

**EMPIRICAL MODELLING AND FUZZY CONTROL  
SIMULATION OF A HEAT EXCHANGER**

By

MOHD ISA BIN OTHMAN (1173)

**DISSERTATION**

Submitted to the  
Electrical & Electronics Engineering Programme  
in partial fulfillment of the requirements for the  
BACHELOR OF ENGINEERING (Hons)  
(ELECTRICAL AND ELECTRONICS ENGINEERING)

June 2004

Universiti Teknologi PETRONAS

Bandar Seri Iskandar

31750 Tronoh

Perak Darul Ridzuan

t  
TJ  
263  
M/697  
2004

i

1. Heat exchangers - Design and construction  
2. EEE - Tests

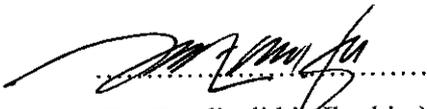
# **CERTIFICATION OF APPROVAL**

## **EMPIRICAL MODELLING AND FUZZY CONTROL SIMULATION OF A HEAT EXCHANGER**

By  
Mohd Isa bin Othman

A project dissertation submitted to the  
Electrical & Electronics Engineering Programme  
Universiti Teknologi PETRONAS  
in partial fulfillment of the requirement for the  
BACHELOR OF ENGINEERING (Hons)  
(ELECTRICAL AND ELECTRONICS ENGINEERING)

Approved by,



(En. Rosdiazli bin Ibrahim)

Mr. Rosdiazli Ibrahim  
Lecturer  
Electrical & Electronic Engineering  
Universiti Teknologi PETRONAS

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 is my own except as specified in the references and acknowledgements, and the original work contained herein have not been undertaken or done by unspecified sources or persons.

  
.....

(MOHD ISA BIN OTHMAN)

## ABSTRACT

A heat exchanger is one of the most important systems that have been installed in many process plants. It is a device that transfers heat from liquid to another without allowing them to mix. In order to ensure its smooth operation, modelling and simulation can be made so that its performance can be analyzed and improved.

At Process Control Lab, there is no simulation model for laboratory-scale heat exchanger pilot plant. Most of the time, the plant is being used for ordinary laboratory practice and the performance of this plant is not being analyzed. This project is therefore conducted to study the plant behavior and to optimize its performance by simulating it with new type of controller.

The first goal of this project is to model the heat exchanger pilot plant by using empirical modelling method. It will yield the plant transfer function,  $G_p$  that can be used for temperature controller analysis. Besides empirical modelling, mathematical modelling is also being carried out to study the heat exchanger behavior. By having the model, there is an alternative way to obtain forecasted data and result without extra cost.

The second part of this project is to analyze the model temperature controller performance. Two controllers are being compared, namely PID and Fuzzy Logic Controller. First, PID controller is tested to yield the best tuning parameters for control valve. Ziegler-Nichols and fine tuning method is used to serve this purpose. Next, the data from PI controller simulation is fed into ANFIS toolbox in MATLAB for adaptive learning process. The FIS generated by ANFIS is based on Takagi-Sugeno fuzzy model. The FIS which is subsequently used by the Fuzzy Logic Controller will imitate the PI controller performance and perform based on range of data it has been trained before by ANFIS toolbox. Finally, the comparison between both controllers is concluded where Fuzzy Logic Controller is successfully imitating the PI controller with slightly better performance in terms of rise time, settling time and overshoot percentage.

## ACKNOWLEDGMENT

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

First, I would like to express my greatest gratitude to my supervisor, Mr. Rosdiazli bin Ibrahim for his expert guidance, attention and suggestions, supports and advices regarding the project and difficulties faced during the project execution.

Not forget to mention, I would like to thank several lecturers who also help me during the project especially to Mr. Fawnizu for his assistance in mathematical modeling development using MATLAB S-function and Mr. Nasir from Chemical Engineering department.

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

In addition, I am grateful to Mr. Jefferi bin Kamaruddin, Mr. Marzuki bin Nordin, and Mr. Azmin bin Ishak because of given freely of their time to discuss together in improving materials of my 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 successful.

Last but not least, my appreciation goes out to the 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 until this.

## TABLE OF CONTENTS

<b>CERTIFICATION</b>	i
<b>ABSTRACT</b>	iii
<b>ACKNOWLEDGEMENT</b>	iv
<b>TABLE OF CONTENTS</b>	v
<b>LIST OF FIGURES</b>	viii
<b>LIST OF TABLES</b>	ix
<b>LIST OF APPENDICES</b>	ix
<b>ABBREVIATIONS AND NOMENCLATURES</b>	x
<b>CHAPTER 1: INTRODUCTION</b>	<b>1</b>
1.1 Background of Study	1
1.2 Problem Statement	2
1.3 Objectives and Scope of Study	2
1.3.1. Objectives	2
1.3.2. Scope of Study	2
1.3.3. The Relevancy of the Project	3
<b>CHAPTER 2: LITERATURE REVIEW</b>	<b>4</b>
2.1 Process Flow of Heat Exchanger Pilot Plant	4
2.2 Mathematical Model of Heat Exchanger	5
2.3 Adaptive Network-based Fuzzy Inference System	7
2.4 Training and Test Validation of ANFIS	8
2.5 Training Method of ANFIS	9
<b>CHAPTER 3: METHODOLOGY / PROJECT WORK</b>	<b>11</b>
3.1 Procedure Identification	11
3.1.1. Overall Project Flow Chart	11
3.1.2. Mathematical Modelling	12
3.1.3. Empirical Modelling	13
3.1.4. ANFIS Modelling	14

3.2	Tools and Software	15
3.2.1.	MATLAB-Simulink	15
3.2.2.	Fuzzy Logic Toolbox ( <i>anfisedit</i> ).	16
3.2.3.	Heat Exchanger Pilot Plant with DCS	18
<b>CHAPTER 4:</b>	<b>RESULT AND DISCUSSION</b>	<b>19</b>
4.1	Dynamic Behavior of Mathematical Model	19
4.2	Empirical Model	22
4.2.1.	Open Loop Test	24
4.2.1.	Closed Loop Test	25
4.3	ANFIS Training.	29
4.4	Comparison: PID versus Fuzzy Logic Controller	33
4.5	Discussion	37
<b>CHAPTER 5:</b>	<b>CONCLUSION AND RECOMMENDATION</b>	<b>42</b>
5.1	Relevancies to the Objectives	42
5.2	Next Project Recommendations	43
5.3	Conclusion	44
<b>REFERENCES</b>		<b>45</b>
<b>APPENDICES</b>		

## LIST OF FIGURES

- Figure 2.1 Simplified Process Flow Diagram for Heat Exchanger
- Figure 2.2 Cocurrent Shell and Tube Heat Exchanger
- Figure 3.1 Overall Project Flow Chart
- Figure 3.2 Mathematical Modelling Procedure
- Figure 3.3 Procedure for Empirical Transfer Function Model Identification
- Figure 3.4 Flow chart for ANFIS modelling
- Figure 3.5 Graphical User Interface (GUI) of *anfisedit* toolbox
- Figure 4.1 Basic model block for Shell and Tube Heat Exchanger
- Figure 4.2 Components masked under Heat Exchanger Block
- Figure 4.3 Mathematical Model of Heat Exchanger with PID Controller
- Figure 4.4 Process Reaction Curve from HEX Mathematical Model
- Figure 4.5 Temperature Response of Mathematical Model with PID controller
- Figure 4.6 Manipulated Variable Behavior of Mathematical Model
- Figure 4.7 Heat Exchanger Input and Output Plot
- Figure 4.8 Simulink block diagram of HEX Model
- Figure 4.9 Open Loop Response of Empirical Model
- Figure 4.10 PV and MV Response from Different Tuning Parameters
- Figure 4.11 PV Response for  $K_C = 1.763$  and  $T_i = 30$
- Figure 4.12 MV Response for  $K_C = 1.763$  and  $T_i = 30$
- Figure 4.13 Grid partition (left) and Sub-Clustering (right) network structure
- Figure 4.14 ANFIS Plot of Math Model (above) & Empirical Model (below)
- Figure 4.15 Surface View of Math Model (left) & Empirical Model (right)
- Figure 4.16 Rules View of Math Model (above) & Empirical Model (below)
- Figure 4.17 Comparison Test for Mathematical Model
- Figure 4.18 Comparison Test for Empirical Model
- Figure 4.19 Fuzzy Logic Controller step response
- Figure 4.20 PID controller configuration
- Figure 4.21 Performance test on FLC using FIS from Mamdani method

## LIST OF TABLES

Table 4.1	Ziegler-Nichols open-loop tuning based on reaction curve
Table 4.2	PI tuning parameters for Empirical Model
Table 4.3	Characteristics from LTI viewer (PV)
Table 4.4	Characteristics from LTI viewer (MV)
Table 4.5	Retuning PI parameters with variable $T_i$ value (PV)
Table 4.6	Retuning PI parameters with variable $T_i$ value (MV)
Table 4.7	Average Error of Different Number of MF
Table 4.8	Average Error of Different Type of Optimization Method
Table 4.9	Comparison Data for FLC and PI controller

## LIST OF APPENDICES

Appendix 1	Project Gantt chart
Appendix 2	MATLAB S-Function file for Mathematical Model of Heat Exchanger
Appendix 3	Modification on Mathematical Model of Heat Exchanger
Appendix 4	Process Reaction Curve from Open Loop Test
Appendix 5	Simulink Block Diagram for Empirical Model with PID Controller
Appendix 6	PI Tuning Parameters Calculation for Giancone and Lopez
Appendix 7	PV and MV Characteristics from Different Tuning Parameters
Appendix 8	Fine Tuning on PI Controller – (retuning $T_i$ )
Appendix 9	Simulink diagram of FLC and PID Controller for Math Model
Appendix 10	Simulink diagram of FLC and PID Controller for Empirical Model

## ABBREVIATIONS AND NOMENCLATURES

HEX	Heat Exchanger
DCS	Distributed Control System
PID	Proportional, Integral and Derivative
GUI	Graphical User Interface
TT	Temperature Transmitter
TIC	Temperature Indicator Controller
FT	Flow Transmitter
FCV	Flow Control Valve
EH	Electrical Heater
P	Pump
LCV	Level Control Valve
LT	Level Transmitter
VE	Vessel
ANFIS	Adaptive Network-based Fuzzy Inference System
UTP	Universiti Teknologi PETRONAS
SP	Set Point
PV	Process Variable
MV	Manipulated Variable
MF	Membership Function
FLC	Fuzzy Logic Controller

# CHAPTER 1

## INTRODUCTION

### 1.1. Background of Study

By definition, heat exchanger is a device that transfers heat from liquid to another without allowing them to mix. The word “exchanger” really applies to all types of equipment in which heat is exchanged but it is often used specifically to denote equipment in which heat is exchanged between two process streams <sup>[1]</sup>.

In UTP, there are several pilot plants available for students to study the process control topic. The pilot plant designs are closely resembles the actual plant with all the transmitter and control valves but in the laboratory-scale. One of the plants that are going to be modeled in this project is the heat exchanger plant. The main components of this plant are heat exchanger, cooling tower, heater, heating medium tanks and product tanks.

In general, this project aims at modelling and simulation of heat exchanger pilot plant. There are various methods can be used to model the plant for example neural network, system identification, and empirical method. For this project, the modelling part is done based on the heat exchanger inputs and outputs. Simulation part involves with validation and testing the functionality of the model.

By using the model developed, the temperature controller performance will be analyzed. Existing PID controller will be tested and compared to new Fuzzy Logic Controller (FLC). Adaptive learning process will be used to create FLC fuzzy inference system. The outcome from this analysis can be used to optimize the heat exchanger performance. The introduction of FLC is expected to be promising since it has many features that can make the performance better than the existing PID controller.

## **1.2. Problem Statement**

Currently, there is no model to heat exchanger pilot plant, which can be used for students to analyze its overall performance. Modelling here refers to the process of analysis and synthesis at arriving to a suitable mathematical description of the plant parameters.

By having this project, a model can be developed and it can be used for further analysis particularly for optimizing the heat exchanging process. The advantage of this model is we can evaluate any part of equipment in the system and predict its output by just key-in an input value.

At the Process Control lab, the plant is used mostly for temperature control experiment using existing PID controller. Based on experience, there are several occasions where the temperature controller performance exhibit poor result. Therefore, the PID controller analysis and feasibility study of the new controller implementation using Fuzzy Logic is required. Perhaps, its control strategy can be improved and smooth operation can be ensured.

## **1.3. Objectives and Scope of Study**

### **1.3.1. Objectives**

- ❑ To model and simulate the heat exchanger pilot plant.
- ❑ To test the existing PID controller performance.
- ❑ To improve performance by using Fuzzy Logic controller.

### **1.3.2. Scope of Study**

The modelling and simulation will be done on UTP's heat exchanger pilot plant. The study will be based on input and output of the heat exchanger together with its controller action. The modelling part is mostly involves with empirical and mathematical modelling approach. Mathematical model is more detail compared to empirical model since it involves with more calculations.

The accuracy of the model will be observed based on its output reaction to input variation. To validate the model, MATLAB Simulink is used to simulate its behavior based on certain input variation. Comparison is made based on collected real-time data from plant experiment. The controller design is also being implemented using Simulink specifically the fuzzy logic toolbox. After controller design, the performance is being tested and compared. Several criteria are being measured such as rise and settling time, overshoot percentage and peak amplitude.

### 1.3.3. The Relevancy of the Project

The transfer of heat to and from process fluids is an essential part of most chemical processes. The most commonly used type of heat-transfer equipment is the ubiquitous shell and tube exchanger [1]. Currently, there are lots of research has been done to improve heat exchanger performance and its control system. So, by modelling the pilot plant heat exchanger, students are expected to be familiar when encounter real process control.

By simulating the model and test the controller performance, the outcome of this project is very promising in future development of new breed process controller. Test on the existing PID controller and redesign it using proposed Fuzzy Logic technology is very useful in providing better controller performance.

In summary, this project can be considered as an educational tool to familiarize with plant activities. Analysis on controller design is also a beneficial step to enhance plant process control environment.

## CHAPTER 2

### LITERATURE REVIEW

#### 2.1. Process Flow of Heat Exchanger Pilot Plant

The laboratory-scale plant is divided into two sections, heated and heating medium sides. The right side is the heat supplying section where hot water (heating medium) is heated by electrical heater. The hot water then is fed into the heat exchanger shell side by using pump P613. There is one control valve (FY631) available to control the amount of hot water flowing into heat exchanger shell side.

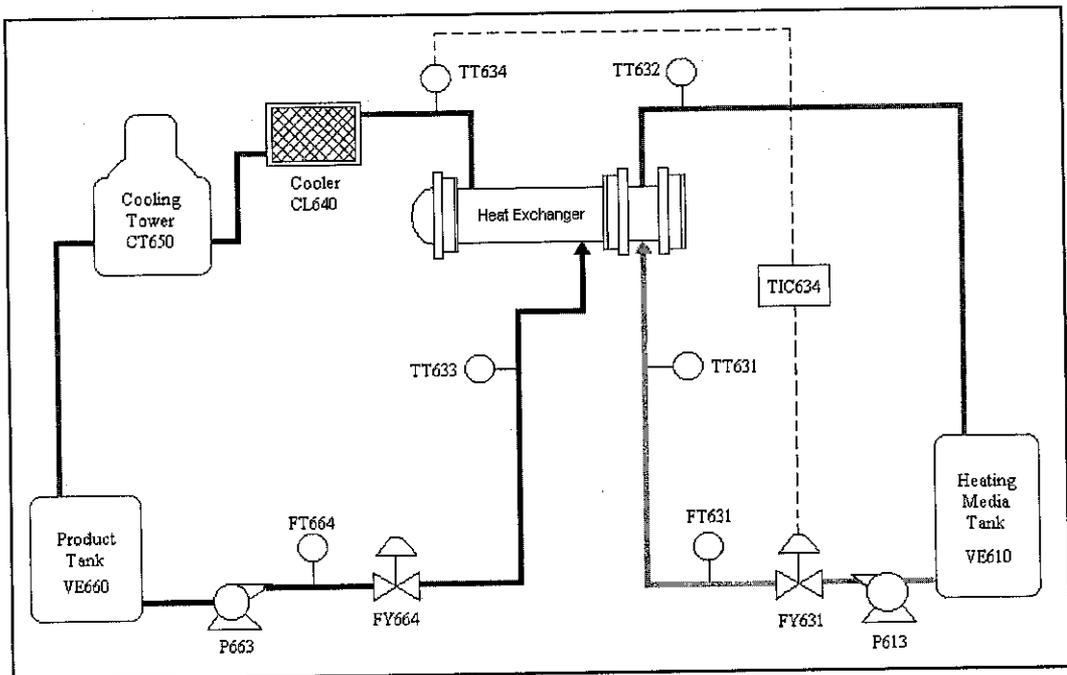


Figure 2.1: Simplified Process Flow Diagram for Heat Exchanger

The left side is the heat receiving section where cold water (heated medium) is pumped into heat exchanger tube side by using P663. Inside the heat exchanger, the cold water will experience heat-exchanging process, which will be monitored by its outlet temperature transmitter, TT633. This transmitter is the sensing element for temperature controller module, TIC634.

The temperature controller is linked to control valve FY631. Therefore, the temperature changes at TT634 will be affected by control valve FY631 behavior. This system is referred as Single Input Single Output (SISO) or single loop control system.

There are other important parameters available at the plant but for this case, the parameters are kept at constant values for model simplification. For example, the control valve at heated medium side, FY664 which is configured to open at 25% valve opening. Thus, flow rate at FT664 is about 1.2m<sup>3</sup>/h. In addition, the temperature of hot water at TT631 is set up by heater to be at approximately 60°C.

## **2.2. Mathematical Model of Shell and Tube Heat Exchanger**

A heat exchanger is one of the most important systems that have been installed in many process plants. Modelling a heat exchanger is indeed a difficult task <sup>[2]</sup> for the following reasons:

- Since the dynamics of the heat exchanger is described by partial differential equations, it is truly an infinite dimensional system, which makes it difficult for theories developed for lumped systems to be applied.
- Using certain approximation techniques, the resulting system with acceptable accuracy usually has too high an order for easy dynamic analysis and control design.

The following paragraphs will illustrate the modelling of nonlinear heat exchanger using lumped approximation technique <sup>[3]</sup>. 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 2.2. The temperature and the flow rate of the liquid not only change with time but also change along the axial direction  $x$ .

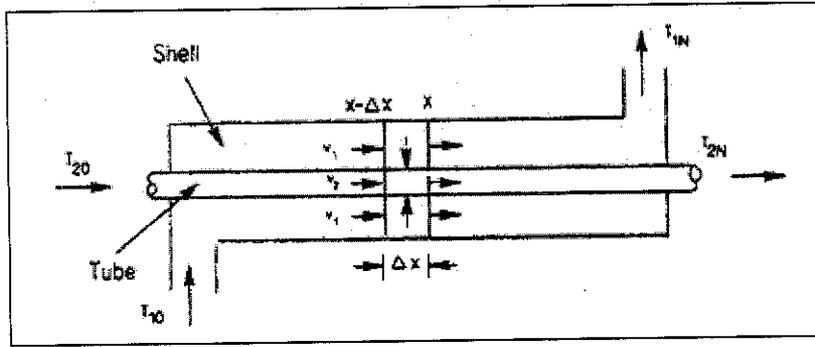


Figure 2.2: Co-current Shell and Tube Heat Exchanger

To model the heat exchanger, several assumptions is made which are:

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

Consider an incremental element  $\Delta x$ , along  $x$  to which the principle of conservation of energy is applied. Energy balance on the tube side (either shell or inner tube) fluid, in the element at a particular distance  $x-\Delta x$  and time,  $t$  is given by:

$$\left[ \begin{array}{c} \text{accumulation} \\ \text{of energy in} \\ \text{the element} \\ \Delta x \end{array} \right] = \left[ \begin{array}{c} \text{convective} \\ \text{flow of} \\ \text{energy} \\ \text{into the} \\ \text{element} \\ \Delta x \end{array} \right] - \left[ \begin{array}{c} \text{convective} \\ \text{flow of} \\ \text{energy} \\ \text{out the} \\ \text{element} \\ \Delta x \end{array} \right] + \left[ \begin{array}{c} \text{heat transfer} \\ \text{to the} \\ \text{element} \\ \Delta x \end{array} \right]$$

...(1)

If the velocity of the fluid averaged across the tube is constant, i.e. independent of  $x$ , then:

$$\frac{\delta(a\Delta x\rho C_p T)}{\delta t} = a v \rho C_p [T_{x-\Delta x} - T_x] + U \pi d \Delta x \Delta T$$

...(2)

where,

- a = cross sectional area of tube (m<sup>2</sup>)
- ρ = fluid density (Kg/m<sup>3</sup>)
- C<sub>p</sub> = heat capacity (Cal/Kg K)
- T = temperature of tube considered (K)
- t = time (sec)
- v = fluid velocity (m/sec)
- U = overall heat transfer coefficient (Cal/sec m<sup>2</sup>K)
- d = internal diameter of the tube considered (m)
- ΔT = change in temperature (K)
- Δx = incremental distance (m)

By using subscripts 1 and 2 to correspond to the shell tube and the inner tube respectively, and considering the limit as Δx → 0 and assuming no acceleration of fluid, equation (2) can be used to derive the following equation for the shell tube:

$$\frac{\partial T_1}{\partial t} = -v_1 \frac{\partial T_1}{\partial x} - \frac{U \pi d}{a_1 \rho_1 C_{p1}} (T_1 - T_2) \quad \dots(3)$$

Similarly the energy balance for the inner tube side fluid can be derived and the partial differential equation is as given below.

$$\frac{\partial T_2}{\partial t} = -v_2 \frac{\partial T_2}{\partial x} - \frac{U \pi d}{a_2 \rho_2 C_{p2}} (T_1 - T_2) \quad \dots(4)$$

Both of these linear differential equations will be used to develop mathematical model of heat exchanger.

### 2.3. Adaptive Network-based Fuzzy Inference System (ANFIS)

ANFIS derives its name from Adaptive Network-based Fuzzy Inference System. By using input and output data set, the toolbox function ANFIS are able to construct a fuzzy inference system (FIS) whose membership function parameters are tuned (adjusted) using either backpropagation algorithm alone, or in combination with least square type of method. This allows fuzzy system to learn from the data they are modeling.

ANFIS network-type structure is similar to that of neural network. It maps inputs through input membership functions and associated parameters and then through output membership functions and associated parameters to outputs, can be used to interpret the input/output map.

The parameter associated with the membership functions will change through the learning process. The computation of these parameters (or their adjustment) is facilitated by a gradient vector, which provides a measure of how well the fuzzy inference is modeling the input/output data for a given set of parameters.

Once the gradient vector is obtained, any of several optimization routines could be applied in order to adjust the parameters as to reduce some error measure (usually defined by the sum of squared difference between actual and desired outputs). ANFIS uses either backpropagation or combination of least square estimation and backpropagation for membership function parameter estimation.

ANFIS is much more complex than the fuzzy inference systems and is not available for all fuzzy inference system options. Specifically, *anfis* only support Sugeno-type systems, and these must be:

- First or zero order Sugeno-type system.
- Single output, obtained using weighted average defuzzication (linear or constant output membership functions)

An error occurs if FIS structure does not comply with these constraints.

#### **2.4. Training and Test Validation of ANFIS**

Neuro-fuzzy (ANFIS) training should be selected to cover entire region where the network is expected to operate. Usually large amount of data is collected and a subset of data is used to train the network. Another subset of data is used as a test data to verify the correct generalization of the network.

If the ANFIS does not generalize well on several data points, that data is added to training and the network is retrained. This process continues until the performance of the network is acceptable. The training data should bind the entire region because a Neuro-fuzzy's performance cannot be relied upon outside the operating region.

## **2.5. Training Method of ANFIS**

In ANFIS, there are two methods that ANFIS learning employs for updating membership function parameters, which are backpropagation and hybrid method. In general, the backpropagation method applies for all parameters (a steepest decent method). On the other hand, a hybrid method consists of backpropagation for the parameters associated with the input membership function, and least square estimation for the parameters associated with the output membership function.

Training of artificial neural network involves two passes: forward and reverse. In the forward pass, the input signals propagate from the network input to the output. The calculation of the output is carried out, layer by layer. The output of one layer is the input to the next layer. On the other hand, in the reverse pass, the calculated error signals propagate backward through the network, where they are used to adjust the weights. The training becomes complicated, as the middle layer neurons have no target values. Thus, the backpropagation method is applied.

Backpropagation is a systematic method for training multiple (three or more) layer of neural network. The multilayer networks have greater representational power than the single-layer network only if the non-linearities are introduced. The need for non-linearities is due to the logistic function ('squashing' function). In the use of backpropagation method, any non-linear function can be used if it is everywhere differentiable and monotonically increasing.

The training objective using the backpropagation method is to adjust the weights so that the application of a set of inputs produces the desired outputs. In accomplishing the task, the network is usually trained with large number of input-output pairs.

The least mean square error (LMS) algorithm is an example of supervised learning in which the learning is provided with a set of examples of desired network behaviour. The objective of LMS is to minimize the average of sum of errors. The errors are calculated as the difference between the target output and the network output.

By combining both backpropagation and least mean square methods, the hybrid method is achieved. The implementation of hybrid method results in decrement of training error throughout the learning process.

# CHAPTER 3

## METHODOLOGY / PROJECT WORK

### 3.1. Procedure Identification

There are several procedures are required for completing this project. First is overall project flow chart. Next is mathematical modelling procedure while the later involves with empirical modelling procedure.

#### 3.1.1. Overall Project Flow Chart

Figure 3.1 shows the important steps for modelling and simulation of heat exchanger project. These steps are to be completed in two-semester period. It consists both modelling and simulation exercises. Appendix 1 shows the overall project Gantt chart.

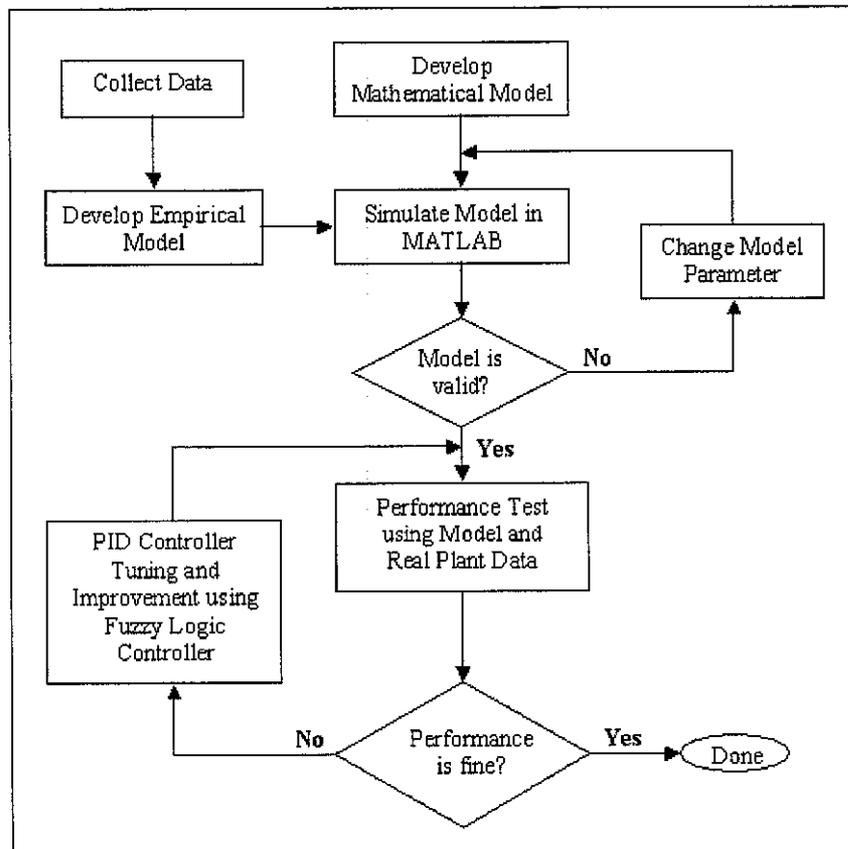


Figure 3.1: Overall Project Flow Chart

### 3.1.2. Mathematical Modelling

According to Marlin, the general mathematical procedure consists of six major steps [4]. First step is to define goals, which involves with functional relationships in this case the temperature between heating medium and heated fluid. Next is to prepare information for example by stating assumptions and data.

To formulate model, conservative energy balance principle is used to develop the partial differential equation. This solution of this equation can be developed either numerically or analytically.

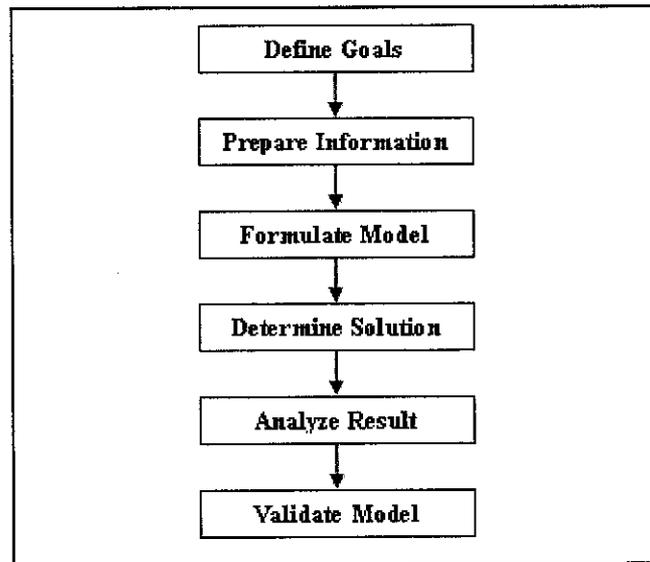


Figure 3.2: Mathematical Modelling Procedure

During result analysis, a step to be taken is check for correctness. These include limiting and approximate answers. After analyze the result, several interpretation can be made for example characteristic behavior like oscillations or extrema, and result relationship with data and assumptions. To validate model, select key values for validation, compare with experimental results, and compare with results from more complex model.

### 3.1.3. Empirical Modelling

There are six steps for developing empirical model of one system [4]. The steps are shown in Figure 3.3 below.

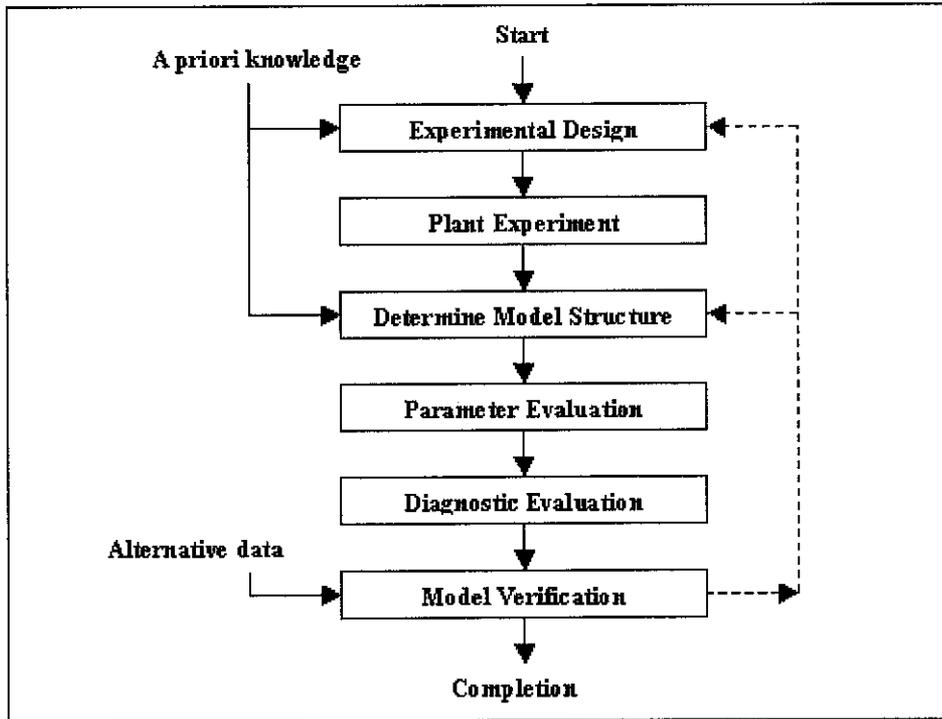


Figure 3.3: Procedure for Empirical Transfer Function Model Identification

For a start, proper experimental design is required so that its shape, duration, and base operating condition can be determined. In plant experiment, it should be executed as close to the experimental design plan as possible. To determine model structure, many methods are available but initial structure is selected based on prior knowledge.

For parameter estimation, two methods can be used which are a graphical technique or statistical principles. The diagnostic level of evaluation determines how well the model fits the data used for parameter estimation. Lastly, the final check on the model is to verify it by comparison with additional data not used in the parameter estimation.

### 3.1.4. ANFIS Modelling

Figure 3.4 shows the methodology for training a Neuro-fuzzy system. First, data must be collected or generated to be used for training and testing the Neuro-fuzzy. Once the data is collected, it must be divided into training set and test set. Training and test set are usually in the mode of input and output pair.

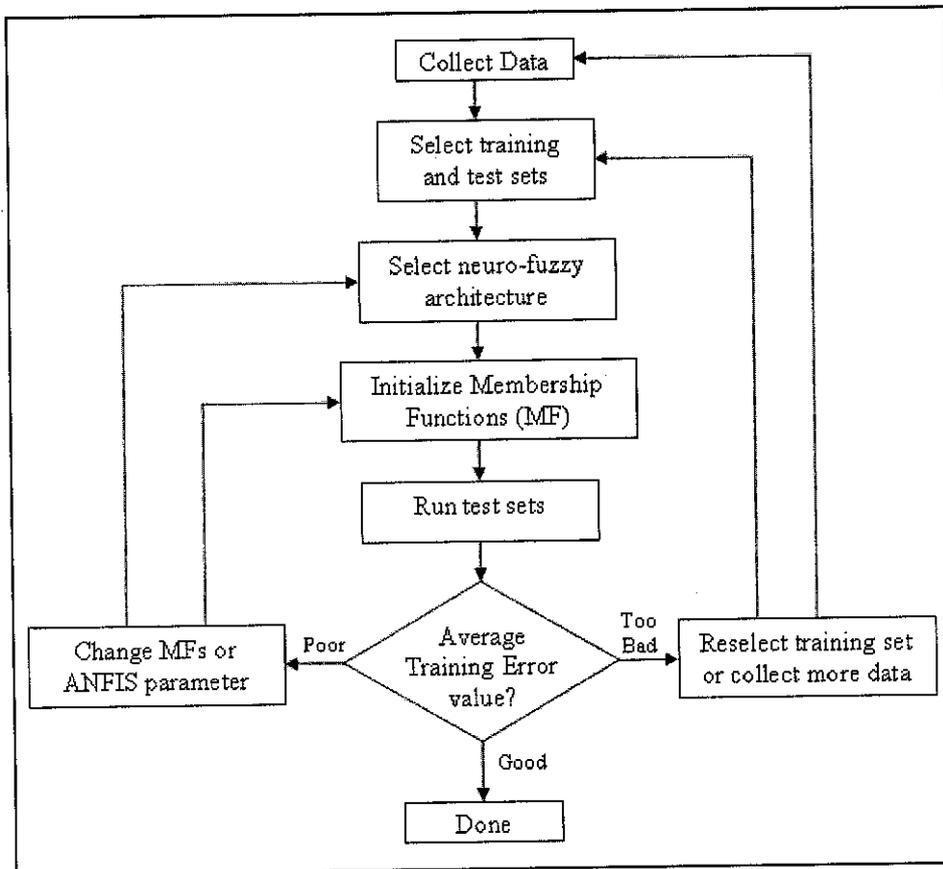


Figure 3.4: Flow chart for ANFIS modeling

The training set should cover the input space or at least should cover the space in which the ANFIS will be expected to operate. If there is not training data for certain conditions, the output of the ANFIS should not be trusted for those inputs. The division of the data into training and test sets is somewhat of an art and somewhat of trial and error procedure.

Once training sets is selected, ANFIS parameters must be selected to find the smallest desired error goal. Once it is found, it must be tested with the test data set. The test data should cover the operating region well. Testing the network involves presenting the test data set to the network and calculating the error. If the error goal is met, training is complete.

If an incomplete test set is causing the poor performance, the test patterns that have high error level should be added to the training set, new test should be chosen, and the network should be retrained. If there is not enough data left for training and testing, data may be collected again or be regenerated.

## **3.2. Tools and Software**

### **3.2.1. MATLAB-Simulink**

MATLAB offers array operations that allow one to quickly manipulate sets of data in a wide variety of ways. MATLAB also offers programming features similar to those of other computer programming languages. In addition, MATLAB offers graphical user interface (GUI) tools that allow one to use it as an application development tool <sup>[5]</sup>. Therefore, this project will utilize most of MATLAB programming application and its GUI development feature.

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 <sup>[6]</sup>.

### 3.2.2. Fuzzy Logic Toolbox (*anfisedit*)

The fuzzy logic toolbox is a collection of functions built on the MATLAB numeric computing environment. It provides tool to create and edit fuzzy inference systems into simulations with Simulink, or can even build stand-alone C programs that call on fuzzy systems you can build with MATLAB. This toolbox relies heavily on Graphical User Interface (GUI) tools to help to accomplish work, although it can work entirely from the command line.

The toolbox provides three categories of tools:

- Command line functions
- Graphical, interactive tools
- Simulink block and examples.

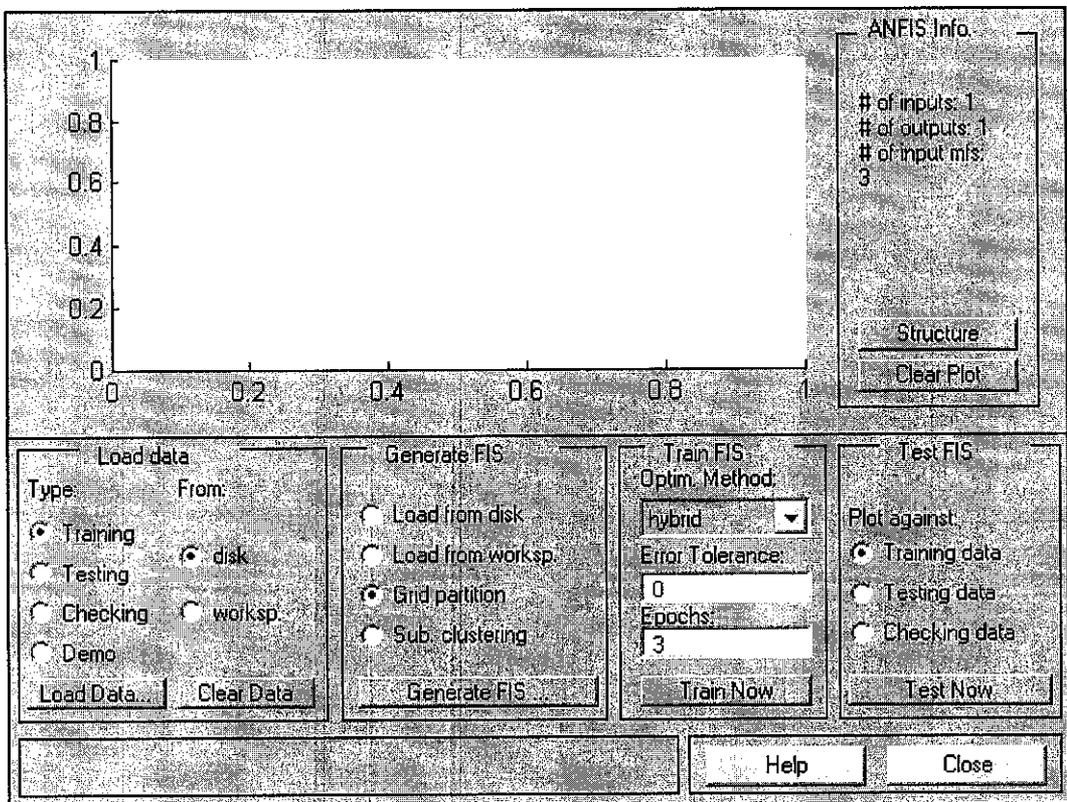


Figure 3.5: Graphical User Interface (GUI) of *anfisedit* toolbox

### Membership Function Editor (*mfedit*)

The *mfedit('a')* generates a membership function editor that allows user to modify all the membership functions for your FIS stored in the file *a.fis*. The *mfedit(a)* operates on a MATLAB workspace variable for a FIS structure *a*. The *mfedit* alone opens the membership function editor with no FIS load.

The membership function (MF) editor is used to create, remove and modify the MFs for a given fuzzy system. On the left side of the diagram is a “variable palette” region that you use to select the current variable by clicking once on one of the displayed boxes. Information about the current variable is displayed in the text region below the palette area.

### Rule Editor (*ruleedit*)

The Rule Editor, when invoked using *ruleedit('a')*, is used to modify the rules of a FIS structure stored in a file, *a.fis*. It can also be used to inspect the rules being used by a fuzzy inference system. To use this editor to create rules must first have all of the inputs and output variables you want to use defined with the FIS editor. Rules can be created using the list box and check box choices for input and output variables, connections, and weights. The syntax *ruleedit(a)* is used to when you want to operate on a workspace variable for a FIS structure called *a*.

On the Rule Editor, there is a menu bar that allows to open related GUI tools, open and save systems, and so on. The File menu for the Rule Editor is the same as the one found on the FIS editor.

### Rule Viewer

The Rule Viewer displays, in one screen, all parts of the fuzzy inference process from inputs to outputs. Each row of plots corresponds to one rule, and each column of plots corresponds to either an input variable (yellow, on the left) or an output variable (blue, on the right). System input can be changed either by typing a specific value into the input window or by moving the long yellow index lines that go down each input variable's column of plots.

### Surface Viewer

The Surface Viewer invoked using *surfview('a')* is a GUI tool that enable to examine the output surface of a FIS, a.fis, for any one or two inputs. Since it does not alter the fuzzy system or its associated FIS matrix in any way, it is a read-only editor. Using the pop-up menus, two input variables can be selected when it is assigned to the two input axes (x and y), as well the output variable that is assigned to the output (or z) axis. Select the evaluation button to perform the calculation and plot the output surface. By clicking on the plot axes and dragging the mouse, you can actually manipulate the surface so that it can be viewed from different angles.

### 3.2.3. Heat Exchanger Pilot Plant with DCS

The plant is definitely a tool needed for completing this project. The elements that are important for analysis are temperature transmitter, control valve and heat exchanger itself. DCS will perform data acquisition which is vital for modeling and analysis of this project.

# CHAPTER 4

## RESULTS AND DISCUSSION

### 4.1. Dynamic Behavior of Mathematical Model

In general, the basic block diagram of heat exchanger developed using mathematical model method and MATLAB Simulink software is as shown in Figure 4.1 below. The configuration of inputs and outputs are made by referring to actual pilot plant. It has four inputs and two outputs, which are flow rate and temperature at both inlet and outlet of shell and tube line.

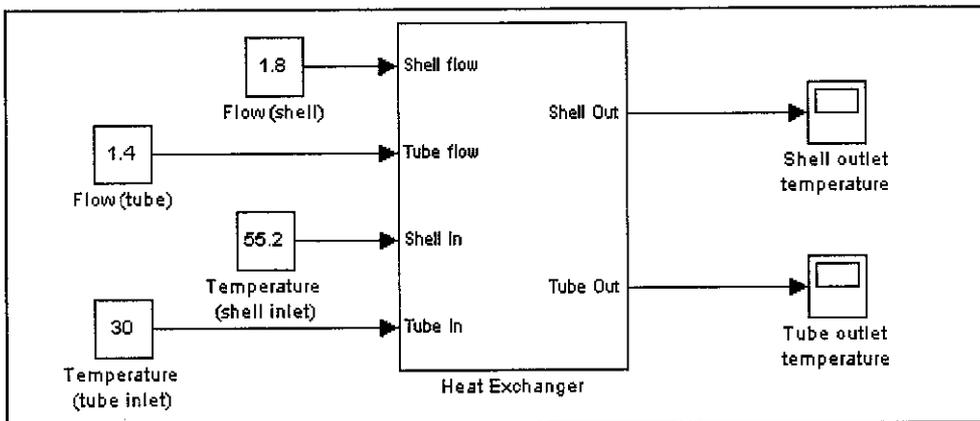


Figure 4.1: Basic Mathematical Model block for Heat Exchanger

Inside the heat exchanger model block, it contains partial differential equation that relates the input with output. Figure 4.2 shows the components masked inside the heat exchanger model block.

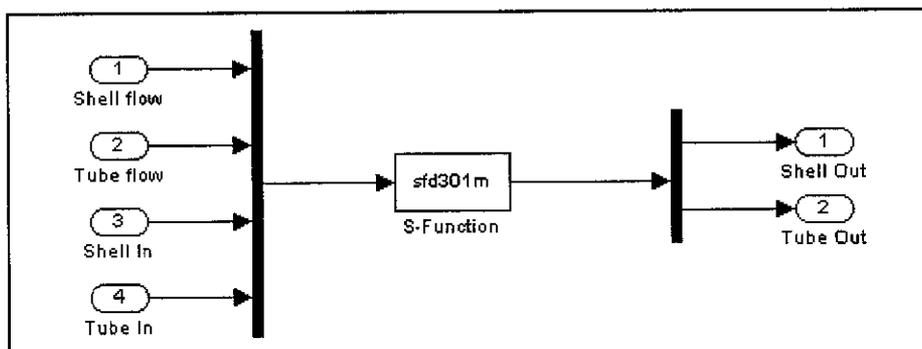


Figure 4.2: Components masked under Heat Exchanger Block

From Figure 4.2, the partial differential equation is put inside an S-function file, named as *sfd301m*. This file contains all the algorithm and calculations that involve with model input-output relationship. The syntax for *sfd301m* file is shown in Appendix 2.

Since the equations 3 and 4 shown in literature review are not enough to model the heat exchanger temperature behavior, several modifications are made to the *sfd301m* file. One example of modification made is by adding flow rate calculation to the linear differential equation. The details of this modification are shown in Appendix 3.

After completing the basic heat exchanger model using mathematical approach, its dynamic behavior is tested. In addition, the PID controller is also incorporated inside the HEX closed-loop system so that it can be used to control the heat exchanger product temperature. Figure 4.3 shows the Simulink diagram for heat exchanger with PID controller.

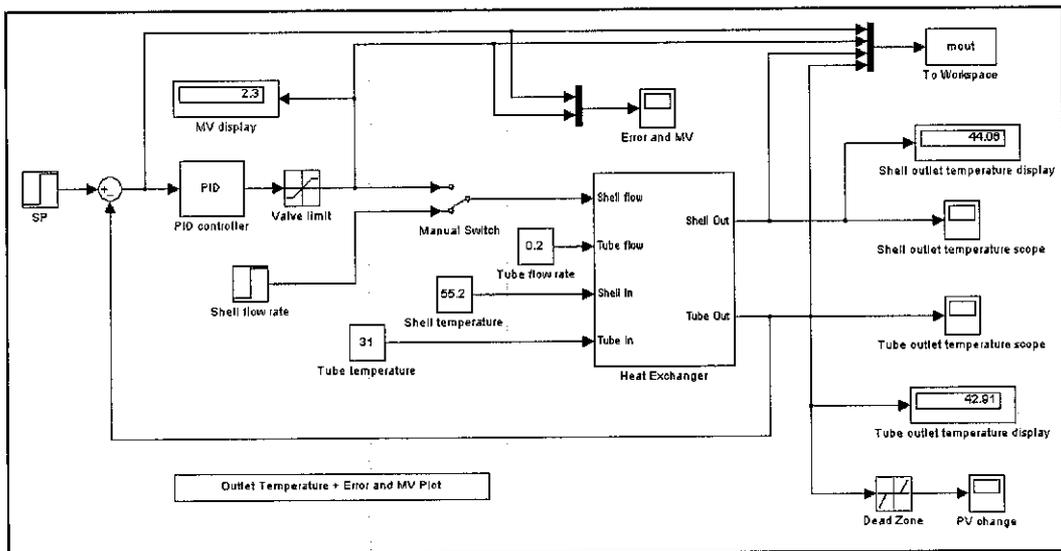


Figure 4.3: Mathematical Model of Heat Exchanger with PID Controller

First test conducted to examine the dynamic behavior of mathematical model is the open loop test. It is conducted so that the output response can be compared with the actual process reaction curve of heat exchanger at pilot plant. This test is also conducted without PID controller presence.

Figure 4.4 shows the reaction curve plotted using MATLAB. In general, the curve is quite similar with actual reaction curve obtained during empirical modelling experiment. Therefore, this mathematical model is valid to be used in simulating the actual heat exchanger plant. In addition to that, the output response to various types of input changes can also be observed with this model since it has four types of input.

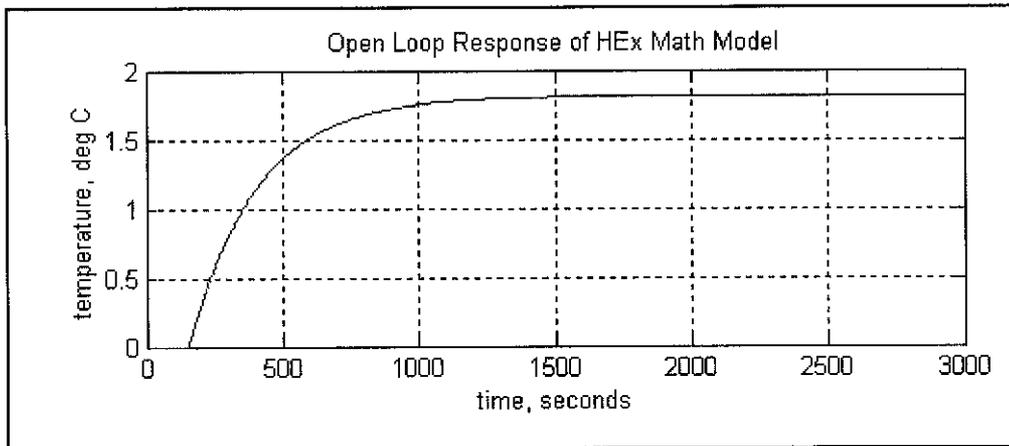


Figure 4.4: Process Reaction Curve from HEX Mathematical Model

Second test conducted for this model is the closed loop test. The objective of this simulation is to observe the PID controller response to step change in set point (SP) of process variable (PV). The SP increment in this experiment is from 41.1°C to 45°C. This model uses PI controller where the parameters are  $P = 1.763$  and  $I = 30$ . The shell and tube outlets temperature behavior is in Figure 4.5 below.

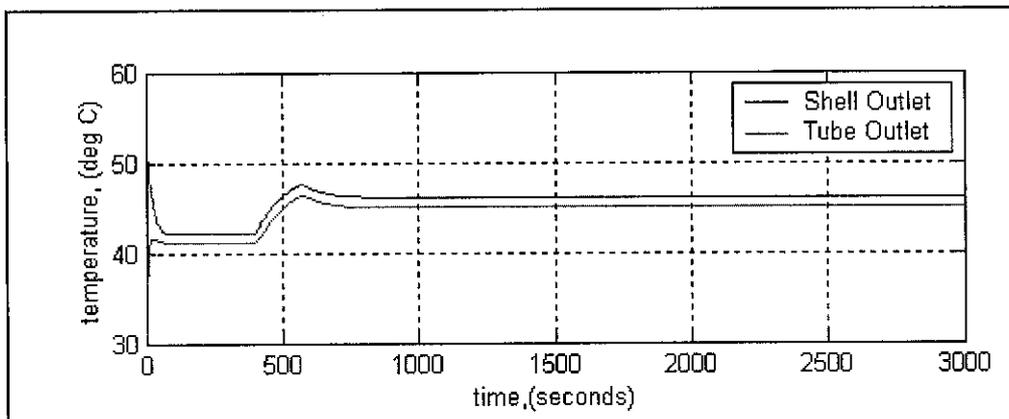


Figure 4.5: Temperature Response of Mathematical Model with PID controller

The temperature at tube outlet temperature is settling at 45°C mark just as desired in SP block. There is a small overshoot in both outlet lines as a result from PI controller action. The manipulated variable (MV), which is the flow rate at shell side, behavior is plotted in Figure 4.6 below. The MV is limited to 2.3m<sup>3</sup>/h in conjunction with actual control valve opening limit.

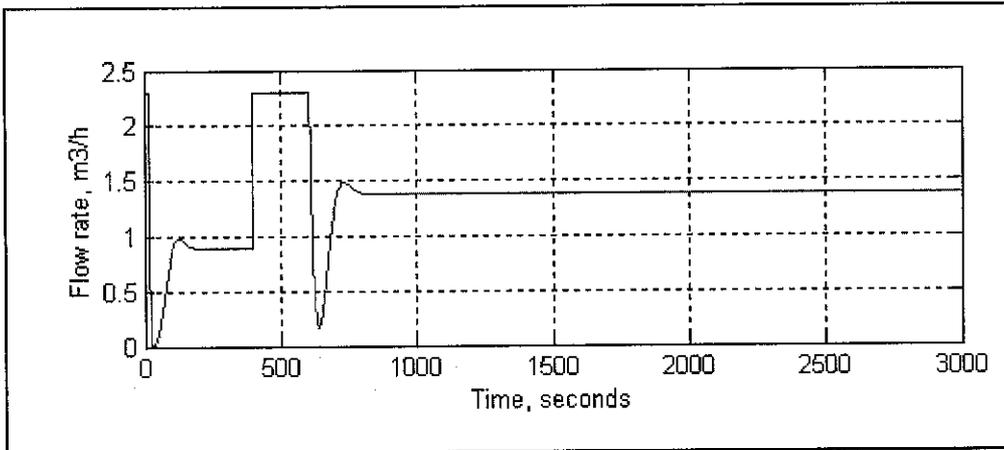


Figure 4.6: Manipulated Variable Behavior of Mathematical Model

The MV will vary depending on the PID action to error in SP and PV. PI controller will adjust the flow rate so that it will stabilize at rate where the tube outlet temperature is having temperature at desired set point. Since the MV is limited to 2.3m<sup>3</sup>/h, the response can be consider as poor because in real situation, control valve have to open up to 100% opening and suddenly close to certain opening percentage to give the desired output temperature.

#### 4.2. Empirical Model

The heat exchanger empirical model is developed based on process reaction curve where the heat exchanger transfer function is estimated to be first order plus dead time model. Figure 4.7 shows the process reaction curve obtained from experiment. The step change is 10% and the tube outlet temperature increment is about 1.7°C. Both plots are then used for heat exchanger transfer function calculation.

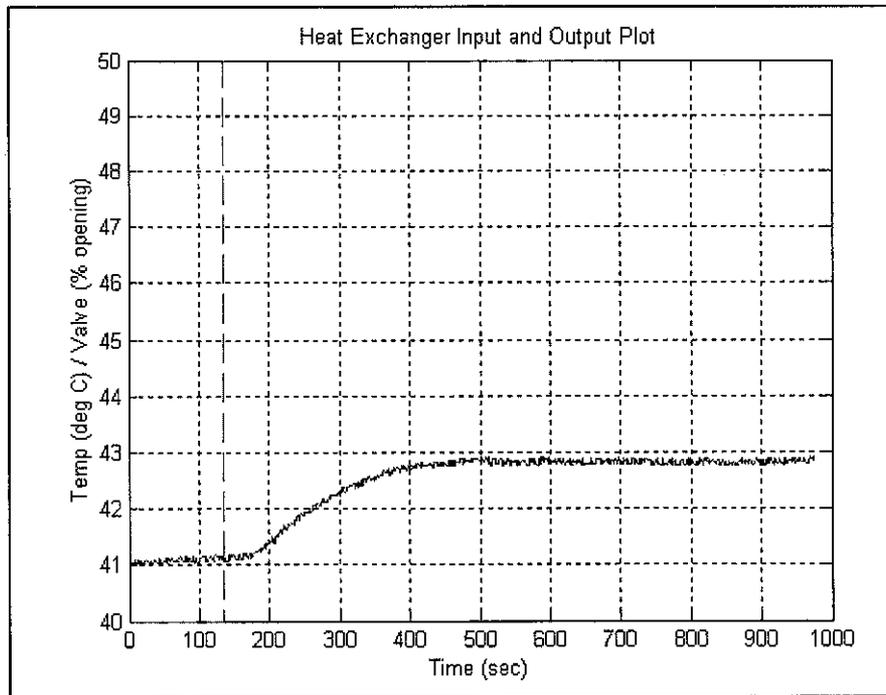


Figure 4.7: Heat Exchanger Input and Output Plot

There are two methods that can be used to calculate the transfer function parameters. Method I concerns with the value of maximum slope of the output-versus-time plot. Because of difficulty in evaluating the slope, especially when the signal has high-frequency noise, Method I typically has larger errors in the parameter estimates; thus, Method II is preferred [4]. Method II concerns with the time at which the output reaches 28% and 63% of its final value. Appendix 4 shows the detail calculation on how to get transfer function from process reaction curve. The simplified results of model parameter calculated are:

Process Gain,  $K_p = 0.17 \text{ } ^\circ\text{C}/\% \text{ opening}$

Time Constant,  $\tau = 120 \text{ seconds}$

Time Delay,  $\theta = 40 \text{ seconds}$

Hence, the general first order plus dead time model transfer function is:

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

#### 4.2.1. Open Loop Test

Figure 4.8 shows the block diagram for empirical model of HEX. The simulation input is a step input that resembles valve opening from 0% to 10%. Then, it is fed to the transfer function. Output from the transfer function is displayed by a scope. Output from the transfer function is displayed by a scope.

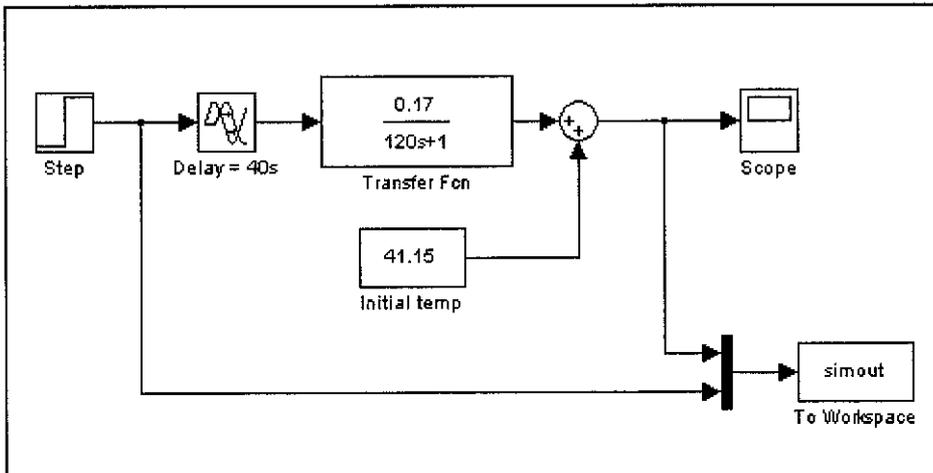


Figure 4.8: Simulink block diagram of HEX Model

The output from the model is as shown in the following Figure 4.9. From that curve, we can see that the temperature increment is about 1.7°C. The change is quite similar to reaction curve temperature change as shown in Figure 4.7. Therefore, this model is valid to resemble the actual HEX at pilot plant.

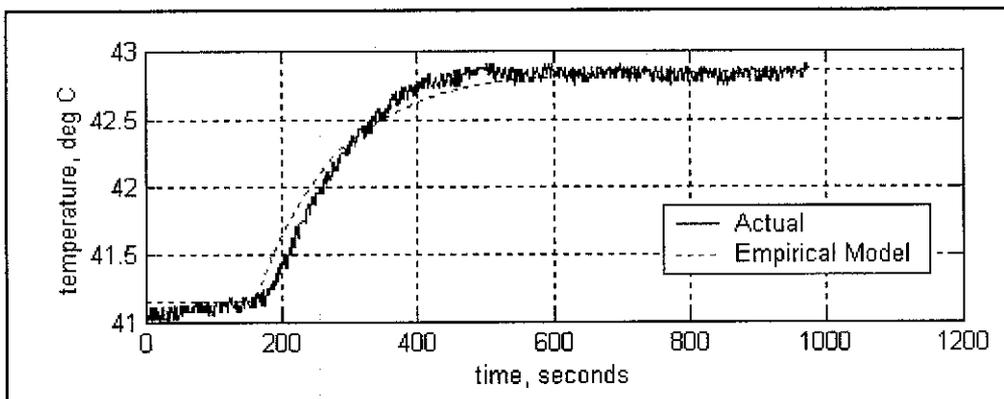


Figure 4.9: Open Loop Response of Empirical Model

#### 4.2.2. Closed Loop Test

Appendix 5 shows the empirical model with PID controller. By using parameters obtained from process reaction curve, the tuning coefficient for PID controller can be calculated by referring to Table 4.1 below. The formula shown is based on Ziegler-Nichols method.

Table 4.1: Ziegler-Nichols open-loop tuning based on reaction curve

Controller	$K_c$	$T_i$	$T_d$
PI	$(1/K_p)/(\tau/\theta)$	$3.3 \theta$	—
PID	$(0.9/K_p)/(\tau/\theta)$	$2.2 \theta$	$0.5 \theta$

In addition, there is another one tuning parameter suggested by Fertik(1974) which emphasizes on minimum ITAE (Integral Absolute Error) with limit on overshoot<sup>[4]</sup>. The formula to calculate the PI tuning coefficient is:

$$K_c = 0.859 (\theta / \tau)^{-0.977}$$

$$T_i = (\tau / 0.674) (\theta / \tau)^{0.680}$$

By referring to Marlin<sup>[4]</sup>, there are another two tuning parameter calculations suggested by Giancone and Lopez. The detail calculation for this method is shown in Appendix 6. Finally, all the tuning parameters are listed as in Table 4.2 below.

Table 4.2: PI tuning parameters for Empirical Model

PI Controller	Ziegler-Nichols	Fertik	Giancone	Lopez
$K_c$	1.763	2.571	8.82	14.12
$T_i$	132	80.291	144	108.8

Figure 4.10 shows the PV and MV response of all controllers after a step change in SP (from 45°C to 50°C). The plots are extracted from Simulink figure file. In general, all of them settle at the new set point mark but with various transient response behaviors. Lopez parameter is far more aggressive than Ziegler-Nichols since it has bigger controller gain,  $K_c$ , value. However, its MV behavior is too aggressive and not suitable for normal control valve.

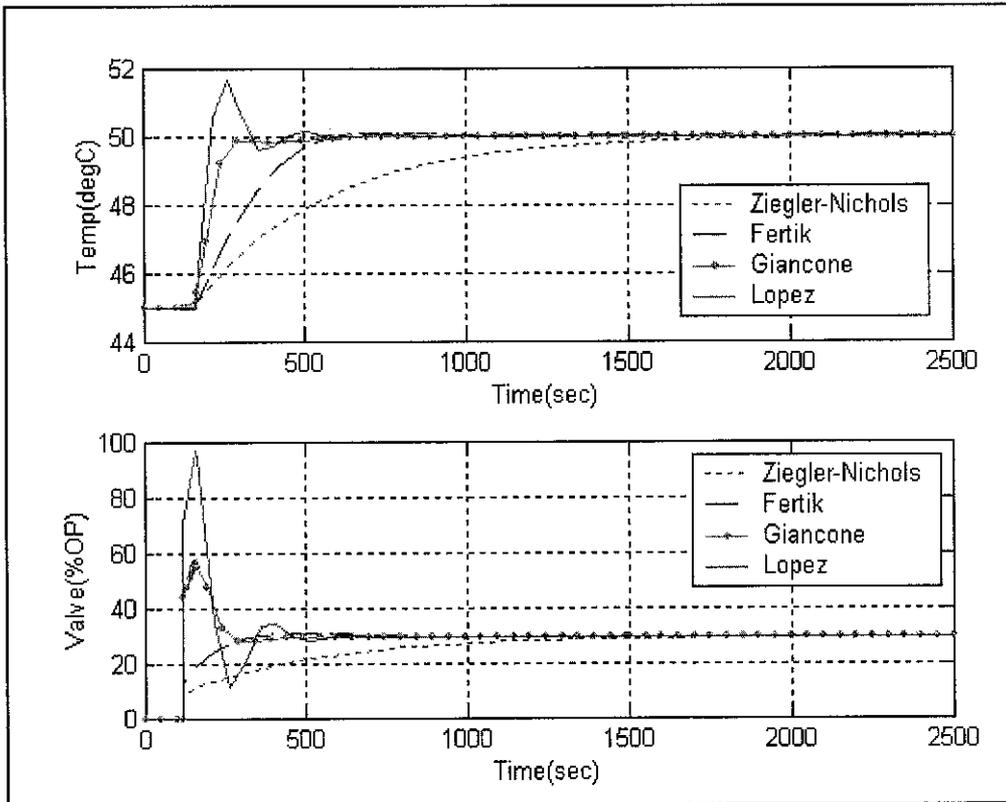


Figure 4.10: PV and MV Response from Different Tuning Parameters

Table 4.3 shows the process variable (PV) numerical analysis done using LTI Viewer. The actual plot from LTI Viewer is attached in Appendix 7. The values collected are based on a unit step in input or in other words from 0°C to 1°C. The first three parameters; Ziegler-Nichols, Fertik and Giancone did exhibit sluggish PV performance. The rise and settling time are quite high compared to Lopez parameter. In addition, the overshoot percentage is also small with low peak amplitude value. This analysis proves that the PV response is not suitable and the parameters are required to be retuned.

Table 4.3: Characteristics from LTI viewer (PV)

Characteristics	Ziegler-Nichols	Fertik	Giancone	Lopez
Rise Time (sec)	986	367	217	102
Peak Amplitude	0.999	1	1	1.01
Overshoot (%)	0	0.306	0	0.533
Settling Time (sec)	1760	564	462	164
Steady State Gain	1	1	1	1

Table 4.4: Characteristics from LTI viewer (MV)

Characteristics	Ziegler-Nichols	Fertik	Giancone	Lopez
Rise Time (sec)	1000	249	116	117
Peak Amplitude	5.88	5.93	8.82	14.1
Overshoot (%)	0	0.85	49.9	140
Settling Time (sec)	1790	366	177	206
Steady State Gain	5.88	5.88	5.88	5.88

Table 4.4 shows the MV analysis using LTI Viewer. The high value of  $K_C$  in Lopez parameter has resulting high value of MV overshoot. However, this behavior is not in favor since it can lead to oscillation if simulated for longer period. In general, none of these tuning parameters have exactly good PI combination.

To have a good combination, the parameters are to be retuned. The  $K_C$  value used is 1.763, obtained from Ziegler-Nichols parameters. This value is suitable since it is small and can prevent oscillation. With constant  $K_C$  value, the variation of tuning parameters is made at  $T_i$  values. The basic rule-of-thumb when varying  $T_i$  values is a decrease to  $T_i$  value will increase overshoot percentage, while decreasing the rise and settling time.

The simulation result for this retuning exercise is tabulated in Table 4.5 and Table 4.6. Both are data collected from LTI Viewer analysis for PV and MV respectively. The actual plots from LTI Viewer are attached in Appendix 8. Based on the result, it is clearly proved that smaller  $T_i$  value will yield more aggressive and faster output response.

Table 4.5: Retuning PI parameters with variable  $T_i$  value (PV)

$T_i$ value	Rise time (sec)	Peak Amplitude	Overshoot (%)	Settling time (sec)
132	986	0.999	0	1760
100	703	1	0	1230
80	532	1	0.000523	879
60	578	1.01	0.881	566
40	248	5.55	1.06	718
20	139	18	1.18	714

Table 4.6: Retuning PI parameters with variable  $T_i$  value (MV)

$T_i$ value	Rise time (sec)	Peak Amplitude	Overshoot (%)	Settling time (sec)
132	1000	5.88	0	1790
100	667	5.88	0	1180
80	449	5.88	0.00115	759
60	253	5.99	1.87	702
40	123	6.6	12.1	624
20	46.9	8.52	44.9	661

Since retuning exercise is quite subjective, the final tuning parameter that is suitable for both PV and MV response is decided to be:

$$K_C=1.763 \text{ and } T_i = 30$$

The  $T_i$  value is selected to be 30 so that it will yield moderate and nice output response that is between  $T_i$  value at 40 (not so aggressive) and 20 (too aggressive). Figure 4.11 shows the PV response while for MV response is shown in Figure 4.12. The overshoot percentage of PV is 10.3% while for MV is 23.6%. In theory, the best overshoot allowable for MV is about 25%.

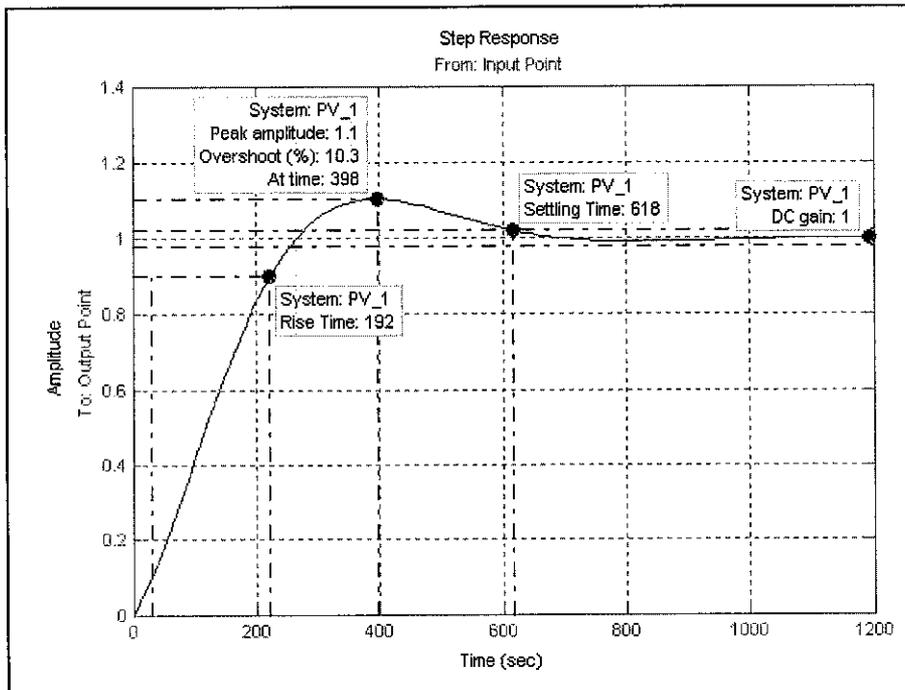


Figure 4.11: PV Response for  $K_C = 1.763$  and  $T_i = 30$

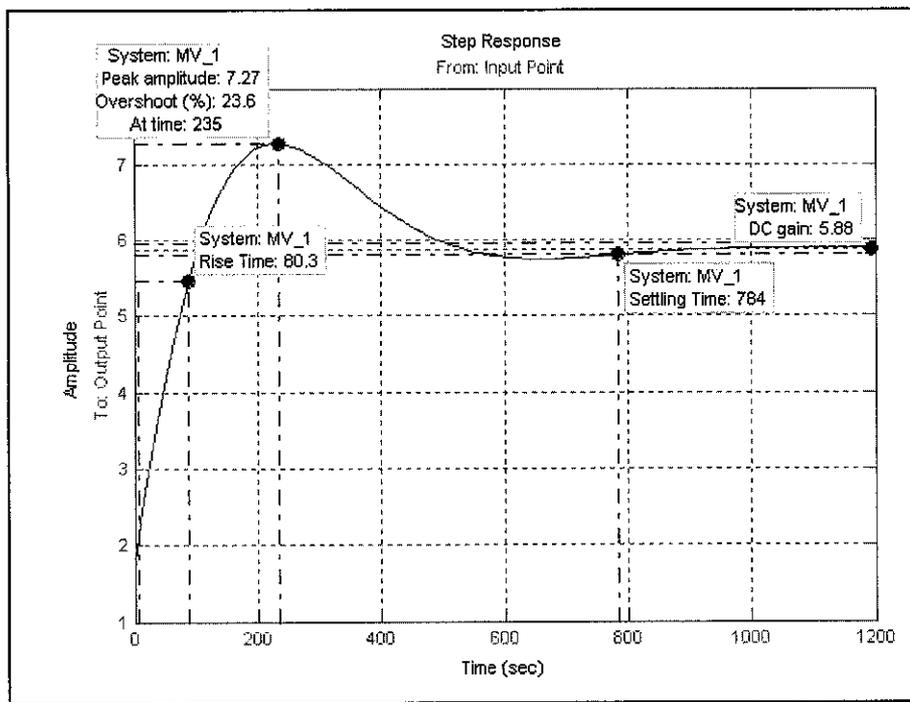


Figure 4.12: MV Response for  $K_C = 1.763$  and  $T_i = 30$

In summary, the PID tuning exercises conducted are quite subjective and lots of references are available for tuning parameters calculation. However, final decision is depending on process requirement and objectives defined by user. For this project, the best tuning parameters tuned for PI controller are 1.763 and 30 for  $K_C$  and  $T_i$  respectively. In the next part, the output response from this closed-loop control system is being used for adaptive learning process for Fuzzy Logic Controller comparison study.

### 4.3. ANFIS Training

To begin the ANFIS training, several initialization steps are required to yield a good model predictor that has minimum average training error. Therefore, several adjustments are made at the following parameters to optimize the error measure. The parameters are:

- Mapping of training data as input and output pair
- Number of membership function (MF)
- Type of membership function
- Number of epochs (iteration)

For this project, the inputs for training data are the error and integral of error while the output is the manipulated variable. The reason in choosing the input types is depending on user requirement. For this project, the objective of using FLC is to imitate the PI controller response from previous simulation. The FLC is also aims to improve the response while imitating its basic behavior (overshoot and transient response).

In general, the number of MF that will give the least average training error actually can be found by trial and error. For this project, training data collected from mathematical model and empirical model are trained for several number of membership functions. Partitioning method used during this training is grid partitioning. Table 4.3 shows the average error of corresponding number of membership function after 40 epochs.

Table 4.7: Average Error of Different Number of MF

Number of Membership Function	Average Error	
	Mathematical	Empirical
2	0.033778	0.0003128
3	0.019526	$5.2152 \times 10^{-6}$
4	0.0055864	$5.3521 \times 10^{-8}$

From Table 4.3 above, an increment in number of MF will decrease the average error. However, since the training uses grid partitioning method, it will contribute to the “curse of dimensionality” which refers to such situation where the number of fuzzy rules increases exponentially with the number of input variables<sup>[8]</sup>. If higher number of MFs is used, the training process will be slow as ANFIS has to compromise with large number of fuzzy rules.

An alternative way to counter this problem is by using subtractive clustering method. Subtractive clustering is a fast, one-pass algorithm for estimating the number of clusters and the clusters centers in a set of data<sup>[9]</sup>. The difference between grid partition and sub-clustering method can be seen clearly at their respective ANFIS network structure. Figure 4.13 shows the network structure from both grid partition and sub-clustering method.

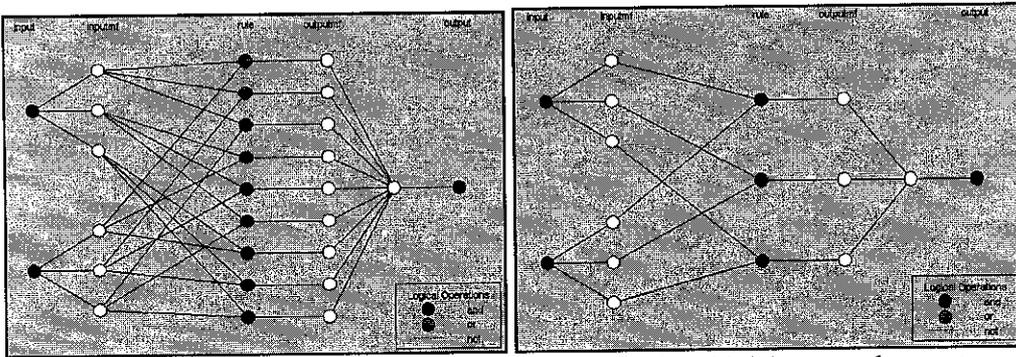


Figure 4.13: Grid partition (left) and Sub-Clustering (right) network structure

For this project, since the average error is still in tolerable range, 3 MF is enough for training the model data using ANFIS. So, grid partitioning method will be used to generate the desired FIS. In addition, the average error value also varies according to number of epochs set as stopping criterion. In this case, the training is stopped after 40 epochs.

In ANFIS training, the selection of optimization method is also important. The method used normally is selected based on least average error yield during training. For both model's training data, the average training error at each optimization method is shown in Table 4.4. During this training, grid partitioning method is used with 3 membership functions.

Table 4.8: Average Error of Different Type of Optimization Method

Optimization Method	Average Error	
	Mathematical	Empirical
Backpropagation	0.29644	20.9689
Hybrid	0.019526	$5.2152 \times 10^{-6}$

From Table 4.4, it is found that the hybrid method (mixed least squares and backpropagation) gives the least average error. This is due to the least mean square error algorithm which learns to minimize the average sum of errors. Finally, the FIS generated are tested against training data and the plots are shown in Figure 4.14.

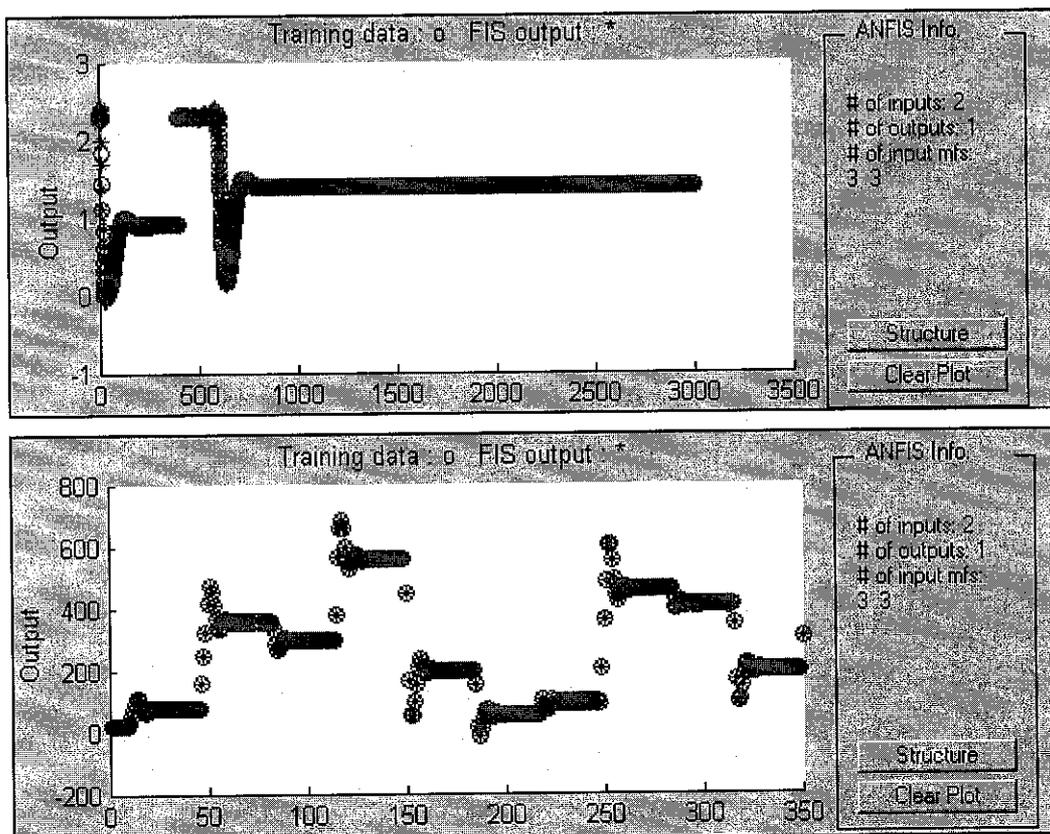


Figure 4.14: ANFIS Plot of Math Model (above) & Empirical Model (below)

In addition, Figure 4.15 shows the Surface View for both mathematical model and empirical model. This plotted are generated using Surface Viewer command. The rules for both FIS are shown in Figure 4.16. Input 1 is for error while Input 2 is for integral of error. Output is the manipulated variable.

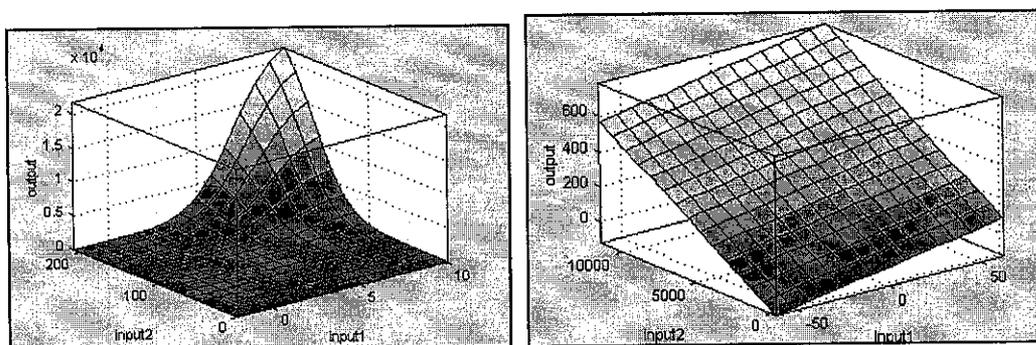


Figure 4.15: Surface View of Math Model (left) & Empirical Model (right)

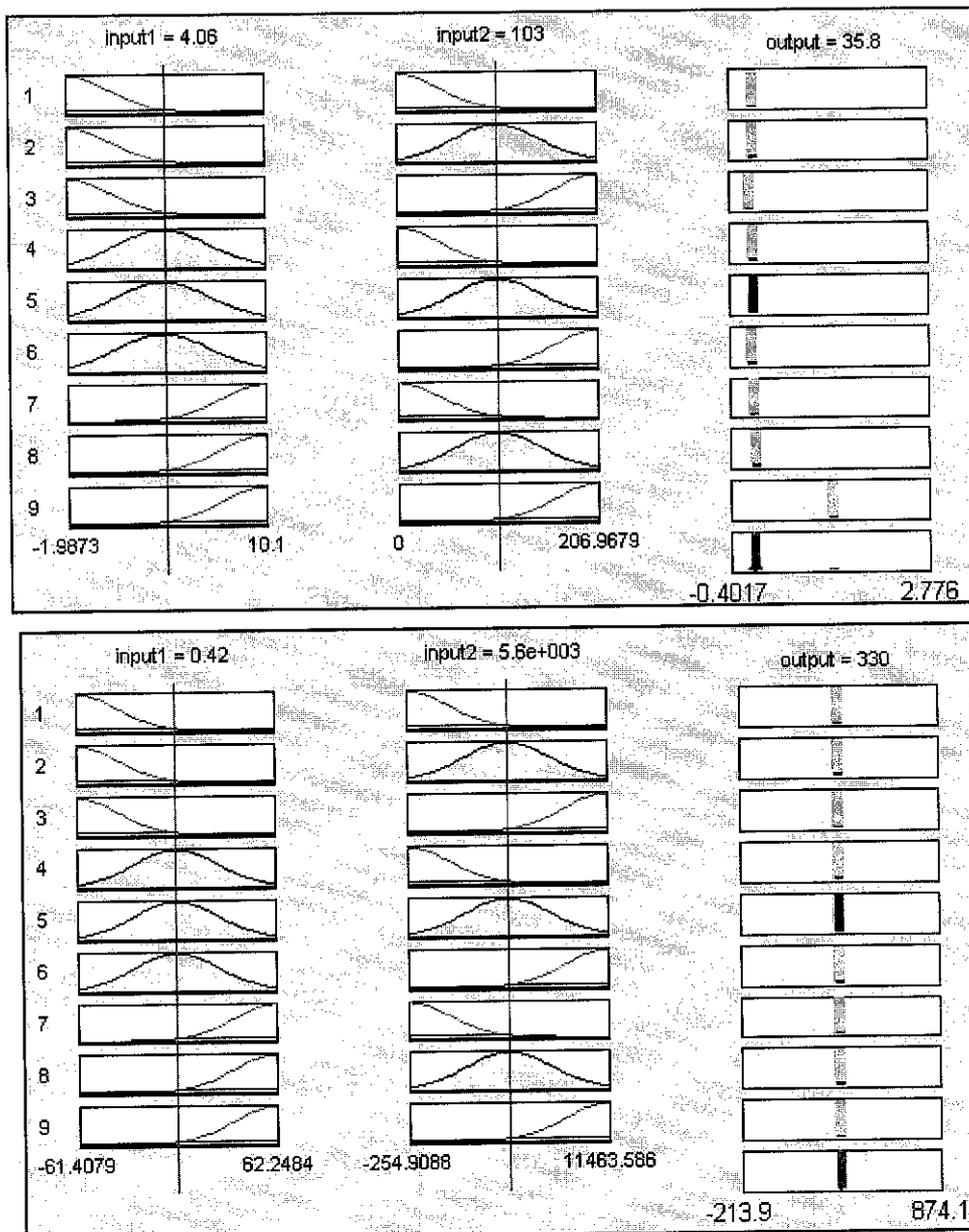


Figure 4.16: Rules View of Math Model (above) & Empirical Model (below)

#### 4.4. Comparison: PID versus Fuzzy Logic Controller

Comparison between PID and FLC is done for both mathematical and empirical model. The main difference between PID and FLC is the algorithm it uses to control the process. The PID uses typical algorithm where error is the input while MV is the output. On the other hand, FLC can use as many input as it required as long as the input has been trained and has membership function with the fuzzy rules.

For this project, inputs for FLC are; the error between SP and PV itself like the one who goes into the P controller input, plus the integral of error which resembles the input for I controller. All the membership functions and fuzzy rules are extracted from ANFIS training conducted in previous section. So, when simulation is running, the FLC will call the FIS in the workspace to predict the output.

First comparison study is about fuzzy logic controller performance in the mathematical model. Simulink block diagram for this experiment is attached in Appendix 9. The input is having a step change at 250s mark. Figure 4.17 shows comparison plot between PID controller and FLC.

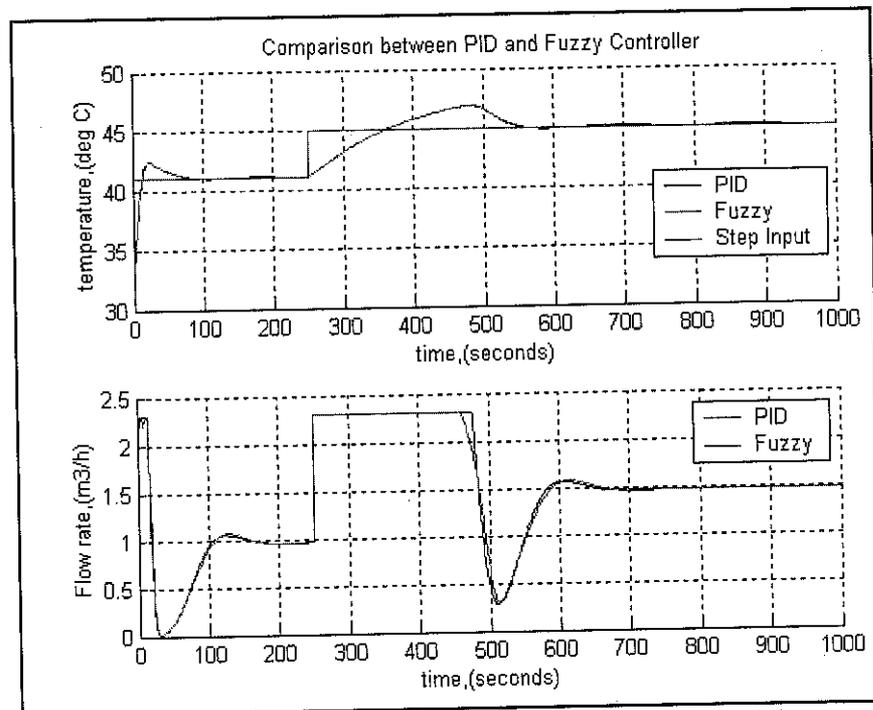


Figure 4.17: Comparison Test for Mathematical Model

As referred to Figure 4.15, it seems that both controllers did adjust the PV according to the new SP value. The output response to a set point change (41.1°C to 45°C) is quite the same for both controllers. Therefore, the imitation process is successful but the improvement is still not clear to be concluded.

Next, the comparison between PI controller and FLC is simulated. The FIS used is generated from previous ANFIS training section. The input step variation used is a uniform random number type which is available in Simulink library. The input step change is ranging from 0°C to 100°C. The Simulink block diagram is attached in Appendix 10.

The simulation result is attached in Figure 4.18 below. The plot shows that the FLC dynamic behavior is almost similar to the PI controller. There is a slight difference between these two output response where FLC has smaller rise and settling time as well as its overshoot percentage.

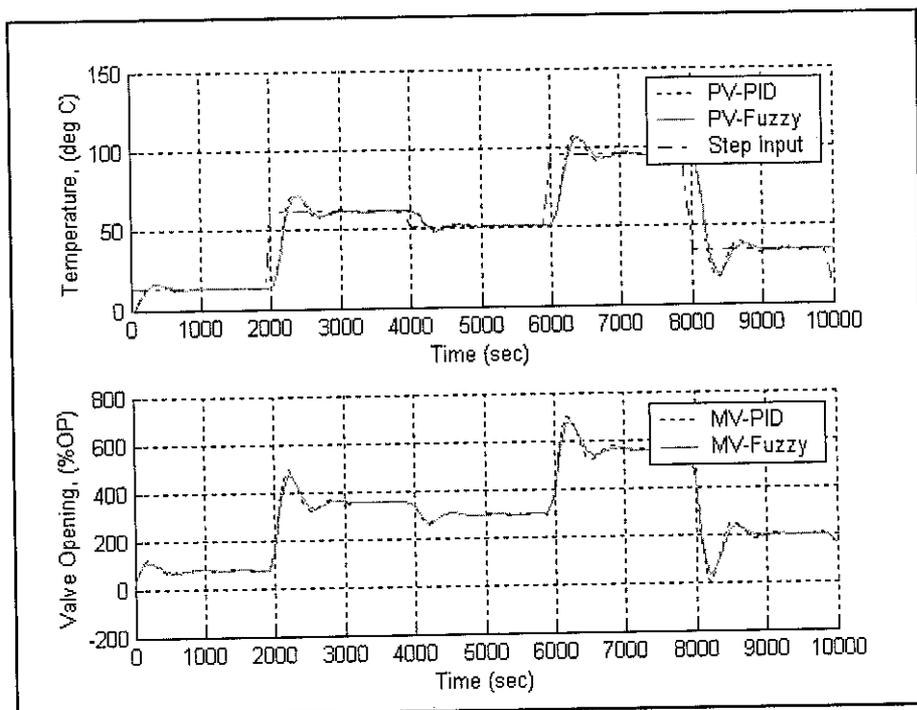


Figure 4.18: Comparison Test for Empirical Model

The difference can be clearly seen in Table 4.9. These data are collected from linear analysis using LTI Viewer. Figure 4.19 shows the FLC step response plot. The similarity of dynamic behavior proves that the FLC is successfully imitating the PI controller response. The slight differences in several characteristics are due to ANFIS training session and adaptive learning process during FIS generation.

Table 4.9: Comparison Data for FLC and PI controller

Characteristics	PI Controller	Fuzzy Logic Controller
Rise Time (sec)	192	189
Peak Amplitude	1.1	1.09
Overshoot (%)	10.3	9.4
Settling Time (sec)	618	603

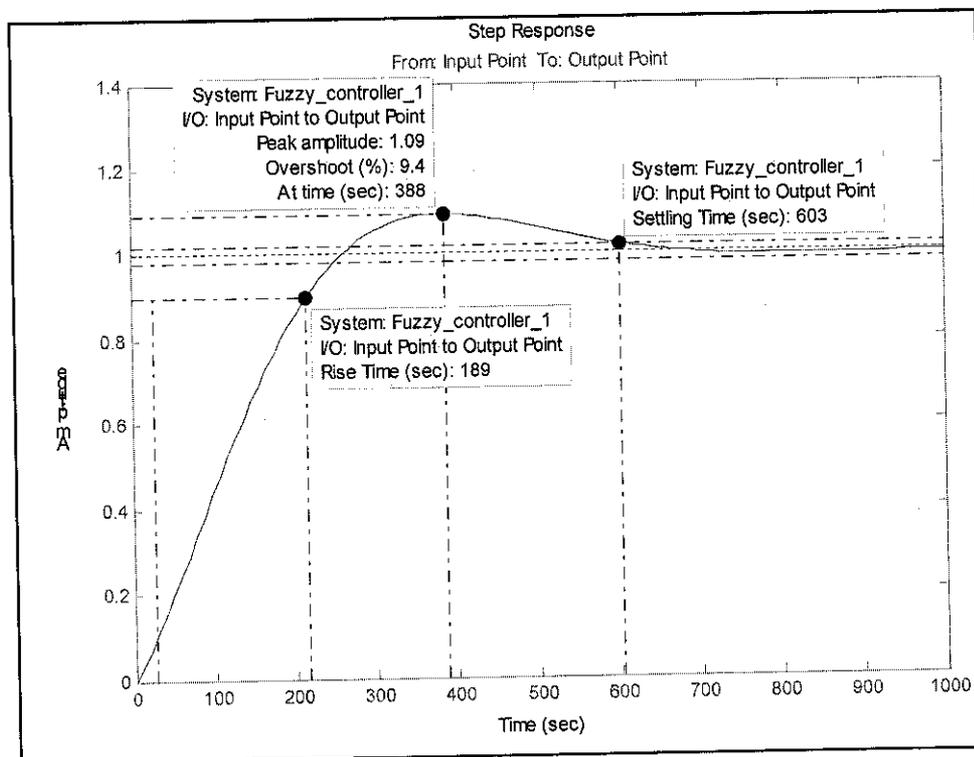


Figure 4.19: Fuzzy Logic Controller step response

In brief, the FLC is more flexible than PI controller since it can be trained again and again in order to yield a better result. Its number of inputs can also be added easily if there is a requirement to do so.

## 4.5. Discussion

In general, this project is divided into four main categories, which are:

- Mathematical modelling
- Empirical modelling
- ANFIS training
- Comparison Study between FLC and PID Controller

For the first part, the mathematical model developed is very flexible since it can be modified easily. The modification can be made at the *sfd301m* file which contains all the mathematical equations and initial plant parameters. The performance of this model is quite good since it has many inputs and outputs to be monitored.

Furthermore, the input/output configuration is also resembles the actual heat exchanger pilot plant. By varying any one of the inputs, the output response will be different depending on the changes applied. Since this project concerns only on single loop controller performance, all other inputs are held constant. So, not much analysis on the input changes and output response are conducted.

In most cases, every model has its own deficiencies. The problem of this model is its basic modelling data is not available since there is no specification sheet available at laboratory. This specification sheet is important since it contains physical data, tube and valve rating, and basic operational data, etc. All of these things are important for model initialization. Up to this stage, all the data put in model is only approximation based on physical observation.

One more problem for this mathematical model is there is no valve opening relationship with flow rate. For the open loop test, a step change in input is applied in terms of flow rate, not the valve opening. Therefore, PID controller will have a flow rate unit as MV not the valve opening when responding to error in closed loop performance test. Although this is not really big problem, it is better to have MV in opening percentage unit rather than flow rate.

For empirical model, the modelling exercise is quite simple and first order process approximation is not really convincing. The heat exchanging process might be a second order process but after all, the model did show a good estimate based on comparison of its output plot with actual reaction curve. There is another option to develop the empirical model which is using the system identification method.

Theoretically, system identification method can predict the process transfer function based on input and output data fed into its system. It also can give higher order transfer function that resemble closely with actual plant transfer function. However, this method does not give the desired result throughout the whole project duration. Therefore, this option is not in favor and the model just use the transfer function obtained from reaction curve.

In addition, the PID inside the MATLAB Simulink library is somehow different with PID controller algorithm in Marlin, Process Control [4]. The algorithm is as shown below:

$$G_c(s) = MV(s) / E(s) = K_C (1 + 1 / T_I s + T_D s)$$

Therefore, the major differences are:

- The P controller will supply gain to I and D controller.
- The Integral controller gain is in inverse configuration (1/I) compared to (I/s) as in default Simulink library PID controller.

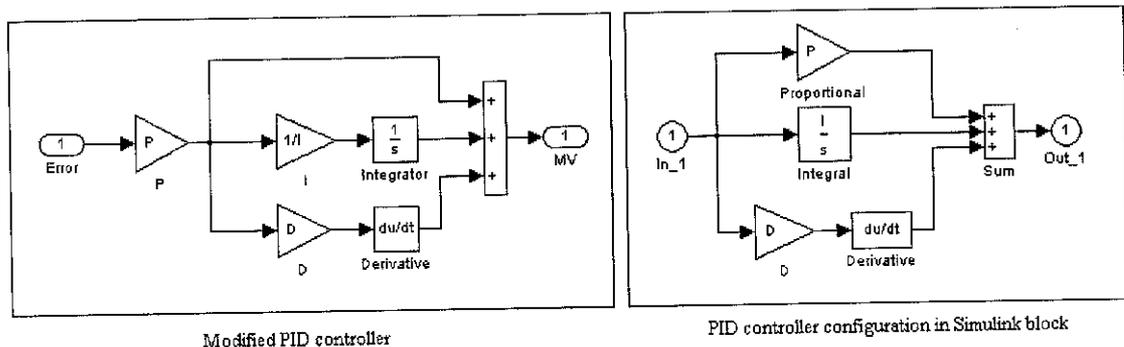


Figure 4.20: PID controller configuration

In summary, the models developed by both approaches (mathematical and empirical method) are quite similar to the actual process behavior and the model error is still in tolerable region. Therefore, several criteria which are important to be considered when choosing the modelling approach are:

- Complexity
- Degree of understanding of internal plant mechanisms
- Ability, quantity and quality of measurements
- Availability of expert knowledge about the plant
- Level of uncertainty

The PID controller performance test for both models is quite good but comparison with actual PID controller performance is not being done since the plant has lots of technical problems.

For ANFIS training, there are several factors to consider before training sets of data using ANFIS Editor. The factors include type of partitioning method, number of membership functions, and type of membership function. The success in choosing appropriate factor can lead to higher model accuracy.

In ANFIS, there are two type of partitioning method which are grid partitioning and subtractive-clustering. Grid partition has some disadvantages since it deals with number of membership functions. Increment in number of MF will increase number of fuzzy rules. Training will be inevitable slow and not practical for multi-input system.

Another partitioning method is sub-clustering method. The purpose of clustering is to identify natural groupings of data from large data sets to produce a concise representation of a system's behavior. Cluster information can be used to generate a Sugeno-type fuzzy inference system that best models the data behavior using minimum number of rules. The rules partition themselves according to the fuzzy qualities associated with each of data clusters.

Beside ANFIS, there is another method to generate a fuzzy inference system (FIS), which is the Mamdani Method. By using this method, the number of inputs and outputs are very flexible and it can be adjusted freely according to model requirement. The membership function is also depends on user selection where the ranges and values are set manually. For empirical model, there is a FIS generated by using Mamdani Mehtod but the response is very poor due to inappropriate membership function and rules adjustment. The response can be seen in Figure 4.21.

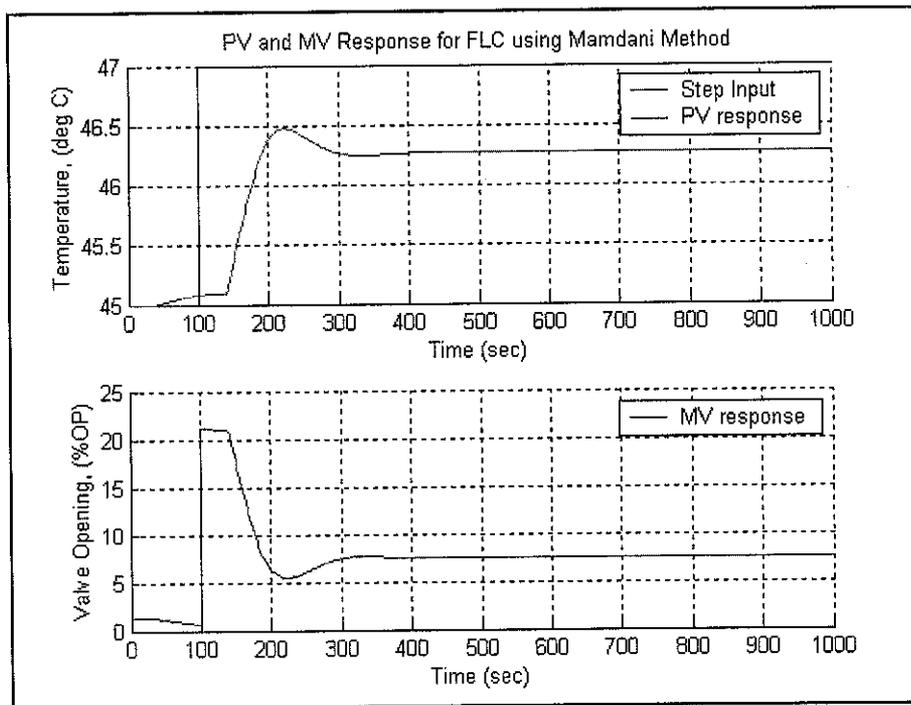


Figure 4.21: Performance test on FLC using FIS from Mamdani method

The decision whether to use Mamdani or Sugeno-type (ANFIS) in modelling a system is depending on user requirement. Some advantages of Sugeno-type fuzzy inference system are:

- It is computationally efficient
- It works well with linear techniques (i.e. PID control)
- It works well with optimization and adaptive techniques
- It has guaranteed continuity of the output surface.
- It well-suited to mathematical analysis

The advantages of Mamdani method are:

- It is intuitive
- It has widespread acceptance
- It is well-suited to human input.

Finally, the comparison between FLC and PID controller is somehow subjective and the differences between those two controller performances are not much obvious. FLC is definitely better in several aspects but then it needs training and learning process before it can control the situation.

## 5.2. Next Project Recommendations

The project development currently involves with Single Input and Single Output (SISO) system. It only concerns with one input; shell inlet flow rate and one output; tube outlet temperature. After completing this project, hopefully it can be extended to Multiple Inputs and Multiple Outputs (MIMO) system where the complete model of heat exchanger can be used to analyze various performances from various disturbances.

The reason why this project is recommended to involve with MIMO system is because with multiple inputs and outputs, the vast potential of FLC can be fully utilized. As mention before, the advantage of FLC is the ability to imitate (learn) another process behavior and it can be configured to handle more inputs than PID controller. Therefore, it is possible that a single FLC can replace several PID controllers in controlling a MIMO system.

In addition, if this project is going large at MIMO scale, more literature reviews are needed since it must consider lots of parameters and process characteristics involved with heat exchanger. There are lots of journals, thesis and term paper available but it must be selected wisely since not all information are useful for the project development.

For PID controller analysis, it is recommended that all the tuning parameters calculated are to be tested at actual plant. By doing this, the simulation result can be compared with actual one and strong validations can be made. The problem faced for this project during conducting this proposal is the laboratory plant is always being used by students for their Process Control course. If given ample time, perhaps this important step can be carried out.

The GUI for this project can not be completed since project duration given is not enough. However, the Simulink block diagram developed is quite useful for analysis the heat exchanger temperature behaviour. Perhaps next time, the diagram can be translated into a nice interface where clearer analysis presentation can be viewed by user.

### 5.3. Conclusion

Modelling and simulation of heat exchanger is one of useful learning tools to understand process control technique in petrochemical industries. To start a modelling process, a good understanding of process behavior is required as it will determine the important parameters and characteristics to be analyzed. The input and output correlation are also important to be studied since it will affect the model performance and accuracy.

There are many modelling approaches that can be used to model the heat exchanger and the selection is depending user requirement. Most popular and simplest method is empirical modelling method. Mathematical approach in the other hand does give a good prediction but the calculation part is somehow complicated. ANFIS training is also a good option to model the HEX based on its real time input and output data.

The fuzzy logic controller and PID controller are both good in controlling a chemical process and which method is in favor actually depending on the owner of the system. In conclusion, the project is indeed successful in giving a better understanding on heat exchanger control system.

## CHAPTER 5

### CONCLUSION AND RECOMMENDATION

#### 5.1. Relevancies to Objectives

The first objective, which is to model and simulate the heat exchanger, is finally complete. Two modelling approaches are used; mathematical and empirical method; where both give quite similar response after being compared with actual one. Therefore, it is valid to use this model as an output predictor to current heat exchanger plant.

The second objective which involves with PID controller tuning and performance test is also accomplished. Several analyses on PID performance are being conducted by using Simulink model. From the tuning exercise, the behavior of both process variable and manipulated variable are being studied and the result obtained is quite useful to be implemented at actual pilot plant.

The third objective is completed at the comparison study stage. From the study, it is found that FLC is slightly better than PID in several characteristics. One clear advantage of FLC over PID controller is the FLC can be configured to process more number of input signals compared to PID. However, the implementation of FLC at pilot plant is not feasible as the plant is still new and extensively used by other Process Control students.

## REFERENCES

1. R.K. Sinnott; "Heat Transfer Equipment", *Chemical Engineering*, 2000, Volume 6, Butterworth-Heinemann.
2. L. Xia, J. A. D. Abreu Garcia, and Tom T. Hartley, "Modelling and Simulation of a heat exchanger," *IEEE International Conference on System Engineering*, (Ohio), pp. 453-456, 1991.
3. M.H.R Fazlur Rahman and R. Devanathan Sr. "Modelling and Dynamic Feedback Linearization of a Heat Exchanger Model," *IEEE Journal*, 1994.
4. Marlin, Thomas E., "Mathematical Modelling Principles and Empirical Model Identification," *Process Control*, 2000, 2<sup>nd</sup> Edition, Singapore, McGraw Hill.
5. D. Hanselman, B. Littlefield, "Introduction," *Mastering MATLAB 6*, 2001, Prentice Hall Inc.
6. J. B. Dabney, T. L. Harman, "What is Simulink", *Mastering Simulink 4*, 2001, Prentice Hall Inc.
7. J. P. Holman, "Properties of Water", *Heat Transfer*, 2001, McGraw-Hill
8. Khairul Arifin bin Dolah, "ANFIS Training", *Modelling and Simulation of Process Boiler Plant*, 2003, UTP.
9. Chiu, S., "Fuzzy Model Identification Based on Cluster Estimation", *Journal of Intelligent & Fuzzy Systems*, Vol. 2, No. 3, Sept. 1994.
10. Jose C. Principe, Neil R. Euliano, W. Curf Lefebvre, "Neural and Adaptive System", *Fundamental through Simulation*, John Wiley & Sons Inc.
11. J. Wesley Hines, "MATLAB Supplement to Fuzzy and Neural Approaches in Engineering", Wiley Inter. Science.

## APPENDIX 2

### MATLAB S-Function file for Mathematical Model of Heat Exchanger

```
function [sys, x0, str, ts] = sfd301m(t, x, u, flag, x_init)
% sfd301 S-function version of the two state model
% This version is compatible with simulink block diagrams.

% Heat Exchanger Data-----

% Physical Data
Tod = 0.01;           % tube outside diameter (m)
Tid = 0.0095225;     % tube inside diameter (m)
Att = pi*Tod*1;      %21.336^2; % total tube surface area (m^2)
Sid = 0.254;         % shell inside diameter (m)
Ntube = 133;         % number of tubes
Npass = 1;           % number of tube passes

% Water Properties-----
% (at 60 degC from Holman J.P. p. 650 Table A-9)
cp = 4.179;          % specific heat (kJ/kg.K)
rho = 983.3;         % density (kg/m^3)
k = 0.654;           % thermal conductivity (W/m.K)
mu = 4.71;           % viscosity (kg/m.s)
Pr = 3.01;           % prandtl number (dimensionless)

%-----
if abs(flag) == 1, %COMPUTE STATE DERIVATIVES

% State initial conditions
Ts = x(1);           % final shell temperature
Tt = x(2);           % final tube temperature

% Controls or inputs
Qs = u(1);           % shell side inlet flow rate (m^3/h)
Qt = u(2);           % tube side inlet flow rate (m^3/h)
Ts_init = u(3);      % initial shell water temperature
Tt_init = u(4);      % initial tube water temperature

% Perimeter Calculation

% Calc heat transfer coeff on inside of tube
Aft = (pi*Tid*Tid/4)*Ntube; % tube flow area
Velt = Qt/(60*Aft);        % flow velocity tube side
Ret = rho*Velt*Tid/mu;     % Reynolds number tube side
Pet = Ret*Pr;              % Peclet number tube side
Nut = 0.625*Pet^(0.4);     % Nusselt number tube side
ht = k*Nut/Tid;           % heat transfer coeff tube side

% Calc heat transfer coeff on shell side
L = Att/(pi*Tod*Ntube*Npass); % tube length
Vtot = (pi*Sid*Sid/4)*L;     % total vol inside shell
Ltot = L*Ntube*Npass;        % total length of all tubes
Vtube = (pi*Tod*Tod/4)*Ltot; % vol occupied by tubes + tube fluid
Vs = Vtot-Vtube;            % shell side volume
Afs = Vs/L;                 % shell side flow area
Phs = pi*Tod*Ntube*Npass;   % heated perimeter on shell side
De = 4*Afs/Phs;             % equiv diameter
Vels = Qs/(60*Afs);         % flow velocity shell side
Res = rho*Vels*De/mu;       % Reynolds number shell side
```

```

Pes = Res*Pr;           % Peclet number shell side
Nus = 0.106*Pes^(0.6); % Nusselt number shell side
hs = k*Nus/De;         % heat transfer coeff shell side

% Simplified constant
b1=ht*pi*Tid/(Aft*rho*cp);
b2=hs*pi*Sid/(Afs*rho*cp);

% State equations

e = [1 0; 0 1];
b = [-Vels*(Ts-Ts_init)/100-b1*(Ts-Tt); -Velt*(Tt-Tt_init)/100+b2*(Ts-Tt)];

% Solve linear equation for derivatives
dx = e\b; %backward slash dx = inv(e)*b

% Step the derivatives, dx is a 2X1 vector
sys = [dx];

%-----
elseif flag == 3, % COMPUTE OUTPUTS

% Extract state variables
Ts = x(1);
Tt = x(2);
sys = [Ts, Tt];

%-----
elseif flag == 0, % INITIALIZATIONS
sizes(1) = 2; % 2 continuous states
sizes(2) = 0; % 0 discrete states
sizes(3) = 2; % 2 outputs
sizes(4) = 4; % 4 inputs
sizes(5) = 0; % No discontinuous roots
sizes(6) = 0; % No direct term
sizes(7) = 1; ts = [0,0]; % Efficiency fix (see [1])
sys = sizes';
x0(1) = x_init(1);
x0(2) = x_init(2);

else % Flags 2 or 4, not considered since we have no
sys = []; % discrete states. Also flag 5 is of no interest
end;

```

## **APPENDIX 3**

### Modification on Mathematical Model of Heat Exchanger

#### **Flow Rate Calculation**

Basically, the modification is quite straightforward with introduction of several calculation parts in order to get the process heat transfer coefficient. Then this coefficient is insert into existing partial differential equation (equation 3 and 4), which is currently inside the model.

Theoretically, the heat-transfer coefficient for water flow on the inside of the shell and tube heat exchanger is determined from the flow conditions with the properties evaluated at the bulk temperature. The flow condition can be determined by estimating it using Reynolds (Re) number:

$$Re = \rho v d / \mu$$

where  $\rho$  is density,  $v$  is velocity,  $d$  is density and  $\mu$  is viscosity.

Fom Reynolds number, we can know whether the flow is laminar or turbulent. Then this Reynolds number is used to calculate Nusselt number (Nu), which is for calculating the heat transfer coefficient. For example, in turbulent flow condition, the estimated approximation Nusselt number is:

$$Nu = 0.023 Re^{0.8} Pr^{0.4}$$

where Pr is Prandtl number, usually given in water properties table.

The Nusselt number is not fixed and it is depending on Reynolds number. For the shell and tube exchanger model, the Nusselt approximations used are:

$$\begin{aligned} \text{Nu}_{\text{tube}} &= 0.625 (\text{Re}_{\text{tube}} \text{Pr})^{0.6} && \text{(for tube side)} \\ \text{Nu}_{\text{shell}} &= 0.106 (\text{Re}_{\text{shell}} \text{Pr})^{0.6} && \text{(for shell side)} \end{aligned}$$

After we get the Nusselt number, the heat transfer coefficient is given by:

$$h = \text{Nu} (k / d)$$

where  $k$  is thermal conductivity and  $d$  is the tube diameter.

All these are equations that show that the inlet flow is affecting the heat transfer coefficient. In addition, by having flow input variable, we can have variable velocity since it is equal to flow divided by area. As a result, velocity will affect the heat transfer coefficient as well as the temperature differential equation as shown below:

$$dT_1/dt = -v (dT_1/dx) - [h\pi d / a\rho C] (T_1 - T_2)$$

#### Equation Involved with S-function Code

From equation (3), by putting  $dT_1/dx = (T_1 - T_0) / s$ , where  $s$  is sampling interval along time axis, we can rearrange equation (3) into:

$$\boxed{\frac{\delta T_1}{\delta t} = -v_1 \frac{T_1 - T_0}{s} - \frac{h_1 \pi d}{a_1 \rho C_p} (T_1 - T_2)} \dots(5)$$

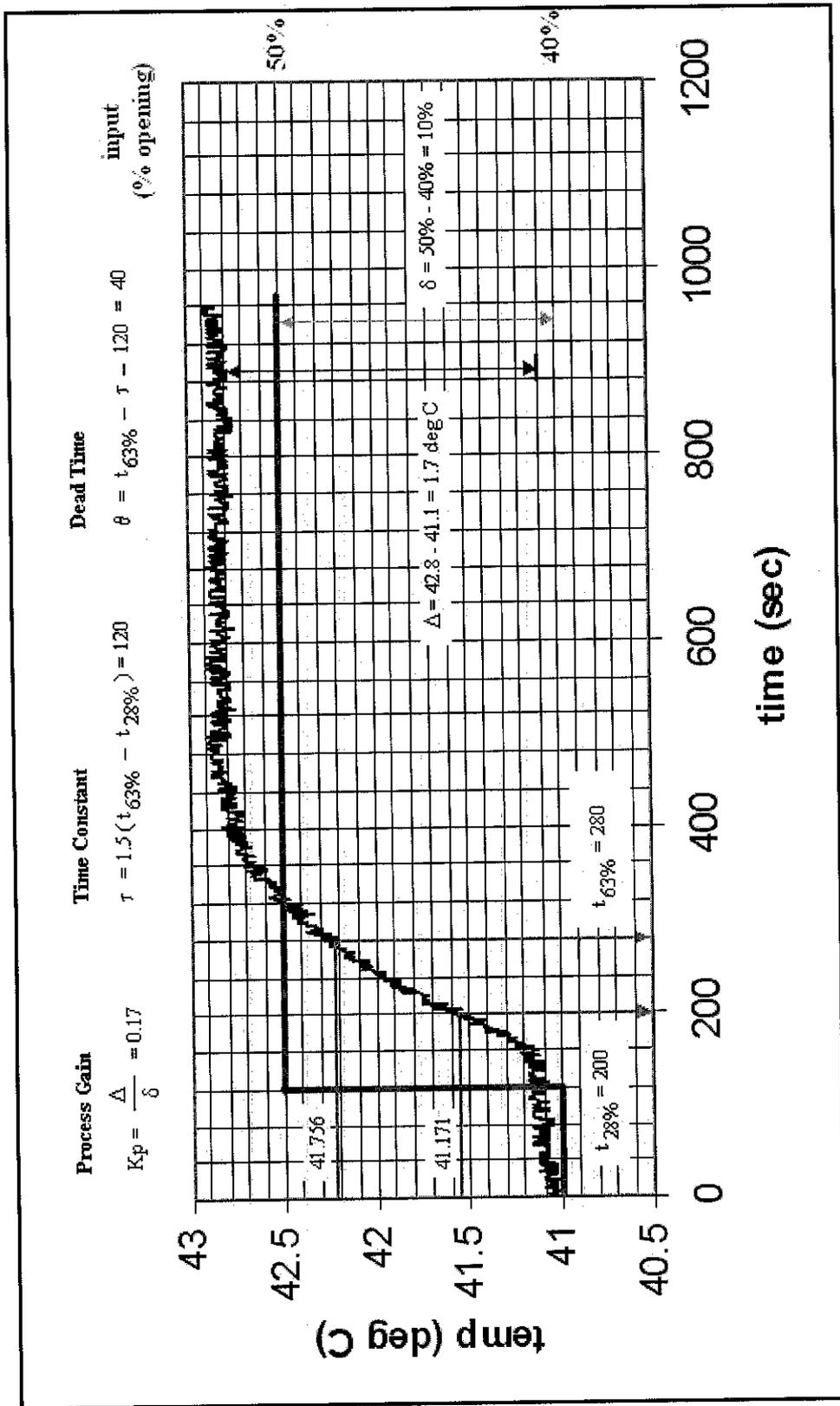
By using equation (5), we can have the heat exchanging temperature trend where supposedly the heating medium temperature is decreasing while the heated medium is increasing in temperature.

Before proceeding with the parameter calculations, we need to have basic modelling data, which is vital in determining the temperature response. The physical data should be collected from manufacturer specification sheet while water properties are taken from Holman [7].

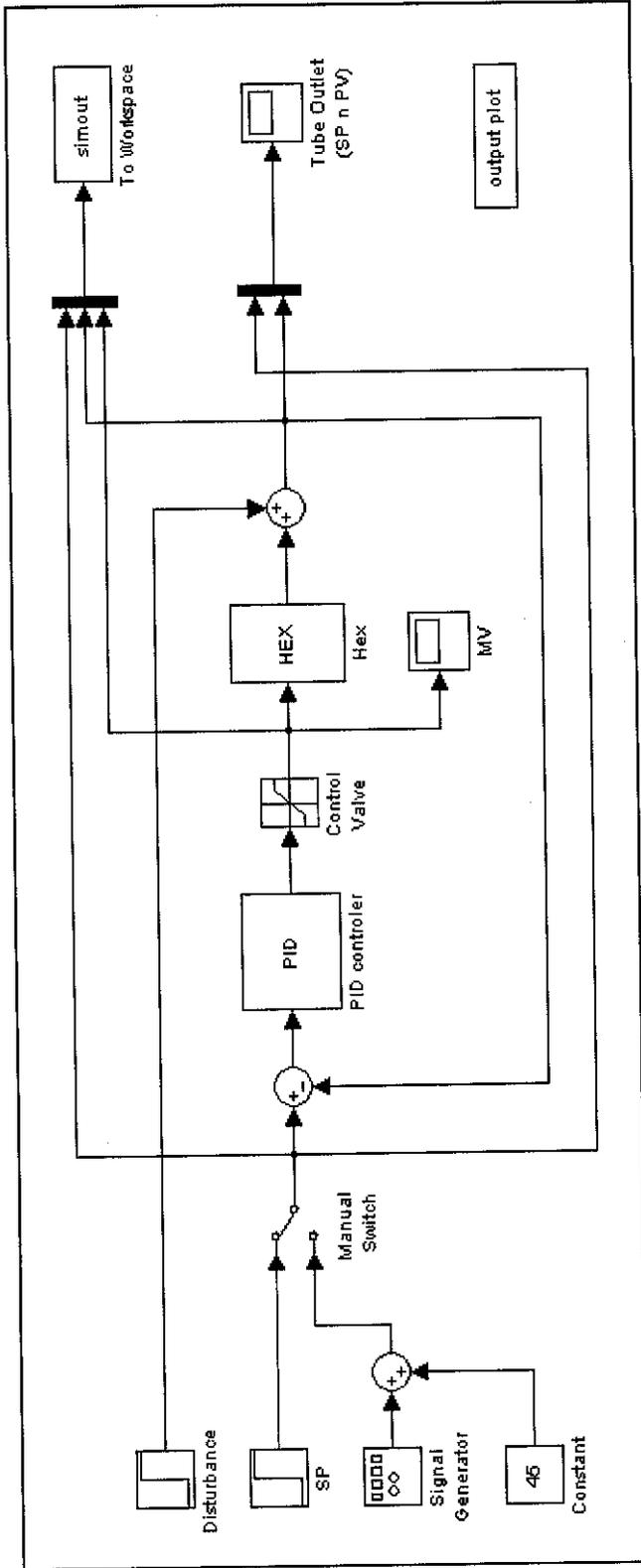
However, due to several reasons, the physical data is collected based on physical approximation. The basic modelling data for the HEX is:

HEX Physical Data		Water Properties	
Parameter	Value	Parameter	Value
Tube outside diameter	0.01 (m)	Specific heat	4.179 (kJ/kg.K)
Tube inside diameter	0.0095225 (m)	Density	983.3 (kg/m <sup>3</sup> )
Total tube surface area	0.03142 (m <sup>2</sup> )	Thermal conductivity	0.654 (W/m.K)
Shell inside diameter	0.254 (m)	Viscosity	4.71 (kg/m.s)
Number of tubes inside HEX	133	Prandtl number	3.01(dimensionless)
Number of tube passes	1		

Basically, all above values are used to calculate the heat transfer coefficient for both shell and tube side. This is because instead of using overall heat transfer coefficient, we use specific shell and tube heat transfer coefficient, which is denoted as  $h_1$  that supposedly give better approximation to outlet shell and tube temperature.



APPENDIX 4: Process Reaction Curve from Open Loop Test



APPENDIX 5: Simulink Block Diagram for Empirical Model with PID Controller

## APPENDIX 6

### PI Tuning Parameters Calculation for Giancone and Lopez

Given:

Process Gain,  $K_p = 0.17 \text{ } ^\circ\text{C}/\% \text{ opening}$

Time Constant,  $\tau = 120 \text{ seconds}$

Time Delay,  $\theta = 40 \text{ seconds}$

**Fraction dead time,  $\theta / (\theta + \tau) = 0.25$**

#### Giancone

From Figure 9.5 at Marlin<sup>[4]</sup> (pg281)

$$K_c K_p = 1.5$$

$$T_i / (\theta + \tau) = 0.9$$

Thus,

$$K_c = 1.5 K_p = 1.5 \times 0.17 = 8.82$$

$$T_i = 0.9 \times (\theta + \tau) = 0.9 \times 160 = 144$$

#### Lopez

From Figure 9.10 at Marlin<sup>[4]</sup> (pg287)

$$K_c K_p = 2.4$$

$$T_i / (\theta + \tau) = 0.68$$

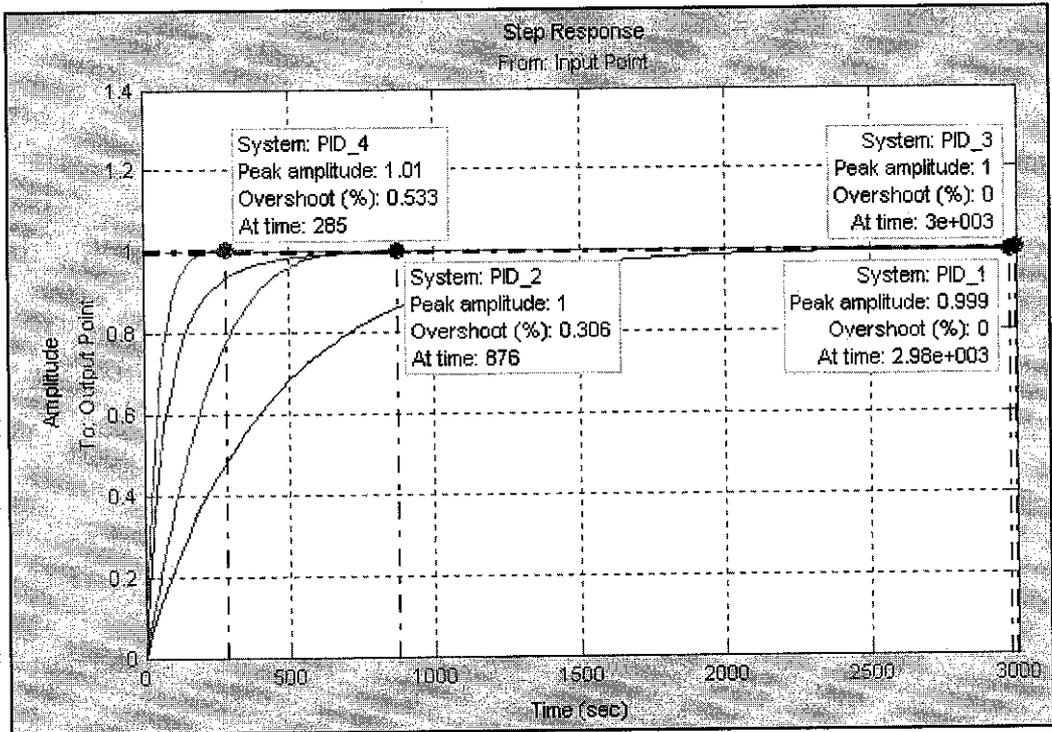
Thus,

$$K_c = 2.4 K_p = 2.4 \times 0.17 = 14.12$$

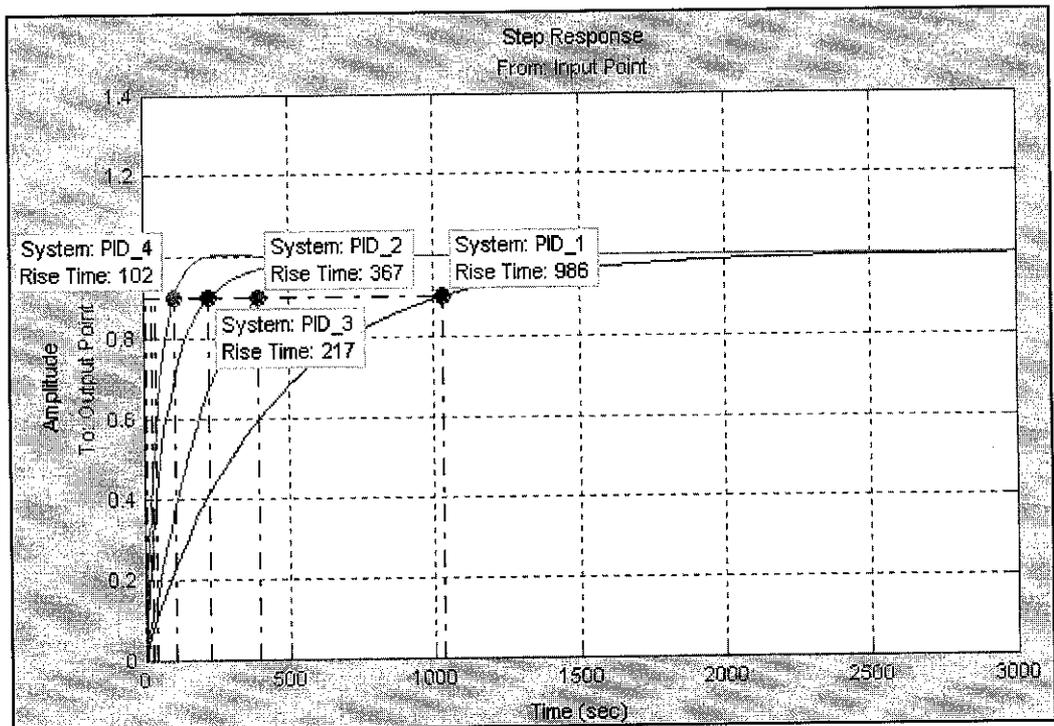
$$T_i = 0.68 \times (\theta + \tau) = 0.68 \times 160 = 108.8$$

## APPENDIX 7

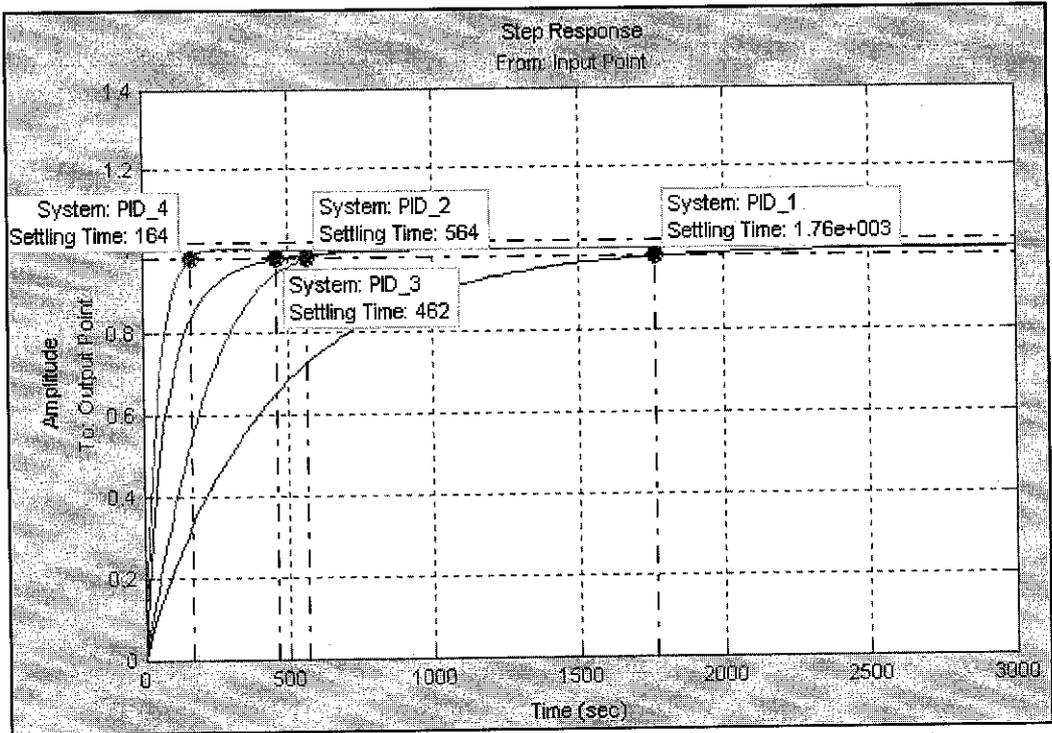
### PV and MV Characteristics from Different Tuning Parameters



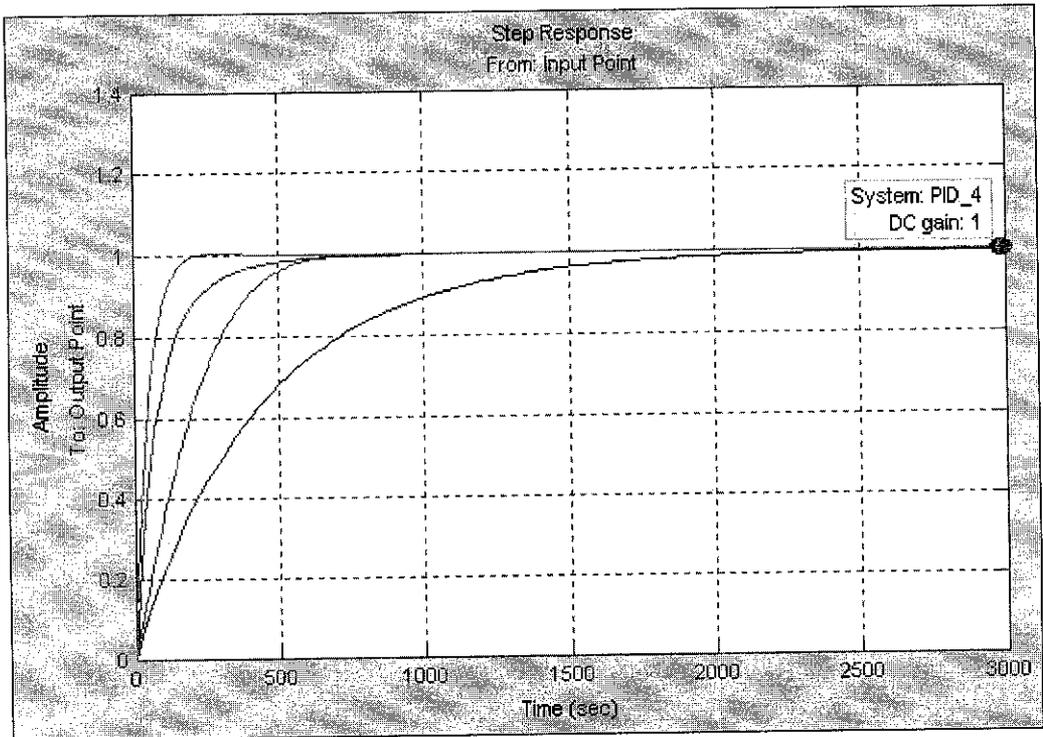
Peak response for PV



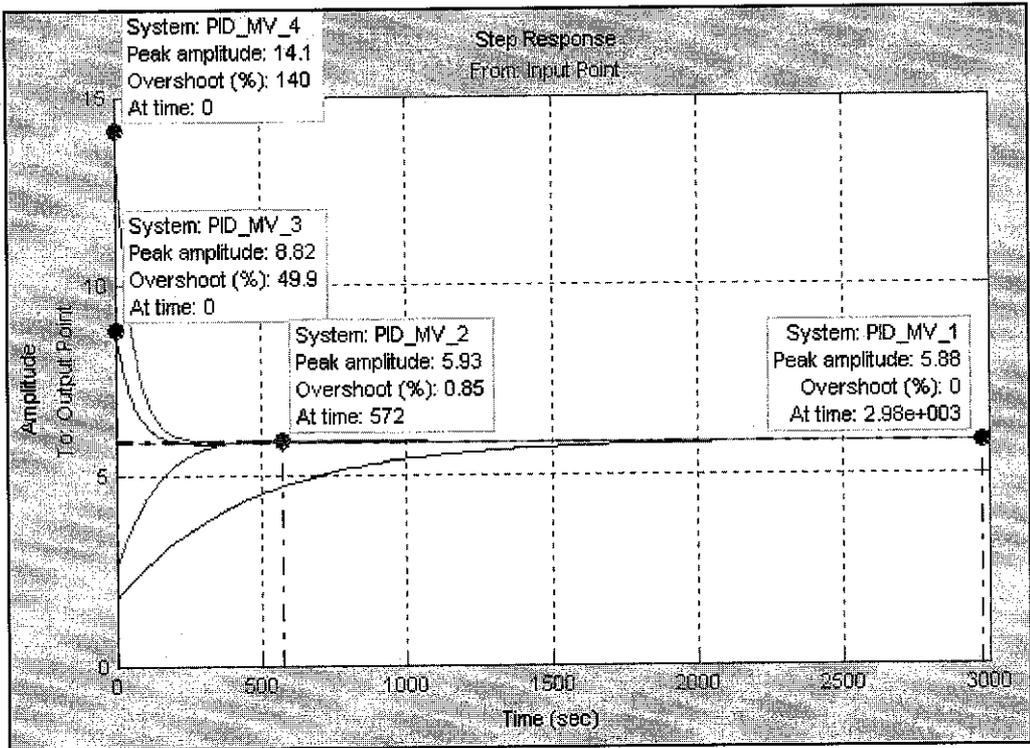
Rise time for PV



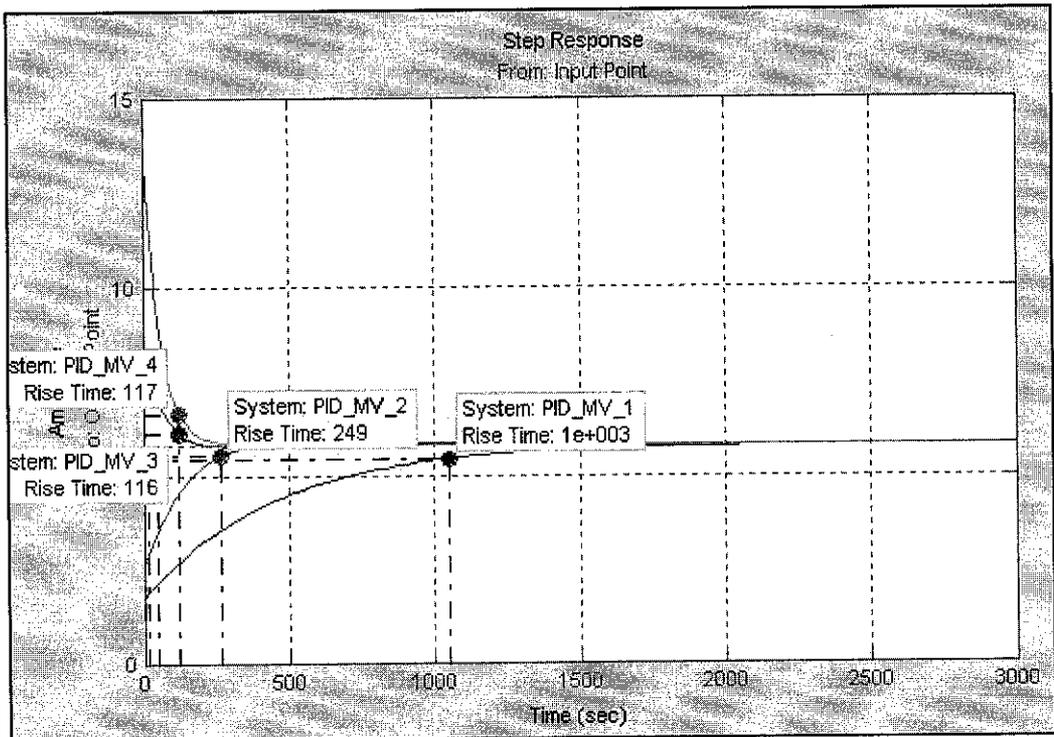
Settling time for PV



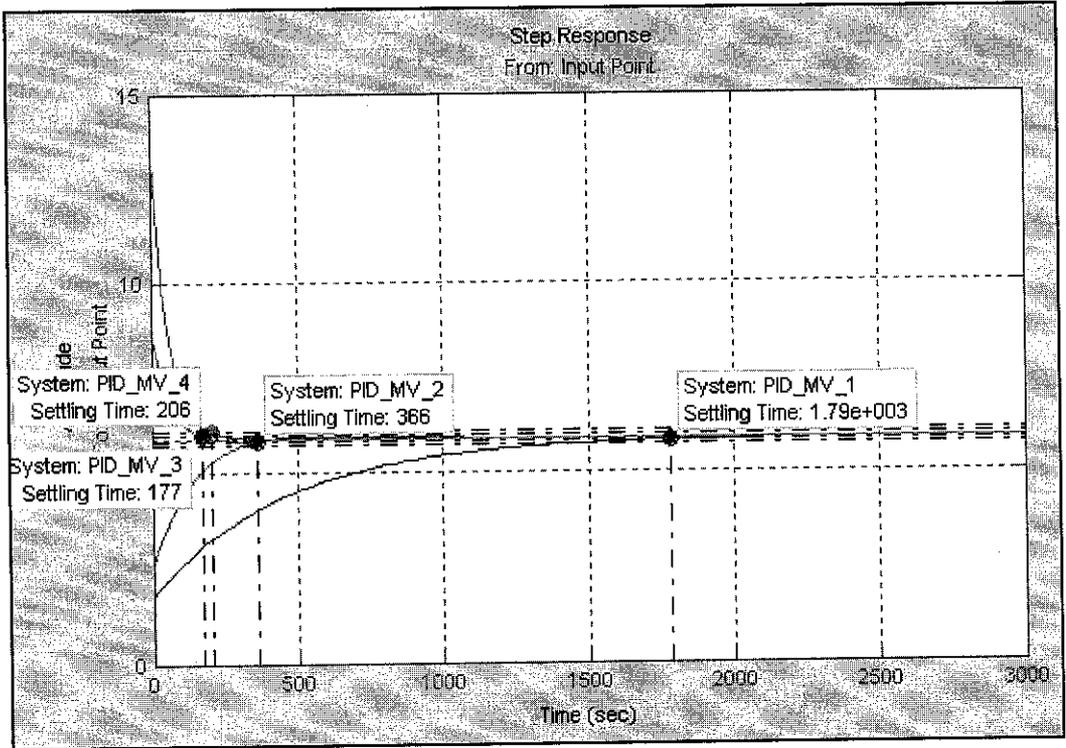
Steady State Gain for PV



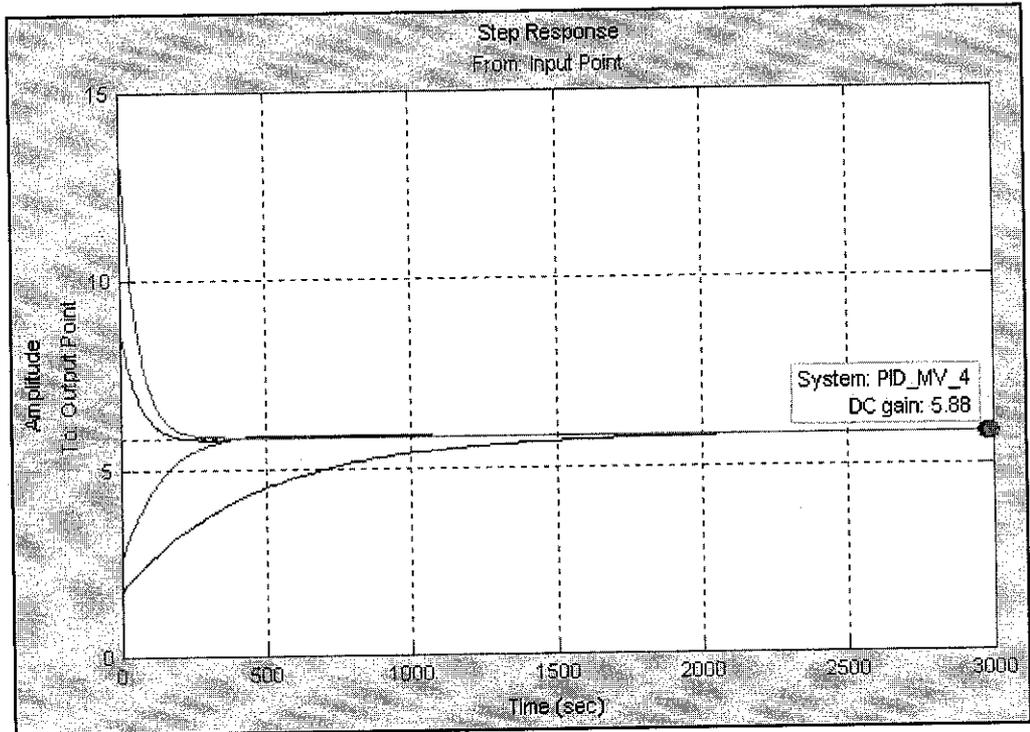
Peak Response for MV



Rise time for MV



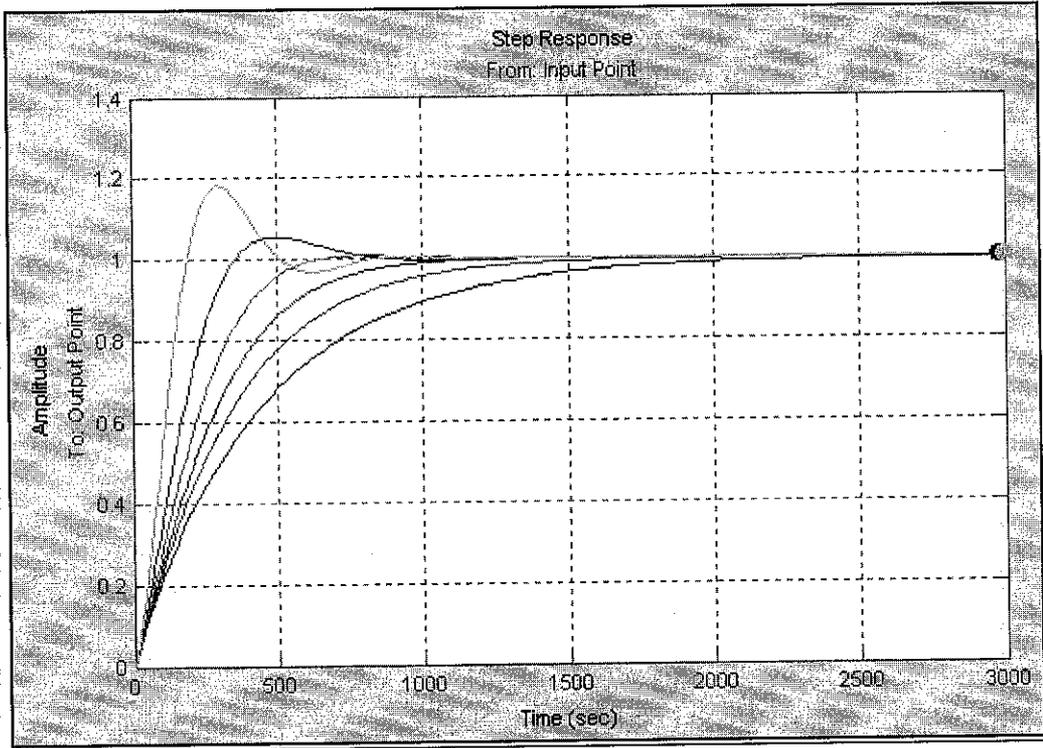
Settling time for MV



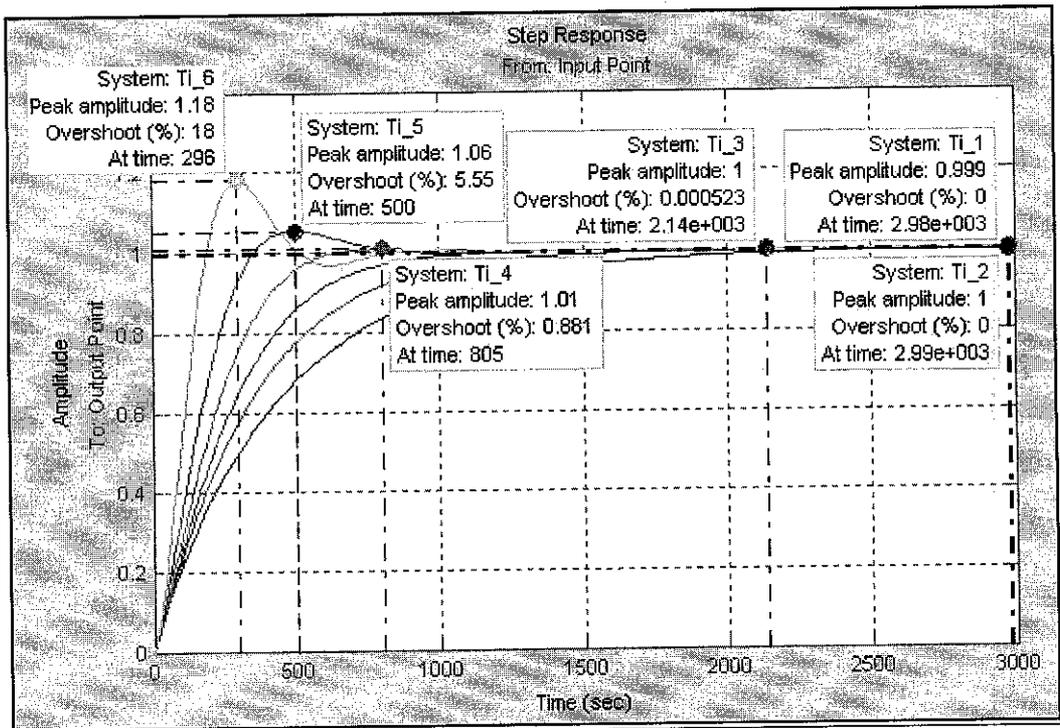
Steady State Gain for MV

## APPENDIX 8

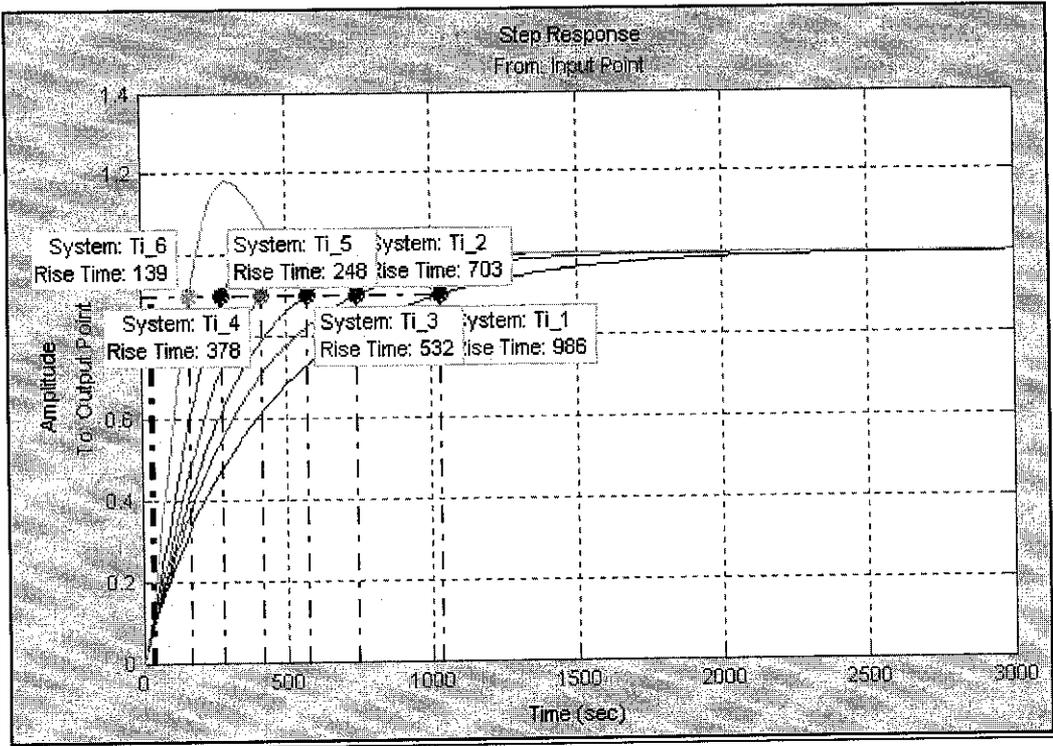
### Fine Tuning on PI Controller – (retuning $T_i$ )



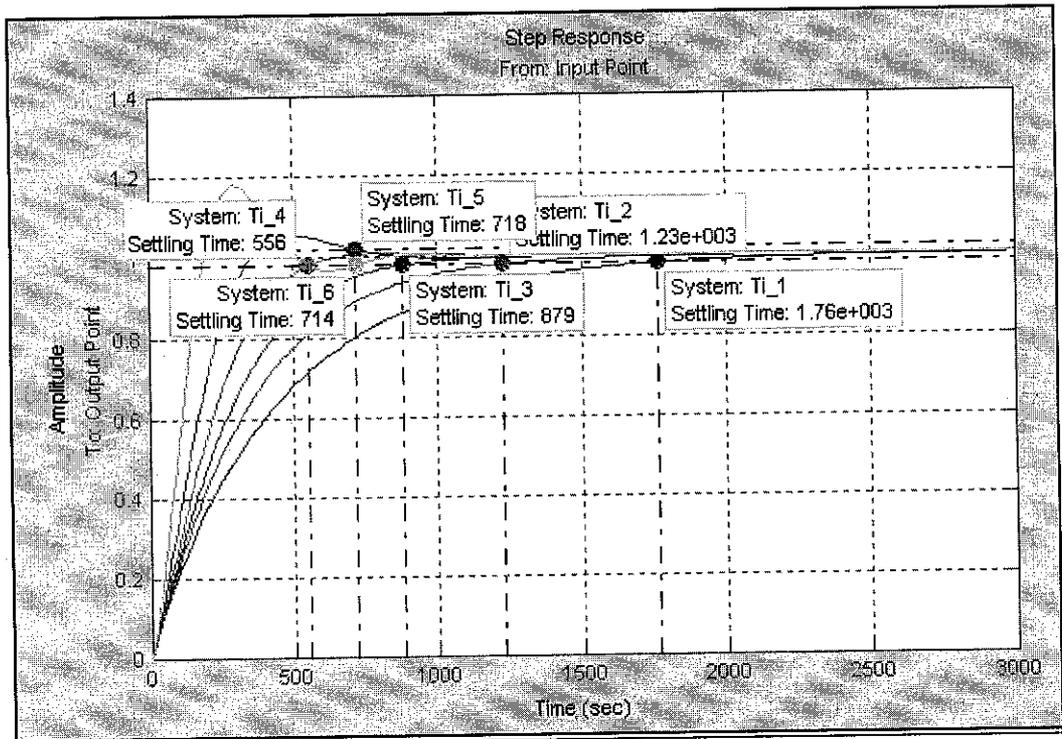
Step Response Overview for PV



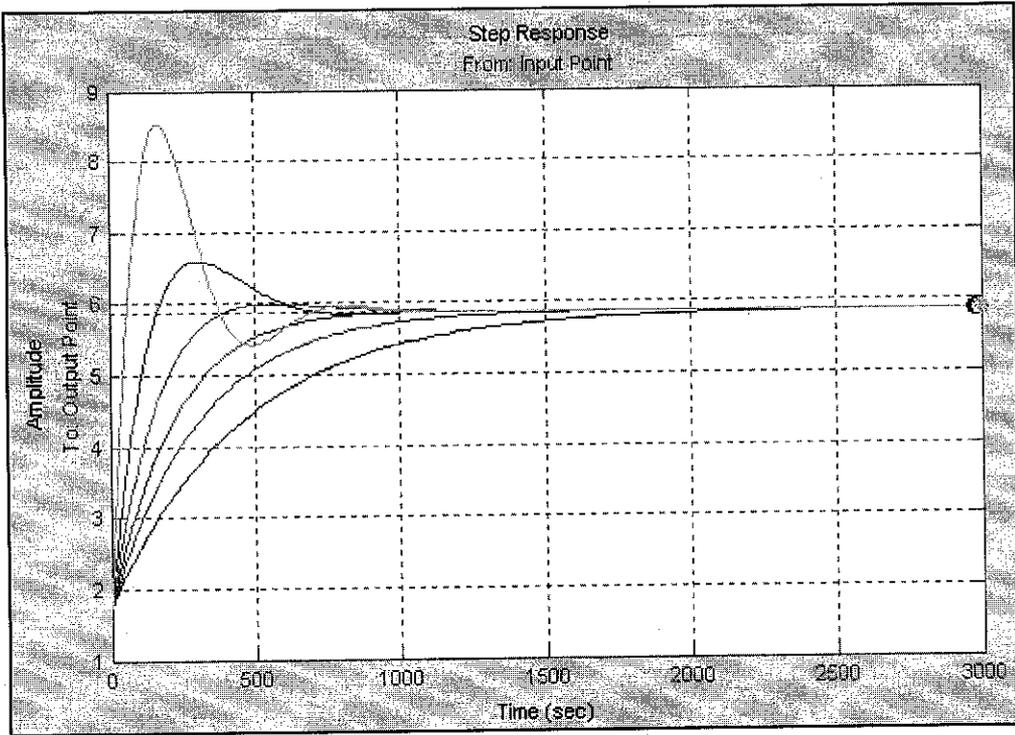
Peak Response for PV



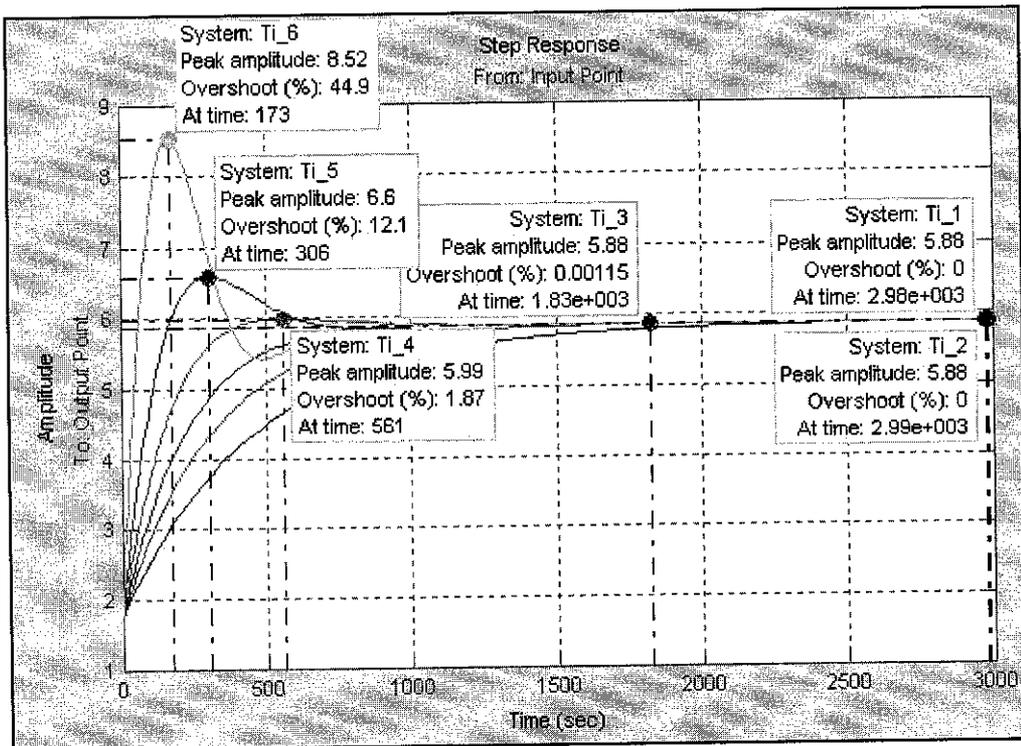
Rise Time for PV



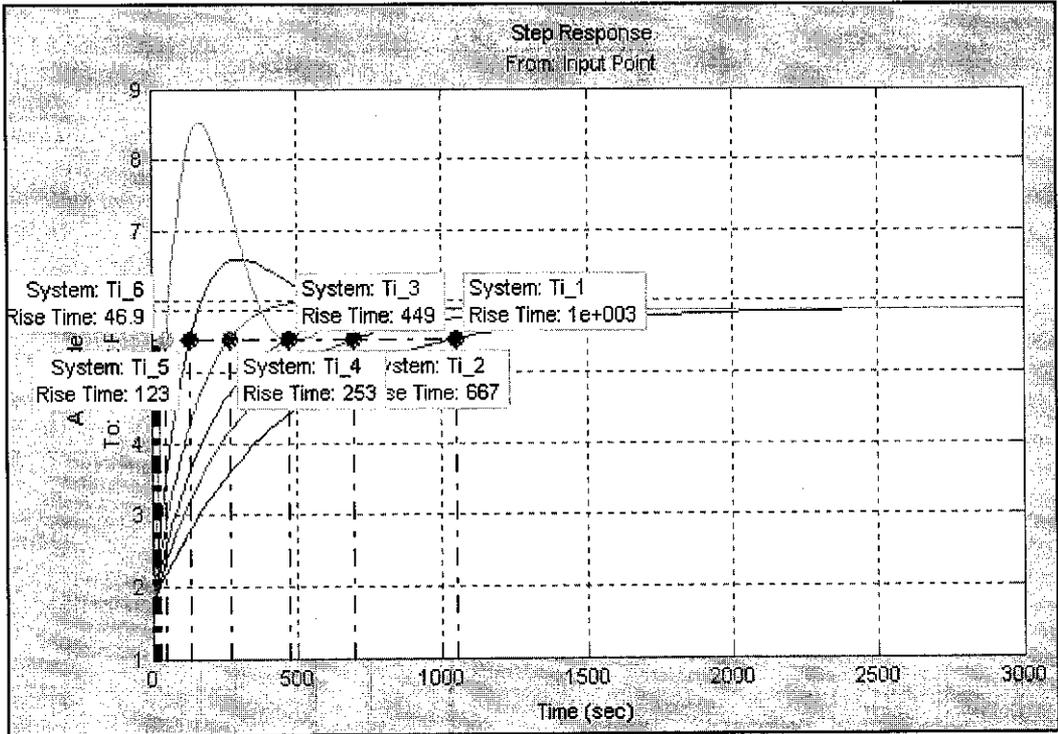
Settling Time for PV



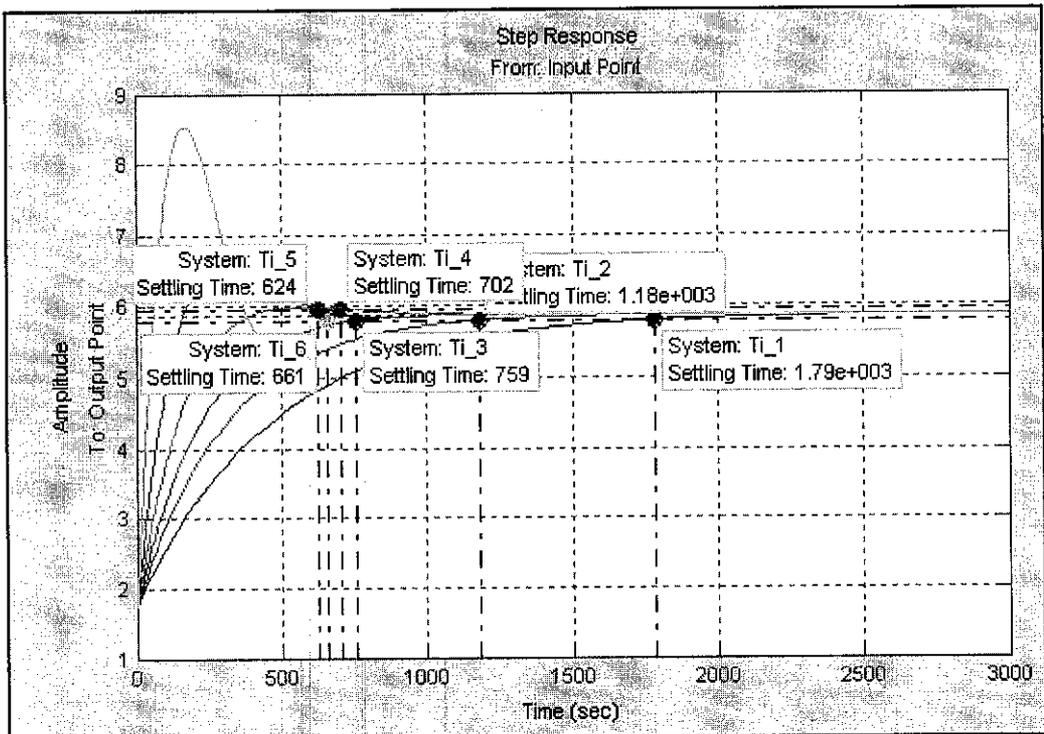
Step Response Overview for MV



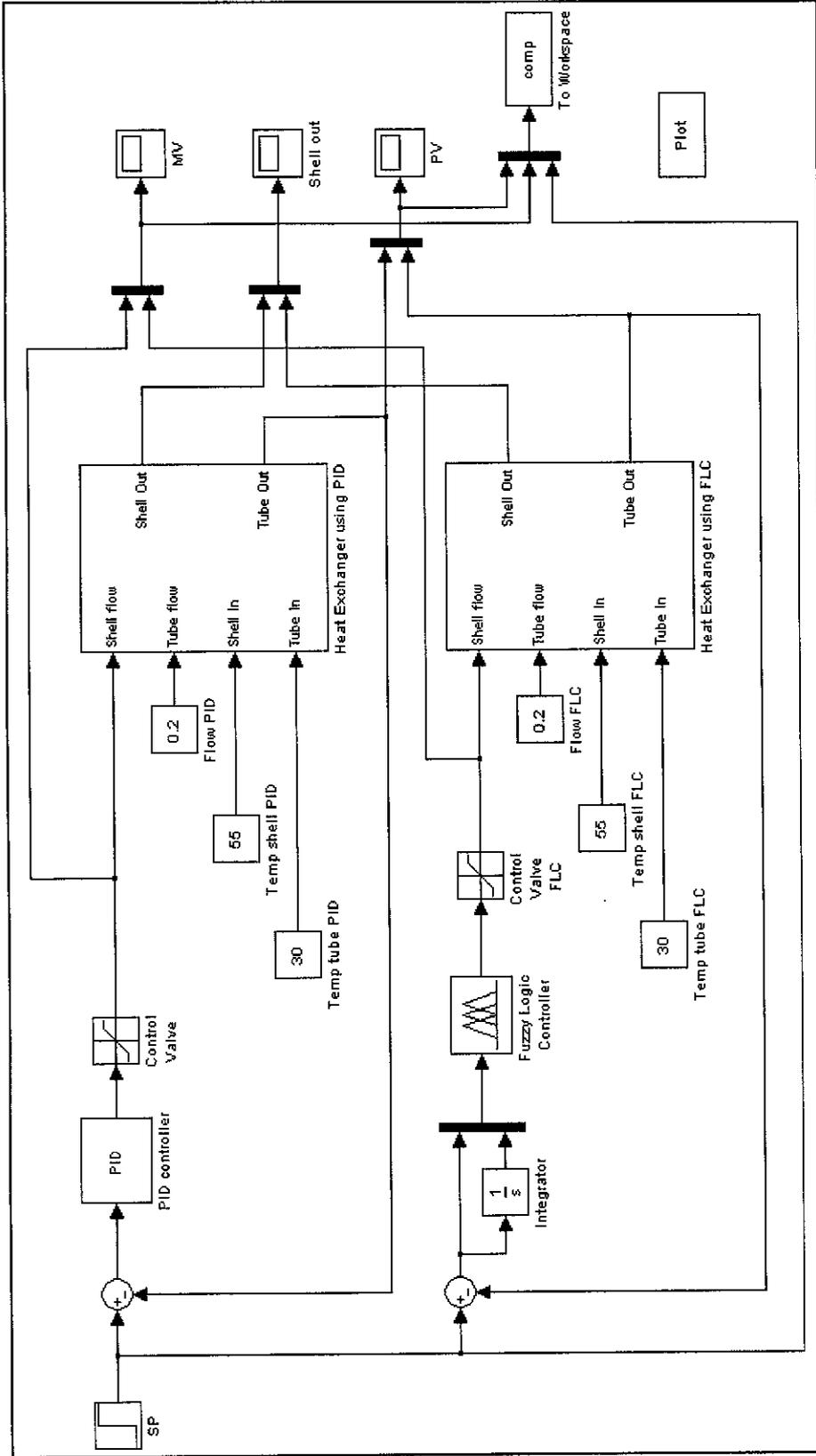
Peak Response for MV



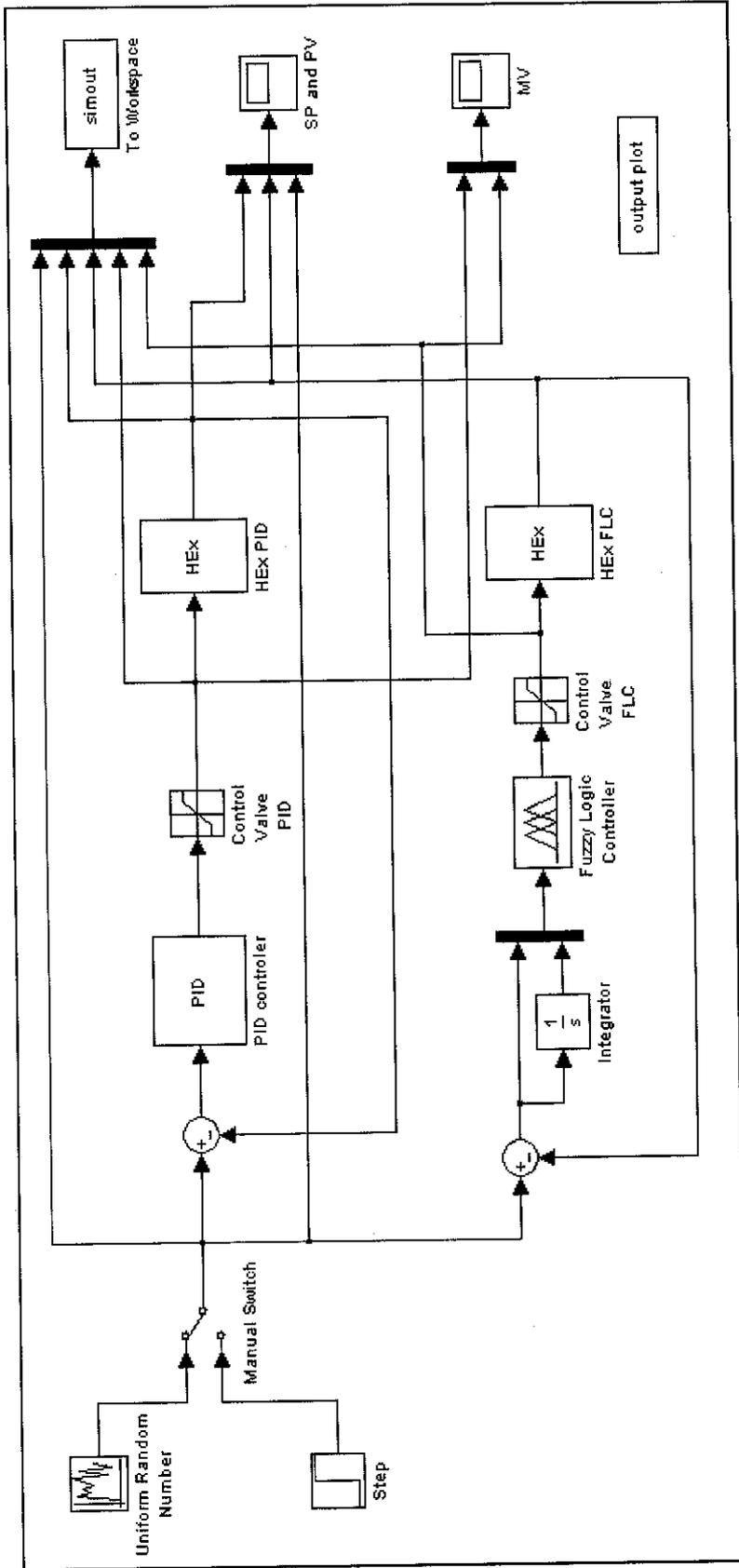
Rise Time for MV



Settling Time for MV



**APPENDIX 9: Simulink diagram of FLC and PID Controller for Math Model**



APPENDIX 10: Simulink diagram of FLC and PID Controller for Empirical Model