

SIMULATION PRACTICE ON MATLAB MANUAL



For 4th SEMESTER, DIPLOMA ELECTRICAL Students

Name of the Student _____

Roll No of the student _____

Laboratory Section _____

[This laboratory manual will guide you to perform the necessary experiments to meet the fulfillment of SCTE&VT syllabus for **SIMULATION PRACTICE ON MATLAB MANUAL Pr.3** for 4th semester Electrical students. This manual also consists of the supplementary materials like data sheets for some useful components to be used throughout the course.]

TABLE OF CONTENTS

RULES & REGULATIONS	2
FORMAT FOR RECORD WRITING	3
MARK DISTRIBUTION	4
LIST OF EXPERIMENTS.....	5
EXPERIMENT NO. -1(A)	6
EXPERIMENT NO. -1 (B).....	9
EXPERIMENT NO. - 2	11
EXPERIMENT NO. - 3	16
EXPERIMENT NO. - 4	20
EXPERIMENT NO. - 5	23
EXPERIMENT NO. - 6	25
EXPERIMENT NO. - 7	28
EXPERIMENT NO. – 8(A).....	30
EXPERIMENT NO. – 8(B).....	32
EXPERIMENT NO. – 8(C).....	34
EXPERIMENT NO. - 9	36
EXPERIMENT NO. – 10	38

RULES & REGULATIONS

1. **Don't be late** to the lab.
2. Put your signature on attendance sheet first, submit the record and then go to your place correspondence to experiment kit.
3. **Without fair record and lab manual** you are **not allowed** to the lab.
4. Handlings of other things, which are not related to the experiment, are strictly prohibited inside the lab.
5. You may be charged with full cost and additional penalty for destroying the equipments.
6. You should put on your own identity card.
7. Till the time you are inside the lab you should stick to your place and don't move here & there.
8. Before leaving the lab, be sure that power supply is switched off and you have returned the components proper manner otherwise all the group members will be punished.
9. Your duly filled lab manual along with stressing sheet must be verified by any of the instructors present there else your record will not be considered.
10. You yourself will be responsible for exchange or missing of any precious materials.
11. You can demand for marks on the basis of performance ONLY, but can't beg.
12. **Extra labs may be allowed** on your request if your application is forwarded through proper channel **within the time limit**. Beyond the time limit it is never accepted at any situation.

FORMAT FOR RECORD WRITING

1. Maintain the contain page with date of performing the experiment, experiment name, page number.
2. Put the page numbers on the top of the individual page, Date & experiment no.
3. Write the aim of the experiment, apparatus required short theoretical description with required circuit/block diagram, observation & conclusion.
4. Attached the tracing sheet to the left side of the paper.
5. Put your full signature at bottom right corner of the last page of that experiment.
6. Record should be very neat and clean and the entire circuits/block diagram should be drawn with **pencil only**.
7. **Bad hand** writing will **affect** the record mark.
8. **Incomplete record** will lead towards **negative** marking
9. **Data on lab manual** should match with fair record.
10. Arrange the pages properly according to the experiment list (ascending order).
11. Use the record and internal pages **without rolling (SUPPLIED format ONLY)**.
12. Don't submit loose sheets without record at any condition.

FORMAT

1. Aim of the Experiment
2. Software Requirements
3. Theory
4. Circuit simulation diagram /block diagram (if any)
5. Procedure
6. Program
7. Output
8. Conclusion (On the basis of experiment output)

MARK DISTRIBUTION

Sl. No.	Parts	Marks
1	Attendance	20
2	Record	20+10
3	Experiment	30
4	Via-voice/ Quiz	20
	TOTAL	100

I hereby Mr./Ms.....with roll no..... have understood the above rules and regulations of the **Simulation Practice On Mat lab** and **will accept any kind of punishment given by the instructor for violating the rules** mentioned above.

Date:

Signature of the student

LIST OF EXPERIMENTS

EXPT. NO.	LIST OF EXPERIMENT	PAGE NO.	DATE	REMARK
1.	1(A)MATLAB Software, Basic Features, Tool Box And Different Functions	6-8		
	1.(B) To Learn Arithmetic, Relational And Logic Operator	9-10		
2.	Matrix Formation And Its Manipulation	11-15		
3.	Vector Manipulation	16-19		
4.	To Get Familiar With Plotting Commands Used In MATLAB	20-22		
5.	To Plot A Unit Circle	23-24		
6.	To Plot Signals Using Impulse, Step, Ramp Functions	25-26		
7.	To Plot A Sine And Cosine Wave	28-29		
8.	8(A)Verification Of Superposition Theorem	30-31		
	8(B)Verification Of Thevenin's Theorem	32-33		
	8(C)Verification Of Norton's Theorem	34-35		
9.	Simulation Of A Half Wave Uncontrolled Rectifier	36-37		
10.	Simulation Of 1-Phase Full Bridge Controlled Rectifier	38-39		

EXPERIMENT NO-1(A)

AIM OF THE EXPERIMENT:-

To Understand MATLAB Software, Basic Features, Tool Box And Different functions Such Algebraic, Trigonometric And Exponential function used in MATLAB

SOFTWARE REOURIED:-

1. MATLAB 7.12.0 (R2011a) SOFTWARE
2. Personal Computer loaded with Windows 7/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. 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.

Features of MATLAB:-

1. It is a high-level language for numerical computation, visualization and application development.
 2. It also provides an interactive environment for iterative exploration, design and problem solving.
 3. It provides vast library of mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, numerical integration and solving ordinary differential equations.
 4. It provides built-in graphics for visualizing data and tools for creating custom plots.
- MATLAB's programming interface gives development tools for improving code quality, maintainability, and maximizing performance.
5. It provides tools for building applications with custom graphical interfaces.
 6. It provides functions for integrating MATLAB based algorithms with external applications and languages such as C, Java, .NET and Microsoft Excel.

Different windows of MATLAB desktop:-

MATLAB has the following four basic window normally and other windows are also available:=

1. The Command Window
2. The editor window
3. The Workspace
4. The graphic window.

The desktop has the following windows:-

Command Window - This is the main area where commands can be entered at the command line. It is indicated by the command prompt (>>).

Editor window:-Here we can write, edit ,create and save our own program in file with “m .files” extension. we can use any text editor to carry out these tasks.

Workspace - The workspace shows all the variables created and/or imported from files.

Graphic Window :- Graphic window is also known as figure window and it is the display of the figure result of the program . the user can create as many figure window of the system memory will allow.

Command History - This panel shows or rerun commands that are entered at the command line.

Current Directory - show the files in Current Directory

Current Folder - This panel allows you to access the project folders and files.

Different Toolbox in MATLAB:-

Tools box are the software device of MATLAB. It has number of tool boxes working in different areas of engineering. They are as follows:-

1. Communications System Toolbox.
2. SimPower System Toolbox.
3. Control System Toolbox.
4. DSP System Toolbox
5. Simspace Toolbox
6. Robust Control Toolbox
7. Fuzzy Logic Toolbox
8. Instrument Control Toolbox

Different function in MATLAB:-

Algebraic function:-

Matrix function

- 1.det- determinant of a square matrix.
- 2.rank-rank of a matrix
- 3.trace-sum of diagonal element

linear equation function

1. inv- inverse of a matrix
2. linsolve-solve linear system equation

eigen value and singular value function

- 1.eig- eigenvalues and eigenvectors
- 2.eigs- largest eigenvalues and eigenvectors
- 3.poly- characteristic polynomial
- 4.sqrtm-matrix square root

Trigonometric function:-

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)

Exponential Function:-

- 1.exp - exponential
- 2.log - natural logarithm
- 3.abs - absolute value
- 4.sqrt - square root
- 5.rem – remainder
- 6.power-array power

CONCLUSION:-

From the above experiment we have study about MATLAB Software, Basic Features, Tool Box and Different Functions Such Algebraic, Trigonometric and Exponential function present in MATLAB. _

Date:**Signature of the Student****Name:****Roll no.:****Semester:****Group no.****Date:****Instructor's Signature**

EXPERIMENT NO-1(B)

AIM OF THE EXPERIMENT:-

To get familiar with different operator such as Arithmetic, Relational and Logic operator used in MATLAB

SOFTWARE REOURIED:-

1. MATLAB 7.12.0 (R2011a) SOFTWARE
2. Personal Computer Loaded with Windows 7/XP SP2.

THEORY:-

An operator is a symbol that tells the compiler to perform specific mathematical or logical manipulations. MATLAB is designed to operate primarily on whole matrices and arrays. Therefore, operators in MATLAB work both on scalar and non-scalar data.

MATLAB allows the following types of elementary operations:-

Arithmetic operators:-

MATLAB allows two different types of arithmetic operations:

1. Matrix arithmetic operations
2. Array arithmetic operations

Matrix arithmetic operations are same as defined in linear algebra. Array operations are executed element by element, both on one-dimensional and multidimensional array.

The matrix operators and array operators are differentiated by the period (.) symbol. However, as the addition and subtraction operation is same for matrices and arrays, the operator is same for both cases. The following table gives brief description of the operators

operator	Description	operator
A+B	Binary addition -	plus(A,B)
+A	Unary plus	uplus(A)
A-B	Binary subtraction	minus(A,B)
-A	Unary minus	uminus(A)
A*B	Matrix multiplication	mtimes(A,B)
A.*B	Array-wise multiplication	times(A,B)
A/B	Matrix right division	mrdivide(A,B)
A./B	Array-wise right division	rdivide(A,B)
A\B	Matrix left division	ldivide(A,B)
A.\B	Array-wise left division	ldivide(A,B)
A^B	Matrix power	mpower(A,B)
A.^B	Array-wise power	power(A,B)
A'	Complex transpose	ctranspose(A)

Relational operators:-

Relational operators can also work on both scalar and non-scalar data. Relational operators for arrays perform element-by-element comparisons between two arrays and return a logical array of the same size, with elements set to logical 1 (true) where the relation is true and elements set to logical 0 (false) where it is not

operator	Description
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
==	Equal to
~=	Not equal to

Logic operator:-

MATLAB offers two types of logical operators and functions:

Element-wise - These operators operate on corresponding elements of logical arrays. Element-wise logical operators operate element-by-element on logical arrays. The symbols &, |, and ~ are the logical array operators AND, OR, and NOT.

Short-circuit - These operators operate on scalar and logical expressions. Short-circuit logical operators allow short-circuiting on logical operations. The symbols && and || are the logical short-circuit operators AND and OR.

operator	Description
&	Logical or element-wise AND
/	Logical or element-wise OR
~	logical or element-wise NOT
&&	Short-circuit AND
//	Short-circuit OR

CONCLUSION:-

From the above experiment we have study about different operators such as Arithmetic, Relational and Logic operator present in MATLAB.

Date:**Signature of the Student****Name:****Roll no.:****Semester:****Group no.****Date:****Instructor's Signature**

EXPERIMENT NO. - 2

AIM OF THE EXPERIMENT:-

Generate a matrix and perform some basic operation on matrices such as addition, subtraction, multiplication and special matrix functions using MATLAB Software.

SOFTWARE REQUIRED:-

1. MATLAB 7.12.0 (R2011a) SOFTWARE
2. Personal Computer Loaded with Windows 7/XP SP2.

THEORY:-

A matrix is a two-dimensional array of numbers. Matrices can be defined by separating the elements of a row with blank space or comma and using a semicolon to terminate each row. The list of elements should be surrounded by square brackets `[]`. Parentheses `()` are used to access elements and sub arrays (they are also used to denote a function argument list).

Matrix Functions:-

eye - identity matrix
zeros - matrix of zeros
ones - matrix of ones
diag - extract diagonal of a matrix or create diagonal matrices
triu - upper triangular part of a matrix
tril - lower triangular part of a matrix
rand - randomly generated matrix
Size-size of a matrix
det- determinant of a square matrix.
inv- inverse of a matrix
rank-rank of a matrix

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:-

```
%arithmetic operation of matrix  
clc;  
clear all;
```

```
close all;
a=[3 2 1;0 1 2;3 0 1]
b=[2 1 0;2 0 0;1 2 1]
c=a+b
d=a-b
e=a*b
f=5*a
g=a.*b
h=a/b
i=a/2
j=a./b

%special matrix function
clc;
clear all;
close all;
k=[2 4 6;1 3 7;5 8 9]
l=det(k)
m=inv(k)
n=rank(k)
o=size(k)
p=diag(k)
q=triu(k)
r=tril(k)
s=eye(3,3)
t=zeros(4,3)
u=ones(3,3)
v=rand(3,4)
```

OUTPUT:-

%arithmetic operation with matrix

```
a =
     3     2     1
     0     1     2
     3     0     1

b =
     2     1     0
     2     0     0
     1     2     1
```

c =

5	3	1
2	1	2
4	2	2

d =

1	1	1
-2	1	2
2	-2	0

e =

11	5	1
4	4	2
7	5	1

f =

15	10	5
0	5	10
15	0	5

g =

6	2	0
0	0	0
3	0	1

h =

-0.0000	1.0000	1.0000
-3.0000	2.0000	2.0000
-2.0000	3.0000	1.0000

i =

1.5000	1.0000	0.5000
0	0.5000	1.0000
1.5000	0	0.5000

j =

1.5000	2.0000	Inf
0	Inf	Inf
3.0000	0	1.0000

k =

2 4 6

1 3 7

5 8 9

l = 4.0000

m =

-7.2500 3.0000 2.5000

6.5000 -3.0000 -2.0000

-1.7500 1.0000 0.5000

n = 3

o = 3 3

p =

2

3

9

q =

2 4 6

0 3 7

0 0 9

r =

2 0 0

$$\begin{bmatrix} 1 & 3 & 0 \\ 5 & 8 & 9 \end{bmatrix}$$

s =

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

t =

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

u =

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

v =

$$\begin{bmatrix} 0.8143 & 0.3500 & 0.6160 & 0.8308 \\ 0.2435 & 0.1966 & 0.4733 & 0.5853 \\ 0.9293 & 0.2511 & 0.3517 & 0.5497 \end{bmatrix}$$

CONCLUSION:-

From the above experiment we have generate matrices and performed some basic operation on matrices such as addition, subtraction, multiplication and special matrix function using MATLAB Software.

EXPERIMENT NO. - 3

AIM OF THE EXPERIMENT:-

Create a vector using linspace and perform some basic vector operation on such as addition, subtraction, multiplication. using MATLAB Software.

SOFTWARE REOURIED:-

1. MATLAB 7.12.0 (R2011a) SOFTWARE
2. Personal Computer Loaded with Windows 7/XP SP2.

THEORY:-

Vector: - is a one-dimensional array of numbers. MATLAB allows creating two types of vectors: - Row vectors and Column vectors

Row vectors: - are created by enclosing the set of elements in square brackets, using space or comma to delimit the elements.

Column vectors: - are created by enclosing the set of elements in square brackets, using semicolon (;) to delimit the elements.

Linear spacing: - is a command to generate linearly spaced vectors.

Syntax:-variable=linspace(start number, end number, number of element)

Example: - theta = linspace(0,2*pi,50)

1. Scaler function:-

sin - trigonometric sine

cos - trigonometric cosine

tan - trigonometric tangent

asin - trigonometric inverse sine (arcsine)

acos - trigonometric inverse cosine (arccosine)

atan - trigonometric inverse tangent (arctangent)

.exp - exponential

.log - natural logarithm

.abs - absolute value

.sqrt - square root

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

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:-

%different method for creating row & column vector

```
clc;
clear all;
a=[1 2 3 4 5 ]
b=[1,2,3,4,5]
c=1:5
%generate vector when range is given with interval 2
d=1:2:10
%generate vector when range is given with interval -3
e=10:-3:1
% creating column vector
f=[5;6;5;6]
```

% creating row vector using linspace

```
clc
clear all
g=linspace(1,4,5)
h=linspace(0,2*pi,5)
```

%arithmetic operation with vector

```
clc
clear all
a=[1 3 5 7]
b=[2 4 6 8]
c=a+b
d=a-b
e=a*b'
f=5*a
g=a.*b
h=a/b
i=a/2
j=a./b
```

```
%sin & sqrt function with vector argument
```

```
clc;  
clear all;  
a=1:4  
b=sin(a)  
c=sqrt(a)
```

OUTPUT:-

```
%different method for creating row & column vector
```

```
a = 1 2 3 4 5
```

```
b = 1 2 3 4 5
```

```
c = 1 2 3 4 5
```

```
d = 1 3 5 7 9
```

```
e = 10 7 4 1
```

```
f =
```

```
5
```

```
6
```

```
5
```

```
6
```

```
% creating row vector using linspace
```

```
g = 1.0000 1.7500 2.5000 3.2500 4.0000
```

```
h = 0 1.5708 3.1416 4.7124 6.2832
```

```
%arithmetic operation with vectors
```

```
a = 1 3 5 7
```

```
b = 2 4 6 8
```

```
c = 3 7 11 15
```

```
d = -1 -1 -1 -1
```

e = 100

f = 5 15 25 35

g = 2 12 30 56

h = 0.8333

i = 0.5000 1.5000 2.5000 3.5000

j = 0.5000 0.7500 0.8333 0.8750

%sin & sqrt function with vector argument

a = 1 2 3 4

b = 0.8415 0.9093 0.1411 -0.7568

c = 1.0000 1.4142 1.7321 2.0000

CONCLUSION:-

From the above experiment we have study and performed vector manipulation & functions using MATLAB

Date:

Signature of the Student

Name:

Roll no.:

Semester:

Group no.

Date:

Instructor's Signature

EXPERIMENT NO. - 4

AIM OF THE EXPERIMENT:-

To get familiar with plotting commands used in MATLAB.

SOFTWARE REOURIED:-

1. MATLAB 7.12.0 (R2011a) SOFTWARE
2. Personal Computer Loaded with Windows 7/XP SP2.

THEORY:-

MATLAB provides numerous commands for plotting graphs. The commonly used commands are:-

plot: - It is the command used to plot a continuous signal.

stem:-it is used to draw a discrete signal i.e. Non-continuous spectral amplitude signal

subplot:-it is the command used to create no. of plots in the same figure, each of these plots is called a subplot. *Syntax for the command is subplot(m, n, p)*

where, *m* and *n* are the number of rows and columns of the plot array and *p* specifies where to put a particular plot.

axis equal:- this command allows generating the plot with the same scale factors and the spaces on both axes.

The following table shows some of the commonly used commands for plotting.

Plotting commands:-

Command	Purpose
close	Closes the current plot.
close all	Closes all plots.
Axis	Sets axis limits. <code>axis([xmin xmax ymin ymax])</code>
Grid on/off	Displays gridlines.
Plot	Generates xy plot. Ex-plot(x,y)
stem	Creates stem plot.
subplot	Creates plots in sub windows.
title	-Puts text at top of plot.
xlabel	Adds text label to x-axis.
ylabel	Adds text label to y-axis.
Figure	-Opens a new figure window.
Hold on/off	Freezes/unfreezes current plot.
text	Places string in figure
print	Prints plot or saves plot to a file.
bar	Creates bar chart.

Specifying colors, line styles and markers used in plotting :-

Symbol	Color	Symbol	Line Style	Symbol	Marker
k	Black	-	Solid	+	Plus sign
r	Red	--	Dashed	o	Circle
b	Blue	:	Dotted	/	Asterisk
g	Green	-.	Dash-dot	:	Point
c	Cyan	none	No line	£	Cross
m	Magenta			s	Square
y	Yellow			d	Diamond

PROCEDURE:-

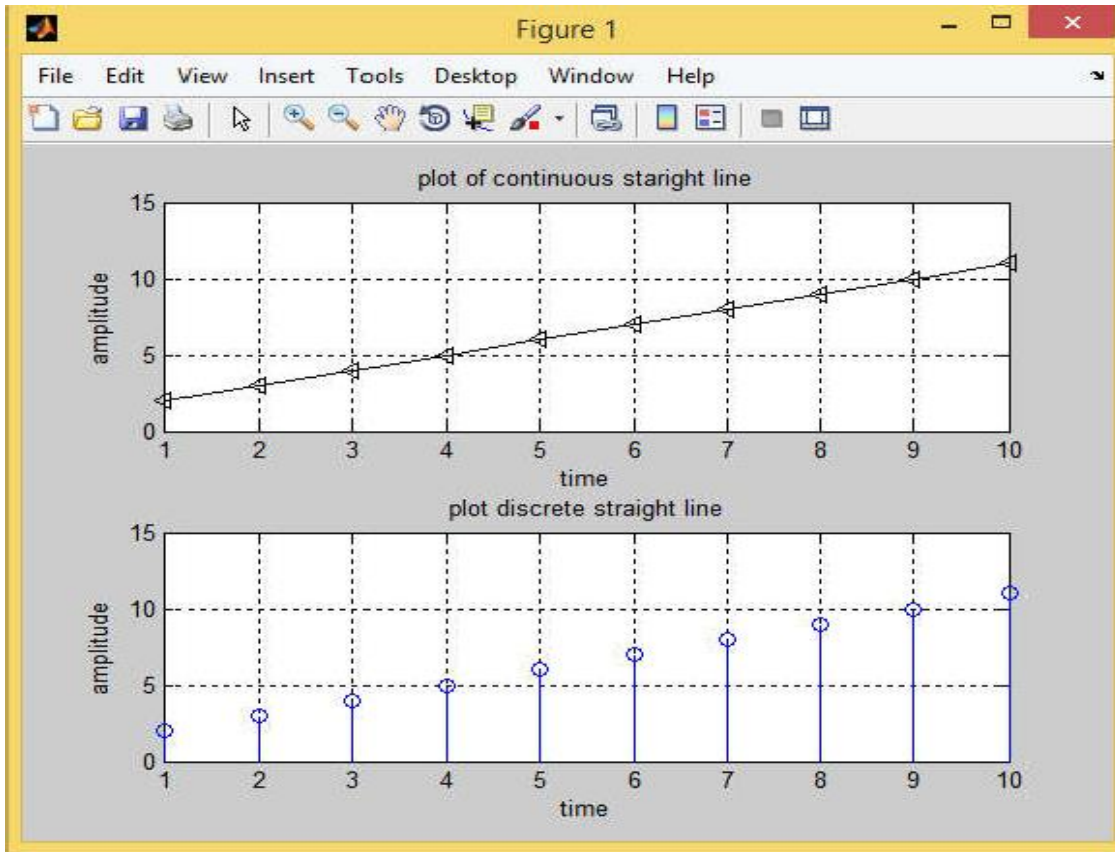
1. Open MATLAB
2. Open new M-file
3. Type the program
4. Save in current directory
5. Compile and Run the program
6. For the output see command window/Figure window

PROGRAM:-

```

clc;
clear all;
close all;
x=1:10;
m=1;c=1;
y=m*x+c;
subplot(2,1,1);
plot(x,y,'k-<');
xlabel('time');
ylabel('amplitude');
title('plot of continuous straight line');
grid on
subplot(2,1,2);
stem(x,y)
xlabel('time');
ylabel('amplitude');
title('plot discrete straight line');
grid on

```

OUTPUT:-**CONCLUSION:-**

from the above experiment we have study about different plotting command provided in MATLAB.

Date:

Signature of the Student

Name:

Roll no.:

Semester:

Group no.

Date:

Instructor's Signature

EXPERIMENT NO. - 5

AIM OF THE EXPERIMENT:-

To plot a circle of unit radius using MATLAB Software

SOFTWARE REQUIRED:-

1. MATLAB 7.12.0 (R2011a) SOFTWARE
2. Personal Computer Loaded with Windows 7/XP SP2.

THEORY:-

A circle is defined as the locus of a point which moves in a plane such that its distance from a fixed point in that plane is always constant.

The equation of circle is: $(x - a)^2 + (y - b)^2 = r^2$

If we place the circle center at (0,0) and set the radius to 1 we get:

$$\begin{aligned}(x - a)^2 + (y - b)^2 &= r^2 \\(x - 0)^2 + (y - 0)^2 &= 1^2 \\x^2 + y^2 &= 1\end{aligned}$$

This is the Equation of the Unit Circle.

PROCEDURE:

1. Open MATLAB
2. Open new M-file
3. Type the program to plot a circle of unit radius
4. Save in current directory
5. Compile and Run the program
6. For the output see command window/Figure window

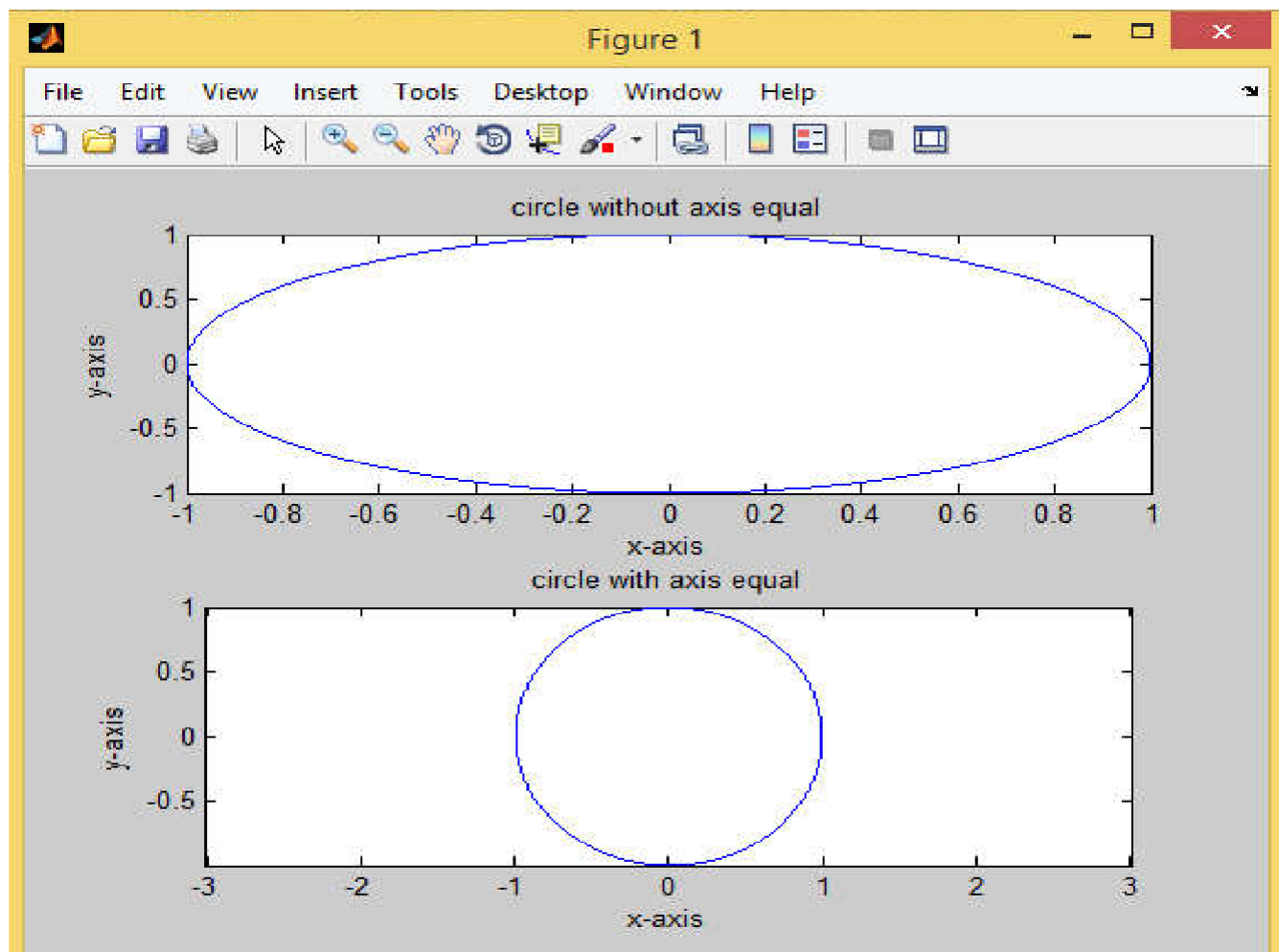
PROGRAM:-

```
%To Plot Circle Of Unit Radius
clc;
clear all;
close all;
theta=(0:0.001:2*pi);
x=sin(theta);
y=cos(theta);
subplot(2,1,1);
plot(x,y);
xlabel('x-axis');
ylabel('y-axis');
```



```
title('circle without axis equal');  
subplot(2,1,2);  
plot(x,y);  
axis equal;  
xlabel('x-axis');  
ylabel('y-axis');  
title('circle with axis equal');
```

OUTPUT:-



CONCLUSION:-

From the above experiment a circle of unit radius is drawn using MATLAB function.

EXPERIMENT NO. - 6

AIM OF THE EXPERIMENT:-

To plot the fundamental signals like unit impulse signal, unit step signal & unit ramp signal using MATLAB.

SOFTWARE REQUIRED:-

1. MATLAB 7.12.0 (R2011a) SOFTWARE
2. Personal Computer Loaded with Windows 7/XP SP2.

THEORY:-

The unit impulse function, unit step function and unit ramp function are considered to be fundamental functions.

1. Unit impulse function:- $\delta(n) = \begin{cases} 1, k = 0 \\ 0, k = \text{any other } k \end{cases}$
2. Unit step function:- $u(n) = \begin{cases} 1, k \geq 0 \\ 0, k < 0 \end{cases}$
3. Unit ramp function:- $r(n) = \begin{cases} k, k \geq 0 \\ 0, k < 0 \end{cases}$

PROCEDURE:-

1. Open MATLAB
2. Open new M-file
3. Type the program of unit impulse, step & ramp function.
4. Save in current directory
5. Compile and Run the program
6. For the output see command window/Figure window

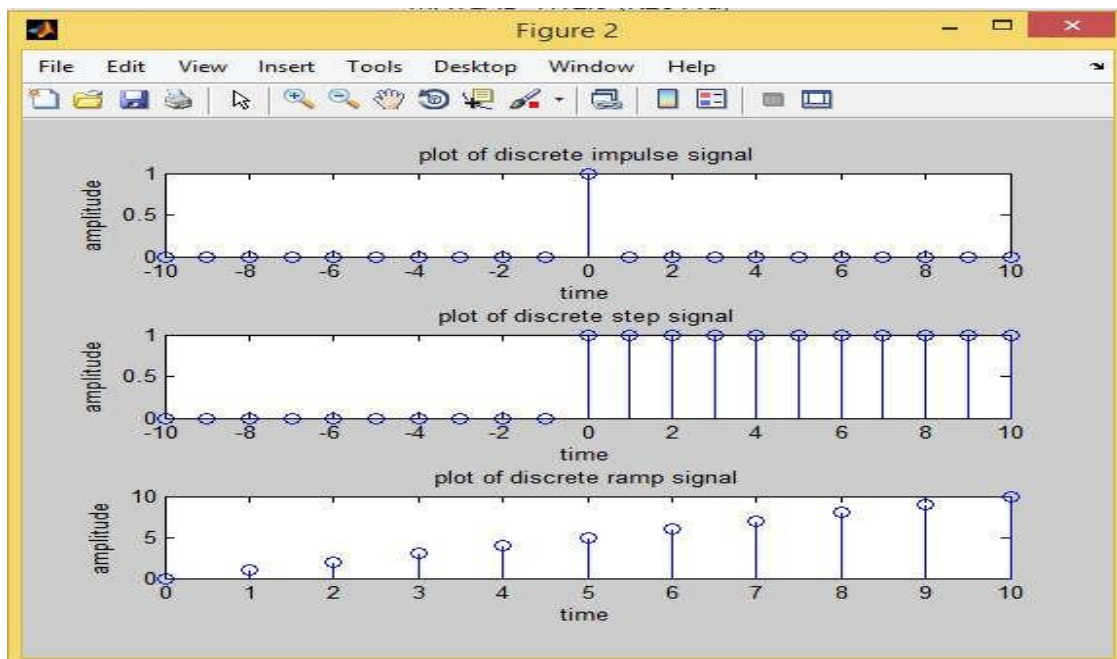
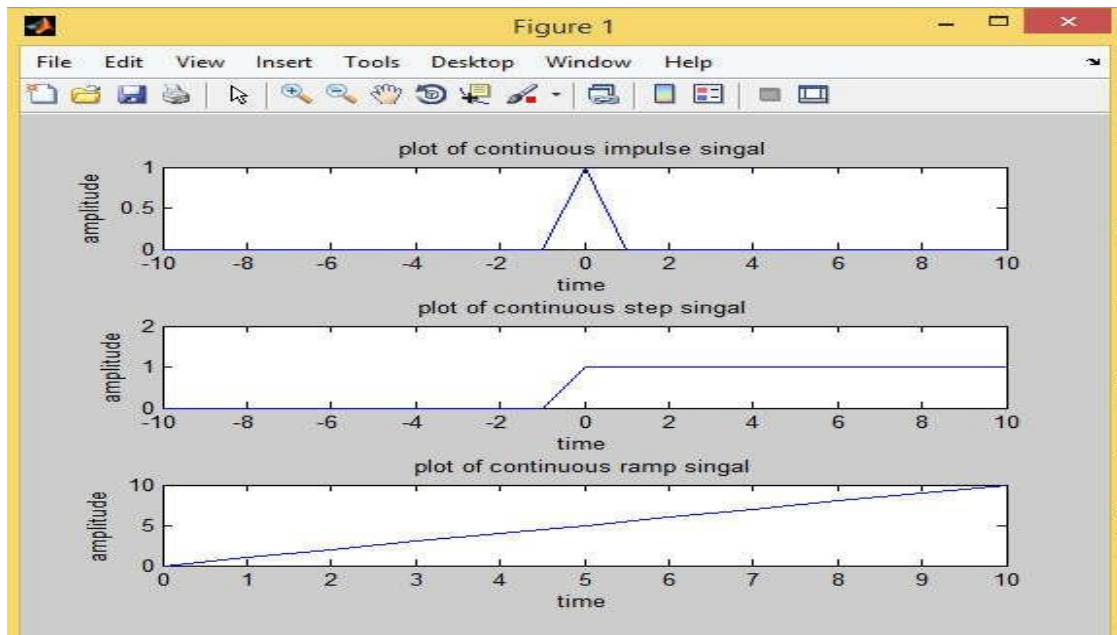
PROGRAM:-

```

clc;
clear all;
close all;
%%To plot of continuous impulse, step & ramp signal
n=-10:10;
delta=(n==0)
step=(n>=0);
n1=0:10;
ramp=n1;

```

```
subplot(3,1,1)
plot(n,delta)
xlabel('time');
ylabel('amplitude');
title('plot of continuous impulse signal');
subplot(3,1,2)
plot(n,step)
axis([-10 10 0 2])
xlabel('time')
ylabel('amplitude')
title('plot of continuous step signal')
subplot(3,1,3)
plot(n1,ramp)
xlabel('time');
ylabel('amplitude');
title('plot of continuous ramp signal');
%To plot of discrete impulse, step & ramp signal
figure
subplot(3,1,1)
stem(n,delta)
xlabel('time');
ylabel('amplitude');
title('plot of discrete impulse signal');
subplot(3,1,2)
stem(n,step)
xlabel('time');
ylabel('amplitude');
title('plot of discrete step signal');
subplot(3,1,3)
stem(n1,ramp)
xlabel('time');
ylabel('amplitude');
title('plot of discrete ramp signal');
```

OUTPUT:-**CONCLUSION:-**

From the above experiment we have plotted unit impulse signal, unit step signal & unit ramp signal using MATLAB function.

EXPERIMENT NO. - 7

AIM OF THE EXPERIMENT:-

To generate the plot of sine and cosine wave using MATLAB functions.

SOFTWARE REOURIED:-

1. MATLAB 7.12.0 (R2011a) SOFTWARE
2. Personal Computer Loaded with Windows 7/XP SP2.

THEORY:-

Sine is a sinusoidal wave which range from $0-2\pi$ and forms a periodic wave whose amplitude ranges from 1 to -1. The sine function is defined as $y=\sin(x)$

Cosine is also a sinusoidal wave which range from $0-2\pi$ and forms a periodic wave whose amplitude ranges from 1 to -1. The cosine function is defined as $y=\cos(x)$

PROCEDURE:-

1. Open MATLAB
2. Open new M-file
3. Type the program to generate the plot of sine and cosine wave.
4. Save in current directory
5. Compile and Run the program
6. For the output see command window/Figure window

PROGRAM:-

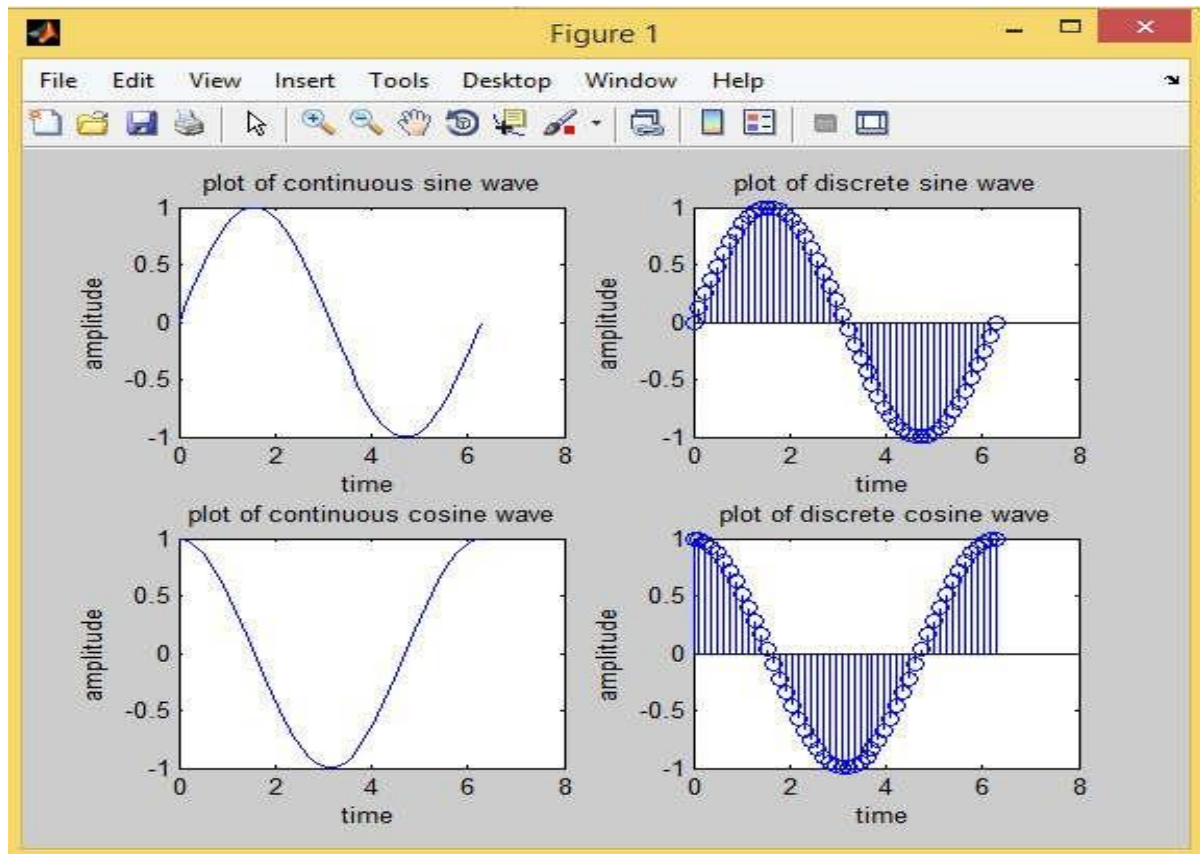
```

clc;
clear all;
close all;
% plot of continuous sine wave
x1=linspace(0,2*pi,50);
y1=sin(x1);
subplot(2,2,1);
plot(x1,y1);
xlabel('time');
ylabel('amplitude');
title('plot of continuous sine wave');
% plot of discrete sine wave
subplot(2,2,2);
stem(x1,y1);
xlabel('time');
ylabel('amplitude');
title('plot of discrete sine wave');
%plot of continuous cosine wave
x2=linspace(0,2*pi,50);

```

```
y2=cos(x2);  
subplot(2,2,3);  
plot(x2,y2);  
xlabel('time');  
ylabel('amplitude');  
title('plot of continuous cosine wave');  
%plot of discrete cosine wave  
subplot(2,2,4);  
stem(x2,y2);  
xlabel('time');  
ylabel('amplitude');  
title('plot of discrete cosine wave');
```

OUTPUT:-



CONCLUSION:-

From the above experiment we have studied and generated sine and cosine wave using MATLAB functions.

EXPERIMENT NO. - 8(A)

AIM OF THE EXPERIMENT:-

To verify Superposition theorem using MATLAB SIMULINK

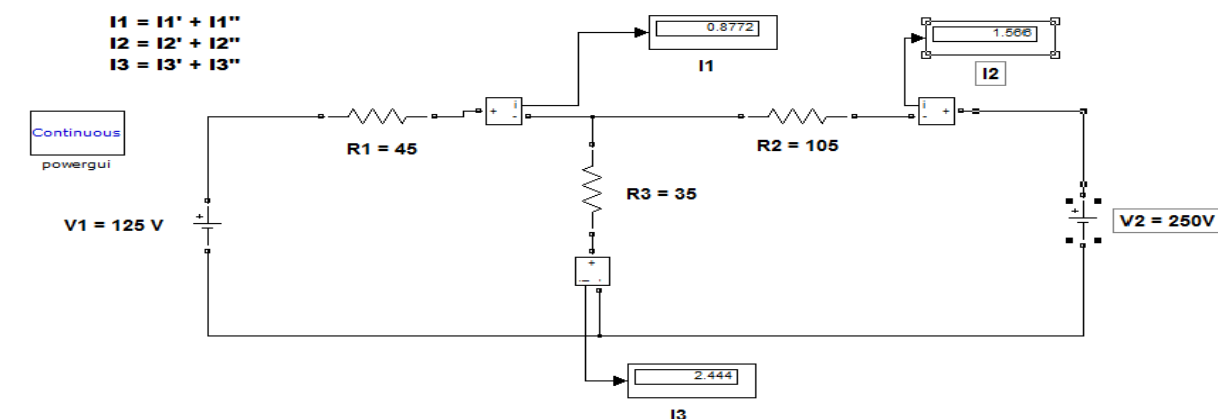
SOFTWARE REOURIED:-

1. MATLAB 7.12.0 (R2011a) SOFTWARE
2. Personal Computer Loaded with Windows 7/XP SP2.

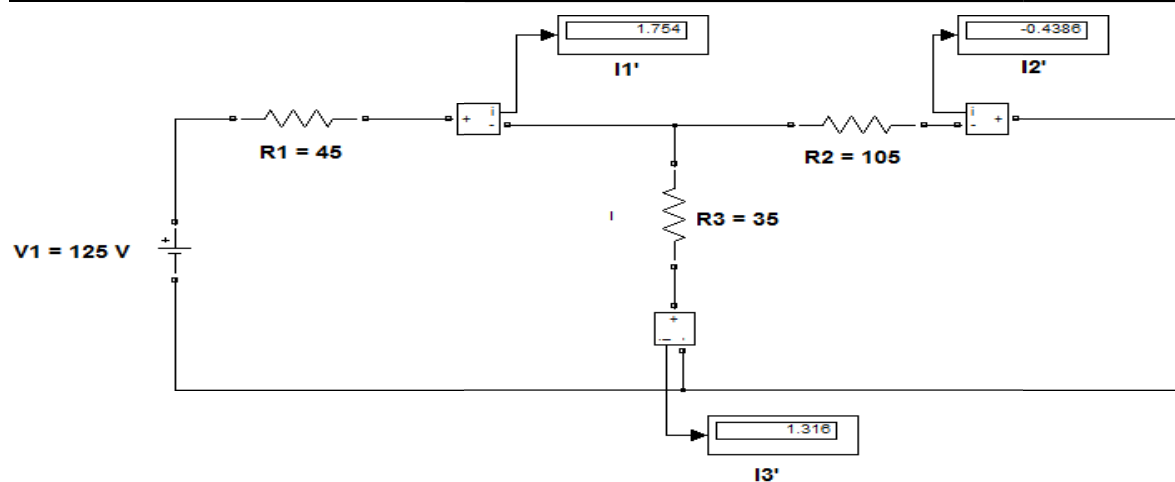
THEORY:-

“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”.

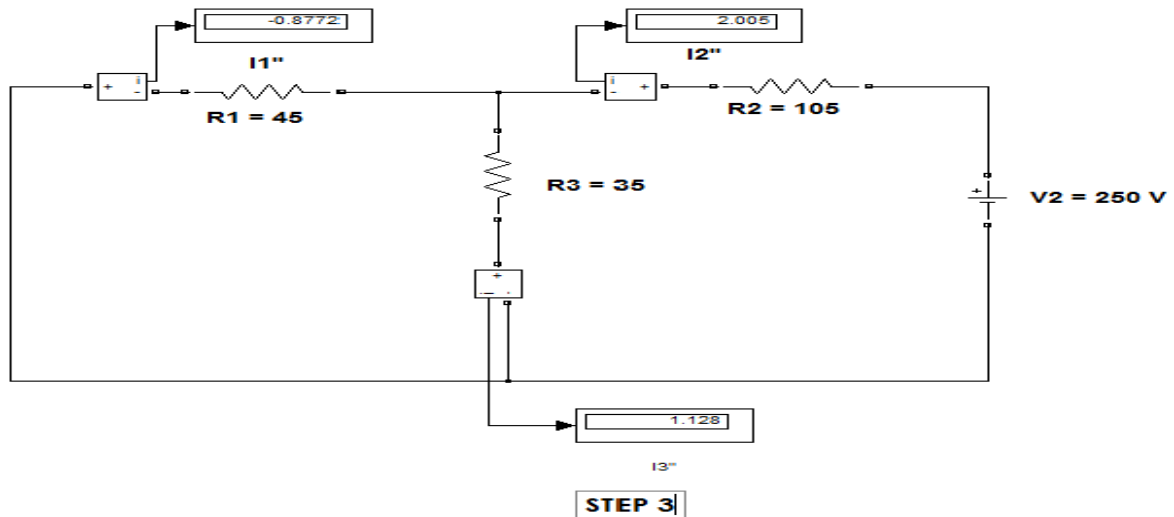
CIRCUIT SIMULATION DIAGRAM



STEP 1



STEP 2



PROCEDURE:-

Step 1:

1. Make the connections as shown in the circuit diagram by using MATLAB SIMULINK.
2. Measure the response (I_3) in the load resistor by considering all the sources 125V and 250V in the network.

Step 2:

1. Replace the sources 250v with their internal impedances (short circuited).
2. Measure the response (I_3') in the load resistor by considering 125V source in the network.

Step 3:

1. Replace the sources 125V with their internal impedances (short circuited).
2. Measure the response (I_3'') in the load resistor by c considering 250V source in the network.

Step 4:

1. The responses obtained in step 1 should be equal to the sum of the responses obtained in step 2 and 3 i.e. $I_3 = I_3' + I_3''$

Hence Superposition Theorem is verified.

OUTPUT:-

CONCLUSION:-

From the above experiment we studied and verified Superposition theorem using MATLAB SIMULINK.

EXPERIMENT NO. - 8(B)

AIM OF THE EXPERIMENT:-

To verify Thevenin's theorem using MATLAB SIMULINK

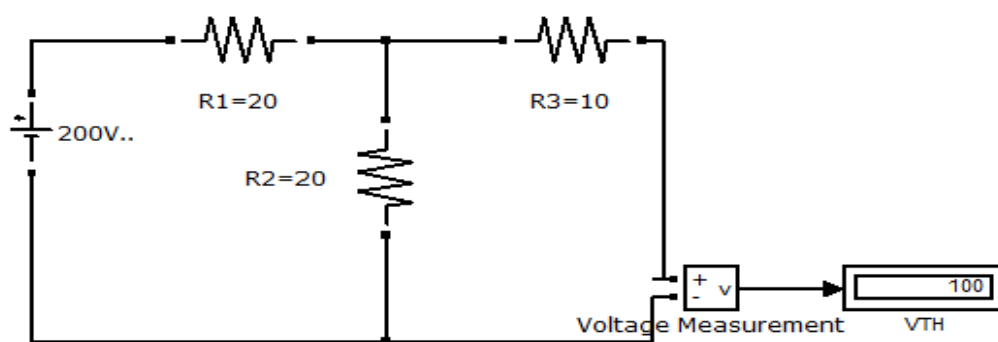
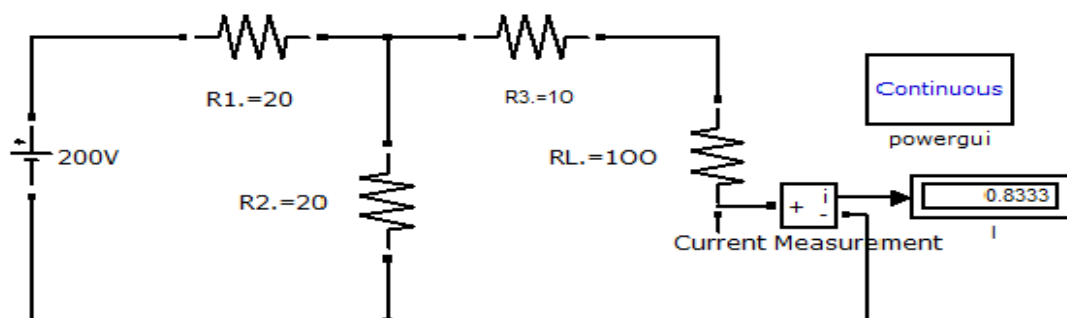
SOFTWARE REOURIED:-

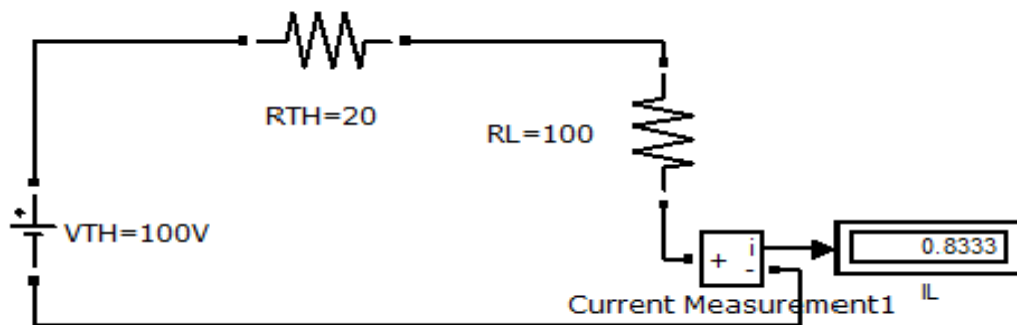
1. MATLAB 7.12.0 (R2011a) SOFTWARE
2. Personal Computer Loaded with Windows 7/XP.

THEORY:-

“Any linear active two terminal networks consisting of voltage and current source with some resistances. It can be replaced by an equivalent Thevenin's voltage source or voltage source having its value is equal to the Thevenin's equivalent voltage with a series resistor which is known as Thevenin's resistance. The equivalent voltage source (V_{TH}) is the open circuit voltage measured at the terminals of the network and the resistance, known as Thevenin's equivalent resistance, R_{TH} , is the resistance measured at the terminals with all the independent sources in the network reduced to zero”.

CIRCUIT SIMULATION DIAGRAM:-





PROCEDURE:-

Step 1:

1. Make the connections as shown in the circuit diagram by using 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.e. $I = \underline{\hspace{2cm}}$

Current measured from Thevenin's Equivalent Circuit should be same as current obtained from the actual circuit i.e. $I = I_L$.

Hence Thevenin's Theorem is Verified.

OUTPUT:-

CONCLUSION:-

From the above experiment we studied and verified Superposition theorem using MATLAB SIMULINK.

EXPERIMENT NO. - 8(C)

AIM OF THE EXPERIMENT:-

To verify Norton's theorem using MATLAB SIMULINK

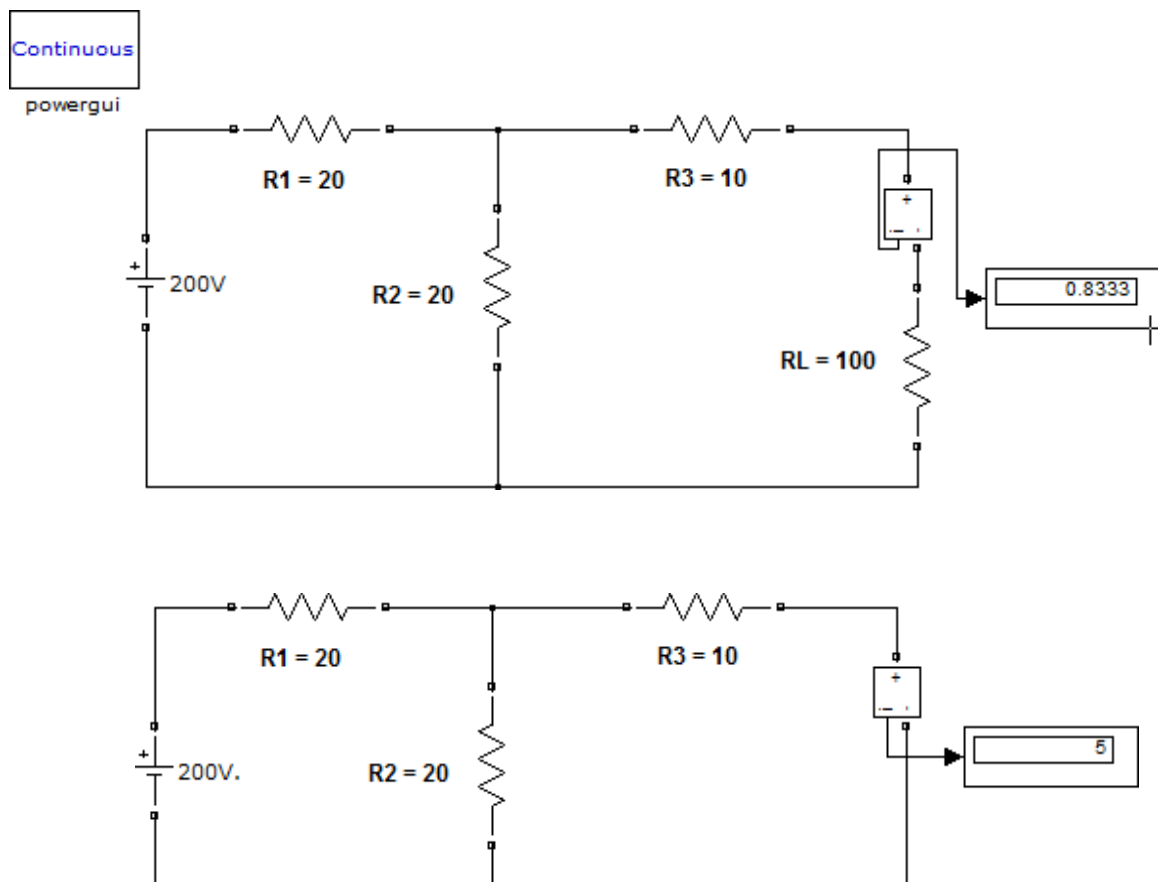
SOFTWARE REOURIED:-

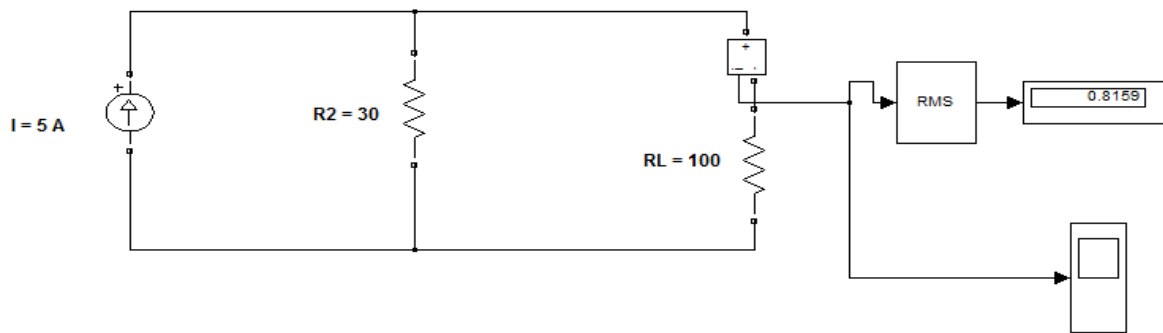
1. MATLAB 7.12.0 (R2011a) SOFTWARE
2. Personal Computer Loaded with Windows 7/XP SP2.

THEORY:-

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

CIRCUIT SIMULATION DIAGRAM:-





PROCEDURE:-

Step 1:

1. Make the connections as shown in the circuit diagram by using 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)

1. Short the load terminals and measure the current through the short circuited terminal
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 $I = \underline{\hspace{2cm}}$

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

.i.e. $I = I_L$.

Hence Norton's Theorem is verified.

OUTPUT:-

CONCLUSION:

From the above experiment we studied and verified Norton's theorem using MATLAB SIMULINK.

EXPERIMENT NO. - 9

AIM OF THE EXPERIMENT:-

To Simulate a half wave uncontrolled rectifier using MATLAB SIMULINK.

SOFTWARE REOURIED:-

1. MATLAB 7.12.0 (R2011a) SOFTWARE
2. Personal Computer Loaded with Windows 7/XP SP2.

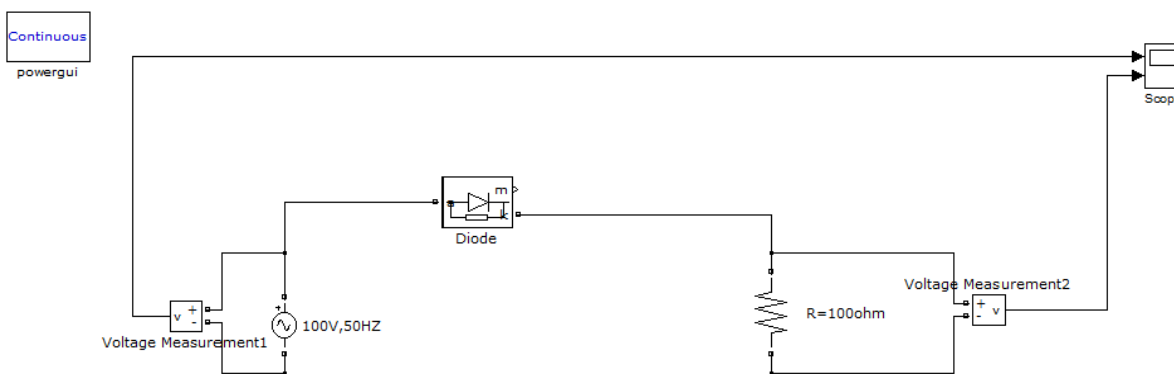
THEORY:-

Rectification is a process of conversion of alternating input voltage to direct output voltage. A rectifier converts ac power to dc power. The half wave rectifier consists of a single diode connected in series with the load. In half wave rectification, when a single-phase AC supply is connected to a half-wave rectifier, the diode passes only the positive half cycle of the supply and blocks the negative half cycle. As the rectifier passes only a half of the supply it is called a half wave rectifier.

During each positive half cycle of the AC sine wave, the anode is positive with respect to the cathode. Hence the diode is *forward biased*. Under the forward biased condition, the diode acts like a closed switch resulting in current flowing through the diode.

During each “negative” half cycle of the AC sine wave, the anode is negative with respect to the cathode. Hence the diode is *reverse biased* and acts as an open switch. No current flows through the diode or circuit. In this way, current flows through load R_L always in the same direction. Hence D.C. output is obtained across R_L

CIRCUIT SIMULATION DIAGRAM:-

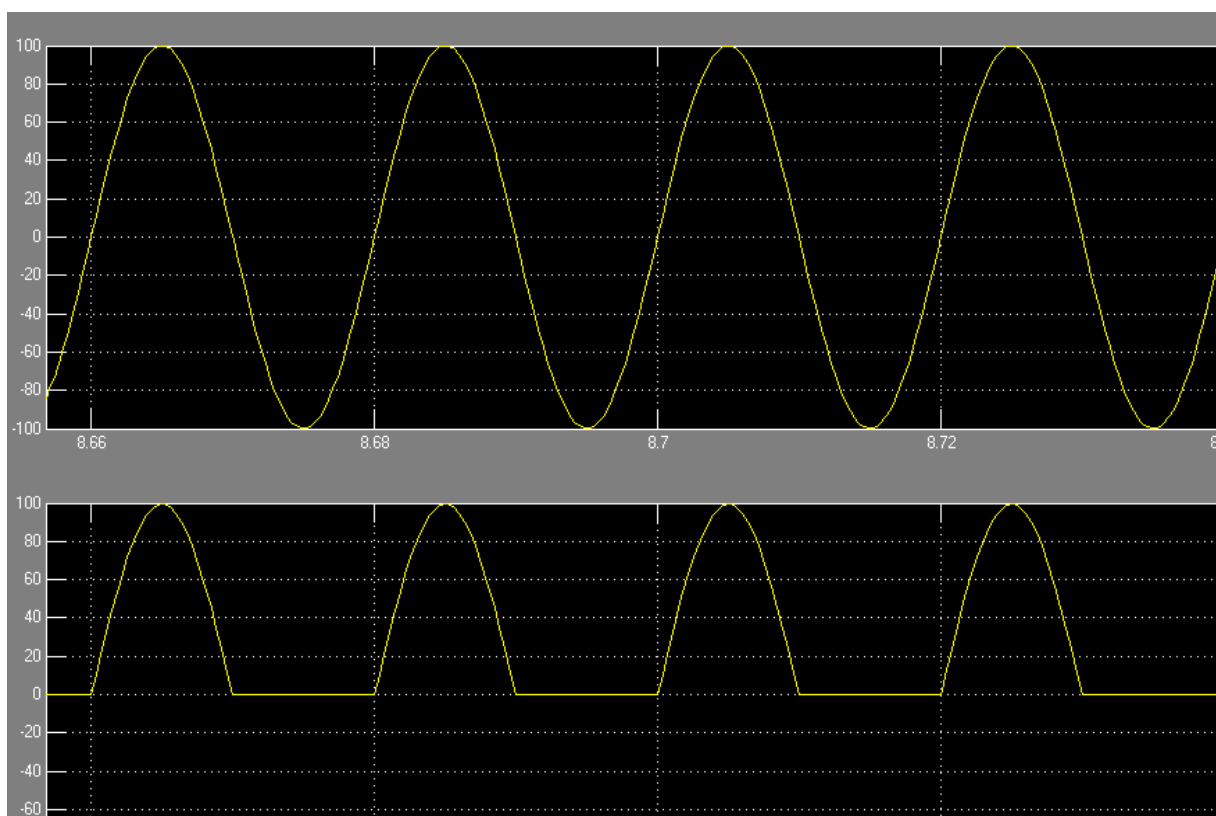


PROCEDURE:-

1. Open MATLAB and open Simulink then create a new file(new module)
2. Connections are made as per the circuit diagram by taking the required components from the respective blocks and joining them by left clicking on component ends .

3. For setting value of different component, double click on that component then block parameter dialogue box will appear then enter value as per circuit then press ok.
4. Go to "simulation tab" and click on "run" to simulate the circuit.
5. Observe the result in the display and waveforms in the scope.
6. Finally the print out of the MATLAB circuit and the waveform is taken.

OUTPUT:-



CONCLUSION:

From the above experiment we have studied and Simulated the waveform of a half wave uncontrolled rectifier using MATLAB SIMULINK.

Date:

Signature of the Student

Name:

Roll no.:

Semester:

Group no.

Date:

Instructor's Signature

EXPERIMENT NO. - 10

AIM OF THE EXPERIMENT:-

To simulate 1-phase full wave bridge controlled rectifier using MATLAB SIMULINK

SOFTWARE REOURIED:-

1. MATLAB 7.12.0 (R2011a) SOFTWARE
2. Personal Computer Loaded with Windows 7/XP SP2.

THEORY:-

The circuit consist of four thyristors T1, T2, T3 and T4, a voltage source Vs and a R Load.

During the positive half cycle of the input voltage, the thyristors T1 & T2 is for ward biased but it does not conduct until a gate signal is applied to it.

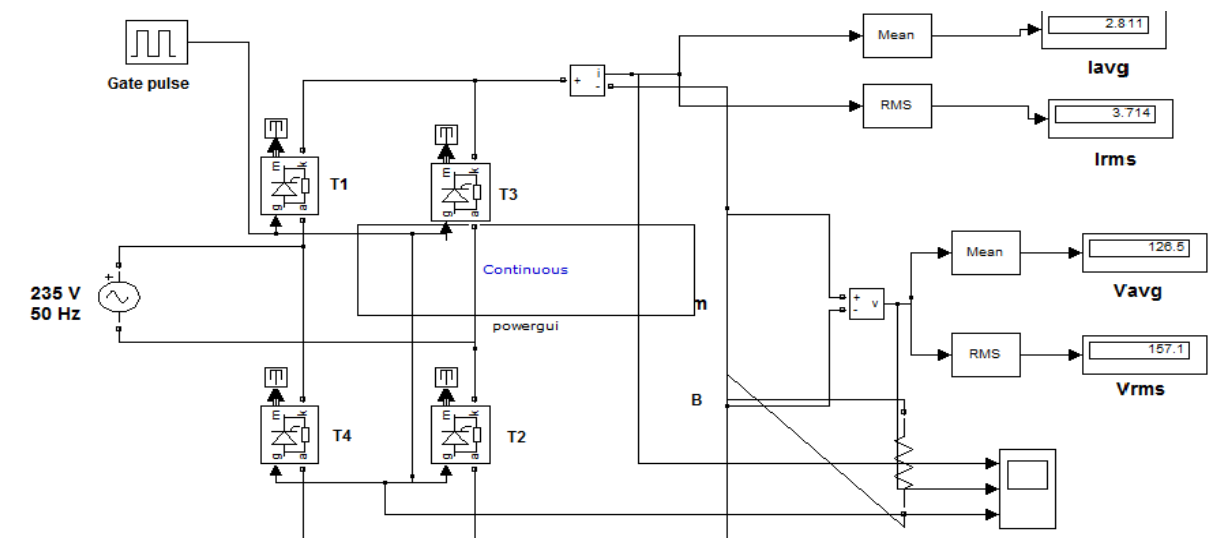
When a gate pulse is given to the thyristors T1 & T2 at $\omega t = \alpha$, it gets turned ON and begins to conduct. When the T1 & T2 is ON, the input voltage is applied to the load through the path Vs-T1-Load-T2-Vs.

During the negative half cycle, T3 & T4 is forward biased, the thyristor T1 & T2 gets reverse biased and turns OFF. When a gate pulse is given to the thyristor T3 & T4 at $\omega t = \pi + \alpha$, it gets turned ON and begins to conduct. When T3 & T4 is ON, the input voltage is applied to the load Vs-T3-Load-T4-Vs. Here the load receives voltage during both the half cycles

The average value of output voltage can be varied by varying the firing angle α .

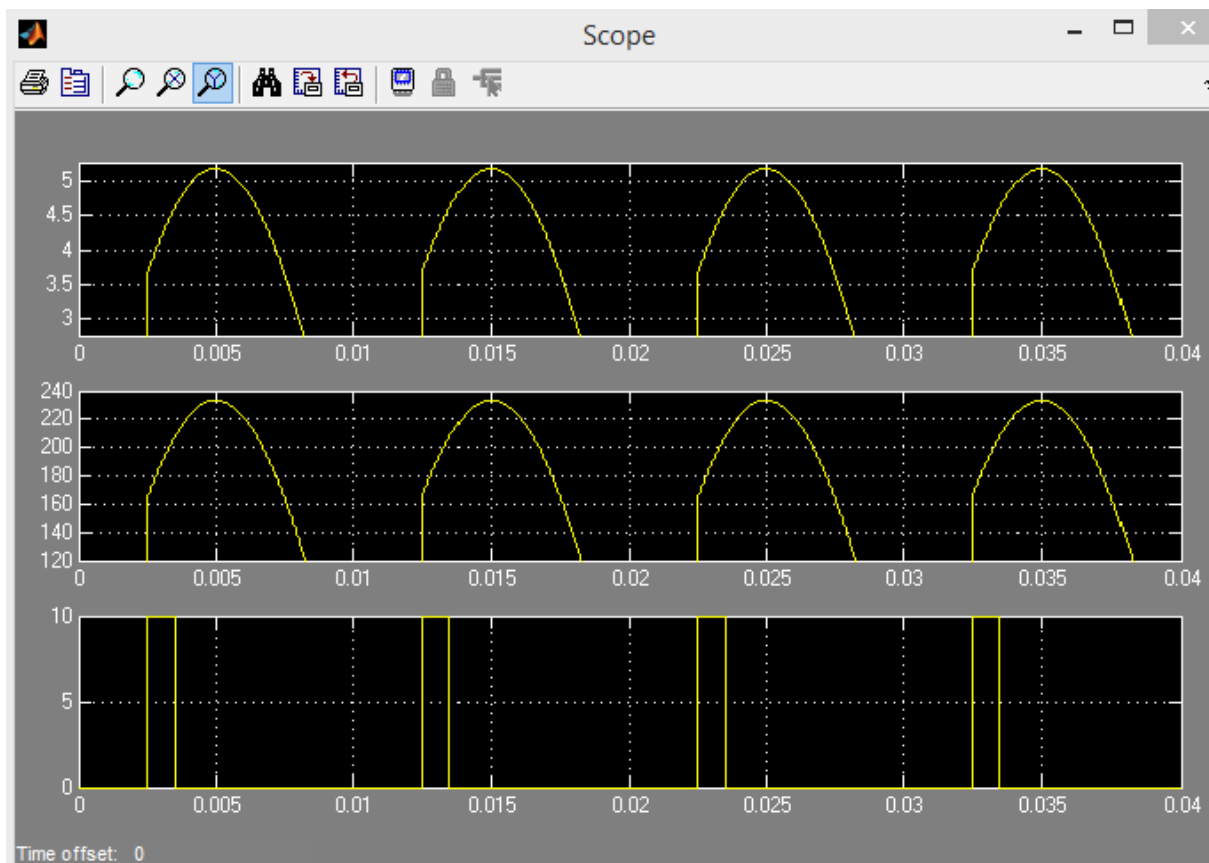
The waveform shows the plot of input voltage, gate current, output voltage, output current and voltage across thyristor.

CIRCUIT SIMULATION DIAGRAM:-



PROCEDURE:-

1. Open MATLAB and open Simulink then create a new file(new module)
2. Connections are made as per the circuit diagram by taking the required components from the respective blocks and joining them by left clicking on component end .
3. For setting value of different component,double click on that component then block parameter dialogue box will appear their enter value as per circuit then press ok.
4. Go to “simulation tab” and click on “run” to simulate the circuit.
5. Observe the result in the display and waveforms in the scope.
6. Finally the print out of the MATLAB circuit and the waveform is taken.

OUTPUT:-**CONCLUSION:-**

From the above experiment we have studied and Simulated the waveform of 1-phase full wave bridge controlled rectifier using MATLAB SIMULINK