2) - t(1));  % Sampling frequency
f = linspace(-Fs/2, Fs/2, length(t));

% Plot the magnitude spectrum
figure;
plot(f, abs(Y));
xlabel('Fre (Hz)');
ylabel('Magnitude');
title('Spectral Representation');
grid on;
```

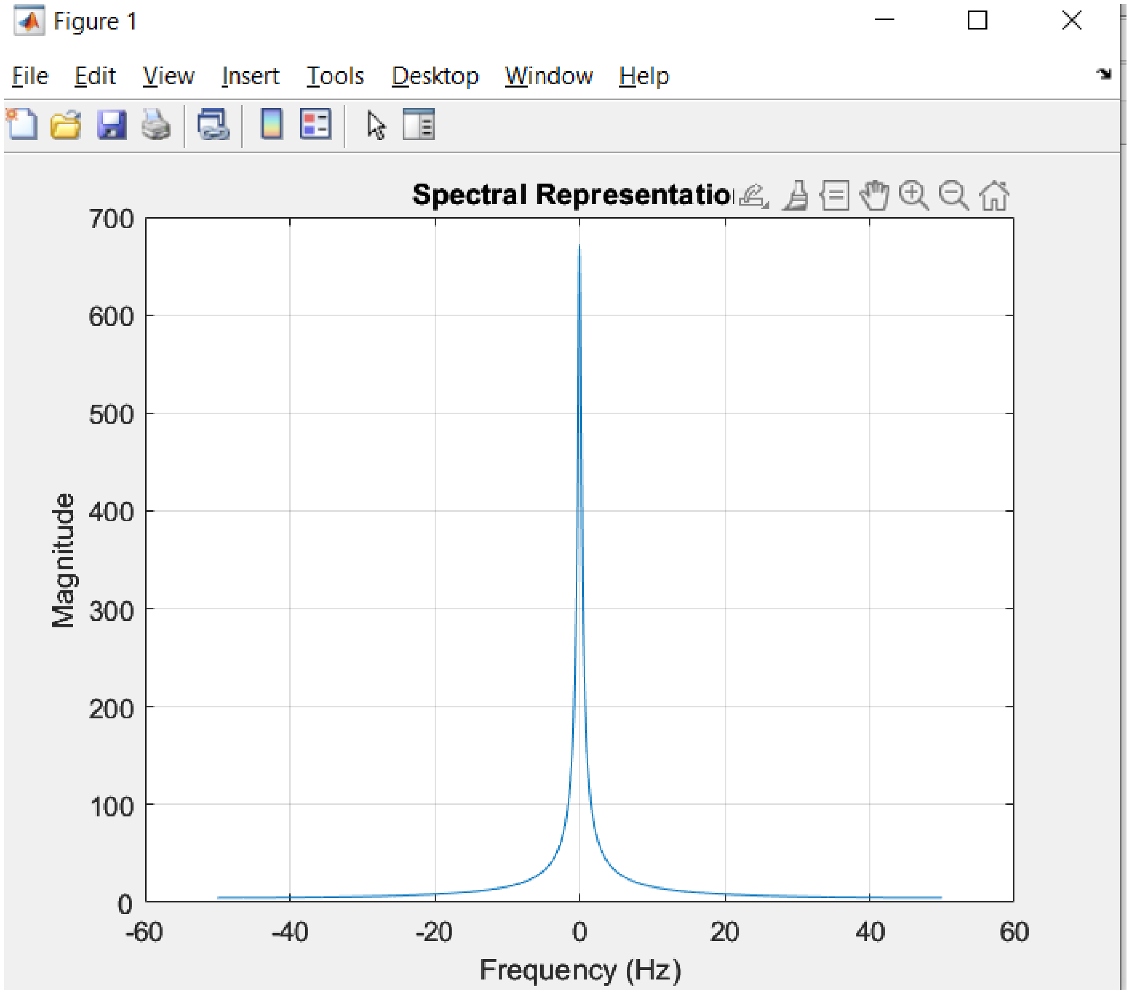


Figure 1
File
Edit
View
Insert
Tools
Desktop
Window
Help
Spectral Representatio
700
600
500
400
300
200
100
0
Magnitude
-60
-40
-20
0
20
40
60
Frequency (Hz)

Q7_2:

```
% Q7 branch 2
% Mohammad Salem

% Define the time range
t = -10:0.01:10;

% Define the function y(t)
y = @(t) (10 * exp(-0.5 * t) .* cos(300 * t)) .* (t >= 0);

% Compute the Fourier transform of y(t)
Y = fftshift(fft(y(t)));

% Compute the frequency axis
Fs = 1 / (t(2) - t(1));  % Sampling frequency
f = linspace(-Fs/2, Fs/2, length(t));

% Plot the magnitude spectrum
figure;
plot(f, abs(Y));
xlabel('Freq (Hz)');
ylabel('Magnitude');
title('Spectral Representation');
grid on;
```

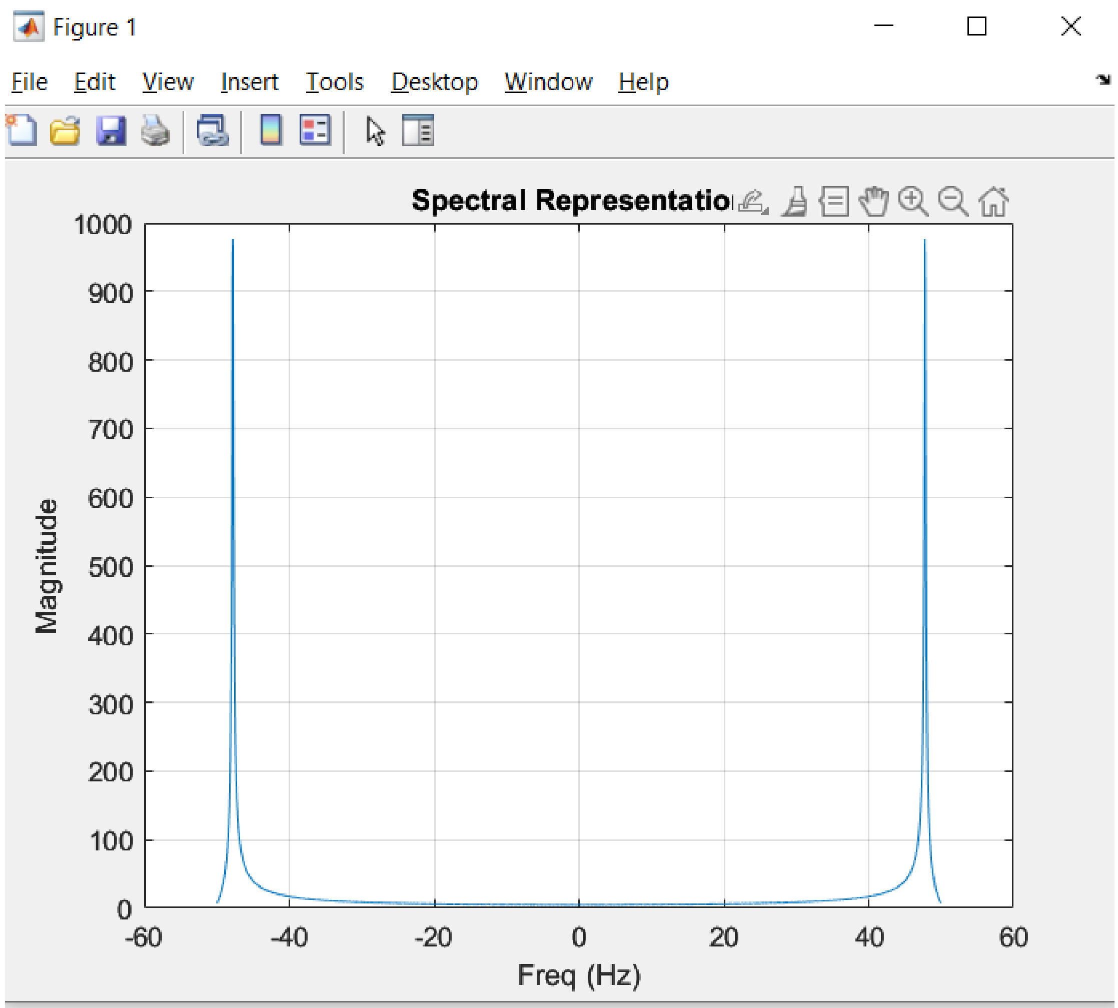


Figure 1
File Edit View Insert Tools Desktop Window Help
Spectral Representatio
Magnitude
Freq (Hz)
1000
900
800
700
600
500
400
300
200
100
0
-60
-40
-20
0
20
40
60

Q8_3(according assignment counting):

```
%Q8 branch 3

syms t s;   % Define symbolic variables

% Define the function y(t)
y = (15 - 15*exp(-0.25*t)) * heaviside(t);

% Compute the Laplace transform of y(t)
Y = laplace(y, t, s);

% Display the Laplace transform
pretty(Y)
```

```
>> Q7
               /   3               \     /   3                \     /   3            \
               |  ---              |     |  ---             2 |     |  ---           |
               |  \   exp(#2 t) #2 |     |  \   exp(t #2) #2  |     |  \   exp(t #2)  |
   /   t \     |  /   ------------ | 6   |  /   ------------- | 6   |  /   ---------  | 6
exp| - - | 3   |  ---      #1      |     |  ---       #1      |     |  ---     #1     |
   \   2 /     \ k = 1             /     \ k = 1              /     \ k = 1           /
------------ - ----------------------- + ------------------------ - ---------------------
      7                   7                          7                         7

where

                    2
   #1 == 2 #2 - 6 #2  + 3

             /       2                 \
             |  3   z    3 z           |
   #2 == root| z  - -- - --- + 3, z, k |
             \       2    2            /


>> Q7
>> Q8
15     15
-- - -----
 s       1
     s + -
         4

fx >>
```

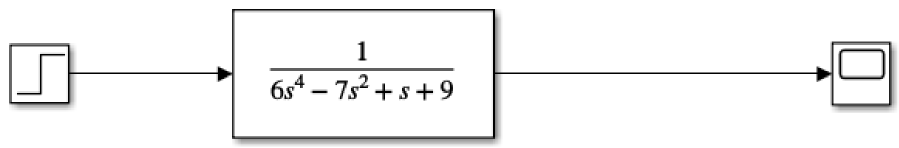
Q8_4(according assignment counting):

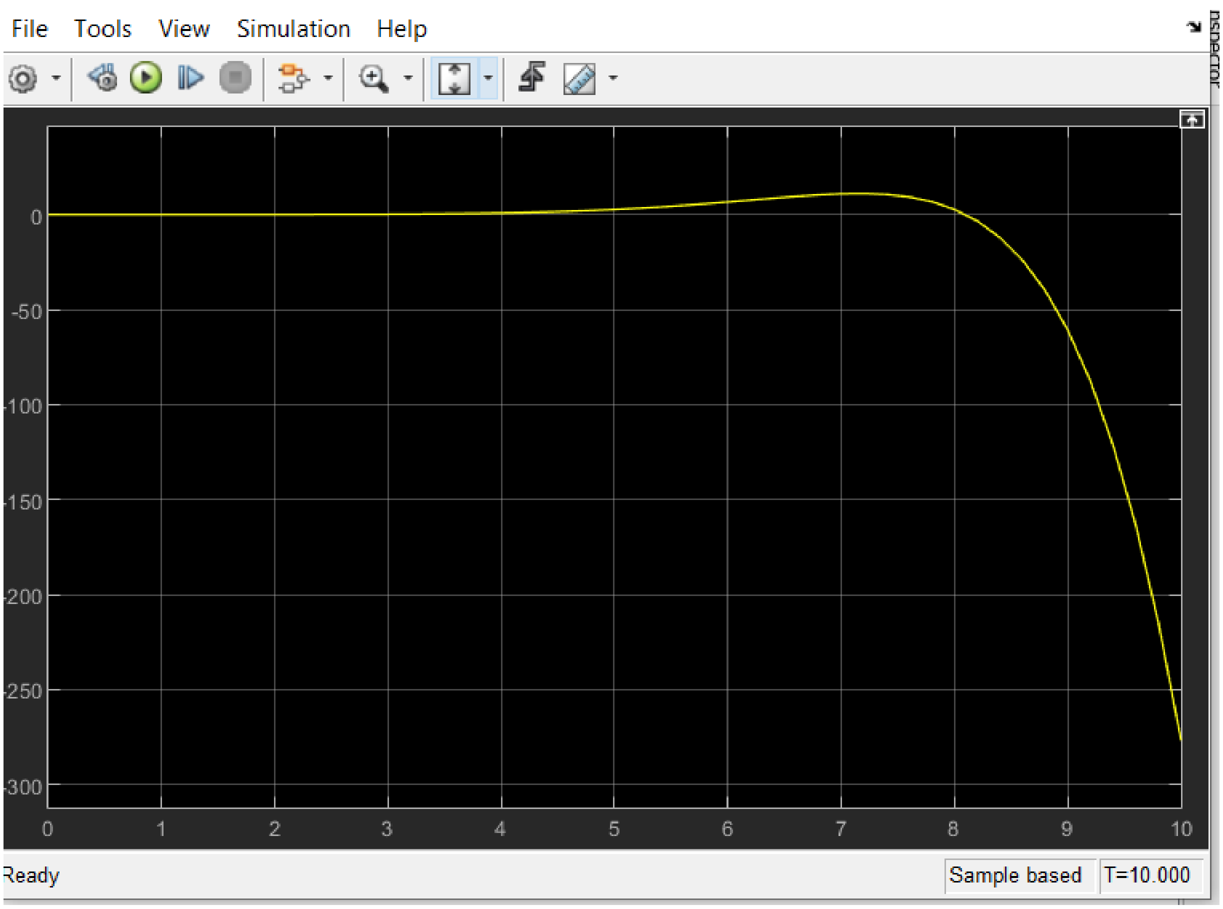
```
%Q8 branch 4

syms t s;   % Define symbolic variables

% Define the function y(t)
y = (20 - 8*exp(-3*t)*cos(100*t)) * heaviside(t);

% Compute the Laplace transform of y(t)
Y = laplace(y, t, s);

% Display the Laplace transform
pretty(Y)
```

Command Window

```
>> Q8
20        8 (s + 3)
-- - ----------------
 s          2
     (s + 3)  + 10000

>>
```

Q9:


$\frac{1}{6s^4 - 7s^2 + s + 9}$


File Tools View Simulation Help
0
-50
-100
-150
-200
-250
-300
0 1 2 3 4 5 6 7 8 9 10
Ready
Sample based
T=10.000

Q10:

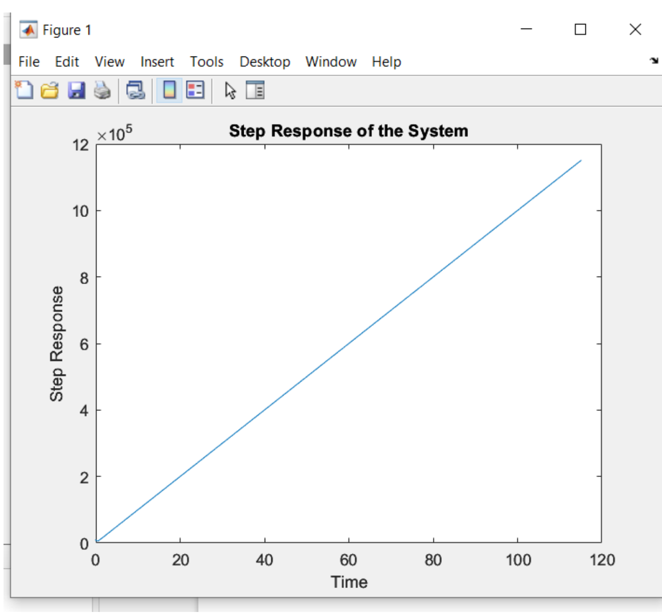
```matlab
% Q10
% Mohammad Salem

syms s t;

% Define the transfer function
H = (6/(4*s^4 - 7*s^2 + 1 + 9*s + 5));

% Compute the inverse Laplace transform
h = ilaplace(H, s, t);

% Display the inverse Laplace transform
pretty(h)
```

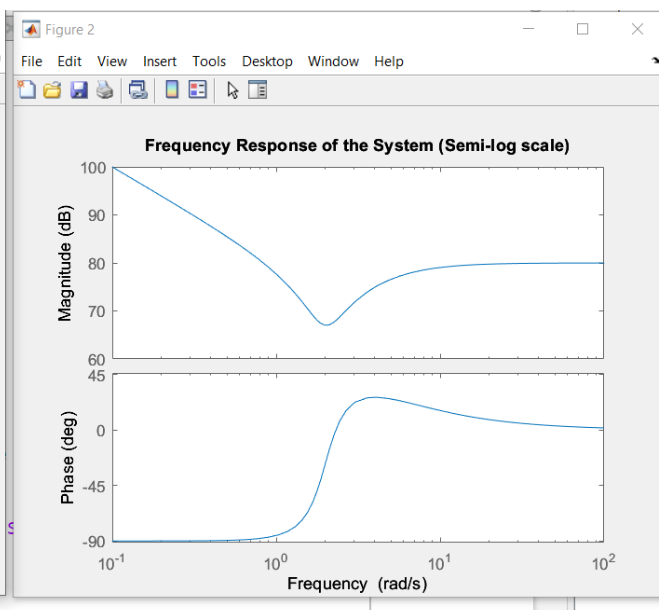
Command Window

```
>> Q7
               /   3               \     /   3                \     /   3            \
               |  ---              |     |  ---             2 |     |  ---           |
               |  \   exp(#2 t) #2 |     |  \   exp(t #2) #2  |     |  \   exp(t #2) |
   /  t \      |  /   ------------ | 6   |  /   ------------- | 6   |  /   --------- | 6
exp| - - | 3   |  ---      #1      |     |  ---      #1       |     |  ---     #1    |
   \  2 /      \ k = 1             /     \ k = 1              /     \ k = 1          /
------------ - ----------------------- + ------------------------ - --------------------
      7                  7                          7                         7

where

                    2
   #1 == 2 #2 - 6 #2   + 3

             /       2          \
             |  3   z    3 z    |
   #2 == root| z  - -- - --- + 3, z, k |
             \       2    2     /

fx >>
```

Q12(according assignment counting):

```
%Mohammad Salem 1200651

num = [10000 30000 60000 80000];
den = [1 6 8 0];

% Compute the step response
sys = tf(num, den);
[y, t] = step(sys);

% Plot the step response
figure;
plot(t, y);
xlabel('Time');
ylabel('Step Response');
title('Step Response of the System');
disp(step(sys));
% Compute and plot the frequency response
figure;
bode(sys);
title('Frequency Response of the System (Semi-log scale)');

```





Thanks….

Mohammad Salem 1200651