Simulation and control of processes

Maninder Gill

University of New Hampshire, Durham

Follow this and additional works at: https://scholars.unh.edu/thesis

Recommended Citation
https://scholars.unh.edu/thesis/298

This Thesis is brought to you for free and open access by the Student Scholarship at University of New Hampshire Scholars' Repository. It has been accepted for inclusion in Master's Theses and Capstones by an authorized administrator of University of New Hampshire Scholars' Repository. For more information, please contact nicole.hentz@unh.edu.
SIMULATION AND CONTROL OF PROCESSES

BY

MANINDER GILL
B.Tech., Punjab Technical University, 2001

THESIS

Submitted to the University of New Hampshire
in partial Fulfillment of the
Requirements for the Degree of

Master of Science
in
Chemical Engineering
September, 2007
INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.
This thesis has been examined and approved.

P. T. " J k —
Thesis Director, Palligarnai T. Vasudevan
Professor of Chemical Engineering

Stephen S.T. Fan
Professor and Chair of Chemical engineering

Nivedita Gupta
Assistant Professor of Chemical Engineering

August 15, 2007
Date
DEDICATION

To my Parents and my Grandparents
ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor, Dr. P.T. Vasudevan for all his support and guidance. He has been an endless source of motivation and encouragement. He has always been there to give advice and help throughout my thesis without which I could not have finished my Masters thesis. Dr. Vasudevan is an exceptional professor and an outstanding human being. Thank you Dr. Vasu.

I would like to thank Dr. Stephen Fan and Dr. Nivedita Gupta for serving as members of this thesis committee.

Finally, I would like to take this opportunity to express my gratitude to all my friends for their support, in particular Xianping Shen who helped and encouraged me during last two years. Special thanks to my mentors, Dr. Rakesh Minocha and Dr. Subhash Minocha for being there for me throughout my Masters program to provide support as family.
# TABLE OF CONTENTS

DEDICTION .................................................................................................................iii

ACKNOWLEDGEMENT ...............................................................................................iv

LIST OF TABLES ............................................................................................................vii

LIST OF FIGURES ........................................................................................................viii

ABSTRACT ......................................................................................................................xii

CHAPTER | PAGE
--- | ---
1 | INTRODUCTION .................................................................1

2 | LABVIEW SIMULATION .........................................................6

2.1 Introduction .........................................................................................6

2.2 First order systems ........................................................................10

2.3 Two-tank liquid level system ..............................................22

2.4 Second order systems ......................................................25

2.5 Stability .....................................................................................30

2.6 Frequency analysis ...................................................................33

2.7 Controller tuning .......................................................................42

2.8 MIMO systems ...........................................................................45

3 | TEMPERATURE CONTROL......................................................53
LIST OF TABLES

Table 2.7 Ziegler-Nichols settings for controller tuning .......................... 42
LIST OF FIGURES

Figure 2.1a Front panel of LabVIEW ................................................................. 7
Figure 2.1b Numeric control and numeric indicator ........................................... 8
Figure 2.1c Block diagram of LabVIEW ............................................................ 9
Figure 2.2 Liquid level process ........................................................................... 10
Figure 2.2a Step function .................................................................................. 12
Figure 2.2b Impulse function ............................................................................. 13
Figure 2.3a Front panel showing step and impulse response of a first order system... 14
Figure 2.3b Block diagram for step and impulse response of a first order system.... 15
Figure 2.4a Front panel of a liquid level system .................................................. 16
Figure 2.4b Block diagram of a liquid level system ............................................ 17
Figure 2.5a Front panel of step response of transportation lag ......................... 18
Figure 2.5b Block diagram of step response of transportation lag ....................... 19
Figure 2.6a Front panel showing transfer function from given data .................... 20
Figure 2.6b Block diagram showing transfer function from given data ............... 21
Figure 2.7a Front panel of interacting system ..................................................... 23
Figure 2.7b Block diagram of interacting system ............................................... 23
Figure 2.8a Front panel of noninteracting system .............................................. 24
Figure 2.8b Block diagram of noninteracting system ......................................... 24
Figure 2.8c The characteristics of an underdamped second-order response ......... 25
Figure 2.9a Front panel of a second order (underdamped) system ..................... 27
Figure 2.20b Block diagram showing Relative Gain Array from steady state gain...50
Figure 2.21a Front panel showing Relative Gain Array from transfer functions......51
Figure 2.21b Block diagram showing Relative Gain Array from transfer functions...52
Figure 3.1 Temperature control experimental set up.............................................54
Figure 3.2a Front panel showing generation of (thermocouple) data.......................56
Figure 3.2b Block diagram panel showing generation of (thermocouple) data...........57
Figure 3.3 Process Reaction curve (PRC).................................................................59
Figure 3.4 FOPDT model fitting on PRC data...........................................................60
Figure 3.5 t*T versus t plot.......................................................................................61
Figure 3.6a Front panel showing controller tuning VI..............................................64
Figure 6b Block diagram showing controller tuning VI..........................................65
Figure 3.7 IAE versus time plot for heater temperature control..............................67
Figure 3.8 Temperature versus time plot (I=0, D=0)..............................................68
Figure 3.9 Temperature versus time plot (Kc=2, D=0)...........................................69
Figure 3.10 Temperature versus time plot (Kc=2, I=0.02)......................................69
Figure 4.1 Liquid level control system set up............................................................71
Figure 4.2a Front panel of VI to obtain step response curve.....................................73
Figure 4.2b Block diagram of VI to obtain step response curve...............................74
Figure 4.3 First order step response curve...............................................................75
Figure 4.4 NI Compact FieldPoint cFP-2010 ...........................................................77
Figure 4.5 Measurement and Automation Explorer window...................................79
Figure 4.6a Front panel showing data acquisition using compact FieldPoint..........80
Figure 4.6b  Block diagram showing data acquisition using compact FieldPoint......81

Figure 4.7  FieldPoint Project Explorer window.........................................................83

Figure 4.8  Target.vi of level Project Explorer.............................................................84

Figure 4.9  Host.vi of level Project Explorer.................................................................85

Figure 4.10a  Front panel for level controller tuning.......................................................87

Figure 4.10b  Block diagram for level controller tuning....................................................88

Figure 4.11  IAE versus time plot (I =0, P =20, 22, 24, 26, 28).......................................89

Figure 4.12  IAE versus time plot (P =24, I = 0.25, 0.35, 0.45, 0.55)................................89
ABSTRACT

LABVIEW SIMULATION AND CONTROLLER TUNING

by

Maninder Gill

University of New Hampshire, September, 2007

Process control is required in most of the industries to regulate the output of a specific chemical process. Any improvement in control design results in process optimization, consistent production, hence less waste. In this thesis, process dynamics of first and second order systems was studied in terms of a response to a step input change. A software called LabVIEW from National Instruments was used to simulate a number of processes to observe time response and frequency response to calculate gain, phase margin and Bode plots. This was followed by system stability analysis. Programs in LabVIEW were created to calculate Ziegler Nichols settings for controller tuning and dynamic Relative Gain Array for Multiple-Input Multiple-Output (MIMO) systems.

Studies were carried out on the dynamic performance criteria for controller tuning by minimizing Integral Absolute Error (IAE). This was possible since data could be collected and analyzed real time. LabVIEW programs were created to fine tune the controller starting with P, I, D values obtained using Cohen and Coon method. Process parameters required for the calculations were determined from Process Reaction Curve (PRC). Open loop circuit was used to measure the
temperature/level to obtain a Process Reaction curve. Control of temperature in a
heater was achieved by means of closed loop in which power supplied to the heater by
a solid state relay was regulated according to the feedback obtained from the
thermocouple. Results showed that PRC method was unsuitable for this process.
Temperature controller was tuned using trial and error method and best settings were
obtained as $P = 2$, $I = 0.02$, $D = 0.5$. It was desired to use Compact FieldPoint by
National Instruments (NI) for liquid level controller tuning. After configuration and
testing, it was found, however that output signal from the FieldPoint was in the range
of 4-12mA which resulted in opening the control valve to only half of its full capacity.
The problem was solved by using traditional Data acquisition device from NI to
acquire data. PI controller was tuned from the starting values obtained by Cohen and
Coon method using error-integral criteria. The best controller settings were obtained
as $P = 24$, $I = 0.35$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
INTRODUCTION

Process control is a field of engineering which deals with controlling various parameters of a specific process according to a model with the help of controllers. It focuses on modeling various kinds of dynamic systems and the design and tuning of controllers that will cause these systems to behave in the desired manner. Control systems running automatically enable a process to operate safely. This can be achieved by continually monitoring process operating parameters such as liquid level, temperature, pressure, concentration and then making decisions like opening or closing a valve, turning up heater, slowing down the flow rate and so on so that the required parameters are maintained at the desired set points.

In this thesis, the first task was to determine the time response of first and second order systems to various input disturbances. This may be important to know because in simple terms, one may be interested in modeling cruise control in an automobile or determine the response of the car to a change in the incline of the road. Or, one may be interested in designing a temperature control system to avoid cold or very hot showers. This was followed by stability analysis, frequency response, determination of interactions within a system and controller tuning. The physics of a process and mathematics involved in deriving an overall transfer function and designing a suitable control system was...
thoroughly understood. Chemical engineering related examples include control of liquid level in a reactor, maintaining temperature at a desired value in a purification column or a heater and so on. It is important to understand the transient behavior of such processes before designing a suitable feedback control system.

There are number of process control softwares available to automatically monitor and control chemical processes. **Control Station** is a simulation software that enables control engineers to tackle complex control issues effectively and quickly. It is used by engineers for PID controller tuning, performance analysis, model fitting of process data and customized control loop simulations [1]. **MATLAB** (MATrix LABoratory) is a high level programming language developed by Mathworks Inc. It is a tool which allows doing numerical computations with matrices and vectors, plotting of functions and implementation of algorithms [2]. Control system toolbox has tools to design, analyze and tune linear control systems. **LabVIEW** is another important process control software developed by National Instruments (NI) used to create programs for automated control in addition to other measurement and control applications. The graphical dataflow language used in LabVIEW is called “G”. LabVIEW has built-in measurement and analysis functions that can be simply dragged and dropped and connected through wires (for data flow) to build programs. Hence virtual representations of lab equipment can be created in the program to correlate with the actual environment. One of the benefits of LabVIEW over other development softwares is the support for accessing data acquisition and instrumentation hardware thus allowing standard software interfaces to communicate with hardware devices. LabVIEW can acquire data from a number of such devices in no time and analyze with more than 500 built-in functions and then present the results in the
form of graphical and numeric indicators. Using LabVIEW, one can develop real-time embedded control application on a PC and then download and execute the program to run on an independent hardware target such as Compact Fieldpoint developed by NI [3]. Due to the above reasons, LabVIEW was chosen among other software for this thesis to simulate chemical processes and tune PID controllers for temperature and liquid level control. LabVIEW was used to simulate various dynamic systems to know their transient response to any forcing function. The most common forcing functions: step and pulse have been found very useful in theoretical and experimental analysis of numerous control systems. In order to design and model various dynamic systems, LabVIEW Control Design toolkit and Simulation Module were used.

Once this was accomplished, it was important to gain an understanding of the hardware associated with such systems. These include sensors, transmitters, control valves, data acquisition (DAQ) devices and controllers. On a simple level, one would consider a single input single output system (for example, maintaining the temperature of the water from a shower head at 90°F by regulating the flow rate of hot water.) Under certain conditions, the process can be rendered unstable, making it impossible to control the process. The problem of coming up with the best set of control parameters that will result in the quickest response to any perturbation is a complicated one and is referred as “Controller Tuning.”

There are different kinds of controllers being used in industries depending on different applications and price ranges. An on-off controller is the simplest form of controller used by almost all domestic thermostats which has output either off or on. For example, for heating control, it turns the heater on if the temperature of the system is
below set point and switches off the heater when above set point. A proportional controller works better in a way that it applies power proportional to the difference in temperature between the heater and the set point. Increase in its gain results in faster response to the changes in set point but becomes progressively under damped and eventually system becomes stable. The most commonly and most effective controller being used in industries is PID controller which has three modes- Proportional, Integral and Derivative all combined into one. The proportional mode acts based on the current error, the integral error acts based on recent errors and derivative action acts based on the rate by which error has been changing. PID controller can be used as PI, PD, P or I controller by turning off respective actions. Controller tuning is the adjustment of P, I and D actions to the optimum values for which the system is most stable with minimum or no oscillations [3-4]. Generally, tuning is done in industry using simple performance criteria such as ¾ DR.

In this thesis, experiments were done to attain liquid level control using a simulated program in LabVIEW with the help of an embedded controller called Compact FieldPoint (cFP) from NI. cFP is composed of analog and discrete I/O modules which are highly expandable and easy to use. Various kinds of sensors can be directly connected to these high-accuracy I/O modules for filtration and calibration of raw sensor signals to engineering units and perform analysis to look for problems. cFP is highly reliable and can withstand high temperatures (-25 to 60 °C) and tolerate shock up to 50g [5]. What differentiates cFP from a standard DAQ is that DAQ needs to be hosted by a computer and other human interface devices, but cFP’s processor is designed in such a way that it does not require direct connection to monitor or other human interface to monitor and
control the system. The LabVIEW program was downloaded to the cFP for stand-alone operation and optimization models were created where data are processed real time and the best controller parameters are obtained based on the minimization of dynamic performance criteria. Thus, the controller tuning was based on continuous analysis of data and using minimization techniques based on data gathered real-time. Additionally, the experiments were operated from remote destinations with the FieldPoint module. Thus the acquisition of a real-time Ethernet module enables continuous monitoring of the process, which is an important aspect of supervisory control.

However due to output restrictions, cFP could not be used for tuning. Instead, DAQ device from NI was used to acquire real time data. For liquid level controller tuning, the integral and derivative actions were turned off and optimum value of P was found out by minimizing the Integral of Absolute Error (IAE) using LabVIEW. Then P was fixed at that value and I varied. The Process Reaction Curve was generated using open loop. Dynamic tuning performance criteria were used to fine tune the controller. This was possible since data could be collected and analyzed real time on LabVIEW. For heater temperature control, traditional DAQ was used for controller tuning and Reaction Curve (open loop) method was used. Cohen and Coon tuning criteria were used to find the initial PID parameters.

The work presented in the thesis is separated into three sections. Chapter 2 presents the simulation programs in LabVIEW along with background information and introduction. Chapter 3 describes the temperature control of an electric heater. Chapter 4 presents the liquid level control. Chapter 5 provides the summary and future work.
CHAPTER II

LABVIEW SIMULATION

2.1 Introduction

LabVIEW is a very useful graphical language which stands for Laboratory Virtual Instrumentation Engineering Workbench. This is an important software tool by National Instruments (NI) used extensively in industries to design test, measurement and control systems since 1986. LabVIEW, being a graphical dataflow language, one can easily view, modify and control inputs which is of particular interest to scientists and engineers [6]. LabVIEW is widely used in academic research and numerous industries including pharmaceutical, paint, aerospace, automotive, semiconductor etc. in manufacturing and all stages of the product development process. Not only offline simulation, LabVIEW can be used to design real time data acquisition and advanced process control implementation.

A program in LabVIEW is called a Virtual Instrument (VI). LabVIEW has two components- Front Panel and Block Diagram. The Front Panel (as shown in Figure 2.1a) is the interactive user interface and contains controls like knobs, toggle switches, dials; and indicators like graphs, charts, LEDs and other displays on the controls palette. These controls and indicators allow an operator to enter and retrieve data from a running VI.
The front panel workspace can be right clicked to display the controls palette which has subpalettes containing objects needed to build measurement applications. Controls palette contains number of objects such as: Numeric, Boolean, Array, Matrix, Graph, String and Path, Ring and Enum etc. Each one of these has subpalettes. The subpalette for ‘Numeric’ is shown in the Figure 2.1a.

Figure 2.1a: Front panel of LabVIEW

The most commonly used numeric objects are: numeric control and numeric indicator are...
shown in figure 2.1b.

Figure 2.1b: 1. Increment and decrement buttons, 2. Numeric control, 3. Numeric indicator

A number is entered as input in the numeric control. The value can be changed using increment and decrement buttons and output is shown in the numeric indicator.

After the front panel is built, the code (in language ‘G’) is added using graphical representations to control the front panel objects on the block diagram (as shown in Figure 2.1c). LabVIEW gives a pictorial representation of a program unlike other traditional languages such as C or Fortran which are text based languages. All built-in functions and VIs used to build a block diagram are available on functions palette. To build a VI or program, functions are dragged and dropped on the block diagram from the palette which has about 500 measurement and analysis functions. Functions palette also contains nodes, terminals and VIs. Nodes are objects that perform operations while VI is running and have inputs and/or outputs. Nodes can be structures such as For loops, While loops, case structures or functions such as addition and subtraction nodes. Front panel objects appear as terminals on the block diagram. Terminals exchange information
between the front panel and block diagram as entry and exit ports and are analogous to constants and parameters in text based language. Some VIs can be dragged and dropped from functions palette and LabVIEW considers it to be a subVI. When such VI is double-clicked its front panel and block diagram appear on the workplace. Various functions, nodes, terminals and VIs are then connected through wires within the simulation mode which represent the flow of data.

Figure 2.1c: Block Diagram of LabVIEW

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
2.2 First order systems

One of the major goals of this chapter of the thesis is to understand the process dynamics in terms of the response of a process to a step input change. A first order system has a differential equation:

\[ \tau \frac{dx}{dt} + x(t) = Ky(t) \]  

(1)

where:

- \( \tau \) = time constant of the system
- \( y(t) \) = response of the system
- \( K \) = steady-state gain

Laplace transformation of above differential equation gives the following standard form of a first order transfer function:

\[ \frac{x(s)}{y(s)} = \frac{K}{\tau s + 1} \]  

(2)

In this thesis, a liquid-level process was studied and simulated in LabVIEW. Consider a tank of uniform cross sectional area \( A \) and valve resistance \( R \) as shown in the Figure 2.2:

\[ q(t) \]

\[ \downarrow \]

\[ h(t) \]

\[ R \]

\[ \times \]

\[ q_0(t) \]

Figure 2.2 : Liquid level process

\( q \) and \( q_0 \) are the volumetric inlet and outlet flow rates.
It is desired to determine the transfer function that relates head to flow

\[ q_0 = \frac{h}{R} \]  

(3)

Mass balance around the tank gives:

\[ \rho q(t) - \rho q_0(t) = d(\rho Ah)/dt \]  

(4)

\[ q(t) - q_0(t) = d(Ah)/dt \] (assuming constant density)

Substituting \( q_0 = \frac{h}{R} \) in above equation, we get

\[ q - \frac{h}{R} = Adh/dt \]

At steady state,

\[ q_s - \frac{h_s}{R} = 0 \]

Subtracting the above two equations and replacing \( q - q_s \) by \( Q \) and \( h - h_s \) by \( H \), we get

\[ Q = \frac{H}{R} + AdH/dt \]

Laplace transform of above equation gives:

\[ Q(s) = \frac{H(s)}{R} + AsH(s) \]  

(5)

Rearranging the above expression gives:

\[ \frac{H(s)}{Q(s)} = \frac{R}{\tau s + 1} \] where \( \tau = AR \)

(6)

This is the standard transfer function for a first order system [4].

**Forcing functions**

Once the transfer function has been obtained, it is important to know its transient response to any forcing function. These forcing functions are very useful in theoretical and experimental aspects of process control. Two forcing functions used in this thesis are:

**2.2.1 Step Function**

A step function of magnitude A is shown in Figure 2.2a and can be represented as:

\[ X(t) = Au(t) \]  

where \( u(t) \) is a unit step function
Step Response: The step response gives time behavior of an output of a dynamical system when its control input is a step function [4]. When a step change of magnitude $A$ is introduced into a first order system, $X(s) = A/s$

The first order transfer function $Y(s)/X(s) = 1/(\tau s + 1)$ becomes

$Y(s) = (A/s)( 1/(\tau s + 1) )$

Using partial fractions, this can be expanded as:

$Y(s) = (A/\tau)\{ (s)(s + 1/\tau) \} = C_1/s + C_2/(s + 1/\tau)$  \hspace{1cm} (7)

On solving, $C_1 = A$ and $C_2 = -A$

Substituting the values of constants in the above equation and taking the inverse transform give the time response for $Y$:

$Y(t) = 0 \hspace{0.5cm} t < 0$

$Y(t) = A(1 - e^{-t/\tau}) \hspace{0.5cm} t \geq 0$  \hspace{1cm} (8)

2.2.2 Impulse Function

Mathematically, an impulse function of magnitude $A$ is defined as:
\(X(t) = A\delta(t)\) where \(\delta(t)\) is the unit-impulse function (shown in Figure 2.2b).

\[
\begin{align*}
X &= 0; \quad t < 0 \\
X &= \frac{A}{b}; \quad 0 < t < b \\
X &= 0; \quad t > b
\end{align*}
\]

Figure 2.2b: Impulse function

Impulse Response: When an impulse function of unit magnitude is introduced into a first order system,

\[X(s) = 1\]

The first order transfer function \(Y(s)/X(s) = \frac{1}{\tau s + 1}\) becomes

\[Y(s) = \frac{1}{\tau s + 1} = \frac{1/\tau}{s + 1/\tau}\]

The inverse of \(Y(s)\) is given as:

\[\tau Y(t) = e^{-t/\tau}\]
When a step and impulse change of unit magnitude was introduced into a first-order system, step response curve obtained is shown in Figure 2.3a. The transfer function was entered in the form of numerator and denominator on the front panel and output was obtained as shown in Figure 2.3a.

Figure 2.3a: Front panel showing step and impulse response of a first order system
Figure 2.3b shows the block diagram of the above VI. A function called 'CD Construct Transfer Function Model' was placed from Control Design palette with numerator and denominator as the controls to be entered on front panel. This function generates the transfer function which is connected to ‘CD Step Response’ and ‘CD Impulse Response’ to get the respective response as output. To see the graphs, step and impulse response graphs are connected as indicators. ‘CD Draw Transfer Function Equation’ is used to display the transfer function on the front panel as shown in Figure 2.3b.

Figure 2.3b: Block diagram for step and impulse response of a first order system
In an example of liquid level, one might be interested to know the transfer function between input and output flow rates when relation between head and input, output are known. A VI was created to display the final transfer function as shown in Figure 2.4a.

Figure 2.4a: Front panel of a liquid level system

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 2.4b shows the block diagram of the above VI. Two transfer functions were generated using ‘CD Construct Transfer Function’ and then connected to ‘CD Series’ function to obtain the overall transfer function. The output is displayed on the front panel.

Figure 2.4b: Block diagram of a liquid level system
Figure 2.5a shows the simple step response curve for a given transportation lag or dead time. A dead time of .75 seconds was entered on the front panel and it was observed when the step change was made at $t = 0$ in $x(t)$, the change was not detected until 0.75 seconds.

Figure 2.5a: Front panel of step response of transportation lag
Figure 2.5b shows the block diagram of the above VI. A VI from Control design palette was placed on the block diagram to generate the transfer function. Constant value of unity was entered for gain and time constant and delay (or transportation lag) was made to vary from control panel to obtain its step response.

![Block diagram for step response of transportation lag](image)

Figure 2.5b: Block diagram for step response of transportation lag
A VI was created (as shown in Figure 2.6a) to fit the given experimental data into a first order plus dead time (FOPDT) model. The graph obtained looks similar to time response graph in Figure 2.3a. The FOPDT parameters: gain, time constant and dead time were calculated using this VI and a transfer function was obtained as displayed on the front panel in the figure. This modeling approach is well suited for modeling high-order processes making it useful for representing industrial processes which exhibit high-order behavior.

Figure 2.6a: Front panel showing transfer function from given data
Figure 2.6b shows the block diagram of the above VI. The given x-y data were added to individual n-dimensional arrays as elements using 'Build Array' function. All elements of array containing y data were compared with the next element using subtraction node. A particular value was found such that all following elements had the same value. This value was the **gain**. 63.2% of gain was calculated and a value corresponding to this was found out in x data using 'Index Array' function. This value was the **time constant**. Both gain and time constant were entered to generate the first order transfer function whose step response was obtained as shown in the front panel above.

Figure 2.6b: Block diagram showing transfer function from given data
2.3 Two-tank liquid-level system

Two-tank liquid-level systems can be aligned as interacting and noninteracting systems shown in Figure 2.7a and 2.8a respectively. In an interacting system, the outlet flow from the first tank depends on the difference between the heights of two tanks whereas in a noninteracting system, the variation in height in the second tank does not affect the transient response occurring in tank 1 [7]. It was found that as the area of the tank is increased, the response time of interacting system goes on increasing but change in area has no affect on response time of noninteracting system. Also Figure 2.8a shows the comparison of a single tank and two tanks in noninteracting system. The shape of the response curve for two tanks was found to be sigmoidal unlike the response curve of a single tank which is a first order system. Respective block diagrams are shown in Figure 2.7b and 2.8b.

For non interacting system,

\[ \frac{H_2(s)}{Q(s)} = R_2/\{(\tau_1 s + 1)(\tau_2 s + 1)\} = R_2/(\tau_1 \tau_2 s^2 + (\tau_1 + \tau_2) s + 1) \]  

(9)

For interacting system,

\[ \frac{H_2(s)}{Q(s)} = R_2/\{\tau_1 \tau_2 s^2 + (\tau_1 + \tau_2 + A_1 R_2) s + 1\} \]  

(10)

where:

\[ \tau_1, \tau_2 \text{ and } h_1, h_2 \text{ are the time constants and heights of first and second tank respectively.} \]

\[ Q = \text{inlet flow rate} \]

\[ R_2 = \text{valve resistance} \]

\[ A_1 = \text{area of the first tank} \]
Figure 2.7a: Front panel of interacting system

Figure 2.7b: Block diagram of interacting system

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 2.8a: Front panel of noninteracting system

Figure 2.8b: Block diagram of noninteracting system

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
2.4 Second order systems

A second order system is represented by a second order differential equation given by:

\[ \tau^2 (d^2 Y/dt^2) + 2\zeta \tau (dY/dt) + Y = X(t) \]  \hspace{1cm} (11)

where \( \zeta \) = damping coefficient (or factor) of the system

The above equation can be rearranged to give following standard transfer function for a second order transfer system:

\[ \frac{Y(s)}{X(s)} = \frac{1}{\tau^2 s^2 + 2\zeta \tau s + 1} \]  \hspace{1cm} (12)

The system is said to be underdamped when \( \zeta < 1 \) (two complex conjugate roots), overdamped when \( \zeta > 1 \) (two real, distinct roots) and critically damped when \( \zeta = 1 \) (two real equal roots). Step response characteristics of underdamped second-order processes are:

Figure 2.8c: The characteristics of an underdamped second-order response [8]
1. **Overshoot.** Overshoot is a measure of the distance between the first peak and the new steady state value and is expressed as the ratio a/c as shown in Figure 2.8c. In terms of $\zeta$, the overshoot for a unit step is given as:

$$\text{Overshoot} = \exp \left( -\frac{\pi \zeta}{(1- \xi^2)^{0.5}} \right)$$

2. **Decay ratio.** Decay ratio(b/a) is a measure of ratio of the sizes of successive peaks and is given as:

$$\text{Decay ratio} = (\text{overshoot})^2 = \exp \left( -\frac{2\pi \zeta}{(1- \xi^2)^{0.5}} \right)$$

Increase in damping factor means greater damping and hence greater decay.

3. **Rise time.** Rise time is the time required for the response to reach the new steady state value and it increases with increasing value of damping factor.

4. **Period of oscillation (T).** Period of oscillation(time/cycle) is defined as the time between successive peaks and is given by:

$$T = \frac{2\pi}{(1- \xi^2)^{0.5}}$$

5. **Peak time.** Peak time is the time it takes to first reach the peak of response.

6. **Settling time.** Output of the system changes when an input is applied to it. The time taken by the system to settle within a specified value (lets say, 2%) is known as the settling time.
The step and impulse response for underdamped system were obtained using LabVIEW as a function of damping factor and is shown in the Figure 2.9a. The response curves obtained by varying the value of damping factor from 0.2 to 0.8 were all found to be oscillatory in nature and became less oscillatory as $\zeta$ was increased. Also shown in the figure are underdamped step response characteristics for a second order system. Gain, damping ratio and time constant were entered on the front panel as inputs and all important parameters such as overshoot, decay ratio, time period, rise time, peak time and settling time were obtained as outputs.

![Figure 2.9a: Front panel of a second order (underdamped) system](image)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 2.9b shows the block diagram of the above VI. 'CD Construct special TF Model' was used to generate a second order transfer function by choosing '2nd order' from the dropbox. All parameters required to represent a second order transfer function were entered from the front panel and then used in formulas to calculate overshoot and decay ratio. A function called 'CD Parametric Time Response' was used to extract parametric information such as rise time, peak time, settling time and steady gain based on specified time response. 'Time Response Parametric Data' was connected as indicator to display the data on the front panel.

Figure 2.9b: Block diagram of a second order (underdamped) system
Step and impulse response for overdamped system (see Figure 2.10) were obtained using the VI shown in Figure 2.9a, b by entering $\zeta > 1$ on front panel. The response curves were obtained by varying $\zeta$ from 1.2 to 1.8 and were all found to be nonoscillatory and became more sluggish as $\zeta$ was increased. In this case there is no overshoot, decay ratio and time period because of absence of oscillations.

Figure 2.10: Second order (overdamped) system
2.5 Stability

Once first and second order systems were studied and simulated, stability analysis was carried out. As shown in Figure 2.11a, \( c(s) \), \( p(s) \) and \( h(s) \) represent transfer functions of a controller, process and a sensor respectively. Up to second order, the systems were found to be inherently stable. Let us consider the response of a third order system and vary \( K_c \) (controller gain) keeping \( \tau_1 \), \( \tau_2 \), \( \tau_3 \) constant. As \( K_c \) increases, the system becomes more oscillatory and beyond a certain value of \( K_c \) the successive peaks of the response grow instead of decay; this type of response is called unstable.

Consider a basic single-loop control system shown in Figure 2.11.

![Basic single-loop control system](image)

**Figure 2.11: Basic single-loop control system**

From the above block diagram of the control system, the overall transfer function can be written as:

\[
C = \frac{G_1 G_2 R}{(1 + G_1 G_2 H)} + \frac{G_2 U}{(1 + G_1 G_2 H)} \quad (13)
\]

or

\[
C = \frac{G_1 G_2 R}{(1 + G)} + \frac{G_2 U}{(1 + G)} \quad \text{where} \quad G = G_1 G_2 H
\]

where \( G_1 \), \( G_2 \), \( H \) are the transfer functions of a controller, process and sensor respectively. \( U \) represents the disturbance. The characteristic equation for the above transfer function is:
A control system is unstable if any roots of its characteristic equation are to the right of imaginary axis, otherwise the system is stable. A system is said to be marginally stable if the roots lie on the imaginary axis.

If the transfer functions of various block elements of a simple control system are known, overall transfer function can be easily found out using a program in LabVIEW (shown in Figure 2.11a). Based on an overall transfer function, dynamic characteristics such as stability can be determined. Controller gain can be estimated by varying $K_c$ at which the system will be marginally stable or unstable.

![Front panel showing stability of a control system](image-url)
Figure 2.11b shows the block diagram of the above VI. The transfer functions $c(s)$ and $p(s)$ were connected using a function called ‘CD Series’ and the resulting function and $H(s)$ were sent to ‘CD Feedback’ to give overall transfer function which is displayed on the front panel. The input parameters were entered on the front panel itself. A function called ‘Stability’ was used to determine the stability of the system based on characteristic equation of the overall function. All block diagram objects were put into simulation loop as functions from simulation palette can only be dragged and dropped inside the loop; they can’t be placed anywhere outside the loop.

Figure 2.11b: Block diagram showing stability of a control system
2.6 Frequency analysis

Frequency response of first and second order systems is a valuable tool in the design and analysis of control systems. To begin with, let us consider a simple first order system with transfer function:

\[ G(s) = \frac{1}{(\tau s + 1)} \]

On substituting \( s = i\omega \) and converting the transfer function in rectangular form, \( z = a + ib \) to polar form, \( |z| \rho \theta \), we get

\[ G(i\omega) = \frac{1}{(\omega^2 \tau^2 + 1)^{0.5}} \tan^{-1}(-\omega \tau) \]

where \( \omega \) is the frequency.

Hence for the frequency response of a first order system,

- Amplitude Ratio (AR) = \( |G(i\omega)| \)
- Phase angle = \( \theta G(i\omega) \)
A VI in LabVIEW (Figure 2.12a) was created to determine the AR and phase angle of any first order system. Also a Bode plot is shown in the figure which is a combination of a Bode magnitude plot and Bode phase plot. A graph of log magnitude in AR ratio or Decibel ($= 20 \log_{10}(\text{AR})$) versus frequency is called a Bode magnitude plot and a graph of phase versus frequency is known as Bode phase plot.

Figure 2.12a: Front panel showing Bode plot of a first order system
Figure 2.12b shows the block diagram for the above VI. First order transfer function was connected to ‘CD Evaluate at frequency’ to determine its magnitude and phase. ‘CD Bode’ was used from the control design palette to obtain the Bode plot. First order transfer function can be replaced by second order transfer function, transportation lag or PID to obtain their respective Bode plots as shown in following figures.

Figure 2.12b: Block diagram showing Bode plot of a first order system
Similarly a VI (as shown in Fig. 2.13a) was created to enter the second order system parameters to give output in the form of magnitude and phase Bode plots.

Figure 2.13a: Front panel showing Bode plot of a second order system

Figure 2.13b: Block diagram showing Bode plot of a second order system
If only transportation lag is given, following VI can be used to calculate its phase and magnitude.

Figure 2.14a: Front panel showing Bode plot of transportation lag

Figure 2.14b: Block diagram showing Bode plot of transportation lag
For a PID controller alone, a VI (Fig. 2.15a) was created to find its Bode plot, phase and magnitude. For example, for $K_c = 1$, $\tau_i = 0.1(I = K_c/\tau_i)$, $\tau_D = 25(D = K_c\tau_D)$, magnitude and phase were obtained as 24.92 and 87.7 degrees respectively.

Figure 2.15a: Front panel showing Bode plot of a PID controller

Figure 2.15b: Block diagram showing Bode plot of a PID controller
Degree of stability is determined by two important factors called Gain Margin (GM) and Phase Margin (PM). The crossover frequency \( \omega_{co} \) is the frequency at which the phase lag is \( 180^\circ \) and AR is A. From the Bode criterion, if A exceeds unity the system becomes unstable. In Figure 2.15c, A is assumed to be below unity and hence the system is stable. In order to have an adequate safety factor, A should be much lesser than unity otherwise the system is only "almost stable".

![Open loop Bode diagram]

Figure 2.15c: Open loop Bode diagram

Gain margin term is used to give quantitative measure to the above considerations and is defined as: Gain margin = 1/A

Phase margin is the difference between \( 180^\circ \) and the phase lag at the frequency for which the gain is unity. Typical specifications for design are that GM and PM must be greater than 1.7 and \( 30^\circ \) respectively [4].
A VI was created on LabVIEW to determine GM and PM for the second order process. The parameters required to represent a second order system were entered as inputs on the front panel and GM and PM were obtained as outputs as shown in Figure 2.16a.

Figure 2.16a: Front panel showing gain and phase margin
Figure 2.16b shows the block diagram of the above VI. A second order (or first order) transfer function was created and connected to ‘CD Gain and phase margin’ which calculates gain and phase margin and also the Bode plot.

Figure 2.16b: Block diagram showing gain and phase margin
2.7 Controller tuning

A controller is tuned in order to adjust a PID loop to see how aggressively the controller reacts to errors between desired set point and measured variable [9]. Ziegler-Nichols (ZN) settings give starting values for controller tuning which can be used to obtain most optimum parameters based on the minimization of error using dynamic performance criteria. According to Bode criterion, the ultimate controller gain ($K_u$) is the one which would cause the system to be on the verge of instability and is equal to $1/A$. The period of the sustained cycling that would occur if a proportional controller with gain $K_u$ were used is known as the ultimate period ($P_u$) and is given by:

$$P_u = 2\pi/\omega_{co}$$

The Z-N settings for controllers are determined directly from $K_u$ and $P_u$. The Z-N controller tuning chart is as shown below:

<table>
<thead>
<tr>
<th>Control Type</th>
<th>$K_c$</th>
<th>$\tau$</th>
<th>$TP$</th>
</tr>
</thead>
<tbody>
<tr>
<td>P control</td>
<td>$K_u/2$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PI control</td>
<td>$K_u/2.2$</td>
<td>$P_u/1.2$</td>
<td></td>
</tr>
<tr>
<td>PID control</td>
<td>$K_u/1.7$</td>
<td>$P_u/2$</td>
<td>$P_u/8$</td>
</tr>
</tbody>
</table>

Table 2.7: Ziegler-Nichols Settings for controller tuning [10]
The parameters required to represent FOPDT model were entered on the Front panel of a LabVIEW program and controller settings were chosen as PID out of P, PI and PID. Figure 2.17a shows the simulation created to determine $K_c$, $T_i$, $T_d$ according to Z-N criteria along with GM and PM and the Bode plot for a PID controller.

Figure 2.17a: Front panel showing Ziegler Nichols Settings
Figure 2.17b shows the block diagram of the above VI. ‘CD Series’ was used to find the overall transfer function for the first order system and PID controller. The output was connected to ‘CD Step response’ to obtain its step response graph. Also the output was wired to ‘CD Gain and Phase Margin’ to obtain the Bode plot and all other output parameters were bundled together in a cluster from which the required data was extracted by specifying the name of a cluster element. The ultimate gain and period were determined to be used in the formulas to calculate Z-N settings for a PID controller using formula node.

Figure 2.17b: Block diagram showing Ziegler Nichols Settings
2.8 MIMO systems

The systems simulated so far are single-input single-output (SISO) systems in which one measured variable (MV) is used to control one controlled variable (CV). The processes encountered in almost all industries are usually multiple-input multiple-output (MIMO) systems in which there are two or more than two CVs and MVs. Multiple inputs and outputs lead to coupling or interaction between control loops in a MIMO process. In some cases, weak interaction can be ignored; in other cases it may be necessary to consider all interactions simultaneously. The pairing of CVs and MVs has a significant effect on the control performance for MIMO processes. The block diagram for a two-input/two-output system is as shown below:

Figure 2.18: Block diagram of a two input/two-output process

Mathematically, above figure can be represented as:
\[ y_1(s) = G_{11}'(s)c_1(s) + G_{12}'(s)c_2(s) \]

\[ y_2(s) = G_{22}'(s)c_2(s) + G_{21}'(s)c_1(s) \]

Input \( c_1 \) affects both outputs \( y_1 \) and \( y_2 \) and input \( c_2 \) affects both \( y_1 \) and \( y_2 \). The process is coupled because both inputs affect both outputs. Coupled MIMO systems are commonly observed in chemical process industries and bio-tech industries. Deciding which MV should be used to control which CV is called choosing the pairings for example \((c, y)\) pairing in above example.

Relative Gain Array (RGA), a measure of steady-state coupling is the ratio of the process gain without coupling to the process gain with coupling. For a 2X2 system, RGA is given by:

\[
\text{RGA} = \begin{pmatrix}
\lambda_{11} & \lambda_{12} \\
\lambda_{21} & \lambda_{22}
\end{pmatrix}
\]

where

\( \lambda_{11} \) = ratio of open loop gain for the effect of \( c_1 \) on \( y_1 \) (keeping \( c_2 \) constant) to gain between \( c_1 \) and \( y_1 \) while keeping \( y_2 \) at its setpoint

\( \lambda_{12} \) = ratio of open loop gain for the effect of \( c_1 \) on \( y_2 \) (keeping \( c_1 \) constant) to gain between \( c_1 \) and \( y_2 \) while keeping \( y_2 \) at its setpoint

\( \lambda_{21} \) = ratio of open loop gain for the effect of \( c_2 \) on \( y_1 \) (keeping \( c_2 \) constant) to gain between \( c_2 \) and \( y_1 \) while keeping \( y_1 \) at its setpoint

\( \lambda_{22} \) = ratio of open loop gain for the effect of \( c_2 \) on \( y_2 \) (keeping \( c_1 \) constant) to gain between \( c_2 \) and \( y_2 \) while keeping \( y_1 \) at its setpoint [3]
In case of no coupling among inputs and outputs, numerator of $\lambda_{11}$ equals the denominator. Thus for the least steady state coupling, $\lambda_{11}$ is almost equal to 1.

Figure 2.19a shows a VI created to determine the type of model such as SISO, SIMO, MISO, MIMO etc. depending on the number of inputs and outputs entered in the form of transfer functions. Constant parameters for the transfer function were added on the block diagram and the data was extracted in the form of numerator and denominator.

![Image of VI front panel](image-url)

Figure 2.19a: Front panel to determine the type of model
Figure 2.19b shows the block diagram of the above VI. ‘CD Construct TF Model’ was used to enter multiple inputs and outputs. In the following example, constant values were entered as single input and single output. ‘CD Get Data from Model’ was used to extract the parameters from the transfer function in the form of numerator and denominator. ‘CD Verify Model Type’ was used to determine the type of model for example, SISO, SIMO, MISO or MIMO. ‘CD Verify MIMO properties’ was used to determine the number of inputs and outputs entered based on the model chosen.

Figure 2.19b: Block diagram to determine the type of model
A LabVIEW VI was created which gives the RGA matrix as output when the steady-state gain matrix is entered as input for a 2X2 system as shown in Figure 2.20a.

Figure 2.20a: Front panel showing Relative Gain Array from steady state gain
Figure 2.20b shows the block diagram of the above VI. \( \lambda_{11} \) was calculated using:

\[
\lambda_{11} = \frac{1}{1 - \frac{(K_{12}K_{21})}{(K_{11}K_{22})}}
\]

where \( K_{11}, K_{12}, K_{22}, K_{21} \) are the elements of a 2X2 steady-state gain matrix. Relative gain matrix was obtained using the concept that the sum of \( \lambda \)s in any row or column is equal to 1.

Figure 2.20b: Block diagram showing Relative Gain Array from steady state gain.
A LabVIEW program was then created for the RGA when only the transfer functions are known instead of steady state gains. The program calculates the magnitude of individual transfer functions and then calculates the required RGA as shown in Figure 2.21a. The sum of a row and a column of the RGA matrix was found to be unity.

Figure 2.21a: Front panel showing Relative Gain Array from transfer functions
Figure 2.21b shows the block diagram for the above VI. Transfer functions were created using the given parameters and then wired to ‘CD Evaluate at Frequency’ VI which gives magnitude of a transfer function as an output. The magnitude is the steady state gain from which RGA matrix can be obtained following the same steps as used in Figure 2.20b.

Figure 2.21b: Block diagram showing Relative Gain Array from transfer functions
CHAPTER III

TEMPERATURE CONTROL

Temperature control is very important in almost all industries, especially chemical industries, where control and uniformity of temperature are directly correlated with higher yields, improved quality, maximum conversion, and lower costs. Temperature is a critical parameter for reaction, crystallization, combustion, distillation or drying in process industries. Temperature control allows the process to run at exact specified temperature with minimum fluctuations. Overshoot and undershoot of the preset temperature is not acceptable in pharmaceutical, food and biotech companies where products such as biomaterials and healthcare products are required to be treated at certain specified temperature to reduce contamination and degradation. In this thesis, we investigated the control aspects of a home-built temperature controller.

3.1 Experimental setup

A vertical tube electric furnace (HOSKINS) of type FA120 was used for the temperature control experiments. The furnace was ordered from Hoskins Manufacturing Co., Detroit, Michigan with specifications: 110V and 3.27Amp. Experimental setup is shown is Figure 3.1.
The heater was connected to a digital solid state relay (SSR) of dimensions 2-1/4” x 1-3/4” x 1” having input range 3-32 VDC and output 240 VAC and 20 Amperes. SSR was connected to the SCB-68 Data Acquisition (DAQ) device by National Instruments. Two wires of a thermocouple were connected to pin #33 and #36 respectively and corresponding channels 1 and 9 on DAQ. Low level ground reference was established by adding an extra wire to reduce noise. One of the ends of which was connected to channel 33 and the other one was wired to the adjacent channel 67, the low level ground. The data acquired from DAQ was displayed, analyzed and saved on the computer using LabVIEW which provides graphical environment optimized for data acquisition. Various transducers such as pressure or temperature sensors can be connected with a DAQ device. A J-type thermocouple was used for the thesis which has an appropriate sensor to convert temperature to an electric signal.
The application software, Measurement and Automation Explorer (MAX) by National Instruments, was used to set up the thermocouple. MAX is a graphical interface installed with NI-DAQ to program data acquisition applications in LabVIEW. With MAX, NI software and hardware can be configured; new channels, interfaces and VIs can be added; and various devices and instruments connected to the system can be viewed.

A new task which is a collection of virtual channels and contains all information required to generate or acquire data from DAQ was created for the thermocouple. This was done by right clicking on data neighborhood in MAX. The input type was selected as analog and thermocouple was chosen from the drop down box. The next step was to choose physical channels and channel 1 was selected for this task. The task was created and different settings just created could be seen on the test panel of MAX. One sample was chosen to be read at a sampling rate of 1 Hertz which means the temperature signal was acquired at the frequency of 1 Hertz. The task was tested and saved under the name of ‘thermocouple J’.

A data acquisition platform was developed using LabVIEW Real-time in which the DAQ Assistant Express VI was created for continuous single point input or output using the following steps:

1. Input DAQ Assistant from Express VIs was placed on the block diagram to gather data or acquire the signals.

2. Already configured task ‘thermocouple J’ on MAX opened and could be edited by double clicking the DAQ Assistant Express VI.
3. Numeric and graphic indicators were created which were visible on the front panel.

4. Output DAQ Assistant from Express VIs was also placed and control was created for the input on the front panel.

5. A while loop was placed around it for continuous measurement.

The front panel and Block diagram are shown in Figure 3.2a and 3.2b respectively.

Figure 3.2a: Front panel showing generation of (thermocouple) data
Figure 3.2b: Block diagram panel showing generation of (thermocouple) data
Graphical and numeric indicators display the temperature in degree Celsius measured by the thermocouple connected to SCB-68 DAQ. It was found to be giving accurate results when tested by placing the other end of thermocouple in hot and cold water. The temperature is shown as 24.9°C in fig. 2a when thermocouple was kept at room temperature.

3.2 Process reaction curve

The heater was connected to a solid state relay which is like an electric switch that opens and closes when power is supplied and cut off. That makes the input to the heater digital. Since the final control element is digital, a step change would mean increasing the voltage from 0 to 1 which would turn the heater on indefinitely. Therefore pulse input was chosen instead to obtain a process reaction curve. As shown in figure 3.2a, when 0(OFF setting) is entered as input, the power sent to the relay is cut off and since relay is connected to the furnace, the furnace turns off. On the other hand, when 1(ON setting) is entered as input, power gets supplied to the relay and as a result heater turns on. The thermocouple was placed inside the isothermal zone of the furnace taking care not to touch the furnace walls to avoid unnecessary noise/disturbance.

In order to generate a Process Reaction Curve, 0 was entered as the input in the VI shown above to begin with. A pulse input was given for about 30 seconds by changing the input from 0 to 1 and back to 0. The graph thus obtained is called the Process Reaction Curve (PRC) shown in Figure 3.3.
Figure 3.3: Process Reaction curve (PRC)
3.3 FOPDT Model Fitting

FOPDT model was fitted to the response plot using Design Tools of a process control software called Control Station v3.7. Figure 3.4 shows a display of a plot that shows a model fit on top of the PRC data:

![FOPDT model fitting on PRC data](image)

Figure 3.4: FOPDT model fitting on PRC data

The first order model parameters were found to be as:

- Gain = 660.13,
- Time constant = 2030s
- Dead time = 3286s
The results obtained above were verified using $T*t$ versus $t$ plot to obtain time constant[11].

![Figure 3.5: $T*T$ versus $t$ plot](image)

From Figure 3.5, time corresponding to the highest value of $T*T$ is the time constant of the system which is about 2300 s (close to 2030s). This shows FOPDT model fitting gave correct results.

### 3.4 Cohen and Coon tuning method

In order to tune the controller, it must be first tuned to the system. The controller is synchronized with the process variable using this tuning and hence allows the process to
be kept at its desired operating condition. The initial set of controller settings for the temperature control system were chosen using Cohen and Coon method using following formulae [12-13]:

<table>
<thead>
<tr>
<th>Controller</th>
<th>$K_c$</th>
<th>$\tau_I$</th>
<th>$\tau_D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>$\frac{1}{K} \frac{\tau}{\theta} \left( 1 + \frac{\theta}{3\tau} \right)$</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>PI</td>
<td>$\frac{1}{K} \frac{\tau}{\theta} \left( 0.9 + \frac{\theta}{12\tau} \right)$</td>
<td>$\frac{30 + 3(\theta/\tau)}{9 + 20(\theta/\tau)}$</td>
<td>—</td>
</tr>
<tr>
<td>PID</td>
<td>$\frac{1}{K} \frac{\tau}{\theta} \left( \frac{4 + \theta}{3 + 4\tau} \right)$</td>
<td>$\frac{32 + 6(\theta/\tau)}{13 + 8(\theta/\tau)}$</td>
<td>$\frac{4}{11 + 2(\theta/\tau)}$</td>
</tr>
</tbody>
</table>

From FOPDT model fitting, we have:

$K = 660.13$

$\tau = 2030s$

$\theta = 32.86s$

For a PID controller, the parameters were calculated as:

$K = 0.06$

$\tau_I = 2750.68$

$\tau_D = 16.30$

PRC method was found to be not suitable for this temperature control system. This is because process gain obtained from curve fitting method is very high; Cohen and Coon settings give negligible values for P, I and D; which cannot be the used as starting values to fine tune the controller. Due to these reasons, PRC method was eliminated and trial and error method was followed to tune the PID controller.

Proportional controller works better than on-off controller by applying power to the heater which is proportional to the error (difference in heater temperature and set
point). The system responds faster to a perturbation (caused by set point or load changes) as gain is increased, but eventually becomes unstable. Integral action is introduced to minimize overshoot and eliminate offset. Due to high thermal capacity of heater, response time is usually large. Thus derivative action is added for faster response time. Thus PID controller was required for this temperature control system.

### 3.5 Temperature controller tuning

Temperature controller tuning involves setting P, I and D values to get the best possible control for a particular control system in order to achieve fastest response time and minimum overshoot. After tuning, control loop should be able to return to setpoint quickly after a set point change or load disturbance. Since the input to the heater is digital, conventional tuning techniques are unsuitable. For tuning this temperature control system, a VI was prepared in LabVIEW. The front panel of the VI is shown in figure 3.6a.

The VI shows a knob with range 20-120 to vary setpoint. Tuning parameters: P, I, D values and upper and lower limits of a controller were entered in numeric controls as shown in the figure. Integral of the Absolute value of Error (IAE), Integral of the Square of the Error (ISE) and Integral of time-weighted absolute error (ITAE) can be read as outputs. The temperature was being read by the thermocouple inserted in the heater and the value is shown as PV (process variable) on the front panel. A waveform chart shows the temperature versus time plot based on varying PV values.
Figure 3.6a shows the block diagram of the above VI. Thermocouple J was chosen as the input channel and the signal sent to ‘DAQmx Read’ to read the temperature data (PV). ‘Write LabVIEW Measurement File’ was used to save the data file on hard drive. Set point was entered on the front panel which when subtracted from PV gave error signal. The error signal was sent to the formula nodes to calculate IAE, ITAE and ISE using formulas:

\[ IAE = \int_{0}^{\infty} |e(t)| \, dt \]

\[ ITAE = \int_{0}^{\infty} t \cdot |e(t)| \, dt \]
\[ ISE = \int_{t_0}^{t_{\text{end}}} (e(t))^2 \, dt \]

Set point and tuning parameters to be entered on front panel were wired to an inbuilt VI called ‘Simple PID’ which gives out digital output. If the output value is less than 10 (chosen upper limit), heater turns off and if it is more than 10, it is sent to ‘DAQmx Write’ which writes a single Boolean sample to a digital output channel.

Figure 6b: Block diagram showing controller tuning VI

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
The above VI was tested for an electric bulb first because the system has small response time. Thermocouple was kept near the bulb to read the temperature of the surrounding air. Set point was given on the front panel by adjusting the knob and VI was run. As long as temperature (PV) stayed below the set point, PID output remained at 10V, sending power supply to bulb to keep it on. As soon as the PV crossed the setpoint, the PID output changed to zero stopping voltage supply to the bulb through solid state relay. This proved there was no error in the VI and could be used to tune electric heater.

The first step for heater tuning involved shutting off integral and derivative action. This is similar to the Ziegler and Nichols closed loop tuning method. The values of P (or Kc) were varied as 2, 4, 6, 8 and 10 to obtain IAE versus time and temperature versus time plots. Smaller values of controller gain were chosen to avoid oscillatory responses which results in longer settling times. Values smaller than 2 would not have enough strength to return the system to the power to pull the system to the set point.

The thermocouple at steady state (26°C) was given a set point of 36°C. Controller gain was entered as 2 and the VI was run. As a result, temperature started rising; overshot the set point and after reaching a peak value, returned back to the set point. As soon as it reached the set point, the VI was stopped and the values of IAE and time were noted down. Then set point was increased to 46°C and values were noted keeping P=2 constant. Five sets of experiments were run for a controller gain of 2 and IAE versus time plot was obtained. For next set of experiments, controller gain was changed to 4 and set point increased to 36°C, then 46°C and so on. Similarly, experiments were run using all other values of P and plotted together as shown in Figure 3.7. It was found that IAE was minimum for Kc= 2.
Figure 3.7: IAE versus time plot for heater temperature control

By keeping setpoint constant at 46°C, another set of experiments was performed for different values of P. Temperature versus time plot thus obtained is shown in Figure 3.8.
The same result was obtained here. It was found that response time was minimum for a controller gain of 2 as shown by the blue line in the figure.

For next set of experiments, $K_c$ was kept constant at 2 and integral time ($\tau_i$) varied from 100 to 20. $K_c$ when divided by $\tau_i$ gives $I$. The plot is shown in Figure 3.9. The best $P$, $I$ setting was found to be $P = 2$, $I = 0.02$ based on minimum response time as shown by blue line in the Figure.
In the final step, P and I were kept constant at 2 and 0.02 respectively and D varied as 0.5, 2.5, 4.5, 6.5, 8.5. The best controller setting was found to be $P = 2$, $I = 0.02$ and $D = 0.5$.
CHAPTER IV

LIQUID LEVEL CONTROL

Liquid level control is one of the most common types of control in use in industries. Level measurement is important to determine the amount of liquid in a reactor or storage vessel to regulate a process or control inventory. In continuous level monitoring, the sensor monitors the liquid level and displays the actual level of liquid as it changes. While point-level monitoring is simpler to control the liquid level in a tank and involves measurement of liquid level at specific points within the tank, it however is less accurate. Liquid level control is required in different industries from wineries to food industries and aeronautical to pharmaceutical industries.

As explained in section 2.2 for liquid level process, transfer function relating height \( h \) of the tank and inlet volumetric flow rate \( q(t) \) is given by:

\[
\frac{H(s)}{Q(s)} = \frac{R}{(\tau s + 1)} \text{ where } \tau = AR
\]

Controller tuning involves a selection of controller parameters such that the control system performance specifications are met. There are numerous open loop and closed loop methods available for controller tuning. Tuning using error-integral criteria was studied in this section which can be directly applied to MIMO systems and gives improved results over traditional tuning methods [14]. The tuning procedure used here is based on dynamic performance criteria for controller tuning in which the error is a
function of time for the duration of the response unlike simple performance criteria. In this method, the sum of the error at each point of time (IAE, ITAE or ISE) is minimized.

In this thesis, liquid level control in a storage tank was investigated using a PI controller.

4.1 Experimental set up

![Liquid level control system set up](image)

Figure 4.1: Liquid level control system set up

A tank A of diameter 18” and height 41” was used for the liquid level control experiments shown in Figure 4.1. Water is pumped to the tank through a research control valve ordered from Badger Meter Inc. The pressure to the valve has 3-15psig range of operation. Water is drained out using a non linear valve in a smaller tank B. The water from a tank B is recycled back using a pump to tank A through a control valve. The level sensor used in the bottom of the tank is a piezoelectric device which outputs an electric
voltage in proportion to the applied pressure by water head in the tank A. The transducer used can measure pressure up to 5psi (27.684*5 inches water column) and convert into voltage in the range of 1-6V. The actual voltage observed when the tank was empty is 1.17V.

The signal from the sensor is sent to a Data Acquisition Device (SCB-68) where acquired data is displayed, analyzed and saved on the computer using a program in LabVIEW. The data from DAQ enters a controller (in the computer) where it is compared to the set point to produce an error signal. The error signal is sent to transducer to convert electric signal to pneumatic (pressure) signal in the range of 3-15 psig which is a linear function of the input. Finally, the output from the transducer is sent to the top of control valve, which adjusts the inlet flow rate of water going in the tank to keep the level at a desired point. Each component is supplied with an external power (120 V).

The application software, Measurement and Automation Explorer (MAX) by National Instruments, was used to set up the level sensor. Two virtual channels- input and output channels were configured in MAX named PT and PTout respectively. The voltage range for both the channels was entered as 0-5V. A new scale (PTscale) was created in MAX and conversion constants entered to get appropriate units.

In this thesis, inlet flow rate to the tank was manipulated to control the level of the tank using a PI controller keeping outlet flow rate at a constant value. The starting parameters for PI controller were determined from step response curve obtained by introducing a step change in inlet flow rate by changing transducer voltage from 4 to 5 V. The PI controller was fine tuned using trial and error tuning method.
4.2 Process Reaction Curve

A step change was introduced in the inlet flow rate of water going into the tank using a control valve by changing the voltage of the transducer. Voltage was increased from 0 to 4V and a height versus time plot obtained when height reached a steady state value. Similarly, another plot was obtained by increasing the voltage from 0 to 5V. Steady-state gain (K) was calculated which is a ratio of percentage change in output to percentage change in input. Figures 4.2a and 4.2b show the front panel and block diagram of a program in LabVIEW created to obtain step response curve using open loop response. The valve was fully closed at voltage of 0V and fully open at 5V. At voltage of 4V, valve was 80% open.

Figure 4.2a: Front panel of VI to obtain step response curve
DAQ Assistant is an express VI which when placed on block diagram launches to create a new task. The task was edited by double clicking on it; one sample was made to be read at the rate of one hertz and acquisition mode was selected to be continuous. The voltage signal (volts) was converted to height in inches using conversion constants and output was obtained using graphical and numerical indicators. 'Write LabVIEW Measurement File' was used to download data on hard drive. Voltage in the range of 0-5V was entered as input and wired to DAQ Assistant2.

Figure 4.2b: Block diagram of VI to obtain step response curve
Figure 4.3 shows the step response curve obtained for a step change of magnitude of 4 and 5 units.

Figure 4.3: First order step response curve

Steady state gain, $K = (\% \text{ change in output})/(\% \text{ change in input})$

$$K = (10/16)/(80/100)$$

$$K = 0.78$$

Response time, $\tau = 372s$

Dead time, $t_0 = 20s$

The FOPDT model can be represented as:

$$\frac{H(s)}{Q(s)} = \left\{ \frac{1}{372s + 1} \right\} e^{-20s}$$

Thus process parameters were calculated based on Process reaction curve. Using Cohen and Coon formulas for a PI controller,

$$K_c = 21.63; \tau_1 = 60 \ (I = 0.36)$$

75
4.3 Compact FieldPoint (cFP)

4.3.1 Introduction

Compact FieldPoint from National Instruments is an easy to use embedded controller for reliable and stand alone operation. It is composed of rugged analog and digital I/O modules and many communication interfaces. Any kind of sensor can be directly connected to I/O modules and data can be acquired and displayed on a computer using real time LabVIEW applications. Compact FieldPoint I/O modules refine, calibrate and spread raw sensor signals to engineering units and analyze to look for problems, such as an open thermocouple. Compact Fieldpoint network communication interfaces publish the data on web making it accessible to everyone with an Ethernet network. The most remarkable feature of Compact FieldPoint is that it is most suitable for stand alone operation and does not require direct connection to monitor or other human interface to monitor and control the system. It is highly reliable and can withstand high temperatures (-25 to 60 °C) and tolerate shock up to 50g [5].

cFP-2010 from National Instruments was set up with the following features:

1. Embedded web with remote-panel user interface
2. Runs LabVIEW Real-Time for control and signal processing
3. 32 MB RAM memory
4. Two RS232 serial ports for connection to peripherals
5. Stand-alone embedded real-time controller
6. Removable 64 MB nonvolatile storage
7. Heavy industrial electrical immunity
External power in the range of 11 to 30 VDC power supply connected to cFP-2010 filters and distributes power to the bank of up to eight I/O modules that it controls and handles up to 100Mb/s data communication rates. Each bank can be made accessible to unlimited number of host computers forming a distributed computing system. LabVIEW application can be downloaded on cFP-2010 to perform many applications like stand-alone operation, data logging, in control systems such as to run PID loops, actuate pumps and valves, perform real time analysis and simulation, take measurements and communicate over the ethernet.

Figure 4.4 is a picture of a compact FieldPoint showing eight I/O modules connected with steel screw fasteners. Fieldpoint was mounted directly on the solid metal panel called backplane.

Figure 4.4: NI Compact FieldPoint cFP-2010 [5]
4.3.2 Configuration

cFP-2010 network module was connected to an ethernet network using an ethernet cable. Fieldpoint Explorer, the software driver for FieldPoint hardware, was launched and ‘Configuration device’ was chosen. System configuration dialog box opened and IP address, subnet mask, gateway and DNS address values were entered on Network settings tab. Host name ‘vasud’ was entered on the system tab and with this, FieldPoint was configured.

Next task was to configure the devices and channels in order to be able to locate Fieldpoint units, read data from input devices and send data to output devices. This was done using National Instruments Measurement and Automation Explorer (MAX). Figure 4.5 shows the MAX window after expanding the ‘Data Neighborhood’ item to locate FieldPoint items. The figure shows ‘vasud’ installed at IP address 132.177.88.200. This unit was connected to the computer in process control lab to exchange data between the computer and the FieldPoint unit. ‘vasud’ unit is expanded in the figure to show four interface modules. Each analog module contains 8 channels and each digital module contains 16 bits.

The installation and configuration was verified by monitoring an I/O channel. The thermocouple was connected to one of the channels of an analog module shown as cFP-TC-120 in the figure. cFP-TC-120 was as selected and ‘start’ button on the top of window was clicked to read the data from channel ‘1’ where thermocouple was connected. It was found to give satisfactory results.
### Attribute | Value | Comment
--- | --- | ---
Comm Resource Name | vasud | The name of the selected comm resource
Type | Ethernet (PP-2010) | The type of comm resource
Physical Comm Resource Name | 132.177.88.200 | The name/IP address of the corresponding physical device under Devices & Interfaces

Figure 4.5: Measurement and Automation Explorer window
4.3.3 Data acquisition using cFP-2010

When FieldPoint software was installed, a library of FieldPoint VIs was created which included VIs like FP Read, FP Write etc. The same set of VIs was used to access the cFP-2010 from the host PC and to run embedded applications. A simple VI was created to read temperature of a thermocouple connected to channel 1 on connector block of cFP-TC-120 of Fieldpoint. Figure 4.6a and 4.6b show the front panel and block diagram respectively.

Figure 4.6a: Front panel showing data acquisition using compact FieldPoint
Figure 4.6b: Block diagram showing data acquisition using compact FieldPoint

‘FieldPoint IO Point’ was dragged from Functions palette on the block diagram to specify the FieldPoint tag to be used for IO operations. I/O item needed to be accessed was chosen from drop down box which shows three string names – host name (vasud), the device (cFP-TC-120) and the I/O item (thermocouple) as shown in Figure 4.6b. ‘FP Read.vi’ was used to monitor a thermocouple. Delay in the loop was added to the VI to control the loop execution rate.

FieldPoint has the unique ability to share the LabVIEW front panel over the net using web publishing tool. Remote panels can control LabVIEW applications over the Web through a standard Web browser. LabVIEW front panel can be viewed and controlled simultaneously by several web browser clients.
4.3.4 Targeting LabVIEW RT to the FP-2010

When LabVIEW Real Time (RT) engine is run on an FP-2010, the module can run applications without input from the host computer. A separate host PC can be used to control the FP-2010 through an ethernet connection. When FP-2010 is selected as a target platform, LabVIEW RT application gets downloaded on the FP-2010 and can be run from there.

A FieldPoint project was created which contained inbuilt host and target VIs which could be modified as required. To create a new project, new Real-Time project was opened from the starting window for LabVIEW. ‘Application includes deterministic components’ was unchecked to avoid restriction of options on the next page. A project name ‘level.lvproj’ was given and target configuration was chosen. ‘Include user interface’ was checked to select host VI to communicate with target. Selected target was browsed and ‘vasud’ was found under ‘Existing target or device’ in Real-Time FieldPoint. Hence level project was configured. Figure 4.7 shows the top half of the screen after expanding ‘vasud’ item to locate fieldPoint items.
The target.vi and host.vi were modified to read the level changes in the tank. The level sensor was connected to Fieldpoint to channel 0 of cFP-AI-110. Timed loop was placed on the block diagram of Target VI which executes each iteration of the loop at the specified period. ‘Period’ specifies the amount of time elapsed between two consecutive iterations of the Timed loop. The FieldPoint tag was specified in ‘FieldPoint IO Point’ as shown in Figure 4.8 and sent to ‘FP Read’. A shared variable was dragged from the Project Explorer window to create a Shared Variable node. Shared variable was double
clicked to choose ‘Array of double’. The block diagram of Target VI is shown in Figure 4.8.

![Block Diagram of Target VI](image)

Figure 4.8: Target.vi of level Project Explorer

Host VI of level Project Explorer is shown in Figure 4.9. Acquired data from target.vi using shared variable was connected to a numeric indicator for the data to be read from the front panel.
An analog electrical transmission standard for industrial instrumentation and communication is 4-20mA. The signal is a current loop where zero percent signal is represented by 4mA and 100% signal is represented by 20mA. However it was found that the output signal from I/O modules of compact FieldPoint was 10mA. A signal of 12mA sent to transducer converted to a pressure of maximum 9 psig resulting in opening the valve to only half of its full capacity. It was desired to use compact FieldPoint in the thesis for PI controller tuning. But due to this limitation, FieldPoint could not be used. This problem has now been solved with the acquisition of data from a traditional SCB-68 DAQ from NI.

4.4 Level controller tuning

Next step was to create a LabVIEW VI for level controller tuning. Figure 4.10a shows the front panel of the VI. Starting P, I, D parameters and set point were entered as input
whereas PID output and height versus plot were obtained as outputs on the front panel. Loop delay (in milliseconds) was entered to control the loop execution rate.

Figure 4.10b shows the block diagram of the above VI. ‘AI Read One Scan’ was used to measure the signal on the specified channel (PT) and return the measurement of height (PV) in the form of an array of binary values. ‘Write LabVIEW Measurement file’ was used to download the data on hard disk. Set point was entered on the front panel which when subtracted from PV gave error signal. The error signal was sent to the formula nodes to calculate Integral of the Absolute value of Error (IAE), Integral of the Square of the Error (ISE) and Integral of time-weighted absolute error (ITAE). ‘Elapsed Time’ indicates the amount of time elapsed since the specified start time. Set point and PID parameters were sent to a VI called ‘PID.VI’ to obtain PID output which was wired to ‘AO Write One Update’ to write a single voltage value to the specified analog channel (PTout). All objects were placed in a while loop for continuous measurement.
Figure 4.10a: Front panel for level controller tuning
The first step for level controller tuning was shutting off integral and derivative action. This is similar to the Ziegler and Nichols closed loop tuning method. Based on starting values obtained from Cohen and Coon formulae, the values of P (or Kc) were varied as 20, 22, 24, 26, and 28 to obtain IAE versus time plot as shown in Figure 4.11. It was found that IAE was minimum for P = 24.

Figure 4.10b: Block diagram for level controller tuning
For the next set of experiments, $P$ was kept constant at 24 and $I$ varied as 0.25, 0.35, 0.45 and 0.55. The best $P$, $I$ setting was found to be $P = 24$, $I = 0.35$ from Figure 4.12.
CHAPTER V

SUMMARY AND FUTURE WORK

5.1 Summary

The thesis was separated into three sections. The first one involved working on NI LabVIEW, a very useful software to study dynamic behavior of first and higher order systems. Simulations in LabVIEW were created to observe time and frequency response of various systems based on their transfer functions. Single-input single-output (SISO) and multiple-input multiple-output (MIMO) systems were analyzed and control loop interactions were estimated from dynamic RGA calculations for MIMO systems. A LabVIEW program was created to determine Z-N settings to be used as starting values for controller tuning.

In the second section of thesis, control aspects of a home built temperature controller were investigated. Process parameters calculated from FOPDT model fitting on PRC data were obtained as: Gain = 660.13, Time constant = 2030s, Dead time = 3286s. Digital solid state relay was used to provide power supply to the heater. Results obtained from Cohen and Coon calculations for a heater showed that Process Reaction Curve method cannot be used to find the starting P, I, D values for tuning a controller. It was decided to use trial and error method to tune the PID controller. Small values of P, I, D were chosen as the starting values to avoid oscillatory responses which result in longer settling times. The experiments were run and the best PID values were obtained as: P = 2,
I = 0.02, D = 0.5 for the liquid level experiment.

In the third section, liquid level control was achieved using Process Reaction Curve method in which step input was given by changing the voltage of transducer from 0 to 4V. Process parameters were calculated as: Gain = 0.78, Time constant = 372s, Dead time = 20s. Cohen and Coon calculations were performed to find the initial set of P, I values. It was desired to use compact FieldPoint developed by NI to fine tune the level controller. However it could not be used due to output limitations. Hence, experiments were run using the traditional data acquisition device to acquire data and tuning method involving error-integral criteria was used. First step of Ziegler Nichols Tuning method was used for the level controller tuning. Most optimum values of P, I were obtained as P = 24, I = 0.35.

5.2 Recommendations

Based on the results of the study, several recommendations regarding further study can be made. More work is needed to fully understand the applications of all other functions on Functions Palette in LabVIEW that have not been used here. Analog solid state relay should be used to see its effect on process gain and hence temperature controller tuning. A new I/O module for FieldPoint was ordered from National Instruments. Experiments should be run using the same I/O module to compare the results with those obtained from traditional data acquisition device.
ABBREVIATIONS

K Process gain
τ Time constant
θ Dead time
ζ Damping factor
ω Frequency
ω_c Crossover frequency
P or K_c Controller gain
I Ratio of P to integral time (τ_i)
D Product of P and derivative time (τ_D)
K_u Ultimate gain
P_u Ultimate Period
IAE Integral of the Absolute error
ITAE Integral Time Average of the error
ISE Integral of the square of the error
DAQ Data Acquisition
VI Virtual Instrument
MAX Measurement and Automation Explorer
PV Process variable
RGA Relative Gain Array
LT Level sensor/transmitter
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LC</td>
<td>Level Controller</td>
</tr>
<tr>
<td>cFP</td>
<td>Compact FieldPoint</td>
</tr>
<tr>
<td>I/O</td>
<td>Input/output</td>
</tr>
<tr>
<td>TC</td>
<td>Thermocouple</td>
</tr>
<tr>
<td>RT</td>
<td>Real-time</td>
</tr>
</tbody>
</table>
REFERENCES


