LIST OF EXPERIMENTS

1. **Basic Operations on Matrices.**
2. **Generation of Various Signals such as Unit impulse, unit step, square, saw tooth, triangular, sinusoidal, ramp etc.**
3. **Operations on signals and sequences such as addition, multiplication, scaling, shifting, folding, computation of energy and average power.**
4. **Mesh and Nodal analysis of electrical circuits.**
5. **Application of network theorems such as Thevenin's, Norton's, Superposition etc. to electrical networks.**
6. **Locating Zeroes and poles and plotting the pole-zero maps in S plane and for the given TF**
7. **Simulation of DC circuits**
8. **Measurement of Active power of three phase circuit for balanced oads.**
9. **Simulation of single-phase diode bridge rectifiers with filter for R and RL loads.**

VISION

To evolve into a center of excellence in Engineering Technology through creative and innovative practices in teaching-learning, promoting academic achievement & research excellence to produce internationally accepted competitive and world class professionals.

MISSION

To provide high quality academic programmes, training activities, research facilities and opportunities supported by continuous industry institute interaction aimed at employability, entrepreneurship, leadership and research aptitude among students.

QUALITY POLICY

- Impart up-to-date knowledge to the students in Electronics & Communication area to make them quality engineers.
- Make the students experience the applications on quality equipment and tools.
- Provide systems, resources and training opportunities to achieve continuous improvement.
- Maintain global standards in education, training and services.

PROGRAMME EDUCATIONAL OBJECTIVES

PEO1: PROFESSIONALISM & CITIZENSHIP To create and sustain a community of learning in which students acquire knowledge and learn to apply it professionally with due consideration for ethical, ecological and economic issues.
PEO2: TECHNICAL ACCOMPLISHMENTS To provide knowledge based services to satisfy the needs of society and the industry by providing hands on experience in various technologies in core field.
PEO3: INVENTION, INNOVATION AND CREATIVITY To make the students to design, experiment, analyze, interpret in the core field with the help of other multi disciplinary concepts wherever applicable.
PEO4: PROFESSIONAL DEVELOPMENT To educate the students to disseminate research findings with good soft skills and become a successful entrepreneur.
PEO5: HUMAN RESOURCE DEVELOPMENT To graduate the students in building national capabilities in technology, education and research.

CODE OF CONDUCT FOR THE LABORATORIES

- All students must observe the Dress Code while in the laboratory.
- Sandals or open-toed shoes are NOT allowed.
- Foods, drinks and smoking are NOT allowed.
- All bags must be left at the indicated place.
- The lab timetable must be strictly followed.
- Be PUNCTUAL for your laboratory session.
- Program must be executed within the given time.
- Noise must be kept to a minimum.
- Workspace must be kept clean and tidy at all time.
- Handle the systems and interfacing kits with care.
- All students are liable for any damage to the accessories due to their own negligence.
- All interfacing kits connecting cables must be RETURNED if you taken from the labsupervisor.
- Students are strictly PROHIBITED from taking out any items from the laboratory.
- Students are NOT allowed to work alone in the laboratory without the Lab Supervisor
- USB Ports have been disabled if you want to use USB drive consult lab supervisor.
- Report immediately to the Lab Supervisor if any malfunction of the accessories, is there.

Before leaving the lab

- Place the chairs properly.
- Turn off the system properly
- Turn off the monitor.
- Please check the laboratory notice board regularly for updates.

Experiment No-1

BASIC OPERATIONS ON MATRICES

AIM: Generate a matrix and perform basic operation on matrices using MATLAB software.

Software Required: MATLAB software

Theory:

MATLAB treats all variables as matrices. Vectors are special forms of matrices and contain only one row or one column. Whereas scalars are special forms of matrices and contain only one row and one column. A matrix with one row is called row vector and a matrix with single column is called column vector.

The first one consists of convenient matrix building functions, some of which are given below.

1. eye - identity matrix
2. zeros - matrix of zeros
3. ones - matrix of ones
4. diag - extract diagonal of a matrix or create diagonal matrices
5. triu - upper triangular part of a matrix
6. tril - lower triangular part of a matrix
7. rand - ran

commands in the second sub-category of matrix functions are

1. size- size of a matrix
2. det -determinant of a square matrix
3. inv- inverse of a matrix
4. rank- rank of a matrix
5. rref- reduced row echelon form
6. eig- eigenvalues and eigenvectors
7. poly- characteristic polynomialdomly generated matrix

Program:

% Creating a column vector

>> a=[1;2;3]

a =

1

2

3

% Creating a row vector

>> b=[1 2 3]

b=

1 2 3

% Creating a matrix

>> m=[1 2 3;4 6 9;2 6 9]

m =

1 2 3

4 6 9

2 6 9

% Extracting sub matrix from matrix

>> sub_m=m(2:3,2:3)

sub_m =

6 9

6 9

% extracting column vector from matrix

>> c=m (:,2)

c =

2
6
6
% extracting row vector from matrix

>> d=m (3,:)

d =

2 6 9

% creation of two matrices a and b

>> a=[2 4 -1;-2 1 9;-1 -1 0]

a =

2 4 -1

-2 1 9

-1 -1 0

>> b=[0 2 3;1 0 2;1 4 6]

b =

0 2 3

1 0 2

1 4 6

% matrix multiplication

>>

x1=a

*bx1

=

3 0 8

10 32 50

```
-1 -2 -5
% element to element multiplication
>>
x2=a.
*bx2
=
0 8 -3
-2 0 18
-1 -4 0
% matrix addition
>>
x3=a
+bx3
=
2 6 2
-1 1 11
0 3 6
% matrix subtraction
>> x4=a-b
x4 =
2 2 -4
-3 1 7
-2 -5 -6
% matrix division
>> x5=a/b
x5 =
-9.0000 -3.5000 5.5000
12.0000 3.7500 -5.7500
3.0000 0.7500 -1.7500
% element to element division
>> x6=a./b
Warning: Divide by zero.
x6 =
Inf 2.0000 -0.3333
-2.0000 Inf 4.5000
-1.0000 -0.2500 0
```

% inverse of matrix a

>> x7=inv(a)

x7 =

-0.4286 -0.0476 -1.7619

0.4286 0.0476 0.7619

-0.1429 0.0952 -0.4762

% transpose of matrix a

>> x8=a'

x8 =

2 -2 -1

4 1 -1

-1 9 0

RESULT: Matrix operations are performed using MATLAB software.

EXPERIMENT NO-2

Generation of signals and sequences

AIM: Generate various signals and sequences (Periodic and aperiodic), such as Unit Impulse, Unit Step, Square, Saw tooth, Triangular, Sinusoidal, Ramp, Sinc.

Software Required: Matlab software

Theory: If the amplitude of the signal is defined at every instant of time then it is called continuous time signal. If the amplitude of the signal is defined at only at some instants of time then it is called discrete time signal. If the signal repeats itself at regular intervals then it is called periodic signal. Otherwise they are called aperiodic signals.

EX: ramp,Impulse,unit step, sinc- Aperiodic signals

square,sawtooth,triangular sinusoidal – periodic signals.

Ramp sinal: The **ramp function** is a unitary real function, easily computable as the mean of the independent variable and its absolute value.This function is applied in engineering. The name *ramp function* is derived from the appearance of its graph.

$$\mathbf{r(t)=} \begin{cases} \mathbf{t \ \ when} \\ \mathbf{0 \quad else} \end{cases}$$

Unit impulse signal: One of the more useful functions in the study of linear systems is the "unit impulse function."An ideal impulse function is a function that is zero everywhere but at the origin, where it isinfinitely high. However, the *area* of the impulse is finite

Y(t)= 1 when t=0

=0 other wise

Unit step signal: The unit step function and the impulse function are considered to be fundamental functions in engineering, and it is strongly recommended that the reader becomes very familiar with both of these functions.

$$\mathbf{u(t)=} \begin{cases} \mathbf{0 \text{ if } t < 0} \\ \mathbf{1 \text{ if } t > 0 If} \\ \mathbf{\frac{1}{2} \quad t=0} \end{cases}$$

Sinc signal: There is a particular form that appears so frequently in communications engineering, that we give it its own name. This function is called the "Sinc function”.
The Sinc function is defined in the following manner:

$$\boldsymbol{sinc(x) = \frac{\sin \pi x}{\pi x}} \quad \textbf{if } \boldsymbol{x \neq 0} \textbf{ and sinc(0) =1}$$

The value of sinc(x) is defined as 1 at x = 0, since

$$\mathbf{\lim_{x \to 0} \boldsymbol{sinc(x)} = 1}$$

PROCEDURE:-

- Open MATLAB
- Open new M-file
- Type the program
- Save in current directory
- Compile and Run the program
- For the output see command window\ Figure window

PROGRAM:

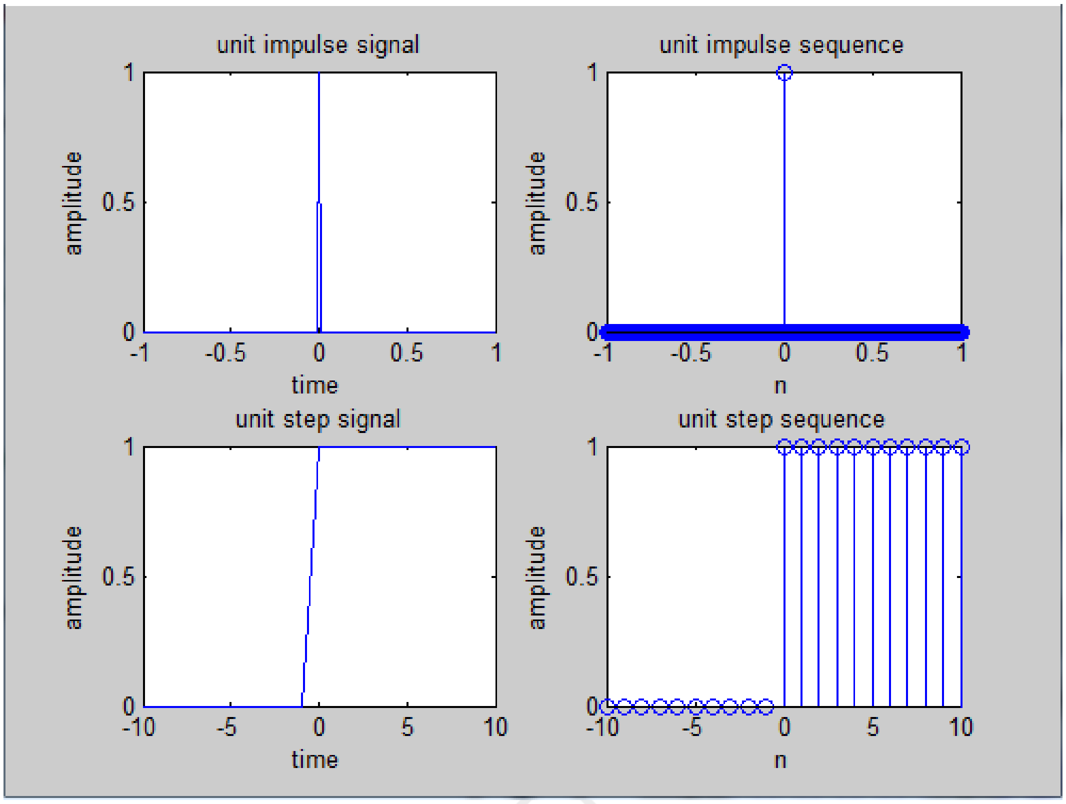
```
% Generation of signals and sequences
clc;
clear all;
close all;
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%generation of unit impulse signal
t1=-1:0.01:1
y1=(t1==0);
subplot(2,2,1);
plot(t1,y1);
xlabel('time');
ylabel('amplitude');
title('unit impulse signal');
%generation of impulse sequence
subplot(2,2,2);
stem(t1,y1);
xlabel('n');
ylabel('amplitude');
title('unit impulse sequence');
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

%generation of unit step signal
t2=-10:1:10;
y2=(t2>=0);
subplot(2,2,3);
plot(t2,y2);
xlabel('time');
ylabel('amplitude');
title('unit step signal');
%generation of unit step sequence
subplot(2,2,4);
stem(t2,y2);
xlabel('n');
ylabel('amplitude');
title('unit step sequence');
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

%generation of square wave signal
t=0:0.002:0.1;
y3=square(2*pi*50*t);
```

```
figure;
subplot(2,2,1);
plot(t,y3);
axis([0 0.1 -2 2]);
xlabel('time');
ylabel('amplitude');
title('square wave signal');
%generation of square wave sequence
subplot(2,2,2);
stem(t,y3);
axis([0 0.1 -2 2]);
xlabel('n');
ylabel('amplitude');
title('square wave sequence');
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

%generation of sawtooth signal
y4=sawtooth(2*pi*50*t);
subplot(2,2,3);
plot(t,y4);
axis([0 0.1 -2 2]);
xlabel('time');
ylabel('amplitude');
title('sawtooth wave signal');
%generation of sawtooth sequence
subplot(2,2,4);
stem(t,y4);
axis([0 0.1 -2 2]);
xlabel('n');
ylabel('amplitude');
title('sawtooth wave sequence');
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

%generation of triangular wave signal
y5=sawtooth(2*pi*50*t,.5);
figure;
subplot(2,2,1);
plot(t,y5);
axis([0 0.1 -2 2]);
xlabel('time');
ylabel('amplitude');
title(' triangular wave signal');
%generation of triangular wave sequence
subplot(2,2,2);
stem(t,y5);
axis([0 0.1 -2 2]);
xlabel('n');
ylabel('amplitude');
title('triangular wave sequence');
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%generation of sinsoidal wave signal
```

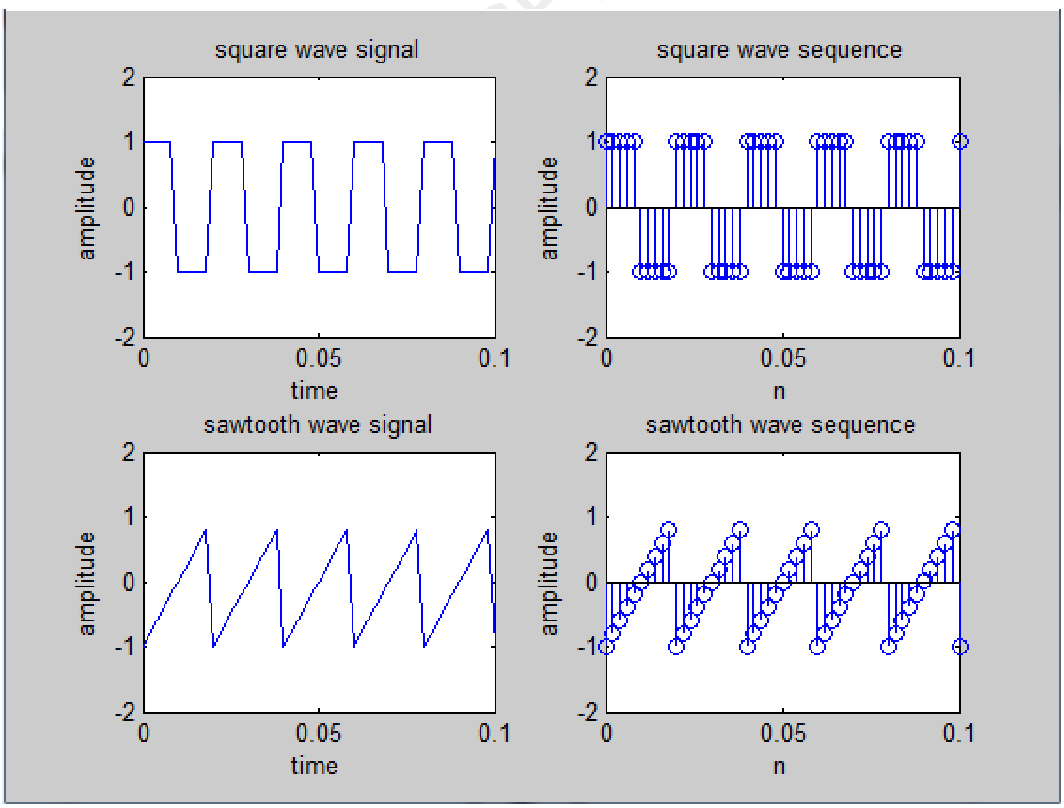
```
y6=sin(2*pi*40*t);
subplot(2,2,3);
plot(t,y6);
axis([0 0.1 -2 2]);
xlabel('time');
ylabel('amplitude');
title(' sinsoidal wave signal');
%generation of sin wave sequence
subplot(2,2,4);
stem(t,y6);
axis([0 0.1 -2 2]);
xlabel('n');
ylabel('amplitude');
title('sin wave sequence');
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

%generation of ramp signal
y7=t;
figure;
subplot(2,2,1);
plot(t,y7);
xlabel('time');
ylabel('amplitude');
title('ramp signal');
%generation of ramp sequence
subplot(2,2,2);
stem(t,y7);
xlabel('n');
ylabel('amplitude');
title('ramp  sequence');
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

%generation of sinc signal
t3=linspace(-5,5);
y8=sinc(t3);
subplot(2,2,3);
plot(t3,y8);
xlabel('time');
ylabel('amplitude');
title(' sinc signal');
%generation of sinc  sequence
subplot(2,2,4);
stem(y8);
xlabel('n');
ylabel('amplitude');
title('sinc sequence');
```

Result: Various signals & sequences generated using Matlab software.

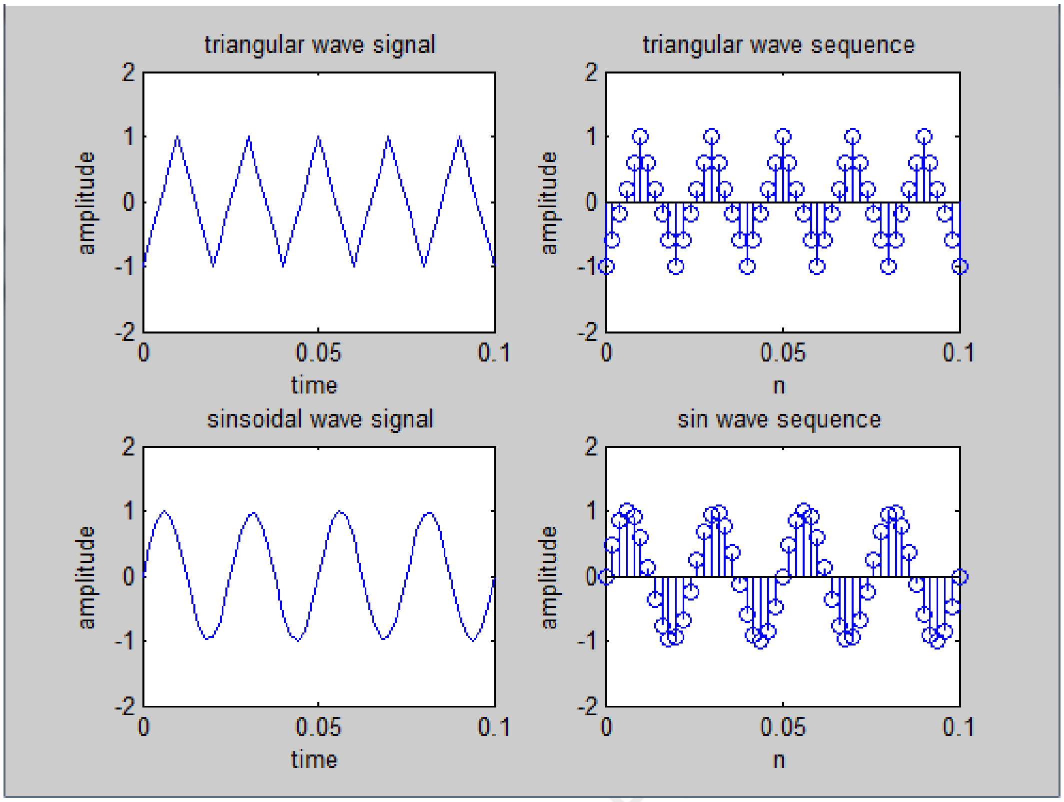
output:



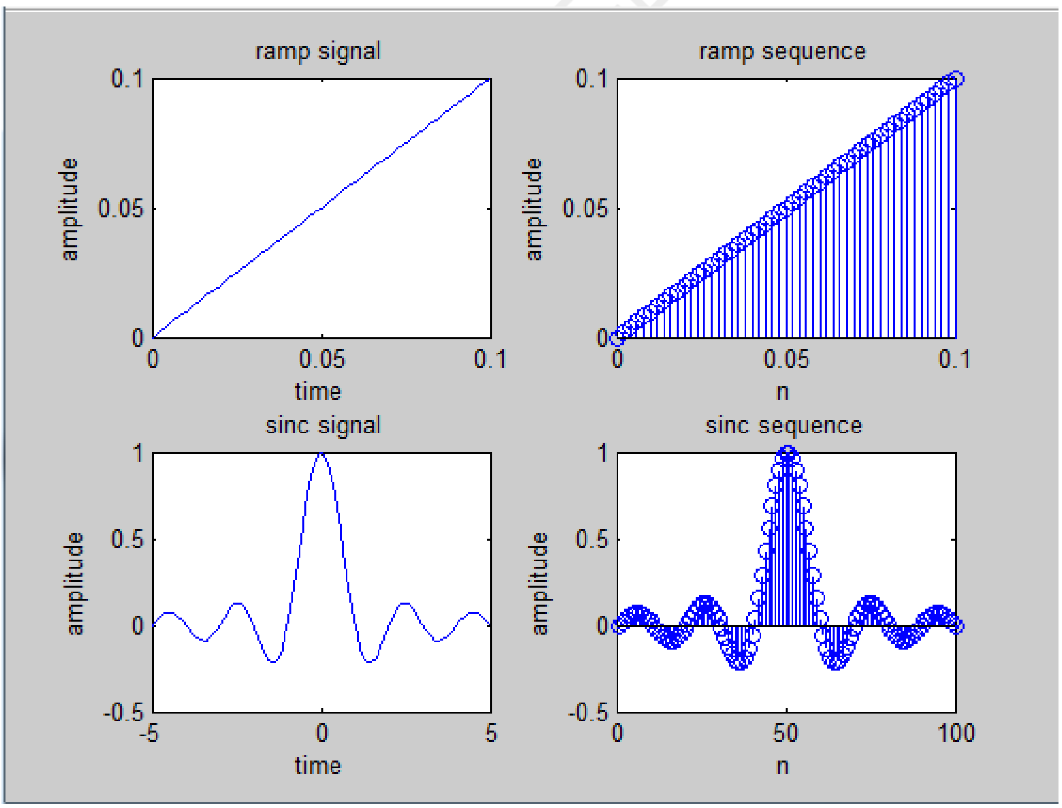
unit impulse signal
unit impulse sequence
unit step signal
unit step sequence
amplitude
time
n



square wave signal
square wave sequence
sawtooth wave signal
sawtooth wave sequence
amplitude
time
n



triangular wave signal
triangular wave sequence
sinsoidal wave signal
sin wave sequence
amplitude
time
n



ramp signal
ramp sequence
sinc signal
sinc sequence
amplitude
time
n

EXPERIMENT NO-3
Basic Operations on Signals and sequences

AIM: perform the operations on signals and sequences such as addition, multiplication, scaling, shifting, folding and also compute energy and power.

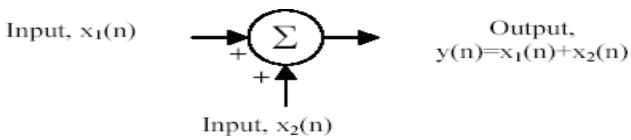
Software Required: Matlab software.

Theory:

Signal Addition

Addition: any two signals can be added to form a third signal,

z (t) = x (t) + y (t)



Multiplication :

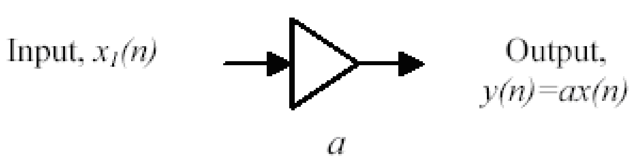
Multiplication of two signals can be obtained by multiplying their values at every instants . z
z(t) = x (t) y (t)

Time reversal/Folding:

Time reversal of a signal x(t) can be obtained by folding the signal about t=0.
Y(t)=y(-t)

Signal Amplification/Scaling : Y(n)=ax(n) if a < 1 attnuation

a >1 amplification



Time shifting: The time shifting of x(n) obtained by delay or advance the signal in time by using y(n)=x(n+k)

If k is a positive number, y(n) shifted to the right i e the shifting delays the signal

If k is a negative number, y(n) it gets shifted left. Signal Shifting advances the signal

Energy:

$$E[n] = \lim_{N \to \infty} \sum_{n=-N}^{N} |x[n]|^2 = \sum_{n=-\infty}^{\infty} |x[n]|^2$$

Average power

$$P[n] = \lim_{N \to \infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |x[n]|^2$$

Program:

```
clc;
clear all;
close all;
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% generating two input signals
t=0:.01:1;
x1=sin(2*pi*4*t);
x2=sin(2*pi*8*t);
subplot(2,2,1);
plot(t,x1);
xlabel('time');
ylabel('amplitude');
title('input signal 1');
subplot(2,2,2);
plot(t,x2);
xlabel('time');
ylabel('amplitude');
title('input signal 2');

% addition of signals
y1=x1+x2;
subplot(2,2,3);
plot(t,y1);
xlabel('time');
ylabel('amplitude');
title('addition of two signals');

% multiplication of signals
y2=x1.*x2;
subplot(2,2,4);
plot(t,y2);
xlabel('time');
ylabel('amplitude');
title('multiplication of two signals');
% scaling of a signal1
A=2;
y3=A*x1;
figure;
subplot(2,2,1);
plot(t,x1);
xlabel('time');
ylabel('amplitude');
title('input signal')
subplot(2,2,2);
plot(t,y3);
xlabel('time');
ylabel('amplitude');
title('amplified input signal');

% folding of a signal1
```

```
h=length(x1);
nx=0:h-1;
subplot(2,2,3);
plot(nx,x1);
xlabel('nx');
ylabel('amplitude');
title('input signal')
y4=fliplr(x1);
nf=-fliplr(nx);
subplot(2,2,4);
plot(nf,y4);
xlabel('nf');
ylabel('amplitude');
title('folded signal');

%shifting of a signal 1
figure;
subplot(3,1,1);
plot(t,x1);
xlabel('time t');
ylabel('amplitude');
title('input signal');
subplot(3,1,2);
plot(t+2,x1);
xlabel('t+2');
ylabel('amplitude');
title('right shifted signal');
subplot(3,1,3);
plot(t-2,x1);
xlabel('t-2');
ylabel('amplitude');
title('left shifted signal');
%~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~
~~~~~~%operations on
sequencesn1=1:1:9;
s1=[1 2 3 0 5 8 0 2 4];
figure;
subplot(2,2,1);
stem(n1,s1);
xlabel('n1');
ylabel('amplitude');
title('input sequence1');
s2=[1 1 2 4 6 0 5 3 6];
subplot(2,2,2);
stem(n1,s2);
xlabel('n2');
ylabel('amplitude');
title('input  sequence2');

% addition of sequences
s3=s1+s2;
subplot(2,2,3);
stem(n1,s3);
```

```
xlabel('n1');
ylabel('amplitude');
title('sum of two sequences');

% multiplication  of sequences
s4=s1.*s2;
subplot(2,2,4);
stem(n1,s4);
xlabel('n1');
ylabel('amplitude');
title('product of two sequences');
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% program for energy of a sequence
z1=input('enter the input sequence');
e1=sum(abs(z1).^2);
disp('energy of given sequence is');e1

% program for energy of a signal
t=0:pi:10*pi;
z2=cos(2*pi*50*t).^2;
e2=sum(abs(z2).^2);
disp('energy of given signal is');e2

% program for power of a sequence
p1= (sum(abs(z1).^2))/length(z1);
disp('power of given sequence is');p1

% program for power of a signal
p2=(sum(abs(z2).^2))/length(z2);
disp('power of given signal is');p2
```

OUTPUT:

enter the input sequence[1 3 2 4 1]

energy of given sequence is

e1 = 31

energy of given signal is

e2 = 4.0388

power of given sequence is

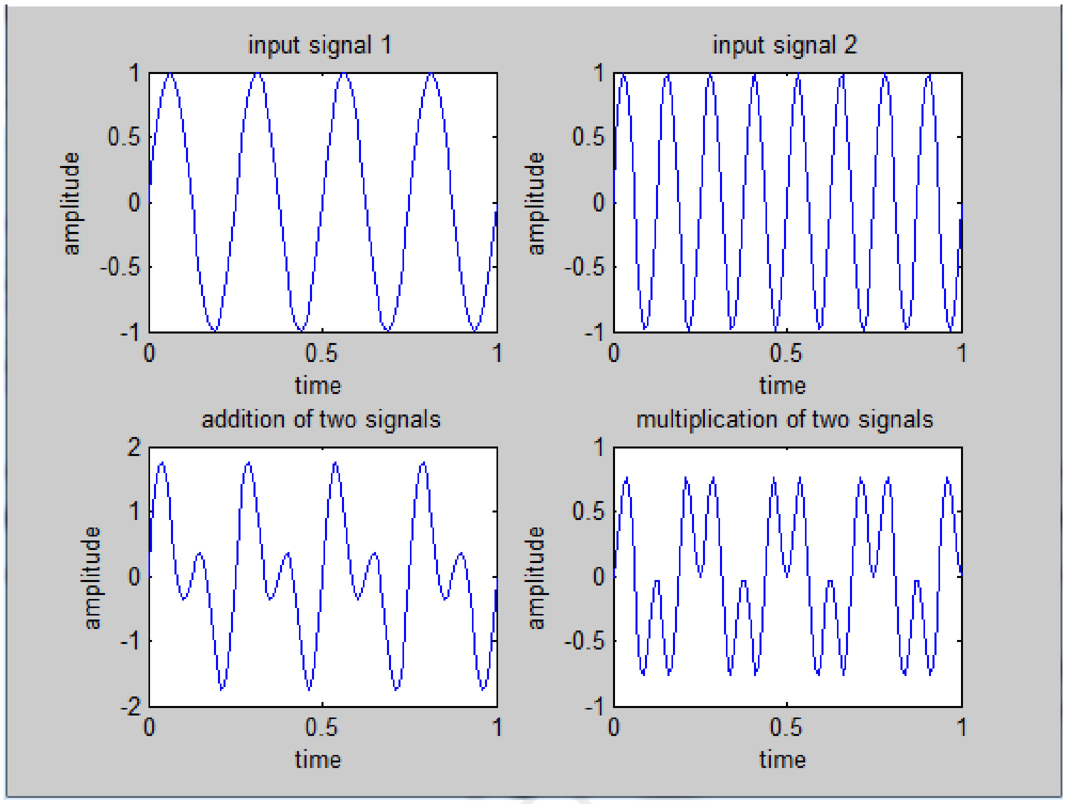
p1 = 6.2000

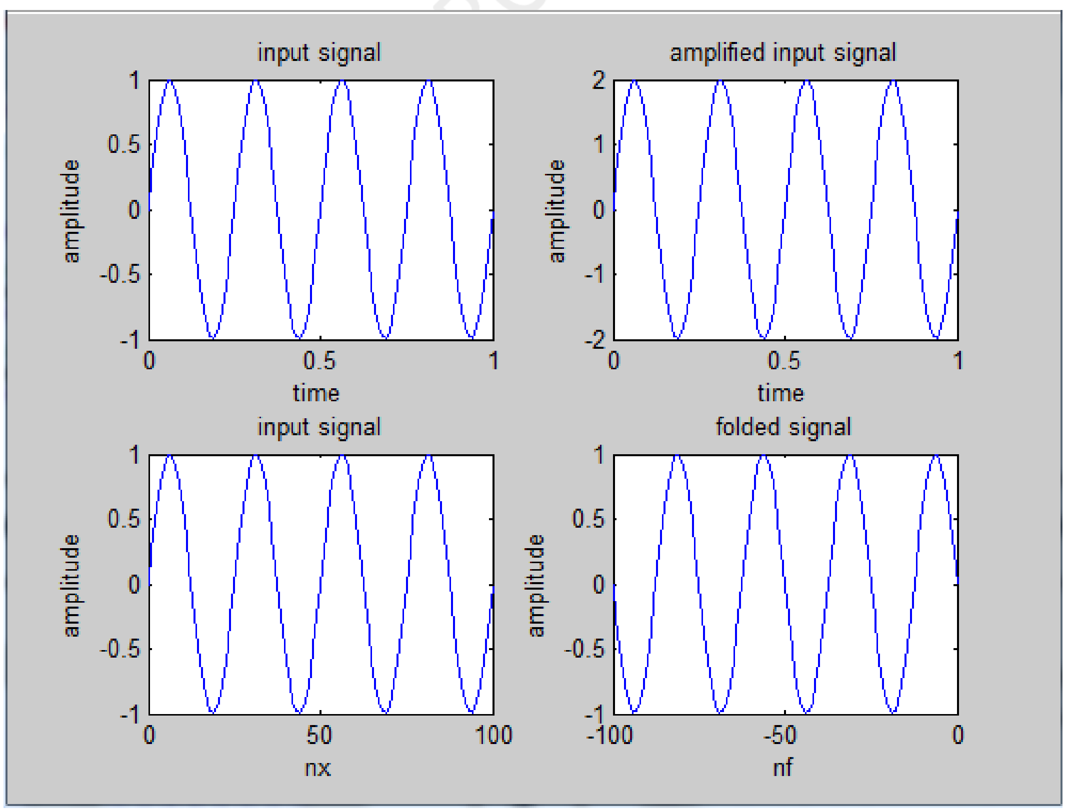
power of given signal is

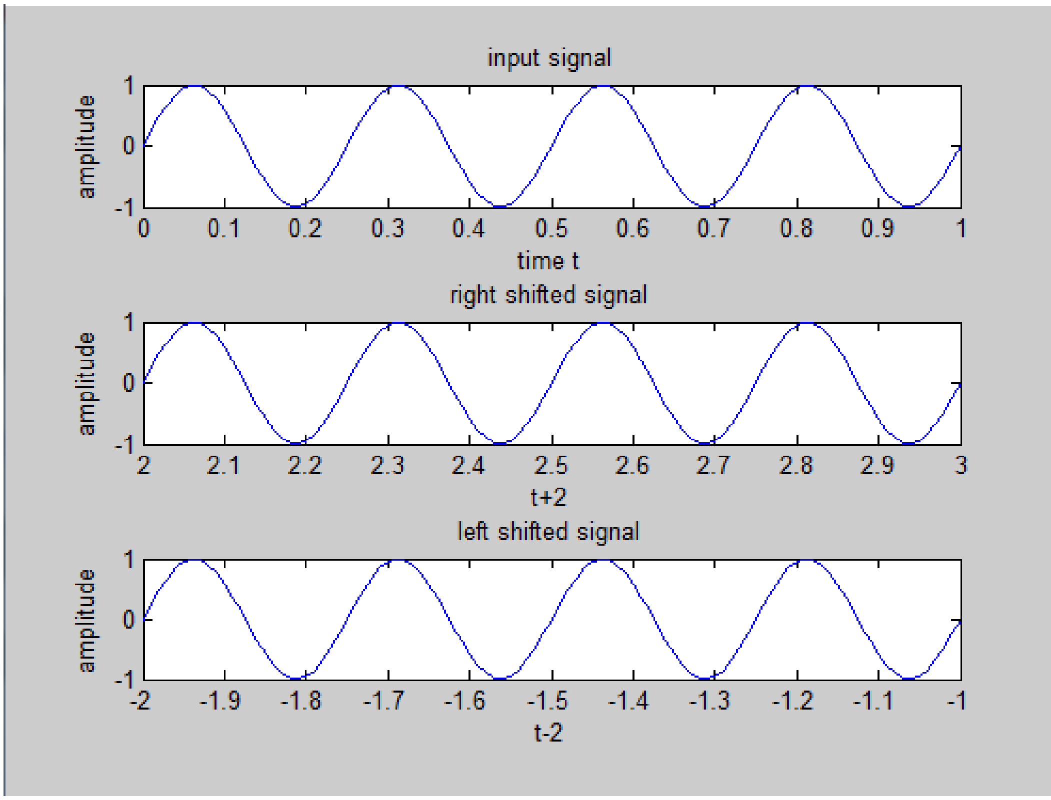
p2 =   0.3672

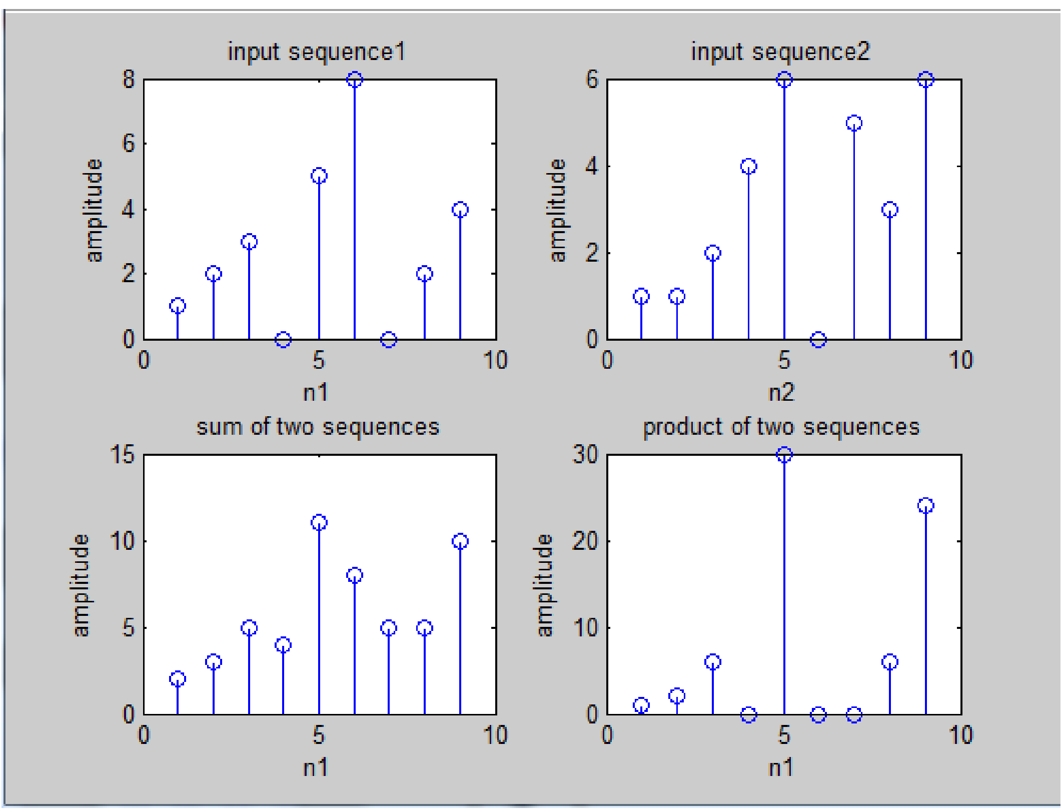
Result: Various operations on signals and sequences are performed.

Output:


input signal 1
input signal 2
addition of two signals
multiplication of two signals
amplitude
time


input signal
amplified input signal
input signal
folded signal
amplitude
time
nx
nf

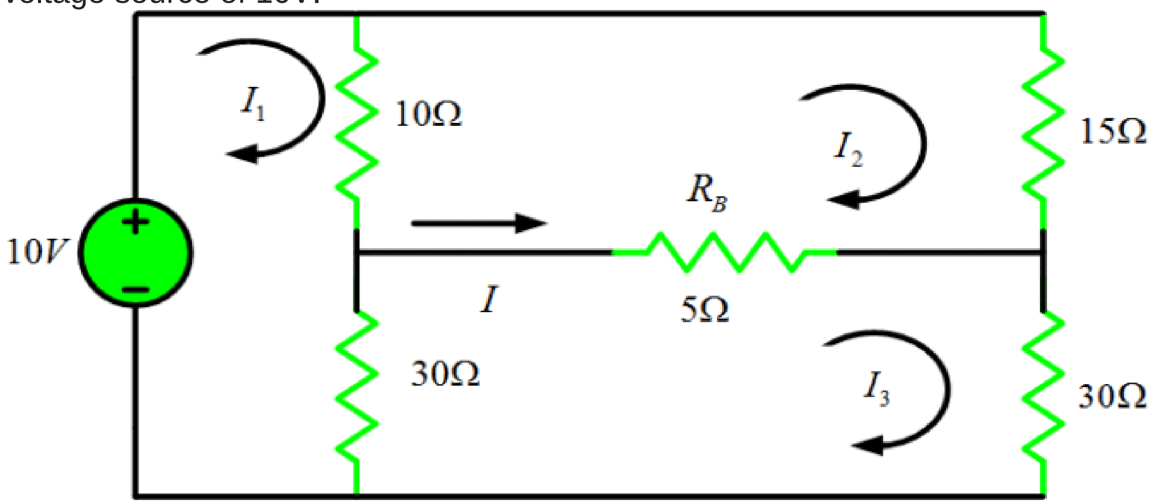

input signal
amplitude
time t
right shifted signal
amplitude
t+2
left shifted signal
amplitude
t-2


input sequence1
amplitude
n1
input sequence2
amplitude
n2
sum of two sequences
amplitude
n1
product of two sequences
amplitude
n1

Experiment No-4
Mesh and Nodal analysis of electrical circuits.

Aim – analyzing mesh and node of electrical circuit.
Software Required: Matlab software

we will find current which is flowing through resistor R_B and the power supplied by the voltage source of 10V.



First, let's assign currents for each loop as I_1, I_2 and I_3 and the power supplied by the source is $10*I_1$ as we can see from the circuit.
Now, let's write the loop equations for each loop:
Loop 1:

$$10(I_1-I_2)+30(I_1-I_3)-10=0$$
$$-10I_1-10I_2-30I_3=10\cdots(1)$$

Loop 2:

$$10(I_2-I_1)+15I_2+5(I_2-I_3)=0$$
$$-10I_1+30I_2-5I_3=0\cdots(2)$$

Loop 3:

$$30(I_3-I_1)+5(I_3-I_2)+30I_3=0$$
$$-30I_1-5I_2+65I_3=0\cdots(3)$$

Now, let's write (1), (2), and (3) in matrix form as:

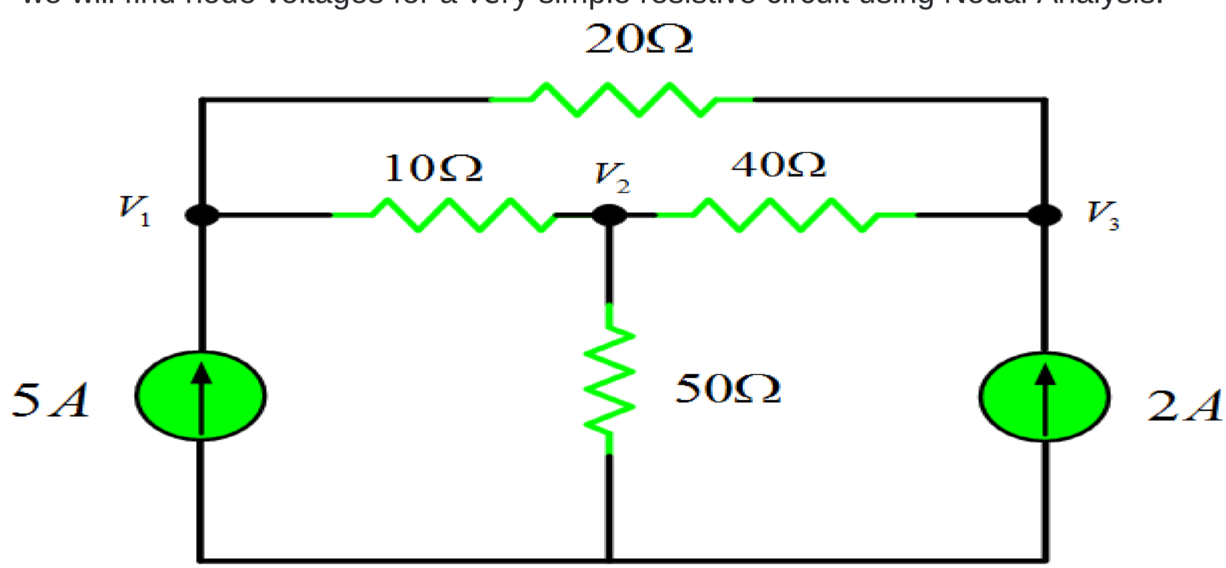
$$\begin{bmatrix} 40 & -10 & -30 \\ -10 & 30 & -5 \\ -30 & -5 & 65 \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix}$$

MATLAB CODING-

```
clear all;close all;clc
% Loop Analysis using Matlab
R_Mat = [40 -10 -30;
-10 30 -5; % Impedance (or Resistance) Matrix obtain from Loop equations
-30 -5 65];
V_vec = [10 0 0]'; % Voltage Vector (again from Loop equations)
%% % Loop Currents Calculations
I_Loop = inv(R_Mat)*V_vec;
% Calculate current flowing through Resistor R_B
I_RB = I_Loop(3) - I_Loop(2); % (I=I3-I2)
fprintf('The current flowing through Resistor R_B is %8.3f A \n',I_RB)
% Calculate the total power supplied by 10V source
P_Source = I_Loop(1)*10; % (P=10*I_1)
fprintf('The power supplied by voltage source of 10V is %8.4f watts
\n',P_Source)
```

Results:
The current flowing through Resistor R_B is 0.037 A
The power supplied by voltage source of 10V is 4.7531 watts

*we will find node voltages for a very simple resistive circuit using Nodal Analysis.



While applying KCL, we will assume that currents leaving the node are positive and entering the node are negative. Keeping that fact in mind, let's write node voltages for each node in the circuit.

Node 1

$$(V_1 - V_2)10 + (V_1 - V_3)10 - 5 = 0$$

$$0.15V_1 - 0.1V_2 - 0.05V_3 = 5 \cdots (1)$$

Node 2:

$$(V_2-V_1)10+V_2 50+(V_2-V_3)40=0$$
$$-0.10V_1+0.145V_2-0.025V_3=0\cdots(2)$$

Node 3:

$$(V_3-V_1)20+(V_3-V_2)40-2=0$$
$$-0.05V_1-0.025V_2+0.075V_3=2\cdots(3)$$

Let's combine all (1), (2), and (3) in Matrix form

$$\begin{bmatrix} 0.15 & -0.1 & -0.05 \\ -0.1 & 0.145 & -0.025 \\ -0.05 & -0.025 & 0.075 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} 5 \\ 0 \\ 2 \end{bmatrix}$$

Matlab coding

```
clear all;close all;clc
% Nodal Analysis using Matlab
Y_Mat = [ 0.15 -0.1 -0.05;
-0.1 0.145 -0.025; % Admittance Matrix (YV=I) obtain from Node equations
-0.05 -0.025 0.075];
I_vec = [5;
0; % Current Vector (again from Node equations)
2];
%% Node Voltages Calculation
fprintf('Nodal voltages V1, V2 and V3 are \n')
V_Node = inv(Y_Mat)*I_vec
```

Results:

Nodal voltages V_1, V_2, and V_3 are

V_Node =

404.2857

350.0000

412.8571

EXPERIMENT NO: 5

VERIFICATION OF NETWORK THEOREMS

I) **SUPERPOSITION THEOREM.**
II) **THEVENIN'S THEOREM.**
III) **MAXIMUM POWER TRANSFER THEOREM.**

AIM: To verify Superposition theorem, Thevenin's theorem, Norton's theorem and Maximum power Transfer theorem.

SOFTWARE USED : MULTISIM / MATLAB Simulink

SUPERPOSITION THEOREM:

"In a linear network with several independent sources which include equivalent sources due to initial conditions, and linear dependent sources, the overall response in any part of the network is equal to the sum of individual responses due to each independent source, considered separately, with all other independent sources reduced to zero".

Procedure:

Step 1:

1. Make the connections as shown in the circuit diagram by using MULTISIM/MATLAB Simulink.

2. Measure the response 'I' in the load resistor by considering all the sources 10V, 15V and 8V in the network.

Step 2:

1. Replace the sources 15V and 8V with their internal impedances (short circuited).

2. Measure the response 'I1' in the load resistor by considering 10V source in the network.

Step 3:

1. Replace the sources 10V and 8V with their internal impedances (short circuited).

2. Measure the response 'I2' in the load resistor by considering 15V source in the network.

Step 4:

1. Replace the sources 10V and 15V with their internal impedances (short circuited).

2. Measure the response 'I3' in the load resistor by considering 8V source in the network.

The responses obtained in step 1 should be equal to the sum of the responses obtained in step 2, 3 and 4.

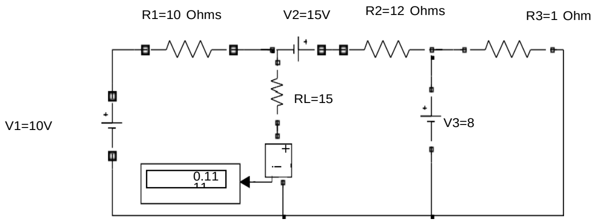
$$I=I1+I2+I3$$

Hence Superposition Theorem is verified.

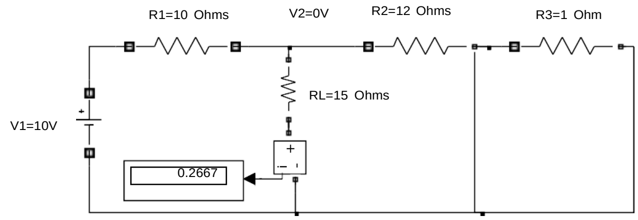
Continuous

powergui

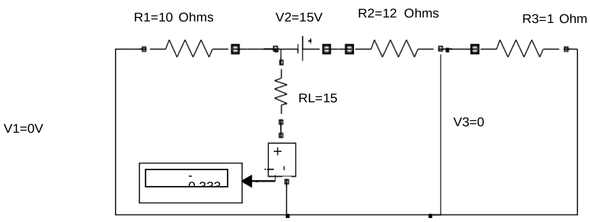
Step 1 : By Considering All Sources In The Network



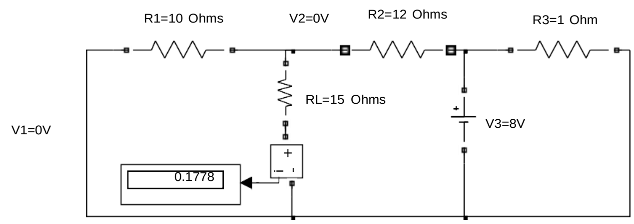
Step 2 : By Considering 10 V Sources In The Network



Step 3 : By Considering 15 V Sources In The Network



Step 4 : By Considering 8V Sources In The Network



Current through Load Resistor 15 Ohms :

Considerning 10V Source I1: 0.2667A
Considering 15V Source I2 : - 0.3333A
Considering 8V Source I3 : 0.1778A

Total Current : I1+I2+I3=0.2667-0.3333+0.1778
=0.1112A

With all the sources in the network I = 0.1111A
I=I1+I2+I3

Hence SuperPosition Theorem is Verified.

THEVENIN'S THEOREM:

"Any two terminal network consisting of linear impedances and generators may be replaced at the two terminals by a single voltage source acting in series with an impedance. The voltage of the equivalent source is the open circuit voltage measured at the terminals of the network and the impedance, known as Thevenin's equivalent impedance, Z_{TH}, is the impedance measured at the terminals with all the independent sources in the network reduced to zero".

Procedure:

Step 1:

1. Make the connections as shown in the circuit diagram by using MULTISIM/MATLAB Simulink.

2. Measure the response 'I' in the load resistor by considering all the sources in the network.

Step 2: Finding Thevenin's Resistance(R_{TH})

1. Open the load terminals and replace all the sources with their internal impedances.

2. Measure the impedance across the open circuited terminal which is known as Thevenin's Resistance.

Step 3: Finding Thevenin's Voltage(V_{TH})

1. Open the load terminals and measure the voltage across the open circuited terminals.

2. Measured voltage will be known as Thevenin's Voltage.

Step 4: Thevenin's Equivalent Circuit

1. V_{TH} and R_{TH} are connected in series with the load.

2. Measure the current through the load resistor $I_L = \frac{V_{TH}}{R_{TH}+R_L}$.

Current measured from Thevenin's Equivalent Circuit should be same as current obtained from the actual circuit.

$$I = I_L.$$

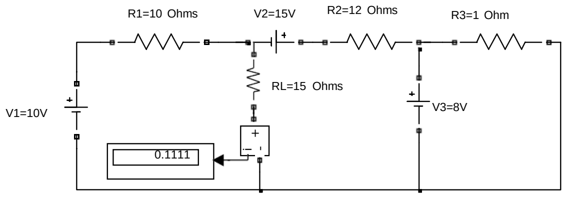
Hence Thevenin's Theorem is Verified.

Continuous

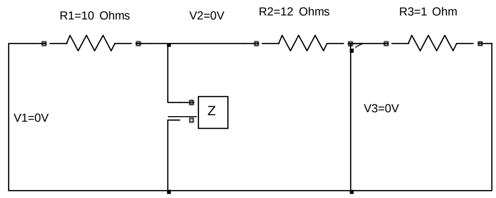
powergui

THEVENIN'S THEOREM

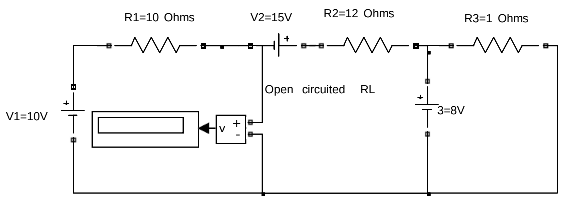
Step 1 : By Considering All Sources In The Network



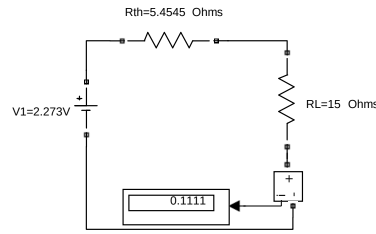
Step 2: Finding Thevenin's Resistance



Step 3 : Finding Thevenin's Voltage



Step 4 : Thevenin's Equivalent Network



Open Circuit Voltage Vth = 2.273V
Thevenin's Resistance = 5.4545Ohms
Current through Load Resistor 15 Ohms IL = 0.1111A

With all the sources in the network Current through Load Resistor 15 Ohms : I=0.1111A

I=IL

Hence Thevenin's Theorem is Verified.

NORTON’S THEOREM:

“Any two terminal network consisting of linear impedances and generators may be replaced at its two terminals, by an equivalent network consisting of a single current source in parallel with an impedance. The equivalent current source is the short circuit current measured at the terminals and the equivalent impedance is same as the Thevenin’s equivalent impedance”.

Procedure:

Step 1:

1. Make the connections as shown in the circuit diagram by using MULTISIM/MATLAB Simulink.
2. Measure the response ‘I’ in the load resistor by considering all the sources in the network.

Step 2: Finding Norton’s Resistance(R_N)

1. Open the load terminals and replace all the sources with their internal impedances.
2. Measure the impedance across the open circuited terminal which is known as Norton’s Resistance.

Step 3: Finding Norton’s Current(I_N)

1. Short the load terminals and measure the current through the short circuited terminals.
2. Measured current is be known as Norton’s Current.

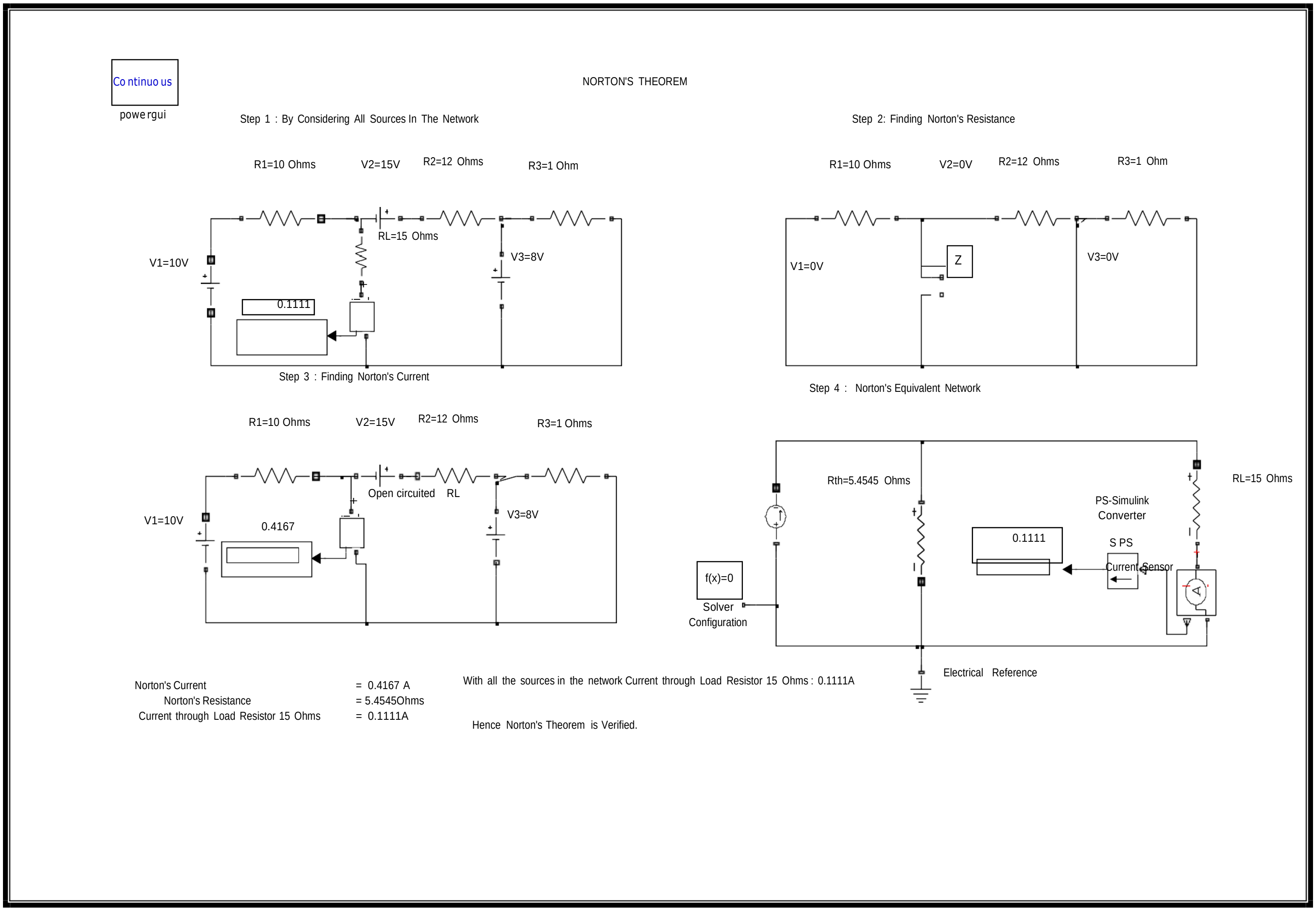
Step 4: Norton’s Equivalent Circuit

1. R_N and I_N are connected in parallel to the load.
2. Measure the current through the load resistor $I_L = \frac{I_N R_N}{R_N + R_L}$.

Current measured from Norton’s Equivalent Circuit should be same as current obtained from the actual circuit.

$$I = I_L.$$

Hence Norton’s Theorem is Verified.


Continuous
powergui
NORTON'S THEOREM
Step 1 : By Considering All Sources In The Network
R1=10 Ohms
V2=15V
R2=12 Ohms
R3=1 Ohm
RL=15 Ohms
V1=10V
V3=8V
0.1111
Step 2: Finding Norton's Resistance
R1=10 Ohms
V2=0V
R2=12 Ohms
R3=1 Ohm
V1=0V
Z
V3=0V
Step 3 : Finding Norton's Current
R1=10 Ohms
V2=15V
R2=12 Ohms
R3=1 Ohms
Open circuited RL
V1=10V
0.4167
V3=8V
Step 4 : Norton's Equivalent Network
Rth=5.4545 Ohms
RL=15 Ohms
PS-Simulink Converter
0.1111
S PS
Current Sensor
f(x)=0
Solver Configuration
Electrical Reference
Norton's Current = 0.4167 A
Norton's Resistance = 5.4545Ohms
Current through Load Resistor 15 Ohms = 0.1111A
With all the sources in the network Current through Load Resistor 15 Ohms : 0.1111A
Hence Norton's Theorem is Verified.

MAXIMUM POWER TRANSFER THEOREM:

“In any circuit the maximum power is transferred to the load when the load resistance is equal to the source resistance. The source resistance is equal to the Thevenin’s equal resistance ”.

Procedure:

Step 1:

1. Make the connections as shown in the circuit diagram by using Multisim/MATLAB Simulink.
2. Measure the Power across the load resistor by considering all the sources in the network.

Step 2: Finding Thevenin’s Resistance(R_{TH})

1. Open the load terminals and replace all the sources with their internal impedances.
2. Measure the impedance across the open circuited terminal which is known as Thevenin’s Resistance.

Step 3: Finding Thevenin’s Voltage(V_{TH})

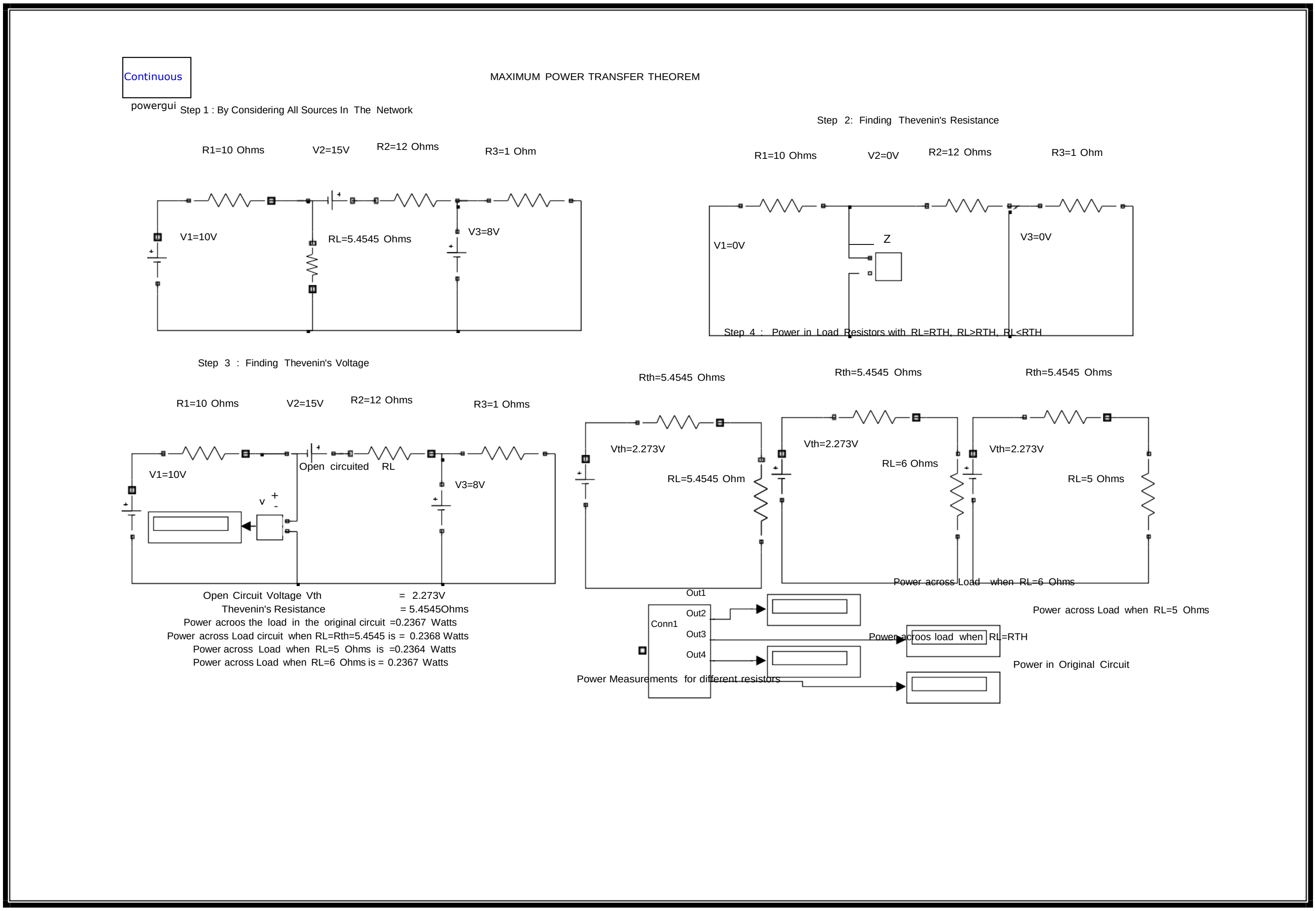
1. Open the load terminals and measure the voltage across the open circuited terminals.
2. Measured voltage will be known as Thevenin’s Voltage.

Step 4: Measuring Power for different Load Resistors

1. V_{TH} and R_{TH} are connected in series with the load.
2. Measure power across the load by considering $R_L=R_{TH}$.
3. Measure power by using $P = \frac{V_{TH}^2}{4R_L}$
4. Verify the power for different values of load resistors(i.e. $R_L>R_{TH}$ and $R_L<R_{TH}$)

Power measured from the above steps results in maximum power dissipation when $R_L=R_{TH}$.

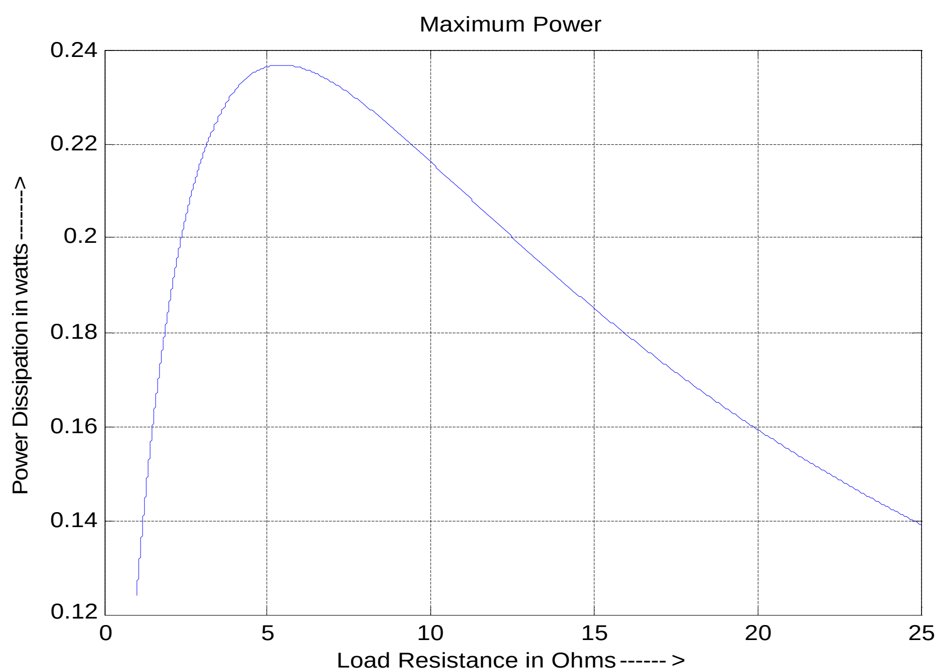
Hence Maximum Power Transfer Theorem is verified.



Continuous
powergui
MAXIMUM POWER TRANSFER THEOREM
Step 1 : By Considering All Sources In The Network
R1=10 Ohms
V2=15V
R2=12 Ohms
R3=1 Ohm
V1=10V
RL=5.4545 Ohms
V3=8V
Step 2: Finding Thevenin's Resistance
R1=10 Ohms
V2=0V
R2=12 Ohms
R3=1 Ohm
V1=0V
Z
V3=0V
Step 4 : Power in Load Resistors with RL=RTH, RL>RTH, RL<RTH
Step 3 : Finding Thevenin's Voltage
R1=10 Ohms
V2=15V
R2=12 Ohms
R3=1 Ohms
V1=10V
Open circuited RL
V3=8V
v + -
Rth=5.4545 Ohms
Rth=5.4545 Ohms
Rth=5.4545 Ohms
Vth=2.273V
RL=5.4545 Ohm
Vth=2.273V
RL=6 Ohms
Vth=2.273V
RL=5 Ohms
Power across Load when RL=6 Ohms
Open Circuit Voltage Vth = 2.273V
Thevenin's Resistance = 5.4545Ohms
Power acroos the load in the original circuit =0.2367 Watts
Power across Load circuit when RL=Rth=5.4545 is = 0.2368 Watts
Power across Load when RL=5 Ohms is =0.2364 Watts
Power across Load when RL=6 Ohms is = 0.2367 Watts
Out1
Out2
Conn1
Out3
Out4
Power Measurements for different resistors
Power across Load when RL=5 Ohms
Power acroos load when RL=RTH
Power in Original Circuit

M-File Program for Maximum Power Transfer Theorem:

```
clc;
close all;
clear all;

v=input('Enter the Voltage in Volts :');
rth=input('Enter the value of Thevenins Resistance:');
rl=1:0.0001:12;
i=v./(rth+rl);
p=i.^2.*rl;
plot(rl,p);
grid;
title('Maximum Power');
xlabel('Load Resistance in Ohms------->');
ylabel('Power Dissipation in watts-------->');
```



Results and Discussions: Super Position Theorem, Thevenin's Theorem, Norton's Theorem and Maximum Power Transfer Theorem are verified by using MATLAB Simulink /MULTISIM.

- The various circuit components are identified and circuits are formed in simulation environment.
- Use of network theorem in analysis can be demonstrated in this simulation exercise

EXPERIMENT NO-6

Locating Poles and Zeros in s-plane & z-plane

AIM: Write the program for locating poles and zeros and plotting pole-zero maps in s-plane and z-plane for the given transfer function.

Software Required: Matlab software

Theory:

Z-transforms

The Z-transform, like many other integral transforms, can be defined as either a *one-sided* or *two-sided* transform.

Bilateral Z-transform

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

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

Unilateral Z-transform

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

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

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

where $z = r.e^{j\omega}$

$$X(z) = \frac{P(z)}{Q(z)}$$

The roots of the equation P(z) = 0 correspond to the 'zeros' of X(z)
The roots of the equation Q(z) = 0 correspond to the 'poles' of X(z)

Example:

$$H(z) = \frac{z+1}{\left(z-\frac{1}{2}\right)\left(z+\frac{3}{4}\right)}$$

The zeros are: $\{-1\}$

The poles are: $\left\{ \frac{1}{2}, -\left(\frac{3}{4}\right) \right\}$

Program:
```
clc;
clear all;
close all;
%enter the numerator and denamenator cofficients in square brackets
num=input('enter numerator co-efficients');

den=input('enter denominator co-efficients');

% find poles and zeros
poles=roots(den)
zeros=roots(num)

% find transfer function H(s)
h=tf(num,den);

% plot the pole-zero map in s-plane
sgrid;
pzmap(h);
grid on;
title('locating poles and zeros on s-plane');

%plot the pole zero map in z-plane
figure
zplane(poles,zeros);
grid on;
title('locating poler and zeros on z-plane');
```

Result: Pole-zero maps are plotted in s-plane and z-plane for the given transfer function.

Output:

enter numerator co-efficients[1 -1 4 3.5]

enter denominator co-efficients[2 3 -2.5 6]

poles =

-2.4874

0.4937 + 0.9810i

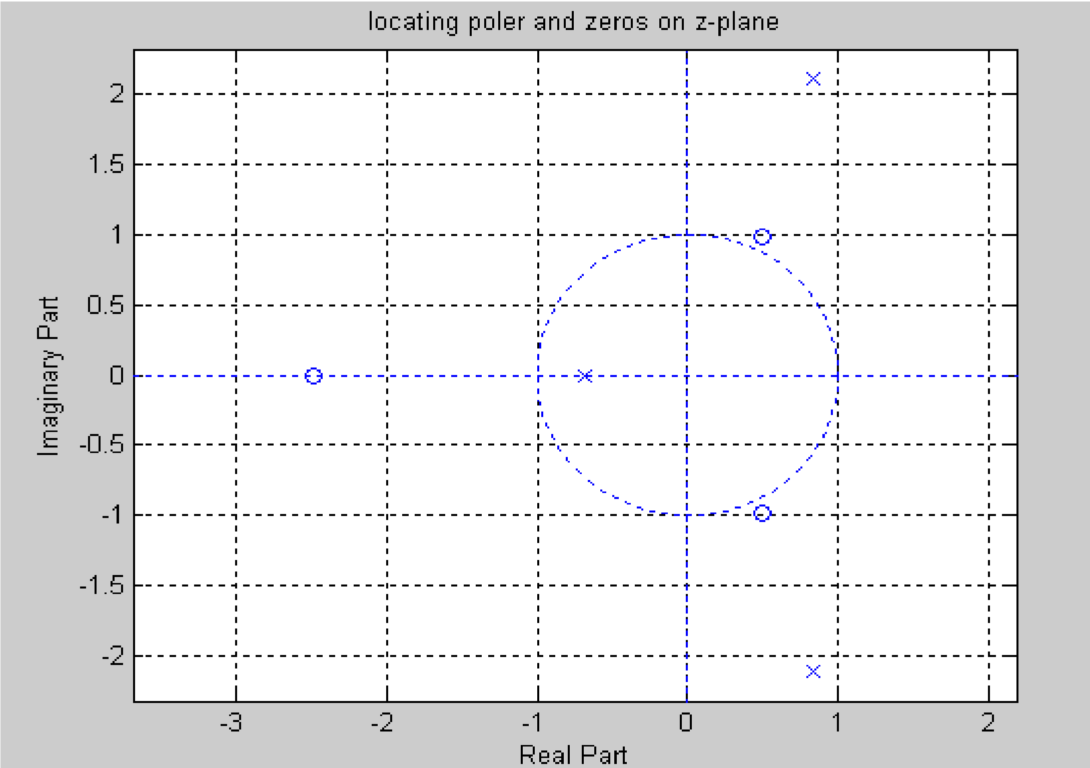
0.4937 - 0.9810i

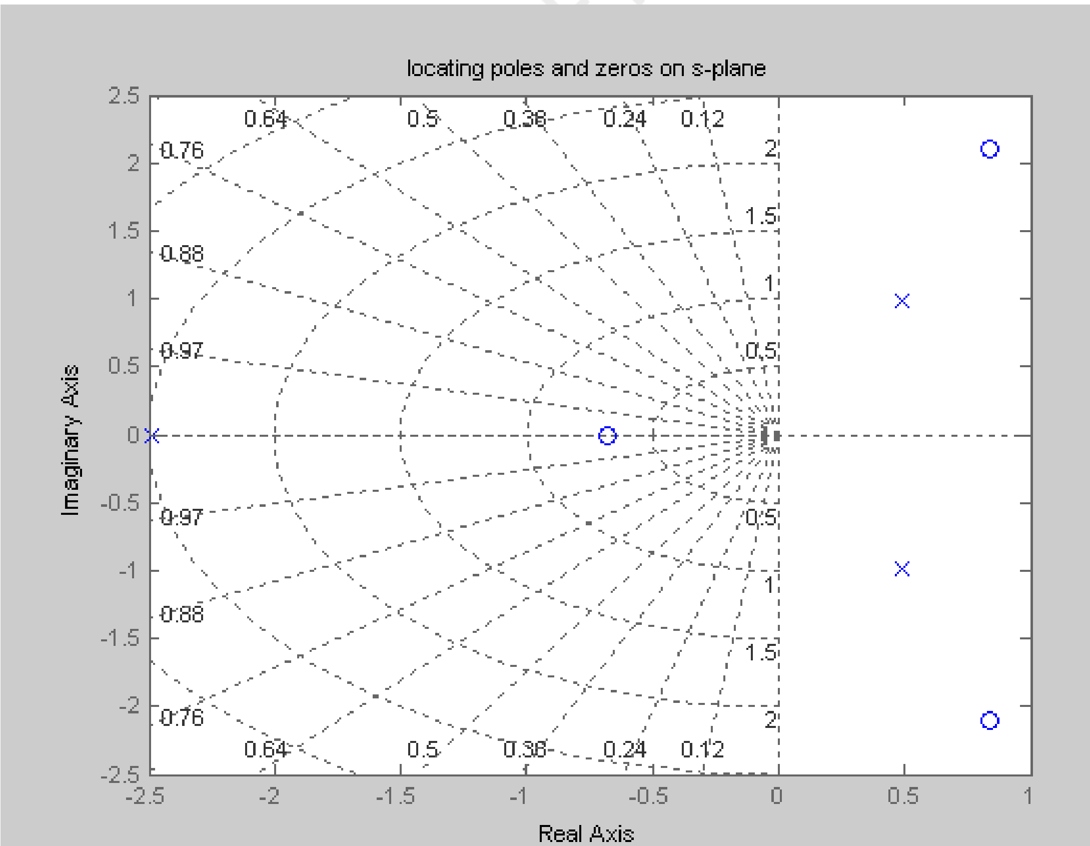
zeros =

0.8402 + 2.1065i

0.8402 - 2.1065i

-0.6805


locating poler and zeros on z-plane
Imaginary Part
Real Part
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-3
-2
-1
0
1
2


locating poles and zeros on s-plane
Imaginary Axis
Real Axis
0.64
0.5
0.38
0.24
0.12
0.76
0.88
0.97
2
1.5
1
0.5
2.5
-2.5
-2
-1.5
-1
-0.5
0
0.5
1