

**DEPARTMENT OF ELECTRONICS & COMMUNICATION
ENGINEERING**

COLLEGE OF ENGINEERING TRIVANDRUM

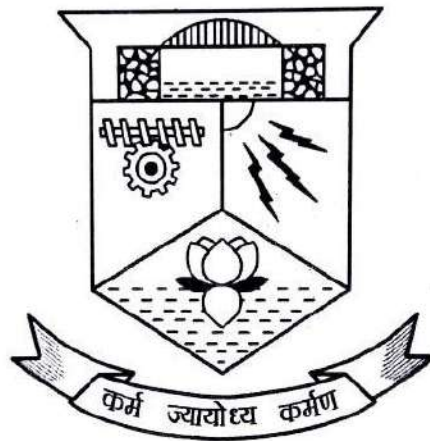
Laboratory Manual

of

AE332

PROCESS CONTROL LAB

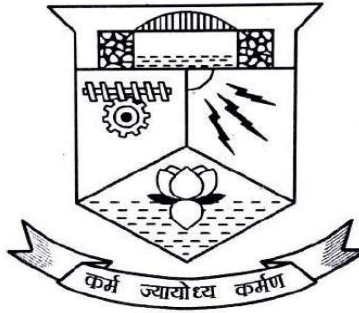
(PART 2 of 2)



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
COLLEGE OF ENGINEERING TRIVANDRUM**

2019

**DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING
COLLEGE OF ENGINEERING TRIVANDRUM**



CERTIFICATE

*This is a controlled document of **Department of Electronics and Communication Engineering of College of Engineering Trivandrum**. No part of this can be reproduced in any form by any means without the prior written permission of the Head of the Department, Department of Electronics & Communication Engineering, College of Engineering Trivandrum. This is prepared as per **2016 B.Tech Applied Electronics & Instrumentation Engineering** and **2015 M. Tech Applied Electronics & Instrumentation schemes**.*

Prepared by:

Nirmal Syam S. J., Sariga Sachit
Students (M₃ AEI)
Dept. of ECE
College of Engineering Trivandrum

Reviewed by:

Prof. Sumi George
Asst Professor
Dept. of ECE
College of Engineering Trivandrum

Compiled by:

Dept. of ECE
College of Engineering Trivandrum

Dr. Santhosh Kumar K S

Head of Dept.
Dept. of ECE
College of Engineering Trivandrum

ADVANCED PROCESS CONTROL LAB

SERIAL NO.	NAME OF THE EXPERIMENT	PAGE NO
1.	Controller Tuning - Simulation	4
2.	Study of Feed-forward control	11
3.	Study of Cascade control	16
4.	Study of Ratio control	25
5.	Data Logger	30
6.	Study of PLC-Ladder diagram and implementation of logic circuits	37
7.	PLC controlled Bottle filling system	40
8.	PLC controlled Water level system	44

EXPERIMENT NO. 1

CONTROLLER TUNING- SIMULATION

Aim

To study the Controller tuning by simulation:

- To PID tune a Second Order system in MATLAB
- To PID tune a DC motor in MATLAB.

Apparatus Required

MATLAB Software

Theory

MATLAB (*matrix-laboratory*) is a proprietary multi-paradigm programming language and numeric computing environment developed by MathWorks. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages. including C, C++, C#, Java, Fortran and Python.

SIMULINK

Simulink, developed by MathWorks, is a graphical programming environment for modeling, simulating and analyzing multi-domain dynamical systems. Its primary interface is a graphical block diagramming tool and a customizable set of block libraries. It offers tight integration with the rest of the MATLAB environment and can either drive MATLAB or be scripted from it. Simulink is widely used in automatic control and digital signal processing for multi-domain simulation and model-based design. The construction of a model is simplified with click-and-drag mouse operations. Simulink includes a comprehensive block library of toolboxes for both linear and nonlinear analyses.

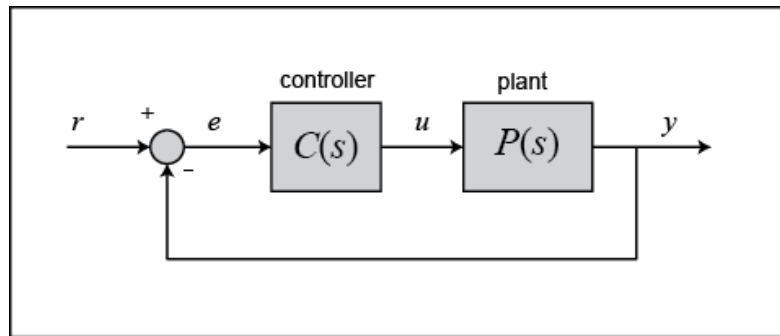


Fig. 1.1 Block diagram of a Unity-feedback System

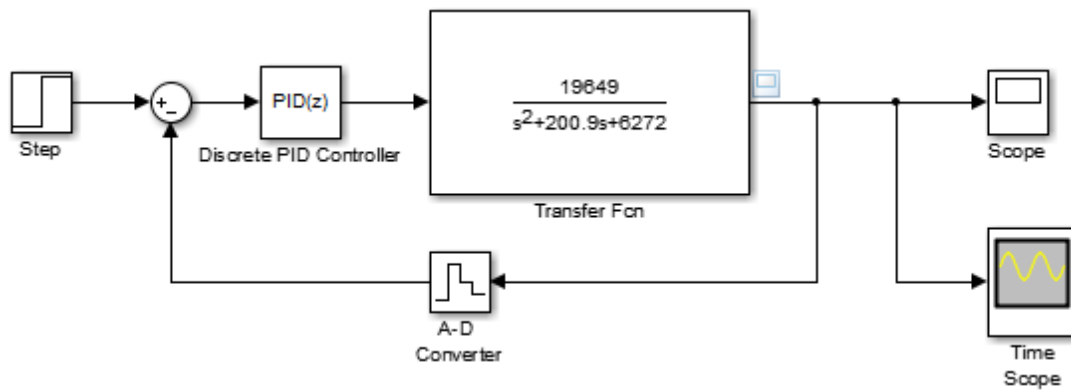


Fig. 1.2 PID tuning of Second order system; Simulink model

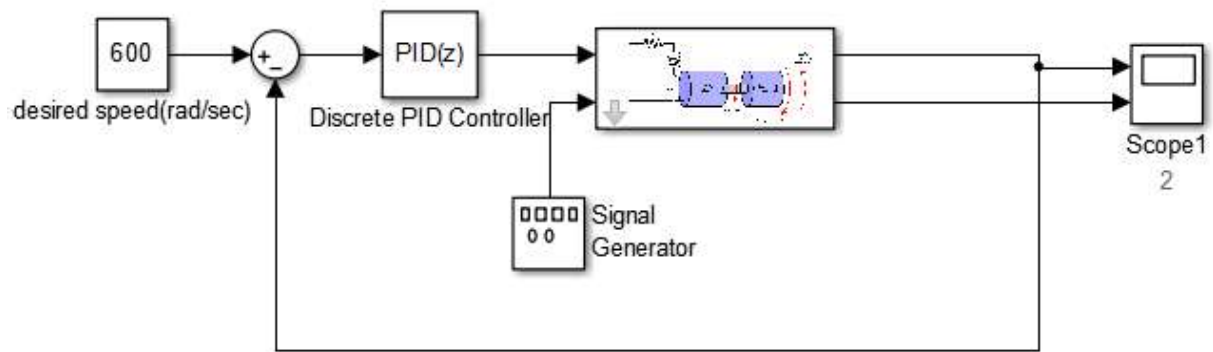


Fig.1.3 PID tuning of DC Motor; Simulink model

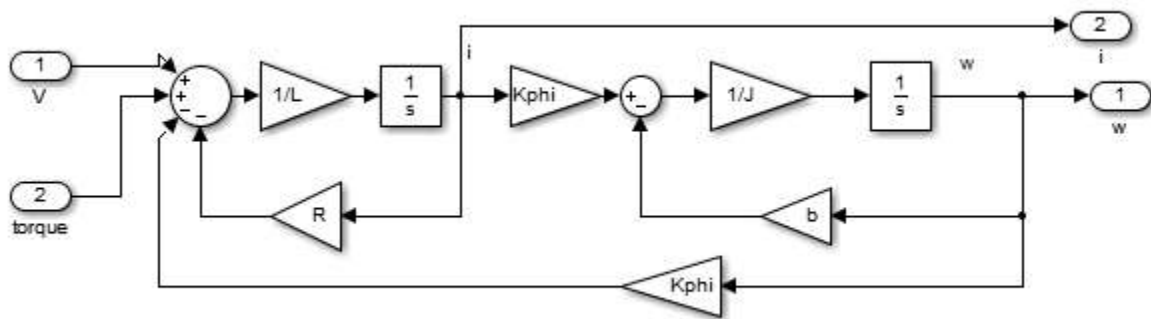


Fig.1.4 Equivalent subsystem model DC motor in Simulink

From MATLAB command window enter Simulink, or alternately, click on the Simulink icon located on the toolbar

The block library is organized into functional groups. For instance

- Sources: Blocks for the generation of input signals (steps, sinusoids etc.)
- Sinks: Blocks for the graphical visualization of signals
- Math: Blocks for the mathematical elaboration of signals
- Continuous: Blocks for the definition of (continuous) and so on

SIMULATION

With Simulink, you can interactively simulate your system model and view the results on scopes and graphical displays. For simulation of continuous, discrete, and mixed-signal systems, you can choose from a range of fixed-step and variable-step solvers. Solvers are integration algorithms that compute system dynamics over time.

PID CONTROLLER

The PID Controller block implements a PID controller (PID, PI, PD, P only, or I only). The block is identical to the Discrete PID Controller block with the Time domain parameter set to Continuous-time. The block output is a weighted sum of the input signal, the integral of the input signal, and the derivative of the input signal. The weights are the proportional, integral, and derivative gain parameters. A first-order pole filters the derivative action. The output of a PID controller, which is equal to the control input to the plant, is calculated in the time domain from the feedback error as follows:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de}{dt}$$

The block supports several controller types and structures. Configurable options in the block include:

- Controller type (PID, PI, PD, P only, or I only)
- Controller form (Parallel or Ideal)
- Time domain (continuous or discrete)

Observations

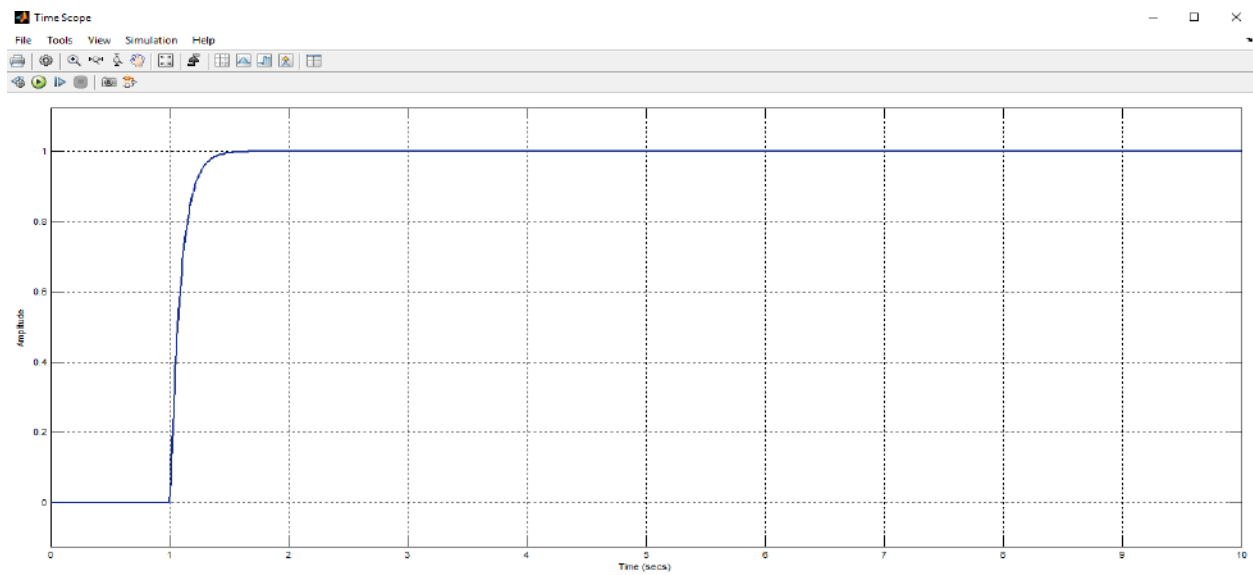


Fig.1.6 Step Response of Second Order System

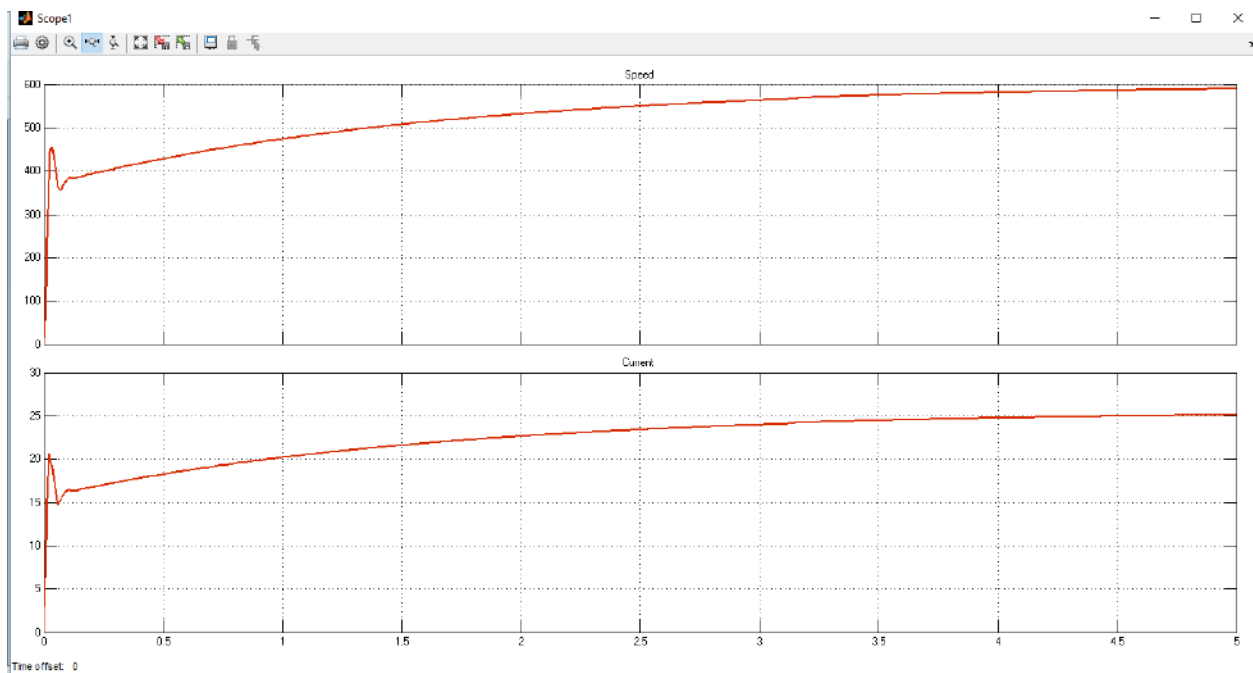


Fig.1. 7Step Response of DC motor for Set Point Speed = 600 rpm

- Initial conditions and reset trigger
- Output saturation limits and built-in anti-windup mechanism
- Signal tracking for bump less control transfer and multi-loop control

As you change these options, the internal structure of the block changes by activating different variant subsystems. To examine the internal structure of the block and its variant subsystems, right-click the block and select Mask > Look under Mask.

PID TUNING

PID tuning is the process of finding the values of proportional, integral, and derivative gains of a PID controller to achieve desired performance and meet design requirements. PID controller tuning appears easy, but finding the set of gains that ensures the best performance of your control system is a complex task. Traditionally, PID controllers are tuned either manually or using rule-based methods. PID Tuning can be done manually by Zeigler or Cohen coon, (the selection of which depends on the system) or automatically by the feature in MATLAB PID Tuner. The PID Tuner app automatically tunes the gains of a PID controller for a SISO plant to achieve a balance between performance and robustness. You can specify the controller type, such as PI, PID with derivative filter, or two-degree-of-freedom (2-DOF) PID controllers. Analysis plots let you examine controller performance in time and frequency domains. You can interactively refine the performance of the controller to adjust loop bandwidth and phase margin, or to favor setpoint tracking or disturbance rejection.

For instance, simple manual tuning based on Zeigler method is implemented on the examples referred below. Here all P, I, D values are reset at the beginning and varied gradually by trial and error (Vary P only, $I = D = 0$ then I is varied, $D=0$, then D is varied) to get the best response. Manual tuning methods are iterative and time-consuming, and if used on hardware, they can cause damage. Rule-based methods also have serious limitations: they do not support certain types of plant models, such as unstable plants, high-order plants, or plants with little or no time delay.

You can automatically tune PID controllers to achieve the optimal system design and to meet design requirements, even for plant models that traditional rule-based methods cannot handle well. An automated PID tuning workflow involves:

- Identifying plant model from input-output test data
- Modelling PID controllers in MATLAB using PID objects or in Simulink using PID Controller blocks.
- Automatically tuning PID controller gains and fine-tuning your design interactively
- Tuning multiple controllers in batch mode
- Tuning single-input single-output PID controllers as well as multi-loop PID controller architectures.

Procedure

- Open MATLAB
- Home>> Click on New>> Simulink Model
- Create a model as in fig 1.2 and 1.3
- Click on Run button.
- Tune the PID values manually by trial and error until the best response is obtained.

NB

- For Transfer Function Block; Simulink Library>>Simulink>>Continuous>> Transfer fcn.
- For PID Controller; Simulink Library>>Simulink>>Discrete>> Discrete PID Controller.
- For Signal Sources; Simulink Library>>Simulink>>Sources
- For SCOPES; Simulink Library>>Simulink>>Sinks

Result

Controller Tuning of a Second Order System and DC Motor is studied, simulated using MATLAB.

EXPERIMENT NO. 2

STUDY OF FEED-FORWARD CONTROL

Aim

To learn feed forward control for Flow control & temperature control

Apparatus required

- Feed forward control trainer
- Patch cords
- Reservoir filled with enough water

Valve positions:

HV1 - Min open [Process tank inlet valve]

HV2 - Open [Bypass valve from overhead tank to reservoir]

HV3 - Min open [Process tank outlet valve]

HV4 - Closed [Process and heat valve]

HV5 - At Least 75% open [By pass valve]

Theory

This LTPC-034 feed forward control trainer is an integrated table top model to establish closed loop systems via feed forward and feedback control. This trainer contains necessary components for establishing the effect of adding feed forward control on a conventional feedback heating system. From this we can study feedback control alone & feedback system supported by feed forward action. During operating the system in feedback configuration, we can study P, PI, PD, PID controllers for process and heating process independently so that the two different systems can be done simultaneously. Connection sockets are provided to enable the study of the characteristics of transmitters and final control elements which are used in the trainer.

TECHNICAL SPECIFICATION

- The Feed forward control trainer will be an integrated table top model to establish closed loop control systems viz., feed forward and feedback control. The trainer will contain necessary components for establishing the effect of adding feed heating system. The system will enable the study
 - a) Feedback control alone and
 - b) Feedback system supported by feed forward action

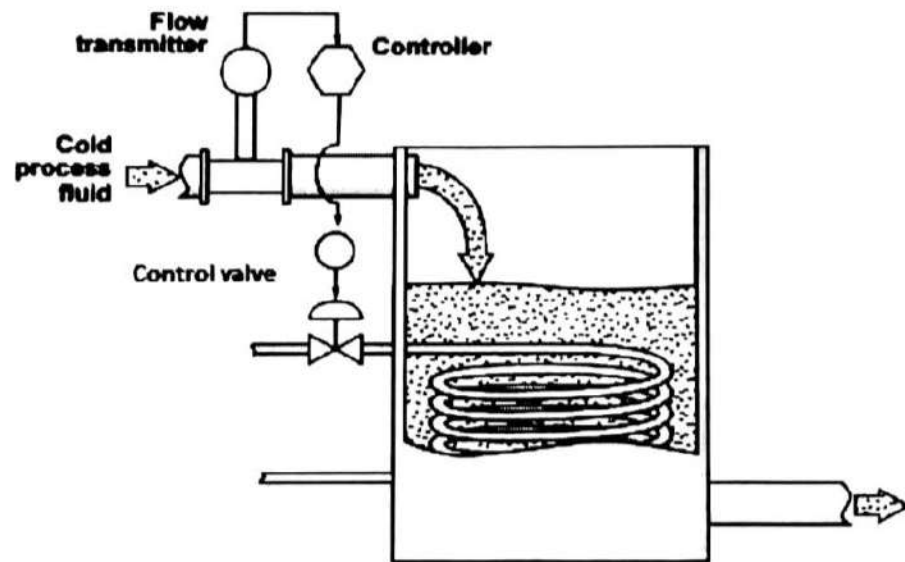


Fig. 2.1 Feed –Forward control system

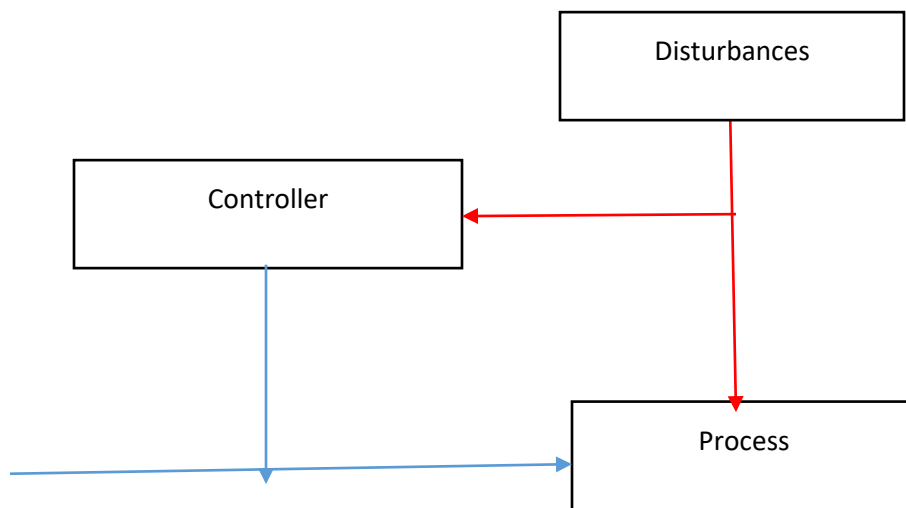


Fig.2.2 Block diagram of feed-forward system

- While operating the system in feedback configuration, study of P, PI, PD and PID controllers are to be made possible on Flow process and Heating Process independently so that two different experiments can be done with the system simultaneously.
- Connection sockets will be provided to enable the study of the characteristics of transmitters and final control elements etc. used in the trainer.
- Turbine type flow transmitter to sense the flow
- Rotameter of (10-100) LPH to visualize the flow rate
- Motorised Control valve to control the flow rate of water.
- A water heater for heating the water circulated through it
- Temperature Transmitter
- Sensor: RTD (PT100)
- Max. Temperature Rating: 1000C
- Output: (4-20) Ma
- Thyristor based Power Driver
- Input: (4-20) mA (Or) (0-5) V
- Output: (0-230V AC/20 Amps
- An overhead tank to give constant water flow to the heater
- Rust proof storage tank of 25-litre capacity
- All the water path guided by rust proof tube.
- One FHP Pump for water circulation
- Flexible front panel wiring facility to study the characteristics.
- PID controllers with communication interface RS232/RS422
- All will be mounted in a fine, powder coated, finished frame/cabinet

FEEDFORWARD CONTROL

Feedforward control measures the disturbances as they enter the process. The configuration uses a controller to adjust the manipulated variable so the effect of disturbances on the controlled variable is reduced or eliminated. Feedforward controllers can calculate the exact amount by which the manipulated variable by which the manipulated variables should change to compensate the disturbances. Feedforward configuration requires an accurate measurement of disturbance, manipulated variable and controlled output.

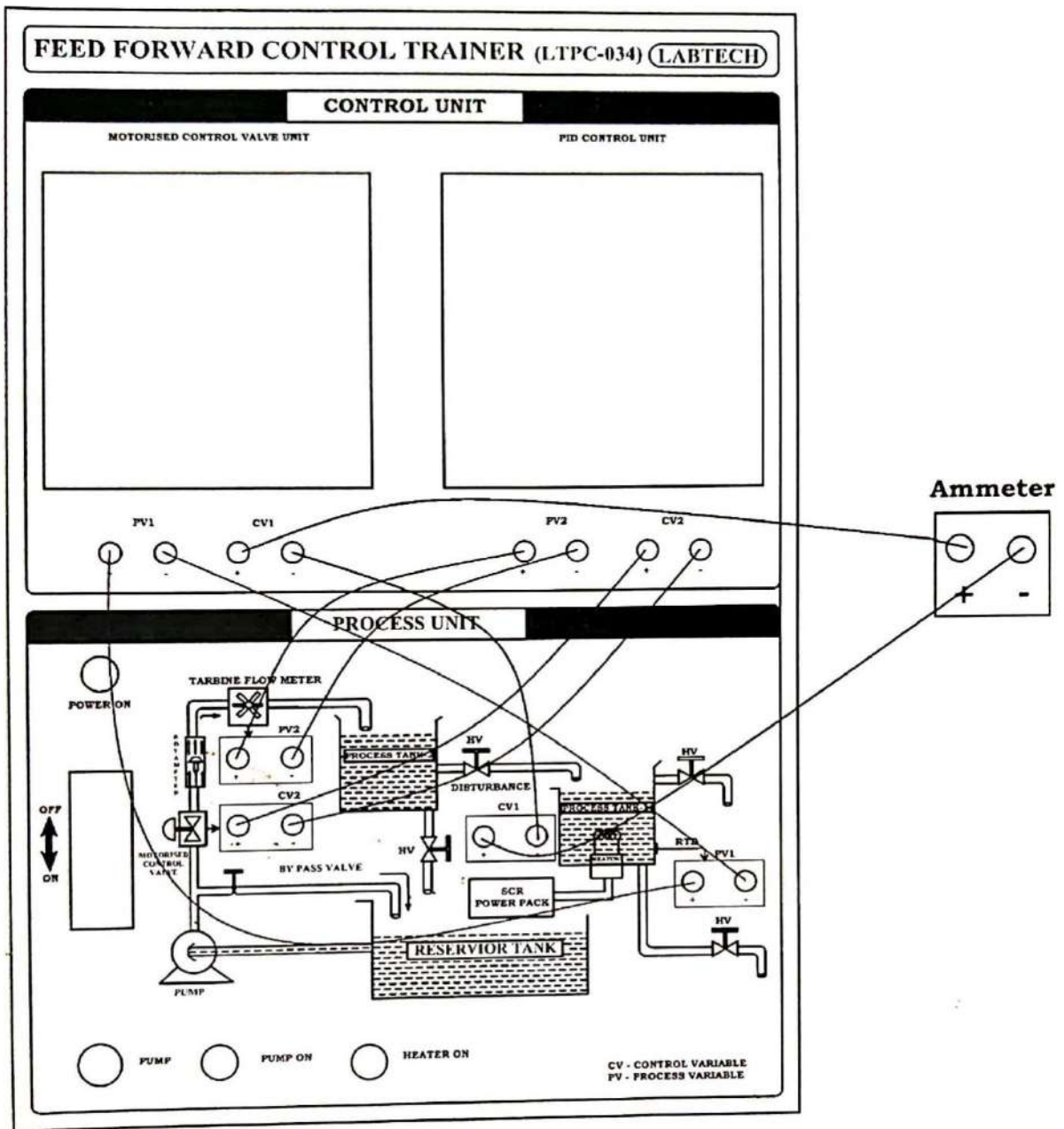


Fig.2.3 wiring diagram of Feedforward Trainer (LTPC-034)

Feed forward control configuration measures the disturbance directly and takes control action to eliminate its impact on the process output. Feedforward controllers have the theoretical potential for perfect control. Different experiments like conventional feedback temperature control, feedback flow control, feedforward and feedback feedforward control can be done in this trainer.

LabTech Feed forward process station consists of water supply tank, pump. Turbine flow meter, rotameter, Electronic control valve, Process tank for creating disturbance, Process tank with heater and Temperature sensor. These units along with necessary piping are mounted on stand-alone type structure. The setup is connected to computer through USB port for monitoring and control by using PID logics.

Procedure:

- Turn on the MCB then list of experiments will come P, I, D window
- Press the function key according to the experiment First press 0 to edit or initialize any data in the process. For feed forward control press 3 to go the experiment window
- Set the temperature and Disturbance flow rate (FC)
- Observe the feed forward controller output through an ammeter connected in series of PV I (Refer patching diagram). Then increase the disturbance flow rate to 40 and observe the increase in output (Feedforward action is taking). Up to 100LPH do the same experiment and observe the controller variations.

Result.

Feedforward control is studied and responses are plotted.

EXPERIMENT NO.3

STUDY OF CASCADE CONTROL

Aim

To design and control the cascade control flow (inner loop) and level (outer loop)

Apparatus

- LabTech multiprocessor trainer
- PC
- Compressor
- Patch cards

Theory

This Cascade control trainer is an integrated table top model to establish closed loop control systems viz. feedback and cascade. The system has provision to study the performance with and without cascade control. There are provisions to operate Flow control and Level control independently in feedback configuration. The system enables the study of P,PL,PD and PID controllers on level and flow when operated independently and in cascade. Connection sockets are provided to enable the study of the characteristics of transmitters, T/P or P/I converters and control valves used in this trainer.

TECHNICAL SPECIFICATION:

- The Cascade control trainer will be an integrated table top model to establish closed loop control systems viz., feedback and cascade.
- Provision will be there to study the performance of the system with and without cascade control.

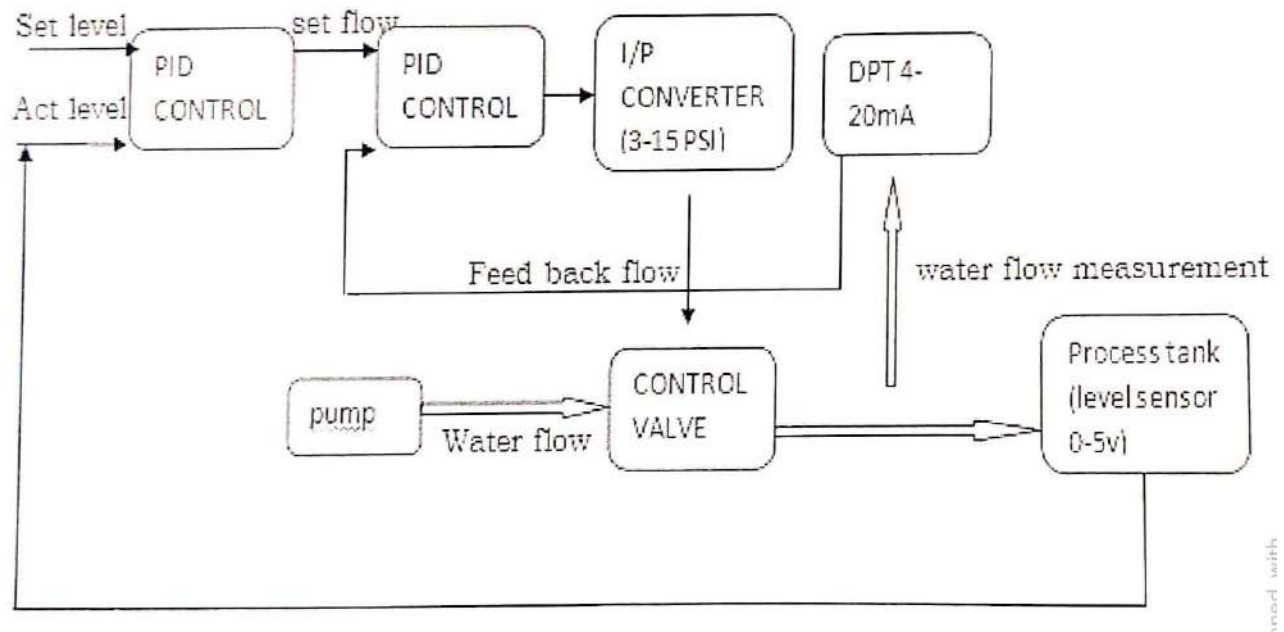


Fig.3.1 Block diagram of cascade control system; Process tank

- Orifice plate with differential pressure transmitter of Range of 0-100 LPH.
- Dual Probe RF capacitance type Level Transmitter to sense the water level up to 250mm.
- Industrial standard electro pneumatic converter to control the control valve of (Input: 4-20mA. Output: 0 - 1.2 Bar).
- Pneumatic Control valve to control the flow rate of water
- One (10-100) LPH turbine type flow indicator with sight glass and pointer.
- Transparent process tank of 5 litre capacity with graduated level scale.
- Rust proof water reservoir of capacity (12 litres) to store water.
- One FHP pump for water circulations.
- AIR Regulator cum filter of capacity (0-2.5 Bar) to regulate the Air pressure from the compressor.
- Bypass line provision in the Reservoir tank to avoid pump loading.
- Hand valve provision in process tank to make the disturbance study.
- Master and slave PID controllers with communication interface RS232/RS422.
- All will be mounted in a fine, powder coated, finished frame/cabinet.
- All the water paths will be guided by rust proof tubes

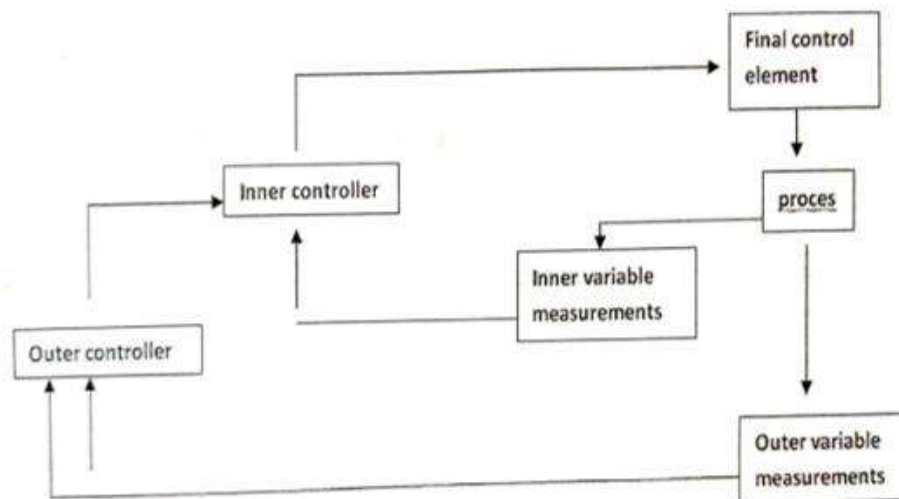


Fig.3.3 General Block diagram of Cascade Control system

OVERVIEW OF CASCADE CONTROL

In Cascade control configuration, we have one manipulated variable and more than one measurement. It is an alternative to consider if direct feedback control using the primary variable is not satisfactory and a secondary variable measurement is available. Cascade control uses the output of primary controller to manipulate the set point of secondary controller.

The basic principle of cascade control is that if the secondary variable responds to the disturbance sooner than the primary variable then there is a possibility to capture and nullify the effect of the disturbance before it propagates into the primary variable. The concepts of cascade control are illustrated in Fig 11.3.

The two measurements are taken from the system and used in the irrespective control loops. In the outer loop, the control output is the set point of the inner loop. The outer loop is called primary loop and inner loop is called secondary loop. Thus, if the outer loop dynamic variable changes, the error signal affects a change in set point of the inner loop, even though the measured value of the inner loop has not changed, the inner loop experience an error signal and produces a new output by virtue of the set point change. Cascade control generally provides better control of the outer loop variable than is accomplished through a single variable system. The primary objective of cascade control is to divide an otherwise difficult control process into two portions, whereby a Secondary control loop is formed around major disturbance, leaving only minor disturbance to be controlled by the primary controller.

FEATURES OF CASCADE CONTROL:

The outer controller is otherwise termed as primary or master controller. In the same way, the inner controller is otherwise termed as secondary or slave controller.

The main features of cascade control are:

- More than one measurement, but one manipulated variable
- The output of master controller serves as set point for the slave controller.
- Two feedback loops are nested together
- Especially useful in eliminating effects of disturbance that move through the system very slowly.

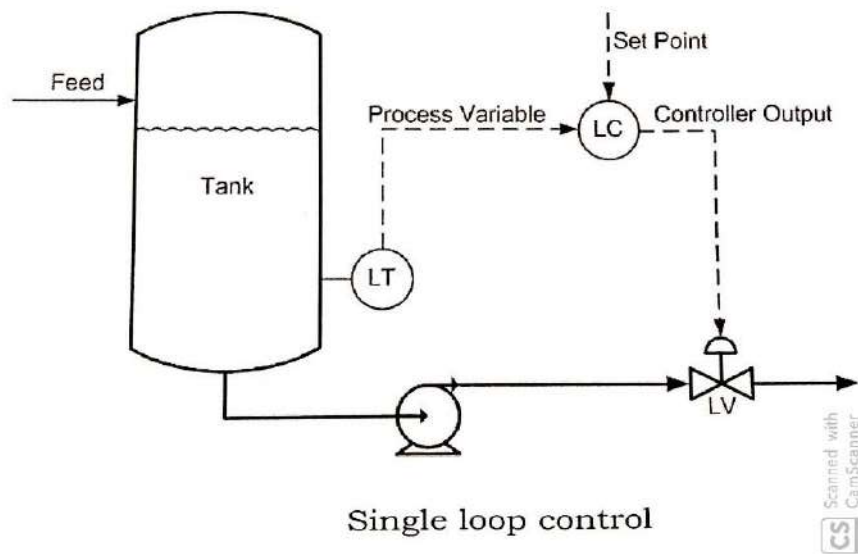


Fig.3.4 Single loop control (Conventional feedback control)

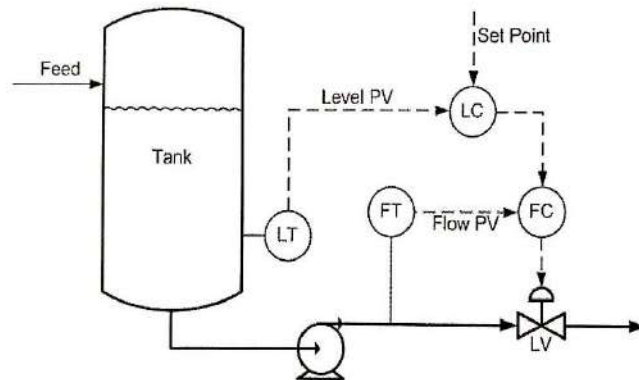


Fig.3.5 Cascade control

- Inner loop has the effect of reducing time lag in outer loop, so that cascade control responds very quickly and improves dynamic performance.
- Decreases variation in primary variable.
- Enhance stability characteristics
- Insensitive to modelling errors
- The secondary controller corrects the disturbances arising within the secondary loop before they affects the primary variables.
- The secondary controller is mostly proportional with high gain value. Tuning of this single proportional mode is simple, and moreover the offset associated with proportional mode can be easily removed by integral action present in primary controller.

CASCADED CONTROL [LEVEL AND FLOW]

In single-loop control, the controller set point is set by an operator, and its output drives a final control element. For example: a level controller driving a control valve to keep the level at its set point as in Fig11.4. In a cascade control arrangement Fig11.5, there are two (or more) controllers of which one controller's output drives the set point of another controller. For example: a level controller driving the set point of a flow controller to keep the level at its set point. The flow controller, in turn, drives a control valve to match the flow with the set point the level controller is requesting. The controller driving the set point (the level controller in example above) is called the primary, outer, or master controller. The controller receiving the set point (flow controller in the example) is called the secondary, inner or slave controller.

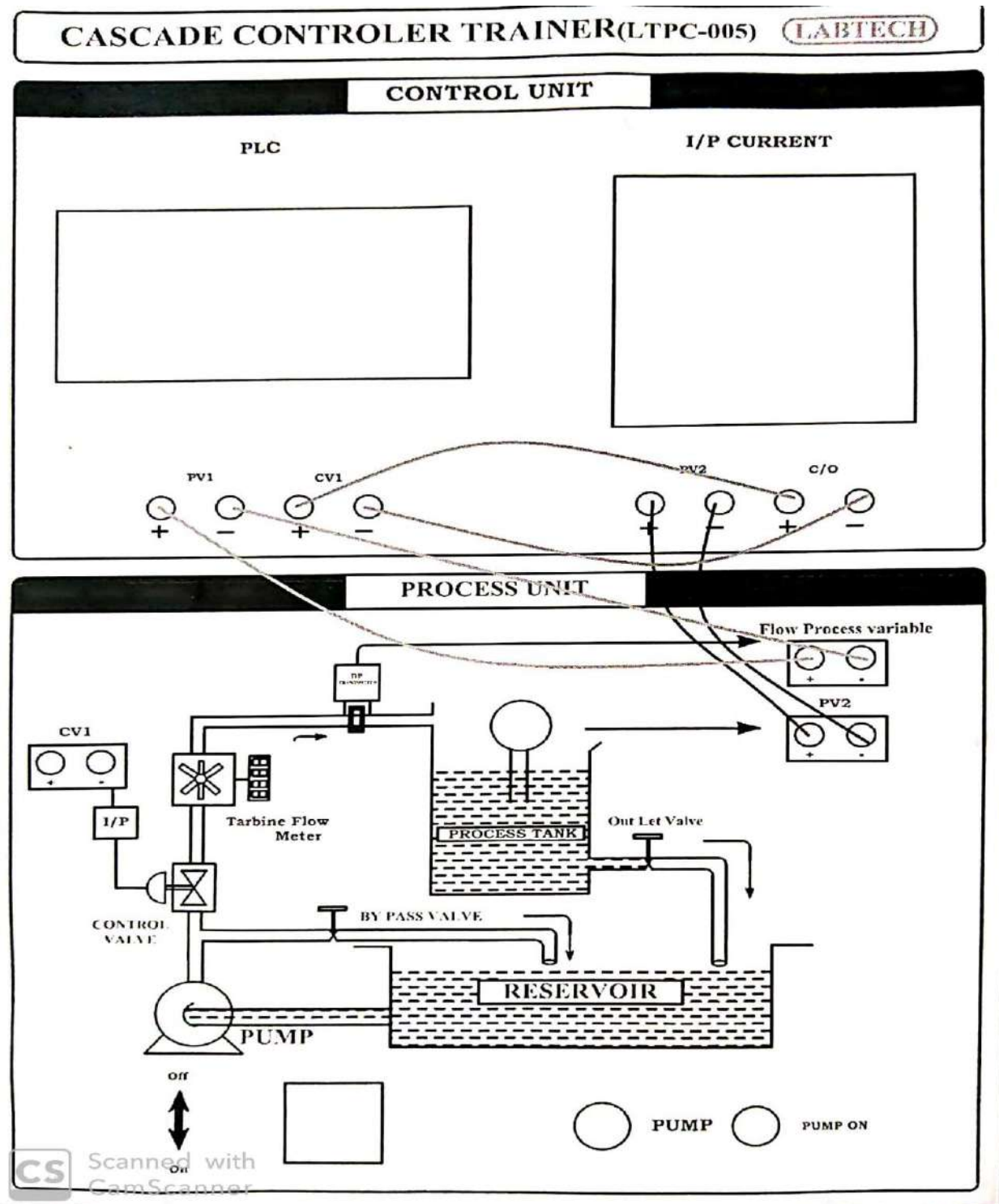


Fig.3.6 Wiring diagram of Cascade Controller Trainer (LTPC-005)

Procedure

- To fill the water in LabTech multi process station reservoir tank
- To connect the LabTech multi process station 230V ac supply >>Switch on the trainer
- Air compressor to connect the air regulator, air regulator regulates the air to set out put air 20PSI
- Patching connections are given as per diagram
- Press F₂ to go to the next window. Set the PID value for tuning level and achieve the set value digit. Press SET button to edit it and press F₁ to move to the next digit.
- Press F₂ to move to next window. In this window cascade now P.I.D values will come which is not required for this experiment so again press F₂ to move to the next experimental window.
- In this experimental window we have to define the upper and the lower limit of the level.
- Press F₂ to move next window i.e. PID values for flow tuning which is not required for the current experiment so again press F₂ to go the next window which is for defining the flow upper limit and lower limit which is also not required for the current experiment so again press F₂ to go the initial experimental window.
- In this experimental window PID values will come for tuning the flow rate of the fluid. Press SET button to edit it and press F₁ to move to the next digit. Then press F₂ to move the next experimental window.
- In this experimental window we have to enter the value of Flow upper limit and flow lower limit, again press F₂ to go the initial experimental window. For tuning the proper PID value the process will achieve the flow rate which will equal to the set point flow rate.

Result

Thus, Cascade control was done using P, Pi, PD and PID controllers

EXPERIMENT NO. 4

STUDY OF RATIO CONTROL

Aim

- To design and control multiplier technique in ratio control.
- To design and control Divide technique in ratio control.

Apparatus

- LabTech ratio control trainer
- PC
- Compressor
- Patch cards

Theory

This ratio control trainer is an integrated top model to establish closed loop control systems via feedback and ratio control. The system has built such a way that the ratio of flow rates between two liquid streams to be maintained at the desired value. Out of the two liquid streams one has manipulated and the other left as wild while doing ratio control the system is enable the study of ratio control strategies using multiplier and divider techniques. There is also provision for conducting the experiment of simple feedback flow control when needed to control the liquid flow rate through a pipe and enable to study of P, PI, PD, PID controllers. Connections are provided to study the characteristics of transmitters, 1/P or P/I converters and control valves used in the trainer.

TECHNICAL SPECIFICATION

- The Ratio control trainer will be an integrated table top model to establish closed loop control systems viz. feedback and ratio control.
- The system will be built in such a way that the ratio of flow rates between two liquid streams to be maintained at the desired value.

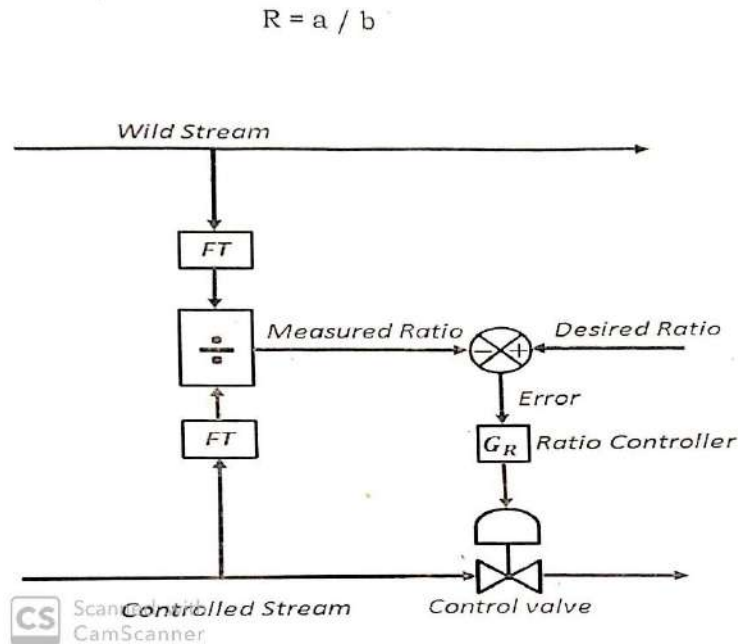


Fig.4.1 Ratio Control system

- Out of the two liquid streams, one will be manipulated and the other will be left as wild while doing ratio control.
- The system will enable the study of ratio control strategies using multiplier and divider techniques.
- There will be provision for conducting the experiment of simple feedback flow control when needed to control the liquid flow rate through a pipe and to enable the study of P, Pi, PD and PID controllers.
- Connection sockets will be provided to enable the study of the characteristics of transmitters, I/P, P/I converters and control valves etc. used in the trainer.
- Orifice plate with differential pressure transmitter of Range of 0-100 LPH. 2 Nos. for sensing the flow rates of wild stream and manipulated stream.
- Rust proof water reservoir of capacity (12 liters) to store water.
- Air regulator cum filter of Range (0-2.1) Bar to regulate the pressure to feed the input for Electro Pneumatic Converter.
- Industrial standard electro pneumatic converter to control the control valve of [Input (4-20mA, Output (0.2-1) Bar]
- Pneumatic Control Valve to control the liquid flow.
- Two Nos. of (10-100) LPH turbine type flow indicators with sight glass and pointer for monitoring wild and manipulated streams

- One FHP pump for water circulations.
- Bypass line provision in the Reservoir tank to avoid pump loading.
- Hand valve provision in wild stream to make the disturbance study.
- Flexible front panel wiring facility to study the characteristics.
- Lab VIEW based with communication interface RS232.
- All will be mounted in a fine, powder coated, finished frame/cabinet.

OVERVIEW OF RATIO CONTROL

Ratio control is used to ensure that two more flows are kept at a constant ratio even if the flows are changing. Ratio control is a special type of feed forward control where two disturbances are measured and held at constant ratio with each other. Out of two stream/flow one can be controlled and measured but the stream whose flow rate is uncontrollable is called as wild stream. The common applications of ratio control are

- a. To keep a constant ratio between the feed flow rate and the steam in the reboiler of a distillation column.
- b. To hold the reflux ratio constant in a distillation column.
- c. To control the ratio of two reactants entering a reactor at a desired ratio.
- d. To hold the ratio of two blended streams constant, in order to maintain the composition of the blend at a desired value.
- e. To maintain correct air and fuel mixture for optimal combustion.

Suppose in a process in where an acid must be diluted with water in the proportion two parts water to one-part acid. If a tank has an acid supply on one side of a mixing vessel and a water supply on the other, a control system could be developed to control the ratio of acid to water, even though the water supply itself may not be controlled. This type of control system is called ratio control. Ratio control is used in many applications and involves a controller that receives input from a flow measurement device on the unregulated (wild) flow. The controller performs a ratio calculation and signals the appropriate set point to another controller that sets the flow of the second fluid so that the proper proportion of the second fluid can be added.

The objective of a ratio control scheme is to keep the ratio of two variables at a specified value. Thus, the ratio (n) of two variables (a and b),

$$R = a/b$$

Ratio Control is the most elementary form of feed forward control. These control systems are almost exclusively applied to flow rate controls. There are many common usages of ratio controls in the context of chemical engineering. They are frequently used to control the flows on chemical reactors. In these cases, they keep the ratio of reactants entering a reaction vessel in correct proportions in order to keep the reaction conditions ideal. They are also frequently used for large scale dilutions.

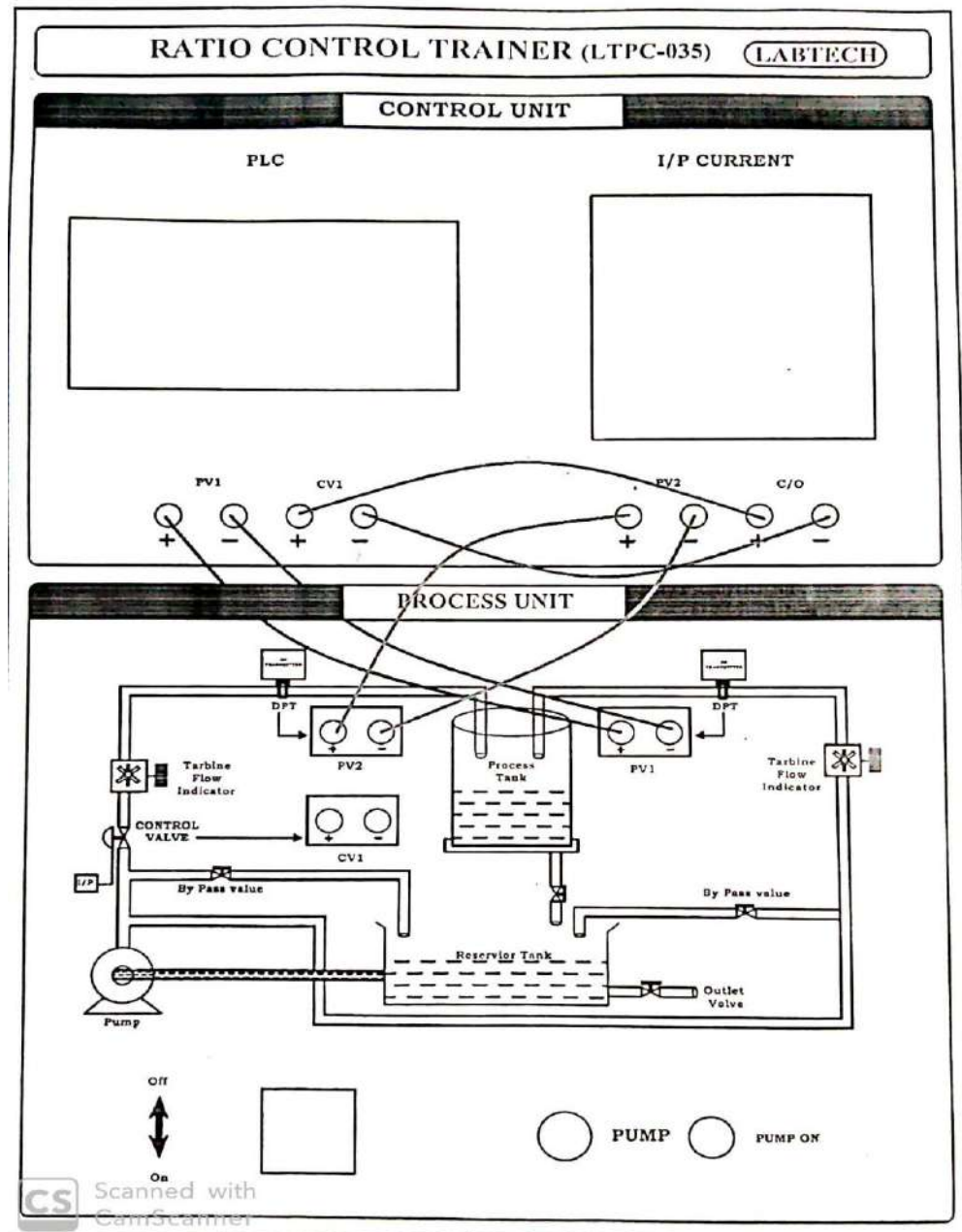


Fig.4. 2Wiring diagram of Ratio Controller Trainer (LTPC-035)

Procedure

- Switch ON the system.
- Set the system in DIV technique mode (10)
- Set the division factor manually, with help of F3 button to navigate through the menu, SET button to change the values and up/down button to increment/decrement, Press ENT to enter.
- Switch ON both the pumps.
- Allow the system to settle
- Provide a disturbance by means of bypass valve in the wild stream path.
- Note the manipulated stream flow rate values at particular time intervals till the system settles down and plot it.
- Set the bypass valve of wild stream back to initial position.
- Set the system in MUL mode (01).
- Set the multiplication factor.
- Repeat steps 4-7.
- Switch OFF the pumps and the system.

Result

Thus, Ratio control was studied is implemented using MULTIPLIER and DIVIDE techniques and responses are plotted.

EXPERIMENT NO. 5

DATA LOGGER: SPEED MONITORING SYSTEM FOR DC MOTOR

Aim

To set up hardware and to develop a LabVIEW VI for monitoring the speed of the DC MOTOR.

Apparatus

- DC-Motor, shaft, disc with perforations.
- IR LED, Phototransistor, resistors, voltage sources.
- DAQ card: NI6024E
- LabView installed in PC.

Theory

INTRODUCTION TO LABVIEW

LabVIEW stands for Laboratory Virtual Instrumentation Engineering Workbench. LabVIEW is a platform and development environment for a visual programming language from National Instruments. The graphical language is named "G" Originally released for the Apple Macintosh in 1986 the latest version of LabVIEW is version 86, released in August of 2008. LabVIEW is a graphical programming environment developed by National Instruments (NI) which allows high level or system-level designs. It uses a graphical programming language to create so-called Virtual Instruments (VI) blocks in an intuitive flowchart-like manner. A design is achieved by integrating different components or subsystems within a graphical framework. LabVIEW provide data acquisition, analysis, and visualization features well suited for DSP system-level design. LabVIEW constitutes a graphical programming language that allows one to design and analyze a DSP system in a shorter time as compared to text-based programming environments.

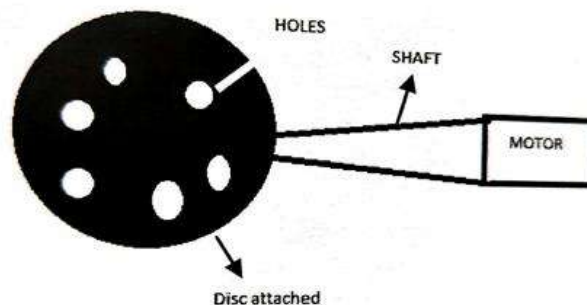


Fig. 5.1 DC motor with shaft and disc assembly

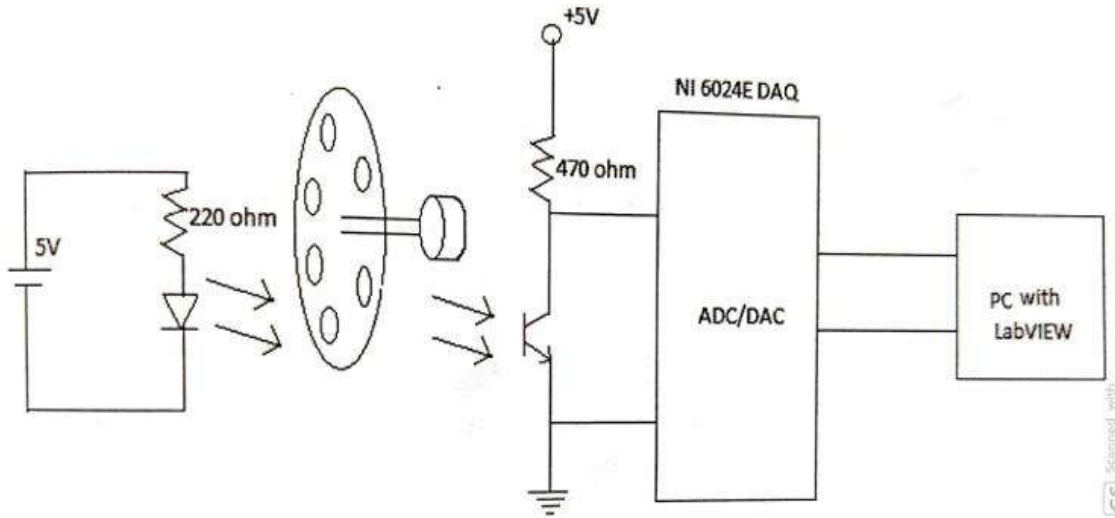


Fig. 5.2 Circuit diagram for data logger

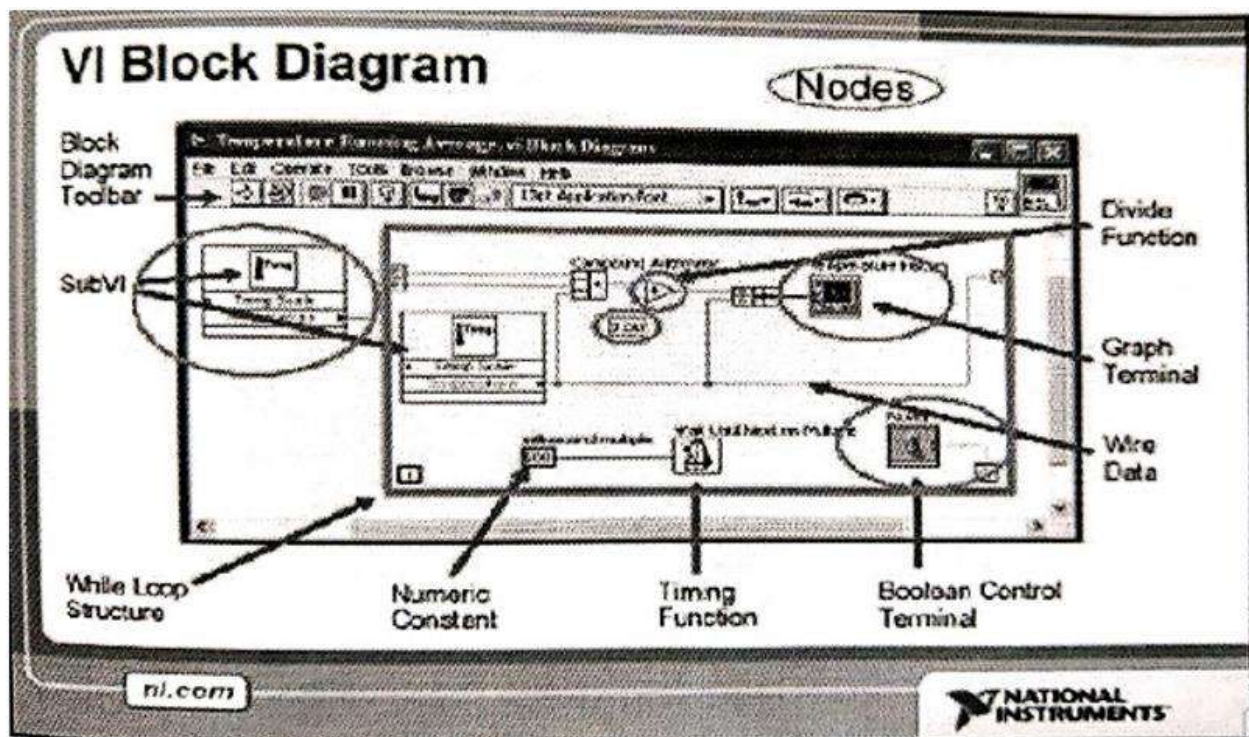


Fig. 5.3 LabVIEW Block Diagram

VIRTUAL INSTRUMENTS

LabVIEW graphical programs are called Virtual Instruments (VIs). VIs run based on the concept of data flow programming. This means that execution of a block or a graphical component is dependent on the flow of data, or more specifically a block executes when data is made available at all of its inputs. They are analogous to functions from conventional language programs. VIs have both an interactive user interface and a source code equivalent, and accept parameters from higher-level VIs. The following are descriptions of these three VI features.

1. VIs contain an interactive user interface, which is called the front panel, because it simulates the panel of a physical instrument. The front panel can contain knobs, push buttons, graphs, and other controls and indicators. You input data using a keyboard and mouse, and then view the results on the computer screen.
2. VIs receive instructions from a block diagram, which you construct in G. The block diagram supplies a pictorial solution to a programming problem. The block diagram contains the source code for the VI.
3. VIs use a hierarchical and modular structure. We can use them as top level programs, or as subprograms within other programs or subprograms. A VI within another VI is called a sub VI. The icon and connector pane of a VI work like a graphical parameter list so that other VIs can pass data to it as a sub VI.

BLOCK DIAGRAM [BD]:

The block diagram represents a complete LabVIEW application, and is an example of how intricate LabVIEW programming can be. A BD contains terminal icons, nodes, wires, and structures. Terminal icons correspond to controls or indicators that appear on an FP. Whenever a control or indicator is placed on an FP, a terminal icon gets added to the corresponding BD. A node represents an object which has input and/or output connectors and performs a certain function. Sub VIs and functions are examples of nodes. Wires establish the flow of data in a BD. Structures such as repetitions or conditional executions are used to control the flow of a program.

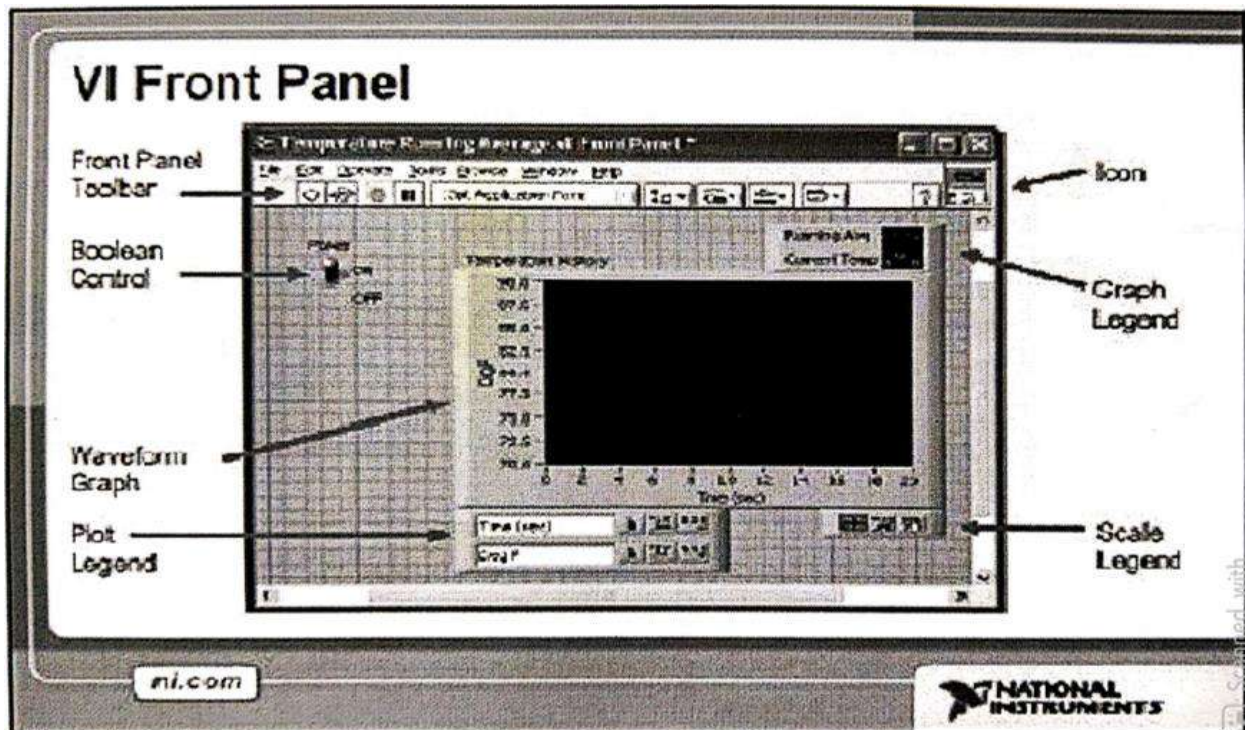


Fig. 5.4 LabVIEW Front Panel

FRONT PANEL [FP]:

An FP contains the user interfaces of a VI shown in a BD. Inputs to a VI are represented by so-called controls. Knobs, push buttons and dials are a few examples of controls. Outputs from a VI are represented by so-called indicators. Graphs, LEDs (light indicators) and meters are a few examples of indicators. As VI runs, its FP provides a display or user interface of controls (inputs) and indicators (outputs).

TOOLS PALETTE:

LabVIEW uses a floating Tools palette, which can be used to edit and debug VIs.

FUNCTION PALETTE:

The Functions palette consists of a graphical, floating palette that automatically opens when switch to the block diagram. Use this palette to place nodes (constants, indicators, Vis, and so on) on the block diagram of a VL. Each top-level icon contains sub palettes.

CONTROL PALETTE:

The Controls palette consists of a graphical, floating palette that automatically opens when we launch LabVIEW. Use this palette to place controls and indicators on the front panel of a VI.

ICON AND CONNECTOR:

A VI icon is a graphical representation of a VI. It appears in the top right corner of a BD or an FP window. When a VI is inserted in a BD as a sub VI, its icon gets displayed. A connector pane defines inputs (controls) and outputs (indicators) of a VI. The number of inputs and outputs can be changed by using different connector pane patterns. A VI icon is shown at the top right corner of the BD and its corresponding connector pane having two inputs and one output is shown at the top right corner of the FP.



Fig. 5.5 Tools Palette

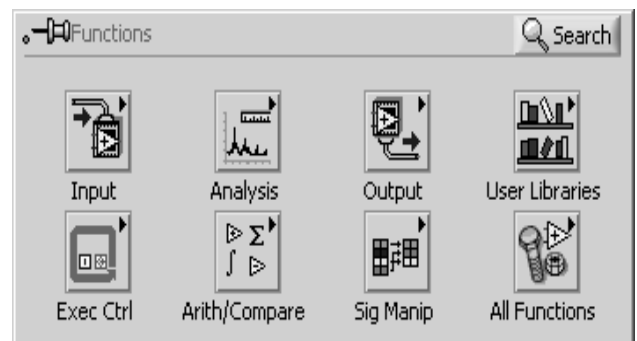


Fig. 5. 6 Function Palette

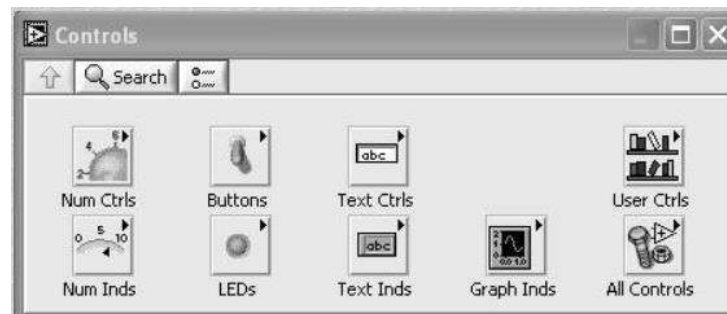


Fig. 5. 7Control Palette

DAQ ASSISTANT:

- A graphical way to configure channels and measurement tasks for your device for use in LabVIEW, Lab windows/CVI, and Measurement
- Studio. You also can use the DAQ Assistant to generate NI-DAQMX code based on your channels and tasks for use in NI application software programs.
- Increased performance, including faster single-point analog I/O and multi-threading
- Simpler, more intuitive APIs for creating DAQ applications using fewer functions and VIs than earlier versions of NI-DAQ.
- Expanded functionality in the NI-DAQmx API for LabVIEW, including property nodes for data acquisition and improved waveform data type support for analog I/O.
- Similar APIs and functionality for ANSI C, Lab Windows/CVI, and Measurement Studio, including native NET and C++ interfaces.

Procedure

- For sensing the speed of the motor, a disc with perforations has to be attached to the shaft.
- At one side of the disc an IR LED is kept ON and at the other side a phototransistor is shown in Fig 13.2
- As the motor rotates a train of pulses will be produced at the collector of the photo transistor. It can be connected to any one analog input channels of the DAQ card (NI 6024E).

Develop a VI in LabVIEW for data acquisition. There is an express VI called DAQ assistant for data acquisition. Just set the channel number, type of the data etc. and we can acquire data very easily. A Lab VIEW program (VI) has two parts. A front panel and a block diagram. Front panel is the GUI (Graphical User Interface) where we will place buttons, -indicators, control switches, charts, XY plots etc. for handling data. In the block diagram we will be specifying the things to be done using a graphical program.

- For speed monitoring first we have to acquire the train of pulses from the collector of the phototransistor. Process the data using the tools available and convert the data into rpm. The rpm data can be displayed numerically in a text indicator or in a meter indicator display.

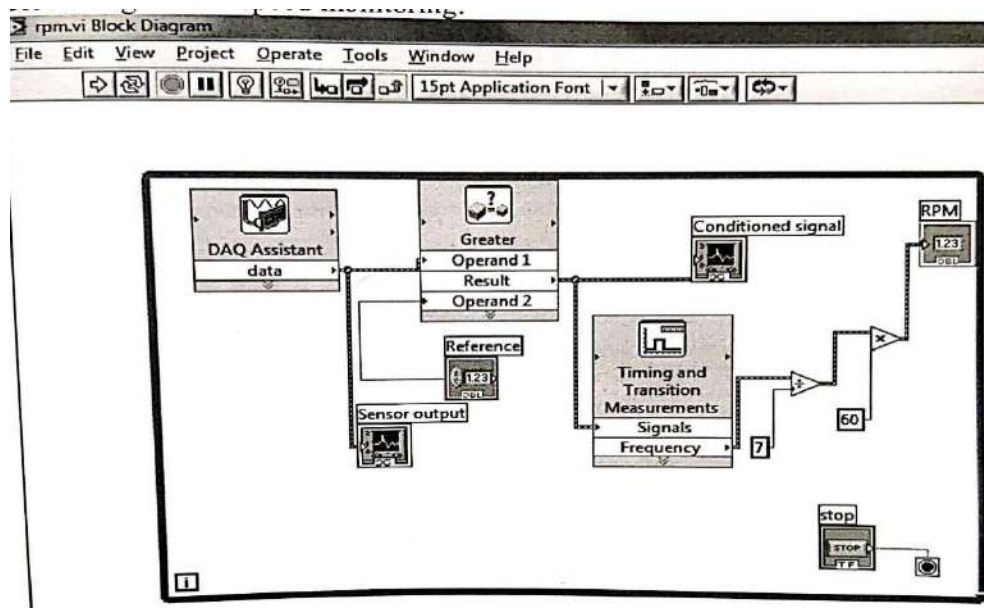


Fig. 5.8 LabVIEW G program for data logger

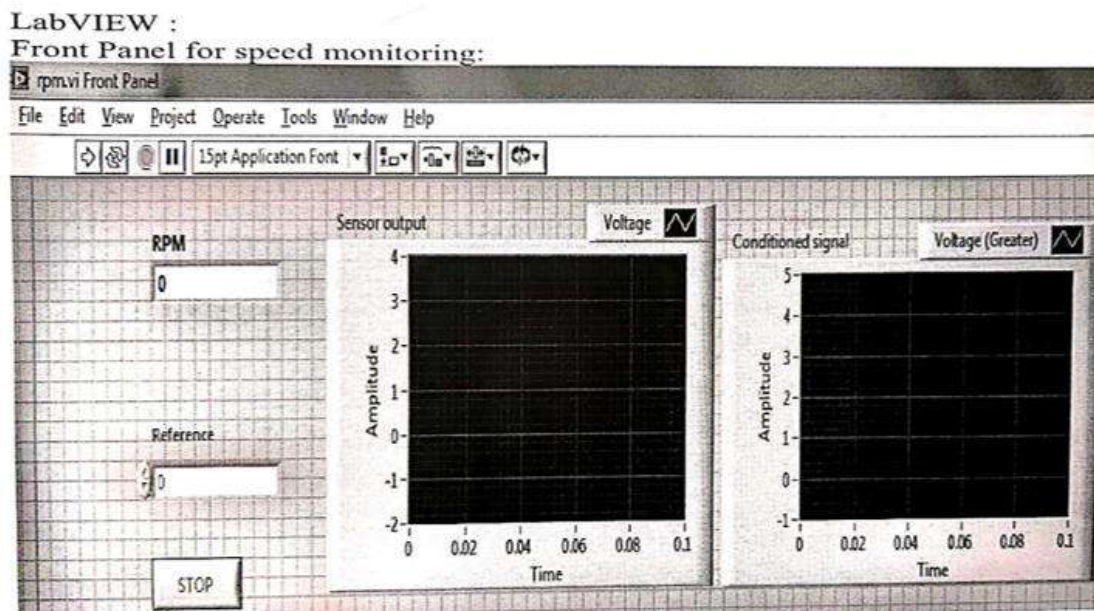


Fig. 5.9 LabVIEW Front Panel for data logger

Result

Hardware for monitoring the speed of the DC MOTOR is set up and developed a LabVIEW VI for data logging via NI DAQ card (NI6024E).

EXPERIMENT NO. 6

STUDY OF PLC-LADDER DIAGRAM AND IMPLEMENTATION OF LOGIC CIRCUITS

Aim

To familiarize with VersaPro simulation software and implement logic gates AND, OR and XOR gates using ladder logic.

Principle

Versa Pro is the name of ladder or ladder Basic programming software for GE Fanuc Versa max microcontroller series of PLC. Software incorporates all the components required to implement a ladder logic program completely. The desired components can be included in the program by selecting and labeling at approximately. Alternately components can be added creating an input table and then selecting the desired components from the input table. Input table has following components, inputs, outputs, counters, relays, timers, custom function.

CONTACT: If a contact is connected to a load and the contact is closed the load will be turned ON. These are normally ON and normally OFF contact with 'x' across body is a normally closed contact.

RELAY COIL: It is a ladder programming equation of the electromagnetic relay.

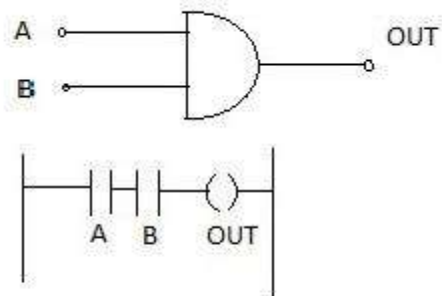
OUT COIL: A PLC output is just an internal delay with a physical connection that can supply electrical power to control on internal load.

TIME COIL: The timers are a special kind of relay when the coil is engaged must wait for a load length of time, before closing its contact during time depends on set values.

IC200UAL005 MICRO PLC

- 23 Discrete Points and 3 Analog Channels
- (13) 24VDC In, (1) 24VDC Out
- (9) Relay Out
- (2) Analog In and (1) Analog Out
- 24VDC Power Supply

AND GATE



6.1 Ladder Logic: AND GATE

OR GATE

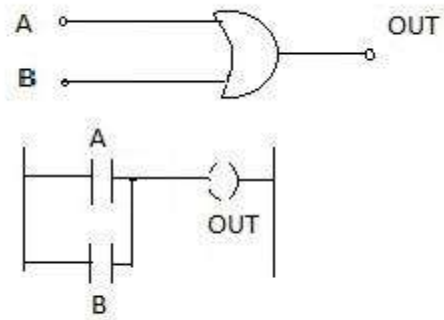
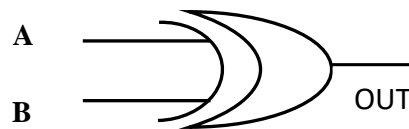


Fig. 6.2 Ladder Logic: OR GATE

Fig.

XOR GATE



A B OUT

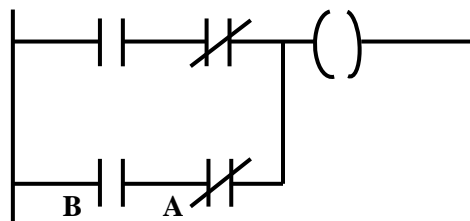


Fig.6.3Ladder Logic: XOR GATE

Procedure

- Start the VERSA PRO software in PC.
- Selecting insert circuit from circuit tab and then select the required components and create the required ladder program.
- Open input table from Edit and define the sequential input, output and delays.
- Select 'insert circuit' from circuit tab and then select the required components.
- Construct the ladder diagram for AND, OR, XOR
- Simulate program by selecting RUN from the simulate table select reset all input stop simulation.
- Save program by selecting 'save' from file for future reference.

Result

Familiarized the logic gates with VERSA PRO software.

EXPEREMENT NO. 7

PLC CONTROLLED BOTTLE FILLING SYSTEM

Aim

To develop a ladder logic program to control the given Bottle Filling System.

Apparatus Required

- GE FANUC's 23-point micro-PLC
- Bottle filling system.
- Versa Pro Software.
- PC.
- RS-232 Cable
- Patch chords

Theory

Let, I_1 = Input from proximity sensor-Bottle indicator.

I_2 = Input from position sensor.

I_3 = Input from low level sensor in process tank

I_4 = Input from high level sensor in process tank

Q_1, Q_2, Q_3, Q_4 = Output for stepper motor coil.

Q_5 = Output for drives a pump.

Q_6 = Output for drives a solenoid valve.

In this system, a pump is used to fill the process tank from the reservoir. The motor switched ON and OFF by two sensors which are placed in the process tank at two levels (LL & HL). When the water level is below the low level, both of the sensor outputs are low (I_3 & I_4). So the motor is switched ON and water is poured into the process tank from the reservoir. When the water level reaches the high level sensor (I_4), -motor is turned OFF. I_1, I_2 are high which in turn energize solenoid valve (Q_6) and open for

specified time determined by on-delay timer PV. The next step is rotate stepper motor by energizing the four coils

The starting addresses of the data are stored in a sequence as follows. SNX is to increment the starting address register in array move function. The input to SNX is given through up counter register R₁. The timing is decided with the help of on-delay Timer PV. First register data is given in Q₁ and incremented till Q₄. The rotation of the stepper motor stops when the Q₆ is low as I₁ and I₂ are high. The operation continues in these sequences

Wiring Diagram

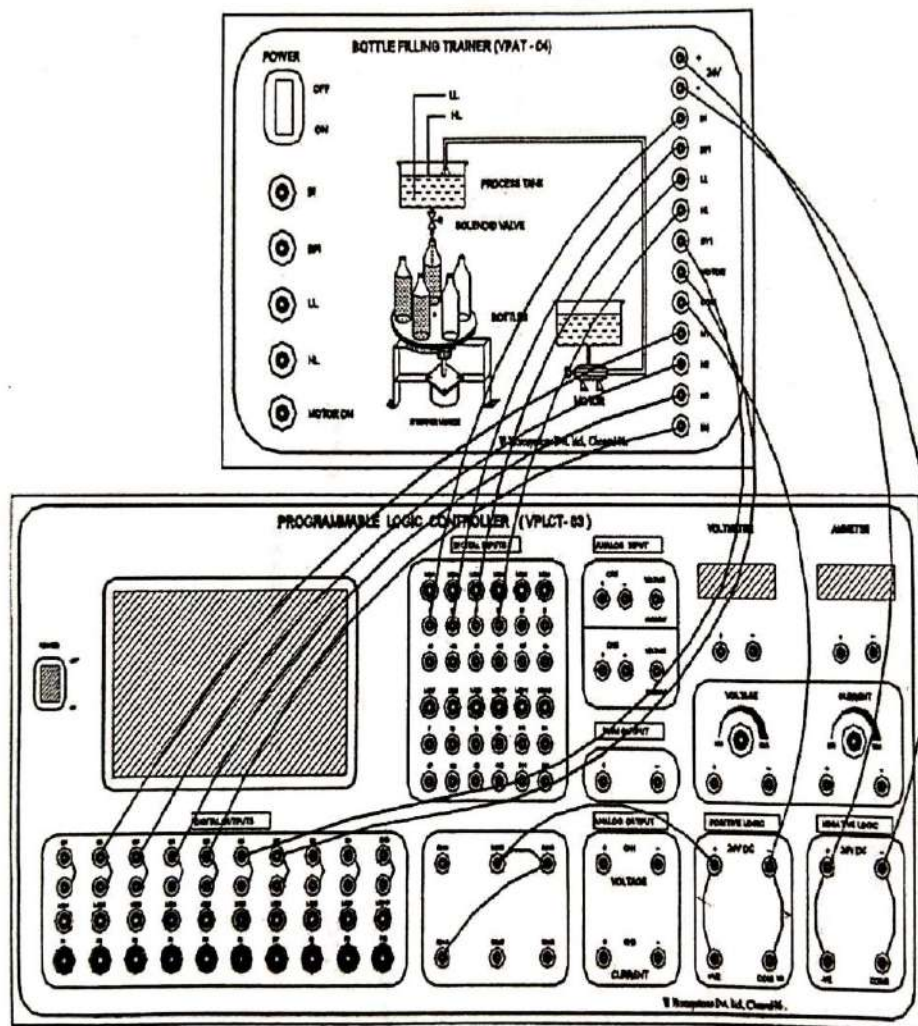


Fig. 7.1 Wiring Diagram

PLC Ladder Logic Program

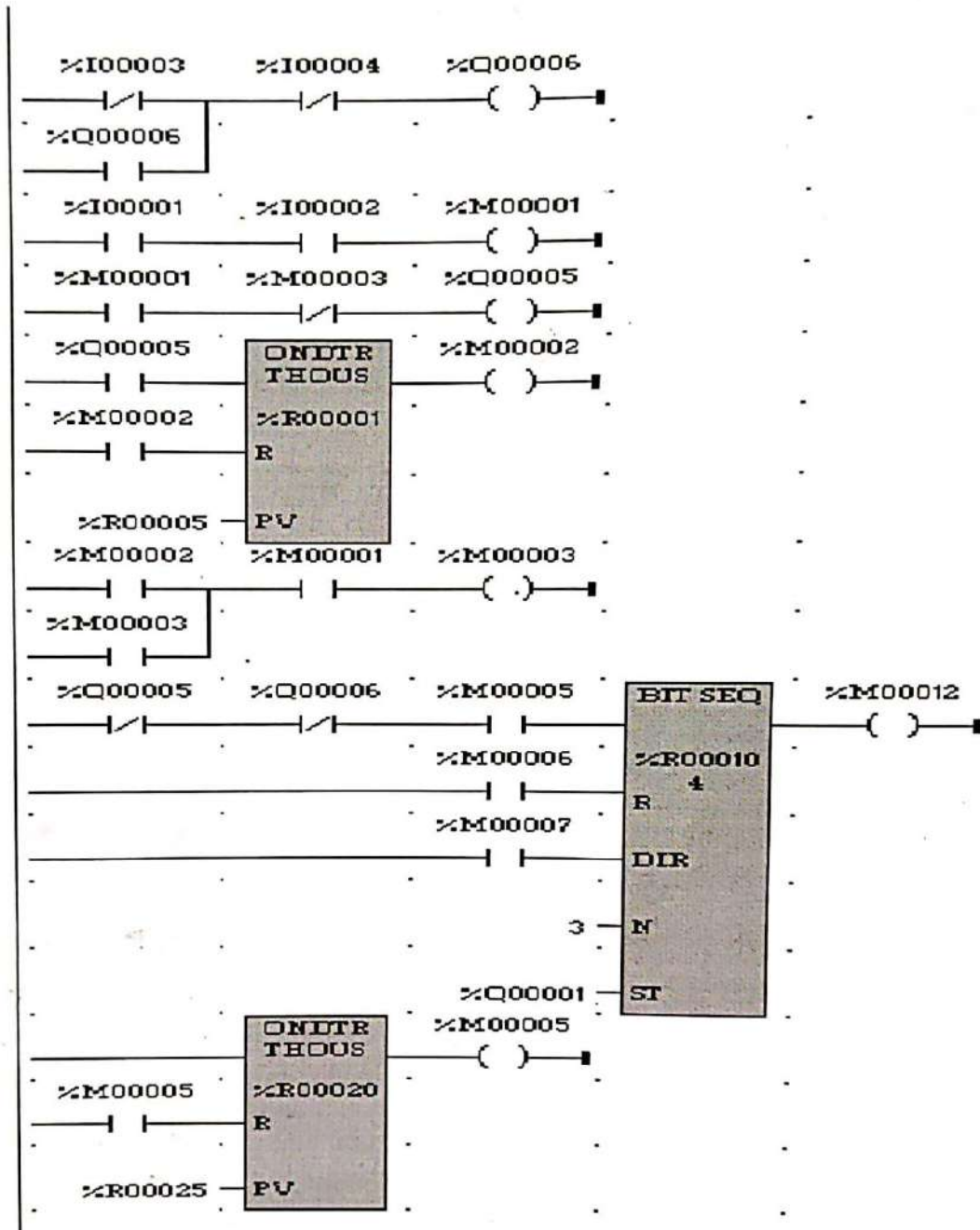


Fig. 7.2 Ladder Logic for Bottle Filling System

Procedure

- Load Versa Pro software to the PC
- Open the Versa Pro software
- Switch ON the PLC trainer and Bottle Filling System
- Connect PLC and bottle filling system kit
- Open the new folder and draw the ladder logic program
- Connect PLC to PC
- Select the correct hardware configuration
- Store the program to PLC
- Run the program
- Verify the performance of the Bottle Filing System

Result

Developed a ladder logic program and studied the performance of PLC controlled bottle Filling System

EXPERIMENT NO. 8

PLC BASED WATER LEVEL CONTROL

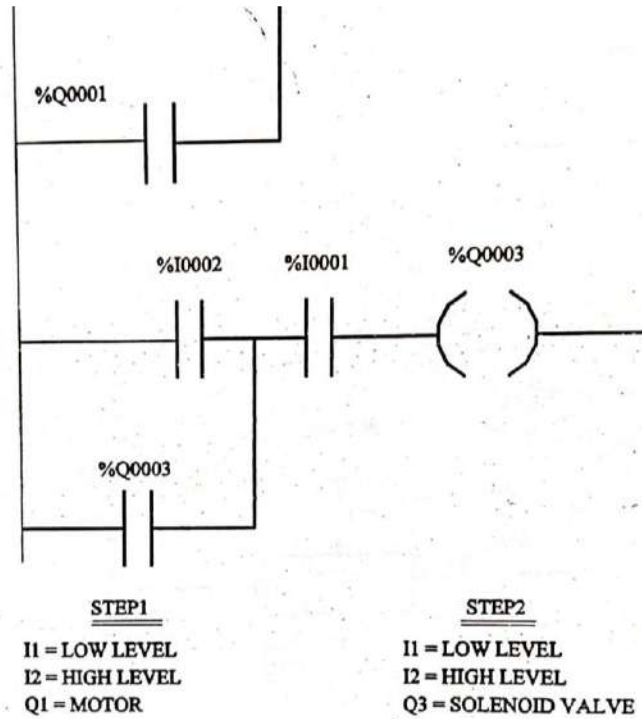
Aim

To develop a ladder diagram for programming the PLC as per the following requirement. When the process tank level goes below lower limit or when empty (HL and LL sensor outputs low), the pump has to be turned ON and valve should be closed. When the process tank level reaches upper limit (LL and HL outputs high), the pump has to be turned off and valve is to be opened.

Apparatus required

- VPLCT-01
- Water Level Controller (VWLC-01).
- PC
- RS232 Cable
- Patch Chords

ON/OFF CONTROL SYSTEM LADDER



STEP 1

I1	-	Low Level
I2	-	High Level
Q1	-	Motor

STEP 2

I1	-	Low Level
I2	-	High Level
Q3	-	Solenoid Valve

Fig. 7.1 Ladder Logic for Water Level Control

Design

LL	HL	Valve	Pump
0	0	0	1
1	0	0	1
1	1	1	0
1	0	1	0

Table 8.1

From Table 7.1 it is clear that a sequential circuit is necessary for implementing this operation. While the level is increasing (condition 2) and while the level is decreasing condition 4) the outputs are different for the same inputs.

WATER LEVEL CONTROLLER WIRING DIAGRAM (VPAT 06)

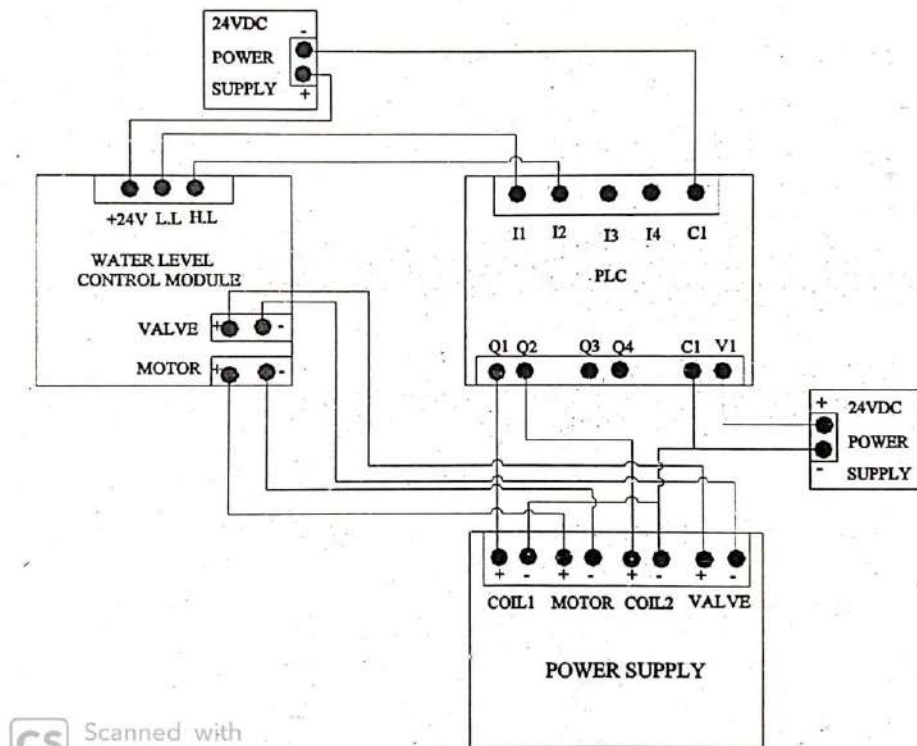


Fig.8.2 Wiring Diagram

The ladder has 2 rungs. The first rung is for controlling the pump and the second for the valve. When the tank is empty pump will be ON as 11 and 12 are specified as normally closed contacts. As the level increases and reaches the lower limit LL, 11 breaks. But we want the pump to be ON till the level reaches HL. Thus in order to keep the pump ON a latching action is given in the ladder by providing a bypass path Q2 across LL. This arrangement will keep the pump ON till the level reaches HL.

Procedure

- Make the connection as shown in wiring diagram.
- Load the Versa Pro software to the PLC
- Open the Versa Pro Software.
- Switch ON the PLC kit and the power supply kit.
- Interface PC to PLC through RS232 cable
- Open the new folder and draw the ladder logic diagram.
- Connect the PLC
- Select the correct hardware configuration (SI. No. IC200NDD101).
- Store the program to PLC
- Run the program.
- To run the program and verify the performance of a water level controller.

Result

Developed the ladder logic diagram and studied the performance of a water level controller