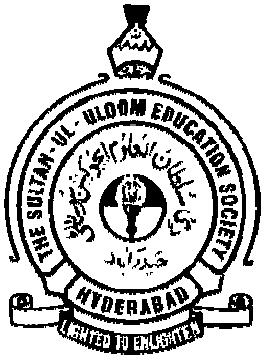
INSTRUMENTATION SIMULATION LAB

(EE435)

LABORATORY MANUAL

DEPARTMENT OF ELECTRICAL ENGINEERING



MUFFAKHAM JAH COLLEGE OF ENGINEERRING AND TECHNOLOGY

(Affiliated to Osmania University)

Banjara Hills, Hyderabad

2016

Revised by Prof. Shaik Abdul Qadeer, Srujana R. U

LIST OF EXPERIMENTS IN INSTRUMENTATION SIMULATION LAB

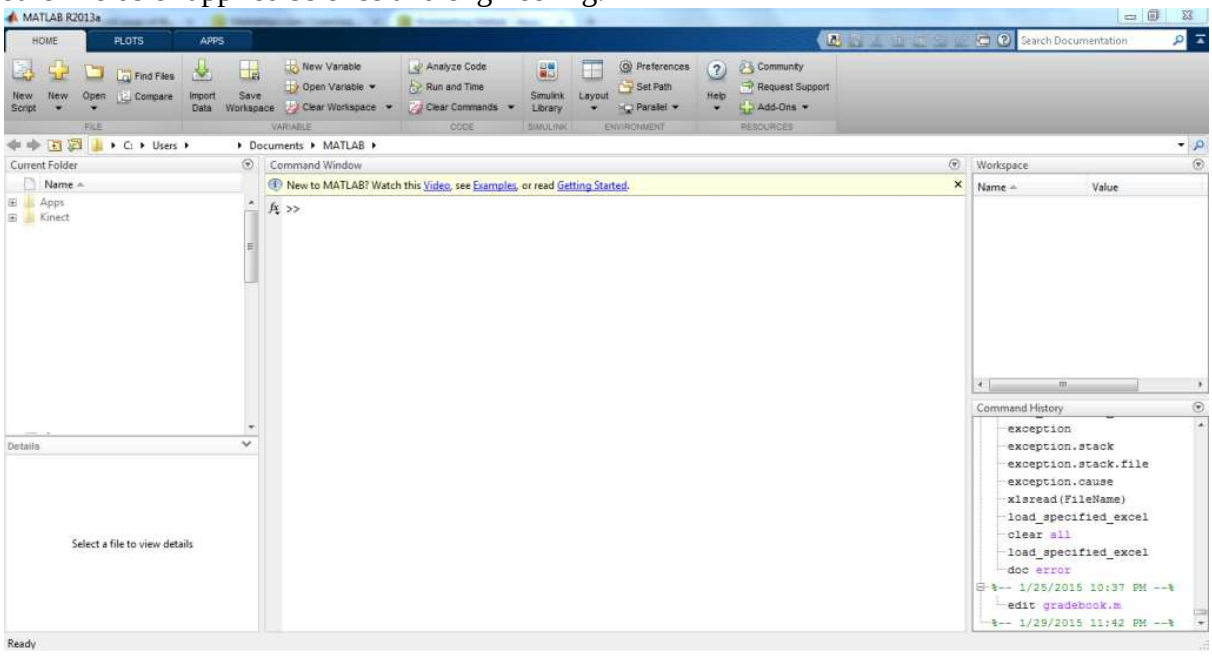
1. Verification of Network Theorems
   i) Superposition theorem.
   ii) Thevenin's theorem.
   iii) Maximum power transfer theorem.
2. Transient responses of series RLC, RL, RC circuits with Sine and Step inputs.
3. Series and Parallel resonance.
4. Bode plot, Root-locus plot and Nyquist plot.
5. Transfer function analysis of
   i) Time response of step input
   ii) Frequency response for sinusoidal input.
6. Design of Lag, Lead and Lag-Lead compensators.
7. Design and Simulation of Pressure Monitoring System using LabVIEW
8. Simulation of Tank Level Control System using LabVIEW
9. Analysis of an ECG Waveform using LabVIEW
10. Design of Temperature Monitoring System using LabVIEW

INTRODUCTION TO MATLAB

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

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

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



This is the default layout of MATLAB version used in our laboratory.

The main window is the Command Window. You can type in there any command that is available in MATLAB.

The second window in importance is the workspace. This is the current state of memory in MATLAB. The entire variables that are being used go there. The command history and the current folder are just useful tool that you can use but they are not essential to understand MATLAB.

Using MATLAB as a calculator:

As an example of a simple interactive calculation, just type the expression you want to evaluate. Let's start at the very beginning. For example, let's suppose you want to calculate the expression, $1 + 2 \times 3$.

You type it at the prompt command (>>) as follows,

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

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

For example,

>> x = 1+2*3

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

>> 4*x

ans = 28.0000

Basic arithmetic operators

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

Elementary functions

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

Predefined constant values

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

MATLAB by default displays only 4 decimals in the result of the calculations, for example −163.6667, as shown in above examples. However, MATLAB does numerical calculations in double precision, which is 15 digits. The command format controls how the results of computations are displayed. Here are some examples of the different formats together with the resulting outputs.

>> format short

>> x=-163.6667

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

>> format long

>> x= -1.636666666666667e+002

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

Managing the workspace:

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

>> clear

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

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

>> who

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

Here are few additional useful commands:

• To clear the Command Window, type clc

• To abort a MATLAB computation, type ctrl-c

• To continue a line, type . . .

HELP:

To view the online documentation, select MATLAB Help from Help menu or MATLAB Help directly in the Command Window. The preferred method is to use the Help Browser. The Help Browser can be started by selecting the ? icon from the desktop toolbar. On the other hand, information about any command is available by typing

>> help Command

INTRODUCTION TO LABVIEW

LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) is a platform and development environment for a virtual programming language from National Instruments. The Graphical language is named 'G'. Originally selected for the Apple Macintosh in 1986, LabVIEW is commonly used for data acquisition, instrument control, and Industrial automation on a variety of platforms including Microsoft Windows, various flavors of UNIX, LINUX and MACOSX.

The code files have extension .VI, which is an abbreviation for Virtual Instrument.

LabVIEW offers lots of additional Add-ons and Tool kits. It is a data flow programming language. It ties the creation of user interfaces (called front panel) into the development panel cycle. Each VI has three components, a block diagram, a front panel and a connector panel.

One benefit of LabVIEW over other development environments is the extensive support for accessing instrumentation hardware. Drivers and abstraction layers for many different types of instruments and buses are included or an available for inclusion. These present themselves as graphical nodes.

The LabVIEW introduces the following theme

Start using LabVIEW

1. The lab view environment
2. Front panel and block diagram
3. Palettes
4. Data types
5. Property nodes
6. Sub VIs
7. Loops and structures
8. Troubleshooting and debugging
9. Arrays... etc

These programs are called virtual instruments, because of their appearance and operation initiate physical instruments such as oscilloscopes and multiplexers.

EXPERIMENT NO: 1

VERIFICATION OF NETWORK THEOREMS

I) SUPERPOSITION THEOREM.

II) THEVENIN'S THEOREM.

III) MAXIMUM POWER TRANSFER THEOREM.

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

SOFTWARE USED : MULTISIM / MATLAB Simulink

SUPERPOSITION THEOREM:

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

Procedure:

Step 1:

1. Make the connections as shown in the circuit diagram by using MULTISIM/MATLAB Simulink.
2. Measure the response 'I' in the load resistor by considering all the sources 10V, 15V and 8V in the network.

Step 2:

1. Replace the sources 15V and 8V with their internal impedances (short circuited).
2. Measure the response 'I1' in the load resistor by considering 10V source in the network.

Step 3:

1. Replace the sources 10V and 8V with their internal impedances (short circuited).
2. Measure the response 'I2' in the load resistor by considering 15V source in the network.

Step 4:

1. Replace the sources 10V and 15V with their internal impedances (short circuited).
2. Measure the response 'I3' in the load resistor by considering 8V source in the network.

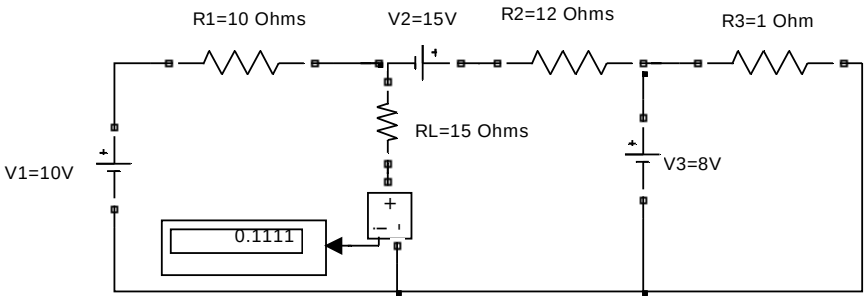
The responses obtained in step 1 should be equal to the sum of the responses obtained in step 2, 3 and 4. $I=I1+I2+I3$

Hence Superposition Theorem is verified.

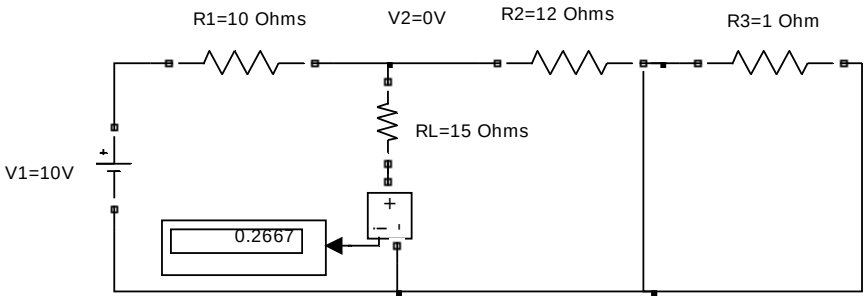
Continuous

powergui

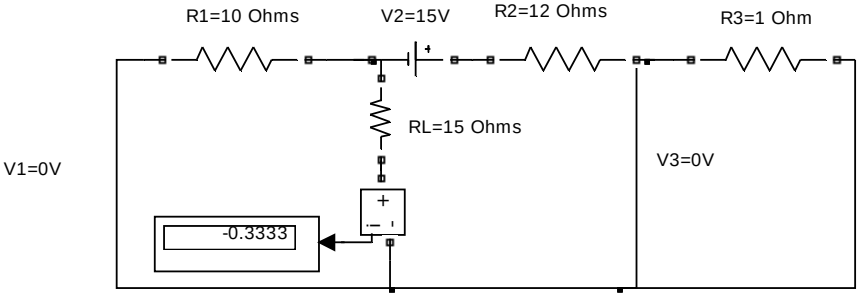
Step 1 : By Considering All Sources In The Network



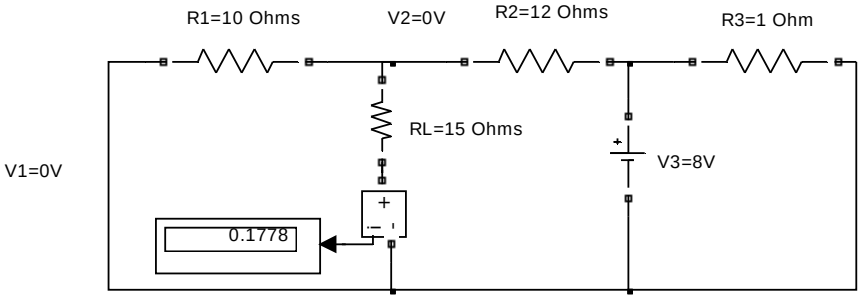
Step 2 : By Considering 10 V Sources In The Network



Step 3 : By Considering 15 V Sources In The Network



Step 4 : By Considering 8V Sources In The Network



Current through Load Resistor 15 Ohms :

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

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

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

Hence SuperPosition Theorem is Verified.

THEVENIN'S THEOREM:

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

Procedure:

Step 1:

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

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

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

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

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

Step 4: Thevenin's Equivalent Circuit

1. V_{TH} and R_{TH} are connected in series with the load.
2. Measure the current through the load resistor $I_L = \frac{V_{TH}}{R_{TH}+R_L}$.

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

$$I = I_L.$$

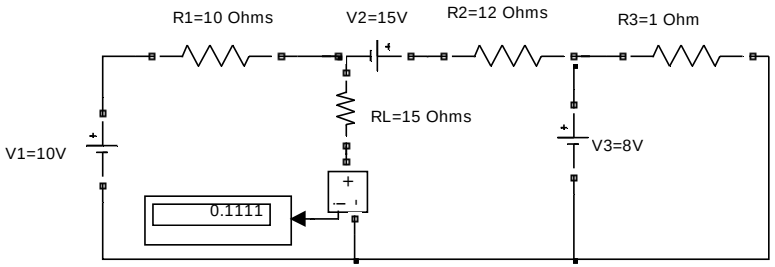
Hence Thevenin's Theorem is Verified.

Continuous

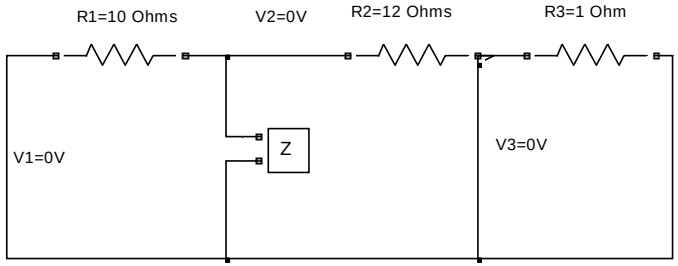
powergui

THEVENIN'S THEOREM

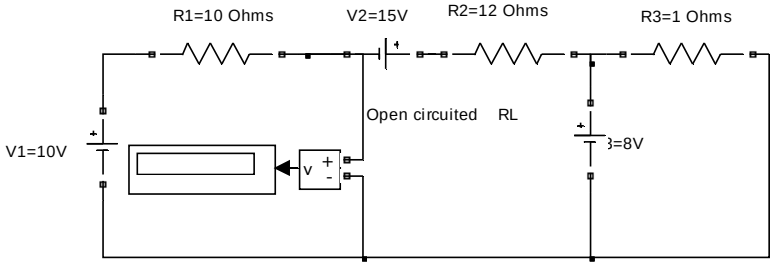
Step 1 : By Considering All Sources In The Network



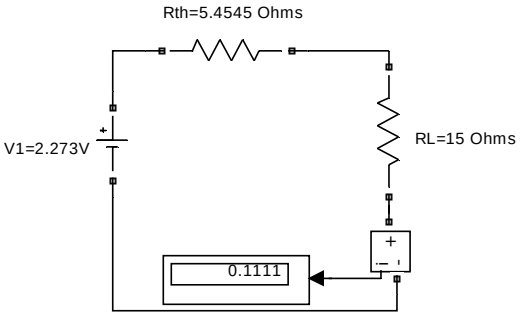
Step 2: Finding Thevenin's Resistance



Step 3 : Finding Thevenin's Voltage



Step 4 : Thevenin's Equivalent Network



Open Circuit Voltage Vth = 2.273V

Thevenin's Resistance = 5.4545Ohms

Current through Load Resistor 15 Ohms IL = 0.1111A

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

I=IL

Hence Thevenin's Theorem is Verified.

NORTON'S THEOREM:

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

Procedure:

Step 1:

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

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

Step 2: Finding Norton's Resistance(R_N)

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

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

Step 3: Finding Norton's Current(I_N)

1. Short the load terminals and measure the current through the short circuited terminals.

2. Measured current is be known as Norton's Current.

Step 4: Norton's Equivalent Circuit

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

2. Measure the current through the load resistor $I_L = \frac{I_N R_N}{R_N + R_L}$.

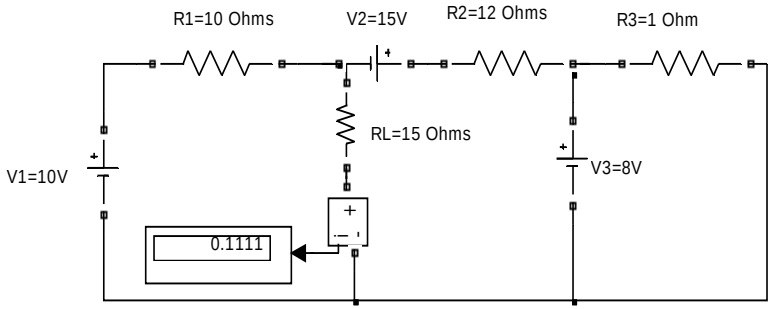
Current measured from Norton's Equivalent Circuit should be same as current obtained from the actual circuit.

$$I = I_L.$$

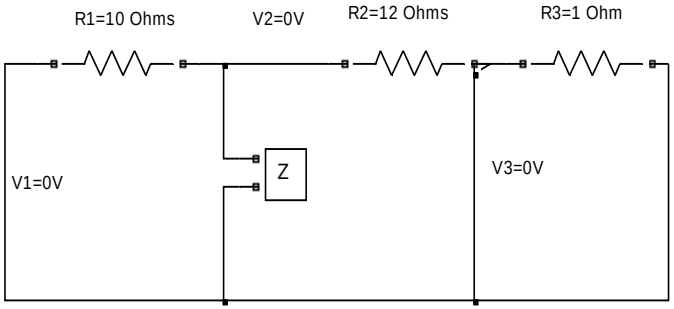
Hence Norton's Theorem is Verified.

NORTON'S THEOREM

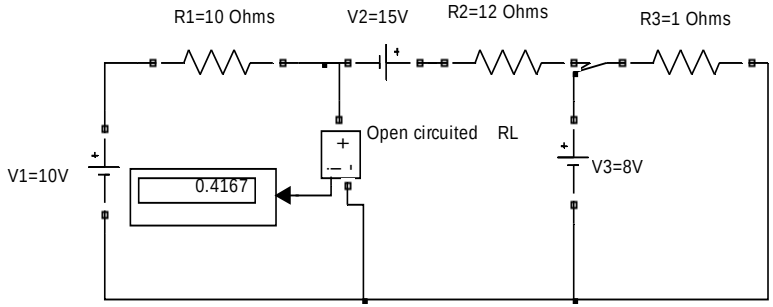
Step 1 : By Considering All Sources In The Network



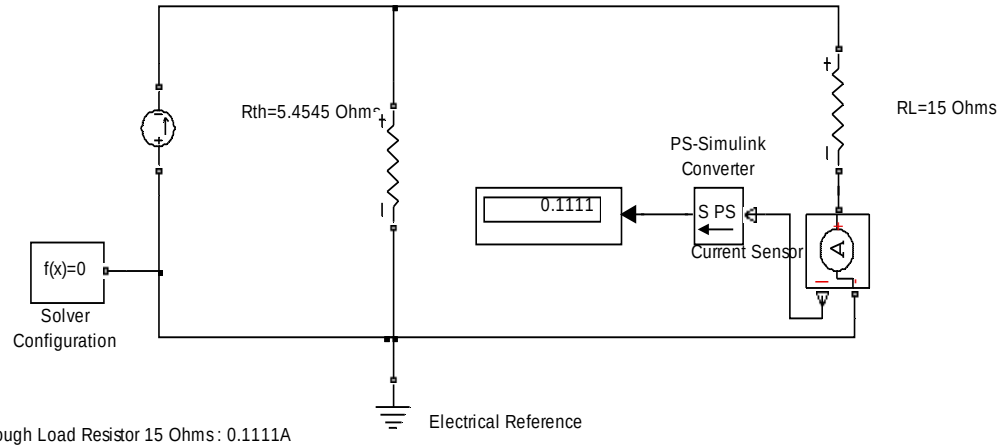
Step 2: Finding Norton's Resistance



Step 3 : Finding Norton's Current



Step 4 : Norton's Equivalent Network



Norton's Current = 0.4167 A
Norton's Resistance = 5.4545Ohms
Current through Load Resistor 15 Ohms = 0.1111A

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

Hence Norton's Theorem is Verified.

MAXIMUM POWER TRANSFER THEOREM:

"In any circuit the maximum power is transferred to the load when the load resistance is equal to the source resistance. The source resistance is equal to the Thevenin's equal resistance ".

Procedure:

Step 1:

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

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

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

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

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

Step 4: Measuring Power for different Load Resistors

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

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

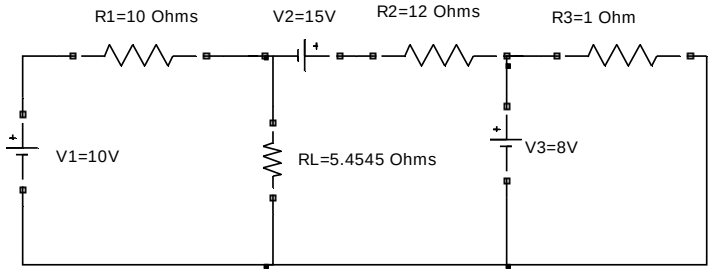
Hence Maximum Power Transfer Theorem is verified.

Continuous

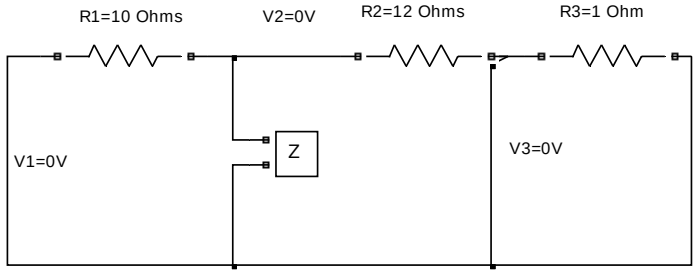
powergui

MAXIMUM POWER TRANSFER THEOREM

Step 1 : By Considering All Sources In The Network

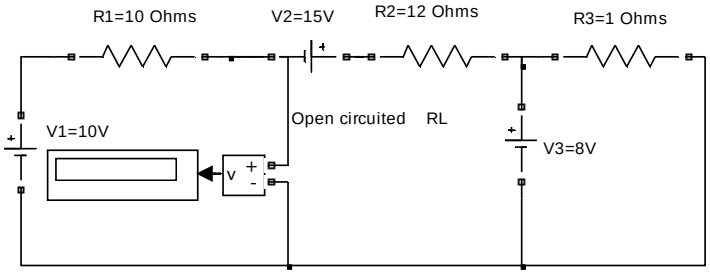


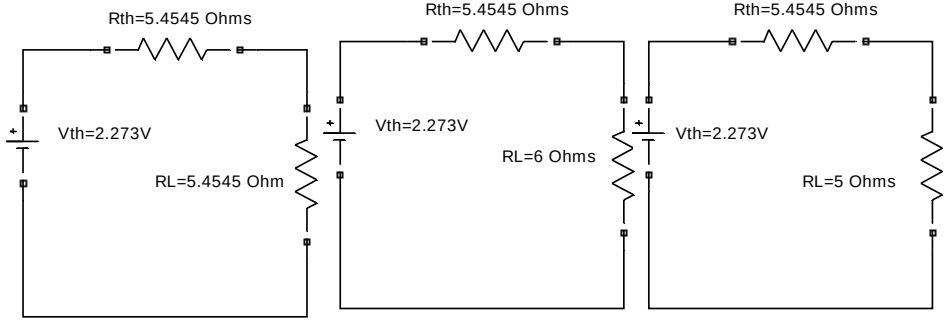
Step 2: Finding Thevenin's Resistance



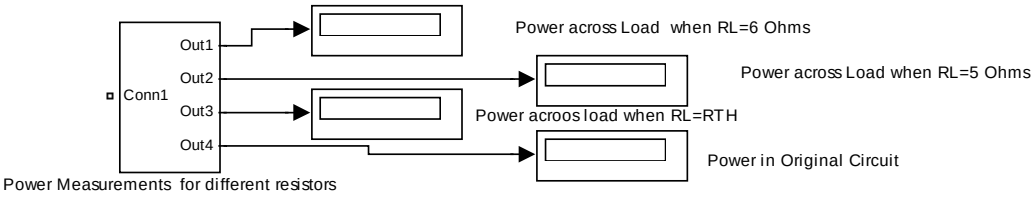
Step 4 : Power in Load Resistors with RL=RTH, RL>RTH, RL<RTH

Step 3 : Finding Thevenin's Voltage





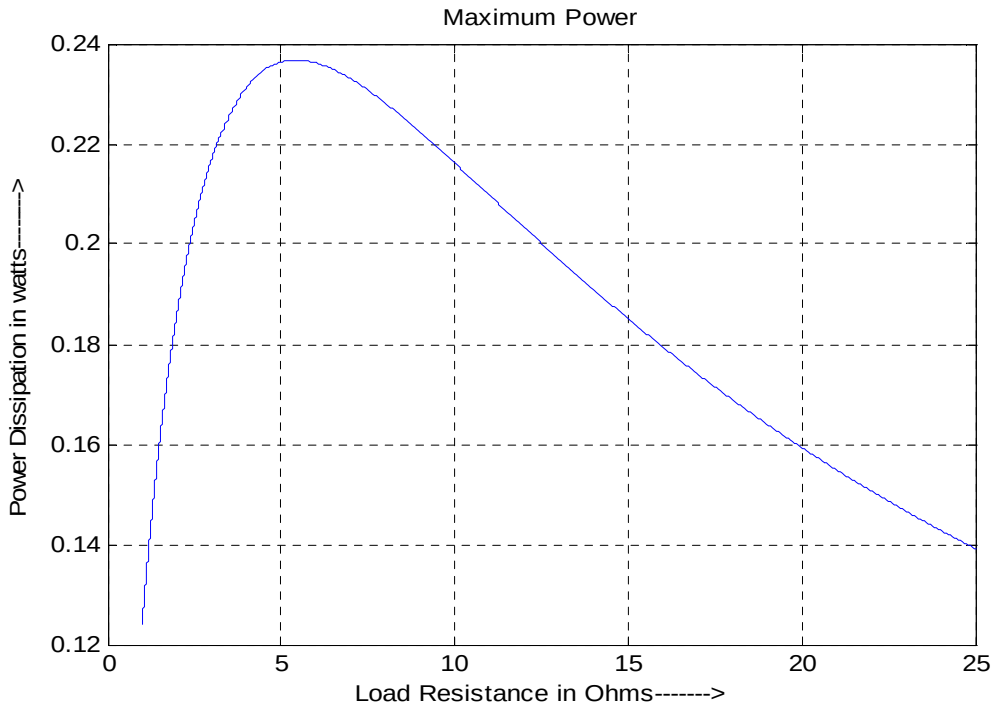
Open Circuit Voltage Vth = 2.273V
Thevenin's Resistance = 5.4545Ohms
Power acroos the load in the original circuit =0.2367 Watts
Power across Load circuit when RL=Rth=5.4545 is = 0.2368 Watts
Power across Load when RL=5 Ohms is =0.2364 Watts
Power across Load when RL=6 Ohms is = 0.2367 Watts



Power Measurements for different resistors

M-File Program for Maximum Power Transfer Theorem:

```
clc;
close all;
clear all;

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



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

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

EXPERIMENT NO: 2

TRANSIENT RESPONSES OF SERIES RLC, RL, AND RC CIRCUITS WITH SINE AND STEP INPUTS

AIM: To study the transient analysis of RLC, RL and RC circuits for sinusoidal and step inputs.

SOFTWARES USED: MATLAB Simulink / MULTISIM

THEORY:

The transient response is the fluctuation in current and voltage in a circuit (after the application of a step voltage or current) before it settles down to its steady state. This lab will focus on simulation of series RL (resistor-inductor), RC (resistor-capacitor), and RLC (resistor inductor-capacitor) circuits to demonstrate transient analysis.

Transient Response of Circuit Elements:

A. Resistors: As has been studied before, the application of a voltage V to a resistor (with resistance R ohms), results in a current I, according to the formula:

   I = V/R

   The current response to voltage change is instantaneous; a resistor has no transient response.

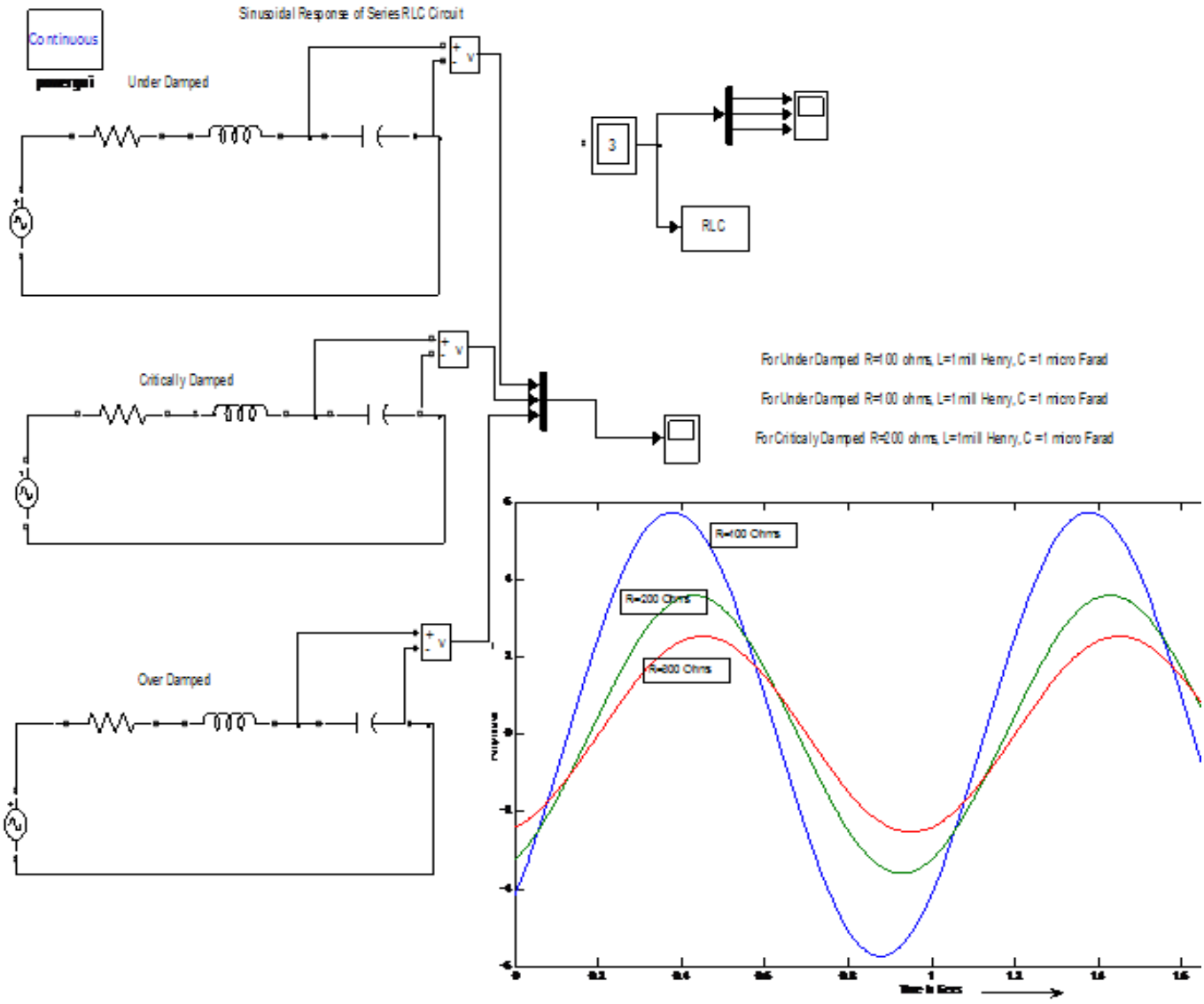
B. Inductors: A change in voltage across an inductor (with inductance L Henrys) does not result in an instantaneous change in the current through it. The i-v relationship is described with the equation: v=L di/ dt

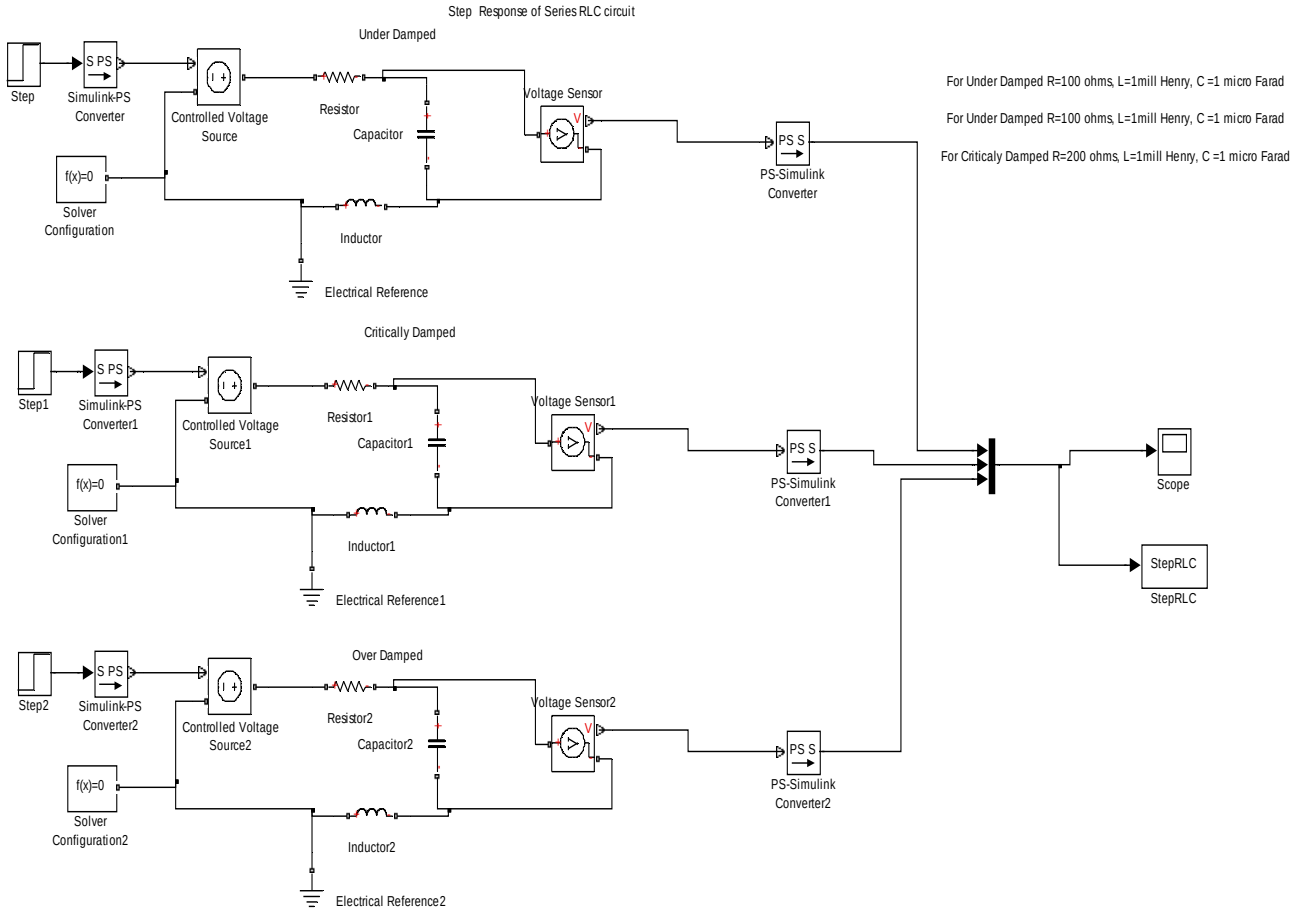
   This relationship implies that the voltage across an inductor approaches zero as the current in the circuit reaches a steady value. This means that in a DC circuit, an inductor will eventually act like a short circuit.

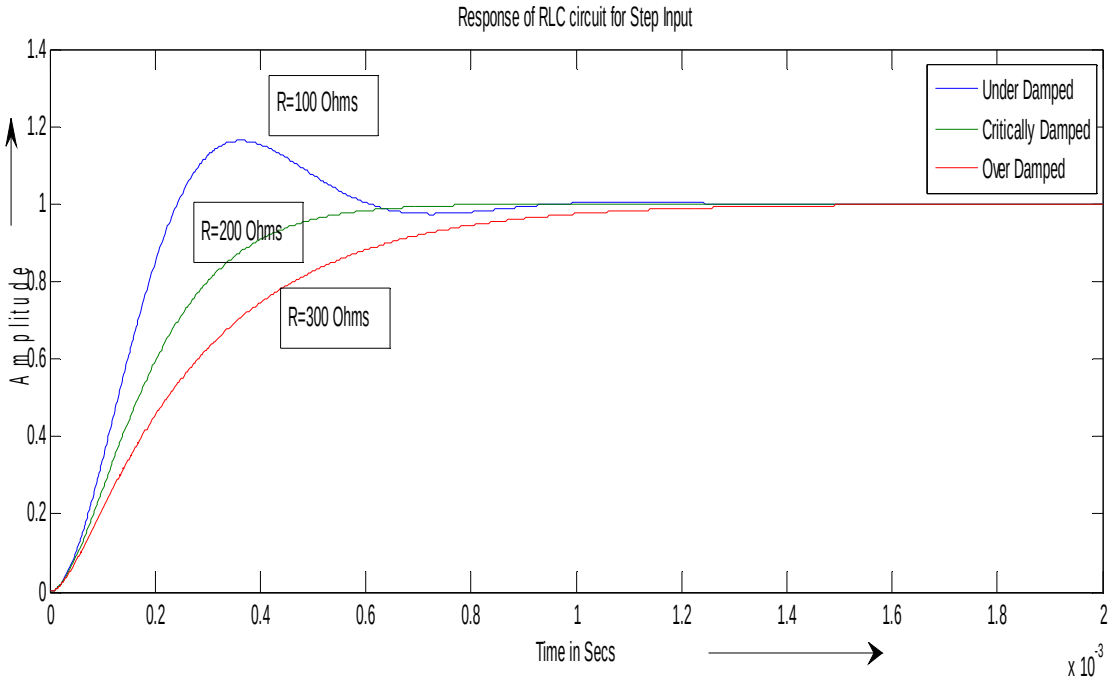
C. Capacitors: The transient response of a capacitor is such that it resists instantaneous change in the voltage across it. Its i-v relationship is described by: i=C dv /dt

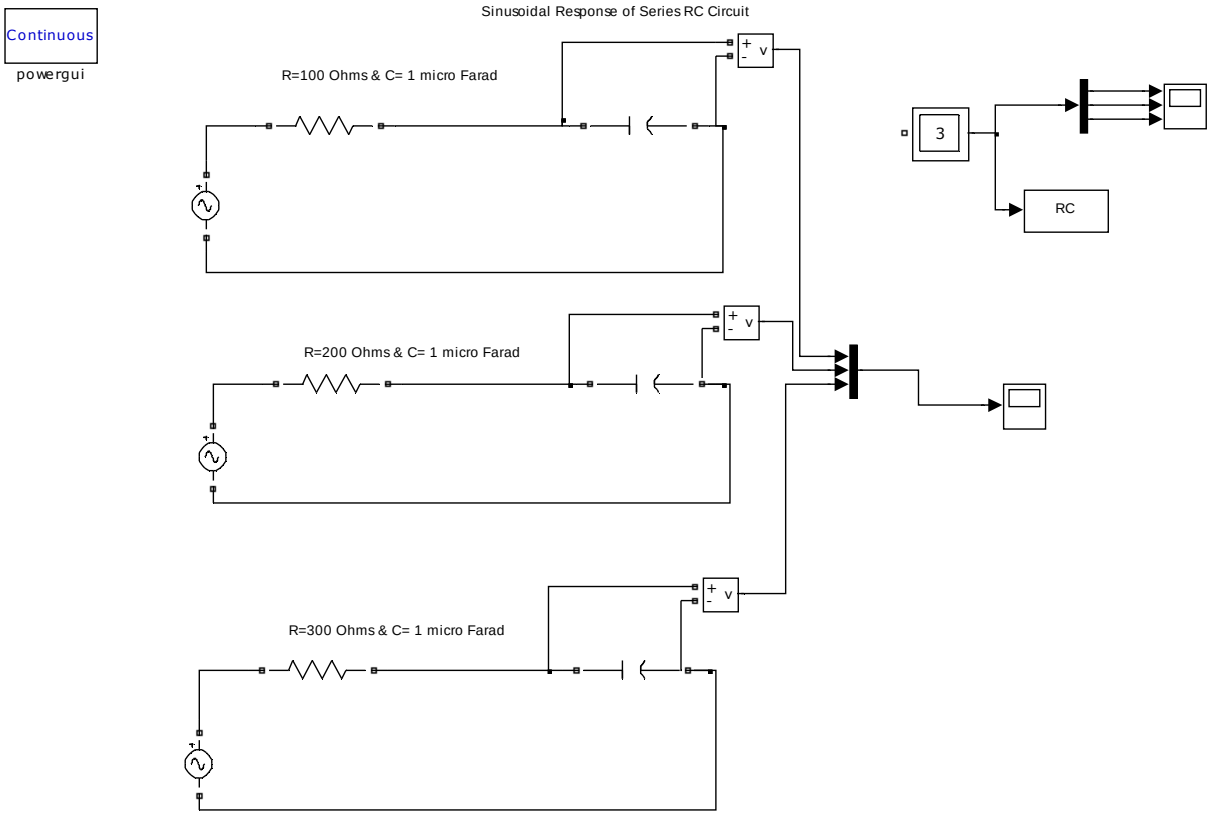
   This implies that as the voltage across the capacitor reaches a steady value, the current through it approaches zero. In other words, a capacitor eventually acts like an open circuit in a DC circuit.

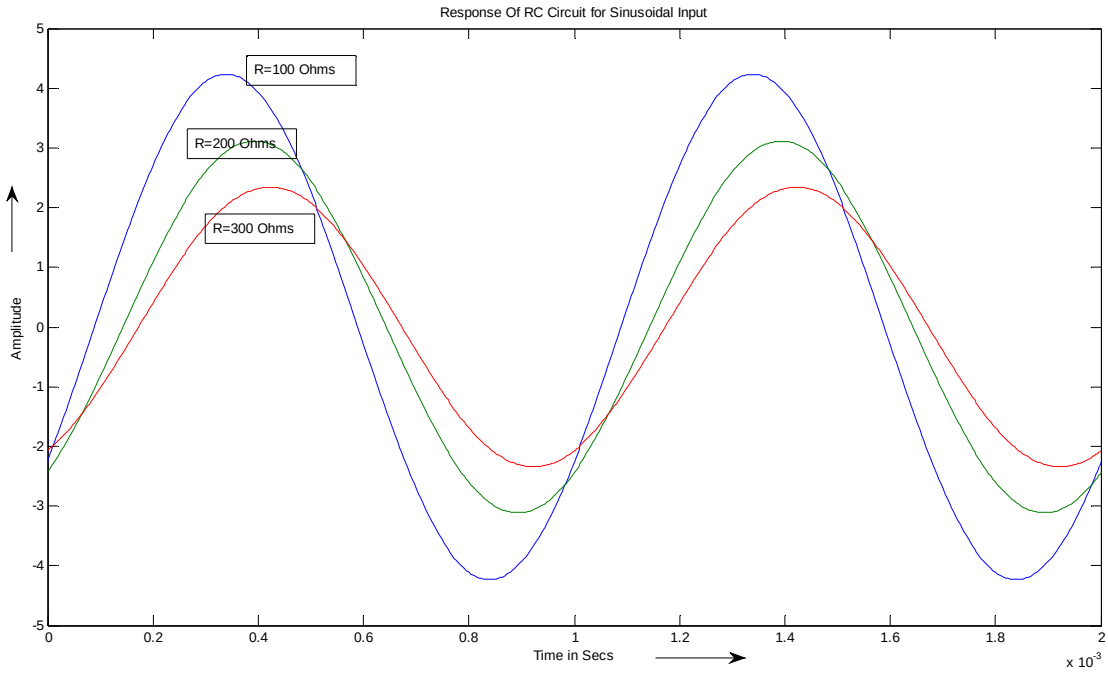
Series Combinations of Circuit Elements: Solving the circuits involves the solution of first and second order differential equations.

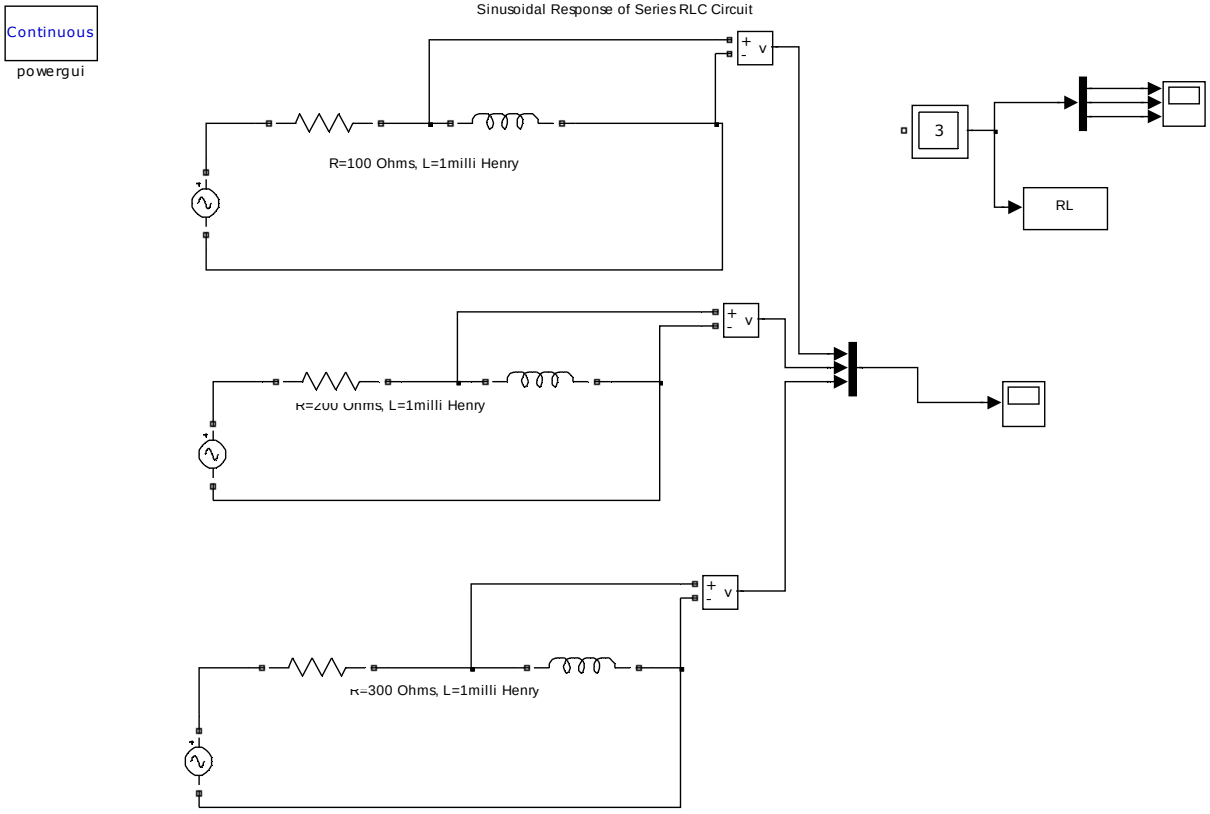


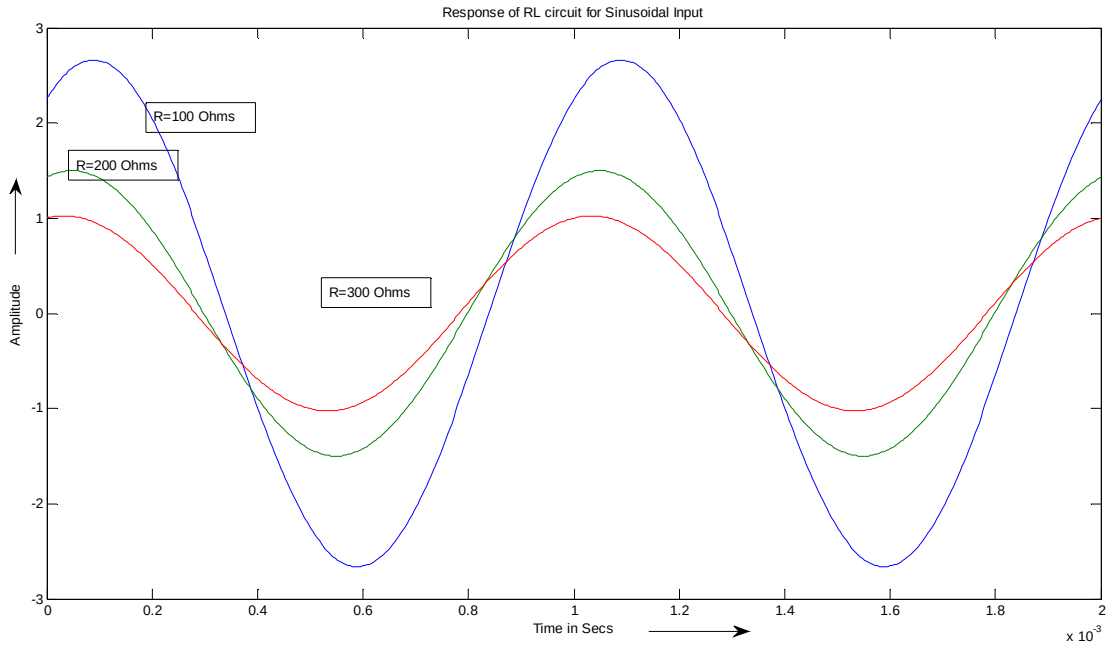

Step Response of Series RLC circuit
Under Damped
Step
Simulink-PS Converter
Controlled Voltage Source
Resistor
Capacitor
Voltage Sensor
PS-Simulink Converter
f(x)=0
Solver Configuration
Inductor
Electrical Reference
For Under Damped R=100 ohms, L=1mill Henry, C =1 micro Farad
For Under Damped R=100 ohms, L=1mill Henry, C =1 micro Farad
For Criticaly Damped R=200 ohms, L=1mill Henry, C =1 micro Farad
Critically Damped
Step1
Simulink-PS Converter1
Controlled Voltage Source1
Resistor1
Capacitor1
Voltage Sensor1
PS-Simulink Converter1
Solver Configuration1
Inductor1
Electrical Reference1
Scope
StepRLC
Over Damped
Step2
Simulink-PS Converter2
Controlled Voltage Source2
Resistor2
Capacitor2
Voltage Sensor2
PS-Simulink Converter2
Solver Configuration2
Inductor2
Electrical Reference2

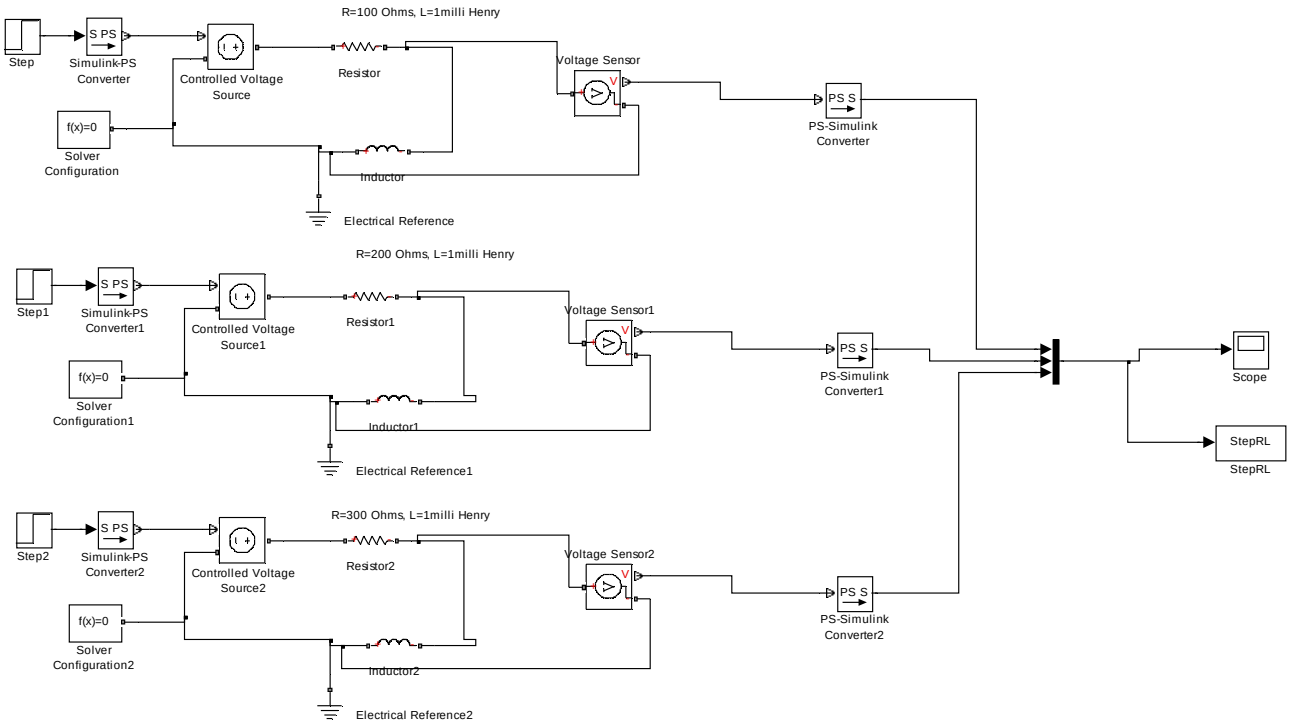

Response of RLC circuit for Step Input
R=100 Ohms
R=200 Ohms
R=300 Ohms
Under Damped
Critically Damped
Over Damped
Amplitude
Time in Secs
x 10^-3
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2

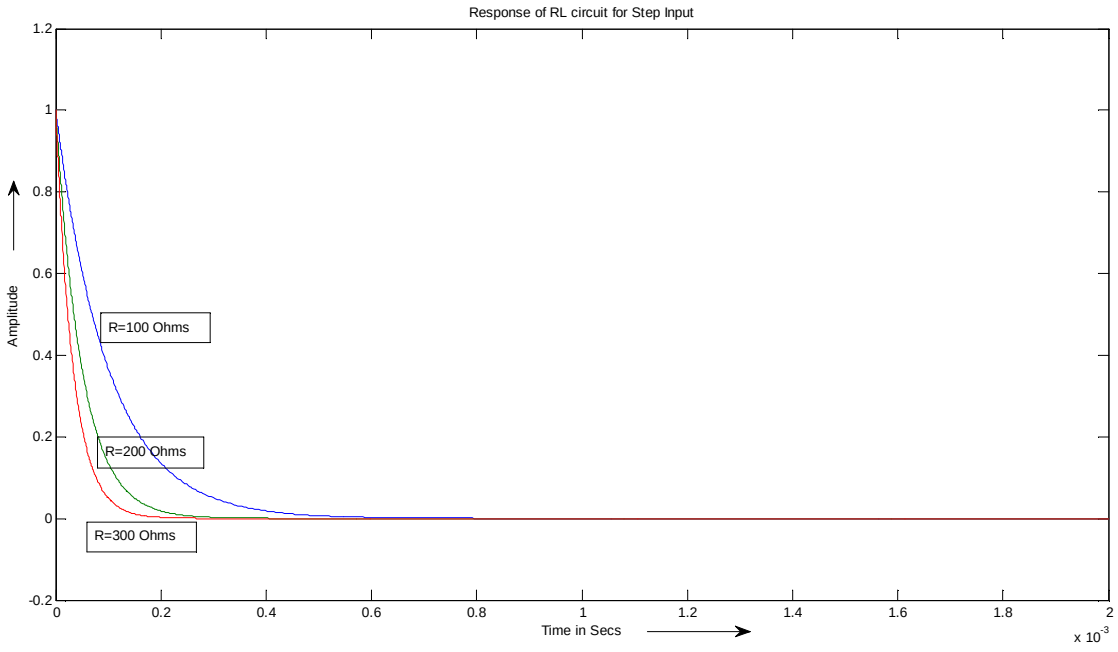

Sinusoidal Response of Series RC Circuit
Continuous
powergui
R=100 Ohms & C= 1 micro Farad
R=200 Ohms & C= 1 micro Farad
R=300 Ohms & C= 1 micro Farad
3
RC


Response Of RC Circuit for Sinusoidal Input
R=100 Ohms
R=200 Ohms
R=300 Ohms
Amplitude
Time in Secs
x 10^-3


Continuous
powergui
Sinusoidal Response of Series RLC Circuit
R=100 Ohms, L=1milli Henry
R=200 Ohms, L=1milli Henry
R=300 Ohms, L=1milli Henry
3
RL


Response of RL circuit for Sinusoidal Input
R=100 Ohms
R=200 Ohms
R=300 Ohms
Amplitude
Time in Secs
x 10^-3


R=100 Ohms, L=1milli Henry
Step
S PS
Simulink-PS Converter
Controlled Voltage Source
Resistor
Voltage Sensor
PS S
PS-Simulink Converter
f(x)=0
Solver Configuration
Inductor
Electrical Reference
R=200 Ohms, L=1milli Henry
Step1
Simulink-PS Converter1
Controlled Voltage Source1
Resistor1
Voltage Sensor1
PS-Simulink Converter1
Scope
Solver Configuration1
Inductor1
Electrical Reference1
StepRL
R=300 Ohms, L=1milli Henry
Step2
Simulink-PS Converter2
Controlled Voltage Source2
Resistor2
Voltage Sensor2
PS-Simulink Converter2
Solver Configuration2
Inductor2
Electrical Reference2


Response of RL circuit for Step Input
Amplitude
Time in Secs
x 10^-3
R=100 Ohms
R=200 Ohms
R=300 Ohms
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2

PROCEDURE:

1. Make the connections as shown in connection diagram.
2. Observe the output waveforms across a) RLC b) RC c) RL.
3. Change the value of resistance such that the output obtained at each oscilloscope is
   i) Critically damped.
   ii) Under damped.
   iii) Over damped.

RESULTS & DISCUSSIONS: The critically damped, under damped, damped response is observed for an RLC network in the simulation environment.

- The response to various inputs can be simulated .
- The response of any system designed can be simulated to verify its performance and design.

EXPERIMENT NO: 03

SERIES AND PARALLEL RESONANCE

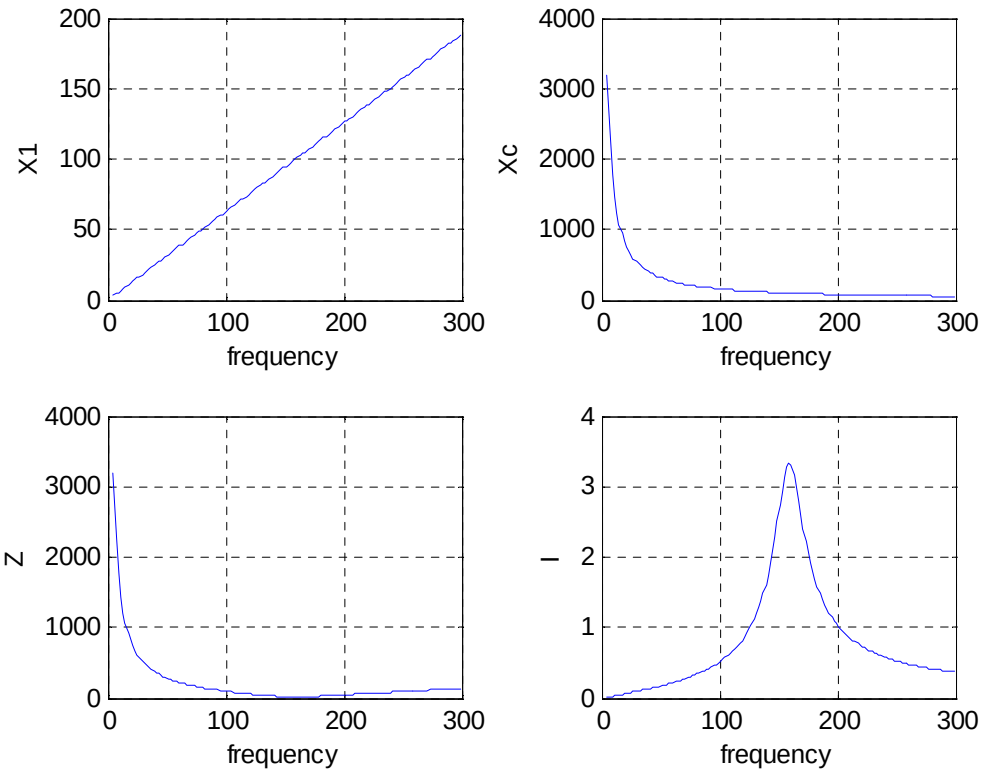
I) SERIES RESONANCE:

Aim: - To obtain the plot of of frequency vs. X_L, frequency vs. X_C , frequency vs. impedance and frequency vs. current for the given series RLC circuit and determine the resonant frequency and check by theoretical calculations.
R = 15 Ω , C = 10 μ F, L = 0.1 H, V = 50V vary frequency in steps of 1 Hz using Matlab.

```
%Program to find the Parallel Resonance
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
```



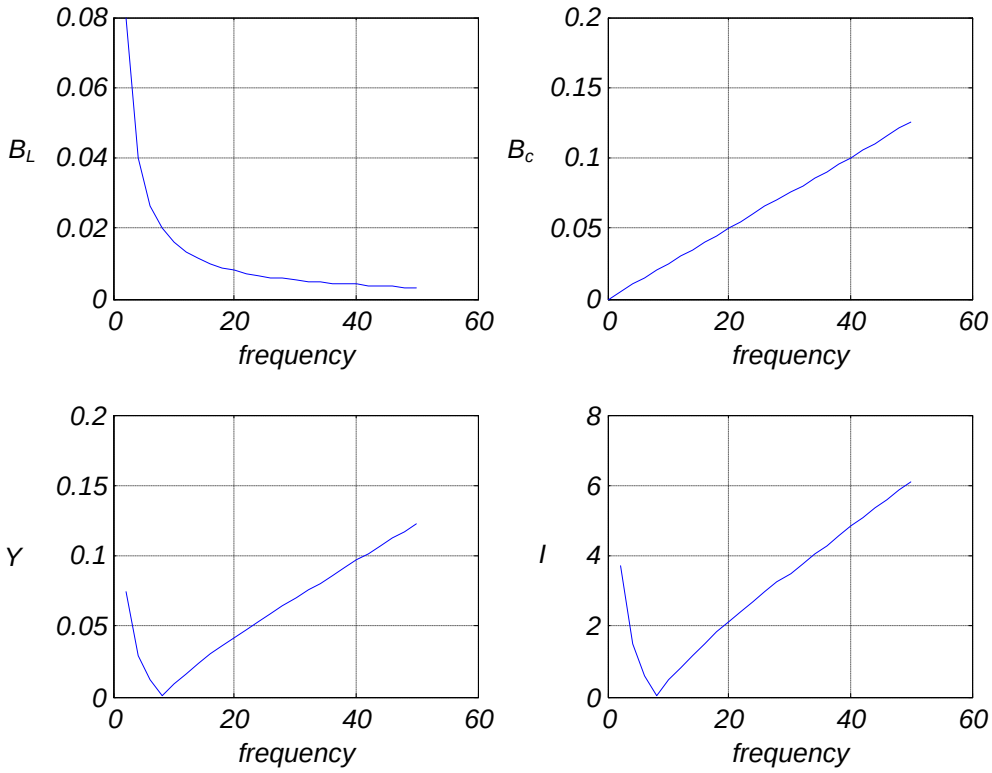
II) PARALLEL RESONANCE(Ideal Circuit):- To obtain the graphs of frequency vs. B_L , frequency vs. B_C , frequency vs. admittance and frequency vs. current vary frequency in steps for the given circuit and find the resonant frequency and check by theoretical calculations.
R = 1000 Ω , C = 400 μ F, L = 1 H, V = 50V vary frequency in steps of 1 Hz using Matlab.

```
%Program to find the Parallel Resonance
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
```



RESULTS & DISCUSSIONS: Resonance phenomena for series and parallel circuits were simulated using MATLAB m-programming.

- MATLAB m- programming allows customizing to our simulation requirement and required results/graphs can be studied and analyzed.
- Effect of resonance on current and other quantities can be seen
- Effect of L,C parameters on resonant frequency can be seen from the simulation.
- Current amplification for series circuit are observed

EXPERIMENT – 4

ROOT LOCUS, BODE AND NYQUIST PLOT

ROOT LOCUS:

AIM: To obtain the root locus of the system whose transfer function is defined by

$$G(S)= \frac{(S+5)}{S^2+7S+25}$$

PROCEDURE:

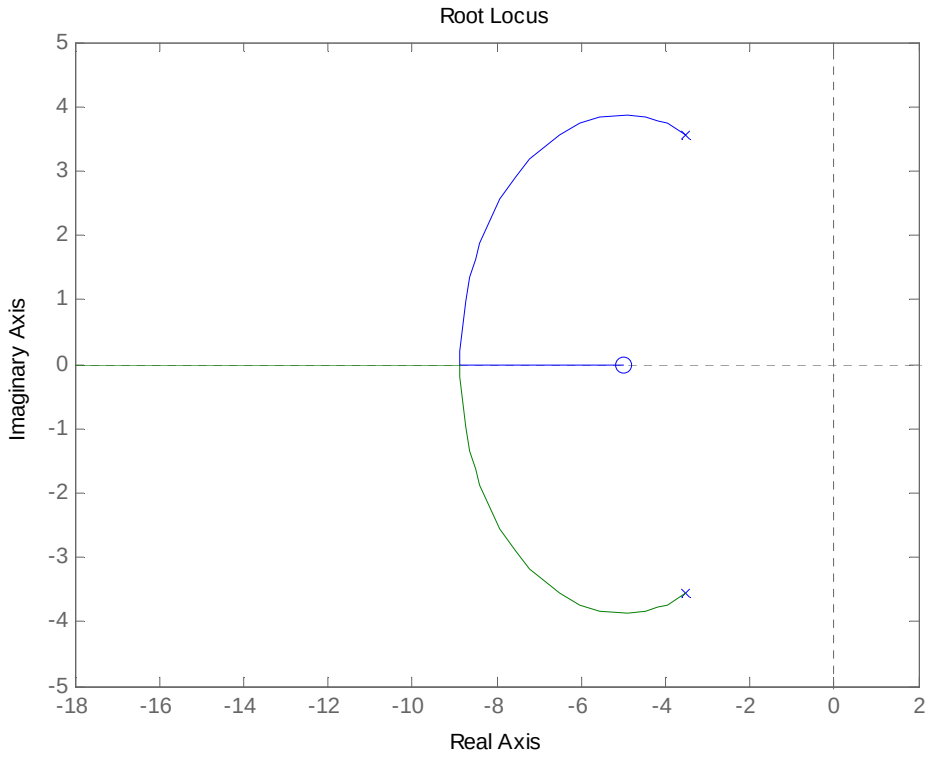
1. Input the numerator and denominator co-efficient.
2. Formulate the transfer function using the numerator and denominators co-efficient with the help of function $T = t_f(num, den)$
3. Plot the root locus of the above transfer function using rlocus(t).

PROGRAM:

```
%Program to find the root locus of transfer function%
     s+5)
% -----------
% s^2+7s+25

clc;
clear all;
close all;
% initialzations
num=input('enter the numerator coefficients---->');
den=input('enter the denominator coefficients---->');
%Transfer function
sys=tf(num,den);
rlocus(sys);
```

PROGRAM RESULT:

```
enter the numerator coefficients---->[1 5]
enter the denominator coefficients---->[1 7 25]
```



BODE PLOT:

THEORY: The gain margin is defined as the change in open loop gain required to make the system unstable. Systems with greater gain margins can withstand greater changes in system parameters before becoming unstable in closed loop. Keep in mind that unity gain in magnitude is equal to a gain of zero in dB.

The phase margin is defined as the change in open loop phase shift required to make a closed loop system unstable.

The phase margin is the difference in phase between the phase curve and -180 deg at the point corresponding to the frequency that gives us a gain of 0dB (the gain cross over frequency, Wgc).

Likewise, the gain margin is the difference between the magnitude curve and 0dB at the point corresponding to the frequency that gives us a phase of -180 deg (the phase cross over frequency, Wpc).

AIM: To obtain the bode plot and to calculate the phase margin, gain margin, phase cross over and gain cross over frequency for the systems whose open loop transfer function is given as follows.

$$G(s) = \frac{25(S+1)(S+7)}{S(S+2)(S+4)(S+8)}$$

PROCEDURE:

1. Input the zeroes, poles and gain of the given system.

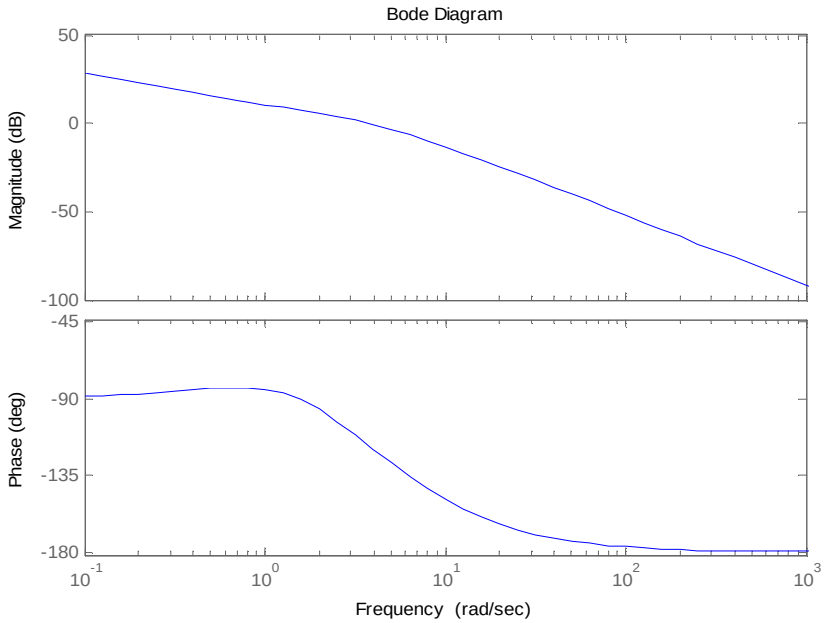
2. Formulate the transfer function from zeroes, poles and gain of the system.
3. Plot the bode plot using function bode (t).
4. Estimate PM,GM, W_{PC} , and W_{GC}. Using function margin.

PROGRAM:

```
%Program to find Bode Plot
%  25(s+1)(s+7)
% -------------
% s(s+2)(s+4)(s+8)
clc;
clear all;
close all;
% initialzations
k=input('enter the gain---->');
z=input('enter the zeros---->');
p=input('enter the ploes---->');
t=zpk(z,p,k);
bode(t);
[Gm,Pm,Wcg,Wcp]=margin(t);
disp(Gm);
disp(Pm);
disp(Wgc);
disp(Wpc);
```

PROGRAM RESULT:

```
enter the gain---->25
enter the zeros---->[-1 -7]
enter the ploes---->[0 -2 -4 -8]

Gm= Inf
Pm= 63.1105
Wgc= Inf
Wpc= 3.7440
```



NYQUIST PLOT:

AIM:

To obtain the Nyquist plot and to calculate the phase margin, gain margin, phase cross over and gain cross over frequency for the systems whose open loop transfer function is given as follows.

$$G(S) = \frac{50(S+1)}{S(S+3)(S+5)}$$

PROCEDURE:

1. Input the zeroes, poles and gain of the given system.
2. Formulate the transfer function from zeroes, poles and gain of the system.
3. Plot the nyquist plot using function nyquist(t).
4. Estimate PM,GM, W_{PC} , and W_{GC}. Using function margin.

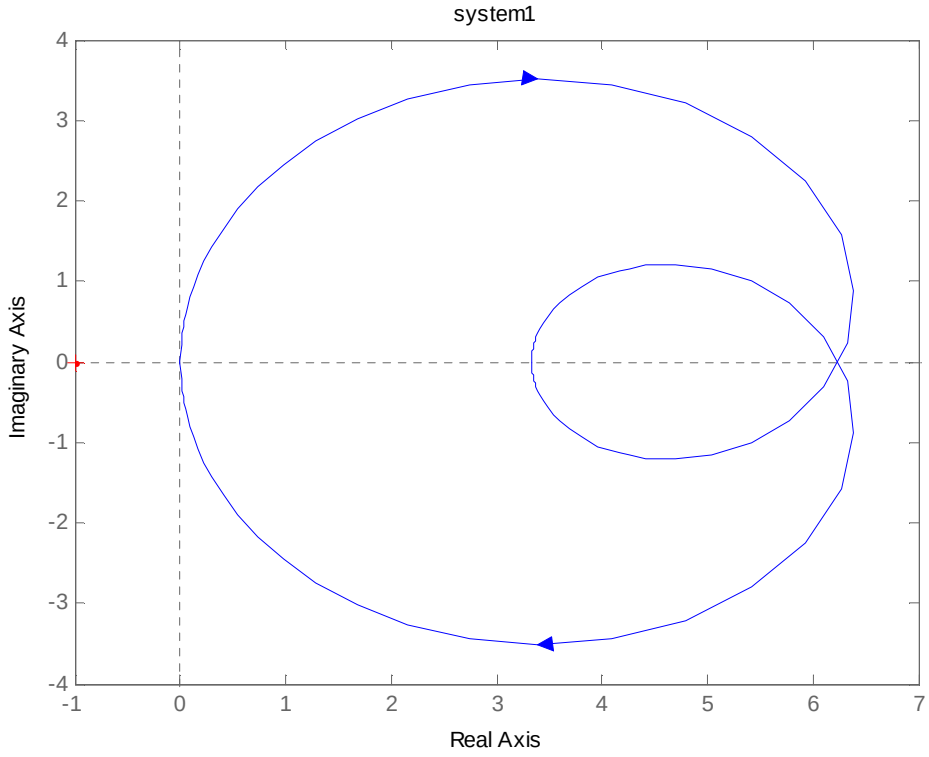
PROGRAM:

```
%Program to find the Nyquist Plot

%  50(s+1)
% --------
% s(s+3)(s+5)

clc;
clear all;
close all;
```

```
% initialzations
num=input('enter the numerator coefficients---->');
den=input('enter the denominator coefficients---->');
sys=tf(num,den);
nyquist(sys);
title('system1');
[Gm,Pm,Wcg,Wcp]=margin(sys);
disp(Gm);
disp(Pm);
disp(Wgc);
disp(Wpc);
```

PROGRAM RESULT:

```
enter the numerator coefficients---->[50 50]
enter the denominator coefficients---->[1 8 15]

Gm= Inf
Pm= 98.0516
Wgc=Inf
Wpc=49.6681
```



RESULTS & DISCUSSIONS: Root Locus, Bode plot and Nyquist plot determined using the built-in functions of MATLAB.

- They are a powerful tool to design systems to required performance.

- In order to determine the stability of the system using the **root locus technique** we find the range of values of k for which the complete performance of the system will be satisfactory and the operation is stable.
- **Bode plots** provides relative stability in terms of **gain margin** and **phase margin**

EXPERIMENT – 5

TRANSFER FUNCTION ANALYSIS OF I) TIME RESPONSE FOR STEP INPUT II) FREQUENCY RESPONSE FOR SINUSOIDAL INPUT.

AIM: To find the I) Time response for step input II) Frequency response for sinusoidal input.

I .TIME RESPONSE FOR STEP INPUT:

SOFTWARES USED: MATLAB

THEORY: The general expression of transfer function of a second order control system is given as

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

Here, ζ and ω_n are damping ratio and natural frequency of the system respectively
There are number of common terms in transient response characteristics and which are

1. **Delay time** (t_d) is the time required to reach at 50% of its final value by a time response signal during its first cycle of oscillation. $T_d = \frac{1+0.7\zeta}{\omega_n}$

2. **Rise time** (t_r) is the time required to reach at final value by a under damped time response signal during its first cycle of oscillation. If the signal is over damped, then rise time is counted as the time required by the response to rise from 10% to 90% of its final value. $\beta = \tan^{-1}\left(\frac{\sqrt{1-\zeta^2}}{\zeta}\right)$

$$T_r = \frac{1}{\omega_d}\tan^{-1}\left(-\frac{\sqrt{1-\zeta^2}}{\zeta}\right) = \frac{\pi-\beta}{\omega_d}$$

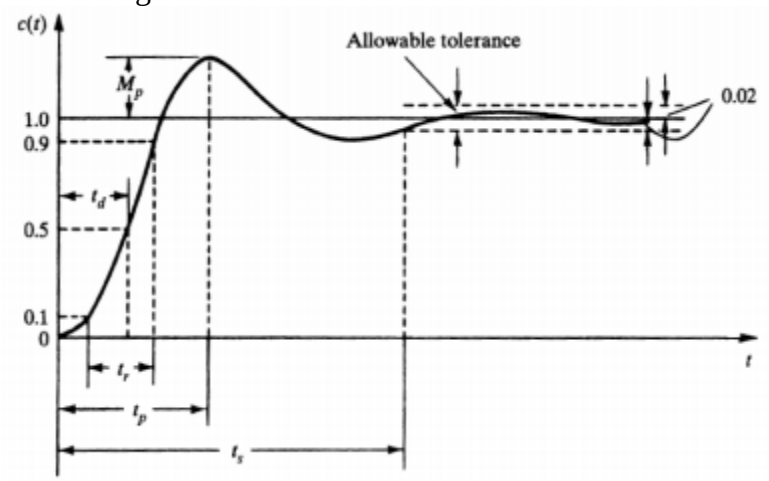
3. **Peak time** (t_p) is simply the time required by response to reach its first peak i.e. the peak of first cycle of oscillation, or first overshoot. $T_p = \frac{\pi}{\omega_d} = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}}$

4. **Maximum overshoot** (M_p) is straight way difference between the magnitude of the highest peak of time response and magnitude of its steady state. Maximum overshoot is expressed in term of percentage of steady-state value of the response. As the first peak of response is normally maximum in magnitude, maximum overshoot is simply normalized difference between first peak and steady-state value of a response. $M_p = e^{-\pi\zeta/\sqrt{1-\zeta^2}}$

$$M_p\% = e^{-\pi\zeta/\sqrt{1-\zeta^2}} \times 100\%$$

5. **Settling time** (t_s) is the time required for a response to become steady. It is defined as the time required by the response to reach and steady within specified range of 2 % to 5 % of its final value. $T_s = \frac{4}{\zeta\omega_n} \qquad (2\%\ \text{Criterion})$

6. **Steady-state error** (e_{ss}) is the difference between actual output and desired output at the infinite range of time. $e_{ss} = \lim_{t\to\infty} [r(t) - c(t)]$



PROBLEM STATEMENT: For the closed loop system defined by

$$\frac{C(S)}{R(S)} = \frac{100}{S^2 + 12S + 100}$$

Plot the unit step response curve and find time domain specifications

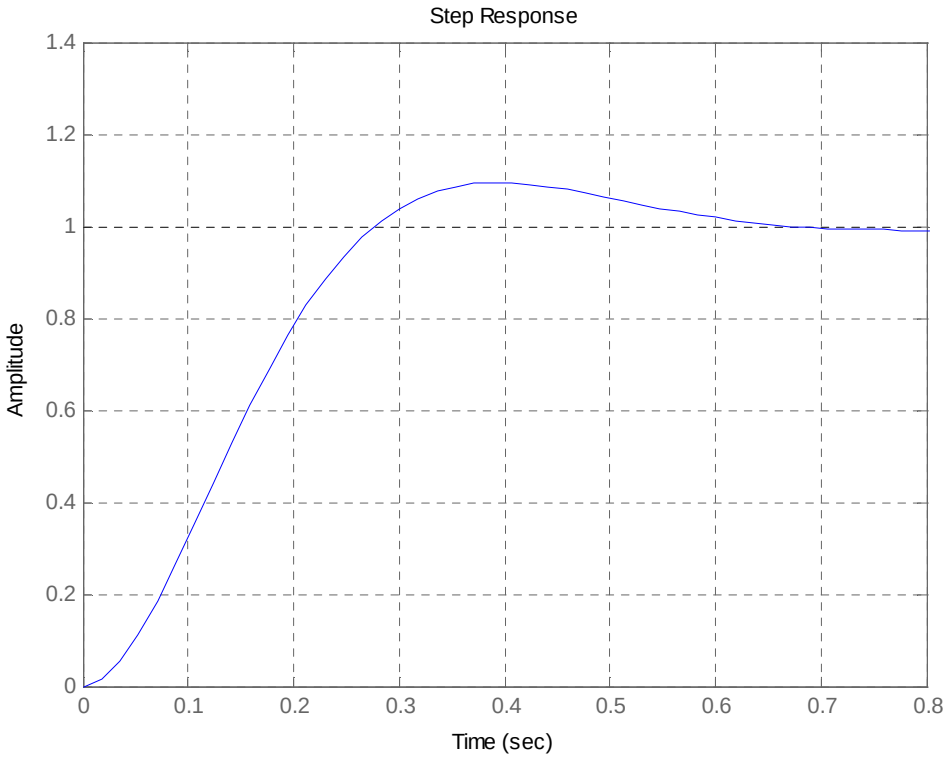
PROGRAM:

```
clc;
clear all;
close all;
num=input('enter the numerator coefficients---->');
den=input('enter the denominator coefficients---->');
system=tf(num,den);
system
step(system)
grid on;
wn=sqrt(den(1,3));
zeta= den(1,2)/(2*wn);
wd=wn*sqrt(1-zeta^2);
disp('Delay time in seconds is')
td=(1+0.7*zeta)/wd
disp('Rise time in seconds is')
theta=atan(sqrt(1-zeta^2)/zeta);
tr=(pi-theta)/wd
disp('Peak time in seconds');
```

```
tp=pi/wd
disp('Peak overshoot is');
mp=exp(-zeta*pi/sqrt(1-zeta^2))*100
disp('settling time in seconds is');
ts=4/(zeta*wn)
```

PROGRAM RESULT:

```
enter the numerator coefficients---->100
enter the denominator coefficients---->[1 12 100]

Transfer function:
      100
----------------
s^2 + 12 s + 100

Delay time in seconds is

td =

    0.1775

Rise time in seconds is

tr =

    0.2768

Peak time in seconds

tp =

    0.3927

Peak overshoot is

mp =

    9.4780

settling time in seconds is

ts =

    0.6667
```



II. FREQUENCY RESPONSE FOR SINUSOIDAL INPUT

By the term frequency response, we mean the steady-state response of a system to a sinusoidal input. Industrial control systems are often designed using frequency response methods. Many techniques are available in the frequency response methods for the analysis and design of control systems.

Consider a system with sinusoidal input $r(t) = A\sin\omega t$. The steady-state output may be written as, $c(t) = B\sin(\omega t + \phi)$. The magnitude and the phase relationship between the sinusoidal input and the steady-state output of a system is called *frequency response*. The frequency response test is performed by keeping the amplitude *A* fixed and determining *B* and Φ for a suitable range of frequencies. Whenever it is not possible to obtain the transfer function of a system through analytical techniques, frequency response test can be used to compute its transfer function.

The design and adjustment of open-loop transfer function of a system for specified closed-loop performance is carried out more easily in frequency domain. Further, the effects of noise and parameter variations are relatively easy to visualize and assess through frequency response. The Nyquist criteria is used to extract information about the stability and the relative stability of a system in frequency domain.

The transfer function of a standard second-order system can be written as,

$$T(s) = \frac{C(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}.$$

Substituting *s* by $j\omega$ we obtain, $T(j\omega) = \dfrac{\omega_n^2}{(j\omega)^2 + 2\zeta\omega_n(j\omega) + \omega_n^2} = \dfrac{1}{(1-u^2) + j2\zeta u}$.

Where, $u = \omega/\omega_n$ is the normalized signal frequency. From the above equation we get,

$$|T(j\omega)| = M = \frac{1}{\sqrt{(1-u^2)^2 + (2\zeta u)^2}} \ .$$

$$\angle T(j\omega) = \phi = -\tan^{-1}[2\zeta u/(1-u^2)]$$

The steady-state output of the system for a sinusoidal input of unit magnitude and variable frequency ω is given by,

$$c(t) = \frac{1}{\sqrt{(1-u^2)^2 + (2\zeta u)^2}} \sin\left(\omega t - \tan^{-1}\frac{2\zeta u}{1-u^2}\right).$$

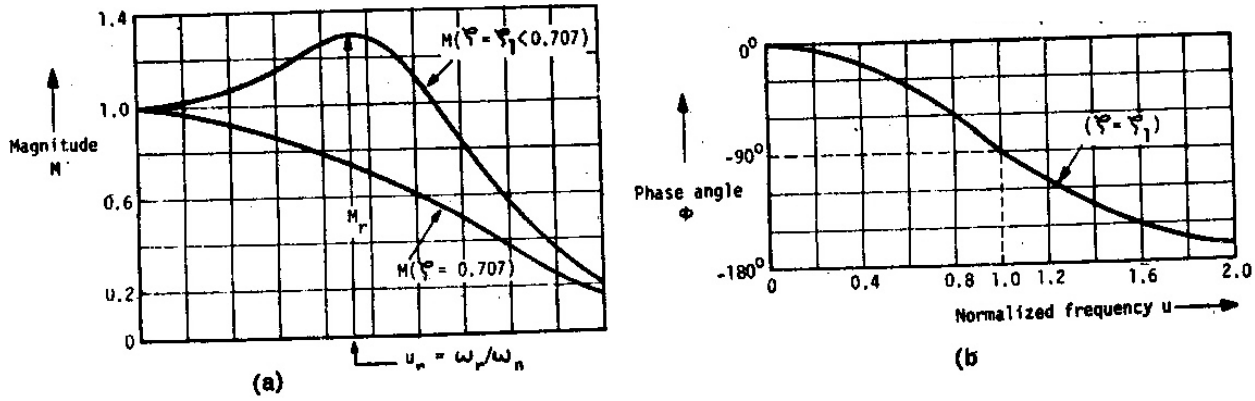
It is seen from the above equation that when,

$u = 0, \quad M = 1 \quad$ and $\quad \phi = 0$

$u = 1, \quad M = \frac{1}{2\zeta} \quad$ and $\quad \phi = -\pi/2$

$u = \infty, \quad M \to 0 \quad$ and $\quad \phi \to -\pi$

The magnitude and phase angle characteristics for normalized frequency u for certain values of ζ are shown in figure in the next page.



The frequency where ***M*** has a peak value is called *resonant frequency*. At this point the slope of the magnitude curve is zero. Setting $\left.\frac{dM}{du}\right|_{u=u_r} = 0$ we get, $\quad -\frac{1}{2}\frac{\left[-4(1-u_r^2)u_r + 8\zeta^2 u_r\right]}{\left[(1-u_r^2)^2 + (2\zeta u_r)^2\right]^{3/2}} = 0.$

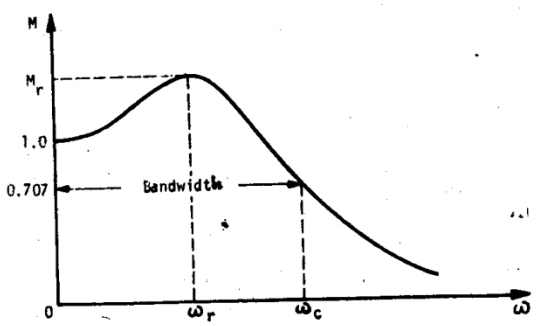
Solving, $u_r = \sqrt{1-2\zeta^2}$ or, resonant frequency $\omega_r = \omega_n\sqrt{1-2\zeta^2}$. …… …… (01)

The *resonant peak* is given by resonant peak, $M_r = \frac{1}{2\zeta\sqrt{1-\zeta^2}}$. …… (02)

- For, $\zeta > \frac{1}{\sqrt{2}}$ $(= 0.707)$, the resonant frequency does not exist and *M* decreases monotonically with increasing *u*.
- For $0 < \zeta < \frac{1}{\sqrt{2}}$, the resonant frequency is always less than ω_n and the resonant peak has a value greater than 1.

From equation (01) and (02) it is seen that The resonant peak M_r of frequency response is indicative of **damping factor** and the resonant frequency ω_r is indicative of **natural frequency** for a given ζ and hence indicative of settling time.

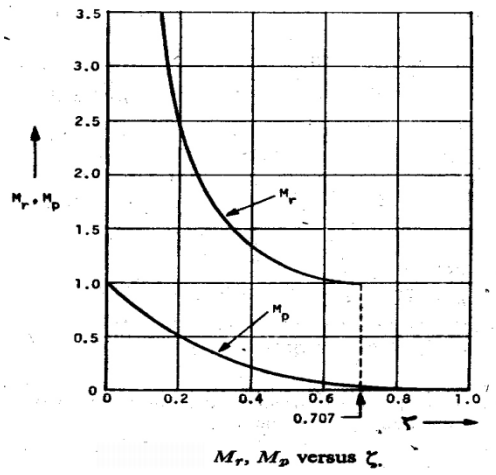
For $\omega > \omega_r$, M decreases monotonically. The frequency at which M has a value of $\frac{1}{\sqrt{2}}$ is called the cut-off frequency ω_c. The range of frequencies over which M is equal to or greater than $\frac{1}{\sqrt{2}}$ is defined as bandwidth, ω_b.



The bandwidth of a second-order system is given by,

$$\omega_b = \omega_n\left[1-2\zeta^2+\sqrt{2-4\zeta^2+4\zeta^4}\right]^{1/2} \quad \text{.....(03)}$$

Figure below shows the plot of resonant peak of frequency response and the peak overshoot of step response as a function of ζ.



M_r, M_p versus ζ.

It is seen that the two performance indices are correlated as both are the functions of the system damping factor ζ only. For $\zeta > \frac{1}{\sqrt{2}}$ $(=0.707)$ the resonant peak does not exist and the correlation breaks down. For this range of ζ, M_p is hardly perceptible.

From equation (03) it is seen that the bandwidth is indicative of natural frequency and hence indicative of settling time, i.e., the speed of response for a given ζ.

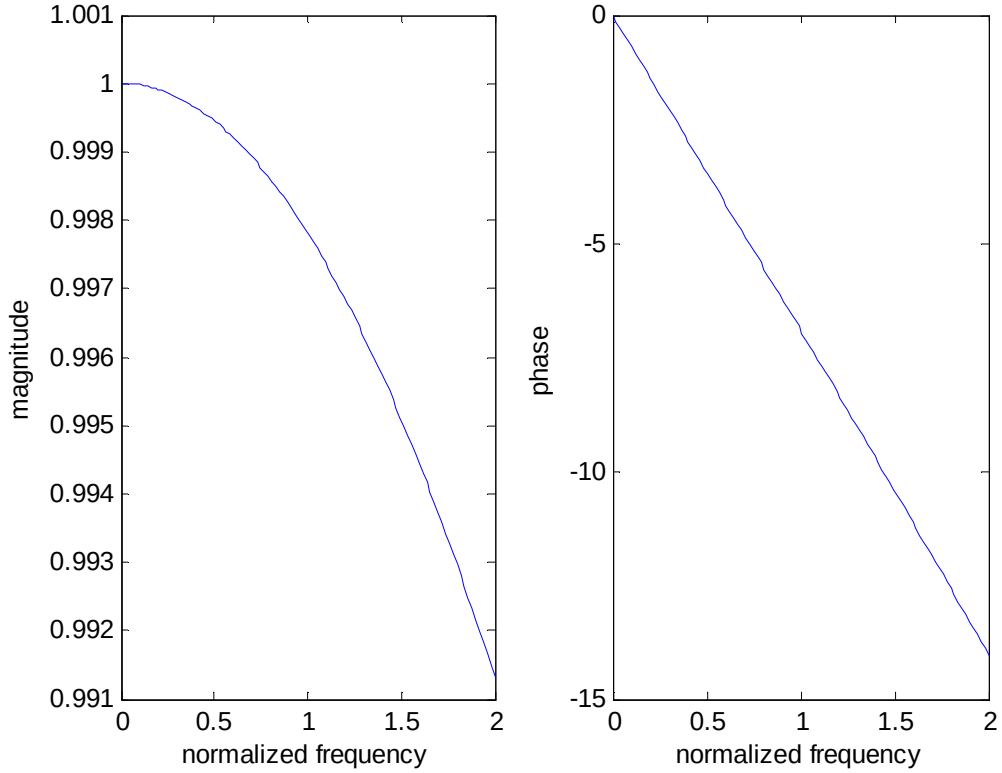
PROGRAM:

```
%Frequency Response of second order system
clc;
clear all;
close all;
num=input('enter the numerator coefficients---->');
den=input('enter the denominator coefficients---->');
%Transfer function
sys=tf(num,den);
wn=sqrt(den(1,3));
zeta= den(1,2)/(2*wn);
w=linspace(0,2);
u=w/wn;
len=length(u);
for k=1:len
    m(k)=1/(sqrt((1-u(k)^2)+(2*zeta*u(k))^2));
    phi(k)=-atan((2*zeta*u(k))/(1-u(k)^2))*180/pi;
end
subplot(1,2,1)
plot(w,m)
xlabel('normalized frequency')
```

```
ylabel('magnitude')
subplot(1,2,2)
plot(w,phi)
xlabel('normalized frequency')
ylabel('phase')
disp('resonant peak is');
mr=1/(2*zeta*sqrt(1-zeta^2))
disp('resonant frequency in rad/sec is');
wr=wn*sqrt(1-2*zeta^2)
disp('bandwidth in rad/sec is');
wb=wn*sqrt(1-2*zeta^2+sqrt(2-4*zeta^2+4*zeta^4))
disp('phase margin in degrees is')
pm=180+(atan(2*zeta/sqrt(-2*zeta^2+sqrt(4*zeta^4 +1))))*180/pi
```

PROGRAM RESULT:

```
enter the numerator coefficients---->100
enter the denominator coefficients---->[1 12 100]
resonant peak is

mr =

    1.0417
resonant frequency in rad/sec is

wr =

    5.2915

bandwidth in rad/sec is

wb =

   11.4824

phase margin in degrees is

pm =

  239.1873
```



RESULTS & DISCUSSIONS: Defining transfer functions and finding response using these transfer functions has been simulated using MATLAB.

Responses can be studied with addition of controllers and their effect on performance.

EXPERIMENT NO: 6

DESIGN OF LAG, LEAD AND LAG-LEAD COMPENSATOR

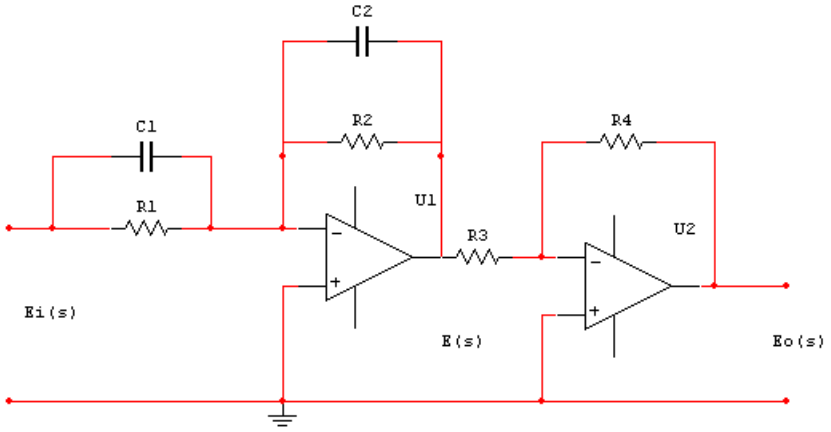
AIM: To design lag, lead compensator, lag-lead compensator

THEORY:

The primary objective of this experiment is to design the compensation of single –input-single-output linear time invariant control system.

Compensation is the modification of the system dynamics to satisfy the given specification. The compensation is done by adding some suitable device in which is called as compensator. Compensator is realized by such a way as to meet the performance specifications.

If sinusoidal input is applied to a network and if the steady state output has a phase lead, then the network is called a lead network, and if the output has a phase lag then the network is called as a phase lag network. Compensators are realized in our experiments using op-amps , electrical RC network as shown in figure.



$$\frac{Eo(s)}{Ei(s)} = \frac{R_2}{R_1}\frac{R_4}{R_3}\frac{R_1C_1s+1}{R_2C_2s+1} = \frac{R_4C_1s+\dfrac{1}{R_1C_1}}{R_2C_2s+\dfrac{1}{R_2C_2}} = K_c\alpha\frac{Ts+1}{\alpha Ts+1}$$

where

$$T = R_1C_1$$

$$\alpha T = R_2C_2$$

$$K_c = \frac{R_4C_1}{R_3C_2}$$

$$\alpha = \frac{R_2C_2}{R_1C_1}$$

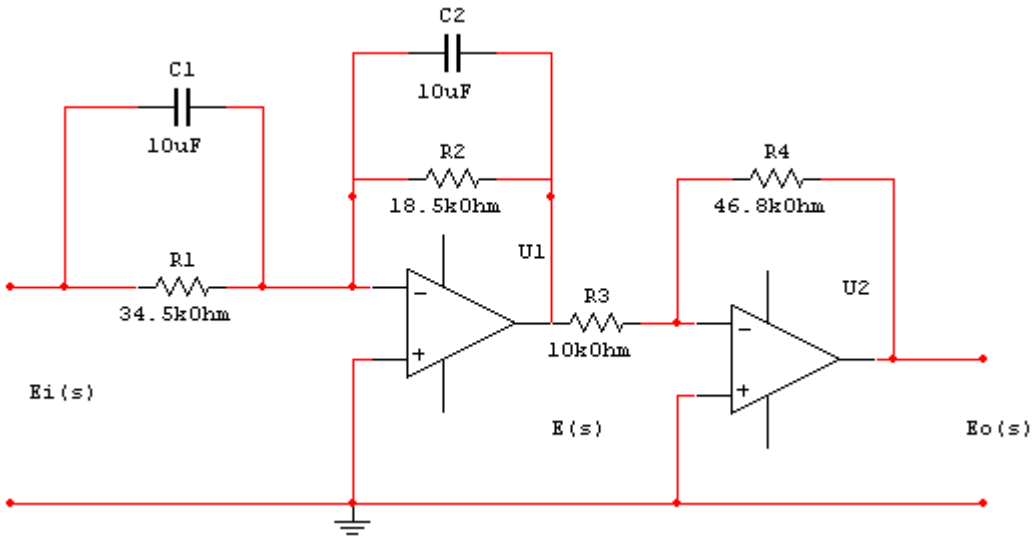
This network is a lead network if $R_1 C_1 > R_2 C_2$ or $\alpha<1$.
Or this is a lag network if $\alpha>1$ or $R_1 C_1 < R_2 C_2$

PROCEDURE

1) Consider any uncompensated system

2) Design the lead and lag compensator from the given circuit using the above equations.
3) Connect this design compensator to uncompensated system in series compensation.
4) Then find the closed loop transfer function equation for this compensated system
5) Plot the response for both uncompensated and compensated system

For Lead Compensator



The closed loop transfer function equation for the compensated system becomes

$$\frac{C(s)}{R(S)} = \frac{18.7(S+2.9)}{s(s+2)(s+5.4)+18.7(s+2.9)}$$

$$= \frac{18.7S+54.23}{s^3+7.4s^2+29.5s+54.23}$$

Hence

numc= [0 0 18.7 54.23]
denc= [1 7.4 29.5 54.23]

for the uncompensated system the closed loop transfer function is

$$\frac{C(s)}{R(S)} = \frac{4}{s^2+2s+4}$$

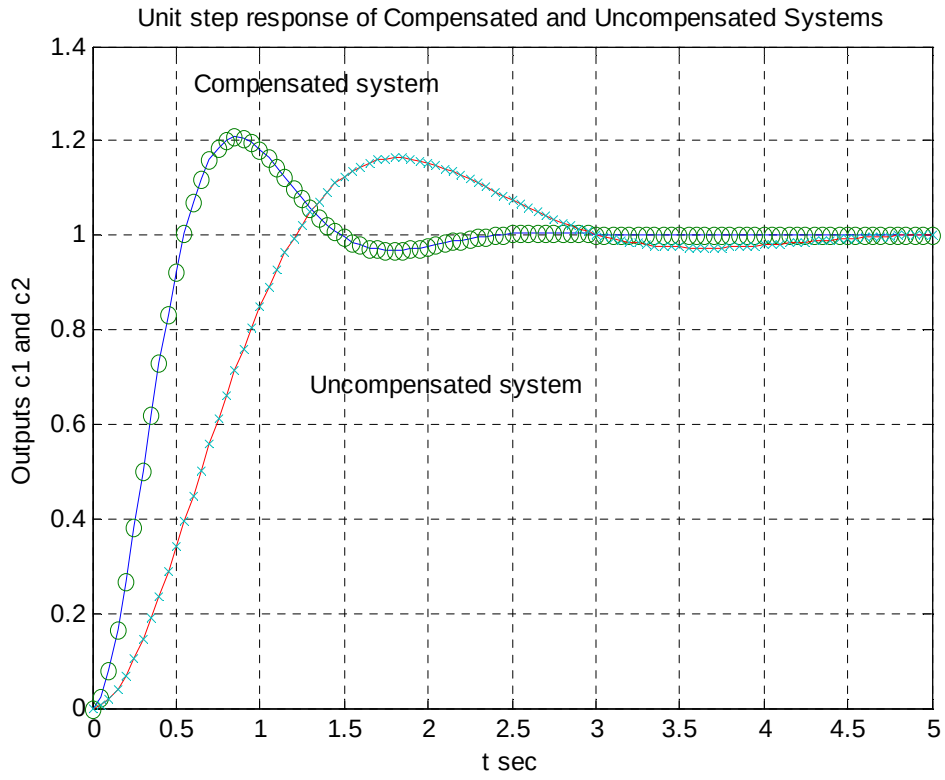
Hence

numc= [0 0 4]
denc= [1 2 4]

PROGRAM:

```
% Unit Step Response of Compensated and Uncompensated systems

numc=[0 0 18.7 54.23];
denc=[1 7.4 29.5 54.23];
```

```
num=[0 0 4];
den=[1 2 4];
t=0:0.05:5;
[c1,x1,t]=step(numc,denc,t);
[c2,x2,t]=step(num,den,t);
plot(t,c1,t,c1,'o',t,c2,t,c2,'x');
grid;
title('Unit step response of Compensated and Uncompensated Systems');
xlabel('t sec')
ylabel('Outputs c1 and c2');
text(0.6,1.32,'Compensated system');
text(1.3,0.68,'Uncompensated system');
```



For Lag Compensator

The closed loop transfer function equation for the compensated system becomes

$$\frac{C(s)}{R(S)} = \frac{1.0235(s+0.05)}{s(s+0.005)(s+1)(s+2)+1.0235(s+0.05)}$$

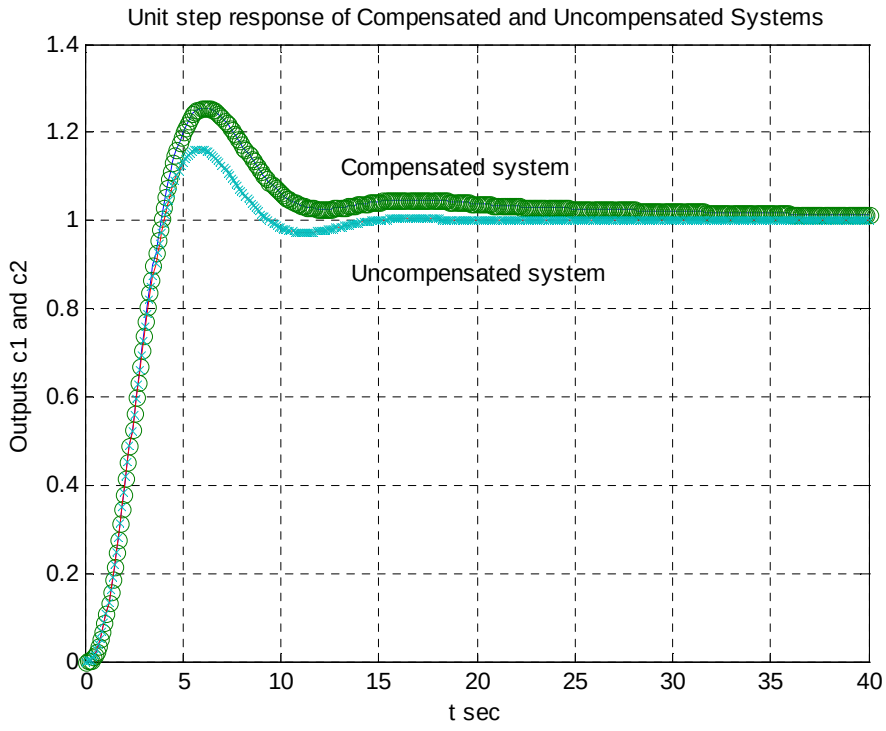
$$= \frac{1.0235s+0.0512}{s^4+3.005s^3+2.015s^2+1.0335s+0.0512}$$

for the uncompensated system the closed loop transfer function is

$$\frac{C(s)}{R(S)} = \frac{1.06}{s(s+1)(s+2)+1.06}$$

PROGRAM:

```
% Unit Step Response of Compensated and Uncompensated systems

numc=[0 0 0 1.0235 0.0512];
denc=[1 3.005 2.015 1.0335 0.0512];
num=[0 0 0 1.06];
den=[1 3 2 1.06];
t=0:0.1:40;
[c1,x1,t]=step(numc,denc,t);
[c2,x2,t]=step(num,den,t);
plot(t,c1,t,c1,'o',t,c2,t,c2,'x');
grid;
text(13,1.12,'Compensated system');
text(13.6,0.88,'Uncompensated system');
title('Unit step response of Compensated and Uncompensated Systems');
xlabel('t sec')
ylabel('Outputs c1 and c2');
```



RESULTS & DISCUSSIONS: Compensators are added to existing systems to improve their performance.

- Such compensators based on the change required in performance of the system have been designed and improvement in performance analyzed using MATLAB.
- Change in the performance of the system with compensator is observed.

EXPERIMENT NO: 7

SIMULATION OF TEMPERATURE MONITORING USING LABVIEW

AIM: To monitor the temperature of a tank

.

APPARATUS:

- LabVIEW Software
- Desktop computer.

THEORY: Temperature is monitored using LabVIEW. The process variable here is the temperature of liquid. The formula to find the temperature is given by the formula.

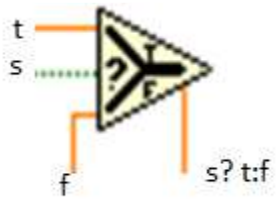
$\Delta T = Q/(M*S)$

where,
M=mass
Q=heat
ΔT=difference in temperature
S=specific heat

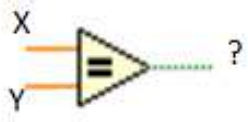
Here temperature is monitored in both ℃ and °F. A switch is provided to manually switch between c and f vice versa. A sub VI is created to convert ℃ to°F.
The switch is provided manually switch between c and f and vice versa. A sub VI is created to convert ℃ to°F.

The **element** used in the temperature monitoring system is as follows:

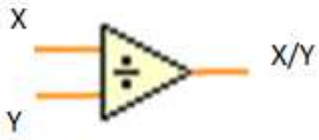
1) **Select switch**: Returns the value wired to the t input, depending on the value of's'. If 's' is true, this function returns the value wired to 't '.if 's' is false ,this function returns the value wired to 'f'.



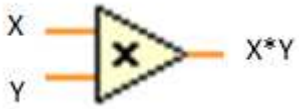
2)**Comparison functions** compare data and operations that are based on a comparison such as finding the minimum and maximum ranges for a group or array of values.



3) Divide: compute coefficient of input.



4)Multiply: returns the product of the inputs.



5) Sub-Vi:

The procedure to create a Sub-VI is as follows:

1. Select the path of your code you want to turn into a sub VI.
2. From the edit menu, select create sub VI
3. Lab view will automatically create a sub VI.
4. Clean up automatically create wires etc...
5. Create a suitable icon for you sub VI.

6) While Loop:

A while loop repeat the sub diagram inside it until the conditional terminal, an input terminal receives a particular Boolean value. The Boolean value depends upon the continuation behavior of the while loop. Right click the conditional terminal and select stop if true or continue if true from the shortcut menu. You can also wire an error to the conditional terminal. The while loop always executes at least once.

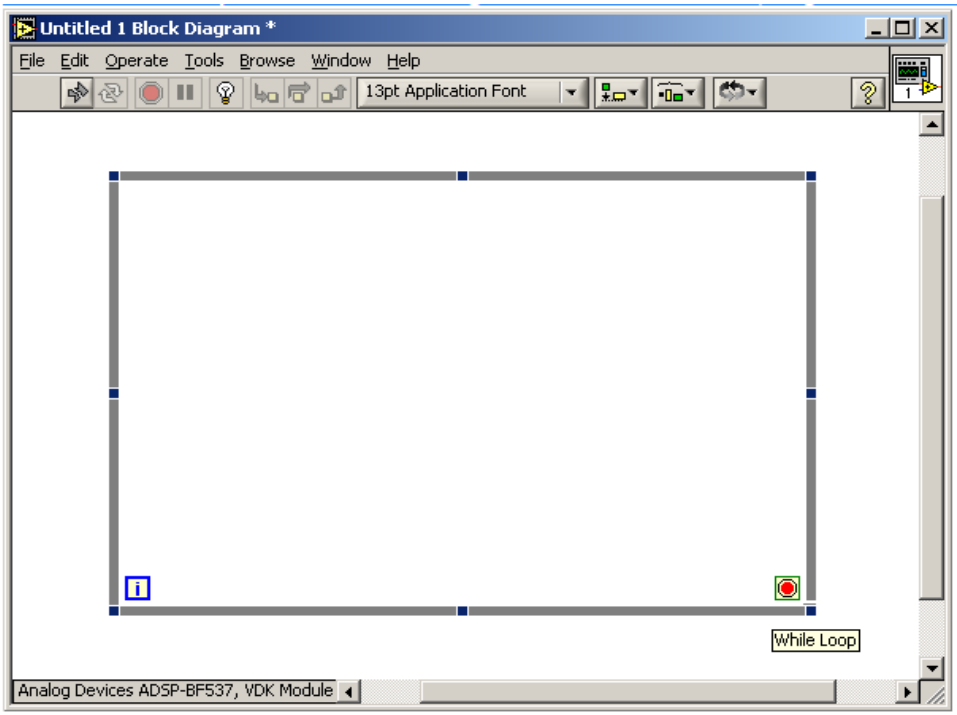
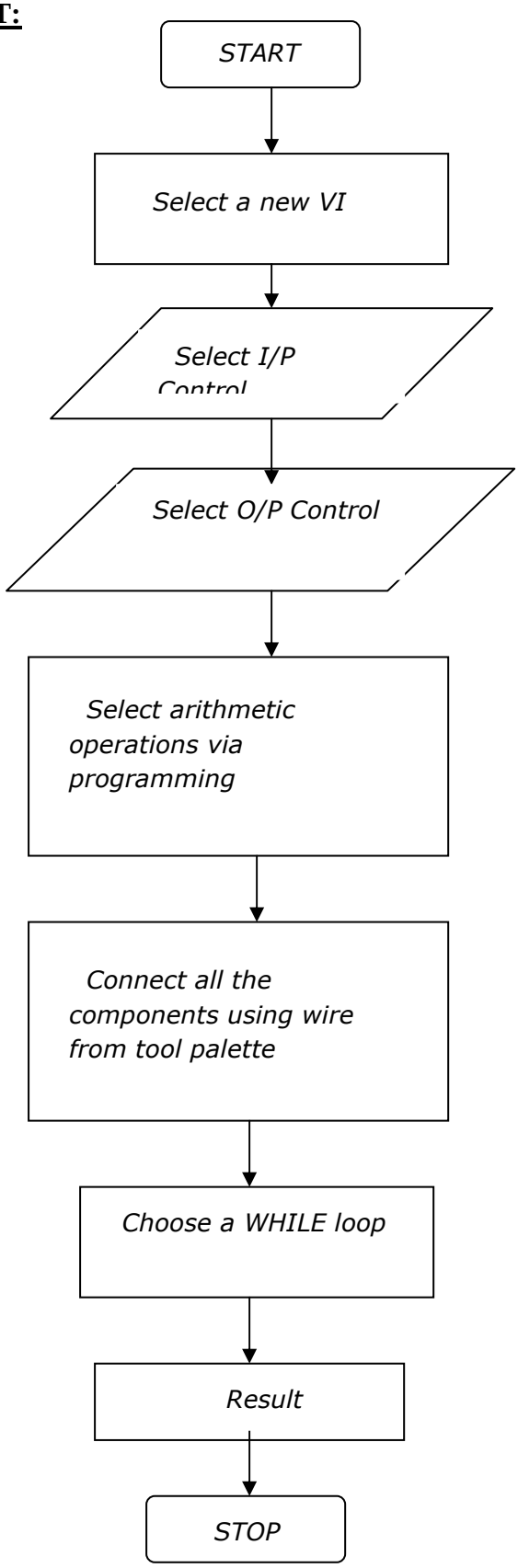


Figure 1: Snap Shot of While Loop

FLOW CHART:


START
Select a new VI
Select I/P Control
Select O/P Control
Select arithmetic operations via programming
Connect all the components using wire from tool palette
Choose a WHILE loop
Result
STOP

ALGORITHM:

STEP1: open lab view i.e. Start.
STEP2: select a new VI in lab view.
STEP3: select and place all the input controls on the block diagram.
STEP4: select and place all the output controls.
STEP5: select the required arithmetic operations for design.
STEP6: connect all the suitable components using wire from the tool palette.
STEP7: choose a while loop and pace all the components on it.
STEP8: obtain the results.
STEP9: save the result.
STEP10: Exit the lab view.

PROCEDURE:

1. Choose a new vi in the lab view.
2. By right clicking on the front panel select the input control for input parameters.
3. By right clicking on the front panel select the output control for output parameters.
4. By right clicking on the block diagram select mathematical operations from all functions.
5. Connect the components of circuit by selecting the wire from the palate.
6. By right clicking on the block diagram, choose a while loop for the entire block diagram to have a continuous monitoring of temperature.
7. Monitor the process variable in the waveform chart so it reaches set point value.

RESULT: The temperature is monitored for the given values of Q, M and S using LabVIEW simulation.

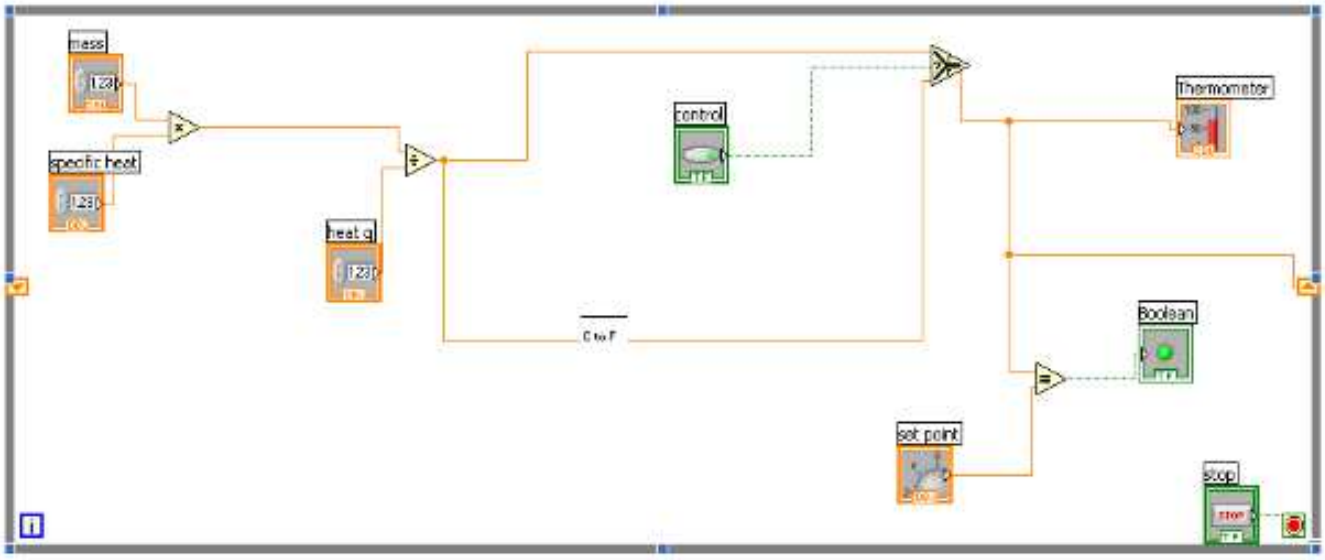


Figure 2: Block Diagram of Temperature Monitoring System

EXPERIMENT NO: 8

DESIGN OF P, PI AND PID CONTROLLER FOR PRESSURE MONITORING SYSTEM USING LABVIEW

AIM: To design and stimulate pressure monitoring system.

APPARATUS:

- LabVIEW software.
- Desktop computer.

THEORY:

Pressure is monitored and controlled using PID controller. The tank pressure shown by the gauge is the process variable and manipulated variable is the pressure leakage or outflow

PROPORTIONAL CONTROLLER (P CONTROLLER):

In this controller mode the smooth relationship exists between control and error. Over some range of errors about the set point, every value of error will have unit value of controller output.

$P \alpha e(t)$

$P = K_P e(t) + P(0)$ ……(1)

$P(x) = 0$

Where,

K_P = proportional gain

e (t) =error signal

p (0) =proportional output at t=0

Applying LT to eqn. 1

$P(s) = K_P E(s)$

Transfer Function = $P(s)/E(s) = K_P$.

Proportional band:

The range of errors to cover from 0-100% of controller output is known proportional band.

$P.B \alpha 1/K_P$

$P.B = 100/K_P$

INTEGRAL CONTROLLER:

It is also extension of floating controller unlike in floating controller output is not constant but it is proportional to error signal. For floating controller.

$dp/dt = \pm Ri$

But in integral controller the rate of change of integral output is proportional to e(t) or error signal.

$dp/dt \alpha e(t)$

$dp/dt = K_i e(t)$

$P = K_i \int e(t) + Pi(0)$

where,

$K_i = 1/T_i$ = integral gain.

$P = (1/T_i) * \int e(t)dt + Pi(0)$

Apply Laplace transform.

$P(s) = (1/T_i(s)) * E(s)$

$P(s)/E(s) = 1/T_i(s)$

DERIVATIVE CONTROLLER:

In this mode the controller output is promotional with the change of error, this mode is also known as rate controller mode.

Pα $de(t)/dt$

P=K_d $de(t)/dt$

Where

K_d = derivative gain=$1/T_d$

Applying L.T

P = K_d s E(S)

P(s)/E(s) = K_d s

P+I+D:

It is a combination of proportional controller, integral controller and derivative controller.

Pα [e (t) +1/Ti $\int e(t)dt$ + [Kd $\frac{de(t)}{dt}$]

P=K_P e (t) +(K_p/Ti) $*$ $\int e(t).dt + Kp\text{Kd}\frac{de(t)}{dt} + Pi(0)$

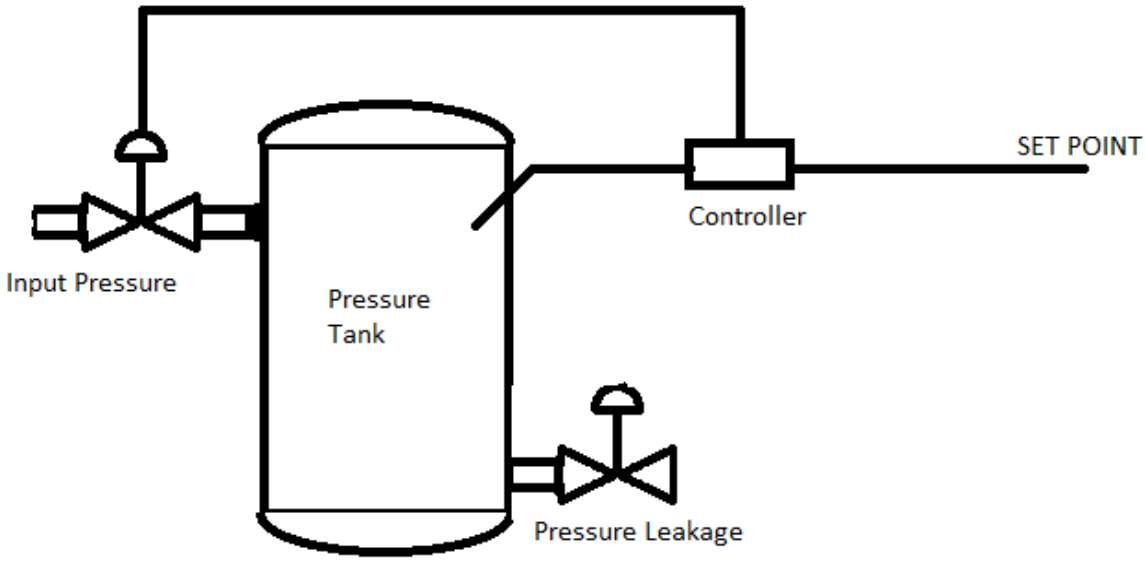
Applying LT to the above equation,

P =K_P E(s) +(K_p/T_i s)*E(s) +K_p(s) K_d sE(s)

P (s) =K_P [1+ (1/Ti s + s/Td)])] E(s)

P (s)/E(s) =K_P [1+ ($\frac{1}{Ti\ s}$)+ $\frac{s}{Td}$] E(s)

SCHEMATIC DIAGRAM:

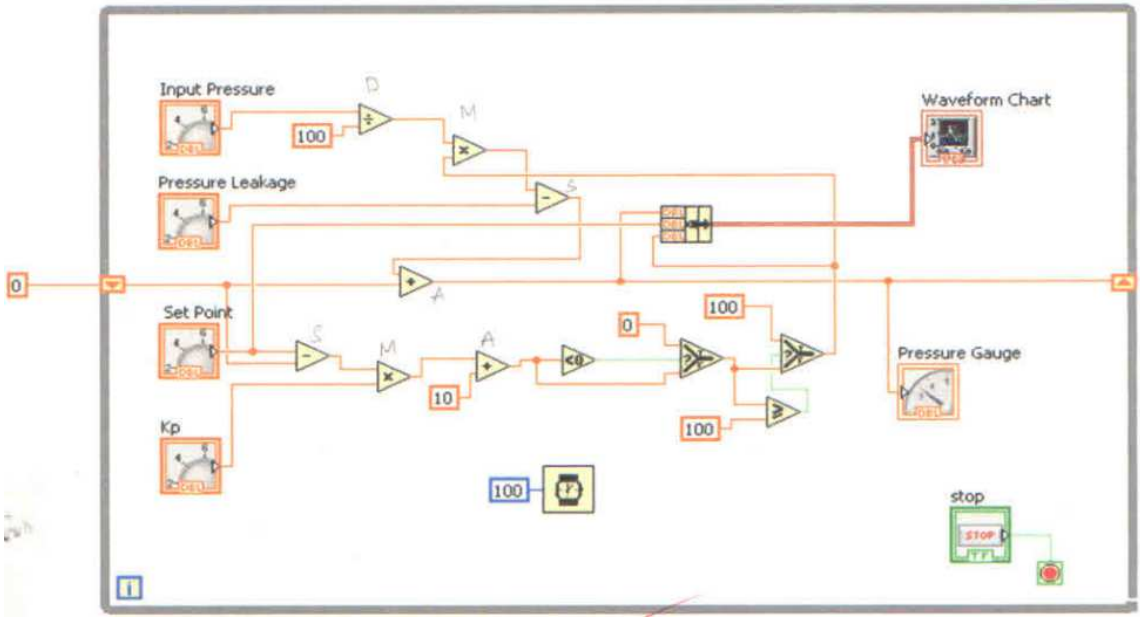


PROCEDURE:

1. Choose a new VI in the lab view.
2. By right clicking on the front panel select the input control for input parameters for proportional gain, set point, pressure gauge, wave chart.
3. By right clicking on the front panel select the output control for output parameters.
4. By right clicking on the block diagram select mathematical operations from all functions.

5. Connect the components of circuit by selecting the wire from the palette.
6. By right clicking on the block diagram, choose a while loop for the entire block diagram to have a continuous monitoring of pressure in the vessel.
7. Monitor the process variable in the waveform chart so it reaches set point value.

RESULT: The simulation of pressure monitoring system is studied and also implementation of P, PI and PID controller is done which is verified with different values of PV and SPs.



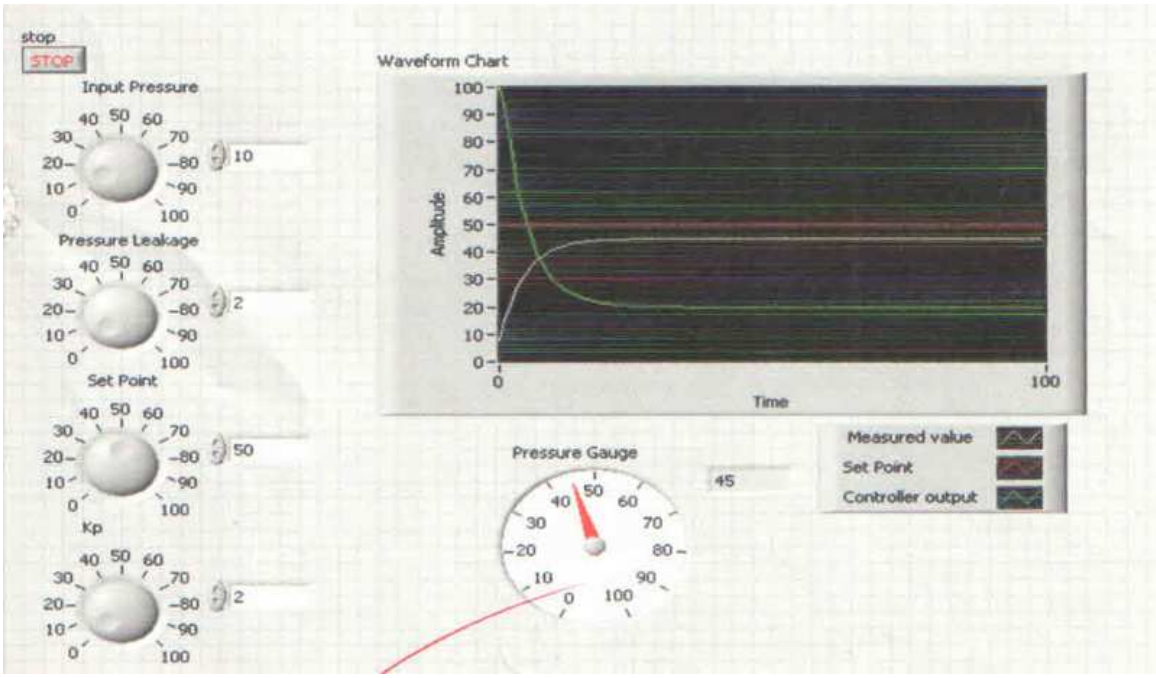


Figure 2: Front Panel of Pressure monitoring System in closed loop with only P effect.

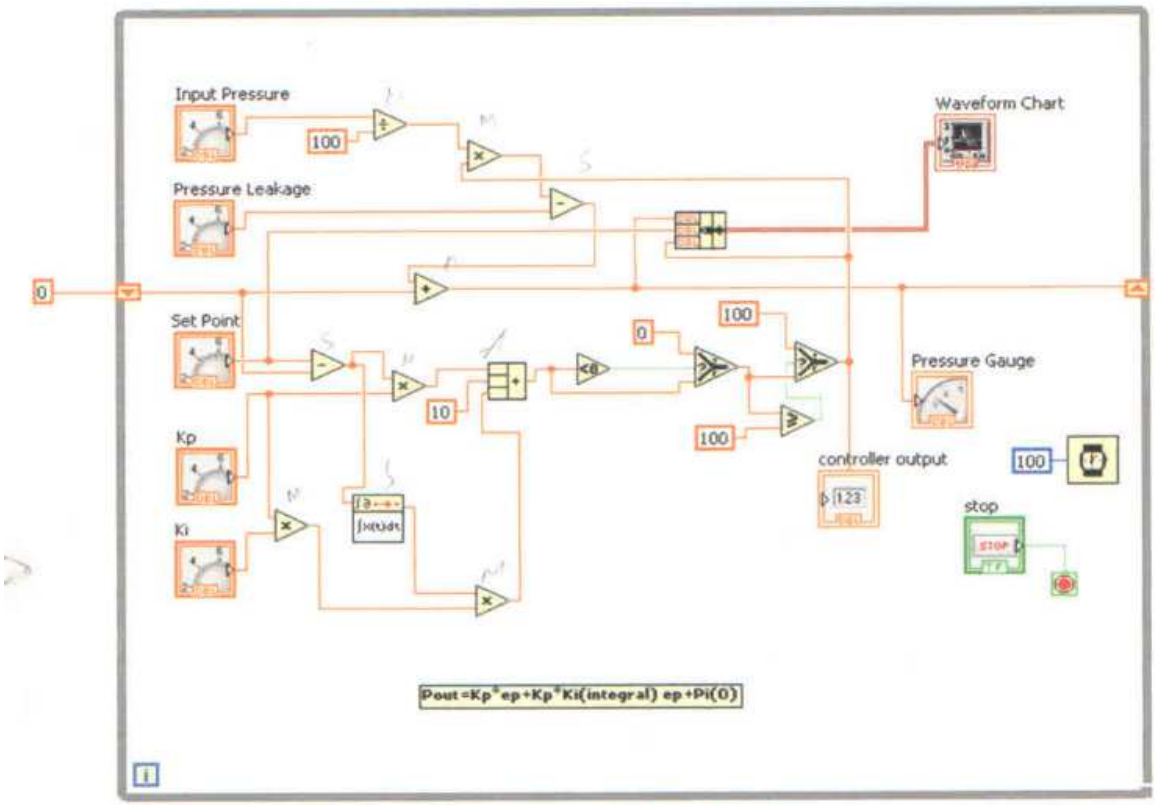


Figure 3: Block diagram of Pressure monitoring System in closed loop with only PI effect.

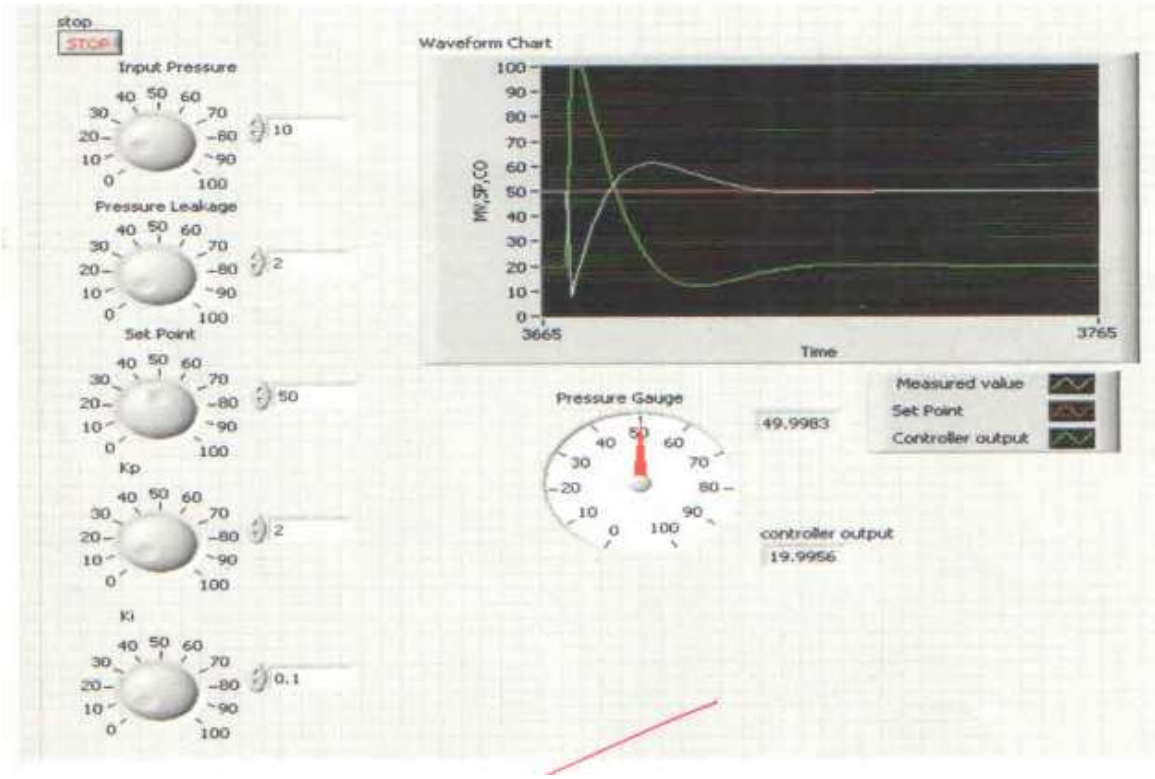


Figure 4: Front Panel of Pressure monitoring System in closed loop with only PI effect.

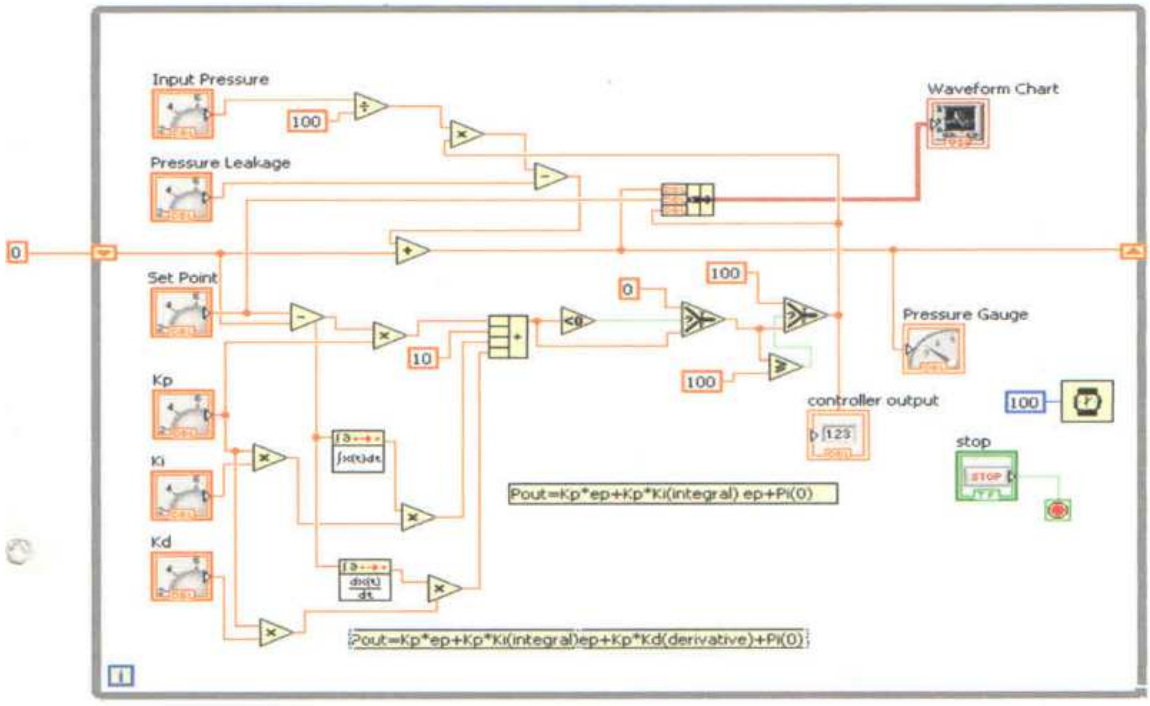


Figure 5: Block diagram of Pressure monitoring System in closed loop with PID effect.

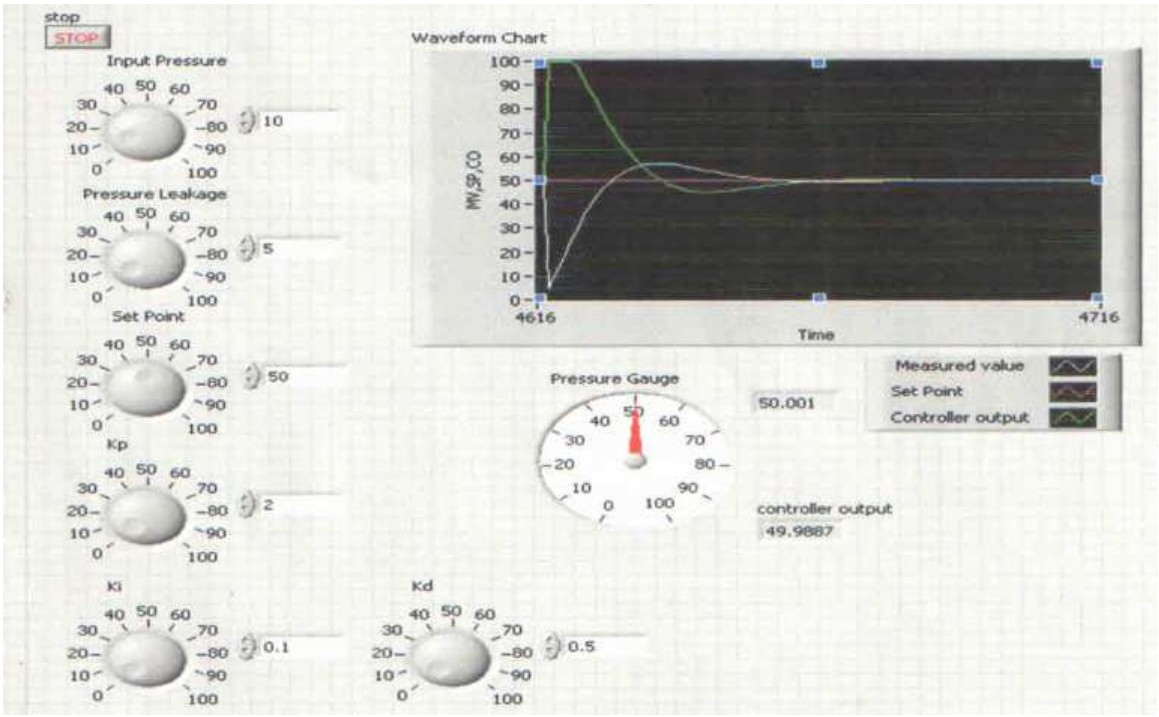


Figure 6: Front Panel of Pressure monitoring System in closed loop with PID effect.

EXPERIMENT NO: 9

DESIGN OF PID CONTROLLER FOR LEVEL MONITORING SYSTEM USING LABVIEW

AIM: To design and stimulate tank level monitoring system.

APPARATUS:

- LabVIEW software.
- Desktop computer.

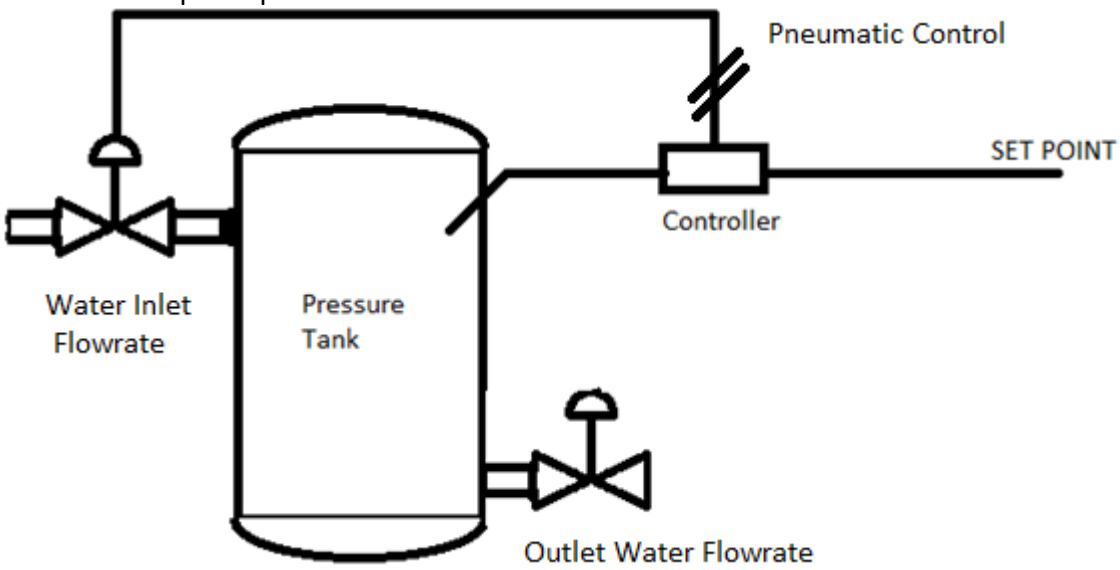


Figure 4: **Schematic Diagram of LEVEL Monitoring System.**

PROCEDURE:

1. Start the computer and double click on the lab view software.
2. Now choose the blank VI and start arranging the components using level monitoring system
3. Open the front panel and place the components like adder, subtractor.
4. By left clicking on adder take the control and also the constant and indicator.
5. Join the components of block diagram and arrange in such a way that all other components should be within the loop.
6. In the front panel, the value of controller, the set pointer, inputs are changed.
7. Observe the waveform graph and change the values and observe for different values.

OBSERVATIONS:

Controller	Inflow Rate (m^3/s)	Outflow Rate (m^3/s)	Set Point (SP)	Process Variable (PV)	Kp	Ki	Kd	Offset error ($\frac{SP-PV}{SP}$)	Controller Speed or Settling time(sec)
P									
I									
D									
PID									

RESULT: The simulation of LEVEL monitoring system is studied and also implementation of PID controller is done which is verified with different values of PV and SPs.

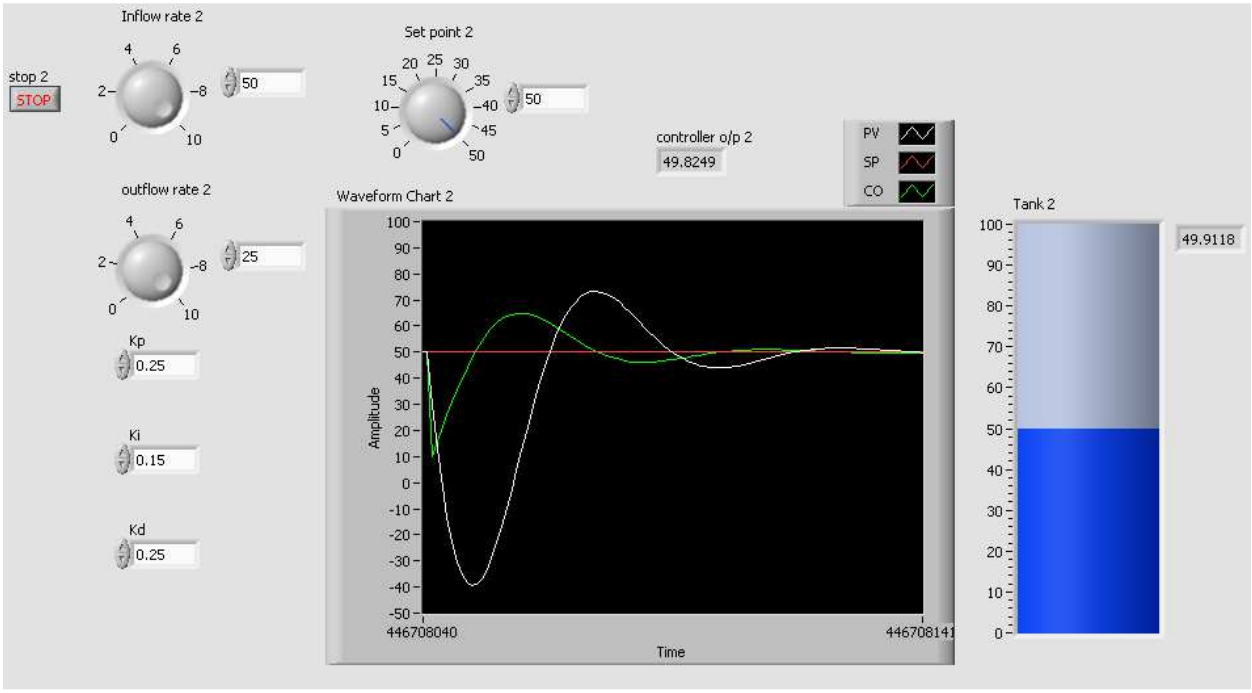


Figure 5: Front Panel of Tank Level Control System

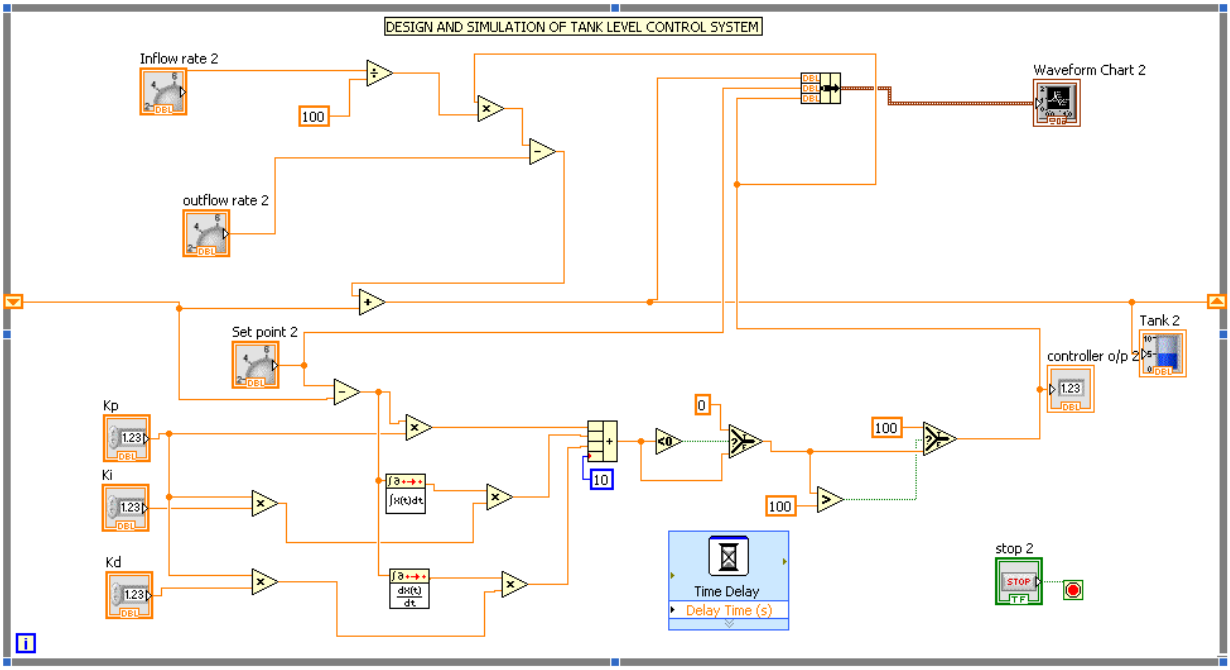


Figure 6: Block diagram of Tank Level Control System

EXPERIMENT NO: 10

ANALYSIS OF ECG WAVEFORM USING LABVIEW

AIM: To perform Analysis on ECG Waveform.
APPARATUS: MATLAB and LabVIEW Software and PC.
THEORY: ECG (Electro Cardio Gram) is trans thoracic interpretation of the electrical activity of the heart over a period of time as detected by the electrodes attached to the outer surface of the skin and recorded by a device external to the body. The recording produced by the non invasive procedure is termed as electro cardiogram. ECG is used to measure rate and regularity of heart as well as size, position of chamber and presence of damage to heart etc...

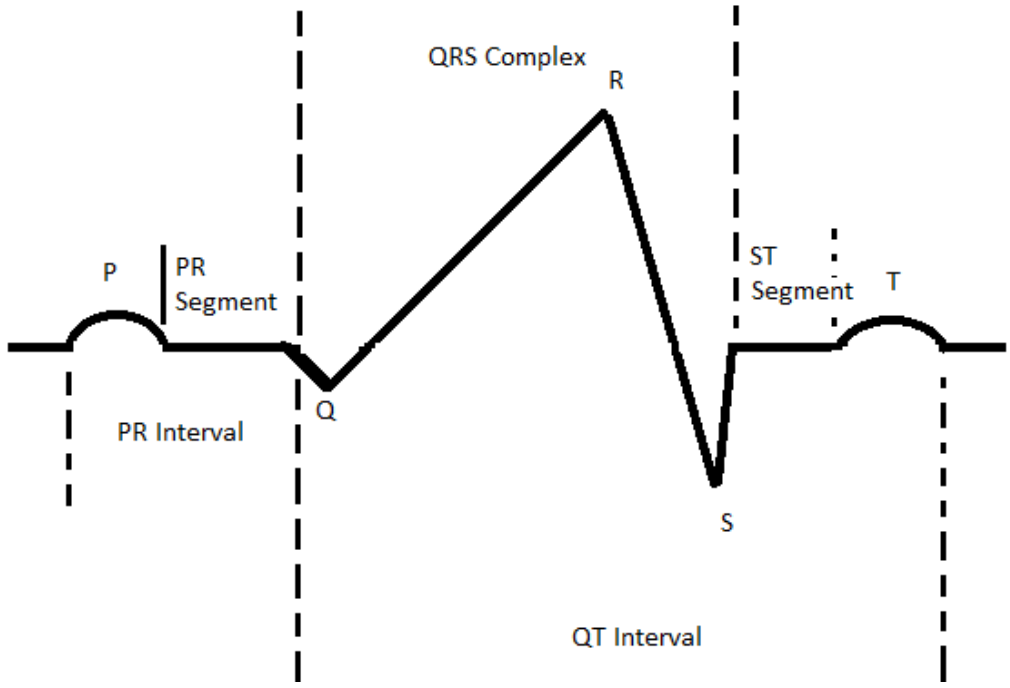


Figure 7: Schematic Representation of Normal ECG

WAVES AND INTERVAL:

FEATURES	DESCRIPTION	DURATION
RR INTERVAL	The interval between the R-wave is the R wave normal resting heart rate is between 60and 100bpm.	0.6 to 1.2 sec.
PR SEGMENT	The PR segment connects the P-wave and QRS Complex. This co-incidence with the electrical conduction from the AV node to bundle of His to the branches and then to the fiber .This shows up flat on the ECG. The PR wave interval is more clinically relevant.	50 to 120 msec.
QRS COMPLEX	This reflects the rapid depolarization of the right and left ventricle. They have a larger muscle most compared to the atria and so the QRS complex usually has a muscle much larger amplitude than the P wave.	80 to 120msec.
P WAVE	During normal atria depolarization the main electrical vector is directed from the SA Node towards the AV node and speeds from the right atrium. This turns into the P-wave on the ECG.	80sec
PR Interval	The PR Interval is measured from the beginning of the P-wave to the beginning of the QRS Complex. The PR wave reflects the time electrical impulse takes to travel from the SA node to the AV node and entering the ventricles.	120 to 200msec.

T wave	This interval represents the re polarization of the ventricles, the interval from the beginning of the QRS complex to the apex of the T Wave is reflected to as the absolute re factory period. The last tray of the T wave is reflected to as the relative re factory period.	160msec.
ST segment	The ST Segment connects the QRS complex to the T wave. The ST represents the period when the ventricles are depolarised is as localised.	8 to 120ms.
P wave	QRS complex usually has a muscle much larger than the p wave.	80msec.
ST wave	The ST Segment connects the QRS complex to the T wave. The ST represents the period when the ventricles are depolarised is as localised.	80 to 120msec.

PROCEDURE:

1. Open MATLAB software.
2. ECG is inbuilt in MATLAB. Write the command, t =ECG (20); in the command window. Use command plot(t) to see ECG in MATLAB.
3. Open the workspace; see the values of t in it.
4. Copy it into an excel sheet, close MATLAB.
5. Open lab view, open new VI.
6. Right click on the front panel and select array.
7. By right clicking on the front panel and select numeric control and place it inside the array.
8. Extend the array to a value equal to the MATLAB.
9. Copy all the values in the array from excel sheet.
10. Right click on the front panel, select waveform graph.
11. Connect array and waveform graph in the block diagram.
12. Run the program and observe the graph.

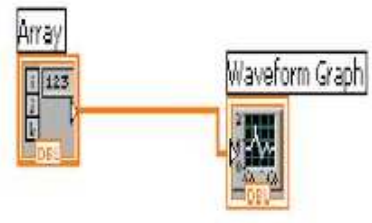


Figure 8: Block diagram for ECG waveform Analysis

OBSERVATION:

P wave: ……ms
PR wave: ……ms
PR segment: ……ms
QRS segment: ……ms
ST segment: ……ms
T wave……ms
Result: The simulation of ECG signal using LabVIEW is shown.

ECG FRONT PANEL:

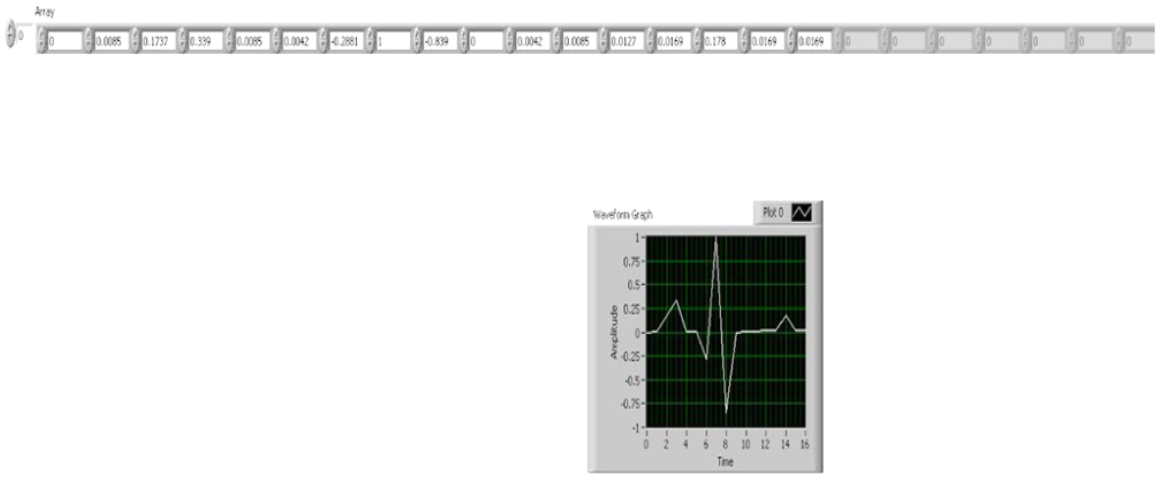


Figure 9: Front Panel for ECG waveform Analysis