Modeling of Primary Reformer Tube Metal Temperature (TMT)

by

Vishal Nanji Patel

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

JUNE 2009

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Modeling of Primary Reformer Tube Metal Temperature (TMT)

By

Vishal Nanji Patel

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 & ELECTRONICS ENGINEERING)

Approved by,

....

(Dr. Rosdiazli b. Ibrahim)

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

June 2009

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.

....

(VISHAL NANJI PATEL)

ABSTRACT

This report discusses the research done on the chosen topic, which is Modeling of Primary Reformer Tube Metal Temperature (TMT). The objective of the project is to develop a model that can predict the temperature of the reformer tubes. The scope of study focused on the modeling of the primary reformer TMT of PETRONAS Ammonia Sdn. Bhd. (PASB) plant. The primary reformer in PASB plant is used to crack hydrocarbons such as natural gas into its constituents which are carbon dioxide, carbon monoxide and hydrogen. Pressurized feed (30barg) of hydrocarbon and steam is fed into the reformer tubes and heated by the burners at about 800-1000°C to facilitate the hydrocarbon conversion. The temperature of the tubes is an important parameter to determine the life-time of the tubes. Operating the reformer beyond the TMT design limits can cause premature failures on the tubes which lead to production losses and higher downtime. Based on the literature survey, it shows that mathematical modeling and simulation approaches are used to determine the behavior of the reformer tubes. For this project, empirical model developed by integrating the process variable will be used to predict the reformer tubes temperature. Empirical model is developed based on the real-time data obtained from PASB plant. There is none journals or papers that have used this approach of modeling in predicting the primary reformer tubes temperature. Robust Quality Estimator (RQE) and MATLAB (System Identification Toolbox and Neural Network Toolbox) software are being used in developing models. Four types of model are developed and compared; those are RQE Linear model, RQE RBF model, MATLAB ARX model and MATLAB Neural Network model. Proper modeling techniques are discussed and being implemented in developing each models. Neural Network model is the best model developed in predicting the reformer TMT.

ACKNOWLEDGEMENT

First and foremost, the author would like to express his heartfelt gratitude and thankfulness to God; for His never ending blessings and gifted strength upon the author in conducting and completing this project successfully.

His deepest gratitude and thankfulness also goes to his immediate supervisor Dr. Rosdiazli b. Ibrahim for his never ending motivational encouragement, guidance, support, and confidence in the author throughout the entire project. Sincere thankfulness also goes to Ir. Idris for his guidance, advice and sharing of his valuable knowledge during the tenure of the research project.

The author expresses his greatest appreciation to Mr V. R. Harindran, Mr Saifol, Mrs Aneeta bt. Kajah, Mr S. Karthikeyan, Mrs Fauziah bt. Ali, Mr Adam Zakaria and Miss Normaya Shabudin from Research & Technology Division of PETRONAS Group Technology Solution (GTS) for their valuable contributions especially in sharing their expertise and knowledge as well as allowing the author to utilize its facilities and technical information.

The author would also like to take the opportunity to thank all executives personnel's of the PETRONAS Ammonia Sdn Bhd (PASB) for sharing plant technical information.

Last but not least, the author would like to thank his family and friends for the never ending support and advice contributing to the successful completion of his Final Year Project.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER 2

LITERATURE REVIEW

2.1 Failure Analysis

This section discusses the causes of primary reformer tubes failure. It shows that overheating is the main cause that reduces the life time of reformer tubes. Overheating will cause creep damage on the tubes which lead to the failure of the tubes. This section also shows the importance of conducting this project. Few pictures of tubes failure are attached in Appendix 3.

2.1.1 Failure analysis of HP40-Nb modified primary reformer tube of an ammonia plant [5]

Micro-structural failure analysis of the heat resisting HP40 Nb modified alloy was studied by light and electron methods. Samples from the failed reformer furnace tube were cut and prepared for metallographic examination. Examination with electron microscope was carried out with secondary and backscattered electron detectors; x-ray analysis was conducted at the grain boundary areas. Tubes which are filled with a supported nickel catalyst, methane reacts with steam, carbon dioxide and oxygen into synthesis gas. The overall heat of reactions may be positive, zero, or negative, depending on the process conditions. The catalyst plays a key role in developing overheating in the tubes.

When heavier than designed hydrocarbon was fed to the reformer. This heavier hydrocarbon was not compatible with the catalyst, and carbon formation occurred. Due to carbon formation and chocking, there was no consumption of heat flux and therefore, local overheating occurs which finally caused creep and failure of the tubes.

2.1.2 Reformer furnaces: Materials, damage mechanisms and assessment [6]

The assessment of damage in reformer furnaces is an important factor in determining their remaining safe life. In this paper the methodology of damage assessment is reviewed, and the concept of characteristic curves to assess damage is introduced: this provides a simplified procedure to give a realistic estimate of the extent of damage and the remaining life of reformer furnace tubes. An example is also given of a case study to determine remaining life in the presence of a part through-wall crack in a component in the header of a reformer furnace using a nonlinear fracture mechanics approach.

The paper shows that over-heating (increase of temperature) plays a major role in reducing the life-time of reformer tubes.

2.1.3 Failure analysis and remaining life assessment of service exposed primary reformer heater tubes [7]

Catalyst filled heater tubes made of cast HP-micro alloyed grade 35Ni25Cr1NbTi alloy used in the primary reformer furnace section of a fertilizer complex failed after 8 years in service. Failure analysis and remaining life assessment of the tubes were carried out based on mechanical strength evaluation, micro structural observations and accelerated stress rupture tests for Larson–Miller parameter (LMP) based remaining life prediction. Failed tube portions showed coarsened primary carbides of chromium and niobium at the inter-dendrite boundaries. Degradation of niobium carbide (NbC) into Ni–Nb–Si phase and partial conversion to the phase back to NbC was observed.

Significant expansion of O.D. of the tubes was observed for the failed portions indicating creep damage. Failed tubes microstructure showed advanced stage of ageing for the carbides. Transformation of NbC into Ni–Nb-Si phase and its reversion to NbC was observed. Longitudinal cracks have originated from the surface side of the tube. The failure of primary reformer furnace tubes in service is due to localized overheating which led to poor creep strength. Micro cracks have appeared only at the overheated portion of the tube. The tubes got overheated mainly due to flame irregularities. The unfailed portions showed lesser O.D. expansion and lesser degree of coarsening. Life extension was recommended for tubes with levels of O.D. expansion and micro structural conditions similar to the unfailed portions.

2.2 Modeling Technique of Reformer

This section helps in the designing and recognizing the technique in creating an empirical model for a linear and non-linear system. From the literature review that was carried out, most of the modeling of a reformer used a mathematical model to explain the temperature behavior of the reformer. Mathematical modeling of reformer tubes uses equations such as heat and mass transfer, energy balance, chemical kinetics and others. This publish literature are from university or research institution. This type of modeling is not applicable in industry to measure the reformer TMT. There is none paper that discuss the empirical modeling in predicting the reformer TMT. Empirical modeling technique such as black box modeling, neural network and fuzzy logic is discussed to determine the modeling technique relevancy with this project. Due to the availability of real-time data available from the PASB, empirical modeling technique will be used in building the reformer TMT predictive model.

2.2.1 Simulation of natural gas steam reforming furnace [8]

The paper proposed a set of mathematical model that describes the natural gas reformer. A total of 18 mathematical equations were described in developing the model. The paper has established to draw up homogeneous phase one-dimensional reaction kinetics equation in the reforming tubes, and compute the tube external radiant heat transfer with zone method. To verify the developed model, the result from the simulation was compared with the operating data from the production system in Urumqi Second Ammonia. Development of this model helps in design variable optimization research, such as, tube outlet temperature, tube pressure drop, maximum tube-wall temperature, production load, water carbon ratio and fuel distribution.

This paper also demonstrates the effect of tube diameter and number of tubes on operation behaviors. When the tube diameter increase, the number of furnace tubes reduces; overall heat duty, flue gas temperature and maximum tube-wall temperature all increases but the tube outlet temperature and tube pressure reduces. Analytic basis is provided for operators to choose correspondence operation parameters from the effect of production load, water carbon, fuel distribution on the reforming process. The simulation results were very encouraging.

The mathematical equations discuss are very complex and not easily applicable in the industry reformer.

2.2.2 Automation of reformer process in petroleum plant using fuzzy supervisory and model predictive multivariable control system [9]

The paper describes a hierarchical control system with fuzzy supervisory control system and model predictive multivariable control system (PMC system) in a petroleum plant. In the area of time-delay and interference PMC system is effective while fuzzy logic controller is effective for plants with large time-delay and non-linearity. The fuzzy supervisory control system which determines set points for the PMC system consists of two blocks, an estimation block and a compensation block. In estimation block, statistical model using multi-regression analysis is used to estimate parameter of plant operation. In compensation block, fuzzy logic is used to correct the output of the statistical model.

The hierarchical control system has been applied to the actual plant in an oil refinery. A satisfactory performance was obtained where the PMC system controls the average temperature of three reactors within the range that is given from the fuzzy supervisory control system, and showed a satisfactory performance. This system achieves stable control of the octane number of the reformer gasoline and reduction of operators' burden regardless whether the unit is operated at steady state or transient state.

The discussion of the paper is more on the control strategy in having an optimum plant operation. The use of hierarchical control system has increase the performance of the plant. Overall, this paper is not directly applicable in predicting the reformer TMT but the hierarchical system and fuzzy logic can be explore and applied in developing a predictive model.

2.2.3 Artificial neural networks for modeling the mechanical properties of steels in various application[10]

In this paper, the key properties of steels are predicted with the application of neural network. Three different back-propagation neural network models is introduced which are able to predict the (i) impact toughness of quenched and tempered steels exposed to various postweld heat treatment (PWHT) cycles, (ii) simulated heat affected zone toughness of pipeline steels resulting from in-service welding and (iii) hot ductility (reduction of area (ROA)) and hot tensile strength of microalloyed steels. The most common type of neural network is back-propagation neural network. It undergoes its learning phase by calculating an error between the predicted and actual output. This Hidden layer is used to establish the inter-relationships between the input variables and their relationship with the output to minimize the error between the actual and predicted output. Three data files were used in the modeling process which is training file, testing file and the sensitivity file. The data are normalized between 0-1 or -1-1 based on the intended transfer function.

The neural network models manage to successfully predicted impact toughness, HAZ hardness, ROA and hot tensile strength for variations in as-received material characteristics and process/test parameters. It also shows that NNs could successfully predict multiple mechanical properties. Manipulation of the sensitivity data showed the complex nature of NN and their ability to solve complex patterns. The need for expensive experimental investigation or inspection of steels used in various applications can be reduce by using NN, hence resulting in large economic benefits for organizations.

Overall, this paper is not directly related to this project since it discusses the ability of NN in predicting the key properties of steel. However, the usefulness of NN is explored further since it is a prevailing method in developing a predictive model. Hence a reformer TMT model is developed by using neural network. MATLAB Neural Network Toolbox is used for the model development and validation.

2.2.4 Black box modeling of steam temperature [11]

This paper discusses the black-box modeling of steam temperature. The model is developed based on the real-time data. The model was developed using the AutoRegressive eXogeneous (ARX) model structure by using the System Identification Toolbox. Black-box model are sometimes referred as an empirical model which is directly developed from experimental data. The black-box technique is used for four main objectives, which are:

- To understand the input-output behavior of a process.
- To predict future response of a process.
- To develop control system and tuning algorithms.
- To filter signals.

In describing the model accuracy, cross-validation technique was used. From the available set of input-output data, it was separated in training data and testing data. The training data set was used for model building while testing data for the model validation. The model for estimating steam temperature was successfully developed with a linear fourth order ARX model.

Overall, the paper shows the application of system identification toolbox in developing a predictive model. The paper also discusses the validation technique for the model developed. The toolbox and validation technique are used in developing and validating the reformer TMT model.

2.3 Modeling Software

Mainly there are two software used for modeling the primary reformer, those are Robust Quality Estimator (software by Yokogawa Kontron Sdn. Bhd.) and MATLAB (System Identification Toolbox and Neural Network Toolbox).

2.3.1 RQE

Robust Quality Estimation (RQE) is industrial software which is used as a modeling tool for inferential measurements. Process data are used to identify linear and non-linear, static and dynamic models. RQE is capable of solving practical problems such as choosing model inputs, identifying input-output deadtimes, handling collinear or dependent inputs, handling nonlinearities, and designing an update mechanism for robust online performance.

For error calculation, this software will calculate the root mean square error (RMSE), Sigma and also Index.

RMSE is the root mean square of the prediction error which is regarded as a statistical criterion that captures both the bias and the standard deviation effects of a model. The precise definition of RMSE is,

$$RMSE = \sqrt{\frac{\sum e_i^2}{N}}$$

In this formula, $e_i = Y_{m,i} - Y_{p,i}$ is the prediction error at sample i, $Y_{m,i}$ is the measured reference value at sample i, $Y_{p,i}$ is the fit prediction at sample i, and N is the number of data points.

Sigma is the standard deviation of the prediction error which captures the error distribution. The precise definition of Sigma is,

$$Sigma - \sqrt{\frac{\sum(e_i - \bar{e})^2}{N-1}}$$

In this formula, $e_i = Y_{m,i} - Y_{p,i}$ is the prediction error at sample i, $\bar{e}$ is the average prediction error, $Y_{m,i}$ is the measured reference value at sample i, $Y_{p,i}$ is the fit prediction at sample i, and N is the number of data points.

Index is the ratio between the mean squared prediction error and the reference value variance. For instance, if the prediction error is bigger than the reference value variability, then Index > 100; this indicates a poor prediction model. The precise definition of Index is,

$$Index = \frac{\sum e_i^2}{\sum(Y_{m,i} - \bar{Y}_m)^2} \times 100$$

In these formulas, $e_i = Y_{m,i} - Y_{p,i}$ is the prediction error at sample i, $Y_{m,i}$ is the measured reference value at sample i, $Y_{p,i}$ is the fit prediction at sample i, and $\bar{Y}_m$ is the average measured reference value.

There are mainly two types of different model that can be developed using the RQE software which are Linear Model and RBF Model. Linear model uses the Multiple Linear Regression (MLR) method in developing a model. MLR is the simplest modeling method offered which uses multiple least squares regression to find the best (minimum squared error) static linear model between several input variables and a single output variable. This is done by training the data. The training data set must contain sufficient excitation of the variables to calculate a good fit.

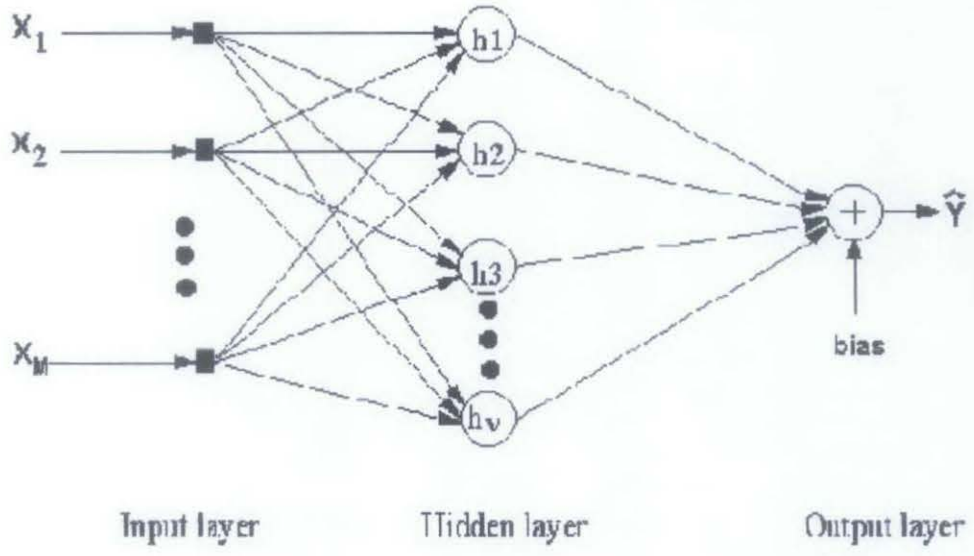

Figure 1: RBF Network Overview

Radial Basis Function (RBF) models are a special kind of neural network that is very powerful for nonlinear modeling purposes. RBF is used for interpolation and approximation problems and is regarded as feedforward neural networks where the activation functions (or transfer functions) in the nodes are of RBF type. An RBF network is characterized by the type of radial basis function used and by a number of parameters, namely the number of nodes, the width of the RBF function, the node centers, the output weights, and a bias term.

Using RQE, all the parameters are automatically calculated using state-of-the-art methods. RQE is also capable of avoiding the over fitting phenomenon during the model development stage. The figure below displays an RBF model with M inputs and í nodes. The h_i transfer functions are RBF functions that depend on some parameters such as the width β and the centers c_i.

In a mathematical setting, this RBF model can be written as:

$$Y = \theta_0 + \sum_{i=1}^{v} \theta_i h_i(\|X - c_i\|, \beta)$$

where $X = [X_1, X_2, \dots X_M]^T$ is the input vector, $\| \; \|$ denotes the Euclidean norm, $\theta_i (i = 1 \dots v)$ are the output weights and θ_0 is the bias term. The RBF h_i functions can be of different types:

1) Multi Quadratic (MQ)

$$h_i = \sqrt{r^2 + \beta^2}, \qquad r^2 = \|X - c_i\|^2$$

2) Reciprocal Multi Quadratic (RMQ)

$$h_i = 1/\sqrt{r^2 + \beta^2}, \qquad r^2 = \|X - c_i\|^2$$

3) Gaussian

$$h_i = e^{\left(-\frac{r^2}{\beta^2}\right)}, \qquad r^2 = \|X - c_i\|^2$$

2.3.2 MATLAB (System Identification Toolbox)

System identification toolbox is useful in creating a black-box models of a dynamic systems based on the real-time input and output data from plant. The toolbox is helpful in modeling systems that is not easily represented in term of first principal. This toolbox has a simple graphical user interface (GUI) in helping the user to easily organize the data and build a black-box model. The following figure shows the GUI screen shoot:

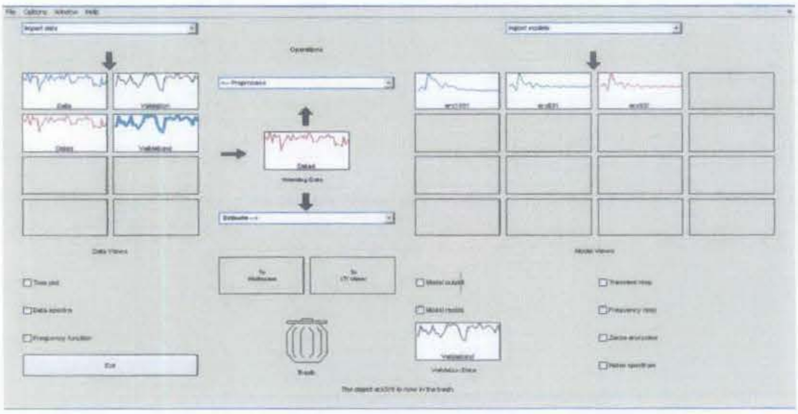

Figure 2: System Identification Toolbox GUI

Data will be imported to the GUI and pre process before developing a new model and validating the model. The criterion for choosing the best model is based on the fit value that will be calculated by the software. The precise definition of the fit is:

$$FIT = \left[1 - \frac{\sqrt{\Sigma(e_i)^2}}{\sqrt{\Sigma(Y_{m,i} - \overline{Y}_m)^2}}\right] \times 100$$

Where $e_i = Y_{m,i} - Y_{p,i}$ is the prediction error at sample i, $Y_{m,i}$ is the measured reference value (output value) at sample i, $Y_{p,i}$ is the fit prediction (simulated output value) at sample i, and $\overline{Y}_m$ is the average measured reference value. A higher number for fit value means a better model. For instance, if the prediction error is bigger than the reference value variability then the FIT value will be less than 0.

Parametric model can be developed using the system identification toolbox. A general, linear, discrete-time model can be written as:

$y(t) = G(q,\theta)u(t) + H(q,\theta)e(t)$

where $G(q,\theta) = B(q)/A(q)$ and $H(q,\theta) = C(q)/D(q)$

There is four types of model, those are:

- ARX model – Easiest to estimate but the disadvantage is that the disturbance model $H(q,\theta)e(t) = 1/A(q)$ is combined with the system's poles. If signal-to-noise ratio is good, this advantage is less important.

  $A(q)y(t) = B(q)u(t)+e(t)$

- ARXMAX model – More flexible and capable of handling disturbance model because of the C polynomial.

  $A(q)y(t) = B(q)u(t)+C(q)e(t)$

- OE model – The system dynamics can be described separately and no parameters are wasted on the disturbance model.

  $y(t) = [B(q)/A(q)]u(t)+e(t)$

- BJ model – Complete model which the disturbance properties are model separately from the system dynamic

  $y(t) = [B(q)/A(q)]u(t)+[C(q)/D(q)]e(t)$

2.3.3 MATLAB (Neural Network Toolbox)

Neural Network is composed of simple elements operating in parallel. The network is trained so that a particular input leads to a specific target output by mathematical computation. As shown in figure 3, the network adjustment is done by comparing the output target until the network output matches the target. Many pairs of the inputs and targets are needed to train the network.

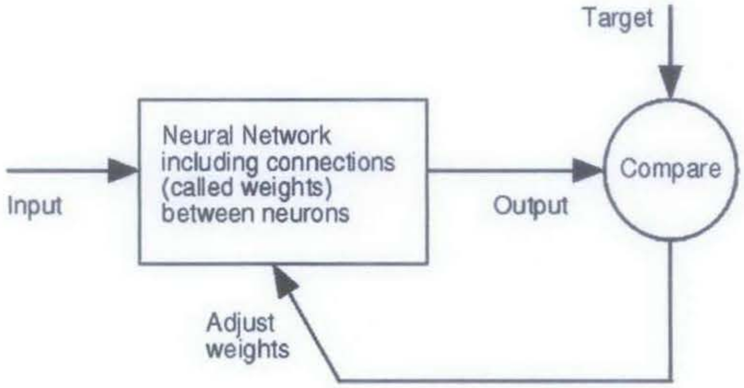

Figure 3: Flow in Neural Network

A network can have several layers as shown in figure 4. Each layer has a weight matrix w, a bias vector b and an output vector a. The network has R inputs, S^1 neurons in the first layer, S^2 neurons in the second layer and S^N neurons for N layer. The outputs of each intermediate layer are the inputs to the following layer. Different roles are played by each of the layers of the multilayer network. Output layer is referred for layer that produces the network output. All others layers are called hidden layers. The network shown below has one output layer (layer 3) and two hidden layer (layer 1 and 2).

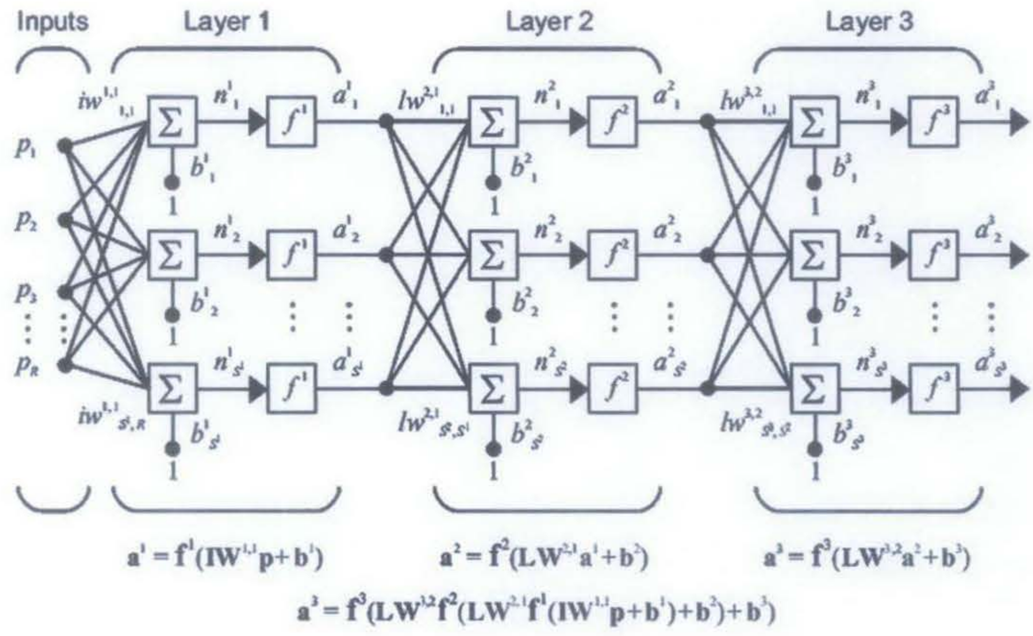

Figure 4: Multilayer Neural Network

Few critical parameters that need to be considered in developing a model using neural network are:

- Network Types
  - Shows the architecture of the network.
  - Examples are feedforward (newff), perceptron (newp) and elman back propagation (newelm).
- Number of Layers and Neurons
  - Adjusted based on complication of the available data.
  - Selection is done by trial and error.
- Transfer Function
  - Function that calculates a layer's output from its net input.
  - Examples are hardlim, purelin, logsig, tansig, tribas and others.

- Training Function
    - Function that define and execute a predefine algorithms in training the network.
    - The weights and biases of a network is modified to achieve the require target during training.
    - Examples are trainlm, trainbfg, trainrp, traingd and others.
- Performance Function
    - Calculated the error after every epoch.
    - Examples are mean square error (mse), sum squared error (sse) and mean absolute error (mae).
- Training Styles
    - There are two types of training which are called incremental training and batch training.
    - In incremental training the weights and biases of the network are updated each time an input is presented to the network.
    - In batch training the weights and biases are updated after all the inputs are presented.

2.4 PASB Primary Reformer

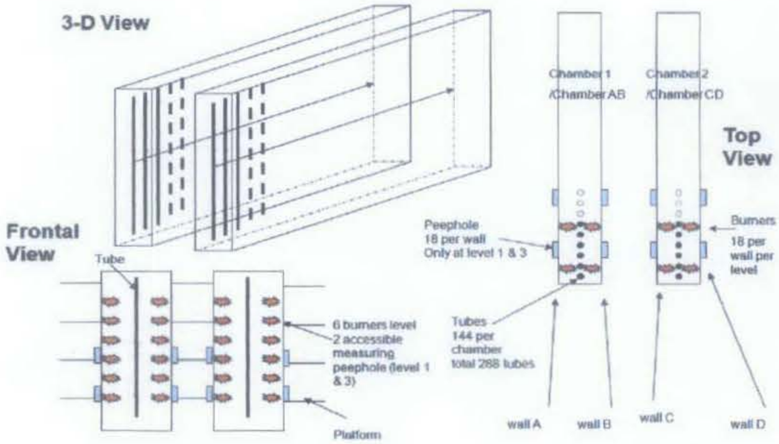

Figure 5: Overview of PASB Primary Reformer

Referring to figure 5, PASB is using a steam reformer consisting of two chambers with 144 reformer tubes installed at each chamber (total of 288 tubes). This primary reformer is using side burner with 6 burners' level. At each level, there are 18 peep holes on each side of the wall. Peep holes are used to measure the tube temperature by using a portable infra-red temperature measuring device. The tubes temperatures are manually measured by the operator at an interval of one day (refer to Appendix 4 for the manual entry form). Only level 1 and 3 are accessible for the operator to measure the tubes temperature. The tubes temperature reading obtains from PASB will be used to develop the model.

Each tube is fed with pressurized hydrocarbon and steam (approximately 30 bars) and heated between 800 to 900°C by numbers of burners. The tubes are designed for a lifetime of 100,000 hours with a maximum TMT of 1020°C. Currently in PASB plant, there are 3 primary reformer tubes failures in the past 7 years.

These failures have cause huge losses to PASB:

- RM 50,000 for replacement cost per tube – RM 14.4 million for all 288 tubes.
- RM 20 million per catalyst batch.
- RM 30 million production loss.

CHAPTER 3

METHODOLOGY

3.1 Procedure Identification

There are two main procedures required for completing this project. Those are the overall project flow and model development and validation flow for each types of model. The timeline and brief description for each stage taken in completing this project is shown in the Gantt chart attached in Appendix 5.

3.1.1 Overall Project Flow Chart

Figure 6 shows the overall important steps taken for modeling the primary reformer TMT. Understanding of the process and behavior of the complex reformer is necessary in achieving the best model for predicting the reformer TMT.

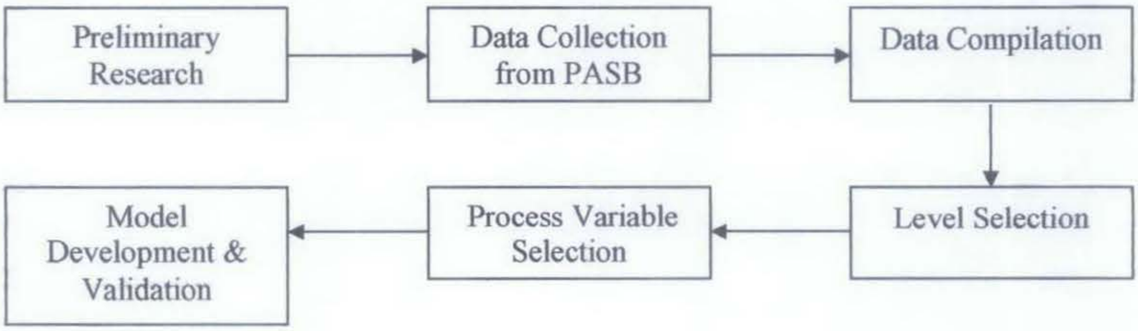

Figure 6: Overall Project Flow Chart

3.1.2 Model Development and Validation Flow Chart

In the model development and validation stage, two different types of software are used. Those are RQE and MATLAB software.

The collected data from PASB should cover the space in which the model will be expected to be operating. Collected data is divided into training and validation data set. The same set of data is used in developing model using each software; hence model comparison can be done at the end of the project. Error analysis is done by calculating the RMSE and Index value for each developed model. Error analysis will help in choosing the best model.

First, RQE is used in developing a model. This model is used as the baseline model for future model development using MATLAB software. Figure 7 shows the flow chart for RQE modeling.

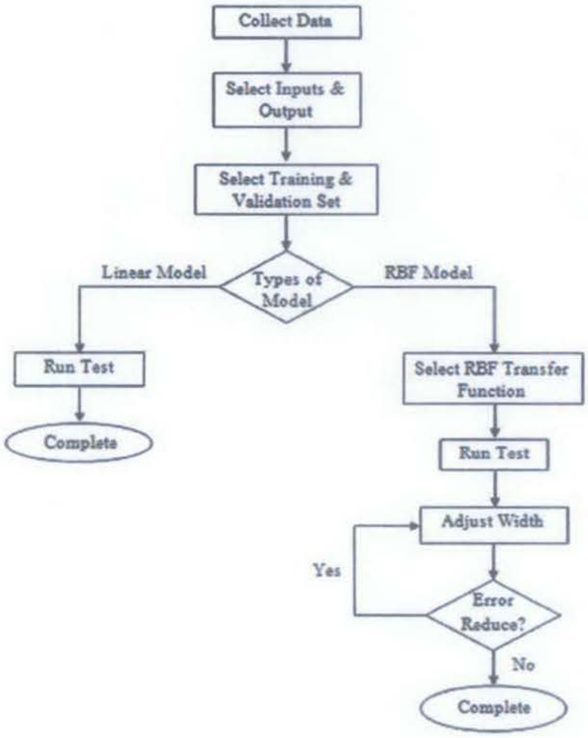

Figure 7: RQE Modeling Flow Chart

Second, MATLAB System Identification Toolbox is used in developing a model. Figure 8 shows the flow chart for System Identification modeling.

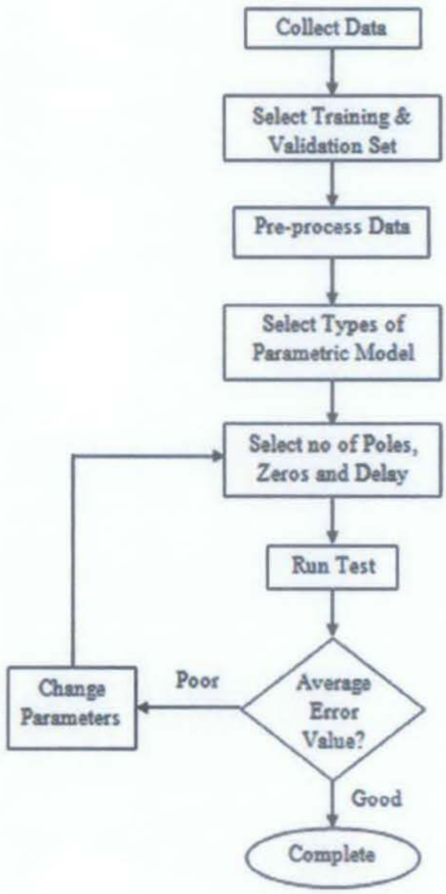

Figure 8: System Identification Modeling Flow Chart

Finally, model is developed using MATLAB Neural Network Toolbox. Figure 9 shows the flow chart for Neural Network modeling.

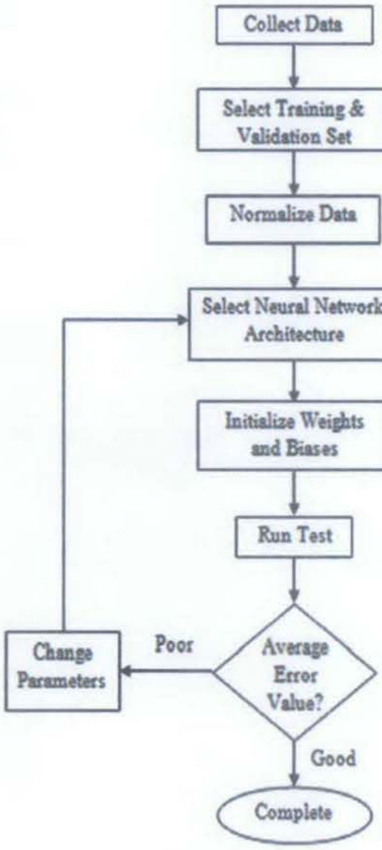

Figure 9: Neural Network Modeling Flow Chart

3.2 Tools and Equipment

The tools that will be used thorough out the project are:

1) Robust Quality Estimator software
2) MATLAB Toolbox. There is basically two types of modeling toolbox that will be used, which are:
    - System Identification Toolbox
    - Neural Network Toolbox

CHAPTER 4

RESULT AND DISCUSSION

4.1 Input and Output Selection

PASB have provided two sets of data. The following table shows the description of the data:

Table 1: Description of PASB Data

Set	No. of Process Variable	No. of Data		Process Variable Sampling Time	Period
		Level 1	Level 3		
A	19	-	187	1 hour	June 07-June 08
B	19	27	43	1 hour	July 08-October 08

4.1.1 Data compilation

Both set of data are compiled and being used for model development based on the following assumption:

1) Average reading of process variable value from 11pm to 2am is taken since this is the period where the operator will be manually obtaining the reading of reformer TMT. The average value of process variables is use as the inputs for the model. The reading of process variable was obtained from the DCS.
2) Since PASB primary reformer consist of two chamber, for each chamber the average temperature value of the 144 tubes is calculated to be used as the output of the model. Hence, each chamber will be defined with a unique model.
3) After compiling the process variable average values with the corresponding TMT average values, irrelevant data are removed. The removed data are readings beyond the operating range of the process, it could be due to instrument failure or

taken during the start-up of the plant. The process variables values used for model development are steady-state values.

4.1.2 Level selection

The TMT readings for data of set B are from two different levels (level 1 and level 3). The following graph shows the average tubes temperature distribution for different level and chamber:

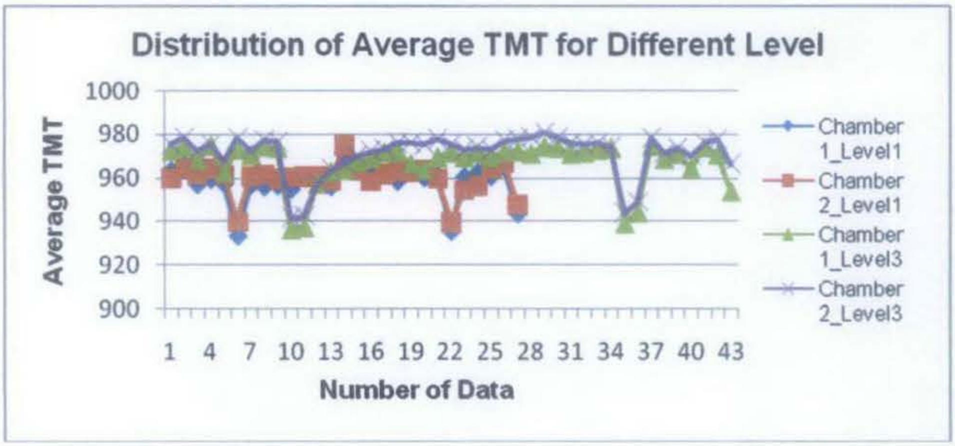

Figure 10: Distribution of Average TMT for Level 1 and 3

Based on the average TMT distribution from figure 10, level 3 has a higher temperature reading compare to level 1. Hence, data from level 3 will be used for future model development. Selection of level 3 also shows that the data from set A and B can be combine and used for model development. This will increase the amount of data for developing a model since data from Set B is not sufficent.

4.1.3 Correlation

The correlation is one of the most common and useful statistics. A correlation coefficient is a single number that describes the degree of relationship between two variables. It is define as a measurement of how much one random variable depends upon the other.

The correlation coefficient of two variables is defined in terms of their covariance and standard deviation,

$$\rho = \frac{cov(x,y)}{\sigma_x \sigma_y}$$

where $-1 \leq \rho \leq 1$. If there is no relationship between the variables then the correlation coefficient will be zero and if there is a perfect positive match it will be one. If there is a perfect inverse relationship, where one set of variables increases while the other decreases, then the correlation coefficient will be negative one. This type of correlation is often referred as the Pearson's Correlation Coefficient. Using MATLAB function called 'corrcoef' the correlation coefficient between the process variables (inputs) and TMT reading (outputs) is obtained. The table below summarizes the correlation coefficient,

Table 2: Correlation Coefficient

Process Variables	**Correlation Coefficient**	
	Chamber 1	**Chamber 2**
FRCA1202.PV	0.795	0.803
FRCA1204.PV	0.549	0.541
FRCA1304.PV	-0.335	-0.295
TIA1212.PV	0.704	0.711
PR1210.PV	0.637	0.622
FIC1250.PV	0.018	-0.026
FIC1254.PV	-0.310	-0.365
FIC1252.PV	0.405	0.431
FIC1253.PV	0.514	0.537
FIA1216.PV	0.784	0.779
PICA1253.PV	0.679	0.626
FFRA1208.PV	-0.791	-0.796
FICA1251.PV	0.766	0.767
C1.PV	-0.085	-0.098
TI1220.PV	0.295	0.252
QIA1204.PV	0.390	0.407
TRA1232.PV	0.799	0.773
TIA1234.PV	0.494	0.569
TIA1235.PV	0.631	0.659

The obtained correlation coefficient is useful in determining the important process variable that influences the temperature of reformer tubes.

4.1.4 Process variable selection

The number of process variable used for model development need to be reduced by selecting the best process variables that affect the tubes temperature. The following are the location and description of the 10 process variable chosen for model development:

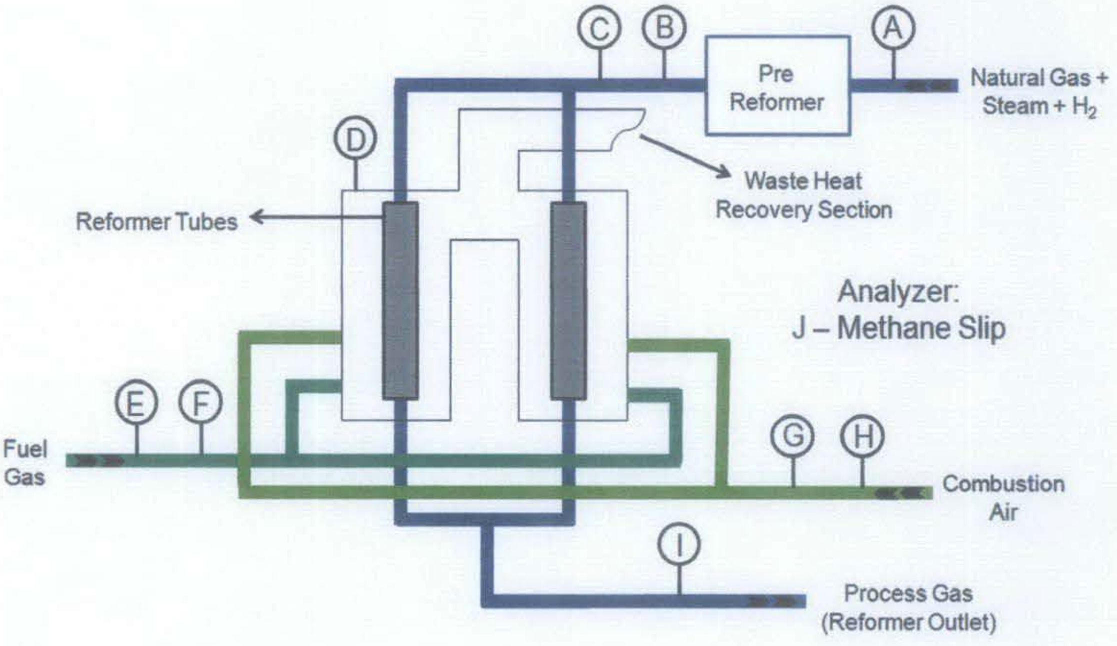

Figure 11: Location of Selected Process Variables

Table 3: List of Selected Process Variables

Tag	Instrument Tag No	Unit	Description
A	FFRA1208		S/C Ratio
B	PR1210	KG G	Feed Pressure
C	TIA1212	DEG C	Feed Temperature
D	TIA1234	DEG C	Flue Gas Temperature
E	FICA1251	GJ/H	Total Calorific
F	PR1210	KG G	Fuel Pressure
G	TI1220	DEG C	Air Temperature
H	FIA1216	NM3/H	Combustion Air
I	TRA1232	DEG C	Reformer Outlet Temperature
J	QIA1204	%	Methane Slip

The remaining process variables are neglected in model development. These process variables are neglected based on the process understanding and also based on correlation value between the process variable and TMT. The following are the list of the neglected process variable and reasons for neglecting it in model development:

Table 4: List of Rejected Process Variables

No	Instrument Tag No	Data Unit	Description	Reason
1	FRCA1202	NM3/H	Natural Gas Feed	The value is represented by the mathematical calculation of FFRA1208
2	FRCA1204	KG/H	Total Process Steam	
3	FRCA1304	NM3/H	Recycled CO2	Not a critical process variable and low correlation value (-0.30)
4	FIC1250	NM3/H	Natural Gas Fuel	The value is represented by the mathematical calculation of FICA1251
5	FIC1254	NM3/H	Off Gas N2 Wash	
6	FIC1252	NM3/H	Off Gas PSA	
7	FIC1253	NM3/H	Off Gas Cold Box	
8	C1		C1 Composition	Very low correlation value (-0.09)
9	TIA 1235	DEGC	Flue Gas Temperature	The value represented by TIA1234

4.2 Model Development and Validation

A total of 200 healthy data are used for model development and validation. This data is divided into two set of data consist of 100 data for model development and 100 data for model validation. Developed models are based on the 10 inputs selected earlier and the output for the model is Chamber 1 and Chamber 2 average TMT (note: only one chamber is considered as the output for each model).

4.2.1 RQE software

There are mainly two types of different model that are developed using the RQE software which are Linear Model and RBF Model.

4.2.1.1 Linear Model

The structure of this model is given by simple linear equation,

$$y = a_1x_1 + a_2x_2 + \dots\dots\dots + a_nx_n + b$$

where y is the output, x is the input, a is the weight of particular input and b is the bias value. The following is the graphs for Linear model:

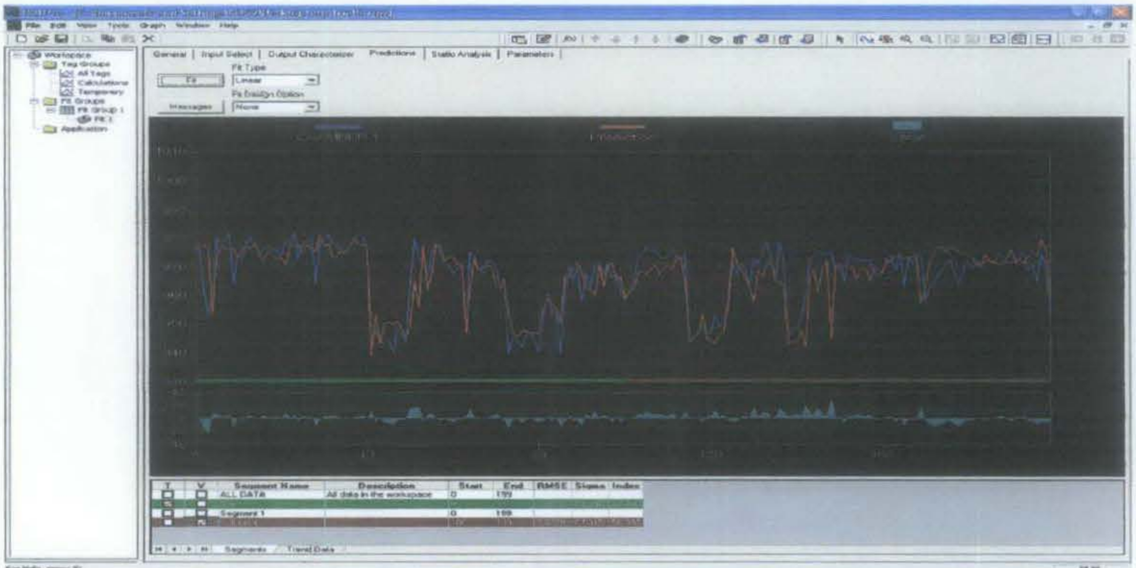

Figure 12: Graph for RQE Linear Model (Chamber 1)

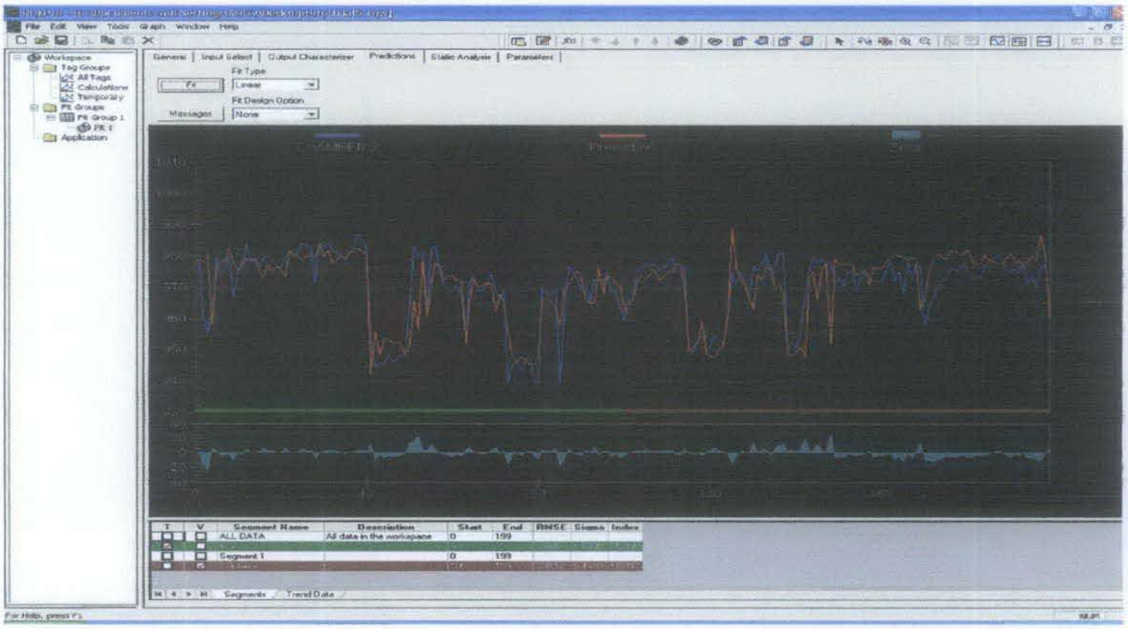

Figure 13: Graph for RQE Linear Model (Chamber 2)

The mathematical equation for RQE Linear Model is attached in Appendix 6. The table below shows the error analysis:

Table 5: Error Analysis for RQE Linear model

Data Set	Chamber	RMSE	Index
Training	1	6.039	22.703
	2	6.567	25.399
Validation	1	6.627	56.092
	2	6.451	56.347

4.2.1.2 RBF Model

The structure of this model is as shown in figure 1 previously which depends on the number of neurons and RBF transfer function. RMQ transfer function is used for creating a model since the output is better compare to other RBF transfer function. By adjusting the width value, the following is the best graphs for RBF model:

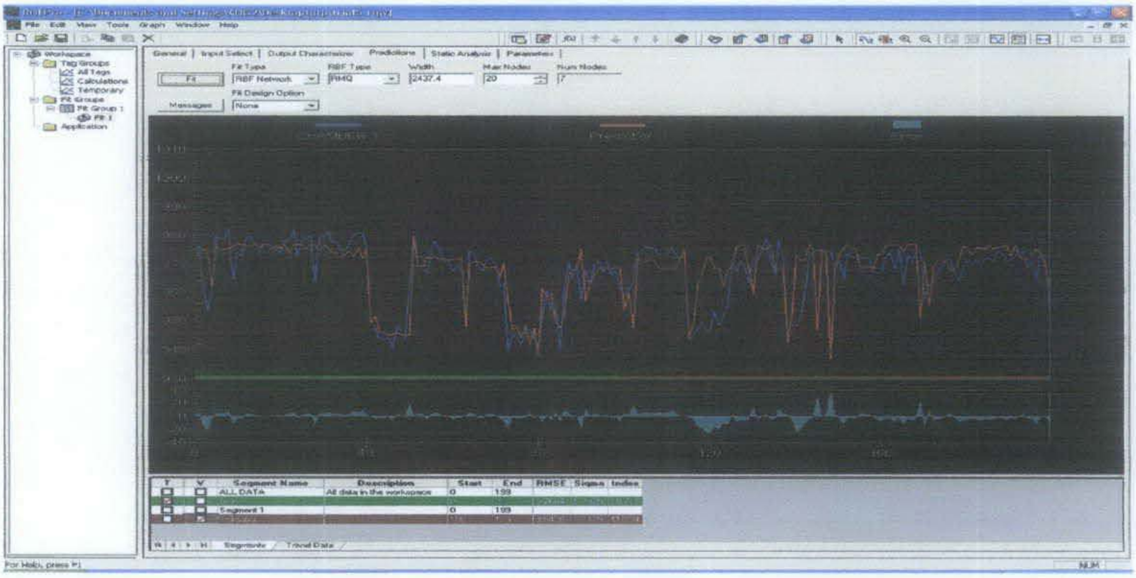

Figure 14: Graph for RQE RBF Model (Chamber 1)

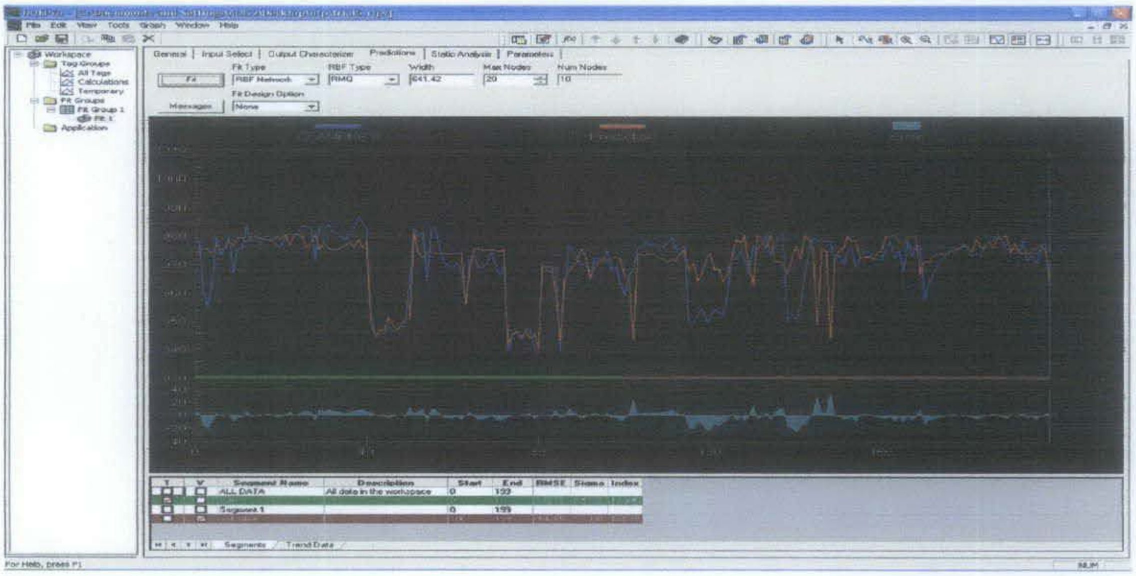

Figure 15: Graph for RQE RBF Model (Chamber 2)

The mathematical equation for RQE RBF model is attached in Appendix 7. The table below shows the error analysis:

Table 6: Error Analysis for RQE RBF model

Data Set	Chamber	RMSE	Index
Training	1	5.768	20.717
	2	5.527	17.995
Validation	1	9.348	116.34
	2	9.477	121.59

4.2.2 MATLAB software

Two types of toolbox are used in developing model using MATLAB. Those toolbox are System Identification Toolbox and Neural Network Toolbox.

4.2.2.1 System Identification Toolbox

The simplest parametric model which is ARX model is developed using System Identification Toolbox. The structure of this model is the simple linear difference equation,

$$y(t) + a_1y(t\text{-}1) + \ldots\ldots + a_{na}y(t\text{-}na) = b_1u(t\text{-}nk) + \ldots\ldots + b_{nb}u(t\text{-}nk\text{-}nb\text{+}1)$$

which relates the current output y(t) to a finite number of past outputs y(t-k) and inputs u(t-k). The structure is thus entirely defined by the three integers na, nb, and nk. na is equal to the number of poles and nb-1 is the number of zeros, while nk is the pure time delay (the dead time) in the system. For multi input systems, nb and nk are row vectors, where the ith element gives the order/delay associated with the ith input. Based on trial and error by changing the value of na, nb and nk; the following is the graphs of the best ARX model:

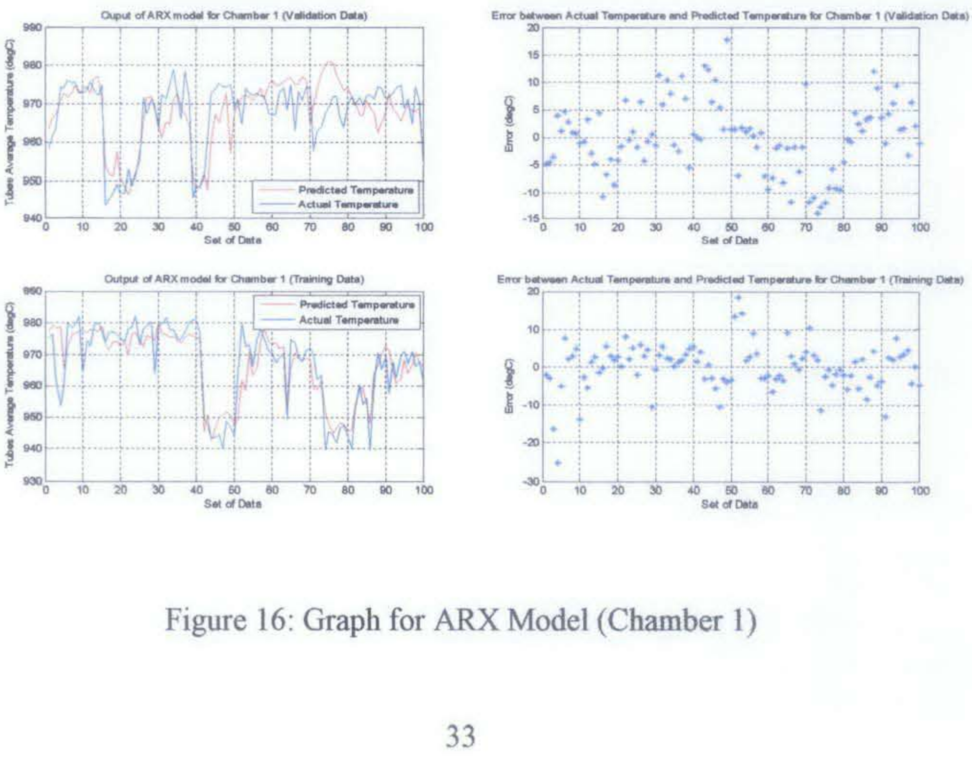

Figure 16: Graph for ARX Model (Chamber 1)

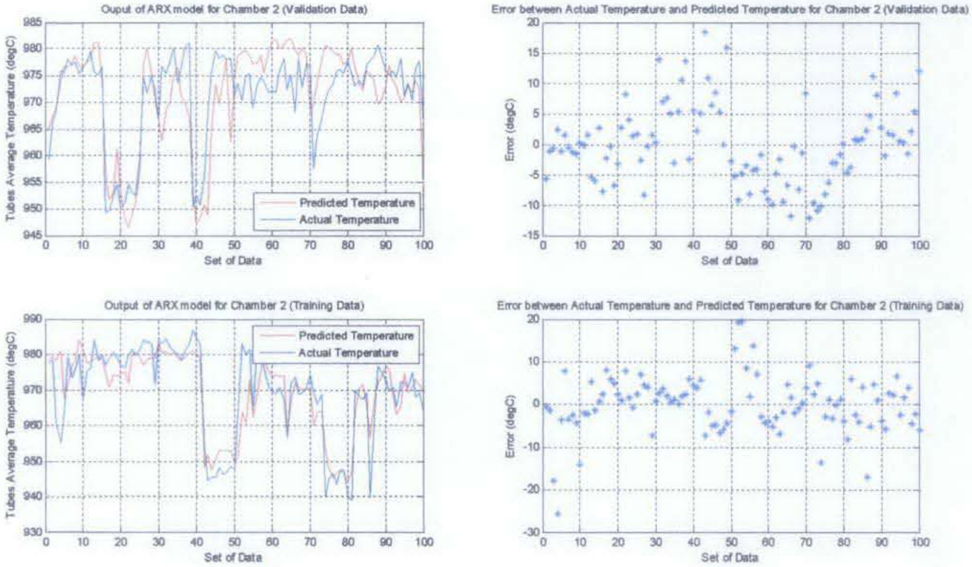

Figure 17: Graph for ARX Model (Chamber 2)

GUI is used for selecting the best ARX model. MATLAB programming code is written to detrend the data and for error analysis. The programming code is attached in Appendix 8. The mathematical equation for ARX model is attached in Appendix 9. The table below shows the error analysis:

Table 7: Error Analysis for ARX Model

Data Set	Chamber	RMSE	Index
Training	1	6.170	23.705
	2	6.673	26.228
Validation	1	6.3986	52.295
	2	6.349	54.576

4.2.2.2 Neural Network Toolbox

The general structure of neural network model is as shown in figure 4 previously. The architecture of the neural network can be design by the user. For this project, 2 layer network with feedforward network is chosen for the overall network architecture. The network is train using batch training method since it is a fast training method. The transfer function of second layer (output layer) is set to 'purelin' with 1 neuron.

After defining the architecture and training method, the important parameters such as transfer function, number of neuron and training function is chosen based on trial and error. A detailed MATLAB programming code (attached in Appendix 10) is written to help the model development and validation since this method is easier and faster for finding optimum parameters instead of using the GUI. The tables below show the variation in the parameter and effect on the error. This test is carried out for Chamber 1 data.

Table 8: Variation in Number of Neuron

Number of Neurons	RMSE	
	Training	Validation
40	9.653	6.819
30	9.663	6.810
20	6.084	5.966
10	6.015	6.282

Table 9: Variation in Transfer Function

Transfer Function	RMSE	
	Training	Validation
logsig	6.856	5.463
purelin	12.673	8.995
tribas	12.673	8.995

Table 10: Variation in Training Function

Training Function	RMSE	
	Training	Validation
trainlm	1.3318	33.0157
trainbr	29.7867	25.6759
traingdx	6.3182	5.8528
traincgb	6.856	5.463

During the neural network training, a performance graph is plotted (as shown in Figure 18). This graph shows the error Goal set and the improvement in the error after every epoch. The network may not reach the error goal if the training is stuck at local minima. This case can be overcome by increasing the number of neuron.

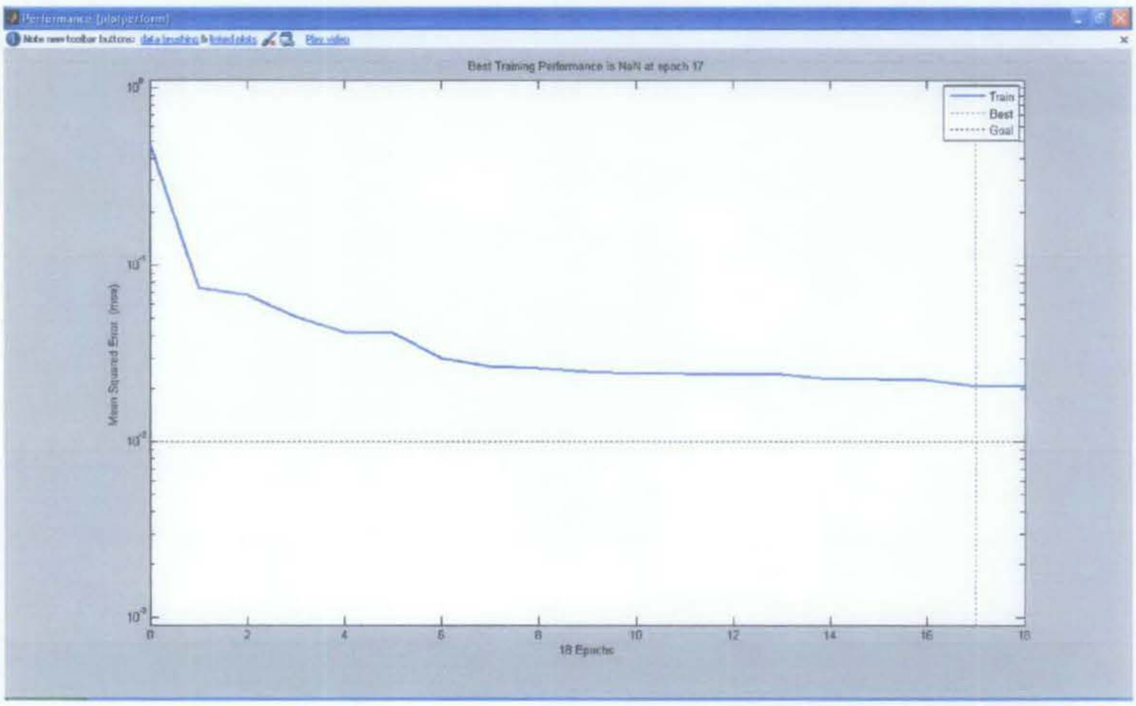

Figure 18: Graph of Training Performance

Once the network with desired error goal is found, it is tested with the validation data set. The error is calculated and if the error is acceptable, the model is accepted. If the error is too large, it could be due to 'overfitting' case during the training stage. Overfitting is a case in which the error on the training set is a very small value but when new data is presented to the network, the error is very large. Example of overfitting case is as shown in the figure below:

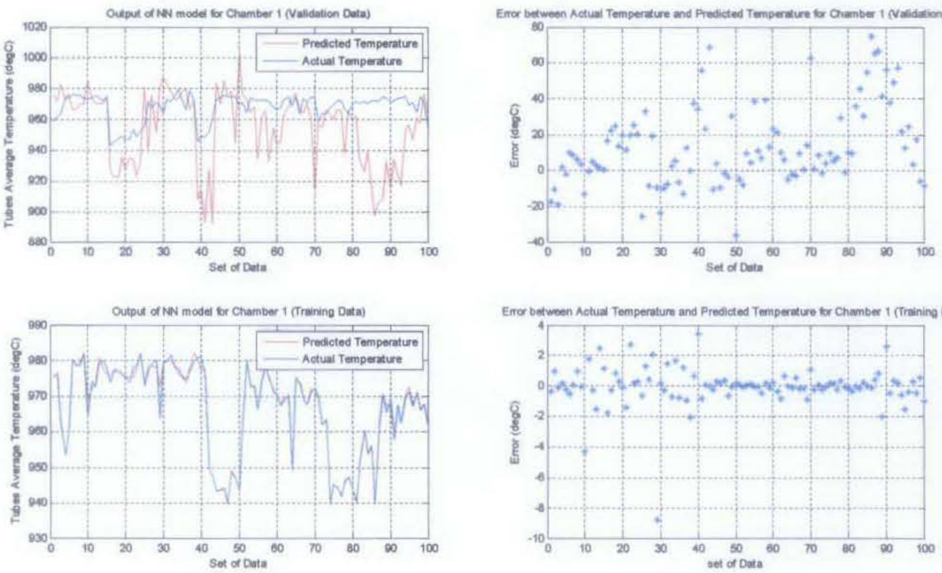

Figure 19: Graph of Overfitting Phenomena

The graph shows the error for training data set is extremely small while the error for validation data set is large and unacceptable.

By changing the important parameters and considering all the constrain in neural network modeling, the following graphs are the best Neural Network model,

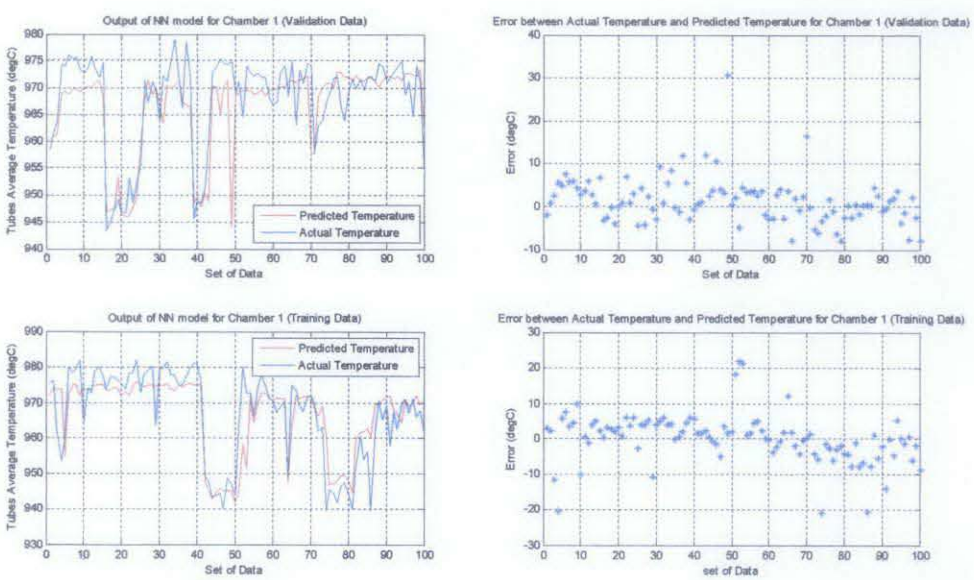

Figure 20: Graph for Neural Network Model (Chamber 1)

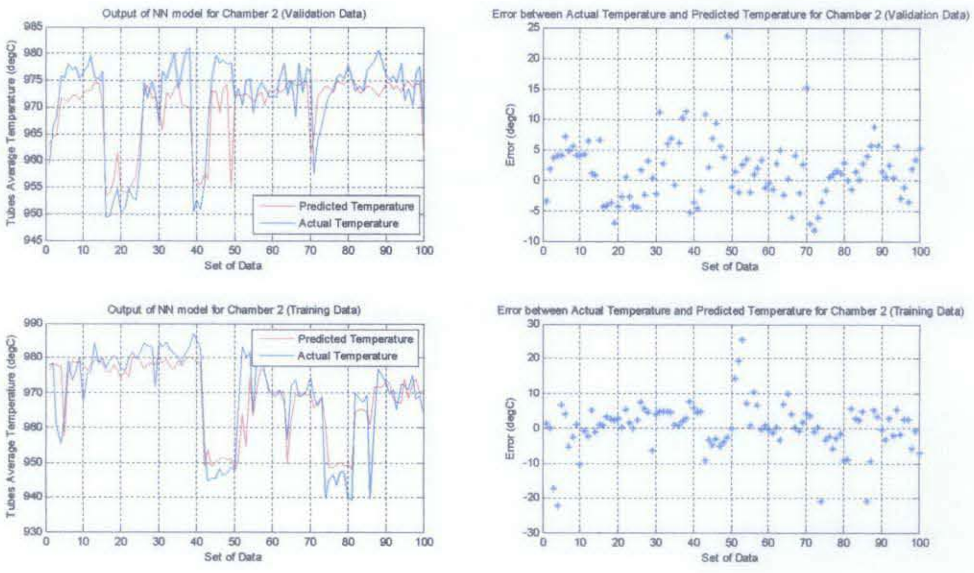

Figure 21: Graph for Neural Network Model (Chamber 2)

The following table summarizes the architecture and parameters value for the Neural Network models:

Table 11: Neural Network Model Architecture

Parameters	Chamber 1	Chamber 2
Types of Network	newff	newff
Network Layer	2	2
Number of Neuron	18	13
Transfer Function	logsig	logsig
Learning rate	0.2	0.2
Training Function	traincgb	traincgb
Number of Epoch	9	18

The mathematical equation for Neural Network model is attached in Appendix 11. The table below shows the error analysis for Neural Network model:

Table 12: Error Analysis for Neural Network Model

Data Set	Chamber	RMSE	Index
Training	1	6.859	29.295
	2	6.884	27.910
Validation	1	5.463	38.113
	2	5.233	37.083

4.3 Discussion

In general, four types of model are developed in predicting reformer TMT using RQE and MATLAB software, those models are:

- Linear Model
- RBF Model
- ARX Model
- Neural Network Model

Collected data from PASB are divided into training and validation set, each consist of 100 data sets. The data is assumed to cover the operating region for each process variables since this is critical in developing a robust model. If the data does not cover the operating region, the developed model will not be applicable to be used in the plant since it would not be able to predict accurately the reformer TMT. Same amount of data are used for developing all types of models with the intention that error analysis can be done for model comparison and selection for best model. Since the objective of developing the predictive model is to reduce overheating on reformer tubes, TMT readings from level 3 are chosen for model development because it has a higher temperature profile compare to level 1.

Total of 10 process variables are chosen from 19 available process variables. Process understanding is necessary in selecting the best process variable. This stage is very critical since it determine the success of developing accurate model. The selections are done based on the relationship between the process variables and TMT. The relationship is shown by the correlation coefficient table. Most of the selected process variable has more than 0.5 correlation coefficient except QIA1204 (methane slip) and TI1220 (air temperature). However, this two process variables are critical process variables which are affected by the performance of the reformer itself. Hence it does have the dynamic information of the reformer. Rejection of other process variables are done based on low correlation coefficient value, for example the C1 compo; or the process variable is highly correlated with other process variables since there are some mathematical calculation performance to determine a new process variable, for example the ratio of FRCA1202 (natural gas feed) and FRCA1204 (total process steam) is calculated and represented by FFRA1208 (steam to carbon ratio).

RQE software is industrial software used for inferential modeling and online inferential measurement. The software is user friendly where the user is capable of learning the function of the software within a small time period. However, the software has very limited function. Hence, the user is constrained within the function of the software in creating a model. The software also performance mathematical analysis such as plotting input/output, correlation, cross correlation, error analysis and best input selection. Using RQE, Linear model and RBF model are developed. Based on the error analysis, Linear model is the best model developed using RQE.

Two types of toolbox are utilized in developing models using MATLAB software. Those are System Identification Toolbox and Neural Network Toolbox. The advantage of MATLAB software is that the user has freedom in developing techniques to obtain the best model. Long hours need to be spend to developed a model but the results will be promising if correct techniques are used during the model development stage.

ARX model are developed using System Identification Toolbox. The data are being pre process before model development by using the 'detrend' function. Detrending will remove mean values or linear trends from time-series and input/output signals. For multiple inputs and outputs, detrending removes trends independently from each signal. The model is developed based on trial and error technique by changing the number of poles and zeros for the difference equation. The number of delay is set to 0 since this is not a continues process and there is no time taken for the process variables to predict the TMT. ARX model structure with 2 poles (second order system) and 1 zero is the best model for both of the chamber. The different between the chamber's models are the coefficient values of the linear difference equation. MATLAB programming code is written to assist in developing ARX model. The developed ARX model is slightly better than the RQE Linear model.

Neural Network Toolbox is used in developing the Neural Network model. Neural Network has a very open architecture where the user has the freedom in designing in details the architecture. The neural network architecture chosen for this project is the multilayer feedforward which is commonly used with the backpropagation algorithm. Feedforward networks often have one or more hidden layers of sigmoid neurons followed by an output layer of linear neurons. Multiple layers of neurons with nonlinear transfer functions allow the network to learn nonlinear and linear relationships between input and output vectors. The linear output layer lets the network produce values outside the range 0 to 1.Training the neural network is the critical stage since it will determine the accuracy of the developed model. As shown in the tables earlier, the identify parameters (number of neuron, transfer function and training function) certainly has a high influence on the model error. The selection of number of neuron was done based on trial and error. High number of neuron can cause overfitting phenomena or poor model. 'Logsig' is taken as the optimum transfer function since it will cover the nonlinearity of the input. It was proven by the result obtained from table 9. The selected training function is 'traincgb'. This training function is from the category of fast algorithms that uses standard numerical optimization techniques called conjugate gradient. 'traincgb' is a network training function that updates weight and bias values according to the conjugate gradient backpropagation with Powell-Beale restarts. Since it is a fast algorthims, the number of epoch is a small value which shows that the training is completed in few cycles. Since model development is using trial and error approach, MATLAB programming code is written for neural network modeling. This code is capable of changing all the important parameters values and run the test in a short period of time.

Validation data is important in determining the accuracy of the developed model. This set of data has a new combination of input and output which have never been used during the training stage of the model. Any newly developed model should be validated with new set of data to determine its accuracy. The model is expected to have error which is within the acceptable range and almost similar with the training data error. RBF model is certainly not an acceptable model since the error for the validation data is extremely large in comparison with its training data error. Other models have acceptable error. Table 13 summarizes the error analysis for all the models using validation data. This information is crucial in determining the best model.

Table 13: Error Analysis for All Models Using Validation Data

Types of Model	Chamber 1		Chamber 2	
	RMSE	Index	RMSE	Index
Neural Network Model (MATLAB)	5.4625	38.11	5.2334	37.08
ARX Model (MATLAB)	6.3986	52.29	6.3489	54.58
Linear Model (RQE)	6.6268	56.09	6.4510	56.35
RBF Model (RQE)	9.5438	116.34	9.4765	121.59

From the error analysis, Neural Network model is undoubtedly the best model in predicting the reformer average TMT. This is followed by ARX model which has almost similar performance with the Linear model. RBF model is an unacceptable model. This result also shows that MATLAB software is capable of developing better model in comparison with the RQE software.

From the plotted graph for Neural Network model, the error is always within +/- 10°C. However there are few outliers in the graph. This could be due to bad set of data such as the process variables is not within the operating range. Other issues could be due to the error of the actual TMT data since the measurement of TMT is done manually by operator. Hence, there could be possibility of human error in getting the actual data.

Overall, critical process variables have been determined and these process variables are used for model development. MATLAB Neural Network Toolbox able to developed the best model in predicting the reformer TMT.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

Primary reformer is a critical component of a chemical plant and any failures on it will cause huge losses to the company. Most of the reformer failures are due to overheating; hence developing a model is essential to predict the reformer TMT. However, there is no known modeling techniques which is capable of the predicting the reformer TMT and applicable in industry for on-line monitoring of reformer TMT. Most of the reformer model uses rigorous mathematical equation which made the model not easily applicable in industry.

Important process variables that affect the reformer TMT are determine based on mathematical analysis and process understanding. Some process variables are eliminated since they are related to each other, which will lead to redundancy and model complexity. This stage is an important stage since it will determine the accuracy of the develop model. Since the error of the developed model is within the acceptable range, the first objective of the project which is selection of process variables is achieved.

With the availability of real-time data, empirical modeling technique is used to develop a predictive TMT model. This technique integrates the process variables values based on the structure of the model and will predict the TMT. Four types of model structure are developed using RQE and MATLAB. Second objective of the project is achieved by considering proper modeling techniques in developing each model and analyzing the models with error analysis. Neural Network model is the best model in comparison with other models. From the results obtained, MATLAB software has a better capability in developing a predictive model for reformer TMT which shows the success of completing the final objective of the project.

By integrating the developed model with the Distributed Control System (DCS), on-line monitoring of the reformer TMT can be achieved. The predicted operating temperature of reformer tubes should always be below the maximum design limit temperature, hence overheating cases can be reduced.

5.2 Recommendations

Since this is a project that is newly initiated, there is plenty of room for improvement on the development of the reformer TMT predictive model. The research done so far only shows that there is high possibility of developing a model that is capable of predicting a reformer TMT.

The impact of developing an accurate model will certainly save large amount of cost by reducing the tubes failures. Hence, industrial should invest on having more advance research in order to increase the approach in modeling a reformer TMT. The research should be done on the basis that it will be practically applicable in the industry.

Due to the complex behavior of the reformer, each reformer tube has a different temperature reading. Individual model predicting each tube temperature should be develop using Neural Network. Improvement on Neural Network model can be done by varying other parameters since Neural Network gives freedom in defining its model architecture. Other artificial intelligence software such as fuzzy logic can be considered for future model development.

The quality of data used in developing the model can be increase since it will determine the success of developing an accurate model. Some suggestions in increasing the quality of the data used for modeling are:

- Reduce sampling time of process variable to 1 minute instead an hour. This will increase the process variables data accuracy as an input for the model.

- TMT measurement accuracy should be increase by having proper procedure for the operator to manually collect the TMT readings. High accurate measurement device such as FLIR should be used in obtaining TMT reading instead of pyrometer which has low accuracy due to the surrounding disturbance and human error.
- Other critical process variables that affect the reformer TMT should be taken into consideration for future model development, such as (based on PASB available measurement):
  - PDI 1211 – This is the difference pressure between the inlet and outlet of reformer tubes. It contains information such as the flow of feed and chocking of catalyst in the reformer tubes.
  - PICA 1209 – This is the difference pressure between the inlet and outlet of Waste Heat Recovery Section (WHRS). This process variable shows how much heat are being sucked into the WHS. Higher suction will reduce the reformer temperature since more heat energy are send to WHRS.
  - Plant Load – This process variable will determine the status of the plant. Data with low plant load should be eliminated since it is not a steady state data.
  - Burner Status – This variable should be considered in developing individual tubes model. Failure of particular burner will disturb the overall heat distribution in the reformer. Hence, data obtained during the failure period should not be considered for model development.

GTS can consider the effectiveness of using Neural Network for inferential modeling in comparison with RQE software. Intellectual Property (IP) can be obtained by integrating MATLAB Neural Network Toolbox with RQE for online inferential measurement. The application of artificial intelligence, such as Neural Network which has potential of solving complex problem can be considered in solving other technology advancement problems faced by GTS.

REFERENCES

[1] John Brightling, Mar/Apr 2002, " Managing Steam Reformer Tubes," *ABI/INFORM Trade & Industry*, 256, pg. 29.

[2] Ashok Kumar Ray, Samarendra Kumar Sinha, Yogendra Nath Tiwari, Jagannathan Swaminathan, Gautam Das, Satyabrata Chaudhuri, Raghubir Singh, 2003, "Analysis of failed reformer tubes," *Engineering Failure Analysis*, 10, 351-362.

[3] Ammonia: The Next Step, retrieved on 8 August 2008 from http://www.cheresources.com/ammonia.shtml

[4] S.K Bhaumik, R.Rangaraju, M.A. Parameswara, T.A. Bhaskaran, M.A. Venkataswamy, A.C. Raghuram, R.V. Krishna, 2002, "Failure of reformer tube of an ammonia plant," *Engineering Failure Analysis*, 9, 553-561.

[5] S.A Jenabali Jahromi, M. Naghikhani, 2004, "Failure analysis of HP40-Nb modified primary reformer tube of an ammonia plant," *Iranian Journal of Science & Technology*, 28, 269-271.

[6] Tito Luiz da Silveira, Iain Le May, 2006, " Reformer Furnaces: Materials, Damage Mechanisms and Assessment," *The Arabian Journal for Science and Engineering*, 31, 99-119.

[7] Jaganathan Swaminathan, Krishna Guguloth, Manojkumar Gunjan, Prabirkumar Roy, Rabindranath Ghosh, 2008, "Failure analysis and remaining life assessment of service exposed primary reformer heater tubes," *Engineering Failure Analysis*, 15, 311-331.

[8] Zunhong Yu, Enhong Cao, Yifie Wang, Zhijie Zhou, Zhenghua Dai, 2006, "Simulation of natural gas steam reforming furnace," *Fuel Processing Technology*, 87, 695-704.

[9] Takahiro Kobayashi, Tetsuji Tani, Sadaaki Miyamoto, 2000, "Automation of Reformer Process in Petroleum Plant Using Fuzzy Supervisory and Model Predictive Multivariable Control System," *IEEE*, 1021-1024.

[10] Z. Sterjovski, D. Nolan, K.R. Carpenter, D.P. Dunne, J. Norrish, 2005, "Artificial Neural Networks for Modeling the Mechanical Properties of Steels in Various Applications," *Journal of Material Processing Technology*, 170, 536-544.

[11] M.H. Fazalul Rahiman M.N. Taib, Y. Md Salleh, 2006, "Black Box Modeling of Steam Temperature," 4th Student Conference on Research and Development, 178-182.

[12] Neural Network Toolbox for use with MATLAB; Howard Demuth and Mark Beale; The Mathworks Inc.

[13] Practical MATLAB Basics for Engineers; Misza Kalechman; Taylor & Francis Group.

[14] System Identification Toolbox for use with MATLAB; The Mathworks Inc.

[15] User Guide for RQE^{Pro}, Shell Global Solutions.

APPENDICES

APPENDIX 1 – Layout of Ammonia Plant

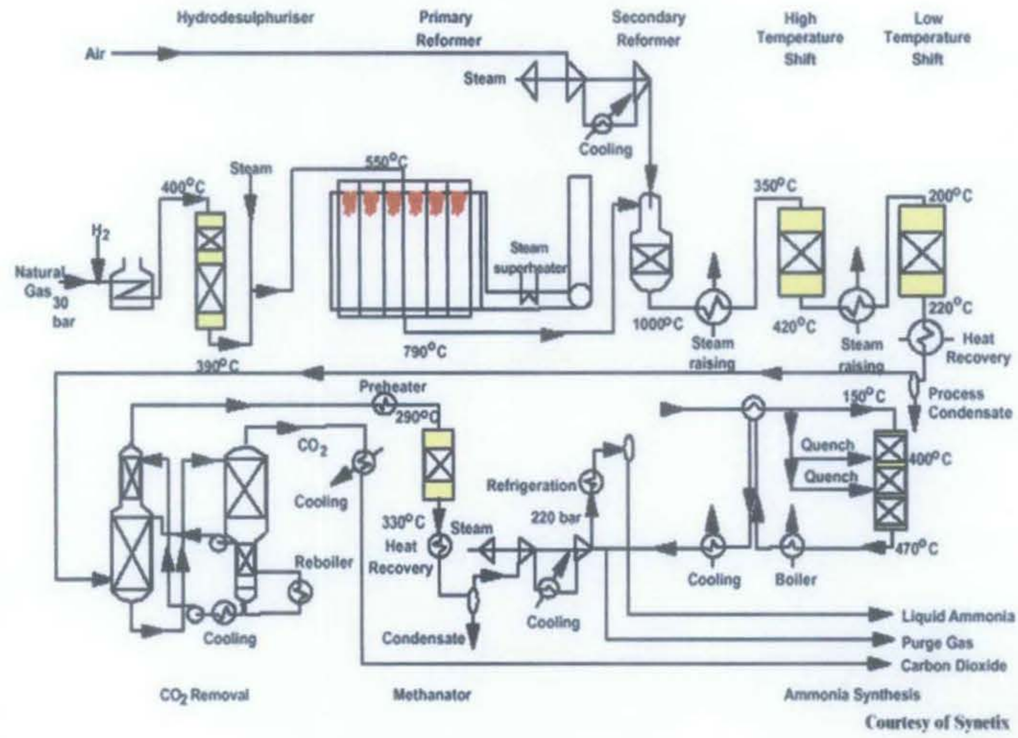

Figure 1.1: Layout of a Steam Reforming Plant for Ammonia Synthesis

Figure 1.2: Primary Reformer (Left) and Secondary Reformer (Right) of Ammonal Plant

APPENDIX 2 – Temperature Measurement of Primary Reformer Tubes

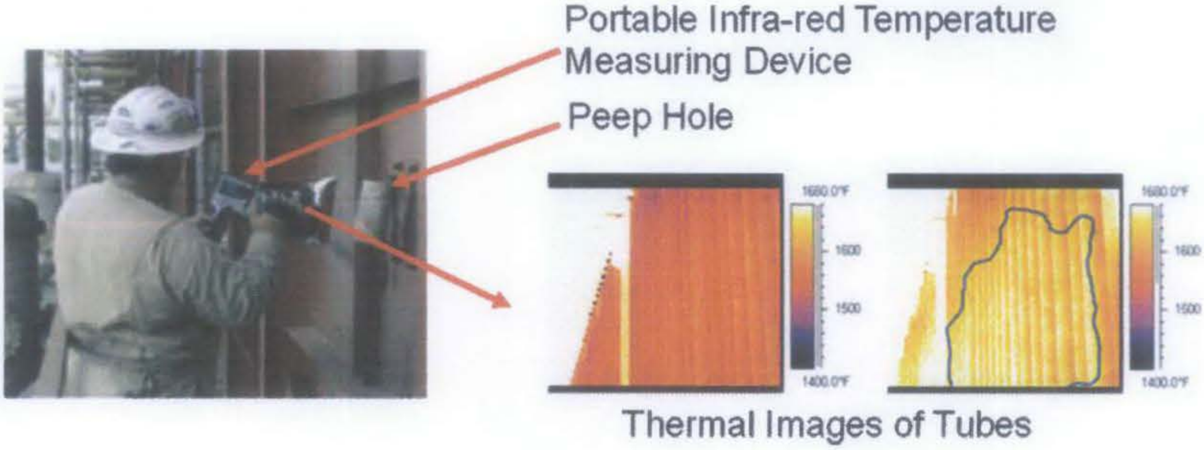

Figure 2.1: Temperature Measurement of Primary Reformer Tubes Using Forward Looking Infra-Red (FLIR) Measuring Device

APPENDIX 3 – Primary Reformer Tubes Failure

Figure 3.1: 100% Reformer Tubes Failure

Figure 3.2: Reformer Tubes Bending

Figure 3.3: Reformer Tubes Rupture

APPENDIX 4 – TMT Manual Data Entry Form

REFORMER (F3-1201) TUBES METAL TEMPERATURE SURVEY (TMT)

CHAMBER 1

Tube No.	LEVEL 1					LEVEL 3				
		Row A		Row B	delta T		Row A		Row B	delta T
1N		987		981	6		995		991	4
2N	W	980	W	978	2	W	993	W	990	3
3N		979		980	-1		990		990	0
4N	1026	973	1029	972	1	1069	990	1069	989	1
5N		958		956	2		995		990	5
6		973		977	-4		984		989	-5
7		978		984	-6		989		989	0
8		988		989	-1		989		993	-4
9		993		994	-1		995		995	0
10N	W	992	W	994	-2	W	995	W	995	0
11		994		995	-1		992		994	-2
12N	1026	986	1023	986	0	1122	995	1125	995	0
13N		978		977	1		994		994	0
14		984		989	-5		994		994	0
15N		993		995	-2		994		994	0
16		990		995	-5		991		991	0
17N		994		990	4		995		995	0
18	W	991	W	989	2	W	994	W	995	-1
19N		994		995	-1		993		992	1
20	1030	977	1023	975	2	1111	991	1131	994	-3
21N		976		974	2		995		993	2
22N		987		989	-2		989		985	4
23N		992		990	2		991		988	3

CHAMBER 2

Tube No.	LEVEL 1					LEVEL 3				
		Row C		Row D	delta T		Row C		Row D	delta T
145N		983		980	3		972		969	3
146N	W	972	W	971	1	W	976	W	975	1
147N		971		969	2		982		976	6
148N	1004	969	996	965	4	1141	975	1153	976	-1
149N		957		961	-4		965		968	-3
150N		967		970	-3		974		973	1
151N		974		975	-1		972		969	3
152		979		979	0		967		972	-5
153N		977		982	-5		968		970	-2
154N	W	977	W	979	-2	W	985	W	980	5
155N		974		974	0		972		970	2
156	1002	959	1003	964	-5	1010	968	1012	966	2
157N		954		949	5		965		960	5
158		980		981	-1		964		961	3
159N		971		976	-5		970		973	-3
160		975		976	-1		958		956	2
161N		988		982	-2		979		980	-1
162N	W	978	W	982	-4	W	964	W	966	-2
163N		980		983	-3		969		963	6
164N	1008	974	1012	972	2	1082	977	1075	979	-2
165		969		969	0		970		971	-1
166N		980		986	-6		984		980	4
167N		982		988	-6		986		988	-2

Figure 4.1: TMT Data Sheet

APPENDIX 5 – Project Milestone for Final Year Project (FYP)

Table 5.1: Milestone for First Semester

No.	Detail/ Week	1	2	3	4	5	6	7	8	9	10		11	12	13	14	15
1	Selection of Project Topic											Mid-semester break					
	Meet supervisor to discuss regarding the topic																
	Submission of project title																
2	Preliminary Research																
	Understand the process of ammonia plant																
	Download journals and conference papers regarding the current research																
	Read selected journal																
	Submission of literature review to GTS																
3	Preliminary Report Submission																
4	Seminar 1 (Optional)																
5	Data Collection																
	Obtain preliminary data from PASB																
	Real-time data collection at PASB (from October to November)																
6	Preliminary Modeling																
	PASB pre-data compilation in Excel document																
	Learning system identification toolbox																
	Use pre-data to develop a model in system identification																
7	Progress Report Submission																

8	Seminar 2 (Compulsory)																
9	Mathematical Modeling using System Identification Toolbox																
	Mastering system identification toolbox																
	Develop source code																
	Analyzing and improving the best model																
10	Interim Report Submission																
11	Oral Presentation																

Table 5.2: Milestone for Second Semester

No.	Detail/ Week	1	2	3	4	5	6	7	8	9		10	11	12	13	14
1	Modeling (RQE)										Mid-semester break					
	Leaning of RQE software															
	Model development and validation															
2	Progress Report 1															
3	Modeling (Neural Network & RQE)															
	Leaning of neural network toolbox															
	Develop source code															
4	Progress Report 2															
5	Seminar															
6	Modeling (Neural Network)															
	Model development															
	Model improvement															
	Error analysis of all the model															
7	Poster Exhibition															
8	Dissertation Report (Soft Bound)															
9	Oral Presentation															
10	Dissertation Report (Hard Bound)															

APPENDIX 6 – Mathematical Equation for RQE Linear Model

Table 6.1: Values for Linear Model (Chamber 1)

Input	Weight (a)
FFRA1208	-9.0787
FIA1216	-4.30E-07
FICA1251	0.0061664
PICA1253	188.57
PR1210	0.57501
QIA1204	5.6514
TI1220	0.062404
TIA1212	-0.21760
TIA1234	0.029003
TRA1232	0.74255
Bias	314.76

Table 6.2: Values for Linear Model (Chamber 2)

Input	Weight (a)
FFRA1208	-33.517
FIA1216	0.00012632
FICA1251	-0.018442
PICA1253	160.52
PR1210	2.3229
QIA1204	3.8655
TI1220	-0.44701
TIA1212	-0.41292
TIA1234	0.063439
TRA1232	0.61180
Bias	666.13

APPENDIX 7 – Mathematical Equation for RQE RBF Model

Values for RBF Model (Chamber 1)

Width Values: 2437.4

Table 7.1: RBF Nodes (Chamber 1)

Inputs	Center 1	Center 2	Center 3	Center 4	Center 5	Center 6	Center 7
FFRA1208	1.5087	1.7525	1.6852	1.6153	1.5333	1.5058	1.7103
FIA1216	2.0844E+05	1.7365E+05	1.7944E+05	1.9469E+05	1.9082E+05	2.0101E+05	1.7941E+05
FICA1251	826.61	559.79	640.13	699.16	684.76	784.26	622.98
PICA1253	0.23752	0.19171	0.19408	0.16642	0.22656	0.22949	0.21175
PR1210	22.812	22.308	22.339	22.238	22.731	22.500	22.258
QIA1204	4.8468	4.7707	4.8002	4.9045	5.2329	5.1790	4.7514
TI1220	261.38	256.44	258.68	257.58	259.48	260.71	259.40
TIA1212	659.48	628.39	636.31	648.38	639.82	656.43	628.40
TIA1234	1084.7	871.84	1023.2	1093.5	936.06	1116.9	926.23
TRA1232	919.31	893.64	897.79	899.34	899.41	905.33	898.39

Table 7.2: MLR Thetas (Chamber 1)

Node	Theta
1	55187.
2	-81397.
3	-9.7077E+06
4	-64470.
5	69378.
6	31682.
7	9.7605E+06
Bias	951.83

Values for RBF Model (Chamber 2)

Width Values: 641.42

Table 7.3: RBF Nodes (Chamber 2)

Inputs	Center 1	Center 2	Center 3	Center 4	Center 5	Center 6	Center 7	Center 8	Center 9	Center 10
FFRA1208	1.7525	1.5118	1.7581	1.6153	1.746	1.7376	1.6145	1.7137	1.6696	1.5536
FIA1216	1.74E+05	2.09E+05	1.69E+05	1.95E+05	1.59E+05	1.75E+05	1.79E+05	1.82E+05	1.98E+05	1.98E+05
FICA1251	559.79	765.86	631.54	699.16	556.53	618.83	638.22	616.81	643.72	755.86
PICA1253	0.19171	0.22098	0.15727	0.16642	0.18232	0.19279	0.20915	0.17552	0.1809	0.20701
PR1210	22.308	22.957	22.084	22.238	22.03	22.383	22.408	22.427	22.67	22.439
QIA1204	4.7707	4.8734	4.6479	4.9045	4.6679	4.8558	4.5936	5.02	4.3942	5.0071
TI1220	256.44	260.25	257.41	257.58	256.79	258.35	259.92	254.77	260.81	260.41
TIA1212	628.39	660.44	619.4	648.38	628.6	627.08	632.47	624.18	644.73	654.83
TIA1234	871.84	1124.1	974.72	1093.5	974.64	885.38	939.95	894.96	951.25	1109.8
TRA1232	893.64	918.69	892.14	899.34	902.41	891.42	911.07	893.61	908.18	903.88

Table 7.4: MLR Thetas (Chamber 2)

Node	Theta
1	-14976
2	5130.5
3	-13822
4	-18580
5	-17928
6	-10859
7	11440
8	-10363
9	-2.32E+05
10	2.24E+05
Bias	976.1

APPENDIX 8 – MATLAB Programming Code for ARX Model

```
%ARX model
%Program written by Vishal Nanji Patel

%clear workspace and command window
clear;
clc;

%load all the input and output
x = load ('all_input.txt'); %load the INPUT data
y = load ('all_output.txt'); %load the OUTPUT data
load ('ARX_model'); %load model saved in workspace

%get the number of input and number of data
train_data = 100; %number of TRAINING data
validation_data = 100; %number of VALIDATION data
numofvar = size(x,2); %number of input
numofout = size(y,2); %number of output

%devide the data into training and validation data
for m=1:numofvar
    for n=1:train_data
    x_t(n,m)=x(n,m); %training input data
    end
end


for m=1:numofvar
    for n=1:validation_data
    x_v(n,m)=x(n+train_data,m); %validation input data
    end
end

for m=1:numofout
    for n=1:train_data
    y_t(n,m)=y(n,m); %training output data
    end
end

for m=1:numofout
    for n=1:validation_data
    y_v(n,m)=y(n+train_data,m); %validation output data
    end
end

%remove trend and remove mean (detrend)
x_t1 = detrend(x_t); %detrend the TRAINING INPUT data
y_t1 = detrend(y_t); %detrend the TRAINING OUTPUT data
x_v1 = detrend(x_v); %detrend the VALIDATION INPUT data
y_v1 = detrend(y_v); %detrend the VALIDATION OUTPUT data

%find the detrend value, this value will be used to trend back the
final output
```

```
xt_0 = x_t-x_t1;
yt_0 = y_t-y_t1;
xv_0 = x_v-x_v1;
yv_0 = y_v-y_v1;

%simulate the network with TRAINING data
ytrain = sim(chamber1,x_t1); %simulate the model
ytrain1=ytrain+yt_0; %trend the output
etrain=y_t-ytrain1; %calculate the different between the actual and
predicted temperature value

%simulate the network with VALIDATION data
yvalid=sim(chamber1,x_v1); %simulate the model
yvalid1=yvalid+yv_0; %trend the output
evalid=y_v-yvalid1; %calculate the different between the actual and
predicted temperature value

%plot the actual and predicted TMT from VALIDATION data
subplot(2,2,1);
plot (yvalid1,'r');
hold on;
plot (y_v,'b');
xlabel('Set of Data');
ylabel('Tubes Average Temperature (degC)');
title('Ouput of ARX model for Chamber 1 (Validation Data)');
legend('Predicted Temperature','Actual Temperature');
grid on;

%plot the different between the actual and predicted TMT from
VALIDATION data
subplot(2,2,2);
plot(evalid,'*');
xlabel('Set of Data');
ylabel('Error (degC)');
title('Error between Actual Temperature and Predicted Temperature for
Chamber 1 (Validation Data)');
grid on;

%plot the actual and predicted TMT from TRAINING data
subplot(2,2,3);
plot (ytrain1,'r');
hold on;
plot (y_t,'b');
xlabel('Set of Data');
ylabel('Tubes Average Temperature (degC)');
title('Output of ARX model for Chamber 1 (Training Data)');
legend('Predicted Temperature','Actual Temperature');
grid on;

%plot the different between the actual and predicted TMT from TRAINING
data
subplot(2,2,4);
plot(etrain,'*');
xlabel('Set of Data');
ylabel('Error (degC)');
```

```
title('Error between Actual Temperature and Predicted Temperature for
Chamber 1 (Training Data)');
grid on;

%error analysis for the VALIDATION data
rmse_valid = sqrt(mse(evalid)) %mean square error
index_valid = (sum((evalid).^2)/sum((y_v-mean(y_v)).^2))*100 %index
value

%error analysis for the TRAINING data
rmse_train = sqrt(mse(etrain)) %mean square error
index_train = (sum((etrain).^2)/sum((y_t-mean(y_t)).^2))*100 %index
value
```

APPENDIX 9 – Mathematical Equation for MATLAB ARX Model

Values for ARX Model (Chamber 1)

na = [2]

nb = [1 1 1 1 1 1 1 1 1 1]

nk = [0 0 0 0 0 0 0 0 0 0]

Discrete-time IDPOLY model: A(q)y(t) = B(q)u(t) + e(t)

Table 9.1: Parameters Value (Chamber 1)

Tag No.	Parameter	Value
Chamber 1	A(q)	1 - 0.3564 q^-1 + 0.003297 q^-2
TI1220.PV	B1(q)	-0.1468
TIA1212.PV	B2(q)	0.04853
PR1210.PV	B3(q)	-0.9219
FIA1216.PV	B4(q)	-0.0002295
PICA1253.PV	B5(q)	201.9
FFRA1208.PV	B6(q)	-72.75
FICA1251.PV	B7(q)	-0.009947
QIA1204.PV	B8(q)	-3.038
TRA1232.PV	B9(q)	0.05667
TIA1234.PV	B10(q)	0.001336

Values for ARX Model (Chamber 2)

na = [2]

nb = [1 1 1 1 1 1 1 1 1 1]

nk = [0 0 0 0 0 0 0 0 0 0]

Discrete-time IDPOLY model: A(q)y(t) = B(q)u(t) + e(t)

Table 9.2: Parameters Value (Chamber 2)

Tag No.	Parameter	Value
Chamber 1	A(q)	1 - 0.4436 q^-1 + 0.1051 q^-2
TI1220.PV	B1(q)	-0.8091
TIA1212.PV	B2(q)	-0.1715
PR1210.PV	B3(q)	2.77
FIA1216.PV	B4(q)	-0.0001492
PICA1253.PV	B5(q)	170
FFRA1208.PV	B6(q)	-77.62
FICA1251.PV	B7(q)	-0.02792
QIA1204.PV	B8(q)	-1.738
TRA1232.PV	B9(q)	0.1587
TIA1234.PV	B10(q)	0.03351

APPENDIX 10 – MATLAB Programming Code for Neural Network Model

```
%Neural Network model
%2 layer network
%Normalization from 0 to 1
%Program written by Vishal Nanji Patel

% Clear workspace and command window
clear;
clc;

%load all the input and output
x = load ('all_input.txt')'; %load the INPUT data
y = load ('all_output.txt')'; %load the OUTPUT data


%get the number of input and number of data
train_data = 100; %number of TRAINING data
validation_data = 100; %number of VALIDATION data
numofvar = size(x,1); %number of input
numofout = size(y,1); %number of input

%devide the data into training and validation data
for m=1:numofvar
    for n=1:train_data
    x_t(m,n)=x(m,n);
    end
end


for m=1:numofvar
    for n=1:validation_data
    x_v(m,n)=x(m,n+train_data);
    end
end

for m=1:numofout
    for n=1:train_data
    y_t(m,n)=y(m,n);
    end
end

for m=1:numofout
    for n=1:validation_data
    y_v(m,n)=y(m,n+train_data);
    end
end

%normalize the input and output from 0-1

%normalize the TRAINING INPUT data
```

```
for row = 1:numofvar
    x_t1(row,:)=((1/max(x_t(row,:)-min(x_t(row,:)))).*(x_t(row,:)-
(min(x_t(row,:)))));
end

%normalize the VALIDATION INPUT data
for row = 1:numofvar
    x_v1(row,:)=((1/max(x_v(row,:)-min(x_v(row,:)))).*(x_v(row,:)-
(min(x_v(row,:)))));
end

%normalize the TRAINING OUTPUT data
y_t1=((1/(max(y_t)-min(y_t)))*(y_t-min(y_t)));

%normalize the VALIDATION OUTPUT data
y_v1=((1/(max(y_v)-min(y_v)))*(y_v-min(y_v)));

%minimum and maximum value for the training data after normalization
(should be 0 and 1)
t = minmax(x_t1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%number of neurons for layer 1 and layer 2
neuron_1 = 18; %number of neurons for layer 1
neuron_2 = 1; %number of neurons for layer 2

%network and parameters
net=newff(t,[neuron_1,neuron_2],{'logsig','purelin'},'traincgb');
net.trainParam.show = 50;
net.trainParam.lr = 0.2;
net.trainParam.epochs = 1000;
net.trainParam.goal = 0.01;
net=init(net);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%set the weights and biases to 0

%set the weights for 1st layer to 0
for m=1:neuron_1
    for n=1:numofvar
    w_1(m,n)=0;
    end
end
net.IW{1,1}=w_1;

%set the weights for 2nd layer to 0
for m=1:neuron_2
    for n=1:neuron_1
    w_2(m,n)=0;
    end
end
net.LW{2,1}=w_2;
```

```
%set the bias for 1st layer to 0
for m=1:neuron_1
    b_1(m,1)=0;
end
net.b{1}=b_1;

%set the bias for 2nd layer to 0
for m=1:neuron_2
    b_2(m,1)=0;
end
net.b{2}=b_2;

%checking the weights and biases (make sure all are 0)
net.IW{1,1}; %weights of 1st layer
net.LW{2,1}; %weights of 2nd layer
net.b{1}; %bias of 1st layer
net.b{2}; %bias of 2nd layer

%train the network
[net,tr]=train(net,x_t1,y_t1);

%simulate the network with TRAINING data
ytrain=sim(net,x_t1); %simulate the network
test1=mse(ytrain-y_t1)
ytrain1 =(((max(y_t)-min(y_t)))*(ytrain))+min(y_t); %denormalize the
output
etrain=y_t-ytrain1; %calculate the different between the actual and
predicted temperature value

%simulate the network with VALIDATION data
yvalid=sim(net,x_v1); %simulate the network
yvalid1 =(((max(y_v)-min(y_v)))*(yvalid))+min(y_v); %denormalize the
output
evalid=y_v-yvalid1; %calculate the different between the actual and
predicted temperature value

yvalid1=yvalid1';
ytrain1=ytrain1';

%plot the actual and predicted TMT from VALIDATION data
subplot(2,2,1);
plot (yvalid1','r');
hold on;
plot (y_v,'b');
xlabel('Set of Data');
ylabel('Tubes Average Temperature (degC)');
title('Output of NN model for Chamber 1 (Validation Data)');
legend('Predicted Temperature','Actual Temperature');
grid on;

%plot the different between the actual and predicted TMT from
VALIDATION data
subplot(2,2,2);
plot(evalid,'*');
xlabel('Set of Data');
```

```
ylabel('Error (degC)');
title('Error between Actual Temperature and Predicted Temperature for
Chamber 1 (Validation Data)');
grid on;

%plot the actual and predicted TMT from TRAINING data
subplot(2,2,3);
plot (ytrain1','r');
hold on;
plot (y_t,'b');
xlabel('Set of Data');
ylabel('Tubes Average Temperature (degC)');
title('Output of NN model for Chamber 1 (Training Data)');
legend('Predicted Temperature','Actual Temperature');
grid on;

%plot the different between the actual and predicted TMT from TRAINING
data
subplot(2,2,4);
plot(etrain,'*');
xlabel('set of Data');
ylabel('Error (degC)');
title('Error between Actual Temperature and Predicted Temperature for
Chamber 1 (Training Data)');
grid on;

%error analysis for the VALIDATION data
rmse_valid = sqrt(mse(evalid)) %mean square error
index_valid = (sum((evalid).^2)/sum((y_v-mean(y_v)).^2))*100 %index
value

%error analysis for the TRAINING data
rmse_train = sqrt(mse(etrain)) %mean square error
index_train = (sum((etrain).^2)/sum((y_t-mean(y_t)).^2))*100 %index
value
```

APPENDIX 11 – Mathematical Equation for MATLAB Neural Network Model

<u>Values for Neural Network Model (Chamber 1)</u>

Table 11.1: Input Weight (Chamber 1)

Neuron	TI 1220	TIA 1212	PR 1210	FIA 1216	PICA 1253	FFRA 1208	FICA 1251	QIA 1204	TRA 1232	TI 123
1 to 18	0.104	0.298	0.271	0.283	0.209	-0.337	0.271	0.043	0.373	0.2

Input Weight is a {18x10} matrix

Table 11.2: Layer Weight (Chamber 1)

Neuron (Output)	Node (1 to 18)
Chamber 1	0.1204

Layer Weight is a {1x18} matrix

Table 11.3: Input Bias (Chamber 1)

Neuron	Bias
1 to 18	-0.0370

Input Bias is a {18x1} matrix

Table 11.4: Layer Bias (Chamber 1)

Neuron (Output)	Bias
Chamber 1	-1.0122

Layer Bias is a {1x1} matrix

Values for Neural Network Model (Chamber 2)

Table 11.5: Input Weight (Chamber 2)

Neuron	TI 1220	TIA 1212	PR 1210	FIA 1216	PICA 1253	FFRA 1208	FICA 1251	QIA 1204	TRA 1232	TI 12
1 to 13	-0.062	-0.021	0.086	0.019	0.211	-0.263	0.111	-0.105	0.342	0.2

Input Weight is a {13x10} matrix

Table 11.6: Layer Weight (Chamber 2)

Neuron (Output)	Node (1 to 13)
Chamber 2	0.2322

Layer Weight is a {1x13} matrix

Table 11.7: Input Bias (Chamber 2)

Neuron	Bias
1 to 13	-0.0606

Input Bias is a {13x1} matrix

Table 11.8: Layer Bias (Chamber 2)

Neuron (Output)	Bias
Chamber 2	-1.1797

Layer Bias is a {1x1} matrix

Values for Neural Network Model (Chamber 2)

Table 11.5: Input Weight (Chamber 2)

Neuron	TI 1220	TIA 1212	PR 1210	FIA 1216	PICA 1253	PERA 1208	FICA 1251	QIA 1203	TRA 1232	TI 12
1 to 13	-0.062	-0.021	0.086	0.019	0.211	-0.263	0.111	-0.105	0.342	0.24

Input Weight is a {13x10} matrix

Table 11.6: Layer Weight (Chamber 2)

Neuron (Output)	Node (1 to 13)
Chamber 2	0.2322

Layer Weight is a {1x13} matrix

Table 11.7: Input Bias (Chamber 2)

Neuron	Bias
1 to 13	-0.0606

Input Bias is a {13x1} matrix

Table 11.8: Layer Bias (Chamber 2)

Neuron (Output)	Bias
Chamber 2	-1.1797

Layer Bias is a {1x1} matrix