

**MODELING OF PRIMARY REFORMER TUBE METAL TEMPERATURE  
(TMT) USING LS-SVM**

By

YAP WI SON

FINAL PROJECT REPORT

Submitted to the Department of Electrical & Electronic Engineering  
in Partial Fulfillment of the Requirements  
for the Degree  
Bachelor of Engineering (Hons)  
(Electrical & Electronic Engineering)

Universiti Teknologi PETRONAS

Bandar Seri Iskandar

31750 Tronoh

Perak Darul Ridzuan

© Copyright 2012

by

Yap Wi Son, 2012

# **CERTIFICATION OF APPROVAL**

## **MODELING OF PRIMARY REFORMER TUBE METAL TEMPERATURE (TMT) USING LS-SVM**

by

Yap Wi Son

A project dissertation submitted to the  
Department of Electrical & Electronic Engineering  
Universiti Teknologi PETRONAS  
in partial fulfilment of the requirement for the  
Bachelor of Engineering (Hons)  
(Electrical & Electronic Engineering)

Approved:

---

Dr Rosdiazli b. Ibrahim  
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS  
TRONOH, PERAK

August 2012

## **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.

---

Yap Wi Son

## ABSTRACT

This report discusses the research done on the chosen topic, which is Modeling of Primary Reformer Tube Metal Temperature (TMT) using LS-SVM. The objective of the project is to develop a model that can predict the temperature of the reformer tubes. The scope of the study focused on the modeling of the primary reformer TMT of PETRONAS Ammonia hydrocarbons such as natural gas into its constituents which are carbon dioxide, carbon monoxide and hydrogen. Pressurized feed (300barg) 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 the 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 real-time data obtained from PASB plant. LS-SVM is used in developing the model and Back Propagation Neural Network is used to develop a model that serves as the benchmark for this project.

## **ACKNOWLEDGEMENT**

Firstly, the author would like to express his utmost gratitude to God for His endless blessings and the strength He gifted upon the author to complete this Final Year Project.

The author would also like to express his greatest appreciation and thankfulness to his Final Year Project supervisor Dr. Rosdiazli b. Ibrahim for his time and effort in guiding, supporting and motivating the author despite his many other obligations. The author would also like to extend his gratitude to Mr. ElangeshwaranPathmanathan for his guidance, advice and sharing of his valuable knowledge during the completion of this research project.

Last but not least, the author would like to thank this family, friends and anyone else who contributed in a way or another to the successful of this Final Year Project.

## TABLE OF CONTENTS

<b>CERTIFICATION OF APPROVAL</b>	.	.	.	.	<b>i</b>
<b>CERTIFICATION OF ORIGINALITY</b>	.	.	.	.	<b>ii</b>
<b>ABSTRACT</b>	.	.	.	.	<b>iii</b>
<b>ACKNOWLEDGEMENT</b>	.	.	.	.	<b>iv</b>
<b>TABLE OF CONTENTS</b>	.	.	.	.	<b>v</b>
<b>LIST OF FIGURES</b>	.	.	.	.	<b>vii</b>
<b>LIST OF TABLES</b>	.	.	.	.	<b>viii</b>
<b>CHAPTER 1:</b>					
<b>INTRODUCTION</b>	.	.	.	.	<b>1</b>
1.1.	Project Background	.	.	.	1
1.2.	Problem Statements	.	.	.	4
1.3.	Objectives	.	.	.	5
1.4	Scope of Study	.	.	.	5
1.5	Relevance of Project	.	.	.	6
1.6	Feasibility of Project	.	.	.	6
<b>CHAPTER 2:</b>					
<b>LITERATURE REVIEW</b>	.	.	.	.	<b>7</b>
2.1	PASB Primary Reformer	.	.	.	7
2.2.	Modeling technique of Reformer	.	.	.	8
2.2.1	Artificial Neural Network	.	.	.	8
2.2.2	SVM	.	.	.	10
2.2.3	LS-SVM	.	.	.	12

<b>CHAPTER 3:</b>	<b>METHODOLOGY . . . . .</b>	<b>15</b>
	3.1. Research Methodology . . . . .	15
	3.2. Project Activities . . . . .	15
	3.3. Tool Required . . . . .	17
	3.5. Gantt Chart . . . . .	18
<b>CHAPTER 4:</b>	<b>RESULTS AND ANALYSIS. . . . .</b>	<b>20</b>
	4.1. Input and Output Selection . . . . .	20
	4.2. Modeling Background . . . . .	22
	4.3. Results and Analysis . . . . .	24
	4.3.1 BPNN Model for Chamber 1 . . . . .	24
	4.3.2 LS-SVM Model for Chamber 1 . . . . .	25
	4.3.3 BPNN Model for Chamber 2 . . . . .	27
	4.3.4 LS-SVM Model for Chamber 2 . . . . .	29
<b>CHAPTER 5:</b>	<b>CONCLUSION AND RECOMMENDATION</b>	<b>31</b>
	5.1 Conclusion . . . . .	31
	5.2 Recommendation. . . . .	32
<b>REFERENCES</b>	. . . . .	<b>34</b>
<b>APPENDIX .</b>	. . . . .	<b>36</b>

## LIST OF FIGURES

Figure 1: Photograph of failed tubes . . . . .	1
Figure 2: Photograph of failed tubes . . . . .	1
Figure 3: TMT profile vs. creep damage . . . . .	2
Figure 4: Tube metal temperature readings using pyrometer . . . . .	3
Figure 5: Typical Feedforward BP Network . . . . .	9
Figure 6: $\varepsilon$ -sensitive loss function . . . . .	11
Figure 7: Overall Project Activities Flow . . . . .	16
Figure 8: The arrangement of tubes in the PASB reformer . . . . .	20
Figure 9: Nelder-Mead LS-SVM flowchart . . . . .	23
Figure 10: Chamber 1 training data for BPNN . . . . .	25
Figure 11: Chamber 1 testing data for BPNN . . . . .	25
Figure 12: Chamber 1 training data for LS-SVM . . . . .	27
Figure 13: Chamber 1 testing data for LS-SVM . . . . .	27
Figure 14: Chamber 2 training data for BPNN . . . . .	28
Figure 15: Chamber 2 testing data for BPNN . . . . .	29
Figure 16: Chamber 2 training data for LS-SVM . . . . .	30
Figure 17: Chamber 1 testing data for LS-SVM . . . . .	30

## LIST OF TABLES

Table 1: Gantt chart for FYP I	18
Table 2: Gantt chart for FYP II	19
Table 3: List of Selected Process Variables	21
Table 4: Parameters for BPNN Model for Chamber 1	24
Table 5: Performance of BPNN model for Chamber 1	24
Table 6: Hyperparameter optimization steps for Chamber 1	26
Table 7: Performance of LS-SVM model for Chamber 1	26
Table 8: Parameters for BPNN Model for Chamber 2	28
Table 9: Performance of BPNN model for Chamber 2	28
Table 10: Hyperparameter optimization steps for Chamber 2	29
Table 11: Performance of LS-SVM model for Chamber 2	30

## CHAPTER 1

### INTRODUCTION

#### 1.1 Project Background

Reformer is the largest and most expensive piece of equipment used in ammonia, methanol, or hydrogen plants [1]. Steam reforming is the most widespread process and an economical route to produce synthesis gas from light hydrocarbon. The process is a very endothermic process and it occurs in catalyst-filled tubes.



Figure 1: Photograph of failed tubes



Figure 2: Photograph of failed tubes

The thermal stress due to overheating is amongst the main factor contributing to the failure of tubes. The temperature of the tube metal (also called tube wall) influences the tube life as when overheating occurs; it causes dramatic reduction in the tube life. The rule of thumb is that a rise in operating temperature of only 20°C will approximately halve the remaining operating life of a tube [2]. Tube failure should therefore be avoided as it will lead to equipment damages, and unscheduled plant downtime that could result to losses.

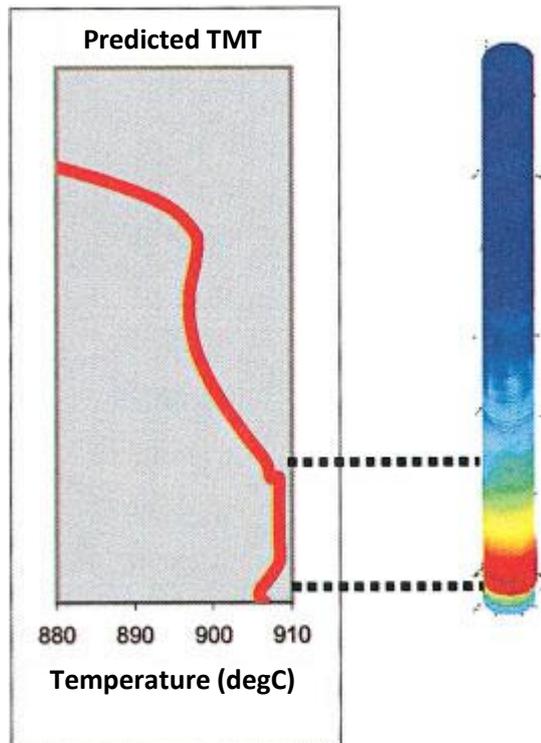


Figure 3: TMT profile vs. creep damage (graph vertical scale = tube height)

Currently there is no online monitoring system to directly monitor temperature of the reformer tubes in real-time. This is due to the high operating temperature range (about 800-1000°C) as well as its expensive cost. The current practice done by operators in oil and gas industries in order to prevent failure is

by manually measuring the outer tube metal temperature (TMT) using pyrometer through peepholes.

However, this imposes problems regarding the personnel health and safety because of the high ambient temperature that reaches  $40^{\circ}\text{C}$  as well as the human error that affects the accuracy of the measurements [3]. The other common precautionary act to prevent tube failure is to operate the reformer below the designed temperature limit, but the issue of using this method is that the production of the plant cannot be optimized.

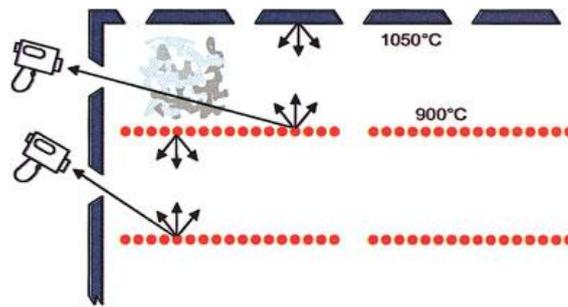


Figure 4: Tube metal temperature readings using pyrometer

In order to overcome these problems, it is vital to have a predicting tool that can predict the output temperature of the reformer tube. In realizing this objective, empirical modeling which is also called black-box model can be used to provide an adequate tube metal temperature prediction. The development of the model focuses on modeling using LS-SVM, but the Artificial Neural Network is also used to compare the performances.

## 1.2 Problem Statement

The common failure of reformers is due to overheating cases where the reformer is operating beyond the TMT design limit. Overheating of the tubes for a short period of time can be extremely dangerous to the tube life [1].

The factors that affect overheating of the reformers tubes are:

- Restricted process flow due to catalyst choking.
- Increasing temperatures to maintain production yield when catalyst life decreases.
- Feedstock or steam failures.

Due to operating the reformer at very high temperature, there is no continuous online measurement technology that can monitor the temperature of the reformer tubes. Current practice of preventing the premature tubes failure is operating the reformer below its TMT design limit. This is achieved by controlling the reformer outlet temperature. However, such method is not being able to give the actual indication of TMT measurement. Hence, operator will be using a manual method in measuring the TMT. This is done by opening the peep hole and manually measuring the TMT using pyrometer or forward looking infra-red (FLIR) devices. This method is carried out occasionally due to concern regarding the health and safety issues of the personal taking the readings since the ambient temperature around the reformer is very hot (around 40°C). Hence, there is a need to have a model that can accurately predict the reformer TMT to ensure the reformer tubes are operating at optimum level but within the design limit. This project will also decrease the premature tubes failure which leads to huge losses in term of damage to the equipment, production losses and safety hazard [4].

### **1.3 Objectives**

The objectives of this project are:

- To identify important process variables in predicting the reformer TMT.
- To build and analyse a model that is able to predict the temperature of the reformer tubes.
- To compare model developed using LS-SVM with Back Propagation Neural Network.

### **1.4 Scope of Study**

The scope of study focused on modelling of reformer TMT. The model will be developed by integrating the critical process variables using LS-SVM. PASB, Kertih will be used as a pilot plant. Real-time data from PASB plant was collected to be used throughout the project.

The model should be able to predict the behaviour of the reformer TMT so that we can control the temperature of the tubes. The selected approach to construct the model is using LS-SVM. LS-SVM will learn the relationship between input parameters and variables by studying the previous recorded data. The output will be predicted by the model based on the trained data earlier for other input.

### **1.5 Relevance of Project**

The success of this project will determine a model that can predict the reformer TMT and the critical process variables in controlling the temperature of tubes. In return, it will increase the life-time of tubes by reducing the premature failures due to overheating where the reformer tubes are being operated below the maximum design limit temperature. Operating the reformer at optimum level and within the TMT design limit will also help in plant optimization since an increase of 10°C in reformer temperature will increase the production by 3%.

### **1.6 Feasibility of Project**

The model developed is based on the actual situation in PASB, Kertih. The idea of the investigation is to find the best modelling technique for the primary reformer TMT. According to the literature review done, LS-SVM seems to be a suitable solution for the project. This project is scheduled to be completed within two semesters. As long as full commitment is given to get the most optimum results, this project should be able to be completed within the scope and timeline.

## CHAPTER 2

### LITERATURE REVIEW

#### 2.1 PASB Primary Reformer

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. Only level 1 and 3 are accessible for the operator to measure the tubes temperature. The tubes temperature readings obtained 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 the PASB plants there are 3 primary reformer tubes failures in the past 7 years.

These failures have caused huge losses to PASB:

- RM50,000 for replacement cost per tube – RM14.4 million for all 288 tubes.
- RM20,000 million per catalyst batch
- RM30 million production loss.

## **2.2 Modeling Technique of Reformer**

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 type of modeling is not applicable in industry to measure the reformer TMT.

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. In this case, the empirical modeling technique used are LS-SVM and Back Propagation Neural Network (which serves as benchmark).

### **2.2.1 Artificial Neural Network**

Inspired by the biological nervous systems, Artificial Neural Network (ANN) is a type of mathematical model that is represented as non-linear interconnected layers of processing nodes which are normally referred to as neurons, a term borrowed from neurobiology[5].

ANN consists of a group of interconnected neurons which operate in parallel. Neurons work by processing information. They receive information from input and process them. ANN can be trained to perform complex functions in various fields of applications.

Up to this date, many network architectures and learning techniques has been developed and applied in different fields. One of the most commonly used algorithms is the back-propagation (BP) algorithm. BP network is actually type of multi-layer feedforward neural networks where information flows from the input layer to output layer.

BP network learning is divided into two stages: the first is to enter a known learning samples, by setting the network structure and the previous iteration of the right value and the threshold value, from the network's first layer of backward calculation of the output of each neuron; then weight and threshold are modified to move from the last layer of the right to calculate their value and the threshold value on the overall impact of the error, the error transfers back layer by layer, whereby pairs of weights and thresholds will be modified[6]. The weight and biases of the network are iteratively adjusted until the network performance function is minimized[7].

Some of the weaknesses of BPNN are the need for numerous controlling parameters, difficulty in obtaining a stable solution and the danger of over fitting and it is shown that that BPNN is unreliable even if all of network objects are pre-determined[8].

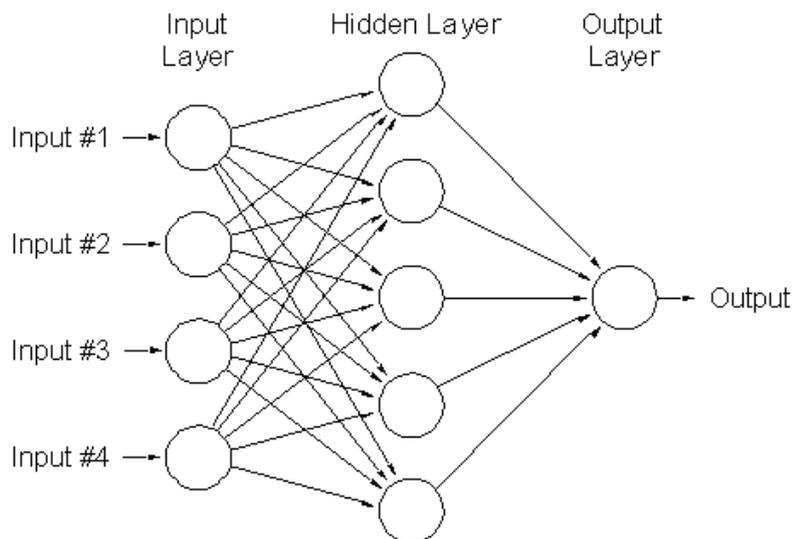


Figure 5: Typical Feedforward BP Network

### 2.2.2 Support Vector Machines (SVM)

The support vector machines (SVM) is a very powerful classifier that is first introduced by laid by Vapnik et al.[9]and is later extended to the domain of regression problems [10]. SVM is commonly used in many pattern recognition problems and has been applied in many fields today.The formulation for SVR begins with a set of data [11]:

$$D = \{(x^1, y^1), \dots, (x^l, y^l)\}, x \in R^n, y \in R \quad (1)$$

That is to be approximated with a nonlinear function;

$$f(x) = w^T \phi(x) + b \quad (2)$$

Where  $\phi(\cdot): R^n \rightarrow R^{n_h}$  is a mapping function to higher or infinite dimensional feature space. The mapping function is implicitly defined. The optimal regression function is given by the minimum of the functional;

$$\Phi(w, \xi) = \frac{1}{2} \|w\|^2 + C \sum_i (\xi_i^- + \xi_i^+) \quad (3)$$

Where  $C$  is a pre-specified value and  $\xi_i^+$ ,  $\xi_i^-$  are slack variables that represents the upper and lower constraints on the system outputs. Eq. 7 is a quadratic programming problem that is solved first by introducing a Hessian matrix ( $H$ ). If  $H$  is proven to be a positive semi-definitematrix, then  $\Phi(w, \xi)$  is a convex function and a conditional global minimizer exists. On the other hand, if  $H$  is found the positive definite, then the solution to the convex minimization problem is global and unique. [12].

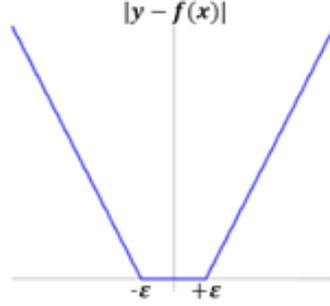


Figure 6:  $\varepsilon$ -sensitive loss function

Using an  $\varepsilon$ -sensitive loss function, Fig. 2,

$$L_\varepsilon(y) = \begin{cases} 0 & \text{for } |f(x) - y| < \varepsilon \\ |f(x) - y| - \varepsilon & \text{otherwise} \end{cases} \quad (4)$$

Employing Lagrangian duality theorem, the minimization problem becomes;

$$\max W(\alpha, \alpha^*) = \max \sum_{i=1}^l \alpha_i^*(y_i - \varepsilon) - \alpha(y_i + \varepsilon) - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\alpha_i^* - \alpha_j)(\alpha_j^* - \alpha_j) K(x_i, x_j) \quad (5)$$

With constraints,

$$\begin{aligned} 0 &\leq \alpha_i^*, \alpha_i \leq C, \quad i = 1, \dots, l \\ \sum_{i=1}^l (\alpha_i^* - \alpha_i) &= 0 \end{aligned} \quad (6)$$

Solving Eq. 9 and Eq. 10 determines the Lagrange multipliers,  $\alpha_i^*, \alpha_i$ , and the regression function is given by,

$$f(x) = \sum_{svs} (\bar{\alpha}_i^* - \bar{\alpha}_i) K(x_i, x_j) + \bar{b} \quad (7)$$

Where

$$\begin{aligned} \langle \bar{w}, x \rangle &= \sum_{i=1}^l (\bar{\alpha}_i^* - \bar{\alpha}_i) K(x_i, x_j) \\ \bar{b} &= -\frac{1}{2} (\bar{\alpha}_i^* - \bar{\alpha}_i) (K(x_i, x_r) + K(x_i, x_s)) \end{aligned} \quad (8)$$

### 2.2.3 Least Squares Support Vector Machines (LS-SVM)

Compared with BPNN, SVM aims in minimizing an upper bound of the generalization error through maximizing the margin between the separating hyperplane and the data on the structure based on structure risk minimization principle. Therefore SVM can be said has overcome certain weak points of BPNN such as local minimal solutions and the selection of the number of hidden units [13]. However, there is a problem with SVM where its computation and storage requirement increases rapidly with the number of training vectors[14].

Least squares support vector machines (LS-SVM) are least squares versions of support vector machines (SVM).The standard support vector machine formulation will lead to a quadratic programming (QP) problem with linear constraints; and the size of the matrix involved in the QP problem is directly proportional to the size of the training data [15]. Therefore to reduce the complexity of the optimization problem, Suykens et al[16] introduced a modified version of SVM called least squares support vector machines (LS-SVM). LSSVM formulation results in a set of linear equations instead of a quadratic programming problem [17].Therefore, LS-SVM has a smaller calculation complexity and faster calculation[18].

LS-SVM is used for both classification and regression problems.The formulation for LS-SVR is begun by taking a training set as in Eq. 5 and it is to be estimated using a non-linear function as in Eq. 6 where  $\phi(\cdot): R^n \rightarrow R^{n_h}$  is a mapping function to a high dimensional and potentially infinite dimensional feature space; in this paper, there are 22 dimensions in the low dimensional space and 484 dimensions in high dimension space. Next, the optimization is formulated in primal weight space;

$$\min \Phi(w, e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{i=1}^l e_i^2 \quad (9)$$

Subject to;

$$y_i = w^T \phi(x_i) + b + e_i \quad i = 1, \dots, l$$

The optimization formulation in Eq. 13 is ridge regression cost function formulated in feature space. Constructing the Lagrangian of the problem, the dual problem is derived;

$$L(w, b, e, \alpha) = \Phi(w, e) - \sum_{i=1}^l \alpha_i \{w^T \phi(x_i) + b + e_i - y_i\} \quad (10)$$

The conditions for optimality are given by[16];

$$\begin{aligned} \frac{\partial L}{\partial w} = 0 &\rightarrow w = \sum_{i=1}^l \alpha_i \phi(x_i) \\ \frac{\partial L}{\partial b} = 0 &\rightarrow \sum_{i=1}^l \alpha_i = 0 \\ \frac{\partial L}{\partial e_i} = 0 &\rightarrow \alpha_i = \gamma e_i \quad i = 1, \dots, l \\ \frac{\partial L}{\partial \alpha_i} = 0 &\rightarrow w^T \phi(x_i) + b + e_i - y_i \quad i = 1, \dots, l \end{aligned} \quad (11)$$

Upon elimination of the variables  $w$  and  $e$  and solving in  $\alpha$  and  $b$  one gets the following solution in dual space

$$\begin{bmatrix} 0 & 1_v^T \\ 1_v & \Omega + 1/\gamma \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \quad (12)$$

Where  $y = [y_1; \dots; y_l]$ ,  $1_v = [1; \dots; 1]$ , and  $\alpha = [\alpha_1; \dots; \alpha_l]$  The “kernel trick” [19] is applied here as shown;

$$\begin{aligned} \Omega_{ij} &= \phi(x_i)^T \phi(x_j) \\ &= K(x_i, x_j) \quad i, j = 1, \dots, l \end{aligned} \quad (13)$$

Hence the resulting LS-SVR model becomes; (14)

$$f(x) = \sum_{i=1}^l \alpha_i K(x_i, x_j) + b$$

In this paper, RBF kernel function is used;

$$K(x_i, x_j) = \exp\left(-\frac{|x - x_i|^2}{\sigma}\right)$$

Where  $\alpha_i$  and  $b$  are solutions to the linear system represented by Eq. 16. The LS-SVR formulation can be used to handle large datasets with no dimensionality problem. In Eq. 17, a kernel function is used to replace the high order mapping function. In this paper, the RBF kernel will be used exclusively for all computations involving kernel operations. In the case of RBF kernel, only two hyperparameters need to be tuned ( $\gamma, \sigma$ ), which is less than standard SVM.

## **CHAPTER 3**

### **METHODOLOGY**

#### **3.1 Research Methodology**

There are some procedures required for completing this project. To gain information regarding the major scope of the project, preliminary research is done by a careful investigation or inquiry. In this project, it is mostly done by reading and understanding research papers relevant to the topic, then modeling the primary reformer tube accordingly. After preliminary research is done, the author would proceed to model development and validation. The overall project flow chart will be discussed in the next section.

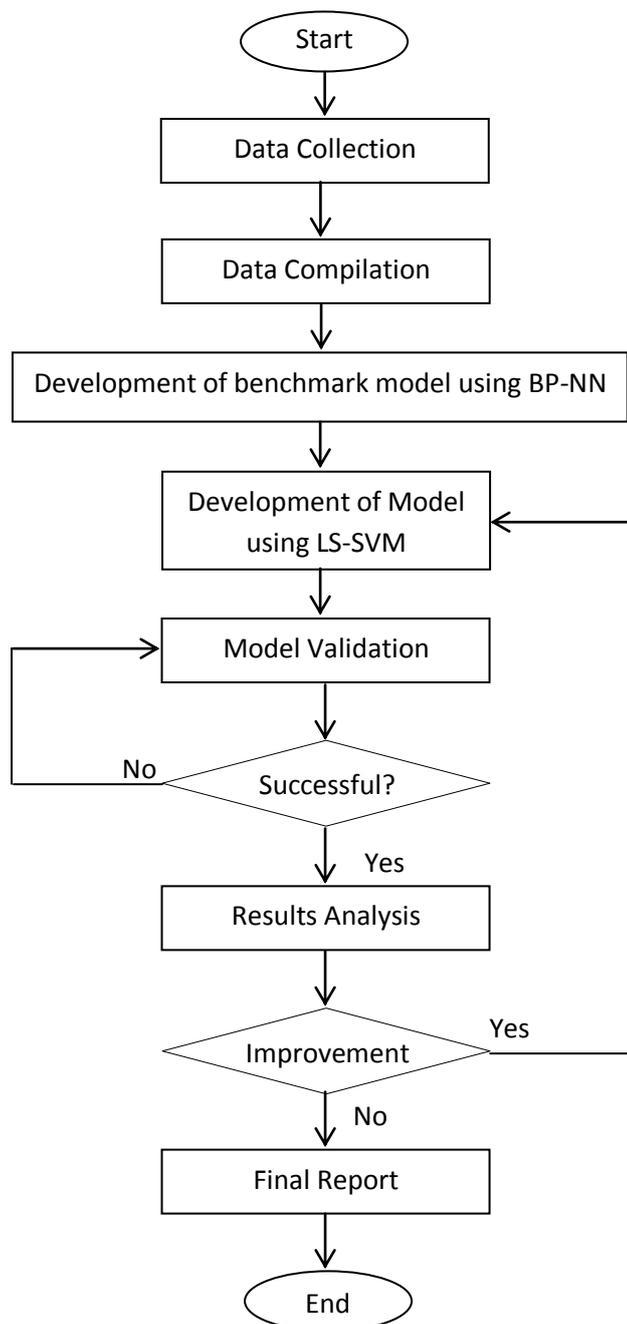
#### **3.2 Project Activities**

Firstly, the project will begin with the preliminary research on several issues which had been mention in the research methodology below. With the collective information, the project will proceed with the literature review on the modeling of the reformer TMT. Understanding of the process and behavior of the complex reformer is necessary in achieving the best model for predicting the reformer TMT. After completing the literature review, data from PASB will be collected.

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. These data is used in developing model using LS-SVM and BPNN (benchmark). The same data set will be used for both LS-SVM and BPNN, hence model comparison can be done at the end of the project. Finally,

error analysis will be done on the models to see which model is better. Error analysis is done by calculating the RMSE and accuracy values for each developed model. The figure below shows the overall project activities flow:

Figure 7: Overall Project Activities Flow



### **3.3 Tool Required**

The project will only cover the modeling of the primary reformer TMT. No prototype will be made for this project. Therefore, the only tool required will be the MATLAB software.

### 3.5 Gantt Chart

No	Detail/Week	1	2	3	4	5	6	Mid-Semester Break	7	8	9	10	11	12	13	14	
1	FYP Title Selection: Modeling of Primary Reformer Tube Metal Temperature (TMT) using LS-SVM																
2	Preliminary Research Work: Research on literatures related to the topic																
3	Data Collection																
4	Data Compilation																
5	Submission of Extended Proposal						●										
6	Preliminary Modeling																
7	Modeling of Primary Reformer TMT using BP-Neural Network (benchmark)																
8	Submission of Interim Report Final Draft																●

Table 1: Gantt chart for FYP I

No	Detail/Week	1	2	3	4	5	6	Mid-Semester Break	7	8	9	10	11	12	13	14	
1	Modeling of Primary Reformer Tube Metal Temperature (TMT) using LS-SVM																
2	Validation of LS-SVM Model																
3	Progress Report																
4	Results Analysis																
5	Pre-SEDEX																
6	Draft Report																
7	Technical Paper																
8	Dissertation Report (Soft Bound)																
9	Viva																
10	Dissertation Report (Hard Bound)																

Table 2: Gantt chart for FYP II

## CHAPTER 4

### RESULTS AND ANALYSIS

#### 4.1 Input and Output Selection

PASB provided two sets of input-output data which represents the two chambers of reformer respectively as shown in Fig.3. The average values of the process variables are used as the inputs of the model. The reading of the process variables are obtained from the distributed control system (DCS). On the other hand, the values of output used are the average tube metal temperature readings of the 144 tubes which are measured manually by plant personnel using pyrometer.

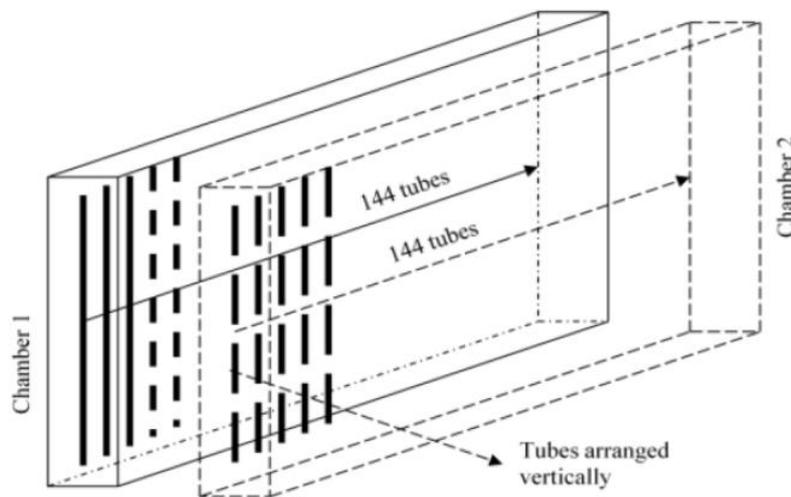


Figure 8: The arrangement of tubes in the PASB reformer

The data set from PASB is 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 readings of process variables obtained from the DCS are used as the inputs for the model.

- 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, leaving 200 input-output pairs for each chamber. 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. The total number of process variables that serve as inputs is 19. However, the number of process variable used for model development used to be reduced by selecting the best process variables that affect the tubes temperature. Based on highest correlation coefficient between the process variables (inputs) and TMT reading (output), only 7 process variables are chosen, and the following is the description of the 7 process variables chosen for model development.

Table 3: List of Selected Process Variables

No	Instrument Tag No	Unit	Description
1	FRCA1202	Nm <sup>3</sup> /h	Natural Gas Feed
2	TIA1212	°C	Feed Temperature
3	FIA1216	Nm <sup>3</sup> /h	Combustion Air Flow
4	PR1210	kg G	Fuel Pressure
5	FFRA1208	-	S/C Ratio
6	FICA1251	GJ/h	Total Caloric Energy
7	TRA1232	°C	Reformer Outlet Temperature

A total of 200 healthy data are used for model development and validation. Developed models are based on the inputs selected earlier and the output for the model is Chamber 1 and Chamber 2 average TMT.

## 4.2 Modeling Background

The LS-SVM model was developed using Nelder-Mead (NM) and Simulated Annealing (SA) as its optimizer. NM algorithm is nicknamed the “amoeba” due to its biological-like search patterns. In 2-D, it consists of a search-triangle or “crawler” or “simplex” with three points that represent the highest (worst) point, next highest point and the lowest (best) point. The intuition is to move away from high point towards the low point. The simplex moves in several transformations that are known as “reflection”, “contraction”, “reflection and expansion” and “multiple contractions” to find the optimal value for the minimization problem[20]. The parameter settings for the simplex algorithm are shown below:

- Expansion steps = 2
- Size of initial simplex = 1.2
- Contraction steps = 0.5
- Reflection steps = 1
- Shrinkage steps = 0.5
- No. of optimization steps = 200
- No. of function evaluations = 50
- Stopping criterion based on value of function
- =  $1e-6$
- Stopping criterion based on change in minimizer
- =  $1e-6$

The initial values of the hyperparameters ( $\gamma$  and  $\sigma$ ) is found using Simulated Annealing (SA). This technique is borrowed from metallurgy where it is a global optimizer for a large search space. SA is designed to find an optimized solution in a given time, rather than finding the best possible solution [21], this saves computation time and it is suitable for finding initial values for further optimizations. SA parameter settings are as follows;

- Initial temperature = 1
- Max no. of function evaluations = 40
- No. of steps at fixed temperature = 20
- Energy tolerance =  $1e-45$

An overall flow of the Nelder-Mead LS-SVM algorithm is presented in the figure below.

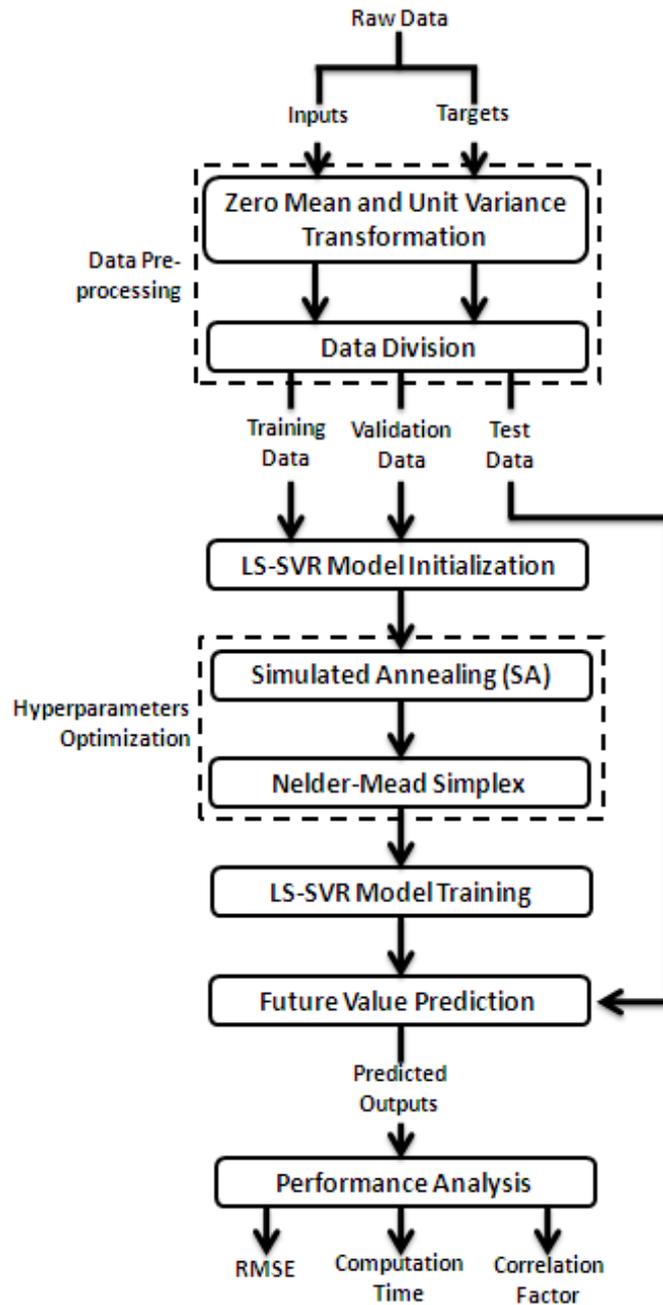


Figure 9: Nelder-Mead LS-SVM flowchart

### 4.3 Results and Analysis

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 In this section, the performance of the predicting model of primary reformer tube metal temperature using both BP-NN and LS-SVM will be presented.

#### 4.3.1 BPNN Model for Chamber 1

In this project, the benchmark model is developed using BPNN. To optimize the BPNN model, the selection of the optimal network parameters is done by trial and error. After varying the important parameters such as number of layers and number of neurons, the parameters settings that produce the best performance are as shown in the table below.

Table 4: Parameters for BPNN Model for Chamber 1

Parameters	Description
Number of Layers	2
Number of Neurons	20
Input Layer Transfer Function	tan-sigmoid
Hidden Layer Transfer Function	linear
Training algorithm	Levenberg-Marquardt
Parameters	Description
Number of Layers	2

The table below shows the a comparison of the performance of the BPNN model for Chamber 1 in terms of correlation, root mean squared and accuracy of the forecasted values. The table also shows the computational time of the model. The distribution of real and predicted s is presented in Figure 5 and 6 below.

Table 5: Performance of BPNN model for Chamber 1

	RMSE	R	Accuracy	Time
Training Data	7.5414	0.8689	99.36%	3.5292 sec
Testing Data	7.7789	0.7715	99.40%	

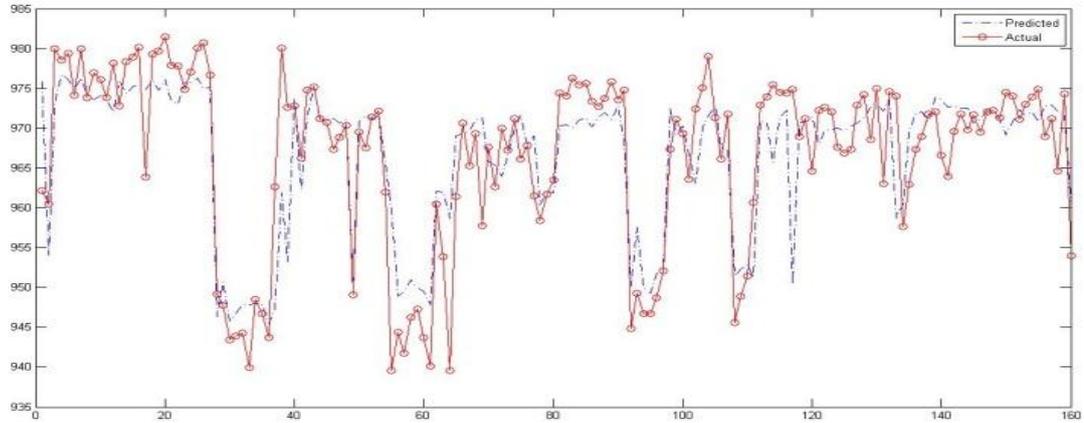


Figure 10: Chamber 1 training data for BPNN

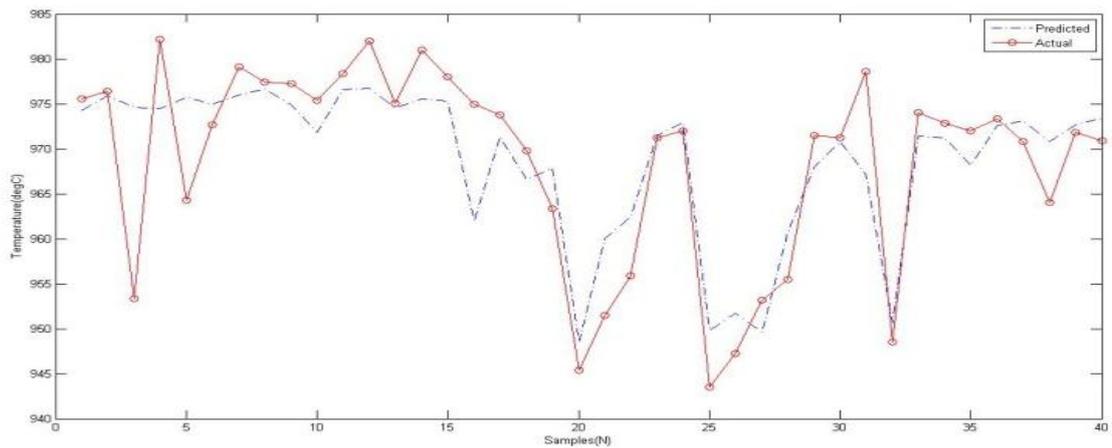


Figure 11: Chamber 1 testing data for BPNN

#### 4.3.2 LS-SVM Model for Chamber 1

In this project, the LS-SVM model is developed using the Nelder-Mead (NM) simplex algorithm. The initial values of the hyperparameters computed using simulated annealing (SA) for Chamber 1 are  $\gamma = 55.4465$  and  $\sigma = 2525.9979$ . After optimized by using the Nelder-Mead simplex algorithm, the optimized values for the hyperparameters are  $\gamma = 208.81135$  and  $\sigma = 9709.12$ . The simplex computation for this model is shown in Table 6.

Table 6: Hyperparameter optimization steps for Chamber 1

Iteration	Function evaluation	min f(x)	log(y)	log( $\sigma^2$ )	Procedure
1	3	3.774002e+001	4.0154	7.8344	Initial
2	7	3.774002e+001	4.0154	7.8344	Shrink
3	11	3.774002e+001	4.0154	7.8344	Shrink
4	15	3.773906e+001	4.0154	7.8344	Shrink
5	17	3.773906e+001	4.0154	7.8344	Contract inside
6	18	3.773906e+001	4.0154	7.8344	Reflect
7	20	3.773487e+001	4.0904	8.0219	Reflect
8	24	3.773487e+001	4.0904	8.0219	Shrink
9	26	3.773400e+001	4.1279	7.9656	Reflect
10	28	3.773310e+001	4.1279	8.0406	Reflect
11	29	3.773310e+001	4.1279	8.0406	Reflect
12	31	3.773128e+001	4.1842	8.1063	Expand
13	35	3.773128e+001	4.1748	8.0453	Shrink
14	37	3.772960e+001	4.2264	8.0805	Expand
15	39	3.772960e+001	4.2264	8.0805	Contract inside
16	41	3.772931e+001	4.3031	8.0919	Expand
17	43	3.772597e+001	4.3766	8.1761	Expand
18	44	3.772597e+001	4.3766	8.1761	Reflect
19	46	3.772210e+001	4.6389	8.3617	Expand
20	48	3.771780e+001	4.6164	8.4317	Expand
21	50	3.771036e+001	5.1297	8.8378	Expand
22	52	3.770349e+001	5.3414	9.1808	Expand

The table below shows the a comparison of the performance of the BPNN model for Chamber 1 in terms of correlation, root mean squared and accuracy of the forecasted values. The table also shows the computational time of the model. The distribution of real and predicted s is presented in Figure 7 and 8 below.

Table 7: Performance of LS-SVM model for Chamber 1

	RMSE	R	Accuracy	Time
<b>Training Data</b>	5.8932	0.8431	99.56%	0.0640 sec
<b>Testing Data</b>	5.9361	0.8388	99.53%	

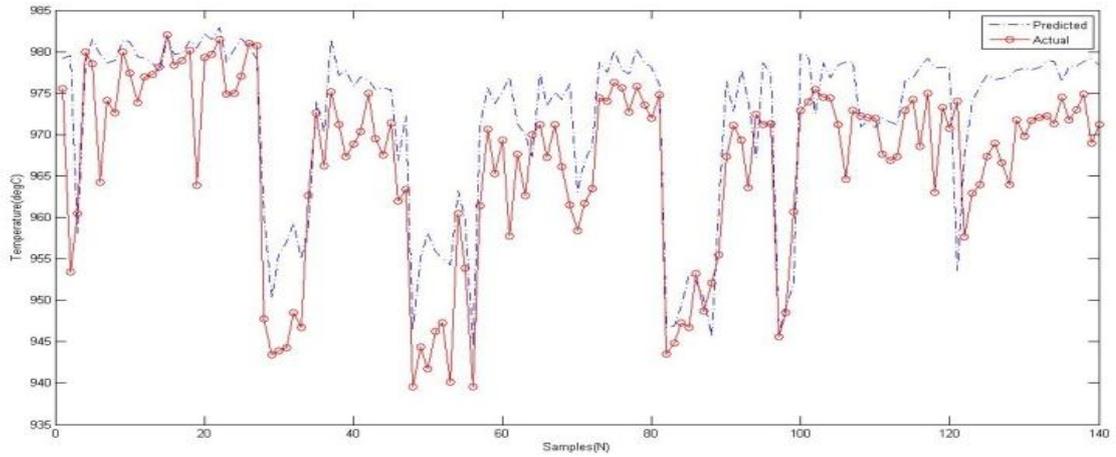


Figure 12: Chamber 1 training data for LS-SVM

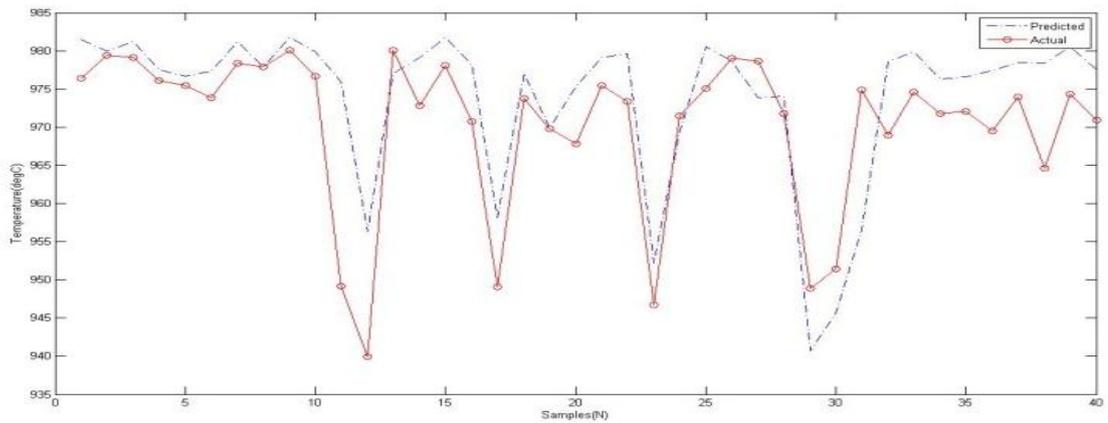


Figure 13: Chamber 1 testing data for LS-SVM

#### 4.3.3 BPNN Model for Chamber 2

After varying the important parameters such as number of layers and number of neurons, the parameters settings that produce the best performance are as shown in the table below.

Table 8: Parameters for BPNN Model for Chamber 2

Parameters	Description
Number of Layers	2
Number of Neurons	20
Input Layer Transfer Function	tan-sigmoid
Hidden Layer Transfer Function	linear
Training algorithm	Levenberg-Marquardt
Parameters	Description
Number of Layers	2

The table below shows the a comparison of the performance of the BPNN model for Chamber 2 in terms of correlation, root mean squared and accuracy of the forecasted values. The table also shows the computational time of the model. The distribution of real and predicted s is presented in Figure 9 and 10 below.

Table 9: Performance of BPNN model for Chamber 2

	RMSE	R	Accuracy	Time
Training Data	8.4458	0.7359	99.60%	1.1158 sec
Testing Data	16.7975	0.3196	99.31%	

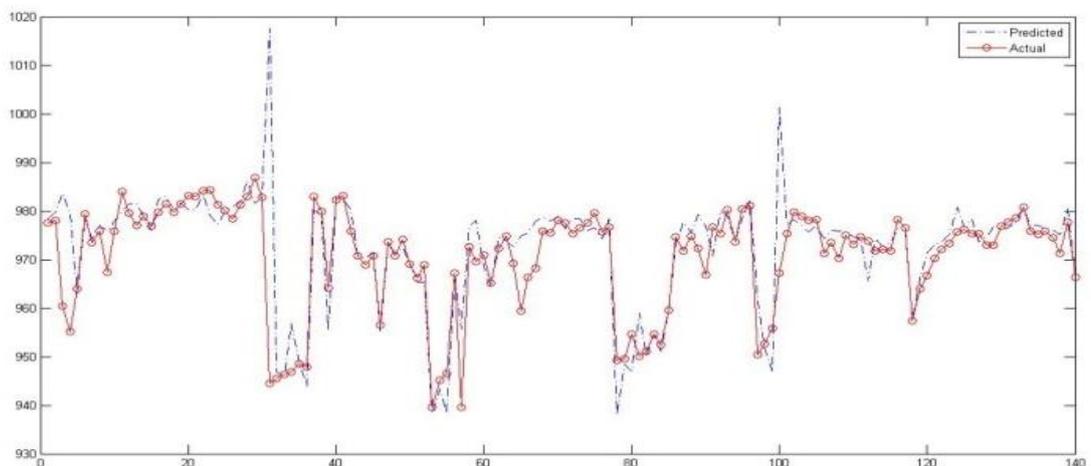


Figure 14: Chamber 2 training data for BPNN

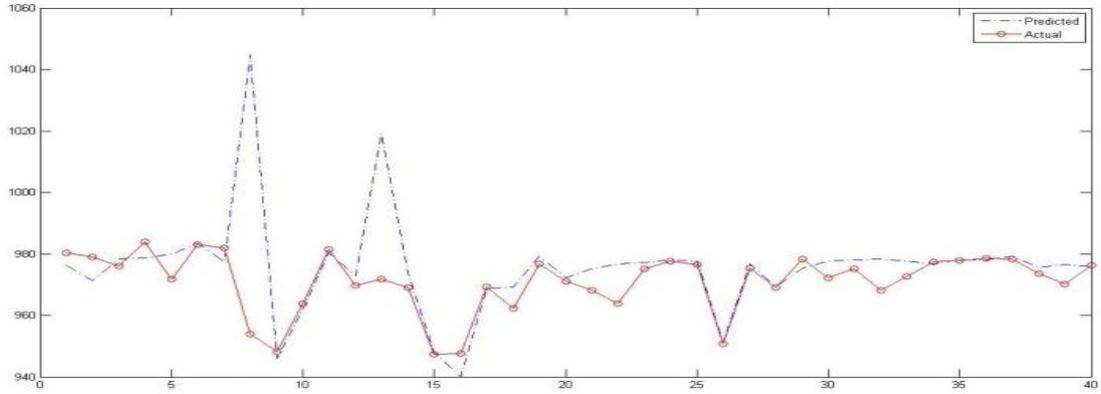


Figure 15: Chamber 2 testing data for BPNN

#### 4.3.4 LS-SVM Model for Chamber 2

In this project, the LS-SVM model is developed using the Nelder-Mead (NM) simplex algorithm. For Chamber 2, the initial values of the hyperparameters computed using SA are  $\gamma = 24.6123$  and  $\sigma = 14.276$ . After optimized by using the Nelder-Mead simplex algorithm, the optimized values for the hyperparameters are  $\gamma = 29.6827$  and  $\sigma = 14.8139$ . The simplex computation for this model is shown in Table 10.

Table 10: Hyperparameter optimization steps for Chamber 2

Iteration	Function evaluation	min f(x)	log( $\gamma$ )	log( $\sigma^2$ )	Procedure
1	3	3.099186e+001	3.2032	2.6586	Initial
2	7	3.099186e+001	3.2032	2.6586	Shrink
3	9	3.099186e+001	3.2032	2.6586	Contract inside
4	11	3.099186e+001	3.2032	2.6586	Contract outside
5	13	3.099186e+001	3.2032	2.6586	Contract inside
6	14	3.099186e+001	3.2032	2.6586	Reflect
7	16	3.099186e+001	3.2032	2.6586	Contract inside
8	18	3.098950e+001	3.4048	2.7430	Reflect
9	22	3.098518e+001	3.4822	2.7159	Shrink
10	24	3.098369e+001	3.3814	2.6773	Reflect
11	28	3.098353e+001	3.4318	2.6984	Shrink
12	30	3.098321e+001	3.3902	2.7019	Contract inside
13	34	3.098312e+001	3.4110	2.7002	Shrink
14	36	3.098312e+001	3.4110	2.7002	Contract inside
15	37	3.098312e+001	3.4110	2.7002	Reflect
16	39	3.098306e+001	3.4043	2.6965	Reflect
17	43	3.098303e+001	3.3939	2.6974	Shrink
18	45	3.098301e+001	3.3906	2.6956	Reflect
19	47	3.098301e+001	3.3906	2.6956	Contract inside
20	48	3.098301e+001	3.3906	2.6956	Reflect

The table below shows the a comparison of the performance of the BPNN model for Chamber 2 in terms of correlation, root mean squared and accuracy of the forecasted values. The table also shows the computational time of the model. The distribution of real and predicted s is presented in Figure 11 and 12 below.

Table 11: Performance of LS-SVM model for Chamber 2

	RMSE	R	Accuracy	Time
Training Data	3.9871	0.9373	99.69%	0.0373 sec
Testing Data	5.5963	0.8325	99.60%	

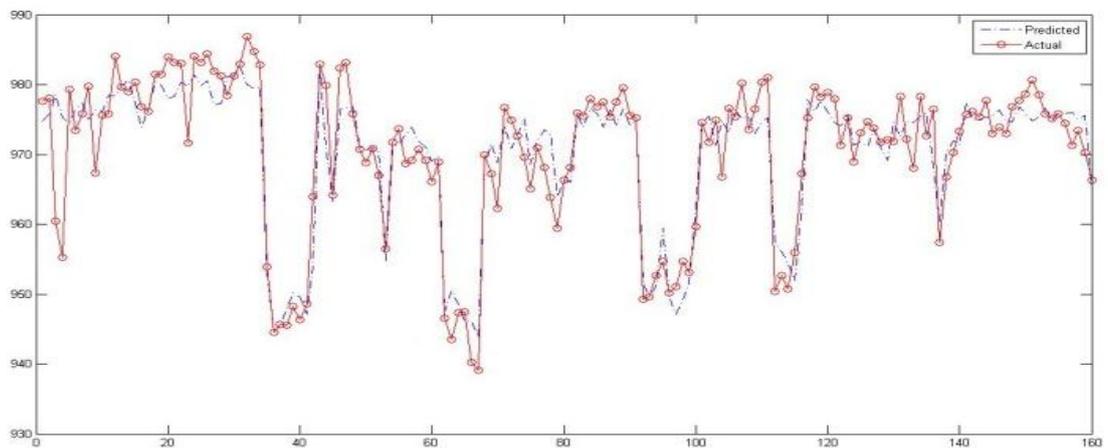


Figure 16: Chamber 2 training data for LS-SVM

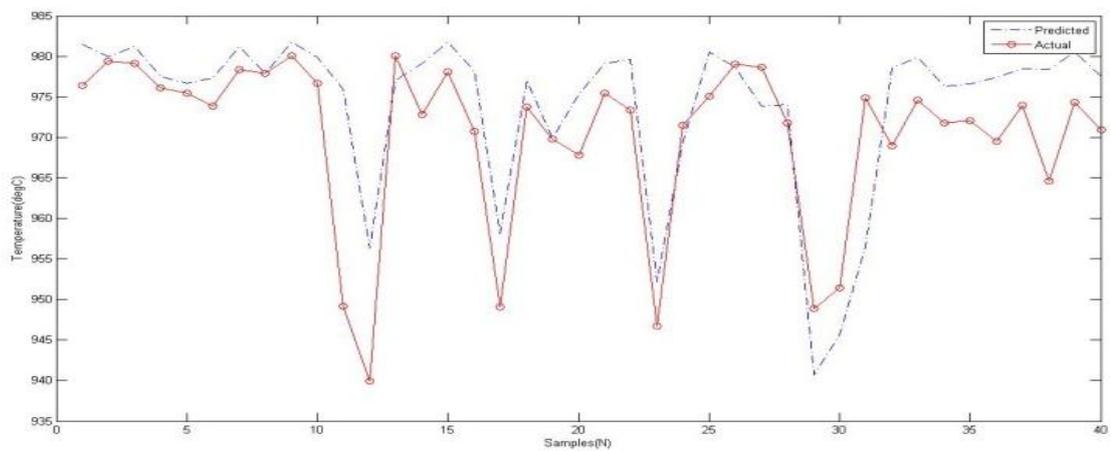


Figure 17: Chamber 1 testing data for LS-SVM

## 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 die to overheating, hence developing a model is essential to predict the reformer TMT. 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. In this case, the model is developed using LS-SVM and is compared to a benchmark model done using BPNN.

In conclusion, LS-SVM significantly outperforms BPNN in terms of both training and validation data. Besides that, the LS-SVM model has also demonstrated its ability to track and predict future values with remarkable accuracy. Therefore, from the results obtained, the accuracy of LS-SVM to model the primary reformer TMT has shown the success of completing the objectives of this project.

If the developed model is integrated to the Distributed Control System (DCS) in the plant, 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

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

The quality of the data used for modeling can be increased by having better TMT measurement. The TMT measurement accuracy should be increased by having proper procedure for the operator to manually collect the TMT readings. High accuracy measurement device such as FLIR should be used in obtaining TMT readings instead of pyrometer which has low accuracy due to the surrounding disturbance and human errors.

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 – This is the difference of 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 of 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 sent 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 be considered for model development

In this project, it is realized that one of the drawbacks of LS-SVM model includes heavy computational dependencies when it comes to large datasets. Therefore, in the future, a hybrid model that involves support vector machines with complex optimization can be developed in order to further improve the predictive accuracy and computational efficiencies.

## REFERENCES

- [1] J. Brightling, "Managing Steam Reformer Tubes," *ABI/INFORM Trade & Industry*, no. 256, p. 29, Mar/Apr 2002.
- [2] Anonymous, "Primary Reformer Problems," *ABI/INFORM Trade & Industry*, no. 250, p. 30, Mar/Apr 2001.
- [3] V.N. Patel, Modeling of Primary Reformer Tube Metal Temperature (TMT), June 2009, Thesis.
- [4] R.Rangaraju, M.A. Parameswara, T.A. Bhaskaran, M.A. Venkataswamy, A.C. Raghuram, R.V. Krishna S.K Bhaumik, "Failure of reformer tube of an ammonia plant," *Engineering Failure Analysis*, no. 9, pp. 553-561, 2002.
- [5] Hailin Jia, Shuijun Yu, Mingguo Yu Ligang Zheng, "Prediction of NO<sub>x</sub> Concentration from Coal Combustion Using LS-SVR," in *IEEE*, Jiaozuo Henan, China , 2010, pp. 1-4.
- [6] H. Lu, L. Chen B. Yang, "BPNN and RBFNN based modeling analysis and comparison for cement calcination process," in *Advanced Computational Intelligence (IWACI), 2010 Third International Workshop*, 2010, pp. 101 – 106.
- [7] J. Amini, "Optimum learning rate in back-propagation neural network for classification of satellite images," *Scientia Iranica*, vol. Vol.15, no. No.6, pp. 558-567, 2008.
- [8] Rosdiazli Ibrahim, Vijanth Sagayan Asirvadam Elangeshwaran Pathmanathan, "Development of CO<sub>2</sub> emission model of an acid gas incinerator using Nelder-Mead least squares support vector regression," *Transactions of the Institute of Measurement and Control*, 2012.
- [9] V. Vapnik, "The Nature of Statistical Learning Theory," *Springer*, 1995.
- [10] S. Golowich, and A. Smola V. Vapnik, "Support vector method for function approximation, regression estimation, and signal processing," *Advances in Neural Information Processing Systems 9*, pp. 281-287, 1997.
- [11] Steve R. Gunn, "Support Vector Machines for Classification and Regression," Faculty of Engineering, Science and Mathematics, School of Electronics and Computer Science,

University of Southampton, Technical Report 1998.

- [12] Robert J. Vanderbei, "LOQO: An Interior Point Code for Quadratic Programming," Statistics and Operations Research, Princeton University, Technical Report 1998.
- [13] C.C. Chan, Kaipei Liu, Li Ma Xiao Lei, "Pruning LS-SVM Based Battery Model for Electric Vehicles," , 2007.
- [14] Zhong-Wei Li, Jing Yang Jian-Pei Zhang, "A Parallel SVM Training Algorithm On Large-scale Classification Problems," , Guangzhou, 2005.
- [15] Da Ruan, "Applied Artificial Intelligence," in *Proceedings of the 7th International FLINS Conference*, Genova, Italy, 2006.
- [16] T.V. Gestel, J.D Brabenter, B.D Moor, J. Vandewelle J.A.K Suykens, *Least Squares Support Vector Machines*. NJ 07661: World Scientific Publishing , 2002.
- [17] Li Ying Wang Yi, "Applying LS-SVM to Predict Primary Energy Consumption,".
- [18] Wang Gang, Zhao Ke-ke, Tan De-jian Zhao Xiao-hu, "On-line least squares support vector machine," , 2009.
- [19] Cristianini N and Shawe-Taylor J, *An introduction to support vector machine and other kernel-based learning method*. Cambridge: Cambridge University Press, 2000.
- [20] Vetterling, Flannery Teukolsky, *Numerical Recipes in C*, 2nd ed. Cambridge University Press, 1998.
- [21] C. D. Gelatt and M. P. Vecchi S. Kirkpatrick, "Optimization by Simulated Annealing," *Science, New Series*, vol. 220, no. 4598, pp. 671-680, May 1983.

## **APPENDIX**

## APPENDIXA- MATLAB PROGRAMMING CODE FOR LS-SVM MODEL

```
function [Y,Yl,model] = simlssvm(model,Xt,A3,A4,A5)
% Evaluate the LS-SVM at the given points
% Copyright (c) 2002, KULeuven-ESAT-SCD, License & help @
http://www.esat.kuleuven.ac.be/sista/lssvmlab

%
% control inputs
%
ifiscell(model),
iscell_model = 1;
model = initlssvm(model{:});
ifiscell(Xt),
model.alpha = Xt{1};
model.b = Xt{2};
model.status = 'trained';
eval('Xt = A3;', ' ');
end
eval('nb_to_sim = A4;', 'nb_to_sim = size(Xt,1)-model.x_delays;');
Yt = [];
else
iscell_model = 0;
ifnargin>3,
Yt = A3;
eval('nb_to_sim = A4;', 'nb_to_sim = size(Xt,1)-model.x_delays;');
else
eval('nb_to_sim = A3;', 'nb_to_sim = size(Xt,1)-model.x_delays;');
Yt=[];
end
end

eval('Xt;', 'error(''Test data Xtest undefined...'');');

%
% check dimensions
%
if size(Xt,2)~=model.x_dim,
error('dimensions of new datapointsXt not equal to trainingsset...');
end
if ~isempty(Yt) && size(Yt,2)~=model.y_dim,
error('dimensions of new targetpointsYt not equal to
trainingsset...');
end

%
% preprocessing ...
%
ifmodel.preprocess(1)=='p',
[Xt,Yt] = prelssvm(model,Xt,Yt);
```

```

end

%
% train if status is not 'trained'
%
if model.status(1) ~= 't', % not 'trained'
warning('Model is not trained --> training now...')
model = trainlssvm(model);
end

%
% if dimension of output >1
%
if model.y_dim > 1,
if length(model.kernel_type) > 1 || size(model.kernel_pars, 2) > 1 ||
size(model.gam, 2) == model.y_dim,
%disp('multi dimensional output...');
fprintf('m');
[Y Y1] = simmultidimoutput(model, Xt, Yt, nb_to_sim);
if iscell_model, model = Y1; end
return
end
end

%
% set parameters: how much points to evaluate and to simulate
%
if (model.type(1) == 'c'),
nb_sim = nb_to_sim;
Yt = [];
elseif (model.type(1) == 'f'),
nb_sim = nb_to_sim;
Yt = [];
end

%
% simulate the model (blockwise) using the MATLAB implementations;
%

bz = 3000;
N = size(Xt, 1);
NRofBlocks = floor(N/bz);
modu = N - NRofBlocks * bz;

Y = zeros(N, 1);
for i = 1:NRofBlocks,
indb = (i-1)*bz + 1:i*bz;
Y(indb, :) = simFct(model, Xt(indb, :));
end

if modu ~= 0

```

```

indb=NrofBlocks*bz+1:NrofBlocks*bz+modu;
Y(indb,:)=simFct(model,Xt(indb,:));
end

%
% for classification
%
Yl = Y;
if model.type(1)=='c' && strcmp(model.latent,'no'),
    Y = 2*(Y>0)-1;
end

%
% postprocessing...
%
if model.preprocess(1)=='p' &&
~(model.type(1)=='c' && strcmp(model.latent,'yes')),
    [ff,Y] = postlssvm(model,[],Y);
end

%
% decode if multiclass
%
if model.type(1)=='c' && ~strcmpi(model.codetype,'none' ) &&
~strcmpi(model.code,'original'),
    Y = codelssvm(model,Y);
end

%
% Simulation
%
function Y = simFct(model,X)
model.selector = ~isnan(model.ytrain);
kx = kernel_matrix(model.xtrain(model.selector, 1:model.x_dim),
model.kernel_type, model.kernel_pars,X);
Y =
kx'*model.alpha(model.selector,1:model.y_dim)+ones(size(kx,2),1)*mod
el.b(:,1:model.y_dim);
%this is the ultimate output of SVM

function [Yt,Yl] = simmultidimoutput(model, Xt, Y,n)
%
% what to do if output multidimensional?
%
Yt = []; Yl = [];
for d=1:model.y_dim,
eval('gam = model.gam(:,d);','gam = model.gam;');
eval('sig2 = model.kernel_pars(:,d);','sig2 = model.kernel_pars;');
eval('kernel = model.kernel_type{d};','kernel=model.kernel_type;');
% not yet timeseries nor NARX

```

```

        [YtnYln] = simlssvm({model.xtrain,
model.ytrain(:,d),model.type,gam,sig2,kernel,'original'},{model.alpha
a(:,d),model.b(d)},Xt);

Yt = [YtYtn]; Yl = [YlYln];
end

% postprocessing...
if model.preprocess(1)=='p' &&
~(model.type(1)=='c' && strcmp(model.latent,'yes')),
    [ff,Yt] = postlssvm(model,[],Yt);
end

% decode if multiclass
if model.type(1)=='c' && ~strcmpi(model.codetype,'none') &&
~strcmpi(model.code,'original'),
    Yt = codelssvm(model,Yt);
end

function [model,b,X,Y] = trainlssvm(model,X,Y)
% Train the support values and the bias term of an LS-SVM for
classification or function approximation

% Copyright (c) 2010, KULeuven-ESAT-SCD, License & help @
http://www.esat.kuleuven.be/sista/lssvmlab

%
% initialise the model 'model'
%
if (iscell(model)),
model = initlssvm(model{:});
end

%
% given X and Y?
%
%model = codelssvm(model);
eval('model = changelssvm(model, 'xtrain',X);',';');
eval('model = changelssvm(model, 'ytrain',Y);',';');
eval('model = changelssvm(model, 'selector',1:size(X,1));',';');

%
% no training needed if status = 'trained'
%
if model.status(1) == 't',
if (nargout>1),
% [alpha,b]
    X = model.xtrain;
    Y = model.ytrain;
    b = model.b;
end
end

```

```

model = model.alpha;
end
return
end

%
% control of the inputs
%
if ~( (strcmp(model.kernel_type, 'RBF_kernel') &&
length(model.kernel_pars)>=1) || ...
      (strcmp(model.kernel_type, 'lin_kernel') &&
length(model.kernel_pars)>=0) || ...
      (strcmp(model.kernel_type, 'MLP_kernel') &&
length(model.kernel_pars)>=2) || ...
      (strcmp(model.kernel_type, 'poly_kernel') &&
length(model.kernel_pars)>=1)),

eval('feval(model.kernel_type,model.xtrain(1,:),model.xtrain(2,:),mo
del.kernel_pars);model.implementation='MATLAB';',...
'error('The kernel type is not valid or to few arguments');');
elseif (model.steps<=0),
error('steps must be larger then 0');
elseif (model.gam<=0),
error('gamma must be larger then 0');
% elseif (model.kernel_pars<=0),
%   error('sig2 must be larger then 0');
elseif or(model.x_dim<=0, model.y_dim<=0),
error('dimension of datapoints must be larger than 0');
end

%
% coding if needed
%
if model.code(1) == 'c', % changed
model = codelssvm(model);
end

%
% preprocess
%
eval('if model.prestatus(1)=='c', changed=1; else
changed=0;end;', 'changed=0;');
if model.preprocess(1) == 'p' && changed,
model = prelssvm(model);
elseif model.preprocess(1) == 'o' && changed
model = postlssvm(model);
end

% clock
tic;

```

```

%
% set & control input variables and dimensions
%
if (model.type(1) == 'f'), % function
dyn_pars=[];
elseif (model.type(1) == 'c'), % class
dyn_pars=[];
end

% only MATLAB
if size(model.gam,1)>1,
model.implementation='MATLAB';
end

%
% output dimension > 1...recursive call on each dimension
%
if model.y_dim>1,
if (length(model.kernel_pars)==model.y_dim ||
size(model.gam,2)==model.y_dim ||
numel(model.kernel_type,2)==model.y_dim)
disp('multidimensional output...');
model = trainmultidimoutput(model);
%
% wich output is wanted?
%
if (nargout>1),
    X = model.xtrain;
    Y = model.ytrain;
    b = model.b;
model = model.alpha;
else
model.duration = toc;
model.status = 'trained';
end
return
end
end

%
% call lssvmMATLAB.m
%
model = lssvmMATLAB(model);

%
% wich output is wanted?
%

```

```

if (nargout>1),
    X = model.xtrain;
    Y = model.ytrain;
    b = model.b;
model = model.alpha;
else
model.duration = toc;
model.status = 'trained';
end

function model = trainmultidimoutput(model)

model.alpha = zeros(model.nb_data, model.y_dim);
model.b = zeros(1,model.y_dim);
model.cga_startvalues = [];
for d=1:model.y_dim,
eval('gam = model.gam(:,d);','gam = model.gam(:);');
eval('sig2 = model.kernel_pars(:,d);','sig2 =
model.kernel_pars(:);');
eval('kernel = model.kernel_type{d};','kernel=model.kernel_type;');
    [model.alpha(:,d),model.b(d)] =
trainlssvm({model.xtrain,model.ytrain(:,d),model.type,gam,sig2,kerne
l,'original'});
end

%
% wich output is wanted?
%
if (nargout>1),
    X = model.xtrain;
    Y = model.ytrain;
    b = model.b;
model = model.alpha;
else
model.duration = toc;
model.status = 'trained';
end

function model = initlssvm(X,Y,type, gam,sig2, kernel_type,
preprocess)
% Initiate the object oriented structure representing the LS-SVM
model
%
% check enough arguments?
if nargin<5,
error('Not enough arguments to initialize model..');
elseif ~isnumeric(sig2),
error(['Kernel parameter ''sig2'' needs to be a (array of) reals'...
' or the empty matrix..']);
end

```

```

%
% CHECK TYPE
%
if type(1)~='f'
if type(1)~='c'
if type(1)~='t'
if type(1)~='N'
    error('type has to be 'function (estimation)',
    'classification', 'timeserie' or 'NARX');
end
end
end
end
model.type = type;

%
% check datapoints
%
model.x_dim = size(X,2);
model.y_dim = size(Y,2);

if and(type(1)~='t',and(size(X,1)~=size(Y,1),size(X,2)~=0)),
error('number of datapoints not equal to number of targetpoints...');
end
model.nb_data = size(X,1); %number of instances
%if size(X,1)<size(X,2), warning('less datapoints than dimension of
a datapoint ?'); end
%if size(Y,1)<size(Y,2), warning('less targetpoints than dimension
of a targetpoint ?'); end
if isempty(Y), error('empty datapoint vector...'); end

%
% initializing kernel type
%
try model.kernel_type = kernel_type; catch, model.kernel_type =
'RBF_kernel'; end

%
% using preprocessing {'preprocess','original'}
%
try model.preprocess=preprocess; catch,
model.preprocess='preprocess';end
if model.preprocess(1) == 'p',
model.prestatus='changed';
else
model.prestatus='ok';
end

%
% initiate datapoint selector
%
model.xtrain = X;
model.ytrain = Y;

```

```

model.selector=1:model.nb_data; % row matrix of 1 till the number of
instances X has

%
% regularisation term and kernel parameters
%
if(gam<=0), error('gam must be larger then 0');end
model.gam = gam;

%
% initializing kernel type
%
trymodel.kernel_type = kernel_type; catch, model.kernel_type =
'RBF_kernel';end
if sig2<=0,
model.kernel_pars = (model.x_dim); %if sigma is unspecified then use
the number of attributes of X as sigma

else
model.kernel_pars = sig2; %otherwise use sigma that are specified
end

%
% dynamic models
%
model.x_delays = 0;
model.y_delays = 0;
model.steps = 1;

% for classification: one is interested in the latent variables or
% in the class labels
model.latent = 'no';

% coding type used for classification
model.code = 'original';
trymodel.codetype=codetype; catch, model.codetype ='none';end

% preprocessing step
model = prelssvm(model);

% status of the model: 'changed' or 'trained'
model.status = 'changed';

%settings for weight function
model.weights = [];
function cost = leaveoneoutlssvm(model,Y,omega, estfct)
% Fast leave-one-out cross-validation for the LS-SVM based on one
full matrix inversion
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% INTERNAL FUNCTION %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

% Estimate the model performance of a model with fast LOO
crossvalidation.
% This implementation is based on one full matrix inverse.
Implementation
% based on "Z. Ying and K.C. Keong: Fast Leave-One-Out Evaluation
and
% Improvement on Inference for LS-SVM's, Proc. ICPR, 2004"

% Copyright (c) 2010, KULeuven-ESAT-SCD, License & help @%
http://www.esat.kuleuven.ac.be/sista/lssvmlab

%
% See also:
% leaveoneout, crossvalidate, trainlssvm

% Copyright (c) 2002, KULeuven-ESAT-SCD, License & help @
http://www.esat.kuleuven.ac.be/sista/lssvmlab

% LS-SVmlab
eval('model = initlssvm(model{:});',' ');
model.status = 'changed';

eval('estfct;','estfct='mse''');
eval('combinefct;','combinefct='mean''');

gams = model.gamcsa; try sig2s = model.kernel_parscsa; catch, sig2s
= [];end

%
%initialize: no incremental memory allocation
%
cost = zeros(1,length(gams));
py = Y;
[ff,Y] = postlssvm(model,[],Y); % Y is raw data, non preprocessed

% check whether there are more than one gamma or sigma
for g =1:numel(gams)
ifstrcmp(model.kernel_type,'RBF_kernel')
model =
changelssvm(changelssvm(model,'gam',gams(g)), 'kernel_pars',sig2s(g));
elseifstrcmp(model.kernel_type,'lin_kernel')
model = changelssvm(model,'gam',gams(g));
elseifstrcmp(model.kernel_type,'poly_kernel')
model =
changelssvm(changelssvm(model,'gam',gams(g)), 'kernel_pars', [sig2s(1,
g);sig2s(2,g)]);
else
model =
changelssvm(changelssvm(model,'gam',gams(g)), 'kernel_pars', [sig2s(1,
g);sig2s(2,g);sig2s(3,g)]);
end

```

```

% kernel matrix computation
    K = kernel_matrix2(omega,model.kernel_type,model.kernel_pars);

Ka = pinv([K+eye(model.nb_data)./model.gam
ones(model.nb_data,1);ones(1,model.nb_data) 0]);
sol = Ka*[py;0]; model.alpha = sol(1:end-1); model.b = sol(end);
yh = py - model.alpha./diag(Ka(1:model.nb_data,1:model.nb_data));

    [ff,yh] = postlssvm(model,[],yh);
if ~(model.type(1)=='c')
cost(g) = feval(estfct,yh-Y);
else
cost(g) = feval(estfct,Y,sign(yh));
end

end

```