

Multiple-Input Single-Output System Identification Techniques Using the Pebble Bed Modular Reactor Data.

E.C. Lamprecht

Bachelors in Computer and Electronic Engineering

**Mini-dissertation submitted as partial fulfilment of
requirements for the Masters Degree in Computer and
Electronic Engineering**

Supervisor: Prof. W.C. Venter

March 2004

Acknowledgements

Firstly I would like to thank God for all the blessings I have received in my life.

I'd like to thank my parents, Erwin and Marlenè Lamprecht as well as my sister Marion for their support and encouragement throughout this study.

I'd like to thank Nicole Nys for giving me inspiration during the latter stages of the study.

I'd like to thank Prof. Willie C. Venter for his help and guidance.

Thank you to all those people not mentioned here for their support and interest during this study.

Thank you to the PBMR (Pty.) Ltd. and M-Tech industrial for making this study possible.



Abstract

Models are used to describe the dynamic behaviour of a system, to predict future outputs of the system and are useful when designing certain control schemes.

An effective control scheme could be used to influence the dynamic behaviour of a system in such a way that it exhibits more desirable dynamic behaviour. A control system could be designed to increase the efficiency of a system. This makes it obvious that accurate models are very useful.

The focus of this study is to use Multiple-Input Single-Output (MISO) system identification techniques on data obtained from the Flownex simulation package. These techniques are used to obtain a (MISO) mathematical model for the Pebble Bed Modular Reactor (PBMR).

MISO system identification techniques are used in this project to study the effect that the inputs have on each other. This information helps in the understanding of processes within the system. The reason for studying the MISO systems and not the Single-Input Single-Output (SISO) systems is because the field of interest focuses on the effects the inputs have on each other.



Opsomming

Modelle word in ingenieurswese gebruik om die dinamiese gedrag van stelsels te beskryf. Modelle word ook gebruik om toekomstige uitsette van 'n stelsel te voorspel. Modelle word ook gebruik wanneer beheerstelsels ontwerp word.

Effektiewe beheerstelsels kan gebruik word om die dinamiese gedrag van 'n stelsel te beïnvloed. Die beheerstelsel kan so ontwerp word dat die uitset van die stelsel so beïnvloed word dat die stelsel meer effektief is.

In die studie is gefokus op die identifikasie van Meervoudige-Inset Enkel-Uitset (MIEU) stelsels. Die data wat gebruik is, is verkry van die simulatie pakket Flownex. Stelsel identifikasie tegnieke word in die studie gebruik om 'n (MIEU) oordragsfunksie vir die Korrel Bed Modulêre Reaktor te verkry.

Die stelsel identifikasie tegnieke word in die studie gebruik om die effekte wat die verskillende insette op mekaar het, te ondersoek. Die ondersoek na die effekte wat die insette op mekaar het kan help om die onderlinge prosesse van die stelsel te verstaan. Die rede vir die identifikasie van 'n meervoudige inset stelsel en nie 'n enkelvoudige inset stelsel nie, is dat die fokus van die studie val op die effekte wat die insette op mekaar het.



Table of content

ABSTRACT	III
1. INTRODUCTION	1
1.1 BACKGROUND	2
1.2 OTHER METHODS THAT USE OBSERVED DATA TO CONSTRUCT MODELS	4
1.3 PROBLEM STATEMENT	5
1.4 PROPOSED SOLUTION	6
1.5 RESEARCH SUB PROBLEMS	6
1.6 IMPORTANT TERMS	7
1.7 OVERVIEW ON RESEARCH	8
2. SYSTEM IDENTIFICATION THEORY	9
2.1 INTRODUCTION TO THEORY IN THIS CHAPTER	10
2.2 LINEAR, TIME INVARIANT AND CAUSAL SYSTEMS	11
2.3 THE SYSTEM IDENTIFICATION PROCEDURE	12
2.4 DISTURBANCES	15
2.5 THE COVARIANCE FUNCTION	18
2.6 TRANSFER FUNCTIONS	19
2.7 PARAMETRIC AND NONPARAMETRIC IDENTIFICATION	20
2.8 NON-PARAMETRIC PARAMETER IDENTIFICATION	21
2.8.1 <i>The least means squared method</i>	22
2.8.2 <i>The prediction error method</i>	24
2.8.2.1 One-step-ahead prediction of v	25
2.8.2.2 One-step-ahead prediction of y	26
2.9 MODELS OF LINEAR TIME INVARIANT SYSTEM IDENTIFICATION	27
2.9.1 <i>The general model structures for linear time-invariant models</i>	27
2.9.1.1 The SISO model structures	28
2.9.1.2 The MIMO model structures	28
2.9.2 <i>Box-Jenkins model structure</i>	31
2.9.2.1 The algorithm that is used to calculate the Box Jenkins model	31
2.9.3 <i>Output error model structure</i>	32
2.9.3.1 The algorithm that is used to calculate the Output Error model	32
2.9.4 <i>ARMAX model structure</i>	33
2.9.4.1 The algorithm that is used to calculate the ARMAX model	34
2.9.5 <i>ARX model structure</i>	34



2.9.5.1 The algorithm that is used to calculate the ARX model.....	35
2.9.6 <i>State Space model structure</i>	36
2.9.6.1 The algorithm that is used to calculate the Sate Space model	36
2.9.7 <i>The advantages and disadvantages of the model structures</i>	38
2.10 MODEL ERRORS	39
2.10.1 <i>Bias-errors</i>	39
2.10.2 <i>Variance errors</i>	40
3. THE PEBBLE BED MODULAR REACTOR.....	41
3.1 AN INTRODUCTION TO THE PEBBLE BED MODULAR REACTOR.....	42
3.2 HOW THE PBMR WORKS	42
3.2.1 <i>The reactor</i>	44
3.2.2 <i>The power conversion unit (PCU)</i>	45
3.3 CONTROL OVERVIEW	48
3.4 THE MISO AND SISO SUBSYSTEMS.....	49
3.4.1 <i>The three SISO subsystems</i>	49
3.4.1.1 Low-Pressure injection (LPI)	49
3.4.1.2 High-Pressure extraction (HPE).....	49
3.4.1.3 Bypass valve regulation (BPV)	50
3.4.2 <i>The MISO subsystem</i>	51
3.5 THE PEBBLE BED MICRO MODEL	52
3.6 FLOWNEX	53
4. SYSTEM IDENTIFICATION OF THE PEBBLE BED MICRO MODEL....	54
4.1 INTRODUCTION TO MISO IDENTIFICATION OF PBMR	55
4.2 THE SYSTEM IDENTIFICATION TOOLBOX IN MATLAB©.....	55
4.2.1 <i>The implementation of the ARX LMS algorithm</i>	56
4.2.2 <i>The implementation of the Prediction Error algorithm</i>	57
4.2.3 <i>The implementation of the State Space (n4sid) algorithm</i>	60
4.3 THE EXPERIMENTAL SETUP	61
4.3.1 <i>The choice of a input signal</i>	63
4.3.2 <i>The sampling interval</i>	64
4.3.3 <i>The pre-processing of data</i>	64
4.3.4 <i>Results and their interpretation</i>	65



4.4	SISO MODEL IDENTIFICATION	67
4.4.1	<i>Low-Pressure injection system identification</i>	67
4.4.1.1	Low-Pressure injection simulation.....	67
4.4.1.2	ARX model identification	70
4.4.1.3	ARMAX model identification	77
4.4.1.4	Output Error model identification.....	82
4.4.1.5	Box-Jenkins model identification	87
4.4.1.6	State space model identification	92
4.4.1.7	Validation of best model for Low-Pressure injection system identification.....	96
4.4.2	<i>High-Pressure extraction system identification</i>	97
4.4.2.1	High-Pressure extraction simulation	97
4.4.2.2	ARX model identification	99
4.4.2.3	ARMAX model identification	105
4.4.2.4	Output Error model identification.....	111
4.4.2.5	Box-Jenkins model identification	115
4.4.2.6	State space model identification	120
4.4.2.7	Validation of best model for High-Pressure extraction system identification.....	123
4.4.3	<i>Bypass valve regulation system identification</i>	123
4.4.3.1	Bypass valve regulation simulation.....	123
4.4.3.2	ARX model identification	127
4.4.3.3	ARMAX model identification	130
4.4.3.4	Output Error model identification.....	135
4.4.3.5	Box-Jenkins model identification	139
4.4.3.6	State Space model identification	145
4.4.3.7	Validation of best model for Bypass valve regulation system identification.....	149
4.5	MISO SYSTEM IDENTIFICATION	150
4.5.1	<i>Simulation of the MISO model</i>	150



4.5.2	MISO model identification	154
4.5.2.1	ARX model identification	155
4.5.2.2	ARMAX model identification	164
4.5.2.3	Output Error model identification	165
4.5.2.4	Box-Jenkins model identification	168
4.5.2.5	State space model identification	170
4.5.2.6	Validation of best model for the MISO system	170
5.	MISO SYSTEM ANALYSIS	172
5.1	DESCRIPTION OF MISO SYSTEM ANALYSIS	173
5.2	MISO SYSTEM ANALYSIS	173
5.2.1	The influence that the BPV and HPE have on each other	175
5.2.2	The influence that the BPV and LPI have on each other	177
5.2.3	The influence that the LPI and HPE have on each other	179
5.2.4	The influence that the BPV, LPI and HPE have on each other ..	181
6.	CONCLUSION	184
6.1	SUMMARY OF RESULTS	185
6.2	CONTRIBUTIONS OF STUDY	185
6.3	SUGGESTIONS FOR FUTURE WORK	186
6.4	CONCLUDING REMARKS	186
	LIST OF REFERENCES	188



ABBREVIATIONS AND ACRONYMS

AVR	Arbeitsgemeinschaft Versuchsreaktor
ARX	AutoRegressive with eXogenous input
ARMAX	AutoRegressive Moving Average with eXogenous input
BJ	Box-Jenkins
BPV	Bypass Valve
HTGR	High Temperature Gas-cooled Reactor
HTR	High Temperature Reactor
HP	High Pressure
HPBV	High Pressure Bypass Valve
HPC	High Pressure Compressor
HPE	High Pressure Extraction
HPT	High Pressure Turbine
IV	Instrumental Variables
LP	Low Pressure
LPBV	Low Pressure Bypass Valve
LPC	Low Pressure Compressor
LPI	Low Pressure Injection
LPT	Low Pressure Turbine
LPBV	Low Pressure Bypass Valve
LTI	Linear Time-Invariant
MA	Moving Average
MIMO	Multi-Input Multi-Output
MISO	Multi-Input Single-Output
OE	Output Error
PBMM	Pebble Bed Micro Model
PBMR	Pebble Bed Modular Reactor
PCU	Power Conversion Unit



MISO system identification techniques using the PBMR data.

PEM	Prediction Error Method
SIMO	Single-Input Multi-Output
SISO	Single-Input Single-Output
SS	State-Space
TF	Transfer Function
THTR	High Temperature Gas Cooled Reactors



1. Introduction

Background information on white-box, grey-box and black-box modelling is given and a problem statement and proposed solution for modelling the Pebble Bed Modular Reactor is given, followed by the dissertation outline.



1.1 Background

Engineering is concerned with the understanding and controlling of materials, processes, plants, and some parts of nature [1]. In order to understand and control these systems, models of these processes or plants must be build.

In engineering models are used to describe the dynamic behaviour of a system to predict the future outputs of the system. A model of the system is also used when designing control schemes.

An effective control scheme could be used to influence the dynamic behaviour of a system in such a way that it exhibits more desirable dynamic behaviour. The control system could also be designed to increase the efficiency of a system.

There are three approaches to build an accurate model. The three approaches include white-box, grey-box and black-box modelling. Each of these methods has advantages and disadvantages making them more suited for certain applications [2].

The modelling process that builds models using first principals is called white-box modelling. This method uses physical knowledge and fundamental laws, such as Newton's second law to construct models [30]. These models are very difficult to construct as well as very time consuming. Problems arise due to the fact that a very detailed understanding of the process is needed to build a model. A model derived using the white box modelling process is very accurate, expensive to build and computationally intensive to implement.



Black-box modelling refers to modelling techniques that use only input-output data to build a model of the system. Models build using black-box modelling techniques are often used in model-based control schemes, because they are very efficient for modelling the dynamical behaviour of a system and require less engineering time to construct.

Grey box modelling makes use of prior knowledge of a white-box model with a global structural representation of the physical system under study, and incorporates accurate black-box models. This modelling technique is not widely used.

The field of interest is "system identification" (also known as black-box modelling) for linear time-variant multiple-input single-output systems. System identification is a field that builds models that describe the dynamic behaviour of system from observed data. The different methods of identifying a model for a process are illustrated in Figure 1-1.

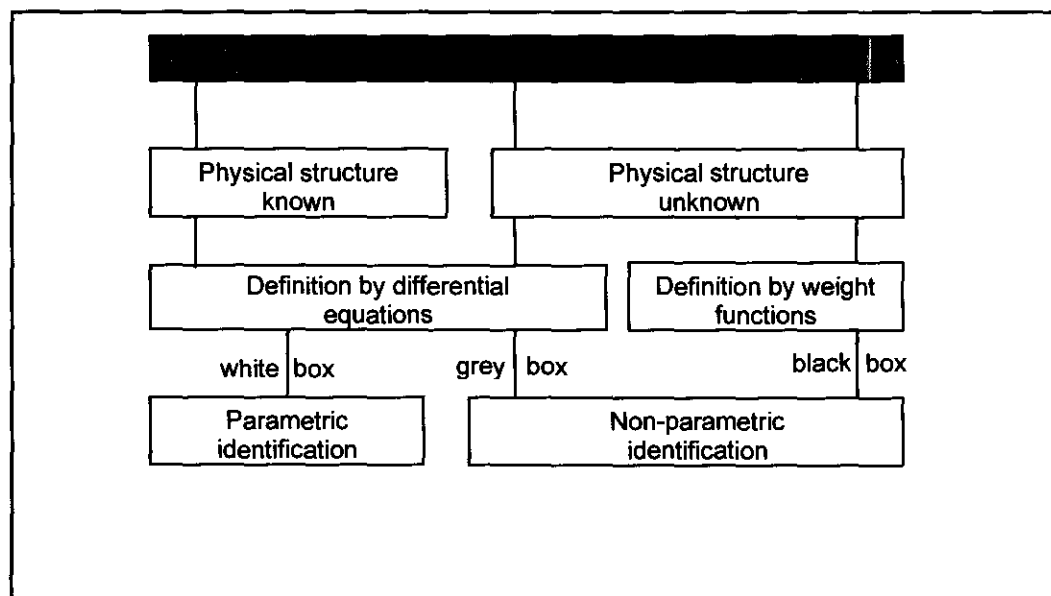


Figure 1-1: Methods used to identify a model [14, 15].



The data that is used originates from a software package known as Flownex. Flownex is a physical model builder and simulator for thermo hydraulic networks (Flownex is described in more detail in section 3.6). The system that is used to produce the data is the Pebble Bed Modular Reactor (PBMR).

1.2 Other methods that use observed data to construct models

There are two other methods that use observed data to construct models that describe the dynamical behaviour of a system. The one method makes use of neural networks to learn the data. The other method is fuzzy logic system identification.

The first method uses neural networks to construct models. This method was used by C.H.Dercksen [3] to describe the dynamical behaviour of the PBMR. Three SISO as well as a MISO model of the PBMR was constructed.

The other method, known as fuzzy logic system identification, uses fuzzy logic techniques to build a model that describes the dynamical behaviour of the system. Fuzzy logic systems are capable of uniformly approximating any nonlinear function over a collection of objects to any degree of accuracy if the collection of objects is compact [4].

In a recent study the ARX model structure (described in section 2.9.5), a system identification model was used in conjunction with fuzzy logic identification techniques to identify more complex non-linear Multi-Input Multi-Output (MIMO) models. [23]



1.3 Problem statement

The effectiveness of the model-based control scheme depends on the quality of the model that is used for control. It is important that an accurate model that can describe the dynamic behaviour of the PBMR is derived to help control the plant more effectively.

In the meantime there are several models available for the PBMR. Some of the models include a model derived from an artificial neural network [3], a simplified state space model developed by the engineers at PBMR, and a single-input single-output model developed by W.O.Erasmus [5]. There is a need to develop a multiple-input single-output model that describes the dynamic behaviour of the PBMR. The reason a MISO model is needed instead of different SISO models is because it is better to incorporate all inputs into a single model for prediction and control purposes [25].

This model could then aid in the development of a model-based control system that is able to control the dynamic behaviour of the PBMR in a desirable manner.

In complicated systems (for example the PBMR) it is difficult to understand all the processes in the system. In most MISO systems the effect the inputs has on the output, influences each other. When neural networks and fuzzy logic methods are used for identification the effects the inputs have on each other could not be predicted.



1.4 Proposed solution

System identification is a field of study that uses observed input-output data from a system to build mathematical models for dynamical systems. These models are actually approximations of real life processes. Some of the models used in system identification also incorporate disturbances that the system might include. This makes it ideal to use system identification to model the dynamical behaviour of real life processes.

It is proposed that system identification techniques be used to obtain a multiple-input single-output mathematical model for the PBMR. This model could aid in the design of the control scheme.

The MISO system identification techniques used will also help understand the effects that the inputs have on each other. The cross correlation between inputs are used to measure the effect that the inputs have on each other.

1.5 Research sub problems

The research and identification of a multiple-input single-output system can be divided into the following sub problems:

- a) The theory of the different models used in linear time-variant system identification must be studied.
- b) The Matlab© System Identification Toolbox must be studied and simple systems must be identified.
- c) Data must be collected for the identification process.
- d) The best model must be selected after the identification process.
- e) The relationship between SISO and MISO systems must be



investigated.

1.6 Important terms

There are certain terms that are important in this field of research. These terms are:

- a) A **system** is an object or process with a certain structure. A system has certain input and output signals. The signals that could be changed by the user are known as inputs. The signals that are produced after the variables in the system have interacted with each other are known as outputs [6].
- b) A **model** according to the Oxford dictionary is a representation or a replica. The interest of this study is the computation of a mathematical model. A mathematical model of a system is a model that describes the behaviour of the system utilizing a mathematical expression of the system. This is very important for some systems, because the stability of the system could be tested with the aid of this mathematical model [10].



1.7 Overview on research

This proposal will be divided into the chapters described below:

- Chapter two describes the theory of system identification. The mathematical models that will be used in this project are also discussed.
- Chapter three is a description of the system that was used to obtain the data, the Pebble Bed Modular Reactor.
- In chapter four the results of the system identification process are documented.
- Chapter five discusses the system analysis that was done on the MISO system.
- The conclusions of this study are described in chapter six.



2. System identification theory

A basic introduction to system identification theory and the system identification procedure is given. The characterizations of linear time-invariant system are given followed by a discussion on the parametric identification method and different parametric model structures for linear time-invariant systems.



2.1 Introduction to theory in this chapter

In this chapter the theory that is relevant to this study is discussed. The system that is used in this study is a non-linear time invariant casual MISO system. The terms non-linear, linear, time invariant, time variant and casual are discussed in section 2.2.

In this study a distinctive procedure is followed to derive a transfer function. This procedure is known as system identification procedure. This procedure is discussed in section 2.3 .

An advantage of system identification to derive transfer functions for sampled dynamical systems is that it is noise tolerant. This makes it suitable for identification of practical systems. The noise or disturbances that occur when data is sampled is discussed in section 2.4.

In system identification the covariance of the input(s) and the output is important, because it is used to estimate the models. Covariance is defined and described in section 2.5.

The result of system identification is a transfer function. Section 2.6 describes how the transfer functions are obtained from the output of the models and also describe the backshift operator that is used.

In this study the software package Matlab© was used to estimate the models. The different algorithms that are used to estimate the models for the system are described in section 2.8.



This study has focused on certain model structures. These different model structures are described in section 2.9.

2.2 Linear, time invariant and causal systems

Consider the system illustrated in Figure 2-1. The system receives the input signal $u(t)$ and produces the output signal $y(t)$.

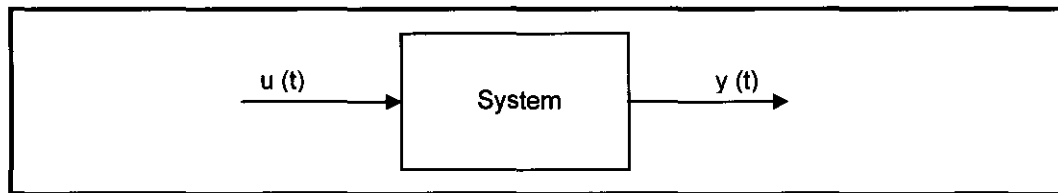


Figure 2-1: The system

A system is time invariant if its response to input $u(t)$ does not depend on absolute time, or in mathematical terms [9]:

$$u(t-t_0) \rightarrow y(t-t_0) \quad (2.1)$$

This means that if a time invariant system receives a time shift at the input signal it will result in the same time shift at the output signal.

The system illustrated in Figure 2-1 receives an input $u_1(t)$. This input produces the output $y_1(t)$, and when it receives input $u_2(t)$ it produces output $y_2(t)$. The system can be defined as a linear system when the following equation is applicable to the system [9]:

$$a_1 u_1(t) + a_2 u_2(t) \rightarrow a_1 y_1(t) + a_2 y_2(t) \quad (2.2)$$



The system can be described as a causal system if at time t_1 the output of the system is dependant only on inputs received at time $t \leq t_1$. This implies that at any time the output of the system only depends on the input at time as well as past inputs, and not future inputs [9].

A system can be described by its impulse response if the system is a linear, time-invariant, causal system. In this case the impulse response of the system is presented by $g(t)$. To predict the output ($y(t)$) of the system when it receives an input $u(t)$, the following equation is used [9]:

$$y(t) = \int_0^{\infty} g(t)u(t-\tau)d\tau \quad (2.3)$$

In the equation above (τ) is a time delay. This implies that the impulse response gives a complete characterization of the system [6].

In system identification sampled data is used. This alters the form of equation (2.3) to the following [6]:

$$y(t) = \sum_{k=1}^{\infty} g(k)u(t-k) \quad (2.4)$$

where $k = 1, 2, \dots$ and $kT \leq t < (k+1)T$.

2.3 The system identification procedure

As stated previously system identification deals with building mathematical models from observed data of dynamical systems [6, 28]. System identification follows a specific procedure to identify dynamical systems. This process is graphically illustrated in Figure 2-2.



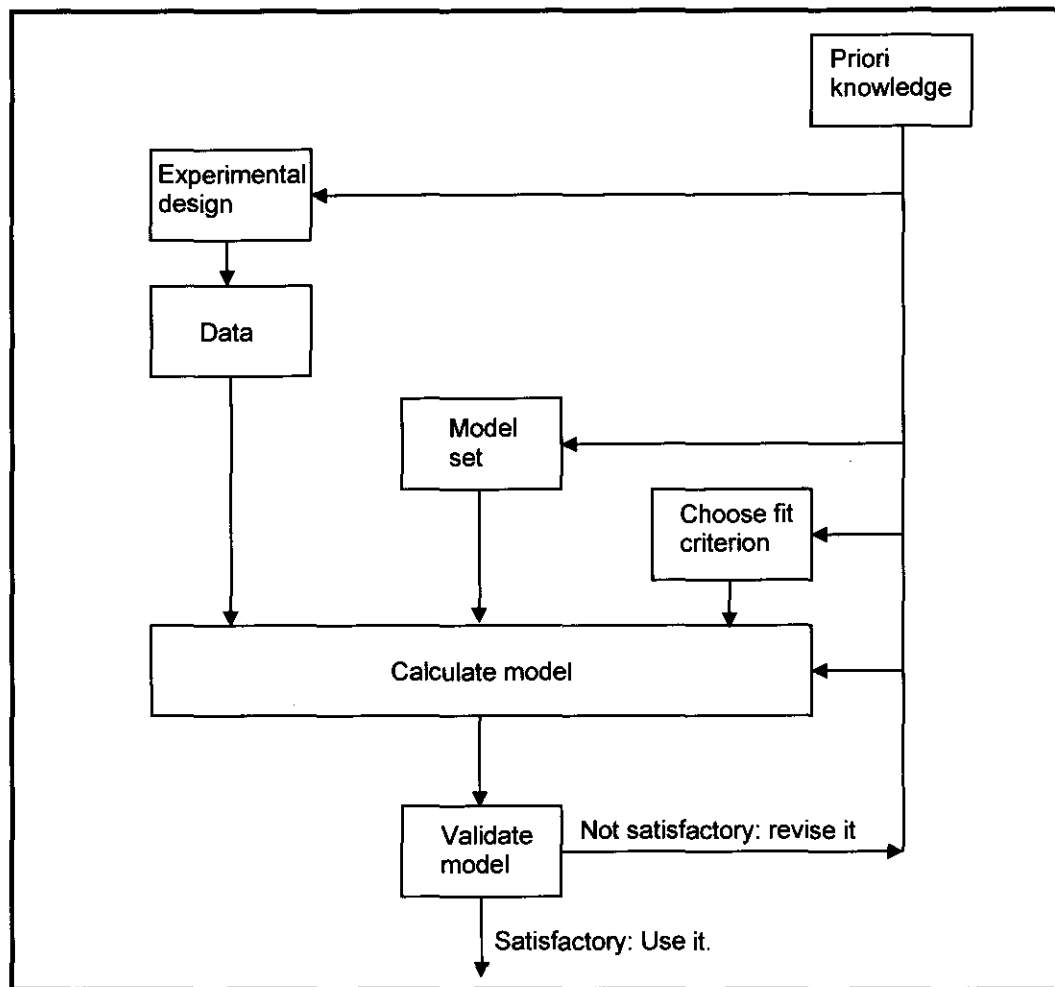


Figure 2-2: The system identification procedure [6].

The system identification procedure starts with the prior knowledge that the engineer gathers utilizing engineering insight. Figure 2-2 clearly illustrates that there are four stages that have to be completed before the model could be calculated.

- The first stage is the designing of an experiment that is able to produce data that is maximally informative. It is important that various input signals are used during the experimental phase. The recorded data could be used to test the system for linearity [6].



The sampling frequency is very important. In case of a bandlimited signal with highest frequency Ω_a , aliasing could be avoided by sampling at a frequency of:

$$\Omega_s = \frac{2\pi}{T} \geq 2\Omega_a \quad (2.5)$$

The frequency Ω_a is referred to as the Nyquist frequency. In practice a signal that is sampled is never bandlimited, thus aliasing almost always occurs. The reason the signal is never bandlimited is due to the introduction of white noise to a system [7]. White noise is discussed in section 2.4.

- The first stage produces a data set (Z^N). The data set is recorded during the specifically designed experiment. This is the second stage in the procedure. In these experiments the user specified the inputs and records the input and output signals [6].
- The third stage in the procedure is the selection of a set of candidate models that will be used later in the identification process. This is the most difficult choice in system identification [6]. Prior knowledge and engineering insight into the physical system is a great asset in this stage.

These models must be chosen carefully because most of them must be able to identify the behaviour of the dynamical system. If the wrong sets of models are chosen the result could be that the system could not be identified as accurately as desired [6].

- The last stage before the model could be calculated is the selection of



a rule to help calculate the best model. The best rule for determining the accuracy of models is also very important. An example of such a rule is the least mean squares method. The rule must be used to determine the accuracy of the model when it attempts to reproduce the measured data [6].

At this stage of the procedure the model could be calculated. All the possible models that are part of the model set that was chosen in the third stage must be used. After the best model is chosen according to the chosen criteria the model must be validated [6].

The reason for the validation process is to verify if the model behaves the same or almost the same as the system. The prior knowledge of the system is used here to evaluate the behaviour of the model. Deficient behaviour of the model leads to rejection of the model. When a model is rejected the system identification procedure or at least part of the procedure is repeated [6].

A model of a system can never be accepted as the true representation of the system. The mathematical model of the system only strives to replicate the behaviour of the system. The better the performance of the model the better the model duplicates the behaviour of the system [6].

2.4 Disturbances

The systems discussed up to this stage were all accepted to be systems without noise. This means that the output can be calculated exactly if the transfer function and the input signal are known. Unfortunately this is almost never the case, because almost all real world systems include noise. Figure



2-3 illustrates a system that includes noise [6].

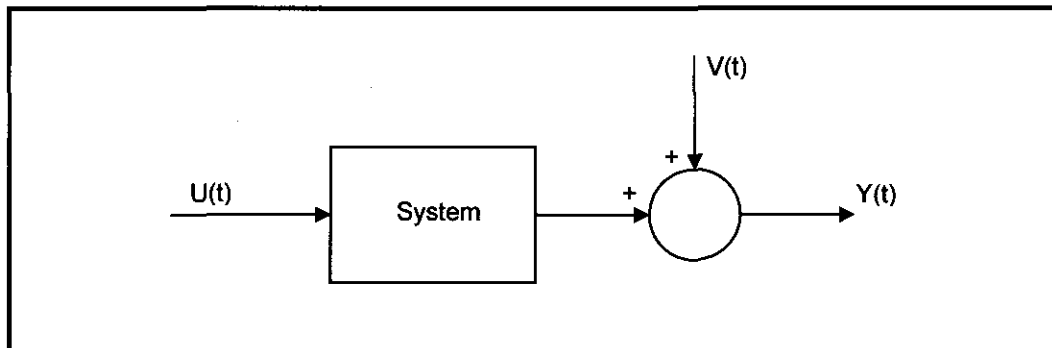


Figure 2-3: A system that includes noise [6].

Noise is totally unpredictable and has many sources. The sources of noise can be divided into two main categories, namely:

- a) *Uncontrolled inputs.* Some systems are influenced by inputs that can not be controlled by the user. These are controlled unwanted inputs. Think of a car as system. There are controllable inputs to the system, but there are also uncontrolled inputs, for example, crosswinds [6].
- b) *Measured noise.* The data that is collected is measured by sensors. These sensors are not perfect. The output of the sensor is almost never exactly the value of the signal that was measured [6].

The system described by equation (2.4) was classified as a linear system. In a linear system the effect of noise could be incorporated by the addition of an extra term. Equation (2.4) could be altered to incorporate disturbances ($v(t)$), the result would be:

$$y(t) = \sum_{k=1}^{\infty} g(k)u(t-k) + v(t) \quad (2.6)$$

White noise ($e(t)$) is defined to be a stationary random process having a



constant spectral density function [11]. The specification of different probability density functions for $\{e(t)\}$ results in different characteristic features for white noise. As an example, if the probability density functions of $e(t)$ is defined as:

$e(t) = 0$, with probability $1 - \mu$ and

$e(t) = r$, with probability μ

where r is a normally distributed random variable: $r \in \mathbb{N}(0, \gamma)$ [6].

This specification leads to disturbance sequences with characteristic profiles occurring at random instances. The probability density function could also be specified as [6]:

$$e(t) \in \mathbb{N}(0, \lambda) \quad (2.7)$$

This specification of $e(t)$ is used to describe noise that occurs irregularly and more frequently.

The most important characteristic of disturbances is that its value is not known beforehand. The past values of disturbances are known, thus the future value of disturbances could then be predicted with a certain probability. Utilizing the characteristics of white noise, that was described, the disturbances in the system illustrated in Figure 2-3 could then be described as [6]:

$$v(t) = \sum_{k=1}^{\infty} h(k) e(t-k) \quad (2.8)$$

Utilizing equation (2.6) the output of a linear, time-invariant, causal system that incorporates noise could be described by the following equation:

$$y(t) = \sum_{k=1}^{\infty} [g(k) u(t-k) + h(k) e(t-k)] \quad (2.9)$$



2.5 The covariance function

The mean or expected value of a discrete random variable X could be computed utilizing the following equation [8]:

$$E[X] = \sum_x x f(x) \quad (2.10)$$

Equation (2.10) could now be used to define the mean of the disturbance that occurred in the system illustrated in Figure 2-3. The white noise defined with equation (2.7) is assumed to have a mean value of zero and a variance of λ . The mean of the disturbance could be computed as [6]:

$$E[v(t)] = \sum_{k=0}^{\infty} h(k) E[e(t-k)] = 0 \quad (2.11)$$

The covariance of the disturbance is then defined as [6]:

$$E[v(t)v(t-\tau)] = \sum_{k=0}^{\infty} \sum_{s=0}^{\infty} h(k)h(s) E[e(t-k)e(t-\tau-s)] \quad (2.12)$$

Equation (2.12) can be simplified to:

$$R_v(\tau) = \sum_{k=0}^{\infty} \sum_{s=0}^{\infty} h(k)h(s) \delta(t-\tau-s) \lambda \quad (2.13)$$

where $R_v(\tau)$ is known as the covariance of the disturbance $v(t)$. Equation (2.13) is then simplified to [6]:

$$R_v(\tau) = \lambda \sum_{k=0}^{\infty} h(k)h(k-\tau) \quad (2.14)$$

When evaluating equation (2.14) it is interesting that the covariance is independent of time and thus the process is stationary [6].



2.6 Transfer functions

The output of a system could be obtained by numerous methods if the input signal is known. The one method that was already briefly discussed was the impulse response of the system. In this section equation (2.4) will be used to obtain a transfer function for a system that could also be used to obtain the output of a system if the input is known.

The following two operators are introduced: the forward shift operator (q), and the backward shift operator q^{-1} . The forward shift operator (q) is defined as [6]:

$$qu(t) = u(t+1)$$

The backward shift operator (q^{-1}) is defined as [6]:

$$q^{-1}u(t) = u(t-1)$$

Equation (2.4) can be used to obtain the output [6]:

$$\begin{aligned} y(t) &= \sum_{k=1}^{\infty} g(k)u(t-k) \\ &= \sum_{k=1}^{\infty} g(k)(q^{-k}u(t)) \\ &= \left[\sum_{k=1}^{\infty} g(k)q^{-k} \right] u(t) \\ &= G(q)u(t) \end{aligned} \tag{2.15}$$

The transfer function is now defined as [6]:

$$G(q) = \sum_{k=1}^{\infty} g(k)q^{-k} \tag{2.16}$$

Equation (2.16) could be transformed to the z-transform and the following is obtained [6]:

$$G(z) = \sum_{k=1}^{\infty} g(z)Z^{-k} \tag{2.17}$$



Similarly $H(q)$ could be defined as (from equation (2.8)) [6]:

$$H(q) = \sum_{k=1}^{\infty} h(k)q^{-k} \quad (2.18)$$

This implies that equation (2.8) now becomes [6]:

$$v(t) = H(q)e(t) \quad (2.19)$$

Equation (2.9) changes to the following transfer function for the system illustrated in Figure 2-3:

$$y(t) = G(q)u(t) + H(q)e(t) \quad (2.20)$$

with $e(t)$ defined the same as equation (2.7) [6].

2.7 Parametric and non-parametric identification

The procedure of identifying models could broadly be divided into two branches. The two fields are: parametric identification and non-parametric identification (as illustrated in Figure 1-1) [6].

When the model of a very sophisticated process has to be built, an approach known as non-parametric identification is followed. This is often done when the physical structure of the system is not known [6].

When the non-parametric identification process is followed, the model structure is chosen prior to model fitting. The problem is to identify the model parameters (coefficients) using the recorded data set. These parameters of the process under investigation relate the input to the output without having any physical significance [13].

The best model is chosen by evaluating the output of the model that was



estimated. The identification process is often an iterative process, because a model is estimated then evaluated. If the evaluation reveals that the model is deficient a different model is chosen and estimated [6].

In parametric identification, no (or very little) assumption is made with respect to the model structure. The model is build by describing the transformation from input to output by known differential equations. The parameters of the model are ones that interpret the dynamic process in a physical manner. This method of model identification is also called white-box modeling [13].

The focus of this study is to concentrate on non-parametric identification. The model structures were pre-selected for the identification process. The models that were selected are described in section 2.9. The orders of the models will be determined using an iterative process.

2.8 Non-parametric parameter identification

The algorithms that are used must minimize the error function. The error is defined as [6]:

$$\varepsilon(t) = y(t, \theta) - \hat{y}(t, \theta) \quad (2.21)$$

In equation (2.21) $y(t)$ is the desired output and $\hat{y}(t)$ is the model output.

The vector θ is the adjustable variables that must be calculated. The vector θ is uniquely defined for each model structure. Assume that N data points were sampled. The loss function is now defined as [6]:

$$V_N(\theta) = \frac{1}{N} \sum_{t=1}^N \varepsilon^2(t, \theta) \quad (2.22)$$



The loss function must be minimized, thus it is logical that θ must be calculated that [6]:

$$\hat{\theta}_N = \arg \min_{\theta} V_N(\theta) \quad (2.23)$$

There are different algorithms that are used to minimize this error function. The two main methods that are used are the least squared method and the prediction error method. These two methods are discussed in more detail in the following sections [6].

2.8.1 The least means squared method

The basic relationship between the input and output is defined as the following linear differential equation [6]:

$$y(t) = a_1 y(t-1) + \dots + a_n y(t-n) = b_1 u(t-1) + \dots + b_m u(t-m) \quad (2.24)$$

In (2.24) y is the output and u is the input. Equation (2.24) could also be expressed as:

$$y(t) = -a_1 y(t-1) - \dots - a_n y(t-n) + b_1 u(t-1) + \dots + b_m u(t-m) \quad (2.25)$$

The following two vectors are now defined:

$$\begin{aligned} \theta &= [a_1 \dots a_n b_1 \dots b_m]^T \\ \varphi(t) &= [-y(t-1) \dots -y(t-n) u(t-1) \dots u(t-m)]^T \end{aligned} \quad (2.26)$$

The least squares method determines the values of the parameters of the vector θ (matrix in the multivariate case). The method uses the vectors θ and φ that are defined uniquely for each model structure.

A function that describes the error between the output of the model and the desired output is defined as:



$$\begin{aligned}
 V_N(\theta, Z^N) &= \frac{1}{N} \sum_{t=1}^N (y(t) - \hat{y}(t, \theta))^2 \\
 &= \frac{1}{N} \sum_{t=1}^N (y(t) - \varphi(t)^T \theta)^2
 \end{aligned} \tag{2.27}$$

In equation (2.27) Z^N is defined as:

$$Z^N = \{u(1), u(2), \dots, u(N), y(1), y(2), \dots, y(N)\} \tag{2.28}$$

and $y(t)$ is the desired output and $\hat{y}(t, \theta)$ is the output of the model. This must be minimized. This means that θ must be calculated in such a way that $V_N(\theta, Z^N)$ is minimized. Equation (2.27) is a second order equation. This means that the function has only one minimum. This minimum is obtained by differentiating equation (2.27) and set it equal to zero. The following is obtained:

$$\begin{aligned}
 \frac{\partial}{\partial \theta} [V_N(\theta, Z^N)] &= 0 \\
 \therefore \frac{2}{N} \sum_{t=1}^N \varphi(t) [y(t) - \varphi(t)^T \theta] &= 0
 \end{aligned} \tag{2.29}$$

solving equation (2.29) for θ :

$$\begin{aligned}
 \sum_{t=1}^N \varphi(t) y(t) &= \sum_{t=1}^N \varphi(t) \varphi^T(t) \theta \\
 \therefore \hat{\theta}_N &= \left[\sum_{t=1}^N \varphi(t) \varphi^T(t) \right]^{-1} \sum_{t=1}^N \varphi(t) y(t)
 \end{aligned} \tag{2.30}$$

In equation (2.30) $\hat{\theta}_N$ is the vector that contains the values that minimizes $V_N(\theta, Z^N)$.

Equation (2.30) is then evaluated for each time step. This results in an over determined set of linear equations [6].



2.8.2 The prediction error method

A common and general method of estimating the parameters is the prediction error approach, where the parameters of the model are chosen so that the difference between the model's (predicted) output and the measured output is minimized. The estimation involves an iterative, numerical search for the best fit.

The model structure that the prediction error method uses is described by the following equation:

$$\begin{aligned}y(t) &= G(q, \theta)u(t) + v(t) \\ &= G(q, \theta)u(t) + H(q, \theta)e(t)\end{aligned}\tag{2.31}$$

For a stationary process, without loss of generality, $H(q, \theta)$ is assumed to be a stable, stably invertible and a normalized ($H(\infty, \theta) = 1$) transfer function [6]. The prediction error method predicts the next output of the model. This is achieved by first predicting the next value of v and then using the prediction of v the next output of the model (y) is predicted.

These two methods are discussed in the following two sections. The output is compared to the desired output. The parameter vector θ is adjusted. This process is repeated until certain criteria are met. This method is based on the maximum likelihood methods [29].



2.8.2.1 One-step-ahead prediction of v

A pragmatic view of a model is to view it as a way of determining the next output value given previous observations. To gain the one-step-ahead prediction for equation (2.31), we need to compute the one-step-ahead prediction value of the disturbance $v(t)$ in equation (2.32). For normalizations it is assumed that $h(0)=1$ [6]. Suppose we observed $v(s), s \leq t-1$, the disturbance $v(t)$ is defined as [6]:

$$v(t) = \sum_{k=0}^{\infty} h(k)e(t-k) = e(t) + \sum_{k=1}^{\infty} h(k)e(t-k) \quad (2.32)$$

According to the invertibility of the noise model [6], the knowledge of $v(s), s \leq t-1$ implies the knowledge of $e(s), s \leq t-1$. The term $\sum_{k=1}^{\infty} h(k)e(t-k)$ in equation (2.32) is therefore known at time $t-1$. To determine the one-step-ahead prediction of v the mean value of the disturbance is used. Since the white noise signal $e(t)$ has zero mean, one-step-ahead prediction of v ($\hat{v}(t-1)$) is defined as [6]:

$$\hat{v}(t|t-1) = \sum_{k=1}^{\infty} h(k)e(t-k) \quad (2.33)$$

The invertibility of the noise model could again be used to write equation (2.33) as [6]:

$$\begin{aligned} \hat{v}(t|t-1) &= \left[\sum_{k=1}^{\infty} h(k)q^{-k} \right] \\ &= [H(q)-1]e(t) \\ &= \left[\frac{H(q)-1}{H(q)} \right] v(t) \\ &= [1-H^{-1}(q)]v(t) \end{aligned} \quad (2.34)$$



$H(q)$ is now applied to both sides of the equation (2.35) and the following is obtained [6]:

$$H(q)\hat{v}(t|t-1) = [H(q)-1]v(t) = \sum_{k=1}^{\infty} h(k)e(t-k) \quad (2.36)$$

This equation is now used to predict the next output of the model.

2.8.2.2 One-step-ahead prediction of y

The one-step-ahead prediction of y can now be determined using the one-step-ahead prediction of v. Assume that y(s) and u(s) are known for $s \leq t-1$.

From equation (2.31) the following is defined [6]:

$$v(s) = y(s) - G(q)u(s) \quad (2.37)$$

This means that v(s) is also known for $s \leq t-1$ and we would like to predict the value [6]:

$$y(t) = G(q)u(t) + v(t) \quad (2.38)$$

The one-step-ahead prediction of y from equation (2.38) is then given as [6]:

$$\begin{aligned} \hat{y}(y|y-1) &= G(q)u(t) + \hat{v}(t|v-1) \\ &= G(q)u(t) + [1 - H^{-1}(q)]v(t) \\ &= G(q)u(t) + [1 - H^{-1}(q)][y(t) - G(q)u(t)] \\ &= H^{-1}(q)G(q)u(t) + [1 - H^{-1}(q)]y(t) \end{aligned} \quad (2.39)$$

Equation (2.39) is used to predict the next output of the model and this value is compared to the desired output. This error between the models predicted output and the desired output are then used to calculate a value that is used to update the parameters. This value is calculated by the gradient and a search direction. The various search direction methods that are used to calculate the update value are described in section 4.2.2.



2.9 Models of linear time invariant system identification

2.9.1 The general model structures for linear time-invariant models

A linear time-invariant system could be described by the following equation [6]:

$$y(t) = G(t, \theta)u(t) + w(t) \quad (2.40)$$

where $w(t)$ is a disturbance that is defined as [6]:

$$w(t) = H(t, \theta)e(t) \quad (2.41)$$

In equation (2.41) $e(t)$ is white noise. Combining equation (2.40) and (2.41) the following is obtained [6]:

$$y(t) = G(t, \theta)u(t) + H(t, \theta)e(t) \quad (2.42)$$

The backshift operator introduced in section 2.6 is then used to define polynomials $B(q)$, $C(q)$, $D(q)$ and $F(q)$ (defined in section 2.9.1.1) These polynomials are then used to describe a general model structure [6]:

$$y(t) = \frac{B(q)}{F(q)}u(t - nk) + \frac{C(q)}{D(q)}e(t) \quad (2.43)$$

In equation (2.43) nk is the amount of time delays that passes before an input signal produces an effect on the output. When equation (2.42) and equation (2.43) is compared it is clear that [6]:

$$G(t, \theta) = \frac{B(q)}{F(q)} \quad (2.44)$$

and

$$H(t, \theta) = \frac{C(q)}{D(q)} \quad (2.45)$$

The orders of the polynomials $B(q)$, $C(q)$, $D(q)$ and $F(q)$ are defined by n_b , n_f , n_c , and n_d .



2.9.1.1 The SISO model structures

The polynomials $B(q)$, $C(q)$, $D(q)$ and $F(q)$ are defined as [34]:

$$B(q) = b_1 + b_2 q^{-1} + b_3 q^{-2} + \dots + b_{n_b} q^{-n_b+1} \quad (2.46)$$

$$C(q) = 1 + c_1 q^{-1} + \dots + b_{n_c} q^{-n_c} \quad (2.47)$$

$$D(q) = 1 + d_1 q^{-1} + \dots + d_{n_d} q^{-n_d} \quad (2.48)$$

$$F(q) = 1 + f_1 q^{-1} + \dots + f_{n_f} q^{-n_f} \quad (2.49)$$

The values of b_j , c_j , d_j and f_j are scalar values that are determined by the different algorithms [25].

2.9.1.2 The MIMO model structures

When a system has multiple inputs and outputs it is referred to as a MIMO system. MIMO systems are very complex and have a very rich internal structure. The polynomial that was defined in the previous section is now redefined in order to be used for identification of MIMO models.

The $B(q)$ in equation (2.46) will change for the multivariate case. It is now defined as [34]:

$$B(q) = B_0 + B_1 q^{-1} + B_2 q^{-2} + \dots + B_{n_b} q^{-n_b} \quad (2.50)$$

Equation (2.50) could also be expressed in the following matrix form [34]:

$$B(q) = \begin{pmatrix} b_{11}(q) & b_{12}(q) & \dots & b_{1n_u}(q) \\ b_{21}(q) & b_{22}(q) & \dots & b_{2n_u}(q) \\ \dots & \dots & \dots & \dots \\ b_{n_y1}(q) & b_{n_y2}(q) & \dots & b_{n_y n_u}(q) \end{pmatrix} \quad (2.51)$$



The matrix has ny (number of outputs) rows and nu (number of inputs) columns. The values (b_{mj}) of the matrix above could also be described as a polynomial. The polynomial is defined as [34]:

$$b_{mj}(q) = b_{mj}^1 q^{-nk_{mj}} + \dots + b_{mj}^{nb} q^{-nk_{mj}-nb+1} \quad (2.52)$$

The values b_{mj}^{nb} are scalars that are calculated by the algorithm described in section 2.9.5.1.

The polynomial (vector) $F(q)$ changes to a matrix for the multivariate case. It is now defined as [34]:

$$F(q) = F_0 + F_1 q^{-1} + \dots + F_{n_f} q^{-n_f} \quad (2.53)$$

Equation (2.53) could also be expressed in the following matrix form [34]:

$$F(q) = \begin{pmatrix} f_{11}(q) & f_{12}(q) & \dots & f_{1n_u}(q) \\ f_{21}(q) & f_{22}(q) & \dots & f_{2n_u}(q) \\ \dots & \dots & \dots & \dots \\ f_{n_y1}(q) & f_{n_y2}(q) & \dots & f_{n_y n_u}(q) \end{pmatrix} \quad (2.54)$$

The matrix has ny (number of outputs) rows and nu (number of inputs) columns. The values (f_{mj}) of the matrix above could also be described as a polynomial. The polynomial is defined as [34]:

$$f_{mj}(q) = f_{mj}^1 q^{-nk_{mj}} + \dots + f_{mj}^{n_f} q^{-nk_{mj}-n_f+1} \quad (2.55)$$

The values $f_{mj}^{n_f}$ are scalars that are calculated by the algorithm described in section 2.9.3.1.

The $C(q)$ in equation (2.47) will change for the multivariate case. It is now defined as [34]:

$$C(q) = I_{n_y} + C_1 q^{-1} + \dots + C_{n_c} q^{-n_c} \quad (2.56)$$

Equation (2.56) could also be expressed in the following matrix form [34]:



$$C(q) = \begin{pmatrix} c_{11}(q) & c_{12}(q) & \dots & c_{1n_u}(q) \\ c_{21}(q) & c_{22}(q) & \dots & c_{2n_u}(q) \\ \dots & \dots & \dots & \dots \\ c_{n_y,1}(q) & c_{n_y,2}(q) & \dots & c_{n_y,n_u}(q) \end{pmatrix} \quad (2.57)$$

The matrix has n_y (number of outputs) rows and n_u (number of inputs) columns. The values $c_{mj}^{n_c}$ of the matrix above could also be described as a polynomial. The polynomial is defined as [34]:

$$c_{mj}(q) = c_{mj}^1 q^{-1} + \dots + c_{mj}^{-n_c} q^{-n_c} \quad (2.58)$$

The values $c_{mj}^{n_c}$ are scalars that are calculated by the algorithm described in section 2.9.5.1.

The $D(q)$ in equation (2.48) will change for the multivariate case. It is now defined as [34]:

$$D(q) = I_{n_u} + D_1 q^{-1} + \dots + D_{n_d} q^{-n_d} \quad (2.59)$$

Equation (2.59) could also be expressed in the following matrix form [34]:

$$D(q) = \begin{pmatrix} d_{11}(q) & d_{12}(q) & \dots & d_{1n_u}(q) \\ d_{21}(q) & d_{22}(q) & \dots & d_{2n_u}(q) \\ \dots & \dots & \dots & \dots \\ d_{n_y,1}(q) & d_{n_y,2}(q) & \dots & d_{n_y,n_u}(q) \end{pmatrix} \quad (2.60)$$

The matrix has n_y (number of outputs) rows and n_u (number of inputs) columns. The values $d_{mj}^{n_d}$ of the matrix above could also be described as a polynomial. The polynomial is defined as [34]:

$$d_{mj}(q) = d_{mj}^1 q^{-1} + \dots + d_{mj}^{-n_d} q^{-n_d} \quad (2.61)$$

The values $d_{mj}^{n_d}$ are scalars that are calculated by the algorithm described in section 2.9.5.1 [6].



2.9.2 Box-Jenkins model structure

When all of the polynomials in equations (2.46) – (2.49) are used to determine a model, the Box-Jenkins (BJ) model structure is defined as [6]:

$$y(t) = \frac{B(q)}{F(q)} u(t - nk) + \frac{C(q)}{D(q)} e(t) \quad (2.62)$$

Using the BJ model structure the disturbance properties are modelled separately from the system dynamics. The model structure could also be represented by the following block diagram [25]:

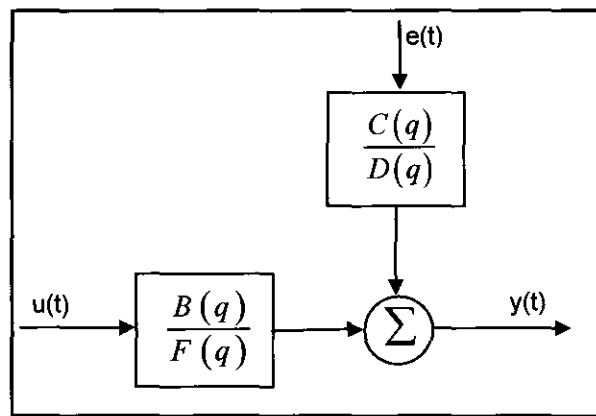


Figure 2-4: A block diagram of the BJ model structure [25].

2.9.2.1 The algorithm that is used to calculate the Box Jenkins model

The prediction error algorithm is used to calculate the parameters. The parameters are calculated iteratively. After one data cycle (one iteration) the parameters are updated by a calculated amount. This amount of change in parameters is determined by different search direction methods as well as the gradient. The search direction methods and their selection are discussed in more detail in section 4.2.2 [34].

The iterative search stops when a local minimum is reached or the maximum amount of iterations is reached. The property that is tested if a local minimum



is reached is “Tolerance” which is described in more detail in the Matlab® implementation section, section 4.2.2. The “MaxIter” property specifies the maximum amount of iterations that should be used. The property is described in more detail in the Matlab® implementation section, section 4.2.2 [34].

2.9.3 Output error model structure

When the noise signal is not modelled at all, ($H(t, \theta) = 1$) the Output Error (OE) model structure is obtained. In Figure 2-5 a block diagram representation of the OE model is illustrated [25].

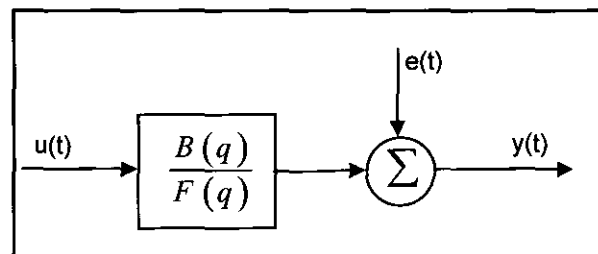


Figure 2-5: A block diagram of the OE model structure [25].

2.9.3.1 The algorithm that is used to calculate the Output Error model

The same algorithm that was described in the previous section that is used to calculate the BJ model is also used to calculate the OE model. The only difference is that a different search direction method is used to calculate the amount that the parameters are updated. This method has less parameters to calculate and thus is less computationally extensive [34].



2.9.4 ARMAX model structure

When the denominators of equation (2.62) are equal, a new polynomial $A(q)$ is formed and is defined as [34]:

$$A(q) = 1 + a_1 q^{-1} + a_2 q^{-2} + \dots + a_{n_a} q^{-n_a} \quad (2.63)$$

for the SISO case the values of a_r are scalars that are determined by the algorithm described in section 2.9.4.1. When a multivariate system is modelled $A(q)$ is defined as [34]:

$$A(q) = I_{n_y} + A_1 q^{-1} + \dots + A_{n_a} q^{-n_a} \quad (2.64)$$

The I in equation (2.64) is an identity matrix with n_y rows and n_y columns.

Equation (2.64) could be expressed in the following matrix form [34]:

$$A(q) = \begin{pmatrix} a_{11}(q) & a_{12}(q) & \dots & a_{1n_y}(q) \\ a_{21}(q) & a_{22}(q) & \dots & a_{2n_y}(q) \\ \dots & \dots & \dots & \dots \\ a_{n_y1}(q) & a_{n_y2}(q) & \dots & a_{n_y n_y}(q) \end{pmatrix} \quad (2.65)$$

The matrix defined in equation (2.65) has n_y rows and n_y columns. The values ($a_{kj}(q)$) of the matrix above could also be described as a polynomial. The polynomial is defined as [34]:

$$a_{mj}(q) = \delta(m-j) + a_{mj}^1 q^{-1} + \dots + a_{mj}^{n_{a_{mj}}} q^{-n_{a_{mj}}} \quad (2.66)$$

In equation (2.66) δ is the delta operator and has the following characteristic:

$$\delta(m-j) = 1 \text{ if } j = m$$

In the values $a_{mj}^{n_{a_{mj}}}$ in equation (2.66) are scalars that are calculated with the algorithm described in section 2.9.4.1 [6]. The characteristic equation for the ARMAX model structure could now be defined as [34]:

$$A(q)y(t) = B(q)u(t-nk) + C(q)e(t) \quad (2.67)$$

A block diagram representation of the ARMAX model structure is illustrated below [25]:



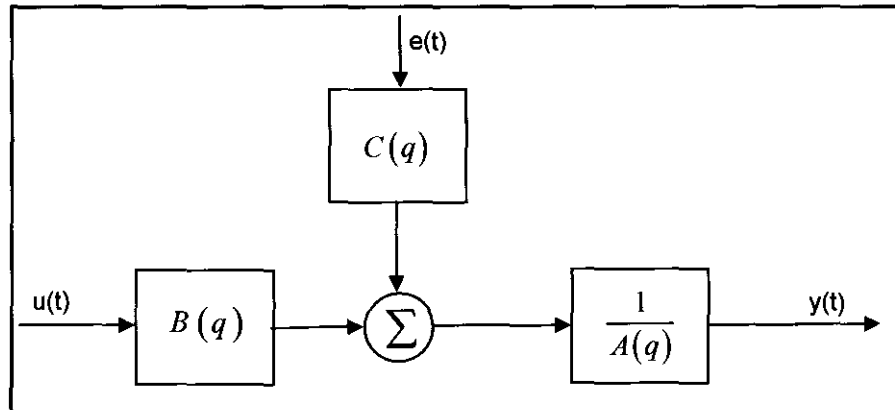


Figure 2-6: A block diagram that describes the ARMAX model structure [25].

2.9.4.1 The algorithm that is used to calculate the ARMAX model

The same algorithm that was described in the previous two sections that were used to calculate the BJ and OE models are used to calculate the ARMAX model. The difference was that a different search direction method is used to calculate the amount that the parameters are updated.

2.9.5 ARX model structure

When the order of polynomial $C(q)$ of the ARMAX model structure is equal to one, the characteristic model equation changes to [34]:

$$A(q)y(t) = B(q)u(t) + e(t) \quad (2.68)$$



This model structure is known as the ARX model structure. This model structure could be used to describe the model as a block diagram [25]:

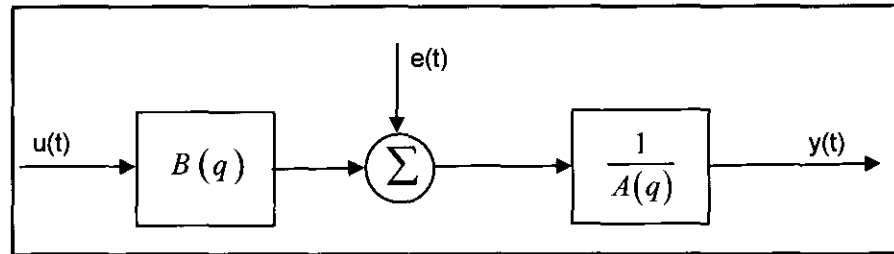


Figure 2-7: The ARX model structure [25].

2.9.5.1 The algorithm that is used to calculate the ARX model

The characteristic linear difference equation [6]:

$$y(t) + a_1 y(t-1) + \dots + a_{n_a} y(t-n_a) = b_1 u(t-1) + \dots + b_{n_b} u(t-n_b) + e(t) \quad (2.69)$$

The output could now be computed with the aid of the vectors defined in equation (2.26) [6]:

$$y(t) = \varphi^T(t) \theta \quad (2.70)$$

The adjustable variable in this equation is θ . This means that equation (2.70) is dependant on t and θ . The output is thus calculated with [6]:

$$\hat{y}(t|\theta) = \varphi^T(t) \theta \quad (2.71)$$

The least means squared method described in section 2.8.1 is used to calculate the parameters of θ . The implementation of this algorithm in Matlab© is described in section 4.2.1.



2.9.6 State Space model structure

The state space model structure assumes that the system could be described by the following equations [6]:

$$x(t+Ts) = Ax(t) + Bu(t) + Ke(t) \quad (2.72)$$

$$y(t) = Cx(t) + Du(t) + e(t) \quad (2.73)$$

In equation (2.72) – (2.73) $u(t)$ is the input, $y(t)$ the output, and $e(t)$ is a disturbance that was described in section 2.4. The State Space (SS) model structure is more often used in MIMO systems than the other models structures. The input is described by a column vector that has m rows. The output y is also a column vector that has p rows.

2.9.6.1 The algorithm that is used to calculate the State Space model

Equations (2.72) – (2.73) could also be described by the following equations [6]:

$$\tilde{x}(t+Ts) = T^{-1}AT\tilde{x}(t) + T^{-1}Bu(t) + \tilde{w}(t) \quad (2.74)$$

$$y(t) = CT\tilde{x}(t) + Du(t) + v(t) \quad (2.75)$$

where T is any invertible matrix, $\tilde{x}(t) = T^{-1}x(t)$, $\tilde{w}(t)$ and $v(t)$ are disturbances as described in section 2.4. The parameters that must be estimated are A , B , C and D . The estimation of A and C is described first.



A extended observable matrix could be defined as:

$$O_r = \begin{bmatrix} C \\ CA \\ \dots \\ CA^{r-1} \end{bmatrix} \quad (2.76)$$

This matrix could be obtained from the input-output data. Assume that the order (n) of the system is known. The observable matrix is now used to find the values of A and C :

$\hat{C}$ is equal to the rows one to p and columns 1 to n of the observable matrix.

$\hat{A}$ is equal to row $p+1$ to row pr and columns one to n of the observable matrix.

When the order of the system is not known Singular Value Decomposition (SVD) is used to reduce the dimensions of the observable matrix. SVD is described in more detail in [6] and [35].

The values of B and D could now be calculated with the aid of the values of A and C . The LMS algorithm that was described in section 2.8.1 is used to calculate the values of B and D . The matrix θ is defined as [6]:

$$\theta = \begin{bmatrix} \text{Vec}(B) \\ \text{Vec}(C) \end{bmatrix} \quad (2.77)$$

In (2.77) Vec is an operator that stacks the columns of the matrix on top of each other. The estimated value $\hat{y}(t)$ is calculated as [6]:

$$\hat{y}(t) = \varphi(t) \theta \quad (2.78)$$

The matrix φ has p rows and $mn+mp$ columns. Let the value of $r = (k-1)n + j$ where k is the amount of delays that it takes an input to have an effect on the output and n is the order of the model.



To estimate the r^{th} column of matrix φ equation (2.78) is differentiated. The r^{th} column of φ is now equal to [6]:

$$\varphi_r(t) = \hat{C} \left(qI - \hat{A} \right)^{-1} E_j u_k(t) \quad (2.79)$$

In equation (2.79) E_j is a column vector with the j^{th} element equal to one and the other element all equal to zero.

Equation (2.79) is now used with the LMS algorithm and evaluated for each time step. An over determined set of linear equations is formed and the solved. The implementation of this algorithm in Matlab© is described in section 4.2.3.

2.9.7 The advantages and disadvantages of the model structures

The different model structures give the models different characteristics. These characteristic make certain models structures more suited for certain applications than others. The advantages and disadvantages of the models structure are:

- The ARX model structure has the disadvantage that the process and the noise get the same poles ($A(q)$). This rarely happens in reality.
- The ARMAX models structure describes the noise better than the ARX model structure because the polynomial $C(q)$ models the noise, but the poles of the process is still the same as the poles of the noise.
- The OE model structure has the advantage that it models the dynamics of the system separately, but the disadvantage is that the noise is not modelled at all.



- The BJ model structure models the dynamics of the system separately. The noise is also modelled separately from the rest of the system. This structure models all aspects of the system separately.

2.10 Model errors

The models that are built could never exactly describe the system under investigation. In this section the different kinds of errors are described. In the field of system identification there are two types of critical errors. The first error is the bias error and the second is the variance error. The bias errors and their possible causes are described in section 2.10.1. The variance errors that occur and their possible causes are described in section 2.10.2 [21].

2.10.1 Bias-errors

Bias-errors occur due to the fact that real systems do not fit into the model structures described in the section 2.9 [22]. Real systems are more complicated than the models described in section 2.9. The parameters of the models are calculated to minimize the error between the model output and the desired output. This means that the best parameters are calculated for that specific model structure. Often this structure is incapable to describe the dynamics of the real system [21].



2.10.2 Variance errors

Variance errors occur due to the fact that the data contains disturbances and that only a limited amount of data is available. The effect of the bias- and variance errors is difficult to explain when both of them exist. In this section it is assumed that no bias-error exists. The covariance of the vector $\hat{\theta}_N$ (defined in section 2.8.1) can be described by [6]:

$$\text{Cov}\hat{\theta}_N = E\left(\hat{\theta}_N - \theta_0\right)\left(\hat{\theta}_N - \theta_0\right)^T \approx \frac{1}{N} \lambda \bar{R}^{-1} \quad (2.80)$$

where

$$\begin{aligned} \bar{R} &= E_{\psi}(t, \theta_0) \psi^T(t, \theta_0) \\ \psi(t, \theta_0) &= \frac{d}{d\theta} \hat{y}(t, \theta_0) \end{aligned} \quad (2.81)$$

In equation (2.81) vector θ_0 is the parameters that are chosen in such a way that the frequency function of the model $G(e^{j\omega}, \theta)$ is as close as possible to the true frequency function $G_0(e^{j\omega})$. In equation (2.80) λ is the noise intensity.

When equations (2.80) and (2.81) are evaluated, it is clear that when the number of data points increases the covariance of $\hat{\theta}_N$ will decrease. It is also very important to note that covariance decreases as the noise intensity (λ) increases [21].



3. The Pebble Bed Modular Reactor

A description of the Pebble Bed Modular Reactor (PBMR) along with its two main components is given, followed by a description of the three-shaft closed-loop recuperative Brayton cycle of the Pebble Bed Micro Model. A control overview of the Pebble Bed Micro Model with its subsystem is also given.



3.1 An introduction to the Pebble Bed Modular Reactor

Internationally, coal is currently the most widely used primary fuel, accounting for 36% of total fuel consumption for electricity generation. Coal is expected to maintain its share of the overall electricity generation market until 2020 [24].

In South Africa 90% of electricity is generated by coal-fired power stations. The other 10% is generated by hydro-electric (5%) and nuclear (5%) power stations. There are few, if any, new sites where hydro electric power stations could be built that will be economically viable. The country's natural gas resources are also too limited to qualify as a viable option for power generation in South Africa [24].

This led to the interest in power generation utilizing nuclear power. Currently the nuclear power station at Koeberg is the only nuclear power station on the African continent. South African power utility giant Eskom started investigating the PBMR in 1993 as part of their Integrated Electricity Planning process. After evaluation of the PBMR it was found to be a solution for alternative electricity generation.

3.2 How the PBMR works

The Pebble Bed Modular Reactor is a nuclear power station that is being developed by PBMR (Pty.) Ltd. and partners. The power plant is based on an idea by Prof. Dr. Rudolf Schulten. The idea was to use compact silicon



MISO system identification techniques using the PBMR data.

The Pebble Bed Modular Reactor

carbide-coated uranium granules into hard billiard-ball-like spheres to be used as fuel for a new high temperature, helium-cooled type reactor.

Prof. Dr. Rudolf Schulten's idea was implemented in the AVR and THTR demonstration pebble bed nuclear reactors that was built in Germany and was operational for 21 years until all nuclear research in Europe was halted after the Chernobyl disaster in 1986.

This accident placed great emphasis on the safety of nuclear reactors worldwide. These safety concerns favours the PBMR more than conventional water-cooled nuclear reactors, because the design, materials, fuel used and natural physics involved rather than active safety systems ensures safety in the PBMR plant.

The PBMR could be described as a high temperature, graphite-moderated, gas cooled reactor (HTGR) that uses a three-shaft recuperative Brayton cycle with helium as gas coolant [32]. As illustrated in Figure 3-1 the main power unit essentially consists of two units: the reactor and the power conversion unit. These units are described in detail in the following sections.



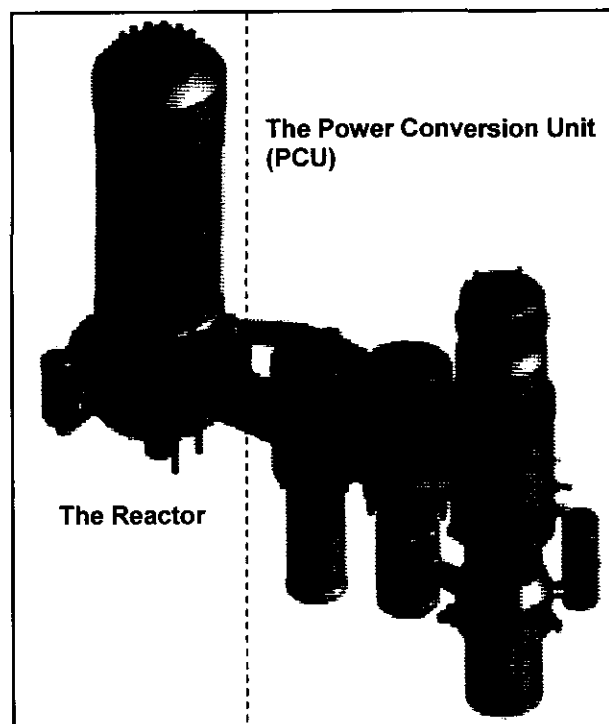


Figure 3-1: The two main units of the PBMR

3.2.1 The reactor

The reactor consists of an outer vertical steel pressure vessel. This steel vessel has a diameter of 6 m and is 20 m high. The pressure vessel houses the core of the reactor. The core is separated from the steel vessel by a layer of graphite bricks that is lined inside the vessel.

This graphite layer has two main functions. The first is to reflect neutrons generated by the fuel, and the second is to serve as a passive heat transfer medium. Vertical holes are drilled in the graphite layer. These holes house the control elements.



MISO system identification techniques using the PBMR data.

The Pebble Bed Modular Reactor

The core that is housed inside the vessel is the section of the reactor where the nuclear reaction takes place. The core of the PBMR has a diameter of 3,7m and is 9m high. The core has two main zones. The inner zone is filled with approximately 150 000 graphite fuel spheres, and the outer zone with approximately 380 000 fuel spheres. The nuclear reaction takes place in the outer layer.

The helium enters the core from above and flows over the graphite fuel spheres. The helium cools the fuel spheres. The helium absorbs some of the heat of the fuel and is thus heated. The helium then exits the core at the bottom.

3.2.2 The power conversion unit (PCU)

The PCU is based on a three shaft closed-loop recuperative Brayton cycle. A block diagram of the cycle is illustrated in Figure 3-2. This figure will now be used to describe the cycle in more detail.



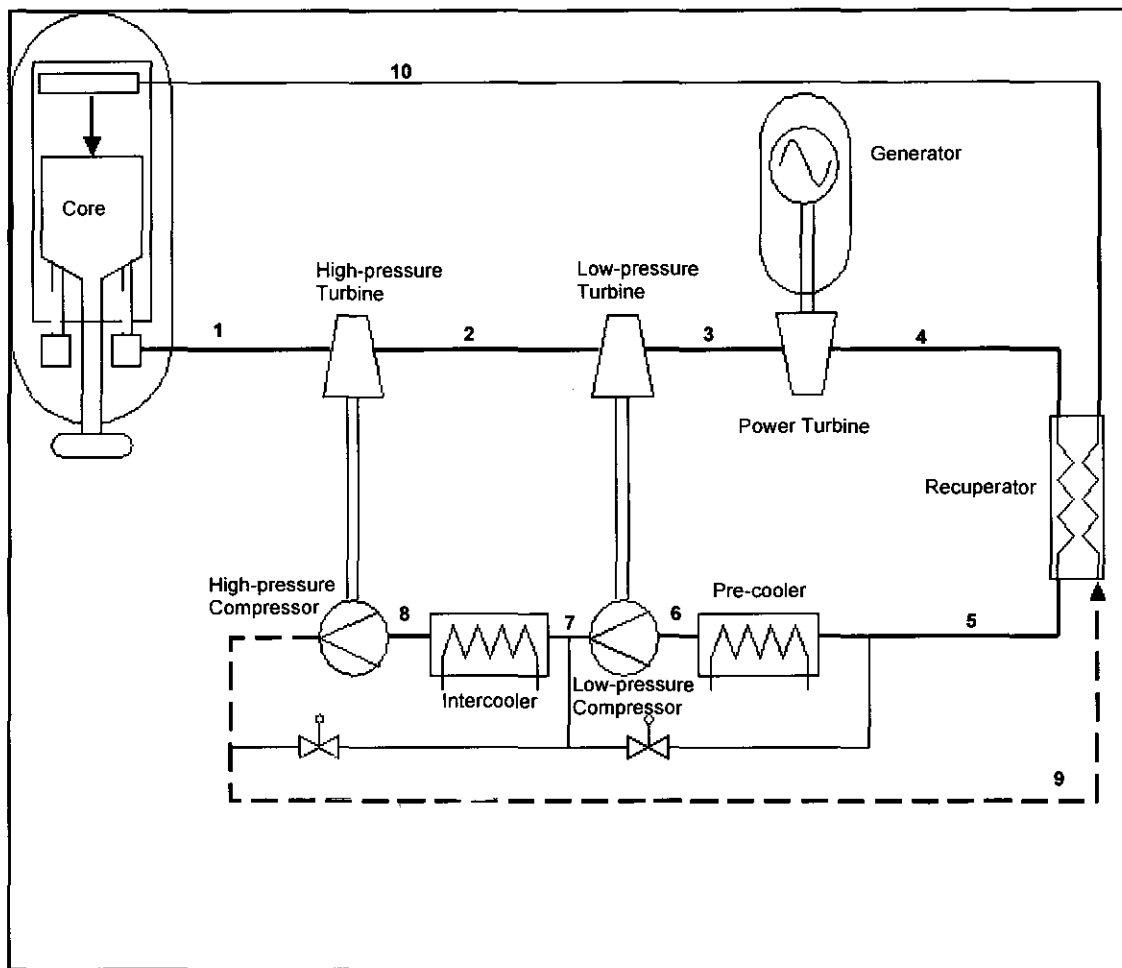


Figure 3-2: The Brayton cycle of the PBMR

The cycle is a closed-loop cycle and it is difficult to specify a starting point, but let's begin to describe the cycle at point 1. The helium exits the core at a temperature of 900°C .

The high-pressure helium gas is then expanded in a high-pressure turbine. This turbine drives the high-pressure compressor that will be encountered later in the cycle. The high-pressure turbine and the high-pressure compressor form a high-pressure turbo.



The gas is now at point two. The gas is then further expanded in the low-pressure turbine. This turbine drives a low-pressure compressor that is also encountered later in the cycle. Together the low-pressure turbine and the low-pressure compressor form a low-pressure turbo.

The gas is at point three in the cycle and then passes through the power turbine where it is further expanded. The turbine drives a generator that generates the electrical power. The gas exits the power turbine at point four.

At this stage the helium is at a temperature of 500°C and at a pressure of $2,6\text{MPa}$. The helium then passes over the primary side of the recuperator where it transfers heat to the secondary side of the recuperator (see second last step for secondary side). The helium is now at point five of the cycle.

The helium then passes through a pre-cooler. The helium is now at point six. After the helium is cooled down in the pre-cooler it enters the low-pressure compressor, that was mentioned earlier, which raises the pressure of the helium. The helium is now at point seven in the cycle.

The helium is then again cooled down, this time by an intercooler until it reaches point eight in the cycle. The helium is then further compressed by the high-pressure compressor and the gas exits the compressor at point nine of the cycle.

The helium then flows through the secondary side of the recuperator, which heats the gas to a temperature of 500°C . The helium is at a pressure of 7MPa when it enters the core at the top. The helium is then heated to 900°C by the fuel as it flows over the spheres. This concludes the description of the Brayton cycle used in the PBMR.



3.3 Control overview

The demand of electricity varies as people turn electrical appliances on and off. In the winter more electricity is used than in the summer. It is expensive to generate electricity; this means that ESKOM does not want to generate more electricity than necessary at a particular moment.

The PBMR was designed to be able to change its power output to match the demand. The power produced by the generator is proportional to the helium inventory in the system. This means the power can be changed by changing the inventory. The PBMR was designed with three methods to change the output of the system. The three methods are:

- The first method was designed to increase the power output of the system when the demand increases. This is achieved by increasing the inventory in the system by injecting helium into the system. This method is known as low-pressure injection and is discussed in more detail in 3.4.1.1.
- The second method was designed to decrease the output of the system. To decrease the output of the PBMR the inventory of helium must be decreased. This could be achieved by extracting helium from the system. This is known as high-pressure extraction and is discussed in more detail in 3.4.1.2.
- The first two processes are relatively slow. These two methods are thus used for load following. The third method is faster. This method that was designed to change the output of the generator is the adjustment of the bypass-valves. This method is designed for load rejection. The method is discussed in more detail in 3.4.1.3.



3.4 The MISO and SISO subsystems

3.4.1 The three SISO subsystems

3.4.1.1 Low-Pressure injection (LPI)

Helium is injected into the system; which increases the inventory of the system. This causes the generator (point 4 in Figure 3-3) to generate more power.

The helium is injected into the system just before the pre-cooler (illustrated in Figure 3-3 point 3). This is the point where the pressure is at its lowest in the cycle. This ensures that the least amount of energy is needed to increase the inventory in the system.

This method is viewed as the first SISO subsystem. The input of this subsystem is the rate of injection (kg/sec) and the output is the power output (kW).

3.4.1.2 High-Pressure extraction (HPE)

Helium is extracted from the system; which decreases the inventory of the system. This causes the generator to generate less power. This process is almost the direct opposite to the low-pressure injection.

The helium is extracted from the system after the high-pressure compressor (illustrated in Figure 3-3 point 2). This is the point where the pressure is at its highest in the cycle. This helps to ease the extraction.



This method is viewed as the second SISO subsystem. The input of this subsystem is the rate of extraction (kg/sec) and the output is the power output (kW).

3.4.1.3 Bypass valve regulation (BPV)

The affects of the HPE and LPI processes are slow. The BPV is used to follow a fast increase in power demand. The location of the bypass-valves is illustrated in Figure 3-3 at point 1. The first bypass valve is located across the pre-cooler and the low-pressure compressor (LPC) and is called the low-pressure bypass valve (LPBV). The second valve is located across the intercooler and the high-pressure compressor (HPC) and is called the high-pressure bypass valve (HPBV).

It is expected that these two valves will be operated in unison and therefore the two valves can be viewed as a single input to the system. The flow rate through the two bypass valves are assumed to be in the following proportion $1.5LPBV = HPBV$.

This method is viewed as the third SISO subsystem. The area of the pipe that the gas flows though increases exponentially. This will lead to a non-linear system when the diameter of the pipe is used as input to the system. Instead the area of the pipe is used as input (m^2) and the output of the system is the power generated (kW).



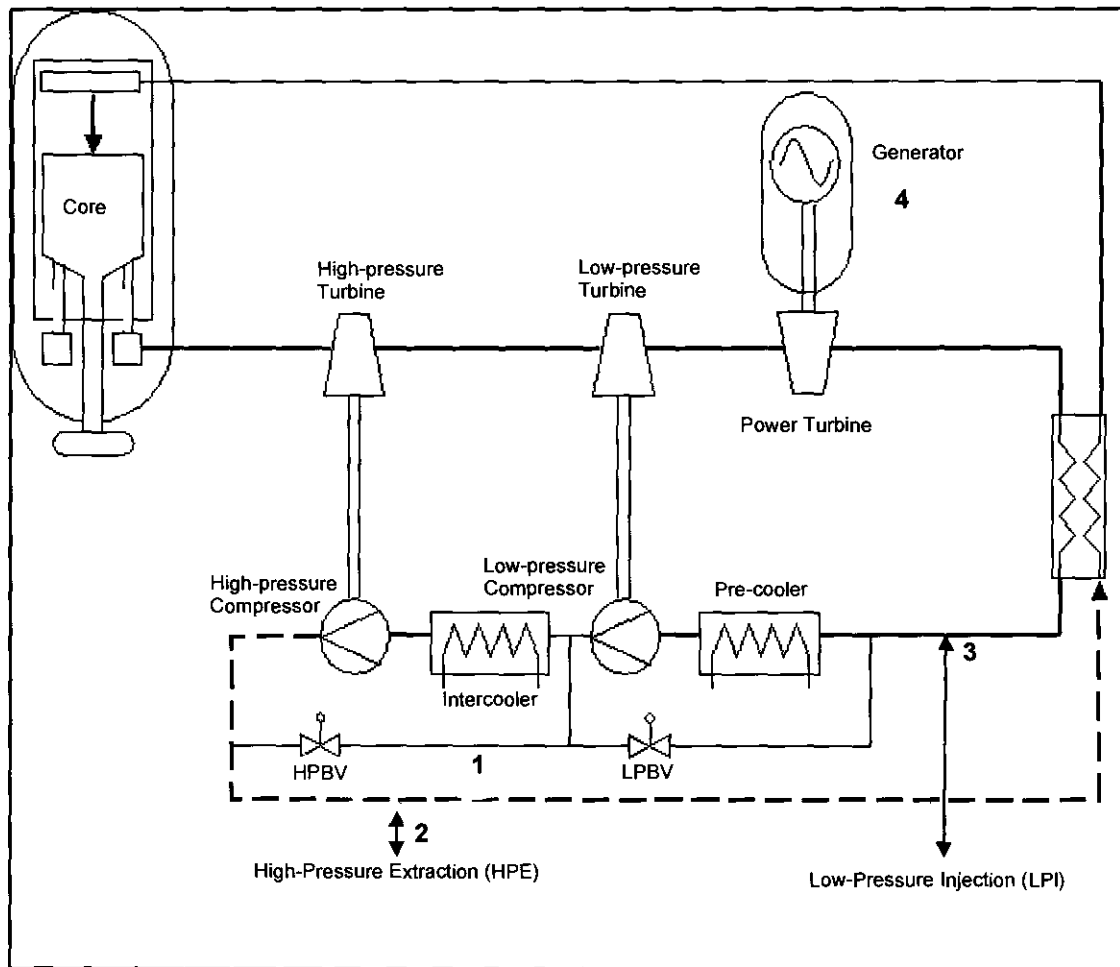


Figure 3-3: The location of the inputs of the subsystems in the PBMR.

3.4.2 The MISO subsystem

The previous section described three SISO subsystems. These subsystems were modelled individually by W.O. Erasmus [5]. The focus of this study is to use the same subsystems, but to use them to estimate a MISO model. This MISO system will combine the three SISO subsystems.



The inputs of the MISO system are: the rate of injection (LPI), the rate of extraction (HPE) and the opening of the bypass-valve (BPV). The output of this system is the power that is generated. The system is illustrated in Figure 3-4.

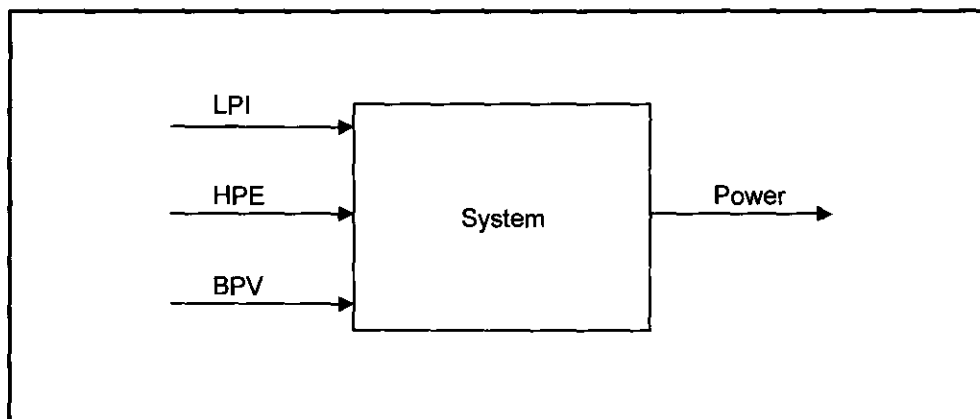


Figure 3-4: The MISO system that is modelled in this study.

3.5 The Pebble Bed Micro Model

The PBMR is still under development. A working model (Pebble Bed Micro Model or PBMM) of the PBMR was built at the University of Potchefstroom to test the idea of the high-temperature gas-cooled reactor (HTGR). The objective was to build a prototype of the PBMR that will have the same overall performance characteristics as the PBMR.

The PBMM is also based on the same three-shaft recuperative Brayton cycle. The PBMM is a conceptual model of the PBMR with the same control topology as the PBMR.

The differences between the PBMR and the PBMM are:



- In the PBMM nitrogen is used as gas coolant instead of helium as in the PBMR.
- In the PBMM an electrical heater is used as energy source instead of a nuclear reactor core in the PBMR.
- The PBMM uses off-the-shelf single stage centrifugal compressors and turbines rather than the axial flow machines that will be used in the PBMR.
- The generator is emulated by a load compressor that is connected to a power dissipation loop consisting of a flow control valve and a heat exchanger.

The PBMM was modelled with the aid of Flownex (explained in more detail in section 3.6) [32].

3.6 Flownex

Flownex is a general network analysis code package that solves the flow, pressure and temperature distribution in large arbitrary-structured thermal-fluid networks. The code can deal with both steady-state and transient flows. The code has been extensively validated against other codes and experimental data [32].

The data that is used in this study was generated with Flownex and the model of the PBMM.



MISO system identification techniques using the PBMR data.

System identification of the PBMR

4. System identification of the Pebble Bed Micro Model

In the following sections, the System Identification Toolbox in Matlab© is used to obtain models for the SISO and MISO subsystems



4.1 Introduction to MISO identification of PBMR

In this chapter the identification of the different models will be discussed. All the models were identified with the aid of the system identification toolbox of Matlab© version 6.5 release 13. The toolbox and the settings that were used are discussed in more detail in 4.2.

Firstly the three different SISO models are identified and a transfer function is obtained for the models. The structures that are used for identification are: ARX, ARMAX, Output Error, Box-Jenkins and the State Space model structures. These structures were discussed in 2.9. The identification of the SISO models was done by W.O. Erasmus [5]. This identification process is repeated in this study to compare the results of the SISO models with the MISO model. The difference between MISO and SISO models is then investigated.

4.2 The system identification toolbox in Matlab©

The System Identification Toolbox is for building accurate, simplified models of complex systems from noisy time-series data. It provides tools for creating linear mathematical models of dynamic systems based on observed input/output data. The identification techniques provided with this toolbox are useful for applications ranging from control system design and signal processing to time-series analysis and vibration analysis.

The toolbox estimates the parameters by adjusting parameters within a given model until its output coincides as well as possible with the measured output.



MISO system identification techniques using the PBMR data.

System identification of the PBMM

The implementation of the algorithms described in sections 2.9.2.1, 2.9.3.1, 2.9.4.1 and 2.9.5.1 is described in sections 4.2.1 and 4.2.2.

In this study the functions of the toolbox were used to identify the models. These results were verified afterward by using the graphical user interface.

The specification of the software that was used is:

- MATLAB® Version 6.5.0.180913a (R13)
- Microsoft Windows XP Version 5.1 (Build 2600)
- Java VM Version: Java 1.3.1_01 with Sun Microsystems Inc. Java HotSpot(TM) Client VM
- System Identification Toolbox Version 5.0.2 (R13)

4.2.1 The implementation of the ARX LMS algorithm

As mentioned in section 2.8.1 the Least Squared estimation is an over determined set of linear equations. A method known as QR factorization is then used to solve the set of linear equations. QR factorization uses a modified Gram-Schmidt iteration process. The regression matrix is formed so that only measured quantities are used (no fill-out zeros) [34].

The algorithm includes certain properties that could be specified or else the default value for the properties is used (recommended). The properties include [34]:

1. The property "Focus" has the following values:
 - 'Prediction' – This is the default value and means that the model is determined by minimizing the prediction error.
 - 'Simulation' – This means that the frequency weighing of the transfer function is given by the input spectrum.



- 'Stability' – The resulting model is guaranteed to be stable.
2. The property "InitialState": This is the value of the model at $t = 0$. There are three different settings that could be chosen. The first is "Estimate". This means that the first value is calculated. The next setting is "Zero". This setting is chosen when the output of the model at $t(0) = 0$. This property could also be set to "Backcast". This value means that the value is predicted backwards by a prediction algorithm. The default setting is "Auto". This setting lets the algorithm decide between the previous three settings.
 3. The property "Covariance": This property could be set to "Estimate" to estimate the covariance or to "None" in which case the covariance will not be calculated. The setting "None" saves a lot of computational time.

4.2.2 The implementation of the Prediction Error algorithm

As described in section 2.8.2 the Prediction Error method is an iterative process. The algorithm initiates the parameters of the model with the "n4sid" method. This method estimates models with the aid of the state space method and is described in section 2.9.6 [34].

The iterative search for the best parameters is then started. The next value of the model is calculated and compared to the desired value. The error between the desired output and the model output is calculated. This value is used with a parameter update algorithm (that is calculated with the aid of the gradient) to update the parameters [34].



The algorithm includes certain properties that could be specified or else the default value for the properties is used. The properties include [34]:

1. The property "Focus" has the following values:
 - 'Prediction' – This is the default value and means that the model is determined minimized by minimizing the prediction error.
 - 'Simulation' – This means that the frequency weighing of the transfer function is given by the input spectrum.
 - 'Stability' – The resulting model is guaranteed to be stable.
2. The property "MaxSize" is to specify the size of the matrixes that will be formed during identification. This is to save memory.
3. The property 'FixedParameter'. This property is a list of parameters that will be kept fixed and will not be estimated.
4. The property 'LimitError'. Variable determines how the iteration from the quadratic to one that gives linear weight to large errors. Errors larger than "LimitError" times the estimated standard deviation will carry a linear weight in the iterations. The default value of "LimitError" is 1,6. When the value of "LimitError" is equal to zero the robustification is disabled and a purely quadratic criterion is used.
5. The property 'MaxIter'. This property specifies the maximum amount of iterations that will be used to calculate the parameters.
6. The property 'Tolerance'. Based on the Gause-Newton vector computed at the current parameter value, an estimate is made of the expected improvement of the criterion at the next iteration. When the improvement is less than "Tolerance", measured at present, the iteration process is stopped. The default value is 0.01 and the value is specified as a percentage.
7. The property "SearchDirection". The property specifies the direction along which a line search will be performed to find the lower value of the search criterion. The property could have the following values:



- “gn”: The Gauss-Newton direction (inverse of Hessian times the gradient direction). If no improvement is found along this direction the gradient is also tried out.
 - “gns”: A regularized version of the Gauss-Newton direction. Eigenvalues less than 10^{-9} of the Hessian are neglected, and the Gauss-Newton direction is computed in the remaining subspace and used.
 - “lm”: The Levenberg-Marquard method is used. This means that the next parameter value is $-psuedoinverse(H + d * I) * gradient$ from the previous one. H is the Hessian matrix. I the identity matrix and d is a number that is increased until a lower value of the criterion is found.
 - “Auto”: A choice between the above three methods is made in the algorithm. This is the default choice.
8. The property “InitialState”: This is the value of the model at $t = 0$. There are three different settings that could be chosen. The first is “Estimate”. This means that the first value is calculated. The next setting is “Zero”. This setting is chosen when the output of the model at $t(0) = 0$. This property could also be set to “Backcast”. This value means that the value is predicted backwards by a prediction algorithm. The default setting is “Auto”. This setting gives the algorithm freedom to decide between the previous three settings.
9. The property “Covariance”: This property could be set to “Estimate” to estimate the covariance or to “None” in which case the covariance will not be calculated. The setting “None” saves a lot of computational time.



4.2.3 The implementation of the State Space (n4sid) algorithm

As described in section 2.9.6.1 a set of over determined linear equations is formed. A method known as QR factorization is then used to solve the set of linear equations. QR factorization uses a modified Gram-Schmidt iteration process. The regression matrix is formed so that only measured quantities are used (no fill-out zeros) [34].

The algorithm includes certain properties that could be specified or else the default value for the properties is used (recommended). The properties include [34]:

1. The property "Focus" has the following values:
 - 'Prediction' – This is the default value and means that the model is determined by minimizing the prediction error.
 - 'Simulation' – This means that then frequency weighing of the transfer function is given by the input spectrum.
 - 'Stability' – The resulting model is guaranteed to be stable.
2. The property "CovarianceMatrix" is to specify if the covariance matrix must be calculated in the model estimation. When the property is set to "None" the calculations of uncertainty measurements is blocked. The default value is "co", that means calculate the matrix.
3. The property 'DisturbanceModel'. Setting this property to "None" will deliver a model with $k = 0$. This will have no direct effects on the dynamics of the model, but the default choice of "N4Horizon" will involve past outputs.
4. The property 'N4Weight'. This property determines some weighing matrixes used in the SVD that is a central step in the algorithm. When the property could be set to "MOESP" the MOESP algorithm by Verhaegen is used. When the property could be set to "CVA" the canonical variable algorithm by Larimore is used. The default value is



“Auto” which means that in the algorithm chooses between the two algorithms.

5. The property ‘N4Horizon’. This property determines the prediction horizons forward and backward that is used by the algorithm. This is a row vector with three elements. The first element (r) is the maximum amount of forward prediction horizon. The second element (su) is the amount of past inputs that is used for predictors. The last element (sy) is the amount of past outputs that is used in prediction.
6. The property “InitialState”: This is the value of the model at $t = 0$. There are three different settings that could be chosen. The first is “Estimate”. This means that the first value is calculated. The next setting is “Zero”. This setting is chosen when the output of the model at $t(0) = 0$. This property could also be set to “Backcast”. This setting means that the value is predicted backwards by a prediction algorithm. The default setting is “Auto”. This setting gives the algorithm freedom to decide between the previous three settings.
7. The property “Covariance”: This property could be set to “Estimate” to estimate the covariance or to “None” in which case the covariance will not be calculated. The setting “None” saves a lot of computational time.

4.3 The experimental setup

As mentioned previously the PBMR is still under development. The data that was used in this study was generated by using the M400 model of the PBMM in Flownex.



As mentioned previously the PBMR and the PBMM for that matter are a non-linear system. This could pose a serious problem for the models that is used in this study, because the models could only describe linear systems. There are several techniques to overcome this problem. One method is to operate the system inside certain boundaries. These boundaries could be chosen in such a manner that the system is almost linear inside these boundaries. In another study a system was broken up into different phases of operation. Each of these phases was identified [31].

The non-linear problem is overcome in this study by only operating the PBMM in the normal operation mode. This mode is between 40 and 100% of maximum continuous rating. The maximum continuous rating of the PBMM is 50kW .

Another step that was done to ensure that the output is only dependant on the input that was given was to simulate the system for a few seconds without receiving an input. This is done to make sure that output is stable before an input is given to the system. This ensures that the response to the input is a result of only the input that was received.

The system that was used to record the data set did not contain any feedback. This simplifies the identification process.

There are two other issues that must be addressed before the simulation. The first is the choice of the input signal and the second is the sampling interval. These issues are discussed in the sections 4.3.1 and 4.3.2.

The next step in system identification according to Figure 2-2 is the gathering of data. The raw data was generated by Flownex. The data was then imported to Matlab®.



The first step after data was gathered is to pre-process the data in order to make it more suited for identification. The pre-processing of the data is discussed in section 4.3.3.

After pre-processing the data is ready for identification. The identification and the results are described in section 4.4 and 4.5. The interpretation of the results and the validation is discussed in section 4.3.4 [25].

4.3.1 The choice of an input signal

The choice of an input signal is a very important step in the identification procedure. The input signal must have large frequency content in order to obtain the response of the system at almost every frequency [6].

The input signal must also vary between different levels, because this exposes any non-linearity's that the system may contain. The signal must vary between at least three levels to expose non-linearity's [6].

In the time domain the input signal must be chosen so that the smallest time constraints appear in the data [25]. It is also very important to have such constant intervals so that the slow frequencies appear in the output [26].

In the end it was very difficult to choose an input signal for this system, because the physical system could only receive positive inputs in some cases (LPI, BPV) and only negative inputs in others (HPE). The choice of an input was further complicated by the fact that the PBMM system could not operate outside the normal power mode operation. This means that less gas could be



extracted at a time because this would cause the system to operate outside the preferred power mode.

4.3.2 The sampling interval

Sampling leads to loss in information [25]. This is the reason why the choice of sampling interval is very important. It is better to sample too fast rather than too slow, because if the sampling interval is too large critical information is lost [25].

The disadvantage when sampling is very fast is that the extra data does not give new information and numerical problems may arise in the identification process. When the sampling interval is too large it would be very difficult for the identification algorithms to calculate the best parameters [25].

In this study the sampling interval was chosen as $3,33Hz$, because this was the highest sampling frequency that Flownex could compute without producing errors.

4.3.3 The pre-processing of data

After the data is gathered it must first be examined and pre-processed if necessary. The pre-processing of data is important, because if the data is not pre-processed correctly the identification of a model could fail completely.

The data that is recorded must be examined for the following characteristics [6]:

- Offsets and drifts



- Outliners and missing data

When a system is operated at a certain operating point (like the PBMM for example) the data will have a certain offset. The first way to deal with this problem is to subtract the means from all the data. The second way to deal with this problem is to calculate the offset and subtract this value from the data.

The PBMM started its simulation at a stable state. This state is $36,6kW$. This value was subtracted from the data.

In some cases the data may contain outliners. The causes of outliners are not always known, but it may be caused for example by faulty sensors. Outliners often cause poor model identification.

The data is sampled and recorded. In this process many problems may occur. A problem that also leads to poor identification is when data is missing. This problem could be overcome by interpolating between data points [25].

4.3.4 Results and their interpretation

The model that is identified will be used in a control system. The lower the order of the model the easier it is to implement a control system. The fact that the models will be used in a control system restricted the identification process to second and third order models only.

The main result that is obtained from the identification is the transfer function. This transfer function for each model will be presented in the discrete time.



The results will also include the output of the diagnostics. These diagnostics are used to see if the model that is obtained is of high-quality or not. The diagnostics include: model output and residual analysis.

The model output is given as a percentage fit. The percentage is calculated by the following equation [34]:

$$Fit = 100 * \left(\frac{1 - norm(yh - y)}{norm(y - mean(y))} \right) \quad (4.1)$$

where yh and y are vectors. The vector y is the output vector of the system. The vector yh is the simulated output vector of the model that was calculated with the same input that produced the output vector y . In the equation $norm(x)$ is the Euclidean length of a vector x , and $mean(x)$ is the average of vector x . The two vectors yh and y is also plotted against each other.

The residual analysis is the last diagnostics that is used to decide if the model that was estimated is satisfactory or not. The residual analysis is two fold. The first residual analysis is the auto-correlation of the error, where the error is defined as [34]:

$$\varepsilon(t) = y(t) - \hat{y}(t) \quad (4.2)$$

In equation (4.2) $y(t)$ is the desired output and $\hat{y}(t)$ is the output of the model.

The auto-correlation function is mathematically defined as [34]:

$$\hat{R}_\varepsilon = \frac{1}{N} \sum_{t=\tau}^N \varepsilon(t) \varepsilon(t - \tau) \quad (4.3)$$

The auto-correlation should be a normally distributed, white noise process with zero means that is uncorrelated with the past data.



MISO system identification techniques using the PBMR data.

System identification of the PBMM

The second residual analysis is the evaluation of the cross-correlation function. The cross-correlation function determines the correlation between the input and the error function ($\varepsilon(t)$) of the estimated model. The input and the error functions should be uncorrelated. The auto- and cross-correlation functions are both presented as figures.

A confidence interval of 95% is defined to help determine if the residuals are “small enough”. The confidence interval is also presented in the figures of the auto- and cross-correlation functions. The residuals are small enough if all the residuals (except the residual at zero lag of the auto-correlation function) lie within this confidence interval. Then the residuals are said to be Gaussian and all residuals fall within a sigma of three [12].

4.4 SISO model identification

4.4.1 Low-Pressure injection system identification

4.4.1.1 Low-Pressure injection simulation

In section 4.3.1 the problem of the choice of input signal was mentioned. After considering the constraints of the physical system and the criterion that the input signal must meet the input signal that was used is illustrated in Figure 4-1.



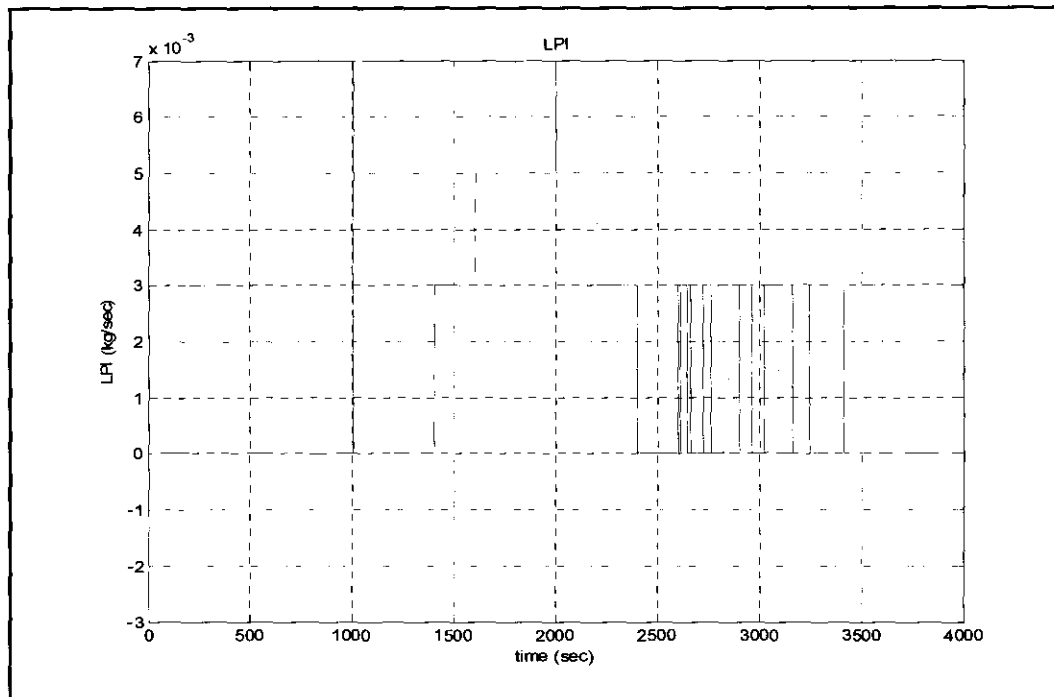


Figure 4-1: The input signal of the LPI SISO model

The best input signal must be positive and negative, but this is not possible for this system as the physical system does not allow a negative value for LPI. The output that was recorded is illustrated in Figure 4-2.



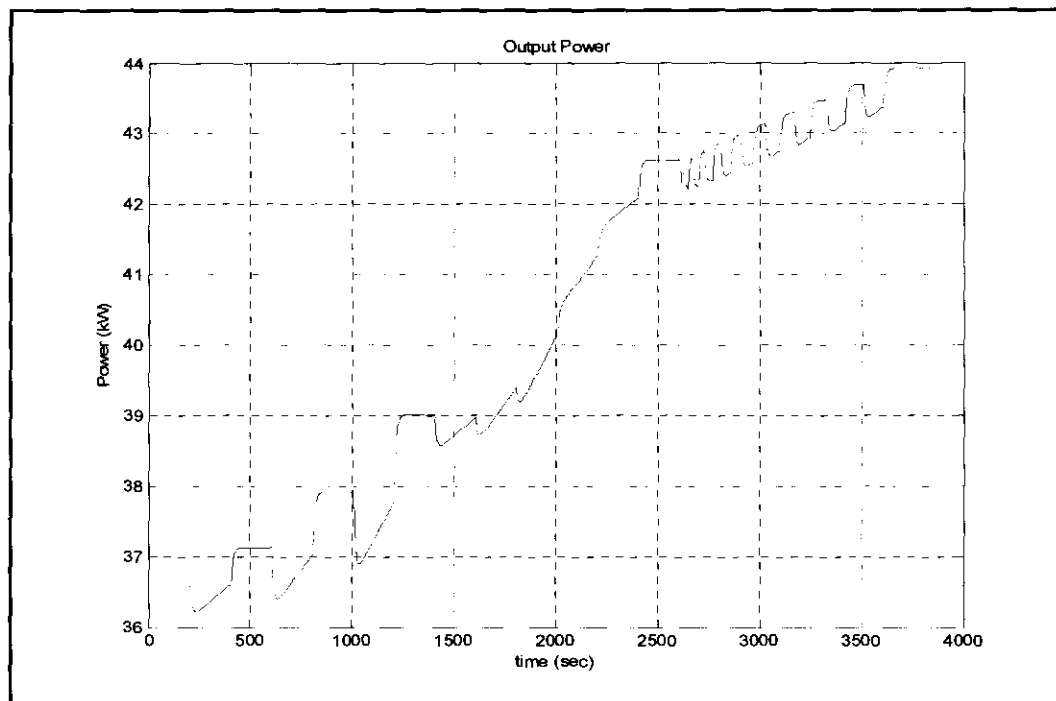


Figure 4-2: The output of the LPI SISO system

In Figure 4-2 it is clear that the system was always inside the parameters for normal power mode operation. In Figure 4-2 the offset of 36.6 kW is obvious. This offset must be subtracted to make the data suited for identification. The output that is then obtained after the offset is subtracted is illustrated in Figure 4-3.



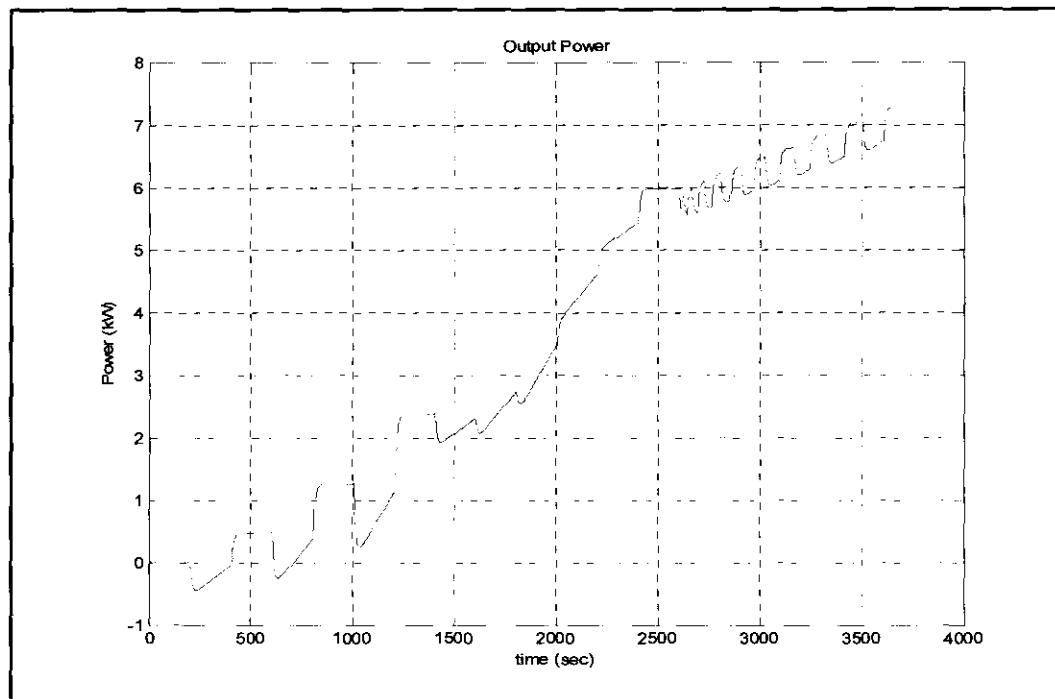


Figure 4-3: The output of the LPI SISO system without offset

There are no other problems with the input or output data. The data described in Figure 4-1 and Figure 4-3 is used in the following sections to identify different models.

4.4.1.2 ARX model identification

As described previously the ARX model structure is estimated with the aid of the LMS algorithm. The following property settings were used in Matlab©:

- Initial condition – Zero
- Focus – Simulation
- Covariance – Estimate

The first step is to calculate the value of k . This is the amount of delays it takes an input to have an effect on the output. When the data is observed it is



clear that $k=1$ for this input, because the input occurs at $t=x$ its effect is seen at $t=x+Ts$. Ts is the period of one sample. The value of k will be used in the whole section that the LPI process is identified. The results of the identification of different second and third order models are listed in Table 4-1.

Table 4-1

Name	n_a	n_b	Fit
ARX331	3	3	99.82%
ARX321	3	2	99.73%
ARX231	2	3	99.59%
ARX221	2	2	99.19%

In this study it is essential that the lowest order model that describe the system accurately is estimated. The first model that is evaluated is the ARX221 model. The simulated output of the ARX221 is illustrated in Figure 4-4. The simulated output of the ARX221 model is the red line and the desired (measured) output is the black line in the figure.



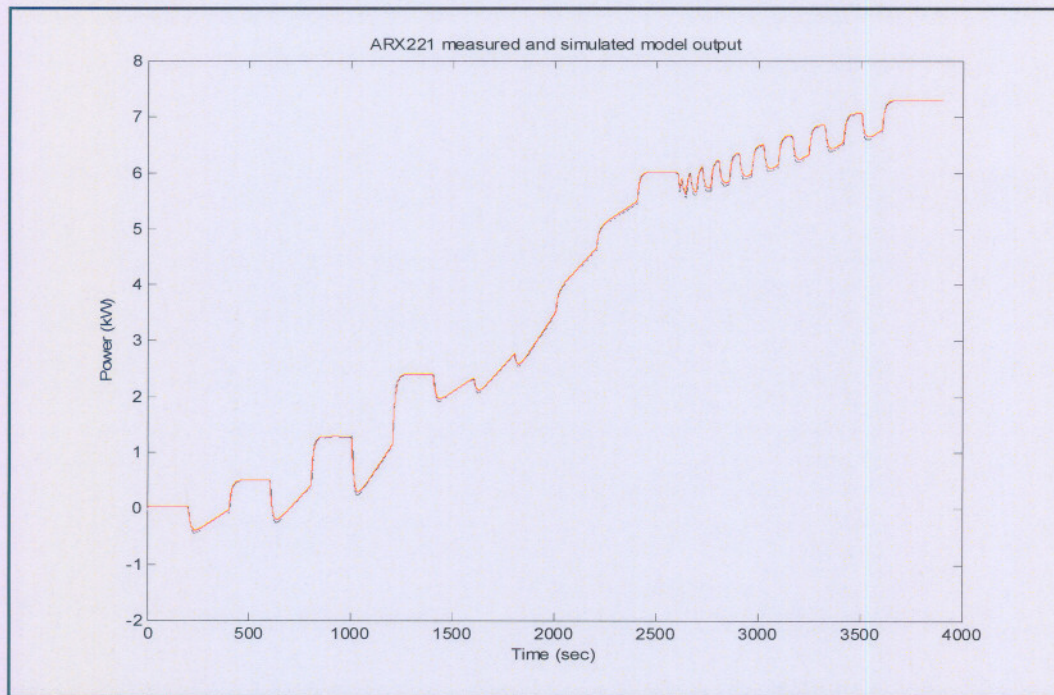


Figure 4-4: The simulated output of the ARX221 model (red) and the measured output

When the simulated output of the ARX221 model is evaluated very critically a more descriptive figure than Figure 4-4 is needed. When a certain area of Figure 4-4 is isolated and evaluated it is clear that the system is under modelled and the dynamics of the system are not described.

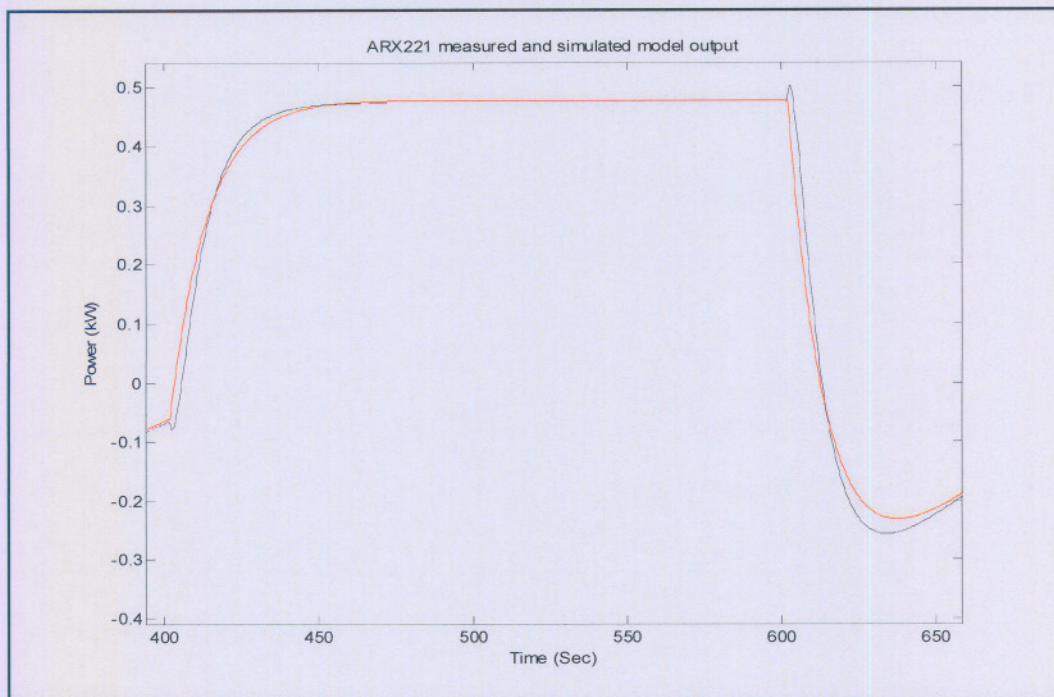


Figure 4-5: A more descriptive version of Figure 4-4

The residual analysis of the ARX221 model is illustrated in Figure 4-6. The residual analysis reveals that the ARX221 model has a good auto- or cross-correlation.

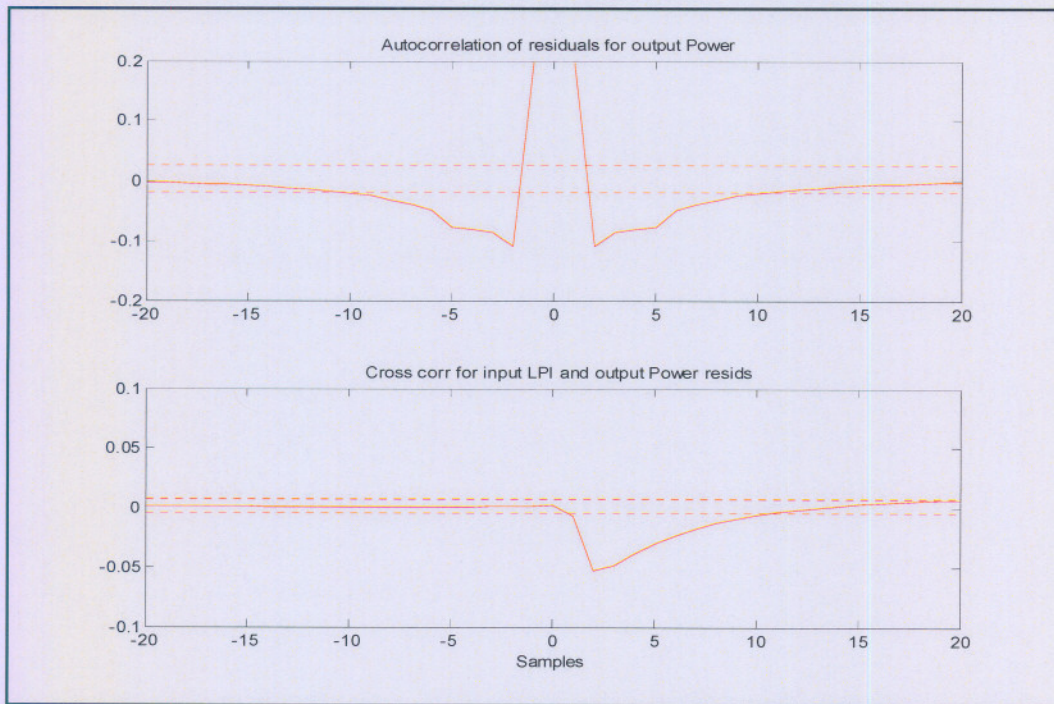


Figure 4-6: The residual analysis of the ARX221 LPI model

The descriptive polynomial of the ARX221 LPI model is:

$$(1 - 1.976q^{-1} + 0.9757q^{-2})y(t) = (-4.132q^{-1} + 4.138q^{-2})u(t) + e(t)$$

The fact that the ARX221 model fails to describe all the dynamical behaviour of the system led to the evaluation of the ARX331 model. Both the ARX231 and ARX231 models fail to describe the system accurately.

The simulated output (red line) of the ARX331 model and the measured output of the system (black line) are illustrated in Figure 4-7. A more descriptive version of Figure 4-7 is illustrated in Figure 4-8.



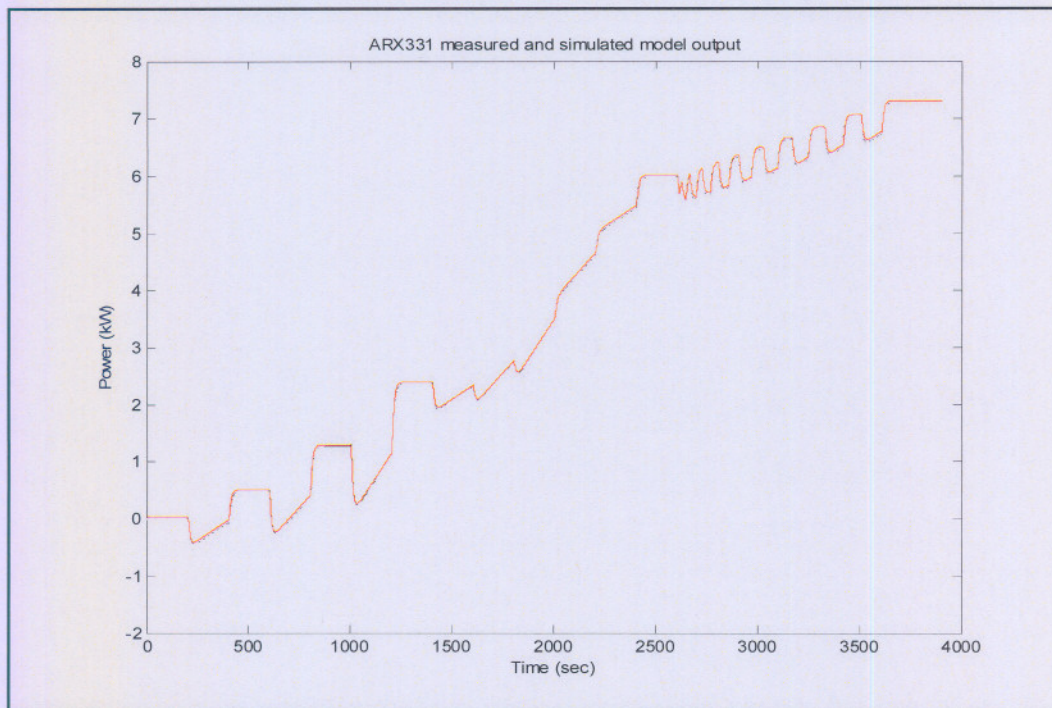


Figure 4-7: The simulated output of the ARX331 (red) system and the measured output

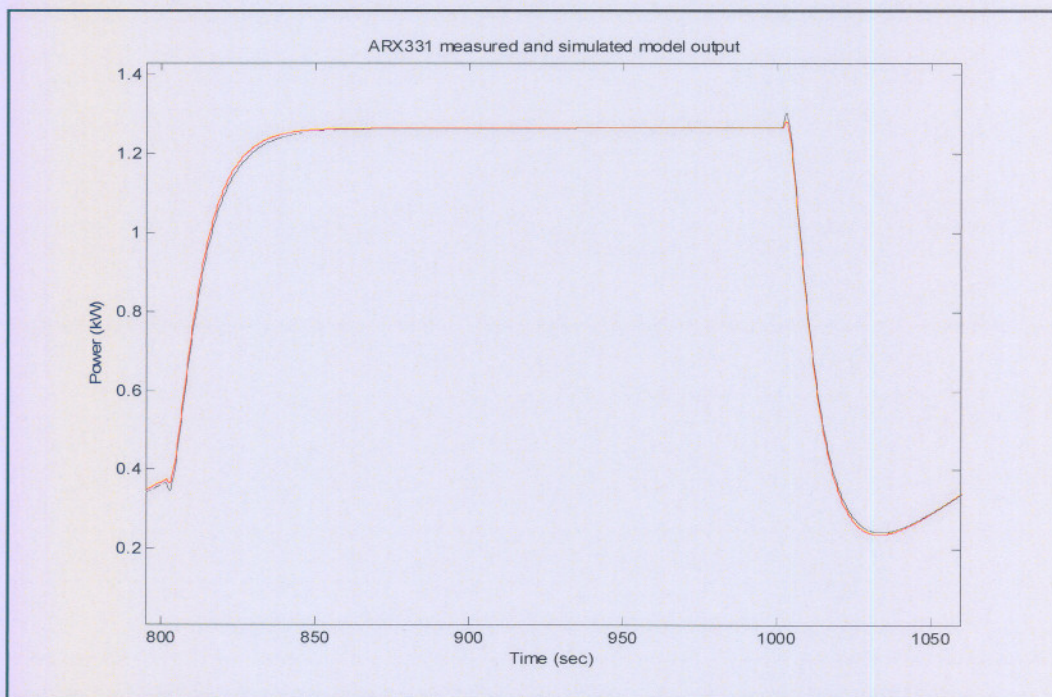


Figure 4-8: A more descriptive version of Figure 4-7.



In Figure 4-8 it is clear that the ARX331 model captures the dynamical behaviour very well. The residual analysis of the ARX331 model is illustrated in Figure 4-9.

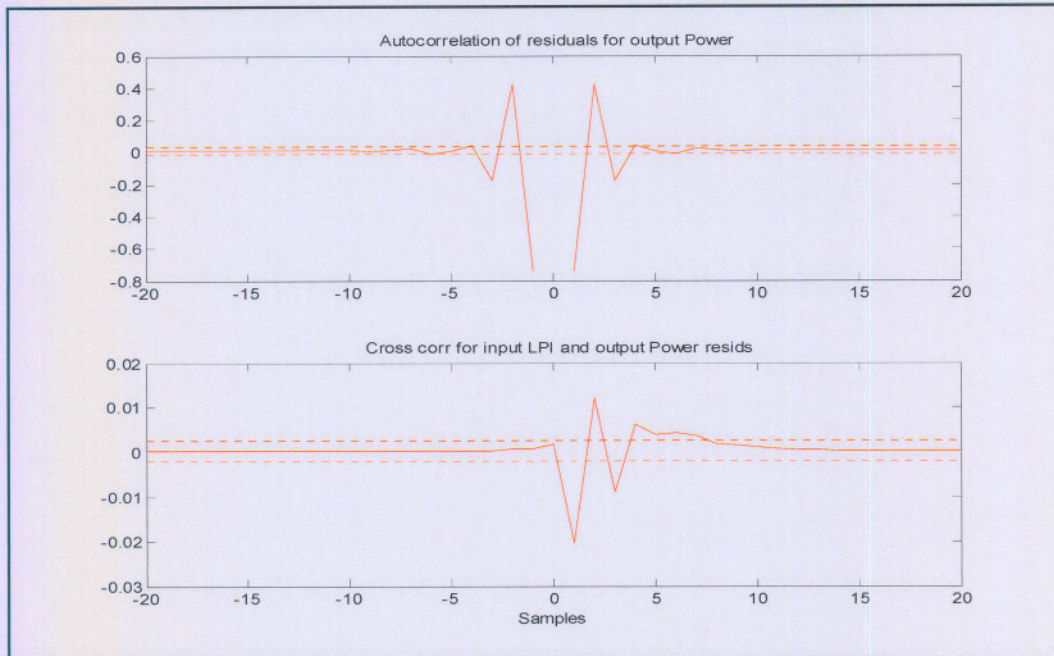


Figure 4-9: The residual analysis of the ARX331 model

The residual analysis of the ARX331 model reveals that the auto- and cross-correlation are better than the ARX221 models.

The ARX331 model is described by the following equation:

$$(1 - 2.853q^{-1} + 2.712q^{-2} - 0.8579q^{-3})y(t) = (1.68q^{-1} - 4.024q^{-2} + 2.344q^{-3})u(t) + e(t)$$

4.4.1.3 ARMAX model identification

The Prediction Error algorithm is used to estimate the parameters of the ARMAX model structure.

The following property settings were used in Matlab©:

- Initial condition – Zero
- Focus – Simulation
- Covariance – Estimate

The default settings were used for all the other properties.

The result of all the second and third order models that were calculated are listed in Table 4-2.

Table 4-2

Name	n_a	n_b	n_c	Fit
AMX3331	3	3	3	99.82%
AMX3321	3	3	2	99.82%
AMX2321	2	3	2	99.60%
AMX2331	2	3	3	99.60%
AMX2221	2	2	2	99.19%
AMX2231	2	2	3	99.19%
AMX3221	3	2	2	-50.63%
AMX3231	3	2	3	-50.83%

According to Table 4-2 the best model is the AMX3331 model. This model will be evaluated later in the section. The first model that produced a reasonably good fit is the AMX2221 model. The simulated output of the model is illustrated in Figure 4-10 with the measured output of the system. A more descriptive version of Figure 4-10 is illustrated in Figure 4-11.



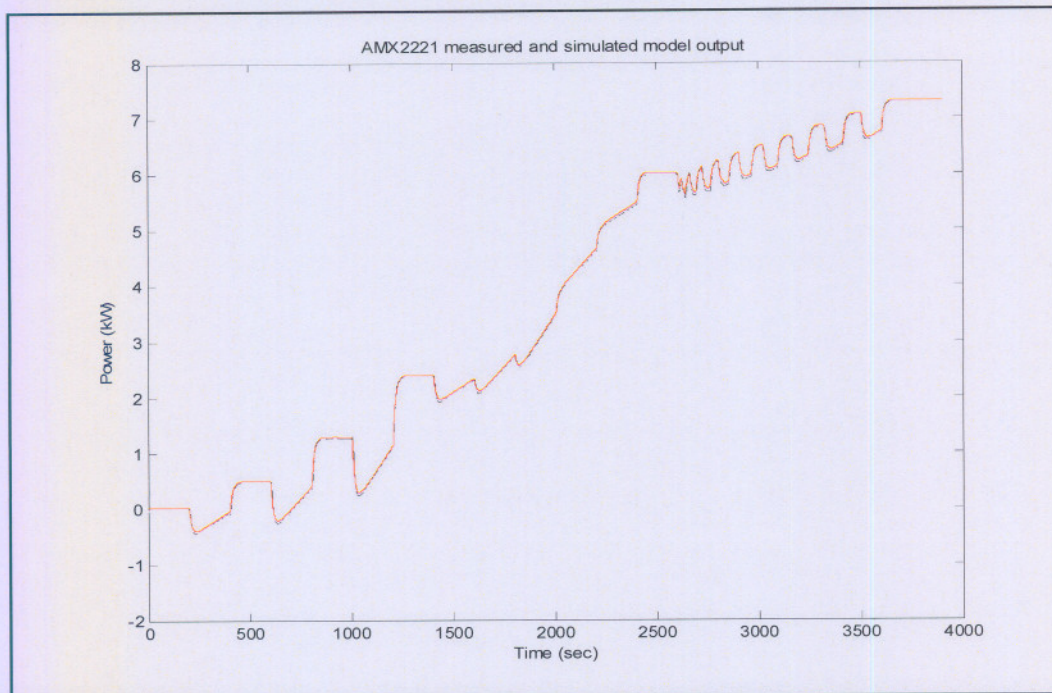


Figure 4-10: The simulated output of the AMX2221 model (red) and the measured output

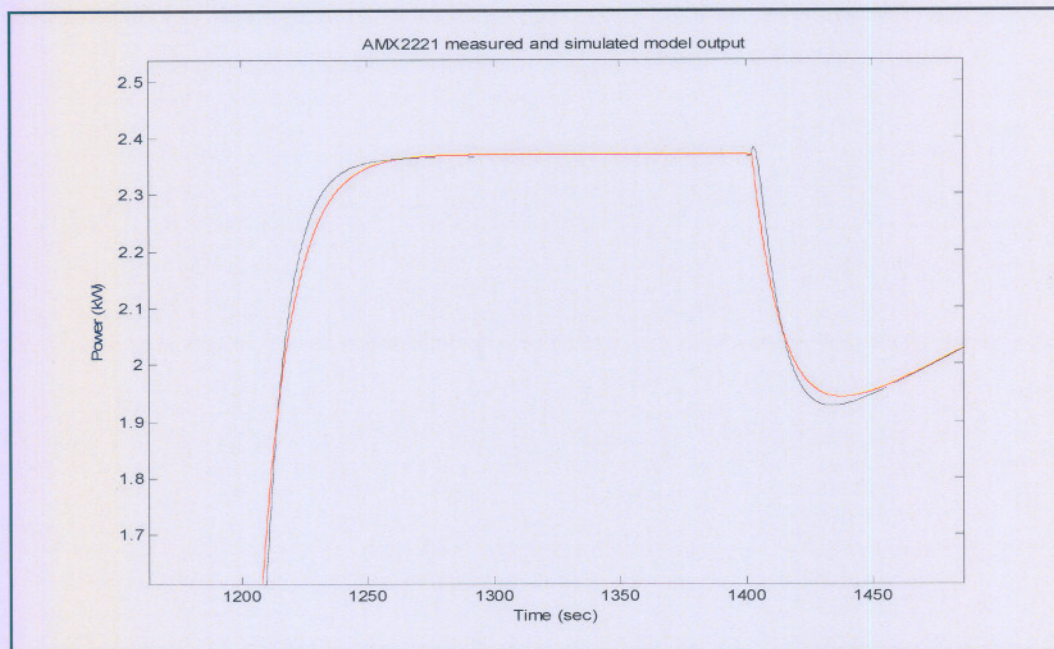


Figure 4-11: A more descriptive version of Figure 4-10

The residual analysis of the AMX2221 model is illustrated in Figure 4-12.



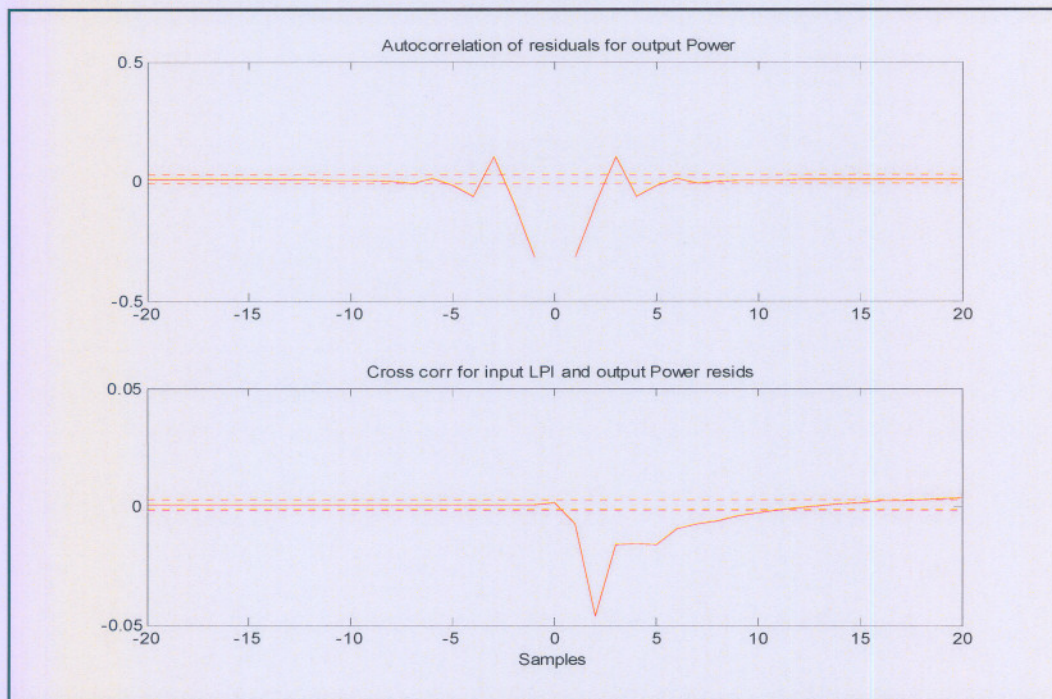


Figure 4-12: The residual analysis of the AMX2221 model

The residual analysis of the AMX2221 model reveals that that the model has a good auto-correlation function, but the cross-correlation is not that good. The cross-correlation has a large spike at lag 2. The calculated equation of the AMX2221 model is:

$$(1 - 1.976q^{-1} + 0.9755q^{-2})y(t) = (-4.168q^{-1} + 4.174q^{-2})u(t) + (1 + 0.622q^{-1} + 0.2361q^{-2})e(t)$$

The best model according to the fit criteria is the AMX3331 model. The simulated output of the model and the measured output of the system are illustrated in Figure 4-13. A more descriptive version of Figure 4-13 is illustrated in Figure 4-14.



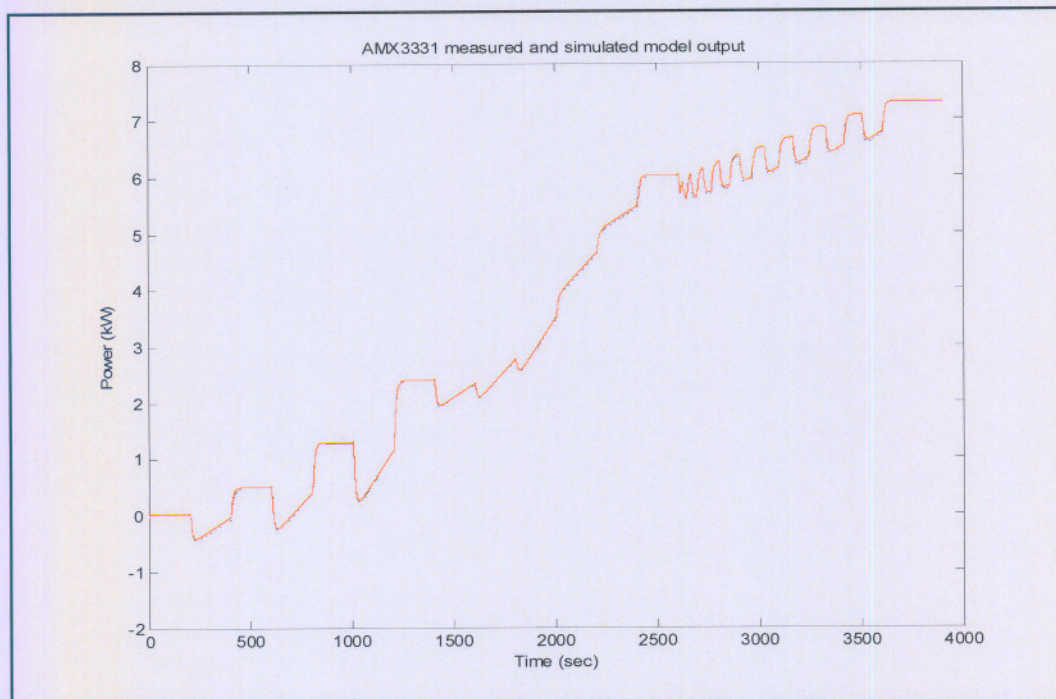


Figure 4-13: The simulated output of the AMX3331 model (red) with the measured output

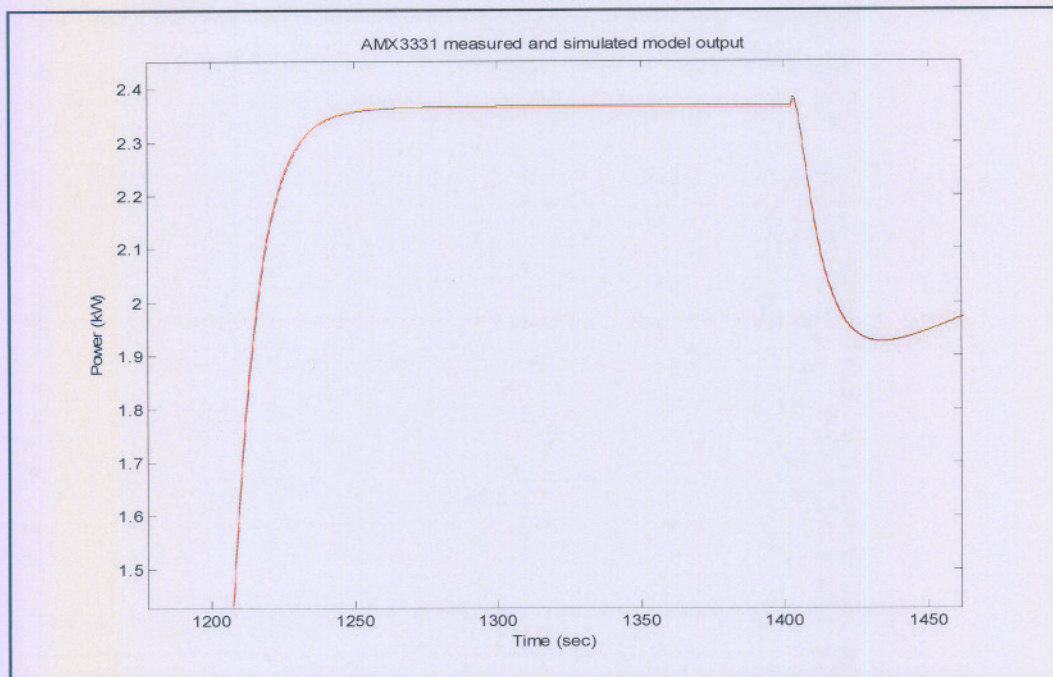


Figure 4-14: A more descriptive version of Figure 4-13



MISO system identification techniques using the PBMR data.

System identification of the PBMM

In Figure 4-14 it is clear that the AMX3331 model describes a lot more of the dynamical behaviour of the system than the AMX2221 model. The residual analysis is illustrated in Figure 4-15.

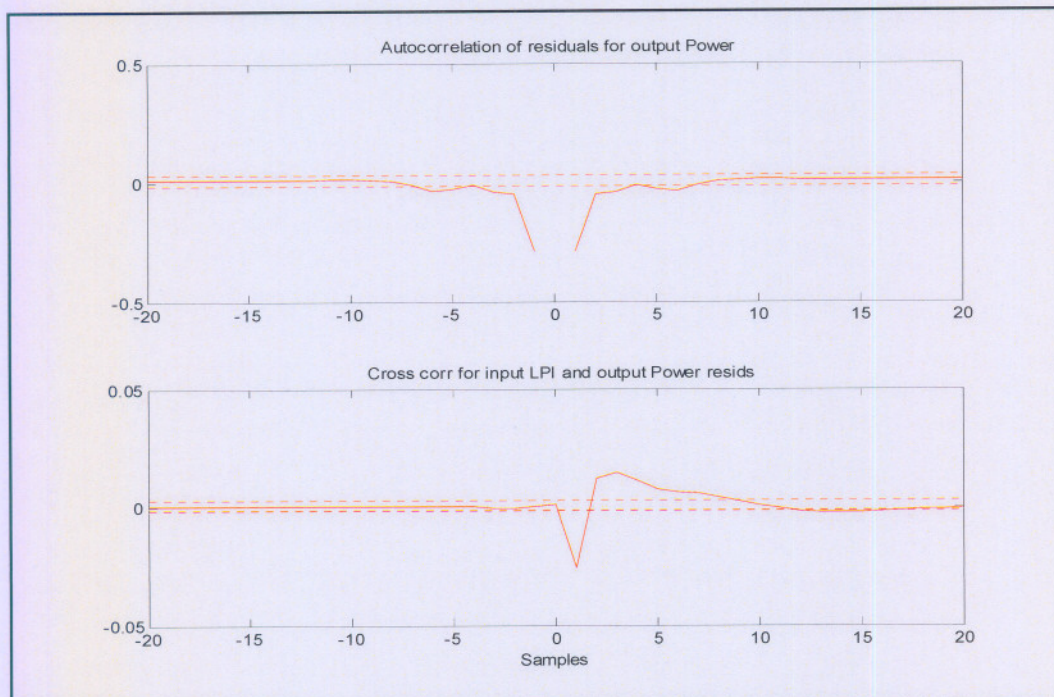


Figure 4-15: The residual analysis of the AMX3331 model

When the residual analysis of the AMX3331 model (Figure 4-15) are compared to the residual analysis of the AMX2221 model (Figure 4-12) the AMX3331 model has superior cross-correlation, but the AMX2221 model is a better model according to the auto-correlation functions.

The parameters of the AMX3331 model were calculated as:

$$\begin{aligned} (1 - 2.826q^{-1} + 2.657q^{-2} - 0.8307q^{-3})y(t) &= (2.807q^{-1} - 6.418q^{-2} + 3.612q^{-3})u(t) + \\ (1 - 1.296q^{-1} + 0.6039q^{-2} - 0.09614q^{-3})e(t) \end{aligned}$$



4.4.1.4 Output Error model identification

The Prediction Error algorithm is used to estimate the parameters of the OE model structure. The algorithm and its properties were described in section 4.2.2.

The following property settings were used in Matlab©:

- Initial condition – Zero
- Focus – Simulation
- Covariance – Estimate

The default settings were used for all the other properties. Various second and third order models were calculated. Table 4-3 is a list of all the second and third order models that were calculated and their fit.

Table 4-3

Name	n_b	n_f	Fit
OE331	3	3	99.82%
OE321	2	2	99.60%
OE221	2	2	99.19%
OE231	2	3	-49.48%

The first model that will be evaluated is the OE221 model. This model has a good fit and very low order. The simulated output (which is the same as infinite steps ahead prediction [5]) of the OE221 model and the measured output of the system is illustrated in Figure 4-16. The more descriptive version of Figure 4-16 is illustrated in Figure 4-17.



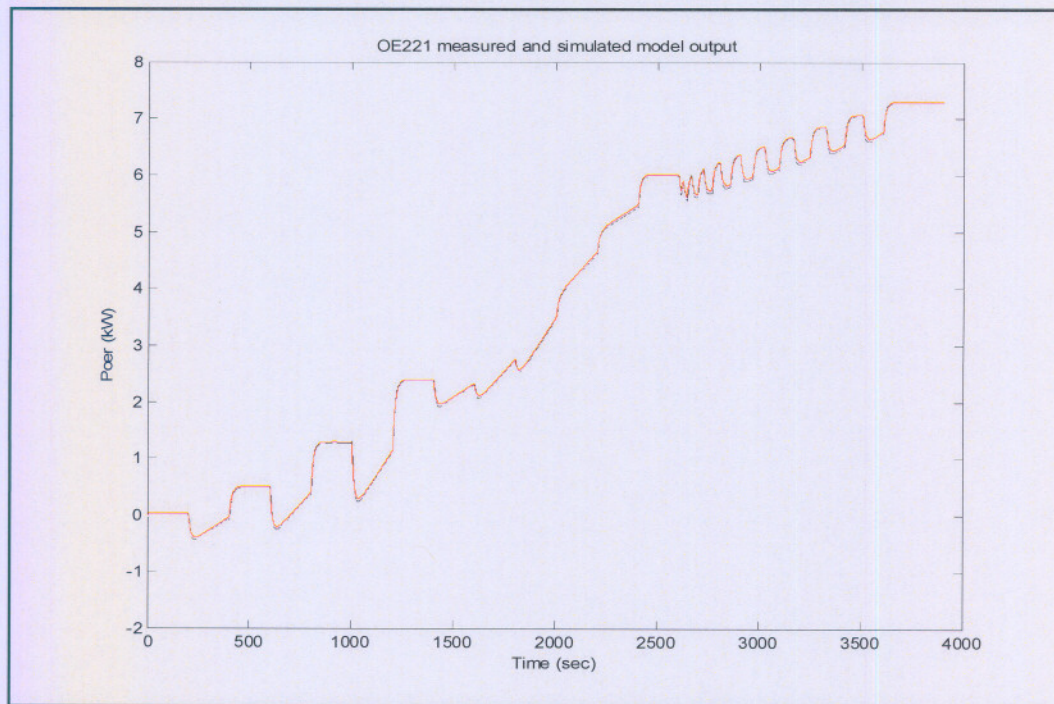


Figure 4-16: The simulated output of the OE221 model (red) and the measured output

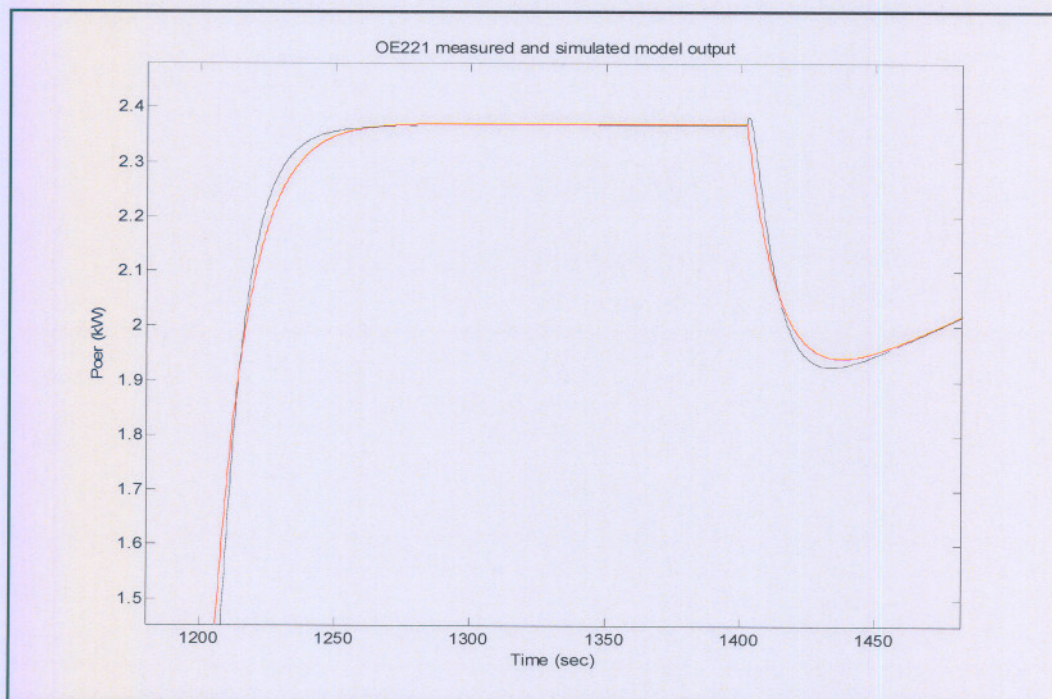


Figure 4-17: The more descriptive simulated output of the OE221 model (red)



The simulated output suggests that there are still some dynamical aspects of the system that are not modelled with the OE221 model. The residuals of the OE221 model are illustrated in Figure 4-18.

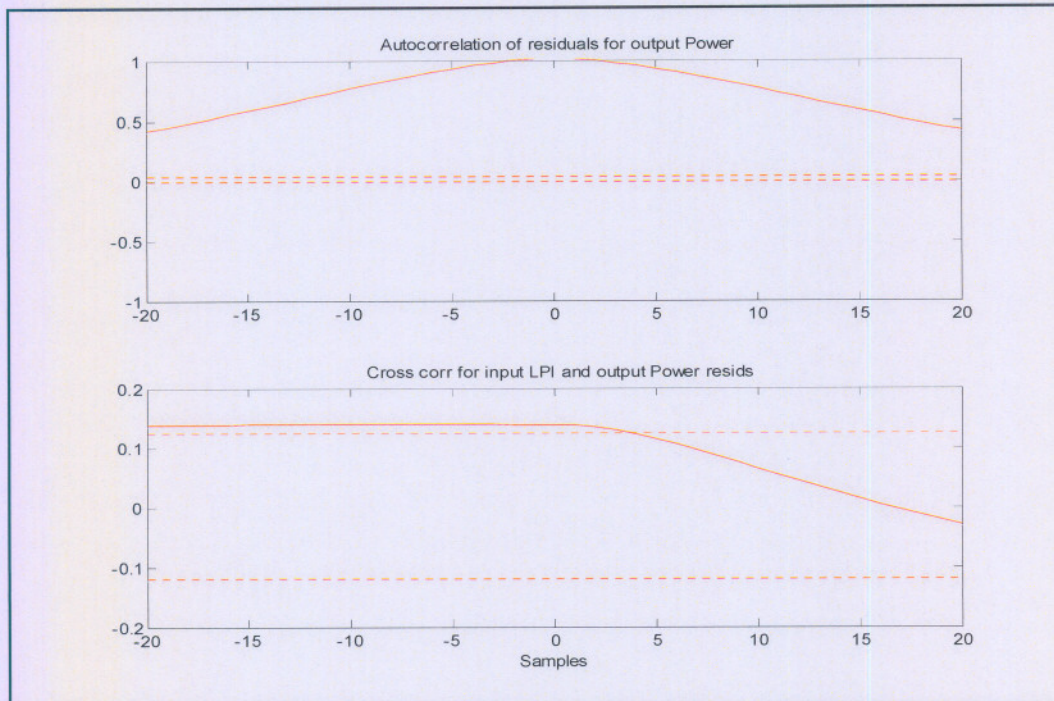


Figure 4-18: The residual analysis of the OE221 model

The residuals of the OE221 model are very poor. The cross- and the auto-correlations indicate that the model is dependant on past input values. This is not desirable. The parameters of the OE221 model are:

$$y(t) = \frac{(-4.168q^{-1} + 4.174q^{-2})}{(1 - 1.976q^{-1} + 0.9755q^{-2})} u(t) + e(t)$$

According to Table 4-3 the superior model, when only the fit criteria is considered, must be the OE331 model. The simulated output of this model is illustrated with the measured output of the system in Figure 4-19. The enlarged version of Figure 4-19 is illustrated in Figure 4-20.



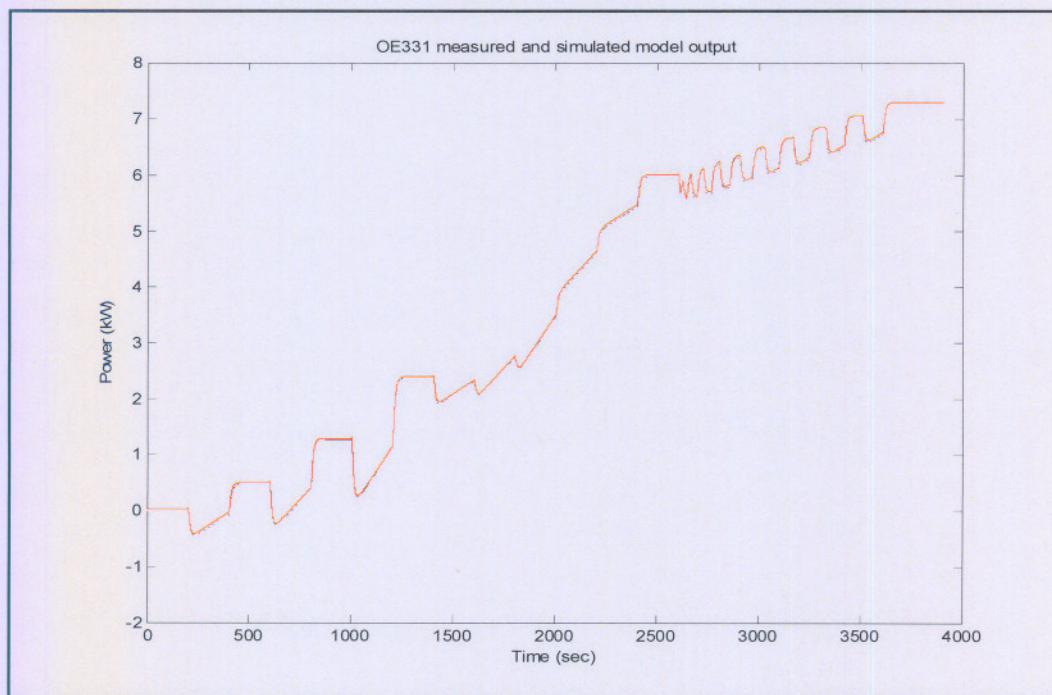


Figure 4-19: The red line is the simulated output of the OE331 model, the (unseen) black line is the measured output

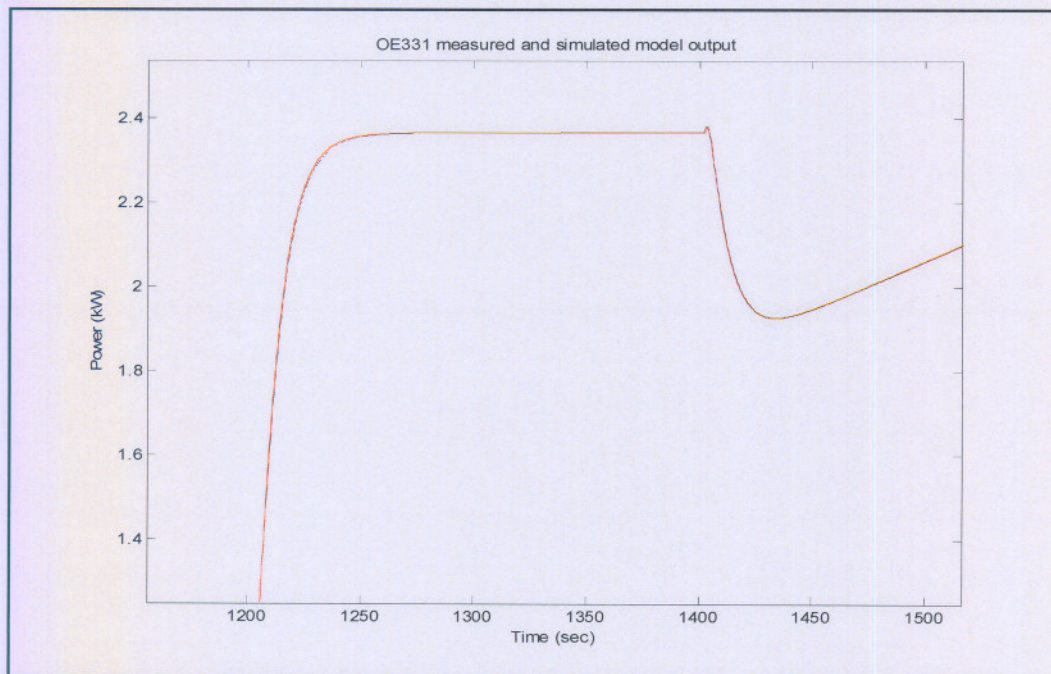


Figure 4-20: The enlarged version of Figure 4-19



The enlarged version of the simulated output graphic illustrates how good the dynamics of the system is modelled by the OE331 model. The residual analysis is illustrated in Figure 4-21.

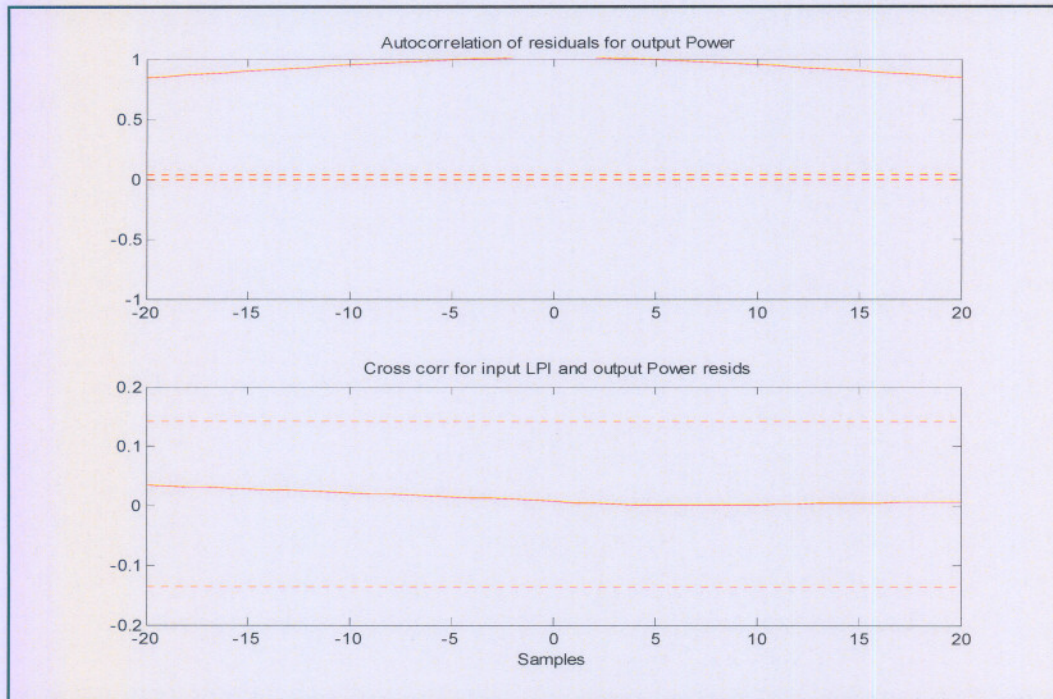


Figure 4-21: The residual analysis of the OE331 model

Unfortunately the residuals of the OE331 model are not ideal. Even worse than the residuals of the OE221 model. The autocorrelation residual must be normally distributed. If it is not, the model does not capture all the information of the data. This model could not be used for prediction or simulation of the system. The parameters that were calculated were:

$$y(t) = \frac{(2.807q^{-1} - 6.418q^{-2} + 3.612q^{-3})}{(1 - 2.826q^{-1} + 2.657q^{-2} - 0.8307q^{-3})}u(t) + e(t)$$



4.4.1.5 Box-Jenkins model identification

The algorithm that is used to calculate the parameters of the Box-Jenkins model structure (Prediction Error) and its properties were described in section 4.2.2.

The following property settings were used in Matlab© for the identification:

- Initial condition – Zero
- Focus – Simulation
- Covariance – Estimate

The default settings were used for all the other properties. Various second and third order models were calculated and the result of their identification is listed in Table 4-4.

Table 4-4

Name	n_b	n_c	n_d	n_f	Fit
BJ33331	3	3	3	3	99.82
BJ32331	3	2	3	3	99.82
BJ33231	3	3	2	3	99.82
BJ33321	3	3	3	2	99.60
BJ32321	3	2	3	2	99.60
BJ33221	3	3	2	2	99.60
BJ22331	2	2	3	3	99.44
BJ22221	2	2	2	2	99.19
BJ32221	3	2	2	2	84.69
BJ23221	2	3	2	2	84.56
BJ22231	2	2	2	3	80.49
BJ22321	2	2	3	2	54.84
BJ23321	2	3	3	2	54.53
BJ23231	2	3	2	3	2.28
BJ23331	2	3	3	3	-50.38



In this section the BJ22221 and BJ33331 models will be evaluated. The BJ33331 model has the best fit of the models that was calculated, this usually means it captures the dynamics of the system best. The BJ22221 model will be evaluated first. In Figure 4-22 the measured output of the system is illustrated with the simulated output of the BJ22221 model. The more descriptive enlarged version of Figure 4-22 is illustrated in Figure 4-23.

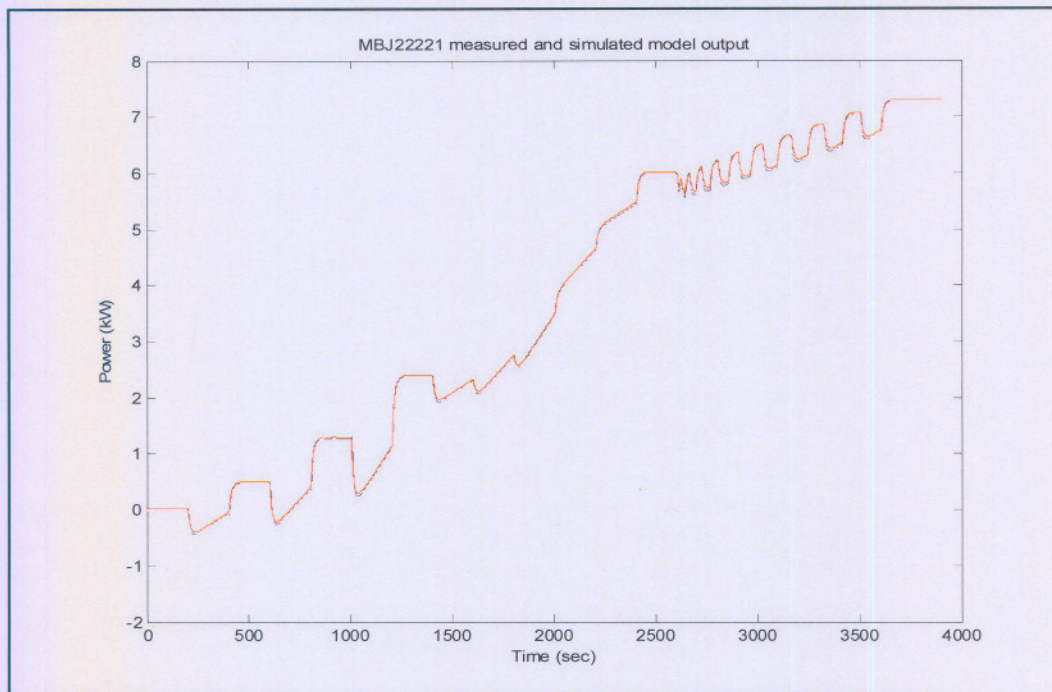


Figure 4-22: The measured output of the system (black) and the simulated output of the BJ22221 model (red)

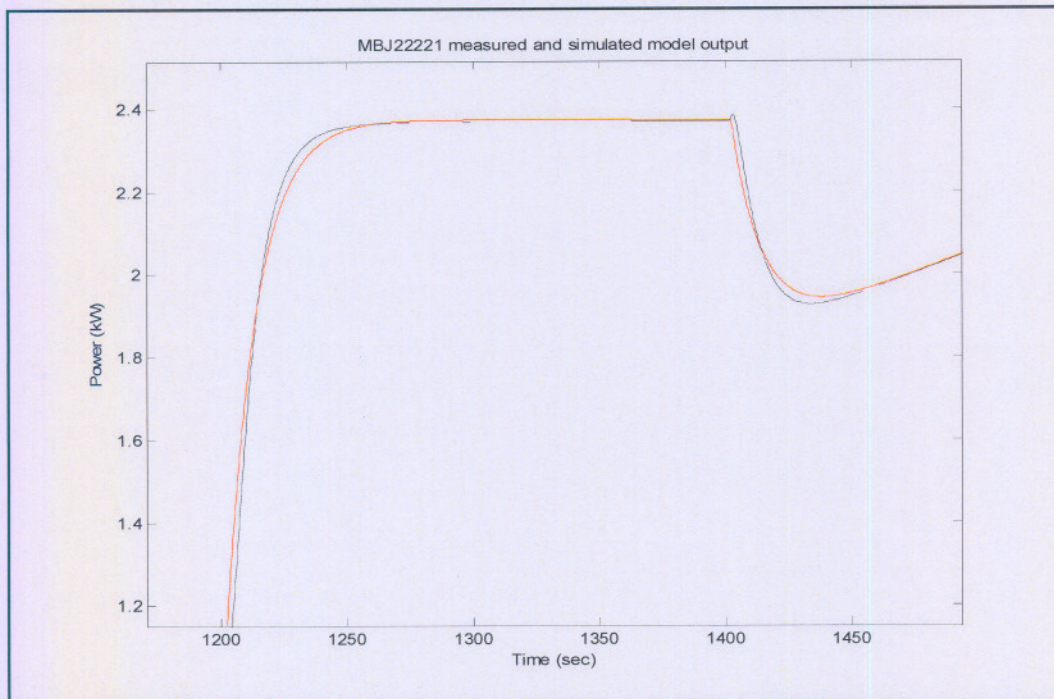


Figure 4-23: The more descriptive version of Figure 4-22

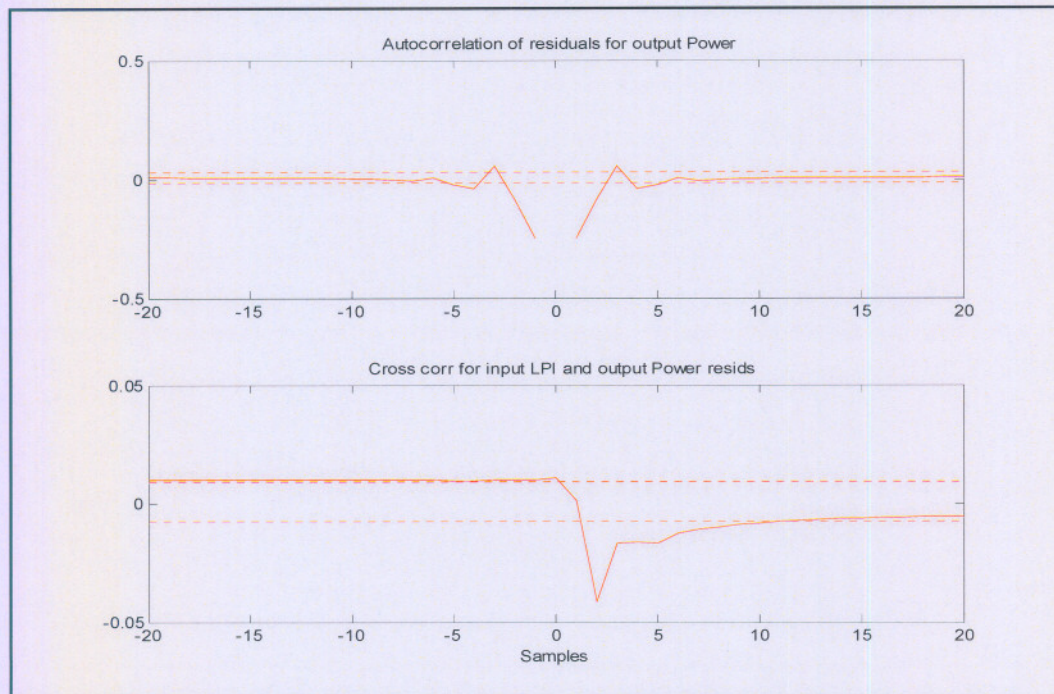


Figure 4-24: The residual analysis of the BJ22221 model



The BJ22221 model does not describe the dynamical behaviour perfectly according to Figure 4-23. The residual analysis of the BJ22221 model (Figure 4-24) reveals that the model has a very good auto-correlation. The cross-correlation is also very good and the function hardly leaves the 95% confidence interval. The parameters that were calculated for the BJ22221 model are:

$$y(t) = \frac{(-4.168q^{-1} + 4.174q^{-2})}{(1 - 1.976q^{-1} + 0.9755q^{-2})}u(t) + \frac{(1 + 0.5363q^{-1} + 0.1658q^{-2})}{(1 - 1.899q^{-1} + 0.9043q^{-2})}e(t)$$

The BJ33331 model is the best model in Table 4-4 according to the fit criteria. The simulated output of this model is illustrated in Figure 4-25. This figure is also zoomed into to produce Figure 4-26.

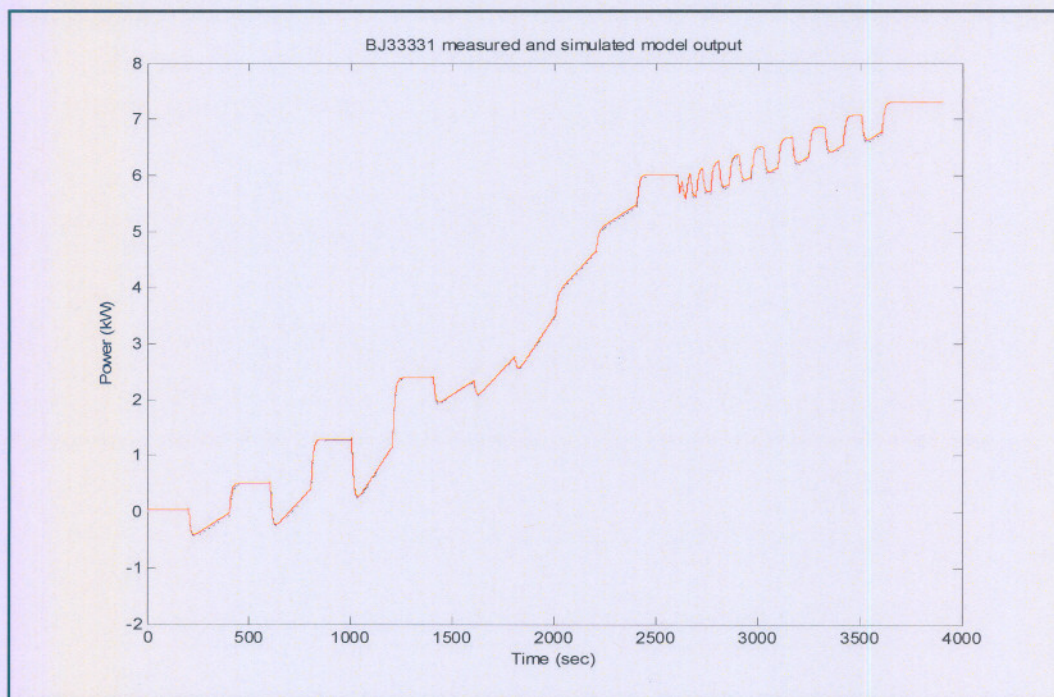


Figure 4-25: The simulated output of the BJ33331 model (red) and the system output



In Figure 4-25 the output of the system is almost invisible. The model describes the behaviour of the system well. This is even better illustrated in Figure 4-26.

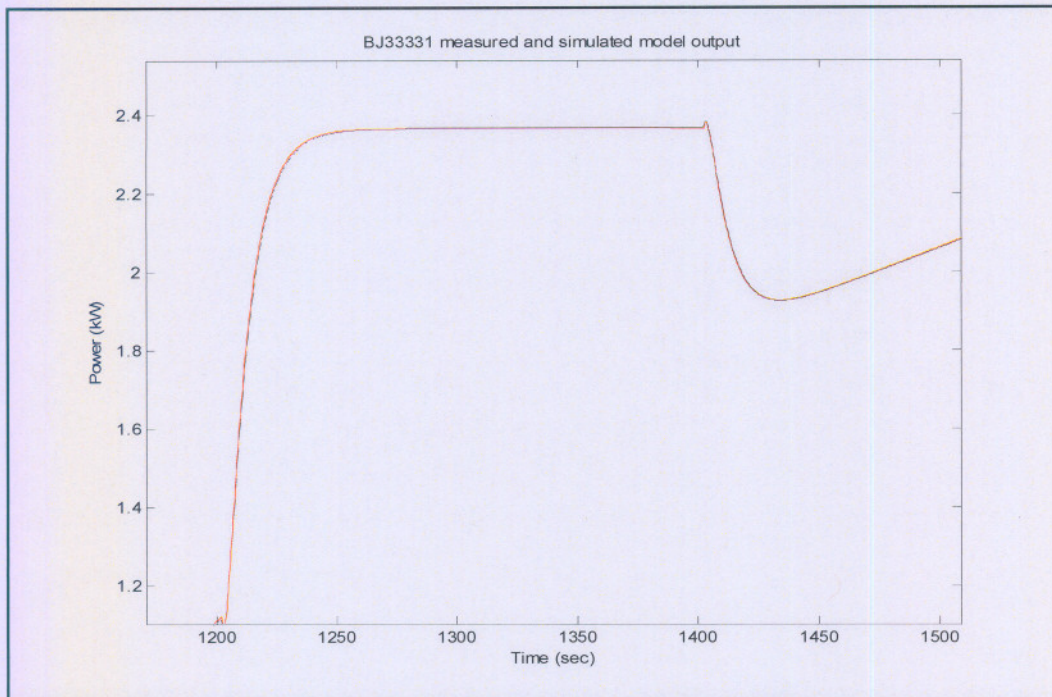


Figure 4-26: The enlarged version of Figure 4-25

The only other test that must be performed on the BJ33331 model is the residual analysis. The residual analysis of the BJ33331 model is illustrated in Figure 4-27. The residuals of the BJ33331 model are very good. The auto-correlation function settles within the 95% confidence interval within three time lags. The cross-correlation has only one time lag outside the interval. The BJ33331 model is a very good model. The parameters of this model were calculated to be:



$$y(t) = \frac{(2.807q^{-1} - 6.418q^{-2} + 3.612q^{-3})}{(1 - 2.826q^{-1} + 2.657q^{-2} - 0.8307q^{-3})}u(t) + \frac{(1 - 0.8449q^{-1} + 0.3583q^{-2} + 0.0549q^{-3})}{(1 - 2.302q^{-1} + 1.675q^{-2} - 0.3731q^{-3})}e(t)$$

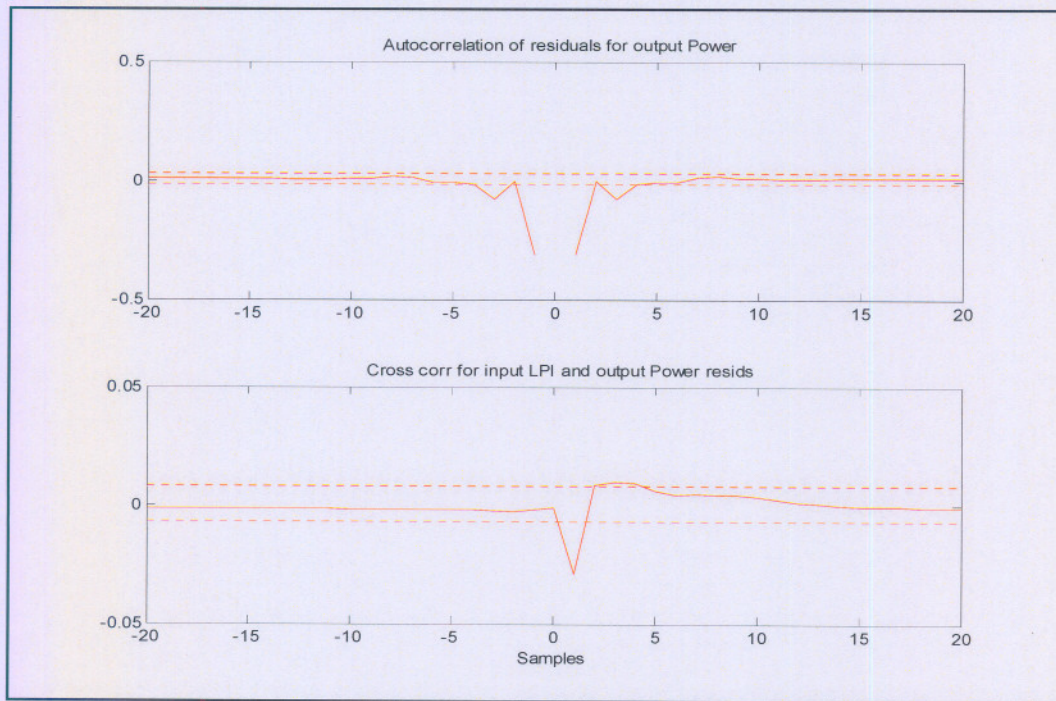


Figure 4-27: The residuals of the BJ33331 model

4.4.1.6 State space model identification

The algorithm that is used to calculate the parameters of the State-Space models were described in section 2.9.6.1. The following settings were used:

- Initial condition – Zero
- Focus – Simulation
- Covariance – Estimate



MISO system identification techniques using the PBMR data.

System identification of the PBMM

The default settings were used for all the other properties.

The results of the various models that were estimated in this section are listed in

Table 4-5.

Table 4-5

Name	<i>n</i>	Fit
N4S8	2	99.32
N4S4	2	98.92
N4S3	3	91.56
N4S2	2	68.03

The order of some of the models that were calculated is outside the specification of this study. This was done just to see if the system could be modelled with SS models. According to Table 4-5 only SS models with an order of higher than four is good enough. The best fit was obtained with the eight order model.

The poor fit of the second order SS model (n4s2) is illustrated in Figure 4-28.



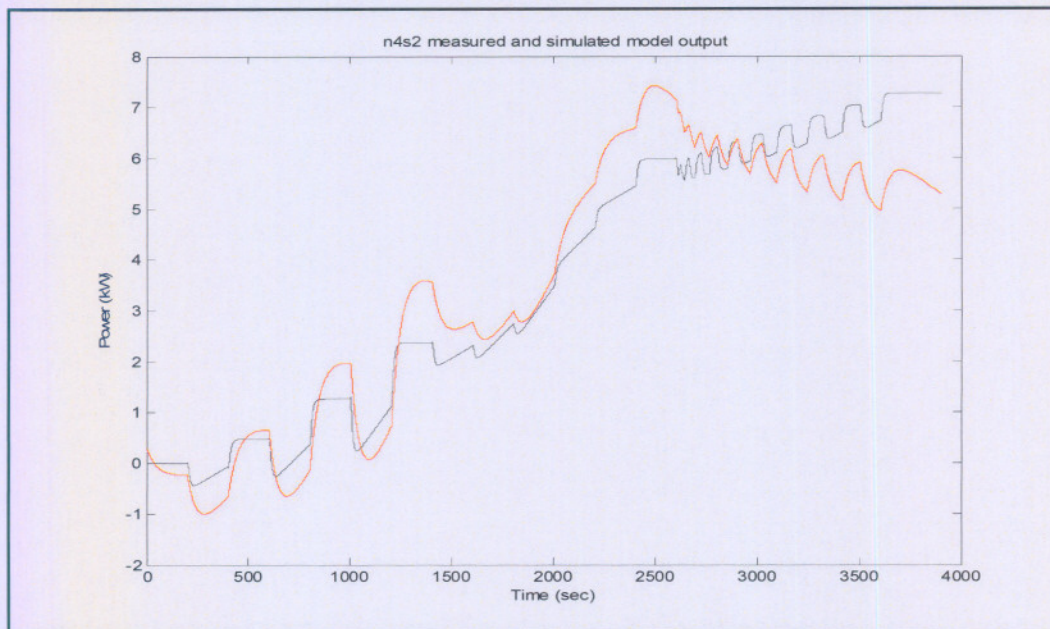


Figure 4-28: The simulated output of the second order SS model (red) and the system output

The third order SS model (Figure 4-29) captures more of the dynamical behaviour of the system, but not nearly enough.

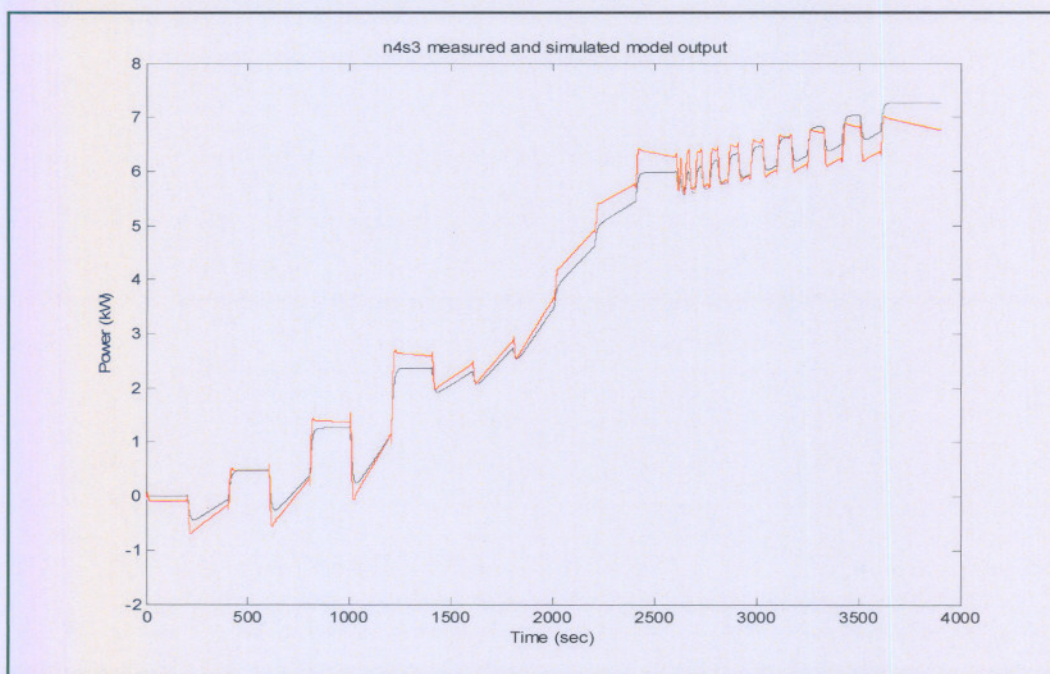


Figure 4-29: The simulated output of the n4s3 model (red) and the output of the system



The simulated output of the eight order model that was calculated purely out of curiosity is illustrated in Figure 4-30. In Figure 4-30 the performance of the eight order model seems to be good, but when the figure is closely investigated (Figure 4-31) it becomes clear that the model does not capture the higher frequency characteristics of the system. The poor performance of the second and third order models is the reason that the models will not be evaluated further.

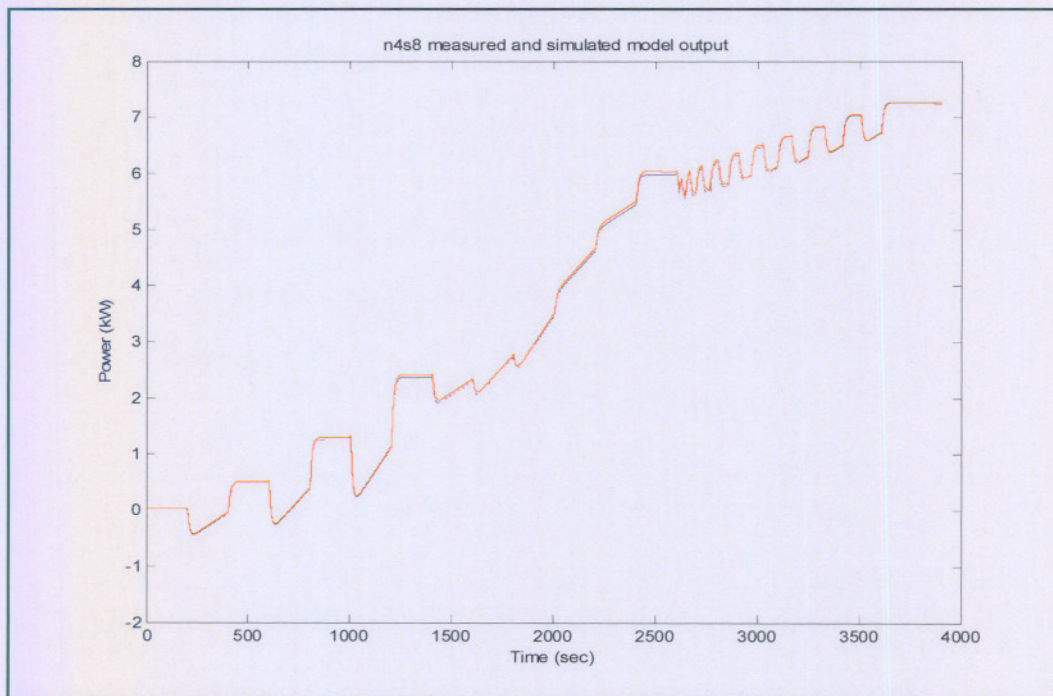


Figure 4-30: The simulated output of the n4s8 model (red) and the output of the system

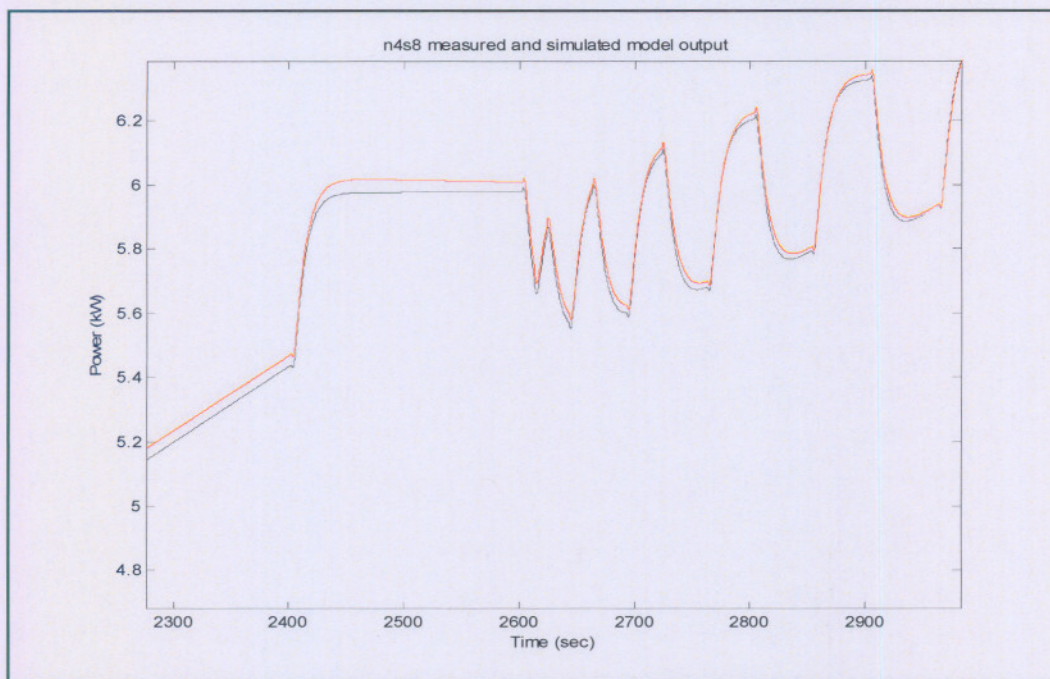


Figure 4-31: The enlarged version of Figure 4-30

4.4.1.7 Validation of best model for Low-Pressure injection system identification

The best model for the LPI process is the BJ33331 model. This model has, with the exception of the BJ22221 model, the best auto-correlation function. The model has the best fit and the best cross-correlation. Unfortunately a sixth order equation will describe this model when represented in discrete time. The AMX3331 and ARX331 models are the best third order discrete time models. The AMX3331 model has better residuals and would be the best choice.



4.4.2 High-Pressure extraction system identification

4.4.2.1 High-Pressure extraction simulation

The constraints of the physical system were taken into account as well as the criterion that the input signal must meet, and the following input signal was chosen:

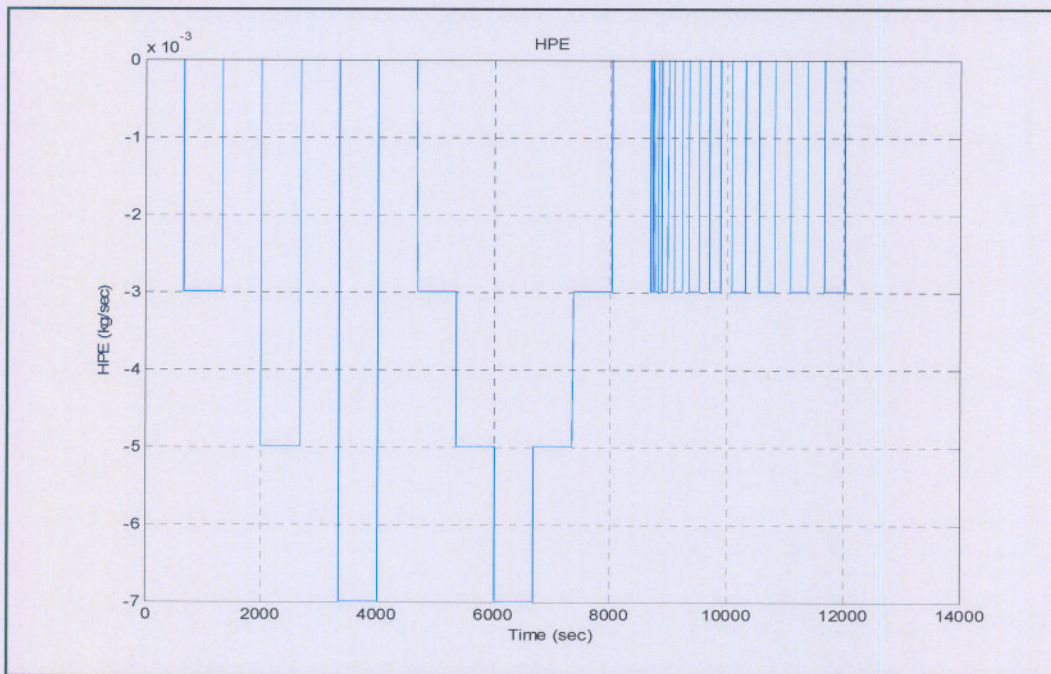


Figure 4-32: The input used for identification of the HPE process

The best input signal must be positive and negative, but this is not possible for this system as the physical system does not allow this. The output that was recorded is illustrated in Figure 4-33.



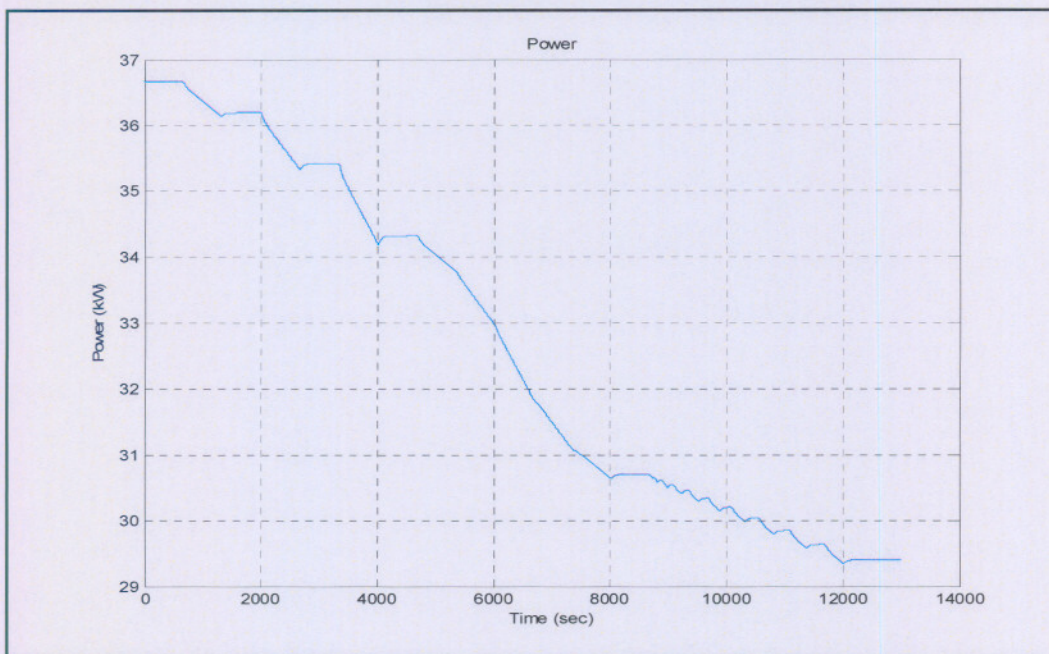


Figure 4-33: The output of the system when it receives input described in Figure 4-32

In Figure 4-33 it is clear that the system was always inside the parameters for normal power mode operation. In Figure 4-33 the offset of 36.6 kW is obvious. This offset must be subtracted to make the data suited for identification. The output that is then obtained after the offset is subtracted is illustrated in Figure 4-34.



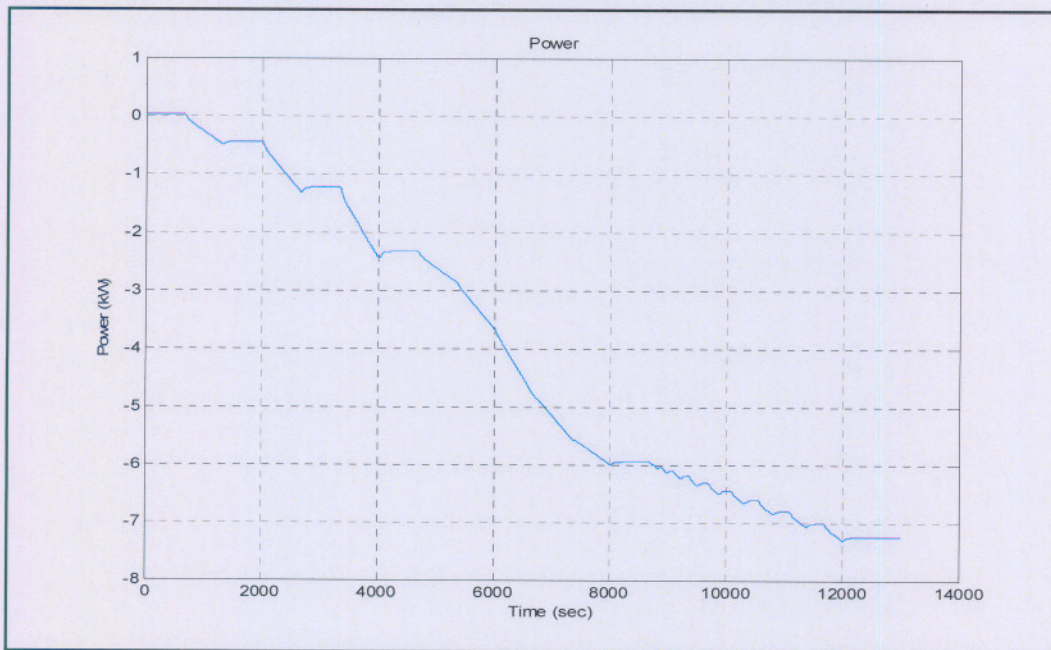


Figure 4-34: The output that is used for identification

There are no other problems with the input or output data and thus the data described in Figure 4-32 and Figure 4-34 is used in the following sections to identify different models:

4.4.2.2 ARX model identification

The first parameter that must be estimated before any models can be identified is the value of k . When the input (Figure 4-32) and the output (Figure 4-33) of the system are investigated the value of k for this process is one. This means that it takes an input one sample time to have an effect on the output.



The different second and third order ARX models could now be calculated for this process. The same settings were used in this section as those used in section 4.4.1.2.

The results of the second and third order ARX models that were calculated for the HPE process are listed in Table 4-6.

Table 4-6

Name	n_a	n_b	Fit
ARX221	2	2	99.85%
ARX331	3	3	99.83%
ARX231	2	3	99.78%
ARX321	2	2	99.74%

According to the results in Table 4-6 the ARX221 model has the best fit. This model will be evaluated first to see if its residuals are satisfactory. The next best model, the ARX331 model, will be evaluated as well to compare its results with the ARX221 model. The simulated output of the ARX221 model and the measured output of the system are compared in Figure 4-35. The more insightful version of Figure 4-35 is illustrated in Figure 4-36. The model captures the dynamical behaviour of the system very well according to Figure 4-35.

The result of the residual analysis is illustrated in Figure 4-37. According to this figure the model has a relatively good auto-correlation function, but the cross-correlation function is not that good. The cross-correlation function leaves the confidence interval for two time lags and only returns at lag ten.



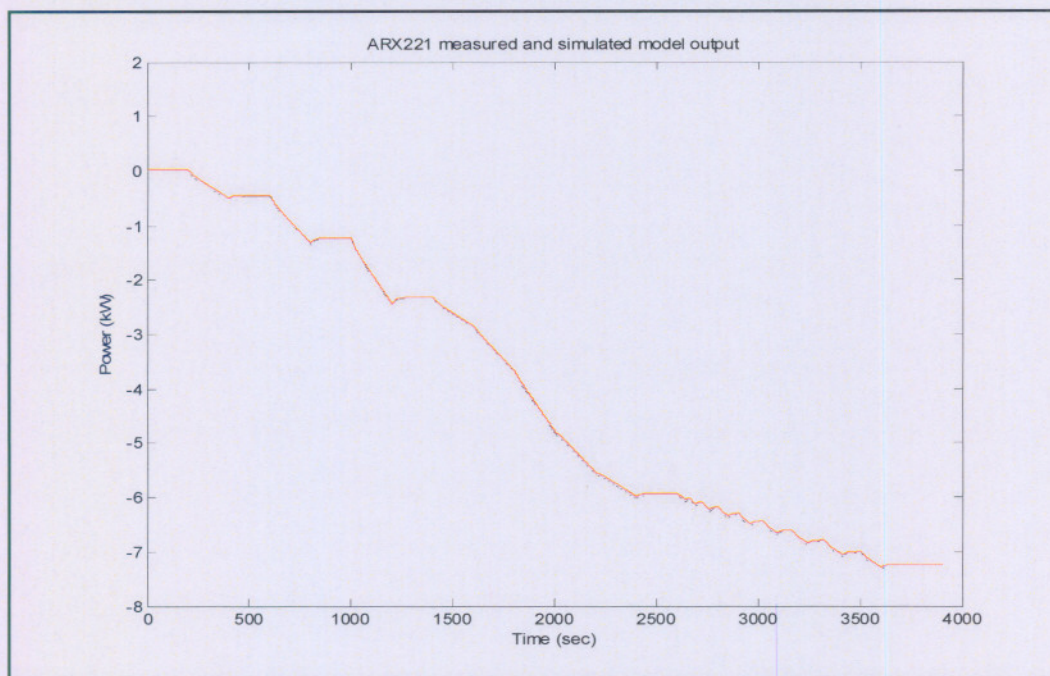


Figure 4-35: The simulated output of the ARX221 model (red) and the measured output of the system

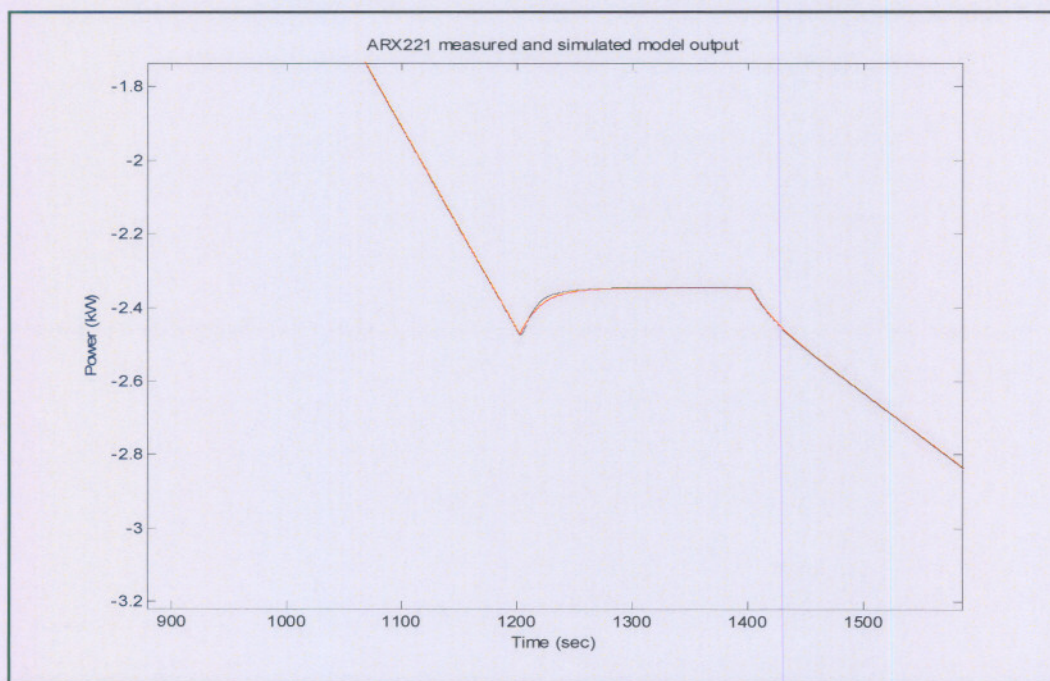


Figure 4-36: The enlarged version of Figure 4-35



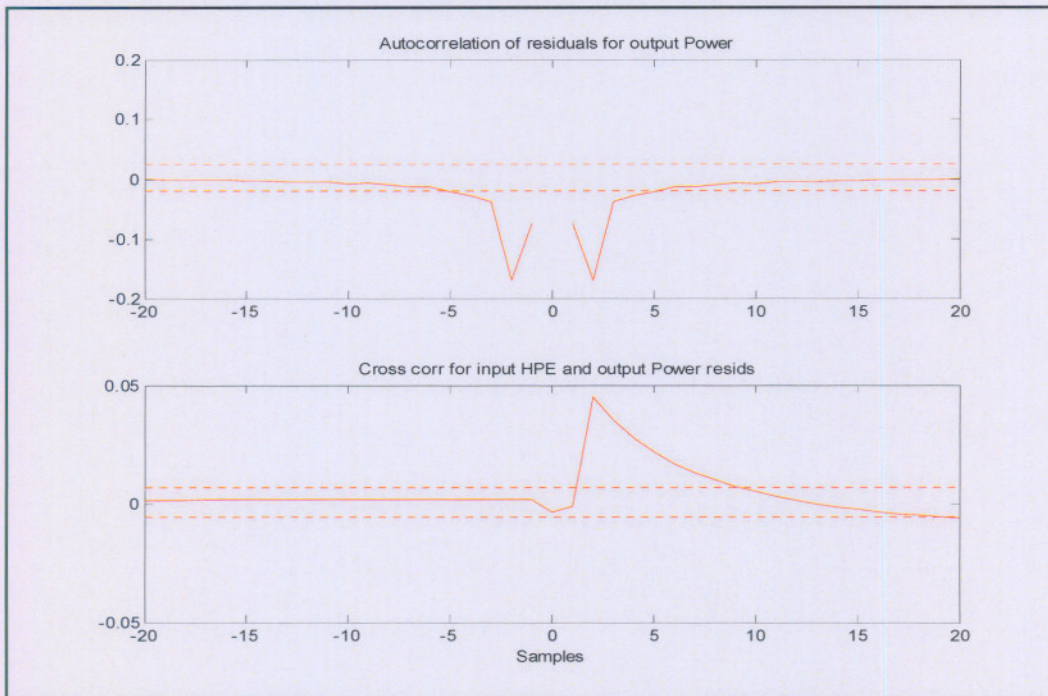


Figure 4-37: The residual analysis of the ARX221 model

The parameters of the ARX221 model that were calculated are:

$$(1 - 1.984q^{-1} + 0.9837q^{-2})y(t) = (0.5247q^{-1} - 0.5209q^{-2})u(t) + e(t)$$

The ARX331 model is now evaluated in order to compare its residuals to the residuals of the ARX221 model. The simulated output of the ARX331 model and the output of the system are illustrated in Figure 4-38. The more descriptive version of Figure 4-38 is illustrated in Figure 4-39. The simulated output of the ARX331 model also describes the system very accurately according to Figure 4-39.

The residuals of the ARX331 model are illustrated in Figure 4-40. The auto-correlation function of the ARX331 model is better than the auto-correlation of the ARX221 model. The auto-correlation is better, because it almost has the auto-correlation of white noise. This is desired. The cross-correlation on the



MISO system identification techniques using the PBMR data.

System identification of the PBMM

other hand is not ideal. There is the correlation between the error and input signals. The cross-correlation must be within the horizontal lines indicated on the figure. If the cross-correlation is not inside the indicated lines it usually means all the dynamics of the system are not fully modelled.

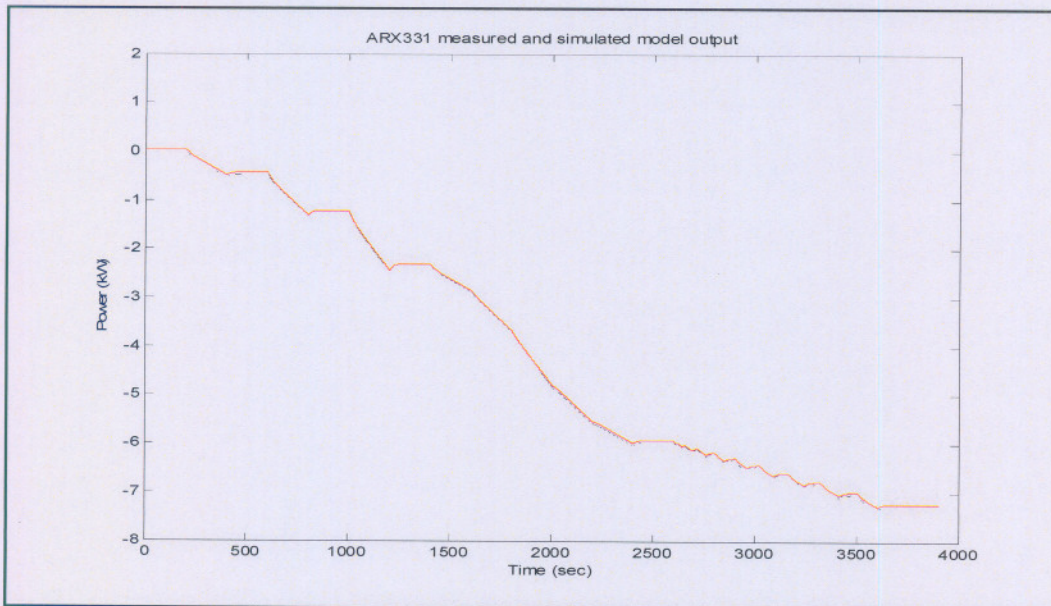


Figure 4-38: The simulated output of the ARX331 model (red) and the measured output



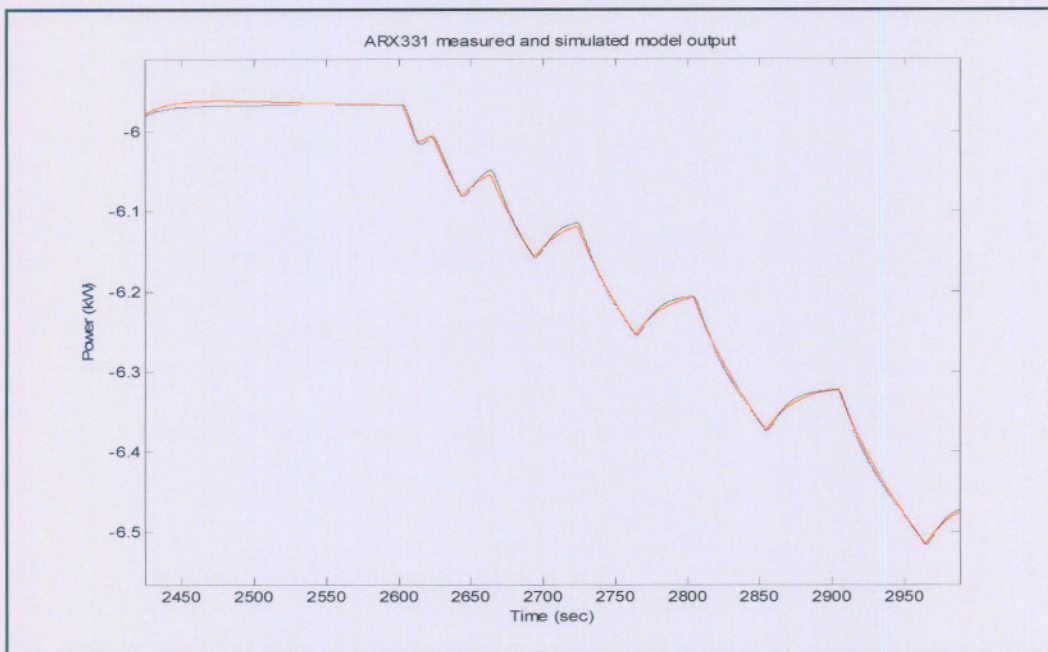


Figure 4-39: The more descriptive version of Figure 4-38

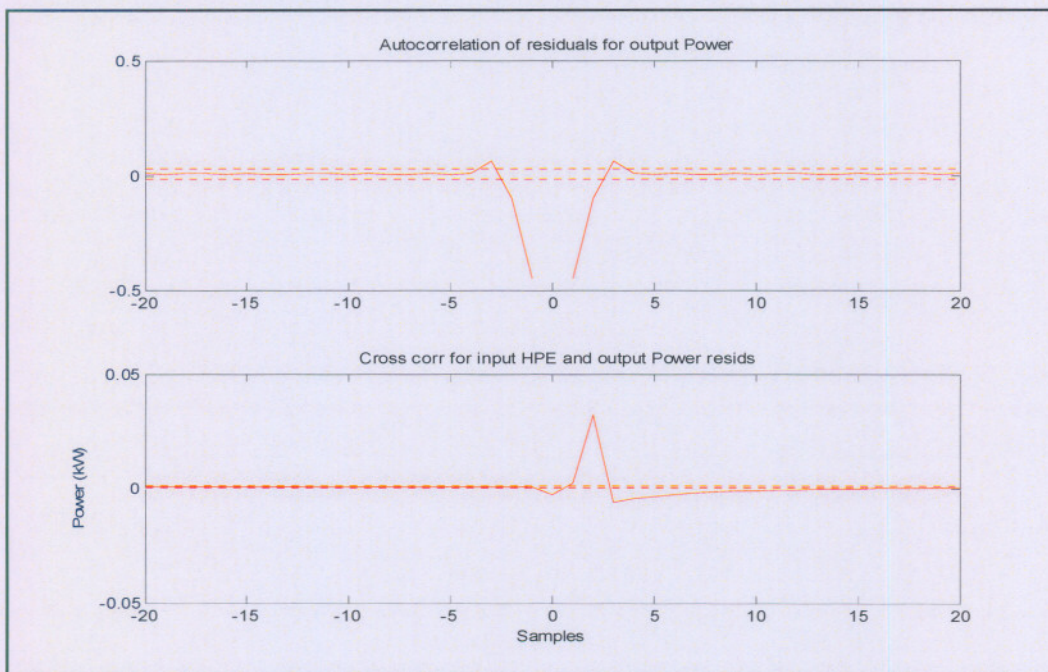


Figure 4-40: The results of the residual analysis



MISO system identification techniques using the PBMR data.

System identification of the PBMM

The parameters that were calculated for the ARX331 model:

$$(1 - 2.984q^{-1} + 2.968q^{-2} - 0.9838q^{-3})y(t) = (0.5192q^{-1} - 1.034q^{-2} + 0.5146q^{-3})u(t) + e(t)$$

4.4.2.3 ARMAX model identification

Different second and third order ARMAX models were calculated for the HPE process. The same settings were used in this section as was used in section 4.4.1.3. The results of the identification process are listed in Table 4-7.

Table 4-7

Name	n_a	n_b	n_c	Fit
AMX3221	3	2	2	99.88%
AMX3231	3	2	3	99.88%
AMX2321	2	3	2	99.87%
AMX2331	2	3	3	99.87%
AMX2221	2	2	2	99.86%
AMX2231	2	2	3	99.86%
AMX3331	3	3	3	-82.52%
AMX3321	3	3	2	-82.52%

The ARMAX models that best fit the data of the HPE process are the AMX3221 and AMX3231 models. The AMX2221 model almost has the same performance as the AMX3221 and AMX3231 models. The AMX2221 model will be evaluated first and then the AMX3231 model. The simulated output of the AMX2221 model is illustrated in Figure 4-41.



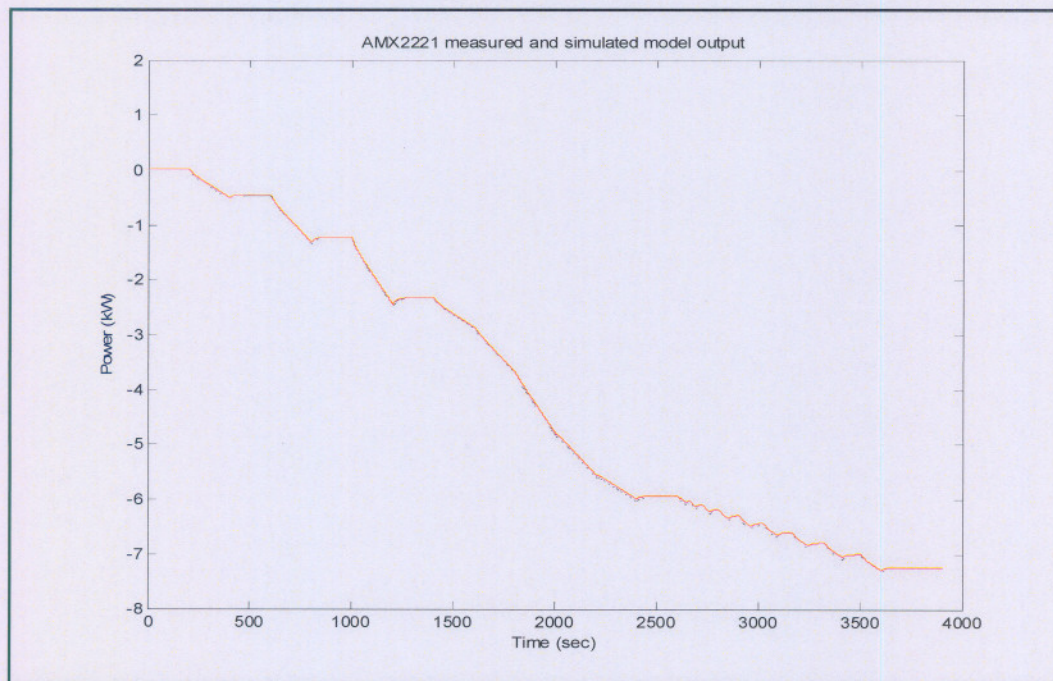


Figure 4-41: The simulated output of the AMX2221 model (red) and the measured output

When the simulated output of the AMX2221 model was analyzed at certain critical changes it could be seen that the model describes the dynamical behaviour of the system very well (Figure 4-42).

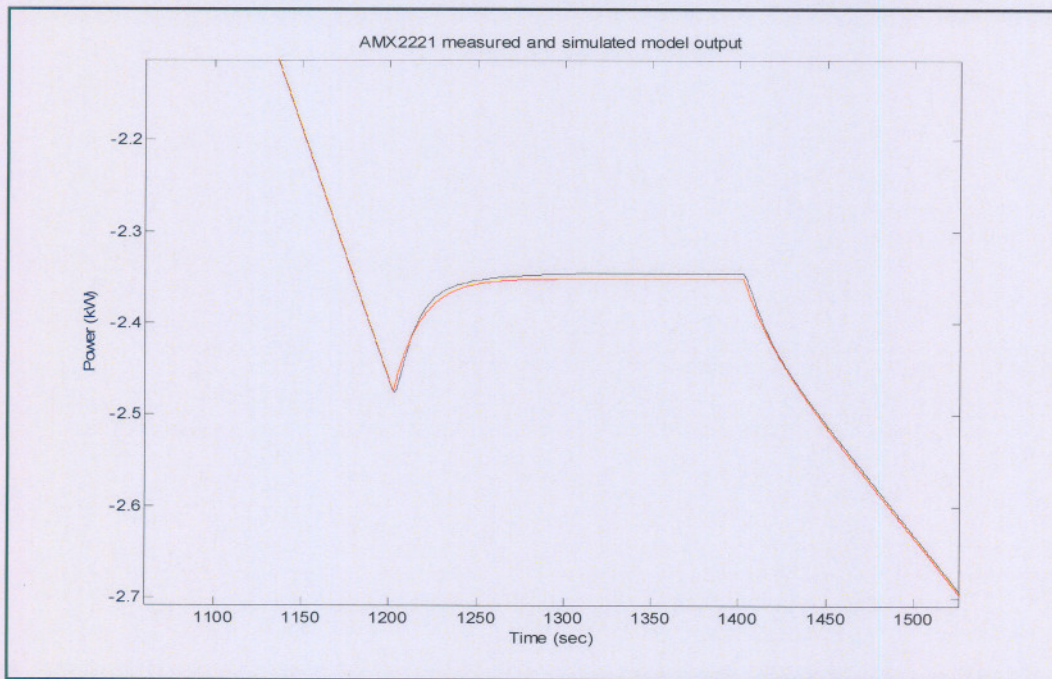


Figure 4-42: The critical analysis of the AMX2221 model at certain inputs

The residual analysis of the AMX2221 model (Figure 4-43) reveals that the model has a good auto-correlation function and the model does not depend on past outputs. The cross-correlation function of this model is not ideal. The input of the model is correlated with the error according to the cross-correlation function. This means that the model is not ideal. The parameters of the model that were calculated are:

$$\begin{aligned} (1 - 1.981q^{-1} + 0.9812q^{-2})y(t) &= (0.5676q^{-1} - 0.5632q^{-2})u(t) + \\ &+ (1 + 0.00295q^{-1} + 0.00104q^{-2})e(t) \end{aligned}$$



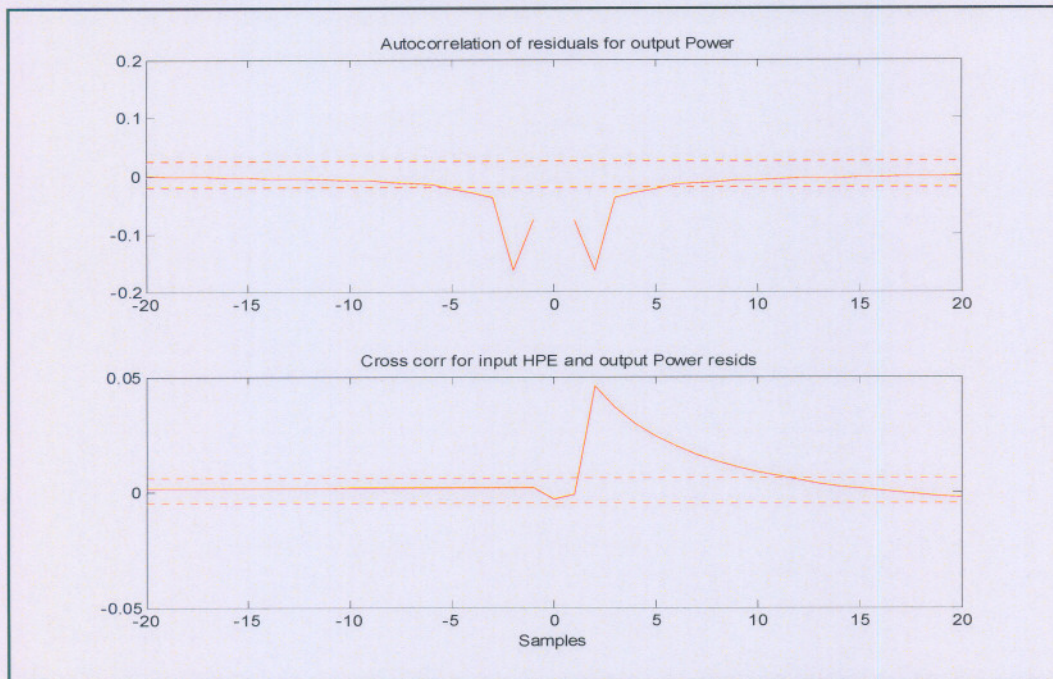


Figure 4-43: The results of the residual analysis

The model with the best fit, the AMX3231 will now be evaluated. The almost perfect simulated output of the model is illustrated in Figure 4-44. The small size of the error is illustrated in Figure 4-45.



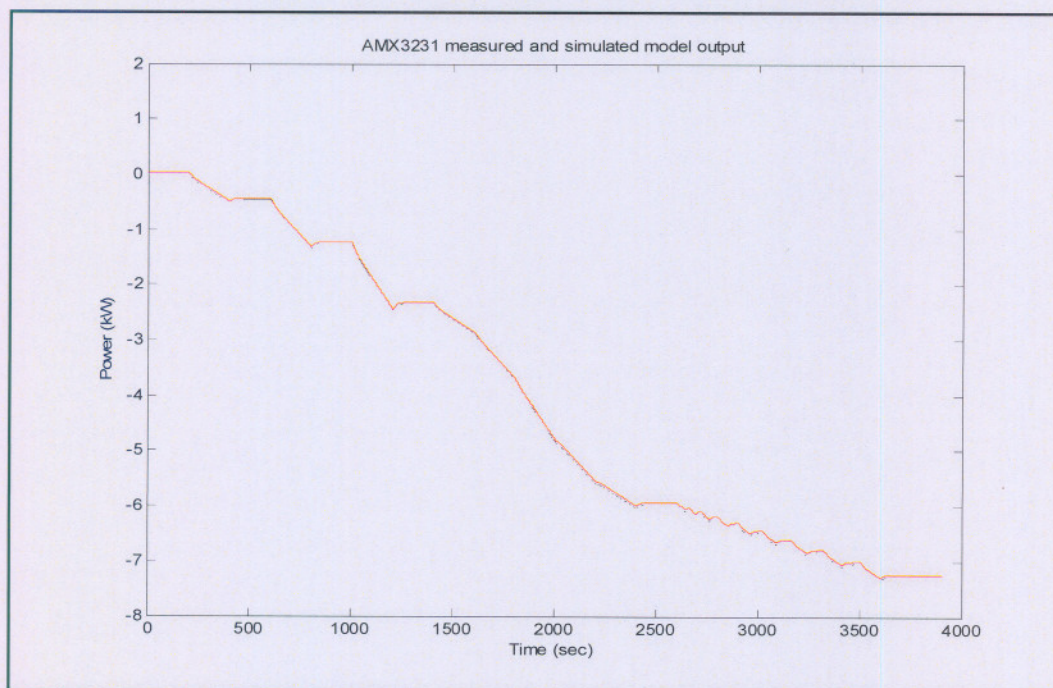


Figure 4-44: The simulated output of the AMX3231 model (red) and the desired output

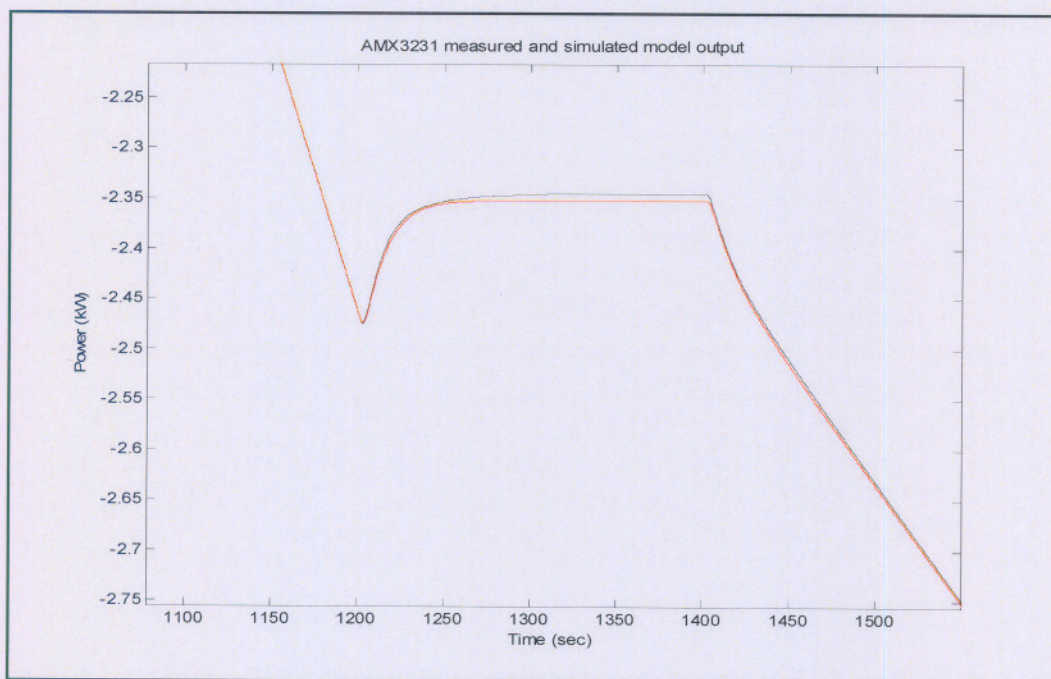


Figure 4-45: A partial view of Figure 4-44



The residuals of the AMX3231 model (Figure 4-46) reveal that the model is very suited according to the auto-correlation function. The cross-correlation function tells a different story though. The negative lags of the cross-correlation settles just outside the confidence interval. This means some dynamics of the system are not modelled.

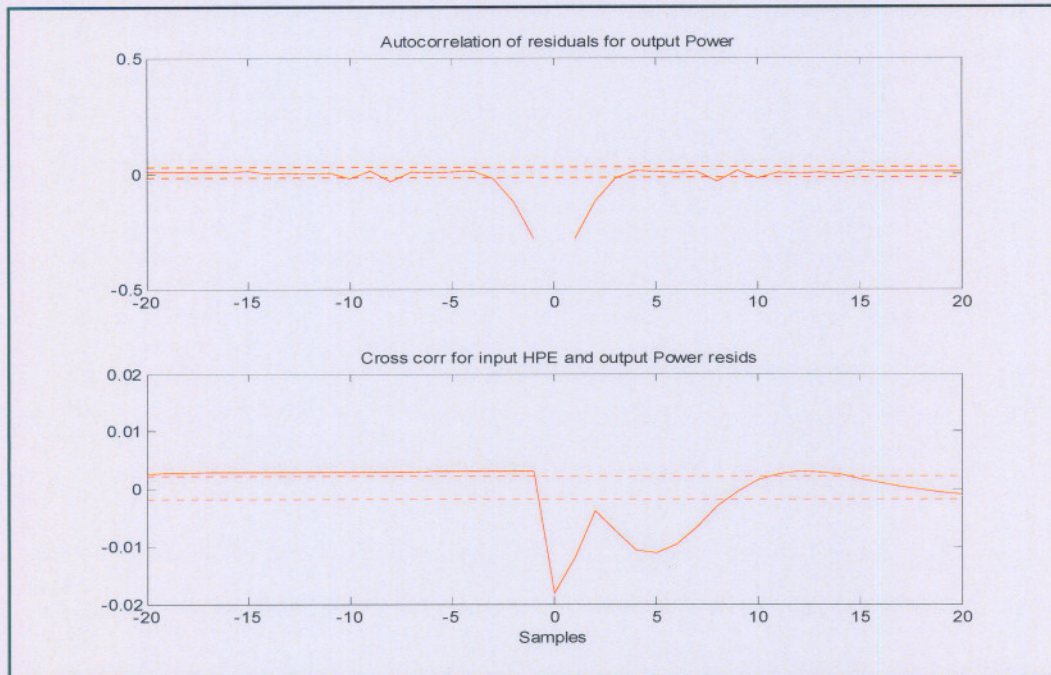


Figure 4-46: The residuals of the AMX3231 model

The parameters that were calculated for the AMX3231 model were:

$$\begin{aligned} (1 - 2.842q^{-1} + 2.688q^{-2} - 0.8455q^{-3})y(t) = & (0.0924q^{-1} - 0.0916q^{-2})u(t) + \\ & (1 - 1.025q^{-1} + 0.2099q^{-2} + 0.064q^{-3})e(t) \end{aligned}$$



4.4.2.4 Output Error model identification

Different second and third order Output Error models were calculated for the HPE process. The results of the identification process are listed in Table 4-8. The same settings were used in this section as were used in section 4.4.1.4

Table 4-8

Name	n_b	n_f	Fit
OE231	2	3	99.88%
OE221	2	2	99.86%
OE321	3	2	99.83%
OE331	3	3	-82.52%

The two models with best fits will be evaluated in this section. The OE221 model models the dynamics of the system very well. The partial view of the simulated output and the desired measured output is illustrated in Figure 4-47. There is a very small difference between the measured output and the simulated output of the model.

The residual analysis of the OE221 model reveals that the model has almost the perfect cross-correlation function. The input of the system is totally uncorrelated to the error of the model. The auto-correlation function of this model is the reason this model is unsuited for use. The error of this model is correlated to past errors.

The characteristic equation of this OE221 model was estimate to be:

$$y(t) = \frac{(0.5676q^{-1} - 0.5632q^{-2})}{(1 - 1.981q^{-1} + 0.9812q^{-2})} u(t) + e(t)$$



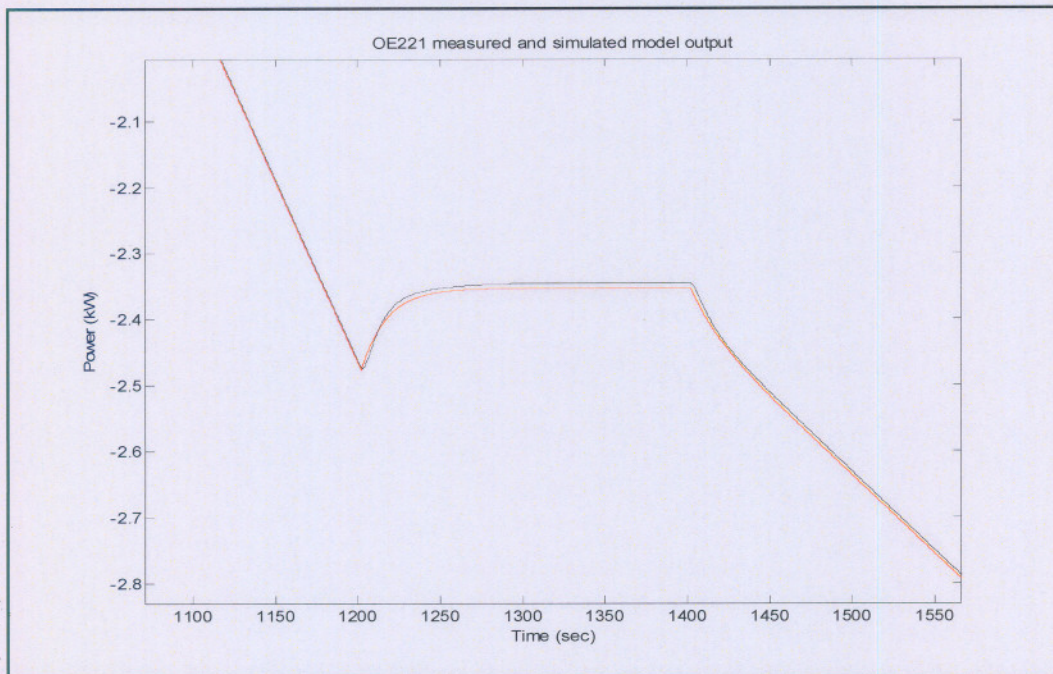


Figure 4-47: The partial view of the simulated output of the OE221 model and the output

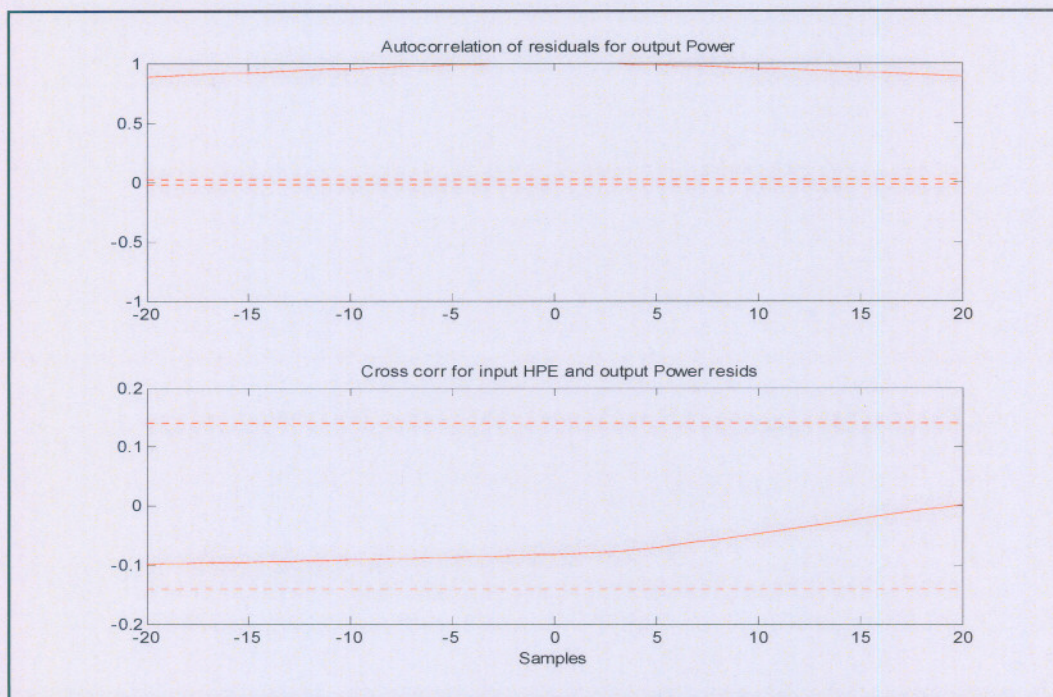


Figure 4-48: The residual analysis of the OE221 model



The OE model that best describes the dynamical behaviour of the HPE process is the OE231 model. The simulated output of the model overshadows the measured output of the system in Figure 4-49. The partial view of Figure 4-49, Figure 4-50 illustrates how small the difference between the simulated output and the measured output is.

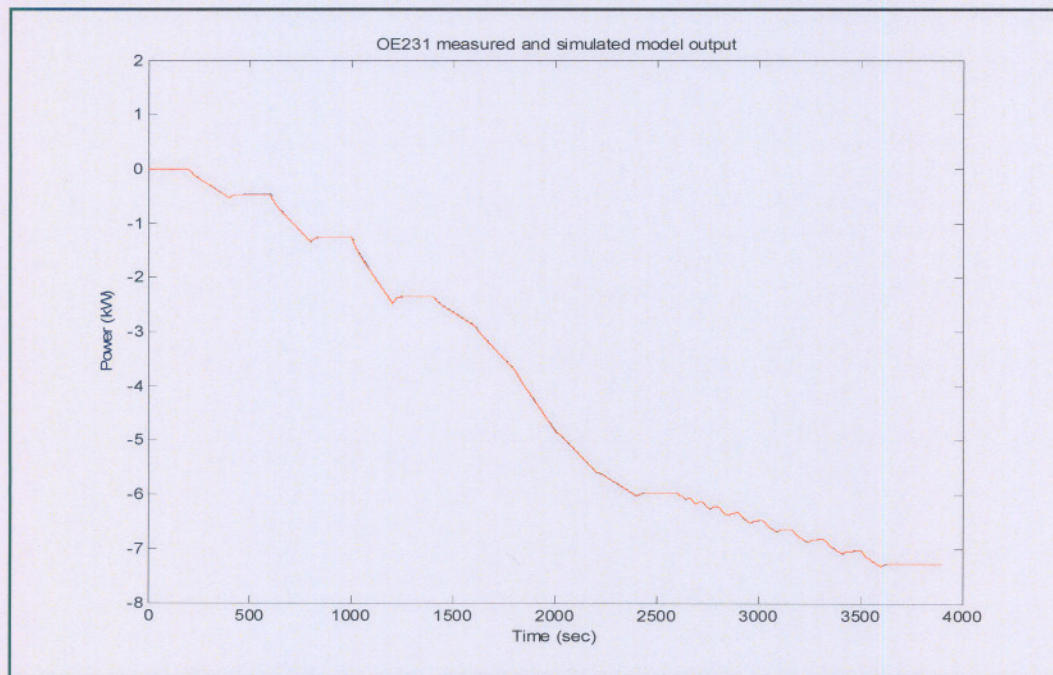


Figure 4-49: The simulated output of the OE231 model (red) and the measured output

The OE231 model has almost identical residuals to the OE221 model (Figure 4-51). This means this model is also unsuited for prediction. This model is described by:

$$y(t) = \frac{(0.09235q^{-1} - 0.0916q^{-2})}{(1 - 2.842q^{-1} + 2.688q^{-2} - 0.8455q^{-3})} u(t) + e(t)$$





Figure 4-50: The partial view of the simulated output and the measured output

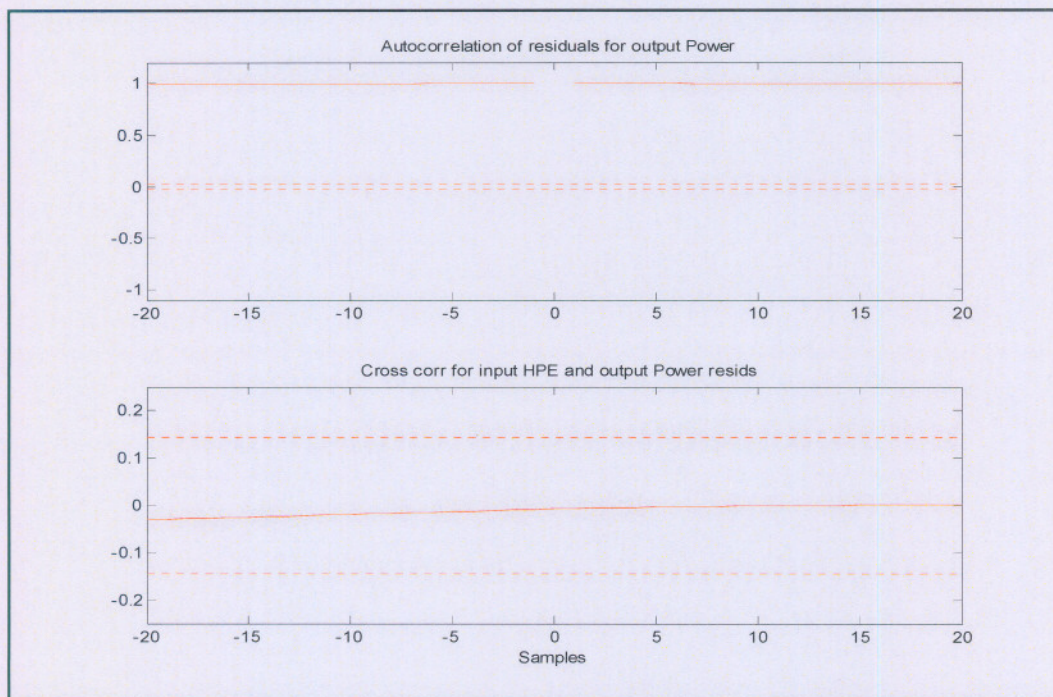


Figure 4-51: The residuals of the OE231 model



4.4.2.5 Box-Jenkins model identification

The results of the identification of different second and third order Box-Jenkins models were calculated for the HPE process listed in Table 4-9. The same settings were used in this section as those used in section 4.4.1.5.

Table 4-9

Name	n_b	n_c	n_d	n_f	Fit
BJ23331	2	3	3	3	99.88%
BJ33321	3	3	3	2	99.88%
BJ22331	2	2	3	3	99.88%
BJ23231	2	3	2	3	99.88%
BJ32321	3	2	3	2	99.88%
BJ33221	3	3	2	2	99.88%
BJ22231	2	2	2	3	99.88%
BJ32221	3	2	2	2	99.88%
BJ22221	2	2	2	2	99.86%
BJ23321	2	3	3	2	99.86%
BJ22321	2	2	3	2	99.86%
BJ23221	2	3	2	2	99.86%
BJ32331	3	2	3	3	47.93%
BJ33331	3	3	3	3	47.76%
BJ33231	3	3	2	3	46.61%

The lowest order model, the BJ22221 model, performs quite well to its higher order counterparts according to the fit criteria. This is the reason this model will be investigated in this section. The other model that will be investigated is one of the models that have the best performance according to Table 4-9, the BJ22331 model.

The simulated output of the BJ22221 model, when it receives the input as described in Figure 4-32, is illustrated in Figure 4-52. The partial view of the



The partial view (Figure 4-53) shows that the BJ22221 model captures the dynamical behaviour of the system very well. The residuals of the BJ22221 model are illustrated in Figure 4-54. The auto-correlation function of the BJ22221 model is not ideal, but has the form of the auto-correlation function that is desired. The auto-correlation function only enters the confidence interval at lag seven. The cross-correlation function of the BJ22221 model is not exceptionally good. The input and the error are correlated in some areas. The cross-correlation function only enters the confidence interval at lag seventeen. The parameters that were calculated for the BJ22221 HPE model are:

$$y(t) = \frac{(0.5676q^{-1} - 0.5632q^{-2})}{(1 - 1.981q^{-1} + 0.9812q^{-2})}u(t) + \frac{(1 - 0.0925q^{-1} - 0.0726q^{-2})}{(1 - 1.1935q^{-1} + 0.9358q^{-2})}e(t)$$

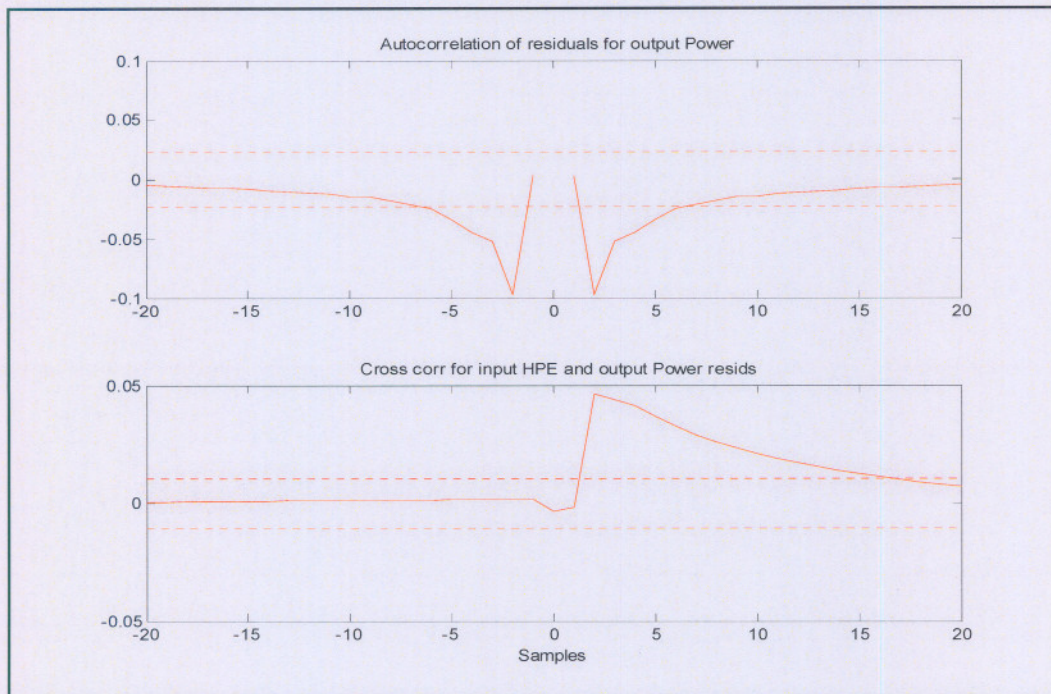


Figure 4-54: The residuals of the BJ22221 model



The simulated output of the BJ22331 model is illustrated in Figure 4-55.

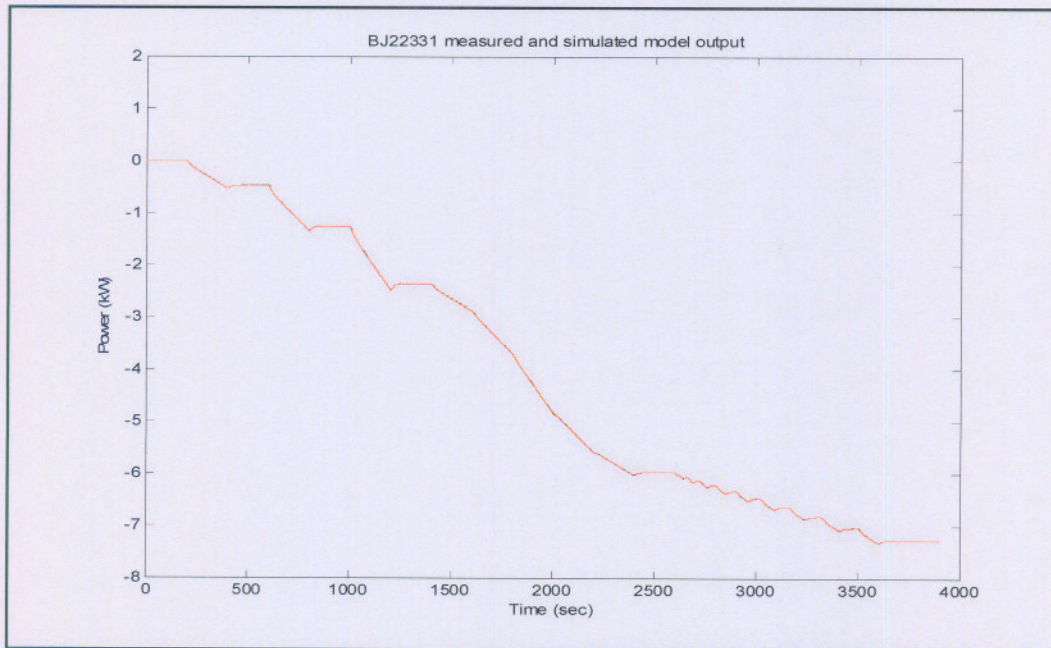


Figure 4-55: The simulated output of the BJ22331 model (red) and the measured output



Figure 4-56: A partial view of Figure 4-55



The partial view of the simulated output of the BJ22331 model verifies that this model captures the dynamical behaviour of the system accurately. The residuals of this model are illustrated in Figure 4-57. The cross-correlation of this model is not ideal. The cross-correlation function settles just outside the confidence interval for negative lags, and for positive lags it settles inside the interval in eight lags. The auto-correlation of the BJ22331 model is almost perfect.

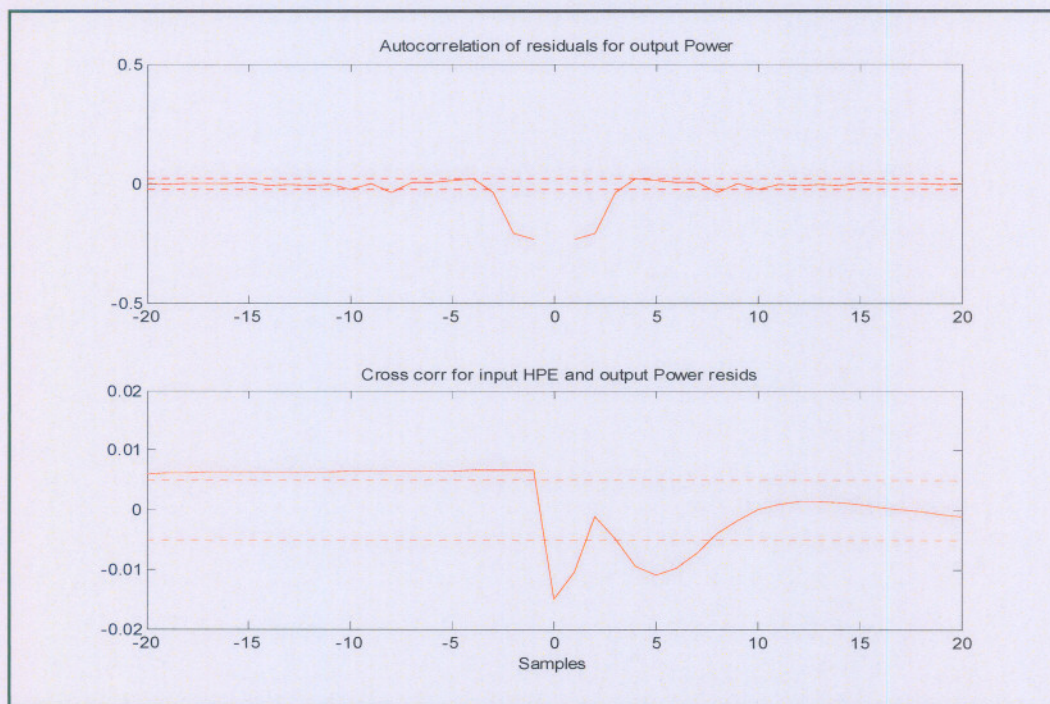


Figure 4-57: The residuals of the BJ22331 model

The parameters of the BJ22331 HPE model were calculated to be:

$$y(t) = \frac{(0.09235q^{-1} - 0.0916q^{-2})}{(1 - 2.842q^{-1} + 2.688q^{-2} - 0.8455q^{-3})}u(t) + \frac{(1 - 0.9935q^{-1} + 0.3053q^{-2})}{(1 - 2.726q^{-1} + 2.463q^{-2} - 0.7365q^{-3})}e(t)$$



4.4.2.6 State space model identification

The results of the identification of different State-Space models are listed in Table 4-10. The second and third order models were not that accurate, but an eight order model proved to be quite accurate. The same settings were used in this section as those used in section 4.4.1.6.

Table 4-10

Name	n	Fit
N4S2	2	92.51%
N4S3	3	98.65%
N4S8	8	99.71%

The inability of the second order SS model to capture the dynamical behaviour of the system is illustrated in Figure 4-58.

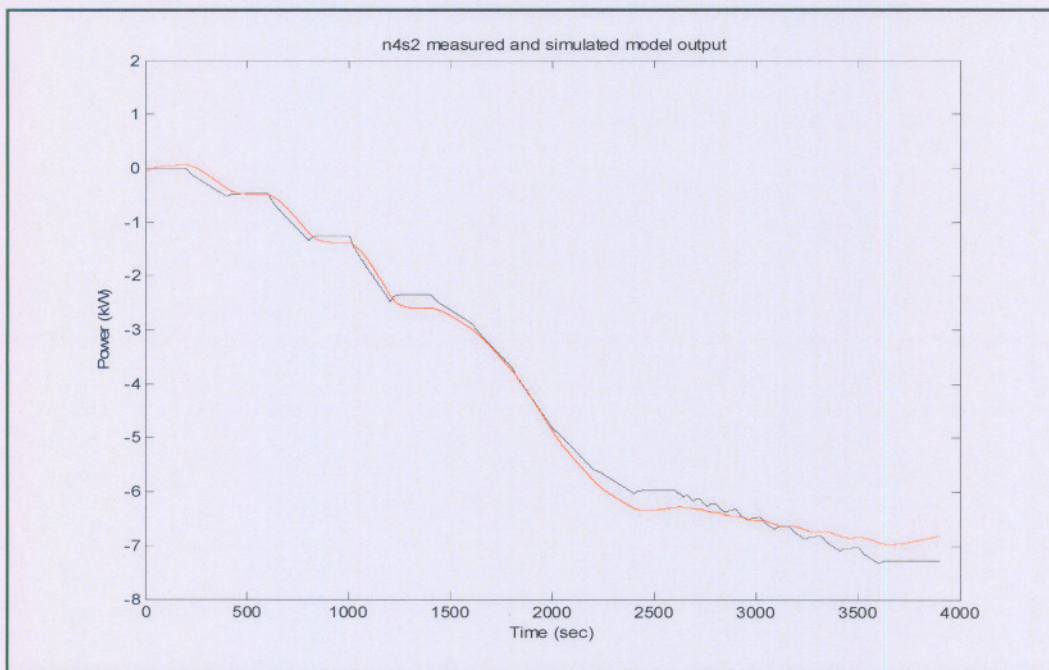


Figure 4-58: The simulated output of the n4s2 model (red) and the measured output



The second order model is too inaccurate and the rest of the analysis is not completed. The more accurate third order model was calculated. The simulated output of this model is illustrated in Figure 4-59.

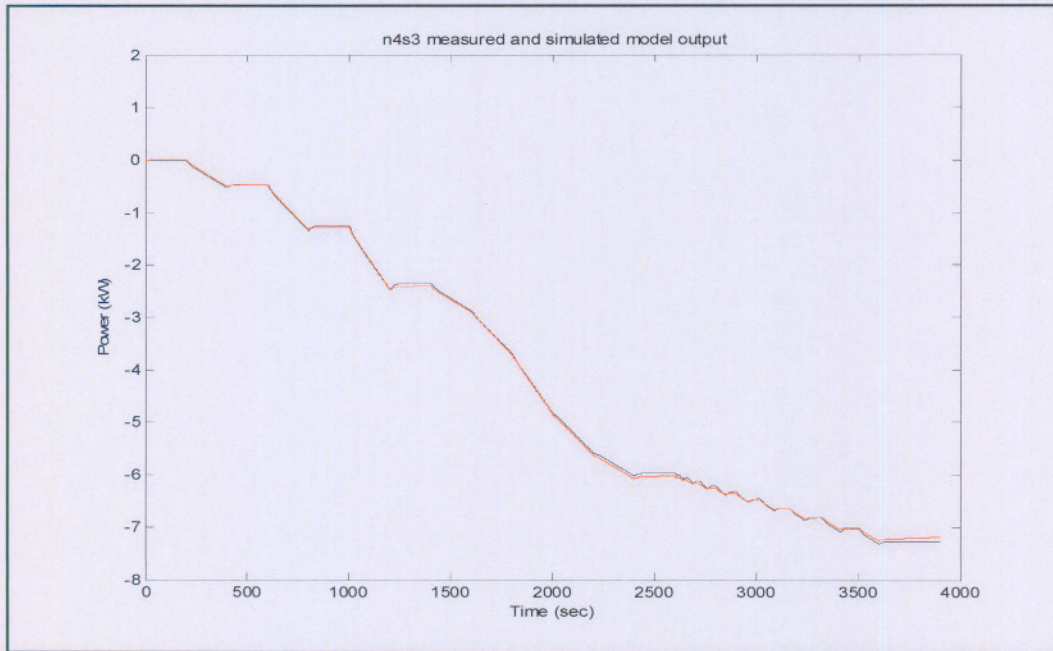


Figure 4-59: The simulated output of the n4s3 model (red) and the measured output

The partial view of the n4s3 model reveals that the model does not describe the dynamical behaviour of the system very accurately. The auto-correlation function of the error has the form of the auto-correlation of white noise. The cross-correlation function of the input and the error is not totally uncorrelated, but settles nicely inside the confidence interval.

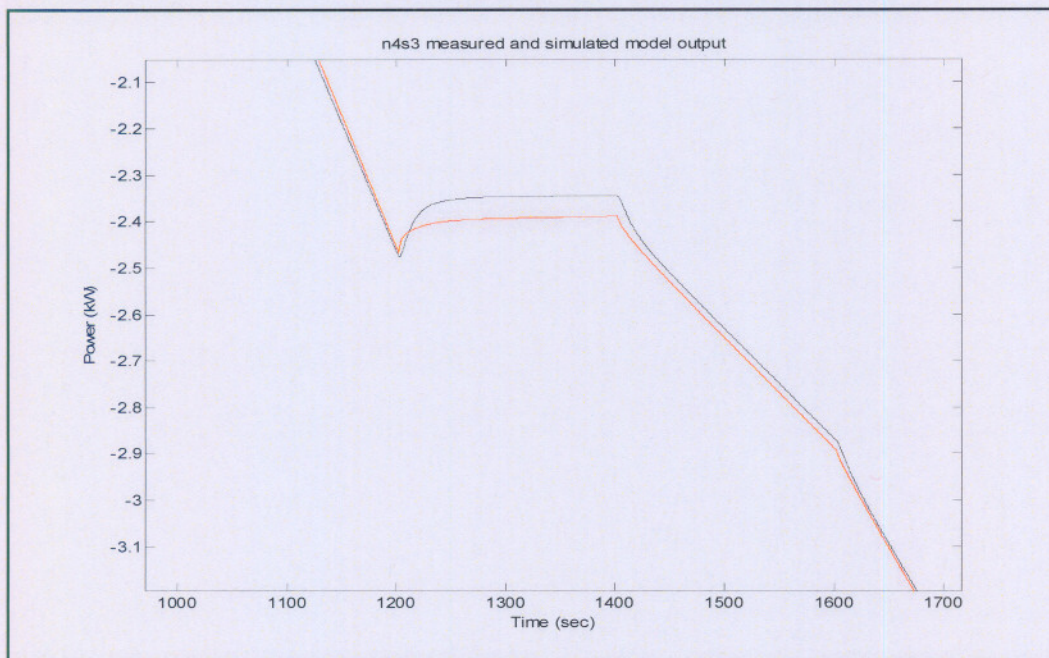


Figure 4-60: The partial view of the simulated output of the n4s3 model

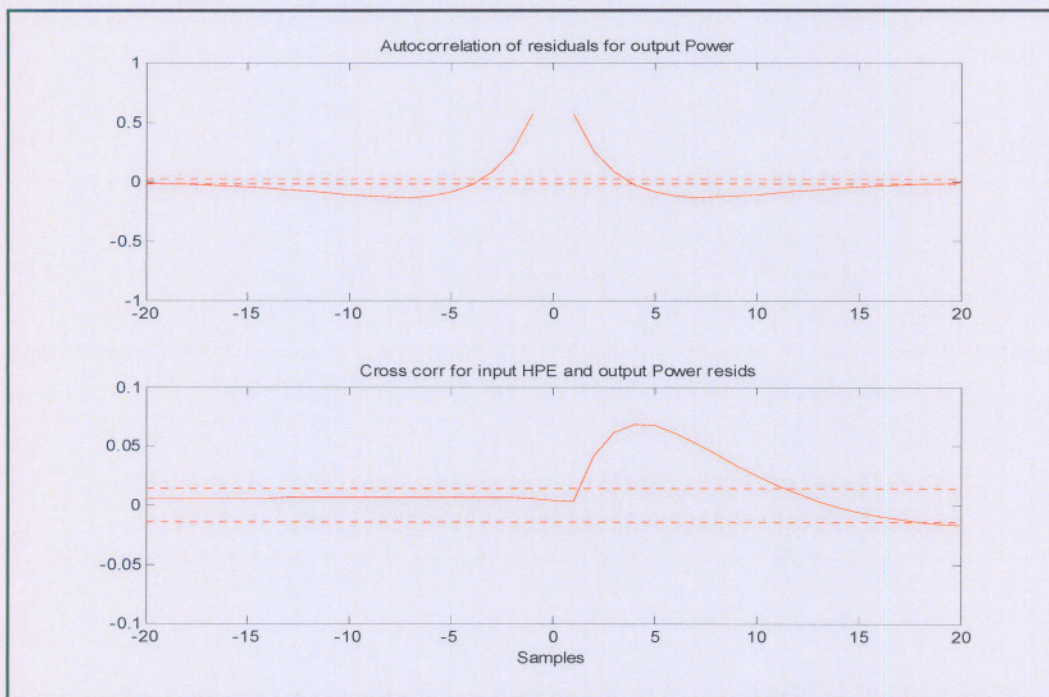


Figure 4-61: The residuals of the n4s3 model



4.4.2.7 Validation of best model for High-Pressure extraction system identification

The best model to use for the simulation and prediction of the HPE process is not as easy to select as for the LPI process. The best model according to fit is the BJ22331 model, but this model's residuals are not ideal. The ARX221 and AMX2221 model have better residuals. The AMX2221 model would be a better model when implemented, because the noise is partially modelled in this model, and will perform better in noisy environments.

4.4.3 Bypass valve regulation system identification

4.4.3.1 Bypass valve regulation simulation

The constraints of the physical system were taken into account as well as the criterion that the input signal must meet, and the following input signal was chosen:



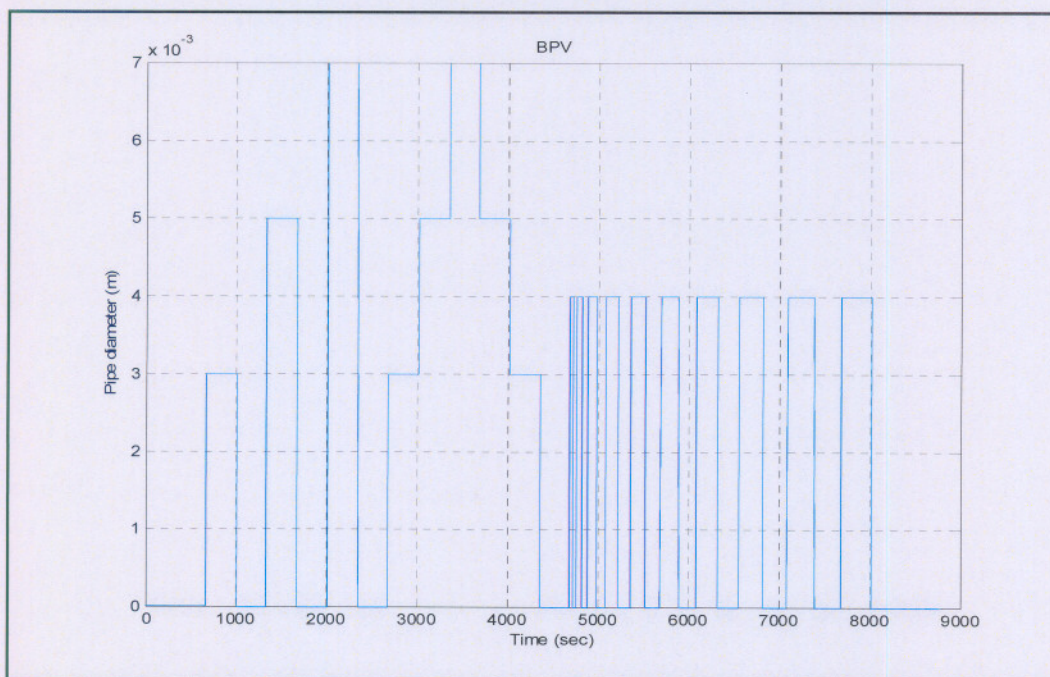


Figure 4-62: The input that was used to record the data for the BPV

This input value is the diameter of the pipe that is inserted. The area of the pipe that the gas flows through increases quadratic. This would lead to a non-linear input-output relationship. This input is thus converted to an input that describes the input as a value that describes the area of the pipe. The input that is used in identification after the transformation is illustrated in Figure 4-63. The best input signal must be positive and negative, but this is not possible for this system as the physical system does not allow this.



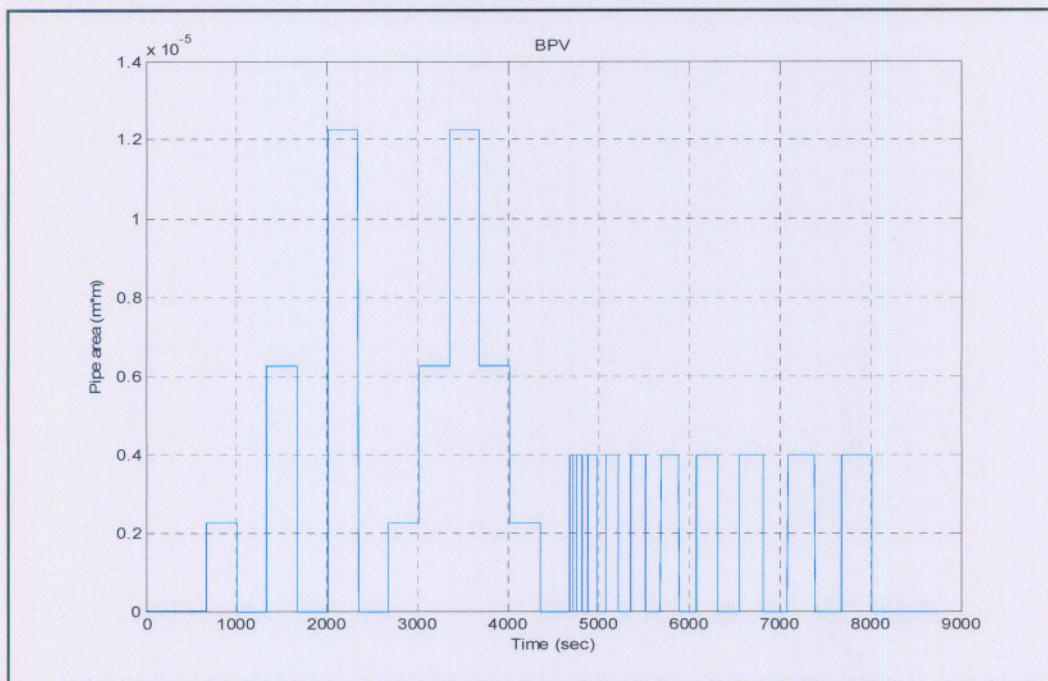


Figure 4-63: The input that was used for identification

The output that was recorded is illustrated in Figure 4-64.

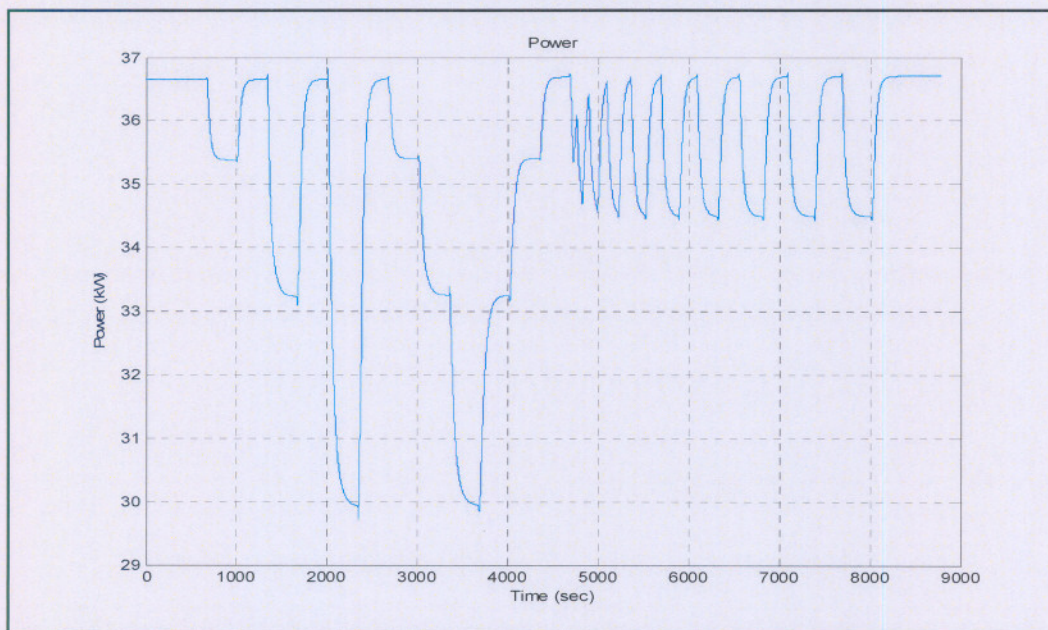


Figure 4-64: The output that was recorded



System identification of the PBMM

In Figure 4-64 it is clear that the system was always inside the parameters for normal power mode operation. In Figure 4-64 the offset of 36.6 kW is obvious. This offset must be subtracted to make the data suitable for identification. The output that is then obtained after the offset is subtracted is illustrated in Figure 4-65.

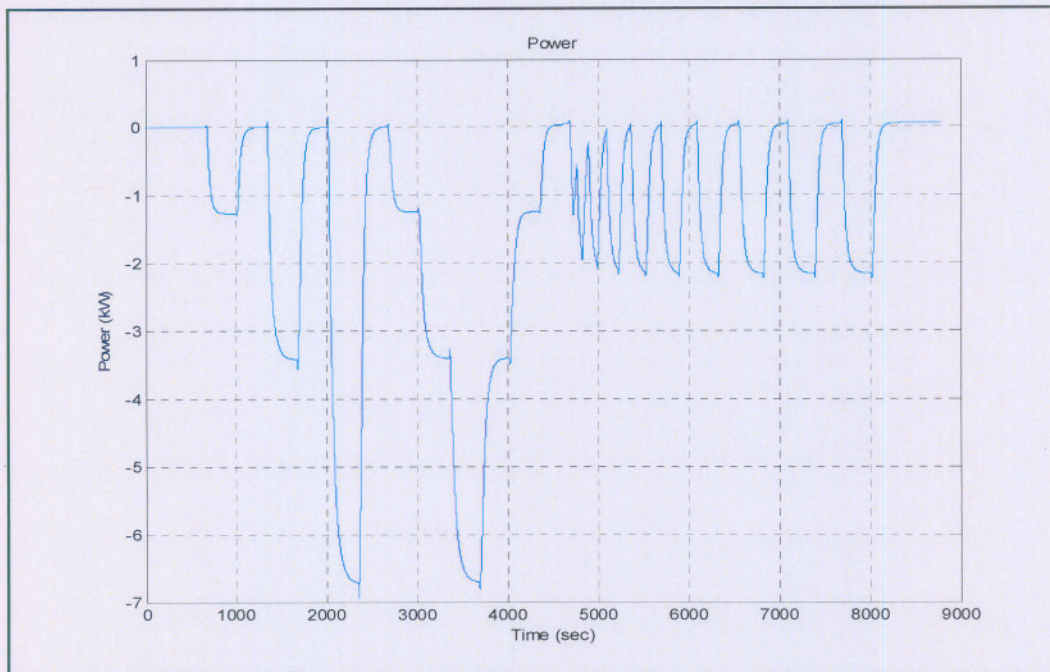


Figure 4-65: The output that was used in identification

There are no other problems with the input or output data and thus the data described in Figure 4-63 and Figure 4-65 is used in the following sections to identify different models:



4.4.3.2 ARX model identification

The first parameter that must be estimated before any models could be identified is the value of k . When the input (Figure 4-63) and the output (Figure 4-65) of the system are investigated the value of k for this process is one. This means that it takes an input one sample time to have an effect on the output.

The different second and third order ARX models could now be calculated for this process. The same settings were used in this section as those used in section 4.4.1.2.

The results of the second and third order ARX models that were calculated for the BPV process are listed in Table 4-11.

Table 4-11

Name	n_a	n_b	Fit
ARX231	2	3	91.23%
ARX221	2	2	90.92%
ARX331	3	3	90.92%
ARX321	2	2	90.26%

The two best models that will be evaluated in this section are the ARX221 and ARX231 models. The first model that is evaluated is the ARX221 model. The simulated output of the model is illustrated in Figure 4-66. In this figure it is obvious that the second order model is unable to capture all the dynamical behaviour of the process. This model especially suffers when the input changes between levels. This is because the BPV process is not totally linear [3].



The residuals of this model are illustrated in Figure 4-67. The model has a good auto- and cross-correlation. The inability of the model to describe the dynamical behaviour makes it unsuitable for use. The parameters that were calculated for this model were:

$$(1 - 1.967q^{-1} + 0.9672q^{-2})y(t) = (-2060q^{-1} + 1974q^{-2})u(t) + e(t)$$

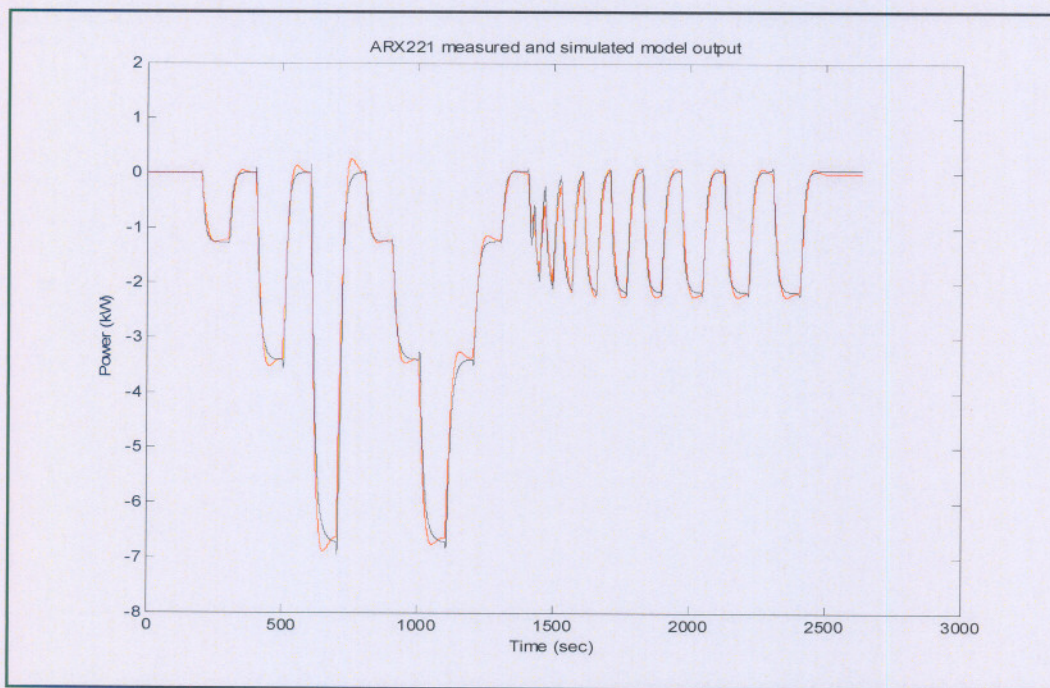


Figure 4-66: The simulated output of the ARX221 model (red) and the measured output

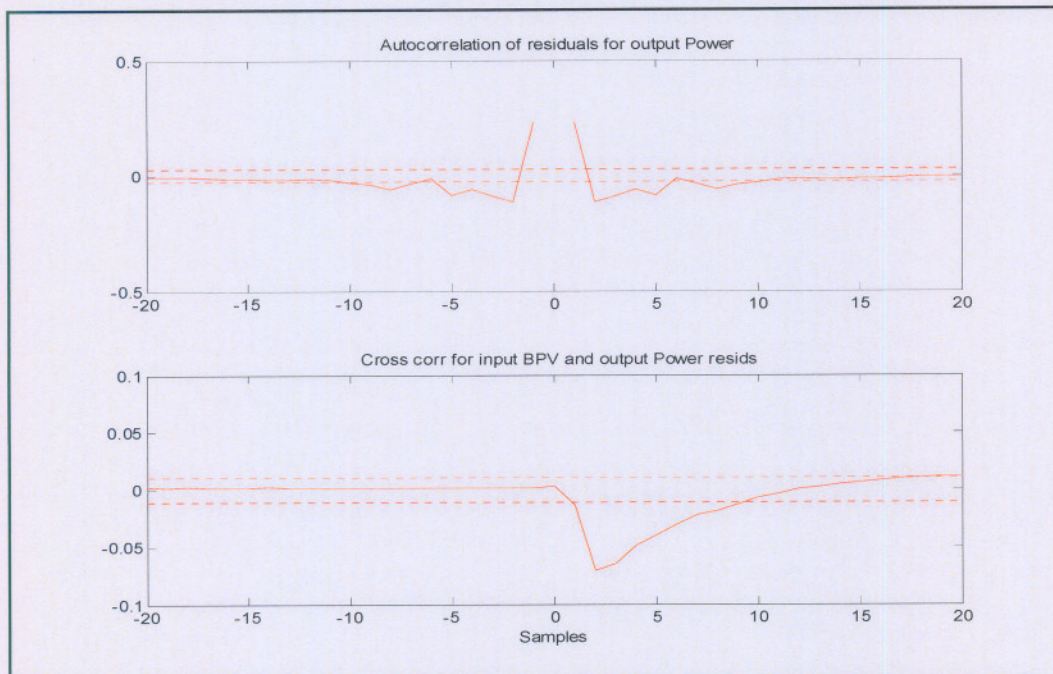


Figure 4-67: The residuals of the ARX221 model

The best ARX model according to Table 4-11 is the ARX231 model. The simulated output of this model (Figure 4-68) however reveals that this model causes spikes at the transition of inputs. These spikes are too large and could cause problems in a control system. The rest of the analysis on this model will not be preformed.



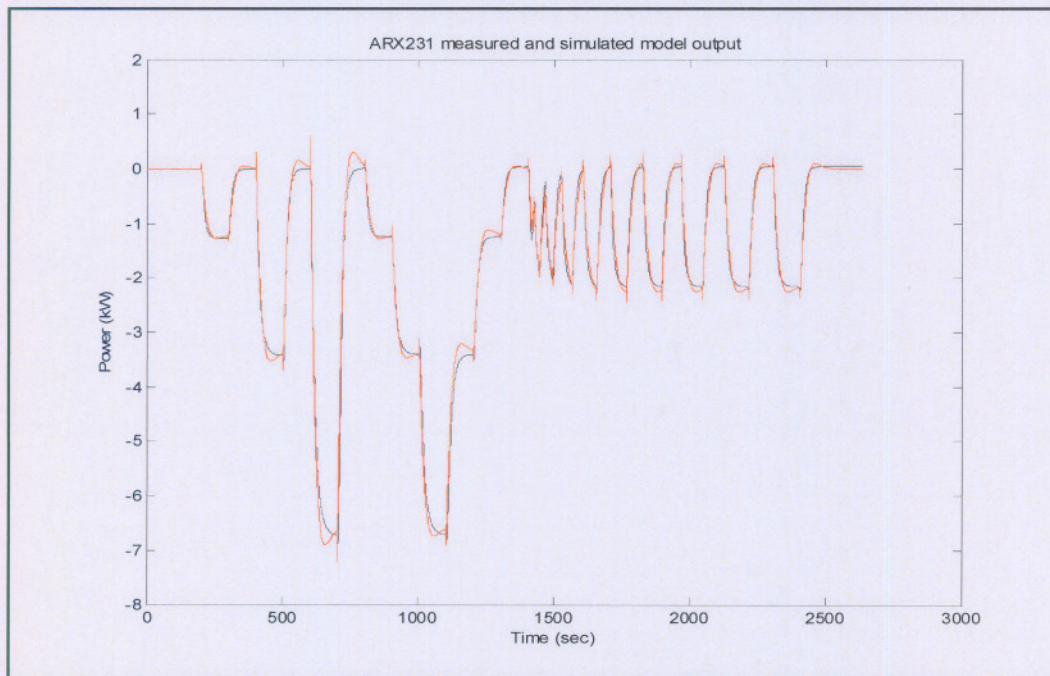


Figure 4-68: The simulated output of the ARX231 model (red) and the measured output

4.4.3.3 ARMAX model identification

Different second and third order ARMAX models were calculated for the BPV process. The same settings were used in this section as used in section 4.4.1.3 and 4.4.2.3. The results of the identification process are listed in Table 4-12.



Table 4-12

Name	n_a	n_b	n_c	Fit
AMX3331	3	3	3	92.81%
AMX3321	3	3	2	92.81%
AMX2221	2	2	2	92.77%
AMX2321	2	3	2	92.76%
AMX2231	2	2	3	92.76%
AMX2331	2	3	3	92.76%
AMX3221	3	2	2	92.69%
AMX3231	3	2	3	92.69%

The ARMAX model that best describes the dynamical behaviour according to Table 4-12 is the AMX3331. The AMX3331 model will be evaluated second in this section. The lowest order model that achieved good performance, the AMX2221 model, will be evaluated first.

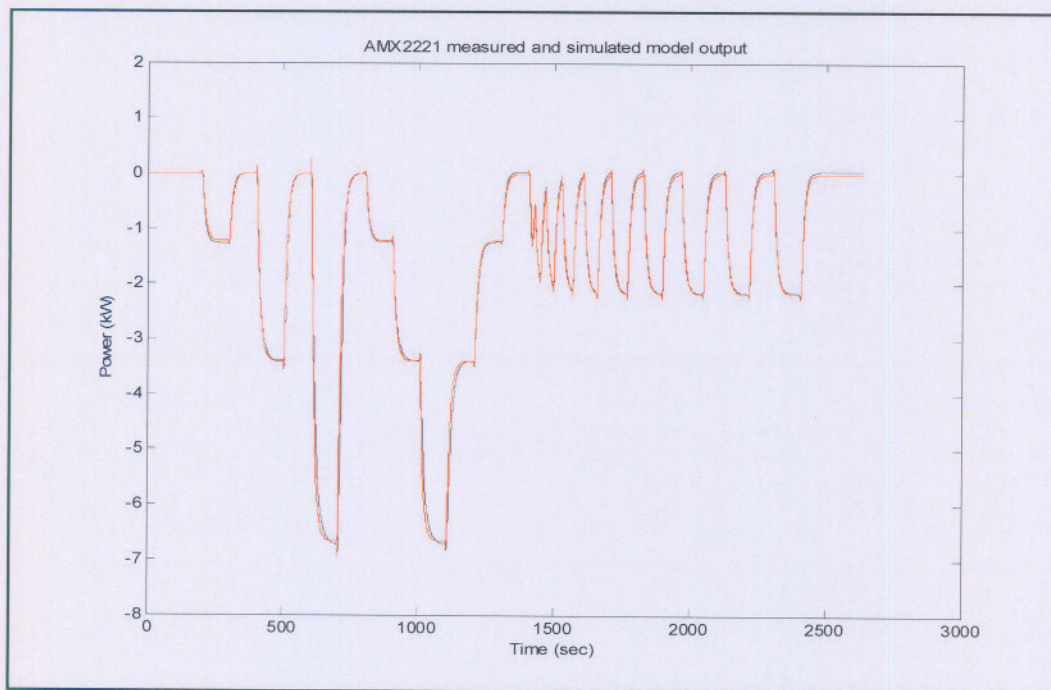


Figure 4-69: The simulated output of the AMX2221 model (red) and the measured output



The simulated output of the AMX2221 model (Figure 4-69) verifies that this model captures the dynamical behaviour of this non-linear process quite well for a second order model. The residual analysis of the AMX2221 model (Figure 4-70) confirms that this is a good model. The auto-correlation function confirms that the error of the model is white noise and contains no more information. The cross-correlation confirms that there is not correlation between the error and the input signal. The parameters that were calculated for the AMX2221 model were:

$$(1 - 1.798q^{-1} + 0.8023q^{-2})y(t) = (3932q^{-1} - 4757q^{-2})u(t) + (1 - 0.02036q^{-1} - 0.2813q^{-2})e(t)$$

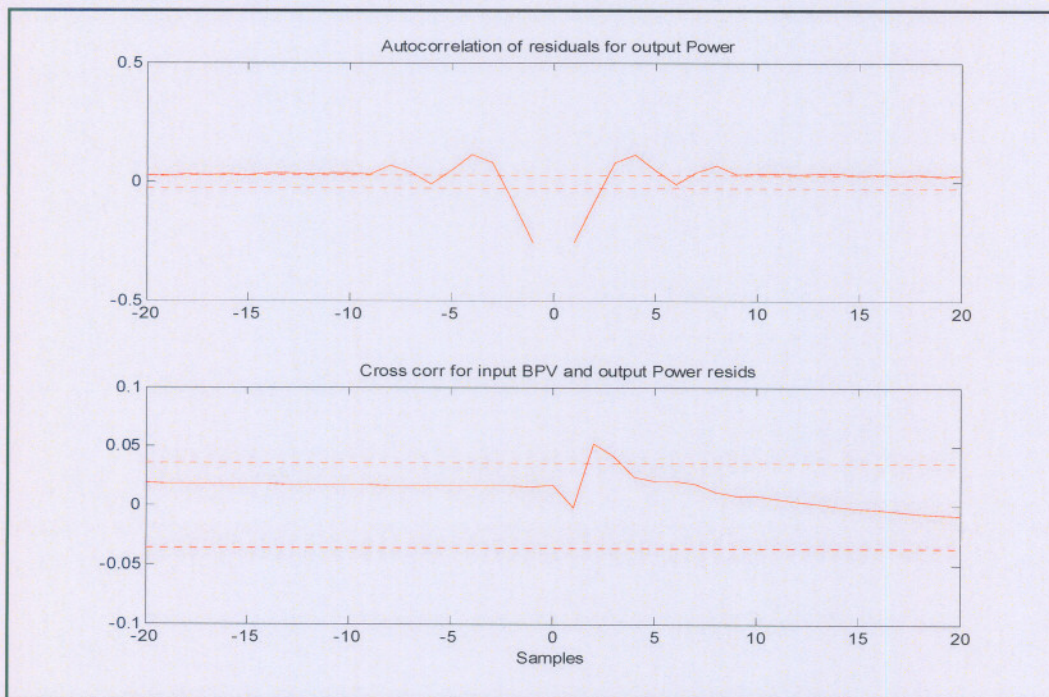


Figure 4-70: The residuals of the AMX2221 model

The simulated output of the AMX3331 model is illustrated in Figure 4-71. A more descriptive figure, Figure 4-72, illustrates the difference between the simulated output of the model and the measured output.



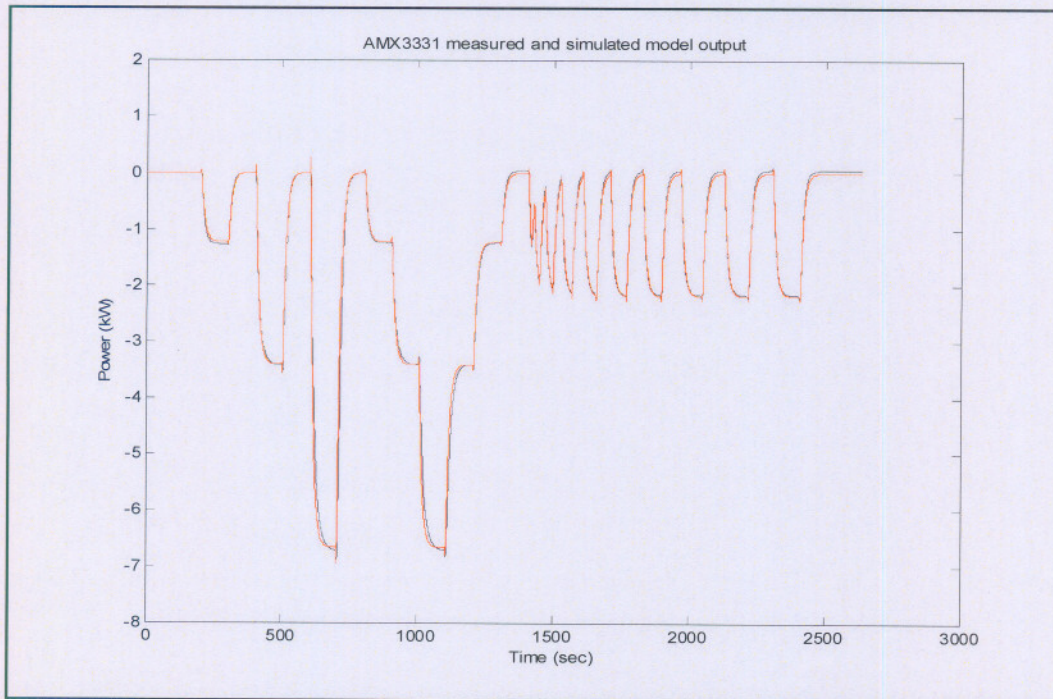


Figure 4-71: The simulated output of the AMX3331 model (red) and the measured output

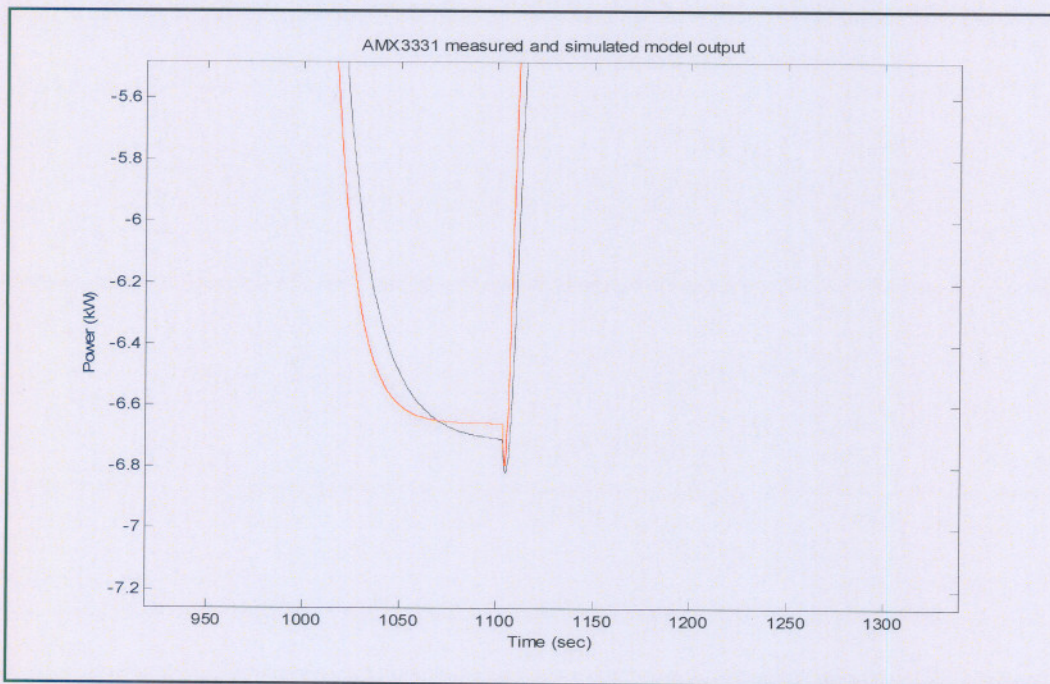


Figure 4-72: The partial view of Figure 4-71



The residuals of the AMX3331 model are illustrated in Figure 4-73. The autocorrelation function of the error reveals that the error is a white noise process. This means that there is no information that was not modelled. The cross-correlation between the error and the input reveals that they are uncorrelated.

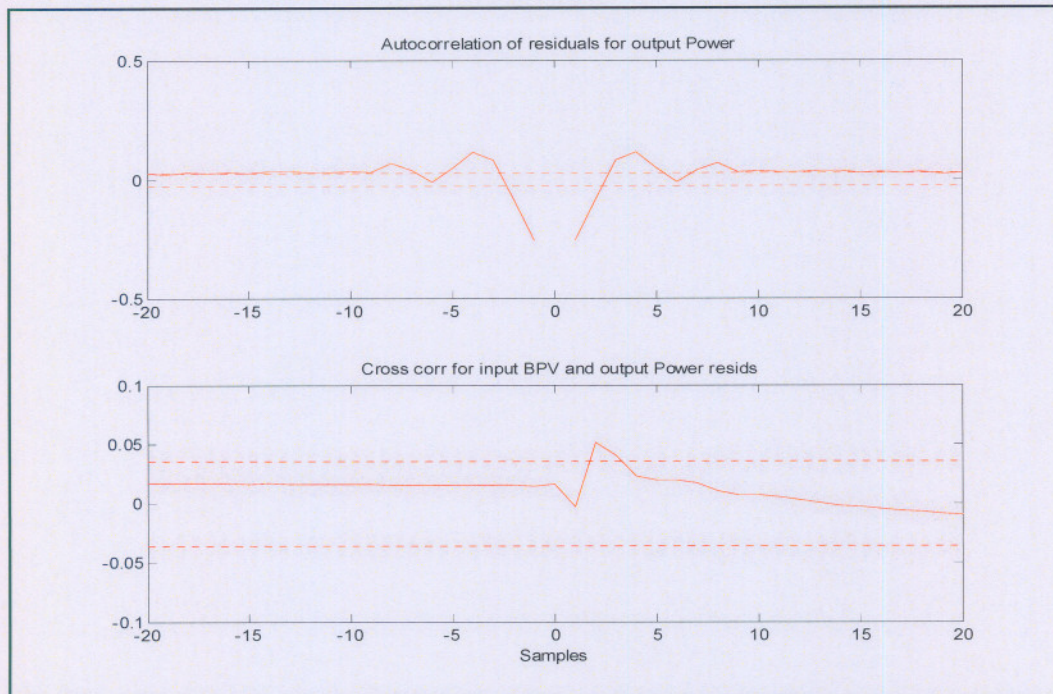


Figure 4-73: The residuals of the AMX3331 model

The parameters of the AMX3331 BPV model are:

$$\begin{aligned} (1 - 0.797q^{-1} - 0.9952q^{-2} + 0.8018q^{-3})y(t) = & (3948q^{-1} - 824.9q^{-2} - 4774q^{-3})u(t) + \\ & (1 + 1.02q^{-1} + 0.2995q^{-2} + 0.2798q^{-3})e(t) \end{aligned}$$

The AMX3321 model is the best model in this section, because it describes the output of the system more accurately. The models have almost identical residuals; this means the system that best describes the system in the time domain is the best.



4.4.3.4 Output Error model identification

The different Output Error models that were calculated and their performance are listed in Table 4-13. The same settings were used as those described in section 4.4.1.4.

Table 4-13

Name	n_b	n_f	Fit
OE331	3	3	92.81%
OE221	2	2	92.77%
OE321	2	2	92.77%
OE231	2	3	92.69%

The two best models will be evaluated in this section. The first model that will be evaluated is the OE221 model. The simulated output of this model is illustrated in Figure 4-74. The partial view of the same model is illustrated in Figure 4-75. The partial view illustrates that the model captures the dynamical behaviour quite well.

The residual analysis (Figure 4-76) reveals that this model has an exceptional good cross-correlation function. The input and the error are totally uncorrelated. The auto-correlation is not ideal. The fact that the error of the model is not white noise, suggests that some unmodeled information is left in the error.



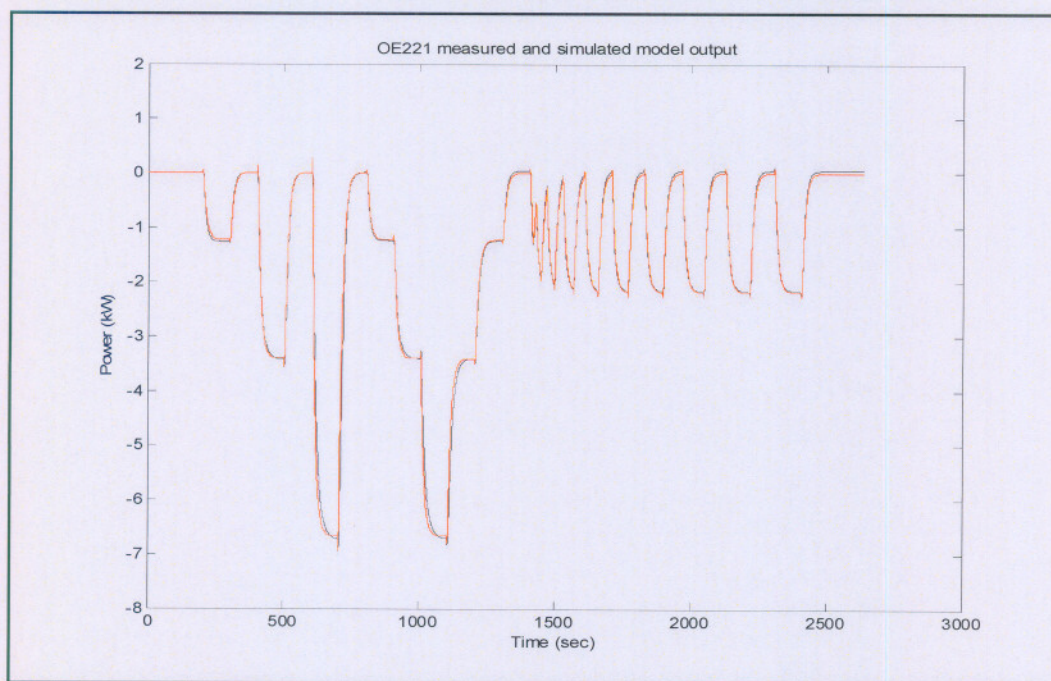


Figure 4-74: The simulated output of the OE221 model (red) and the measured output

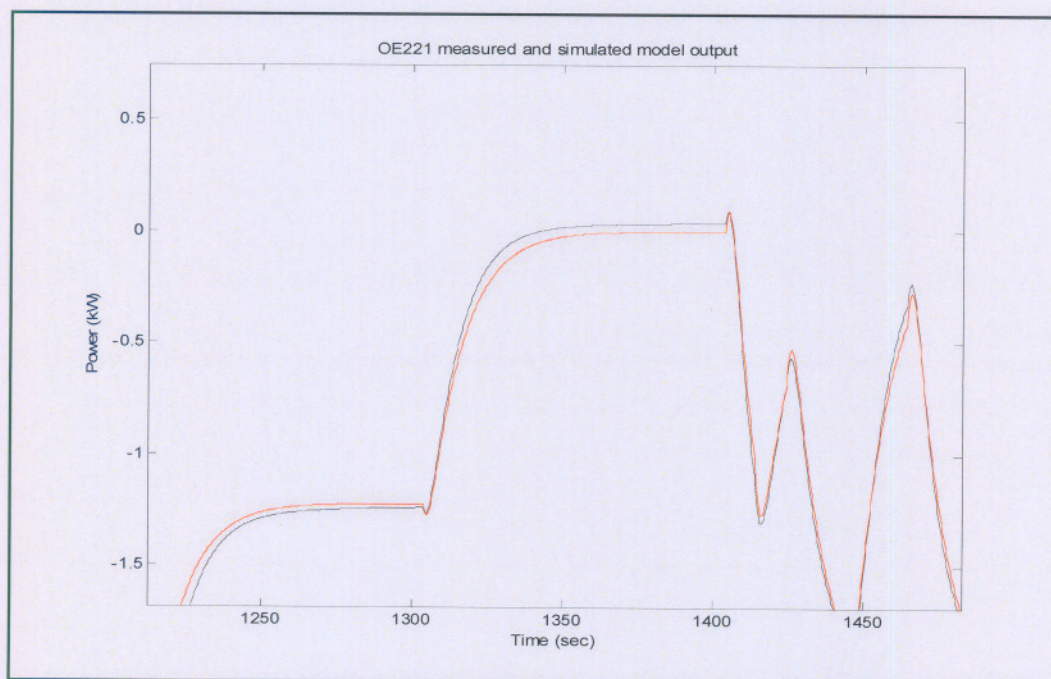


Figure 4-75: The partial view of Figure 4-74



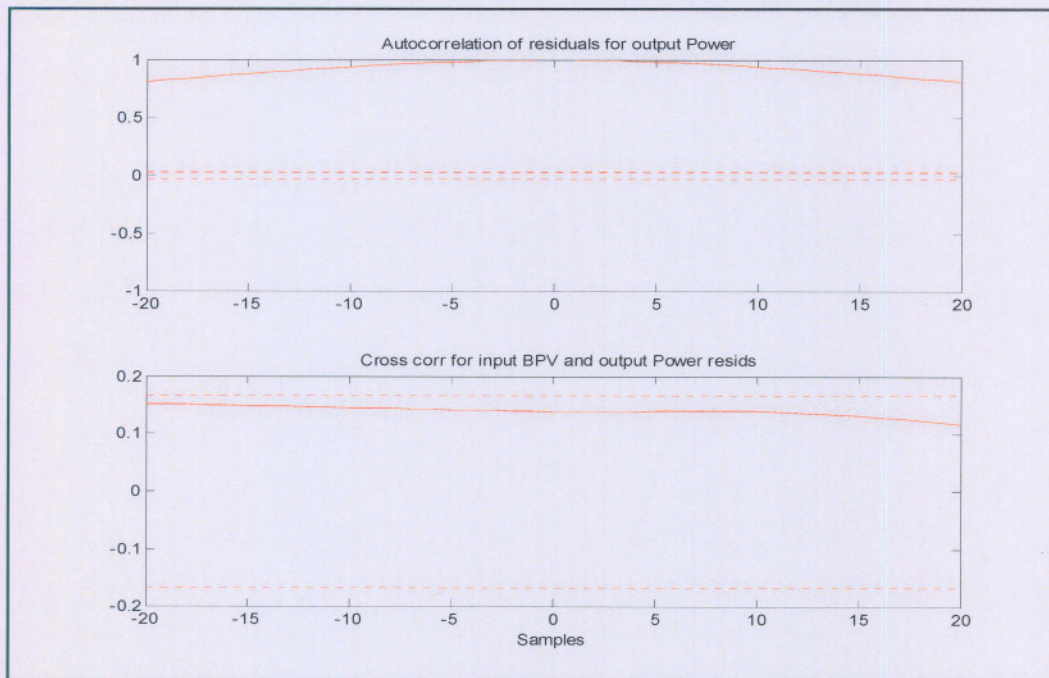


Figure 4-76: The residuals of the OE221 BPV model

The parameters of the OE221 model are:

$$y(t) = \frac{(3932q^{-1} - 4757q^{-2})}{(1 - 1.798q^{-1} + 0.8023q^{-2})} u(t) + e(t)$$

The next model that is evaluated is the OE331 model. This model has the best fit of all the OE models that were estimated. The simulated output of the OE331 model is illustrated in Figure 4-77. The partial view that reveals that the OE331 model described the process very accurately is illustrated in Figure 4-78. The residuals of the OE331 (Figure 4-79) is very similar to the residuals of the OE221 model and the same conclusions could be made. The parameters of the OE331 BPV model are:

$$y(t) = \frac{(3948q^{-1} - 824q^{-2} - 4774q^{-3})}{(1 - 0.797q^{-1} - 0.9952q^{-2} + 0.8018q^{-3})} u(t) + e(t)$$



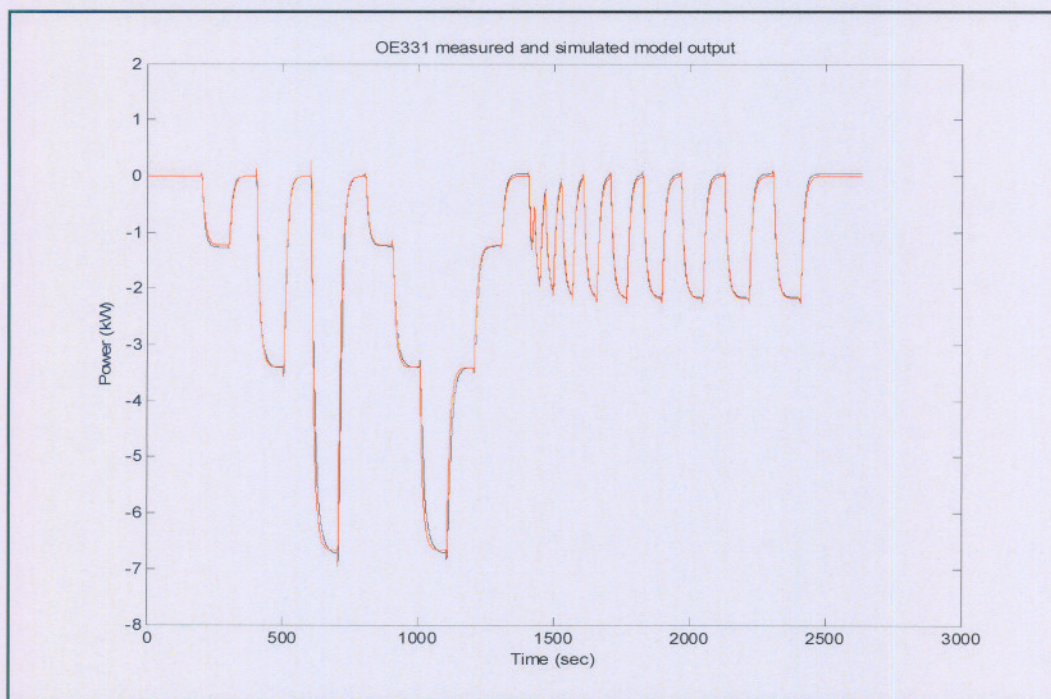


Figure 4-77: The simulated output if the OE331 model (red) and the measured output

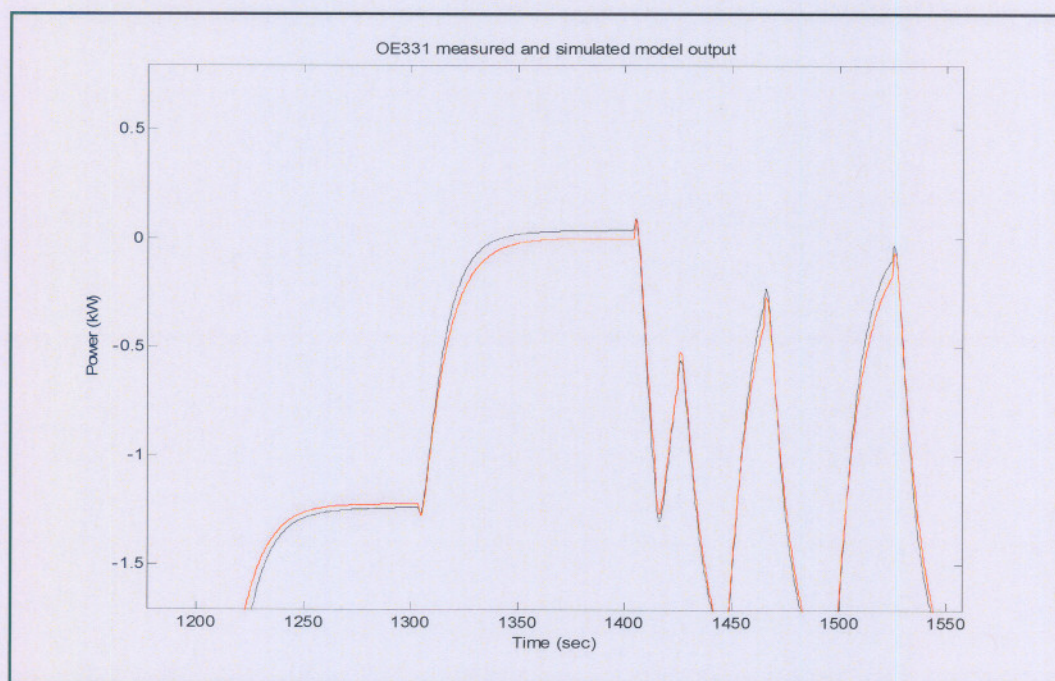


Figure 4-78: The partial view of the OE331 model



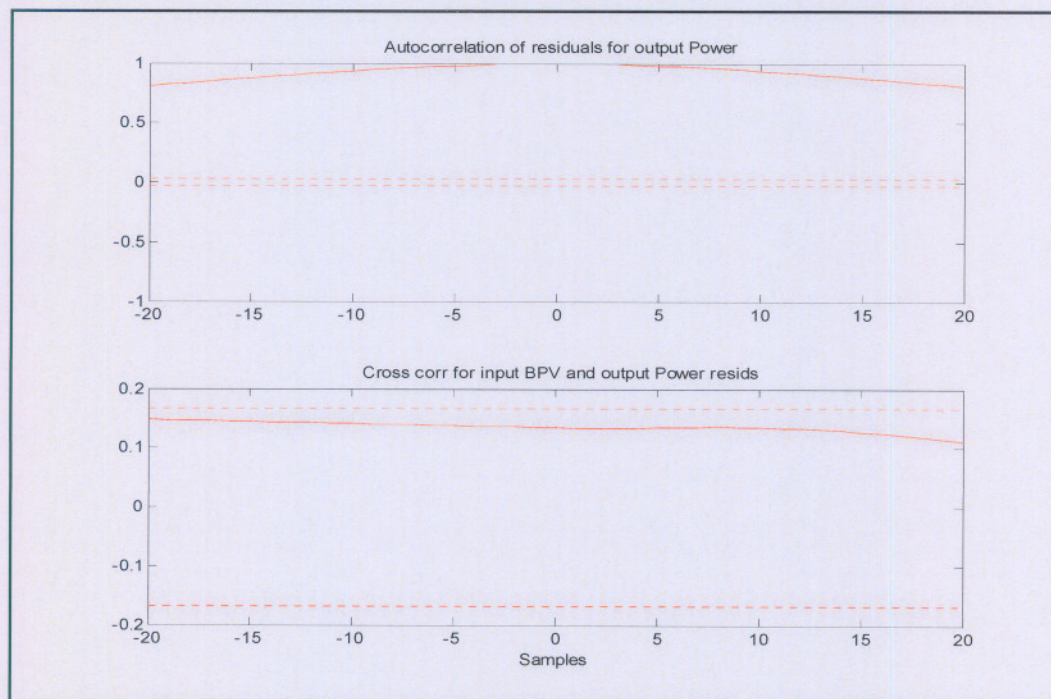


Figure 4-79: The residuals of the OE331 model

The OE models in this section are not suited for prediction or simulation.

4.4.3.5 Box-Jenkins model identification

In this section 15 different second and third order Box-Jenkins models were estimated to model the BPV process. The different models and their performance are listed in Table 4-14.



Table 4-14

Name	n_b	n_c	n_d	n_f	Fit
BJ33331	3	3	3	3	92.81%
BJ32331	3	2	3	3	92.81%
BJ33231	3	3	2	3	92.81%
BJ22221	2	2	2	2	92.77%
BJ2332	2	3	3	2	92.77%
BJ2232	2	2	3	2	92.77%
BJ2322	2	3	2	2	92.77%
BJ33321	3	3	3	2	92.76%
BJ3232	3	2	3	2	92.76%
BJ3322	3	3	2	2	92.76%
BJ3222	3	2	2	2	92.76%
BJ23331	2	3	3	3	92.69%
BJ2233	2	2	3	3	92.69%
BJ2323	2	3	2	3	92.69%
BJ2223	2	2	2	3	92.69%

The best model according to Table 4-14 is the BJ33331 model. This model will be evaluated later in this section. The model that will be evaluated first in this section is the lowest order model, the BJ22221 model. This model has the least amount of adjustable parameters, but almost describes the process as well as some of its higher order counterparts.

The simulated output of the BJ22221 model is illustrated in Figure 4-80. The partial view of the simulated output of the BJ22221 model is illustrated in Figure 4-81. The partial view illustrates that the model describes the behaviour of the process very well. The residuals of the BJ22221 model are illustrated in Figure 4-82. The residuals of the model reveal that the error between the model and the process contains no more information that could be included in the model. The fact that there is little cross-correlation between the input and the error means that the error of the model does not depend on the input of the system.



The parameters of the BJ22221 model were estimated as:

$$y(t) = \frac{(3932q^{-1} - 4757q^{-2})}{(1 - 1.798q^{-1} + 0.8023q^{-2})}u(t) + \frac{(1 - 0.2671q^{-1} + 0.1942q^{-2})}{(1 - 1.941q^{-1} + 0.9416q^{-2})}e(t)$$

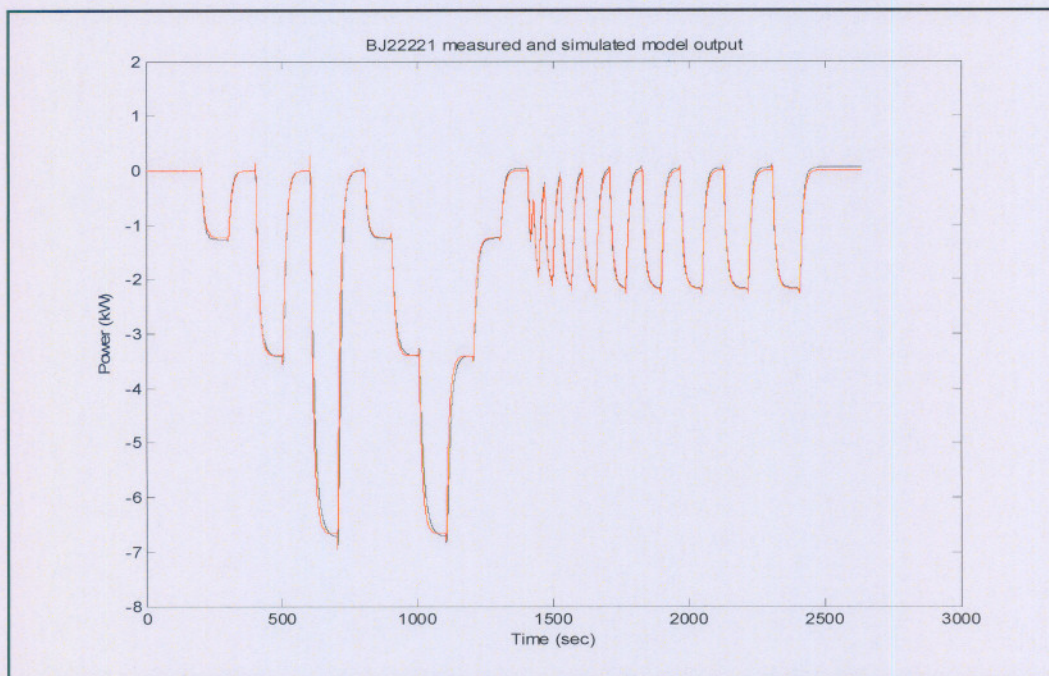


Figure 4-80: The simulated output of the BJ22221 model (red) and the measured output

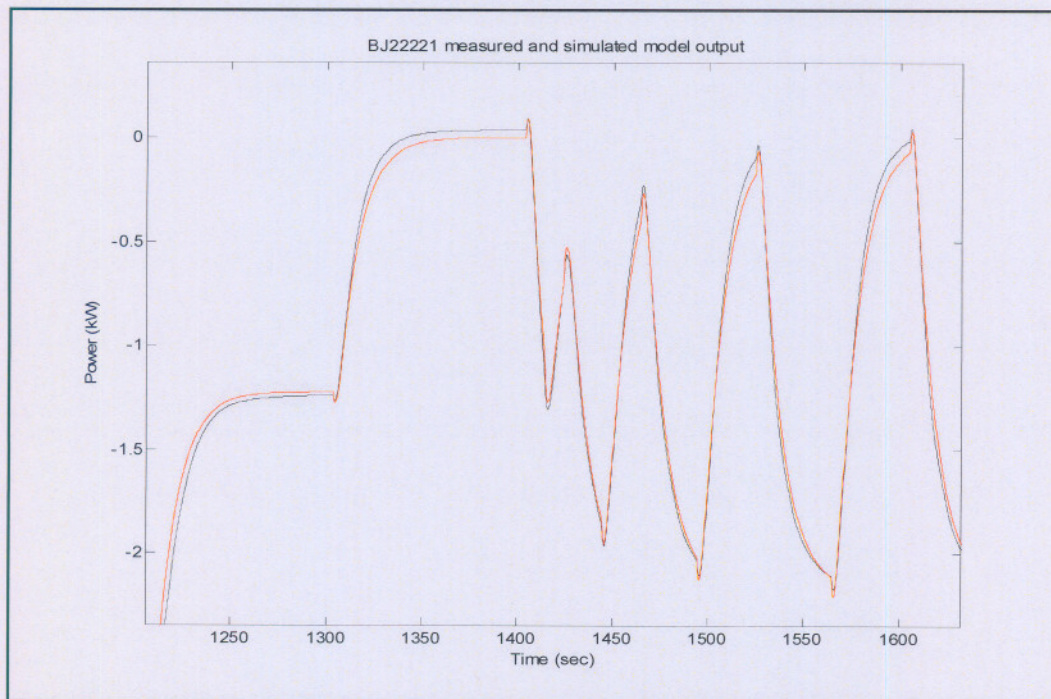


Figure 4-81: The partial view of the simulated output of the BJ22221 model

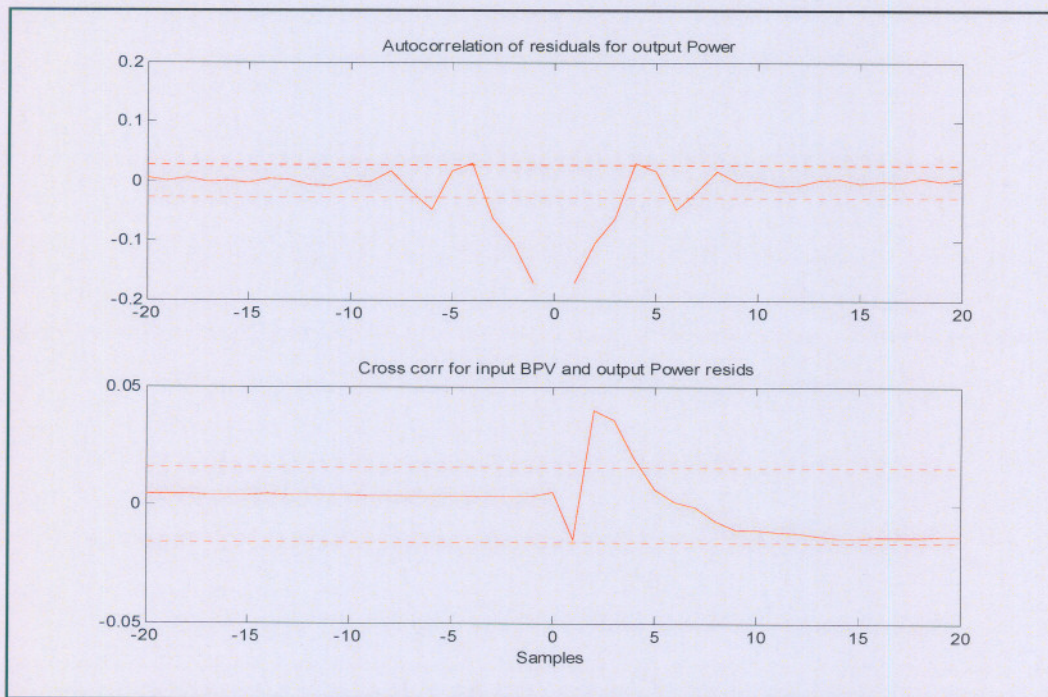


Figure 4-82: The residuals of the BJ22221 model



The simulated output of the BJ33331 model is illustrated in Figure 4-83. The partial view of Figure 4-83 is illustrated in Figure 4-84. The partial view of the simulated output reveals that even when a high frequency of changes occurs in the input, the model could predict it.

The residuals of the BJ33331 model are illustrated in Figure 4-85. The same conclusions made for the BJ22221 model could be made for the residuals of the BJ33331 model. The auto-correlation of the error is the same as the auto-correlation of white noise. The cross-correlation between the error and the input shows very little correlation. The parameters of the BJ33331 model are:

$$y(t) = \frac{(3948q^{-1} - 824.9q^{-2} - 4774q^{-3})}{(1 - 0.797q^{-1} - 0.9952q^{-2} + 0.8018q^{-3})}u(t) + \frac{(1 + 0.3637q^{-1} + 0.1948q^{-2} + 0.04805q^{-3})}{(1 - 1.357q^{-1} - 0.1924q^{-2} + 0.5504q^{-3})}e(t)$$

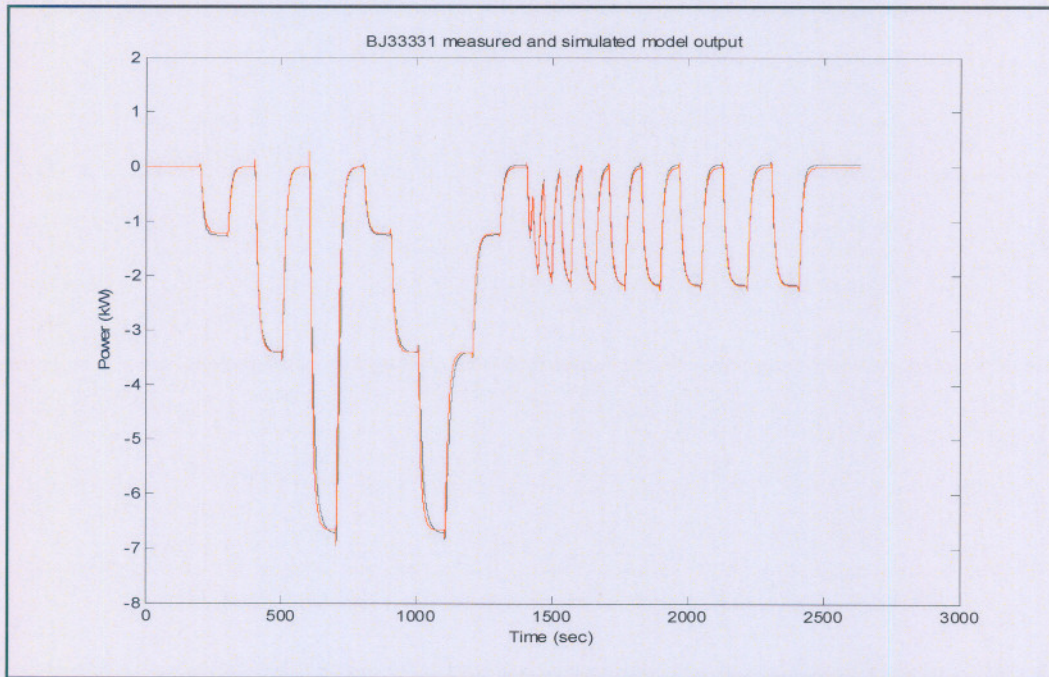


Figure 4-83: The simulated output of the BJ33331 model (red) and the measured output



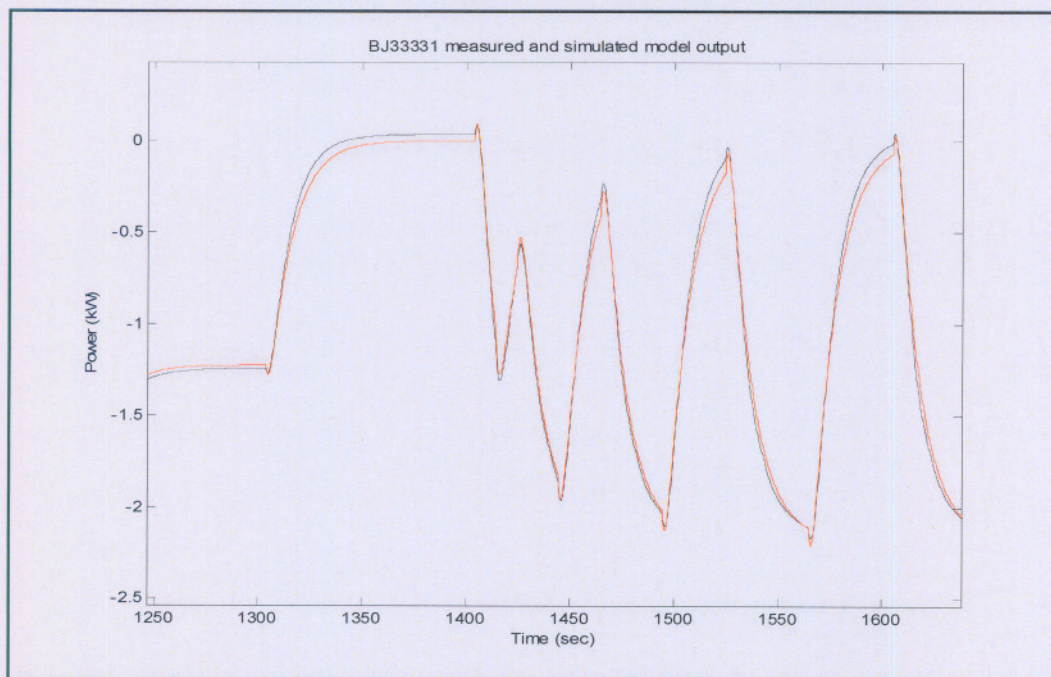


Figure 4-84: The partial view of Figure 4-83

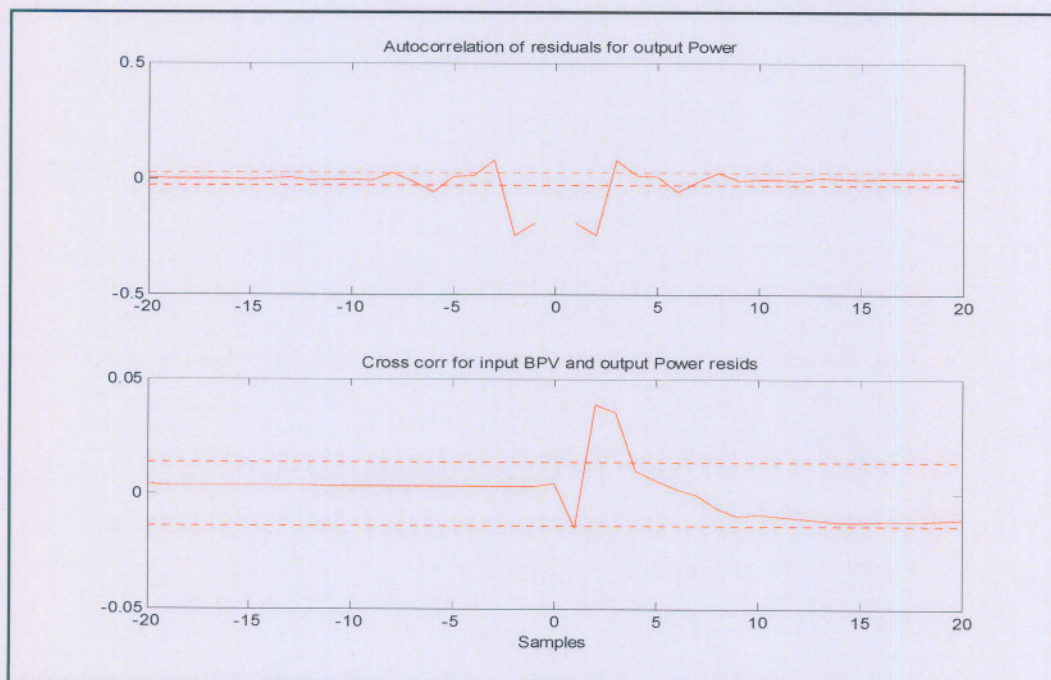


Figure 4-85: The residuals of the BJ33331 model



4.4.3.6 State Space model identification

In the last section of SISO model identification two different State-Space models were calculated to describe the BPV process. The same settings were used as in section 4.4.1.6. The performance of the two models that were calculated are listed in Table 4-15.

Table 4-15

Name	n	Fit
N4S2	2	93.77%
N4S3	3	93.69%

According to Table 4-15 the second order SS model is superior to the third order model. Both these models will be evaluated in this section.

The simulated output of the n4s2 model is illustrated in Figure 4-86. The partial view of the n4s2 model, which illustrates the accuracy of the n4s2 model, is illustrated in Figure 4-87. The cross-correlation between the input and the error is almost non-existent (Figure 4-88). The auto-correlation suggests that the error is a white noise process. The residuals discussed above conclude that this is a very good model.

The parameters of the model are:

$$\begin{aligned}
 x(t+Ts) &= \begin{bmatrix} 0.98508 & 0.01259 \\ -0.086757 & 0.8841 \end{bmatrix} x(t) + \begin{bmatrix} 30.551 \\ -743.37 \end{bmatrix} u(t) + \begin{bmatrix} 0.015957 \\ 0.900994 \end{bmatrix} e(t) \\
 y(t) &= [82.972 \quad 0.47827] x(t) + [0] u(t) + e(t) \\
 x(0) &= \begin{bmatrix} 0 \\ 0 \end{bmatrix}
 \end{aligned}$$



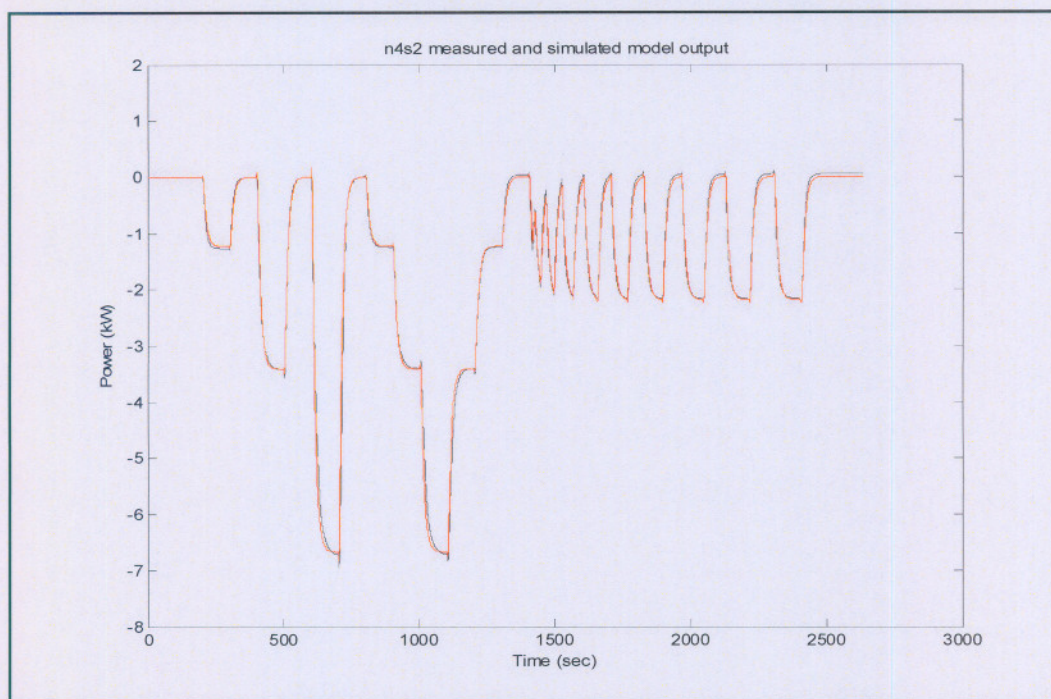


Figure 4-86: The simulated output of the n4s2 model (red) and the measured output

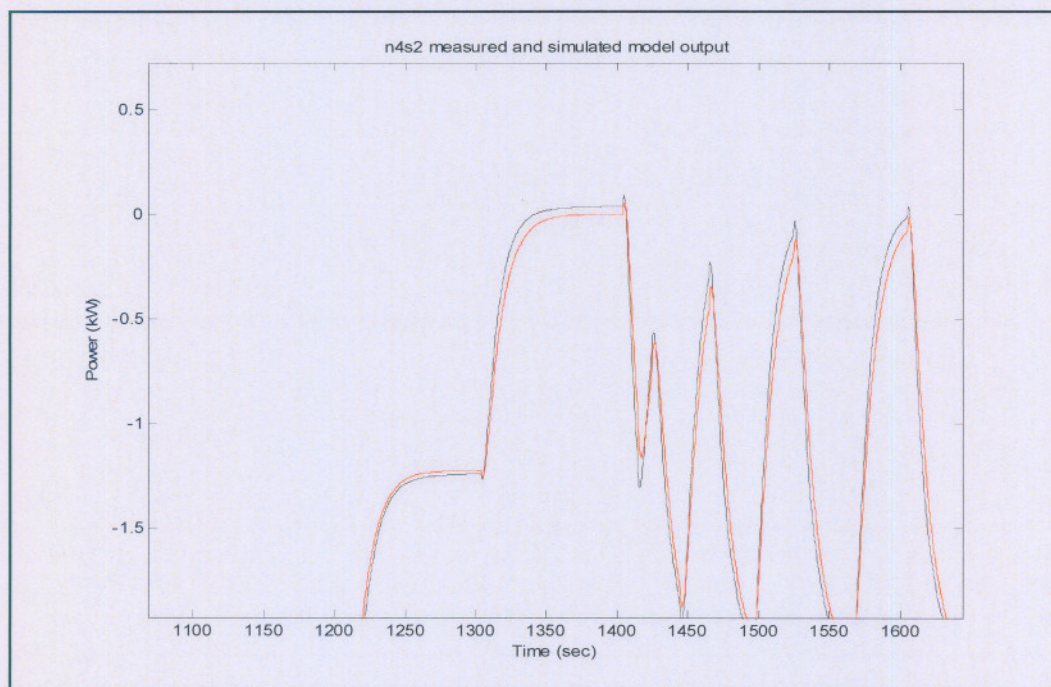


Figure 4-87: The partial view of the simulated output of the n4s2 model



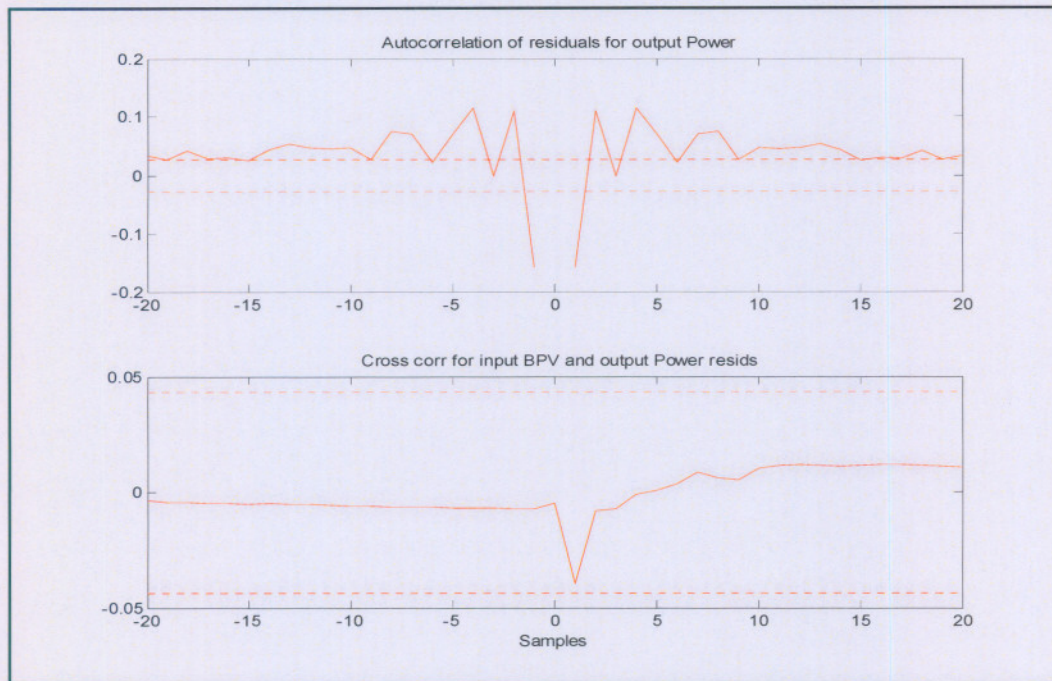


Figure 4-88: The residuals of the n4s2 model

The simulated output of the n4s3 model is illustrated in Figure 4-89. The partial view of the n4s3 model is illustrated in Figure 4-90. The cross-correlation between the input and the error is almost non-existent (Figure 4-91). The auto-correlation suggests that the error is a white noise process. The auto-correlation function of this model is not as good as the auto-correlation function of the n4s2 model. This makes the n4s2 model more superior to the n4s3 model. The parameters of the n4s3 model are:

$$x(t+Ts) = \begin{bmatrix} 0.9851 & -0.012613 & 3.16e^{-5} \\ 0.088068 & 0.819 & 0.11007 \\ 0.0031083 & -0.032772 & 0.7308 \end{bmatrix} x(t) + \begin{bmatrix} 38.355 \\ 924.97 \\ -53.476 \end{bmatrix} u(t) + \begin{bmatrix} 0.01621 \\ -0.90266 \\ -0.95747 \end{bmatrix} e(t)$$

$$y(t) = [82.974 \quad -0.4913 \quad -0.030165] x(t) + [0] u(t) + e(t)$$

$$x(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$



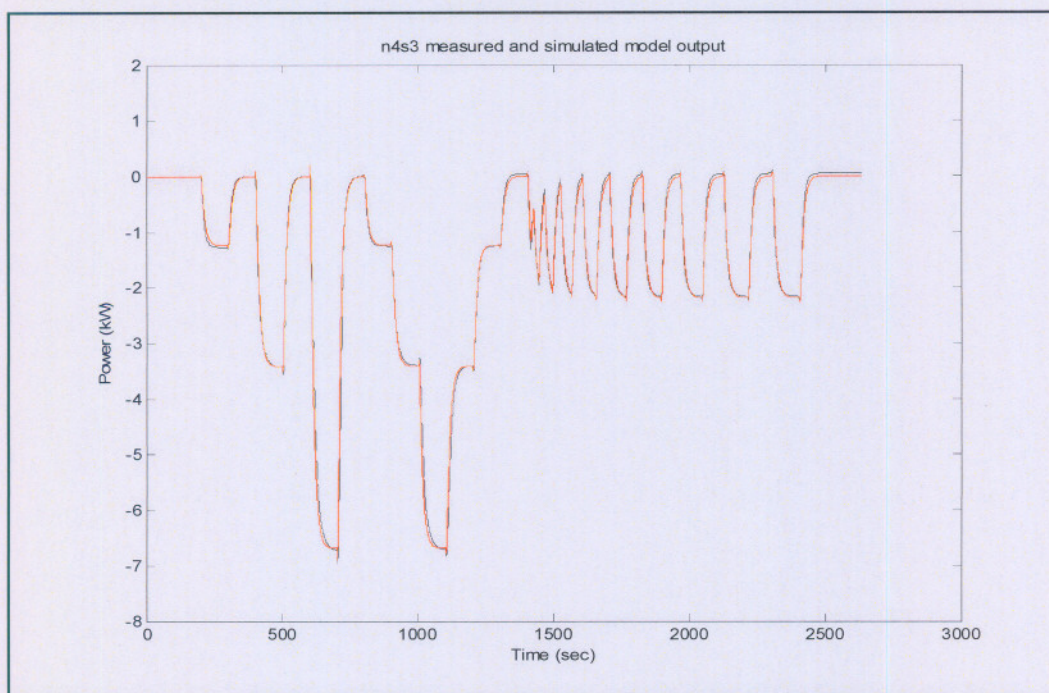


Figure 4-89: The simulated output of the n4s3 model (red) and the measured output

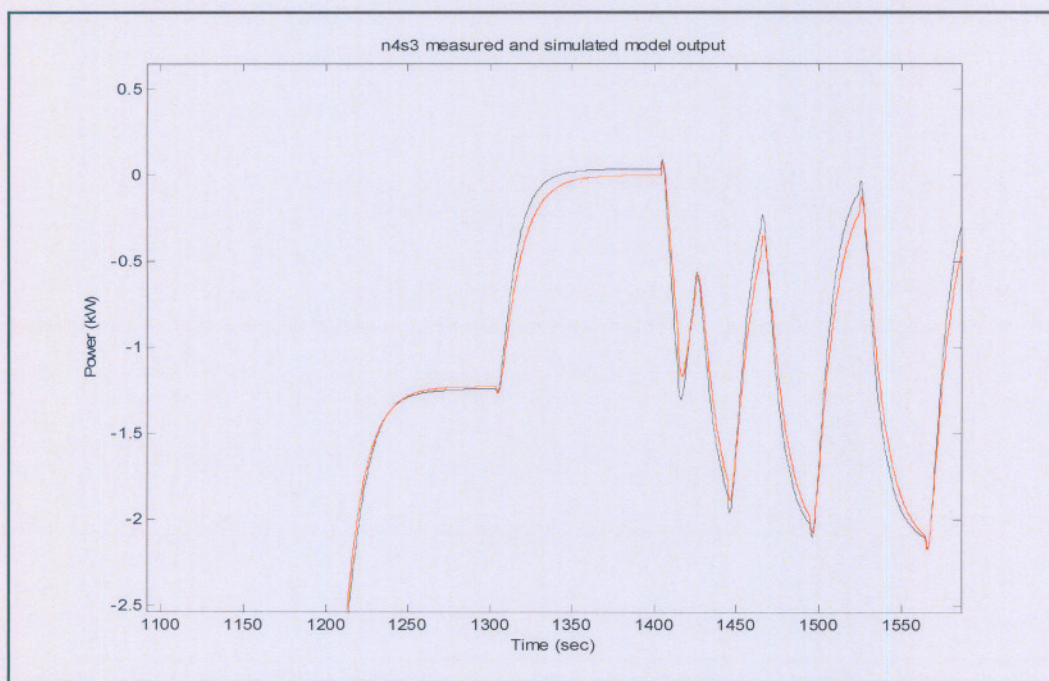


Figure 4-90: The partial view of Figure 4-89



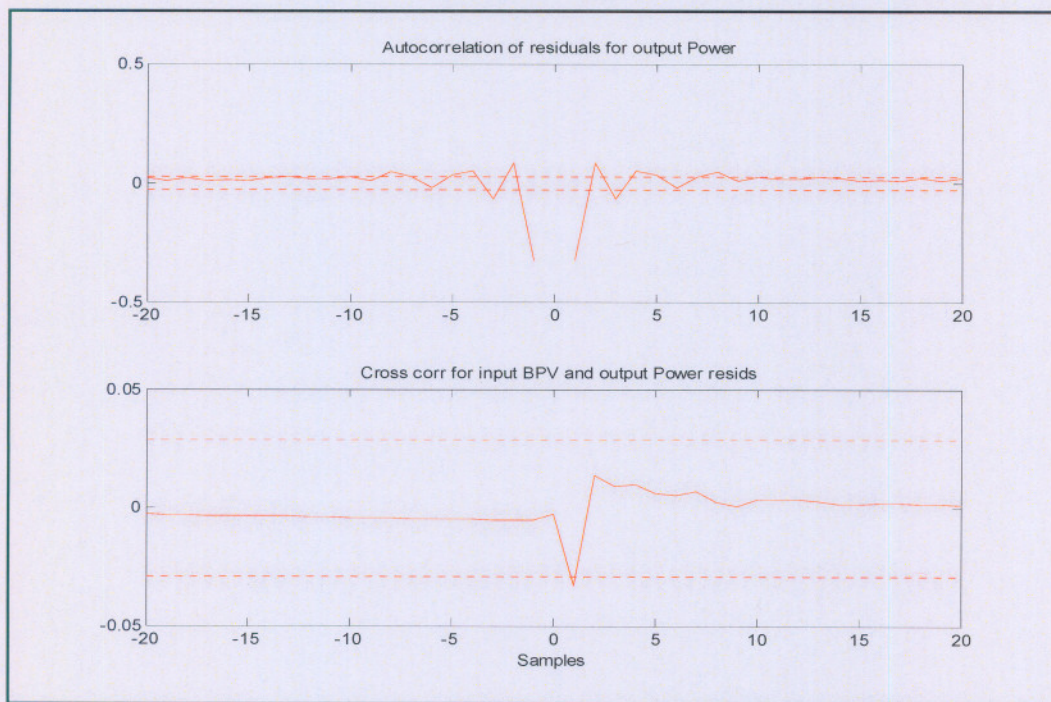


Figure 4-91: The residuals of the n4s3 model

4.4.3.7 Validation of best model for Bypass valve regulation system identification

The best model to predict and simulate the BPV process according to the best fit is the n4s2 model. This model also has the best cross-correlation function, with almost no correlation between the input and the error. This model also has a better auto-correlation function than the second best model (according to fit), the n4s3 model. This makes this model more superior to the rest.

It is also important to note that this is the first process the second and third order SS models are even in contention for the best model. The reason for this is that the SS model may be less susceptible to non-linearity, if it is kept in mind that the BPV process is actually non-linear [3].



4.5 MISO system identification

4.5.1 Simulation of the MISO model

The inputs that were used for the MISO model were chosen in such a way that the output could contain data that differentiates between the effects of the different inputs, as well as the simultaneous effects of the inputs. The constraints of the physical system were taken into account, as well as the criterion that the input signal must meet and the following input signal was chosen for the LPI input:

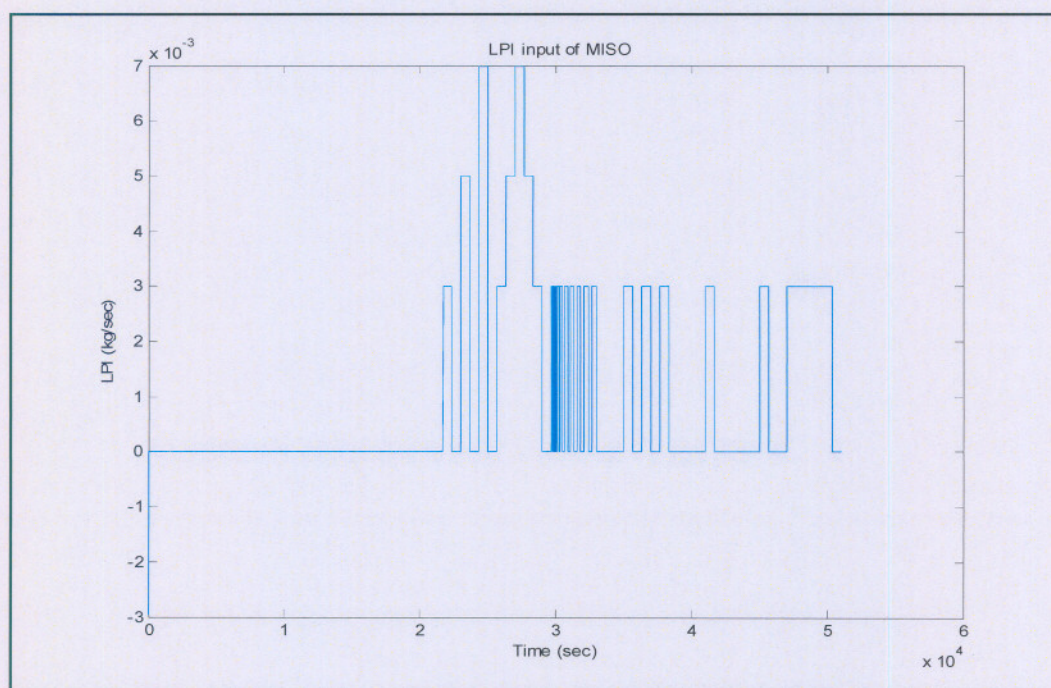


Figure 4-92: The LPI input that was used for identification

The input that was used for the HPE input is illustrated in Figure 4-93.



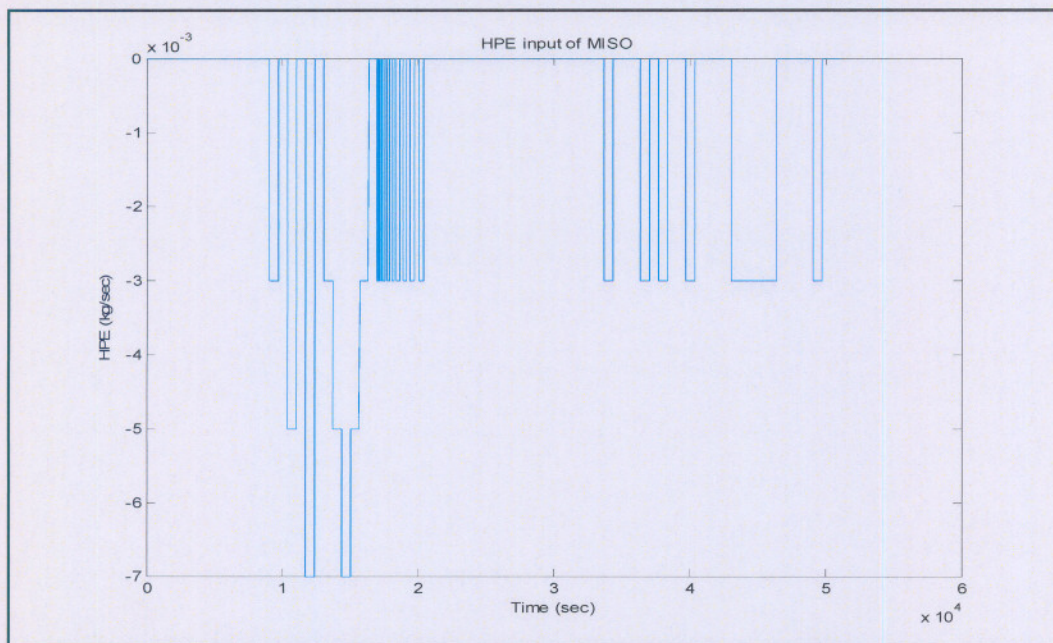


Figure 4-93: The HPE that was used for identification

The input that was used for the BPV input is illustrated in Figure 4-94.

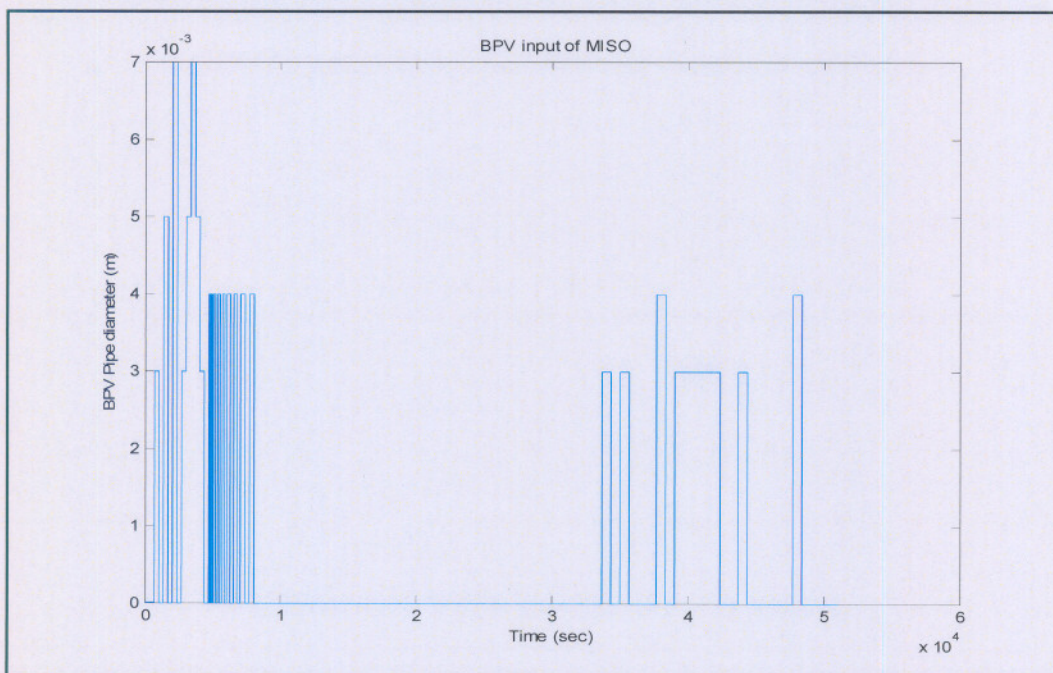


Figure 4-94: The BPV input that was used in data recording



This input was again transformed to represent the area of the pipe and not the diameter of the pipe. This transformed input that was used for identification is illustrated in Figure 4-95.

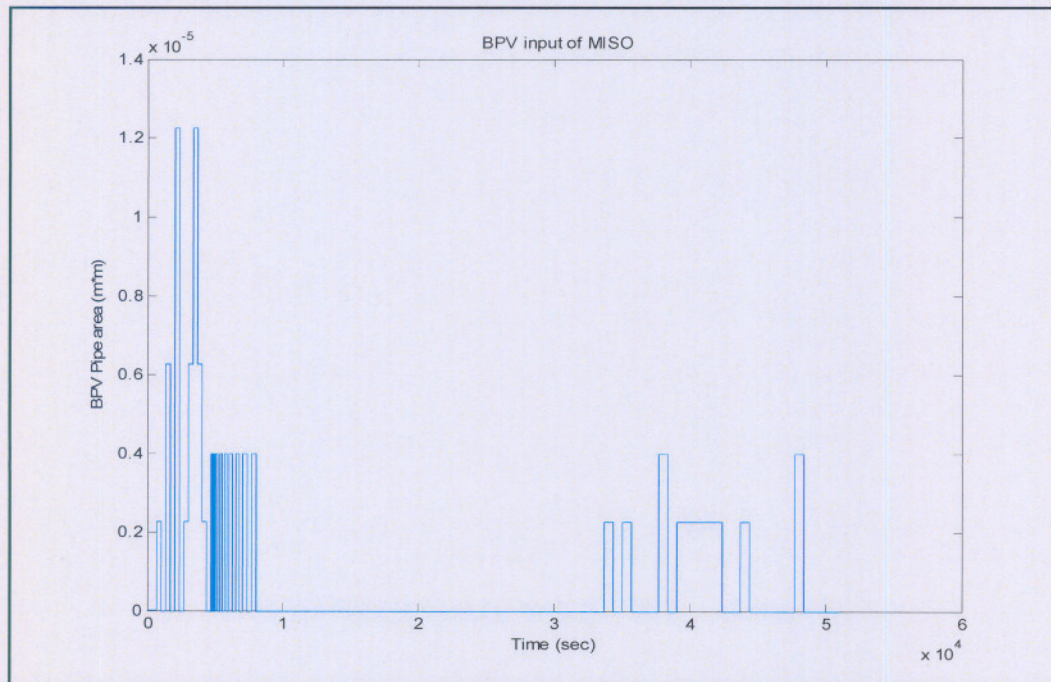


Figure 4-95: The BPV input that was used for identification

The output that was recorded is illustrated in Figure 4-96.



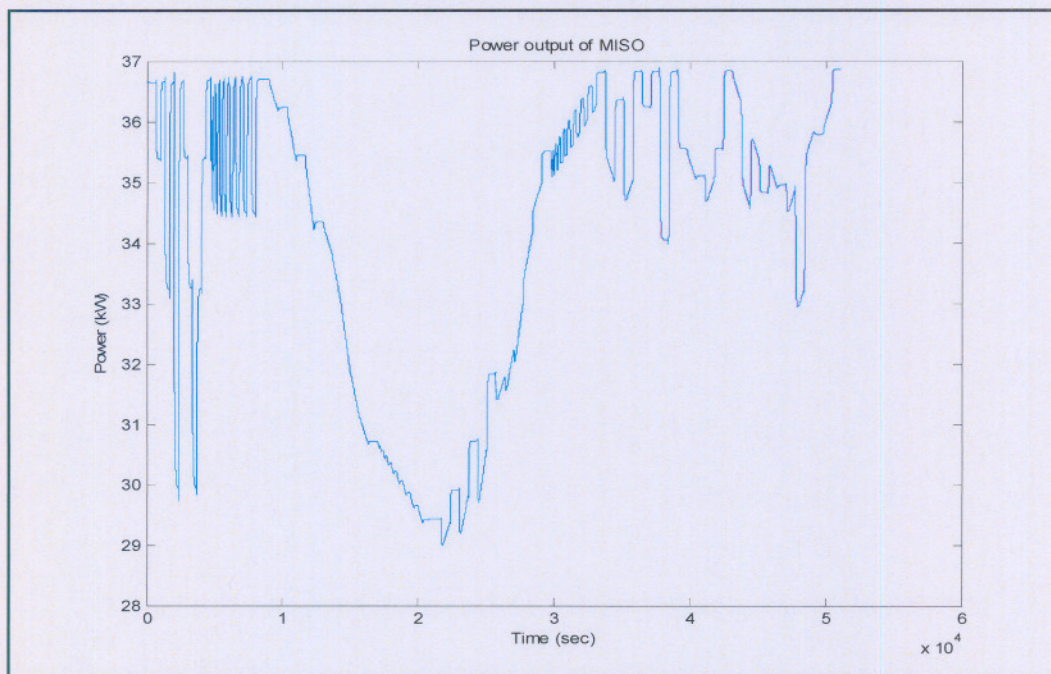


Figure 4-96: The output that was recorded

In Figure 4-96 it is clear that the system was always inside the parameters for normal power mode operation. In Figure 4-96 the offset of 36.6 kW is obvious. This offset must be subtracted to make the data suitable for identification. The output that is obtained after the offset is subtracted is illustrated in Figure 4-97.



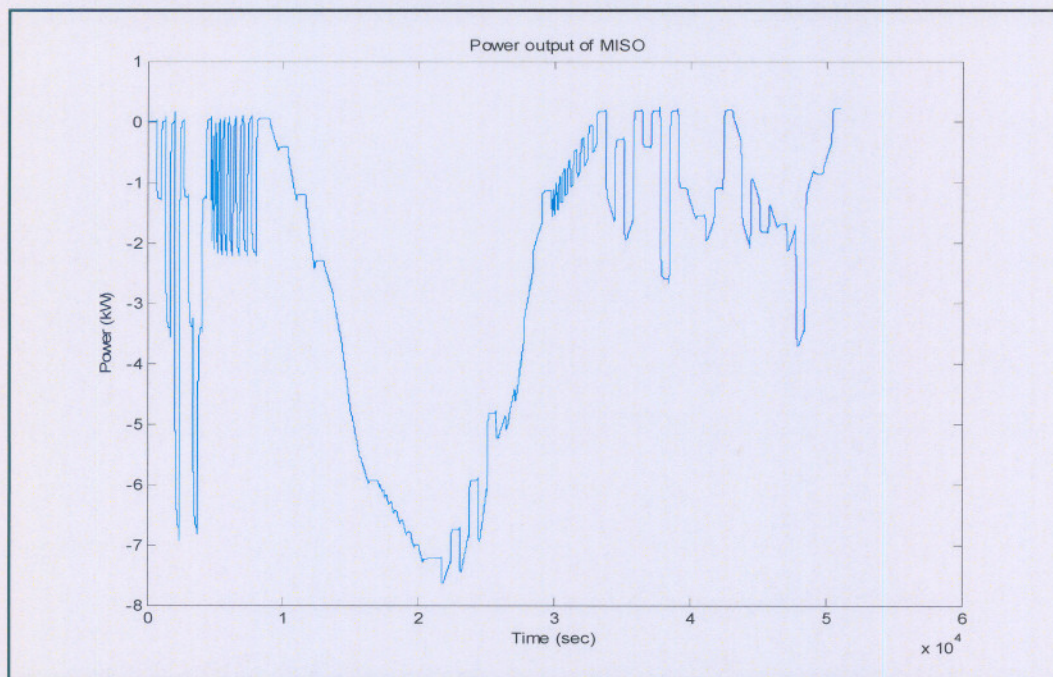


Figure 4-97: The output that was used in identification

There are no other problems with the input or output data and thus the data described in Figure 4-92, Figure 4-93, Figure 4-95 and Figure 4-97 is used in the following sections to identify different models.

4.5.2 MISO model identification

The MISO model is more complicated to identify than the SISO models. In this section, the identification of the MISO model discussed in section 4.3 is described. The more complicated models means that the settings of the algorithms had to be adjusted from their default values. The adjustments help to compute more accurate models.



4.5.2.1 ARX model identification

There are a lot more models to calculate when multivariate systems are observed. In this case the system has three inputs and one output. This means that the order of the polynomial $B(q)$ becomes a vector with one row and three columns. The order of the polynomial $A(q)$ is still a scalar value, because the system has only one output.

The algorithm that is used to calculate the parameters of the models are described in section 4.2.1. It is the same algorithm that is used to calculate the SISO models. This algorithm proved to be good for identification when the SISO models were calculated.

The first step in identification is again to find the delay parameter k . The delay of all the different SISO models were one. This is also the case for the MISO system. The same settings were used as in section 4.4.1.2. The different second and third order models that were calculated and their performance are listed in Table 4-16.



Table 4-16

Name	n_a	n_b	Fit
ARX23321	2	3 3 2	97.60%
ARX23221	2	3 2 2	97.59%
ARX23331	2	3 3 3	97.57%
ARX23231	2	3 2 3	97.56%
ARX32231	3	2 2 3	97.19%
ARX32331	3	2 3 3	97.17%
ARX32221	3	2 2 2	97.15%
ARX32321	3	2 3 2	97.13%
ARX33231	3	3 2 3	97.04%
ARX331	3	3 3 3	97.01%
ARX33221	3	3 2 2	96.93%
ARX33321	3	3 3 2	96.92%
ARX22331	2	2 3 3	96.66%
ARX22231	2	2 2 3	96.65%
ARX22321	2	2 3 2	96.64%
ARX221	2	2 2 2	96.63%

The system is now more complex. This is the reason the lowest order model, the ARX221 model, does not perform as well as the other models. All the models were investigated, but only the evaluation of four models will be described. The four models are the ARX221, the ARX331, ARX3223 and the ARX23321 model. The ARX221 model is evaluated first.

The simulated output of the ARX221 model is illustrated in Figure 4-98. The partial view of the simulated output is illustrated in Figure 4-99. The partial view of the simulated output confirms that the model does not capture all the dynamical behaviour of the system.



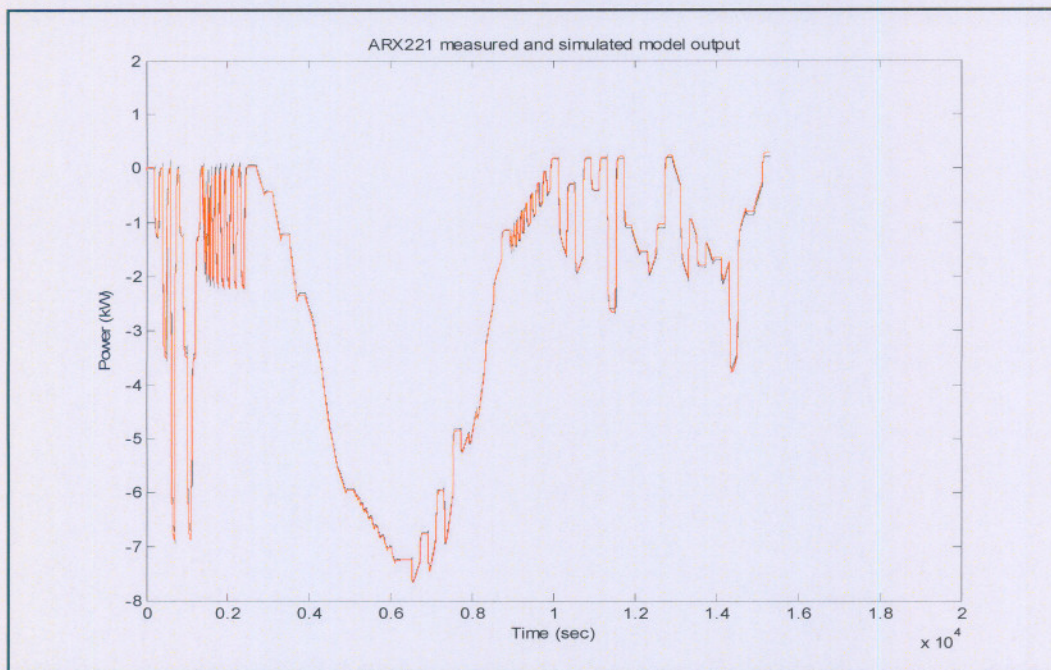


Figure 4-98: The simulated output of the ARX221 model (red) and the measured output

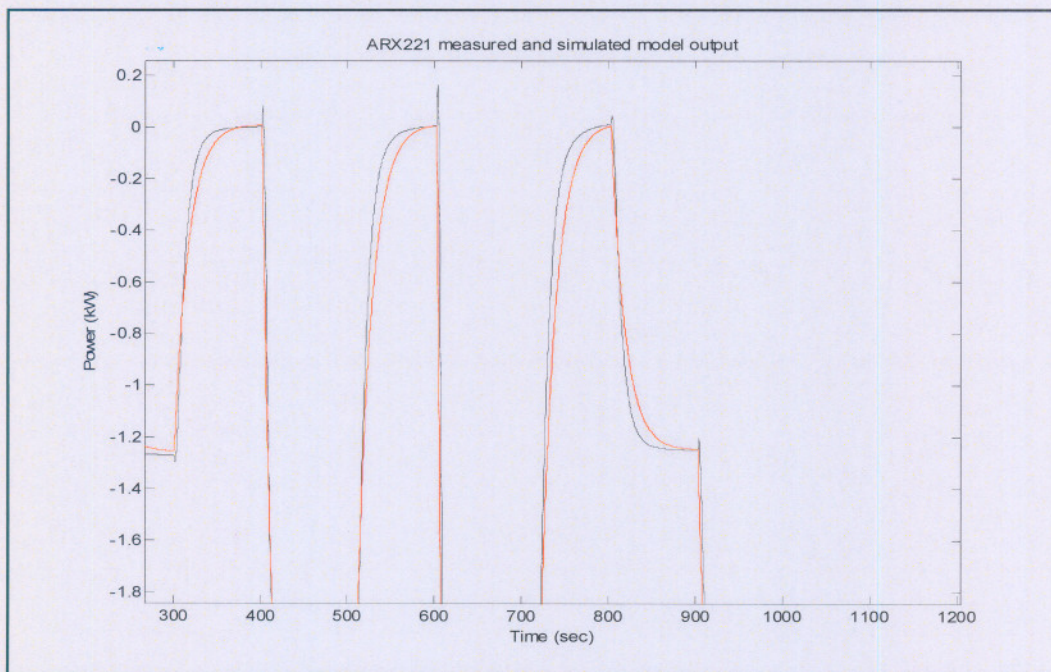


Figure 4-99: The partial view of the simulated output in Figure 4-98



The residuals of the model are illustrated in Figure 4-100, Figure 4-101 and Figure 4-102. The residuals of the first channel (Figure 4-100) of this model are good. The auto-correlation function illustrates that the error is a white noise process and thus contains no more information. The low cross-correlation values indicate that the error is uncorrelated to the input. The same conclusions could be made of the residuals of the other two channels illustrated in Figure 4-101 and Figure 4-102. The parameters that were calculated for the ARX221 model are:

$$\begin{aligned} (1 - 1.982q^{-1} + 0.9816q^{-2})y(t) = & (-3299q^{-1} + 3299q^{-2})u_1(t) + \\ & (0.5879q^{-1} - 0.5836q^{-2})u_2(t) + (-3.153q^{-1} + 3.158q^{-2})u_3(t) + e(t) \end{aligned}$$

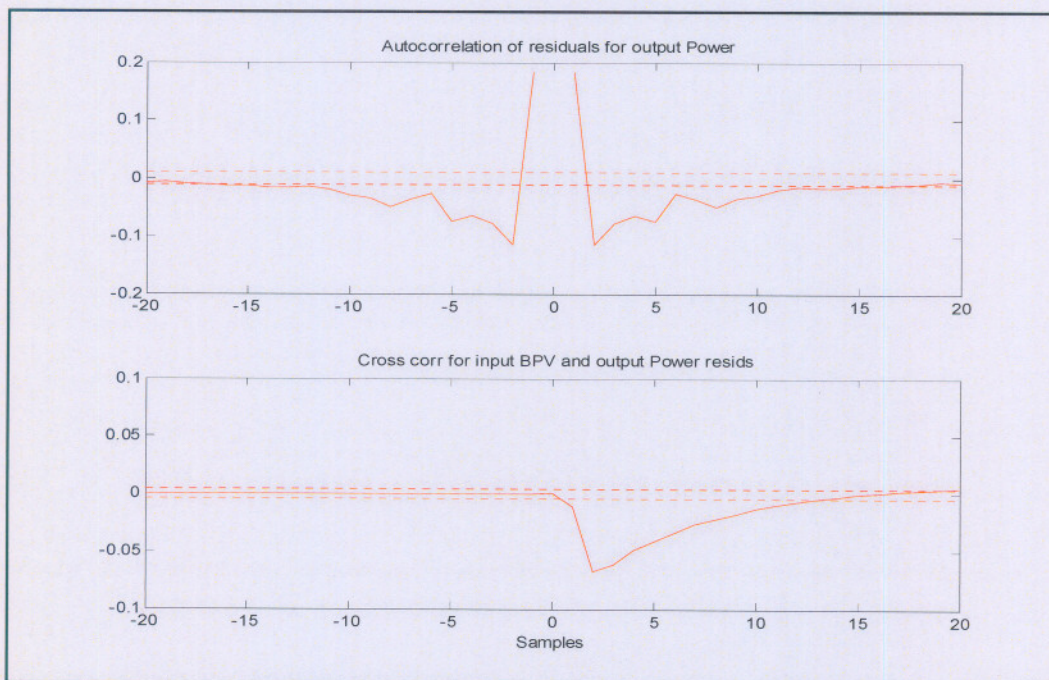


Figure 4-100: The residuals of the first channel (BPV-Power)



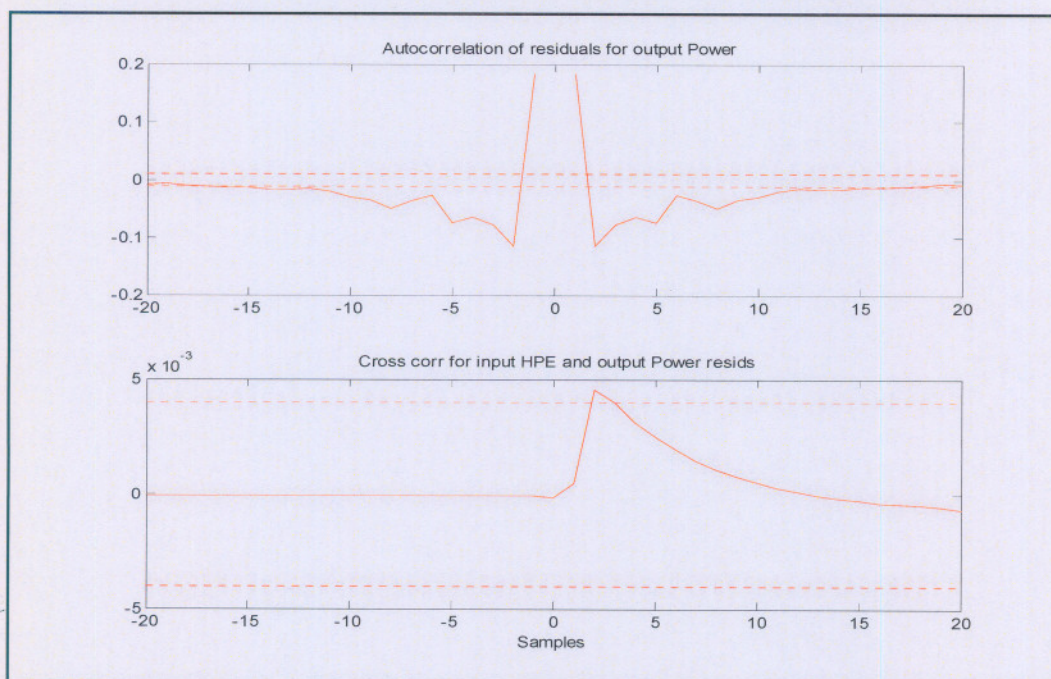


Figure 4-101: The residuals of the second channel (HPE-Power)

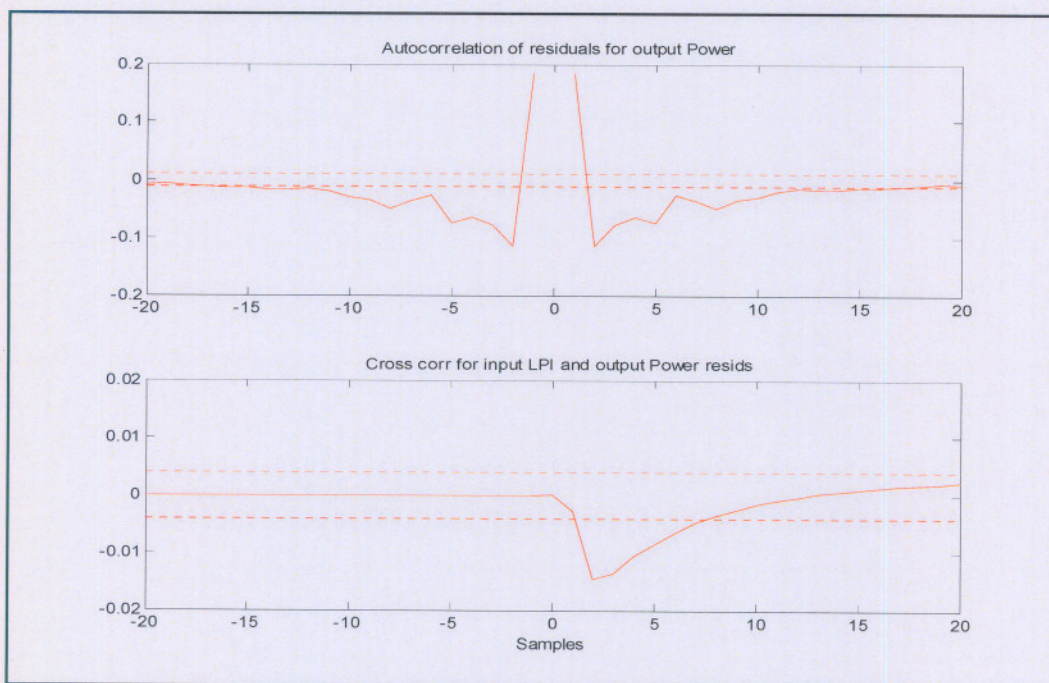


Figure 4-102: The residuals of the third channel (LPI-Power)



The partial view of the simulated output of the ARX331 model is illustrated in Figure 4-103. In this figure it is obvious that this model does not describe the behaviour of the system well.

The best model according to the fit criteria is the ARX23321 model. The simulated output of this system is illustrated in Figure 4-104. The spikes in Figure 4-104 that are encountered when the input changes levels will cause problems. The spikes are the reason this model is rejected and will not be evaluated further. The next three models in Table 4-16 all suffer from the same problem as the ARX23321 model.

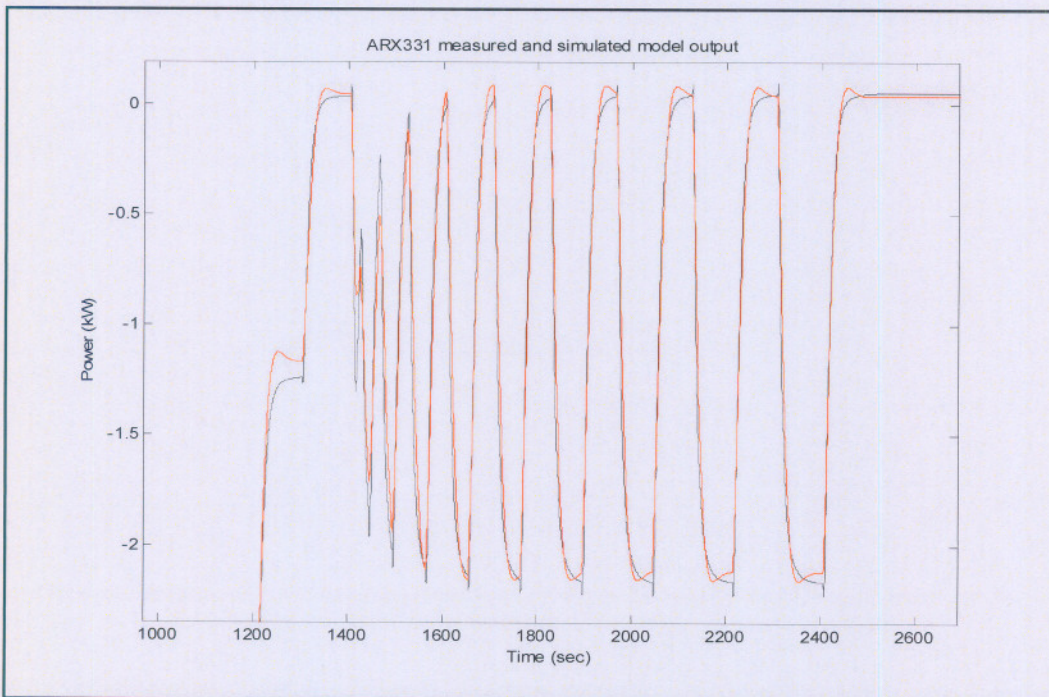


Figure 4-103: The partial view of the simulated output of the ARX331 model (red) and the measured output



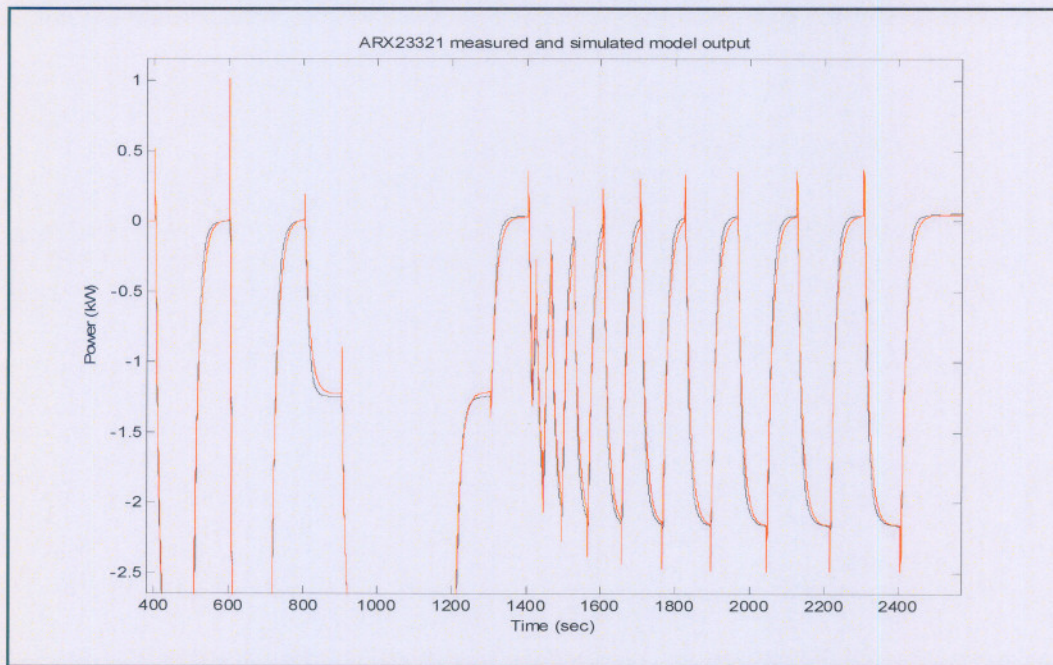


Figure 4-104: The partial view of the simulated output of the ARX23321 model (red)

The model that describes the dynamic behaviour of the MISO system the best is the ARX32231 model. The partial view of the simulated output of this model is illustrated in Figure 4-105.



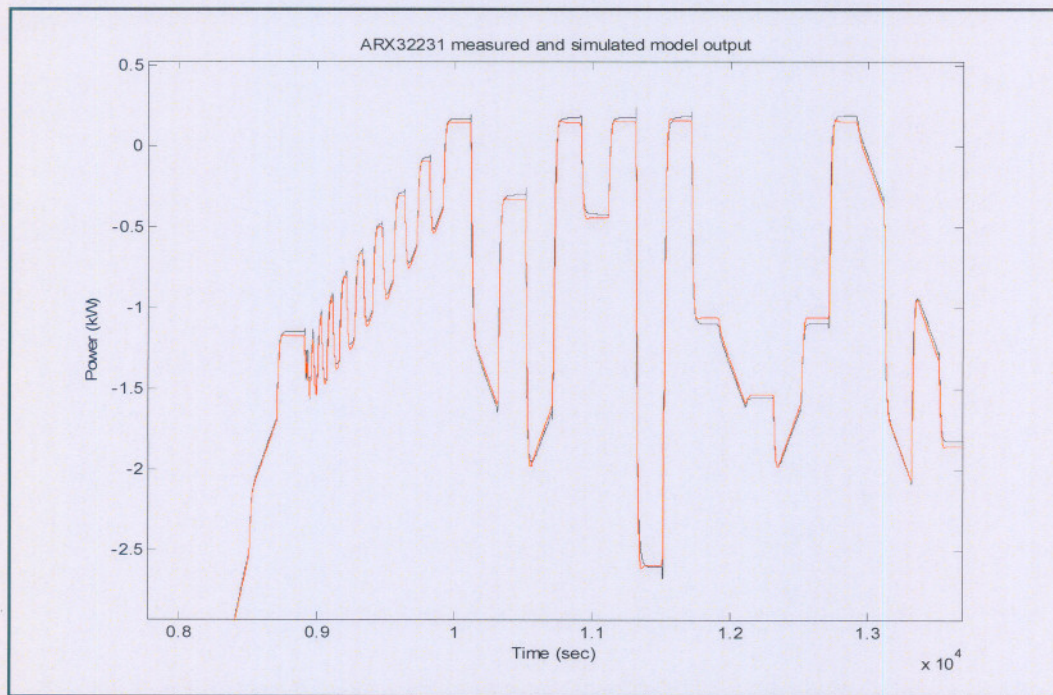


Figure 4-105: The partial view of the simulated output of the ARX32231 model (red) and system output

The residuals of the ARX32231 model are illustrated in Figure 4-106, Figure 4-107 and Figure 4-108. The residuals of the first channel (Figure 4-106) of this model are good. The auto-correlation function illustrates that the error is a white noise process and thus contains no more information. The low cross-correlation values indicate that the error is uncorrelated to the input. The same conclusions could be made of the residuals of the other two channels illustrated in Figure 4-107 and Figure 4-108.



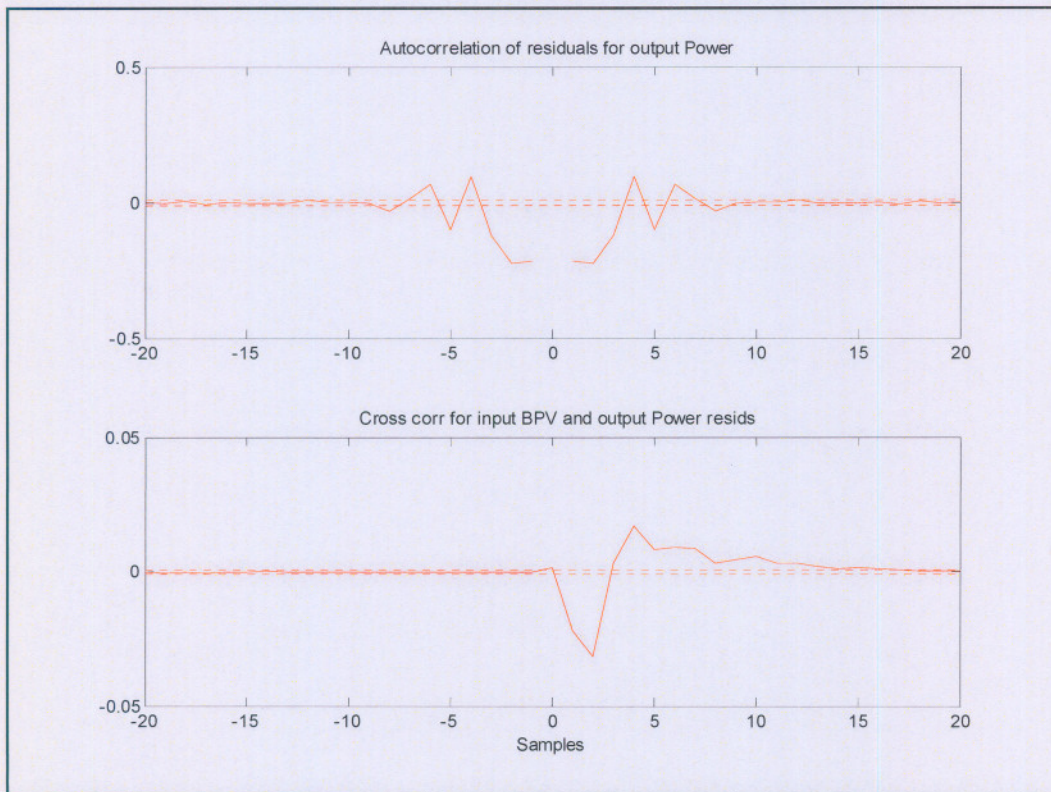


Figure 4-106: The residuals of the first channel (BPV-Power) of the ARX32231 model

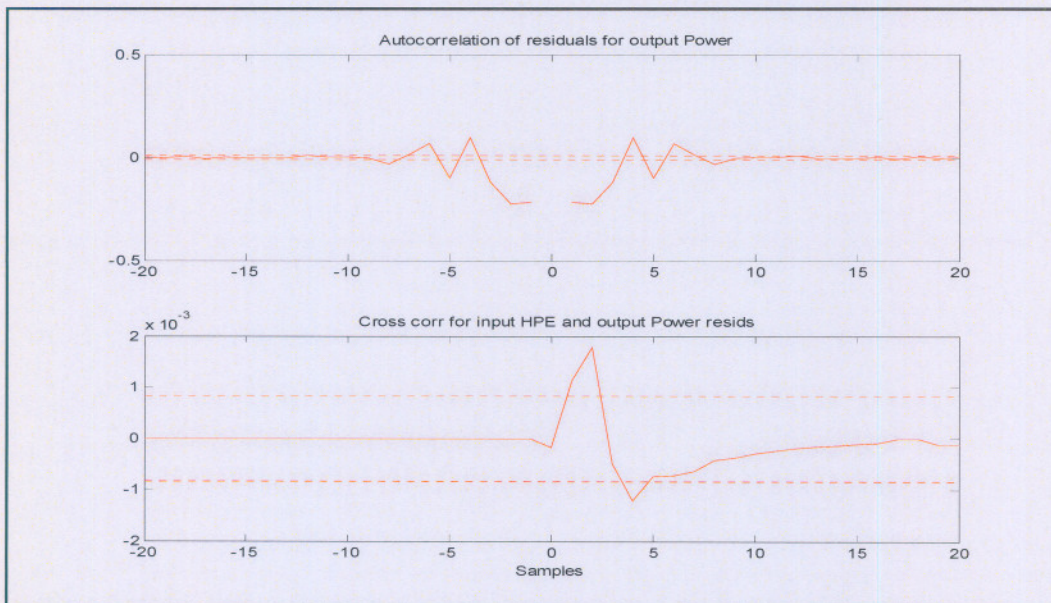


Figure 4-107: The residuals of the second channel (HPE-Power) of the ARX32231 model



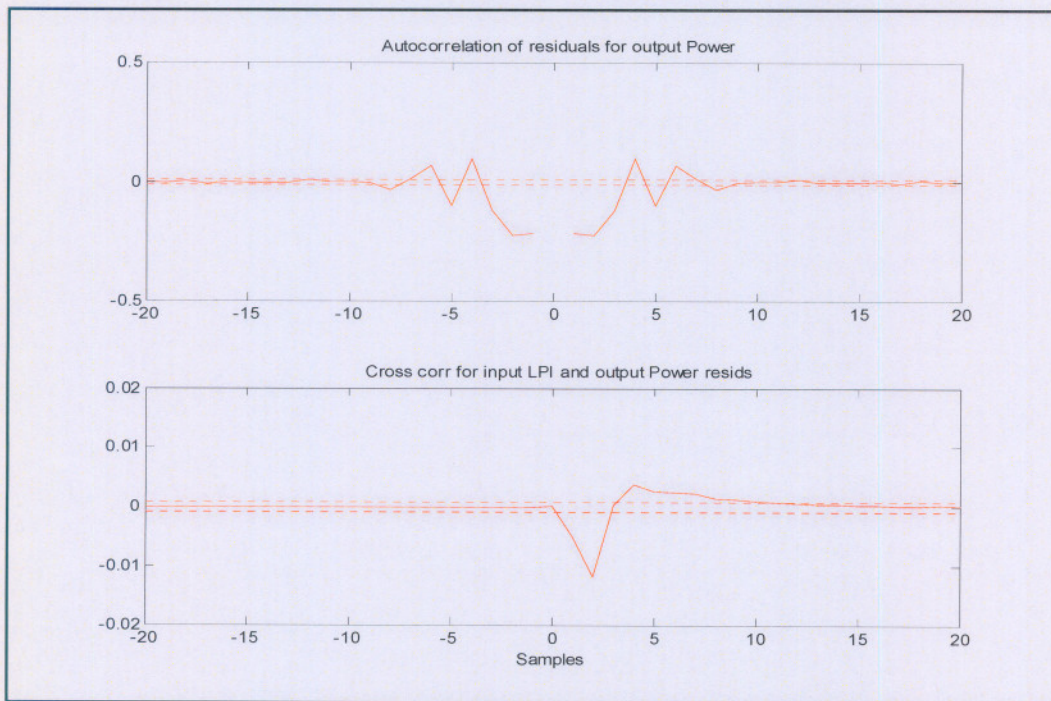


Figure 4-108: The residuals of the third channel (LPI-Power) of the ARX32231 model

The parameters that were calculated for the ARX32231 model are:

$$\begin{aligned} (1 - 2.94q^{-1} + 2.882q^{-2} - 0.9416q^{-3})y(t) = & (-208.1q^{-1} + 208.1q^{-2})u_1(t) + \\ & (0.03574q^{-1} - 0.03546q^{-2})u_2(t) + (-1.377q^{-1} + 2.551q^{-2} - 1.174q^{-3})u_3(t) + e(t) \end{aligned}$$

4.5.2.2 ARMAX model identification

According to the specifications of the toolbox it should be able to calculate ARMAX MISO models, but the version that was used was unable to do this.



4.5.2.3 Output Error model identification

The algorithm that is used to calculate the Output Error models is described in 4.2.2. The settings that were used proved to be the critical aspect of the identification in this section. The default settings that were used previously to identify OE models produced poor results. The best results that could be obtained after numerous attempts were those from the OE221 model. The order of polynomial $B(q)$ is two for each input and the order of polynomial $F(q)$ is also two for each input of the OE221 model.

This model resulted in a fit of 97.58% if the following settings were used:

- Fixedpara=default
- Tolerance=0.000001
- LimitError=0
- MaxIter=50

All the other settings were kept the same as for the SISO model identification. When the "MaxIter" changes to a value of 49 only an 85.77% fit was obtained. This places emphasis on the choice of settings. A change of one iteration made a big difference. The iterative search for the best parameters often results in the selection of a local minimum being selected instead of global minimum.

The OE221 model was the only model that was estimated that could describe the dynamical behaviour of the system to an extent. The results of the other model were less than 50% fit, after all the parameters were changed between certain reasonable values.

The partial view of the simulated output of the OE221 model is illustrated in Figure 4-109.



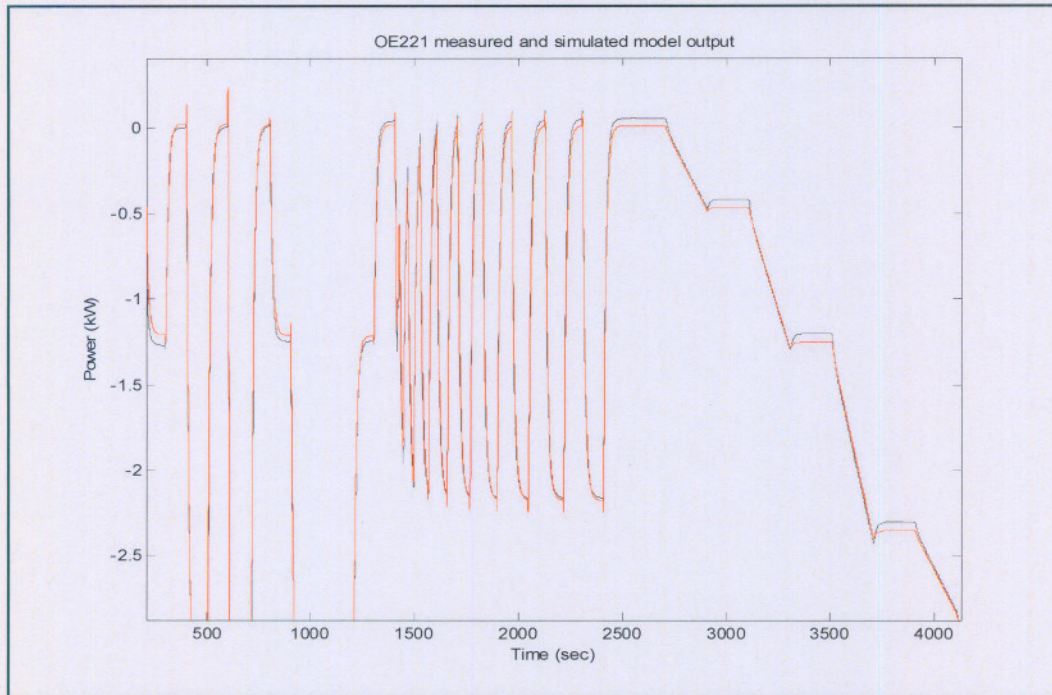


Figure 4-109: The partial view of the simulated output of the OE221 model (red) and the system output

The residuals analysis of the OE221 model is illustrated in Figure 4-110, Figure 4-111 and Figure 4-112. The OE models that were calculated in this study so far have had excellent cross-correlation functions, but very bad auto-correlation functions. This model is no exception. The bad auto-correlation means that there is still information of the systems behaviour in the error that is not included in the model.



System identification of the PBMM

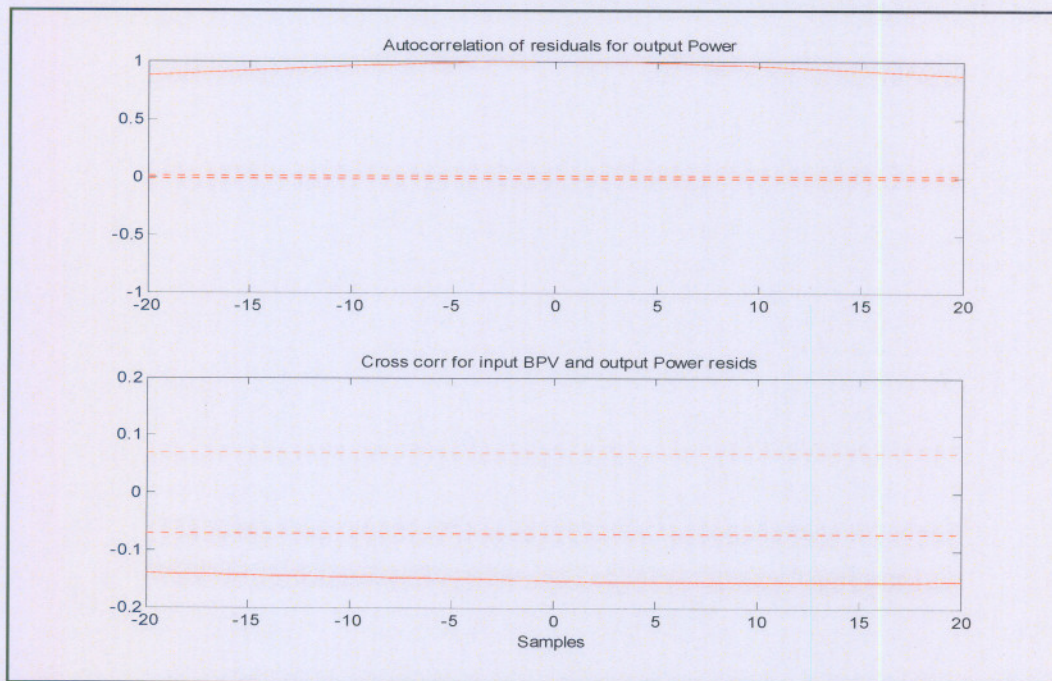


Figure 4-110: The residuals of the first channel (BPV-Power) of the OE221 model

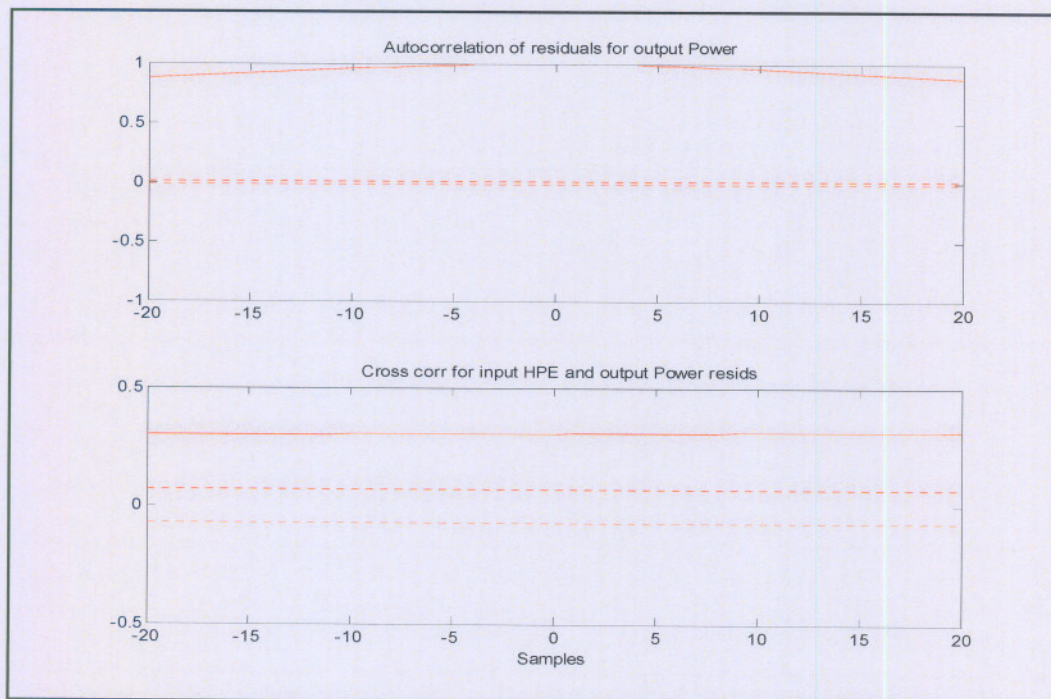


Figure 4-111: The residuals of the second channel (HPE-Power) of the OE221 model



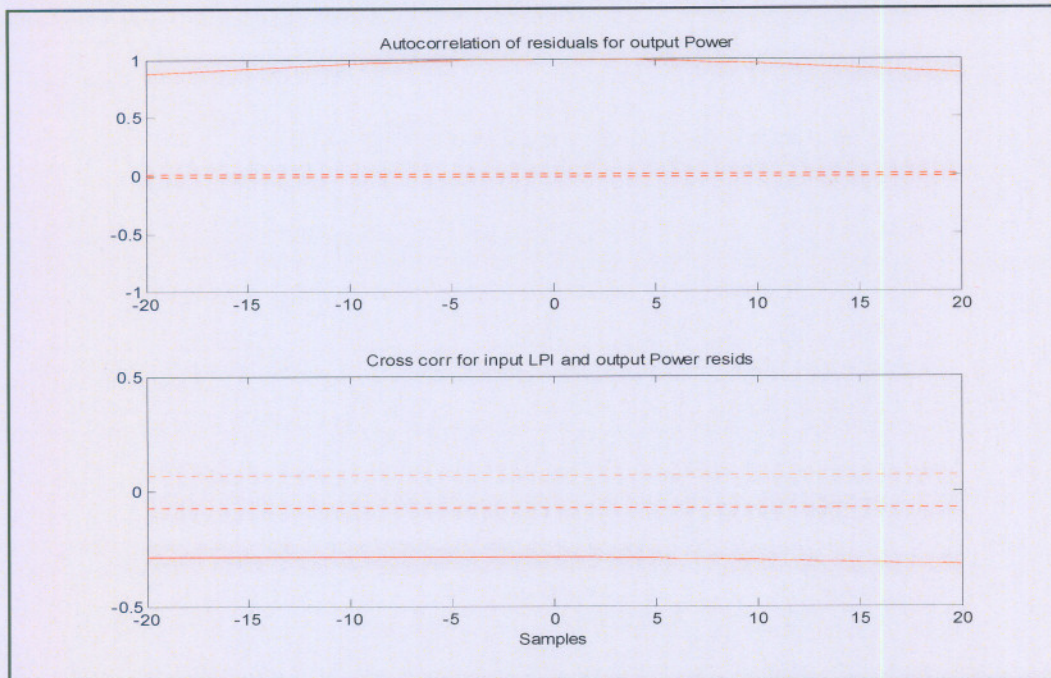


Figure 4-112: The residuals of the third channel (LPI-Power) of the OE221 model

The bad residuals of the OE221 model makes this model unsuitable for use.

The parameters of the model were estimated to be:

$$y(t) = \frac{(2891q^{-1} - 3493q^{-2})}{(1 - 1.839q^{-1} + 0.8425q^{-2})}u_1(t) + \frac{(0.3545q^{-1} - 0.3502q^{-2})}{(1 - 1.982q^{-1} + 0.982q^{-2})}u_2(t) + \frac{(-3.887q^{-1} + 3.892q^{-2})}{(1 - 1.977q^{-1} + 0.9769q^{-2})}u_3(t) + e(t)$$

4.5.2.4 Box-Jenkins model identification

The complexity of the system (investigated in the next chapter) resulted in poor models being estimated with the Prediction Error method and the BJ model structure. The iterative search could not find the parameters that result



in the global minima of the error function. All the settings were changed and various combinations of settings were used, but the best models that could be calculated had a fit of less than 10%.

The best results were obtained when the “MaxIter” was set to zero. This means that the initially estimated parameters (calculated by the n4sid algorithm) are used. The n4sid algorithm is used to estimate the parameters of the SS MISO model in the next section.

The BJ models have less parameters than the SS model. This means that the n4s3 model (calculated in the next section) describes the system a lot better than a third order BJ model (“MaxIter” =0) .

The best model that was calculated for the MISO system with the BJ model structure was a first order model (the orders of all the polynomials equal to one). The following property settings were used in Matlab©:

- Initial condition – Zero
- Focus – Simulation
- Covariance – Estimate

All the other settings were set to default. A 48.7% fit was the result. The first order model has less parameters to calculate. The more parameters that must be calculated the more difficult it is to estimate the parameters iteratively.

This led to the conclusion that the prediction error method struggled to find the global minima of the error of a complex system. As the number of parameters that must be calculated increases the worse the fit of the calculated model becomes when the prediction error method is used.



4.5.2.5 State space model identification

The algorithm that is used to calculate the parameters of the models are described in section 4.2.3.

The same settings were used as in section 4.4.1.6. Different models were calculated and their performances are listed in Table 4-17. The second and third order model produces poor results. The poor results led to the calculation of higher order models, but these models will not be evaluated.

Table 4-17

Name	n	Fit
N4S7	7	95.22%
N4S6	6	92.06%
N4S5	5	89.50%
N4S4	4	88.34%
N4S8	8	87.08%
N4S3	3	79.69%

The poor fit of the second and third order models makes them unsuitable for simulation or prediction. These models will not be evaluated further on account of their poor fit.

4.5.2.6 Validation of best model for the MISO system

The best model that was calculated for the MISO system is the ARX32231 model. This model describes the dynamical behaviour of the system relatively accurately. The worst predictions that this model predicts is for the BPV input, which is actually a non-linear input as mentioned previously [3]. The residuals of this model verify that this is the best model to use. There are models that



MISO system identification techniques using the PBMR data.

System identification of the PBMM

are better when the fit percentage is the selection criteria, but all of these models produced undesired spikes when simulated.



5. MISO system analysis

In this chapter the effects that different inputs have on each other is examined and documented.



5.1 Description of MISO system analysis

In this chapter the analysis of the interactions of the different inputs are analyzed. In the previous section some of the models failed to describe the behaviour of the system due to the fact that a very complex system is used.

The complexity of the system is obvious when the parameters of the ARX221 model for the different SISO models are compared to the parameters of the MISO ARX221 model.

In this chapter a MISO system is constructed by combining the outputs of the SISO systems. The dynamical behaviour of the new constructed MISO system is compared to the dynamical behaviour of the real MISO system. Various SISO systems are combined and compared to the MISO system.

5.2 MISO system analysis

The MISO analysis is done by giving a step input (Figure 5-1, Figure 5-2, Figure 5-3) to the different SISO systems and recording the outputs. The outputs are then added up to form one output. The same inputs are then given to the MISO system and the output of the MISO system is also recorded.

In the analysis of the different MISO and SISO models the output will be plotted in the same figure in order to compare their values. A difference function is then computed by subtracting the output of the different SISO models from the output of the MISO model.



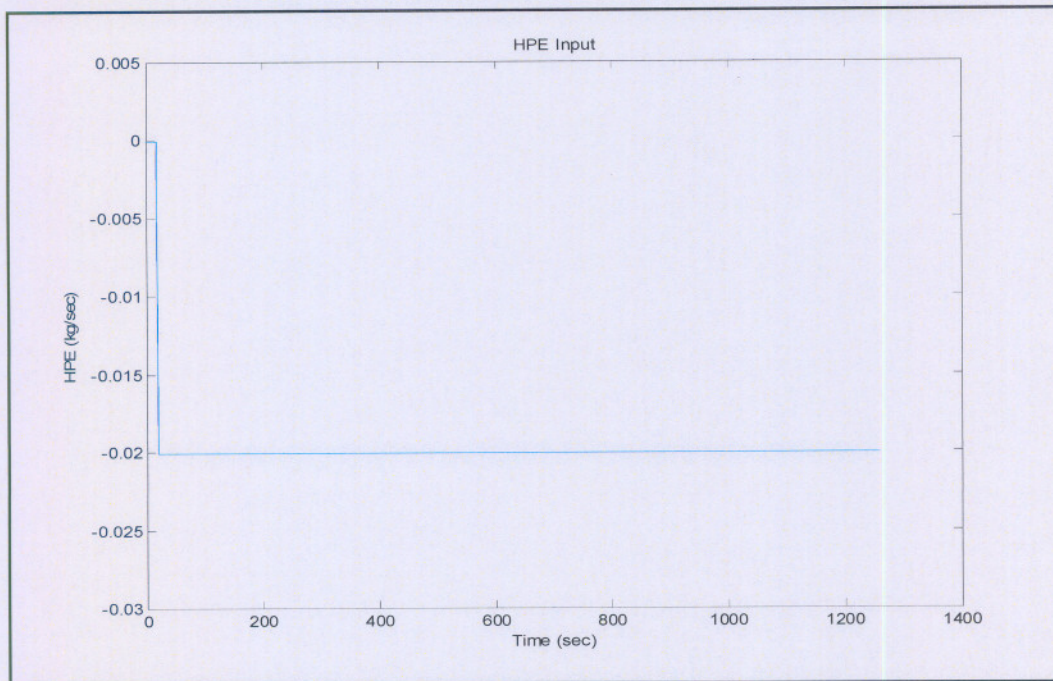


Figure 5-1: The step input that was used for the HPE process

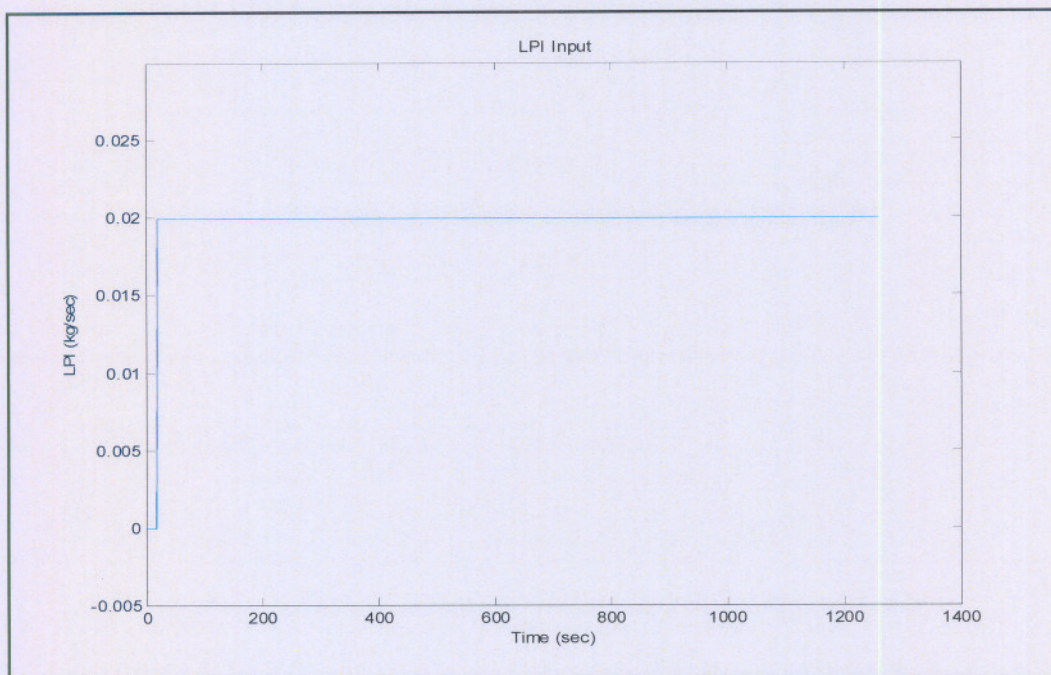


Figure 5-2: The step input that was used for the LPI process



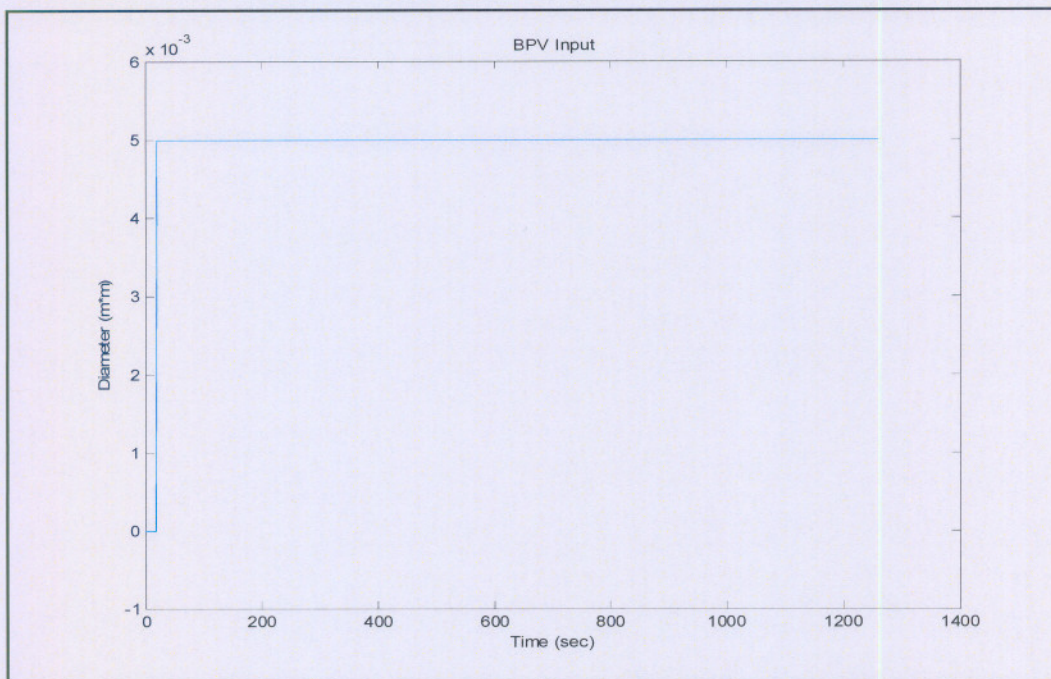


Figure 5-3: The step input that was used for the BPV process

5.2.1 The influence that the BPV and HPE have on each other

The first MISO analysis will be done by evaluating the effect the HPE process has on the BPV process.

The output of the SISO systems was calculated after the input illustrated in Figure 5-3 was given to the BPV SISO system and the input illustrated in Figure 5-1 was given to the HPE SISO system. The outputs were then added to form the output SISO Output BPV + SISO Output HPE. This output is illustrated in Figure 5-4.

The same inputs (illustrated in Figure 5-3 and Figure 5-1) were given to a MISO system and the output (MISO Output BPV HPE) was recorded. This



output is illustrated in Figure 5-4.

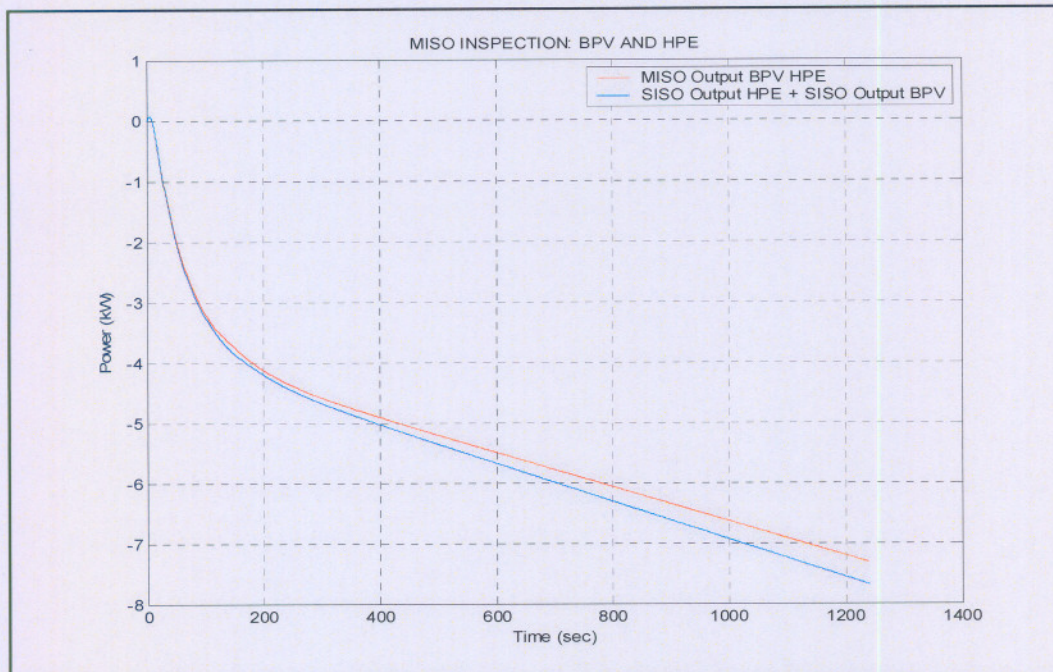


Figure 5-4: The outputs of the MISO and SISO systems

The difference output is obtained by subtracting SISO Output BPV + SISO Output HPE from MISO Output BPV HPE. The function obtained after subtraction is illustrated in Figure 5-5.

When Figure 5-5 is evaluated it is clear that the difference function makes strange curves for the first 250 seconds and then settles into a straight line. The straight line could be described by the following equation:

$$y(\text{Power}) = 0.0003t$$



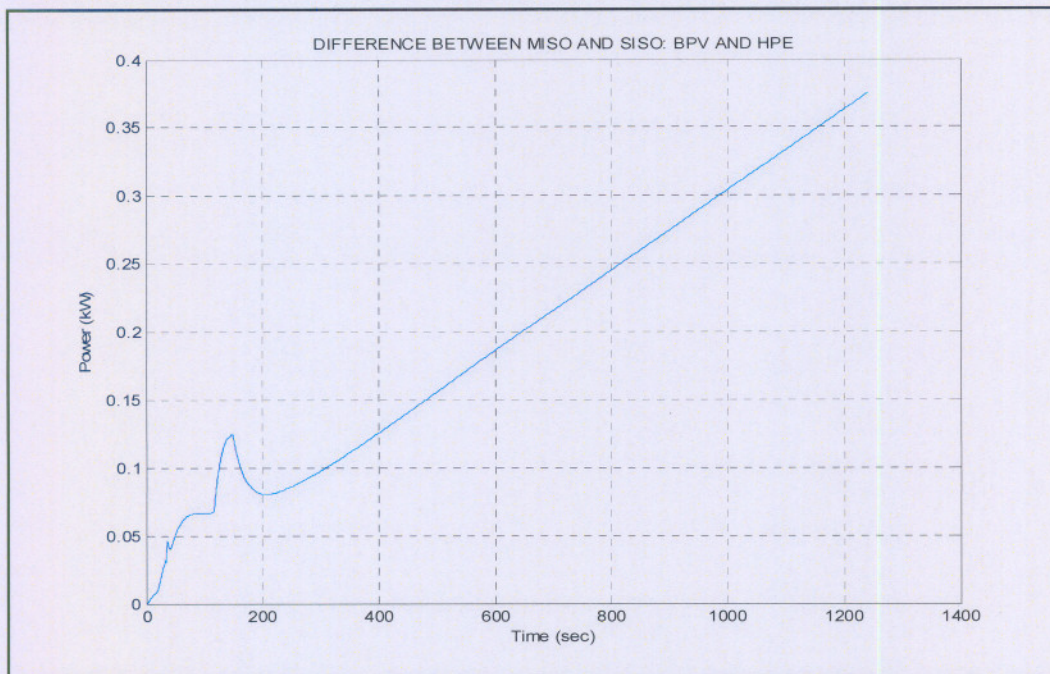


Figure 5-5: The difference function

5.2.2 The influence that the BPV and LPI have on each other

The second MISO analysis will be done by evaluating that the effect the LPI process has on the BPV process.

The output of the SISO systems was calculated after the input illustrated in Figure 5-3 was given to the BPV SISO system and the input illustrated in Figure 5-2 was given to the LPI SISO system. The outputs were then added to form the output SISO Output BPV + SISO Output LPI. This output is illustrated in Figure 5-6.

The same inputs (illustrated in Figure 5-2 and Figure 5-2) were given to a MISO system and the output (MISO Output BPV LPI) was recorded. This



output is illustrated in Figure 5-6.

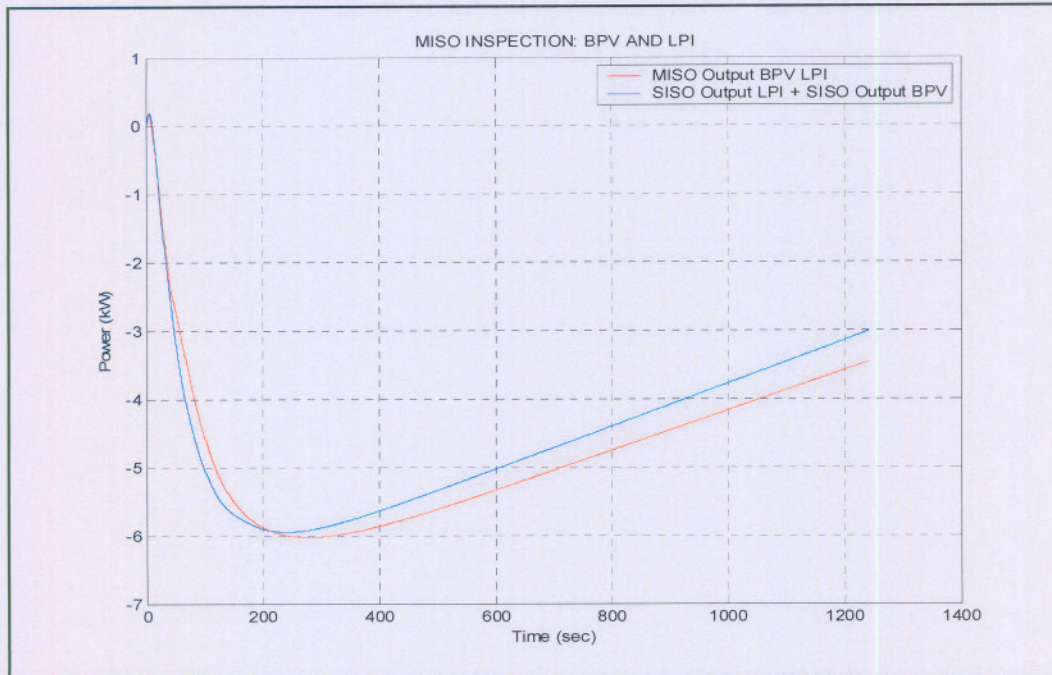


Figure 5-6: The outputs of the MISO and SISO systems

The difference output is obtained by subtracting SISO Output BPV + SISO Output LPI from MISO Output BPV LPI. The function obtained after subtraction is illustrated in Figure 5-7.

When Figure 5-7 is evaluated it is clear that the difference function makes strange curves for the first 600 seconds and then settles into a straight line. The straight line could be described by the following equation:

$$y(\text{Power}) = -0.000222t - 0.166$$



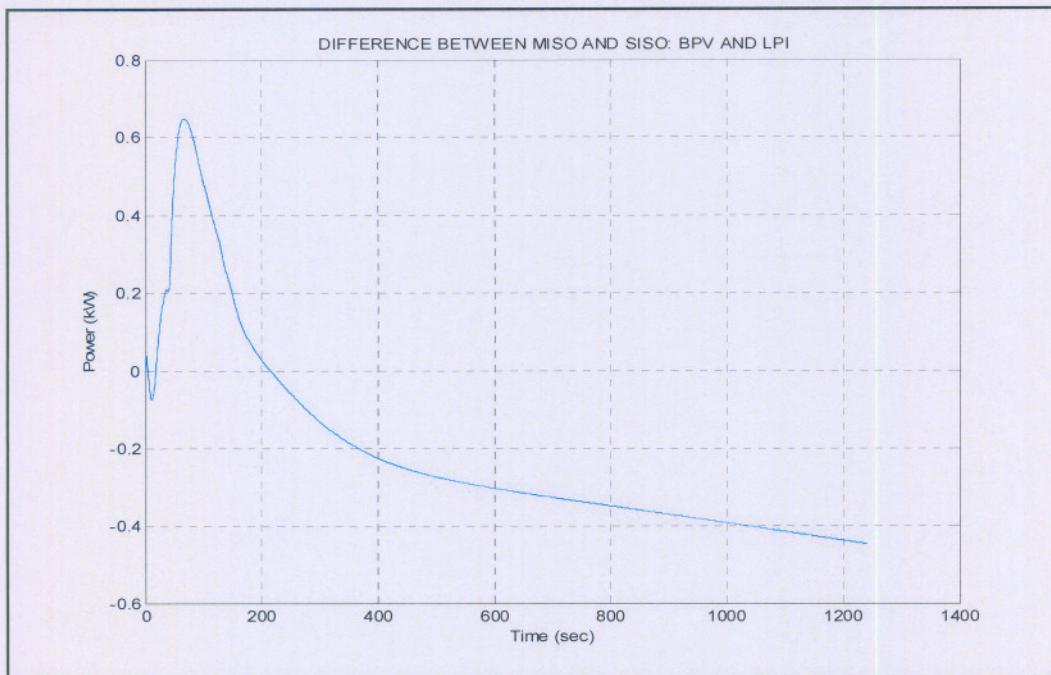


Figure 5-7: The difference function

5.2.3 The influence that the LPI and HPE have on each other

The third MISO analysis will be done by evaluating the effect that the HPE process has on the LPI process.

The output of the SISO systems was calculated after the input illustrated in Figure 5-2 was given to the LPI SISO system and the input illustrated in Figure 5-1 was given to the HPE SISO system. The outputs were then added to form the output SISO Output LPI + SISO Output HPE. This output is illustrated in Figure 5-8.

The same inputs (illustrated in Figure 5-2 and Figure 5-1) were given to a MISO system and the output (MISO Output LPI HPE) was recorded. This



output is illustrated in Figure 5-8.

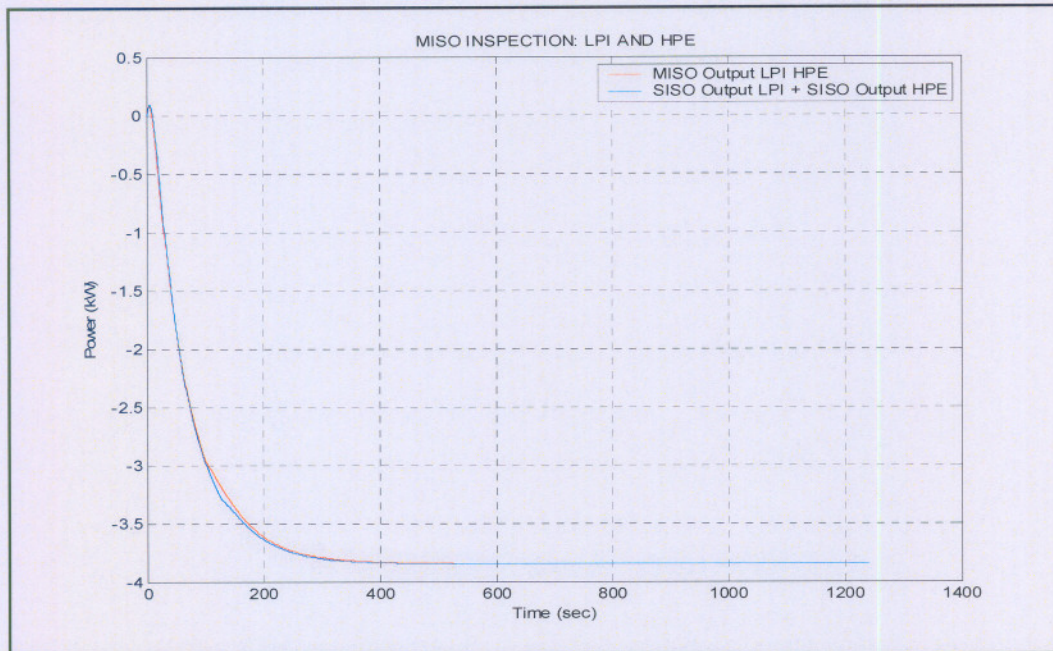


Figure 5-8: The outputs of the SISO and MISO systems

The difference output is obtained by subtracting SISO Output LPI + SISO Output HPE from MISO Output LPI HPE. The function obtained after subtraction is illustrated in Figure 5-9.

When Figure 5-9 is evaluated it is clear that the difference function makes strange curves for the first 800 seconds and then settles into a straight line. The straight line could be described by the following equation:

$$y(\text{Power}) = 0$$

The equation above proves that the two processes cancel each other out.



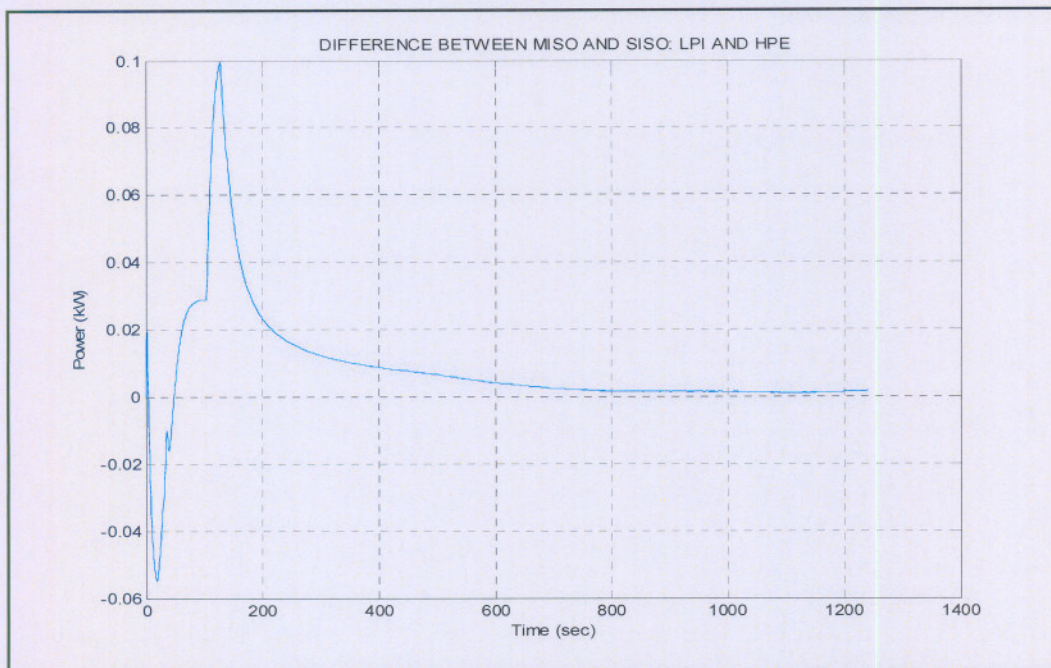


Figure 5-9: The difference function

5.2.4 The influence that the BPV, LPI and HPE have on each other

The last MISO analysis will be done by evaluating the effect that the HPE process and the LPI process have on the BPV process.

The output of the SISO systems was calculated after the input illustrated in Figure 5-3 was given to the BPV SISO system, the input illustrated in Figure 5-1 was given to the HPE SISO system and the input illustrated in Figure 5-2 was given to the LPI SISO system. The outputs were then added to form the output $\text{SISO Output BPV} + \text{SISO Output HPE} + \text{SISO Output LPI}$. This output is illustrated in Figure 5-10.

The same inputs (illustrated in Figure 5-3, Figure 5-2 and Figure 5-1) were given to a MISO system and the output (MISO Output BPV HPE LPI) was



recorded. This output is illustrated in Figure 5-10.

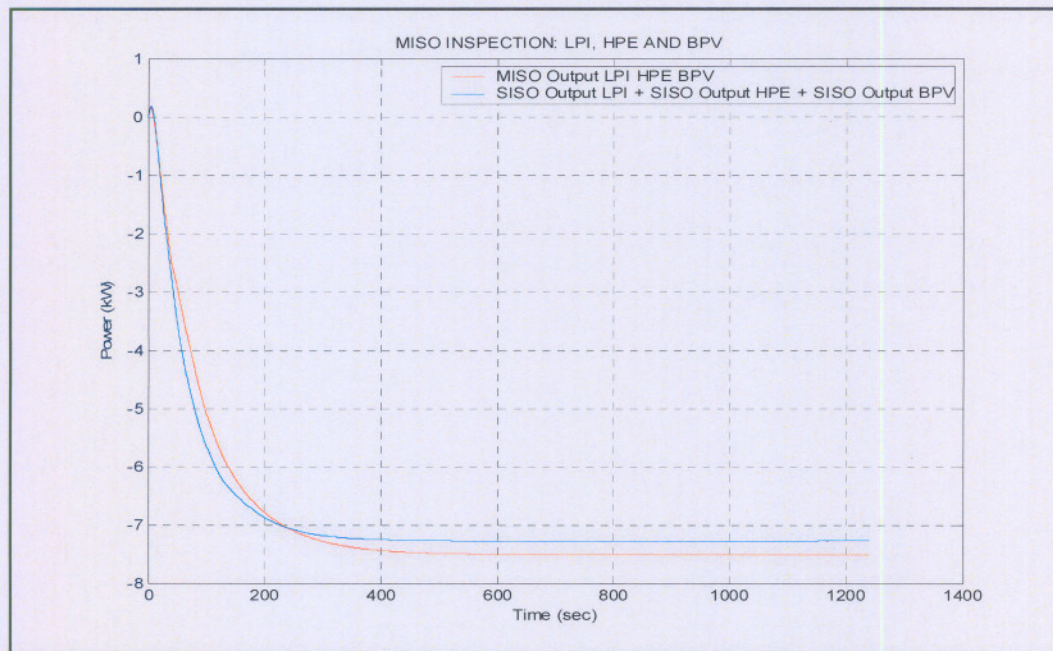


Figure 5-10: The outputs of the MISO and SISO systems

The difference output is obtained by subtracting SISO Output BPV + SISO Output HPE + SISO Output LPI from MISO Output BPV HPE LPI. The function obtained after subtraction is illustrated in Figure 5-11.

When Figure 5-11 is evaluated it is clear that the difference function makes strange curves for the first 600 seconds and then settles into a straight line. The straight line could be described by the following equation:

$$y(\text{Power}) = -0.22$$



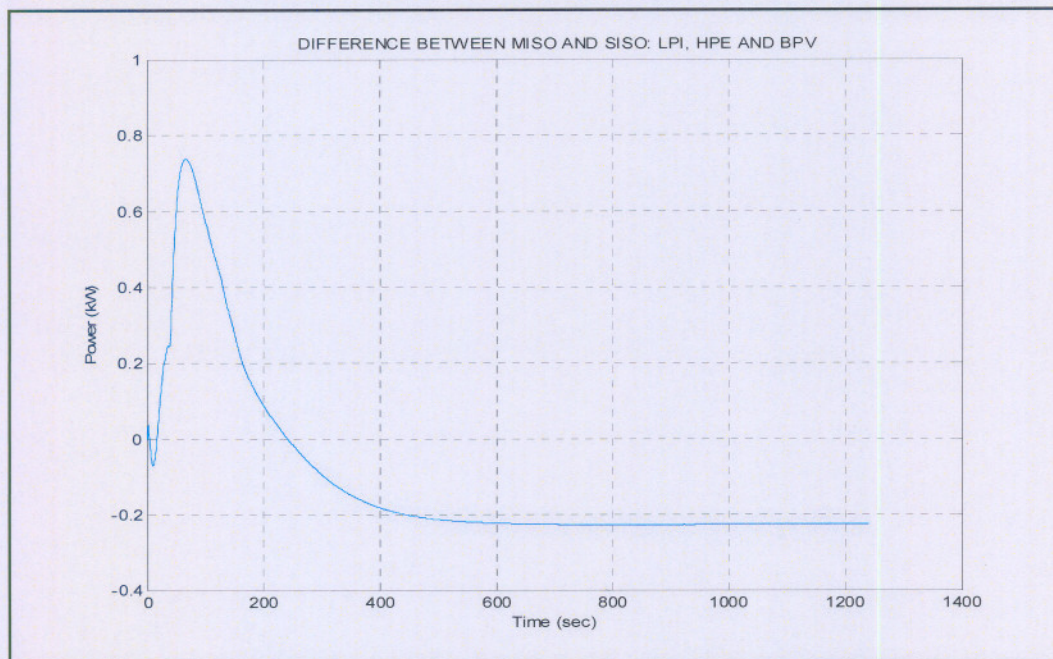


Figure 5-11: The difference function



6. Conclusion

A summary of the study done in this dissertation is given, followed by the contributions of this study. Suggestions for future research are given as well as concluding remarks.



6.1 Summary of results

In this dissertation a more comprehensive study was done on developing SISO models for the PBMR. The models that were calculated here are more robust than the models that were calculated by W.O. Erasmus [5], because the inputs that was used in this study exposes the non-linearity's of the BPV process.

The results of the system identification of the BPV process proved that the process is non-linear. The process should be treated as non-linear. A linear control system should not be used for the control of this process.

In this mini-dissertation an introduction to system identification and the modelling of MISO systems was also presented. The identification of the MISO system exposed a problem that the prediction error method has. The problem is the inability to find the global minimum.

The results in chapter five show that a control system that consists of different SISO control systems would be less sufficient as a MISO control system.

In this study it is also clear that system identification can not estimate perfect models, but can estimate very accurate models.

6.2 Contributions of study

Mathematical models of SISO and MISO systems were calculated for the BPV, HPE and LPI processes of the PBMR. These models could be used to



MISO system identification techniques using the PBMR data.

Conclusion

design a control system for the PBMR. When the MISO model is used to design the control system, better results will be obtained.

The algorithms that the Matlab toolbox implements to obtain the models are extremely sensitive to the initial parameter settings of the algorithm (with the exception of the ARX model). The settings have a big affect on the local minima that the algorithm settles at.

6.3 Suggestions for future work

The non-linearity of the BPV process resulted in the poor results obtained for the identification of this process. Better models would be obtained when neural networks and fuzzy logic are used to estimate models.

The data that was used in this study was obtained with the aid of the simulation package Flownex. The best models (more robust) will be obtained when the models are calculated from data obtained from the PBMR or PBMM. The data from real life systems will contain noise that will test the models acceptance to noise.

6.4 Concluding remarks

In this dissertation system identification techniques were used to calculate SISO and MISO models of subsystems of the PBMR from observed data. These models could be used to predict the dynamical behaviour of the PBMR. The models that were calculated are of good quality according to the residual



MISO system identification techniques using the PBMR data.

Conclusion

analysis. The models that were calculated are not perfect and are only approximations. It must be kept in mind that these models were calculated in a short period of time, much less than what would be needed to calculate a white box model, the loss of accuracy is acceptable.



List of references

- [1] DORF, R.C., BISHOP, R.H. 2001. Modern control systems (ninth edition), Prentice-Hall, New Jersey, ISBN 0-13-030660-6.
- [2] HOGG, B.W. & LU, S. 2000. Dynamic non-linear modelling of power plant physical principles and neural networks. Electrical power and energy systems, vol. 22, p67-78. [In ScienceDirect: <http://www.sciencedirect.com>].
- [3] DERCKSEN, C.H. 2001. Modelling of some dynamic aspects of the Pebble Bed Modular Reactor using neural networks. Potchefstroom: Potchefstrooms University for Higher Education, School for Electric and Electronic Engineering. (Dissertation - M.Eng.).
- [4] WANG, L., 1994, Adaptive fuzzy systems and control. Design and stability analysis, Prentice-Hall, New Jersey, ISBN: 0-13-147109-0.
- [5] ERASMUS, W.O., 2003. Modelling the Pebble Bed Modular Reactor using system identification techniques. . Potchefstroom: Potchefstrooms University for Higher Education, School for Electric and Electronic Engineering. (Dissertation - M.Eng.).
- [6] LJUNG, L. 1999. System identification: Theory for the user (second edition), Prentice-Hall, New Jersey, ISBN 0-13-656695-2.
- [7] JOHNSON, J.R., 1989, Introduction to digital signal processing, Prentice-Hall, New Jersey, ISBN 0-13-480634-4.
- [8] VARDEMAN, S.B., 1993, Statistics for engineering problem solving, PWS



Publishing Company, Boston, ISBN 0-534-92871-4.

- [9] PHILLIPS, C.L., PARR, J.M., Signals, systems and transforms (second edition), Prentice-Hall, New Jersey, ISBN 0-13-095322-9.
- [10] OXFORD UNIVERSITY PRESS, 1998, The oxford essential thesaurus, Oxford University press, Oxford, ISBN 0-42-516421-7.
- [11] BROWN, R.G., HWANG, P.Y.C., 1992, Introduction to random signals and applied kalman filtering, John Wiley & sons, Inc., United States of America., ISBN 0-471-12839-2.
- [12] HOUCK, S., POWELLI, J.D., 2000, Visual, cruise formation flying dynamics, Stanford University, <http://waas.stanford.edu>.
- [13] HEUNECKE, O., WELSCH, W., 2000, Terminology and classification of deforming models in engineering surveys, Journal of Geospatial engineering, Vol. 2, No. 1, pp 35-44.
- [14] HEUNECKE, O., 1995. On the identification and verification of deformation processes by adaptive KALMAN-Filtering (Hannover Filter). *Wissenschaftliche Arbeiten der Fachrichtung Vermessungswesen der Universität Hannover*, No. 208, Hannover (in German).
- [15] WELSCH, W., 1996. Geodetic analysis of dynamic processes: classification and terminology, *8th FIG International Symposium on Deformation Measurements*, Hong Kong, pp.147-156.
- [16] O'DWYER, A.: 'The estimation and compensation of processes with time delays', *Ph.D. thesis*, 1996, Dublin City University, Dublin 9, Ireland.



- [17] O'DWYER, A, Ringwood, J.V. 1999 , A classification of techniques for the estimation of the model parameters of a time delayed process, Proceedings of the Irish Signals and Systems Conference, National University of Ireland.
- [18] RENDERS, J.-M. and FLASSE, S.P.: 'Hybrid methods using genetic algorithms for global optimisation', *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, 1996, **26**, pp. 243-258.
- [19] YANG, Z.-J., HACHINO, T. and TSUJI, T.: 'Model reduction with time delay combining the least-squares method with the genetic algorithm', *IEE Proceedings - Control Theory and Applications*, 1996, **143**, pp. 247-254.
- [20] YANG, Z.-J., HACHINO, T. and TSUJI, T.: 'On-line identification of continuous time-delay systems combining least-squares techniques with a genetic algorithm', *International Journal of Control*, 1997, **66**, pp. 23-42.
- [21] SCHÖN, T., 2001, Identification for Predictive Control- A Multiple Model Approach, Master's Thesis, Division of Automatic Control, Department of Electrical Engineering, Linköping University, Sweden.
- [22] WAHLBERG, B. and LJUNG, L. 1986, Design variables for bias distribution in transfer function estimation. *IEEE Transactions on Automatic Control*, AC-31(2):134-144.



List of references

- [23] ABONYI, J., TARr, J., SZEIFERT, F., Identification of MIMO Processes by Fuzzy Clustering, University of Veszprem, Department of Process Engineering, [<http://www.fmt.vein.hu/softcomp>]
- [24] Coal in South Africa, Escom, <http://www.escom.com>
- [25] KENSING, V., Black-Box Model development of the JAS 39 Gripen fuel tankpressurization system, Master's thesis, Linköping University.
- [26] GLAD, T., LJUNG L, *Modellbygge och simulering*, Studentlitteratur, 1991.
- [27] LINDSKOG, P., 1996, A system identification software tool for general MISO ARX-type model structures, Linköping University.
- [28] SÖDERSTRÖM, T., STOICA, P., System identification, Prentice Hall International, 1989.
- [29] ODWYER, A., 2000, A survey of techniques for the estimation and compensation of process with time delay, Dublin Institute of Technology.
- [30] ANDERSON, P., Identification of civil engineering structures using vector ARMA models, Aalborg University, <http://www.civil.auc.dk/~i6pa/thesis.htm>.
- [31] HUI, K., LU, J.C.S., SUN, K.C., A case study of furnace modeling from a Wafer Fab's perspective, Semiconductor Fabtech.
- [32] GREYVENSTEIN, G.P., ROUSSEAU, P.G., 2002, Design of a physical model of the PBMR with the aid of Flownex, Nuclear Engineering and design.



MISO system identification techniques using the PBMR data.

List of references

- [33] GODFREY, K.R., McCORMACK, A.S., Flower, J.O., 1995, Applying system identification using commercially available software and hardware, University of Warwick.
- [34] LJUNG, L., System identification toolbox, Mathworks Inc., Cochituate Place, Natick.
- [35] HAYKIN, S., Neural networks, second edition, Prentice Hall, New Jersey, 1999.

