NED University of Engineering & Technology

Department of Electrical Engineering

LAB MANUAL

For the course

SIGNALS AND SYSTEMS

(EE-232) For S.E.(EE)

Instructor name:_______________________________________

Student name:____

Roll no:_______________________ **Batch:**______________________

Semester:_________________________ **Year:**__________________

LAB MANUAL

For the course

<u>SIGNALS AND SYSTEMS</u>

<u>(EE-232) For S.E.(EE)</u>

Content Revision Team:
Dr. Krishan Lal Khatri, Engr. Rashid Hussain

Approved By

The Board of Studies of Department of Electrical Engineering

__________________________ __________________________

__________________________ __________________________

__________________________ __________________________

Summary of Rubric Sheets for Data Entry in OBE MIS System [to be entered as they are on OBE portal with no scaling]

Lab #	Extent of achievement	Lab #	Extent of achievement	Lab #	Extent of achievement
	Total out of 32		Total out of 32		Total out of 32

Lab #	Extent of achievement	Lab #	Extent of achievement	Lab #	Extent of achievement
	Total out of 32		Total out of 32		Total out of 32

Add six total scores from above to get total Rubric score Out of 192	R=
Scaled to 25 marks (R/192)*25	S=
Lab Attendance percentage from portal:	A=
Attendance scaled to 5 marks: (A/100)*5	P=
For entry on NED exam portal Total lab sessionals out of 30: S + P	T=

PBL/OEL	Extent of achievement	Final	Extent of achievement
	Total out of 32		Total out of 32

Add two total scores from above to get total final Rubric Score Out of 64	Z=
For entry on NED exam portal Total Final Lab Marks scaled to 20 marks (Z/64)*20	F=

CONTENTS

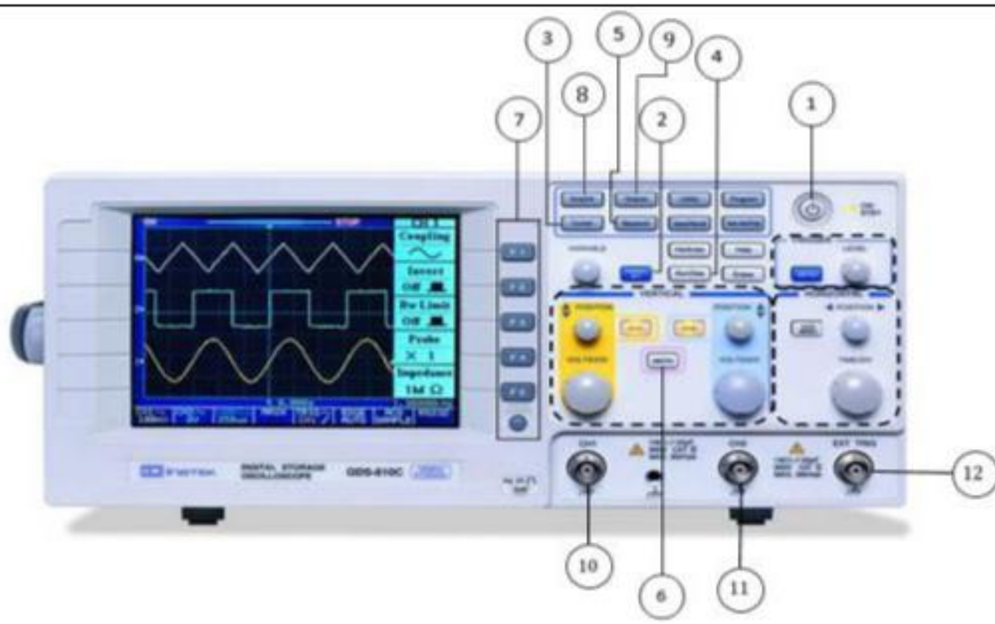
Laboratory Session No. 01

Objective:

To get introduced with the basic features and functions of Oscilloscope, function generator and DC power supply.

Theory:

Introduction to Oscilloscope:
It is an instrument which is used to display the waveform or time vs value curves of electrical signals.



Front View of Oscilloscope

Control	Function	Purpose
1.	Power Button	Turn the oscilloscope on and off.
2.	Auto-set	It automatically identifies suitable scales (horizontal and vertical) for the channels.
3.	Cursor	It measures the time domain characteristics of a circuit.
4.	Run/Stop	It freezes the waveform until the run/stop button is pressed again.
5.	Measure	It measures two types of characteristics; Time characteristics and voltage characteristics.
6.	Math	Used to define math waveform using the math menu. This operation allows us to add two waveforms together, creating a new waveform display.
7.	Channel Parameters	Used to access signal parameters which includes coupling, invert, probe, impedance. These parameters can be changed by function keys F1, F2, F3, F4 and F5.
8.	Acquire	Used to control how waveform points are generated from sample points referring to digital values taken directly from ADC.
9.	Display	Used to access screen display parameters which includes dots and vectors, accumulate, refresh, contrast, grid and bandwidth. These parameters can be changed by function keys F1, F2, F3, F4 and F5.
10.	Channel1	Used to detect signal from the Oscilloscope.
11.	Channel2	Used to detect signal from the Oscilloscope.
12.	EXT. Trigger	Used to detect signal from the Oscilloscope with repetitive transient wave forms.
		Characteristics of Oscilloscope

a) Basic Purpose

It is an instrument which is used to display the waveform or time vs value curves of electrical signals.

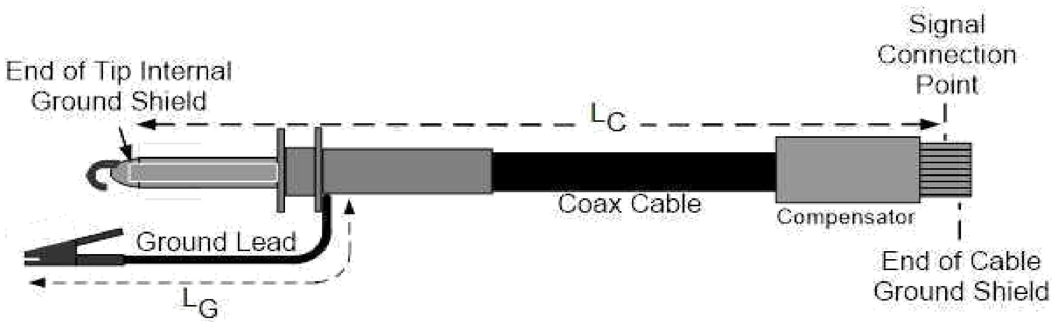
Setting Up Connections;

Before using the Oscilloscope, the proper grounding is very important for safety purpose, not for the user but also for testing the integrated circuits (ICs), necessary because if a high voltage contacts with any grounded part, the current travels through the grounding path to earth, rather harming the user.

Connecting the Probes;

Once the Oscilloscope the properly grounded, the probes are to be connected which will enable a user to access all of the power and performance in the Oscilloscope and will ensure the integrity of the Signal which is to be measure. It requires two connections to measure a Signal.

➢ Passive Oscilloscope Probe: In passive Oscilloscope probe there are no active components such as transistors or amplifiers in the probe and therefore passive probes do not need to be powered.



Probe of an Oscilloscope

Most of the Oscilloscope have at least two input channels and each channel display a waveform on the screen. Both of the channels are used to compare the waveforms. Digital Oscilloscope has AUTOSET button, which is used to produce a stable waveform when waveform scale is larger than the screen scale. During setting up the Oscilloscope make sure the following things are done.

b) Features Explored

1. Vertical Controls

The Vertical control is used to set or modify the Vertical Scale, position and other signal conditioning for each of the analog input channels. There is a set of vertical controls for each input channel. These controls are used to scale, position, and modify that channel's input signal so it can be viewed appropriately on the oscilloscope display.

2. Horizontal Controls

The horizontal controls are used to scale and position the time axis of the oscilloscope display. There is a dedicated front-panel control for setting the horizontal scale (time/division) of the display and another for setting the horizontal position of the displayed signals.

3. Trigger Controls

The trigger defined when a signal is acquired and stored in memory. For a repetitive signal, a trigger is required to stabilize the display.

4. Channel Parameters

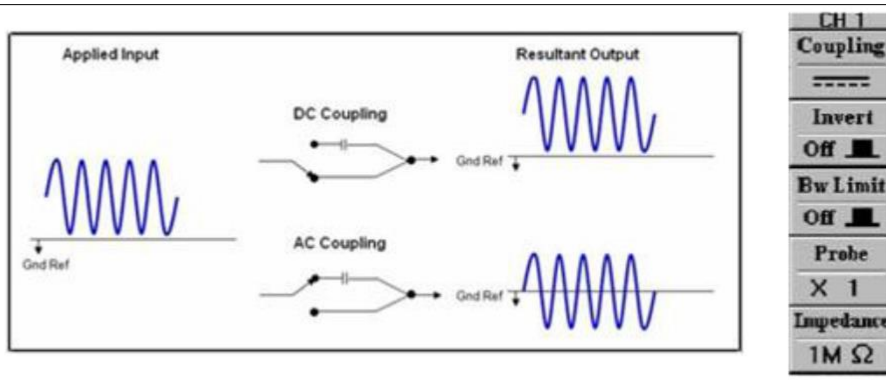
For selecting channel parameters press CH button, a menu will open. It will have several more parameters and those parameters can be changed by function keys F1, F2, F3, F4 and F5.

Coupling: In our case the coupling is the connection from function generator to the oscilloscope. Input coupling determines which part of the signal is displayed. The coupling can be set to DC, AC and ground.

i) **DC COUPLING:** DC coupling allows all the components of input signal. DC offset will be considered. When using DC coupling, no additional capacitor is added to filter the signal.

ii) **AC COUPLING:** AC coupling blocks DC component of the signal, centering the waveform at 0 volts. If the signal will have any constant value of DC offset then it will move the waveform upward of origin. AC coupling consists of a capacitor to filter out the DC signal component from a signal with both AC and DC components. The capacitor must be in series with the signal. AC coupling is useful because the DC component of a signal acts as a voltage offset and removing it from the signal can increase the resolution of signal measurements. AC coupling is also known as capacitive coupling.

iii) **GROUND:** Ground coupling disconnects the input signal to show where 0 volts is on the screen, it means no signal will be read at ground. We can easily bring position of signal or channel to the central reference line or mean after selecting ground.



AC and DC Coupling

Invert: It is used to change the orientation of the voltage signal i.e. negative voltages become positive and positive voltages become negative.

Probe: It is basically the multiplication factor of the signal. We can multiply our signal by 1, 10 and 100.

THE 1X PROBE: The 1X probes are suitable for low frequency applications, they offer the same impedance as the oscilloscope which is 1M ohm.

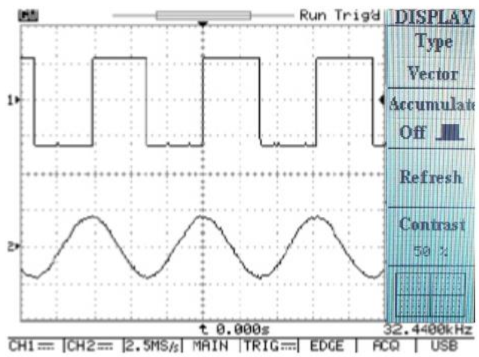
THE 10X PROBE: The 10X probe will reduced our signal by ten it means that the signal entering the oscilloscope itself is reduced.

THE 100X PROBE: The 100X probes are used when very high voltages are need to be monitored.

Impedance: Every oscilloscope will add a certain impedance to a circuit called the input impedance which is generally represented as a large resistive impedance of 1Mohm.

Oscilloscope can bear 1MV /1A. If more than 1 ampere current is applied, it can damage the oscilloscope.

5. Display



Display Characteristics

1. **Dots and Vectors:** We can represent our waveform in dots and in vectors. Dot drawing displays the waveform in points form. Vector drawing connects the dot with the lines.

2. **Accumulate:** Press the second function button accumulates will be ON. When accumulate is ON our waveform is recorded continuously on the display. If our waveform is changing with time then it will show our previous image until accumulate is switched OFF by pressing the second function button again.

3. **Refresh:** Refresh is selected by pressing the third function button.

4. **Contrast:** Contrast is selected by pressing the fourth function button. To increase the contrast, rotate the VARIABLE knob clockwise and to decrease the contrast rotate the VARIABLE knob counter clockwise.

5. **Grid:** Grid is selected by pressing the fifth function button. There are three grid settings; full, cross and frame. In full grid, the axes and all grid lines are displayed. In cross grid, only the axes are displayed. In frame grid, no axes and grid lines are displayed.

6. **Bandwidth:** Oscilloscopes are most commonly used to measure waveforms which have a defined frequency. The bandwidth of the oscilloscope specifies the range of the frequencies. Most oscilloscope have a circuit that limits the bandwidth of an oscilloscope. By limiting the bandwidth, we can reduce the noise that sometimes appears on the displayed waveform which results in a cleaner display signal.

6. Cursor

The cursors allow you to get very accurate readings of either time or voltage differences between different parts of a wave. Its application is in under-damped, over-damped, critically damped circuits etc.

There are two types of cursors:

1. **Horizontal Cursor:** It is parallel to x-axis (time axis). It is selected by pressing the F2 key. There are two horizontal cursors. Select the one you want to move with the function key and vary it with the help of variable knob. It can be used to obtain voltage value of small part of a waveform etc.

2. Vertical Cursor: It is parallel to y-axis (voltage axis). It is selected by pressing the F3 key. There are two vertical cursors. Select the one you want to move with the function key and vary it with the help of the variable knob. It can be used to obtain peak values, peak to peak values and other time domain characteristics.

7. Measure

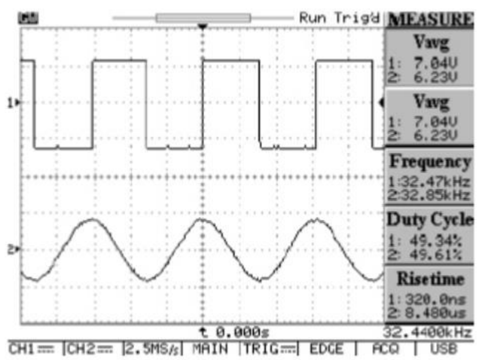


Fig. 1.9: Measure Characteristics

With the help of measure button, we can measure two types of characteristics: Time characteristics and voltage characteristics.

1. Time Characteristics: -

i. Frequency and Time period: Frequency is defined as the number of cycles in 1 second while time period is its reciprocal. We can measure frequency and time period of a wave form with the help of an oscilloscope. It can be selected with the help of function key parallel to it.

ii. Duty Cycle: Duty cycle is defined as the percentage of on time to total fundamental time. It is a property of a square wave. It can be measured through an oscilloscope. It can be selected with the help of function key parallel to it.

iii. Rise and Fall time: The duration of a wave going from a low point to a high point is called rise time. It is important when we determine how fast a circuit can respond to signals. It can be measured through an oscilloscope. It can be selected with the help of function key parallel to it.

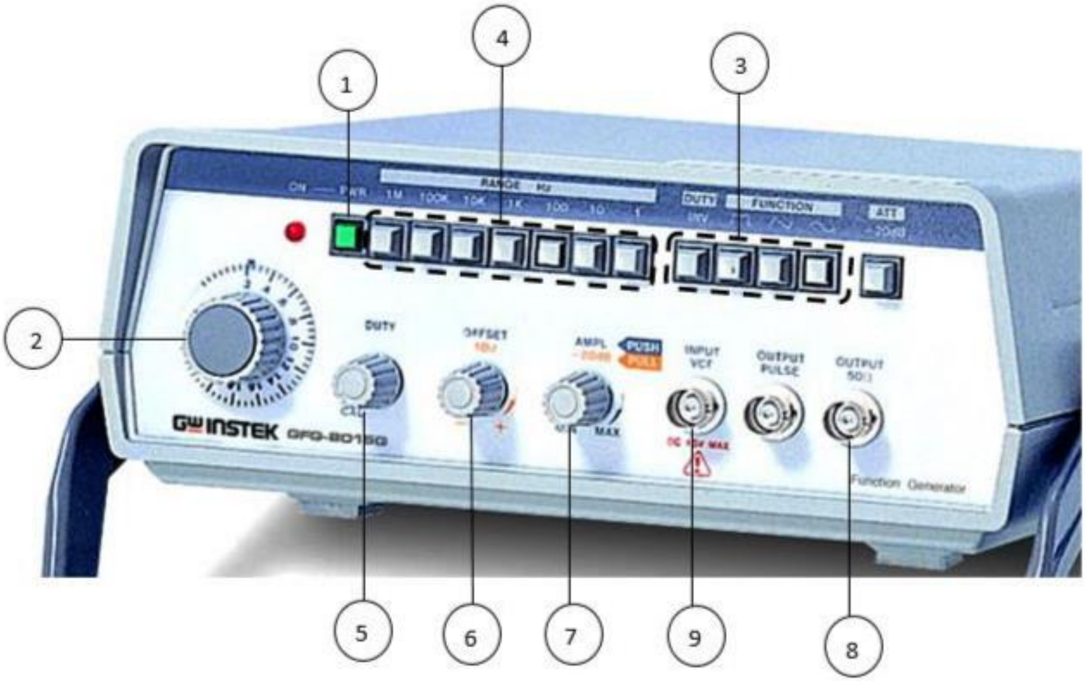
2. Voltage Characteristics: -

i. Amplitude: It is the measure of the magnitude of a signal. There are two types of amplitudes: Peak voltage and peak to peak voltage. Peak voltage is the measure from 0V to the peak value. Peak to peak voltage is the measure from negative peak to positive peak. It can be measured through an oscilloscope. It can be selected with the help of function key parallel to it.

ii. Maximum and Minimum Voltage: Oscilloscope can tell you exactly how high and low the voltage of your signal can get. Voltage from peak to peak can be calculated as Vpp=Vmax-min.

iii. Average and Vrms Voltage: Oscilloscope can measure the mean and root mean square value of your signal and it can also tell the mean and average value of your signal's maximum and minimum voltage. Also, Oscilloscope helps to calculate Vamp=Vhigh-Vlow.

Introduction to Function Generator



Analog Function Generator

Control	Function	Purpose
1.	Power Button	Turn the Function Generator on and off.
2.	Frequency Dial	Determines the frequency of the function generator output, within the range set by the MULTIPLIER buttons.
3.	Function Selection Switches	Used to select the type of waveform generated as sine, triangle, or square.
4.	Multiplier Buttons	Set the frequency range.
5.	Duty Cycle Knob	Adjusts the range of frequencies that are traversed by each sweep.
6.	DC Offset Knob	Sets the DC level (and therefore the polarity) of the MAIN OUT signal.
7.	Amplitude Control Knob	Adjusts the voltage within the presently selected range.
8.	Output Port	It is used to get the desirable output waveform through channel probes (having impedance of 50 Ω).
9.	Voltage Dependent Frequency Port	It takes voltage as an input and with the change of voltage its value changes.

Characteristics of Analog Function Generator

Analog Function Generator

Control	Function	Purpose
1.	Power Switch	Turns the Instrument ON and OFF.
2.	Setting adjustment knob	Adjust the parameter selected by the other buttons.
3.	Sine Wave Selection	Selects Sine Wave output.
4.	Counter Wave Selection Port	Selects Sine wave output.
5.	Ramp wave Selection Port	Selects ramp (triangle) wave output.
6.	Modulation signal output Port	Input terminal for external modulation signal.
7.	Square wave Selection Port	Selects square wave output.
8.	Synchronization signal output	Provides a signal (Square wave or pulse) that is in phase with the output signal.
9.	Amplitude-offset adjustment	Knob to adjust either the signal amplitude or DC offset voltage.
10.	Signal output	Output terminal for the function generator signal.
11.	Set to counter mode	Enables the counter input and displays the frequency of the input signal on item 4.
12.	Change utility settings	Adjust frequency sweep start frequency, sweep stop frequency, and display intensity.
13.	Set DC offset	Enable the adjustment of the DC voltage added to the signal output (Control 10).
14.	Select Modulation	Selects number of modulation, internal AM modulation, FM modulation deviation, and external FM modulation.
15.	% duty Cycle	Adjusts the duty cycle or symmetry of the displayed waveform.
16.	Sweep	Turns the frequency sweep mode ON and OFF.
17.	Frequency	After pressing this button, the adjustment knob (Control 2) will adjust.

18.	Mode	Selects the type of operations: continues output, trigger repetition rate (sets the interval between the internal trigger; each trigger signal causes the generator to output one period), external trigger, manual trigger (pressing the → button causes one cycle to be output), external gated (waveform cycles are output while the gate signal is above a threshold).
19.	Digit Adjustment ←	Moves the digit selection left.
20.	Digit Adjustment →	Moves the digit selection right.
21.	Display	Shows the function generator's settings, such as frequency, amplitude, waveform, etc.

Characteristics of Digital Function Generator

a) Basic Purpose

A function generator is an electronic instrument used to generate different types of electrical waveforms over a wide range of frequencies.

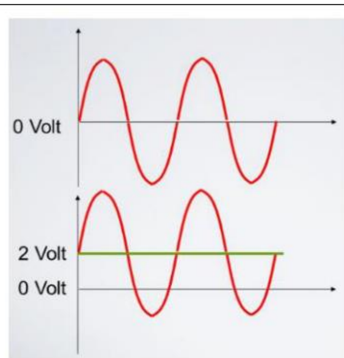
Setting Up Connections;

Power on the generator and select the desired output signal: square wave, sine wave or triangle wave. Connect the output leads to an oscilloscope to visualize the output signal and set its parameters using the amplitude and frequency controls. Attach the output leads of the function generator to the input of the circuit you wish to test. Attach the output of your circuit to a meter or oscilloscope to visualize the resulting change in signal.

b) Features Explored

1. DC Offset

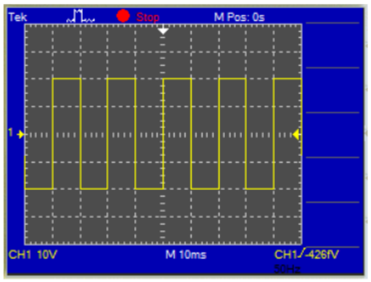
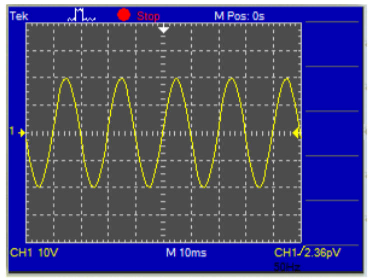
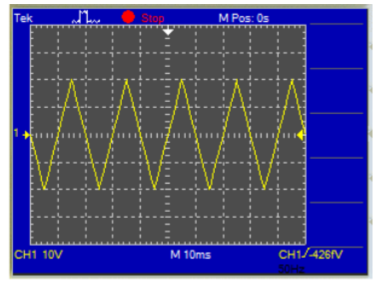
DC offset is a mean amplitude displacement from zero. With this utility of function generator, we can clamp up or down the output wave form. It adds a specific amount of DC voltage to the time-varying waveform. It basically changes the DC reference level.



DC Offset

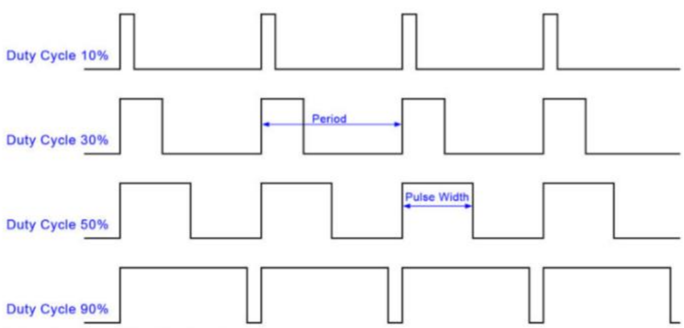
2. Types of Waves

Following are the waves produced by the function generator;

		
Sine Wave	*Square Wave*	*Triangular Wave*

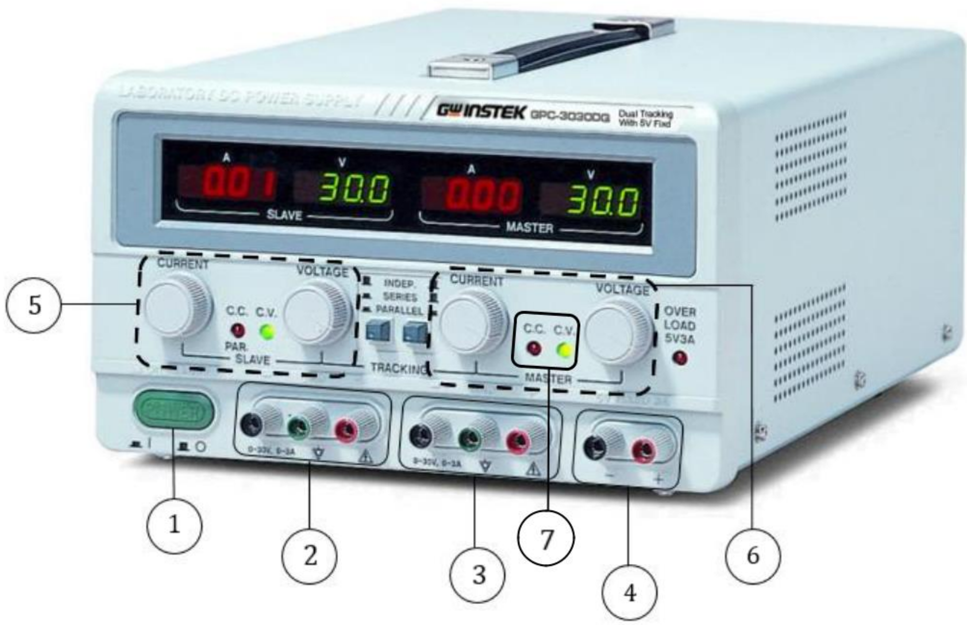
3. Duty Cycle

The term **duty cycle** describes the proportion of 'on' time to the total fundamental time. For 90% it is 'ON' time and for 10% it is for 'OFF' time. It is only for square waves.



Duty Cycle

Introduction to Power Supply



DC Power Supply

Control	Function	Purpose
1.	Power Button	Turn the Power Supply on and off.
2.	Slave Battery (CH2) Terminals	Use to connect the source to a circuit provided that the correct polarity conventions are obeyed.
3.	Master Battery (CH1) Terminals	Use to connect the source to a circuit provided that the correct polarity conventions are obeyed.
4.	Output Terminals	Used as common power supply.
5.	Slave Battery Knobs	Used to control voltage and Current given by Slave Battery
6.	Master Battery Knobs	Used to control voltage and Current given by Master Battery.
7.	CV and CC LED indicators	Represent the Constant Voltage and Constant Current mode.

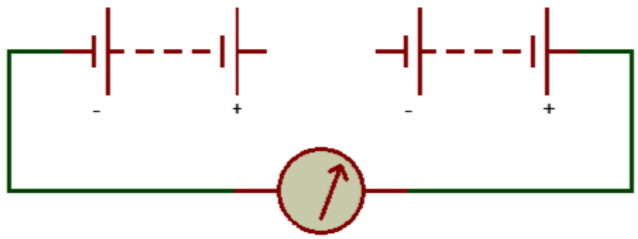
Characteristics of DC Power Supply

a) Basic Purpose

It is an instrument use to supply electric power to electric loads, also it is used to convert current draw from electric source to appropriate voltage, current and frequency to power the load because of this ability power supplies are called electric power converters.

Setting Up Connections;

In power supply, there are 2 independent voltage sources which are internally connected. For internal connections the buttons are available. These independent sources can be connected in series and parallel. To achieve higher voltage, these sources are connected in series.



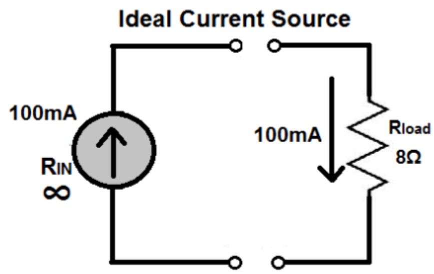
Internal View of Power Supply

b) Features Explored

1. Controlling Current

Two voltage sources are present in a power supply, if we short positive and negative terminals of anyone of the voltage source we can control the current to maintain the healthy condition of the circuit. We can set a current by rotating a knob basically we are selecting its input impedance that how much current we want to flow through the circuit. If the value of current exceeds the value which we have set our circuit can destroy. To prevent this condition, we short the two terminals to safe our circuit. At voltage zero we set the maximum value of current which circuit can bear.

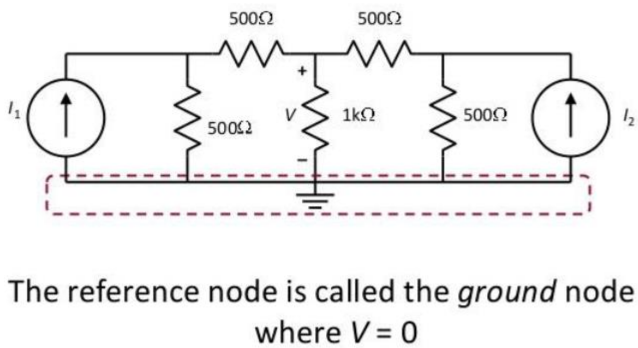
In a voltage source, voltage is controlled but infinite current is drawn. We can use short circuit method for infinite current. Similarly, in current source we should have controlled current. To maintain constant current, we need to apply infinite impedance. As impedance becomes infinite voltage also becomes infinite between terminals. Due to infinite voltage it starts flowing without any medium because of this device destroys itself that is why we do not have any current source practically but it can exist in any branch or in any circuit as shown in Fig. 1.17.



Ideal Current Source

2. Reference Node

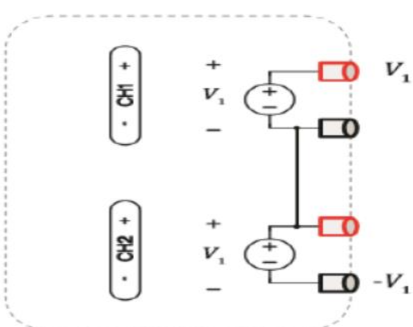
Reference node is a node with maximum number of elements. Generally, grounds have the most elements and will therefore be chosen as the reference node. The ground of a circuit is only a reference for the circuit and it is common for the whole circuit. By referencing everything to a common ground, different circuit components are able to interact with each other else the connection is supposed to be incomplete due to the reason that a common reference point is not provided to the circuit.



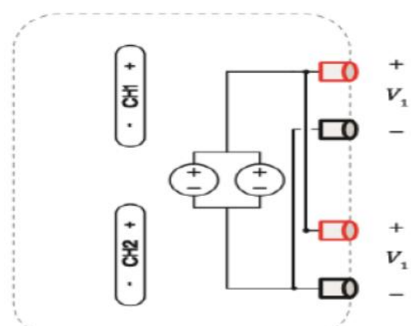
Reference Node

3. Master and Slave Mode

We can control one battery by another battery, connecting them in parallel combination to act as Master and Slave. The Fig. 1.15 illustrates the Channel1 as Master and Channel2 as Slave, connecting these batteries in parallel combination as shown in Fig. 1.19(b), we can have a control on Channel2(Slave Battery). In series combination of batteries there is no such controlling or Master and Slave mode exist.



Internal Series Combination



Internal Parallel Combination

<u>POST LAB ACTIVITY:</u>

Activity: Add two waves of 50Hz and 100Hz.

Procedure:

1. Power the Oscilloscope and function generator. Use two channels of function generator to generate two signals of 50V, one with a frequency of 100Hz and another with 50Hz.
2. Select Sine wave as input signal by the function generator.
3. Connect two signals in series so that the signals can add.
4. Connect wires to complete the rest of the circuitry.
5. Select the reference node (ground) to which the reference is common to the whole circuit.
6. Connect probes to display the signal on the oscilloscope and use Oscilloscope to align the signal.
7. Check the insights of the signal on the oscilloscope and use multimeter to verify the voltages at different nodes.

OBSERVATION

CONCLUSION

NED University of Engineering & Technology
Department of Electrical Engineering

Course Code and Title: ____

Laboratory Session: No.______________________________ Date: ____________________

Psychomotor Domain Assessment Rubric-Level P3					
Skill Sets	Extent of Achievement				
	0	1	2	3	4
Equipment Identification Sensory skill to identify equipment and/or its component for a lab work.	Not able to identify the equipment.	--	--	--	Able to identify equipment as well as its components.
Equipment Use Sensory skills to describe the use of the equipment for the lab work.	Never describes the use of equipment.	Rarely able to describe the use of equipment.	Occasionally describe the use of equipment.	Often able to describe the use of equipment.	Frequently able to describe the use of equipment.
Procedural Skills Displays skills to act upon sequence of steps in lab work.	Not able to either learn or perform lab work procedure.	Able to slightly understand lab work procedure and perform lab work.	Able to somewhat understand lab work procedure and perform lab work.	Able to moderately understand lab work procedure and perform lab work.	Able to fully understand lab work procedure and perform lab work.
Response Ability to imitate the lab work on his/her own.	Not able to imitate the lab work.	Able to slightly imitate the lab work.	Able to somewhat imitate the lab work.	Able to moderately imitate the lab work.	Able to fully imitate the lab work.
Observation's Use Displays skills to perform related mathematical calculations using the observations from lab work.	Not able to use lab work observations into mathematical calculations.	Able to slightly use lab work observations into mathematical calculations.	Able to somewhat use lab work observations into mathematical calculations.	Able to moderately use lab work observations into mathematical calculations.	Able to fully use lab work observations into mathematical calculations.
Safety Adherence Adherence to safety procedures.	Doesn't adhere to safety procedures.	Slightly adheres to safety procedures.	Somewhat adheres to safety procedures.	Moderately adheres to safety procedures.	Fully adheres to safety procedures.
Equipment Handling Equipment care during the use.	Doesn't handle equipment with required care.	Rarely handles equipment with required care.	Occasionally handles equipment with required care	Often handles equipment with required care.	Handles equipment with required care.
Group Work Contributes in a group-based lab work.	Never participates.	Rarely participates.	Occasionally participates and contributes.	Often participates and contributes.	Frequently participates and contributes.

Laboratory Session No. 02

Objective:

Get introduced with the MATLAB software.

Theory:

Defining Variables

- **Delimit to suppress command window results**

>> a = 2; % Initializes and defines in a single step

*[**Note:** Semicolon's are not used if user want to display value on command window.]*

```
>> a = 2

   a =

          2

>> y = 3+a;    % Creates variable y
>> z = y+a;    % Creates variable y
>> clear a;  % Deletes variable a
>> clear all;  % Delete all variables in single step
```

***********************Above is the command window output***********************

*[**Note:*** Variable names should start from letter, MATLAB default variables cannot be used such *as pi, i, ans, NaN etc.]*

Basic Operations

```
>> a=2;b=3;c=4;  % Initialize some variables
>> a+b     %Addition

    ans =

                5

>> a-b     %Subtraction

    ans =

               -1

>> a*b     %Multiplication

    ans =

                6

>> a/b     %Division

    ans =

               0.6667

>> a\b     %Inverse

    ans =

               1.5000


>> a+b-a*c/a\b

    ans =

                 0.5000
```

**********************Above is the command window output**********************

[**Note:** *All operations can be used as at the same time simultaneous.]*

Numerical Formats

```
>> format short  % Display only five Digits
>>pi

    ans =

        3.1416

>> format long  % Display upto 15 Digits
>>pi

    ans =

        3.141592653589793

>> format rat  % Replaces number by its closest rational number
>>pi

    ans =

        355/113

>> format short e  % expresses numbers in power of 10
>>pi

    ans =

        3.1416e+00
```

*************************Above is the command window output*************************

More on Mathematical Operations

```
>>sqrt(3) % get square root of given argument

    ans =

        1.7321

>>3^(1/2) % Evaluates number for any given power

    ans =

        1.7321

>>log(1000) %Natural Logarithm

    ans =

        6.9078
```

```
>> log10(1000)  %Log to the base 10

    ans =

         3
>> cos(60*pi/180) %Trigonometric Functions are evaluated with arguments in radians
    ans =


         0.5000

>>acos(0.5) % Cosine inverse

    ans =

         1.0472

>>floor(3.3) % Round Toward Negative Infinity

    ans =

         3

>>ceil(4.23) % Round Toward Positive Infinity

    ans =

         5

>>round(0.5) % Round to the nearest Integer or Decimal

    ans =

         1

>> abs(1+i) % Evaluates magnitude of a complex number
    ans =


         1.4142

>>angle(1+i)*180/pi %Evaluates Angular distance from '+x' axis for 2-D problems

    ans =

         45
```

***********************Above is the command window output************************

Introduction to Vectors and its Operations

```
>> a=[1;2;3]  %Initialize a 3-D vector

    a =

            1
            2
            3

>>b=[4,5,6] %Initialize a row-vector b

    =

            4     5  6

>>a' %Transpose of a vector

    ans =

            1     2  3

>> c=2*a  %Multiplication by a scalar

    c =

            2
            4
            6

>>a+2* a % Addition of two vectors

    ans =

            3
            6
            9

>> sum(a)     % Adds all elements of a vector

    ans =

            6

>>prod(b) %Multiply all elements

    ans =

           120
```

***********************Above is the command window output***********************

```
start = 0;
final = 1;
increment = 0.2;
x = start: increment: final

>> x = 0 0.2000 0.4000 0.6000 0.8000 1.0000;
```

Code 01: Generates equally row-vector

• Element wise operation

```
>> y = x.^2     %Each Element gets squared

    y =

          0   0.0400    0.1600    0.3600    0.6400    1.0000

>> c = y.*x     %Element wise Multiplication

    c =

          0 0.0080 0.0640 0.2160 0.5120 1.0000 >> d =

y./x %Element wise division

    d =

         NaN   0.2000    0.4000    0.6000    0.8000    1.0000

>> e = y'*x     %Matrix Multiplication

    e =

          0        0          0        0       0        0
          0   0.0080     0.0160  0.0240   0.0320   0.0400
          0   0.0320     0.0640  0.0960   0.1280   0.1600
          0   0.0720     0.1440  0.2160   0.2880   0.3600
          0   0.1280     0.2560  0.3840   0.5120   0.6400
          0   0.2000     0.4000  0.6000   0.8000   1.0000

>> f = y*x'     % Matrix Multiplication

    f =

        1.8000

>> length(x)     %Count number of elements in a vector

    ans =

         6
```

```
>> max(x)       % Find maximum value in vector

    ans =

         1

>> min(x)       % Find maximum value in vector

    ans =

         0

>> sqrt(sum((x.^2))) %Determine magnitude of higher order vector
    ans =


        1.4832

>> sqrt(((x*x'))) % Alternate method but efficient
    ans =


        1.4832

>>size(x)

    ans =

         1    6
```

***********************Above is the command window output***********************

Introduction to Matrix and its Operations

```
>> A = [1 2 3;4 5 6;7 8 9]      % Learn to initialize Matrix

    A =

         1     2   3
         4     5   6
         7     8   9

>> c = A (1,2) % Address element of 1st row and 2nd column
```

```
>>B=2*A       % Multiplication by scalar

    B =

             2     4     6
             8      10  12
            14     16     18

>>C = A + B % Matrix Addition C

    =

             3     6      9
            12     15     18
            21     24     27

>> D = A*B'    % Matrix Multiplication

    D =

            28    64 100
            64   154 244
           100 244 388

>> E = A.*B     %Element wise multiplication


    E =

             2     8      18
            32     50    72
            98    128   162

>> d = A(:,2)     % Access all row but 2nd column


    d =

             2
             5
             8

>> e = A(2,:)     % Access all columns but second row


    e =

             4     5      6
```

**********************Above is the command window output***********************

Plotting Command

```
>> t = 0 : 0.001 : 0.5;
>> y = sin(2*pi*30*t);
>> plot(t,y)     % plots labeled graph
>> plot(t,y), xlabel('Time'), ylabel('f(t)'), title('f(t)=sin(2\pi30t)'), grid on;  %plots with other attributes like labels, grid and title
>> plot(t,y, 'r'), title('sin(2\pi30t)', 'Color' ,'b'); %plots with coloured titles
```

Code 02: Plotting using MATLAB

Observation:

Advanced plotting tools

• Contour Plots

```
>> [x,y] = meshgrid(-5:.1:5, -5:.1:5);
>> z = -(x.^2 + y.^2);
>> contour(x,y,z)     %Develop 2-D contour plots from 3-D information
```

Code 03: Contour Plot from 3D information using MATLAB

Observation:

• Surface Plots

```
>> surf(x,y,z), xlabel('x'), ylabel('y'), zlabel('-(x^2+y^2)');    % Develop 3-D surface
>> surfc(x,y,z), xlabel('x'), ylabel('y'), zlabel('-(x^2+y^2)');    % Develop 3-D surface with their contour on xy plane
```

Code 04: Contour Plot from 3D information using MATLAB

Observation:

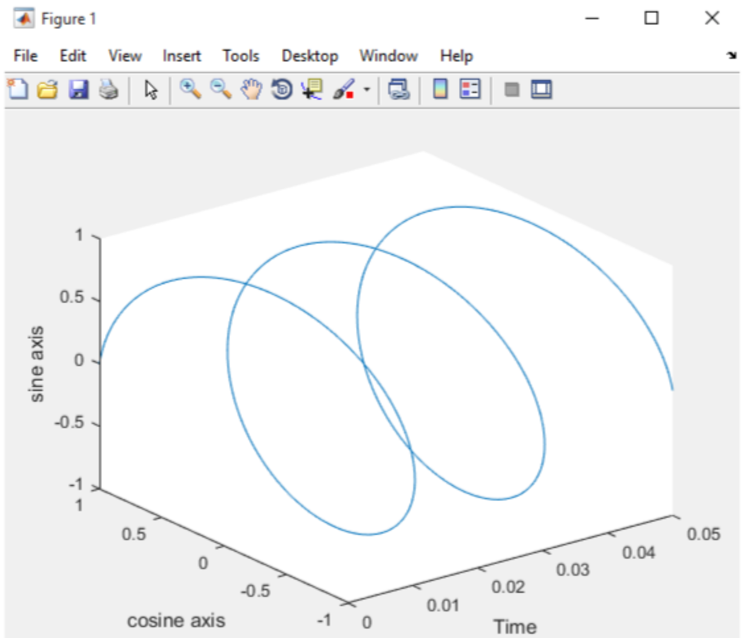
• 3-D Line Plot

```
>> clear all;
>> t = 0:0.0001:0.05;
>> x = cos(2*pi*50*t);
>> y = sin(2*pi*50*t);
>> plot3(t,x,y), xlabel('Time'),
ylabel('cosine axis'), zlabel('sine axis')
```

Observation:

MATLAB Plotting GUI: -

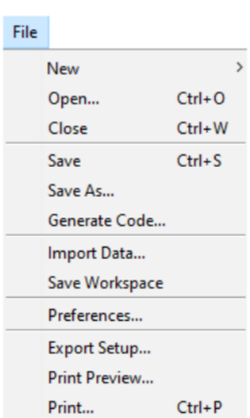
Fig. 2.6 illustrates the GUI (Graphical User Interface) of MATLAB plotting. GUI helps to customize the plotted figures; labels, orientation, colours and many of its insights.



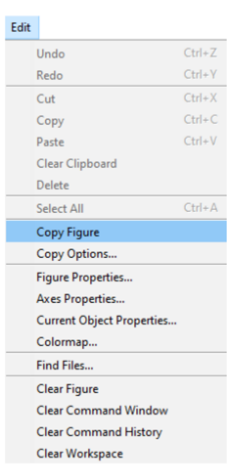
GUI

Fig.2.7 depicts the menu bar and tool bar of GUI of plotting figures. Below Following fig. 2.8 shows the options of menu bar. They are very helpful to copy image with high quality, label axis, Zoom In-Out, scaling etc. On a side, tool bar has shortcut keys like 3D rotation, coloring, Zoom In-Out etc.

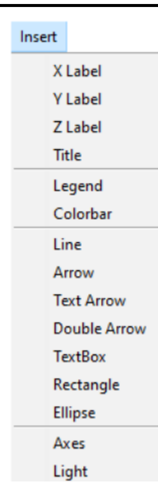
Menu bar and Tool bar



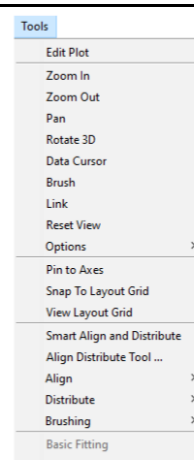
File Menu



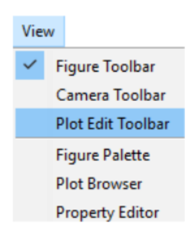
Edit Menu



Insert Menu



Tools Menu



View Menu

POST LAB ACTIVITY:

Activity 1:
For given Matrices:

$$A= \begin{bmatrix} 2 & 9 & 0 & 0 \\ 0 & 4 & 1 & 4 \\ 7 & 5 & 5 & 1 \\ 7 & 8 & 7 & 4 \end{bmatrix} \quad b= \begin{bmatrix} -1 \\ 6 \\ 0 \\ 9 \end{bmatrix} \quad a= \begin{bmatrix} 3 & -2 & 4 & -5 \end{bmatrix}$$

Find results of:

a) A.b
b) a+4
c) b.a
d) $a.b^T$
e) $A.a^T$

Activity 2:
Difference between a^2, a.^2 & a*a where 'a' is any matrix

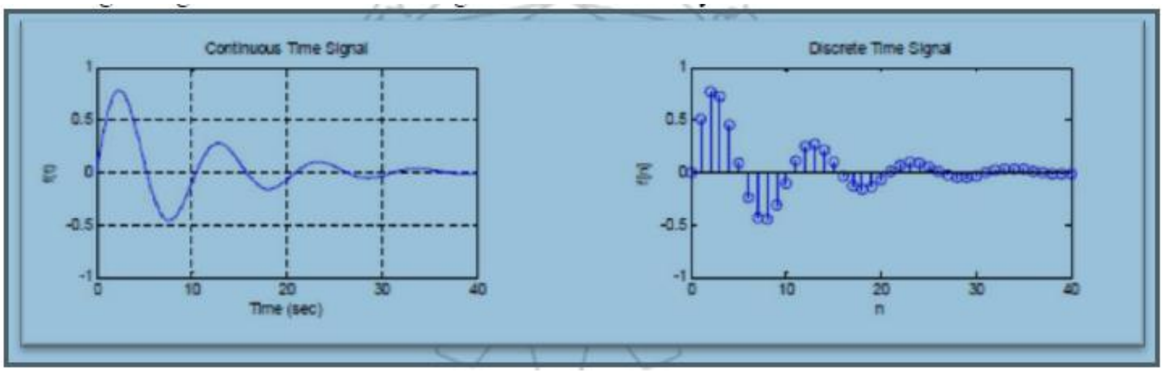
Laboratory Session No. 03 (a)

Object:
Plotting of Basic Signals in MATLAB & familiarization with procedural programming in MATLAB

Theory:

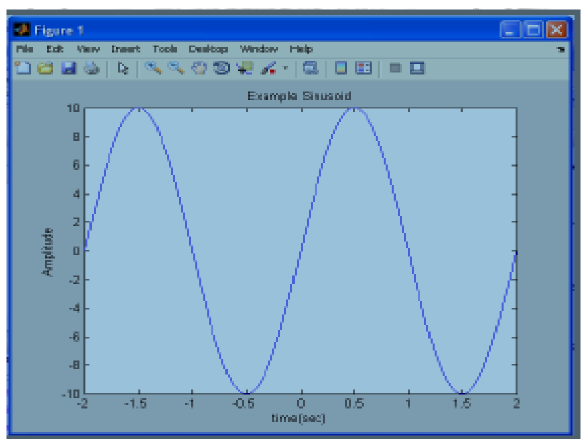
In the fields of communications, signal processing, and in electrical engineering more generally, a signal is any time-varying or spatial-varying quantity. This variable (quantity) changes in time.

- Speech or audio signal: Sound amplitude that varies in time
- Temperature readings at different hours of a day
- Stock price changes over days etc.



Signals can be classified by continues-time signal and discrete-time signal:

- A discrete signal or discrete-time signal is a time series, perhaps a signal that has been sampled from continuous time signal.
- A digital signal is a discrete-time signal that takes on only a discrete set of values.



While plotting in MATLAB one must be careful that a vector is plotted against a vector and lengths of vectors must match. Two functions are used for plotting:

- plot (for CT signals)
- stem (for DT signal)

Colours		Line Styles	
y	yellow	.	point
m	magenta	o	circle
c	cyan	x	x-mark
r	red	+	plus
g	green	-	solid
b	blue	*	star
w	white	:	dotted
k	black	-.	dashdot
		--	dashed

Plotting in MATLAB

Plot the function y=cos(x) between -π ≤x≤π

Code (write in m file):

```
clear all;
close all;
clc;
x=-pi:0.001:pi;
y=cos(x);
plot(x,y);
xlabel('Time')
ylabel('y=cos(x)');
legend('cos(x)');
title('Graph of cosine waveform');
grid on;
```

Exercise: Plot the function e-x/3sin(x) between 0≤x≤4π

Example:

x = 10sinπt

MATLAB Commands:

```
t = [-2:0.002:2]
x = 10 * sin (pi * t)
plot(t, x)
title('Example Sinusoid')
xlabel('time(sec)')
ylabel('Amplitude')
```

Multiple Plots:

For drawing multiple signals on the same graph, write first signal's x and y axis vectors followed by the next signal.

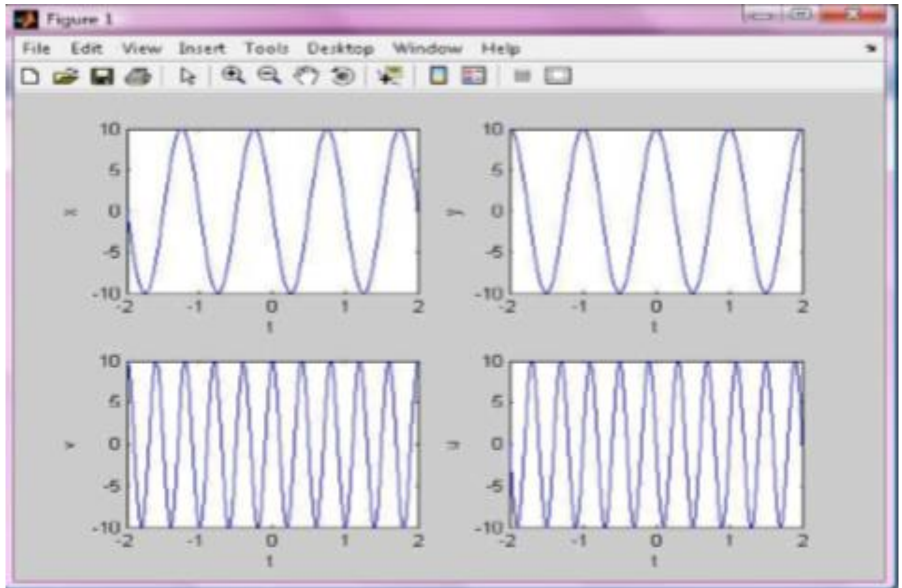
Syntax:

plot(X1,Y1,...,Xn,Yn)

In order to differentiate them by colors, write line style specifies and color code.

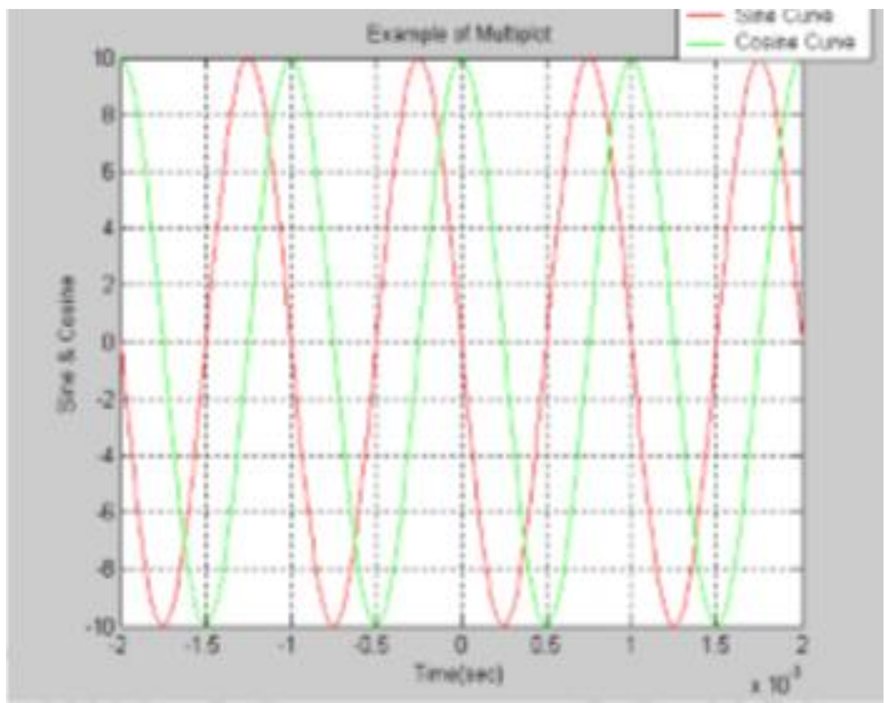
Example

plot (t, y, 'r-', t, x, 'g-');
legend ('Sine curve', 'Cosine curve')



Generating Subplots

x=10*sin(-2*pi*t)
y=10*cos(-2*pi*t)
u=10*sin(-5*pi*t)
v=10*cos(-5*pi*t)
t = [-2:0.002:2]
subplot(2, 2, 1), (t, x);
xlabel('t'),ylabel('x');
subplot(2, 2, 2), plot(t, y);
xlabel('t'),ylabel('y');
\subplot(2, 2, 4), plot(t, u);
xlabel('t'),ylabel('u');
subplot(2, 2, 3), plot(t, v);
xlabel('t'), ylabel('v');



DT Plots:

Example: Plot the DT sequences:

n = -6:3;
x = [2, 3, -1, 5, 4, 2, 3, 4, 6, 1];
stem(n, x);

Zero & One Vectors:

To generate zero or one vectors, use following statements:

zeros(1, 5)
Output: [0 0 0 0 0]
ones(1, 5)
Output: [1 1 1 1 1]

Using 'For loop' in MATLAB

Also called a do loop in other languages. Used when you want the calculations to be performed a defined number of times. In MATLAB, a for loop begins with the statement indicating how many times the statements in the loop will be executed
A counter is defined within this statement.

Example: for k = 1:100
(counter = k, the loop will be executed 100 times)
for i = 1:2:7
(counter = i, the counter will be incremented by a value of 2 each time until its value reaches 7. Therefore, the loop will be executed 4 times (i = 1,3,5, and 7)

The loop ends with an end statement
In M-files, the MATLAB editor will automatically indent text between the for and end statements.

Procedural Programming in MATLAB:

In MATLAB we can do procedural programming using the same constructs as in C/C++.
Looping Constructs in MATLAB:

a) while-loop

```
while(termination condition)
        statement1
        statement2
        statement3
end
```

b) for-loop

```
for var=startvalue: step : endvalue
        statement1
        statement2
        statement3
end
```

Using for Loops:

We will use for loops extensively in MATLAB, use of while loop will be seldom.

- In MATLAB, a for loop begins with the statement indicating how many times the statements in the loop will be executed
- A counter is defined within this statement
- Examples:

  for k = 1:100
  (counter = k, the loop will be executed 100 times)
  for i = 1:2:7

(counter = i, the counter will be incremented by a value of 2 each time until its value reaches 7. Therefore, the loop will be executed 4 times (i = 1,3,5, and 7)

- The loop ends with an end statement
- In M-files, the MATLAB editor will automatically indent text between the for and end statements:

```
for j = 1:10
    x(j) = 5*j;
end
```

Can you determine what the variable x will be after running this M-file?
Explanation of the loop run:

- The first time through the loop, j = 1
- Because of the single value in parentheses, x will be a one-dimensional array
- x(1) will be set equal to 5*1 = 5
- The second time through the loop, j = 2
- x(2) will be set equal to 5*2 = 10
- This will be repeated until j = 10 and x(10) = 50
- x will be a one-dimensional array (a row matrix) with 10 elements:

```
>> x
x =
     5    10    15    20    25    30    35    40    45    50
```

Loops in the Command Window:

- Loop commands can be entered directly from the command prompt
- The calculations are not performed until the end statement is entered

```
>> for j = 1:10
x(j) = j*5;
end
>> x
x =
     5    10    15    20    25    30    35    40    45    50
```

- Remember that if you leave off the semi-colon, the results of the calculations will be written to the screen in every loop:

```
>> clear x
>> for j = 1:10
x(j) = j*5
end
x =
      5
x =
      5     10
x =
      5     10     15
x =
      5     10     15     20
x =
      5     10     15     20     25
x =
      5     10     15     20     25     30
x =
      5     10     15     20     25     30     35
x =
      5     10     15     20     25     30     35     40
x =
      5     10     15     20     25     30     35     40     45
x =
      5     10     15     20     25     30     35     40     45     50
```

More exercises with for loop:

- What result will be output to the screen in each of the following examples?

```
y = 0;
for k = 1:5
  y = y + k;
end
y
```

```
y =
       15
```

- What is the value of y:

```
y = 0;
for k = 2:2:8
   y = y + k;
end
y
```

```
y =
       20
```

- What would be contained in the vector y?

```
for k = 1:5
   y(k)=k^2;
end
y
```

```
y =
       1       4       9      16      25
```

Nested Loops:

```
for j = 1:3
   for k = 1:3
      T(j,k) = j*k;
   end
end
T
```

```
T =
     1     2     3
     2     4     6
     3     6     9
```

Decision Making Constructs:

if(condition)
   statement1
   statement2
end
	if(condition)
	   statement1
	   statement2
	else
	   statement3
	   statement4
	end

Plotting with Loops:

- Consider this equation:

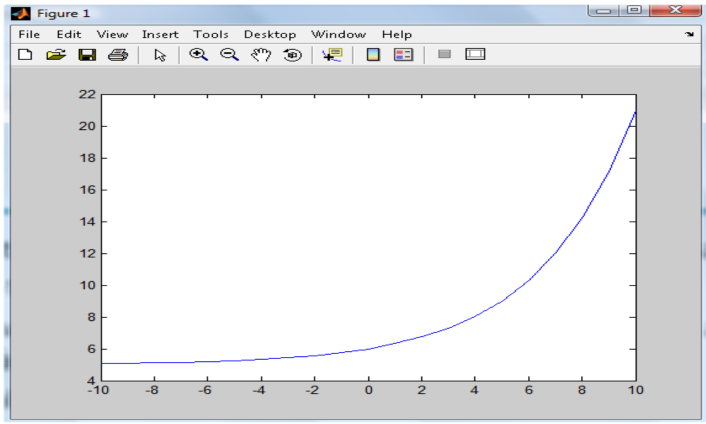
$$y = 2^{0.4x} + 5$$

- Plot this equation using 'for loop' for values of x from -10 to 10.
- Hint: Use a for loop to calculate and store x and y values in one-dimensional arrays

```
for i = 1:21
   x(i) = -10 +(i-1);
   y(i) = 2^(0.4*x(i)) + 5;end
```

- After running these lines of code, two one-dimensional arrays, x and y, have been created, each with 21 elements
- The stored arrays can be plotted with the command: plot(x,y)

- Any two one-dimensional arrays can be plotted, as long as they are exactly the same size
- The plot will be created in a new window



Now, add x label, y label and set x limit and y limit yourself.

POST LAB ACTIVITY:

Activity: Acquire 1000 points of cosine wave having frequency of 60Hz using for loop. Plot the Signal.

Observation

Conclusion

Laboratory Session No. 03 (b)

Objective:

Manipulating signals and developing systems on MATLAB software.

Theory

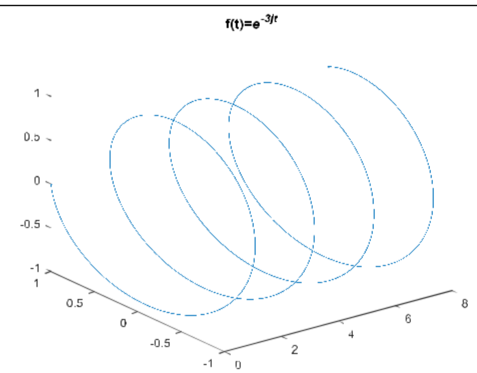
Numerical models of Exponential Signals

- Imaginary Power
- Real Power
- Complex Power

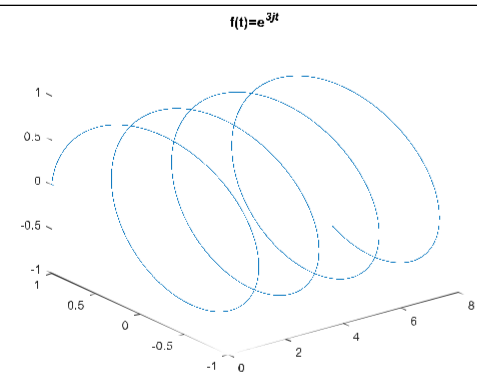
Numerical models of Exponential Signals with Imaginary Power

```
>> t = 0 : 0.001 : 0.8;
>> y1 = exp(-3*i*t); % Developing imaginary numerical model of simple decaying exponential signal
>> plot3(t,real(y1),imag(y1)),title('f(t)=\ite^{-3jt}'); % plots negative frequency signal
>> y2 = exp(+3*i*t); % Developing imaginary numerical model of simple rising exponential signal
>> plot3(t,real(y2),imag(y2)),title('f(t)=\ite^{3jt}'); % plots positive frequency signal
```

Code 01: Plotting imaginary numerical model using MATLAB



Imaginary Decay Exponential Plot



Imaginary Oscillatory Exponential Plot

The plot is in the form of Sine and Cosine wave curves. When we plot imaginary values, cycles are to be appeared, negative cycles classified them into decay of exponential plot as they are dealt in negative signs. Similarly, for the oscillatory exponential signal these cycles are positive cycles as they are dealt in positive sign. They just differ by signs which kept them different and classified into Decay and Oscillatory Exponential cycles when they are imaginary values. But these are not useful to us because they cannot be described in our physical world.

- ## Numerical models of Exponential Signals with Real Power

Run the following codes on MATLAB and attach the output and comment on it.

```
>> t = 0 : 0.001 : 0.8;
>> y3 = exp(-2*t); % Developing real numerical model of simple decaying exponential signal
>> plot(t,y3),xlabel('Time'),ylabel('f(t)'),title('f(t)=\ite^{-2t}'). grid on; % plots real decaying exponential signal
>> y4 = exp(2*t); % Developing real numerical model of simple oscillatory exponential signal
>> plot(t,y4),xlabel('Time'),ylabel('f(t)'),title('f(t)=\ite^{2t}'), grid on; % plots real rising exponential signal
```

OBSERVATION

```
>> t = 0 : 0.001 : 0.8;
>> y5 = exp(-20*t); % Developing numerical model of simple rising exponential signal
>> plot(t,y5),xlabel('Time'),ylabel('f(t)'),title('f(t)=\ite^{-20t}'), grid on; % plots rising exponential signal
```

OBSERVATION

- **Numerical models of Exponential Signals with Complex Power**

```
>> t = 0 : 0.001 : 0.8;
>> y6 = exp((-0.2+3*i)*t); % Developing complex numerical model of simple decaying exponential signal
>> plot3(t,real(y6),imag(y6)),title('f(t)=\ite^{-0.2+3jt}'); % plots decaying complex exponential signal
>> y7 = exp((0.2+3*i)*t); % Developing complex numerical model of simple oscillating exponential signal
>> plot3(t,real(y7),imag(y7)),title('f(t)=\ite^{0.2+3jt}'); % plots oscillating complex exponential signal
```

OBSERVATION

Run the following codes on MATLAB and attach the output and comment on it.

```
>> t2 = 0 : 0.001 : 0.8;
>> y8 = exp((0.2+3*i)*t2))+ exp((0.2-3*i)*t2);% Developing real signal using two complex exponential signal
>> plot3(t2,real(y6),imag(y6)),title('f(t)=\ite^{-0.2+3jt}'); % plots real signal using two complex exponential signal
>> plot(t2,y8),title('f(t)=\ite^{0.2+3jt}+\ite^{0.2-3jt}'); % plots oscillating complex exponential signal
```

Code 05: Plotting real signal using two complex exponential signal using MATLAB

OBSERVATION

Basic Operations and Manipulation of Signals

i) Multiplying fundamental signal with itself

```
>> F = 1;  % setting fundamental frequency
>> instants = 1000;
>> t = 0:1/(instants*F):1/F;
>> y1 = sin(2*pi*F*t);    % generates fundamental signal
>> plot(t,y1.*y1);    % plots the signal multiplied by itself
>> area(t,y1.*y1); xlim([0 1]);    %area under the curve of the plot
```

Code 06: Plotting the product of fundamental signal with itself using MATLAB

OBSERVATION

ii) Multiplying signal with second harmonically related signal

```
>> F = 1;  % setting fundamental frequency
>> instants = 1000;
>> t = 0:1/(instants*F):1/F;
>> y1 = sin(2*pi*F*t);    % generates fundamental signal
>> y2 = sin(2*pi*2*F*t);    % generates second harmonic signal
>> plot(t,y1.*y2); % plots the signal multiplied by second harmonic signal
>> area(t,y1.*y2); xlim([0 1]); %area under the curve of the plot
```

Code 07: Plotting the product of fundamental signal with second harmonic signal using MATLAB

OBSERVATION

iii) Multiplying cosine signal with sine signal of same frequency

```
>> F = 1;  % setting fundamental frequency
>> instants = 1000;
>> t = 0:1/(instants*F):1/F;
>> y3 = cos(2*pi*3*F*t);     % third harmonic cosine signal
>> y4 = sin(2*pi*3*F*t);     % third harmonic sine signal
>> plot(t,y3.*y4); % plots the signal multiplied by sine and cosine signal with same frequency
>> area(t,y4.*y5); xlim([0 1]); %area under the curve of the plot
```

Code 08: Plotting the product of sine signal and cosine signal with same frequency using MATLAB

OBSERVATION

iv) Multiplying cosine signal with sine signal of different frequency

```
>> F = 1;  % setting fundamental frequency
>> instants = 1000;
>> t = 0:1/(instants*F):1/F;
>> y1 = sin(2*pi*F*t);     % fundamental sine signal
>> y3 = cos(2*pi*3*F*t);     % third harmonic cosine signal
>> plot(t,y3.*y1); % plots the signal multiplied by sine and cosine signal with different frequency
>> area(t,y3*y1); xlim([0 1]); %area under the curve of the plot
```

Code 09: Plotting the product of sine signal and cosine signal with different frequency using MATLAB

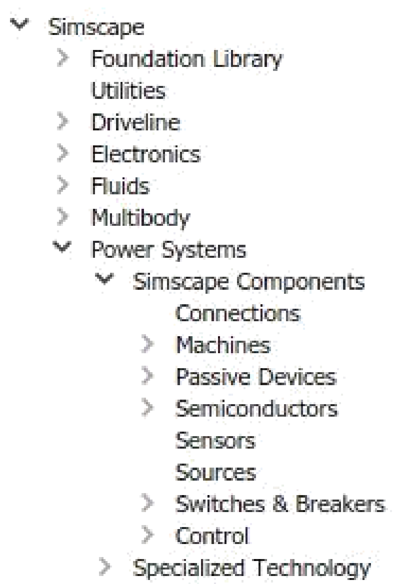
OBSERVATION

Basic circuit models on Simulink

Analyzing the RC response.

PROCEDURE

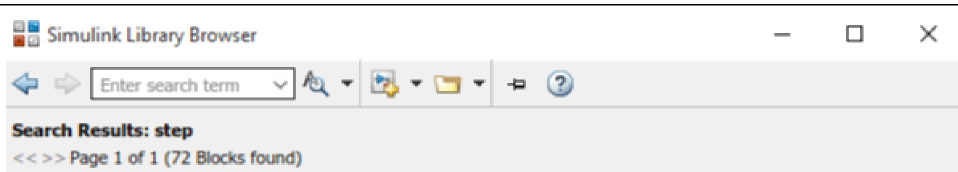
For basic circuit model on Simulink we use Simcscape and its sub-libraries, power systems library is most frequently used. All the required components can be find out in the libraries. The sample model is shown in Fig. 3.14.



Libraries

Power System Library

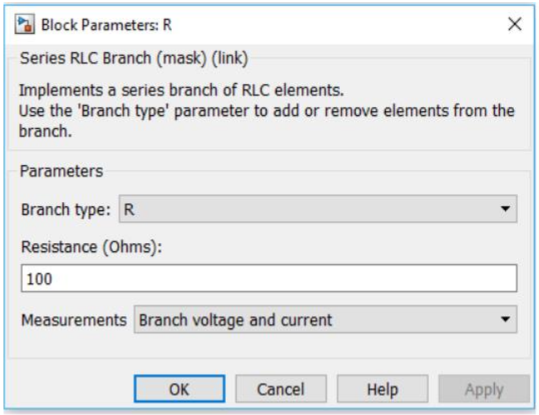
Searching the required components from the tool bar as shown in fig. 3.11. RLC branches, PowerGUI, Step, Controlled Voltage source, Multimeter, Scope. This search may reduce our work and search throughout all the sub-libraries.



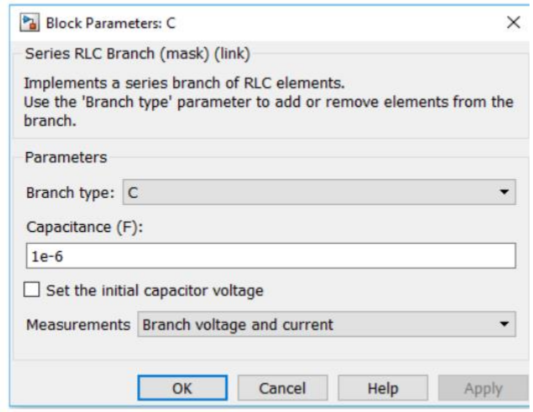
Component Searching

BLOCK PARAMETERS

The basic parameters of branches can be changed by simply double clicking on them as shown in figure below. It sets the value for RLC and also sets initial and final value of storing elements (C,L). Voltage and current response can also be measured by using multimeter.



Block parameters of resistor

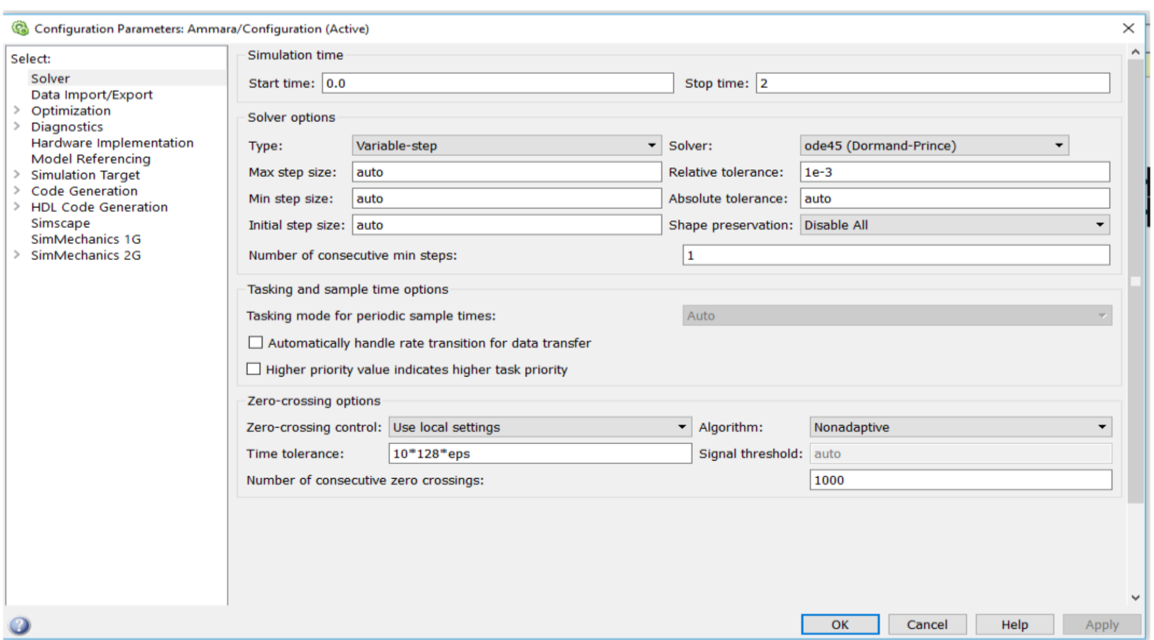


Block parameters of capacitor

SOLVER SELECTION

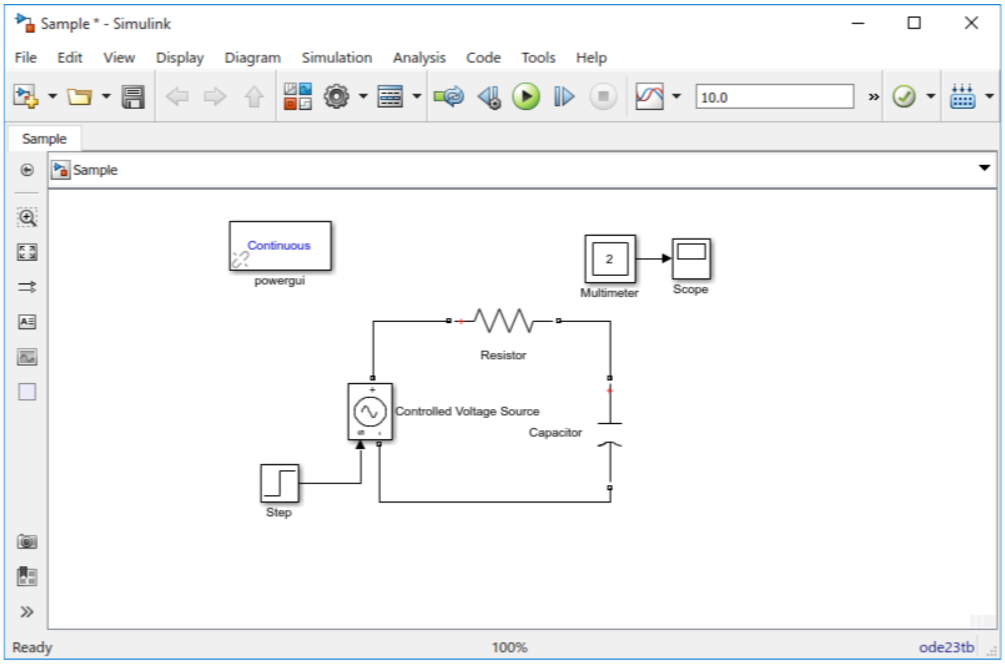
1. **<u>Ode 45 solver:</u>**
   It works on the numerical method called R-K method. It selects a step size at the beginning of a response and plots the whole response with the same step size regardless of the changing differences between the values.

2. **<u>Ode 23 solver:</u>**
   It is the most suitable solver for ODEs as it adjusts its step size according to the responses changing with respect to time. It is a dynamic solver.



Configuration Parameters Window

*[**Note**: We prefer "ode23tb" over "ode45tb" for fast simulation and more accurate result.]*



Sample Model

OBSERVATION

CONCLUSION

Laboratory Session No. 04

Objective:

Understand concepts of Fourier series hypothesis through MATLAB.

Theory:

Fourier Series Hypothesis

a) Statement

“Any periodic signal can be expressed as a linear combination of sinusoids or infinite sum of sines and cosines”

b) Analysis Equation

The Fourier series can be analyzed in trigonometric form, compact trigonometric form and exponential form. Where, “a0” is the zero harmonic (i.e. DC Offset at zeroth harmonic) and “n” is no. of harmonics.

i. Trigonometric Form

$$x(t) = a_0 + \sum_{n=1}^{\infty}(a_n \cos n\omega_o t \ + \ bn \sin n\omega_o t)$$

$a_0 = \frac{1}{T_o}\int_{T_o} x(t)\ dt$	$a_n = \frac{2}{T_o}\int_{T_o} x(t)\cos n\omega_o\,(t)dt$	$b_n = \frac{2}{T_o}\int_{T_o} x(t)\sin n\omega_o\,(t)dt$	$\omega_o = 2\pi f_o$

Table No. 4.1: Components of Fourier Series in Trigonometric form

Laboratory Session#04 Signals &Systems (EE-232)
Department of Electrical Engineering NED University of Engineering and Technology

ii. Compact Trigonometric Form

$$x(t) = C_o + \sum_{n=1}^{\infty} C_n \cos(n\omega_0 t + \theta_n)$$

$C_o = a_0 = \frac{1}{T_o}\int_{T_o} x(t)dt$	$C_n = \sqrt{{a_n}^2 + {b_n}^2}$	$\theta_n = tan^{-1}\left(\frac{-b_n}{a_n}\right)$	$\omega_o = 2\pi f_o$

Table No. 4.2: Components of Fourier Series in Compact Trigonometric form

iii. Exponential Form

$$x(t) = \sum_{n=1}^{\infty} D_n \, e^{jn\omega_o t}$$

$D_o = C_o = a_0 = \frac{1}{T_o}\int_{T_o} x(t)dt$	$D_n = \frac{1}{T_o}\int_{T_o} x(t)\, e^{-jn\omega_o t}\, dt$	$\omega_o = 2\pi f_o$

Table No. 4.3: Components of Fourier Series in Exponential form

Fourier Synthesis using MATLAB

(a) Find the trigonometric Fourier series for y(t) shown below.

MATLAB CODE

OBSERVATIONS

CONCLUSION

b. The signal y(t) can be obtained by time reversal of x(t) shown below. Use this fact to obtain the Fourier series for y(t) from the results in your textbook “Example 6.1”. Verify that the Fourier series thus obtained is identical to that found in part (a).

c. Show that, in general, time reversal of a periodic signal does not affect the amplitude spectrum, and the phase spectrum is also unchanged except for the change of sign.

NED University of Engineering & Technology
Department of Electrical Engineering

Course Code: **EE-232** Course Title: **Signals and Systems**
Laboratory Session No.: ________________________ Date: ______________________________

Psychomotor Domain Assessment Rubric for Laboratory (Level P3)					
Skill(s) to be assessed	Extent of Achievement				
	0	1	2	3	4
Software Menu Identification and Usage: Ability to initialise, configure and *operate* software environment *under supervision*, using menus, shortcuts, instructions etc.	Unable to understand and use software menu	Little ability and understanding of software menu operation, makes many mistake	Moderate ability and understanding of software menu operation, makes lesser mistakes	Reasonable understanding of software menu operation, makes no major mistakes	Demonstrates command over software menu usage with frequent use of advance menu options
Modeling of given Network or Signal Processing Scheme: Ability to *operate* software environment in order to create simulation model of given network parameters or signal processing scheme	Unable to operate software, could not create simulation model	Moderately able to operate software, could not create simulation model	Adequately able to operate software, simulation model contains errors	Adequately able to operate software, simulation model is error free	Demonstrates mastery over software, error free simulation model is created and simulation is started successfully
Procedural Programming of given Signal Processing Scheme: *Practice* procedural programming techniques, in order to code specific signal processing schemes	Little to no understanding of procedural programming techniques	Slight ability to use procedural programming techniques for coding given algorithm	Mostly correct recognition and application of procedural programming techniques but makes crucial errors for the given processing scheme	Correctly recognises and uses procedural programming techniques with no errors but unable to run processing scheme successfully	Correctly recognises and uses procedural programming techniques with no errors and runs processing successfully
Detecting and Removing Errors: *Detect* Errors/Exceptions and in simulation model and *manipulate* model to rectify the simulation	Unable to check and detect error messages and indications in software	Able to find error messages and indications in software but no understanding of detecting those errors and their types	Able to find error messages and indications in software as well as understanding of detecting some of those errors and their types	Able to find error messages in software as well as understanding of detecting all of those errors and their types	Able to find error messages in software along with the understanding to detect and rectify them

Psychomotor Domain Assessment Rubric for Laboratory (Level P3)					
Skill(s) to be assessed	Extent of Achievement				
	0	1	2	3	4
Graphical Visualisation and Comparison of Model/Signal Processing Scheme Parameters: <u>*Manipulate*</u> given model/simulation under supervision, in order to produce graphs/plots for measuring and comparing model / signal processing parameters	Unable to understand and utilise visualisation or plotting features	Ability to understand and utilise visualisation and plotting features with frequent errors	Ability to understand and utilise visualisation and plotting features successfully but unable to compare and analyse them	Ability to understand and utilise visualisation and plotting features successfully, partially able to compare and analyse them	Ability to understand and utilise visualisation and plotting features successfully, also able to compare and analyse them
Following step-by-step procedure to complete lab work: <u>*Observe, imitate and operate*</u> software to complete the provided sequence of steps	Inability to recognise and perform given lab procedures	Able to recognise given lab procedures and perform them but could not follow the prescribed order of steps	Able to recognise given lab procedures and perform them by following prescribed order of steps, with frequent mistakes	Able to recognise given lab procedures and perform them by following prescribed order of steps, with occasional mistakes	Able to recognise given lab procedures and perform them by following prescribed order of steps, with no mistakes
Recording Simulation Observations: <u>*Observe and copy*</u> prescribed or required simulation results in accordance with lab manual instructions	Inability to recognise prescribed or required simulation measurements	Able to recognise prescribed or required simulation measurements but does not record according to given instructions	—	Able to recognise prescribed or required simulation measurements but records them incompletely	Able to recognise prescribed or required simulation measurements and records them completely, in tabular form
Discussion and Conclusion: <u>*Demonstrate*</u> discussion capacity on the recorded observations and draw conclusions from it, relating them to theoretical principles/concepts	Complete inability to discuss recorded observations and draw conclusions	Slight ability to discuss recorded observations and draw conclusions	Moderate ability to discuss recorded observations and draw conclusions	Reasonable ability to discuss recorded observations and draw conclusions	Full ability to discuss recorded observations and draw conclusions

Laboratory Session No. 05

Objective:
To study different types of signal using MATLAB

THEORY

- **Singularity Functions:**

Singularity functions are discontinuous functions, or their derivatives are discontinuous. A singularity is a point at which a function does not possess a derivative. In other words, a singularity function is discontinuous at its singular points.
Singularity functions are a class of discontinuous functions that contain singularities, i.e. they contain points in which their derivatives do not exist. In other words, a singularity function is discontinuous at its singular points.

- **Elementary Signals:**

The elementary signals are used for analysis of systems. Such signals are,

1. Step
2. Impulse
3. Ramp
4. Exponential
5. Sinusoidal

- **Some Useful Signals:**

1. Dirac Delta Function:

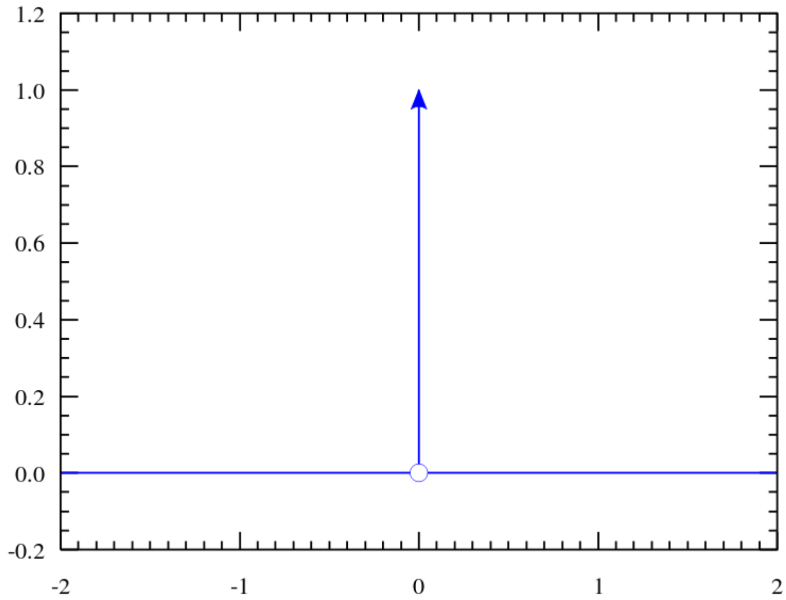
The Dirac delta can be loosely thought of as a function on the real line which is zero everywhere except at the origin, where it is infinite,

$$\delta(x) = \begin{cases} +\infty, & x = 0 \\ 0, & x \neq 0 \end{cases}$$

and which is also constrained to satisfy the identity

$$\int_{-\infty}^{\infty} \delta(x)\, dx = 1.$$

This is also called unit impulse function.

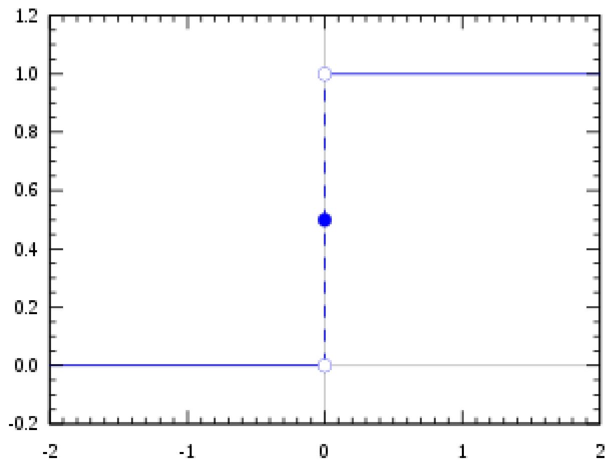


2. Unit Step Sequence:

The unit step function is a discontinuous function, whose value is zero for negative arguments and one for positive arguments.

$$u(t) = \begin{cases} 1 & t \geq 0 \\ 0 & t < 0 \end{cases}$$

It can also be defined as the integral of the Dirac delta function.

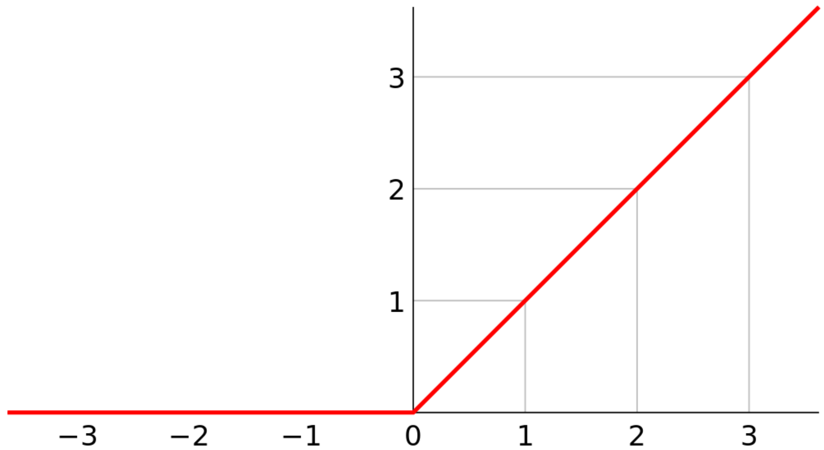


3. Ramp Sequence:

The ramp function is a unary real function which is 0 for negative inputs; output equals input for non-negative inputs.

$$R(x) := \begin{cases} x, & x \geq 0; \\ 0, & x < 0 \end{cases}$$

It can also be defined as the integral of the unit step function.



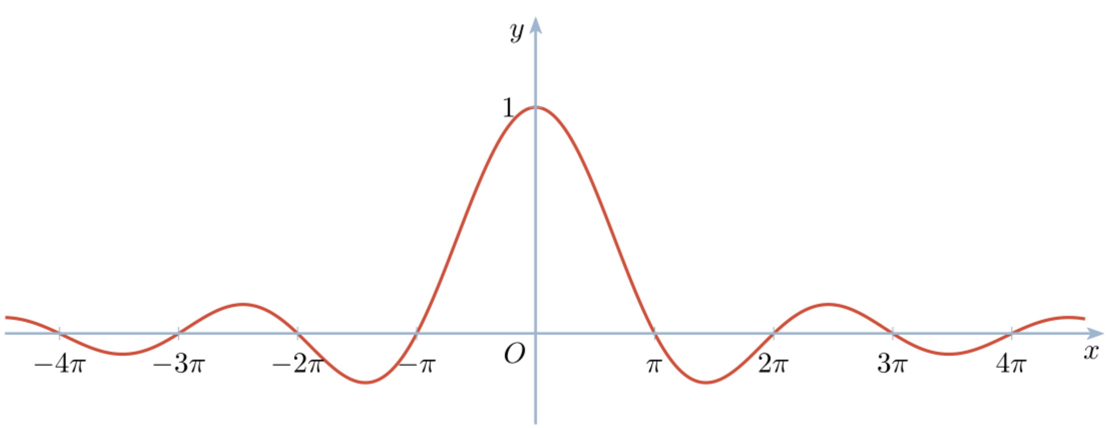
4. **Sinc Function**:

The sinc function $\operatorname{sinc}(x)$, also called the "sampling function," is a function that arises frequently in signal processing and the theory of Fourier transforms. The full name of the function is "sine cardinal," but it is commonly referred to by its abbreviation, "sinc."

$$\operatorname{sinc}(x) \equiv \begin{cases} 1 & \text{for } x = 0 \\ \dfrac{\sin x}{x} & \text{otherwise,} \end{cases}$$

This has the normalization

$$\int_{-\infty}^{\infty} \operatorname{sinc}(x)\, dx = \pi.$$

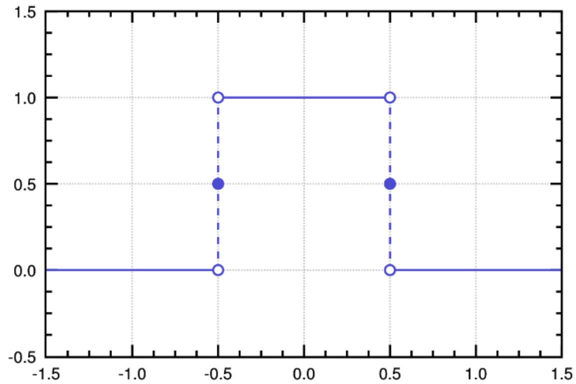


5. Rect Function:

The Rect Function is a function which produces a rectangular-shaped pulse with a width of 1 centered at t = 0. The Rect function pulse also has a height of 1. A rect function cn be written in the form :

$$\text{Rect}\ (\frac{t-X}{Y})$$

where the pulse is centered at X and has width Y.



6. Signum Function:

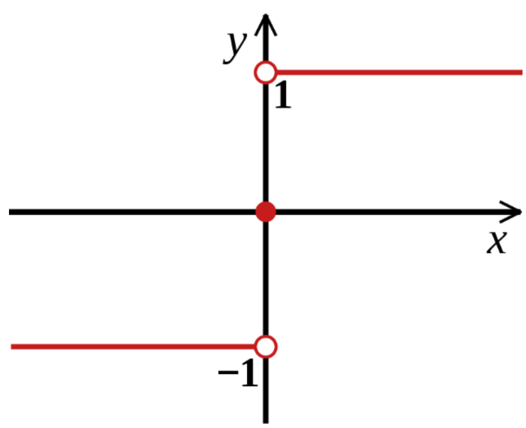
The sign function or signum function is an odd mathematical function that extracts the sign of a real number.

The signum function of a real number x is defined as follows:

$$\text{sgr}\,(x) := \begin{cases} -1 & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ 1 & \text{if } x > 0. \end{cases}$$

Alternatively,

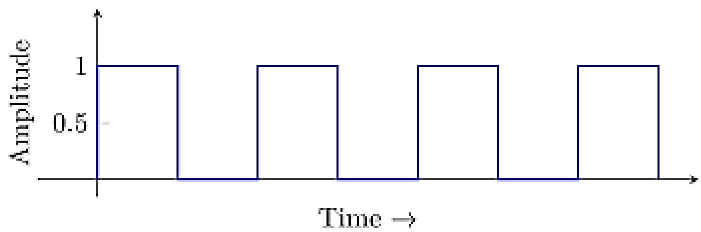
$$\text{sgn}(x) = \frac{\text{d}}{\text{d}x}\,|x|\,, \quad x \neq 0$$



7. **Square Wave:**

A square wave is a non-sinusoidal periodic waveform in which the amplitude alternates at a steady frequency between fixed minimum and maximum values, with the same duration at minimum and maximum.
It can be defined as simply the sign function of a sinusoid:

$$x(t) = \operatorname{sgn}\left(\sin\frac{2\pi t}{T}\right) = \operatorname{sgn}(\sin 2\pi f t)$$
$$v(t) = \operatorname{sgn}\left(\cos\frac{2\pi t}{T}\right) = \operatorname{sgn}(\cos 2\pi f t).$$



8. **Sawtooth wave:**

The sawtooth wave, called the "castle rim function" is the periodic function given by

$$S(x) = A \operatorname{frac}\left(\frac{x}{T} + \phi\right),$$

where $\operatorname{frac}(x)$ is the fractional part $\operatorname{frac}(x) \equiv x - \lfloor x \rfloor$, A is the amplitude, T is the period of the wave, and ϕ is its phase. It therefore consists of an infinite sequence of truncated functions concatenated together.

POST LAB ACTIVITY:

Activity: Plot various functions as outlined above using Matlab, both as continuous-time signals, and discrete-time signals

Laboratory Session No. 06

Objective:

Using MATLAB Simulink, explore the response of a simple RC circuit.

Theory:

Time Delay Circuits
A circuit in which the output signal is delayed by a specified time interval with respect to the input signal is known as Time delay circuit or simply a delay circuit. All the Electrical systems experiences some sort of "time delay" between its input and output when the voltage or signal is applied to it.

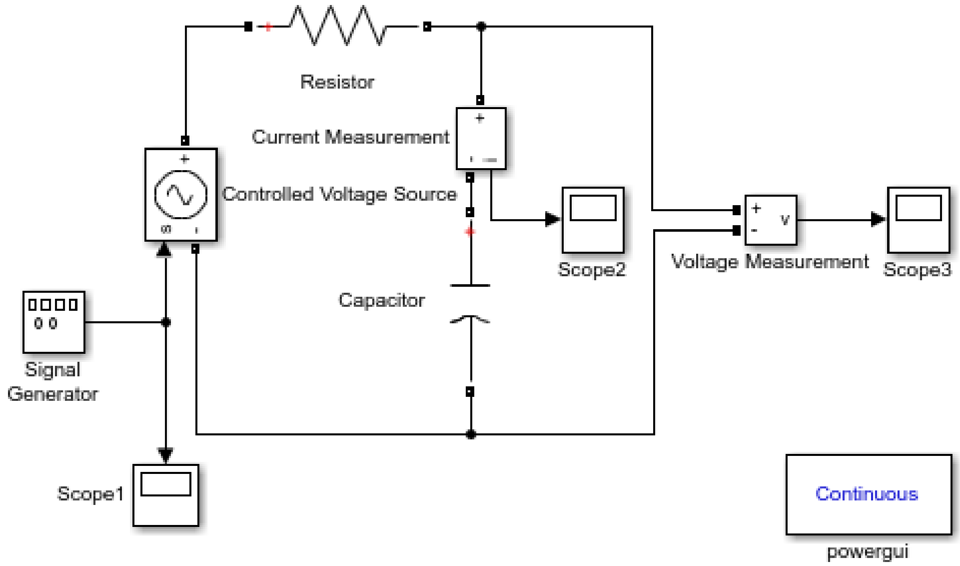
Time Constant (τ)
The delay in the time delay circuits is generally known as the Time Constant of the circuit and is defined as;

"The time required for the response to decay to a factor of 1/e or ≈36.8% of its initial value, conversely, the time required to charge the capacitor from an initial voltage of zero to ≈63.2% of the value of an applied source voltage"

The time constant of a circuit mainly depends upon the reactive components either capacitive or inductive connected to it and is a measurement of the response time with the unit of Tau (τ).

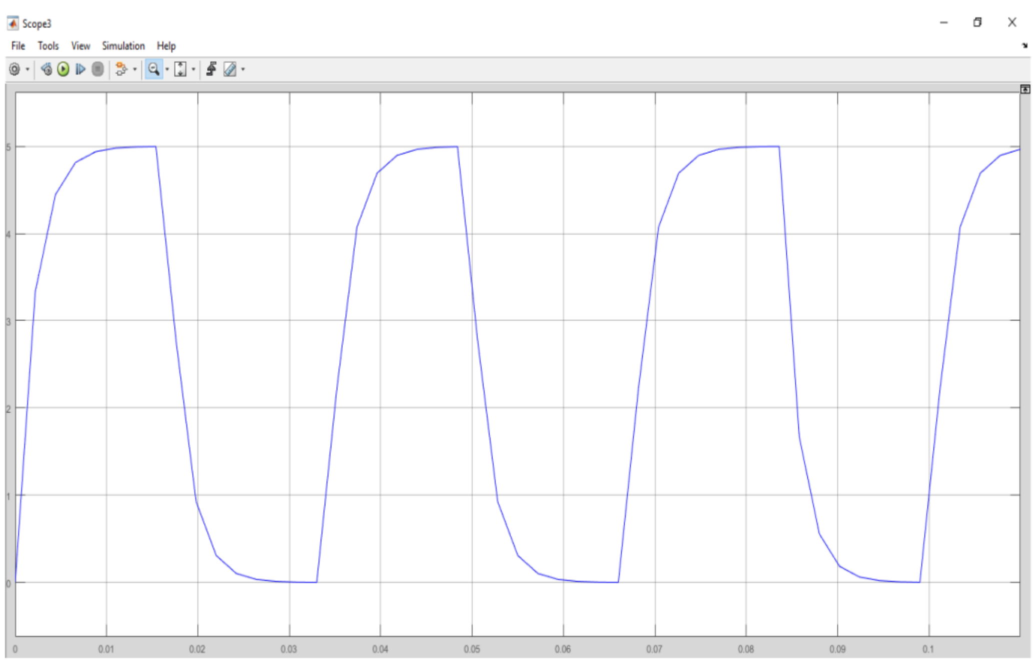
RC as a Time Delay Circuit
Consider a simple series RC circuit having;

- R=1.8k Ω
- C=1μF
- VS = 5Vpp



Series RC circuit

The Fig. 5.1 shows a resistor (R) in series with a capacitor (C) forming a series RC Circuit connected to a function generator that gives a square wave of peak to peak voltage. The output of this simulink model is shown in Fig.5.2 (as shown below).



Series RC circuit response [v(t) vs t]

RC time constant:

The capacitor will charge up gradually through the resistor until the voltage across the capacitor reaches that of the supplied voltage. The transient response required for the capacitor to fully charge is equivalent to about 5-time constants or 5τ.

$$\tau = RC = (1.8x1000)(1x10-6)$$
$$\boldsymbol{\tau = 1.8ms} \text{ (Transient response)}$$

In our time delay circuit, we would like to observe the complete charging and discharging of the capacitor this is why we selected the elemental values carefully for a time constant of 1.8ms. As the time required by a capacitor to fully charge or discharge is approximately equal to 5τ, we will setup the function generator to produce a square wave of 5VPP with ON time and OFF time both of 5τ = 9ms or greater than that, setting frequency f = 30 Hz.

$$\because T = 1/f = 1/30 = 33ms$$

Since, the time period of square wave is 33ms > 18 ms (i.e. 10 τ) which implies that the RC response can be easily observed for this input. However, the circuit impedance should be high for an input of low frequency otherwise the input waveform will be deformed. For the circuit in Fig. 5.1 the total impedance is;

$$Z = R + X = 7.105k\Omega$$

ANALYSIS OF RC RESPONSE

The total RC response is given by:

Total response = natural response + forced response

a) Natural Response

In RC circuit, natural response is the system's response to initial conditions with all external independent voltage sources set to zero volts (short circuit).

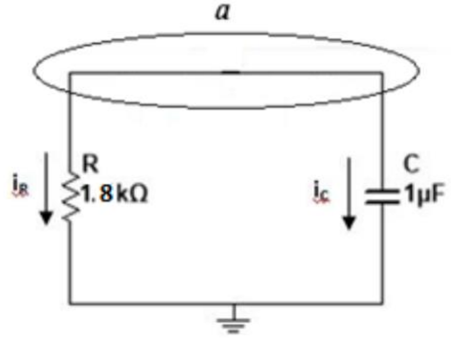


Fig. 5.3. Series RC circuit for zero input condition

Apply KCL at node “a” in Fig.5.3;

$$i_c + i_r = 0 \quad => \quad C\frac{dV}{dt} + \frac{V}{R} = 0 \quad => \quad \frac{dV}{dt} + \frac{V}{RC} = 0 \quad => \quad \frac{dV}{V} = -\frac{1}{RC}\,dt$$

Integrating;

$$\ln V = -\frac{t}{RC} + \ln C \quad => \quad \ln V - \ln C = -\frac{t}{RC}$$

Taking exponentials both sides,

$$\frac{V}{C} = e - t/RC \quad => \quad V(t) = V_o\, e^{-t/\tau}$$

Where, Vo is the initial voltage stored in capacitor at the instant of discharging i.e. t=0. This shows that the natural response of an RC circuit is an exponential decay of its initial voltage.

For the desired circuit, V_0 = 5 and τ = 1.8ms.

$$\boxed{V(t) = 5e^{-t/0.0018}}$$

b) Forced Response

The forced response is the system's response with source turned on but with the initial conditions set to zero. The forced response of a circuit is its behavior for a long time after the excitation is applied.

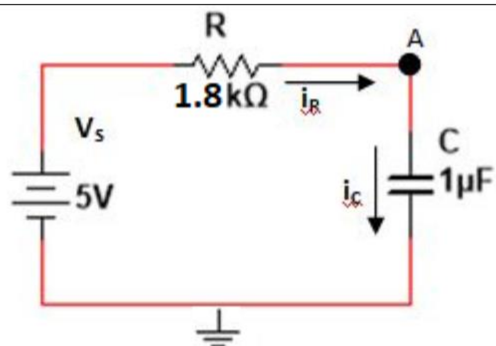


Fig. 5.4. Series RC circuit when voltage source is connected

Applying KCL at node A in Fig. 5.4;

$$C\frac{dV}{dt} + \frac{V-Vs}{R} = 0 \quad => \quad \frac{dV}{dt} + \frac{V}{RC} = \frac{Vs}{RC} \quad => \quad \frac{dV}{dt} = -\frac{V-Vs}{RC} \quad => \quad \frac{dV}{V-Vs} = -\frac{dt}{RC}$$

Integrating;

$$\ln(V - V_s) = -\frac{t}{RC} + \ln C \quad => \quad \ln(V - V_s) - \ln C = -\frac{t}{RC} \quad => \quad \ln\left(\frac{V-Vs}{C}\right) = -\frac{t}{RC}$$

$$\boxed{V(t) = V_s + Ce^{-t/RC}}$$

This is the response of the RC circuit to abrupt change of source when the capacitor was initially charged to some voltage Vo at t – instant. Since the voltage of a capacitor cannot change abruptly, therefore

$$v(t^-) = v(t^+) = V_0 \quad \text{and} \quad C = V_o - V_S$$

$$\mathbf{V(t) = V_S + (V_0 - V_s)\, e^{-t/\tau}}$$

When the initial conditions are zero then,

$$v(t^-) = v(t^+) = 0 \quad \text{and} \quad C = -V_S;$$

$$V(t) = V_S\,(1 - e^{-t/\tau})$$

For the desired circuit VS = 5V and τ = 1.8 ms;

$$\mathbf{V(t) = 5\,(1 - e^{-t/0.0018})}$$

<u>POST LAB ACTIVITY:</u>

Plot the natural and forced response of the circuit using MATLAB and Validate its total response.

Laboratory Session No. 7

Objective:

Time Domain Analysis of RC Circuit – To simulate, understand, and validate Zero-Input Response of RC Series Circuit

Theory:

TIME CONSTANT

In an RC circuit, the product of Resistance and capacitance of the circuit is called RC time constant (τ). It has a good contribution in the time delay of RC circuits. Mathematically; Where R is the resistance of the circuit and C is the capacitance of the circuit.

Significance of Time Constant

The value of time constant is measured in seconds. It is the value of time that shows how much a capacitor will be charged and discharged. One must note that five times the time constant (tau) refers to the state when a capacitor is either fully charged or fully discharged depending on whether an RC circuit is charging or discharging.

RESPONSE OFSYSTEM:

System's output is the result of two causes that are independent:

I. Initial conditions at $t = 0$
II. Input of system at $t \geq 0$

The total response of system is sum of two components.

TYPES OF RESPONSE*:*

Zero input response:

That results from initial conditions at $t = 0$, with input $x(t) = 0$, for $t \geq 0$.

Zero-state response:
Response that results from input x(t) for t ≥ 0, with initial conditions (at t = 0) are zero.

TOTALRESPONSE

TOTAL RESPONSE = Zero-input response + Zero-state response

To calculate zero input response of RC Series circuit,

$$y_0(t) = y(0).\, e^{-t/RC}$$

Activity 1: Simulate RC Circuit using simulink, given,

C = 220μF , R = 10000 Ohm, Initial Conditions: $V_c(0) = y(0) = 5V$

Activity 2: Calculate Time Constant and draw the graph of Vc

Activity 3: Observe Zero-Input response using Simulink's scope element and observe values as mentioned in each column below:

Time (t)	Observed Value y(t)	Observed %age y(0)	Calculated Value y(t)	Calculated % of y(0)	Remarks
t = 1.T					
t = 2.T					
t = 3.T					
t = 4.T					
t = 5.T					

POST LAB ACTIVITY:

Activity 1: Simulate RL Circuit using Simulink. Assume the values of each element

Activity 2: Calculate Time Constant and draw the graph of current through inductor

Activity 3: Observe Zero-Input response using Simulink's scope element and observe values of parameters mentioned in the table below:

Time (t)	Observed Value y(t)	Observed %age y(0)	Calculated Value y(t)	Calculated % of y(0)	Remarks
t = 1.T					
t = 2.T					
t = 3.T					
t = 4.T					
t = 5.T					

NED University of Engineering & Technology
Department of Electrical Engineering

Course Code: **EE-232** Course Title: **Signals and Systems**
Laboratory Session No.: ________________________ Date: ________________________________

Psychomotor Domain Assessment Rubric for Laboratory (Level P3)					
Skill(s) to be assessed	Extent of Achievement				
	0	1	2	3	4
Software Menu Identification and Usage: Ability to initialise, configure and *operate* software environment *under supervision*, using menus, shortcuts, instructions etc.	Unable to understand and use software menu	Little ability and understanding of software menu operation, makes many mistake	Moderate ability and understanding of software menu operation, makes lesser mistakes	Reasonable understanding of software menu operation, makes no major mistakes	Demonstrates command over software menu usage with frequent use of advance menu options
Modeling of given Network or Signal Processing Scheme: Ability to *operate* software environment in order to create simulation model of given network parameters or signal processing scheme	Unable to operate software, could not create simulation model	Moderately able to operate software, could not create simulation model	Adequately able to operate software, simulation model contains errors	Adequately able to operate software, simulation model is error free	Demonstrates mastery over software, error free simulation model is created and simulation is started successfully
Procedural Programming of given Signal Processing Scheme: *Practice* procedural programming techniques, in order to code specific signal processing schemes	Little to no understanding of procedural programming techniques	Slight ability to use procedural programming techniques for coding given algorithm	Mostly correct recognition and application of procedural programming techniques but makes crucial errors for the given processing scheme	Correctly recognises and uses procedural programming techniques with no errors but unable to run processing scheme successfully	Correctly recognises and uses procedural programming techniques with no errors and runs processing successfully
Detecting and Removing Errors: *Detect* Errors/Exceptions and in simulation model and *manipulate* model to rectify the simulation	Unable to check and detect error messages and indications in software	Able to find error messages and indications in software but no understanding of detecting those errors and their types	Able to find error messages and indications in software as well as understanding of detecting some of those errors and their types	Able to find error messages in software as well as understanding of detecting all of those errors and their types	Able to find error messages in software along with the understanding to detect and rectify them

Psychomotor Domain Assessment Rubric for Laboratory (Level P3)					
Skill(s) to be assessed	Extent of Achievement				
	0	1	2	3	4
Graphical Visualisation and Comparison of Model/Signal Processing Scheme Parameters: *Manipulate* given model/simulation under supervision, in order to produce graphs/plots for measuring and comparing model / signal processing parameters	Unable to understand and utilise visualisation or plotting features	Ability to understand and utilise visualisation and plotting features with frequent errors	Ability to understand and utilise visualisation and plotting features successfully but unable to compare and analyse them	Ability to understand and utilise visualisation and plotting features successfully, partially able to compare and analyse them	Ability to understand and utilise visualisation and plotting features successfully, also able to compare and analyse them
Following step-by-step procedure to complete lab work: *Observe, imitate and operate* software to complete the provided sequence of steps	Inability to recognise and perform given lab procedures	Able to recognise given lab procedures and perform them but could not follow the prescribed order of steps	Able to recognise given lab procedures and perform them by following prescribed order of steps, with frequent mistakes	Able to recognise given lab procedures and perform them by following prescribed order of steps, with occasional mistakes	Able to recognise given lab procedures and perform them by following prescribed order of steps, with no mistakes
Recording Simulation Observations: *Observe and copy* prescribed or required simulation results in accordance with lab manual instructions	Inability to recognise prescribed or required simulation measurements	Able to recognise prescribed or required simulation measurements but does not record according to given instructions	—	Able to recognise prescribed or required simulation measurements but records them incompletely	Able to recognise prescribed or required simulation measurements and records them completely, in tabular form
Discussion and Conclusion: *Demonstrate* discussion capacity on the recorded observations and draw conclusions from it, relating them to theoretical principles/concepts	Complete inability to discuss recorded observations and draw conclusions	Slight ability to discuss recorded observations and draw conclusions	Moderate ability to discuss recorded observations and draw conclusions	Reasonable ability to discuss recorded observations and draw conclusions	Full ability to discuss recorded observations and draw conclusions

Laboratory Session No. 08

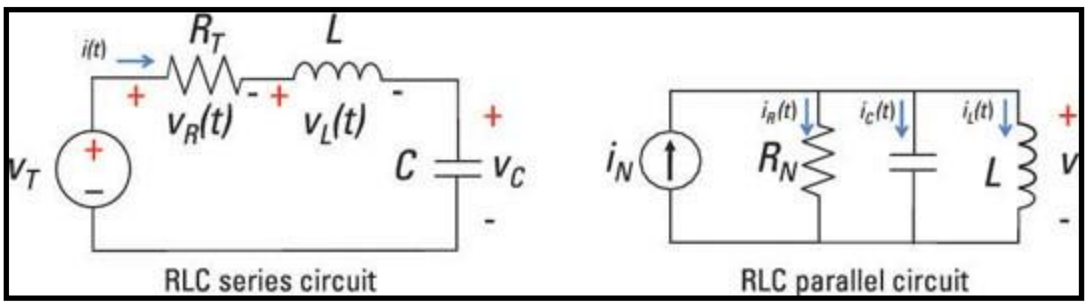
Objective:

Time Domain Analysis of SECOND ORDER Circuit – To simulate, understand, and validate Zero-Input Response of RLC Series Circuit

Theory:

SECOND ORDER CIRCUITS

Circuits that include an inductor, capacitor, and resistor connected in series or in parallel are second-order circuits.



If you can use a second-order differential equation to describe the circuit you're looking at, then you're dealing with a second-order circuit.

TIME CONSTANT

In an RC circuit, the product of Resistance and capacitance of the circuit is called RC time constant or (τ). It has a good contribution in the time delay of RC circuits.
Mathematically; for RC Circuit,

$$\tau = RC$$

For RL circuit,

$$\tau = \frac{R}{L}$$

Where R is the resistance of the circuit, C is the capacitance, and L is inductance of the circuit.

Activity:

Construct a series RLC circuit using Simulink with R = 3 Ohms, L=1 H, C=1.5 F. Visualize the response of circuit as voltage across capacitor and current through the circuit.

POST LAB ACTIVITIES

Activity 1:

Simulate zero input response of RLC Series circuit, given V_C (0) = y (0) = 5V and visualize the response as current through inductor

C = 0.22F, R = 1-Ohm, L = 1H, Initial Conditions: V_C (0) = y (0)=5V

Activity 2:

Simulate zero input response of RLC Series circuit, given V_C (0) = y (0) = 5V and visualize the response as current through inductor

C = 4F, R = 1-Ohm, L = 1H, Initial Conditions: V_C (0) = y (0)=5V

NED University of Engineering & Technology
Department of Electrical Engineering

Course Code: **EE-232** Course Title: **Signals and Systems**
Laboratory Session No.: ________________________ Date: ________________________________

Psychomotor Domain Assessment Rubric for Laboratory (Level P3)					
Skill(s) to be assessed	Extent of Achievement				
	0	1	2	3	4
Software Menu Identification and Usage: Ability to initialise, configure and *operate* software environment *under supervision*, using menus, shortcuts, instructions etc.	Unable to understand and use software menu	Little ability and understanding of software menu operation, makes many mistake	Moderate ability and understanding of software menu operation, makes lesser mistakes	Reasonable understanding of software menu operation, makes no major mistakes	Demonstrates command over software menu usage with frequent use of advance menu options
Modeling of given Network or Signal Processing Scheme: Ability to *operate* software environment in order to create simulation model of given network parameters or signal processing scheme	Unable to operate software, could not create simulation model	Moderately able to operate software, could not create simulation model	Adequately able to operate software, simulation model contains errors	Adequately able to operate software, simulation model is error free	Demonstrates mastery over software, error free simulation model is created and simulation is started successfully
Procedural Programming of given Signal Processing Scheme: *Practice* procedural programming techniques, in order to code specific signal processing schemes	Little to no understanding of procedural programming techniques	Slight ability to use procedural programming techniques for coding given algorithm	Mostly correct recognition and application of procedural programming techniques but makes crucial errors for the given processing scheme	Correctly recognises and uses procedural programming techniques with no errors but unable to run processing scheme successfully	Correctly recognises and uses procedural programming techniques with no errors and runs processing successfully
Detecting and Removing Errors: *Detect* Errors/Exceptions and in simulation model and *manipulate* model to rectify the simulation	Unable to check and detect error messages and indications in software	Able to find error messages and indications in software but no understanding of detecting those errors and their types	Able to find error messages and indications in software as well as understanding of detecting some of those errors and their types	Able to find error messages in software as well as understanding of detecting all of those errors and their types	Able to find error messages in software along with the understanding to detect and rectify them

Psychomotor Domain Assessment Rubric for Laboratory (Level P3)					
Skill(s) to be assessed	Extent of Achievement				
	0	1	2	3	4
Graphical Visualisation and Comparison of Model/Signal Processing Scheme Parameters: <u>*Manipulate*</u> given model/simulation under supervision, in order to produce graphs/plots for measuring and comparing model / signal processing parameters	Unable to understand and utilise visualisation or plotting features	Ability to understand and utilise visualisation and plotting features with frequent errors	Ability to understand and utilise visualisation and plotting features successfully but unable to compare and analyse them	Ability to understand and utilise visualisation and plotting features successfully, partially able to compare and analyse them	Ability to understand and utilise visualisation and plotting features successfully, also able to compare and analyse them
Following step-by-step procedure to complete lab work: <u>*Observe, imitate and operate*</u> software to complete the provided sequence of steps	Inability to recognise and perform given lab procedures	Able to recognise given lab procedures and perform them but could not follow the prescribed order of steps	Able to recognise given lab procedures and perform them by following prescribed order of steps, with frequent mistakes	Able to recognise given lab procedures and perform them by following prescribed order of steps, with occasional mistakes	Able to recognise given lab procedures and perform them by following prescribed order of steps, with no mistakes
Recording Simulation Observations: <u>*Observe and copy*</u> prescribed or required simulation results in accordance with lab manual instructions	Inability to recognise prescribed or required simulation measurements	Able to recognise prescribed or required simulation measurements but does not record according to given instructions	—	Able to recognise prescribed or required simulation measurements but records them incompletely	Able to recognise prescribed or required simulation measurements and records them completely, in tabular form
Discussion and Conclusion: <u>*Demonstrate*</u> discussion capacity on the recorded observations and draw conclusions from it, relating them to theoretical principles/concepts	Complete inability to discuss recorded observations and draw conclusions	Slight ability to discuss recorded observations and draw conclusions	Moderate ability to discuss recorded observations and draw conclusions	Reasonable ability to discuss recorded observations and draw conclusions	Full ability to discuss recorded observations and draw conclusions

Laboratory Session No. 9

Objective:

To perform and understand Convolutional Integral (C.I) of two signals using Matlab and compare it with C.I calculated analytically

Theory:

CONVOLUTION

Convolution between two continuous time signals $x(t)$ and $h(t)$ is defined as:

$$x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau$$

Where, τ is a dummy variable for integration.

Activity 1: Perform Continuous Time Convolution using Matlab and draw graphs of signals and convolutional integral using Matlab for following signal,

$$x(t) = 0.5, \quad 2 < t < 4$$
$$x(t) = 0, \quad elsewhere$$

And

$$h(t) = e^{-3t}u(t)$$

Matlab Script:

```
t0=0;
tf=12;
N=5000;
dt=(tf-t0)/N;
t=t0:dt:tf;
for k=1:length(t)
if t(k) < 2
x(k)=0;
else if t(k) < 4
```

```
x(k)= 0.05;
else
x(k)=0;
end
end
end
h=exp(-3*t);
y=conv(x,h);
subplot(3,1,1), plot(t,x,'r')
axis([t0,tf,0,0.1])
ylabel('x(t)'), xlabel('t')
subplot(3,1,2), plot(t,h,'r')
axis([t0,tf,0,2])
ylabel('h(t)'), xlabel('t')
subplot(3,1,3), plot(t,y(1:length(t)),'r')
axis([t0,tf,0,10])
ylabel('y(t)'),xlabel('t')
```

OUTCOME

Activity 2: Perform convolution analytically and draw graphs. Compare the graphs drawn in matlab to the manual graph of analytically derived C.I.

NED University of Engineering & Technology
Department of Electrical Engineering

Course Code: **EE-232** Course Title: **Signals and Systems**
Laboratory Session No.: ________________________ Date: ________________________________

Psychomotor Domain Assessment Rubric for Laboratory (Level P3)					
Skill(s) to be assessed	Extent of Achievement				
	0	1	2	3	4
Software Menu Identification and Usage: Ability to initialise, configure and *operate* software environment *under supervision*, using menus, shortcuts, instructions etc.	Unable to understand and use software menu	Little ability and understanding of software menu operation, makes many mistake	Moderate ability and understanding of software menu operation, makes lesser mistakes	Reasonable understanding of software menu operation, makes no major mistakes	Demonstrates command over software menu usage with frequent use of advance menu options
Modeling of given Network or Signal Processing Scheme: Ability to *operate* software environment in order to create simulation model of given network parameters or signal processing scheme	Unable to operate software, could not create simulation model	Moderately able to operate software, could not create simulation model	Adequately able to operate software, simulation model contains errors	Adequately able to operate software, simulation model is error free	Demonstrates mastery over software, error free simulation model is created and simulation is started successfully
Procedural Programming of given Signal Processing Scheme: *Practice* procedural programming techniques, in order to code specific signal processing schemes	Little to no understanding of procedural programming techniques	Slight ability to use procedural programming techniques for coding given algorithm	Mostly correct recognition and application of procedural programming techniques but makes crucial errors for the given processing scheme	Correctly recognises and uses procedural programming techniques with no errors but unable to run processing scheme successfully	Correctly recognises and uses procedural programming techniques with no errors and runs processing successfully
Detecting and Removing Errors: *Detect* Errors/Exceptions and in simulation model and *manipulate* model to rectify the simulation	Unable to check and detect error messages and indications in software	Able to find error messages and indications in software but no understanding of detecting those errors and their types	Able to find error messages and indications in software as well as understanding of detecting some of those errors and their types	Able to find error messages in software as well as understanding of detecting all of those errors and their types	Able to find error messages in software along with the understanding to detect and rectify them

Psychomotor Domain Assessment Rubric for Laboratory (Level P3)					
Skill(s) to be assessed	Extent of Achievement				
	0	1	2	3	4
Graphical Visualisation and Comparison of Model/Signal Processing Scheme Parameters: <u>*Manipulate*</u> given model/simulation under supervision, in order to produce graphs/plots for measuring and comparing model / signal processing parameters	Unable to understand and utilise visualisation or plotting features	Ability to understand and utilise visualisation and plotting features with frequent errors	Ability to understand and utilise visualisation and plotting features successfully but unable to compare and analyse them	Ability to understand and utilise visualisation and plotting features successfully, partially able to compare and analyse them	Ability to understand and utilise visualisation and plotting features successfully, also able to compare and analyse them
Following step-by-step procedure to complete lab work: <u>*Observe, imitate and operate*</u> software to complete the provided sequence of steps	Inability to recognise and perform given lab procedures	Able to recognise given lab procedures and perform them but could not follow the prescribed order of steps	Able to recognise given lab procedures and perform them by following prescribed order of steps, with frequent mistakes	Able to recognise given lab procedures and perform them by following prescribed order of steps, with occasional mistakes	Able to recognise given lab procedures and perform them by following prescribed order of steps, with no mistakes
Recording Simulation Observations: <u>*Observe and copy*</u> prescribed or required simulation results in accordance with lab manual instructions	Inability to recognise prescribed or required simulation measurements	Able to recognise prescribed or required simulation measurements but does not record according to given instructions	—	Able to recognise prescribed or required simulation measurements but records them incompletely	Able to recognise prescribed or required simulation measurements and records them completely, in tabular form
Discussion and Conclusion: <u>*Demonstrate*</u> discussion capacity on the recorded observations and draw conclusions from it, relating them to theoretical principles/concepts	Complete inability to discuss recorded observations and draw conclusions	Slight ability to discuss recorded observations and draw conclusions	Moderate ability to discuss recorded observations and draw conclusions	Reasonable ability to discuss recorded observations and draw conclusions	Full ability to discuss recorded observations and draw conclusions

Laboratory Session No. 10

Objective:

To solve Ordinary Differential Equation (ODE) using Matlab

Theory:

An ordinary differential equation (ODE) is defined as the equation that contains one or more functions of one independent variable and its derivatives.
$\frac{dy(t)}{dt} + y(t) = t$, where y(t) is function of independent variable t
The solution of such equation contains two parts.

1) Homogeneous Solution, which is computed with assumption that Input to the system is Zero.
2) Particular Solution, which is calculated assuming Zero Initial Conditions.
3) The total solution is linear sum of both homogeneous solution and particular solution as given below:

$$y(t) = y_h(t) + \ y_p(t)$$

Activity 1: Solve following 2nd order ODE using Matlab
$\frac{d^2y(t)}{dt} + \frac{dy(t)}{dt} = x(t)$, given x(t) = 2t, and initial conditions, y(0) = 0 and $\frac{dy(t)}{dt} = 0$

Matlab Script:

POST-LAB ACTIVITIES

Activity 1:

Solve a differential equations solved examples from B.P. Lathi's book suggested in your course using this command and compare it with analytical solutions.
(Write example equation, Matlab expression and fill following table)

Example No.	Differential Equation	Analytical Solution	Remarks

Plot the solution using Matlab

Activity 2: Explore the command 'ODE solver' and use it in the solving of differential equation to achieve complete response choosing any solved examples from your textbook.

NED University of Engineering & Technology
Department of Electrical Engineering

Course Code: **EE-232** Course Title: **Signals and Systems**
Laboratory Session No.: ________________________ Date: ______________________________

Psychomotor Domain Assessment Rubric for Laboratory (Level P3)					
Skill(s) to be assessed	Extent of Achievement				
	0	1	2	3	4
Software Menu Identification and Usage: Ability to initialise, configure and *operate* software environment *under supervision*, using menus, shortcuts, instructions etc.	Unable to understand and use software menu	Little ability and understanding of software menu operation, makes many mistake	Moderate ability and understanding of software menu operation, makes lesser mistakes	Reasonable understanding of software menu operation, makes no major mistakes	Demonstrates command over software menu usage with frequent use of advance menu options
Modeling of given Network or Signal Processing Scheme: Ability to *operate* software environment in order to create simulation model of given network parameters or signal processing scheme	Unable to operate software, could not create simulation model	Moderately able to operate software, could not create simulation model	Adequately able to operate software, simulation model contains errors	Adequately able to operate software, simulation model is error free	Demonstrates mastery over software, error free simulation model is created and simulation is started successfully
Procedural Programming of given Signal Processing Scheme: *Practice* procedural programming techniques, in order to code specific signal processing schemes	Little to no understanding of procedural programming techniques	Slight ability to use procedural programming techniques for coding given algorithm	Mostly correct recognition and application of procedural programming techniques but makes crucial errors for the given processing scheme	Correctly recognises and uses procedural programming techniques with no errors but unable to run processing scheme successfully	Correctly recognises and uses procedural programming techniques with no errors and runs processing successfully
Detecting and Removing Errors: *Detect* Errors/Exceptions and in simulation model and *manipulate* model to rectify the simulation	Unable to check and detect error messages and indications in software	Able to find error messages and indications in software but no understanding of detecting those errors and their types	Able to find error messages and indications in software as well as understanding of detecting some of those errors and their types	Able to find error messages in software as well as understanding of detecting all of those errors and their types	Able to find error messages in software along with the understanding to detect and rectify them

Psychomotor Domain Assessment Rubric for Laboratory (Level P3)					
Skill(s) to be assessed	Extent of Achievement				
	0	1	2	3	4
Graphical Visualisation and Comparison of Model/Signal Processing Scheme Parameters: <u>*Manipulate*</u> given model/simulation under supervision, in order to produce graphs/plots for measuring and comparing model / signal processing parameters	Unable to understand and utilise visualisation or plotting features	Ability to understand and utilise visualisation and plotting features with frequent errors	Ability to understand and utilise visualisation and plotting features successfully but unable to compare and analyse them	Ability to understand and utilise visualisation and plotting features successfully, partially able to compare and analyse them	Ability to understand and utilise visualisation and plotting features successfully, also able to compare and analyse them
Following step-by-step procedure to complete lab work: <u>*Observe, imitate and operate*</u> software to complete the provided sequence of steps	Inability to recognise and perform given lab procedures	Able to recognise given lab procedures and perform them but could not follow the prescribed order of steps	Able to recognise given lab procedures and perform them by following prescribed order of steps, with frequent mistakes	Able to recognise given lab procedures and perform them by following prescribed order of steps, with occasional mistakes	Able to recognise given lab procedures and perform them by following prescribed order of steps, with no mistakes
Recording Simulation Observations: <u>*Observe and copy*</u> prescribed or required simulation results in accordance with lab manual instructions	Inability to recognise prescribed or required simulation measurements	Able to recognise prescribed or required simulation measurements but does not record according to given instructions	—	Able to recognise prescribed or required simulation measurements but records them incompletely	Able to recognise prescribed or required simulation measurements and records them completely, in tabular form
Discussion and Conclusion: <u>*Demonstrate*</u> discussion capacity on the recorded observations and draw conclusions from it, relating them to theoretical principles/concepts	Complete inability to discuss recorded observations and draw conclusions	Slight ability to discuss recorded observations and draw conclusions	Moderate ability to discuss recorded observations and draw conclusions	Reasonable ability to discuss recorded observations and draw conclusions	Full ability to discuss recorded observations and draw conclusions

Laboratory Session No. 11

Objective:

Design and Observe the working of Low Pass Filter

Theory:

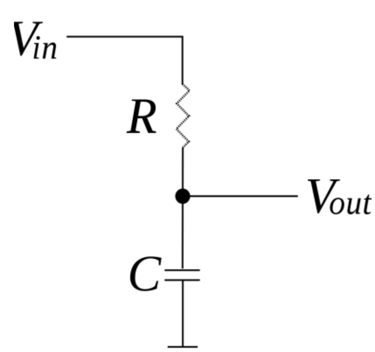
LOW PASS FILTER

A **low-pass filter** (**LPF**) is a filter that passes signals with a frequency lower than a selected cutoff frequency and attenuates signals with frequencies higher than the cutoff frequency. The exact frequency response of the filter depends on the filter design. The filter is sometimes called a **high-cut filter**, or **treble-cut filter** in audio applications.

RC FILTER

A simple RC low-pass filter circuit consists of a resistor in series with a load, and a capacitor in parallel with the load. The capacitor exhibits high reactance at low frequencies, and hence forcing lo the load instead. At higher frequencies the reactance drops, and the capacitor effectively functions as a short circuit. The combination of resistance and capacitance gives the time constant of the filter

$$\tau = RC$$



This circuit may be understood by considering the time the capacitor needs to charge or discharge through the resistor:

- At low frequencies, there is plenty of time for the capacitor to charge up to practically the same voltage as the input voltage.
- At high frequencies, the capacitor only has time to charge up a small amount before the input switches direction. The output goes up and down only a small fraction of the amount the input goes up and down. At double the frequency, there's only time for it to charge up half the amount.

CUT OFF FREQUENCY:

A **cutoff frequency**, **corner frequency**, or **break frequency** of a low-pass filter is a boundary in a system's frequency response at the gain of the system starts to decrease.

It is determined by the time constant:

$$f = \frac{1}{2\pi\tau}$$

GAIN

Gain is the ration of output voltage to input voltage

$$Gain = \frac{Vout}{Vin}$$

Activity 1:

You are given a Cut off frequency. Choose your components accordingly such that when you make observations, the gain starts to drops when frequency of sinusoidal input (from function generator) increases the cut off frequency. The cut off frequency is your roll number if it's one digit, roll number / 2 if it's 2 digit and below 40, if two digit and between 40 and 60, it is roll number /3. The frequencies should be from 5 hertz to 20 hertz. Choose your own formula if you have bigger roll numbers. You may also choose cut off frequencies higher that 20 Hz if supported by function generator and you can get the components accordingly. Plot the graph, Frequency Vs Gain.

Frequency	**Gain(Vout/Vin)**	**Remarks**

Graph Frequency Vs Gain

Cover Page for Each PBL/OEL

Course Code:	EE-232
Course Name:	Signals and Systems
Semester:	
Year:	
Section:	
Batch:	
Lab Instructor name:	
Submission deadline:	

PBL or OEL Statement:

Design a High Pass and a Low Pass Passive RC Filter Using SIMULINK according for given cut-off frequency

Deliverables:

A well written report that will include:

1. Description and circuit diagram of each filter filter, values of resistance and capacitance used to construct the filter, calculation for evaluating the values of resistance and capacitance for given cut-off frequency

2. Plot of Gain Vs Frequency

3. SIMULINK model of both RC Filters

Methodology:

1. Calculate the values of components (resistance and capacitance) for given cut-off frequency and draw circuit diagrams

2. Develop a SIMULINK model of each filter according to circuit diagram with calculated values of the components that also includes variable frequency AC source with signal generator, and necessary scopes.

3. Run the simulation with several values of input wave forms and observe output wave forms

4. Compare the magnitudes of input and output to calculate the gain

5. Plot Gain Vs Frequency

Guidelines:

The report should follow the format given in lab manual

Rubrics:

Attached with the report.

NED University of Engineering & Technology
Department of Electrical Engineering

Course Code: **EE-232** Course Title: **Signals and Systems**
Laboratory Session No.: ________________________ Date: ________________________________

Psychomotor Domain Assessment Rubric for Laboratory (Level P3)					
Skill(s) to be assessed	Extent of Achievement				
	0	1	2	3	4
Software Menu Identification and Usage: Ability to initialise, configure and *operate* software environment *under supervision*, using menus, shortcuts, instructions etc.	Unable to understand and use software menu	Little ability and understanding of software menu operation, makes many mistake	Moderate ability and understanding of software menu operation, makes lesser mistakes	Reasonable understanding of software menu operation, makes no major mistakes	Demonstrates command over software menu usage with frequent use of advance menu options
Modeling of given Network or Signal Processing Scheme: Ability to *operate* software environment in order to create simulation model of given network parameters or signal processing scheme	Unable to operate software, could not create simulation model	Moderately able to operate software, could not create simulation model	Adequately able to operate software, simulation model contains errors	Adequately able to operate software, simulation model is error free	Demonstrates mastery over software, error free simulation model is created and simulation is started successfully
Procedural Programming of given Signal Processing Scheme: *Practice* procedural programming techniques, in order to code specific signal processing schemes	Little to no understanding of procedural programming techniques	Slight ability to use procedural programming techniques for coding given algorithm	Mostly correct recognition and application of procedural programming techniques but makes crucial errors for the given processing scheme	Correctly recognises and uses procedural programming techniques with no errors but unable to run processing scheme successfully	Correctly recognises and uses procedural programming techniques with no errors and runs processing successfully
Detecting and Removing Errors: *Detect* Errors/Exceptions and in simulation model and *manipulate* model to rectify the simulation	Unable to check and detect error messages and indications in software	Able to find error messages and indications in software but no understanding of detecting those errors and their types	Able to find error messages and indications in software as well as understanding of detecting some of those errors and their types	Able to find error messages in software as well as understanding of detecting all of those errors and their types	Able to find error messages in software along with the understanding to detect and rectify them

Psychomotor Domain Assessment Rubric for Laboratory (Level P3)					
Skill(s) to be assessed	Extent of Achievement				
	0	1	2	3	4
Graphical Visualisation and Comparison of Model/Signal Processing Scheme Parameters: *Manipulate* given model/simulation under supervision, in order to produce graphs/plots for measuring and comparing model / signal processing parameters	Unable to understand and utilise visualisation or plotting features	Ability to understand and utilise visualisation and plotting features with frequent errors	Ability to understand and utilise visualisation and plotting features successfully but unable to compare and analyse them	Ability to understand and utilise visualisation and plotting features successfully, partially able to compare and analyse them	Ability to understand and utilise visualisation and plotting features successfully, also able to compare and analyse them
Following step-by-step procedure to complete lab work: *Observe, imitate and operate* software to complete the provided sequence of steps	Inability to recognise and perform given lab procedures	Able to recognise given lab procedures and perform them but could not follow the prescribed order of steps	Able to recognise given lab procedures and perform them by following prescribed order of steps, with frequent mistakes	Able to recognise given lab procedures and perform them by following prescribed order of steps, with occasional mistakes	Able to recognise given lab procedures and perform them by following prescribed order of steps, with no mistakes
Recording Simulation Observations: *Observe and copy* prescribed or required simulation results in accordance with lab manual instructions	Inability to recognise prescribed or required simulation measurements	Able to recognise prescribed or required simulation measurements but does not record according to given instructions	—	Able to recognise prescribed or required simulation measurements but records them incompletely	Able to recognise prescribed or required simulation measurements and records them completely, in tabular form
Discussion and Conclusion: *Demonstrate* discussion capacity on the recorded observations and draw conclusions from it, relating them to theoretical principles/concepts	Complete inability to discuss recorded observations and draw conclusions	Slight ability to discuss recorded observations and draw conclusions	Moderate ability to discuss recorded observations and draw conclusions	Reasonable ability to discuss recorded observations and draw conclusions	Full ability to discuss recorded observations and draw conclusions