INVESTIGATION ON EFFECT OF ERROR ORDER SELECTION IN SYSTEM IDENTIFICATION

MOHAMAD SYAHMI BIN MOHD BAKRI

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

SUPERVISOR DECLARATION

I hereby declare that I have read this project report and in my opinion this report is sufficient in terms of scope and quality for the award of the degree of Bachelor of Mechanical Engineering (Structure & Materials).

Signature :. ....

Name of Supervisor : DR. MD FAHMI BIN ABD SAMAD @ MAHMOOD

Date : ....

INVESTIGATION ON EFFECT OF ERROR ORDER SELECTION IN SYSTEM IDENTIFICATION

MOHAMAD SYAHMI BIN MOHD BAKRI

This report is submitted
in fulfillment of the requirement for the degree of
Bachelor of Mechanical Engineering (Structure & Material)

Faculty of Mechanical Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

MAY 2017

DECLARATION

I declare that this project report entitled "Investigation On Effect Of Error Order Selection In System Identification" is the result of my own work except as cited in the references

Signature : ....

Name : MOHAMAD SYAHMI BIN MOHD BAKRI

Date : ....

DEDICATION

To my beloved mother and father

ACKNOWLEDGEMENT

First and above all, I praise Allah the almighty for providing me this opportunity and granting me the capability to proceed successfully. This project appears in its current form due to the assistance and guidance of several people. Special appreciation goes to my supervisor, Dr. Md Fahmi bin Abd Samad @ Mahmood, for his relentless support, thoughtful guidance and correction of the project. I attribute the level of my bachelor degree to his encouragement and effort and without him, this project would not have been completed or written.

My deepest gratitude goes to my beloved parents, Mohamad Bakri Bin Abd Razak and Anim Binti Hamdan for their material and spiritual support in all aspects of my life. Thank you both for giving me strength to reach for the stars and chase my dreams. I also would like to thank to all my siblings for their endless love, prayers and encouragement. I am particularly grateful to my sweetheart for his devoted support throughout my study.

Last but not least, thanks to all my close friends Jamry Bin Samut for the joyful gatherings and all his supports. I deeply appreciate to all my friends that helped me stay through these difficult years. Their support and care helped me overcome setbacks and stay focused on my graduate study. I greatly value their friendship. To those who indirectly contributed in this research, your kindness means a lot to me. Thank you very much.

ABSTRACT

System identification is a technique or application that aims to develop mathematical models for dynamical systems using measured input and output signals. There are 4 important steps which is the observed data, model, parameter estimation and validation. In this project, model structure selection is important because the objective is to find the effect of the error order selection on system identification. Linear model which is ARX and ARMAX model is used to make the simulation to investigate effect of error order. This project was carried out using the Graphical User Interface (GUI) in MATLAB application. Firstly, 3 equations were randomly created with supervisor guide and data acquisition was run to generate 500 data by using the MATLAB software command and save it into folder for next step use. After that, system identification toolbox is open and imports the data that generate from the equation. Next step is modelling where the data use ARMAX model and different sequence in error order or n_c. This project result is based on 5 performance indicator like Means Square Error (MSE), Model Output, Model Residuals, Akaike's Final Prediction Error (FPE) and Parameter Values. This entire indicator will pop out after click on button indicator or from the data of each model. Based on result that acquire from this investigation, the effect or error order selection have a very small difference compared to the true model. This is maybe because the data was not completely scatter and need to bigger. So the conclusion is, there is small effect of error order selection in system identification.

ABSTRAK

Pengenalanpastian sistem adalah satu teknik atau aplikasi yang bertujuan untuk membangunkan model matematik untuk sistem dinamik menggunakan isyarat input dan output yang diukur. Terdapat 4 langkah penting yang merupakan data yang diperhatikan, model, parameter anggaran dan pengesahan. Dalam projek ini, pemilihan struktur model adalah penting kerana tujuannya adalah untuk mencari kesan pemilihan turutan ralat pada pengenalpastian sistem. Model linear yang merupakan model ARX dan ARMAX adalah digunakan untuk membuat simulasi untuk menyiasat kesan nilai turutan ralat. Projek ini telah dijalankan dengan menggunakan antara muka grafik pengguna (GUI) dalam perisian MATLAB. Pertama, 3 persamaan diciptakan secara rawak dengan panduan penyelia dan perolehan data telah dijalankan untuk menjana 500 data dengan menggunakan arahan perisian MATLAB dan simpan ke dalam folder untuk digunakan langkah seterusnya. Selepas itu, toolbox pengenalpastian sistem dibuka dan mengimport data yang dijana membentuk persamaan. Langkah seterusnya ialah memodelkan data dengan menggunakan model ARMAX dan menggunakan turutan yang berbeza untuk ralat atau n_c. Hasil projek ini adalah berdasarkan kepada 5 petunjuk prestasi seperti Means Square Error (MSE), keluaran model, sisa model, Ramalan Ralat Akhir Akaike (FPE) dan Parameter. Kesemua penunjuk akan ditunjukkan selepas klik pada penunjuk butang atau dari maklumat setiap model. Berdasarkan keputusan yang diperolehi daripada penyiasatan ini, kesan turutan ralat sangat kecil apabila dibanding dengan model sebenar. Ini masih boleh diterima kerana mungkin data tidak sepenuhnya berselerak dan data perlu lebih besar. Jadi kesimpulannya ialah, terdapat kesan kecil pada pemilihan turutan ralat dalam pengenalpastian sistem.

CONTENT

LIST OF FIGURES

LIST OF ABBEREVATIONS

ARX	Exogenous Input
ARMAX	Auto Regressive Moving Average Exogenous Input
MATLAB	Matrix Laboratory
GUI	Graphical User Interface
RELS	Recursive Extended Least Square
PEM	Prediction Error Minimization
y(t)	Output at time t.
n_a	Number of poles.
n_b	Number of zeroes plus 1.
n_k	Number of input samples that occur before the input affects the output, also called the dead time in the system.
e(t)	White-noise disturbance value.
n_c	Number of C coefficients.

CHAPTER 1

INTRODUCTION

1.1 Background

In recent years, system identification has been discussed and gets attention from many universities. This is because system identification was start use in some industries. In the other word, system identification can be categories as the basic necessity in regions, for example, control, correspondence, power system and instrumentation for acquiring a model of a system of interest or another system to be produced.

System identification is the art and science of building mathematical models of dynamic systems from observed input–output data. It can be seen as the interface between the real world of applications and the mathematical world of control theory and model abstractions. As such, it is a ubiquitous necessity for successful applications (Ljung, 2010). Actually system identification is a one term in control area that refers as a technique constructing model from observed data.
There are four basic ingredients in system identification:

1) The observed data
2) Model structure
3) A criterion of fit
4) Validation

When formulating and solving an identification problem it is important to have the purpose of the identification in mind. In control problems the final goal is often to design control strategies for a particular system. There are, however, also situations where the primary interest is to analyse the properties of a system. The purpose of the identification is to design a control system. The character of the problem might vary widely depending on the nature of the control problem (Astrom, 1971). For the example, design a stable regulator or design a control program for optimal transition from one state to another.

In this project, the main objective is to investigate the effect of error order selection in system identification. The error order that only occur in ARMAX model will be discuss to. So to make this experiment happen, this experiment will be carry out using MATLAB‘s system identification graphical user interface (GUI) named _IDENT‘ found in System Identification Toolbox.

1.2 Problem Statement

There are many aspects that we can discuss in system identification such as type of model, the techniques and many more. System identification deals with the problem of building mathematical models of dynamic system based on observed data from the system. The subject is thus part of basic scientific methodology and since dynamical system is abundant in our environment, the techniques of system identification have a wide application area (Ljung., 1987). Besides, the most important in system identification is to achieve model from system data and from the selection of error order will cause an accurate model. This experiment will discuss about vary the value of error order that happen to the accurate model. Since the presence of measurement error order only occur in ARMAX model, the discussion about why wrong selection of error order will cause the developed ARMAX model unable to represent system behaviour.

ARMAX model are widely used in identification and are a standard tool in control engineering for both system description and control design. These models, however, can be non-realistic in many practical contexts because of the presence of measurement errors that play an important role in applications like fault diagnosis and optimal filtering. ARMAX models can be enhanced by introducing also additive error terms on the input and output observations. How much the error order effects the output between the actual result and simulation result in ARMAX model. Besides that, we want to know what the different output result when the wrong selection of error order is estimate in the system.

1.3 OBJECTIVE

The objectives of this project are as follows:

1. To investigate the effect of error order on performance of identification.
2. To perform system identification using linear difference equation which is auto regressive moving average exogenous input (ARMAX) model.

1.4 Scope of Project

The scopes of this project are:

1. Identification will be performed using GUI (Graphical User Interfaces) 'ident' on MATLAB (matrix laboratory) software.
2. Performance comparison will be made from various perspectives such as final prediction error and established loss function.

CHAPTER 2

LITERATURE REVIEW

2.1 System Identification

System identification is the art and science of building mathematical model of dynamic systems from observed input-output data. It can be seen as the interface between the real world of application and the mathematical world of control theory and model (Ljung, 2010). Or in other way, system identification is the process of developing or improving a quantitative numerical model from a set of input and output data that represents the response of a dynamic system. It is necessary to use model to describe the relationships among the system variables. The developed model has the trademark performance similar like the unknown system.

The parameter estimation step decides inside of the arrangement of models, the model that is the best guess or gives the best clarification of the observed data. The estimation of the model parameters relates to the minimization of the chosen criterion. The decision of basis relies on upon the accessible data about and the motivation behind the model. Model validation is conceivably the most vital stride in the model building arrangement. It is likewise a standout amongst the most disregarded. Regularly the approval of a model appears to comprise of simply citing the measurement from the fit (which measures the part of the aggregate variability in the reaction that is represented by the model).

As mention before, there are four steps in system identification which is the observed data, model structure, a criterion of fit and validation and the loop of this all four step can be determine as Figure 2.1 and will be discuss more detail in next.

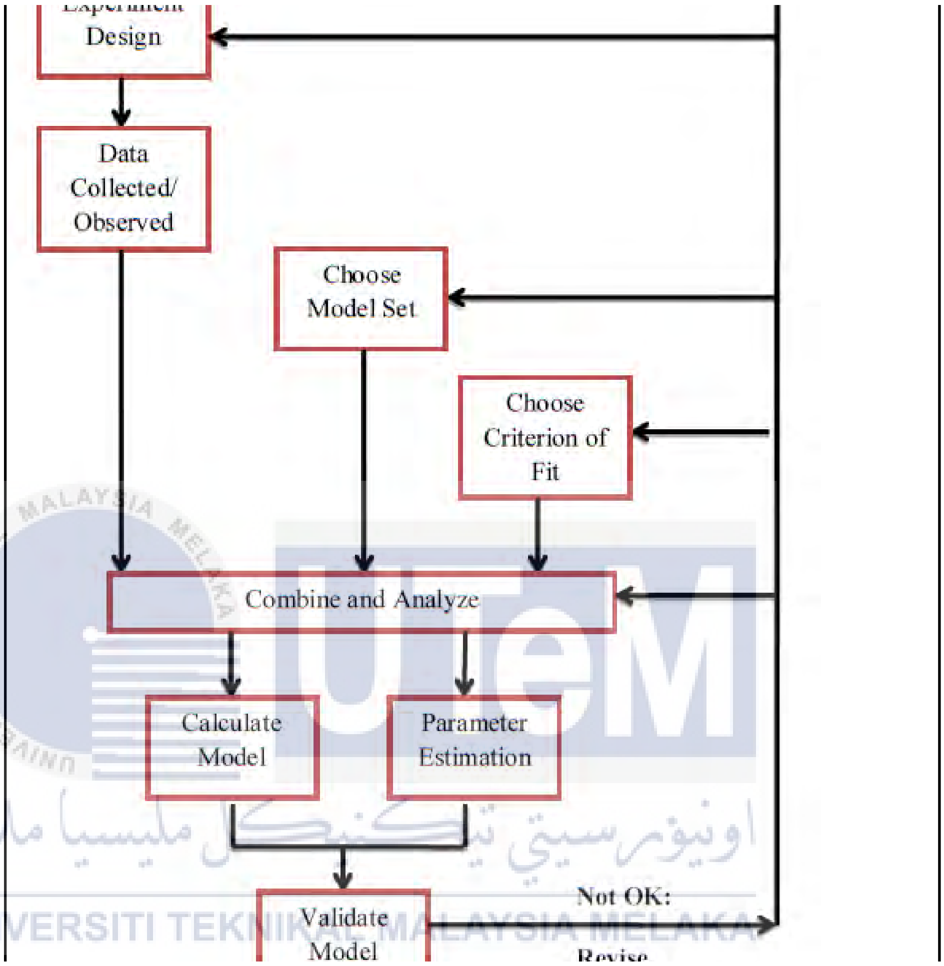


Figure 2.1: Loop of System Identification Step

2.1.1 The Observed Data

If we want to build a model for a system, we should get data or information about it. This can be done by just watching the natural fluctuation (e.g., vibration analysis of a bridge that is excited by normal traffic), but most often efficient to set up dedicated experiments that is actively excites the system (e.g., controlled excitation that optimize his own goal (for example, minimum cost, minimum time, or minimum power consumption for a given measurement accuracy) within the operator constraints (e.g., the excitation should remain below a maximum allowable level). The quality of the final result can depend heavily on the choices that are made

2.1.2 Model Structure

A choice should be made within all the possible mathematical models that can be used to represent the system. Again a wide variety of possibilities exist, such as below;

- Parametric versus non-parametric models

In a parametric model, the system is described using a limited number of characteristic quantities called the parameter of the model, whereas in a non-parametric model the system is characterized by measurements of a system function at a large number of points. Example of parametric models is the transfer function of a filter described by its poles and zeros and the motion equation of a piston. An example of a non-parametric model is the description of a filter by its impulse response at a large number of points.

Usually it is simpler to create a non-parametric model than a parametric one because the modeller needs less knowledge about the system itself in the former case. However, physical insight and concentration of information are more substantial for parametric models than for non-parametric ones.

- White box models versus black box models

In the construction of a model, physical laws whose availability and applicability depend on the insight and skills of the experimenter can be used (Kirchhoff's laws, Newton's laws, etc.) Specialized knowledge related to different scientific fields may be brought into this phase of the identification process. The modelling of a loud speaker, for example, requires extensive understanding of mechanical, electrical and acoustical phenomena. The result may be a physical model, based on comprehensive knowledge of the internal functioning of the system. Such a model is called a white box model.

The choice between the different methods depends on the aim of the study: the white box approach is better for gaining insight into the working principles of a system, but black box model may be sufficient if the model will be used only for prediction of the output.

Although, as a rule of thumb, it is advisable to include as much prior knowledge as possible during the modelling process, it is not simple to express this information if the polynomial coefficients are used as parameters.

- Linear models versus non-linear models.

In real life, almost every system is non-linear. It is because mostly approximated by linear models, assuming that in the operation region the behaviour can be linearized. This kind of approximation makes it possible to use simple models without jeopardizing properties that are of importance to the modeller. This choice depends strongly on the intended use of the model. For example, a non-linear model is needed to describe the distortion of an amplifier but a linear model will be sufficient to represent its transfer characteristics if the linear behaviour is dominant and is the only interest.

- Linear in the parameter versus non-linear in the parameters

A model is called linear in the parameter if there exists a linear relation between these parameters and the error that is minimized. This does not imply that the system itself is linear. For example $\mathcal{E} = y - (a_1u + a_2u^2)$ is linear in the parameter a_1 and a_2 but describes a non-linear system.

$$\varepsilon(j\omega) = Y(j\omega) - \frac{a_0 + a_1 j\omega}{b_0 + b_1 j\omega} U(j\omega) \tag{2.1}$$

Equation 2.1 describe a linear system but the model is non-linear in the b_1 and b_2 parameters. Linearity in the parameters is a very important aspect of models because it has a strong impact on the complexity of the estimators if a (weighted) least squares cost function is used. In that case, the problem can be solved analytically for models that are linear in the parameters so that an iterative optimization problem is avoided.

2.1.3 Criterion Of Fit

Once a model structure is chosen (e.g., a parametric function model), it should be matched as well as possible with the available information about the system. Mostly, this is done by minimizing a criterion that measures a goodness of the fits. The choice of this criterion is extremely important because it determines the stochastic properties of the final estimator. For the example like resistance, many choice are possible and each of them can lead to a different estimator with its owns properties. Usually, the cost function defines a distance between the experiment data and the model.

2.1.4 Validation

Finally, the validity of the selected model should be tested: does this model describe the available data properly or are there still indications that some of the data are not well modelled, indicating remaining model errors? In practice, the best model (the smallest errors) is always preferred. Some tools that guide the user through this process by separating the remaining errors into different classes, for example, modelled linear dynamics and non-linear distortions. From this information, further improvement of the model can be proposed, if necessary.

During the validation tests it is always important to keep the application in mind. The model should be tested under the same conditions as it will be used later. Extrapolation should be avoided as much as possible. The application also determines what properties are critical.

2.1.5 System Identification Step Overview

This brief overview of the identification process shows that it is a complex task with a number of interacting choices. It is important to pay attention to all aspects of this procedure, from the experiment design to the model validation, in order to get the best results. The reader should be aware that besides these actions, other aspects are also important. A short inspection of the measurement setup can reveal important shortcomings that can jeopardize a lot of information.

Good understanding of the intended applications helps to set up good experiments, and is very important to make the proper simplifications during the model building process. Many times, choices are made that are not based on complicated theories but are dictated by the practical circumstances. In these cases a good theoretical understanding of the applied methods will help the user to be aware of the sensitive aspect of his techniques. This will enable him to put all his effort on the most critical decisions. Moreover, he will become aware of the weak points of the final model.

2.2 Type of Model

Model is a relationship between observed quantities. In loose terms, a model allows for prediction of properties or behaviours of the object. Typically the relationship is a mathematical expression, but it could also be a table or a graph.

Model in system identification also be known as mathematical model. Model is represented in mathematical terms of behavior of real device and object. Model can be used on various types of science and engineering such as physic, biology and electrical engineering. Eykhoff (1974) defined a mathematical model as a representation of the essential aspects of an existing system (or a system to be constructed) which presents knowledge of that system in usable form. Mathematical models can take many forms, including but not limited to dynamical systems, statistical models, differential equations, or game theoretic models. These and other types of models can overlap with a given model involving variety abstract structures. In general, mathematical model may include logic models, as far as logic is taken as a part of mathematics

Choosing a suitable model structure is prerequisite before its estimation. The choice of model structure is based upon understanding of the physical systems. Three types of models are common in system identification that is the black-box model, grey-box model, and user-defined model. The black-box model assumes that systems are unknown and all model parameters are adjustable without considering the physical background. The grey-box model assumes part of the information about the underlying dynamics or some of the physical parameters are known and the model parameters might have some constraints. The user-defined model assumes commonly used parametric models that cannot represent the model that one wants to estimate.

2.2.1 ARX

Autoregressive regressive model with exogenous inputs (ARX) is often used for modelling of controlled systems especially in self-tuning control. Popularity of ARX models stems mainly from plausibility of least squares (LS) for estimating its parameters. The ARX model therefore is preferable, especially when the model order is high.

(Vincent, 2005) define ARX model has been proposed as a representation of the speech production process and combined with the RK (Rosenberg-Klatt) glottal source model. However, this method has to cope with a very difficult optimization problem for which no practical satisfactory solution has been given.

The ARX model structure shown in equation 2.2:

$$y(t)+a_1 yy(t-1)+...+a_{n_a}\, y(t-n_a)= b_1 u(t-n_k)+...+ b_{n_b} u(t-n_b-n_k+1)+e(t) \qquad (2.2)$$

The parameters n_a and $n_b$are the orders of the ARX model and n_k is the delay.

$y(t)$— Output at time t.

n_a — Number of poles.

n_b — Number of zeroes plus 1.

n_k — Number of input samples that occur before the input affects the output, also called the *dead time* in the system.

$y(t-1)\ldots y(t-n_a)$ — Previous outputs on which the current output depends.

$u(t-n_k)\ldots u(t-n_k-n_b+1)$ — Previous and delayed inputs on which the current output depends.

$e(t)$ — White-noise disturbance value.

A more compact way to write the difference equation is

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

q is the delay operator. Specifically,

$$A(q)=1+a_1 q^{-1}+\ldots+a_{n_a} q^{-n_a}$$

$$B(q)=b_1+b_2 q^{-1}+\ldots+b_{n_b} q^{-n_b+1}$$

2.2.2 ARMAX

ARMAX (auto regressive moving average exogenous input) model in particular, can lead to reliable estimates of the dynamic characteristics of a structure even in the presence of strong measurement noise. ARMAX models have been used extensively to represent the relationship of the system output with system input in the presence of noise in many linear dynamic systems. The parameters are usually estimated by means of the recursive extended least square (RELS) method (Fung, 2003).

ARMAX model structure is:

$y(t)+a_1y(t-1)+\ldots+a_{na}y(t-n_a)=b_1u(t-n_k)+\ldots+b_{nb}u(t-n_k-n_b+1)+ c_1e(t-1)+\ldots.+c_{nc}e(t-n_c)$ $+e(t)$

A more compact way to write the difference equation is:

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

where,

$y(t)$— Output at time t.

n_a — Number of poles.

n_b — Number of zeroes plus 1.

n_c — Number of error order.

n_k — Number of input samples that occur before the input affects the output, also called the *dead time* in the system.

$y(t-1)\ldots y(t-n_a)$ — Previous outputs on which the current output depends.

$u(t-n_k)\ldots u(t-n_k-n_b+1)$ — Previous and delayed inputs on which the current output depends.

$e(t-1)\ldots e(t-n_c)$ — White-noise disturbance value.

The parameters na, nb, and nc are the orders of the ARMAX model, and nk is the delay. q is the delay operator. Specifically,

$$A(q)=1+a_1q^{-1}+\ldots+a_{n_a}q^{-n_a}$$

$$B(q)=b_1+b_2q^{-1}+\ldots+b_{n_b}q^{-n_b+1}$$

$$C(q)=1+c_1q^{-1}+\ldots+c_{n_c}q^{-n_c}$$

If data is a time series that has no input channels and one output channel, then ARMAX calculates an ARMA model for the time series

$$A(q)y(t)=C(q)e(t)$$

2.3 Error Order

Error order only occurs in ARMAX model and it can be known as n_c in the equation. Besides that, it can be a number of model's parameter in the C vector.
n_c must be zero if want to estimating a model using frequency domain data. As we can see, the objective of this investigate is to find the effect of error order, so ARMAX model will be used to perform the simulation and discuss what the effect is.

2.4 Parameter Estimation

Simultaneous with the development of increased need of parameter estimation, computers have been built that make parameter estimation practicable for great array of application. Estimation was first extensively discussed by Legendre in 1806 and Gauss in 1809 (V.Beck, 1977).The problem of identifying a dynamic process has received considerable attention in recent years.

The various techniques developed for input-output data collection and evaluation range from the simplest form of the deterministic procedures to elegant numerical and statistical methods based on results of optimal estimation theory. It may be stressed that in the field of parameter estimation, the least squares technique has reached a significant level of popularity and perfection (Strejc, 1980).

The least squares method is a statistical means for finding a line or curve of "best" fit for a set of observed measurements. When the collected data are plotted as points on a graph and appear to fall near some line drawn through the midst of them, the distance of the points from the line varies with the line chosen. The average of the squares of these distances is taken as a measure of "goodness of fit" of the line. The "best-fit" line is that one for which this mean square deviation is minimal. It can be shown that the desired line must pass through the arithmetic mean ($\bar{x}$,$\bar{y}$) of the array. The least squares method can be extended to find a second-degree curve to fit a given set of data and generalized to other curve.

For the black-box model, deciding the delay and model order for the parametric model is commonly an experimentation process. The accompanying is an arrangement of steps that can prompt a suitable model.

1. Obtain helpful data about the model request by watching the quantity of reverberation tops in the nonparametric recurrence reaction capacity. Ordinarily, the quantity of tops in the size reaction breaks even with a large portion of the order.

2. Obtain a sensible evaluation of postponement utilizing relationship examination and/or by testing sensible qualities in a medium size ARX model. Pick the postponement that gives the best model fit in prediction error or other fit rule.

3. Test different ARX model requests with this deferral picking those that give the best fit.

4. Since the ARX model portrays both the system flow and commotion properties utilizing the same arrangement of posts, the subsequent model may be pointlessly high all together. By plotting the zeros and posts (with the vulnerability interims) and searching for cancelations you can decrease the model request. The subsequent request of the posts and zeros are a decent beginning stage for ARMAX, OE and/or BJ models with these requests utilized as the B(q) and F(q) model parameters and first or second request models for the commotion qualities.

5. If a suitable model is not got as of right now endeavour to figure out whether there are extra flags that may impact the output. Estimations of these signals can be joined as additional input signs.

.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter describes the methodology used in this project to investigate the effect of error order selection in system identification. The flow chart of the project is shown in Figure 3.1. This project starts by studying system modelling; only two types of model are chosen that is auto regressive with exogenous input (ARX) and auto regressive moving average with exogenous (ARMAX) model.

For this investigation, MATLAB software is used to achieve the objective. For the early step, only familiarization on the software like try follows some tutorial and video to make sure there are no problem and error during running the research. Trial run and data simulation is the most important part because from this process, data will be found and some analysis will be makes to find the result.

Since the main objective is to find the error order made effect, so the simulation needed to be stressed and tried many times to achieve the effect of error order on model selection. Finally some research discussion with supervisor will be held based on the result that is obtained from the simulation.

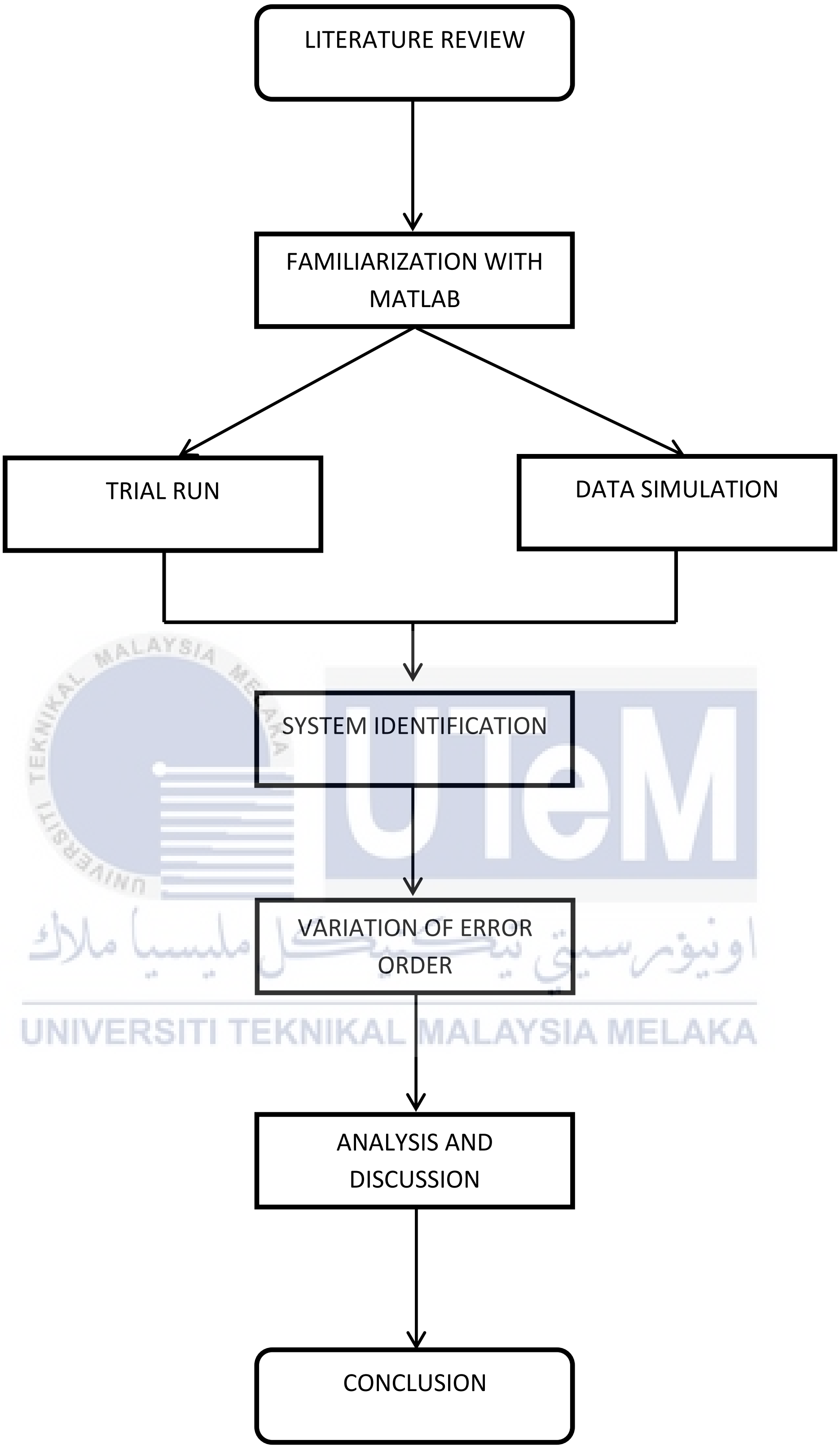


Figure 3.1: Flow chart of the methodology.

3.2 Familiarization With MATLAB

MATLAB is software that uses high performance language for technical computing that developed by Math Works. It is one interactive system whose basic data element is an array that does not require dimensioning. This allows you one solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non-interactive language. It also integrates computation, visualization and programming to the user. Some function of MATLAB is:

- Math and computation
- Algorithm development
- Modelling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including Graphical User Interface building

Like other language program, MATLAB also provide many varieties of extension and program libraries to ease the user. Or in other word, it is call toolbox and it is define as a collection of routines that are designed to do common things. This toolbox is more involved than the simple one-lines and normal programming syntax that the base MATLAB has.

MATLAB toolbox that is used in this investigation is system identification toolbox, it provides identification techniques such as maximum likelihood, prediction-error minimization (PEM), and subspace system identification. To represent nonlinear system dynamics, one can estimate Hammerstein-Wiener models and nonlinear ARX models with wavelet network, tree-partition, and sigmoid network nonlinearities.

System identification toolbox can do work such as:

- Model Identification From Data
- Linear Model Identification
- Non-Linear Model Identification
- Parameter Estimation in User-Defined Models
- Online Parameter Estimation
- Time-Series Data Modelling

3.3 Trial Run

This section will show about all step of first trial run with the MATLAB with following the interactive demo of system identification tool guide that system identification tool window provided. Below are step by step to fulfil the first trial run as mention on methodology flow chart.

First step, type —*ident”* in the MATLAB command window as Figure 3.2. This action opens the System Identification application. A GUI for the toolbox will appear as Figure 3.3.

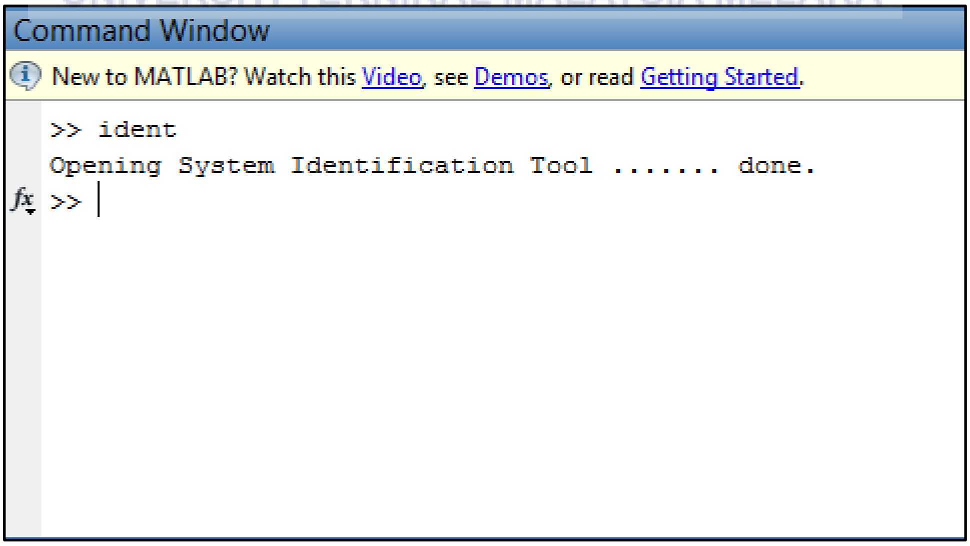


Figure 3.2: Command Window

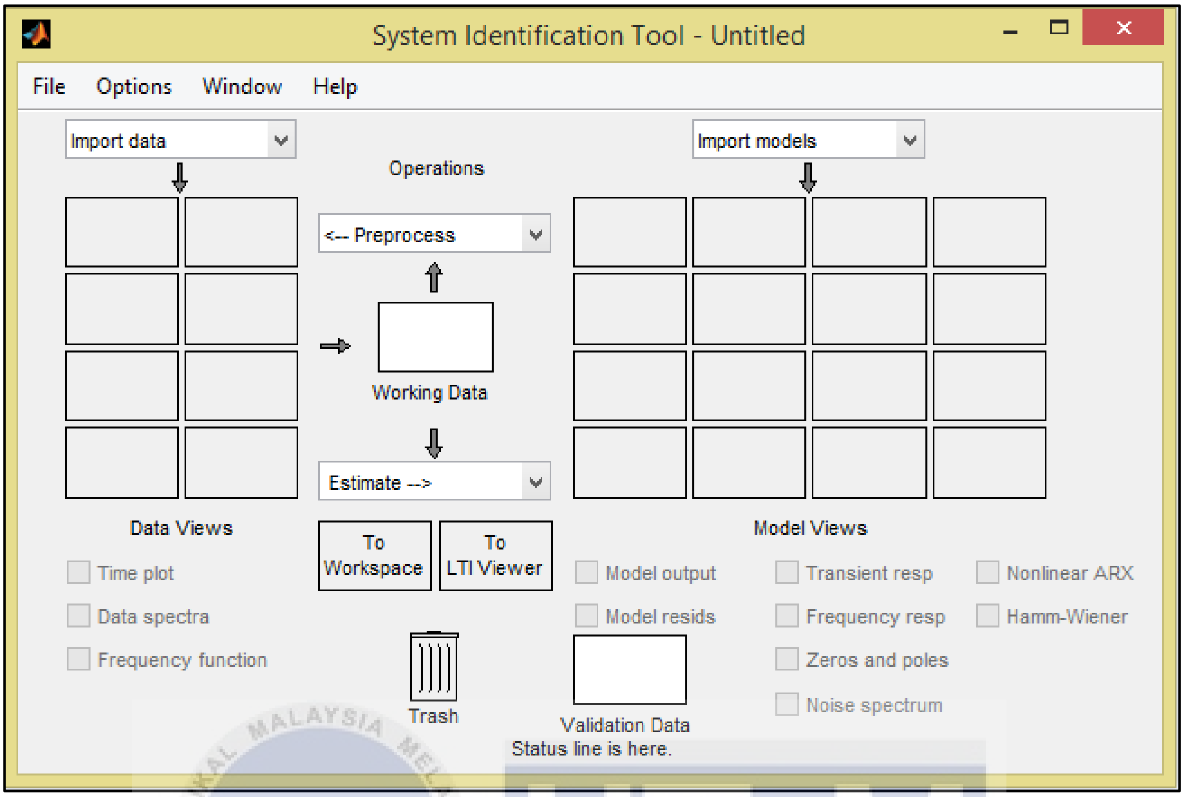


Figure 3.3: System Identification Toolbox

To Import data, just write load dryer2 in command window as Figure 3.4

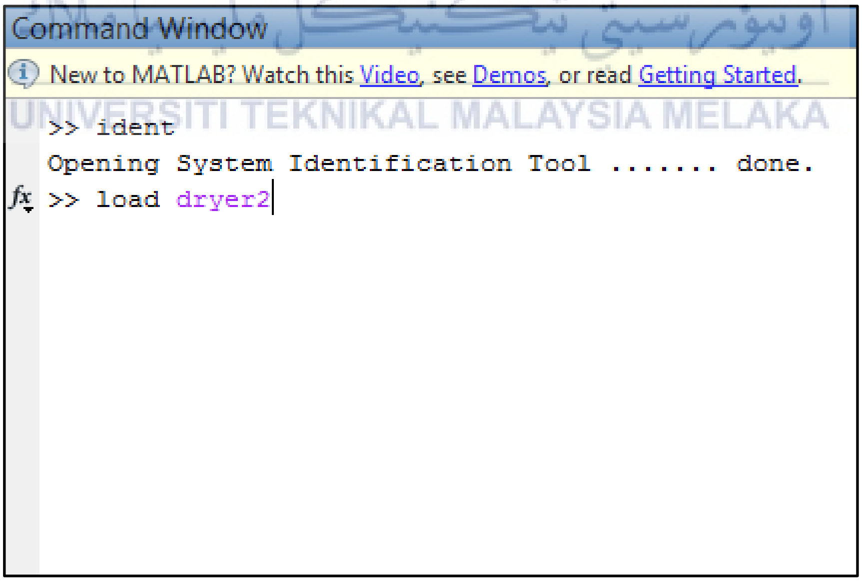


Figure 3.4: Import Data Command Window

In system identification tool, press the pop up menu Import Data and select Time Domain Data like Figure 3.5 and dialog box will open, and then edit the input, output, sampling interval and data name like Figure 3.6.

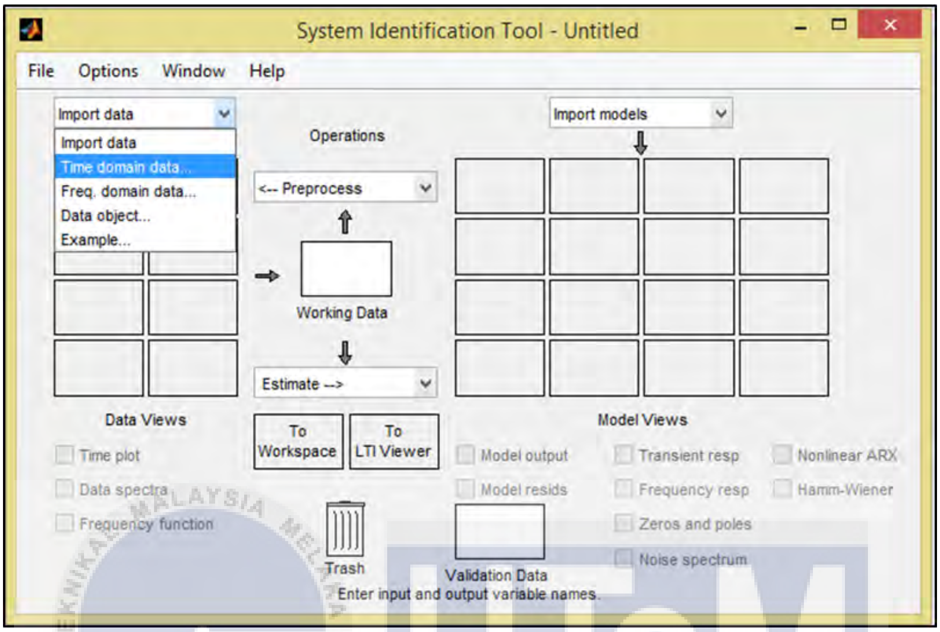


Figure 3.5: System Identification Dialog Box

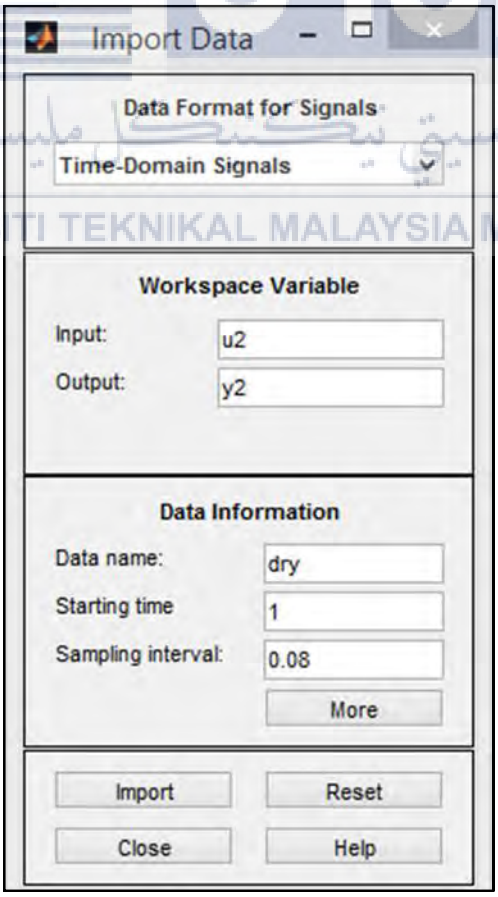


Figure 3.6: Import Data Dialog

After click import button on import data dialog, data will present on data board in system identification like Figure 3.7.

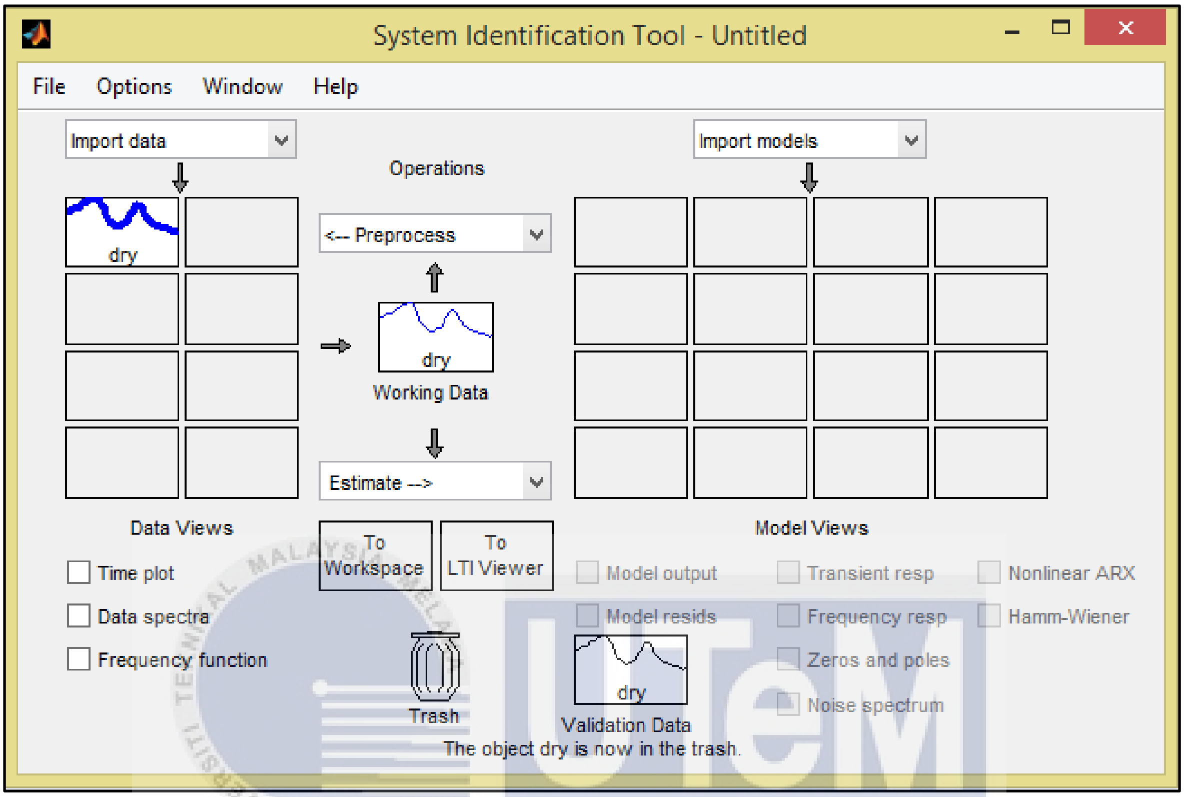


Figure 3.7: Appear data on data board

Time plot figure will shows like Figure 3.8 after click on time plot for examination purpose.

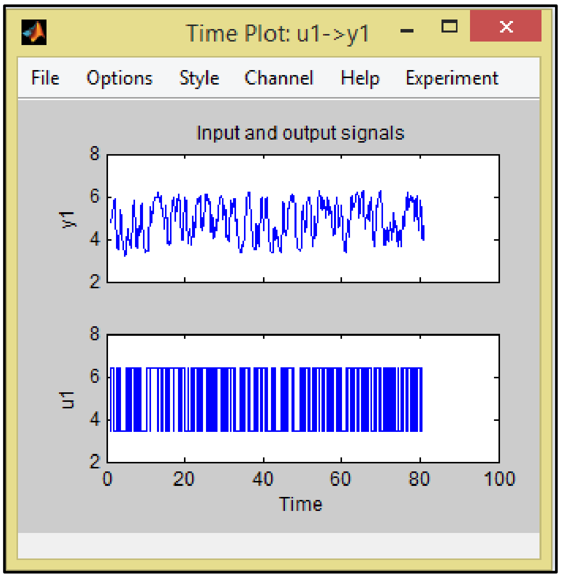


Figure 3.8: Time Plot

Select remove mean from preprocess popup menu like Figure 3.9 for insert new data set that has zero mean and new data will appear like Figure 3.10. The data is also automatically inserted in time plot figure.

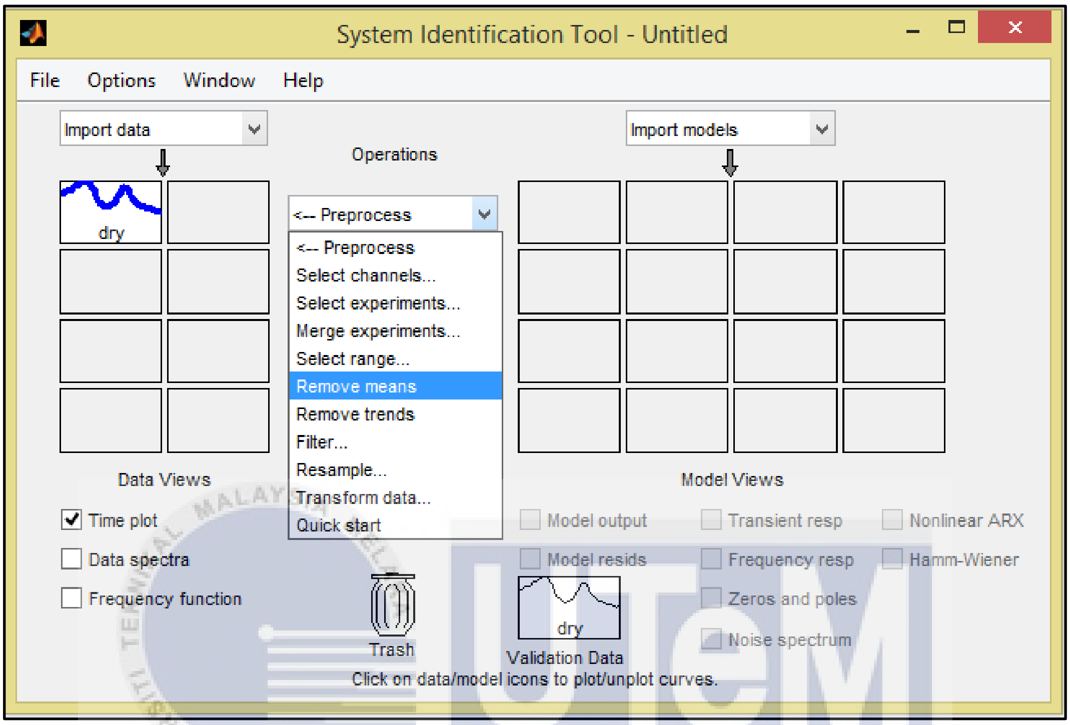


Figure 3.9: Remove means on preprocess popup menu

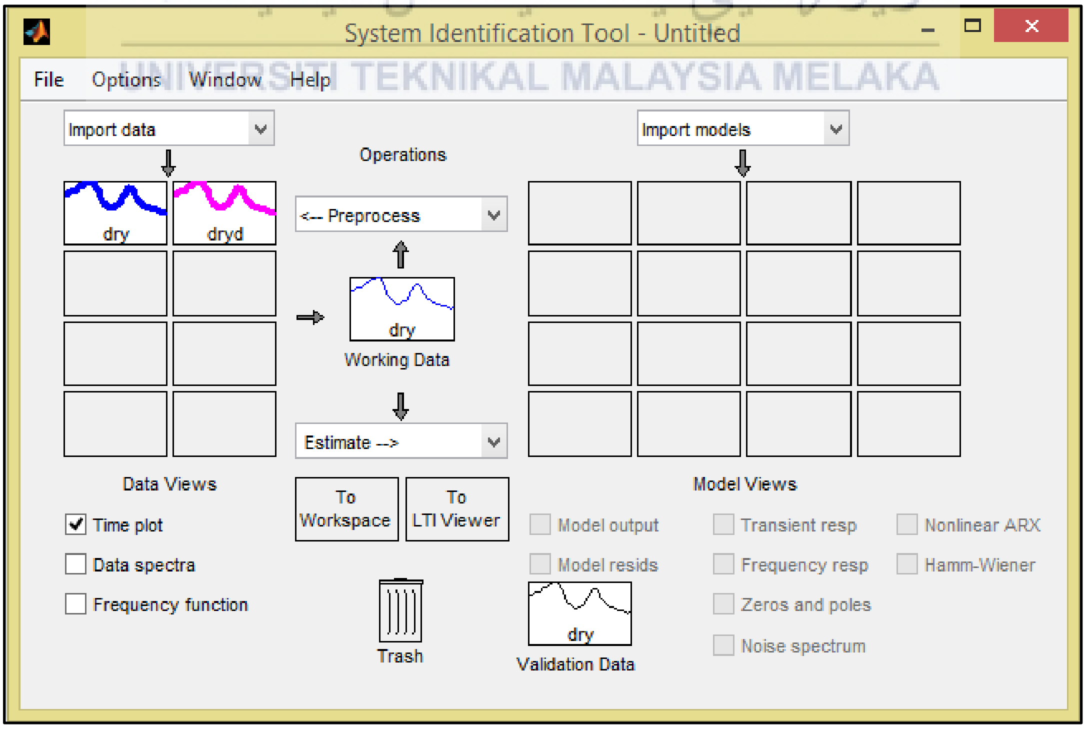


Figure 3.10: New data that has zero mean.

Drag and drop the data 'dryd' from data board onto Working Data icon like Figure 3.11. After that, click Select Range on Preprocess popup menu. Select range window will out and select the graph from 1 to 50 with curser or key in number in Time Span like Figure 3.12.

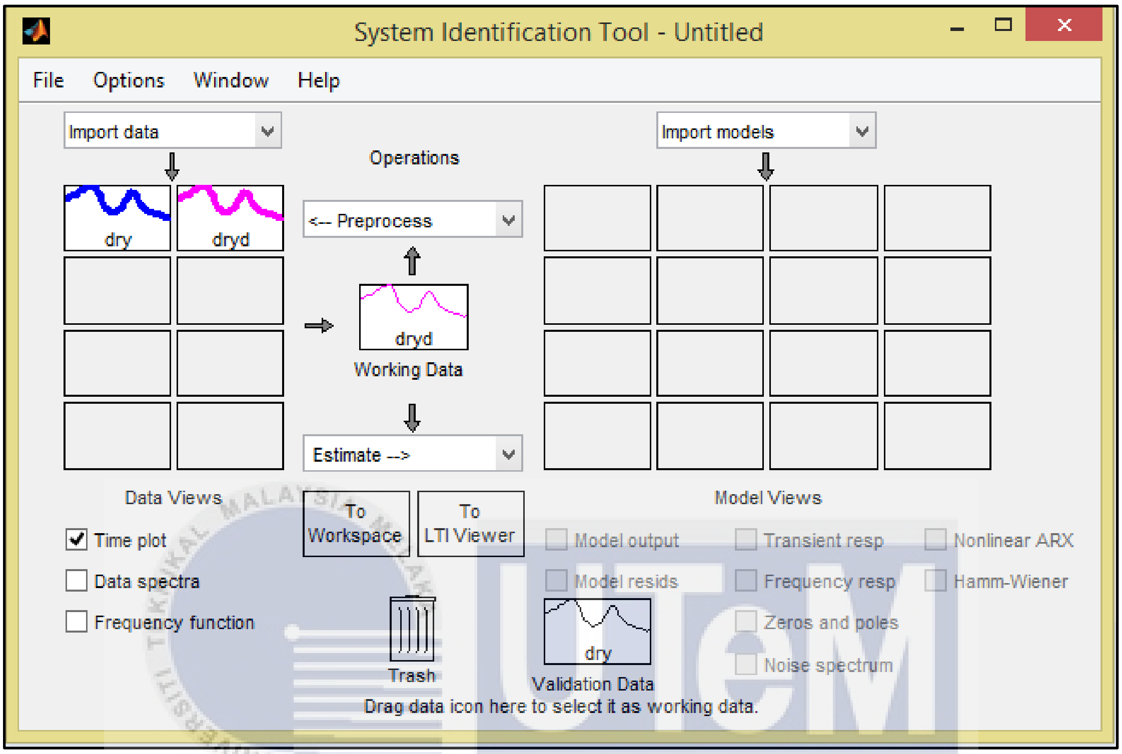


Figure 3.11: Move data 'dryd' to Working Data icon

Figure 3.12: Select Range from 1 to 50

After click insert in Select Range, new data named ‘dryde’ will appear like Figure 3.13.

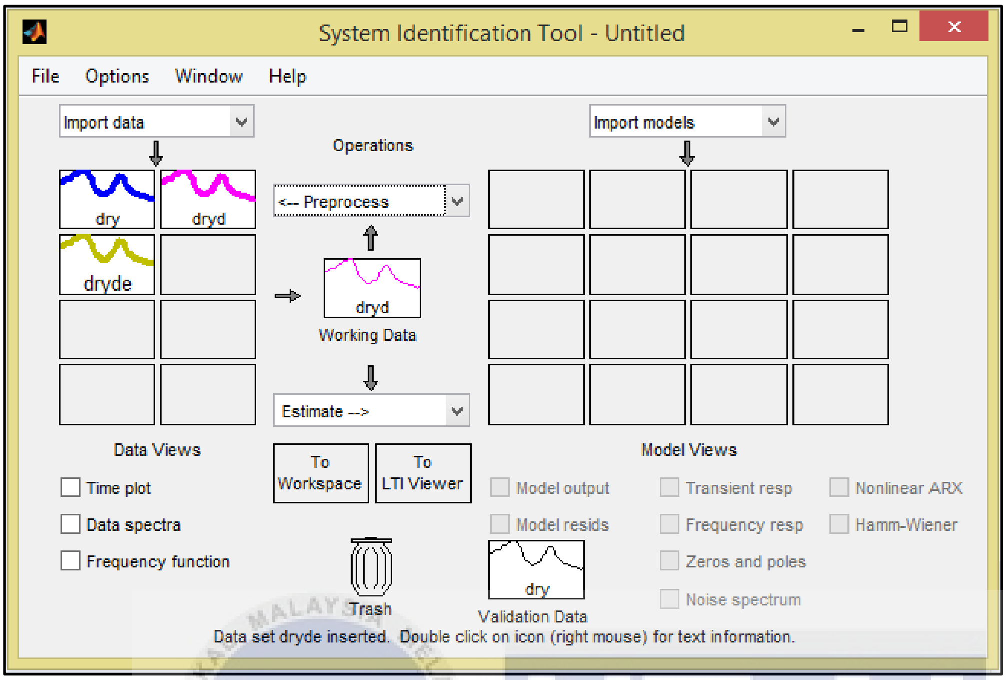


Figure 3.13: Data ‘dryde’ insert on data board

Same like before, but this time selects range from 50 to final like Figure 3.14 for validation data set. After click insert new data v will appear like Figure 3.15

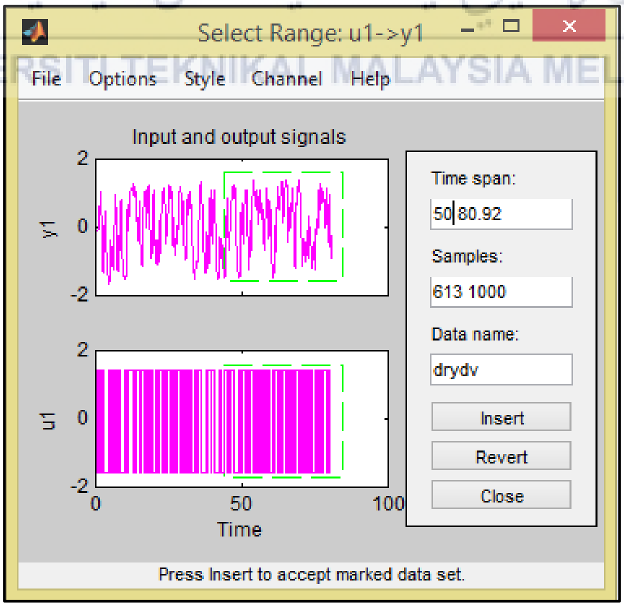


Figure 3.14: Select Range from 50 to final

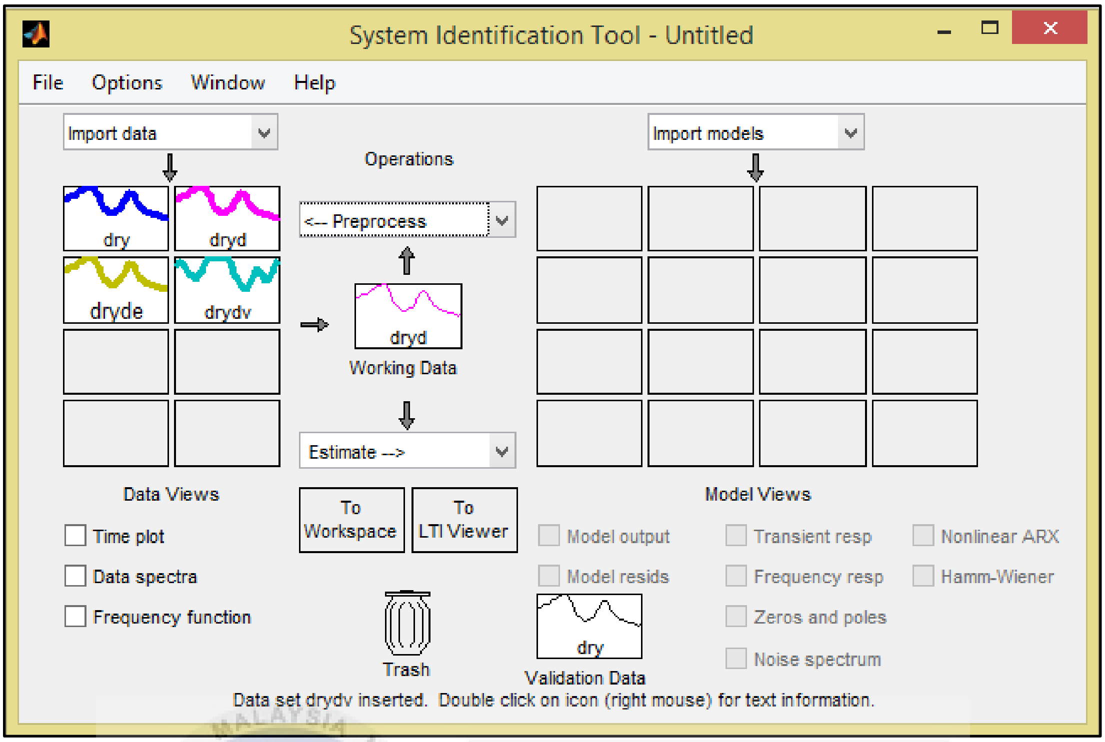


Figure 3.15: Data 'drydv' appear on data board

Drag data 'dryde' onto Working Data and data 'drydv' onto Validation Data like Figure 3.16

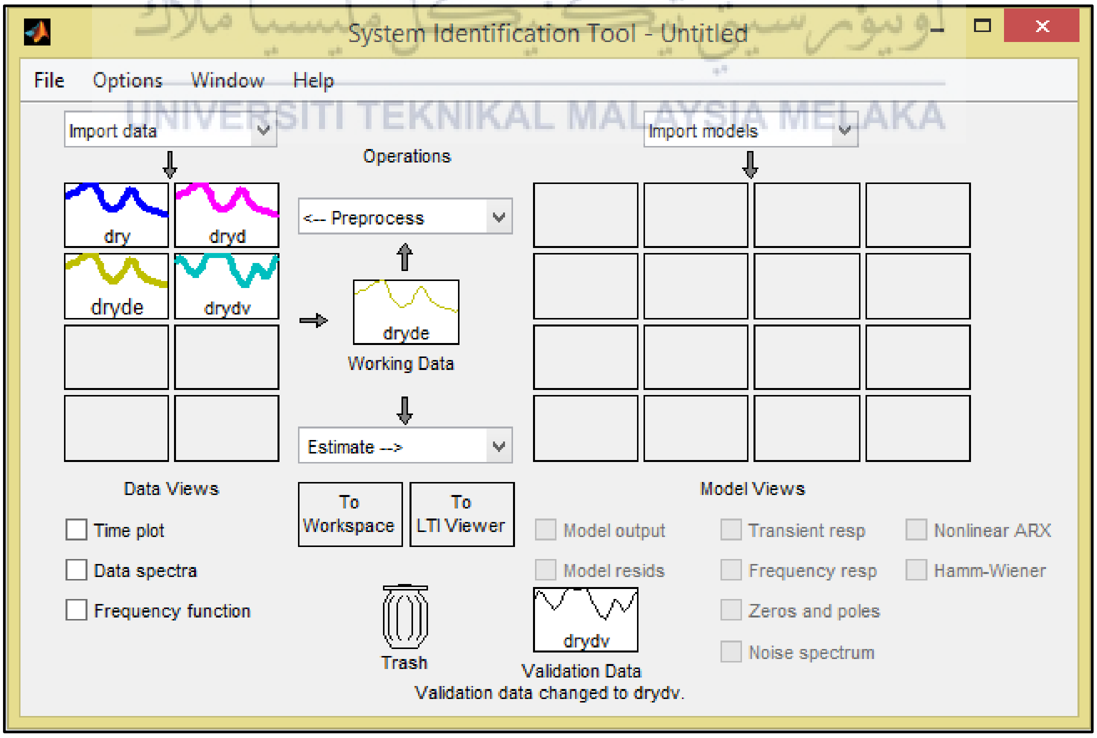


Figure 3.16: Drag data 'dryde' and data 'drydv'

Click Correlation Model on popup menu of Estimation like Figure 3.17 and Correlation Model window will pop out like Figure 3.18

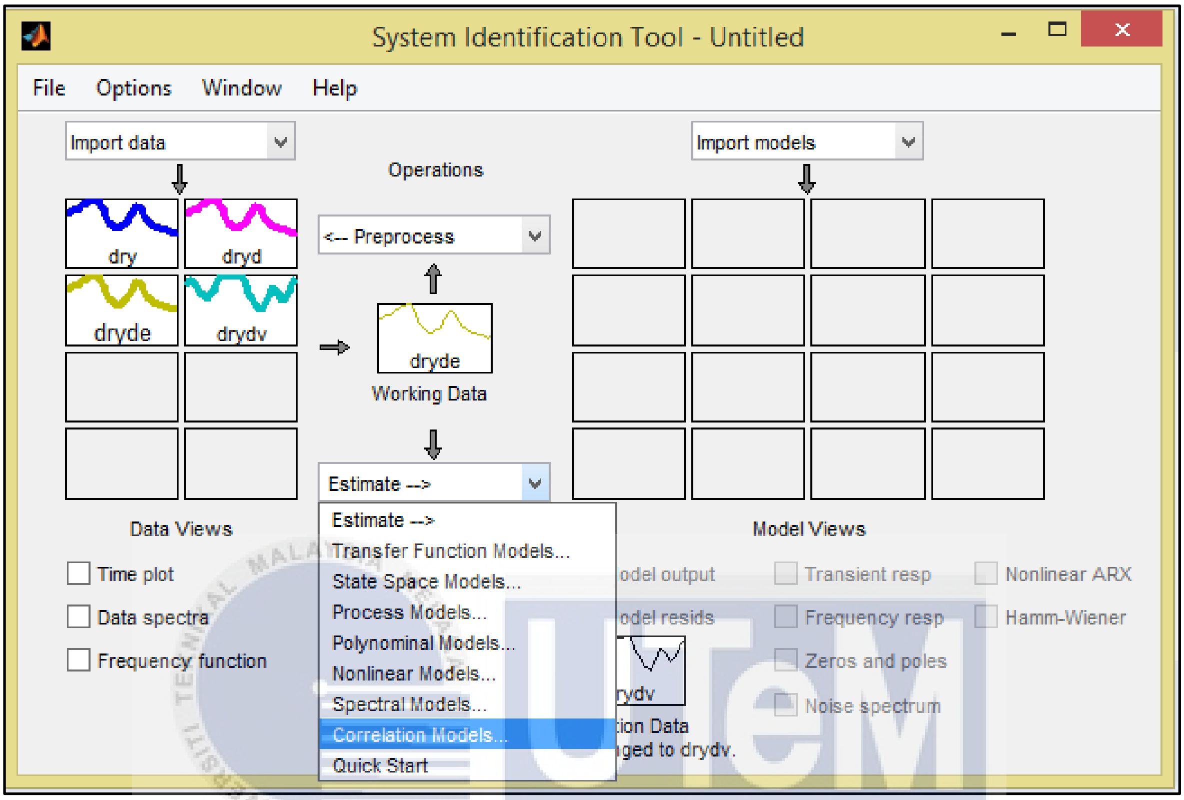


Figure 3.17: Click Correlation Model

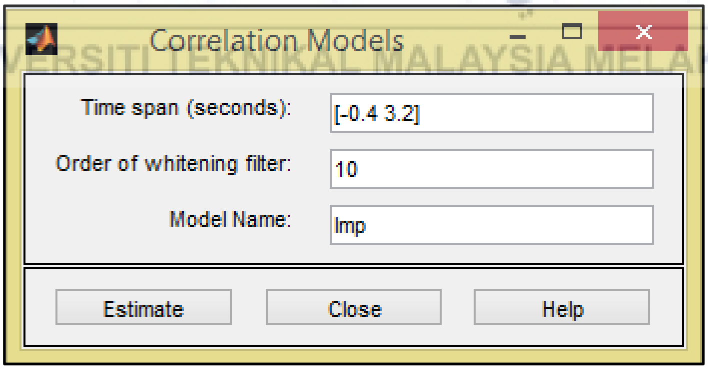


Figure 3.18: Correlation Model Window

Just click Estimate on Correlation Model Window and a correlation model will appear in model board like Figure 3.19.

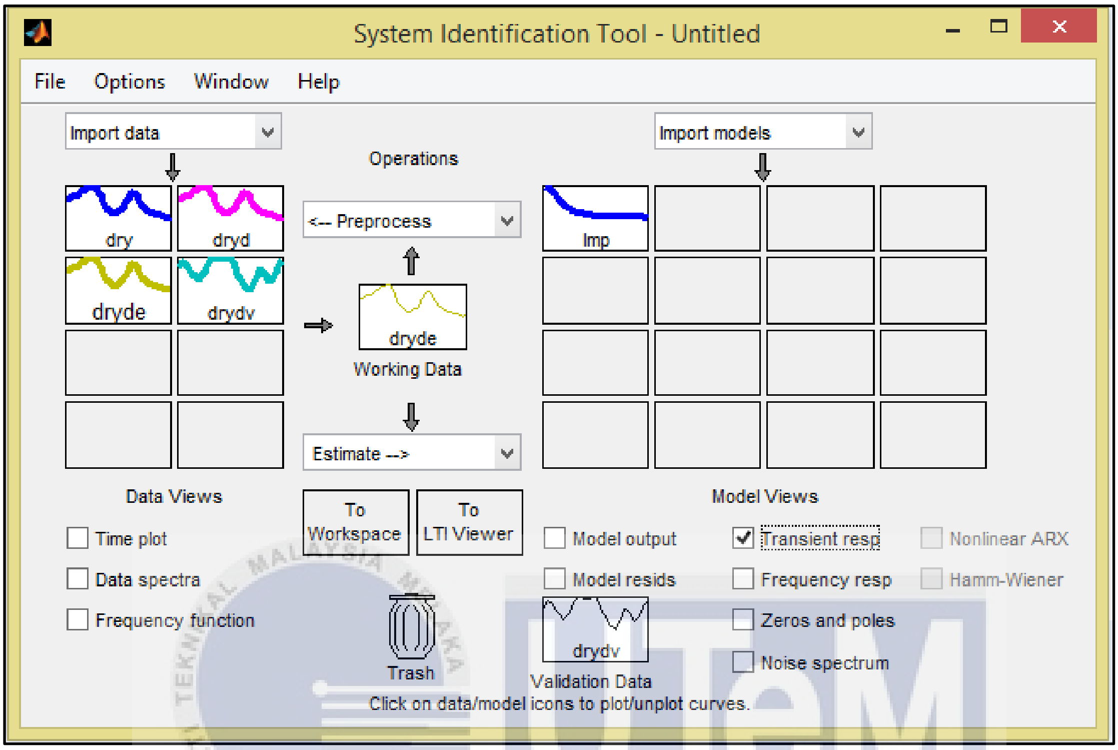


Figure 3.19: New Model appear on Model Board

To open Spectral Model window like Figure 3.21, just click Spectral Model on popup Estimate menu like Figure 3.20.

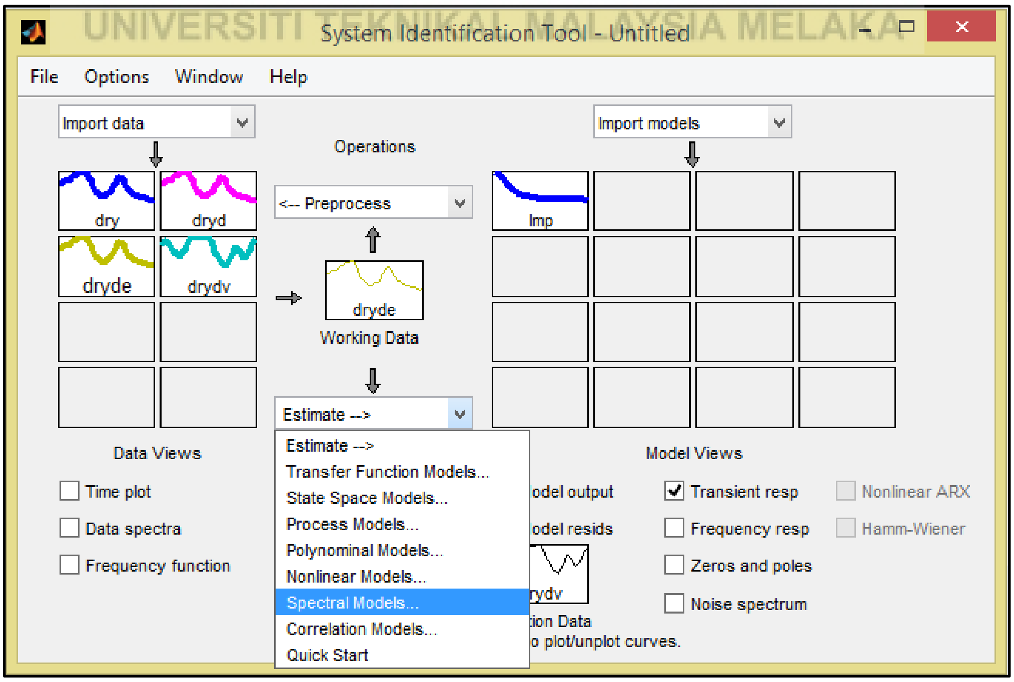


Figure 3.20: Click Spectral Model

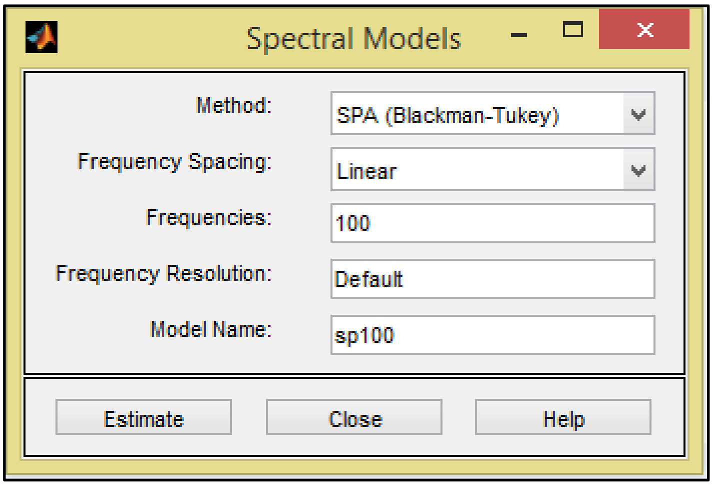


Figure 3.21: Spectral Model Window

After click Estimate on Spectral Model window, spectral model will appear on model board like Figure 3.22.

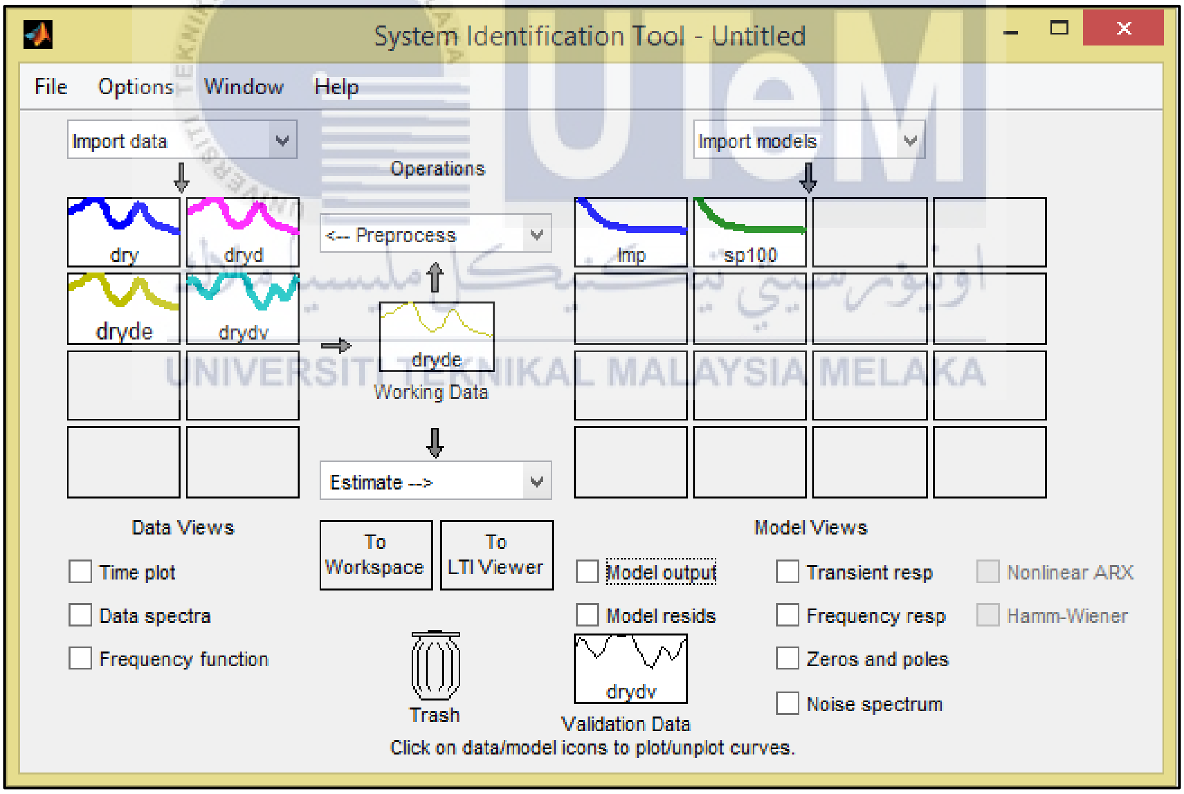


Figure 3.22: Spectral model appear on model board

Click Polynomial Model on popup Estimate menu like Figure 3.23 and Polynomial Model window will open like Figure 3.24.

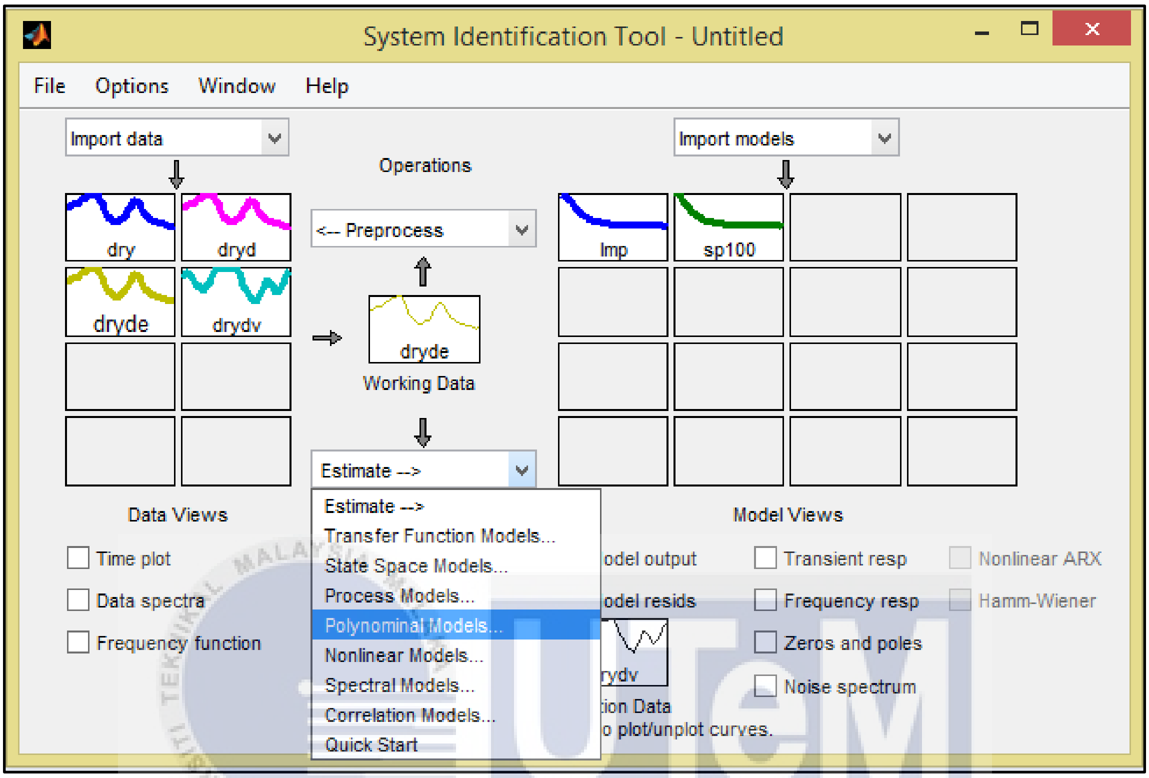


Figure 3.23: Click Polynomial Model

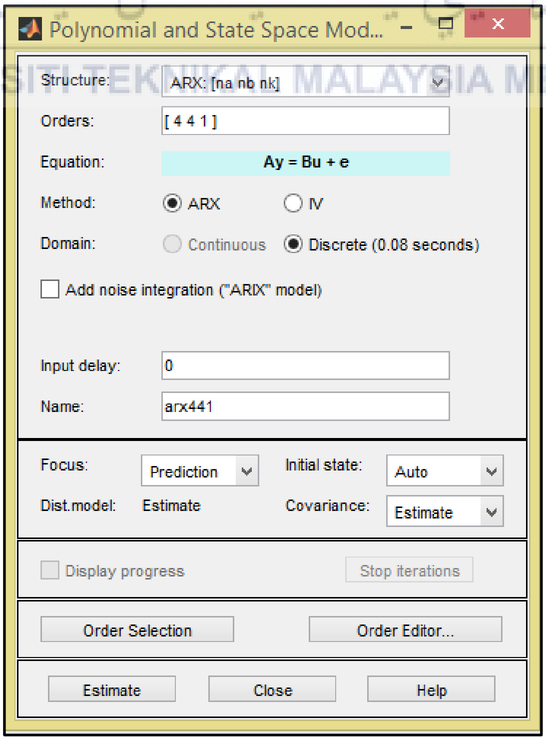


Figure 3.24: Polynomial Model Window

After click Estimate on Polynomial Model window, the new model will insert and appear on model board like Figure 3.25.

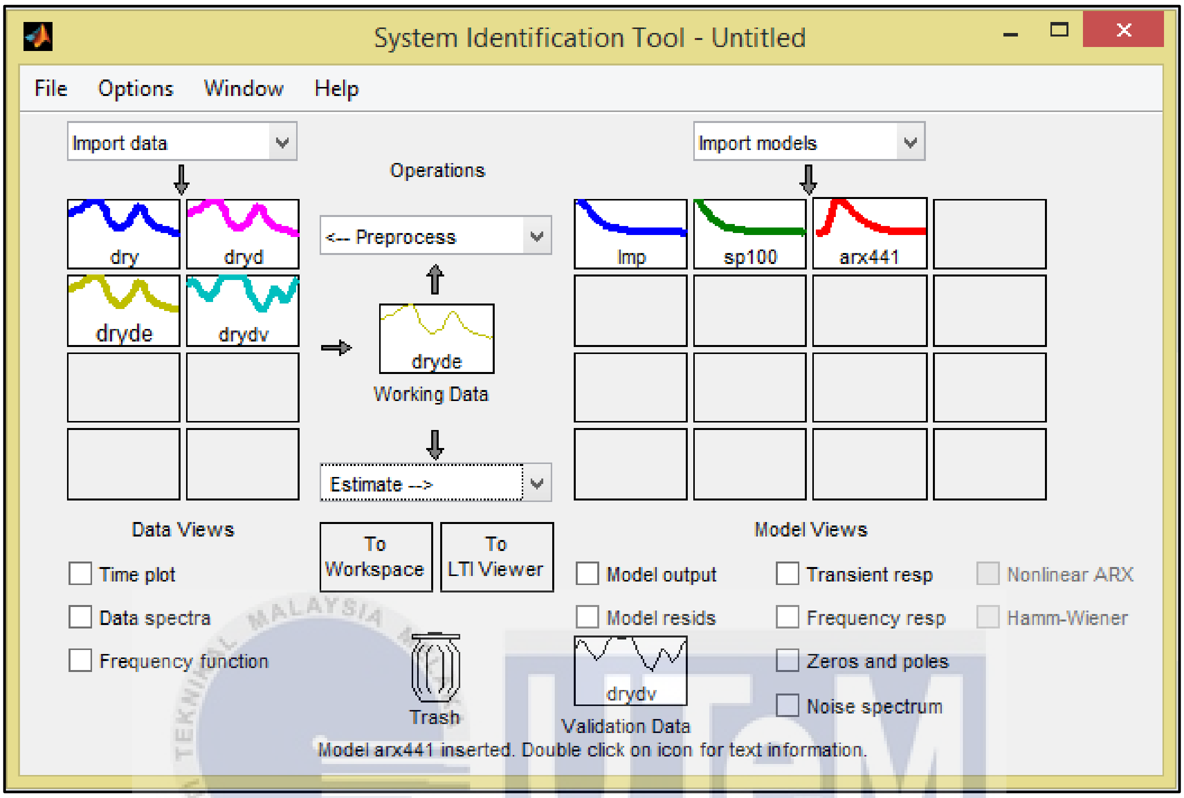


Figure 3.25: Polynomial Model With Order [4 4 1] Appear on Model Board

On Polynomial Model window, click Order Editor and select the order n_a=2, n_b=2 and n_k=3 like Figure 3.26.

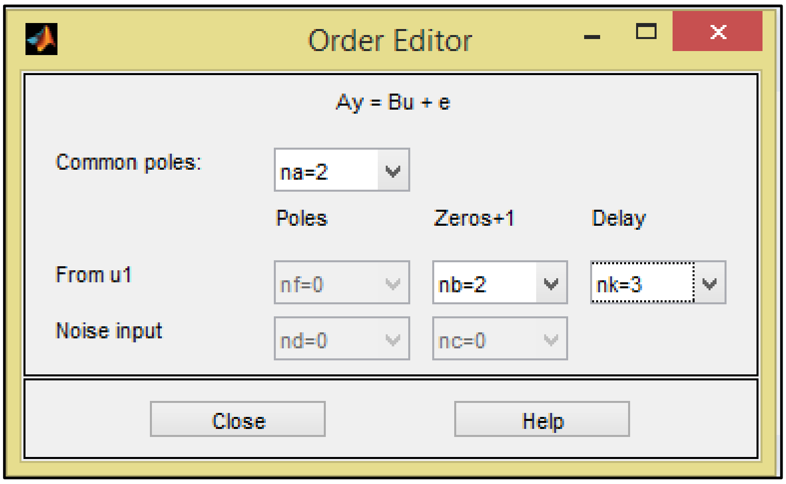


Figure 3.26: Order Editor Window

Click estimate on Polynomial Model window and model with new order will appear like Figure 3.27.

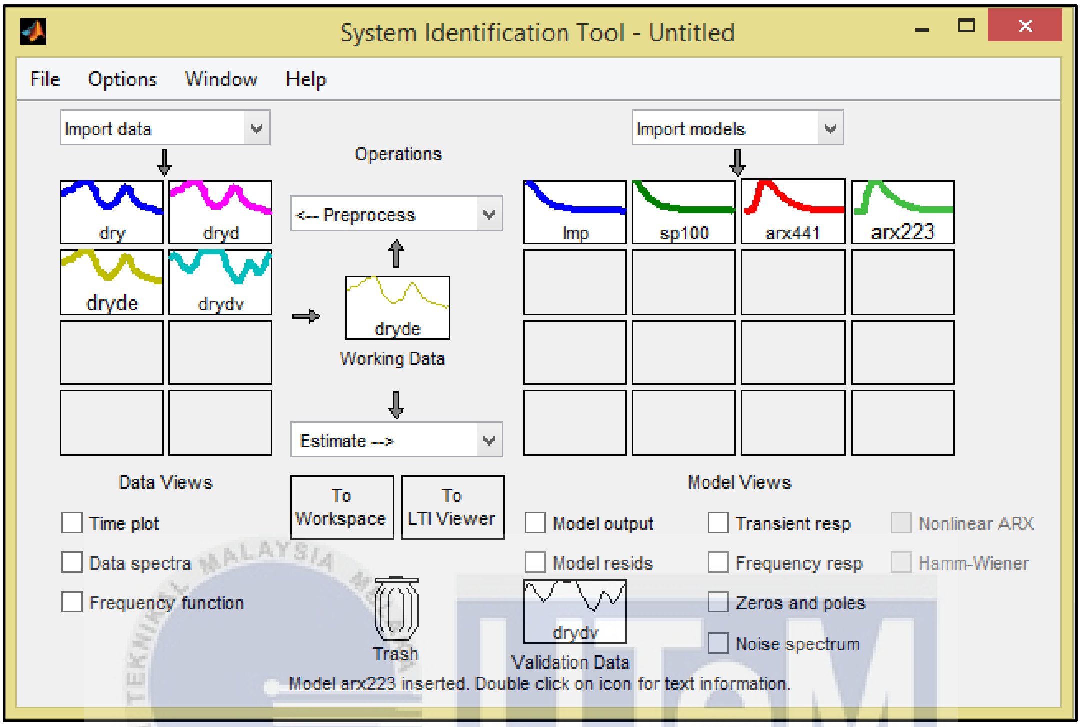


Figure 3.27: Polynomial Model With Order [2 2 3]

Click the Order Selection as shown in Figure 3.28, after that click estimate to know what the good order is.

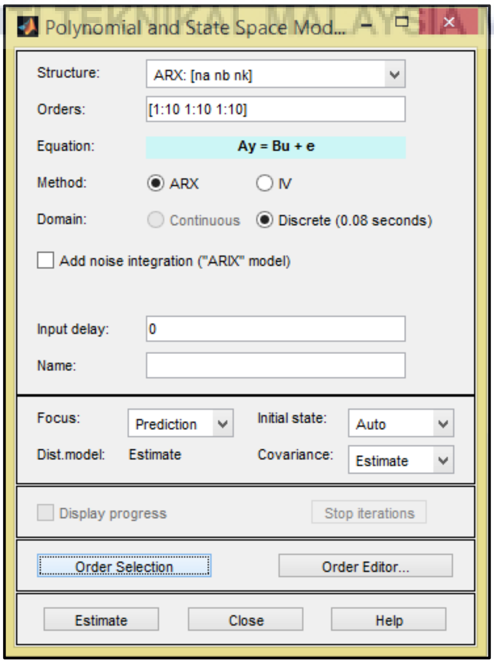


Figure 3.28: Click Order Selection

3.4 Result Based On Trial Run

All these result are getting from the trial run in 3.3 that been discussed before. There are 6 result that will be discussed which is frequency response, model output , residual analysis, noise spectrum, transient response and lastly zero and pole.

3.4.1 Frequency Response

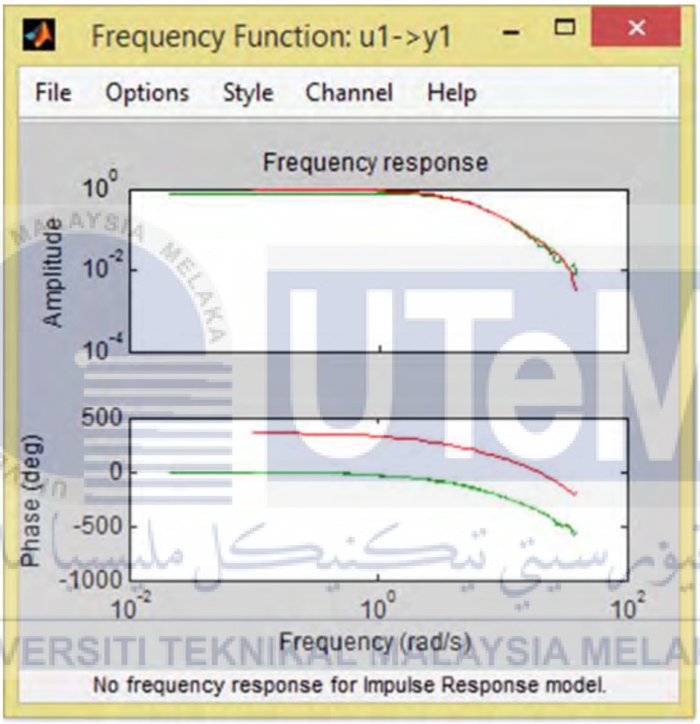


Figure 3.29: Frequency Response Graph

Figure 3.29 shown the example frequency responds graph. ’Bode(sys)’ creates a Bode plot of the frequency response of a dynamic system model ‘sys’. The plot displays the magnitude (in dB) and phase (in degrees) of the system response as a function of frequency. When ‘sys’ is a multi-input, multi-output (MIMO) model,’ bode’ produces an array of Bode plots, each plot showing the frequency response of one I/O pair. ‘Bode’ automatically determines the plot frequency range based on system dynamics.

'Bode' computes the frequency response using these steps:

1. Computes the zero-pole-gain (zpk) representation of the dynamic system.

2. Evaluates the gain and phase of the frequency response based on the zero, pole, and gain data for each input/output channel of the system.

    a. For continuous-time systems, bode evaluates the frequency response on the imaginary axis $s = j\omega$ and considers only positive frequencies.

    b. For discrete-time systems, bode evaluates the frequency response on the unit circle. To facilitate interpretation, the command parameterizes the upper half of the unit circle as equation 3.1.

$$z=e^{j\omega T_s} \qquad 0\leq\omega\leq\omega_N=\frac{\pi}{T_s}, \tag{3.1}$$

    where T_s is the sample time. ω_N is the *Nyquist frequency*. The equivalent continuous-time frequency ω is then used as the x-axis variable. Because $H(ej\omega T_s)$ is periodic and has a period $2\,\omega_N$, 'bode' plots the response only up to the Nyquist frequency ω_N. If you do not specify a sample time, bode uses $T_s = 1$.

3.4.2 Model Output

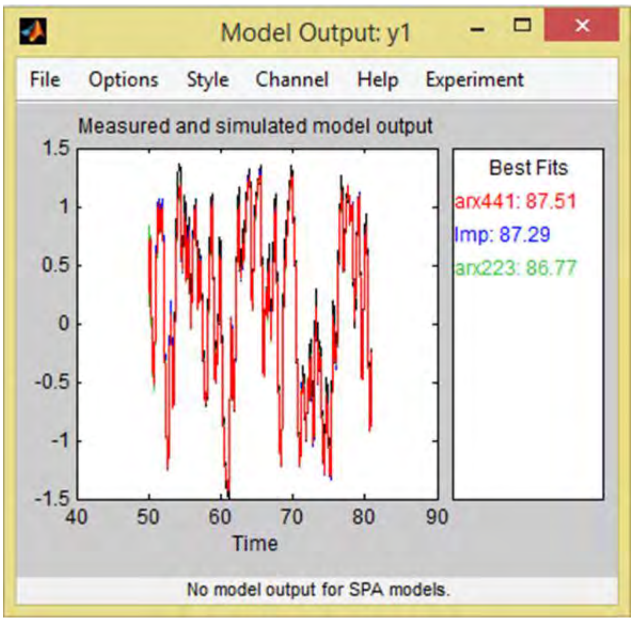


Figure 3.30: Model Output

Figure 3.30 model output plots for time domain validation data shows simulated or predicted model output. At the right side of the plot shows Best Fit that show which model is the best. A higher number means a better model. 100% corresponds to a perfect fit and the worst fit is close to 0%.

3.4.3 Residual Analysis

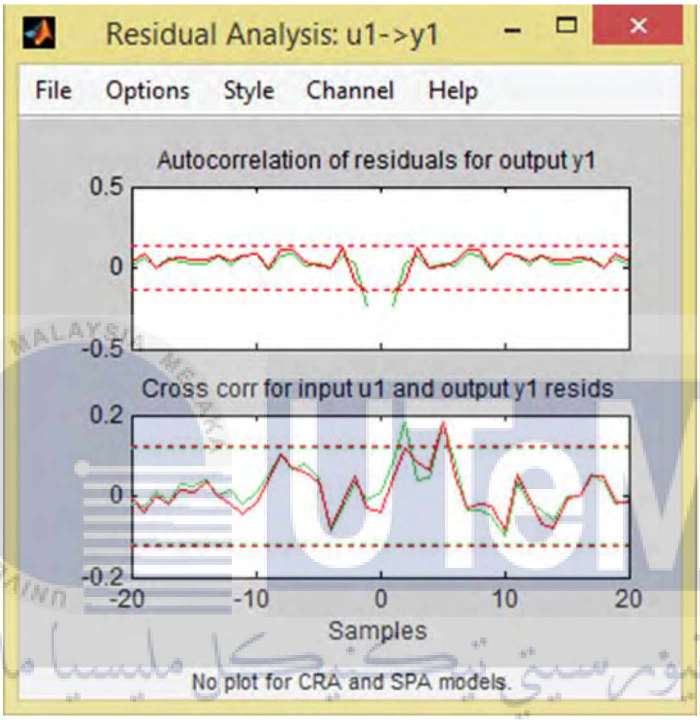


Figure 3.31: Residual Analysis

In Figure 3.31 Model Residual plot shows a residual analysis of the selected models. For time-domain validation data, the plot shows two graphs:

i. Top graph shows autocorrelation function of the residuals for the output.
ii. Bottom graph shows cross-correlation between the input and the residuals for each input-output pair
iii. The horizontal dashed lines on the plot represent the confidence interval of the corresponding estimates.

A good model should have residuals uncorrelated with past inputs (independence test). Evidence of correlation indicates that the model does not describe how the output is formed from the corresponding input.

3.4.4 Noise Spectrum

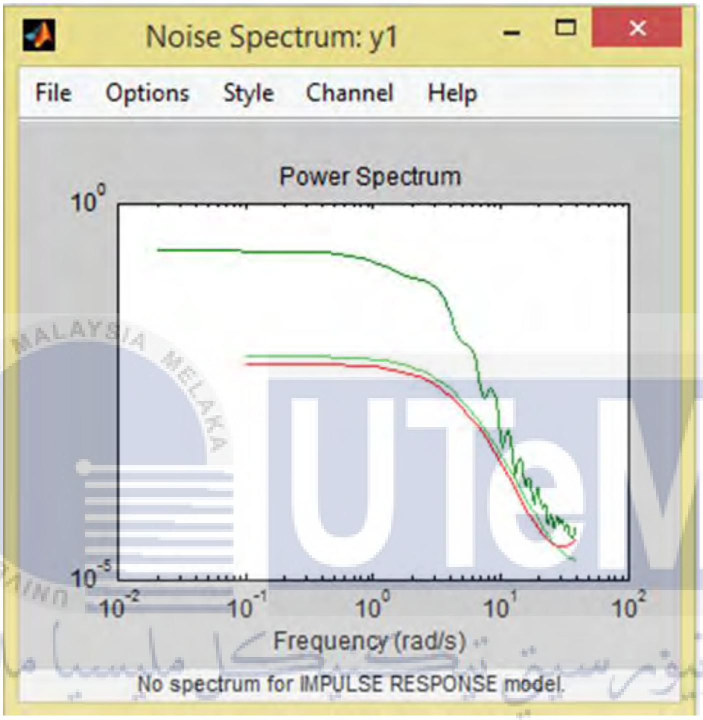


Figure 3.32: Noise Spectrum

Noise spectrum in Figure 3.32 curve displays a confidence interval on the plot. The confidence interval corresponds to the range of power-spectrum values with a specific probability of being the actual noise spectrum of the system. The toolbox uses the estimated uncertainty in the model parameters to calculate confidence intervals and assumes the estimates have a Gaussian distribution.

3.4.5 Transient Response

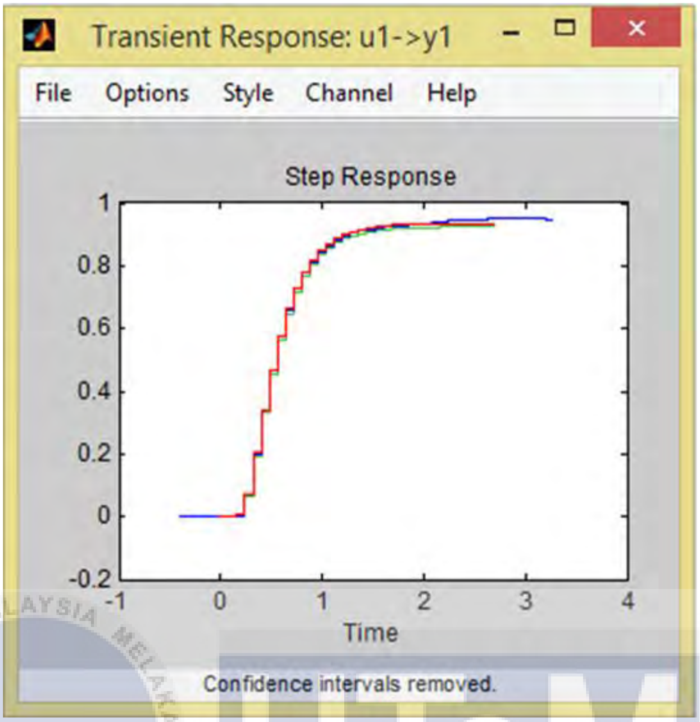


Figure 3.33: Transient Response

Figure 3.33 show Transient Response shape how the closed-loop system responds to a specific input signal when using Control System Tuner. In this case, a step input is assumed. Use a reference model to specify the desired transient response.

3.4.6 Zero And Pole

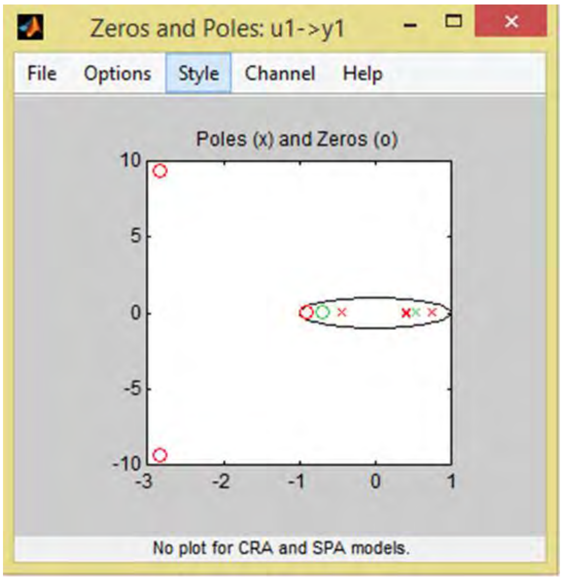


Figure 3.34: Zeros and Poles

The general equation of a linear dynamic system as shown in equation 3.2:

$$y(t)=G(z)u(t)+v(t) \tag{3.2}$$

In this equation, *G* is an operator that takes the input to the output and captures the system dynamics, and *v* is the additive noise term.

The poles of a linear system are the roots of the denominator of the transfer function *G*. The poles have a direct influence on the dynamic properties of the system. The zeros are the roots of the numerator of *G*. If you estimated a noise model *H* in addition to the dynamic model *G*, you can also view the poles and zeros of the noise model.

Zeros and the poles as shown in Figure 3.34 are equivalent ways of describing the coefficients of a linear difference equation, such as the ARX model. Poles are associated with the output side of the difference equation, and zeros are associated with the input side of the equation. The number of poles is equal to the number of sampling intervals between the most-delayed and least-delayed output. The number of zeros is equal to the number of sampling intervals between the most-delayed and least-delayed input.

3.5 Data Acquisition

For data program in this project there are three different equations. The three equations are as follows:

i) $y(t) = 0.2y(t-1)-0.6y(t-3)+0.5u(t-2)+u(t-6)+e(t)$ (3.3)

ii) $y(t) = -0.3y(t-3)+0.4y(t-4)+u(t-3)+0.3u(t-4)+0.7e(t)-0.9e(t-3)$ (3.4)

iii) $y(t) = 0.1y(t-2)-0.3y(t-5)-0.4u(t-1)+0.3e(t-5)$ (3.5)

These equation represent ARX and ARMAX model equation, first equation is is ARX model and another two is ARMAX . After some calculation, will find value na=3 nb=5 and nk=2 for first equation. For second equation, will find na=4 nb=2 nk=3 and nc=3. For last equation, the value of order na,nb,nc and nk are 5,2,5 and 3 respectively. After that, open the file to modified the program with change the equation in line number 13, make sure the folder path that want to be save also be edit like Figure 3.35.

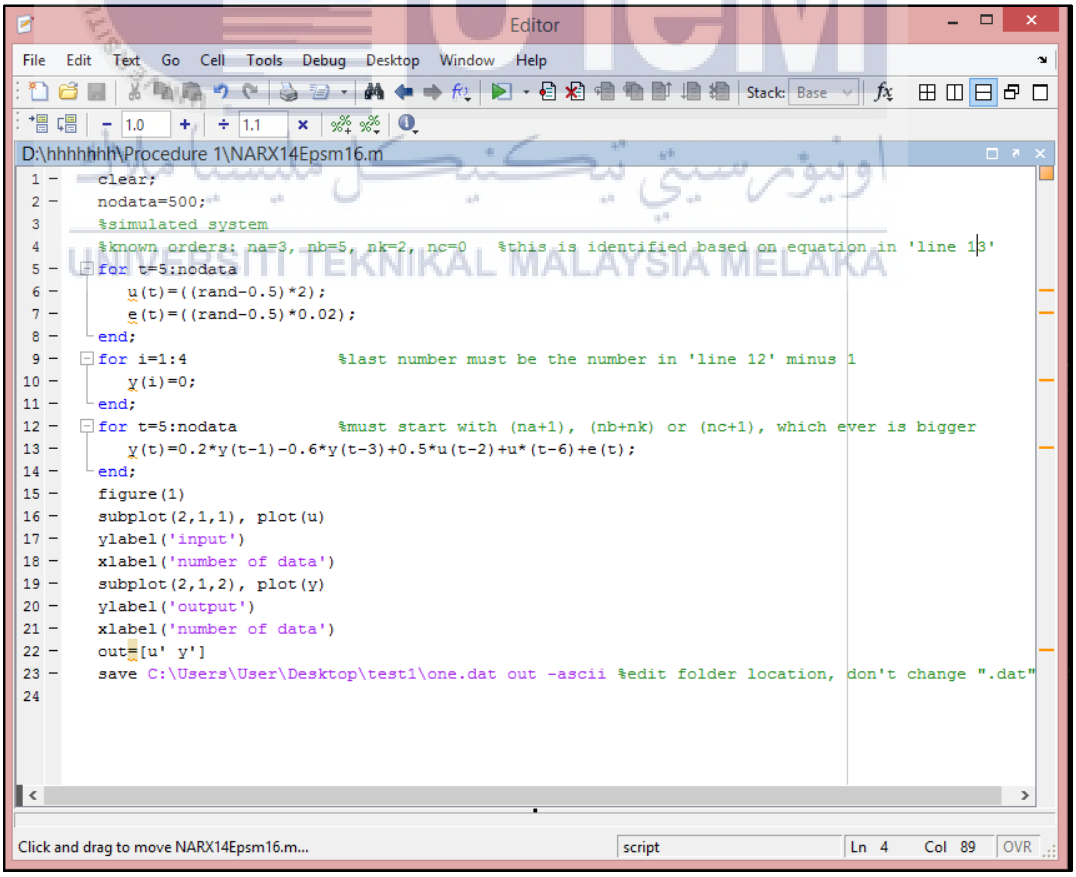


Figure 3.35: Editor Window

After the program was run , the data was generate as shown in Figure 3.36 and ready to be perform in system identification

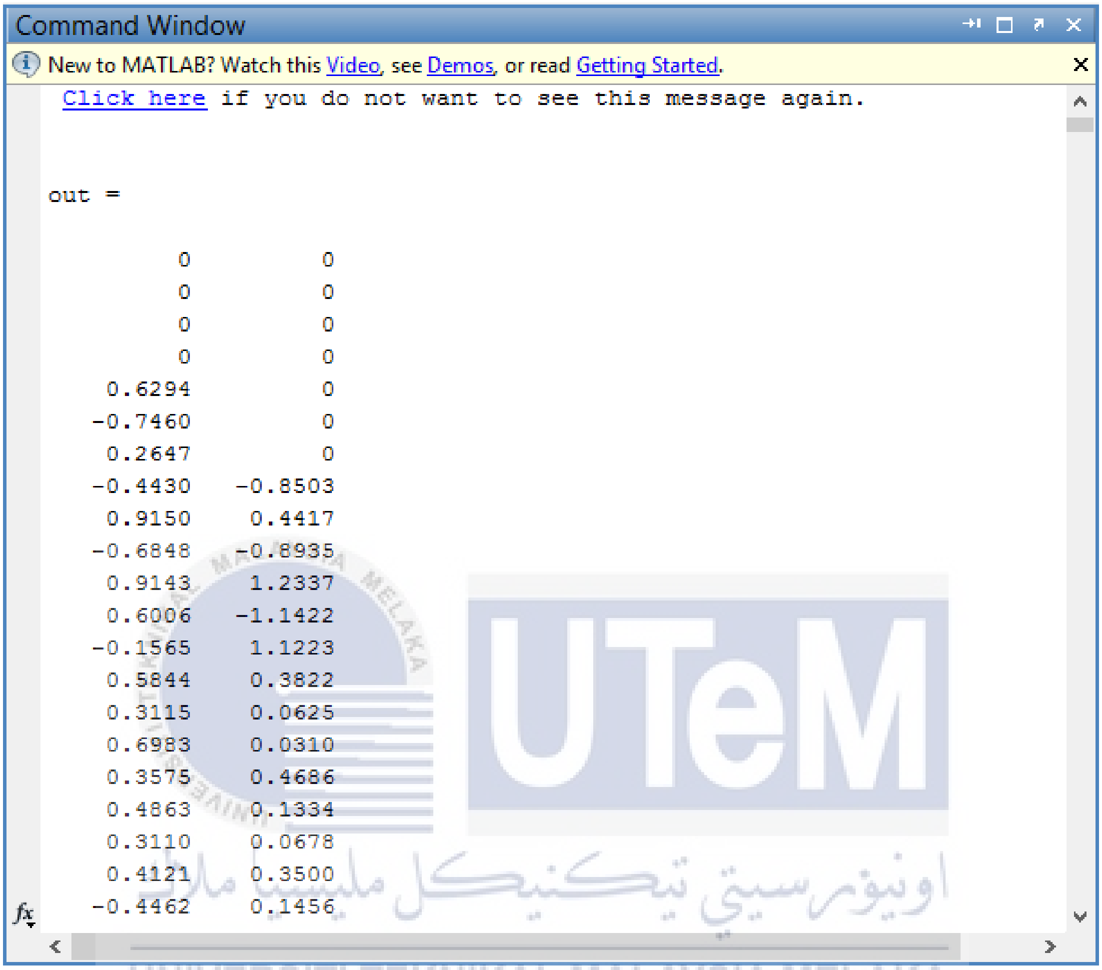


Figure 3.36: Command Window

3.6 Performance Indicator

Before this, we had discussed about the variation of model in system identification. So, to choose the best model between the bunches of model, we will use the performance indicator. The performance indicator that we use for this project is loss function, model output, model residuals akaike's final prediction Error and parameter value. Next we will discuss about this entire performance indicator one by one for the more detail.

3.6.1 Loss function

The system identification toolbox software estimates model parameters by minimizing the error between the model output and the measured response. This error, called loss function or cost function, is a positive function of prediction errors $e(t)$. In general, this function is a weighted sum of squares of the errors. For a model with n_y-outputs, the loss function $V(\theta)$ has the following general form:

$$V(\theta) = \frac{1}{N} \sum_{T=1}^{N} e^T(t,\theta)\, W(\theta)\, e(t,\theta) \tag{3.3}$$

where:

- N is the number of data samples.
- $e(t,\theta)$ is n_y-by-1 error vector at a given time t, parameterized by the parameter vector θ.
- $W(\theta)$ is the weighting matrix, specified as a positive semi definite matrix. If W is a diagonal matrix, you can think of it as a way to control the relative importance of outputs during multi-output estimations. When W is a fixed or known weight, it does not depend on θ.

The software determines the parameter values by minimizing $V(\theta)$ with respect to θ. For notational convenience, $V(\theta)$ is expressed in its matrix form:

$$V(\theta) = \frac{1}{N} trace\,(E^T(\theta)\, E(\theta)\, W(\theta)) \tag{3.4}$$

$E(\theta)$ is the error matrix of size N-by-n_y. The i:th row of $E(\theta)$ represents the error value at time $t = i$.

The exact form of $V(\theta)$ depends on the following factors:

- Model structure. For example, whether the model that you want to estimate is an ARX or a state-space model.
- Estimator and estimation options. For example, whether you are using *n4sid* or *ssest* estimator and specifying options such as ‘Focus’ and ‘OutputWeight'.

3.6.2 Model output

The Model output window is opened by checking the corresponding check box in the 'ident' window. The plots show the simulated (predicted) outputs of selected models. The models are fed with inputs from the Validation Data set. The plot takes somewhat different forms depending on the character of the validation data. This could be:

- Time domain data

  The simulated or predicted model output is shown together with the measured validation data.

- Frequency domain data

  The amplitude of the (complex-valued) model output is shown together with the measured output's amplitude.

- Frequency Function data

  The amplitude curves of the data's and the models' frequency responses are shown.

In all the cases, the percentage of the output variations that is reproduced by the model is displayed at the side of the plot. A higher number means a better model. The precise definition of the fit shown in equation (3.5)

$$Fit\ (R^2) = (1 - \sqrt{\frac{\Sigma(y-\hat{y})^2}{\Sigma(y-\bar{y})^2}}\ \text{x}\ 100\% \quad (3.5)$$

Where is the measured output and it is the stimulated/predicted model output. The time span over which the fit is measured can be changed under the Options sub-menu Customized time span for fit. There are sub-menus under the Options menu, which allows one to choose between simulated and predicted model output. There are also options to show measured and model outputs together or to show the difference between them.

3.6.3 Akaike's Final Prediction Error

From the prediction error standpoint, the higher the order of the model is, the better the model fits the data because the model has more degrees of freedom. However, you need more computation time and memory for higher orders. The parsimony principle says to choose the model with the smallest degree of freedom, or number of parameters, if all the models fit the data well and pass the verification test. The criteria to assess the model order therefore not only must rely on the prediction error but also must incorporate a penalty when the order increases. Akaike's Information Criterion (AIC), Final Prediction Error Criterion (FPE), and the Minimum Description Length Criterion (MDL) are criteria one can use to estimate the model order.

Akaike's Final Prediction Error (FPE) criterion provides a measure of model quality by simulating the situation where the model is tested on a different data set. After computing several different models, you can compare them using this criterion. According to Akaike's theory, the most accurate model has the smallest FPE.

If you use the same data set for both model estimation and validation, the fit always improves as you increase the model order and, therefore, the flexibility of the model structure. Akaike's Final Prediction Error (FPE) is defined by the following equation:

$$FPE = \det\left(\frac{1}{N}\sum_{1}^{N} e\left(t,\hat{\theta}_N\right)\left(e\left(t,\hat{\theta}_N\right)\right)^T\right)\left(\frac{1+\frac{d}{N}}{1-\frac{d}{N}}\right) \tag{3.6}$$

where:

- N is the number of values in the estimation data set.
- $e(t)$ is a $n_y$by-1 vector of prediction errors.
- $\hat{\theta}_N$ represents the estimated parameters.
- d is the number of estimated parameters.

If number of parameters exceeds the number of samples, FPE is not computed when model estimation is performed (model.Report.FPE is empty). The fpe command returns NaN

3.6.4 Parameter Value

In the context of block diagram when representing system, blocks have numeric parameters that determine how they calculate output values. To control the calculations that blocks perform, one can specify parameter values. For example, a 'Gain' block has a Gain parameter, and a 'Transfer Fcn' block has multiple parameters that represent the transfer function coefficients.

One can use numbers, variables, and expressions to set block parameter values. Choose a technique based on your modelling goals. For example, you can:

- Share parameter values between blocks and models by creating variables.
- Control parameter characteristics such as data type and dimensions by creating parameter objects.
- Model an algorithm as code by using mathematical expressions.

CHAPTER 4

RESULT AND ANALYSIS

4.1. Introduction

In this chapter, the results that were obtained from the simulation will be discussed in detail one by one for the three equations that were tested before. The result that want be discuss and analyse is based on MSE (Mean Squared Error, Model Output, FPE (Final Prediction Error), Residual Analysis and Parameter Values. For the last part in this chapter, overall discussion about all this result will be state.

4.2. Mean Square Error

MSE is a network performance function. It measures the network's performance according to the mean of squared errors. It can be set to any value between 0 and 1. The greater the regularization value, the more squared weights and biases are included in the performance calculation relative to errors. The default is 0, corresponding to no regularization. It ensures that the relative accuracy of output elements with differing target value ranges are treated as equally important, instead of prioritizing the relative accuracy of the output element with the largest target value range.

4.2.1. Equation 1

For the MSE data that show in Table 4.1, the left column shows the model order and for the right column shows the MSE data. From the table one graph is creating as shows in Figure 4.1. The highest MSE is a model name AMX3501 which is having 0.0001301 and the lowest MSE is model name AMX3591 that have 3.012e-5 means that model AMX3591 is the best model.

Model	MSE
Amx3501	0.0001301
Amx3511	0.00013
Amx3521	0.00013
Amx3531	0.0001296
Amx3541	0.0001299
Amx3551	0.0001298
Amx3561	3.096e-05
Amx3571	3.067e-05
Amx3581	3.289e-05
Amx3591	3.012e-05

Table 4.1: MSE for Equation 1

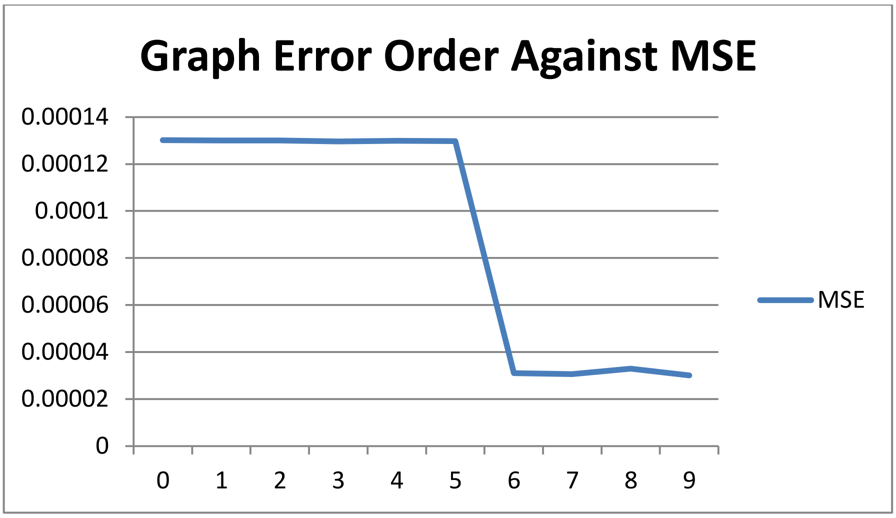


Figure 4.1: Graph Error Order against MSE for Equation 1

4.2.2. Equation 2

For the MSE data that show in Table 4.2, the left column shows the model order and for the right column shows the MSE data. From this table, one graph is created to make comparison between all models as shows in Figure 4.2. The highest MSE is a model name AMX4213 which is having 0.000909 and the lowest MSE is model name AMX4273 that have 6.044e-05. This is a proof that the best model is AMX4273.

Model	MSE
amx4203	0.0009076
amx4213	0.000909
amx4223	0.0004986
amx4233	0.0004986
amx4243	0.0004907
amx4253	0.0004835
amx4263	0.0003557
amx4273	6.044e-05
amx4283	6.184e-05
amx4293	5.887e-05

Table 4.2: MSE for Equation 2

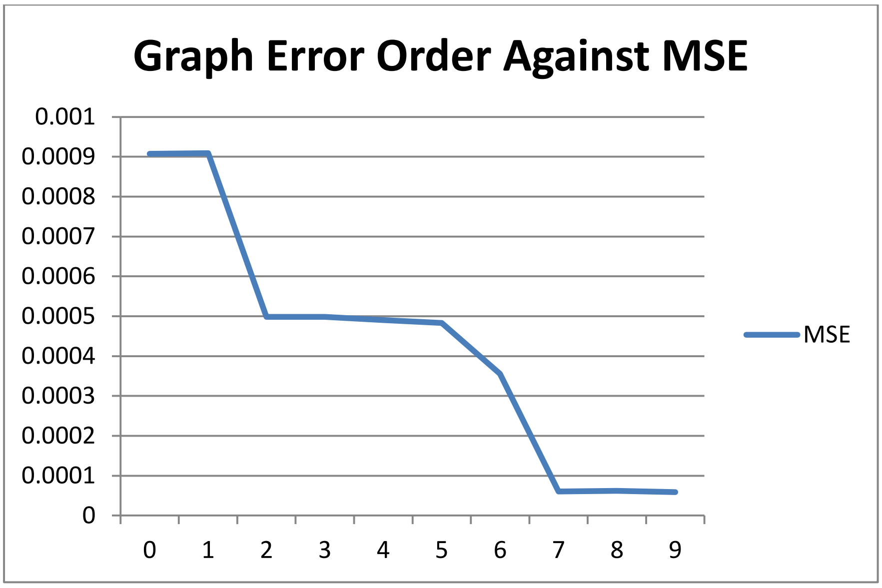


Figure 4.2: Graph Error Order against MSE for Equation 2

4.2.3. Equation 3

For the MSE data that show in Table 4.3, the left column shows the model order and for the right column shows the MSE data. From the table, graph is created to better vision during the comparison as shows in Figure 4.3. The highest MSE is a model name AMX5201 which is having 6.222e-05 and the lowest MSE is model name AMX5291 that have 5.813e-06. According this, we can say that the model AMX5291 is the best model.

Model	MSE
amx5201	6.222e-05
amx5211	4.237e-05
amx5221	4.193e-05
amx5231	4.193e-05
amx5241	4.187e-05
amx5251	4.186e-05
amx5261	4.359e-05
amx5271	5.979e-06
amx5281	5.819e-06
amx5291	5.813e-06

Table 4.3: MSE for Equation 3

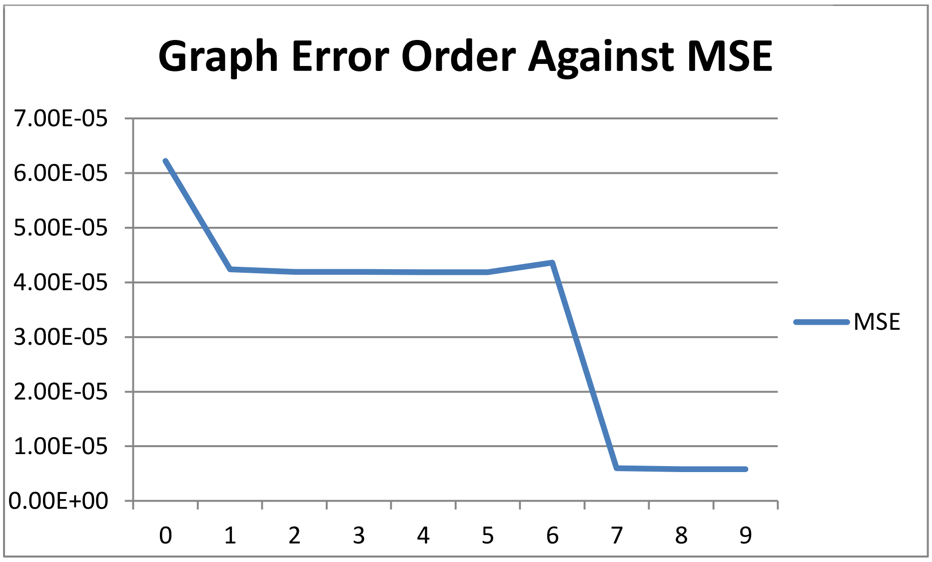


Figure 4.2: Graph Order against MSE for Equation 3

4.3. Model Output

4.3.1. Equation 1

Figure 4.3 show the model output dialog for the first equation. From this figure, the best fit for this run is model AMX3562 follow by AMX3582, AMX3572, AMX35923, AMX5023, AMX5123, AMX3532, AMX3522, AMX3552 and AMX3542. As we can see on the figure, the model AMX3562, AMX3582, AMX3572 and AMX3592 share the same value of the best fit and we can say that this all 4 model is the best model.

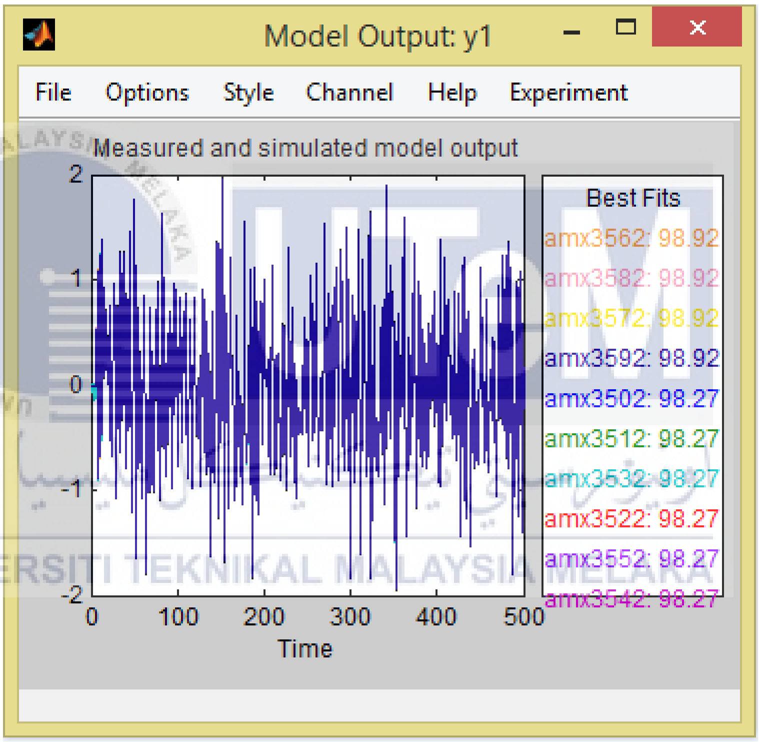


Figure 4.3: Model Output Percentage for Equation 1

4.3.2. Equation 2

Figure 4.4 show the model output dialog for the second equation. From this figure, the best fit for this run is model AMX4273 follow by AMX4283, AMX4293, AMX4263, AMX4253, AMX4213, AMX4203, AMX4233, AMX4223 and AMX4243. The model AMX4273 that hold 98.78 is the only best model since it the only model that have a best fit result.

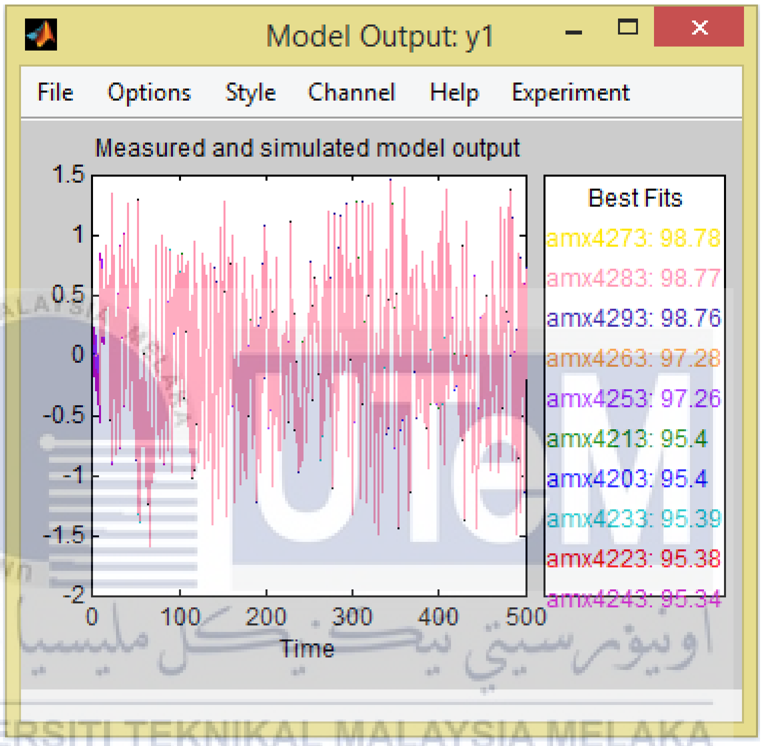


Figure 4.4: Model Output Percentage for Equation 2

4.3.3. Equation 3

Figure 4.4 show the model output dialog for the third equation. From this figure, the best fit for this run is model AMX5281 follow by AMX5291, AMX5271, AMX5261, AMX5201, AMX5221, AMX5241, AMX5231, AMX5251 and AMX5211. This result quite similar as the equation 1 result since the 3 model have the same 99.68 best fit which is model AMX5281, AMX5291 and AMX5271. So we can state that the best model for the equation 3 it is all of this 3 models.

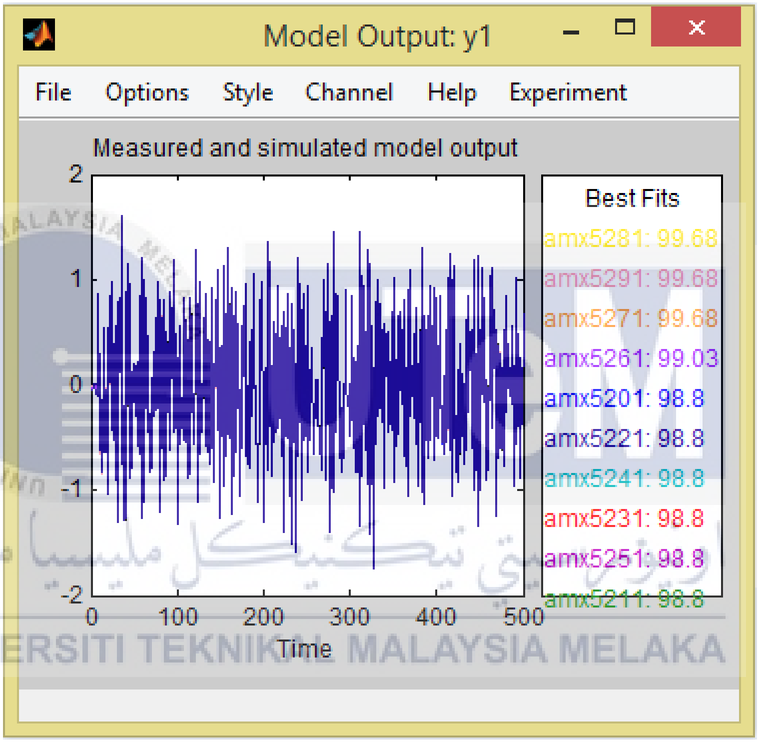


Figure 4.5: Model Output Percentage for Equation 3

4.4. Model Residuals

The top axes show the autocorrelation of residuals for the output (whiteness test). The horizontal scale is the number of lags, which is the time difference (in samples) between the signals at which the correlation is estimated. The horizontal dashed lines on the plot represent the confidence interval of the corresponding estimates. Any fluctuations within the confidence interval are considered to be insignificant.

A good model should have a residual autocorrelation function within the confidence interval, indicating that the residuals are uncorrelated. The bottom axes show the cross-correlation of the residuals with the input. A good model should have residuals uncorrelated with past inputs (independence test). Evidence of correlation indicates that the model does not describe how the output is formed from the corresponding input.

For the equation 1, as we can see on figure 4.6 proof that all model show line between the interval confident and we can say that all model is accepted models.

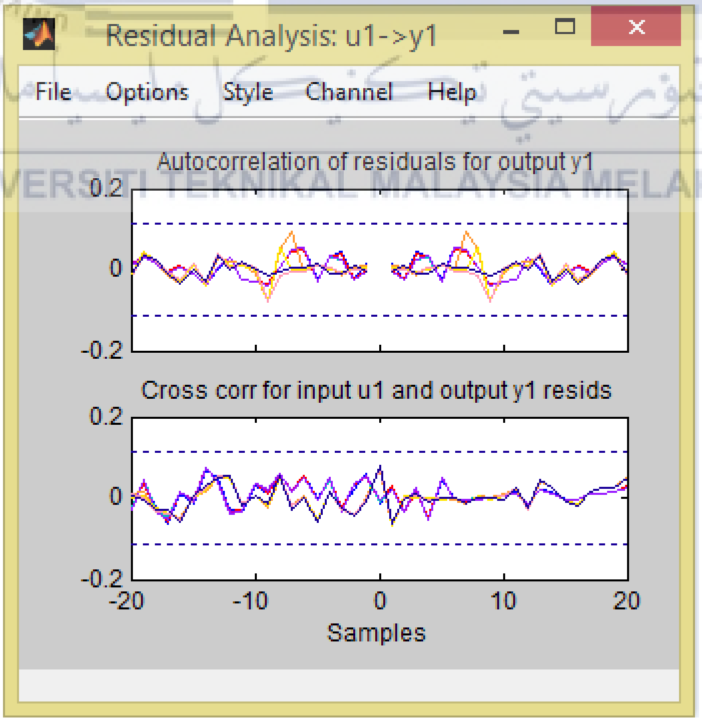


Figure 4.6: Best Model on Residual Analysis for Equation 1

For equation 2, there is only one model that fulfil the criteria of the best model which is model AMX4263 that show on Figure 4.7. And the other model which is model AMX4203, AMX4213, AMX4223, AMX4233, AMX4243, AMX4253, AMX4273 ,AMX4283 and AMX4293 it is not good because the graph of all model is not between the confidence interval as shown on Figure 4.8.

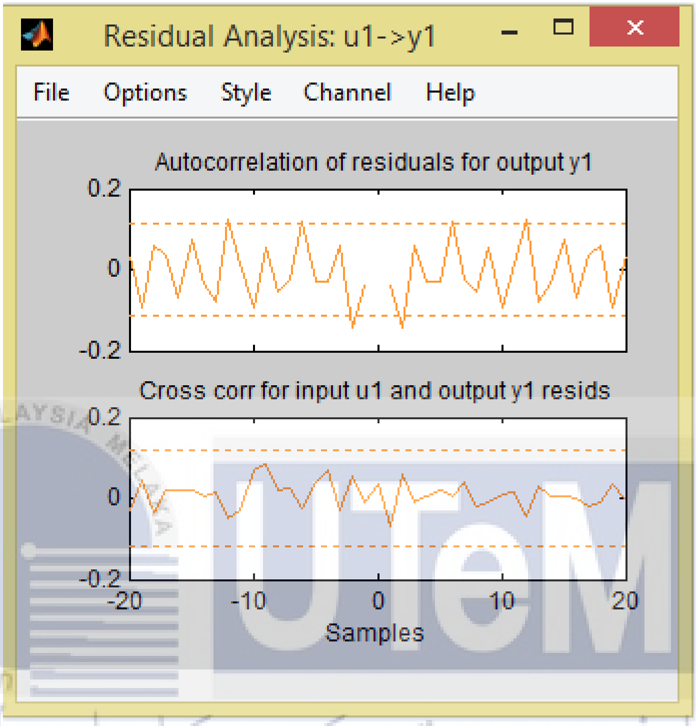


Figure 4.7: Best Model on Residual Analysis for Equation 2

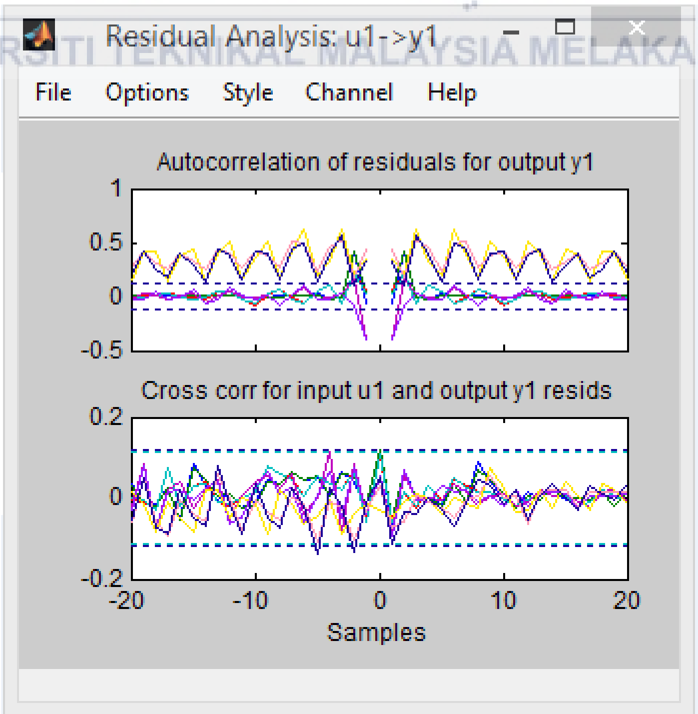


Figure 4.8: Rejected Model on Residual Analysis for Equation 2

For the equation 3, the good model is the AMX5211, AMX5221, AMX5231, AMX5241, AMX5251 and AMX5261 as shown on Figure 4.8. Besides that, for another model which is model AMX5201, AMX5271, AMX5281 and AMX5291 shows on Figure 4.9 proof it is not good model because not fit the criteria of accepted model.

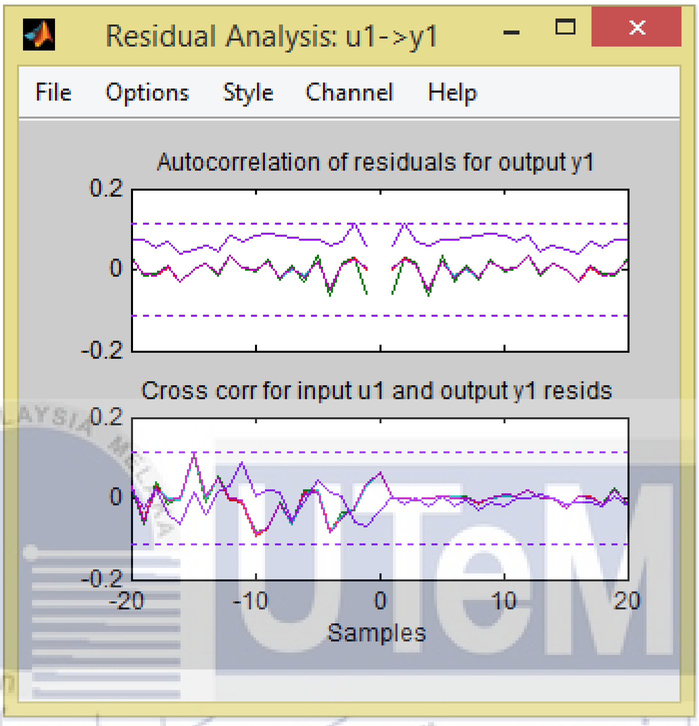


Figure 4.8: Best Model on Residual Analysis for Equation 3

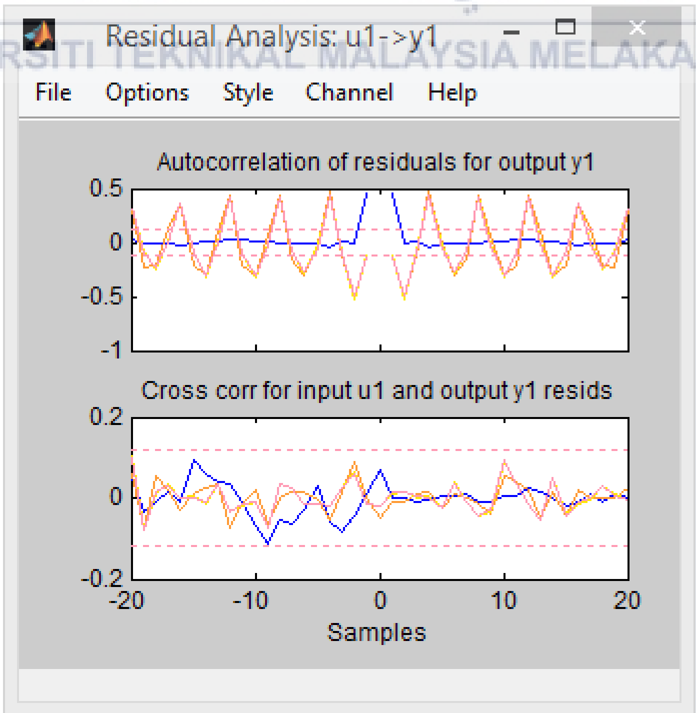


Figure 4.9: Rejected Model on Residual Analysis for Equation 3

4.5. Akaike's Final Prediction Error

All FPE data were taken from every data model by click the model name and the model data will pop up. All data that take was fill in one table and convert to one graph for every equation to make the comparison and analyse easier. To recognize the best model in this section, the models that have smallest FPE value is the best model.

4.5.1. Equation 1

Table 4.4 shows the value of final prediction error for the first equation. The first column is the name of the models that have been test. The second column is the value of the final prediction error for each model. From the Figure 4.10, the line graph show the best model is the model AMX3591 because it has the smallest value.

Model	Final Prediction Error
AMX3501	3.304e-05
AMX3511	3.297e-05
AMX3521	3.31e-05
AMX3531	3.309e-05
AMX3541	3.319e-05
AMX3551	3.33e-05
AMX3561	3.31e-05
AMX3571	3.292e-05
AMX3581	3.289e-05
AMX3591	3.284e-05

Table 4.4: Final Prediction Error for Equation 1

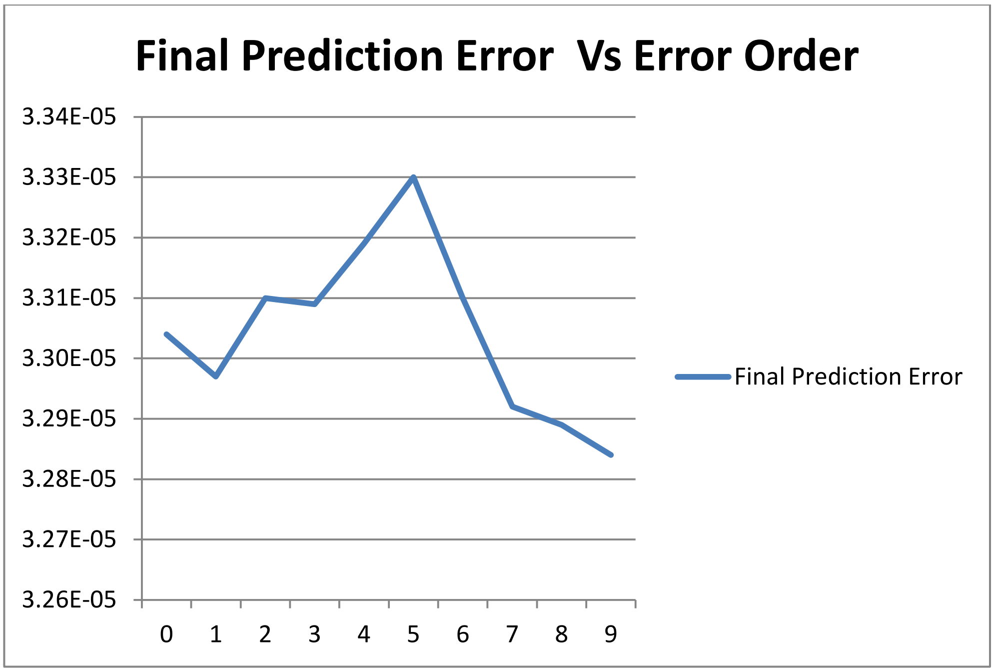


Figure 4.10: Graph of FPE against Error Order for Equation 1

4.5.2. Equation 2

Table 4.5 shows the value of final prediction error for the first equation. As we can see at the Figure 4.11, the graph shows that model AMX4273 is the best model because it has small value.

Model	Final Prediction Error
amx4203	0.000937
amx4213	0.0009339
amx4223	0.0004194
amx4233	0.0004199
amx4243	0.0001437
amx4253	0.0001248
amx4263	0.0001136
amx4273	6.5889e-05
amx4283	8.127e-05
amx4293	6.653e-05

Table 4.5: Final Prediction Error for Equation 2

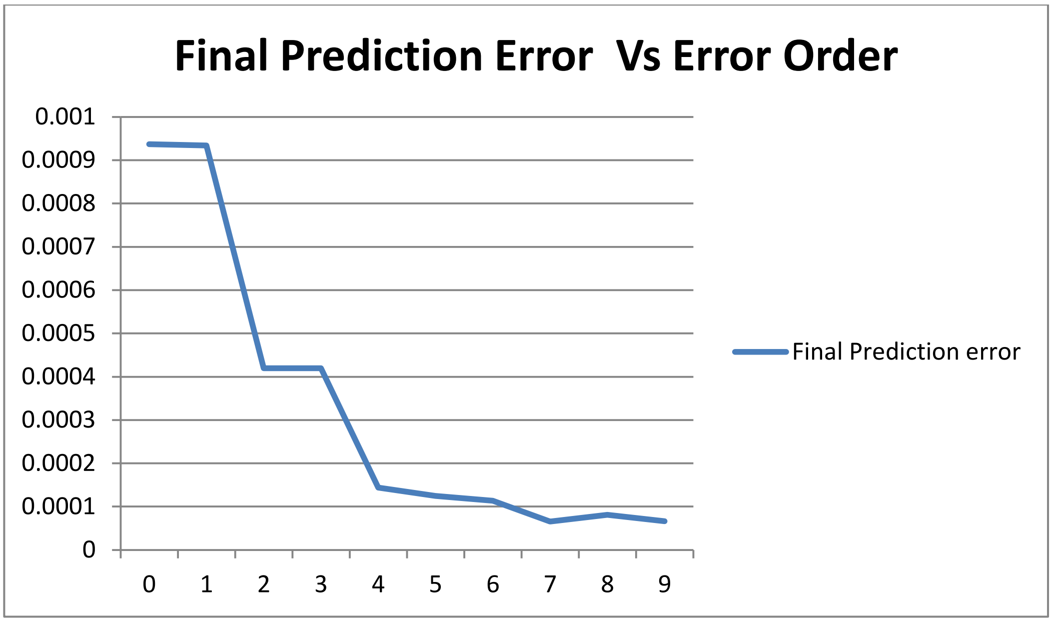


Figure 4.11: Graph of FPE against Error Order for Equation 2

4.5.3. Equation 3

Table 4.6 shows the value of final prediction error for the first equation. The first column is the name of the models that have been test. The second column is the value of the final prediction error for each model. Based on Figure 14.2 proof that the model AMX5201 has the smallest value it is the best model.

Model	Final Prediction Error
AMX5201	6.544e-0.5
AMX5211	4.484e-05
AMX5221	4.482e-05
AMX5231	4.491e-05
AMX5241	4.512e-05
AMX5251	4.538e-05
AMX5261	1.157e-05
AMX5271	6.905e-06
AMX5281	6.808e-06
AMX5291	6.792e-06

Table 4.6: Final Prediction Error for Equation 2

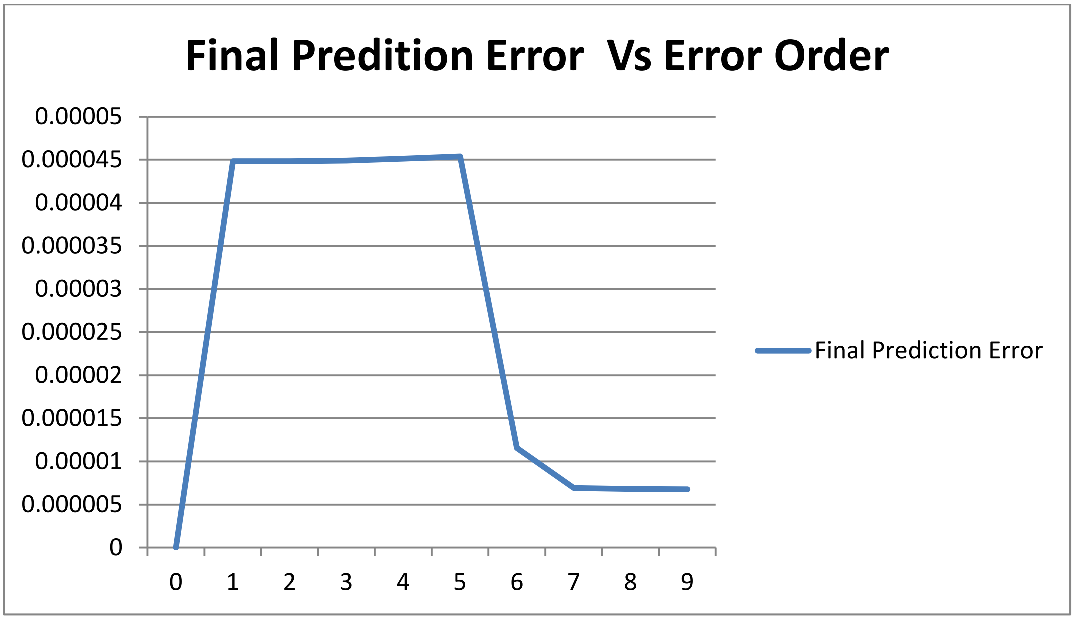


Figure 4.12: Graph of FPE against Error Order for Equation 3

4.6. Parameter Values

For the parameter value, table were made to make the comparison with the true value that take from the equation and the result of each value of the all model is obtaines from the data of model.

4.6.1. Equation 1

From the Table 4.7, we can see that the parameter value for all model and true value. To proof which model is the best, the parameter must be same or close to the true value. All the same or close value with the true value of the model, the value is bold to make the comparison. For equation 1, the best is model AMX3502 because the y(t-1) and y(t-3) is close to the value, also for the u(t-2) and u(t-1) has the same value.

Table 4.7: Parameter Value for Equation 1

Model	y(t-1)	y(t-3)	u(t-2)	u(t-6)
True Value	**0.2**	**0.6**	**0.5**	**1**
AMX3502	**0.2002**	**0.5994**	**0.5**	**1**
AMX3512	**0.2002**	**0.5994**	0.5001	**1**
AMX3522	**0.2002**	**0.5994**	0.5001	**1**
AMX3532	0.2003	**0.5994**	0.5001	**1**
AMX3542	0.2003	**0.5994**	0.5001	**1**
AMX3552	0.2003	**0.5994**	0.5001	**1**
AMX3562	0.2003	0.5993	**0.5**	0.9995
AMX3572	**0.2002**	0.5993	0.5001	0.9994
AMX3582	**0.2002**	0.5993	0.5001	0.9995
AMX3592	**0.2002**	0.5993	0.5001	0.9994

4.6.2. Equation 2

For the Equation 2, from the Table 4.8 we can say that the AMX4243 is the best model because it has 3 value that same and close to the true value which is y(t-4), u(t-3) and u(t-4). Besides that, the close value for the data y(t-3) is model AMX4263 and AMX4273 that has same value.

Table 4.8: Parameter Value for Equation 2

Model	y(t-3)	y(t-4)	u(t-3)	u(t-4)
True Value	**0.3**	**0.4**	**1**	**0.3**
AMX4203	0.3006	0.3981	0.995	0.314
AMX4213	0.3006	0.3991	0.9953	0.3118
AMX4223	0.3009	0.3991	0.998	0.3052
AMX4233	0.3011	0.399	0.9979	0.3058
AMX4243	0.3012	**0.4**	0.9988	**0.3003**
AMX4253	0.2997	0.3983	0.9988	0.2989
AMX4263	**0.2998**	0.3989	0.9986	0.2976
AMX4273	**0.2998**	0.4011	0.9985	0.2987
AMX4283	0.2991	0.4004	0.9994	0.298
AMX4293	0.2996	0.4003	**1**	0.2985

4.6.3. Equation 3

The best model is AMX5271, AMX5281 and AMX5291 that share the same all 4 parameter value. As we can see based on Table 4.9, the value y(t-2) and y(t-5) are pretty close with the true value while the value for u(t-1) and y(t-2) is same to the true value.

Table 4.9: Parameter Value for Equation 3

Model	y(t-2)	y(t-5)	u(t-1)	y(t-2)
True Value	**0.1**	**0.3**	**0.4**	**1**
AMX5201	0.1003	**0.3799**	0.399	**1**
AMX5211	0.1003	**0.3799**	0.3996	**1**
AMX5221	0.1003	**0.3799**	0.3995	**1**
AMX5231	0.1003	**0.3799**	0.3995	**1**
AMX5241	0.1003	**0.3799**	0.3995	**1**
AMX5251	0.1003	0.3798	0.3995	1
AMX5261	0.1003	0.3797	0.4	1
AMX5271	**0.1002**	**0.3799**	**0.4**	**1**
AMX5281	**0.1002**	**0.3799**	**0.4**	**1**
AMX5291	**0.1002**	**0.3799**	**0.4**	**1**

4.7. Overall Discussion

Theoretically, the best model should be AMX3502 for equation 1, AMX4233 for equation 2 and AMX5251 for equation 3. This all best model was get from calculate then_a,n_b , n_c and nk for each equation. For the indicator which is name MSE and FPE, the result shows that the best model is not the same with the theoretical model for the all three equation. It is because more terms will cause more accurate for the model.

Same with the model output indicator, the result show the best model not same as with the theoretical. It shows the best model is other model it not should not be the best model. This happen to all 3 equation results because there might be coincidence in the simulated error term.

For the model residual, the equation 1 result show the all model are best because the graph line are between the confidence interval. But for equation 2, it supposed to model AMX4233 to be best model but the result show the best model is AMX4263, This may due to ill-conditioning problem. Ill-conditioning happens when the residuals are correlated. This is because the variable is too large. If the condition number is large, then the matrix is said to be ill-conditioned. However, the second graph for both models shows that the residual lies at confidence lines and shows no correlation between the residuals and input. For the equation 3, the result is good because one of the best model is a true model which is model AMX5251.

For the parameter value, the equation 1 show that the only one model is the best model, it is AMX3502 and it same with the true model. But in equation 2, the result show the best model is AMX4243, the result should show the model AMX4233 as the best model. And also, for equation 3, the result still shows the wrong model as the best model. However, Amx5251 which is the true model is not for from expectation

The accuracy of a measurement system is the degree of closeness of measurements of a quantity to that quantity's true value. Increasing the number of terms or samples lead to greater confidence in calculating an accurate average measurement and to eliminate the error. Moreover, repeating the experiment more than once helps determine if the data was a fluke or represents the normal case.

CHAPTER 5

CONCLUSION AND RECOMMENDERATION

4.1. Conclusion

For this project, the objective is to investigate the effect of error order in system identification that performs by using linear difference equation model. Model ARX and ARMAX was applied throughout this investigation. In the beginning of project, much research through book and internet was made to make sure, the understanding about all system identification application and process flow. After that, before performing the experiment, GUI familiarization was made by read some information on internet and journal.

To perform this project, three equations was randomly create in two model which is one ARX model and two another is ARMAX model. From this equation, we place it into a coding program and it will create five hundred data to be performed in system identification. The coding was modified with the supervisor guide and discussions, the program must be perfectly run to create the data and the explanation was being discussed in data acquisition on chapter 3.

System identification was performing with the created data and obtained the result with the different performance indicator. For the result, even it not show the good result compare to the theoretical but it can still consider it which in term to investigate the effect. It still show that equation supposed to be the best result is near to the true result.

4.2. Recommendation

Last but not least, recommendation for future work is the data need to be more scattered for a better result. It is because the data play the important which is to maximize the accuracy of the data and to minimize the error so that one can draw accurate conclusion. In this investigation we can state that the data maybe not scatter, so the probability to get a good result is low.

Besides, another recommendation is increasing sample size is a standout amongst the most well-known approaches to lessen the experimental error. Essentially, the larger the sample size in a test, the more likelihood to detect effects from changing a variable. If the sample size is larger, we can get the more precise result compare to the small sample size.

REFERENCES

Ales Prochazka, N. K. (2013). *Signal Analysis and Prediction.* Prague, Czech Republic: Springer Science & Business Media.

Astrom, K. J. (1971). System Identification A Survey. *Automatica*, 125.

Fung, E. H. (2003). Modelling and prediction of machining errors using. *Applied Mathematical Modelling*, 614.

J.Bruls. (199). Linear and Non-linear System Identification Using Separable Least-Squares. *Europe Journal of Control*, 116-117.

Ljung, L. (2010). Perspectives on system identification. *Annual Reviews in Control*, 1.

Ljung., L. (1987). System Identification : Theory For The Use. *Introduction*, 1.

Strejc, V. (1980). Least Squares Parameter Estimation. *Introduction*, 535.

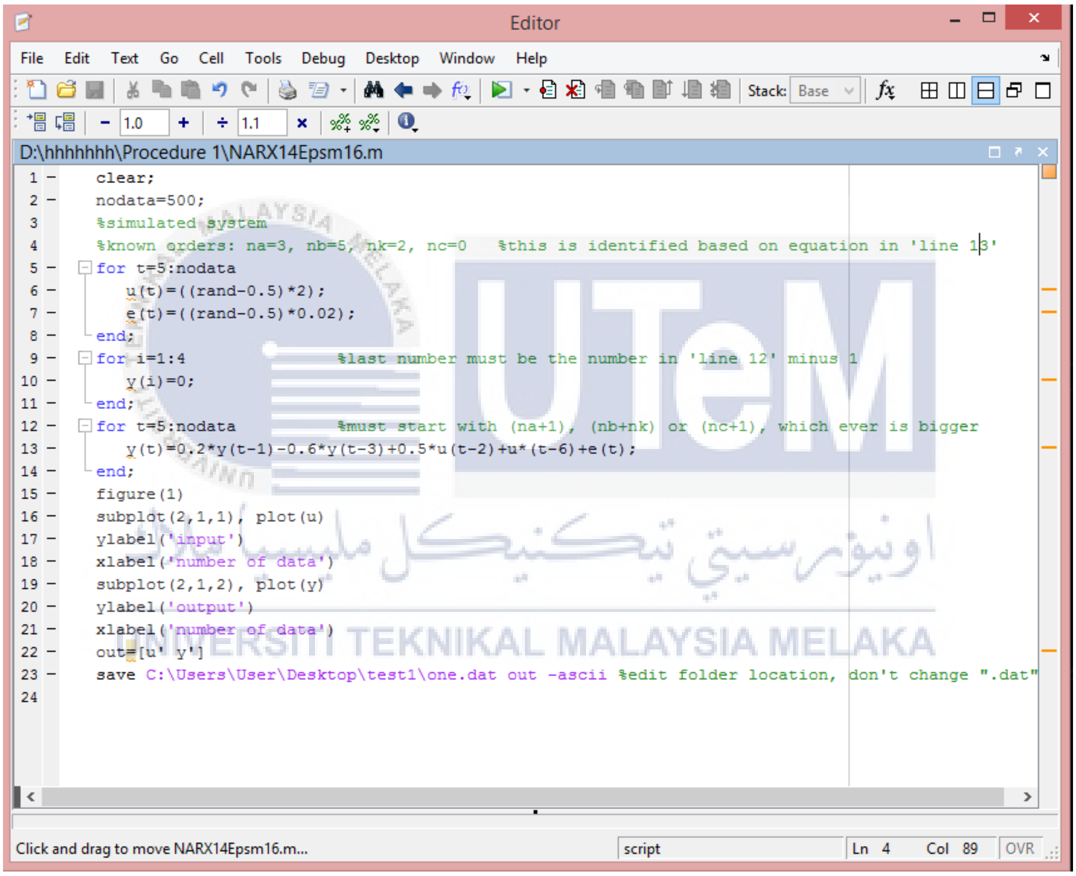
V.Beck, J. (1977). *Parameter Estimation In Engineering And Science.* New York: John Wiley & Sons.

Ales Prochazka, N. K. (2013). *Signal Analysis and Prediction.* Prague, Czech Republic: Springer Science & Business Media.

Fung, E. H. (2003). Modelling and prediction of machining errors using. *Applied Mathematical Modelling*, 614.

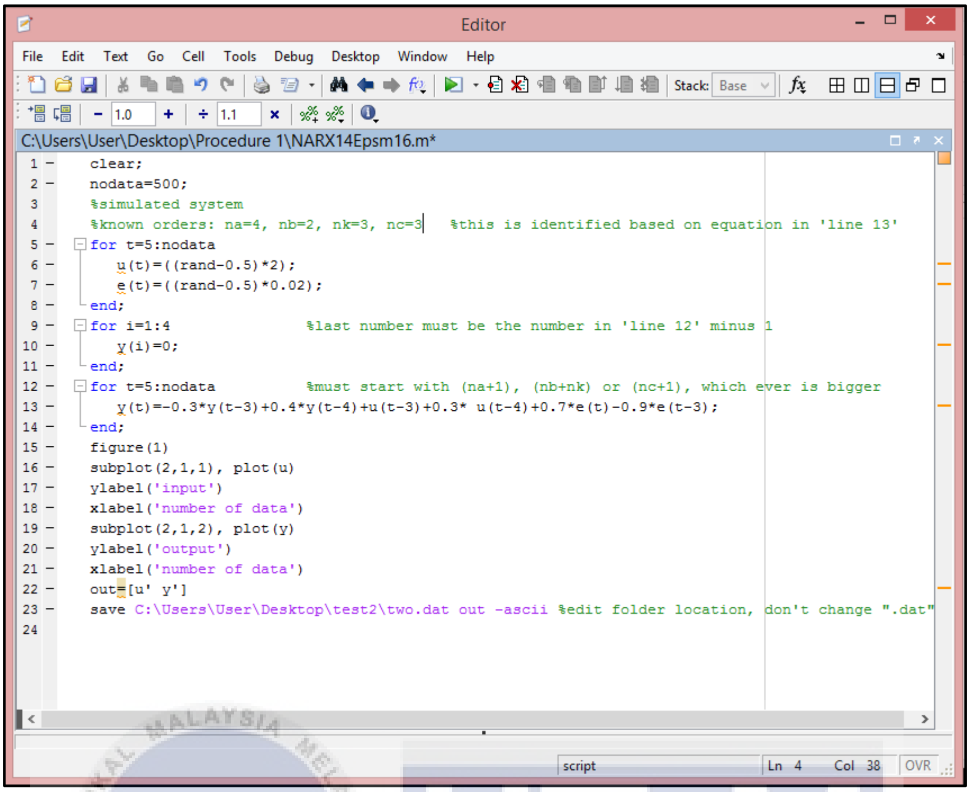
Vincent, D. (2005). Estimation of LF glottal source parameters based on an ARX model. *INTERSPEECH*, 333.

APPENDIX

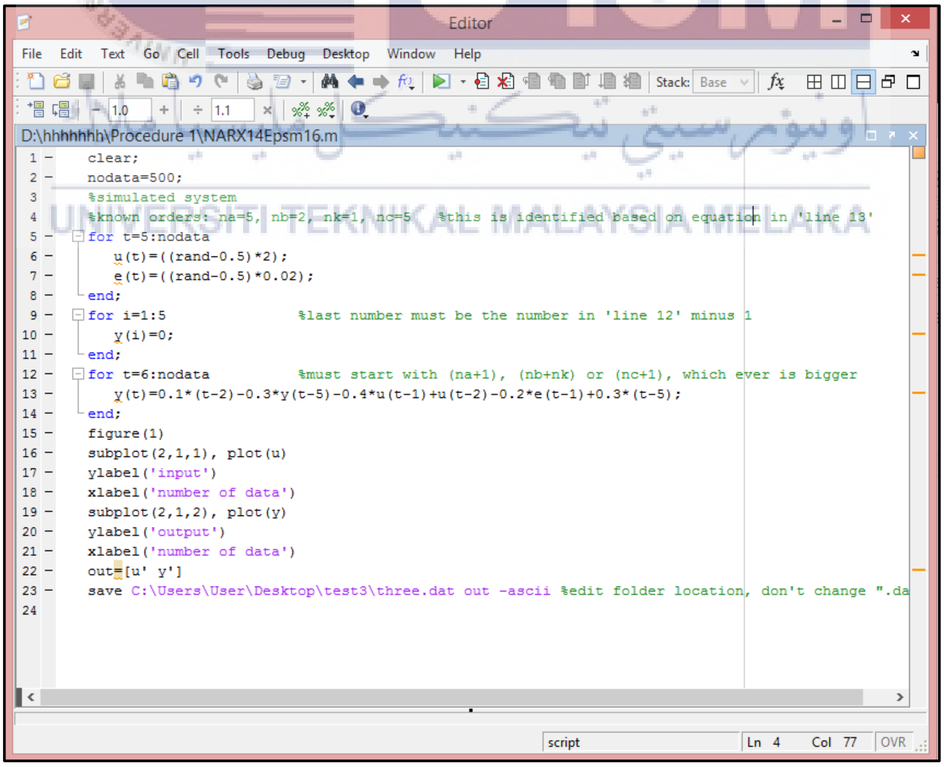



```
1  clear;
2  nodata=500;
3  %simulated system
4  %known orders: na=3, nb=5, nk=2, nc=0   %this is identified based on equation in 'line 13'
5  for t=5:nodata
6      u(t)=((rand-0.5)*2);
7      e(t)=((rand-0.5)*0.02);
8  end;
9  for i=1:4                  %last number must be the number in 'line 12' minus 1
10     y(i)=0;
11 end;
12 for t=5:nodata             %must start with (na+1), (nb+nk) or (nc+1), which ever is bigger
13     y(t)=0.2*y(t-1)-0.6*y(t-3)+0.5*u(t-2)+u*(t-6)+e(t);
14 end;
15 figure(1)
16 subplot(2,1,1), plot(u)
17 ylabel('input')
18 xlabel('number of data')
19 subplot(2,1,2), plot(y)
20 ylabel('output')
21 xlabel('number of data')
22 out=[u' y']
23 save C:\Users\User\Desktop\test1\one.dat out -ascii %edit folder location, don't change ".dat"
24
```


Program Coding Equation 1



Program Coding Equation 2



Program Coding Equation 3

PSM 1 Gantt Chart

Task	Week													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1.0 Collect Information	■	■	■	■										
1.1 Find Journal, Book, Internet	■	■												
1.3 Study About SI			■	■										
2.0 Familiarization					■	■	■	■						
2.1 Information About MATLAB					■	■								
2.2 Explore MATLAB							■	■						
3.0 Trial Run									■	■	■	■	■	
3.1 Try Demo GUI									■	■				
3.2 Preliminary Result											■	■		
3.3 Result Discussion													■	
4.0 Conclusion														■

PSM 2 Gantt Chart

Task	Week													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1.0 PSM 1 Report Correction	■	■	■											
1.1 Chapter 1	■													
1.2 Chapter 2		■												
1.3 Chapter 3			■											
2.0 Methodology				■	■	■	■	■	■	■	■			
2.1 Data Acquisition				■	■	■								
2.2 Perform SI					■	■	■	■	■					
3.0 Result									■	■	■			
4.0 Discussion												■	■	
5.0 Conclusion														■