BASIC ELECTRICAL SIMULATION LAB

III B.Tech - I Sem.

M A N U A L

EEE

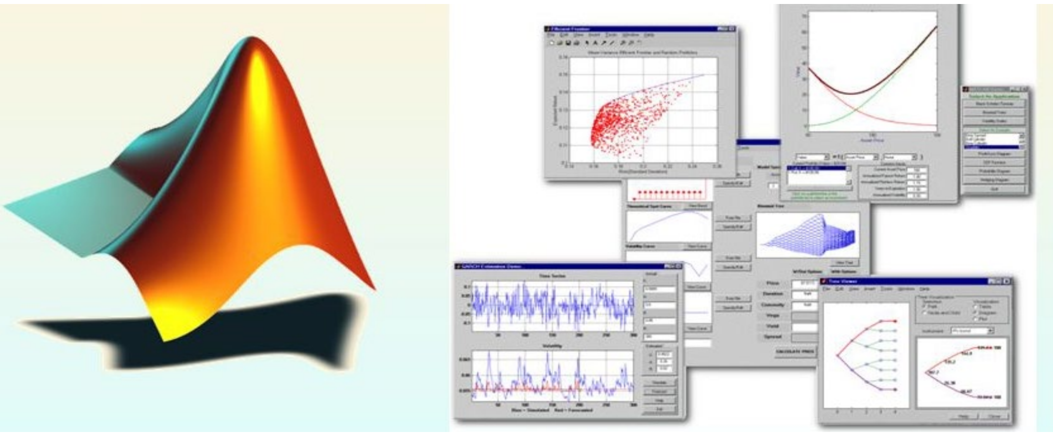
DEPARTMENT OF ELECRICAL & ELECTRONICS ENGINEERING

BALAJI INSTITUTE OF TECHNOLOGY AND SCIENCE

NARSAMPET, WARANGAL.

BITS
NARSAMPET

EEE



Prepared by

S. SRIDHAR

M.Tech(Power Electronics)

Asst. Professor

EEED

S. MALLIKARJUN REDDY

M.Tech(Power Engineering)

Head of the Department

EEED

Vision and Mission of EEE Department

Vision:

To nurture excellence in the field of Electrical & Electronics Engineering by imparting core values to the learners and to mould the institution into a centre of academic excellence and advanced research.

Mission:

M1: To impart students with high technical knowledge to make globally adept to the new Technologies

M2: To create, disseminate and integrate knowledge of engineering, science and technology that expands the electrical engineering knowledge base towards research

M3: To provide the students with a platform for developing new products and systems that can help industry and society as a whole.

Program Outcomes

PO1	**Engineering knowledge:** Apply the knowledge of basic sciences and fundamental engineering concepts in solving engineering problems.
PO 2	**Problem analysis:** Identify and define engineering problems, conduct experiments and investigate to analyze and interpret data to arrive at substantial conclusions.
PO 3	**Design/development of solutions:** Propose an appropriate solution for engineering problems complying with functional constraints such as economic, environmental, societal, ethical, safety and sustainability.
PO 4	**Conduct investigations of complex problems:** Perform investigations, design and conduct experiments, analyze and interpret the results to provide valid conclusions.
PO 5	**Modern tool usage:** Select/ develop and apply appropriate techniques and IT tools for the design and analysis of the systems.
PO 6	**The engineer and society:** Give reasoning and assess societal, health, legal and cultural issues with competency in professional engineering practice.
PO 7	**Environment and sustainability:** Demonstrate professional skills and contextual reasoning to assess environmental/ societal issues for sustainable development.
PO 8	**Ethics:** An ability to apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO 9	**Individual and team work:** Function effectively as an individual and as a member or leader in diverse teams and in multi-disciplinary situations.
PO 10	**Communication:** An ability to communicate effectively.
PO 11	**Project management and finance:** Demonstrate apply engineering and management principles in their own / team projects in multi-disciplinary environment.
PO 12	**Life-long learning:** An ability to do the needs of current technological trends at electrical industry by bridging the gap between academic and industry.

Program Specific Outcomes

PSO1	Apply fundamental knowledge to identify, analyze diverse problems associated with electrical and electronic circuits, power electronics drives and power systems.
PSO2	Understand the current technological developments in Electrical & Electronics Engineering and develop the innovative products/software to cater to the needs of society & Industry.

Program Educational Objectives

PEO1	To prepare students with solid foundation in Mathematics, Sciences and Basic Engineering to cover multi-disciplinary subjects enabling them to comprehend, analyze Electrical & Electronics Engineering problems and develop solutions.
PEO2	To design and develop an electrical system component or process to meet the needs of society and industry with in realistic constraints.
PEO3	To prepare students with technical competence to use advance techniques, skills and modern engineering tools that allow them to work effectively as electrical and electronics engineer.

ATTAINMENT OF PROGRAM OUTCOMES & PROGRAM SPECIFIC OUTCOMES

Exp .No	Name of the Experiment	Program Outcomes Attained	Program Specific Outcomes Attained
1	Basic Operations on Matrices	PO1, PO2, PO5	PSO2
2	Generation of various signals and sequences (Periodic and Aperiodic), such as unit Impulse, Step, Square, Saw tooth, Triangular, Sinusoidal, Ramp, Sinc.	PO1, PO2, PO5	PSO2
3	Operations on signals and sequences such as Addition, Multiplication, Scaling, Shifting, Folding, Computation of Energy, and Average Power	PO1, PO2, PO5	PSO2
4	Mesh and Nodal Analysis of Electrical circuits	PO1, PO2, PO5	PSO2
5	Application of Network Theorems to Electrical Networks	PO1, PO2, PO5	PSO2
6	Waveform Synthesis using Laplace Transform	PO1, PO2, PO5	PSO2
7	Locating the Zeros and Poles and Plotting the Pole-Zero maps in S plane and Z-Plane for the given transfer function	PO1, PO2, PO5	PSO2
8	Harmonic analysis of non sinusoidal waveforms	PO1, PO2, PO5	PSO2
	Simulation of DC Circuits	PO1, PO2, PO5	PSO2
9	Transient Analysis	PO1, PO2, PO5	PSO2
10	Measurement of active Power of three phase circuit for balanced and unbalanced load	PO1, PO2, PO5	PSO2
11	Simulation of single phase diode bridge rectifiers with filter for R & RL load	PO1, PO2, PO5	PSO2
12	Simulation of three phase diode bridge rectifiers with R, RL load	PO1, PO2, PO5	PSO2
13	Design of Low Pass and High Pass filters	PO1, PO2, PO5	PSO2
14	Finding the Even and Odd parts of Signal / Sequence and Real and imaginary parts of Signal	PO1, PO2, PO5	PSO2
15	Finding the Fourier Transform of a given signal and plotting its magnitude and phase spectrum	PO1, PO2, PO5	PSO2

PREFACE

This Laboratory book in Electrical Measurements has been revised in order to be up to date with Curriculum changes, laboratory equipment upgrading and the latest circuit simulation.

Every effort has been made to correct all the known errors, but nobody is perfect, if you find any additional errors or anything else you think is an error, please contact the HOD/EEE at mallik95_eee@yahoo.com

The Authors thanked all the staff members from the department for their valuable Suggestion and contribution.

The author would welcome the advice and suggestions leading to the improvement of the book.

The Authors,
Department of EEE.

DO’S and Don’t’s in Computer Lab

Do’s:

1. Remove your shoes or wear foot socks before you enter the lab.

2. Clean your computer with a soft, dry cloth.

3. Always keep quiet. Be considerate to other lab users.

4. Report any problems with the computer to the person in charge.

5. Shut down the computer properly.

Don’ts:

1. Do not bring any food or drinks in the computer room.

2. Do not touch any part of the computer with wet hands.

3. Do not hit the keys on the computer too hard.

4. Don’t damage, remove, or disconnect any labels, parts, cables or equipment.

5. Do not install or download any software or modify or delete any system files on any lab computers.

6. CD – ROM’s and other multimedia equipment are for school work only. Do not use them for playing music or other recreational activities.

7. If you leave the lab, do not leave your personal belongings unattended.

LABORATORY PRACTICE

SAFETY RULES

1. SAFETY is of paramount importance in the Electrical Engineering Laboratories.

2. Electricity NEVER EXECUSES careless persons. So, exercise enough care and attention in handling electrical equipment and follow safety practices in the laboratory. (Electricity is a good servant but a bad master).

3. Avoid direct contact with any voltage source and power line voltages. (Otherwise, any such contact may subject you to electrical shock)

4. Wear rubber-soled shoes. (To insulate you from earth so that even if you accidentally contact a live point, current will not flow through your body to earth and hence you will be protected from electrical shock)

5. Wear laboratory-coat and avoid loose clothing. (Loose clothing may get caught on an equipment/instrument and this may lead to an accident particularly if the equipment happens to be a rotating machine)

6. Girl students should have their hair tucked under their coat or have it in a knot.

7. Do not wear any metallic rings, bangles, bracelets, wristwatches and neck chains. (When you move your hand/body, such conducting items may create a short circuit or may touch a live point and thereby subject you to Electrical shock)

8. Be certain that your hands are dry and that you are not standing on wet floor. (Wet parts of the body reduce the contact resistance thereby increasing the severity of the shock)

9. Ensure that the power is OFF before you start connecting up the circuit. (Otherwise you will be touching the live parts in the circuit).

10. Get your circuit diagram approved by the staff member and connect up the circuit strictly as per the approved circuit diagram.

11. Check power chords for any sign of damage and be certain that the chords use safety plugs and do not defeat the safety feature of these plugs by using ungrounded plugs.

12. When using connection leads, check for any insulation damage in the leads and avoid such defective leads.

13. Do not defeat any safety devices such as fuse or circuit breaker by shorting across it. Safety devices protect YOU and your equipment.

14. Switch on the power to your circuit and equipment only after getting them checked up and approved by the staff member.

15. Take the measurement with one hand in your pocket. (To avoid shock in case you accidentally touch two points at different potentials with your two hands)

16. Do not make any change in the connection without the approval of the staff member.

17. In case you notice any abnormal condition in your circuit (like insulation heating up, resistor heating up etc), switch off the power to your circuit immediately and inform the staff member.

18. Keep hot soldering iron in the holder when not in use.

19. After completing the experiment show your readings to the staff member and switch off the power to your circuit after getting approval from the staff member.

20. Determine the correct rating of the fuse/s to be connected in the circuit after understanding correctly the type of the experiment to be performed: no-load test or full-load test, the maximum current expected in the circuit and accordingly use that fuse-rating.(While an over-rated fuse will damage the equipment and other instruments like ammeters and watt-meters in case of over load, an under-rated fuse may not allow one even to start the experiment)

21. Moving iron ammeters and current coils of wattmeters are not so delicate and hence these can stand short time overload due to high starting current. Moving iron meters are cheaper and more rugged compared to moving coil meters. Moving iron meters can be used for both a.c. and d.c. measurement. Moving coil instruments are however more sensitive and more accurate as compared to their moving iron counterparts and these can be used for d.c. measurements only. Good features of moving coil instruments are not of much consequence for you as other sources of errors in the experiments are many times more than those caused by these meters.

22. Some students have been found to damage meters by mishandling in the following ways:

i. Keeping unnecessary material like books, labrecords, unused meters etc. causing meters to fall down the table.

ii. Putting pressure on the meter (especially glass) while making connections or while talking or listening somebody.

STUDENTS ARE STRICTLY WARNED THAT FULL COST OF THE METER WILL BE RECOVERED FROM THE INDIVIDUAL WHO HAS DAMAGED IT IN SUCH A MANNER.

Copy these rules in your Lab Record. Observe these yourself and help your friends to observe.

I have read and understand these rules and procedures. I agree to abide by these rules and procedures at all times while using these facilities. I understand that failure to follow these rules and procedures will result in my immediate dismissal from the laboratory and additional disciplinary action may be taken.

Signature Date Lab

GUIDELINES FOR LABORATORY NOTEBOOK

The laboratory notebook is a record of all work pertaining to the experiment. This record should be sufficiently complete so that you or anyone else of similar technical background can duplicate the experiment and data by simply following your laboratory notebook. Record everything directly into the notebook during the experiment. Do not use scratch paper for recording data. Do not trust your memory to fill in the details at a later time.

Organization in your notebook is important. Descriptive headings should be used to separate and identify the various parts of the experiment. Record data in chronological order. A neat, organized and complete record of an experiment is just as important as the experimental work.

1. **Heading:**
The experiment identification (number) should be at the top of each page.Your name and date should be at the top of the first page of each day's experimental work.

2. **Object:**
A brief but complete statement of what you intend to find out or verify in the experiment should be at the beginning of each experiment

3. **Diagram:**
A circuit diagram should be drawn and labeled so that the actual experiment circuitry could be easily duplicated at any time in the future. Be especially careful to record all circuit changes made during the experiment.

4. **Equipment List:**
List those items of equipment which have a direct effect on the accuracy of the data. It may be necessary later to locate specific items of equipment for rechecks if discrepancies develop in the results.

5. **Procedure:**
In general, lengthy explanations of procedures are unnecessary. Be brief. Short commentaries alongside the corresponding data may be used. Keep in mind the fact that the experiment must be reproducible from the information given in your notebook.

6. **Data:**
Think carefully about what data is required and prepare suitable data tables. Record instrument readings directly. Do not use calculated results in place of direct data; however, calculated results may be recorded in the same table with the direct data. Data tables should be clearly identified and each data column labeled and headed by the proper units of measure.

7. **Calculations:**
Not always necessary but equations and sample calculations are often given to illustrate the treatment of the experimental data in obtaining the results.

8. **Graphs:**
Graphs are used to present large amounts of data in a concise visual form. Data to be presented in graphical form should be plotted in the laboratory so that any questionable data points can be checked while the experiment is still set up. The grid lines in the notebook can be used for most graphs. If special graph paper is required, affix the graph permanently into the notebook. Give all graphs a short descriptive title. Label and scale the axes. Use units of measure. Label each curve if more than one on a graph.

9. **Results:**

The results should be presented in a form which makes the interpretation easy. Large amounts of numerical results are generally presented in graphical form. Tables are generally used for small amounts of results. Theoretical and experimental results should be on the same graph or arrange in the same table in a way for easy correlation of these results.

10. **Conclusion:**

This is your interpretation of the results of the experiment as an engineer. Be brief and specific. Give reasons for important discrepancies.

INSTRUCTIONS TO THE STUDENT FJHINSTRUCTIONS TO TE STUDENT

FJHFHINSTRUCTIONS TO THE STUDENT

1. Students are required to attend all labs.
2. Students will work individually in hardware laboratories and in computer laboratories.
3. While coming to the lab bring the lab manual cum observation book, record etc.
4. Take only the lab manual, calculator (if needed) and a pen or pencil to the work area.
5. Before coming to the lab, prepare the prelab questions. Read through the lab experiment to familiarize yourself with the components and assembly sequence.
6. Utilize 3 hours time properly to perform the experiment (both in software and hardware) and note down the readings properly. Do the calculations, draw the graph and take signature from the instructor.
7. If the experiment is not completed in the prescribed time, the pending work has to be done in the leisure hour or extended hours.
8. You have to submit the completed record book according to the deadlines set up by your instructor.
9. For practical subjects there shall be a continuous evaluation during the semester for 25 sessional marks and 50 end examination marks.
10. Of the 25 marks for internal, 15 marks shall be awarded for day-to-day work and 10 marks to be awarded by conducting an internal laboratory test.

Sl.No	Experiment
1	Basic Operations on Matrices.
2	Generation of Various Signals and Sequences (Periodic and Aperiodic), such as Unit impulse, unit step, square, saw tooth, triangular, sinusoidal, ramp, sinc.
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 to electrical networks
6	Waveform synthesis using laplace transforms
7	Locating zeroes and poles and ploting the pole-zero maps in S plane and for the given TF
8	Harmonic analysis of non sinusoidal waveforms
9	Simulation of DC circuits
10	Transient analysis
11	Measurement of Active power of three phase circuit for balanced and unbalanced loads
12	Simulation of single phase diode bridge rectifiers with filter for R and RL loads
13	Simulation of Three phase diode bridge rectifiers with filter for R and RL loads
14	Design of LOW pass and High pass filter
15	Finding the even and odd parts of signal / sequence and real and imaginary parts of signal.
16	Finding the Fourier transform of a given signal and plotting its magnitude and Phase spectrum.

EXPERMENT NO:1

BASIC OPERATIONS ON MATRICES

AIM: -

To write a MATLAB program to perform some basic operation on matrices
such as addition, subtraction, multiplication.

SOFTWARE REQURIED:-

1.MATLAB R2010a.
2.Windows XP SP2.

THEORY:-

MATLAB, which stands for MATrix LABoratory, is a state-of-the-art mathematical software package, which is used extensively in both academia and industry. It is an interactive program for numerical computation and data visualization, which along with its programming capabilities provides a very useful tool for almost all areas of science and engineering. Unlike other mathematical packages, such as MAPLE or MATHEMATICA, MATLAB cannot perform symbolic manipulations without the use of additional Toolboxes. It remains however, one of the leading software packages for numerical computation. As you might guess from its name, MATLAB deals mainly with matrices. A scalar is a 1-by-1 matrix and a row vector of length say 5, is a 1-by-5 matrix.. One of the many advantages of MATLAB is the natural notation used. It looks a lot like the notation that you encounter in a linear algebra. This makes the use of the program especially easy and it is what makes MATLAB a natural choice for numerical computations. The purpose of this experiment is to familiarize MATLAB, by introducing the basic features and commands of the program.

Built in Functions:

Scalar Functions:

Certain MATLAB functions are essentially used on scalars, but operate element-wise when applied to a matrix (or vector). They are summarized below.

1. sin - trigonometric sine

2. cos - trigonometric cosine
3. tan - trigonometric tangent
4. asin - trigonometric inverse sine (arcsine)
5. acos - trigonometric inverse cosine (arccosine)
6. atan - trigonometric inverse tangent (arctangent)
7. exp - exponential
8. log - natural logarithm
9. abs - absolute value
10. sqrt - square root
11. rem - remainder
12. round - round towards nearest integer
13. floor - round towards negative infinity
14. ceil - round towards positive infinity

2. Vector Functions:

Other MATLAB functions operate essentially on vectors returning a scalar value. Some of these functions are given below.

1. max largest component : get the row in which the maximum element lies
2. min smallest component
3. length length of a vector
4. sort sort in ascending order
5. sum sum of elements
6. prod product of elements
7. median median value
8. mean mean value std standard deviation

3. Matrix Functions:

Much of MATLAB‟ s power comes from its matrix functions. These can be further separated into two sub-categories.

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 - randomly generated matrix

```
eg: diag([0.9092;0.5163;0.2661])
ans =
0.9092 0 0
0 0.5163 0
0 0 0.2661
```

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 polynomial

PROCEDURE:-

2. Open MATLAB
3. Open new M-file
4. Type the program
5. Save in current directory
6. Compile and Run the program
7. For the output see command window\ Figure window

PROGRAM:-

```
clc;
close all;
clear all;
a=[1 2 -9 ; 2 -1 2; 3 -4 3];
b=[1 2 3; 4 5 6; 7 8 9];
disp('The matrix a= ');
```

```
a
disp('The matrix b= ');
b
% to find sum of a and b
c=a+b;
disp('The sum of a and b is ');
c
% to find difference of a and b
d=a-b;
disp('The difference of a and b is ');
d
%to find multiplication of a and b
e=a*b;
disp('The product of a and b is ');
e
```

OUTPUT:

```
The matrix a=
a =
1 2 -9
2 -1 2
3 -4 3

The matrix b=
b =
1 2 3
4 5 6
7 8 9

The sum of a and b is
c =
2 4 -6
```

```
6 4 8
10 4 12

The difference of a and b is
d =
0 0 -12
-2 -6 -4
-4 -12 -6

The product of a and b is
e =
-54 -60 -66
12 15 18
8 10 12
```

RESULT:-

Finding addition, subtraction, multiplication using MATLAB was Successfully completed.

VIVA QUESTIONS:-

1.Expand MATLAB?And importance of MATLAB?

2.What is clear all and close all will do?

3.What is disp() and input()?

4.What is the syntax to find the eigen values and eigenvectors of the matrix?

5. What is the syntax to find the rank of the matrix?

EXPERMENT NO:2
GENERATION OF VARIOUS SIGNALS&SEQUENCES

AIM:- To write a “MATLAB” Program to generate various signals and sequences,such as unit impulse, unit step, unit ramp, sinusoidal,square,saw tooth,triangular,sinc signals.

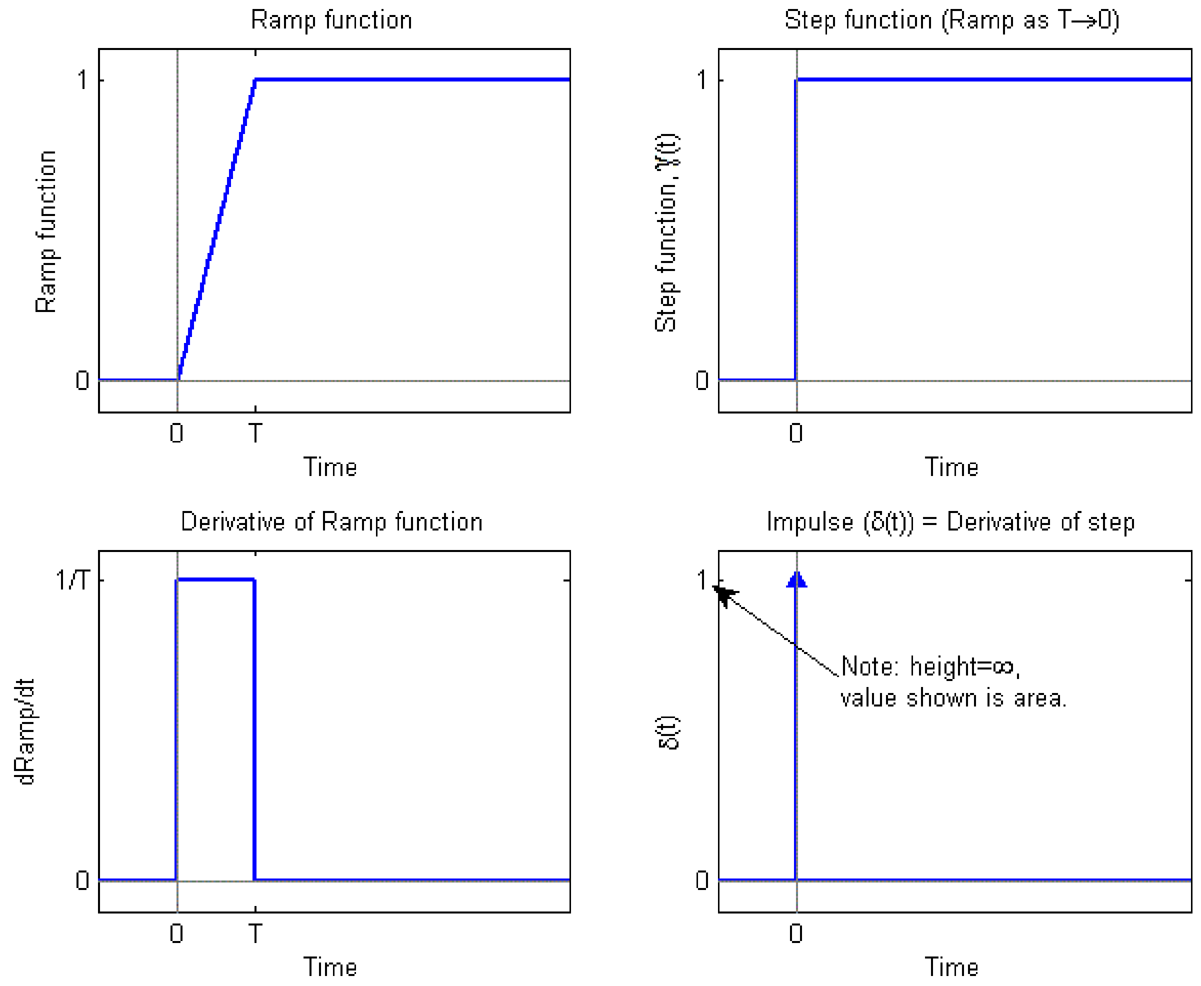
SOFTWARE REQURIED:-

1.MATLAB R2010a.

2.Windows XP SP2.

THEORY:-

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 is infinitely high. However, the *area* of the impulse is finite. This is, at first hard to visualize but we can do so by using the graphs shown below.



Key Concept: Sifting Property of the Impulse

If b>a, then

$$\int_a^b \delta(t-T)\cdot f(t)dt = \begin{cases} f(T), & a < T < b \\ 0, & \text{otherwise} \end{cases}$$

Example: Another integral problem

Assume a<b, and evaluate the integral

$$\int_a^b \delta(t)\cdot f(t-T)dt$$

Solution:

Solution:

We now that the impulse is zero except at t=0 so

$$\delta(t)\cdot f(t-T) = \delta(t)\cdot f(0-T) = \delta(t)\cdot f(-T)$$

And

$$\int_a^b \delta(t)\cdot f(t-T)dt = \int_a^b \delta(t)\cdot f(-T)dt$$

$$= f(-T)\cdot\int_a^b \delta(t)dt$$

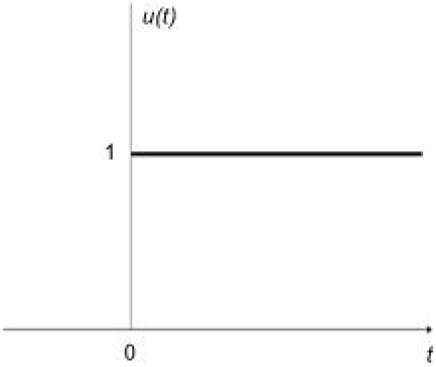
$$= \begin{cases} f(-T), & a < 0 < b \\ 0, & \text{otherwise} \end{cases}$$

Unit Step Function

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.

The unit step function, also known as the Heaviside function, is defined as such:

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



Sometimes, u(0) is given other values, usually either 0 or 1. For many applications, it is irrelevant what the value at zero is. u(0) is generally written as undefined.

Derivative

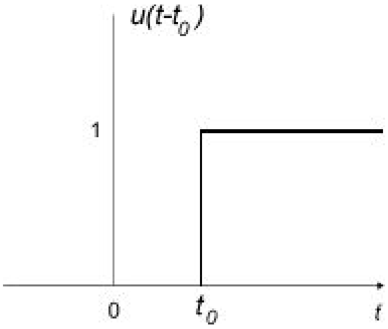
The unit step function is level in all places except for a discontinuity at t = 0. For this reason, the derivative of the unit step function is 0 at all points t, except where t = 0. Where t = 0, the derivative of the unit step function is infinite.

The derivative of a unit step function is called an impulse function. The impulse function will be described in more detail next.

Integral

The integral of a unit step function is computed as such:

$$\int_{-\infty}^{t} u(s)ds = \begin{Bmatrix} 0, & \text{if } t < 0 \\ \int_0^t ds = t, & \text{if } t \geq 0 \end{Bmatrix} = tu(t)$$



Sinc Function

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 fu

The Sinc function is defined in the following manner:

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x} \text{ if } x \neq 0$$

And

Sinc(0)=1

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

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

This fact can be proven by noting that for x near 0,

$$1 > \frac{\sin(x)}{x} > \cos(x)$$

Then, since

Then, since cos(0) = 1, we can apply the approaches one as x goes to zero. Thus, defining s continuous.

Also, the Sinc function approaches zero as x goes towards infinity, with the envelope of sinc(x) tapering off as 1/x.

Rect Function

The Rect Function is a function which produces a rectangular

centered at t = 0. The Rect function pulse also has a height of 1. The Sinc function and the rectangular function form a Fourier transform pair.

A Rect function can be written in the form:

$$\operatorname{rect}\left(\frac{t-X}{Y}\right)$$

where the pulse is centered at X and has width Y. We can define the impulse function above in terms of the rectangle function by centering the pulse at zero (X = 0), setting it's height to 1/A and setting the pulse width to A, which approaches zero:

$$\delta(t) = \lim_{A\to 0} \frac{1}{A} \operatorname{rect}\left(\frac{t-0}{A}\right)$$

We can also construct a Rect function out of a pair of unit step functions

$$\operatorname{rect}\left(\frac{t-X}{Y}\right) = u(t-X+Y/2) - u(t-X-Y/2)$$

Here, both unit step functions are set a distance of Y/2 away from the center point of (t - X).

SAW TOOTH:-

The sawtooth wave (or saw wave) is a kind of non-sinusoidal waveform. It is named a sawtooth based on its resemblance to the teeth on the blade of a saw. The convention is that a sawtooth wave ramps upward and then sharply drops. However, there are also sawtooth waves in which the wave ramps downward and then sharply rises. The latter type of sawtooth wave is called a 'reverse sawtooth wave' or 'inverse sawtooth wave'. As audio signals, the two orientations of sawtooth wave sound identical. The piecewise linear function based on the floor function of time t, is an example of a sawtooth wave with period 1.

$$x(t) = 2\left(\frac{t}{a} - \operatorname{floor}\left(\frac{t}{a}+\frac{1}{2}\right)\right)$$

Triangle wave

A triangle wave is a non-sinusoidal waveform named for its triangular shape.A bandlimited triangle wave pictured in the time domain (top) and frequency domain (bottom). The fundamental is at 220 Hz (A2).Like a square wave, the triangle wave contains only odd harmonics. However, the higher harmonics roll off much faster than in a square wave (proportional to the inverse square of the harmonic number as opposed to just the inverse).It is possible to approximate a triangle wave with additive synthesis by adding odd harmonics of the fundamental, multiplying every (4n□1)th harmonic by □1 (or changing its phase by □), and rolling off the harmonics by the inverse square of their relative frequency to the fundamental.This infinite Fourier series converges to the triangle wave:

$$\begin{aligned} x_{\text{triangle}}(t) &= \frac{8}{\pi^2}\sum_{k=0}^{\infty}(-1)^k \frac{\sin((2k+1)\omega t)}{(2k+1)^2} \\ &= \frac{8}{\pi^2}\left(\sin(\omega t) - \frac{1}{9}\sin(3\omega t) + \frac{1}{25}\sin(5\omega t) - \cdots\right) \end{aligned}$$

where ω is the angular frequency.

Sinusoidal Signal Generation

The sine wave or sinusoid is a mathematical function that describes a smooth repetitive oscillation. It occurs often in pure mathematics, as well as physics, signal processing, electrical engineering and many other fields. Its most basic form as a function of time (t) is:

where:

- A, the amplitude, is the peak deviation of the function from its center position.
- □, the angular frequency, specifies how many oscillations occur in a unit time interval, in radians per second
- □, the phase, specifies where in its cycle the oscillation begins at t = 0.

A sampled sinusoid may be written as:

$$x(n) = A\sin(2\pi \frac{f}{f_s} n + \vartheta)$$

where f is the signal frequency, fs is the sampling frequency, □ is the phase and A is the amplitude of the signal.

PROCEDURE:-

1. Open MATLAB
2. Open new M-file
3. Type the program
4. Save in current directory

5. Compile and Run the program
6. For the output see command window\ Figure window

PROGRAM:-

```
%unit impulse function%
clc;
clear all;
close all;
t=-10:1:10;
x=(t==0);
subplot(2,1,1);
plot(t,x,'g');
xlabel('time');
ylabel('amplitude');
title('unit impulse function');
subplot(2,1,2);
stem(t,x,'r');
xlabel('time');
ylabel('amplitude');
title('unit impulse discreat function');
%unit step function%
clc;
clear all;
close all;
N=100;
t=1:100;
x=ones(1,N);
subplot(2,1,1);
plot(t,x,'g');
xlabel('time');
ylabel('amplitude');
title('unit step function');
subplot(2,1,2);
```

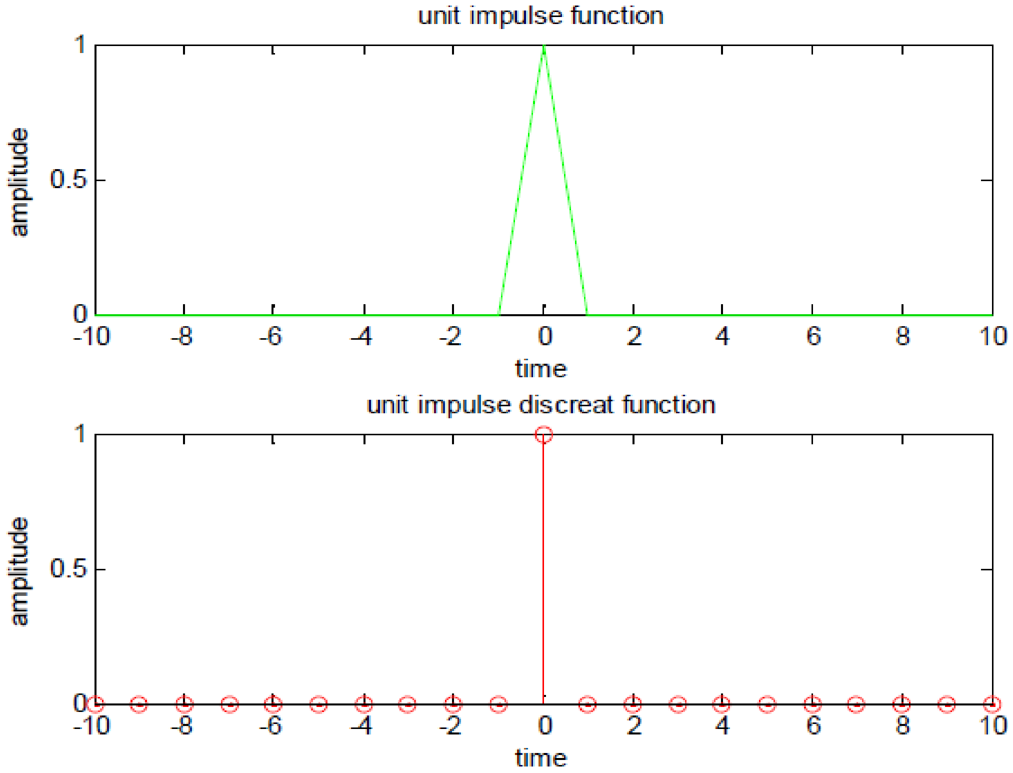
```
stem(t,x,'r');
xlabel('time');
ylabel('amplitude');
title('unit step discreat function');
%unit ramp function%
%unit ramp function%
clc;
clear all;
close all;
t=0:20;
x=t;
subplot(2,1,1);
plot(t,x,'g');
xlabel('time');
ylabel('amplitude');
title('unit ramp function');
subplot(2,1,2);
stem(t,x,'r');
xlabel('time');
ylabel('amplitude');
title('unit ramp discreat function');
%sinusoidal function%
clc;
clear all;
close all;
t=0:0.01:2;
x=sin(2*pi*t);
subplot(2,1,1);
plot(t,x,'g');
xlabel('time');
ylabel('amplitude');
title('sinusoidal signal');
subplot(2,1,2);
```

```
stem(t,x,'r');
xlabel('time');
ylabel('amplitude');
title('sinusoidal sequence');
%square function%
clc;
clear all;
close all;
t=0:0.01:2;
x=square(2*pi*t);
subplot(2,1,1);
plot(t,x,'g');
xlabel('time');
ylabel('amplitude');
title('square signal');
subplot(2,1,2);
stem(t,x,'r');
xlabel('time');
ylabel('amplitude');
title('square sequence');
%sawtooth function%
clc;
clear all;
close all;
t=0:0.01:2;
x=sawtooth(2*pi*5*t);
subplot(2,1,1);
plot(t,x,'g');
xlabel('time');
ylabel('amplitude');
title('sawtooth signal');
subplot(2,1,2);
stem(t,x,'r');
```

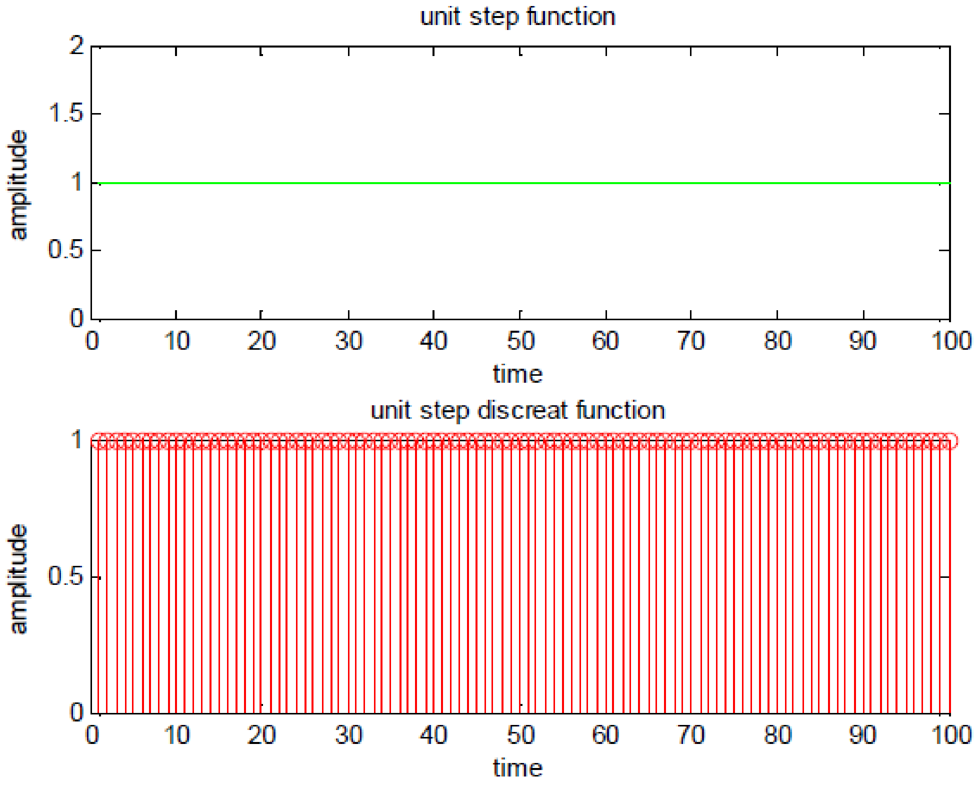
```
xlabel('time');
ylabel('amplitude');
title('sawtooth sequence');
%trianguler function%
clc;
clear all;
close all;
t=0:0.01:2;
x=sawtooth(2*pi*5*t,0.5);
subplot(2,1,1);
plot(t,x,'g');
xlabel('time');
ylabel('amplitude');
title('trianguler signal');
subplot(2,1,2);
stem(t,x,'r');
xlabel('time');
ylabel('amplitude');
title('trianguler sequence');
%sinc function%
clc;
clear all;
close all;
t=linspace(-5,5);
x=sinc(t);
subplot(2,1,1);
plot(t,x,'g');
xlabel('time');
ylabel('amplitude');
title('sinc signal');
subplot(2,1,2);
stem(t,x,'r');
xlabel('time');
```

```
ylabel('amplitude');
title('sinc sequence');
```

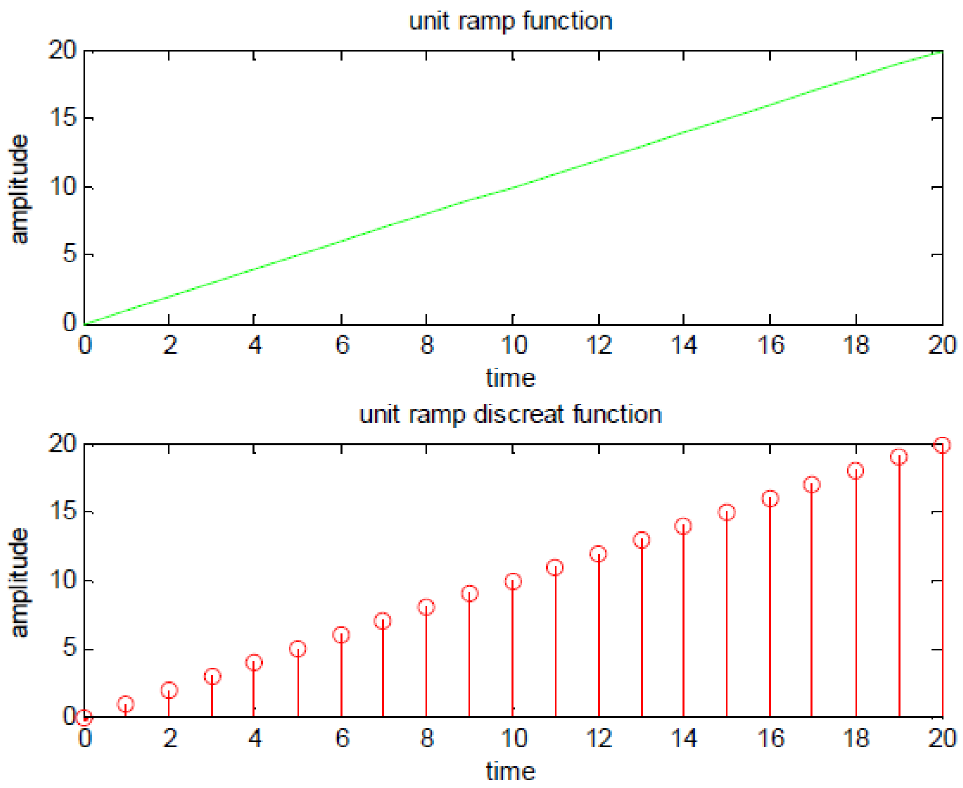
unit impulse function



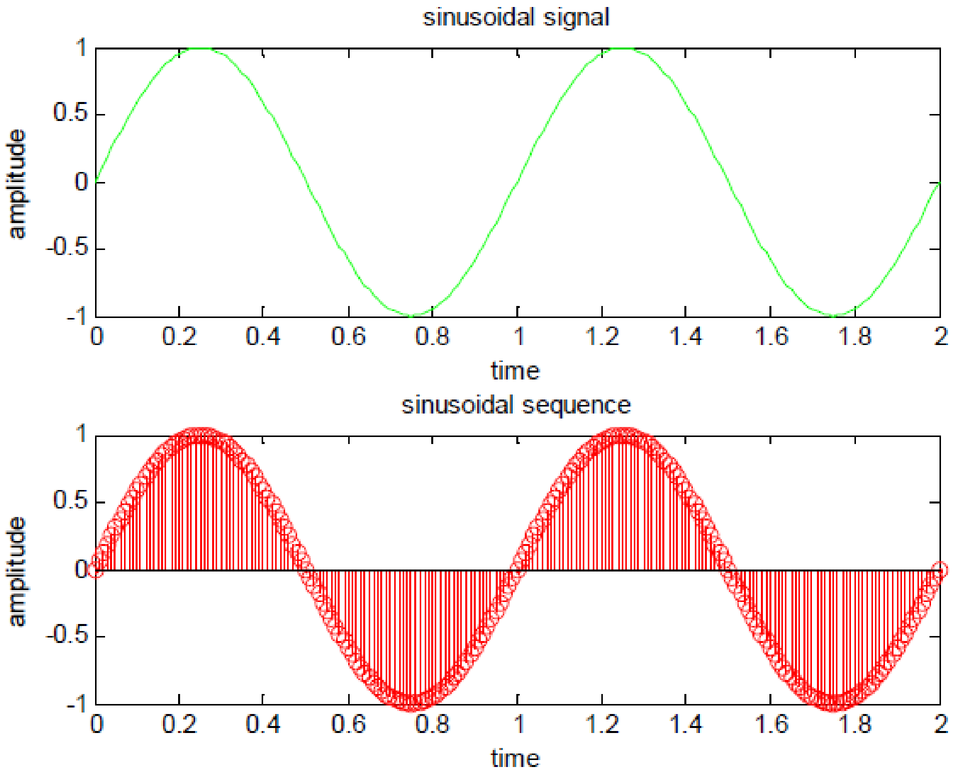
unit step function



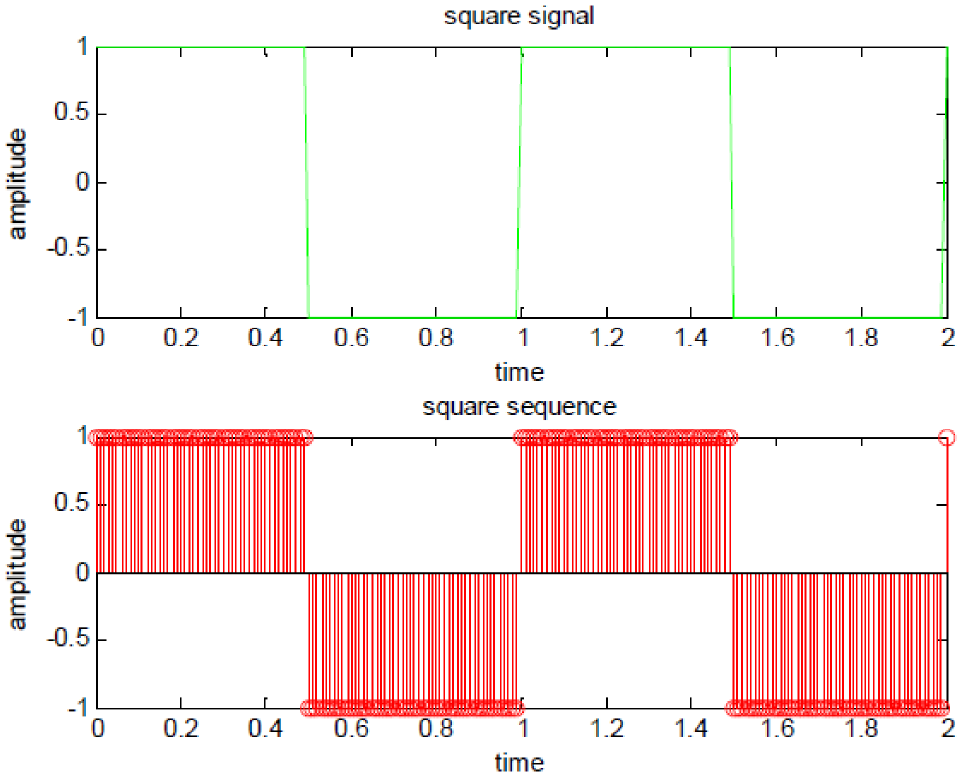
unit ramp function



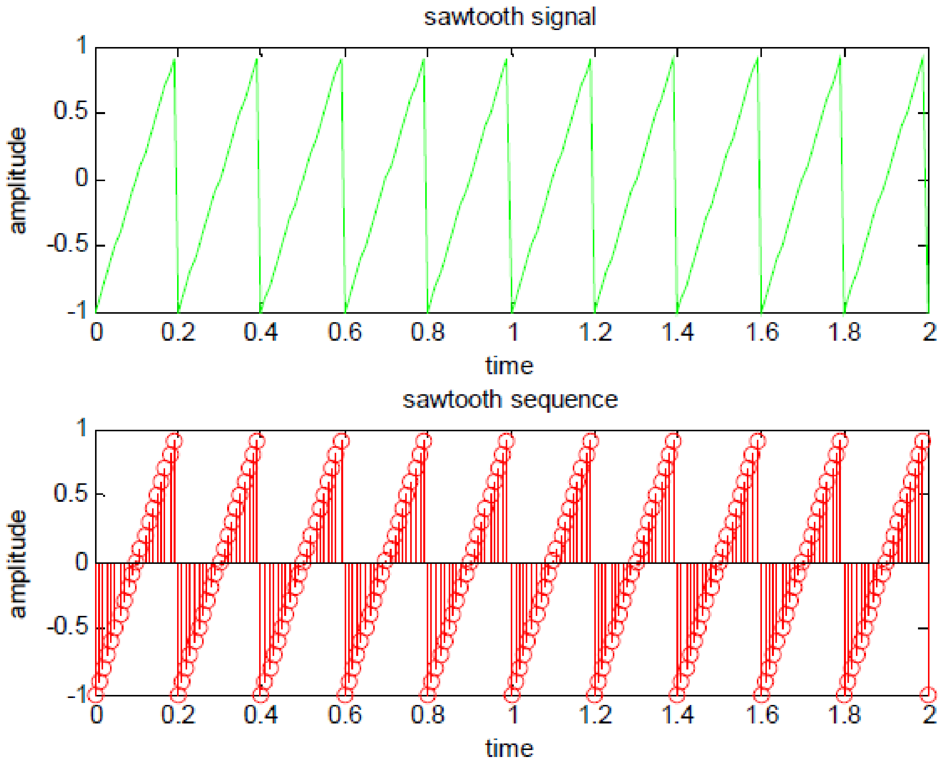
sinusoidal function



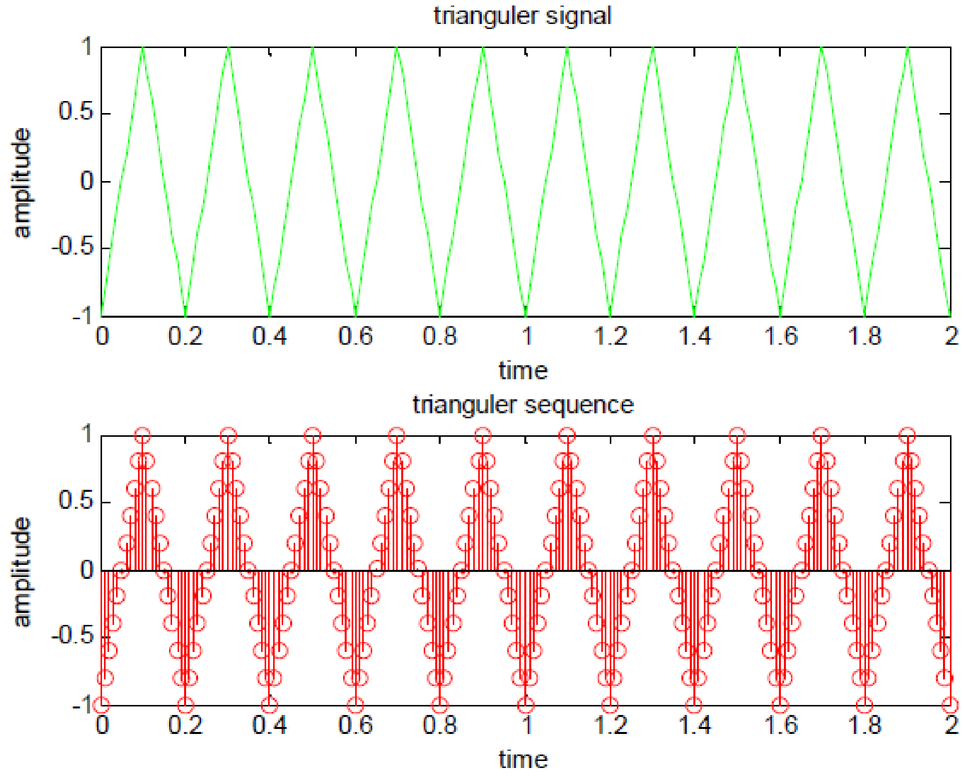
square function



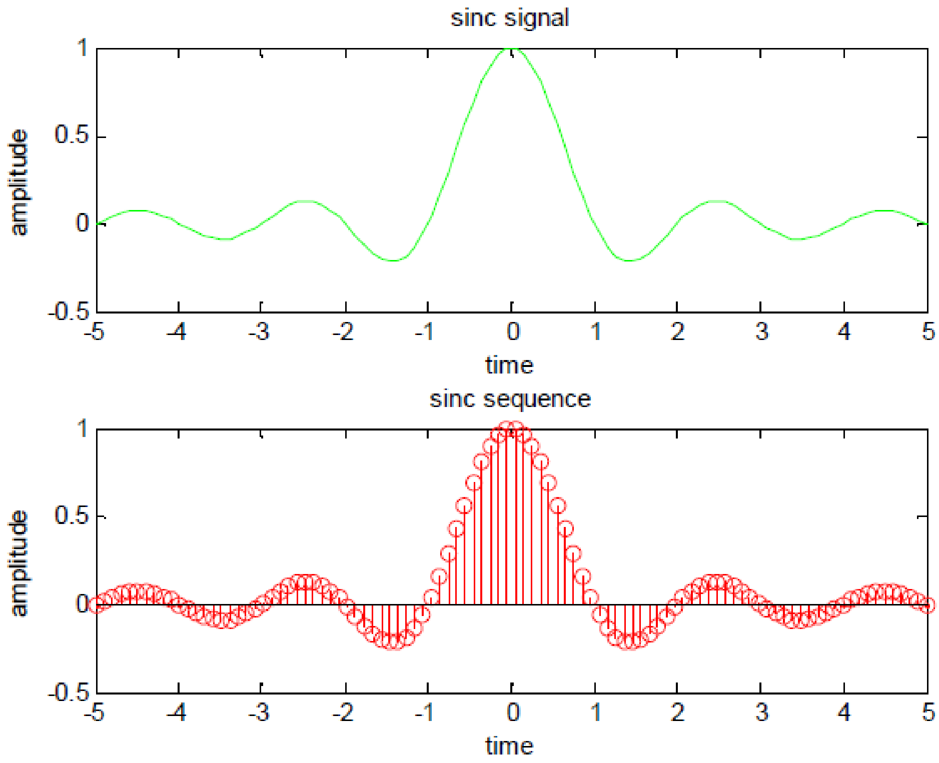
sawtooth function



trianguler function



sinc function



.RESULT:-

Thus the Generation of continuous time signals like unit step, sawtooth, triangular, sinusoidal, ramp and sinc functions are successfully completed by using MATLAB.

VIVA QUESTIONS:-

1. Define Signal?
2. Define determistic and Random Signal?
3. Define Delta Function?
4. What is Signal Modeling?
5. Define Periodic and a periodic Signal?

EXPERMENT NO:3

OPERATION ON SIGNALS&SEQUENCES

AIM:-

To performs operations on signals and sequences such as addition, multiplication, scaling, shifting, folding, computation of energy and average power.

SOFTWARE REQURIED:-

1.MATLAB R2010a.

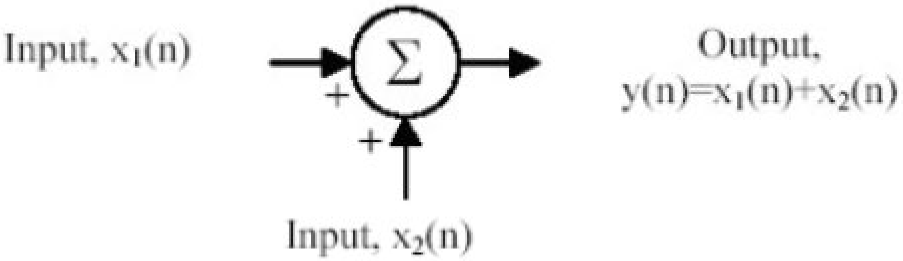
2.Windows XP SP2.

THEORY:-

Basic Operation on Signals:

Time shifting: y(t)=x(t-T)The effect that a time shift has on the appearance of a signal If T is a positive number, the time shifted signal, x (t -T) gets shifted to the right, otherwise it gets shifted left.

Signal Shifting and Delay:



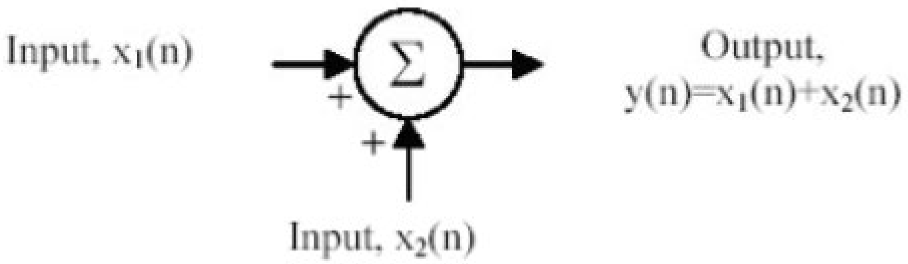
Shifting : y(n)={x(n-k)} ; m=n-k; y=x;

Time reversal: Y(t)=y(-t) Time reversal _ips the signal about t = 0 as seen in Figure 1.

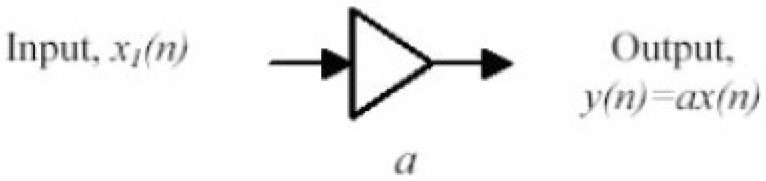
Signal Addition and Substraction :

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

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



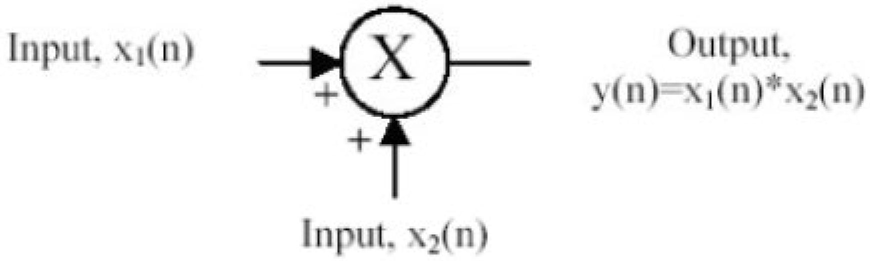
Signal Amplification/Attuation :



Multiplication/Divition :

of two signals, their product is also a signal.

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



folding:

y(n)={x(-n)} ; y=fliplr(x); n=-fliplr(n);

PROCEDURE:-

1. Open MATLAB
2. Open new M-file
3. Type the program
4. Save in current directory
5. Compile and Run the program
6. For the output see command window\ Figure window

PROGRAM:-

```
%Addition and multiplication of two signals%
clc;
```

```
clear all;
close all;
t=0:0.001:2;
s1=6*sin(2*pi*5*t);
subplot(4,1,1);
plot(t,s1,'g');
xlabel('time');
ylabel('amplitude');
title('first signal');
s2=8*sin(2*pi*5*t);
subplot(4,1,2);
plot(t,s2,'r');
xlabel('time');
ylabel('amplitude');
title('second signal');
s3=s1+s2;
subplot(4,1,3);
plot(t,s3,'g');
xlabel('time');
ylabel('amplitude');
title('sum of two signals');
s4=s1.*s2;
subplot(4,1,4);
plot(t,s4,'g');
xlabel('time');
ylabel('amplitude');
title('multiplication of two signals');
%Amplitude scaling for signals%
clc;
clear all;
close all;
t=0:0.001:2;
s1=6*sin(2*pi*5*t);
```

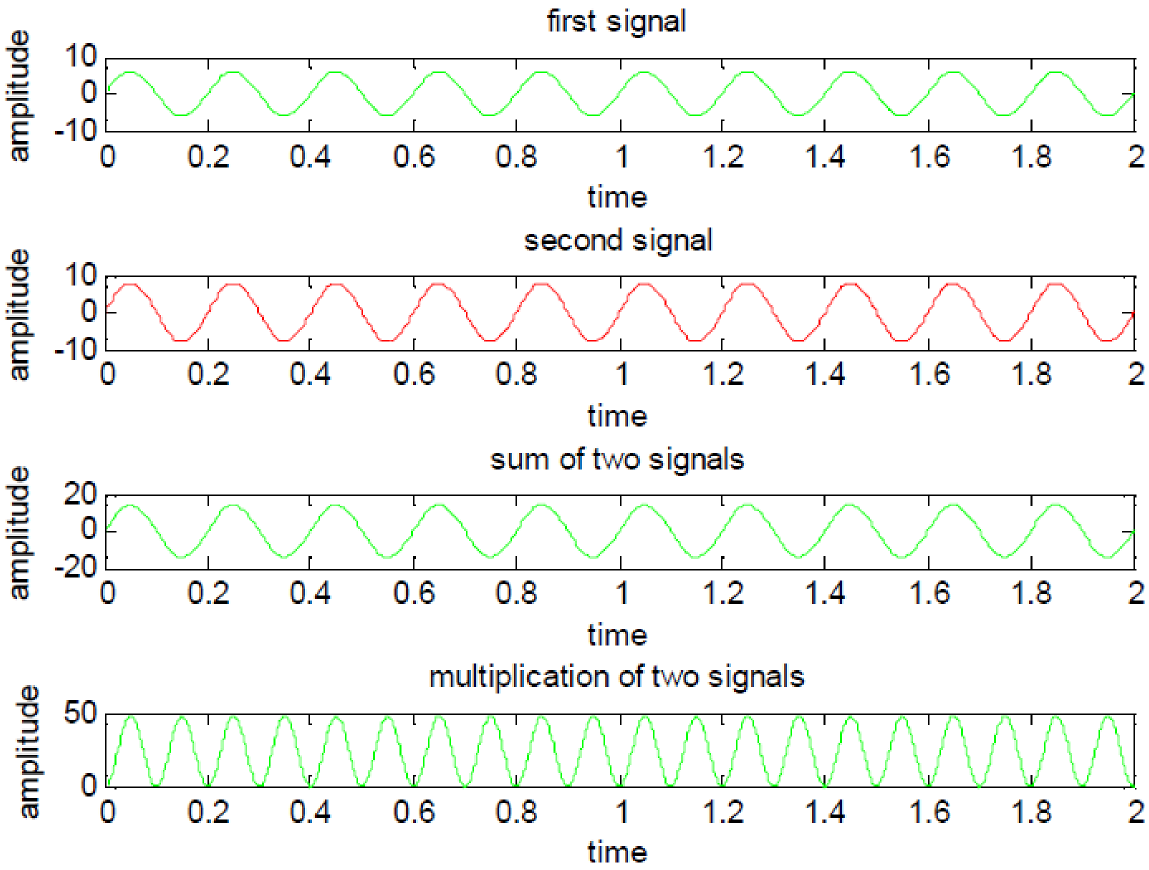
```
subplot(3,1,1);
plot(t,s1,'g');
xlabel('time');
ylabel('amplitude');
title('sinusoidal signal');
s2=3*s1;
subplot(3,1,2);
plot(t,s2,'r');
xlabel('time');
ylabel('amplitude');
title('amplified signal');
s3=s1/3;
subplot(3,1,3);
plot(t,s3,'g');
xlabel('time');
ylabel('amplitude');
title('attenuated signal');
%Time scaling for signals%
clc;
clear all;
close all;
t=0:0.001:2;
s1=6*sin(2*pi*5*t);
subplot(3,1,1);
plot(t,s1,'g');
xlabel('time');
ylabel('amplitude');
title('sinusoidal signal');
t1=3*t;
subplot(3,1,2);
plot(t1,s1,'r');
xlabel('time');
ylabel('amplitude');
```

```
title('compressed signal');
t2=t/3;
subplot(3,1,3);
plot(t2,s1,'g');
xlabel('time');
ylabel('amplitude');
title('enlarged signal');
%Time shifting of a signal%
clc;
clear all;
close all;
t=0:0.001:3;
s1=6*sin(2*pi*5*t);
subplot(3,1,1);
plot(t,s1,'g');
xlabel('time');
ylabel('amplitude');
title('sinusoidal signal');
t1=t+10;
subplot(3,1,2);
plot(t1,s1,'r');
xlabel('time');
ylabel('amplitude');
title('right shift of the signal');
t2=t-10;
subplot(3,1,3);
plot(t2,s1,'g');
xlabel('time');
ylabel('amplitude');
title('left shift of the signal');
%Time folding of a signal%
clc;
clear all;
```

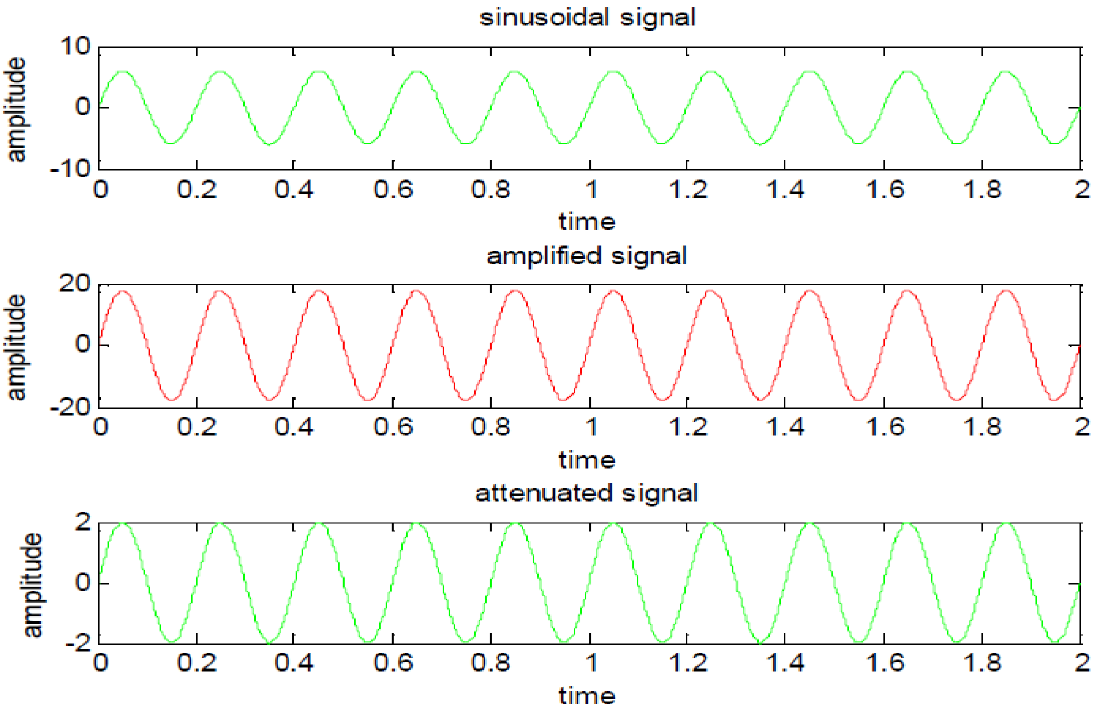
```
close all;
t=0:0.001:2;
s=sin(2*pi*5*t);
m=length(s);
n=[-m:m];
y=[0,zeros(1,m),s];
subplot(2,1,1);
plot(n,y,'g');
xlabel('time');
ylabel('amplitude');
title('original signal');
y1=[fliplr(s),0,zeros(1,m)];
subplot(2,1,2);
plot(n,y1,'r');
xlabel('time');
ylabel('amplitude');
title('folded signal');
```

OUTPUT:-

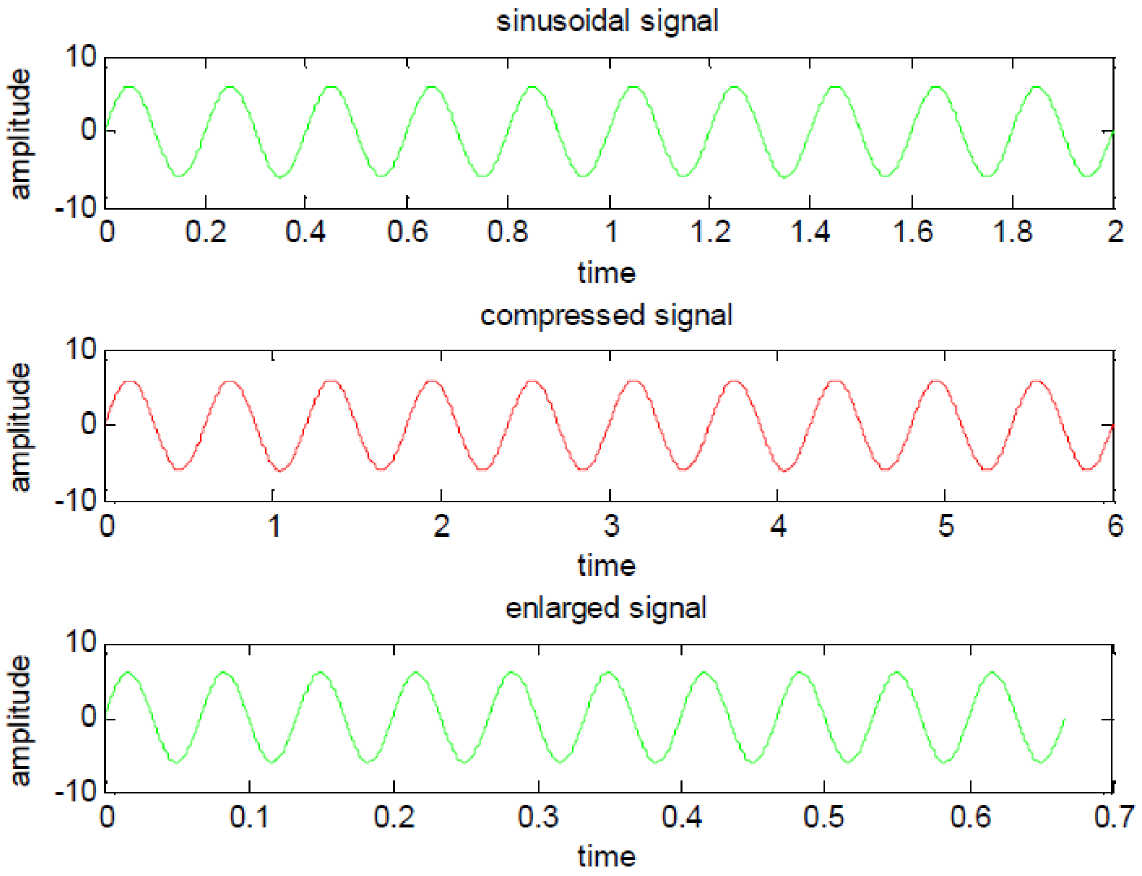
Addition and multiplication of two signals



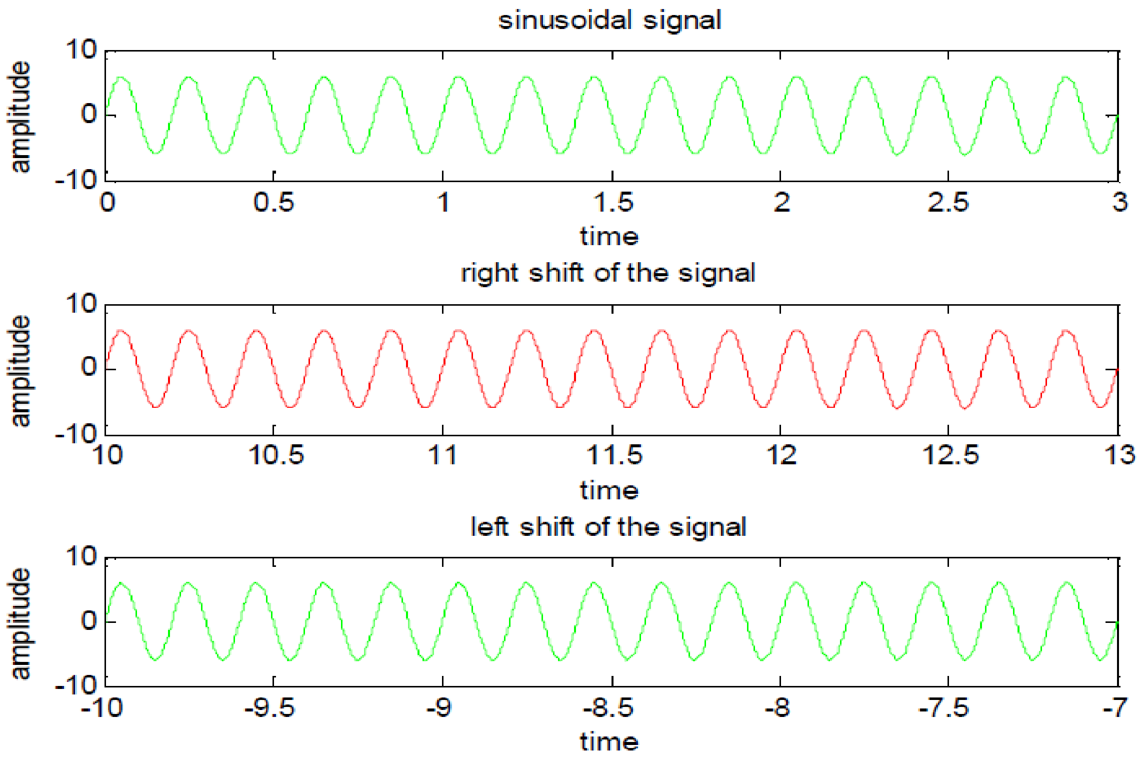
Amplitude scaling for signals



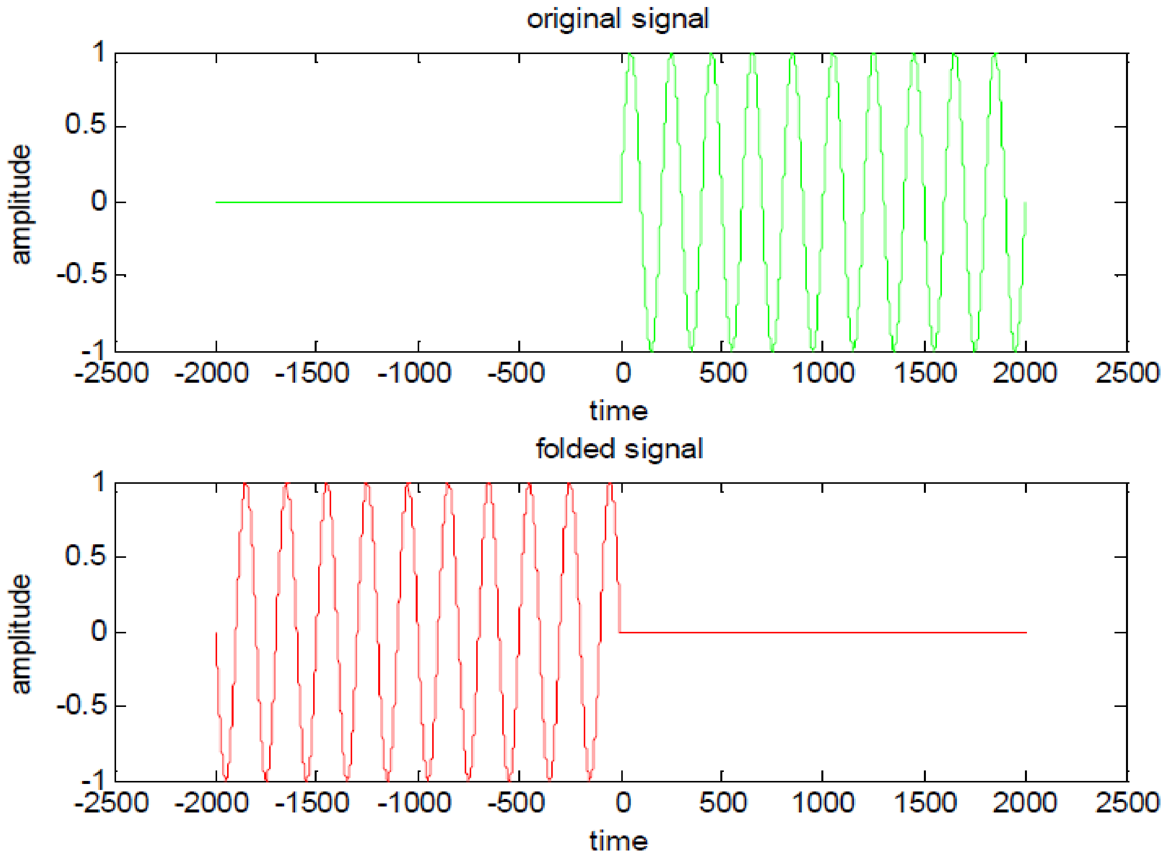
Time scaling for signals



Time shifting of a signal



Time folding of a signal



RESULT:-

In this experiment the various oprations on signals have been
Performed Using MATLAB have been demonstrated.

VIVA QUESTIONS:-

1. Define Symetric and Anti-Symmetric Signals?
2. Define Continuous and Discrete Time Signals?
3. What are the Different types of representation of discrete time signals?
4. What are the Different types of Operation performed on signals?
5. What is System?

EXPERMENT NO:15

FINDING EVEN AND ODD &REAL AND IMAGINARY PARTS OF SEQUENCES

AIM: -

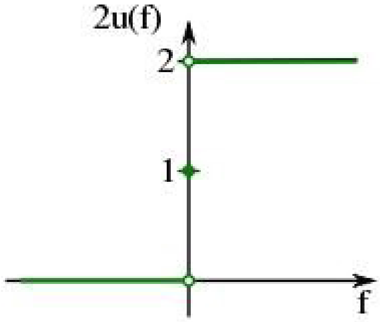
program for finding even and odd parts of sequences Using MATLAB Software&
program for finding real and imaginary parts of sequences Using MATLAB Software
SOFTWARE REQURIED:-
1.MATLAB R2010a.
2.Windows XP SP2.
THEORY:-
Even and Odd Signal
One of characteristics of signal is symmetry that may be useful for signal analysis. Even signals are symmetric around vertical axis, and Odd signals are symmetric about origin.
Even Signal: A signal is referred to as an even if it is identical to its time-reversed counterparts; x(t) = x(-t).
Odd Signal: A signal is odd if x(t) = -x(-t).
An odd signal must be 0 at t=0, in other words, odd signal passes the origin.
Using the definition of even and odd signal, any signal may be decomposed into a sum of its even part, xe(t), and its odd part, xo(t), as follows:
x(t)=xe(t)+xo(t);
x(t)=1/2{x(t)+x(-t)} +1/2{x(t)-x(-t)}
where
xe(t)=1/2{x(t)+x(-t)} &xo(t)=1/2{x(t)-x(-t)}
It is an important fact because it is relative concept of Fourier series. In Fourier series, a periodic signal can be broken into a sum of sine and cosine signals. Notice that sine function is odd signal and cosine function is even signal□
ENERGY AND POWER SIGNAL:
A signal can be categorized into energy signal or power
signal: An energy signal has a finite energy, 0 < E < □. In other words, energy signals have values only in the limited time duration. For example, a signal having only one square pulse is energy signal. A signal that decays exponentially has finite energy, so, it is also an energy signal. The power of an energy signal is 0, because of dividing finite

energy by infinite time (or length).

If *x*(*t*) is a real-valued signal with Fourier transform *X*(*f*), and u(*f*) is the Heaviside step function, then the function**:**

$$X_a(f) \overset{\text{def}}{=} \begin{cases} 2X(f), & \text{for } f > 0, \\ X(f), & \text{for } f = 0, \\ 0, & \text{for } f < 0, \end{cases}$$

$$= X(f) \cdot \underbrace{2\mathrm{u}(f)}_{1+\mathrm{sgn}(f)} = X(f) + X(f) \cdot \mathrm{sgn}(f)$$



contains only the non-negative frequency components of *X*(*f*). And the operation is reversible, due to the Hermitian property of *X*(*f*)**:**

$$X(f) = \begin{cases} \frac{1}{2}X_a(f), & \text{for } f > 0, \\ X_a(f) & \text{for} f = 0, \\ \frac{1}{2}X_a(-f)^*, & \text{for } f < 0 \end{cases}$$

$$= \frac{1}{2}\left(X_a(f) + X_a(-f)^*\right) \ .$$

X(*f*)* denotes the complex conjugate of *X*(*f*) .

The inverse Fourier transform of *X*a(*f*) is the **analytic signal:**

$$\begin{aligned} x_a(t) &= \mathcal{F}^{-1}\{X(f) + X(f)\cdot \mathrm{sgn}(f)\} \\ &= \mathcal{F}^{-1}\{X(f)\} + \underbrace{\mathcal{F}^{-1}\{X(f)\} * \mathcal{F}^{-1}\{\mathrm{sgn}(f)\}}_{convolution} \\ &= x(t) + j\underbrace{\left[x(t) * \frac{1}{\pi t}\right]}_{\hat{x}(t)}, \end{aligned}$$

where x^(t) is the Hilbert transform of x(t) and J is the imaginary unit.

PROCEDURE:-

1. Open MATLAB

2. Open new M-file
3. Type the program
4. Save in current directory
5. Compile and Run the program
6. For the output see command window\ Figure window

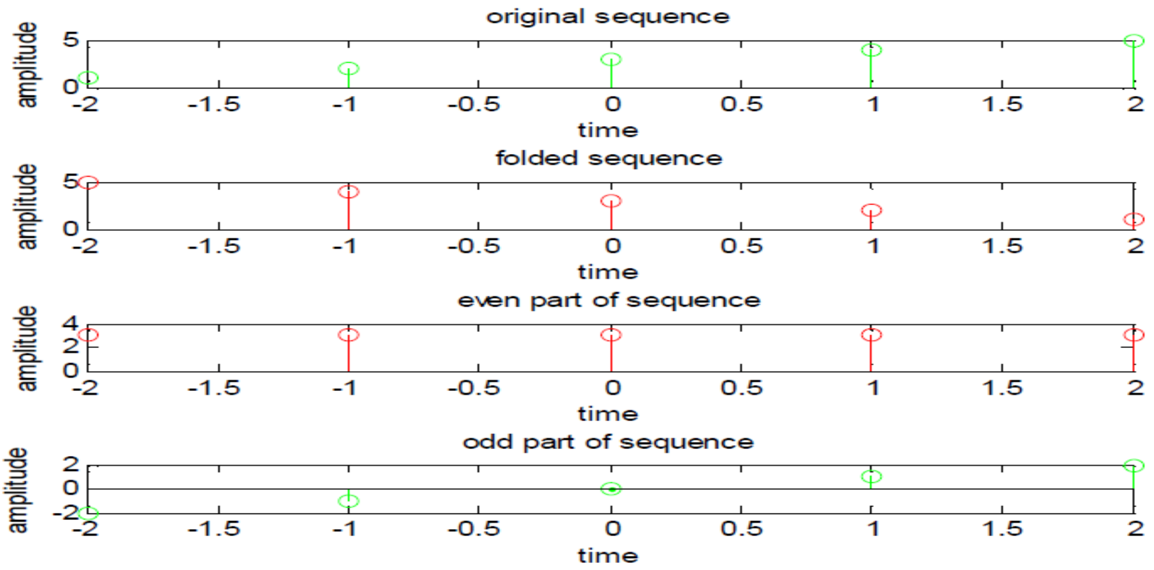
PROGRAM:-

```
%Even,odd,real,imaginary parts of a sequences%
clc;
clear all;
close all;
h=input('enter no.of samples');
m=(h-1)/2;
n=-m:m;
x=input('enter sample values');
subplot(4,1,1);
stem(n,x,'g');
xlabel('time');
ylabel('amplitude');
title('original sequence');
xmir=fliplr(x);
subplot(4,1,2);
stem(n,xmir,'r');
xlabel('time');
ylabel('amplitude');
title('folded sequence');
%even part of sequence%
xeven=(x+xmir)/2;
subplot(4,1,3);
stem(n,xeven,'r');
xlabel('time');
ylabel('amplitude');
title('even part of sequence');
%odd part of sequence%
```

```
xodd=(x-xmir)/2;
subplot(4,1,4);
stem(n,xodd,'g');
xlabel('time');
ylabel('amplitude');
title('odd part of sequence');
%Real&Imaginary parts of a sequences%
clc;
clear all;
close all;
y=input('enter complex numbers');
yreal=real(y);
disp('real values of y');
yreal
yimag=imag(y);
disp('imaginary values of y');
yimag
```

OUTPUT:-

```
Even,odd,real,imaginary parts of a sequences
enter no.of samples5
enter sample values[1 2 3 4 5]
```



```
Real&Imaginary parts of a sequences
enter complex numbers[1+6i 2-5i 3 4+3i 5i]
real values of y
yreal =
1 2 3 4 0
imaginary values of y
yimag =
6 -5 0 3 5
```

RESULT:-

In this experiment even and odd parts of various signals and energy and power of signals have been calculated Using MATLAB.

VIVA QUESTIONS:-

1. 1. What is the formula to find odd part of signal?
2. 2. What is Even Signal?
3. 3. What is Odd Signal?
4. 4. What is the formula to find even part of signal?
5. 5.What is the difference b/w stem&plot?

EXPERMENT NO:16

FINDING THE FOURIER TRANSFORM OF A GIVEN SIGNAL AND PLOTTING ITS MAGNITUDE AND PHASE SPECTRUM

AIM: -

To obtain Fourier Transform and Inverse Fourier Transform of a given signal / sequence and to plot its Magnitude and Phase Spectra.

SOFTWARE REQURIED:-

1.MATLAB R2010a.

2.Windows XP SP2.

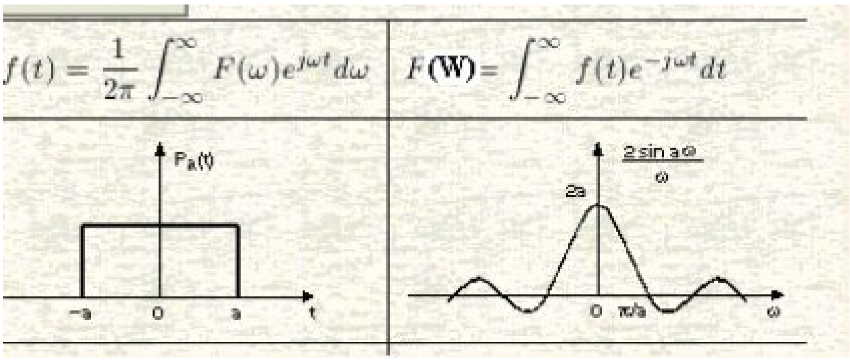
THEORY:-

Fourier Transform Theorems:

We may use Fourier series to motivate the Fourier transform as follows. Suppose that *f* is a function which is zero outside of some interval [□L/2, L/2]. Then for any T ≥ L we may expand *f* in a Fourier series on the interval [□T/2,T/2], where the "amount" of the wave e2πinx/T in the Fourier series of *f* is given by

By definition

$$F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-i\omega x}\, dx$$



$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega) e^{j\omega t}\, d\omega \qquad F(\mathbf{W}) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t}\, dt$$

The fast Fourier transform (FFT) is an efficient algorithm for computing the DFT of a sequence; it is not a separate transform. It is particularly useful in areas such as signal and image processing, where its uses range from filtering, convolution, and frequency analysis to power spectrum estimation

For length N input vector x, the DFT is a length N vector X, with elements N

X(k) = sum x(n)*exp(-j*2*pi*(k-1)*(n-1)/N), 1 <= k <= N.

n=1

The inverse DFT (computed by IFFT) is given by N

x(n) = (1/N) sum X(k)*exp(j*2*pi*(k-1)*(n-1)/N), 1 <= n <= N.

k=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

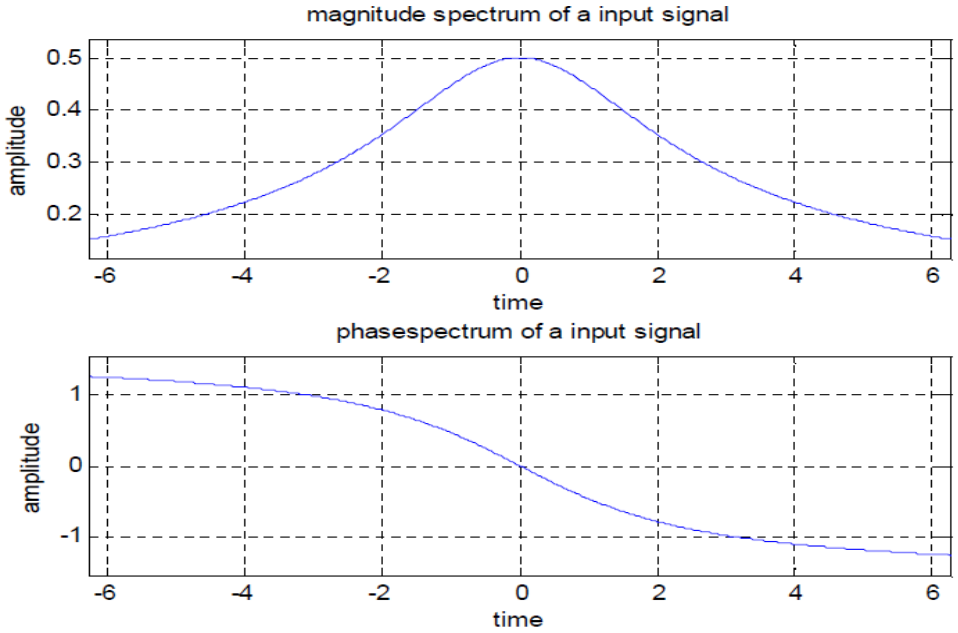
PROGRAM:-

```
%Fourier Transform%
clc
clear all;
close all;
syms t;
x=exp(-2*t)*heaviside(t);
y=fourier(x);
disp('Fourier Transform of input signal');
y
z=ifourier(y);
disp('Inverse Fourier Transform of input signal');
z
mg=abs(y);
subplot(2,1,1);
ezplot(mg);
xlabel('time');
```

```
ylabel('amplitude');
title('magnitude spectrum of a input signal');
grid;
pha=atan(imag(y)/real(y));
subplot(2,1,2);
ezplot(pha);
xlabel('time');
ylabel('amplitude');
title('phasespectrum of a input signal');
grid;
```

OUTPUT:-

```
Fourier Transform
Fourier Transform of input signal
Y=
1/(2+w*i)
Inverse Fourier Transform of input signal
Z=
Heaviside(x)/exp(2*x)
```



RESULT:-

In this experiment the fourier transform of a given signal and plotting its magnitude and phase spectrum have been demonstrated using MATLAB.

VIVA QUESTIONS:-

1.Define Fourier Series?

2.What are the properties of Continuous-Time Fourier Series?

3. What is the Sufficient condition for the existence of F.T?

4. Define the F.T of a signal?

5. What is the difference b/w F.T&F.S?

EXERCISE PROGRAMS

1.Write a MATLAB program to find the cross correlation using FFT.

EXPERMENT NO:7

LOCATING THE ZEROS AND POLES AND PLOTTING THE POLE ZERO MAPS IN S-PLANE AND Z-PLANE FOR THE GIVEN TRANSFER FUNCTION.

AIM: -

To locating the zeros and poles and plotting the pole zero maps in s-plane and z-plane for the given transfer function.

SOFTWARE REQURIED:-

1.MATLAB R2010a.

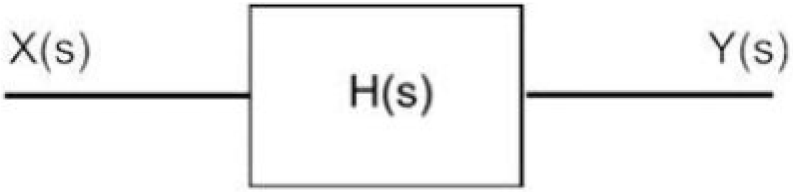
2.Windows XP SP2.

THEORY:-

A Transfer Function is the ratio of the output of a system to the input of a system, in the Laplace domain considering its initial conditions to be zero. If we have an input function of X(s), and an output function Y(s), we define the transfer function H(s) to be:

$$H(s) = \frac{Y(s)}{X(s)}$$

transfer function is the Laplace transform of a system‟ s impulse response.



Given a continuous-time transfer function in the Laplace domain, *H*(*s*) or a discrete-time one in the Z-domain, *H*(*z*), a zero is any value of *s* or *z* such that the transfer function is zero, and a pole is any value of *s* or *z* such that the transfer function is infinite.

Zeros**:**1. The value(s) for *z* where the *numerator* of the transfer function equals zero

2. The complex frequencies that make the overall gain of the filter transfer function zero.

Poles: 1. The value(s) for *z* where the *denominator* of the transfer function equals zero

2. The complex frequencies that make the overall gain of the filter transfer function infinite.

Z-transforms

the Z-transform converts a discrete time-domain signal, which is a sequence of real or complex numbers, into a complex frequency-domain representation.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 ≥ 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

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

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

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)

PROCEDURE:-

1. Open MATLAB
2. Open new M-file
3. Type the program

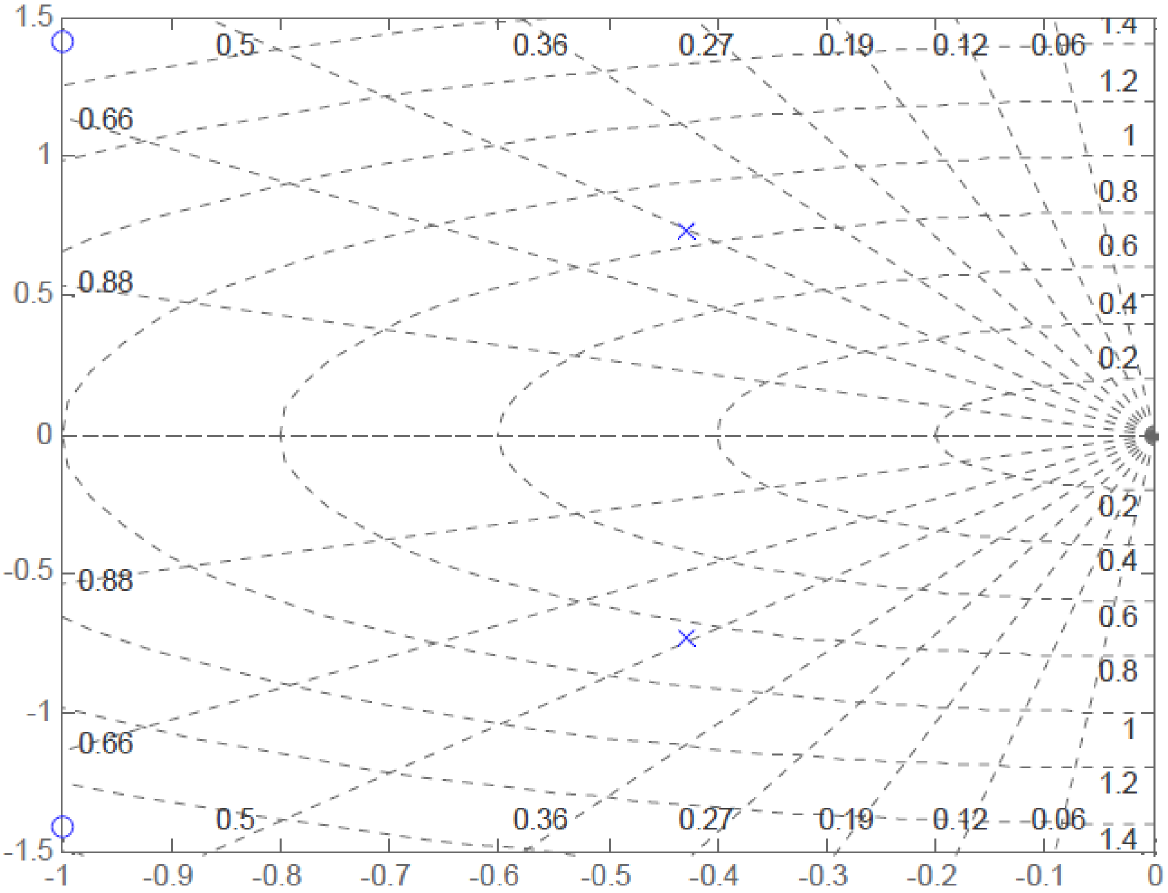
4. Save in current directory
5. Compile and Run the program
6. For the output see command window\ Figure window

PROGRAM:-

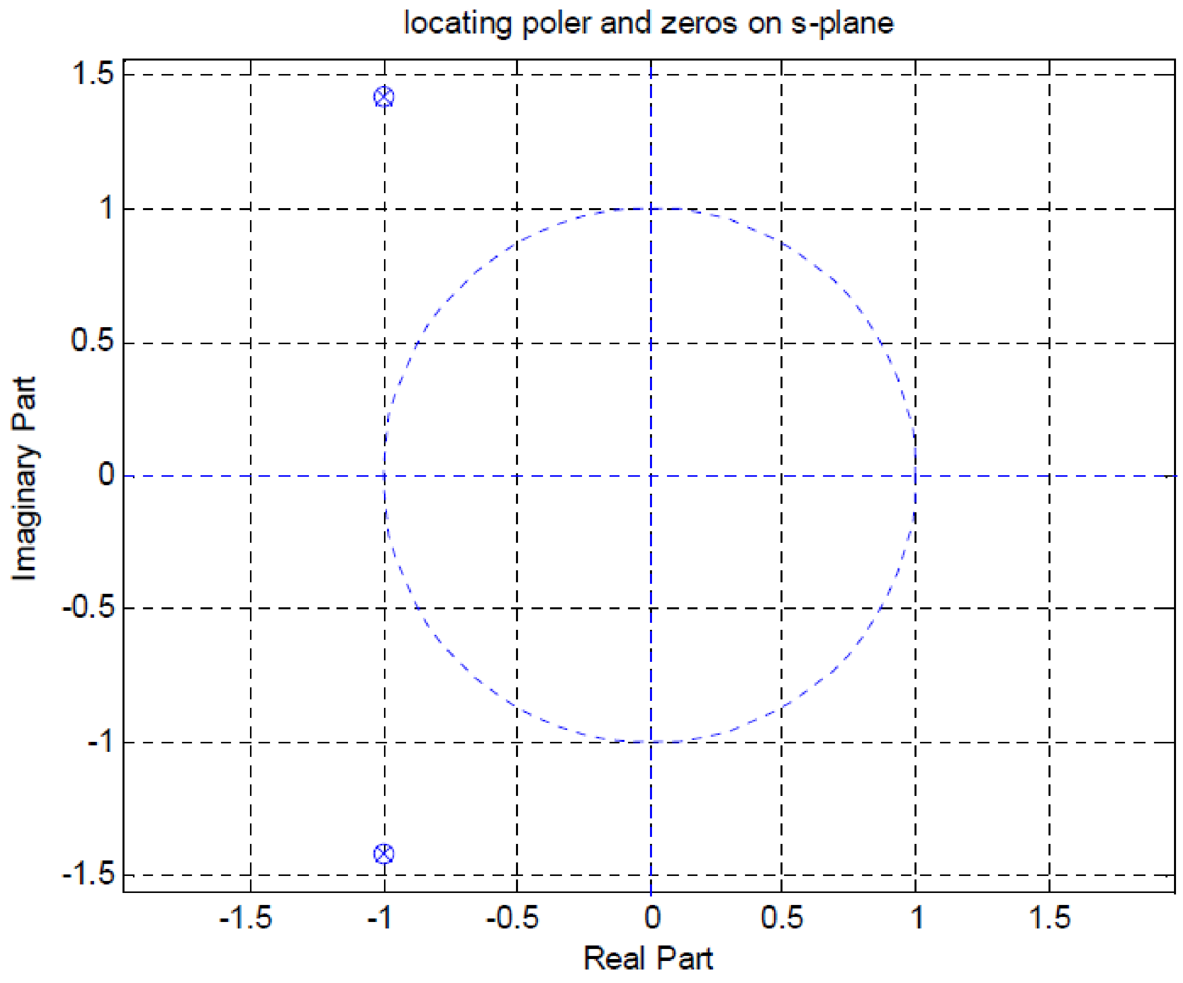
```
%locating poles of zero on s-plane%
clc;
clear all;
close all;
num=input('enter numerator co-efficients');
den=input('enter denominator co-efficients');
h=tf(num,den);
poles=roots(den);
zeros=roots(num);
sgrid;
pzmap(h);
grid on;
title('locating poles of zeros on s-plane');
%locating poles &zeros on z-plane%
clc;
clear all;
close all;
num=input('enter numerator coefficient');
den=input('enter denominator coefficient');
p=roots(den);
z=roots(num);
zplane(p,z);
grid;
title('locating poler and zeros on s-plane');
.
```

OUTPUT:-

```
locating poles of zero on s-plane
enter numerator coefficient[1 2 3]
enter denominator coefficient[7 6 5]
```



```
locating poles &zeros on z-plane
enter numerator coefficient[1 2 3]
enter denominator coefficient[1 2 3]
```



RESULT:-

In this experiment the zeros and poles and plotting the pole zero maps in s-plane and z-plane for the given transfer function using MATLAB.

VIVA QUESTIONS:-

1.Study the details of ztrans() and iztrans() functions?

2.What are poles and zeros?

3.How you specify the stability based on poles and zeros?

4.Define S-plane and Z-plane?

5.What is the difference b/w S-plane and Z-plane?

Experiment No: 11

Aim:-

: Wave form synthesis using Laplace Transforms**.**

AIM: Finding the Laplace transform & Inverse Laplace transform of some signals.

Software Required/ Equipment Required:-

1. MATLAB 7.6 / 7.8
2. Windows 7 SP1

Theory:

Bilateral Laplace transforms:

The Laplace transform of a signal f(t) can be defined as follows:

$$F(s) = \mathcal{L}\left\{f(t)\right\} = \int_{-\infty}^{\infty} e^{-st} f(t)\, dt.$$

Inverse Laplace transform

The inverse Laplace transform is given by the following formula :

$$f(t) = \mathcal{L}^{-1}\{F(s)\} = \frac{1}{2\pi i} \lim_{T\to\infty} \int_{\gamma-iT}^{\gamma+iT} e^{st} F(s)\, ds,$$

Program:

```
clc;
clear all;
close all;
%representation of symbolic variables
syms f t w s;
%laplace transform of t
f=t;
z=laplace(f);
disp('the laplace transform of f = ');
disp(z);
% laplace transform of a signal
%f1=sin(w*t);
f1=-1.25+3.5*t*exp(-2*t)+1.25*exp(-2*t);
v=laplace(f1);
disp('the laplace transform of f1 = ');
disp(v);
lv=simplify(v);
pretty(lv)
%inverse laplace transform
y1=ilaplace(z);
disp('the inverse laplace transform of z = ');
disp(y1);
```

```
y2=ilaplace(v);
disp('the inverse laplace transform of v = ');
disp(y2);
ezplot(y1);
figure;
ezplot(y2)
```

Output:

```
the laplace transform of f =
1/s^2
the laplace transform of f1 =
5/(4*(s + 2)) + 7/(2*(s + 2)^2) - 5/(4*s)
s - 5
----------
s (s + 2)2
the inverse laplace transform of z =
t
the inverse laplace transform of v =
5/(4*exp(2*t)) + (7*t)/(2*exp(2*t)) - 5/4
```

VIVA QUESTIONS:-

1. Define Laplace-Transform?
2. What is the Condition for Convergence of the L.T?
3. What is the Region of Convergence (ROC)?
4. State the Shifting property of L.T?
5. State convolution Property of L.T?

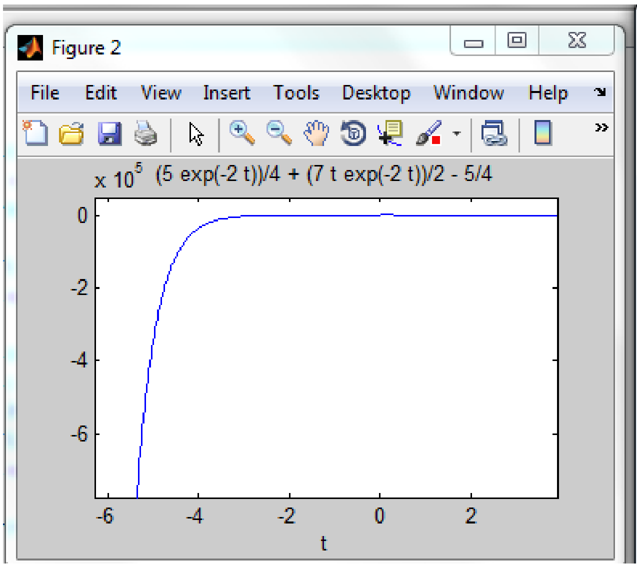
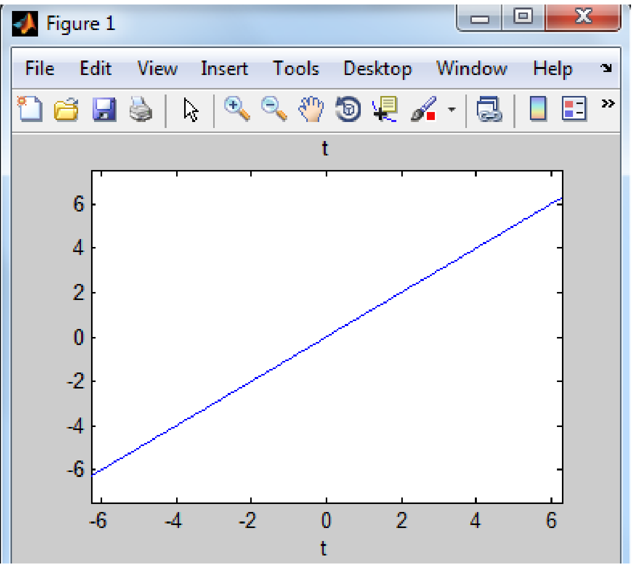


Figure 2
File Edit View Insert Tools Desktop Window Help
x 10^5 (5 exp(-2 t))/4 + (7 t exp(-2 t))/2 - 5/4
0
-2
-4
-6
-6
-4
-2
0
2
t



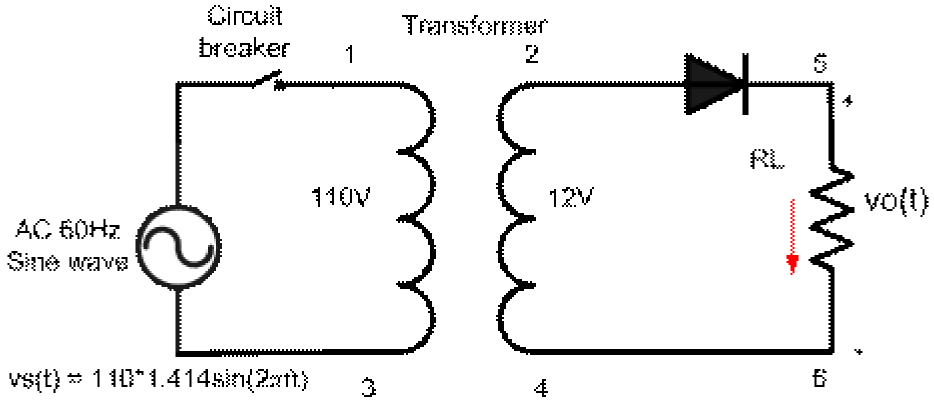
Experiment

Half wave rectifier

Half-Wave Rectifier

- Transformer voltage Vin = Vm sin ωt
- Diode rating
  - Forward voltage drop 0.6-0.7 v
  - Peak Inverse Voltage (PIV)
- Vout calculation
  - Vdc = Vavg = 0.318*Vm = 0.45*Vrms
  - Vrms across the load RL
- Consider the non-ideal diode with a 0.7 voltage drop, so Vout = Vdc – 0.7 v
- We will study the harmonics of this waveform later
- Ripple factor
  - RMS value of the ac components/ DC value of the component, or
  - R = Vrms/Vdc
- Ripple voltage = $(Vrms^2 - Vdc^2)^{1/2}$

The Half-wave Rectifier Circuit (without filter circuit)



Vout Calculation

$$v(t) = \begin{cases} V_m \sin \omega t & for \ t \le \frac{T}{2} \\ 0 & for \ \frac{T}{2} < t < T \end{cases}$$

Vdc or Vavg

We then compute the average voltage (dc voltage) Vavg or Vdc for one complete cycle

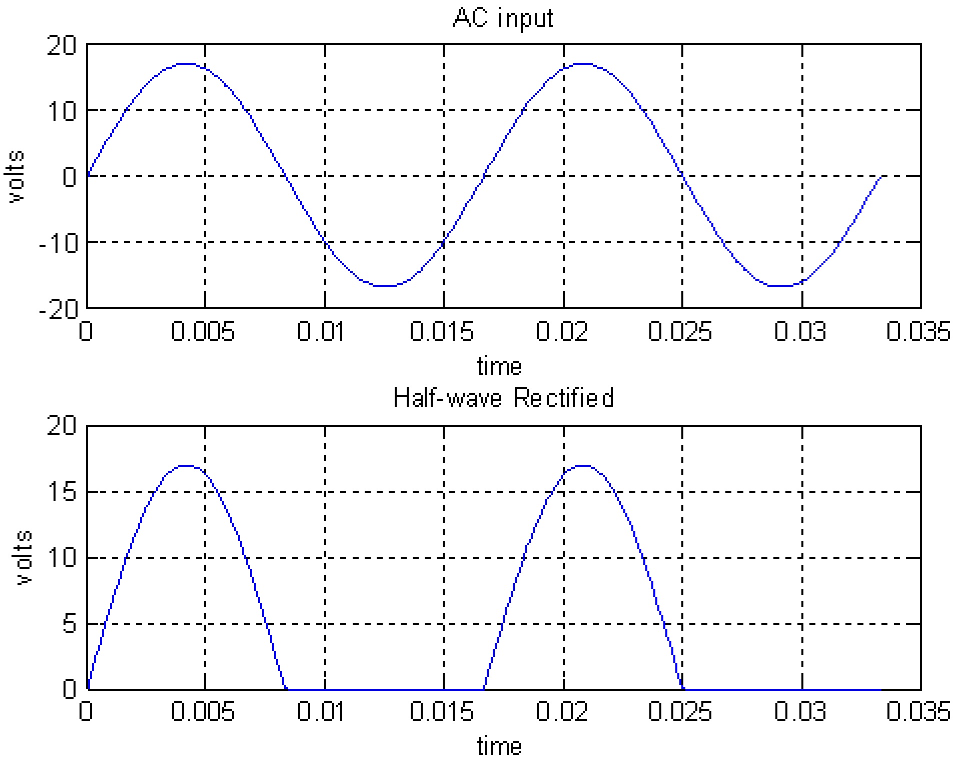
$$V_{dc} = \frac{1}{T}\int_0^T v(t)dt$$

$$= \frac{1}{2\pi}[\int_0^\pi V_m \sin\ \omega t\, dt + \int_\pi^{2\pi} 0dt]$$

$$= \frac{V_m}{2\pi}[-\cos\ \omega t]_0^\pi = \frac{V_m}{2\pi}[-\cos(\pi) - (-\cos 0)]$$

$$= \frac{V_m}{2\pi}[-(-1) - (-1)] = \frac{V_m}{\pi} = 0.318V_m$$

$$= 0.318(\sqrt{2})V_{rms} = 0.45V_{rms}$$


AC input

Half-wave Rectified

MATLAB Solution

```
% halfwave_rect.m
% 9/10/2006
% Paul Lin
f = 60;
T = 1/f;
Vacrms = 12;
Vm = Vacrms*1.414;
dt = T/100;
t = 0: dt: T;
vt = Vm*sin(2*pi*f*t);
vt_half = zeros(size(vt));
for n = 1: length(t)
  if vt(n) >= 0
     vt_half(n) = vt(n);
  else
```

```
        vt_half(n) = 0.0;
    end
end
row = 2;
col = 1;
figure(1), subplot(row, col, 1), plot(t, vt), grid on, title('AC input')
xlabel('time'), ylabel('volts')
subplot(row, col, 2), plot(t, vt_half), grid on, title('Half-wave Rectified')
xlabel('time'), ylabel('volts')
% MATLAB Numerical Integration
% Trapezoidal Integration: split the area under the curve into rectangles.
%  If the rectangles are fine enough, the sum of these areas gives the
% approximate value of the integral.
%
%     _pi
%    /              |pi
%   / sinx dx = -cos x|   = cos 0 - cos pi = 1- (-1) = 2
%  /                |0
% -0
%
x = linspace(0, pi,10);  % 1.9797, gives 1 percent error
y = sin(x);
trapz(x,y)
x = linspace(0, pi,100); %1.9998 gives 0.1 percent error
y = sin(x);
trapz(x,y)
x = linspace(0, pi,1000); %2.0000
y = sin(x);
trapz(x,y)
%vt_hal
% Exact Integration to obtain
% Vdc = 0.45*Vrms = 5.4 Volts
% Vdc = 0.318*Vm  = 5.4 Volts
% Numerical Integration
%  Vdc = 5.394 volt
w = 2*pi*f;
theta = w*t;
Vdc = trapz(theta(1:50), vt_half(1:50))/(2*pi)
```

Vrms at the Load Resistance

$$v(t) = \begin{cases} V_m \sin \omega t & for \;\; t \le \dfrac{T}{2} \\ 0 & for \;\; \dfrac{T}{2} < t < T \end{cases}$$

$$V_{rms} = \sqrt{\frac{1}{T}\int_0^T v^2(t)dt}$$

$$= [\frac{1}{2\pi}[\int_0^{T/2} V_m^2 \sin^2 \varpi t\, d(\varpi t) + \int_{T/2}^{T} 0 d(\varpi t)]^{1/2}$$

$$= [\frac{V_m^2}{2\pi}\int_0^{T/2} \sin^2 \varpi t\, d(\varpi t)]^{1/2}$$

$$\sin^2 \varpi t = \frac{1}{2}(1-\cos 2\varpi t),\ \varpi T = 2\pi,\ \theta = \varpi t$$

$$V_{rms} = [\frac{V_m^2}{4\pi}\int_0^{\pi}(1-\cos 2\theta)d\theta]^{1/2}$$

$$= [\frac{V_m^2}{4\pi}(t - \frac{1}{2}\sin\ 2\theta)\,|_0^{\pi}\,]^{1/2}$$

$$= [\frac{V_m^2}{4\pi}(\pi - \frac{1}{2}\sin(\ 2\pi) - 0 + \frac{1}{2}\sin\ 2(0)]^{1/2}$$

$$= [\frac{V_m^2}{4\pi}(\pi - 0 - 0 + 0)]^{1/2}$$

$$= \frac{V_m}{2}$$

Compute Ripple Factor

Ripple Factor = RMS value of the AC component/DC value of the component

$$Ripple = \frac{Vr_{rms}}{V_{dc}}$$

$$= \frac{\sqrt{V_{rms}{}^2 - V_{dc}{}^2}}{V_{dc}}$$

$$= \sqrt{\left[\frac{V_{rms}}{V_{dc}}\right]^2 - 1}$$

$$= \sqrt{\left[\frac{V_m/2}{V_m/\pi}\right]^2 - 1}$$

$$= \sqrt{\left[\frac{\pi}{2}\right]^2 - 1}$$

$$= 1.2114$$

Efficiency

η = (dc output power/ac input power) x 100%

$$\eta = \frac{V_{dc}{}^2 / R_L}{V_{rms}{}^2 / R_L} = \frac{(V_m / \pi)^2}{(V_m / 2)^2} = \frac{4}{\pi^2} = 0.406$$

or

$$\eta = 40.6\%$$

Form Factor

FF = rms value / average value = (Vm/2)/(Vm/π)= π/2 = 1.57

Peak Factor

Peak value / rms value = Vm/(Vm/2) = 2

Filter Capacitor Design

- Electrolytic capacitor (reservoir)
- Increase the average DC voltage to almost peak value (1.414*Vrms)
- For 10 % ripple, C = (5 * Io)/(Vs* f), where Io is the output current from the power supply in amps, Vs is the supply voltage in volts (peak value of the unsmoothed DC), and f is the frequency of the AC supply in Hz
- Capacitor must be doubled fro smoothing half-wave DC

Regulator

- Zener diode regulator
- IC regulator

Zener Diode Regulator

- Select a Zener diode with proper voltage rating, 4.7 V, 5.1V, and wattage rating (Vz*Iz) etc
- Use a Rz in series with Zener diode, compute Rz = (Vs – Vz)/Iz, and chose proper wattage rating

IC Voltage Regulator

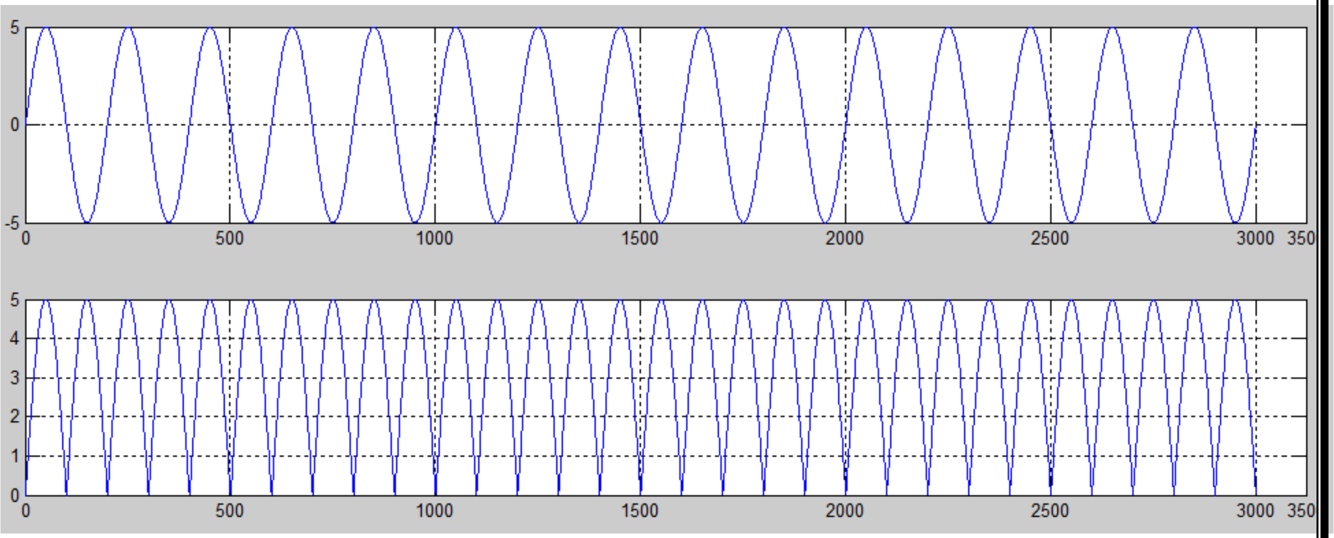
- LM7805, LM7812, etc

Full controller Rectifer

```
 t=(0:1/10000:0.3);
A= 5;
f=50; % 50 hz freq
sig=A*sin(2*pi*50*t); % sin wave of 0.3 sec, 50 Hz, and 5v amplitude
subplot(211)
plot(sig);
grid

u=1:9;
for l=1:3000
% checking for all 0.3*10000 samples %
if sin(2*pi*50*t(l))<=0
    sig(l)= -A*sin(2*pi*50*t(l));
else
    sig(l) = A*sin(2*pi*50*t(l));

end
end
subplot(212)
plot(sig);
grid
```



SPICE

1.1 INTRODUCTION

Electronic circuit design requires accurate methods of evaluating circuit performance. Because of the enormous complexity of modern integrated circuits, computer- aided circuit analysis is essential and can provide information about circuit performance that is almost impossible to obtain with laboratory prototype measurements.
Computer-aided analysis makes possible the following procedures:

1. Evaluation of the effects of variations in such elements as resistors, transistors, and transformers
2. Assessment of performance improvements or degradations
3. Evaluation of the effects of noise and signal distortion without the need for expensive measuring instruments
4. Sensitivity analysis to determine the permissible bounds determined by the tolerances of all element values or parameters of active elements
5. Fourier analysis without expensive wave analyzers
6. Evaluation of the effects of nonlinear elements on circuit performance
7. Optimization of the design of electronic circuits in terms of circuit parameters

SPICE (simulation program with integrated circuit emphasis) is a general-purpose circuit program that simulates electronic circuits. It can perform analyses on various aspects of electronic circuits, such as the operating (or quiescent) points of transistors, time-domain response, small-signal frequency response, and so on. SPICE contains models for common circuit elements, active as well as passive, and it is capable of simulating most electronic circuits. It is a versatile program and is widely used in both industry and academic institutions.
Until recently, SPICE was available only on mainframe computers.

1.4 TYPES OF ANALYSIS

PSpice allows various types of analysis. Each analysis is invoked by including its command statement. For example, a statement beginning with the .DC command invokes the DC sweep. The types of analysis and their corresponding . dot commands are described in the following text.

DC analysis is used for circuits with time-invariant sources (e.g., steady-state DC sources). It calculates all node voltages and branch currents for a range of values, and their quiescent (DC) values are the outputs. The dot commands and their functions are:

- DC sweep of an input voltage or current source, a model parameter, or temperature over a range of values (.DC)
- Determination of the linearized model parameters of nonlinear devices (.OP)
- DC operating point to obtain all node voltages
- Small-signal transfer function with small-signal gain, input resistance, and output resistance (Thevenin’s equivalent; .TF)
- DC small-signal sensitivities (.SENS)

Transient analysis is used for circuits with time-variant sources (e.g., AC sources and switched DC sources). It calculates all node voltages and branch currents over a time interval, and their instantaneous values are the outputs.

The dot commands and their functions are:

- Circuit behavior in response to time-varying sources (.TRAN)
- DC and Fourier components of the transient analysis results (.FOUR)

AC analysis is used for small-signal analysis of circuits with sources of variable frequencies. It calculates all node voltages and branch currents over a range of frequencies, and their magnitudes and phase angles are the outputs. The dot commands and their functions are:

- Circuit response over a range of source frequencies (.AC)
- Noise generation at an output node for every frequency (.NOISE)

2.1 INTRODUCTION

PSpice is a general-purpose circuit program that can be applied to simulate electronic and electrical circuits. A circuit must be specified in terms of element names, element values, nodes, variable parameters, and sources. The input to the circuit is to be simulated for calculating and plotting the transient response from 0 to 400 sec with an increment of 1 sec. The Fourier series coefficients and THD are to be printed. We discuss how to

(1) describe this circuit to PSpice,
(2) specify the type of analysis to be performed, and
(3) define the output variables required.

Description and analysis of a circuit require that the following be specified:

- ✓ Input files
- ✓ Nodes
- ✓ Element values
- ✓ Circuit elements
- ✓ Element models
- ✓ Sources
- ✓ Output variables
- ✓ Types of analysis
- ✓ PSpice output commands
- ✓ Format of circuit files
- ✓ Format of output files

2.2 INPUT FILES

The input to the SPICE simulation can be either a Schematics file or a net-list file (also known as the *circuit file*). In a circuit file, the user assigns the node numbers to the circuit .

2.3 NODES

For PSpice A/D: Node numbers, which must be integers from 0 to 9999 but need connected between nodes. The node numbers are specified after the name of the element connected to the node. Node 0 is predefined as the ground. All nodes must be connected to at least two elements and should therefore appear at least twice. All nodes must have a DC path to the ground node.

2.4 ELEMENT VALUES

The value of a circuit element is written after the nodes to which the element is connected. The values are written in standard floating-point notation with optional scale and units suffixes. Some values without suffixes that are allowed by PSpice are

55. 5.0 5E□3 5.0E□3 5.E3

There are two types of suffixes: the scale suffix and the units suffix. The scale suffixes multiply the numbers that they follow. Scale suffixes recognized by PSpice are:

F	1E–15
P	1E–12
N	1E–9
U	1E–6
MIL	25.4E–6
M	1E–3
K	1E3
MEG	1E6
G	1E9

T 1E12

The units suffixes that are normally used are:

V- volt
A -ampere
HZ- hertz
OHM -ohm
H -henry
F -farad
DEG –degree

Symbols of Circuit Elements and Sources

First Letter	**Circuit Elements and Sources**
B	GaAs MES field-effect transistor
C	Capacitor
D	Diode
E	Voltage-controlled voltage source
F	Current-controlled current source
G	Voltage-controlled current source
H	Current-controlled voltage source
I	Independent current source
J	Junction field-effect transistor
K	Mutual inductors (transformer)
L	Inductor
M	MOS field-effect transistor
Q	Bipolar junction transistor
R	Resistor
S	Voltage-controlled switcha
T	Transmission line
V	Independent voltage source
W	Current-controlled switcha

Commands:

.PROBE : Probe is a *graphical waveform analyzer* for PSpice

.PLOT : This command generates the plot on the output file

.PRINT : This command gives a table of data on the output file

.END : End of file statement

2.5 CIRCUIT ELEMENTS

For PSpice A/D: Circuit elements are identified by name. A name must start with a letter symbol corresponding to the element, but after that it can contain either letters or numbers. Names can be up to eight characters long. Table 2.2 must start with a C.

The format for describing passive elements is

□element name□□□positive node□□□negative node□□□value□

where the current is assumed to flow from the positive node N□□to the negative node N−□

The statement that R_1 has a value of 2 □□and is connected between nodes 7 and 5 is

R_1 7 5 2

The statement that L_1 has a value of 50 □H and is connected between nodes 5 and 3 is

L_1 5 3 50UH

The statement that C_1 has a value of 10 □F and is connected between nodes 3 and 0 is

C_1 3 0 10UF

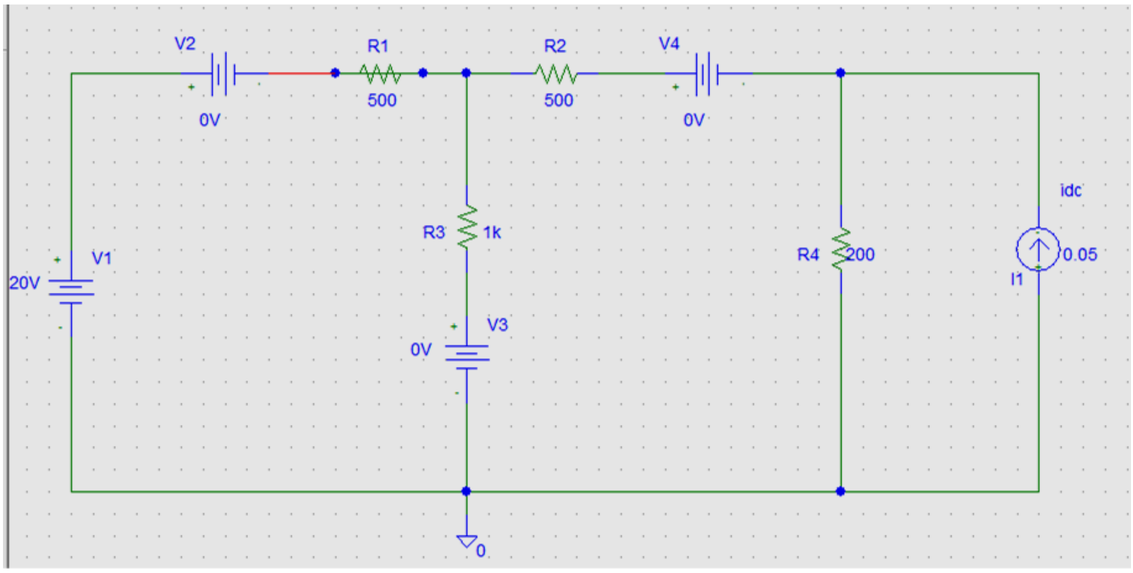
TITLE: SIMULATION OF DC CIRCUIT

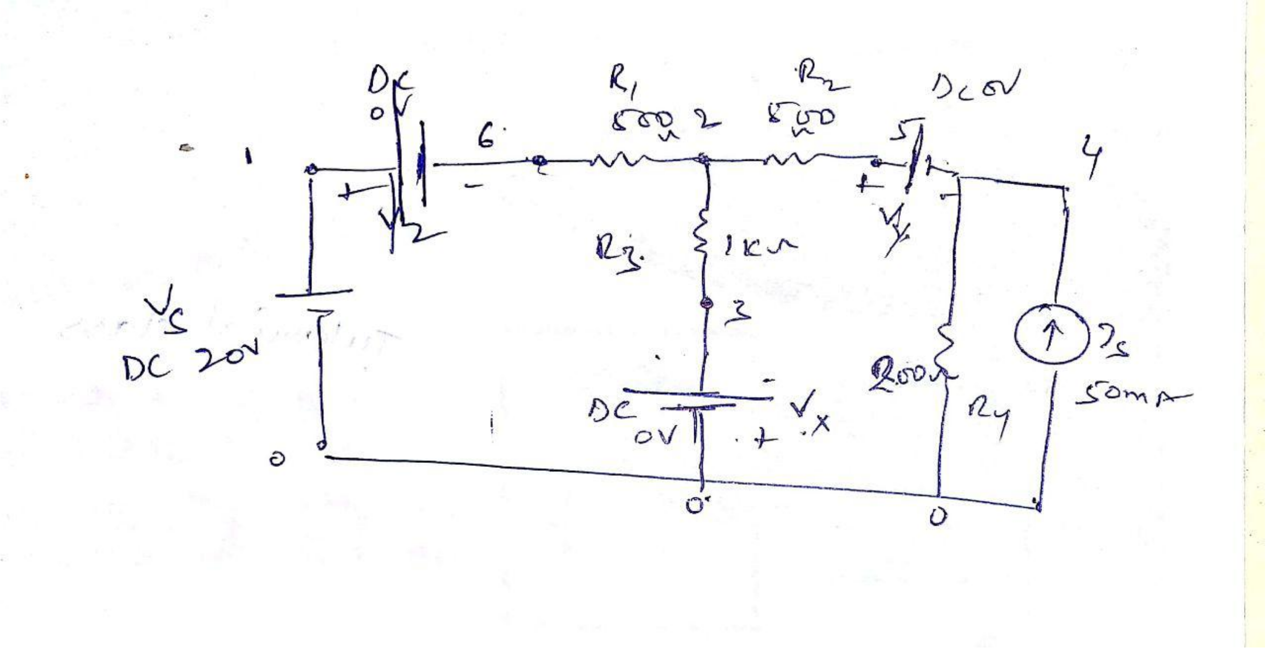
Aim: To simulate the circuit on Pspice and to find out the node voltages and respective branch currents.

Software: PSPICE

Version: MICROSIM EVALUATION 8.0

Circuit diagram:





Program:

```
VS 1 0 DC 20V
IS  0 4 DC 50MA
R1 6 2  500
R2  2  5 800
R3 2   3  1KOHM
R4  4 0  200
VX  3 0 DC 0V
VY 5 4 DC 0V
VZ 1 6 DC 0V
.DC  VS 10V 30V 10V
.PRINT DC V(4)  I(VX)  I(VY) I(VZ)
.END
```

Output file

RESULT:

THEORITICAL CALCULATIONS:

Note :File name is given with < ***file name .cir*** >extension

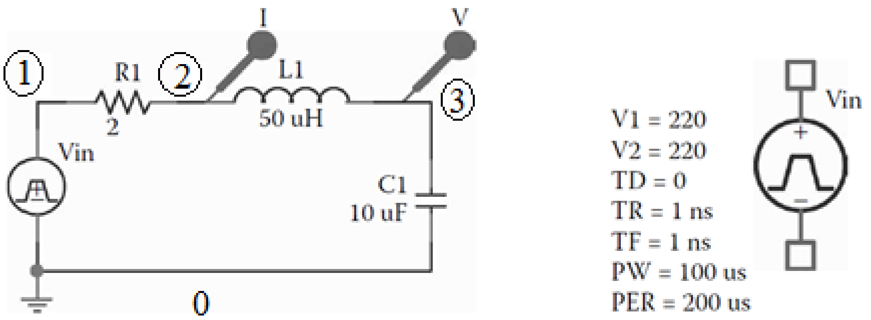
TITLE : TRANSIENT RESPONSE OF A DC CIRCUIT

Aim: To find the dc transient response of a series RLC circuit for a PULSE input.

Software: PSPICE

Version: MICROSIM EVALUATION 8.0

Circuit diagram:



(a) (b)

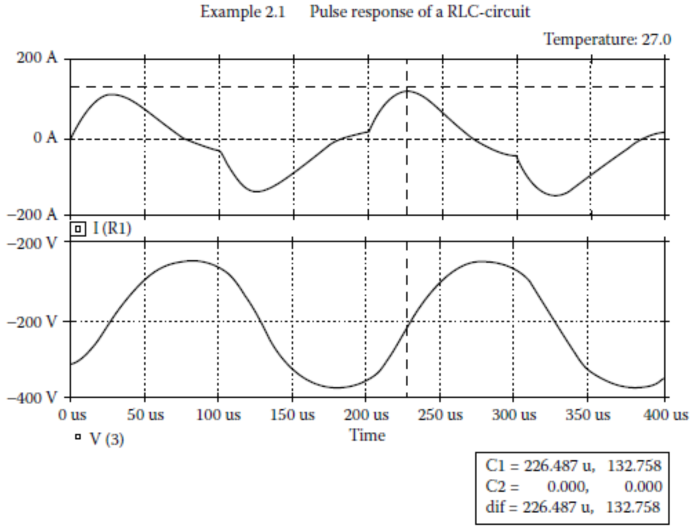
Program:

```
VIN 1 0  PULSE(-220 220  0 0 1NS  1NS 100US   200US)
R1  1  2  2


L1  2  3  50UH
C1  3  0  10UF
.TRAN  1US  400US
.PROBE
.END
```

Output file:

RESULT:



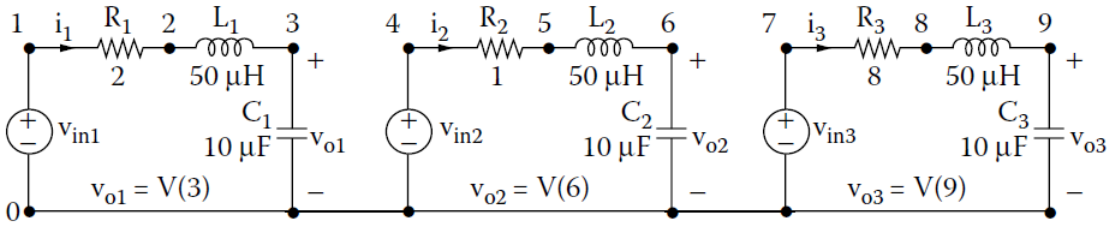
DC TRANSIENT RESPONSE OF A SERIES RLC CIRCUIT FOR A STEP INPUT.

Aim: To find the dc transient response of a series RLC circuit for a STEP input.

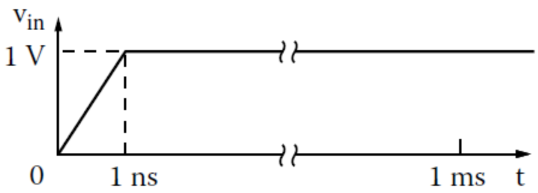
Software: PSPICE

Version: MICROSIM EVALUATION 8.0

Circuit diagram:



(a) Circuit



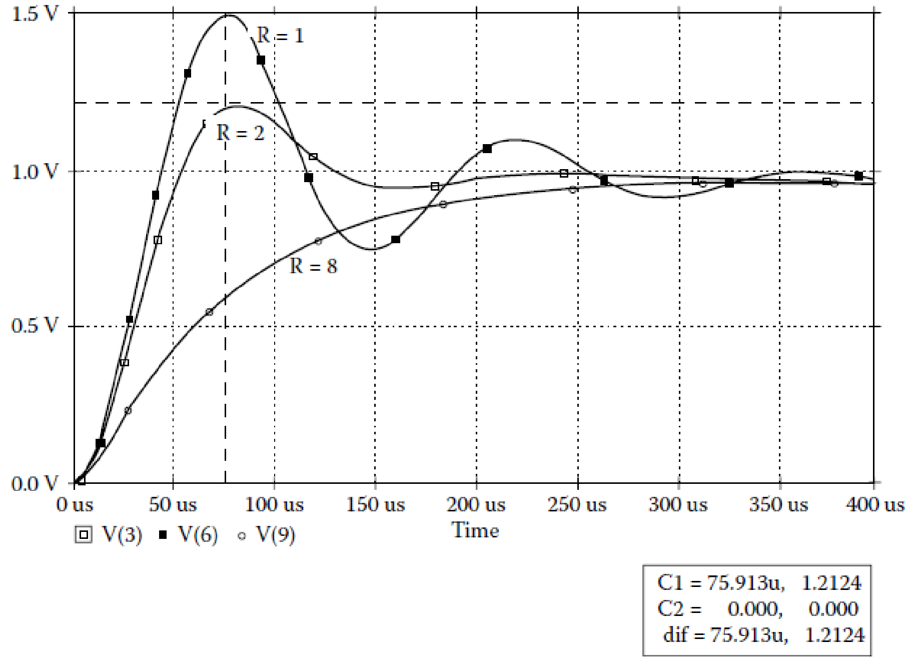
(b) Step voltage

Program:

```
VI1   1  0 PWL  (0 0 1NS 1V 1MS 1V)
VI2   4  0 PWL (0 0 1NS 1V 1MS 1V)
VI3   7  0 PWL (0 0 1NS 1V 1MS 1V)
R1  1  2  2
L1   2 3 50UH
C1  3 0 10UF
R2  4 5 1
L2  5 6 50UH
C2  6 0 10UF
R3  7 8 8
L3  8 9 50UH
C3  9 0 10UF
.TRAN 1US 400US
.PROBE
.END
```

Output file:

RESULT:



```
C1 = 75.913u,  1.2124
C2 =    0.000,   0.000
 dif = 75.913u,  1.2124
```

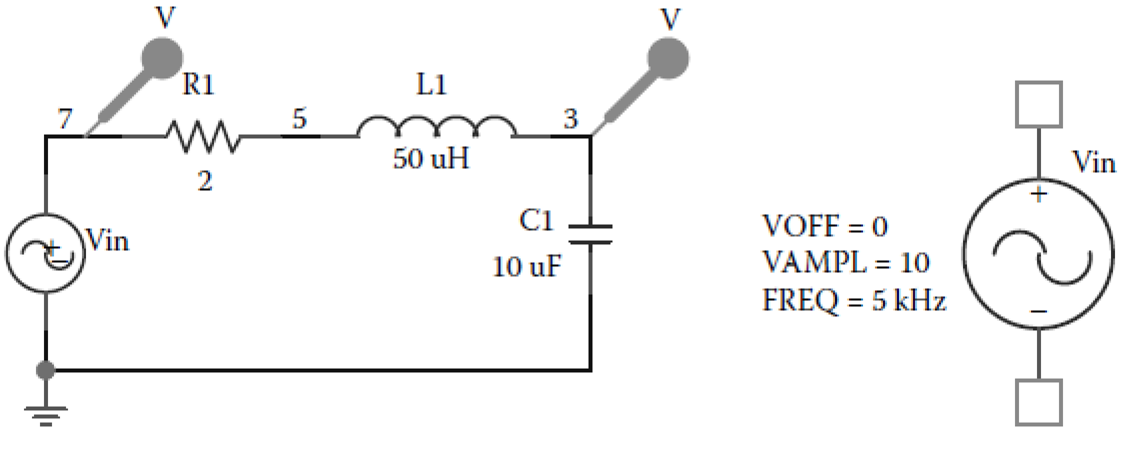
b) **DC TRANSIENT RESPONSE OF A SERIES RLC CIRCUIT FOR A SINE INPUT.**

Aim: To find the dc transient response of a series RLC circuit for a SINE input.

Software: PSPICE

Version: MICROSIM EVALUATION 8.0

Circuit diagram:



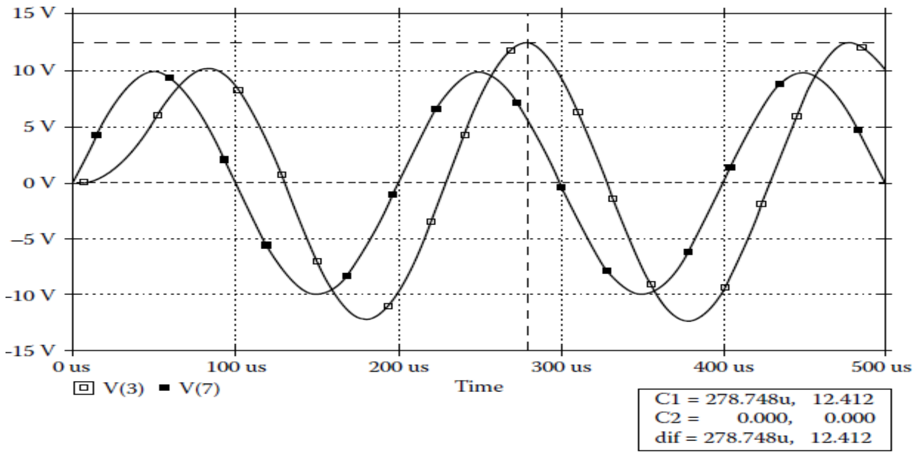
(a) (b)

Program:

```
VIN   7  0 SIN(0  10V   5KHZ)
R1  7  5  2
L1   5 3 50UH
C1  3 0 10UF
.TRAN 1US 500US
.PLOT TRAN V(3) V(7)
.PROBE
.END
```

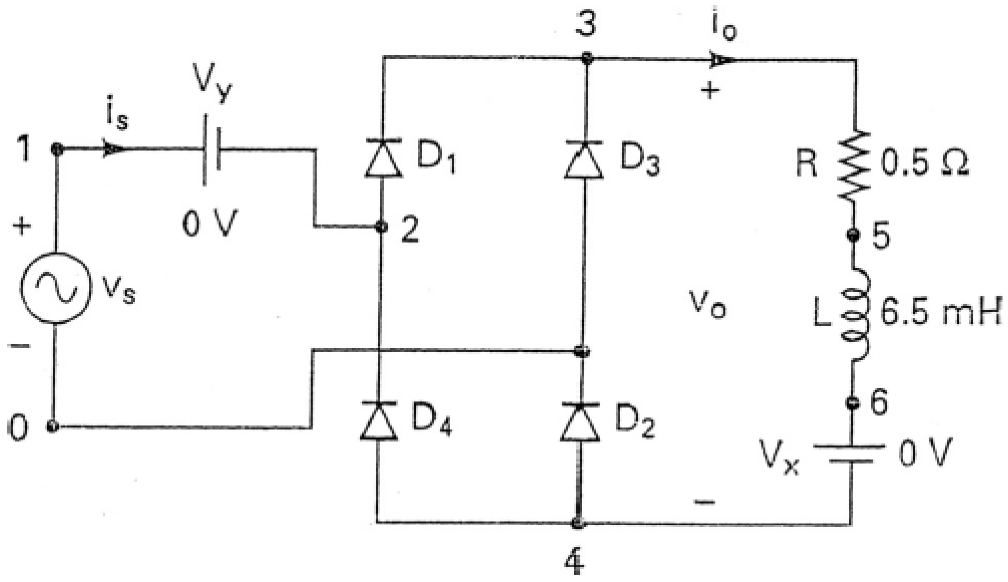
Output file:

RESULT:

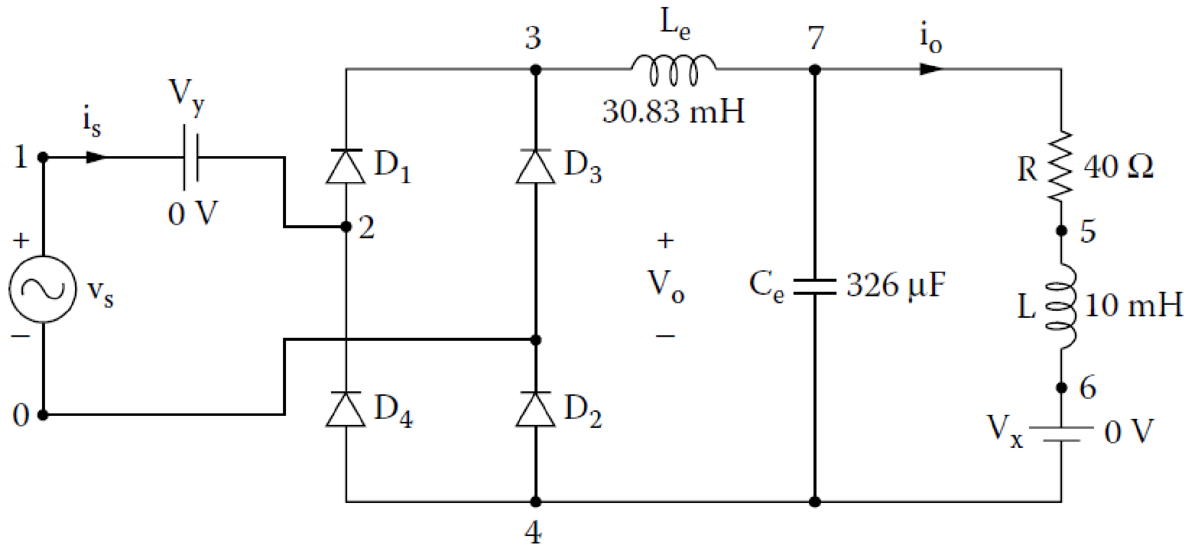


Title: PERFORMANCE OF A SINGLE-PHASE BRIDGE RECTIFIER

Aim: to find the performance of single phase bridge rectifier
Software required: PSPICE-AD
Circuit Diagram:



PSPICE Circuit:

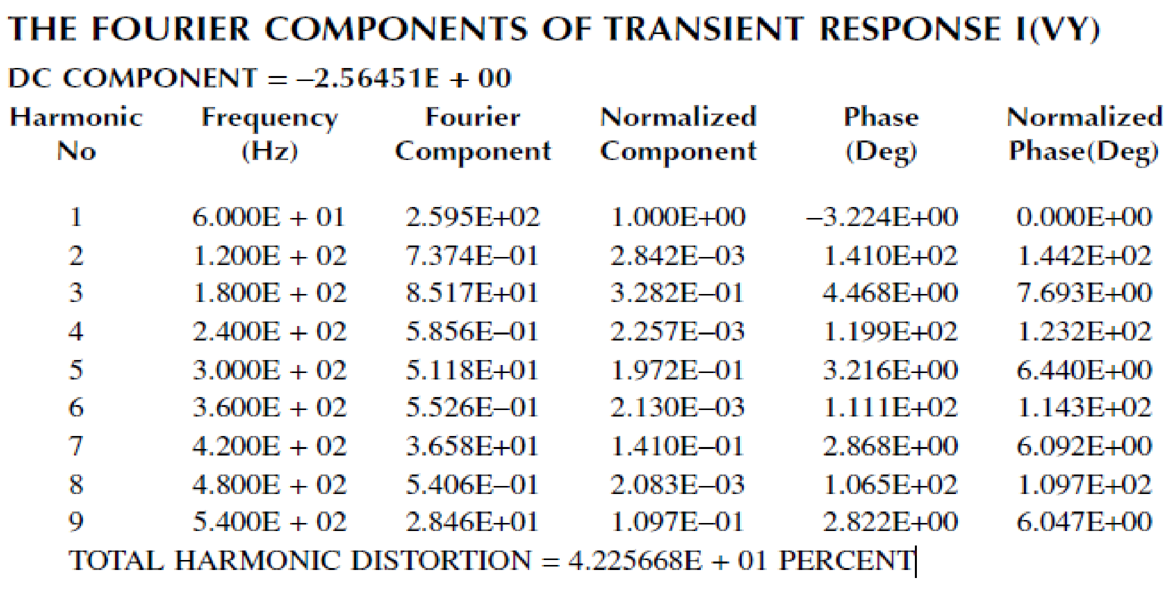


Program:

Single-phase bridge rectifier with RL load

```
VS 1 0 SIN (0 169.7V  60HZ)
R 3 5 0.5
L 5 6 6.5MH
VX 6 4 DC 0V ; Voltage source to measure the output current
VY 1 2 DC 0V ; Voltage source to measure the output Current
D1 2 3 DMOD
D3 0 3 DMOD
D2 4 0 DMOD
D4 4 2 DMOD
.MODEL DMOD D(IS=2.22E-15 BV=1200V IBV=13E3 CJO=2PF TT=1US)
TRAN 10US 50MS 33.3333MS 10US ; Transient analysis
.FOUR 60HZ 1(VY) ; Fourier analysis of input current (optional)
.PROBE ; Graphic POSTpost- processor
.OPTIONS ABSTOL = 1.0 N RELTOL = .01 BNTOL = 1.0M ITL5=10000 ; (optional)
.END
```

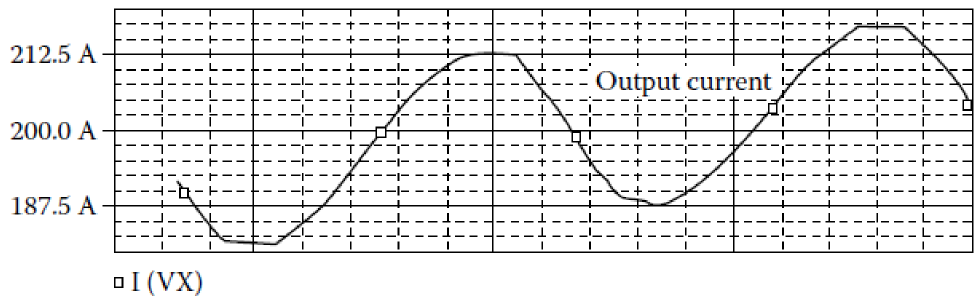
THE FOURIER COMPONENTS OF TRANSIENT RESPONSE I(VY)

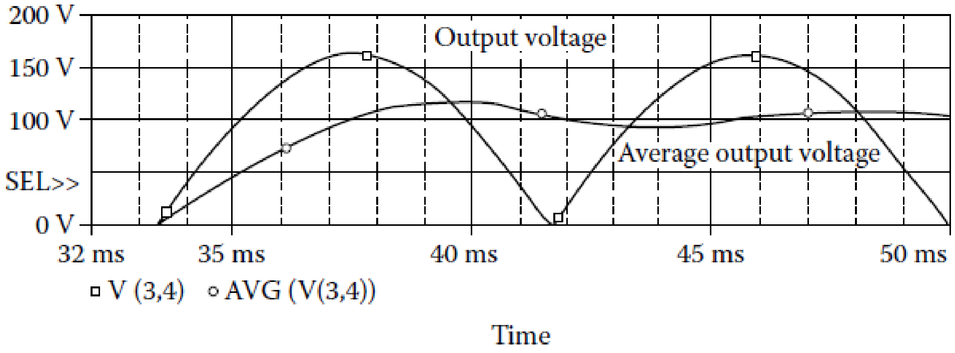


```
THE FOURIER COMPONENTS OF TRANSIENT RESPONSE I(VY)
DC COMPONENT = –2.56451E + 00
Harmonic   Frequency    Fourier     Normalized    Phase      Normalized
   No        (Hz)       Component   Component     (Deg)      Phase(Deg)

   1      6.000E + 01   2.595E+02   1.000E+00   –3.224E+00   0.000E+00
   2      1.200E + 02   7.374E–01   2.842E–03    1.410E+02   1.442E+02
   3      1.800E + 02   8.517E+01   3.282E–01    4.468E+00   7.693E+00
   4      2.400E + 02   5.856E–01   2.257E–03    1.199E+02   1.232E+02
   5      3.000E + 02   5.118E+01   1.972E–01    3.216E+00   6.440E+00
   6      3.600E + 02   5.526E–01   2.130E–03    1.111E+02   1.143E+02
   7      4.200E + 02   3.658E+01   1.410E–01    2.868E+00   6.092E+00
   8      4.800E + 02   5.406E–01   2.083E–03    1.065E+02   1.097E+02
   9      5.400E + 02   2.846E+01   1.097E–01    2.822E+00   6.047E+00
   TOTAL HARMONIC DISTORTION = 4.225668E + 01 PERCENT
```

DC input current *I*in(DC) □□□2.56 A, which should ideally be zero
Rms fundamental input current *I*1(rms) = 259.5/√2 = 183.49 A
THD of input current *THD* □□42.26% □□0.4226
Rms harmonic current *I*h(rms) □□*I*1(rms) □□*THD* □□183.49 □□0.4226 □□77.54 A





Rms input current $I_s = (I^2_{\text{in(DC)}} + I^2_{1(\text{rms})} + I^2_{h(\text{rms})})^{1/2}$

$= (2.56^2+183.49^2+77.54^2)^{1/2}=199.22$ A

Displacement angle $\phi_1 = -3.22$

Displacement factor $DF = \cos\,\phi_1 = \cos(-3.22) = 0.998$ (lagging)

Thus, the input power factor is

$$\text{PF} = \frac{I_{1(\text{rms})}}{I_s}\cos\phi_1 = \frac{183.49}{199.22}\times 0.998 = 0.9192\ \ \text{(lagging)}$$

Assuming that $I_{\text{in(DC)}} = 0$, Equation 7.3 gives the power factor as

$$\text{PF} = \frac{1}{(1+0.4226^2)1/2}\times 0.9981 = 0.9193\ \ \text{(lagging)}$$

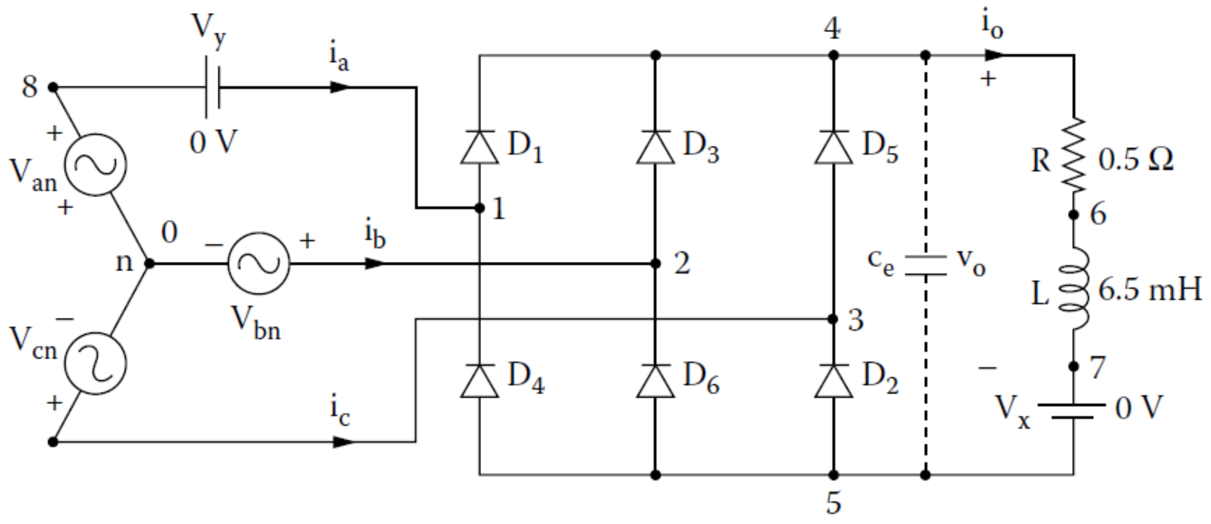
Note: The load current is continuous. The input power factor (0.919) is much higher compared to that (0.3802) of the half-wave rectifier.

PERFORMANCE OF A THREE-PHASE BRIDGE RECTIFIER

Aim: To find performance of a three-phase bridge rectifier

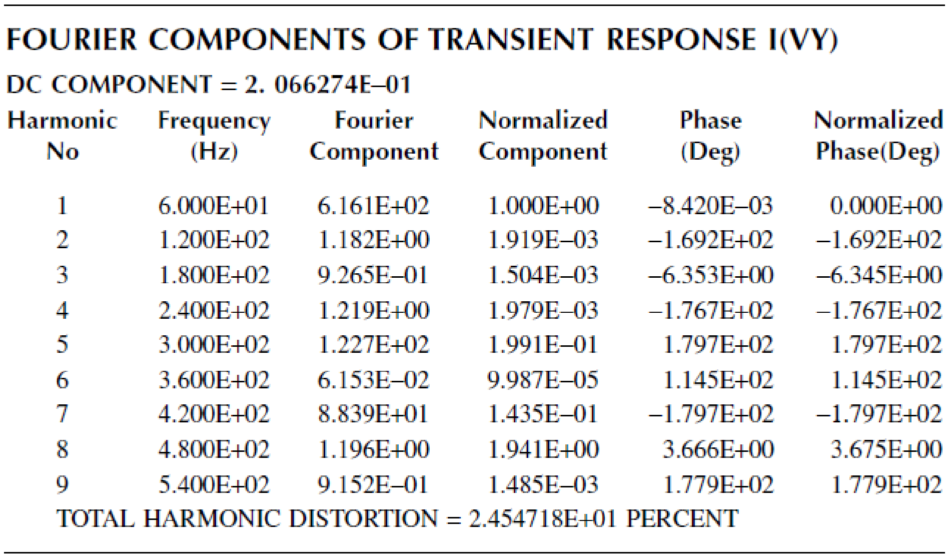
Software required: PSPICE-AD

Circuit Diagram:



Example 7.5 Three-phase bridge rectifier

```
Van 8 0 SIN (0 169.7V 60HZ)
Vbn 2 0 SIN (0 169.7V 60Hz 0 0 120DEG)
Vcn 3 0 SIN (0 169.7V 60Hz 0 0 240DEG)
CE 4 5 1UF ; Small capacitance to aid convergence
R 4 6 0.5
L 6 7 6.5MH
VX 7 5 DC 0V ; Voltage source to measure the output current
VY 8 1 DC 0V ; Voltage source to measure the input current
D1  1  4  DMOD
D3  2  4  DMOD
D5  3  4  DMOD
D2  5  3  DMOD
D6  5  2  DMOD
D4  5  1  DMOD
.MODEL  DMOD  D (IS=2.2 2E-15 BV=1200V IBV=13E-3 CJO=2PF TT=1US)
.TRAN  10US  33.3333MS  0  10US ; Transient analysis
.FOUR 60Hz 1(VY) ; Fourier analysis of line current
.PROBE ; Graphics post-processor
.OPTIONS ABSTOL = 1.0N RENTOL = 1.0M VNTOL = 1.0M ITL5=10000 ;
. END
```



FOURIER COMPONENTS OF TRANSIENT RESPONSE I(VY)

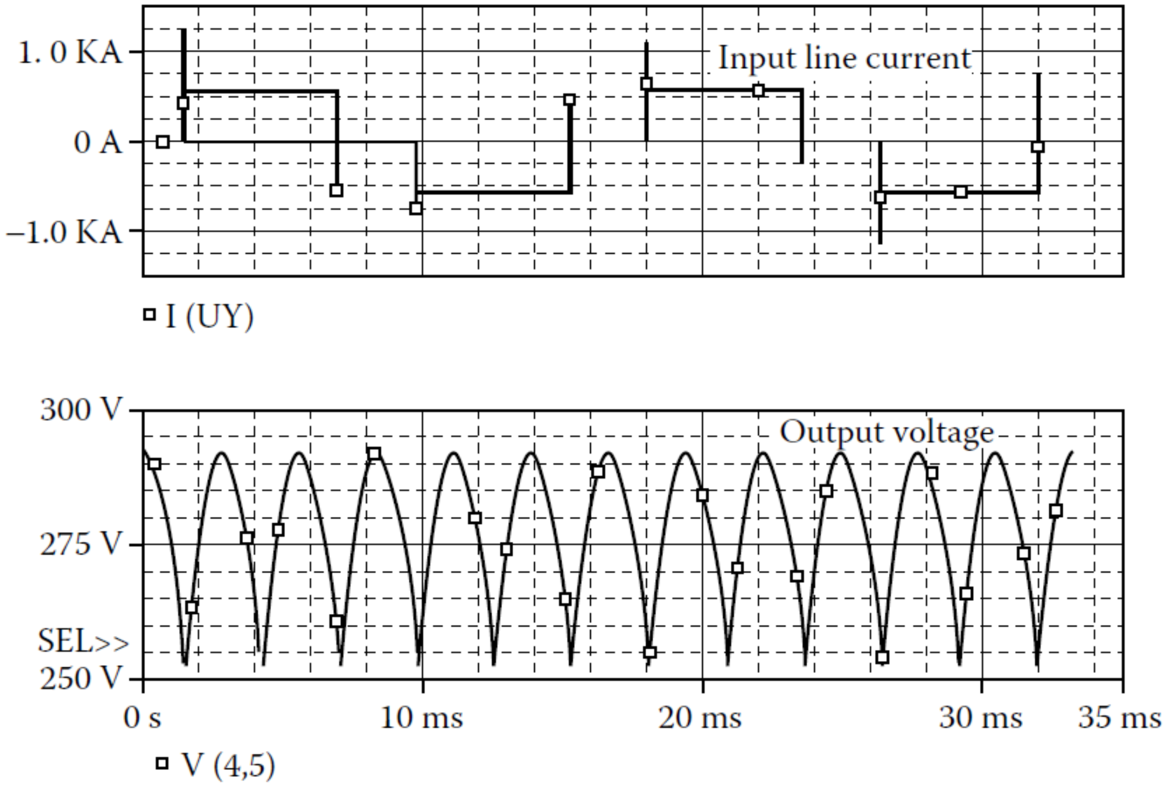
DC COMPONENT = 2. 066274E–01

Harmonic No	Frequency (Hz)	Fourier Component	Normalized Component	Phase (Deg)	Normalized Phase(Deg)
1	6.000E+01	6.161E+02	1.000E+00	–8.420E–03	0.000E+00
2	1.200E+02	1.182E+00	1.919E–03	–1.692E+02	–1.692E+02
3	1.800E+02	9.265E–01	1.504E–03	–6.353E+00	–6.345E+00
4	2.400E+02	1.219E+00	1.979E–03	–1.767E+02	–1.767E+02
5	3.000E+02	1.227E+02	1.991E–01	1.797E+02	1.797E+02
6	3.600E+02	6.153E–02	9.987E–05	1.145E+02	1.145E+02
7	4.200E+02	8.839E+01	1.435E–01	–1.797E+02	–1.797E+02
8	4.800E+02	1.196E+00	1.941E+00	3.666E+00	3.675E+00
9	5.400E+02	9.152E–01	1.485E–03	1.779E+02	1.779E+02

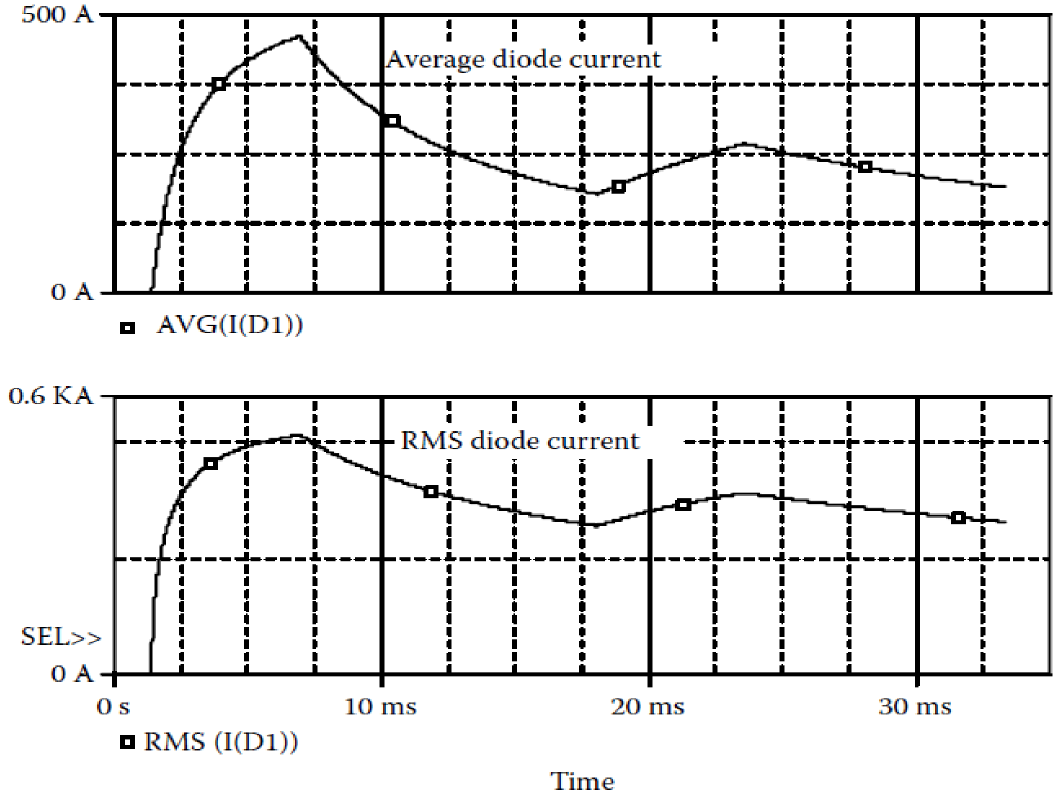
TOTAL HARMONIC DISTORTION = 2.454718E+01 PERCENT

THD of input current, *THD* □□24.55% □□0.2455
Displacement angle, Φ1 = □0□
Displacement factor, *DF* □□cos Φ1□□cos (0) =□□
Neglecting the DC input current *I***in(DC)** □□0.207 A, which is small relative to the fundamental component, we can find power factor *PF* from Equation

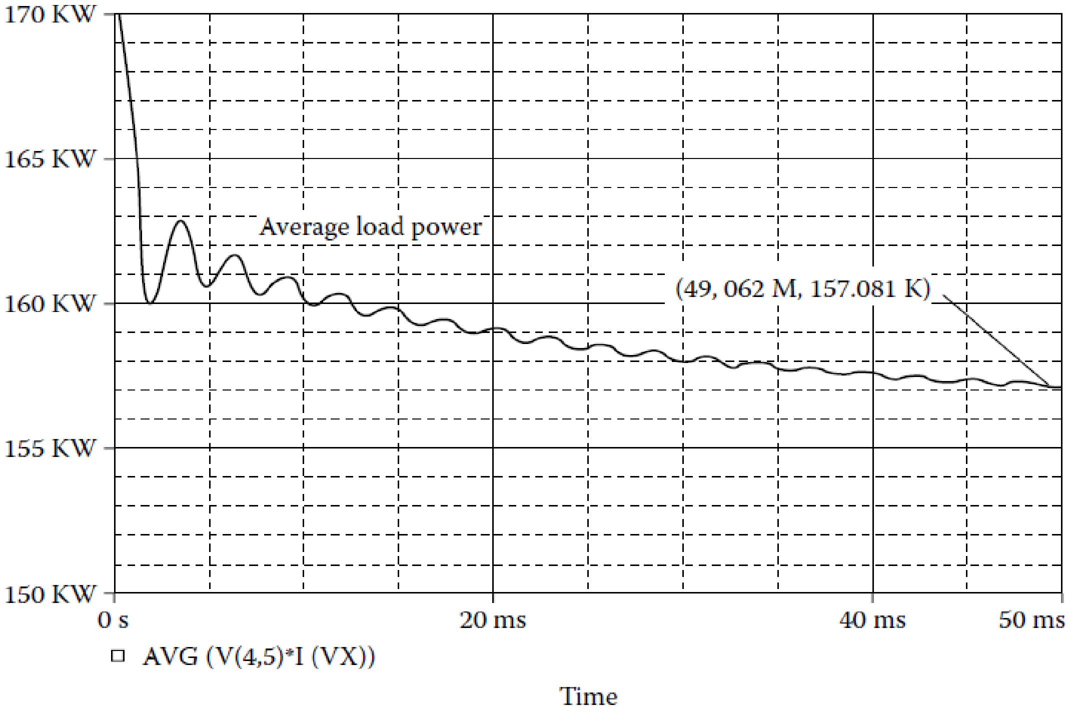
$$\mathrm{PF} = \frac{1}{(1+0.2455^2)^{1/2}} \times|1 = 0.971 \ \text{(lagging)}$$



Plots of output voltage (4,5) and line current I(VY)



Plots of rms and average currents through diode D1



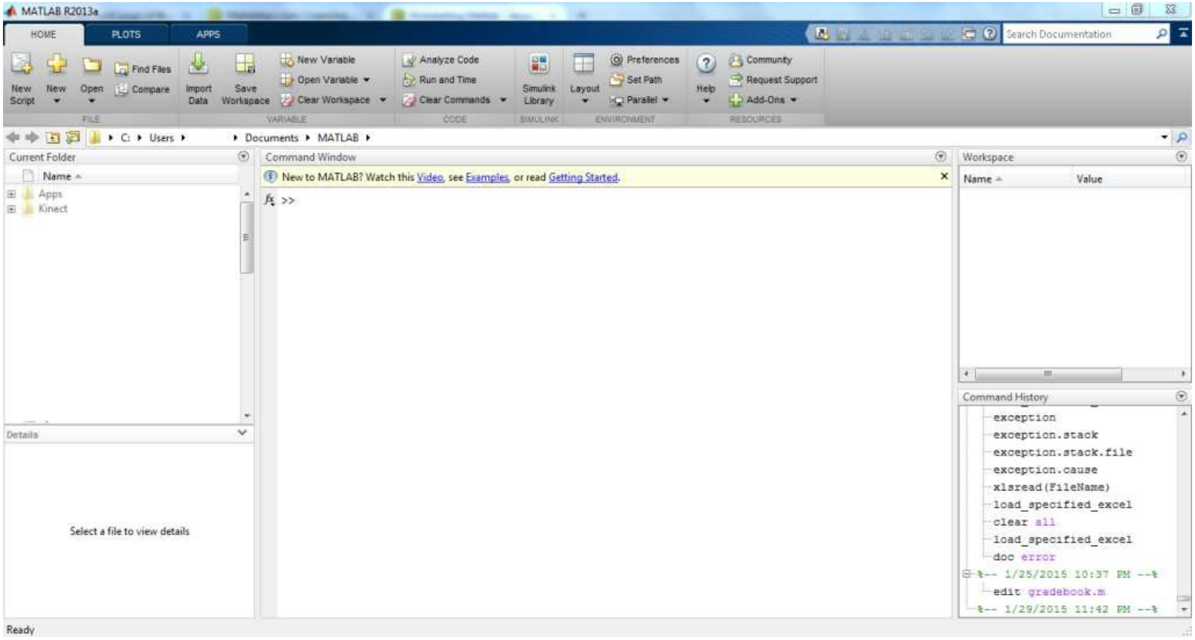
Verification of theorems using matlab

INTRODUCTION TO MATLAB

The name MATLAB stands for MATrix LABoratory. MATLAB was written originally to provide easy access to matrix software developed by the LINPACK (linear system package) and EISPACK (Eigen system package) projects.

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming environment. Furthermore, MATLAB is a modern programming language environment: it has sophisticated data structures, contains built-in editing and debugging tools, and supports object-oriented programming. These factors make MATLAB an excellent tool for teaching and research. MATLAB has many advantages compared to conventional computer languages (e.g., C, FORTRAN) for solving technical problems. MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. The software package has been commercially available since 1984 and is now considered as a standard tool at most universities and industries worldwide.

It has powerful built-in routines that enable a very wide variety of computations. It also has easy to use graphics commands that make the visualization of results immediately available. Specific applications are collected in packages referred to as toolbox. There are toolboxes for signal processing, symbolic computation, control theory, simulation, optimization, and several other fields of applied science and engineering.



New to MATLAB? Watch this Video, see Examples, or read Getting Started.

```
fx >>
```

Select a file to view details

```
exception
exception.stack
exception.stack.file
exception.cause
xlsread(FileName)
load_specified_excel
clear all
load_specified_excel
doc error
%-- 1/25/2015 10:37 PM --%
edit gradebook.m
%-- 1/29/2015 11:42 PM --%
```

Ready

This is the default layout of MATLAB version used in our laboratory.
The main window is the Command Window. You can type in there any command that is available in MATLAB.
The second window in importance is the workspace. This is the current state of memory in

MATLAB. The entire variables that are being used go there. The command history and the current folder are just useful tool that you can use but they are not essential to understand MATLAB.

Using MATLAB as a calculator:

As an example of a simple interactive calculation, just type the expression you want to evaluate. Let's start at the very beginning. For example, let's suppose you want to calculate the expression, 1 + 2 × 3. You type it at the prompt command (>>) as follows,

```
>> 1+2*3
ans = 7
```

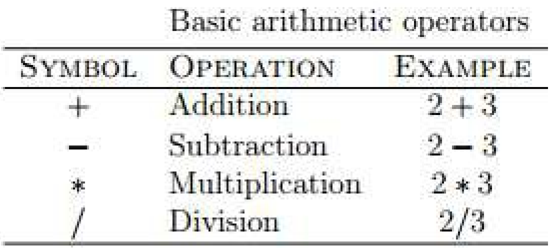
You will have noticed that if you do not specify an output variable, MATLAB uses a default variable ans, short for answer, to store the results of the current calculation. Note that the variable ans is created (or overwritten, if it is already existed). To avoid this, you may assign a value to a variable or output argument name.

For example,

```
>> x = 1+2*3
```

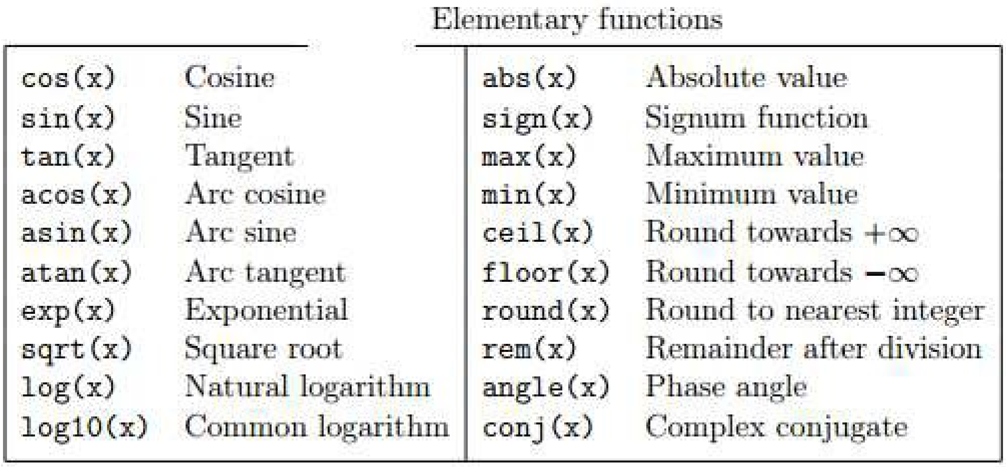
x = 7 will result in x being given the value 1 + 2 × 3 = 7. This variable name can always be used to refer to the results of the previous computations. Therefore, computing 4x will result in

```
>> 4*x
ans = 28.0000
```



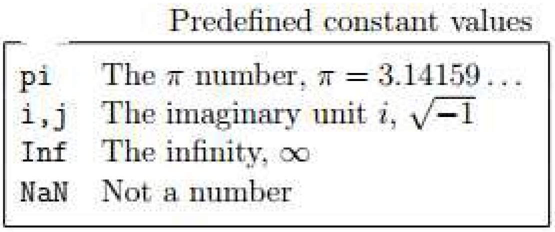
Basic arithmetic operators

SYMBOL	OPERATION	EXAMPLE
+	Addition	$2+3$
−	Subtraction	$2-3$
*	Multiplication	$2*3$
/	Division	$2/3$



Elementary functions

`cos(x)`	Cosine	`abs(x)`	Absolute value
`sin(x)`	Sine	`sign(x)`	Signum function
`tan(x)`	Tangent	`max(x)`	Maximum value
`acos(x)`	Arc cosine	`min(x)`	Minimum value
`asin(x)`	Arc sine	`ceil(x)`	Round towards $+\infty$
`atan(x)`	Arc tangent	`floor(x)`	Round towards $-\infty$
`exp(x)`	Exponential	`round(x)`	Round to nearest integer
`sqrt(x)`	Square root	`rem(x)`	Remainder after division
`log(x)`	Natural logarithm	`angle(x)`	Phase angle
`log10(x)`	Common logarithm	`conj(x)`	Complex conjugate



Predefined constant values

`pi`	The π number, $\pi = 3.14159\ldots$
`i,j`	The imaginary unit i, $\sqrt{-1}$
`Inf`	The infinity, ∞
`NaN`	Not a number

MATLAB by default displays only 4 decimals in the result of the calculations, for example

−163.6667, as shown in above examples. However, MATLAB does numerical calculations in double precision, which is 15 digits. The command format controls how the results of computations are displayed. Here are some examples of the different formats together with the resulting outputs.

```
>> format short
```

```
>> x=-163.6667
```

If we want to see all 15 digits, we use the command format long

```
>> format long
```

```
>> x= -1.636666666666667e+002
```

To return to the standard format, enter format short, or simply format. There are several other formats. For more details, see the MATLAB documentation, or type help format.

Managing the workspace:

The contents of the workspace persist between the executions of separate commands. Therefore, it is possible for the results of one problem to have an effect on the next one. To avoid this possibility, it is a good idea to issue a clear command at the start of each new independent calculation.

```
>> clear
```

The command clear or clear all removes all variables from the workspace. This frees up system memory.

In order to display a list of the variables currently in the memory, type

```
>> who
```

while, whos will give more details which include size, space allocation, and class of the variables.

Here are few additional useful commands:

- To clear the Command Window, type clc
- To abort a MATLAB computation, type ctrl-c
- To continue a line, type . . .

HELP:

To view the online documentation, select MATLAB Help from Help menu or MATLAB Help directly in the Command Window. The preferred method is to use the Help Browser. The Help

Browser can be started by selecting the ? icon from the desktop toolbar. On the other hand, information about any command is available by typing

```
>> help Command
```

EXPERIMENT NO: 1

1. VERIFICATION OF NETWORK THEOREMS
2. SUPERPOSITION THEOREM.
3. THEVENIN'S THEOREM.
4. 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".

CIRCUIT DIAGRAM:-

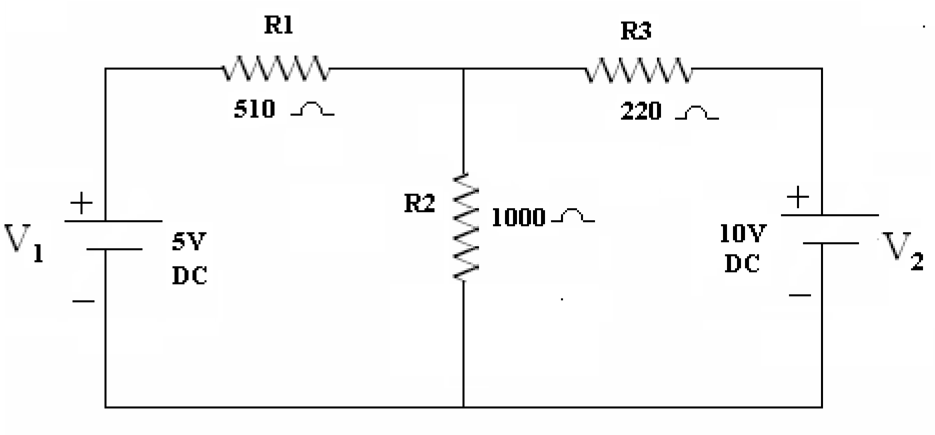


Fig-1

i) Circuit to find current in 1K ohm resistor due to 5V source (short the terminals of 10V source) for fig-2.

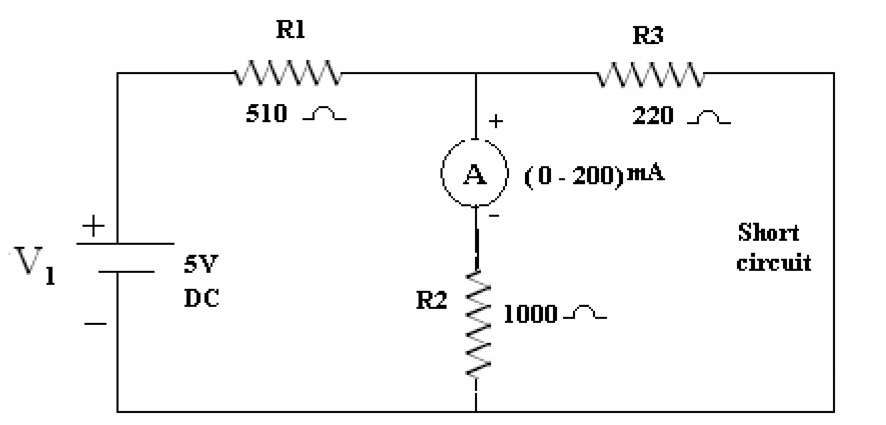


Fig-2

ii) Circuit to find current in 1000ohm resistor due to 10V source (short the terminals of 5V source) for fig-3

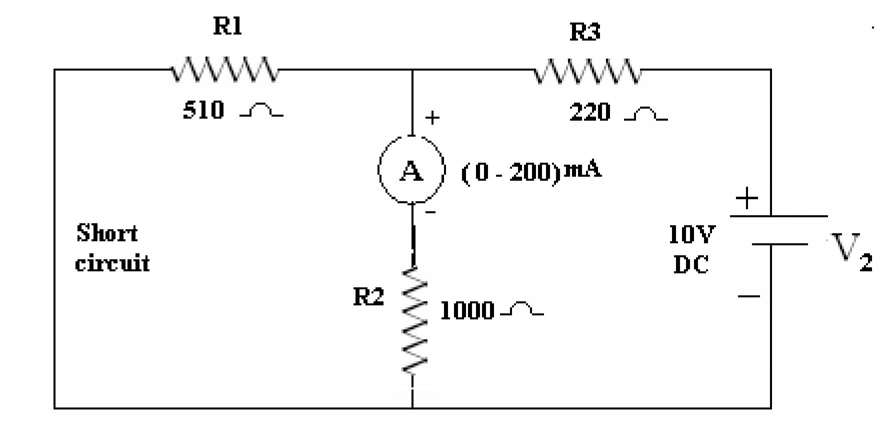


Fig-3

iii) Circuit to find current in 1000ohm resistor due to 10V source & 5V source acting simultaneously for fig-4.

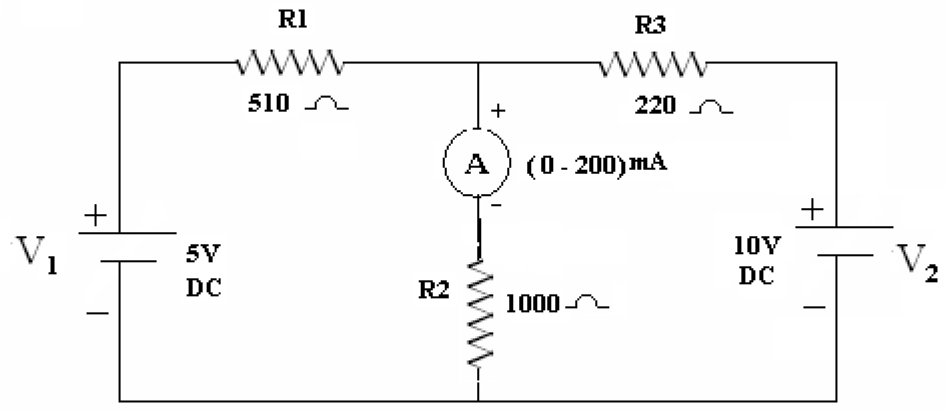


Fig-4

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.

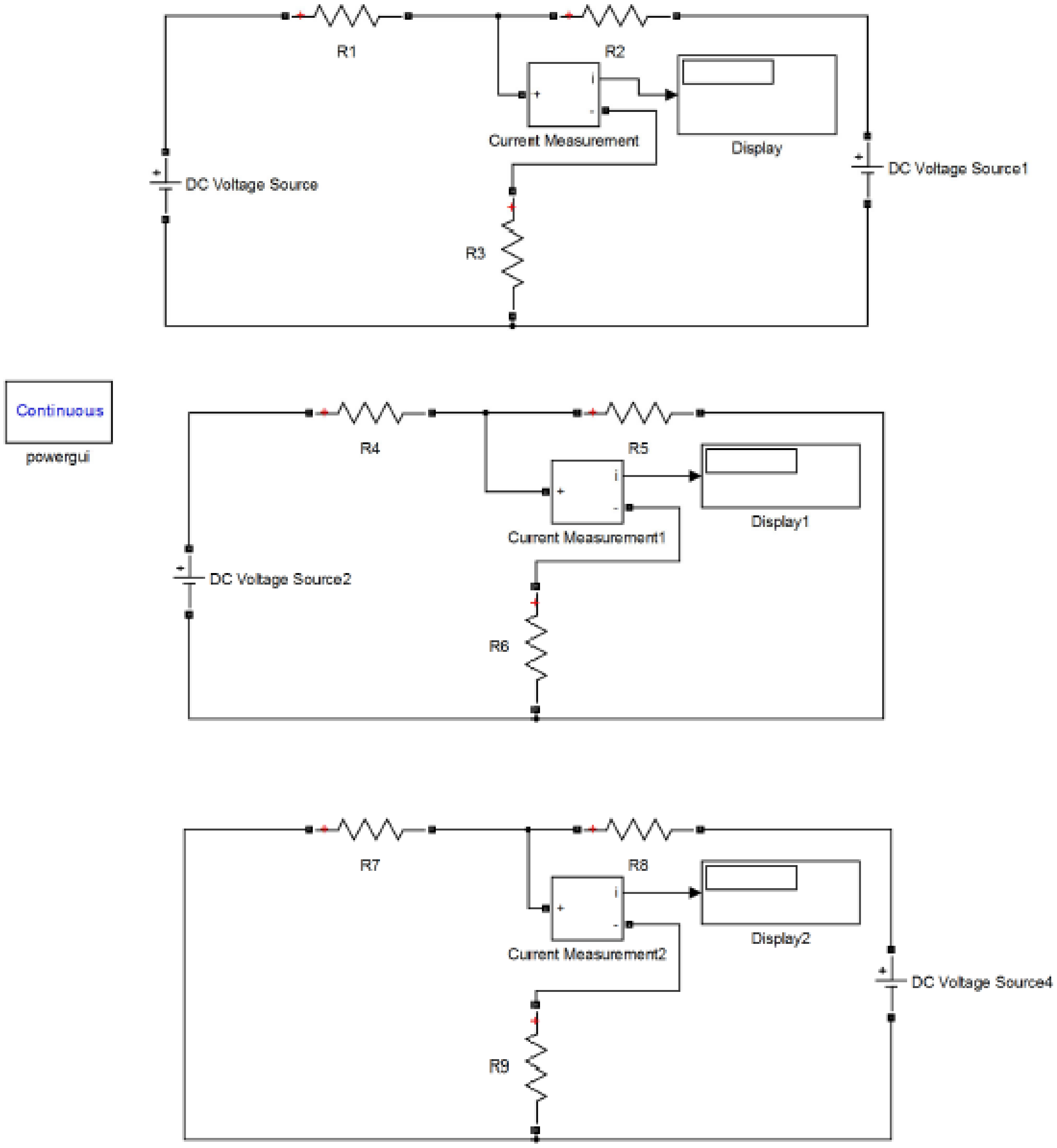
$$I=I_1+I_2+I_3$$

Hence Superposition Theorem is verified.

Observation table :-

	I_1 (mA)	I_2 (mA)	$I = I_1 + I_2$ (mA)
Theoritical			
Practical			

MATLAB IMPLEMENTATION



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.

Circuit Diagrams:

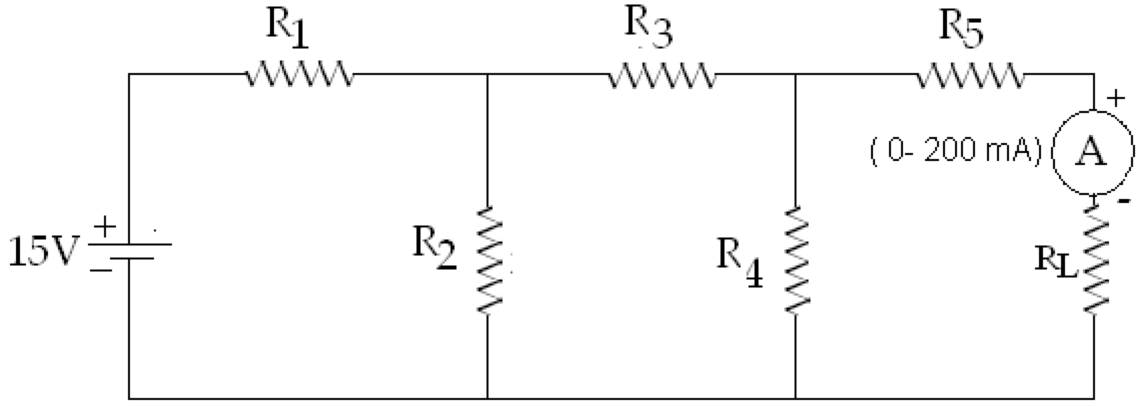


Fig.1

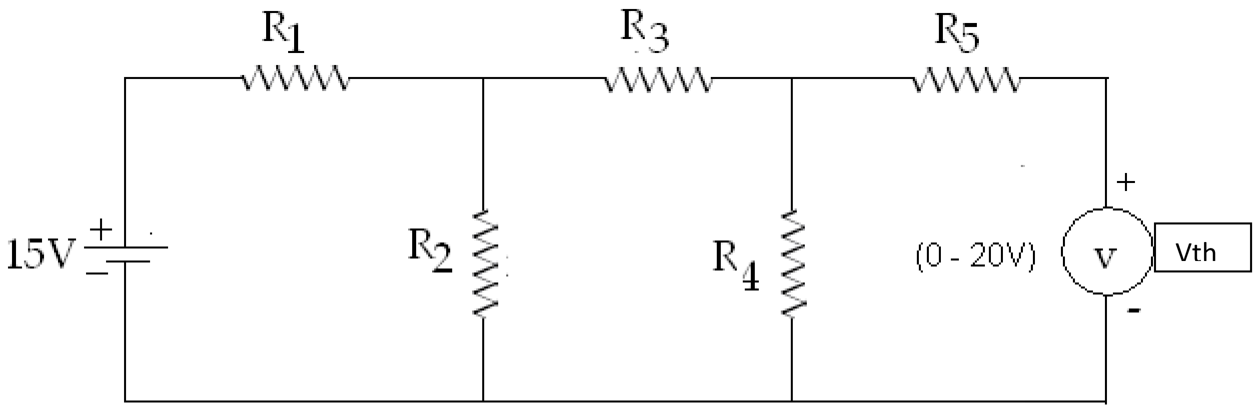


Fig.2

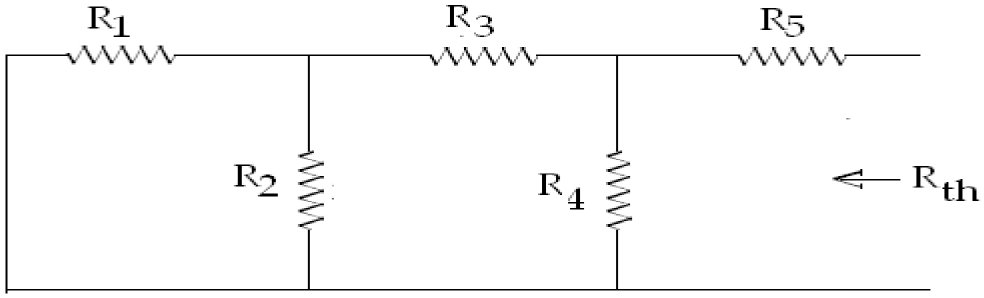


Fig.3

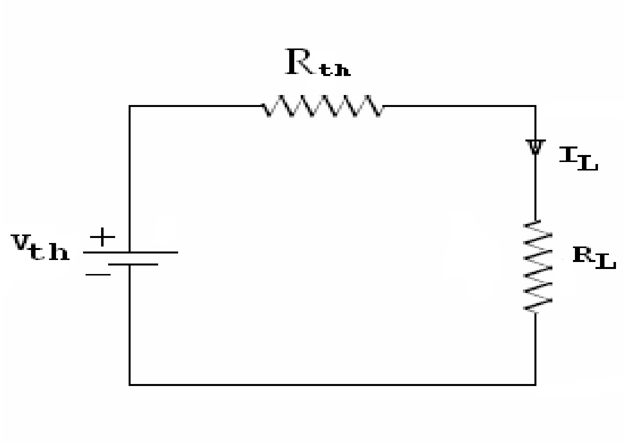
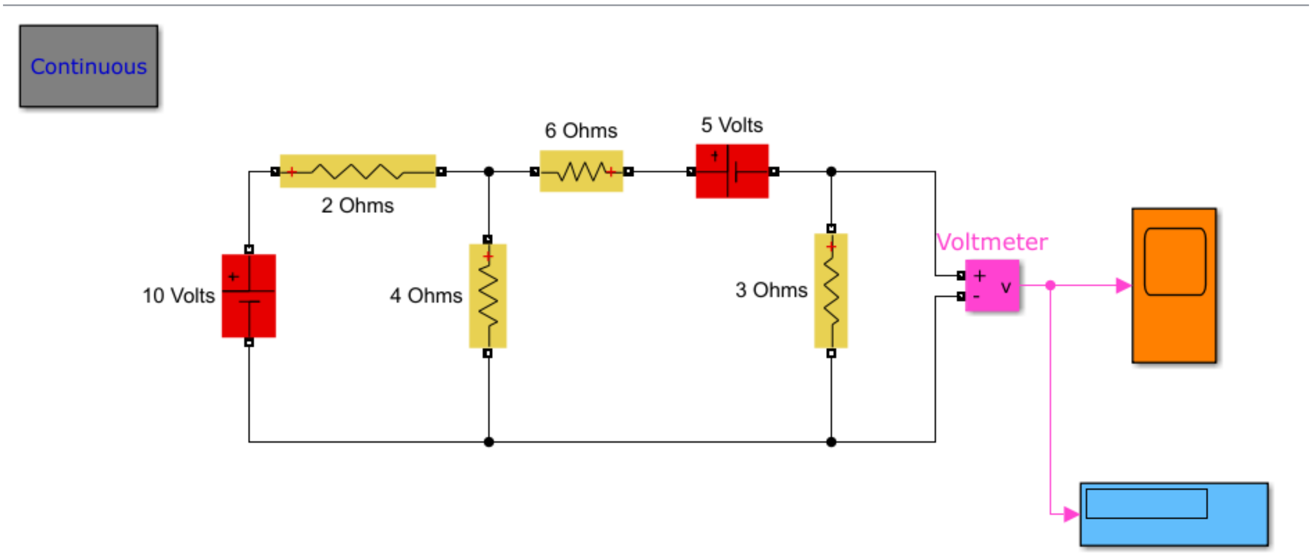


Fig.4

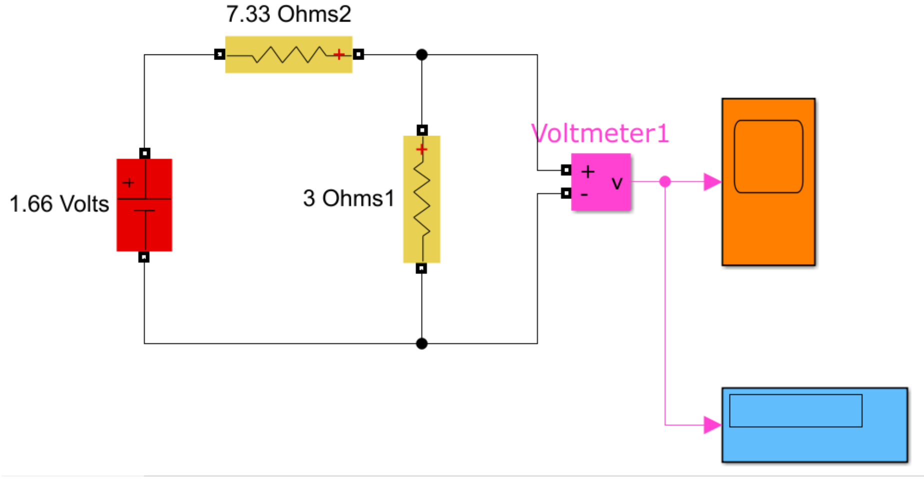
Tabular Form:

	Source Voltage (Vs)	Thevenin’s voltage Vth.	Thevenin’s Resistance Rth	Current(I_{L1}) mA (fig-1)	Current(I_{L2}) mA (fig-4)
Theoretical values					
Practical values					

MATLAB CIRCUIT



Thevinin’s equivalent circuit



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

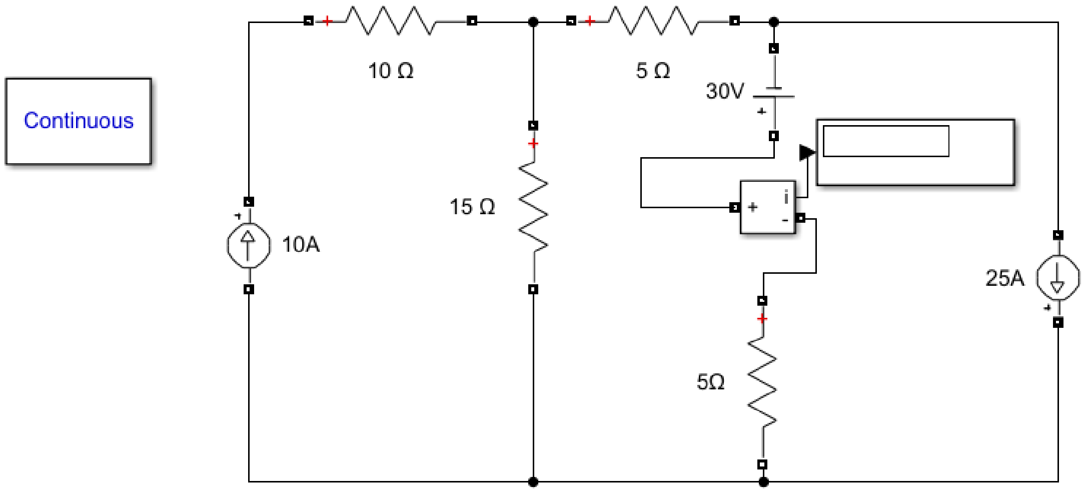
$$I_L = \frac{V_{TH}}{R_{TH}+R_L}$$

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

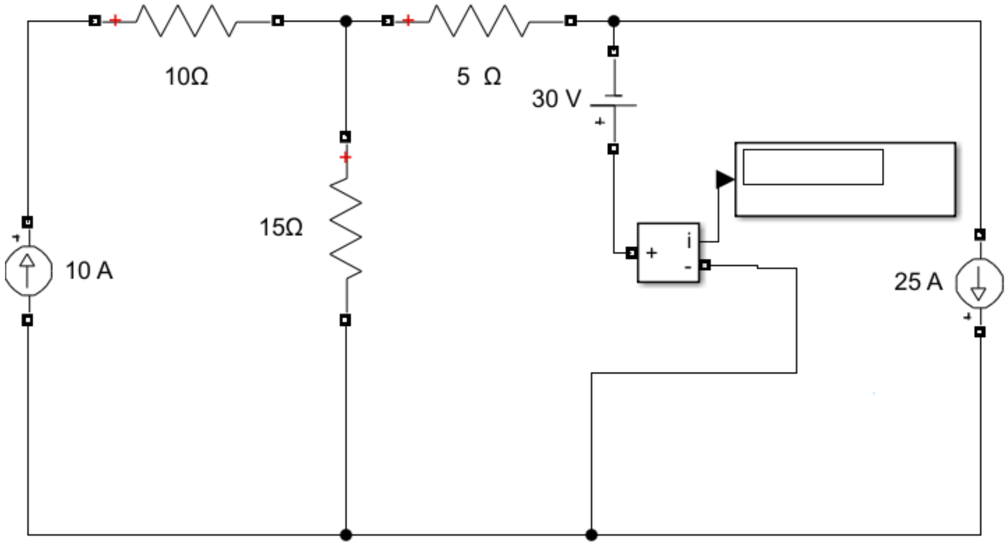
$$I = I_L.$$

Hence Thevenin’s Theorem is Verified.

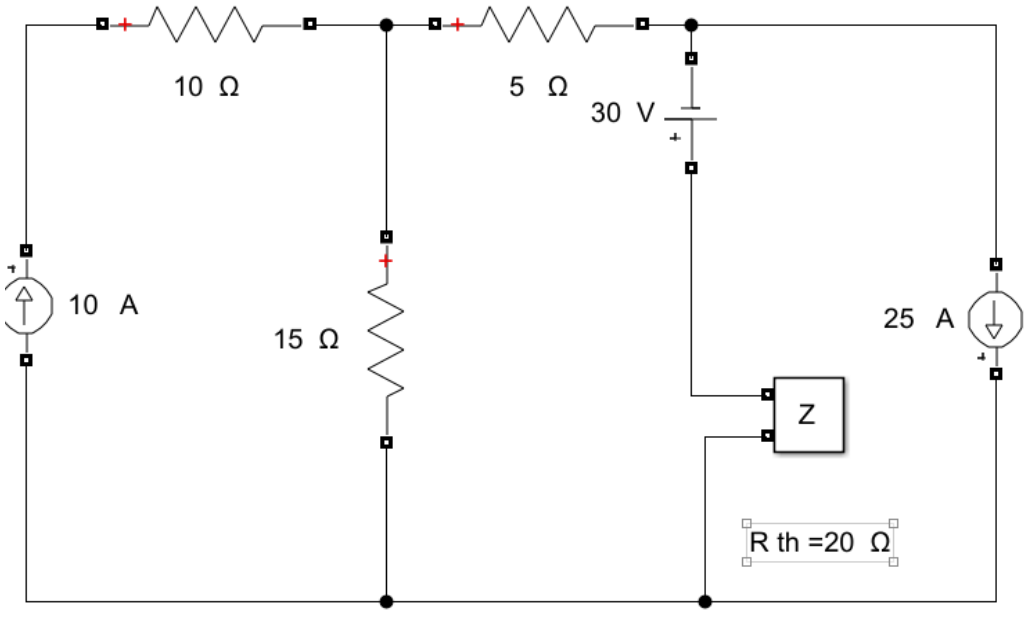
MATLAB IMPLEMENTATION



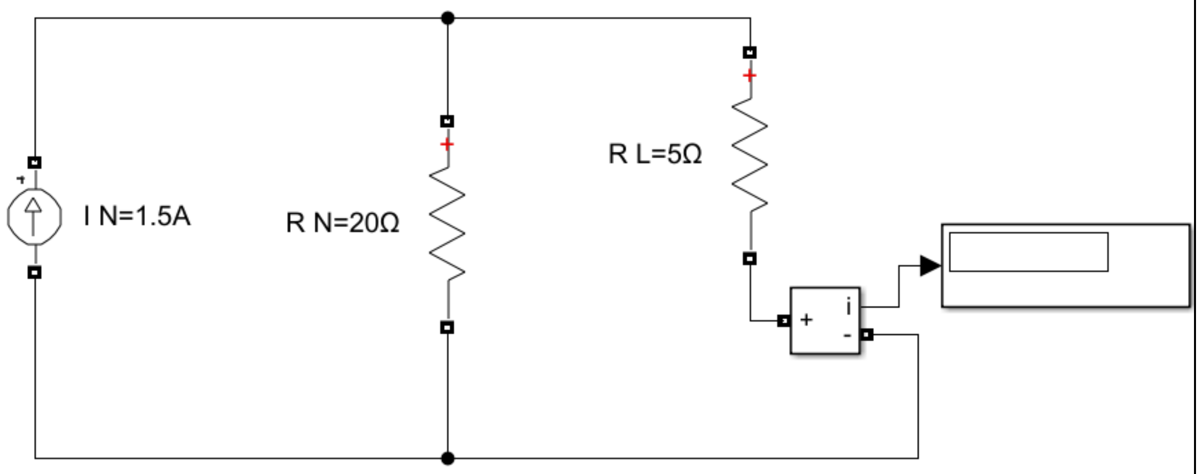
(1)BY CONSIDERING ALL SOURCES



NORTON'S EQUIVALENT CURRENT I N



NORTON'S EQUIVALENT RESISTANCE R N



NORTON'S EQUIVALENT CIRCUIT

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.

   Measure power across the load by considering R_L=R_{TH}.

2. Verify the power for different values of load resistors(i.e. R_L>R_{TH} andR_L<R_{TH}) Measure power by using

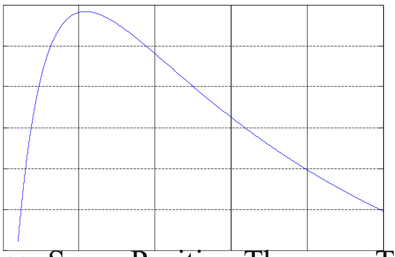
   $$P = \frac{V_{TH}^2}{4R_L}$$

3. Power measured from the above steps results in maximum power dissipation when R_L=R_{TH}.
4. Hence Maximum Power Transfer Theorem is verified

Program CODE

```
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  >');
```

GRAPH



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.

Maximum Power Transfer Theorem Definition

> *Maximum power transfer theorem states that maximum power output is obtained when the load resistance R_L is equal to Thevenin resistance R_{th} as seen from load Terminals.*

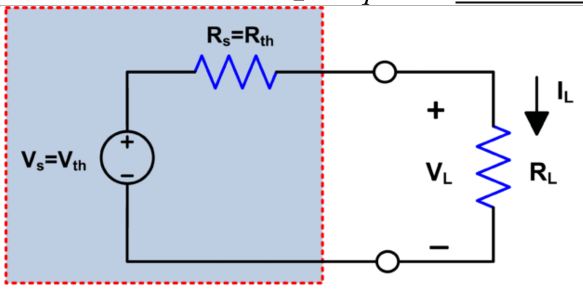


Fig.1: Maximum Power Transfer Theorem

Any circuit or network may be represented by a Thevenin equivalent circuit. **The Thevenin resistance R_{th} is comparable to a source internal resistance (R_S) which absorbs some of the power available from the ideal voltage source**. In above figure, a variable load resistance R_L is connected to a Thevenin circuit. The current for any value of load resistance R_L is connected to a Thevenin circuit. The current for any value of load resistance is;

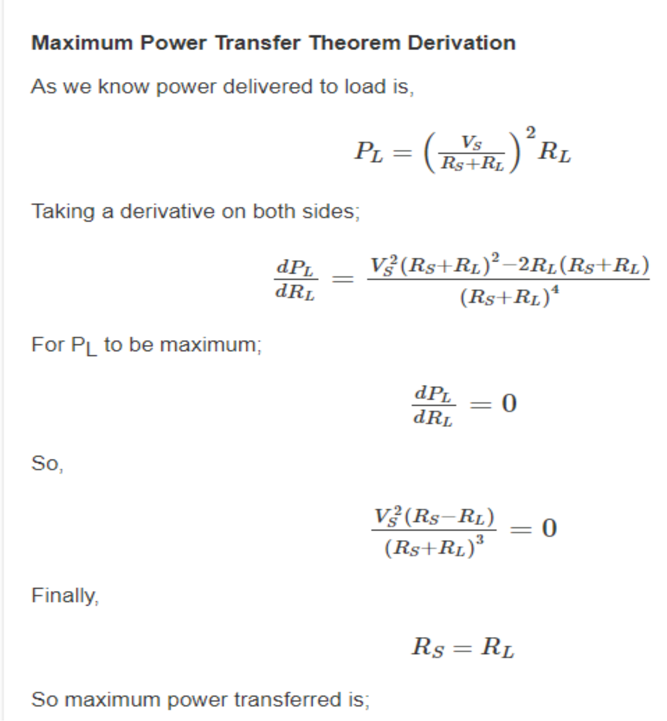
$$I_L=V_SR_S+R_LI_L=V_SR_S+R_L$$

Then by using I^2R , the power delivered to the load is,

$$P_L=(V_SR_S+R_L)^2R_L \cdots \quad (1$$

$$)P_L=(V_SR_S+R_L)^2R_L \cdots \quad (1)$$

The load power depends on both R_{th} (R_S) and R_L; however, R_{th} (R_S) is considered constant for any particular network. Then one might get an idea of how P_L varies with a change in R_L by assuming values for Thevenin circuit of above figure and, in turn, calculating P_L for different values of R_L.



Maximum Power Transfer Theorem Derivation

As we know power delivered to load is,

$$P_L = \left(\frac{V_S}{R_S+R_L}\right)^2 R_L$$

Taking a derivative on both sides;

$$\frac{dP_L}{dR_L} = \frac{V_S^2 (R_S+R_L)^2 - 2R_L(R_S+R_L)}{(R_S+R_L)^4}$$

For P_L to be maximum;

$$\frac{dP_L}{dR_L} = 0$$

So,

$$\frac{V_S^2(R_S-R_L)}{(R_S+R_L)^3} = 0$$

Finally,

$$R_S = R_L$$

So maximum power transferred is;

Maximum Power Transfer Theorem Derivation

As we know power delivered to load is,

PL=(VSRS+RL)2RLPL=(VSRS+RL)2RL

Taking a derivative on both sides;

dPLdRL=V2S(RS+RL)2−2RL(RS+RL)(RS+RL)4dPLdRL=VS2(RS+RL)2−2RL(RS+RL)(RS+RL)4

For P_L to be maximum;

dPLdRL=0dPLdRL=0

So,

V2S(RS−RL)(RS+RL)3=0VS2(RS−RL)(RS+RL)3=0

Finally,

RS=RLRS=RL

So maximum power transferred is;

Pmax=V2S4RSPmax=VS24RS

We got above expression by substituting $R_S=R_L$ into equation (1).

Maximum Power Transfer and Efficiency of Transmission

We observe that power transfer from a real source always produces ohmic heating in the source resistance. Calculations of such internal effects require information about the internal structure and cannot, in general, be based upon Thevenin or Nortonequivalent networks. However, the entire load current i_L usually passes through the internal resistance of a real source, so we represent the internal conditions by lumped parameters as shown in figure 1. The resulting internal power dissipated by R_{TH} or $R_S$is then

PS=RSI2L=RS(RS+RL)2∗V2SPS=RSIL2=RS(RS+RL)2∗VS2

The dashed curve in figure 2 shows that P_S steadily decreases as R_L increases and that $P_S=P_L$ when $R_L/R_S=1$.

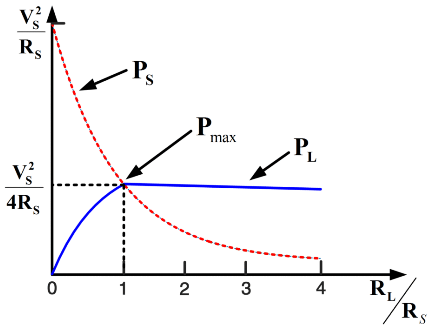


Fig.2: Maximum Power Transfer and Transmission Efficiency

Since the total power generated by the source is P_L+P_S, the wasted internal power $P_S$should be small compared to P_L for efficient operation. Formally, we define the power-transfer efficiency as

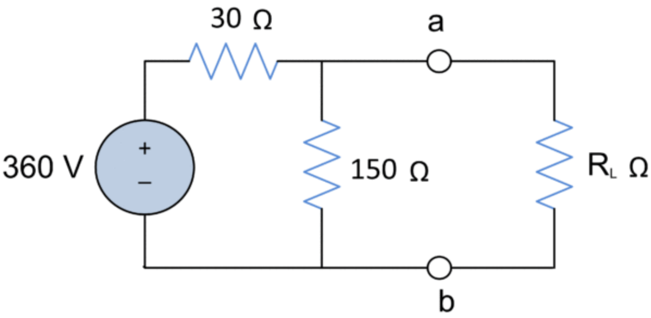
Efficiency=PLPL+PSEfficiency=PLPL+PS

Which is often expressed as a percentage. If the load has been matched for maximum power transfer, then P_S=P_L, and so efficiency,

Efficiency=PL2PL=50 %Efficiency=PL2PL=50 %

Moreover, with R_L=R_S, the terminal voltage drops to $V=V_{TH}/2$. Clearly, electrical utilities would not, and should not, strive for maximum power transfer. Instead, they seek higher power-transfer efficiency by making P_S as small as possible.

Maximum Power Transfer Solved Example



Find R_L

Solution

Let's find V_{th} first across 150 Ω resistance

Vth=VS=360∗150150+30Vth=VS=360∗150150+30

Vth=VS=300 VVth=VS=300 V

To find R_{th} or R_S, short circuit the voltage source

Rth=RS=150 || 30=25ΩRth=RS=150 || 30=25Ω

So, for maximum power transfer, we know that

RL=Rth=25 ΩRL=Rth=25 Ω

Now, Find Maximum power transfer to the load

Pmax=V2S4RS=900 WPmax=VS24RS=900 W

Maximum Power Transfer Theorem using Matlab Code

Here is the MATLAB code to implement maximum power transfer theorem in Matlab.

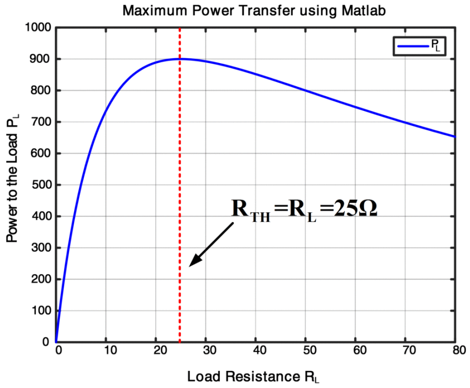
```
clear all;close all;clc

%% Circuit Parameters as given in the example (text)
```

```
% Matlab Code for Maximum Power Transfer Theorem
V_TH = 300; % Thevenin's Voltage
R_TH = 25; % Thevenin's Equivalent Resistance
R_L = 0:0.5:80; % Load Resistance
%%
%% Load Current &amp; Power Calculation
IL = V_TH./(R_TH + R_L); % Load Current
P_L = IL.^2 .* R_L; % Load Power
%%
% As we know that maximum power transfer occurs when R_TH=R_L
%% Plotting the Results
plot(R_L,P_L,'b')
hold on
title('Maximum Power Transfer using Matlab');
xlabel('Load Resistance R_L');
ylabel('Power to the Load P_L');
gtext('R_TH = R_L = 25 Ohm')
legend('P_L')
grid on
```

Result

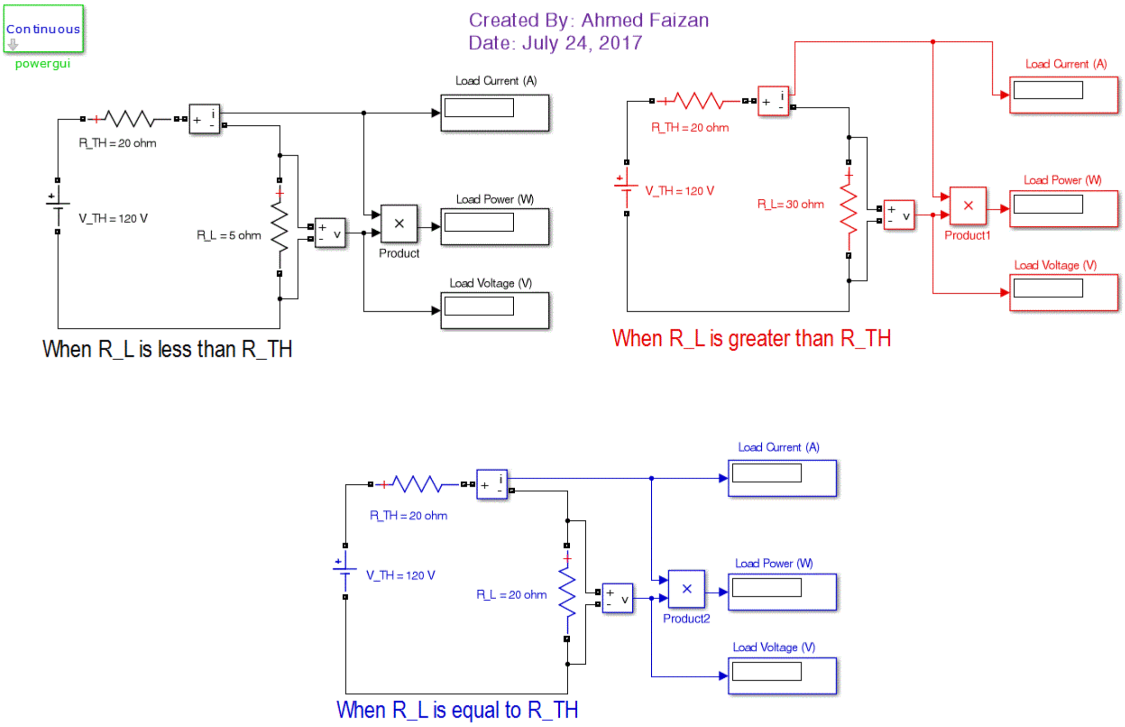
Here is a graph which clearly shows that maximum power transfer occurs when R_{th}=R_L.



Maximum Power Transfer Theorem using Matlab Simulink

Here, a Matlab Simulink model has been developed for three different cases:

1. When $R_L>R_{TH}$
2. When $R_L=R_{TH}$
3. When $R_L<R_{TH}$



To download and run the model, Click Here.

Maximum Power Transfer Theorem Application

When do we want maximum power transfer?

Primarily, in applications where voltage and current signals are used to convey information rather than to deliver large amounts of power. For instance, the first stage of a radio or television receiver should get as much power as possible out of the information-bearing signals that arrive via antenna or cable. Those tiny signals account for only a small fraction of the total power consumed by the receiver, and so power-transfer efficiency is not a significant concern.