DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

COLLEGE OF ENGINEERING, THIRUVANANTHAPURAM

AE431: CONTROL SYSYTEM AND SIGNAL PROCESSING LAB

LAB MANUAL

2021

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

COLLEGE OF ENGINEERING, THIRUVANANTHAPURAM

CERTIFICATE

This is a controlled document of **Department of Electronics and Communication** 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, Electronics & Communication, College of Engineering, Trivandrum. This is prepared as per **2015 B.Tech** Electronics and Communication scheme.

Prepared by: **Ms. Jeeshma Mary Paul,** **Ms. Kavya Manohar** Research Scholar Department of Electronics and Communication, College of Engineering, Trivandrum	Compiled by: **Dr. Anitha Edison,** **Ms. Narendramudra N G** Assistant Professor Department of Electronics and Communication, College of Engineering, Trivandrum
Reviewed by:	**Approved by:**

Course Code	Course name	L-T-P Credits	Year of Introduction
AE431	CONTROL SYSTEM AND SIGNAL PROCESSING LAB	0-0-3-1	2016

SIGNAL PROCESSING LAB

1. Familiarization of signal processing commands used in MATLAB Software.
2. Developing elementary signal function modules (m-files) for unit impulse, step, exponent and ramp sequence.
3. Generating continuous and discrete time sequences.
4. Carrying out mathematical operations on signals.
5. Response of LTI system described by difference and differential equation.
6. Developing a program for computing inverse Z-Transform.
7. Developing program for finding magnitude & phase response of LTI System 8. Developing program for computing DFT & IDFT.
8. Developing a program for computing circular convolution.
9. Design of filter: FIR, IIR, ECG Signal filter (can be done as 3 separate experiments).

CONTROL SYSTEM LAB using MATLAB

1. Familiarization of MATLAB commands used in control system design
2. Representation of system in MATLAB: state space representation & transfer function representation
3. Stability analysis using Bode plot, root locus & their pole-zero-gain representation.
4. Implementation of PID control using both m-file and Simulink.
5. Pole placement technique applied to stabilize a system.
6. Realization of a compensator design.
7. Modelling and analysis of a first order system.
8. Modelling of an unstable system (inverted pendulum, ball & plate system etc.)

PC Based Control

1. PLC programming: familiarization of instruction set.
2. PLC programming: simulation of process control.

LabVIEW based Virtual Instrumentation

1. Getting started with LabVIEW: Basic operations, controls, indicators, and simple Programming structures.
2. Debugging a VI and sub-VI.
3. Familiarization of DAQ card.

PART C
PC Based Control

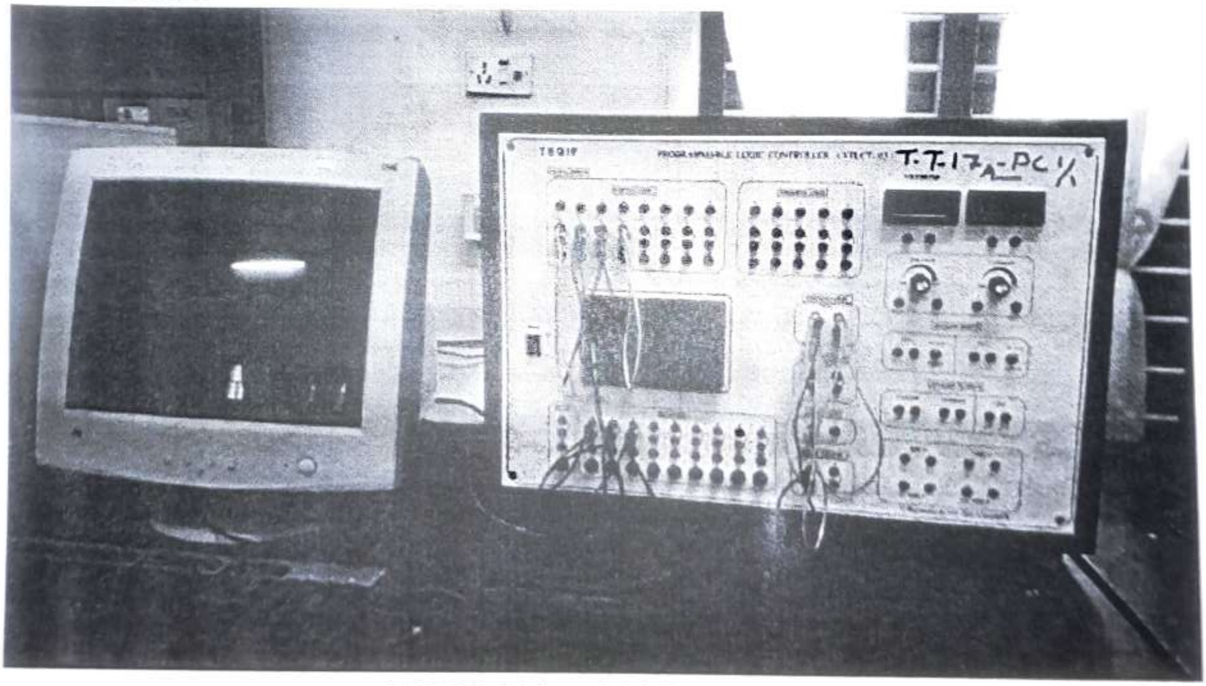
EXPERIMENT 1: FAMILIARIZATION OF PLC INSTRUCTION SET

Aim

Familiarise basic instruction set in ladder programming

Programmable Logic Controller

Programmable logic controller (PLC) is a specialized control computer for industrial and commercial applications. It can be programmed to do a variety of logical functions. It replaces electromechanical relays logic elements with a solid-state digital computer with stored program, which can emulate the interconnection of many relays to perform certain logical tasks. A PLC has many input terminals, through which it interprets 'high' and 'low' logical states from sensors and switches. It also has many output terminals, through which it outputs 'high' and 'low' signals to power lights, solenoids, contactors, small motors and other devices lending themselves to on/off control



VPLCT-03 from Vi Microsystems

Ladder Logic Programming Basics

Ladder logic (also known as ladder diagram or LD) is a programming language used to program a PLC. It is a graphical programming. A ladder logic is drawn between the two lines (power line/Signal line) Each line of code is called a rung, The PLC takes one rung and executes that and then goes to the next line. Each PLC has its own specific IDE to create ladder programme. PLC in the lab is programmed using a Versa Pro.

Basic ladder logic symbols

Symbol	Logic
—| |—	Normally open contact
—|/|—	Normally closed contact
—( )—	Normally open output
—(/)—	Normally closed output

To do

Create ladder programs for basic logic gates like AND, OR, XOR etc and verify output

PART D

LabVIEW based Virtual Instrumentation

EXPERIMENT 1: INTRODUCTION TO LABVIEW

Aim

Familiarise:

a. Block diagram and front panel windows in LabVIEW
b. Controls and indicators in LabVIEW
c. Generate and display sinewave in LabVIEW

LabVIEW

LabVIEW is a graphical programming tool developed by National Instruments. In LabVIEW, we programme by connecting different functional blocks.

LabVIEW programs are called Virtual Instruments (VI). A VI has two parts: a front panel and a block diagram. Front panel is to build user interface. Front panel has options to place controls and indicators, which are the interactive input and output terminals of the VI, respectively. Controls are knobs, push buttons, dials, and other input devices. Indicators are graphs, LEDs, and other displays. Controls simulate instrument input devices and supply data to the block diagram of the VI. Indicators simulate instrument output devices and display data the block diagram acquires or generates. After you build the user interface, you add graphical code using VIs and structures to control the front panel objects. The block diagram contains this graphical source code.

Steps

Familiarize controls and indicators

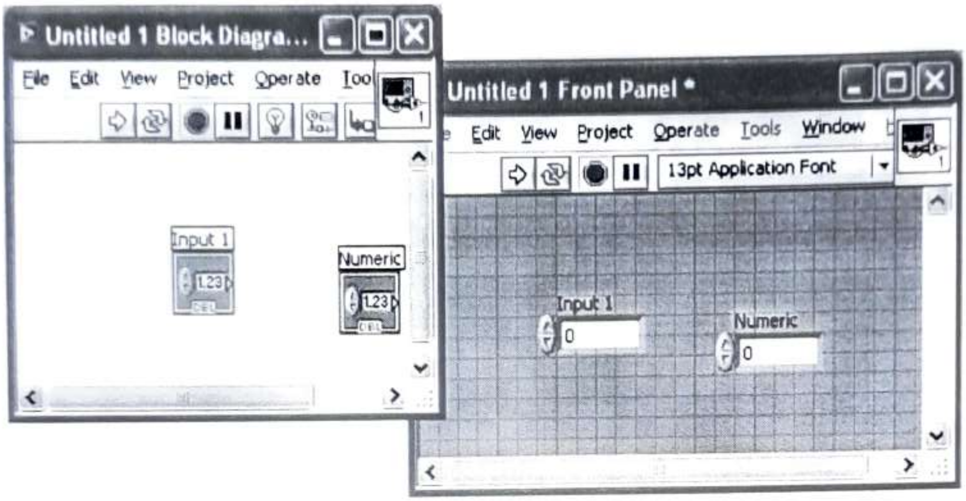
1. Open LabVIEW and create a blank VI. There will be two windows: front panel and block diagram
2. Place some indicators and controls in front panel window
3. Connect the controls and indicators using connecting wires in block diagram window
4. Observe the change in indicators as controls are changed

Generate a sinewave and display the output waveform

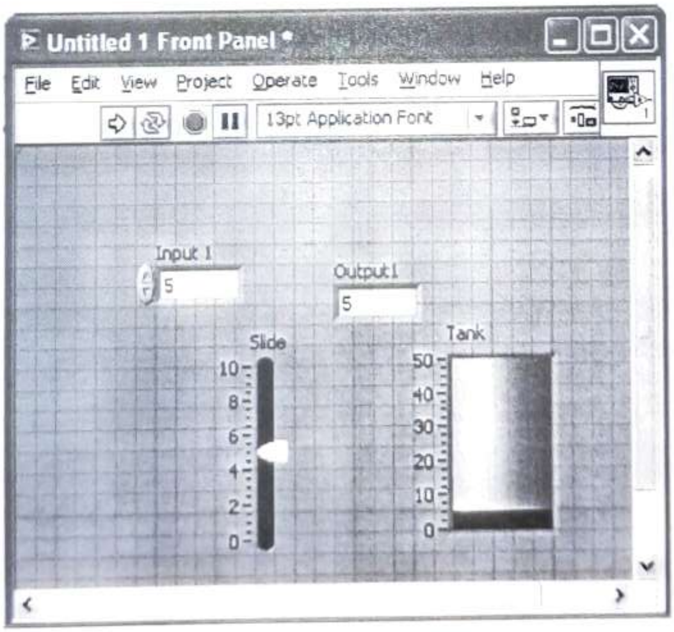
1. In front panel, place waveform graph indicator to display the output
2. Switch to block diagram panel and create waveform generation function. Waveform generation functions are available in signal processing toolbox.
3. Place controls to change the amplitude and frequency of waveform

Expected Output

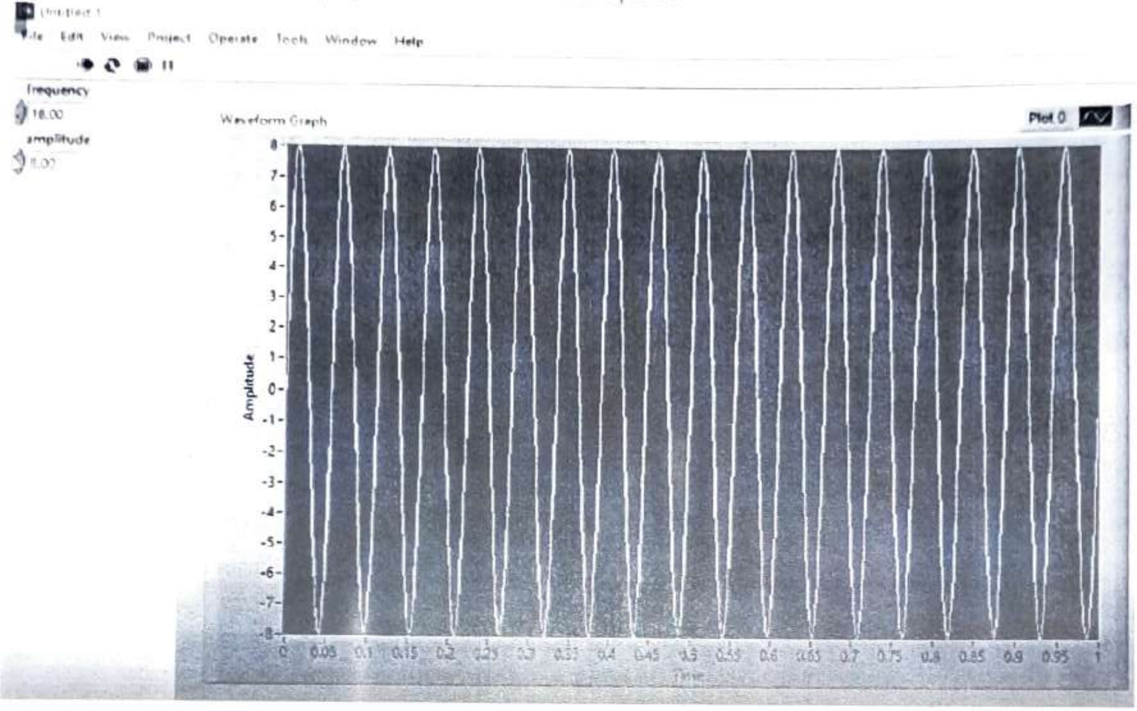
a. Familiarisation of front panel and block diagram panel with numeric indicators and controls


Untitled 1 Block Diagra...
File Edit View Project Operate Tool
Input 1
1.23
Numeric
1.23
Untitled 1 Front Panel *
Edit View Project Operate Tools Window
13pt Application Font
Input 1
0
Numeric
0

b. Front panel with slider control and tank indicator



c. Sine waveform displayed in LabVIEW front panel



frequency 18.00

amplitude 8.00

EXPERIMENT 2: RESPONSE OF A SECOND ORDER SYSTEM USING LABVIEW

Aim

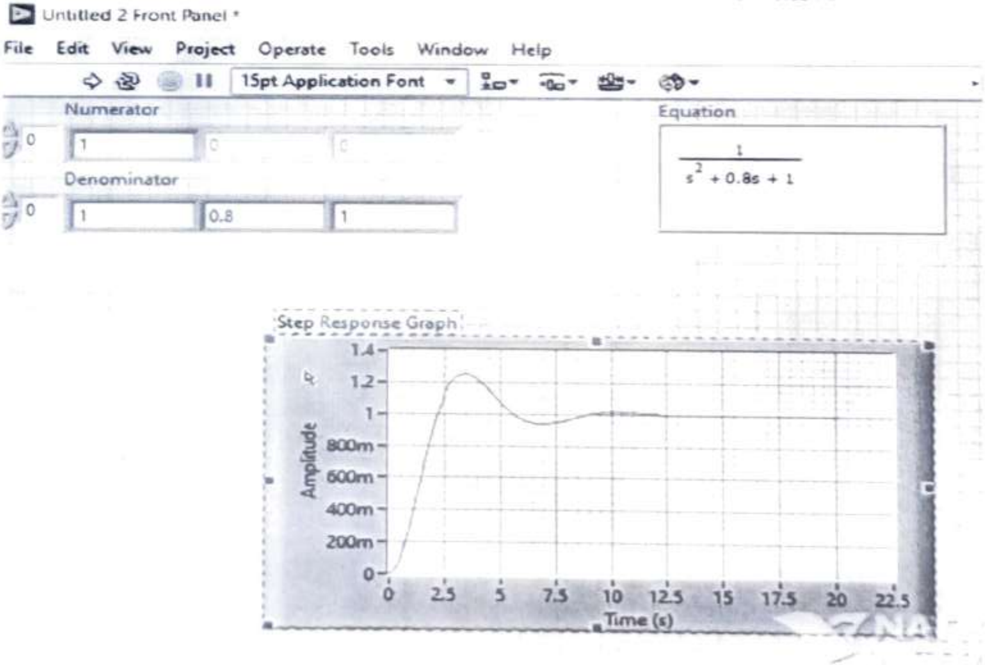
Obtain the step response of a second order system specified by the transfer function $H(s)$

Steps

1. In block diagram panel, create block to generate transfer function (Already available as subVI in control and simulation>>control design>>model construction>>CD construct transfer function model.vi)
2. Create controls to change numerator and denominator of transfer function
3. Use draw transfer function block to display the transfer function (subVI in control and simulation>>control design>>model construction>>CD draw transfer function equation.vi) and create an indicator for it
4. To obtain the step response, use the corresponding VI (subVI in control and simulation>>control design>>Time Response>>CD step response.vi)
5. Create a graph indicator to view step response

Expected Output

step response of the system specified by the transfer function $H(S) = \frac{1}{s^2+0.8s+1}$



Numerator: 0 | 1 | 0 | 0

Denominator: 0 | 1 | 0.8 | 1

Equation: $\frac{1}{s^2 + 0.8s + 1}$

EXPERIMENT 3: FAMILIARIZATION OF DAQ CARD

Aim

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

Steps

1. A disc with perforations is attached to the shaft of the motor
2. An IR LED is kept ON on one side of disc and a phototransistor at the other.
3. As the motor rotates a train of pulses will be produced at the collector of the photo transistor. It can be connected to an analog input channel of the DAQ card.
4. Develop a VI in LabVIEW for data acquisition. There is an express VI called DAQ assistant for data acquisition. Set the channel number, type of the data etc. and data can be acquired.
5. For speed monitoring, train of pulses from the collector of the phototransistor is acquired and processed to the data into rpm.
6. The rpm data can be displayed numerically in a text indicator or in a meter indicator display.

Expected Output

A GUI displaying the speed of motor in rpm