Experiment Number -01
Control Systems (Lab)
PCEE-613
GEE-2018

Professor J.S. Dhillon
EIE Department

Experiment 01: To study the performance of various types of controllers used to control the temperature of an oven.

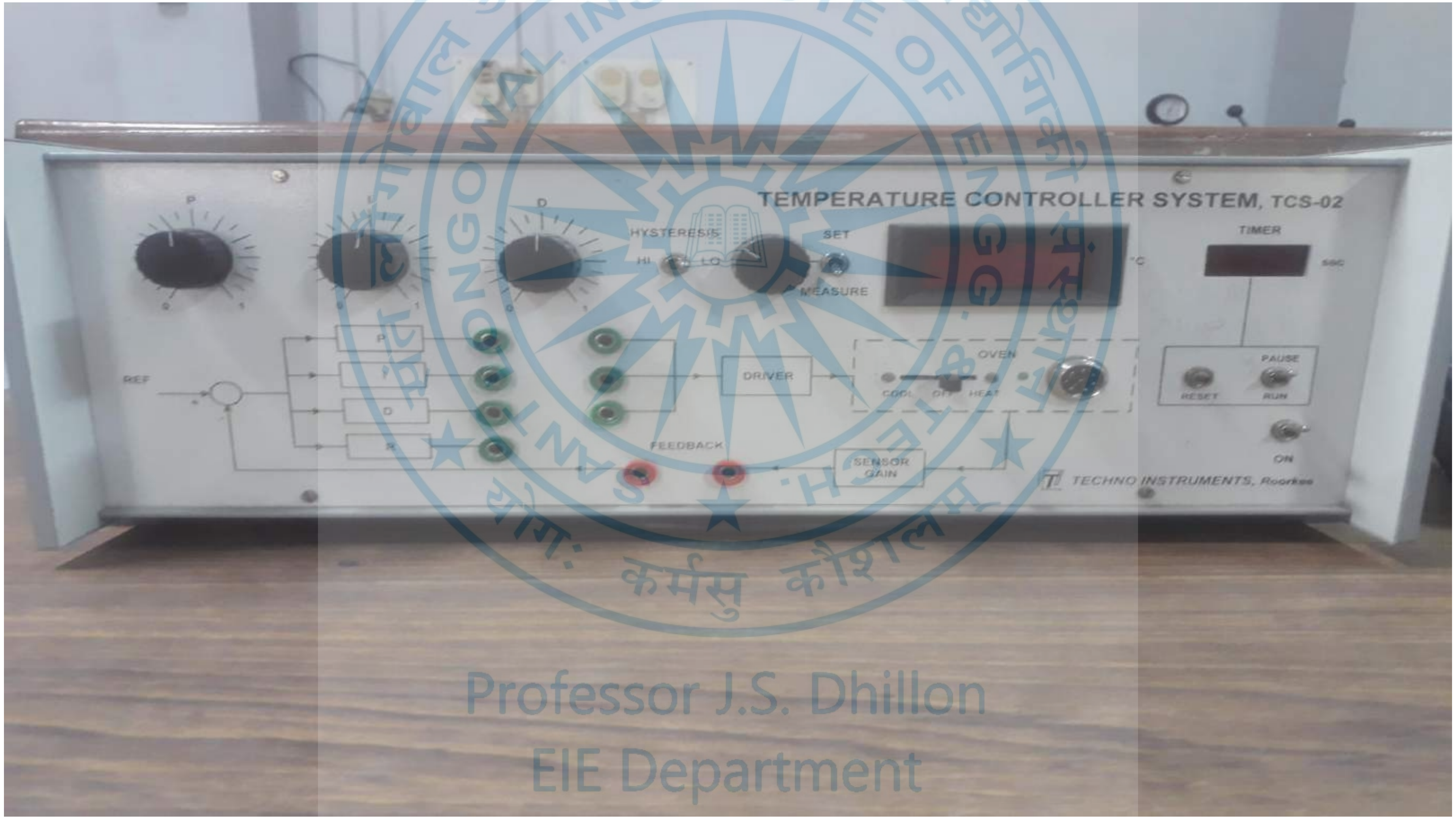
Apparatus Required:

- Temperature control unit
- Oven
- Multimeter
- Connecting leads

Theory

- The Controller unit compares the reference and the measured signals to generate the error.
- Controller options available to the users consist of ON-OFF or relay with two hysteresis settings and combination of proportional, derivative and integral blocks having independent coefficient settings.
- A block diagram of the complete system is shown in Figure
- The first step in the analysis of any control system is to derive its mathematical model. The various blocks shown in figure are studied in detail and their mathematical descriptions are developed.
- This would help in understanding the working of the complete system and also to implement control strategies.

Block diagram



Mathematical model

- The equation of an oven can be written by :

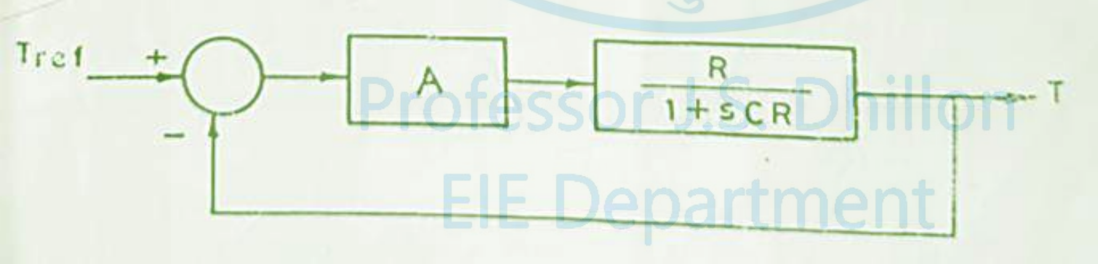
$$\theta = C\frac{dT}{dt} + \left(\frac{1}{R} \times T\right)$$

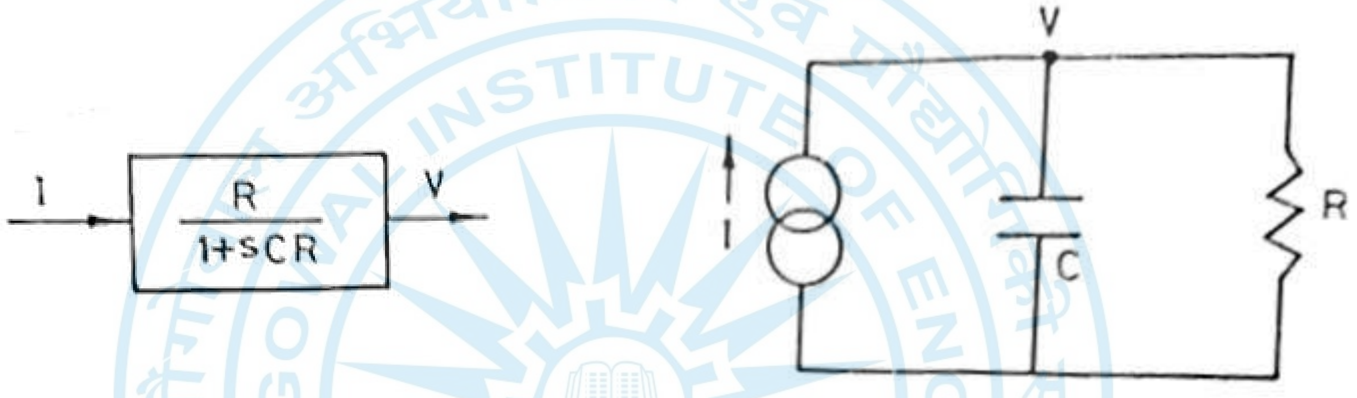
with the initial condition $T(t = 0) = T_{amb}$. Now, taking the Laplace transform with zero initial condition

$$\frac{T(s)}{\theta(s)} = \frac{R}{1 + sCR}$$

- An analogous electrical network and block diagram may be defined by the equation

$$I = C\frac{dV}{dt} + \frac{V}{R}$$





Electrical analog

Closed loop temperature control system

Procedure of open loop system using Proportional controller

- Keep switch S_1 to 'WAIT', S_2 to 'SET' and open 'FEEDBACK' terminals.
- Connect P output to the driver input and switch ON the unit.
- Set P potentiometer to 0.5 which gives K_P=10. Adjust reference potentiometer to read 5.0 on the DVM. This provides an input of 0.5V to the driver.
- Put switch S_2 to the 'MEASURE' position and note temperature readings every 15 sec, till the temperature becomes almost constant.
- Plot the temperature-time curve on a graph paper. Calculate T_1 and T_2 and hence write the transfer function of the oven including its driver.

Set Point

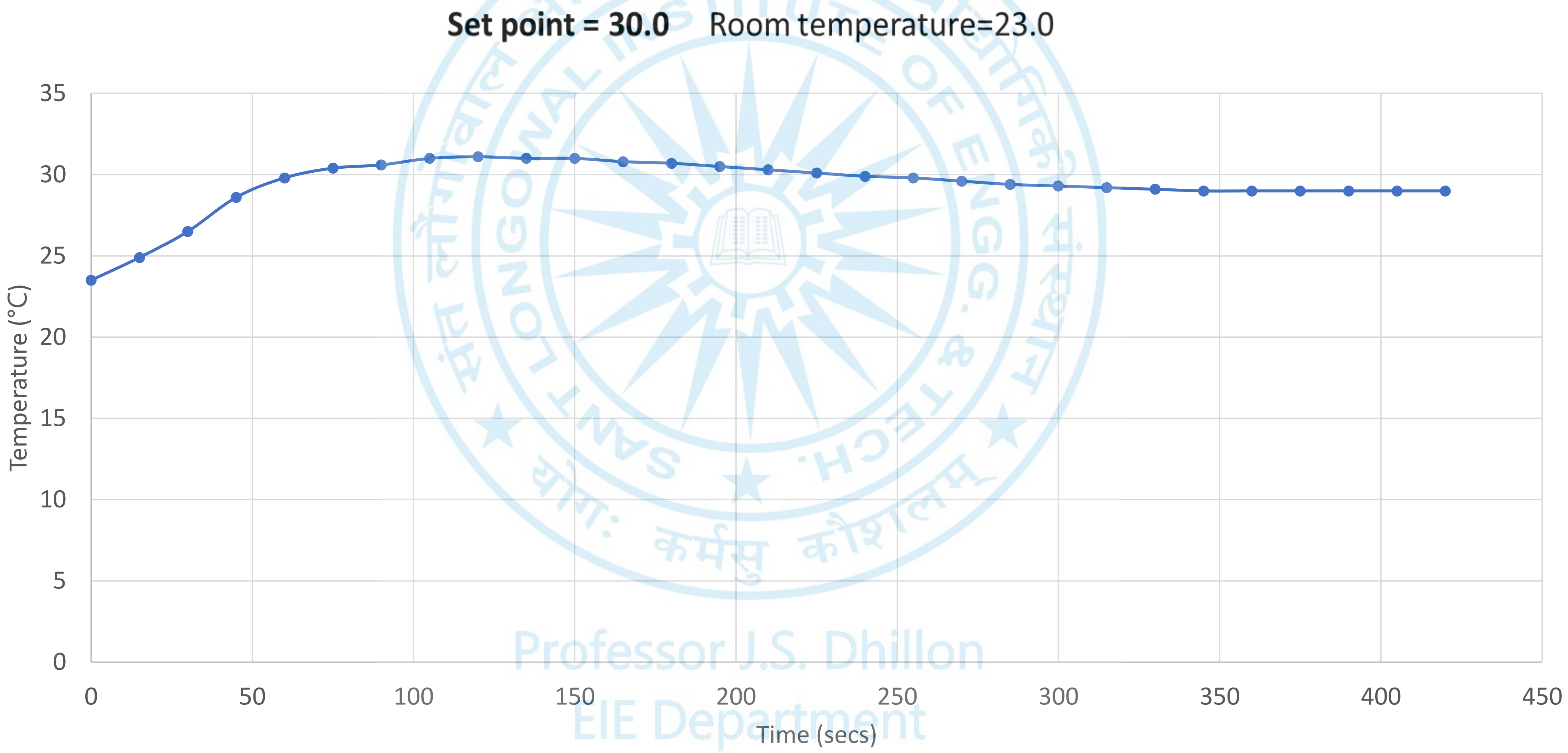

TEMPERATURE CONTROLLER SYSTEM, TCS-02
HYSTERESIS
HI
LO
SET
MEASURE
30.0
TIMER
000
REF
P
I
D
R
DRIVER
OVEN
COOL
HEAT
FEEDBACK
SENSOR GAIN
RESET
PAUSE
RUN
ON

Snapshots during experiment

Observation table of open loop system using Proportional controller

Time (secs)	Temp. (degree c)	Time (secs)	Temp. (degree c)
0	23.7	225	79.0
15	24.6	240	81.5
30	27.3	255	83.8
45	31.1	270	86.0
60	35.5	285	87.9
75	40.2	300	89.6
90	45.0	315	91.2
105	49.8	330	92.7
120	54.4	345	94.0
135	58.7	360	95.2
150	62.8	375	96.2
165	66.5	390	97.2
180	70.0	405	98.2
195	73.3	420	99.0
210	76.2		

Results of open loop system using Proportional controller



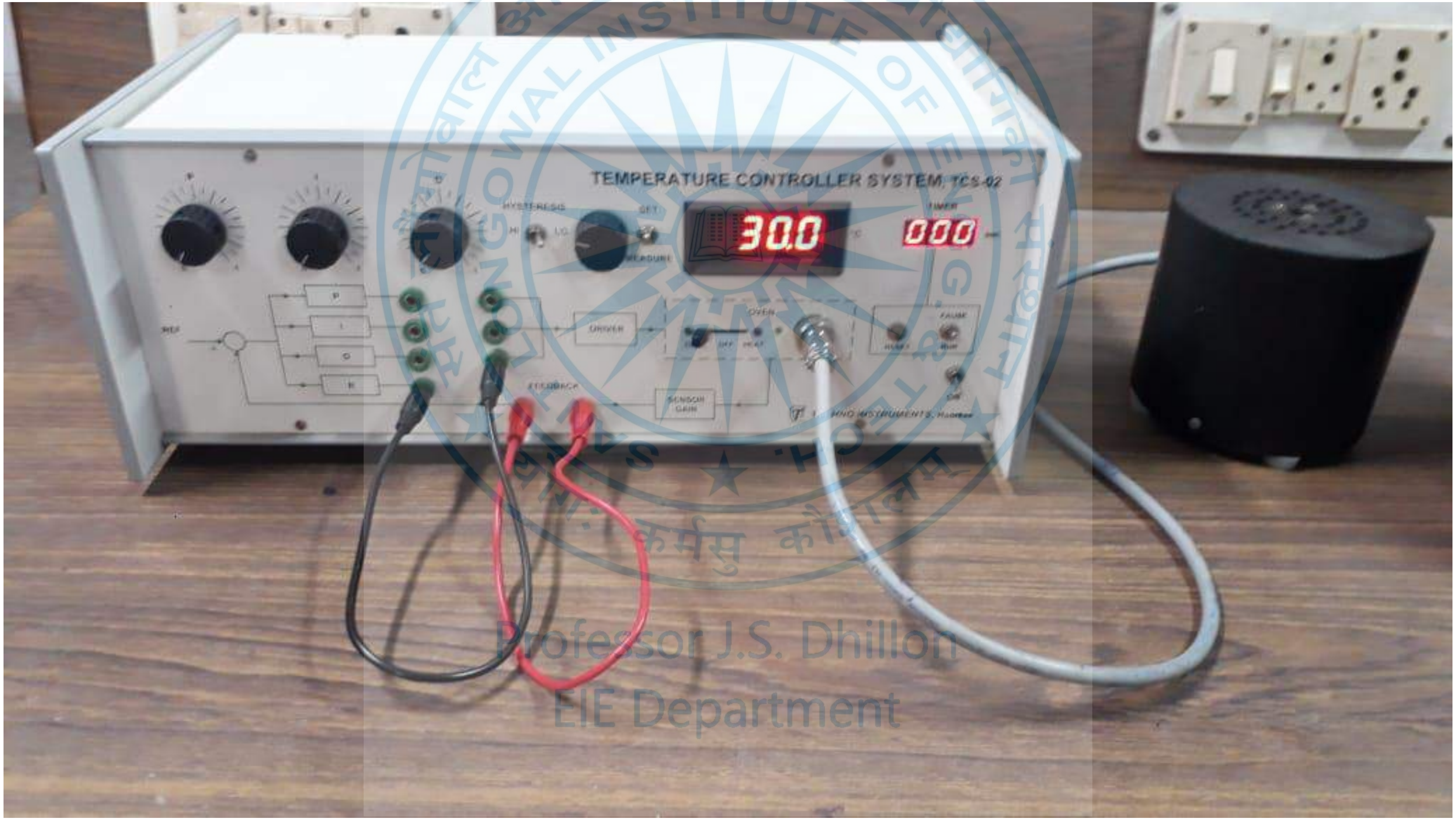
ON-OFF controller

- This controller is also referred as two position controllers, consist of a simple and inexpensive switch/relay and are, therefore, used very commonly in both industrial and domestic control systems.
- Solenoid operated two position valves are commonly used in hydraulic and pneumatic systems.
- The basic input-output behavior of this controller is shown in graph.

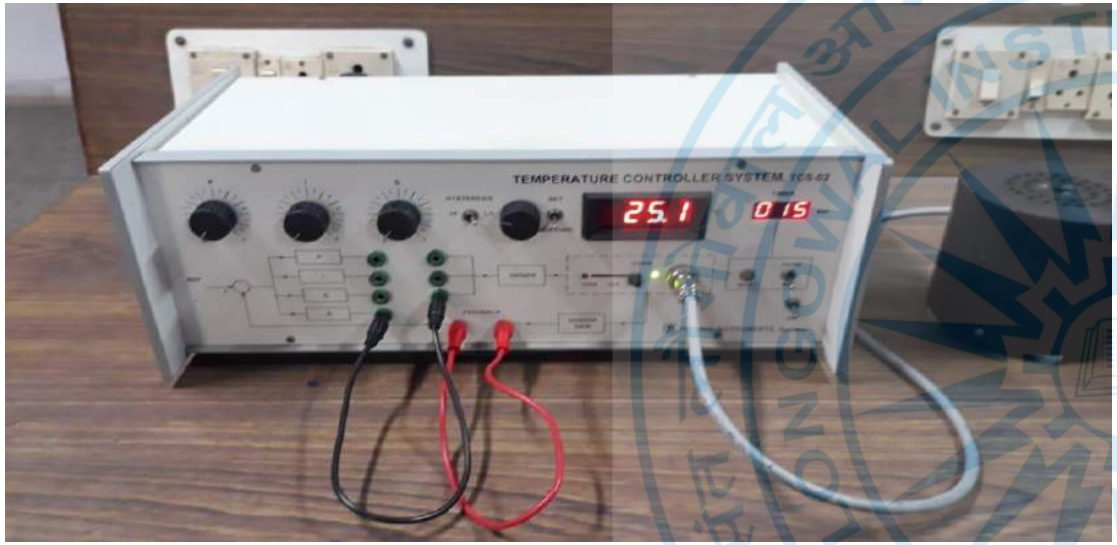
Procedure of ON-OFF controller

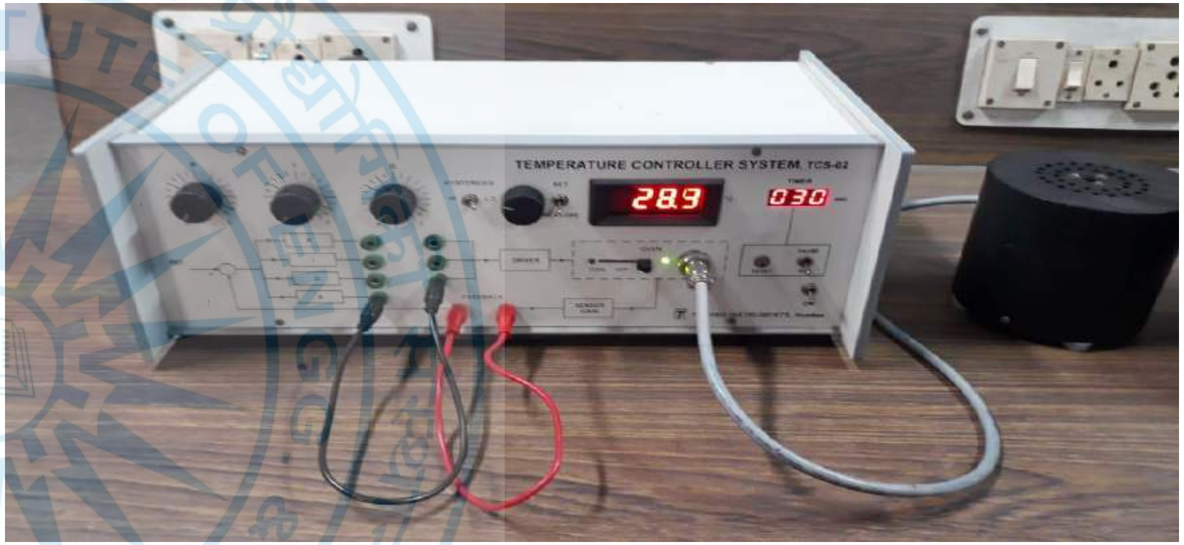
- Keep switch S_1 to 'WAIT' position and allow the oven to cool to room temperature. Short 'FEEDBACK' terminals.
- Keep switch S_2 to the 'SET' position and adjust reference potentiometer to the desired output temperature, say 60.0°C, by seeing on the digital display.
- Connect R output to the driver input. Outputs of P, D and I must be disconnected from driver input. Select 'HI' or 'LO' value of hysteresis. (First keep the hysteresis switch to 'LO')
- Switch S_2 to 'MEASURE' and S_1 to 'RUN' position. Read and record oven temperature every 15/30 sec., for about 20 minutes.
- Plot a graph between temperature and time and observe the oscillations in the steady state. Note down the magnitude of oscillations.
- Repeat above steps with the 'HI' setting for hysteresis and observe the rise time, steady -state error and percent overshoot.

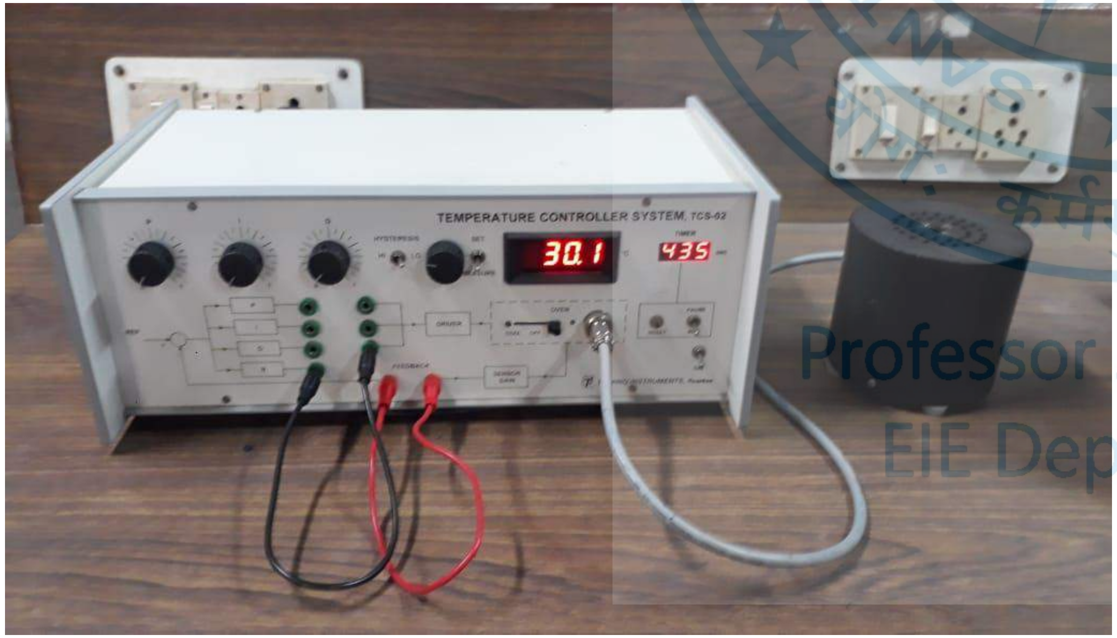
Set Point

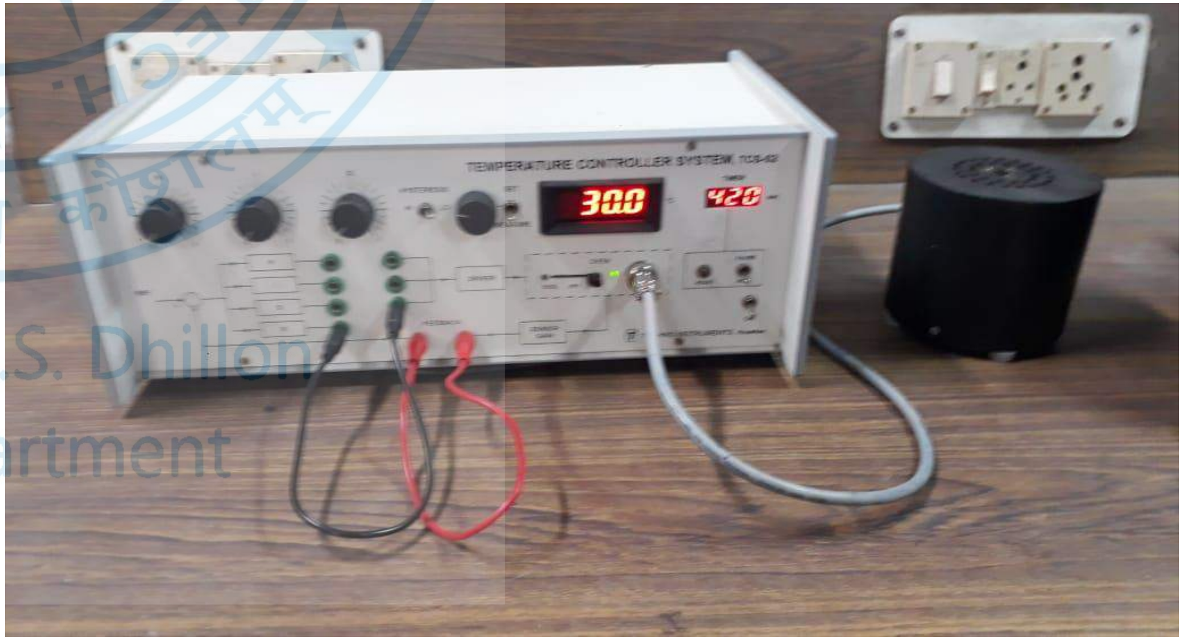


Snapshots during experiment





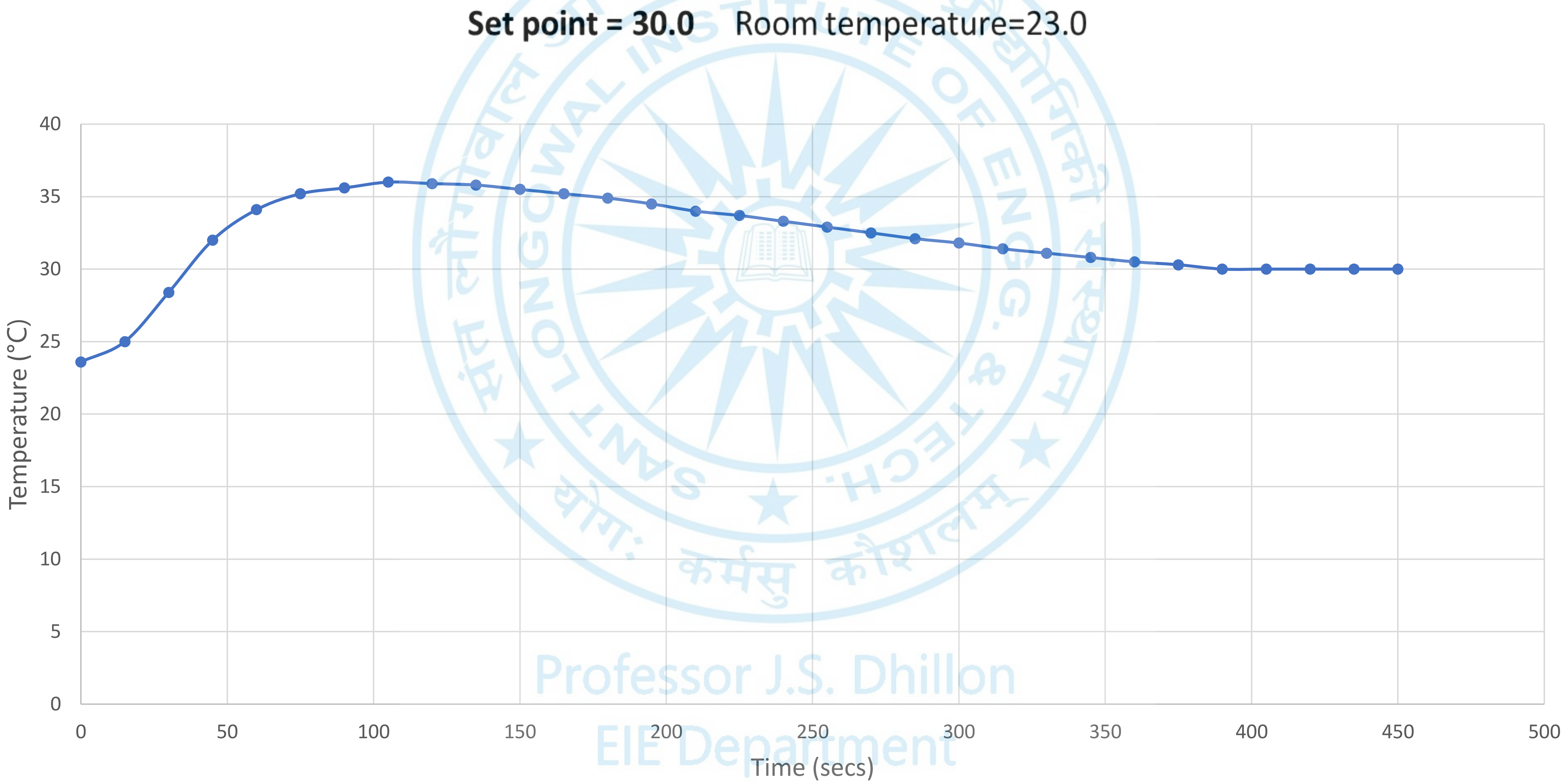




Observation table using ON-OFF controller

Time (secs)	Temp. (degree c)	Time (secs)	Temp. (degree c)
0	23.6	240	33.3
15	25.0	255	32.9
30	28.4	270	32.5
45	32.0	285	32.1
60	34.1	300	31.8
75	35.2	315	31.4
90	35.6	330	31.1
105	36.0	345	30.8
120	35.9	360	30.5
135	35.8	375	30.3
150	35.5	390	30.0
165	35.2	405	30.0
180	34.9	420	30.0
195	34.5	435	30.0
210	34.0	450	30.0
225	33.7		

Results using ON-OFF controller



Proportional (P) controller

- *Proportional controller* is simply an amplifier of gain K_p which amplifies the error signal and passes it to actuator.

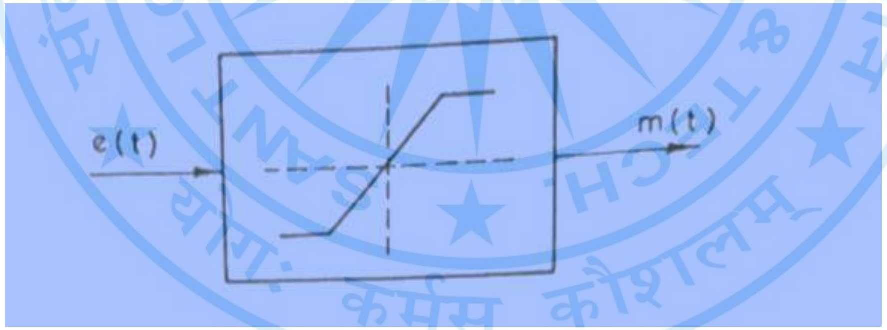


Figure: Proportional Controller with Saturation

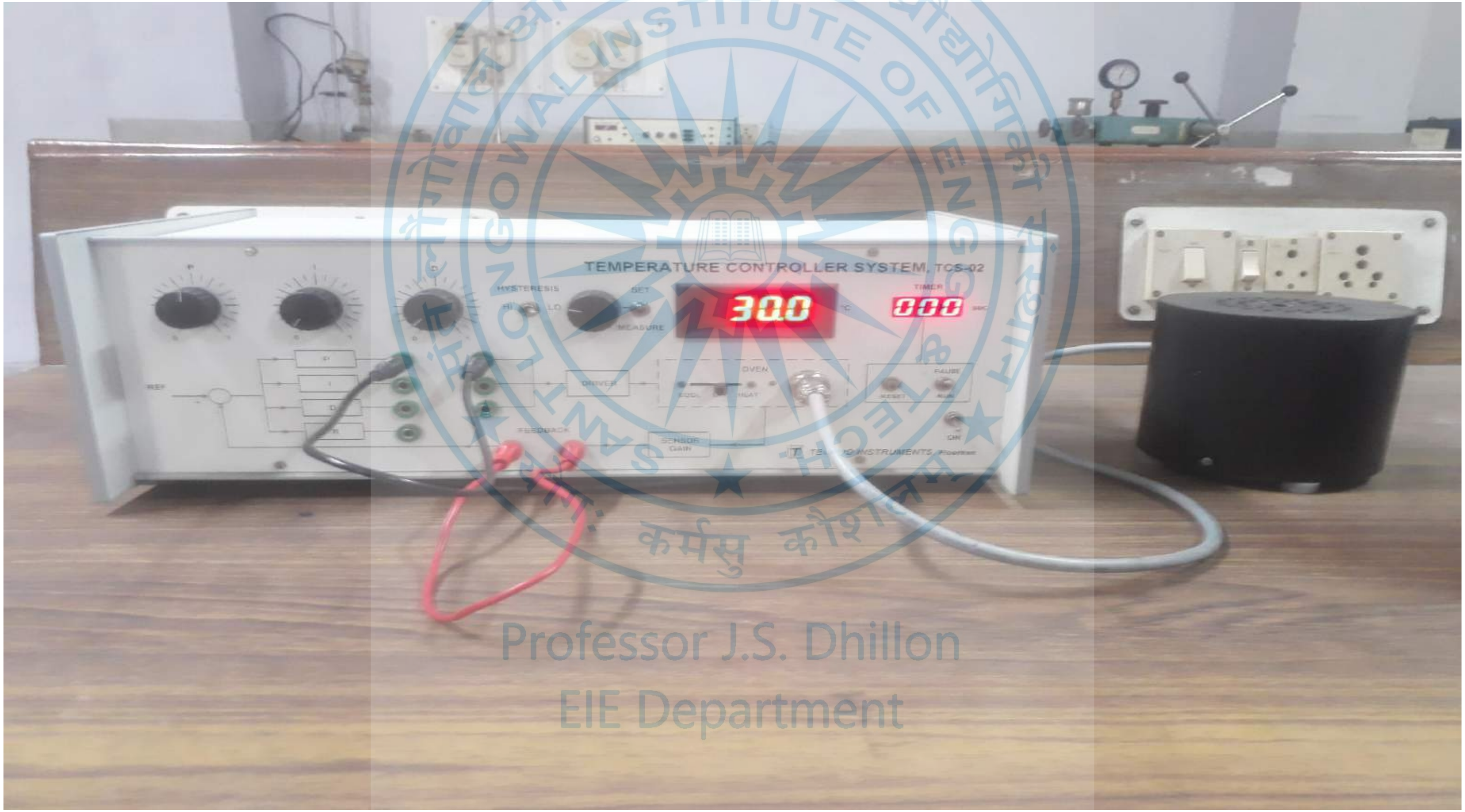
Procedure of Proportional controller

- Ziegler and Nichols suggest the value of K_P for P-Controller as:

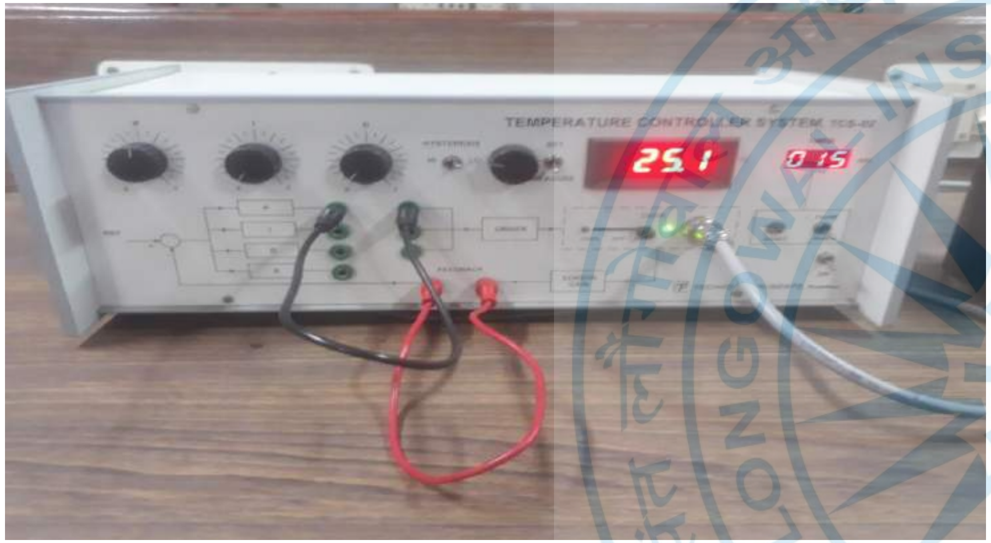
$$K_P = \left(\frac{1}{K}\right) \times \frac{T_1}{T_2}$$

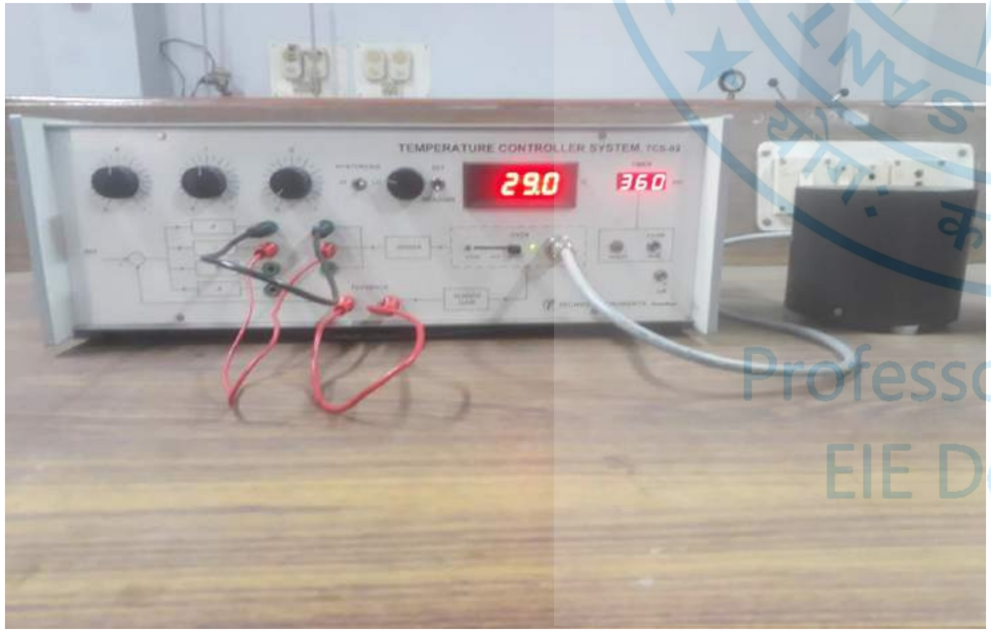
- Starting with a cool oven, keep switch S_1 to 'WAIT' position and connect P output to the driver input. Keep R, D and I outputs disconnected. Short 'FEEDBACK' terminals.
- Set P potentiometer to the above calculated value of K_P, keeping in mind that the maximum gain is 10.
- Plot the observations on a linear graph paper and observe the rise time, steady-state error and percent overshoot.

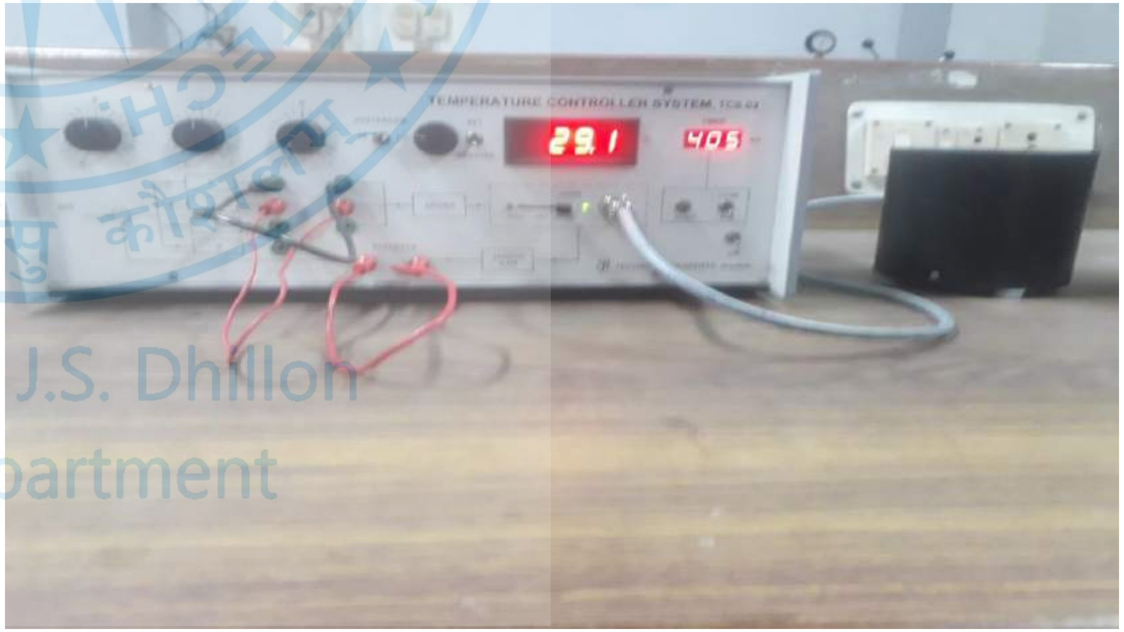
Set point


TEMPERATURE CONTROLLER SYSTEM, TCS-02
HYSTERESIS
SET
MEASURE
30.0
TIMER
000
OVEN
DRIVER
FEEDBACK
SENSOR GAIN

Snapshots during experiment



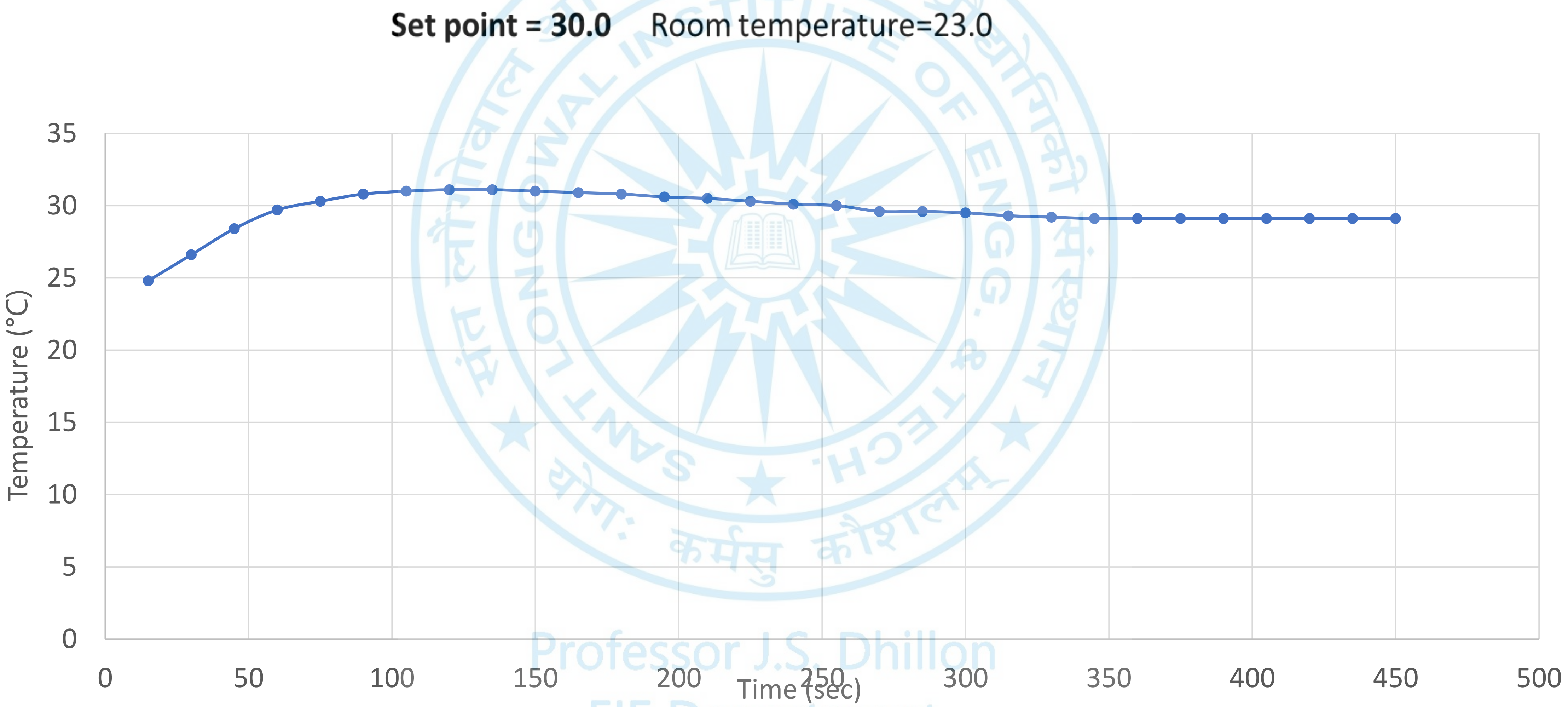




Observation table using Proportional controller

Time (secs)	Temp. (degree c)	Time (secs)	Temp. (degree c)
0	23	240	28.8
15	25.2	255	28.7
30	26.6	270	28.7
45	28.1	285	28.6
60	29.1	300	28.6
75	29.6	315	28.6
90	29.9	330	28.6
105	30.0	345	28.6
120	30.0	360	28.6
135	29.9	375	28.6
150	29.8	390	28.6
165	29.7	405	28.6
180	29.5	420	28.6
195	29.3	435	28.6
210	29.2	450	28.6
225	29.0		

Results using Proportional controller



Proportional-Integral (PI) controller

- Mathematical equation of such a controller is given by :

$$m(t) = K_p e(t) + K_I \int_0^t e(t)dt = K_p e(t) + \frac{1}{T_I}\int_0^t e(t)dt$$

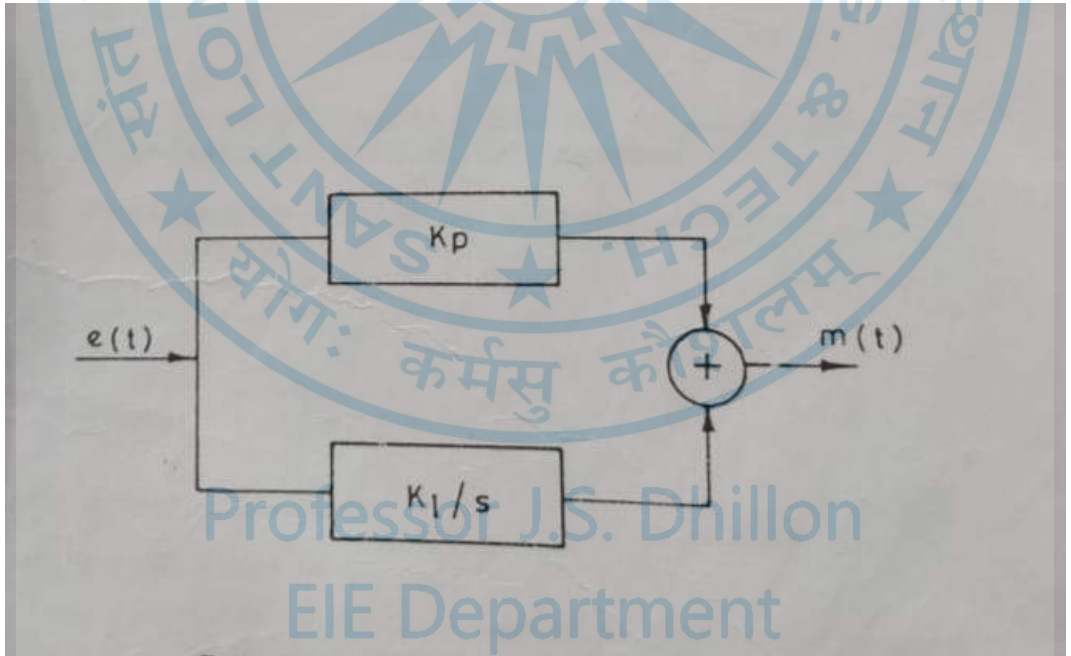


FIG. 7. P-I CONTROLLER

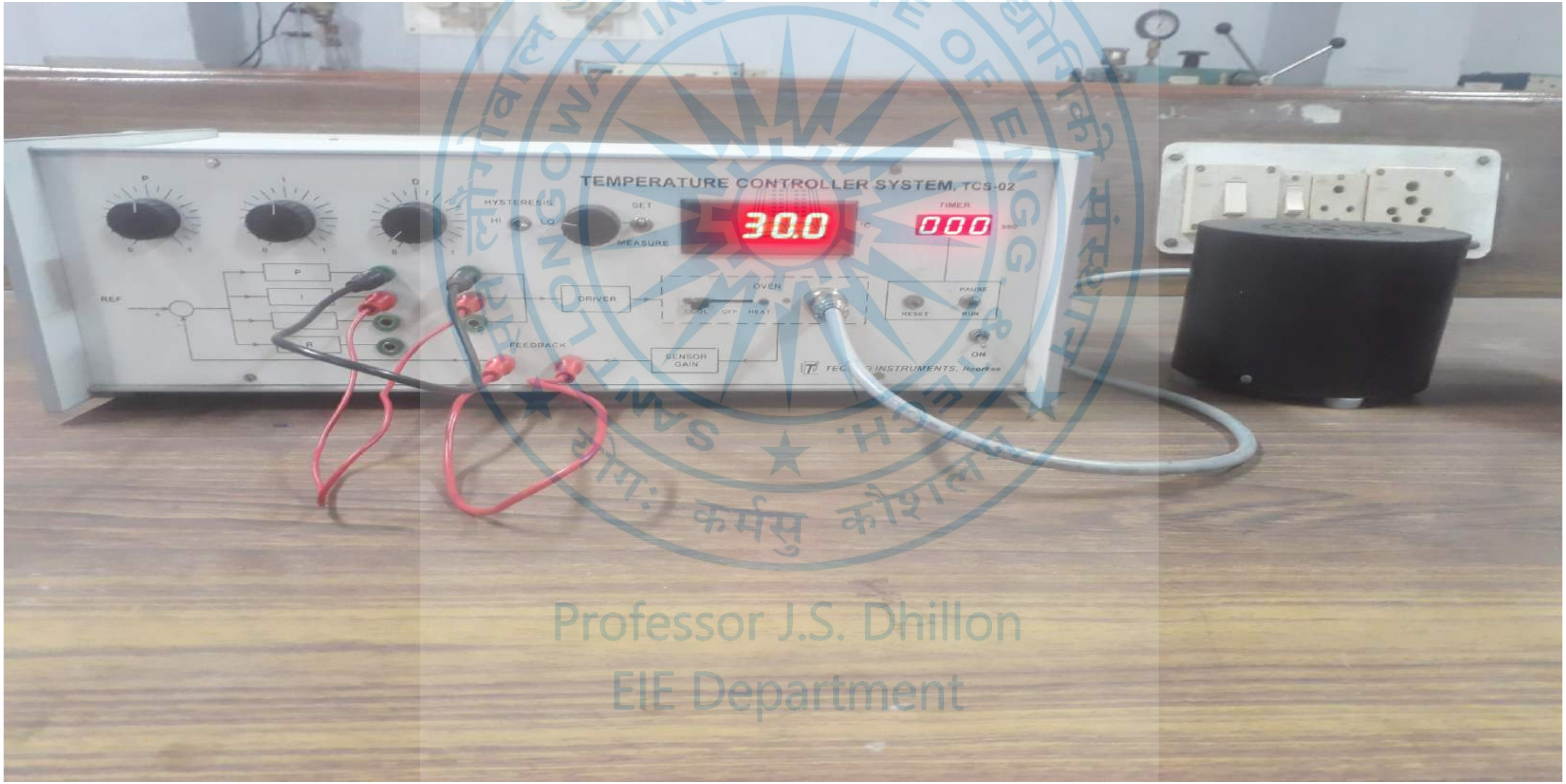
Procedure of Proportional-Integral (PI) controller

- Ziegler and Nichols suggested the value of K_P and K_1 for P-I controller as

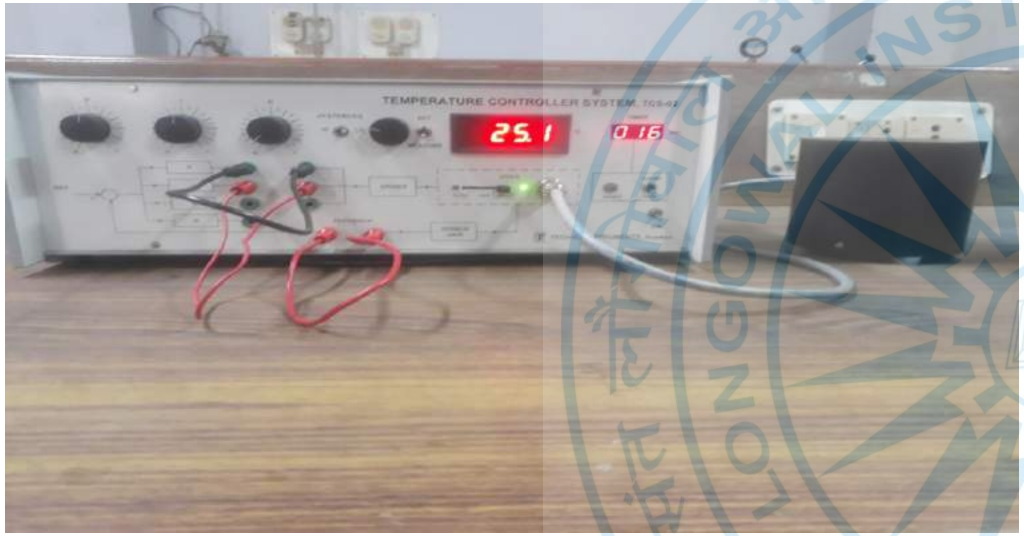
$$K_P = \left(\frac{0.9}{K}\right) \times \frac{T_1}{T_2};\ T_1 = \frac{1}{K_1} = 3.3T_2;\ giving\ K_1 = \frac{1}{3.3T_2}$$

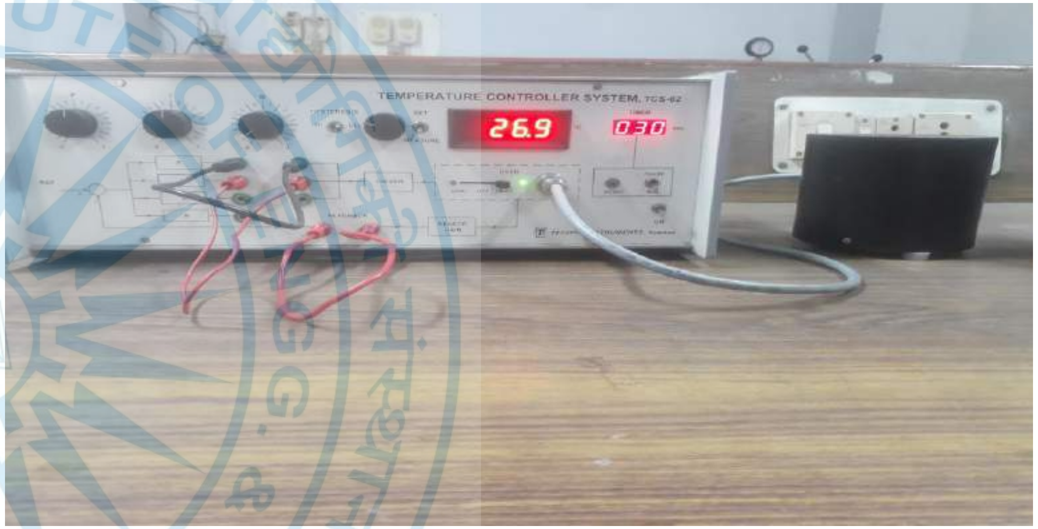
- Starting with a cool oven, keep switch S_1 to 'WAIT' connect P and I outputs to driver input and disconnected R and D outputs. Short feedback terminals.
- Set P and I potentiometers to the above values of K_P and K_1 respectively, keeping in mind that the maximum value of K_P is 20 and that of K_1 is 0.024.
- Select and set the desired temperature to say 60.0°C.
- Keep switch S_1 to 'RUN' position and record temperature readings as before.
- Plot the response on a graph paper and observe the steady state error and percent overshoot.

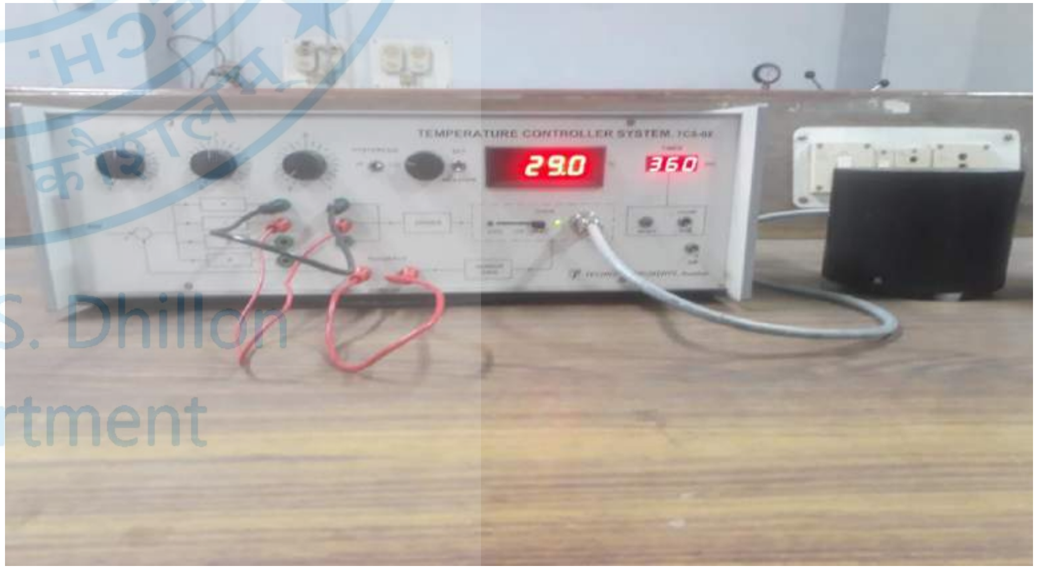
Set point


TEMPERATURE CONTROLLER SYSTEM, TCS-02
HYSTERESIS
SET
MEASURE
30.0
TIMER
000
REF
DRIVER
OVEN
FEEDBACK
SENSOR GAIN
RESET
RUN
PAUSE
ON

Snapshots during experiment



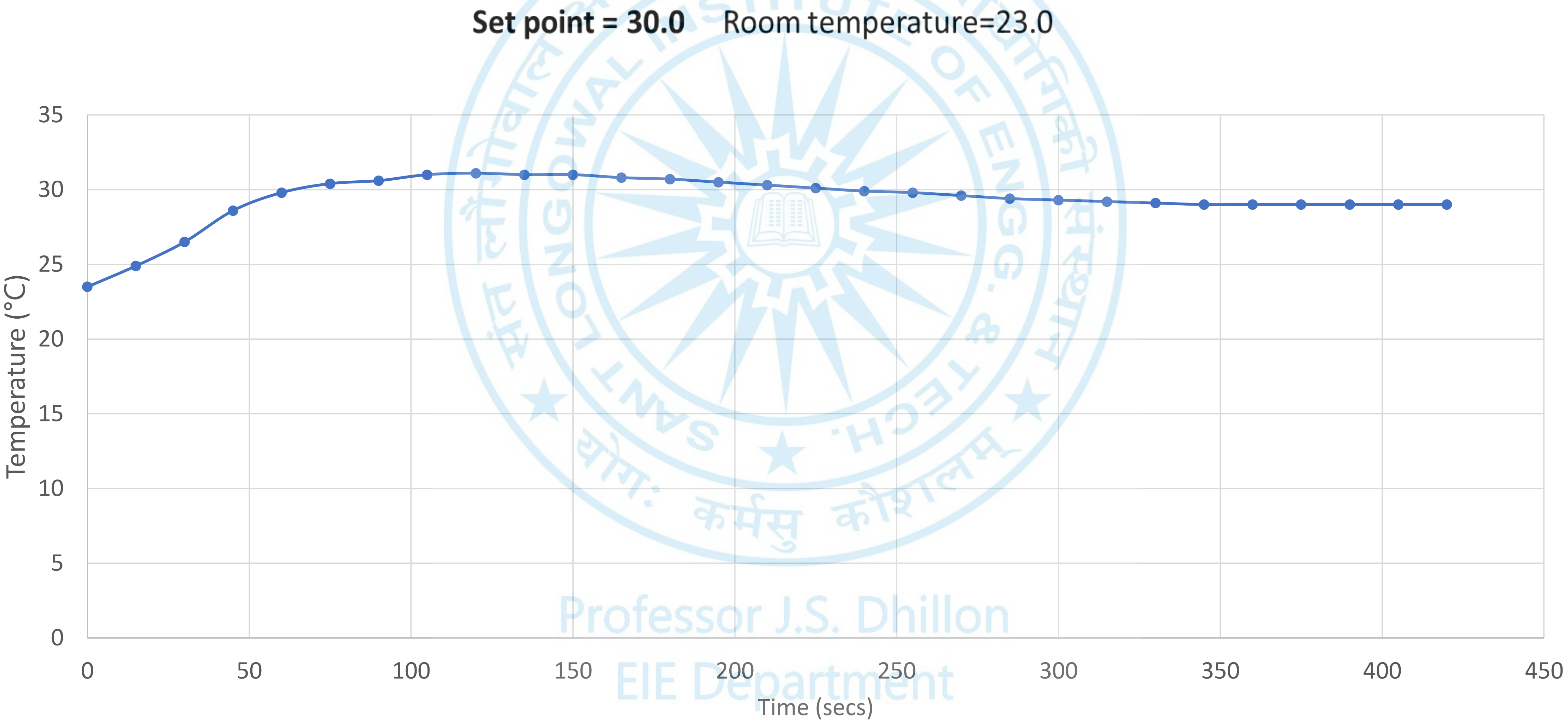




Observation table using Proportional-Integral (PI) controller

Time (secs)	Temp. (degree c)	Time (secs)	Temp. (degree c)
0	23.5	240	29.9
15	24.9	255	29.8
30	26.5	270	29.6
45	28.6	285	29.4
60	29.8	300	29.3
75	30.4	315	29.2
90	30.6	330	29.1
105	31.0	345	29.0
120	31.1	360	29.0
135	31.0	375	29.0
150	31.0	390	29.0
165	30.8	405	29.0
180	30.7	420	29.0
195	30.5	435	29.0
210	30.3	450	29.0
225	30.1		

Results using Proportional-Integral (PI) controller



Proportional-Integral-Derivative (PID) controller

- Mathematical equations governing the operation of this controller is as

$$m(t) = K_p e(t) + K_I \int_0^t e(t)dt + K_D \frac{de(t)}{dt}$$

$$m(t) = K_p e(t) + \frac{1}{T_I} \int_0^t e(t)dt \ + T_D \frac{de(t)}{dt}$$

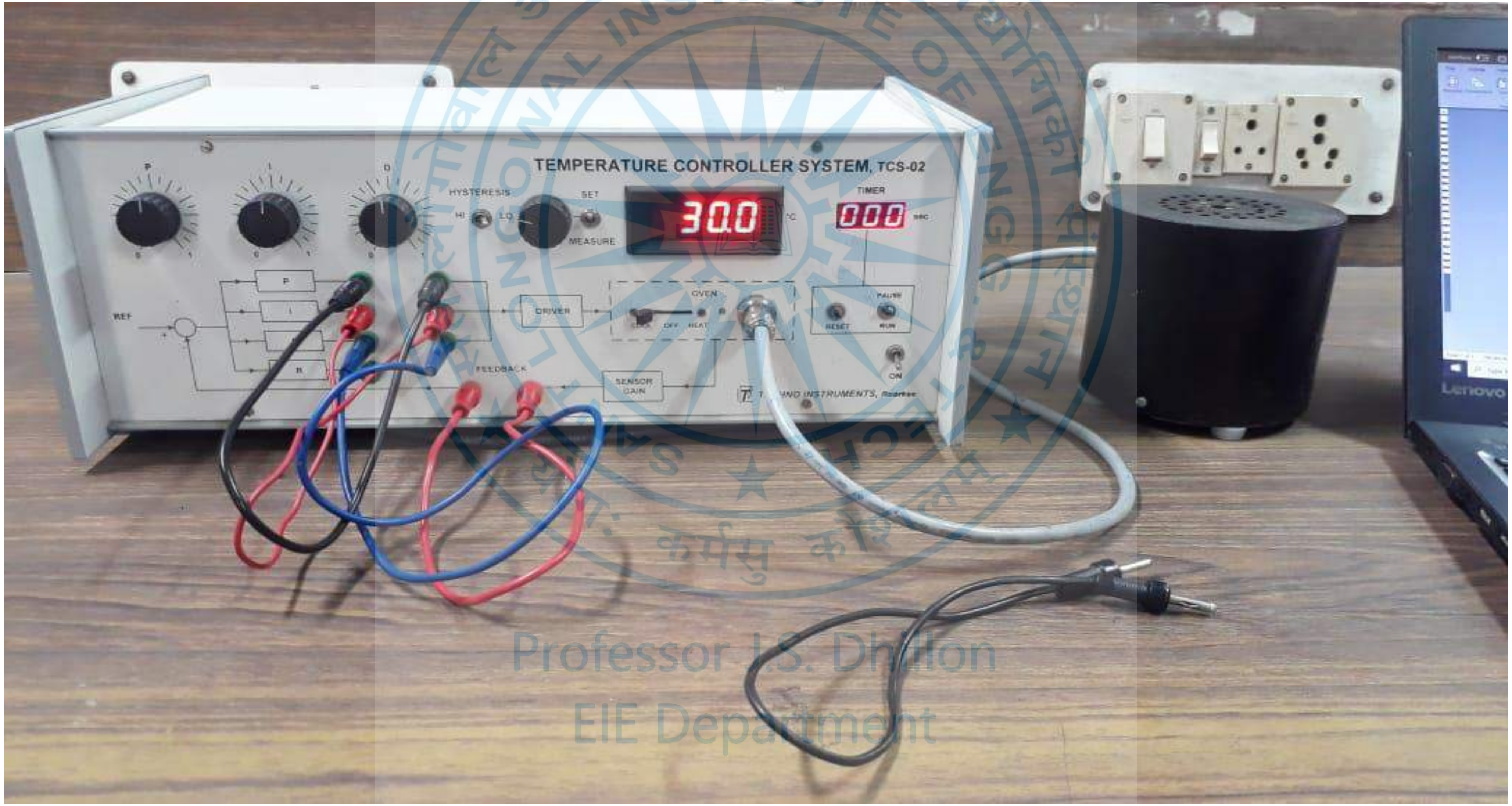
Procedure of Proportional-Integral-Derivative (PID) controller

- Ziegler and Nichols suggested the value of K_P and K_D and K_1 for tis controller as

$$K_P = \left(\frac{1.2}{K}\right) \times \frac{T_1}{T_2};\ T_1 = \frac{1}{K_1} = 2T_2;\ giving\ K_1 = \frac{1}{2T_2}$$

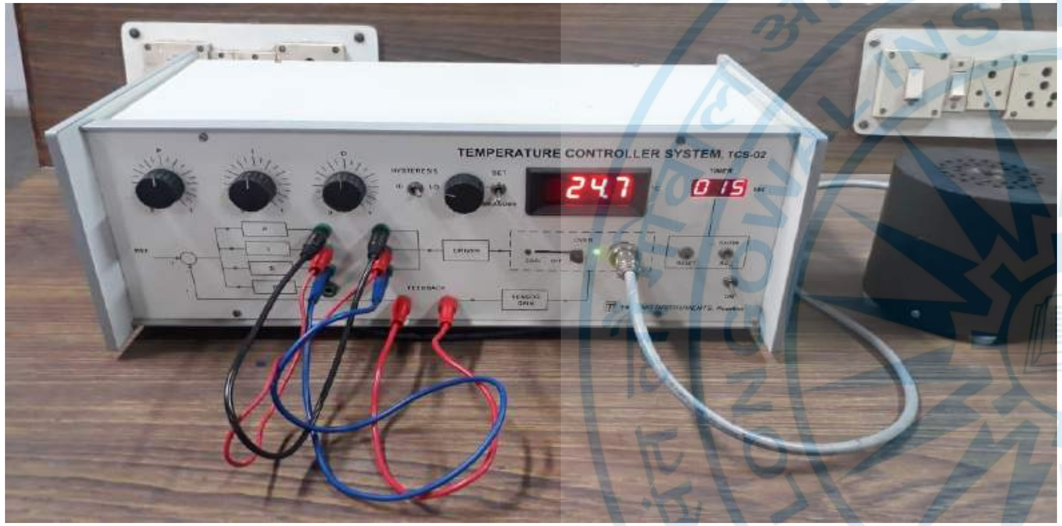
- Starting with a cool oven, keep switch S_1 to 'WAIT' connect P, D and I outputs to driver input. Keep R output disconnected. Short feedback terminals.
- Set P, I and D potentiometers according to the above calculated values of K_P, K_1 K_D keeping in mind that the maximum value for these are 20, K_P is 20, 0.024 and 23.5 respectively.
- Select and set the desired temperature, say 60.0°C.
- Switch S_1 to 'RUN' and record temperature readings.
- Plot the response on a linear graph paper and observe the rise time, steady state error and percent overshoot.
- Compare the results of the various controller options.

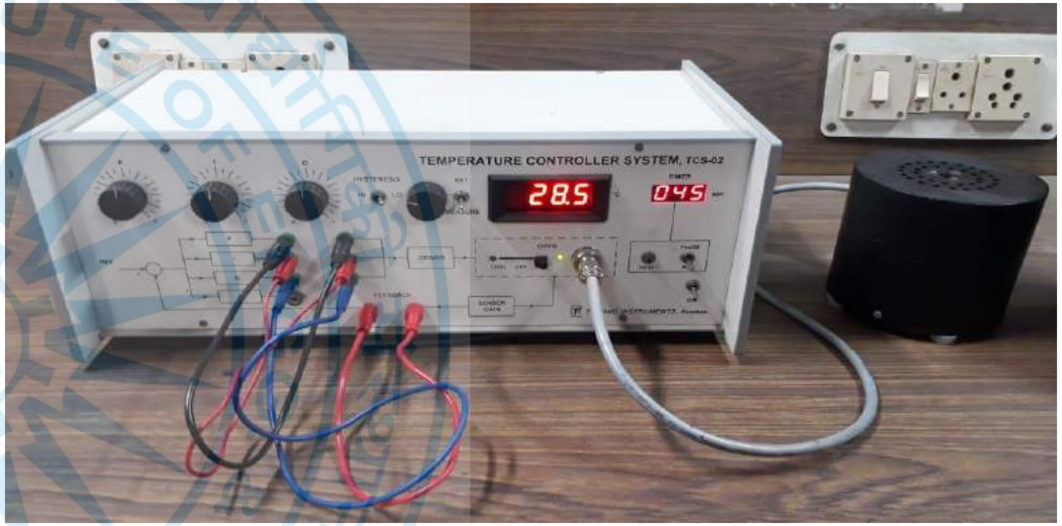
Set point

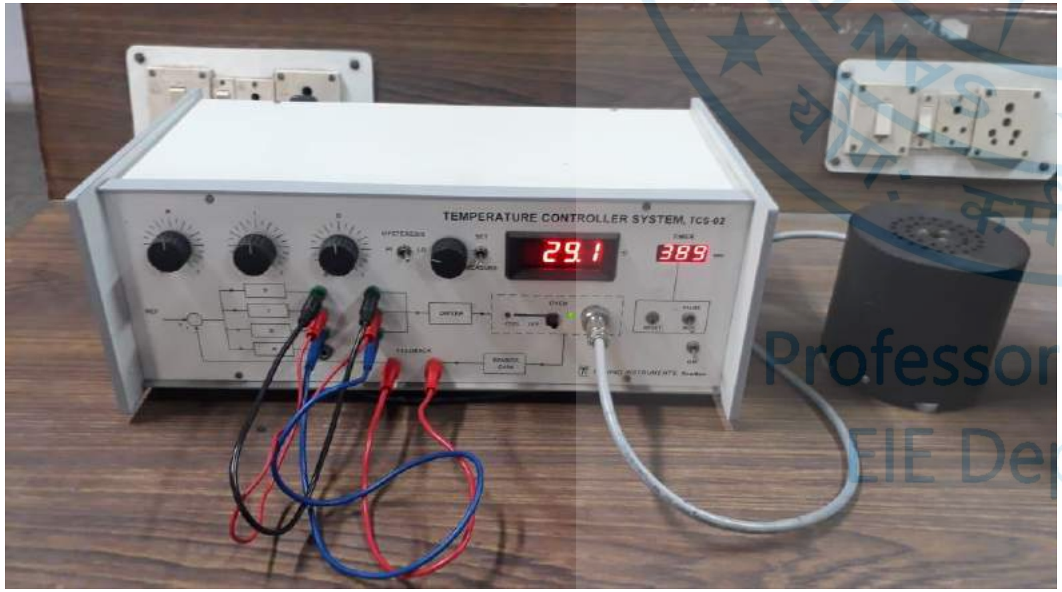


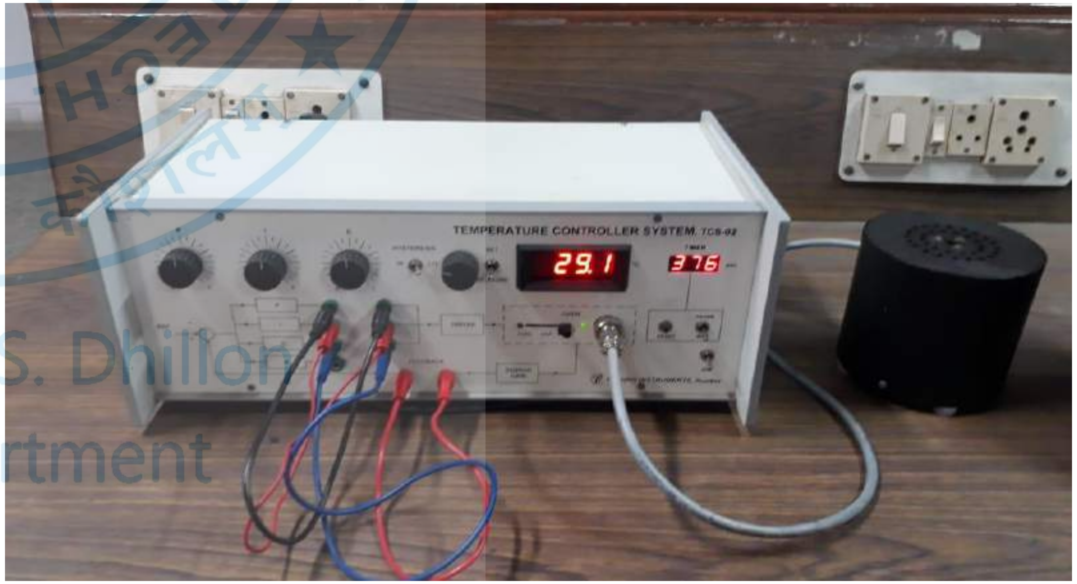
TEMPERATURE CONTROLLER SYSTEM, TCS-02
HYSTERESIS
HI
LO
SET
MEASURE
30.0
TIMER
000
OVEN
DRIVER
REF
FEEDBACK
SENSOR GAIN
RESET
PAUSE
RUN
ON

Shots during experiment





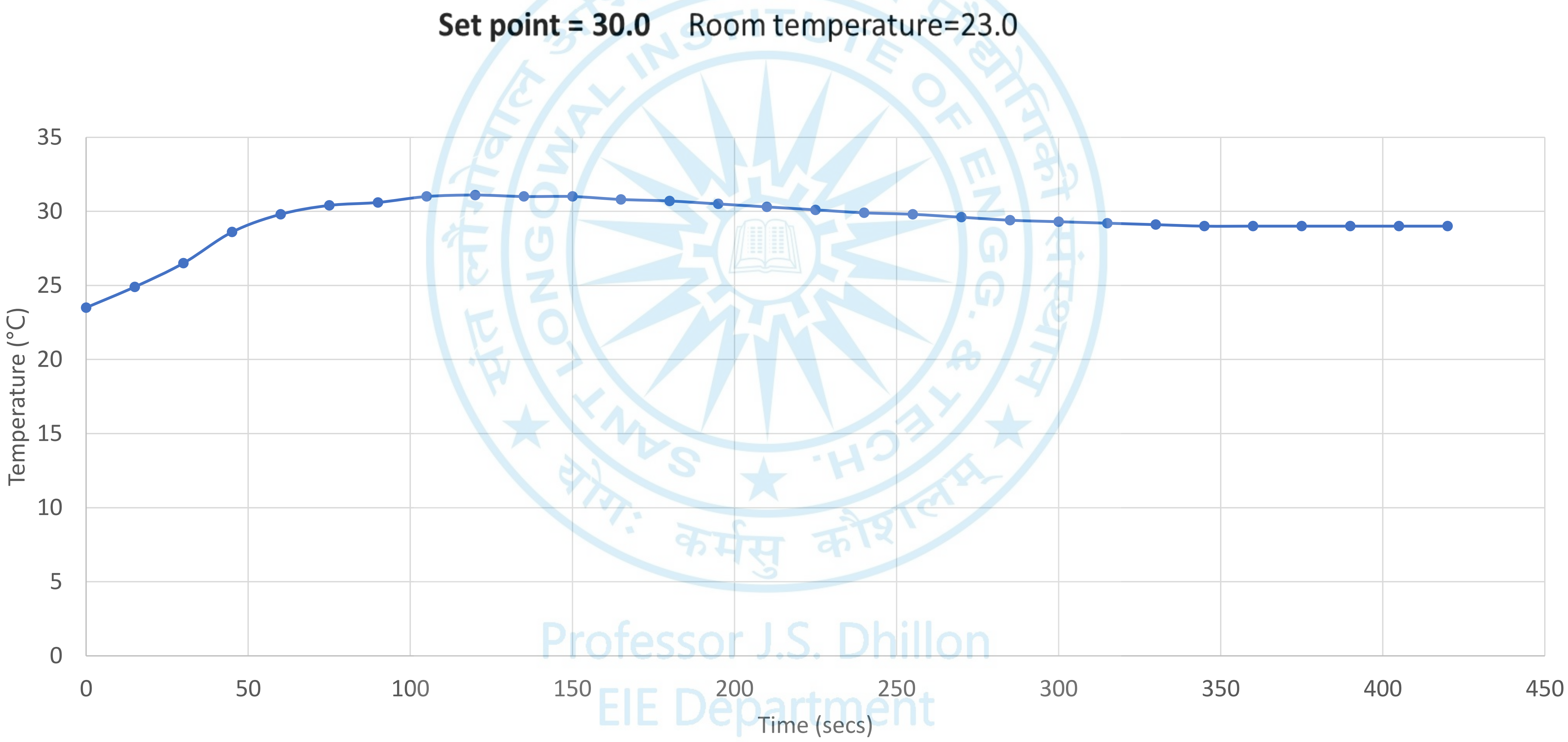




Observation table using Proportional-Integral-Derivative (PID) controller

Time (secs)	Temp. (degree c)	Time (secs)	Temp. (degree c)
0	23.5	240	30.1
15	24.8	255	30.0
30	26.6	270	29.6
45	28.4	285	29.6
60	29.7	300	29.5
75	30.3	315	29.3
90	30.8	330	29.2
105	31.0	345	29.1
120	31.1	360	29.1
135	31.1	375	29.1
150	31.0	390	29.1
165	30.9	405	29.1
180	30.8	420	29.1
195	30.6	435	29.1
210	30.5	450	29.1
225	30.3		

Results using Proportional-Integral-Derivative (PID) controller



Professor J.S. Dhillon
EIE Department

Experiment- 02

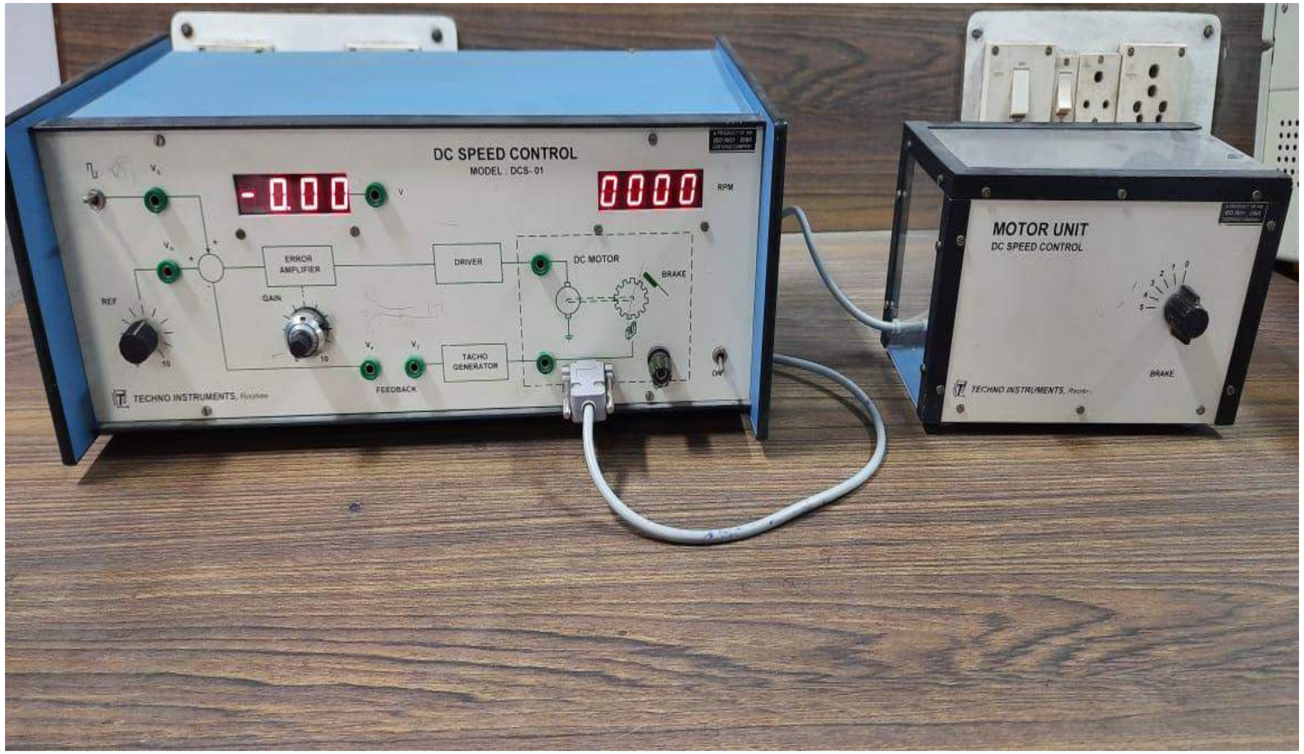
PCEE- 613, Control systems Lab.

GEE-2018

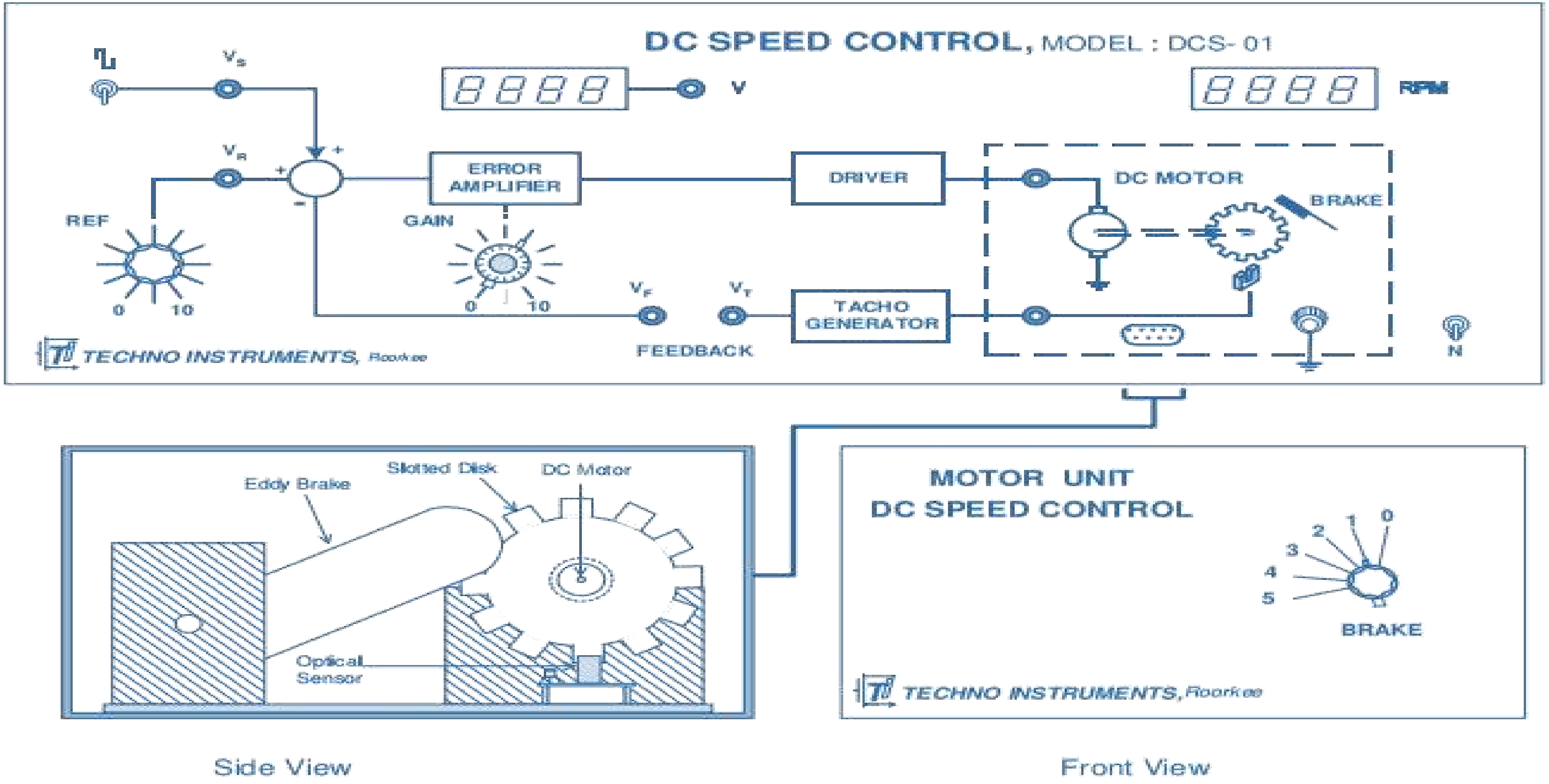
Experiment 02: To study the performance characteristics of a DC motor speed control system.

Apparatus Required:

- DC motor (12V, 200mA, 3000rpm)
- Tachogenerator
- Error detector and forward gain
- Driver circuit
- Power and signal source
- Multimeter
- Connecting leads



Panel drawing



Panel drawing DC Speed Control, Model DCS-01

System schematic

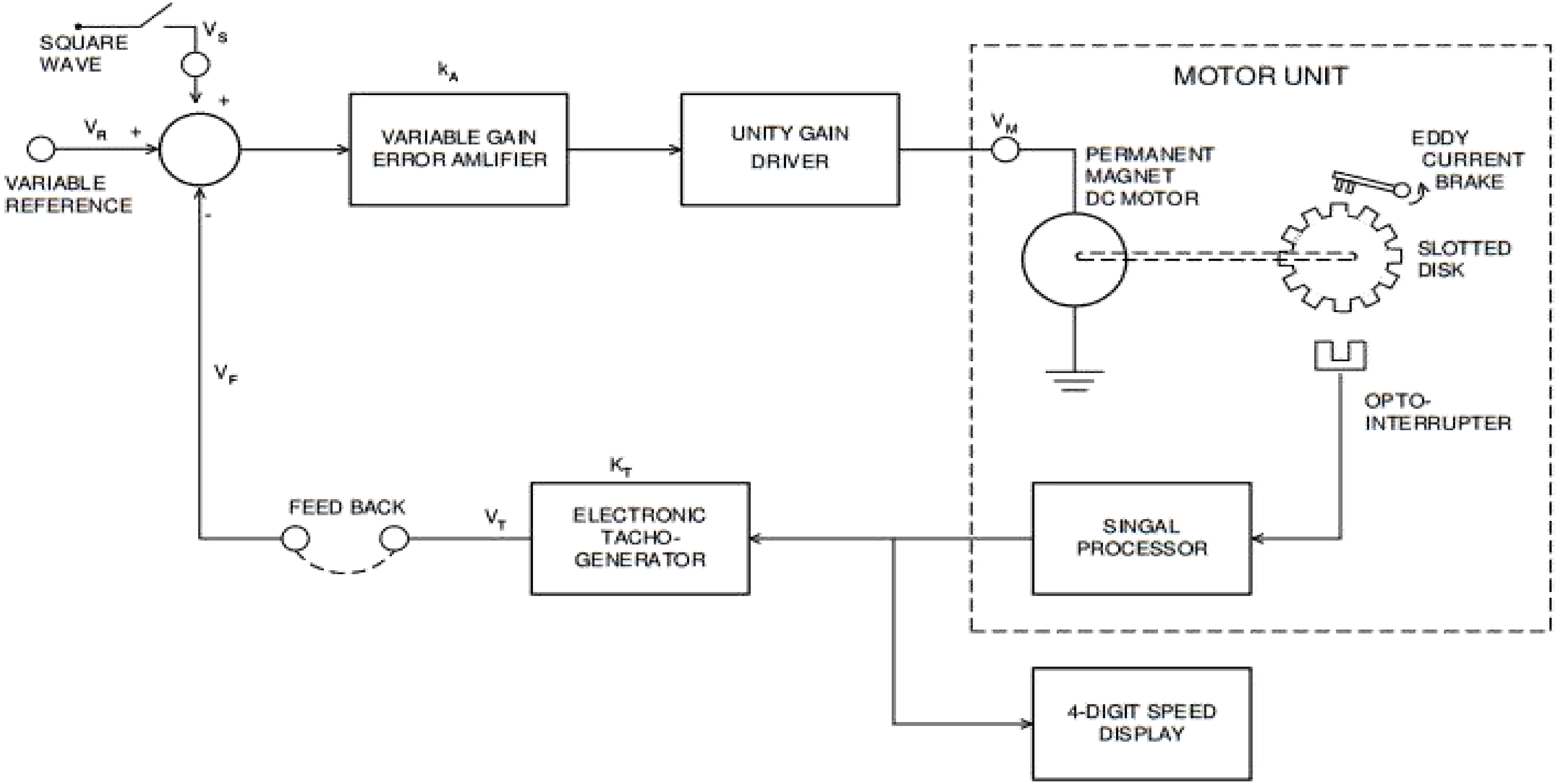


Fig.1 System schematic

Theory & Mathematical model

A basic block diagram of the d.c. motor speed control system is shown in Fig. 2. In order to evaluate the system performance, it is necessary to compute the overall transfer function in terms of the transfer functions of the different blocks. To start with, the transfer function of an armature controlled d.c. motor of Fig. 3

$$\frac{\theta(s)}{V(s)} = \frac{K_M}{s(sT+1)}$$

where K_M is motor gain constant, and T is the mechanical time constant. Note that a permanent magnet d.c. motor should behave similar to a shunt motor with constant field excitation. Considering motor speed ω rad/sec (=dθ/dt) as the output variable, the forward path transfer function may be written as,

$$G(s) = \frac{\omega(s)}{V_E(s)} = K_A \cdot \frac{K_M}{(sT+1)} \quad (1)$$

where K_A is the gain of amplifier. Again, the tachogenerator transfer function (or gain) may be written as,

$$H(s) = \frac{V_T(s)}{\omega(s)} = K_T$$

Block diagram

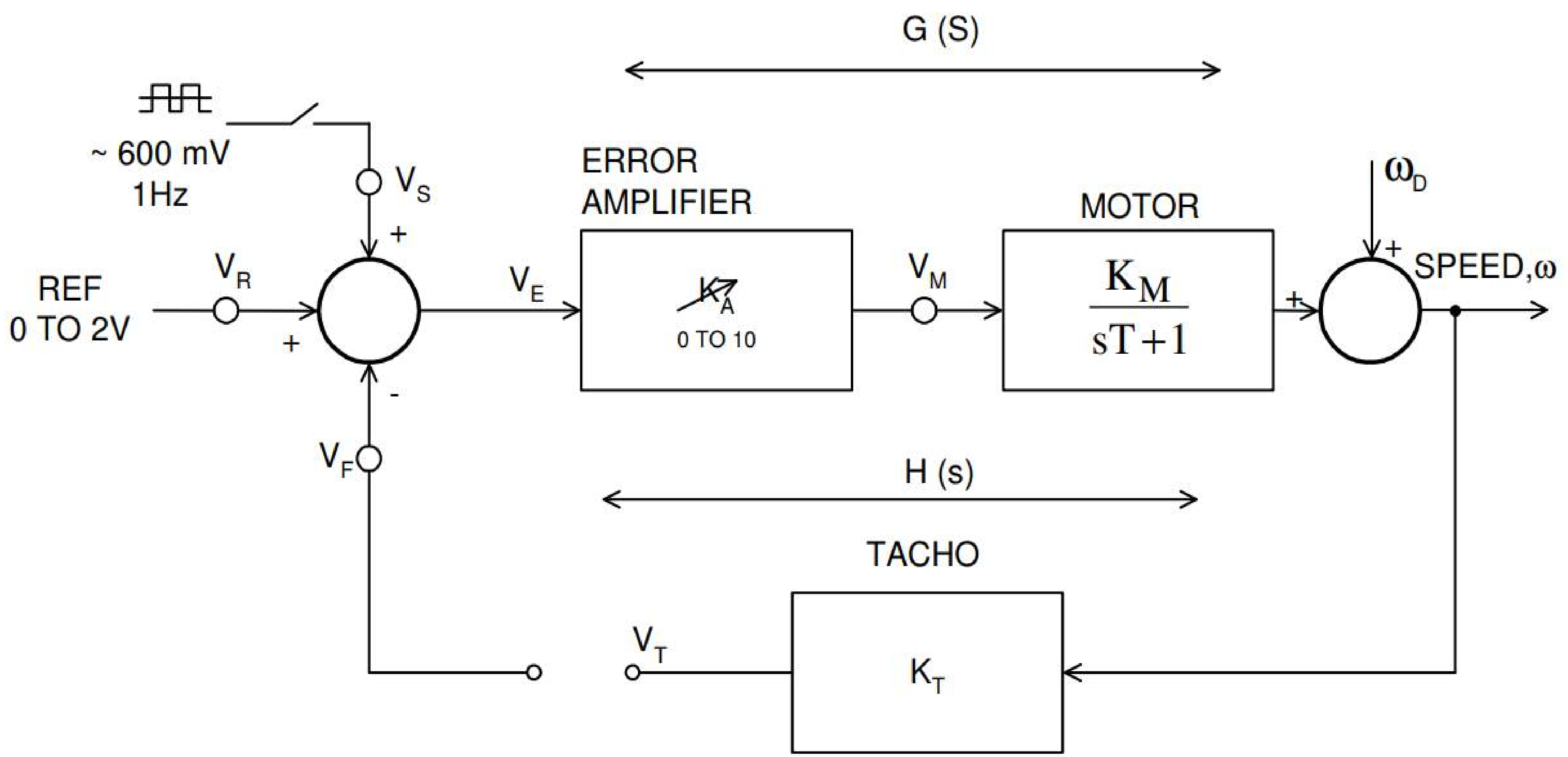
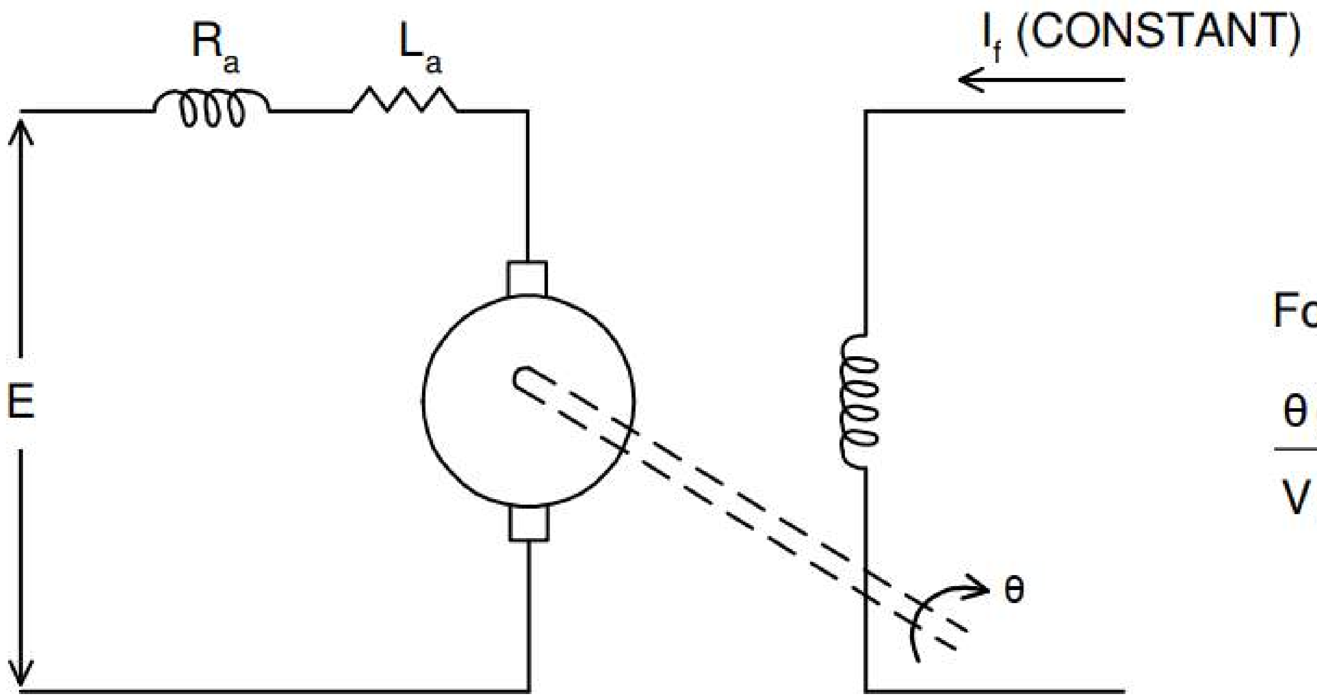


Fig.2 Block diagram

Block diagram



For $L_a \rightarrow 0$

$$\frac{\theta(s)}{V(s)} = \frac{K_M}{S(s\tau + 1)}$$

Fig. 3 Armature controlled DC motor

Procedure

- Set V_R=1 Volt and K_A=3. The motor may be running at a low speed. Record speed N in rpm, and the Tachogenerator output V_T.
- Repeat with V_R=1 and K_A=4, 5, ..., 10, and tabulate measured motor voltage V_M (=$V_R K_A$), steady state motor speed N in rpm (or ω_{ss}=N×2π/60 in radians/sec.) and tachogenerator output V_T..
- Plot N vs. V_M, and V_T vs. N. Obtain K_M and K_T from the linear region of the curves (see Fig. 5).

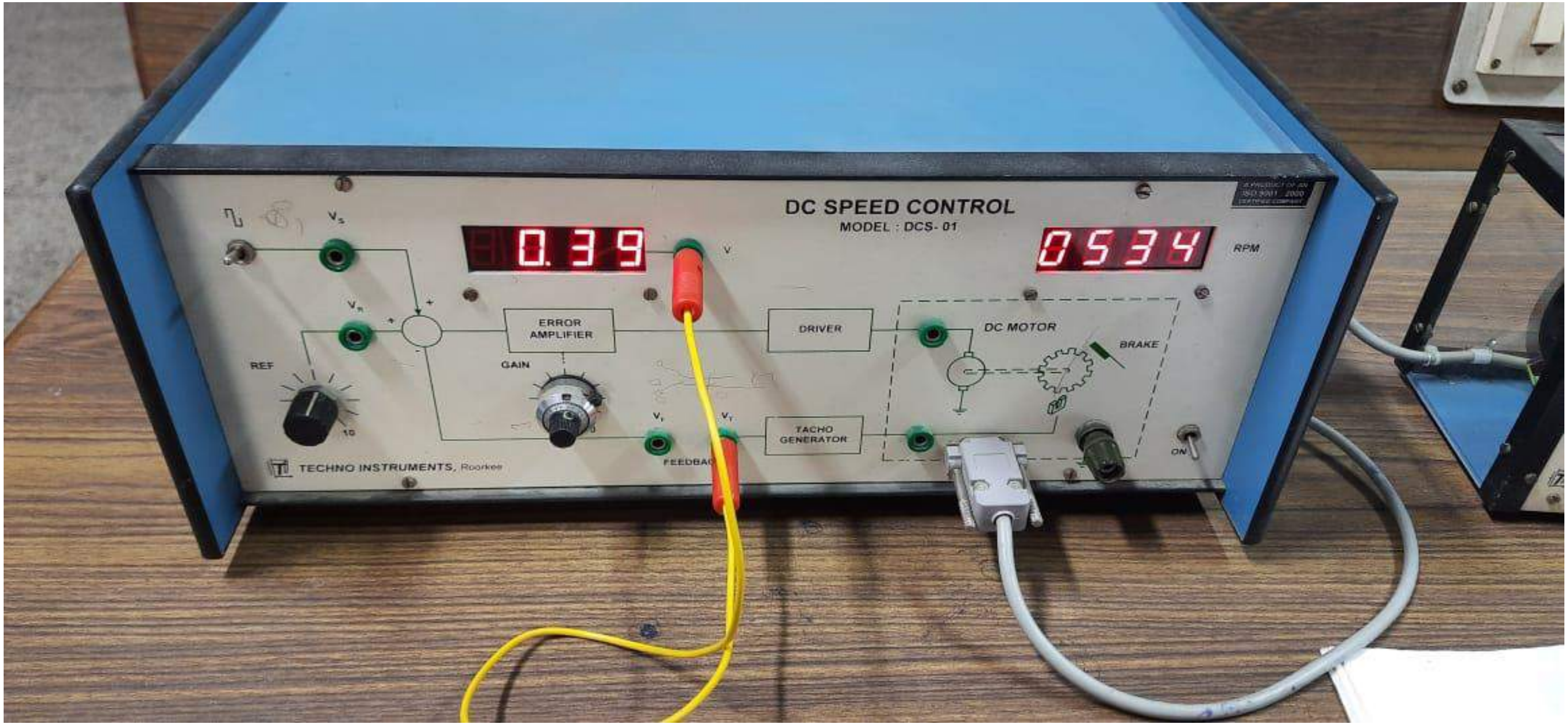
$$\text{Motor gain constant, } K_M = \frac{\text{shaft speed in rad/ sec, } \omega_{ss}}{\text{Motor voltage, } V_M}\text{, and}$$

$$\text{Tachogenerator gain, } K_T = \frac{V_T}{\omega_{ss}} \quad \frac{\text{volt - sec}}{\text{rad}}$$

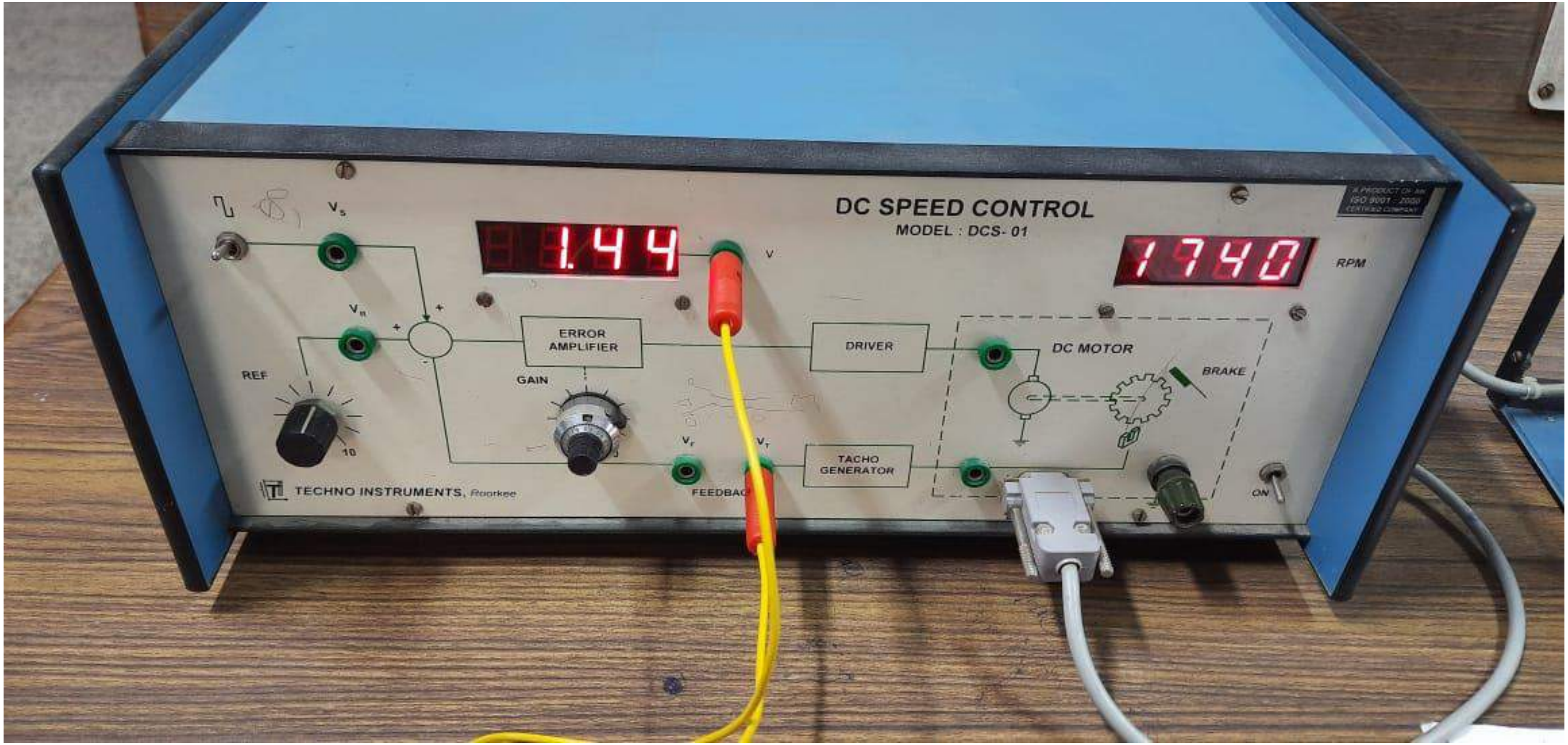
- Obtain the motor transfer function using

$$G(s) = \frac{K_M}{sT + 1}$$

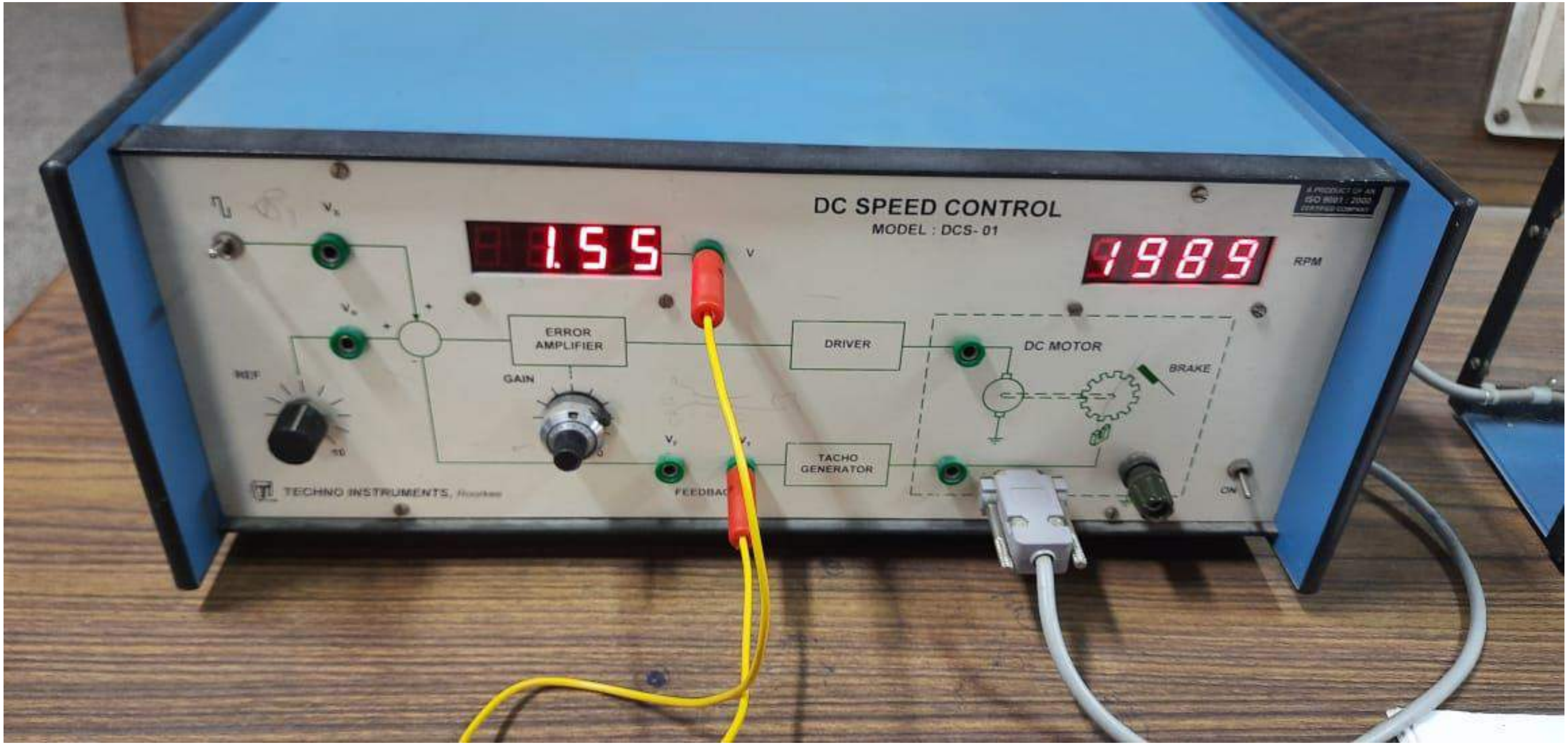
Snapshots during experiment



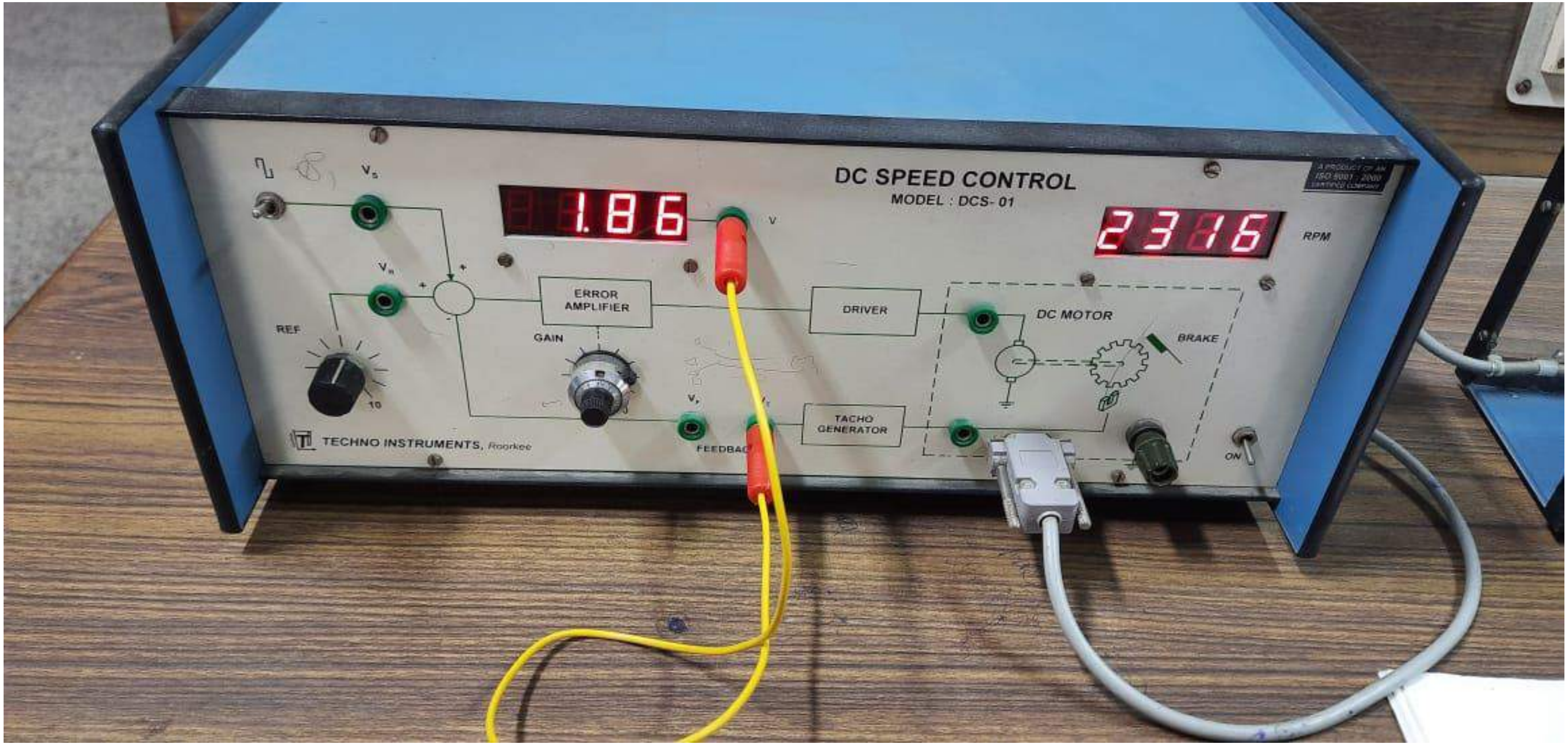
Snapshots during experiment



Snapshots during experiment



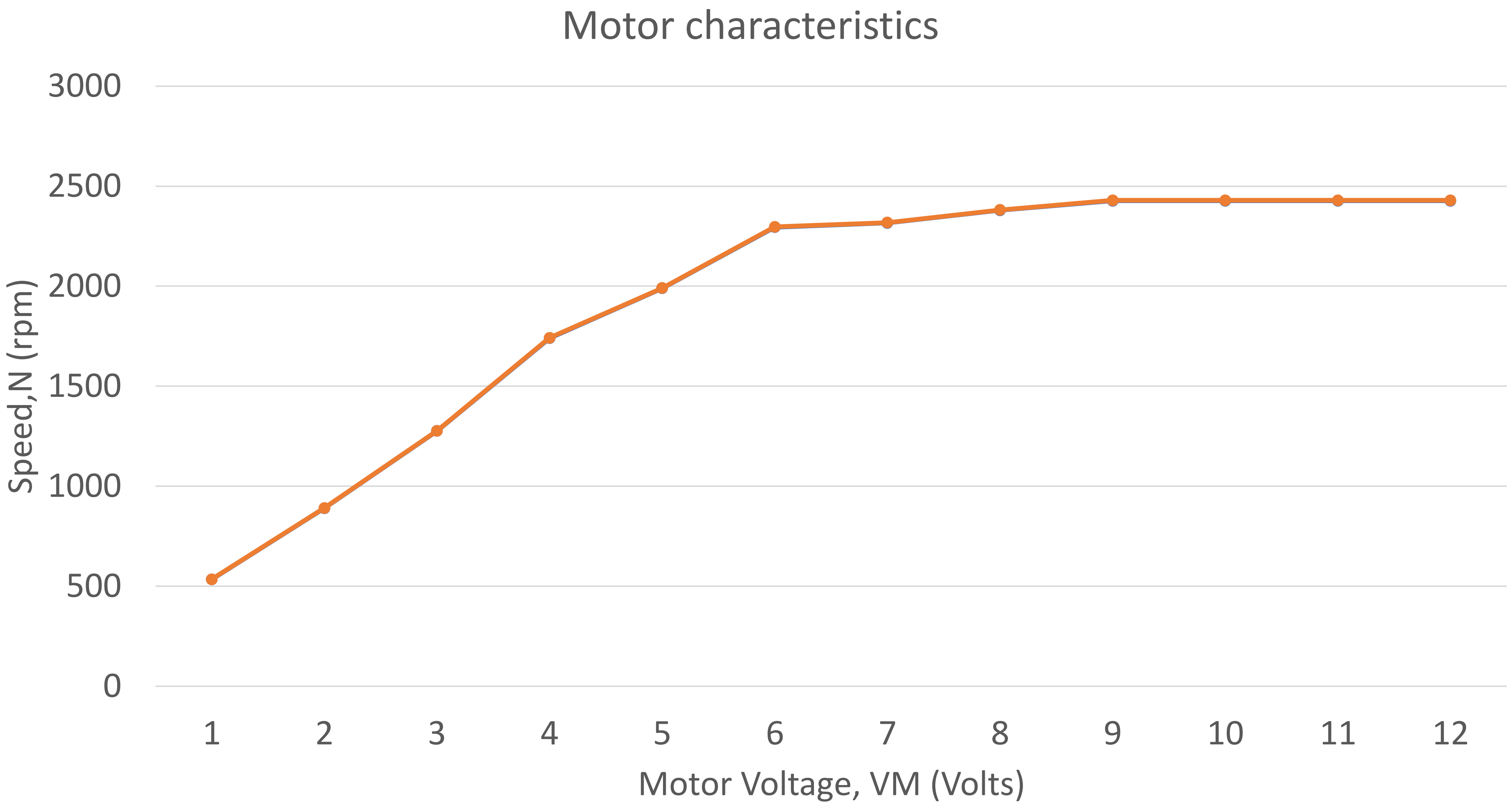
Snapshots during experiment



Observation table

Sr.no	K_A setting	N (rpm)	V_T (Volts)	V_M (Volts)	Experimental $K_A=V_M/V_R$
1	3	534	0.39	3.01	3.01
2	4	890	0.69	4.19	4.19
3	5	1276	1.03	5.45	5.45
4	6	1740	1.44	6.46	6.46
5	7	1989	1.55	7.44	7.44
6	8	2295	1.73	8.55	8.55
7	9	2316	1.86	9.59	9.59
8	10	2379	1.93	10.55	10.55
9	11	2427	1.95	11.45	11.45
10	12	2427	1.95	12.70	12.70
11	13	2427	1.95	13.56	13.56
12	14	2427	1.95	14.60	14.60

Results



Motor characteristics
Speed,N (rpm)
3000
2500
2000
1500
1000
500
0
1
2
3
4
5
6
7
8
9
10
11
12
Motor Voltage, VM (Volts)

Continued....

Experiment- 03

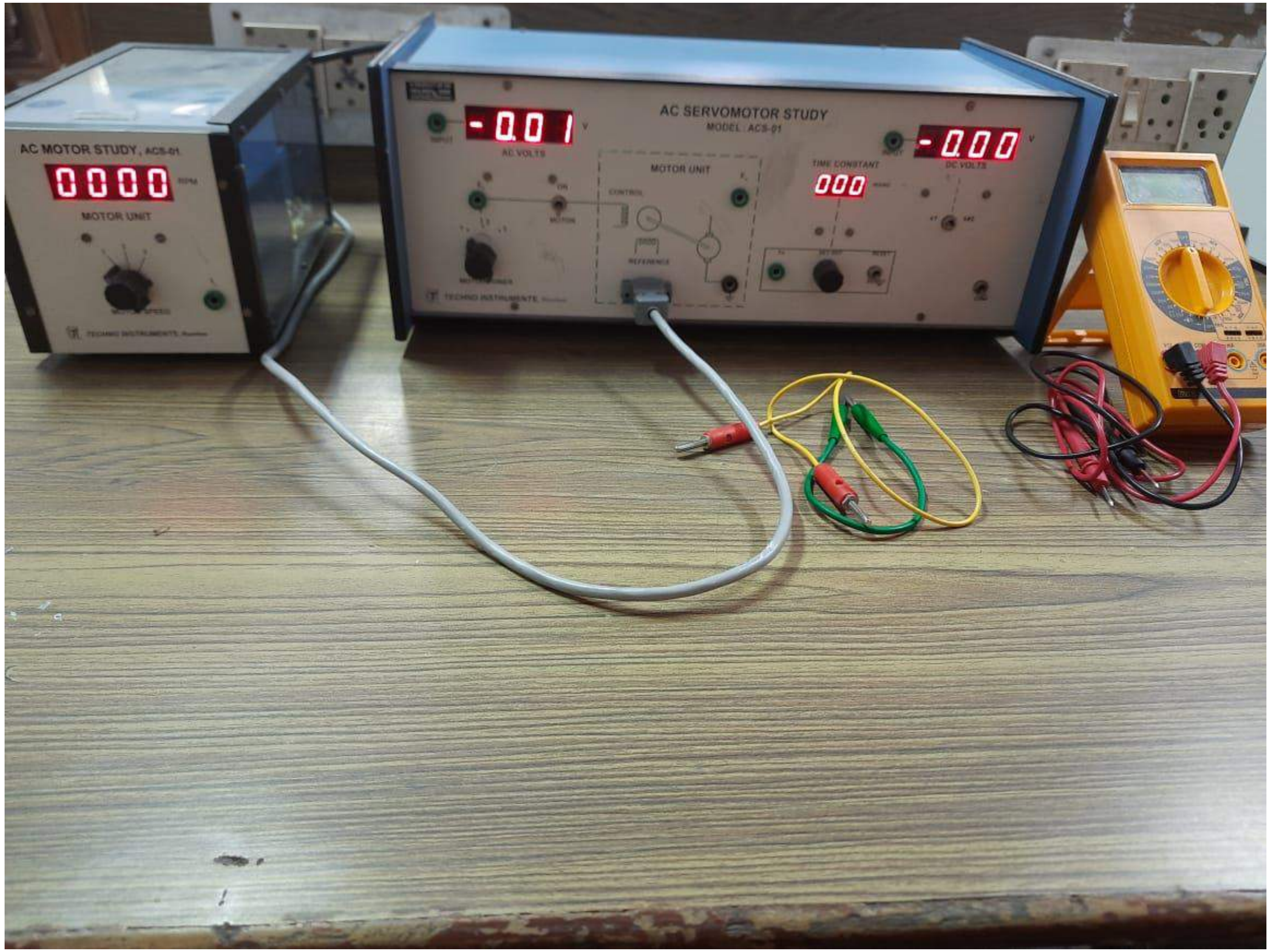
PCEE- 613, Control systems Lab.

GEE-2018

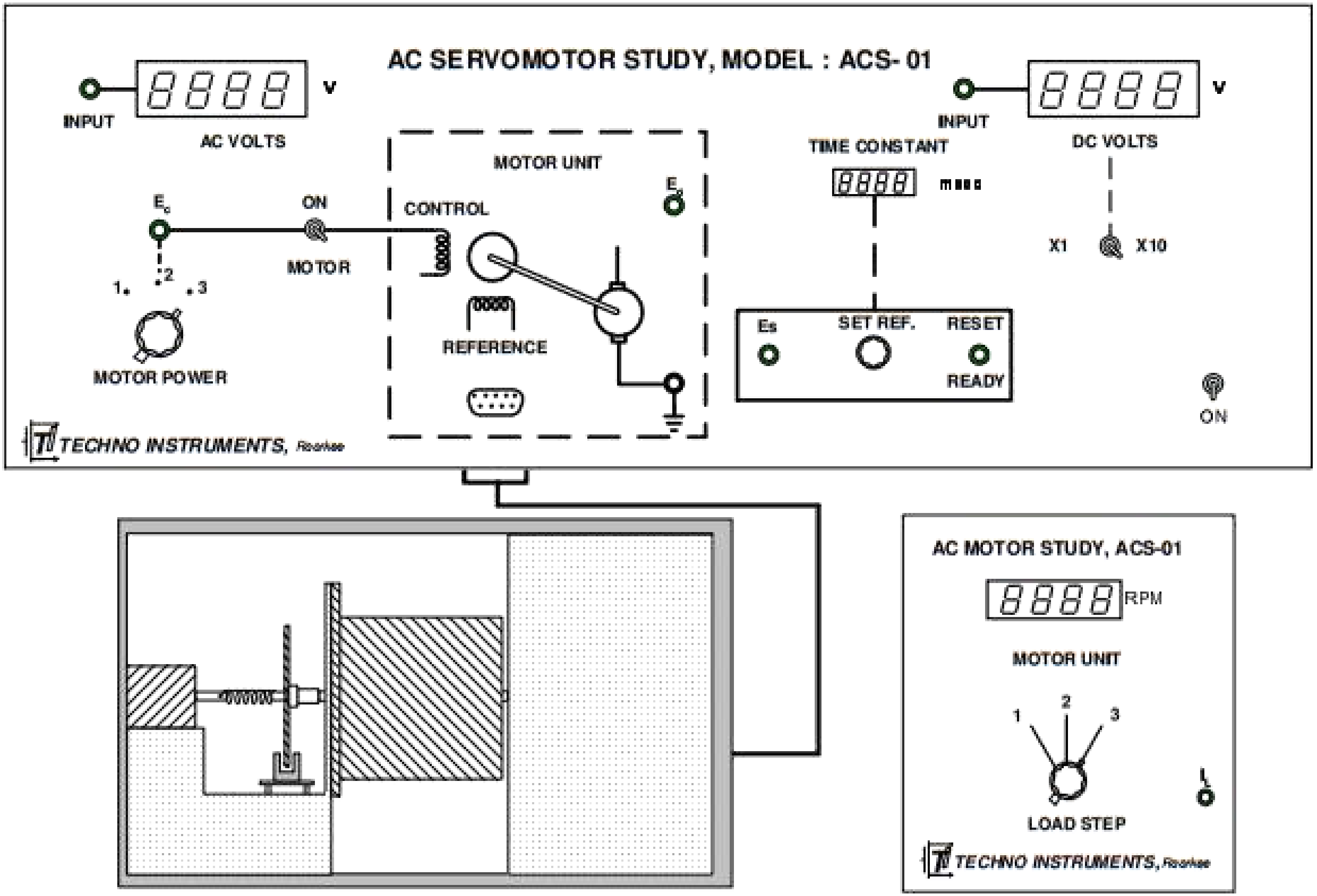
Aim of the experiment: To study the characteristics of a small A.C. servomotor and determine its transfer function.

Apparatus Required:

- AC servomotor study motor unit
- AC servomotor study
- Patch cords
- Multimeter
- Power supply



Panel drawing



Panel drawing AC Servomotor Study, ACS - 01

Theory & Mathematical model

A.C. Servo Motors are basically two-phase, reversible, induction motors modified for servo operation. A schematic diagram of the motor is shown in Fig.1. The two windings, reference and control, may or may not have identical ratings. In the present unit both are rated at 12 volts r.m.s. at 50Hz. A phase shifting capacitor of appropriate value must be connected in series with one of the windings to produce a 90 degree phase shift.

These servo motors are used in applications requiring rapid and accurate response characteristics. A typical torque-speed characteristics of an induction motor is shown in fig.2 for two values of rotor resistance. A servomotor however must have negative slope in its torque-speed characteristics in order to ensure stable operation. To meet the above requirements, these ac servo motors have small diameter, light weight, low inertia and high resistance rotors. The motor's small diameter provides low inertia for fast starts, stops, and reversals. High resistance provides nearly linear torque-speed characteristics. A common structure is a drag-cup rotor. The a.c. servomotors have distinct advantages over d.c. servomotors. The commutator and brush assembly of a d.c. servomotor has limited maintenance free life. These are absent in the a.c.servomotor.

Theory & Mathematical model

From the block diagram of Fig.3 the transfer function of the motor may be written as

$$\frac{\theta(s)}{E_c(s)} = \frac{Km}{s(\tau_m s+1)} \quad \text{for } T_L(s) \equiv 0 \qquad (1)$$

where, $K_m = \dfrac{K_1}{B+K_2}$, and $\tau_m = \dfrac{J}{B+K_2}$,

are the motor gain constant and the motor time constant respectively. As students of control system, our interest is to evaluate the transfer function and the parameters of the ac servomotor.

Again for $E_c(s) \equiv 0$,

$$\frac{\theta(s)}{T_L(s)} = -\frac{\frac{1}{B+K_2}}{s(\tau_m s+1)} = -\frac{K_n}{\tau_m s+1}, \text{ where } K_n = \frac{1}{B+K_2}$$

Combining the above two transfer functions (under assumption of linearity),

$$s\theta(s) = \omega(s) = \frac{K_m}{(\tau_m s+1)}E_c(s) - \frac{K_n}{(\tau_m s+1)}T_L(s)$$

Theory & Mathematical model

The computation of K_m and K_n can be done by using the final value theorem, i.e.,

Steady state speed, $\omega_{ss} = \lim_{s \to 0} s\omega(s) = K_m E_c - K_n T_L$ (2)

where

E_c = Constant voltage applied to the control winding

T_L = Constant Load torque

E_c is measured by the a.c. voltmeter on the panel and T_L is calculated from the loading of the coupled d.c. generator as ,

$$T_L = \frac{\text{Electrical Power drawn from the generator in watts}}{\text{Angular velocity of the shaft in radians / sec}}$$

Block diagram

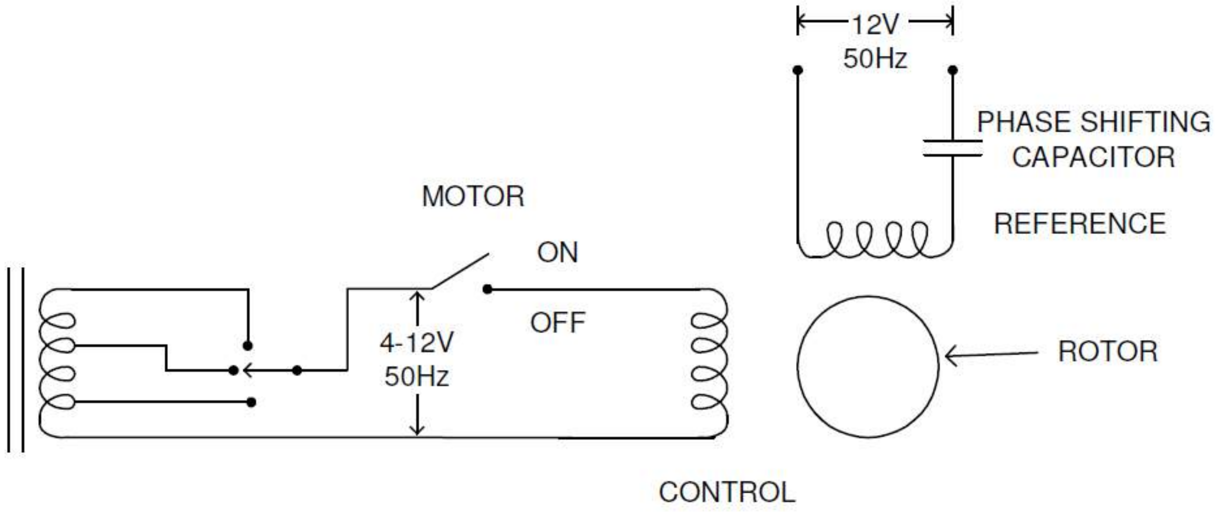


Fig 1 : 2-Phase A.C. Servomotor

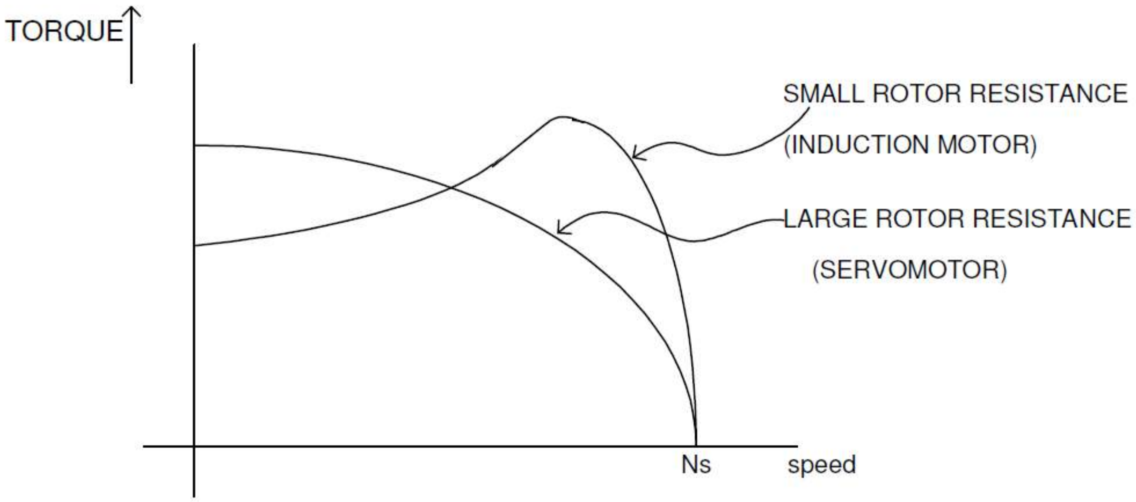


Fig 2 : Torque-speed characteristics of induction motors

Block diagram

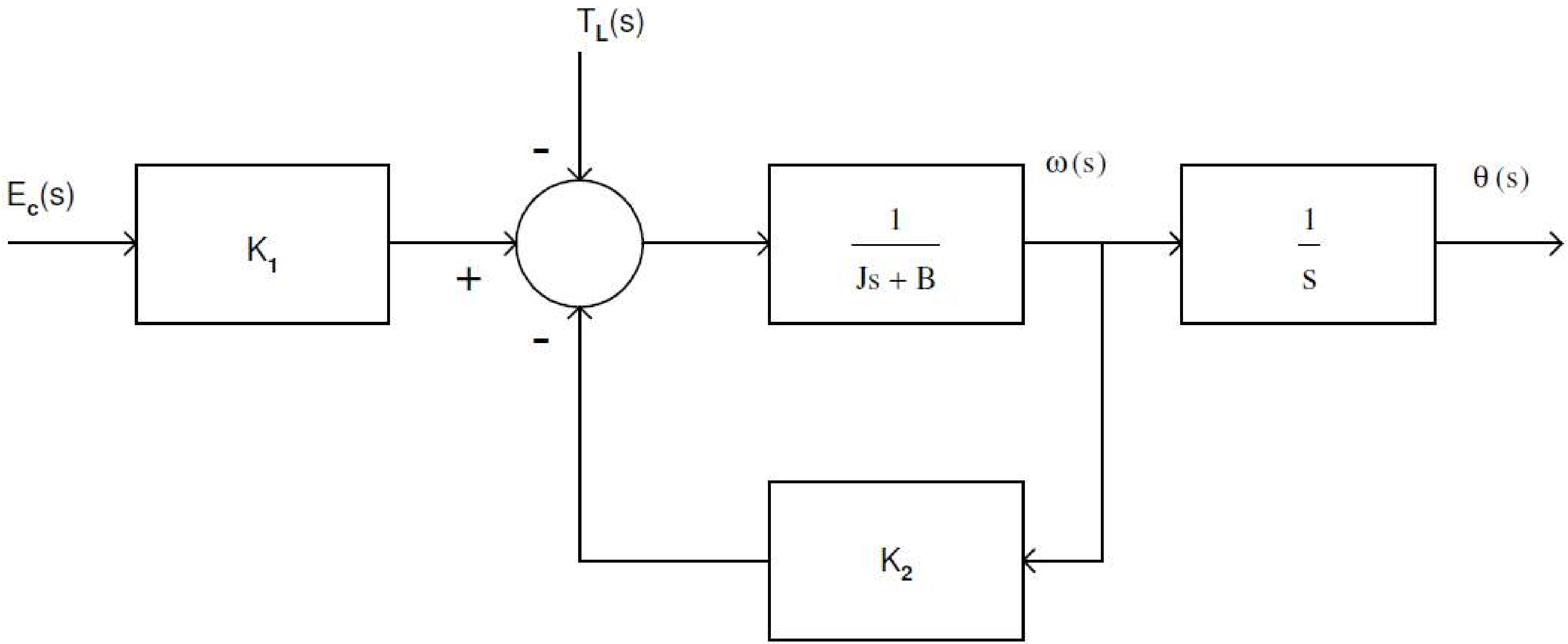


Fig 3 : Block diagram of an a.c.motorsystem

Procedure

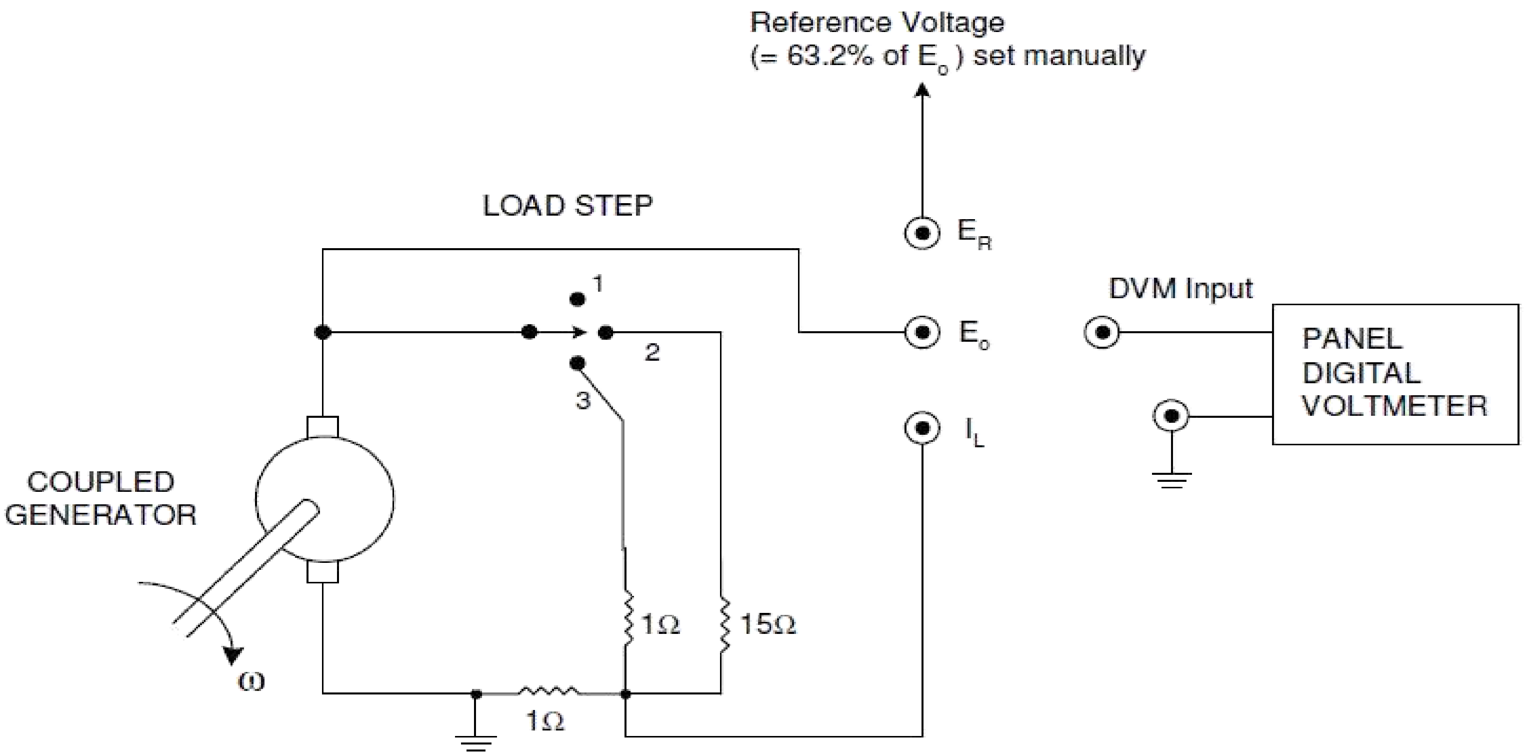
Determination of Generator Constant

The generator constant, K_G, in volts/rpm, may be computed from the no load generator voltage data at various speeds. This would enable one to calculate the generated voltage under loaded condition, which is needed for torque computation in the next section.

Determination of Motor Parameters

The motor is operated at various combination of control phase voltage, E_c, and external loading, T_L, and the data is recorded as in Table-2. E_c is measured with the help of the a.c. voltmeter on the panel in three steps while no load generator voltage, E_0 and load currents, I_L are measured by a switchable d.c. panel meter provided. The loading circuit is shown in Fig. 4

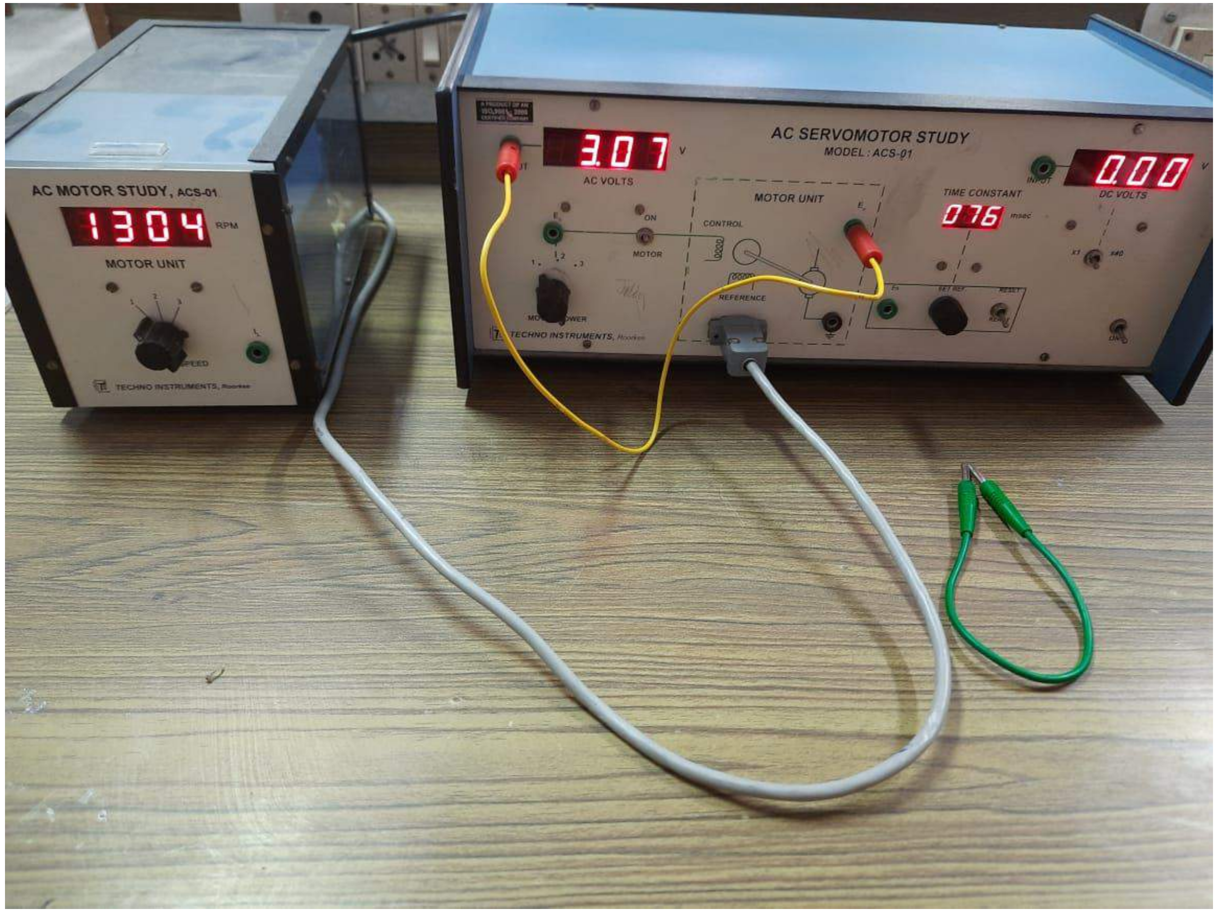
Procedure

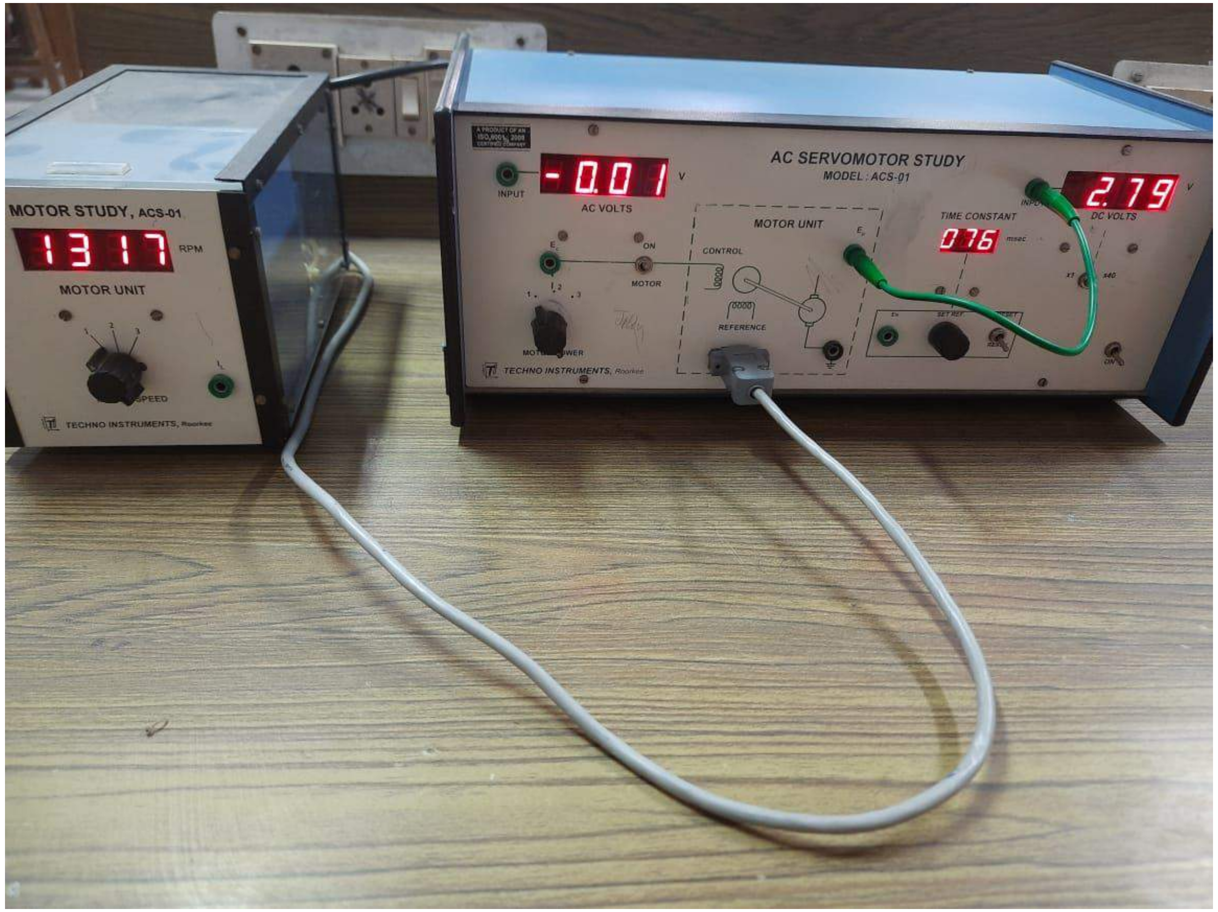


DVM input to be externally connected to terminal
E_o To measure generator output voltage
E_R To measure 63.2% of E_o for Time-constant measurement
I_L To measure load current as the drop across 1Ω resistance

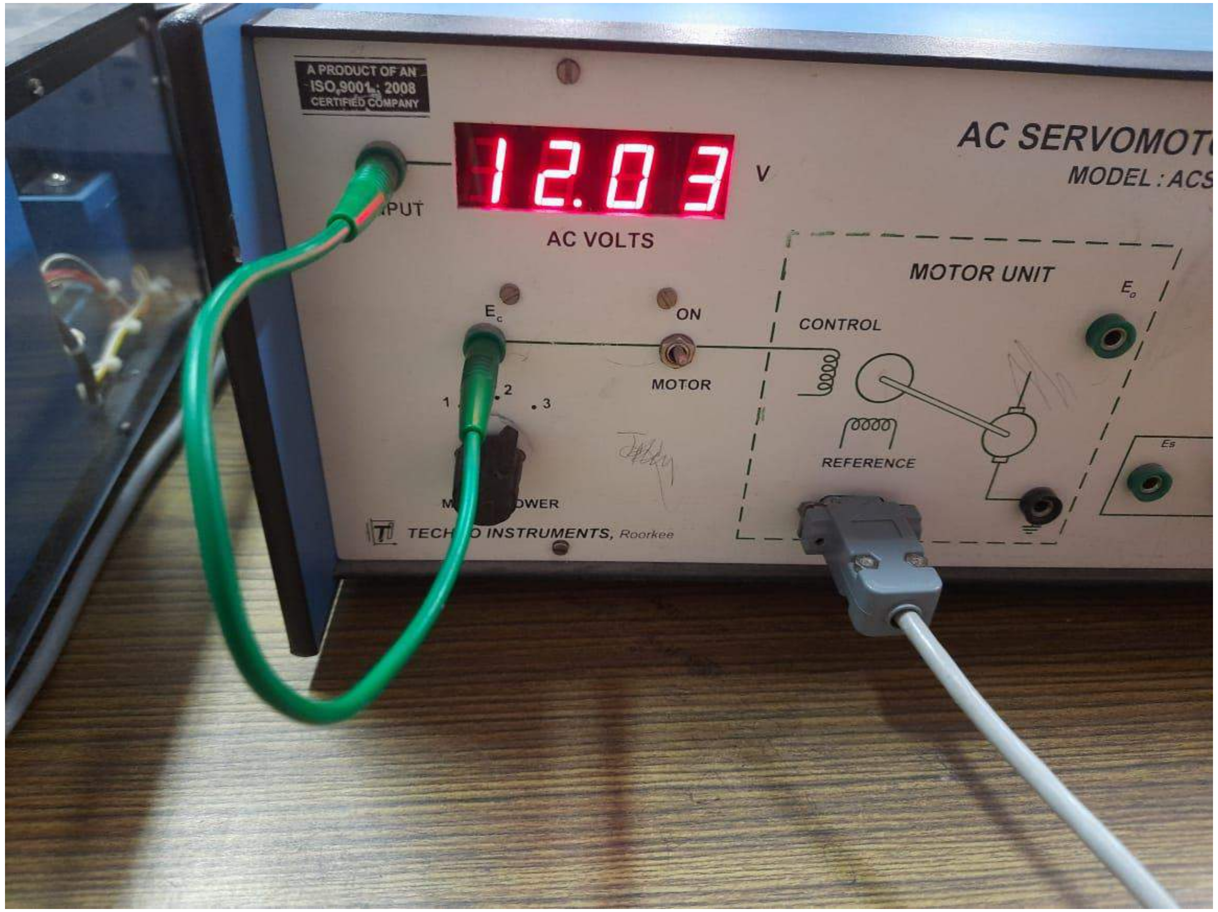
Fig : 4 Loading circuit arrangement

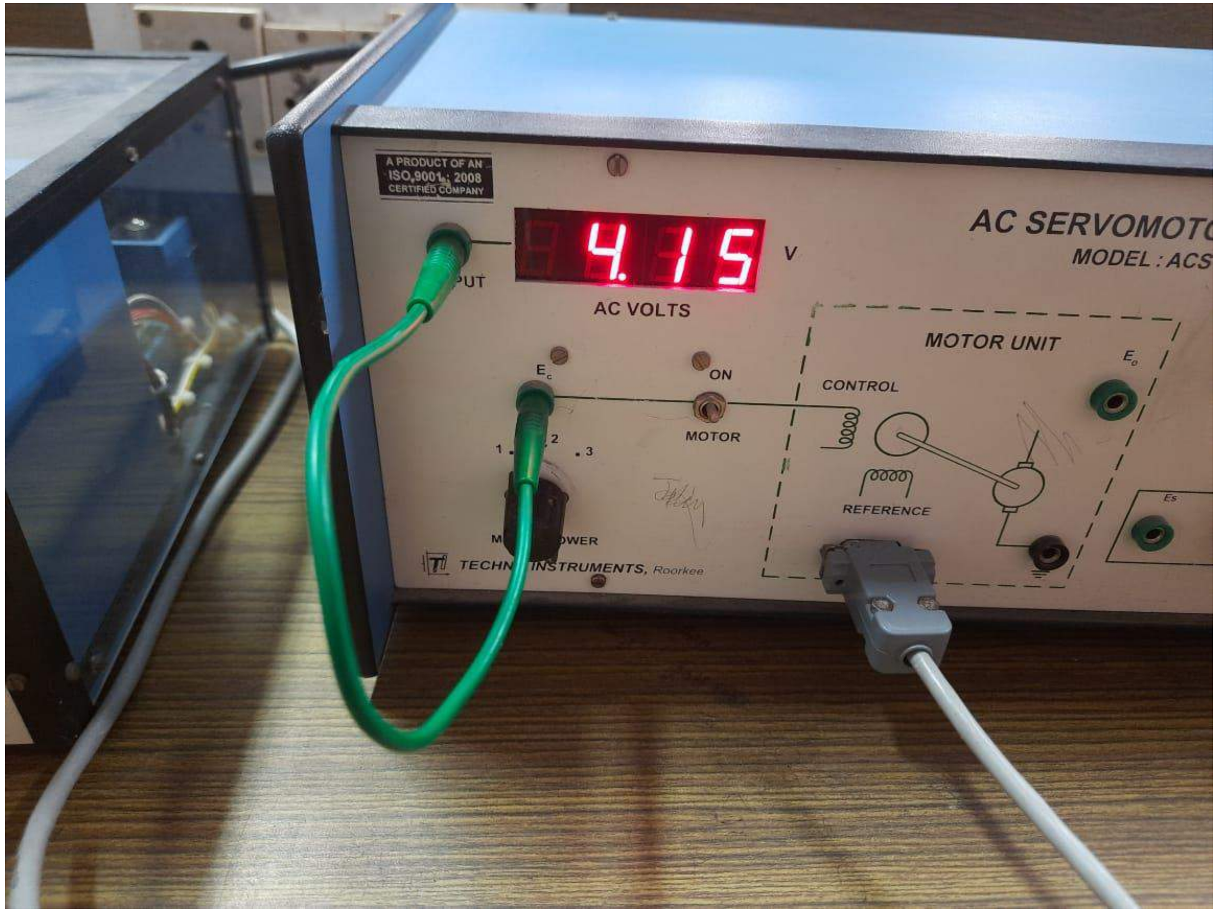
Snapshots during experiment



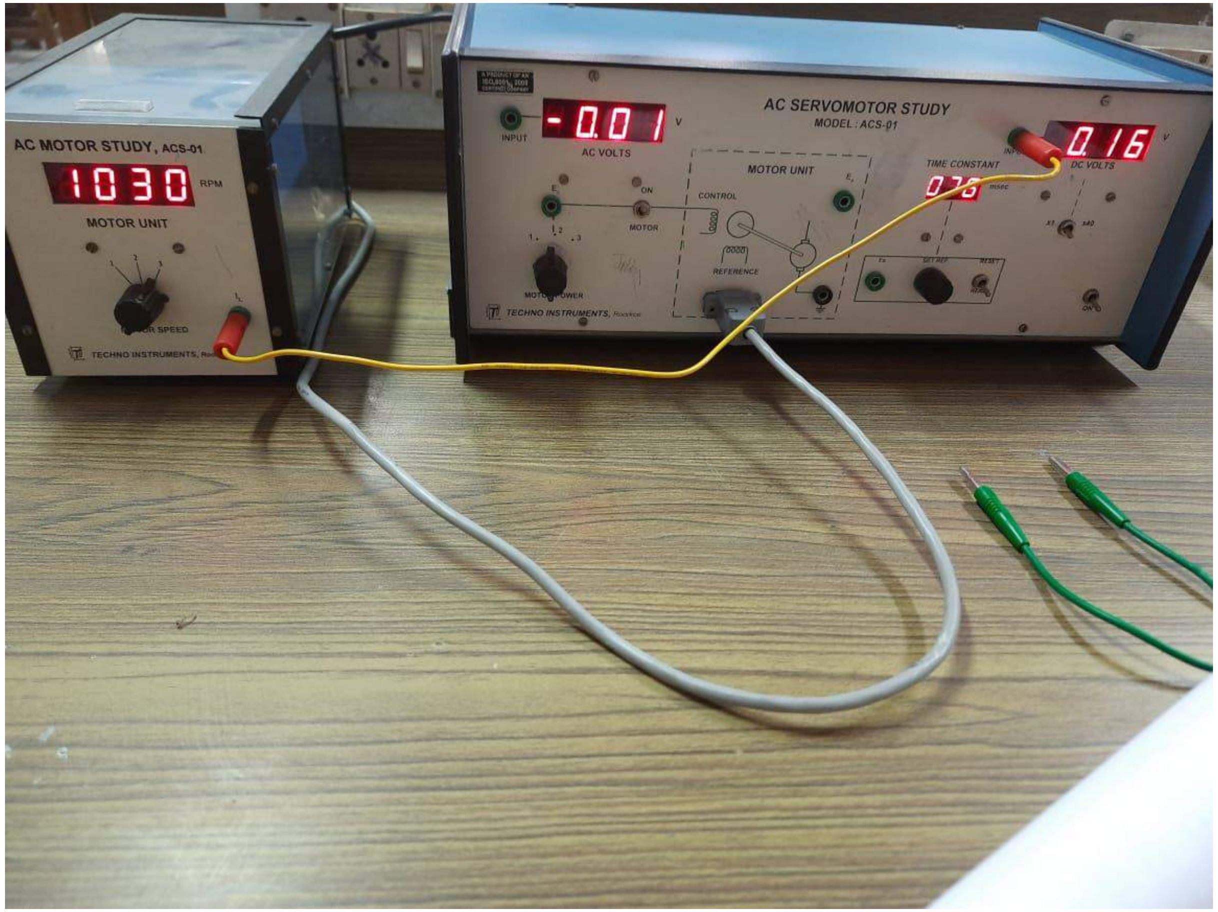


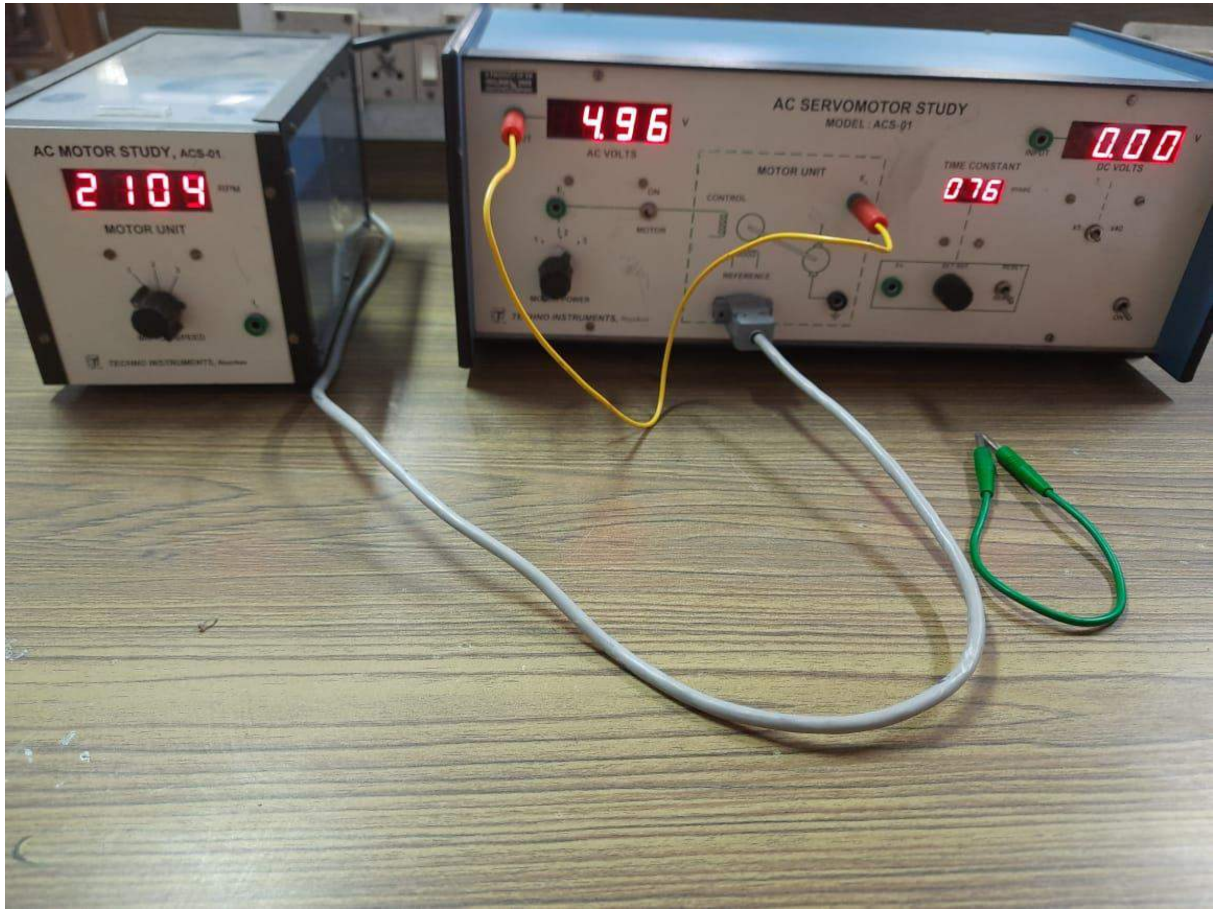
Snapshots during experiment





Snapshots during experiment





Observation Table-01

Input step	Motor speed, N_r, rpm	Load step-1 (no load), voltage, Volts	Generator constant, K_G, Volts/rpm
1	1304	3.07	0.00235
2	1930	4.56	0.00236
3	2104	4.96	0.00236
Average $K_G = 0.00236$ Volts/rpm			

Observation Table-02

Input		Load Step-1 (No Load)				Load Step-2			Load Step-3		
Step	E_c, rms V	E_0, volts	I_L, amp	N, rpm	$T_L = \frac{(30K_G I_L)}{\pi}$	I_L, amp	N, rpm	$T_L = \frac{(30K_G I_L)}{\pi}$	I_L, amp	N, rpm	$T_L = \frac{(30K_G I_L)}{\pi}$
1	4.15	2.79	0	1317	0	0.61	770	0.0137	0.08	520	0.0018
2	8.08	4.10	0	1937	0	1.07	1386	0.0241	0.16	1030	0.0036
3	12.03	4.48	0	2109	0	1.29	1674	0.0291	0.20	1349	0.0045

Continued....

Experiment -04

PCEE-613,Control Systems (Lab)

GEE-2018

Aim of the Experiment: To Obtain the pole, zeros and gain constant of a transfer functions using MATLAB.

Example 01:

$$H(s) = \frac{2s^2 + 3s}{s^2 + \frac{1}{\sqrt{2}}s + \frac{1}{4}}$$

MATLAB code and Output

Code

Editor - C:\Users\Venus\Desktop\control lab\expt 10 ok\example01.m

example01.m | example02.m | example03.m

```
num = [2 3];
den = [1 1/sqrt(2) 1/4];
[num,den] = eqtflength(num,den);
[z,p,k] = tf2zp(num,den)
```

Output

Command Window

```
>> example01

z =

         0
   -1.5000


p =

  -0.3536 + 0.3536i
  -0.3536 - 0.3536i


k =

     2

>>
```

m.File link

example01.m

Example 02:

$$G_1(s) = \frac{10}{s^2 + 2s + 10}$$

MATLAB code and Output

Code

Editor - C:\Users\Venus\Desktop\control lab\expt 10 ok\example02.m

example01.m | example02.m | example03.m

```matlab
num = [10];
den = [1 2 10];
[num,den] = eqtflength(num,den);
[z,p,k] = tf2zp(num,den)
```

Output

Command Window

```
>> example02

z =

     0
     0


p =

  -1.0000 + 3.0000i
  -1.0000 - 3.0000i


k =

    10

fx >>
```

m.File link

example02.m

Example 03:

$$G_2(s) = \frac{5}{s + 5}$$

MATLAB code and Output

Code

Editor - C:\Users\Venus\Desktop\control lab\expt 10 ok\example03.m

example01.m | example02.m | example03.m

```
num = [5];
den = [1 5];
[num,den] = eqtflength(num,den);
[z,p,k] = tf2zp(num,den)
```

Output

Command Window

```
>> example03

z =

     0


p =

    -5


k =

     5

>>
```

m.File link

example03.m

Continued....

Experiment -05
PCEE-613,Control Systems (Lab)
GEE-2018

Aim of the Experiment: To Obtain the Cascaded, Parallel, and Feedback (Closed-Loop) Transfer Functions with MATLAB.

Theory:

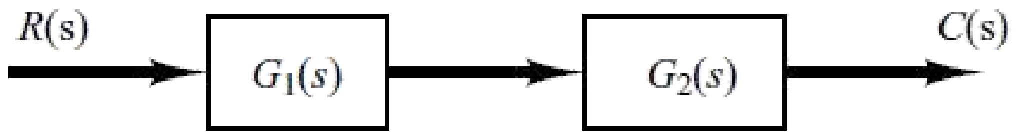
In control-systems analysis, we frequently need to calculate the cascaded transfer functions, parallel-connected transfer functions, and feedback-connected (closed-loop) transfer functions. MATLAB has convenient commands to obtain the cascaded, parallel, and feedback (closed-loop) transfer functions.

Suppose that there are two components $G_1(s)$ and $G_2(s)$ connected differently as shown in Figure

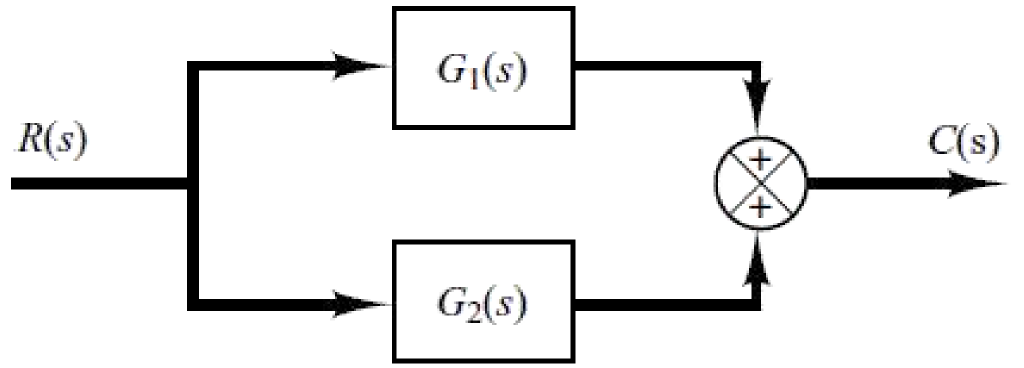
$$G_1(s) = \frac{\text{num1}}{\text{den1}}, \qquad G_2(s) = \frac{\text{num2}}{\text{den2}}$$

Contd…

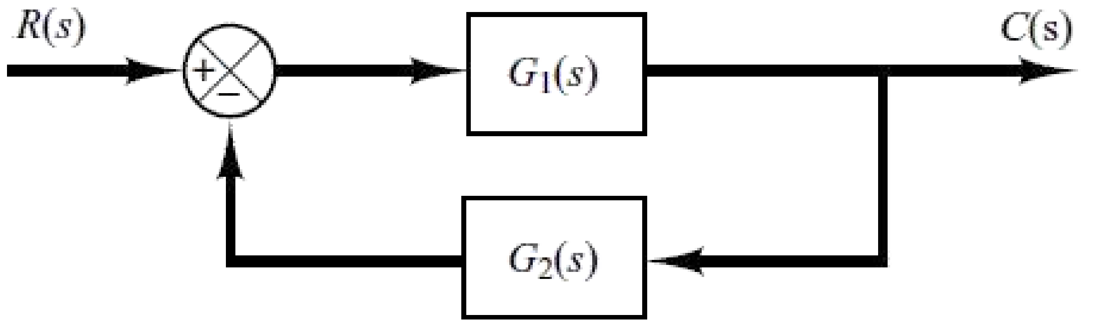
(a)



(b)



(c)



Example

Example 01: Consider the following transfer function and convert it into the state space representation using MATLAB code.

$$G_1(s) = \frac{10}{s^2 + 2s + 10} = \frac{\text{num1}}{\text{den1}}, \qquad G_2(s) = \frac{5}{s + 5} = \frac{\text{num2}}{\text{den2}}$$

MATLAB code and Output

Code

Editor - C:\Users\Venus\Desktop\control lab\expt 04\Example01.m

Example01.m

```
num1 = [10];
den1 = [1 2 10];
num2 = [5];
den2 = [1 5];
[num, den] = series(num1,den1,num2,den2);
printsys(num,den)
[num, den] = parallel(num1,den1,num2,den2);
printsys(num,den)
[num, den] = feedback(num1,den1,num2,den2);
printsys(num,den)
```

Output

Command Window

```
>> Example01

num/den =

                 50
   ------------------------
   s^3 + 7 s^2 + 20 s + 50

num/den =

       5 s^2 + 20 s + 100
   ------------------------
   s^3 + 7 s^2 + 20 s + 50

num/den =

            10 s + 50
   -------------------------
   s^3 + 7 s^2 + 20 s + 100
>>
```

m.File link

Example01.m

Continued....

Experiment -06

PCEE-613,Control Systems (Lab)

GEE-2018

Aim of the Experiment: To find the rise time, peak time, % maximum overshoot, and settling time of the second-order system.

Example 01:

$$\frac{C(s)}{R(s)} = \frac{25}{s^2 + 6s + 25}$$

MATLAB code and Output

Code

Editor - C:\Users\Venus\Desktop\control lab\expt 09\Example01.m

Example01.m

```matlab
% ------- In this example, we assume zeta = 0.6 and wn = 5 -------
num = [25];
den = [1 6 25];
t = 0:0.005:5;
[y,x,t] = step(num,den,t);
r = 1; while y(r) < 1.0001; r = r + 1; end;
rise_time = (r - 1)*0.005
[ymax,tp] = max(y);
peak_time = (tp - 1)*0.005
max_overshoot = ymax-1
s = 1001; while y(s) > 0.98 & y(s) < 1.02; s = s - 1; end;
settling_time = (s - 1)*0.005
```

Output

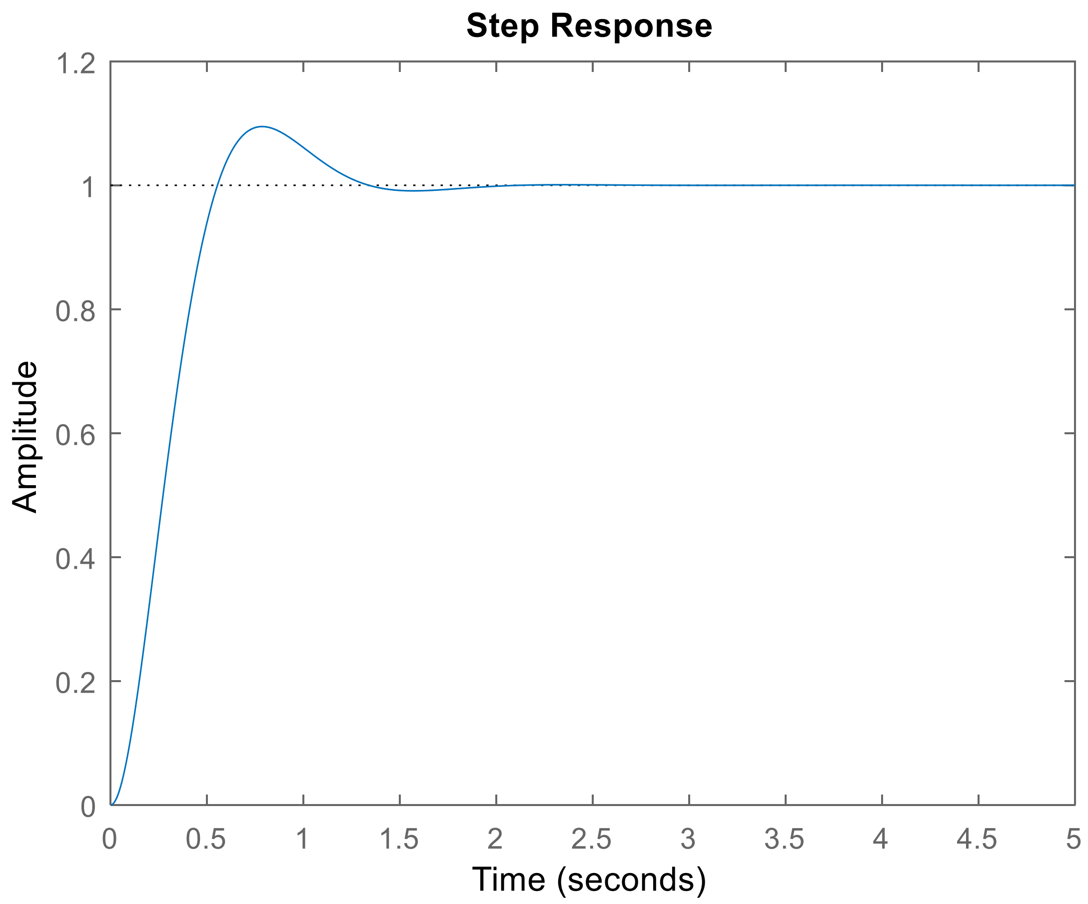
Command Window

```
    Academic License

>> Example01

rise_time =

    0.5550


peak_time =

    0.7850


max_overshoot =

    0.0948


settling_time =

    1.1850

fx >>
```

m.File link

Example01.m

MATLAB code and Output



Aim of the Experiment: To find the rise time, peak time, % maximum overshoot, and settling time of the second-order system.

Example 02:

$$\frac{C(s)}{R(s)} = G(s) = \frac{1}{s^2 + 0.2s + 1}$$

MATLAB code and Output

Code

Editor - C:\Users\Venus\Desktop\control lab\expt 09\Example02.m

Example01.m | Example02.m | Untitled

```
% ------- In this example, we assume zeta = 0.6 and wn = 5 -------
num = [1];
den = [1 0.2 1];
t = 0:0.005:5;
[y,x,t] = step(num,den,t);
r = 1; while y(r) < 1.0001; r = r + 1; end;
rise_time = (r - 1)*0.005
[ymax,tp] = max(y);
peak_time = (tp - 1)*0.005
max_overshoot = ymax-1
s = 1001; while y(s) > 0.98 & y(s) < 1.02; s = s - 1; end;
settling_time = (s - 1)*0.005
```

Output

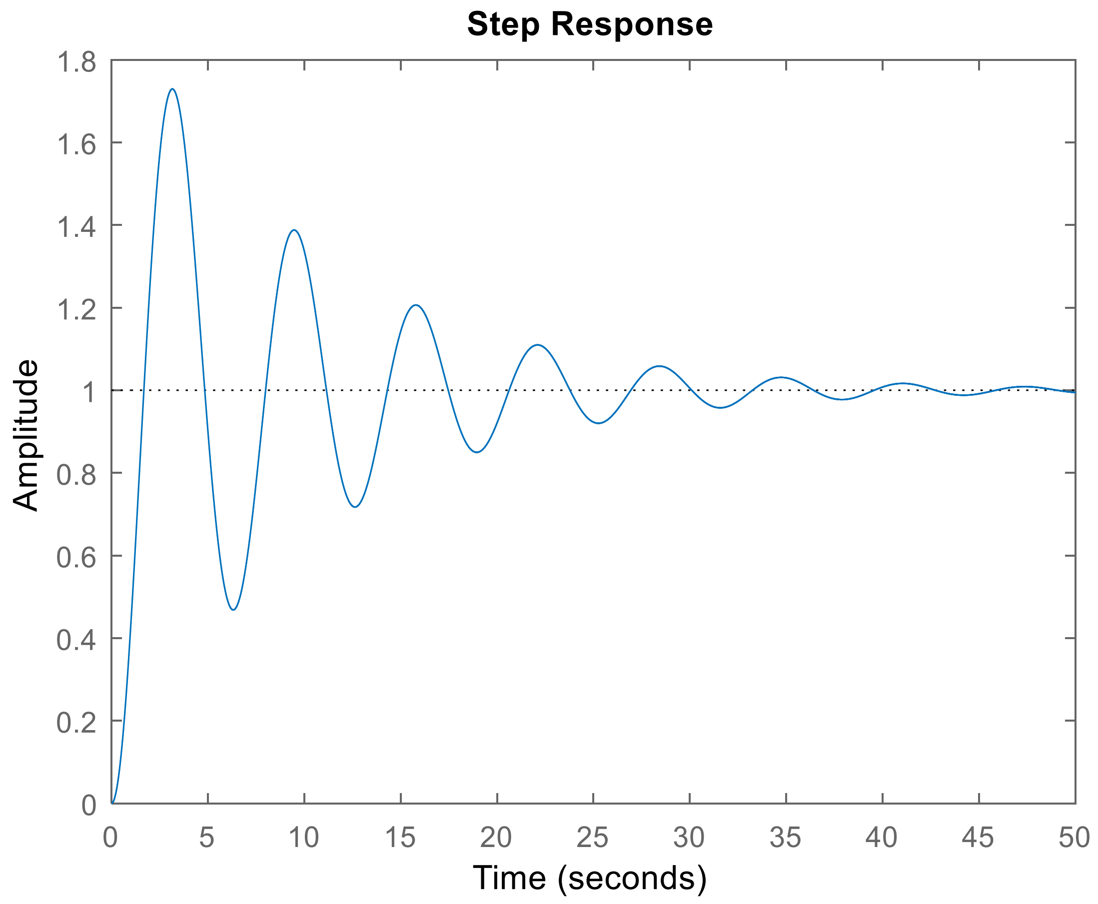
Command Window

```
>> Example02

rise_time =

    1.6800


peak_time =

    3.1550


max_overshoot =

    0.7292


settling_time =

     5

fx >>
```

m.File link

Example02.m

MATLAB code and Output



Aim of the Experiment: To find the rise time, peak time, % maximum overshoot, and settling time of the second-order system.

Example 03:

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

MATLAB code and Output

Code

Editor - C:\Users\Venus\Desktop\control lab\expt 09\example03.m

example03.m

```matlab
% ------- In this example, we assume zeta = 0.6 and wn = 5 -------
num = [2 1];
den = [1 1 1];
t = 0:0.005:5;
[y,x,t] = step(num,den,t);
r = 1; while y(r) < 1.0001; r = r + 1; end;
rise_time = (r - 1)*0.005
[ymax,tp] = max(y);
peak_time = (tp - 1)*0.005
max_overshoot = ymax-1
s = 1001; while y(s) > 0.98 & y(s) < 1.02; s = s - 1; end;
settling_time = (s - 1)*0.005
```

Output

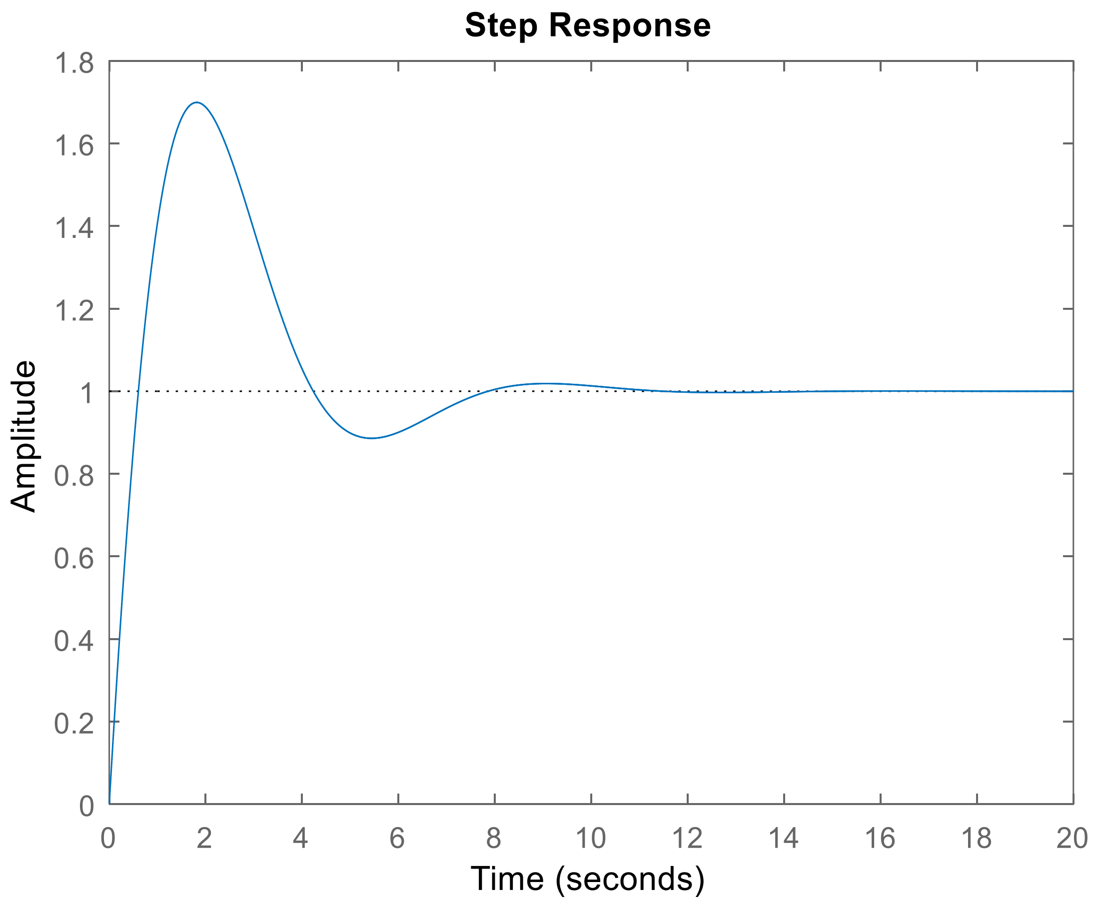
Command Window

```
>> example03

rise_time =

    0.6050


peak_time =

    1.8150


max_overshoot =

    0.6994


settling_time =

     5

fx >>
```

m.File link

example03.m

MATLAB code and Output



Continued….

Experiment -07

PCEE-613,Control Systems (Lab)

GEE-2018

Aim of the Experiment: To plot root locus diagram of a closed loop transfer function.

Procedure:

1. Determine the root loci on the real axis
2. Determine the asymptotes of the root loci
3. Determine the breakaway point
4. Determine the points where the root loci cross the imaginary axis
5. Choose a test point in the broad neighborhood of the jv axis and the origin
6. Draw the root loci
7. Determine a pair of dominant complex-conjugate closed-loop poles such that the damping ratio z is 0.5

Example

Consider the following transfer functions to plot the root locus diagram using MATLAB code.

Example 01:

$$\frac{Y(s)}{U(s)} = \frac{s}{(s + 10)(s^2 + 4s + 16)}$$

$$= \frac{s}{s^3 + 14s^2 + 56s + 160}$$

MATLAB code and Output

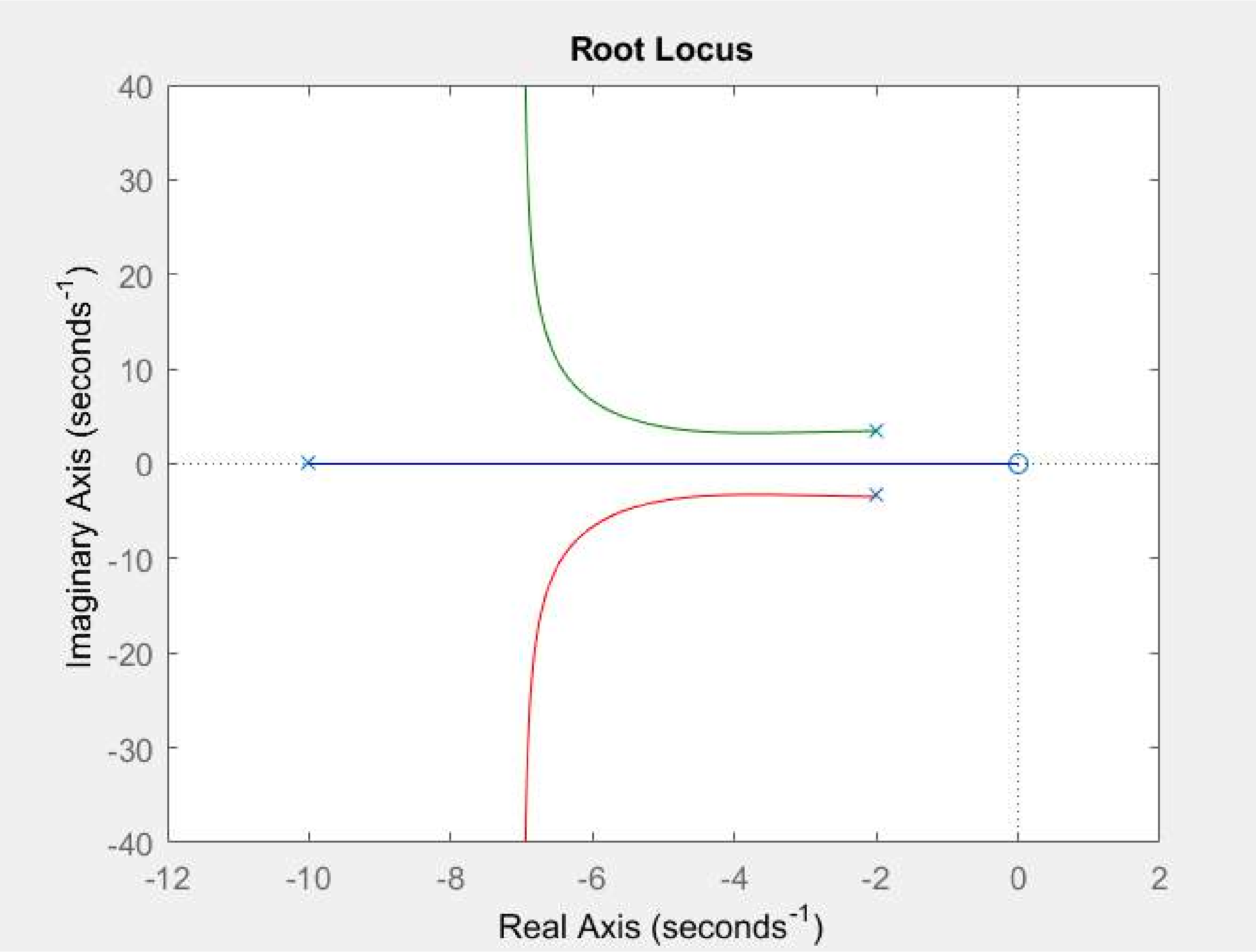
Code

Editor - C:\Users\Venus\Desktop\tfss.m

tfss.m

```
num = [1 0];
den = [1 14 56 160];
rlocus(num,den)
```

Output



m.File link

example01.m

Example

Consider the following transfer functions to plot the root locus diagram using MATLAB code.

Example 02:

$$\frac{K(s+3)}{s(s+1)(s^2+4s+16)}$$

MATLAB code and Output

Code

Editor - C:\Users\Venus\Desktop\control lab\expt 06\example02.m

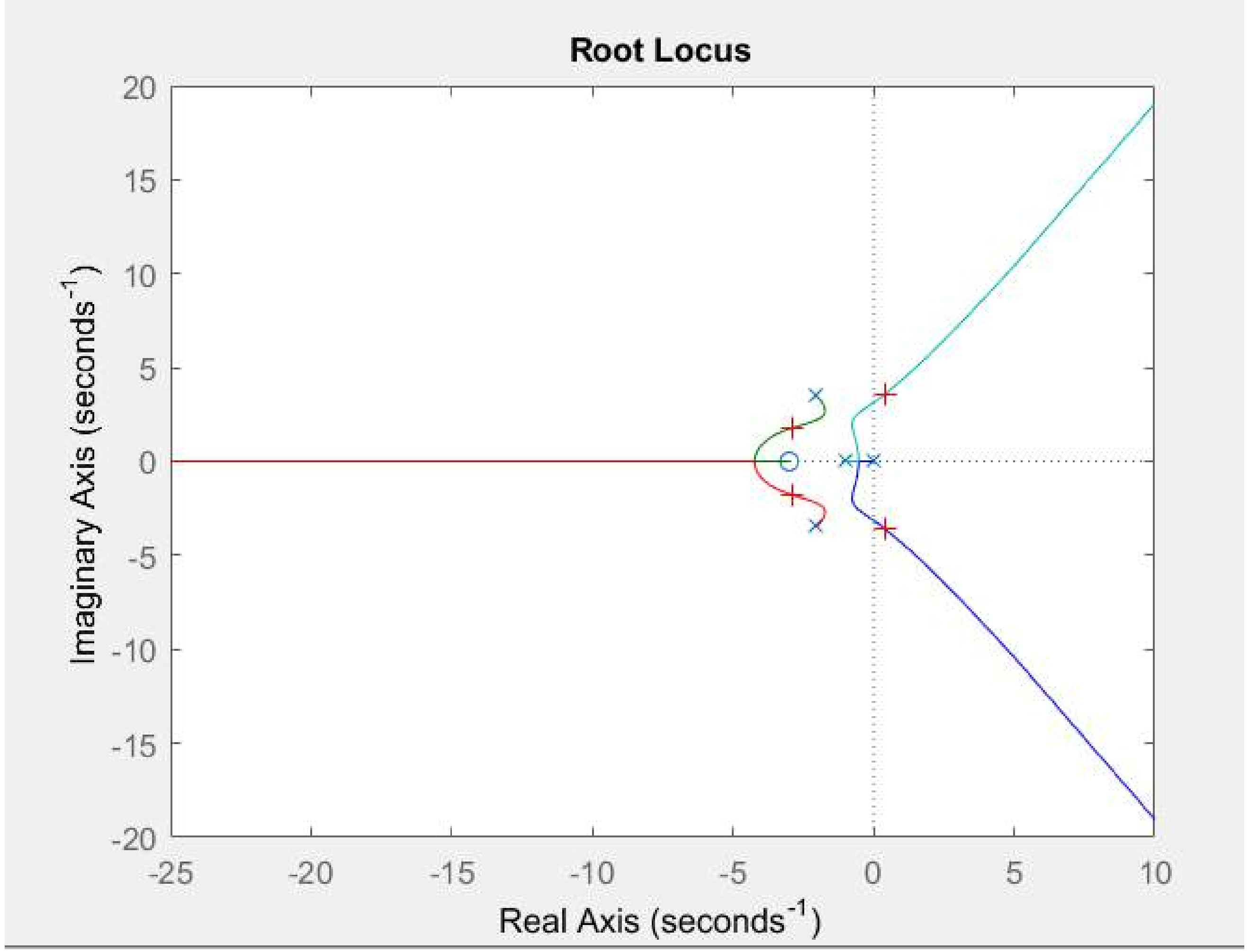
example01.m | example02.m | example03.m

```
num = [1 3];
den = [1 5 20 16 0];
rlocus(num,den);
[k,p]=rlocfind(num,den)
```

m.File link

example02.m

Output



Example

Consider the following transfer functions to plot the root locus diagram using MATLAB code.

Example 03:

$$G(s)H(s) = \frac{K}{s(s+0.5)(s^2+0.6s+10)}$$

$$= \frac{K}{s^4 + 1.1s^3 + 10.3s^2 + 5s}$$

MATLAB code and Output

Code

Editor - C:\Users\Venus\Desktop\control lab\expt 06\example03.m

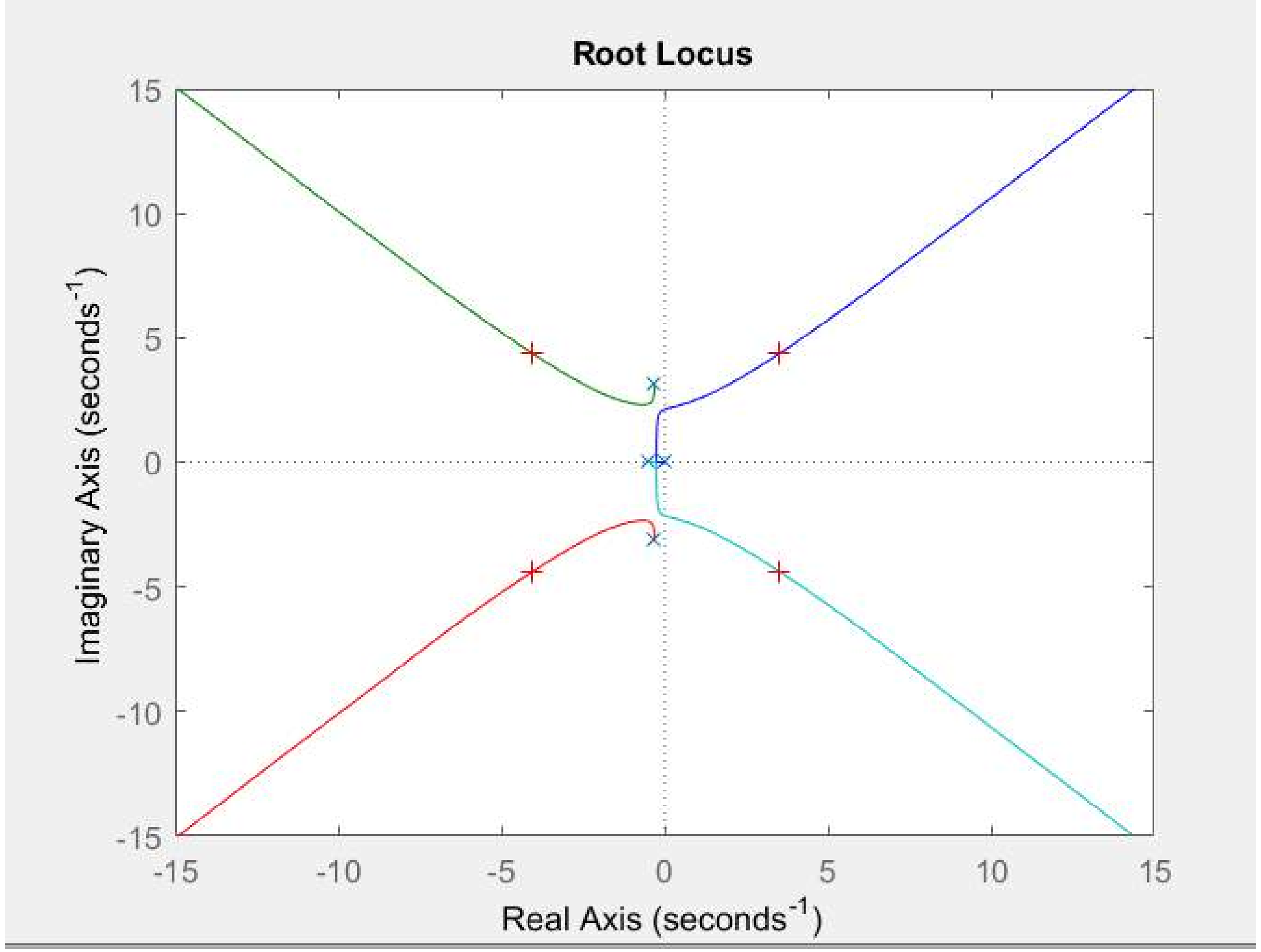
example01.m | example02.m | example03.m

```
num = [1];
den = [1 1.1 10.3 5 0];
rlocus(num,den);
[k,p]=rlocfind(num,den)
```

m.File link

example03.m

Output



Continued….

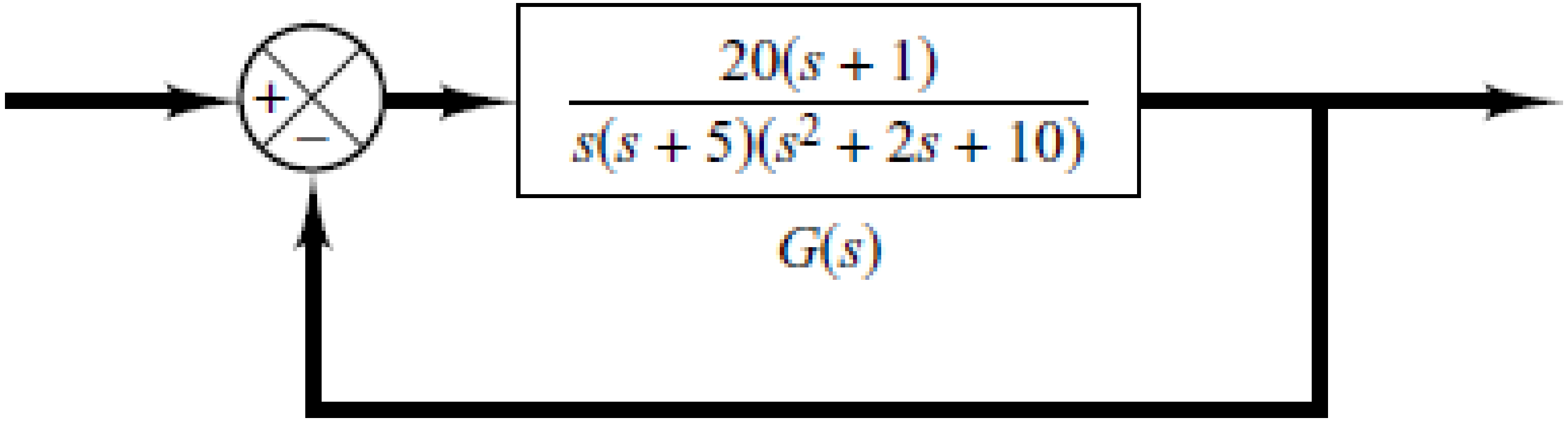
Experiment -08

PCEE-613,Control Systems (Lab)

GEE-2018

Aim of the Experiment: Draw a Bode diagram of the open-loop transfer function G(s) of the closed-loop system shown in Figure. Determine the gain margin, phase margin, phase-crossover frequency, and gain crossover frequency with MATLAB

Example 01: Consider the following transfer function to plot the bode diagram using MATLAB.



MATLAB code and Output

Code

Editor - C:\Users\Venus\Desktop\tfss.m

tfss.m

```matlab
num = [20 20];
den = conv([1 5 0], [1 2 10]);
sys=tf(num,den);
w=logspace(-1,2,100);
bode(sys,w)
[Gm,pm,wcp,wcg]=margin(sys);
Gmdb=20*log10(Gm);
[Gmdb pm wcp wcg]
```

Output

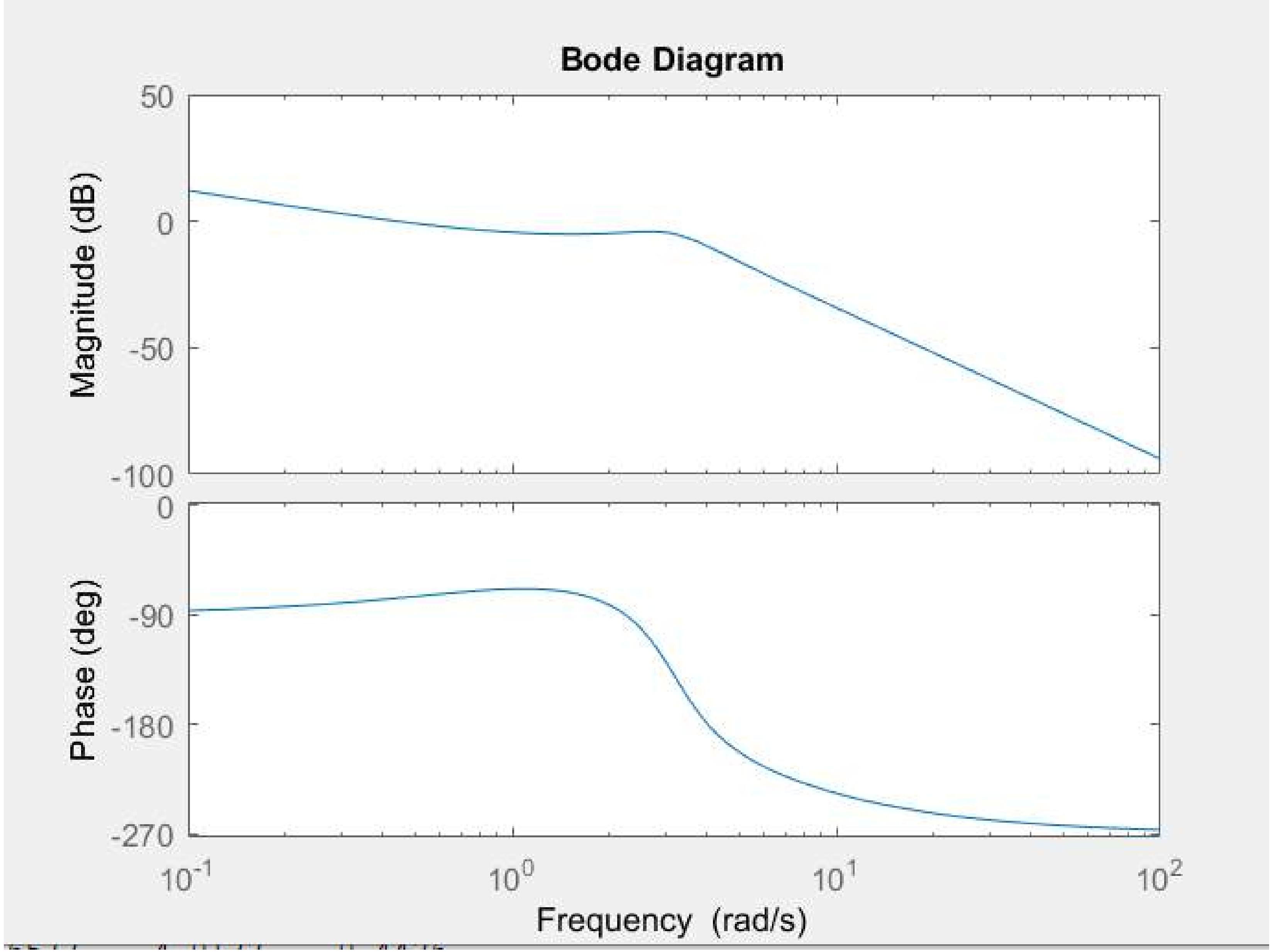
Command Window

```
>> tfss

ans =

    9.9301  103.6573    4.0132    0.4426

>>
```

m.File link

example01.m

Bode plot



Bode Diagram
Magnitude (dB)
50
0
-50
-100
Phase (deg)
0
-90
-180
-270
10^{-1}
10^{0}
10^{1}
10^{2}
Frequency (rad/s)

Example

Example02: Consider the following transfer function to plot the bode diagram using MATLAB.

$$G(s) = \frac{25}{s^2 + 4s + 25}$$

MATLAB code and Output

Code

Editor - C:\Users\Venus\Desktop\control lab\expt 07\example02.m

example01.m | example02.m | example03.m

```
num=[0 0 25];
den=[1 4 25];
bode(num,den);
[Gmdb pm wcp wcg]
```

Output

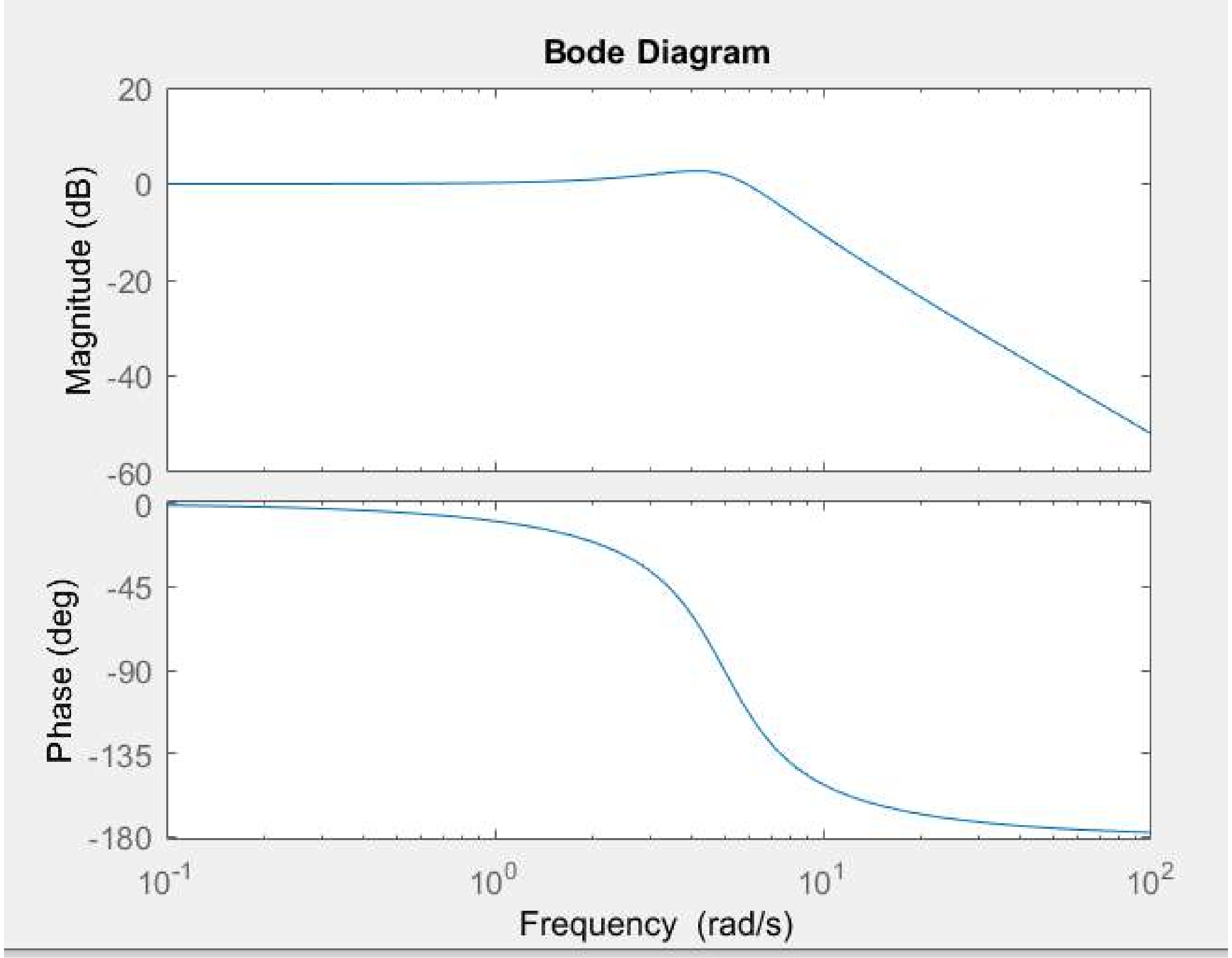
Command Window

```
>> example02

ans =

   16.6503       Inf    3.8730       NaN

>>
```

m.File link

example02.m

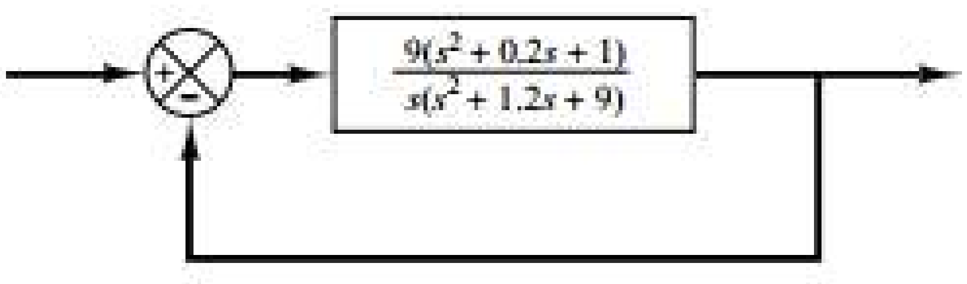
Bode plot



Bode Diagram
Magnitude (dB)
20
0
-20
-40
-60
Phase (deg)
0
-45
-90
-135
-180
10^{-1}
10^{0}
10^{1}
10^{2}
Frequency (rad/s)

Example

Example03: Consider the following transfer function to plot the bode diagram using MATLAB.



MATLAB code and Output

Code

Editor - C:\Users\Venus\Desktop\control lab\expt 07\example03.m

example01.m | example02.m | example03.m

```
num=[9 1.8 9];
den=[1 1.2 9 0];
bode(num,den);
[Gmdb pm wcp wcg]
```

Output

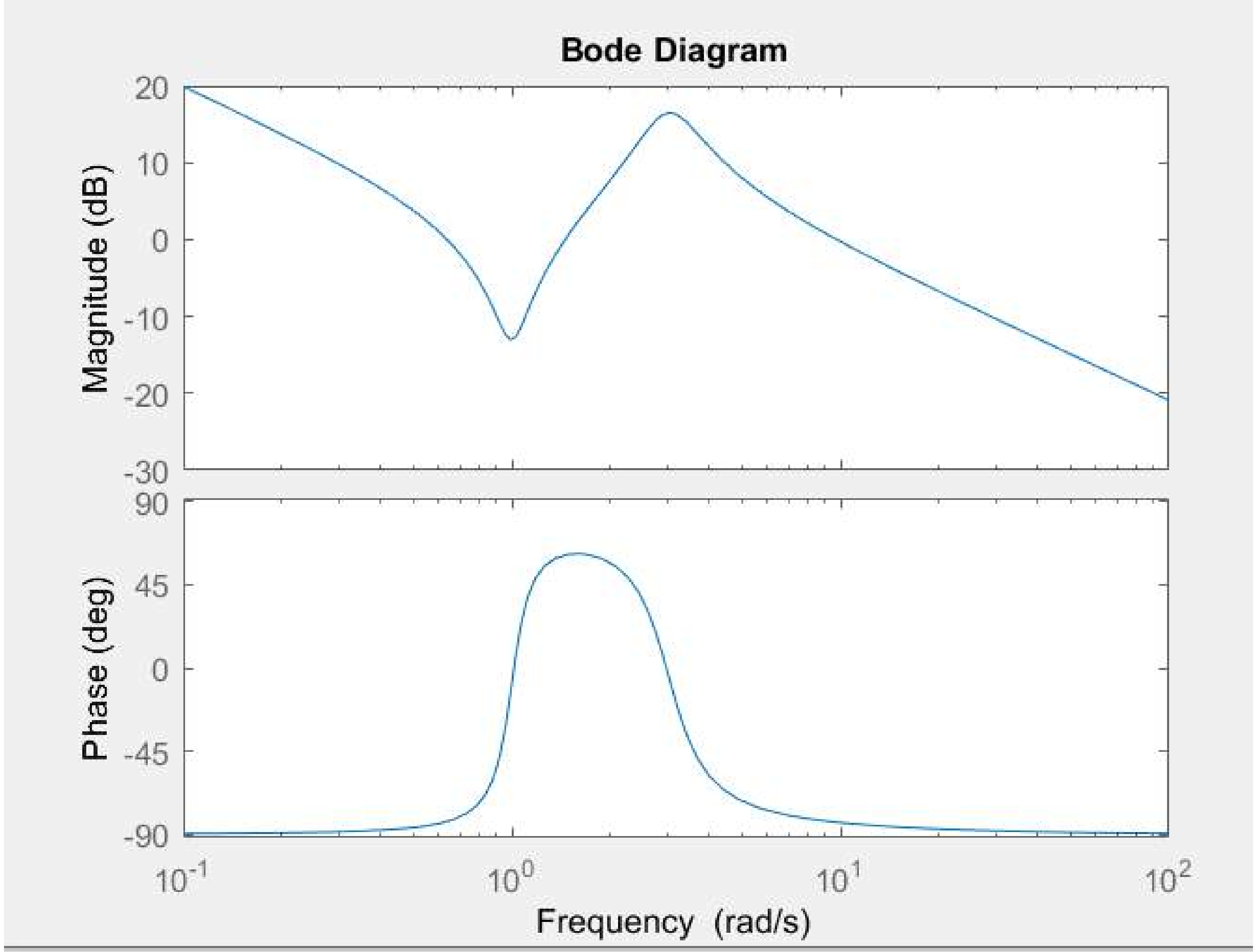
Command Window

```
>> example03

ans =

   16.6503       Inf    3.8730       NaN

>>
```

m.File link

example03.m

Bode plot



Bode Diagram
Magnitude (dB)
20
10
0
-10
-20
-30
Phase (deg)
90
45
0
-45
-90
10^{-1}
10^{0}
10^{1}
10^{2}
Frequency (rad/s)

Continued....

Experiment -09

PCEE-613,Control Systems (Lab)

GEE-2018

Aim of the Experiment: To draw a Nyquist plot of an open loop transfer function and examines the stability of the closed loop system.

Example01: Consider the following transfer function to draw a Nyquist plot with MATLAB and examine the stability of the closed-loop system

$$G(s) = \frac{20(s^2 + s + 0.5)}{s(s+1)(s+10)}$$

MATLAB code and Output

Code

Editor - C:\Users\Venus\Desktop\tfss.m

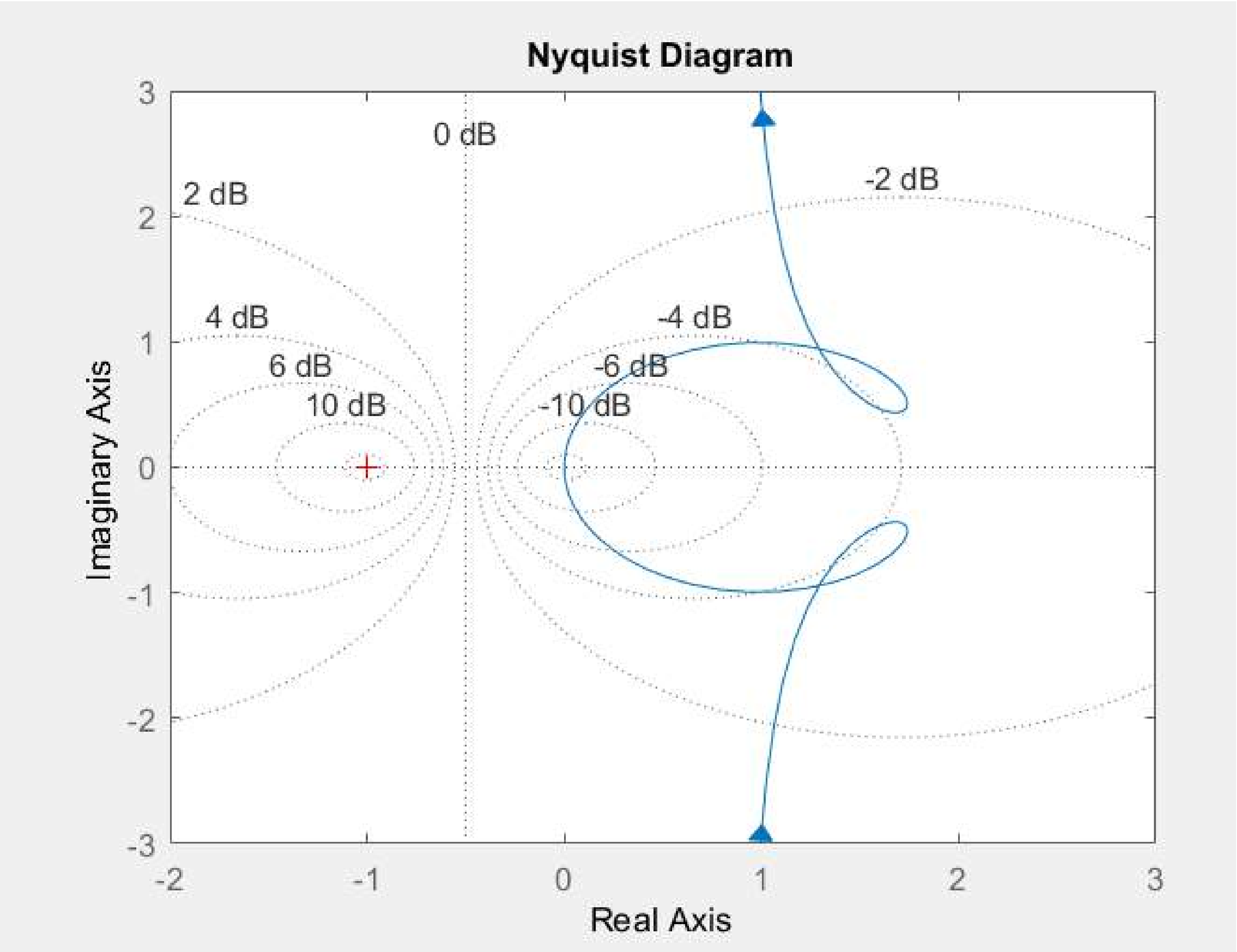
tfss.m

```
num = [20 20 10];
den = [1 11 10 0];
nyquist(num,den);
v=[-2 3 -3 3]; axis(v)
grid
```

m.File link

example01.m

Output



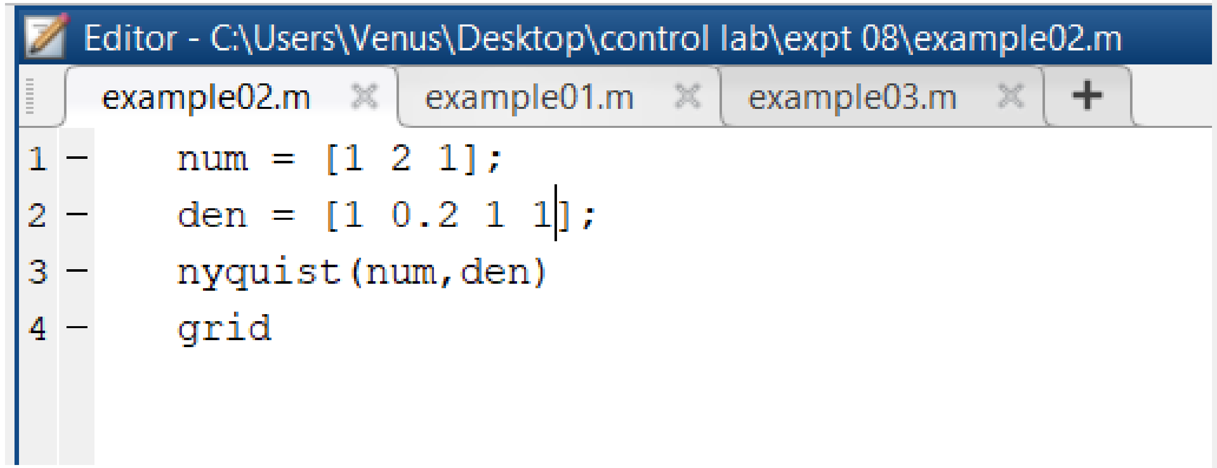
Example

Example02: Consider the following transfer function to draw a Nyquist plot with MATLAB and examine the stability of the closed-loop system

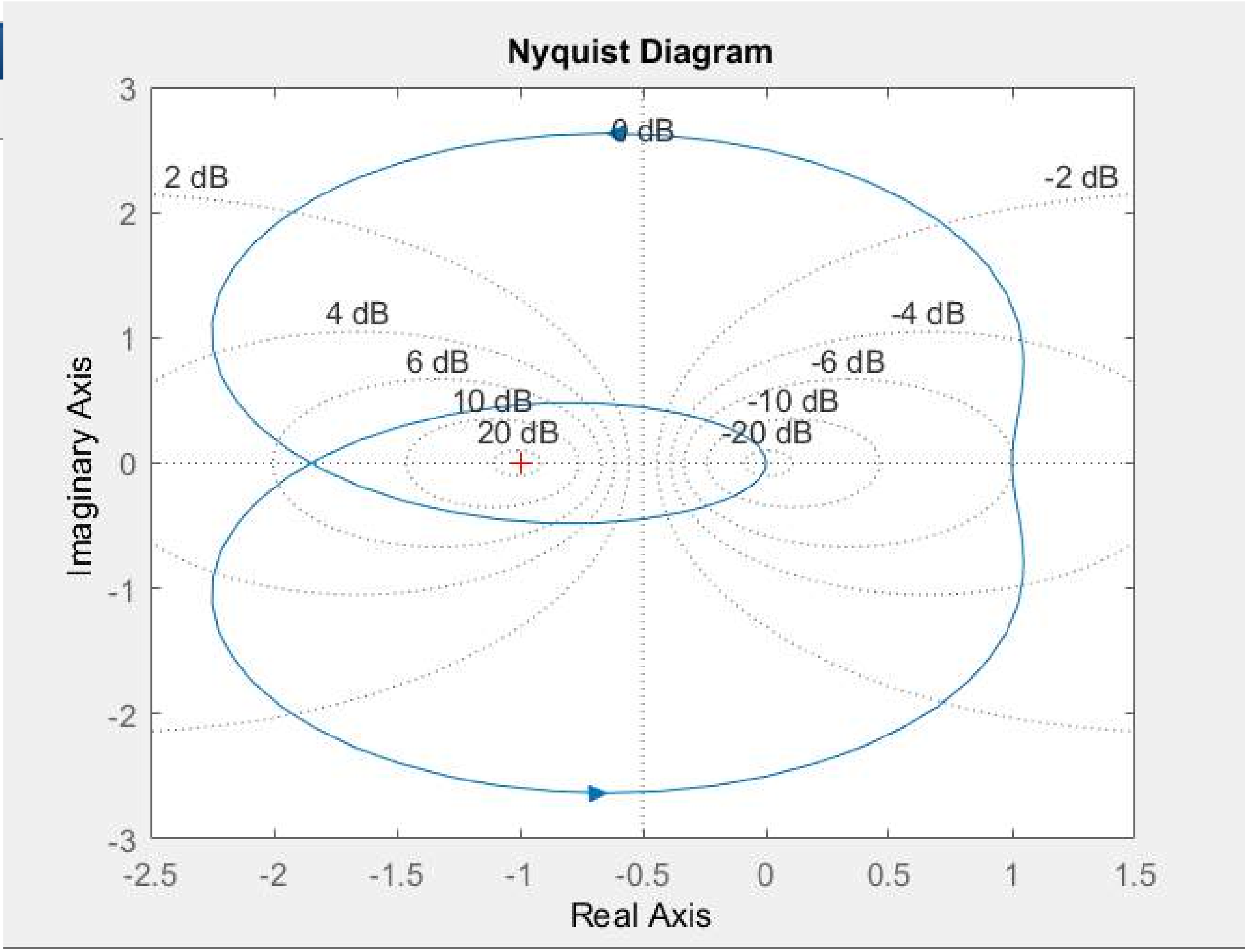
$$G(s) = \frac{s^2 + 2s + 1}{s^3 + 0.2s^2 + s + 1}$$

MATLAB code and Output

Code



Output



m.File link

example02.m

Example

Example03: Consider the following transfer function to draw a Nyquist plot with MATLAB and examine the stability of the closed-loop system

$$G(s) = \frac{s^2 + 4s + 6}{s^2 + 5s + 4}$$

MATLAB code and Output

Code

Editor - C:\Users\Venus\Desktop\control lab\expt 08\example03.m

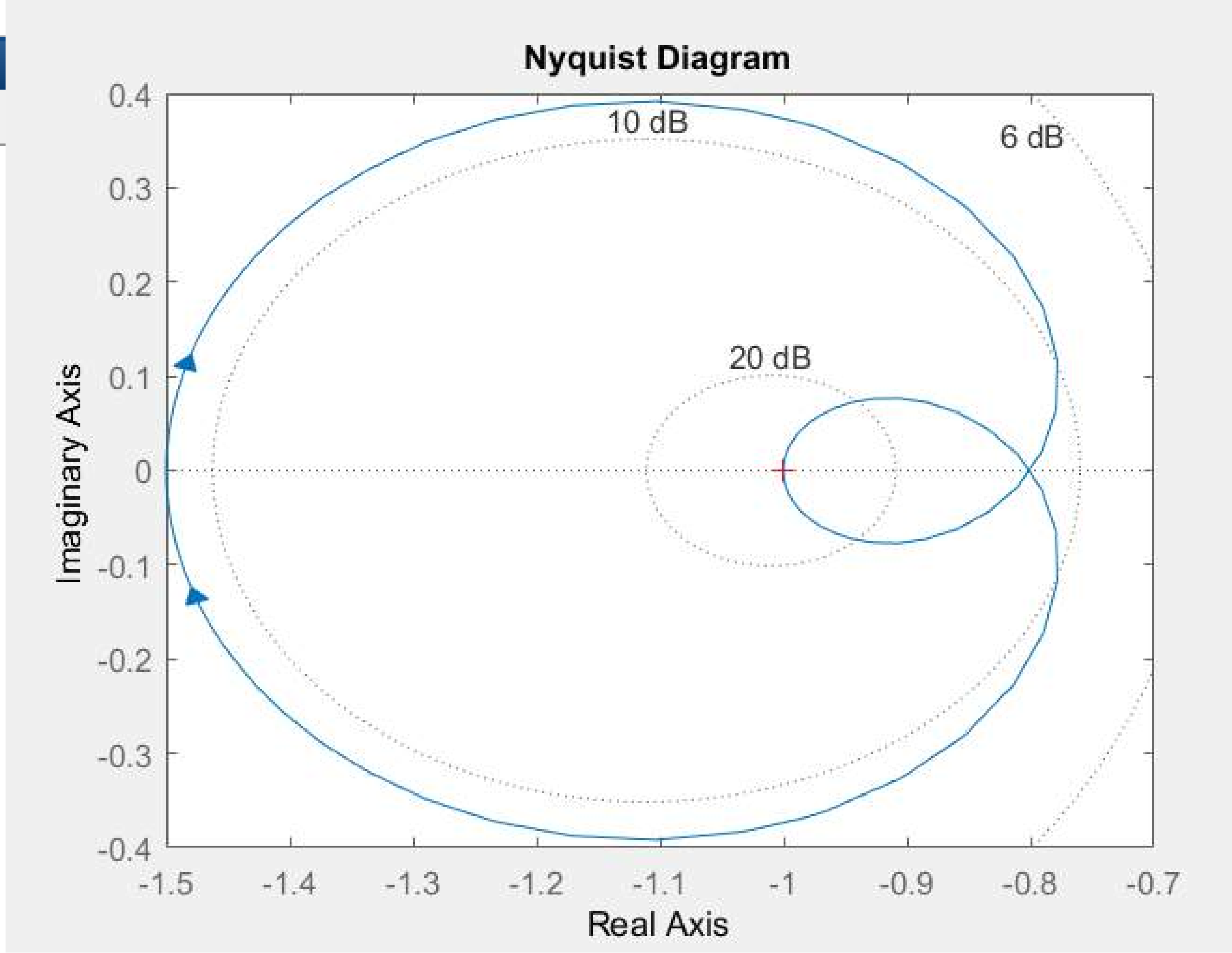
example02.m | example03.m

```
num = [-1 -4 -6];
den = [1 5 4];
nyquist(num,den);
grid
```

m.File link

example03.m

Output



Continued….

Experiment -10

PCEE-613,Control Systems (Lab)

GEE-2018

Aim of the Experiment:

To convert the transfer function of a system into state space form and vice-versa.

Theory: For the transfer function

$$\frac{Y(s)}{U(s)} = \frac{b_0 s^n + b_1 s^{n-1} + \cdots + b_{n-1} s + b_n}{s^n + a_1 s^{n-1} + \cdots + a_{n-1} s + a_n}$$

The state space representation is

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$$

$$y = \mathbf{C}\mathbf{x} + Du$$

Contd…

where

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_{n-1} \\ x_n \end{bmatrix}, \qquad \mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ 0 & 0 & 0 & \cdots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \cdots & -a_1 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \cdot \\ \cdot \\ \cdot \\ \beta_{n-1} \\ \beta_n \end{bmatrix}, \qquad \mathbf{C} = [1 \quad 0 \quad \cdots \quad 0], \qquad D = \beta_0 = b_0$$

Example

Consider the following transfer function and convert it into the state space representation using MATLAB code.

$$\frac{Y(s)}{U(s)} = \frac{s}{(s+10)(s^2+4s+16)}$$

$$= \frac{s}{s^3+14s^2+56s+160}$$

MATLAB code and Output

Code

Editor - C:\Users\Venus\Desktop\tfss.m

tfss.m

```
num = [1 0];
den = [1 14 56 160];
[A,B,C,D] = tf2ss(num,den)
```

m.File link

example01.m

Output

Command Window

```
A =

   -14   -56  -160
     1     0     0
     0     1     0


B =

     1
     0
     0


C =

     0     1     0


D =

     0

>>
```

Example

Consider the following state space equations and convert it into the transfer function using MATLAB code.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -5 & -25 & -5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 25 \\ -120 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

MATLAB code and Output

Code

Editor - C:\Users\Venus\Desktop\tfss.m

tfss.m

```matlab
A = [0 1 0; 0 0 1; -5 -25 -5];
B = [0; 25; -120];
C = [1 0 0];
D = [0];
[num,den] = ss2tf(A,B,C,D)
```

Output

Command Window

```
>> tfss

num =

         0         0   25.0000    5.0000


den =

    1.0000    5.0000   25.0000    5.0000

>>
```

m.File link

example02.m

Continued….