MODELLING OF NONLINEAR SYSTEMS USING INTEGRATED OBFARX PLUS NEURAL NETWORK MODELS

ARIFF BIN OMAR

CHEMICAL ENGINEERING
UNIVERSITI TEKNOLOGI PETRONAS
JANUARY 2014

Modelling Of Nonlinear Systems Using Integrated OBFARX Plus Neural Network Models

by

ARIFF BIN OMAR

Dissertation submitted in partial fulfilment of
the requirement for the
BACHELOR OF ENGINEERING (Hons)
CHEMICAL ENGINEERING

JANUARY 2014

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Modelling Of Nonlinear Systems Using Integrated OBFARX Plus Neural Network Models

by

Ariff bin Omar

A project dissertation submitted to the
Chemical Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfillment of the requirement for the
BACHELOR OF ENGINEERING (Hons)
(CHEMICAL ENGINEERING)

Approved by,

____

(PN. HASLINDA BT ZABIRI)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
JANUARY 2014

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.

____

ARIFF BIN OMAR

ABSTRACT

The objective of this project is to develop a new model, which is by combining OBFARX linear model with nonlinear NN model. The results obtained will be compared with the previous models to show performance improvement by the new model. The new model development is based on the model developed by (Zabiri et al 2011) which is OBF linear model combination with nonlinear NN model. The OBF-NN model cannot work efficiently on some problems due to the limitations of the OBF part of the equation. So it is important to analyze the new model which is OBFARX-NN with OBF-NN model. The scope for this project will be the development of the parallel OBFARX-NN model, methods for estimating the model parameter, simulation analysis using MATLAB and evaluation on OBFARX-NN model performance. The method for completing the project will be firstly, make sure all the necessary information about the individual model is available. Then develop a theoretically working OBFARX-NN model. After that, analysis of the performance of the created model is done and also alterations here and there for better clarification. All in all, the result are the improve performance of process control by OBFARX-NN model compared to OBF-NN model.The most important aspect of the model development is the extrapolation capabilities of the model itself. When a model is forced to perform prediction in regions beyond the space of original training, then it can be said that the model can function well even when the process parameter is changed. This aspect is very important because in practical plant, the process conditions are continually changing making extrapolation inevitable. Thus, by testing the extrapolation capabilities of the OBFARX-NN model, the project had come up with the subsequent RMSE value and compared with previous model. The RMSE value indicates superior performance in the extrapolation region.

AKNOWLEDGEMENT

First of all, I wish to express my gratitude to ALLAH for giving me the strength to complete this project on time. Even at times when I was stuck or facing difficulties in completing the project, my faith in HIM gets me through eventually. Next, I would like to extend my gratitude towards my supervisor, Pn Haslinda Zabiri, for her immense help, guidance, dedication and encouragement which contribute greatly upon the completion of this project. Without her guidance and knowledge, I will not be able to continue working on this project and ultimately, complete it on time. I also would like to thank Universiti Teknologi PETRONAS for providing me the platform in the first place to work on this project. Last but not least, I would like to thank my family for their never ending support whenever I need them the most. Thank you.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
PROJECT BACKGROUND

1.1 Project Background

In current age of competitive technology development, it is essential to have extensive control of a plant processes. Long gone the days of hands on monitoring of equipment and processes by the person in charge themselves. Now, most of the process monitoring done using computer generated algorithm to reduce human-made error and improve efficiency. Besides that, stringent environmental and safety requirement as well as competitive market especially for a global scale market makes process control important as process efficiency will determine whether that process plant makes substantial profit or not. One example of specialization area that utilizes fully the capabilities of a mathematical model in process control is Model Predictive Control (MPC). MPC describes a class of computer control algorithms that control the future behavior of a plant using an explicit process dynamics model.

Process monitoring that utilize computer generate algorithm consist of mathematical models, which consist of two major groups; one is white box model and the other one is empirical or black box model. To use a white box model approach, all data and information regarding a process as well as the detail formula about it are required. Therefore, for process control in industrial scale, application of white box model is not practical because there will be too many unknowns to solve which will be time consuming to formulate and study the process in detail which will also be very expensive. Thus, empirical model are the preferred choice in industrial process control because for black box model, it is base on the input and the output value of the process itself. In other words, even if no priori information about the process is present, a working model can be created. With this information in mind, it is much more practical to choose empirical model because it is almost impossible to know each bits of information about a certain process.

Empirical model is classified into two main groups, which is linear model and nonlinear model. Between these, two groups, majority of process industry use linear model for its simplicity and cost efficient (Cutler and Ramaker 1980). However, the study of nonlinear models is becoming more and more important nowadays. This is because in various researches, modeling nonlinear models does improve the performance of a more complex plant design. For example, for MPC describe above, although there are more than 6000 applications of linear MPCs out there in industry, many complex chemical process are nonlinear in nature (Zabiri et al. 2011). These nonlinearities can be from the dynamics in chemical kinetics or thermodynamic properties. The failures of linear MPC for such cases attributed to the fact that linear model in general is unable to effectively capture complex process dynamics, which often results in poor closed-loop performance of the process or even worse, instability (Ljung 1999). Therefore, the development of nonlinear model predictive control (NMPC) is an interesting subject to study for performance improvement of processes. To prove this, recent survey indicated that roughly more than 100 industrial applications of NMPC reported which shows the possibilities of performance improvement by utilizing nonlinear models.

The results are more prominent in linear-and-nonlinear-based empirical models especially for parallel models. This is because according to (Zabiri et al. 2011) a wider range of combination is possible for processes as well as extrapolation enhancement especially for parallel models. Some of the parallel models that has been developed include extended Wiener model (Zhao et al. 2001), two-point gain-scheduling method (Piche et al. 2000), integrated linear space state model with neural network models (Yasui et al. 1996) and Linear Orthonormal Basis Filter (OBF) with nonlinear neural network models (Zabiri et al. 2011).

Another important criteria that needs to be taken into consideration is the extrapolation capabilities of the purpose model. Extrapolation can be define as a situation whereby a model is forced to perform prediction in regions beyond the space of original training. As we all know, all the plants and processes in industry does not operate with the same operating condition all the time. There will be some cases whereby, the plant needs to operate below or above the range of specified operating condition. It is unavoidable that there will be degradation of performance to some extent. However, it is essential to produce a model that can withstand this degradation when subjugate to higher values of extrapolation. Thus, it is important to test the purposed model here, which is OBFARX-NN model to extrapolate to some extent for performance checking.

1.2 Problem Statement

Development of OBF-NN model by (Zabiri et al. 2011) will serve as reference point because the analysis of performance of the control system observed shows very promising result. Not only that, the comparing the performance of the two models seems logical as OBFARX-NN model is a modification of OBF-NN model in which the OBF part of OBF-NN model changed to OBFARX. Therefore, it is extremely important to compare and then evaluate the performance of the linear part of the model that is OBFARX with existing model OBF when paired with neural network model in parallel linear-nonlinear-based empirical model. OBFARX model by itself hold several advantages over OBF model. According to (Lemma 2007), conventional OBF model does not include explicit noise model, which means that the effect of disturbance on the system output is not included. OBFARX resolve this problem by implementing the properties of explicit noise model from the AR model with the OBF model. This is important because according to (Patwardhan et al. 2005) and (Patwardhan et al. 2006), noise model is essential in improving the regulatory performance of the control system. Therefore, it is important to analyze and compare the behavior and structure of OBF-NN model with OBFARX-NN model. Besides that, the extrapolation capabilities of the purpose model needs to be taken into consideration as it will determine the performance of the model under various operating condition outside the specified range which always happen in plants.

1.3 Objective

The objective of this project is as follow:

- To develop a new model consisting of static linear OBFARX with nonlinear neural network models in parallel with the purpose of improving the performance of processes.

- To compare and evaluate the performance of OBFARX and neural network model with existing OBF and neural network model.

1.4 Scope of study

The scope of study for this FYP project will cover the following:

1- Development of relevant schemes and methods to develop the parallel OBFARX-NN model

2- Methods for estimating the model parameters

3- Development of MATLAB model algorithm for the simulation analysis

4- Nonlinear system identification simulation studies to evaluate the OBFARX-NN model performance

1.5 Relevancy of the project

As time goes by, it is essential for a process to be as efficient as possible to maximize the performance of a process. Models created over the years pave way to an even better model created. This new model also is an important part of this cycle, as not only it will improve performance of process but also usable as a reference for another model.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction to models

In general, a mathematical model is a description of a certain system by utilizing mathematical concepts. Mathematical models are widely used in many field of study including natural science (such as physics and biology) engineering (computer science and artificial intelligence) and economics. Important properties that makes mathematical model an essential part in these field is that model may help explain the workings of a system, help in study of component inside the system and to make predictions about behaviors. In many cases, the quality of the model depends on how well the mathematical models created on the theoretical side conform to the experimental data obtained. If the difference between the theoretical value and the experimental data is too distinct, it will lead to a better models created afterwards to solve the issue.

From chemical engineering perspective, a model can be describe as a set of equations that will form a system ranging from control systems to fault detection and diagnostics , to name a few (Lemma 2007). Typical models can be develop either from physical and chemical principles (white-box models) or from experimental data (empirical or black-box models). In industrial scale process, white-box models are not that practical to implement because of lack of knowledge on the physical and chemical properties of certain complex industrial processes. To do research to know these properties will take time and very costly. Thus, empirical or black box models are the preferred choice in current industrial processes. This is because, usually, an input-output data of a process is use to determine which models will be most appropriate of a process. However, one needs to take into consideration that obtaining as much as priori information as possible to make a model more accurate. This is because if the correct information is used, the model will behave with greater precision. Therefore, many aspects need to be taken into consideration before choosing which approach to use for a certain process such as plant complexity, priori information on hand and prediction on plan behavior.

2.2 System Identification

Data extracted from experimental results can be use to develop models. This method is system identification. According to (Ljung 2010), system identification is the science of building mathematical models of dynamic systems from given input-output data. It can be seen as the interface between the real applications and the mathematical world of control theory and model abstraction. System identification is a wide topic, different techniques that usually dependent on the characteristic of the model itself. There are many facets and approaches as well as method associates with system identification. Whether the model is linear, nonlinear or hybrid, system identification is crucial one way or another. Thus, it is very important to do system identification for successful applications. System identification is an important part for the analysis and control design in most control applications (Hsu and Wang 2009). To develop a reliable model, the process must be either simple or the experiment done in a control environment (Lemma 2007). However, these conditions are near impossible to replicate on an industrial scale. Therefore, according to (Lemma 2007), modern system identification usually relies on properly developed identification test with more complex computational facilities. The major steps in system identification are as below

i) Design of experiment
ii) Selection of the class of models
iii) Selection of the models structures
iv) Model validation

Systems identification do covers a wide range of topic but here will focus only on linear and non-linear model characteristics.

2.3 Linear Models

Linear models are the most common use models in the current industry. The most widely used linear models are Step response, ARX and FIR models (Ljung 1999; Nelles, 2001). This is because of its simplicity and the most establish method up to date. Below are some of the main properties of linear models.

i) Using a parameter that is constant and do not vary throughout simulation

ii) All parameters are independent of each other

Below are list of some of the most common linear models

Auto Regressive with Exogenous Input (ARX)

$$y(k) = \frac{B(q)}{A(q)}u(k) + \frac{1}{A(q)}e(q)$$

Finite Impulse Response (FIR)

$$y(k) = B\ (q)u(k) + e(q)$$

Auto Regressive Moving Average with Exogenous Input (ARMAX)

$$y(k) = \frac{B(q)}{A(q)}u(k) + \frac{C(q)}{A(q)}e(q)$$

Box Jenkins (BJ)

$$y(k) = \frac{B(q)}{F(q)}u(k) + \frac{C(q)}{D(q)}e(q)$$

Orthonormal Basis Filter (OBF)

$$y(k) = \left(\sum_{j=1}^{N} c_j L_j(q)\right)u(k) + e(k)$$

Orthonormal Basis Filter - Auto Regressive with Exogenous Input (OBFARX)

$$y(k) = \frac{G_{OBF}(q)}{A(q)} u(k) + \frac{1}{A(q)} e(q)$$

Based on (Lemma 2007), the selection of the appropriate model structure for a specific purpose, among other factors, depends on the consistency of the model parameters, the number of parameters required to describe a system with agreeable degree of accuracy and the computational load in estimating the model parameters. Meaning that selectivity of a structure does not, simply the most sophisticated. The complexity of the process itself is an important considerable factor.

2.4 Non-Linear Models

Non-linear models has become the primary focus of study in this past few years as researchers are trying to explore different options to develop a more efficient control process. Below are some properties of non-linear models:

i) They do not follow the principle of linearity and homogeneity.

ii) They might have multiple equilibrium points

iii) The state of unstable non-linear system can go to infinity in finite time.

iv) For sinusoidal input, the output signal may contain many harmonics and sub-harmonics with various amplitudes and phase difference.

Over the years, many empirical models have been develop in which can be divided into two major groups; single structure-based empirical models and linear-and-nonlinear-based empirical models. Most of the existing models fall under single structured-based empirical models. Below are some of the most common models of this type.

Artificial neural network (NN)

$$y_i = f\left(\sum_j w_{ij} y_j\right)$$

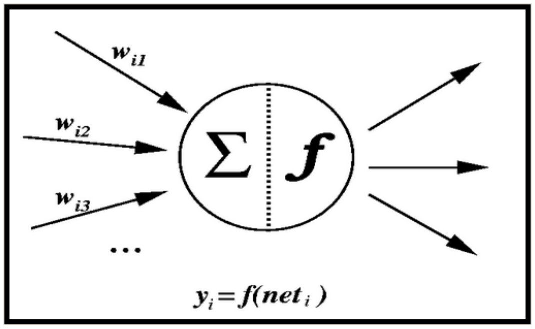

Figure 2.1 : Simple artificial neuron

Volterra models (1st kind)

$$f(t) = \int_a^t K(t,s)x(s)\,ds$$

Volterra models (2nd kind)

$$x(t) = f(t) + \int_a^t K(t,s)\,x(s)ds$$

Although the single structure-based empirical models like the above example can improve extrapolation performance (Liu et al. 2010), the linear-and-nonlinear-based empirical models seem an interesting prospect as wider range of possible combinations for extrapolation enhancement can be explored (Zabiri et al. 2011). Linear-and-nonlinear-based empirical models can be categorized into two classes, which is series and parallel. One of the most studied classes of non-linear models of this type in series is by utilizing block-oriented representations. Separate block in series will represent linear dynamic systems and non-linear static mappings. Among others, Wiener model structure is one of the most frequently use block-oriented (BO) models (Ramasamy et al. 2011). Its application ranging from distillation columns to pH processes (Saha et al., 2004).

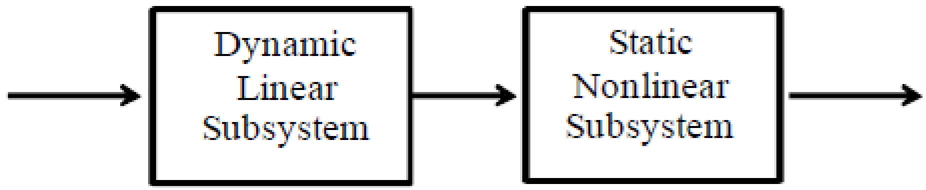

Figure 2.2: Block diagram of a Wiener model

Figure 1 above shows a typical Wiener model consist of a dynamic linear part cascaded with a static nonlinear component. The parallel case holds the most potential for performance improvement. According to (Zabiri et al. 2011), through the usage of residual of the linear model part, the parallel structure ensures that the value obtained is the desirable one. Besides that, the performance of the nonlinear model part will be as good as or better than the linear model, and the opportunity for the underlying nonlinear properties to be captured by the subsequent residuals (Sjoberg et al. 1995).

One particular case study shows in (Ramasamy et al. 2011) prove that parallel linear-and-nonlinear-based empirical models shows much better potential for performance improvement. Four models was chosen to represent each type of model classification which is subjected to extrapolated data which goes up to 30% increase feed flow rate, F beyond the original range (Training set) used to identify the model. The results analyzing the root mean squared error were as followed. Detail observation show that in the identification stage, all models, except for linear OBF, the values obtained shows that the performance of the models are comparably similar which can be seen from the resulting RMSE values. However, when subjected to data that slowly drifting away from the original training set, it can be clearly observed that the parallel Laguerre-NN models have superior extrapolation performance in comparison to the series Laguerre-NN model, as well as against pure NN. Pure NN behavior is as expected due to its constant extrapolation behavior. Series Laguerre-NN structure on the other hand, relies on pure linear behavior when subjected to extrapolation. The results obtained clearly shows that a parallel series hold much greater potential for performance improvement.

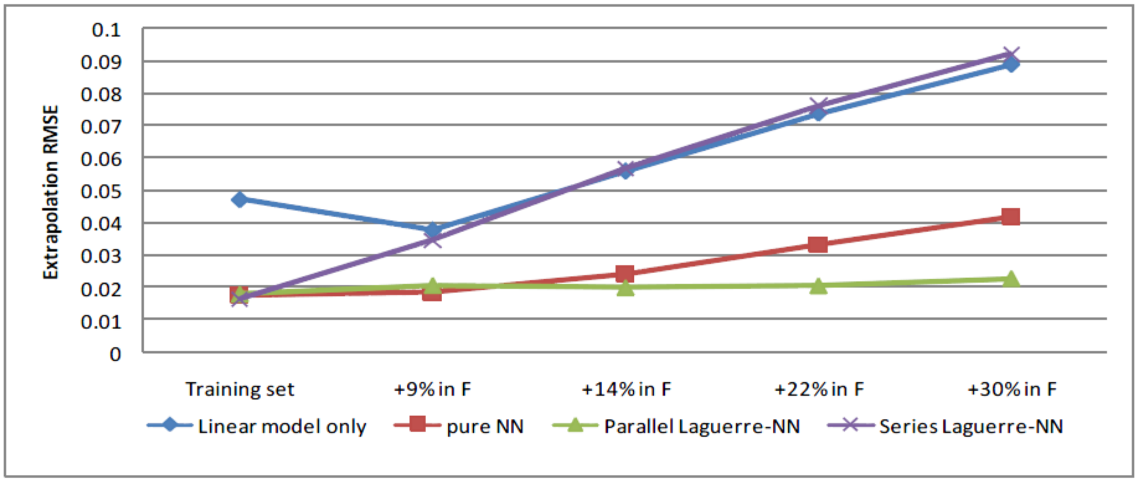

Figure 2.3: Root mean squared error (RMSE) for the training set and extrapolated data sets. (Adopted from Ramasamy et al 2011)

Some of the models develop throughout the years are Integrated linear partial least squares (PLS) with nonlinear static feed-forward neural network. (Known as extended Wiener model) (Zhao et al. 2001), Two-point gain-scheduling method using a static neural network model and a quadratic difference equation (Piche et al. 2000), Integrated linear space state model with neural network models (For identification and control of a one-degree-of-freedom vibration system.) (Yasui et al., 1996), and Linear Orthonormal Basis Filter (OBF) with nonlinear neural network (NN) models (Zabiri et al. 2011).

Among these models, further analysis of OBF-NN especially OBF properties and key features is necessary because it will act as a benchmark for comparing the performance of the model with OBFARX-NN model.

2.5 Orthonormal basis filter

Orthonormal basis filter (OBF) model key feature is that it can capture the dynamics of a linear system within an acceptable accuracy range even with relatively fewer number of parameters than other linear model in which the parameter of the OBF models usually found using linear least square method (Lemma 2007). The term best describing the parameter of the model is parsimonious. Nevertheless, this is true only if the poles used when developing the model are relatively close to the dominant poles of the system (Patwardhan and Shah 2005). This statement is also present in (Heuberger et al. 2005). In other words, for poles with position farther away from the dominant poles of the system, usage of more parameters is necessary. This condition will lead to a condition in which the model is non-parsimonious.

To overcome this, combine OBF with either AR or ARMA model so that the two models will complement each other weaknesses to produce a better linear model. According to (Lemma 2007), by combining the deterministic part of OBF model an noise part of either AR or ARMA model, the subsequent model will inherit both advantages of OBF model as well as explicit noise model. Noise model by definition is the effect of disturbance on the system output (Lemma 2007). Therefore, in theory, the performance of OBFARX-NN model should be better than OBF-NN model.

Before that, understanding the basic concept of the novel NN model is also crucial. This is because it will be much easier to study, understand and analyze the proposed model here.

2.6 Neural Network models

Neural networks are a wide class of flexible nonlinear regression and discriminant models, data reduction models, and nonlinear dynamical systems (Sarle 1994). They often consist of large number of "neurons," i.e. simple linear or nonlinear computing elements, which interconnected in complex ways and often organized into layers. Usually, artificial neural networks are used in three main ways:

- as models of biological nervous systems and "intelligence"
- as real-time adaptive signal processors or controllers implemented in hardware for applications such as robots
- as data analytic methods

Many NN models are similar or identical to popular statistical techniques such as generalized linear models and polynomial regression especially when the emphasis is on prediction of complicated phenomena rather than on explanation. These models such as counter propagation and learning vector quantization are useful especially for data analysis. Figure 4 and 5 shows some of the important characteristic of a NN model. Detail study by (Rumelhart and McClelland 1986) shows that NN model showed that it is a perfect application for parallel-distributed process. Therefore, by considering this, it is a very concrete reason to use NN model for the nonlinear part as it works well for parallel models.

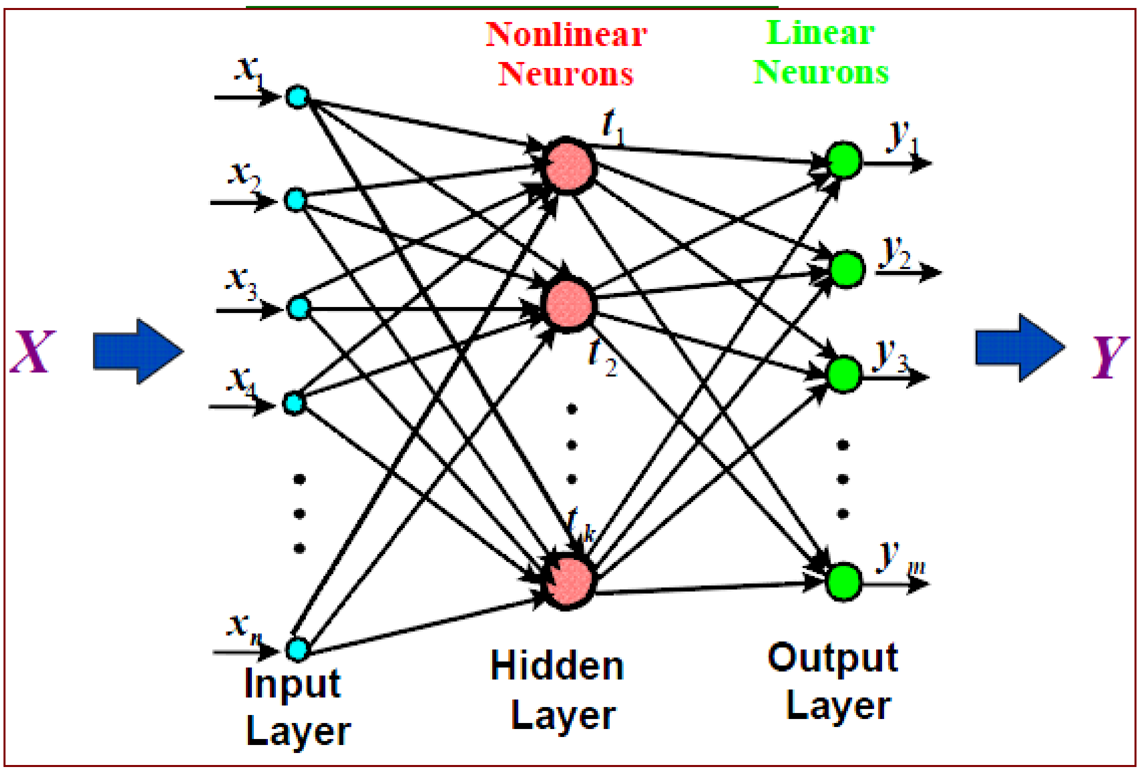

Figure 2.4: Example of multiple layer NN model

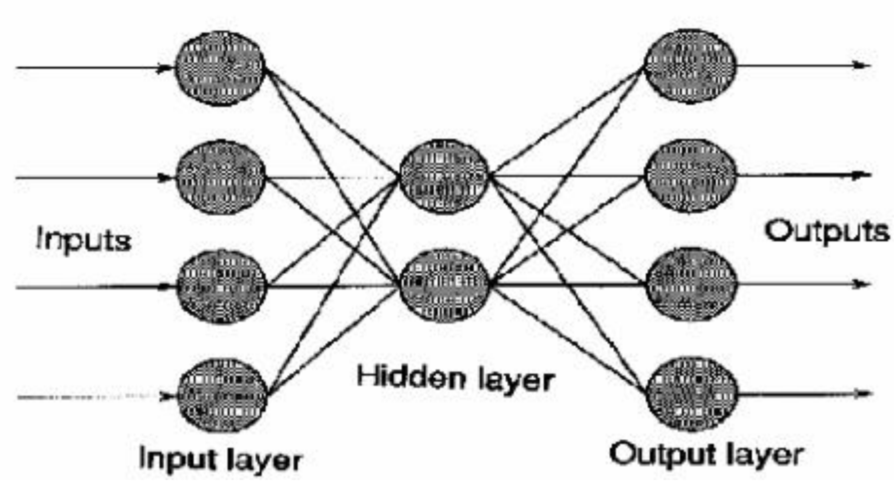

Figure 2.5: First example of multiple layer NN model

2.7 Other Application of OBF and other models

OBF model and its variation as well as NN model application in real industry is very important. Below are some of the applications for the models:

- Neural Networks on Vibration Fault Diagnosis for Wind Turbine Gearbox

  According to Huang, Jiang, etc., they use a wavelet to study fault diagnosis of gearbox conditions. The project has been successfully done by proving the accurate diagnostic results effectiveness of the method.
  Model used:

$$W\ (a, b) = \frac{1}{\sqrt{a}} \int_c \psi^*_{a,b}\ (t) x\ (t) dt$$

- Neural Network modeling of plain and grooved microstrips

  According to (Chubukjan 2000), a feed forward network with back propagation, having one or more hidden layers with non-linear transfer functions and one output layer with a linear transfer function, is capable of approximating any function with a finite number of discontinuities with arbitrary accuracy. By using “Vector Finite

Element Method" VFEM, the values of the microstrip characteristic impedance for both plain and grooved geometries were obtained. These values were used to train the NN models.

$$H\left(r'\right)=\frac{1}{4\pi}\int_{s1}\left[G\left(r,r'\right)\frac{\partial H(r)}{\partial n}-H(r)\frac{\partial G\left(r,r'\right)}{\partial n}\right]dS_1$$

CHAPTER 3

METHODOLOGY

3.1 PROJECT FLOW CHART

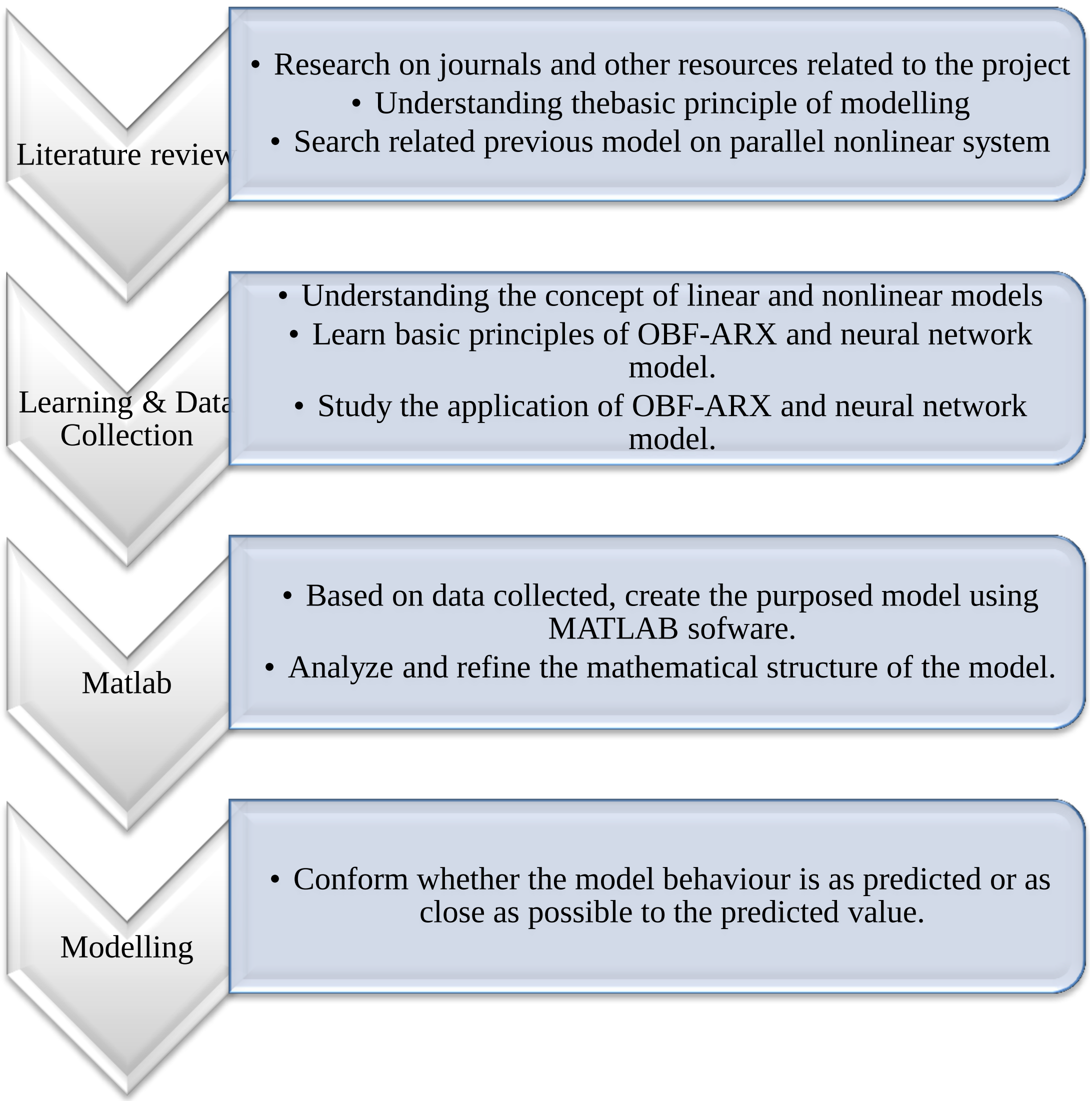

Figure 3.1: Project flow

3.2 DATA COLLECTION METHODS

Data collection method using two type of approach:

- Research and data collection by extracting information from existing parallel linear-nonlinear-based model, which will act as base guide for implementation of OBFARX with neural network models.
- Trial and error approach when using simulation (MATLAB) to test the compatibility of OBFARX with neural network models meaning that the value obtained will be as close as possible to the predicted value.

3.3 OVERALL MODEL DEVELOPMENT PLANNING

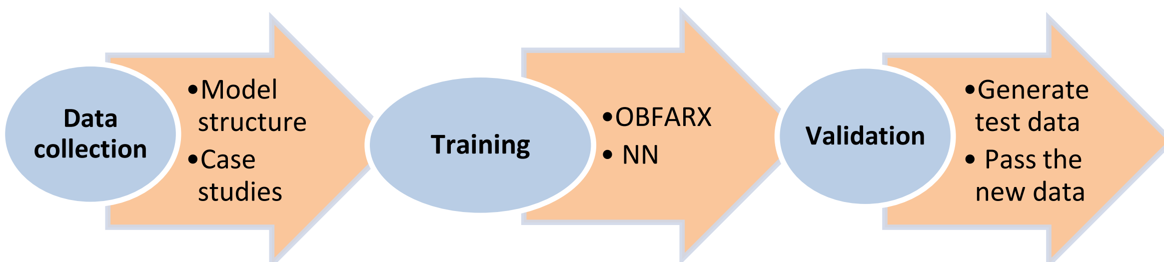

Figure 3.2: Overall planning for model development

3.4 GANTT CHART

Refer to appendix 1

3.5 TOOLS FOR DEVELOPMENT

Below are the tools used over the duration of the research

Table 3.1: List of software use

Activity	Software
Collecting Data and Analysis	Internet (Utepedia, Scorpus, etc) / Data simulation (MATLAB)
Development Tools	MATLAB®
Documentation	Microsoft word / Adobe Reader/ Paint

3.6 OBFARX – NN Model Development

This section contains the development of the proposed parallel OBFARX-NN model which will be further analysed using MATLAB.

3.6.1 Model Structure

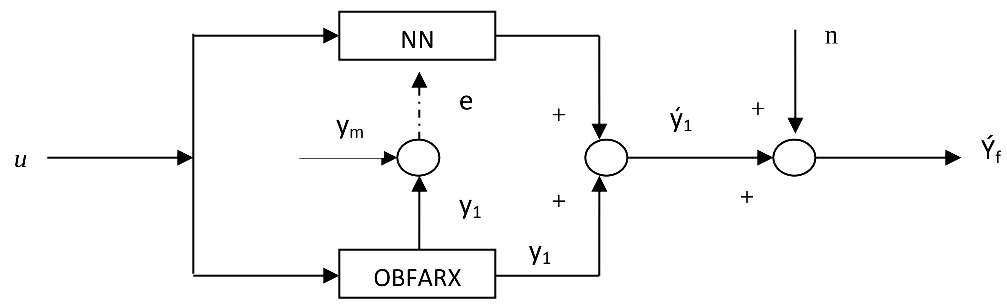

Figure 3.3: Proposed model for OBFARX-NN model (block diagram)

The figure above shows the overall configuration for the purposed OBFARX-NN model. The linear OBF model is identified first, with the residual value use to calculate the ARMA part. The OBFARX value obtained then will be used to calculate the nonlinear part which is NN. The nonlinear NN model is trained with predicted residuals of the OBFARX. Simple algorithm to identify the OBFARX-NN model can be described as follow:

1. Develop OBF model to get y_{OBF}
2. Calculate the predicted residual using $y_{RES} = y_{ACTUAL} - y_{OBF}$
3. Use the residual to calculate the ARMA part.
4. Calculate the predicted residual from the OBFARX using $y_{RES1} = y_{ACTUAL} - y_{OBFARMA}$

3.6.2 OBFARX-NN Nonlinear Identification Algorithm

It is important to clearly understand the exact structure for the purpose OBFARX-NN model to produce the best possible result. Below is the illustration of the sequential identification structure purposed for the parallel OBFARX-NN model.

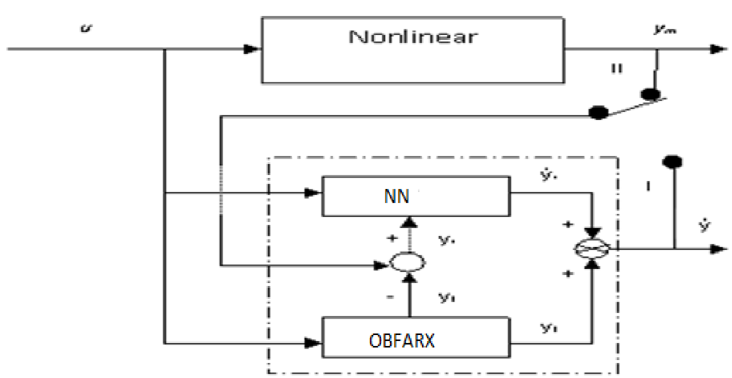

Figure 3.4: Proposed sequential identification of residual-based parallel OBFARX-NN models

From the figure, linear OBFARX model is first identified and develop, with the residual value used to train the nonlinear NN model. The nature of this parallel structure allows both the models to capture the important characteristics of the process separately resulting in more accurate reading.

3.6.3 Case study descriptions

To demonstrate and test whether the models develop does indeed improve the performance of a process it needs to be compared with other already established model. Therefore, appropriate case studies for this project will be based on previous research by (Zabiri et. al 2011). The case study is the CSTR case study used as the demo problem in MATLAB Neural Network Predictive Control Toolbox shown in Figure 4.7. The case study will basically be about the development of Nonlinear Model Predictive control (NMPC). Proposed OBFARX-NN will be used in this demo model to evaluate its performance based on effectiveness and validity.

The dynamic model of the process is given by:

$$\frac{dh_{cstr}(t)}{dt} = w_1(t) + w_2(t) - 0.2\sqrt{h_{cstr}(t)}$$

$$\frac{dC_b(t)}{dt} = \left(C_{b1} - C_b(t)\right)\frac{w_1(t)}{h_{cstr}(t)} + \left(C_{b2} - C_b(t)\right)\frac{w_2(t)}{h_{cstr}(t)} - \frac{k_{r1}C_b(t)}{(1 + k_{r2}C_b(t))^2}$$

Where;

$h_{cstr}(t)$ is the liquid level

$C_b(t)$ is the product concentration at output of the process

$w_1(t)$ is the flow rate of concentrated feed

$w_2(t)$ is the flow rate of the diluted feed, C_{b2}

The input concentrations are set to C_{b1}= 2.14 and C_{b2}= 1.09. The constants associated with the rate of consumptions are k_{r1}=1 and k_{r2}=1. The objective of the controller is to maintain the product concentration by adjusting the flow $w_1(t)$. The flow $w_2(t)$ is set at 0.1. The level of the tank is not controlled under the current analysis.

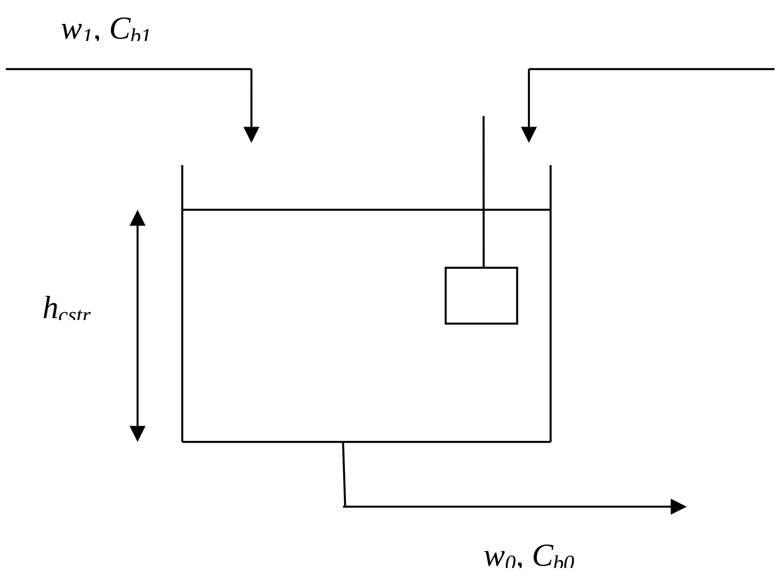

Figure 3.5: A Catalytic Continuous Stirred Tank Reactor (CSTR)

For this case study, the input design sequence for the identification is based on the idea of the Level change at random instances signal. Data is generated for the identification purposes with sampling time of 0.2s for 6502 samples. The development of NN models typically requires large amount of data.

For the data partitioning, (75%) of the generated data is set into the training while (25%) for validation test sets. The aim is to evaluate how well the model generalizes (predicts) when subjected to out-of-sample data that is not used during the identification stage.

To show further details about the process at hand, an application inside MATLAB called Simulink is used to relate the case study with the model that being develop. By doing this, the validation of the model can be done using MATLAB by producing the RMSE value which will be discussed further in the next section.

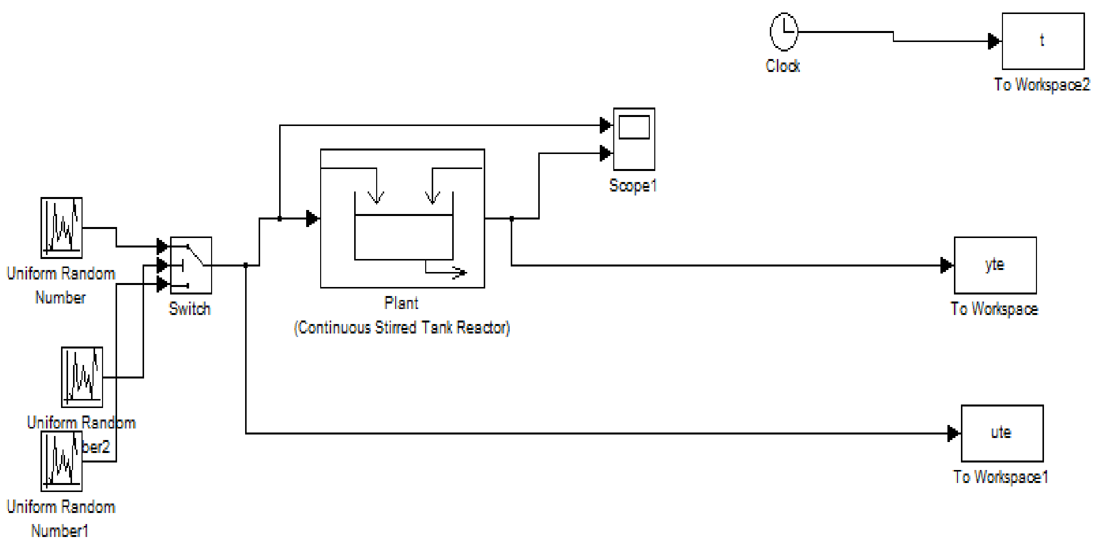

Figure 3.6: Representation of the case study (whole blending process) using Simulink

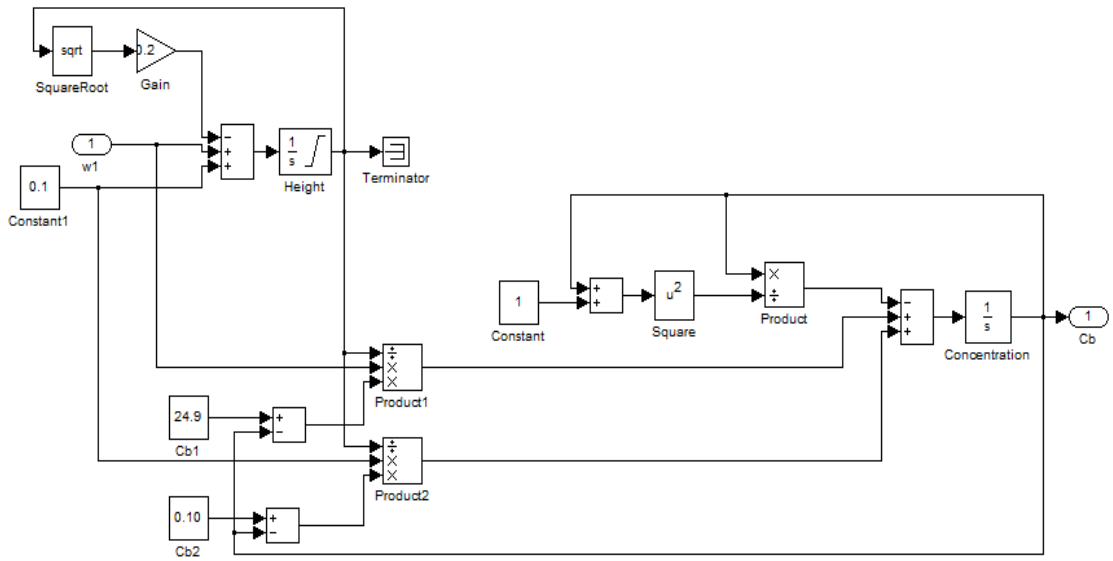

Figure 3.7: Representation of the case study (CSTR logic) using Simulink

3.6.4 OBFARX-NN model validation

For this section, the aim is to observe the graphical plots between model output and measured process output. There are several ways to obtain this information. For this project, two methods will be considered: (1) numerical method which use Mean Squared Error (MSE) as well as the Root Mean Square Error (RMSE), and (2) the graphical method which involve the residual plot and plot between predicted and measured process outputs.

The MSE is defined as

$$MSE = \frac{\sum_{i=1}^{n}(y - \hat{y})^2}{n}$$

Where

y is measured output

ŷ is predicted value of y.

The RMSE is define as

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y - \hat{y})^2}{n}}$$

CHAPTER 4

RESULTS AND DISCUSSION

Based on the research done, these are the findings so far. Firstly, the steps in finding the current results need to be analysed.

After collecting all the necessary information and data from various journals and research paper, it is necessary to test whether the data collected can be used to develop a working OBFARX-NN model. Thus, to illustrate this, a program called MATLAB is used.

First of all, it is required to reproduce a working OBF model which will act as the starting point for subsequent development of the desired model.

[model H0 yobfv yv yobf] = soptdm(tdata,vdata)

yobfv	= the vdata value after calculation
yobf	= the tdata value after calculation
yv	= the overall output value after calculation
soptdm	= syntax that represents the OBF formula
tdata	= data that have been categorized for training purpose (3/4 of overall)
vdata	= data that have been categorized for validation purpose (1/4 of overall)

The above equation shows the formula that is used to generate a working OBF model using MATLAB. The data needs to be separate into two parts because for any system identification, it is required to separate all the data collected into 75% for training and 25% for validation. The overall syntax for the OBF part with 6502 data is as below.

```
%% 1. Generate data
clear;
clc;
close all;

load u;
```

```
load y;

u_length = length (u);

Ts = 0.2;
data = iddata(y,u,Ts);

trstart = 1; trend = round (0.75*6502);
vstart = trend + 1; vend = 6502;
tdata = data(trstart:trend);
vdata = data(vstart:vend);

[model H0 yobfv yv yobf]=soptdm(tdata,vdata);
```

From the syntax above, OBF model generated is as below.

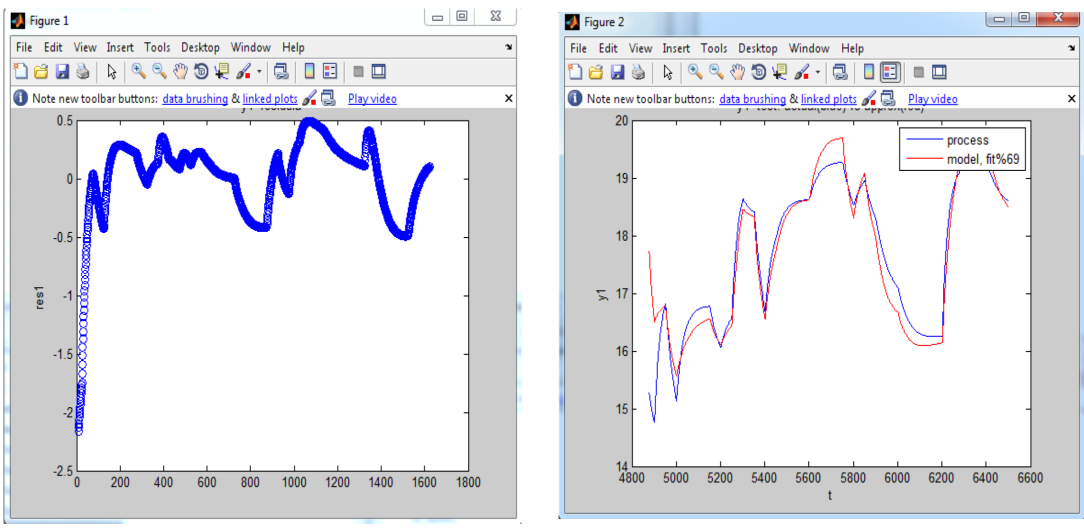

Figure 4.1: Residuals value and OBF model interpretation

Next, it is required to calculate the residual value which is then used to calculate the ARMA part of the model. The general ARMA model structure is as follow:

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

Where:

- $y(t)$ — Output at time t.
- n_a — Number of poles.
- n_b — Number of zeroes plus 1.
- n_c — Number of C coefficients.
- n_k — Number of input samples that occur before the input affects the output, also called the *dead time* in the system. For discrete systems with no dead time, there is a minimum 1-sample delay because the output depends on the previous input and $n_k = 1$.

For this ARMA model, only na and nc value of the model needs to be specified. This is because here, we only deal with a model consisting of no input channel and 1 output channel.

```
residual = y - (yobf + yobfv)

>> armamodel=armax(residual,[1 [] 2 []])
Discrete-time IDPOLY model: A(q)y(t) = C(q)e(t)
A(q) = 1 - 0.992 q^-1

C(q) = 1 + 0.06713 q^-1 + 0.06006 q^-2

Estimated using ARMAX from data set residual
Loss function 0.000660403 and FPE 0.000661013
Sampling interval: 1
```

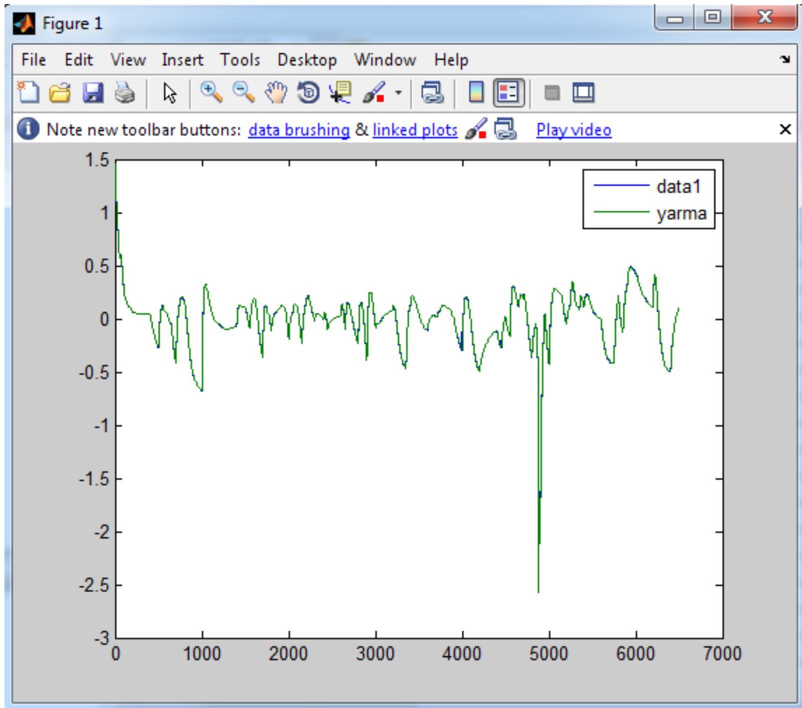

Figure 4.2: Comparison between the predicted and actual OBFARX model (nc=2)

Further analysis shows that by using different value of nc (nc = number of C coefficients) when developing the ARMA model, a more suitable model can be obtained.

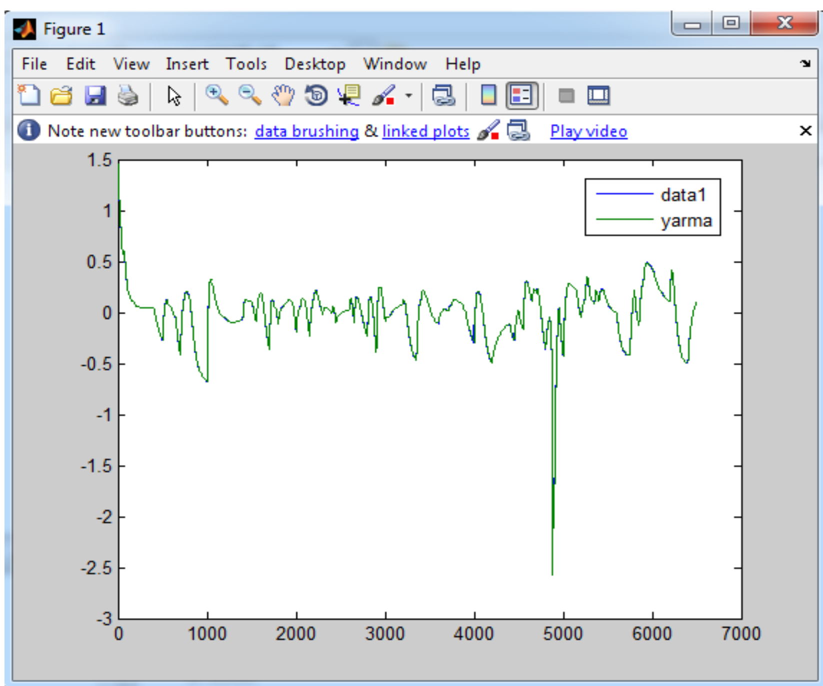

Figure 4.3: Comparison between the predicted and actual OBFARX model (nc=3)

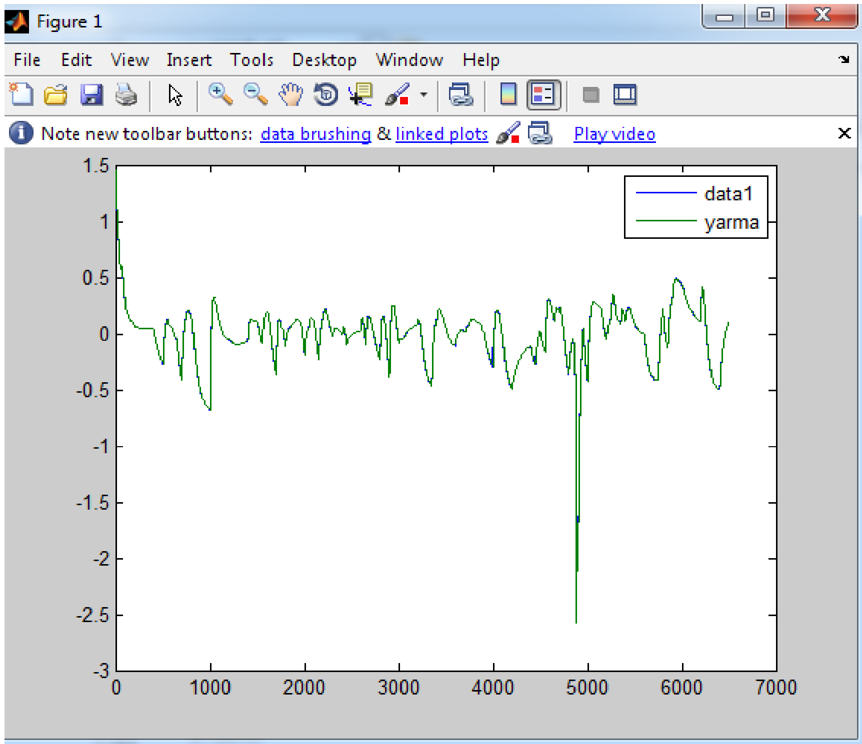

Figure 4.4: Comparison between the predicted and actual OBFARX model (nc=4)

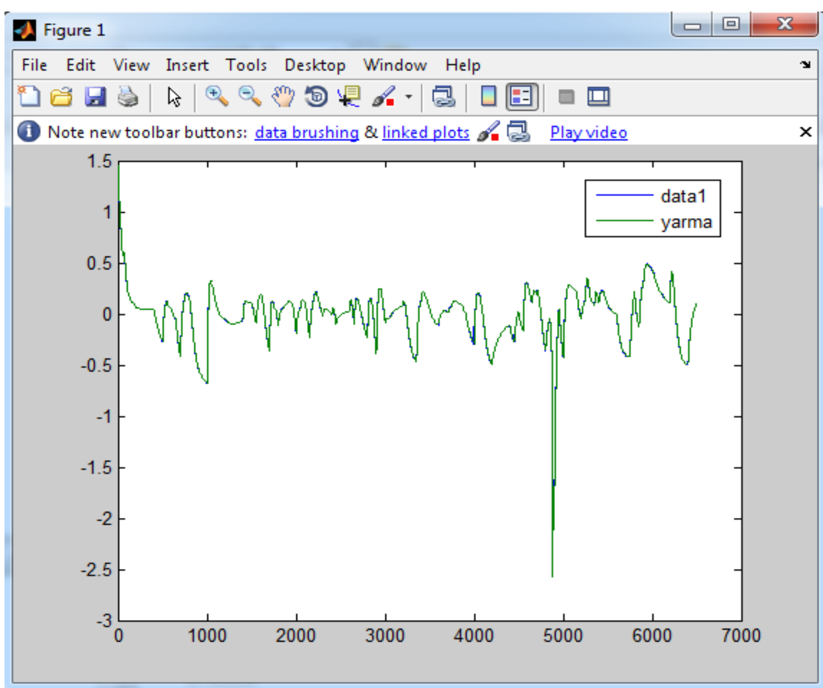

Figure 4.5: Comparison between the predicted and actual OBFARX model (nc=5)

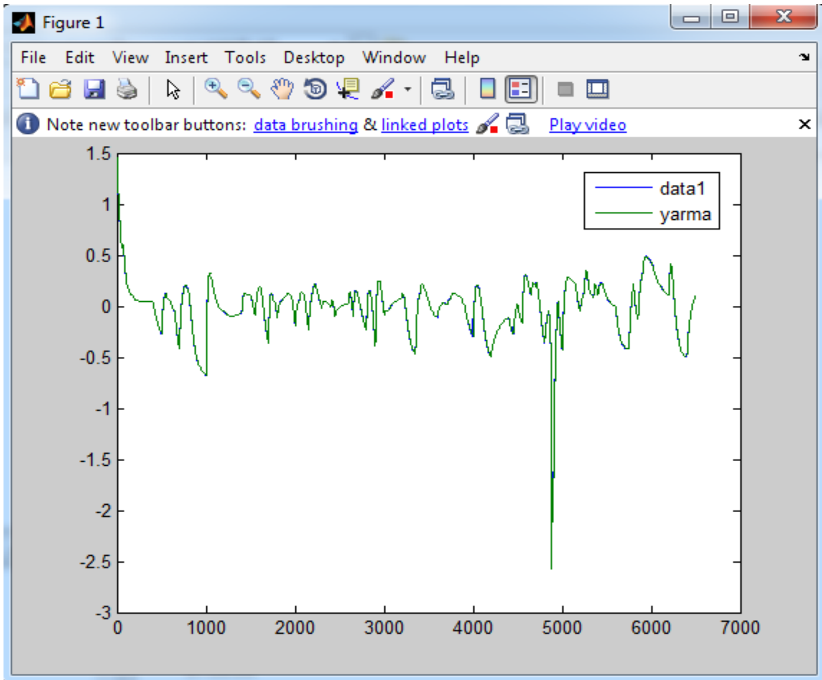

Figure 4.6: Comparison between the predicted and actual OBFARX model (nc=6)

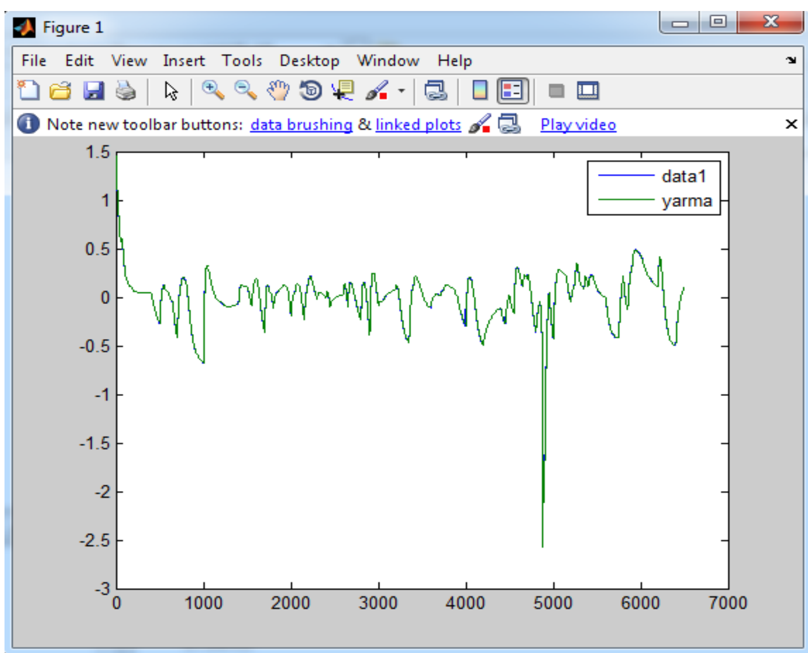

Figure 4.7: Comparison between the predicted and actual OBFARX model (nc=7)

Based on the graph obtained for the comparison between predicted and actual OBFARX model, it can be said that the different values of nc does change the performance but very slightly, meaning the difference is very small. With this in mind, we choose nc value of 2 for simplicity. Below are the values that are affected by changing the nc value. Loss function is the value which corresponds to the difference in values between the estimated and true values for an instance of data while fpe is the final prediction error for the estimated model.

Table 4.1: Values of loss function and Fpe for nc = 2, 3, 4,5,6,7

nc	Loss function	Fpe
2	0.000660403	0.000661013
3	0.000658331	0.000659141
4	0.000656662	0.000657672
5	0.000655287	0.000656496
6	0.000654132	0.000655541
7	0.000653153	0.000654761

After producing the appropriate OBFARX model, now is the section where the neural network part of the model is developed. Below is the syntax for the development of the neural network part of the model.

```
clear;
clc;
load u
load y

y1=y(1:4877)
y2=y(4878:6502)

u1=u(1:4877)
u2=u(4878:6502)

data1=iddata(y1,u1,0.2)
data2=iddata(y2,u2,0.2)

m=nlarx(data1,[1 0 0],'sigmoidnet')

NL=m.Nonlinearity;
get(NL)
NL.Parameters
```

```
Param=NL.Parameter

yp=predict(m,data2)
y_sigmoidnet=yp.OutputData
t1=[4878:1:6502]
plot(t1,y2,t1,y_sigmoidnet)

legend('actual','sigmoidnet')
```

For this part, it is required to specify a new input and output data based on the values obtained from the OBFARX model that has been develop earlier. This new input and output data are denote by u1, u2, y1, and y2 whereby u1 and y1 are input output values for the test set while u2 and y2 are the input output values for the validation set. To estimate the values when implying the data with neural network model, a nonlinearity estimator for nonlinear ARX and Hammerstein-Wiener models called sigmoidnet is used.

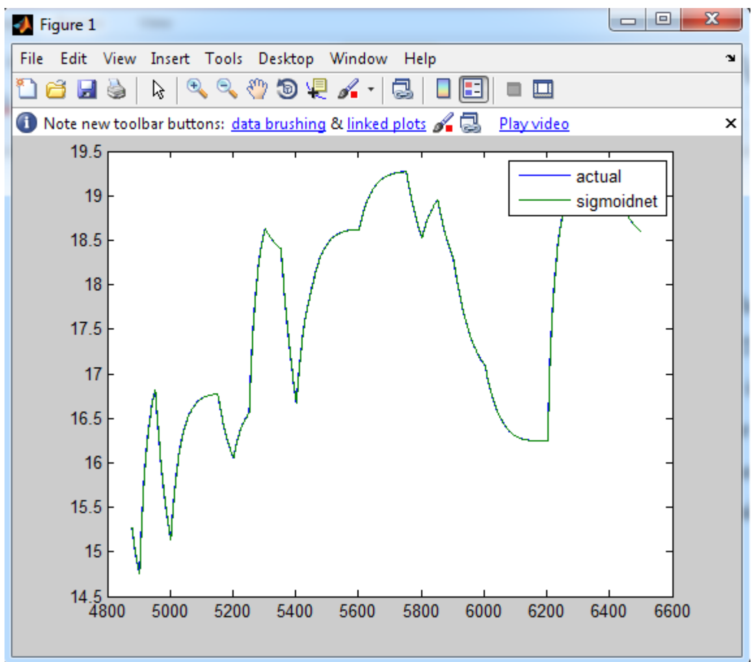

Figure 4.8: Comparison between the predicted and actual neural network model using sigmoidnet nonlinear estimator.

By analysing the data produced, it is noticeable that the value of the actual neural network model is very close to its predicted value. Thus, the model produce is good enough and can be used to represent the neural network part of the model.

The next step is to combine the OBFARX part with the neural network part of the model to produce the purposed model here, which is OBFARX-NN model. With the model combine, the RMSE value of the test set is evaluate and compared it with the RMSE value of the previous model which this project is based on, which is OBF-NN. Comparison from the graph itself shows that the difference in performance between the predicted and actual model is very little, showing good prediction.

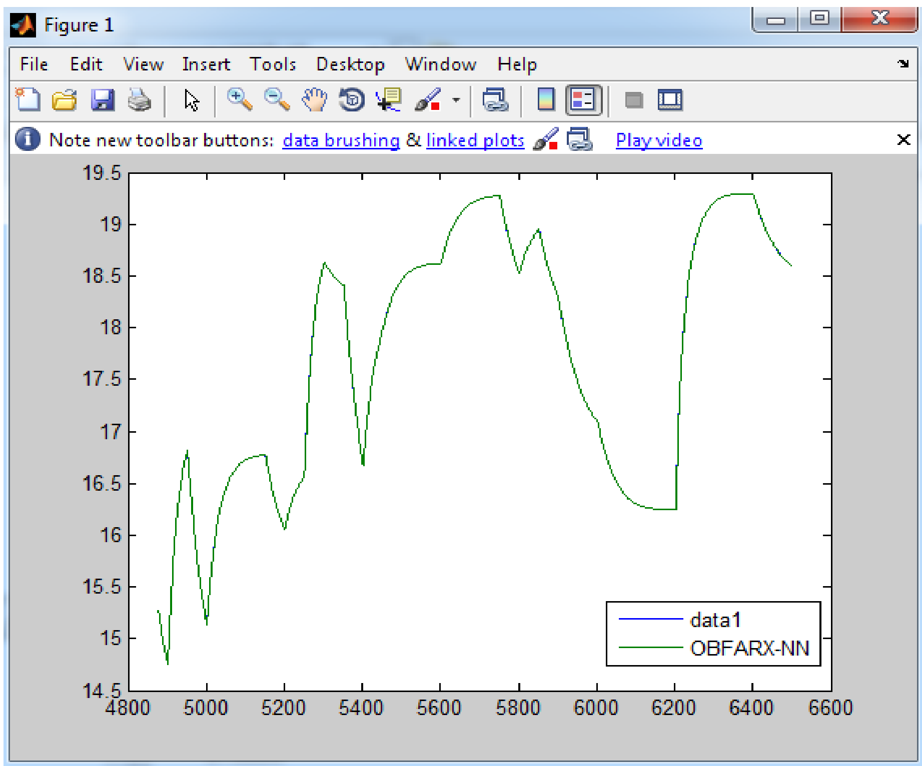

Figure 4.9: Comparison between the predicted and actual OBFARX-NN model

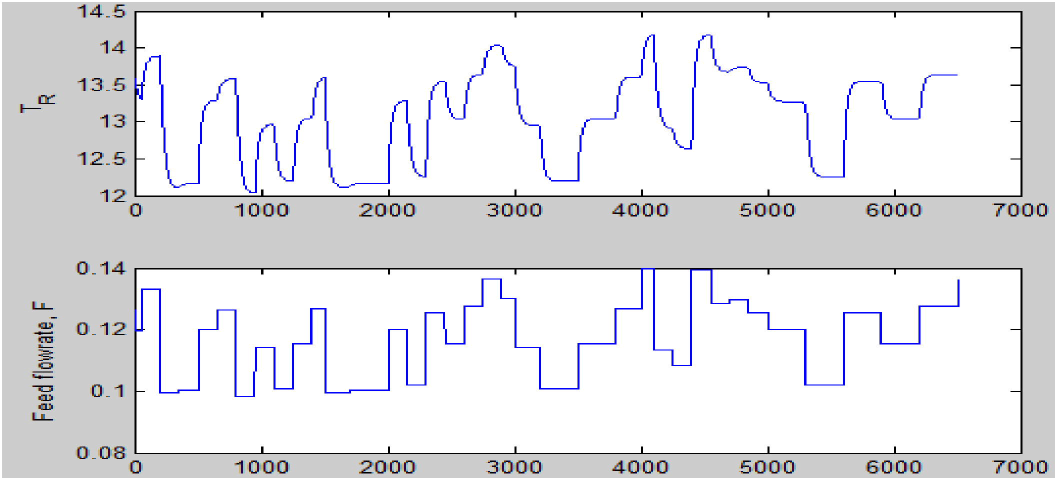

Figure 4.10: Input and output value of the process after applying OBFARX-NN model algorithm

Next, the results of the RMSE value shows decent enough performance when compared to the previous models especially OBF-NN. Although the value is much bigger than OBF-NN, it is not the final result since extrapolation is not conducted yet.

Table 4.2: Comparison between the RMSE values of different models for the case study

Case Study	**Conventional NN**	**Series Weiner-MLP**	**Parallel OBF-NN**	**Parallel OBFARX-NN**
Plant	0.000214	0.0083	0.0008	0.0160

Now to test the performance of the model, it will be subjugated to a series of extrapolation values (9%, 20%, and 28%). By doing these it is ensured that the model will produce a good result even under extreme extrapolation. In this case, the inlet flow rate, w1 is decrease to 9%, 20% and 28% of its original value with the purposed of evaluating the performance of the model. The results is as shown below

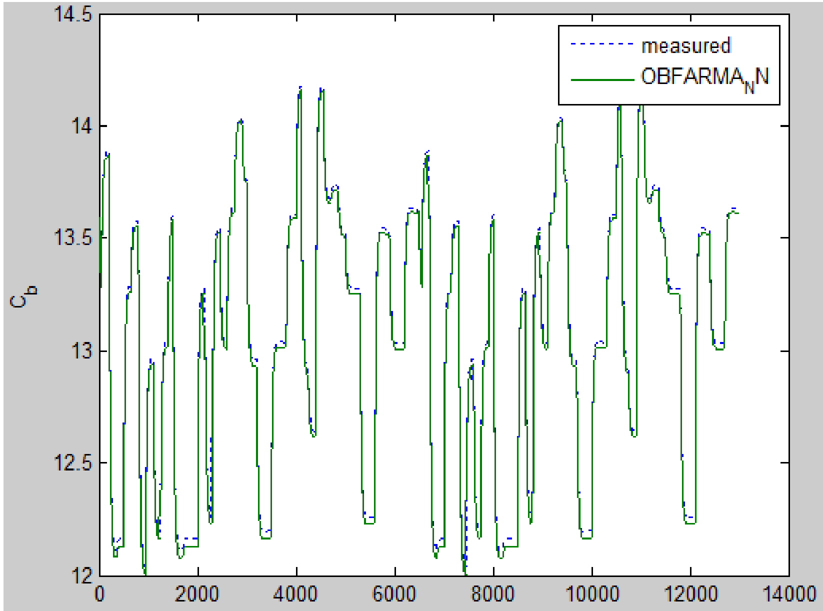

Figure 4.11: 9% decrease in w1 value

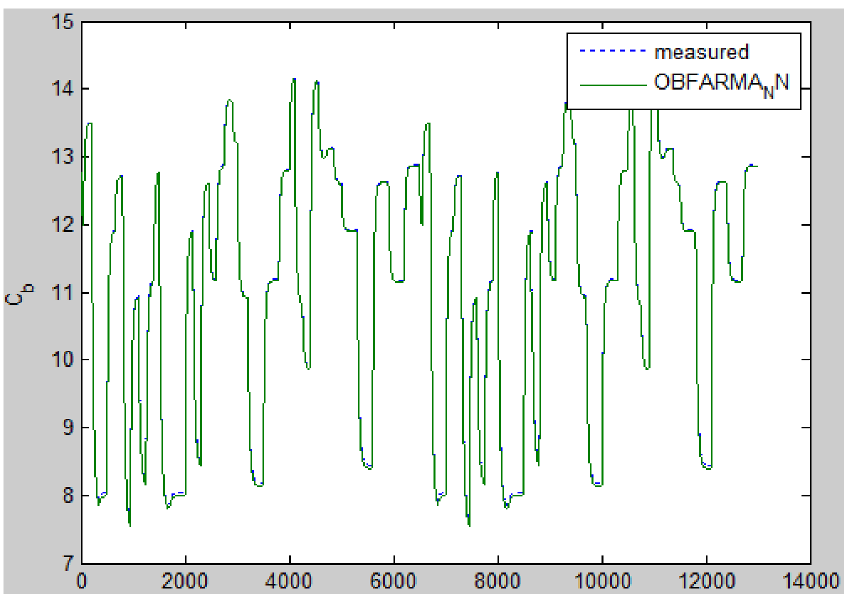

Figure 4.12: 20% decrease in w1 value

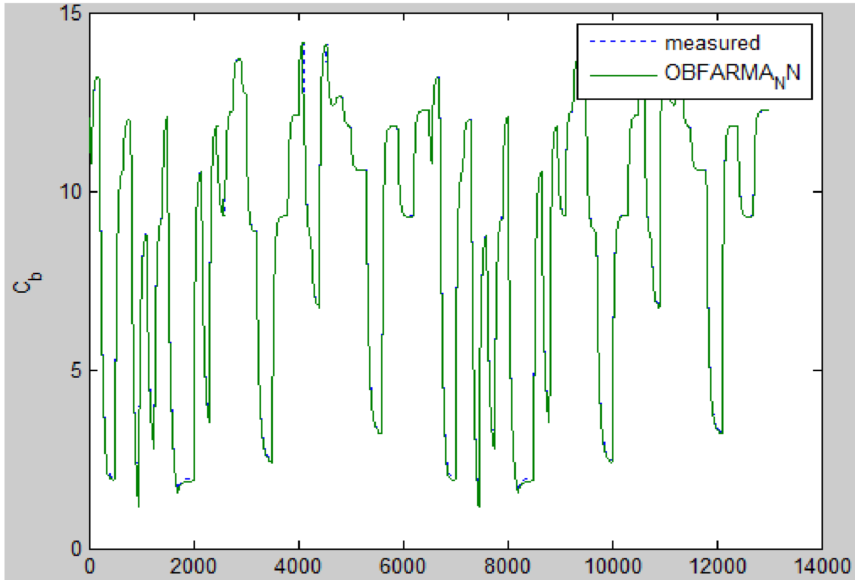

Figure 4.13: 28% decrease in w1 value

The RMSE values for the indicated extrapolation region are 0.0258, 0.0302 and 0.0424 respectively. Now by comparing these values with the previous model, which is OBF-NN, the performance of OBFARX-NN in the specified extrapolation region is better than OBF-NN model. Thus, it can be said that OBFARX-NN model works better for this case study when the parameters is varied continuously.

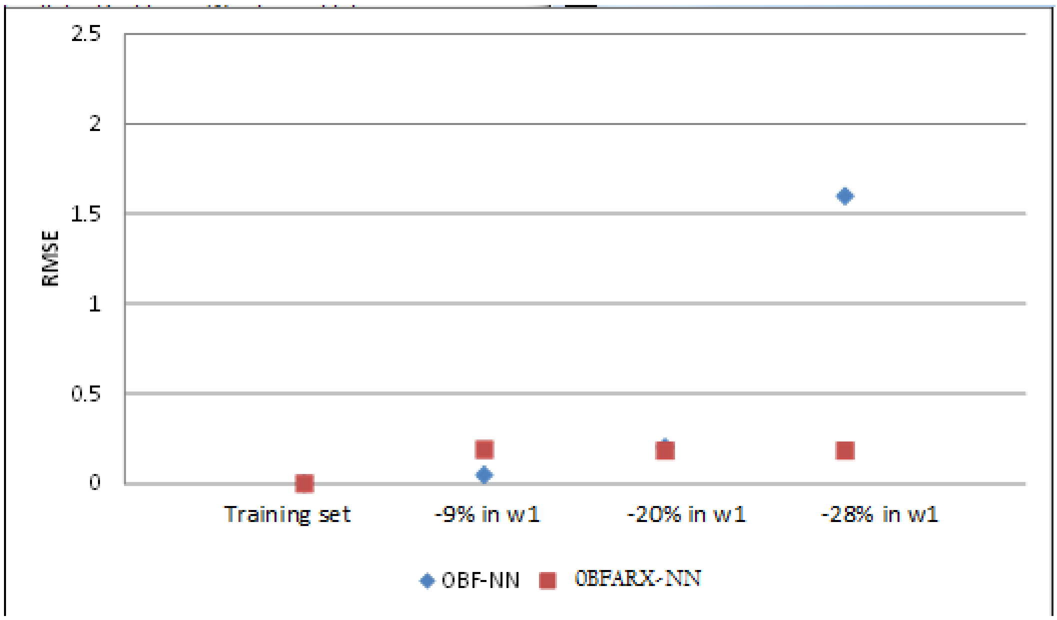

Figure 4.14: Comparison between different extrapolation values for OBFARX-NN and OBF-NN models

CHAPTER 5

CONCLUSION AND RECOMENDATION

In conclusion, this research project purpose is to establish a new combination of mathematical model for linear-nonlinear-based model in parallel in order to improve performance of process control and overcome limitations of established models present now. In this case, OBFARX-NN model development is evaluated to test whether the performance of the OBF-NN model can be further improved. Therefore, the analysis of the comparison in terms of performance of OBF-NN model and OBFARX-NN is indispensable because it will act as a stepping-stone to further help establishing better models in the future. Throughout the project, it is shown that forcing the models to extrapolate to regions beyond the original range encountered during the process will results in degradation of the overall process performance. It is still the same for the purposed model here, which is OBFARX-NN model. However, the degradation of the performance is not too severe compared to previously established models for this particular case study, which is the catalytic continuous stirred tank reactor (CSTR) model. This can be proof by analysing the values of RMSE for the specified extrapolation regions for both OBFARX-NN and OBF-NN model. All in all, the project achieve its objective and shows improve performance for OBFARX-NN compared to previous models.

REFERENCES

Lemma Dendena Tufa 2007, *Control Relevant System Identification Using Orthonormal Basis Filter Models,* Ph.D. Thesis, Universiti Teknologi Petronas, Malaysia

H Zabiri, M. Ramasamy, T D Lemma, and A Maulud 2011, *International Conference on Modelling, Simulation and Control: Nonlinear system identification using integrated linear-NN models: series vs. parallel structures,* Tokyo, Japan 2011. Perak, Malaysia

Zhao, H., Guiver, J., Neelakantan, R., and Biegler, L.T., *A nonlinear industrial model predictive controller using integrated PLS and neural net state-space model. Control Engineering Practice, 2001. 9(2): p. 125-133.*

Piche, S., Sayyar-Rodsari, B., Johnson, D., Gerules, M., *Nonlinear Model Predictive Control using Neural Networks, in Proc. IEE Contr. Syst. Mag. . 2000. p. 53-62.*

Yasui, T., Moran, A., Hayase, M., *Integration of linear systems and neural networks for identification and control of nonlinear systems,* in Proc. 35th SICE Annual Conference(SICE'96). 1996. Tottori.

Haslinda, Z., et al.,. 2011, *Nonlinear Model Predictive Control (NMPC) based on parallel OBF-NN models*. Universiti Teknologi Petronas, Malaysia

Haslinda, Z., et al., 2011, *Identification of nonlinear systems using parallel Laguerre-NN model.* Universiti Teknologi Petronas, Malaysia

S. C. Patwardhan and S. L. Shah, "From data to diagnosis and control using generalized orthonormal basis filters, Part I: Development of state observers," *Journal of Process Control*, vol. 15, pp. 819-835, 2005

Heuberger, P. S. C., Van den Hof, P. M. J., and B. Wahlberg, *Modeling and Identification with Rational Orthogonal Basis Functions*, Springer-Verlag London Limited, 2005.

Hsu, Y.L., and Wang, J.S., "A Wiener-type recurrent neural network and its control strategy for nonlinear dynamic applications", *Journal of Process Control 19* (2009) 942–953

S. C. Patwardhan and S. L. Shah, "From data to diagnosis and control using generalized orthonormal basis filters, Part I: Development of state observers," *Journal of Process Control*, vol. 15, pp. 819-835, 2005.

S. C. Patwardhan, S. Manuja, S. Narasimhan , S. L. Shah, "From data to diagnosis and control using generalized orthonormal basis filters, Part II: Model predictive and Fault Tolerant Control," *Journal of Process Control*, vol. 16, pp. 157-175, 2006.

Cutler, C. R., and Ramaker, B. C., "Dynamic Matrix Control – A Computer Control Algorithm," in *Automatica Control Conference*, San Fransisco, 1980

Saha, P., Krishnan, S.H., Rao, V.S.R., Patwardhan, S.C., 2004. "Modeling and predictive control of MIMO nonlinear systems using Wiener--Laguerre models". *Chemical Engineering Communication 191*, 1083-1119

A. Chubukjan, "Computational Aspects in Modelling Electromagnetic Field Parameters in Microstrips, " Ph.D. Thesis, University of Ottawa, 2000

Appendix 1

PROJECT ACTIVITIES																												
	WEEKS																											
	Final Year Project 1														**Final Year Project 2**													
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Project Scope Validation																												
Project Introduction & Initial research																												
Submission of Extended Proposal																												
Proposal Defence																												
Detail Study and modelling																												
Submission of Interim Draft Report																												
Additional research and modelling																												
Modelling and MATLAB analysis																												
Result analysis and discussion																												
Submission of progress report																												
Preparation for Pre-SEDEX																												
Pre-SEDEX																												

Submission of draft report																												
Submission of technical paper and dissertation																												
Oral presentation																												
Submission of project dissertation																												