Alternative Water Level Controller using Artificial Intelligence for Industrial Drum Boiler.

by

Hasinah bt. Mohamed

Dissertation submitted in partial fulfilment 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

CERTIFICATION OF APPROVAL

Alternative Water Level Controller using Artificial Intelligence for Industrial Drum Boiler.

by:

Hasinah Mohamed

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

Approved by,

Mr. Fawnizu Azmadi Hussin

Fawnizu Azmadi Hussin
Lecturer
Electrical & Electronics Engineering
New Academic Block NO 22
Universiti Teknologi PETRONAS
31750 Tronoh
Perak Darul Ridzuan, MALAYSIA.

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
June 2004

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.

HASINAH MOHAMED

Abstract

This work is based on the paper entitled '*Overcoming the shrink and swell effect in water level control strategy on industrial drum boiler*' [1]. The basic function of this boiler is to produce steam. The steam will allow the turbine to function and therefore, will generate electricity to users. This paper illustrates the optimisation of load-following scheme by applying 3-element controller scheme to regulate the drum boiler's water level. The PID controller is, by far the most commonly used for controlling and monitoring purposes in process plants. Nevertheless, the performance specifications of a PID controller could be further improved.

This Final Year Project entitled '*Alternative Water Level Controller using Artificial Intelligence (AI) for Industrial Drum Boiler*'. It applies neural network, neuro-fuzzy and pre-processing to compare the control performance of the steam pressure and water level of the boiler model.

The objective of the project is to design and construct software by using Matlab, which can simulate and enhance the controller of water level and steam pressure in drum boiler. The Artificial Intelligence approach makes it easier to conceptualize and implement control systems. The process is reduced to a set of visualizable steps. This is a very important point. Actually implementing a control system, even a simple control system, is more difficult than it appears. Unexpected aberrations and physical anomalies inevitably occur.

In reading about AI control applications in industry, one of the significant points that stand out is: AI is used because it shortens the time for engineering development. AI enables engineers to configure systems quickly without extensive experimentation and to make use of information from expert human operators who have been performing the task manually.

Acknowledgement

In the name of Allah, The Most Graceful and Most Merciful

The author would like to give express her deepest gratitude to Mr. Fawnizu Azmadi Hussin for his assistance during the whole course of this final year project. Without his advice, motivation and support as a supervisor, this project could not be completed within the time frame

Appreciation is given to Miss Hawa, En. Musa, Miss Hazrin Hanny bt. Mohd. Hanif. Miss faizah bt. Othman, Miss Ziyadatina bt. Abdul Rahman, Miss Asmalilah bt. Abdul Rahim, Miss Azuwa bt. Ali, Miss Nur Azlin bt. Mohd. Nor, Miss Nik Fadzliena bt. Nik Ibrahim and all the technicians in Electrical Department and IT Department for providing inspiration and caution to the author to effectively carry out the project.

Lastly, but never the least, the author would like to convey her appreciation to her beloved parents, siblings and to all friends, for the prayers, for moral and emotional support and for being there when needed most.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

1.0 INTRODUCTION

1.1 Background of Study

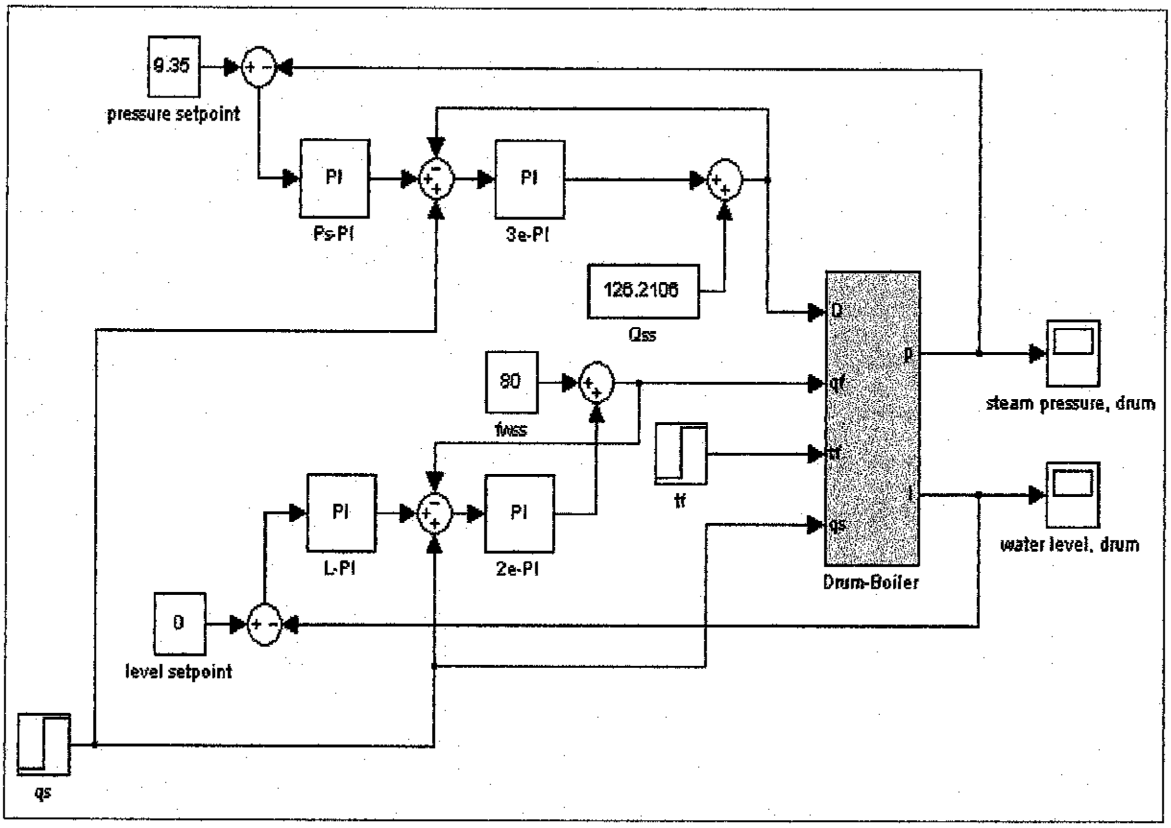


Figure 1: The model of drum boiler by using PID controller

This work is based on the paper entitled '*Overcoming the shrink and swell effect in water level control strategy on industrial drum boiler*' by Fawnizu A.H. [1]. This paper investigates 3-element controller using purely conventional PID control scheme to regulate the drum boiler's water level. The controller is then cascaded with another feedback loop that regulates steam pressure. For this paper, tuning was done using Ziegler-Nichols method.

Therefore, the writer had done some continuation based on the previous project by illustrates a few other alternative control methods using Artificial Intelligence

- Neural Network
- Neuro-Fuzzy
- Combination (Pre-processing and Neural Network)

This project will base on the simulation of the above boiler model. Based on the model, we can see that this model is using four proportional-integral (PI) controller. Proportional-Integral controllers consist of a feedback and feed forward loop. Each element of the PI controller refers to a particular action taken on the error. To most practitioners, the PI controller is robust, reliable and very well understood. Basically, this model is using MATLAB Simulink as a main tool.

In order to increase the performance of this model, the writer had used neural network toolbox, fuzzy logic toolbox and also Adaptive Neuro-Fuzzy Inference System (ANFIS). By using these toolboxes, the writer had tried to reduce the fluctuation that occurred to the outputs which are water level and steam pressure. On the other hand, a neuro-fuzzy system tends to be both robust and intelligent. It is the combination of neural-network and fuzzy systems in such a way that neural-network are used to determine parameters of fuzzy systems. The main intention of this particular approach is to create or improve a fuzzy system automatically by means of neural network methods. Neural networks are computer algorithms inspired by the way information is processed in the nervous system.

An important difference between neural networks and standard IT solutions is their ability to learn. This learning property has yielded a new generation of algorithms that can learn from the past, to predict the future; extract rules for reasoning in complex environments; offer solutions when explicit algorithms and models are unavailable or too cumbersome. The main idea of neural network control is to build a model of a human control expert who is capable of controlling a plant without thinking too much in terms of a mathematical model. The model specifies control actions in the form of linguistic rules. These control rules are translated into the framework of fuzzy set theory providing a calculus, which can simulate the behaviour of the model.

1.2 Problem statement

The output for water level and steam pressure would overshoot for a few seconds and then arrive at the desired set point value after a series of oscillations, as shown below.

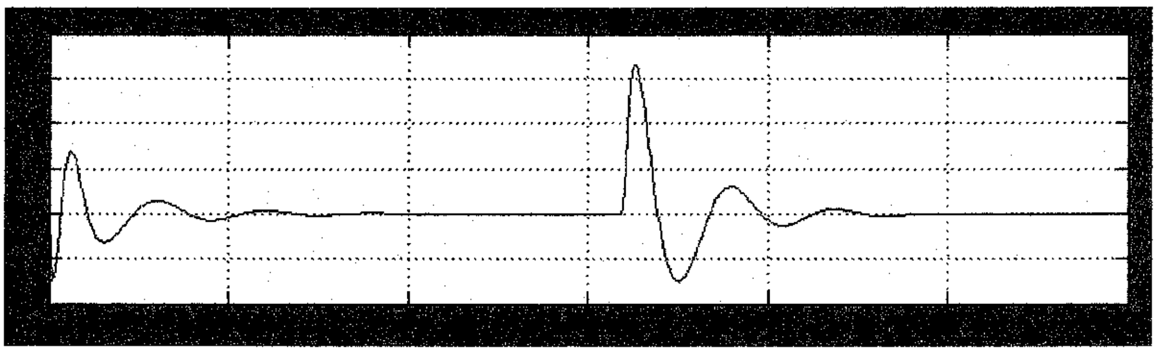


Figure 2 : System response with PID Controller (Water Level)

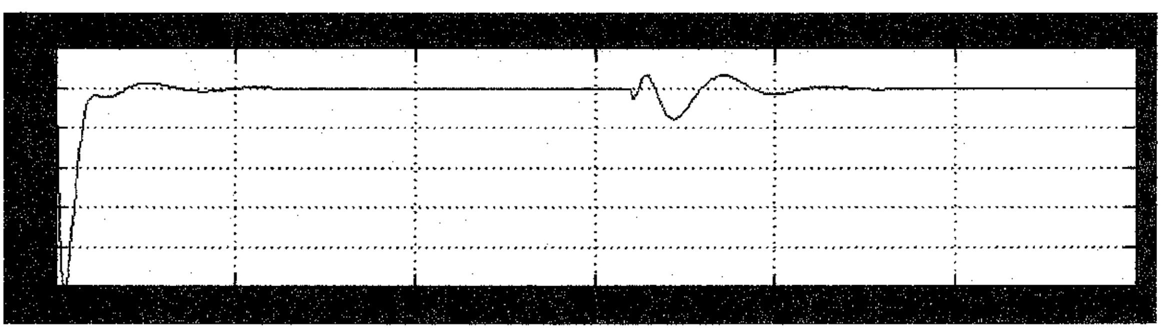


Figure 3 : System response with PID Controller (Steam Pressure)

The PID controller is, by far the most commonly used for controlling and monitoring purposes in process plants. As previously stated, the PID controller is robust. Nevertheless, the performance of a PID controller could be further improved. The proposed Artificial Intelligent (AI) controller, which applies neural network and fuzzy logic, is used for comparison.

1.3 Significance of the project

Through this project, the investigation on the development of the artificial intelligence controller for a boiler model is carried out. The outcome of the investigation serves as the base for designing an alternative controller, which uses the artificial intelligence for a power plant. This result would eventually be used to replace control systems using conventional PID.

1.4 Objectives

The objectives are:

1. To conduct research on the present PID controller of water level on Industrial Drum Boiler
2. To design alternative AI controllers using Neural Network Toolbox, Fuzzy Logic Toolbox and ANFIS.

1.5 Scope of study

The study of the Water Level Drum Boiler will cover several of engineering subjects such as control system, power plant and control system, engineering mathematics and also probability and statistics. In designing the system, the knowledge of programming is highly required. It is also essential to understand the boiler's process in order to make the alternative controller. In term of developing the controller, subjects such as MATLAB and Artificial Intelligence are required in order to produce good controller

1.6 The relevance of the project.

A lot of research has been done on the applications of artificial intelligence controller in process plants through out the past few years. The outcome was very promising in which some are already tried in the real process plant.

1.7 Feasibility of the Project within the Time Frame

The time frame for this project is split into two semesters:-

Semester 1:

1. Understanding the process of boiler
2. Understanding the PID algorithm for this boiler model
3. Feasibility studies to the Matlab toolboxes (ANFIS Editor)

Semester 2:

1. Feasibility studies to the Matlab toolboxes (Neural Network)
2. Feasibility to enhance knowledge through literature reading for pre-processing.
3. Feasibility to implement alternatives AI controller to the boiler model
4. Feasibility to come out with a brief final report.

CHAPTER 2

2.0 LITERATURE REVIEW.

2.1 Big Bend Power Station Neural Network-Sootblower Optimization

The intent of this project is to apply a neural network intelligent soot blowing system in conjunction with state-of-the-art controls and instruments to optimize the operation of a utility boiler and systematically control boiler fouling. This optimisation process is targeted to reduce NO generation by 30% or more, improve heat rate by 2% and reduce PM emissions by 5%. As compared to competing technologies, this could be an extremely cost-effective technology, which has the ability to be readily and easily adapted to virtually and pulverized coal boiler. [11]

2.2 Wide Range Operation of a Power Unit via Feedforward Fuzzy Control

A two level hierarchical control scheme for wide range operation of fossil fuel power units is presented. At the supervisory level, a fuzzy reference governor generates, according to a variable pressure operating policy, the set-point trajectories to command the unit along any load demand pattern. At the control level, a combination of feed forward and feedback control strategies are implemented. The feed forward control path contains a set of multi-input single-output fuzzy inference systems, designed from steady-state input-output plant data. The feedback control path consists of PID controllers in a multi-loop configuration, as currently available at power units. With this strategy, the feed forward path provides most of the control signal component for regulation and disturbance rejection in small neighbourhoods about the commanded trajectories. Simulation results demonstrate the feasibility of the control scheme to attain cyclic load-following operation. [13]

2.3 Use of Artificial Neural Networks Process Analyzers: A Case Study

Boilers are found in many industrial facilities to be used both as power source and for processing purposes. They consist of a furnace, where air and fuel are combined and burned to produce combustion gases to a water-tube system. The tubes are connected to the steam drum, where the generated water vapour is withdrawn.

Optimization of the operation of boilers can result is large savings. One of the areas in which the optimisation can be performed is through the minimization of excess air. Lowering the excess oxygen from 1% to 0.5% will increase boiler efficiency by 0.25%, a savings of about $5000/year in a 100,000lb/hour boiler. Oxygen measurements are obtained through oxygen analysers.

In this paper, artificial neural network (ANN), which is known for their ability to model non-linear systems and their inherent noise-filtering abilities, are used as oxygen analyser to predict oxygen contents in a boiler at SHARQ petrochemical company in Saudi Arabia. The training data has been collected over duration do one month and used to train a neural network to develop neural based oxygen analyser. The results are very promising. [12]

2.4 Tuning of Fuzzy PID Controllers

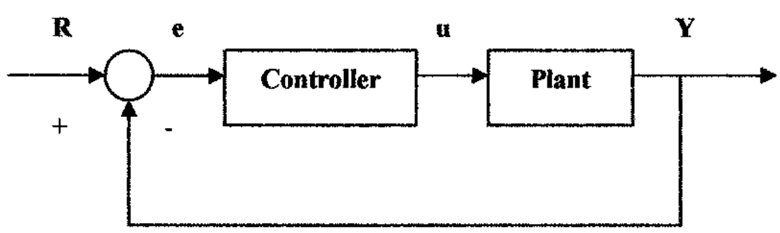
Since fuzzy controllers are non-linear, it is more difficult to set the controller gains compared to proportional-integral-derivative (PID) controllers. This research paper purposes a design procedure and a tuning procedure that carries tuning rules from the PID domain over to fuzzy single-loop controllers. The idea is to start with a tuned, conventional PID controller, replace it with and equivalent linear fuzzy controller, make the fuzzy controller non-linear, and eventually fine-tune the non-linear fuzzy controller. This is relevant whenever a PID controller is possible or already implemented. [14]

CHAPTER 3

3.0 THEORY

3.1 PID Controller.

This subtopic will show us the characteristics of the each of proportional (P), the integral (I), and the derivative (D) controls, and how to use them to obtain a desired response. In this example, we will consider the following unity feedback system:



<u>Plant</u>: A system to be controlled

<u>Controller</u>: Provides the excitation for the plant; Designed to control the overall system behaviour

3.1.1 The three-term controller

The transfer function of the PID controller looks like the following:

$$K_p + \frac{K_I}{s} + K_D s = \frac{K_D s^2 + K_p s + K_I}{s} \tag{1}$$

- K_p = Proportional gain
- K_I = Integral gain
- K_d = Derivative gain

First, let's take a look at how the PID controller works in a closed-loop system using the schematic shown above. The variable (e) represents the tracking error, the difference between the desired input value (R) and the actual output (Y). This error signal (e) will be sent to the PID controller, and the controller computes both the derivative and the integral of this error signal. The signal (u) just past the controller is now equal to the proportional gain (K_p) times the magnitude of the error plus the integral gain (K_I) times the integral of the error plus the derivative gain (K_d) times the derivative of the error.

$$U = K_p e + K_I \int edt + K_D \frac{de}{dt} \tag{2}$$

This signal (u) will be sent to the plant, and the new output (Y) will be obtained. This new output (Y) will be sent back to the sensor again to find the new error signal (e). The controller takes this new error signal and computes its derivative and it's integral again. This process goes on and on.

3.1.2 The characteristics of P, I, and D controllers

A proportional controller (K_p) will have the effect of reducing the rise time and will reduce, but never eliminate, the steady-state error. An integral control (K_I) will have the effect of eliminating the steady-state error, but it may make the transient response worse. A derivative control (K_d) will have the effect of increasing the stability of the system, reducing the overshoot, and improving the transient response. Effects of each of controllers K_p, K_d, and K_I on a closed-loop system are summarized in the table shown below.

Table 1: The effect of PID value to the graph performance

CL RESPONSE	RISE TIME	OVERSHOOT	SETTLING TIME	S-S ERROR
K_p	Decrease	Increase	Small Change	Decrease
K_I	Decrease	Increase	Increase	Eliminate
K_d	Small Change	Decrease	Decrease	Small Change

Note that these correlations may not be exactly accurate, because K_p, K_I, and K_d are dependent of each other. In fact, changing one of these variables can change the effect of the other two. For this reason, the table should only be used as a reference when we are determining the values for K_I, K_p and K_d.

3.2 NEURO-FUZZY SYSTEM.

3.2.1 Foundation of Fuzzy

Classical methods usually try to avoid vague, imprecise and uncertain information, because it is considered as having a negative influence in an inference process. Fuzzy systems on the other hand deliberately make use of this kind of information. This usually leads to simpler, more suitable models, which are easier to handle and are more familiar to human thinking.

Fuzzy logic is used in system control and analysis design, because it shortens the time for engineering development and sometimes, in the case of highly complex systems, is the only way to solve the problem.

Human beings have the ability to take in and evaluate all sorts of information from the physical world they are in contact with and to mentally analyze, average and summarize all this input data into an optimum course of action. All living things do this, but humans do it more and do it better and have become the dominant species of the planet.

Fuzzy control applications are based on if-then rules. The antecedent of a rule consists of fuzzy descriptions of measured input values, and the consequent defines a possible fuzzy, output value for the given input.

3.2.2 Fuzzy Logic Controller

The purpose of the fuzzy logic controller is to compute values of action variables from observation of state variables of the process under control. Fuzzy logic, which is the logic on which fuzzy control is based, is much closer in spirit to human thinking and natural language than the traditional logical systems. It provides an effective means of capturing the approximate, inexact nature of the real world.

Figure 4 shows the basic configuration of a fuzzy logic controller, which comprises four principal components:

- the fuzzification interface,
- the knowledge base,
- the decision-making logic
- the defuzzification interface

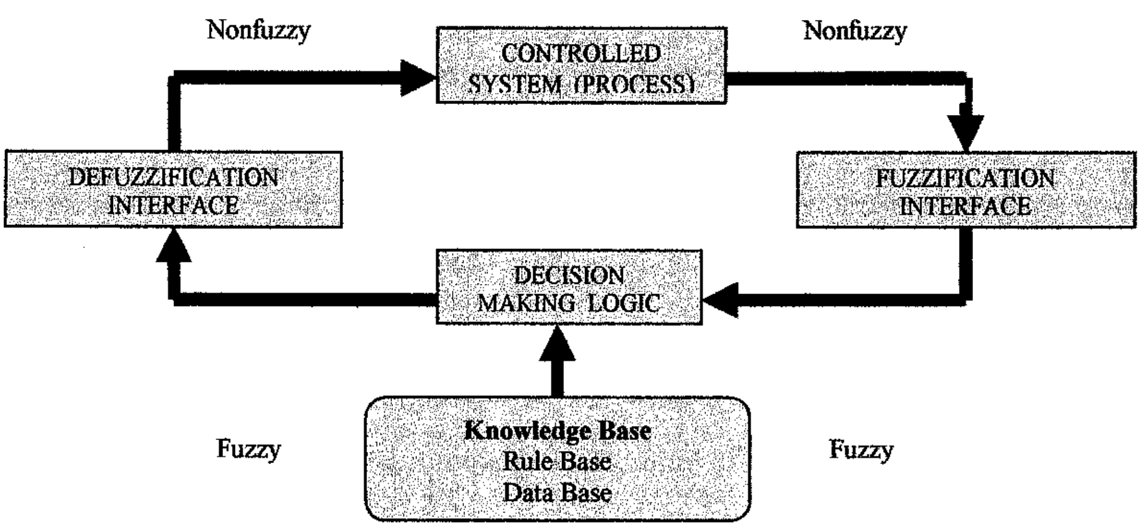


Figure 4 : Basic configuration of a fuzzy logic controller

a) Fuzzification interface

The first component is the fuzzification interface, which performs the function of fuzzification that converts input data into linguistic values, which may be viewed as labels of fuzzy sets.

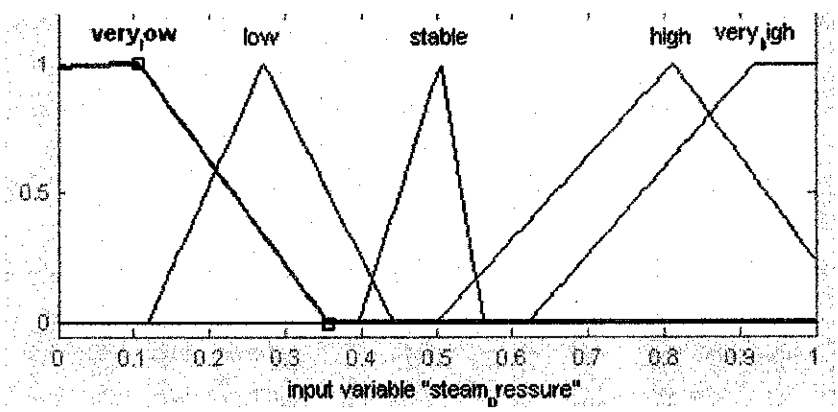


Figure 5: Fuzzification interface of steam pressure

b) Knowledge base

The second component is the knowledge base, which comprises the database and the rule base. The essential part of the fuzzy logic controller is a set of linguistic rules such as

IF the steam pressure decrease *THEN* heat input will increase (3)

The antecedent is the controlled variable and the consequent is the controlling variable. In many cases it is easy to translate an expert's knowledge in such rules. So this type of controller is used to control complex process when no precise model of the process exists and most of the information is available only in qualitative form. The database provides necessary definitions used to characterize fuzzy control rules and fuzzy data manipulation in the fuzzy logic controller. From the previous linguistic values of steam pressure and heat input, truth values will be computed.

c) Decision-making logic

The third component is the decision-making logic which simulates human decision-making and infers fuzzy control actions employing fuzzy implication and rules of inference in fuzzy logic

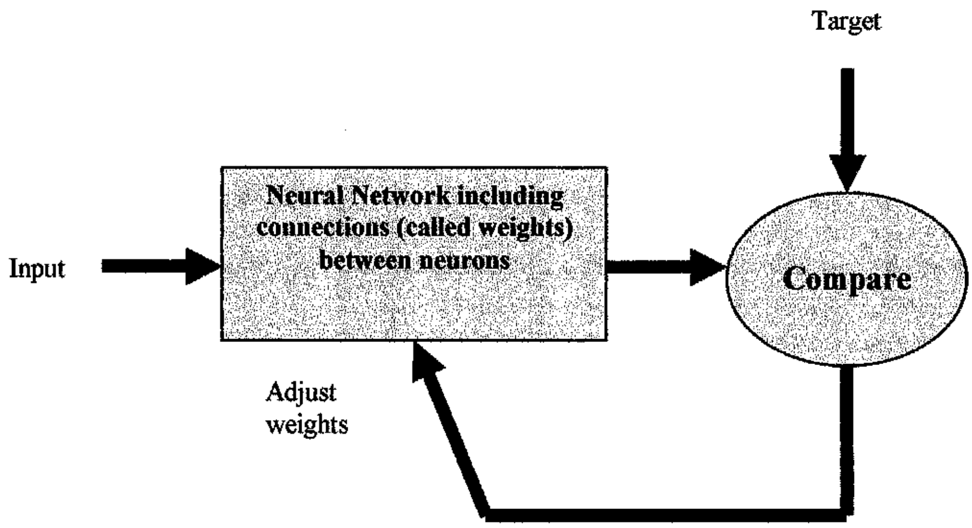
d) Defuzzification interface

And the last component is the defuzzification interfaces, which compute a nonfuzzy control action from an inferred fuzzy control action. Finally linguistic values of the control action are translated in numerical value.

3.2.3 Neural Networks

Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the network function is determined largely by the connection between elements. We can train a neural network to perform a particular function by adjusting the values of the connections (weights) between elements.

Commonly neural networks are adjusted, or trained, so that a particular input leads to a specific target output. Such a situation is shown below. There, the network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically many such input/target pairs are used, in this supervised learning, to train a network.



Neural network have been trained to perform complex functions in various fields of application including pattern recognition, identification, classification, speech, vision and control systems.

Today neural networks can be trained to solve problems that are difficult for conventional computers or human beings. Throughout the toolbox emphasis is placed on neural network paradigms that build up to or are themselves used in engineering, financial and other practical applications.

Every newcomer to the field of artificial neural networks, who wants to build own applications based on own software simulators, faces two major problems: turning the theory of a particular network model into the design for a simulator implementation can be a challenging task, and often it is not obvious how to embed an application into a particular network model. [3]

3.3 The ANFIS Architecture

One of the first hybrid neuro-fuzzy systems for function approximation was Jang's ANFIS model [15]. It represents a Sugeno-type fuzzy system in special 5-layer feed forward network architecture. Figure 6 shows the architecture of the ANFIS model generated by Jang. The inputs are not counted as a layer. ANFIS implement rules of the form:

$$\text{If } X_1 \text{ is } A_1 \text{ and } X_2 \text{ is } A_2 \text{ then } Y_1 = W_1{*}X_1 + W_2{*}X_2 + r_i \quad (4)$$

$$\text{If } X_1 \text{ is } B_1 \text{ and } X_2 \text{ is } B_2 \text{ then } Y_2 = V_1{*}X_1 + V_2{*}X_2 + r_i \quad (5)$$

The rule base must be known in advance. ANFIS adjust only the MF function of the antecedents and the consequent parameters. The rule shown above uses only 2 outputs variables. However, it is easy to use more than 2 variables. For each output variables, an additional linear combination must be specified by using an additional set of consequent parameters for each rule. For the sake of simplicity, the writer only considers ANFIS system with 2 inputs and 1 output.

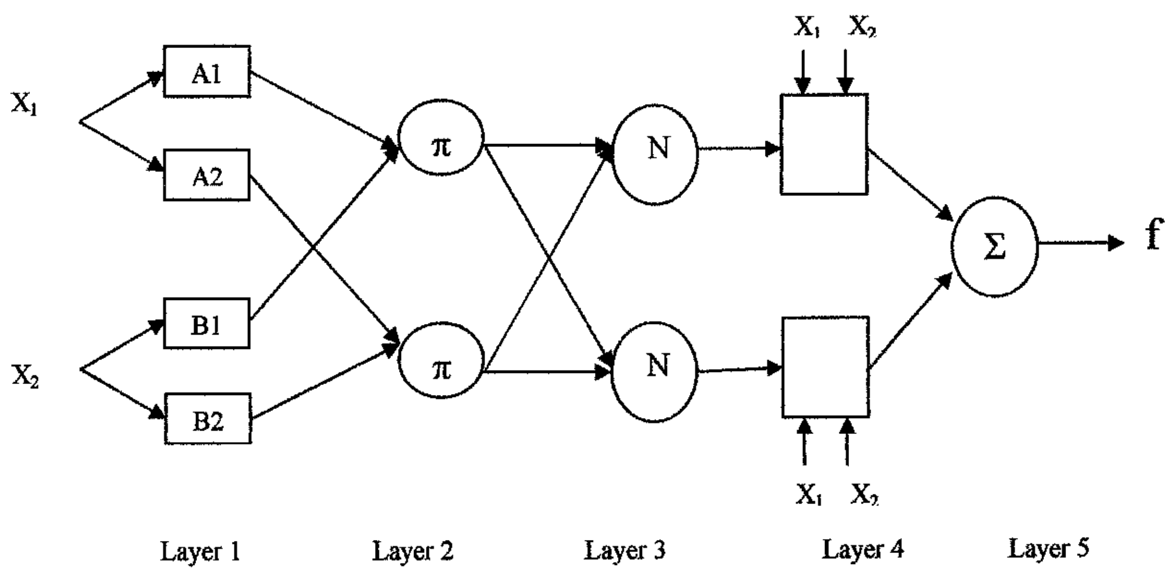


Figure 6 : ANFIS Model

The first layer is the MF layer. An output of any node in this layer gives the membership degrees of an input. The second layer is the multiplication layer. Fuzzy node here multiplies the inputs or membership degrees and produces the firing strength of the rule on the degree in which the corresponding rule is fixed.

Layer 1:

Each unit in layer 1 stores 3 parameters to define a MF.

Layer 2:

One unit in layer 2 represents each rule. Each unit is connected to those units in the previous layer which are from the antecedent of the rule. The inputs into a unit are degrees of membership which are multiplied to determine the degrees of fulfilment for the rule.

Layer 3:

In this layer for each rule, there is a unit that computes its relative degree of fulfilment. Each unit is connected to all the rule units in layer 2.

Layer 4:

The units of layer 4 are connected to all inputs and exactly one unit in layer 3. Each unit computes the output of a rule.

Layer 5:

An output unit computes the final output **f** by summing all the outputs from layer 4.

Because ANFIS uses only differentiable functions, it is easy to apply standard learning procedures from Neural Network theory. For ANFIS, a mixture of back propagation (gradient descent) and least squares estimation (LSE) is used. Back propagation is used to learn the antecedent parameters, i.e. the MF and LSE is used to determine the coefficients of the linear combinations in the rules consequents.

A step in the learning procedures has 2 parts. In the first part, the input patterns are propagated, and the optimal consequent parameters are estimated by an iterative least mean squares procedure, while the antecedent parameters are assumed to be fixed for the current cycle through the training set. In the second part, the patterns are propagated again, and in this epoch back propagation is used to modify the antecedent parameters, while the consequent parameters remain fixed.

3.4 Neuro-Fuzzy Hybrid System

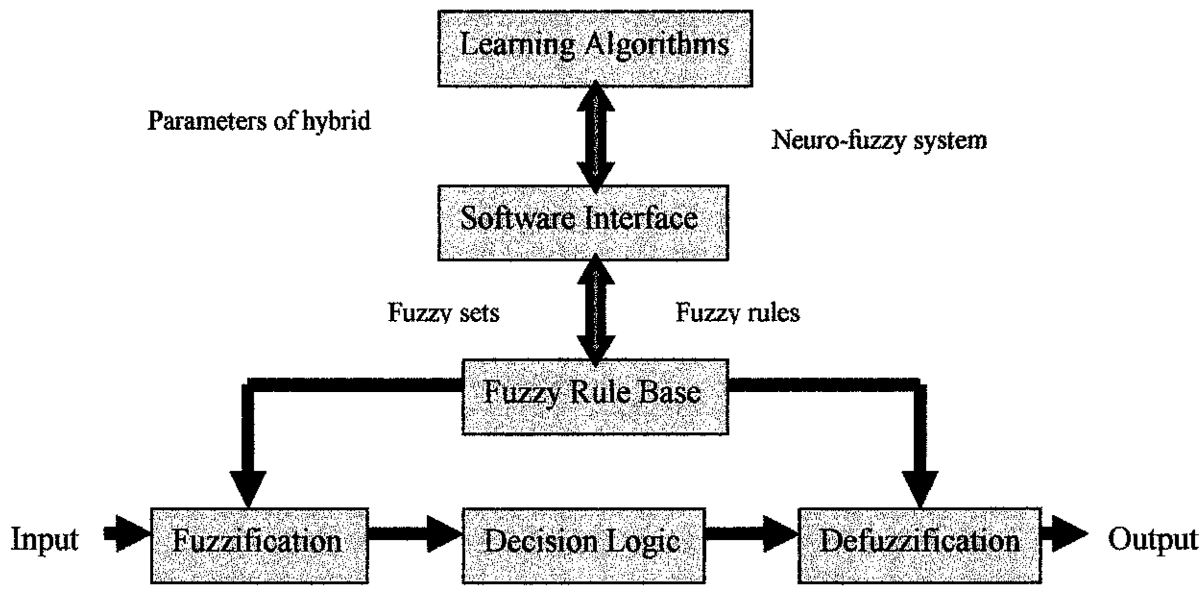


Figure 7 : Neural and Fuzzy hardware in hybrid neuro-fuzzy systems

Neuro Fuzzy systems combine the advantages of fuzzy systems, which deal with explicit knowledge which can be explained and understood whereas for Neural Network (NN) which deal with implicit knowledge which can be acquired by learning.

NN learning provides a good way to adjust the expert's knowledge and automatically generate additional fuzzy rules and MF, to meet certain specifications and reduce design tie and costs. On the other hand, Fuzzy Logic enhances the generalization capability of a NN system by providing more reliable output when extrapolation is needed beyond the limits of the training data.

3.5 Pre-processing

3.5.1 Data Collection and Analysis.

The data used for the development of the neural network was recorded by sampling the steam pressure output and water level output for 600 second. These samples were taken form the boiler model by using PID controller. Due to the large variation in magnitudes of input data, a pre-processing block is added to the neural network controller as shown in figure 8. The pre-processing block performs linear transformation of the input variables such that all inputs have similar values. To do this, each variable is treated independently, and for each input variable *Xi* we calculate its mean and variance with respect to the training set, using:

$$\overline{X_i} = \frac{1}{N}\sum_{n=1}^{N} X_i^n \tag{6}$$

$$\sigma_i^2 = \frac{1}{N-1}\sum_{n=1}^{N}(X_i^n - \overline{X_i})^2 \tag{7}$$

Where N is the number of patterns in the training set. The transformed variables are given by:

$$\overline{X_i^n} = \frac{X_i^n - \overline{X_i}}{\sigma_i} \tag{8}$$

The transformed variables will have zero mean and unit standard deviation over the transformed training set.

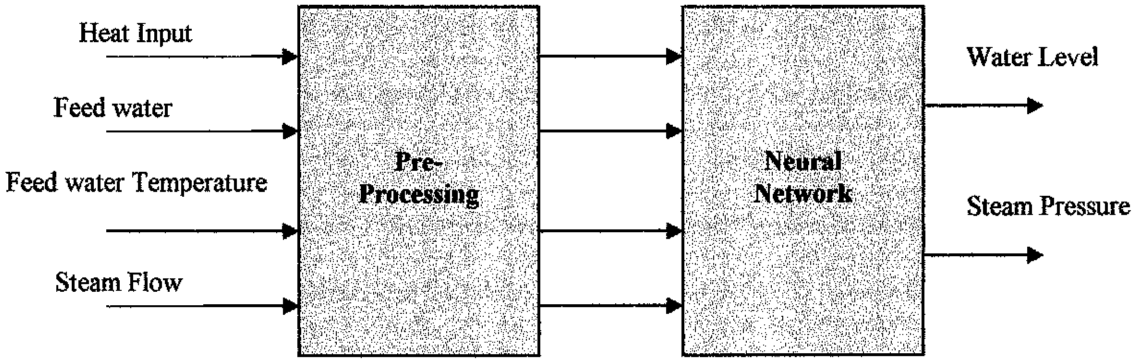


Figure 8: Neural Network controller with Pre-Processing.

CHAPTER 4

METHODOLOGY

4.1 Procedure Identification for ANFIS

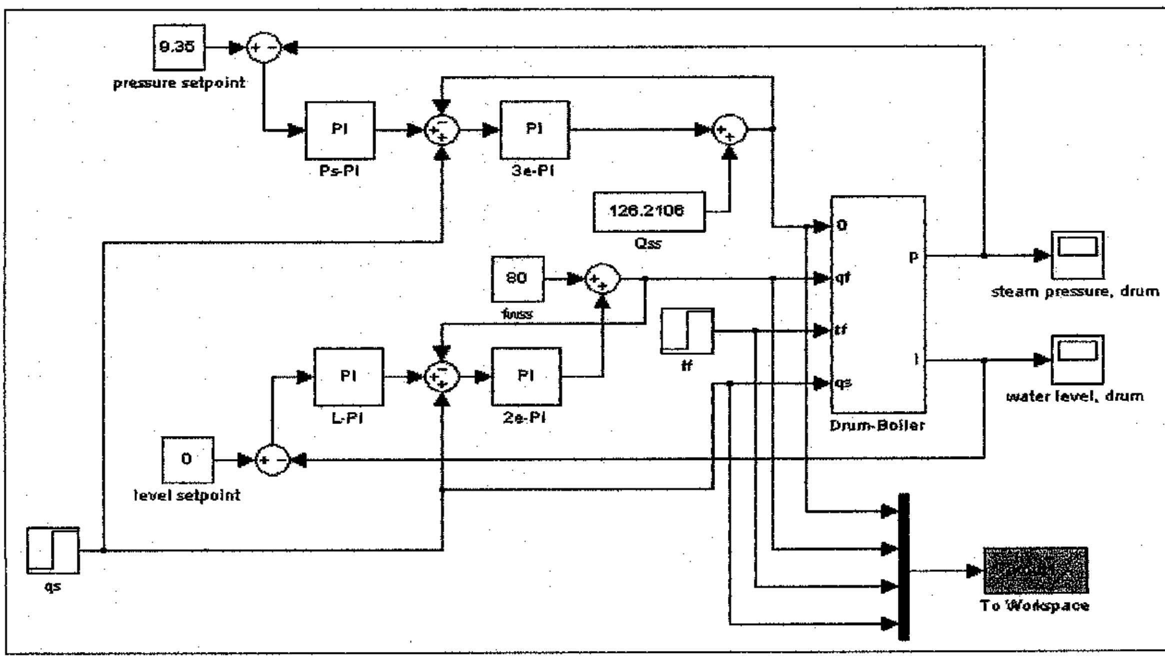


Figure 9 : A drum boiler system by using PID controller

4.1.1 Save data in workspace

The above figure showed the simulation of boiler system by using PID controller. The orange box is one of the application boxes in Matlab. It will allow the simulation inputs data to be automatically loaded into Matlab workspace. The obtained inputs data were need for ANFIS data loading as well as to ensure the data could be retrieved for later use. Please refer the obtained inputs data in ***Appendix A***.

4.1.2 Simulation of Neuro-Fuzzy based PID controller ANFIS

The first step in implementing the neuro-fuzzy controller using the Fuzzy Logic Toolbox (specifically in ANFIS Editor GUI) is to load the saved data from workspace. This was done in the MATLAB command window. Once the data was loaded, the training and checking data could be predetermined. This task was made possible by segregating the loaded data into even and odd rows. Even rows were treated as checking data while the odd rows would function as training data. The separated checking and training data were later loaded into the ANFIS Editor GUI. The syntax fir this task is as shown below

```
>>filename
>>trndata = simout (1:2:length(simout),:);
>>chkdata = simout (2:2:length(simout),:);
>>save filename trndata chkdata
>>anfisedit
```

4.1.3 Loading the data in ANFISEDIT

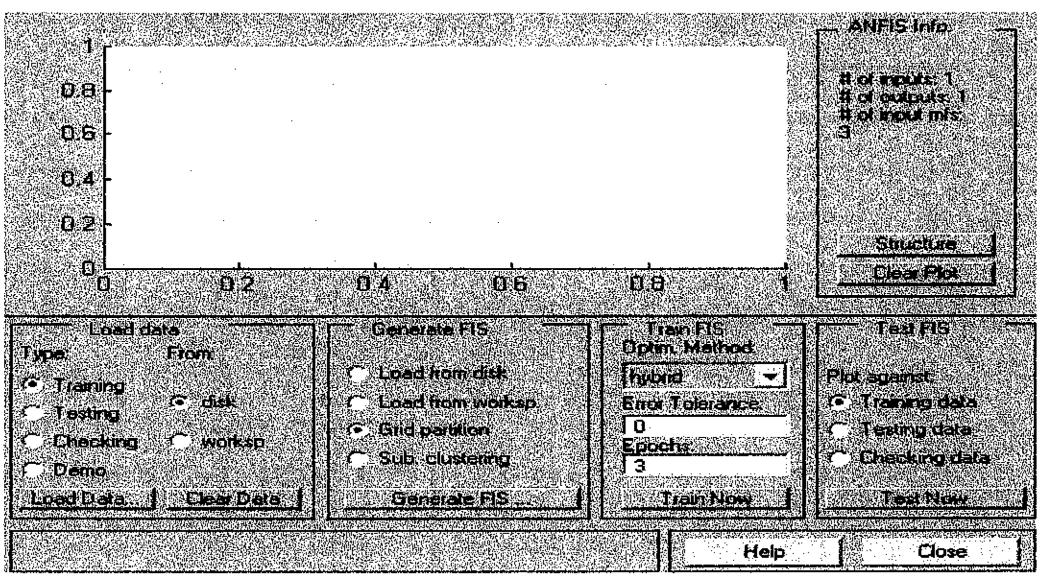


Figure 10 : The view of ANFIS Editor in Matlab

The above figure showed the view of ANFIS Editor. The writer had done several steps as below:

- Loading Data
- Generate FIS
- ANFIS training
- Testing trained FIS against training data
- Viewing the FIS structure

After performing the above steps, the writer had replaced the PID controllers with the trained NF controllers. Each steps in ANFIS Editor will be discuss in details in ***Chapter 5 (Results and Discussion)***

4.2 Procedure Identification for Neural Network.

4.2.1 Network Architecture and Training Parameters for NN Controller

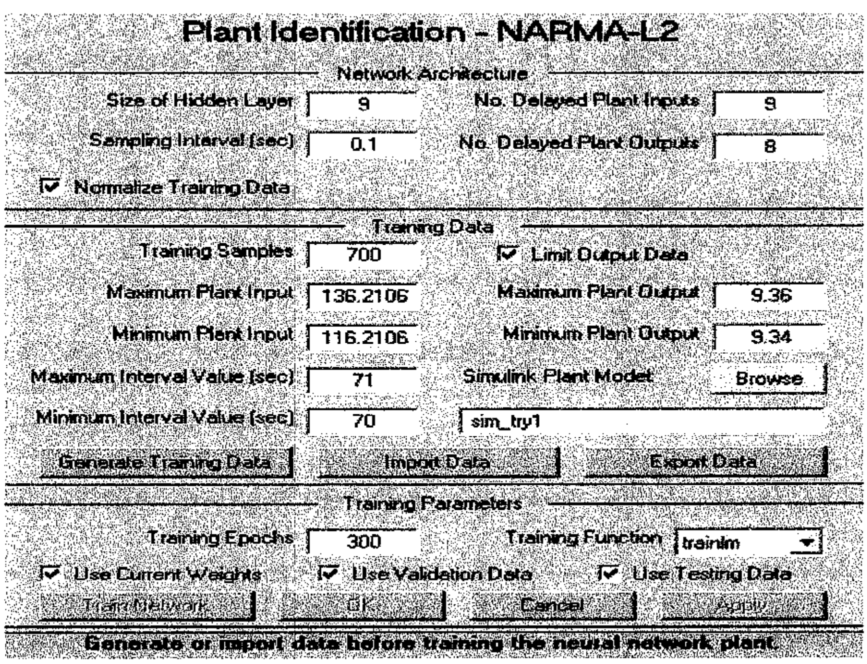


Figure 11: Plant Identification for steam pressure

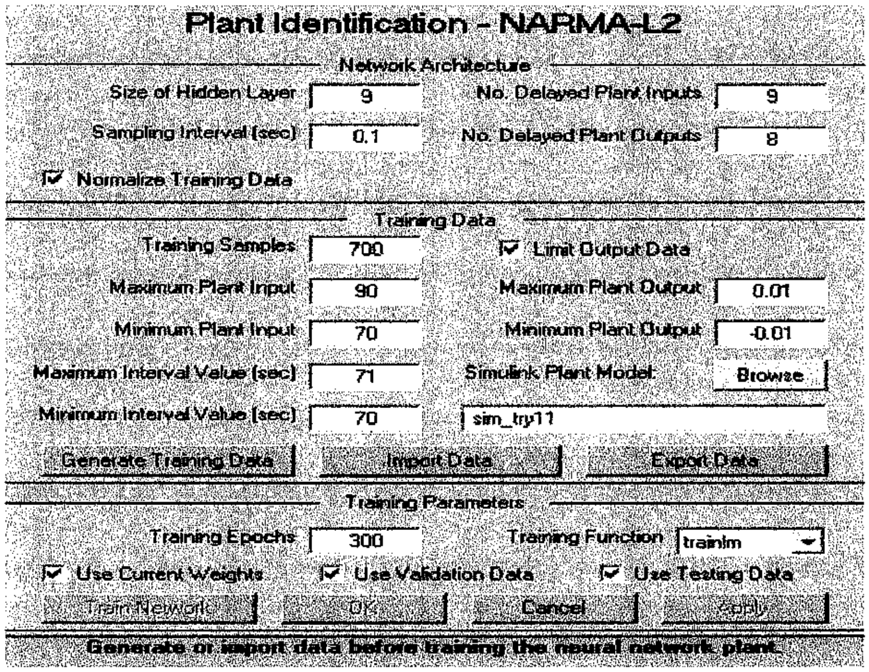


Figure 12 : Plant Identification for water level

Figure 11 and 12 showed the data of water level and steam pressure controllers by using NN. Basically, the plant identification is divided into three subtopics which are network architecture, training data and training parameters. The data in network architecture and training parameters were obtained after did some simulation by using different values. All the results were compared for determination of better NN controller. However, for the subtopic of training data, please refer ***4.2.2*** to know deeply the methodology in approaching the value.

4.2.2 The training data for water level and steam pressure controllers

- Value for inputs

Heat Input, Q (MW)	126.2106
Feed water Temperature, T_{fw} (°C)	274.2419
Steam Flow, q_s (kg/s)	80
Feed water Flow, q_{fw} (kg/s)	80

- Value for outputs

Steam Pressure (MPa)	9.35
Water Level, p (m^3)	0

By using the above information, the writer had tried to design the neural network controller for the steam pressure and water level.

- For the plant input value, the writer had taken +/- 10 MW from the set value for steam pressure whereas for water level is +/- 10kg/s. Therefore, it will allow the controller to widen its control range.

- It same goes to the plant output value which the writer had taken +/- 0.01 MPa for the steam pressure and +/-0.01 m^3 for water level.

- Previously, the training samples for PID controller was 600s, therefore the writer had set 700s for NN controller. So that, it will give a more output to make a better comparison.

- In order to determine the interval value, some basic calculation had been used as shown as below:

0.1 s = 1 sample,
Therefore, 1 s = 1/ 0.1
= 10 samples.

For 700 samples,
= (700 samples / 10 samples) * 1 s
= 70 s

- Simulink Plant Model

  The writer had designed two simulink plant model, steam pressure and water level. These plant models were saved in the plant identification under subtopic training data. The objective to design this plant model is to ensure that all the data in the NN controller will be train according to the boiler model specifications.

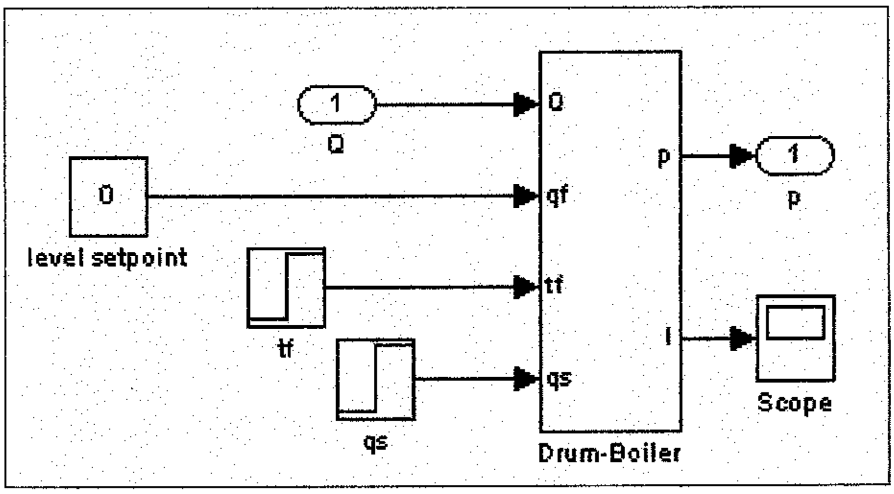


Figure 13 : The Simulink Plant Model for steam pressure

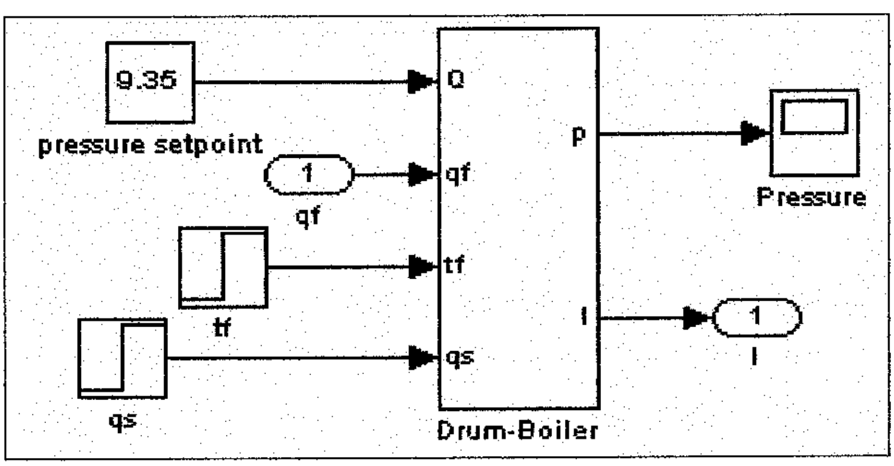


Figure 14 : The Simulink Plant Model for water level

4.2.3 To replace PID controller with NN controller

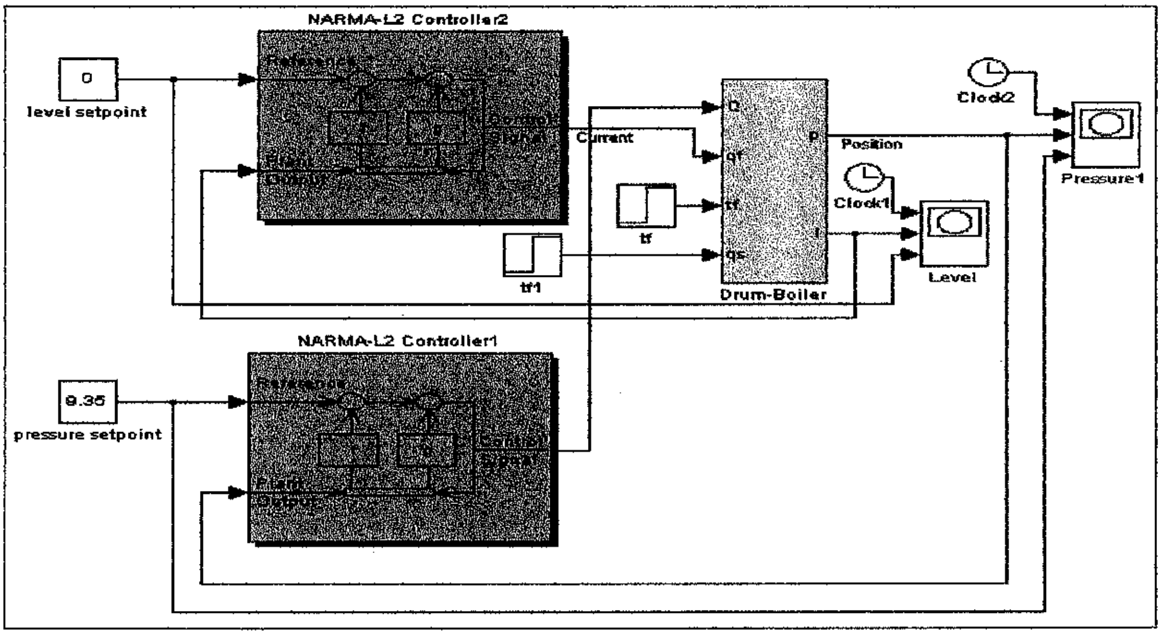


Figure 15 : A drum boiler system by using neural network controller

After the plant identifications were completed with the data that had been discussed above, the NN controller will be train by using the circuit in figure 15. The NN controllers need to be train one by one. Please refer ***Appendix B*** to know the performance of NN controller during training.

4.2.4 The procedure identification of NN controller

For Neural Network controller, the procedure identification is more to the implementation of Graphical User Interface (GUI). Therefore, the knowledge about this toolbox is highly recommended. Please refer figure 16 to know deeply about the procedures

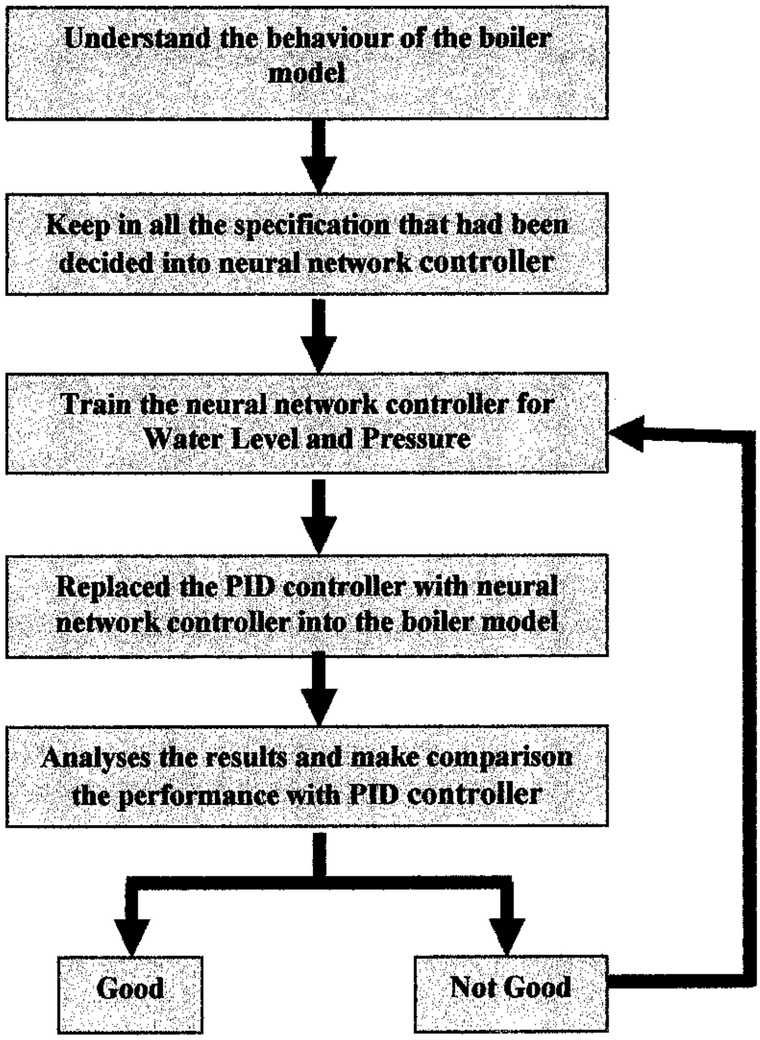


Figure 16 : Procedure Identification for Neural Network Controller

4.3 Procedure Identification for Pre-processing plus Neural Network.

4.3.1 Data Collection for pre-processing

All the inputs data is still same as in ***Appendix A***. Due to the large variation in magnitudes of inputs data in Appendix A, ***equation 6, 7 and 8*** were applied to the data. Therefore, it will perform linear transformation of the inputs variables such that all inputs have similar values.

4.3.2 Plant Identification for pre-processing plus NN controller.

Please refer subchapter ***4.2.1 and 4.2.2*** for further details about the plant identification for this controller. All the information that had been applied for this controller was still same as previous controller.

4.3.3 To replace PID controller with pre-processing plus NN controller.

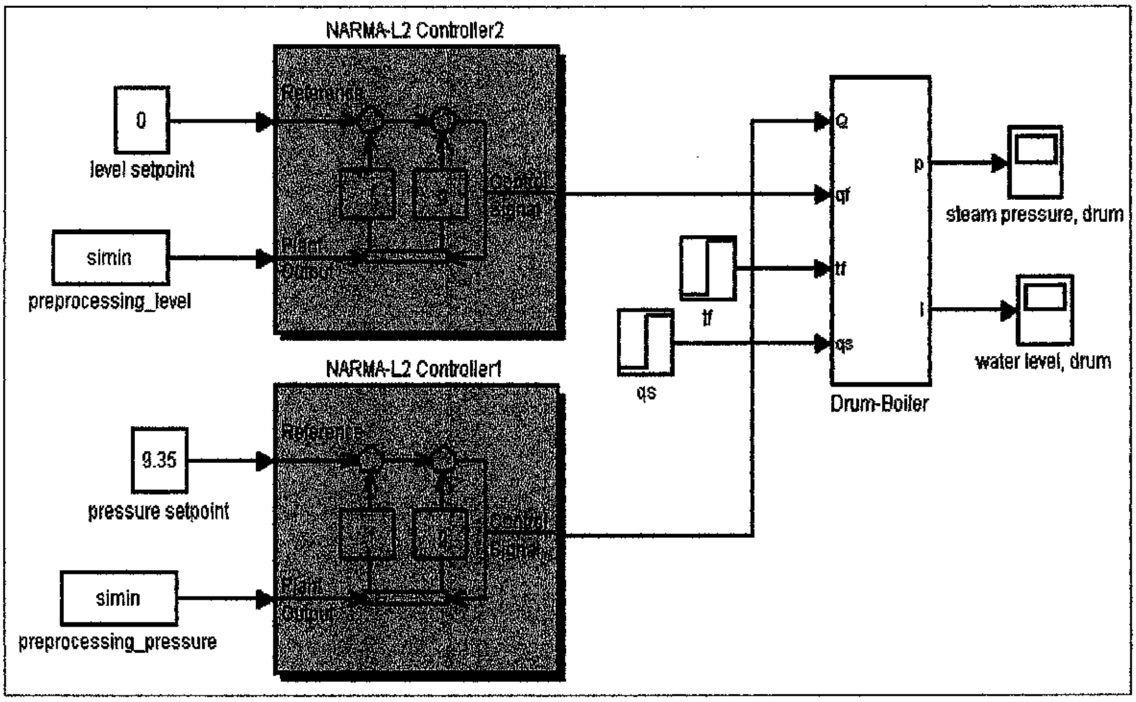


Figure 17 : Neural Network Controller had been trained by using Pre-Processing inputs

After the plant identifications were completed with the data that had been discussed above, the writer will train the NN controller by using Pre-Processing input. Figure 17 showed the circuit that the writer used to train the controller. Once the controllers were trained completely, the writer will change the circuit to become a closed loop as shown in Figure 18.

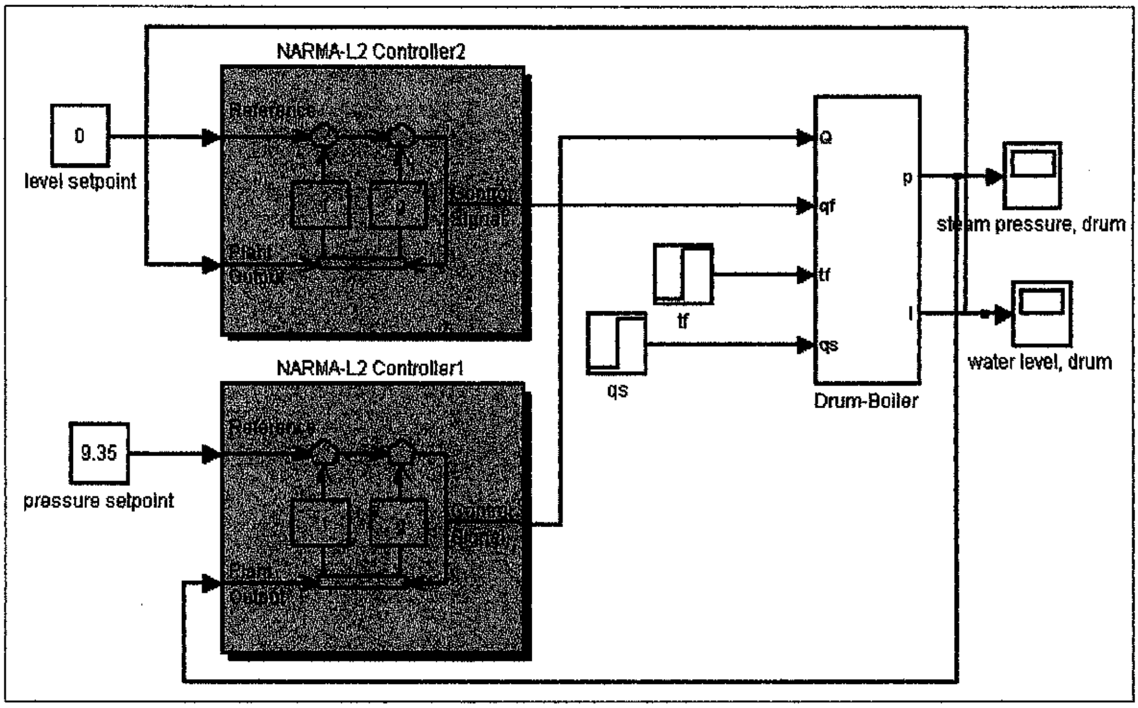


Figure 18 : The trained NN controller by using Pre-Processing inputs

4.3.4 The procedure identification of pre-processing plus NN controller

For pre-processing plus NN controller, it goes same as NN controller, the procedure identification is more to the implementation of Graphical User Interface (GUI). Therefore, the knowledge about this toolbox is highly recommended. Please refer figure 19 to know deeply about the procedures.

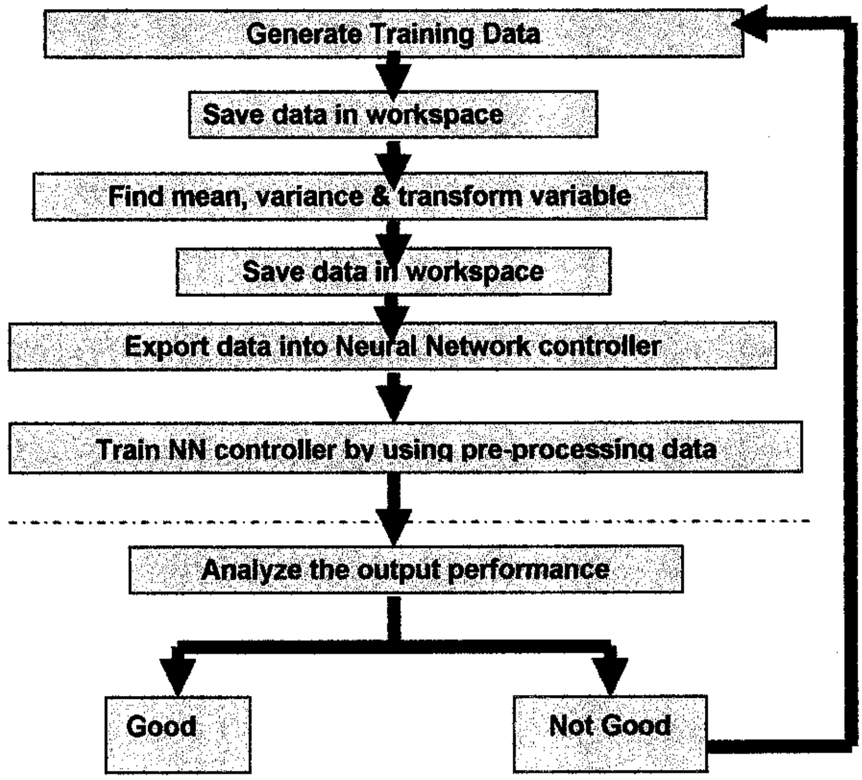


Figure 19 : Procedure Identification for Pre-Processing plus Neural Network Controller

4.4 Tools (MATLAB 6.5 Version)

4.4.1 Simulink

Simulink is software for modelling, simulating and analyzing dynamical systems. It supports linear and nonlinear systems, modelled in continuous time, sampled time, or a hybrid of the two. Simulink provides graphical user interface (GUI) for building models as block diagrams. With this interface, the desired models could be easily drawn without having to formulate the related differential equations in a language or program. Simulink includes a comprehensive block library of sinks, sources, linear and nonlinear components and connectors.

After a model has been defined, it could be simulated, either from the Simulink menus or by entering command in MATLAB Command Window. The scopes and other display blocks enable the author to see the simulation results while the simulation is running. The simulation results can be put in the MATLAB workspace for post-processing and visualization.

4.4.2 Fuzzy Logic Toolbox

The Fuzzy Logic Toolbox supports the design and analysis of fuzzy logic based systems. It supports all phases of the process, including development, research, design, simulation, and real-time implementation. Built-in graphical user interfaces (GUIs) provide an intuitive environment to guide the student through the steps of fuzzy inference system design. Functions are provided for many fuzzy logic methods, such as fuzzy clustering and adaptive neuro-fuzzy learning.

4.4.3 Neural Network Toolbox

This toolbox provides a complete set of functions and a graphical user interface for the design, implementation, visualization, and simulation of neural networks. It supports the most commonly used supervised and unsupervised network architectures and a comprehensive set of training and learning functions.

CHAPTER 5

5.0 RESULTS AND DISCUSSION

5.1 Target results

By introducing Artificial Intelligence Controller into the boiler system, the performance of the outputs (water level and steam pressure) will be more stable as shown below.

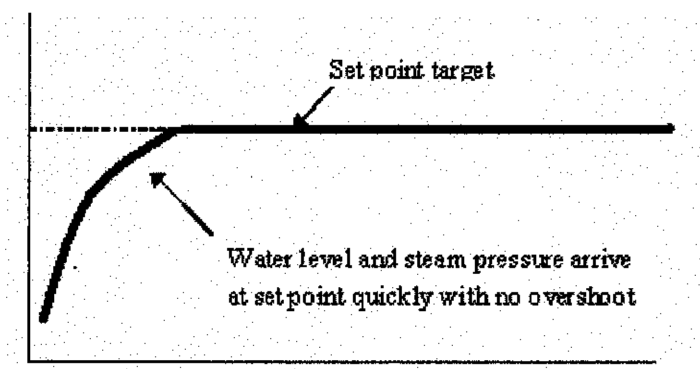


Figure 20: System response using Artificial Intelligence Controller

As we can see, the outputs approached the desired value very quickly, did not overshoot and remained stable at the target. In addition to that, by introducing neural network into the Artificial Intelligence controller, we can enhance the learning algorithm of the controller in order to boost the performance of the outputs.

5.2 Data Gathering

5.2.1 Findings by using Neuro-Fuzzy (ANFIS) plus PID Controller

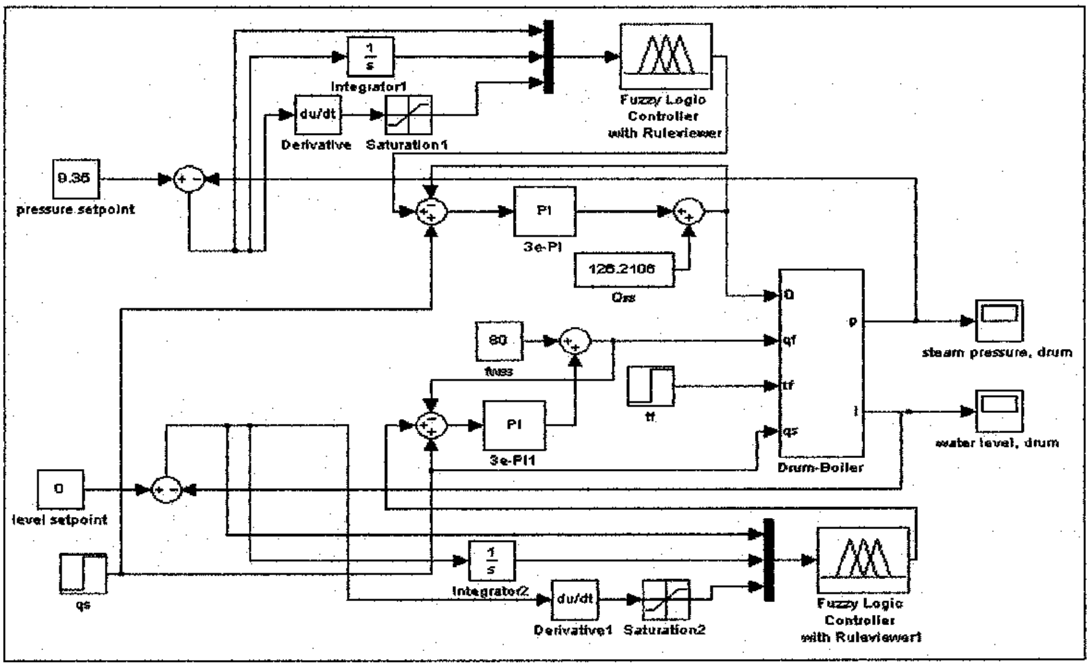


Figure 21 : A drum boiler system by using ANFIS plus PID controller

a) ANFIS learning.

The writer had taken several steps during the training process in ANFIS Editor GUI of Matlab Fuzzy Logic Toolbox. The steps are as follows:

- **Loading Data**

For training data, it only load the odd data from the workspace into ANFIS

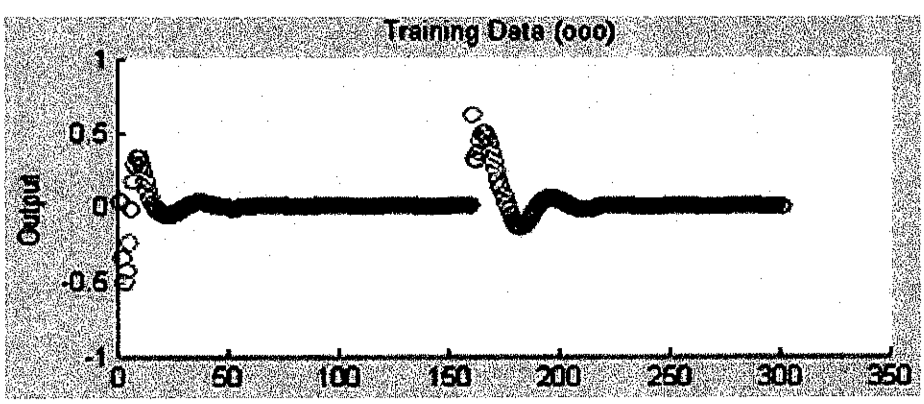


Figure 22 : Load odd data for training

For checking data, it only load the even data from the workspace into ANFIS

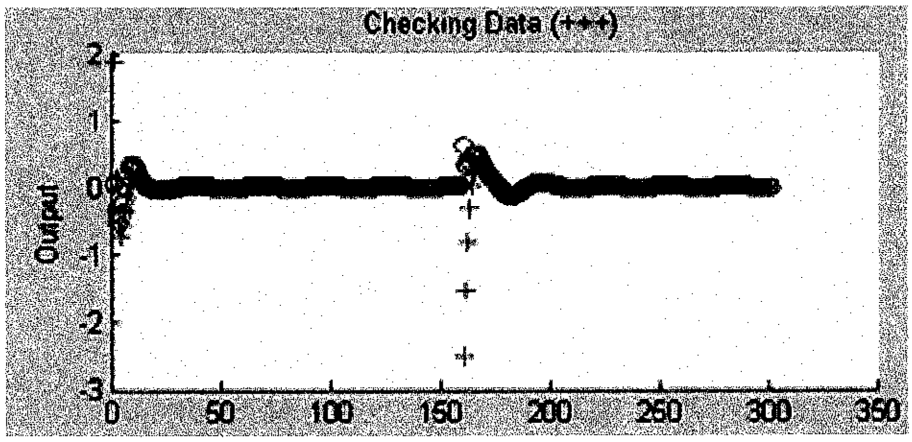


Figure 23 : Load even data for checking

Appendix A is the data loaded into the ANFIS Editor. The training data set appeared in the plot as a set of circles (Figure 22) whereas checking data set appeared in the plot as a set of plus (Figure 23). Notice that the data set index. This index indicates the row from which the input data value was obtained.

- **Generate FIS**

After all the odd and even data in workspace had been loaded into ANFIS, the next step is to initialize and generate the FIS structure. The Grid Partition is used and ANFIS will automatically generate the FIS structure.

- **Viewing the FIS structure**

Referring to figure 24, the circles represent node label. The leftmost node is the input while the rightmost node is the output node. There are four inputs altogether (error, integral, derivative). Each inputs consists of 3 membership function (low, medium, high), which results in 36 rules. The single white circles represent the weighted sum output of the system.

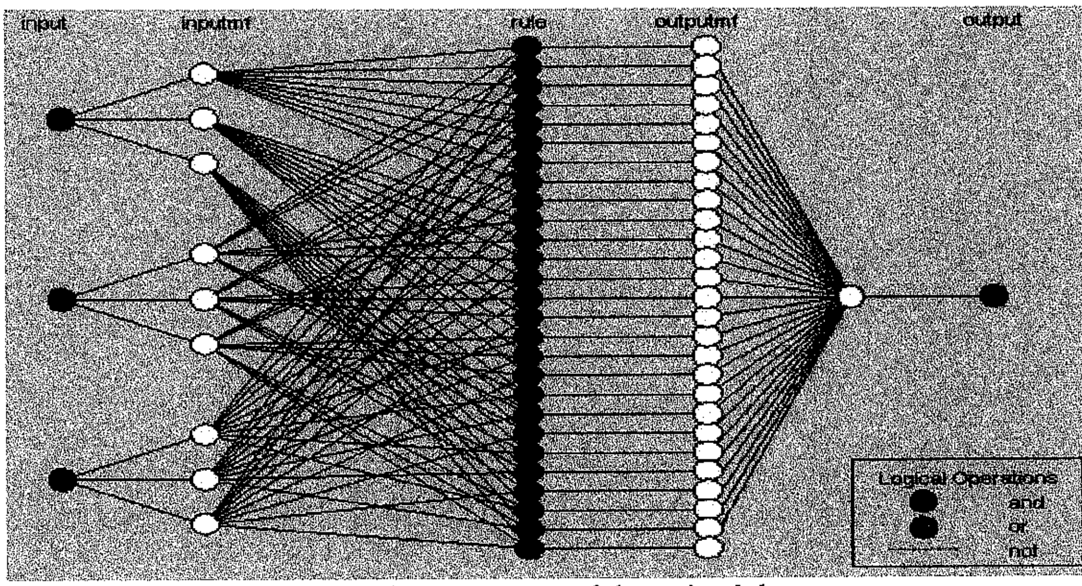


Figure 24 : Structure of the trained data

- **ANFIS Training**

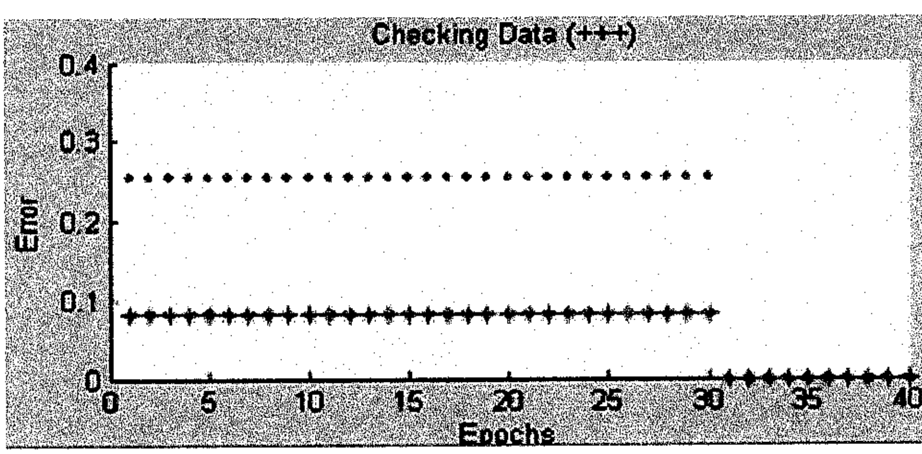


Figure 25 : Training Error (Epochs 40)

The writer used the hybrid optimisation method to train the data. The Error Tolerance is set to 0. The behaviour of the training error was did not know yet. It is used to create a training-stopping criterion, which is related to the error size. The training will stop after the training data error remains within this tolerance. The training epochs is set to 40. Figure 25 shows the training error of the trained data.

The upper line (small asterisk) represent checking error while the lower line (large asterisk) indicates the training error. As previously stated, the Error Tolerance is set to 0. Therefore, ANFIS will try to train the odd and even data to approach zero tolerance. It could be observed from the response that the checking and training data are constant as number of epochs increases. The error obtained at epoch 40 is 0.0491. The result from this training is exported to the Matlab workspace and being used for the generation of the NF controller.

- **Testing trained FIS against training data**

Figure 26 shows the testing data result of the trained FIS against the training data set. The blue circles showed the original value of odd data whereas the red asterisk showed the value of odd data that had been trained by ANFIS. Based on figure 26, we can say that the fluctuation of the data had been reduced after been trained.

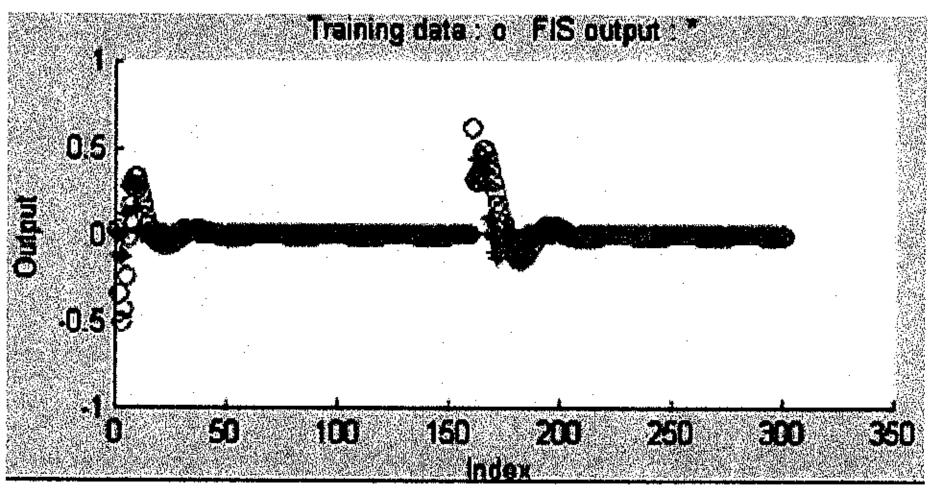


Figure 26 : Test FIS for Training Data

Figure 27, it shows the testing data result of the trained FIS against the checking data set. The blue pluses showed the original value of even data whereas the red asterisk showed the value of checking data that had been trained by ANFIS. Based on figure 27, we can say that the fluctuation of the data had been reduced after been trained.

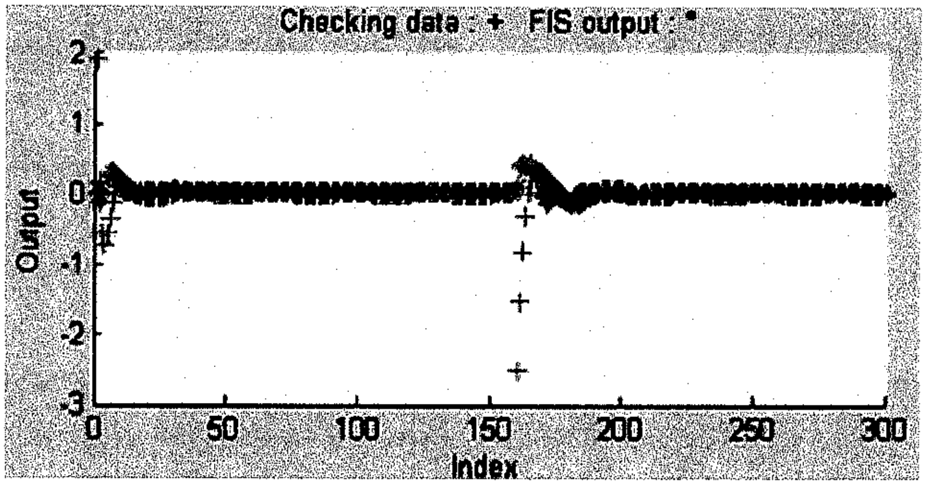


Figure 27 : Test FIS for Checking Data

b) Fuzzy Rule Viewer

The rule viewer displays a roadmap of the whole fuzzy inference process. If the input data is specified manually, the corresponding output appears automatically. The outputs are consistent if the same inputs is applied and run in the M-file. So, fuzzy rule viewer is another way of viewing the process in determining the outputs for a given set of inputs data.

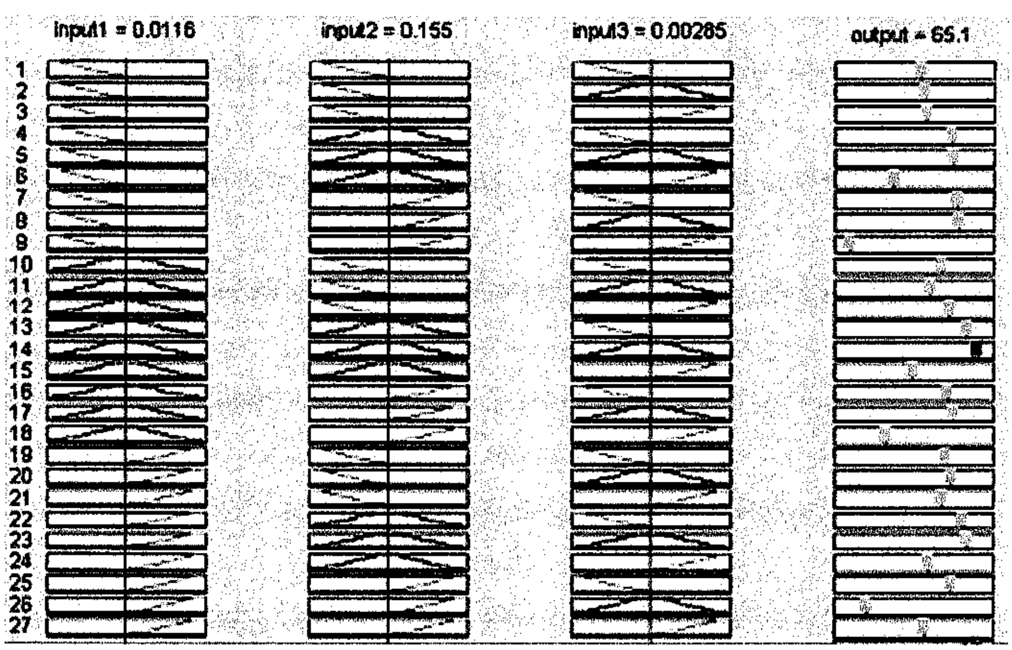


Figure 28: Rule Viewer for Steam Pressure

All the data under input1 means the data of error input, input2 means the data of integral input whereas input3 means the data of derivative input. All these three inputs will be trained in ANFIS Editor; then combination of these inputs will come out with the output of NF controller.

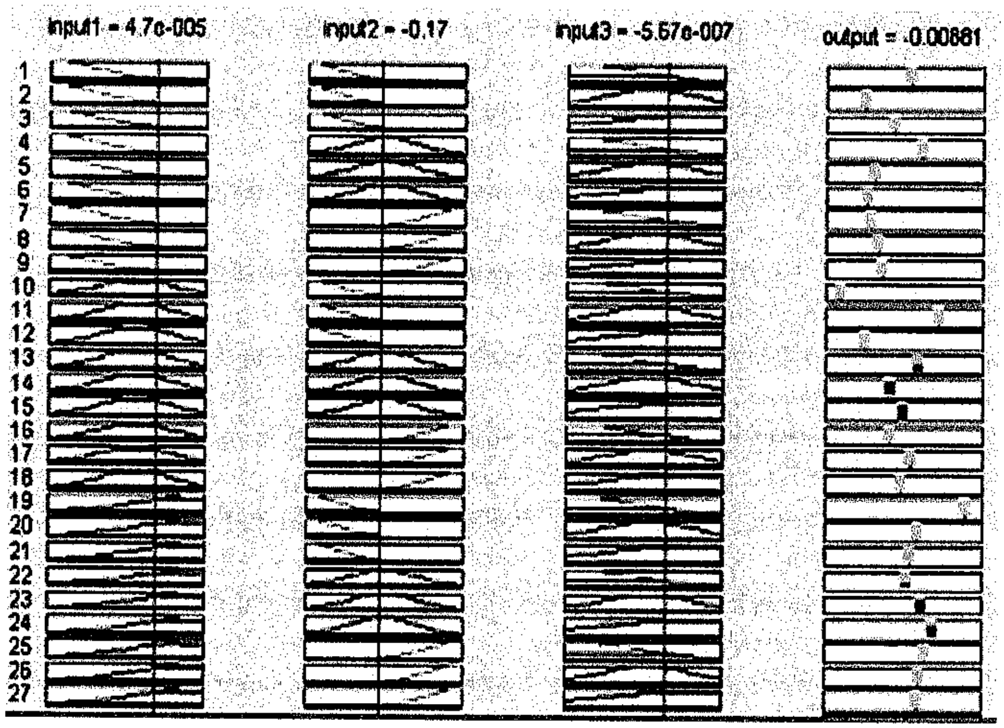


Figure 29: Rule Viewer for Water Level

C) The outputs by using NF plus PID controller

Two neuro-fuzzy controllers had been replaced two of the PID controllers, after the boiler model had been simulated by using the combination of two NF controller and two PID controllers, the performance of the outputs were shown as below.

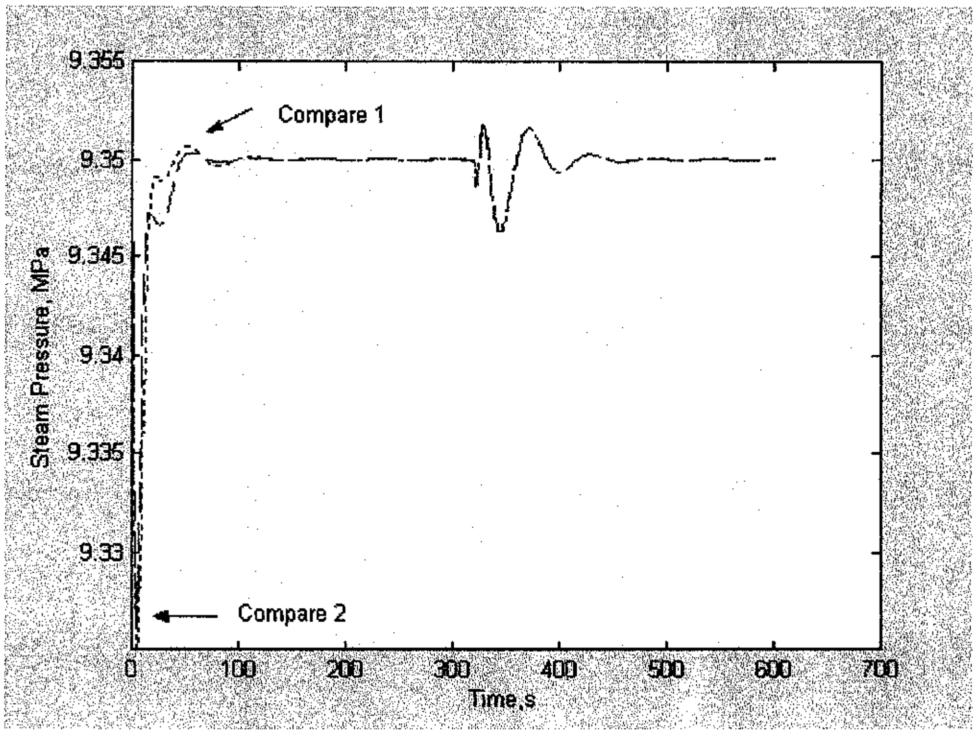


Figure 30: The performance of the steam pressure

The red dash showed the output of steam pressure by using the combination of NF and PID controller whereas the blue dot showed the performance of the output by using conventional PID controller only. Based on figure 31 and 32, we can say that the performance of the output is better using NF controller than PID controller. It is due to the peak value of NF controller is less than PID controller. Therefore, it showed that the overshoot value is less and it attempted to approach the set point value which is 9.35 MPa.

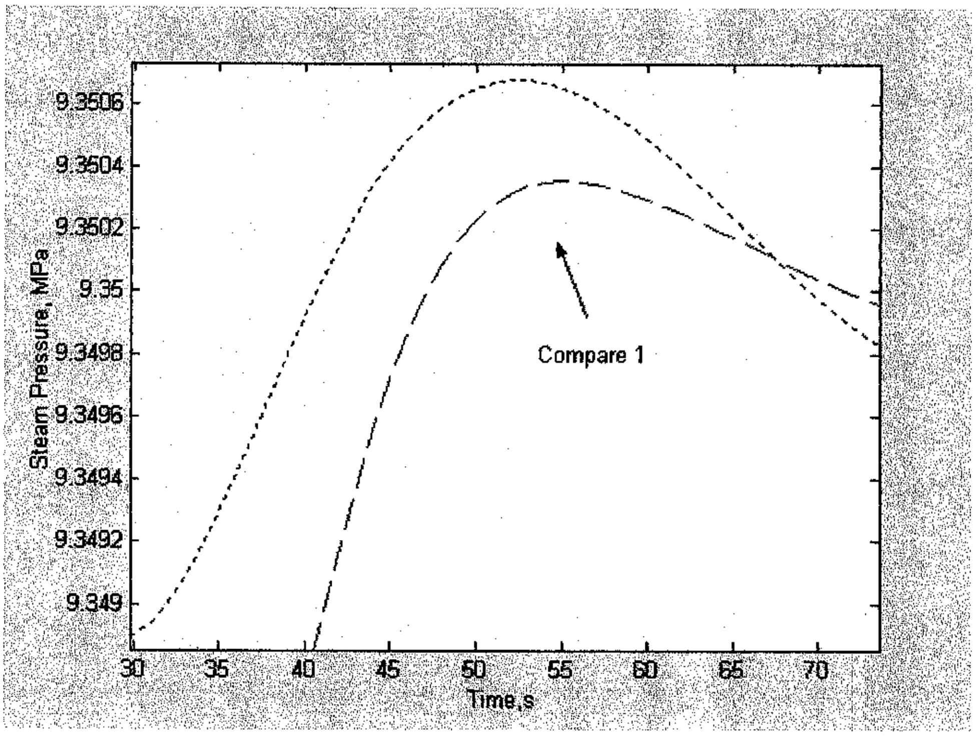


Figure 31 : The performance of the steam pressure for compare 1 that had been magnified

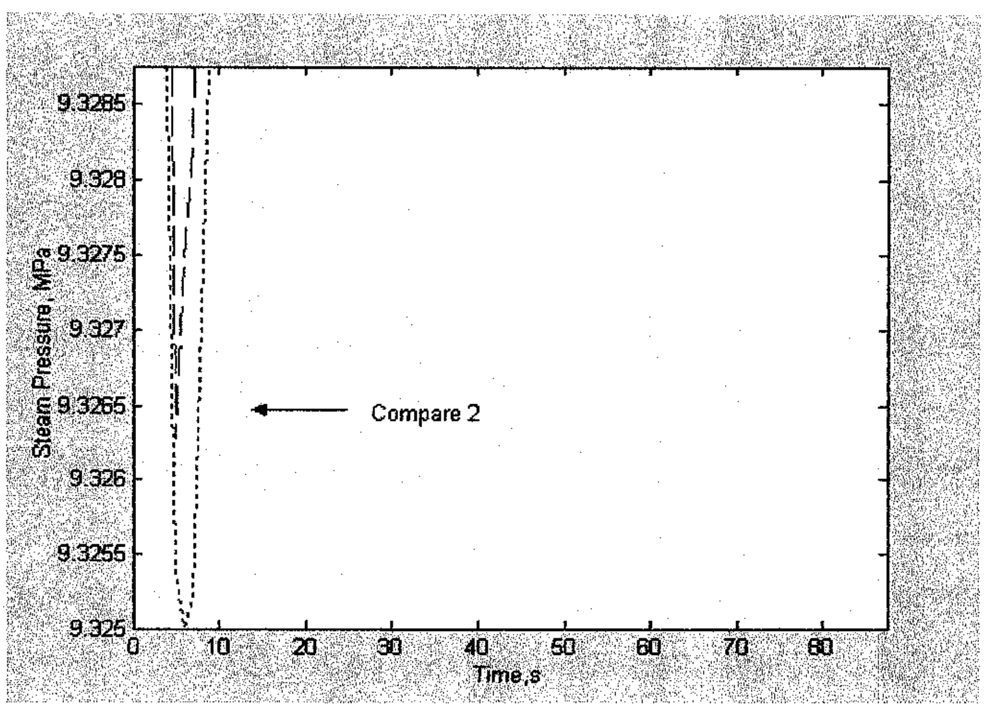


Figure 32 : The performance of the steam pressure for compare 2 that had been magnified

Figure 33 showed the comparison of output performance for water level. The red dash showed the output of water level by using the combination of NF and PID controller whereas the blue dot showed the performance of the output by using conventional PID controller only. Based on figure 33, we can say that the performance of the output is better using NF controller than PID controller. It is due to the peak value of NF controller is less than PID controller. Therefore, it showed that the overshoot value is less and it attempted to approach the set point value which is 0 m^3.

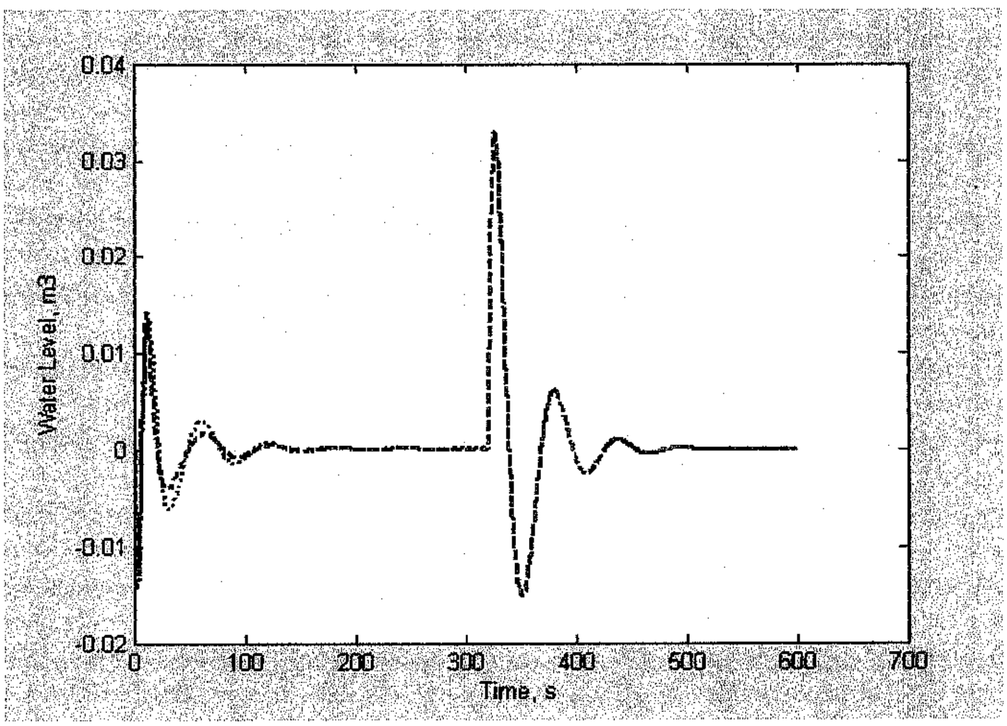


Figure 33 : The performance of the water level.

5.2.2 Findings by using Neural Network(NN) Controller

Two NN controllers had been used in this boiler model which is one controller was placed at water level loop and another one was placed at steam pressure loop. After the boiler model had been simulated by using these two NN controllers, the performance of the outputs were shown as below. Please refer *5.3.2* to know further details about the findings.

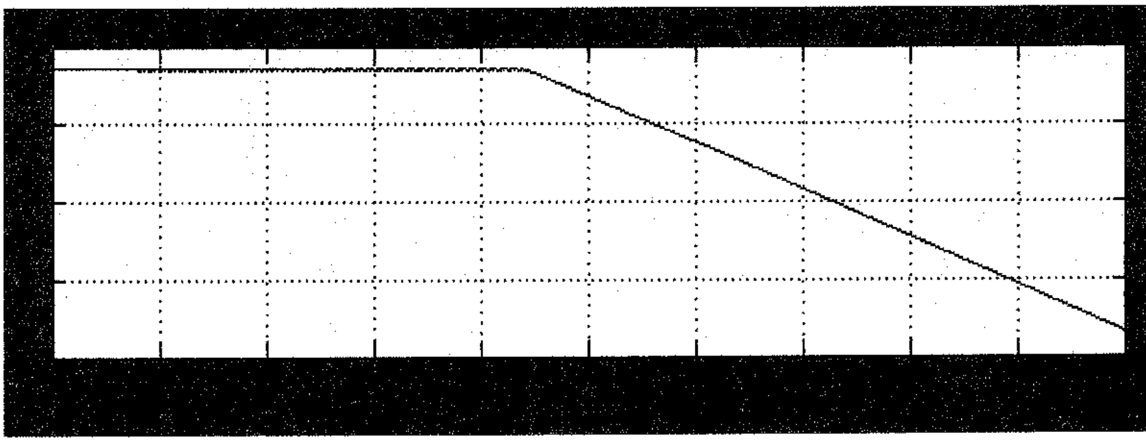


Figure 34 : The output of steam pressure by using NN controller.

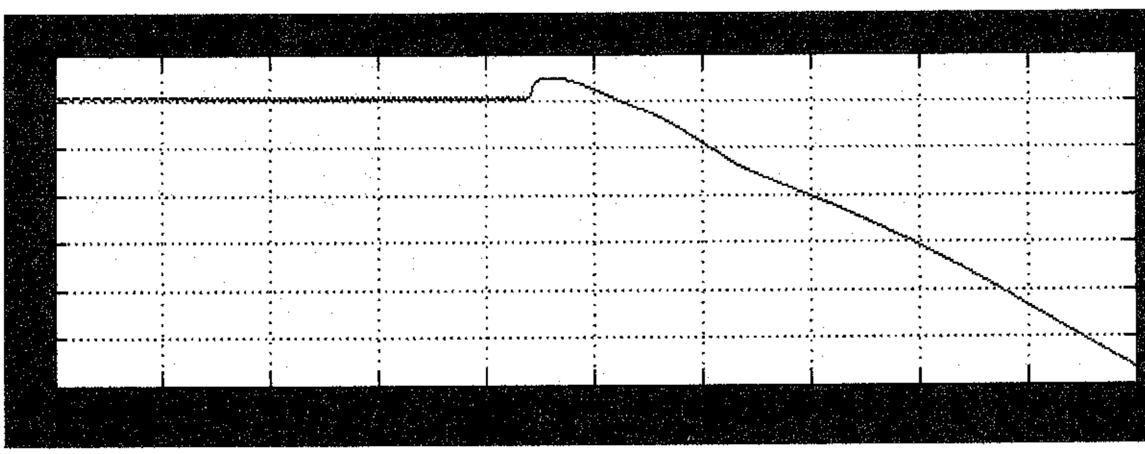


Figure 35 : The performance of water level by using NN controller

5.2.3 Findings by using Pre-Processing plus Neural Network Controller

Table 2 : The value of Pre-Processing for boiler inputs

Boiler Inputs	Mean	Variance	Pre-processing
Heat Input	133.5052	70.03814	8.368879
Feed water Flow	83.91505	170.9358	13.0742
Feed water Temperature	274.24	0	0
Steam Flow	84.67554	24.93622	4.99362

Based on table 2, the writer had used the entire data as inputs for pre-processing. Then, the writer had combined it with the NN controller. All the methodology to perform this controller is still same as previous NN controller. The only different between these two is this NN controller is using pre-processing data as inputs to train the NN controller whereas the NN controller in subchapter 5.2.2 is using the current outputs data as inputs to train the NN controller. Figure 36 and 37 showed the performance of the outputs by using pre-processing input in this controller. Please refer ***5.3.2*** to know further details about the findings.

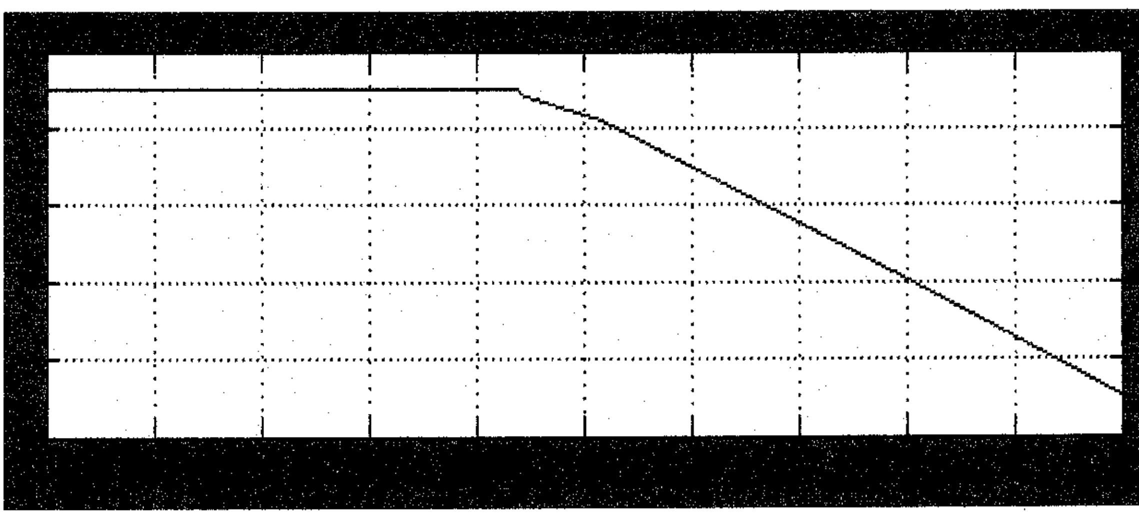


Figure 36: The performance of steam pressure by using Pre-Processing

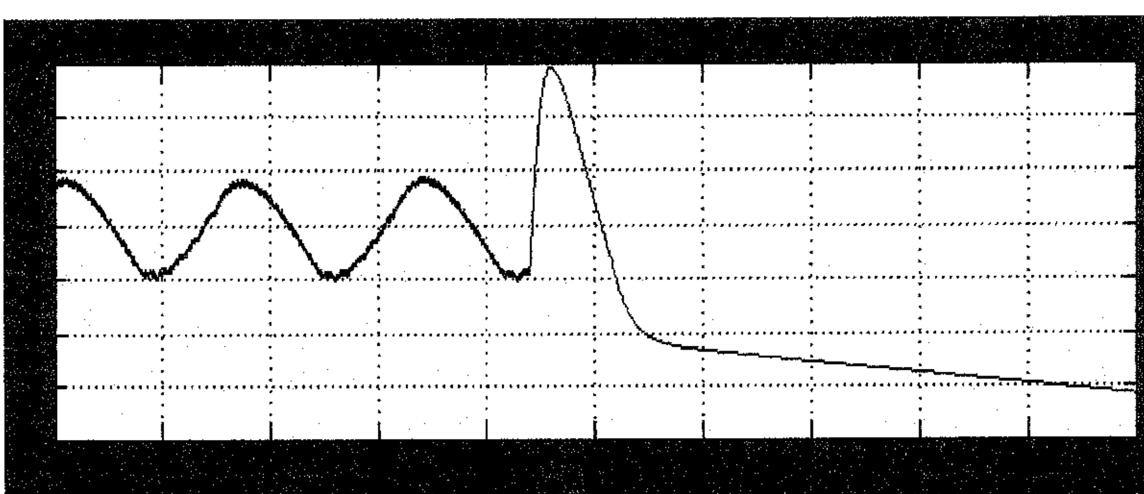


Figure 37 : The performance of water level by using Pre-Processing

5.3 Discussion of Results

5.3.1 From Theory

The Artificial Intelligence approach makes it easier to conceptualize and implement control systems. The process is reduced to a set of visualizable steps. This is a very important point. Actually implementing a control system, even a simple control system, is more difficult than it appears. Unexpected aberrations and physical anomalies inevitably occur.

Experienced, professional digital control engineers using conventional control might know how to proceed to fine tune a system. But, it can be difficult for us just plain folks. Fuzzy logic control makes it easier to visualize and set up a system.

In reading about AI control applications in industry, one of the significant points that stand out is: AI is used because it shortens the time for engineering development. AI enables engineers to configure systems quickly without extensive experimentation and to make use of information from expert human operators who have been performing the task manually.

5.3.2 From Results

Referring to the outputs of conventional PID controller, we can see that the fluctuation was happened at the starting point as the boiler start to generate. Fluctuation also was occurred at time 320 second. This was happened due to the value of steam flow, qs increased.

In this report, the writer had compared the outputs performance by using

1. NF plus PID controller
2. NN controller
3. Pre-processing plus NN controller.

Based on the above three AI controllers, only the first controller can perform better if there is have any disturbance from the steam flow. Steam flow was considered as third element which means uncontrollable and depends on the number of electricity that the users need. However, for the second and third controllers, it can only perform better in ideal case, which means without any disturbance from the steam flow. The condition of ideal case was approved by doing some research and literature review.

Table 3 : The overshoot value for different types of controllers during step input at 320s

Types of controllers	Steam Pressure (Set Point = 0)	Water Level (Set Point = 9.35)
PID	2.5%	1.38%
Neural Network	infinite	infinite
Pre-Processing	infinite	infinite
Neuro-Fuzzy	2.37%	1.43%

The above table showed the overshoot value for different types of controllers. Referring to the figure, we can see that the NF controller gave the best performance compares to the other AI controllers. It is due to the overshoot value for steam pressure is only 2.37% whereas for water level is 1.43%.

For NF controller, it performs a slightly better for steam pressure and slightly worst for water level as we compare with the performance of conventional PID controller. Theoretically, neuro-fuzzy supposed will perform far better than PID controller. However, in this case, the writer use a combination of neuro-fuzzy and PID controller to control the drum boiler. The reason beyond that is due to the Matlab Simulink will have some error as the writer replaced the entire PID controller with NF controller. To design NF controller, we need to put the error input, integral input and also derivative input. Either for steam pressure loop or water level loop, both are using two PI controllers, therefore both PI controller can not perform as the writer gave the same error input, integral input and also derivative input. Because of time limitation, the writer suggests to do more research regarding the suitable value of error input, integral input and also derivative input to put in the second PI controller

For Pre-processing plus NN controller, it was only performing better during ideal case. Therefore, the writer suggests that we need to review the formula of equation 6, 7 and 8 in order to enhance the performance of this controller.

CHAPTER 6

6.0 CONCLUSION & RECOMMENDATIONS

6.1 Conclusion

As a conclusion, MATLAB 6.5 version is very helpful for this final year project. Four major tools boxes that the student had used were Simulink, Neural Network Toolbox, Fuzzy Logic Toolbox and ANFIS Editor GUI. Simulink is very important due to design the drum boiler model and at the same time come out with the results of PID controller, for Neural Network Toolbox, Fuzzy Logic Toolbox and ANFIS Editor GUI, the student need those three toolboxes in order to design the Artificial Intelligence Controller,

From literatures reviewed, it seems that to integrate different soft computing paradigms such as fuzzy logic, artificial neural networks, neuro-fuzzy and et cetera can develop hybrid intelligent systems that provide more flexibility to exploit tolerance and uncertainty of real life situation. Combining the advantages of fuzzy logic (use the expert knowledge), the advantages of genetic algorithm (function optimization) and the advantages of neural networks (learning and adaptability) can develop a robust adaptive system.

Therefore, in determining the best AI controller, it actually depends on several factors such as the design of the drum boiler, the environment effects and also the value of inputs and outputs. Some controller might be good in several factors but weak in others. Thus, in optimizing the potential of an alternative AI controller to perform good outputs for this drum boiler, NN controller is recommended. This will lead to nearly achieve the set point for the steam pressure and also water level.

6.2 Recommendations

While conducting this project, the writer has encountered with several lesson learnt to be shared and recommended. The following steps are recommended to be followed: -

1. Set purpose of the project and what types of justification to be studied.
2. Prepare the alternative AI controller accordingly by knowing the drum boiler details such as chemical composition, PID controller, and also feedback and feed forwad loop
3. Learn some knowledge regarding Artificial Intelligence from Information Technology lecturer specifically Mr. Jale Ahmad.

The advance literature research should be continually updated because new information may be gathered while executing other stages of the project. More information on the recent application of AI controller in power plant is needed to provide guidelines to proceed in designing an alternative AI controller, which provide better performance than the conventional PID controller.

In the future, as continuation, it is highly recommended that this project is being pursuit by the integration of Chemical Engineering student together with Mechanical Engineering student to cover the field of study on the impressed Artificial Intelligence Controller.

Future considerations might also include testing the controllers in real life environment therefore, Universiti Teknologi PETRONAS is also recommended to have a pilot power plant set-up that consists of boiler instalment as well. This will enable student to conduct a real AI controller application using the circuit introduced in figure 1.

REFERENCES

[1] Fawnizu A.H, Rees (2003). Overcoming the shrink and swell effect n water level control strategy on industrial drum boiler, ICSE 2003.

[2] Fawnizu A.H, Rees , N.W. (2001) *Drum Water Level Control: A Study by Simulation.* MEngSc thesis submitted to UNSW, Australia.

[3] Fuzzy Logic Toolbox for use with MATLAB, The MATH WORKS Inc.,Version 2, 1995.

[4] Zadeh, L.A (1973). Outline of a new approach to the analysis of complex system and decision processes. IEEE Trans. SMC-3, 28.

[5] Fuzzy Logic Toolbox for use with MATLAB, The MATH WORKS Inc.,Version 2, 1995.

[6] The student Edition of MATLAB, The language of technical Computing, The MATH WORKS Inc., Version 5, 1997.

[7] A first course in fuzzy logic, Hung T.Nguyen, Chapman & Hall/ CRC, 2nd Edition, 2000.

[8] Artificial Neural Networks, Robert J. Schalhoff, Mc Graw Hill International Editions, 1997

[9] Foundations of Neuro-Fuzzy Systems, Wiley, Detlef Netwek, 1997.

[10] Neural & Adaptive Systems (Fundamentals through Simulations), Jose' C. Principle,Wiley, 1999.

[11] Dr. Harold F. Chambers, Jr., 2001, Big Bend Power Station Neural Network-Sootblower Optimization. <http://www.lanl.gov/projects/ppii/factsheets/neural/neural_demo.html>, (last updated 3 May 2004, 7.00 pm)

[12] Al-Duwaish, 2002, Use of Artificial Neural Networks Process Analyzers: A Case Study. < http://www.dice.ucl.ac.be/Proceedings/esann/esannpdf/es2002-30.pdf>, (last updated 3 May 2004, 7.10 pm)

[13] Raul Garduno-Ramirez, 2000, Wide Range Operation of a Power Unit via Feedforward Fuzzy Control.

<http://labs.ee.psu.edu/labs/powerlab/papers/energy_00154D.pdf> (last updated 3 May 2004, 7.30 pm)

[14] Jan Jantzen, 1998, Tuning of Fuzzy PID Controllers. <http://www.iau.dtu.dk/~jj/pubs/fpid.pdf > (last updated 3 May 2004, 9.15 pm)

[15] M. Brown & C. Harris, *Neuro Fuzzy Adaptive Modeling and Control*, Prentice Hall Inc., New York, (1994)

APPENDIX A

These are the inputs data from the simulation by using PID controller. These data had been trained by using ANFIS. Therefore, the performance of NF controller could be analysed.

0	0	0	0
0.0089087	0.0046562	0.0089087	13.363
0.017235	0.01793	0.0083265	27.456
0.021363	0.037346	0.0041283	36.751
0.02462	0.060435	0.0032562	45.481
0.024996	0.085275	0.00037621	50.477
0.024794	0.11019	-0.00020154	54.674
0.022898	0.134	-0.0018962	56.292
0.020858	0.15584	-0.0020402	57.354
0.018026	0.17523	-0.0028315	56.861
0.015405	0.19188	-0.0026212	56.174
0.012583	0.20581	-0.002822	54.714
0.010172	0.21712	-0.0024117	53.361
0.0078777	0.22608	-0.0022939	51.751
0.0060477	0.23299	-0.0018350.424	
0.0044553	0.23819	-0.0015923	49.124
0.0032786	0.24202	-0.0011767	48.161
0.0023379	0.24479	-0.00094072	47.34
0.0017135	0.24679	-0.00062441	46.824
0.0012658	0.24826	-0.00044766	46.461
0.0010249	0.24939	-0.0002409	46.328
0.00088763	0.25034	-0.00013728	46.306
0.00086257	0.2512	-2.5063e-005	46.429
0.00087854	0.25207	1.5977e-005	46.608
0.00093815	0.25297	5.9605e-005	46.855
0.0009963	0.25394	5.8152e-005	47.111
0.0010562	0.25496	5.9945e-005	47.381
0.0010924	0.25603	3.6203e-005	47.625
0.0011105	0.25713	1.8071e-005	47.849
0.0010982	0.25824	-1.2278e-005	48.03
0.001063	0.25931	-3.5261e-005	48.175
0.0010006	0.26034	-6.2344e-005	48.273
0.00091905	0.2613	-8.159e-005	48.331
0.00081836	0.26217	-0.00010068	48.345
0.00070577	0.26293	-0.0001126	48.324
0.00058311	0.26357	-0.00012265	48.267
0.00045624	0.26409	-0.00012687	48.181
0.00032726	0.26448	-0.00012898	48.07
0.00020044	0.26474	-0.00012682	47.939
7.7449e-005	0.26488	-0.00012299	47.79
-3.8881e-005	0.2649	-0.00011633	47.63
-0.00014752	0.2648	-0.00010864	47.46
-0.00024683	0.26461	-9.9307e-005	47.284
-0.00033636	0.26431	-8.9525e-005	47.105
-0.00041531	0.26394	-7.8953e-005	46.926

-0.00048368	0.26349	-6.8371e-005	46.749
-0.00054125	0.26298	-5.7572e-005	46.576
-0.0005883	0.26241	-4.7049e-005	46.408
-0.00062497	0.26181	-3.6667e-005	46.247
-0.0006517	0.26117	-2.6731e-005	46.094
-0.00066885	0.26051	-1.7151e-005	45.951
-0.00067696	0.25984	-8.1139e-006	45.819
-0.00067652	0.25917	4.3851e-007	45.697
-0.00066814	0.2585	8.3893e-006	45.588
-0.00065236	0.25784	1.5774e-005	45.492
-0.00062985	0.2572	2.2516e-005	45.408
-0.00060121	0.25658	2.8637e-005	45.338
-0.00056712	0.256	3.4086e-005	45.28
-0.00052825	0.25545	3.8876e-005	45.237
-0.00048527	0.25495	4.2977e-005	45.206
-0.00043887	0.25449	4.6398e-005	45.188
-0.00038974	0.25408	4.9128e-005	45.183
-0.00033856	0.25371	5.118e-005	45.19
-0.00028601	0.2534	5.2556e-005	45.208
-0.00023273	0.25314	5.328e-005	45.236
-0.00017936	0.25294	5.3368e-005	45.274
-0.00012651	0.25279	5.2852e-005	45.321
-7.4747e-005	0.25269	5.1761e-005	45.376
-2.4611e-005	0.25264	5.0136e-005	45.438
2.3408e-005	0.25264	4.8018e-005	45.505
6.8863e-005	0.25268	4.5456e-005	45.578
0.00011136	0.25277	4.2497e-005	45.654
0.00015055	0.2529	3.9194e-005	45.733
0.00018616	0.25307	3.5602e-005	45.813
0.00021793	0.25327	3.1775e-005	45.894
0.0002457	0.25351	2.7769e-005	45.975
0.00026934	0.25376	2.3637e-005	46.055
0.00028877	0.25404	1.9434e-005	46.133
0.00030398	0.25434	1.5211e-005	46.207
0.000315	0.25465	1.1017e-005	46.279
0.0003219	0.25496	6.8995e-006	46.346
0.0003248	0.25529	2.9015e-006	46.408
0.00032386	0.25561	-9.3698e-007	46.465
0.00031928	0.25593	-4.5801e-006	46.516
0.00031129	0.25624	-7.9956e-006	46.562
0.00030013	0.25655	-1.1156e-005	46.601
0.00028609	0.25684	-1.4037e-005	46.634
0.00026947	0.25712	-1.662e-005	46.661
0.00025058	0.25738	-1.8891e-005	46.681
0.00022975	0.25762	-2.0837e-005	46.695
0.00020729	0.25784	-2.2454e-005	46.702
0.00018355	0.25803	-2.3738e-005	46.704
0.00015886	0.2582	-2.4691e-005	46.7
0.00013355	0.25835	-2.5318e-005	46.691
0.00010792	0.25847	-2.5627e-005	46.676
8.2286e-005	0.25856	-2.5631e-005	46.657
5.6943e-005	0.25863	-2.5343e-005	46.634
3.2161e-005	0.25868	-2.4782e-005	46.607
8.1967e-006	0.2587	-2.3965e-005	46.577
-1.4718e-005	0.25869	-2.2914e-005	46.544
-3.6371e-005	0.25867	-2.1654e-005	46.509

APPENDIX B

1) The train data for steam pressure by using NN controller

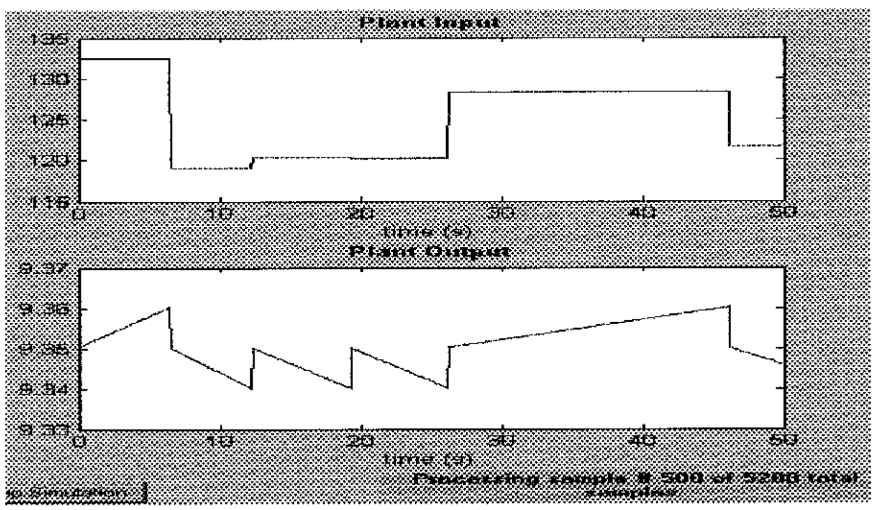


Figure 1: Performance of Heat Input Controller during generate training data

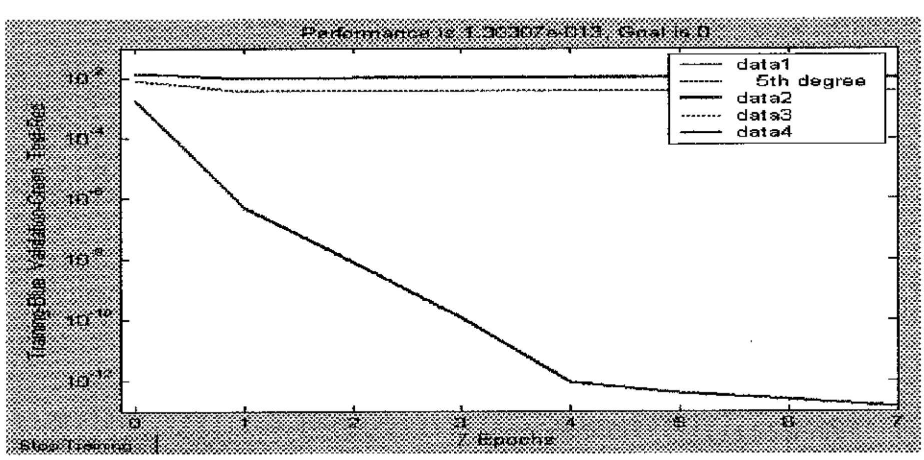


Figure 2 : Performance of the epochs after Train Network

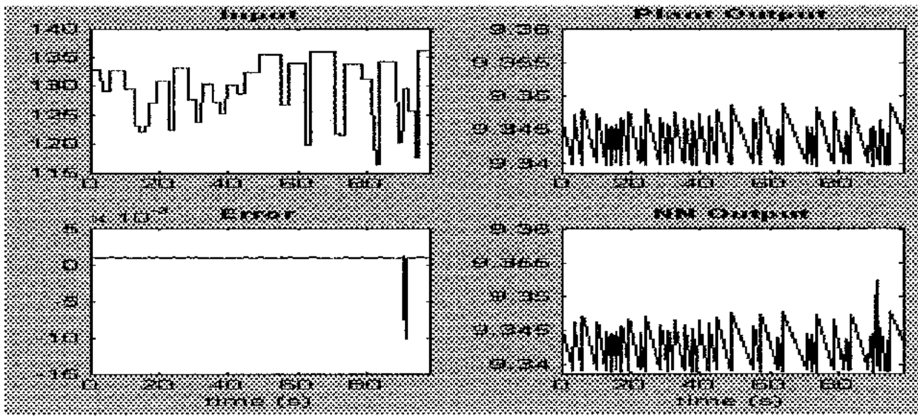


Figure 3 : Testing Data after Train Network

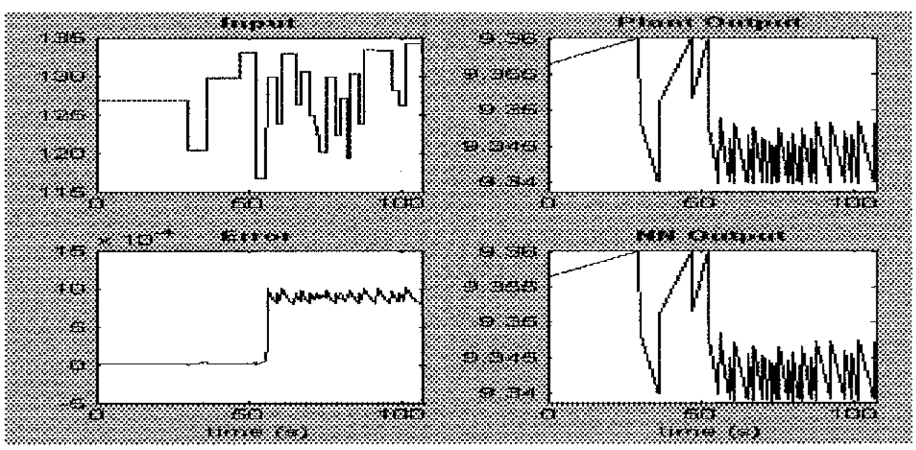


Figure 4 : Validation Data after Train Network

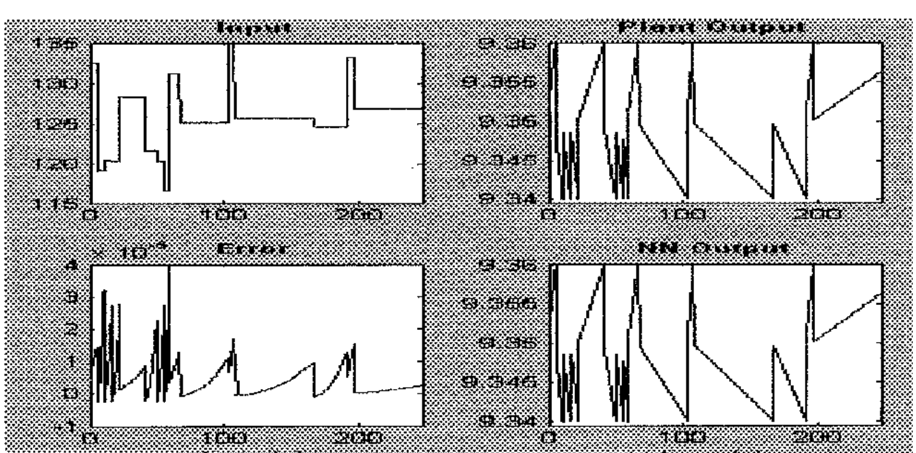


Figure 5 : Training Data after Train Network

2) The train data for water level by using NN controller

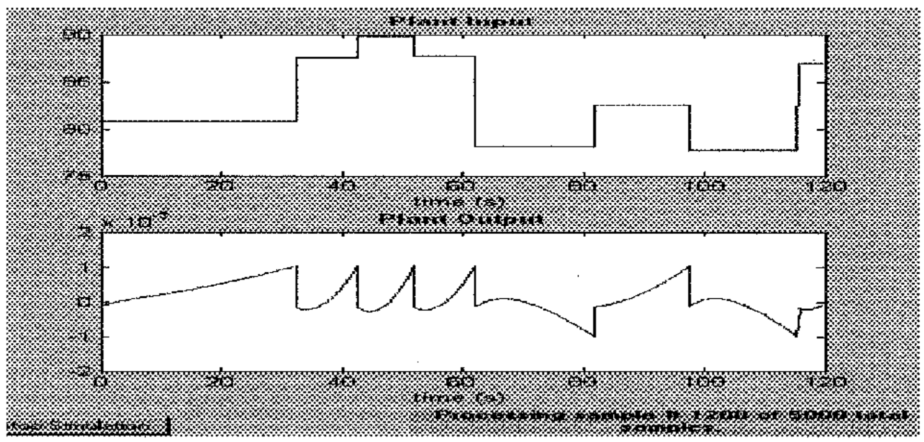


Figure 6 : Performance of Feed water Flow during Training Data

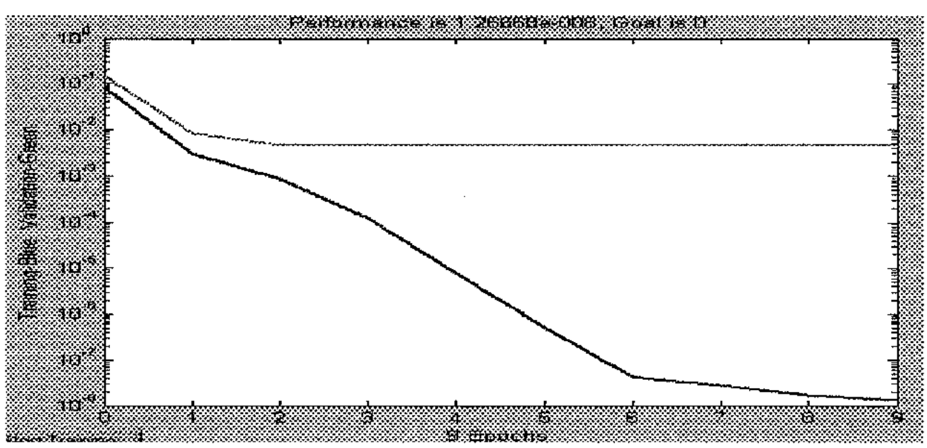


Figure 7 : Performance of the epochs after Train Network

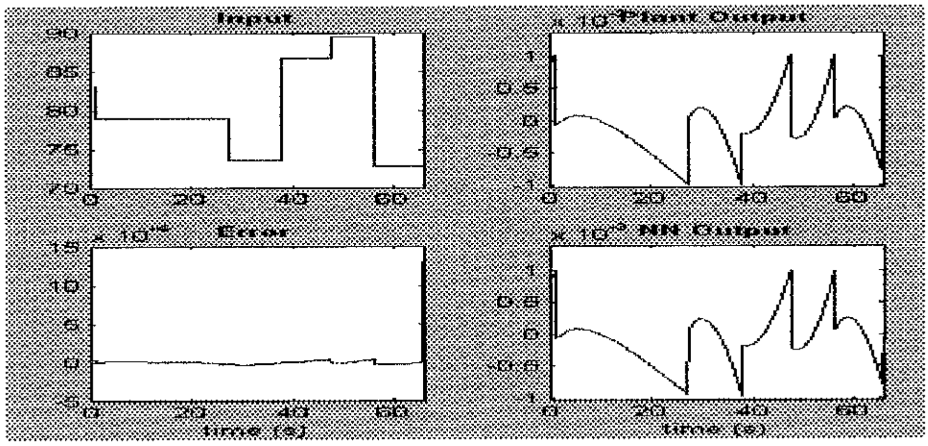


Figure 8 : Testing Data after Train Network

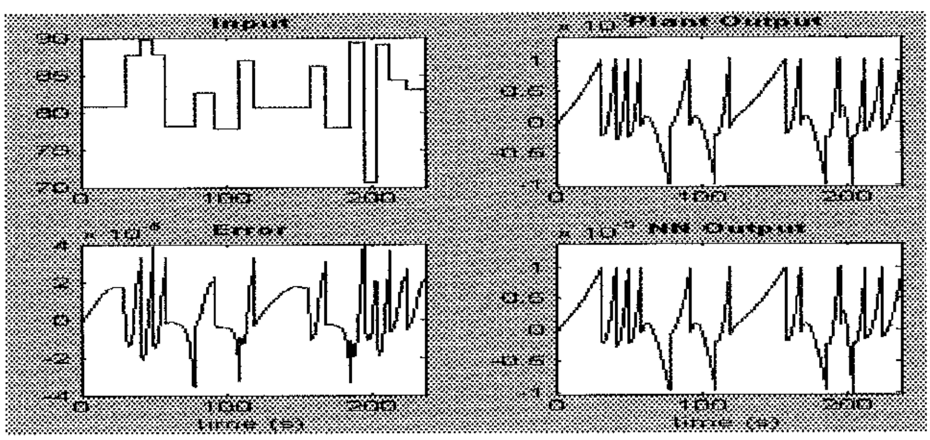


Figure 9 : Training Data after Train Network