A DESIGN STUDY OF HEAT EXCHANGER PROCESS CONTROL BETWEEN CONVENTIONAL AND MAMDANI'S FUZZY LOGIC CONTROLLERS

by

KHAIRUL ANWAR BIN HJ. IDRIS

Dissertation submitted in partial fulfilment of the requirements for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

DECEMBER 2004

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

A DESIGN STUDY OF HEAT EXCHANGER PROCESS CONTROL BETWEEN CONVENTIONAL AND MAMDANI'S FUZZY LOGIC CONTROLLERS

by

KHAIRUL ANWAR BIN HJ. IDRIS

A project dissertation submitted to the
Electrical & Electronics Engineering Program
Universiti Teknologi PETRONAS
in partial fulfillment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:

Miss Suhaila Badarol Hisham
Project Supervisor

DECEMBER 2004

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

Khairul Anwar bin Hj. Idris

DECEMBER 2004

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

ACKNOWLEDGMENTS

In the name of ALLAH, the Most Graceful and Most Merciful,

First, I would like to express my greatest gratitude to both my supervisors, Mr. Rosdiazli Ibrahim and Miss Suhaila Badarol Hisham for their expert guidance, attention and suggestions, support and advices pertaining to the project and the difficulties faced during the project execution.

I would also like to thank several lecturers who have also helped me during the project, especially Mr. Zuhairi Baharudin, who helped me in the development of the fuzzy membership functions and fuzzy rules and Dr. Nordin Saad who helped me with the process control.

Special thanks to the Instrumentation and Control lab technician, Mr. Azhar bin Zainal Abidin, for his guidance, support and concern during the project.

To my family, love and thank you. Without your enormous support and concern, all my effort in preparing this final year project would have not been successful.

Last but not least, my appreciation goes out to individuals or groups that have helped me in any possible way to complete this project.

Above all, I would like to thank God for making it possible for this project to be completed.

ABSTRACT

A heat exchanger is a piece of equipment that continually transfers heat from one medium to another in order to carry process energy. In order to ensure its smooth operation, modeling and simulation of the system can be made so that its performance can be analyzed and improved. The scope of this study is more on simulation and software implementation of the control system design by using MATLAB. The main issue tackle in this study is to improve the performance of the heat exchanger process control. In this study, the heat exchanger is modeled using an empirical model to simulate the heat exchanger temperature response. A controller is then designed for the process using two approaches, one using a conventional PI method and another based on a fuzzy logic controller employing Mamdani inference method as an alternative approach. From the results obtained, it has been proven that both controllers are proven stable with good output temperature response. The responses of both controllers are further scrutinized where the fuzzy logic controller is shown to have better control performance compared to the PI controller. As a conclusion, intelligent control is better than the conventional PID control.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

PID – Proportional, Integral & Derivatives

UTP - Universiti Teknologi PETRONAS

.SP - Set point

PV - Process variable

MV - Manipulated variable

FLC - Fuzzy logic controller

FIS - Fuzzy inference system

MF - Membership function

GUI - Graphical user interface

TT - Temperature transmitter

CHAPTER 1
INTRODUCTION

1.1 Background of study

Process control plays an essential role in the safe manufacture of quality products at market demand, while protecting the environment. Flow rates, pressures and temperatures within pipes and vessels, inventories of liquids and solids, and product quality are all examples of measured variables that must be controlled to meet the above objectives.

In UTP, there are several pilot plants available for students to explore the area of process control. The pilot plants closely resemble actual plants in a smaller scale complete with relevant field instrumentations. For this study, the author had utilized Plant 6: Drum-Heat Exchanger Process Pilot Plant available in UTP Instrumentation & Control System Lab. In general, this project aims to model and simulate the heat exchanger so that the model can be used for the plant performance analysis.

In this study, the main issue tackle is to improve the performance of the heat exchanger process control by developing another controller approach other than PID controller and the author choose to use an intelligent approach; a fuzzy logic controller. Finally, process responses from each controller will be compared against each other in terms of their control performance.

1.2 Problem statement

1.2.1 Problem identification

At the start of the project, there was no model describing the heat exchanger pilot plant available in UTP that can be used by the student for plant performance analysis. A model here refers to a suitable mathematical description of the plant parameters. By conducting this project, a model can be developed and it can be used for further analysis, particularly for the optimization of the heat exchanging process.

A significant research had been conducted to improve the performance of heat exchanger and its corresponding process control. Thus, this study can be used to find an alternative controller design other than PID controller in order to improve the performance of the heat exchanging process.

1.2.2 Significance of the project

PID controllers are widely used in most industrial processes. However, it is difficult to find an optimal set of PID gains for a particular system. An intelligent control application such as fuzzy logic can help to control non – repeating or unpredictable systems and it is developed to resemble human reactions and consequently further improves the process response.

1.3 Objectives and scope of study

1.3.1 Objectives

- To model and simulate the heat exchanger pilot plant
- To design PID and Mamdani's fuzzy logic controllers for the heat exchanger process
- To analyze the performance of both controllers
- To make an investigative and comparative study between the conventional PID control versus the intelligent fuzzy logic control

1.3.2 Scope of study

The modeling and simulation was done on Plant 6: Drum-Heat Exchanger Process Pilot Plant. The study was done based on input and output of the heat exchanger together with its controller action. The identification of the system study such as the transfer function needs to be obtained for the implementation of this project. Thus, experiment was done to obtain the plant identification. The process reaction curve was used to obtain the plant parameters. In this project, the scope of the analysis was on the first-order-with-dead-time model. Then, modeling was done to develop the heat exchanger process plant model using the empirical modeling. The accuracy of the model was observed based on its output reaction to input variation. The model developed was validated using the MATLAB software to verify that the parameters obtained and calculated are correct in the real application.

1.3.3 The Relevancy of the Project

Currently, significant research has been conducted to improve the performance of heat exchanger and its corresponding control system. The overall performance of a heat exchanger depends on the design and specification of the exchanger being used. Thus, it is imperative to develop the best control system in order to optimize the performance of the heat exchanger. This project aims to achieve this particular objective in optimizing the heat exchanger performance.

The outcome of this project is very promising in terms of future development of a new breed of process controllers. Testing the existing PID controller and redesigning it using the proposed fuzzy logic technology is very useful in providing better controller performance. In summary, this project can be considered as an enhancement step to plant process control.

CHAPTER 2
LITERATURE REVIEW & THEORY

2.1 Heat exchanger

[1]The natural laws of physics always allow the driving energy in a system to flow until equilibrium is reached. Heat leaves the warmer body or the hottest fluid, as long as there is a temperature difference, and will be transferred to the cold medium. A heat exchanger follows this principle in its endeavor to reach equalizations. The theory of heat transfer from one media to another, or from one fluid to another is determined by several basic rules.

- Heat will always be transferred from a hot medium to a cold medium.
- There must always be a temperature difference between the media.
- The heat lost by the hot medium is equal to the amount of heat gained by the cold medium, except for losses to the surroundings.

[2]Heat exchanger is a piece of equipment that continually transfers heat from one medium to another in order to carry process energy. It is where two or more fluids that don't physically touch each other but a transfer heat or energy take place between them. A type of heat exchanger widely used in the chemical-process industries is that of the shell and tube arrangement as shown in *Figure 1*. One fluid flows on the inside of the tubes, while the other fluid is forced through the shell and over the outside of the tubes.

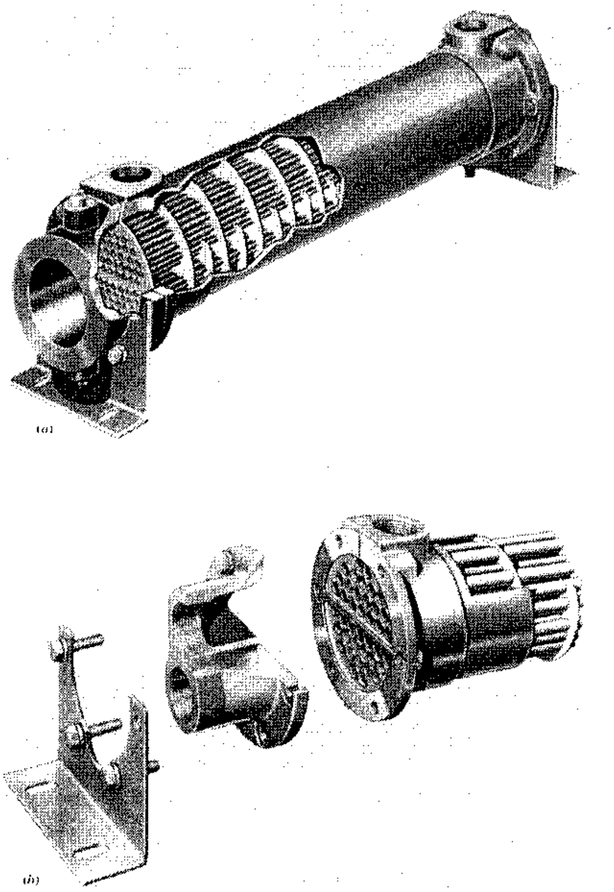


Figure 1 (a) Shell-and-tube heat exchanger with one tube passes. (b) Head arrangement for shell-and-tube heat exchanger with two tube passes. (Young Radiator Company.)

In a heat exchanger, the liquid flows through the inner tube and it is heated by another liquid that flows co-currently around the tube as shown in Figure 2 below. The temperature and the flow rate of the liquid not only change with time but also change along the axial direction x.

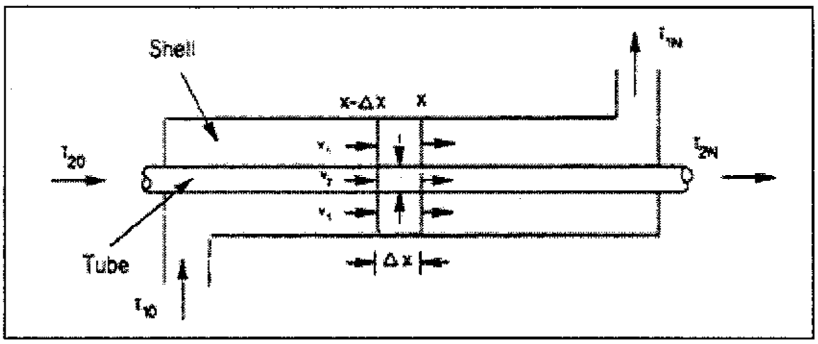


Figure 2 Co-current Shell and Tube Heat Exchanger

To model the heat exchanger, several assumptions are made:

a) The physical and chemical properties of the fluids under consideration should be constant,
b) The variation in fluid velocity and temperature radially is negligible,
c) No significant heat transfer to the surroundings and,
d) Overall heat transfer co-efficient must be constant.

2.2 Plant 6: Drum-Heat Exchanger Process Pilot Plant

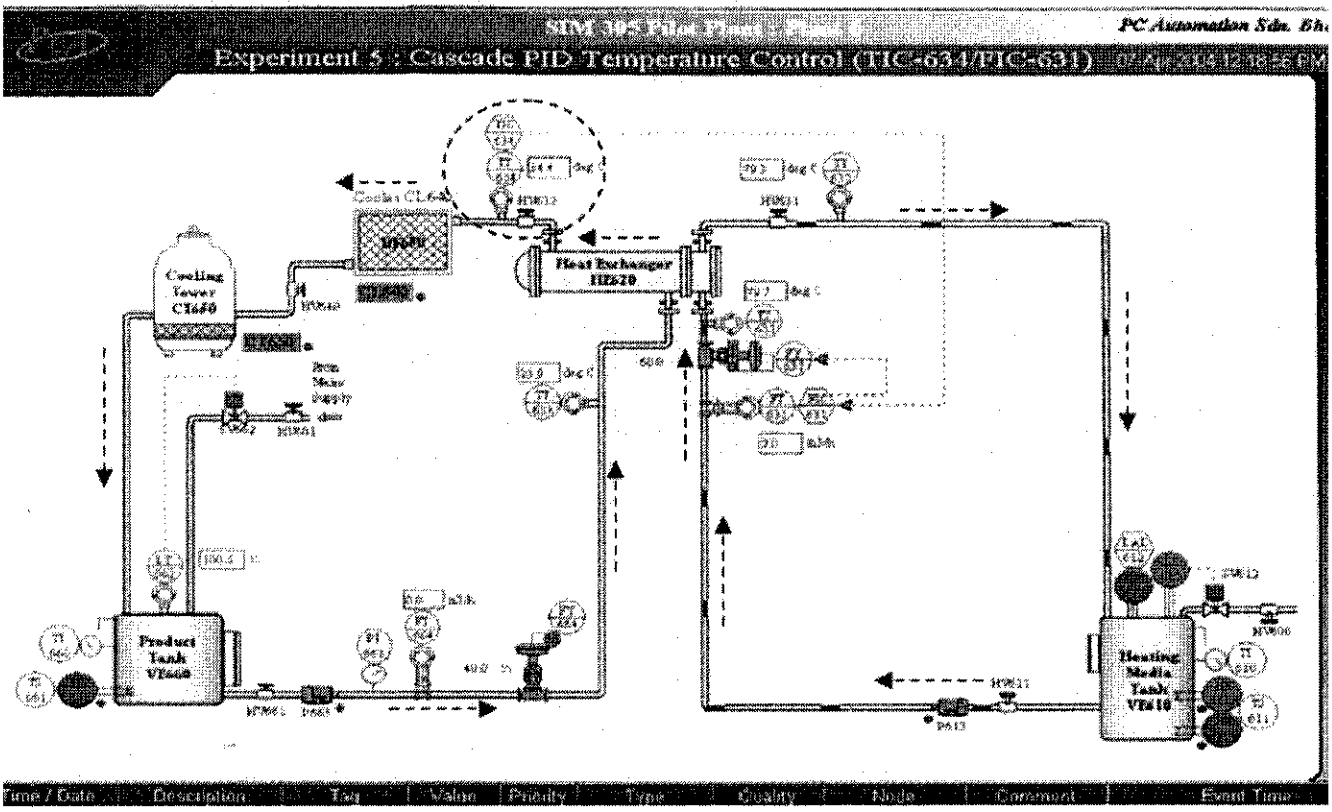


Figure 3 Plant 6: Drum-heat exchanger process pilot plant

Legend: Water flow ------▶

Figure 3 shows the 'Plant 6: Drum-heat exchanger process pilot plant' used for this study. Heated water flows from the heating media tank to the heat exchanger [*refer to the dashed lines*]. The heated water will then be used to increase the temperature of the liquid product through a heat transfer process in the heat exchanger. The temperature of the product is monitored by a temperature transmitter [*in the circle*], which will give a feedback input to the controller on the actual process temperature.

2.3 PID control

[10]PID algorithm is the most popular and successful feedback controller used within the process industries for over 50 years. It is a robust and easily understood algorithm that can provide excellent control performance despite the varied dynamic characteristics of a process plant. As the name suggests, the PID algorithm consists of three basic modes, the Proportional mode, the Integral and the Derivative modes.

2.3.1 A Proportional algorithm

The mathematical representation is,

$\frac{mv(s)}{e(s)} = k_c$ (Laplace domain) or $mv(t) = mv_{ss} + k_c e(t)$ (time domain)

The proportional mode adjusts the output signal in direct proportion to the controller input (which is the error signal, e). The adjustable parameter to be specified is the controller gain, k_c. This is not to be confused with the process gain, k_p. The larger k_c the more the controller output will change for a given error. For instance, with a gain of 1 an error of 10% of scale will change the controller output by 10% of scale.

The time domain expression also indicates that the controller requires calibration around the steady-state operating point. This is indicated by the constant term MV_{ss}. This represents the 'steady-state' signal for the MV and is used to ensure that at zero error the CV is at set point. In the Laplace domain this term disappears, because of the ‘deviation variable’ representation.

A proportional controller reduces error but does not eliminate it (unless the process has naturally integrating properties), i.e. an offset between the actual and desired value will normally exist.

2.3.2 A proportional integral algorithm

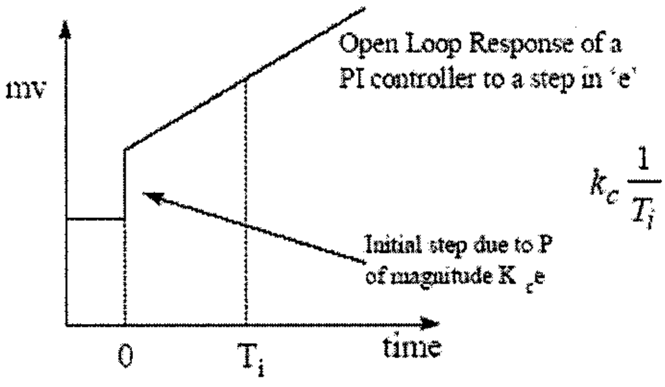
The mathematical representation is,

$$\frac{mv(s)}{e(s)} = k_c\left[1+\frac{1}{T_i s}\right] \text{ or } mv(t) = mv_{ss} + k_c\left[e(t)+\frac{1}{T_i}\int e(t)dt\right]$$

The additional integral mode (often referred to as reset) corrects for any offset (error) that may occur between the desired value (set point) and the process output automatically over time. The adjustable parameter to be specified is the integral time (Ti) of the controller.

Where does the term reset come from?

Reset is often used to describe the integral mode. Reset is the time it takes for the integral action to produce the same change in MV as the P modes initial (static) change. Consider the following *Figure 4*,



$$k_c\frac{1}{T_i}\int edt \;= k_c e$$

Figure 4 The response of a PI algorithm to a step in error

Figure 4 shows the output that would be obtained from a PI controller given a step change in error. The output immediately steps due to the P mode. The magnitude of the step up is $K_c e$. The integral mode then causes the MV to 'ramp'. Over the period time 0 to time T_I the MV again increases by $K_c e$.

2.3.3 A Proportional Integral Derivative algorithm

The mathematical representation is,

$$\frac{mv(s)}{e(s)} = k_c\left[1+\frac{1}{T_i s}+T_D s\right] \text{ or } mv(t) = mv_{ss}+k_c\left[e(t)+\frac{1}{T_i}\int e(t)dt+T_D\frac{de(t)}{dt}\right]$$

Derivative action (also called rate or pre-act) anticipates where the process is heading by looking at the time rate of change of the controlled variable (its derivative). T_D is the 'rate time' and this characterizes the derivative action (with units of minutes). In theory derivative action should always improve dynamic response and it does in many loops. In others, however, the problem of noisy signals makes the use of derivative action undesirable (differentiating noisy signals can translate into excessive MV movement). Derivative action depends on the slope of the error, unlike P and I. If the error is constant derivative action has no effect.

2.4 Fuzzy control

2.4.1 Introduction

[13]The fuzzy set and logic theory, the basis of fuzzy logic, was developed by Professor Lotfi Zadeh of University of California Berkeley in 1965. His remarks on the problem of multi valued logic: 'As the complexity of a system increase, our ability to make precise and significant statements about its behavior diminishes until a threshold is reached beyond which precision and significance (or relevance) become almost mutually exclusive characteristics'. A corollary principle may be stated succinctly as, 'the closer one looks at a real-world problem, the fuzzier becomes its solution.'

2.4.2 Fuzzy logic

[6]Fuzzy logic is an innovative approach to help control non-repeating or unpredictable systems with accuracy. It uses a list of rules rather than complicated mathematical expressions. These rules are modeled after rational decisions previously made by humans in unpredictable situations. Therefore, fuzzy logic more closely approximates human thought process than standard PID control methods do. Since some process control systems are difficult to control using only PID, the addition of fuzzy logic provides an excellent solution.

Fuzzy logic is a continuous logic pattern after the approximate reasoning of human beings. As a theoretical mathematical discipline, fuzzy logic is designed to react to continuously changing variables and challenge traditional logic by not being restricted to the conventional binary computer values of 0 and 1. Instead, it allows for partial and multi-valued truths. This discipline is especially advantageous for problems that cannot be easily represented by mathematical modeling because data is either unavailable, incomplete, or the process is too complex. The real-world language used in fuzzy control allows programmers to incorporate the ambiguous, approximate nature of human logic into computers. The use of linguistic modeling – instead of mathematical modeling – greatly enhances system transparency and modification potential. It leads to quick development cycles, easy programming and accurate control.

2.4.3 Fuzzy sets

[4]Fuzzy set is a range of values. Each value has a grade of membership between 0 and 1. Logic Boolean expressions define values as either true or false. Fuzzy logic uses linguistic variables such as "moderate", "somewhat", and "a little" to express degrees of intensity. This is illustrated in *Figure 5*. The figure on the left is a Fuzzy membership and figure on the right is a Boolean set. Actually, a fuzzy set is given by its membership function. The value of this function determines if the element belongs to the fuzzy set and in what degree.

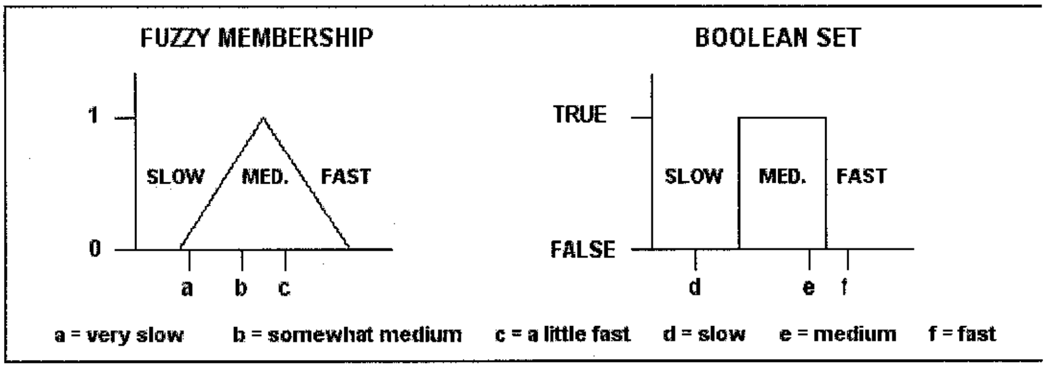


Figure 5 An example of fuzzy membership graph

2.4.4 Fuzzy control

[3][5][6][11]In control applications, fuzzy logic is used to devise a control strategy using everyday spoken language. The goal of any control strategy is to obtain a desired output, like crane motor power, from given inputs such as crane position or load angle. Because cranes cannot interpret linguistic concepts, two-way translations between crisp values and linguistic concepts are necessary. Thus, a fuzzy logic process controller is created in three steps as shown in *Figure 6*.

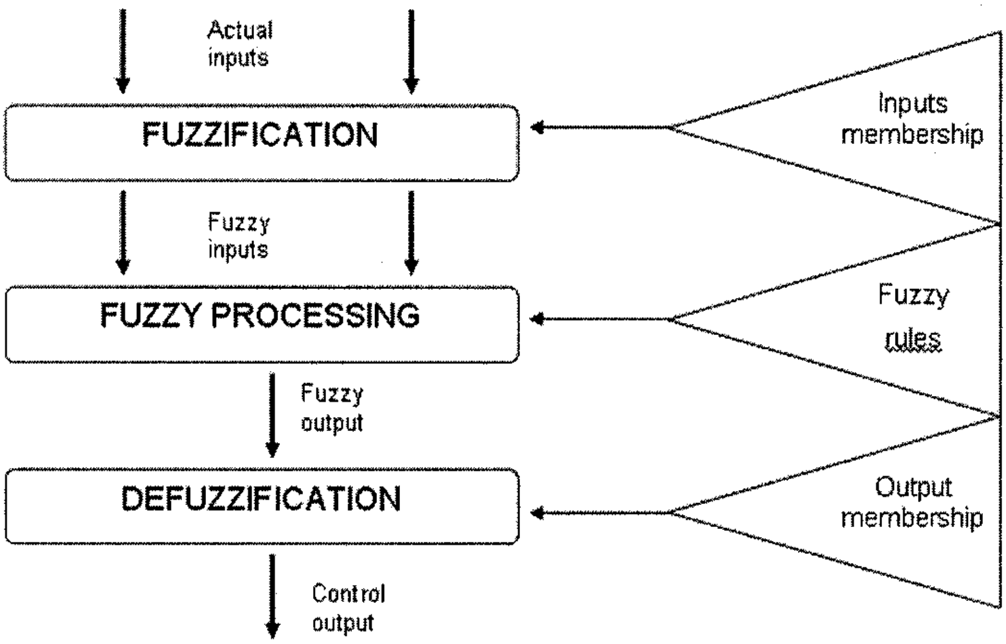


Figure 6 The fuzzy logic process controller sequence

- Fuzzification

Fuzzification is a mathematical procedure for converting an element in the universe of discourse into the membership value of the fuzzy set. Crisp input values are translated into linguistic concepts, which are represented by fuzzy sets. These concepts are called linguistic variables. Degrees of membership for all input values are assigned.

- Fuzzy processing

Fuzzy processing uses fuzzy rules which are linguistic *IF-THEN* statements involving fuzzy sets, fuzzy logic and fuzzy inference. These *IF...THEN* rules that define the relationship between the linguistic variables. These rules determine the course of action that the controller must follow.

In this study, the emphasis is on Mamdani fuzzy rules and a general Mamdani fuzzy rule can be expressed as

$$IF\ v_1\ is\ \widetilde{S}_1\ AND \ldots AND\ v_M\ is\ \widetilde{S}_M\ THEN\ z_1\ is\ \widetilde{W}_1, \ldots, z_P\ is\ \widetilde{W}_P$$

where v_i, $i = 1 \ldots M$ is an input variables and z_j, $j = 1, P$ is an output variable. $\widetilde{S}_i$ is an input fuzzy set and $\widetilde{W}_j$ is an output fuzzy set.

Fuzzy inference is sometimes called a fuzzy reasoning or approximate reasoning. It is used in a fuzzy rule to determine the rule outcome from the given rule input information. Fuzzy rules represent control strategy or modeling knowledge / experience. When specific information is assigned to input variables in the rule antecedent, fuzzy inference is needed to calculate the outcome for input variables in the rule consequent.

For the general Mamdani fuzzy rule above, the question about fuzzy inference is the following: given $v_i = \alpha_i$ for all *i*, where α_i are real numbers, what should z_j be? For fuzzy control and modeling, after fuzzifying v_i *at* α_i and applying fuzzy logic AND operations on the resulting membership values in the fuzzy rule, we attain a combined membership value, μ, which is the outcome for the rule antecedent. Then the question is how to compute *"THEN"* in the rule. Calculating *"THEN"* is called fuzzy inference. Specifically, the question is: given μ, how should z_j be computed? Definitions for Mamdani minimum inference method is, R_M: $\min(\mu, \mu_{\widetilde{W}}(z))$, *for all z*. Where $\mu_{\widetilde{W}}(z)$ is the membership function of fuzzy set $\widetilde{W}$ representing $\widetilde{W}_j$ in the rule consequent, whereas μ is the final membership yielded by fuzzy logic AND operators in the rule antecedent.

- Defuzzification

Defuzzification is a mathematical process used to convert a fuzzy set or fuzzy sets to a real number. The result of the fuzzy inference is retranslated from a linguistic concept to a crisp output value. After all, actuators for control systems can accept only one value as their input signal, whereas measurement data from physical systems being modeled are always crisp.

Every fuzzy controller and model uses a defuzzifier, which is simply a mathematical formula to achieve defuzzification. For fuzzy controllers and models with a more than one output variable, defuzzification carried out for each of them separately but in a very similar fashion. In most cases, only one defuzzifier is employed for all output variables, although it is theoretically possible to use different defuzzifiers for different output variables.

The general defuzzifier represents many different defuzzifiers in one simple mathematical formula. Assume that output variable of fuzzy controller or model is z. suppose that evaluating N Mamdani fuzzy rules using some fuzzy inference method produces N membership values, $\mu_1,\ldots,\mu_N$, for N singleton output fuzzy sets in the rules (one value for each rule). Let us say that these fuzzy sets are nonzero only at $z=\beta_1,\ldots,\beta_N$. The generalized defuzzifier produces the following defuzzification result:

$$z = \frac{\sum_{k=1}^{N} \mu_k^{\alpha} \cdot \beta_k}{\sum_{k=1}^{N} \mu_k^{\alpha}}$$ where α is a design parameter.

- Mamdani fuzzy logic controller of steam engine

  [11]It was Mamdani who demonstrated the way to use fuzzy logic for control by constructing the first fuzzy controller. The controller was designed for a plant comprised of a steam engine and boiler combination. The model of the plant had two inputs: the heat input to the boiler and the throttle opening at the input of the engine cylinder, and two outputs: the steam pressure in the boiler and the speed of the engine. The problem in classical control found by Mamdani was that the plant model was highly nonlinear with both magnitude and polarity of the input variables.

  For the fuzzy processing, Mamdani proposed to control the plant by realizing some fuzzy rules or fuzzy conditional statements, for example:

  > *if* pressure error (PE) is negative big (NB)
  > *then* heat change (HC) is positive big (PB)

  So he can measure outputs of a plant and calculate a control action according to these rules. Mamdani has also proposed a modification to the controller. In order to improve the quality, he increased the number of control inputs and used the change in pressure error (CPE), defined as the difference between the present PE and the last one (corresponding to a last sampling instant) and the change in speed error (CSE) as well.

CHAPTER 3
METHODOLOGY

3.1 Procedure identification

There are several procedures need to be follow to accomplish this project. This project will be done in two semester period which is approximately one year. The author has chosen the empirical modeling as the modeling method. In this procedure identification, the overall project flow, the empirical modeling procedures and the procedures to design fuzzy logic controller will be discussed.

3.1.1 Project flow

Figure 7 shows the overall project flow of the project that will be accomplish in two semester period. The initial work is to obtain the identification of the system that is the transfer function of the plant so that the heat exchanger process can be modeled. In order to get the transfer function, the experiment is done at the heat exchanger pilot plant at the laboratory. Then, the block diagram of the system is developed using the MATLAB and Simulink. Next, the simulation of the model is done using MATLAB. After that the output obtained is compared with the real plant experiment to validate the results. Next, the PID controller is used to control the system and the tuning is done to the PID controller in order to optimize the heat exchanger performance. After finish tuning the PID controller, the fuzzy logic controller is design to further improve the process response. Then, both PID controller and fuzzy logic controller responses are compared to investigate which controller gives better control performance.

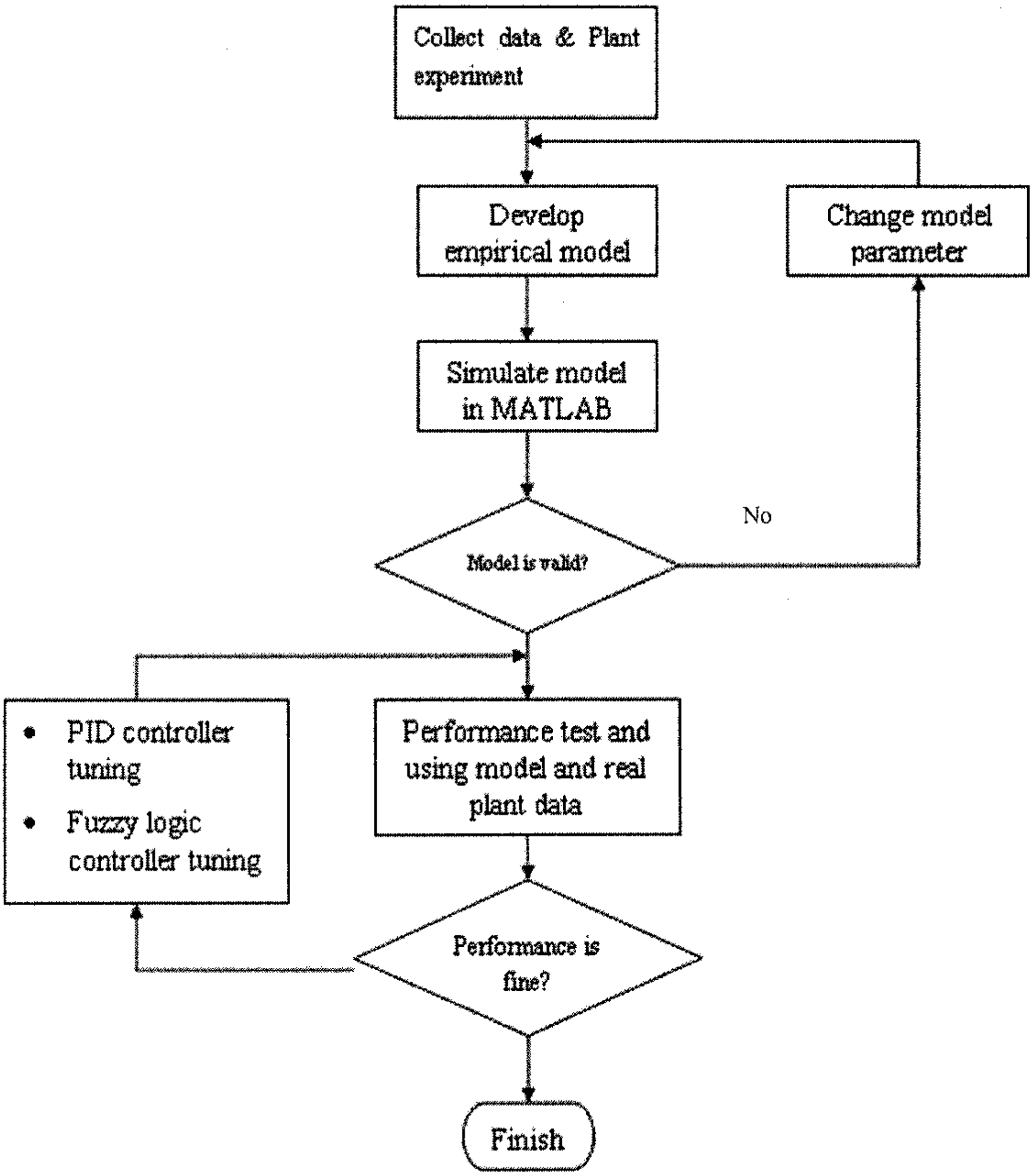


Figure 7 Project flow chart

3.1.2 Empirical modeling

The author will use the empirical modeling method [9] in order to develop the PID controlled and tuned process control. Empirical modeling is a modeling method specifically designed for process control and the models developed using this method provides the dynamic relationship between selected input and output variables. In empirical model building, models are determined by making small changes in the input variables about a nominal operating condition. The resulting dynamic response is used to determine the model. This general procedure is essentially an experimental linearization of the process that is valid for some region about the nominal conditions. The process reaction curve identification method will be used to determine the parameters. There are six-step procedure for the empirical model building as shown in *Figure 8* below, where this procedure endures that proper data is generated through careful experimental design and execution.

- Experimental design
    - In this step, the base operating condition, the perturbation and the variables to be measured are determined.

- Plant experiment
    - The experiment should be executed as close to the plan as possible. While variation in plant operation is inevitable, large disturbances during the experiment can invalidate the results; therefore plant operation should be monitored during the experiment.

- Determining model structure
    - Empirical methods typically use low-order models with dead time. Often (but not always), first order with dead time models are adequate for process control analysis and design.

- Parameter estimation
    - Estimates the parameters in transfer function models such as gain, time constant and dead time using graphical technique or statistical principles.

- Diagnostic evaluation
    - The evaluation is done to determine how well the model fits the data used for parameter estimation.

- Model verification
    - Verify it by comparison with additional data not used in the parameter estimation.

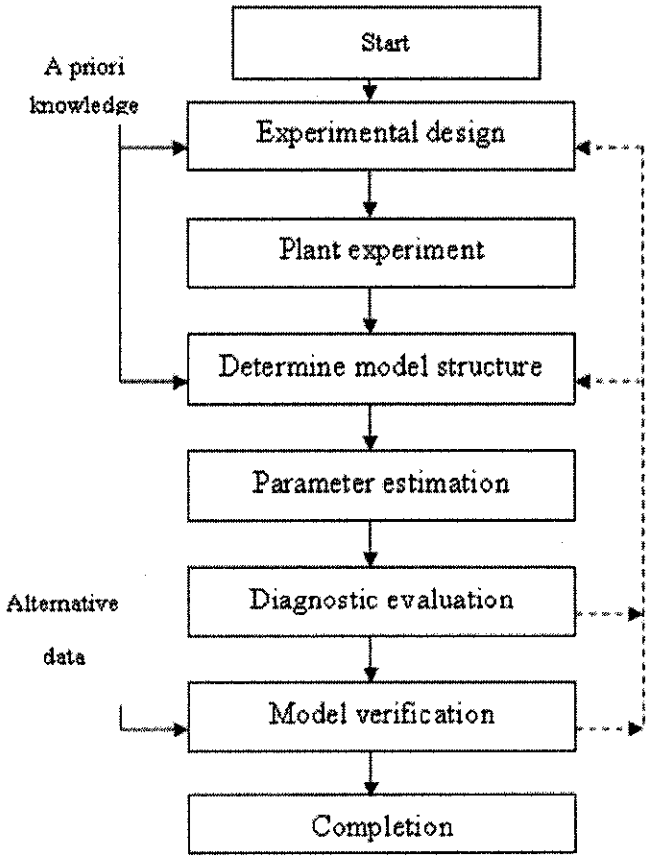


Figure 8 Procedure for Empirical Transfer Function Model Identification

3.1.3 Fuzzy logic controller

[11]Fuzzy logic controller is quite a complicated approach of control. However, it gives us a rather simple to use method for producing high quality controller with complicated input/output characteristics. To design fuzzy controllers, there is some design scheme need to be followed. The design scheme contains the following steps:

1. Define the input and control variables – to determine which states of the process shall be observed and which control actions are to be considered.
2. Define the condition interface – to fix the way in which observations of the process are expressed as fuzzy sets.
3. Design the rule base – to determine which rules are to be applied under which conditions.
4. Design the computational unit – to supply algorithms to perform fuzzy computations. Those will generally lead to fuzzy outputs. For this study, this part is mainly come from the internal Fuzzy Logic Toolbox function provided by MATLAB.
5. Determine rules according to which fuzzy control statements can be transformed into crisp control actions.

Figure 9 shows the procedure to design the fuzzy logic controller as define before. After pass through all the design steps, when the output response obtained is optimize and has achieve the goal set, the design is stop and that is the final fuzzy logic controller system to be used in this study.

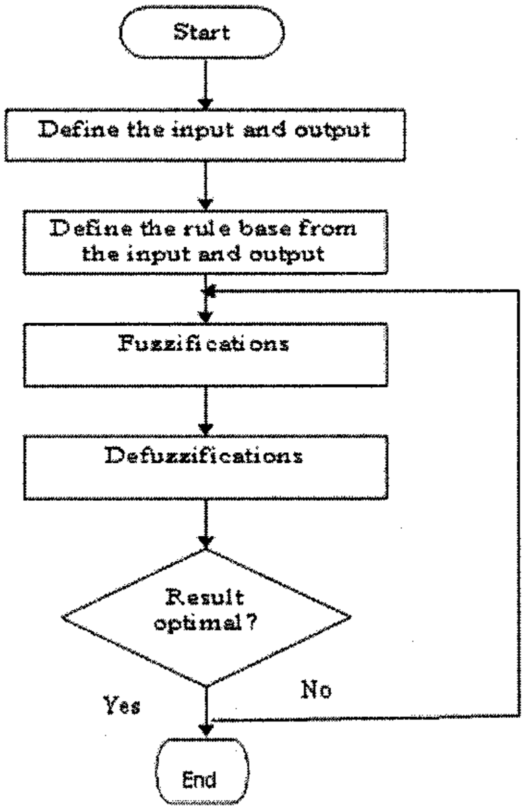


Figure 9 Procedure for fuzzy logic controller

3.2 Tools and software

3.2.1 Heat exchanger pilot plant with DCS

This is the main tool required in the project and as a conclusion; the project is totally dependent on the plant. If the plant has some problems, and it needs to be shutdown, no work can be done on the plant. The author has experience this problem before, however the author manage to finish the work. The pilot plant used is the Plant 6: Drum-Heat Exchanger Process Pilot Plant available in the UTP Instrumentation & Control System Lab. The important elements in this project are the temperature transmitter, flow transmitter, heat exchanger, control valve, server and the Distributed Control System (DCS) in the plant.

3.2.2 MATLAB and Simulink

MATLAB is a powerful, comprehensive and easy to use environment for performing technical computations. It is an interactive program that helps us with numeric computation and data visualization. It has features such as interactive mode of work, immediate graphing facilities, built in functions, the possibility of adding user written functions and simple programming. MATLAB offers array operations that allow one to quickly manipulate sets of data in a wide variety of ways. The Graphical User Interface (GUI) available in MATLAB allows one to use it as an application development tool [7].

Simulink is an extension to MATLAB that allows engineers to rapidly and accurately build computer models of dynamic systems, using block diagram notation. With Simulink, it is easy to model complex nonlinear systems. Additionally, a Simulink model can produce graphical animations that show the progress of a simulation visually, significantly enhancing understanding of system behavior [8].

3.2.3 Fuzzy logic toolbox

In order to design the fuzzy logic controller for the project, the author needs to use MATLAB Fuzzy Logic Toolbox as a tool in designing the controller. So, the author did some self study about the toolbox to get familiar with the interface and the working principle of the toolbox.

[14]What is Fuzzy Logic Toolbox? The Fuzzy Logic Toolbox is a collection of functions built on the MATLAB® numeric computing environment. It provides tools to create and edit fuzzy inference systems within the framework of MATLAB and also can integrate fuzzy systems into simulations with Simulink®. This toolbox relies heavily on graphical user

interface (GUI) tools to help user to accomplish their work. The Fuzzy Logic Toolbox allows user to do several things, but the most important thing is, it allow user to create and edit fuzzy inference systems. User can create these systems using graphical tools or command-line functions. There are five primary GUI tools (as shown in *Figure 10)* for building, editing, and observing fuzzy inference systems in the Fuzzy Logic Toolbox that are:

- **Fuzzy Inference System or FIS Editor**

  The FIS Editor displays general information about a fuzzy inference system. It shows the names of each input variables on the left, and those of each output variable on the right.

- **The membership function editor**

  The Membership Function Editor is a tool that lets user display and edits all of the membership functions associated with all of the input and output variables for the entire fuzzy inference system.

- **The Rule Editor**

  The Rule Editor is used to construct rule statements that define the behavior of the system.

- **The Rule Viewer**

  The Rule Viewer displays a roadmap of the whole fuzzy inference process and allows user to interpret the entire fuzzy inference process at once. It also shows how the shape of certain membership functions influences the overall result.

- **The Surface viewer.**

  The Surface Viewer is used to display the dependency of one of the outputs on any one or two of the inputs—that is, it generates and plots an output surface map for the system.

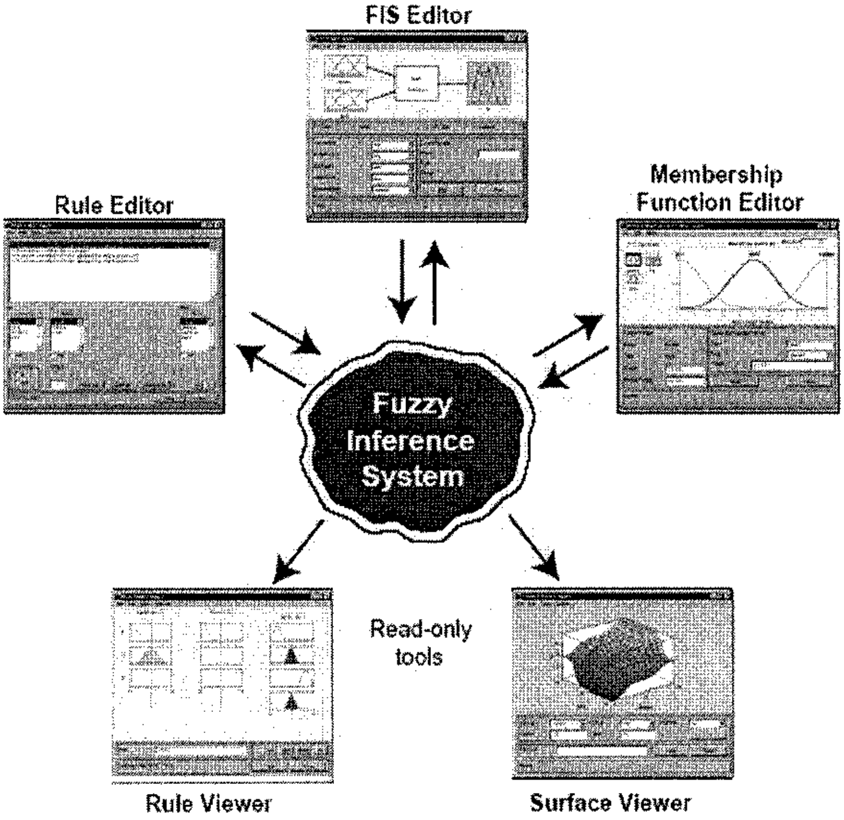


Figure 10 Fuzzy logic toolbox tools

3.2.4 Fuzzy inference system development

[14]Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. The mapping then provides a basis from which decisions can be made, or patterns discerned. The process of fuzzy inference involves membership functions, fuzzy logic operators, and if-then rules. There are two types of fuzzy inference systems that can be implemented in the Fuzzy Logic Toolbox: Mamdani-type and Sugeno-type. In MATLAB, there are 5 parts of the fuzzy inference process. All figures in this section are referred from the MATLAB Fuzzy logic toolbox help files. The example used is the fuzzy tipping problem explained in the help file.

- Fuzzification of the input variables

The first step is to take the inputs and determine the degree to which they belong to each of the appropriate fuzzy sets via membership functions. In the Fuzzy Logic Toolbox, the input is always a crisp numerical value limited to the universe of discourse of the input variable. *Figure 11* shows how the input of fuzzy tipping problem is fuzzified.

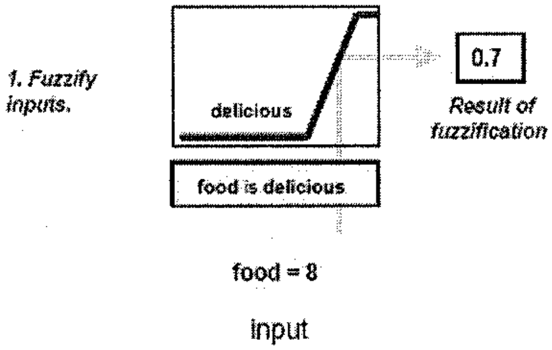


Figure 11 Fuzzification of inputs

- Application of the fuzzy operator (AND or OR) in the antecedent

Once the inputs have been fuzzified, we know the degree to which each part of the antecedent has been satisfied for each rule. If the antecedent of a given rule has more than one part, the fuzzy operator is applied to obtain one number that represents the result of the antecedent for that rule. This number will then be applied to the output function. The input to the fuzzy operator is two or more membership values from fuzzified input variables. The output is a single truth value. In the Fuzzy Logic Toolbox, two built-in AND methods are supported: *min* (minimum) and *prod* (product). Two built-in OR methods are also supported: *max* (maximum), and the probabilistic OR method *probor*. *Figure 12* shows how the fuzzy operator operation is applied to the antecedent of fuzzy tipping problem

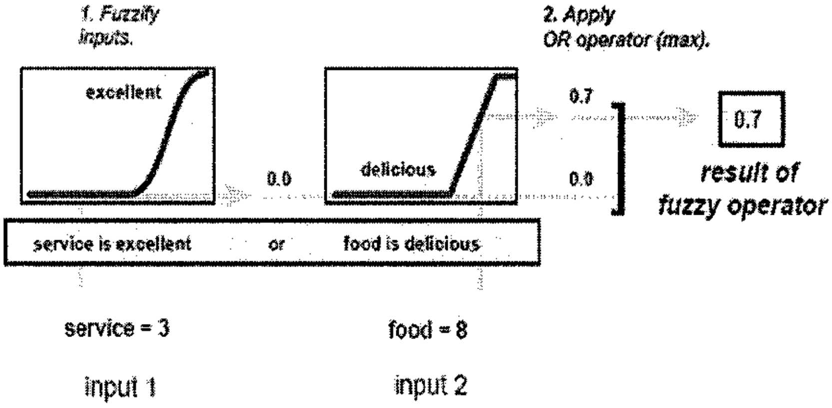


Figure 12 Application of fuzzy operator

- Implication from the antecedent to the consequent

Before applying the implication method, we must take care of the rule's weight. Every rule has a weight (a number between 0 and 1), which is applied to the number given by the antecedent.

Once proper weighting has been assigned to each rule, the implication method is implemented. A consequent is a fuzzy set represented by a membership function, which weights appropriately the linguistic characteristics that are attributed to it. The consequent is reshaped using a function associated with the antecedent (a single number). The input for the implication process is a single number given by the antecedent, and the output is a fuzzy set.

Figure 13 shows how the implication is done from the antecedent to the consequent for the fuzzy tipping problem. The consequent is the output fuzzy set.

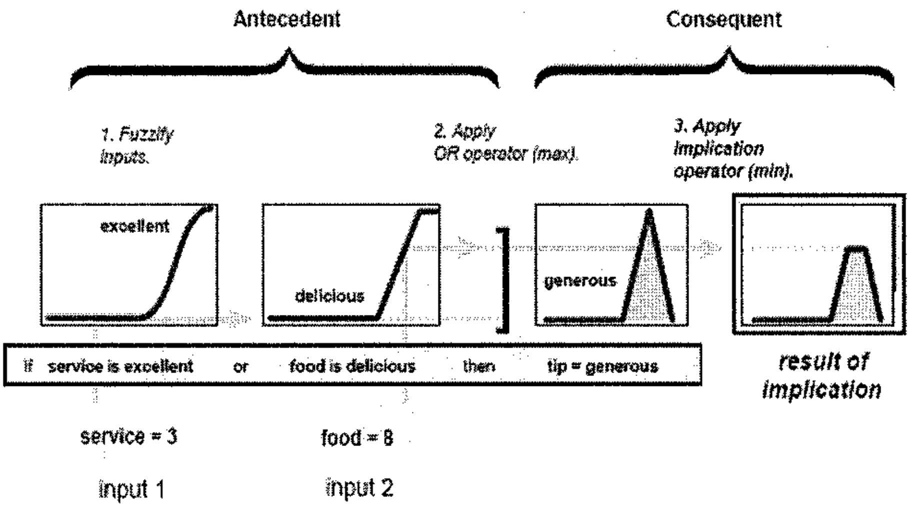


Figure 13 Implication from the antecedent to the consequent

- Aggregation of the consequents across the rules

Since decisions are based on the testing of all of the rules in an FIS, the rules must be combined in some manner in order to make a decision. Aggregation is the process by which the fuzzy sets that represent the outputs of each rule are combined into a single fuzzy set. Aggregation only occurs once for each output variable. The input of the aggregation process is the list of truncated output functions returned by the implication process for each rule. The output of the aggregation process is one fuzzy set for each output variable.

Three built-in methods are supported in the Fuzzy logic toolbox: *max* (maximum), *probor* (probabilistic OR), and *sum* (simply the sum of each rule's output set). In the *Figure 14*, all three rules have been placed together to show how the output of each rule is combined, or aggregated, into a single fuzzy set whose membership function assigns a weighting for every output (tip) value

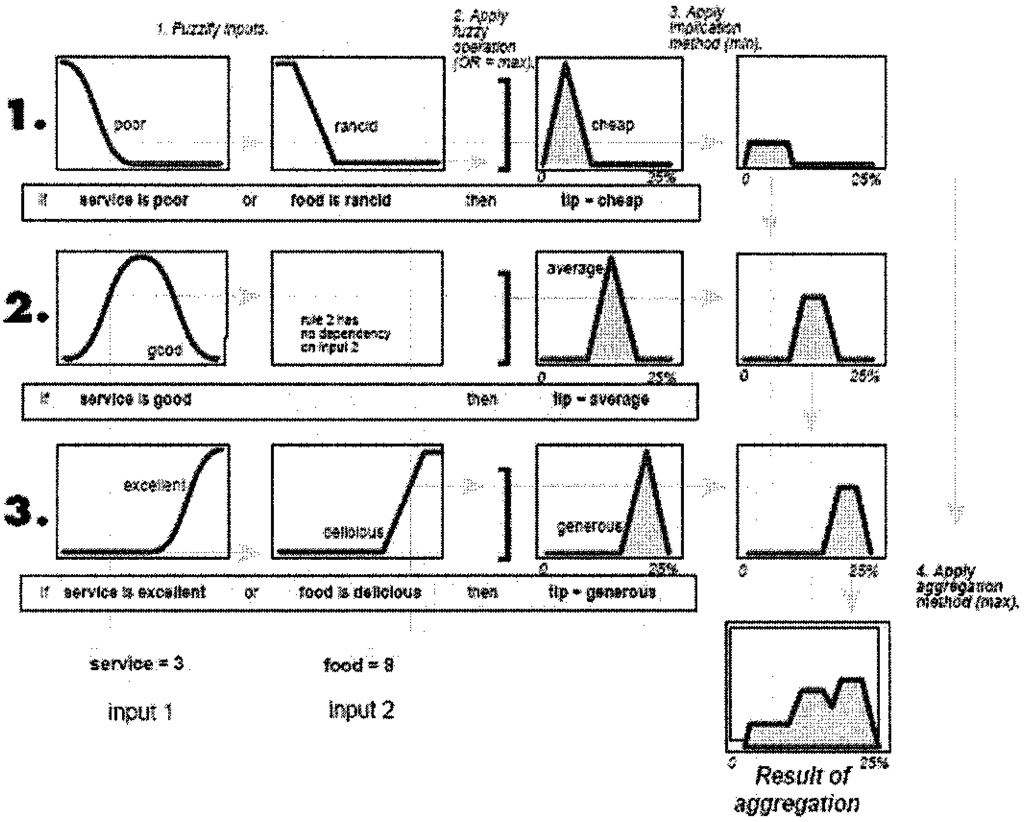


Figure 14 Application of aggregation method

- Defuzzification

The input for the defuzzification process is a fuzzy set (the aggregate output fuzzy set) and the output is a single number. However, the aggregate of a fuzzy set encompasses a range of output values, and so must be defuzzified in order to resolve a single output value from the set.

Perhaps the most popular defuzzification method is the centroid calculation, which returns the center of area under the curve. There are five built-in methods supported: centroid, bisector, middle of maximum (the average of the maximum value of the output set), largest of maximum, and smallest of maximum. *Figure 15 shows* the defuzzification of the aggregate output using centroid method.

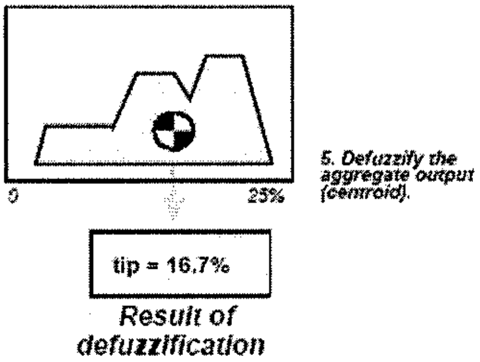


Figure 15 Defuzzification of aggregate output

CHAPTER 4
RESULTS & DISCUSSION

4.1 Empirical model

The heat exchanger empirical model is developed based on process reaction curve where the heat exchanger transfer function is estimated to be a first order with dead time model. *Figure 16* shows the process reaction curve obtained from the plant experiment. [9]There are two slightly different methods of graphical techniques in common use for process reaction curve that are method I and method II. Method I adapted from Ziegler and Nichols (1942) needed the author to find the slope of the measured signal. Because of the difficulty in evaluating the slope especially when the signal has high frequency noise, Method I typically has larger errors in the parameter estimates. On the other hand, Method II uses times at which the output reaches 28 and 63 percent of its final value. The typical times are selected where the transient response is changing rapidly so that the model parameters can be accurately determined despite the presence of measurement noise. Thus, Method II is preferred because it produces less error. Thus it is used in this project to obtain transfer function parameters. The summary of model parameters calculated for the heat exchanger temperature loop is:

Temperature loop

Process Gain, K_P = 0.17 °C/ % opening

Time Constant, τ = 120 seconds

Time Delay, θ = 40 seconds

[9]The general formula for the first order with dead time model transfer function is:

$$\frac{Y(s)}{X(s)} = \frac{K_p e^{-\theta s}}{\tau s + 1}$$

Hence, the first order with dead time model transfer function for the heat exchanger temperature loop is

$$\frac{Y(s)}{X(s)} = \frac{0.17e^{-40s}}{120\ s + 1}$$

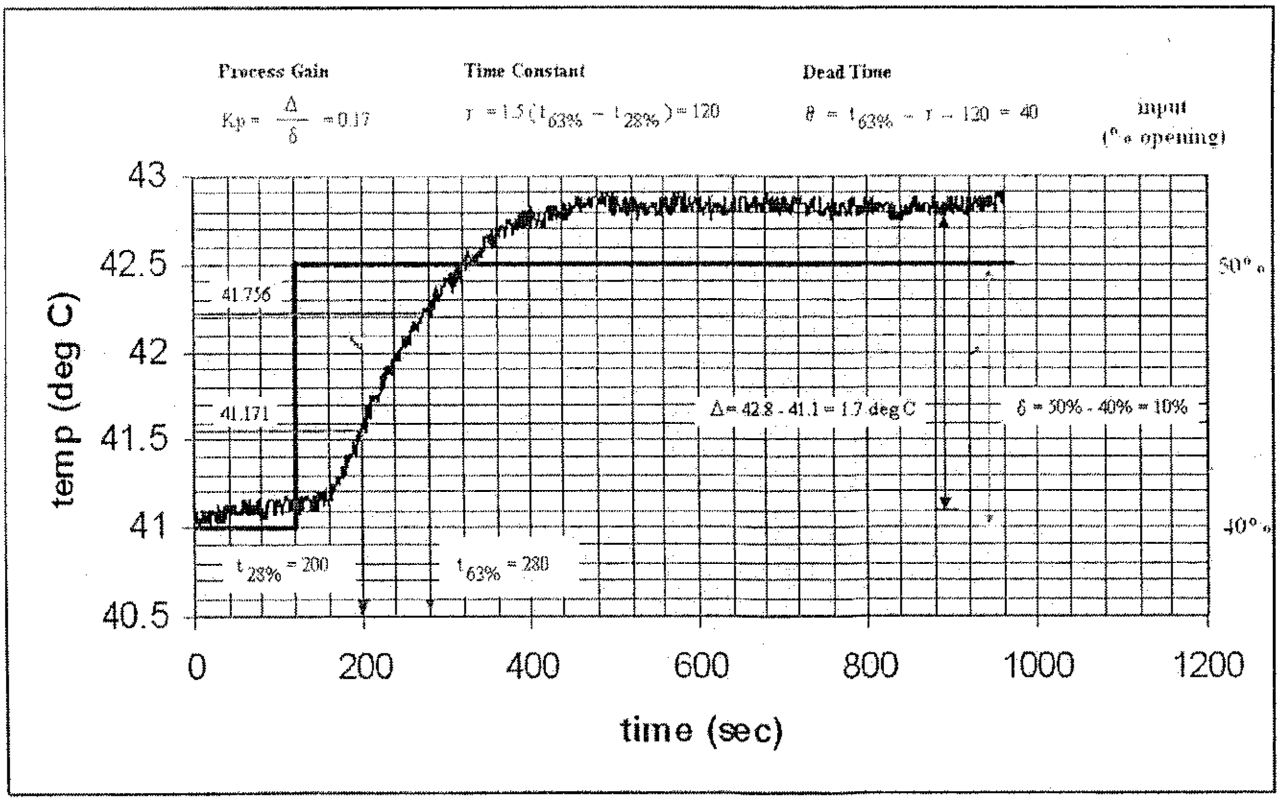


Figure 16 Process reaction curve

4.1.1 Open loop test

Figure 17 shows the Simulink block diagram an open loop test of the temperature loop empirical model. The simulation input is a step input that resembles valve opening from 0% to 10%. Output from the transfer function which is the process variable in this temperature loop is displayed by a scope.

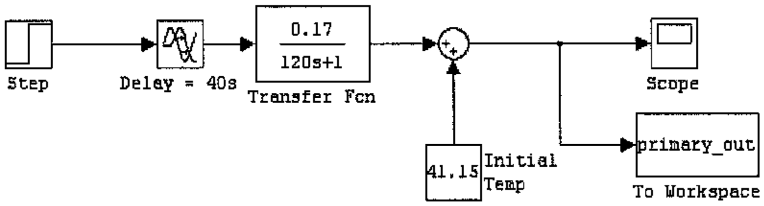


Figure 17 Simulink block diagram for open loop test of temperature

Figure 18 shows the open loop response of temperature from this empirical model open loop test. In this open loop response, the process variable (temperature) is not following the set point but instead is reacting to the percentage opening of the valve. The opening of the valve is increased by an additional 10% from the previous opening as an increased step change in this open loop experiment.

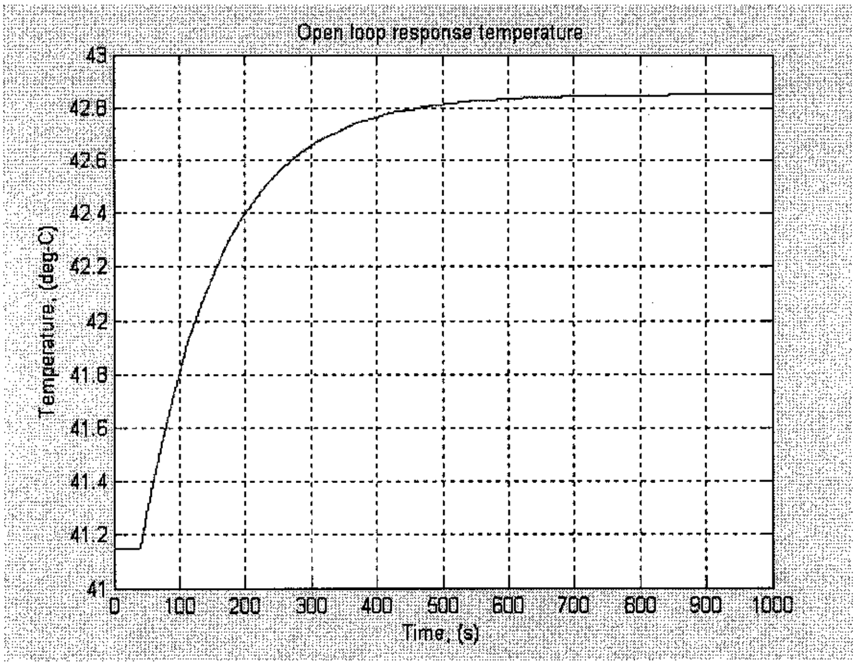


Figure 18 Open loop response of temperature

4.2 PI controller development

4.2.1 Closed loop test

A closed loop test was conducted to develop the PI controller parameters. In this test, only PI parameters are chosen because as experienced from the plant experiment, the D parameter is shown not really needed in this chosen system. Besides, from the manual data of the plant, the experiment result shows that it is acceptable enough to use PI only parameters. The D mode can amplifies sudden changes in the controller input signal and can cause a potentially large variation in the controller output that can lead to unwanted situation. Besides, high frequency noise on the CV measurement can cause excessive variation in the MV. An obvious step to reduce the effects of noise is to reduce the derivative, D time perhaps to zero as done in this closed loop test. With these controller parameters, the system will have an effect of controller, where the process variable will be controlled and should follow the set point specified. The manipulated variable will react to the controller parameters so that the process variable will be maintained at the set point. For this project, the set point for the temperature response is specified at 50 °C. The PI tuning parameters are calculated by using Cohen & Coon open loop tuning method, Ziegler-Nichols open loop tuning method and also the Ciancone correlation. . Please refer to (*Appendix A)* for the calculation of the tuning parameters. The result will be discussed in this section. The Simulink block diagram for the closed loop system is shown in *Figure 19*

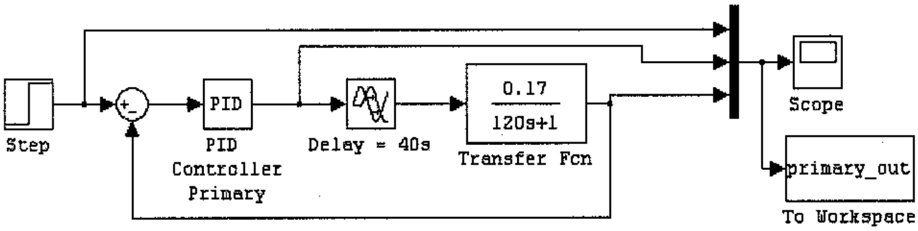


Figure 19 Simulink block diagram for closed loop temperature

- Cohen & Coon tuning method

The Cohen & Coon open loop tuning parameter formula (for PI)

$$K_C = \left[\frac{1}{R \bullet K_P}\right]\left[\frac{9}{10} + \frac{R}{12}\right]$$

$$T_I = T_d \bullet \left[\frac{30 + 3R}{9 + 20R}\right]$$

Figure 20 shows the closed loop response of temperature using the Cohen & Coon tuning method.

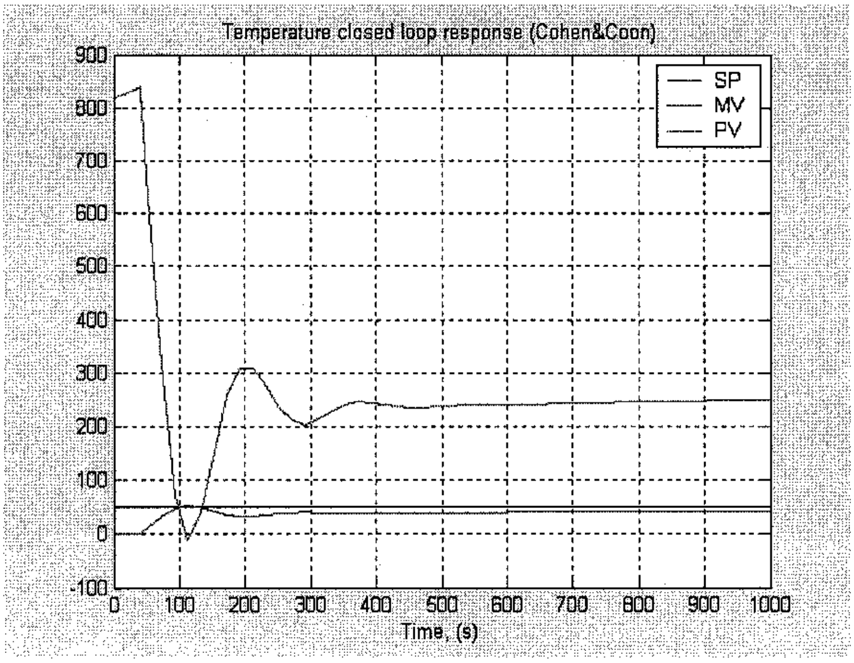


Figure 20 Cohen & Coon temperature response [$K_C = 16.3741$ $T_I = 0.01$]

- Ziegler-Nichols open loop tuning method

Ziegler – Nichols open loop tuning method (PI only)

$$K c = \left(\frac{0.9}{Kp}\right)\left(\frac{\tau}{\theta}\right)$$

$$T i = 3.3\,\theta$$

Figure 21 shows the closed loop response of temperature using Ziegler-Nichols tuning method.

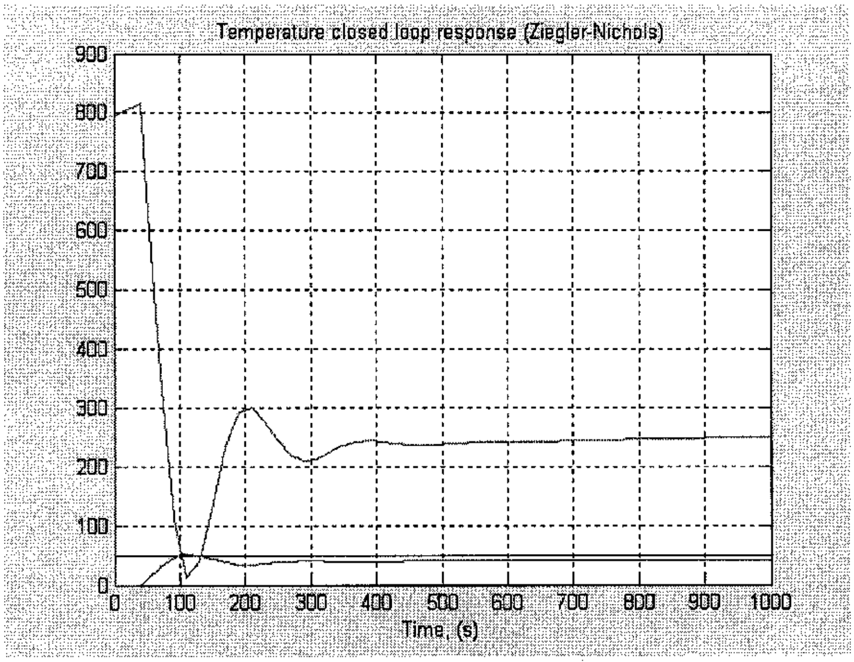


Figure 21 Ziegler-Nichols temperature response [$K_C = 15.8823$ $T_I = 0.01$]

- Ciancone correlations

The purpose of tuning correlations is to calculate tuning constants that achieve the goals targeted. The goals are to minimize the Integral Absolute Error (IAE), considering the error in the process model parameters and also to limit the variation of the MV. This correlation is done using the Ciancone correlation developed by Ciancone and Marlin (1992). [10]

The Ciancone correlations consist of the following steps:

- Ensure that the performance goals and assumptions are appropriate
- Determine the dynamic model using an empirical method (process reaction curve) giving K_p, θ, and τ
- Calculate the fraction dead time $\frac{\theta}{\theta + \tau}$
- Select the appropriate correlation, disturbance or set point

- Determine the dimensionless tuning values from the graphs for $K_c\ K_p$, $\frac{Ti}{\theta+\tau}$, and $\frac{Td}{\theta+\tau}$
- Calculate the dimensional controller tuning
- Implement and fine tune as required.

Figure 22 shows the Ciancone correlation closed loop response for temperature loop.

$\frac{\theta}{\theta+\tau} = 0.25 \qquad Kc = 7.3529 \qquad Ti = 147.2 \quad \frac{1}{Ti} = 0.00679$

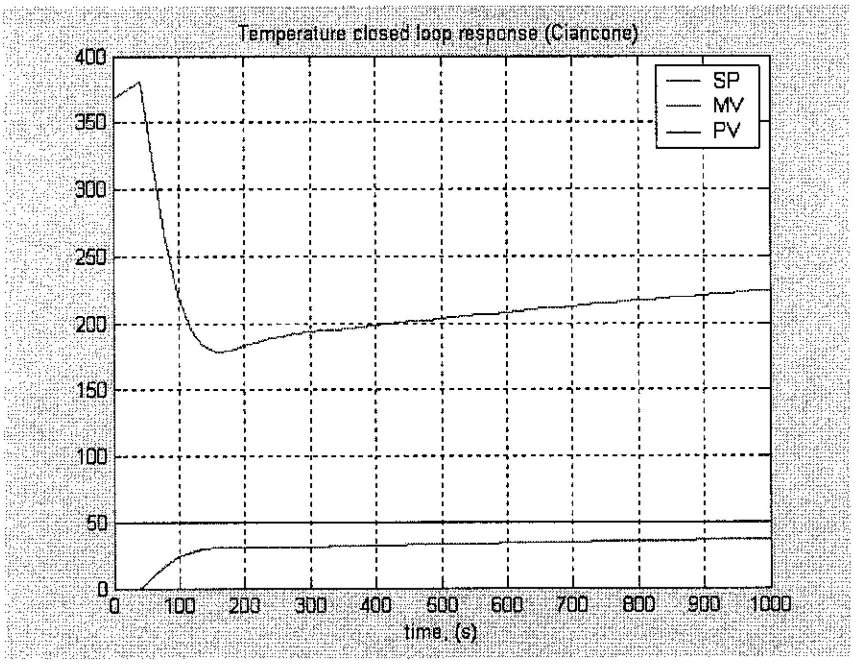


Figure 22 Ciancone temperature response [$K_C = 7.3529$ $T_I = 0.01$]

4.2.2 Control performance analysis of tuning parameters

The definition of the control performance parameters are discussed below:

- Rise time, T_r [*refer to Figure 23*] is the time from the step change in set point until the CV first reaches the new set point. Usually short rise time is desired.
- The settling time is time the system takes to attain a nearly constant value usually ±5% of its final value. A short settling time is usually favored.
- Offset is a difference between final, steady-state values of the set point and of the controlled variable.
- Decay ratio, (B/A) [*refer to Figure 23*] is the ratio of neighboring peaks in an underdamped controlled-variable response. Usually, periodic behavior with large amplitudes is avoided in process variables; therefore, a small decay ratio is usually desired and an overdamped response is sometimes desired.
- MV overshoot, (C/D) [*refer to Figure 23*] is use as an indication of how aggressively the MV has been adjusted.
- CV overshoot is an important measure of the process degradation experienced due to disturbance.

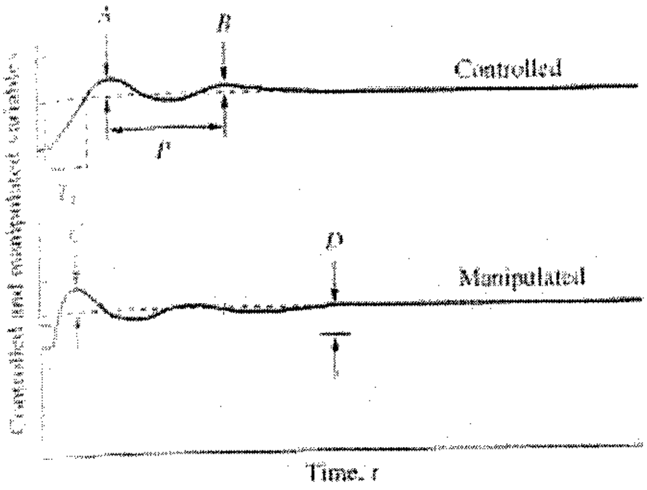


Figure 23 Typical transient response of a feedback control system to a step change

The calculated values of PI parameter obtained from the plant experiment using different open loop tuning methods are shown in *Table 1*.

Parameters	**Cohen Coon**	**Ziegler Nichols**	**Ciancone correlations**
K_C	16.3741	15.8823	7.3529
T_I	100	100	100

Table 1 PI parameter values

Control performance analysis was conducted on the model of the plant in a closed loop system with step input from 0 to 50 and set point of 50 °C. The control performance analysis for each open loop tuning methods is given in *Table 2*. Please refer to (*Appendix C)* for the detail calculations.

Control performances	**Cohen Coon**	**Ziegler Nichols**	**Ciancone correlations**
Rise time	108.85 s	125.65 s	> 80 min
Settling time	> 50 min	> 50 min	> 80 min
Offset	4 °C	3.3 °C	0.5 °C
CV overshoot	0.56%	2.5%	n/a
MV overshoot	203.21%	227%	25.89%
Decay ratio	n/a	n/a	n/a

Table 2 Control performance for each open loop tuning methods

4.2.3 Fine tuning

The closed loop tuning method provides the author with a basic calculation for the PI controller parameters. From the control performance analysis, it is shown that the responses were not satisfactory because of higher settling and rise time and also with offset. Thus, there is a need to fine tune the controllers to get an acceptable response for the system. The goal set for the PI controller is to achieve 25% damping ratio or quarter decay ratio. This is the standard criteria set in the industry as guideline especially for the process control. With the quarter decay ratio response, the others control performance parameters such as settling time and rise time can be accepted.

For the fine tuning, the basic calculation used is the Ziegler Nichols open loop tuning value. The Ziegler Nichols open loop tuning value is chosen because it has the smallest offset with acceptable rise time compared to the another two methods as shown in *Table 2*. From the basic value of P = 15.8823 and I = 100, the system response is fine tuned to achieve quarter decay ratio. After a few trials of fine tuning, this is the value of fine tune P and I that bring the system response to achieve quarter decay ratio.

→ P = 5.5

→ I = 6.25

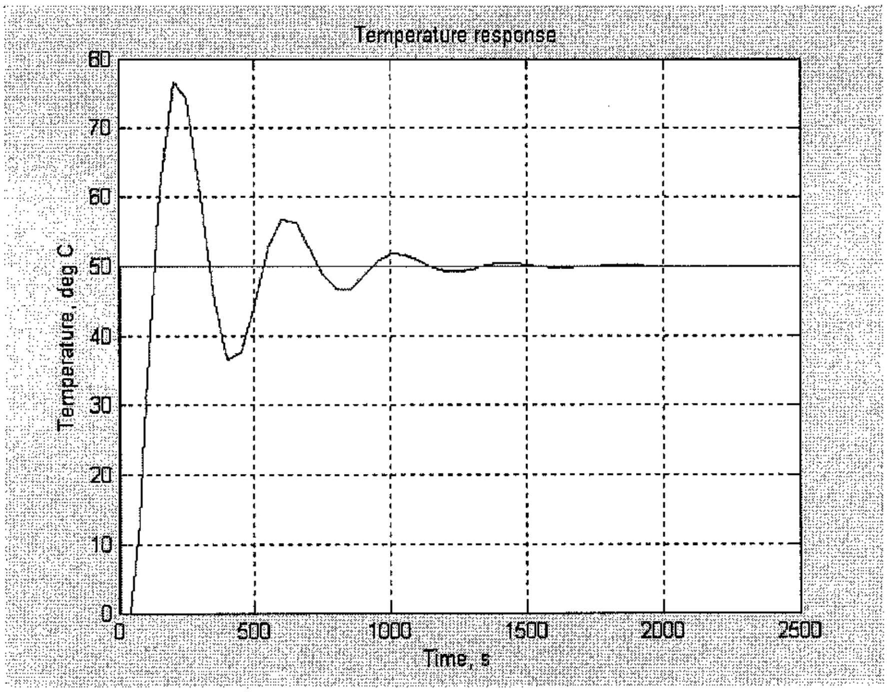


Figure 24 Temperature response with quarter decay ratio

Figure 24 and 25 shows the temperature response and MV response of temperature loop with quarter decay ratio respectively.

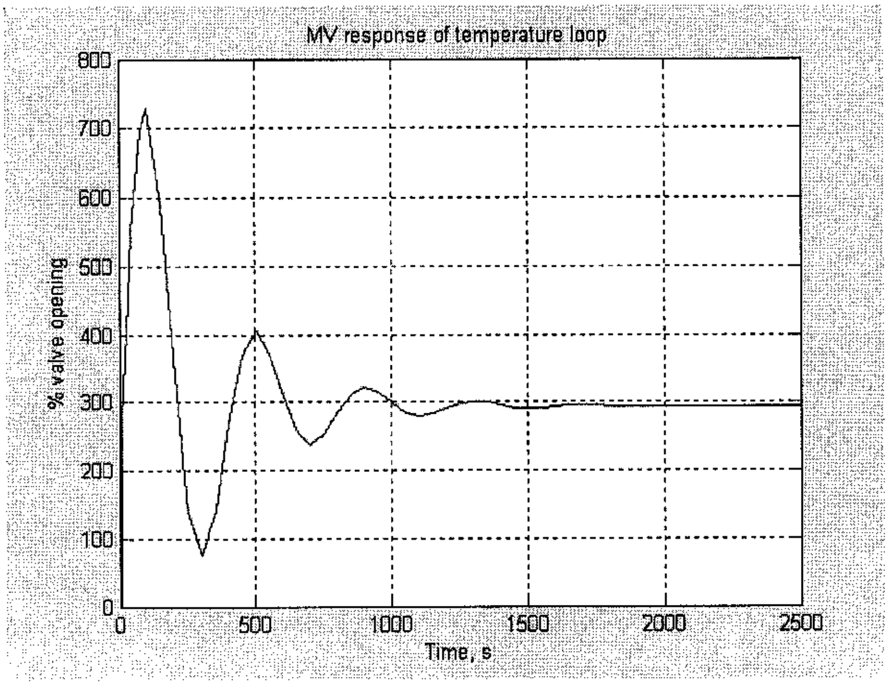


Figure 25 MV response of temperature loop with quarter decay ratio

4.2.4 Control performance analysis of fine tuning parameters

Table 3 shows the control performance analysis for the fine tuned PI controller. Please refer to (*Appendix C)* for the detail calculations.

Control performance	**Fine tuning**
Rise time	134.7 s
Settling time	30 min < x < 35 min
Offset	0.01 °C
CV overshoot	53.36%
MV overshoot	147.62%
Decay ratio	25.30%

Table 3 Control performance for fine tuning

After the fine tune, the responses achieved quarter decay ratio and acceptable rise time and settling time. The settling time for the fine tuned system is lower compared to the basic Ziegler Nichols settling time. Besides, there is no offset as can be seen from the graph as the integral parameter has bring the system to zero offset.

4.2.5 Issues

Some issues rise here such as the value of valve opening is more than 100% and the set point is specified at 50°C. The issues are discussed here.

- Valve opening

  For the valve opening issue, the author made an assumption that for an instance of 300% of valve opening, it correlates to 30% of real plant valve opening. The real percentage of valve opening at the heat exchanger pilot plant is from 0 – 100%. With that assumptions, it means that the simulation value of valve opening is a multiply of constant 10 and the simulation range is 0 – 1000%.

- Set point

  While for the set point, it is specified at 50°C. This means that, the final temperature value that the response should have is at 50°C. This 50°C value is chosen because from the experiment done in the laboratory, it is shown that this temperature value is still in the temperature range of the heat exchanger output. Besides, the author made an assumption that the product from this plant should be heated at 50°C by the heat exchanger process. The maximum temperature value of the heat exchanger is known at 70°C and this value should not be achieved because it will trigger the high alarm of the system. The rated value for the heat exchanger at the pilot plant is at 90°C and if this value is achieved, the system will trigger the high-high alarm to shutdown the process.

4.3 Fuzzy logic controller

4.3.1 Introduction

Fuzzy logic controller is another approach for process control and it differs significantly from the conventional control. There are two approaches to a fuzzy controller design: an expert approach and a control engineering approach. In this project, the author will look into the first approach where the fuzzy controller structure and parameters choice are assumed to be the responsibility of the experts. This approach is called Mamdani fuzzy inference method.

4.3.2 Fuzzy inputs and outputs definition

This fuzzy inference system is developed based on the heat exchanger process application done in previous PI controller tuning. In this project, the author will look into the Mamdani fuzzy inference method and developed the system as the PI-like fuzzy controller. Thus, the integral will be one of the inputs for this system.

[11]Fuzzy input is the input variables where it is the states of the process shall be observed and measured. Fuzzy output is the output variable which determines the control actions to be considered. For this design, the definition of input and output is defined in *Table 4* according to the heat exchanger process.

Fuzzy input	Fuzzy output
Temperature error	% of valve opening
Integral temperature error	

Table 4 Fuzzy input and output definition

4.3.3 PI controller simulation data

Mamdani fuzzy inference system is developed using the experience from the human expertise or from the experience obtained through the experiment. The experience of the system can be obtained from the relationship of the PI controller simulation data. Thorough analysis of the PI controller data will provide the relationship between the inputs and output for the fuzzy logic controller. This relationship then will be used to develop the membership functions and fuzzy rules of the system. *Figure 26* shows the Simulink block diagram of the PI controller to obtain the simulation data. There are inputs of temperature error, integral of temperature error and the MV of the PI controller for the data. Please refer to *(Appendix B)* for the data simulation.

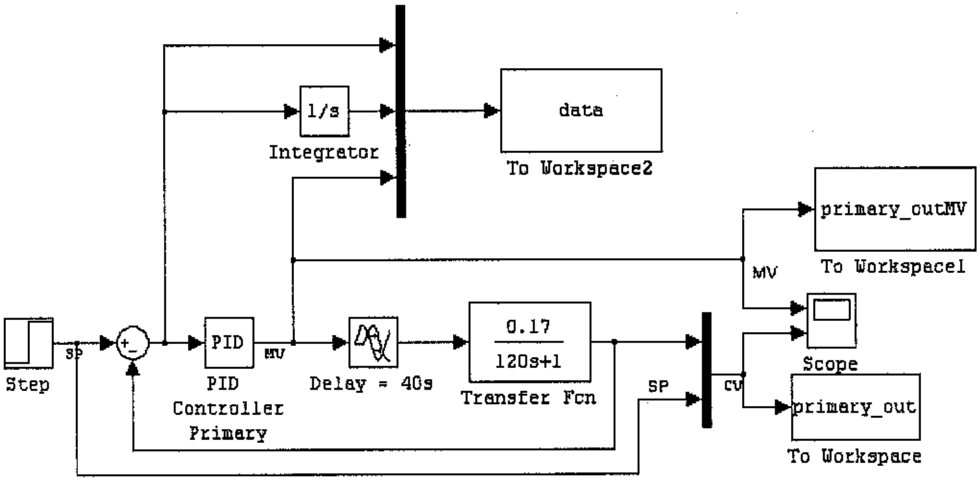


Figure 26 PI controller of temperature loop

4.3.4 Simulink design of fuzzy logic controller

The fuzzy controller designed can be integrated into simulations with Simulink. The design of the Simulink block diagram is same with the PI controller arrangement but the controller is replaced with fuzzy logic controller (FLC) block. *Figure 27* shows the Simulink block diagram of fuzzy logic controller.

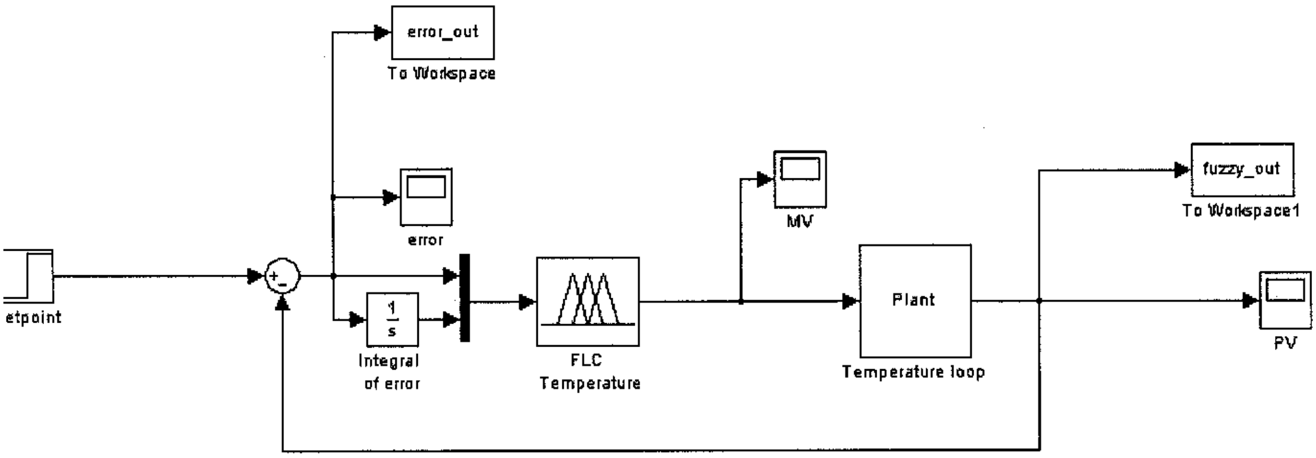


Figure 27 Simulink block diagram of fuzzy logic controller

4.3.5 Membership functions design

[12]A fuzzy set is given by its membership function. The value of this function determines if the element belongs to the fuzzy set and in what degree. There are different shapes of membership functions such as triangular, trapezoidal, quadratic and Gaussian. For this project, the author chose Gaussian and trapezoidal shape for the inputs membership function, while triangular and trapezoidal shape for the output membership function. The author has chosen Gaussian shape over other shapes because it gives a smoother transition between fuzzy sets.

Table 5 shows the definition of membership function input and outputs for the FLC. The linguistic variables chosen are determined from the simulation data of PI controller and also using some rational about the process studied. The linguistic variable is a value using natural language expression referring to some quantity of interest. These natural language expressions are also the names for fuzzy sets composed of the possible numerical values that the variable of interest can assume.

Temperature error [Input]	Integral of error [Input]	% valve opening [Output]
very-neg	very-low	Close big
neg	low	Close small
zero	average	Average
pos	high	Open small
very-pos	very-high	Open big

Table 5 Membership function input and outputs

*neg = negative

*pos = positive

Figure 28 to 30 show the design of the input and output membership functions for this project obtained from the MF editor tools in MATLAB Fuzzy Logic Toolbox. The range of the inputs and outputs are determined from the PI controller simulation data and also from the trial and error done along the membership functions development. The overlapping of the membership functions as can be seen in the figures are obtained from some rational definition about the process studied. Most of the method involve in this membership functions development is involving trial and error and it is guided by the author own experience of the plant.

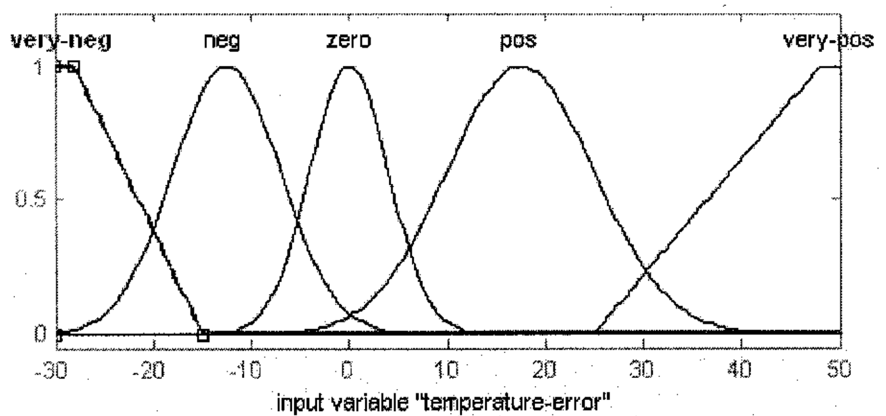


Figure 28 Temperature error membership functions

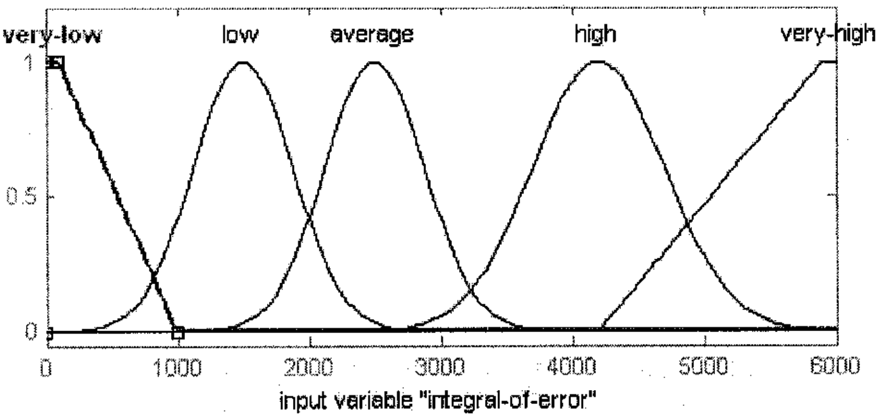


Figure 29 Integral of error membership functions

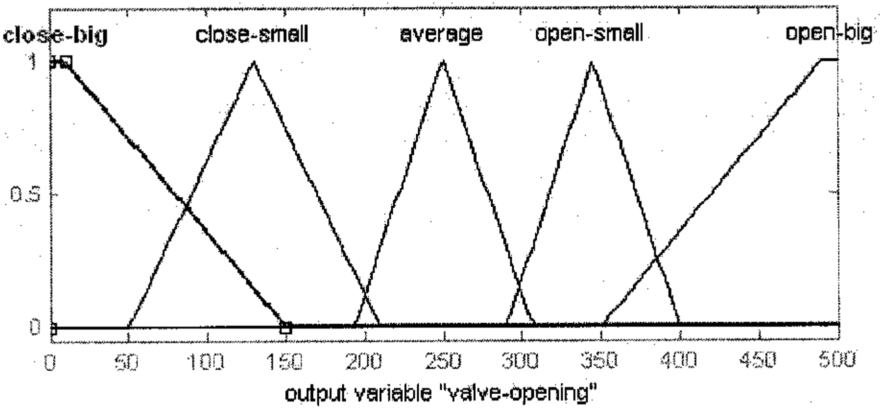


Figure 30 % of valve opening (output) membership functions

4.3.6 Fuzzy rules design

If – then rules or simply fuzzy rules is constructed to control the behavior of the system by following the fuzzy relations between the attributes involved in the process. This rule gives the dependence of output on input and establishes a relation between output and input. These rules are developed using the Rule Editor in the fuzzy logic toolbox. The relationships between the input and output of the process control are obtained from the PI controller simulation data. In this section, the author lists down the fuzzy rules constructed for the FLC of heat exchanger. *Table 6* shows the rules designed for the FLC system. These fuzzy rules will be fired accordingly during the fuzzy processing of the FLC.

*neg = negative

*pos = positive

Rules
1. If temperature error is zero and integral of error is very low then valve opening is close big
2. If temperature error is pos and integral of error is low then valve opening is open big
3. If temperature error is pos zero and integral of error is high then valve opening is open big
4. If temperature error is neg zero and integral of error is high then valve opening is open big
5. If temperature error is neg zero and integral of error is average then valve opening is open small
6. If temperature error is neg and integral of error is average then valve opening is average
7. If temperature error is zero and integral of error is average then valve opening is open small
8. If temperature error is zero and integral of error is very high then valve opening is open big **9. If temperature error is neg zero and integral of error is very high then valve opening is open big**
10. If temperature error is neg zero and integral of error is low then valve opening is close small
11. If temperature error is pos and integral of error is average then valve opening is open big
12. If temperature error is pos zero and integral of error is average then valve opening is average
13. If temperature error is pos zero and integral of error is low then valve opening is average

Table 6 If – then rules

4.3.7 Fuzzy logic controller response

After all the design process, the FLC is implemented with the Simulink block diagram as in *Figure 27.* In this study, the author will look into the step disturbance of the process. The FLC system is given a step response from 0°C to 50°C where, the 50°C is the set point for the heat exchanger process. The feedback system of this FLC will compensate the error from this step disturbance. The author will observe the temperature response and also the MV response of the FLC. The goal set for this response is quarter decay ratio. *Figure 31* and *Figure 32* show the temperature response and MV response of FLC respectively.

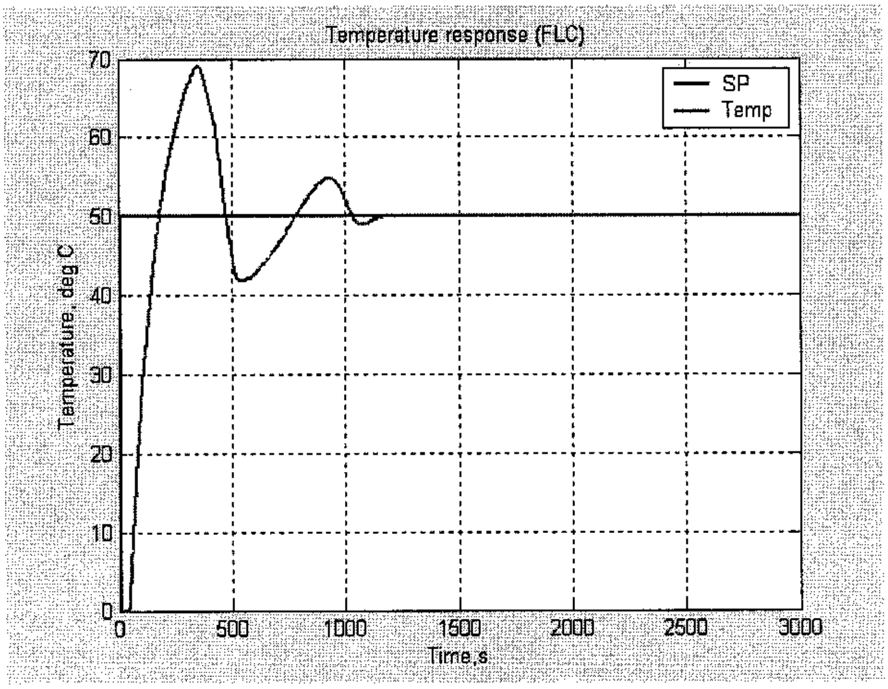


Figure 31 Temperature response for fuzzy logic controller

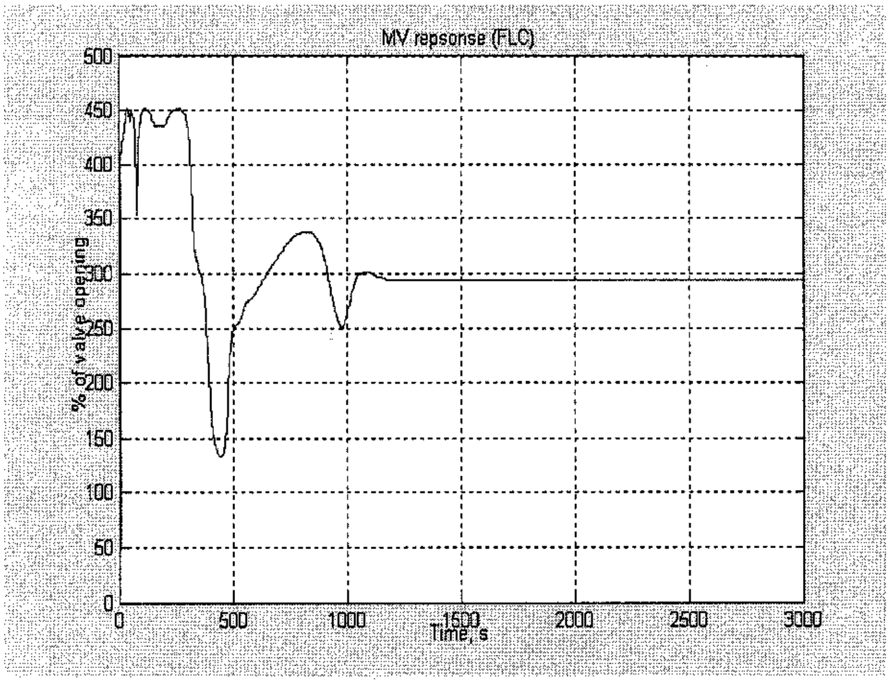


Figure 32 MV response (valve opening) for fuzzy logic controller

4.3.8 Control performance analysis of fuzzy logic controller

Table 7 shows the control performances analysis for the fine tuned FLC. Please refer to (*Appendix C)* for the detail calculations.

Control performance	**Fuzzy logic controller**
Rise time	182.17 s
Settling time	21.67 min
Offset	0 °C
CV overshoot	38.3%
MV overshoot	53.34%
Decay ratio	25.17%.

Table 7 Control performance for fuzzy logic controller

From the control performances analysis, it can be seen that the FLC response has achieved zero offset and also have quarter decay ratio response. The overshoot of both MV and CV is also in acceptable range for the process.

4.4 Comparison between PI controller and Fuzzy Logic controller

One of the objectives of this study is to make an investigative and comparative analysis between the PI controllers; a conventional method with the fuzzy logic controller; an intelligent method. In this comparative study, the author looked into the control performance analysis for each controller such as rise time, settling time and overshoot. *Table 8* shows the control performance comparison between PI controller and fuzzy logic controller. *Figure 33* and *Figure 34* show the Simulink block diagram use for the comparison analysis and the comparison temperature response between the two controllers respectively.

Control performances	Fuzzy logic controller	PI controller
Rise time	182.17 s	134.7 s
Settling time	21.67 min	> 30 min
Offset	0°C	0.01°C
CV overshoot	38.3%	53.36%
MV overshoot	53.34%	147.62%
Decay ratio	25.17%.	25.30%

Table 8 Control performance comparison between PI and fuzzy logic controller

Table 8 shows that:

- FLC has reduced the settling time for the process to achieve zero offset and settling at the set point of 50°C. The settling time has been reduced more than 10 minutes compared to the PI controller value. This means that, the FLC has faster error compensation compared to PI controller.
- FLC also has reduced the overshoot for both CV and MV if compared to the PI controller overshoot.

- The decay ratio for this comparison is at quarter decay ratio which is the guideline used by the process industry for the process response at the plant.
- Both PI and fuzzy logic controller has achieved zero offset, which is known that PID controller has that characteristic of zero offset with the usage of integral and fuzzy logic controller is designed to achieve that zero offset.
- However, for the settling time, PI controller is faster by 50 seconds compared to the FLC value. Since this study is on temperature with slow dynamic response, the value is acceptable.
- The FLC has also reduced the oscillation of the temperature response compared to the PI controller. This makes the FLC response more stable.
- The FLC has also reduced the overshoot of the system compared to PI controller.

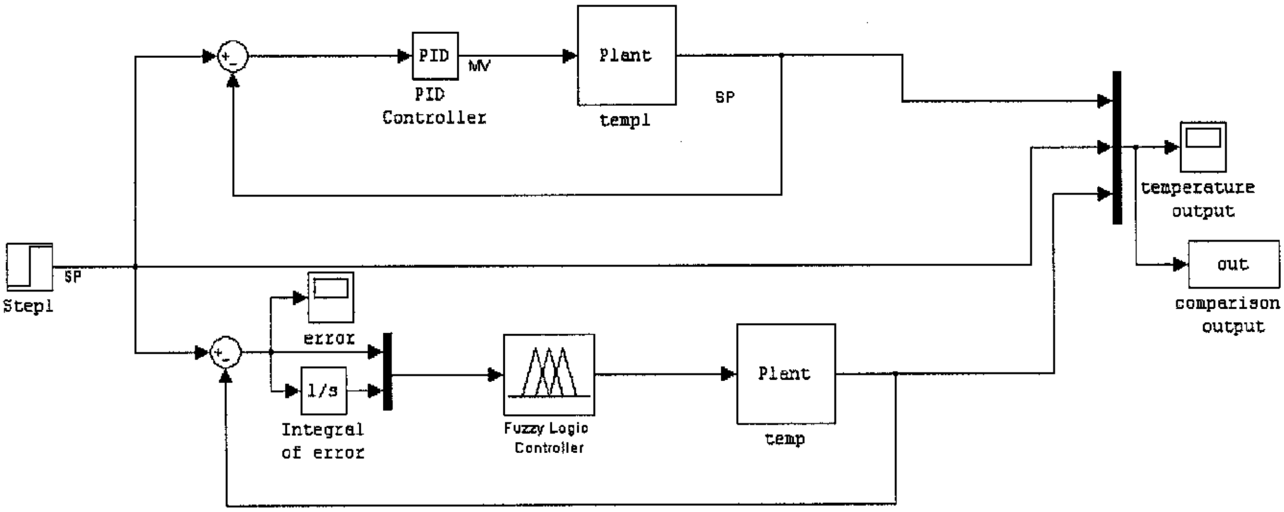


Figure 33 Simulink block diagram for comparison of temperature response between PI and fuzzy logic controller

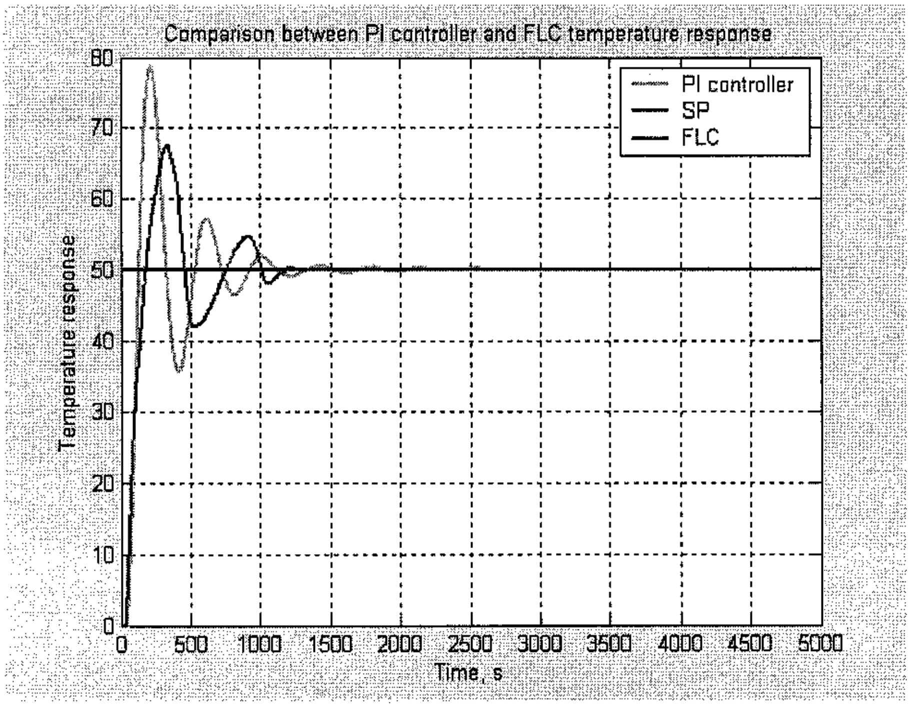


Figure 34 Comparison between PI controller and FLC temperature response

4.5 Discussion

4.5.1 Model

The modeling part of this study has proved that the model developed is a good model even though it is a first order whereas commonly this heat exchanger process is at second order. This shows that the first order with dead time model is adequate for process control analysis and design. However, if the model is done at higher order, the author believes that the response will be more accurate.

4.5.2 Controller

From the study, it can be concluded that PID controller is robust and useful in the process control. It has been widely used for a long time until now. However, there are areas that cannot be handled by the PID controller such as the process becomes more complicated. FLC can be used where the PID controller is not giving good response and FLC can cover a wider range of processes because it is using human-like techniques to define the process. Since FLC is still new compared to PID controller, many industrial player thinks it is not economical to change their controllers in the plant because so far the PID controller has work well for their process. However, the author thinks that the industrial player should look for the usefulness of FLC and the author believes that it will bring improvement to their system if they change the conventional controller to the FLC or any new intelligent control approach.

4.5.3 Fuzzy inference method

This study has put main emphasis on the Mamdani fuzzy inference method. Mamdani fuzzy controller is good for capturing the expertise of a human operator and it is also easy understandable by a human expert. Besides, it is commonly used in the industry and simpler to formulate rules.

The main difference with Sugeno fuzzy inference method is the consequent/output part, where for Sugeno it is a mathematical function/singleton while the Mamdani are fuzzy sets. In Mamdani, each rule output is described by a membership functions.

From the control performance analysis, the author has proved that, PI-like fuzzy logic controller using Mamdani inference method is having better control performance compared to the PI controller. Thus, the author has achieved the objective of this study that is to prove that intelligent control is better than the conventional control.

4.5.4 Drawbacks of FLC

However, FLC have some drawbacks compared to the PID controller. It is time consuming to design the FLC system such as to develop the membership functions and to construct the fuzzy rules. From the author experiences, designing the membership functions is the most challenging part and it also consumed a lot of time. However, with the result obtained, it is worth it to take up the challenge of designing the FLC system.

It also shown that, when the fuzzy sets value is changed, the membership functions of the system need to be changed by shifting the shape value to the left or to the right of the previous value.

It is also known that it is hard to move over from the conventional PID controller to the FLC system, since PID controller is already widely accepted worldwide in the process control industry. There are many factors need to be considered by the industry before move over to this new FLC controller.

RECOMMENDATIONS

Disturbance

For this study, the author only uses one type of disturbance that is step response to evaluate the process response. Since, there are a lot of disturbance types at the real plant, the author recommend to evaluate the process response with various disturbances to analyze the performance of FLC response. With various disturbances, the robustness of the FLC can be evaluated, thus prove that the FLC is better than the conventional PID controller.

Multiple Input Multiple Output (MIMO)

For this study, the author is using single input and single output system (SISO). Even though previously, the author is using cascade control strategy, the difficulties in the FLC development of the cascade control strategy with time constraint, make the author to concentrate on the SISO system. Thus, the author hope the FLC can be implemented with multiple inputs and single output or multiple outputs system. This MIMO system ensures that, the analysis is considering all the parameters in the heat exchanger plant thus it is much better for the evaluation of control performance of the plant.

Physical Implementation

The study done is merely on the software and simulations part of the system. Thus, with the incoming of fuzzy hardware kit, which is still under the procurement process, the FLC design can be implemented for online monitoring of process response at the pilot plant. This will further strengthen the analysis done in this study.

CONCLUSION

The first objective of modeling and simulation of heat exchanger process is completed. Using empirical modeling, the model of the heat exchanger process is obtained and the output response is quite similar to the actual process response. Thus, this model can be used in further study of the heat exchanger process improvement at the pilot plant.

The second objective, which designing the controller for both PID and fuzzy control is completed. Both controllers gave good output response after some fine tuning. The most important part is that, both of the controllers are proven stable. The decision to use whether, the conventional PID controller or intelligent fuzzy controller is dependent on what is economically wise for the system and also the need of the industry. Perhaps, with the usage of fuzzy logic controller it can further enhance the system performance.

From the comparative study, it is shown that fuzzy logic controller has better control performance and output response compared to the PI controller. The FLC is designed using Mamdani inference method where it is using the idea or the expertise of a human operator.

Modeling and simulation of heat exchanger is one of useful learning tools to understand process control technique in process industries. Throughout the modeling process, the author has the opportunity to go in depth to understand the process control technique especially in process industry using the heat exchanger process. It is important to understand the correlation between the input and output of the system in order to design a process control strategy. In a conclusion, this study has given the author better understanding in controlling process type of system especially the heat exchanger process. It also gave the author another way of looking the process control approach especially using the intelligent control approach.

REFERENCES

1. http://www.eng.fsu.edu/~schreiber/lab_manual/facts/heat_transfer_theory.pdf
2. J. P. Holman, "Heat exchangers", *Heat Transfer, 2001*, Eighth SI Metric Edition, McGraw-Hill
3. Hao Ying, "Basic Fuzzy Mathematics for Fuzzy Control and Modeling", *Fuzzy Control and Modeling: Analytical Foundations and Applications*, 2000, IEEE Press Series on Biomedical Engineering.
4. http://www.minco.com/site/wordpdf/pdf00201-fuzzy.pdf
5. Guanrong Chen, Trung Tat Pham, "Fuzzy Logic Theory", *Introduction to Fuzzy Sets, Fuzzy Logic and Fuzzy Control Systems*, 2001, CRC Press LLC.
6. fuzzyTECH 5.5 Editions User's Manual, INFORM GmbH / Inform Software Corp., 2001.
7. D. Hanselman, B. Littlefield, "Introduction," *Mastering MATLAB 6*, 2001, Prentice Hall Inc.
8. J. B. Dabney, T. L. Harman, "What is Simulink", *Mastering Simulink 4*, 2001, Prentice Hall Inc.
9. Thomas E. Marlin, "Chapter 6: Empirical Model Identification", *Process Control: Designing Processes and Control Systems for Dynamic Performance*, 2nd Edition, 2000, McGraw Hill.
10. Thomas E. Marlin, "Chapter 9: PID Controller Tuning for Dynamic Performance", *Process Control: Designing Processes and Control Systems for Dynamic Performance*, 2nd Edition, 2000, McGraw Hill.

11. Leonid Reznik, "The structure and operation of a fuzzy controller", *Fuzzy Controllers*, 1997, Newnes
12. Leonid Reznik, "Basic mathematical concepts of fuzzy sets", *Fuzzy Controllers*, 1997, Newnes
13. Leonid Reznik, "Fuzzy sets, logic and control", *Fuzzy Controllers*, 1997, Newnes
14. MATLAB 6.5, *MATLAB Help Manual*, The MathWorks Inc. 1984 – 2003
15. Mohd Isa bin Othman, *Modeling and Simulation of Heat Exchanger*, 2004, Universiti Teknologi PETRONAS
16. Kevin M. Passino, Stephen Yurkovich, Fuzzy Control: The Basics, *Fuzzy Control,* 1998, Addison-Wesley

APPENDICES

A. Calculations of tuning parameters

B. PI controller simulation data

C. Control performance analysis calculations

APPENDIX A

Calculations of tuning parameters

1. Cohen & Coon open loop tuning method

The Cohen & Coon open loop tuning parameter formula

$$K_C = \left[\frac{1}{R \bullet K_P}\right]\left[\frac{9}{10} + \frac{R}{12}\right]$$

$$T_I = T_d \bullet \left[\frac{30 + 3R}{9 + 20R}\right]$$

$\mathrm{Kp} = 0.17,\ \tau = 120s\ \ \vartheta = 40s$

Gain, K_C

$$K_C = \left[\frac{1}{0.17 \bullet 0.3333}\right]\left[\frac{9}{10} + \frac{0.3333}{12}\right]$$

$$K_C = 17.6488 \times 0.9278$$

$$K_C = 16.3741$$

Integral time, T_I

$$T_I = 40 \bullet \left[\frac{30 + (3 \times 0.3333)}{9 + (20 \times 0.3333)}\right]$$

$$T_I = 40 \times \frac{30.9999}{15.666}$$

$$T_I = 102.1318$$

$$\frac{1}{T_I} = \frac{1}{102.1318} = 0.009791 \approx 0.01 \quad (in\ MATLAB)$$

2. Ziegler – Nichols open loop tuning method

PI only

$$K\,c = \left(\frac{0.9}{Kp}\right)\left(\frac{\tau}{\theta}\right)$$

$$T\,i = 3.3\,\theta$$

$$K\,c = \left(\frac{0.9}{0.17}\right)\left(\frac{120}{40}\right)$$

$$K\,c = 5.2941 \times 3$$

$$K\,c = 15.8823$$

$$T\,i = 3.3 \times 40 = 132$$

$$\frac{1}{T\,i} = \frac{1}{132} = 0.007576$$

3. Ciancone correlation

The fraction dead time $\frac{\theta}{\theta + \tau}$

Then, the dimensionless tuning values is calculated from the graphs [*Figure 35*] for $K_c\ K_p$, $\frac{Ti}{\theta + \tau}$, and $\frac{Td}{\theta + \tau}$

$\frac{\theta}{\theta + \tau} = 0.25$ $\quad Kc = 7.3529$ $\quad Ti = 147.2$ $\quad \frac{1}{Ti} = 0.00679$

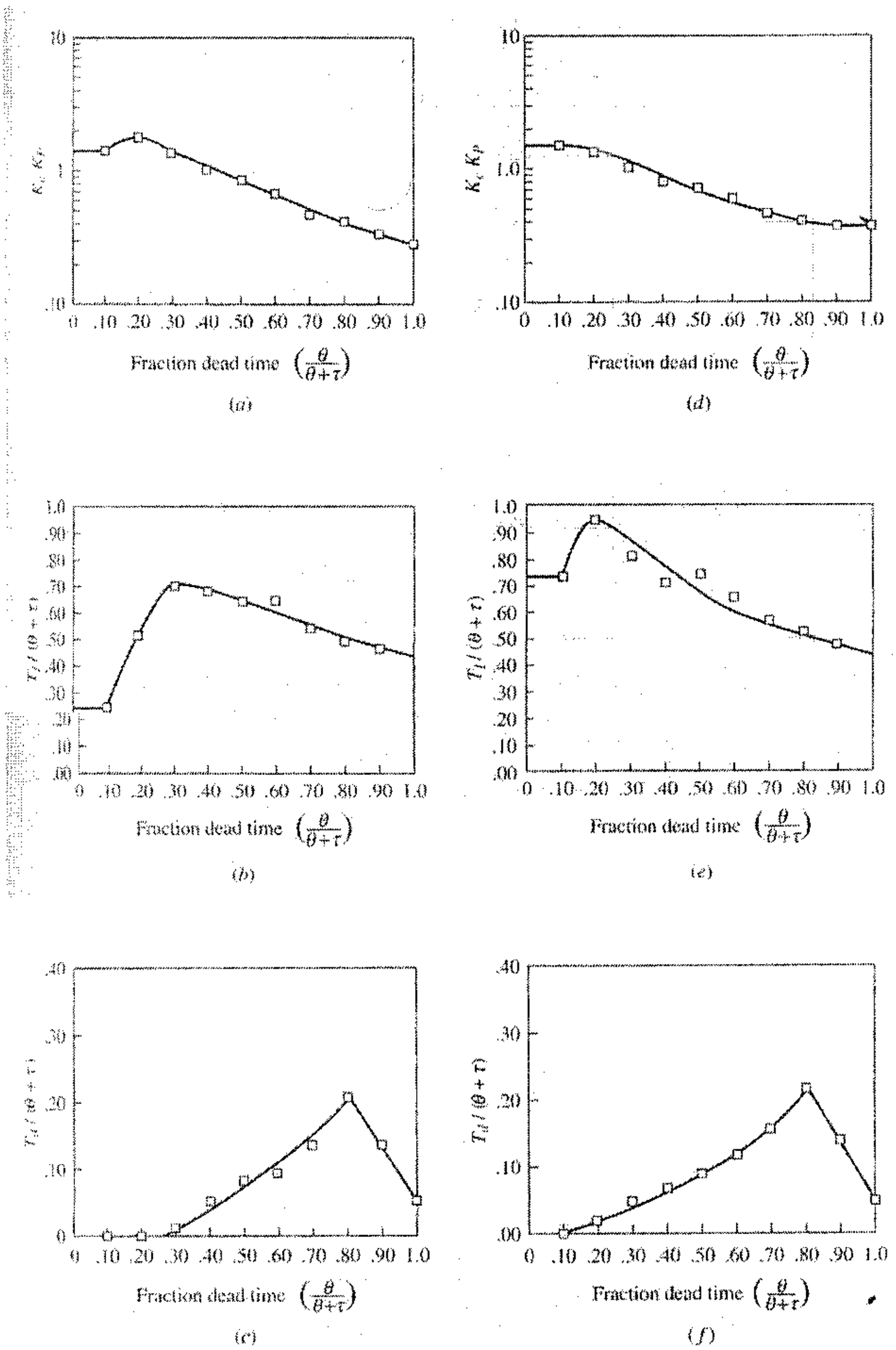


Figure 35 Ciancone correlations for dimensionless tuning constants, PID algorithm. For disturbance response: (a) control system gain, (b) integral time. (c) derivative time. For set point response: (d) gain, (e) integral time, (f) derivative time.

APPENDIX B

PI controller simulation data

Temperature error	Integral of temperature error	% of valve opening
0	0	0
50	0	400
50	2.20E-11	400
50	357.14	450
50	357.14	450
50	357.14	450
50	714.29	450
50	1071.4	450
50	1785.7	450
35.407	3022.6	450
24.236	3730.3	450
-0.44928	4572.6	450
-13.124	3973.6	450
-13.177	3967.3	450
-13.177	3967.3	450
-16.751	2669.9	239.77
-3.1398	1776.7	223.62
2.5063	1784	269.82
3.7663	1968.8	305.77
1.5653	2157.6	314.59
-1.0092	2171	295.87
-0.6872	2087.9	286.81
0.24956	2070.8	291.91
0.31897	2100.6	296.63
-0.06763	2112.3	295.19
-0.13209	2101.7	293.18
0.017151	2096.2	293.6
0.055831	2100.1	294.45
-0.00202	2102.7	294.36
-0.02305	2101.3	294
-0.00124	2100.1	294.01
0.009328	2100.6	294.16
0.001344	2101.1	294.17
-0.0037	2101	294.11
-0.00088	2100.7	294.1
0.001436	2100.8	294.12
0.00049	2100.9	294.13
-0.00054	2100.9	294.12
-0.00025	2100.8	294.11
0.0002	2100.8	294.12
0.00012	2100.8	294.12
-7.14E-05	2100.8	294.12

Temperature error	Integral of temperature error	% of valve opening
-5.55E-05	2100.8	294.12
2.42E-05	2100.8	294.12
2.49E-05	2100.8	294.12
-7.70E-06	2100.8	294.12
-1.09E-05	2100.8	294.12
2.18E-06	2100.8	294.12
-3.22E-07	2100.8	294.12
-5.49E-08	2100.8	294.12
6.36E-08	2100.8	294.12
1.27E-07	2100.8	294.12
3.38E-08	2100.8	294.12
-4.89E-08	2100.8	294.12
-1.82E-08	2100.8	294.12
1.84E-08	2100.8	294.12
9.11E-09	2100.8	294.12
-2.09E-09	2100.8	294.12
4.65E-06	2100.8	294.12
-4.74E-07	2100.8	294.12
-1.95E-06	2100.8	294.12
1.96E-08	2100.8	294.12
8.00E-07	2100.8	294.12

APPENDIX C

Control performance analysis calculations

Formulas

Decay ratio $= \frac{B}{A}$

CV Overshoot $= \frac{C_{max} - C_{final}}{C_{final}}$

MV Overshoot $= \frac{C}{D}$

Cohen & Coon open loop tuning method

Decay ratio $= n / a$

CV Overshoot $= \frac{C_{max} - C_{final}}{C_{final}} = \frac{50.28 - 50}{50} \times 100 = 0.56\%$

MV Overshoot $= \frac{C}{D} = \frac{838 - 276.38}{276.38} \times 100 = 203.21\%$

Ziegler – Nichols open loop tuning method

Decay ratio $= n / a$

CV Overshoot $= \frac{C_{max} - C_{final}}{C_{final}} = \frac{51.25 - 50}{50} \times 100 = 2.5\%$

MV Overshoot $= \frac{C}{D} = \frac{814.1 - 249}{249} \times 100 = 227\%$

Ciancone correlation

Decay ratio $= n/a$

CV Overshoot $= n/a$

$$\text{MV Overshoot} = \frac{C}{D} = \frac{367.6-292}{292} \times 100 = 25.89\%$$

Fine tuning PI parameters

$$\text{Decay ratio} = \frac{6.75}{26.68} \times 100 = 25.30\%$$

$$\text{CV Overshoot} = \frac{C_{\max} - C_{\text{final}}}{C_{\text{final}}} = \frac{76.65-50}{50} \times 100 = 53.36\%$$

$$\text{MV Overshoot} = \frac{C}{D} = \frac{728-294}{294} \times 100 = 147.62\%$$

Fuzzy logic controller

$$\text{Decay ratio} = \frac{4.822}{19.158} \times 100 = 25.17\%$$

$$\text{CV Overshoot} = \frac{C_{\max} - C_{\text{final}}}{C_{\text{final}}} = \frac{69.15-50}{50} \times 100 = 38.3\%$$

$$\text{MV Overshoot} = \frac{C}{D} = \frac{457-294.11}{294.11} \times 100 = 53.34\%$$