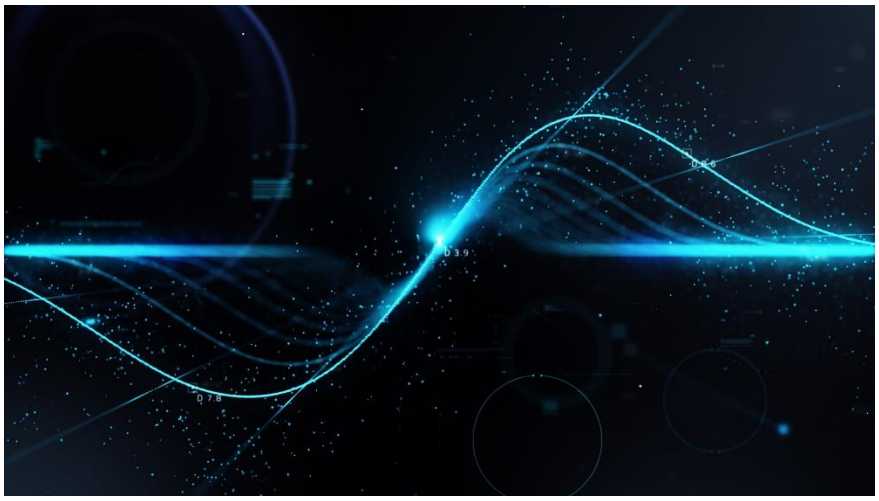
MEHRAN UNIVERSITY OF ENGINEERING & TECHNOLOGY, PAKISTAN

Department of Electronic Engineering



Signals & Systems

PRACTICAL

20ES-BATCH

Name : karan vaswani

Roll # : 20ES062

CERTIFICATE

This is to certify that Mr/Ms. _____karan vaswani_____ , Roll Number _20ES062_____, studying in 3rd year 5th semester in the year 2023 in Department of Electronic Engineering, MUET, Jamshoro, has completed practical course based on the course contents of subject *Signals & Systems-(ES-304)* and given a satisfactory account of it in this lab-book containing a record the laboratory work.

_____ Dated:_____

(Subject Teacher)

INDEX

Sr. #	LIST OF EXPERIMENTS	Date	Signature	Obt. Marks
1.	Lab#01 Introduction to MATLAB			
2.	Lab#02 Signal Generation			
3.	Lab#03: Introduction to Matlab GUI development Environment			
4.	Lab#04: Introduction to Simulink			
5.	Lab#05: Convolution			
6.	Lab#06: Even and Odd Signals			
7.	Lab#07: Sampling, Nyquist Theorem & Effects of Aliasing in Time Domain			
8.	Lab#08: Fourier Series			
9.	Lab#09 : Fourier Transform			
10.	Lab#10: Correlation			
11.	Lab#11: Z Transform			
12.	Lab#12: Butterworth Filter Design			
13.	Lab#13 Chebyshev filter Design			
14.	Lab#14 Introduction to Speech Processing			
15.	Lab#15 Introduction to Image Processing			
16.	Lab#16: Open Ended Lab			
	Total Marks Obtained (Out of 5):			

MEHRAN UNIVERSITY OF ENGINEERING AND TECHNOLOGY, JAMSHORO
Department of Electronic Engineering
SIGNALS & SYSTEMS (ES-304)

LAB # 01: INTRODUCTION TO MATLAB

Name	Karan vaswani	Roll #	20ES062
Signature of Lab Tutor		Date	

OBJECTIVE(S)

#	*Topic*	# Of Lectures	CLO	Taxonomy level
1	The purpose of this lab is to *introduce* you to software tool namely MATLAB.	3	4,5	P3,A4

OUTCOME(S)

a. An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.	**PLO5**:Modern Tool Usage
b. An ability to communicate effectively (written/oral)	**PLO10**: Communication

LAB RUBRICS:

Performance Metric	Good (5-4)	Average (3-2)	Poor (1-0)	Score
Use of modern engineering software [PLO5]	Demonstrates knowledge and application of modern engineering software through accurate development and interpretation of computer programs to solve problems.	Demonstrates awareness of modern engineering software through mostly correct development and interpretation of computer programs to solve problems, but may contain minor mistakes or syntax errors	Unable to use modern engineering software to develop or interpret computer programs to solve problems.	
Level of understanding of the learned skill [PLO5]	Demonstration of full knowledge of the handout with explanations and elaboration.	At ease with content and able to elaborate and explain to some degree.	No grasp of information. Clearly no knowledge of subject matter. No questions are answered. No interpretation made.	
Conducting simulation [PLO5]	Has an excellent simulations skill in the simulator. Always able to make the logical code for the given task.	Has a good simulation skill in the simulator. Always able to make the logical code for the given task.	Has poor simulation skill in the simulator. Unable to make the logical code for the given task.	
Responsiveness to Questions/ Accuracy [PLO10]	Responds well, quick and very accurate all the time.	Generally responsive and accurate most of the times.	Non-responsive at all or the candidate giving only one correct response of viva voce questions.	
Documentation: contents and organization [PLO10]	Report well organized, Appropriately sectioned, uses diagram/dscription when	Report reasonably well documented. May lack some minor aspects.	Report not well organized, lack key aspects.	

	appropriate, important issues clearly stated.			
			Total	

EQUIPMENT/SOFTWARE TOOL:

MATLAB - The Language of Technical Computing.

DISCUSSION AND SETUP:

The MATLAB and Simulink product families are fundamental computational tools at the world's educational institutions. Adopted by more than 3500 universities and colleges, MathWorks products accelerate the pace of learning, discovery, and research in engineering and science. MathWorks products also help prepare students for careers in industry, where they are widely used tools for research and development.

Application Areas:

Technical Computing
Mathematical computation, analysis, visualization, and algorithm development.
Control Design
Model-Based Design for control systems, including simulation, rapid prototyping, and code generation for embedded systems.
DSP - Digital Signal Processing & Communications
Model-Based Design for signal processing and communication systems, including simulation, code generation, and verification.
Image Processing
Image acquisition, analysis, visualization, and algorithm development.
Test & Measurement
Hardware connectivity and data analysis for test and measurement applications.
Computational Biology
Analysis, visualization, and simulation of biological data and systems.
Computational Finance
Financial modeling, analysis, and application deployment.

Some fundamental commands:

Help
To get command-line function help, you can use the MATLAB help function.
For example, to get help for the ‘imread’ function, type,
Help imread

Who
Use who to list the current workspace variables.

Whos
Use whos to list the variables and information about their size and class.

Exist
Use exist to see if the specified variable is in the workspace.

Clear all
All the variables are deleted and the workspace becomes empty. Executing 'who' can verify this.

Close all
All existing figures are closed.

%
Commenting in M-files
You can make any line in an M-file a comment by typing % at the beginning of the line. To put a comment within a line, type % followed by the comment text; MATLAB treats all the information after the % on a line as a comment.

;
Output Suppression
When placed at the end of a command, the semicolon tells MATLAB not to display any output from that command in the command window.

LAB TASKS, OBSERVATIONS & RESULTS:

Generation / Fundamental operations on arrays and matrices:
Write your results and observations after executing each of the following:

1. A=[1 2 3]

Ans: in this type of task the MATLAB software will execute the result as in the matrix fror A= 1,2,3 and save the result in the MATLAB library for the use if needed again to be executed and will show the result by pressing Enter key if semi-colon terminator is not used.

2. Size(A)

Ans: by using the option of size we can easily know about the number of rows and columns in our matrix and we can also create matrix of our need according to our desire and need, and it is instructed as (2,2) for 2-row and 2-columns and is true for n-rows and n-columns.

3. B=[1 2 3;4 5 6;7 8 9]

Ans: by using this command one can create (3x3) matrix where 1,2,3 will be in the 1st row, 4,5,6 will be in the 2nd row and rest will be in the third row and the semi-colon in between the number work as a row-breaker, if not used all the number will be on the same row.

4. A*A'

Ans: this command is used when we want to multiply any two matrix and the MATLAB will check the condition that number of columns of 1st matrix must be equal to the number of rows in the 2nd matrix, if not satisfied MATLAB will show the error in the instruction and will not multiply the matrices

(Please use matrix A defined in '1')

5. A.*A

Ans: this command is used when we want to multiply any two matrix and in this case the MATLAB will not check the condition that number of columns of 1st matrix must be equal to the number of rows in the 2nd matrix, MATLAB will multiply the matrices elements according to the order, 1st row 1st element of 1st matrix with 1st row 1st element of 2nd matrix and so on

6. A+A

Ans: by using this task we can easily add two matrices in the MATLAB but it is kept in consideration the order of matrices must be same

7. Ones(1,4)

Ans: by using this command we can easily generate the matrix having one row and will contain four elements and all the elements would be one in it as [1 1 1 1]

8. Zeros(2,2)

Ans: by using this command one can create (2x2) matrices having elements are zero

9. KK=0:2:10

Ans: this instruction will produce the list of ten number having difference is 2.like 0,2,4,6,8,10,12,14,16,18.

10. B(1,:)

Ans: This instruction is used to extract the row one of the matrix of order (3x3) or any order matrix.The the highlighted will come in the output.

1	**2**	**3**
4	5	6
7	8	9

(Please use matrix B defined in '3')

11. B(:,2)

Ans: This instruction is used to extract the column-2 of the matrix of order (3x3) or any order matrix.The the highlighted will come in the output.

1	**2**	**3**
4	5	6
7	8	9

Signal Generation:

The following piece of code generates an exponential signal. Its output is shown in figure 1.

```
t=0:0.1:1;
x=exp(-10*t);
%figure
subplot(1,2,1)
stem(t,x)
subplot(1,2,2)
plot(t,x)
grid on;
```

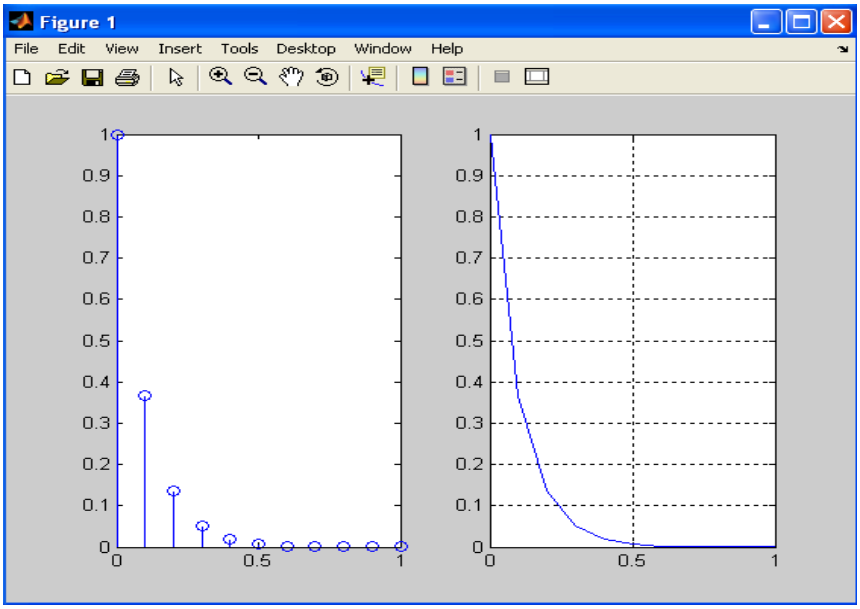


Figure 1.

1. With the help of the code provided above and MATLAB help, write a program to generate any decaying exponential signal. Also label the axes and give your figure a title. (Use ‘xlabel’, ‘ylabel’ and ‘title’ commands. See MATLAB help for syntax.)

```
>> title('exponential decying')
xlabel('x-axis');
ylabel('y-axis');
>>  t=0:0.1:1;
                        x=exp(-10*t);
                        %figure


                      subplot(1,2,1)
                        stem(t,x);title('exponential decying')
xlabel('x-axis');
ylabel('y-axis');
                        subplot(1,2,2)
                        plot(t,x)
                        grid on;title('exponential decying')
xlabel('x-axis');
ylabel('y-axis');
``
```

Introduction to Loops:
Now consider the following code:

```
p=0;
for k = 1:10
   p=p+2;
   disp(p)
end
```

2. What is the output of this program?

Ans: the output of this program is 2,4,6,8,10,12,14,16,18,20.

3. Please explore the following MATLAB functions, and state their purpose in your report.
4.

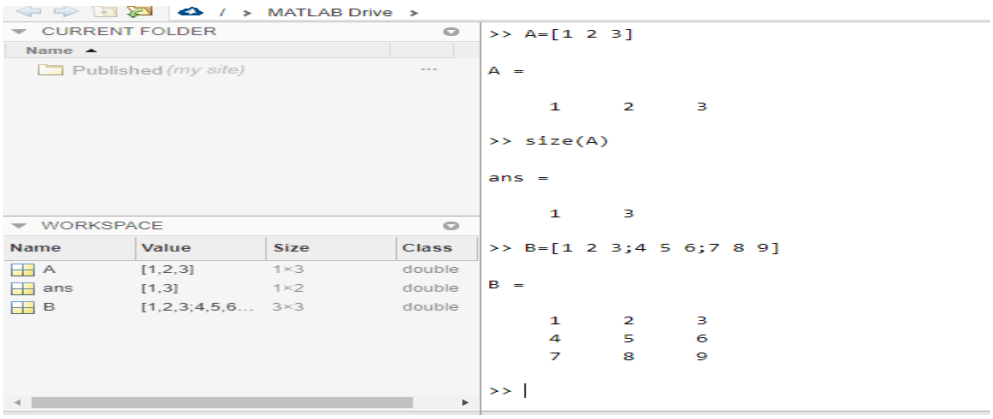
i. **Ver** will show the version and regarding information about MATLAB.	ii. Save Save variables whose names contain digits	iii. Load is the command form of the syntax, for convenient loading from the command line
iv. Exit Used to close the MATLAB.	v. Disp Used to show the results of programs or instructions.	vi. Input Prompt for user input.
vii. Warndlg allows CREATEMODE options that are the same as those offered by MSGBOX	viii. Max Used to maximize the screen.	ix. Sort Used for arrangements.
x. Strcat Used to extract any particular file from the list with proper syntax.	xi. Length Used to define the length elements or matrix	xii. Eye identity matrix with the same data type
xiii. str2num str2num uses EVAL to convert the input argument	xiv. Hold Hold current graph	xv. Eval Execute MATLAB expression in text.
xvi. Tic tic Start a stopwatch timer	xvii. Toc toc Read the stopwatch timer	xviii. Clock Current date and time as date vector
xix. exp Exponential	xx. Acos Inverse cosine, result in radians	xxi. Sind Sine of argument in degrees
xxii. Sinh Sine hyperbolic	xxiii. Findstr Find sequence of characters within string	xxiv. Rand Uniformly distributed pseudorandom numbers
xxv. Ceil Round towards plus infinity	xxvi. Floor floor Round towards minus infinity	xxvii. Round round rounds towards nearest decimal or integer

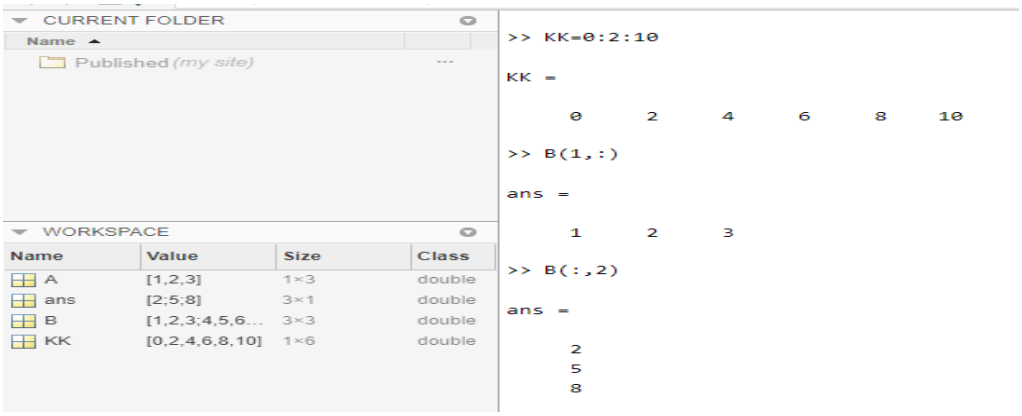
5. What does MATLAB stand for?

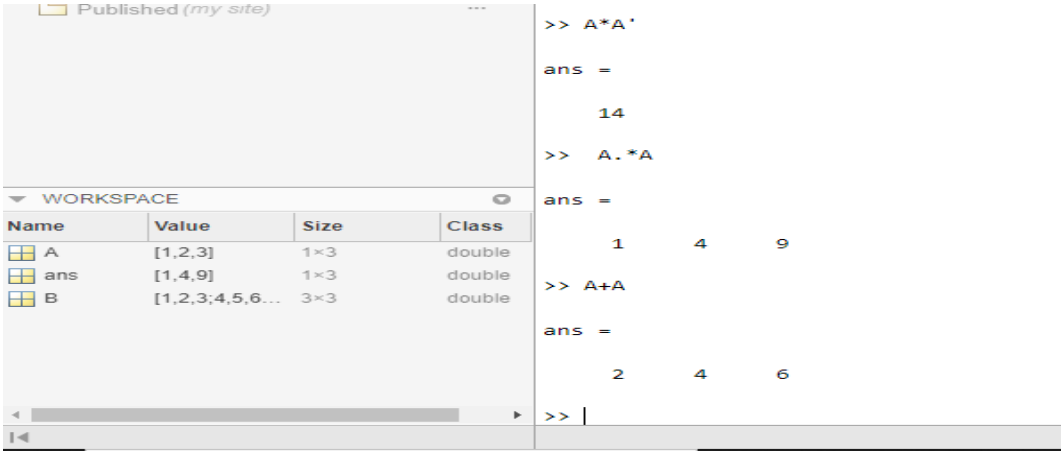
Ans: MATLAB stands for "MATRIX LABORTORY"

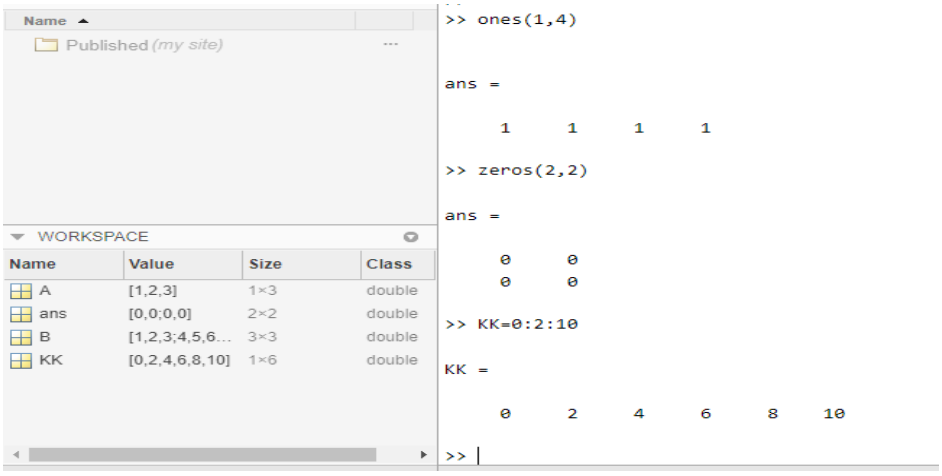
ACTIVITY:

Using Matlab website (https://matlabacademy.mathworks.com/) complete the free Matlab onramp course.

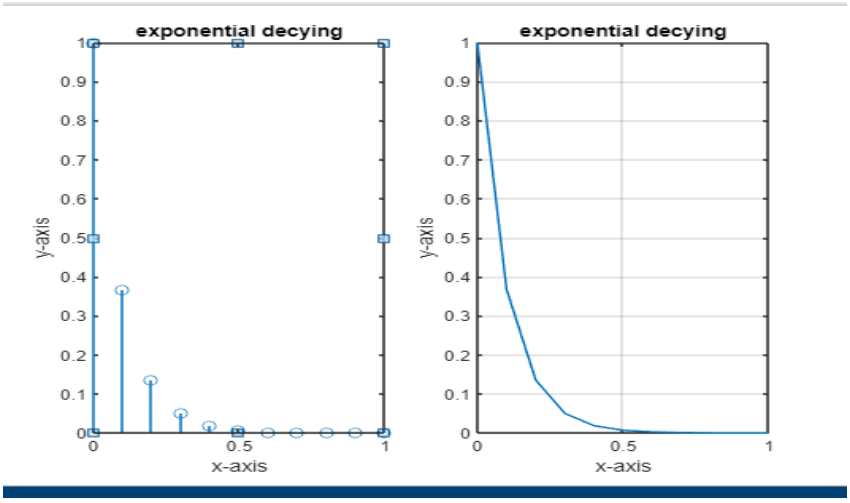









```
>> title('exponential decying')
xlabel('x-axis');
ylabel('y-axis');
>>   t=0:0.1:1;
                              x=exp(-10*t);
                              %figure

                           subplot(1,2,1)
                              stem(t,x);title('exponential decying')
xlabel('x-axis');
ylabel('y-axis');
                              subplot(1,2,2)
                              plot(t,x)
                              grid on;title('exponential decying')
xlabel('x-axis');
ylabel('y-axis');
``
```



MEHRAN UNIVERSITY OF ENGINEERING AND TECHNOLOGY, JAMSHORO
Department of Electronic Engineering
SIGNALS & SYSTEMS (ES-304)

LAB # 02: SIGNAL GENERATION

Name	KARAN VASWANI	Roll #	20ES062
Signature of Lab Tutor		Date	

OBJECTIVE(S)

#	*Topic*	# Of Lectures	CLO	Taxonomy level
1	*Familiarization* of basic commands in MATLAB for signal generation and for plotting the generated signal.	3	4,5	P3,A4

OUTCOME(S)

a. An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.	**PLO5**:Modern Tool Usage
b. An ability to communicate effectively (written/oral)	**PLO10**: Communication

LAB RUBRICS:

Performance Metric	Good (5-4)	Average (3-2)	Poor (1-0)	Score
Use of modern engineering software [PLO5]	Demonstrates knowledge and application of modern engineering software through accurate development and interpretation of computer programs to solve problems.	Demonstrates awareness of modern engineering software through mostly correct development and interpretation of computer programs to solve problems, but may contain minor mistakes or syntax errors	Unable to use modern engineering software to develop or interpret computer programs to solve problems.	
Conducting simulation [PLO5]	Has an excellent simulations skill in the simulator. Always able to make the logical code for the given task.	Has a good simulation skill in the simulator. Always able to make the logical code for the given task.	Has poor simulation skill in the simulator. Unable to make the logical code for the given task.	
Responsiveness to Questions/ Accuracy [PLO10]	Responds well, quick and very accurate all the time.	Generally responsive and accurate most of the times.	Non-responsive at all or the candidate giving only one correct response of viva voce questions.	
Level of understanding of the learned skill [PLO5]	Demonstration of full knowledge of the handout with explanations and elaboration.	At ease with content and able to elaborate and explain to some degree.	No grasp of information. Clearly no knowledge of subject matter. No questions are answered. No interpretation made.	
Documentation: contents and organization [PLO10]	Report well organized, Appropriately sectioned, uses diagram/description when	Report reasonably well documented. May lack some minor aspects.	Report not well organized, lack key aspects.	

	appropriate, important issues clearly stated.			
			Total	

EQUIPMENT/SOFTWARE TOOL:

MATLAB - The Language of Technical Computing.

DISCUSSION AND SETUP:

Analog Signal Processing deals with the processing of continuous time signals whereas in Digital Signal Processing we study discrete-time discrete-valued signals processed by digital computers or other data processing machines. First step in going from analog to digital is 'sampling'. It is the reduction of a continuous signal to a discrete signal. A common example is the conversion of a sound wave (a continuous-time signal) to a sequence of samples (a discrete-time signal). If the amplitude of a DT signal is also made discrete through the process of 'quantization' or rounding off, then it becomes a Digital Signal.
In this lab, we will learn to generate unit sample, unit step and sinusoidal signals using MATLAB.

Unit Sample / Unit Impulse:

$$\delta(n) = \begin{cases} 1 & for \quad n = 0 \\ 0 & \quad n \neq 0 \end{cases}$$

Unit Step:

$$u(n) = \begin{cases} 1 & for \quad n \geq 0 \\ 0 & \quad n \angle 0 \end{cases}$$

Sinusoid:

$$x(n) = A\sin(\omega n + \phi)$$

$$where, \omega = 2\pi f$$

n = integer variable called the sample number,
A = amplitude of the sinusoid,
ω = frequency in radians per sample,
ϕ = phase in radians.

LAB TASKS, OBSERVATIONS & RESULTS:

Generation of Unit Impulse $\delta(n)$**:**
Consider the following code:

```
n=-1:10;
s=[0 1 zeros(1,10)];
stem(n,s);
```

```
xlabel('n');
ylabel('Amplitude');
title('unit sample sequence');
axis([-10 20 0 1]);
```

Its output is as shown in figure 1.

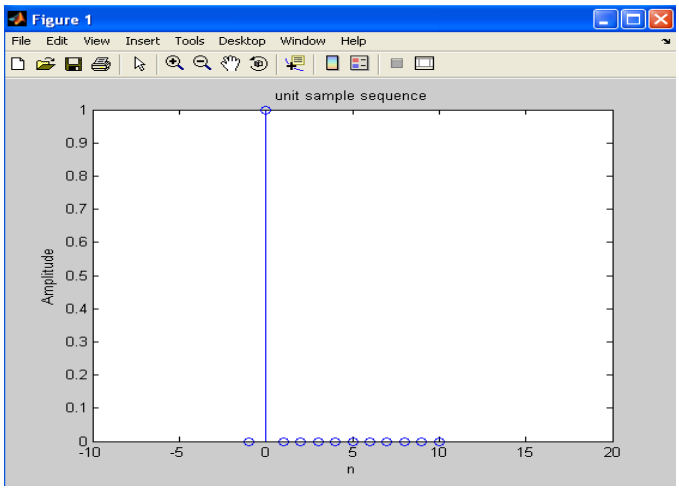
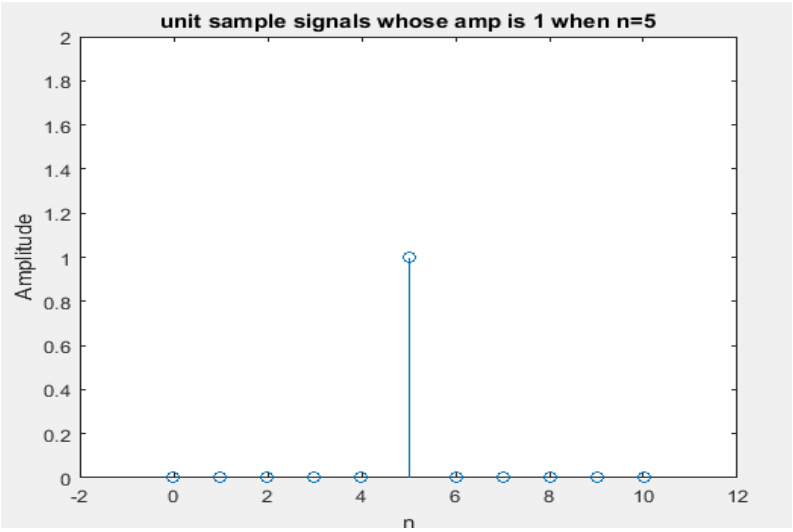


Figure 1.

1. **Write a program to generate $\delta(n-5)$.**

ANSWER:

```
>> n=0:10;
>> s=[zeros(1,5) 1 zeros(1,5)];
>> stem(n,s);
>> xlabel('n');
>> ylabel('Amplitude');
>> title('unit sample signals whose amp is 1 when n=5');
>> axis([-2 12 0 2]);
```



Generation of Unit Step signal $u(n)$:

Consider the following code:

```
                    a=-10:20;
b=[zeros(1,10) ones(1,20)];
                    stem(a,b);
```

Its output is as shown in figure 2.

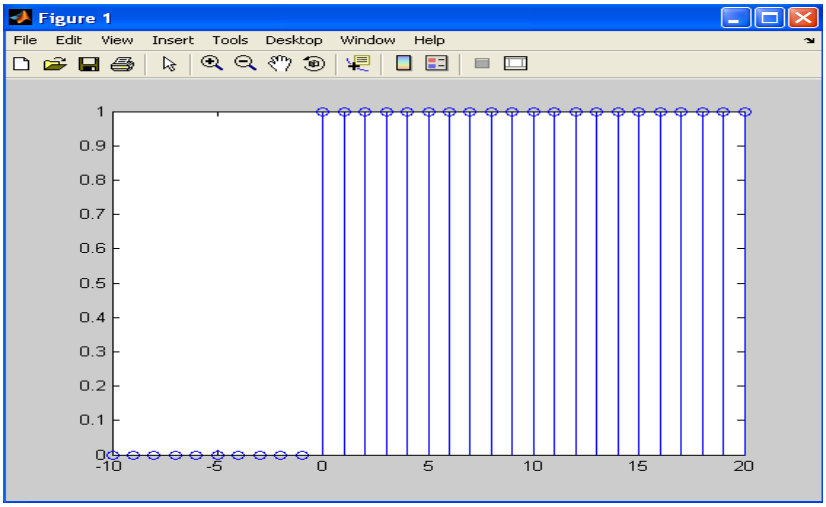
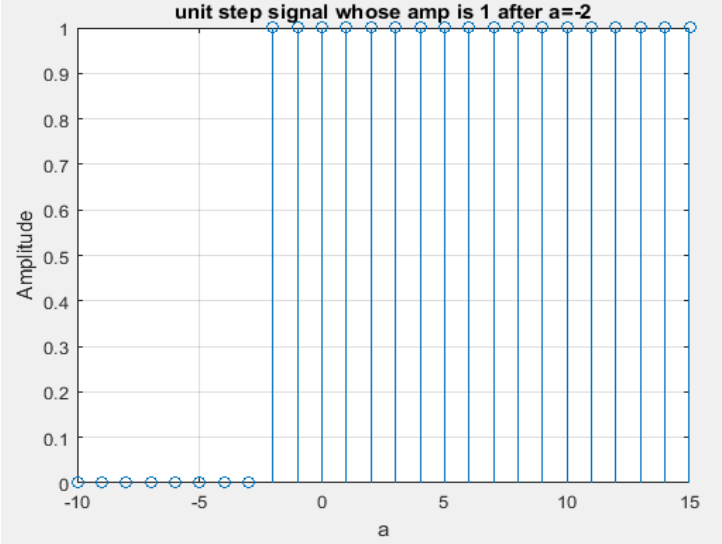


Figure 2.

2. **Write a program to generate $u(n+2)$.**
   (Use 'xlable', 'ylabel', 'title' functions in your code. See MATLAB help for syntax)

ANSWER:

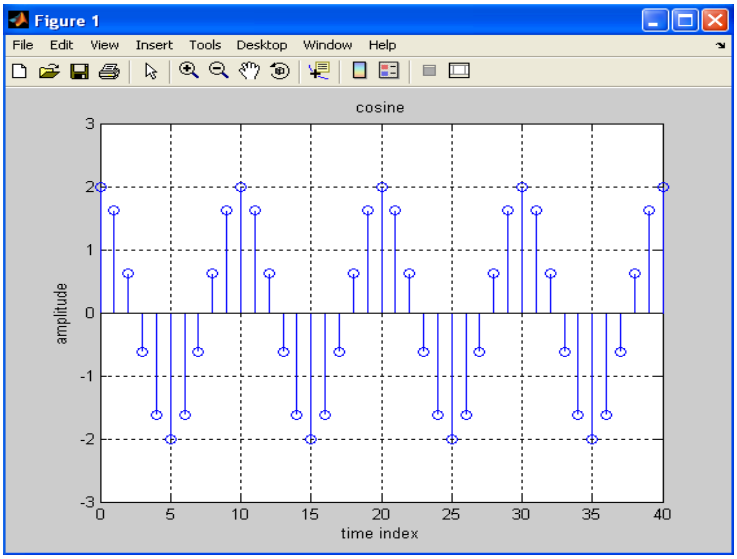
```
>> a=-10:15;
>> b=[zeros(1,8) ones(1,18)];
>> stem(a,b);
>> xlabel('a');
>> ylabel('Amplitude');
>> title('unit step signal whose amp is 1 after a=-2');
>> grid on;
```



Generation of Sinusoidal Signal:
Consider the following code:

```
c=0:40;
freq=.1;
phase=0;
amp=2;
arg=2*pi*freq*c+phase;
ans=amp*cos(arg);
stem(c,ans);
axis([0 40 -3 3]);
grid on;
title('cosine');
xlabel('time index');
ylabel('amplitude');
```

Its output is as shown..



3. **Write a program to generate a sine wave with any suitable frequency. Your code should ask for the amplitude of the sine wave as an input from the user.**
   (Use 'input' function. Please see MATLAB help for syntax)

ANSWER:

```
>> c=0:40; t=0.05; f=1;
>> ang=2*pi*f*c*t;
>> amp=input('plz input the amplitude for the wave');
>> ans=amp*sin(ang);
>> stem(c,ans);
>> xlabel('time axis');
>> ylabel('Amplitude');
>> axis([0 40 -6 6]);
>> title('sine wave');
>> grid on;
```

4. **Please explore the following MATLAB functions and write their purpose in the report.**

i. **Kroneckerdelta**

Set symbolic variable m equal to symbolic variable n and test their equality using kroneckerDelta.

ii. **Heaviside**

Heaviside is the Step function. heaviside(X) is 0 for X < 0 and 1 for X > 0.The value heaviside(0) is 0.5 by default. It can be changed to any value v by the call sympref('HeavisideAtOrigin', v).

5. **Here are some statements that generate a unit impulse, a unit step, a unit ramp, and a unit parabola.**

```
t = (-1:0.01:1)';
impulse = t==0;
unitstep = t>=0;
ramp = t.*unitstep;
quad = t.^2.*unitstep;
```

All of these sequences are column vectors that inherit their shapes from t.
Plot the sequences.

```
plot(t,[impulse unitstep ramp quad])
```

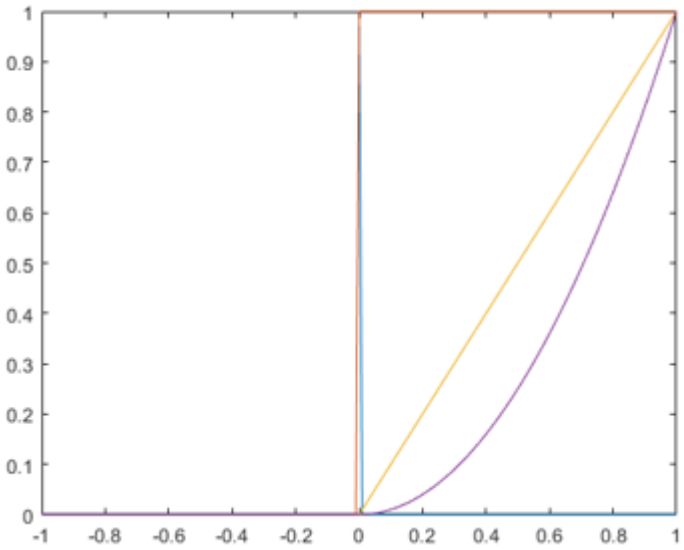


Figure 4.

Please verify the code above and the plots given in figure 4, in MATLAB.

Ans: This result is verified in MATLAB and it provide the same result as illustrated here.

MEHRAN UNIVERSITY OF ENGINEERING AND TECHNOLOGY, JAMSHORO
Department of Electronic Engineering
SIGNALS & SYSTEMS (ES-304)

LAB # 03: INTRODUCTION TO MATLAB GUI DEVELOPMENT ENVIRONMENT

Name	KARAN VASWANI	Roll #	20ES062
Signature of Lab Tutor		Date	

OBJECTIVE(S)

#	*Topic*	# Of Lectures	CLO	Taxonomy level
1	The purpose of this lab is to *introduce* you to GUI development environment of Matlab	3	4,5	P3,A4

OUTCOME(S)

a. An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.	**PLO5**:Modern Tool Usage
b. An ability to communicate effectively (written/oral)	**PLO10**: Communication

LAB RUBRICS:

Performance Metric	Good (5-4)	Average (3-2)	Poor (1-0)	Score
Use of modern engineering software [PLO5]	Demonstrates knowledge and application of modern engineering software through accurate development and interpretation of computer programs to solve problems.	Demonstrates awareness of modern engineering software through mostly correct development and interpretation of computer programs to solve problems, but may contain minor mistakes or syntax errors	Unable to use modern engineering software to develop or interpret computer programs to solve problems.	
Level of understanding of the learned skill	Demonstration of full knowledge of the handout with	At ease with content and able to elaborate and explain to some degree.	No grasp of information. Clearly no knowledge of subject matter. No questions	

[PLO5]	explanations and elaboration.		are answered. No interpretation made.	
Conducting simulation [PLO5]	Has an excellent simulations skill in the simulator. Always able to make the logical code for the given task.	Has a good simulation skill in the simulator. Always able to make the logical code for the given task.	Has poor simulation skill in the simulator. Unable to make the logical code for the given task.	
Responsiveness to Questions/ Accuracy [PLO10]	Responds well, quick and very accurate all the time.	Generally responsive and accurate most of the times.	Non-responsive at all or the candidate giving only one correct response of viva voce questions.	
Documentation: contents and organization [PLO10]	Report well organized, Appropriately sectioned, uses diagram/dscription when appropriate, important issues clearly stated.	Report reasonably well documented. May lack some minor aspects.	Report not well organized, lack key aspects.	
			Total	

EQUIPMENT/SOFTWARE TOOL:

MATLAB - The Language of Technical Computing.

DISCUSSION AND SETUP:

GUIDE, the MATLAB graphical user interface development environment, provides a set of tools for creating graphical user interfaces (GUIs). These tools greatly simplify the process of designing and building GUIs. You can use the GUIDE tools to:

- **Lay out the GUI.**
  Using the GUIDE Layout Editor, you can lay out a GUI easily by clicking and dragging GUI components—such as panels, buttons, text fields, sliders, menus, and so on—into the layout area. GUIDE stores the GUI layout in a FIG-file.

- **Program the GUI.**
  GUIDE automatically generates an M-file that controls how the GUI operates. The M-file initializes the GUI and contains a framework for the most commonly used callbacks for

each component—the commands that execute when a user clicks a GUI component. Using the M-file editor, you can add code to the callbacks to perform the functions you want.

Creating a New GUI using GUIDE:

EXAMPLE#1:

Steps:

1. Click on the GUIDE icon in the toolbar, you will see the window, as shown in figure 1:

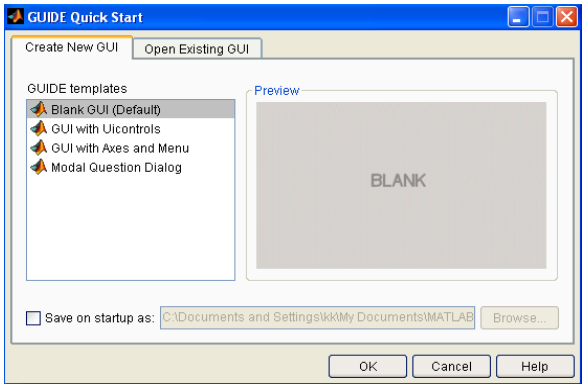


Figure 1.

2. Select ‘Blank GUI’ and click ‘ok’. The following screen will appear (figure 2):

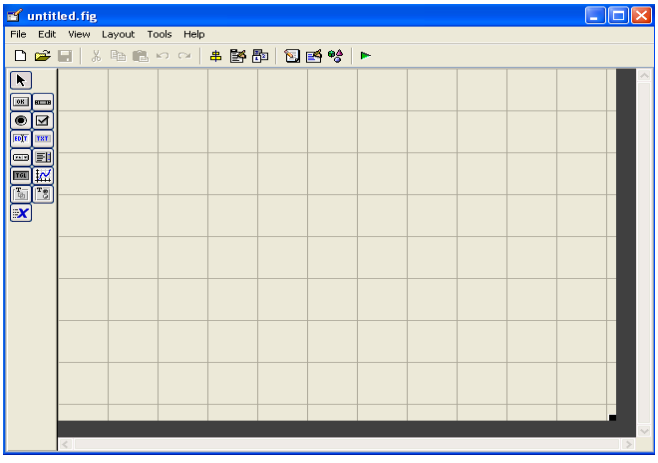


Figure 2.

3. Design your GUI with the help of panel provided to the left, a sample GUI is shown in figure 3:

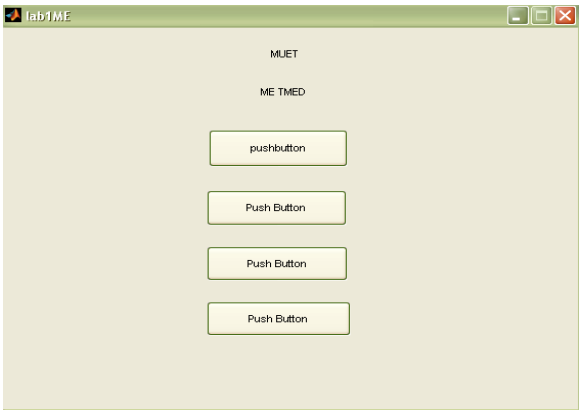


Figure 3.

4. Double click on an item opens 'inspector', shown in figure 4, which can be used to alter the properties of that particular entity.

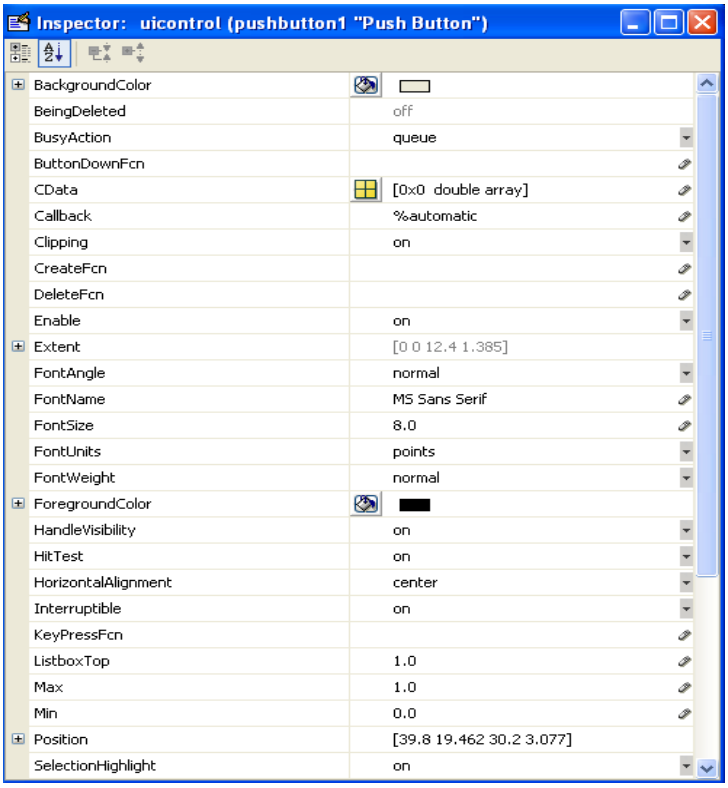


Figure 4.

5. Save your designed GUI.
6. Alter the code to cater for the changes that you want to make.

MATLAB code of this GUI (Changes made by me are commented):

```
function varargout = GUI(varargin)
% GUI M-file for GUI.fig
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @GUI_OpeningFcn, ...
                   'gui_OutputFcn',  @GUI_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end


if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
% --- Executes just before GUI is made visible.
function GUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
```

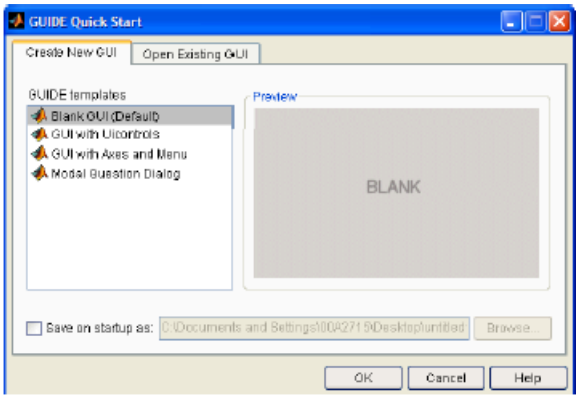
```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUI (see VARARGIN)
% Choose default command line output for GUI
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);
% UIWAIT makes GUI wait for user response (see UIRESUME)
% uiwait(handles.figure1);
% --- Outputs from this function are returned to the command line.
function varargout = GUI_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
varargout{1} = handles.output;
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%--------------------------------------
% my changes
inputdlg('Please enter your name');
%--------------------------------------
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
```

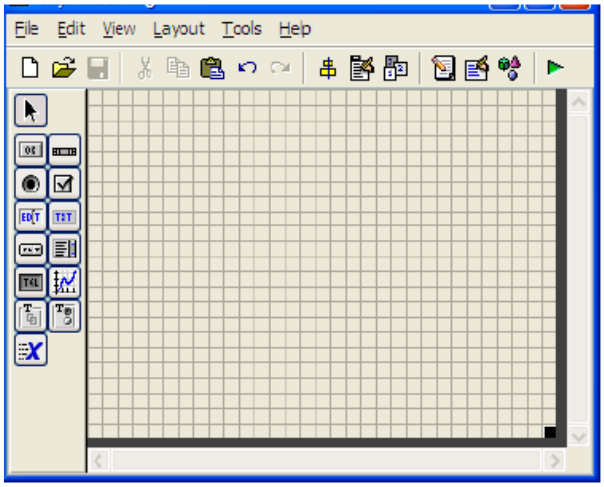
```
%--------------------------------------
% my changes
msgbox('hello there :)')
%--------------------------------------
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%--------------------------------------
% my changes
msgbox('use pushbutton 4 to exit')
%--------------------------------------
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
%--------------------------------------
% my changes
clc
clear all
close all
%--------------------------------------
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
 ---------------------------------------------------------------
```

EXAMPLE#2:

>> guide

Select the blank gui





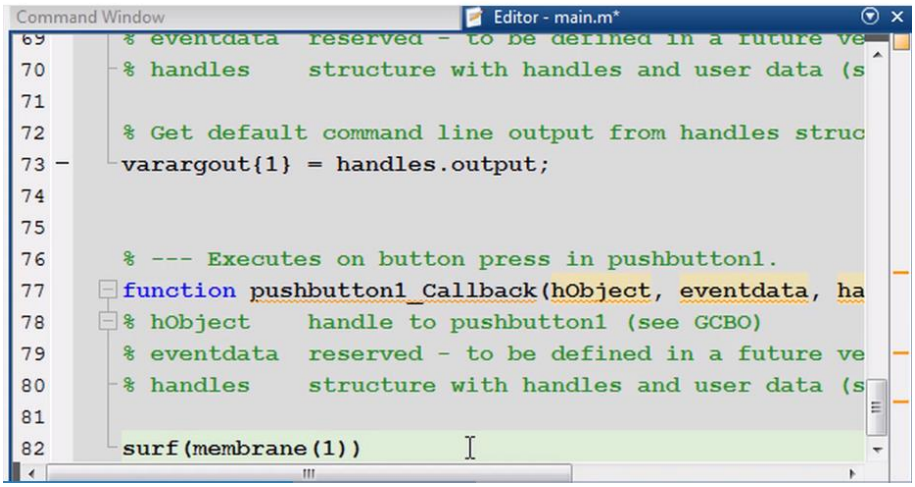
Select an axes.

Select a push button.

Save the gui and run.

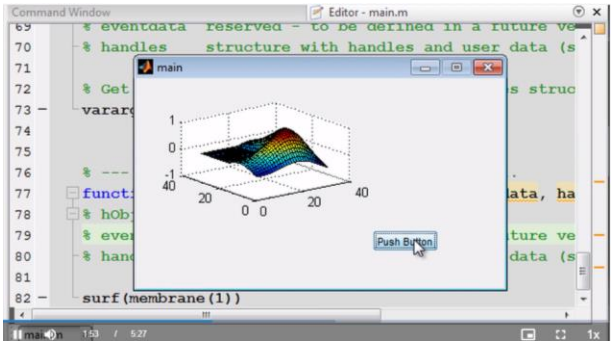
This gui will do nothing unless you add your programming in it.

So lets say I want surf plot membrane. So I will write code in the callback of push button

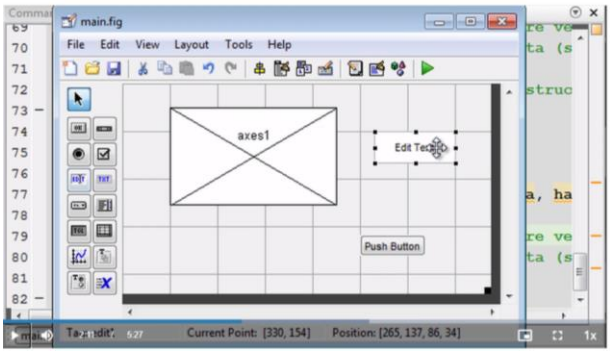


This will use membrane function and automatically generate its data and give us its surf plot.

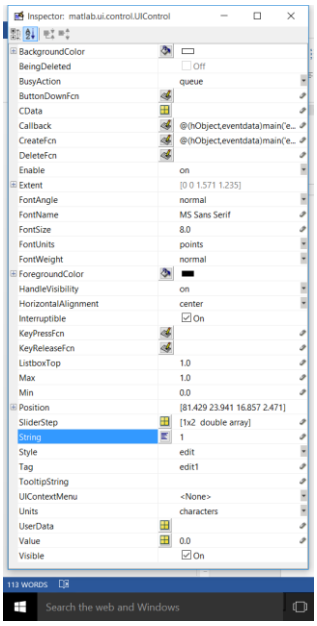
Lets save and run ,and now we see this when we push the button.



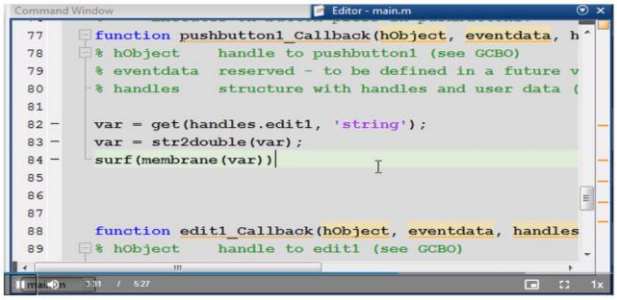
Let's say now, I want to get some data from gui. So to do that , say we drop an edit box in gui



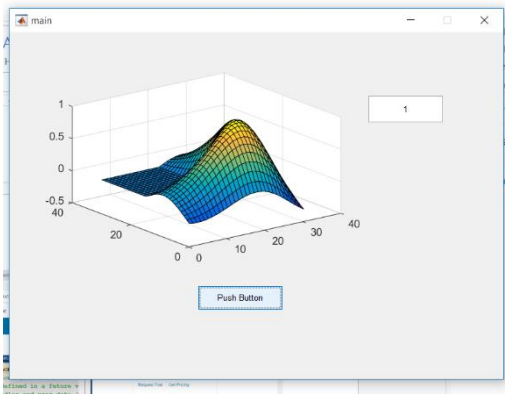
Now double click on it and we can see its properties and all.

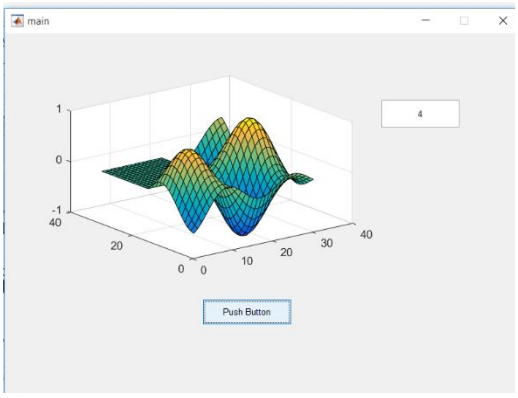


Make the string =1 and lets say I want to write data in the edit box and get surf plot according to that. For that I must write



Now run it again





So now we can see the plot is changing w.r.t the number in the edit box.

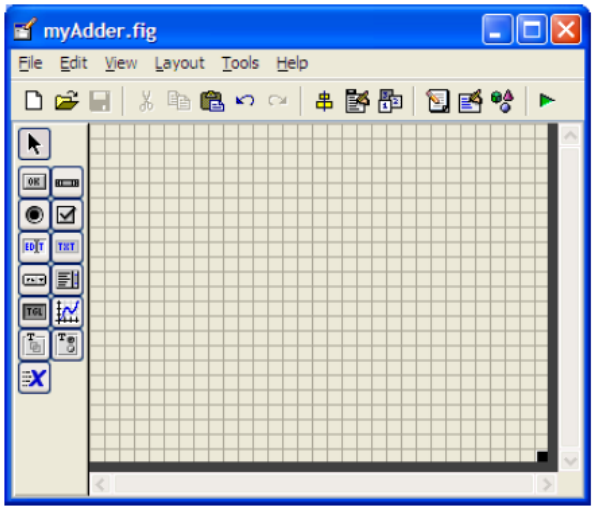
EXAMPLE#3:

A simple addition GUI.

Open guide.

Select blank GUI.

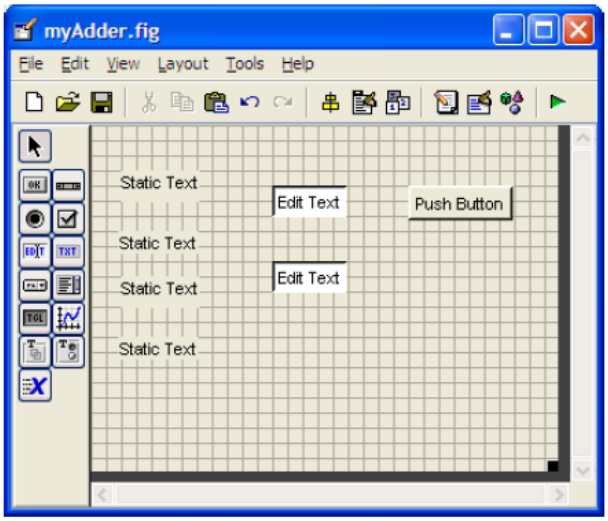
You should see this screen now.



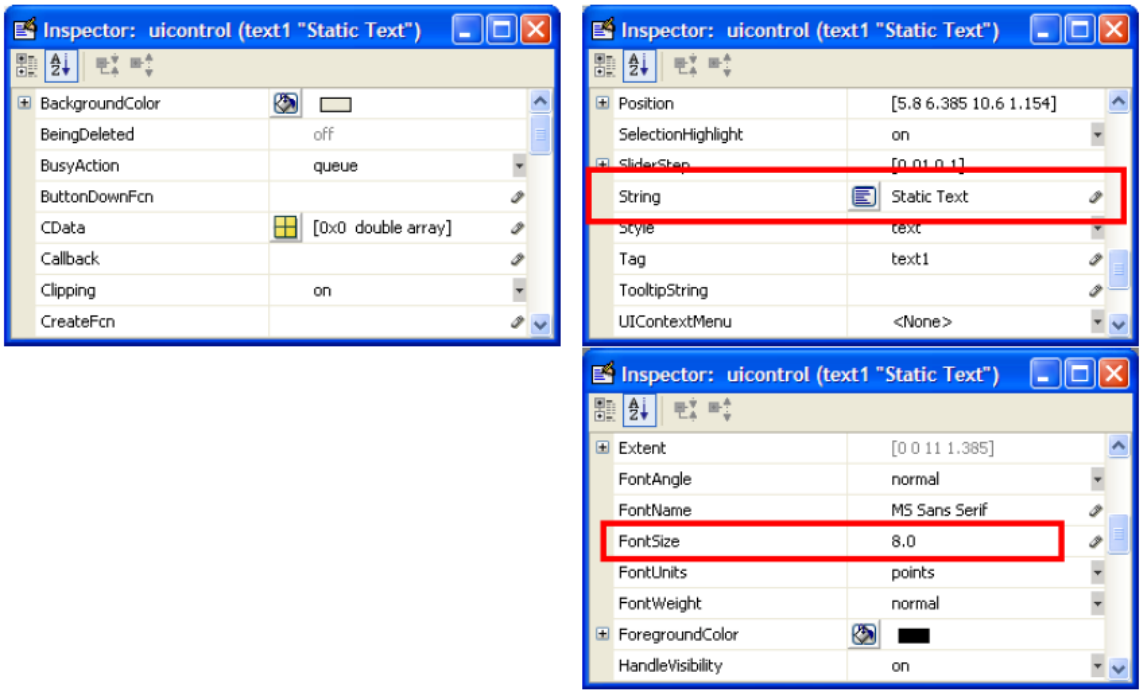
For the Adder gui we will need:

1. Two edit text components
2. Four static text components
3. One push button component

Drag all the required components and drop on the blank gui. You should have this:

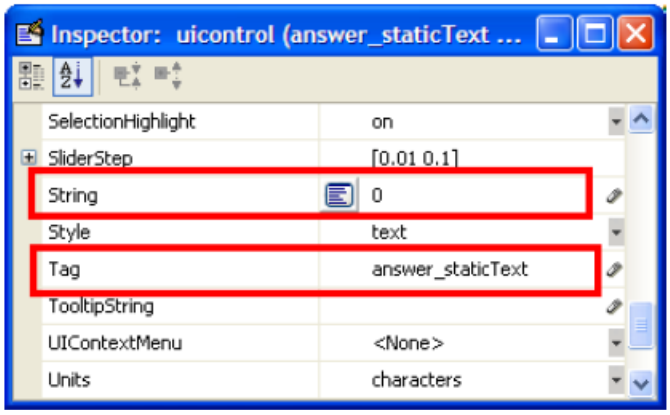


Edit the properties of these components by double clicking on them.

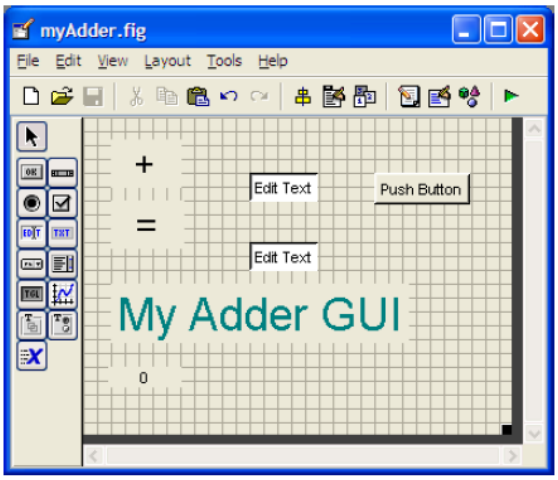


Do the same for the next Static Text component, but instead of changing the String parameter to +, change it to =, and another it to MyAdderGUI.

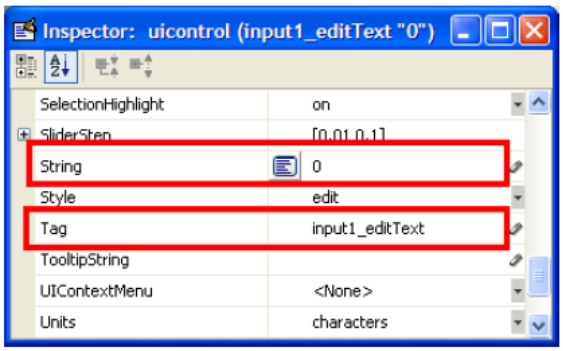
For Static Text Component 0, modify the Tag parameter to answer_staticText.



You should have something the looks like the following.

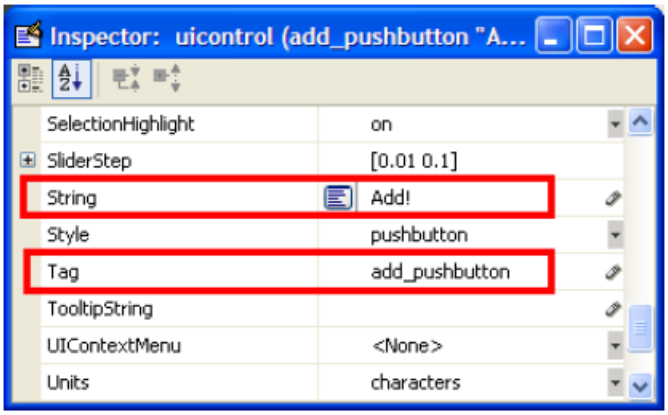


Modify the Edit Text components as shown in figure below

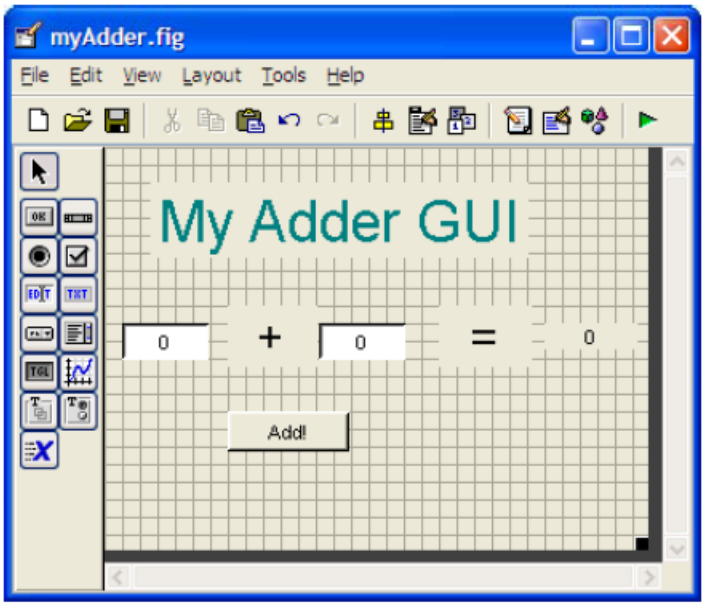


The second Edit Text component, set the stringparameter to 0 and tag to input2_editText.

Similarly modify the pushbutton component as shown below:



You should have something like this:



Writing the code:

```
3. function input1_editText_Callback(hObject, eventdata, handles)
4. % hObject    handle to input1_editText (see GCBO)
5. % eventdata  reserved - to be defined in a future version of MATLAB
6. % handles    structure with handles and user data (see GUIDATA)
7.
8. % Hint: get(hObject,'String') returns contents of input1_editText as text
9. %       str2double(get(hObject,'String')) returns contents of
10.%       input1_editText as a double
```

Add the following block of code :

```
%store the contents of input1_editText as a string. if the string
%is not a number then input will be empty
input = str2num(get(hObject,'String'));

%checks to see if input is empty. if so, default input1_editText to zero
if (isempty(input))
    set(hObject,'String','0')
end
```

Add the same block of code to input2_editText_Callback

Now we need to edit the push button callback

```
13.% --- Executes on button press in add_pushbutton.
14.function add_pushbutton_Callback(hObject, eventdata, handles)
15.% hObject    handle to add_pushbutton (see GCBO)
16.% eventdata  reserved - to be defined in a future version of MATLAB
17.% handles    structure with handles and user data (see GUIDATA)
```

Here is the code that we will add to its callback

```
a = get(handles.input1_editText,'String');
b = get(handles.input2_editText,'String');
% a and b are variables of Strings type, and need to be converted
% to variables of Number type before they can be added together

total = str2num(a) + str2num(b);
c = num2str(total);
% need to convert the answer back into String type to display it
set(handles.answer_staticText,'String',c);
```

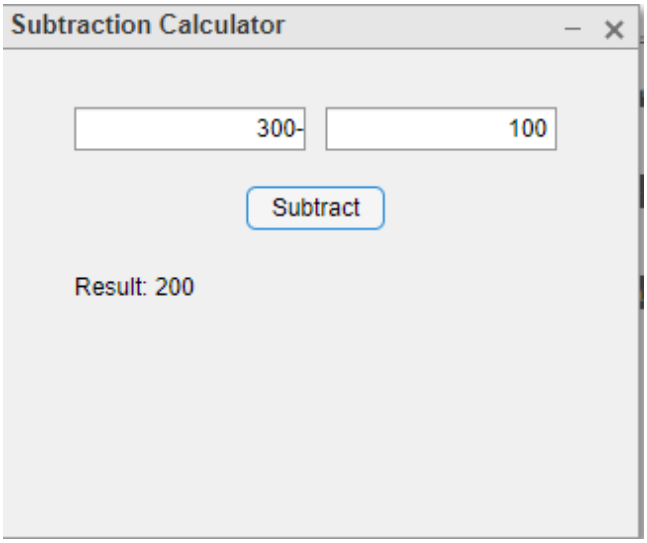
Run the GUI and check the Adder.

LAB TASK, OBSERVATIONS & RESULTS:

Design GUI for subtracting two numbers.

Answer:

```
% Create the GUI figure window
fig = uifigure('Position', [100 100 250 150], 'Name', 'Subtraction Calculator');

% Create the input fields for the two numbers
num1 = uieditfield(fig, 'numeric', 'Position', [40 100 80 22], 'Value', 0);
num2 = uieditfield(fig, 'numeric', 'Position', [130 100 80 22], 'Value', 0);

% Create a text label for the operation
uilabel(fig, 'Position', [115 100 20 22], 'Text', '-');

% Create a button to perform the subtraction
btn = uibutton(fig, 'push', 'Position', [90 60 70 22], 'Text', 'Subtract');

% Create a text label for the result
resultLabel = uilabel(fig, 'Position', [40 20 170 22], 'Text', 'Result:');

% Callback function for the button
btn.ButtonPushedFcn = @(btn,event) subtractNumbers(num1.Value, num2.Value, resultLabel);

function subtractNumbers(num1, num2, resultLabel)
    % Subtract the two numbers and update the result label
    result = num1 - num2;
    resultLabel.Text = sprintf('Result: %g', result);
end
```



REVIEW QUESTIONS:

1. What is GUIDE?
   GUIDE stands for Graphical User Interface Development Environment. It provides the tools to design user interfaces and create custom apps.

2. What is the function of Handles?
   A function handle is a MATLAB® data type that represents a function. A typical use of function handles is to pass a function to another function. For example, you can use function handles as input arguments to functions that evaluate mathematical expressions over a range of values.

3. Explain the function of str2num command.
   X = str2num(txt) converts a character array or string scalar to a numeric matrix. The input can include spaces, commas, and semicolons to indicate separate elements. If str2num cannot parse the input as numeric values, then it returns an empty matrix.

MEHRAN UNIVERSITY OF ENGINEERING AND TECHNOLOGY, JAMSHORO
Department of Electronic Engineering
SIGNALS & SYSTEMS (ES-304)

LAB # 04: INTRODUCTION TO SIMULINK

Name	Karan vaswani	Roll #	20ES062
Signature of Lab Tutor		Date	

OBJECTIVE(S)

#	*Topic*	# Of Lectures	CLO	Taxonomy level
1	The purpose of this lab is to *introduce* you to Simulink.	3	4,5	P3,A4

OUTCOME(S)

a. An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.	**PLO5**:Modern Tool Usage
b. An ability to communicate effectively (written/oral)	**PLO10**: Communication

LAB RUBRICS:

Performance Metric	Good (5-4)	Average (3-2)	Poor (1-0)	Score
Use of modern engineering software [PLO5]	Demonstrates knowledge and application of modern engineering software through accurate development and interpretation of computer programs to solve problems.	Demonstrates awareness of modern engineering software through mostly correct development and interpretation of computer programs to solve problems, but may contain minor mistakes or syntax errors	Unable to use modern engineering software to develop or interpret computer programs to solve problems.	
Level of understanding of the learned skill [PLO5]	Demonstration of full knowledge of the handout with explanations and elaboration.	At ease with content and able to elaborate and explain to some degree.	No grasp of information. Clearly no knowledge of subject matter. No questions are answered. No interpretation made.	
Conducting simulation [PLO5]	Has an excellent simulations skill in the simulator. Always able to make the logical code for the given task.	Has a good simulation skill in the simulator. Always able to make the logical code for the given task.	Has poor simulation skill in the simulator. Unable to make the logical code for the given task.	
Responsiveness to Questions/ Accuracy [PLO10]	Responds well, quick and very accurate all the time.	Generally responsive and accurate most of the times.	Non-responsive at all or the candidate giving only one correct response of viva voce questions.	
Documentation: contents and organization [PLO10]	Report well organized, Appropriately sectioned, uses diagram/dscription when appropriate, important issues clearly stated.	Report reasonably well documented. May lack some minor aspects.	Report not well organized, lack key aspects.	
			Total	

EQUIPMENT/SOFTWARE TOOL:

MATLAB - The Language of Technical Computing.

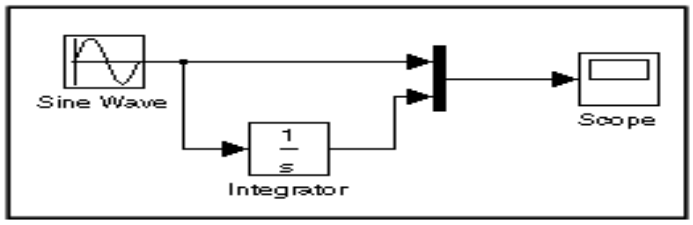
DISCUSSION AND SETUP:

Simulink is a software package for modeling, simulating, and analyzing dynamic systems. It supports linear and nonlinear systems, modeled in continuous time, sampled time, or a hybrid of the two. One can easily build, analyze and visualize models.

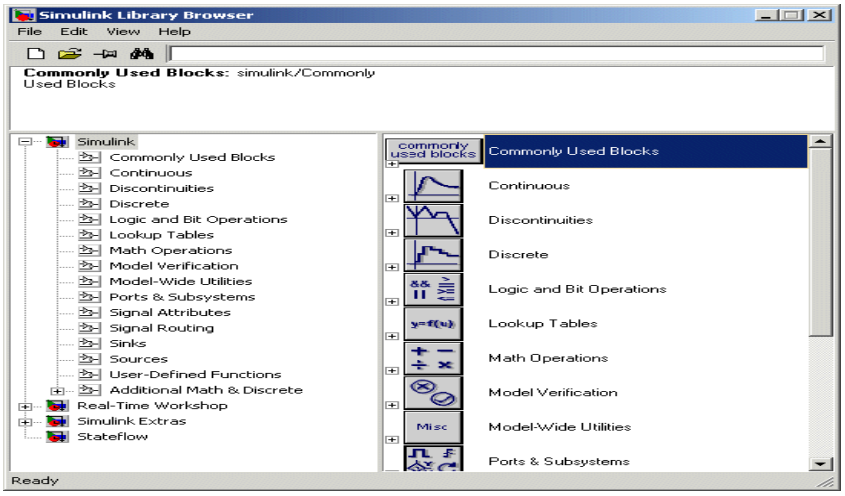
Thousands of engineers around the world using it to model and solve real problems. Simulink provides a graphical user interface (GUI) for building models as block diagrams, using click-and-drag mouse operations. With this interface, you can draw the models just as you would with pencil and paper. Knowledge of this tool will serve you well throughout your professional career.

Building a model

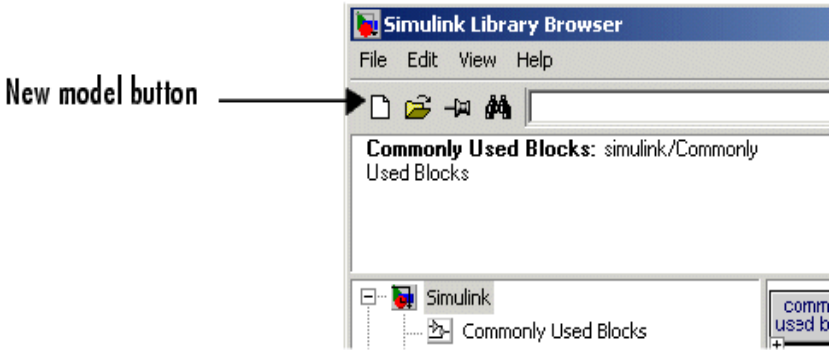
This example shows you how to build a model using many of the model-building commands and actions you will use to build your own models. The model integrates a sine wave and displays the result along with the sine wave. The block diagram of the model looks like this.



To create the model, first enter Simulink in the MATLAB Command Window the Simulink Library Browser appears.



To create a new model click the New Model button on the Library Browser's toolbar. Simulink opens a new model window.

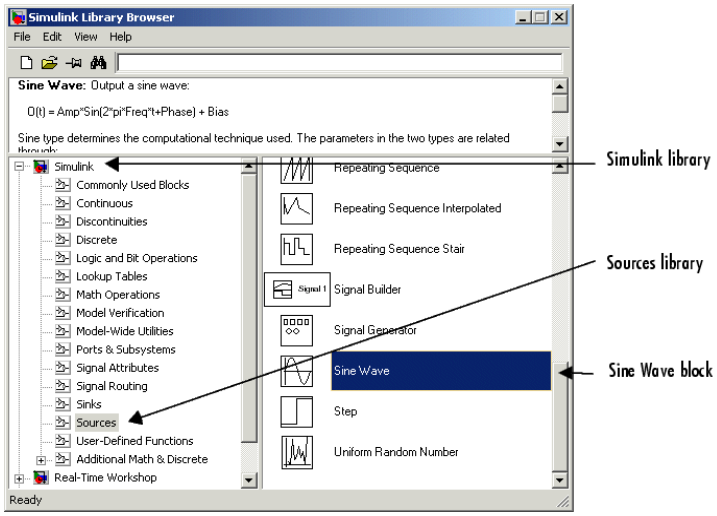


To create this model, you need to copy blocks into the model from the following Simulink block libraries:

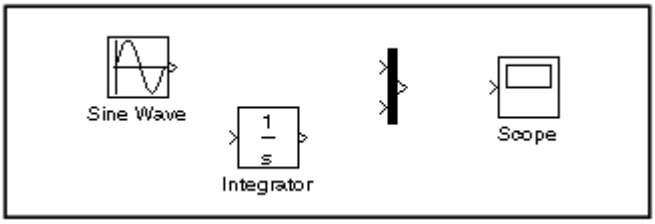
- Sources library (the Sine Wave block)
- Sinks library (the Scope block)
- Continuous library (the Integrator block)
- Signal Routing library (the Mux block)

To copy the Sine Wave block from the Library Browser, first expand the Library Browser tree to display the blocks in the Sources library. Do this by clicking the Sources node to display the Sources library blocks. Finally, click the Sine Wave node to select the Sine Wave block.

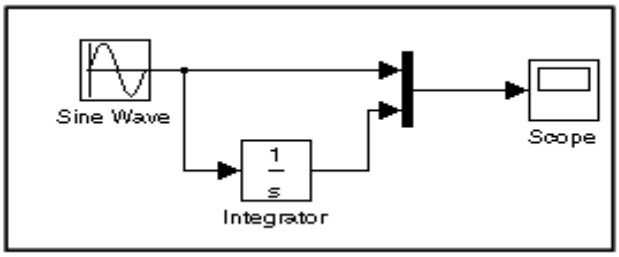
Here is how the Library Browser should look after you have done this.



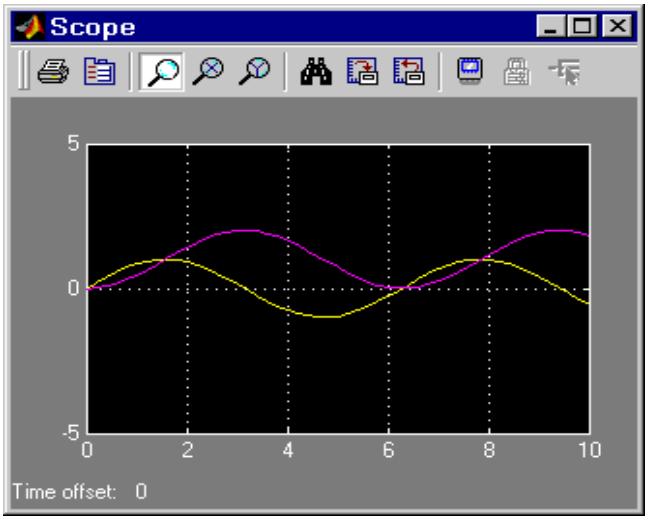
Now drag a copy of the Sine Wave block from the browser and drop it in the model window. Copy the rest of the blocks in a similar manner from their respective libraries into the model window. With all the blocks copied into the model window, the model should look something like this.



Now it's time to connect the blocks. Connect the Sine Wave block to the top input port of the Mux block. Similarly connect all the components as shown in the figure. When all the block are connected with each other your model should look something like this.



Now double-click the Scope block to open its display window. Finally, choose **Start** from the **Simulation** menu and watch the simulation output on the Scope.

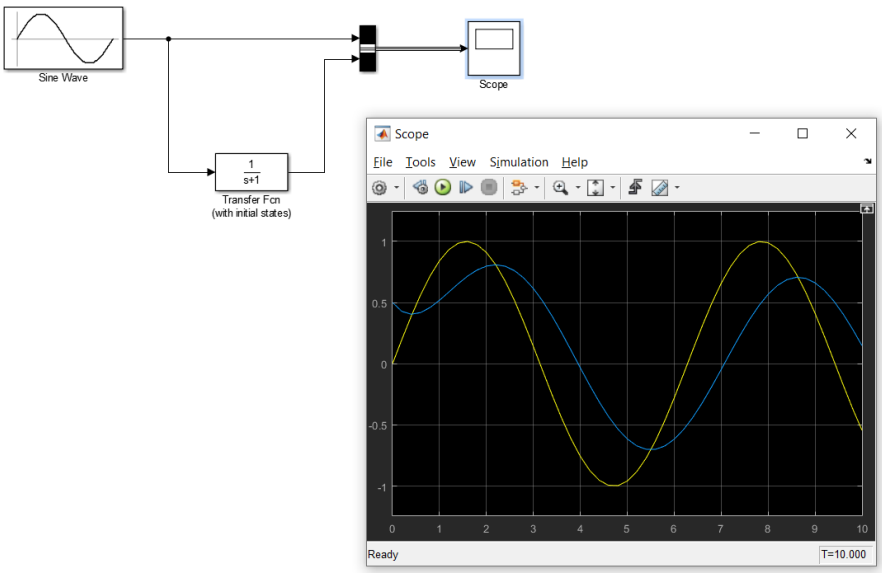


LAB TASK, OBSERVATIONS & RESULTS:

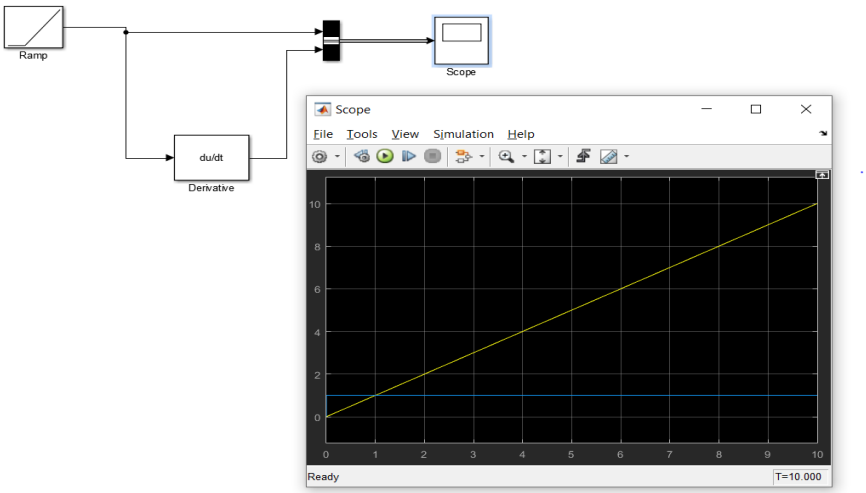
Perform the indicated tasks using SIMULINK and copy the block diagrams and resultant waveforms in your answer book.

(a) Find the step response of the following transfer function.

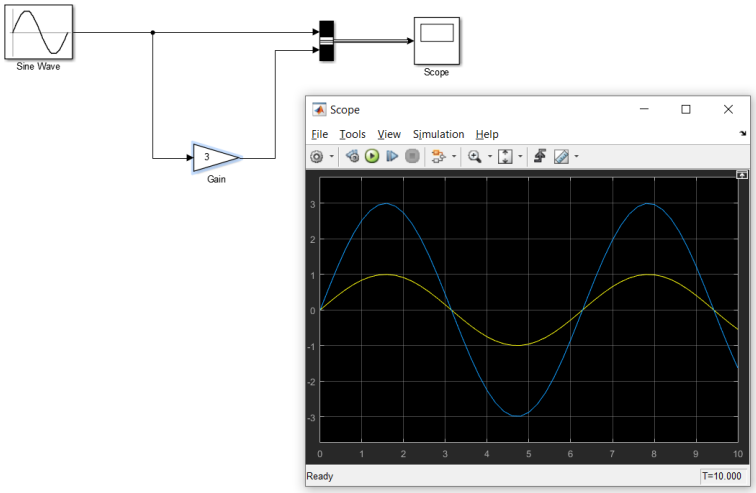
$$G(S) = \frac{1}{S+1}$$



(b) Differentiate the ramp function and check the result on the scope.



(c) Amplify sine wave using gain block.



REVIEW QUESTIONS:

Q:1 What is Simulink?
ANS: Simulink is the platform for Model-Based Design that supports system-level design, simulation, automatic code generation, and continuous test and verification of embedded systems. Key capabilities include: A graphical editor for modeling all components of a system

Q:2 Where is J-K Flip flop in the Simulink library?
ANS: J-K Flip Flop is included in Simulink Extras.

MEHRAN UNIVERSITY OF ENGINEERING AND TECHNOLOGY, JAMSHORO
Department of Electronic Engineering
SIGNALS & SYSTEMS (ES-304)

LAB # 05: CONVOLUTION

Name	Karan vaswani	Roll #	20ES062
Signature of Lab Tutor		Date	

OBJECTIVE(S)

#	*Topic*	# Of Lectures	CLO	Taxonomy level
1	To *demonstrate*, compute and plot Convolution	3	4,5	P3,A4

OUTCOME(S)

• An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.	**PLO5**:Modern Tool Usage
• An ability to communicate effectively (written/oral)	**PLO10**: Communication

LAB RUBRICS:

Performance Metric	Good (5-4)	Average (3-2)	Poor (1-0)	Score
Use of modern engineering software [PLO5]	Demonstrates knowledge and application of modern engineering software through accurate development and interpretation of computer programs to solve problems.	Demonstrates awareness of modern engineering software through mostly correct development and interpretation of computer programs to solve problems, but may contain minor mistakes or syntax errors	Unable to use modern engineering software to develop or interpret computer programs to solve problems.	
Level of understanding of the learned skill [PLO5]	Demonstration of full knowledge of the handout with explanations and elaboration.	At ease with content and able to elaborate and explain to some degree.	No grasp of information. Clearly no knowledge of subject matter. No questions are answered. No interpretation made.	
Conducting simulation [PLO5]	Has an excellent simulations skill in the simulator. Always able to make the logical code for the given task.	Has a good simulation skill in the simulator. Always able to make the logical code for the given task.	Has poor simulation skill in the simulator. Unable to make the logical code for the given task.	
Responsiveness to Questions/ Accuracy [PLO10]	Responds well, quick and very accurate all the time.	Generally responsive and accurate most of the times.	Non-responsive at all or the candidate giving only one correct response of viva voce questions.	

Documentation: contents and organization [PLO10]	Report well organized, Appropriately sectioned, uses diagram/dscription when appropriate, important issues clearly stated.	Report reasonably well documented. May lack some minor aspects.	Report not well organized, lack key aspects.	
			Total	

EQUIPMENT/SOFTWARE TOOL:

MATLAB - The Language of Technical Computing.

DISCUSSION AND SETUP:

The response y(n) of the LTI system as a function of the input signal x(n) and the unit sample (impulse) response h(n) is a convolution sum between x(n) & h(n). The input x(n) is convolved with the impulse response h(n) to yield the output y(n).

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k)$$

The convolution of the equation:

$$y(n) = h(n) \otimes x(n)$$

is implemented in MATLAB by the command, "conv", provided the two sequences to be convolved are of finite length. For example, the output sequence of a finite impulse response (FIR) system can be computed by convolving its impulse response with the given finite length input sequence.

LAB TASK, OBSERVATIONS & RESULTS:

Example1 of Convolution in MATLAB:

Consider the following code:

```
%convolution
h=[5 4 3 2 1]; % first sigal
x=[1 1 1 1 1]; % second signal
c=conv(h,x);   % convolution result stored in variable 'c'
m=length(c)-1;
n=0:1:m;
stem(n,c);     % plot
grid on;
```

Its Output is as shown in figure 1:

n	0	1	2	3	4
h(n)	4	2	-1	3	-2
x(n)	-4	1	3	7	4

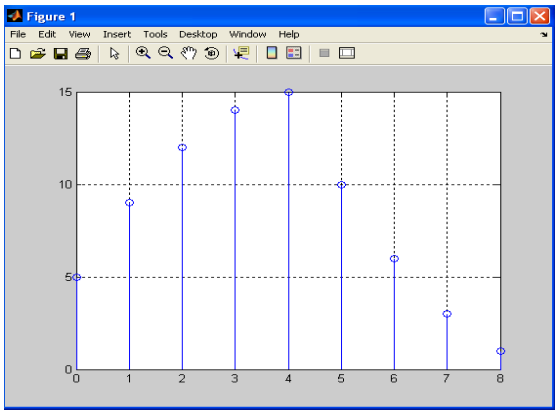


Figure 1.

Example2: Repeat Example 1 with a = [1 2 1 –1] and b = [1 2 3 1]

Code:

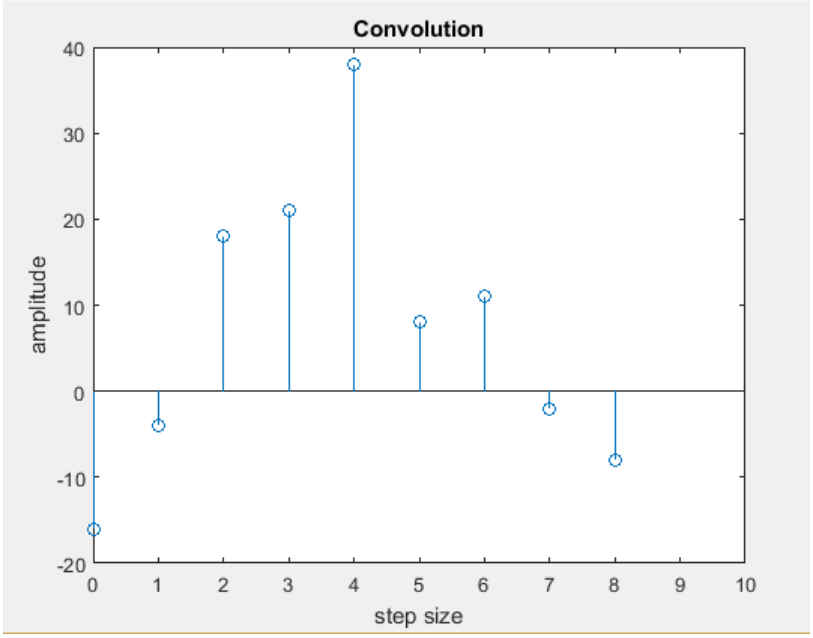
```
a = input('Type in the first sequence =');
b = input('Type in the second sequence =');
c = conv(a,b);
M = length(c)-1;
n = 0:1:M;
disp('output sequence =');
disp(c)
stem(n,c)
xlabel('Time index n')
ylabel('Amplitude')
```

1. Write a MATLAB code and calculate the convolution result of the following 2 sequences. Label your plot using ‘xlabel’, ‘ylable’ and ‘title’ commands.

Answer:

```
h=[4 2 -1 3 -2];
x=[-4 1 3 7 4];
c=conv(h,x);
m=length(c)-1;
n=0:1:m;
stem(n,c);
```

```
xlabel('step size');
ylabel('amplitude');
title('Convolution');
axis([0 10 -20 40]);
```



2. State practical applications of convolution.

Answer:

a) Image processing
b) Signal filtering
c) Polynomial multiplication
d) Audio processing
e) Artificial intelligence
f) Optics
g) Statistics
h) Probability
i) Signal processing
j) Spectroscopy

MEHRAN UNIVERSITY OF ENGINEERING AND TECHNOLOGY, JAMSHORO
Department of Electronic Engineering
SIGNALS & SYSTEMS (ES-304)

LAB # 06: EVEN & ODD SIGNALS

Name	Karan vaswani	Roll #	20ES062
Signature of Lab Tutor		Date	

OBJECTIVE(S)

#	*Topic*	# Of Lectures	CLO	Taxonomy level
1	*Creating* even and odd signals using matlab	3	4,5	P3,A4

OUTCOME(S)

• An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.	**PLO5**:Modern Tool Usage
• An ability to communicate effectively (written/oral)	**PLO10**: Communication

LAB RUBRICS:

Performance Metric	Good (5-4)	Average (3-2)	Poor (1-0)	Score
Use of modern engineering software [PLO5]	Demonstrates knowledge and application of modern engineering software through accurate development and interpretation of computer programs to solve problems.	Demonstrates awareness of modern engineering software through mostly correct development and interpretation of computer programs to solve problems, but may contain minor mistakes or syntax errors	Unable to use modern engineering software to develop or interpret computer programs to solve problems.	
Level of understanding of the learned skill [PLO5]	Demonstration of full knowledge of the handout with explanations and elaboration.	At ease with content and able to elaborate and explain to some degree.	No grasp of information. Clearly no knowledge of subject matter. No questions are answered. No interpretation made.	
Conducting simulation [PLO5]	Has an excellent simulations skill in the simulator. Always able to make the logical code for the given task.	Has a good simulation skill in the simulator. Always able to make the logical code for the given task.	Has poor simulation skill in the simulator. Unable to make the logical code for the given task.	
Responsiveness to Questions/ Accuracy [PLO10]	Responds well, quick and very accurate all the time.	Generally responsive and accurate most of the times.	Non-responsive at all or the candidate giving only one correct response of viva voce questions.	
Documentation: contents and organization [PLO10]	Report well organized, Appropriately sectioned, uses diagram/dscription when appropriate,	Report reasonably well documented. May lack some minor aspects.	Report not well organized, lack key aspects.	

	important issues clearly stated.			
			Total	

EQUIPMENT/SOFTWARE TOOL:

MATLAB - The Language of Technical Computing.

DISCUSSION AND SETUP:

An even function is defined as g(t) = g(-t) or
g[n] = g[-n].

An odd function is defined as g(t) = -g(-t) or
g[n] = -g[-n]

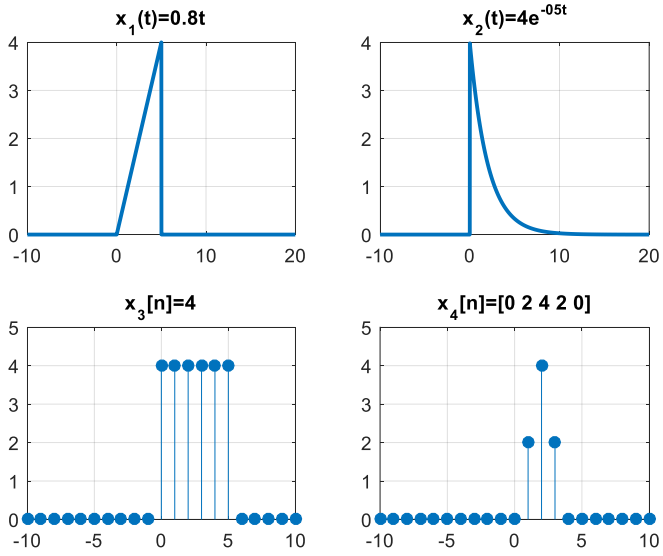
The even part of a function, g(t), is $ge(t) = \frac{g(t)+g(-t)}{2}$

The odd part of a function, g(t), is $go(t) = \frac{g(t)-g(-t)}{2}$

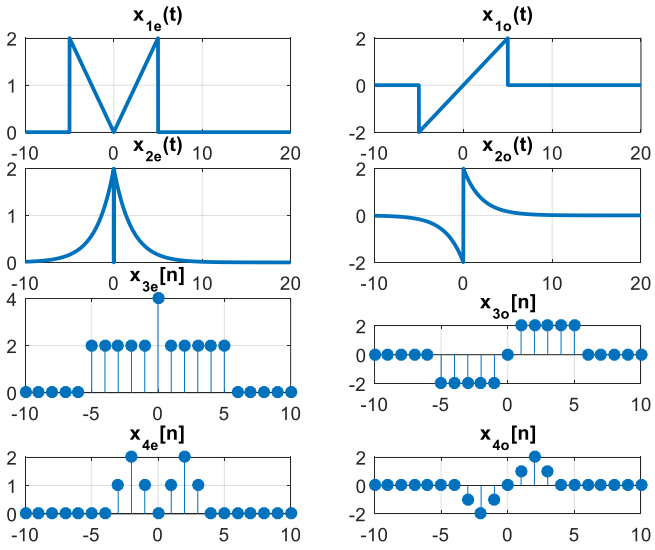
LAB TASKS, OBSERVATIONS & RESULTS:

```
%clear all variables and close all figures
clear variables;
close all;

%define the C-T time signal
t=-10:0.01:20;

%define the D-T time signal
n=-10:10;

%define the first C-T signal
x1= @(t) 0.8*t.*((t>0)&(t<=5));

%define the second C-T signal
x2= @(t) 4*exp(-0.5*t).*(t>0);

%define the first D-T signal
x3= @(n) 4*((n>=0)&(n<=5));

%define the second D-T signal
x4=@(n) 2*n.*((n>0)&(n<=2))+(8-2*n).*((n>2)&(n<=4));

disp('original signals');

%plot the first C-T signal
```

```
subplot(221),plot(t,x1(t),'linewidth', 2);
grid;title('x_1(t)=0.8t');

%plot the second C-T signal
subplot(222),plot(t,x2(t),'linewidth', 2);
grid;title('x_2(t)=4e^{-05t}');


%plot the first D-T signal
subplot(223),stem(n,x3(n),'filled');
grid;title('x_3[n]=4');ylim([0 5]);

%plot the second D-T signal
subplot(224),stem(n,x4(n),'filled');
grid;title('x_4[n]=[0 2 4 2 0]');ylim([0 5]);
```



```
%even and odd componenets
disp('even n odd comp of the signals');

%plot the first C-T even signal
subplot(421),plot(t,0.5*(x1(t)+x1(-t)),'linewidth', 2);
grid;title('x_{1e}(t)');
%plot the first C-T odd signal
subplot(422),plot(t,0.5*(x1(t)-x1(-t)),'linewidth', 2);
grid;title('x_{1o}(t)');

%plot the second C-T even signal
subplot(423),plot(t,0.5*(x2(t)+x2(-t)),'linewidth', 2);
grid;title('x_{2e}(t)');
%plot the second C-T odd signal
subplot(424),plot(t,0.5*(x2(t)-x2(-t)),'linewidth', 2);
grid;title('x_{2o}(t)');

%plot the first D-T even signal
subplot(425),stem(n,0.5*(x3(n)+x3(-n)),'filled');
grid;title('x_{3e}[n]');
%plot the first D-T odd signal
subplot(426),stem(n,0.5*(x3(n)-x3(-n)),'filled');
grid;title('x_{3o}[n]');


%plot the second D-T even signal
subplot(427),stem(n,0.5*(x4(n)+x4(-n)),'filled');
grid;title('x_{4e}[n]');
%plot the second D-T odd signal
subplot(428),stem(n,0.5*(x4(n)-x4(-n)),'filled');
grid;title('x_{4o}[n]');
```



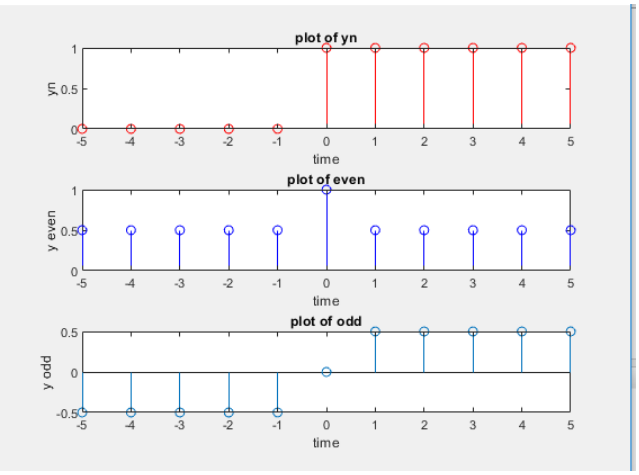
ACTIVITY:

Perform the above task on any 2 signals of your choice and attach the codes and results.

Code:1

```
n=-5:5;
yn= 0*(n<0)+1*(n==0)+1*(n>0);
yflip=fliplr(yn);
subplot(3,1,1);
stem(n,yn,'r');
title('plot of yn');
xlabel('time');
ylabel('yn');

ye = (1/2)*(yn+yflip);
subplot(3,1,2);
stem(n,ye,'b');
title('plot of even');
xlabel('time');
ylabel('y even');

yo= (1/2)*(yn-yflip);
subplot(3,1,3);
stem(n,yo);
title('plot of odd');
xlabel('time');
ylabel('y odd');
```

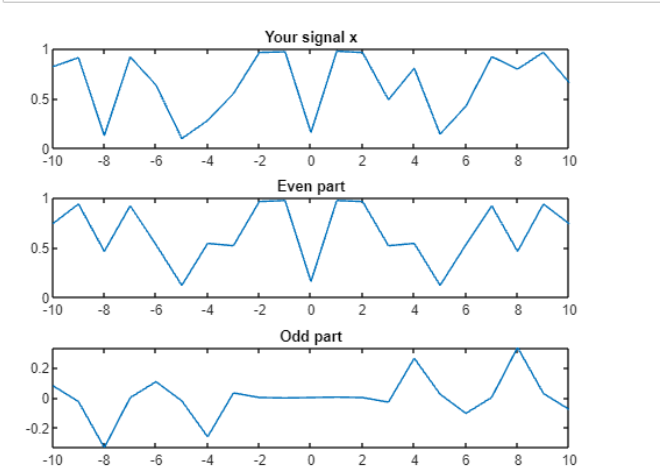
Output Result:



Code:2

```
t=-10:10; %vector time
x=rand(1,numel(t)); % Your signal
xmt=[fliplr(x(t>=0)) fliplr(x(t<0))]
xe=0.5*(xmt+x)
xo=0.5*(x-xmt)
subplot(3,1,1);
plot(t,x);
title('Your signal x')
subplot(3,1,2);
plot(t,xe);
title('Even part')
subplot(3,1,3);
plot(t,xo);
title('Odd part')
```

Output Result :



MEHRAN UNIVERSITY OF ENGINEERING AND TECHNOLOGY, JAMSHORO
Department of Electronic Engineering
SIGNALS & SYSTEMS (ES-304)

LAB # 07: SAMPLING, NYQUIST THEOREM, EFFECTS OF ALIASING IN TIME DOMAIN & QUANTIZATION

Name	Karan vaswani	Roll #	20ES062
Signature of Lab Tutor		Date	

OBJECTIVE(S)

#	*Topic*	# Of Lectures	CLO	Taxonomy level
1	To *perform* sampling using nyquist theorem and observe the effects of aliasing and design a quantizer.	3	4,5	P3,A4

OUTCOME(S)

• An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.	**PLO5**:Modern Tool Usage
• An ability to communicate effectively (written/oral)	**PLO10**: Communication

LAB RUBRICS:

Performance Metric	Good (5-4)	Average (3-2)	Poor (1-0)	Score
Use of modern engineering software [PLO5]	Demonstrates knowledge and application of modern engineering software through accurate development and interpretation of computer programs to solve problems.	Demonstrates awareness of modern engineering software through mostly correct development and interpretation of computer programs to solve problems, but may contain minor mistakes or syntax errors	Unable to use modern engineering software to develop or interpret computer programs to solve problems.	
Level of understanding of the learned skill [PLO5]	Demonstration of full knowledge of the handout with explanations and elaboration.	At ease with content and able to elaborate and explain to some degree.	No grasp of information. Clearly no knowledge of subject matter. No questions are answered. No interpretation made.	
Conducting simulation [PLO5]	Has an excellent simulations skill in the simulator. Always able to make the logical code for the given task.	Has a good simulation skill in the simulator. Always able to make the logical code for the given task.	Has poor simulation skill in the simulator. Unable to make the logical code for the given task.	
Responsiveness to Questions/ Accuracy [PLO10]	Responds well, quick and very accurate all the time.	Generally responsive and accurate most of the times.	Non-responsive at all or the candidate giving only one correct response of viva voce questions.	

Documentation: contents and organization [PLO10]	Report well organized, Appropriately sectioned, uses diagram/dscription when appropriate, important issues clearly stated.	Report reasonably well documented. May lack some minor aspects.	Report not well organized, lack key aspects.	
			Total	

EQUIPMENT/SOFTWARE TOOL:

MATLAB - The Language of Technical Computing.

DISCUSSION AND SETUP:

Sampling:

In signal processing, sampling is the reduction of a continuous signal to a discrete signal, as shown in the following figure.

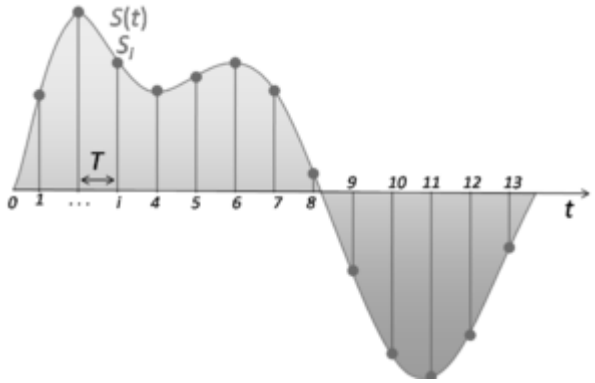


Figure 1: Sampling a Continuous Signal

Nyquist Theorem:

It is one of the most important rules of sampling, which states that the highest frequency in a signal which can be accurately represented is less than one-half of the sampling rate.
Mathematically,

$$f_s \geq 2 f_{\max}$$

Where,

f_s is the sampling frequency and

$f_{\max}$ is the maximum frequency component present in the signal.

For example, if we want a full 20 kHz audio bandwidth, we must sample at least twice that fast, i.e. over 40 kHz.

Aliasing:

Aliasing is the phenomenon that results in loss of information when a continuous time signal is reconstructed from its samples (and the Nyquist sampling theorem is violated).
The figure on the next page shows an alias.

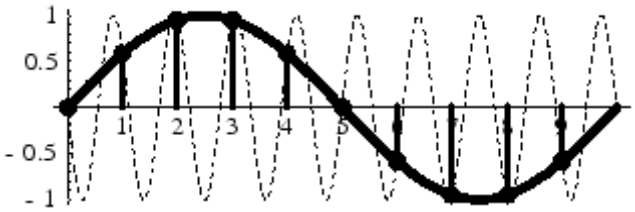


Figure 2: Aliasing due to undersampling (Original signal-Dotted, Sampled Signal: Bold)

There are two approaches to avoid aliasing. One is to raise the sampling frequency to satisfy the sampling theorem. The other is to filter off the unnecessary high-frequency components from the continuous-time signal. We limit the signal frequency by an effective low-pass filter, called **anti-aliasing prefilter**, so that the highest frequency left in the signal is less than half of the intended sampling rate.

LAB TASK, OBSERVATIONS & RESULTS:

Sampling obeying the Nyquist Theorem:

Consider the following code:

```
%Continous  time signal
t=0:0.0005:1; %Time vector for the signal
f=10; %Frequency of the CT signal
ansc=cos(2*pi*f*t); %Compute signal values
figure; %Open a new figure window
subplot(211); %Divide the figure window in to a tiled grid
plot(t,ansc); %Plot the continous time signals
grid on; %Show grid in the plot
xlabel('time'); %Label the x-axis
ylabel('Amplitude'); %Label the y-axis
title('Cont time'); %Give title to the plot
axis([0 1 -2 2]); %Set horizontal and vertical axes limits
%Discrete time signal
Fs=2*f; %Sampling Frequency of the signal
Ts=1/Fs; %Determine sampling time
nTs=0:Ts:1; %Indices of samples in the signal
ansd=cos(2*pi*f*nTs); %Compute sample values of the signal
subplot(212); %Divide the figure window in to a tiled grid
stem(nTs,ansd); %Plot the discrete time signal
grid on; %Show grid in the plot
xlabel('samples'); %Label the x-axis
ylabel('Amplitude'); %Label the y-axis
title('Disct time'); %Give title to the plot
axis([0 1 -2 2]); %Set horizontal and vertical axes limits
```

The output of the above code is shown below:

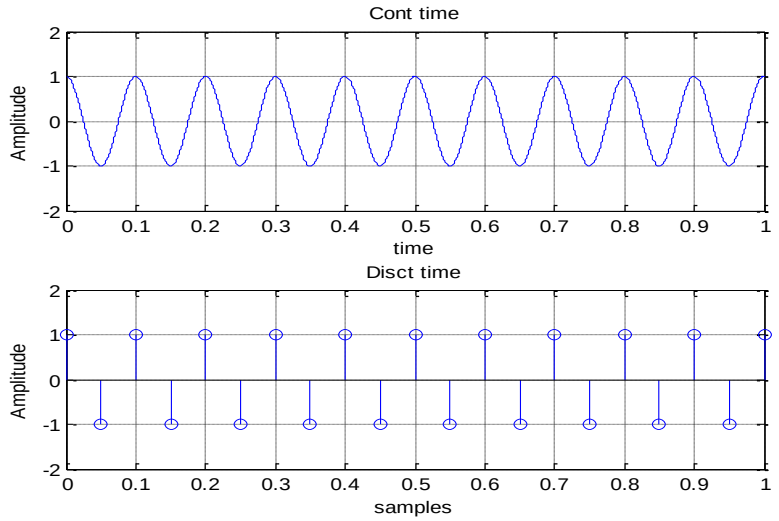


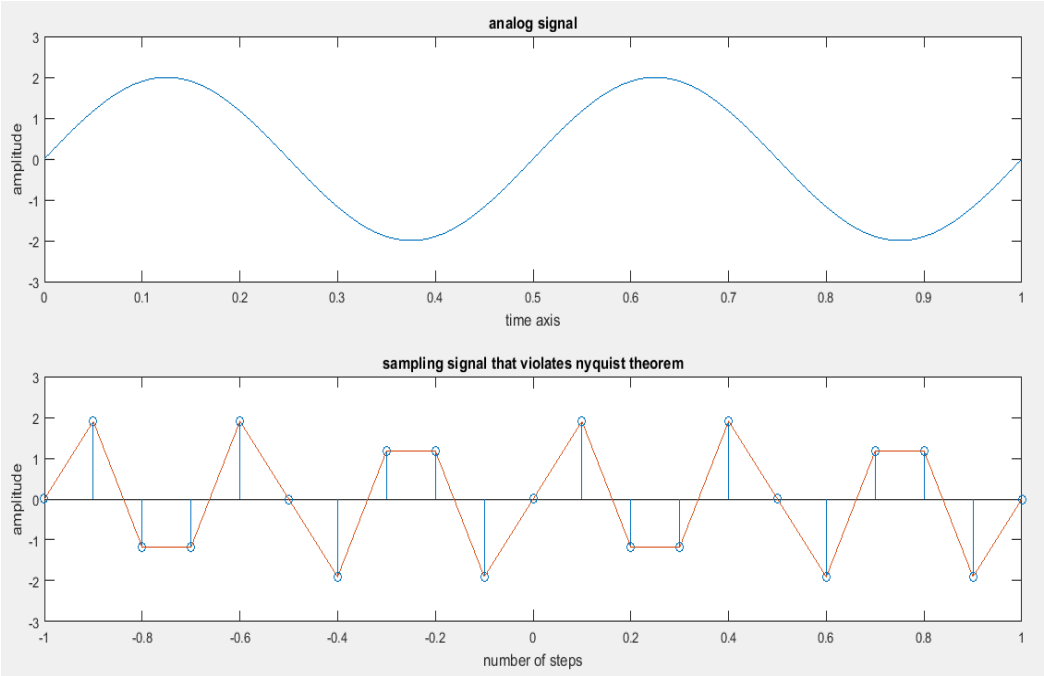
Figure 3: Sampling a sinusoid

1. Modify the code above such that the discrete time signal violates the Nyquist Theorem. Show the plot and write your observations.

   ANS:

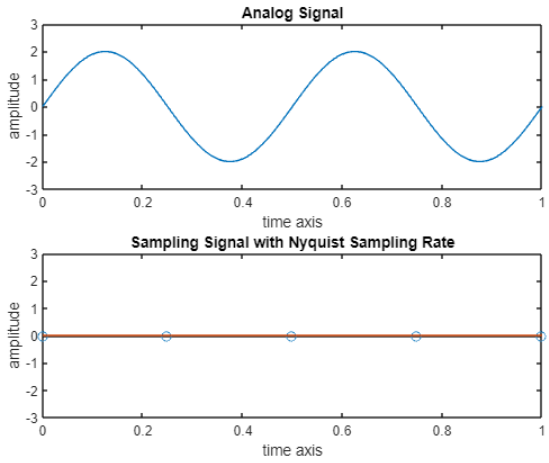
```
%program to violate the nyquist theorem
>> %analog signal
>> t=0:0.01:1;
>> fm=2; %Hz
>> x=2*sin(2*pi*fm*t);
>> subplot(2,1,1);
>> plot(t,x);
>> xlabel('time axis');
>> ylabel('amplitude');
>> title('analog signal');
>> axis([0 1 -3 3]);
%sampling signal that violates nyquist theorem means don't satisfy fs>=fm condition.
>> n=-1:0.1:1;
>> fs=1.5*fm; %Hz
>> y=2*sin(2*pi*fs*n);
>> subplot(2,1,2);
>> stem(n,y);
>> hold on;
>> plot(n,y);
>> xlabel('number of steps');
>> ylabel('amplitude');
>> title('sampling signal that violates nyquist theorem');
>> axis([-1 1 -3 3]);
```

Result:



2. Replace the cosine signals being used in the code above with sine. Show the plots obtained and discuss the results. Explain how it can be improved.

   The main difference is that we replaced the cos function with sin function in both the analog and sampling signals. The resulting plot shows a sine wave instead of a cosine wave, but the overall shape and frequency of the signal remains the same. There is no significant difference in the plot obtained by using sine instead of cosine signals.

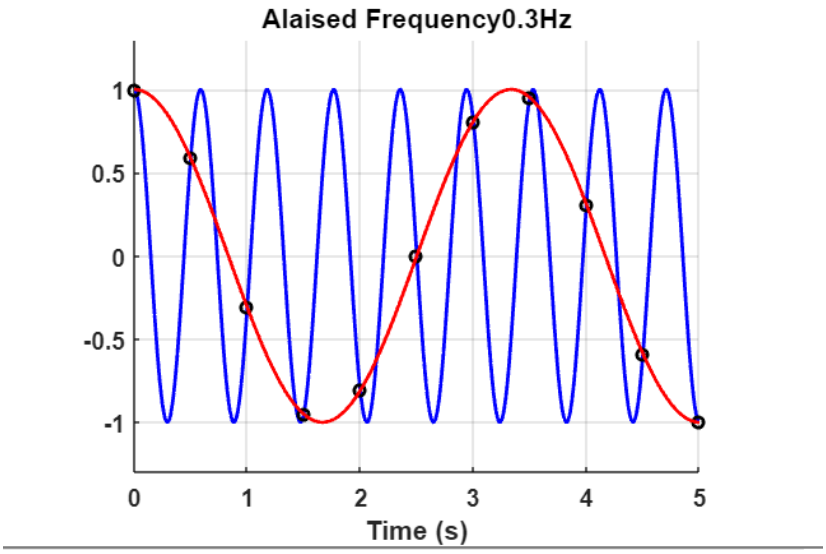


Ways to improve Program:
Add noise reduction techniques: Real-world signals often contain noise that can distort the signal and affect its quality. Adding noise reduction techniques such as filtering or smoothing can help remove the noise and improve the quality of the signal.
Add additional signal types: The program currently only generates a sine wave. Adding additional signal types such as square waves, triangular waves, or sawtooth waves can provide a better understanding of how different signals behave and interact with the Nyquist theorem.

Task 2:
Effect of Aliasing and calculating Aliased frequency

```
close all
% set time factor and frequency for cont time sinusoid
tMAX = 5;
t    = 0:0.0001:tMAX;
f    = 1.7;

%set sampled rate for sampled sinusoid
fs   = 2;
tSamp = 0:1/fs:tMAX;

%create cont and sampled signals
xCont = cos(2*pi*f*t);
xSamp = cos(2*pi*f*tSamp);

figure;
hold on;
set(gca, 'fontsize', 14, 'fontweight', 'bold');
plot(t, xCont, 'b', 'linewidth', 2);
plot(tSamp, xSamp, 'ko', 'linewidth',2);

%compute aliased frequency
if  fs < 2*f;

   %determine aliased frequency
   keepGoing = 1;
   m = 0;
   while (keepGoing)
       fAlias = abs(f - m*fs);
      if fAlias <= fs/2
        keepGoing = 0;
      else
         m = m+1;
      end
```

```
    end

 %plot aliased sinusoid
 plot(t,cos(2*pi*fAlias*t), 'r', 'linewidth', 2);
 title(['Alaised Frequency' num2str(fAlias) 'Hz']);

else
    title('No Aliasing');
end
 ylim([-1.3 1.3]);
 grid on;
 xlabel('Time (s)');
```



3. What are alias frequencies?

ANS: Frequencies above the nyquist rate are alias frequencies .

4. If the maximum frequency component in a signal is 40 hz, what should be the minimum sampling frequency to avoid aliasing?

ANS: Minimum sampling frequency should be 80Hz .

Task 3:

Signal Quantisation:

Sampling results in a signal which is continuous in amplitude but discrete in time. Since the hardware used to process this signal is capable of working using a limited number of bits (word size), the sampled signal needs to be appropriately quantised to be useful for further processing. The code below shows a 3 bit quantiser for a sinusoidal.

```
%% Quantisation
%Original Signal
clear
f=1; %Frequency of the signal
Fs=50*f; %Sampling Frequency of the signal
Ts=1/Fs; %Determine sampling time
nTs=0:Ts:1; %Indices of samples in the signal
ansd=cos(2*pi*f*nTs); %Compute sample values of the signal
subplot(211); %Divide the figure window in to a tiled grid
plot(nTs,ansd); %Plot the discrete time signal
grid on; %Show grid in the plot
xlabel('samples'); %Label the x-axis
ylabel('Amplitude'); %Label the y-axis
title('Original Signal'); %Give title to the plot
%Quantisation of the signal
b=3; %No of bits in the quantiser
max_amp=max(ansd)+abs(min(ansd));
res=max_amp/2^b; %Resolution of the quantiser
quant_ansd=round(ansd/res)*res; %Quantised signal
quant_ansd((quant_ansd>=max(ansd)-res))=max(quant_ansd)-res; %Limit max
%value one level below max possible
subplot(212); %Divide the figure window in to a tiled grid
stem(nTs,quant_ansd); %Plot the discrete time signal
grid on; %Show grid in the plot
xlabel('samples'); %Label the x-axis
ylabel('Amplitude'); %Label the y-axis
title(['Quantised Signal (',num2str(b),' bit quantiser)']); %Give title to
the plot
```

The output of the code is shown in Fig. 2

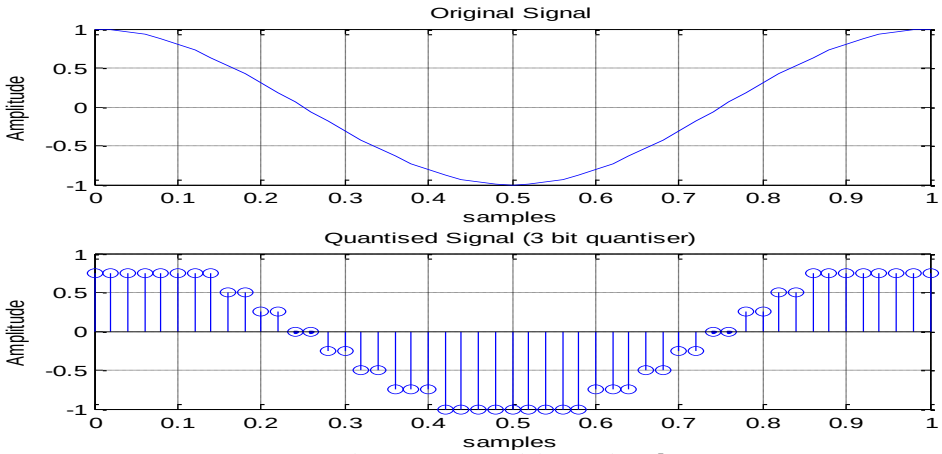


Figure 4: Quantising a signal

REVIEW QUESTIONS:

1. Discuss the term quantisation error?

   Quantization error is an error that occurs in the process of digitizing a continuous analog signal. When an Analog signal is converted to a digital signal, it is first sampled and then quantized. Sampling involves measuring the amplitude of the analog signal at discrete time intervals, and quantization involves assigning a discrete value to each sample based on its amplitude

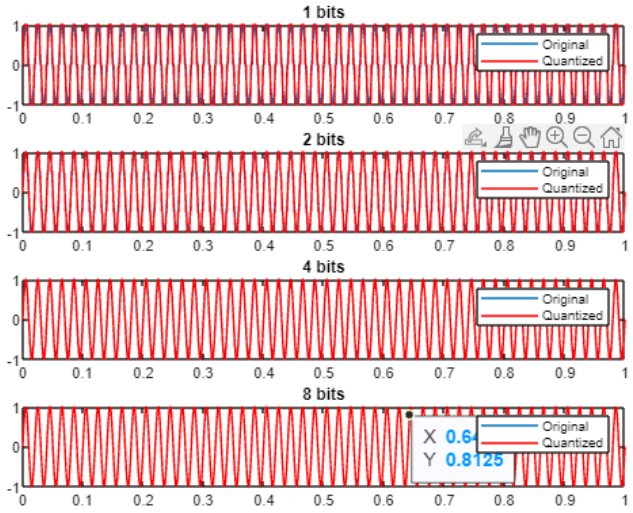
2. Change the number of bits of the quantizer and observe the output signal. Discuss the effect on the quantisation error?

   The quantization error is affected by the number of bits used to represent the quantization levels. Increasing the number of bits reduces the quantization error, while decreasing the number of bits increases the quantization error
   To illustrate this effect, we can use a simple MATLAB code that generates a sine wave and applies quantization with different numbers of bits. Here is an example code:

```
% Generate a sine wave
t = 0:0.001:1;
f = 50; % Hz
x = sin(2*pi*f*t);
% Quantize the signal with different numbers of bits
b = [1, 2, 4, 8];
for i = 1:length(b)
    y = quantize(x, b(i));
    subplot(length(b), 1, i);
    plot(t, x);
    hold on;
    plot(t, y, 'r');
    title(sprintf('%d bits', b(i)));
    legend('Original', 'Quantized');
end
% Quantization function
```

```
function y = quantize(x, b)
    q = 1/(2^(b-1));
    y = q*round(x/q);
end
```

Result:



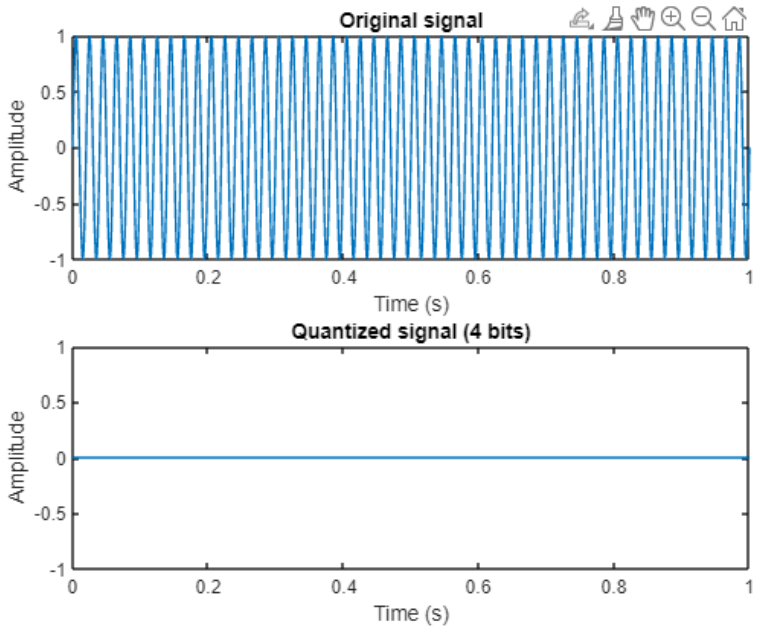
3. Using the Matlab function ‘quant’, use the code above to design your own quantiser.

   The quant function in MATLAB is used to quantize a signal to a specified number of bits. Here's an example code that uses the quant function to design a quantizer:

```
% Generate a sine wave
t = 0:0.001:1;
f = 50; % Hz
x = sin(2*pi*f*t);
% Quantize the signal to 4 bits
b = 4;
y = quant(x, b);
% Plot the original and quantized signals
subplot(2,1,1);
plot(t, x);
xlabel('Time (s)');
ylabel('Amplitude');
title('Original signal');
subplot(2,1,2);
plot(t, y);
```

```
xlabel('Time (s)');
ylabel('Amplitude');
title(sprintf('Quantized signal (%d bits)', b));
```

Result:



MEHRAN UNIVERSITY OF ENGINEERING AND TECHNOLOGY, JAMSHORO
Department of Electronic Engineering
SIGNALS & SYSTEMS (ES-304)

LAB # 08: FOURIER SERIES

Name	Karan vaswani	Roll #	20ES062
Signature of Lab Tutor		Date	

OBJECTIVE(S)

#	*Topic*	# Of Lectures	CLO	Taxonomy level
1	To *perform* fourier series using matlab functions	3	4,5	P3,A4

OUTCOME(S)

• An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.	**PLO5**:Modern Tool Usage
• An ability to communicate effectively (written/oral)	**PLO10**: Communication

LAB RUBRICS:

Performance Metric	Good (5-4)	Average (3-2)	Poor (1-0)	Score
Use of modern engineering software [PLO5]	Demonstrates knowledge and application of modern engineering software through accurate development and interpretation of computer programs to solve problems.	Demonstrates awareness of modern engineering software through mostly correct development and interpretation of computer programs to solve problems, but may contain minor mistakes or syntax errors	Unable to use modern engineering software to develop or interpret computer programs to solve problems.	
Level of understanding of the learned skill [PLO5]	Demonstration of full knowledge of the handout with explanations and elaboration.	At ease with content and able to elaborate and explain to some degree.	No grasp of information. Clearly no knowledge of subject matter. No	

			questions are answered. No interpretation made.	
Conducting simulation [PLO5]	Has an excellent simulations skill in the simulator. Always able to make the logical code for the given task.	Has a good simulation skill in the simulator. Always able to make the logical code for the given task.	Has poor simulation skill in the simulator. Unable to make the logical code for the given task.	
Responsiveness to Questions/ Accuracy [PLO10]	Responds well, quick and very accurate all the time.	Generally responsive and accurate most of the times.	Non-responsive at all or the candidate giving only one correct response of viva voce questions.	
Documentation: contents and organization [PLO10]	Report well organized, Appropriately sectioned, uses diagram/dscription when appropriate, important issues clearly stated.	Report reasonably well documented. May lack some minor aspects.	Report not well organized, lack key aspects.	
			Total	

EQUIPMENT/SOFTWARE TOOL:

MATLAB - The Language of Technical Computing.

DISCUSSION AND SETUP:

Fourier series decomposes a periodic function or aperiodic signal into a sum of simple oscillating functions, namely sines and cosines. Fourier series has many applications in electrical engineering, vibration analysis, acoustics, optics, signal processing, image processing, quantum mechanics, etc.

Fourier's formula for 2π-periodic functions using sines and cosines:
For a 2π-periodic function $f(x)$ that is integrable on $[-\pi, \pi]$:

$$a_0 + \sum_{n=1}^{\infty} (a_n \cos nx + b_n \sin nx)$$

where the coefficients a_n and b_n in this series are defined by

$$a_0 = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x)\, dx$$

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos nx\, dx \qquad b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin nx\, dx$$

and are called the **Fourier coefficients** of f.

Examples of Fourier series with different number of terms:

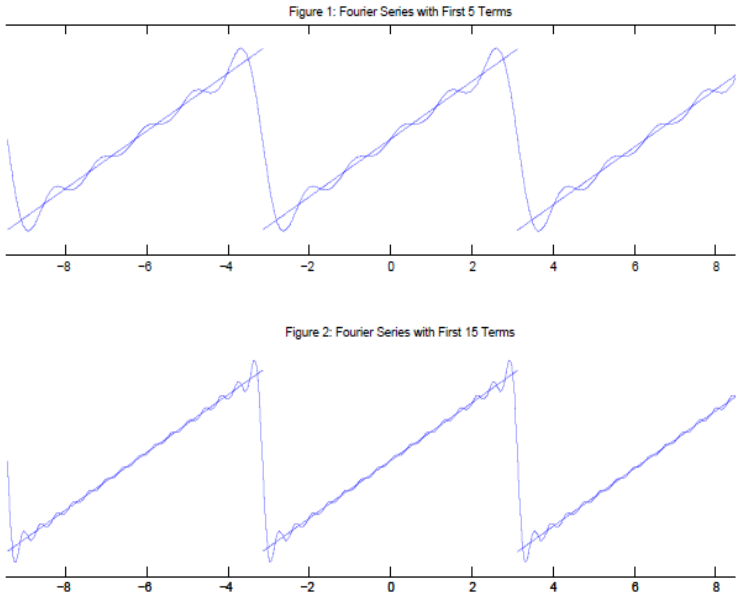


Figure 1

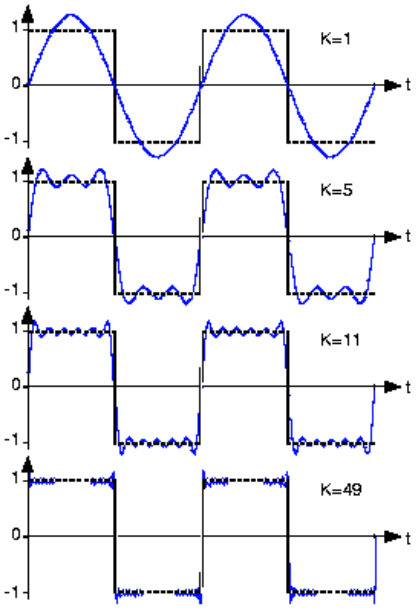


Figure 2

Fourier representation of Sawtooth wave

The sawtooth is sum of all the terms of the Fourier series, i.e., from 1 to infinite. Mathematically,

Y=sin wt + (1/2) sin 2wt + (1/3) sin 3wt + (1/4) sin 4wt + (1/5) sin 5wt + (1/6) sin 6wt + (1/7) sin 7wt + …….

Fourier representation of Square wave

The Square wave is mathematically approximated, using Fourier theorem, as under:

Y=sin wt + (1/3) sin 3wt +(1/5)sin 5wt + (1/7) sin 7wt+(1/9) sin 9wt + (1/11) sin 11wt + (1/13) sin 13wt + ….

LAB TASKS, OBSERVATIONS AND RESULTS:

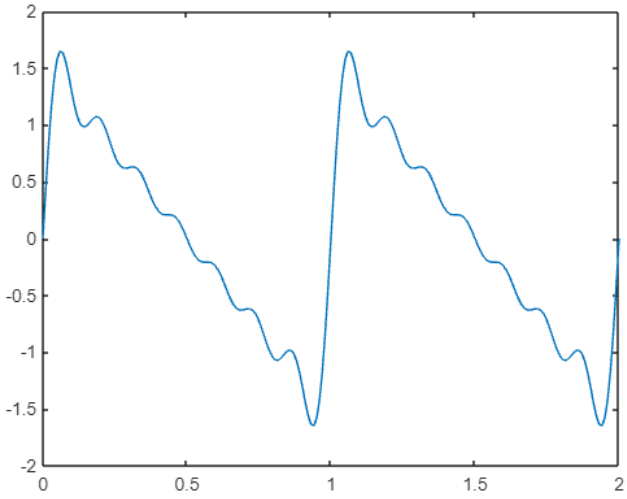
Lab Task 1:

% Fourier_ Sawtooth wave

%Program 1

```
f=1;
w=2*pi*f;
t=0:0.01:2;
a=1*sin(w*t);
b=(1/2)*sin(2*w*t);
c=(1/3)*sin(3*w*t);
d=(1/4)*sin(4*w*t);
e=(1/5)*sin(5*w*t);
f=(1/6)*sin(6*w*t);

g=(1/7)*sin(7*w*t);
y=a+b+c+d+e+f+g;
plot(t,y)
```

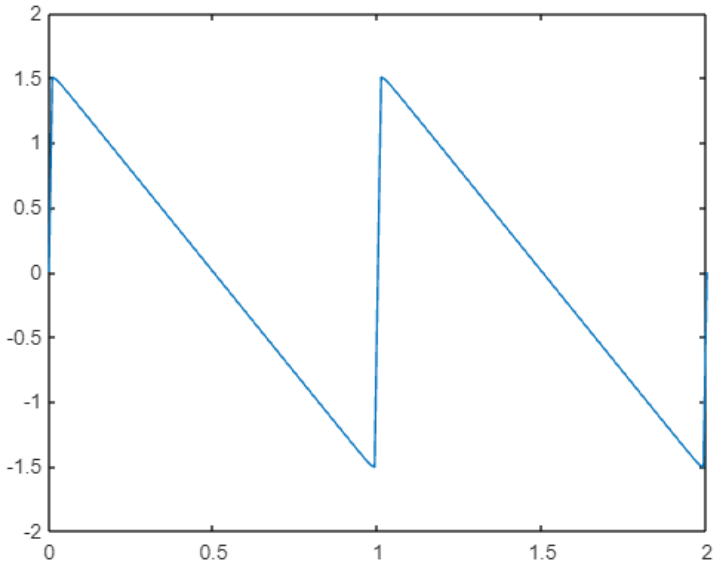
Result:




```
% Program 2
% Fourier_ Sawtooth wave
y=0;
f=1;
```

```
w=2*pi*f;
t=0:0.01:2;
for n=1:500
y=y+(1/n)*sin(n*w*t);
end
plot(t,y)
```

Result:

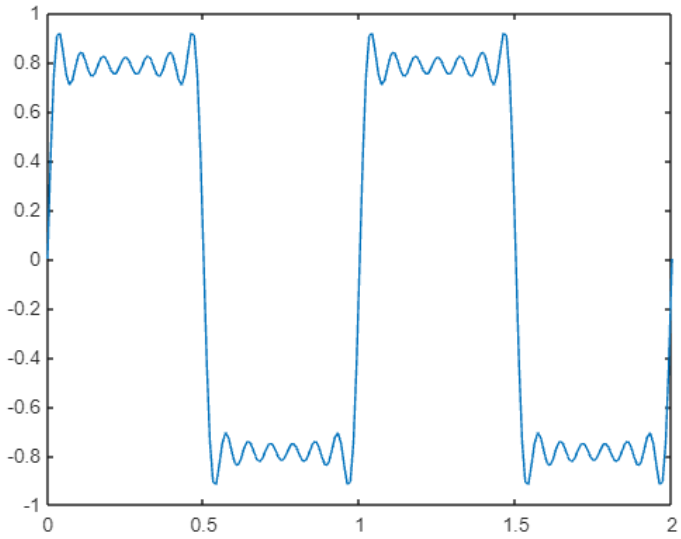


Lab Task 2:

```
% Fourier_ Square wave
%Program 3
f=1;
w=2*pi*f;
t=0:0.01:2;
a=1*sin(w*t);
```

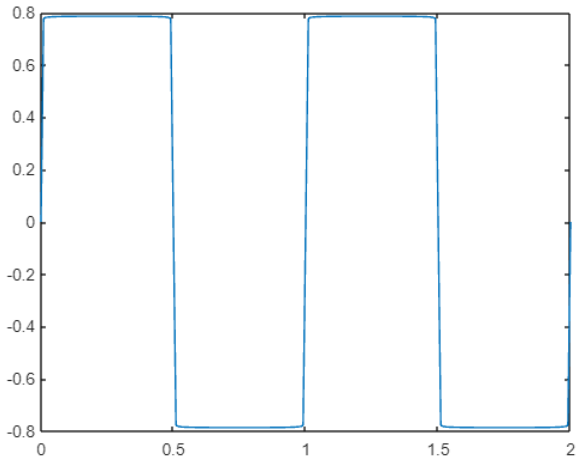
```
b=(1/3)*sin(3*w*t);
c=(1/5)*sin(5*w*t);
d=(1/7)*sin(7*w*t);
e=(1/9)*sin(9*w*t);
f=(1/11)*sin(11*w*t);
g=(1/13)*sin(13*w*t);
y=a+b+c+d+e+f+g;
plot(t,y)
```

Result:




```
% Program 4
% Fourier_ Square Wave
a=0;
f=1;
w=2*pi*f;
t=0:0.01:2;
for n=1:2:1001
a=a+(1/n)*sin(n*w*t);
end
plot(t,a)
```

Result:



REVIEW QUESTIONS:

1. What is Fourier Series?
   Fourier series is a mathematical tool that allows any periodic signal (i.e., a signal that repeats itself over time) to be expressed as an infinite sum of simple sine and cosine waves of varying amplitudes and frequencies. In other words, any periodic signal can be decomposed into its constituent sine and cosine components, which can then be added together to reconstruct the original signal.

2. Briefly mention some applications of Fourier Series.
   Fourier series has numerous applications in various fields of science and engineering. Some of the common applications are:
   Signal processing: Fourier series is widely used in signal processing to analyze and manipulate signals. It is used in audio and image processing to remove noise and distortion from the signals.

   Communication systems: Fourier series is used in the design and analysis of communication systems, such as wireless communication systems, to extract information from signals and transmit them efficiently.

   Control systems: Fourier series is used in control systems to analyze the behavior of the system and design controllers to achieve desired performance.

3. Approximate Triangular Wave with the Fourier Series and write a Matlab program to verify the approximation.
   A triangular wave is a waveform that rises linearly from its lowest value to its highest value, and then falls linearly back to its lowest value. The waveform is characterized by its period (T), amplitude (A), and the slope of its rise and fall (m). The Fourier series for a triangular wave is given by:

   f(t) = (8A/pi^2) * sum(n=0,1,2,...) [(1-(-1)^n)/(n^2)] * sin(2npit/T)

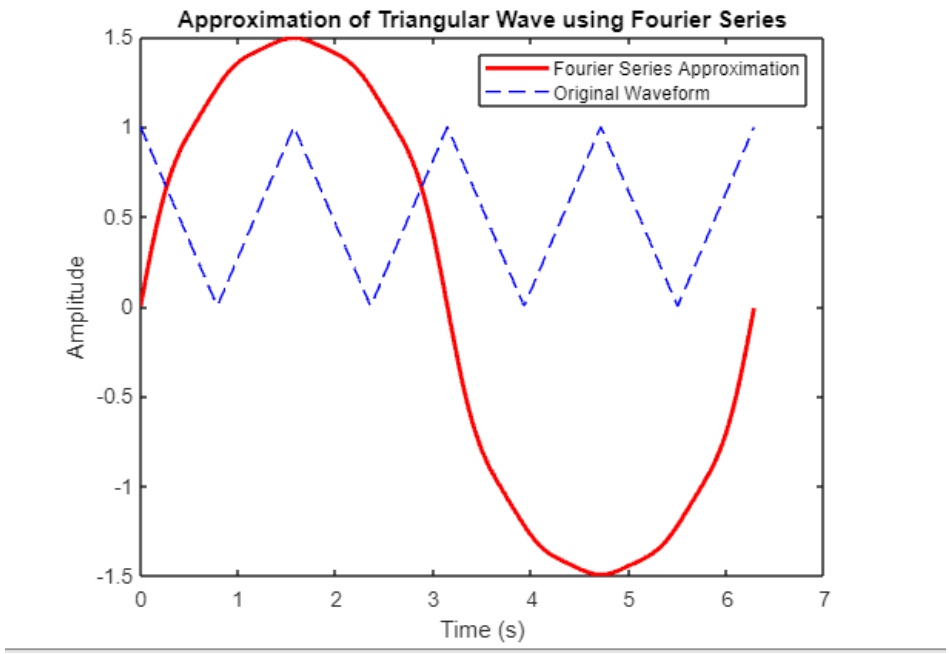
   where t is the time variable and n is the index of the harmonic.

   Here is code :

```
% Define the period, amplitude, and slope of the triangular wave
T = 2*pi;
A = 1;
m = 2;

% Define the time range and sampling interval
t = 0:0.01:T;
dt = t(2) - t(1);

% Initialize the Fourier series sum
f = zeros(size(t));

% Compute the Fourier series sum up to the 10th harmonic
for n = 1:10
    f = f + (8*A/pi^2) * ((1-(-1)^n)/(n^2)) * sin(2*n*pi*t/T);
end

% Plot the Fourier series approximation and the original waveform
plot(t, f, 'r-', 'LineWidth', 2);
hold on;
plot(t, A*abs(sawtooth(m*t, 0.5)), 'b--', 'LineWidth', 1);
```

```
xlabel('Time (s)');
ylabel('Amplitude');
legend('Fourier Series Approximation', 'Original Waveform');
title('Approximation of Triangular Wave using Fourier Series');
```

Result:



MEHRAN UNIVERSITY OF ENGINEERING AND TECHNOLOGY, JAMSHORO
Department of Electronic Engineering
SIGNALS & SYSTEMS (ES-304)

LAB # 09: FOURIER TRANSFORM

Name	Karan vaswani	Roll #	20ES062
Signature of Lab Tutor		Date	

OBJECTIVE(S)

#	*Topic*	# Of Lectures	CLO	Taxonomy level
1	To *perform* Fourier Transform using matlab functions	3	4,5	P3,A4

OUTCOME(S)

• An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.	**PLO5**:Modern Tool Usage
• An ability to communicate effectively (written/oral)	**PLO10**: Communication

LAB RUBRICS:

Performance Metric	Good (5-4)	Average (3-2)	Poor (1-0)	Score
Use of modern engineering software [PLO5]	Demonstrates knowledge and application of modern engineering software through accurate development and interpretation of computer programs to solve problems.	Demonstrates awareness of modern engineering software through mostly correct development and interpretation of computer programs to solve problems, but may contain minor mistakes or syntax errors	Unable to use modern engineering software to develop or interpret computer programs to solve problems.	
Level of understanding of the learned skill [PLO5]	Demonstration of full knowledge of the handout with explanations and elaboration.	At ease with content and able to elaborate and explain to some degree.	No grasp of information. Clearly no knowledge of subject matter. No questions are answered. No interpretation made.	
Conducting simulation [PLO5]	Has an excellent simulations skill in the simulator. Always able to make the logical code for the given task.	Has a good simulation skill in the simulator. Always able to make the logical code for the given task.	Has poor simulation skill in the simulator. Unable to make the logical code for the given task.	
Responsiveness to Questions/ Accuracy [PLO10]	Responds well, quick and very accurate all the time.	Generally responsive and accurate most of the times.	Non-responsive at all or the candidate giving only one correct response of viva voce questions.	
Documentation: contents and organization [PLO10]	Report well organized, Appropriately sectioned, uses diagram/dscription when appropriate,	Report reasonably well documented. May lack some minor aspects.	Report not well organized, lack key aspects.	

	important issues clearly stated.			
			Total	

EQUIPMENT/SOFTWARE TOOL:

MATLAB - The Language of Technical Computing.

DISCUSSION AND SETUP:

The Fourier transform is in fact an extension of the Fourier series, which doesn't accurately approximate continuous-time aperiodic signals. The Fourier transform of a continuous time signal **x(t)** is represented by **X(f)**. The Fourier transform is obtained by extending the expansion interval to infinity, i.e. T→ ∞. The resulting mathematical form is:

$$X(f) = \int_{-\infty}^{+\infty} x(t) e^{-j2\pi ft} dt$$

Amplitude Density Spectrum

The amplitude density spectrum of a continuous time signal **x(t)** is magnitude of its Fourier transform, i.e., **|X(f)|**. If **x(t)** has the unit of amperes, its resulting amplitude density function shall have the units of amperes/Hz.

Energy Density Spectrum

The energy density of a continuous time signal **x(t)** is the square of its amplitude density function, i.e., $|X(f)|^2$. Accordingly, its unit is square of that of **|X(f)|.**

Phase Spectrum

The Phase Spectrum of a continuous time signal **x(t)** is angle of its Fourier Transform **X(f)**, i.e., **∠X(f)**. Its unit is radians.

LAB TASKS, OBSERVATIONS AND RESULTS:

Lab Task 1:

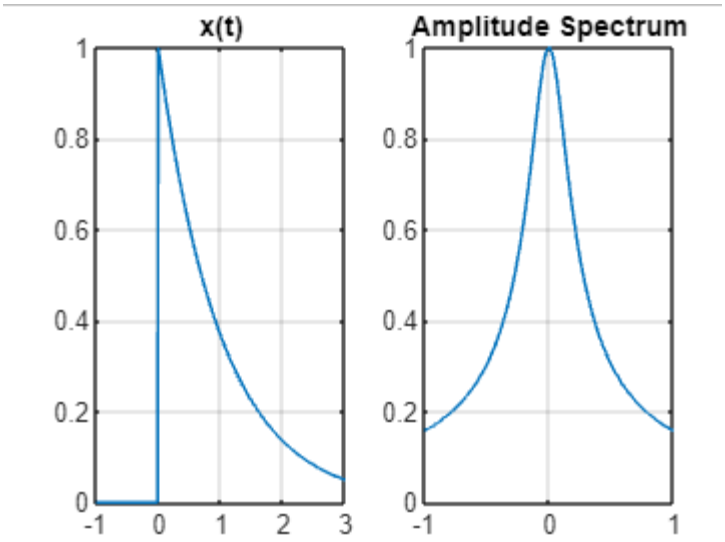
Following Matlab program plots function $\mathbf{x(t)=e^{-t}\,u(t)}$ and its amplitude density spectrum **|X(f)|**.

Code:

```
t=-1:0.01:3;
ut=[zeros(1,100) ones(1,301)] ;
xt=exp(-t).*ut;
f=-1:0.001:1;
xamp=1./sqrt(1+4*(pi^2)*f.^2);
subplot(1,2,1)
plot(t, xt);
title('x(t)')
grid on
```

```
subplot(1,2,2)
plot(f, xamp);
title('Amplitude Spectrum')
grid on
```

Results:



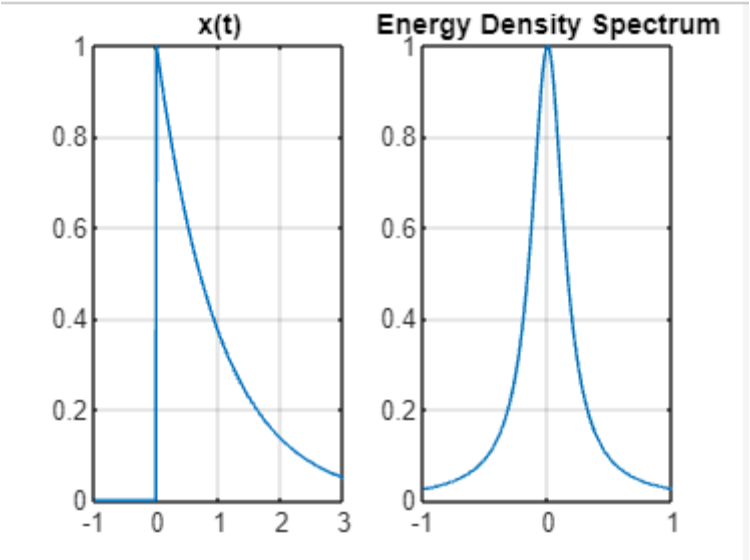
Lab Task 2:

Following Matlab program generates and plots function $x(t) = e^{-t}u(t)$, and its energy density function $|X(f)|^2$.

Code:

```
t=-1:0.01:3;
ut=[zeros(1,100) ones(1,301)] ;
xt=exp(-t).*ut;
f=-1:0.001:1;
xeng=1./(1+4*(pi^2)*f.^2);
subplot(1,2,1)
plot(t, xt)
title('x(t)')
grid on
subplot(1,2,2)
plot(f, xeng)
title('Energy Density Spectrum')
grid on
```

Results:



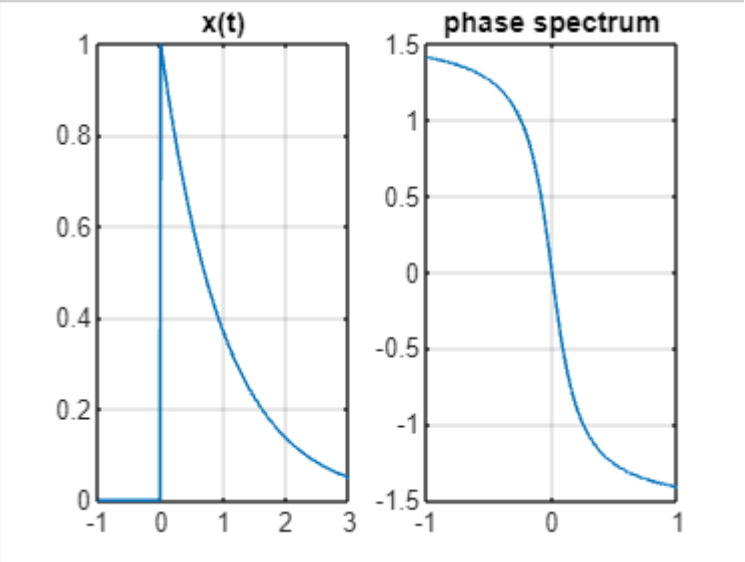
Lab Task 3:

Following Matlab program plots function $x(t)=e^{-t}\,u(t)$, its phase spectrum $\angle X(f)$.

Code:

```
t=-1:0.01:3;
ut=[zeros(1,100) ones(1,301)];
xt=exp(-t).*ut;
f=-1:0.001:1;
xphase=-atan(2*pi*f);
subplot(1,2,1)
plot(t, xt)
title('x(t)')
grid on
subplot(1,2,2)
plot(f, xphase)
title('phase spectrum')
grid on
```

Results:



Fourier transform using matlab function:

`fourier(f)` returns the Fourier Transform of `f`.

Examples

Fourier Transform of Common Inputs

Compute the Fourier transform of common inputs. By default, the transform is in terms of `w`.

1. Rectangular pulse:

```
syms a b t
f=rectangularPulse(a,b,t);
f_FT=fourier(f)
```

2. Unit impulse (Dirac delta)

```
f=dirac(t);
f_FT = fourier(f)
```

3. Step (Heaviside)

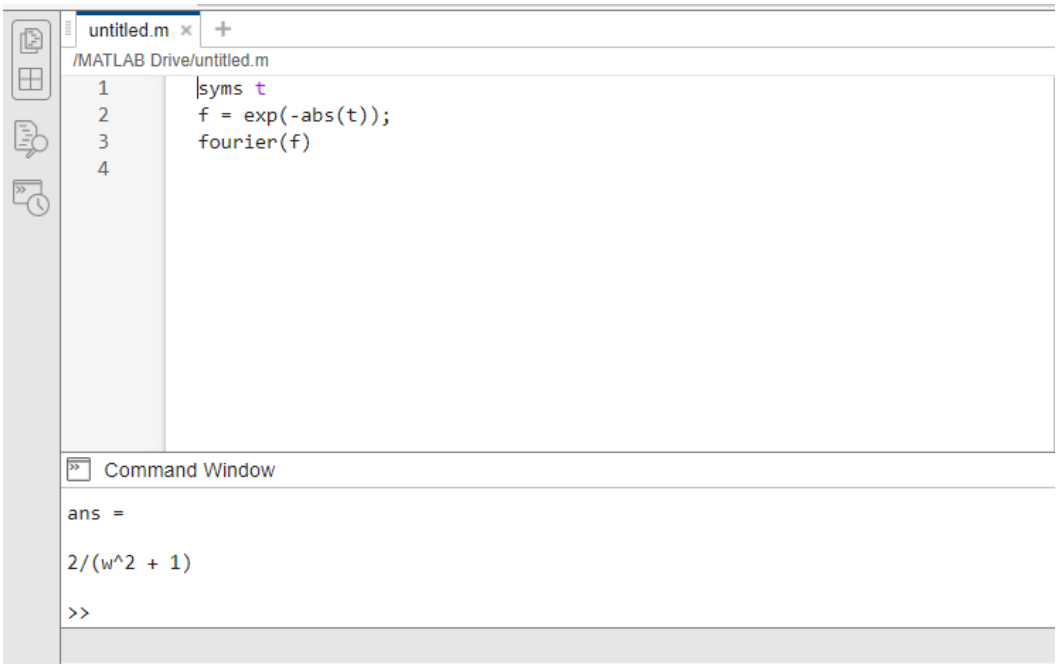
```
f=heaviside(t);
f_FT = fourier(f)
```

Lab Task 4:
Compute the Fourier transform of $e^{-|t|}$.

Code:

```
syms t
f = exp(-abs(t));
fourier(f)
```

Answer:



Review Questions:

1. Why is Fourier Transform used?
   Fourier transform is used to represent a signal in the frequency domain. It converts a signal from the time domain to the frequency domain, allowing us to analyze the signal's frequency content and extract useful information.

2. What is Amplitude Density Spectrum?
   The amplitude density spectrum is the magnitude of the Fourier Transform of a signal, which represents the amplitude of each frequency component in the signal. It provides a visual representation of the signal's frequency content and is useful for identifying dominant frequency components and analyzing their behavior.

3. What is Energy Density Spectrum?
   The energy density spectrum is the squared magnitude of the Fourier Transform of a signal, which represents the energy of each frequency component in the signal. It provides a visual representation of the signal's energy distribution across different frequencies and is useful for applications such as power spectral density estimation and signal classification

4. What is Phase Spectrum?
   The phase spectrum is the phase angle of the Fourier Transform of a signal, which represents the phase shift of each frequency component in the signal relative to a reference signal. It provides a visual representation of the signal's phase characteristics across different frequencies and is useful for applications such as signal filtering and signal reconstruction.

5. What is a **`sinc`** function?
   A sinc function is a mathematical function that is defined as the ratio of the sine function and its argument. Specifically, the sinc function is defined as sin(x)/x, where x is the argument of the function. The sinc function has a central lobe with a width of 2π and decaying side lobes that extend to infinity. The sinc function is used in various applications such as signal processing, filtering, and interpolation.

MEHRAN UNIVERSITY OF ENGINEERING AND TECHNOLOGY, JAMSHORO
Department of Electronic Engineering
SIGNALS & SYSTEMS (ES-304)

LAB # 10: CORRELATION

Name	Karan vaswani	Roll #	20ES062
Signature of Lab Tutor		Date	

OBJECTIVE(S)

#	*Topic*	# Of Lectures	CLO	Taxonomy level
1	To *perform* Correlation operation using matlab functions	3	4,5	P3,A4

OUTCOME(S)

• An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.	**PLO5**:Modern Tool Usage
• An ability to communicate effectively (written/oral)	**PLO10**: Communication

LAB RUBRICS:

Performance Metric	Good (5-4)	Average (3-2)	Poor (1-0)	Score
Use of modern engineering software [PLO5]	Demonstrates knowledge and application of modern engineering software through accurate development and interpretation of computer programs to solve problems.	Demonstrates awareness of modern engineering software through mostly correct development and interpretation of computer programs to solve problems, but may contain minor mistakes or syntax errors	Unable to use modern engineering software to develop or interpret computer programs to solve problems.	
Level of understanding of the learned skill [PLO5]	Demonstration of full knowledge of the handout with explanations and elaboration.	At ease with content and able to elaborate and explain to some degree.	No grasp of information. Clearly no knowledge of subject matter. No questions are answered. No interpretation made.	
Conducting simulation [PLO5]	Has an excellent simulations skill in the simulator. Always able to make the logical code for the given task.	Has a good simulation skill in the simulator. Always able to make the logical code for the given task.	Has poor simulation skill in the simulator. Unable to make the logical code for the given task.	
Responsiveness to Questions/ Accuracy [PLO10]	Responds well, quick and very accurate all the time.	Generally responsive and accurate most of the times.	Non-responsive at all or the candidate giving only one correct response of viva voce questions.	

Documentation: contents and organization [PLO10]	Report well organized, Appropriately sectioned, uses diagram/dscription when appropriate, important issues clearly stated.	Report reasonably well documented. May lack some minor aspects.	Report not well organized, lack key aspects.	
			Total	

EQUIPMENT/SOFTWARE TOOL:

MATLAB - The Language of Technical Computing.

DISCUSSION AND SETUP:

Correlation is a mathematical operation that is very similar to convolution. Just as with convolution, correlation uses two signals to produce a third signal. This third signal is called the cross-correlation of the two input signals. If a signal is correlated with itself, the resulting signal is instead called the autocorrelation.

Convolution is the relationship between a system's input signal, output signal, and impulse response whereas Correlation is a way to detect a known waveform in a noisy background.

Suppose we have two DT signals, x(n) and y(n). There cross correlation $r_{xy}(j)$ is calculated as:

$$r_{xy}(n) = \sum_{k=-\infty}^{\infty} x(k)y(k-n)$$

The cross correlation of two signals is implemented in MATLAB by "xcorr" function.

LAB TASKS, OBSERVATIONS & RESULTS:

Example of Cross-Correlation in MATLAB:

Consider the following code:

```
%cross correlation

x1=[5 4 3 2 1]; % first sigal
x2=[1 2 1 1 1]; % second signal
[y,n]=xcorr(x1,x2);   % correlation result stored in variable 'y'
stem(n,y);      % plot
grid on;
```

Its Output is as shown in figure 1:

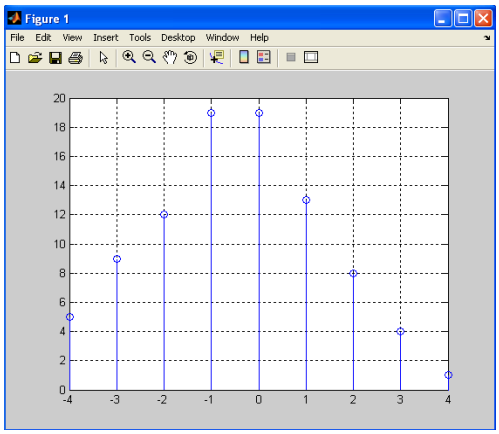
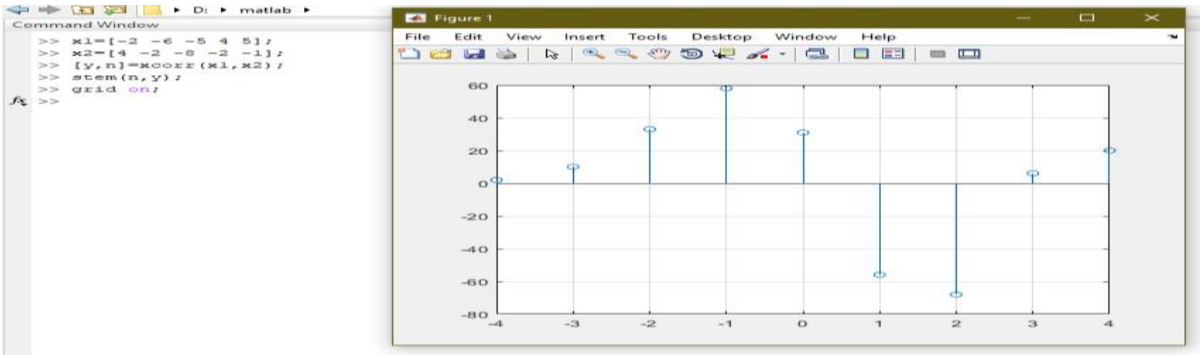


Figure 1

1. Write a MATLAB code to calculate the cross correlation result of the following 2 sequences. Label your plot using ‘xlabel’, ‘ylable’ and ‘title’ commands.

n	0	1	2	3	4
X1	-2	-6	-5	4	5
X2	4	-2	-8	-2	-1

Answer:



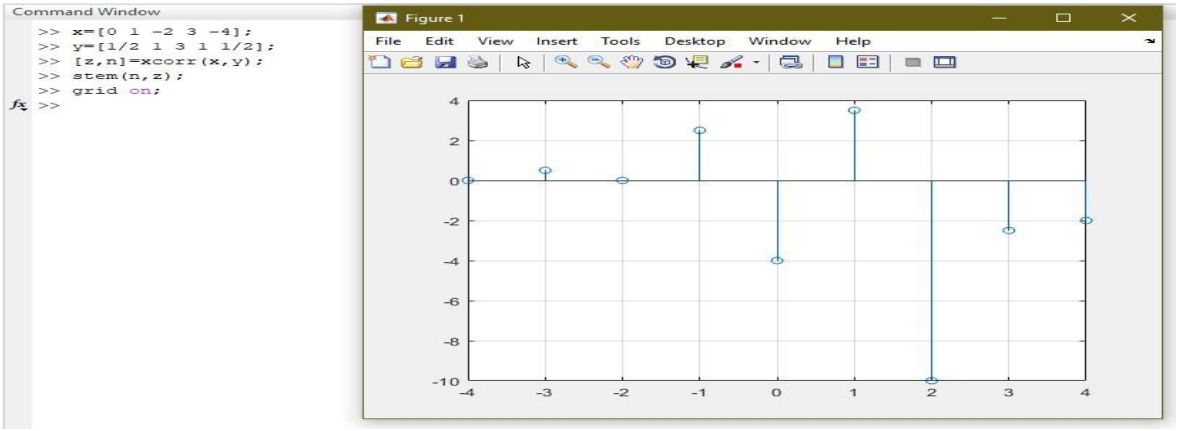
2. Compute the correlation sequences for the following pair of signals:

(a) x[n] = [0, 1, -2, 3, -4], y[n] = [1/2, 1, 2, 1, 1/2]

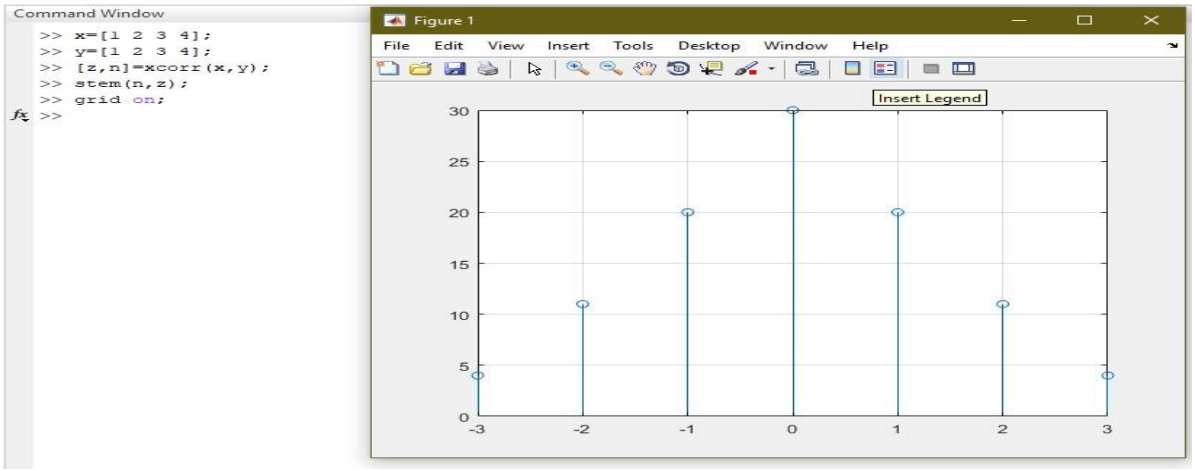
(b) x[n] =[1, 2, 3, 4], y[n] = [1, 2, 3, 4]

Answer:

(a) x[n] = [0, 1, -2, 3, -4], y[n] = [1/2, 1, 2, 1, 1/2]



(b) x[n] =[1, 2, 3, 4], y[n] = [1, 2, 3, 4]



3. State practical applications of correlation.
   Answer:

Correlation has a wide range of practical applications in various fields, including statistics, signal processing, finance, and engineering. Here are some examples:

- Signal Processing: In signal processing, cross-correlation is used to measure the similarity between two signals or to find a delay between them. It is used in audio and video processing, speech recognition, and image registration.
- correlation analysis is used to measure the correlation between the patient's blood pressure and the medication used.
- Climate Science: Correlation is used in climate science to study the relationship between different climate variables, such as temperature, precipitation, and atmospheric pressure. It is used to identify patterns and trends in climate data and to make predictions about future climate change.

MEHRAN UNIVERSITY OF ENGINEERING AND TECHNOLOGY, JAMSHORO
Department of Electronic Engineering
SIGNALS & SYSTEMS (ES-304)

LAB # 11: Z-TRANSFORM

Name	Karan vaswani	Roll #	20ES062
Signature of Lab Tutor		Date	

OBJECTIVE(S)

#	*Topic*	# Of Lectures	CLO	Taxonomy level
1	To *perform* Z-transform using matlab functions	3	4,5	P3,A4

OUTCOME(S)

• An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.	**PLO5**:Modern Tool Usage
• An ability to communicate effectively (written/oral)	**PLO10**: Communication

LAB RUBRICS:

Performance Metric	Good (5-4)	Average (3-2)	Poor (1-0)	Score
Use of modern engineering software [PLO5]	Demonstrates knowledge and application of modern engineering software through accurate development and interpretation of computer programs to solve problems.	Demonstrates awareness of modern engineering software through mostly correct development and interpretation of computer programs to solve problems, but may contain minor mistakes or syntax errors	Unable to use modern engineering software to develop or interpret computer programs to solve problems.	
Level of understanding of the learned skill [PLO5]	Demonstration of full knowledge of the handout with explanations and elaboration.	At ease with content and able to elaborate and explain to some degree.	No grasp of information. Clearly no knowledge of subject matter. No questions are answered. No interpretation made.	
Conducting simulation [PLO5]	Has an excellent simulations skill in the simulator. Always able to make the logical code for the given task.	Has a good simulation skill in the simulator. Always able to make the logical code for the given task.	Has poor simulation skill in the simulator. Unable to make the logical code for the given task.	
Responsiveness to Questions/ Accuracy [PLO10]	Responds well, quick and very accurate all the time.	Generally responsive and accurate most of the times.	Non-responsive at all or the candidate giving only one correct response of viva voce questions.	
Documentation: contents and organization [PLO10]	Report well organized, Appropriately sectioned, uses diagram/dscription when	Report reasonably well documented. May lack some minor aspects.	Report not well organized, lack key aspects.	

	appropriate, important issues clearly stated.			
				Total

EQUIPMENT/SOFTWARE TOOL:

MATLAB - The Language of Technical Computing.

DISCUSSION AND SETUP:

Z-transform is an extremely useful mathematical tool for the analysis and design of discrete time systems. In signal processing, the Z-transform converts a discrete time-domain signal into a complex frequency-domain representation.

Bilateral Z-transform

The bilateral or two-sided Z-transform of a discrete-time signal x[n] is the function X(z) defined as:

$$X(z) = \mathcal{Z}\{x[n]\} = \sum_{n=-\infty}^{\infty} x[n]z^{-n}$$

Unilateral Z-transform

In cases where x[n] is defined only for $n \geq 0$, the single-sided or unilateral Z-transform is defined as:

$$X(z) = \mathcal{Z}\{x[n]\} = \sum_{n=0}^{\infty} x[n]z^{-n}.$$

In signal processing, this definition is used when the signal is causal.

Some MATLAB functions

ztrans

z-transform.

Syntax

F = ztrans(f)

Description

F = ztrans(f) is the z-transform of the scalar symbol f with default independent variable n. The default return is a function of z.

iztrans

Inverse z-transform

Syntax

f = iztrans(F)

Description

f = iztrans(F) is the inverse z-transform of the scalar symbolic object F with default independent variable z. The default return is a function of n.

TASKS,OBSERVATIONS & RESULTS:

a) Find the z-transform:

$$x(n) = \frac{1}{4^n} u(n)$$

Code:

```
n= sym ('n');
x= (1/4)^n;
z= ztrans(x)
```

b) Find the inverse z-transform of the answer you get in a).

iz=iztrans(z)

Result:

iz =

(1/4)^n

ACTIVITY:

Find z-transform and inverse z-transform for the following:

1. x[n] = cos an
2. x[n] = 2^n
3. $x(n) = 8(3)^n u(n)$

Question No:01

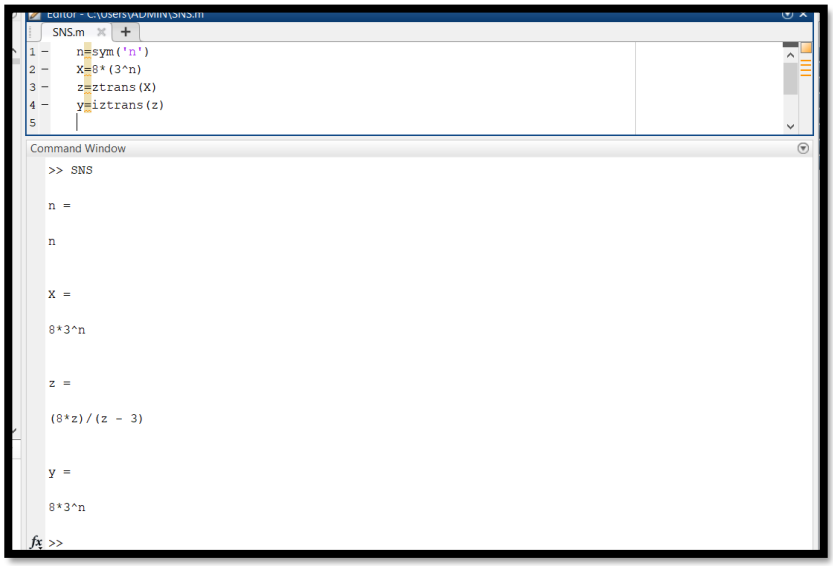
Solution:

Question No:02

Solution

Question No:3

Solution:



REVIEW QUESTIONS:

1. What is the difference between bilateral and unilateral z-transform?

The unilateral z-transform differs from the bilateral transform in that the summation is carried out only over nonnegative values of n, whether or not x[n] is zero for $n < 0$. Thus the unilateral z-transform of x[n] can be thought of as the bilateral transform of x[n]u[n] (i.e., x[n] multiplied by a unit step).

2. What is the purpose of ztrans command?

The ztrans command can be used in conjunction with other MATLAB functions to perform a variety of signal processing tasks, such as filter design, frequency analysis, and system identification. By using the z-transform and related techniques, engineers and scientists can design and analyze digital systems and signals with improved performance and reliability.

3. What is the purpose of 'sym'?

The sym function is a built-in function in MATLAB that is used to create symbolic variables or expressions. It is part of the Symbolic Math Toolbox in MATLAB.When you create a variable using the sym function, MATLAB creates a symbolic object that represents a mathematical symbol or expression. These symbolic variables or expressions can be used to perform mathematical operations symbolically, rather than numerically.

LAB # 12: ANALOG BUTTERWORTH FILTERS

Name	Karan vaswani	Roll #	20ES062
Signature of Lab Tutor		Date	

OBJECTIVE(S)

#	*Topic*	# Of Lectures	CLO	Taxonomy level
1	Plot transfer functions of Butterworth filters. Compute and plot the magnitude response of these filters.	3	4,5	P3,A4

OUTCOME(S)

• An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.	**PLO5**:Modern Tool Usage
• An ability to communicate effectively (written/oral)	**PLO10**: Communication

LAB RUBRICS:

Performance Metric	Good (5-4)	Average (3-2)	Poor (1-0)	Score
Use of modern engineering software [PLO5]	Demonstrates knowledge and application of modern engineering software through accurate development and interpretation of computer programs to solve problems.	Demonstrates awareness of modern engineering software through mostly correct development and interpretation of computer programs to solve problems, but may contain minor mistakes or syntax errors	Unable to use modern engineering software to develop or interpret computer programs to solve problems.	
Level of understanding of the learned skill [PLO5]	Demonstration of full knowledge of the handout with explanations and elaboration.	At ease with content and able to elaborate and explain to some degree.	No grasp of information. Clearly no knowledge of subject matter. No questions are answered. No interpretation made.	
Conducting simulation [PLO5]	Has an excellent simulations skill in the simulator. Always able to make the logical code for the given task.	Has a good simulation skill in the simulator. Always able to make the logical code for the given task.	Has poor simulation skill in the simulator. Unable to make the logical code for the given task.	
Responsiveness to Questions/ Accuracy [PLO10]	Responds well, quick and very accurate all the time.	Generally responsive and accurate most of the times.	Non-responsive at all or the candidate giving only one correct response of viva voce questions.	
Documentation: contents and organization [PLO10]	Report well organized, Appropriately sectioned, uses diagram/dscription	Report reasonably well documented. May lack some minor aspects.	Report not well organized, lack key aspects.	

	when appropriate, important issues clearly stated.			
			Total	

EQUIPMENT/SOFTWARE TOOL:

MATLAB - The Language of Technical Computing.

DISCUSSION AND SETUP:

Butterworth Filter:

Butterworth filter is a commonly used analogue filter. The magnitude response of an nth order Low-pass Butterworth filter is,

$$| H(jw) | = \frac{1}{\sqrt{1 + \varepsilon^2 (w / w_c)^{2N}}}$$

where w_c is the cut-off frequency and ε is a constant (see class notes for details).

The Built-in function **[num, den] = butter(N, w_n,'s')** computes the numerator and denominator of an Nth order Butterworth filter with cut-off frequency w_n.

butter(N,Wn,'s'), butter(N,Wn,'high','s') and butter(N,Wn,'stop','s') design analog Butterworth filters. In this case, Wn is in [rad/s] and it can be greater than 1.0.

For example, the transfer function of a first-order Butterworth filter with **w_n = 1 rad per second** can be computed as,

```
[num, den] = butter(1,1,'s');
printsys(num,den)
```

The magnitude response of the filter can now be computed as follows:

```
[y,w] = freqs(num,den);
% generates frequency response vector y
%  and a frequency vector w
y1 = abs(y);
% magnitude response of the filter
plot(w,y1)
```

When used with three left-hand arguments, as in [Z,P,K] = butter(...), the zeros and poles are returned in length N column vectors Z and P, and the gain in scalar K.

Exercise:

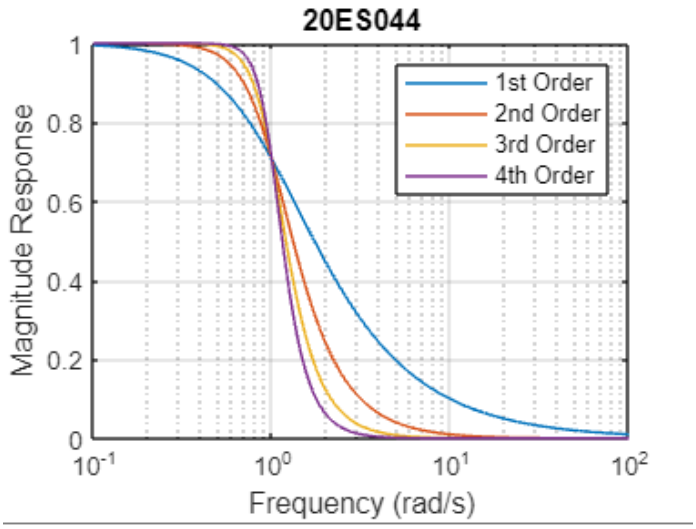
(a) Use the built-in function "hold-on" or otherwise, compute the magnitude response of first order (as computed above), second-order, third order and fourth order low-pass Butterworth filter with cut-off frequency of 1 radian per second. All plots must appear on the same graph paper. Print your Roll No. on top of the graph paper.

(b) Find poles and zeros of all of the above filters and plot them (separate graph for each filter) onto the s-plane.

(c) Repeat (a) for a high-pass Butterworth filter.

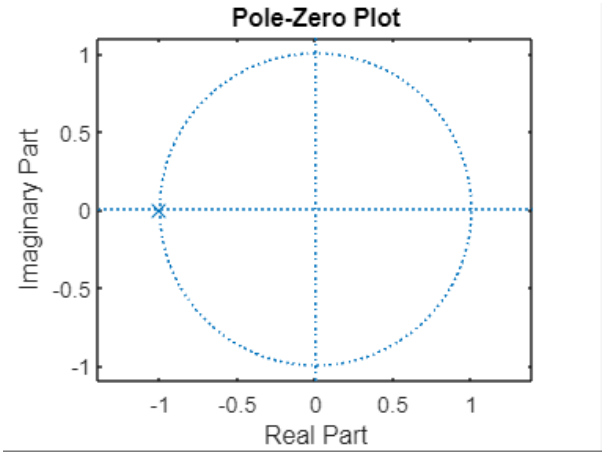
Solution: (a)

```
% Define the cutoff frequency
fc = 1;

% Define the filter orders
orders = 1:4;

% Compute the magnitude response of each filter order
w = logspace(-1, 2, 1000);
Hs = zeros(length(w), length(orders));
for i = 1:length(orders)
    Hs(:, i) = 1 ./ sqrt(1 + (w/fc).^(2*orders(i)));
end
% Plot the magnitude response of each filter order
semilogx(w, Hs)
legend('1st Order', '2nd Order', '3rd Order', '4th Order')
xlabel('Frequency (rad/s)')
ylabel('Magnitude Response')
title('20ES044')
grid on
```



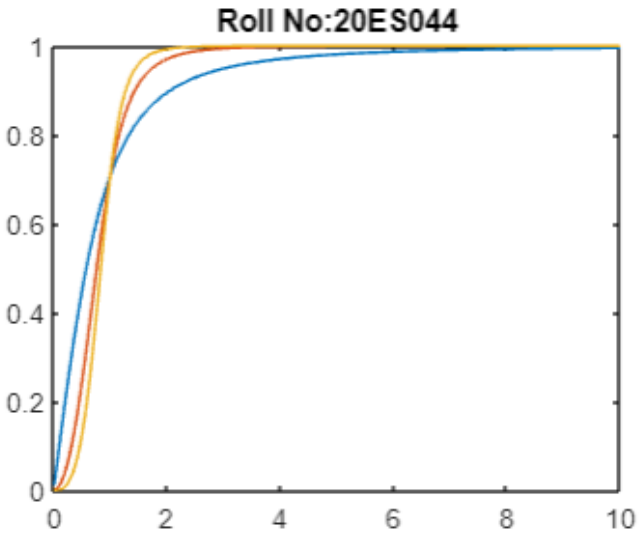
Solution: (b)

```
[z,p,k] = butter(1,1,'s');
zplane(z,p) %pole zero map on z-plane pzmap(p,z) % pole zero map on s-plane
[z1,p1,k1] = butter(2,1,'s'); figure(2)
zplane(z1,p1)
[z2,p2,k2] = butter(3,1,'s'); figure(3)
zplane(z2,p2)
```



Solution: (c)

```
[num, den] = butter(1, 1, 'high', 's');
printsys(num, den)
[y, w] = freqs(num, den);
y1 = abs(y);
plot(w, y1)
hold on

[num1, den1] = butter(2, 1, 'high', 's');
printsys(num1, den1)
[y2, w2] = freqs(num1, den1);
y3 = abs(y2);
plot(w2, y3)

[num2, den2] = butter(3, 1, 'high', 's');
printsys(num2, den2)
[y4, w4] = freqs(num2, den2);
y5 = abs(y4);
plot(w4, y5)

hold off
title('Roll No:20ES044')
```



REVIEW QUESTIONS

1.Explain Butterworth Filter.

A Butterworth filter is a type of signal processing filter designed to have a frequency response as flat as possible in the passband. It is a type of infinite impulse response (IIR) filter and is commonly used in audio and video processing applications. The Butterworth filter is also known for its maximally flat group delay, which means that it doesn't introduce any phase distortion to the filtered signal.

2. Define Filter order.

Filter order refers to the highest power of the complex variable "s" (for continuous-time filters) or "z" (for discrete-time filters) in the denominator of the transfer function of a filter. It is a measure of the complexity of the filter, and determines how well it can perform its filtering function in terms of the steepness of its roll-off and the amount of attenuation it can provide at certain frequencies.

3. What will happen if the order of the filter is increased?

If the order of the filter is increased, it will result in a steeper roll-off, which means that the filter will attenuate the frequency components beyond the cut-off frequency more rapidly. This can lead to better attenuation of unwanted frequencies but at the cost of increased complexity and computation requirements. A higher order filter will also have a narrower transition band, which can result in a higher degree of distortion or phase shift near the cut-off frequency.

LAB # 13: ANALOG CHEBYSHEV FILTERS

Name	Karan vaswani	Roll #	20ES062
Signature of Lab Tutor		Date	

OBJECTIVE(S)

#	*Topic*	# Of Lectures	CLO	Taxonomy level
1	Plot transfer functions of Chebyshev filters. Compute and plot the magnitude response of these filters.	3	4,5	P3,A4

OUTCOME(S)

• An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.	**PLO5**:Modern Tool Usage
• An ability to communicate effectively (written/oral)	**PLO10**: Communication

LAB RUBRICS:

Performance Metric	Good (5-4)	Average (3-2)	Poor (1-0)	Score
Use of modern engineering software [PLO5]	Demonstrates knowledge and application of modern engineering software through accurate development and interpretation of computer programs to solve problems.	Demonstrates awareness of modern engineering software through mostly correct development and interpretation of computer programs to solve problems, but may contain minor mistakes or syntax errors	Unable to use modern engineering software to develop or interpret computer programs to solve problems.	
Level of understanding of the learned skill [PLO5]	Demonstration of full knowledge of the handout with explanations and elaboration.	At ease with content and able to elaborate and explain to some degree.	No grasp of information. Clearly no knowledge of subject matter. No questions are answered. No interpretation made.	
Conducting simulation [PLO5]	Has an excellent simulations skill in the simulator. Always able to make the logical code for the given task.	Has a good simulation skill in the simulator. Always able to make the logical code for the given task.	Has poor simulation skill in the simulator. Unable to make the logical code for the given task.	
Responsiveness to Questions/ Accuracy [PLO10]	Responds well, quick and very accurate all the time.	Generally responsive and accurate most of the times.	Non-responsive at all or the candidate giving only one correct response of viva voce questions.	
Documentation: contents and organization [PLO10]	Report well organized, Appropriately sectioned, uses diagram/dscription	Report reasonably well documented. May lack some minor aspects.	Report not well organized, lack key aspects.	

	when appropriate, important issues clearly stated.			
			Total	

EQUIPMENT/SOFTWARE TOOL:

MATLAB - The Language of Technical Computing.

DISCUSSION AND SETUP:

Chebyshev filters are nothing but analog or digital filters. These filters have a steeper roll off & type-1 filter (more pass band ripple) or type-2 filter (stop band ripple) than Butterworth filters. The property of this filter is, it reduces the error between the characteristic of the actual and idealized filter.

[B,A] = cheby1(N,R,Wp) designs an Nth order lowpass digital Chebyshev filter with R decibels of peak-to-peak ripple in the passband. cheby1 returns the filter coefficients in length N+1 vectors B (numerator) and A (denominator). The passband-edge frequency Wp must be 0.0 < Wp < 1.0, with 1.0 corresponding to half the sample rate. Use R=0.5 as a starting point, if you are unsure about choosing R.

If Wp is a two-element vector, Wp = [W1 W2], cheby1 returns an order 2N bandpass filter with passband W1 < W < W2. [B,A] = cheby1(N,R,Wp,'high') designs a highpass filter. [B,A] = cheby1(N,R,Wp,'low') designs a lowpass filter. [B,A] = cheby1(N,R,Wp,'stop') is a bandstop filter if Wp = [W1 W2].

When used with three left-hand arguments, as in [Z,P,K] = cheby1(...), the zeros and poles are returned in length N column vectors Z and P, and the gain in scalar K.

cheby1(N,R,Wp,'s'), cheby1(N,R,Wp,'high','s') and cheby1(N,R,Wp,'stop','s') design analog Chebyshev Type I filters. In this case, Wp is in [rad/s] and it can be greater than 1.0.

The transfer function of a low-pass type1 chebyshev filter of order 1 with cut off frequency of 1 radian per second and a ripple of 0.5 dB in passband can be determined as follows:

```
[num,den] = cheby1(1, 0.5, 1,'s');
disp('The Transfer Function of the filter is')
printsys(num,den)
```

The magnitude response of the filter can be obtained as,

```
[y,w] = freqs(num,den);
y1 = abs(y);
plot(w,y1)
```

Exercise:

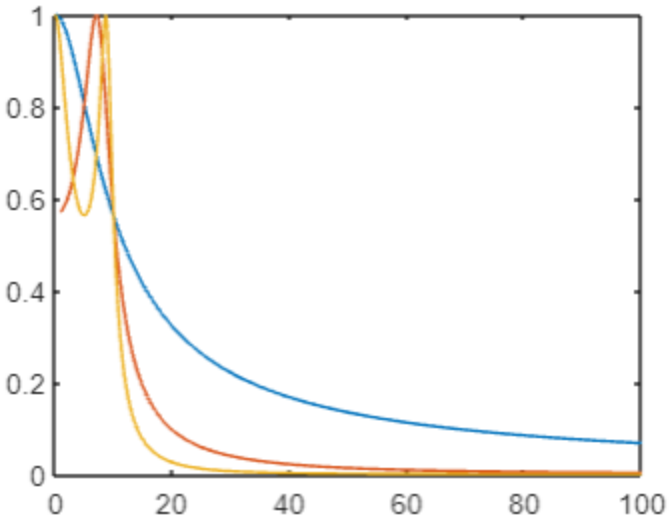
(a) Plot the magnitude and phase response of a first-order, second-order, third-order and fourth order type1 low-pass chebyshev filter with passband edge frequency of 10 radians per second and passband ripple of 5 db. Investigate the effect of changing order on the magnitude response of the system.

(b) Draw magnitude response of a third order low-pass type1 chebyshev filter when the passband ripple is 0.5db, 1 db, 2 db, 3.5 db and 5 db. What is the effect of changing passband ripple on the magnitude response?

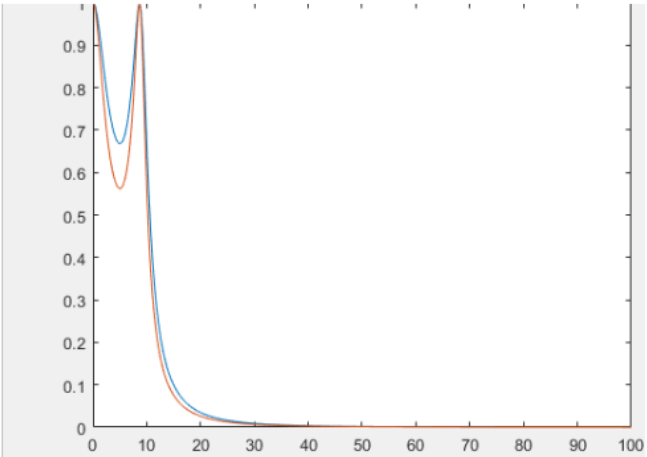
(c) Repeat (a) for a high-pass type 1 Chebyshev filter.

Solution: (a)

```
[num,den] = cheby1(1, 5, 10,'s');
printsys(num,den)
[y,w] = freqs(num,den);
y1 = abs(y);
plot(w,y1)
hold on
[num1, den1] = cheby1(2,5,10,'s');
printsys(num1,den1)
[y2,w2] = freqs(num1,den1);
y3 = abs(y2);
plot(w2,y3)
hold on
[num2, den2] = cheby1(3,5,10,'s');
printsys(num2,den2)
[y4,w4] = freqs(num2,den2);
y5 = abs(y4);
plot(w4,y5)
hold off
title('Roll No: 20ES062')
```

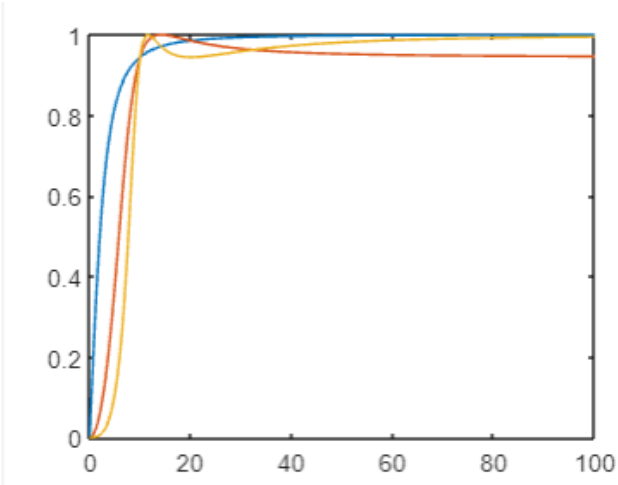


Solution: (b)



Solution: (c)

```
[num,den] = cheby1(1, 0.5, 10,'high','s');
printsys(num,den)
[y,w] = freqs(num,den); y1 = abs(y);
plot(w,y1)
hold on
[num1, den1] = cheby1(2, 0.5, 10,'high','s');
printsys(num1,den1)
[y2,w2] = freqs(num1,den1); y3 = abs(y2);
plot(w2,y3)
hold on
[num2, den2] = cheby1(3, 0.5, 10,'high','s');
printsys(num2,den2)

[y4,w4] = freqs(num2,den2);
y5 = abs(y4);
plot(w4,y5)
hold off
title('Roll No: 20ES062')
```



Review Questions

1.Explain Chebyshev Filter.

Chebyshev filter is a type of filter used in signal processing that has a steeper roll-off and a sharper transition region between the passband and stopband compared to other types of filters, such as Butterworth filters. This is achieved by allowing ripples in the passband or stopband of the filter's frequency response. Chebyshev filters are characterized by the amount of ripple in the passband, which can be controlled by adjusting the filter's design parameters.

2.What is the difference between type 1 and type 2 chebyshev filters?

The primary difference between type 1 and type 2 Chebyshev filters is the location of the ripple in the frequency response. In a type 1 Chebyshev filter, the passband has ripple while the stopband is flat. On the other hand, in a type 2 Chebyshev filter, the stopband has ripple while the passband is flat.

Another difference is that the poles of a type 1 Chebyshev filter are all located on the left half of the S-plane, while the poles of a type 2 Chebyshev filter alternate between the left and right half of the S-plane.

3.What will happen if the order of the filter is increased?

Filter order refers to the number of poles (for analog filters) or the number of taps (for digital filters) in a filter. It is a measure of the complexity of the filter and determines its frequency response characteristics. Increasing the filter order generally results in a steeper roll-off rate, narrower transition band, and a better attenuation of unwanted frequencies. However, higher order filters may also introduce more phase distortion and require more computational resources.

MEHRAN UNIVERSITY OF ENGINEERING AND TECHNOLOGY, JAMSHORO
Department of Electronic Engineering
SIGNALS & SYSTEMS (ES-304)

LAB # 14: INTRODUCTION TO SPEECH PROCESSING

Name	Karan vaswani	Roll #	20ES062
Signature of Lab Tutor		Date	

OBJECTIVE(S)

#	*Topic*	# Of Lectures	CLO	Taxonomy level
1	Introduction to Speech Processing in Matlab	3	4,5	P3,A4

OUTCOME(S)

• An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.	**PLO5**:Modern Tool Usage
• An ability to communicate effectively (written/oral)	**PLO10**: Communication

LAB RUBRICS:

Performance Metric	Good (5-4)	Average (3-2)	Poor (1-0)	Score
Use of modern engineering software [PLO5]	Demonstrates knowledge and application of modern engineering software through accurate development and interpretation of computer programs to solve problems.	Demonstrates awareness of modern engineering software through mostly correct development and interpretation of computer programs to solve problems, but may contain minor mistakes or syntax errors	Unable to use modern engineering software to develop or interpret computer programs to solve problems.	
Level of understanding of the learned skill [PLO5]	Demonstration of full knowledge of the handout with explanations and elaboration.	At ease with content and able to elaborate and explain to some degree.	No grasp of information. Clearly no knowledge of subject matter. No questions are answered. No interpretation made.	
Conducting simulation [PLO5]	Has an excellent simulations skill in the simulator. Always able to make the logical code for the given task.	Has a good simulation skill in the simulator. Always able to make the logical code for the given task.	Has poor simulation skill in the simulator. Unable to make the logical code for the given task.	
Responsiveness to Questions/ Accuracy [PLO10]	Responds well, quick and very accurate all the time.	Generally responsive and accurate most of the times.	Non-responsive at all or the candidate giving only one correct response of viva voce questions.	

Documentation: contents and organization [PLO10]	Report well organized, Appropriately sectioned, uses diagram/dscription when appropriate, important issues clearly stated.	Report reasonably well documented. May lack some minor aspects.	Report not well organized, lack key aspects.	
			Total	

EQUIPMENT/SOFTWARE TOOL:

MATLAB - The Language of Technical Computing.

DISCUSSION AND SETUP:

Introduction

Speech signal processing refers to the acquisition, manipulation, storage, transfer and output of vocal utterances by a computer. The main applications are the recognition, synthesis and compression of human speech.

Speech recognition (also called voice recognition) focuses on capturing the human voice as a digital sound wave and converting it into a computer-readable format. Speech synthesis is the reverse process of speech recognition. Advances in this area improve the computer's usability for the visually impaired. Speech compression is important in the telecommunications area for increasing the amount of information which can be transferred, stored, or heard, for a given set of time and space constraints.

Analysis and Synthesis of Speech

A speech signal is usually represented in digital format, which is a sequence of binary bits. For storage and transmission applications, it is desirable to compress a signal by representing it with as few bits as possible, while maintaining its perceptual quality. In narrowband digital speech compression, speech signals are sampled at a rate of 8000 samples per second. Typically, each sample is represented by 8 bits. This corresponds to a bit rate of 64k bits per second. Further compression is possible at the cost of quality. Most of the current low bit rate speech coders are based on the principle of linear predictive speech coding.

MATLAB functions with examples

1. Audioread and audiowrite

Example:

Create a WAVE file from the example file handel.mat, and read the file back into MATLAB®.
Create a WAVE (.wav) file.

```
%load matlab builtin sound file named handel.mat
load handel.mat;

%convert .mat file to audio .wav file
audiowrite('d:\audio1.wav',y,Fs);
```

```
%read the information about .wav file
info = audioinfo('d:\audio1.wav');

info

% to listen to the new .wav file
[y1,Fs1]=audioread('d:\audio1.wav');

sound(y1,Fs1);
```

2. **audiorecorder**

wavrecord has been removed in recent versions of MATLAB. If you want your code to be compatible with newer versions of MATLAB, use ***audiorecorder***.

Record sound using a PC-based audio input device.

Syntax

y = wavrecord(n,Fs)

Description

y = wavrecord(n,Fs) records n samples of an audio signal, sampled at a rate of Fs Hz (samples per second). The default value for Fs is 11025 Hz.

Example:

Collect a sample of your speech with a microphone, and plot the signal data:

```
% Record your voice for 5 seconds.
y = audiorecorder;
disp('start specking')
recordblocking(y,5);
disp('end of recording')

% Play back the recording.
play(y)

% Store data in double-precision array.
myRecording = getaudiodata(y);

% Plot the waveform.
plot(myRecording);
xlabel('time');
ylabel('Audio signal');
```

3. **audioplayer**

Syntax

`player = audioplayer(Y,Fs)` creates an `audioplayer` object for signal `Y`, using sample rate `Fs`

Example:
Load and play a sample audio file of Handel's "Hallelujah Chorus:"

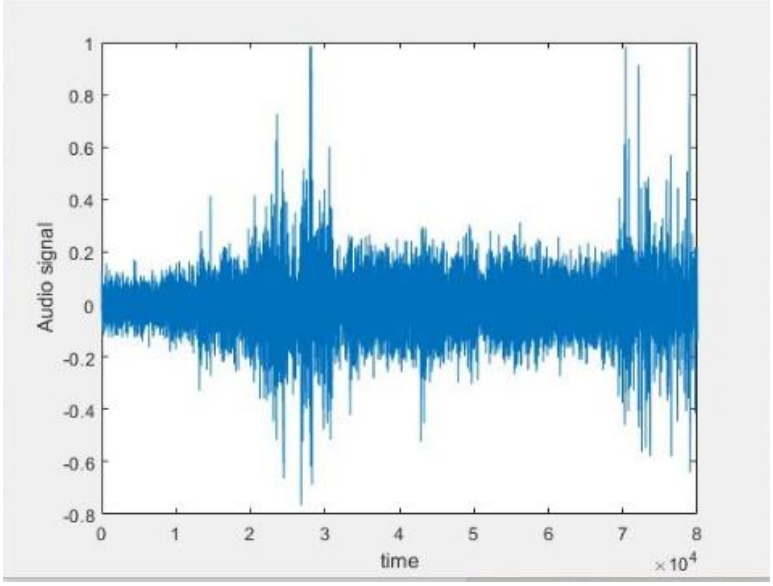
```
load handel;
k = audioplayer(y, Fs);
play(k);
```

Exercise:

1. record your own voice for 10 seconds
2. play the recorded sound file
3. plot your sound file

Solution:

```
Jamshed = audiorecorder;
disp('start specking')
recordblocking(Jamshed,10 );
disp('end of recording')
play(Jamshed)
Jamshed_Memon= getaudiodata(karan vaswani);
plot(karan vaswani);
xlabel('time');
ylabel('Audio signal');
```



MEHRAN UNIVERSITY OF ENGINEERING AND TECHNOLOGY, JAMSHORO
Department of Electronics System Engineering
SIGNALS & SYSTEMS

LAB # 15: OPEN ENDED LAB

Name **Karan**	Roll # 20ES062
Signature of Lab Tutor	Date

OBJECTIVE(S)

#	*Topic*	# Of Lectures	CLO	Taxonomy level
1	To Perform Filters using MATLAB Functions	3	4,5	P3,A4

OUTCOME(S)

• An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.	**PLO5**:Modern Tool Usage
• An ability to communicate effectively (written/oral)	**PLO10**: Communication

LAB RUBRICS:

Performance Metric	Good (5-4)	Average (3-2)	Poor (1-0)	Score
Use of modern engineering software [PLO5]	Demonstrates knowledge and application of modern engineering software through accurate development and interpretation of computer programs to solve problems.	Demonstrates awareness of modern engineering software through mostly correct development and interpretation of computer programs to solve problems, but may contain minor mistakes or syntax errors	Unable to use modern engineering software to develop or interpret computer programs to solve problems.	
Level of understanding of the learned skill [PLO5]	Demonstration of full knowledge of the handout with explanations and elaboration.	At ease with content and able to elaborate and explain to some degree.	No grasp of information. Clearly no knowledge of subject matter. No questions are answered. No interpretation made.	
Conducting simulation [PLO5]	Has an excellent simulations skill in the simulator. Always able to make the logical code for the given task.	Has a good simulation skill in the simulator. Always able to make the logical code for the given task.	Has poor simulation skill in the simulator. Unable to make the logical code for the given task.	
Responsiveness to Questions/ Accuracy [PLO10]	Responds well, quick and very accurate all the time.	Generally responsive and accurate most of the times.	Non-responsive at all or the candidate giving only one correct response of viva voce questions.	
Documentation: contents and organization [PLO10]	Report well organized, Appropriately sectioned, uses diagram/dscription when appropriate, important issues clearly stated.	Report reasonably well documented. May lack some minor aspects.	Report not well organized, lack key aspects.	

Total	

EQUIPMENT/SOFTWARE TOOL:

MATLAB - The Language of Technical Computing.

DISCUSSION AND SETUP:

Introduction:

Lab Tasks

Sinusoids using butterworth, chebyshev1, chebyshev2 and FIR filter. Please run them and generate plots. Youwill note that they all have unit amplitude

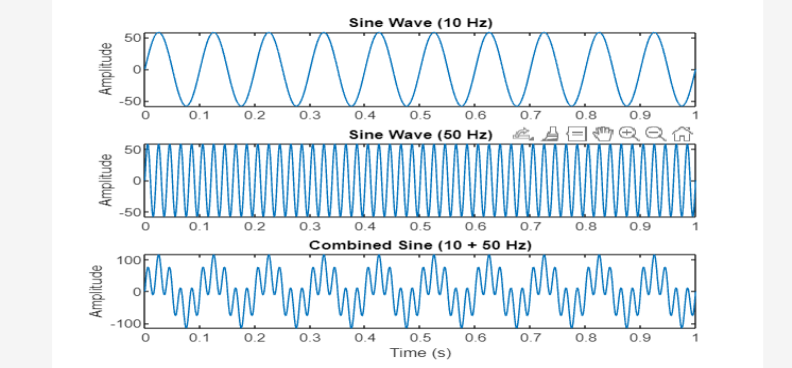
BUTTERWORTH FILTER:

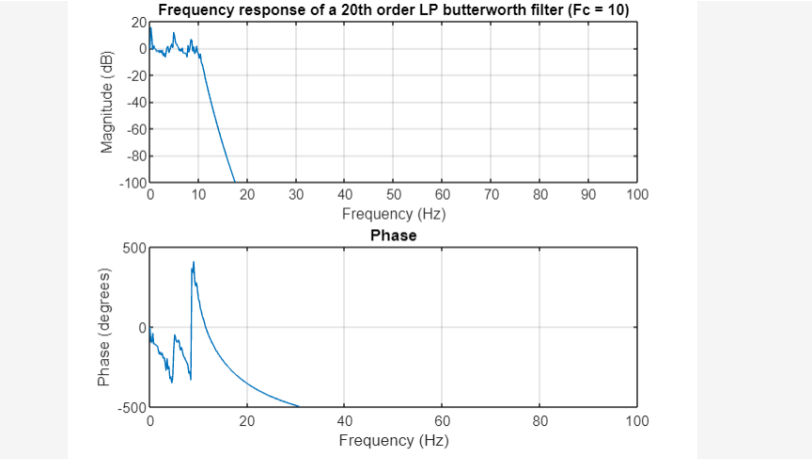
Code:

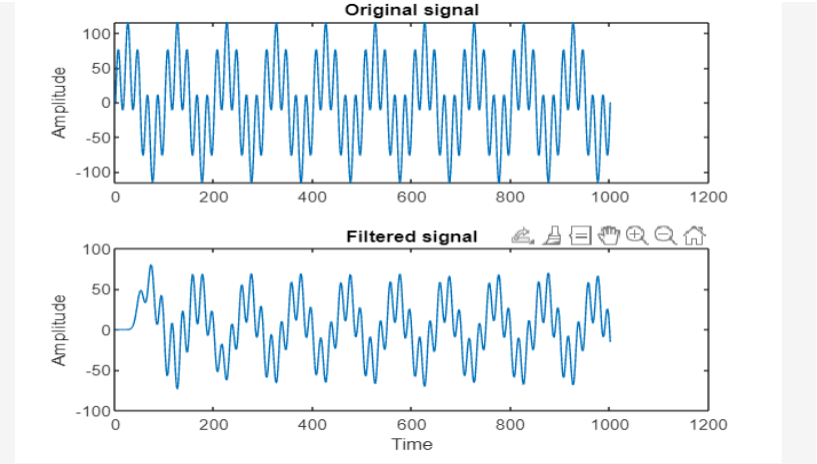
```
%Filtering using a butterworth filter
clear;
%Generate a sine wave
% Define the parameters
freq1 = 10;      % Frequency of the sine wave (in Hz)
freq2 = 50;      % Frequency of the sine wave (in Hz)
amp =62;        % Amplitude of the sine wave
time = 0:0.001:1; % Time vector (from 0 to 1 second with 0.001 sec step)


% Generate the sine wave1
sine_wave1 = amp*sin(2*pi*freq1*time);
sine_wave2 = amp*sin(2*pi*freq2*time);

combined_sine = sine_wave1 + sine_wave2;

%Plot the sine waves
figure;
subplot(311)
plot(time, sine_wave1);
title("Sine Wave (10 Hz)")
ylabel('Amplitude');


subplot(312)
plot(time, sine_wave2);
title("Sine Wave (50 Hz)")
ylabel('Amplitude');



subplot(313)
plot(time, combined_sine);
title("Combined Sine (10 + 50 Hz)")


xlabel('Time (s)');
```

```
ylabel('Amplitude');


% Create a butterworth filter to remove the 50 Hz sine wave,what kind of
% filter do we need

%Cutoff frequency we need to use
fc = 10;

%Set sampling frequency of signal
fs = 200;

%Construct the butterworth filter
[b,a] = butter(20,fc/(fs/2));

%Plot the response
figure;
freqz(b,a,[],fs)
subplot(2,1,1)
title("Frequency response of a 20th order LP butterworth filter (Fc = 10)")
ylim([-100 20])


%Filter the combined signal
filtered_combined_sine = filter(b,a,combined_sine);
figure;
subplot(211)
plot(combined_sine)
ylabel("Amplitude")
title('Original signal')

subplot(212)
plot(filtered_combined_sine)
title('Filtered signal')
ylabel("Amplitude")
xlabel("Time")
%%
dataIn = randn(1000,1);
dataOut = filter(b,a,dataIn);
```

Simulation:


Sine Wave (10 Hz)
Amplitude
Sine Wave (50 Hz)
Amplitude
Combined Sine (10 + 50 Hz)
Amplitude
Time (s)


Frequency response of a 20th order LP butterworth filter (Fc = 10)
Magnitude (dB)
Frequency (Hz)
Phase
Phase (degrees)
Frequency (Hz)


Original signal
Amplitude
Filtered signal
Amplitude
Time

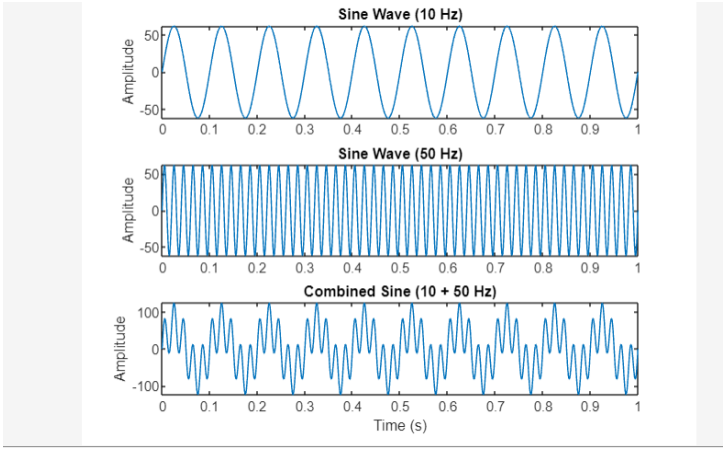
CHEBYSHEV1 FILTER:

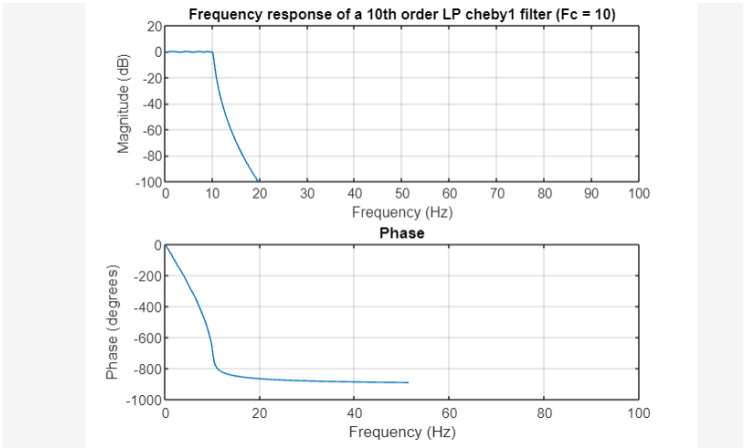
Code:

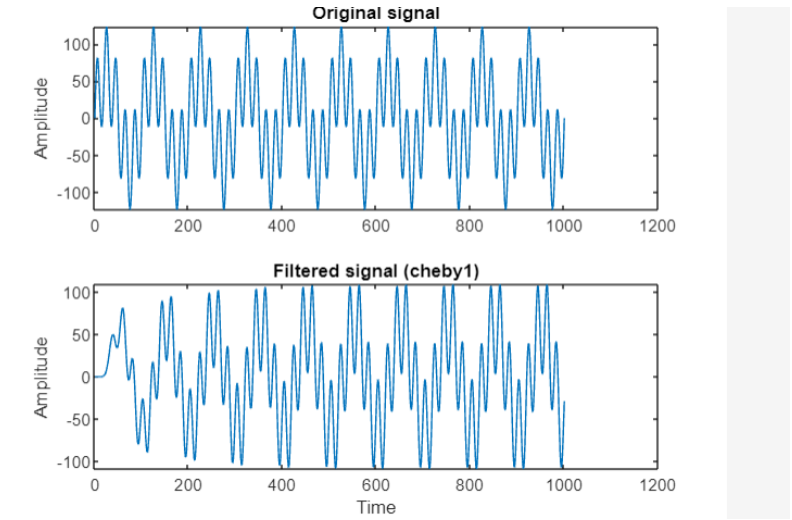
```
clear;
% Generate a sine wave
% Define the parameters
freq1 = 10;      % Frequency of the sine wave (in Hz)
freq2 = 50;      % Frequency of the sine wave (in Hz)
amp = 62;        % Amplitude of the sine wave
time = 0:0.001:1; % Time vector (from 0 to 1 second with 0.001 sec
step)

% Generate the sine wave1
sine_wave1 = amp*sin(2*pi*freq1*time);
sine_wave2 = amp*sin(2*pi*freq2*time);
combined_sine = sine_wave1 + sine_wave2;

% Plot the sine waves
figure;
subplot(311)
plot(time, sine_wave1);
title("Sine Wave (10 Hz)")
ylabel('Amplitude');

subplot(312)
plot(time, sine_wave2);
title("Sine Wave (50 Hz)")
ylabel('Amplitude');

subplot(313)
plot(time, combined_sine);
title("Combined Sine (10 + 50 Hz)")
xlabel('Time (s)');
ylabel('Amplitude');

% Create a chebyshev1 filter to remove the 50 Hz sine wave
fc = 10; % Cutoff frequency we need to use
fs = 200; % Set sampling frequency of signal
[b,a] = cheby1(10,0.5, fc/(fs/2)); % Construct the cheby1 filter

% Plot the filter response
figure;
freqz(b,a,[],fs)
subplot(2,1,1)
title("Frequency response of a 10th order LP cheby1 filter (Fc = 10)")
```

```
ylim([-100 20])

% Filter the combined signal
filtered_combined_sine = filter(b,a,combined_sine);
figure;
subplot(211)
plot(combined_sine)
ylabel("Amplitude")
title('Original signal')

subplot(212)
plot(filtered_combined_sine)
title('Filtered signal (cheby1)')
ylabel("Amplitude")
xlabel("Time")
```

Simulation:







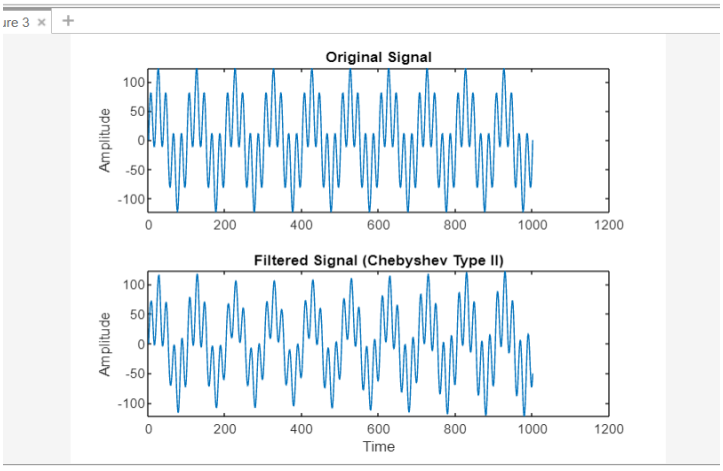
- **CHEBYSHEV2 FILTER:**

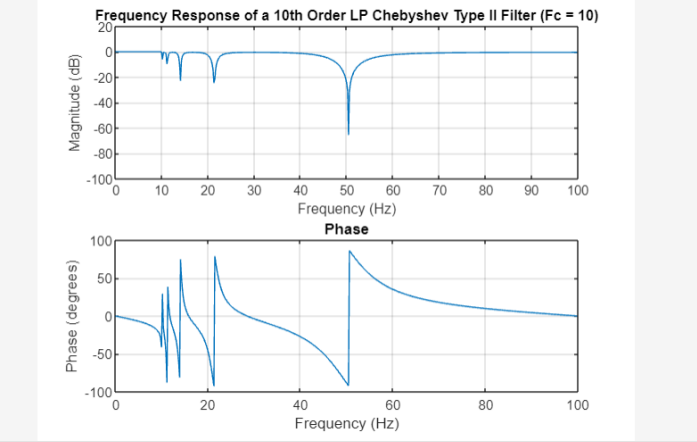
Source Code:

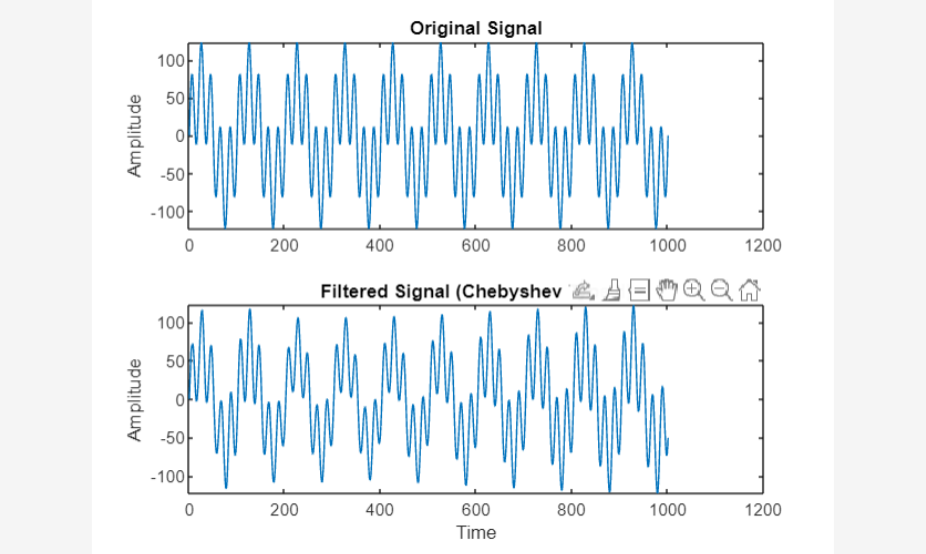
```
clear;

% Define the parameters
freq1 = 10;      % Frequency of the first sine wave (in Hz)
freq2 = 50;      % Frequency of the second sine wave (in Hz)
amp = 62;        % Amplitude of the sine waves
time = 0:0.001:1; % Time vector (from 0 to 1 second with 0.001 sec step)

% Generate the sine waves
sine_wave1 = amp*sin(2*pi*freq1*time);
sine_wave2 = amp*sin(2*pi*freq2*time);

% Combine the sine waves
combined_sine = sine_wave1 + sine_wave2;

% Plot the sine waves
figure;
subplot(311)
plot(time, sine_wave1);
title("Sine Wave (10 Hz)")
ylabel('Amplitude');

subplot(312)
plot(time, sine_wave2);
title("Sine Wave (50 Hz)")
ylabel('Amplitude');
subplot(313)
plot(time, combined_sine);
```

```
 title("Combined Sine (10 + 50 Hz)")
 xlabel('Time (s)');
 ylabel('Amplitude');
 % Create a Chebyshev Type II filter to remove the 50 Hz sine wave
 fc = 10;          % Cutoff frequency
 fs = 200;         % Sampling frequency
 [b,a] = cheby2(10,0.5, fc/(fs/2));  % Construct the filter
 % Plot the filter response
 figure;
 freqz(b,a,[],fs)
 subplot(2,1,1)
 title("Frequency Response of a 10th Order LP Chebyshev Type II Filter (Fc =
10)")
 ylim([-100 20])

 % Filter the combined signal
 filtered_combined_sine = filter(b,a,combined_sine);

 % Plot the original and filtered signals
 figure;
 subplot(211)
 plot(combined_sine)
 ylabel("Amplitude")
 title('Original Signal')
 subplot(212)
 plot(filtered_combined_sine)
 title('Filtered Signal (Chebyshev Type II)')
 ylabel("Amplitude")
 xlabel("Time")
```

Simulation:







- **FIR FILTER:**

Code:

```
% Define the filter specifications
passband_freq = 20; % Hz
stopband_freq = 40; % Hz
Fs = 200; % Hz

% Determine the filter order
f_norm = passband_freq / (Fs/2);
filter_order = 4 / f_norm;

% Design the filter using FIR1
b = fir1(filter_order, f_norm, 'low');
```

```
% Generate a test signal with a 50 Hz component
t = 0:1/Fs:1;
x = 62*sin(2*pi*50*t) + sin(2*pi*10*t); % 62 is my roll number

% Apply the filter to the signal
y = filter(b, 1, x);

% Plot the original and filtered signals
figure;
subplot(2,1,1);
plot(t, x);
xlabel('Time (s)');
ylabel('Amplitude');
title('Original Signal');
subplot(2,1,2);
plot(t, y);
xlabel('Time (s)');
ylabel('Amplitude');
title('Filtered Signal');
```

Simulation: