- **AIM:** Develop a data acquisition system using Arduino.

- **APPARATUS REQUIRED:**
  - → SR-04 Ultrasonic sensor
  - → Push Button
  - → Resistor 10k
  - → Breadboard
  - → Jumper wires
  - → Arduino UNO.
  - → LED

- **THEORY:**

Here, Ultrasonic Sensor is used with Arduino UNO to fetch the data of anything which is in range of it.

→ Arduino is a physical computing platform with open code, based on a board with a simple microcontroller and a development environment to create software such as Arduino IDE.

→ SR-04 ultrasonic sensor is a popular distance measurement system that works by emitting ultrasonic waves and then calculating the time taken for the waves to bounce back after hitting an object.

- PROCEDURE:

Step 1: Make connection as shown alongside page.
Step 2: write the code on the sketch and the upload.

"Code"

```
const int trigPin = 9;
const int echoPin = 10;
const int pb = 3;
const int led = 13;
void setup(){
  pinMode (trigPin, OUTPUT);
  pinMode (echoPin, INPUT);
  pinMode (pb, INPUT);
  pinMode (led, OUTPUT);
  digitalWrite (Led, LOW);
  Serial.begin (9600);
}
void loop (){
  int state = digitalRead (pb);
  if (state == HIGH) { for(int i=0; i<1; i++)
  {
   long duration, cm;
   digitalWrite (trigPin, LOW);
   delay(2000);
   digitalWrite (trigPin, HIGH);
   delay (5000);
   digitalWrite (trigPin, LOW);
   duration = pulseIn (echoPin, HIGH);
```

```
cm = microsecondsToCentimeters (duration);
long inch = cm/2.54;
duration = pulseIn (echoPin, LOW);
Serial.print ("DATA, TIME");
Serial.println (inch);
digitalWrite (led, HIGH);
delay (1000);
digitalWrite (Led, LOW);
}
}
else {
    digitalWrite (trigPin, LOW);
    digitalWrite (Led, LOW);
}
}
long microsecondsToCentimeters (long microseconds){
 return microseconds/29/2;
}
```

Step 3 : Then install PLX-DAQ.

link : https://plx-daq.software.informer.com.

- then open sheet and select enable this conten
- then select proper COM port and Baudrate.

CONCLUSION :

The software will now recieve data every time when we click the push button (Pb).

Thus, our practical is successfully observed.

- **AIM:-** "Temperature control system using PID".

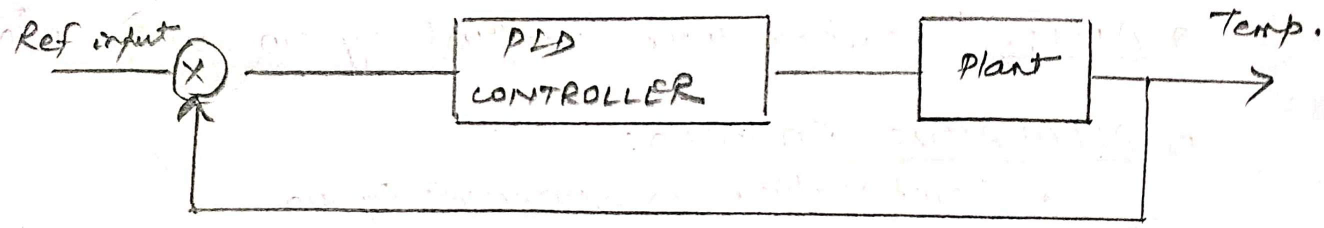
- **APPARATUS REQUIRED:**
  - Temperature measurement system
  - Solid-state relay for driving heater bulbs.
  - Micro controller based control unit.
  - LCD display.

- **THEORY:**

This set up is designed to demonstrate the working of a typical temperature controller using PID mode of operation.

- Proportional controller is a mode of control action in which there is a continuous linear relationship between values of deviation and manipulated variable.
- In order to remove the offset associated with proportional action, combination of P+I is widely used, as a result of integral action, the offset error is almost reduced to zero but the transient response is adversely affected.
- A derivative control action may be added to proportional control to form P+D action. Derivative control action may be defined as control action in which the magnitude of the manipulated variable is proportional to the rate of change of error.
- P+I+D action produces smallest maximum deviation and offset is eliminated because of

Representation of controller monitoring the plant:



integral action. The derivative action provides improved transient response against load variations. In short PID approach to control problem can be summarized in terms of-

The mathematical eqns. governing the operation.

$$P(t) = K_p e(t) + K_i \int e(t)\, dt + K_d \frac{de(t)}{dt} + P(0).$$

A simple analysis would show that the derivative block essentially increase the damping ratio of the system and therefore improves the dynamic performance by reducing overshoot, the integral action eliminates the steady state error.

PROCEDURE :

For Proportional Control (P).

(i) Keep SW3 in Test mode.

(ii) Keep SW2 in mode check.

(iii) Then Keep $K_i = 0$ & $K_d = 0$, now system will be configured for proportional mode.

(iv) Make proper connection for heater cable, RTD cable.

(v) Now select system as SW3 in TEST, SW2 in Normal, SW1 in PID side.

(vi) If selection is kept as above you will be able to set temperature with the help of PI.

(vii) Now select system as SW3 & START, SW2 in NORMAL, SW1 is PID.

(viii) Now system will start in proportional mode.

Topic : Experiment - 2

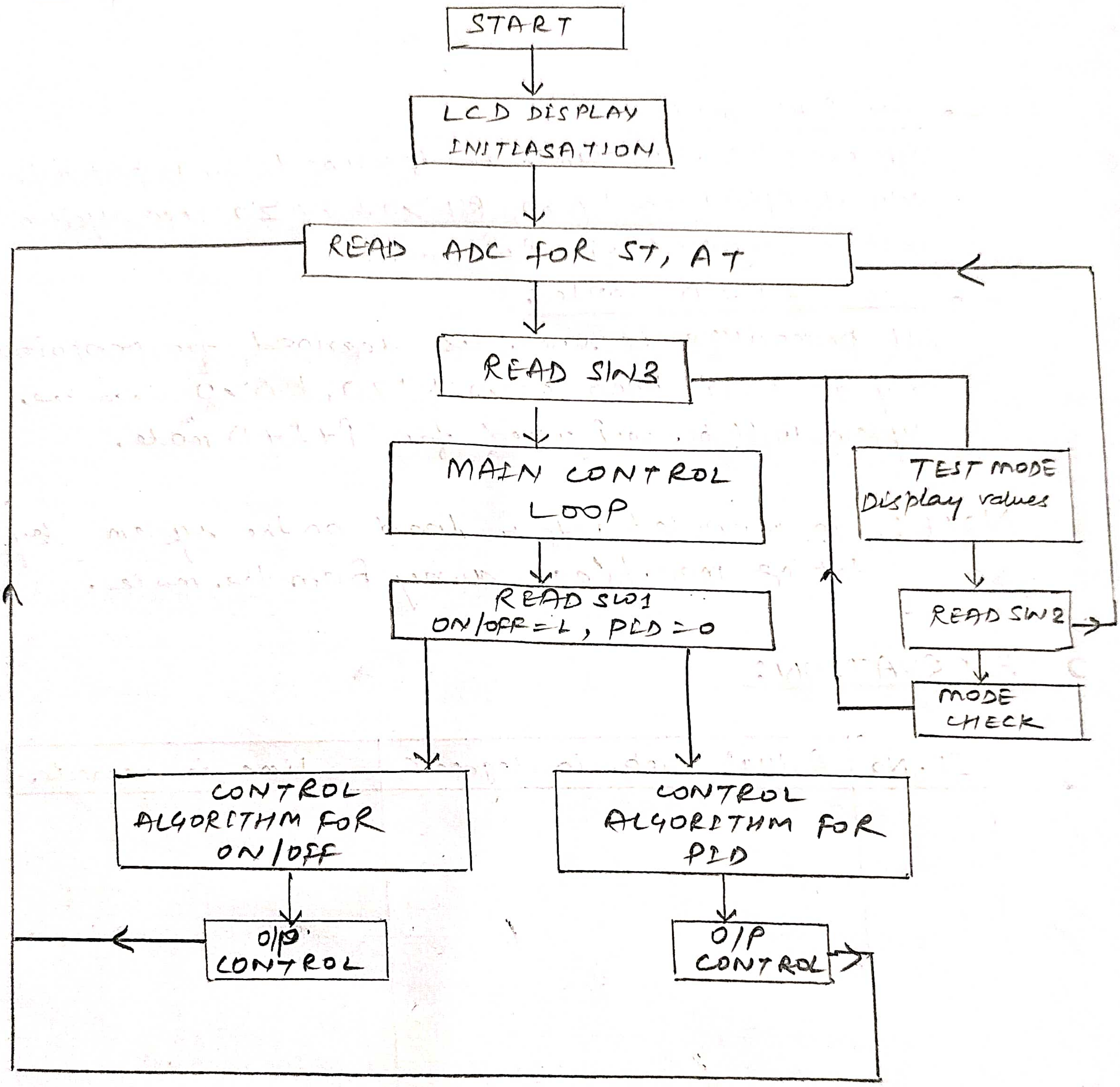
→ for P+I mode :
All procedure is same as described for proportional mode. Only at P+(3) keep Kp>0, KI>0 & KD=0 then system will configured for P+I mode.
→ for P+I+D mode :
All procedure is same as described for proportional. Only at P+(3) keep KP>0, KI>0, KD>0 then the system will be configured for P+I+D mode.

NOTE: fan provided acts as load on the system by taking more/less away from the modes.

- OBSERVATION :

SI. No	Actual temp. in degree C	time in seconds.

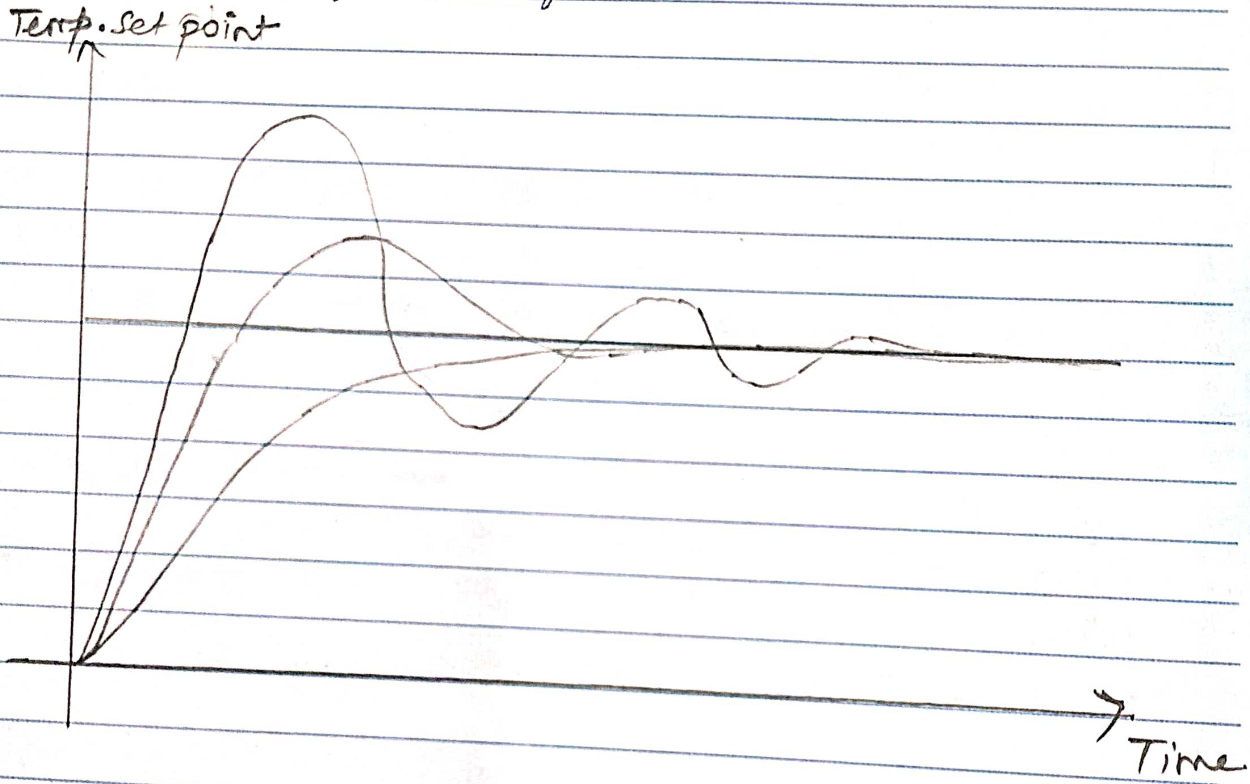
FLOWCHART:



Shot on OnePlus
by Purushotta...

- **Characteristics:**

The typical PID characteristics for different settings may be as follows:



- **RESULT:**

Temperature of oven is controlled by using PID controller. By varying P, I, D values of two graphs of temperature vs time is plotted.

Topic : EXPERIMENT- 3.

- **AIM:** "Level Control System based on error feedback."

- **APPARATUS REQUIRED:**
  - → A tank of water
  - → CE117 Process Trainer Rig
  - → CE2000 software.

- **THEORY:**

— This lab is carried out with a rig consisting a tank of water where the water level in a process vessel tank was used as PV (prese to be controlled.

→ There are two main control systems which are the opened and closed loop control systems. Closed loop system is also known as a feedback control system where the control action depends on the output.
The control action adjusts automatically when a difference betn the set point (SP) and the process variable (PV) is identified. A SP would be fixed in process where the control system alters the manipulated variable (MV) at the desired condition to minimize the error over tim

→ One of the popular feedback control loops includes the proportional- integral - derivative (PID) controller. This is because the controller can calculate the MV which depends on the ~~most~~ combinating of the correcting terms which includes proportional correcting term output, integral correcting term output and derivative correcting term output.
This is shown below:

$$MV(t) = P_{out} + I_{out} + D_{out} \quad \text{---(i)}$$

where,

$$P_{out} = K_c\, \varepsilon(t) \quad \text{---(ii)}$$

$$I_{out} = \frac{1}{\tau_i}\int_0^t \varepsilon(t)\,dt \quad \text{---(iii)}$$

$$D_{out} = K_d \cdot \frac{d}{dt}\varepsilon(t) \quad \text{---(iv)}$$

PROCEDURE:

- The CE117 Process trainer rig and CE2000 software are used.
- And then the file "exp43.ict" was loaded under the CE2000 software in the computer.
- The loop bypass valve was closed while the air vent was opened fully.
- The process drain was opened to approx. 45°.
- CE117 Mimic Panel was connected with external wires.
- Pump 2 was switched to 'external'. By varying K_c to 1,

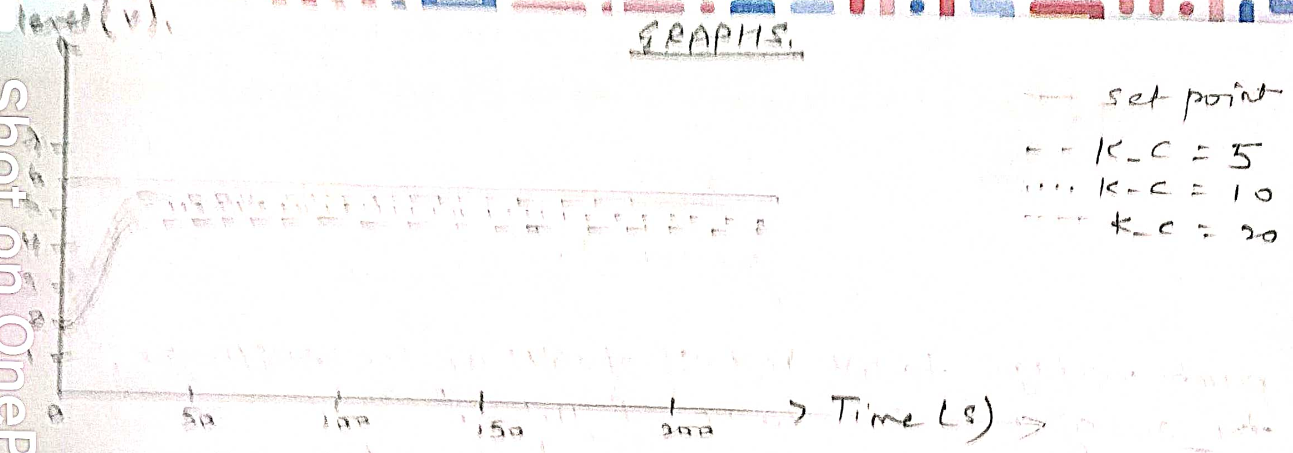
pump voltage to 5V and SP to 6V on the software, the relay was switch on at 0.5V.

- The process vessel drain was fully opened now.
- The SP was adjusted to 6V and valve voltage to 1V.
- The PID controller was set to Proportional Integral control which included P = 5, I = 0 and D = 0.
- The results were recorded once again until the level stabilized.
- The data were exported before the process was repeated by varying the P value to 10 and 20.
- Now, the system was left for 10 minutes to allow the level to stabilize before recording the height of water in the process as level A.
- The pump voltage was increased by 0.5V. Once the level stabilized, the second height was recorded as level B.

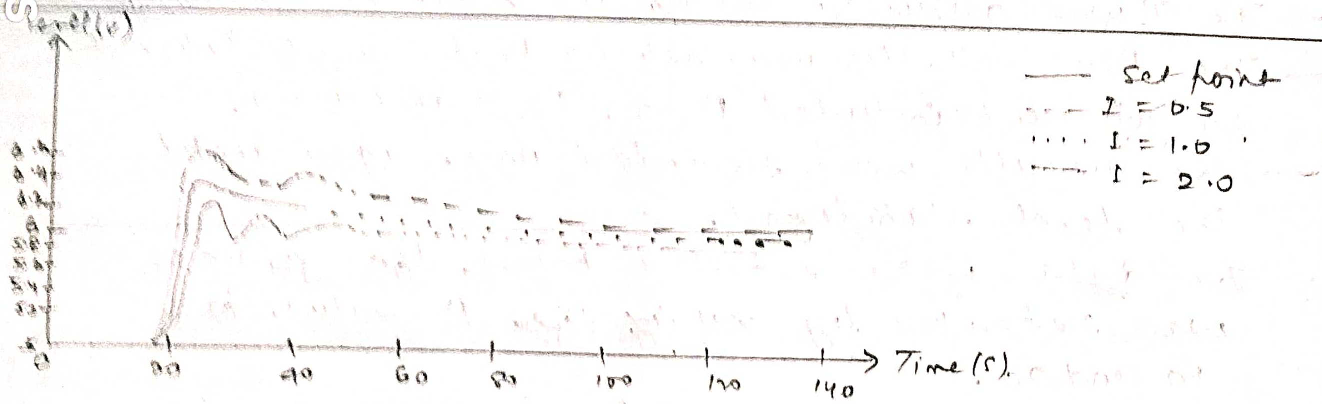
RESULTS AND CALCULATIONS:

- fig 3: shows the results from P, PI and PID controllers as fig 4 and figure 5 also.
- fig 3: was to investigate the proportional control fig 2 and fig 3, as Kc increases, the closer the output is to the set point. From fig 3, the proportional control leaves an offset error in the final steady-state condition.

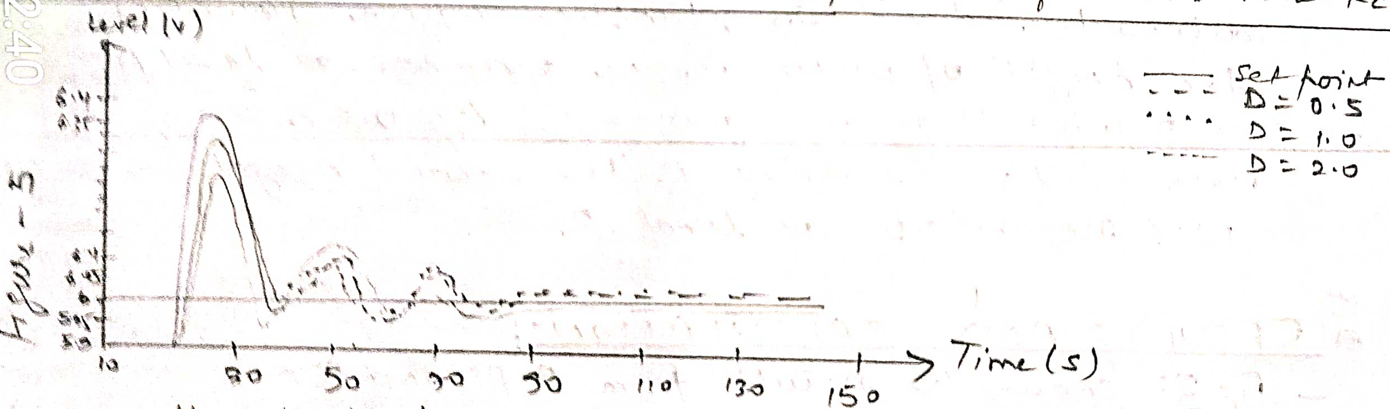
GRAPHS.



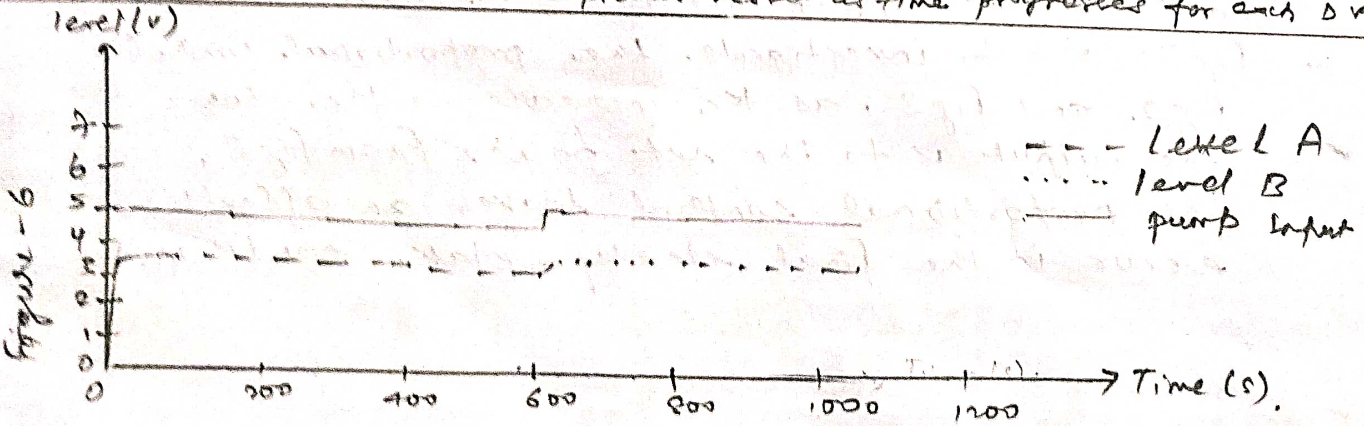
Here, Kc value of 20 has the smallest error.



Here, the process vessel as time progresses for each I value $K_c = 20$.



Here, the level in the process vessel as time progresses for each D value with P = 20 and I = 2.



Here, the process vessel as time progresses with an increase in pump voltage.

The data from uncontrolled system was recorded as follows:

Level	Flowrate, q (V)	Flowrate, Q (L/min)	Flowrate, Q (m^3/s)	Liquid level, h (m)
A	3.6	3.6	6.00 E -05	0.116
B	3.8	3.8	6.33E -05	0.136
		Difference (Δ)	3.33 E -06	0.020

- CONCLUSION:

It can be concluded that the output of controller depends on the value of PV.
A PID controller continuously minimize the calculated error value which the difference between the PV and SP over time by adjusting MV.
Here, the most efficient combinations of the parameters were P = 20, I = 2, D = 2.
The uncontrolled system was less reliable since manual adjustment was required compared to the controlled system where the adjust happens automatically.

~~[illegible]~~

Topic : EXPERIMENT-4

- **AIM:** "PLC programming using Relay ladder logic for AND, OR, XOR and NOR gate".

- **APPARATUS:**
→ Virtual lab.

- **THEORY:**

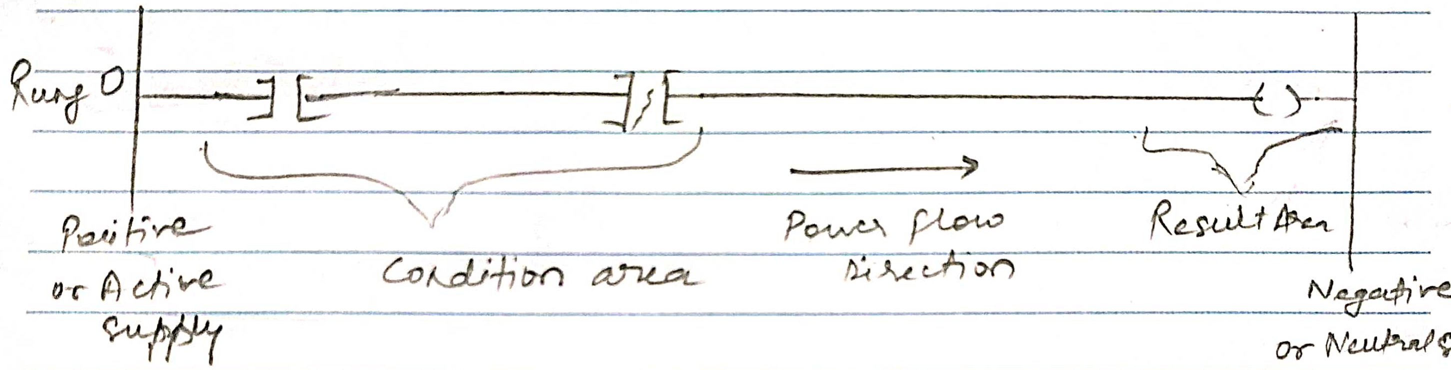
Introduction

→ Each manufacturer of PLC systems has own style of writing the instructions. Different PLCs has different instruction set but even some common basic instructions are shared by all the PLCs.
→ Ladder is most commonly used programming language.
→ Ladders were developed to mimic or imitate relay logic.

Ladder programming:
→ Ladder diagram shows the sequence of the logic execution which is presented diagrammatically.
→ In ladder diagram, there are two vertical lines generally called as Phase (positive) or neutral.
→ Rungs which show current flow in horizontal direction are the sequence in which the logic executes.

→ The analogous to relay, ladder has two main symbols which are contacts and output coil.

→ An instruction in Ladder instructs PLCs how to the bits in I/O files which are stored in the memory.



→ Most commonly used relay instructions used in PLC programming are as shown the table —

Instruction	Symbol	Description
Examine ON (normally open).	┤├	An input condition on that is open when de-energized.
Examine OFF (normally closed)	┤/├	An input condition on that is close when de-energized.
Output coil	—[ ]—	An output instruction that is true when the input conditions become true.
Negated o/p	—[/]—	
Latch o/p coil	—[L]—	
Unlatch	—[U]—	

Truth tables.

I/p		O/p
X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

Truth table of AND

I/p		O/p
X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

OR

I/p		O/p
X	X	Z
0	0	0
0	1	1
1	0	1
1	1	0

XOR

I/p		O/p
X	Y	Z
0	0	1
0	1	0
1	0	0
1	0	0

NOR.

PROCEDURE:

Step 1: Prior to starting of the development of ladder diagram.

Step 2: The step for writing the ladder diagram on the work space as follows.

(i) Add a new rung by clicking on the "Add Rung" icon.

(ii) Place the contacts as per the requirement by left clicking the appropriate conta[...] shown at the top side.

(iii) Right click on the contact or coil and you can give name like "start", "stop" etc. Please ensure that the tag numbers are true replica of process connections.

(iv) Click the compile button so the ladder will be ready for running. For testing the logic you need to click on Run. Both sides of the rung will become green and this is the indication of run mode.

(v) By clicking on the contact toggle the state of contact.

(vi) You can add seven elements in series and 5 elements in parallel.

(vii) Repeat the procedure and verify your logic.

(viii) Similarily you can check the logic for OR, NOR, and NAND gates. Validate the truth tables and confirm the results.

RESULTS AND CONCLUSIONS:

Outcomes are given graphically in the next page. And hence, we have successfully done the ladder programming using PLC to implement the AND, OR, XOR and NOR gates.

- **AIM:** "PLC timer, counter, registers and analog input / output functions."

- **APPARATUS REQUIRED:**
  → Virtual Lab.

- **THEORY:**

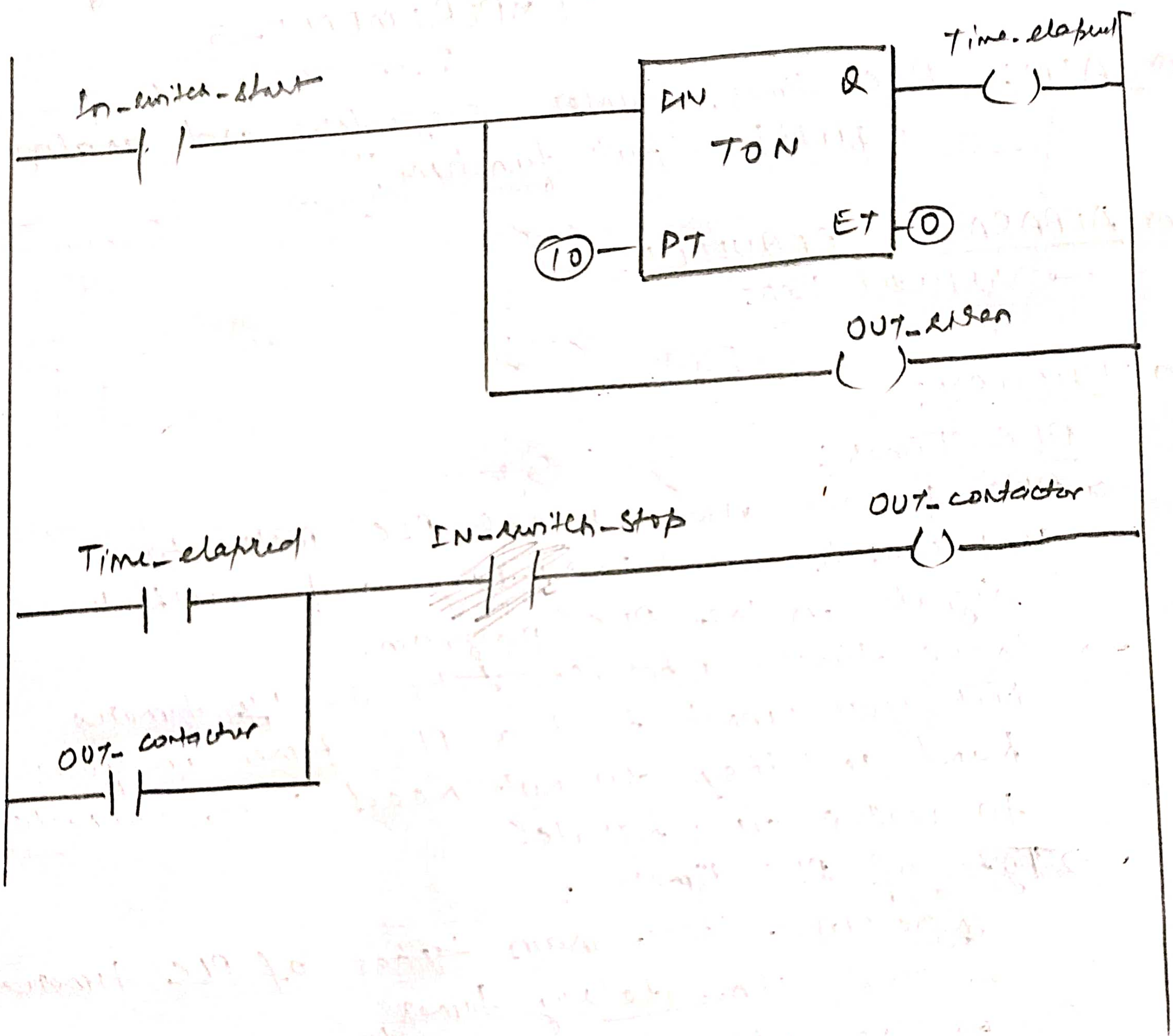
  PLC Timer:

→ PLC timers are internal PLC instructions that can be used to delay and output signals in the PLC program.

→ These timers operate like relay timers but you cannot hold a PLC time in your hand and they do not need to be connected to wires to operate.

→ Types of PLC timers:
There are three main types of PLC timers:
- The On-delay timer,
- The off-delay timer.
- The retentive on-delay timer.

→ All of these PLC timers have a timer field name, a preset value, an accumulated value, a timer enabled bit, a timer timing bit, and a time done bit.



In_switch_start
EN
TON
Q
Time_elapsed
10
PT
ET
0
OUT_siren
Time_elapsed
IN_switch_stop
OUT_contactor
OUT_contactor

PLC Timer

Topic : Experiment -5

PLC counter:

→ PLC counters are internal PLC instructions that can be used to count input or output signals in the PLC program.

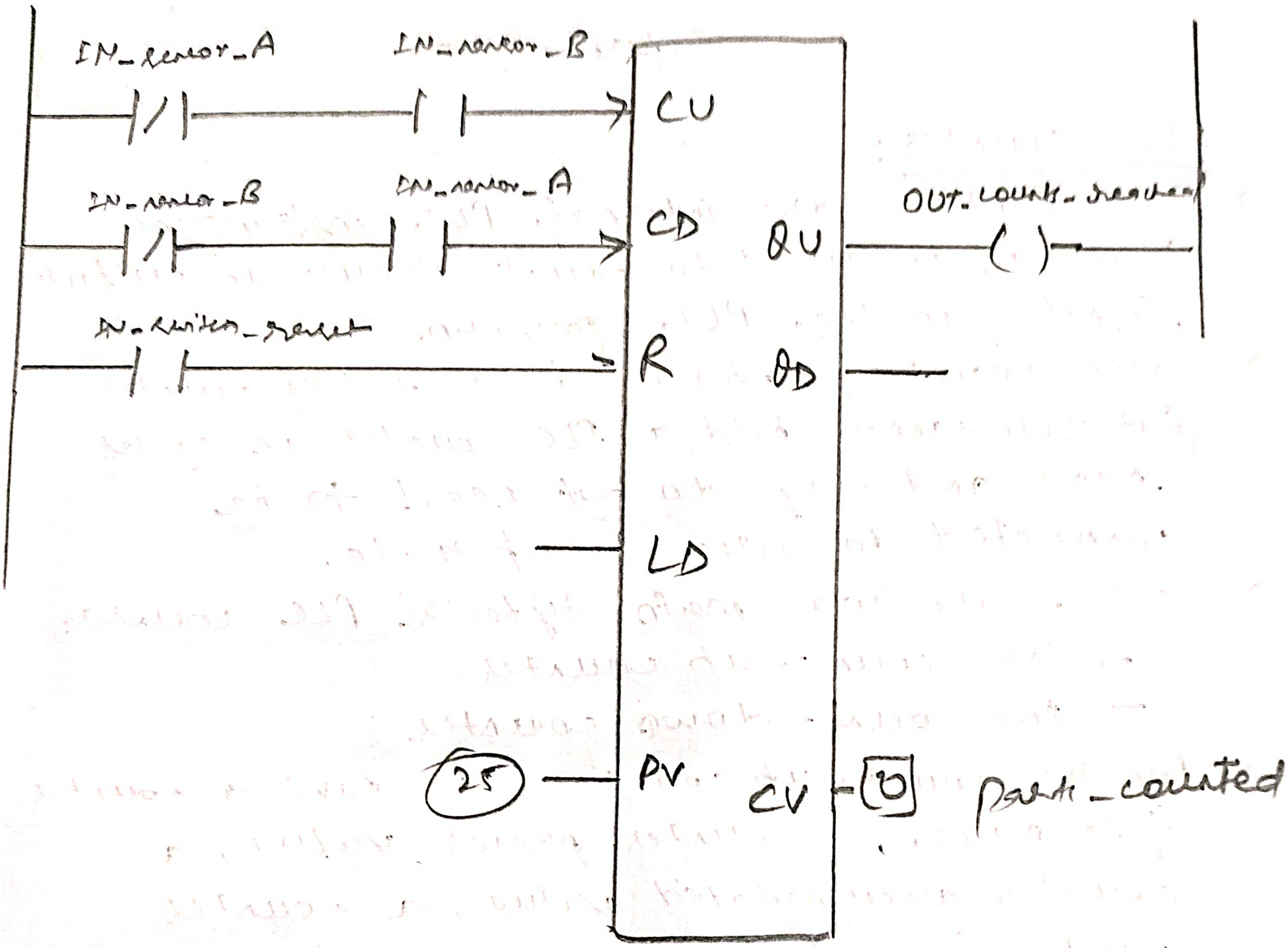
→ These counters operate like relay counters but you cannot hold a PLC counter in your hand and they do not need to be connected to wires to operate.

→ There are two main types of PLC counters
- the count-up counter
- the count-down counter.

→ for the count-up counters, we have a counter file name, a counter preset value, a counter accumulated value, a counter count-bit, and a counter done bit.

→ for the counter-down counters, we have the same bits and values.
The only thing that is different is the counter down bit on the output.

→ There are also two internal bits of counter overflow and counter underflow which we, will more efficient.


IN_sensor_A
IN_sensor_B
CU
IN_sensor_B
IN_sensor_A
CD
QU
OUT_count_reached
R
QD
LD
25
PV
CV
0
parts_counted

PLC counter

PROCEDURE :

Step 1: we use studio 5000, which the PLC programming software for Allen Bradley. Enter the timer preset so that it equal ten seconds.
Now, On-delay timer is set up.

Step 2 : when you turn on the selector switch the timer enabled bit and timer timing bit turn on and the timer's accumulated value starts accumulating time.

Step 3 : At this point, when you turn off the selector switch the timer accumulated value will reset back to zero, and the timer done bit will turn off.

For counter.

Step 1: Download in PLC emulator, counter program.

Step 2: When I push and hold the button, the count-up bit turns ON, and the accumulated value changes to one.

Step 3: When I release the button, the count-up bit turns off but the accumulated value stays at one.
So, every time I push the button, the count, the count-up bit turns on, and the accumulated value increases by one.

Step 4: If I continue to push and release the button, the count-up bit will still turn on while the button is pushed, the done bit will stay on, and the accumulated value will still increase.

RESULT:

we studied and done practically the PLC timer and counter, register and input/output of analog. successfully.