Faculty of Engineering & Technology
Electrical & Computer Engineering Department

SIGNALS AND SYSTEMS (ENEE2312)

"Matlab Assignment"

Prepared by:

Eman Asfour 1200206

Instructor: Dr. Jamal Seyam
Assistant: Eng. Shadi Bannoura

Section: 1

Date: 12/06/2022

Table of contents

Table of figures

1. Abstract

This paper will be answering the given questions by applying them to the "Matlab" software, plots, figures, answers and codes will be all contained in, and the codes will be clear and fully explained by the comments added next to each line.

2. Question I

2.1. Generate and plot the following signals using MATLAB: $x_1 = u(t-4) - u(t-9)$

Starting by clearing the work space and the variables, followed up by defining the time matrix, then the required step function using the built-in heaviside() function.

The Code:

```
clear
clc
t = [0:.1:22]; % to define t values that starts at 0 ends at 22 incrememnt by .1 in a matrix
x1 = heaviside(t-4)- heaviside(t-9); % define the function x1 (step functions combinantion)
figure(); %initiakizing a figure
plot (t,x1,'linewidth',3) %plotting t versus x1 with bold line
ylabel('x1(t)') %adding label to the y axis
xlabel('t') %adding label to the x axis
title('x1(t) = u(t-4) - u(t-9)') % adding title to the plot
axis([0 15 -.5 1.5]) % managing the axises to get the signal visually clear
```

Figure 1: Question I.1 Code

The running output:

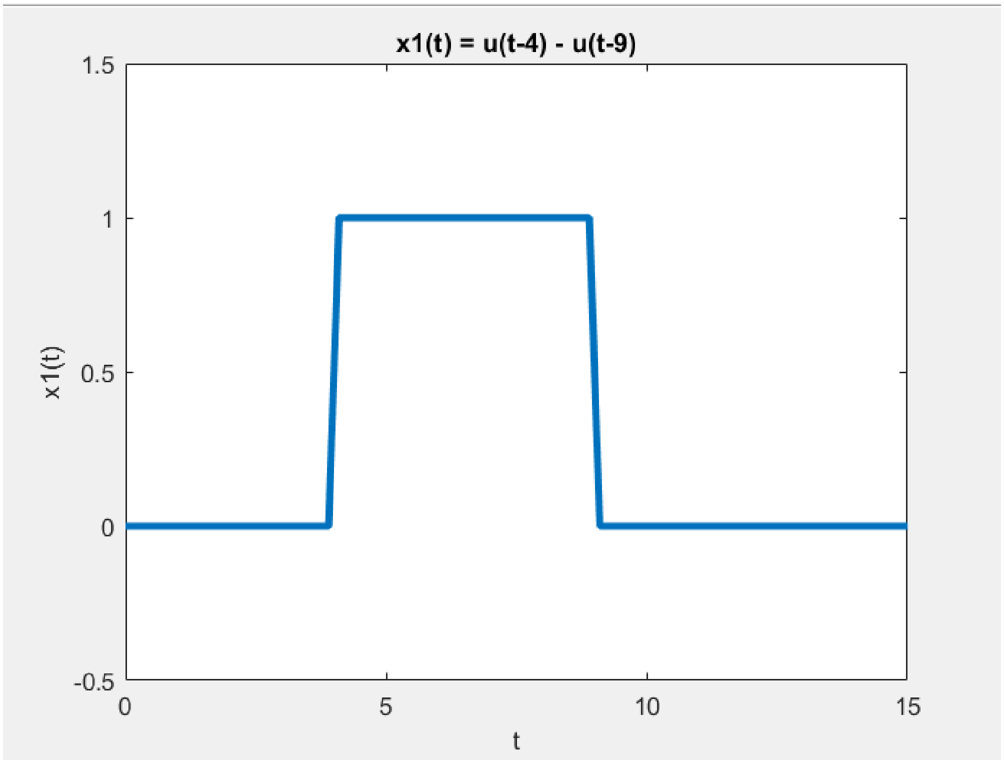


Figure 2: Question I.1 Output

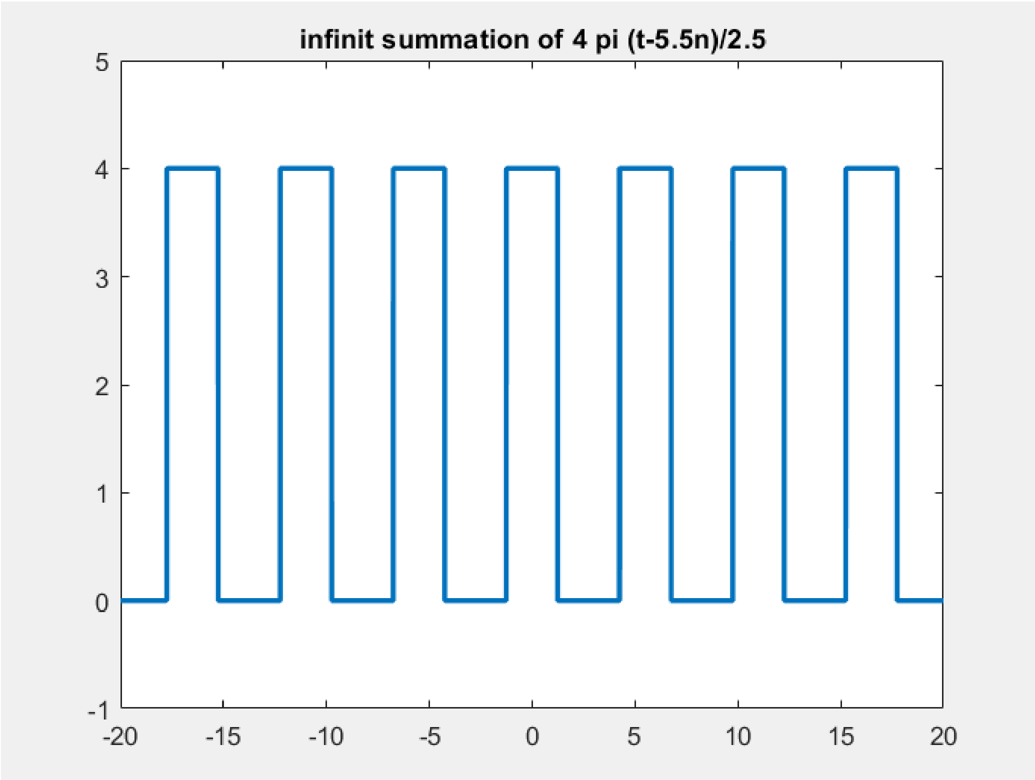
2.2. . Generate and plot the following signals using MATLAB: A finite pulse ($\pi(t)$) with value = 4 and extension between 3 and

Starting by defining symbolics and the infinite summation then plotting and managing plots

The Code:

```
clear
clc
syms t n%defining symboics
x = symsum(4*rectangularPulse((t-5.5*n)/2.5),n,-inf,inf);%dedfining infinit
summation
fplot(x,'linewidth',2)%plotting the summation
title ('infinit summation of 4 pi (t-5.5n)/2.5')%managing plots
axis([-20 20 -1 5])
```

The running output:



2.3. Generate and plot the following signals using MATLAB: $x_2 = u(t-4) + r(t-6) - 2r(t-9) + r(t-11)$

Starting by clearing the work space and the variables, followed up by defining the time matrix, then the required step function using the built-in heaviside() function, noting that r functions aren't defined individually.

The Code:

```
clear
clc
t = [0:.1:22]; % to define t values that starts at 0 ends at 22 incrememnt by .1 in a matrix
x2 = (heaviside(t-4))+((t-6).*heaviside(t-6))-(2*(t-9).*heaviside(t-9))+((t-11).*heaviside(t-11)); % define the function x2
figure(); %initiakizing a figure
plot (t,x2,'linewidth',3) %plotting t versus x2
ylabel('x2(t)') %adding label to the y axis
xlabel('t') %adding label to the x axis
title('x2(t) = u(t-4) + r(t-6) - 2r(t-9) + r(t-11)') % adding title to the plot
axis([0 22 -0.5 4.5]) % managing the axises to get the signal visiually clear
```

Figure 3:Question I.3 Code

The running output:

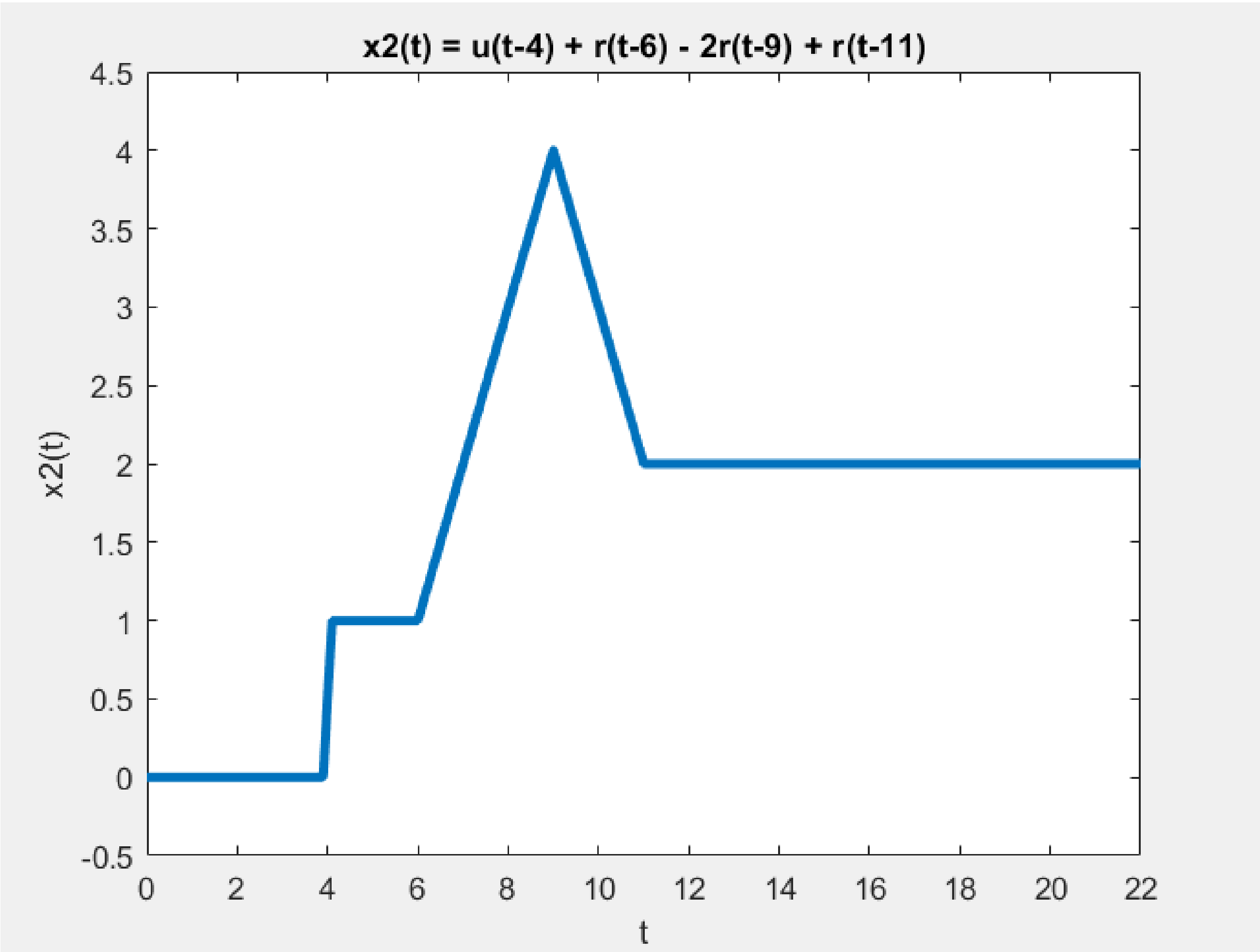


Figure 4:Question I.3 Output

3. Question II

3.1. Generate and plot the signals $y_1(t) = sin(200\,\pi\,t), y_2(t) = cos(750\,\pi\,t)$

Starting by clearing the work space and the variables, followed up by defining the time matrix -which limits is determined by trying many times to get the suitable values-, then defining the required functions and doing the additive and subtraction then plot them in one figure made of 4 subplots, each is defined by itself with the title and labels also with a grid.

The Code:

```
clc
clear
t = [0:.00001:.05];%defining the time matrix
y1 = sin(200*pi.*t);%the first function
y2 = cos(750*pi.*t);%the second function
figure();%initializing a figure
subplot(2,2,1)%inserting subplot in position 1
plot(t,y1)%plotting t versus y1 in the subplot
axis([0 0.02 -1.5 1.5]) %managing axises
xlabel('t')%adding label on the x axis
ylabel('y1(t)')%adding label in the y axis
title('y1 = sin(200*pi.*t)')%adding title to the
subplot
grid on;%turning the grid on
subplot(2,2,2)%inserting subplot in position 2
plot(t,y2)%plotting t versus y2 in the subplot
axis([0 0.02 -1.5 1.5])%managing axises
xlabel('t')%adding label on the x axis
ylabel('y2(t)')%adding label in the y axis
title('y2 = cos(750*pi.*t)')%adding title to the
subplot
grid on;%turning the grid on
y3 = y1 + y2 ;%defining y3 summation of y1 and y2
y4 = y1 - y2 ;%defining y4 subtraction of y1 and y2
subplot(2,2,3)%inserting subplot in position 3
plot(t,y3)%plotting t versus y3 in the subplot
axis([0 0.04 -2.5 2.5]) %managing axises
xlabel('t')%adding label on the x axis
ylabel('y3(t)')%adding label in the y axis
title('y3 = y1 + y2')%adding title to the subplot
grid on;%turning the grid on
subplot(2,2,4)%inserting subplot in position 4
plot(t,y4)%plotting t versus y4 in the subplot
axis([0 0.04 -2.5 2.5])%managing axises
xlabel('t')%adding label on the x axis
ylabel('y4(t)')%adding label in the y axis
title('y4 = y1 - y2')%adding title to the subplot
grid on;%turning the grid on
```

Figure 5:Question II.1 code

The running output:

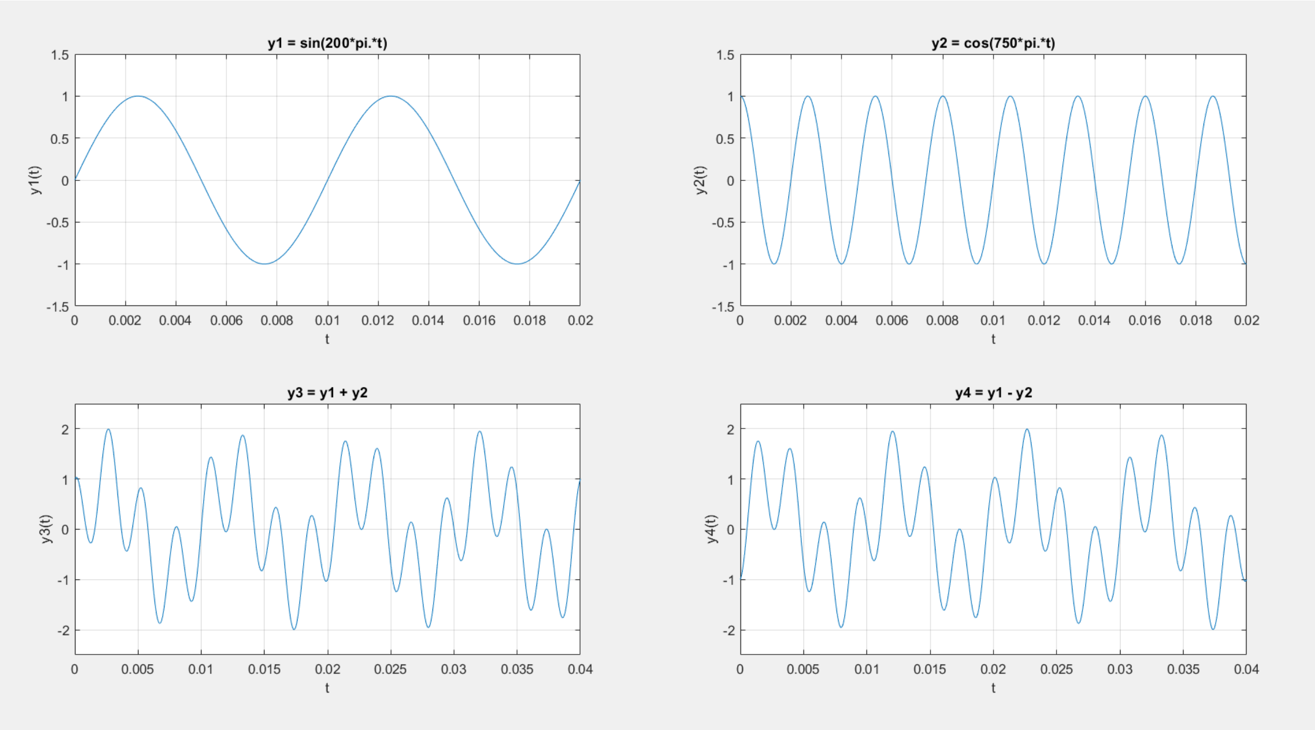


Figure 6:Question II.1 Output

3.2. Determine, using the MATLAB plots, if the sum and/or difference signals are periodic. In case a signal is periodic, determine its fundamental frequency.

From the plot it is clear that y1 is periodic with a fundamental frequency of 100Hz.

For y2, the plot shows that it is periodic with a periodic time of about .0025s and using calculation w=2*pi*f , it is clear that the fundamental frequency is 350Hz.

To determine for the third and fourth functions, more data are needed so, another figure with more values is initialized and the code has been edited to give the following figures:

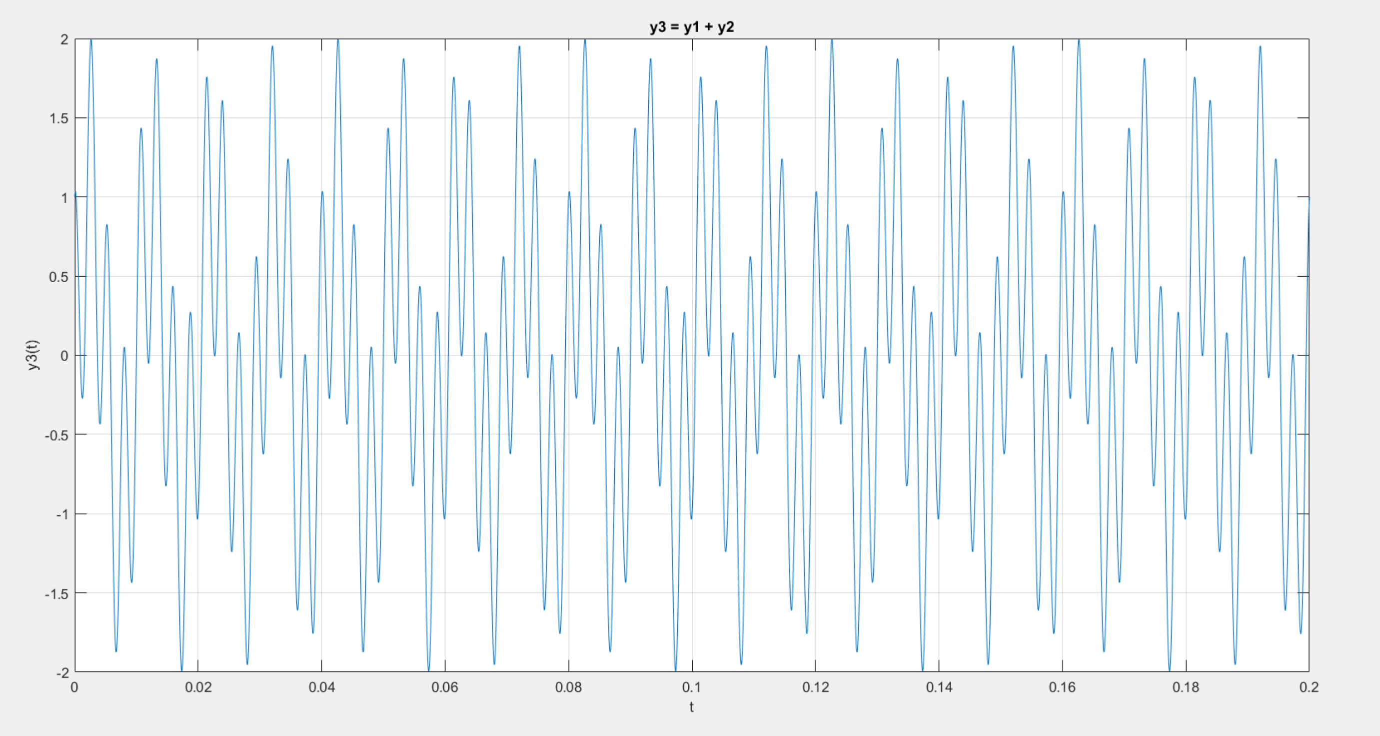


Figure 7: Third function expanded

It is clear that the function is periodic, with the period of .04, which gives a fundamental frequency of 24 Hz.

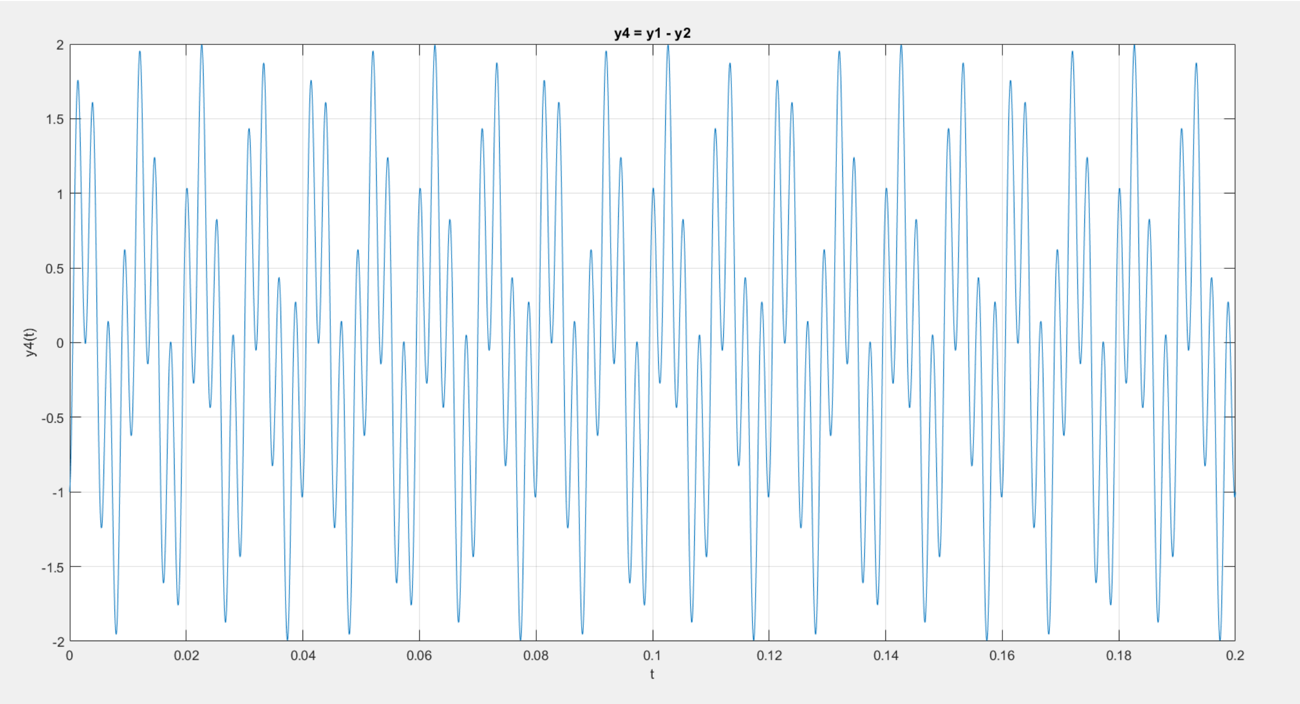


Figure 8: Fourth function expanded

Giving the same result, this function is periodic with 25Hz fundamental frequency.

4. Question III

4.1. Write the programs that solve the following differential equations using zero initial conditions: $10\frac{dy}{dt} + 20\,y(t) = 10$

Starting by rewrite the equation to be: $\frac{dy}{dt} = 1 - 2y(t)$

Then clearing the workspace and variables, the first step is to represent y by symbolic function, then defining the equation and solve it , then represent the solution in y(t) to receive the results.

The Code:

```
clc
clear
syms y(t)%representting y by symbolic function
Given = diff(y,t) == (1 - 2*y(t))%defining the equation
y(t) = dsolve(Given,y(0)==0) %solving the function with the initial condition and
defining it to y(t)
```

Figure 9: Question III.1 Code

The running output:

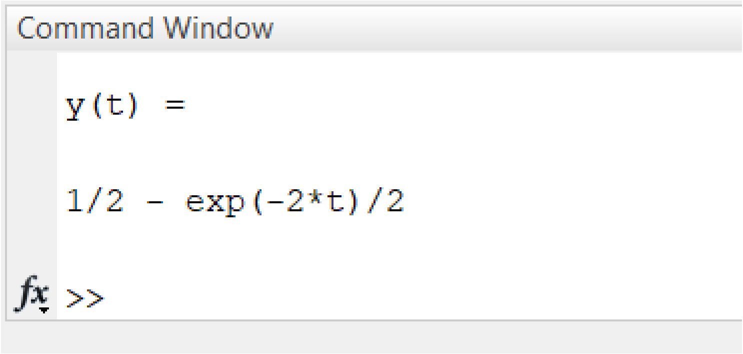


Figure 10: Question III.1 Output

Therefore the solution is: $y(t) = \frac{1}{2}(1 - e^{-2t})$

4.2. Write the programs that solve the following differential equations using zero initial conditions: $\frac{d}{dt}\left(\frac{dy}{dt}\right) + 2\frac{dy}{dt} + 4y(t) = 5\cos(1000t)$

Starting by rewrite the equation to be: $\frac{d}{dt}\left(\frac{dy}{dt}\right) = 5\cos(1000t) - 2\frac{dy}{dt} - 4y(t)$

Then clearing the workspace and variables, the first step is to represent y by symbolic function, then defining the equation and solve it , then represent the solution in y(t) to receive the results.

The Code:

```
clc
clear
syms y(t)
Dy = diff(y);%defining the
ode = diff(y,t,2) == (5*cos(1000*t)) - (4*y) - (2*diff(y,t));%defining the equation
cond1 = y(0) == 0;%defining the conditions
cond2 = Dy(0) == 0;
conds = [cond1 cond2];%adding the conditions into matrix
ySol(t) = dsolve(ode,conds);%solving
ySol = simplify(ySol) %simplifiyng the results to be shown
```

Figure 11:Question III.2 Code

The running output:

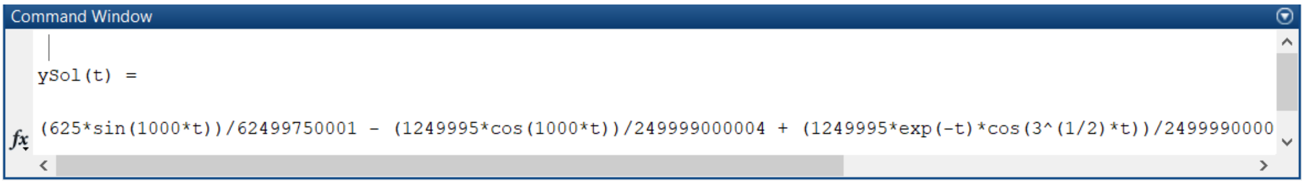


Figure 12:Question III.2 Output

The output solution is:

```
ySol(t) =

(625*sin(1000*t))/62499750001 - (1249995*cos(1000*t))/249999000004 + (1249995*exp(-t)*cos(3^(1/2)*t))/249999000004 - (1250005*3^(1/2)*exp(-t)*sin(3^(1/2)*t))/749997000012
```

Figure 13:Question III.2 expanded Output

5. Question V

5.1. Use Simulink (MATLAB) to simulate the following systems then show and plot the step response of the system: : $4\frac{d^4y(t)}{dt} + 6\frac{dy(t)}{dt} + 8y(t) = 7\frac{d^2y(t)}{dt} + 12(xt)$

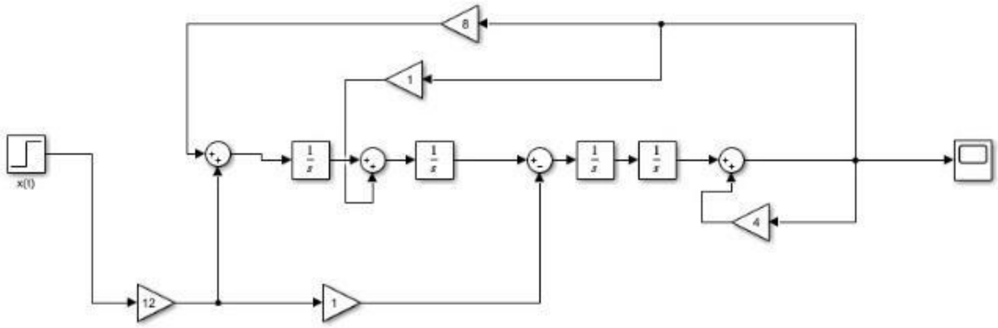


Figure 14:Question V.1 Simulating circuit

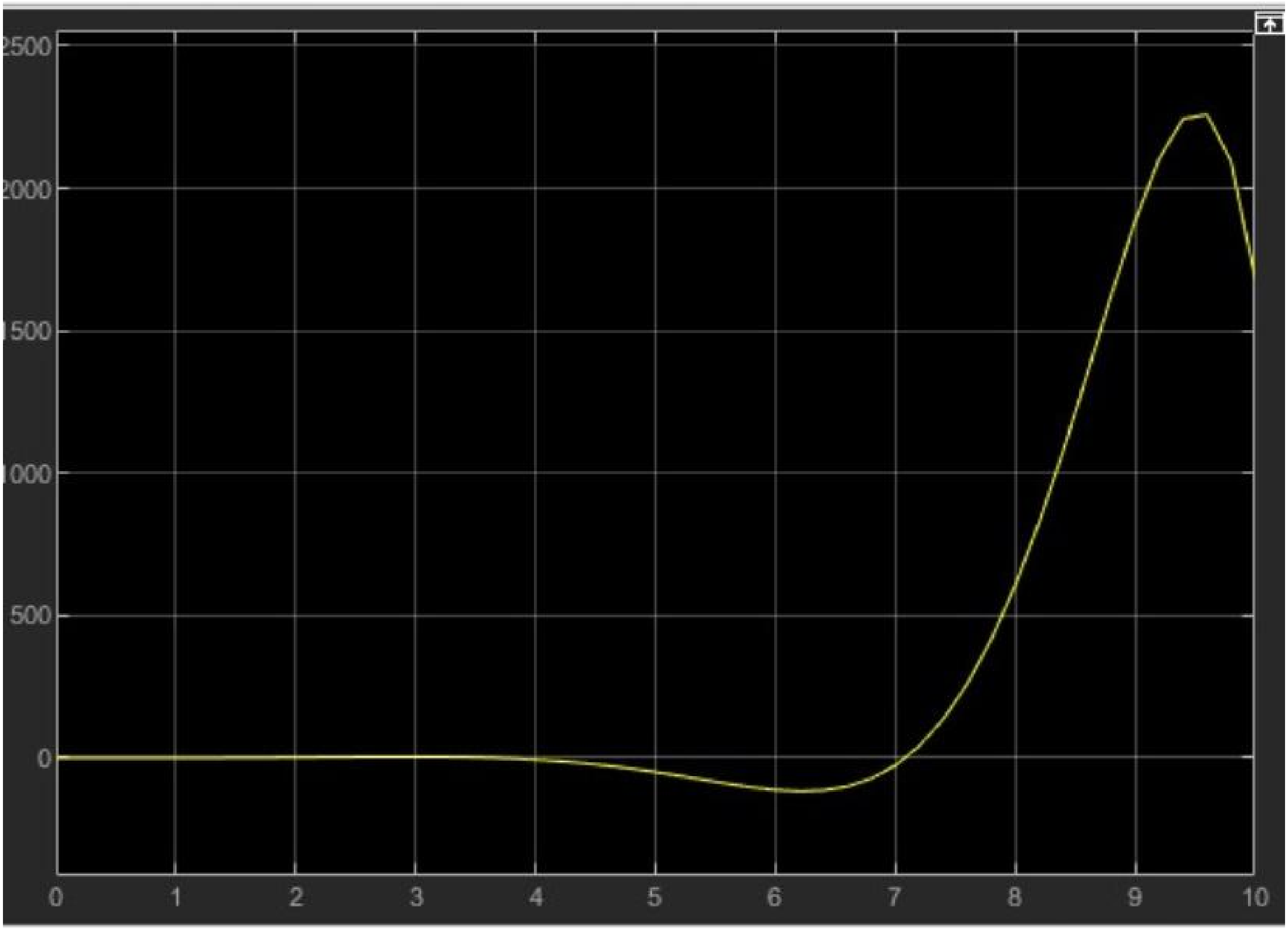


Figure 15: Question V.1 Simulating output

5.2. Use Simulink (MATLAB) to simulate the following systems then show and plot the step response of the system: : $H(s) = \frac{100(s+3)}{(s+1)(s+4)} + \frac{10}{(s+10)}$

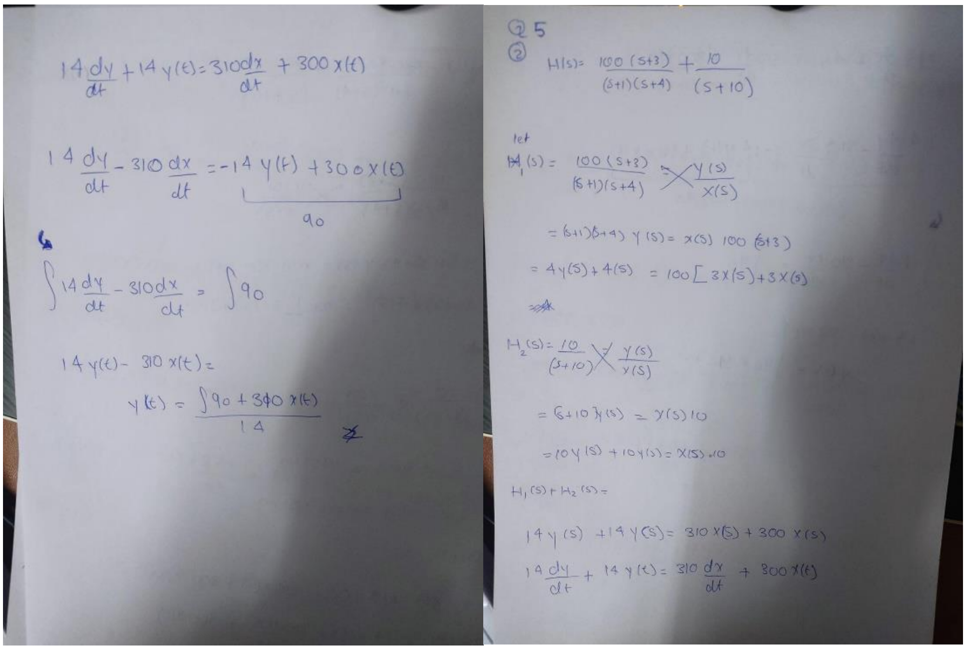


Figure 16: Question V.2 Solving Manually

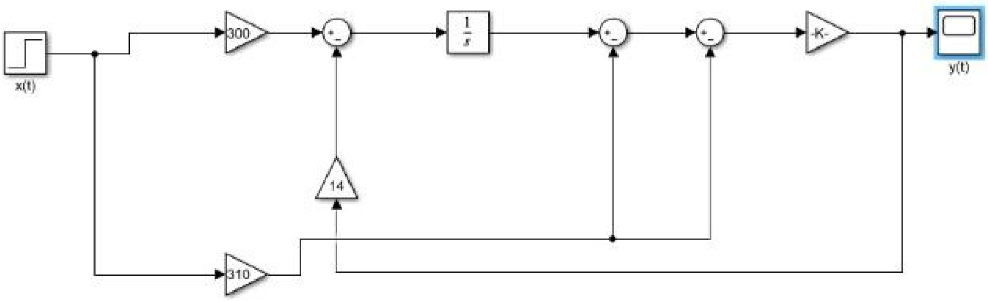


Figure 17: Question V.2 simulating circuit

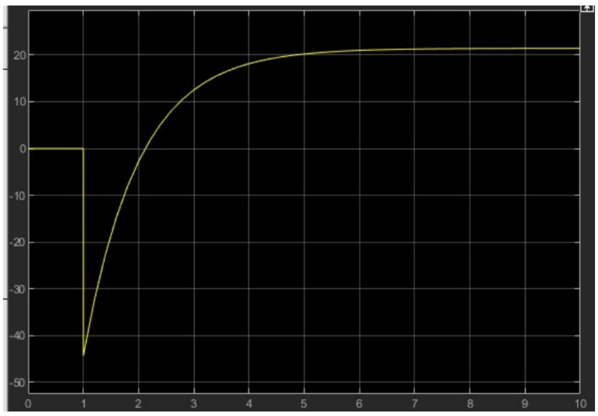


Figure 18: Question V.2 simulation Result

6. Question VI

6.1. Write a program that computes and plots the spectral representation of the function: $y(t) = (10 - 10e^{-10t})u(t)$

Starting by defining symbolic and unit step function and given function then plotting them and use built in functions

The Code:

```
clc
clear
syms t %defining symblolic
u = heaviside(t);%defining the unit step function
f = (10 - 10*(exp(-10*t))).*u; %defining the given function

subplot(2,2,[1,3])
fplot (f,"linewidth",2)%plotting the results and managing them
title('10 - 10*(exp(-10*t)')
axis([-5 5 -5 15])
subplot(2,2,2)
fplot(phase(f),'linewidth',5)
title('phase spectra')
subplot(2,2,4)
fplot(abs(f),'linewidth',2)
title('amplitude spectra')
axis([-5 5 -5 15])
```

Figure 19: Question VI.1 code

The running output:

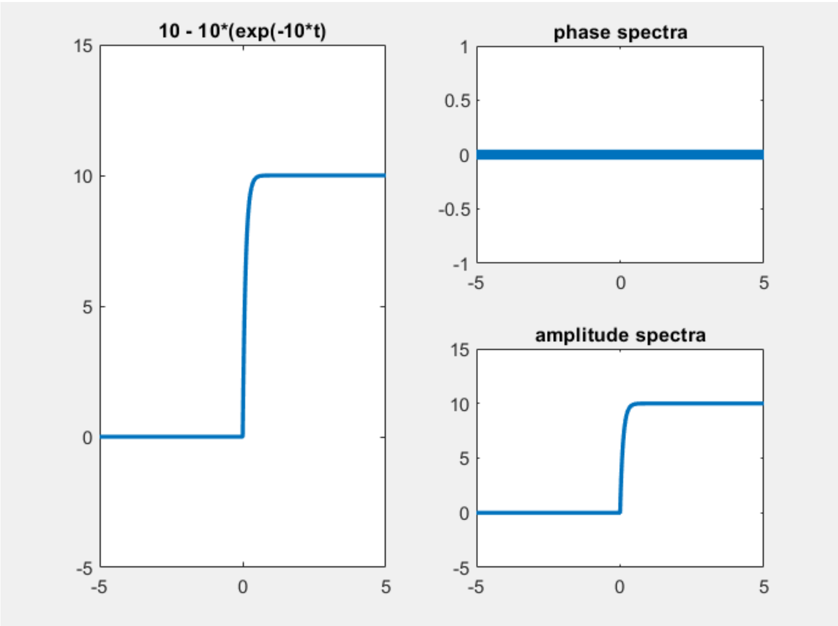


Figure 20: Question VI.1 Output

6.2. Write a program that computes and plots the spectral representation of the function: $y(t) = (10 - 10e^{-10t}cos(100t))u(t)$

Starting by defining symbolic and unit step function and given function then plotting them and use built in functions

The Code:

```
clc
clear
syms t %defining symblolic
u = heaviside(t);%defining the unit step function
f = (10 - cos(100*t)*10*(exp(-10*t))).*u; %defining the given function

subplot(2,2,[1,3])
fplot (f,"linewidth",2)%plotting the results and managing them
title('10 - cos(100*t)*10*(exp(-10*t)')
axis([-2 2 -5 25])
subplot(2,2,2)
fplot(phase(f),'linewidth',5)
title('phase spectra')
subplot(2,2,4)
fplot(abs(f),'linewidth',2)
title('amplitude spectra')
axis([-2 2 -5 25])
```

Figure 21: Question VI.2 Code

The running output:

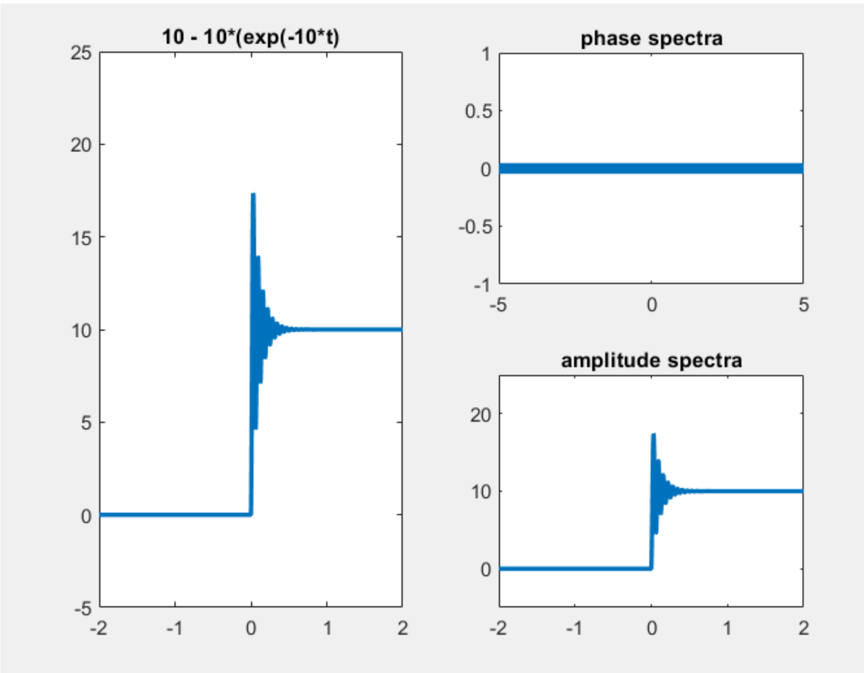


Figure 22:Question VI.2 Output

7. Question VII

7.1. Write a program that computes the Laplace and Fourier transforms of the function and plot the phase and amplitude spectra: $y(t) = (10 - 10e^{-5t})u(t)$

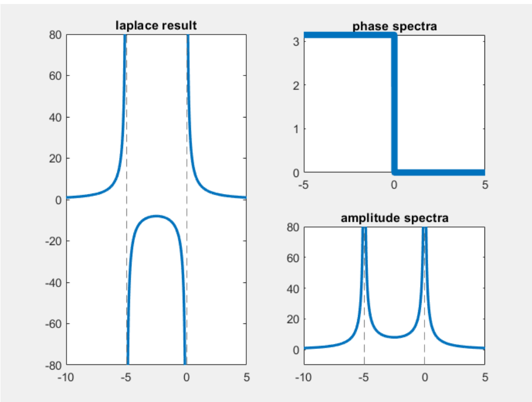
Starting by defining symbolic, step function, the given equation, followed by calculatimg the laplace and fourier then plotting results with the amplitude and phase spectras calculated by the built-in functions (phase, abs).

The Code:

```
clc
clear
syms t %defining symblolic
u = heaviside(t);%defining the unit step function
f = (10 - 10*(exp(-5*t))).*u; %defining the given function
c=laplace(f);% calculating the laplace
fourier(f)%calculating the fourier
subplot(2,2,[1,3])
fplot (c,"linewidth",2)%plotting the results
title('laplace result')
axis([-10 5 -80 80])
subplot(2,2,2)
fplot(phase(c),'linewidth',5)
title('phase spectra')
subplot(2,2,4)
fplot(abs(c),'linewidth',2)
title('amplitude spectra')
axis([-10 5 -10 80])
```

Figure 23: Question VII.1 Code

The running output:



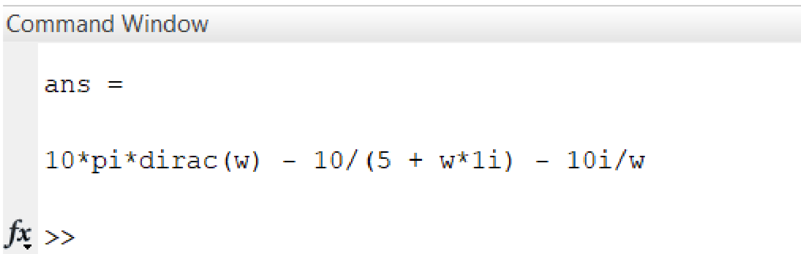


Figure 24:" Question VII.1 Output Function

Figure 25: Question VII.1 Output figure

7.2. Write a program that computes the Laplace and Fourier transforms of the function and plot the phase and amplitude spectra: $y(t) = (30 - 10e^{-8t}\cos(100t))u(t)$

Starting by defining symbolic, step function, the given equation, followed by calculatimg the laplace and fourier then plotting results with the amplitude and phase spectras calculated by the built-in functions (phase, abs).

The Code:

```
clc
clear
syms t %defining symblolic
u = heaviside(t);%defining the unit step function
f = (30 - (10*(exp(-8*t))*cos(100*t))).*u;%defining the given function
c=laplace(f)% calculating the laplace
fourier(f)%calculating the fourier
subplot(2,2,[1,3])
fplot (c,"linewidth",2)%plotting the results
title('laplace result')
axis([-10 10 -80 80])
subplot(2,2,2)
fplot(phase(c),'linewidth',5)
title('phase spectra')
subplot(2,2,4)
fplot(abs(c),'linewidth',2)
title('amplitude spectra')
axis([-10 10 -10 80])
```

Figure 26:Question VII.2 Code

The running output:

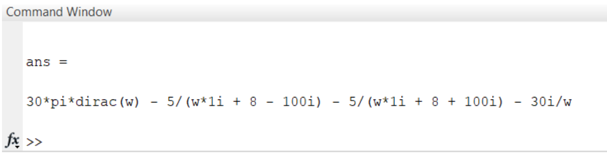


Figure 27: Question VII.2 Output function

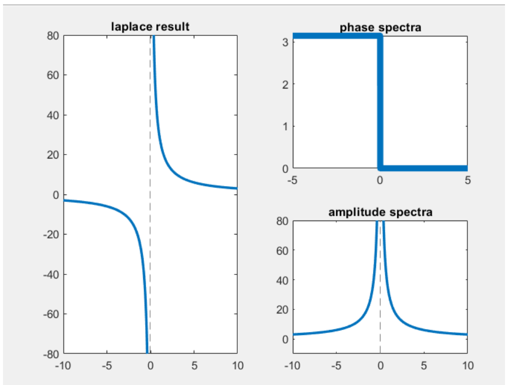


Figure 28:Question VII.2 Output Figure

8. Question VIII

8.1. Write a program that define the transfer functions and plots the zero-pole map of the systems: with poles (-1,-3) and zero (-6)

Starting by defining zeros, poles and gain, then Converting zero-pole-gain filter parameters to transfer function and converting to transfer function model

The Code:

```
clc
clear
z = -6; %defining zeros
p = [-1; -1]; %defining poles
k = 1; %defining gain
[num,den] = zp2tf(z,p,k); %Converting zero-pole-gain filter parameters to transfer
function form
tf(num,den) % converting to transfer function model
```

Figure 29: Question VIII.1 Code

The running output:

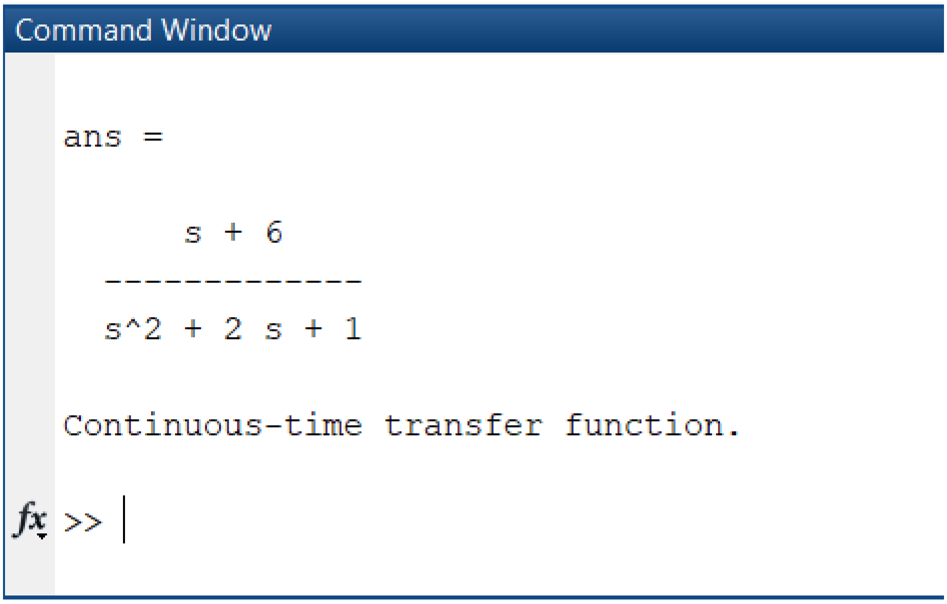


Figure 30: Question VIII.1 Output

8.2. Write a program that define the transfer functions and plots the zero-pole map of the systems: with poles (-1, 1+2j and 1-2j) and zero at (-3)

Starting by defining zeros, poles and gain, then Converting zero-pole-gain filter parameters to transfer function and converting to transfer function model

The Code:

```
clc
clear
z = -3; %defining zeros
p = [-1; 1+2i ; 1-2i]; %defining poles
k = 1; %defining gain
[num,den] = zp2tf(z,p,k); %Converting zero-pole-gain filter parameters to transfer
function form
tf(num,den) % converting to transfer function model
```

Figure 31:Question VIII.2 Code

The running output:

```
Command Window

ans =

          s + 3
  ---------------------
  s^3 - s^2 + 3 s + 5

Continuous-time transfer function.

fx >>
```

Figure 32:Question VIII.2 Output

9. Question IX

9.1. Write a program that determine the inverse Laplace and Fourier transforms of the transfer functions in VIII and plot their phase and magnitude spectra. First result:

Starting by defining symbolic, defining the function resulted in question 7, and initializing the figure. A plot of the function has been made, followed by the phase spectra using the built in command and using the absolute value command for the amplitude spectra.

The Code:

```
clc
clear
syms s %defining sympolic
f = (s+6)/(s^2 + 2*s + 1);% defining the function found in the result of question
7
IL = ilaplace(f)
figure%initializing figure
subplot(2,2,1)
fplot(IL,'linewidth',2)%plotting f
title("Question 7.1 result")
subplot(2,2,2)
fplot(phase(IL),'linewidth',2)
title('phase spectra')
subplot(2,2,3)
fplot(abs(IL),'linewidth',2)
title('magnitude spectra')
```

Figure 33:Question IX.1 Code

The running output:

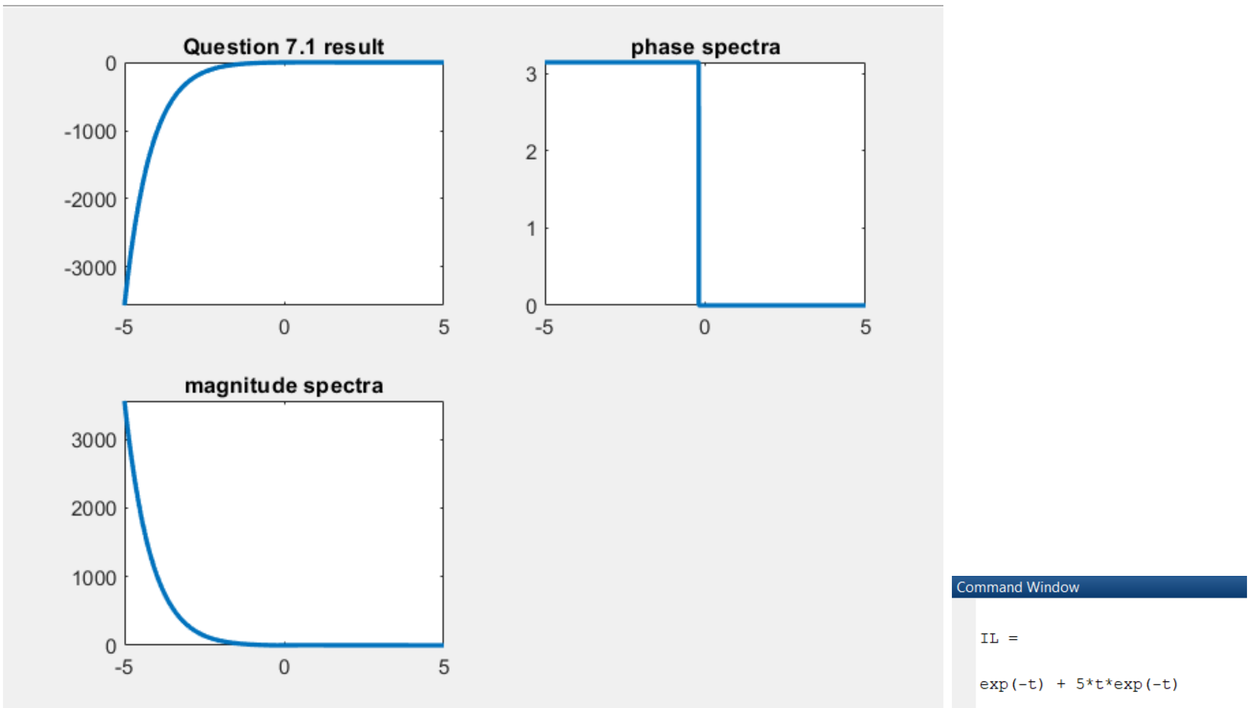


Figure 34Question IX.1 Output

9.2. Write a program that determine the inverse Laplace and Fourier transforms of the transfer functions in VIII and plot their phase and magnitude spectra. Second result:

Starting by defining symbolic, defining the function resulted in question 7, and initializing the figure. A plot of the function has been made, followed by the phase spectra using the built in command and using the absolute value command for the amplitude spectra.

The Code:

```
clc
clear
syms s %defining sympolic
f = 30/s - (10*(s + 8))/((s + 8)^2 + 10000);% defining the function found in the result of
question 7
IL = ilaplace(f)
figure%initializing figure
subplot(2,2,1)
fplot(IL,'linewidth',2)%plotting IL
axis([-5 -4.5 -(10^16) 10^18])
title("Question 7.2 result")
subplot(2,2,2)
fplot(phase(IL),'linewidth',2)
title('phase spectra')
axis([-5 -4.5 -4 1])
subplot(2,2,3)
fplot(abs(IL),'linewidth',2)
title('magnitude spectra')
```

Figure 35:Question IX.2 Code

The running output:

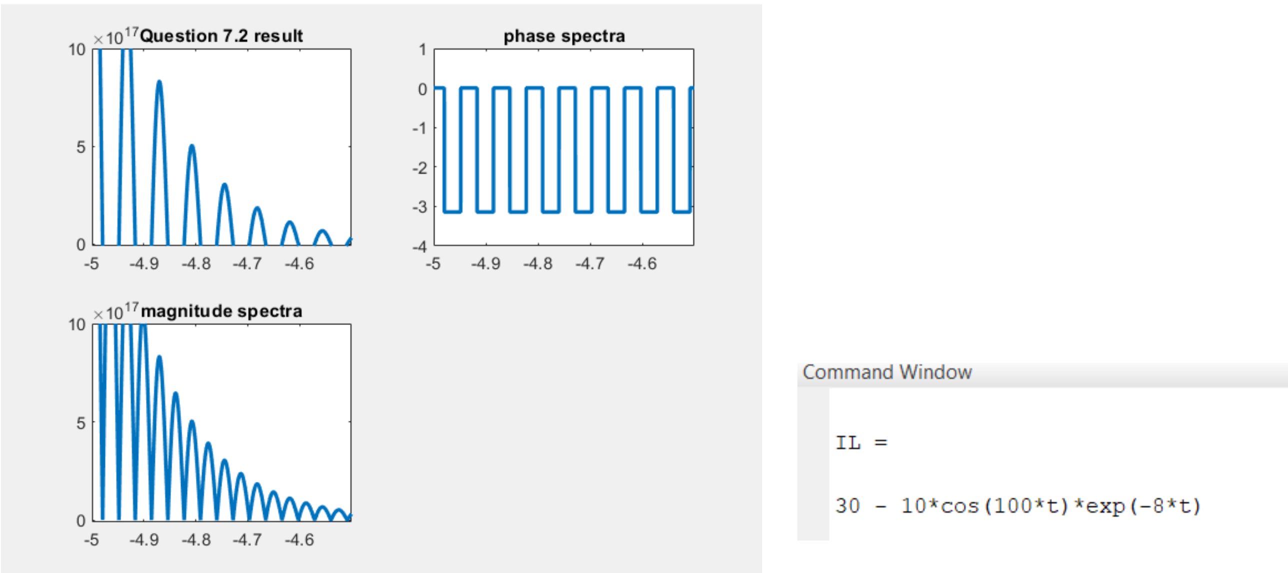


Figure 36:Question IX.2 Output