MADHA ENGINEERING COLLEGE

(A Christian Minority Institution)

KUNDRATHUR, CHENNAI – 600 069

MADHA

Expertise | Empathy | Excellence

ENGINEERING COLLEGE

Control and Instrumentation Lab Manual

Name :

Subject :

Roll No. :

Semester : Year:

EXP 1: Design and simulation of mathematical model for a linear mechanical system

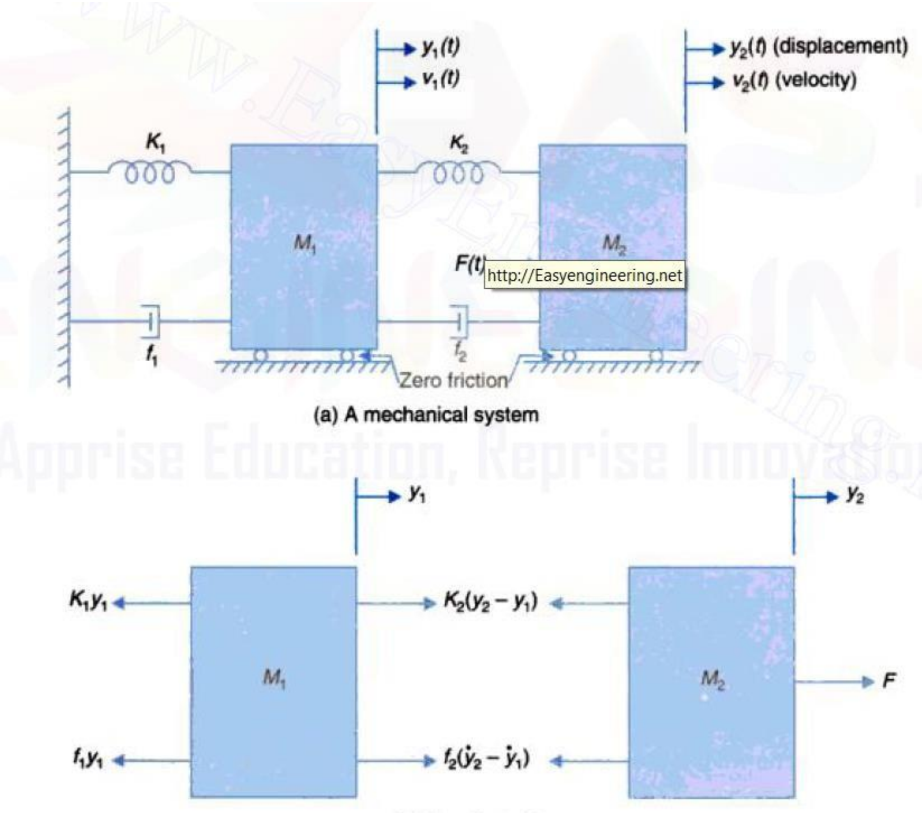
AIM:

To design and simulate a mathematical model for a linear mechanical system.

APPARATUS:

S.no	Software	Version
1.	MATLAB	R2021
2.	Simulink	-

FORMULA:



(a) A mechanical system

(b) Free-body diagram

The corresponding differential equations are as follows:

$$M_2\frac{d^2}{dt^2}(y_2)+f_2\,(\frac{d}{dt}(y_2)-\frac{d}{dt}(y_1))+K_2(y_2-y_1)=F(t)$$

$$M_1\frac{d^2}{dt^2}(y_1)-f_2\,(\frac{d}{dt}(y_2)-\frac{d}{dt}(y_1))-K_2(y_2-y_1)+f_1\,(\frac{d}{dt}(y_1))+K_1y_1=0$$

Consider the parameters as given below:

M1 = M2 = 5 kg

K1 = 2 N/m

K2 = 6 N/m

f1 = 2N/m/s

f2 = 3N/m/s

The transfer function in s domain for the corresponding differential equation is:

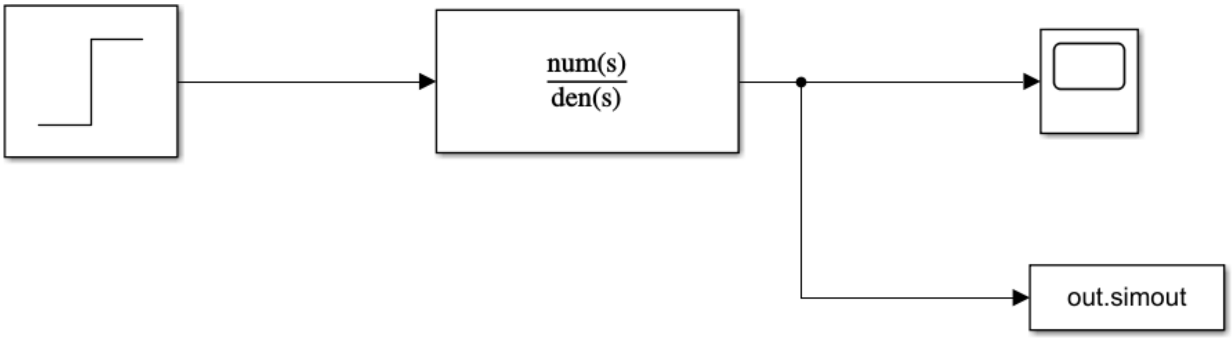
$$\frac{y_2(s)}{f(s)}=\frac{5s^2+5s+11}{25s^4+40s^3+91s^2+27s+30}$$

BLOCK DIAGRAM/CODE:

1. Code:

```
>>num=[5 5 11];

>>den=[25 40 91 27 30];

>>G=tf(num,den);

>>kp=dcgain(G);

>>ess=1/(1+kp);

>>step(G);
```

2. Simulink model:



OBSERVATIONS:

1. Code:

```
num =
    5     5    11
den =
   25    40    91    27    30

G =


            5 s^2 + 5 s + 11
  ------‐-.........-......-.........-.........-.......
  25 s^4 + 40 s^3 + 91 s^2 + 27 s + 30

Continuous-time transfer function.
kp =
   0.3667

ess =
   0.7317
```

2. Simulink:

MEASUREMENTS	VALUES
Overshoot	60.484%
Preshoot	0.806
Undershoot	5.563
Rise Time	1.8585 seconds
Fall Time	1.848 seconds
Settling Time	45.5 seconds
Final Value	0.347
Steady state Error	0.7317

INFERENCE:

- The differential equations for a linear mechanical system were established and it was realized using Simulink.
- The corresponding transfer function graph was plotted for step input, in Simulink and MATLAB, and the properties were analyzed.
- The damping effect of the spring system was established, and the corresponding data was tabulated.

RESULT:

Thus, the mathematical model of the linear mechanical system was simulated, and the corresponding parameters were noted.

EXP 2: Design and simulation of mathematical model for a non-linear mechanical (Ball and Beam) system

AIM:

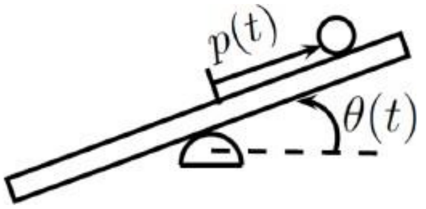
To design and simulate a mathematical model for a non-linear (Ball and Beam) mechanical system.

APPARATUS:

S.no	Software	Version
1.	MATLAB	R2021
2.	Simulink	-

FORMULA/DERIVATION:

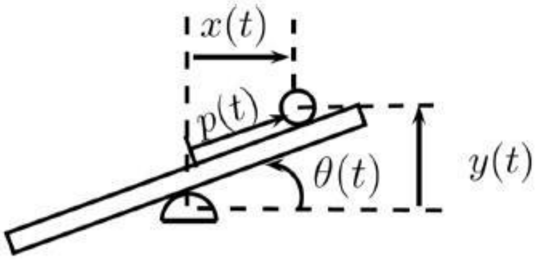
The Lagrange method is an energy-based approach for deriving the equations of motion of a dynamical system. This is convenient since it does not require the use of vectors. We will now derive the equations of motions for the Ball and Beam using this method. Consider the Ball and Beam system:



The ball rolls on the beam without slipping under the action of the force of gravity. The beam is tilted from an external torque to control the position of the ball on the beam. We first define a set of generalized coordinates which fully describe the system. Here, p(t) is the position of the ball and $\theta(t)$ is the angle of the beam. The lagrangian is described as:

L = K - U

where K is the kinetic energy and U is the potential energy of the system.



The kinetic energy of the beam is given by:

$$K_1 = \frac{1}{2} J\theta'^2$$

Where J is the moment of inertia of the beam. The kinetic energy of the ball is:

$$K_2 = \frac{1}{2} J_b \theta_b'^2 + \frac{1}{2} m v_b^2$$

where θ is the angular velocity of the ball and VB is the linear velocity of the ball. The quantity θb can be expressed in terms of the generalized coordinates as

$$\dot{\theta}_b = \frac{p}{r}$$

where r is the radius of the ball. We can also express vb in terms of the generalized coordinates.

$$v_b^2 = x^2 + y^2$$

$$x = p\,cos\,\theta$$

$$\frac{\mathbb{d}x}{\mathbb{d}t} = (\frac{\mathbb{d}p}{\mathbb{d}t})\cos\theta - p\theta\sin\theta$$

$$(\frac{\mathbb{d}x}{\mathbb{d}t})^2 = (\frac{\mathbb{d}p}{\mathbb{d}t})^2\cos^2\theta - 2p\,(\frac{\mathbb{d}p}{\mathbb{d}t})\,\theta\cos\theta\sin\theta + p^2\theta^2\sin^2\theta$$

$$y = p\,sin\,\theta$$

$$(\frac{\mathbb{d}y}{\mathbb{d}t}) = (\frac{\mathbb{d}p}{\mathbb{d}t})\,sin\,\theta + p\theta\,cos\,\theta$$

$$(\frac{dy}{dt})^2 = (\frac{dp}{dt})^2\sin^2\theta + 2p\,(\frac{dp}{dt})\cos\theta\sin\theta + p^2\theta^2\cos^2\theta$$

$$v_b^2 = (\frac{dp}{dt})^2 + p^2\theta^2$$

Substituting for kinetic energy,

$$k_2 = \frac{1}{2}(\frac{J_B}{r^2} + m)\,p^2 + \frac{1}{2}mp^2A^2$$

The potential energy is given by:

$$U = mgpsin\theta$$

The resultant Lagrangian for this system is,

$$L = \frac{1}{2}(\frac{J_b}{r^2} + m)\, p\ ^2 + \frac{1}{2} mp^2\, A^2 - mgpsin\theta$$

The first Lagrange is given by:

$$\frac{d}{dt}(\frac{\partial L}{\partial p'}) - \frac{\partial L}{\partial p} = U$$

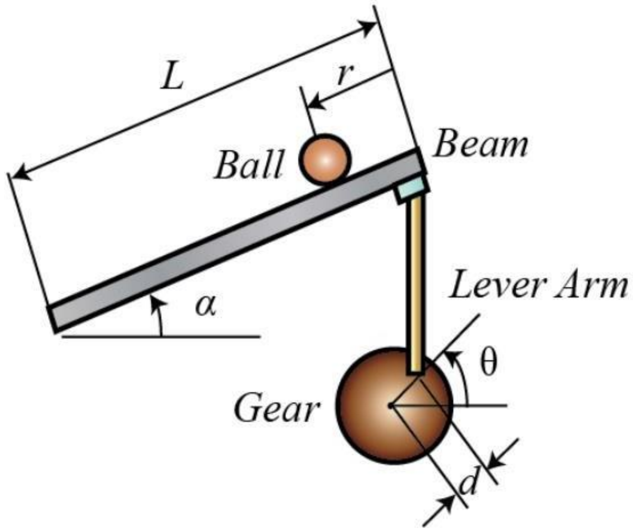
$$\frac{\partial L}{\partial t} = (\frac{J_B}{r^2} + m)\, p''$$

$$\frac{\partial L}{\partial p} = mp\theta - mgsin\theta$$

Thus, the equation of the ball beam system is given by:

$$\boldsymbol{(\frac{J_B}{r^2} + m)\, p'' + mgsin\theta - mp\theta^2 = 0}$$

BLOCK DIAGRAM/CODE:



Here the lever is connected to a servo gear. Thus, as the servo angle changes by theta, the lever changes by an angle alpha.

A controller will be designed for this system so that the ball's position can be manipulated.

$$(\frac{J_b}{R^2} + m)\, r'' + mgsin\alpha - mr\alpha'^2 = 0$$

$$\alpha = \frac{d}{L}\theta$$

Linearization of this equation about the beam angle, alpha = 0, gives us the following linear approximation of the system:

$$(\frac{J}{R^2} + m)\, r'' = -\frac{mgd}{L}\theta$$

- r - Distance of the ball from the center.
- α – angle made by the beam with respect to the horizontal
- R – radius of the ball
- J_B – Moment of inertia of the ball

1. MATLAB initialization parameters and Code:

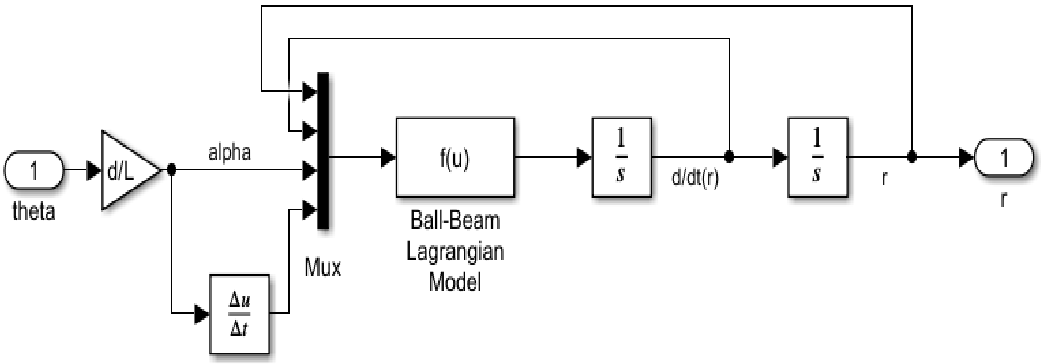
Parameter initialization in MATLAB:

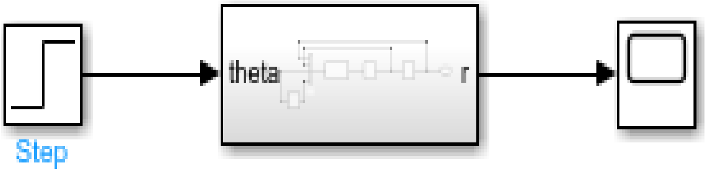
a) m=0.1 kg.
b) R=0.01 m (radius of the ball).
c) g=-9.8 m/s^2
d) L=1 m (length of beam).
e) d=0.01 m (level arm offset).
f) J=4 * 10^{-6} kg/m^2(Moment of inertia of the ball).

Code:

```
>>m = 0.1;
>>R = 0.01;
>>g = -9.8;
>>L = 1;
>>d = 0.01;
>>J = 4e-6;

>>K = (m*g*d)/(L*(J/R^2+m)); %simplifies input

>>num = [-K];
>>den = [1 0 0];
>>step (0.25*num, den);
```

2. Simulink model:





OBSERVATIONS:

INFERENCE:

- We can observe that as θ increases, beam becomes steeper and as a result, the system becomes highly unstable;
- In addition to the existing model, we can design an external controller which helps in maintaining the θ such that the stability of the system is maintained.
- A lead compensator can be designed which prevents the ball from rolling off the beam and hence making the system stable.

RESULT:

Thus, for the given ball beam system, an equation was obtained, and the corresponding Simulink model and MATLAB code was designed. For a step input, the resultant graph was plotted and analyzed.

EXP 3: Design and simulation of mathematical model for Electrical system

1. Series RLC circuit

AIM:

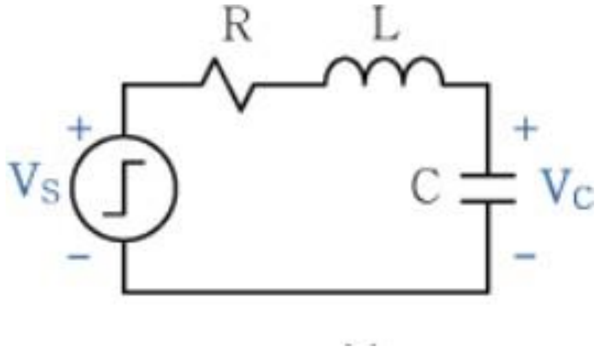
To design and simulate a mathematical model for a series RLC electrical circuit and analyze the response graph.

APPARATUS:

S.no	Software	Version
1.	MATLAB	R2021
2.	Simulink	-

FORMULA/DERIVATION:

In a series RLC circuit, the resistor acts as the dissipating element, while the conductor and inductor act as energy storing elements. We utilize the transfer function, and by giving the input as a unit step signal, we can model the system and obtain the respective graph.



Here, let:

- L = inductance (H)
- R = resistance (R)
- C = capacitance(F)

- V_s = Unit step input
- I(t) = current through the circuit(clockwise)(A)
- q = charge

The current is moving in a clockwise manner. Using Kirchhoff's voltage law:

$$V_s(t) - \frac{Ldi(t)}{dt} - i(t)R - \frac{1}{C}\int i(t)dt \ = 0$$

Here we know that $i(t) = \frac{dq(t)}{dt}$,

$$V(t)_s = \frac{Ld}{dt}(\frac{dq(t)}{dt}) + \frac{Rdq(t)}{dt} + \frac{1}{C}\int dq$$

$$V_s(t) = \frac{Ld^2q(t)}{dt^2} + \frac{Rdq}{dt} + \frac{q}{C}$$

$$\frac{d^2q(t)}{dt^2}L = R\,(\frac{dq(t)}{dt}) + \frac{1}{C}q(t) = V_s\,(t)$$

Taking Laplace transform, assuming V_s = u(t) [unit step response];

$$Ls^2Q(s) + RsQ(s) + \frac{1}{C}Q(s) = V_s(s)$$

$$\frac{Q(s)}{V_s(s)} = \frac{(1/L)}{s^2 + \frac{Rs}{L} + \frac{1}{LC}}$$

BLOCK DIAGRAM/CODE:

1. MATLAB initialization parameters and Code:

Parameter initialization in MATLAB:

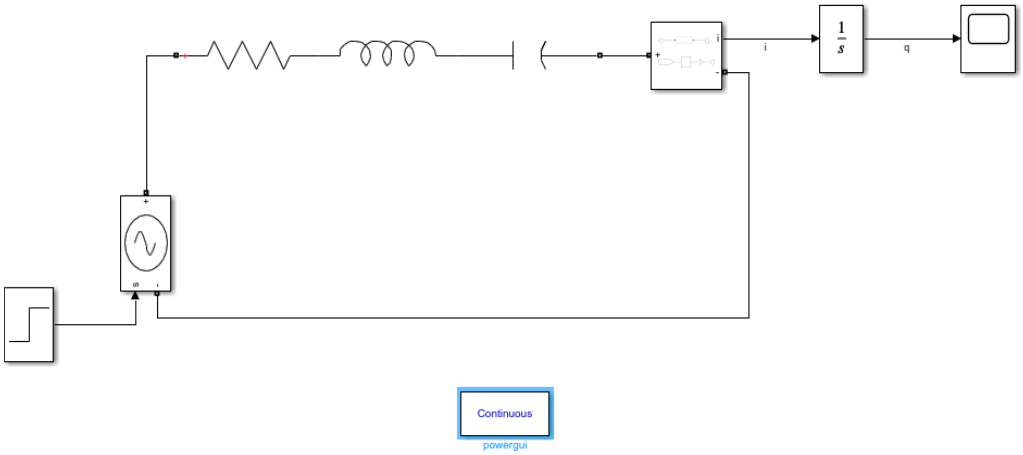
Parameters:

- R=3 Ohm
- L=1 H
- C= 0.05 F

Code:

```
>>R=3;
>>L=1;
>>C=0.05;

>>num=[1./L];
>>den=[1  (R/L)  1./(L*C) ];
>>G=tf(num,den);
>>step(G);

>>[w,zeta,p]=damp(G); %damp() gives the frequency domain
parameters
>>tau=1./w; %time constant
```

2. Simulink model:



OBSERVATIONS:

1. **MATLAB code:**

2. **Simulink:**

MEASUREMENTS	VALUES
Overshoot	30.921%
Pre shoot	0.658%
Undershoot	9.302%
Rise Time	317.030 ms

INFERENCE:

- We know that current lags in the inductor, leads in capacitor and is in phase in resistor.
- Here, the graph is divided into transient and steady state regions, where the transient state is influenced by initial conditions(R,L,C), while the steady state component remains unchanged.
- Hence from the graphs, we can observe the underdamping of the wave, before reaching a steady state value. By changing the values of R, L and C, we can change the damping ratio and in turn the response as well.

RESULT:

Thus, for a series RLC electrical circuit, the transfer function was obtained, and the corresponding Simulink model and MATLAB code was designed. For a step input, the resultant graph was plotted and analyzed.

EXP 3: Design and simulation of mathematical model for Electrical system

2. Parallel RLC circuit

AIM:

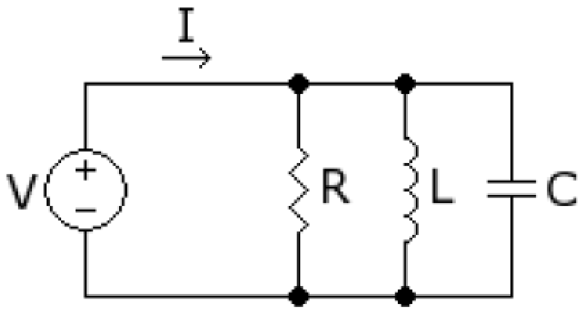
To design and simulate a mathematical model for a parallel RLC electrical circuit and analyze the response graph.

APPARATUS:

S.no	Software	Version
1.	MATLAB	R2021
2.	Simulink	-

FORMULA/DERIVATION:

In a parallel RLC circuit, the resistor acts as the dissipating element, while the conductor and inductor act as energy storing elements. We utilize the transfer function, and by giving the input as a unit step signal, we can model the system and obtain the respective graph.



Here, let:

- L = inductance(H)
- R = resistance(R)

- C = capacitance(F)
- V = input voltage(V)
- V_o=output voltage(V)
- V_R=Voltage across the resistor(V)
- V_L= Voltage across the inductor(V)
- V_C= Voltage across the capacitor(V)
- I(t)= current through the circuit(clockwise)(A)

The current is moving in a clockwise manner. Using Kirchhoff's current law:

$$\frac{V_R(t)}{R} + \frac{1}{L}\int V_L(t)dt + \frac{CdV_c(t)}{dt} = i(t)$$

Here we know that $V_0(t) = V_R(t) = V_L(t) = V_c(t)$,

$$\frac{V_O(t)}{R} + \frac{1}{L}\int V(t)dt + \frac{CdV_o(t)}{dt} = i(t)$$

Taking Laplace transform, we get,

$$\frac{V_O(s)}{R} + \frac{1}{Ls}V_o(s) + CsV_o(s) = I(s)$$

Thus, the transfer function results

$$\frac{V_o(s)}{I(s)} = \frac{RLs}{RLCs^2 + Ls + R}$$

For a unit step function,

$$V_o(s) = \frac{RL}{RLCs^2 + Ls + R}$$

BLOCK DIAGRAM/CODE:

1. MATLAB initialization parameters and Code:

Parameter initialization in MATLAB:

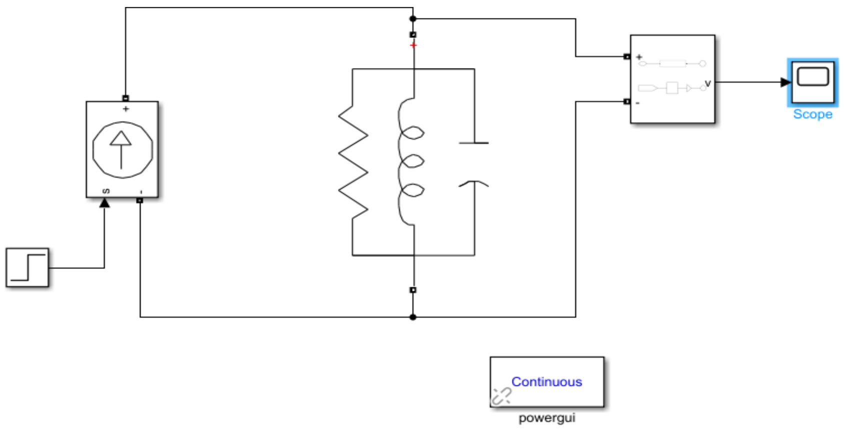
Parameters:

- R=2 Ohm
- L=1 H
- C= 0.5 F

Code:

```
>>R=2;
>>L=1
>>C=0.5

>>num=[0 (R*L) 0 ];
>>den=[(R*L*C) L R];
>>G=tf(num,den);
>>step(G);

>>[w,zeta,p]=damp(G); %damp() gives the frequency domain
parameters
>>tau=1. /w; %time constant
```

2. Simulink model:



OBSERVATIONS:

1. **MATLAB code:**

2. **Simulink:**

MEASUREMENTS	VALUES
Overshoot	6.956%
Pre shoot	0.658%
Undershoot	30.921%
Rise Time	545.708 ms

INFERENCE:

- We know that current lags in the inductor, leads in capacitor and is in phase in resistor.
- Here, we can see that a sinusoidal, gradually declining response signal is initially observed, after which it attains steady state.
- From the graph, we can observe the underdamped response of the parallel RLC circuit. By changing the values of R, L and C, we can change the damping ratio and in turn the response as well.

RESULT:

Thus, for a parallel RLC electrical circuit, the transfer function was obtained, and the corresponding Simulink model and MATLAB code was designed. For a step input, the resultant graph was plotted and analyzed.

EXP 4: System Identification through Process Reaction Curves

AIM:

To identify the given system using the process reaction curve using Simulink.

APPARATUS:

S.no	Software	Version
1.	MATLAB	R2021
2.	Simulink	-

FORMULA/THEORY:

A process reaction curve will be generated in response to a disturbance. This process curve is then used to calculate the controller gain, integral time, and derivative time. It is performed in open loop so that no control action occurs, and the process response can be isolated.

To generate a process reaction curve, the process can reach steady state or as close to steady state as possible. Then, in open loop, so that there is no control action, a small step disturbance is introduced, and the reaction of the process variable is b recorded.

For an oscillatory reaction curve, the dimensionless equation for model 1 will be adopted. To distinguish between process models that do not have a process zero (where $\bar{a}$ = 0) and those which do have a zero (where $\bar{a}$≠ 0) the following lemma is needed.

$$\frac{Y(s)}{U(s)} = G_p(s) = \begin{cases} \dfrac{K(1+as)e^{-\theta s}}{\tau^2 s^2 + 2\tau\zeta s + 1} & 0 < \zeta < 1 \quad \text{(model I)} \\ \dfrac{K(1+as)e^{-\theta s}}{(\tau s+1)(\eta\tau s+1)} & 0 < \eta \le 1 \quad \text{(model II)} \end{cases}$$

Consider a system of type Model I having $\zeta < 1$ and $\bar{a} \in (-\infty, \infty)$. Let $t_{\mathrm{p},i}$ and $t_{\mathrm{m},i}$ be the time instants when the output $y(t)$ reaches its ith peak and ith valley, respectively. Then, if and only if $\bar{a} = 0$, the following relationship holds:

$$t_{\mathrm{p},1} - \theta = t_{\mathrm{p},i+1} - t_{\mathrm{m},i} = t_{\mathrm{m},i} - t_{\mathrm{p},i} = \frac{P}{2} = \frac{\pi\tau}{\sqrt{1-\zeta^2}} \qquad \forall i \geq 1$$

where P represents the oscillating period of the response, that is,

$$P = t_{\mathrm{p},i+1} - t_{\mathrm{p},i}, \quad \forall i \geq 1$$

Consequently, model parameters for model I processes can be estimated using the following algorithm.

Algorithm 8.1: SOPDT models for oscillatory step responses

Step 1 Complete the step response test, Measure P (see Figure 8.5).

Step 2 Compute the time constant τ using

$$\tau = \frac{P\sqrt{1-\zeta^2}}{2\pi} = \frac{P}{\sqrt{4\pi^2 + P^2x^2}}$$

where

$$x = \begin{cases} \dfrac{1}{t_{\mathrm{p},1} - t_{\mathrm{p},2}} \ln\left[\dfrac{\bar{y}_{\mathrm{p},2} - 1}{\bar{y}_{\mathrm{p},1} - 1}\right] & \forall \bar{a} \geq 0 \\[2ex] \dfrac{1}{t_{\mathrm{m},1} - t_{\mathrm{p},1}} \ln\left[\dfrac{\bar{y}_{\mathrm{p},1} - 1}{1 - \bar{y}_{\mathrm{m},1}}\right] & \forall \bar{a} < 0 \end{cases}$$

and

$$\bar{y}_{\mathrm{p},i} = \bar{y}(\bar{t}_{\mathrm{p},i}) \quad i = 1, 2, \ldots$$

Step 3 Compute the damping ratio ζ using

$$\zeta = \begin{cases} \sqrt{\dfrac{\ln^2(\bar{y}_{\mathrm{p},1} - 1)}{\pi^2 + \ln^2(\bar{y}_{\mathrm{p},1} - 1)}} & \text{for } \bar{a} = 0 \\[2ex] \dfrac{Px}{\sqrt{4\pi^2 + P^2x^2}} & \forall \bar{a} \neq 0 \end{cases}$$

Step 4 Compute $\bar{a}$ and process delay time θ using

(a) for $\bar{a}=0$, compute $\theta = t_{\mathrm{p},1} + (\tau/\zeta)\ln(\bar{y}_{\mathrm{p},1}-1)$

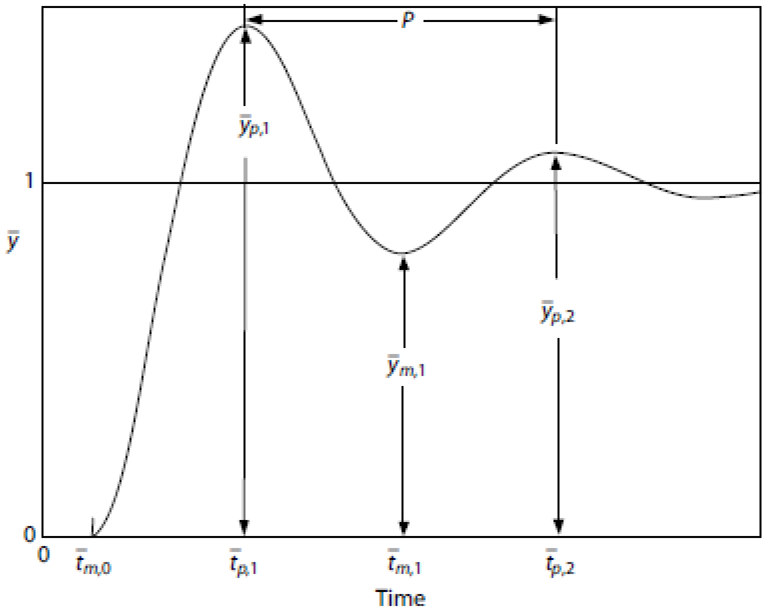
(b) for $\bar{a}\neq 0, \bar{a}$ and θ are computed using

$$\bar{a} = \begin{cases} \zeta + \sqrt{\zeta^2 + \left[1-\left(\dfrac{\bar{y}_{\mathrm{p},1}-1}{\mathrm{e}^{-\zeta \bar{t}_{\mathrm{p},1}}}\right)^2\right]} & \forall \bar{a} > 0 \\ \zeta - \sqrt{\zeta^2 + \left[1-\left(\dfrac{1-\bar{y}_{\mathrm{m},1}}{\mathrm{e}^{-\zeta \bar{t}_{\mathrm{m},1}}}\right)^2\right]} & \forall \bar{a} < 0 \end{cases}$$

and

$$\theta = \begin{cases} t_{\mathrm{p},1} - \dfrac{P}{2\pi}\left[\pi - \tan^{-1}\dfrac{\bar{a}\sqrt{1-\zeta^2}}{1-\bar{a}\zeta}\right] & \forall \bar{a} > 0 \\ t_{\mathrm{m},1} + \dfrac{P}{2\pi}\left[\tan^{-1}\dfrac{\bar{a}\sqrt{1-\zeta^2}}{1-\bar{a}\zeta}\right] & \forall \bar{a} < 0 \end{cases}$$

Algorithm end

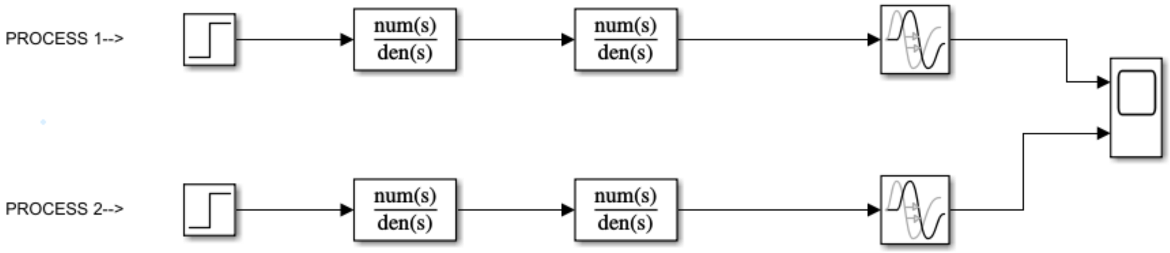


Process (a) $\quad G_{\mathrm{p}}(s) = \dfrac{\mathrm{e}^{-s}}{(4s^2+2s+1)(s^2+s+1)}$

Process (b) $\quad G_{\mathrm{p}}(s) = \dfrac{(2s+1)\mathrm{e}^{-2s}}{(4s^2+2s+1)(s^2+s+1)}$

BLOCK DIAGRAM:

1. Simulink model:



OBSERVATIONS:

1. Simulink:

Process A:

Process B:

S.no	Parameters	Process A	Process B
1.	$t_{m,0}$	2.897s	3.297s
2.	$t_{p,1}$	10.33s	8.531s
3.	$y_{p,1}$	1.206	1.385
4.	$t_{p,2}$	24.73s	22.73s
5.	$y_{p,2}$	1.005	1.01
6.	$y_{m,1}$	0.97097	0.95816
7.	p	14.4	14.3

CALCULATIONS:

1. **Process A:**

$$\chi = \frac{1}{10.33 - 24.73} ln\,(\frac{1.005 - 1}{1.206 - 1})$$

$$\boldsymbol{\chi = 0.258}$$

$$\varsigma = \sqrt{\frac{\ln(1.206 - 1)^2}{\pi^2 + (ln1.206 - 1)^2}}$$

$$\boldsymbol{\varsigma = 0.449}$$

$$r = \frac{14.4}{\sqrt{4\pi^2 + 14.4^2 * 0.258^2}}$$

$$r = \mathbf{1.9727}$$

$$\theta = 10.33 + \frac{1.9727}{0.449}\ln(1.206 - 1)$$

$$\boldsymbol{\theta = 3.39}$$

2. **Process B:**

$$\chi = \frac{1}{8.331 - 22.73}\, ln\,(\frac{1.01 - 1}{1.385 - 1})$$

$$\boldsymbol{\chi = 0.257}$$

$$\varsigma = \frac{14.199 * 0.257}{\sqrt{4\pi^2 + 14.199^2 * 0.257^2}}$$

$$\boldsymbol{\varsigma = 0.5}$$

$$r = \frac{14.199}{\sqrt{4\pi^2 + 14.199^2 * 0.257^2}}$$

$$r = \mathbf{1.954}$$

$$a' = 0.5 + \sqrt{0.5^2 + (1 - (\frac{1.385 - 1}{e^{-0.5*8.531}})^2\,)}$$

$$\boldsymbol{a' = 1.539}$$

$$\theta = 8.537 - \frac{14.199}{6.28}\,(\pi - \tan^{-1}\,(\frac{1.539\sqrt{1 - 0.5^2}}{1 - 1.539 * 0.5}))$$

$$\boldsymbol{\theta = 4.657}$$

S.no	Parameters	Process A	Process B
1.	ς	0.449	0.5
2.	χ	0.258	0.257
3.	τ	1.9727	1.954
4.	a’	0	1.539
5.	θ	3.39	4.657

Thus, the equation of process A is:

$$\frac{e^{-3.39s}}{3.88s^2 + 1.771s + 1}$$

The equation of process B is:

$$\frac{(1 + 1.539s)e^{-4.6575}}{3.818s^2 + 1.954s + 1}$$

INFERENCE:

- From the graphs, we can analyze the process curve and obtain the necessary parameters.
- The damping ratio for both the processes reinforce the fact that the curves are underdamped.
- The delay time θ is higher for process B and it is indicative for the fact that process reaction curve for process B takes longer than that of A.

RESULT:

Thus, using Simulink, it was possible to obtain the process reaction curves of the given process. The parameters were obtained from the graph and additional values were calculated as well. Thus, the resultant equations of the processes were modelled from these values as well.

EXP 5: Stability Analysis using Pole Zero maps and Routh Horwitz Criterion in simulation platform

AIM:

To observe the system stability using Routh Horwitz Criterion and pole zero maps, and to model it using MATLAB.

APPARATUS:

S.no	Software	Version
1.	MATLAB	R2021
2.	Simulink	-

THEORY:

The Routh Horwitz Criterion is an analytical procedure for determining whether the roots of a polynomial have negative real part or not. The criterion states that:

> *The necessary and sufficient condition for stability is that all the elements in the column of a Routh array be positive. If the condition is not met, the system is unstable and the number of the sign changes in the elements of the first column of the Routh array corresponds to the number of roots in the characteristic equation in the right half of the s-plane.*

The poles and zeros of the equation can be plotted on the s-plane, which is known as the pole zero map.

Thus, if all the roots are placed in the left side of s-plane, then the system is *stable*. Even if some roots are placed on the right side of the s-plane, then the system becomes *unstable*.

Here, we take 3 different equations and obtain the characteristic equations. We will model a MATLAB program and plug in the values.

CALCULATION:

1. $\boldsymbol{G(s) = \frac{25}{s^2+6s}}$

The characteristic equation is: $s^2 + 6s + 25 = 0$

s^2	1	25
s^1	6	
s^0	25	

Since there are no sign changes in the first column of the Routh array, the system is 'Stable'. Both roots lie on the left half of s-plane.

Number of Zeros: 0
Number of Roots: 2 [0, -6]

2. $\boldsymbol{G(s) = \frac{8}{s^2-4s+8}}$

The characteristic equation is: $s^2 - 4s + 16 = 0$

s^2	1	16
s^1	-4	
s^0	16	

Since there are two sign changes in the first column of the Routh array, the system is '*Unstable*' and there are two roots present in the right half of the s-plane.

Solving the denominator equation gives the poles as follows:

$$s = \frac{(4 \pm \sqrt{16 - 4(8)})}{2}$$

$$s = 2 \pm 2j$$

Number of zeros: 0
Number of Roots: 2 $(2 \pm 2j)$

3. $\boldsymbol{G(s)} = \frac{\mathbf{16}}{\boldsymbol{s^4+8s^2+16}}$

The characteristic equation is: $s^4 + 0s^3 + 8s^2 + 0s + 32 = 0$

s^4	1	8	32
s^3	0	0	0

Since there are a row of zeros, we will consider an auxiliary equation, differentiate wrt s and use those coefficient values.

$$\frac{dA}{ds} = 4s^3 + 16s = 0$$

s^4	1	8	32
s^3	1	4	
s^2	1	8	
s^1	-4		
s^0	8		

Since there are two sign changes in the first column of the Routh array, the system is '*Unstable*' and there are two roots present in the right half of the s-plane, while the others lie on the left half of s plane.
Solving the denominator equation gives:

$$s = 2j, 2j, -2j, -2j$$

Number of zeros: 0
Number of Roots: 4 $(2j, 2j, -2j, -2j)$

MATLAB CODE:

$$\boxed{\textbf{a. } \boldsymbol{G(s)} = \frac{\mathbf{25}}{\boldsymbol{s^2+6s}}}$$

```
clear;
close all;
clc;
format compact;

% Taking coefficients vector and organizing the first two rows

coeffVector = input (‘input vector of your system coefficients: \n i.e. [an an-1 an-2 ... a0] = ‘);
ceoffLength = length(coeffVector);
rhTableColumn = round(ceoffLength/2);

% Initialize Routh-Hurwitz table with empty zero array

rhTable = zeros(ceoffLength,rhTableColumn);

% Compute first row of the table

rhTable (1,: ) = coeffVector(1,1:2:ceoffLength);

% Check if length of coefficients vector is even or odd

if (rem(ceoffLength,2) ~= 0)
   % if odd, second row of table will be
   rhTable(2,1:rhTableColumn – 1) = coeffVector(1,2:2:ceoffLength);
else
   % if even, second row of table will be
   rhTable (2,: ) = coeffVector(1,2:2:ceoffLength);
end
```

```
%% Calculate Routh-Hurwitz table’s rows

% Set epss as a small value

epss = 0.01;

% Calculate other elements of the table

for I = 3:ceoffLength

   % special case: row of all zeros
   if rhTable(i-1, : )  == 0
      order = (ceoffLength – i);
      cnt1 = 0;
      cnt2 = 1;
      for j = 1: rhTableColumn – 1
         rhTable(i-1,j) = (order – cnt1) * rhTable(i-2,cnt2);
         cnt2 = cnt2 + 1;
         cnt1 = cnt1 + 2;
      end
   end

   for j = 1:rhTableColumn – 1
      % first element of upper row
      firstElemUpperRow = rhTable(i-1,1);

      % compute each element of the table
      rhTable(I,j) = ((rhTable(i-1,1) * rhTable(i-2,j+1)) - ....
         (rhTable(i-2,1) * rhTable(i-1,j+1))) / firstElemUpperRow;
   end

   % special case: zero in the first column
   if rhTable(I,1) == 0
      rhTable(I,1) = epss;
   end
end
%% Compute number of right hand side poles(unstable poles)

%  Initialize unstable poles with zero
```

```
unstablePoles = 0;

%  Check change in signs

for I = 1:ceoffLength – 1
   if sign(rhTable(I,1)) * sign(rhTable(i+1,1)) == -1
      unstablePoles = unstablePoles + 1;
   end
end

%  Print calculated data on screen

fprintf (‘\n Routh-Hurwitz Table:\n’)
rhTable

%  Print the stability result on screen

if unstablePoles == 0
   fprintf (‘~~~~~> it is a stable system! <~~~~~\n’)
else
   fprintf (‘~~~~~> it is an unstable system! <~~~~~\n’)
end
fprintf (‘\n Number of right hand side poles =%2.0f\n’,unstablePoles)
reply = input (‘Do you want roots of system be shown? Y/N ‘, ‘s’);
if reply == ‘y’ | | reply == ‘Y’
   sysRoots = roots(coeffVector);
   fprintf (‘\n Given polynomial coefficients roots :\n’)
   sysRoots
end

H1=tf([0 0 25],[1 6 0])
pzmap(H1)
grid on
```

b. $G(s) = \frac{8}{s^2-4s+8}$

```
clear ;
close all;
clc;
format compact;

% Taking coefficients vector and organizing the first two rows

coeffVector = input('input vector of your system coefficients: \n i.e. [an an-1 an-2 ... a0] = ');
ceoffLength = length(coeffVector);
rhTableColumn = round(ceoffLength/2);

% Initialize Routh-Hurwitz table with empty zero array

rhTable = zeros(ceoffLength,rhTableColumn);

%  Compute first row of the table

rhTable(1,:) = coeffVector(1,1:2:ceoffLength);

%  Check if length of coefficients vector is even or odd

if (rem(ceoffLength,2) ~= 0)
   % if odd, second row of table will be
   rhTable(2,1:rhTableColumn - 1) = coeffVector(1,2:2:ceoffLength);
else
   % if even, second row of table will be
   rhTable(2,:) = coeffVector(1,2:2:ceoffLength);
end
%% Calculate Routh-Hurwitz table's rows

%  Set epss as a small value

epss = 0.01;

%  Calculate other elements of the table
```

```
for i = 3:ceoffLength

    %  special case: row of all zeros
    if rhTable(i-1,:) == 0
        order = (ceoffLength - i);
        cnt1 = 0;
        cnt2 = 1;
        for j = 1:rhTableColumn - 1
            rhTable(i-1,j) = (order - cnt1) * rhTable(i-2,cnt2);
            cnt2 = cnt2 + 1;
            cnt1 = cnt1 + 2;
        end
    end

    for j = 1:rhTableColumn - 1
        % first element of upper row
        firstElemUpperRow = rhTable(i-1,1);

        % compute each element of the table
        rhTable(i,j) = ((rhTable(i-1,1) * rhTable(i-2,j+1)) - ....
            (rhTable(i-2,1) * rhTable(i-1,j+1))) / firstElemUpperRow;
    end

    %  special case: zero in the first column

    if rhTable(i,1) == 0
        rhTable(i,1) = epss;
    end
end
%%  Compute number of right hand side poles(unstable poles)

%  Initialize unstable poles with zero

unstablePoles = 0;

%  Check change in signs

for i = 1:ceoffLength - 1
    if sign(rhTable(i,1)) * sign(rhTable(i+1,1)) == -1
```

```
        unstablePoles = unstablePoles + 1;
    end
end

%  Print calculated data on screen

fprintf('\n Routh-Hurwitz Table:\n')
rhTable

%  Print the stability result on screen

if unstablePoles == 0
    fprintf('~~~~~> it is a stable system! <~~~~~\n')
else
    fprintf('~~~~~> it is an unstable system! <~~~~~\n')
end
fprintf('\n Number of right hand side poles =%2.0f\n',unstablePoles)
reply = input('Do you want roots of system be shown? Y/N ', 's');
if reply == 'y' || reply == 'Y'
    sysRoots = roots(coeffVector);
    fprintf('\n Given polynomial coefficients roots :\n')
    sysRoots
end

H1=tf([0 0 8],[1 -4 8])
pzmap(H1)
grid on
```

$$\boxed{\textbf{c. } \boldsymbol{G(s) = \frac{16}{s^4+8s^2+16}}}$$

```
clear ;
close all;
clc;
format compact;

% Taking coefficients vector and organizing the first two rows
```

```
coeffVector = input('input vector of your system coefficients: \n i.e. [an an-1 an-2 ... a0] = ');
ceoffLength = length(coeffVector);
rhTableColumn = round(ceoffLength/2);

% Initialize Routh-Hurwitz table with empty zero array

rhTable = zeros(ceoffLength,rhTableColumn);

%  Compute first row of the table

rhTable(1,:) = coeffVector(1,1:2:ceoffLength);

%  Check if length of coefficients vector is even or odd

if (rem(ceoffLength,2) ~= 0)
    % if odd, second row of table will be
    rhTable(2,1:rhTableColumn - 1) = coeffVector(1,2:2:ceoffLength);
else
    % if even, second row of table will be
    rhTable(2,:) = coeffVector(1,2:2:ceoffLength);
end
%% Calculate Routh-Hurwitz table's rows

%  Set epss as a small value

epss = 0.01;

%  Calculate other elements of the table

for i = 3:ceoffLength

    % special case: row of all zeros
    if rhTable(i-1,:) == 0
        order = (ceoffLength - i);
        cnt1 = 0;
        cnt2 = 1;
        for j = 1:rhTableColumn - 1
            rhTable(i-1,j) = (order - cnt1) * rhTable(i-2,cnt2);
            cnt2 = cnt2 + 1;
            cnt1 = cnt1 + 2;
        end
    end

    for j = 1:rhTableColumn - 1
        % first element of upper row
        firstElemUpperRow = rhTable(i-1,1);

        % compute each element of the table
        rhTable(i,j) = ((rhTable(i-1,1) * rhTable(i-2,j+1)) - ....
            (rhTable(i-2,1) * rhTable(i-1,j+1))) / firstElemUpperRow;
    end

    %  special case: zero in the first column
    if rhTable(i,1) == 0
        rhTable(i,1) = epss;
    end
end
%%  Compute number of right hand side poles(unstable poles)
%  Initialize unstable poles with zero

unstablePoles = 0;

%  Check change in signs

for i = 1:ceoffLength - 1
    if sign(rhTable(i,1)) * sign(rhTable(i+1,1)) == -1
        unstablePoles = unstablePoles + 1;
    end
end

%  Print calculated data on screen

fprintf('\n Routh-Hurwitz Table:\n')
rhTable

%  Print the stability result on screen

if unstablePoles == 0
    fprintf('~~~~~> it is a stable system! <~~~~~\n')
```

```
else
   fprintf('~~~~~> it is an unstable system! <~~~~~\n')
end
fprintf('\n Number of right hand side poles =%2.0f\n',unstablePoles)
reply = input('Do you want roots of system be shown? Y/N ', 's');
if reply == 'y' || reply == 'Y'
   sysRoots = roots(coeffVector);
   fprintf('\n Given polynomial coefficients roots :\n')
   sysRoots
end

H1=tf([16],[1 0 8 0  16])
pzmap(H1)
grid on
```

OBSERVATIONS:

1. **Code:**

$$\boxed{\text{a. } G(s) = \frac{25}{s^2+6s}}$$

```
input vector of your system coefficients:
 i.e. [an an-1 an-2 ... a0] = [1 6 25]

 Routh-Hurwitz Table:
rhTable =
    1    25
    6     0
   25     0
~~~~~> it is a stable system! <~~~~~

 Number of right hand side poles = 0
Do you want roots of system be shown? Y/N y

 Given polynomial coefficients roots :
sysRoots =
  -3.0000 + 4.0000i
```

```
 -3.0000 - 4.0000i

H1 =


    25
  ----- --
  s^2 + 6 s

Continuous-time transfer function.
```

$$\boxed{\textbf{b. } G(s)=\frac{8}{s^2-4s+8}}$$

```
input vector of your system coefficients:
 i.e. [an an-1 an-2 ... a0] = [1 -4 16]

 Routh-Hurwitz Table:
rhTable =
     1    16
    -4     0
    16     0
~~~~~> it is an unstable system! <~~~~~
```

```
 Number of right-hand side poles = 2
Do you want roots of system be shown? Y/N Y

 Given polynomial coefficients roots:
sysRoots =
   2.0000 + 3.4641i
   2.0000 - 3.4641i

H1 =


       8
  ------.........-
  s^2 - 4 s + 8

Continuous-time transfer function.
```

$$\boxed{\text{c. } \mathbf{G(s)} = \frac{\mathbf{16}}{\mathbf{s^4+8s^2+16}}}$$

```
input vector of your system coefficients:
 i.e. [an an-1 an-2 ... a0] = [1 0 8 0 32]
 Routh-Hurwitz Table:
rhTable =
     1     8    32
     2     0     0
     8    32     0
    -8     0     0
    32     0     0
~~~~~> it is an unstable system! <~~~~~

 Number of right hand side poles = 2
Do you want roots of system be shown? Y/N y

 Given polynomial coefficients roots :
sysRoots =
  -0.9102 + 2.1974i
  -0.9102 - 2.1974i
   0.9102 + 2.1974i
   0.9102 - 2.1974i

H1 =


        16
  ------.........---
  s^4 + 8 s^2 + 16

Continuous-time transfer function.
```

INFERENCE:

- For a stable system, the roots of the characteristic equation must have negative real parts.
- If any root of the characteristic equation has a positive real part or if there is a repeated root on the imaginary axis, then the system is unstable.
- If all the roots of the characteristic equation have negative real parts except for the presence of one or more non repeated roots on the imaginary axis, then the system is limitedly or marginally stable.
- The pole zero map also reinforces the above conditions, in a graphical manner.

RESULT:

Thus, the system stability of different systems was observed by implementing pole zero maps and Routh Horwitz criterion, using MATLAB.

EXP 6: Root Locus Based Analysis in Simulation Platform

AIM:

To analyze the Root Locus plot of a given system using MATLAB.

APPARATUS:

S.no	Software	Version
1.	MATLAB	R2021

THEORY:

The root locus technique in control system was first introduced in the year 1948 by Evans. Any physical system is represented by a transfer function in the form of

$$G(s) = k \times \frac{numerator\ of\ s}{denomerator\ of\ s}$$

We can find poles and zeros from G(s). The location of poles and zeros are crucial keeping view stability, relative stability, transient response, and error analysis. When the system is put to service stray inductance and capacitance get into the system, thus changes the location of poles and zeros.

In root locus technique in control system, we will evaluate the position of the roots, their locus of movement and associated information. This information will be used to comment upon the system performance.

Some of the advantages of root locus technique are:

1. Root locus technique in control system is easy to implement as compared to other methods.
2. With the help of root locus, we can easily predict the performance of the whole system.
3. Root locus provides the better way to indicate the parameters.

Now there are various terms related to root locus technique that we will use frequently:

1. *Characteristic Equation Related to Root Locus Technique:*

   1 + G(s)H(s) = 0 is known as characteristic equation. Now on differentiating the characteristic equation and on equating dk/ds equals to zero, we can get break away points.

2. *Break away Points:*

   Suppose two root loci which start from pole and moves in opposite direction collide with each other such that after collision they start moving in different directions in the symmetrical way. Or the breakaway points at which multiple roots of the characteristic equation 1 + G(s)H(s) = 0 occur. The value of K is maximum at the points where the branches of root loci break away. Break away points may be real, imaginary, or complex.

3. *Break in Point:*

   Condition of break in to be there on the plot is written below: Root locus must be present between two adjacent zeros on the real axis.

4. *Centre of Gravity:*

   It is also known centroid and is defined as the point on the plot from where all the asymptotes start. Mathematically, it is calculated by the difference of summation of poles and zeros in the transfer function when divided by the difference of total number of poles and total number of zeros. Centre of gravity is always real and it is denoted by σA.

$$\sigma_A = \frac{(Sum\ of\ real\ parts\ of\ poles) - (Sum\ of\ real\ parts\ of\ zeros)}{N - M}$$

   Where, N is number of poles and M is number of zeros.

5. *Asymptotes of Root Loci:*

   Asymptote originates from the center of gravity or centroid and goes to infinity at definite some angle. Asymptotes provide direction to the root locus when they depart break away points.

6. *Angle of Asymptotes:*

   Asymptotes makes some angle with the real axis and this angle can be calculated from the given formula,

$$Angle\ of\ asymptotes\ = \frac{(2p+1)\times 180}{N-M}$$

   Where, p = 0, 1, 2 ……. (N-M-1) N is the total number of poles M is the total number of zeros.

7. *Angle of Arrival or Departure:*

   We calculate angle of departure when there exist complex poles in the system. Angle of departure can be calculated as 180-{(sum of angles to a complex pole from the other poles)-(sum of angle to a complex pole from the zeros)}.

8. *Intersection of Root Locus with the Imaginary Axis:*

   To find out the point of intersection root locus with imaginary axis, we must use Routh Hurwitz criterion. First, we find the auxiliary equation then the corresponding value of K will give the value of the point of intersection.

9. *Gain Margin:*

   We define gain margin by which the design value of the gain factor can be multiplied before the system becomes unstable. Mathematically it is given by the formula

$$Gain\ margin = \frac{Value\ of\ K\ at\ the\ imaginary\ axes\ cross\ over}{Design\ value\ of\ K}$$

10. *Phase Margin:*

Phase margin can be calculated from the given formula:

$$Phase\ margin = 180 + \angle(G(jw)H(jw))$$

11. *Symmetry of Root Locus:*

Root locus is symmetric about the x axis or the real axis

Now there are two ways of determining the value of K, each way is described below.

1. *Magnitude Criteria:*

At any points on the root locus, we can apply magnitude criteria as,

$$|G(s)H(s)| = 1$$

Using this formula, we can calculate the value of K at any desired point.

2. *Using Root Locus Plot:*

The value of K at any s on the root locus is given by:

$$K = \frac{product\ of\ all\ of\ the\ vector\ lengths\ drawn\ from\ the\ poles\ of\ G(s)H(s)\ to\ s}{product\ of\ all\ of\ the\ vector\ lengths\ drawn\ from\ the\ zeros\ of\ G(s)H(s)\ to\ s}$$

Root Locus Plot:

This is also known as root locus technique in control system and is used for determining the stability of the given system. Now 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.

Now there are some results that one should remember to plot the root locus. These results are written below:

1. *Region where root locus exists:*

   After plotting all the poles and zeros on the plane, we can easily find out the region of existence of the root locus by using one simple rule which is written below, only that segment will be considered in making root locus if the total number of poles and zeros at the right hand side of the segment is odd.

2. *How to calculate the number of separate root loci:*

   Several separate root loci are equal to the total number of roots if number of roots are greater than the number of poles otherwise number of separate root loci is equal to the total number of poles if number of roots are greater than the number of zeros.

Procedure to Plot Root Locus:

Keeping all these points in mind we are able to draw the root locus plot for any kind of system. Now let us discuss the procedure of making a root locus:

- Find out all the roots and poles from the open loop transfer function and then plot them on the complex plane.

- All the root loci start from the poles where k = 0 and terminates at the zeros where K tends to infinity. The number of branches terminating at infinity equals to the difference between the number of poles & number of zeros of G(s)H(s).

- Find the region of existence of the root loci from the method described above after finding the values of M and N.

- Calculate break away points and break in points if any.

- Plot the asymptotes and centroid point on the complex plane for the root loci by calculating the slope of the asymptotes.

- Now calculate angle of departure and the intersection of root loci with imaginary axis.

- Now determine the value of K by using any one method that I have described above. By following above procedure, you can easily draw the root locus plot for any open loop transfer function.

- Calculate the gain margin.

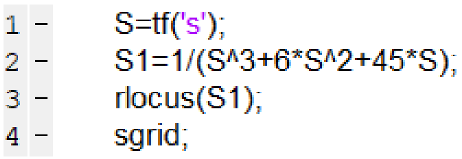
- Calculate the phase margin.

- You can easily comment on the stability of the system by using Routh Array.

MATLAB CODE:

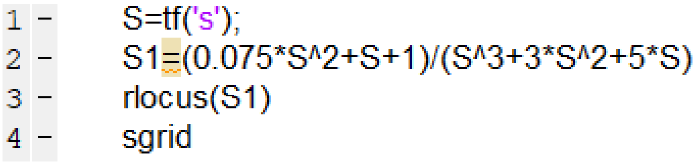
1. Code:

1. $\frac{1}{s^3+6s^2+45s}$



```
S=tf('s');
S1=1/(S^3+6*S^2+45*S);
rlocus(S1);
sgrid;
```

2. $\frac{0.075s^2+s+1}{s^3+3s^2+5s}$

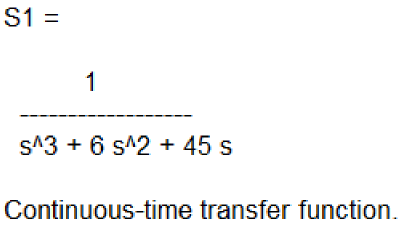


```
S=tf('s');
S1=(0.075*S^2+S+1)/(S^3+3*S^2+5*S)
rlocus(S1)
sgrid
```

OBSERVATIONS:

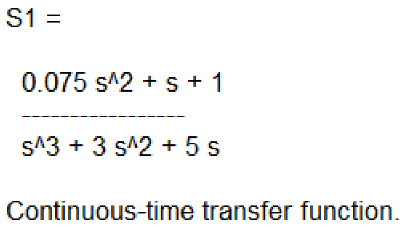
1. Code:

1. $\boxed{\frac{1}{s^3+6s^2+45s}}$



```
S1 =

          1
  ------------------
  s^3 + 6 s^2 + 45 s

Continuous-time transfer function.
```

2. $\boxed{\frac{0.075s^2+s+1}{s^3+3s^2+5s}}$



```
S1 =

  0.075 s^2 + s + 1
  -----------------
  s^3 + 3 s^2 + 5 s

Continuous-time transfer function.
```

INFERENCE:

- Note that if the denominator of G(s) and the numerator of H(s) involve common factors then the corresponding open-loop poles and zeros will cancel each other, reducing the degree of the characteristic equation by one or more.
- A slight change in the pole-zero configuration may cause significant changes in the root- locus configurations.
- The cancelled pole of G(s)H(s) is a closed-loop pole of the system and this must be added to the closed-loop poles obtained from the root-locus plot of G(s)H(s).
- General effects of the addition of zeros:
    - pull root locus to the left
    - makes system more stable
    - speed up the settling of the response

- General effects of the addition of poles:
    - pull root locus to the right
    - lower system's relative stability
    - slow down the settling of the response

RESULT:

Thus, the Root Locus analysis is done on the given systems using MATLAB, and the corresponding characteristics and graphs were obtained and studied.

EXP 7: Frequency Response Analysis of a 3rd Order System Using Bode Plot

AIM:

To analyze the frequency response of a 3rd order system from the Bode Plot.

APPARATUS:

S.no	Software	Version
1.	MATLAB	R2021

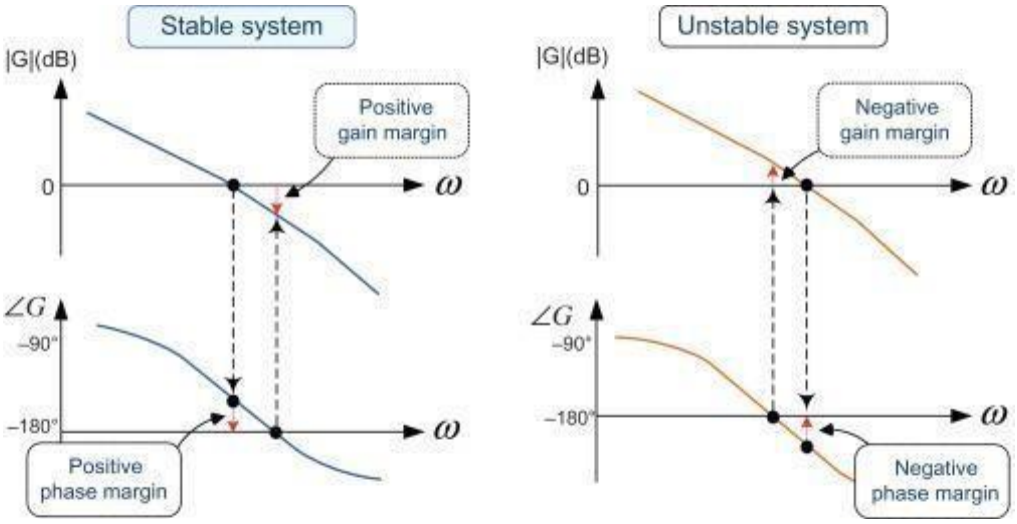
THEORY:

First, the whole system, including the controller, must be stable. Therefore, the stability is the most important requirement for the design of a control system. The response of an unstable system to a bounded input signal will continue to oscillate, but the magnitude of this oscillationwill never diminish or get larger.

The stability of linear systems can be assessed by checking the poles ofthe transfer function of the system, i.e., the roots of the characteristic equation in the *s*-plane. A linear system is stable if *all* the poles lie in theleft-half *s*-plane. However, merely knowing whether a system is stable or unstable is not sufficient. We need to find out how stable the system is.

A Bode plot is a graph commonly used in control system engineering to determine the stability of a control system. A Bode plot maps the frequency response of the system through two graphs — the Bode magnitude plot (expressingthe magnitude in decibels) and the Bode phase plot (expressing the phase shift in degrees). We can measure the relative stability of a systemby examining the *gain margin* and the *phase margin* obtained from the Bode plot of the frequency response.

Phase margin and gain margin are the measures of closed-loop control systems stability. They are illustrated in the figure below.



1. *Gain margin:*

Gain margin is defined as the amount of change in open-loop gain needed to make a closed-loop system unstable.

- The gain margin is the difference between 0 dB and thegain at the phase cross-over frequency that gives a phase of −180°. If the gain |GH(jω) | at the frequency of ∠GH(jω)=−180° is greater than 0 dB as shown in the left of Fig. 2.25, meaning a positive gain margin, then theclosed-loop system is stable.

- We can usually read the gain margin directly from the Bodeplot (as shown in the diagram above).

- This is done by calculating the vertical distance between the magnitude curve (on the Bode magnitude plot) and the x-axisat the frequency where the Bode phase plot = 180°. This point is known as the phase crossover frequency.

- It is important to realize that the Gain and the Gain Margin arenot the same things. In fact, the Gain Margin is the negative of the gain (in decibels, dB). This will make sense when we look at the Gain margin formula.

- The formula for Gain Margin (GM) can be expressed as:

$$GM = 0 - G\ dB$$

Where *G* is the gain. This is the magnitude (in dB) as read fromthe vertical axis of the magnitude plot at the phase crossover frequency.

2. *Phase margin:*

- Phase margin is defined as the amount of change in open-loop phase needed to make a closed-loop systemunstable.

- The phase margin is the difference in phase between −180° and the phase at the gain cross-over frequencythat gives a gain of 0 dB.

- We can usually read the phase margin directly from the Bode plot (as shown in the diagram above). This is doneby calculating the vertical distance between the phase curve (on the Bode phase plot) and the x-axis at the frequency where the Bode magnitude plot = 0 dB. This point is known as the gain crossover frequency.

- It is important to realize that the phase lag and the Phase Margin are not the same things. This will makesense when we look at the phase margin formula.

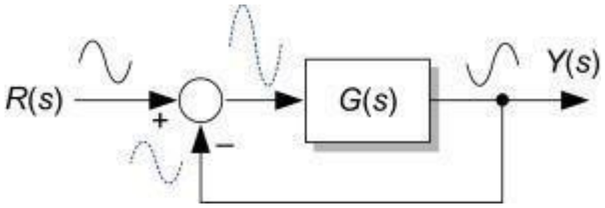
- The formula for Phase Margin (PM) can be expressed as:

$$PM = \phi - (-180^\circ)$$

- Where is the phase lag (a number less than 0). This is thephase as read from the vertical axis of the phase plot at thegain crossover frequency.

In general, as the gain of a system increases, the system becomes lessstable. The gain margin and the phase margin indicate how much the gain increases until the system becomes unstable.

Let us look at the relationship between the gain and phase margins and the stability of a system. For example, assume that the phase margin of a system is zero degree, i.e., ∠GH(jω)=180° at a frequency ω of |G(jω)|=1dB. In this case, as shown in Fig. 2.26, the phase difference between Y(s) and R(s) is 180°, and this closed-loopsystem thus becomes a positive feedback system. This system will continue to amplify its output and become unstable. Example of an unstable system is illustrated below.



If a closed-loop system is stable, both the gain margin and the phase margin need to be positive. In general, the phase margin of 30–60 degrees and the gain margin of 2–10 dB is desirable in theclosed-loop system design. A system with a large gain margin and phase margin is stable but has a sluggish response, while the one with a small gain margin and phase margin has a less sluggish response but is oscillatory.

For the graphs below the stabilities are analyzed.

A) Bode plot A

B) Bode plot B

TABULATION:

s.no	Parameters	Bode Plot A	Bode Plot B
1.			
2.			
3.			
4.			

INFERENCE:

- For both the system the phase starts at -90 and hence there is a pole at the origin for both the systems.

- From the tabular column Gain margin for A is 48dB and that for B it is infinity.

Generally, for stability Gain margin is expected to be more.

- Since for the magnitude Bode plot B the Gain margin is infinity it is inferred that it is a very stable system, but the response is sluggish with no oscillations. It is a very stable system because infinite gain can be increased or decreased, and the stability of the system won't be affected

- For the bode plot B 48db can be increased or decreased till the system becomes unstable.

- Greater the phase margin greater the stability of the system. The phase margin of B is > than the phase margin of A.

- The greater stability of B over A is because B can tolerate a maximum phase change of 94 degrees without making the system unstable whereas A can tolerate a maximum of 84degrees without making the system unstable.

RESULT:

Thus, the 3rd order system was analyzed using bode plot and the frequency parameters were tabulated

EXP 8: Determination of Transfer Function of a Physical System using Frequency Response and Bode's Asymptotes

AIM:

To determine the transfer function of a physical system from the given frequency response curve (Bode plot).

APPARATUS:

S.no	Software	Version
1.	MATLAB	R2021

FORMULA/THEORY:

Frequency response is the quantitative measure of the output spectrum of a system inresponse to a stimulus, usually a sinusoidal input, and is used to characterize the dynamics of the system. Frequency response analysis is the technique whereby a sinusoidal test signal is usedto measure points on the frequency response of a transfer function.

It is sometimes useful, while working on experimental data obtained from physicalsystems, to reverse engineer the frequency response analysis, i.e., obtaining the transfer function of a system from a given Bode plot. This is possible because, the Bode plot completely gives the frequency domain parameters. From the magnitude and phase plots, it is possible to determine the no. of poles andzeros, the relative order of the system, the break frequencies of the poles and zeros and the open-loop system gain.

Method for the estimation of transfer function from the Bode plot:

1. **Relative Order of the system:**

   It is known that each pole in a system contributes $-90°$ to the phase, whereas eachpole in a system contributes $+90°$ to the phase. Thus, by looking at the phase plot, one can determine the relative order of the system,i.e., how many extra poles are there than zeros in a system or vice versa.

2. **Number of zeros and poles:**

   We utilize the magnitude plot to determine the number of poles and zeros in a system. The slope changes in the magnitude plot will help determine the number of poles and zeros. For a pole, the slope change is −20 dB/dec whereas for a zero, it is +20 dB/dec.

3. **Location of poles and zeros:**

   To obtain the locations of poles and zeros, it is needed to find the break frequencies of the poles and zeros. Before that, the starting slope of the magnitude plot will tell us the type of the system.

$$Starting\ slope = 0 => Type - 0\ system$$
$$Starting\ slope = \ -20\ {}^{dB}/_{dec} => Type - 1\ system$$
$$Starting\ slope = \ -40\ {}^{dB}/_{dec} => Type - 2\ system$$

To find the break frequencies of the poles/zeros, it is necessary to see the bode asymptotic plot. The frequencies where the slope changes in the magnitude plot will give the break frequencies of the poles/zeros. At this point, we can estimate the transfer function as shown below:

$$G(s) = \frac{K\left(\frac{s}{w_{c_{z1}}} + 1\right)\left(\frac{s}{w_{c_{z2}}} + 1\right)\dots\dots}{s\left(\frac{s}{w_{c_{p1}}} +|1\right)\left(\frac{s}{w_{c_{p2}}} + 1\right)\dots\dots}$$

Where:

K – *open loop system gain*

w_{czn} $-\ break\ frequencies\ of\ zeros$

$w_{cpn} - break\ frequencies\ of\ poles$

4. **Finding Gain:**

   To find the open loop system gain K, it is necessary to look at the start of the Bode magnitude plot. If the magnitude at the starting frequency ωs is M $(in\ dB)$, then:

$$20\ \log_{10} \frac{K}{\omega_s} = M$$

   From this, system gain can be measured, and the final transfer function can be found out.

CALCULATION:

To determine the Transfer function of the above plot:

1. **Relative order of the system:**

   - ❖ The phase plot ends at −180.

- Thus, the system has *2 more poles than zeros*.

2. **Number of zeros and poles:**

- From the magnitude plot, there are no slope changes from negative to positive (+20 dB/dec). Thus, the system has *no zeros*.

  Number of poles=2
  Number of zeros=0

3. **Location of poles and zeros:**

- From the magnitude plot, it can be seen that the slope with which it starts is -20 dB/dec. Thus, the system is a type-1 system, i.e., it has a pole at s = 0. This can also be said from the fact that the phase plot begins at −90°.

- Secondly, we need to look for change in slope and the corresponding frequency. From the magnitude plot, it is clear that at $\omega cp1$ = 0.5 rad/s, there is a change in slope from −20 $dB/dec\ to$−40 dB/dec. Thus, $\omega cp1$is a break frequency.

$$G(s) = \frac{K}{s\left(\frac{s}{\omega_{c_{p1}}}+1\right)}$$

$$G(s) = \frac{K}{s\left(\frac{s}{0.5}+1\right)}$$

$$G(s) = \frac{K}{s(1+2s)}$$

4. **Finding Gain:**

$$20\ \log_{10}\frac{K}{\omega_s} = M$$

- From the magnitude plot, ωs = 0.1 rad/s and M = 34 dB. Thus:

$$20\ \log_{10}\frac{K}{0.1} = 34$$
$$10^{\left(\frac{34}{20}\right)} = \frac{K}{0.1}$$
$$K = 5.001 \cong 5$$

- Thus, the transfer function is given by:

$$\boldsymbol{G(s) = \frac{5}{s(1+2s)}}$$

MATLAB CODE:

```
>> sys = tf([5], [2 1 0])
>> figure (1)
>>bode(sys)
>>title ('Bode Plot of G(s)')
>>figure (2)
>>asymp(sys)
>>title ('Asymptotic Bode Plot of G(s)')
```

OBSERVATIONS:

MATLAB:

INFERENCE:

- A pole contributes −90° to the phase plot because:

$$G(j\omega) = \frac{1}{(j\omega + k)}$$

$$and, \varphi = -\tan^{-1}\frac{\omega}{5}$$

- The time constant form of the transfer function is:

$$G(s) = \frac{K}{s(\tau s + 1)}$$

$$\tau = \frac{1}{\omega_c}$$

- This method of finding the transfer function is immensely helpful when working with experimental data. From the data, the Bode plot can be obtained and thus, the transfer function of the system can be easily obtained.

RESULT:

Thus, the transfer function of the given system was found using frequency response and Bode’s asymptotes.

EXP 9: Design of Lag, Lead compensators and Evaluation of Closed Loop Performance

AIM:

To design Lag, lead compensators and evaluation of closed loop performance for the given system using MATLAB.

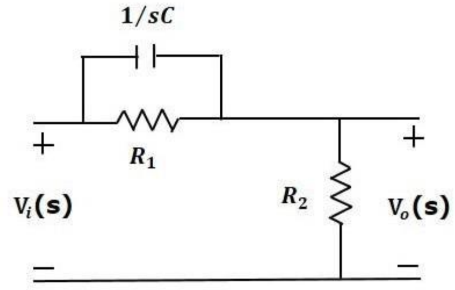
APPARATUS:

S.no	Software	Version
1.	MATLAB	R2021

FORMULA/THEORY:

1. **Lead compensator:**

The lead compensator is an electrical network which produces a sinusoidal output having phase lead when a sinusoidal input is applied. The lead compensator circuit in the ‘s’ domain is shown in the following figure. Here, the capacitor is parallel to the resistor R1 and the output is measured across resistor R2.



The transfer function of this lead compensator is –

$$\frac{V_o(s)}{V_i(s)} = \beta \left(\frac{s\tau + 1}{\beta s\tau + 1} \right)$$

Where,

$$\tau = R_1C$$

$$\beta = \frac{R_2}{R_1 + R_2}$$

From the transfer function, we can conclude that the lead compensator has pole at $s = -\frac{1}{\beta}$ and zero at $s = -\frac{1}{\beta\tau}$.

Substitute, $s = j\omega$ in the transfer function.

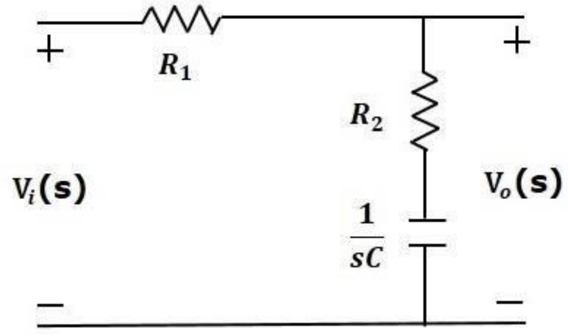
$$\frac{V_o(j\omega)}{V_i(j\omega)} = \beta \left(\frac{j\omega\tau + 1}{\beta j\omega\tau + 1} \right)$$

Phase angle $\phi = tan^{-1}\omega\tau - tan^{-1}\beta\omega\tau$

We know that the phase of the output sinusoidal signal is equal to the sum of the phase angles of input sinusoidal signal and the transfer function. So, to produce the phase lead at the output of this compensator, the phase angle of the transfer function should be positive. This will happen when 0<β <1. Therefore, zero will be nearer to origin in pole-zero configuration of the lead compensator.

2. Lag Compensator:

The Lag Compensator is an electrical network which produces a sinusoidal output having the phase lag when a sinusoidal input is applied. The lag compensator circuit in the ‘s’ domain is shown in the following figure.



Here, the capacitor is in series with the resistor R2 and the output is measured across this combination.

$$\frac{V_o(s)}{V_i(s)} = \frac{1}{\alpha}\left(\frac{s+\frac{1}{\tau}}{s+\frac{1}{\alpha\tau}}\right)$$

Where,

$$\tau = R_2C$$

$$\alpha = \frac{R_1+R_2}{R_2}$$

From the above equation, α is always greater than one.

From the transfer function, we can conclude that the lag compensator has one pole at $s = -\frac{1}{c\alpha}$ and one zero at $s = -\frac{1}{\alpha}$. This means, the pole will be nearer to origin in the pole-zero configuration of the lag compensator.

Substitute, $s = j\omega$ in the transfer function.

$$\frac{V_o(j\omega)}{V_i(j\omega)} = \frac{1}{\alpha}\left(\frac{j\omega+\frac{1}{\tau}}{j\omega+\frac{1}{\alpha\tau}}\right)$$

Phase angle $\phi = \tan^{-1}\omega\tau - tan^{-1}\alpha\omega\tau$

We know that the phase of the output sinusoidal signal is equal to the sum of the phase angles of input sinusoidal signal and the transfer function. So, to produce the phase lag at the output of this compensator, the phase angle of the transfer function should be negative. This will happen when $\alpha > 1$.

PROCEDURE:

Q1. The open loop transfer function of certain unity feedback control system is given by G(s) = K/s(s+4) (s+80). It is desired to have the phase margin to be at least 33° and the velocity error constant Kv= 30 sec-1. Design a phase lag series compensator.

***ANS*:**

1. Determining the value of K.

$$K_v = \underset{s \to 0}{Lt}\ sG(s)$$

2. The bode plot of the uncompensated system.
3. The phase margin is calculated.
4. Suitable values for the phase margin of the compensated system.
5. New gain crossover frequency is determined.

$$\gamma_n = 180° + \phi_{gcn}$$

6. The value β of the compensator is determined.

$$A_{gcn} = 20\log \beta$$

7. The transfer function of lag compensator is determined.

   The zero of the compensator is arbitrarily placed at one tenth of the new gain crossover frequency. Where Zc and Pc is the zero and pole of the compensator.

$$z_c = \frac{1}{T} = \frac{\omega_{gcn}}{10}$$

$$p_c = 1/\beta T$$

   The TF is given as follows

$$\frac{s + \frac{1}{T}}{s + \frac{1}{\beta T}}$$

8. The open loop response is generated by placing the plant in series with the lag compensator.

The actual phase margin is calculated at the new gain crossover frequency.

Q2. Design a lead compensator for a unity feedback system with open loop transfer function, G(s) = K/s(s+8) to satisfy the following specifications (i) percentage peak overshoot = 9.5% (ii) natural frequency response, ωn = 12 rad/sec (iii) Velocity error constant Kv ≥ 10.

ANS:

1. Determination of the dominant pole with the given parameters

$$s_d = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}\ .$$
$$\%M_p = e^{-\zeta\pi/\sqrt{1-\zeta^2}} \times 100$$

2. The pole zero plot is drawn.
3. Angle contributed by the lead network

Let ϕ = Angle to be contributed by lead network to make point, P as a point on root locus.

$$\text{Now, } \phi = \begin{pmatrix}\text{sum of angles}\\ \text{contributed by poles}\\ \text{of uncompensated system}\end{pmatrix} - \begin{pmatrix}\text{sum of angles}\\ \text{contributed by zeros}\\ \text{of uncompensated system}\end{pmatrix} - 180^\circ$$

4. The poles and zeroes of the compensator are found

$$z_c = -1/T$$
$$p_c = -1/\alpha T$$

5. Determination of the TF of the compensator

$$G_c(s) = \frac{(s+\frac{1}{T})}{(s+\frac{1}{\alpha T})}$$

6. Determining the open loop transfer function of the lead compensated system

$$K = \frac{\text{Product of vector lengths from all poles to } s = s_d}{\text{Product of vector lengths from all zeros to } s = s_d}$$

   The lead compensator is connected in series with the plant the value of K is determined by the above formula. And the vector lengths from all poles are represented as l1,l2,l4 and for zeroes it is l3.

7. The check for error requirement is done.

MATLAB CODE:

Q1.

```
>> Kv = 30
>> K = Kv * 4 * 80
>> G=tf(K, conv([1 4 0],[1 80]))
>> [Gm Pm] = margin(G)
>> margin(G)
>> Pd = 33
>> Pd = Pd + 5
>> Phgcm = -142
>> wgcm= 4.58 % from the first bode plot use the mouse pointer and locate wgcm
corresponding to phgcm.
>> [Beta,p]=bode(G,wgcm)
>> T=10/wgcm
>> Zc=1/T
>> Pc=1/(Beta*T)
>> Gc=tf([1 Zc],[1 Pc])
>> sys=Gc*G/Beta
>> figure(1)
>> margin(G)
>> hold on
>> margin(sys)
>> legend('Without Compensator', 'With Lag compensator')
>> figure(2)
>> step(feedback(G,1))
>> hold on
>> step(feedback(sys,1))
>> legend('Without Compensator', 'With Lag compensator')
```

Q2.

```
>> Mp=9.5
>> e=log(Mp)*sqrt(1/(pi^2+(log(Mp)^2)));
>> w=12;

%open loop poles

>> s1 = 0
>> s2 = -8
```

```
%Dominant pole

>> sd1=-e*w+j*w*sqrt(1-e^2)
>> sd2=-e*w-j*w*sqrt(1-e^2)
>> theta1=angle(sd1 - s1)
>> theta2=angle(sd1 - s2)
>> phi = theta1 + theta2 - 3.14
>> phi1 = (phi)/2

%finding the pole and the zero

>> m2 = tan(theta1/2 - phi1)
>> x = [linspace(-100,100)]
>> syms k
>> eqn = m2*(k + 7.2) + 9.6 == 0;
>> Pc = double(solve(eqn)); % X values where y=0
>> m3 = tan((theta1/2) + phi1)
>> syms k
>> eqn = m3*(k + 7.2) + 9.6 == 0;
>> Zc = double(solve(eqn)); % X values where y=0
>> T=-1/Zc;
>> a=-1/(Pc*T);
>> s=tf('s');
>> G=(s+1/T)/(s+1/(a*T));
>> l1 = abs(sd1 - s1)
>> l2 = abs(sd1 - s2)
>> l3 = abs(sd1 - Zc)
>> l4 = abs(sd1 - Pc)

>> K=(l1*l2*l4)/l3;
>> G=K*G/(s*(s+8))
>> sys=K/(s*(s+8))
>> figure(1)
>> margin(sys)
>> hold on
>> margin(G)
>> legend('Uncompensated','Compensated')
>> figure(2)
>> step(feedback(sys,1))
>> hold on
>> step(feedback(G,1))
>> legend('Without Compensator', 'With Lead compensator')
```

OBSERVATION:

Q1.

Q2.

TABULATION:

Parameters	Q1		Q2	
	Uncompensated	Compensated	Uncompensated	Compensated
Rise time(s)				
Settling Time(s)				
Overshoot(%)				
Peak time(s)				
Peak Value				

INFERENCE:

- Lag compensators reduce the gain crossover frequency of the system. Thus, the system response becomes slower.
- Lag compensators reduce the high frequency gain of the system. Hence, the high frequency noise attenuation becomes better.

- The time constant of the lag compensator should not be unnecessarily large. Otherwise, the system's response will be too slow.

- From the observation table for the first question Lag compensators allow a high gain at low frequencies and thus improve the steady state error.

- From the observation table the presence of a lead compensator increases the gain of the system at high frequencies (the amount of this gain is equal to a).

- This can increase the crossover frequency, which will help to decrease the rise time and settling time of the system (but may amplify high frequency noise).

- In a lead compensated design additional positive phase increases the phase margin and thus increases the stability of the system.

- And it is seen that the gain margin of lead compensated system is infinity. A gain margin of infinity means that no matter how much you increase the gain, the system will always be stable.

- Due to the presence of phase lead compensation maximum overshoot of the system decreases.

- The general comparison is summarized below:

    - If the design specifications are given in terms of steady state error and phase margin, either the lead or the lag compensator may be able to meet the requirement.
    - However, if specification on rise time or gain crossover frequency is also to be met, then only one of the two compensators would be able to do the job. Or we need to combine lag and lead compensator.

RESULT:

Thus, the lead and lag compensators were designed as per the requirements and their closed loop performance was analyzed. The corresponding graphs were also plotted and analyzed.

EXP 10: Design of PID controllers and Evaluation of Closed Loop performance

AIM:

To design PI, and PID controllers and evaluate their closed loop performance.

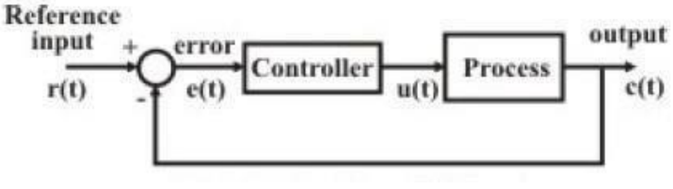
APPARATUS:

S.no	Software	Version
1.	MATLAB	R2021

FORMULA/THEORY:

A controller is a mechanism that seeks to minimize the difference between the actual value of a system (i.e., the process variable) and the desired value of the system (i.e., the setpoint). Controllers are a fundamental part of control engineering and used in all complex control systems. Some of the uses are:

- Controllers improve the steady-state accuracy by decreasing the steady state error. As the steady-state accuracy improves, the stability also improves.
- Controllers also help in reducing the unwanted offsets produced by the system, while reducing the noise signals produced by the system.
- Controllers can help to speed up the slow response of an overdamped system.



Controllers are of different types, but the most common and popular one is the Proportional-Integral-Derivative (PID) controller. The general form of a PID controller in s-domain is as follows:

$$\boldsymbol{C(s) = K_p + sK_d + \frac{K_i}{s}}$$

where Kp, Kd and Ks are the gain constants of the different elements.

1. Proportional Control:

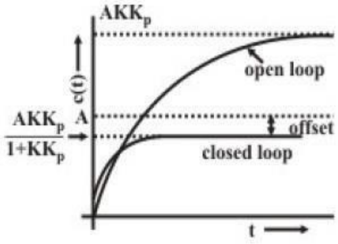
Consider the below block diagram, where $\frac{K}{1+sc}$ is the system that needs to be controlled and τ is the system time constant. The closed loop transfer function of the above system is:

$$C(s) = \frac{KK_p}{1 + KK_p} * \frac{AKK_P}{1 + sr}$$

And the system response c(t) for a unit step input is:

$$c(t) = \frac{AKK_P}{1 + KK_P}(1 - e^{-\frac{st}{c}})$$

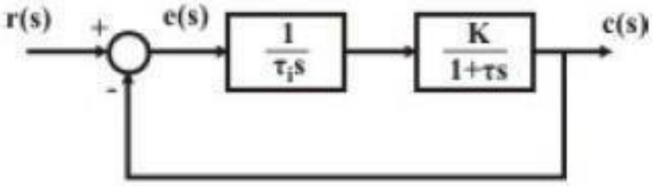
The system response is given below:



By using proportional control, the time response improves by a factor of $\frac{1}{1+KK_P}$ which is good for the system, but we get a steady state offset between the desired response and the output response. This can be reduced by increasing the gain of the controller, but it introduces oscillations.

2. Integral Control:

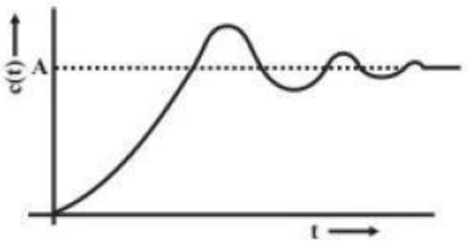
Closed loop block diagram of a system with only integral controller action is given by:



where $K_i = \frac{1}{c_i}$. The closed loop transfer function of this system is given by:

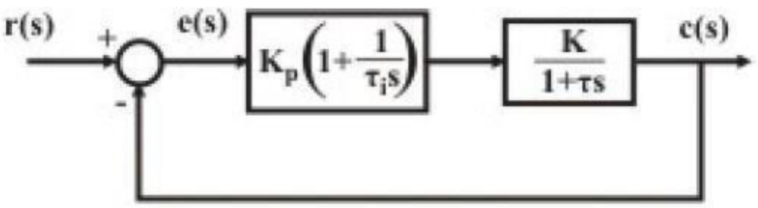
$$C(s) = \frac{K}{K + \tau_i s + \tau\tau_i s^2}$$

With integral control, order of system increases by one. For a step input, using the above transfer function, the steady state error turns out to be zero. Thus, a major advantage of the I-controller is that it eliminates steady state error, but the output response is generally slow and oscillatory.

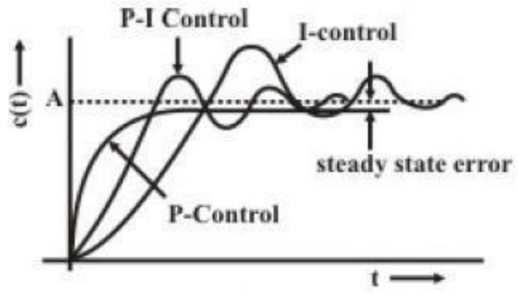


3. Proportional - Integral (PI) control:

Closed loop block diagram of a system with only a PI controller action is given by:

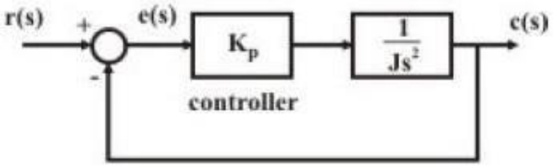


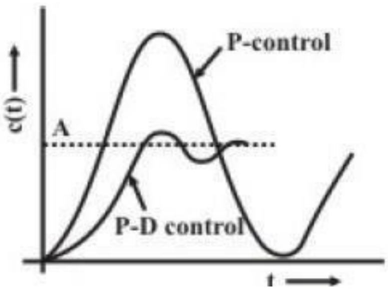
where $K_i = \frac{1}{c_i}$. From the discussions of P and I controllers, a PI controller provides dual advantages of fast response due to P-action and the zero steady state error due to I-action.



By varying Kp, it is possible to vary the damping ratio of the system

4. Proportional Derivative (PD) Control:





For higher order processes, a PD controller improves the stability of the system. It also improves transient response of the system.

5. Proportional-Integral-Derivative (PID) Control:

It is clear from above discussions that a suitable combination of proportional, integral, and derivative actions can provide all the desired performances of a closed loop system. The transfer function of a P-I-D controller is given by:

$$\boldsymbol{C(s) = K_p + sK_d + \frac{K_i}{s}}$$

The order of the controller is low, but this controller has universal applicability; it can be used in any type of SISO system. However, for proper use, a controller must be tuned for a particular process; i.e., selection of P, I, D parameters are especially important and process dependent.

Unless the parameters are carefully chosen, a controller may cause instability to the closed loop system. It is not always necessary that all the features of proportional, derivative, and integral actions should be incorporated in the controller. In fact, in most of the cases, a simple P-I structure will suffice.

Guideline for selection of controller mode:

1. *Proportional Controller:*

It is simple regulating type; tuning is easy. But it normally introduces steady state error. It is recommended for process transfer functions having a pole at origin, or for transfer functions having a single dominating pole; for example, with:

$$P(s) = \frac{K}{(1 + r_1 s)(1 + r_2 s)(1 + r_3 s)}$$

2. *Integral Control:*

It does not exhibit steady state error but is relatively slow responding. It is particularly effective for: fast process, with high noise level, process dominated by dead time, high order system with all-time constants of the same magnitude.

3. *Proportional plus Integral (P-I) Control:*

It does not because offset associated with proportional control. It also yields much faster response than integral action alone. It is widely used for process industries for controlling variables like level, flow, pressure, etc., those do not have large time constants.

4. *Proportional plus Derivative (P-D) Control:*

It is effective for systems having large number of time constants. It results in a more rapid response and less offset than is possible by pure proportional control. But one must be careful while using derivative action in control of fast processes, or if the measurement is noisy (ex: flow measurement).

5. *Proportional plus Integral plus Derivative (P-I-D) Control:*

It finds universal application. But proper tuning of the controller is difficult. It is particularly useful for controlling slow variables, like pH, temperature, etc. in process industries.

DESIGNING A PI/PID CONTROLLER ACCORDING TO GIVEN SPECIFICATIONS:

- The most common way of tuning a P/PI/PID controller is using Ziegler-Nichol's technique, but that method cannot be used if design specifications are given.
- Two types of specifications can be given:
  - ✓ In frequency domain, like phase margin
  - ✓ In time domain, like damping ratio, steady state error
- We employ two different methods of design depending on the type of specification given.

For frequency domain specifications:

When the phase margin and gain crossover frequency are the design parameters, the following steps are followed:

Step 1:

Determine magnitude and phase of uncompensated open loop sinusoidal transfer function G(jw)

$Let\ \emptyset_d\ the\ desired\ phase\ margin\ at\ gain\ crossover\ frequency\ \omega_1.$
$Let\ K1 = |G(jw)|\ at\ \omega = \omega_1\ and\ \emptyset_1 = \angle G(jw)\ at\ \omega = \omega_1$

Step 2:

Determine the phase margin of uncompensated system and the angle to be contributed by the controller to achieve the desired phase margin.

$Let\ \emptyset_u - Phase\ margin\ of\ uncompensated\ system\ and\ \theta - Phase\ angle\ of\ controller\ at\ \omega = \omega_1$

Now, $\emptyset_u = 180° + \emptyset_1$ $and\ \theta = \emptyset_d - \emptyset_u$.

Step 3:

Determine the Transfer function of the controller.

For a PI controller:

$$K_i = \frac{-\omega_1 \sin(\theta)}{A_1}$$
$$K_p = \frac{\cos(\theta)}{A_1}$$

$$G_c(s) = K_p + \frac{K_i}{s}$$

Step 4:

Determine open loop transfer function of compensated system.

$$G_0(s) = G_c(s).G(s)$$

For time domain specifications:

When the damping ratio ζ, error constant and natural frequency ω_d are the design parameters, the following steps are followed:

Step 1:

Determine the dominant pole s_d and calculate its magnitude and phase:

$$s_d = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$$
$$Let\ D = |s_d|\ and\ \beta = \angle s_d$$

By considering the dominant pole

$$D = \sqrt{\zeta^2{\omega_n}^2 + {\omega_n}^2(1-\zeta^2)}\ \ and\ \beta = \tan^{-1}\frac{\sqrt{1-\zeta^2}}{-\zeta}$$

Step 2:

Determine magnitude and phase of G(s) at s=s_d.

Step 3:

Determination of transfer function of PI/PID controller

For PI controller:

$$K_i = \frac{-D sin(\emptyset_d)}{A_d \sin(\beta)}$$

$$K_p = \frac{-sin(\beta + \emptyset_d)}{A_d \sin(\beta)} - \frac{2K_i \cos(\beta)}{D}$$

$$G_c(s) = K_p + \frac{K_i}{s}$$

For PID controller:

$$G_c(s) = K_p + sK_d + \frac{K_i}{s}$$

Determine K_i from the specified error constant, such that the compensated system meets the error requirement.

Calculate the parameter K_D and K_P using the following equations

$$K_p = \frac{-sin(\beta + \emptyset_d)}{A_d \sin(\beta)} - \frac{2K_i \cos(\beta)}{D}$$

$$K_d = \frac{\sin(\emptyset_d)}{D A_d \sin(\beta)} + \frac{K_i}{D^2}$$

$$G_c(s) = K_p + \frac{K_i}{s} + sK_d$$

Step 4:

Determine open loop transfer function of compensated system

Questions Given:

1. Consider a unity feedback system with open loop transfer function G(s) = 100/(s+1)(s+2)(s+5). Design a **PI controller**, so that the phase margin of the system is 60° at a frequency of 0.5 rad/sec.

2. Consider a unity feedback system with open loop transfer function G(s) = 4/(s+1)(s+5). Design a **PI controller** so that the closed loop has a damping ratio of 0.9 and natural frequency of oscillation as 2.5 rad/sec.

3. Consider a unity feedback system with open loop transfer function G(s) = 75/(s+1)(s+3)(s+8). Design a **PID controller** to satisfy the following specifications (a) steady state error for unit ramp input should be less than 0.08, (b) Damping ratio = 0.8 (c) Natural frequency of oscillation = 2.5 rad/sec.

MATLAB CODE:

For 1st question:

```
>> PM=60/57; %60 degrees converted to radians
>> w=0.5;
>> s=tf('s');
>> G=10/((s+1)*(1+0.5*s)*(1+0.2*s))
>> G1=10/((j*w+1)*(1+0.5*j*w)*(1+0.2*j*w))
>> M1=abs(G1)
>> theta1=(angle(G1))

%finding phase margin and theta2

>> PM1=3.14+(theta1)
>> theta2=PM-PM1

%determining constants

>> Ki=(-w*sin(theta2))/M1
>> Kp=cos(theta2)/M1
>> contr=pid(Kp,Ki)
>> OL=contr*G
>> figure(1)
>> step(feedback(G,1))
>> hold on
>> step(feedback(OL,1))
>> legend('Without controller', 'With controller')
>> figure(2)
>> subplot 211
>> rlocus(OL)
>> title('Root Locus of G(s)')
>> subplot 212
>> rlocus(G)
>> title('Root locus of compensated system')
```

For 2nd question:

```
>> e=0.9;
>> w=2.5;
>> s=tf('s');
>> G=4/((s+1) *(s+5));

%Dominant pole

>> sd1=-e*w + j*w*sqrt(1-e^2)
>> sd2=-e*w-j*w*sqrt(1-e^2)
>> M1=abs(sd1)
>> theta1=angle(sd1)

%Magnitude and Phase of G(s) at s=sd

>> G1=4/((sd1+1) *(sd1+5));
>> M2=abs(G1)
>> theta2=angle(G1)

%Determining the constants

>> Ki=(-M1*sin(theta2)/(M2*sin(theta1)))
>> Kp=(-sin(theta2+theta1)/(M2*sin(theta1)))-(2*Ki*cos(theta1)/M1)

%Finding the TF

>> cont=pid(Kp,Ki)
>> OL1=G*cont
>> figure(1)
>> step(feedback(G, 1))
>> hold on
>> step(feedback(OL, 1))
>> legend('No controller', 'With controller')
>> figure(2)
>> subplot 211
>> rlocus(OL)
>> title('Root Locus of G(s)')
>> subplot 212
>> rlocus(G)
>> title('Root locus of compensated system')
```

For 3rd question:

```
>> e=0.8;
>> ess=0.08;
>> w=2.5;
>> s=tf('s');
>> G=75/((s+1)*(s+3)*(s+8));

%Dominant pole

>> sd1=-e*w+j*w*sqrt(1-e^2)
>> sd2=-e*w-j*w*sqrt(1-e^2)
>> M1=abs(sd1)
>> theta1=angle(sd1)

%Magnitude and Phase of G(s) at s=sd

>> G1=75/((sd1+1)*(sd1+3)*(sd1+8));
>> M2=abs(G1)
>> theta2=angle(G1)

%Determining the constants

>> Kv=1/ess
>> Ki=Kv/3.125
>> Kd=(sin(theta2)/(M1*M2*sin(theta1)))+(Ki/M1^2)
>> Kp=(-sin(theta2+theta1)/(M2*sin(theta1)))-(2*Ki*cos(theta1)/M1)

%Finding the TF

>> cont=pid(Kp,Ki,Kd)
>> OL=G*cont
>> figure(1)
>> step(feedback(G, 1))
>> hold on
>> step(feedback(OL, 1))
>> legend('No controller', 'With controller')
>> figure(2)
>> subplot 211
>> rlocus(OL)
>> title('Root Locus of G(s)')
>> subplot 212
>> rlocus(G)
>> title('Root locus of compensated system')
```

OBSERVATIONS:

First question:

Second question:

Third question:

TABULATIONS:

s no	Settling Time(s)		Steady state error	
	Without Controller	With Controller	Without Controller	With Controller
1.				
2.				
3.				

INFERENCE:

- From the observations, a PI/PID controller always makes the steady state error vanish.
- From the plots obtained, the system's closed loop step response reaches steady state more quickly, except in the case of controller 2, where the design specification led to the settling time being higher than the uncompensated system.

- From the Bode plot of controller 1, we can see that the design specification of phase margin has been met, thus, the controller is functioning properly.
- From the root locus plots of the compensated and uncompensated systems, we can see that the addition of the poles/zeros by the controller pulls the root locus towards the left so that it can accommodate the necessary gains.
- The step responses inform us about the closed loop performance of the system. The primary goal of any system is to obtain the desired response, i.e., the input. More the system is close to the input, more closed loop stable it is.

RESULT:

Thus, the PI and PID controllers were designed and their corresponding closed loop performance was analyzed.

EXP 11: Discretization of a Continuous System and the Effect of Sampling

AIM:

To discretize the given continuous system and analyze the effect sampling using MATLAB.

APPARATUS:

S.no	Software	Version
1.	MATLAB	R2021

FORMULA/THEORY:

Discretization is the process of transferring continuous functions, models, variables, and equations into discrete counterparts. This process is usually carried out as a first step toward making them suitable for numerical evaluation and implementation on digital computers. It must be made clear that, while discretizing a system, information/data is always lost. Thus, depending on the necessities, there can be different methods of discretizing a system.

Theoretical Method:

The theoretical way of discretizing a system is by:

1. Converting given s-domain transfer function to time domain

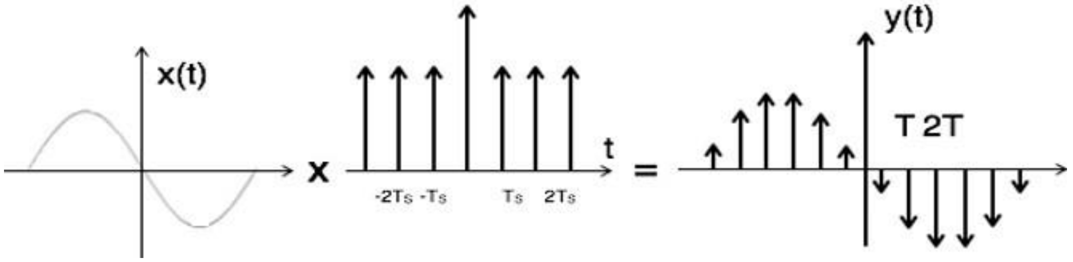
2. Sampling the obtained time domain response

3. Taking z-transform of the sampled response to obtain the z-domain transfer function.

This method is commonly known as the impulse invariant method because, the impulse response of both the continuous system and the discrete system will remain the same. (The discrete time impulse response being the sampled continuous time impulse response).

Let a unit impulse train be defined as follows:

$$Unit\ impulse\ train = \sum_{n=-\infty}^{\infty} \delta(t - nT)$$



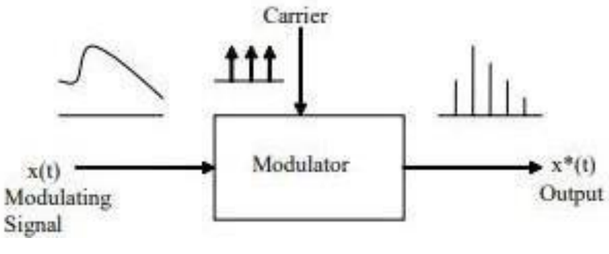
If a continuous-time signal x(t) is sampled in a periodic manner, mathematically, the sampled signal can be represented as follows:

$$x^*(t) = \sum_{n=-\infty}^{\infty} x(t)\,\delta(t - nT)$$

$$or, x^*(t) = \sum_{n=-\infty}^{\infty} x(nT)\,\delta(t - nT)$$

where:

$x*(t)$ – $sampled\ signal$
T – $sampling\ time$



The disadvantage with using this method to sample an actual continuous system is not always ideal. This is because, impulse trains are not usually the type of input that act as a discrete input when you have a discrete system that interacts with a continuous system.

For example, consider a discrete system that constantly inputs a motor with the speed for a defined sample time. With impulse sampling, the motor will throttle only for the period of the impulse and then slow down, only to speed up in the next impulse. But this is not what is ideal.

Ideally, we would want the motor to hold its speed constant until the next sample arrives from the discrete system. **This is achieved through data hold.**

Data Hold:

Data hold is a process of generating a continuous-time signal h(t) from a discrete-time sequence x(nT).

The signal h(t) during the time interval $nT \leq t \leq (k + 1)T$ $(T - sample\ time)$, maybe approximated as a polynomial as follows:

$$h(nT + r) = a_n r^n + a_{n-1}\, r^{n-1} + \cdots + a_1 r + x(nT)$$

If the data hold circuit is an n-th order polynomial extrapolator, it is called an n-th order hold. It uses the past (n+1) discrete data to generate the polynomial.

Sampling:

Sampling is the reduction of a continuous-time signal to a discrete-time signal. A sample is a value or set of values at a point in time and/or space. A **sampler** is a subsystem or operation that extracts samples from a continuous signal.

A theoretical ideal sampler produces samples equivalent to the instantaneous value of the continuous signal at the desired points. While sampling, we come across several problems, one of which is called **aliasing**.

Aliasing:

Aliasing refers to the distortion or artifact that results when a signal reconstructed from samples is different from the original continuous signal. In addition to this, by altering the sampling rate, we also **decimate or extrapolate** the discretized signal.

Decimation:

Consider a signal x(n) compressed by a factor M. This yields: $xd(n) = x(Mn)$

This means that, only every M-th sample is considered. So, we lose information while we decimate a signal.

Interpolation:

Consider a signal x(n) expanded by a factor L. These yields: $xe(n) = x(n/L)$

Doing this, every 2nd sample will become zero. So, we expand the sampled data. The newly inserted sampling points are calculated using a proper interpolating function.

QUESTIONS:

Q1. Determine the z-domain transfer function for the following s-domain transfer function.

$$H(s) = \frac{a}{(s+a)^2}$$

Discretize the given continuous system and study the effect of sampling.

PROCEDURE:

Conversion to s-domain:

Step 1: Take Inverse Laplace transform of H(s)

$$f^{-1}(H(s)) = f^{-1}\left(\frac{a}{(s+a)^2}\right)$$
$$h(t) = a\,t\,e^{-at}$$

Step 2: Sample the continuous response by setting t = nT where T – sampling time:

$$h(nT) = a\,nT\,e^{-anT}$$

Step 3: Finding the z-transform

$$Z_n[a\,nT\,e^{-anT}](z) = \frac{aTze^{aT}}{(ze^{aT}-1)^2}$$

MATLAB CODE:

Q1.

```
>> a = 2;
>> e = 2.71;
>> i = 1;
>> t = 0.3;
>> s = tf('s');
>> z = tf('z');

>> %converting s-domain to z-domain

>> h = a/(s+a)^2;
>> disp("The s-domain transfer function converted to s-domain is: ")
>> sys = c2d(h, 1, 'impulse')
>> step(h)
>> hold on

>> while(t <= 2.7)
>> f=(a*t*z*e^(a*t))/((z*e^(a*t)-1)^2);  %from theory
>> step(f)
>> hold on
>> t = t*3;
>> end
>> legend('s-domain','Ts = 0.3','Ts=0.9','Ts=2.7')
>> t = 1;
>> i = 1;
>> Ar = [2 4 6];

>> while(a<=6)
>> figure(i+1)
>> h = a/(s+a)^2;
>> f=(a*t*z*e^(a*t))/((z*e^(a*t)-1)^2);
>> step(h)
>> hold on
>> step(f)
>> legend(strcat('continuous: a=',num2str(Ar(i))), strcat('discrete:
a=',num2str(Ar(i))))
>> i = i+1;
>> a= a+2;
>> end
```

OBSERVATION:

Q1.

$Plot\ 1: Fixed\ a, varying\ sampling$

$time$

$Plot\ 2: a = 2, Ts = 1$

$Plot\ 3: a = 4, Ts = 1$

$Plot\ 4: a = 6, Ts = 1$

INFERENCE:

- If the signal is slowly varying, then fewer samples per second will be required than if the waveform is rapidly varying.
- The major drawback of random undersampling is that this method can discard potentially useful data that could be important for the induction process.
- It can be seen that undersampling causes poor resolution and the step response is not smooth as well.
- As the sampling time increases the resolution gets clearer. This is because of Oversampling. It can be inferred that oversampling helps in improving the resolution and also data is not discarded
- From the above it can be inferred that:
  1. Oversampling methods duplicate or create new synthetic examples in the minority class.
  2. Undersampling methods delete or merge examples in the majority class.
- As the ‘a’ in the equation increases, the resolution increases in the step response of the z domain

RESULT:

Thus, the given continuous system (in the form of transfer function) was converted to z- domain and was discretized. The effect of sampling was studied using the step response plots.

EXP 12: Test of Observability and Controllability in Continuous and Discrete Domain in Simulation Platform

AIM:

To test the controllability and observability of a given system in both s and z domain using MATLAB.

APPARATUS:

S.no	Software	Version
1.	MATLAB	R2021

FORMULA/THEORY:

Controllability and observability are two important properties of state models which are integral to the designing of a controller. These two properties suggest what modifications should be made to the plant (actuators and sensors) so that the desired action is achieved.

Controllability deals with the possibility of forcing the system to a particular state by application of a control input. More formally, if a state is controllable, it means that there exist some control signals that allow the system to reach any state in a finite amount of time. It is also known as reachability. If a state is uncontrollable, then no input will be able to control that state.

On the other hand, whether the initial states can be observed from the output or not is determined using observability property. If all states of a system can be known from the system output, then the system is observable.

It is impractical to know all the states of a system. So, sometimes it is limited to the critical states. Thus, if a state is not observable then the controller will not be able to determine its behavior from the system output and hence not be able to use that state to stabilize the system.

Controllability and observability work together, as stated in a further theorem.

Controllability:

Consider a dynamic system as follows:

$$x(k+1) = Ax(k) + Bu(k)$$

$$y(k) = Cx(k) + Du(k)$$

$$where\ A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}, C \in \mathbb{R}^{p\times n}, D \in \mathbb{R}^{p\times m}$$

The state equation (or the pair (A, B)) is said to be completely state controllable or simply state controllable if for any initial state x(0) and any final state x(N), there exists an input sequence u(k), k = 0, 1, 2, · · · , N, which transfers x(0) to x(N) for some finite N. Otherwise the state equation is state uncontrollable.

1. The state equation (1) or the pair (A, B) is state controllable **if and only if the n × np state controllability matrix has rank n**, i.e., full row rank.
   The state controllability matrix is given as follows:
   $$Uc = [B\ AB\ A^2B\ \dots\dots\dots\ A^{n-1}\ B]$$
   The full rank condition can also be stated as follows: If the determinant of the the controllability matrix is zero, then the matrix is full rank.

2. When the matrix 'A' has distinct eigenvalues and in Jordan/Diagonal canonical form, the state model is controllable if and only if all the rows of matrix 'B' are non-zero.

Observability:

The state model (or the pair (A, C)) is said to be observable if any initial state x(0) can be uniquely determined from the knowledge of output y(k) and input sequence u(k), for k = 0, 1, 2, · · · , N, where N is some finite time. Otherwise, the state model is unobservable.

1. The state model (1) or the pair (A, C) is observable if the np × n observability matrix has rank n, i.e., full column rank.

$$U_o = \begin{pmatrix} C \\ CA \\ CA^2 \\ . \\ . \\ CA^{n-1} \end{pmatrix}$$

2. If the state model is in observable canonical form, then the system is observable.

Theorem of Duality:

The pair (A, B) is controllable if and only if the pair (A^T,B^T) is observable.

QUESTIONS:

Q1. Test the controllability and observability of the given system in both continuous and discrete domain.

Given characteristic equation: 5z2 − 2z + 2 = 0

Assume: It is a unity feedback system.

PROCEDURE:

Finding the transfer function:

We know that any transfer function is of the form:

$$T(z) = \frac{G(z)}{1 + G(z)H(z)}$$

Here, as it is unity feedback, H(z) = 1.

The characteristic equation of the above transfer function is:

$$1 + G = 0$$
$$G = -1$$
$$5z^2 - 2z + 1 = -1$$

Thus, $G(z) = 5z^2 - 2z + 1$

The transfer function in z-domain is:

$$T(z) = \frac{5z^2 - 2z + 1}{5z^2 - 2z + 2}$$

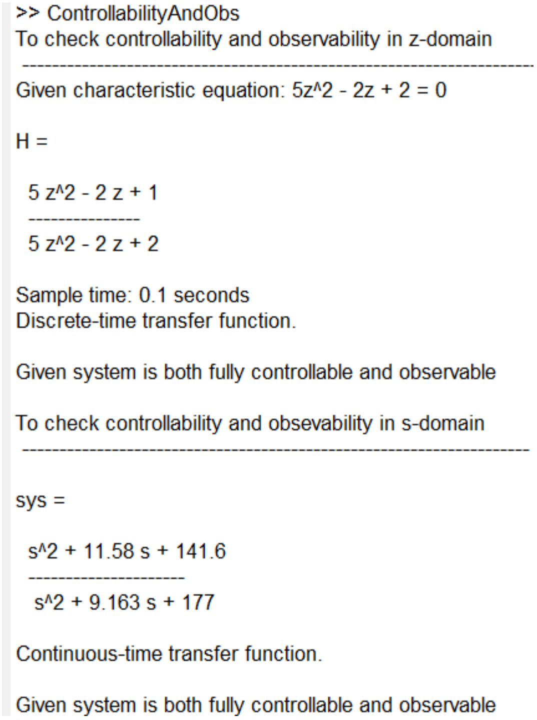
MATLAB CODE:

Q1.

```
disp('To check controllability and observability in z-domain')
disp(' --- ------------------------------------------------------------------ ')
disp('Given characteristic equation: 5z^2 - 2z + 2 = 0')

H=tf([5,-2,1],[5,-2,2],0.1)
[b1 a1] = tfdata(H, 'v');
[A1,B1,C1,D1]=tf2ss(a1,b1);
g1=rank(ctrb(A1,B1));
g2=rank(obsv(A1,C1));
n = size(A1, 1);

if (g1==n || g2==n)
   disp('Given system is both fully controllable and observable')
elseif (g1==n)
   disp('Given system is fully controllable only')
elseif (g2==n)
   disp('Given system is fully observable only')
else
   disp('Given system is neither fully controllable nor observable')
end

fprintf('\n')
disp('To check controllability and obsevability in s-domain')
disp(' --- -------------------------------------------------------------–- ')

sys=d2c(H) %converting given z domain tf to s domain
[a b] = tfdata(H, 'v');
[A,B,C,D]=tf2ss(b,a);
f1=rank(ctrb(A,B));
f2=rank(obsv(A,C));
n = size(A, 1);

if (f1==n || f2==n)
   disp('Given system is both fully controllable and observable')
elseif (f1==n)
   disp('Given system is fully controllable only')
elseif (f2==n)
   disp('Given system is fully observable only')
else
   disp('Given system is neither fully controllable nor observable')
end
```

OBSERVATION:

Q1.



```
>> ControllabilityAndObs
To check controllability and observability in z-domain
 --------------------------------------------------------------------
Given characteristic equation: 5z^2 - 2z + 2 = 0

H =

  5 z^2 - 2 z + 1
  ---------------
  5 z^2 - 2 z + 2

Sample time: 0.1 seconds
Discrete-time transfer function.

Given system is both fully controllable and observable

To check controllability and obsevability in s-domain
 --------------------------------------------------------------------

sys =

  s^2 + 11.58 s + 141.6
  ---------------------
   s^2 + 9.163 s + 177

Continuous-time transfer function.

Given system is both fully controllable and observable
```

INFERENCE:

- If a continuous time system is controllable or observable, then its discrete time version, with any sampling period, is controllable or observable.
- If a continuous time system is not controllable or observable, then its discrete time version, with any sampling period, is not controllable or observable.
- Loss of controllability and/or observability occurs only in presence of oscillatory modes of the system.
- Controllability is a black and white issue. That is a system can either be completely controllable or not controllable.
- It is not always possible to observe all the states of a system in a real-life scenario, in that case, only the critical cases are considered for observability.

RESULT:

Thus, the given system was tested for controllability and observability in both s and z domain.

EXP 13: Design of State Feedback Controller and Evaluation of Closed Loop Performance

AIM:

To design a full state feedback controller for the given systems described using state-space and evaluate their closed-loop performance

APPARATUS:

S.no	Software	Version
1.	MATLAB	R2021
2.	Simulink	-

FORMULA/THEORY:

The state of a system is a minimal set of variables such that the knowledge of these variables at a particular time t_0, with the knowledge of the input(s) to the system for $t>t_0$, can completely determine the behavior of the system. The state space approach is a method for modelling, analyzing, and designing a wide range of both linear and non-linear systems. State space models use state variables to describe a system by a set of first order differential equations.

A system is represented in state space by the following equations:

$$\dot{x}(t) = Ax(t) + Bu(t) - State\ equation$$
$$y(t) = Cx(t) + Du(t) - Output\ equation$$

Where:

$x(t)$: $state\ vector$

$u(t)$: $input/control\ vector$

$x(t)$: $output\ vector$

A: $System\ matrix\ that\ relates\ how\ the\ current\ state\ affects\ the\ state\ change$

B: $Input\ or\ control\ matrix - determines\ how\ input\ affects\ system\ change$

C: $Determines\ the\ relationship\ between\ the\ system\ states\ and\ the\ system\ output$

D: $Allows\ input\ to\ directly\ affect\ the\ system\ output$

Full state feedback:

Full state feedback, or pole placement, is a method employed in feedback control system theory to place the closed-loop poles of a plant in pre-determined locations in the s-plane. This state feedback is done so that the dynamic response of the system becomes more stable. As the pole locations can be chosen by us, it is possible to get highly stable responses.

For this method to work, the system needs to be completely state controllable.

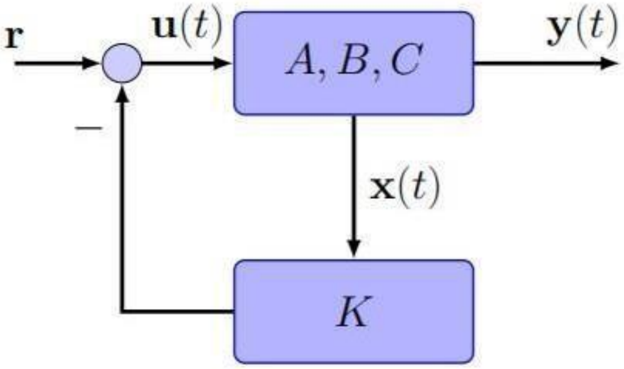
Complete state controllability:

Complete state controllability describes the ability of an external input (the vector of control variables) to move the internal state of a system from any initial state to any final state in a finite time interval.

From the theory of state space model, the matrix ‘A’ captures the dynamics of the system. Thus, a controller that we design will have to modify that matrix if we want to change the dynamic response of the system. One important result here is:

$$Eigenvalues(A) = poles\ of\ the\ system$$

Thus, if we can change the eigen values w.r.t our need, we can stabilize the system.



Assume D=0.

Now, as already stated, the x(t) vector contains all the possible states of the system. So, from the block diagram all the states of the system are fed back to the reference input via an appropriate gain matrix 'K'. Hence, this method of control is known as **full-state feedback control.**

Here:

K: Gain matrix obtained according to the position of the poles

R: reference input

U: input

In some cases, there can be an additional block before the summing point called 'Kr', which adjusts for the steady-state error of the system. Together with the blocks K, Kr and the summing point, we have a full-state feedback controller that can be designed appropriately with reference to the poles given.

State feedback design (via pole placement):

Consider the system:

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t)$$

We define a new input $u(t) = -Kx(t) + r(t)$, where K is a row vector given by:

$$K = (k0\ k1 \ldots \ldots kn{-}1)$$

This 'K' is the feedback gain matrix that we need to find, and r(t) is the new input signal. Thus,

$$u(t) = -k0x1 - k1x2 \ldots \ldots + r(t)$$

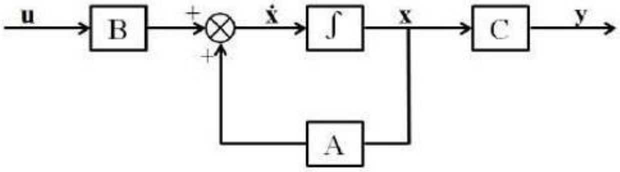
This is called state feedback. As it can be seen, we feed back to the input u(t) the states x1, x2, …., xn multiplied by the elements of the matrix K, which are called gains.

By replacing u(t) in our original equation, we get:

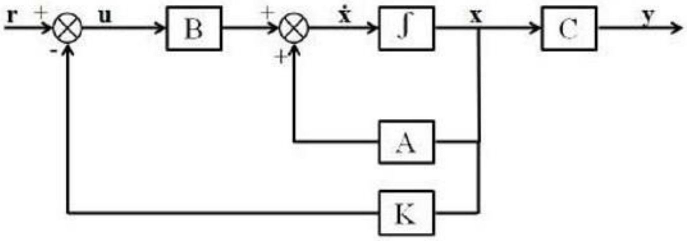
$$\dot{x}(t) = (A - BK)x(t) + Br(t)$$
$$y(t) = Cx(t)$$

As we have seen, the system stability now depends on the eigenvalues of the matrix (A - BK). Now, to design a state feedback controller, we will be given the position where the poles need to be. This means, we know the characteristic equation of the matrix (A - BK).

Using these eigenvalues, we can find the appropriate gains, i.e., the matrix K, to have desired placement of eigenvalues of the matrix (A - BK). In MATLAB, this is achieved via the place( ) function.



State space system representation without state feedback



State space system representation with state feedback

Question:

Q1. A single input system is described by the following equations:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix} = \begin{bmatrix}-1 & 0 & 0\\ 1 & -2 & 0\\ 2 & 1 & -3\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix} + \begin{bmatrix}10\\ 1\\ 0\end{bmatrix}u$$

$$Y = \begin{bmatrix}1 & 0 & 1\end{bmatrix} X$$

Design a state feedback controller which will give closed loop poles at -1±j2, -6.

From the question, we can see that the given system will have 3 state variables. The gain matrix ‘K’ can be found easily, as the closed loop poles, i.e., eigenvalues of the matrix (A – BK) is given. An additional block ‘Kr’ can be added in cascade with the reference input ‘r’ so that the steady state error of the total controlled system goes to zero.

MATLAB CODE/BLOCK DIAGRAM:

1. Code:

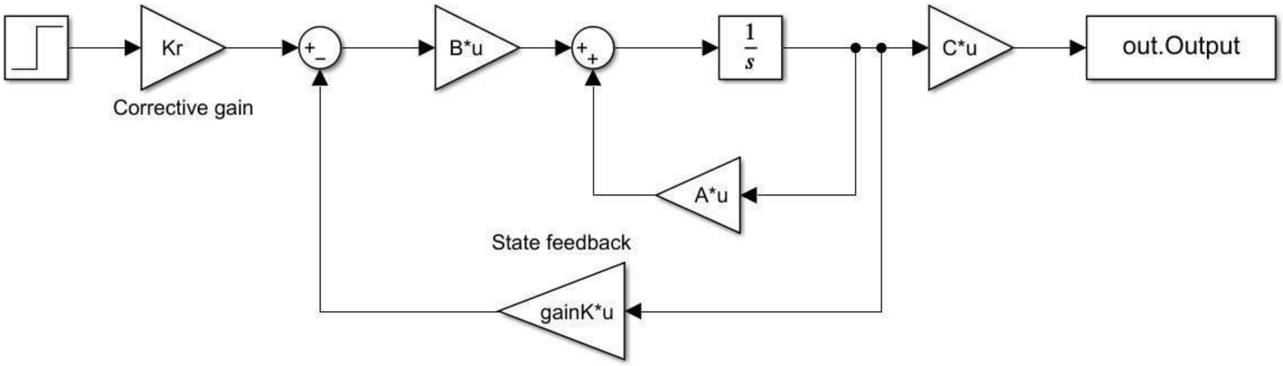
Q1.

```
>> A=[-1 0 0;1 -2 0;2 1 -3];
>> B=[10;1;0];
>> C=[1 0 1];
>> D=0;

%converts the matrices to state space form

>> sys=ss(A,B,C,D);

>> 'The eigenvectors/poles of the given system is:'

>> disp(eig(sys));

>> %pole placement

>> dp1=complex(-1,2);
>> dp2=complex(-1,-2);
>> dp3=-6;
>> dp=[dp1,dp2,dp3];
>> %Calculating Gain K

>> gainK=place(A,B,dp);

>> %new sys

>> sys_new=ss(A-B*gainK,B,C,D);
>> Kr = 1/dcgain(sys_new);
>> sys_new1=ss(A-B*gainK,B*Kr,C,D);

>> 'The eigenvectors/poles of the new state feedback-controlled system is:'

>> disp(eig(sys_new));
>> step(sys)
>> hold on
>> step(sys_new1)
>> hold on
>> plot(out.tout, out.Output, 'r')
>> legend('Without feedback','With state feedback');
```

Explanation:

- To convert given matrices to state space form, we use *ss( )* function.
- The function *eig( )* displays the eigenvalues of the system.
- The *place( )* function computes the requires gain matrix K, in accordance to the needed design parameter, i.e., the required pole placement, which is given as input.
- The *sys_new* element is the new state space description of the state-feedback controlled system.
- The constant *Kr* is computed to offset for the steady state error.
- Also, we plot the step response of the new system by transferring datapoints fromSimulink via the block *To Workspace*.

2. Simulink Diagram:



OBSERVATION:

Q1.

TABULATION:

q.no	Type of system	Rise time(s)	Settling Time(s)	Steady state error
1.	Without state feedback			
	With state feedback			

INFERENCE:

- The state feedback will work only for fully controllable systems. This is because all the states of the system are fed back to the reference input, and if the system is not controllable, then this feedback won’t properly function.

- From the step response plot for state feedback, the step response for the uncontrolled system has a higher rise time and has a negative steady-state error, i.e., the output is higher than the value of the step input. Whereas, with the state feedback, the rise time & settling time gets reduced, the steady state error also goes to zero (This is due to the corrective gain term Kr).

- Only one integrator is used in the Simulink model because all the gain terms are modelled as Matrix K*u, so that the output can be demultiplexed to obtain the different states.

- The method of pole placement is basically an upgraded version of the root locus compensator design. In root locus, we have one gain that we can alter (i.e., ‘K’), and we can move the positions of the poles only along the root locus. But here, we have a gain matrix K and we have the entire s-plane to place our poles.

RESULT:

Thus, the state feedback controller was designed accordingly, and their closed loop performance was analyzed.

EXP 14: State Observer and Evaluation of Closed Loop Performance

AIM:

To design a full state observer for the given systems described using state-space and evaluate their closed-loop performance.

APPARATUS:

S.no	Software	Version
1.	MATLAB	R2021
2.	Simulink	-

FORMULA/THEORY:

The state of a system is a minimal set of variables such that the knowledge of these variables at a particular time t_0, with the knowledge of the input(s) to the system for $t>t_0$, can completely determine the behavior of the system. The state space approach is a method for modelling, analyzing, and designing a wide range of both linear and non-linear systems. State space models use state variables to describe a system by a set of first order differential equations.

A system is represented in state space by the following equations:

$$\dot{x}(t) = Ax(t) + Bu(t) - State\ equation$$
$$y(t) = Cx(t) + Du(t) - Output\ equation$$

Where:

$x(t)$: $state\ vector$

$u(t)$: $input/control\ vector$

$x(t)$: $output\ vector$

A: $System\ matrix\ that\ relates\ how\ the\ current\ state\ affects\ the\ state\ change$

B: *Input or control matrix – determines how input affects system change*

C: *Determines the relationship between the system states and the system output*

D: *Allows input to directly affect the system output*

State Observer:

An observer is a dynamic system that is used to estimate the state of a system or someof the states of a system. A full-state observer is used to **estimate the states** of the system. The observer can be designed as either a continuous-time system or a discrete-timesystem.

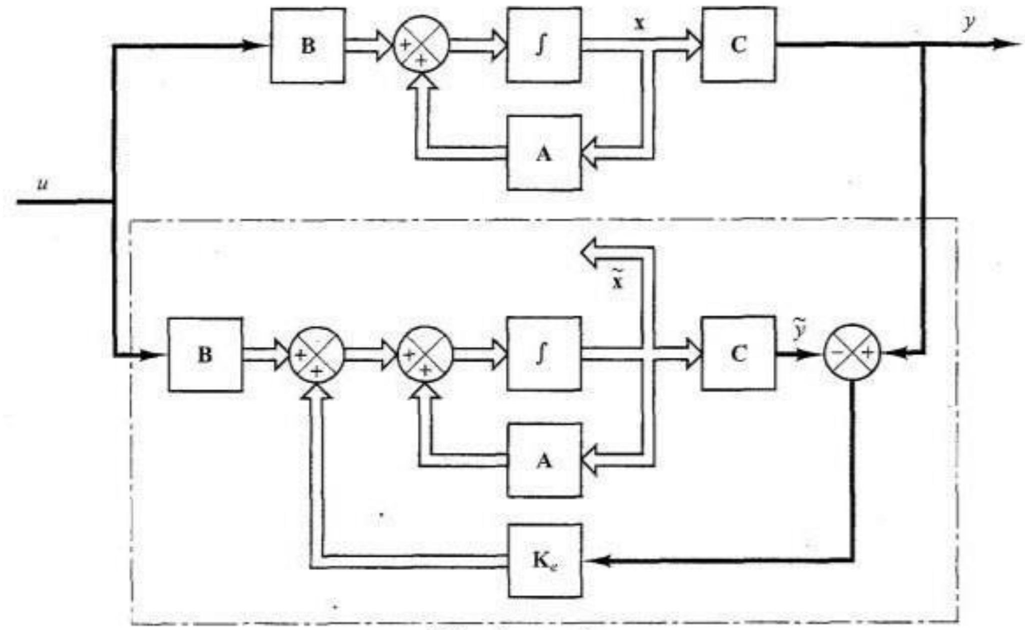
Basically, in full-state feedback, we assumed that **all the states of the system** are available for feedback. But this is not always the case. In such situations, it becomes necessaryto **estimate the unavailable states**. Estimation of unmeasurable state variables is commonly known as *observation* and adevice/program that does this is known as a **state observer**.

A **full-order state observer** is one that observes all the states of the system.

For this observer to work, the system needs to be **completely state observable**. Observability is a measure of how well all the internal states of a system can be inferred from knowledge of its external outputs.

The **observer** is a subsystem to reconstruct the state vectors of the plant (givensystem). The mathematical model of the observer is basically the same as that of the plant,except that we include an additional term that includes the estimation error to compensate for inaccuracies in matrices A and B and the lack of the initial error.

The estimation error or observation error is the difference between the measuredoutput and the estimated output.



Full-order state observer

State observer design:

As we have already seen, an observer is designed to estimate the unmeasurable states of a given system. The observer is an equivalent system with the following state space description, whose sole purpose is to estimate all the states of the system.

$$\hat{\dot{x}}(t) = A\hat{x}(t) + Bu(t)$$
$$y(t) = C\hat{x}(t)$$

To ensure that the estimation of the state variables is accurate, the observer is also fed with the error signal between the outputs, i.e., $y(t)$ − $\hat{y}(t)$, which is then multiplied with the gain matrix, L. Thus, the new system takes the form:

$$\hat{\dot{x}}(t) = (A - LC)\hat{x}(t) + Ly(t) + Bu(t)$$
$$y(t) = C\hat{x}(t)$$

The use of output feedback can be understood by considering the approximation error. Let $\epsilon(t)$ denote the error of the approximation of the state variables.

$$\dot{\epsilon}\ (t) = x(t) - \hat{x}(t)$$

Substituting for the above equation from the previous equations, we get:

$$\dot{\in}\ (t) = (A - LC) \in (t)$$

Converting to time domain:

$$e(t) = e^{(A-LC)t}e(0)$$

This above equation means that, we choose the feedback gain matrix ‘L’ such that our system equation (A - LC) becomes stable, so that e(t) goes to zero and the approximation is accurate.

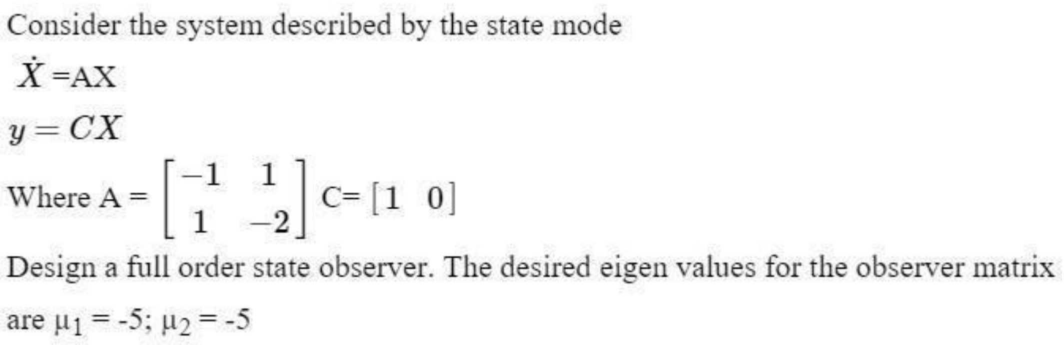
In case of designing an observer, the eigenvalues of the observer will have already been given. In that situation, with the matrices ‘A’ and ‘C’ known, and the characteristic equation of the observer system known (as eigenvalues are known), we can find the feedback gain matrix ‘L’ using linear algebra.

In MATLAB, this is done using the function acker( ) .

The quicker the approximation, better the observer system is. So, based on the value of the gain matrix L, we will get quicker or slower approximations. The process described above is same as the process adopted for the state feedback controller design. So, it can be said that both problems are mathematically the same. This property is called duality.

Question:

Q1.



Consider the system described by the state mode

$\dot{X}$ =AX

$y = CX$

Where A = $\begin{bmatrix} -1 & 1 \\ 1 & -2 \end{bmatrix}$ C= $\begin{bmatrix} 1 & 0 \end{bmatrix}$

Design a full order state observer. The desired eigen values for the observer matrix are μ_1 = -5; μ_2 = -5

- From the question, the given system will have 2 state variables.

- We will assume an initial condition of [1;1] for the system because the output of the system is zero.

- We design the state observer exactly as the given system without the initial conditions.

- The errors are calculated by feeding back the difference of the outputs of the plant and observer via a gain matrix L.

- This matrix L is found in a similar way compared to the state feedback controller.

MATLAB CODE/BLOCK DIAGRAM:

1. **Code:**

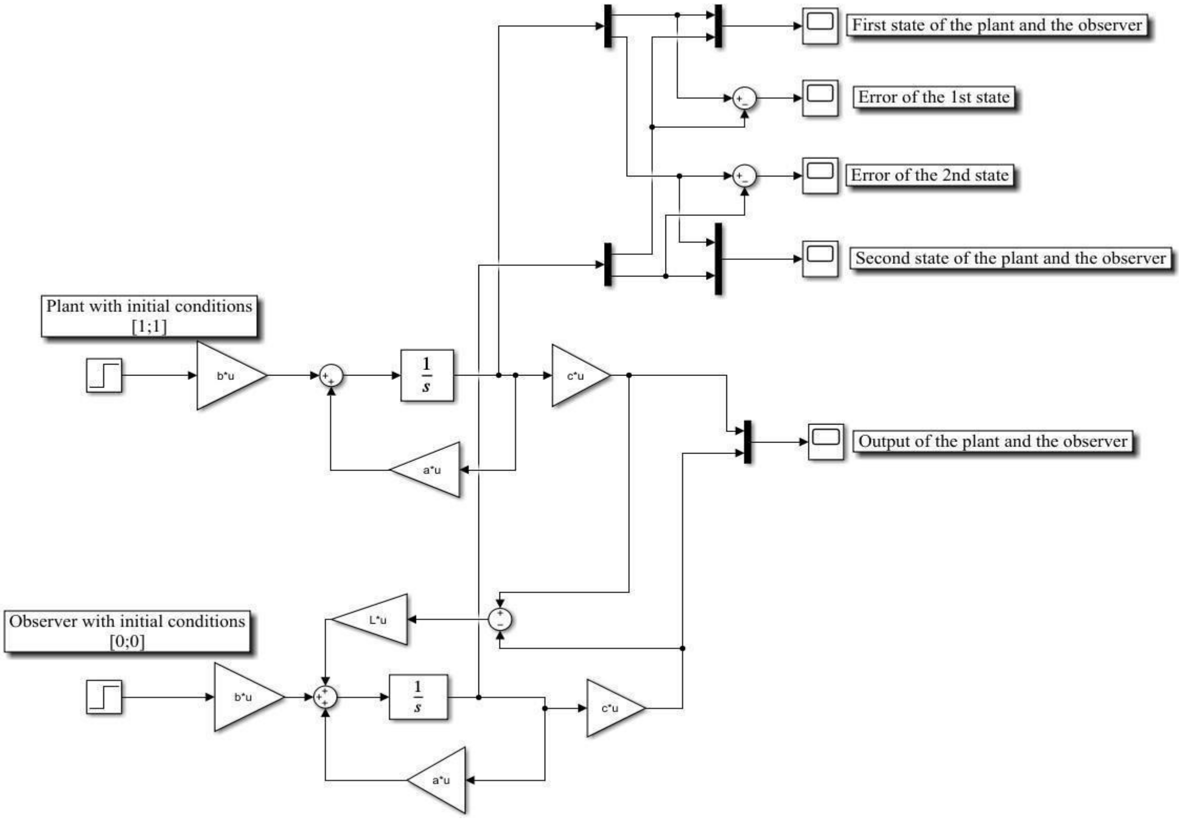
Q1.

```
>> % Original Plant

>> a=[-1 1;1 -2];
>> b=[0;0];
>> c=[1 0];
>> d=0;

>> sys=ss(a,b,c,d);
>> eig(sys);
>> rank(obsv(sys));

>> % Observer pole placement at -5 and -5
>> L = acker(a',c',[-5;-5])
```

Explanation:

- The observability of the system is checked first. Only if the system is observable, it is possible to design a state observer.
- The function acker( ) is very similar to the place( ) function. Here, as we are designing an observer, we consider transposes of the original matrices, and finally take the transpose of the total acker( ) function output. This is our required gain matrix 'L'.

2. Simulink Diagram:


First state of the plant and the observer
Error of the 1st state
Error of the 2nd state
Second state of the plant and the observer
Plant with initial conditions [1;1]
b*u
1/s
c*u
a*u
Output of the plant and the observer
Observer with initial conditions [0;0]
L*u
b*u
1/s
c*u
a*u

OBSERVATION FOR SIMULINK:

Q1.

$\boldsymbol{Plot\ 1}: \boldsymbol{x1(t)},\boldsymbol{\hat{x}1(t)}\ \boldsymbol{vs\ time}$

$\boldsymbol{Plot\ 2}:(\boldsymbol{x1(t)}-\boldsymbol{x^{\wedge}1(t)})\ \boldsymbol{vs\ time}$

$\boldsymbol{Plot\ 3}$: $\boldsymbol{x2(t)}$,$\boldsymbol{x\hat{}2(t)\ vs\ time}$

$\boldsymbol{Plot\ 4}$:$(\boldsymbol{x2(t)}-\boldsymbol{x\hat{}2(t)})\ \boldsymbol{vs\ time}$

$\boldsymbol{Plot\ 5: y(t), y^{\wedge}(t)\ vs\ time}$

INFERENCE:

- From the Simulink plots of the state observer, we can see that the observer estimatesthe states of the given system perfectly, and that is exactly the job of an observer system; to estimate the unmeasurable states of a given system.
- Plots 2 and 4 show the error signal between the original plant and the observer system. It decays to zero within 2 seconds. This means that it is a particularly good approximation, and the placement of poles/eigenvalues is good.
- Plots 1 and 3 show the actual states of the system, and the states which the observer estimates. As we have taken the plant with initial conditions, the states of the plant decay to zero from that initial condition. Here too, we can see that the observer startsto follow the state of the system very quicky (2-3 seconds). So, the observer is tracking the states of the system perfectly.
- Plot 5 shows the output of the plant and output of the observer. Similar to

the statesplot, the observer quickly starts following the output of the system y(t).

- If the system isn’t completely observable, it is not possible to design a full state observer, because some states can’t be observed, that is, it is not possible to say howthe system states vary with the input.
- From the Simulink model, the difference between the outputs of theplant and observer system is fed back to the observer via a gain matrix ‘L’. This process continuously refines the error until it becomes zero, i.e., the observer properlyestimates the states/output of the given system/plant.

RESULT:

Thus, the state observer was designed accordingly, and their closed loop performance was analyzed.

MADHA ENGINEERING COLLEGE

(A Christian Minority Institution)

KUNDRATHUR, CHENNAI – 600 069

MADHA

Expertise | Empathy | Excellence

ENGINEERING COLLEGE

Power Electronics Lab Manual

Name :

Subject :

Roll No. :

Semester : Year:

CONTENTS

Sl. No. *Name of the Experiment* *Page No.*

Ex. No: 1	**CHARACTERISTICS OF SCR**
Date:	

Aim:

To obtain and study the forward conduction mode characteristics of SCR.

Apparatus required:

S.No.	Apparatus	Rating	Quantity
01	SCR Characteristics trainer Kit	PEC 16M1A	01
02	Ammeter	(0 – 50) mA	01
03	Ammeter	(0 – 20) mA	01
04	Voltmeter	(0 – 30)V	01
05	Patch cord		As required

Theory:

Introduction:

- Thyristors is a general term which includes devices such as SCR, TRIAC, GTO, LASCR RCT… and many more.
- Thus thyristors is a family of device. SCR i.e**. Silicon Controlled Rectifier** is the most popular device of the thyristor family.
- SCRs have the **highest power handling capacity** of all the power semiconductor devices.
- They have a four layer (pnpn), three junction (J_1,J_2, & J_3) with three user accessible terminals (Anode, Cathode & Gate).
- SCR is a **latching type device** that can be turned on by the control terminal (gate) but once turned on, the gate loses control on it. (i.e. it cannot be turned off by the gate)
- It is a **current control device**.

V - I Characteristics:

- SCR has three basic mode of operation
    1. Reverse blocking mode (Off – state)
    2. Forward blocking mode (Off – state)
    3. Forward conduction mode (On – state)

1. **Reverse blocking mode:**

- When cathode is made positive with respect to anode, the SCR is said to be reversed biased.
- Junctions J_1, J_3 are reversed biased whereas junction J_2 is forward biased.
- For a small reverse voltage a small reverse leakage current of the order of a few mA or a few µA flows until the avalanche break down takes place at reverse breakdown voltage V_{BR}.
- As soon as the reverse breakdown takes place due to avalanche breakdown, a large current flows through SCR whereas the voltage across the device remains constant.
- The region from 0 volt upto V_{BR} volt in which the SCR is reverse biased and non – conducting is called as **"Reversed blocking state".**

2. **Forward blocking mode:**

- When anode is positive with respect to the cathode, with gate circuit open, SCR is said to be forward blocking mode.
- Junctions J_1, J_3 are forward biased J_2 is reverse biased.
- In this mode, a small leakage current flows due to thermally generated minority carriers.
- In case the forward voltage is increased, then the reversed biased junction J_2 will have an avalanche breakdown at a voltage called forward breakover voltage $V_{BO.}$
- When forward voltage is less than V_{BO}, SCR offers high impedance. Therefore a SCR can be treated as an open switch in a forward blocking mode.

3. **Forward conduction mode:**

- A SCR is brought from forward blocking mode to forward conduction mode by applying a gate pulse between gate and cathode.
- In this mode, SCR is in on state and behaves like a closed switch.
- Voltage drop across SCR in the on-state is of the order of one to few volts.
- In the on-state, the anode current is limited by an external impedance or resistance.

Important terms

1. **Forward breakover voltage (V_{BO})**

- This is the maximum forward voltage that can be applied between anode and cathode without initiating forward conduction.
- This voltage is defined for a zero gate current ($I_G = 0$). In short this is the maximum forward voltage across SCR in its off state.

2. **Holding current (I_H)**

- It represents the minimum current that can flow through SCR and still "hold" it in the on state.
- If the forward anode current is reduced below holding current, SCR will be turned off. The holding current is defined for zero gate current ($I_G = 0$).

3. Latching current (I_L)

- It is minimum anode current that must flow through SCR to latch it into the on state.
- The latching current is higher than the holding current. The latching current is important when SCR is turned on.

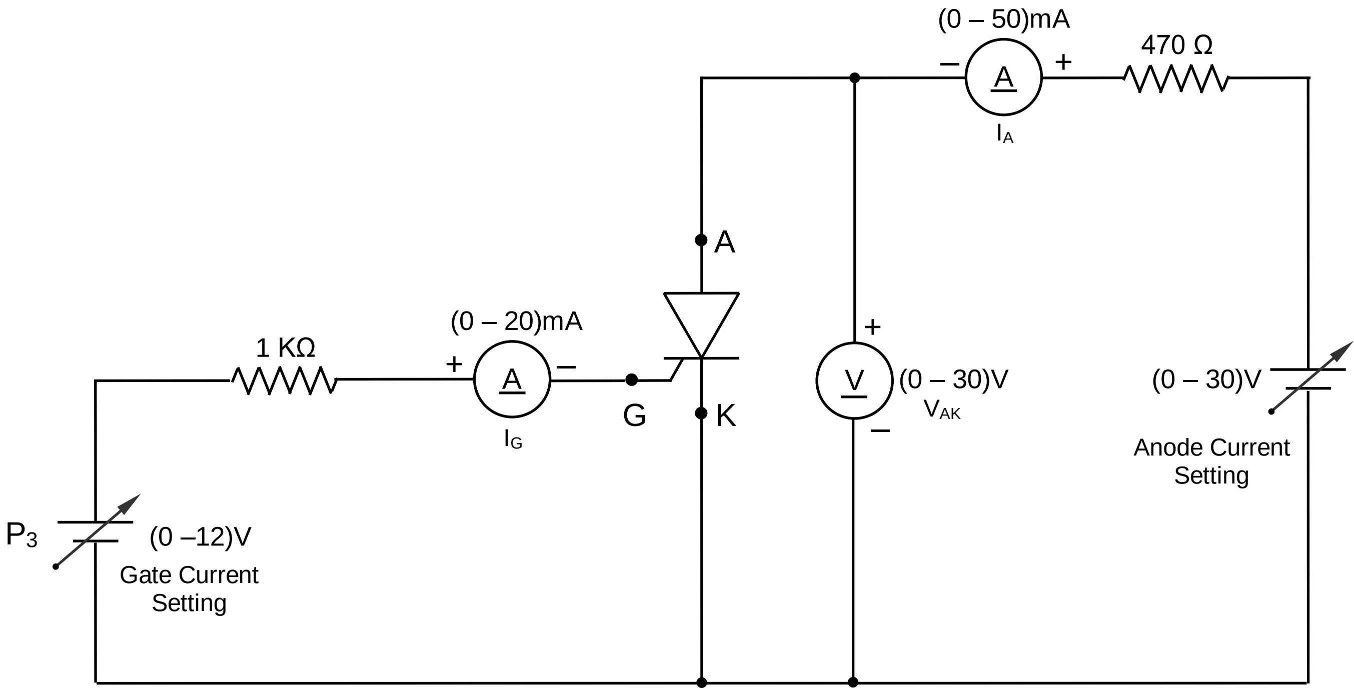
Connection Procedure:

- Connect the anode, cathode and gate terminals to SCR characteristics kit.
- Connect the ammeter, voltmeter in the respective channels in the circuit to measure the gate current (I_G), anode current (I_A) and anode cathode voltage (V_{AK}).

Experimental Procedure:

- Switch on the trainer kit; keep the gate current I_G at some certain value say 5 mA.
- Now slowly increase the anode - cathode voltage V_{AK} by varying the Pot (P_4) till the thyristor get turned on, note down I_A (Latching Current), V_{AK}.
- To find out the break over voltage (V_{BO}) increase V_{AK} and note I_A.
- Now reduce V_{AK} till the thyristor turned OFF and measure the holding current (I_H)
- Repeat the procedure for various gate current take the reading and tabulate it.

CIRCUIT DIAGRAM FOR V - I CHARACTERISTIC OF SCR



Connection Procedure:

1. Connect the anode, cathode and gate terminals to SCR characteristics circuit.
2. Connect the ammeter, voltmeter in the respective channels in the circuit to measure the gate current, anode current and anode cathode voltage V_{AK}.

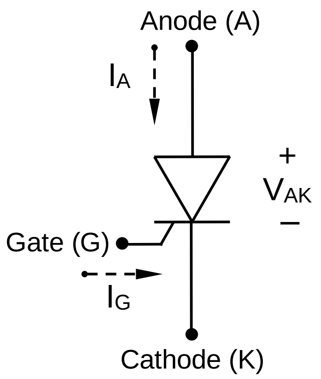
Experimental Procedure

1. Switch on the trainer kit; keep the gate current I_G at some certain value say 5 mA.
2. Now slowly increase the anode- cathode voltage V_{AK} by varying the Pot till the thyristor get turned on, note down I_A, $V_{AK.}$
3. Now find out the break over voltage (V_{BR}) latching current (I_L). Further increase V_{AK} and note $I_{A.}$
4. Now reduce V_{Ak} till the thyristor turn OFF and measure the holding current (I_H)
5. For various gate current take the reading and tabulate it.

Tabulation:

S.No.	I_G = . (mA)		I_G = . (mA)	
	V_{AK} (V)	I_A (mA)	V_{AK} (V)	I_A (mA)
01				
02				
03				
04				
05				
06				
07				
08				

Circuit Symbol for SCR

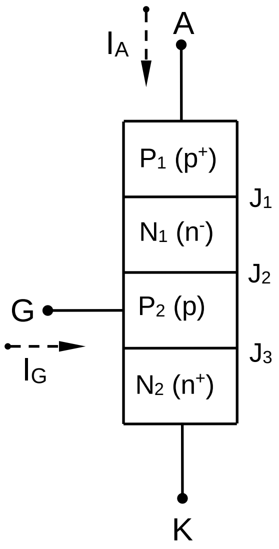


V_{AK} - Anode to Cathode Voltage

I_A - Anode Current

I_G - Gate Current

Basic Structure of SCR



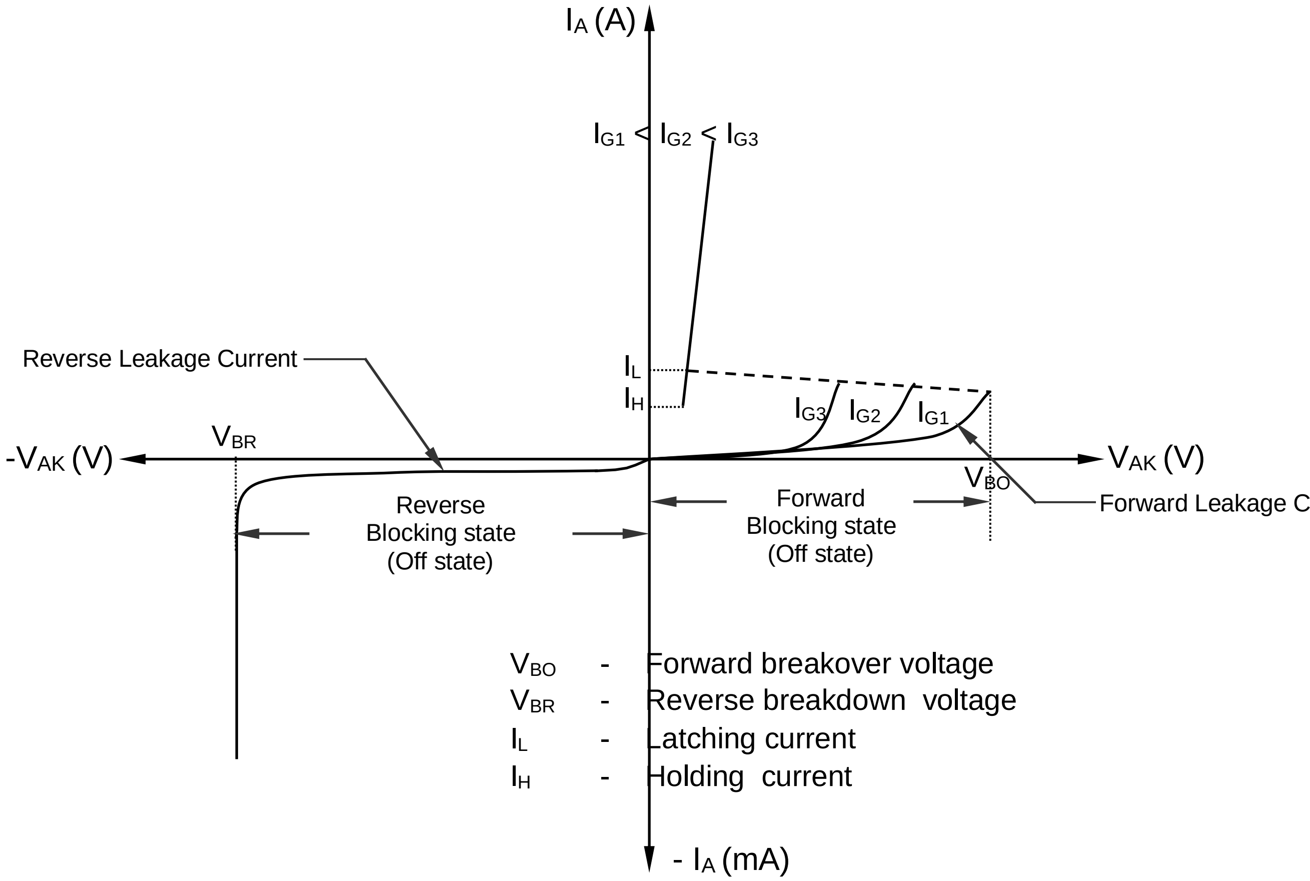
p^+ - Heavily Doped Layer

p - Moderately Doped Layer

n^+ - Heavily Doped Layer

n^- - Lightly Doped Layer

Model Graph



V_{BO} - Forward breakover voltage

V_{BR} - Reverse breakdown voltage

I_L - Latching current

I_H - Holding current

V - I Characteristics of SCR

Result: Thus the V-I characteristics of SCR was drawn.

Ex. No: 2	**CHARACTERISTICS OF TRIAC**
Date:	

Aim:

To draw and study the I-V characteristics of TRIAC.

Apparatus Required

S.No	Apparatus	Rating	Quantity
01	TRIAC Characteristics trainer Kit	PEC 16M1C	01
02	Ammeter	(0 – 100) mA	01
03	Ammeter	(0 – 50) mA	01
04	Voltmeter	(0 – 30)V	01
05	Patch cord		As required

Theory:

Introduction:

- A TRIAC (**TRI**ode + **AC**) is a bidirectional device i.e. a device which allows current to flow though it in either directions.
- In the SCR family, after the SCR, TRIAC is the most widely used device for power control.
- TRIAC is a 3 terminal device, with terminals named as main terminals 1, 2 (MT_1, MT_2) and gate, out of which gate is the control terminal.
- Its operation is equivalent to two SCRs connected in anti - parallel.
- The TRIAC can be turned on by applying either a positive or negative voltage to the gate with respect to terminal MT_1.
- As the two conducting paths from MT_1 to MT_2 or from MT_2 to MT_1 interact with each other in the structure of the TRIAC, their voltage, current and frequency ratings are much lower as compared with conventional SCR.
- TRIAC are used extensively in residential lamp dimmers, heat control and for the speed control of small AC motors.

I - V Characteristics:

- TRIAC has the following mode of operation
  4. Forward & Reverse blocking mode (Off – state)
  5. Forward & Reverse conduction mode (On – state)
- Depending on the polarity of voltage applied between its MT_2 and MT_1 terminals, it will operate in either the first quadrant or third quadrant.

MT_2 positive w.r.t. MT_1 : Operation is in the first quadrant

MT_2 negative w.r.t. MT_1 : Operation is in the third quadrant

Forward blocking mode: (MT_2 positive with respect to MT_1)

- When a forward voltage less than the breakover voltage V_{BO} is applied with gate terminal open, the TRIAC can successfully block the forward voltage without getting turned on.

Reverse blocking mode: (MT_1 positive with respect to MT_2)

- When reverse voltage is less than the breakover voltage $-V_{BO}$ with gate open circuited, the TRIAC will block the reverse voltage without getting turned on.

Forward conduction mode:

- A TRIAC is brought from forward blocking mode to forward conduction mode by applying a positive gate pulse w.r.t. MT_1,
- TRIAC is rarely operated in first quadrant with negative gate current.

Reverse conduction mode:

- A TRIAC is brought from Reverse blocking mode to Reverse conduction mode by applying a negative gate pulse w.r.t. MT_1,
- TRIAC is rarely operated in third quadrant with positive gate current.

Connection Procedure:

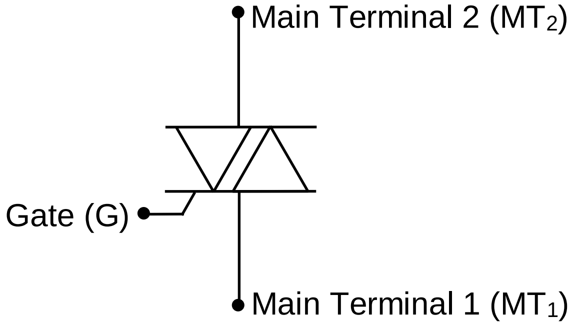
- Connect terminals P_1 & P_3 and P_2 & P_4.
- Connect a ammeter with a range of (0 – 100)mA between P_5 and P_7 terminals.
- Connect a voltmeter with a range of (0 – 30)V between P_8 and P_9 terminals.
- Connect another ammeter with a range of (0 – 50)mA between P_{10} and P_{11} terminals.
- Connect terminals P_{12} & P_{14} and P_{13} & P_{15}.
- Don’t short P_1, P_2, & P_{14}, P_{15}.

Experimental Procedure:

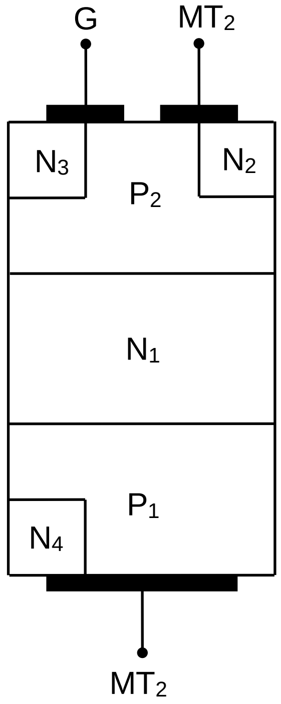
- Switch ON the 230 V AC supply.
- Keep the terminal MT_2 positive with respect to MT_1.
- Now slowly vary the POT (Volt. Adjust) for obtaining maximum voltage across MT_1 MT_2 terminal (V_{MT1MT2}) till the TRIAC is turned ON.
- Moment the TRIAC is turned on note down the value of Voltage across Main terminals (V_{MT1MT2}) i.e. Breakover voltage V_{BO} and Current I_{TRIAC} i.e. Latching current I_L .
- Reset the Main terminal voltage ($V_{MT1MT2} = 0V$).

- Vary the POT (Source) and set the gate current (I_G).
- Now slowly vary the POT (Volt. Adjust) and measure and tabulate V_{MT1MT2}, & I_{TRIAC}.
- Plot the graph for V_{MT1MT2} Vs I_{TRIAC}.
- Connect the MT_2 and gate terminal of TRIAC to negative supply w.r.t. MT_1.
- Repeat the same procedure.
- Tabulate the reading and plot the graph for V_{MT1MT2} Vs I_{TRIAC} in a graph sheet.

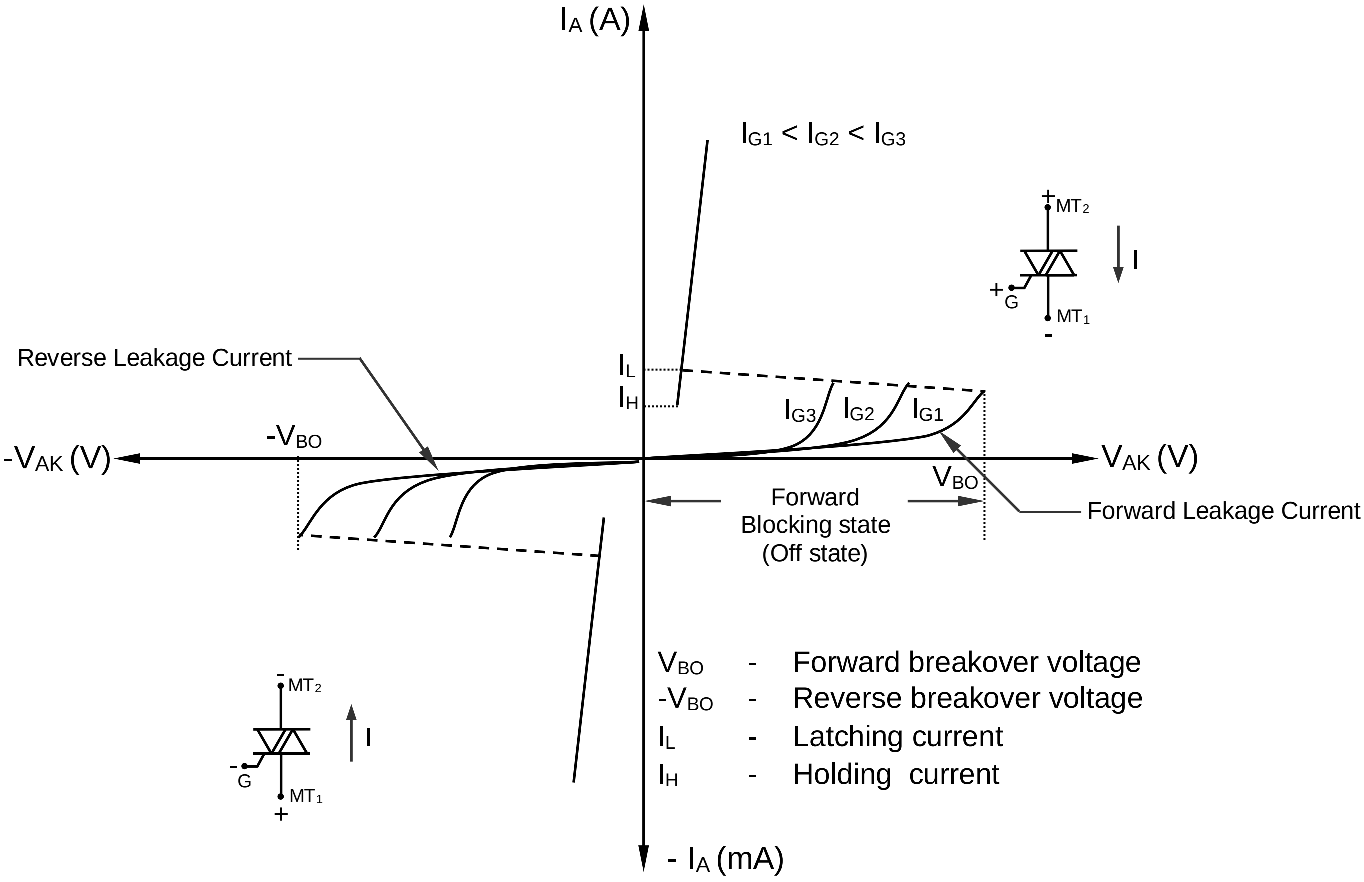
Circuit symbol for TRIAC



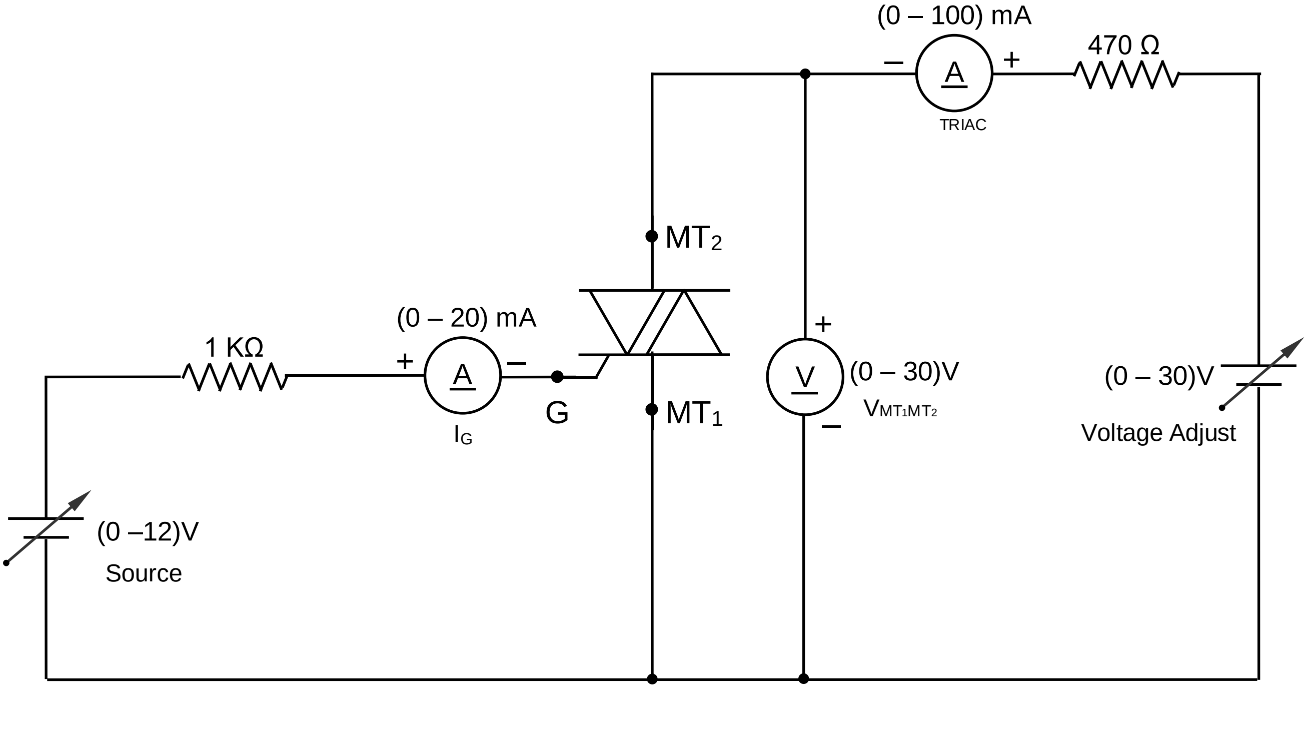
Basic structure of TRIAC



Model Graph



CIRCUIT DIAGRAM FOR V - I CHARACTERISTIC OF TRIAC



Tabulation:

	I Quadrant		III Quadrant	
S.No.	**I_G = . (mA)**		**I_G = . (mA)**	
	V_{MT1MT2} (V)	**I_{TRIAC} (mA)**	**V_{MT1MT2} (V)**	**I_{TRIAC} (mA)**
01				
02				
03				
04				
05				
06				
07				
08				
09				

Result:

Thus the V-I characteristics of TRIAC was studied and graph was drawn

Ex. No: 3	**CHARACTERISTICS OF MOSFET**
Date:	

Aim:

To draw and study the characteristics of MOSFET.

Apparatus Required:

S.No	Apparatus	Rating	Quantity
01	MOSFET Characteristics trainer Kit	VPET 202A	01
02	Ammeter	(0 – 100) mA	01
03	Voltmeter	(0 – 30) V	02
04	Patch cord		As required

Theory:

Introduction:

- The long form of MOSFET is **Metal Oxide Semiconductor Field Effect Transistor.**
- Power MOSFETs are capable of **switching at very high frequencies** upto 100 KHz
- It is a **voltage controlled device**.
- It is a **Unipolar device** because its operation depends upon the flow of majority carrier only.
- It is a three terminal device namely Drain (D), source (S) and gate (G), out of which gate is a control terminal of the device.
- Due to the presence of the SiO_2 layer between gate terminal and n-type channel, the input impedance of MOSFET is very high.
- It is used **in low power high frequency application**.
- In order to increase the voltage blocking rating of the power MOSFET, an n^- drift layer is included.
- The **second breakdown does not take place** in power MOSFETs.
- It is easy to turn on and off the power MOSFETs.
- Power MOSFETs have asymmetric voltage blocking capacity. They can block a high forward voltage but they cannot block high reverser voltage.
- On state voltage drop across the MOSFET is high in the saturation region.

Working of power MOSFET:

- The MOSFET is forward biased by connecting a positive voltage to its drain terminal with respect to source terminal and the gate is made positive with respect to the body.
- Due to positive voltage applied between gate and body electrons are attracted towards the gate and gather below the silicon layer and forms n channel between two n^+ regions.
- This channel allows the current flow to the load.
- The MOSFET now acts as a variable resistor where the resistance of the channel is dependent on the magnitude of gate to source voltage $V_{GS.}$
- Drain current (I_D) is dependent on gate – source voltage $V_{GS.}$

I - V Characteristics:

- The two characteristics of MOSFET are
    1. Static characteristic (Output characteristics)
    2. Transfer characteristic

Static characteristic: (Output characteristics)

- They are drawn between drain current I_D and drain source voltage V_{DS} with gate to source voltage V_{GS} as a variable parameter.
- For low values of V_{DS} the graph between I_D – V_{DS} is almost linear.
- For given value of V_{GS} if V_{DS} is increased, output characteristic is relatively flat, indicating that drain current is nearly constant.
- The three region of operation of power MOSFET are
    1. Cut-off region
    2. Saturation region
    3. Ohmic region
- The gate to source voltage (V_{GS}) should be large enough to drive the MOSFET into Ohmic region, otherwise the MOSFET will operate in the saturation region which is not desired.
- In the Ohmic region $V_{DS\ ON,}$ is small and the MOSFET acting as a closed switch.

Transfer characteristic:

- Transfer characteristic shows the variation of drain current I_D as a function of gate – source Voltage $V_{GS.}$
- From this characteristic we can find the threshold voltage V_{GST}. It is an important parameter of MOSFET.
- V_{GST} is the minimum positive voltage between gate and source to induce n-channel. Thus for gate – source voltage below V_{GST}, device is in off-state.

Connection Procedure:

- Connect the connector P_{20} & P_6, P_{22} & P_{10} and P_{21} & $P_{9.}$
- Connect the connectors P_1 & P_4, P_2 & P_5, P_{12} & P_{14} and P_{13} & P_{15}.
- Don't short P_1 & P_2 and P_{14} & $P_{15.}$
- Connect an ammeter with a range of (0 - 100) mA across connectors P_8 & $P_{11.}$
- Connect a voltmeter (V_{GS}) with a range of (0 – 30) V across connectors P_6 & $P_{7.}$
- Connect another voltmeter (V_{DS}) with a range of (0 – 30) V across connectors P_9 & $P_{10.}$

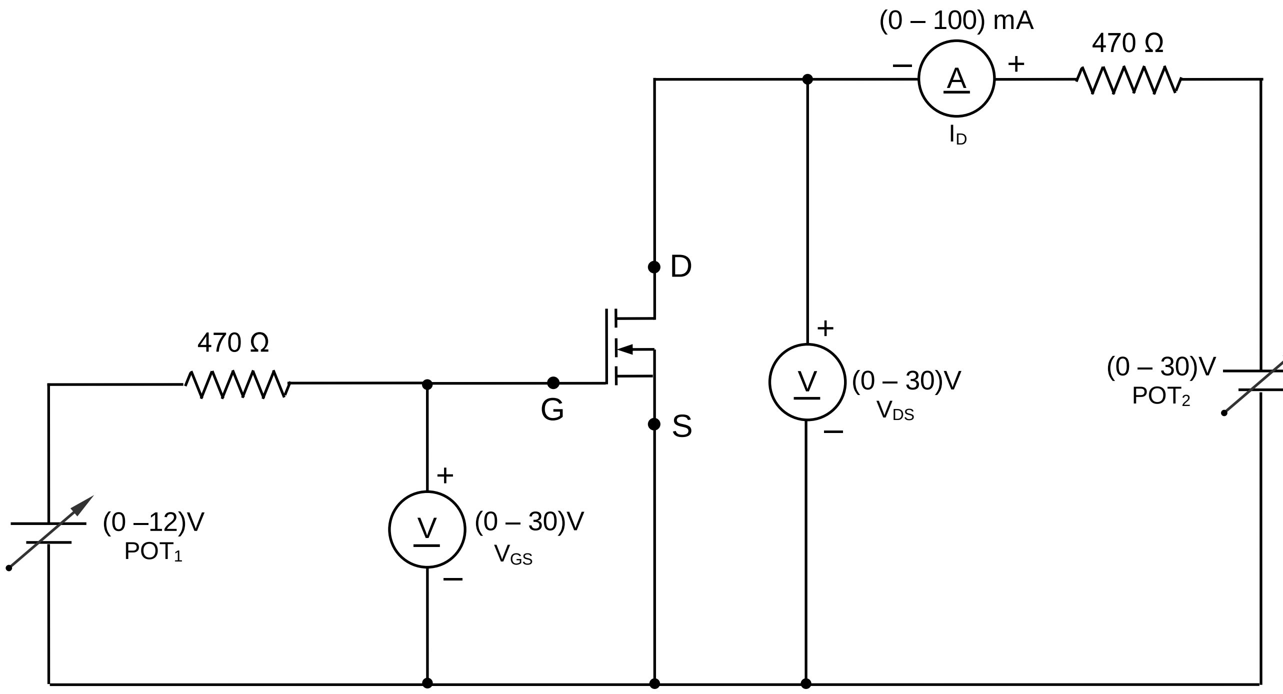
Experimental Procedure: Output Characteristics

- Verify the connection as per the connection procedure.
- Switch ON the power supply.
- Switch ON the power ON/OFF switch.
- Switch ON the switches SW_1 and $SW_{2.}$
- Set the Gate – Source (V_{GS}) voltage at a particular value by varying $POT_{1.}$
- Smoothly vary the (POT_2), Drain – Source (V_{DS}) voltage till the MOSFET gets turn on and note down the voltmeter V_{DS} and Ammeter I_D values.
- Further increase the Drain – Source (V_{DS}) voltage and note down the readings V_{DS} and I_D.

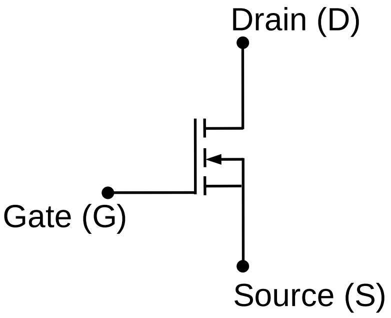
Experimental Procedure: Transfer Characteristics

- Verify the connection as per the connection procedure.
- Switch ON the power supply.
- Switch ON the power ON/OFF switch.
- Switch ON the switches SW_1 and $SW_{2.}$
- Set the Drain – Source (V_{DS}) voltage at a particular value by varying $POT_{2.}$
- Smoothly vary the (POT_1), Gate – Source (V_{DS}) voltage till the MOSFET gets turn on and note down the voltmeter V_{GS} and Ammeter I_D values.
- Further increase the Gate – Source (V_{GS}) voltage and note down the readings V_{GS} and I_D.

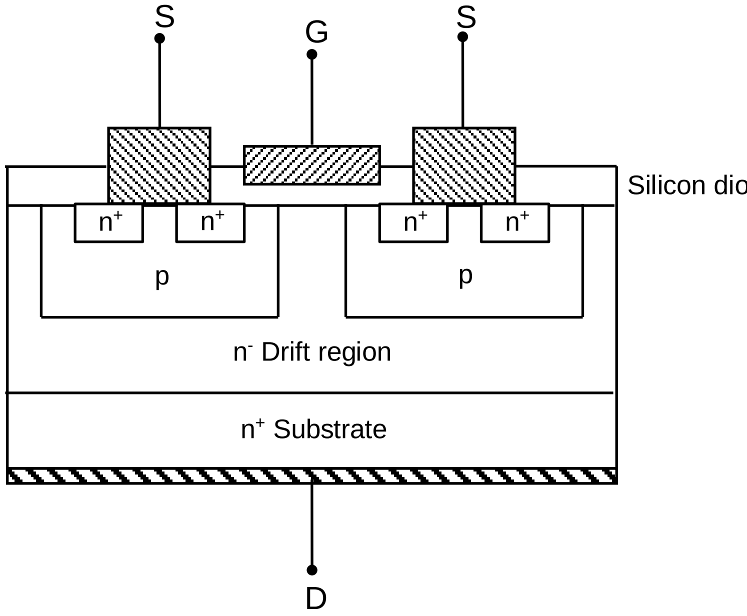
CIRCUIT DIAGRAM FOR CHARACTERISTIC OF MOSFET



Circuit symbol for MOSFET



Basic structure of MOSFET



Tabulation: Output characteristics of MOSFET

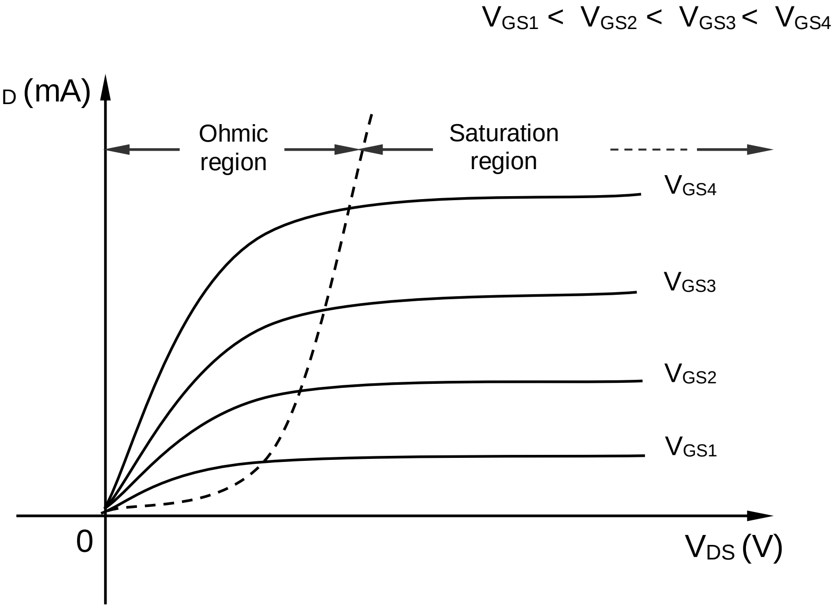
S.No.	V_{GS} = .(V)		V_{GS} = .(V)		V_{GS} = .(V)	
	V_{DS} (V)	I_D (mA)	V_{DS} (V)	I_D (mA)	V_{DS} (V)	I_D (mA)
01						
02						
03						
04						
05						
06						
07						
08						
09						

Tabulation: Transfer characteristics of MOSFET

S.No.	V_{DS} = .(V)	
	V_{GS} (V)	I_D (mA)
01		
02		
03		
04		
05		
06		
07		
08		
09		

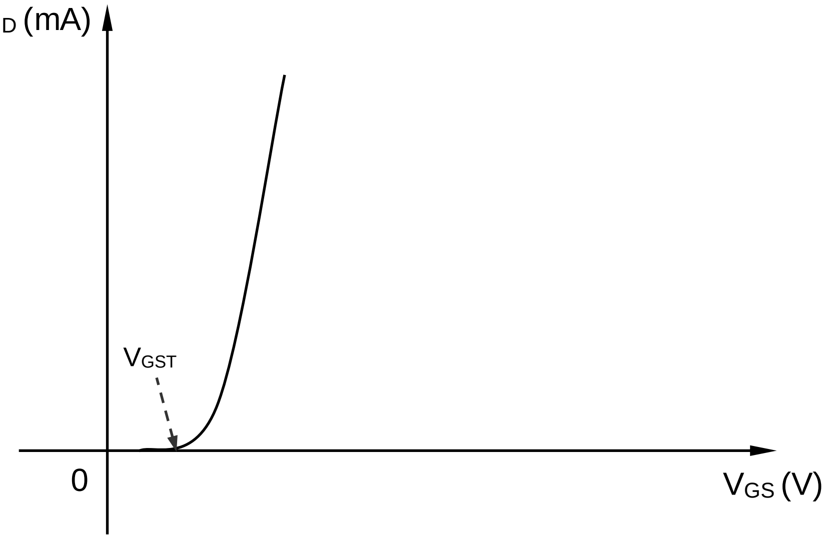
p

Model Graph



Output V - I Characteristics of Power MOSFET

Transfer Characteristics of Power MOSFET



Result: Thus, the characteristics of MOSFET was studied and the graph is plotted.

Ex. No: 4	CHARACTERISTICS OF IGBT
Date:	

Aim:

To draw and study the characteristics of IGBT.

Apparatus Required:

S.No	Apparatus	Rating	Quantity
1.	MOSFET Characteristics trainer Kit	VPET 202A	01
2.	Ammeter	(0 – 100) mA	01
3.	Voltmeter	(0 – 30) V	02
4.	Patch cord		As required

Theory:

Introduction:

- The long form of IGBT is **Insulated Gate Bipolar Transistor**.
- IGBT has been developed by combining into it the best qualities of both BJT and MOSFET.
- Thus an IGBT possesses high input impedance like a MOSFET and has low on state power loss as in a BJT.
- IGBT is free from second breakdown problem present in BJT.
- It has a higher switching frequency than that of a power BJT but lower than MOSFET.
- The three terminal of the device is named as Collector (C), Emitter (E) and Gate (G), gate being the control terminal.
- It is a voltage control device and requires continues gate – emitter voltage V_{GE}
- This device also uses the n^- drift region which improves its breakdown voltage capacity.
- The symbol of IGBT indicates that it has input characteristics similar to MOSFET and output characteristics similar to BJT.
- IGBT can handle larger power than MOSFETs
- IGBT can be easily connected in parallel.
- IGBT do not need snubber circuits for their protection.
- But It has asymmetric blocking capacity. It cannot block high reverse voltage.

Working of power IGBT:

- When collector is made positive with respect to emitter IGBT gets forward biased.

- With no voltage between gate and emitter, two junctions between n^- region and p region are reverse biased, so no current flows from collector to emitter.
- When gate - emitter voltage more than the threshold voltage V_{GET} of IGBT, an n channel or inversion layer, is formed in the upper part of p region just beneath the gate.
- This n – channel short circuits the n^- region with n^+ region as a result IGBT gets turned on and begins to conduct forward current.

I - V Characteristics:

- The two characteristics of IGBT are

  3. Static characteristic (Output characteristics)
  4. Transfer characteristic

Static characteristic: (Output characteristics)

- It is drawn between collector current I_C and collector - emitter voltage V_{CE} with gate to emitter voltage V_{GE} as a variable parameter.
- The shape of the output characteristics is similar to that of BJT. But the controlling parameter is gate – emitter voltage V_{GE}.
- The collector current I_C increases with increase in the voltage between gate and source (V_{GS})
- The forward breakdown voltage is the voltage at which the avalanche breakdown takes place.
- At this point, both, voltage across the device and current through it are high. Hence power dissipated in the device will be very large and will damage it.
- Therefore device must be operated below forward breakdown voltage.

Transfer characteristic:

- Transfer characteristic shows the variation of drain current I_C as a function of gate – emitter voltage V_{GE}
- From this characteristic we can find the threshold voltage V_{GET}. It is an important parameter of IGBT.
- V_{GET} is the minimum positive voltage between gate and emitter to induce n-channel. Thus for gate – emitter voltage below V_{GET}, device is in off-state.

Connection Procedure:

- Connect the connectors P_{17} & P_6, P_{19} & P_{10} and P_{18} & $P_{9.}$
- Connect the connectors P_1 & P_4, P_2 & P_5, P_{12} & P_{14} and P_{13} & P_{15}.
- Don't short P_1 & P_2 and P_{14} & $P_{15.}$
- Connect an ammeter with a range of (0 - 100) mA across connectors P_8 & $P_{11.}$
- Connect a voltmeter (V_{GE}) with a range of (0 – 30) V across connectors P_6 & $P_{7.}$
- Connect another voltmeter (V_{CE}) with a range of (0 – 30) V across connectors P_9 & $P_{10.}$

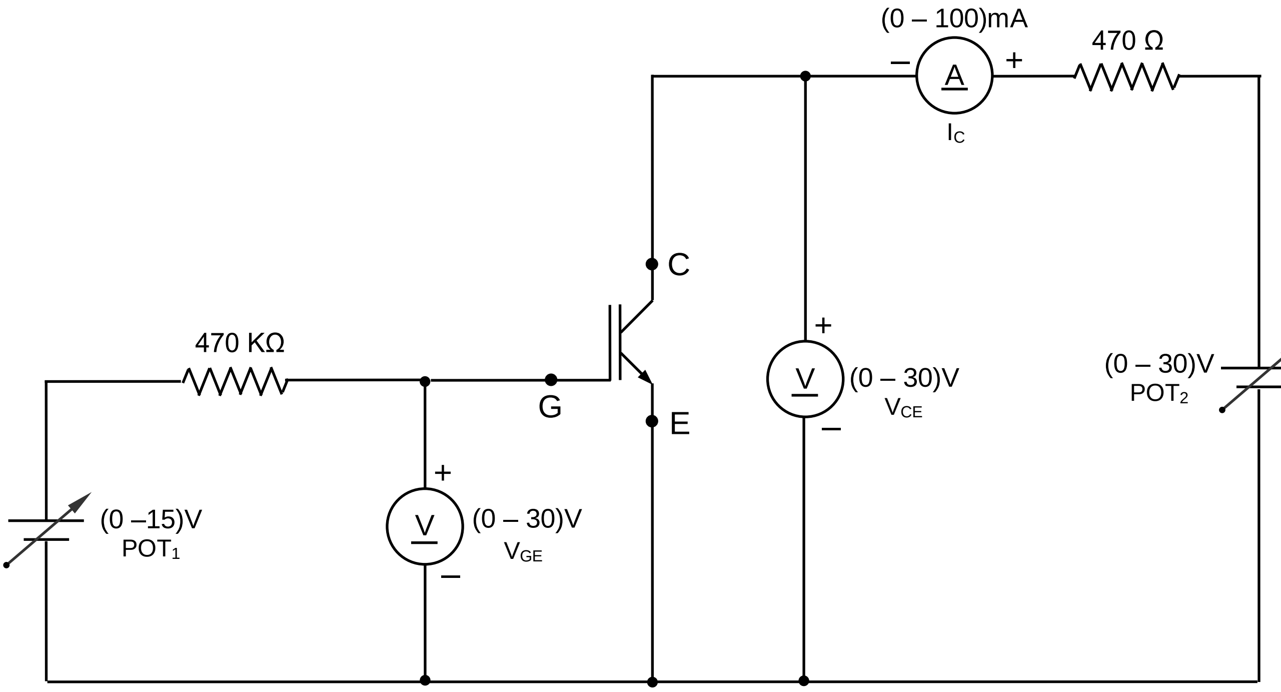
Experimental Procedure: Output Characteristics

- Verify the connection as per the connection procedure.
- Switch ON the power supply.
- Switch ON the power ON/OFF switch.
- Switch ON the switches SW_1 and $SW_{2.}$
- Set the Gate – Source (V_{GE}) voltage at a particular value by varying $POT_{1.}$
- Smoothly vary the (POT_2), Drain – Source (V_{CE}) voltage till the MOSFET gets turn on and note down the voltmeter V_{CE} and Ammeter I_C values.
- Further increase the Drain – Source (V_{CE}) voltage and note down the readings V_{CE} and I_C.

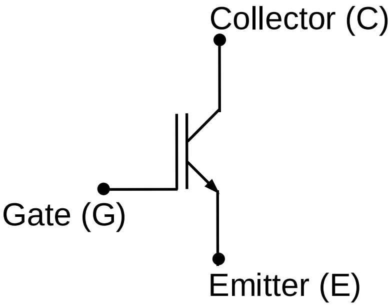
Experimental Procedure: Transfer Characteristics

- Verify the connection as per the connection procedure.
- Switch ON the power supply.
- Switch ON the power ON/OFF switch.
- Switch ON the switches SW_1 and $SW_{2.}$
- Set the Drain – Source (V_{CE}) voltage at a particular value by varying $POT_{2.}$
- Smoothly vary the (POT_1), Gate – Source (V_{CE}) voltage till the MOSFET gets turn on and note down the voltmeter V_{GE} and Ammeter I_C values.
- Further increase the Gate – Source (V_{GE}) voltage and note down the readings V_{GE} and I_C.

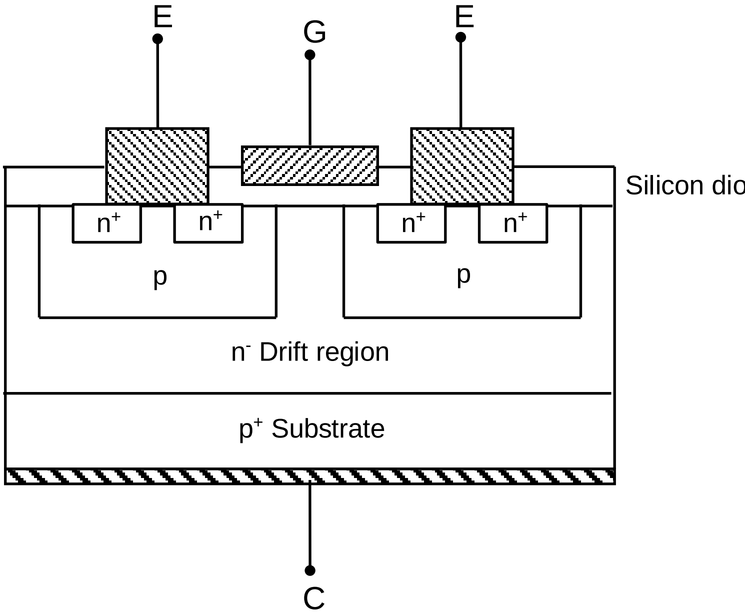
CIRCUIT DIAGRAM FOR CHARACTERISTIC OF IGBT



Circuit symbol for IGBT



Basic structure of IGBT



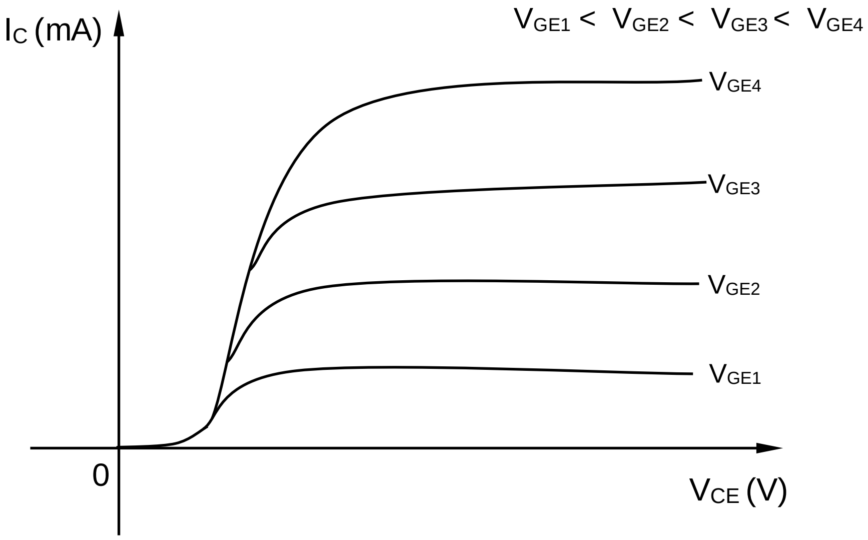
Tabulation: Output characteristics of IGBT

S.No.	V_{GE} = .(V)		V_{GE} = .(V)		V_{GE} = .(V)	
	V_{CE} (V)	I_C (mA)	V_{CE} (V)	I_C (mA)	V_{CE} (V)	I_C (mA)
01						
02						
03						
04						
05						
06						
07						
08						
09						

Tabulation: Transfer characteristics of IGBT

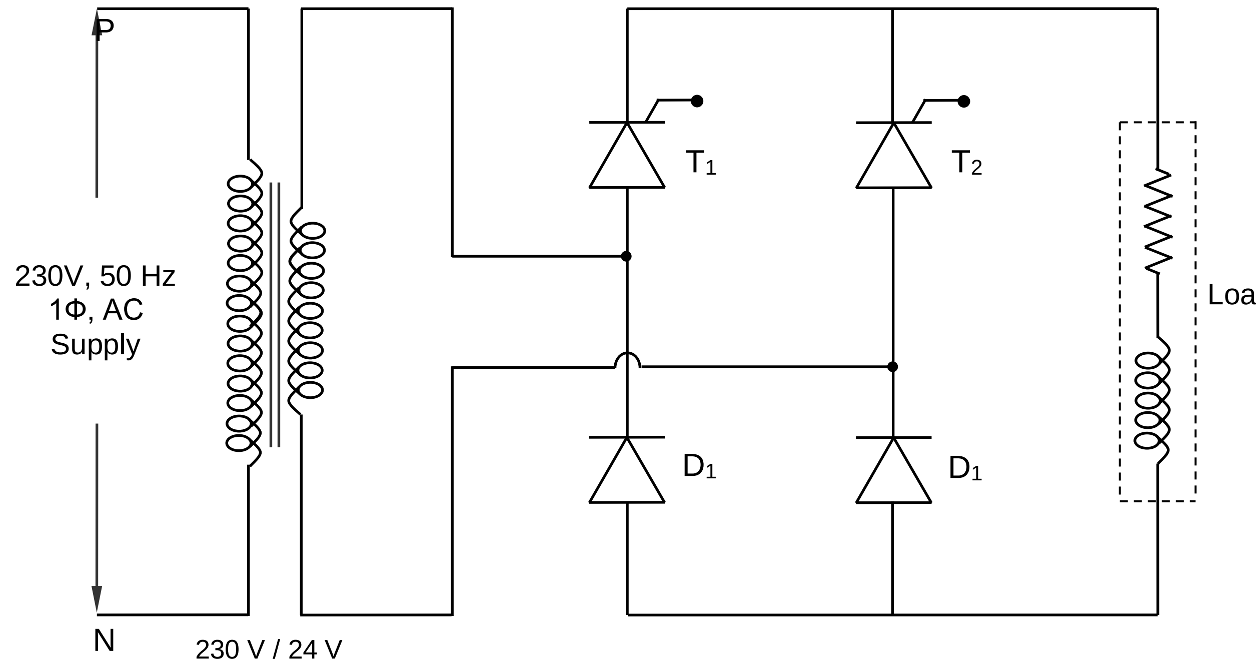
S.No.	V_{CE} = . (V)	
	V_{GE} (V)	I_C (mA)
01		
02		
03		
04		
05		
06		
07		
08		
09		

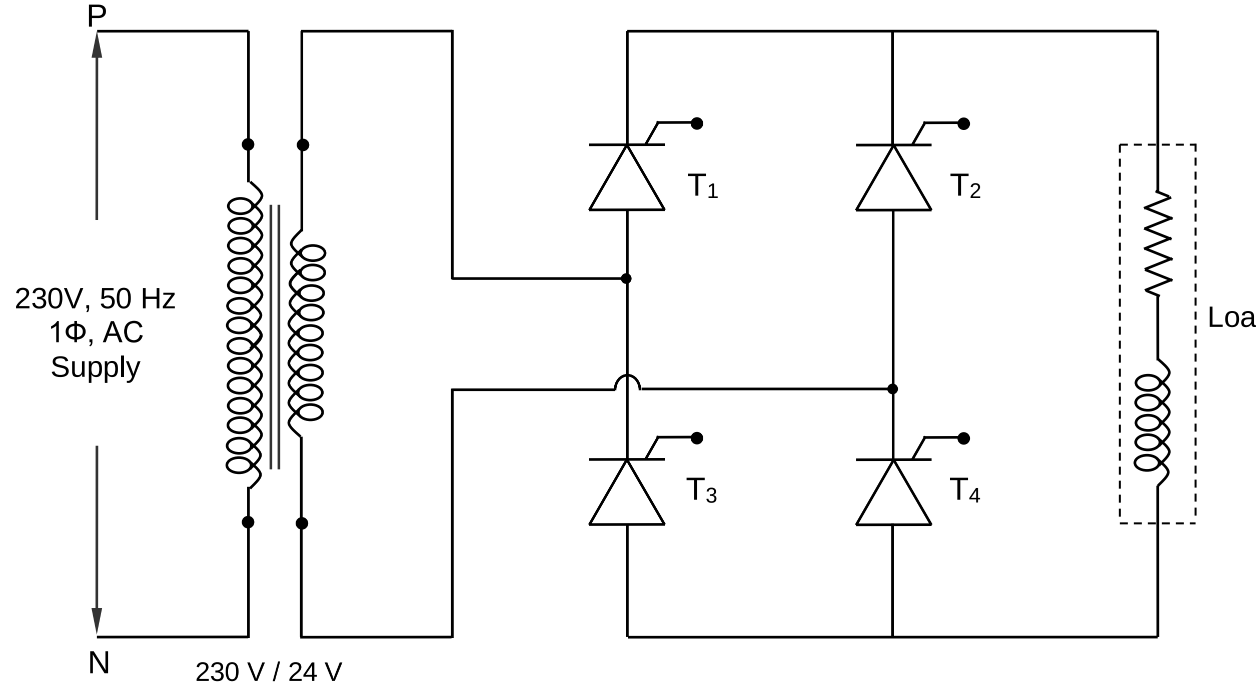
Model Graph



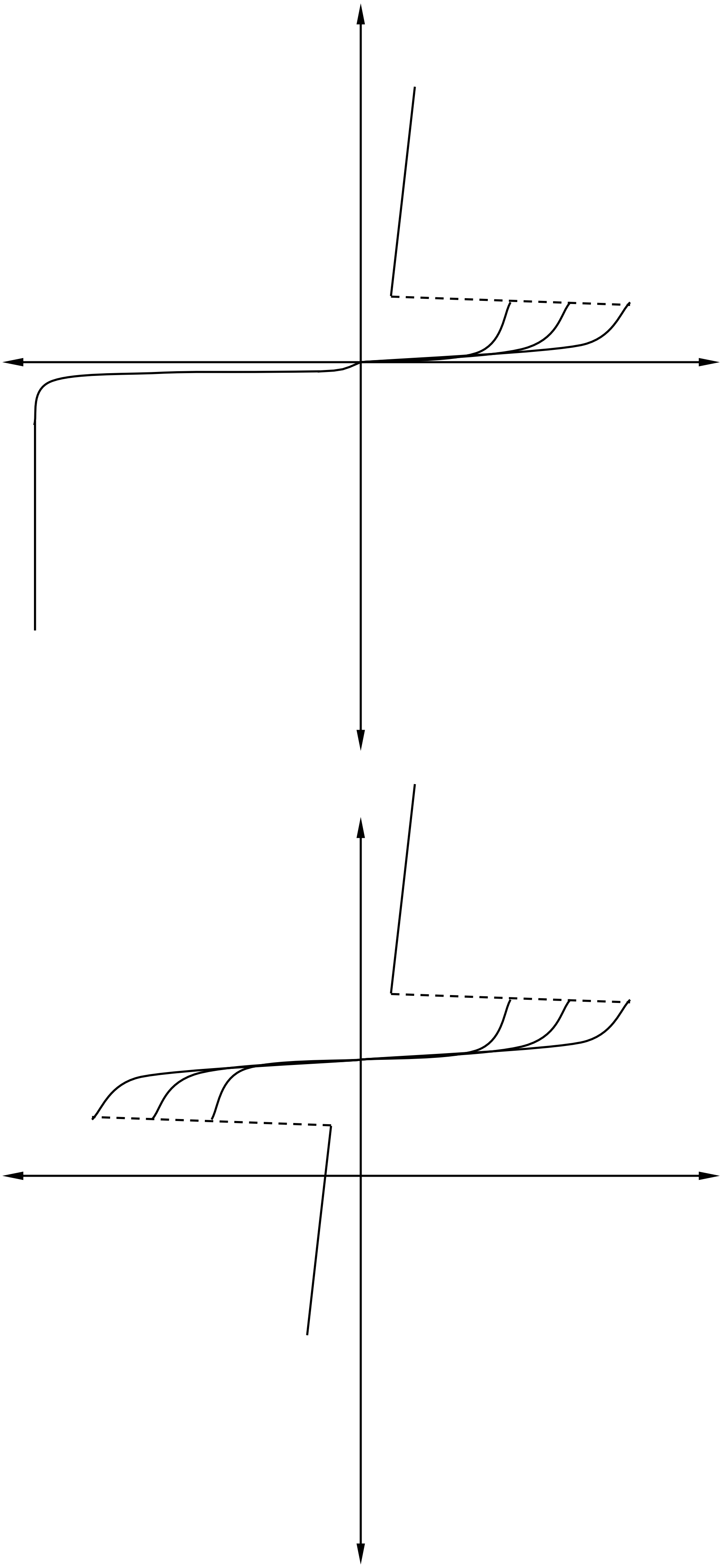
Output I – V Characteristics of IGBT

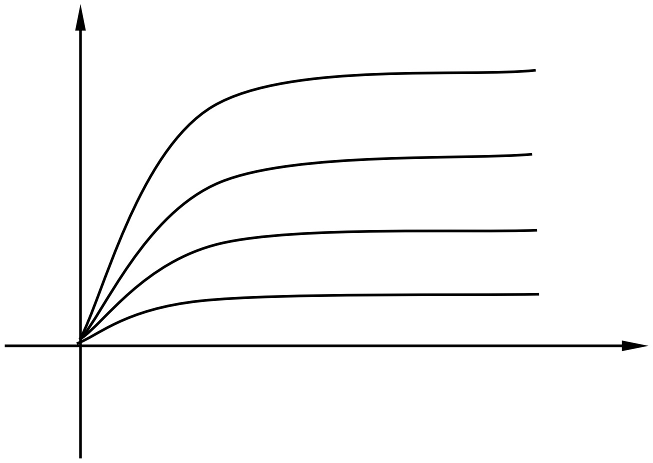
Transfer Characteristics of IGBT

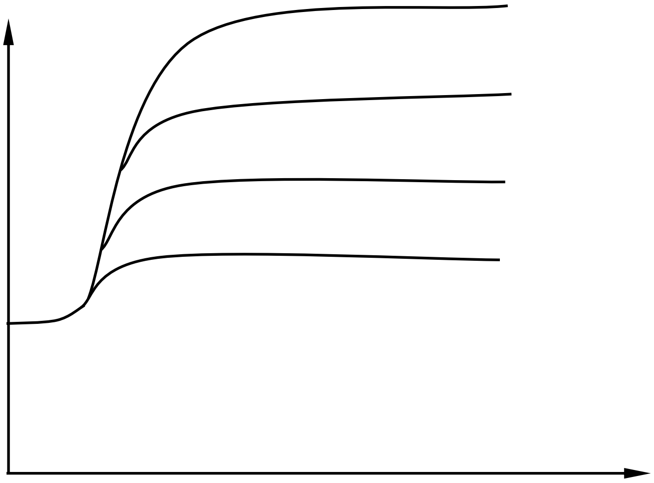




Transfer characteristics of IGBT







Result:

Thus the output characteristics of IGBT was studied and plotted.

Ex. No: 5	AC TO DC HALF CONTROLLED CONVERTER
Date:	

Aim:

To study the operation and characteristics of AC to DC half controlled converter

Apparatus Required:

S. No	Apparatus	Rating	Quantity
1.	Single Phase SCR half controlled converter trainer Kit	VPET - 204	1
2.	CRO	-	1
3.	Patch cord	-	1

Theory:

Introduction:

- The rectifier circuits using diodes operate on AC supply voltage. They convert AC voltage into a fixed DC voltage. Therefore diode rectifiers are known as **"Uncontrolled Rectifier".**
- If we use SCR in place of diode then the output dc voltage can be controlled by varying the firing angle (α) of the SCR.
- Therefore rectifier circuit using SCRs are known as **"Controlled Rectifier"** or **"Controlled Converter".**
- The AC mains voltage itself is used for commutating the thyristors, therefore these converters are known as **natural** or **line commutated converters.**
- Since they convert fixed AC voltage into a variable DC voltage, these converters are used extensively in the industry to control the speed of DC motors.
- Half controlled converter is actually a full wave rectifier (two pulse converter). It can control the power in both the half cycle of the supply.
- Half controlled converter or also called semi controlled converter is a one quadrant converter and it has only one polarity of output voltage and current.

Important terms

Delay Angle (or) Firing Angle (α)

- The delay angle or firing angle is the value of “ωt” at which a thyristor or pair of thyristors is turned on.
- It is denoted by α and measured with respect to the zero crossing instant of the ac supply voltage.
- The values of α varies from 0^o to 180^o or from 0 to 2π.

Freewheeling Diode

- The stored energy of the inductive load is circulated itself with the help of freewheeling diode.
- These diodes are connected in reverse direction in parallel with inductive loads.
- It helps in providing a smooth current to the load and also eliminates the negative voltage across the load.

Operation (RL – Load)

- Single phase half controlled converter has tow legs; each leg is made up of one SCR and one Diode in series.

Mode I $(\alpha \le \omega t \le \pi)$

- During positive half – cycle SCR_1 (T_1) and diode (D_1) are forward biased.
- When SCR T_1 is triggered at an instant ωt = α. The conduction will takes place through SCR T_1 and D_1.
- As both the output voltage and current are positive, the inductive load will store the energy in this mode of operation i.e the flow of power is from source to load.

Mode II $(\pi \le \omega t \le \pi + \alpha)$ Freewheeling

- At instant ωt = π, the ac voltage goes to zero after π it become negative.
- Due to this negative input voltage the conducting SCR T_1 and diode D_1 will try to turn off, and load current trying to change its direction.
- As the load is inductive it will try to oppose any change in the current flowing through it.
- Thus the load inductance will try to maintain the load current unchanged

by inducting a self induced voltage across the load.

- This self induced load voltage will bring diode D_2 into conduction at instant π and D_1 is turned off.
- SCR T_1 continues to conduct. In this way the load is short circuited by the conducting device T_1 & D_2.
- The freewheeling takes place due to stored energy in the inductive load and during freewheeling this stored energy is dissipated through load resistance, T_1 and D_2
- This mode of operation continues upto ωt = π+α, where SCR T_2 is turned on to turn off T_1
- The load voltage and supply current are zero in this mode.

Mode III $(\pi + \alpha \leq \omega t \leq 2\pi)$

- Mode III begins at instant ωt = π+α. At this instant the AC input voltage is negative.
- This negative voltage forward biases SCR T_2 and Diode D_2 and current conduction takes place.
- As load voltage and current both are positive the load will store the energy.

Mode IV $(2\pi \leq \omega t \leq 2\pi + \alpha)$ **Freewheeling**

- This mode is similar to mode II. But in this mode the stored energy in the inductive load is dissipated through load resistance, T_2 and D_1.
- In this mode also the load voltage and supply current are zero.

Formula

- Output voltage V_{dc} for both R and R-L loads.

$$V_{dc} = \frac{V_m}{\pi}(1 + \cos\alpha)$$

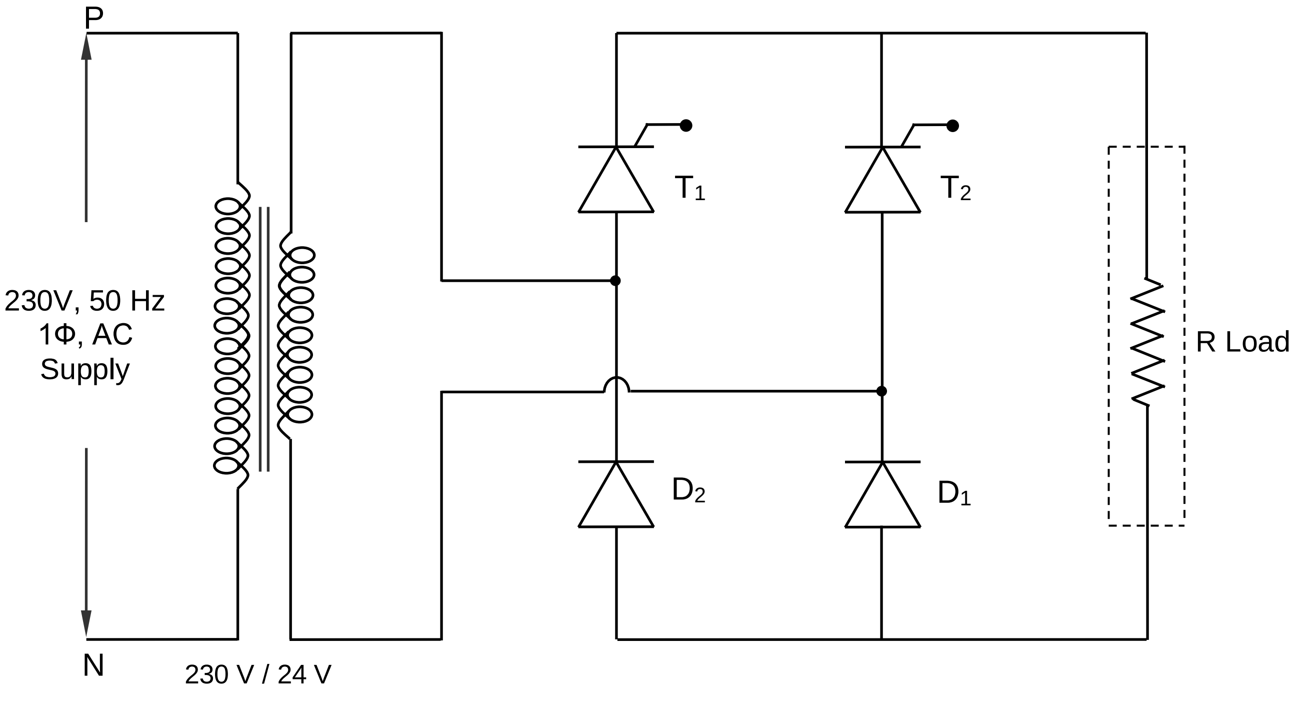
Where

V_{dc}	-	Average output voltage
V_m	-	Peak or Maximum output voltage
α	-	Firing angle

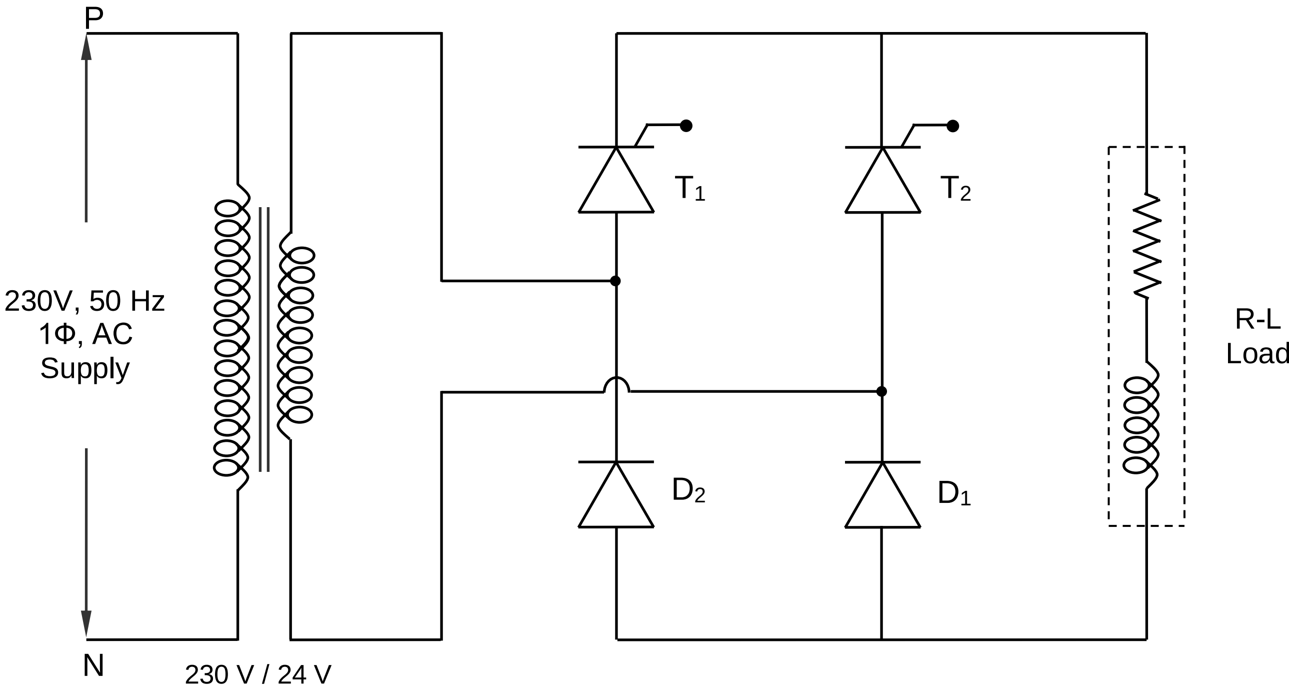
Connection Procedure:

- Initially all the power witches should be off.
- Set the switch SW_2 in 'INT' mode and switch on SW_3
- Connect the pulse connectors P_5 to P_9 and P_7 to P_{12}.
- Switch on SW_4 (AC)
- Connect the CRO across the load.
- Vary the POT (firing angle) from minimum to maximum position.
- Note down the output voltage by using CRO
- For various firing angle note down the output voltage.

CIRCUIT DIAGRAM FOR AC TO DC HALF CONTROLLED CONVERTER WITH R LOAD



CIRCUIT DIAGRAM FOR AC TO DC HALF CONTROLLED CONVERTER WITH R-L LOAD



Tabulation:

Type of Rectifier	**Nature of Load & firing angle**	**Measured Voltage V_{dc} (V)**	**CRO Reading (V)**		**Calculated Voltage E_{dc} (V)**	**Full Cycle Time Period (ms)**
			+ ve Peak	- ve Peak		
Half Controlled Converter	R Load 30°					
	R Load 60°					
	R-L Load 30°					
	R-L Load 60°					

Model Calculation:

To Find Average output Voltage (V_{dc}) for R Load

V_{dc} **=** $\frac{V_m}{\pi}(1+\cos\alpha)$ Volt

V_m – Peak Voltage = Volt

α – Firing Angle =

Output Voltage (V_{dc}) = Volt

To Find Average output Voltage (V_{dc}) for R-L Load

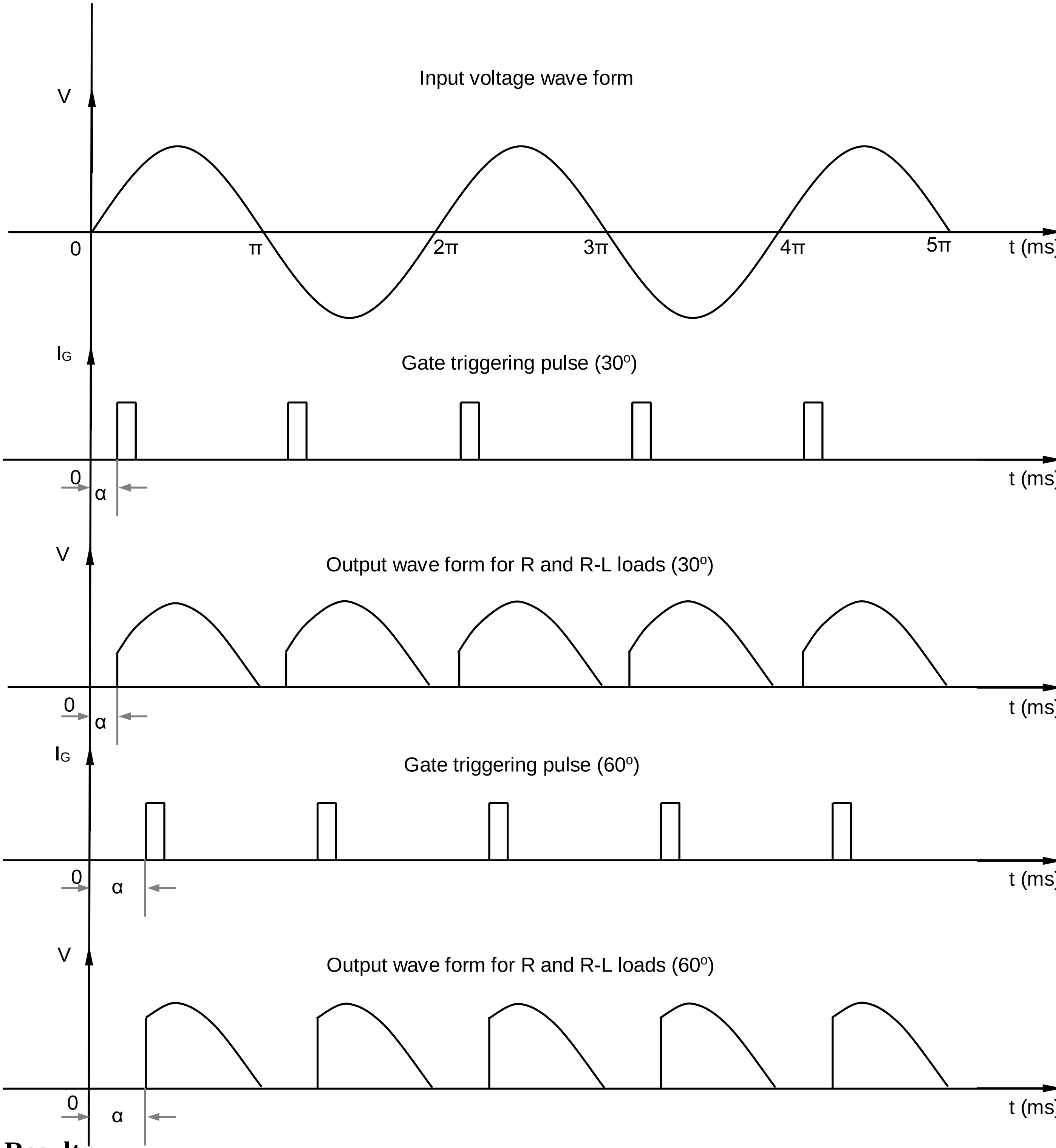
V_{dc} **=** $\frac{V_m}{\pi}(1+\cos\alpha)$ Volt

V_m – Peak Voltage = Volt

α – Firing Angle =

Output Voltage (V_{dc}) = Volt

Model Graph


Input voltage wave form
V
0
π
2π
3π
4π
5π
t (ms)
IG
Gate triggering pulse (30º)
0
α
t (ms)
V
Output wave form for R and R-L loads (30º)
0
α
t (ms)
IG
Gate triggering pulse (60º)
0
α
t (ms)
V
Output wave form for R and R-L loads (60º)
0
α
t (ms)

Result:

Ex. No: 6	**AC TO DC FULL CONTROLLED CONVERTER**
Date:	

Aim:

To study the operation and characteristics of AC to DC Full controlled converter

Apparatus Required:

S. No	Apparatus	Rating	Quantity
1.	Single Phase SCR full bridge controlled converter trainer Kit	PEC14M14FW	1
2.	CRO	-	1
3.	Patch cord	-	1

Theory:

Introduction:

- If the two diodes in the half controlled converter (semi converter) circuit are replaced by two SCRs, then we obtain full bridge converter configuration.
- The full converter configuration consists of four SCRs. At any given instant two SCRs (1&2, or 3&4) conduct simultaneously to connect the load across the input ac supply.
- In full converter the polarity of average load voltage can be positive or negative depending on the value of α. But load current is always positive.
- It can operate in two modes namely rectifier mode & inverter mode.
- Rectifier mode of operation is defined as the mode of operation in which the net flow of power is from ac to dc i.e. from source to load.
- Whereas in the inverter mode of operation the net flow of power is from dc to ac side i.e. load to source.
- As the full converter circuit can operate in any of these two modes, the power flow in the full converter can be bi-directional (but current direction always unidirectional).

Operation (R-L Load)

Mode I $(\alpha \le \omega t \le \pi)$

- During positive half – cycle SCR T_1 and T_2 are forward biased.
- When SCRs are triggered at an instant ωt = α. The conduction will takes place through SCRs T_1 and T_2.
- As both the output voltage and current are positive, the inductive load will store the energy in this mode of operation i.e the flow of power is from source to load.

Mode II $(\pi \le \omega t \le \pi + \alpha)$ **Freewheeling**

- At instant ωt = π, the ac voltage goes to zero after π it become negative.
- However, the inductive load will try to oppose any change in current through it.
- Thus the load inductance will try to maintain the load current unchanged by inducting a self induced voltage across the load.
- This self induced voltage is high enough to maintain the SCRs T_1 and T_2 forward biased inspite of the negative supply voltage. In this way SCRs T_1 and T_2 continue to conduct even after the supply voltage has become negative.
- The load voltage is negative and equal to the instantaneous ac supply voltage whereas the load current continues to be positive.
- Therefore the load act as the source and the stored energy in the inductance is returned back to the ac supply.

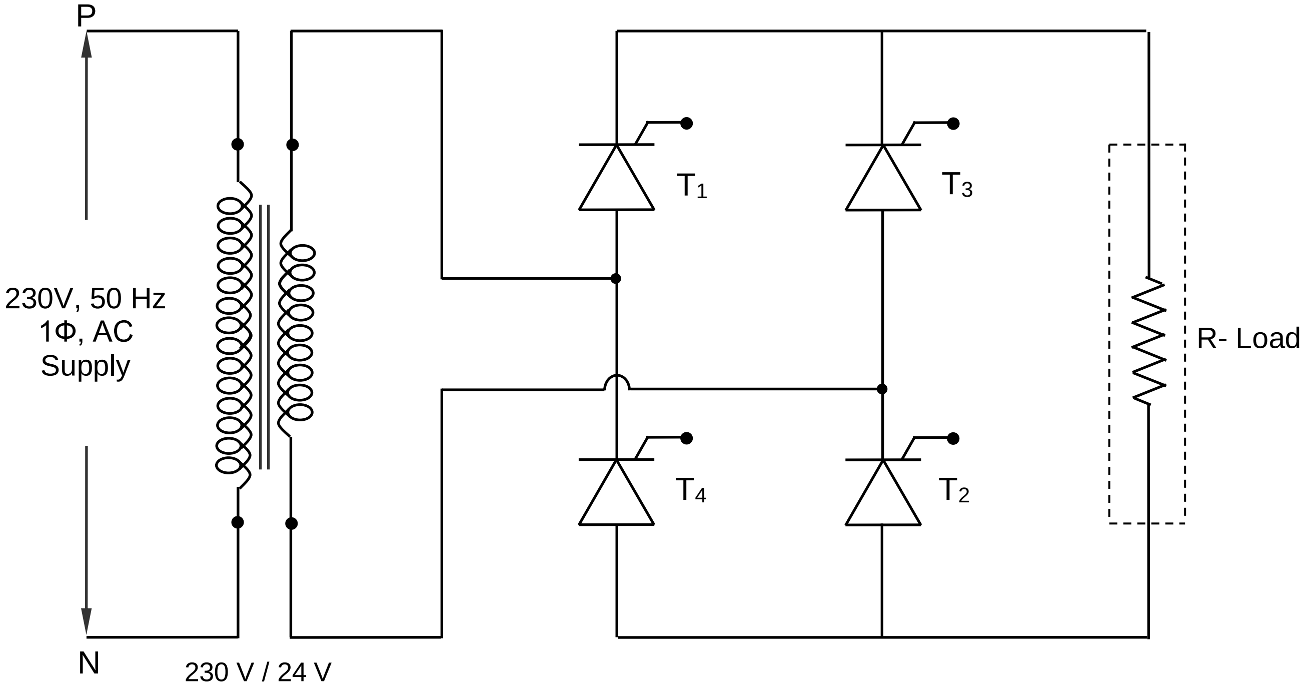
Mode III $(\pi + \alpha \le \omega t \le 2\pi)$

- Mode III begins at instant ωt = π+α. At this instant the ac input voltage is negative.
- This negative voltage forward biases SCRs T_3 & T_4 and current conduction takes place.
- This reverse line voltage will commutate the conducting SCRs (T_1 & T_2) due to line commutation.
- As load voltage and current both are positive the load will store the energy.

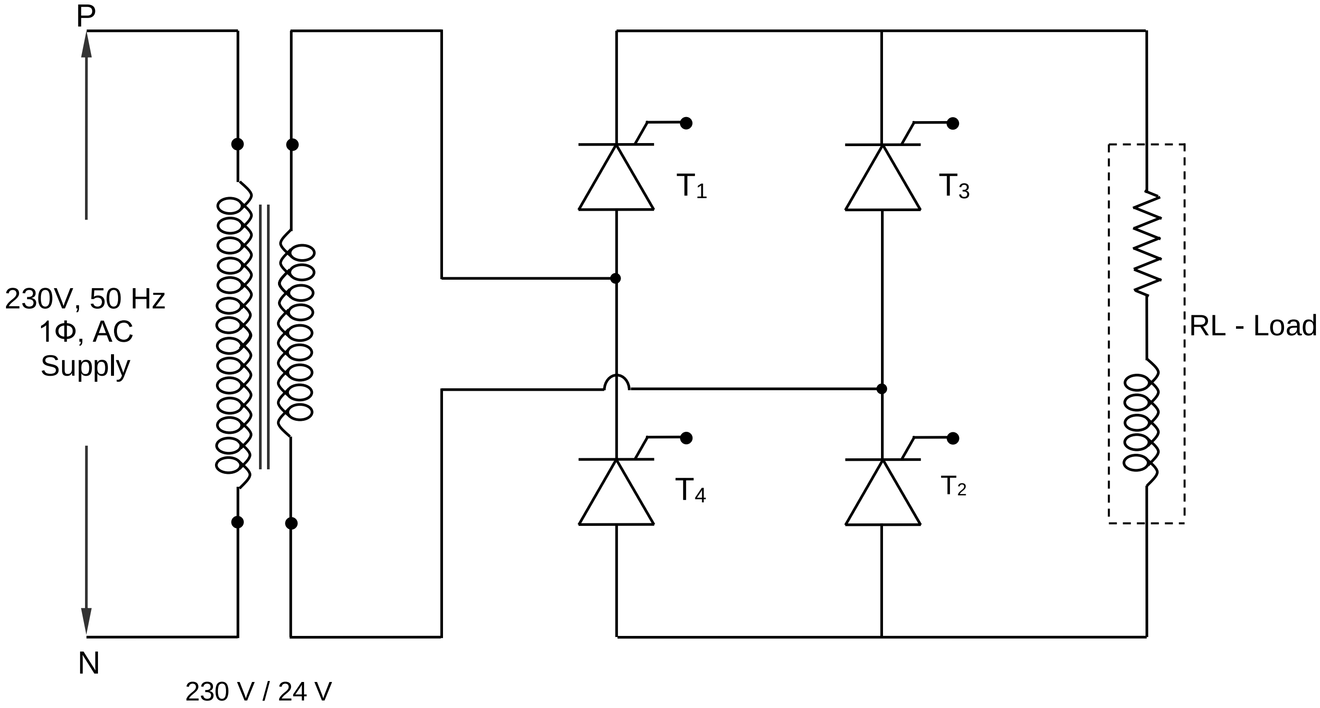
Mode IV $(2\pi \le \omega t \le 2\pi + \alpha)$ **Freewheeling**

- This mode is similar to mode II. But in this mode the stored energy in the inductive load is fed back to the source through SCRs T_3, T_4.

CIRCUIT DIAGRAM FOR AC TO DC FULL CONTROLLED CONVERTER WITH R LOAD



CIRCUIT DIAGRAM FOR AC TO DC FULL CONTROLLED CONVERTER WITH R-L LOAD



Tabulation:

Type of Rectifier	Nature of Load & firing angle	Measured Voltage E_{dc} (V)	CRO Reading (V)		Calculated Voltage E_{dc} (V)	Full Cycle Time Period (ms)
			+ ve Peak	- ve Peak		
Full Controlled Converter	R Load 30^o					
	R Load 60^o					
	R-L Load 30^o					
	R-L Load 60^o					

Model Calculation:

To Find Average output Voltage (V_{dc}) for R Load

V_{dc} **=** $\frac{V_m}{\pi}(1+\cos\alpha)$ Volt

V_m – Peak Voltage = Volt

α – Firing Angle =

V_{dc} =

Output Voltage (V_{dc}) = Volt

To Find Average output Voltage (V_{ac}) for R-L Load

V_{dc} **=** $\frac{2V_m}{\pi}(\cos\alpha)$ Volt

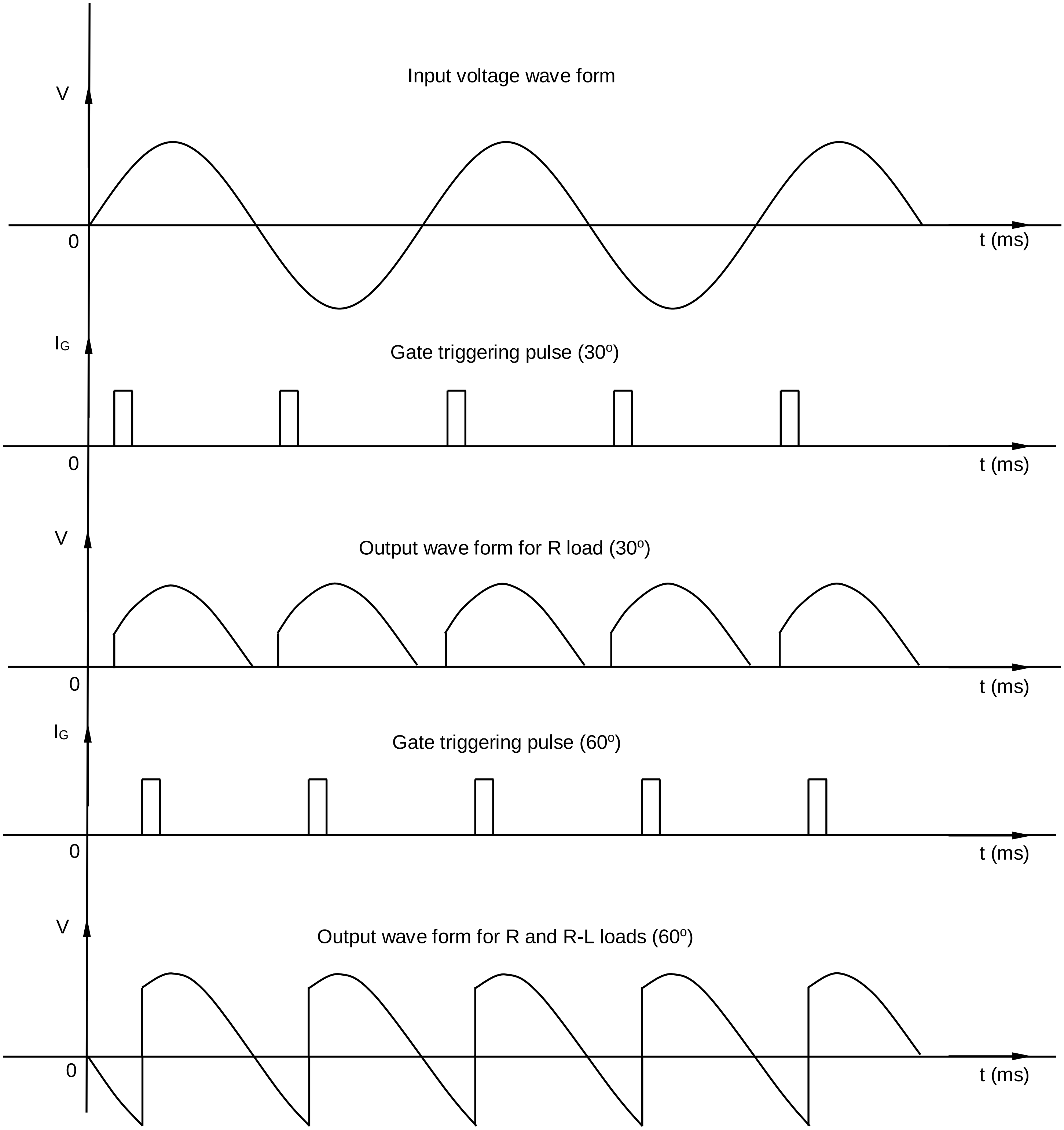
E_M – Peak Voltage = Volt

α – Firing Angle =

V_{dc} =

Output Voltage (V_{dc}) = Volt

Model Graph


Input voltage wave form
V
0
t (ms)
IG
Gate triggering pulse (30°)
0
t (ms)
V
Output wave form for R load (30°)
0
t (ms)
IG
Gate triggering pulse (60°)
0
t (ms)
V
Output wave form for R and R-L loads (60°)
0
t (ms)

Formula

- Output voltage V_{dc} for R-L load.

$$V_{dc} = \frac{2V_m}{\pi}(\cos\alpha)$$

Where

V_{dc}	-	Average output voltage
V_m	-	Peak or Maximum output voltage
α	-	Firing angle

Connection Procedure:

- Connect Anode (A_1) of SCR_1 to Cathode (K_4) of SCR_4.
- Connect Cathode (K_1) of SCR_1 to Cathode (K_3) of SCR_3.
- Connect Anode (A_3) of SCR_3 to Cathode (K_2) of SCR_2.
- Connect Anode (A_4) of SCR_4 to Anode (A_2) of SCR_2.
- Connect 24V AC input P terminal to Anode (A_1) of $SCR_{1.}$
- Connect 24V AC input N terminal to Cathode (K_2) of $SCR_{2.}$
- Connect load resistor (incase of R-L load connect load Resistor & Inductor) across the K_3 and A_2.
- Connect CRO probe across the load resistor.
- Connect digital multimeter or voltmeter across the load resistor.

Experimental Procedure:

- Switch ON the trainer power switch.
- Switch ON the 24 AC voltage switch.
- Switch ON the Debounce logic switch.
- Vary the control voltage minimum to maximum step by step.
- For different values of firing angle α calculate output voltage using formula.

Result:

Thus the operation and characteristics of AC to DC Fully controlled rectifier was studied.

7. STEP UP AND STEP DOWN MOSFET BASED CHOPPERS

Aim:

To examine and to verify the output of a step down and step up MOSFET based chopper using DC-DC Buck Boost trainer kit.

Apparatus Required:

S.No	Item	Range/Type	Quantity
1	DC-DC Buck Boost converter trainer kit	VSMPS – 07A	1
2	CRO	-	1
3	Pulse Patch chords	-	-
4	Regulated power supply	(0-30) V	1

Theory:

DC – DC converter is a device that accepts a DC input voltage and produces DC output voltage. The Buck Boost is a popular non – isolated, inverting power stage topology sometimes called a step-up/step down power stage. Buck Boost power stage is chosen because the output voltage is inverted from the input voltage, and the output voltage can be either higher or lower than the input voltage. However the output voltage is opposite in polarity from the input voltage. The output voltage magnitude depends on the duty cycle ratio K of the converter. The Buck – Boost converter circuit consists of MOSFET, Diode D, Filter Capacitor C and load resistor R.

Precaution:

1. The input DC supply should be a variable DC supply. If 15 v is given directly through AC-DC rectifier it may damage the device.

Formula Used:

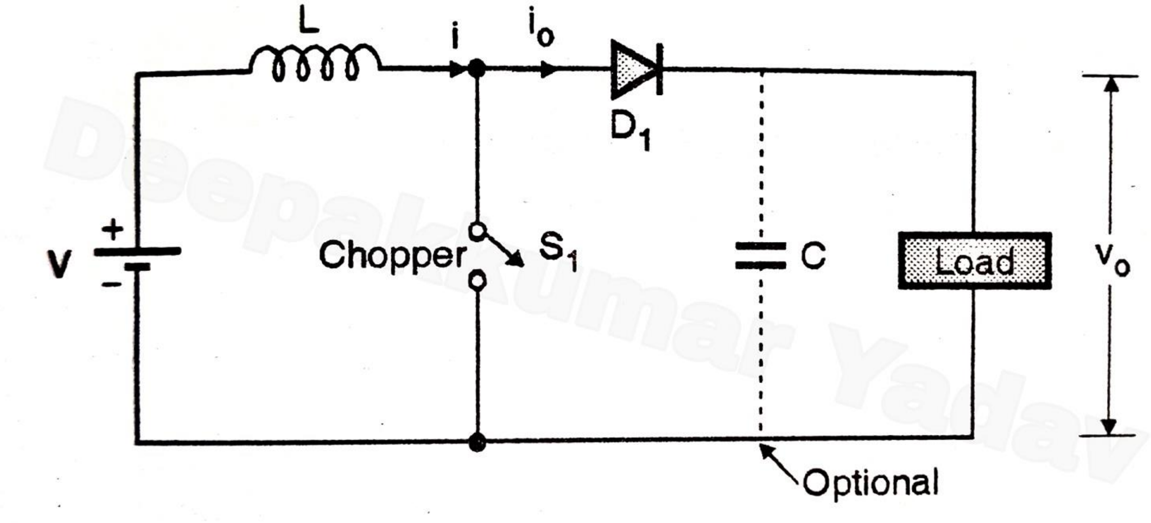
The output voltage

$$V_0 = \frac{D\ V_S}{1 - D}$$

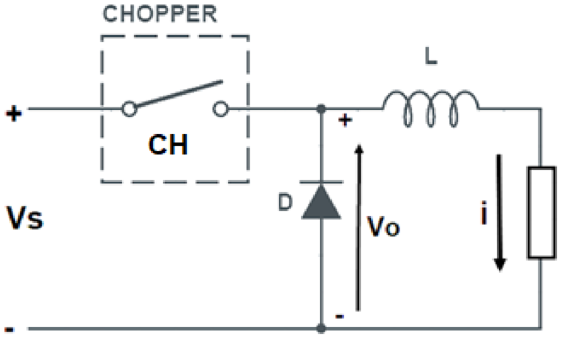
Where,

V_0 = Converter output voltage, volts
V_S = Converter input voltage, volts
D = Duty cycle ratio (t_{on} / T)
t_{on} = Switch on time, sec
T = Device switching time (t_{on+} t_{off})

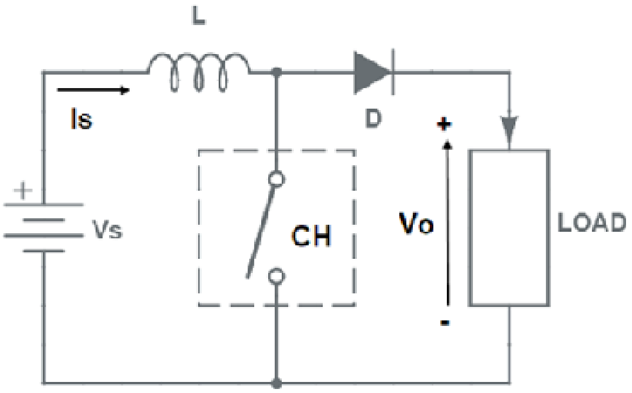
Circuit Diagram:



Step Up chopper



Step-down Chopper



Step-up Chopper

Connection Procedure:

1. Connect the output of the PWM generator to the PWM input of the Buck Boost converter circuit.
2. Connect Feed Back voltage of Buck Boost converter circuit to the feed back volt input of the PWM generator.
3. Connect (0-15) V RPS across P1 and P2 terminals of the trainer module.

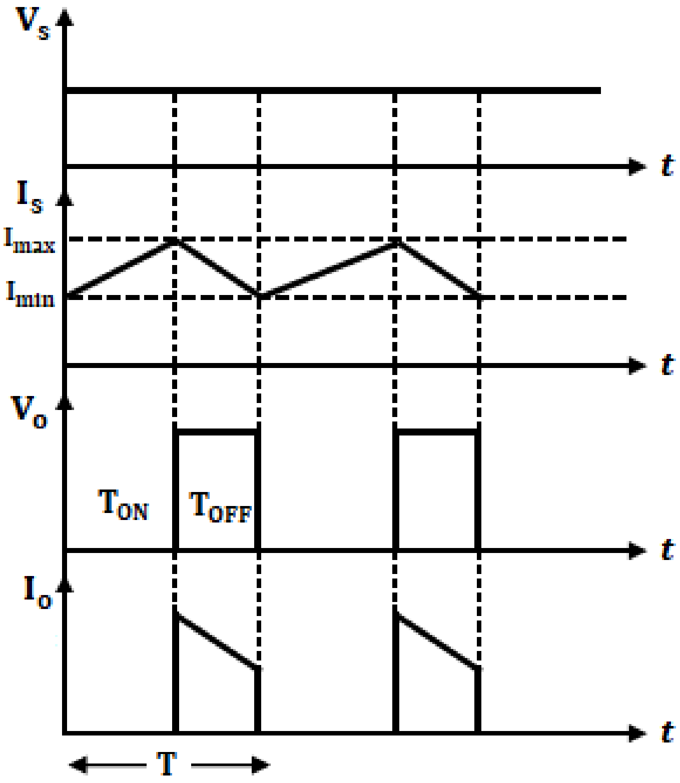
Experimental Procedure:

1. Switch on the power supply.
2. Switch on the variable DC power supply and the voltage at some value say 15 V.
3. Connect the CRO between T1 and GND (T5) to view the PWM signal (pulses)
4. Vary the pot from minimum to maximum to set the value of the pulses at desired duty cycle ratio.
5. Measure T_{ON} and T_{OFF}.
6. Calculate duty cycle ratio
7. Calculate the output voltage
8. Connect the CRO across P5 and P6 (i.e.) the output terminals of the trainer module and view the output voltage.
9. For various input values determine the output voltages and tabulate the readings.

Tabular Column:

S. No.	Duty cycle D	V_S (volts)	Measured V_0 (volts)	Calculated V_0 (volts)

Model Graph:



Voltage and Current Waveforms of Step Up Chopper

Result:

Thus the output of a step down and step up MOSFET based chopper was examined and verified.

Ex. No: 8	**IGBT BASED SINGLE-PHASE PWM INVERTER**
Date:	

Aim:

To study the operation of IGBT based single phase Inverter with sinusoidal Pulse Width Modulation.

Apparatus Required:

S. No.	Apparatus	Rating	Quantity
01	IGBT power Module	-	01
02	Single phase PWM inverter control module	PEC16M4	01
03	CRO	-	01
04	Rheostat	50Ω, 4 A	01
05	Multimeter	-	01
06	Patch cord	-	As required

Theory:

Introduction:

- The most efficient method of controlling the output voltage of an inverter is to incorporate pulse with modulation control (PWM control) within the inverters.
- In this method, a fixed DC input voltage is supplied to the inverter and a controlled AC output voltage is obtained by adjusting the ON and OFF periods of the inverter.
- Advantages of PWM control
  1. The output voltage can be obtained without any additional components.
  2. With this type of control, lower order harmonics can be eliminated or mininised alone with its output voltage control.
  3. The filtering requirements are mininised as higher order harmonics can be filtered easily.
- The main drawback of this method is that the SCR used in this method must have very low turn - on and turn – off times (inverter grade SCRs), therefore, they are very expensive.

- The commonly used PWM control techniques are:
  1. Single-pulse width modulation (SPWM)
  2. Multiple-pulse modulation (MPWM)
  3. Sinusoidal pulse width modulation (sin PWM)
  4. Modified Sinusoidal pulse width modulation
- In sinusoidal pulse width modulation, several pulses per half cycle are used as in the case of multiple pulse width modulation.
- Instead of maintaining the width of all pulses the same as in the case of multiple pulse modulation, the width of each pulse is varied proportional to the amplitude of a sine wave evaluated at the centre of the same pulse.
- By comparing a sinusoidal reference signal with a triangular carrier wave of frequency, f_c, the gating signals are generated.
- The frequency of reference signal, $f_{r,}$ determine the inverter output frequency f_o.
- The peak amplitude of reference signal V_r, control the modulation index MI.
- By varying the modulation index MI, the RMS output voltage can be varied.
- The number of pulses per half cycle depends on the carrier frequency f_r

Formula :

- Modulation Index MI $= V_R / V_C$

  Output voltage V_O $= MI. V_S$

  V_R - Amplitude of the reference sine wave

  V_C - Amplitude of the carrier (triangular) wave

  V_S - Output AC voltage level

Precautions :

- Check whether the AC main switch is OFF condition in both the trainer.
- Check whether control module selector switch is in sine wave mode.
- Check whether control module pulse release switch SW_4 in control module is in OFF condition.
- Check whether 24 V AC SW_1 is in OFF condition.

Connection Procedure:

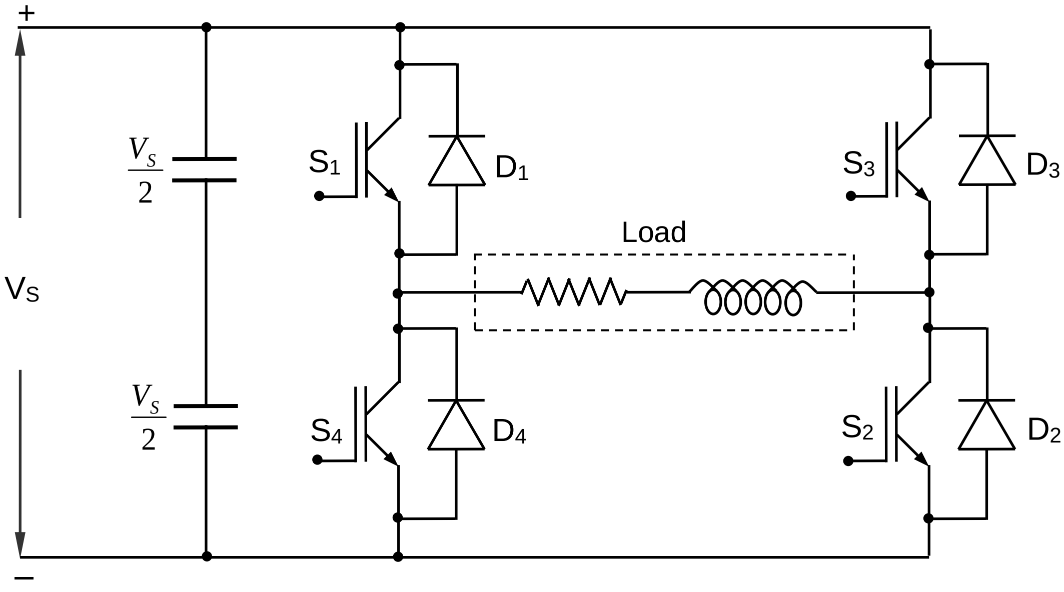
- Make the connection as per the diagram (short P_1&P_2, P_4&P_7, P_8&P_9).
- Connect R load as shown in figure (Between P_5&P_6).
- Connect the gating signals from the inverter control module to the inverter module through single cable provided.
- Connect 24 V AC voltage to IGBT trainer.
- Connect the multimeter across the load terminal to determine the voltage.

Experimental Procedure:

- Switch on the AC main in both trainer.
- Measure the amplitude and frequency of the sine wave and carrier triangular wave and tabulate it. Also adjust sine wave frequency about 50 Hz.
- Connect CRO probe to observe the wave form.
- Release the switch SW_4 in inverter control module and SW_1, which is in IGBT module.
- Measure the output voltage and tabulate the readings.
- Vary the amplitude pot step by step, for each step note down the amplitude and frequency of sine wave and triangular waveform and also measure the output voltage and tabulate it.

  Then find out the output voltage using the formula.

CIRCUIT DIAGRAM FOR IGBT BASED 1 PHASE PWM INVERTER



Tabulation:

S. No	$V_{Carrier}$	V_{Ref} (sine)	Modulation Index M	V_O Measured	V_O Calculated
1					
2					
3					
4					
5					

Result:

Thus the operation of IGBT based single phase Inverter with sinusoidal Pulse width modulation was performed.

Ex. No: 9	**AC PHASE CONTROL USING SCR**
Date:	

Aim:

To control the ac load voltage by phase control using SCR.

Apparatus Required:

S. No	Apparatus	Rating	Quantity
1	Ac phase control using SCR	PEC14M22	1
2	Lamp	40 W	1
3	Step down transformer	230 / 24 V	1
4	CRO	-	1
5	Patch cords	-	As Required

Theory:

Introduction:

- AC voltage controllers are the device which converts fixed alternating voltage into variable alternating voltage without change in supply frequency.
- It is also called as ac voltage regulator.
- AC voltage controllers using thyristors are quite common. The load voltage is controlled by controlling the firing angle of the thyristors.
- The AC voltage controllers are used in applications such as industrial heating intensity control of lighting load, speed control of poly phase induction motors etc.
- Types of AC voltage Control technique
    1. On-off control (Integral cycle control)
    2. Phase angle control
- On – off Control
    - In on-off type ac voltage controllers, the back to back connected SCRs will conduct for a few cycles and then will remain off for next few cycles.
    - Thus SCRs act as diodes.
    - As the SCRs turn on and off at zero crossing instants of ac supply, the current does not get distorted. Therefore no additional harmonics components are introduced due to switching of SCRs.
    - Disturbance to the neighboring electronic circuits due to EMI (electromagnetic interference) minimum.

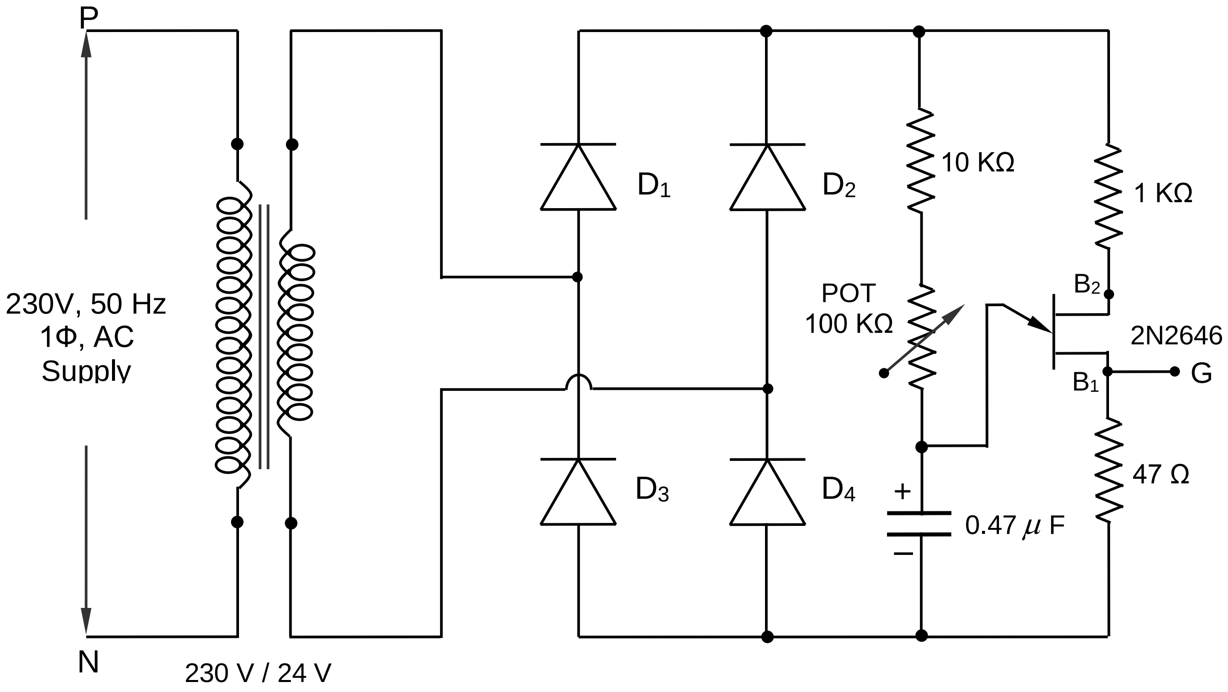
- Phase angle control
  - The AC voltage controllers which use the technique of phase angle control will change the firing angle or delay angle “α” of the SCRs or TRIAC in the ac controller circuit.
  - In AC phase control using SCR, two SCRs are connected in the anti-parallel configuration.
  - Due to the use of two SCRs, it is possible to control the load voltage and power in both the half cycles of ac supply, therefore this controller is also called as bidirectional controller.
  - SCR_1 will control the power delivered to the load in the positive half cycle and SCR_2 will control the load power in the negative half cycle.
  - The RMS output voltage can be controlled by varying the firing angle α from 0^o to 180^o.
  - A conducting SCR will turn off due to natural commutation at the end of the corresponding half cycle of supply voltage.

Procedure:

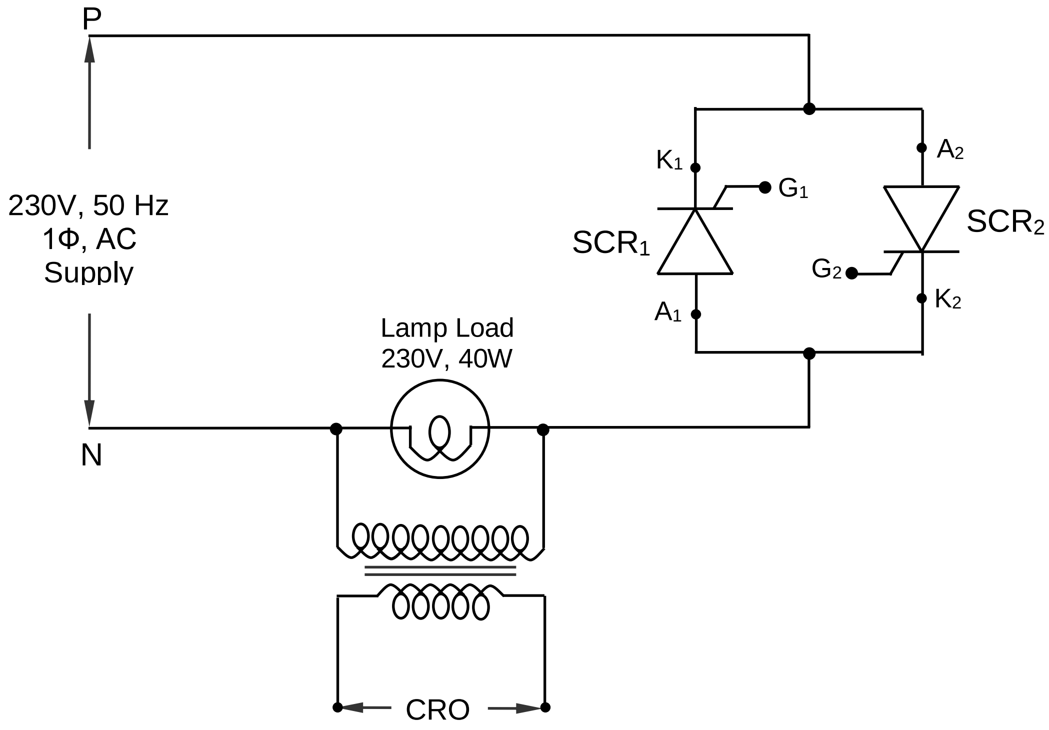
- Connections are made as per the circuit diagram
- Keep the lamp control in a minimum position
- Switch ON the AC main
- Gradually vary the firing angle and observe the intensity of light.
- Tabulate and plot the output voltage waveform for various firing angles using CRO

Circuit Diagram for Ac Phase Controlled Converter Using SCR

1. UJT based Firing Circuit



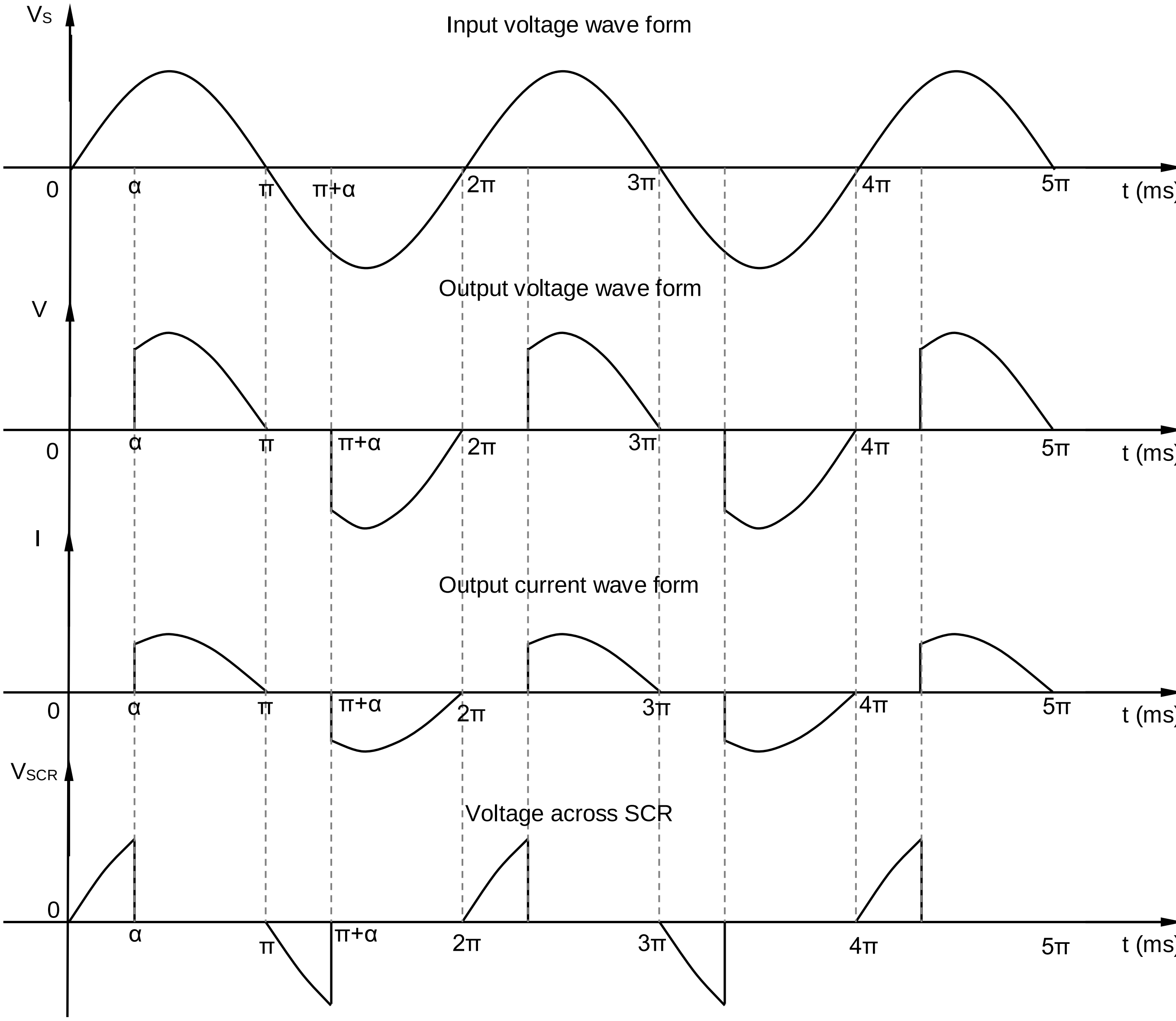
2. Power Circuit



Tabulation:

S. No	Input / Output	Firing Angle	Brightness	Amplitude	Time Period	
					T_{ON}	T_{OFF}
1	Input	0^o	Brightest			
2	Output	135^o	Dim			
3	Output	90^o	Medium			
4	Output	45^o	Bright			

Model Graph



Result:

Thus ac load voltage was controlled by phase control of SCR

Ex. No: 10	**AC PHASE CONTROL USING TRIAC AND DIAC**
Date:	

Aim:

To study the operation of ac phase control using TRIAC and DIAC.

Apparatus Required:

S. No	Apparatus	Rating	Quantity
1	Ac phase control using TRIAC	PEC14M22	1
2	Lamp	40 W	1
3	Step down transformer	230 / 24 V	1
4	CRO	-	1
5	Patch cords	-	As Required

Theory:

Introductio

- AC voltage controllers are the device which converts fixed alternating voltage into variable alternating voltage without change in supply frequency.
- It is also called as ac voltage regulator.
- AC voltage controllers using thyristors are quite common. The load voltage is controlled by controlling the firing angle of the thyristors.
- The AC voltage controllers are used in applications such as industrial heating intensity control of lighting load, speed control of poly phase induction motors etc.

Principle of operation

- The AC voltage is controlled using phase angle control technique and it uses the TRIAC as a power controlling device.
- In the positive half cycle, the TRIAC is turned on at $\omega t = \alpha$. It acts as a closed switch and instantaneous load voltage becomes positive and equal to the instantaneous supply voltage.
- The load current starts flowing and the load voltage and load current both are positive form α to π.
- The conducting TRIAC is turned off at $\omega t = \pi$ due to natural commutation.
- In the negative half cycle, the TRIAC is turned on at $\omega t = \pi + \alpha$. It acts as a closed switch and instantaneous load voltage becomes negative and equal to the instantaneous supply

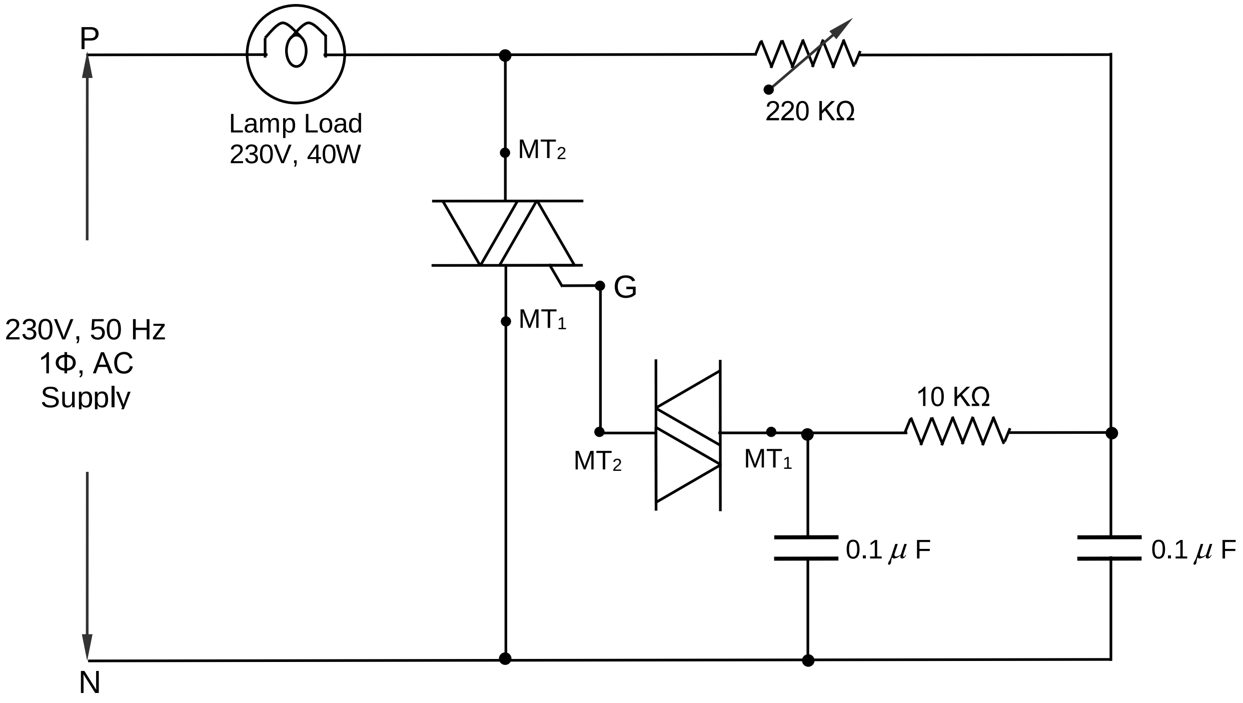
voltage.

- The load current starts flowing in the opposite direction. Thus the load voltage and current both are negative from (π + α) to 2π.
- The conducting TRIAC is turned off at ωt = 2π due to natural commutation.
- The load voltage will be equal to 0 when the TRIAC is not conducting i.e. from 0 to α and from π to π + α.
- The output current i_L or source current i_S waveform has the same shape as the output voltage because the load is resistive.

Procedure:

- Connections are made as per the circuit diagram
- Keep the lamp control in a minimum position
- Switch ON the AC main
- Gradually vary the firing angle and observe the intensity of light.
- Tabulate and plot the output voltage waveform for various firing angles using CRO

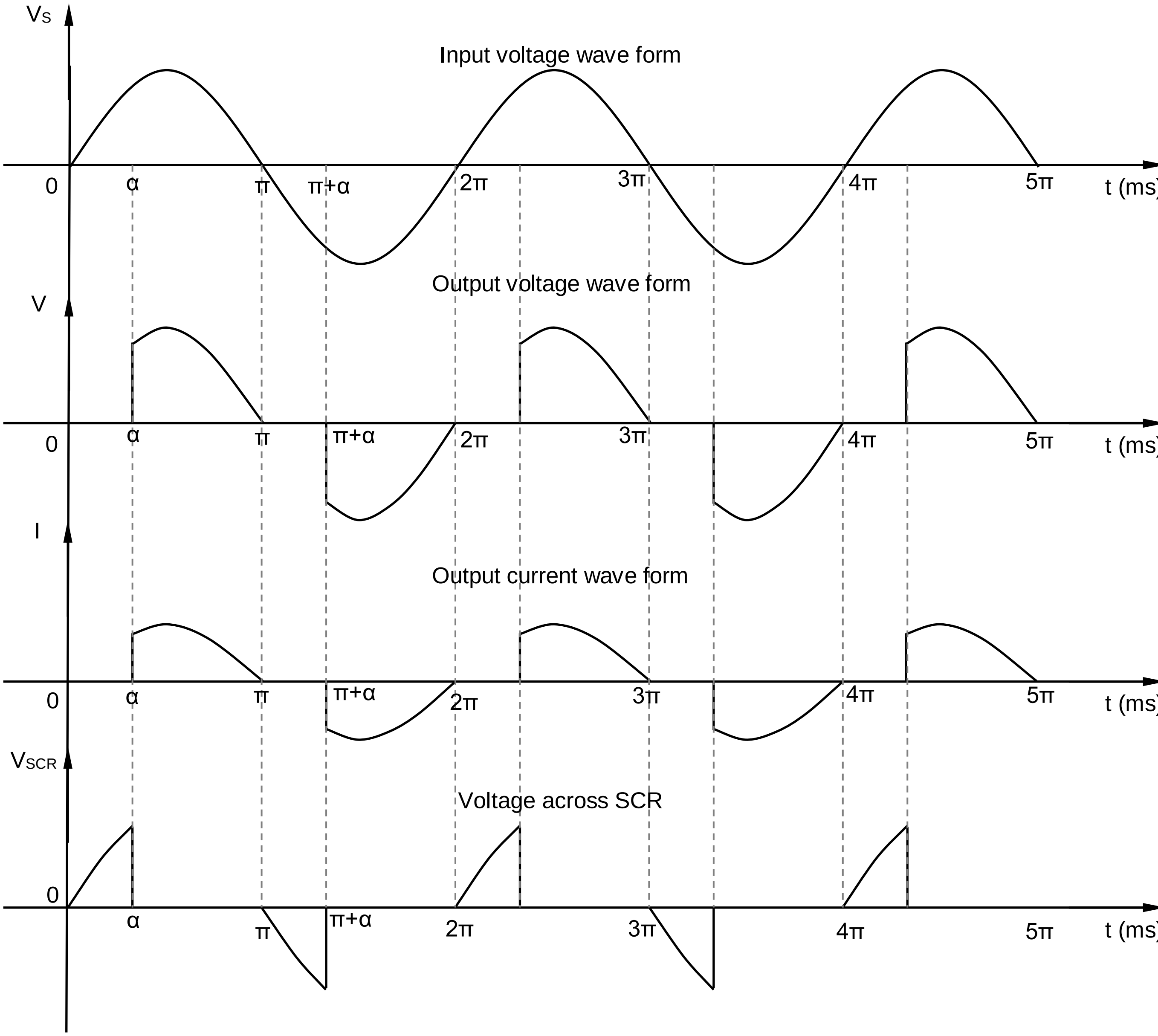
CIRCUIT DIAGRAM FOR AC PHASE CONTROLLED CONVERTER USING TRIAC



Tabulation:

S. No	Input / Output	Firing Angle	Brightness	Amplitude	Time Period	
					T_{ON}	T_{OFF}
1	Input	0°	Brightest			
2	Output	120°	Dim			
3	Output	90°	Medium			
4	Output	60°	Bright			

Model Graph



Result:

Thus the ac phase control using TRIAC and DIAC is studied and performed.

Ex. No: 11	**SERIES INVERTER**
Date:	

Aim:

To study the operation of a series inverter and to obtain the Waveform across the load.

Apparatus Required:

S. No	Apparatus	Rating	Quantity
1	Series inverter module	-	1
2	Rheostat	220Ω / 1A	1
3	CRO	-	1
4	Patch cords	-	As Required

Theory:

Introduction:

- Inverters in which commutating components are permanently connected in series with the load are called series inverters.
- The series circuits so formed must be under damped.
- As the current attains zero value due to the nature of the series circuits, series inverters are also classified as self-commutated inverters or load commutated inverters.
- These inverters operate at high frequencies (200 Hz to 100 KHz) the size of commutating components is, therefore small.
- These inverters are used extensively in induction heating, fluorescent lighting etc.

Working of Basic Series Inverter:

- In basic series inverter the commutating component L, C are connected in series to form an under damped tuned circuit.
- The SCRs turn off themselves hence it is also known self commutated inverters.
- At instant t_0 SCR_1 is turned on. Let the initial voltage on capacitor be V_C with its left plate negative with respect to right plate and the sinusoidal load current starts flowing.
- The load current eventually comes to zero at instant t_1 and SCR_1 comes out of conduction due to natural commutation.

- The voltage on the capacitor C at instant t_1 is greater than V with its left plate positive with respect to its right plate.
- As there is no discharge path for the capacitor, this voltage will be held constant up to instant t_2 where SCR_2 is triggered.
- At instant t_2, SCR_2 is turned on and the load voltage and current both become negative.
- The capacitor now discharge resonantly through SCR_2, load and L.
- At instant t_3 the discharge current goes to zero and SCR_2 is turned off again due to natural commutation. The voltage on C is now equal to VC.

Disadvantages of basic series inverter

- Limitation on the maximum operating frequency.
- Distortion in the output waveform
- High rating of commutating components.
- The peak amplitude and duration of output current depends on the load resulting in poor regulation for the inverter.
- The power flow from the DC source is intermittent.

Modified Series Inverter:

- In the basic series inverter circuit, the power delivered to the load was taken from the DC source in only one half cycle (in +ve half cycle).
- In the –ve half cycle the load power taken from the commutation capacitor C.
- Due to the intermittent flow of power from the DC supply, the DC supply must have a large peak current rating and the input current has high harmonic contents.
- This drawback can be eliminated by rearranging the series inverter circuit. In this circuit the power is drawn from the input DC supply in both the half cycles of output waveform.

Working of Modified Series Inverter:

- Let the initial voltage on C_2 be V_C and voltage across C_1 be $(V+V_C)$.
- When SCR_1 is turned on at instant t_0 the load current will be sum of two currents i_1 and i_2.
- i_1 is the charging current for C_2 whereas i_2 is discharging current of C_1.
- The current i_1 is taken from the DC supply.
- At the end of one half cycle when load current becomes zero SCR_1 is turned off due to natural commutation and voltage across C_1 and C_2 will be reversed. ($V_{C1} = V_C$ and $V_{C2} = V+V_C$ with reverse polarities.)
- Identical operation takes place in the negative half cycle form t_2 to t_3 when

SCR_2 is turned on.

- In this half cycle on a partial current is supplied by discharging capacitor C_2 and the remaining is supplied by DC source.
- However as the load voltage waveform is dependent on the load, the series inverter finds very little practical applications.

Precaution:

- 24 V dc supply should be OFF
- DC logic switch must be OFF
- Load must be maximum

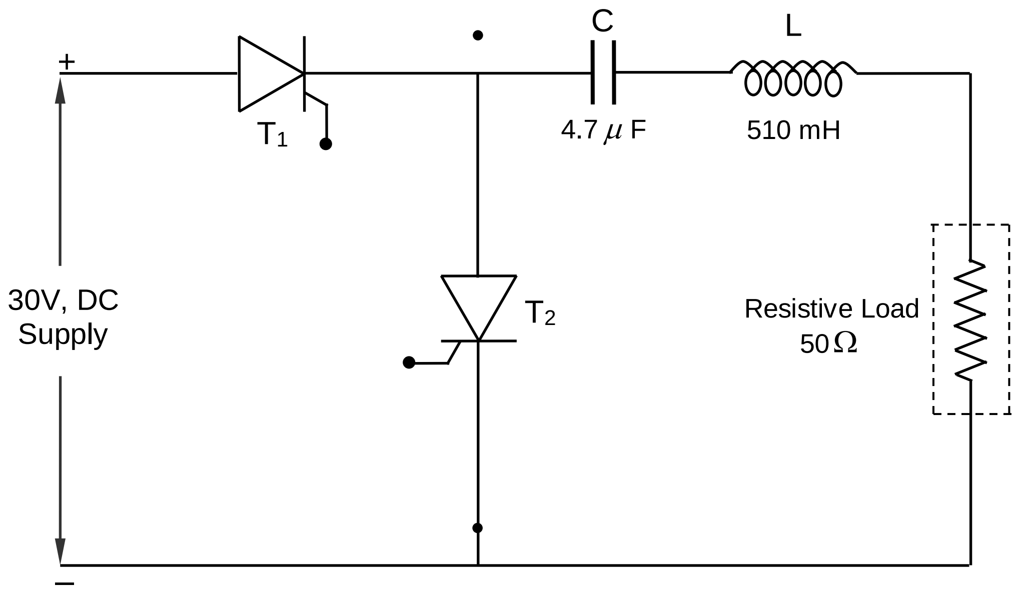
Connection Procedure:

- Connect the positive terminal of 24 V dc to the point '+' marked in the front panel.
- Connect the negative terminal of 24 V dc to the point '-' marked in the front panel.
- Connect C_1 with the connector named A_1 in the front panel diagram shown (dotted lines)
- Connect C_2 with the connector named K_2 in the front panel diagram shown (dotted lines)
- Connect "C.Com" terminal to the other point of input terminal in the filter circuit.
- Connect "L.Com" terminal to the other point of the filter in the filter circuit.
- Connect the Load rheostat across the output of the filter circuit.

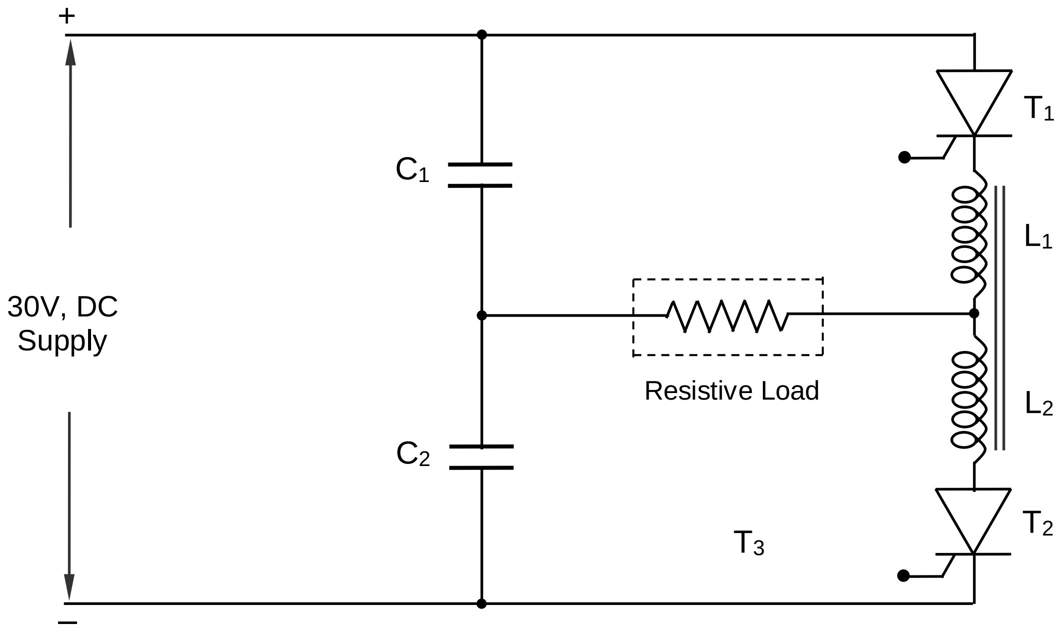
Experimental Procedure:

- Connect the CRO probe across the Rheostat.
- Switch on the power to the CRO.
- Switch on the power supply to the series inverter module.
- Switch on the 24 V DC power supply.
- Switch on the De-bounce Logic switch.
- Observe the wave form across the load in the CRO.
- For various frequency of gate pulse, observe the wave form.

BASIC CIRCUIT DIAGRAM OF SERIES INVERTER



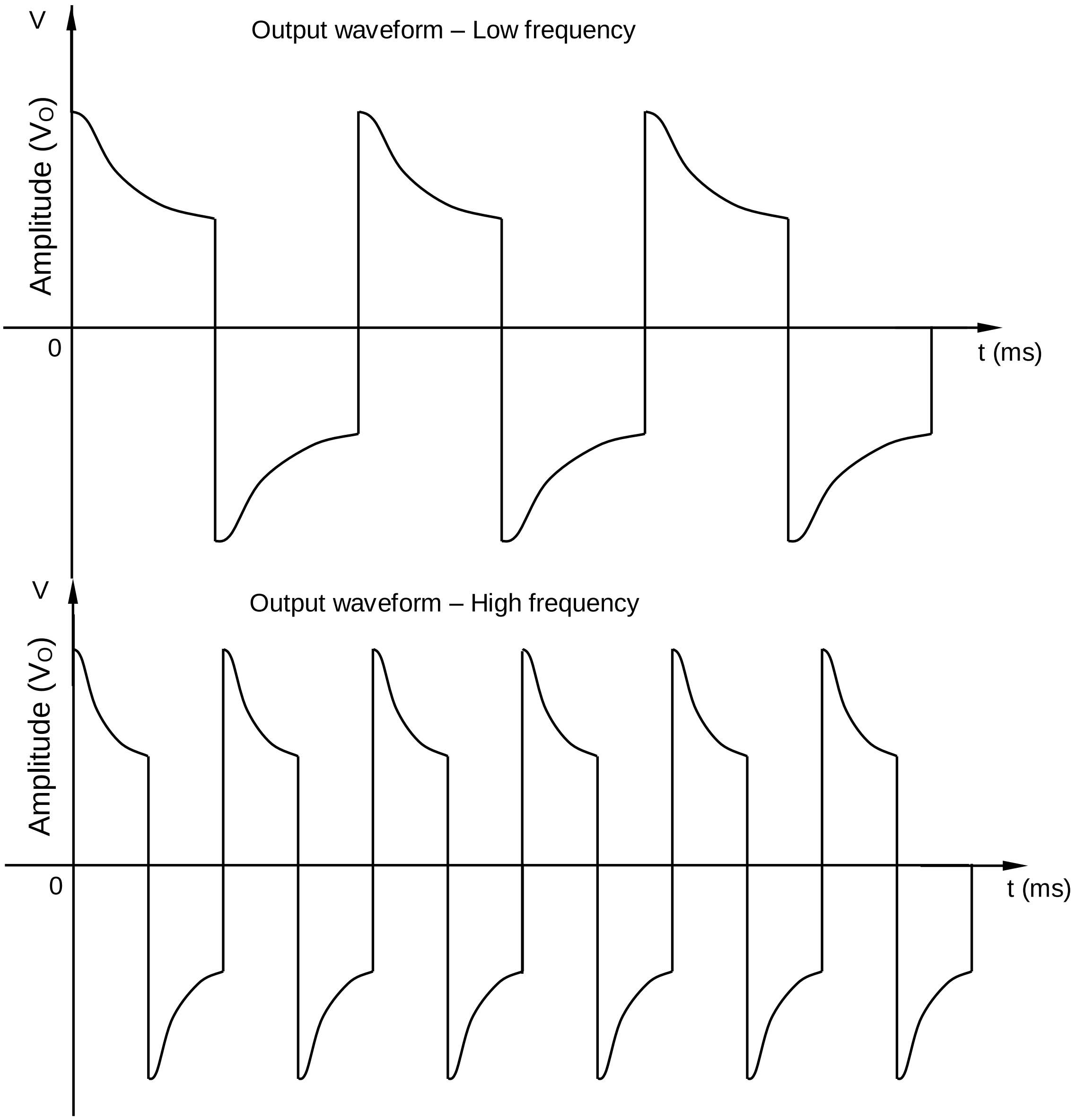
CIRCUIT DIAGRAM OF MODIFIED SERIES INVERTER



Tabulation:

S. No.	Frequency (Hz)	One full cycle Time Period (ms)	One half Amplitude (volt)
1			
2			

Model Graph



Output waveform – Low frequency

Output waveform – High frequency

Result:

Thus the operation of a series inverter was studied and the waveform was obtained.