MALLA REDDY ENGINEERING COLLEGE AND MANAGEMENT SCIENCES

Department of Electrical and Electronics Engineering

Electrical Simulation Tools Laboratory

B.Tech II -YEAR I-SEM

STUDENT LAB MANUAL

Department of Electrical & Electronics Engineering

MALLREDDY ENGINEERING COLLEGE AND MANAGEMENT SCIENCES

(Approved by AICTE New Delhi & Affiliated to JNTU Hyderabad)

Kistapur, Medchal, Medchal – 501401

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

Vision of the Institute:

The aspiration is to emerge as a premier institution in technical education to produce competent engineers and management professionals contributing to Industry and Society.

Mission of the Institute (MI):

The aspirations are fulfilled and continue to be fulfilled:

MI-1: By providing the student supporting systems:
To impart updated pedagogical techniques with supportive learning environment and state-of-the-art facilities.

MI-2: By training the students as per the industry needs:
To cultivate a culture of interdisciplinary approach, problem solving, innovative ecosystem, and entrepreneurship by facilitating critical thinking, teamwork, and research-driven activities with hands-on learning.

MI-3: By educating the students about society's needs:
To instill ethical, social, and environmental values through community engagement resulting in sustainable development of society.

Vision of the EEE-Department:

The aspiration is to produce competent Electrical and Electronics Engineering Graduates capable of making valuable contributions in the field of Electrical and Electronics Engineering.

Mission of the Department:

MD-1: Student Support Systems:
To equip students with advanced learning skills in Electrical and Electronics Engineering, while providing them with the necessary professional competencies to overcome future challenges.

MD-2:Training the students as per the industry needs:
To facilitate the students to acquire interdisciplinary skills in renewable energy, electric vehicles, and power electronics applications through practical knowledge and innovative techniques to meet evolving global challenges.

MD-3:Educating the students, the needs of society:
To develop professional ethics, self-confidence, and leadership qualities among students.

Program Educational Objectives (PEO'S)

PEO 1: MREM B.Tech EEE graduates shall be able to apply technical knowledge in Electrical and Electronics Engineering, empowering them to pursue higher studies or succeed in their professional careers **in the electrical Power Industry**.

PEO 2: MREM B.Tech EEE graduates shall be able to design and implement complex electrical systems, **meeting the electrical and electronics industry** demands.

PEO 3: MREM B.Tech EEE graduates shall be able to handle societal and environmental problems with ethical values **as demanded by society.**

Program Specific Outcomes (PSOs)

PSO1: Provide efficient problem-solving techniques in the areas of Power Electronics, Power Systems, Control systems, and Electrical Machines using MATLAB/MULTISIM.

PSO2: Design and develop a wide range of Electrical and Electronics Systems, specifically emphasizing Electric Drives, Conventional Renewable Energy, and Automation to demonstrate overall knowledge and contribute to the betterment of society.

PROGRAM OUTCOMES (POs)

PO1: ENGINEERING KNOWLEDGE:

Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: PROBLEM ANALYSIS:

Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: DESIGN/DEVELOPMENT OF SOLUTIONS:

Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: CONDUCT INVESTIGATIONS OF COMPLEX PROBLEMS:

Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: MODERN TOOL USAGE:

Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: THE ENGINEER AND SOCIETY:

Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: ENVIRONMENT AND SUSTAINABILITY:

Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: ETHICS:

Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: INDIVIDUAL AND TEAM WORK:

Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: COMMUNICATION:

Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, give and receive clear instructions.

PO11 PROJECT MANAGEMENT AND FINANCE:

Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12 LIFE-LONG LEARNING:

Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

MALLA REDDY ENGINEERING COLLEGE AND MANAGEMENT SCIENCES

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

GENERAL LABORATORY INSTRUCTIONS

- Before entering the lab the student should carry the following things
  - Identity card issued by the college
  - Class notes
  - Lab Manual
  - Lab Record
  - Calculator, scales, pencils etc
- Student must sign in and sign out in the register provided when attending the lab session without fail
- Come to the laboratory in time. Students, who are late more than 5 min., will not be allowed to attend the lab.
- Students need to maintain 100% attendance in lab if not a strict action will be taken.
- All students must follow a Dress Code while in the laboratory
- Foods, drinks are NOT allowed.
- All bags must be left at the indicated place.
- The objective of the laboratory is learning. The experiments are designed to illustrate phenomena in different areas of Physics and to expose you to measuring instruments, conduct the experiments with interest and an attitude of learning
- You need to come well prepared for the experiment.
- Work quietly and carefully
- Be honest in recording and representing your data.
- If a particular reading appears wrong repeat the measurement carefully, to get a better fit for a graph
- All presentations of data, tables and graphs calculations should be neatly and carefully done
- Graphs should be neatly drawn with pencil. Always label graphs and the axes and display units.
- If you finish early, spend the remaining time to complete the calculations and drawing graphs.
- Do not fiddle with apparatus. Handle instruments with care. Report any breakage to the Instructor. Return all the equipment you have signed out for the purpose of your experiment.

MALLA REDDY ENGINEERING COLLEGE AND MANAGEMENT SCIENCES
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

SPECIFIC SAFETY RULES FOR LABORATORY

- NO STUDENT is allowed in to the lab without shoe or apron.
- You must not damage or tamper with the equipment or leads.
- You should inspect laboratory equipment for visible damage before using it. If there is a problem with a piece of equipment, report it to the technician or lecturer. DONOT return equipment to a storage area
- You should not work on circuits where the supply voltage exceeds 40 volts without very specific approval from your lab supervisor. If you need to work on such circuits, you should contact your supervisor for approval and instruction on how to do this safely before commencing the work.
- Never strip insulation from a wire with your teeth or a knife, always use an appropriate wire stripping tool.
- Shield wire with your hands when cutting it with a pliers to prevent bits of wire flying about the bench.

LIST OF EXPERIMENTS

S.No	Title of the Experiment	Page No
1.	Introduction to basic block sets of simulation platforms. Basic matrix operations, Generation of standard test signals	
2.	Solving the linear and nonlinear differential equations	
3.	Measurement of Voltage, Current and Power in DC circuits.	
4.	Verification of different network theorems with dependent and independent sources using suitable simulation tools	
5.	Verification of performance characteristics of basic Electronic Devices using suitable simulation tools	
6.	Analysis of series and parallel resonance circuits using suitable simulation tools	
7.	Obtaining the response of electrical network for standard test signals using suitable simulationtools.	
8.	Modeling and Analysis of Low pass and High pass Filters using suitable simulation tools	
9.	Performance analysis of DC motor using suitable simulation tools	
10.	Modeling and analysis of Equivalent circuit of transformer using suitable simulation tools.	
11.	Analysis of single-phase bridge rectifier with and without filter using suitable Simulation tools	
12.	Modeling and Verification of Voltage Regulator using suitable simulation tools.	
13.	Modeling of transmission line using simulation tools.	
14.	Performance analysis of Solar PV model using suitable simulation tools	

MALLA REDDY ENGINEERING COLLEGE AND MANAGEMENT SCIENCES DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

ELECTRICAL SIMULATION TOOLS LAB

Course Outcome	Course Outcome Statement	Bloom's Taxonomy level
C218.1	Develop knowledge of software packages to model and program electrical and electronics systems	**Understand**
C218.2	Model different electrical and electronic systems and analyze the results	**Analyze**
C218.3	Articulate importance of software packages used for simulation in laboratory experimentation by analyzing the Simulation results	**Evaluate**

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

POWER SYSTEM SIMULATION LAB

Course Outcomes Mapping With Bloom's Taxonomy
PO,PSO,CO's and Mapping

CO-PO MAPPING

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
C218.1	3	3	1	3	2	3	2	3	1	2	2	3	2	3
C218.2	3	2	2	1	2	1	2	1	2	2	2	3	3	3
C218.3	3	2	0	0	2	0	1	0	2	0	2	3	2	3
AVG	3	2.3	1	1.3	2	1.3	1.7	1.3	1.6	1.3	2	3	2.3	3

MALLA REDDY ENGINEERING COLLEGE AND MANAGEMENT SCIENCES DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

ELECTRICAL SIMULATION TOOLS LAB

List of Experiments:

Exp No	Name of the experiment	COs Mapped	PO	PSO	Bloom's Taxonomy
1	Introduction to basic block sets of simulation platforms. Basic matrix operations, Generation of standard test signals	CO1	PO1,PO2,PO5,PO12	PSO2	Apply
2	Solving the linear and nonlinear differential equations	CO1	PO3,PO4,PO12		Analyze
3	Measurement of Voltage, Current and Power in DC circuits.	CO1	PO1,PO2,PO5	PSO2	Understand
4	Verification of different network theorems with dependent and independent sources using suitable simulation tools	CO2	PO1,PO2,PO5	PSO1,PSO2	Analyze
5	Verification of performance characteristics of basic Electronic Devices using suitable simulation tools	CO2	PO1,PO2,PO5,PO12	PSO1	Analyze, Understand
6	Analysis of series and parallel resonance circuits using suitable simulation tools	CO2	PO1,PO2,PO5,PO12	PSO1	Analyze, Understand
7	Obtaining the response of electrical network for standard test signals using suitable simulationtools.	CO2	PO1,PO2,PO5,PO12	PSO1	Analyze, Understand
8	Modeling and Analysis of Low pass and High pass Filters using suitable simulation tools	CO2,CO3	PO3,PO4,PO12		Analyze
9	Performance analysis of DC motor using suitable simulation tools	CO2,CO3	PO1,PO2,PO5	PSO2	Analyze
10	Modeling and analysis of Equivalent circuit of transformer using suitable simulation tools.	CO2,CO3	PO1,PO3,PO4	PSO1	Analyze, Understand
11	Analysis of single-phase bridge rectifier with and without filter using suitable Simulation tools	CO3	PO1,PO3,PO4	PSO2	Analyze
12	Modeling and Verification of Voltage Regulator using suitable simulation tools.	CO2,CO3	PO1,PO3,PO4	PSO2	Analyze, Understand

13	Modeling of transmission line using simulation tools.	CO2,CO3	PO1,PO3,PO4	PSO1	Analyze, Understand
14	Performance analysis of Solar PV model using suitable simulation tools	CO2,CO3	PO1,PO3,PO4,PO5	PSO2	Analyze, Understand

Introduction to MATLAB:

MATLAB is a high performance language for technical computing. It integrates computation, visualization and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. MATLAB is numeric computation software for engineering and scientific calculations. MATLAB is primary tool for matrix computations. MATLAB is being used to simulate random process, power system, control system and communication theory. MATLAB comprising lot of optional tool boxes and block set like control system, optimization, and power system and so on.

Typical uses:

- Mathematics tools and computation
- Algorithm development
- Modeling, simulation and prototype
- Data analysis, exploration and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is a widely used tool in electrical engineering community. It can be used for simple mathematical manipulation with matrices for understanding and teaching basic mathematical and engineering concepts and even for studying and simulating actual power system and electrical system in general. The original concept of a small and handy tool has evolved replace and/or enhance the usage of traditional simulation tool for advanced engineering applications.to becomes an engineering work house. It is now accepted that MATLAB and its numerous tool boxes

Getting started with MATLAB:

To open the MATLAB applications double click the MATLAB icon on the desktop. To quit from MATLAB type…

```
>> quit
(Or)
>>exit
```

To select the (default) current directory click ON the icon […] and browse for the folder named "D:\SIMULAB\xxx", where xxx represents roll number of the individual candidate in which a folder should be created already.

When you start MATLAB you are presented with a window from which you can enter commands interactively. Alternatively, you can put your commands in an M- file and execute it at the MATLAB prompt. In practice you will probably do a little of both. One good approach is to incrementally create your file of commands by first executing them.

M-files can be classified into following 2 categories,

i) Script M-files – Main file contains commands and from which functions can also be called

ii) Function M-files – Function file that contains function command at the first line of the M-file

Simulink

At the MATLAB prompt type simulink and brings up the “Simulink Library Browser”. Each of the items in the Simulink Library Browser are the top level of a hierarchy of palette of elements that you can add to a simulink model of your own creation. The “simulink” pallete contains the majority of the elements used in the MATLAB. Simulink has built into it a variety of integration algorithm for integrating the dynamic equations. You can place the dynamic equations of your system into simulink in four ways.

1 Using integrators

2. Using transfer functions

3. Using state space equations

MATLAB workspace:

The workspace is the window where you execute MATLAB commands (Ref. figure-1). The best way to probe the workspace is to type whos. This command shows you all the variables that are currently in workspace. You should always change working directory to an appropriate location under your user name.Another useful workspace like command is

>>clear all

It eliminates all the variables in your workspace. For example, start MATLAB and execute the following sequence of commands

>>a=2;

>>b=5;

>>whos

>>clear all

The first two commands loaded the two variables a and b to the workspace and assigned value of 2 and 5 respectively. The clear all command clear the variables available in the work space. The arrow keys are real handy in MATLAB. When typing in long expression at the command line, the up arrow scrolls through previous commands and down arrow advances the other direction. Instead of retyping a previously entered command just hit the up arrow until you find it. If you need to change it slightly the other arrows let you position the cursor anywhere. Finally any DOS command can be entered in MATLAB as long as it is preceded by any exclamation mark.

MATLAB data types:

The most distinguishing aspect of MATLAB is that it allows the user to manipulate vectors As for as MATLAB is concerned a scalar is also a 1 x 1 array. For example clear your workspace and execute the commands.

```
>>a=4.2:
>>A=[1 4;6 3];
>>whos
```

Two things should be evident. First MATLAB distinguishes the case of a variable name and that both a and A are considered arrays. Now let's look at the content of A and a.

```
>>a
>>A
```

Again two things are important from this example. First anybody can examine the contents of any variables simply by typing its name at the MATLAB prompt. When typing in a matrix space between elements separate columns, whereas semicolon separate rows. For practice, create the matrix in your workspace by typing it in all the

MATLAB prompt.

```
>>B= [3 0 -1; 4 4 2;7 2 11];
```

(use semicolon(;) to represent the end of a row)

```
>>B
```

Arrays can be constructed automatically. For instance to create a time vector where the time points start at 0 seconds and go up to 5 seconds by increments of 0.001

```
>>mytime =0:0.001:5;
```

Automatic construction of arrays of all ones can also be created as follows,

```
>>myone=ones (3,2)
```

Outcome:

By doing the experiment, the students can understand the concepts of MATLAB programming in solving power systems problems.

Application:

MATLAB Used
Algorithm development
Scientific and engineering graphics
Modeling, simulation, and prototyping
Application development, including Graphical User Interface building
Math and computation
Data analysis, exploration, and visualization

Viva Questions:

1. What is meant by MATLAB?

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation.
2. What are the different functions used in MATLAB?

The different function intersect ,bitshift, categorical, isfield
3. What are the different operators used in MATLAB?

Arithmetic, Relational Operations, Logical Operations, Set Operations, Bit-Wise Operations
4. What are the different looping statements used in MATLAB?

For , while
5. What are the different conditional statements used in MATLAB?

If, else
6. What is Simulink?

Simulink, developed by Math Works, is a graphical programming environment for modeling, simulating and analyzing multi domain dynamical systems. Its primary interface is a graphical block diagramming tool and a customizable set of block libraries.
7. What are the four basic functions to solve Ordinary Differential Equations (ODE)?

ode45, ode15s, ode15i
8. Explain how polynomials can be represented in MATLAB?

poly, polyval, polyvalm , roots
9. What is meant by M-file?

An m-file, or script file, is a simple text file where you can place MATLAB commands. When the file is run, MATLAB reads the commands and executes them exactly as it would if you had typed each command sequentially at the MATLAB prompt.
10. What is Interpolation and Extrapolation in MATLAB?

Interpolation in MATLAB is divided into techniques for data points on a grid and scattered data points.
11. List out some of the common toolboxes present in MATLAB?

Control system tool box, power system tool box, communication tool box,
12. What are the MATLAB System Parts?

MATLAB Language, MATLAB working environment, Graphics handler, MATLAB mathematical library, MATLAB Application Program Interface.
13. What are the different applications of MATLAB?

Algorithm development, Scientific and engineering graphics, Modeling, simulation, and prototyping, Application development, including Graphical User Interface building, Math and computation, Data analysis, exploration, and visualization.

EXPERIMENT NO-1

BASIC OPERATIONS ON MATRICES

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

Software Required: MATLAB software.

COMMANDS:

1. eye - identity matrix
2. zeros - matrix of zeros
3. ones - matrix of ones
4. diag - extract diagonal of a matrix or create diagonal matrices
5. triu - upper triangular part of a matrix
6. tril - lower triangular part of a matrix
7. rand - ran
8. size- size of a matrix
9. det -determinant matrix
10. inv- inverse of a matrix
11. rank- rank of a matrix

PROGRAM

```
%%%% % creating a column vector

>> a= [1; 2; 3]

a =1

2

3


%%%% Creating a row vector

>> b= [1 2 3]

b=

1 2 3

% creating a matrix

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

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

% creating zeros matrix

>>

D=zeros(3,3
) D =

```
0   0   0
0   0   0
0   0   0
```

% creating identity matrix

>>
F=eye(3
,3)F =

```
1   0   0
0   1   0
0   0   1
```

% creating one’s matrix

>>

k=ones(5,
5)k =

```
1   1   1   1   1
1   1   1   1   1
1   1   1   1   1
1   1   1   1   1
1   1   1   1   1
```

% Create a diagonal matrix

>>

diag(k)

ans =

1

1

1

1

1

% Create a upper diagonal matrix

>>

triu(k)

ans =

1	1	1	1	1
0	1	1	1	1
0	0	1	1	1
0	0	0	1	1
0	0	0	0	1

% Create a lower diagonal matrix

>> tril(k)

ans =

1	0	0	0	0
1	1	0	0	0
1	1	1	0	0
1	1	1	1	0
1	1	1	1	1

% Size of a matrix

>> Size(k)

ans =

5 5

% det of a matrix
>> det(k) ans =

0
% Rank of a matrix
>> rank(k) ans =

1
% inv of a matrix
>> inv(k) ans =

Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf

% matrix multiplication
>> l=a*b l =

1	2	3
2	4	6
3	6	9

Result:

EXPERIMENT NO-1
BASIC OPERATIONS ON MATRICES

Basic Matrix Operations:

```
>> %%%%Creating a column vector%%%%
A=[1;2;3]

A =

     1
     2
     3

>> %%%% Creating a Row Vector %%%%
>> B=[1 2 3]

B =

     1     2     3



>> %%%% Creating a 2x2 Matrix %%%%
>> C=[1 2;4 5]

C =

     1     2
     4     5

>> %%%% Creating a 3x3 Matrix %%%%
>> D=[1 2 3;4 5 6;7 8 9]

D =

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

```
>> %%%% Creating Zero Matrix %%%%
>> E=zeros(3,3)

E =

     0     0     0
     0     0     0
     0     0     0

>> %%%% Creating Identity Matrix %%%%
>> F=eye(3,3)

F =

     1     0     0
     0     1     0
     0     0     1

>> %%%% Creating One's Matrix %%%%
>> G=ones (5,5)

G =

     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1

>> %%%% Creating a Diagonal Matrix %%%%
>> diag (G)

ans =

     1
     1
     1
     1
     1

>> %%%% Creating a Upper Diagnoal Matrix %%%%
>> triu (G)

ans =

     1     1     1     1     1
     0     1     1     1     1
     0     0     1     1     1
     0     0     0     1     1
     0     0     0     0     1
```

```
>> %%%% Creating a Lower Diagnoal Matrix %%%%
>> tril (G)

ans =

     1     0     0     0     0
     1     1     0     0     0
     1     1     1     0     0
     1     1     1     1     0
     1     1     1     1     1

>> size (G)

ans =

     5     5

>> %%%% det of Matrix %%%%%
>> det(G)

ans =

     0

>> %%%% Rank of Matrix %%%%
>> rank (G)

ans =

     1

>> %%%% Inv of Matrix %%%%
>> inv (G)
Warning: Matrix is singular to working precision.

ans =

   Inf   Inf   Inf   Inf   Inf
   Inf   Inf   Inf   Inf   Inf
   Inf   Inf   Inf   Inf   Inf
   Inf   Inf   Inf   Inf   Inf
   Inf   Inf   Inf   Inf   Inf

>> %%%% Matrix Multiplication %%%%
>> H= A*B

H =

     1     2     3
     2     4     6
     3     6     9
```

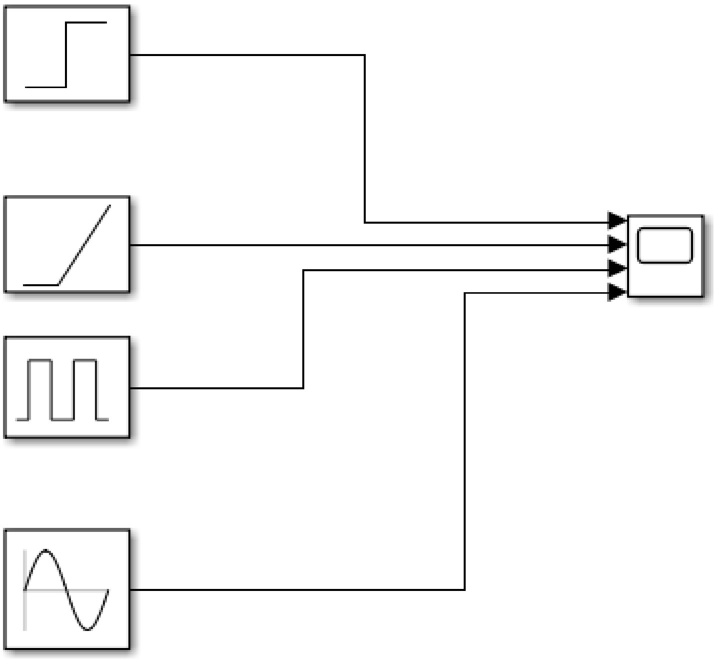
Results:

EXPERIMENT NO-1.1
GENERATION OF STANDARD SIGNALS

AIM: Generate various signals such as Unit Step, Unit Ramp, Unit impulse and Sinusoidalsignals.

Software Required: MATLAB software.

Simulation Diagram:



1. **Unit Step Signal:**

$$\mathbf{r(t) = 1\ for\ t \geq 0, 0\ for\ t < 0}$$

2. **Unit Ramp Signal:**

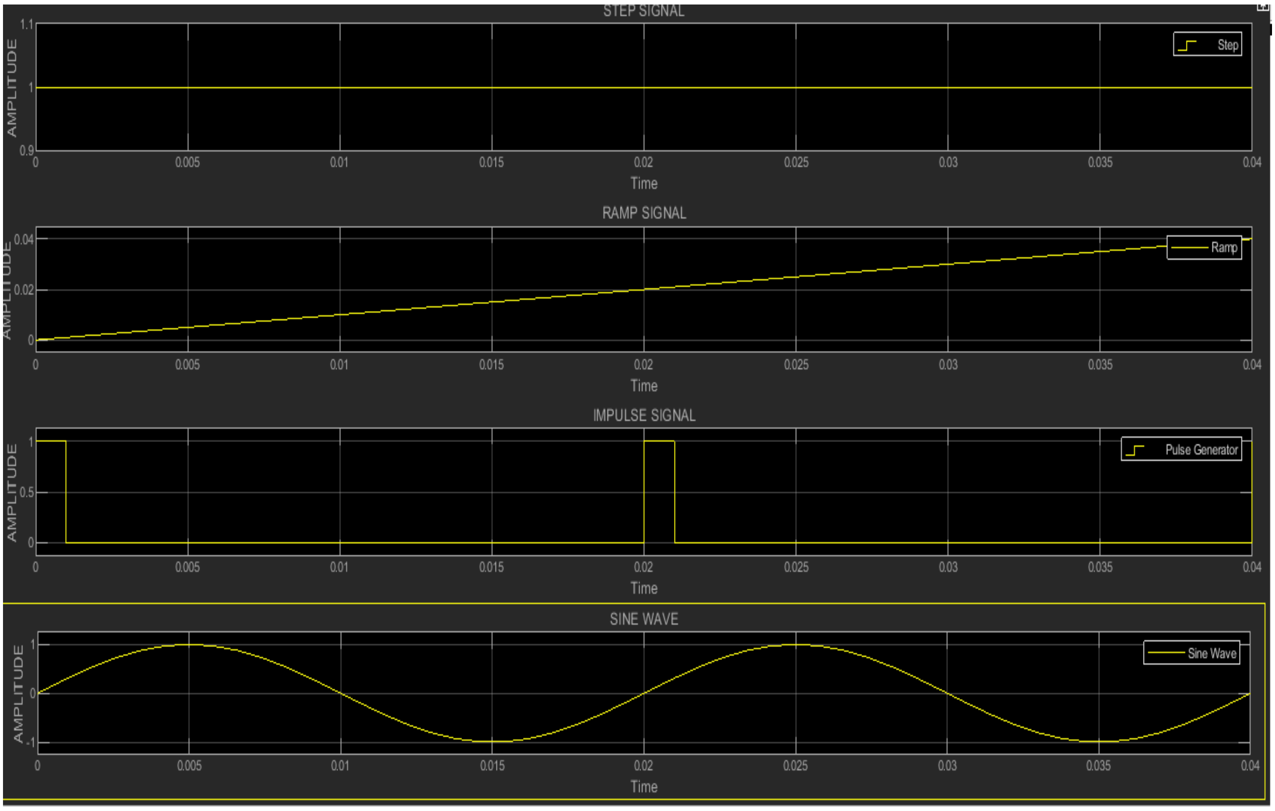
$$\mathbf{r(t) = t\ for\ t \geq 0, 0\ for\ t < 0}$$

3. **Unit Impulse Signal:**

$$\mathbf{\delta(t) = 1\ for\ t = 0, 0\ for\ t \neq 0}$$

4. **Sinusoidal Signal**

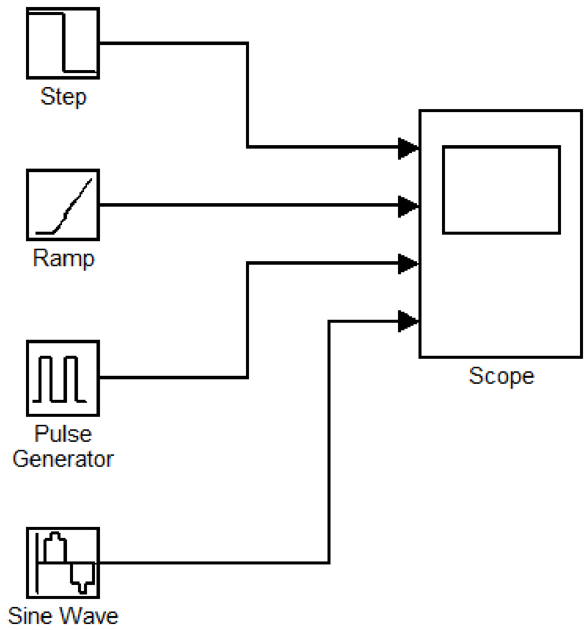
$$x(t) = ASin(2\pi F \pm \phi)$$

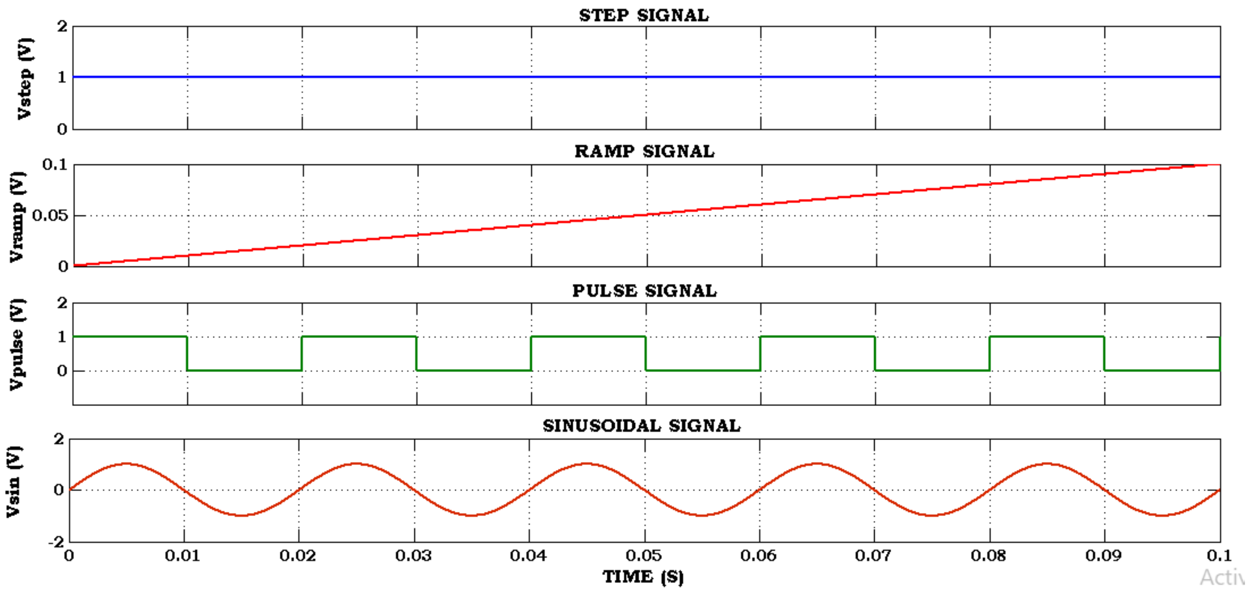

STEP SIGNAL
AMPLITUDE
Step
Time
RAMP SIGNAL
AMPLITUDE
Ramp
Time
IMPULSE SIGNAL
AMPLITUDE
Pulse Generator
Time
SINE WAVE
AMPLITUDE
Sine Wave
Time

Result:

EXPERIMENT NO-1.1

GENERATION OF STANDARD SIGNALS





Standard Signal Results

EXPERIMENT NO-2

Solving the linear and nonlinear differential equations

AIM: To solve the linear and non linear differential equations by using MAT LAB

First-Order Linear ODE

Solve this differential equation.

$$\frac{dy}{dt} = ty.$$

First, represent y by using `syms` to create the symbolic function `y(t)`.

```
syms y(t)
```

Define the equation using `==` and represent differentiation using the `diff` function.

```
ode = diff(y,t) == t*y
```

```
ode(t) =
diff(y(t), t) == t*y(t)
```

Solve the equation using `dsolve`.

```
ySol(t) = dsolve(ode)
```

```
ySol(t) =
C1*exp(t^2/2)
```

Solve Differential Equation with Condition

In the previous solution, the constant `C1` appears because no condition was specified. Solve the equation with the initial condition `y(0) == 2`. The `dsolve` function finds a value of `C1` that satisfies the condition.

```
cond = y(0) == 2;
ySol(t) = dsolve(ode,cond)
```

```
ySol(t) =
2*exp(t^2/2)
```

If `dsolve` cannot solve your equation, then try solving the equation numerically. See Solve a Second-Order Differential Equation Numerically.

Solve this nonlinear differential equation with an initial condition. The equation has multiple solutions.

$$\left(\frac{dy}{dt}+y\right)^2=1,$$
$$y(0)=0.$$

```
syms y(t)
ode = (diff(y,t)+y)^2 == 1;
cond = y(0) == 0;
ySol(t) = dsolve(ode,cond)
```

```
ySol(t) =
 exp(-t) - 1
 1 - exp(-t)
```

Second-Order ODE with Initial Conditions

Solve this second-order differential equation with two initial conditions.

$$\frac{d^2y}{dx^2}=\cos(2x)-y,$$
$$y(0)=1,$$
$$y'(0)=0.$$

Define the equation and conditions. The second initial condition involves the first derivative of `y`. Represent the derivative by creating the symbolic function `Dy = diff(y)` and then define the condition using `Dy(0)==0`.

```
syms y(x)
Dy = diff(y);

ode = diff(y,x,2) == cos(2*x)-y;
cond1 = y(0) == 1;
cond2 = Dy(0) == 0;
```

Solve `ode` for `y`. Simplify the solution using the `simplify` function.

```
conds = [cond1 cond2];
ySol(x) = dsolve(ode,conds);
ySol = simplify(ySol)
```

```
ySol(x) =
1 - (8*sin(x/2)^4)/3
```

Third-Order ODE with Initial Conditions

Solve this third-order differential equation with three initial conditions.

$$\frac{d^3u}{dx^3}=u,$$
$$u(0)=1,$$
$$u'(0)=-1,$$
$$u''(0)=\pi.$$

Because the initial conditions contain the first- and second-order derivatives, create two symbolic functions, `Du =`

`diff(u,x)` and `D2u = diff(u,x,2)`, to specify the initial conditions.

```
syms u(x)
Du = diff(u,x);
D2u = diff(u,x,2);
```

Create the equation and initial conditions, and solve it.

```
ode = diff(u,x,3) == u;
cond1 = u(0) == 1;
cond2 = Du(0) == -1;
cond3 = D2u(0) == pi;
conds = [cond1 cond2 cond3];

uSol(x) = dsolve(ode,conds)
```

```
uSol(x) =

(pi*exp(x))/3 - exp(-x/2)*cos((3^(1/2)*x)/2)*(pi/3 - 1) -...
(3^(1/2)*exp(-x/2)*sin((3^(1/2)*x)/2)*(pi + 1))/3
```

Differential Equation	MATLAB® Commands

$\frac{dy}{dt} + 4y(t) = e^{-t},$
$y(0) = 1.$
	`syms y(t)`
	`ode = diff(y)+4*y == exp(-t);`
	`cond = y(0) == 1;`
	`ySol(t) = dsolve(ode,cond)`
	`ySol(t) =`
	`exp(-t)/3 + (2*exp(-4*t))/3`

$2x^2\frac{d^2y}{dx^2} + 3x\frac{dy}{dx} - y = 0.$	`syms y(x)`
	`ode = 2*x^2*diff(y,x,2)+3*x*diff(y,x)-y == 0;`
	`ySol(x) = dsolve(ode)`
	`ySol(x) =`
	`C1/(3*x) + C2*x^(1/2)`

The Airy equation.
$\frac{d^2y}{dx^2} = xy(x).$
	`syms y(x)`
	`ode = diff(y,x,2) == x*y;`
	`ySol(x) = dsolve(ode)`
	`ySol(x) =`
	`C1*airy(0,x) + C2*airy(2,x)`

Differential Equation	MATLAB® Commands

Puiseux series solution.
$(x^2+1)\frac{d^2y}{dx^2} - 2x\frac{dy}{dx} + y = 0.$
	`syms y(x) a`
	`ode = (x^2+1)*diff(y,x,2)-2*x*diff(y,x)+y == 0;`
	`Dy = diff(y,x);`
	`cond = [Dy(0) == a; y(0) == 5];`
	`ySol(x) = dsolve(ode,cond,'ExpansionPoint',0)`
	`ySol(x) =`
	`- (a*x^5)/120 - (5*x^4)/24 + (a*x^3)/6 - (5*x^2)/2 + a*x + 5`

Results:

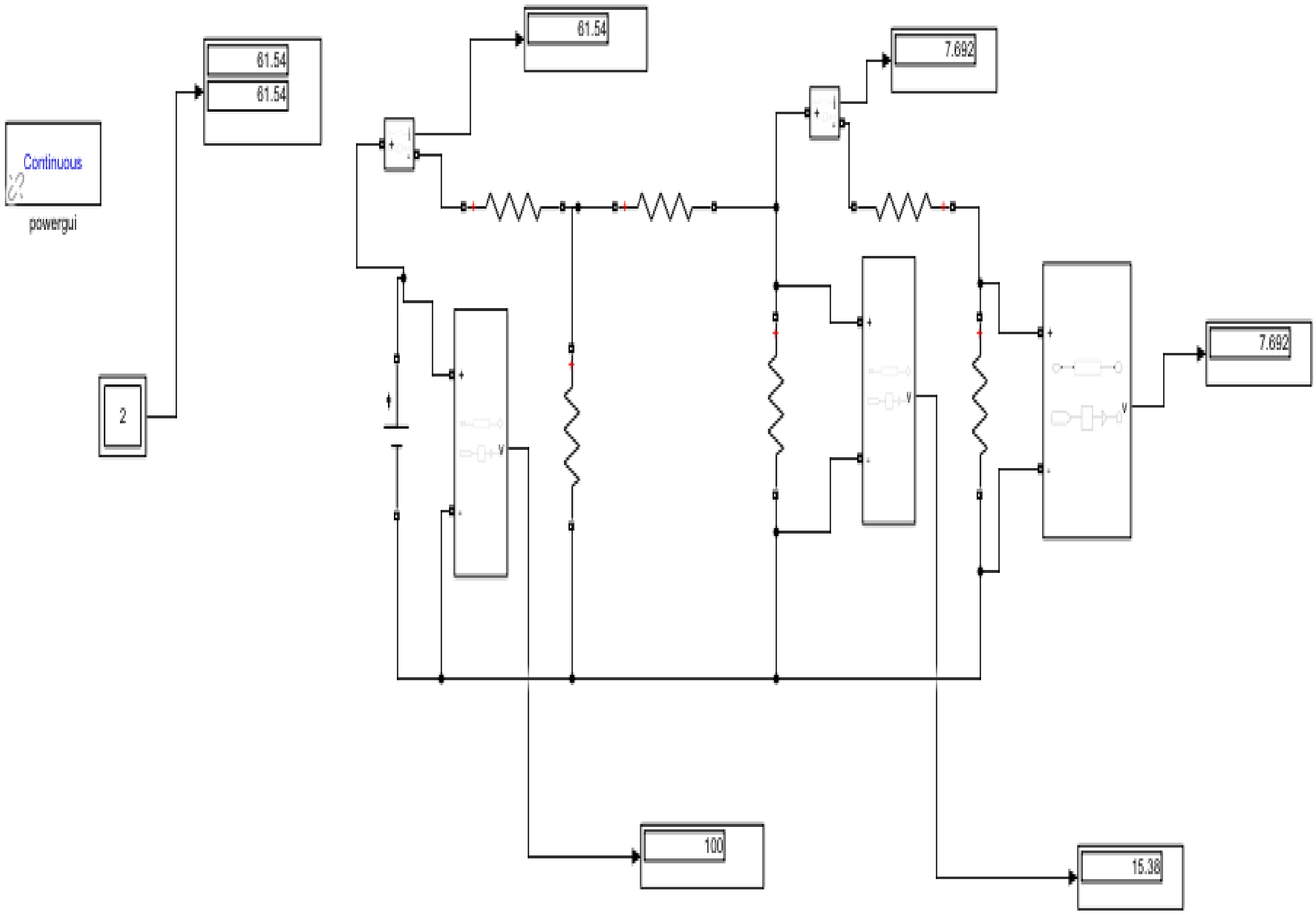
EXPERIMENT NO-3

MEASUREMENT OF VOLTAGE, CURRENT IN DC CIRCUITS

AIM: To measure Voltage and current in Dc circuits using MATLAB Simulation.

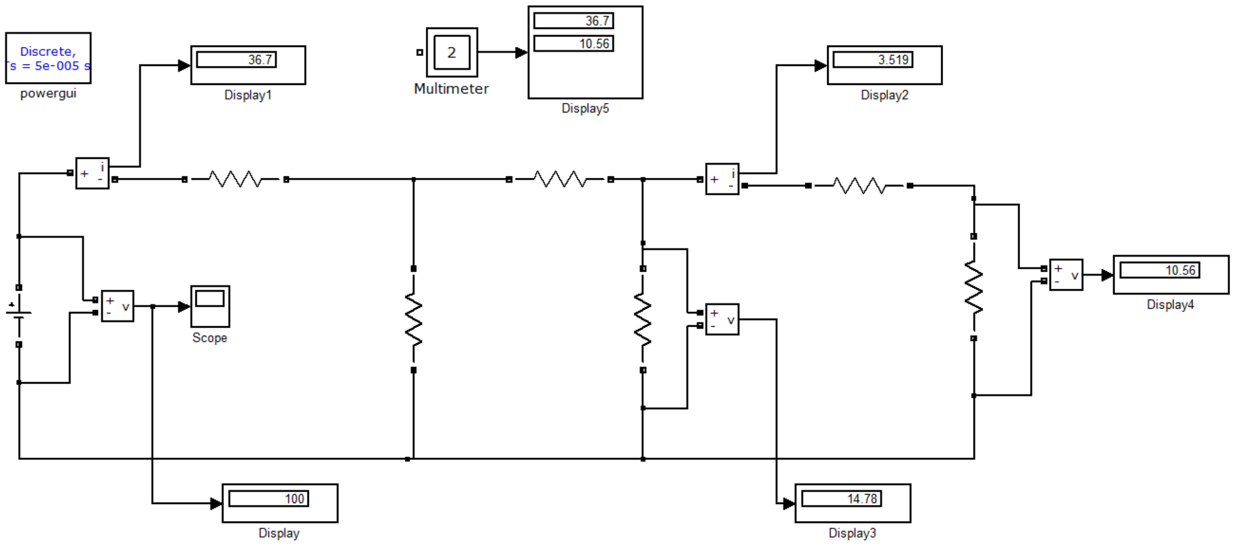
Software Required: MATLAB software.

Simulation Diagram:



Result:

EXPERIMENT NO-3
MEASUREMENT OF VOLTAGE, CURRENT IN DC CIRCUITS



Results:

EXPERIMENT NO-4

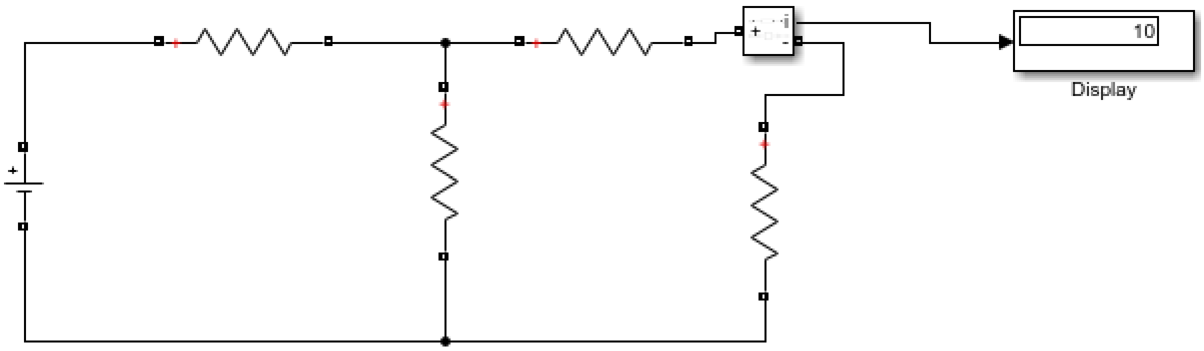
VERIFICATION OF NETWORK THEOREMS

AIM: To verify Thevenin's theorem, Norton's theorem. SOFTWARE USED: MATLAB Simulink THEVENIN'S THEOREM:

Procedure:

Step 1:

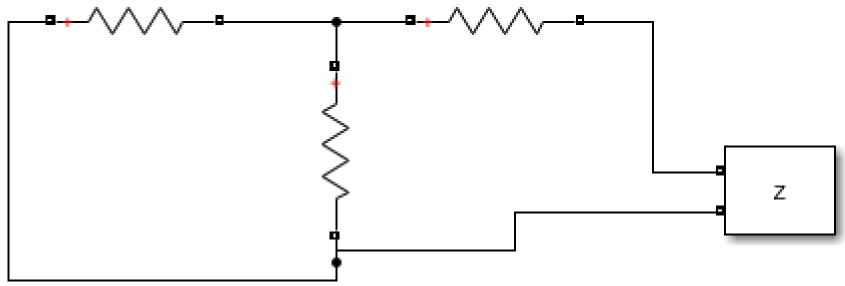
1. Measure the response 'I' in the load resistor by considering all the sources in the network.



Step 2:

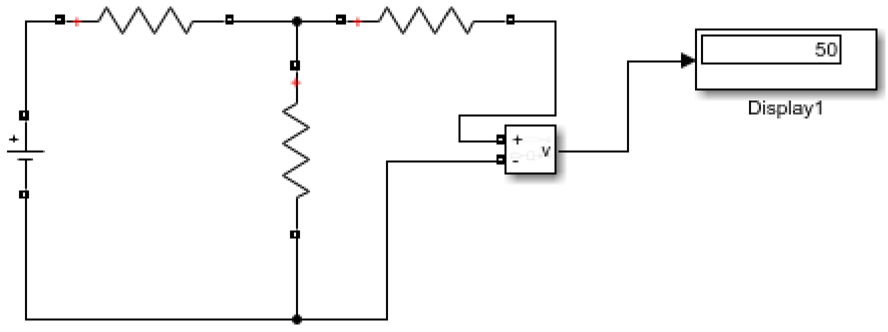
Finding Thevenin's Resistance (Rth)

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 (Vth)

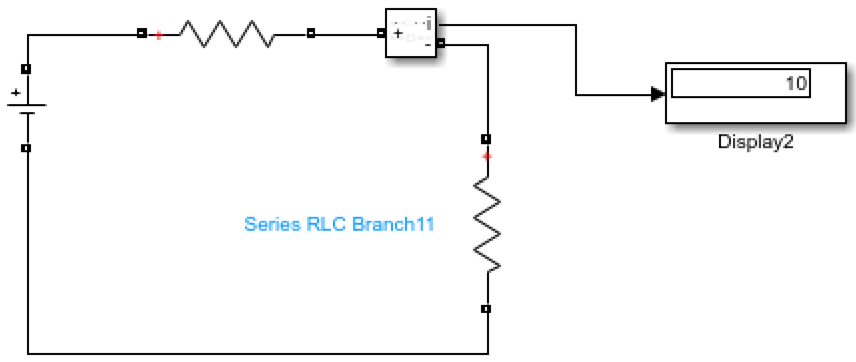
1. Open the load terminals and measure the voltage across the open circuited terminals.
2. Measured voltage will be known as Thevenin's Voltage.



Step 4: Thevenin's Equivalent Circuit

1. Vth and Rth are connected in series with the load.
2. Measure the current through the load resistor

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

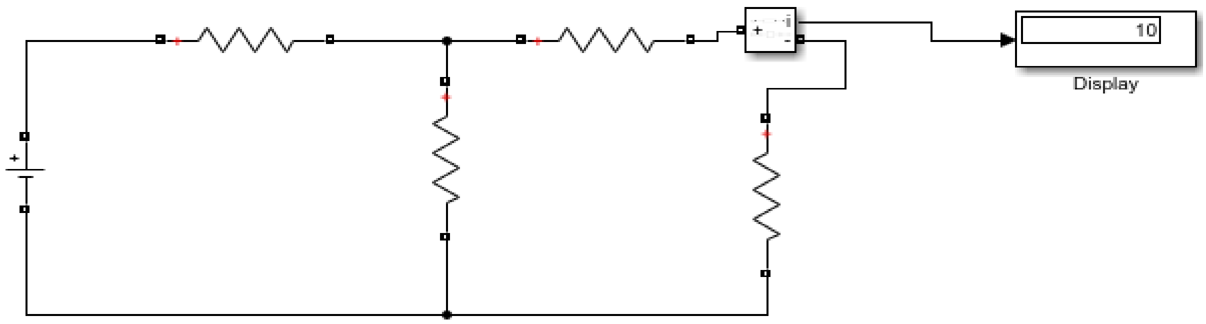


Result:

NORTON'S THEOREM:

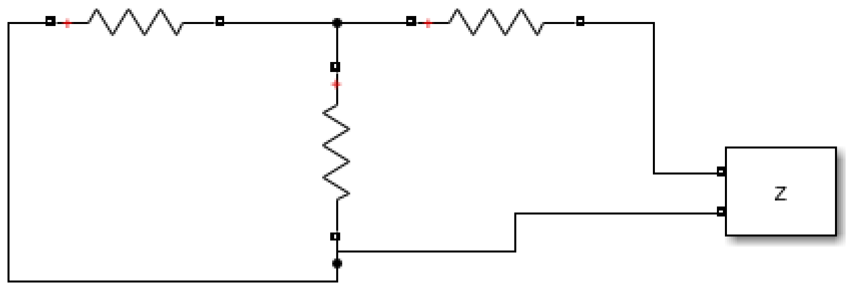
Procedure:

Step 1: 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'sResistance.



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

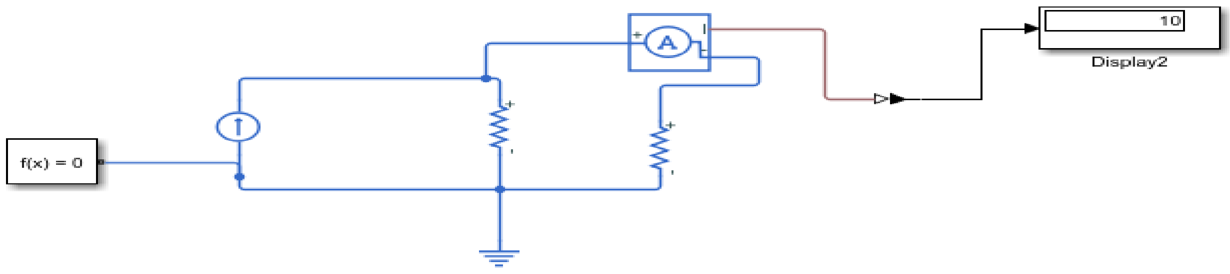
Norton's Current.Step 4: Norton's

Equivalent Circuit

1. R_N and I_N are connected in parallel to the load.

Measure the current through the load resistor

$I_L = (I_N * R_N) / (R_L + R_{TH})$



Result:

EXPERIMENT NO-5

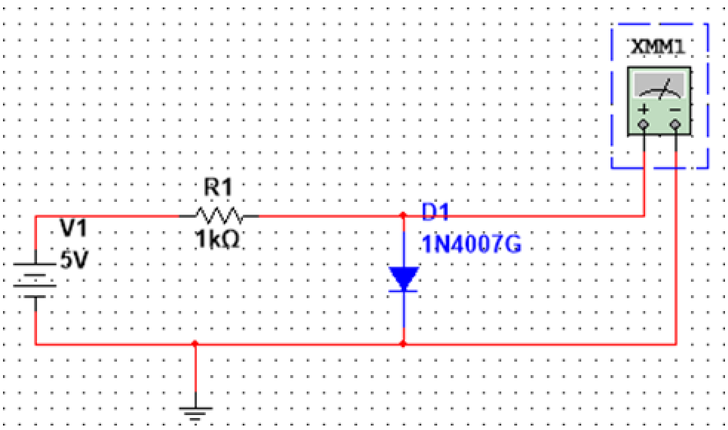
Verification of performance characteristics of basic Electronic Devices using suitable simulation tools

A) PN junction diode: when an n-type semiconductor is joined with the p-type semiconductor, a p-n junction is formed. The region where the p-type and n-type semiconductors are joined is called p-n junction. It is also defined as the boundary between p-type and n-type semiconductor. This p-n junction forms a most popular semiconductor device known as diode.

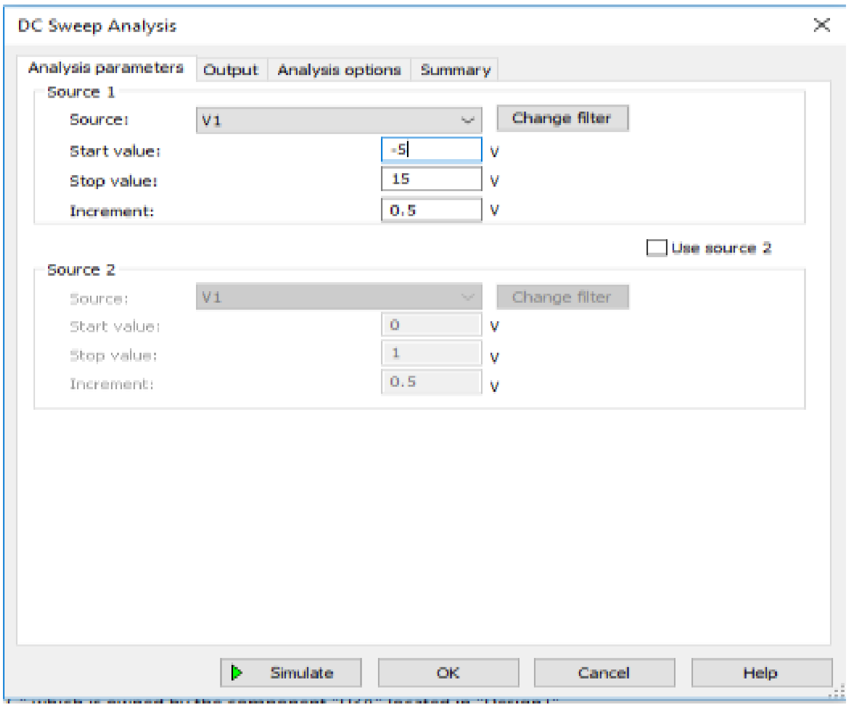
B) Zener Diode: A zener diode is a special type of device designed to operate in the zener breakdown region. Zener diodes acts like normal p-n junction diodes under forward biased condition. When forward biased voltage is applied to the zener diode it allows large amount of electric current and blocks only a small amount of electric current. Zener diode is heavily doped than the normal p-n junction diode. Hence, it has very thin depletion region. Therefore, zener diodes allow more electric current than the normal p-n junction diodes.

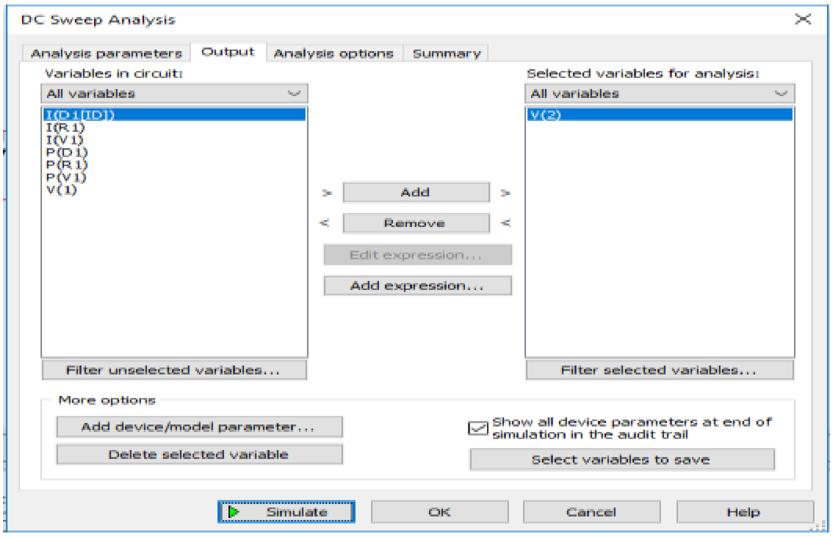
Zener diode allows electric current in forward direction like a normal diode but also allows electric current in the reverse direction if the applied reverse voltage is greater than the zener voltage. Zener diode is always connected in reverse direction because it is specifically designed to work in reverse direction.

1. Connect the circuit as mentioned in the below diagram

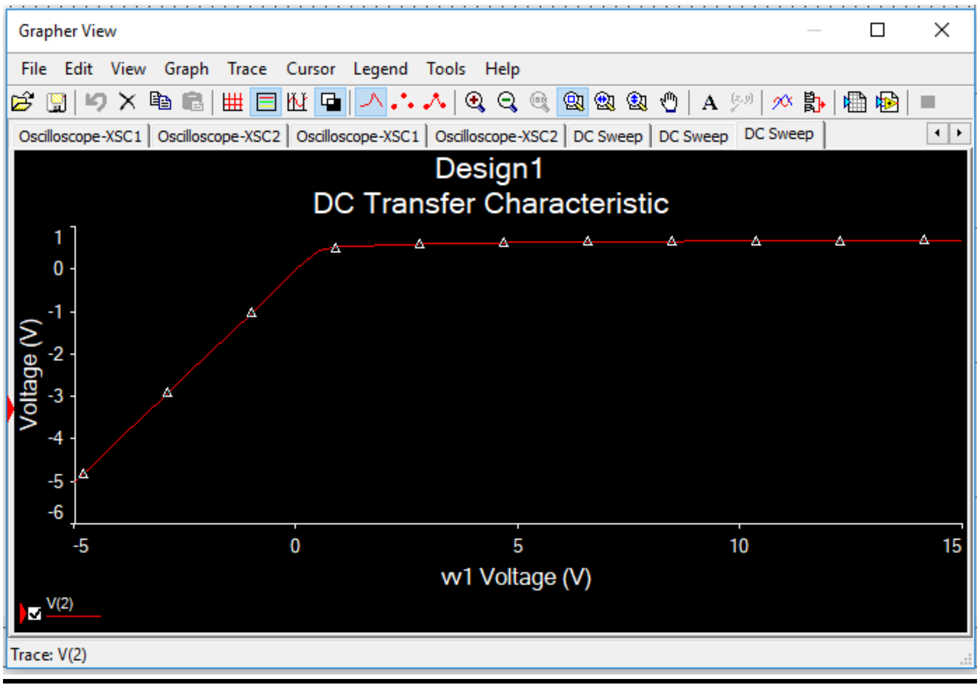


2. Output can be calculated in the multi meter and to get the exact response for the entire circuit we can go for the DC sweep analysis and select the starting and stop voltage to plot the characteristics of the PN junction Diode. For that we have to go to simulation tab and then select analysis from that window you have to select dc sweep analysis and then you have to we have to enter the point where we want to calculate the output value at which node after selecting the node then click on simulate option.


DC Sweep Analysis
Analysis parameters
Output
Analysis options
Summary
Source 1
Source:
V1
Change filter
Start value:
-5
V
Stop value:
15
V
Increment:
0.5
V
Use source 2
Source 2
Source:
V1
Change filter
Start value:
0
V
Stop value:
1
V
Increment:
0.5
V
Simulate
OK
Cancel
Help


DC Sweep Analysis
Analysis parameters
Output
Analysis options
Summary
Variables in circuit:
All variables
I(D1[ID])
I(R1)
I(V1)
P(D1)
P(R1)
P(V1)
V(1)
Selected variables for analysis:
All variables
V(2)
Add
Remove
Edit expression...
Add expression...
Filter unselected variables...
Filter selected variables...
More options
Add device/model parameter...
Delete selected variable
Show all device parameters at end of simulation in the audit trail
Select variables to save
Simulate
OK
Cancel
Help

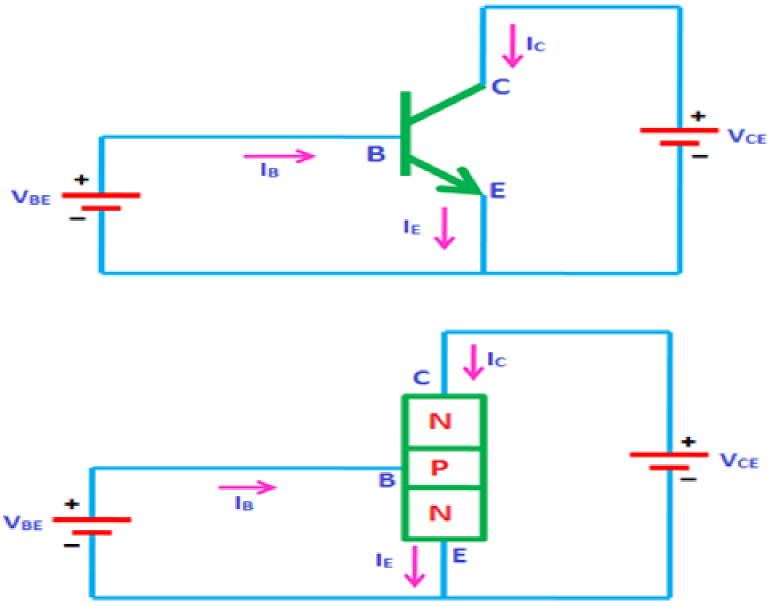
1. After clicking on the simulate option we will get a graph which will plot the characteristics of the PN junction diode with the voltages range mentioned.



Results:

Simulation of Input and output characteristics of Transistor in CE configuration.

In common emitter configuration, base is the input terminal, collector is the output terminal and emitter is the common terminal for both input and output. That means the base terminal and common emitter terminal are known as input terminals whereas collector terminal and common emitter terminal are known as output terminals.
In common emitter configuration, the emitter terminal is grounded so the common emitter configuration is also known as grounded emitter configuration. Sometimes common emitter configuration is also referred to as CE configuration, common emitter amplifier, or CE amplifier. The common emitter (CE) configuration is the most widely used transistor configuration.



Common emitter configuration

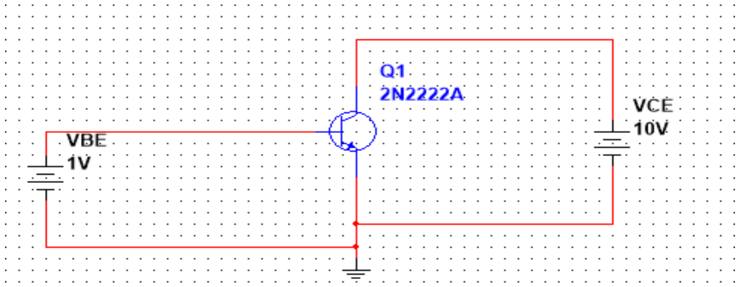
In common emitter configuration, the emitter terminal is grounded so the common emitter configuration is also known as grounded emitter configuration. The common emitter (CE) amplifiers are used when large current gain is needed.

The input signal is applied between the base and emitter terminals while the output signal is taken between the collector and emitter terminals. Thus, the emitter terminal of a transistor is common for both input and output and hence it is named as common emitter configuration.

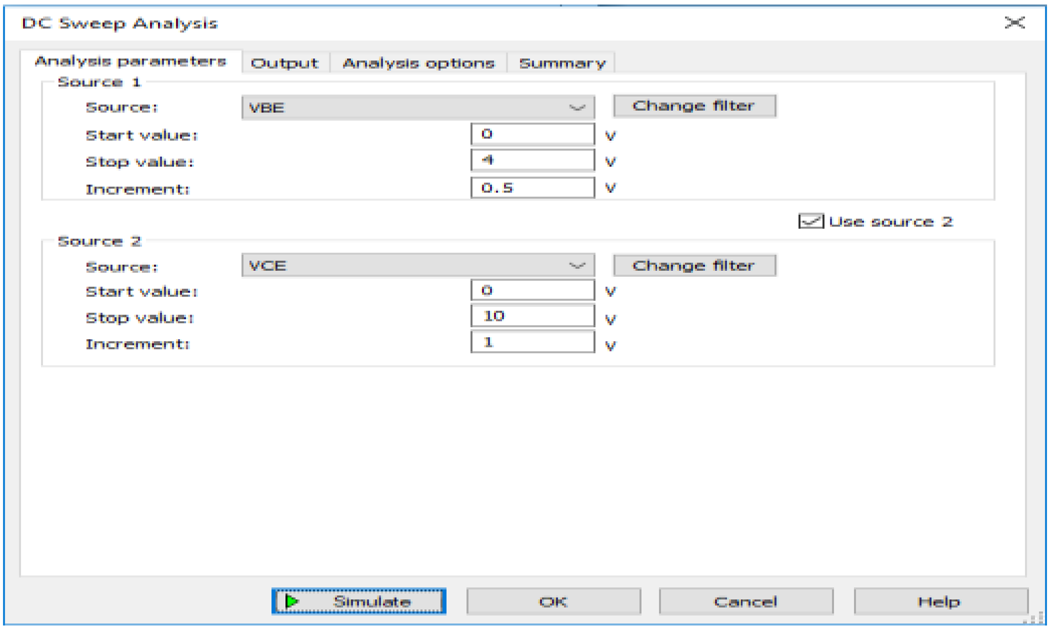
The supply voltage between base and emitter is denoted by VBE while the supply voltage between collector and emitter is denoted by VCE. In common emitter (CE) configuration, input current or base current is denoted by IB and output current or collector current is denoted by IC.

Multisim Steps:

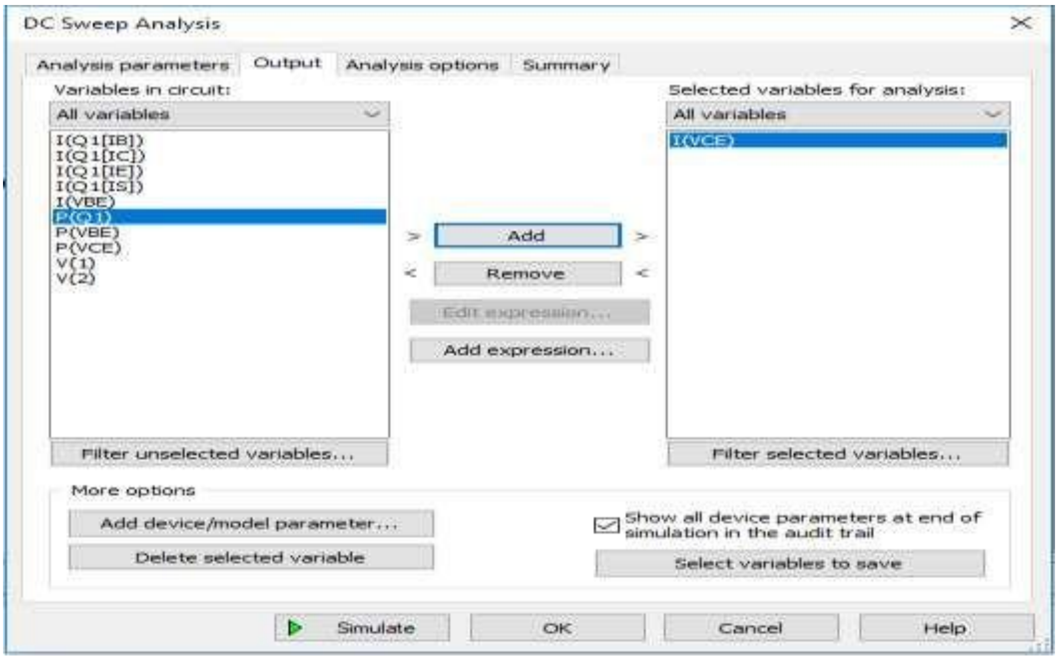
1. Connect the circuit as shown in the figure.

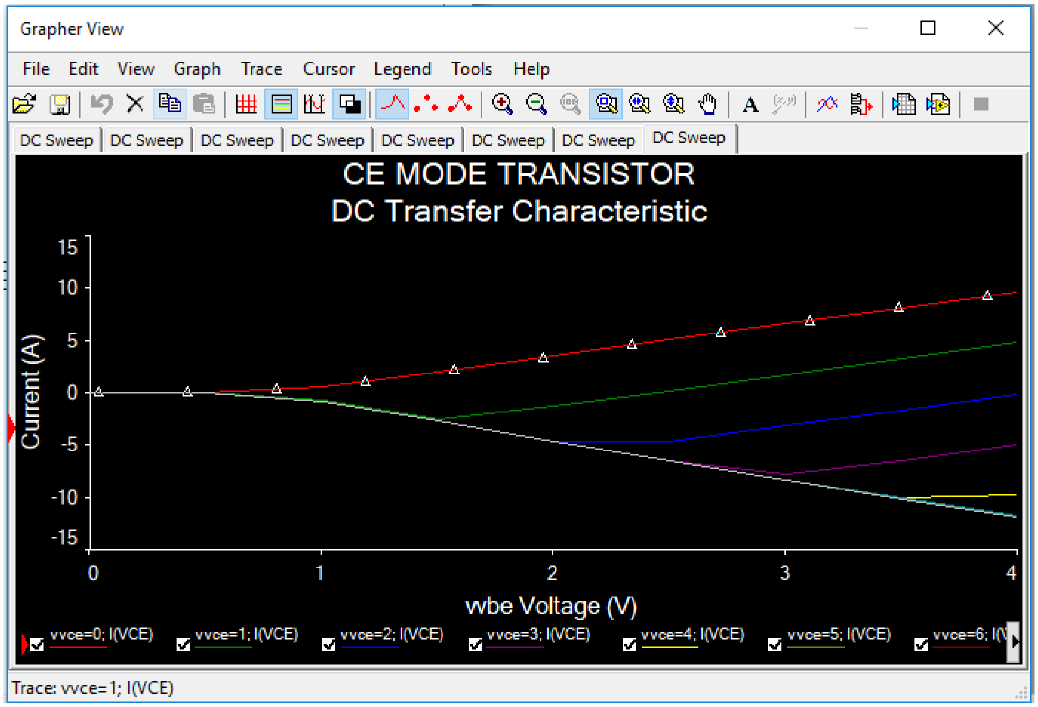


2. For transistors go to transistor family and then select BJT NPN in that library search for the 2n222a transistor. After connecting in this format select the analysis under the simulation tab and then select the dc sweep analysis and then enter the two voltage sources and ranging values then the plots will be plotted for multiple voltages.

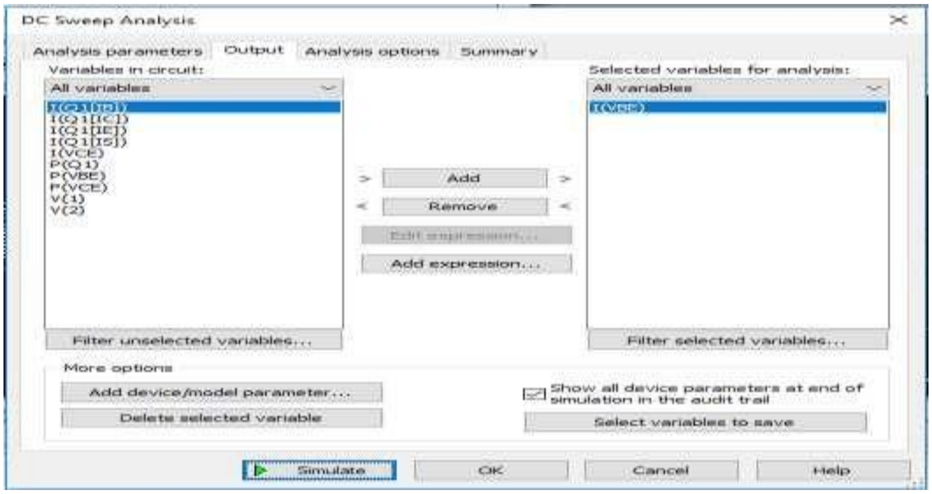


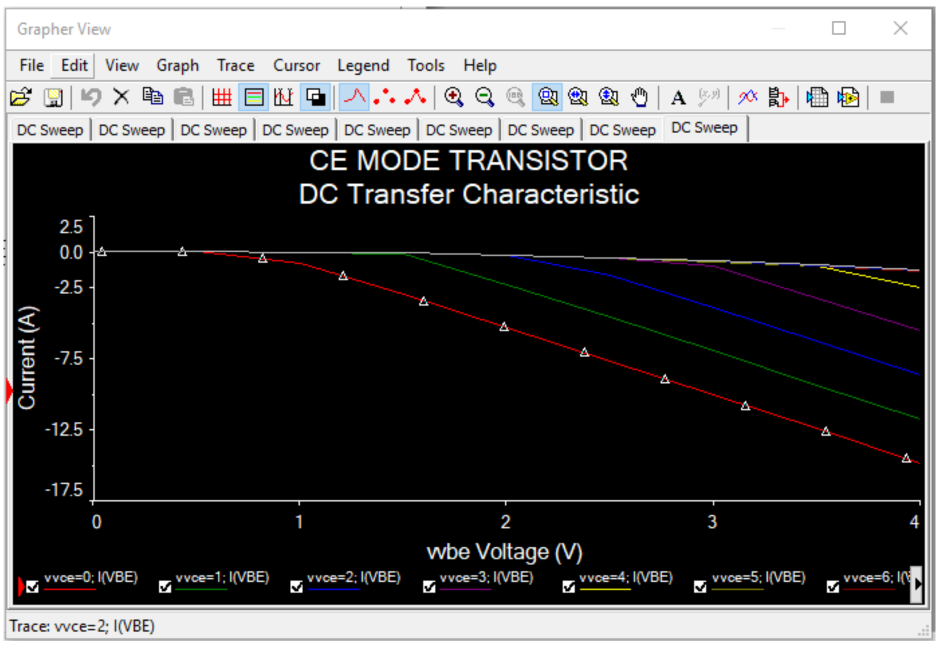
For output characteristics





For input characteristics:


DC Sweep Analysis
Analysis parameters
Output
Analysis options
Summary
Variables in circuit:
All variables
I(Q1[IB])
I(Q1[IC])
I(Q1[IE])
I(Q1[IS])
I(VCE)
P(Q1)
P(VBE)
P(VCE)
V(1)
V(2)
Add
Remove
Edit expression...
Add expression...
Selected variables for analysis:
All variables
I(VBE)
Filter unselected variables...
Filter selected variables...
More options
Add device/model parameter...
Delete selected variable
Show all device parameters at end of simulation in the audit trail
Select variables to save
Simulate
OK
Cancel
Help


Grapher View
File
Edit
View
Graph
Trace
Cursor
Legend
Tools
Help
DC Sweep
CE MODE TRANSISTOR
DC Transfer Characteristic
2.5
0.0
-2.5
-7.5
-12.5
-17.5
Current (A)
0
1
2
3
4
vbe Voltage (V)
vvce=0; I(VBE)
vvce=1; I(VBE)
vvce=2; I(VBE)
vvce=3; I(VBE)
vvce=4; I(VBE)
vvce=5; I(VBE)
Trace: vvce=2; I(VBE)

EXPERIMENT NO-6

SERIES AND PARALLEL RESONANCE

SERIES RESONANCE:

Aim: - To obtain the plot of frequency versus XL, frequency versus XC, frequency impedance and frequency vs. current for the given series RLC circuit and determine the resonant frequency and check by theoretical calculations.

R = 15Ω, C = 10 μ F, L = 0.1 H, V = 50V vary frequency in steps of 1 Hz using MATLAB. clc;

```
clear all; close all;
r=input('enter the resistance value      >');
l=input('enter the inductance value      >');
c=input('enter the capacitance value     >');
v=input('enter the input voltage >');
f=5:2:300;
xl=2*pi*f*l; xc=(1./(2*pi*f*c)); x=xl-xc; z=sqrt((r^2)+(x.^2)); i=v./z;
%plotting the graph subplot(2,2,1);
plot(f,xl); grid;
xlabel('frequency'); ylabel('X1');
subplot(2,2,2);
plot(f,xc); grid;
xlabel('frequency'); ylabel('Xc');
subplot(2,2,3);
plot(f,z); grid;
xlabel('frequency'); ylabel('Z');
subplot(2,2,4);
```

```
plot(f,i);
grid; xlabel('frequency'); ylabel('I');
```

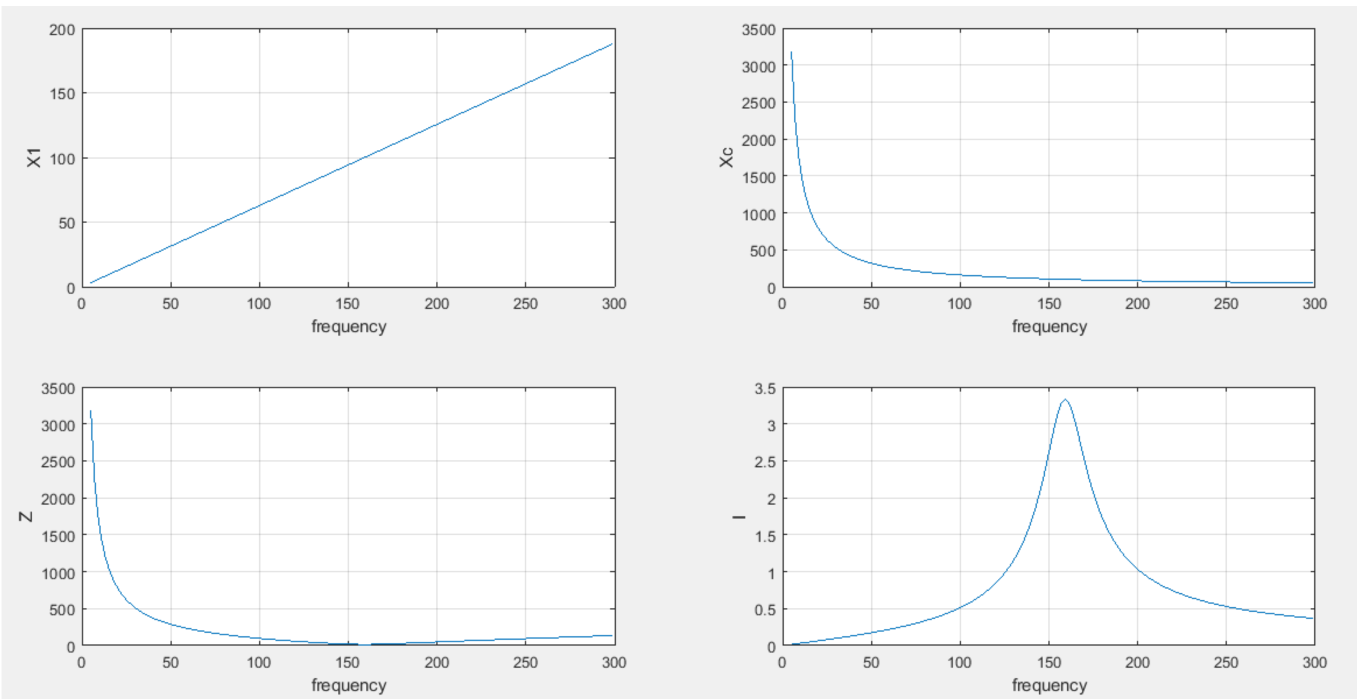
PROGRAM RESULT:

enter the resistance value >15

enter the inductance value >0.1

enter the capacitance value----->10*10^-6

enter the input voltage >50



Result:

PARALLEL RESONANCE: -

To obtain the graphs of frequency vs. BL, frequency vs. BC, frequency vs. admittance and frequency vs. current vary frequency in steps for the given circuit and find the resonant frequency and check by theoretical calculations.

R = 1000Ω , C = 400 μ F, L = 1 H, V = 50V vary frequency in steps of 1 Hz using MATLAB.

```
clc; clear all; close all;
r=input('enter the resistance value      >');
l=input('enter the inductance value      >');
c=input('enter the capacitance value    >');
v=input('enter the input voltage >');
f=0:2:50;
xl=2*pi*f*l; xc=(1./(2*pi*f*c)); b1=1./xl; bc=1./xc;
b=b1-bc; g=1/r;
y=sqrt((g^2)+(b.^2)); i=v*y;
%plotting the graph subplot(2,2,1);
plot(f,b1); grid;
xlabel('frequency'); ylabel('B1');
subplot(2,2,2);
plot(f,bc); grid;
xlabel('frequency'); ylabel('Bc');
subplot(2,2,3);
plot(f,y); grid;
xlabel('frequency'); ylabel('Y');
subplot(2,2,4);
```

plot(f,i); grid;

xlabel('frequency'); ylabel('I');

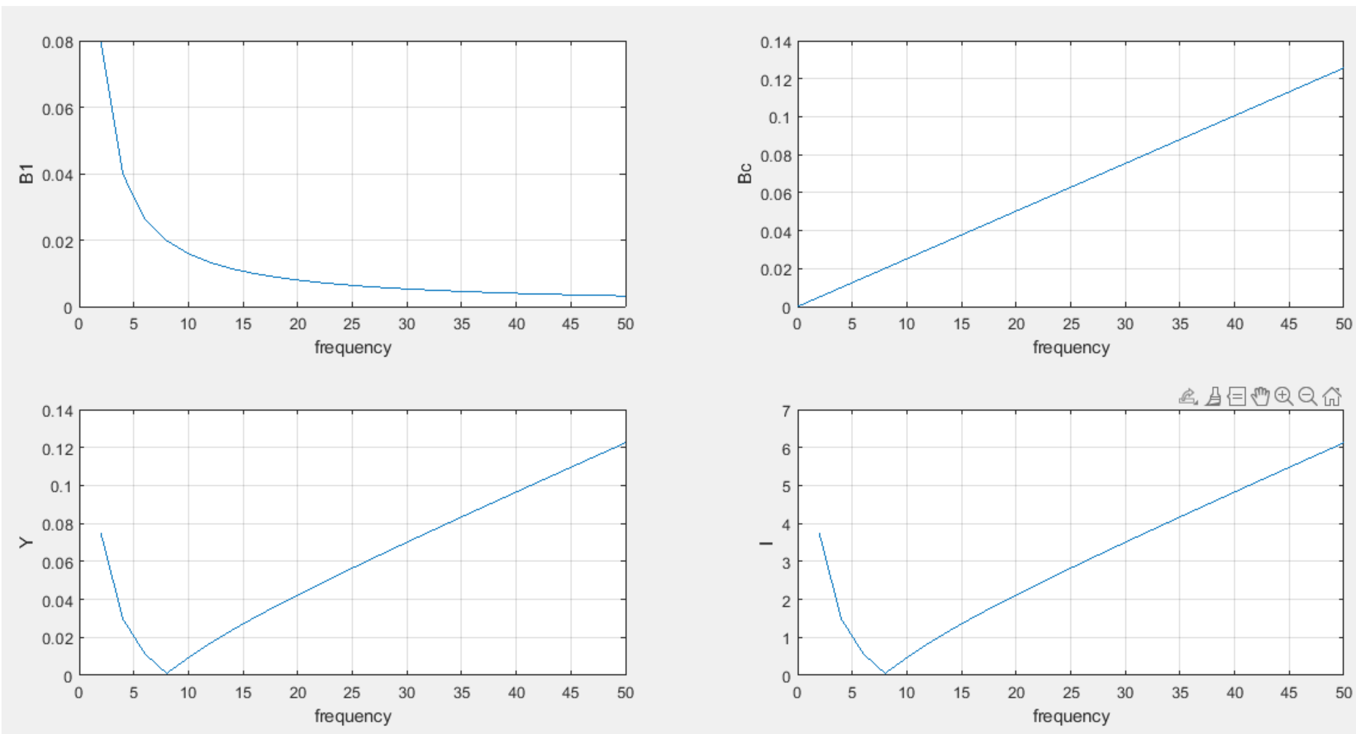
PROGRAM RESULT:

enter the resistance value >1000

enter the inductance value >1

enter the capacitance value----->400*10^-6

enter the input voltage >50

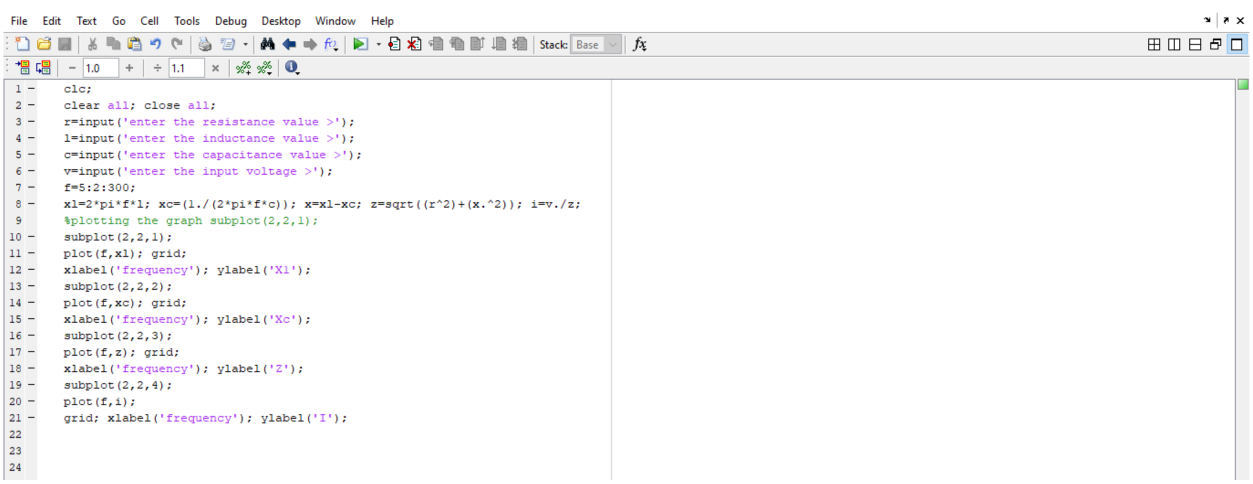


Result:

EXPERIMENT NO-6
SERIES AND PARALLEL RESONANCE

1. Series Resonance Circuit Programming:

```
clc;
clearall; close all;
r=input('enter the resistance value >');
l=input('enter the inductance value >');
c=input('enter the capacitance value >');
v=input('enter the input voltage >');
f=5:2:300;
xl=2*pi*f*l; xc=(1./(2*pi*f*c)); x=xl-xc; z=sqrt((r^2)+(x.^2)); i=v./z;
%plotting the graph subplot(2,2,1);
subplot(2,2,1);
plot(f,xl); grid;
xlabel('frequency'); ylabel('X1');
subplot(2,2,2);
plot(f,xc); grid;
xlabel('frequency'); ylabel('Xc');
subplot(2,2,3);
plot(f,z); grid;
xlabel('frequency'); ylabel('Z');
subplot(2,2,4);
plot(f,i);
grid; xlabel('frequency'); ylabel('I');
```



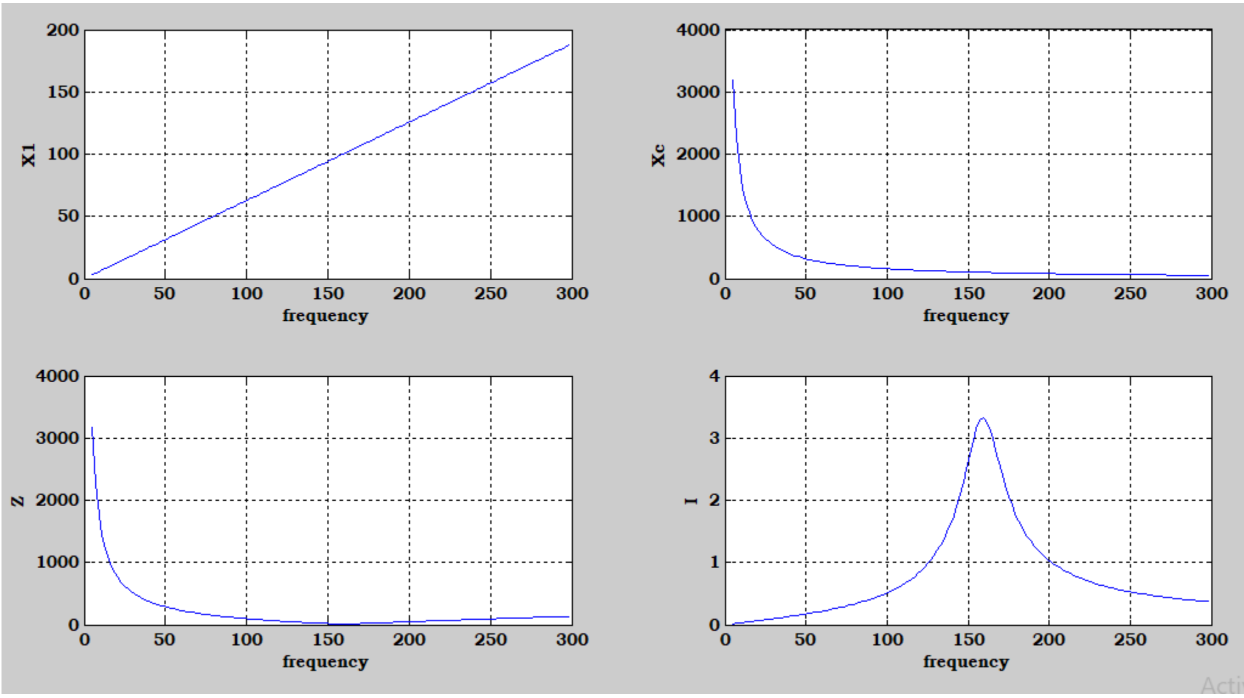
Then enter the variables in command window

Enter the resistance value >15

Enter the inductance value >0.1

Enter the capacitance value >10*10^-6

Enter the input voltage >50

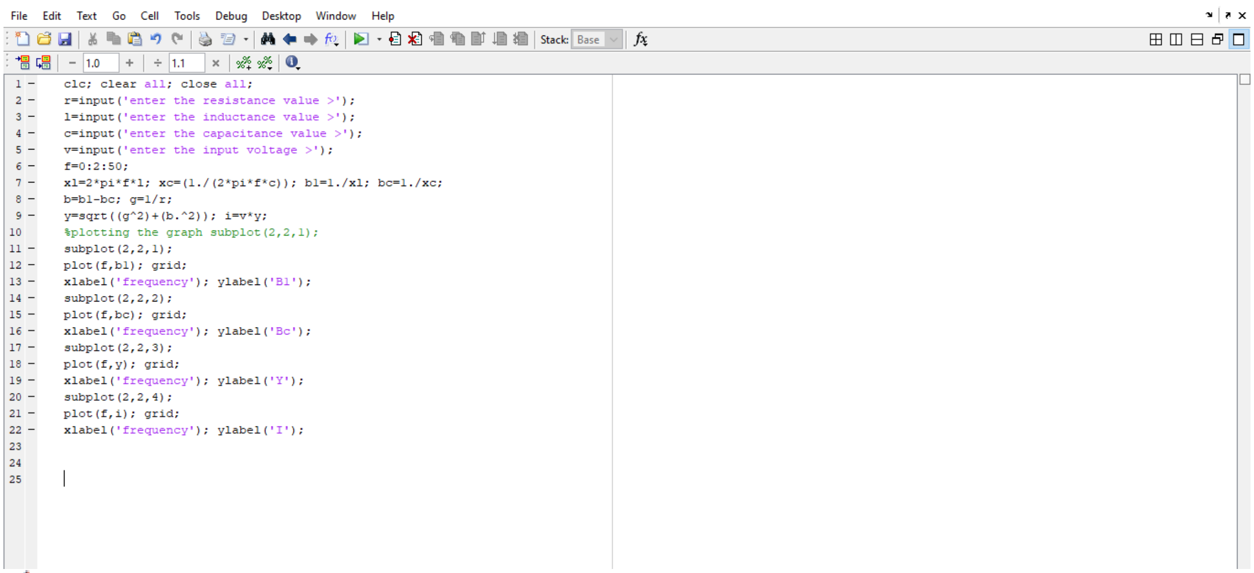


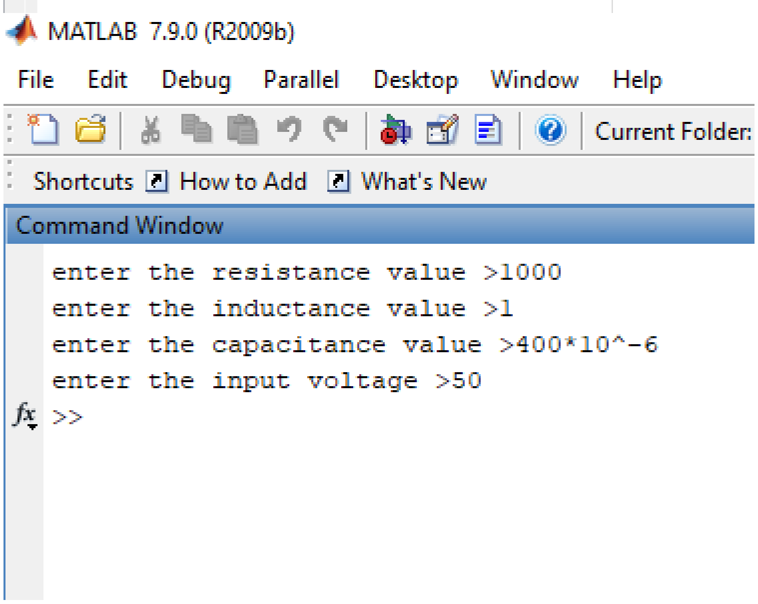
Series Resonance Result

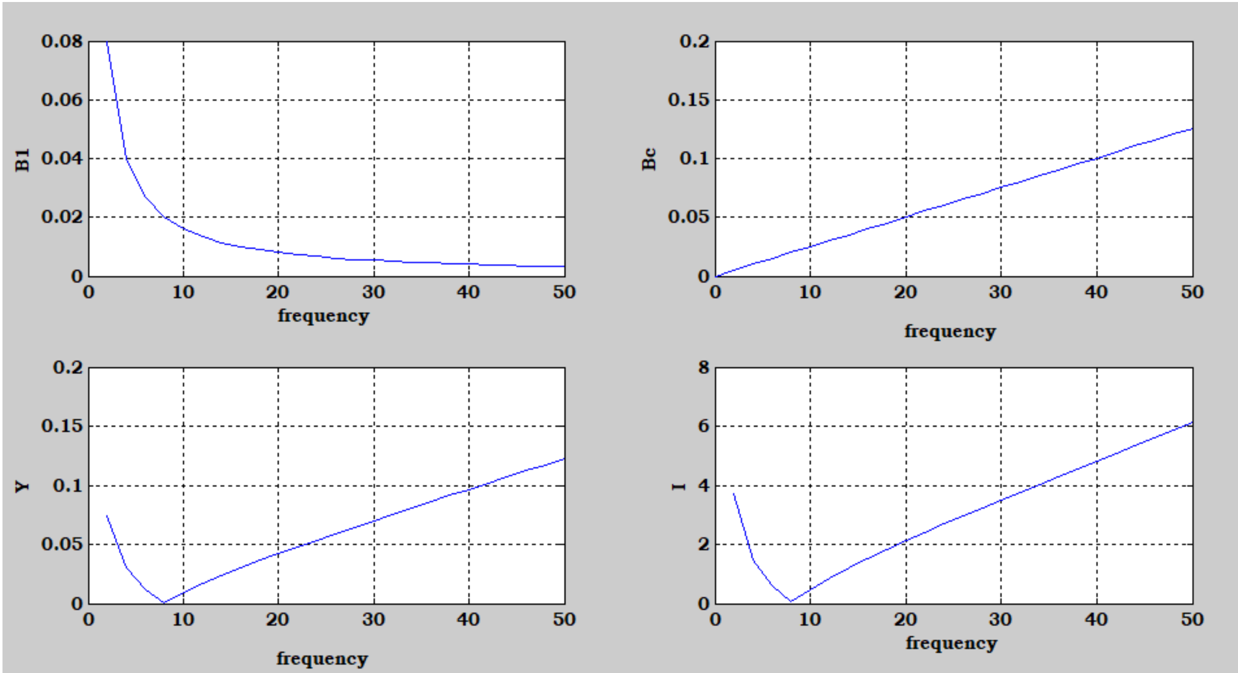
2. Parallel Resonance Circuit Programming:

```
clc; clear all; close all;
r=input('enter the resistance value >');
l=input('enter the inductance value >');
c=input('enter the capacitance value >');
v=input('enter the input voltage >');
f=0:2:50;
xl=2*pi*f*l; xc=(1./(2*pi*f*c)); b1=1./xl; bc=1./xc;
b=b1-bc; g=1/r;
y=sqrt((g^2)+(b.^2)); i=v*y;
%plotting the graph subplot(2,2,1);
subplot(2,2,1);
plot(f,b1); grid;
xlabel('frequency'); ylabel('B1');
subplot(2,2,2);
```

```
plot(f,bc); grid;
xlabel('frequency'); ylabel('Bc');
subplot(2,2,3);
plot(f,y); grid;
xlabel('frequency'); ylabel('Y');
subplot(2,2,4);
plot(f,i); grid;
xlabel('frequency'); ylabel('I');
```







Parallel Resonance Result

Results:

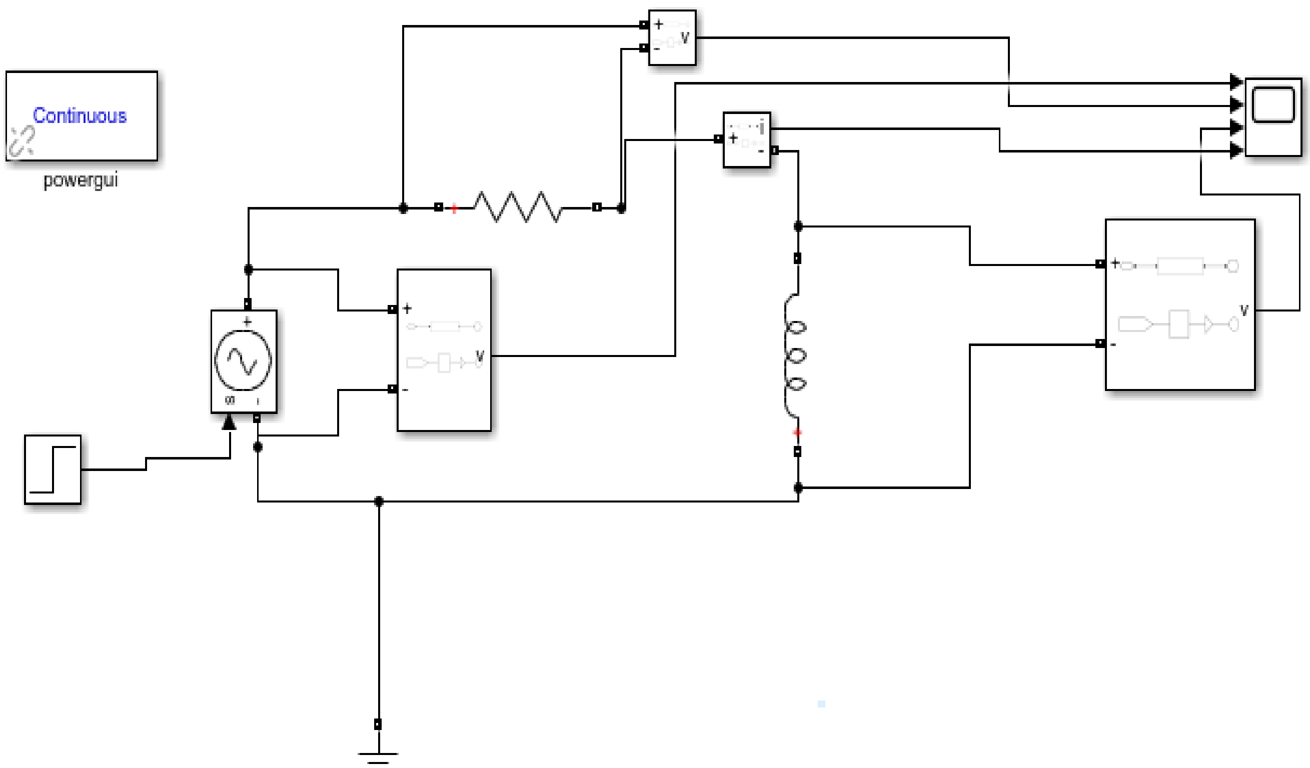
EXPERIMENT NO – 7

RESPONSE OF RL CIRCUIT WITH STEP SIGNAL

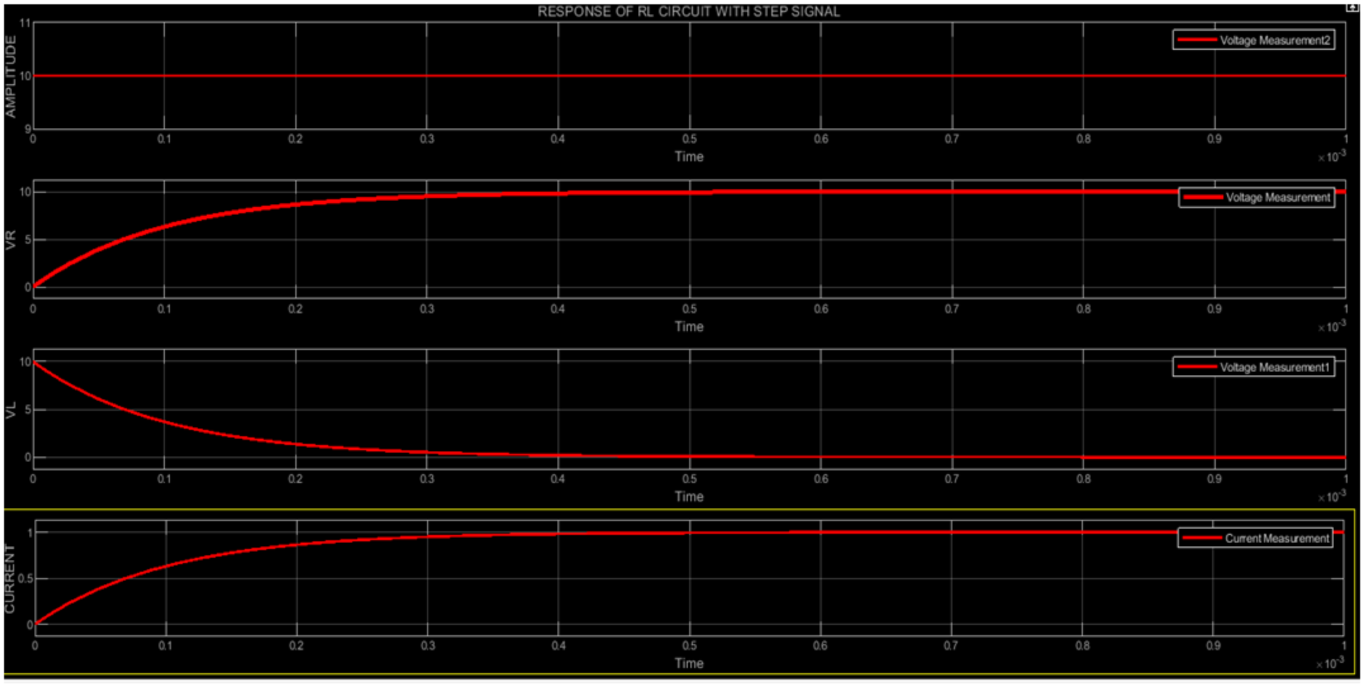
AIM: To Obtain the response of RL circuit with Step Signal.

Software Required: MATLAB software.

Simulation Diagram:



Simulation Output:

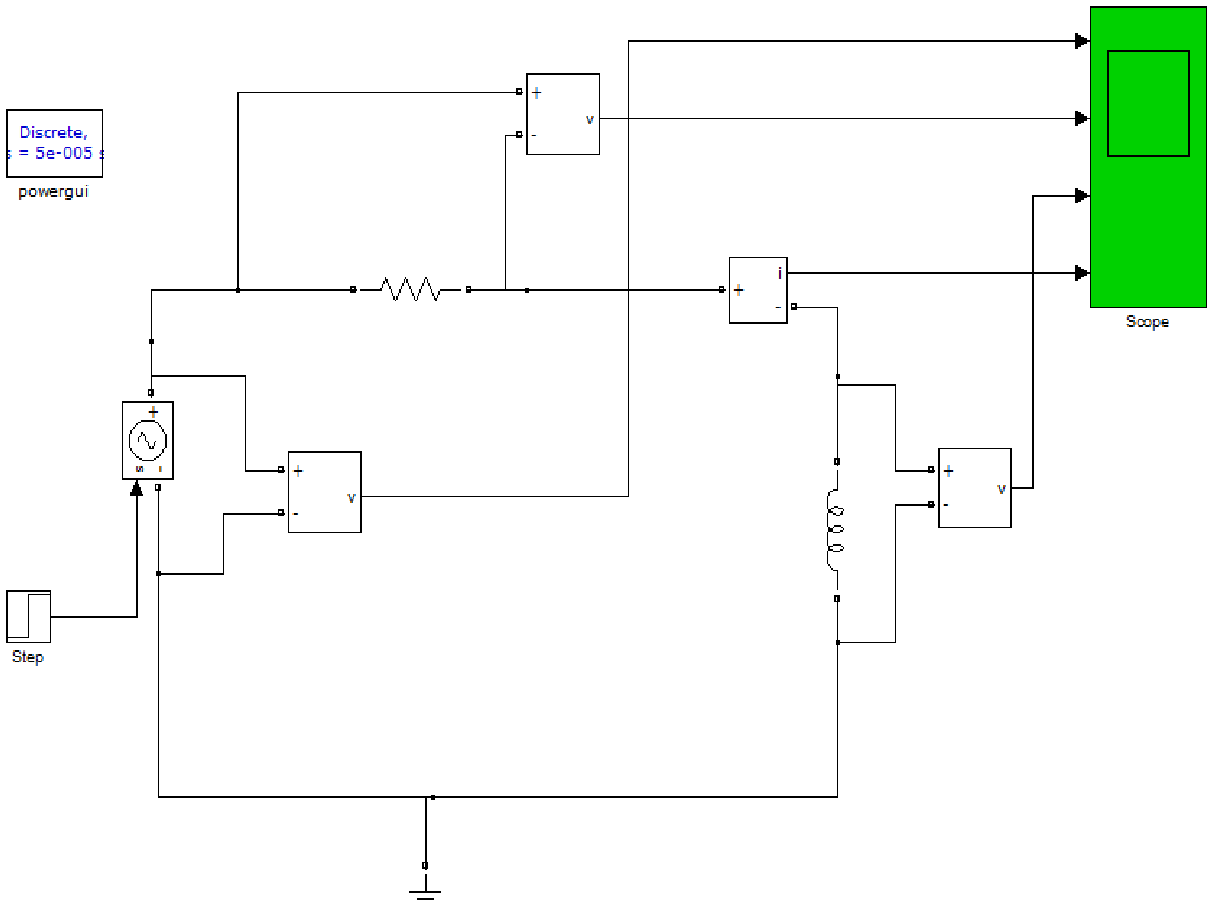


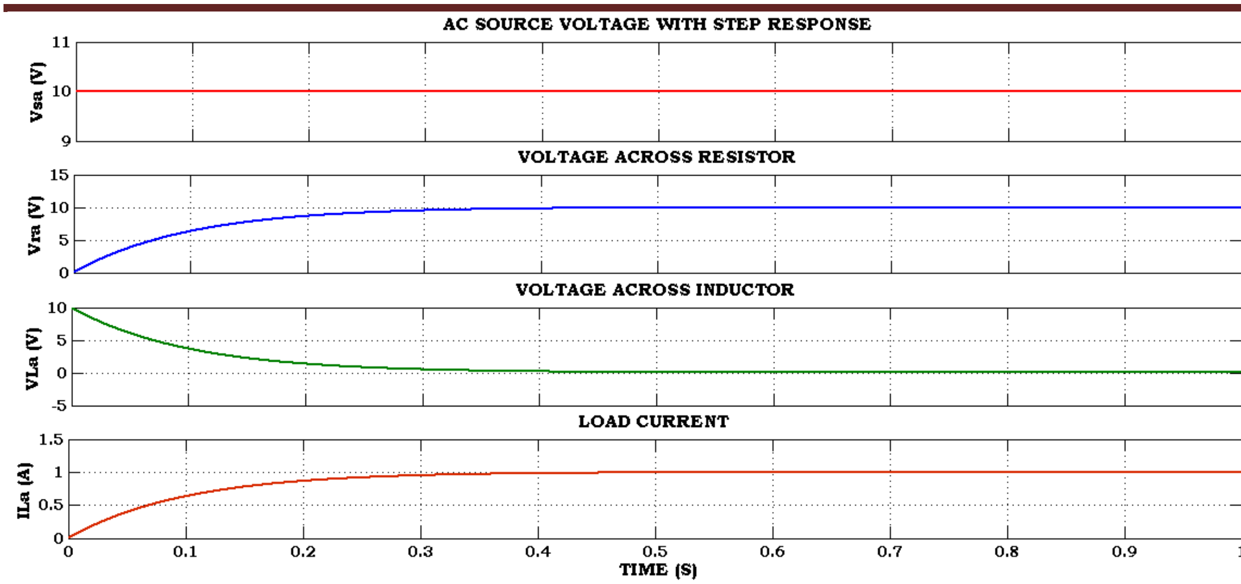
Results:

EXPERIMENT NO – 7

RESPONSE OF RL CIRCUIT WITH STEP SIGNAL

AIM: To find the response of RL circuit by using step signal and draw its characteristics.




AC SOURCE VOLTAGE WITH STEP RESPONSE
Vsa (V)
VOLTAGE ACROSS RESISTOR
Vra (V)
VOLTAGE ACROSS INDUCTOR
VLa (V)
LOAD CURRENT
ILa (A)
TIME (S)

EXPERIMENT-8

Simulation of Low pass and High pass filters

A) Low pass filters: A Low Pass Filter is a circuit that can be designed to modify, reshape or reject all unwanted high frequencies of an electrical signal and accept or pass only those signals wanted by the circuit's designer. A simple passive RC Low Pass Filter or LPF, can be easily made by connecting together in series a single Resistor with a single Capacitor as shown below. In this type of filter arrangement the input signal (V IN) is applied to the series combination (both the Resistor and Capacitor together) but the output signal (V OUT) istaken across the capacitor only.

B) High pass filters: A High Pass Filter is the exact opposite to the low pass filter circuit as the two components have been interchanged with the filters output signal now being taken from across the resistor. The reactance of the capacitor is very high at low frequencies so the capacitor acts like an open circuit and blocks any input signals at VIN until the cut-off frequency point (fc) is reached. Above this cut-off frequency point the reactance of the capacitor has reduced sufficiently as to now act more like a short circuit allowing all of the input signal to pass directly to the output as shown below in the filters response curve.

So high pass filters will allow only high frequencies to pass through it and low pass filters work exactly opposite to high pass filters it will allow only the frequencies less than the cut off frequencies.

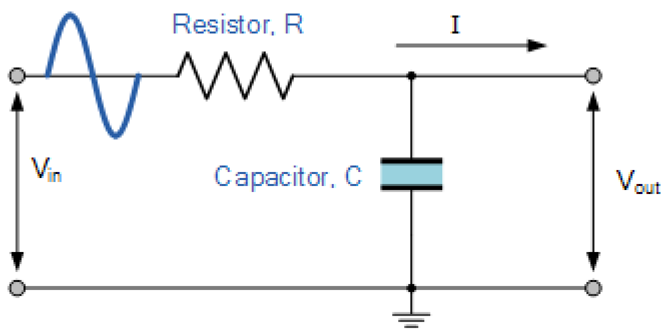


Fig 1 low pass filter circuit.

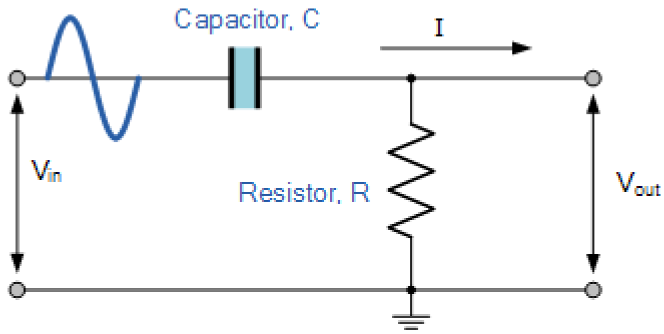
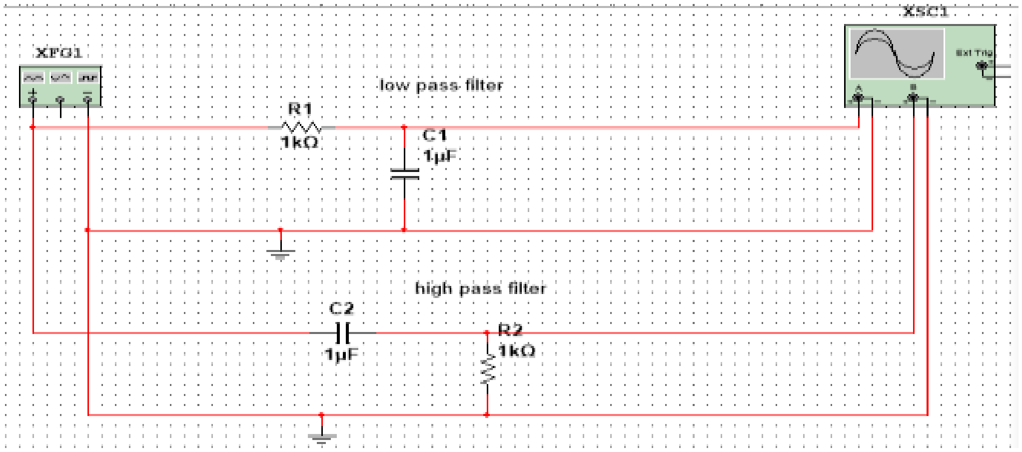


Fig 2 high pass filter circuit

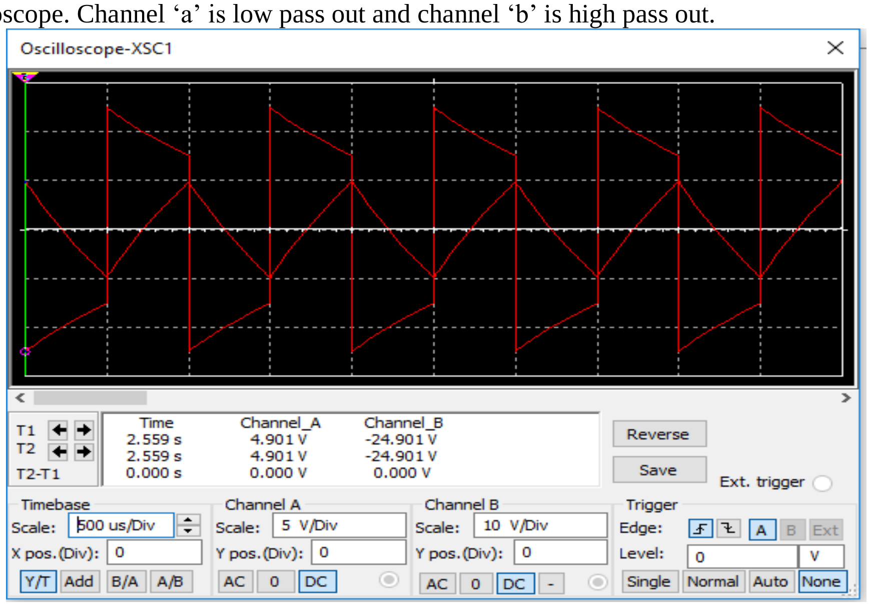
Multisim Steps:

1. Low pass circuit and high pass circuits in Multisim

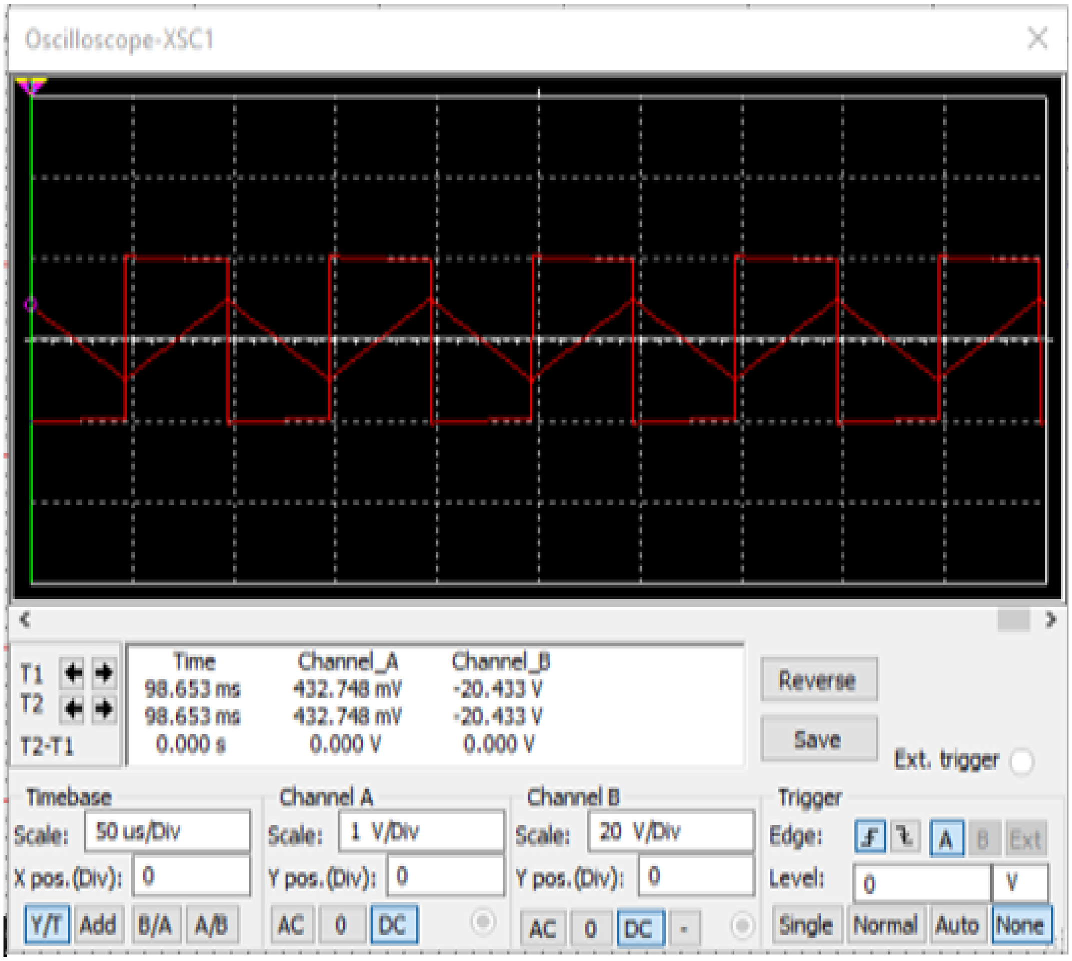


1. The time constants of both the circuits are equal to 1ms i.e. T=R*C=10^ (3)*10^ (-6) =0.001s.So adjust the frequency of square wave input according to the time constant. We shouldmonitor three steps accordingly and those are RC=T, RC<T and RC>T.

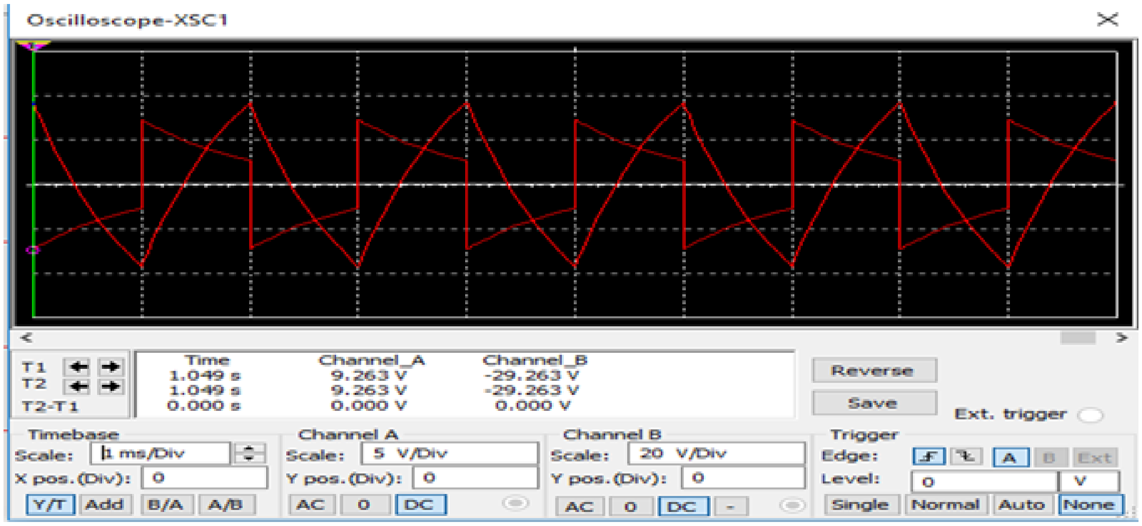
a) In RC=T case the frequency is given as 1 kHz, both the wave form are taken channel 'a' andchannel 'b' of the oscilloscope. Channel 'a' is low pass out and channel 'b' is high pass out.



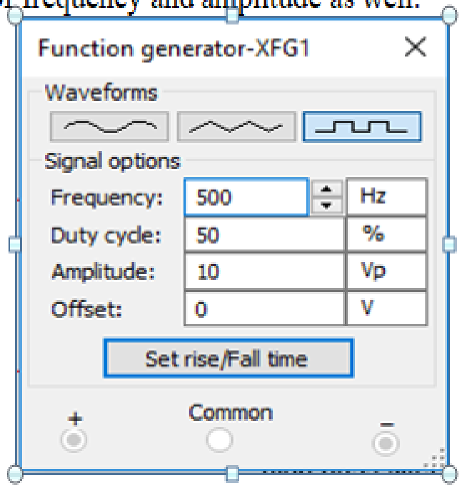
In RC>T case the frequency is given as 10 kHz, and time period will be 0.1 ms less than the RC value.



c) In RC<T the frequency is taken as 500Hz and then the time period will 2 ms greater than the RC value.



3. In order to adjust the frequency in the function generator double click on the function generator and then adjust the value of frequency and amplitude as well.

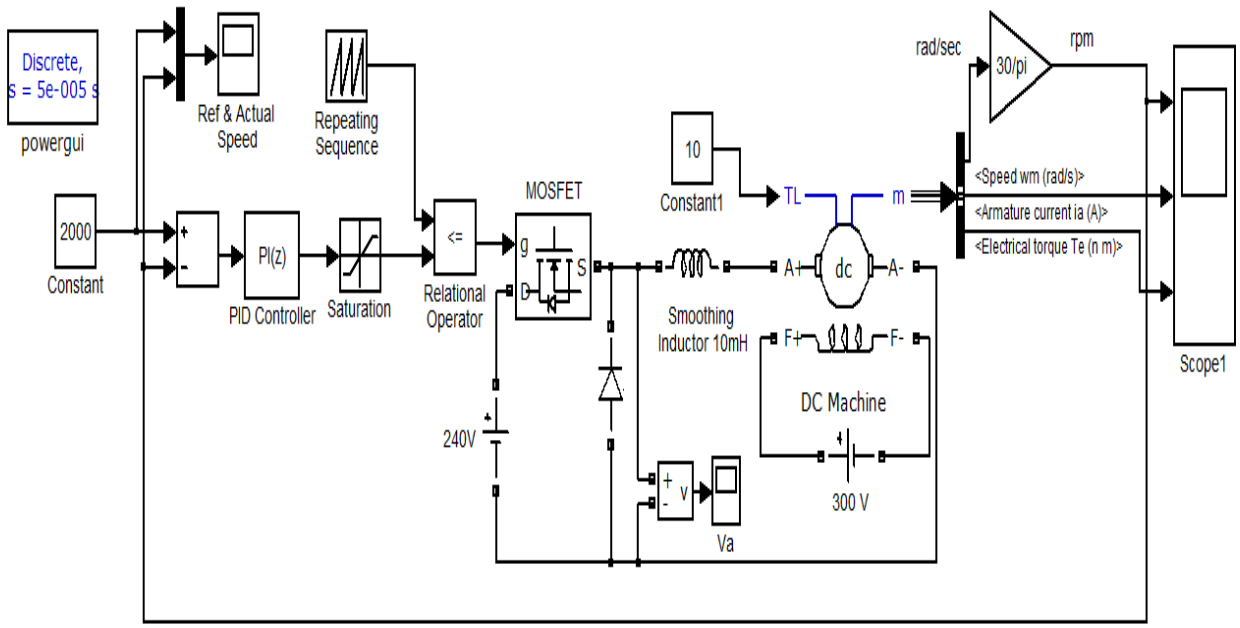


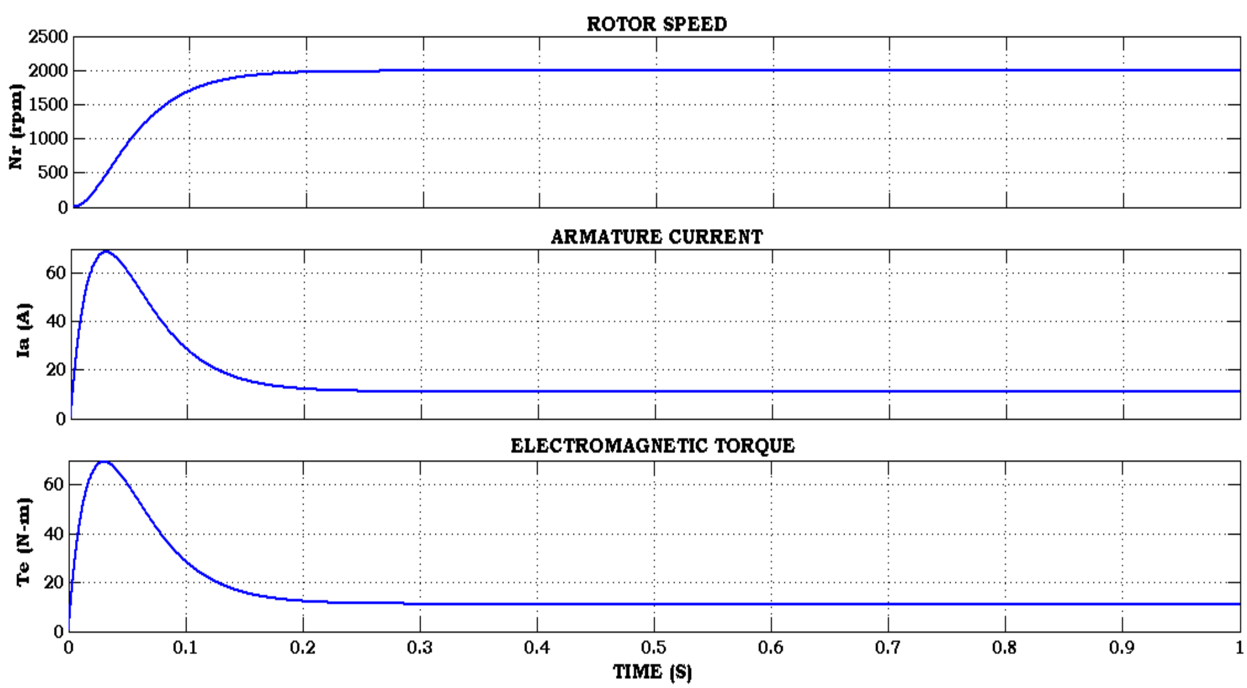
Conclusion: therefore the simulation of the high pass and low pass filters with three different conditions is simulated using Multisim.

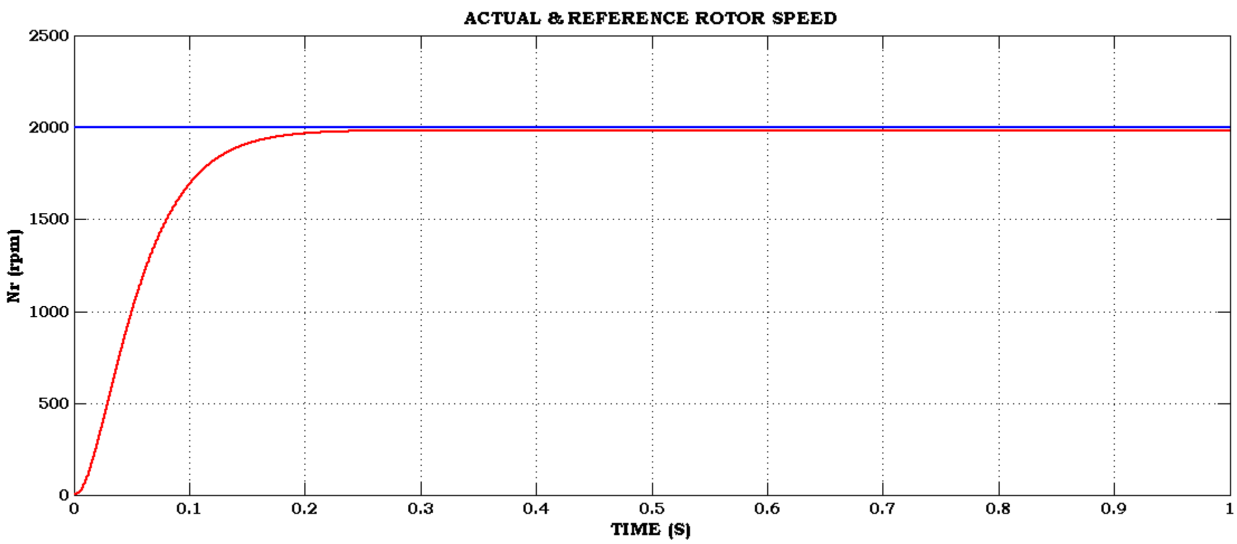
Results:

EXPERIMENT-9
PERFORMANCE ANALYSIS OF DC MOTOR DRIVE UNDER CONSTANT & VARIABLE SPEED CONDITIONS

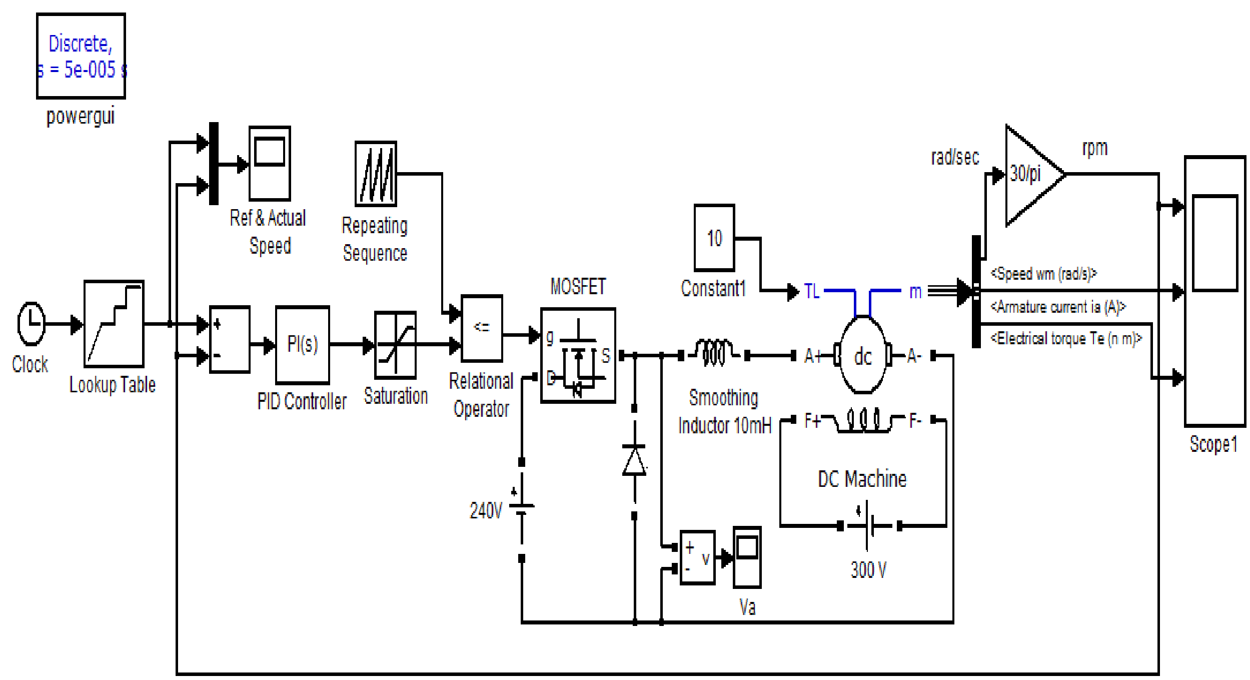
1. **Under Constant Speed Condition**

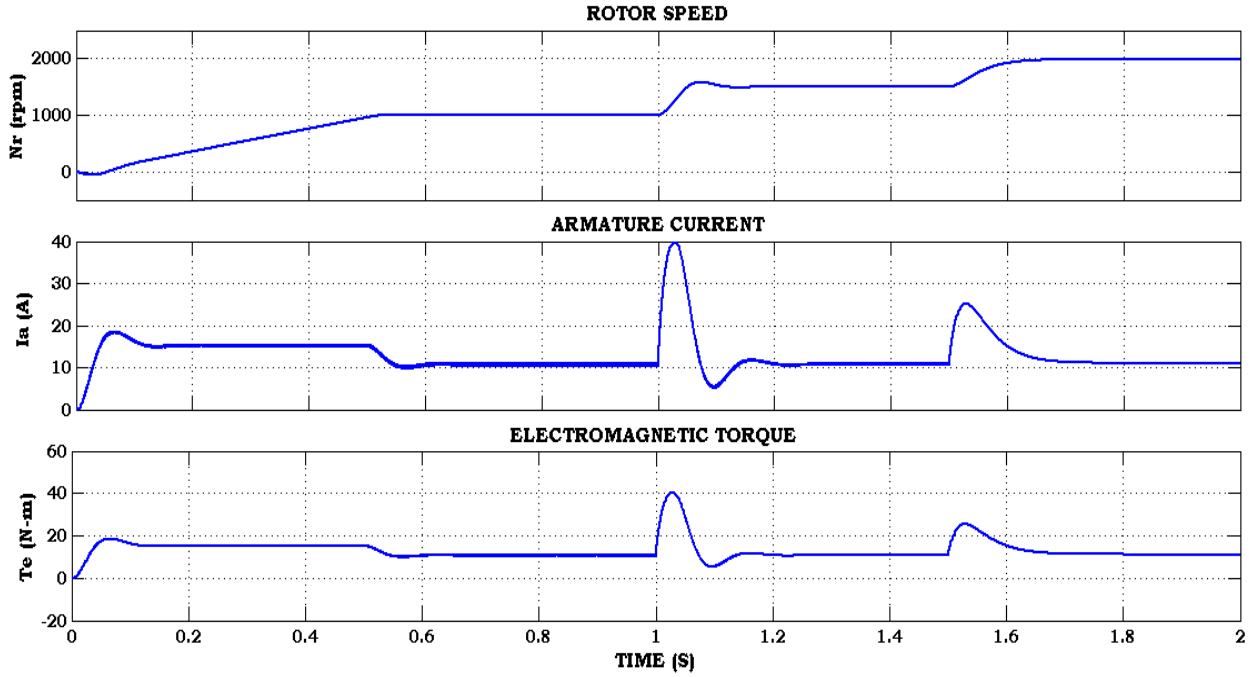


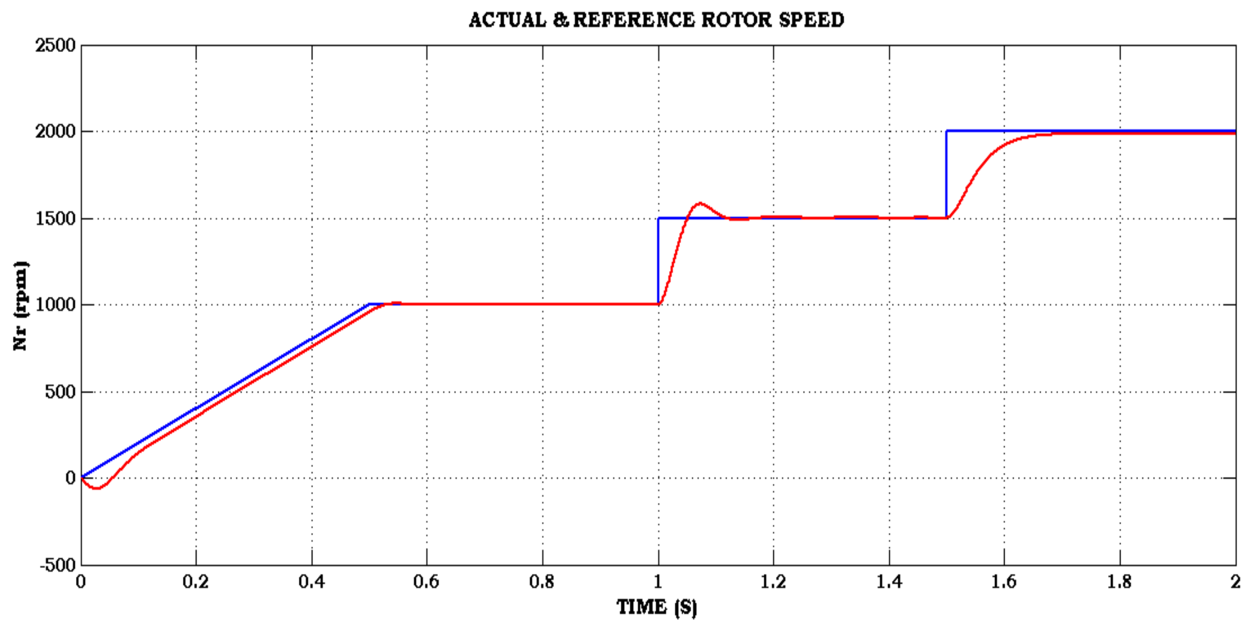




1. Under Variable Speed Condition




ROTOR SPEED
Nr (rpm)
2000
1000
0
ARMATURE CURRENT
Ia (A)
40
30
20
10
0
ELECTROMAGNETIC TORQUE
Te (N-m)
60
40
20
0
-20
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
TIME (S)

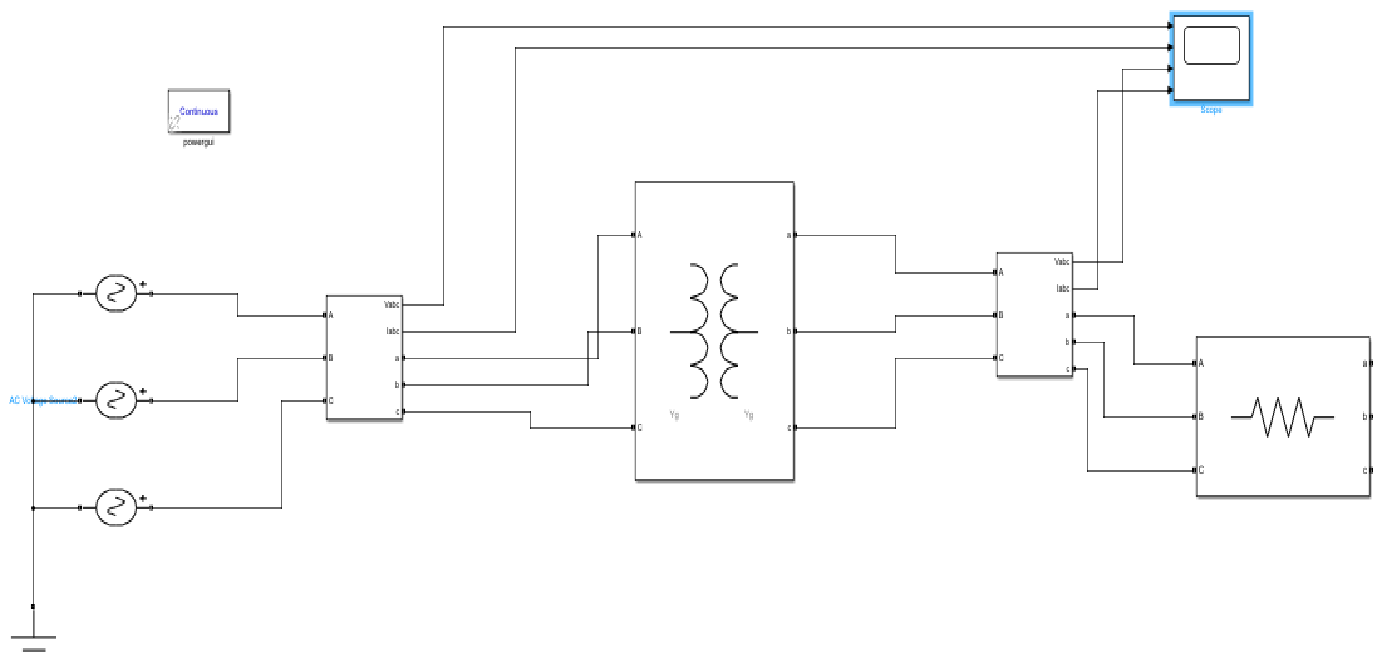

ACTUAL & REFERENCE ROTOR SPEED
Nr (rpm)
2500
2000
1500
1000
500
0
-500
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
TIME (S)

EXPERIMENT NO -10
MODELLING OF TRANSFORMER USING MATLAB

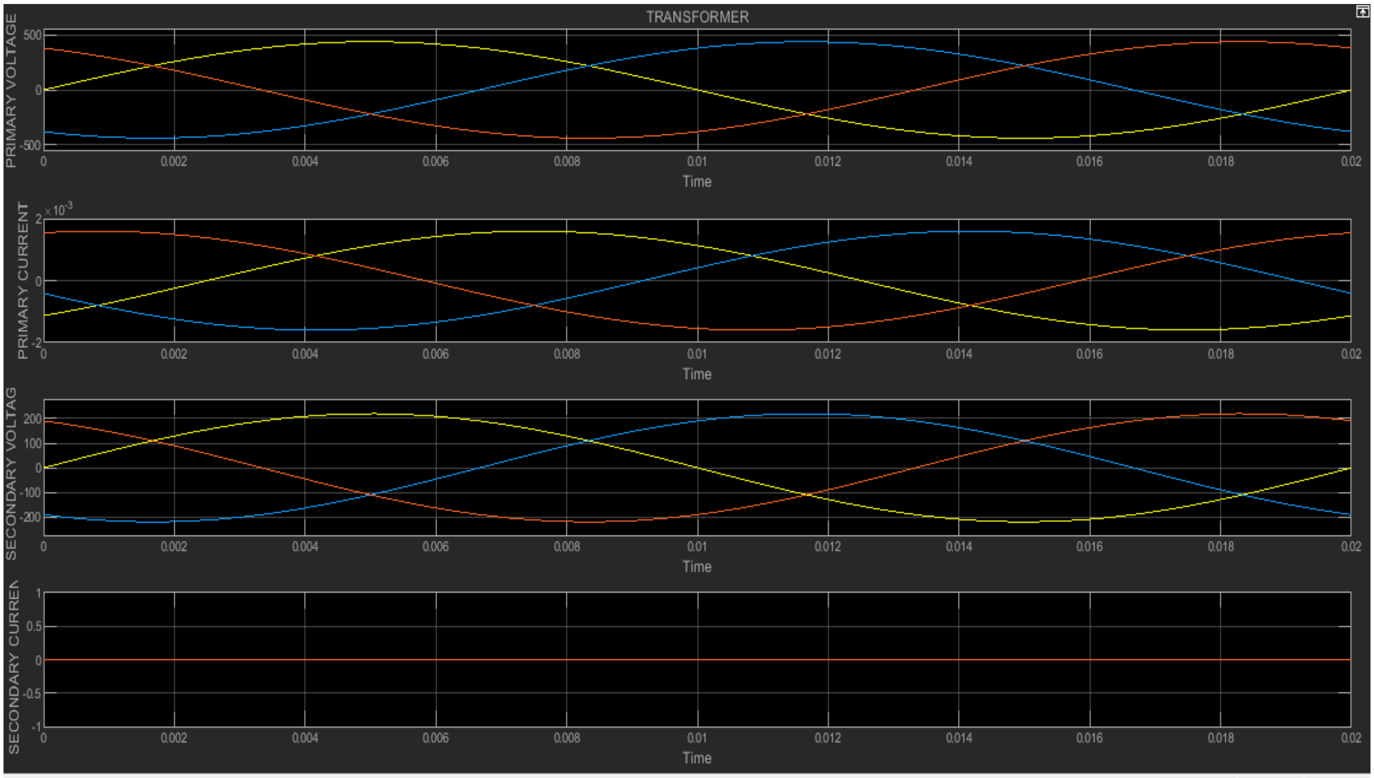
AIM: Analyze of Step-down transformer with MATLAB software.

Software Required: MATLAB software.

SIMULATION DIAGRAM:



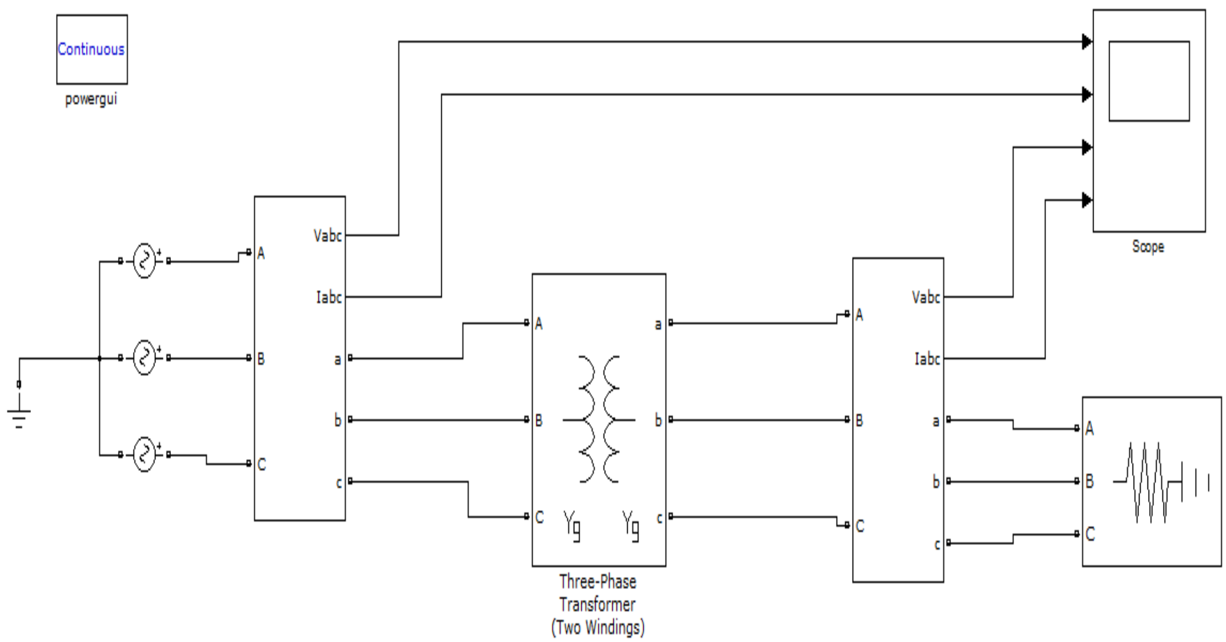
Simulation output:

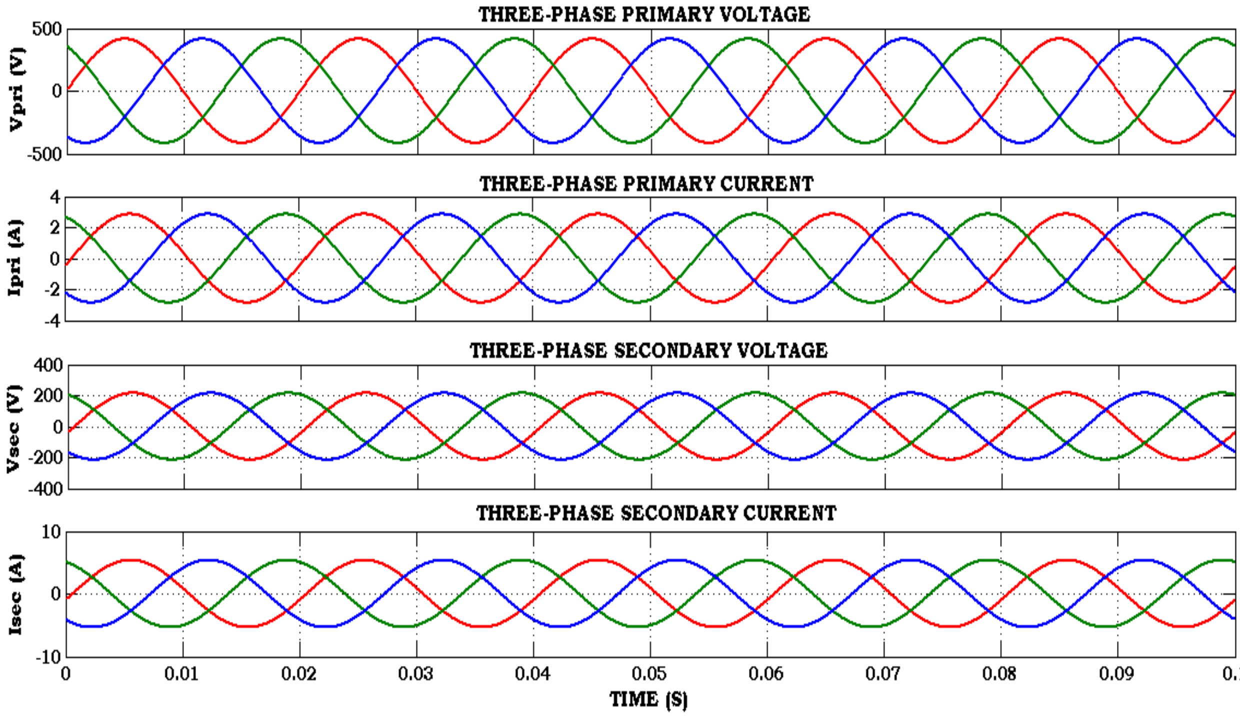


Results:

EXPERIMENT NO -10
MODELLING OF TRANSFORMER USING MATLAB

AIM: Analyze of Step-down transformer with MATLAB software.




THREE-PHASE PRIMARY VOLTAGE
Vpri (V)
500
0
-500
THREE-PHASE PRIMARY CURRENT
Ipri (A)
4
2
0
-2
-4
THREE-PHASE SECONDARY VOLTAGE
Vsec (V)
400
200
0
-200
-400
THREE-PHASE SECONDARY CURRENT
Isec (A)
10
0
-10
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1
TIME (S)

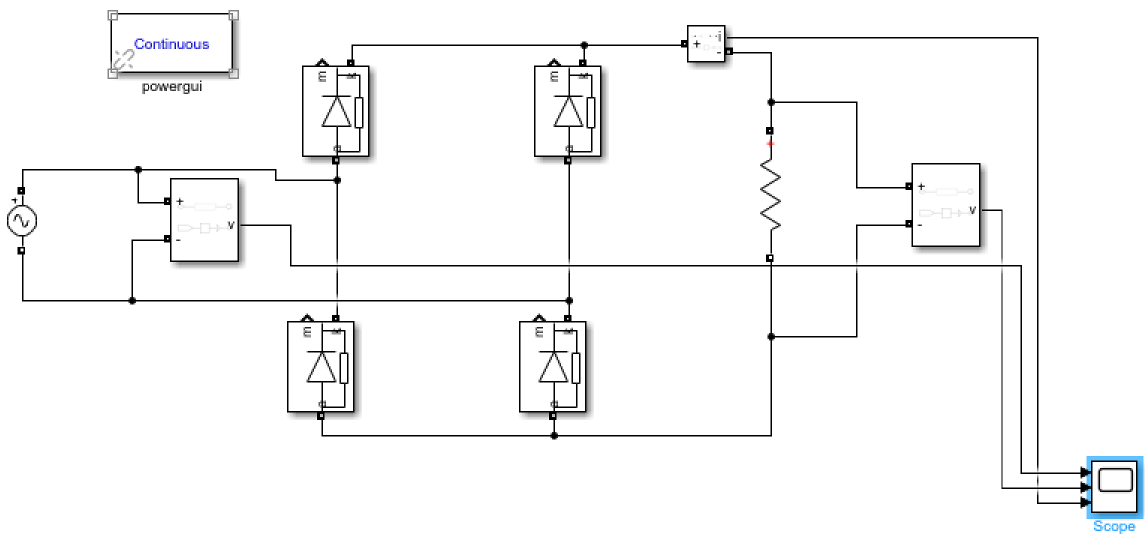
EXPERIMENT NO -11

Single Phase Bridge Rectifier with and without filter

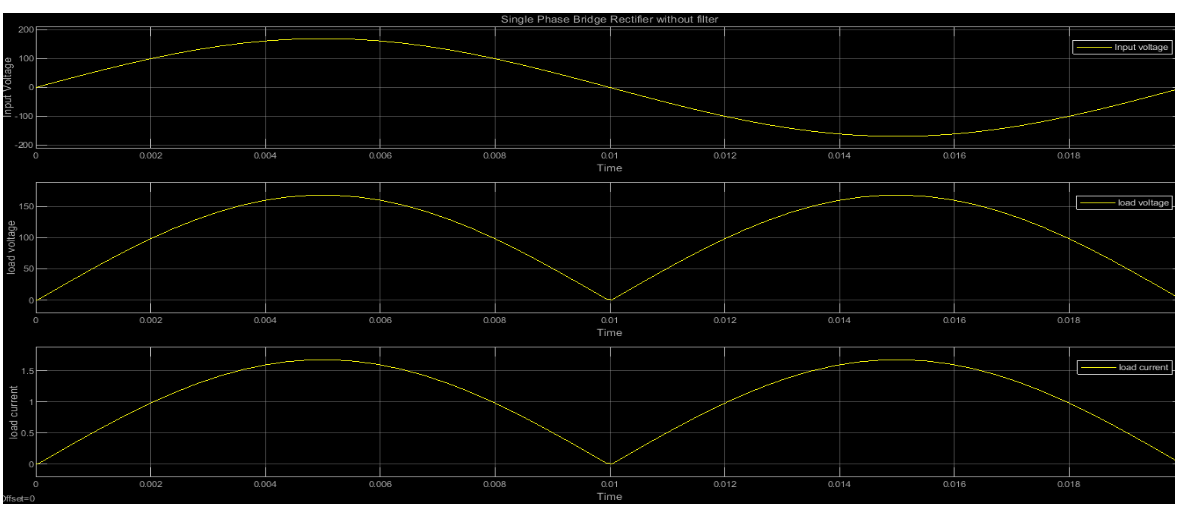
AIM: To analyze the rectifier output with and without C-type filter.

Software Required: MATLAB software.

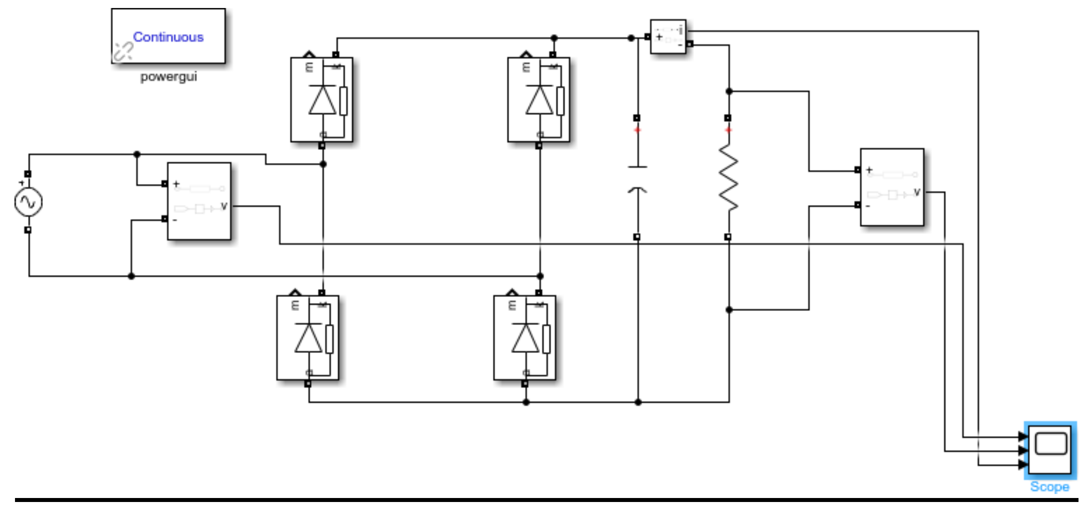
Simulation Diagram without Filter:



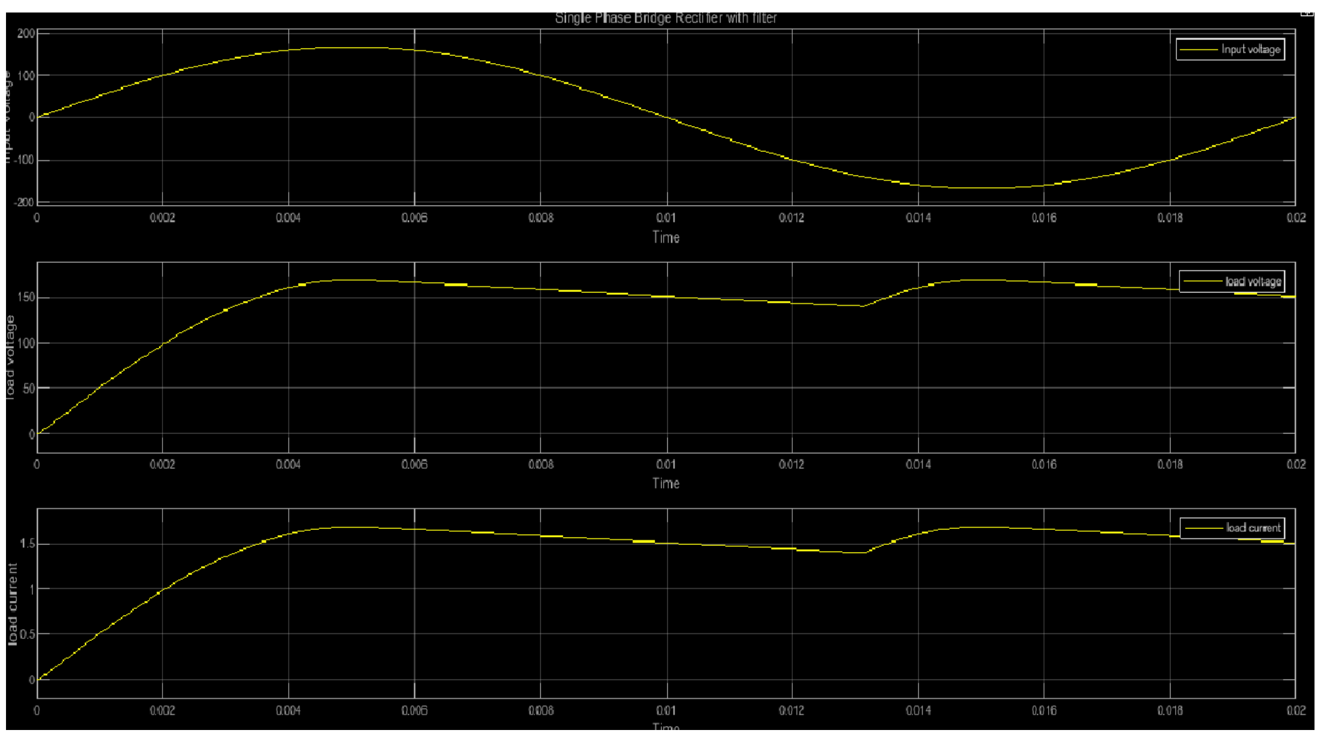
Simulation output without Filter:



Simulation Diagram with Filter:

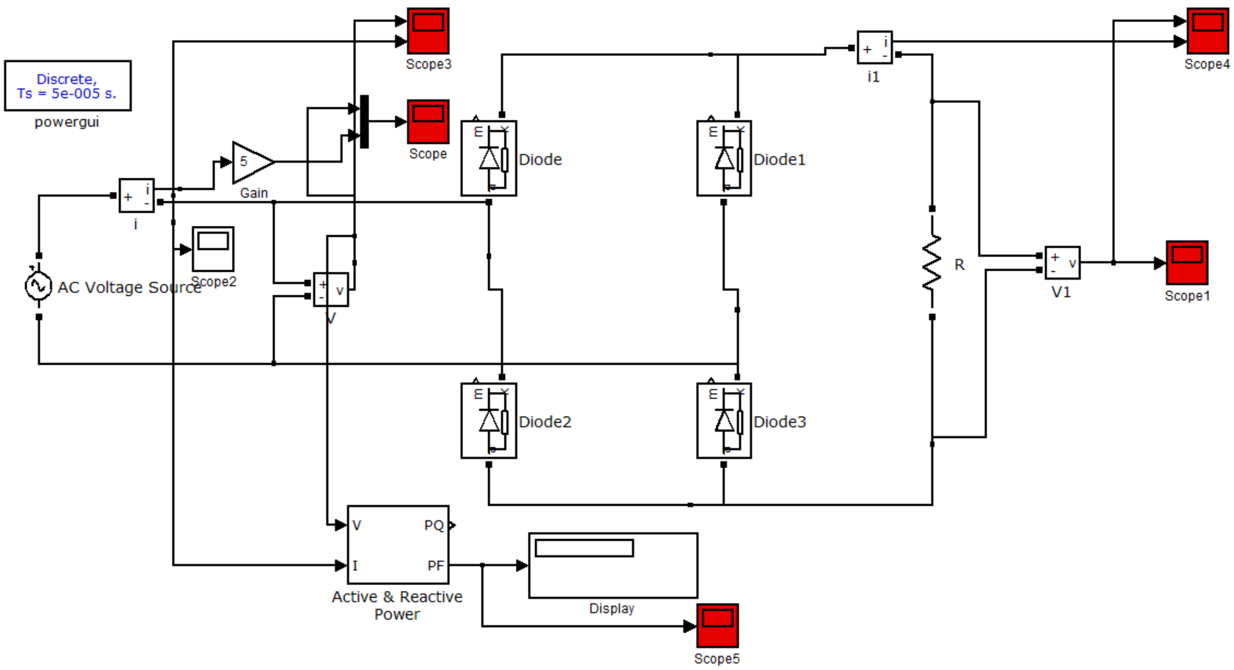


Simulation output with Filter:

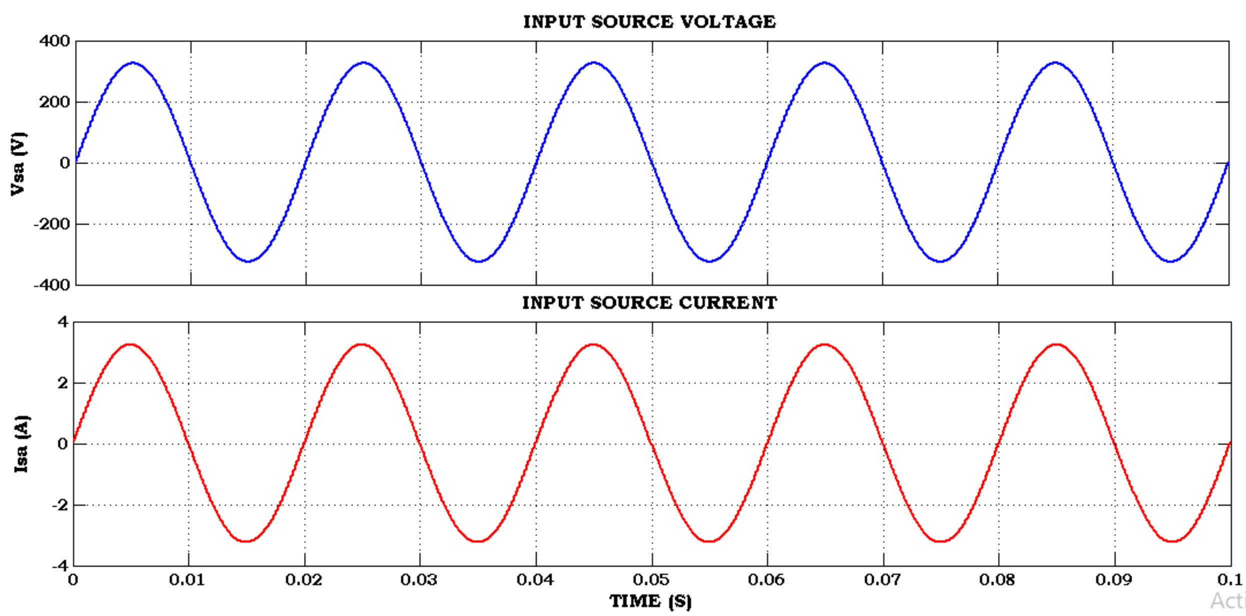


Result:

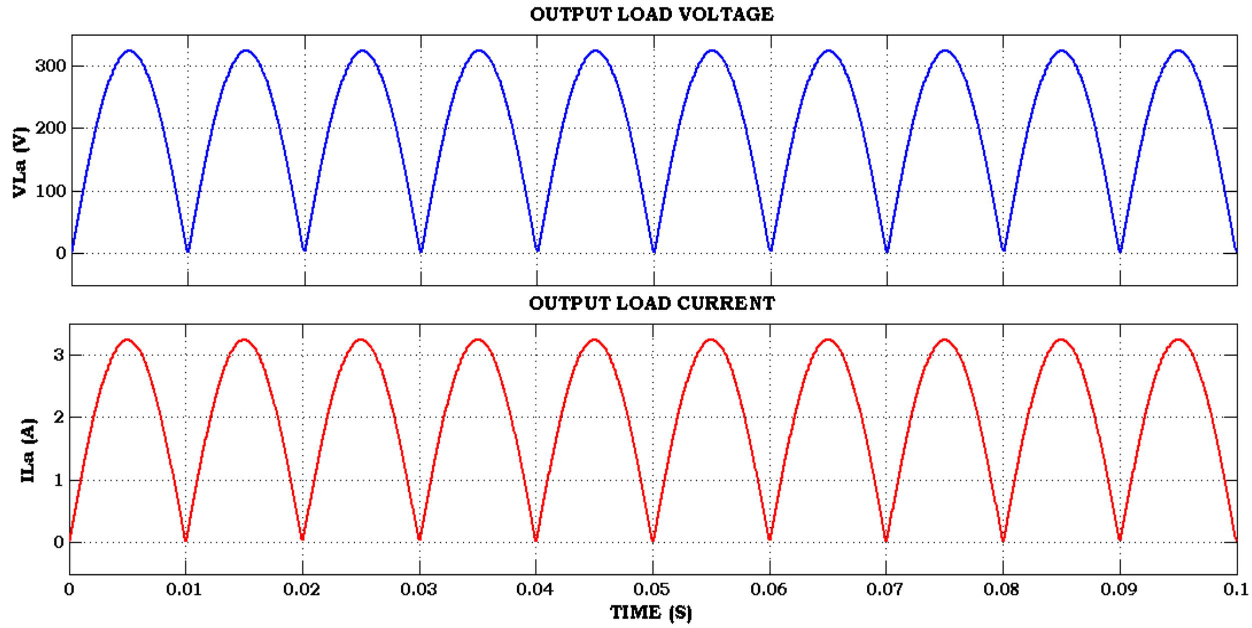
Experiment-11
Single Phase Bridge Rectifier with and without filter



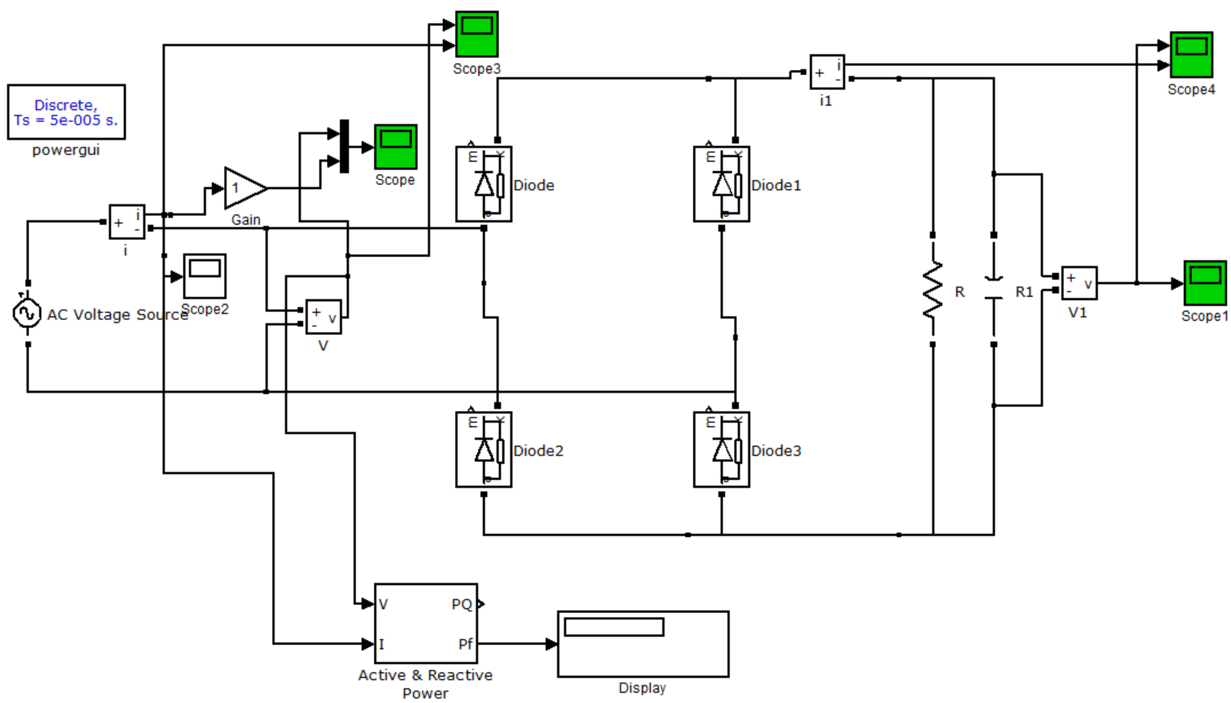
Matlab/Simulink Model of Single-Phase Rectifier without Filter



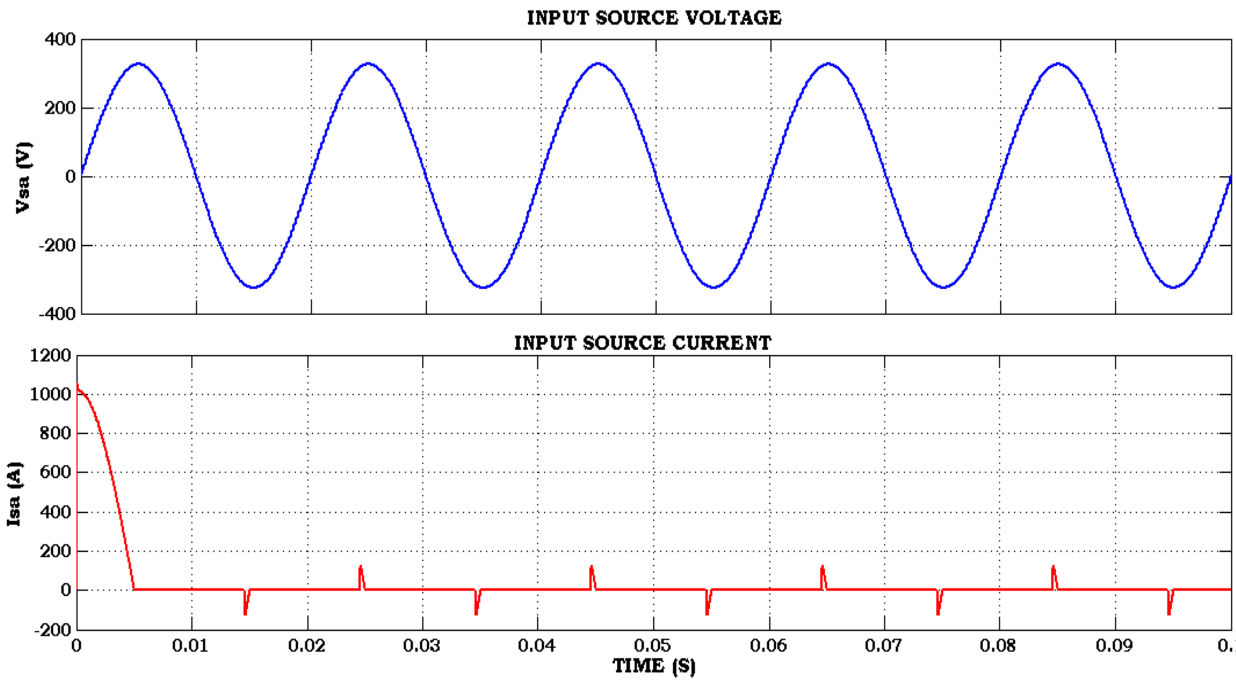
Input Source Voltage & Current



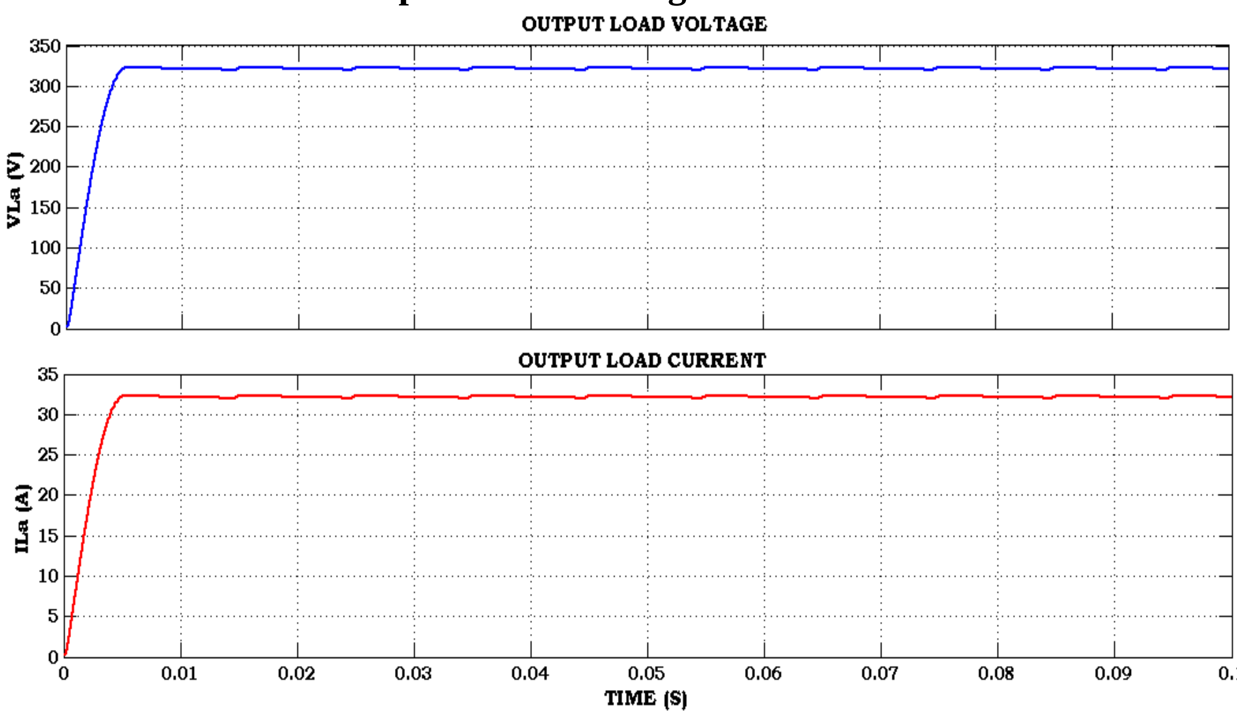
Output Load Voltage & Current



Matlab/Simulink Model of Single-Phase Rectifier with Filter



Input Source Voltage & Current



Output Load Voltage & Current

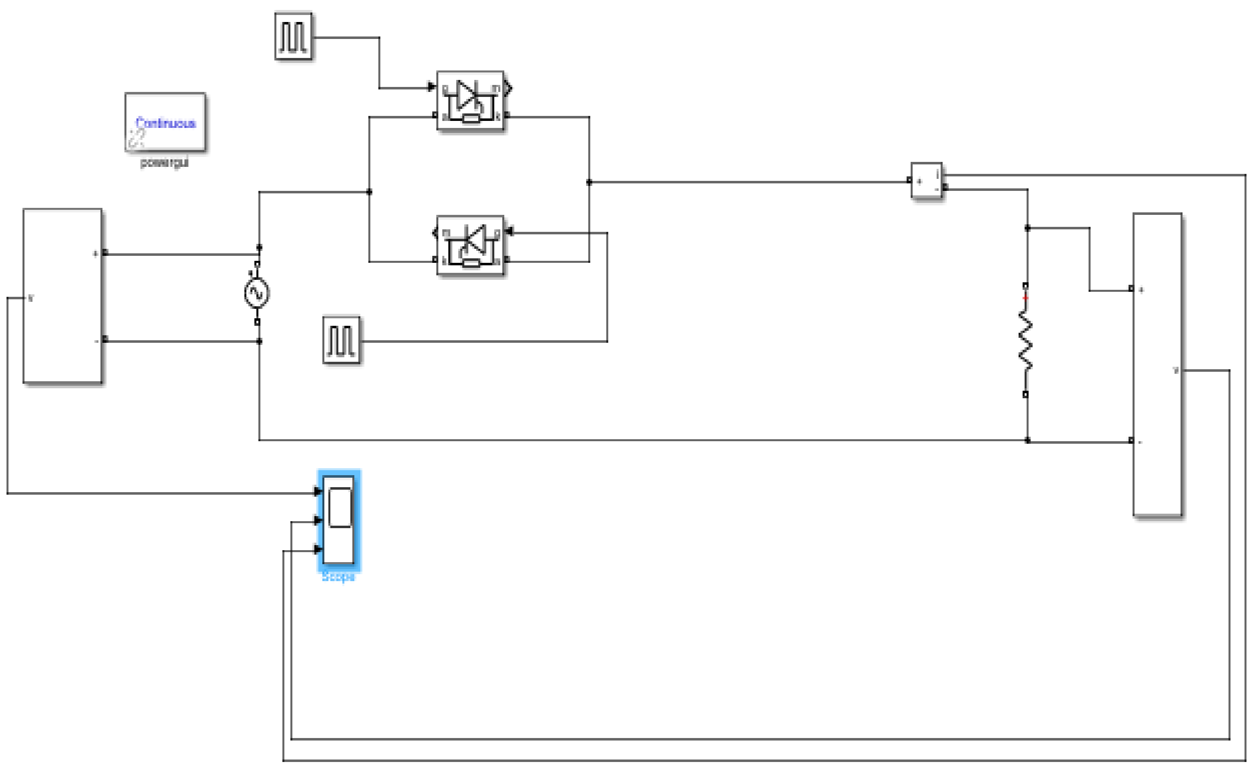
EXPERIMENT NO-12

VOLTAGE REGULATOR

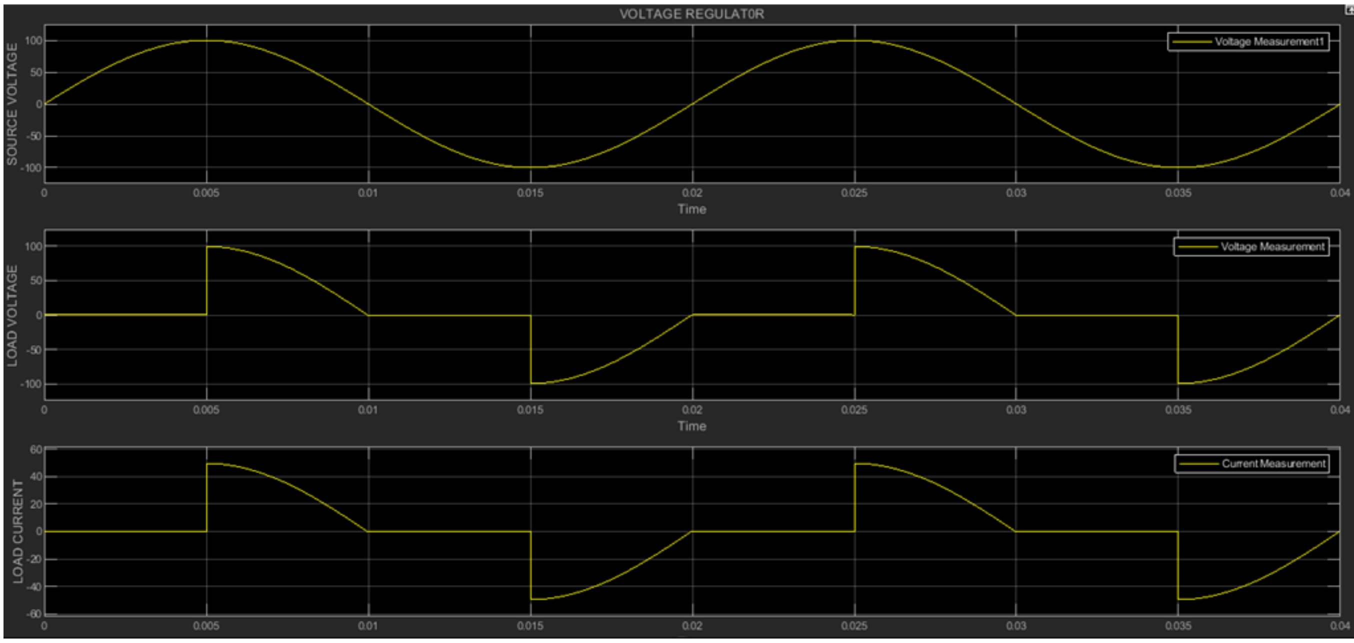
AIM: To analyze the voltage regulator output with R-LOAD.

Software Required: MATLAB software.

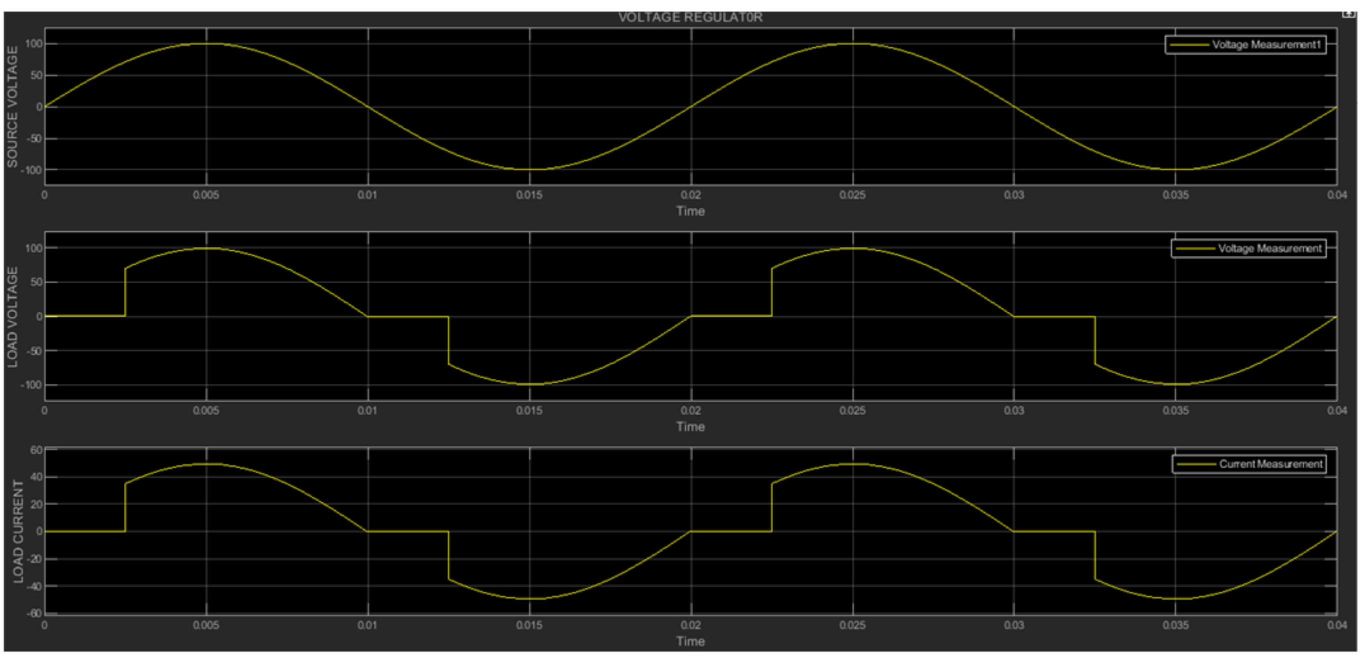
Simulation Diagram:



Simulation Output at α=90⁰:



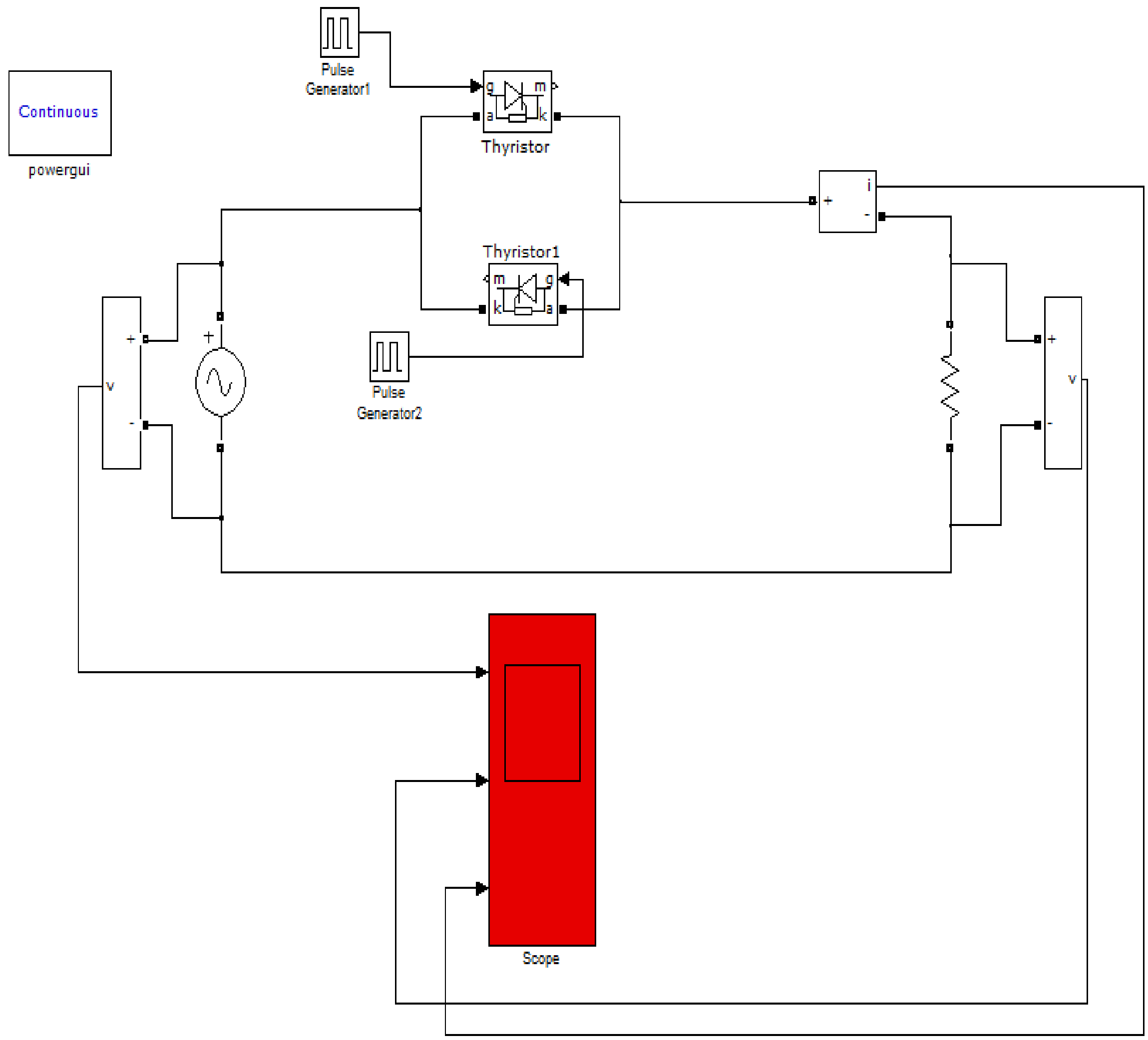
Simulation Output at α=45⁰:



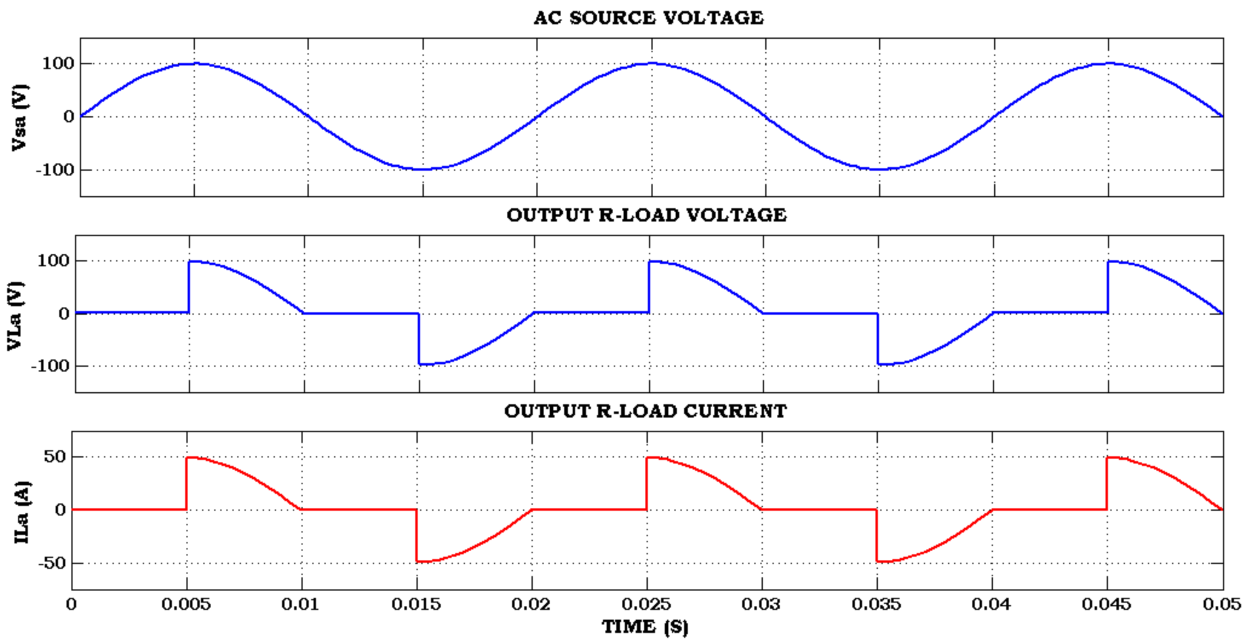
Results:

EXPERIMENT NO-12

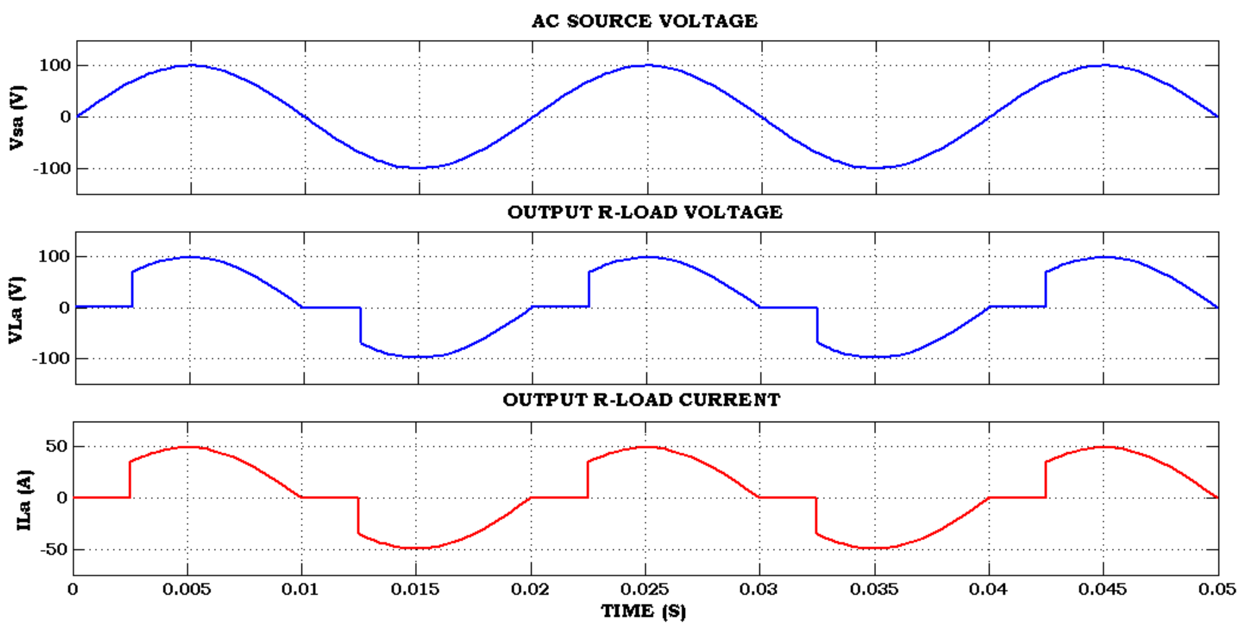
VOLTAGE REGULATOR



Simulation Output at α=90°:



Simulation Output at α=45°:

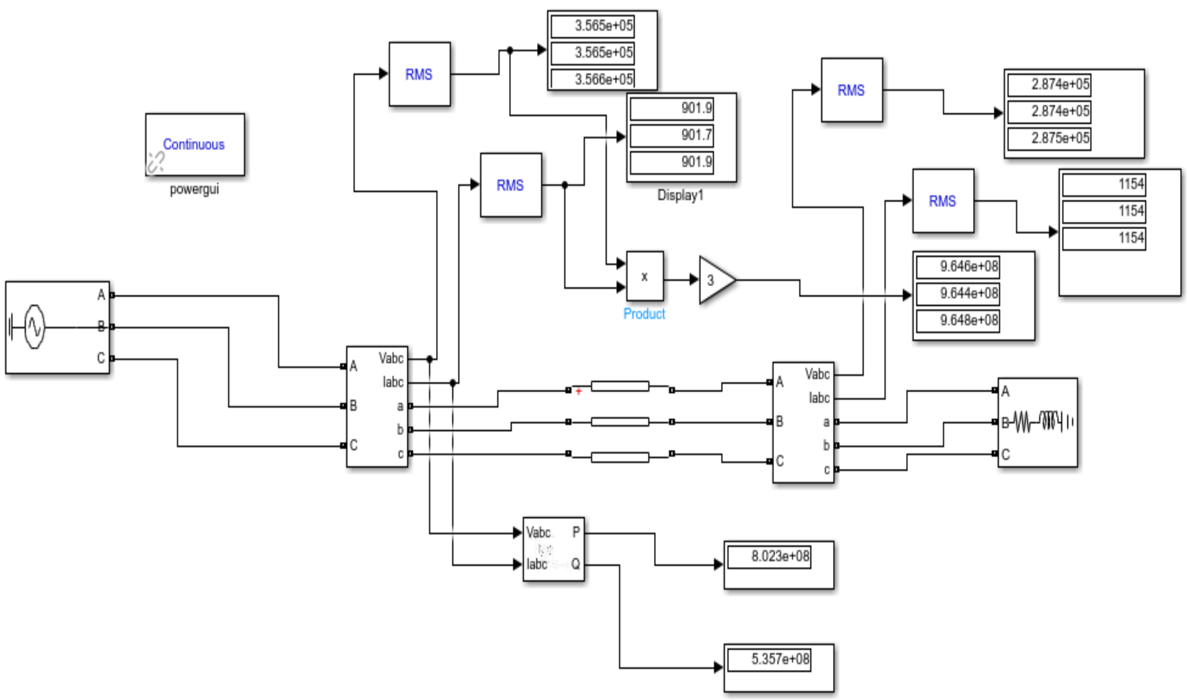


EXPERIMENT NO -13
MODELLING OF TRANSMISSION LINE USING MATLAB

AIM: A three phase, 60Hz, 500 KV transmission line is 300 km long. The line inductance is0.97mh/km per phase and its capacitance is 0.0115 μf/km per phase.Assume a lossless line.

Software Required: MATLAB software.

SIMULATION DIAGRAM:

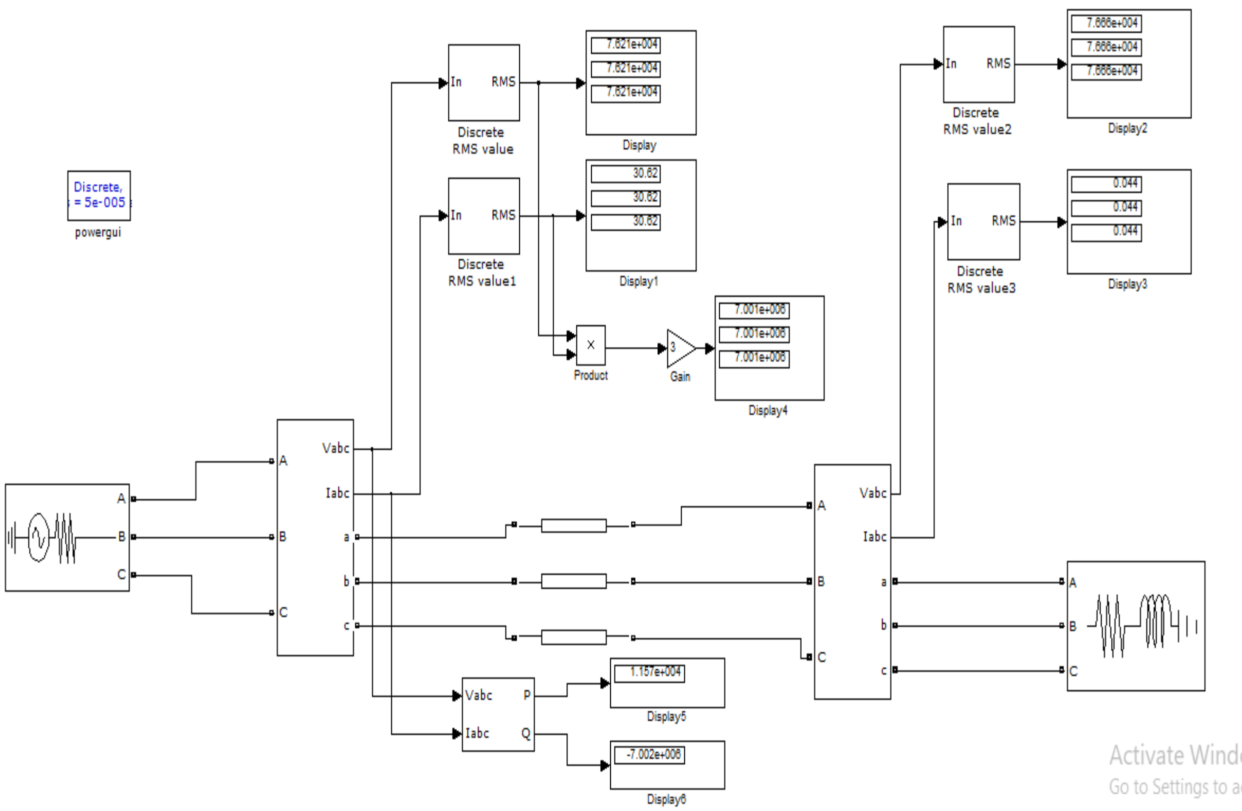


Results:

EXPERIMENT NO -13

MODELLING OF TRANSMISSION LINE USING MATLAB

AIM: To Analyze transmission line with MATLAB software.

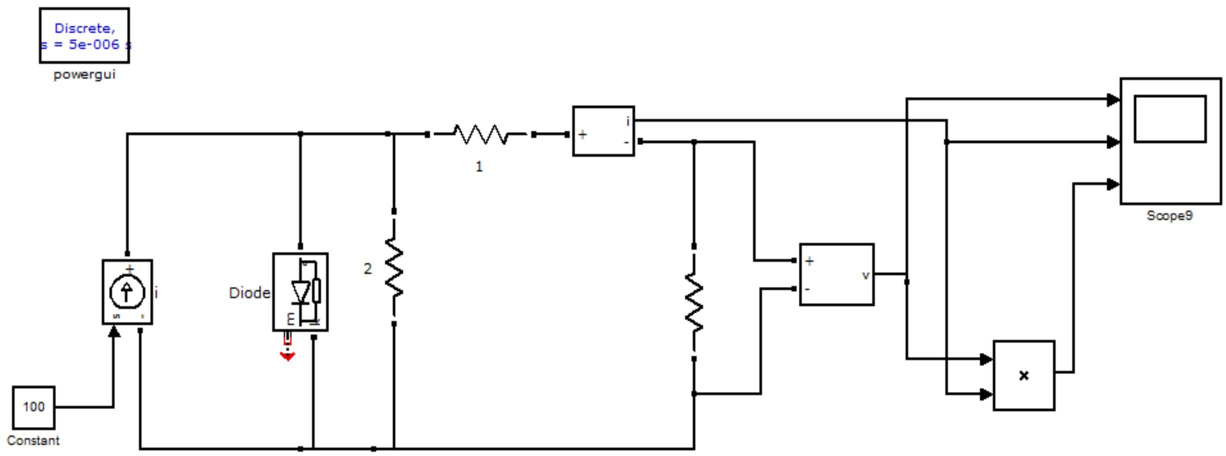


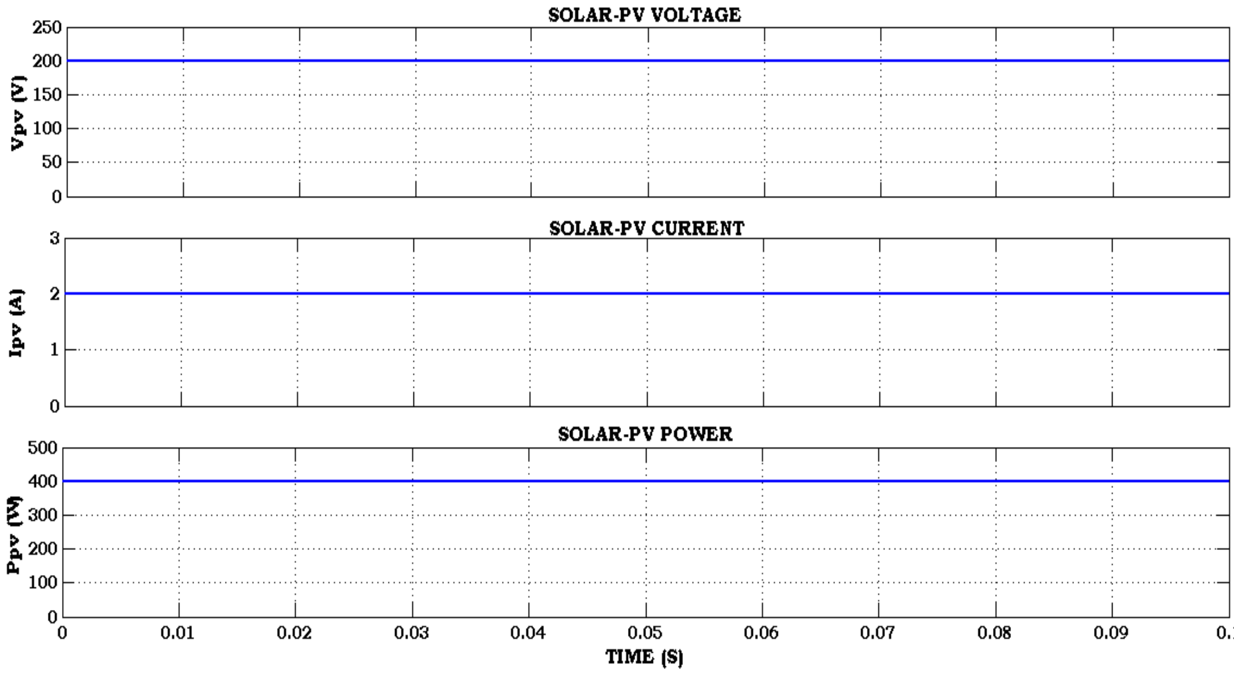
EXPERIMENT-14

PERFORMANCE OF SOLAR-PV MODEL

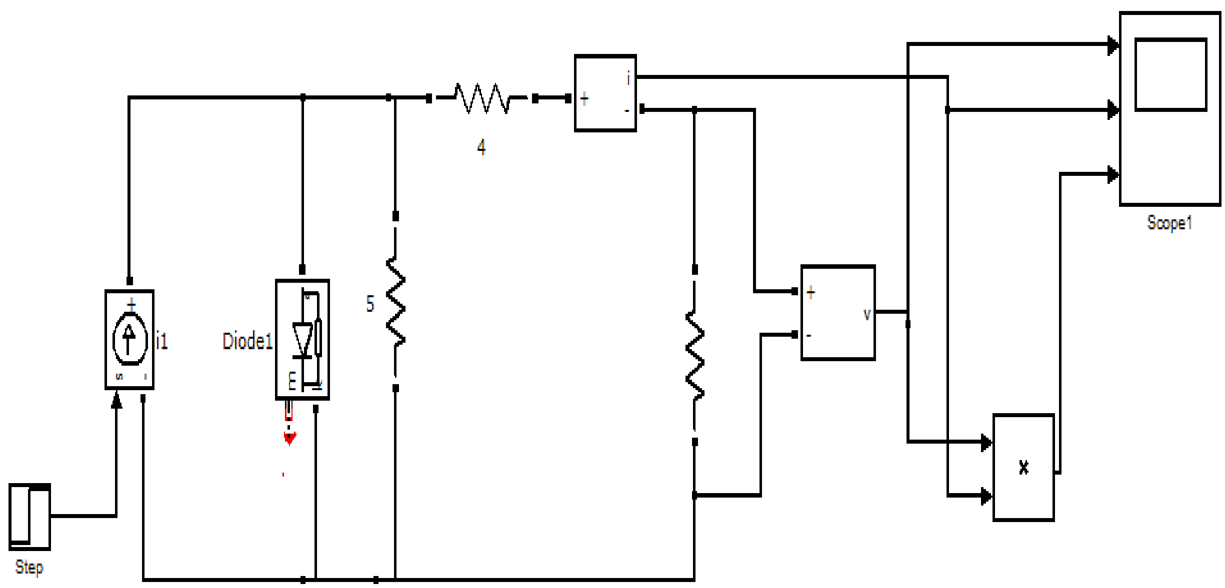
AIM: To Analyze SOLAR PV model with MATLAB software.

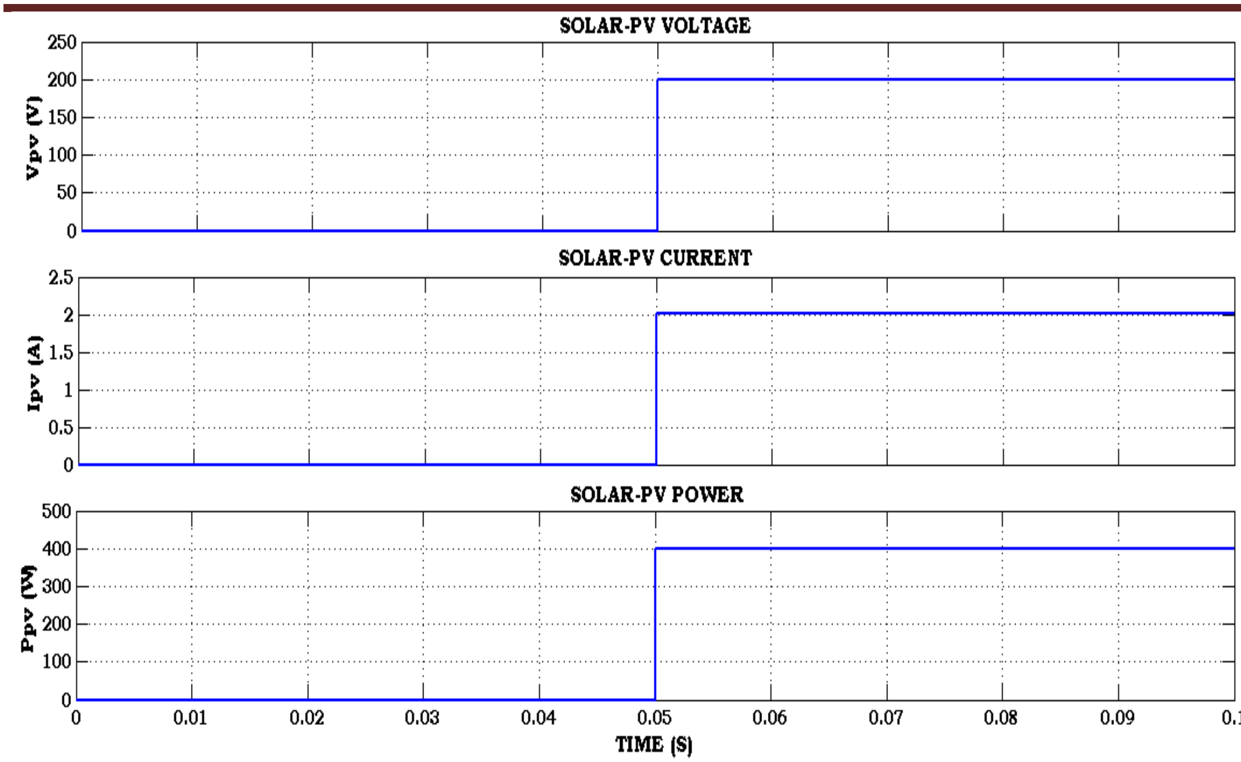
Under Constant Irradiance Level:





Under Variable Irradiance Level:




SOLAR-PV VOLTAGE
Vpv (V)
250
200
150
100
50
0
SOLAR-PV CURRENT
Ipv (A)
2.5
2
1.5
1
0.5
0
SOLAR-PV POWER
Ppv (W)
500
400
300
200
100
0
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1
TIME (S)

Results: