PRACTICAL ON-LINE MODEL VALIDATION FOR MODEL PREDICTIVE CONTROLLERS (MPC)

BY

YUBANTHREN TYRIN NAIDOO

Bachelor of Science in Engineering
(Chemical Engineering)

Submitted in fulfillment of the academic requirements for the degree of Master of Science in Engineering to the Faculty of Engineering, School of Chemical Engineering, University of KwaZulu-Natal, Durban.

UNIVERSITY OF
KWAZULU-NATAL

December 2010

In loving memory of

Thynagee Ammal Naidoo

15 February 1928 – 01 October 2009

‘Grace to you and peace from God our Father
and the Lord Jesus Christ. I give thanks to my God
upon every remembrance of you’
~ Philippians 1: 2-3

ABSTRACT

A typical petro-chemical or oil-refining plant is known to operate with hundreds if not thousands of control loops. All critical loops are primarily required to operate at their respective optimal levels in order for the plant to run efficiently. With such a large number of vital loops, it is difficult for engineers to monitor and maintain these loops with the intention that they are operating under optimum conditions at all times. Parts of processes are interactive, more so nowadays with increasing integration, requiring the use of a more advanced protocol of control systems. The most widely applied advanced process control system is the Model Predictive Controller (MPC). The success of these controllers is noted in the large number of applications worldwide. These controllers rely on a process model in order to predict future plant responses.

Naturally, the performance of model-based controllers is intimately linked to the quality of the process models. Industrial project experience has shown that the most difficult and time-consuming work in an MPC project is modeling and identification. With time, the performance of these controllers degrades due to changes in feed, working regime as well as plant configuration. One of the causes of controller degradation is this degradation of process models. If a discrepancy between the controller's plant model and the plant itself exists, controller performance may be adversely affected. It is important to detect these changes and re-identify the plant model to maintain control performance over time.

In order to avoid the time-consuming process of complete model identification, a model validation tool is developed which provides a model quality indication based on real-time plant data. The focus has been on developing a method that is simple to implement but still robust. The techniques and algorithms presented are developed as far as possible to resemble an on-line software environment and are capable of running parallel to the process in real time. These techniques are based on parametric (regression) and non-parametric (correlation) analyses which complement each other in identifying problems within on-line models. These methods pinpoint the precise location of a mismatch. This implies that only a few inputs have to be perturbed in the re-identification process and only the degraded portion of the model is to be updated. This work is carried out for the benefit of SASOL, exclusively focused on the Secunda plant which has a large number of model predictive controllers that are required to be maintained for optimal economic benefit. The efficacy of the methodology developed is illustrated in several simulation studies with the key intention to mirror occurrences present in industrial processes. The methods were also tested on an industrial application. The key results and shortfalls of the methodology are documented.

PREFACE

The work presented in this thesis was carried at the University of KwaZulu-Natal, Durban, from January 2009 to December 2010 under the supervision of Professor M. Mulholland. This thesis is submitted as the full requirement for the degree M.Sc. in Chemical Engineering. I, Yubanthren Tyrin Naidoo, therefore declare that:

- The research reported in this thesis, except where otherwise indicated, is my original work.
- This thesis has not been submitted for any degree or examination at any other university.
- This thesis does not contain other persons' data, pictures, graphs or other information, unless specifically acknowledged as being sourced from other persons.
- This thesis does not contain other persons' writing, unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted, then:
    - Their words have been re-written but the general information attributed to them has been referenced;
    - Where their exact words have been used, their writing has been placed inside quotation marks, and referenced.
- This thesis does not contain text, graphics or tables copied and pasted from the Internet, unless specifically acknowledged, and the source being detailed in the thesis and in the References sections.

Y.T Naidoo

As the candidate's supervisor, I, Prof. M. Mulholland, approved this thesis for submission.

Prof. M Mulholland

ACKNOWLEDGEMENTS

I have worked with a great number of people whose contribution in assorted ways to the research and the making of the thesis deserved special mention. It is a pleasure to convey my gratitude to them all in my humble acknowledgment.

In the first place I would like to record my gratitude to Professor Michael Mulholland for his supervision, advice, and guidance from the very early stage of this research as well as giving me extraordinary experiences throughout the work. Above all and the most needed, he provided me unflinching encouragement and support in various ways. His truly renowned intuition has made him a constant oasis of ideas which exceptionally inspired and enriched my growth as a student and a future chemical engineer. I am indebted to him more than he knows.

I would like to extend special thanks to the staff at the school of Chemical Engineering for the hospitality and encouragement during my postgraduate studies. To the role models for hard workers in the lab, Sadha and Rekha, I would like to thank them for their unwavering assistance during my time as a practical demonstrator. I offer a humble thank you to the lecturers in the school of chemical engineering, especially Ahmed Bassa, for instilling in me knowledge which has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

An abundance of gratitude is extended to SASOL LTD for their financial support throughout my undergraduate and postgraduate studies. Without this support I would not made accomplished all that I have. A special thank you is extended to Morne Booysen from the Advanced Solutions Department at the SASOL Secunda plant who served as my mentor.

I gratefully thank Ashveer, Taswald and especially Nilesthra for their constructive comments on this thesis. I am thankful that in the midst of all their activity, they offered their assistance to help me. I would like to extend a special thank you to Nilesthra for her calming words of inspiration in my times of doubt, truly a friend for this lifetime and the next. Throughout my tenure as a student at this university as well as throughout my schooling career I have made numerous friends from whom I have learnt so much. My best friends, Alicia and Ashveer have always been there throughout my studies and my life as a whole. I am eternally appreciative of them. I express my sincere apology that I could not mention each one personally one by one, but I am grateful to you all. A special thank you is extended to Andrea, whose persistent dedication and understanding has truly been my beacon of light in recent times.

Where would I be without my family? Words fail to express my deep and loving appreciation to my family for their support throughout my studies. To my uncles and aunts, I offer my humble thanks for everything that you have done for me, especially my uncles Balan, Jay and my late uncle Morgan and Aunt Sharon. It is said that dreams seldom do come true; in my case I have been truly blessed with family who have unselfishly made my dreams come true. A special thank you is extended to my grandfather and grandmother, Reggie and Sheila; they serve as the foundation from which my astounding family draws their strength and character. To my big brother, Jerome, thank you for being my big brother, for your care and true understanding, for the words not said when words could not mend.

A mother’s heart and a mother's faith and a mother's steadfast love were fashioned by the Angels and sent from God above. I dedicate this work my inspiration, my pillars of strength, to my mum, Evonne and my late grandmother, Thynagee. No more encouraging words have ever been written than those of Paul to the Church at Philippi as he encouraged them in life, thought, and spirituality. My mum personifies these very words. Life has never been easy but through her prayer and her desire to live in the path of God’s word, we overcame every obstacle put forth. My grandmother, Thynagee, is the epitome of an Angel of god. Her family always remained her priority. Her entire life had been dedicated to our wellbeing. She is beyond doubt missed and her legacy will live on in the hearts of her family, thank you for raising me with your values and courage to dream.

Lastly, to God I owe all the glory and honor for blessing me with an amazing family and for guiding me in Your path. The source of my existence - Jesus Christ, my Lord and Personal Savior, I love you Lord.

TABLE OF CONTENTS

CHAPTER 1 - INTRODUCTION

CHAPTER 2 - PRELIMINARIES

CHAPTER 3 - IDENTIFICATION PRINCIPLES

CHAPTER 4 - MODEL VALIDATION THEORY

CHAPTER 5 - LINEAR REGRESSION THEORY

CHAPTER 6 – EXPERIMENTAL DESIGN THEORY

CHAPTER 7 – SIMULATION STUDIES

CHAPTER 8 – INDUSTRIAL CASE STUDY

CHAPTER 9 – CONCLUSIONS

CHAPTER 10 – RECOMMENDATIONS

APPENDICES

LIST OF FIGURES

CHAPTER 1

CHAPTER 2

CHAPTER 3

CHAPTER 4

CHAPTER 5

CHAPTER 6

CHAPTER 7

CHAPTER 8

APPENDIX B

APPENDIX C

LIST OF TABLES

NOMENCLATURE

The frequently used symbols in this Thesis are included in the following list. The vectors are written in lower case bold and matrices in upper case bold. The individual elements of a matrix are written in lower case of the same symbol as used for the matrix.

ENGLISH LETTERS

A, B, C, D, F	Lag polynomials
a, b	Discrete time coefficients
c, d	Continuous time coefficients
$\boldsymbol{d}$	Disturbance signals
e	Output error
F(s)	Laplace Transform
F(jw)	Fourier Transform
G	Dynamic system model transfer function
G(q, θ)	Parameterized system transfer function
G^o	True system transfer function
g	Impulse response model
H	Noise model
H(q, θ)	Parameterized noise model transfer function
H^o	True noise transfer function
h	Impulse response noise model
J	Objective function for moving window regression
K	Steady-state gain
K^*	Kalman Filter gain
k	Regression fitting parameter

M	Cross – correlation lag range
m	Control Horizon
m	Number of inputs
N	Number of sample points
n	Number of outputs
P	Covariance matrix in Kalman Filter algorithm
p	Prediction Horizon
Q	State weighting matrix
q	Minimum cost horizon
q	Time shift operator
R	Normalized correlation coefficient
R	Control Weighting matrix
r	Covariance correlation coefficient
T	Sampling interval
u	Input signal
V	Criterion function to be minimized
W	Simplified model representation notation
w	Measured disturbance
y	Output signal
Z	Data set
z	Signal reflecting tracking error or control action

ACRONYMS

APC	Advanced Process Control
AR	Auto-Regressive
ARMAX	Auto-Regressive Moving-Average eXogenous
ARX	Auto-Regressive eXogenous
ASYM	Asymptotic Method

BJ	Box-Jenkins modeling
CTL	Coal-to-Liquid processing
CPC	Conventional Process Control
CV	Control Variable
DMC	Dynamic Matrix Control
FIR	Finite Impulse Response
DV	Disturbance Variable
GD	Gradient Descent
GM	Global Method
GN	Gauss-Newton
HIECON	Hierarchical Constraint Control
IDCOM	Identification and Command
KF	Kalman Filter
LP	Linear Programming
LQG	Linear Quadratic Gaussian
LQR	Linear Quadratic Regulator
LSS	Linear State Space
MA	Moving Average
MBC	Model-Based Control
ML	Multi-Level
MLS	Modified Least Squares
MM	Multi Model
MPC	Model Predictive Control
MPHC	Model Predictive Heuristic Control
MPM	Model-Plant Mismatch
MV	Manipulated Variable
MWR	Moving Window Regression
NAC	Non-Acidic Chemical
OE	Output Error
PCT	Predictive Control Technology
PEM	Prediction Error Method

PV	Process Variable
QDMC	Quadratic Dynamic Matrix Control
QP	Quality Performance
RA	Regression Analysis
RLS	Recursive Least Squares
RMPCT	Robust Multivariable Predictive Control Technology
SMCA	Setpoint Multivariable Control Architecture
SMOC	Shell Multivariable Optimizing Control
TF	Transfer Function

GREEK LETTERS

Γ	Model Transform
Δ	Differential form
Λ	Weighting matrix for fitting parameter suppression
α	Error form for regression
β	Input form for regression
γ	Discrete time approximation factor
δ^*	Impulse
$\mathrm{\delta}$	Input form with the impact of other inputs removed
ε	Gaussian White Noise
ϵ	Error with the impact of other inputs removed
ζ	Data index set
η	Regression fitting parameter multiplicative factors
θ	Parameter vector
π	Minimum Variance Index
ρ	Partial correlation coefficient
τ	Time delay constant
υ	Noise representation

SUBSCRIPTS AND SUPERSCRIPTS

Subscripts

d	disturbance
e	error
i	i^{th} component or i^{th} time interval
m	model
max	Upper constraint limit
min	Lower constraint limit
na, nb, nc, nd, nf	Lag polynomial indices
opt	Optimal sampling interval
p	plant
u	Input

Superscripts

MV	Minimum Variance
N	Number of sampling points

CHAPTER 1 INTRODUCTION

This chapter presents the introduction to this work. It begins with an overview of the main areas of focus in this thesis by providing a general idea and brief background into the field of Model Predictive Control in industry and the necessity of closed-loop on-line model validation. The chapter thereafter delves into the major objectives, research contribution and scope. The last section of this thesis presents the structure to follow, providing an indication as to what each subsequent chapter entails.

1.1. OVERVIEW

In a society where the needs of the people are insatiable, the demands put on consumer products and products for the industrial markets are ever increasing. The highest quality is demanded at the lowest possible price. Many companies are competing to fulfill the day to day needs of both consumers and industries. As a result of increasing demands in production efficiency, industry has turned to more Advanced Process Control systems (APC). These control systems can be encountered in chemical and petrochemical processes such as oil refining, production of petrol and synthetic by-products. One class of advanced controllers is Model Predictive Control (MPC).

Model-based Predictive Control constitutes a class of computer algorithms that make direct use of a process model in order to predict the future response of the plant (Seborg, Edgar and Mellichamp, 2004). Coupled with the capacity to handle multiple-input, multiple-output systems (MIMO), the attractiveness of MPC, to an extent, is owed to the aptitude of MPC algorithms to handle constraints that are frequently met in industrial processes. These constraints are not particularly well addressed within other control approaches (Kocijan, 2003).

Central to the success of the MPC technique is the derivation of accurate process models (Huang and Tamayo, 2000). Modeling and identification has been shown to be the most intricate and time-consuming work in an MPC project (Zhu et al., 1997). These models, which are generally identified at the commissioning stage, are never accurate and contain some mismatch with the plant, due to the inherent non-linearity of most chemical processes. The models are typically linear and thus represent these processes over a restricted range around the operating point. Consequently, changes in the desired plant dynamics together with the non-linearities of the process magnify the gap present between the model and the plant itself which may subsequently lead to a drop in MPC performance (Qiang and Shaoyoun, 2005).

The impact of poor controller performance is immense in that if affects product quality, plant economy and safety. It can be noted that in many industries, intrusive open-loop plant tests, often associated with the re-identification of models, entail high costs due to large production losses. These facts stress the necessity for the development of efficient on-line techniques that are capable of identifying a Model-Plant Mismatch (MPM) under normal operating conditions, especially for MIMO systems. This would go a long way in providing a model quality indication and assist, if necessary, in re-identification and tuning of the controller. A continued maintenance program would maintain the productivity gains from MPC (Brisk, 2004).

1.2. BACKGROUND

Performance monitoring of controllers is a field of interest for academia and industry. However in the case of MPC, the focus has primarily been to ensure for optimal control. According to a recent survey, there have been noted to be over 5500 industrial applications of MPC worldwide (Qin et al., 2003). Various studies indicate that as many as 66% of controllers have some kind of performance problem (Miller, 2001). The major causes of poor control loop performance in Model-Based Control systems (MBC) are depicted in figure 1.1 below:

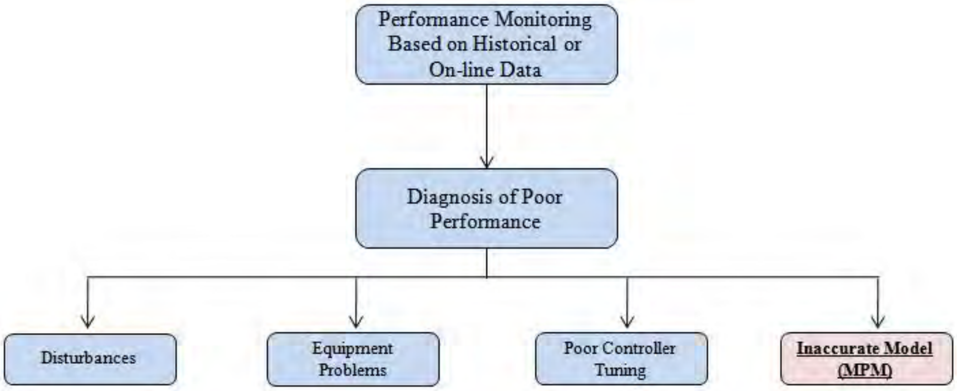

Performance Monitoring Based on Historical or On-line Data
Diagnosis of Poor Performance
Disturbances
Equipment Problems
Poor Controller Tuning
Inaccurate Model (MPM)

Figure 1.1: Information flow diagram for causes of poor controller performance

The majority of the relevant work in the field of controller performance assessment has focused primarily on obtaining statistics for performance evaluation from routine plant data. There are no systematic ways of detecting the underlying issues as shown in figure 1.1 above if these statistics provide an indication of poor performance. Considering the extensive application of MPC systems, it is to some extent rather surprising that the area of diagnostics remains a largely unsolved problem (Loquasto, 2003).

This thesis focuses primarily on the detection and diagnosis of Model-Plant Mismatch (MPM). Loquasto (2003) acknowledges that the variation in plant characteristics are possibly due to feed-stocks changes or a corresponding change feed flow-rates, changes exhibited in operating conditions or product grade, changes in process variations for example fouling in heat exchangers and catalyst deactivation, environmental variations such as weather changes, and so forth. If such a change occurs, the process model(s) may no longer describe the plant behavior adequately.

When performance of a Model-Based Controller degrades, the first step would therefore be to correct for any MPM, due to the performance of such controllers being intimately linked to the quality of the process models. It has been argued that on the basis of the MPC controller tuning, performance may be improved in the presence of MPM (Schafer, 2004). However, from a strictly pragmatic view, the likelihood of achieving satisfying controller performance through retuning increases if the plant model is correct.

1.3. MOTIVATION

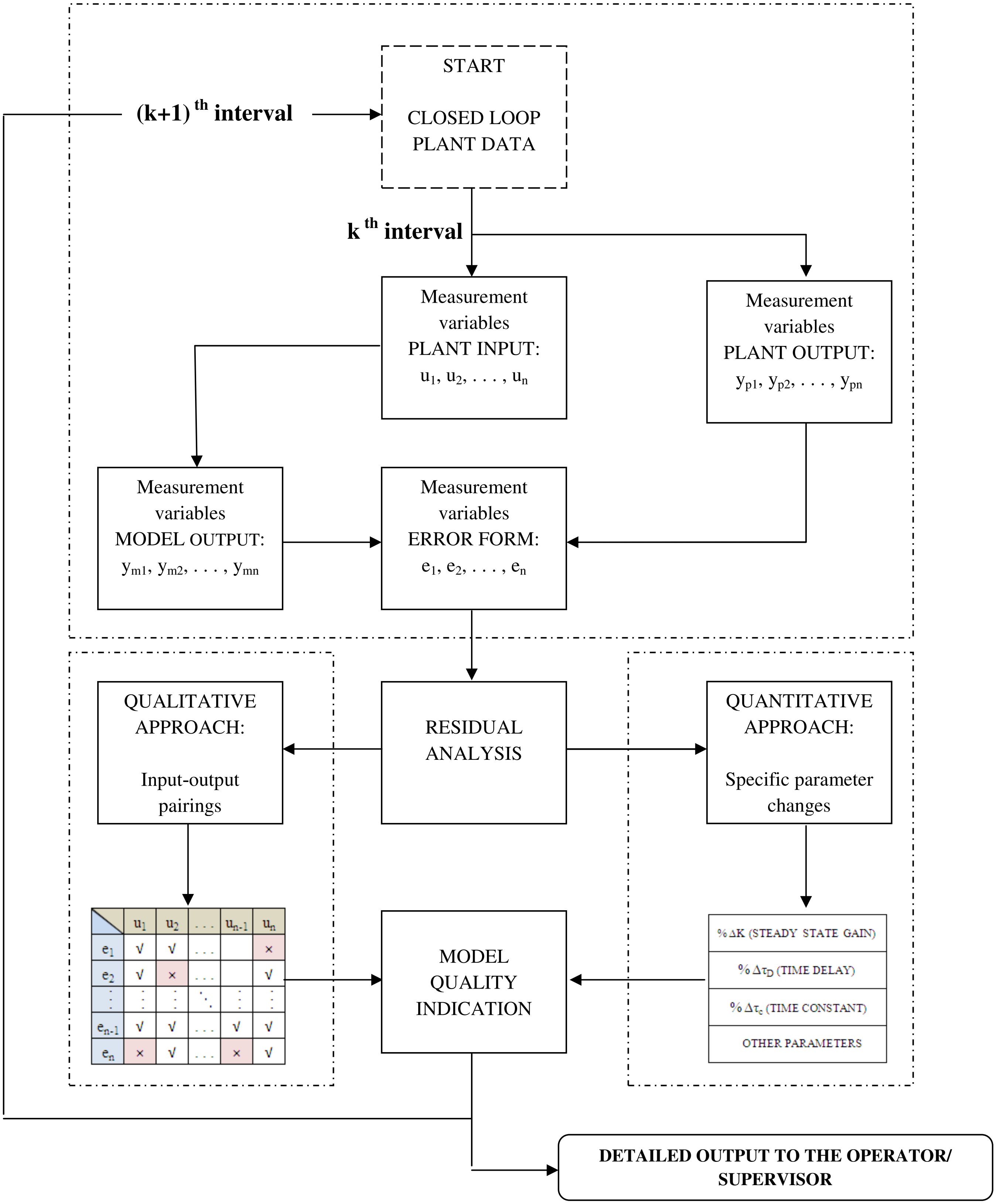
It is reiterated that identification of models is regarded as a very time consuming process and becomes an even more complex process for large MIMO controllers. For example, a 10 × 10 MIMO system would contain 100 univariate models. A full plant test can take 5-15 days to complete (Qin et al., 2003), depending on the size of the MIMO control system. These plant tests involve step testing each input independently in order to determine the relevant dynamical relationship between inputs and outputs. Webber and Gupta (2008) state that large cost savings can be made if one is capable of detecting certain specific input–output pairings that contain the mismatch. This implies that the re-identification process would require a few inputs to be perturbed and can thus focus only on the subset of models that require re-identification.

The work presented in this thesis was carried out for SASOL for their Secunda plant located in Mpumalanga. Applying proven Multivariable Predictive Control coupled with robust online product quality predications has proven to maximize profitability for a large number of units in the SASOL Secunda plant. These controllers are required to be maintained to ensure optimal benefit.

1.4. APPROACH

The role of model validation during the identification process is to guarantee that the delivered model contains the most significant dynamics of the process. Model validation at this stage is normally qualitative. After the designed controller has been implemented, a more challenging task is: how to continuously monitor the model quality under closed-loop conditions.

The proposed methodology is depicted in figure 1.2 below and is based on residual analysis. The techniques and algorithms proposed are developed as far as possible to resemble an on-line software environment. These techniques are based on quantitative (regression) and qualitative (correlation) analyses which complement each other in identifying problems within on-line models.


START
CLOSED LOOP PLANT DATA
(k+1) th interval
k th interval
Measurement variables PLANT INPUT: $u_1, u_2, \ldots, u_n$
Measurement variables PLANT OUTPUT: $y_{p1}, y_{p2}, \ldots, y_{pn}$
Measurement variables MODEL OUTPUT: $y_{m1}, y_{m2}, \ldots, y_{mn}$
Measurement variables ERROR FORM: $e_1, e_2, \ldots, e_n$
RESIDUAL ANALYSIS
QUALITATIVE APPROACH: Input-output pairings
QUANTITATIVE APPROACH: Specific parameter changes
% ΔK (STEADY STATE GAIN)
% $\Delta\tau_D$ (TIME DELAY)
% $\Delta\tau_c$ (TIME CONSTANT)
OTHER PARAMETERS
MODEL QUALITY INDICATION
DETAILED OUTPUT TO THE OPERATOR/ SUPERVISOR

	u_1	u_2	. . .	u_{n-1}	u_n
e_1	√	√	. . .		×
e_2	√	×	. . .		√
⋮	⋮	⋮	⋱	⋮	⋮
e_{n-1}	√	√	. . .	√	√
e_n	×	√	. . .	×	√

Figure 1.2: Proposed scheme for Model validation of MIMO systems

Correlation analysis based on the correlation between the prediction error and corresponding model inputs, developed by Webber and Gupta (2008), is capable of detecting which specific input–output pairings of a model-based controller, within a model matrix, are mismatched. This method may be employed to screen an entire model set and thereby select candidate models for re-identification. Delving further into the sphere of model validation, knowing which parameters within a certain model (albeit the steady state gain or the time delays) are mismatched, would provide the maintenance engineer with extensive information needed when he is required to re-identify mismatched models. It should be noted that SASOL presents their models in an input-output form (refer to chapter 5, section 5.1). Thus there would be no interaction amongst outputs in this model formulation because their interactive contributions are effectively dealt with by substituting their behaviour in terms of the inputs. This effectively implies that an n x m MIMO system can be considered as n separate MISO systems, allowing for simpler computation. The proposed methodology is demonstrated via representative simulation examples as well as an industrial case study.

1.5. OBJECTIVES

The principal objective for this thesis is to develop a tool that is capable of providing model quality indications for the models present in a MIMO system coupled with indications as to which models and correspondingly which parameters within these models are mismatched.

This tool should be easy to implement and applicable to different types of plants. The methods developed within the tool should be automatic in order to run in parallel with an industrial process.

The research objectives are broken down and given by the following points:

- Compile an extensive literature survey on pertinent system identification principles and model validation approaches.

- Specific focus should be on model validation techniques that are capable of being implemented in an on-line environment.

- Specifically concentrate on cross-correlation and regression techniques and use simulation models to test the applicability of the tool developed via Monte Carlo simulations.

- Extend these simulations to emulate industrial situations in order to obtain an understanding regarding practical issues that may arise when the model validation techniques are implemented.

- Test model validation techniques on an industrial case study provided by SASOL for a petrol debutanizer unit.

1.6. RESEARCH CONTRIBUTION

Model validation is a very interesting topic of great practical relevance. Although it has been studied over the past two decades (Soderstrom, 1993), there are still many open problems to address. The conventional method, under open-loop conditions, is based on residual correlation analysis (Ljung, 2002). Many industrial control engineers do not even use most open-loop validation techniques developed. Substitution of off-line techniques with the on-line closed-loop methodology could save SASOL, as well as other companies in the same field, valuable time and production cost. This will in turn ensure the success of the MPC technique.

1.7. SCOPE

This thesis focuses exclusively on performance of MPC controllers being affected by MPM. Diagnosis of other issues, depicted in figure 1.1, resulting in a degradation of controller performance lie beyond the scope of this thesis. However, if it is found that controller performance has dropped and the technique developed does not recognize that it is as a result of MPM, then the maintenance engineer can zero in on the other factors.

The model validation tool is developed and implemented in MATLAB ®. The techniques are designed, as far as possible, to represent an on-line software environment. Success of the methodology would result in the next stage of this topic being developed in VISUAL C++, and hence is not requirement for this thesis.

The argumentation will be based on a simulation study and real data provided by SASOL will be tested. The method is to be used on runtime data. A verification of the method would require plant data, the faulty model and a new updated plant model for comparison. Although provision was made for real-time data for a specific Model-Predictive Controller from the Secunda plant, verification based on additional industrial data will not be achieved as the controller provided has been decommissioned due to modifications on the plant.

1.8. THESIS ORGANIZATION

This thesis begins with an introduction of the main areas of focus by providing an overview and brief background into the field of Model-Predictive Control in industry and the necessity of closed-loop on-line model validation. Chapter 2 introduces the development of the SASOL Secunda plant and products and the turn to advanced process control. Thereafter, the chapter introduces Model-Predictive Control as an innovative means of industrial control, highlighting its history and its concepts. Chapter 3 presents the fundamental theory related to the topic. This theory focuses on the background of model representation and model identification. Chapter 4 offers insight into the related work for closed-loop model validation techniques. Chapter 5 deals with the theory and algorithms developed for the proposed methodology. The regression techniques are introduced and developed. Key industrial situations such as correlation amongst inputs, absence of disturbances and noise levels are addressed and modifications are made to the techniques to deal with the aforementioned industrial situations. Chapter 6 deals with the development of the algorithms within MATLAB ® as well as the challenges faced. Chapter 7 covers simulations on the methods developed in the previous chapters. Simulations are performed on a heavy oil fractionators and a Continuous Stirred Tank Heating (CSTH) system. Key industrial occurrences are simulated and the results compiled and rules of thumb are developed for diagnosis of significant MPM. The effect of signal-to-noise ratio is also tested and a limit to this ratio is proposed. Chapter 8 focuses on the real-plant data provided for a petrol debutanizer unit which is tested and key results are noted. Chapter 9 presents the relevant conclusions related to the research and results obtained. Chapter 10 highlights the recommendations and what is required to implement the techniques online.

CHAPTER 2 PRELIMINARIES

This chapter introduces the development of the SASOL Secunda plant, products and its turn to advanced process control. Thereafter, the chapter introduces Model-Predictive Control as an innovative means of industrial control as opposed to conventional control and provides a brief history of the different types of MPC. The concept of MPC is then introduced and the importance of the process model in MPC techniques is highlighted.

2.1. SASOL INDUSTRY AND ADVANCED PROCESS CONTROL

The SASOL consortium consists of diversified fuel, chemical and related manufacturing and marketing operations. This thesis serves as a project for the SASOL Secunda plant located in Mpumalanga. This 14 km² site is known to be the world's largest production facility of synthetic fuels (synfuels) with its core feedstock obtained from coal. Depicted in figure 2.1 is an aerial view of the Secunda plant. The east and west sections of this plant are exact replicas of each other.

Apart from being the universal leader in Coal-to-Liquid (CTL) gas production, SASOL’s Secunda plant has a diverse range of business operating sectors. These can be broadly divided into the following 4 subdivisions (SASOL PTY LTD, 2010):

The Polymers business operates several plants such as a polypropylene plant as well as a combined ethylene cracking and separation plant. Solvents (inclusive of 12 smaller plants) serve to remove Non-Acidic Chemicals (NACs) from synthol wash water and subsequently converting them into possible marketable products. The Carbo-tar facility processes raw tar and several carbon products. Lastly, the Nitro business comprising 3 plants; its primary function is to produce fertilizers, but it also includes a special facility for the development of explosives.

Globe, Cnes/Spot Image, GeoEye, Map data ©2010 AfriGIS (Pty) Ltd, Tele Atlas, Tracks for Africa

Figure 2.1: Aerial view of the SASOL Secunda plant

These various business sectors require rigorous and efficient control systems in place in order to achieve day-to-day economic targets. SASOL, amongst the majority of industrial companies, have thus turned to Advanced Process Control systems as opposed to Conventional Process Control systems (CPC) in order to cater for the ever-demanding need for financial dominance. In control theory, Advanced Process Control is a broad term implying different kinds of process control tools, often used for solving multivariable control problems (Hasseloff et al., 2007). These include statistical process control, neural network control, and model predictive control.

Advanced Process Control and optimization technologies are applied in industrial processes and in this plant specifically for the following purposes:

- APC technology is capable of maximizing the throughput as well as adhering to operational and safety restrictions thereby increasing operating profits and operational efficiency.

- Techniques applied in APC squeeze the variation of main variables and push them to their respective operating constraints in order to reduce operation loss.

- APC ensures process units are in a safe operating condition, improving plant availability as well as reducing maintenance and utility costs.

The first of the aforementioned points is highlighted by figure 2.2 below. It should be noted that actual savings depend on energy costs, product values and operating objectives, but many APC projects are known to have a Return on Investment (ROI) period of 3-9 months (Selvanathan and Tangirala, 2010).

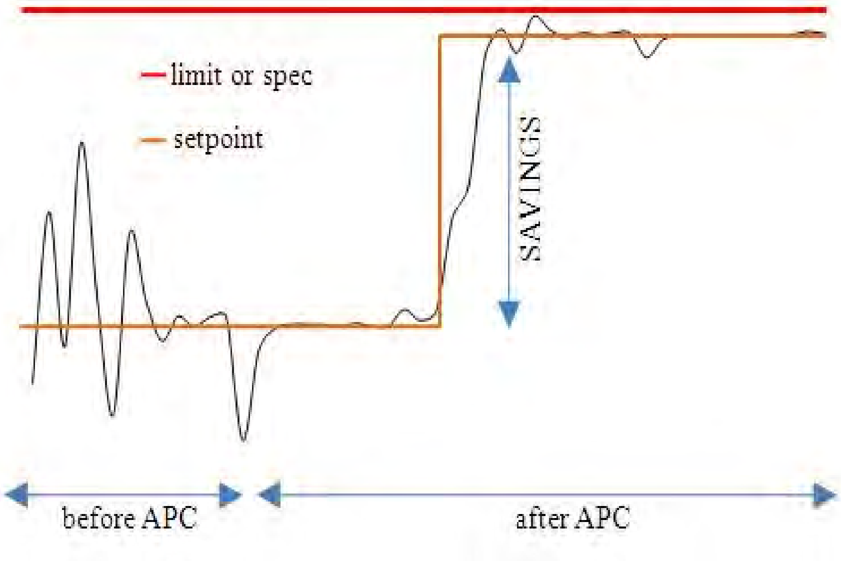


Figure 2.2: Effect of the implementation of an APC control system

2.2. CONTROL STRUCTURES

A complete control system for large scale industrial processes is a combination of several control layers, each with different priorities. Figure 2.3 illustrates this concept, showing a conventional control structure on the left (unit 1) and an APC structure on the right (unit 2), e.g. an MPC structure.

As can be seen, a plant-wide optimizer which is representative of the first control layer evaluates and implements optimal steady-state settings for both units in the plant. This is referred to as the linear programming (LP) layer. This takes material supply and demand, price fluctuation and other economic parameters into consideration and is normally done for a whole industrial site.

These steady-state settings are then sent to the second level, where local optimizers compute further optimal economic steady-states for each unit. Such optimizers operate more regularly than is possible at the above plant-wide level. These optimizers keep the process within specified safe operating ranges and optimize production based on the rules decided in the level above. The steady-states computed at this stage are thereafter passed to base layer control systems (third hierarchical layer) for implementation.

Whilst minimizing any constraint violations experienced along the way, the dynamic constraint control scheme moves the plant from one constrained steady-state to another. The conventional control structure in unit 1 accomplishes this by implementing a distinctive arrangement of PID algorithms, lead-lag (L/L) blocks and high/low select logic. Difficulty is often encountered in embarking on translating the control requirements into an appropriate structure for conventional control.

From a more beneficial stand point, in MPC methodology, the complex inter-connections of base-layer control are replaced by a single MPC controller (unit 2). Coupled with the large savings and the ability to replace a combination of control algorithms, MPC technology is seen to be highly favoured in the control of industrial processes.

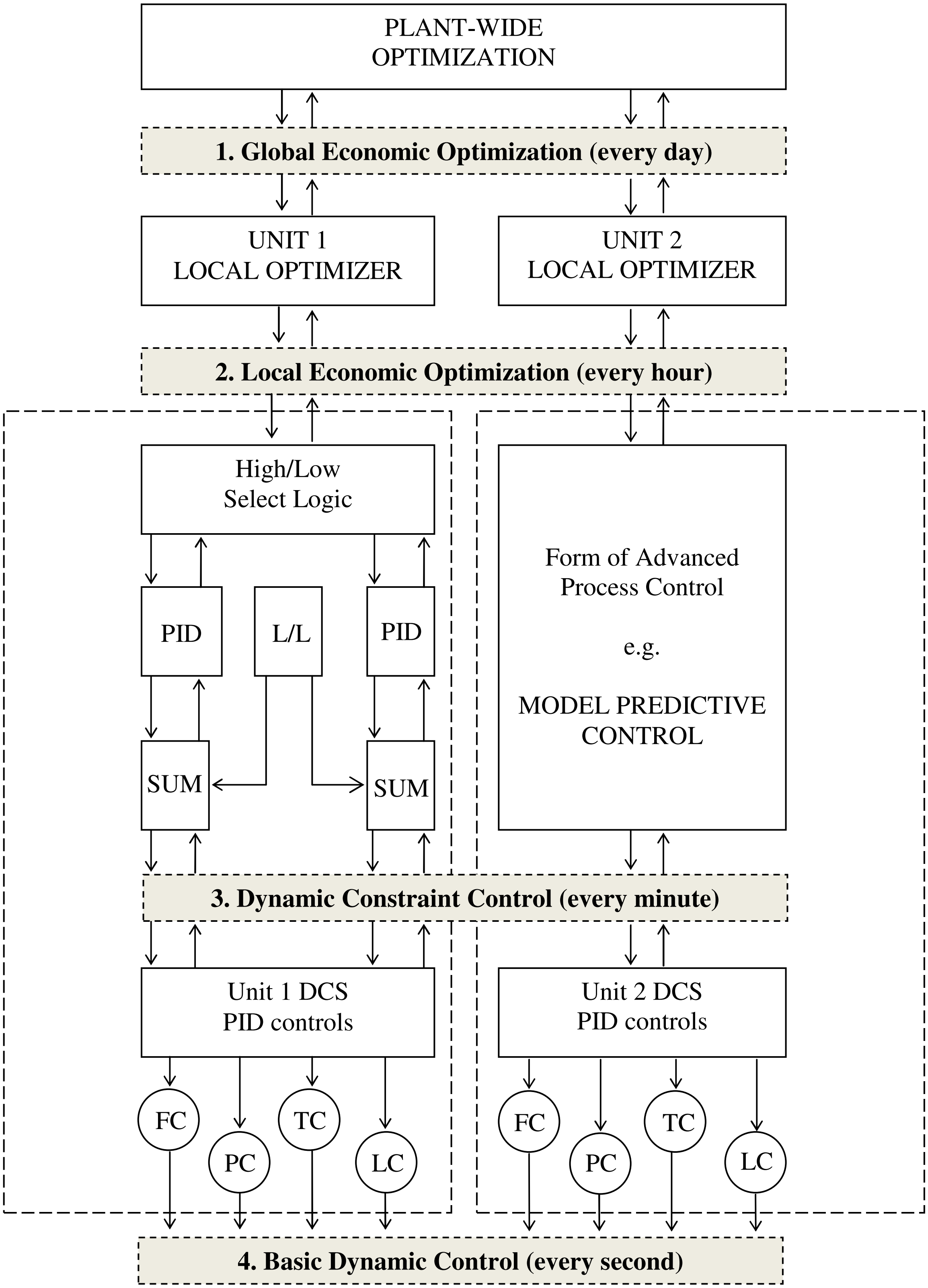

PLANT-WIDE OPTIMIZATION
1. Global Economic Optimization (every day)
UNIT 1 LOCAL OPTIMIZER
UNIT 2 LOCAL OPTIMIZER
2. Local Economic Optimization (every hour)
High/Low Select Logic
PID
L/L
PID
SUM
SUM
Form of Advanced Process Control
e.g.
MODEL PREDICTIVE CONTROL
3. Dynamic Constraint Control (every minute)
Unit 1 DCS PID controls
Unit 2 DCS PID controls
FC
PC
TC
LC
FC
PC
TC
LC
4. Basic Dynamic Control (every second)

Figure 2.3: Hierarchy of control system functions (Qin et al., 2003)

2.3. Model Predictive Control (MPC) Technology

Model Predictive Control is referred to a class of computer control algorithms which make use of an explicit process model to predict the future plant response. MPC has become the new paradigm for industrial process control stemming over the last four decades. This widely applied supervisory technique has gained increasing acceptance due to the following influential factors as indicated by Morari (1989):

- Within its capacity, MPC can handle processes that are unstable, processes containing large dead time shifts as well as non-minimum phase processes.

- MPC can handle constraints employed to provide output products with pre-determined quality specifications in a systematic way irrespective of industrial process limitations such as valve capacity and other technological requirements.

- Finally MPC can adapt to structural changes including equipment failure such as sensor and actuator failures in a limited manner, however its capabilities are inhibited by changes in system parameters and system structure as this represents the heart of MPC operation.

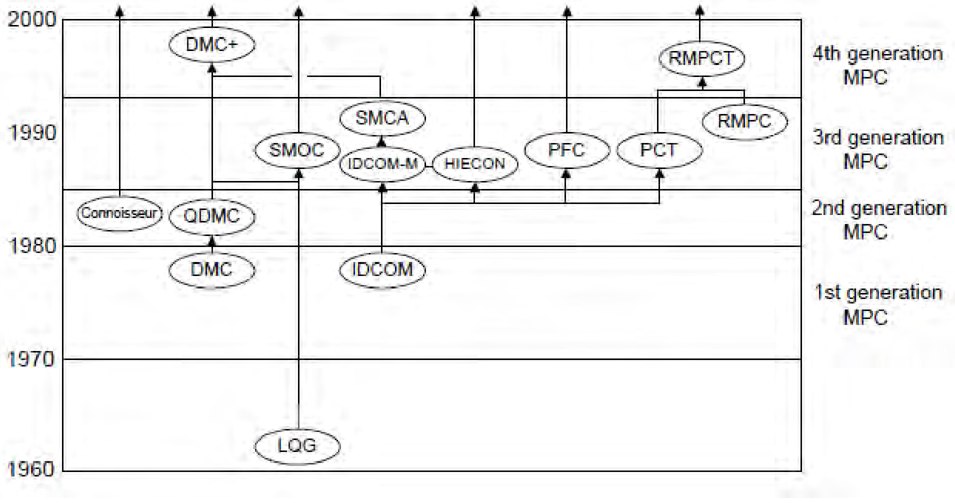


Figure 2.4: Approximate genealogy of linear MPC algorithms (Qin et al., 2003)

2.3.1. Brief History of MPC technology

The development of model-based control used for multivariable control can be traced back to the early 1960s. The most significant industrial algorithms are illustrated in figure 2.4, with emphasis placed on the connections of these algorithms in the evolution tree from generation to generation. From the approximate genealogy shown in figure 2.4, the evolution of MPC technology from basic technology to a technology capable of operating within well defined constraints (Bequette, 1991) is evident. Further details of this development are given below.

The work of Kalman post World War II laid the foundation for the development of modern control concepts (Kalman, 1960). This work entailed the development of an algorithm termed Linear Quadratic Regulator (LQR) which was designed to minimize an unconstrained quadratic objective function of states and inputs. This algorithm was later combined with a Kalman filter and adopted the name Linear Quadratic Gaussian (LQG) controller. The infinite prediction horizon endowed the algorithm with powerful stabilizing properties. Although extensions to deal with practical issues such as obtaining offset-free control and computation of steady-state targets swiftly followed (Kwakermaak, 1972), the LQG algorithm had little impact in the control community. The reason for this lies in the absence of constraints in its formulation, the nonlinearities of the real systems, and above all, the ‘culture of the industrial process control community’ during this period. Control engineers and laboratory technicians lacked foresight or had no exposure to optimal control concepts and thus rendered them impractical.

A number of individuals set out to develop technologies capable of addressing the key concepts noted in the failure of LQG theory. The earliest breakthroughs were made in the late 1970s with the very first generation of MPC technology represented by the IDCOM and DMC algorithms. Their impact on the industrial community was colossal and had immense success in tackling the areas that needed to be addressed and served to describe the ‘industrial MPC paradigm’. The first account of MPC applications during this era were presented by Richalet et al., (1976). Their approach was referred to as Model Predictive Heuristic Control (MPHC). The software package for this approach was called

IDCOM, a contraction for IDentification and COMmand. Three years later, Cutler and Ramaker developed an unconstrained multivariable control algorithm (Cutler and Ramaker, 1979). Their approach was named Dynamic Matrix Control (DMC). Both applications made use of quadratic performance objectives, but differed with the remaining features depicted below:

Table 2.1
Comparison of IDCOM and DMC adapted from Qin et al (2003)

FEATURE	**IDCOM**	**DMC**
Plant Model	Impulse response model for the plant.	Linear step response model for the plant.
Output behavior	Future plant behavior specified by a reference trajectory.	Future plant behavior specified to follow the set point as closely as possible.
Optimal inputs	Optimal inputs computed using heuristic iterative algorithm.	Optimal inputs computed as solutions to a least-squares problem.

Although the first generation of MPC technology provided a large degree of the fundamental progress in the control society, the issue of constraint handling still remained somewhat underdeveloped. This weakness was addressed by posing the DMC algorithm in a manner in which input and output constraints were shown to appear explicitly. The QDMC algorithm, presented by Cutler in a 1983 AIChE conference paper, exhibits similar properties to the original DMC algorithm together with its ability to deal with constraints (Cutler, Morshedi and Haydel, 1983). QDMC represents the second generation of MPC technology.

The challenges and problems faced grew larger and more complex as MPC technology gained wider acceptance. Although the QDMC algorithm provided a systematic approach to handle hard constraints, it could not provide any sustainable results in the case of an infeasible solution. This issue, amongst others, was addressed in the third generation of

MPC technology which included the IDCOM-M, HIECON, SMCA, and SMOC algorithms; others include the PCT algorithm and the RMPC algorithm. These algorithms lead to the following developments:

- These algorithms were capable of distinguishing between several levels of constraints i.e. hard, soft, and ranked.

- They provided some mechanism to recover from an infeasible solution and provided a richer set of options for feedback control.

- They allowed for a wider range of process dynamics (stable, integrating and unstable) and controller specifications.

The only difference amongst these algorithms was that the model for the SMOC (Shell Multivariable Optimizing Controller) algorithm was formulated in a State-Space form as opposed to an Input/Output model formulation. Further details regarding the distinguishing features of these two types of model formulation as well as their major differences can be found in Chapter 3, section 3.2.2.

DMC-plus and RMPCT are representative of the fourth generation of MPC technology and have both been formed as a result of increased competition in the last 15 years. DMC-plus is an Aspen Technology ® product and was formed in early 1996 by merging SMCA and DMC technologies. The RMPC algorithm was merged with the PCT controller to create RMPCT sold by Honeywell. SASOL employs RMPCT technology in its Secunda plant. This fourth generation of MPC technology includes features such as:

- Prioritized control objectives can be addressed as a result of multiple optimization levels and graphical user interfaces.

- Additional flexibility is allowed for quality performance as well as economic objectives.

- This generation allows for direct consideration of model uncertainty in the form of robust control design.

2.4. MPC PRINCIPLES

While the MPC paradigm covers numerous dissimilar variants as depicted in the previous section, each one unique to its desired application, the foundation of all MPC systems lies in the realization of computing future process inputs as solutions to an on-line optimization problem. A process model and process measurements form the basis on which this problem is constructed. Process measurements are obtained attributable to the feedback and, optionally, feed forward constituent in an MPC structure. Figure 2.5 shows the structure of a general MPC scheme. Several possibilities exist for the process model and disturbance prediction, optimization criteria (and objective cost functions), process measurements and lastly constraint handling.

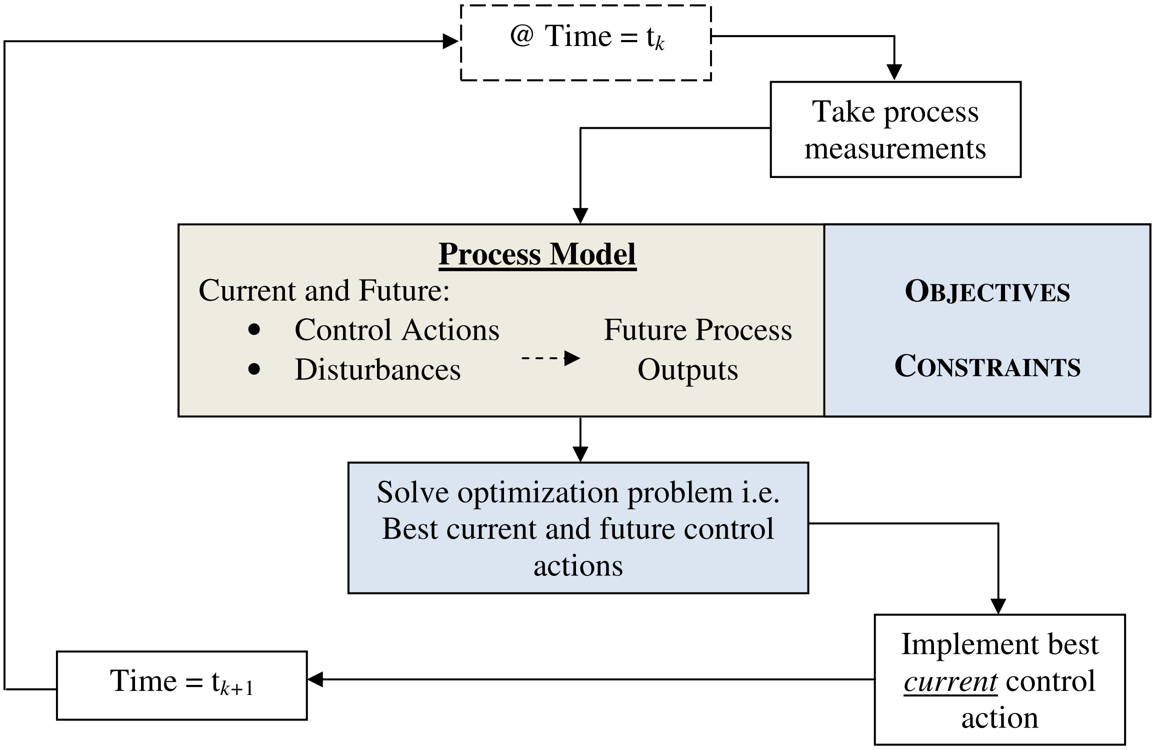

@ Time = t_k
Take process measurements
Process Model
Current and Future:
Control Actions
Disturbances
Future Process Outputs
OBJECTIVES
CONSTRAINTS
Solve optimization problem i.e. Best current and future control actions
Implement best *current* control action
Time = t_{k+1}

Figure 2.5: Model Predictive Control Scheme (Deshpande, 1995)

2.4.1. Structure of MPC

The basic structure of a model predictive controller is shown in figure 2.5 (Deshpande, 1995). An MPC controller is a form of Internal Model Control. Figure 2.6 shows a schematic of IMC configuration where G_p, G_m, Q, and G_d denote process, model, controller, and disturbance transfer functions, respectively. The observed output is denoted by $y_{\mathrm{p}}(k)$, while the prediction of the model is denoted by $y_m(k)$ respectively.

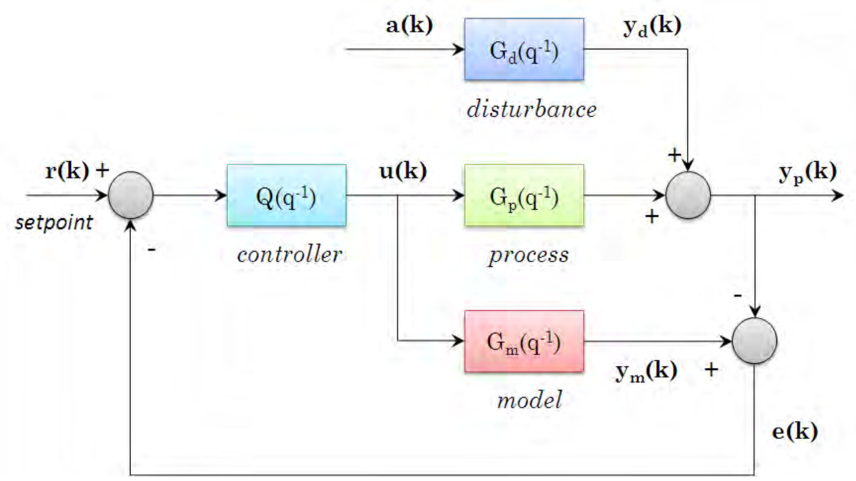


Figure 2.6: Schematic representation of internal model control (Ljung, **1999**)

MPC applications deal with MIMO systems. A MIMO system may have *m* inputs ***u*** and *n* outputs ***y***:

$$\mathbf{u}(k)=\begin{bmatrix} u_1(k) \\ u_2(k) \\ \vdots \\ u_m(k) \end{bmatrix} \qquad \mathbf{y}(k)=\begin{bmatrix} y_1(k) \\ y_2(k) \\ \vdots \\ y_n(k) \end{bmatrix} \tag{2.1}$$

Different structures are available in order to model the relations between $\mathbf{u}(k)$ and $\mathbf{y}(k)$ and will be discussed in detail in the chapters to come.

2.4.2. Control and Prediction Horizons

MPC is based on an iterative, finite horizon optimization of plant performance. As depicted in Figure 2.7, at time t_k the current plant output is sampled. This enables the controller to compute future input movements for a relatively short time horizon in the future in order to track the reference trajectory.

Only one input movement is implemented, with the plant output being sampled again, resulting in the repetition of the calculations at the new state, yielding a new control and new predicted output path. Two different limits are used; control horizon, **m**, and prediction horizon, **p**. At each step the controller calculates a series of m future control signals: $[\boldsymbol{u}(k), \boldsymbol{u}(k+1), \ . \ . \ . \ , \boldsymbol{u}(k+m-1)]$. The system behavior is then evaluated up to prediction horizon $p > m$, with $\mathbf{u}(k)$ kept constant for $m < k < p$.

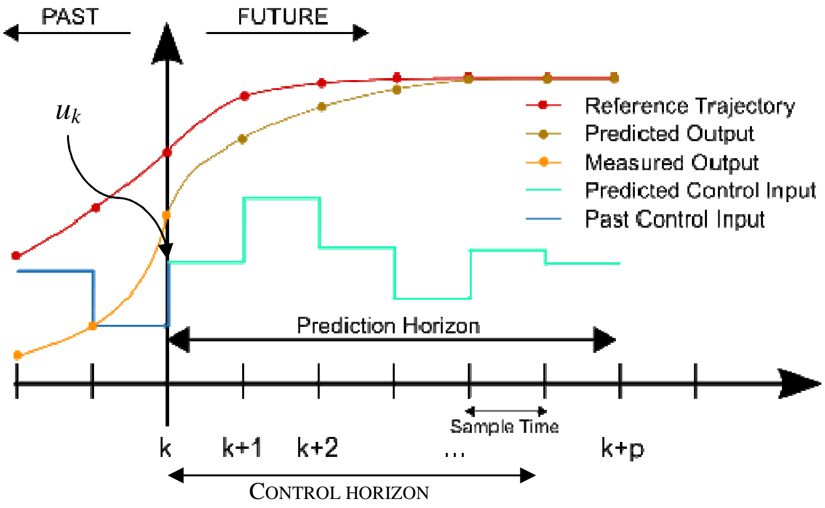


Figure 2.7: A conceptual picture of MPC

2.4.3. Optimization Criteria and Performance Indices

Optimization algorithms are generally applied to calculate a series of future input and output signals which minimize certain performance indices in the presence of constraints. There are various performance indices for MPC optimization criteria. As a general rule, the formulation of a performance-index or cost-criterion is based on the measurement of the reference tracking error (output form) and the control action (input form). The general 2-norm performance index is introduced below:

$$J(v,k)=\sum_{j=q}^{p}\hat{z}_1^T(k+j-1\big|k)\hat{z}_1(k+j-1\big|k)+\sum_{j=1}^{p}\hat{z}_2^T(k+j-1\big|k)\hat{z}_2(k+j-1\big|k) \tag{2.2}$$

N.B. $\hat{z}_i(k+j-1\big|k)$ is the prediction of $\hat{z}_i(k+j-1)$ at time k, for i = 1, 2.

Where $z_1(k)$ is a signal illustrating the reference tracking error and let $z_2(k)$ is a signal reflecting the control action. The prediction horizon is given by the variable **p** and the minimum cost-horizon is denoted by the variable **q**. Three performance indices often appear to be documented in literature and are found in industrial applications of MPC. These include Generalized Predictive Control (GPC) performance index; Linear Quadratic Predictive Control (LQPC) performance index and the Zone performance index. All three performance indices can be displayed in the standard form of (2.2) as shown below. The GPC or the LQPC performance indices are frequently dealt with in most papers on predictive control as they are clearly weighted squared 2-norms.

Table 2.2:
Summary of z_i terms for different performance indices (Stoorvogel, 2010)

z_i	GPC	LQPC	ZONE
z_1	$y_p(k+1)-r(k+1)$	$Q^{0.5}x(k+1)$	$r(k+1)-y(k+1)+\delta(k+1)$
z_2	$\lambda\Delta u(k)$	$R^{0.5}\Delta u(k)$	$\lambda u(k)$

r – Reference trajectory

λ – Weighting on the control signal

Q – Output weighting matrix

R – Control weighting matrix

δ – minimum error

2.4.4. Constraint Handling

In practice, a number of process variables in industrial processes are by and large subject to constraints. Certain signals must not contravene the desired bounds set as a result of the following factors:

- Safety limitations: all units are expected to function in a safe operating region; specific bounds are kept in place to prevent signals from entering an unsafe region.

- Environmental regulations: These limits are in place for water and air pollution control, recycling, waste disposal, and public health issues.

- Consumer specifications: products are required to be in a certain range and meet certain standards for consumer usage, for example, in the production of Biodiesel, the water and sediment content is required to be <0.05 % on a volume basis and a total glycerin content of <0.24 % by mass (Van Gerpen et al., 2004).

- Physical limitations: including temperature and pressure limits; limits on the level in reactor tanks; limits on flows through pipes.

These signals are prevented from reaching these bounds by appropriate setting of the controller parameters. However, control systems are designed to drive process variables as close as possible to their respective constraints without violating them due to fiscal motives. Maximum profit is attained in most cases when these process variables are closer to their respective limits. Hence model predictive control employs a more direct approach simply by implementing the optimal unconstrained solution in a manner in which those constraints are not violated. Optimization techniques such as linear programming (LP) or quadratic programming (QP) techniques are often applied in this instance.

$$
\begin{aligned}
&u_{\min} \le u(k) \le u_{\max}\,,\forall k \qquad \Delta u_{\min} \le \Delta u(k) \le \Delta u_{\max}\,,\forall k \\
&y_{\min} \le y(k) \le y_{\max}\,,\forall k \qquad \Delta y_{\min} \le \Delta y(k) \le \Delta y_{\max}\,,\ \forall k
\end{aligned} \tag{2.3}
$$

2.4.5. Process Model

The process model and the concept of open-loop optimal feedback are at the heart of MPC operation. The process model is used to generate a prediction of future subsystem behavior. The behavior of complex dynamic systems is determined by the applied models. The models compensate for the impact of non-linearities of variables. The prediction $y_m(k)$ in figure 2.5 is based on dynamical models. Common model representations in MPC are polynomial models, step response models, impulse response models or state space models. These models are often linear empirical models found by system identification.

CHAPTER 3 IDENTIFICATION PRINCIPLES

In this chapter the framework required for sound knowledge of developing model identification and validation techniques is described. Pertinent concepts relating to linear system representation are first given together with a brief description of linear response modeling. These concepts also include the representation of models as Laplace and Fourier transforms and the idea of discretization. A general survey of the different classes of models is given, with later sections delving into the structure of models as well the manner in which models may be formulated. The chapter closes with a general description of system identification.

3.1. LINEAR SYSTEM REPRESENTATION

A system or a transform maps input(s), *u(t)*, into output(s), *y(t)*:

$$y(t) = \Gamma\left[u(t)\right] \tag{3.1}$$

: Where Γ denotes the transform, a function from input signals to output signals.

Not all systems can be characterized as being linear, but many important ones in the chemical industry, especially in the hydrocarbon field can be. The possibility of utilizing the responses to a small set of inputs to predict the response to any probable input becomes large when a system qualifies as a linear system.

Figure 3.1 below depicts the manner in which a multivariable **system** and its surroundings are coupled through **signals** which are referred to as Process Variables (PVs).

Input variables, also known as manipulated variables (MVs), are declared as those variables that influence the system by exerting an action upon the system. The controlled input denoted with ***u*** can be manipulated. A measurement of this signal is assumed to be available at all times. Measured and unmeasured disturbance variables (DVs) contribute to a special class of input variables, commonly referred to as uncontrolled input variables. The uncontrolled input or disturbance, denoted with ***d***, cannot be manipulated. This signal is divided into a part ***w*** that is known and a part ***v*** that is not known. Output variables, also known as Controlled Variables (CVs), are those variables that originate in the system. The measured output, denoted ***y***, is available for control.

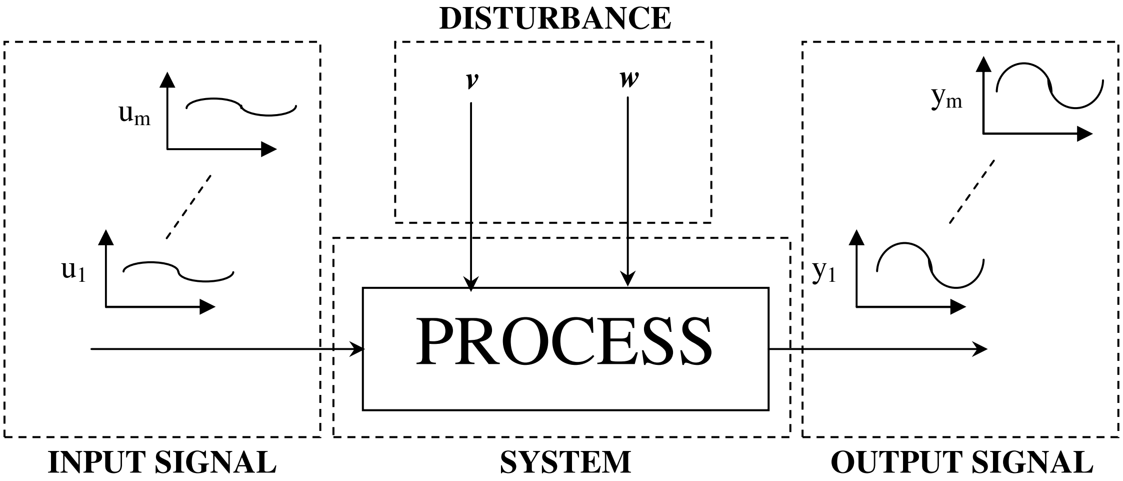


Figure 3.1: Linear multivariable system representation

3.1.1. Signals

A very well understood and common approach to identification is to introduce known input disturbances to the system and record the system's response. From knowledge of the input and the type of response it yields, system dynamics may be extracted and represented in an appropriate form. Typical inputs for such approaches are depicted below:

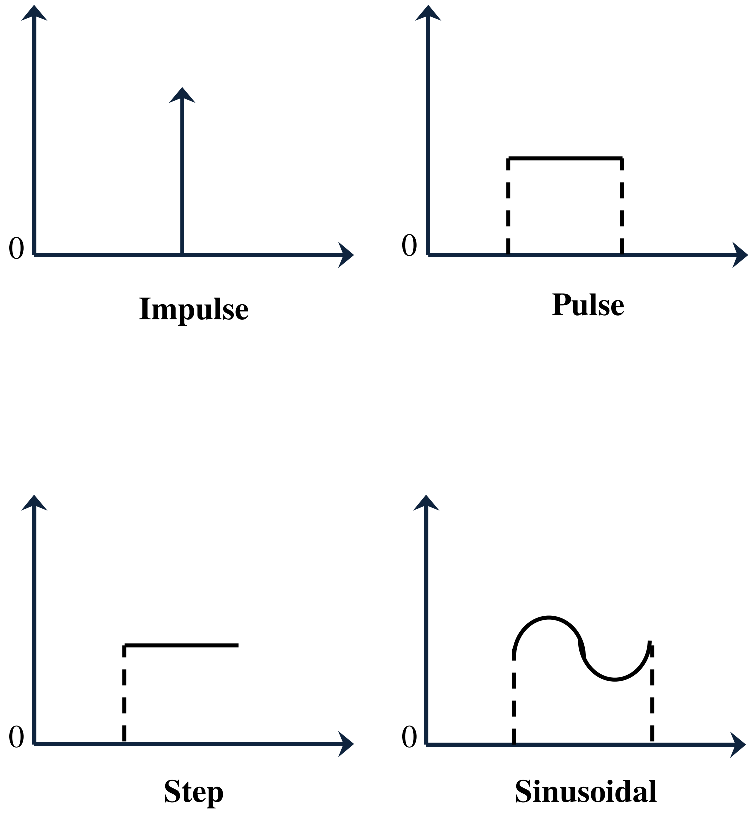


Figure 3.2: Typical input responses for linear system modeling

Impulse and Pulse Signal: mathematically, an impulse response is given as the following representation:

$$\begin{aligned} &\delta^*(t) = 0 \text{ for } t \neq 0 \\ &\int_{-\infty}^{\infty} \delta^*(t)dt = 1 \end{aligned} \qquad (3.2)$$

One very useful way of thinking of an impulse response is as a limiting case of a pulse signal:

$$\delta^*{}_{\Delta}(t) = \begin{cases} {}^{1}\!/_{\Delta} & \mathit{if} \quad 0 < t < \Delta \\ 0 & \text{otherwise} \end{cases} \qquad (3.3)$$

N.B. the time, τ represents the time at which the input occurs and is found by locating $\delta^*(t-\tau)$ and $\delta_\Delta^*(t-\tau)$

The impulse signal is equal to a pulse signal when the pulse gets infinitely short:

$$\delta^*(t) = \lim_{\Delta\to 0}\delta^*{}_\Delta(t) \tag{3.4}$$

Step signal: The unit step signal is zero for all times less than zero, and 1 for all times greater than or equal to zero:

$$u(t) = \begin{cases} 0 & t < 0 \\ 1 & t \geq 0 \end{cases} \tag{3.5}$$

Sinusoidal signal: although the bulk of this project is based on discrete-time approaches, it is useful to note the sine-wave response for system representation:

$$u\ \ (t) = \begin{cases} 0 & t < 0 \\ \sin(wt) & t \geq 0 \end{cases} \tag{3.6}$$

3.1.2. Linear System Description

The concerned systems to be dealt with are known to be time-invariant, linear and causal (Corriou, 2004). The system is said to be **time-invariant** if its "response to a certain input signal does not depend on absolute time", in other words the parameters within the model that depict the system do not change with time. It is said to be **linear** if its "output response to a linear combination of inputs is the same linear combination of the output responses of individual inputs". Furthermore, it is said to be **causal** if the "output at a certain time depends on the input up to that time only".

It should be noted that although the pertinent area of attention is focused on multivariable systems, single-input single-output systems are discussed initially in order to focus on important concepts hence forth. These concepts will be shown to extend to MIMO cases.

System without Disturbance: The linear system of input, *u(t)* and output, *y(t)*, which is time-invariant and causal, can be described by its impulse response *g(k)* such that:

$$y(t) = \sum_{k=1}^{\infty} g(k)u(t-k) \quad for\; t = 0,1,2,... \text{ where } t \text{ is used as a discrete time index} \tag{3.7}$$

The impulse response is often used as it is a complete characterization of the system. This can be seen by the stair-case approximation of a continuous-time signal. Any signal can be expressed as a sum of scaled and shifted impulses.

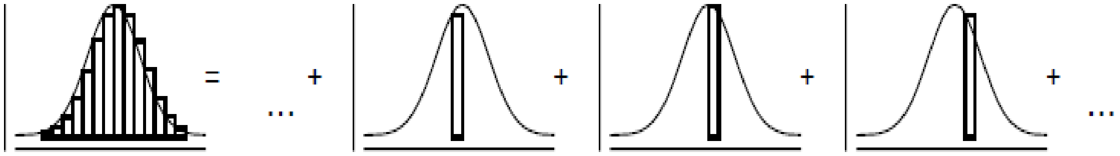


Figure 3.3: Staircase approximation of a continuous-time signal (Heeger, 2000)

In order to simplify the notation the sampling instants are denoted by 0, 1, 2 … as if they were separated by a unit sampling period T_s. It should be noted that *y(t)* depends on *u(t-1)*, *u(t-2)*, …, but not on *u(t)*, because it is estimated that output is not immediately influenced by the input, even if the system presents no time delay.

System with Disturbance: A system is subjected to disturbances, so that the output cannot be calculated in relation to the input alone. The disturbances may come from the measurement noise, also known as uncontrolled inputs. They are simply represented by adding a term to the output as shown below:

$$y(t) = \sum_{k=1}^{\infty} g(k)u(t-k) + \upsilon(t) \tag{3.8}$$

The disturbance model is written as:

$$\upsilon(t) = \sum_{k=0}^{\infty} h(k)\varepsilon(t-k) \tag{3.9}$$

At this point it is convenient to introduce the basic notation that will be used to represent essential components that describe a system from here on. Introduce the delay operator, q^{-1}, such that:

$$q^{-1}y(t)=y(t-1) \tag{3.10}$$

Recall equation (3.6) and introduce the delay operator to simplify the equation:

$$\begin{aligned} y(t) &= \sum_{k=1}^{\infty} g(k)u(t-k) = \sum_{k=1}^{\infty} g(k)q^{-k}u(t) \\ &= \left[\sum_{k=1}^{\infty} q^{-k}g(k)\right]u(t) \\ &= G(q)u(t) \end{aligned} \tag{3.11}$$

Thus the transfer function of the linear system is given by:

$$G(q)=\sum_{k=1}^{\infty} q^{-k}g(k) \tag{3.12}$$

Similarly the disturbance model can be re-written as:

$$\begin{aligned} \upsilon(t) &= \sum_{k=1}^{\infty} h(k)\varepsilon(t-k) = \sum_{k=1}^{\infty} h(k)q^{-k}\varepsilon(t) \\ &= \left[\sum_{k=1}^{\infty} q^{-k}h(k)\right]\varepsilon(t) \\ &= H(q)\varepsilon(t) \end{aligned} \tag{3.13}$$

Finally equation (3.7) becomes:

$$y(t)=G(q)u(t)+H(q)\varepsilon(t) \tag{3.14}$$

This equation is a general representation of a system. *H(q)* is monic, which means that it's a polynomial with its leading coefficient (the coefficient of the highest order) being 1, (*h(0)*=1). In essence, disturbance cannot be predicted and is described by *ε(t)* (a sequence of independent random variables), which is often chosen as white noise with a zero mean and variance λ^2.

3.1.2.1. Transforms and Transfer Functions

Fourier and Laplace transforms form the basis of classical control design. The Laplace transform of a function *f(t)*, defined for all real numbers t>0, is the function *F(s)*, defined by:

$$F(s) = L[f(t)] = \int_0^\infty e^{-st} f(t)dt \tag{3.15}$$

The Laplace transform is a fundamental transform possibly second only to the Fourier transform in its effectiveness in solving physical problems. It is predominantly helpful in solving linear ordinary differential equations.

There are several common conventions for defining the Fourier transform; however for this thesis the following definition will be used:

$$F(jw) = \Im[f(t)] = \int_0^\infty e^{-jwt} f(t)dt \tag{3.16}$$

Applying the Laplace and Fourier Transforms to the underlying linear time-domain differential equations leading to the discrete equation (3.14) yields similar functional forms:

$$Y(s) = G(s)U(s) + H(s)\mathrm{E}(s) \tag{3.17}$$

$$Y(jw) = G(jw)U(jw) + H(jw)\mathrm{E}(jw) \tag{3.18}$$

The advantage of representing the system equation in the form of a transform is that they simplify the input-output convolution relation illustrated in equation (3.7) and (3.8) to a multiplication of rational transforms. The appropriate transform inversion provides the time-domain solution for equations (3.16) and (3.17) respectively. This makes solving system equations much easier. For this reason, transfer functions are very practical in terms of quantitative descriptions of a system.

3.2. MODEL CLASSIFICATION, FORMULATION AND STRUCTURE

It is necessary to choose a type of model that would adequately represent the system before proceeding to system identification. After the choice of the model class, form, and structure is made, it will then be possible to estimate the parameters of this model.

3.2.1. Model Classification

Model classification is largely dependent on the extent of prior knowledge in addition to physical insight of a system. Several classes of models exist in literature which is readily available. The choice of model classification is governed primarily by the following requirements:

- **Required level of flexibility**: A model designed to assist in a long term design project is required to be flexible so as to sustain any unexpected design changes.

- **Available Resources**: On occasion, the type of model implemented is restricted by the available computing power. In such cases, the model must be simplified or broken down to ease the computational strain.

- **The number of approximations***:* The efficiency of a model can be significantly increased by making numerous apposite approximations. This is satisfactory with the provision that the applied approximations do not appreciably reduce model accuracy.

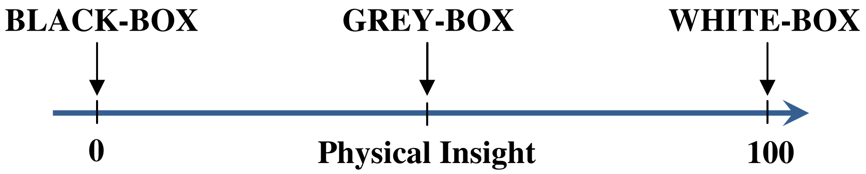


Figure 3.4: Classification of models based on physical insight

The literature customarily distinguishes between three model classes (figure 3.4). A black box model is employed in cases where the underlying mechanisms of a system are not considered. They are generally empirical in nature i.e. they do not have any physical meaning. Grey box models represent the majority of simulation models. In these models, some physical representation is described, but some of the physics is approximated. A white box model characterizes the real process as closely as possible and is regarded as the most detailed type of model.

Black-box models: generally consists of a body of rules and equations that are capable of running very efficiently and optimized easily. The direct implication is that black-box models require minimal computing power. However, a major shortcoming of a black box model is that it is inflexible in addition to its absence of any form of physical meaning.

Grey-box models: In those cases that require flexibility, a more general model is required which can be adapted in the case of model inconsistencies in a design. Grey box models are regarded as more flexible than the latter mentioned class. It also facilitates the use of modeling to optimize a design. A significant disadvantage lies in the fact that several approximations that are made when grey-box models are used may affect the model accuracy adversely.

White-box models: are analogous to real processes or systems. The models representation of a real process unit is extremely close to its actual behavior. As a result of white box models containing no or few approximations, they are the most intricate types of model to implement. The complexity of a white box model means it requires a large amount of computational effort and vast amounts of memory.

Although it can be seen that grey-box and white-box models seem to be the best choice for model usage, industry still employs the black-box as its primary class. Industry believes in the old adage of keeping it simple and black-box modeling has been noted to provide sufficient performance as linear models.

3.2.2. Model Formulation

Models may be represented in several ways. The two most frequently used representations are the state space model (illustrated by figure 3.5) and a transfer function model, also known as an input/output model (see example in figure 3.6 below). Although the input/output model representation was mentioned in the previous section, the mathematical illustration of these 2 forms of model representation is covered in greater detail in section 3.2.3. The two forms stated above are mathematically equivalent, meaning that one can transfer a state space model into a transfer function model and vice versa. However, each presentation has its own distinctive characteristics. For instance, there can be specific parameters to characterize the time delay, gain and time constant in a transfer function model, whereas there are no such parameters in a state space model. It is also easy to define a sensitivity function and a complementary sensitivity function using transfer function models whereas it is not straightforward to do the same using state space model.

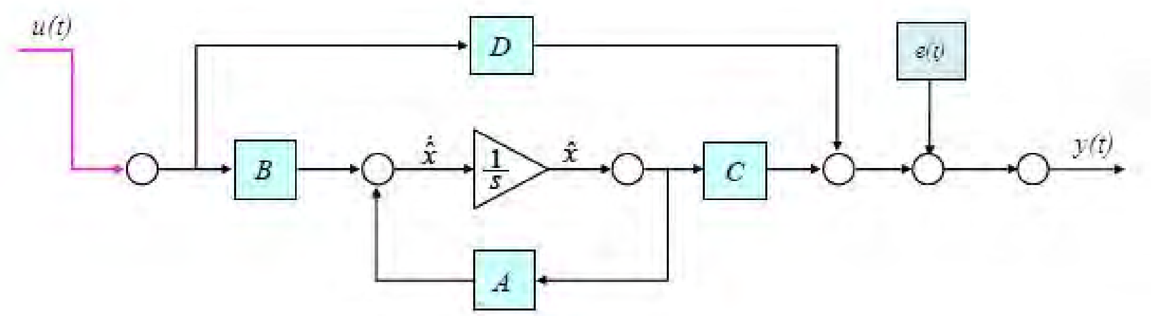


Figure 3.5: State-Space Representation of a model

3.2.3. Model Structure

This section will focus primarily on multivariable model structures. Depending on how one parameterises the model in equation (3.19), different parameter estimation methods or model structures studied in literature can be derived. Model structures can be derived from first principles and may also exhibit non-linearity as depicted in figure 3.6 below, but most models used in industrial Model Predictive Control are empirical in nature and are often linear. The left-hand side of the diagram represents empirical models derived exclusively from operating data.

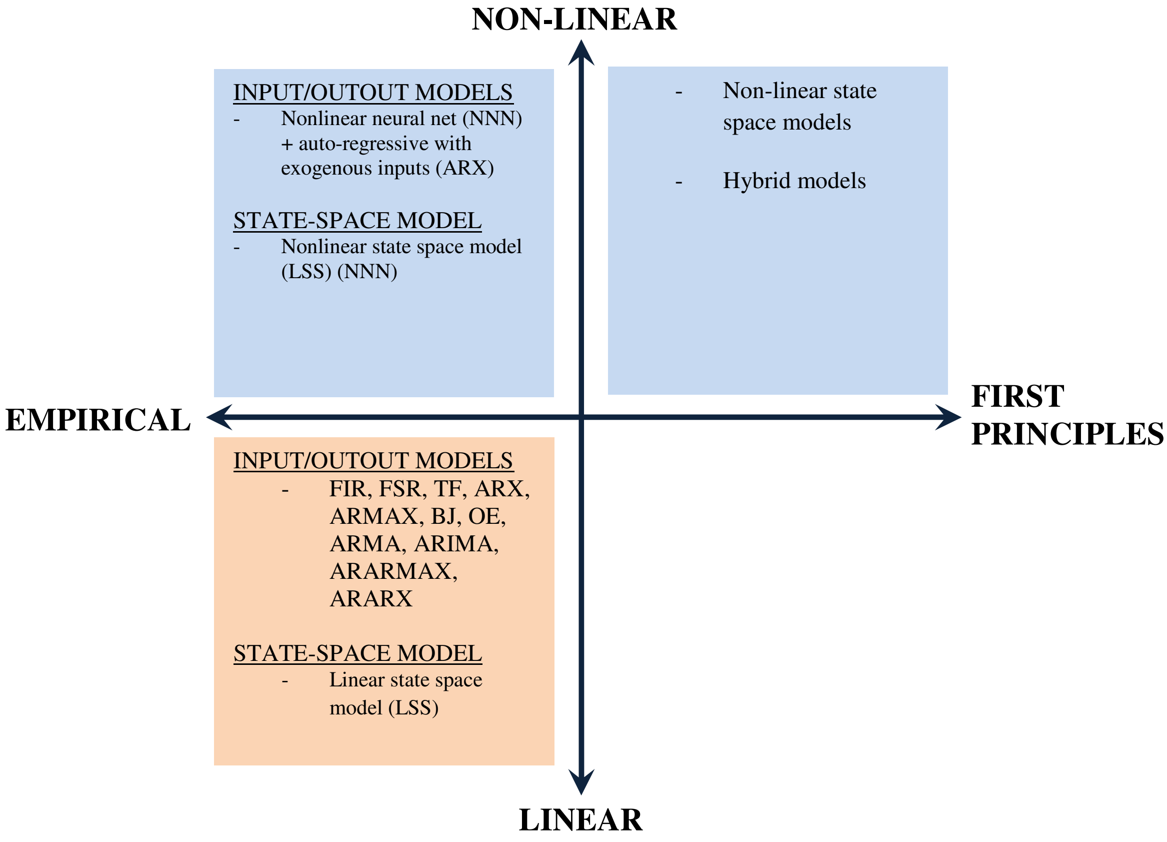

NON-LINEAR
INPUT/OUTOUT MODELS
- Nonlinear neural net (NNN) + auto-regressive with exogenous inputs (ARX)
STATE-SPACE MODEL
- Nonlinear state space model (LSS) (NNN)
- Non-linear state space models
- Hybrid models
EMPIRICAL
FIRST PRINCIPLES
INPUT/OUTOUT MODELS
- FIR, FSR, TF, ARX, ARMAX, BJ, OE, ARMA, ARIMA, ARARMAX, ARARX
STATE-SPACE MODEL
- Linear state space model (LSS)
LINEAR

Figure 3.6: Schematic of model formulations and structures

3.2.3.1. Input/Output Model Structures

Input/output models are classified as parametric and non-parametric model. Considering parametric models first, the most general parametric model structure of input/output models is given by the ARMAX model, with the acronym translating to: **A**uto-**R**egressive **M**oving **A**verage e**X**ogenous. This model structure is given by:

$$\begin{aligned} y(t) &= G(q)u(t) + H(q)\varepsilon(t) \\ &= \frac{B(q)}{A(q)}u(t) + \frac{C(q)}{A(q)}\varepsilon(t) \end{aligned} \qquad (3.19)$$

This model can be rearranged in a linear form:

$$A(q)y(t) = B(q)u(t) + C(q)\varepsilon(t) \tag{3.20}$$

Polynomials, *A(q)*, *B(q)* and *C(q)* are defined as:

$$A(q) = 1 + \sum_{k=1}^{n_a} a_k q^{-k} \tag{3.21}$$

$$B(q) = \sum_{k=1}^{n_b} b_k q^{-k} \tag{3.22}$$

$$C(q) = 1 + \sum_{k=1}^{n_c} c_k q^{-k} \tag{3.23}$$

The model is called ARMAX, as the part *A(q)y(t)* is the regressive part in the expression *y(t)*, *C(q)e(t)* is the moving average term and *B(q)u(t)* is the exogenous part (external input). This model can be translated into different forms of parametric models depending on the particular values of n_a, n_b and n_c:

AR (autoregressive) model if $n_b = n_c = 0$. The output is expressed as a pure time series without any input signal:

$$A(q)y(t) = \varepsilon(t) \tag{3.24}$$

MA (moving average) model if $n_a = n_b = 0$. The output does not depend on the input and is equal to:

$$y(t) = C(q)\varepsilon(t) \tag{3.25}$$

ARMA (autoregressive moving average) model if $n_b = 0$. The output simply describes the influence of a disturbance in a general manner and is expressed as the relation:

$$A(q)y(t) = C(q)\varepsilon(t) \tag{3.26}$$

ARIMA (autoregressive integrated moving average) model if $n_b = 0$ and if one forces the polynomial matrix *A(q)* to contain as a factor a differentiator term $(1 - q^{-1})$ which is **useful in suppressing offset in control.** The output is expressed according to the relation:

$$A(q)y(t) = C(q)\varepsilon(t) \tag{3.27}$$

ARX (autoregressive exogenous) model if $n_c = 0$. This model is the simplest and most applied parametric input/output model in industrial processes. In this structure, the plant model and the disturbance possess the same dynamics as they are both specified by the denominator *A(q)*. The output is expressed as follows:

$$A(q)y(t) = B(q)u(t) + \varepsilon(t) \tag{3.28}$$

FIR (finite impulse response) model if $n_a = n_c = 0$. This model is a special form of an ARX model. This model is referred to as a non-parametric model. The difference between the other models defined above and FIR is that parametric models are much more compact and necessitate fewer parameters to illustrate the similar dynamic behavior. For FIR, the output is simply equal to:

$$y(t) = B(q)u(t) + \varepsilon(t) \tag{3.29}$$

All the models described, include the transfer functions *G* and *H*, having the same polynomial *A(q)* in their transfer functions. This may appear as a limitation. For this reason, other types of model structures have been developed as modifications to the ARMAX form and completely independent model forms. These variations include:

ARARX model: this is obtained from an ARX model by replacing the error term taken as a moving average by an autoregressive error term. The ARARX model is thus written as:

$$A(q)y(t) = B(q)u(t) + \frac{1}{D(q)}\varepsilon(t) \tag{3.30}$$

$$D(q) = 1 + \sum_{k=1}^{n_d} d_k q^{-k}$$

ARARMAX model: this is obtained by using an autoregressive moving average type (ARMA) for the equation error:

$$A(q)y(t) = B(q)u(t) + \frac{C(q)}{D(q)}\varepsilon(t) \tag{3.31}$$

OE (output error) model: the simplest output error model that can be developed is in the following form:

$$y(t) = \frac{B(q)}{F(q)}u(t) + \varepsilon(t) \tag{3.32}$$

In this case the error, *ε(t)*, bears only on the output *y(t)*, hence the name of this model. In this model the transfer function is denoted by *B(q)/F(q)* instead of *B(q)/A(q)* as in the ARX model to distinguish the different roles played by *A(q)* and *F(q)* in each model structure.

BJ (Box-Jenkins) model: the previous model can be improved by introducing a transfer function for white noise ***ε(t)***:

$$y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}\varepsilon(t) \tag{3.33}$$

3.2.3.2. General Model for Identification

The most general model which is illustrated by figure 3.7 can be written as:

$$A(q)y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}\varepsilon(t) \qquad (3.34)$$

The advantage of having various forms of parametric models is that there is more freedom in describing the properties of the disturbance term. Table 3.1 provides a summary of the common model structures as well as a comparison of these structures in terms of the following important points:

- The compactness of the model.
- The numerical complexity in parameter estimation
- The consistency of the model in closed loop.

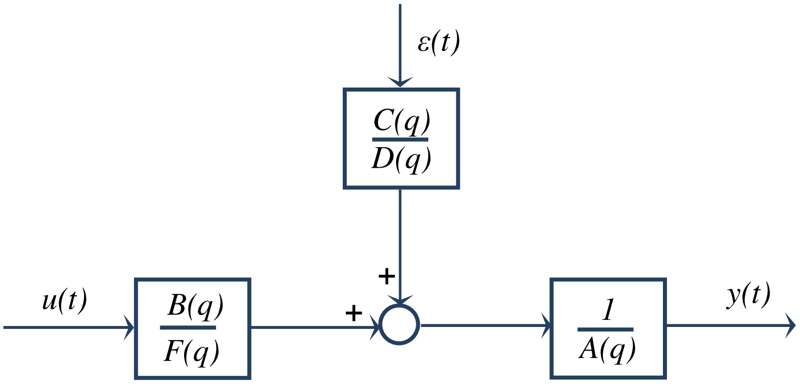


Figure 3.7: General Structure Transfer Function Representation

Table 3.1
Comparison of various model structures adapted from Ljung (1987)

Model Structure	Polynomials used in (3.37)	Numerical Difficulty	Compactness	Consistency in closed-loop?
FIR	B	Low	Low	No
ARX	AB	Low	Medium	Yes
OE	BF	High	Highest	No
ARMAX	ABC	High	High	Yes
Box-Jenkins	BFCD	High	Highest	Yes

3.2.3.3. Multivariable Extensions

The extension of most model structures to the multivariate condition is mostly a matter of notation changes. Equation (3.14) can be extended to a multivariable case by defining the following transfer function matrices: $\boldsymbol{G^o}$ and $\boldsymbol{H^o}$:

$$\boldsymbol{y}(t) = \boldsymbol{G}^o(q)\boldsymbol{u}(t) + \boldsymbol{H}^o(q)\boldsymbol{\varepsilon}(t) \tag{3.35}$$

$\boldsymbol{G^o}$ and $\boldsymbol{H^o}$ is defined by the following matrices:

$$\boldsymbol{G}^o(q) = \begin{bmatrix} G_{11}(q) & G_{12}(q) & \cdots & G_{1m-1}(q) & G_{1m}(q) \\ G_{21}(q) & G_{22}(q) & \cdots & G_{2m-1}(q) & G_{2m}(q) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ G_{n-1,1}(q) & G_{n-1,2}(q) & \cdots & G_{n-1,m-1}(q) & G_{n-1,m}(q) \\ G_{n1}(q) & G_{n2}(q) & \cdots & G_{n,m-1}(q) & G_{nm}(q) \end{bmatrix} \tag{3.36}$$

$$\boldsymbol{H}^o(q) = \begin{bmatrix} H_{11}(q) & H_{12}(q) & \cdots & H_{1n-1}(q) & H_{1n}(q) \\ H_{21}(q) & H_{22}(q) & \cdots & H_{2n-1}(q) & H_{2n}(q) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ H_{n-1,1}(q) & H_{n-1,2}(q) & \cdots & H_{n-1,n-1}(q) & H_{n-1,n}(q) \\ H_{n1}(q) & H_{n2}(q) & \cdots & H_{n,n-1}(q) & H_{nn}(q) \end{bmatrix} \tag{3.37}$$

Where $\boldsymbol{u(t)}$ is an m-dimensional input vector (MVs), $\boldsymbol{y(t)}$ is an n-dimensional output vector (CVs) and $\varepsilon(t)$ is also an n-dimensional vector. $\boldsymbol{G^o(q)}$ is defined as a n x m model matrix and $\boldsymbol{H^o(q)}$ is defined as a n x n matrix.

Polynomials, *A(q), B(q), C(q), D(q)* and *F(q)* are extended as polynomial matrices defined as follows:

$$\boldsymbol{A}(q) = I + \sum_{k=1}^{n_a} \boldsymbol{A_k} q^{-k} \tag{3.38}$$

$$\boldsymbol{B}(q) = \sum_{k=1}^{n_b} \boldsymbol{B_k} q^{-k} \tag{3.39}$$

$$\boldsymbol{C}(q) = 1 + \sum_{k=1}^{n_c} \boldsymbol{C_k} q^{-k} \tag{3.40}$$

$$\boldsymbol{D}(q) = \sum_{k=1}^{n_b} \boldsymbol{D_k} q^{-k} \tag{3.41}$$

$$\boldsymbol{F}(q) = 1 + \sum_{k=1}^{n_c} \boldsymbol{F_k} q^{-k} \tag{3.42}$$

3.2.3.4. Multivariable State-Space Model Structures

State-space models are useful in handling stable, unstable and integrating processes (Zhu *et al.*, 1997). Zhu showed that the state-space model of a linear process with a disturbance can be given as:

$$\begin{aligned} \boldsymbol{x}(t+1) &= \boldsymbol{Ax}(t) + \boldsymbol{Bu}(t) + \boldsymbol{K}^* \boldsymbol{\varepsilon}(t) \\ \boldsymbol{y}(t) &= \boldsymbol{Cx}(t) + \boldsymbol{Du}(t) + \boldsymbol{\varepsilon}(t) \end{aligned} \tag{3.43}$$

Where ***x(t)*** is the state vector, the constant matrices ***A***, ***B***, ***C*** and ***D*** form the state-space description of the process and the constant matrix $\boldsymbol{K}^*$ is the Kalman gain that characterizes the state noise ***ϵ(t)***. The state-space model is equivocal to an input/output model as in equation (3.35) by defining the following transfer functions:

$$\begin{aligned} \boldsymbol{G^o}(q) &= \boldsymbol{C}(q\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B} \\ \boldsymbol{H^o}(q) &= \boldsymbol{C}(q\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{K}^* \end{aligned}$$

3.3. SYSTEM IDENTIFICATION

System identification deals with the problem of building mathematical models of dynamic systems from routine operating data. For a MPC application, the most time consuming process is the identification of models. The different sectors of concern for the implementation of a Model Predictive Control system are shown below together with the percentage of time spent on each sector (Zhu et al., 1997):

- Functional design and benefit study: 10%.

- Pre-test: 10%
- Model Identification: **<u>40%</u>**.
- Controller simulation and tuning: 15%
- Controller commissioning: 25%

It is shown that approximately 40% of the total time is spent of the identification of models due to the fact that MIMO systems possess a large number of models. This is no doubt the most important part of MPC implementation.

3.3.1 System Identification Procedure

Ljung (1989) suggests that three basic entities are involved in the construction of a model as depicted in the system identification loop below (Figure 3.8):

- The data record
- a set of candidate models
- A rule by which candidate models can be assessed using the data

<u>The data record</u>: The input-output data are occasionally recorded during an exclusively planned identification experiment. The user may establish which signals to measure and when to measure them and may also choose the input signals. The objective of the experimental design is consequently to make these choices so that the data becomes more informative, subject to the corresponding constraints that may be at hand.

<u>The set of models</u>: A set of candidate models is obtained by specifying within which collection of models one should look for an apt model. It is at this stage that prior knowledge, engineering perception and insight are combined with recognized properties of models. On occasion the model may be obtained from careful modeling.

<u>Determining 'best' model guided by the data</u>: This stage requires the use of an identification method. The model quality is characteristically assessed on the basis of how the models perform when they attempt to reproduce separately measured data.

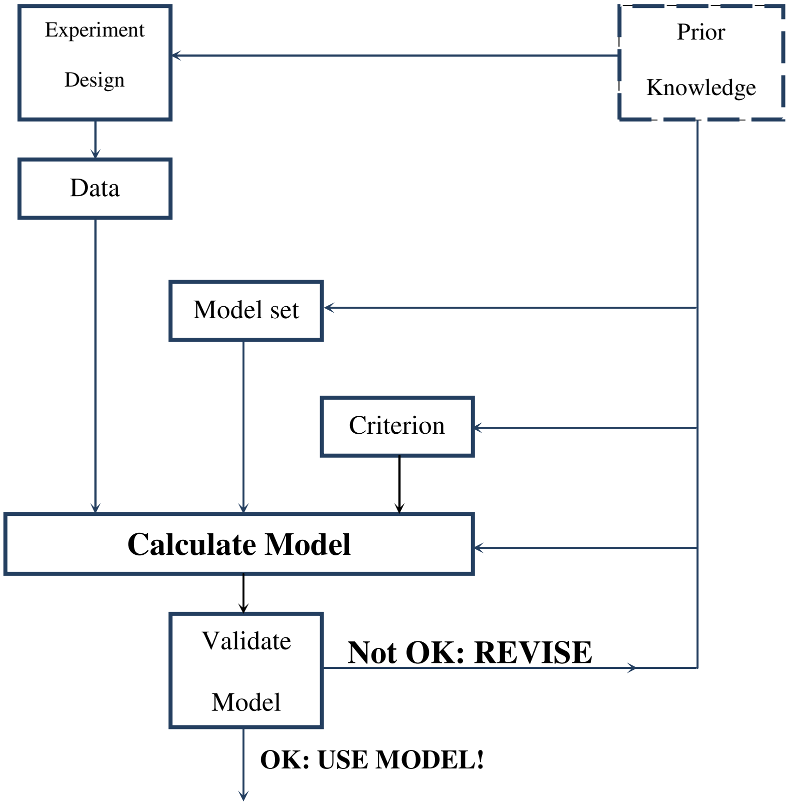


Figure 3.8: The System Identification Procedure adapted from Ljung (1989)

Table 3.2 illustrates the available MPC technology employed by industrial MPC vendors. This table indicates the structure of models used as well as the estimation methods employed.

Table 3.2:
Comparison of linear MPC identification technology (Qin, 2003)

Product	Model Form	Est. Method
DMC-plus	FIR, LSS	MLS
RMPCT	FIR, ARX, BJ	LS, GN, PEM
AIDA	LSS, FIR, TF, MM	PEM-LS, GN
Glide	TF	GD, GN, GM
Connoisseur	FIR, ARX, MM	RLS, PEM

3.3.2. System Identification Methods

The coefficients to be determined, in general, are not known through the knowledge of the physical model but by black-box representation, thus these coefficients enter the model as parameters to be determined by estimation techniques. The parameter vector is denoted by *θ*. The structure of the parameter vector depends on the type of model structure chosen. The system model given by equation (3.14) could be written in these conditions as:

$$y(t) = G(q,\theta)u(t) + H(q,\theta)\varepsilon(t) \tag{3.44}$$

It is advisable to compare this model to the prediction model where the output depends solely on both past inputs and outputs and is given by:

$$\hat{y}\;(t|t-1) = H^{-1}(q,\theta)G(q,\theta)\;\; u(t) + \left[1 - H^{-1}(q,\theta)\right]\;\; y(t) \tag{3.45}$$

In order to simplify the notation of the above equation, the following expressions are established:

$$W_u(q,\theta) = H^{-1}(q,\theta)G(q,\theta)$$
$$W_y(q,\theta) = \left[1 - H^{-1}(q,\theta)\right]$$

$$\hat{y}(t|t-1) = W_u(q,\theta)u(t) + W_y(q,\theta)y(t) \tag{3.46}$$

Model sets: the search for a suitable model is typically conducted over a set of candidate models. Quite naturally, a model set is defined as:

$$M^* = \left\{ W_\varsigma(q) \middle| \varsigma \in \omega \right\} \tag{3.47}$$

This is a collection of models, where *ζ* is an index related to the *u* and *y* models, and *ω* is the index set.

Once a model is chosen, a specific estimation method for the parameter vector, θ, is used to calculate the model.

Data sets: for the calculation of a certain model, experimental data is collected. Data collected for any identification test is denoted by the following sequence:

$$Z^N = [u(1), y(1), u(2), y(2), ..., u(N), y(N)] \tag{3.48}$$

Where ***u(t)*** is an *m*-dimensional input vector (MVs), ***y(t)*** is an *n*-dimensional output vector (CVs) and *N* is the number of samples.

3.3.2.1. Prediction Error Framework: Linear Regression

Consider a single input-output relationship, represented by an ARX model structure, described as a linear difference equation. N.B: The extension to multivariable regression simply involves a change in notation as shown in section 3.3.2.2.

$$\begin{aligned} &A(q)y(t) = B(q)u(t) + \varepsilon(t) \\ &y(t) + a_1 y(t-1) + ... + a_{n_a} y(t-n_a) = b_1 u(t-1) + ... + b_{n_b} u(t-n_b) + \varepsilon(t) \end{aligned} \tag{3.49}$$

The parameter vector is defined as:

$$\theta = \left[a_1, ..., a_{n_a}, b_1, ..., b_{n_b}\right] \tag{3.50}$$

With *G(q,θ) = B(q)/A(q)* and *H(q,θ) = 1/A(q),* the ARX equation can be written in predictor form according to equation (3.45):

$$\hat{y}\ (t|\theta) = [1 - A(q)]\ \ y(t) + B(q)\ \ u(t) \tag{3.51}$$

The observation vector is:

$$\phi(t) = [-y(t-1), ..., -y(t-n_a), u(t-1), ..., u(t-n_b)]^T \tag{3.52}$$

With the definition of the parameter vector coupled with the definition of the observation vector, the predictor of the output can be written in vector form:

$$\hat{y}(t|\theta)=\phi^T(t)\theta=\theta^T\phi(t) \tag{3.53}$$

With the predictor being a linear function with respect to the parameters, the problem is a linear regression problem and the parameters can be searched by least-squares procedures:

Consider that the observed data has been generated by:

$$y(t)=\phi^T(t)\theta_o+v_o(t) \tag{3.54}$$

Where θ_o is depicted as the 'true value' of the parameter vector. The idea of least-squares regression is to minimize the prediction error defined as follows:

$$\varepsilon(t)=y(t)-\phi^T(t)\theta \tag{3.55}$$

Various criterion functions (for minimization) may be defined; the simplest case is given below:

$$V_N(\theta)=\frac{1}{N}\sum_{t=1}^{N}\frac{1}{2}\left[y(t)-\phi^T(t)\theta\right]^2 \tag{3.56}$$

The exclusive characteristic of this criterion is that it is a quadratic function in θ. For that very reason it can be minimized systematically if the indicated inverse exists which gives:

$$\hat{\theta}_N^{LS}=\text{argmin}V_N(\theta)=\left[\frac{1}{N}\sum_{t=1}^{N}\phi(t)\phi^T(t)\right]^{-1}\frac{1}{N}\sum_{t=1}^{N}\phi(t)y(t) \tag{3.57}$$

The desired properties of θ_N would be that:

1. It is close to θ_o.
2. It converges to θ_o as N tends to infinity.

It is noted that for other model structures other than ARX and FIR, the regression is not strictly linear. For example, the observation vector for an ARMAX model structure becomes:

$$\phi(t,\theta)=\left[-y(t-1),...,-y(t-n_a),u(t-1),...,u(t-n_b),\varepsilon(t-1,\theta),...,\varepsilon(t-n_c)\right]^T \tag{3.58}$$

The resulting predictor becomes:

$$\hat{y}(t|\theta)=\phi^T(t,\theta)\theta \tag{3.59}$$

This is a pseudo-linear regression, due to the non-linear effect of θ on $\phi(t,\theta)$.

3.3.2.2. Multivariable Extensions

Representing a multivariable system as a parametric model structure is done in a very similar way to its scalar counterparts. Equation (3.49) is written as follows for a MIMO system:

$$\boldsymbol{y}(t)+\boldsymbol{A_1}\boldsymbol{y}(t-1)+...+\boldsymbol{A_{n_a}}\boldsymbol{y}(t-n_a)=\boldsymbol{B_1}\boldsymbol{u}(t-1)+...+\boldsymbol{B_{n_b}}\boldsymbol{u}(t-n_b)+\boldsymbol{\varepsilon}(t) \tag{3.60}$$

Here $\boldsymbol{A}(q)$ and $\boldsymbol{B}(q)$ are defined by equations (3.38) and (3.39) respectively, where $\boldsymbol{A_k}$ are n x n matrices and $\boldsymbol{B_k}$ are n x m matrices. The parameter vector becomes somewhat complex and large for a system composed of n outputs and m inputs. This is because the parameter vector is now extended and defined as a $[n_a \cdot n + n_b \cdot m]$ x n matrix:

$$\theta=\left[\boldsymbol{A_1},...,\boldsymbol{A_{n_a}},\boldsymbol{B_1},...,\boldsymbol{B_{n_b}}\right] \tag{3.61}$$

The number of parameters that need to be evaluated is found as [$n_a \cdot n^2 + n_b \cdot m \cdot n$]. Table 3.3 provides an indication as to the number of parameters that need to be found for a number of different combinations of MVs and CVs that are found in industrial processes, if n_a and n_b are kept at a constant order of 3. In most processes, this order is often higher.

Table 3.3:
Summary of the number of parameters found by system identification

n_a	n_b	n (CVs)	m (MVs)	N_p
3	3	5	3	120
3	3	10	5	450
3	3	20	10	1800
3	3	40	20	7200

For a relatively small system of 5 outputs and 3 inputs, the number of parameters, N_p, is 120. However for a large system of 40 outputs and 20 inputs it is 7200. This makes the time for system identification of MIMO systems exceedingly long, and of course the data would have to be sufficiently variable to elucidate all parameters.

3.3.2.3. Other System Identification methods

The ASYM method: it was primarily developed for the application in large-scale industrial processes. Initially, the method identifies a higher order ARX model. A model order reduction is then performed to obtain a reduced model that still represents the plant adequately. The final model is then represented in a BJ format. Besides performing parameter estimation, the method dictates solutions to optimal test design, order selection and model validation.

The Subspace Method: Larimore (1990) proposed the subspace method in 1990, which estimates a state-space model from routine input/output data.

3.3.3. Model Validation

The model validation techniques considered for the system identification process are normally carried out in open-loop. After an MPC controller is implemented, constant monitoring of the quality of the process model under closed-loop conditions is imperative for maintaining optimal controller performance.

Chapter 4 seeks to establish the fundamental theory of closed-loop model validation as well as some of the previous work done. This provides the link required for the development of the present work.

CHAPTER 4 MODEL VALIDATION THEORY

The need to constantly monitor MPC controllers by model quality indications is imperative in maintaining the required performance level of such controllers. This chapter presents the relevant theoretical background of closed-loop model validation. The methods involved are capable of running parallel to the plant in an online manner or making use of historical process data. This chapter proceeds with the concept of closed loop vs. open loop. This leads to the need for closed-loop model validation by highlighting some of the issues that hinder open-loop identification. Thereafter, an overview of the different classes of model validation is presented coupled with the related work in each class. The chapter closes with a section on residual analysis, as this forms the mainstay of the most applicable model validation techniques in closed-loop.

4.1. CLOSED-LOOP MODEL VALIDATION

How to continuously monitor the quality of process models under closed-loop conditions is a challenging problem. Closed-loop model validation means that the process models are analyzed for any MPM whilst the underlying process is fully under feedback control.

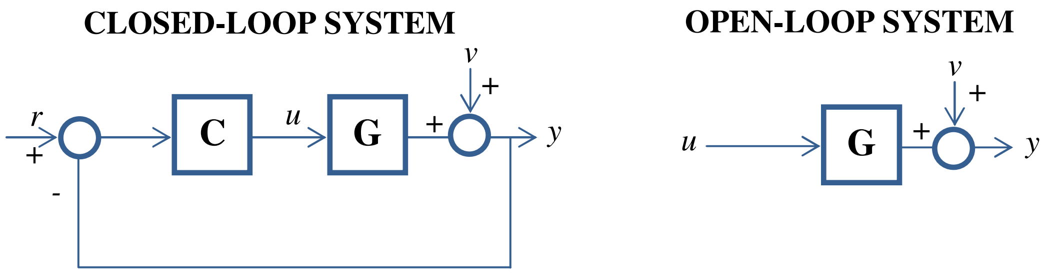


Figure 4.1: Concept of closed-loop vs. open-loop systems

Figure 4.1 illustrates the concepts of closed and open-loop configurations. Under closed-loop conditions, a signal feedback mechanism is incorporated to control a variable(s) by manipulating another. For open-loop conditions there is no control on the input to the system which can thus be manipulated as desired. Most model validation techniques are designed to work under these conditions as it provides data that is rich and informative. However, the need for closed-loop model validation techniques is pivotal from a process operation point of view and a control theoretical point of view. Some processes are inherently unstable in open-loop conditions; in other cases where MPM may be present it will be too costly to perform a complete re-identification. The following factors contribute enormously to the need to monitor process models under closed-loop conditions:

Large Scale and complex plant set-ups: Industrial processes exhibit various combinations of MVs and CVs. In general a small MPC controller may have in the range of 3 to 5 MVs in conjunction with 5 to 10 CVs. On the other hand, a large sized MPC controller will have 10 to 20 MVs controlling CVs in the region of 20 to 40. Some CVs are known to have a very slow response i.e. they possess dominant time constants which range from 30 minutes to several hours. This dictates relatively long times for identification tests. Examples of such CVs are product grades or qualities. Other CVs are very fast with time constants being a few minutes such as valve positions. Common to MPC applications are the existence of inverse responses, non-minimum phase behaviour, oscillating behaviour as well as time delays. For a large controller, a large number of inputs have no effect on some of the outputs i.e. the transfer functions relating these inputs to outputs are practically sparse. Zhu (2003) states that as much as 50 % of the process transfer functions are zero and they need to be located and fixed at zero.

High level and slow disturbances: Unmeasured disturbances typically possess slow and irregular variations. A typical source of such disturbances stem from, for example, feed composition variations and weather changes. During an identification test, the contribution of disturbances can be as high as 40% of that of the CV variation and as a result, the test signal amplitudes become too large. This is not allowed as they will result in off-specification of product and/or nonlinearity.

Local nonlinearity: Models identified for use in MPC applications are often linear in their range of operation. However, in the case where CVs are very pure product qualities or valve positions close to their limits, some non-linear behaviour may still show up.

Based on these observations, model validation under closed-loop conditions will serve to benefit industry immensely as techniques developed will run parallel with the plant and thus will not cause any disruptions to the running of the plant.

4.2. CLOSED-LOOP MODEL VALIDATION TECHNIQUES

Developing automated model quality indications and MPM detection techniques under closed-loop conditions has been studied by researchers not until very recently as demand for MPC maintenance and sustainable performance heightens. A number of methods have been proposed and developed and these can be divided into four different classifications:

Table 4.1:
Summary of the classes of model validation methods

Process history-based methods	Statistical analysis	Qualitative methods	Quantitative methods
Quality Trend Analysis (QTA)	Minimum Variance	Residual Plots	Gain Identification
Principal Components Analysis (PCA)	Statistical Indices (Harris Index)	Residual cross-correlation analysis	Recursive Least-Squares Regression
MPC monitoring by Neural Networks	Generalized Likelihood Ratio (GLR)	Spectral Analysis	Kalman Filter for parameter estimation

4.2.1. Process History-based Methods

Process history-based methods, also known as Pattern Classification methodology, only require large amounts of historical process data in order to monitor the performance of

MPC controllers by providing a diagnosis for a degradation of performance if poor performance has been noted. These techniques are widely applied in process industries since they are easy to implement and require very little modeling.

4.2.1.1. Quality Trend Analysis (QTA)

This is by the far the simplest of the pattern classification methods since it merely requires the graphical representation of process data. It does, however, provide an intuitive idea as to whether or not the data is producing the vital results. This form of analysis seeks to find these significant factors:

- Detection of an outlier(s).
- Detection of a change in process data trends.
- Detection of an abrupt shift in data levels.

Detection of an outlier: This is illustrated in figure 4.2. The point that is circled is an outlier. If the data had been taken in a real industrial process, the manner in which data such as the encircled point is acquired should be reexamined i.e. there could possibly be a sensor or transmitter fault.

Detection of a change in process data trends: Maintenance engineers often have an idea of the trend that certain data sets should follow (with respect the data's set-point). However over time the underlying process may shift from its expected trend as shown in figure 4.3. Often process data shifts are accompanied by the influence of environmental changes or plant changes. This plot provides a good indication of which factors may influence this trend change (e.g. day/night conditions) but cannot provide further information in order to distinguish between the underlying causes.

Detection of an abrupt shift in data levels: often it is seen that abrupt changes in data plots are accompanied by process parameters changes. For example, figure 4.4 illustrates the abrupt change in the data plot over a certain period. This plot illustrates that the shifted plot still exhibits similar variations to the expected plot and could thus be seen as a process gain (*K*) change (see chapter 5 for definition of process gain).

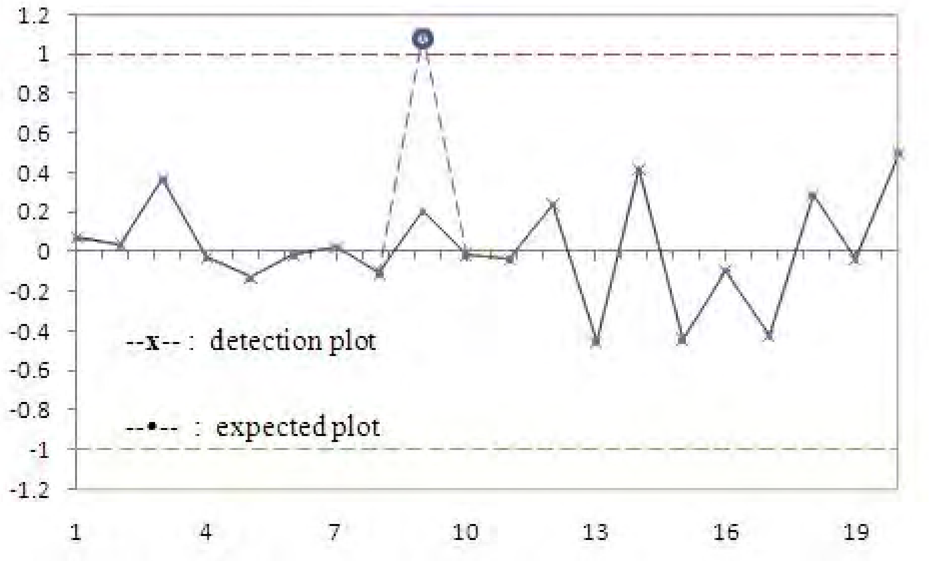


Figure 4.2: Illustration of the detection of an outlier.

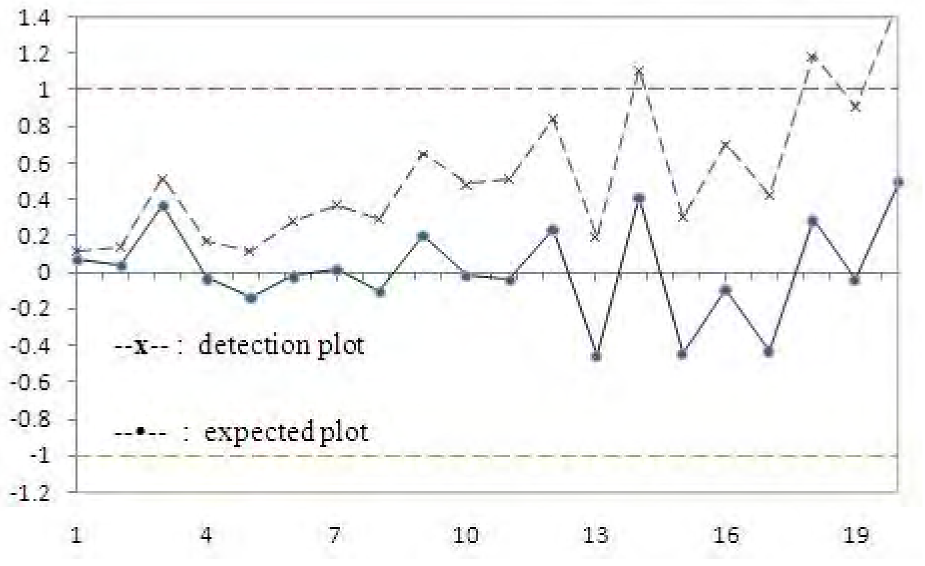


Figure 4.3: Illustration of a change in trend

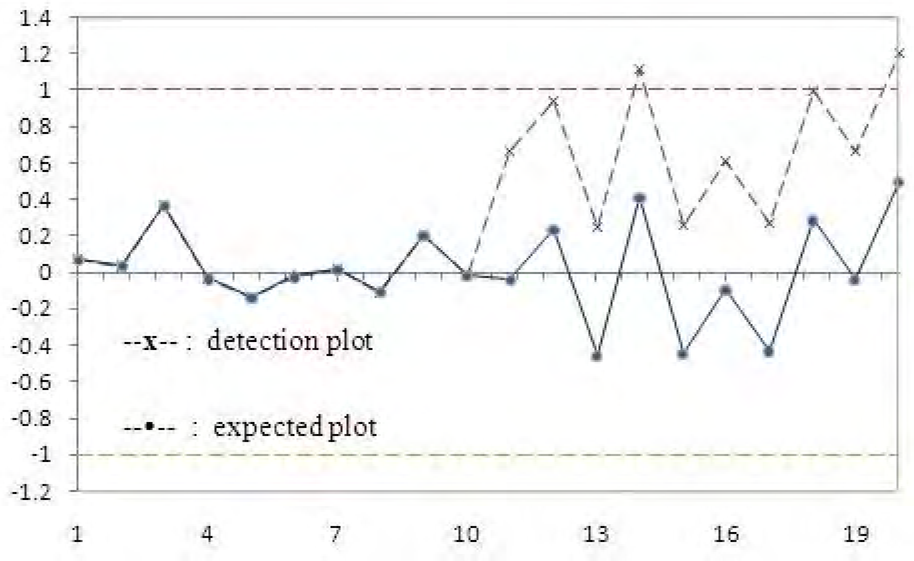


Figure 4.4: Illustration of a detection of an abrupt shift in level

4.2.1.2. Principal Components Analysis and Neural Networks

These two methods form the crux of MPC monitoring by pattern classification. Due to their modest computational requirements and sound theoretical basis, each method has been regarded as highly desirable techniques upon which one may base tools for monitoring processes.

PCA: The primary benefit of Principal Components Analysis is its ability to reduce large data sets to smaller ones that still contain the pertinent information found in the larger sets. A reduced set is much easier to analyze and interpret. Loquasto and Seborg (2003) developed a tool primarily with the use of PCA. This tool required the development of pattern classifiers which depicted several closed-loop behavioral MPC responses. These classifiers coupled with the use of PCA on current operating data aids in classifying the MPC behavior as either normal or abnormal; which is regarded as either an unusual plant disturbance or a significant plant change. If a plant change has occurred, this method does not distinguish which submodel(s) within the model matrix of the MPC is responsible for the mismatch detection.

Neural Networks: Loquasto and Seborg (2003) developed another pattern classification technique by using neural networks instead of PCA. This method works in the same manner as the previous technique in that it requires a database of closed-loop responses for MPC behavior. The additional benefit of this method lies in the fact that it contains other classifiers which subsequently diagnose specific submodel(s) that no longer represent the plant accurately.

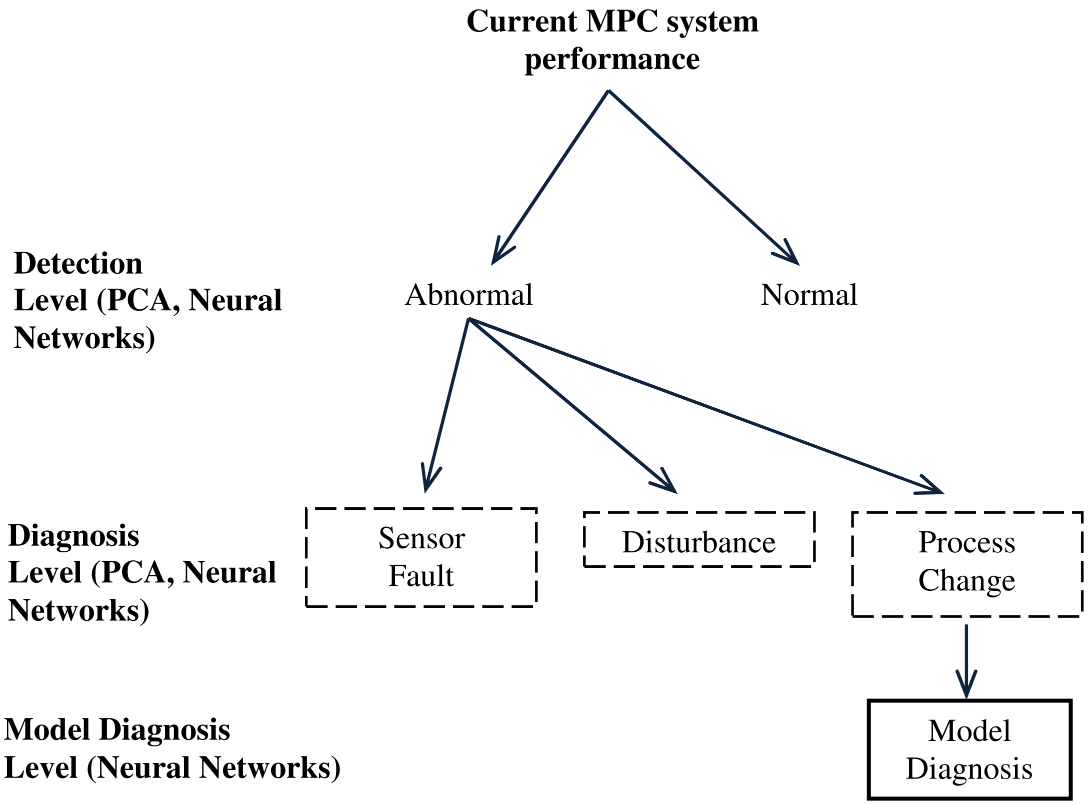

Current MPC system performance
Detection Level (PCA, Neural Networks)
Abnormal
Normal
Diagnosis Level (PCA, Neural Networks)
Sensor Fault
Disturbance
Process Change
Model Diagnosis Level (Neural Networks)
Model Diagnosis

Figure 4.5: monitoring techniques using PCA and Neural Networks (Loquasto, 2003)

Although these techniques are suitable for the diagnosis of model mismatches, its major limitation is in the availability of measurements.

4.2.2. Statistical Analysis

The majority statistical controller performance assessment techniques simply entail the comparison of the current controller behavior to some standard, usually formed at the commissioning stage. Harris (1996) laid the foundations of this research by proposing a performance benchmark based on the performance of a minimum variance controller.

This prediction index compares the minimum achievable prediction error variance to the actual variance.

$$\pi^{MV} = 1 - \frac{\sigma_{MV}^{2}}{\sigma_{ACTUAL}^{2}} \tag{4.1}$$

Numerous other indices have been established based on the tightness of control and all indices are defined as equation (4.1) but with the prediction error variance given by the standard of control. Figure 4.6 depicts these diverse standards of control:

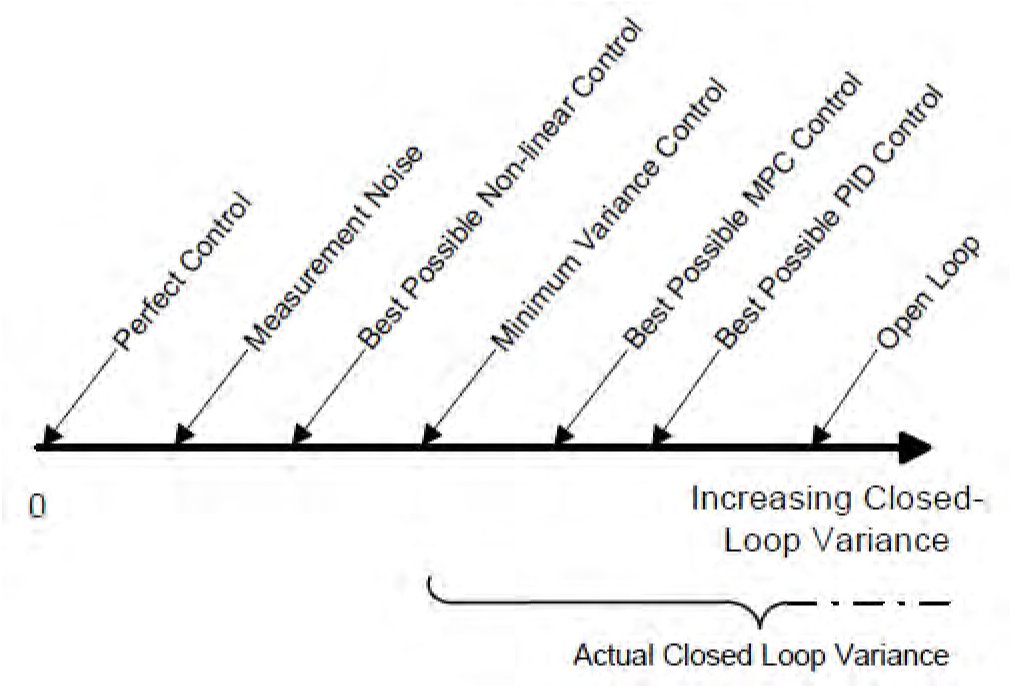

Perfect Control
Measurement Noise
Best Possible Non-linear Control
Minimum Variance Control
Best Possible MPC Control
Best Possible PID Control
Open Loop
0
Increasing Closed-Loop Variance
Actual Closed Loop Variance

Figure 4.6: Set of standard control structures (Hugo, 2001)

This type of benchmark can be significantly affected by unmeasured disturbance characteristics. Consequently, it cannot not differentiate between the effect of a model mismatch or unmeasured disturbances on controller performance. It does however serve as the foundation for controller monitoring.

Another MPC monitoring and diagnosis tool using statistical analysis was developed by Qiang and Shaoyuan (2005). This tool is also based on a bench-mark standard. However the benchmark is achieved by recording a set of output data for when the control performance is good according to the maintenance engineer's discretion. Model validation was done by making use of a method known as the Generalized Likelihood Ratio (GLR) method. This method distinguishes the cause of poor controller performance as either a plant-model mismatch or due to a disturbance term. The shortfall here, although rather minor, is the choice of the benchmark for this tool.

4.2.3. Other Related Work

Algorithms to detect abrupt parameter changes have been favoured over the past few years - primarily due to the fact that they require much less effort in the detection of model changes by parameter detection algorithms than by complete re-identification of models (Zhang et al., 1994). Several parameter change and fault detection algorithms have been developed (Basseville, 1998). Among them is the 'local' approach which has regained noteworthy interest in recent times. This method has been employed in monitoring several critical processes such as nuclear power plants, gas turbines, catalytic converters etc., which reiterate the effectiveness and reliability of this approach. The local approach has a number of distinct features; the most important is perhaps its ability to detect small parameter changes.

Haung (1999) developed a methodology for model validation in MPC systems based on a two-model divergence algorithm. This method is capable of detecting MPM regardless of the nature of the disturbance changes. It relies heavily on input excitation; however the author proposed that such signal excitation may be injected into the system via the optimizer. Parametric techniques such as the local approach and the two-model divergence algorithms based on simulated as well as live plant test results remain exceptionally good quality tools for model validation for MPC systems. They are, however, fairly complicated to implement in an online environment. Selvanathan (2010) developed a new quantity in the frequency domain, called the plant model ratio (PMR), which provides a unique signature plot (or quantitative measure) for parameters such as gain, time delay and time constants. Although this method is very sound in its approach, it is limited only to SISO systems.

4.3. Residual Analysis

Residual analysis forms the basis for the qualitative and quantitative methods given in table 4.1; the residual representing the "difference between various functions of the outputs and the expected values of the functions under normal operating conditions". Residual analysis is a very well developed field of model diagnosis as it allows for the study of the existence and nature of model inadequacies. The methods developed on the basis of residuals serve as the forefront of the present work because of their moderate complexity and efficient applicability to MIMO system model validation. These techniques are also suitable for development in an online environment.

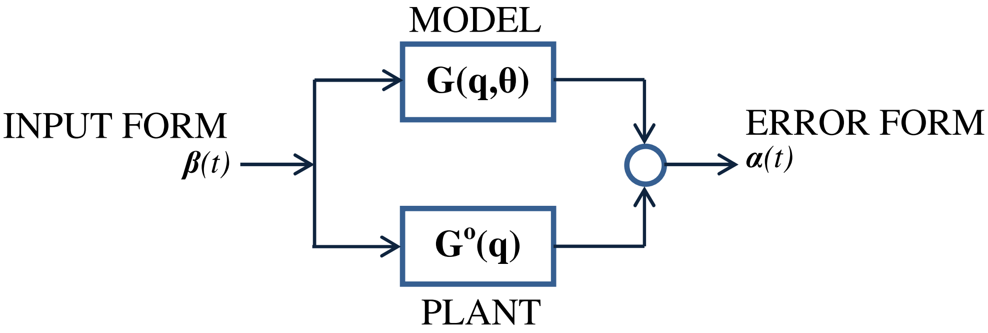


Figure 4.7: Residual schematic

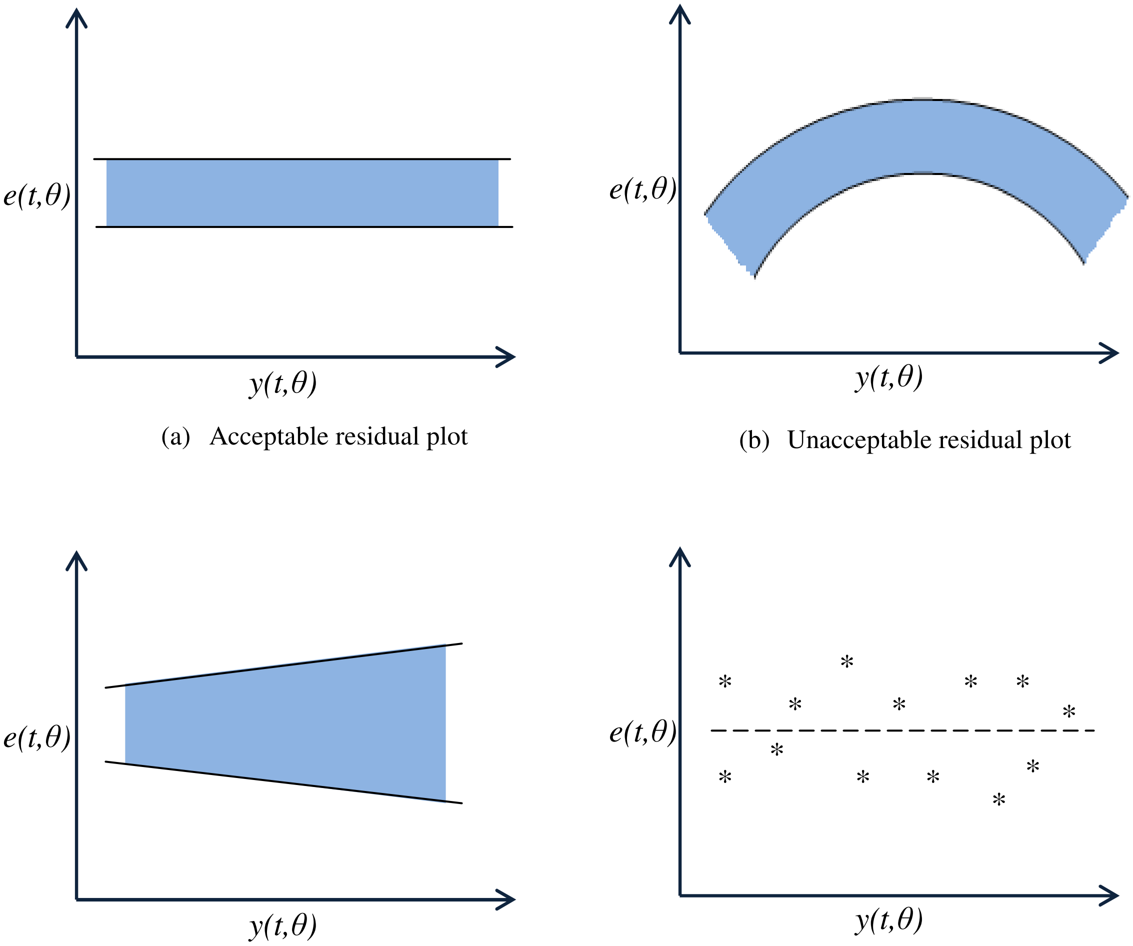
Figure 4.7 depicts the manner in which a residual is obtained. It should be noted that the input and error terms can be denoted in a variety of ways; all notation from this point is in keeping with notation of the methodology developed for the present work. They are denoted as $\boldsymbol{\beta}(t)$ and $\boldsymbol{\alpha}(t)$. The choice of error and input form is linked to quantitative analysis and is useful to handle several industrial situations which will be shown in chapter 5. For the qualitative approach, the input, denoted by $\boldsymbol{u}(t)$ and error, denoted by $\boldsymbol{e}(t)$ will be used.

4.3.1. Qualitative Methods

The qualitative approach in terms of residuals involves standard residual plotting, residual cross-correlation analysis and spectral analysis.

4.3.1.1. Standard Residual Plotting

Standard Residual Plotting is similar to QTA plots, in this case residuals, $\boldsymbol{e}(t,\theta)$, are plotted against predicted model values, $y(t,\theta)$. A residual plot allows you to determine if your regression model is a good fit to your data.



(a) Acceptable residual plot

(b) Unacceptable residual plot

(c) Unequal variance

(d) Normally distributed random scatter

Figure 4.8: Comparison of residual plots

When plotted, the residuals should:

- Be a horizontal band of haphazard points as in figure 4.8(a). If the residual pattern has a slope or is curved, then your regression model is not accounting for all but the random variation in the data. An example of the latter is shown in figure 4.8(b).

- They must have about the same width throughout the range. If they do not, then the model does not meet the requirement for equal variance. This is illustrated by figure 4.8(c).

- They must be uniformly scattered along the horizontal axis as in figure 4.8(d). If they are not then that data is regarded as clustered and the regression model could be biased.

- They must be random. There should be no recognizable pattern. Good regression models give uncorrelated residuals.

The random and haphazard representation of residual plots can be explained by recalling equation (3.44) and (3.45):

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

$$\hat{y}(t|t-1) = H^{-1}(q,\theta)G(q,\theta)\ u(t) + \left[1 - H^{-1}(q,\theta)\right] y(t)$$

The difference between the two equations gives a form of error as:

$$\begin{aligned} e(t) &= y(t) - \hat{y}(t|t-1) \\ &= -H(q,\theta)G(q,\theta)u(t) + H^{-1}(q,\theta)y(t) \\ &= \varepsilon(t) \end{aligned} \tag{4.2}$$

Thus for an accurate model, the residuals should be Gaussian (white noise) with a fixed variance and a zero mean.

4.3.1.2. Residual Correlation Analysis

Analysis of the correlation amongst residuals and between residuals and system inputs are common tools employed for linear model validation approaches. Correlation amongst residuals is referred to as the whiteness test and correlation amongst these residuals and corresponding system inputs is referred to as the independence test.

According to the **whiteness test** criteria (Mathworks, 2010), “a good model has the residual autocorrelation inside the confidence interval of corresponding estimates, indicating that the residuals are uncorrelated”. The residual auto-covariance is given by:

$$r_{ee}(l)=\overline{E}\{[e(t)-\overline{e}(t)][e(t-l)-\overline{e}(t)]\} \tag{4.3}$$

This equation is normalized to range from -1 to 1 by division of the product of the standard deviations of the two variables involved. It is referred to as the residual auto-correlation:

$$R_{ee}(l)=\frac{r_{ee}(l)}{\sigma_e\sigma_e} \tag{4.4}$$

According to the **independence test** criteria (Mathworks, 2010), “a good model has residuals that are uncorrelated with past inputs. Evidence of a correlation indicates that the model does not describe how part of the output relates to the corresponding input”. The covariance between a residual and input say, u_i, is given by:

$$r_{eu_i}(l)=\overline{E}\{[e(t)-\overline{e}(t)][u_i(t-l)-\overline{u}_i(t)]\} \tag{4.5}$$

This equation, referred to as the cross-correlation, can also be normalized as follows:

$$R_{eu_i}(l)=\frac{r_{eu_i}(l)}{\sigma_e\sigma_{u_i}} \tag{4.6}$$

Equation (4.5) and (4.6) are exceptionally useful in dealing with MIMO systems. For example consider the following 3 x 3 system:

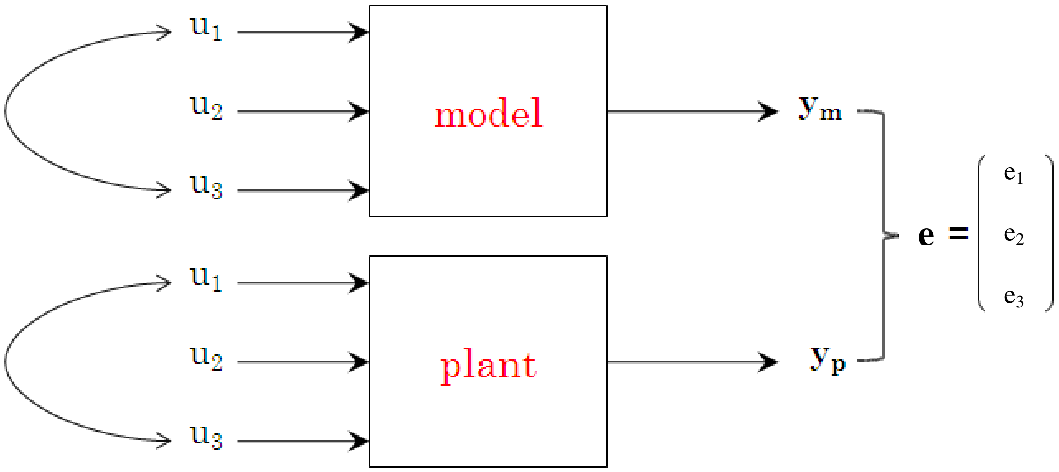


Figure 4.9: illustration of a 3 x 3 system with correlation between u_1 and u_3

$$\hat{\boldsymbol{G}} = \begin{bmatrix} \hat{G}_{11} & \hat{G}_{12} & \hat{G}_{13} \\ \hat{G}_{21} & \hat{G}_{22} & \hat{G}_{23} \\ \hat{G}_{31} & \hat{G}_{32} & \hat{G}_{33} \end{bmatrix} \tag{4.7}$$

The residual matrix, using equation (4.6), for the above 3 x 3 system is given by:

$$\boldsymbol{R}_{\mathrm{eu}} = \begin{bmatrix} R_{e_1 u_1} & R_{e_1 u_2} & R_{e_1 u_3} \\ R_{e_2 u_1} & R_{e_2 u_2} & R_{e_2 u_3} \\ R_{e_3 u_1} & R_{e_3 u_2} & R_{e_3 u_3} \end{bmatrix} \tag{4.8}$$

Suppose G_{11} is contains a mismatch with the plant then:

$R_{e_i u_j} = 0 \forall i, j$; Except $R_{e_1 u_1} \neq 0 \forall i = 1, j = 1$

(N.B. i = 1, 2, 3 for all outputs and j=1, 2, 3 for all inputs)

Due to the feedback mechanism of MPC controllers, some inputs may become correlated with each other, which is unavoidable. Such correlations amongst MVs may confound the regular correlation analysis between residuals and MVs. This may result in an incorrect diagnosis of MPM, for example if u_1 and u_3 are correlated then:

$$R_{\varepsilon_i u_j} \neq 0 \forall i = 1, j = 1,3$$

The cross-correlation indicates that G_{13} may contain a mismatch but this is due to correlation amongst the inputs. One may use a cross-correlation of the inputs (with time shift) to check the correlation of the inputs:

$$r_{u_j u_i}(l) = \overline{E}\{[u_j(t) - \bar{u}_j(t)][u_i(t-l) - \bar{u}_i(t)]\} \tag{4.9}$$

$$R_{u_j u_i}(l) = \frac{r_{u_j u_i}(l)}{\sigma_{u_j} \sigma_{u_i}} \tag{4.10}$$

From these correlation analysis methods, coupled with the analysis of correlation amongst inputs, the following diagnosis table is formulated:

Table 4.2:
Significance of correlation combination results

$R_{e_i u_j}$	$R_{u_i u_j}$	**SIGNIFICANCE**
$= 0$	$= 0$	No MPM: system model is adequate
$\neq 0$	$= 0$	MPM present: system model is inadequate
$\neq 0$	$\neq 0$	MPM present in model G_{ij}; wrong diagnosis for model G_{ii} due to correlation between u_i and u_j.
$= 0$	$\neq 0$	No MPM: correlation present amongst u_i and u_j

Badwe et al., (2008) developed a method of MPM detection based on the correlation approach. Instead of using the conventional residual correlation analysis, they opted to remove the effect of other inputs on a specific input thereby removing the influence of correlated inputs. In this case, only correlations between residuals and inputs (free of the effects of other inputs) are tested. A method developed in the present work, in chapter 5, is also established to handle the case of correlated inputs.

4.3.2. Quantitative Methods

Qualitative methods that are used for the diagnosis of MPM are useful for detecting MPM in MIMO systems by providing an indication of which submodel(s) contain the mismatch. Intuitively this is often what is required. However, they do not provide an indication of the extent of the mismatch i.e. is it 10% mismatch in the time-delay term or is it 50% mismatch in the gain term? Thus, these qualitative methods can be used as an initial indication of the presence of MPM, but quantitative methods are required to provide an exact value(s) that describes by how much a model is mismatched to the plant.

The area of quantitative model validation techniques has received very little attention in this particular field of study. Quantitative model-based methods include linear regression and the use of the Kalman filter adapted for parameter estimation. These methods are often associated with the evaluation of a large number of coefficients and thus seen as time-consuming techniques to be employed in an online environment under closed-loop conditions when the major requirement for regression is rich informative data. However, when the regression is focused on the important parameters such as the gain, time delay and time constant, then these techniques become extremely important in model diagnosis.

Chapter 5 extends on the idea of using regression theory to provide a diagnosis for MPM. Several relevant factors limit the ability to develop these techniques under closed-loop conditions. Such factors include the correlation of inputs, set-point excitation and noise levels. A detailed study of quantitative methods as well as ideas to tackle these factors is dealt with in chapter 5.

CHAPTER 5 REGRESSION IDENTIFICATION THEORY

This chapter presents the methods developed in the present work. Closed-loop model validation for Model Predictive Control has been seen to rely primarily on qualitative approaches. This chapter seeks to shift this paradigm by employing techniques that will detect MPM and subsequently quantify the extent of the mismatch. Although the theory is limited to gain mismatch identification, it is shown to extend to the remaining parameters present in a model(s) to some extent. This chapter illustrates the theory developed for the models employed for SASOL controllers. The various types of error definitions are given. The regression methods, composed of least squares regression and kalman filter parameter estimations, are based on a general form of error which allows for a choice of error definition suitable for a specific industrial situation. Several factors noted at the end of chapter 4 are covered at the end of this chapter.

5.1. GENERAL CONSIDERATIONS

The choice of the MPC algorithm which is employed in the SASOL Secunda plant is the RMPCT algorithm. For the identification of the models used for control in these controllers, RMPCT adopts a three-step technique for the identification of its models. Firstly, a Box-Jenkins model is identified using a LS method or by PEM; alternatively the Cholesky decomposition is employed to identify an FIR model is identified using (Qin et al., 2003). The identified model is then fitted to a low-order ARX model to smooth out large variance due to possible over-parameterization in the FIR model. These lower order ARX models are then converted into Laplace transfer functions.

From these observations, it can be seen that these models are formulated in an input-output form. Thus there would be no interaction amongst outputs in this model formulation because their interactive contributions are effectively dealt with by substituting their behaviour in terms of the inputs. The benefit drawn from here is that each output can be dealt with independently from the rest i.e. an *n* x *m* MIMO system can be considered as *n* separate MISO systems. Another observation is that these models are presented as Laplace domain models (See appendix B). Thus, an approximation for the Laplace domain operator, s, is required to convert these models into a form that can be used on a real-time basis.

5.2. MODELLING

Consider a single output of the MIMO system expressed in input-output form:

$$y(s) = K_1 e^{-\tau_1 s} \frac{1+c_{11}s+c_{12}s^2+...}{1+d_{11}s+d_{12}s^2+...} u_1(s) \;+\; K_2 e^{-\tau_2 s} \frac{1+c_{21}s+c_{22}s^2+...}{1+d_{21}s+d_{22}s^2+...} u_2(s) \;+\; ...$$

$$= \sum_{i=1}^{N} K_i e^{-\tau_i s} \frac{1+c_{i1}s+c_{i2}s^2+...}{1+d_{i1}s+d_{i2}s^2+...} u_i(s)$$

$$= \sum_{i=1}^{N} K_i e^{-\tau_i s} \frac{1+\sum_{j=1}^{M_c} c_{ij}s^j}{1+\sum_{j=1}^{M_d} d_{ij}s^j} \; u_i(s) \tag{5.1}$$

This model needs to be converted from the above continuous form into a discrete form because one of the basic goals of discretization is fulfilling the need for practical realizations. In the process of discretization of a continuous system, one can use the well-known mapping of the *s*-domain into *q*-domain by substituting:

$$q^{-1} = e^{-sT} \tag{5.2}$$

T in this instance represents the sampling period and is interchangeable with Δ*t*.

Starting from the basic relation defined by (5.2) above, the following equivalent can thus be written:

$$\begin{aligned} q^{-1} &= e^{-sT} = e^{-s[(1-\gamma)T+\gamma T]} \\ &= \frac{e^{-s(1-\gamma)Ts}}{e^{\gamma Ts}} \qquad \gamma \in [0,1] \end{aligned} \tag{5.3}$$

After the numerator and denominator on the right hand side of (5.3) have been expanded by Taylor series expansions, with all but the first order approximation retained, (5.3) becomes:

$$q^{-1} = \frac{\sum_{n=0}^{\infty} (-1)^n \frac{[(1-\gamma)Ts]^n}{n!}}{\sum_{k=0}^{\infty} \frac{(\gamma Ts)^k}{k!}} \approx \frac{1-(1-\gamma)Ts}{1+\gamma Ts} \tag{5.4}$$

By solving (5.5) for the complex variable, s, the following first order approximation is obtained:

$$s = \left[\frac{1}{T}\right]\left[\frac{1-q^{-1}}{1+\gamma(q^{-1}-1)}\right] \tag{5.5}$$

Table 5.1:
Generalization of some known transformations (Sekara, 2003)

γ	***s* – *q*$^{-1}$ approximation**	**Name of approximation**
0	$\left[\frac{1-q^{-1}}{T}\right]$	Euler approximation first order – forward difference (FD)
0.5	$\left[\frac{2}{T}\right]\left[\frac{1-q^{-1}}{1+q^{-1}}\right]$	Tustin approximation – Bi linear (BL)
1	$\left[\frac{1}{T}\right]\left[\frac{1-q^{-1}}{q^{-1}}\right]$	Euler approximation first order – Backward difference (BD)
γ ϵ [0, 1]	$\left[\frac{1}{T}\right]\left[\frac{1-q^{-1}}{1+\gamma(q^{-1}-1)}\right]$	γ – approximation of first order

Table 5.1 depicts several common approximations for the Laplace operator, *s*. By transforming the *s*-domain transfer functions (using one of the approximations listed in the table above and replacing dead-time lags, *τ*, by an integer number of sampling periods, *T*), one obtains the following equation:

$$y(t) = \sum_{i=1}^{N} q^{-n_i} \frac{b_{i0} + b_{i1}q^{-1} + b_{i2}q^{-2} + \ldots}{1 + a_{i1}q^{-1} + a_{i2}q^{-2} + \ldots} u_i(t)$$

$$= \sum_{i=1}^{N} q^{-n_i} \frac{b_{i0} + \sum_{j=1}^{M_b} b_{ij}q^{-j}}{1 + \sum_{j=1}^{M_a} a_{ij}q^{-j}} \quad u_i(t) \tag{5.6}$$

In the discrete form, one expects the steady-state gain to be defined as:

$$K_i = \frac{b_{i0} + b_{i1} + b_{i2} + \ldots}{1 + a_{i1} + a_{i2} + \ldots} \tag{5.7}$$

Here one envisages evaluation of *y* at a series of discrete time-steps *t, t+T, t+2T* ... etc, based on u_{i} values at the same intervals. In practice, n_{i} cannot be less than 1 because an output cannot occur at the same time as an input. If the given equation [1] implies that any $\tau_i \leq T$, it is corrected to *T* without much error for small ***T*.**

In order to simplify the notation of equation (5.6), the following lag polynomials are defined:

$$A_i(q^{-1}) = 1 + \sum_{j=1}^{M_a} a_{ij}q^{-j} \tag{5.8}$$

$$B_i(q^{-1}) = b_{i0} + \sum_{j=1}^{M_b} b_{ij}q^{-j} \tag{5.9}$$

Substituting these lag polynomials into equation (5.6), one obtains the following simplified output representation, which is an ARX model structure:

$$y(t) = \sum_{i=1}^{N} q^{-n_i} \frac{B_i(q^{-1})}{A_i(q^{-1})} \ u_i(t) \tag{5.10}$$

This is how the “model output” $y_m(t)$ will be calculated below, i.e. it will be taken that:

$$y_m(t) = \sum_{i=1}^{N} q^{-n_i} \frac{B_i(q^{-1})}{A_i(q^{-1})} \ u_i(t) \tag{5.11}$$

To evaluate this in practice one needs the difference form obtained by multiplying through by the common denominator:

$$\left\{\prod_{i=1}^{N} A_i(q^{-1})\right\} y_m(t) \ = \sum_{i=1}^{N} \left[q^{-n_i} B_i(q^{-1}) \left\{ \prod_{\substack{j=1 \\ j\neq i}}^{N} A_j(q^{-1}) \right\} \right] u_i(t) \tag{5.12}$$

Here the first coefficient (i.e. of q^{-0}) of the lag polynomial, as shown in equation (5.8), acting on y_m will be 1, so a more useful predictive form is obtained by putting all of the lagged components on the right-hand-side, i.e.:

$$y_m(t) \ = \left[1 - \left\{ \prod_{i=1}^{N} A_i(q^{-1}) \right\} \right] y_m(t) \ + \ \sum_{i=1}^{N} \left[q^{-n_i} B_i(q^{-1}) \left\{ \prod_{\substack{j=1 \\ j\neq i}}^{N} A_j(q^{-1}) \right\} \right] u_i(t) \tag{5.13}$$

which will calculate the present (t) value of the output y based on the values of the output y at *previous* time-steps, and the values of the N inputs u_i (i=1,...,N) at *previous* time-steps.

5.2.1. Predictive forms

There are two broad descriptions of how a model can predict current output values. A model can be run like this in an “open-loop” sense along-side a process, taking in the actual values being used at the process inputs u_i (i=1,...,N) and calculating a completely independent sequence of $y_\mathrm{m}(t)$ values. Referring to figure 5.1 below, the resultant $y_\mathrm{m}(t)$

and $y_p(t)$ [actual plant] curves should then have similar shapes and local variations, but will be offset from each other, since the modeled response might only be fixed on the plant response at the initial point, if at all. This type of prediction could be called an **open-loop predictor**.

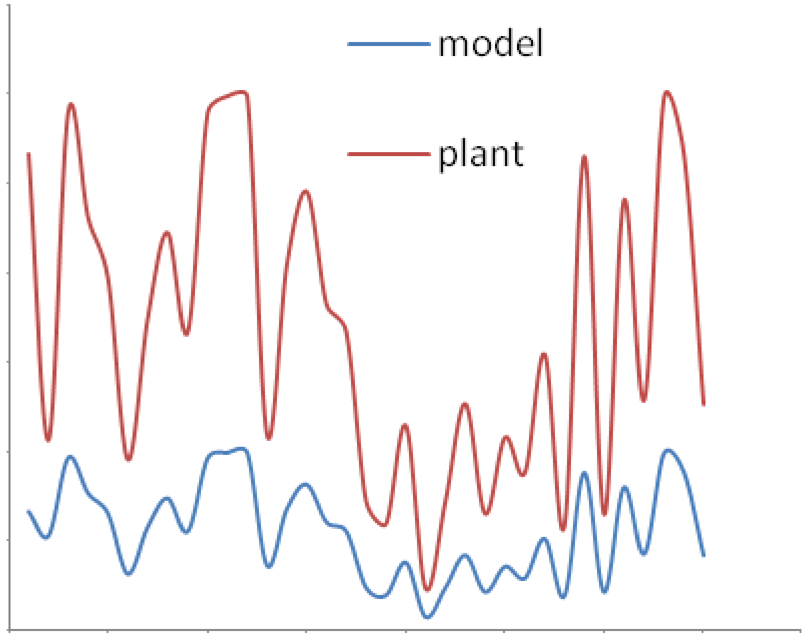


Figure 5.1: graphical representation of the use of open-loop predictor

On the other hand, the **one-step predictor** is obtained from equation (5.13) by "borrowing" previous measured outputs of the plant for the right-hand-side of the equation:

$$y_m(t) = \left[1 - \left\{\prod_{i=1}^{N} A_i(q^{-1})\right\}\right] \; y_p(t) \; + \; \sum_{i=1}^{N}\left[q^{-n_i}B_i(q^{-1})\left\{\prod_{\substack{j=1\\j\neq i}}^{N} A_j(q^{-1})\right\}\right]u_i(t) \qquad (5.14)$$

This form, shown in figure 5.2, would of course track the plant response much more closely, but would still have a discernable error due to an inexact model.

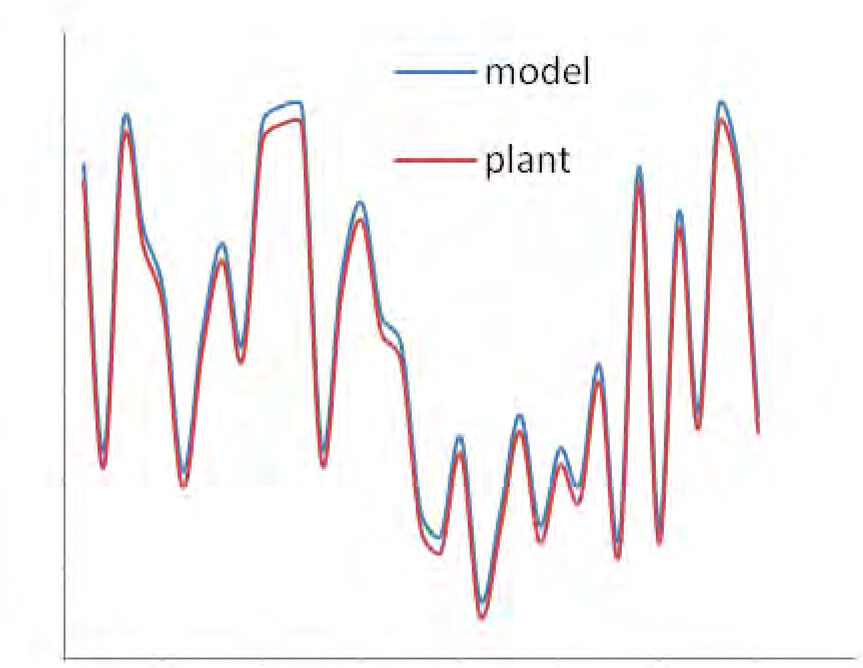


Figure 5.2: graphical representation of the use of one-step predictor

In the proposed regression, it is sought to reveal a “connection” between the predictive error of this model, and a particular $u_{i,}$ in order to determine which part of the original model in (5.1) is the likely source of error.

5.3. Error Definitions

Error definitions may be separated into two classes, with the basic form being the output error (OE), as shown in equation (3.55) and the latter class being the equation error (EE). Each of these classes contains various descriptions of error forms.

5.3.1. Output Error

The very basic definition is given by:

$$e(t) = y_m(t) - y_p(t) \tag{5.15}$$

Applying the definitions of the two predictive forms given by equations (5.13) and (5.14) respectively, the following computational forms are thus obtained:

Output Error, Open-loop Predictor:

$$e(t) = \left[1 - \left\{\prod_{i=1}^{N} A_i(q^{-1})\right\}\right] \; y_m(t) + \sum_{i=1}^{N}\left[q^{-n_i} B_i(q^{-1}) \left\{ \prod_{\substack{j=1 \\ j\neq i}}^{N} A_j(q^{-1}) \right\} \right] u_i(t) - y_p(t) \tag{5.16}$$

Output Error, One-step Predictor:

$$e(t) = \left[\left\{\prod_{i=1}^{N} A_i(q^{-1})\right\}\right] y_p(t) + \sum_{i=1}^{N}\left[q^{-n_i} B_i(q^{-1}) \left\{ \prod_{\substack{j=1 \\ j\neq i}}^{N} A_j(q^{-1}) \right\} \right] u_i(t) \tag{5.17}$$

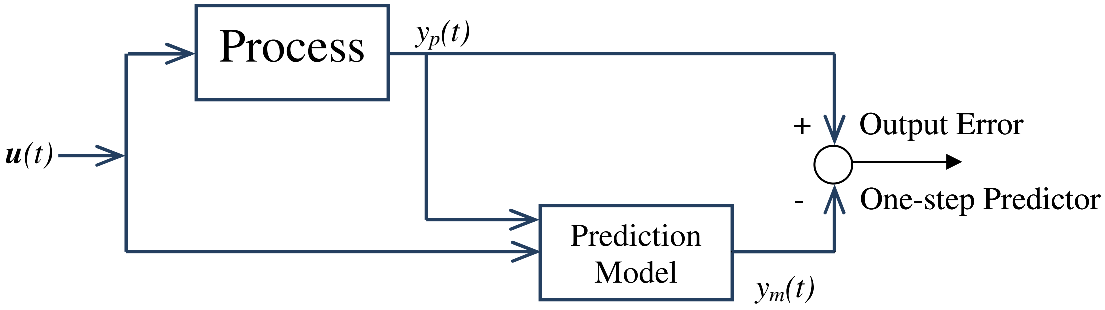


Figure 5.3: Principle of output prediction error

One notes that the error will contain the offset between plant and model, not just the errors due to the individual terms in equation (5.1). The offset can be searched for in a regression, but it is usually convenient to eliminate its effect by differentiating equation (5.15):

$$\Delta e(t) = \Delta y_m(t) - \Delta y_p(t) \tag{5.18}$$

This is a model structure with the combination of an ARX model in conjunction with an integrator. Here a differencing multiplier (1-q^{-1}) is passed through the equation:

$$\Delta e(t) = \left[1 - q^{-1}\right] e(t) = e(t) - e(t-T)$$
$$\Delta y_m(t) = \left[1 - q^{-1}\right] y_m(t) = y_m(t) - y_m(t-T)$$
$$\Delta y_p(t) = \left[1 - q^{-1}\right] y_p(t) = y_p(t) - y_p(t-T)$$

The change of error over the recent step *T* thus arises from the change in the model prediction minus the change in the plant measurement. In this difference form equations (5.16) and (5.17) become:

Output Error, Open-loop Predictor, difference form:

$$\Delta e(t) \ = \left[1 - \left\{\prod_{i=1}^{N} A_i(q^{-1})\right\}\right] \Delta y_m(t) + \sum_{i=1}^{N} \left[q^{-n_i} B_i(q^{-1}) \left\{ \prod_{\substack{j=1 \\ j\neq i}}^{N} A_j(q^{-1}) \right\} \right] \Delta u_i(t) - \Delta y_p(t) \qquad (5.19)$$

Output Error, One-step Predictor, difference form:

$$\Delta e(t) \ = \left[\left\{\prod_{i=1}^{N} A_i(q^{-1})\right\}\right] \Delta y_p(t) + \sum_{i=1}^{N} \left[q^{-n_i} B_i(q^{-1}) \left\{ \prod_{\substack{j=1 \\ j\neq i}}^{N} A_j(q^{-1}) \right\} \right] \Delta u_i(t) \qquad (5.20)$$

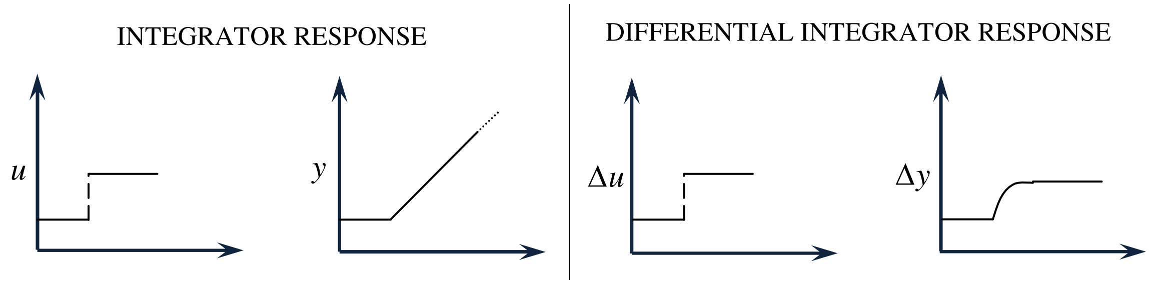
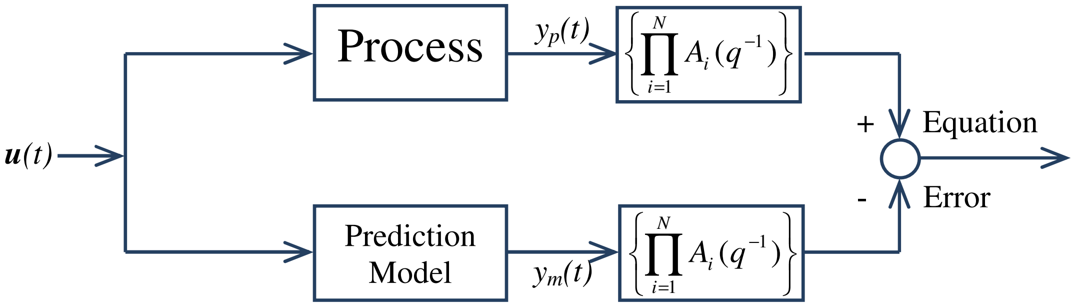


Figure 5.4: conceptual view of the use of differential variables

Apart from eliminating offset, the difference forms make it easier to handle integrating terms in equation (5.1) (i.e. when a factor 1/*s* is present). The figure above illustrates the response to a first order model containing an integrator term. A response to a step input for an integrator results in a constant ramp to infinity. This results in output response values reaching excessively large numbers, making it difficult to assess model inaccuracy. However employing differential forms to both the input and output reveals a bounded response.

5.3.2. Equation Error

So far, four possible forms (5.16), (5.17), (5.19) and (5.20) have been proposed for computation of the "model error". The intention is to regress this $e(t)$ against u_i(t) [i=1,...,N], or $\Delta e(t)$ against Δu_i(t) [i=1,...,N], in order to associate any error with a particular u_i, and thus establish which term in the model is responsible for the error. However, a better synchronization can be obtained by comparing inputs and outputs with their various lags applied (figure 5.5).



Process
$y_p(t)$
$\left\{\prod_{i=1}^{N} A_i(q^{-1})\right\}$
$\boldsymbol{u}(t)$
+
Equation
Error
-
Prediction Model
$y_m(t)$
$\left\{\prod_{i=1}^{N} A_i(q^{-1})\right\}$

Figure 5.5: Principle of equation error

Thus there is another form, the **equation error** form, in which the error $e(t)$ is viewed in terms of equation (5.12) as the difference of the left-hand-side evaluated using y_m *versus* y_p:

$$e(t) = \left\{\prod_{i=1}^{N} A_i(q^{-1})\right\} y_m(t) - \left\{\prod_{i=1}^{N} A_i(q^{-1})\right\} y_p(t) \tag{5.21}$$

So here the relationship for computing *e*(*t*) is:

$$e\ \ (t)\ =\sum_{i=1}^{N}\left[q^{-n_i}B_i(q^{-1})\left\{\prod_{\substack{j=1\\ j\neq i}}^{N}A_j(q^{-1})\right\}\right]\ u_i(t)\ \ -\left\{\prod_{i=1}^{N}A(q^{-1})\right\}y_p(t) \tag{5.22}$$

Because the *y* contributions to the error are now spread over the range of *y*-lags, this turns out to be a smoother regression. In the case where the one-step predictor form is used for the prediction of the model output, $y_{\mathrm{m}}(t)$, note that:

$$\begin{aligned} e(t) &= y_m\ \ (t)+\left\{\prod_{i=1}^{N}A_i(q^{-1})-1\right\}y_p(t)\ \ -\left\{\prod_{i=1}^{N}A(q^{-1})\right\}y_p(t) \\ &= y_m(t)-y_p(t) \end{aligned} \tag{5.23}$$

Finally, equations (5.16), (5.17), (5.19) and (5.20) can be written out for the equation-error form as follows:

Equation-error, open-loop predictor & one-step predictor:

$$e(t)\ =\sum_{i=1}^{N}\left[q^{-n_i}B_i(q^{-1})\left\{\prod_{\substack{j=1\\ j\neq i}}^{N}A_j(q^{-1})\right\}\right]u_i(t)\ \ -\left\{\prod_{i=1}^{N}A(q^{-1})\right\}y_p(t) \tag{5.24}$$

Equation-error, open-loop predictor & one-step predictor, difference form

$$\Delta e(t)\ =\sum_{i=1}^{N}\left[q^{-n_i}B_i(q^{-1})\left\{\prod_{\substack{j=1\\ j\neq i}}^{N}A_j(q^{-1})\right\}\right]\Delta\ \ u_i(t)\ \ -\left\{\prod_{i=1}^{N}A(q^{-1})\right\}\Delta y_p(t) \tag{5.25}$$

5.4. GENERAL REGRESSION EQUATION

One seeks "best fit" k_i in a relationship defined below N.B. k_{N+1} is an offset:

$$\alpha(t) = \sum_{i=1}^{N} k_i \beta_i + k_{N+1} \tag{5.26}$$

From equation (5.10), the output error is:

$$e(t) = y_m(t) - y_p(t) = \sum_{i=1}^{N} q^{-n_i} \frac{B_i(q^{-1})}{A_i(q^{-1})} u_i(t) - y_p(t) \tag{5.27}$$

The difference form of the output error is:

$$\Delta e(t) = \sum_{i=1}^{N} q^{-n_i} \frac{B_i(q^{-1})}{A_i(q^{-1})} \Delta u_i(t) - \Delta y_p(t) \tag{5.28}$$

One should note that at steady state the gain is given by:

$$B_i(1) = b_{i0} + b_{i1} + b_{i2} + ...$$

$$A_i(1) = 1 + a_{i1} + a_{i2} + ...$$

$$K_i = \frac{B_i(1)}{A_i(1)} \tag{5.29}$$

The table below shows α and β_i for the different cases that were considered, bearing in mind equations (5.27) and (5.28):

One notes that the measurements of the inputs u_i could also be faulty, and would give non-zero values of k_i in the same way as erroneous K_i values. Errors in individual coefficients of $A_i(q^{-1})$ and $B_i(q^{-1})$ will also cause variation of k_i away from zero, but will be much more difficult to distinguish.

Table 5.2:
Summary of expected k_i parameter fittings for the various cases

Eqn	**Form**	$\boldsymbol{\alpha}$	$\boldsymbol{\beta}$	**Expected k_i at steady state**	**Expected k_{N+1} at steady state**
5.16	Output-error, open-loop predictor	e	u_i	K_{mi} - K_{pi}	y_p
5.17	Output-error, one-step predictor	e	u_i	K_{mi} - K_{pi}	y_p
5.19	Output-error, open-loop predictor, difference form	Δe	Δu_i	K_{mi} - K_{pi}	0
5.20	Output-error, one-step predictor, difference form	Δe	Δu_i	K_{mi} - K_{pi}	0
5.24	Equation-error, open-loop predictor <u>&</u> one-step predictor	e	$\left[q^{-n_i}B_i\left(q^{-1}\right)\left\{\prod_{\substack{j=1\\j\neq i}}^{N}A_j\left(q^{-1}\right)\right\}\right]u_i\left(t\right)$	$1-\dfrac{K_{pi}\left\{\prod_{j=1}^{N}A_{pj}\left(1\right)\right\}}{K_{mi}\left\{\prod_{j=1}^{N}A_{mj}\left(1\right)\right\}}$	$-\left\{\prod_{i=1}^{N}A_{mi}\left(1\right)\right\}y_p$
5.25	Equation-error, open-loop predictor & one-step predictor, difference form	Δe	$\left[q^{-n_i}B_i\left(q^{-1}\right)\left\{\prod_{\substack{j=1\\j\neq i}}^{N}A_j\left(q^{-1}\right)\right\}\right]\Delta u_i\left(t\right)$	$1-\dfrac{K_{pi}\left\{\prod_{j=1}^{N}A_{pj}\left(1\right)\right\}}{K_{mi}\left\{\prod_{j=1}^{N}A_{mj}\left(1\right)\right\}}$	0

In the cases of the output error forms [(5.16), (5.17), (5.19) and (5.20)], consider a model error caused by a gain K_{mi} in equation (5.1) being a factor of 1.5 × its correct (K_{pi}) value.

The regression would determine a coefficient on the corresponding u_i which increases above its correct value of zero to:

$$k_i = +0.5K_{pi}$$

In the cases of the equation error forms (5.24) and (5.25), k_i gives the relative gain error directly, e.g. if $K_\mathrm{mi}=2K_\mathrm{pi}$ [but $A_\mathrm{mi}(1)=A_\mathrm{pi}(1)$], one would obtain $k_\mathrm{i}=+0.5$.

5.5. Recursive Regression by Moving Window

Regression Analysis (RA) is a conventional statistical technique that is commonly used in control theory. One particular class of RA is the Moving Window Regression (MWR) method. The MWR method uses the same general linear regression model as the RA. The only difference is that the regression is being applied to a smaller area (a window) instead of the whole data record. Therefore, several regressions are performed in order to cover the data period. By doing so, the MWR method is able to capture temporal variations in the study region in which the RA method does not have the capability to do so. An additional benefit is that this form of regression can be implemented in an online environment with a slight reliance on historical data based on the window size.

5.5.1. Moving Window Concept

The moving window concept is depicted below. The notation used for the representation of data is defined as:

$$\mathbf{Z}_i \equiv [y_i, \mathbf{u}_i]$$

In this case the subscript, *i*, indicates the respective time interval *i*.

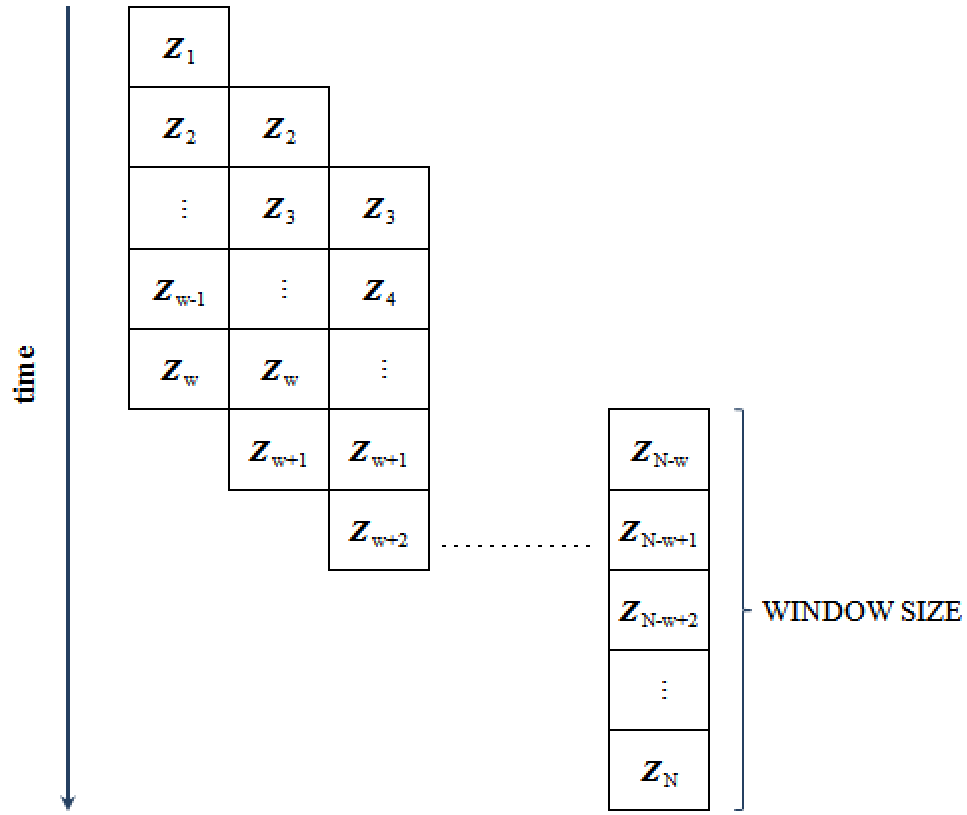


Figure 5.6: Moving Window Concept

Computation is made efficient by advancing the window one position on each time-step and removing the outgoing data point and adding in the incoming point. This maintains the window size at each time-step.

5.5.2. Mathematical Formulation

The general regression equation (5.26) is re-written in matrix vector-form as:

$$\alpha_t = \boldsymbol{k}^T \boldsymbol{\beta}_t \quad \text{where } \boldsymbol{k} = \begin{pmatrix} k_1 \\ k_2 \\ \vdots \\ k_N \\ k_{N+1} \end{pmatrix} \text{ and } \boldsymbol{\beta}_t = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_N \\ 1 \end{pmatrix}_t \tag{5.30}$$

One has corresponding α_t and $\boldsymbol{\beta}_t$ at a series of times *t =1, 2 ... M* in a moving window, and wishes to obtain an appropriate set of $\boldsymbol{k}$ values for the present window. The following objective function is defined:

$$J(\boldsymbol{k}) = \sum_{t=1}^{M} \left[\alpha_t - \boldsymbol{k}^T \boldsymbol{\beta}_t \right]^2 + \left[\boldsymbol{k} - \boldsymbol{k}_{\text{exp}} \right]^{\text{T}} \boldsymbol{\Lambda} \left[\boldsymbol{k} - \boldsymbol{k}_{\text{exp}} \right] \tag{5.31}$$

where $\boldsymbol{k}_{\text{exp}} = \begin{pmatrix} k_{1\text{exp}} \\ k_{2\text{exp}} \\ \vdots \\ k_{N\text{exp}} \\ k_{N+1\text{exp}} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}$

and the weighting matrix $\boldsymbol{\Lambda}$ $eg. = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 0.01 \end{bmatrix}$

Here the expected $\boldsymbol{k}$ values ($\boldsymbol{k}_{\text{exp}}$) are set to zero marking the point where model and plant should be in agreement (see table 5.2 above). The offset term $k_{\text{N+1}}$ is also given a target of zero to reduce its wandering, but with a much lower weighting than the coefficient k_{i}'s.

Setting $\dfrac{\partial J}{\partial \boldsymbol{k}} = \boldsymbol{0}$ leads to the best fit:

$$\boldsymbol{k} = \left[\left\{ \sum_{t=1}^{M} \boldsymbol{\beta}_t \boldsymbol{\beta}_t^T \right\} + M\boldsymbol{\Lambda} \right]^{-1} \left[\left\{ \sum_{t=1}^{M} \alpha_t \boldsymbol{\beta}_t \right\} + M\boldsymbol{\Lambda} \boldsymbol{k}_{\text{exp}} \right] \tag{5.32}$$

The suppression of deviation of $\boldsymbol{k}$ from $\boldsymbol{k}_{\text{exp}}$ gives control of the tendency of the k_{i} values to wander over large ranges seeking marginal reductions in *J*, whereas they are fully expected to be close to the $\boldsymbol{k}_{\text{exp}}$ values.

$$\text{cov ariance matrix} = \left\{ \sum_{t=1}^{M} \boldsymbol{\beta}_t \boldsymbol{\beta}_t^T \right\}$$

$$\text{scalar vector product} = \left\{ \sum_{t=1}^{M} \alpha_t \boldsymbol{\beta}_t \right\}$$

The calculation is made efficient for advancing the window one position on each time-step by maintaining the summed covariance matrix and the summed scalar vector product in registers which merely subtract the outgoing point and add the incoming point on each time-step.

5.6. REGRESSION BY KALMAN FILTER

Since the publication of Kalman's landmark paper in 1960, Kalman filters have become ubiquitous in state estimation, system identification, adaptive control, signal processing and have found many industrial applications. In a chemical engineering process, Kalman filters are frequently used to estimate unmeasurable process variables based on available measurements of other process variables, or to filter the measured process variables if they are noisy.

5.6.1. Mathematical Formulation

Consider a general linear system in the state space:

$$\boldsymbol{x}_{t+\Delta t} = \boldsymbol{A}_t \boldsymbol{x}_t + \boldsymbol{B}_t \boldsymbol{u}_t \tag{5.33}$$

The available observations are determined by:

$$\boldsymbol{y}_t = \boldsymbol{C}\boldsymbol{x}_t \tag{5.34}$$

The appropriate Kalman filter and recursive update for the covariance matrix ***P*** and Kalman gain $\boldsymbol{K}^{*}$ is given by:

$$\begin{aligned}
\boldsymbol{K}^{*}{}_{t} &= \boldsymbol{P}_{t}\boldsymbol{C}_{t}^{T}\left[\boldsymbol{C}_{t}\boldsymbol{P}_{t}\boldsymbol{C}_{t}^{T}+\boldsymbol{R}\right]^{-1}\\
\boldsymbol{x}_{t+\Delta t} &= \boldsymbol{A}_{t}\boldsymbol{x}_{t}+\boldsymbol{B}_{t}\boldsymbol{u}_{t}+\boldsymbol{K}^{*}{}_{t}\left[\hat{\boldsymbol{y}}_{t}-\boldsymbol{C}_{t}\boldsymbol{x}_{t}\right]\\
\boldsymbol{P}_{t+\Delta t} &= \boldsymbol{A}_{t}\left[\boldsymbol{I}-\boldsymbol{K}^{*}{}_{t}\boldsymbol{C}_{t}\right]\boldsymbol{P}_{t}\boldsymbol{A}_{t}^{T}+\boldsymbol{Q}
\end{aligned} \tag{5.35}$$

In the above representation, $\hat{\boldsymbol{y}}$ is defined as the actual observation available.

This filter can be cast in a form to find the coefficients ***k*** of the error regression problem by setting $\boldsymbol{x}\rightarrow\boldsymbol{k}$, $\boldsymbol{A}\rightarrow\boldsymbol{I}$, $\boldsymbol{B}\rightarrow\boldsymbol{0}$, $\hat{\boldsymbol{y}}\rightarrow\alpha$ and $\boldsymbol{C}\rightarrow\boldsymbol{\beta}^{T}$:

$$\begin{aligned}
\boldsymbol{K}^{*}{}_{t} &= \boldsymbol{P}_{t}\boldsymbol{\beta}_{t}\left[\boldsymbol{\beta}_{t}^{T}\boldsymbol{P}_{t}\boldsymbol{\beta}_{t}+\boldsymbol{R}\right]^{-1}\\
\boldsymbol{k}_{t+\Delta t} &= \boldsymbol{k}_{t}+\boldsymbol{K}^{*}{}_{t}\left[\alpha_{t}-\boldsymbol{\beta}_{t}^{T}\boldsymbol{x}_{t}\right]\\
\boldsymbol{P}_{t+\Delta t} &= \left[\boldsymbol{I}-\boldsymbol{K}^{*}{}_{t}\boldsymbol{\beta}_{t}^{T}\right]\boldsymbol{P}_{t}+\boldsymbol{Q}
\end{aligned} \tag{5.36}$$

One can recall that the matrix ***R***, which is usually diagonal represents the variance of errors to be expected in each measurement, the matrix ***Q***, also usually diagonal, represents the error variance to be expected in the state equations, which in this case merely express the expected constant nature of ***k***. The initial covariance matrix $\boldsymbol{P}_0$ is started off as a (small) diagonal.

As in the case of the moving window above, a special measure must be implemented to allow suppression of the movement of the k_i away from their expected values. This is done by extending the observations $\hat{\boldsymbol{y}}$ and the observation matrix ***C*** as follows:

$$\hat{\boldsymbol{y}} = \begin{pmatrix} \alpha_{\mathrm{t}} \\ \mathrm{k}_{1\mathrm{exp}} \\ \mathrm{k}_{2\mathrm{exp}} \\ \vdots \\ \mathrm{k}_{\mathrm{Nexp}} \\ \mathrm{k}_{\mathrm{N+1exp}} \end{pmatrix} \qquad \boldsymbol{C} = \begin{bmatrix} \beta_1 & \beta_2 & \cdots & \beta_{\mathrm{N}} & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \qquad \boldsymbol{k} = \begin{pmatrix} k_1 \\ k_2 \\ \vdots \\ k_N \\ k_{N+1} \end{pmatrix}$$

The updating equations now become:

$$\boldsymbol{K}^{*}{}_{t} = \boldsymbol{P}_t \boldsymbol{C}_t^{\boldsymbol{T}} \left[\boldsymbol{C}_t \boldsymbol{P}_t \boldsymbol{C}_t^{\boldsymbol{T}} + \boldsymbol{R} \right]^{-1}$$

$$\boldsymbol{k}_{t+\Delta t} = \boldsymbol{k}_t + \boldsymbol{K}^{*}{}_{t} \left[\hat{\boldsymbol{y}}_t - \boldsymbol{C}_t \boldsymbol{x}_t \right] \quad \textit{ie.} \quad \left\{ \boldsymbol{k}_{t+\Delta t} = \boldsymbol{k}_t + \boldsymbol{K}^{*}{}_{t} \left[\begin{pmatrix} \alpha_{\mathrm{t}} \\ \mathrm{k}_{1\mathrm{exp}} \\ \mathrm{k}_{2\mathrm{exp}} \\ \vdots \\ \mathrm{k}_{\mathrm{Nexp}} \\ \mathrm{k}_{\mathrm{N+1exp}} \end{pmatrix} - \begin{bmatrix} \beta_1 & \beta_2 & \cdots & \beta_{\mathrm{N}} & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \begin{pmatrix} k_1 \\ k_2 \\ \vdots \\ k_N \\ k_{N+1} \end{pmatrix} \right] \right.$$

$$\boldsymbol{P}_{t+\Delta t} = \left[\boldsymbol{I} - \boldsymbol{K}^{*}{}_{t} \boldsymbol{C}_t \right] \boldsymbol{P}_t + \boldsymbol{Q}$$

The filter attempts to close the gap between the two terms within the square bracket of the ***k*** updating equation. The rate at which it does this is controlled by the individual values on the diagonal of the ***R*** matrix. The smaller these are relative to those on the ***Q*** matrix diagonal, the more closely α and the expected *k* values will be tracked.

5.7. EXTENSIONS FOR INDUSTRIAL SITUATIONS

Correlation of inputs, set-point excitation and noise levels are common factors that need to be considered in MPM. Due to the feedback mechanism of MPC controllers, inputs are often correlated, for example, in a combustion process the flow of fuel and air flow are highly correlated. Correlation amongst inputs may also result in a false diagnosis of MPM (refer to section 4.3.1.2).

The underlying assumption in system identification and validation techniques is that the data is sufficiently rich. However, the primary goal of a controller, in closed-loop, is to reach a desired set-point. On the other hand the upper layer in a typical MPC operation regularly computes new set-point targets. For this reason, it can occasionally be assumed that enough setpoint excitation exists owing to these regular computations.

Noise levels and unmeasured disturbances are critical for the performance of the method. It is necessary that the magnitude of the modeling error dominates over these effects. If this is not the case, then it becomes difficult to determine MPM. Although this thesis focuses on the deterministic modeling errors in plant dynamics, characteristics relating to signal-to-noise ratio are developed.

5.7.1. Partial Correlations

In order to alleviate the effect of correlations amongst inputs, a technique which removes the influence of other variables on a specific variable is required. The method developed below focuses on the work of Badwe et al (2009). Suppose that u_1, u_2 ... u_m are inputs affecting the error, e, and suppose that the u_i's are correlated with each other. Then, to evaluate the partial correlation between, e and u_i, for example, e and u_1 are first linearly regressed on u_2, u_3... u_m:

$$\in_1 = e - \boldsymbol{\vartheta}_1^T \boldsymbol{u} \quad \text{where} \quad \boldsymbol{\vartheta}_1 = \begin{pmatrix} 0 \\ \psi_2 \\ \psi_3 \\ \vdots \\ \psi_m \end{pmatrix} \tag{5.37}$$

Similarly,

$$\in_2 = e - \boldsymbol{\vartheta}_2^T \boldsymbol{u} \quad \text{where} \quad \boldsymbol{\vartheta}_2 = \begin{pmatrix} \psi_1 \\ 0 \\ \psi_3 \\ \vdots \\ \psi_m \end{pmatrix} \tag{5.38}$$

$$\in_3 = e - \boldsymbol{\vartheta}_3^T \boldsymbol{u} \quad \text{where} \quad \boldsymbol{\vartheta}_2 = \begin{pmatrix} \psi_1 \\ \psi_2 \\ 0 \\ \vdots \\ \psi_m \end{pmatrix} \tag{5.39}$$

$$\vdots$$

$$\vdots$$

$$\in_m = e - \boldsymbol{\vartheta}_m^T \boldsymbol{u} \quad \text{where} \quad \boldsymbol{\vartheta}_{\mathrm{m}} = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \vdots \\ 0 \end{pmatrix} \tag{5.40}$$

Equations (5.35) to (5.38) remove the effect of other inputs on *e*. It is also imperative to remove the effect of other inputs on a specific input:

$$\boldsymbol{\delta}_1 = u_1 - \boldsymbol{\phi}_1^T \boldsymbol{u} \quad \text{where} \quad \boldsymbol{\phi}_1 = \begin{pmatrix} 0 \\ \xi_2 \\ \xi_3 \\ \vdots \\ \xi_m \end{pmatrix} \tag{5.41}$$

$$\boldsymbol{\delta}_2 = u_2 - \boldsymbol{\phi}_2^T \boldsymbol{u} \quad \text{where} \quad \boldsymbol{\phi}_2 = \begin{pmatrix} \xi_1 \\ 0 \\ \xi_3 \\ \vdots \\ \xi_m \end{pmatrix} \tag{5.42}$$

$$\delta_3 = u_3 - \boldsymbol{\phi}_3^T \boldsymbol{u} \quad \text{where} \quad \boldsymbol{\phi}_3 = \begin{pmatrix} \xi_1 \\ \xi_2 \\ 0 \\ \vdots \\ \xi_m \end{pmatrix} \tag{5.43}$$

$$\delta_m = u_m - \boldsymbol{\phi}_m^T \boldsymbol{u} \quad \text{where} \quad \boldsymbol{\phi}_{\mathrm{m}} = \begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \vdots \\ 0 \end{pmatrix} \tag{5.44}$$

Equations (5.39) to (5.42) remove the effect of the other inputs on u_{i}. A new equation is defined which relates the two residual errors (ϵ_{i} and δ_{i}) with the influence of other variables removed:

$$\mathrm{E}_1 = \in_1 - \rho_1 \delta_1 \tag{5.45}$$

$$\mathrm{E}_2 = \in_2 - \rho_2 \delta_2 \tag{5.46}$$

$$\mathrm{E}_3 = \in_3 - \rho_3 \delta_3 \tag{5.47}$$

$$\vdots$$

$$\mathrm{E}_m = \in_m - \rho_m \delta_m \tag{5.48}$$

The partial correlation coefficient, ρ_{i}, is found by minimizing E_{i}:

$$\mathrm{E}_i = \in_i - \rho_i \delta_i \tag{5.49}$$

The objective function is defined as:

$$J(\rho_i) = \sum_{j=1}^{M} \left\{ \left[e - \vartheta_i^T u \right]_j - \rho_i \left[u_i - \phi_i^T u \right]_j \right\}^2 \tag{5.50}$$

$$J(\rho_i) = \sum_{j=1}^{M} \left\{ (\in_i)_j - \rho_i (\delta_i)_j \right\}^2 \tag{5.51}$$

Minimizing equation (5.49):

$$\frac{\partial J}{\partial \rho_i} = \sum_{j=1}^{M} -2\left(\in_{i_j} - \rho_i \delta_{i_j} \right) \delta_{i_j} = 0$$

Rearranging for ρ_{i}:

$$\rho_i = \frac{\sum_{j=1}^{M} \in_{i_j} \delta_{i_j}}{\sum_{j=1}^{M} \delta_{i_j} \delta_{i_j}} \tag{5.52}$$

The summations above are computed efficiently in the same way as the covariance matrix summation and the scalar product summation in the moving window recursive regression (see section 5.4).

5.7.2. Set-point excitation

In a limited way one could use the regressed offset (non-differential forms) to detect MPM and relate it to an input when the variability goes ‘quiet’, for example equation (5.24) can be used in this case. This concept can be illustrated graphically:

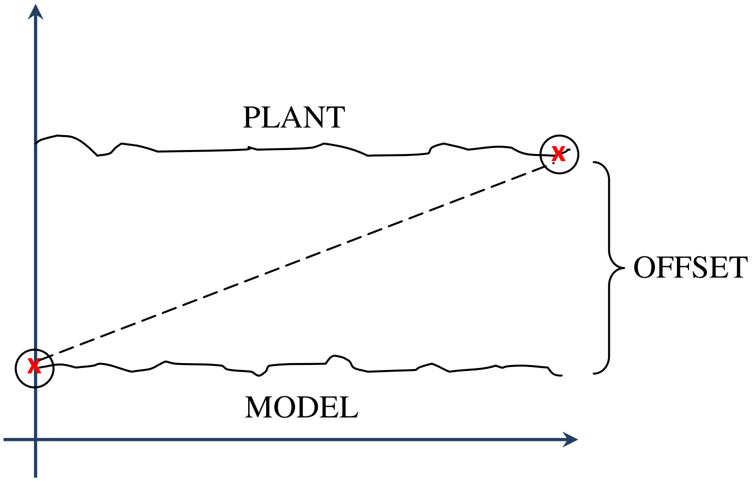


Figure 5.7: Lack of informative data

Coupled with an offset, the model and plant outputs allow for 2 points denoted by (x) in the figure above. The slope found between the 2 points is related to the fitting parameters k_i. For example, the model may expect 20m^{3}h^{-1} through a certain valve which is 50% open, but only 15m^{3}h^{-1} is being measured. So a linear gain error will be determined on the valve using the offset.

5.7.3. Noise Levels

The effect of the noise on a system is determined by a signal-to-noise ratio. Signal-to-noise ratio (SNR or S/N) is defined as a measure employed in science and engineering to quantify the impact that noise has on a signal. The signal in this case is the plant output. A ratio much higher than 1:1 indicates more signal than noise which is desired. For this thesis, the following definition of SNR is used:

$$\mathrm{SNR} = \frac{\sigma_{signal}}{\sigma_{noise}} \tag{5.53}$$

Numerous other ratios can be defined which will equally describe the effect of noise on a signal. Among the most desirable forms of ratios is one relating the standard deviation (or variance) of the output error to the standard deviation (or variance) exhibited by the noise:

$$\mathrm{SNR}_e = \frac{\sigma_{error}}{\sigma_{noise}} \tag{5.54}$$

In the case of no MPM present, this ratio will exhibit a 1:1 relationship according to equation (4.2). A ratio greater than 1:1 implies MPM. Conversely, a ratio close to or less than 1:1 implies that the error formed is principally due to the influence of noise rather than any MPM in the dynamic system.

In cases where the variance of the noise signal cannot be obtained (as it is already embedded in the plant output signal), the following ratio may be employed:

$$\mathrm{SNR}_y = \frac{\sigma_{y_p}}{\sigma_{y_m}} \tag{5.55}$$

This ratio, defined as the variance of the plant output to the model output can be used to determine the presence of noise. If the ratio is greater than 1:1 then can be deduced that the system is influenced by the presence of a noise signal.

CHAPTER 6 EXPERIMENTAL DESIGN THEORY

This chapter provides insight into the manner in which the theory developed in the previous chapter is implemented in a software environment. The software used in this thesis is MATLAB ®. Indeed, MATLAB ® is known for its efficiency concerning matrix manipulations which are required for computational purposes here. Prior to the implementation of the methods developed, several factors need to be addressed. These include the choice of sampling interval for simulations, discrete approximation choice, input data representation (closed-loop forms), output evaluation and data preparation. One should note that in the case of testing real industrial plant data the sampling interval is used according to the intervals in which the data is provided. Once these factors are dealt with, the manner in which the methods are implemented in MATLAB ® subsequently follows.

6.1. PROGRAM

MATLAB ® software was chosen for the implementation of the methodology developed in the previous chapter. MATLAB ® was developed by MathWorks and is highly favored for mathematical computations which require stringent matrix manipulations. MATLAB ® allows for the plotting of functions and data, implementation of graphical user inferfaces and is easily adapted into other programming languages such as C. C++ and FORTRAN. Figure 1.2 (see section 1.4) illustrates a simplified flow diagram of the program. The related function files and main program descriptions can be found in Appendix B and all the programs written and used in this thesis are provided on the attached CD. The code is developed generically for any number of inputs and measured disturbances. Due to the fact that model matrix structure in which SASOL represents its models is in an input-output form, each output can be considered separately within the program.

6.2. PRELIMINARIES

Prior to the implementation of the methodology developed in chapter 5, factors such as sampling interval, discretization properties, input representation and output evaluation need to be addressed.

6.2.1. Sampling Interval

The choice of a data sampling interval, *T*, is critical in the retention of vital information that describes the systems important dynamics. A *T* that is much greater than the resulting time constants of a particular system would then defer data with very little information regarding the plant dynamics. A small *T*, on the other hand, would be inefficient because insufficient process change would occur between samples. A good choice of *T* should thus be a trade-off between noise reduction and relevance for the dynamics. For a MISO system, the optimal choice of the sampling interval lies in the region of the shortest time constant of the system.

Recall equation 5.1:

$$y(s) = K_1\, e^{-\tau_1 s}\, \frac{1 + c_{11}s + c_{12}s^2 + ...}{1 + d_{11}s + d_{12}s^2 + ...} u_1(s) + K_2 e^{-\tau_2 s}\, \frac{1 + c_{21}s + c_{22}s^2 + ...}{1 + d_{21}s + d_{22}s^2 + ...} u_2(s) + ...$$

It is often convenient to factor the polynomials in the numerator and denominator, and to write the transfer function in terms of those factors:

$$y(s) = K_1\, e^{-\tau_1 s}\, \frac{(\tau_{1c1}s + 1)(\tau_{1c2}s + 1)...(\tau_{1M_c}s + 1)}{(\tau_{1d1}s + 1)(\tau_{1d2}s + 1)...(\tau_{1M_d}s + 1)} u_1(s) + ...$$

$$y(s) = \sum_{i=1}^{N} K_i e^{-\tau_i s} \frac{\prod_{j=1}^{M_c} (\tau_{ij}s + 1)}{\prod_{j=1}^{M_d} (\tau_{ij}s + 1)} u_i(s) \tag{6.1}$$

The numerator represents the zeros of the system and the denominator represents the poles. Setting the denominator to zero gives the time constants of the system as shown below:

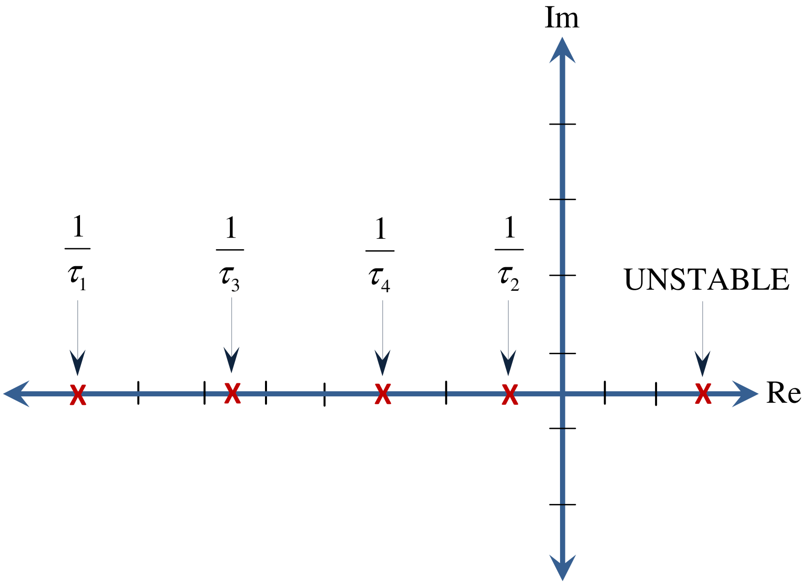


Figure 6.1: conceptual view of the poles of a system

For a MISO system the desirable sampling interval is related to the fastest acting time constant i.e. the most negative pole. For example, the fastest time constant in figure 6.1 is given as τ_1. Poles that are positive imply that the system is unstable. In the case of an integrator where the pole is zero, the time constant is corrected to a value close to zero. Figure 6.2 illustrates the manner in which the discretization error changes with the choice of the sampling interval:

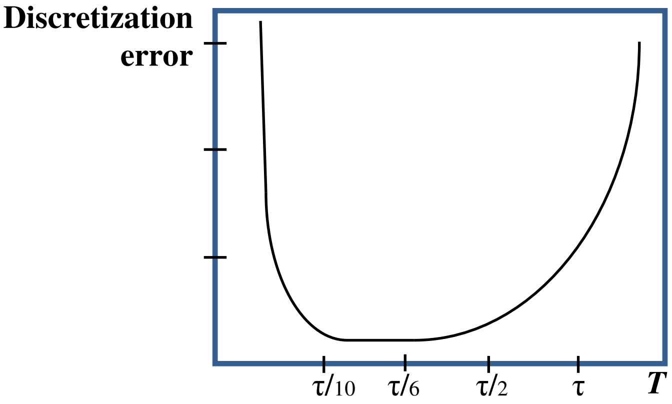


Figure 6.2: Discretization error as a function of sampling interval adapted from Ljung (1989)

Based on the observations of figure 6.2 above, a sampling time close to that at which 1/10 of the full response is achieved, as indicated in Fig. 6.3, was found to be satisfactory:

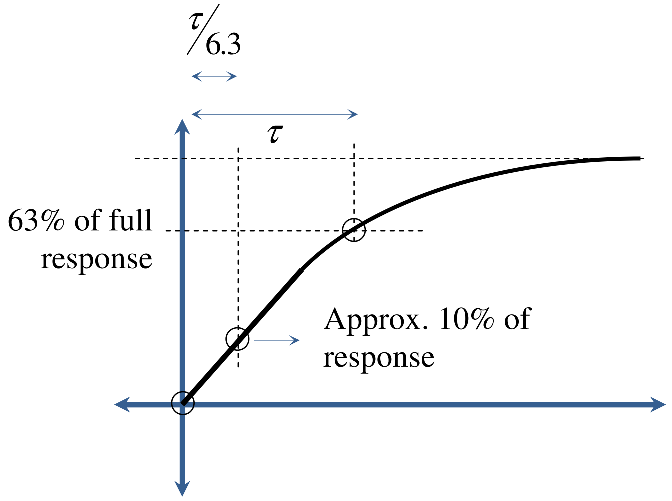


Figure 6.3: Satisfactory sampling interval

$$T_{opt} = \frac{\tau}{6.3} \tag{6.2}$$

From figure 6.3 above, it can be seen that the suitable sampling interval is based on $^{1}/_{10}$ of the slope through the 63% point of a first order step response.

6.2.2. Discretization

The commonly used transformations for the Laplace variable, *s* are given in table 5.1. In this work, each approximation is analyzed for stability in the transformation as well as reduction in discretization errors. Stability implies lying to the left in the *s*-plane or correspondingly within a unit circle in the *q*-plane.

6.2.2.1. Forward Difference Approximation

Figure 6.4 below illustrates the mapping of the left-half *s*-plane for the forward difference approximation onto the region depicted for the *q*-plane. This maps more than the unit circle. This implies that a transfer function, *G*(*s*), with high frequency or lightly damped poles will give a discrete transfer function, *G*(*q*) which is unstable.

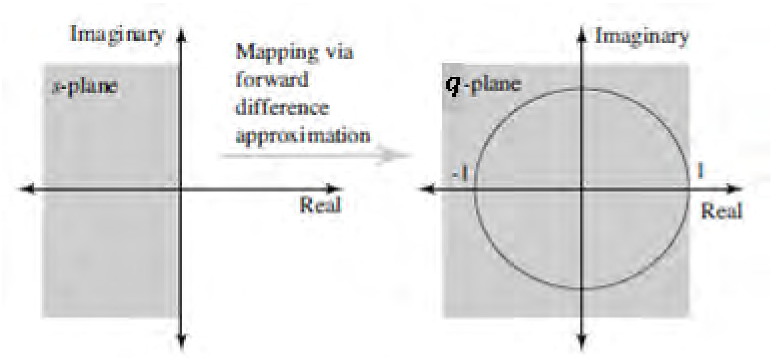


Figure 6.4: Forward difference mapping (Sekara, 2005)

6.2.2.2. Backward Difference Approximation

This mapping (figure 6.5) is shown to be inside the unit circle. Thus a stable *G*(*s*) implies a stable *G*(*q*). *G*(*q*) cannot have lightly damped poles even if *G*(*s*) has lightly damped poles.

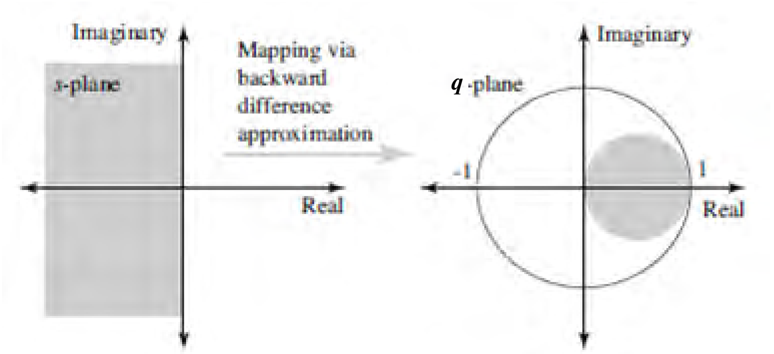


Figure 6.5: Backward difference mapping (Sekara, 2005)

6.2.2.3. Bilinear (TUSTIN) Approximation

This approximation which is shown in figure 6.6 below maps the entire left-half plane onto the unit circle. The implication here is that *G*(*q*) is stable for any stable *G*(*s*). This is why the Tustin approximation is the most commonly used discrete time transformation. Furthermore this approximation allows the reduction of discretization error compared to the other approximation methods. The Tustin approximation was used to convert equation (5.1) into a real-time form.

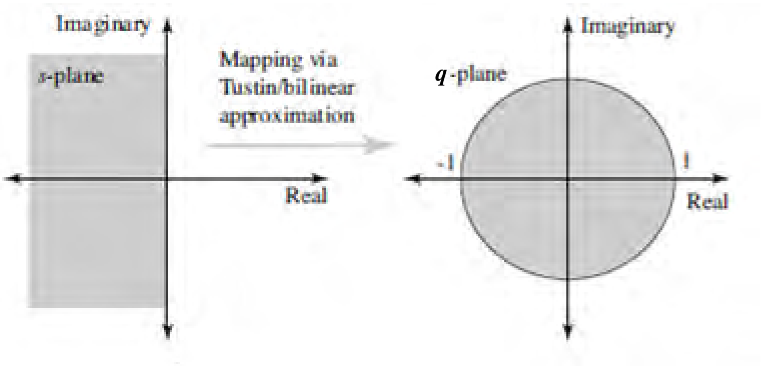


Figure 6.6: Bilinear mapping (Sekara, 2005)

6.2.3. Simulated Input Design

At this point it is desirable to present some input sequences that are encountered in industrial processes.

6.2.3.1. Random (Gaussian) signal

Gaussian signals are white noise signals with Gaussian distributions. These signals are favorable for linear modeling and sometimes occur under closed-loop conditions. This signal is illustrated below for a case where time-correlation has been applied to the signal (smoothing):

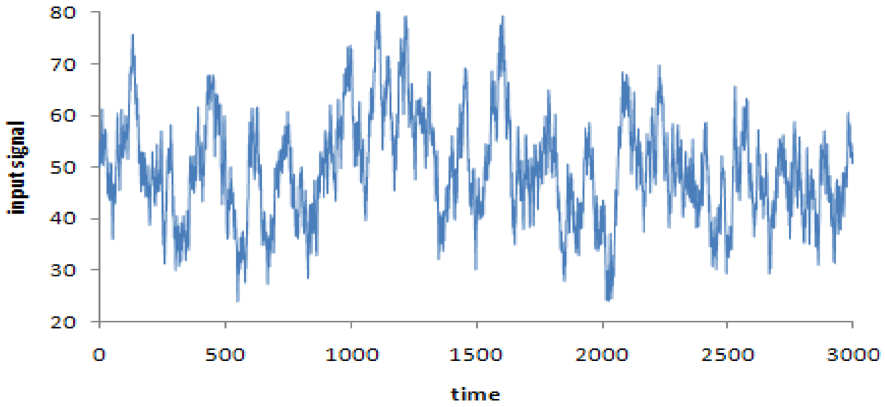


Figure 6.7: Gaussian signal

6.2.3.2. Pseudo-Random Binary Signal (PRBS)

These signals are subjected to upper and lower limit constraints. Mathematically this is given by:

$$u(t) = u_1 \;\; or \;\; u_2 \tag{6.3}$$

N.B. u_1 and u_2 represent the limits placed on the input signal. The signal is called binary as it only switches between two values and pseudo-random because the actual sequence is artificially created:

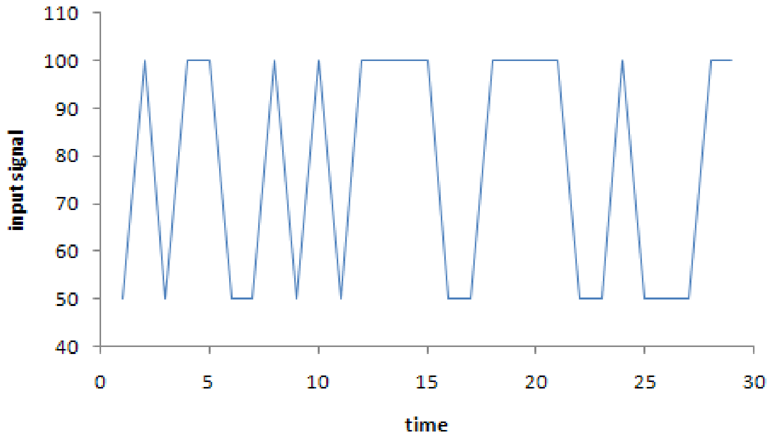


Figure 6.8: PRBS signal

6.2.3.3. Multi-level Signal

Often encountered in industrial processes is a combination of Gaussian and PRBS signals. In this case the signal amplitudes randomly vary in the range between the binary signal values:

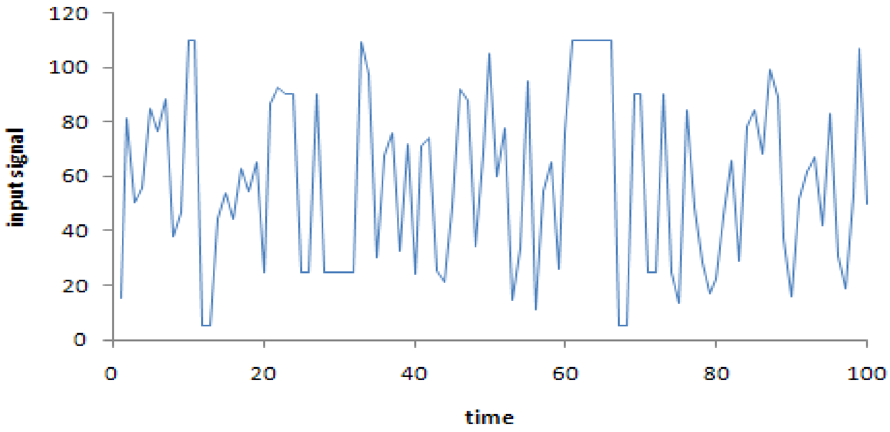


Figure 6.9: Multilevel signal

Signals behave in this manner because Model Predictive Controllers compute the set-points to the regulatory layers recurrently. For simulations, a random sequence with realistic time-correlation from point to point is generated in the following manner:

INPUTS GENERATED IN AN ONLINE MANNER

```
% N_records - recorded time period set by user
for it=1:N_records
    % sampling time interval
    t=it*dt;
    % Nterm - NUMBER OF INPUTS
    for k=1:Nterm

        if (it==1)
            % initially set to the mean input
            ubar(k)=umean(k);
        else
            % random data generated in a range specified by the user
            % the equation below represents a smoothing filter:
            % ufactor - smoothing coefficient
            % umean - mean of each input
            % urange - range of input
            u1(k)=ufactor(k)*umean(k)+ (1-ufactor(k))...
                  *(umean(k)+urange(k)*(rand()-0.5));
            % double filter now
            ubar(k)=ufactor(k)*ubar(k)+(1-ufactor(k))*u1(k);
        end

    end

    for k=1:Nterm
        u(k)=ubar(k);
    end

    % stored for analysis
    u_PLOT(:,it)=u';
.
.

end
```

It should be noted that all MATLAB ® code presented from this stage forward are all found within a single time-loop and only relevant portions are shown to explain certain concepts.

6.2.4. Simulated Output Design

Recall equations (5.12) and (5.13)

$$\left\{\prod_{i=1}^{N} A_i(q^{-1})\right\} y_m(t) \ = \sum_{i=1}^{N}\left[q^{-n_i} B_i(q^{-1}) \left\{ \prod_{\substack{j=1 \\ j\neq i}}^{N} A_j(q^{-1}) \right\} \right] u_i(t)$$

$$y_m(t) \ = \left[1 - \left\{\prod_{i=1}^{N} A_i(q^{-1})\right\} \right] y_m(t) \ + \ \sum_{i=1}^{N}\left[q^{-n_i} B_i(q^{-1}) \left\{ \prod_{\substack{j=1 \\ j\neq i}}^{N} A_j(q^{-1}) \right\} \right] u_i(t)$$

In order to simplify the notation, the following modifications are made to the equations above:

$$\overline{C}(q^{-1}) = \left\{\prod_{i=1}^{N} A_i(q^{-1})\right\} = 1 + c_1(q^{-1}) + c_2(q^{-2}) + c_3(q^{-3}) + ... + c_{\max}(q^{-\max}) \tag{6.4}$$

$$\overline{D}_i(q^{-1}) = B_i(q^{-1}) \left\{ \prod_{\substack{j=1 \\ j\neq i}}^{N} A_j(q^{-1}) \right\} = d_{i1}(q^{-1}) + d_{i2}(q^{-1}) + ... + d_{i\max}(q^{-\max}) \tag{6.5}$$

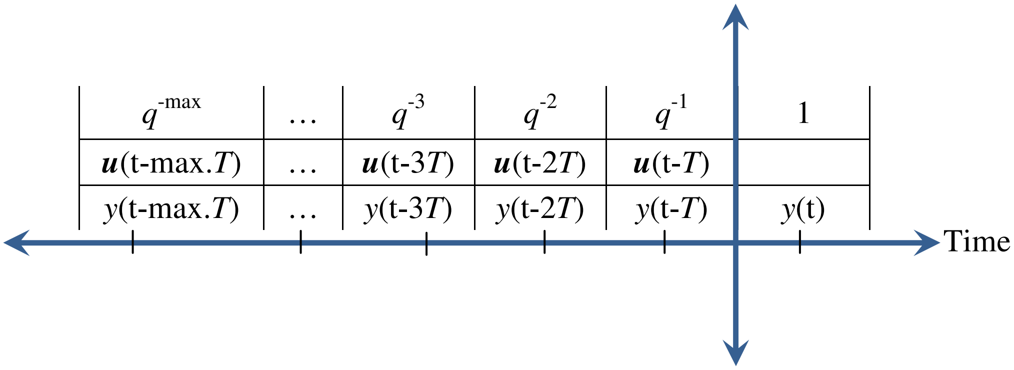
N.B. max is defined as: *N* x maximum [M_c, M_d].

Equations (5.12) and (5.13) can now be re-written as:

$$\overline{C}(q^{-1}) y_m \ \ (t) \ = \sum_{i=1}^{N}\left[q^{-n_i} \overline{D}_i(q^{-1}) \right] \ u_i(t) \tag{6.6}$$

$$y_m(t) \ = \left[1 - \overline{C}(q^{-1}) \right] \ y_m(t) \ + \ \sum_{i=1}^{N}\left[q^{-n_i} \overline{D}_i(q^{-1}) \right] \ u_i(t) \tag{6.7}$$

The model output is evaluated using equations (6.6) and (6.7) depending on the type of error chosen i.e. equation error or output error. It is reiterated that these equations depend on past outputs and inputs in order to evaluate the current output. The new defined polynomials, equations (6.4) and (6.5), show that the lag coefficients are arranged from the highest power to the lowest power. This means that the previous contributions of both inputs and the output need to arranged in a form to match these coefficients. This is accomplished by the development of ‘stacks’:



q^{-max}	…	q^{-3}	q^{-2}	q^{-1}	1
$\boldsymbol{u}$(t-max.T)	…	$\boldsymbol{u}$(t-3T)	$\boldsymbol{u}$(t-2T)	$\boldsymbol{u}$(t-T)	
y(t-max.T)	…	y(t-3T)	y(t-2T)	y(t-T)	y(t)

Figure 6.10: Illustration of ‘stacks’ for past data to match the lag coefficients

Figure 6.10 above illustrates the manner in which the model output is calculated. The implementation of data ‘stacks’ requires 2 stages: updating ‘stacks’ and thereafter manipulating the updated ‘stacks’ to form ordered ‘stacks’ as required:

UPDATING STACKS

```
% 1. Input stack
      if (ipu<1)
          for jj=1:MaxPower*Nterm
              for k=1:Nterm

      ustack(k,jj)=u(k);
              end
          end
          ipu=1;
      else
          ipu=ipu+1;
          if (ipu>MaxPower*Nterm)
              ipu=1;
          end
          for k=1:Nterm
              ustack(k,ipu)=u(k);
          end
      end
```

```
% 2. Output stack
      if (ipy<1)
          for
      ii=1:MaxPower*Nterm
              ystack_m(ii)=y_m;
              ystack_p(ii)=y_p;
          end
          ipy=1;
      else
          ipy=ipy+1;
          if
      (ipy>MaxPower*Nterm)
              ipy=1;
          end
          ystack_m(ipy)=y_m;
          ystack_p(ipy)=y_p;
      end
```

ORDERED STACKS

```
% 1. Ordered output stack
for ii=1:MaxPower*Nterm
    ip=ipy-ii+1;
    % must go backwards
    % to have older values at
end
    if (ip<1)
        ip=ip+MaxPower*Nterm;
    end

y_ordered_p(ii)=ystack_p(ip);

    if (use_plant_stack)
        % ONE-STEP PREDICTOR
        % equation (5.14)

y_ordered_m(ii)=ystack_p(ip);
     else
        % OPEN-LOOP PREDICTOR
        % equation (5.13)

y_ordered_m(ii)=ystack_m(ip);
    end
end
```

```
% 2. Ordered input stack

for jj=1:MaxPower*Nterm
    ip=ipu-jj+1;
    if (ip<1)
        ip=ip+MaxPower*Nterm;
    end
    for k=1:Nterm

u_ordered(k,jj)=ustack(k,ip);
    end
end
```

For simulations of a real-time MPM detector, both the model output and plant output are evaluated using equation (6.7). The plant is the true representation of the system and the program is tested for MPM by inserting a mismatch in the model.

6.2.5. Industrial Data Design

A vital component of testing the validity of a model validation tool is to use real plant data. For this thesis, the mode in which real plant data is fed into the program is analogous to the manner in which data would be received from the plant in real time. The simulated data shown above is seen to be generated at every time step. Conversely, industrial data is stored in a data file and read into the program as follows:

INDUSTRIAL DATA

```
for it=1:N_records

    if (it==1)
        %... open file: example of file
        fid = fopen('plant_CV2.dat');
    end

    % Nout - number of outputs (set to 1 for MISO systems)
    for i=1:(Nout+Nterm)
        % read one-at a time ie. size = 1
        dum = fscanf(fid,'%g',1);
        % select only the output of interest here
        if (i==j_out)
            % stored temporarily
            y_PLANT=dum;
        end

        if (i>Nout)
            % the rest are the inputs (including dvs)
            u(i-Nout)=dum;
        end
    end

    if (it==1)
        % initialising
        y_p=y_PLANT;
        y_m=y_p;
    end
.
.
.
end
```

The sampling interval used for the testing of industrial data is 0.5 min, the sampling rate of the plant data supplied by SASOL. This interval should lie within 0.5x to 2x the optimal interval determined by equation 6.2.

6.2.6. Error Definitions

The choice of error definition depends on the type of model (presence of integrator terms, or higher order models), offset between model and plant outputs as well as the variability of the data (most validation tools cannot extract any information from data that is ‘quiet’).

Six factors contribute to the choice of error definitions to be used for the detection of model degradation. Each of these factors is assigned a binary logic value to depict each choice:

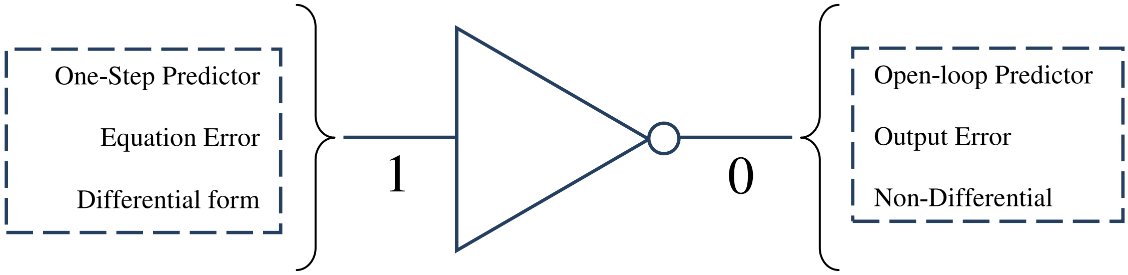


Figure 6.11: Graphical Binary logic inverse symbol

The table below demonstrates the several possible combinations of the six factors stated above:

Table 6.1:
Combinations of Predictor, error and difference form

	COMBINATIONS							
	A	B	C	**D**	E	F	**G**	H
Predictor form	**0**	1	1	**1**	0	0	**1**	**0**
Error form	**0**	0	1	**1**	1	0	**0**	**1**
Difference form	**0**	0	0	**1**	1	1	**1**	**0**

One can recall that the one-step predictor is useful in reducing the offset between the model and plant and allows the model to track the plant more closely. When the variability of data is high, one can use the output error form as the data provides sufficient information. For a more stable regression however, the equation error form should be used. It’s also beneficial in that it considers previous lag contributions. The difference form is particularly useful in eliminating the offset between the model and the plant. It also prevents integrating terms from producing excessively large outputs. Combinations D and G are suitable for cases where data is sufficiently informative. In the case where data is ‘quiet’ one may opt for either A or H in trying to detect MPM in a limited manner. Difference forms are handled in the following means:

DIFFERENCE FORMS

```
% Input difference form
% Deal with difference forms
if (it==1)
    du=0*u;
    ulast=u;
else
    du=u-ulast;
    ulast=u;
end
```

```
% Error difference form
% Deal with difference forms
if (it==1)
    dee=0;
    elast=ee;
else
    dee=ee-elast;
    elast=ee;
end
```

The extract below exemplifies the approach in which the equation error form is put together:

EQUATION ERROR FORM

```
% model-weighted yp, ym & u
yp_LHS_mw=y_p+sum(c_m.*y_ordered_p(1:MaxPower*Nterm)); % PLANT
ym_LHS_mw=y_m+sum(c_m.*y_ordered_m(1:MaxPower*Nterm)); % MODEL

% equation error
e_LHS_mw=-(yp_LHS_mw-ym_LHS_mw);

% RIGHT HAND SIDE OF EQUATION (6.7)
for k=1:Nterm
    u_RHS_mw(k)=sum(d_m(k,:).*u_ordered(k,1:MaxPower*Nterm));
end
```

The ability to provide a more stable regression without any influence of an offset is useful to handle numerous generic cases. This is provided by the combination of an equation error difference form:

EQUATION ERROR DIFFERENCE FORM

```
if (it==1)
    de_LHS_mw=0;
    e_LHS_mw_last=e_LHS_mw;
    for k=1:Nterm
        du_RHS_mw(k)=0;
        u_RHS_mw_last(k)=u_RHS_mw(k);
    end
else
    % equation error difference form
    de_LHS_mw=e_LHS_mw-e_LHS_mw_last;
    e_LHS_mw_last=e_LHS_mw;
    % equation error difference form
    for k=1:Nterm
        du_RHS_mw(k)=u_RHS_mw(k)-u_RHS_mw_last(k);
        u_RHS_mw_last(k)=u_RHS_mw(k);
    end
end
```

6.3. DATA PREPARATION

Computational effort for correlation and regression analyses depends on how the covariance summation in the correlation equation and the summations in the least squares objective function are handled respectively. Both methodologies require a range of data. Computation is made efficient by the development of a ‘moving window’ of data. This window moves forward at each time step taking in the new data point and removing the old data point (see figure 5.6). Depending on the combination chosen as shown in table 6.1, the moving window is developed as follows:

MOVING WINDOW CYCLIC STORAGE

```
p=p+1;
% increment pointer
if (p>window)
    p=1;
end
% save before over-written
ew_lost=ew(p);
% _autocorr - notation used for correlation analysis
% e = ym - yp; used for correlations
```

```
ew_autocorr_lost=ew_autocorr(p);
for k=1:Nterm
    % save before over-written
    uw_lost(k)=uw(p,k);
    % save before over-written
    rel_uw_lost(k)=SSgain_m(k)*uw(p,k);
    uw_autocorr_lost(k)=uw_autocorr(p,k);
end

if (differential_basis)
    if (use_LHS_RHS)
        % differential + equation error
        ew(p)=de_LHS_mw;
        % dee = e - e_last
        ew_autocorr(p)=dee;
        for k=1:Nterm
            uw(p,k)=du_RHS_mw(k);
            uw_autocorr(p,k)=du(k);
            rel_uw(p,k)=SSgain_m(k)*du_RHS_mw(k);
            % SSgain_m -> 1 for "use_LHS_RHS"
        end
    else
        ew(p)=dee;
        ew_autocorr(p)=dee;
        for k=1:Nterm
            uw(p,k)=du(k);
            uw_autocorr(p,k)=du(k);
            rel_uw(p,k)=SSgain_m(k)*du(k);
        end
   end
else
    % non-differential + equation error
    if (use_LHS_RHS)
        ew(p)=e_LHS_mw;
        ew_autocorr(p)=ee;
        for k=1:Nterm
            uw(p,k)=u_RHS_mw(k);
            uw_autocorr(p,k)=u(k);
            rel_uw(p,k)=SSgain_m(k)*u_RHS_mw(k);
            % SSgain_m -> 1 for "use_LHS_RHS"
        end
   % non-differential + output error
   else
       ew(p)=ee;
       ew_autocorr(p)=ee;
       for k=1:Nterm
           uw(p,k)=u(k);
           uw_autocorr(p,k)=u(k);
           rel_uw(p,k)=SSgain_m(k)*u(k);
       end
   end
end
```

6.4. Methods for Detecting MPM

The methods developed as shown in chapter 4 are based on residual analysis. Correlation analyses (correlations between residuals and inputs and correlations between inputs themselves) serve to provide a qualitative view of any discrepancies in a model(s). Regression analyses provide a more detailed view of MPM by giving a quantitative measure of the extent of the mismatch. For these methods the same notation is used for differential and non-differential modes.

6.4.1. Correlation Analyses

The sample cross-covariance between *e* and u_i at lag *m* are calculated as:

$$c_{eu_i}(m) = \frac{1}{M}\sum_{t=1}^{m}(e_t - \bar{e})(u_{it-m} - \bar{u}_i) \qquad (6.8)$$

Computation of equation (6.8) is made efficient by employing the moving window concept:

CROSS-CORRELATION (e – u)

```
for j=1:R_range % range of correlation plot (lag = M)
    if (j==1)
    % covariance summation
    sum_eu(j,k)=sum_eu(j,k)-ew_autocorr_lost*uw_autocorr_lost(k)...
                +ewg(window-R_range+1)*uwg(window-R_range+1,k);
    else
    % covariance summation
    sum_eu(j,k)=sum_eu(j,k)-ewg(j-1)*uw_autocorr_lost(k)...
               +ewg(window-R_range+j)*uwg(window-R_range+1,k);
    end
end
```

Equation (6.8) is adapted for the covariance between u_i and u_j:

$$c_{u_iu_j}(m) = \frac{1}{M}\sum_{t=1}^{m}(u_{i,t} - \bar{u}_i)(u_{j,t-m} - \bar{u}_j) \qquad (6.9)$$

The covariance summation is computed in a similar manner as above:

CROSS-CORRELATION ($u_i – u_j$)

```
for j=1:R_range

   if (j==1)

   sum_uu(j,k,kk)=sum_uu(j,k,kk)...
              -uw_autocorr_lost(k)*uw_autocorr_lost(kk)...
              +uwg(window-R_range+1,k)*uwg(window-R_range+1,kk);


   else


   sum_uu(j,k,kk)=sum_uu(j,k,kk)...
              -uwg(j-1,k)*uw_autocorr_lost(kk)...
              +uwg(window-R_range+j,k)*uwg(window-R_range+1,kk);


   end
end
```

The cross covariance may be normalized by the individual variances to become the sample cross-correlation:

$$R_{eu_i}(m) = \frac{c_{eu_i}(m)}{\sqrt{\sigma_e^{\ 2}\sigma_{u_i}^{\ 2}}} \tag{6.10}$$

$$R_{u_iu_j}(m) = \frac{c_{u_iu_j}(m)}{\sqrt{\sigma_{u_i}^{\ 2}\sigma_{u_j}^{\ 2}}} \tag{6.11}$$

If an error exists, a correlation analysis between the error and the corresponding inputs would reveal that portion of the model causing the error, by means of a significant correlation. At certain time intervals graphical cross-correlation results are displayed.

From these plots, one can check for any detection of MPM as well as correlations amongst inputs which may compound any error that may exist.

6.4.2. Moving Window Regression

The recursive regression based on the moving window approach puts an equal weighting on each point within the window, thus the response time to reach the intended fitting parameter is longer but smoother:

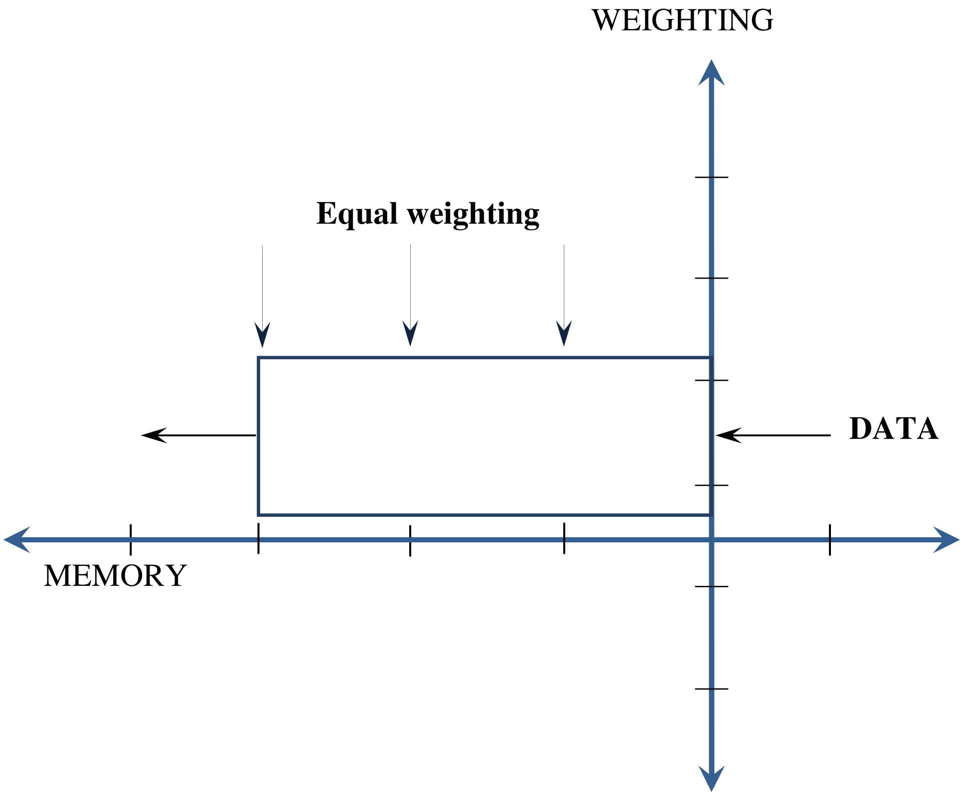


Figure 6.12: Concept of weighting placed on window data for MWR

Recall the covariance matrix and the scalar product terms found in equation (5.32):

$$\text{cov ariance matrix} = \left\{ \sum_{\mathrm{t=1}}^{\mathrm{M}} \boldsymbol{\beta}_{\mathrm{t}} \boldsymbol{\beta}_{\mathrm{t}}^{\mathrm{T}} \right\} \qquad \text{scalar vector product} = \left\{ \sum_{\mathrm{t=1}}^{\mathrm{M}} \alpha_{\mathrm{t}} \boldsymbol{\beta}_{\mathrm{t}} \right\}$$

These terms are computed as shown below in an efficient manner. N.B. In this form of regression, the offset term can be searched for or it can be fixed.

VARIABLE OFFSET MWR

```
% W_C_varo defined as (1:Nterm+1)
% regression searches for an offset (Nterm+1) term
W_C_varo(1:Nterm)=rel_uw(p,:);
W_C_lost_varo(1:Nterm)=rel_uw_lost(:)';

% COVARIANCE MATRIX
W_M_varo=W_M_varo+W_C_varo'*W_C_varo-W_C_lost_varo'*W_C_lost_varo;

% SCALAR PRODUCT
W_EC_varo=W_EC_varo+ew(p)*W_C_varo'-ew_lost*W_C_lost_varo';

if (rank(W_M_varo+Nlamda_VARO)==size(W_M_varo,1))
% FITTING PARAMETER ki
W_param_varo=(W_M_varo+Nlamda_VARO)\(W_EC_varo+Nlamda_kex_VARO);
else
W_param_varo=zeros(Nterm+1,1);
end
% variance of error
sigma_e_varo=sqrt(W_param_varo'*W_M_varo*W_param_varo/window);
```

FIXED OFFSET MWR

```
% W_C_fixo defined as (1:Nterm)
% offset is fixed to a certain point
W_C_fixo=rel_uw(p,:);
W_C_lost_fixo=rel_uw_lost(:)';

% COVARIANCE MATRIX
W_M_fixo=W_M_fixo+W_C_fixo'*W_C_fixo-W_C_lost_fixo'*W_C_lost_fixo;

% SCALAR PRODUCT
W_EC_fixo=W_EC_fixo+ew(p)*W_C_fixo'-ew_lost*W_C_lost_fixo';
if (rank(W_M_fixo+Nlamda_FIXO)==size(W_M_fixo,1))
% FITTING PARAMETER ki
W_param_fixo=(W_M_fixo+Nlamda_FIXO)\(W_EC_fixo+Nlamda_kex_FIXO);
else
W_param_fixo=zeros(Nterm,1);
end
sigma_e_fixo=sqrt(W_param_fixo'*W_M_fixo*W_param_fixo/window);
```

6.4.3. Kalman Filter

The Kalman filter depends largely on more recent values. The rate at which the expected fitting parameters are reached depends on the values along the diagonal of the ***R*** matrix (see section 5.6.1). The ***Q*** matrix (error covariance-model error) provides an indication of the weighting placed on past values that are used to predict current values and is usually diagonal. Large values along the diagonal of the ***Q*** matrix cause a quicker loss of the effect of older measurements; the converse is applicable for smaller values. This matrix can be adjusted depending on the information content of the data. For example, if the operating point suddenly moves to a new location, the older data probably are less relevant and therefore should be 'forgotten'.

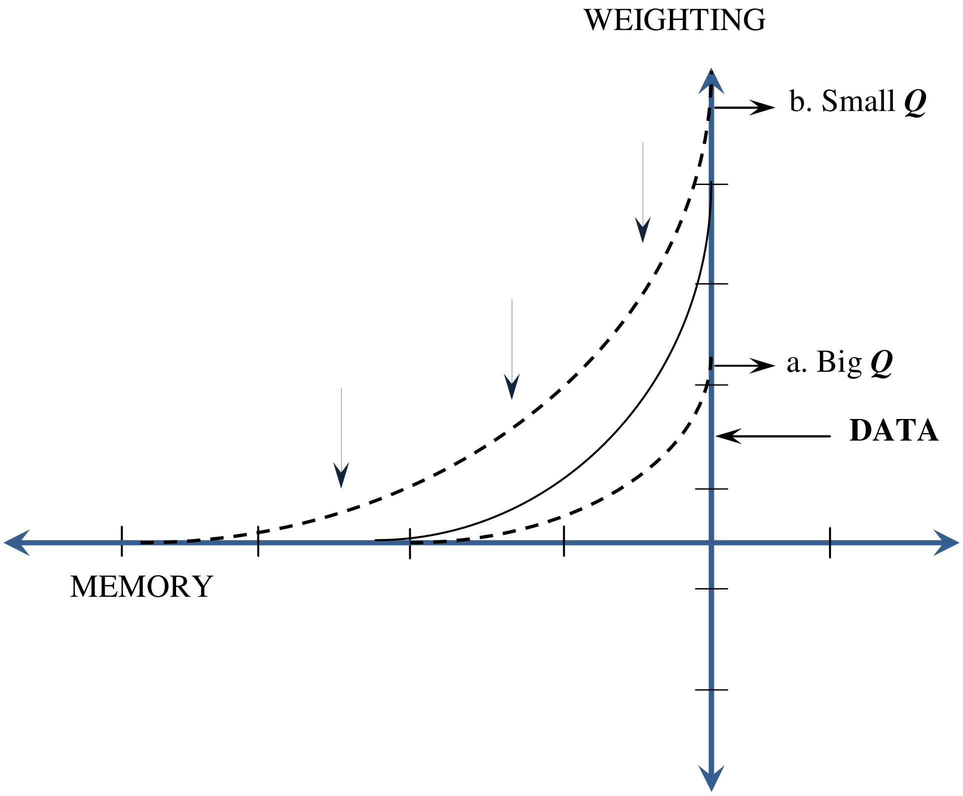


Figure 6.13: Concept of information retention for a Kalman Filter

KALMAN FILTER

```
K_obs(1)=ew(p);
K_C(1,1:Nterm)=rel_uw(p,:);

% KALMAN GAIN: K
K_K=K_M*K_C'/(K_C*K_M*K_C'+K_R);

% FITTING PARAMETERS ki
K_param=K_param+K_K*(K_obs-K_C*K_param);

% smoothing filter
K_param_SMOOTHED=(1-K_alpha)*K_param+K_alpha*K_param_SMOOTHED;

% COVARIANCE MATRIX: P
K_M=(eye(Nterm+1)-K_K*K_C)*K_M+K_Q;
```

The reduced weighting on older measurements results in responses that are variable and requires a smoothing filter to view the desired results.

6.4.4. Partial Correlations

The computational effort of linear partial correlations is dependent on the linear regression performed. In this work, the partial correlation regression is implemented recursively which reduces the runtime at each step and allows for a real-time implementation. However, computation time is much longer in comparison to MWR and the Kalman Filter. This is due to the fact that it requires the computation of 3 sets of partial correlation coefficients:

a) Partial correlation coefficients for the error , equation 5.37: $\boldsymbol{\vartheta}$

b) Partial correlation coefficients for $\boldsymbol{u}$, equation (5.41): $\boldsymbol{\phi}$

c) Partial correlation coefficients between ϵ_i (the error with the influence of other inputs, except u_i completely removed) and δ_i (the input with the effect of the other inputs removed).

The coefficients in a) and b) are computed as follows:

PARTIAL CORRELATION COEFFICIENTS: a) and b)

```
U_C_varo(1:Nterm)=rel_uw(p,:);
U_C_lost_varo(1:Nterm)=rel_uw_lost(:)';

for k=1:Nterm

    U_EC_varo(:,k)=U_EC_varo(:,k)+rel_uw(p,k)*U_C_varo'...
        -rel_uw_lost(k)*U_C_lost_varo';
end
for k=1:Nterm

    % W_M_varo - covariance matrix from MWR
    % Mex_v - extraction matrix
    Mat=Mex_v(:,:,k)*W_M_varo*Mex_v(:,:,k)';
    if (rank(Mat)==Nterm)

        % partial correlation coefficient for u(k)
        param_v(:,k)=Mex_v(:,:,k)'*(Mat\(Mex_v(:,:,k)*U_EC_varo(:,k)));

        % partial correlation coefficient for the error
        % W_EC - scalar product from MWR
        param_ve(:,k)=Mex_v(:,:,k)'*(Mat\(Mex_v(:,:,k)*W_EC_varo));
    else
        param_v(:,k)=zeros(Nterm+1,1);
        param_v(:,k)=zeros(Nterm+1,1);
    end
end
```

These coefficients are calculated at every time step. Once these coefficients are found, the third set of coefficients is evaluated:

PARTIAL CORRELATION COEFFICIENTS: c)

```
for k=1:Nterm
 for i=1:window

 % THIS WILL TAKE A LOT OF TIME
 % pred error - skips the u in question e.g equation(5.35)
 % param_ve - computed by regression
 av(i,k)=ew(i)-rel_uw(i,:)*param_ve(1:Nterm,k)-param_ve(Nterm+1,k);
 % pred error - skips the u in question e.g equation(5.39)
 % param_v - computed by regression
 bv(i,k)=rel_uw(i,k)-rel_uw(i,:)*param_v(1:Nterm,k)-param_v(Nterm+1,k);

 sig_abv(k)=sig_abv(k)+av(i,k)*bv(i,k);
 sig_bbv(k)=sig_bbv(k)+bv(i,k)*bv(i,k);
 end


 if (sig_bbv(k)~=0)

 % one-dimensional fit for desired partial correlation coefficient
 gamma_v(k)=sig_abv(k)/sig_bbv(k);

 else
 gamma_v(k)=0;

 end
end
```

It should be noted that this approach to determining the coefficients relating the error, *e*, to individual input terms, u_i, produces exactly the same coefficients as a direct least squares fit (see chapter 7, section 7.1.2). This is because the direct fit is capable of quantitatively isolating the contributions from each input regardless of the correlation between the input terms.

Within the main program, titled model_error_detector.m (see Appendix A), the user is given the option to skip any of the methods developed above and focus on a single method at a time. At the end of the total recorded time, graphical results are displayed which provides adequate information for the provision of the quality of a model(s).

CHAPTER 7 SIMULATION STUDIES

The theory developed in chapter 5 and the methodology implemented in the previous chapter is applied to two simulation cases. The first case study is based on the Shell heavy oil fractionator. Scenarios such as gain mismatch, correlation amongst inputs, ‘quiet data’, influence of noise levels and time delay mismatch on MPM detection were considered. The second case study is based on a CSTH system and involves the testing of other parameters found in a model for MPM detection. Together with the intuitive qualitative results and quantitative measures, rules of thumb are developed which one may use to detect any MPM in industrial cases. The idea of reducing the model set is thereafter established.

7.1. SHELL HEAVY OIL FRACTIONATOR

For the initial simulation studies the Shell Heavy Oil Fractionator is considered (Figure 7.1). Three product draws and three circulating loops characterize the heavy oil fractionator unit (Patwardhan and Shah, 2002). The feed enters the column in a vapour state and subsequently provides the column’s heat requirement. Economics and operating constraints determine the product specifications for the top and side draws. The bottom draw has no product specification but the lower column temperature is required to be maintained within a desired operating range. The desired product separation is achieved by the three circulating loop present. In order to minimize utility losses, the heat exchangers in the intermediate and upper loops are integrated into other parts of the plant thus having a variation in heat duty requirements. These are regarded as disturbances to the column. Heat removal in the bottom loop is regulated by adjusting the steam intake. This is accomplished by an Enthalpy controller.

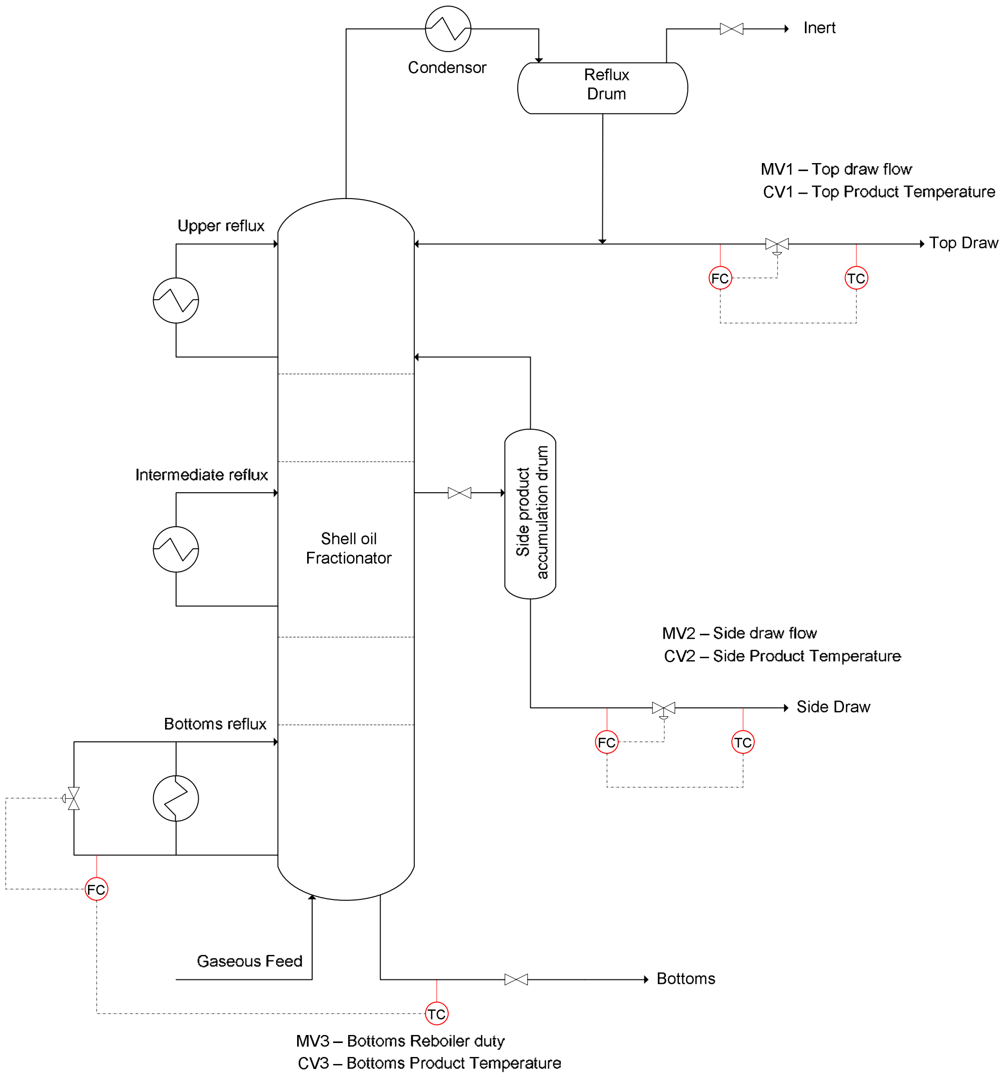


Figure 7.1: Shell Heavy Oil Fractionator

Table 7.1 gives the list of CVs and MVs for this particular system:

Table 7.1:
List of Process variables for the Shell heavy oil fractionator

CV	Description	MV	Description
CV1	Top end temperature	MV1	Top product draw
CV2	Side point temperature	MV2	Side product draw
CV3	Bottoms reflux temperature	MV3	Bottoms reflux duty

The 3 x 3 model is as follows, with time units in minutes:

$$G(s) = \begin{pmatrix} \dfrac{4.05}{50s+1}e^{-27s} & \dfrac{3.50}{60s+1}e^{-28s} & \dfrac{5.88}{50s+1}e^{-27s} \\ \\ \dfrac{5.39}{50s+1}e^{-18s} & \dfrac{5.72}{60s+1}e^{-14s} & \dfrac{6.90}{40s+1}e^{-15s} \\ \\ \dfrac{4.38}{33s+1}e^{-20s} & \dfrac{4.42}{44s+1}e^{-22s} & \dfrac{7.20}{19s+1}e^{-18s} \end{pmatrix} \tag{7.1}$$

Simulations will be carried out on a discretized version of *G* converted with a sampling time related to the dominant time constant obtained from the table below:

Table 7.2:
Sampling interval evaluation

	CV1	**CV2**	**CV3**
Root (min^{-1})	1/50	1/40	1/19
Time constant (min)	50	40	19
Sampling time (min)	7.94	6.35	**3.02**

The MVs were modelled as ML type signals for all simulations unless otherwise stated. The typical constraints placed on each MV are shown below (figure 7.2):

Table 7.3:
Input constraints

MV	**Description**	**Min**	**Max**
MV1	Top product draw	-20	40
MV2	Side product draw	10	90
MV3	Bottoms reflux duty	50	125

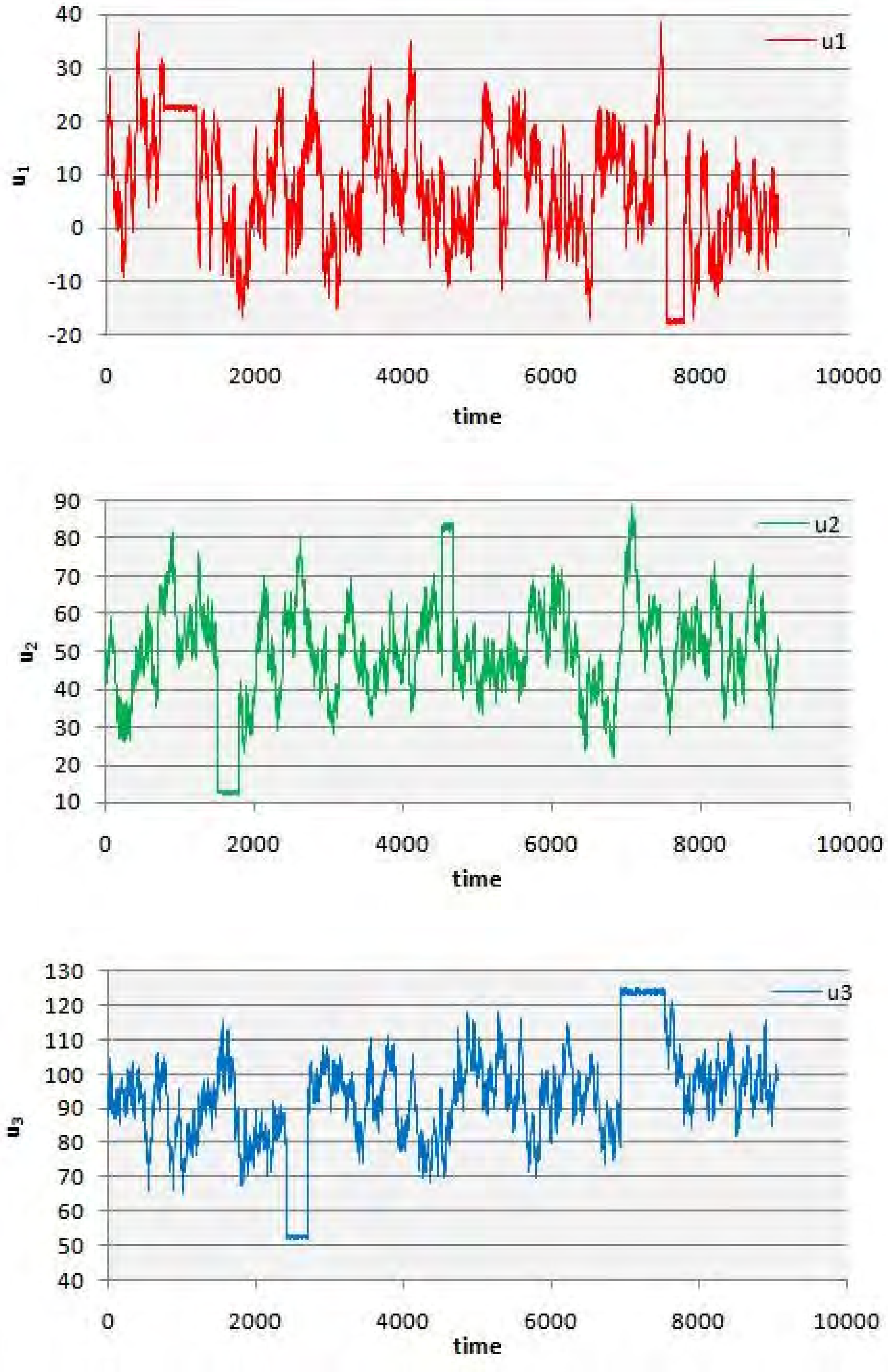


Figure 7.2: Input signals

Correlation plots for these inputs (figure 7.3) reveal that no input is correlated with another, which is to be expected since the model is being excited in open loop.

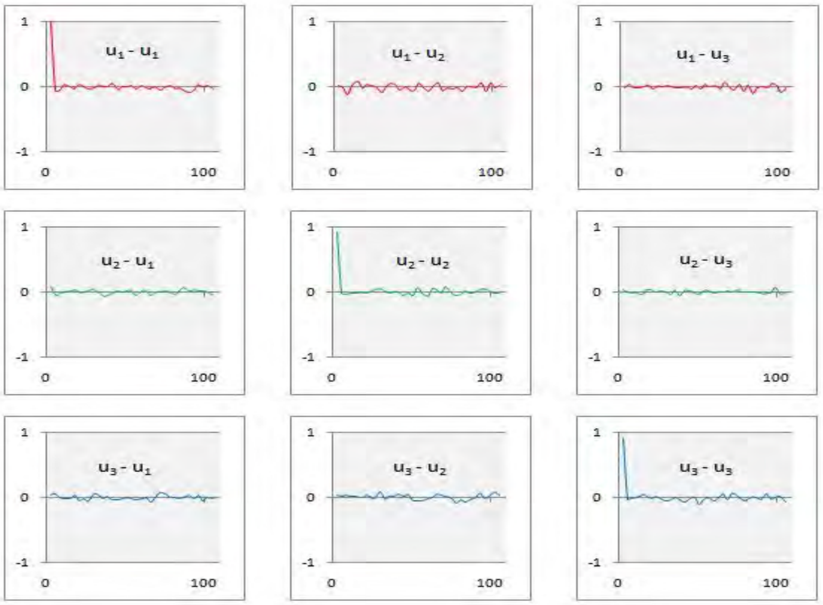


Figure 7.3: Sample Auto-Correlation plots for inputs

All simulations are carried out in an ‘online’ manner where ‘plant’ data is generated and subsequently compared to model generated data. The model and the plant output are simulated in the same way, but with model parameter changes inserted into parameters used to generate the model output to demonstrate the efficacy of the methodology. The resultant error is then being tested for links to u_i by means of correlation analyses as well as regression techniques. These tests would reveal the extent of MPM, if any. The equation error form coupled with differential inputs and outputs (equation 5.25) is used for all simulations except in the case of ‘quiet data’, section 7.1.3.2. Diverse cases were simulated in order to portray realistic occurrences such as gain, time delay and time constant mismatches. In industrial applications, mismatch would be evident in input-output channels: thus the challenge for the methodology lies in its ability to identify the source(s) of significant MPM. Each output, y_j, is dealt with independently. This is possible due to the fact that the models are presented in input-output form. The rate with which a model error is introduced does not alter the likelihood of detection. This is because the models are time-invariant. Simulations were carried out for 3000 sampling points. Model errors are inserted at one-third of the total recorded time. One should be reminded that correlation plots are available at set intervals throughout the data set and only sample plots, spanning 100 second separations, obtained in the middle of the data set are shown. The window size for the moving window regression is taken as 800 sampling points, representing 2416 seconds for the chosen sampling interval of 3.02 s. One can recall that the matrix ***Q*** and the matrix ***R***, both usually diagonal, determine the efficacy of the Kalman Filter operation. For all simulations performed below, the values along the diagonal of ***Q,*** corresponding to the k_{i} estimates, were taken as 0.0001. In terms of the ***R*** matrix, the first value along the diagonal, corresponding to the α observation, was taken as 0.1, with all but the last remaining values set at 1. The final offset term within the R matrix diagonal was given a value of 100.

7.1.1. Scenario 1 – Gain Mismatch

In this case, mismatch was added in such a way as to create a situation where the gain in the MV1 – CV1 channel is over-estimated by +50%. This means that the model gain is 50% higher than the plant gain or conversely, the plant gain in this channel is 33 $^{1}/_{3}$ % lower than the assumed model gain for which the plant was designed. In order to create a more realistic scenario, mismatches of +10% and -10% were added in channels MV2 – CV1 and MV3 – CV1 respectively. Figure 7.4 illustrates the error that forms due to these model mismatches:

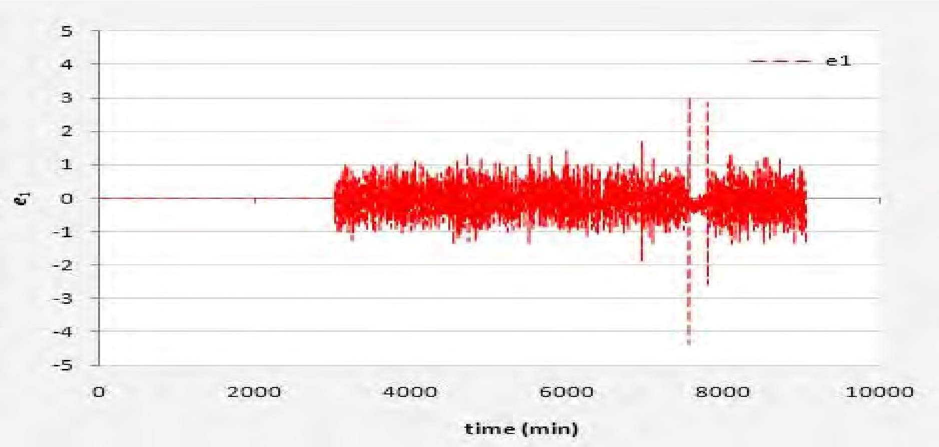


Figure 7.4: Error due to gain MPM

The correlation between this error and the respective inputs is shown below. Figure 7.5(a) shows a significant correlation in model MV1 – CV1. Small correlations are attributed to the channels MV2 – CV1 and MV3 – CV1 as they exhibit small mismatches.

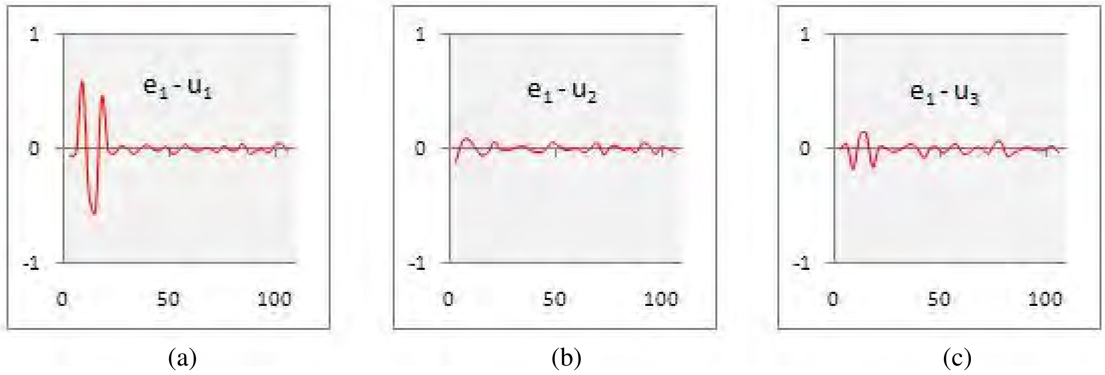


Figure 7.5: Cross - Correlation plots for gain mismatch

Although these plots are useful in isolating channels that exhibit MPM, they do not provide any information as to the extent of the MPM.

Conversely, the regression plots reveal the extent of the mismatch. Recalling table 5.2 and the expected k_i fitting related to equation (5.25), the following plots are obtained for the Moving Window Regression and the Kalman filter correspondingly:

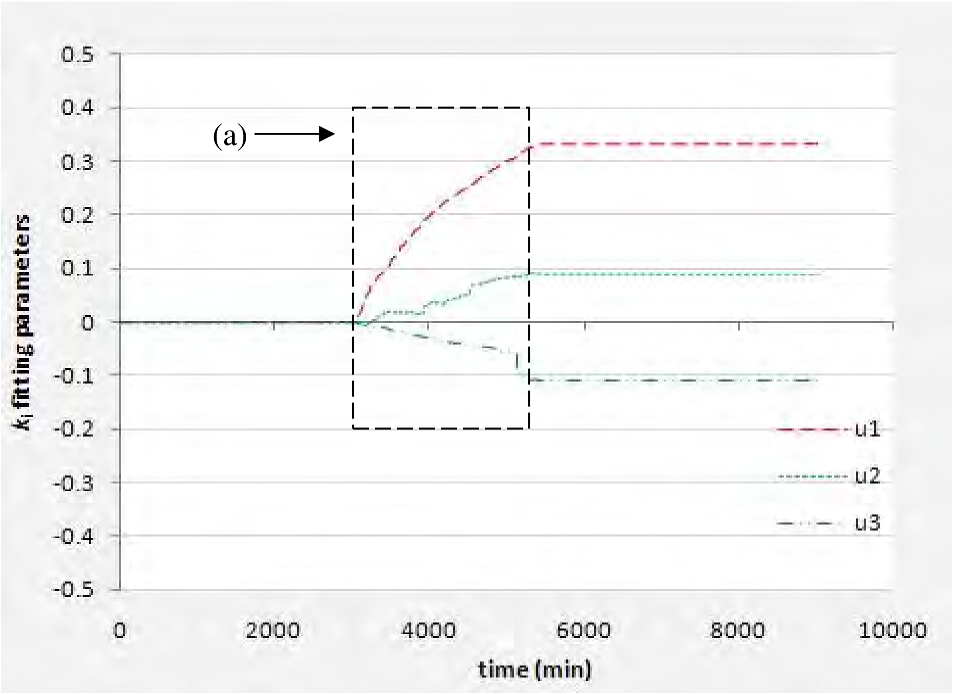


Figure 7.6: Moving Window Regression plot – mismatch detection for CV1

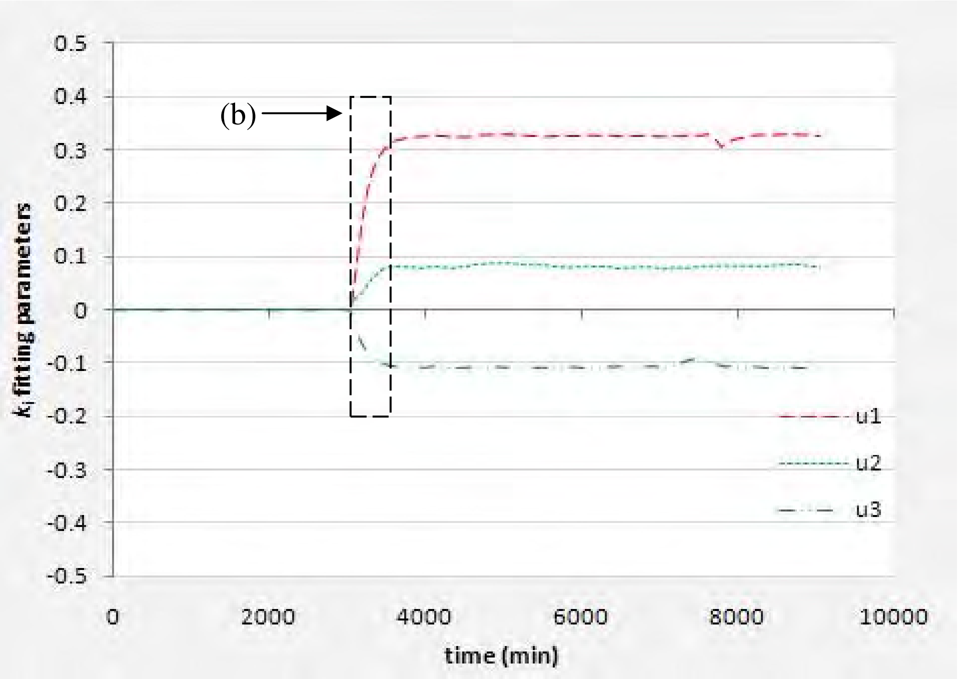


Figure 7.7: Kalman Filter parameter estimation plot – mismatch detection for CV1

Table 7.4 presents a summary of the overall fitting parameters obtained through regression.

Table 7.4:
Summary of the overall fitting parameters obtained

	k_1	k_2	k_3
EXPECTED k_i	0.3333	0.091	-0.1111
MOVING WINDOW REGRESSION	0.3327	0.092	-0.1108
KALMAN FILTER	0.3273	0.089	0.1089

These fitting parameters represent the way the error relates to a particular u_i expressed as a fraction of the gain used in the model for that u_i term. Both regression techniques displayed above, show the expected result. It is interesting to note the time each method takes to reach the correct parameter fittings. The Kalman filter depends largely on more recent values and hence takes a shorter period of time to obtain it's fitting parameters as shown by the box labeled (b) in figure 7.7. This makes the Kalman Filter desirable in terms of time taken for computation. The recursive regression based on the moving window approach puts an equal weighting on each point within the window as illustrated by box (a) in figure 7.6, thus the response to reach the intended result is longer but smoother in comparison to the Kalman filter response. It can also be deduced from table 7.4 that applying equal weighting to past data results in a closer fit to the expected parameters.

7.1.1.1. Fitting Parameter multiplicative factors

In a more practical view, one may obtain the actual factors by which the gain has changed. One may recall the expected fitting parameter $\boldsymbol{k}_\mathbf{i}$ related to equation (5.25):

$$k_i = 1 - \frac{K_{pi}\left\{\prod_{j=1}^{N} A_{pi}(1)\right\}}{K_{mi}\left\{\prod_{j=1}^{N} A_{mi}(1)\right\}} \tag{7.2}$$

Suppose there is no mismatch in the lag polynomials ($A_{mi} = A_{pi}$) and that the model gain (K_{mi}) is η_i x plant gain (K_{pi}). Equation (7.2) then becomes:

$$k_i = 1 - \frac{K_{pi}}{\eta_i K_{pi}} \tag{7.3}$$

Rearranging equation (7.3) and cancelling common factors, one obtains an equation that results in the actual factor by which a model gain has changed:

$$\eta_i = \frac{1}{1 - k_i} \tag{7.4}$$

Table 7.5:
Summary of fitting parameter factors obtained

	η_1	η_2	η_3
EXPECTED $\eta_\mathbf{i}$ FACTORS	1.50	1.10	0.90
MOVING WINDOW REGRESSION	1.4986	1.1017	0.9002
KALMAN FILTER	1.4865	1.0971	0.9018

It should be noted that this form is limited only to the detection of GAIN mismatches. The results relating to the expected factors are shown above and the graphical results are shown below:

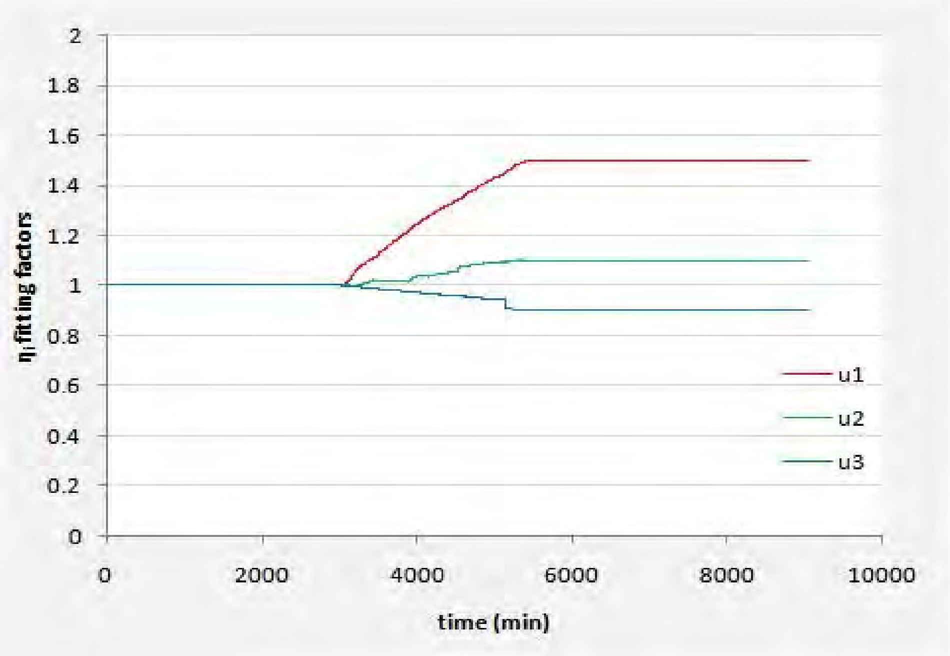


Figure 7.8: Moving Window Regression factor plot – mismatch detection for CV1

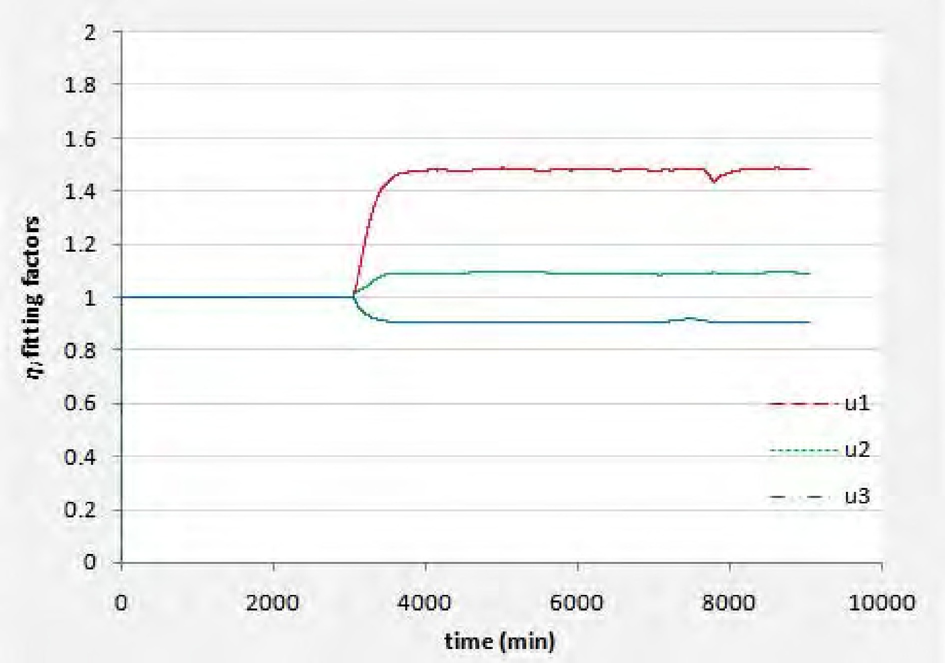


Figure 7.9: Kalman Filter factor estimation plot – mismatch detection for CV1

7.1.2. Scenario 2 – Correlation amongst inputs

In closed-loop operation (see section 2.4.1, figure 2.5), the multivariable controller, Q, computes each MV in **u** at every sampled instant. It can be seen that the same error vector forms the basis for the calculation of each MV. This results in coordinated adjustments of the MVs and depending on the controller design, this may lead to the correlation between MVs. The regular correlation analysis between the residuals and the inputs may be confounded by such correlations amongst inputs resulting in false model error detection.

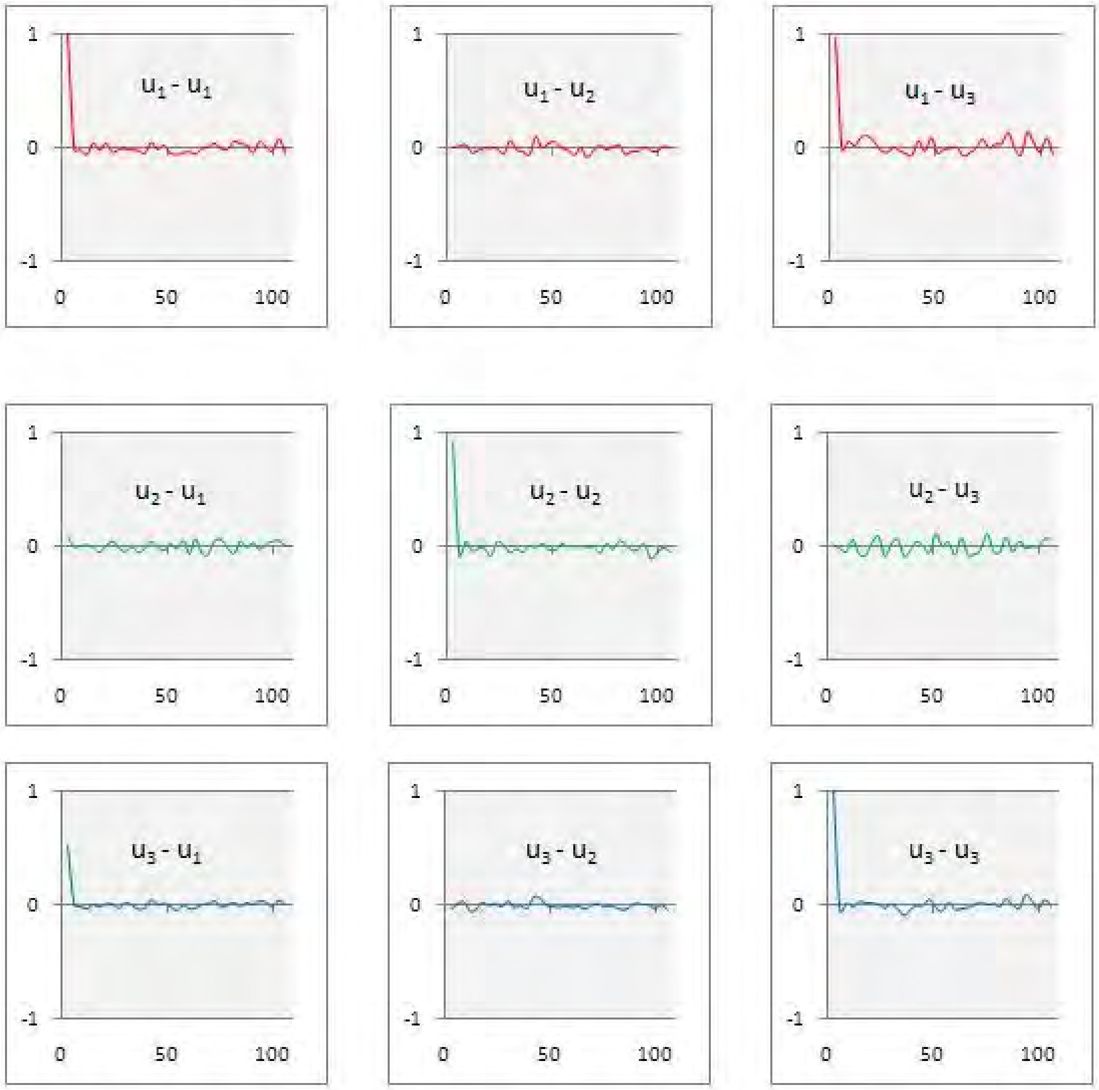
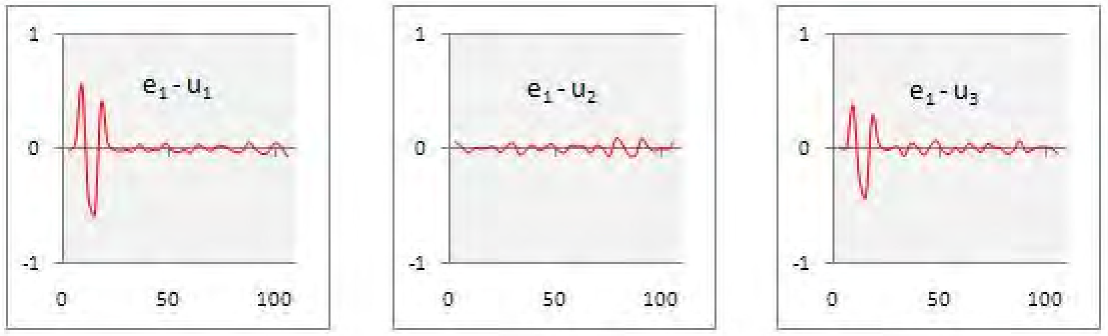


Figure 7.10: Correlation plots for correlation amongst inputs

For example, consider the same case of mismatch in section 7.1.1 (only mismatch exhibited in channel MV1 – CV1 for simplicity), but with MV3 having 50% of the behavior exhibited by MV1. Figure 7.10 shows the correlation plots for the inputs. It can be seen from these plots that MV1 and MV3 are correlated. With reference to figure 7.11, although the first model term in figure 7.11(a) is at fault, there is a misleading correlation of the output error in y with u_3 as shown in figure 7.11(c) (as well as the expected correlation with u_1). This is because the correlation analysis does not view each input independently and thus does not remove their interactive contributions.



(a) (b) (c)

Figure 7.11: Correlation plots with correlation amongst inputs

It should be noted again the Kalman Filter settings for the matrices ***Q*** and ***R*** are the same in this scenario as detailed in the introduction to section 7.1.

The partial correlation methodology serves to provide the correct result, as depicted in figure 7.12, since it removes the influence of other variables when dealing with a specific input-output pairing.

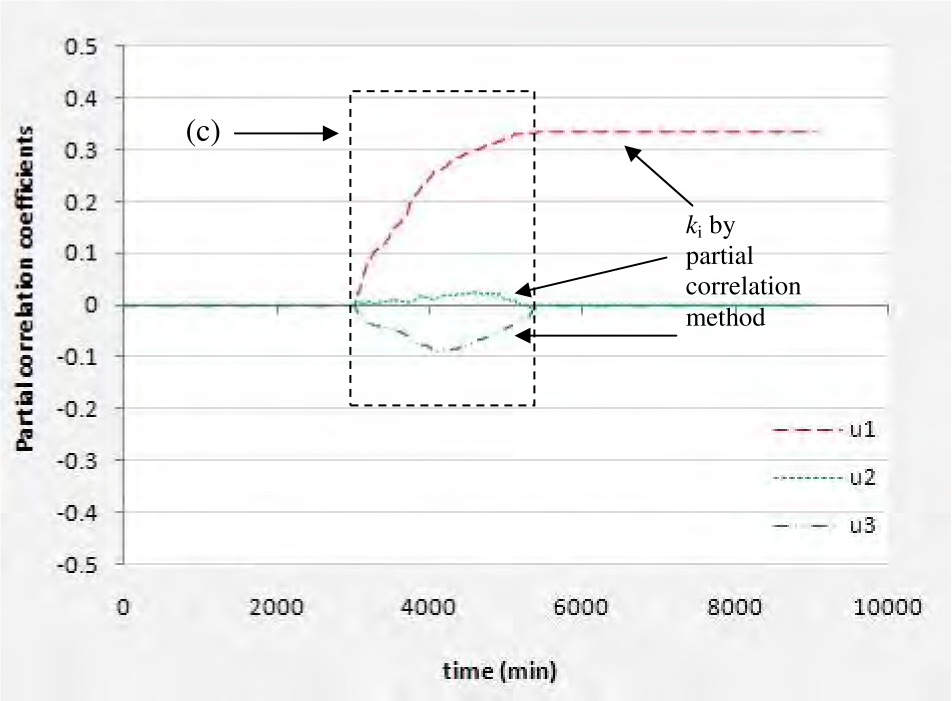


Figure 7.12: Partial correlation coefficient plot

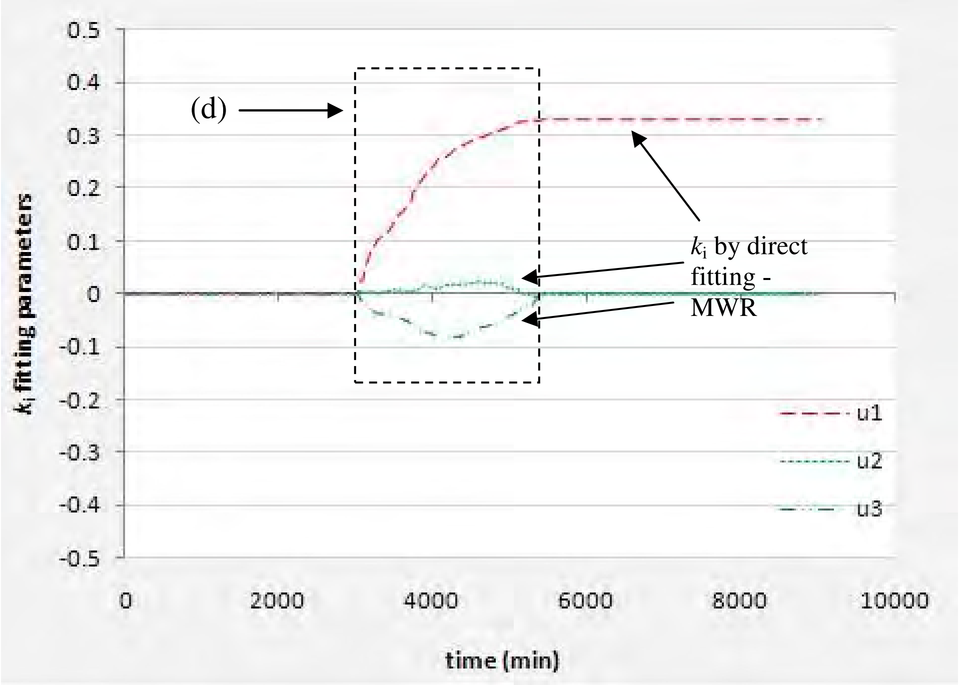


Figure 7.13: Moving Window Regression plot – correlation amongst inputs

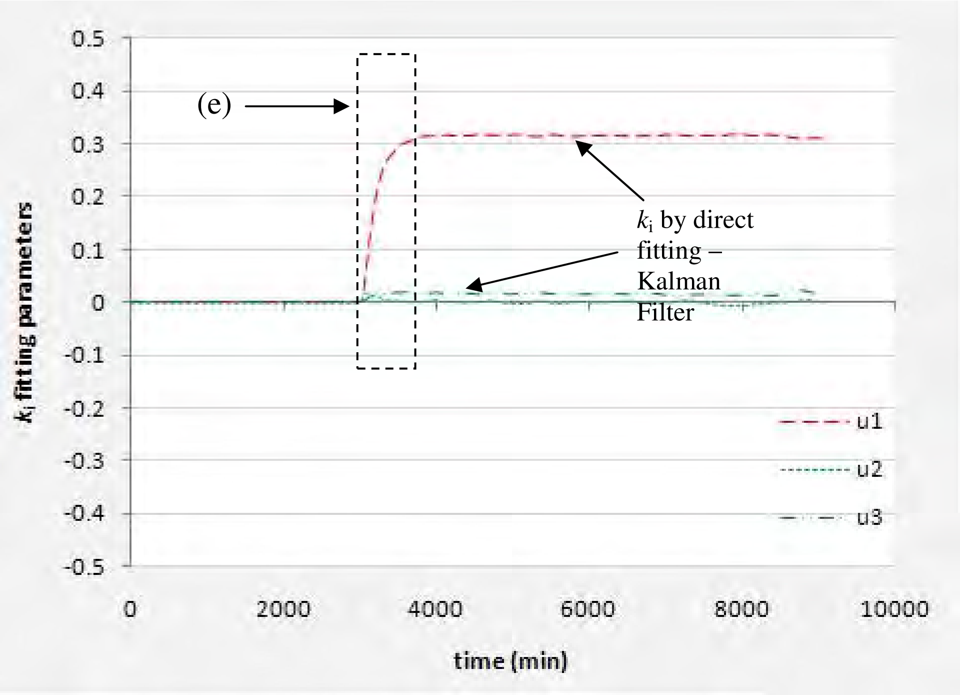


Figure 7.14: Kalman Filter parameter estimation plot – correlation amongst inputs

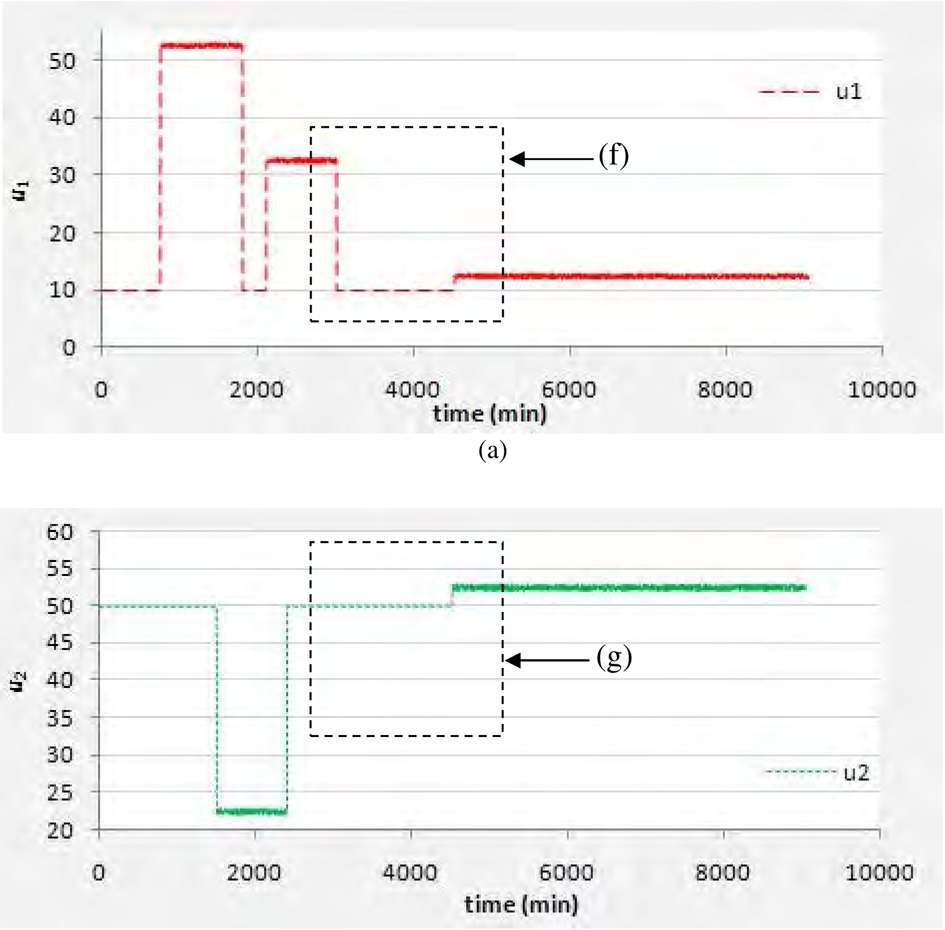
As noted in section 6.4.4, the partial correlation regression (figure 7.12) produces the same results as direct regression (Figure 7.13). Apart from the slight drift away from zero for the partial correlation plot relating to u_3, the results are as expected. The slight drift is due to the fact that the window is still filling up during this period, indicated by box (c). In contrast, to the cross-correlations, the regression techniques illustrated above are shown to be capable of dealing with the quantitative effects of each MV separately.

7.1.3. Scenario 3 – 'Quiet' data

The concept of persistent signal excitation allowing for informative data is often regarded as a given in the development of closed – loop model validation methods (Badwe et al., 2009). The reasoning behind this is that under typical MPC operation, the targets (set-points) are regularly computed by the upper LP layer. However, it is shown that the condition of informative data in a closed – loop system is generally not guaranteed by sufficient signal excitation (Ljung, 1987). Gustavsson (1977), whose work concerned the identification of closed – loop processes, elaborated on the statement by Ljung by articulating that one of the natural purposes of a feedback controller is to minimize the output variance and consequently minimize the output information content.

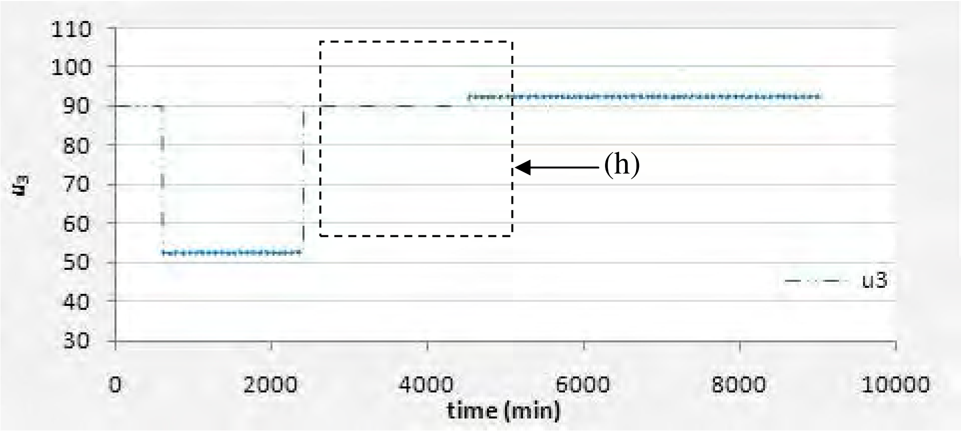
7.1.3.1. Limited informative data

In the absence of sufficient signal excitation, the methodology developed can only display results in a limited capacity. Consider that the output CV1 is nearing its intended target and the input signals are now shown to be exhibiting a few movements as shown in figure 7.15. Due to the few movements of each input signal, the provision of 'enough' informative data is made possible for the limited detection of any MPM. Consider the same mismatch implemented in scenario 1, but with only the gain in channel MV1 – CV1 being overestimated by 50%. The Cross – Correlation results are shown below in figure 7.16 together with the regression results. Ideally one would expect the same results obtained for k_1 in table 7.4.



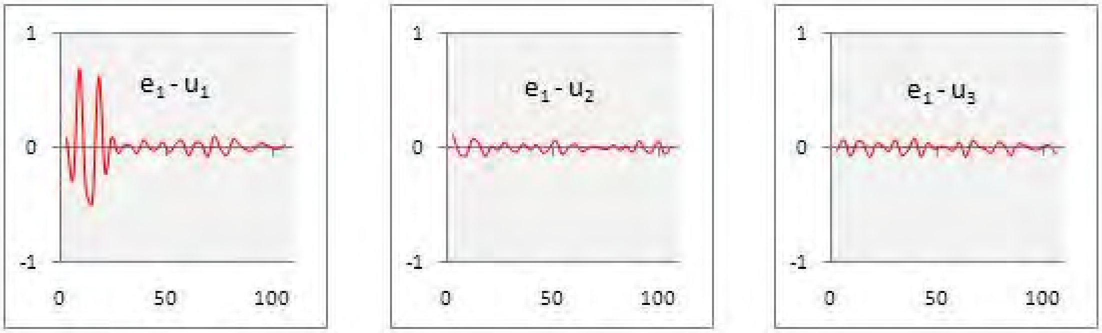
(a)

(b)



(c)

Figure 7.15: Input signals – limited informative data



(a) (b) (c)

Figure 7.16: Correlation plots for gain mismatch with limited informative data

Table 7.6:
Summary of overall fitting parameters: limited informative data

	k_1	k_2	k_3
EXPECTED k_i	0.3333	0.00	0.00
MOVING WINDOW REGRESSION	0.2757	2.83×10^{-3}	3.32×10^{-3}
KALMAN FILTER	0.1703	1.93×10^{-3}	1.27×10^{-3}

The fitting parameters differ from the expected result due to the flat nature of the input data with reference made to boxes labeled (f) through to (h). If the excitation reduces, the input term (β) in equation (5.32) will tend to get smaller and the default k_{exp} values will be given. Also due to the flat nature, the moving window regression takes a while to reach a steady fitting parameter, shown by box (i). The Kalman filter is shown to drop back to zero at one instance, illustrated by box (j). This is because u_1 is relatively flat and thus lacks any informative content.

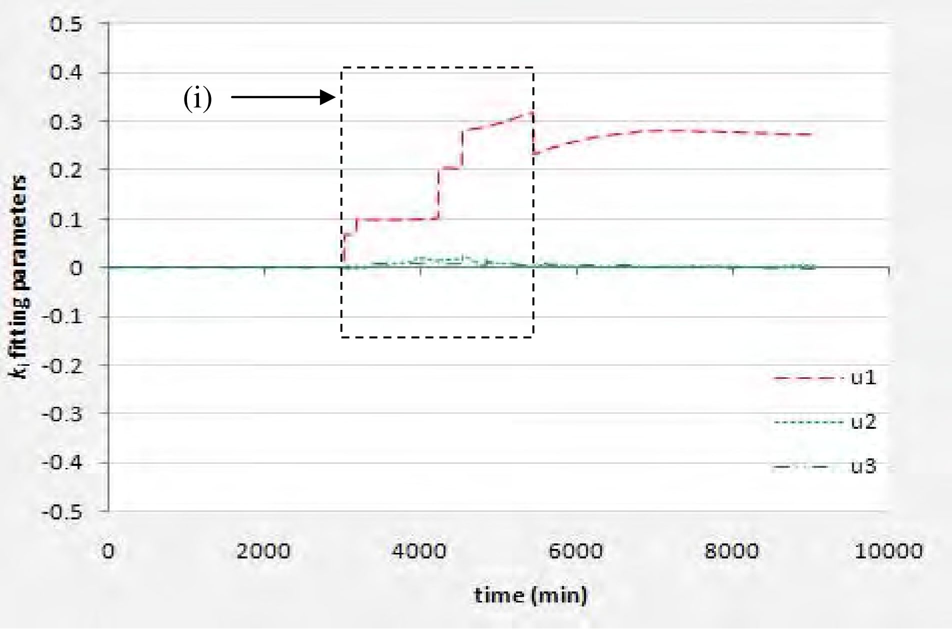


Figure 7.17: Moving Window Regression plot – limited informative data

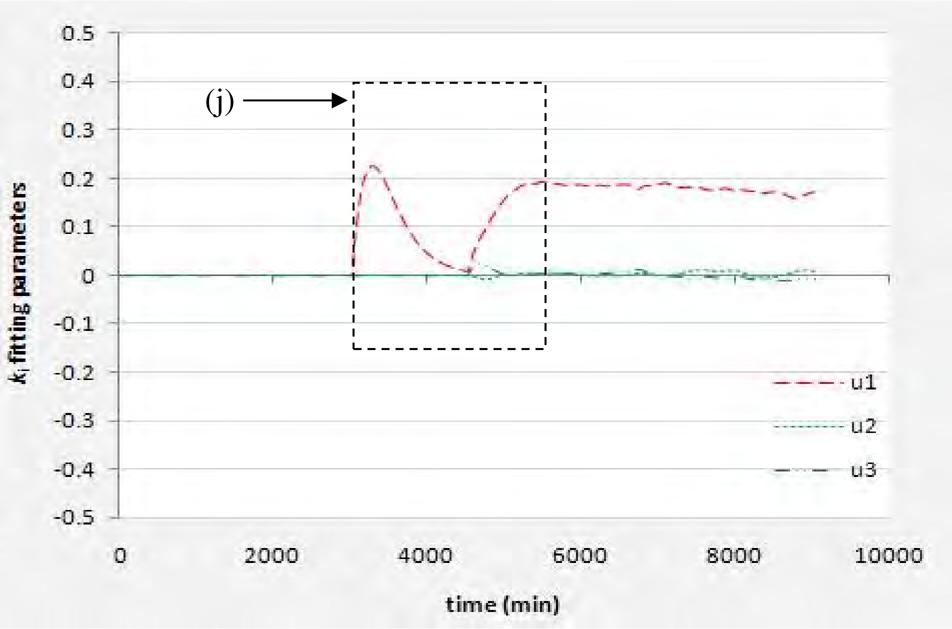
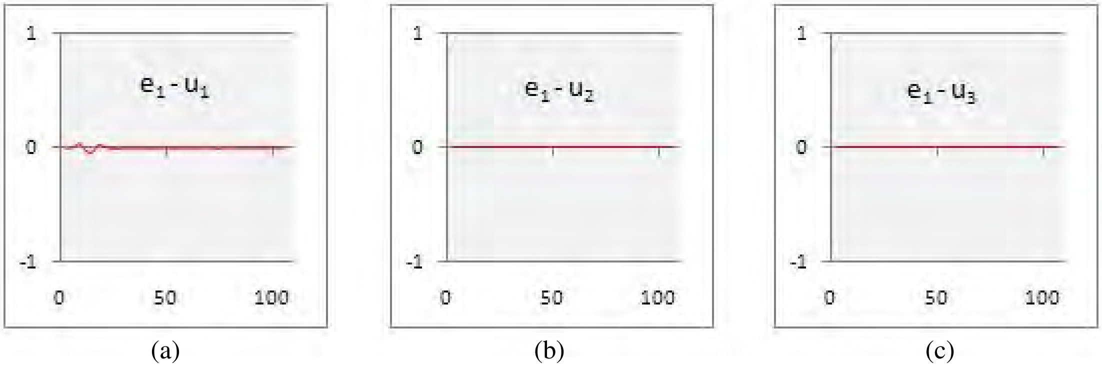


Figure 7.18: Kalman Filter parameter estimation plot – limited informative data

7.1.3.2. Set – point target reached

When a set – point target is reached and is required to be maintained at this respective point, the current error definition form used in the methodology to detect any model degradation fails to provide adequate results. The correlation plots reveal no significant mismatch in channel MV1 – CV1 even though a mismatch is evident in the model gain (figures 7.19 to 7.21):



(a) (b) (c)

Figure 7.19: Correlation plots for gain mismatch – set-point target reached

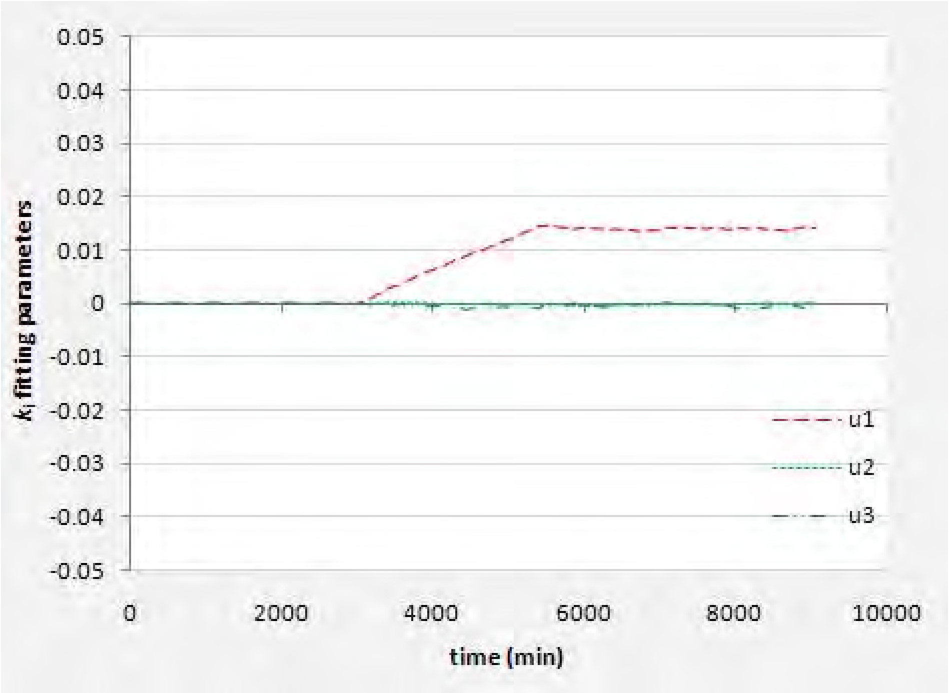


Figure 7.20: Moving Window Regression plot – set-point target reached

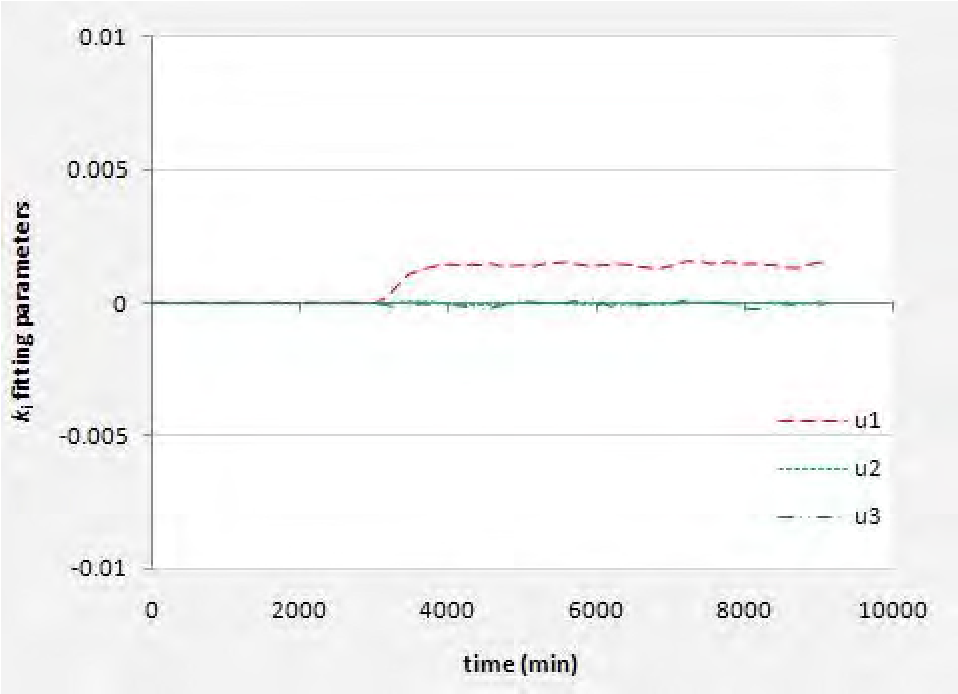


Figure 7.21: Kalman Filter parameter estimation plot – set-point target reached

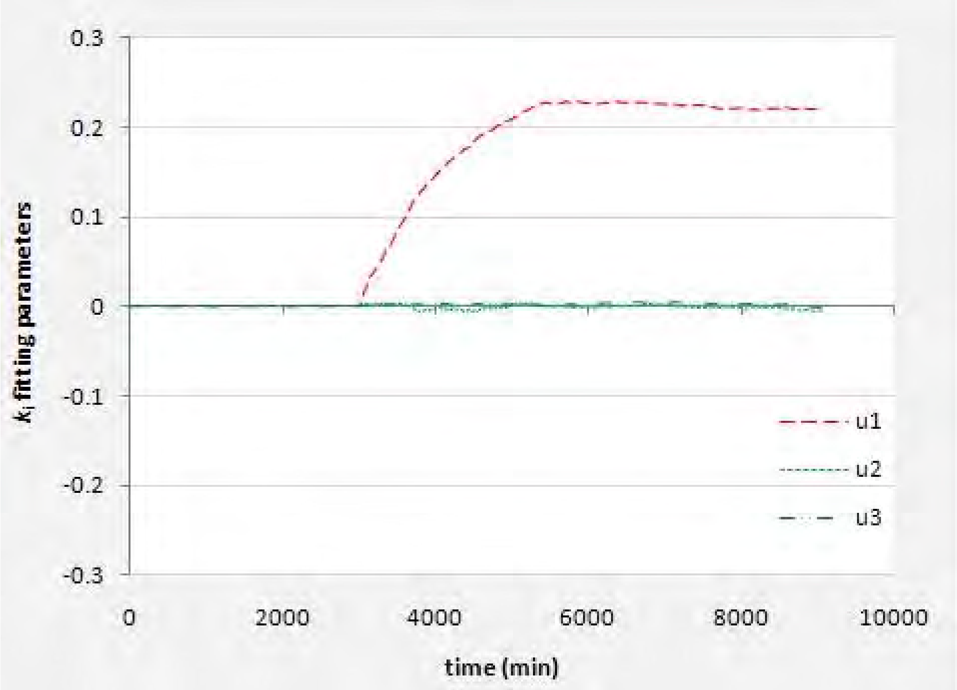


Figure 7.22: Moving Window Regression plot with fixed offset – set-point target reached

Although it can be seen in both figure 7.20 and 7.21 that there does exist a deviation away from zero for k_1, these values are too minute to provide evidence that a mismatch is present. Nevertheless, by using a direct form of inputs and outputs together with an open-loop predictor, as shown by equation 5.24, one is capable of providing an indication of a significant deviation away from zero by fixing the offset term, although it may not provide the expected result due to the lack of information present in the data. This result is shown above in figure 7.22. Recall that the preceding results were based on the differential equation error form, equation 5.25, so that the offset information is lost.

7.1.4. Scenario 4 – Sensitivity to noise levels

Noise levels are critical for the performance of the method. It is necessary that the magnitude of the modelling error dominates over process noise. Noise often corrupts the signal obtained from the plant i.e. the plant inputs and outputs. Consequently these noise signals add to the error that is formed possibly due to MPM. In this case, the effect of dissimilar noise levels on the ability to detect MPM will be demonstrated. These tests will be performed on the gain mismatch in channel MV1 – CV1 to maintain a level of consistency.

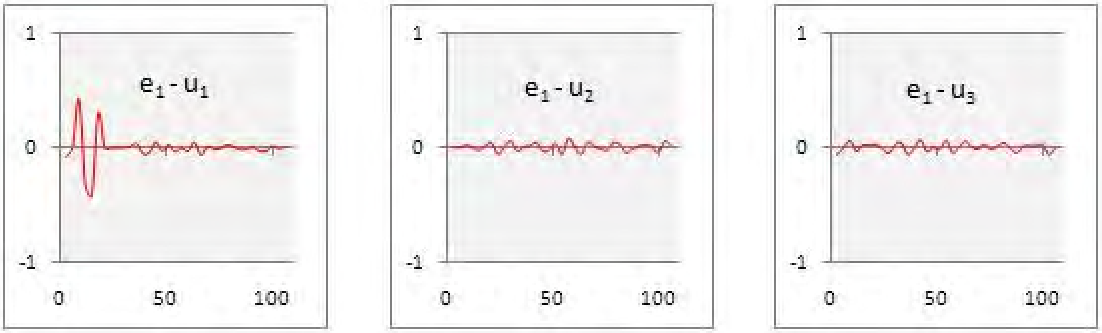
All noise signals are modelled as Gaussian white noise signals with varying ranges which is added to the plant output, y_p. Table 7.7 shows the various standard deviations of the noise signals used as well as the signal – to – noise ratios. One should note that the equation error, differential form for the error formulation is employed in this section.

Table 7.7:
Various SNR ratios

	Noise std deviation σ_{noise}	**Error std deviation** σ_{error}	**Signal – to – noise ratio SNR$_e$**
Case 1	0.0832	0.4179	5.0228
Case 2	1.0091	3.2812	3.2561
Case 3	8.4236	22.5631	2.6785

7.1.4.1. Case 1

In this case the standard deviation of the noise is relatively low compared to the standard deviation of the output error. Due to the fact that the error standard deviation dominates over the noise standard deviation, the results obtained are as expected. This is because the error is predominantly due to MPM rather than excessive noise present. The results obtained are shown below in figures 7.23 to 7.25:



(a) (b) (c)

Figure 7.23: Correlation plots – noise levels case 1

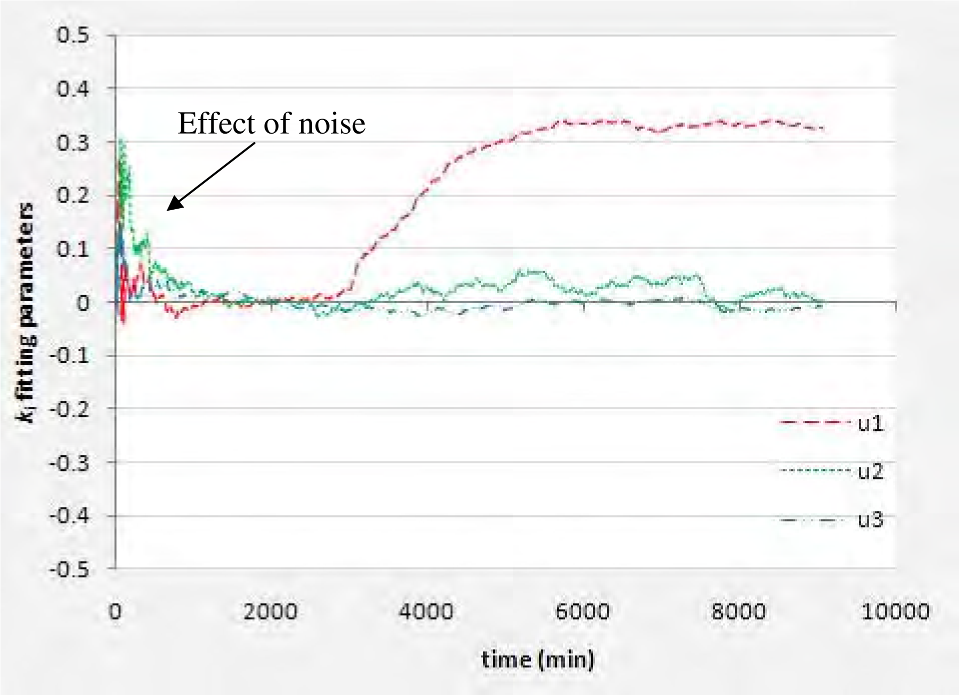


Figure 7.24: Moving Window Regression plot – noise levels case 1

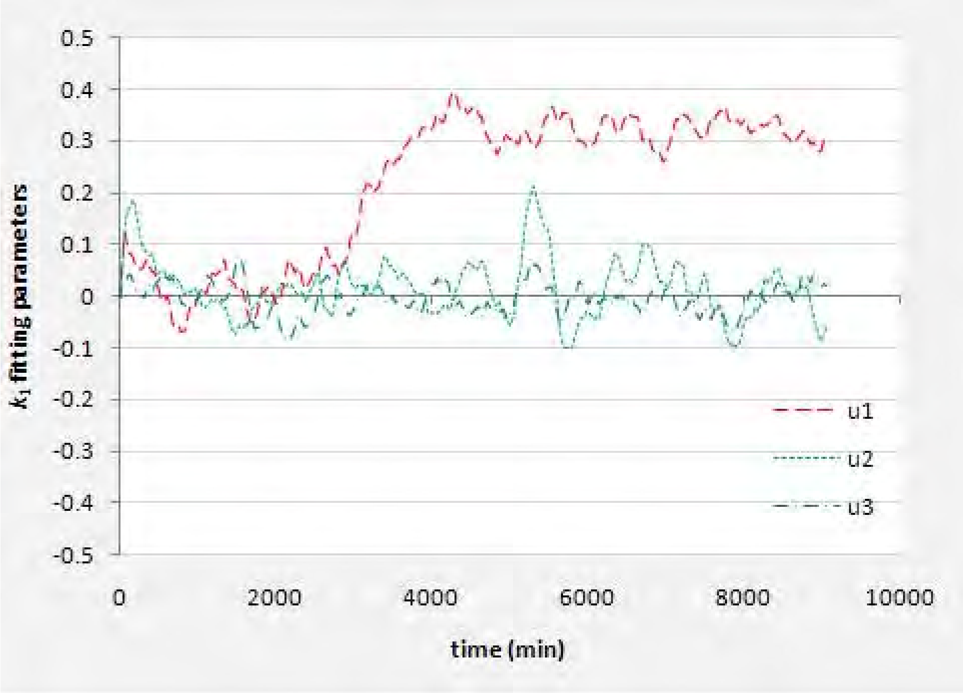


Figure 7.25: Kalman Filter parameter estimation plot – noise levels case 1

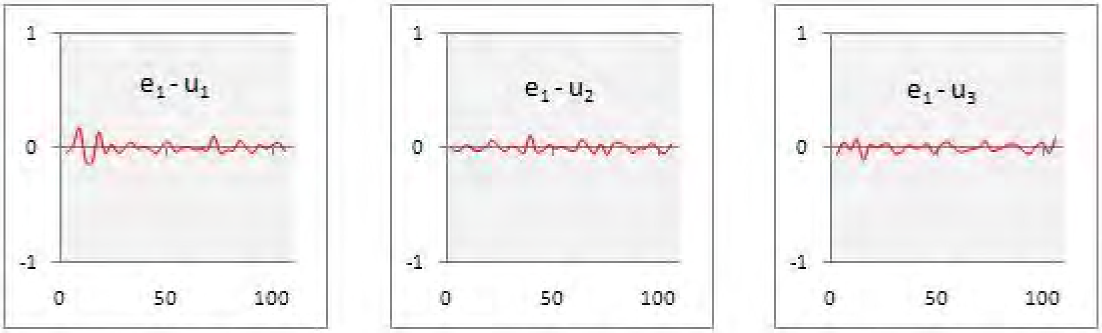
The correlation plot in figure 7.23(a) is similar to the plot shown in figure 7.5(a) except that here the correlation plot is slightly smaller.

The moving window regression plot illustrates the slight effect of the noise in the period before the mismatch is inserted. The regression does not know which model parameters contribute to the error formed prior to the mismatch as the window is being filled for the first time and thereby attributes the error to all parameters involved. However, once the mismatch is inserted, the moving window provides the expected result by attributing the error formed to the gain in channel MV1 – CV1.

The same Kalman Filter settings described in section 7.1 are used for these simulations. The Kalman filter, on the other hand, is shown to be more susceptible to disruption by the presence of noise. This is because the Kalman filter is highly sensitive to the effect of recent data values and thus produces a more variable response in comparison to the moving window regression.

7.1.4.2. Case 2

In this case the standard deviation of the noise signal is increased and the effect to this increase in noise variance is shown below. As the SNR value starts to drop, the methodology no longer views the error formed as simply a modeling error (figures 7.26 to 7.28):



(a) (b) (c)

Figure 7.26: Correlation plots – noise levels case 2

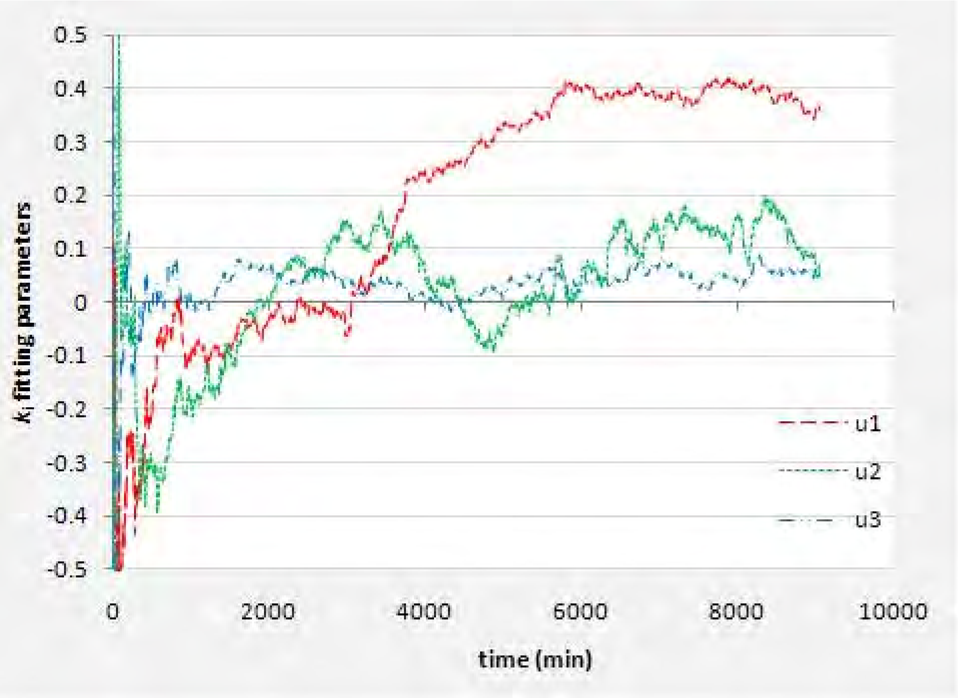


Figure 7.27: Moving Window Regression plot – noise levels case 2

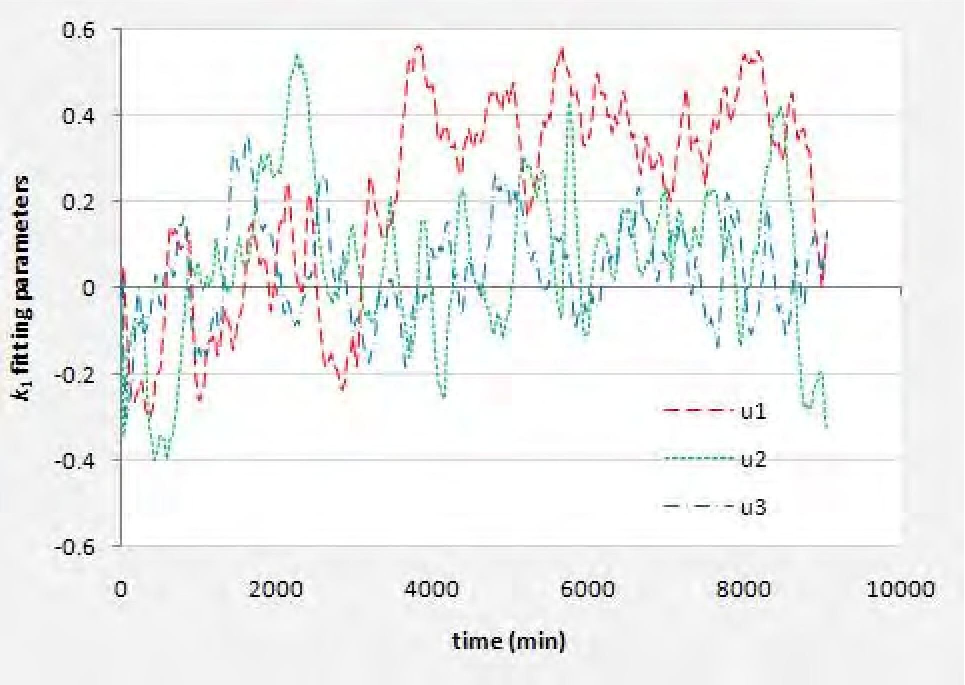


Figure 7.28: Kalman Filter parameter estimation plot – noise levels case 2

Figure 7.26(a) shows a drop in the magnitude of the correlation plot. Cross – Correlation plots deal with the correlation between the output error and the corresponding inputs. If the variance of the noise signal is increased, the magnitude of the correlation coefficients drop as the error formed is not only due to MPM but also due to the influence of noise present.

The influence of the noise signal is more pronounced on the moving window regression plot depicted in figure 7.27. As the noise levels increase, the regression tries to determine the modeling error as shown by the fitting of parameter k_1, but also attributes portions of the noise signal as an error to the remaining parameters.

The Kalman filter produces a rather random display of fitting parameters in figure 7.28 due to the effect of the noise levels. From this plot it is difficult to perceive which parameter deviates away from zero due to a modeling mismatch. Thus for the purpose of determining a modeling mismatch in the presence of noise levels, the parameters obtained from the Kalman filter regression may be required to be passed through a smoothing filter to view adequate results.

7.1.4.3. Case 3

Here, the noise signal is further increased, dropping the SNR value further down. In this case, the noise is primarily dominant over modeling error.

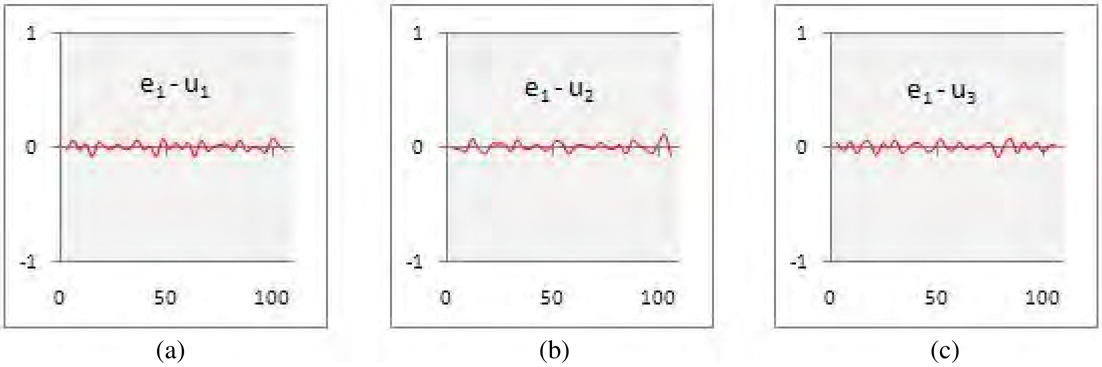


Figure 7.29: Correlation plots – noise levels case 3

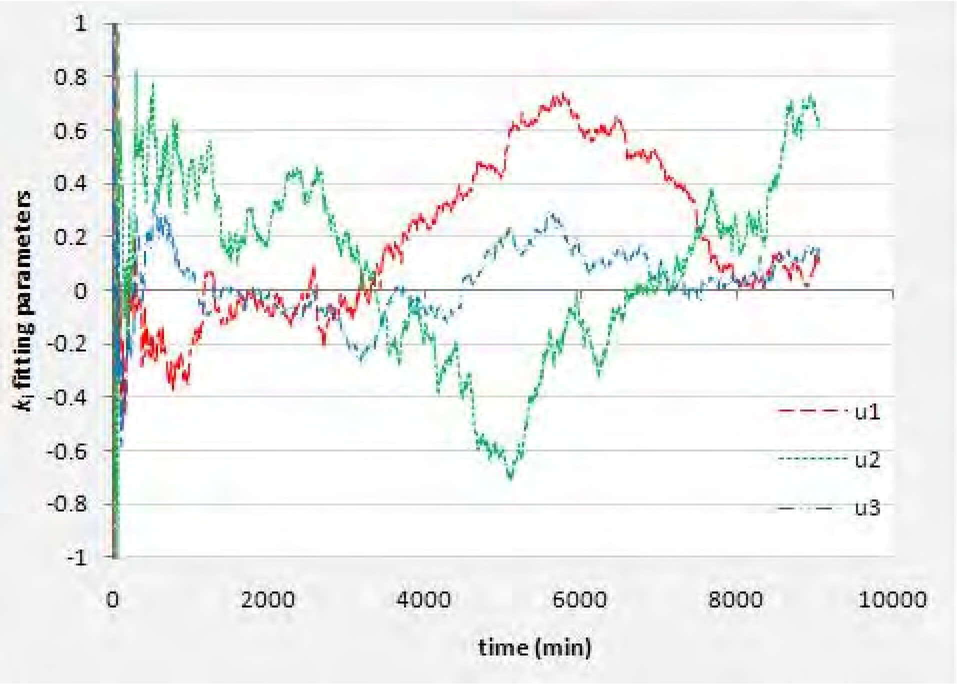


Figure 7.30: Moving Window Regression plot – noise levels case 3

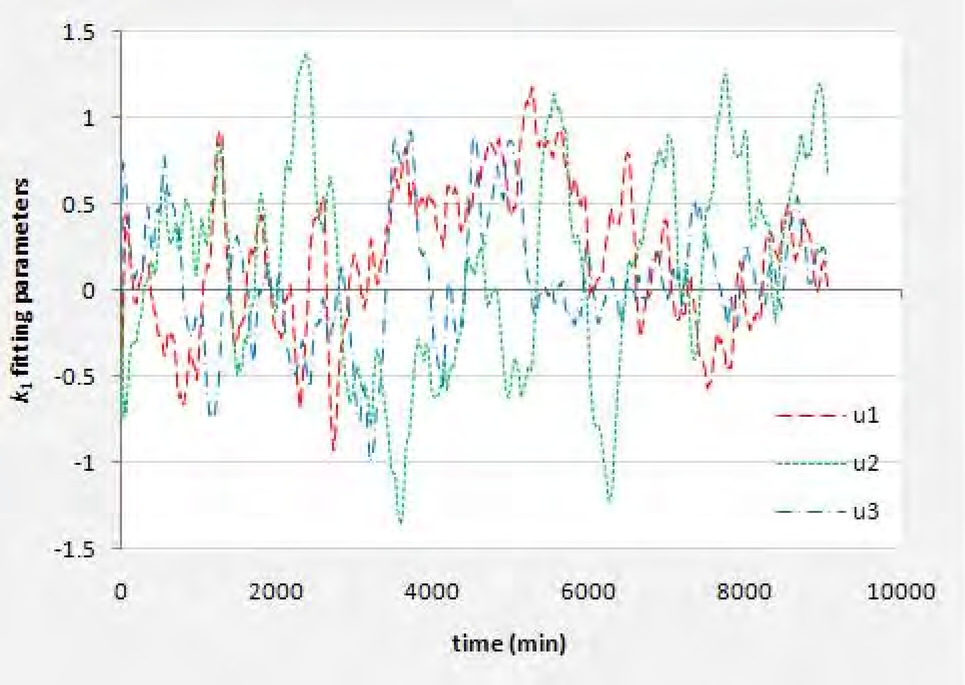


Figure 7.31: Kalman Filter parameter estimation plot – noise levels case 3

All the plots depicted in figure 7.29 show correlation coefficients that are essentially zero. The implication here is that the noise signal dominates over the modeling error obtained. Thus, the correlation analysis cannot determine which input – output pairing is mismatched.

Both regression plots shown in figure 7.30 and 7.31 respectively show that the parameters exhibit a haphazard behavior in trying to fit to the error that is formed. At this stage, with such high levels of noise, it is nearly impossible to determine the extent of a model mismatch. Even though different types of process noise also could be considered as modelling errors, where a noise model is defined (ARMAX form of model definitions, see equation 3.19), this is not considered. This thesis focuses on the deterministic modelling errors in plant dynamics. In order to combat the effect of noise influencing the plant signal, one is required to use a low-pass or high-pass filter (depending on the frequency of noise) to reduce the influence of the noise on the signal to an extent so that the methodology may be applicable in cases where high levels of noise signals are present.

7.1.5. Scenario 5 – Time Delay Mismatch

In this case, the time delay term from equation 7.1 in channel MV1 – CV1 is overestimated by 50%. This means that the plant lags the model by half the time delay value which was initially desired. The mismatch is placed in the same channel (MV1 – CV1) as the scenario in section 7.1.1. This is done in order to compare the results obtained for these two cases. The error plot in figure 7.32 for this mismatch covers a larger range compared to figure 7.4:

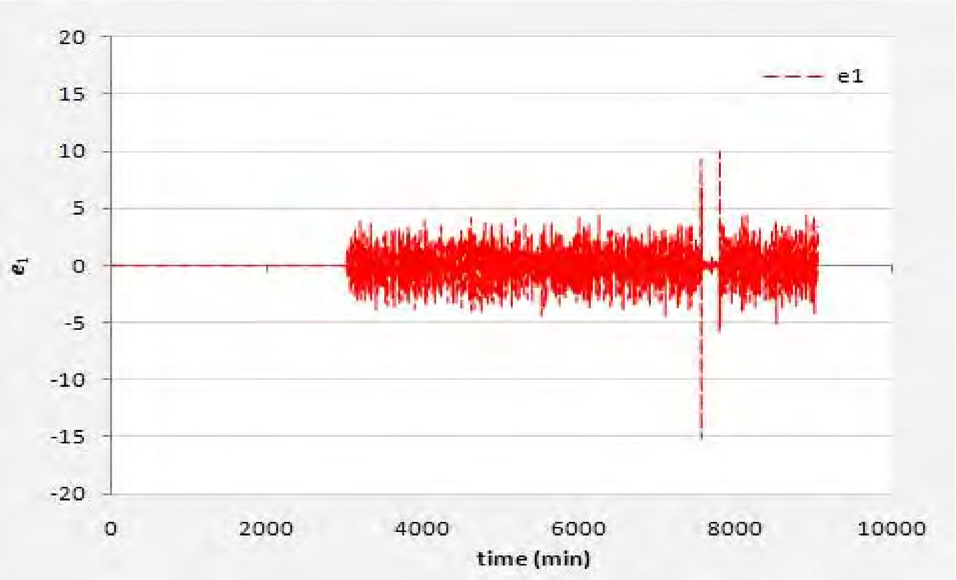
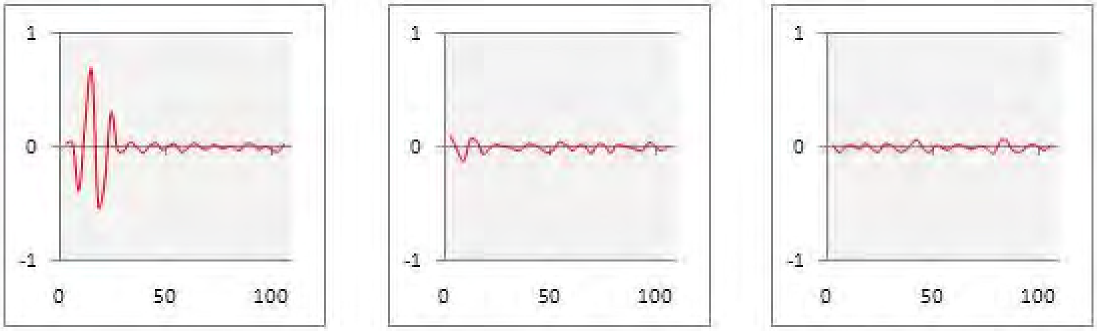


Figure 7.32: Error due to time delay

The Cross - Correlations plots for this case are plotted in figure 7.33:



(a) (b) (c)

Figure 7.33: Cross – Correlation plots for time delay mismatch

It can be seen that significant correlation is exhibited in figure 7.33(a) which is as expected. An interesting feature however is the shape of the correlation plot in figure 7.33(a) compared to that exhibited in figure 7.5(a). Each plot has a unique characteristic pattern depicted. Thus one may study these plots alone in order to gain some insight into which factor may have caused this mismatch, refer to figure 7.5(a) and 7.33(a). For example, for gain mismatch the correlation plot is illustrated in figure 7.5(a) to have a single large peak (either negative or positive) whereas the for a time delay figure 7.33(a) shows that it exhibits two large peaks.

The regression plots also reveal interesting results. Figure 7.34 and figure 7.35 reveal a k_1, reaching a point greater than 1. Although the regression techniques developed are primarily employed to target gain changes, troubleshooting revealed that due to a large shift between the model and plant due to a delay change, large k_i values are possible. This can be explained intuitively by referring to figure 7.36, a typical response to a pulse input.

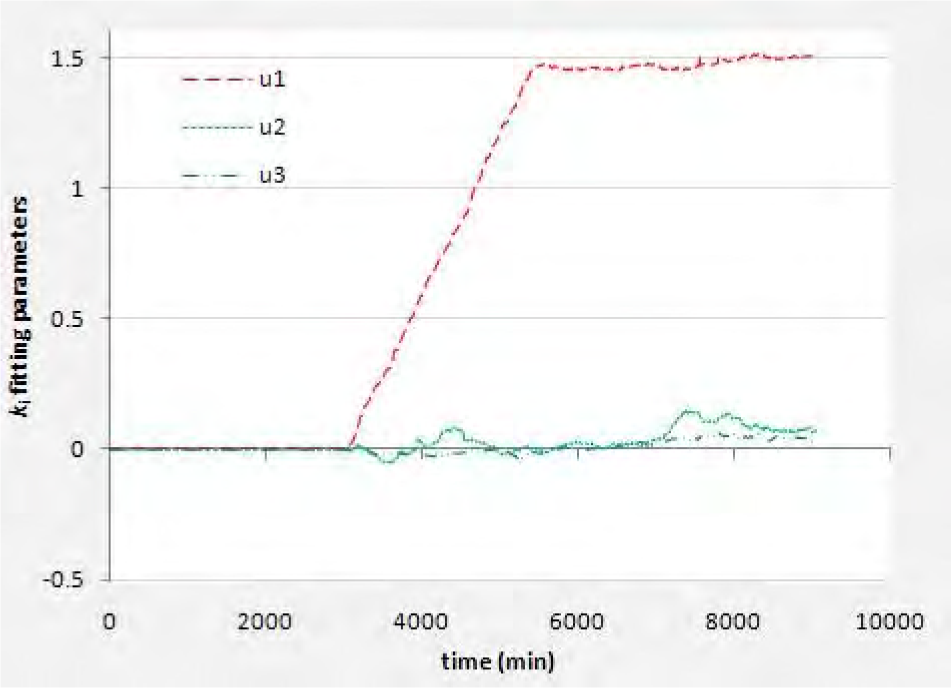


Figure 7.34: Moving Window Regression plot – time delay mismatch

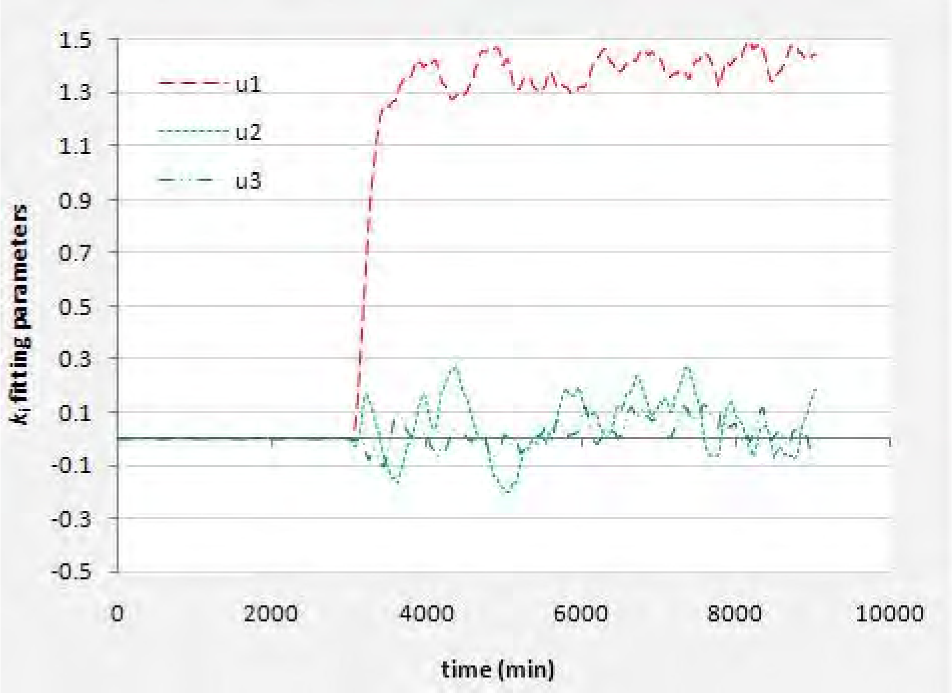


Figure 7.35: Kalman Filter parameter estimation plot – time delay mismatch

In figure 7.36, a typical response to a pulse input together with a shifted response to the same input is shown. Points labeled (a), (b) and (c) represent lag points that are used in the computation of an equation error form. Points x, y and z are representative of the output response at points (a), (b) and (c) respectively. The table below shows the approximate values from the plots depicted below as well as the relative gain at these points. The gain is defined as the output relative to the input. They are negative due to a -10 % change in the input.

Table 7.8:
Summary of intuitive results for a shifted response

	X	Y	Z
Output - Typical Response	58	35	22
Output - Shifted Response	68	68	60
Typical Response Gain (output/input)	**-5.8**	**-3.5**	**-2.2**
Shifted Response Gain (output/input)	**-6.8**	**-6.8**	**-6.0**

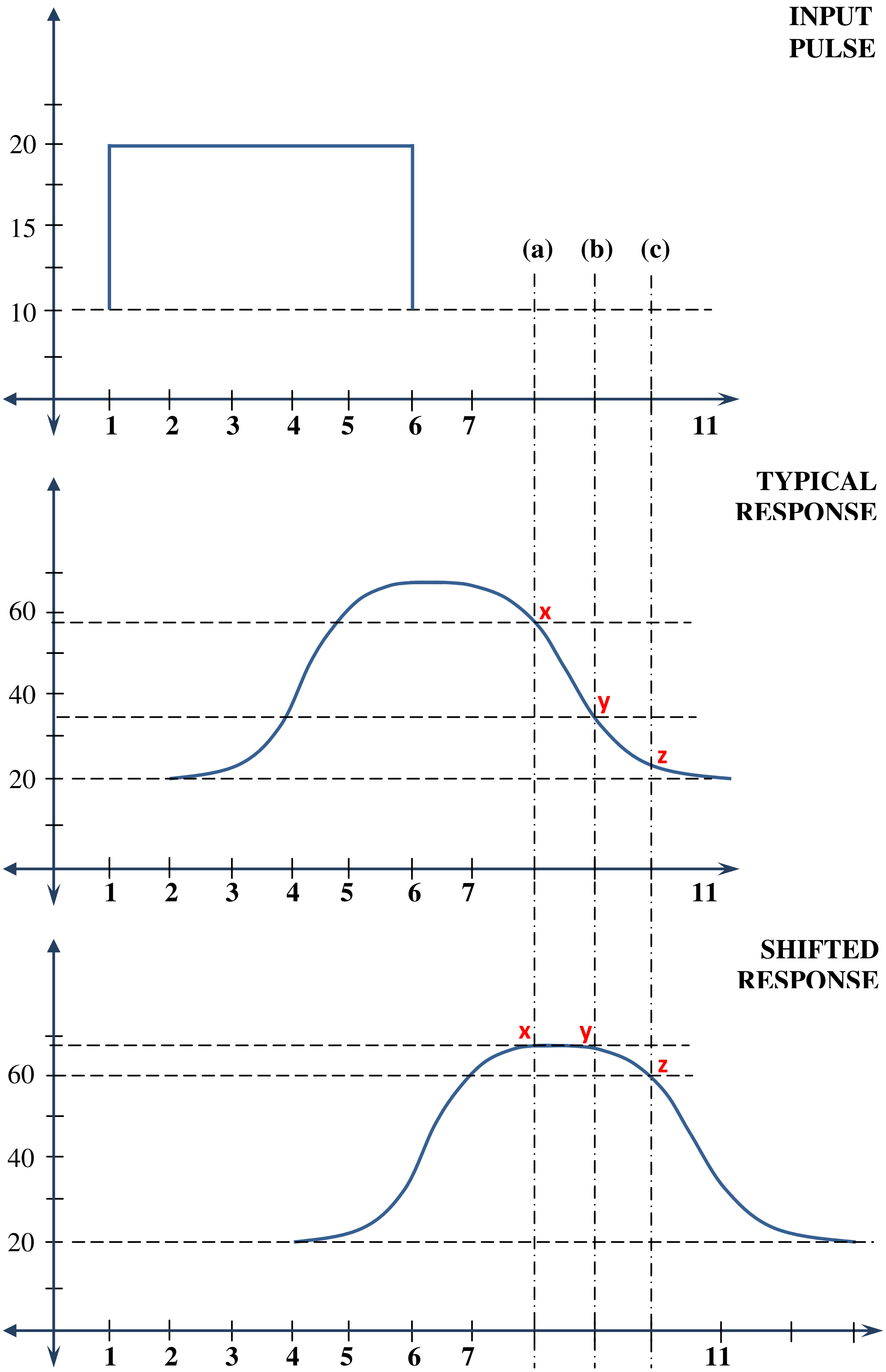


Figure 7.36: Typical response plots to a pulse input

The relative gains in table 7.8 for a shifted response are higher in magnitude than that for a typical response. The equation-error form finds *k* values that relate to the input. In this case it will be equal to the difference in the gain-magnitudes that can be much larger than unity.

7.2. Continuous Stirred Tank Heater

For this scenario, a different unit is used as the models possess higher order lag polynomials in the denominator and numerator. A continuous stirred tank heater (CSTH) is used for this aspect of the study. A simple schematic of the CSTH system is shown below:

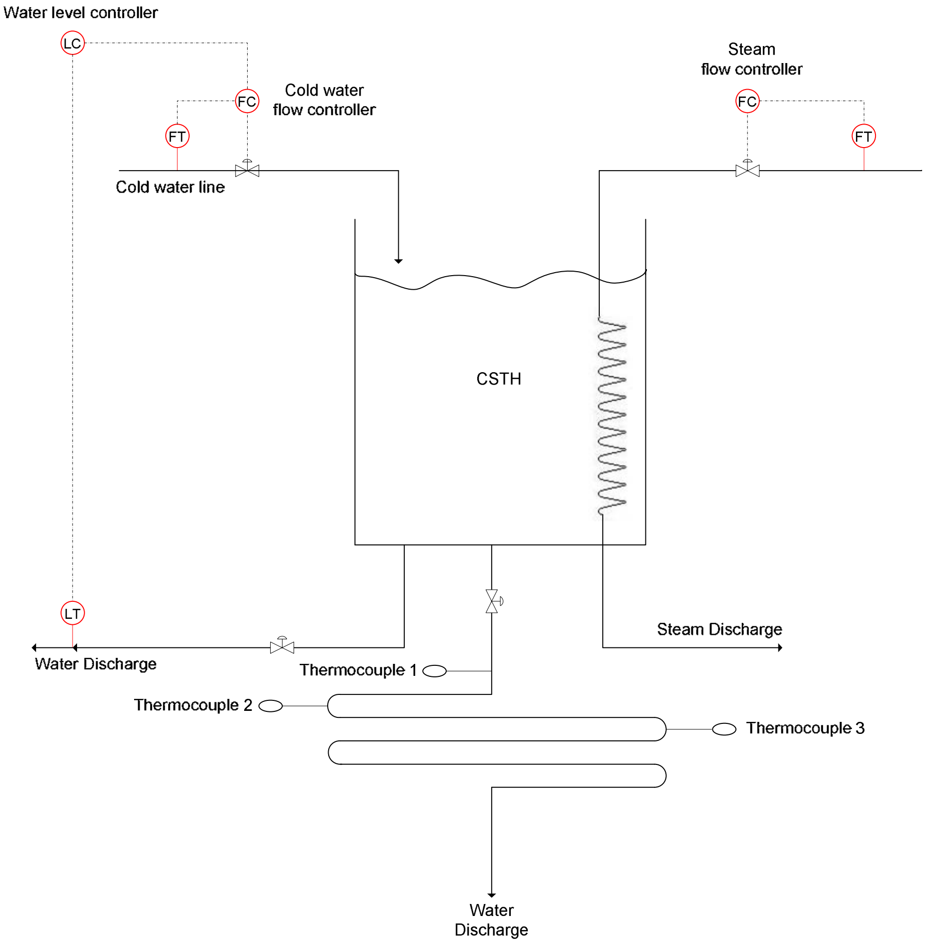


Figure 7.37: Simple schematic of the CSTH system (Thornhill et al., 2008)

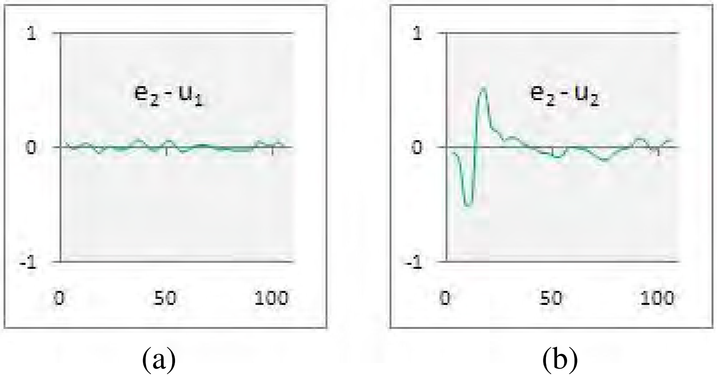
The CSTH system has two inputs; cold water (MV1) and steam (MV2); and two outputs; water level (CV1) and outlet water temperature (CV2). The models relating the outlet water temperature to both MVs is given below (time units – minutes):

$$y_2(s) = \frac{1.028}{20.016s+1} e^{-17.59s} u_1 + \frac{11.59(0.0295s+1)}{240.4s^2+3.6s+1} e^{-8.77s} u_2 \tag{7.5}$$

The sampling interval was found to be <u>3.18</u> minutes and the inputs were modeled as <u>ML</u> (multi-level) signals. The idea behind this case study is to illustrate the effect of the higher order coefficients found in the lag polynomials on the fitting parameters k_i.

<u>7.2.1. Mismatch in higher order coefficient in denominator</u>

For this test, emphasis is placed on model 2 (MV2 – CV2) as it contains higher order polynomials in both the denominator and numerator. A gain underestimated by 50% would give a k_i value of -1.00. Suppose now that the coefficient 240.4 is increased by a factor of 2, the results obtained are shown below:



(a) (b)

Figure 7.38: Correlation plots for denominator coefficient mismatch

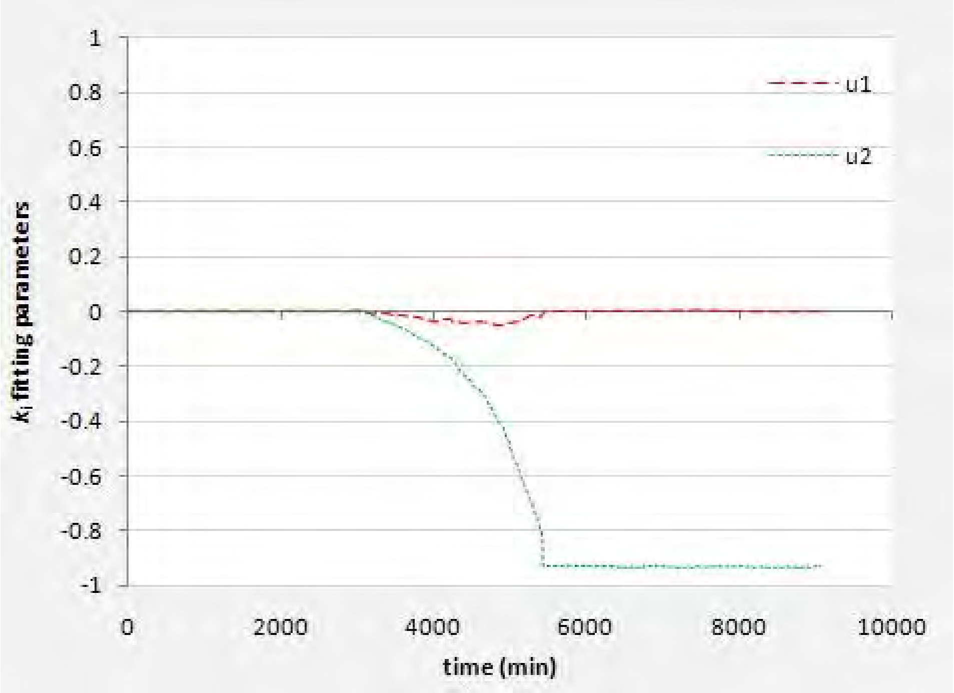


Figure 7.39: Moving Window Regression plot – denominator coefficient mismatch

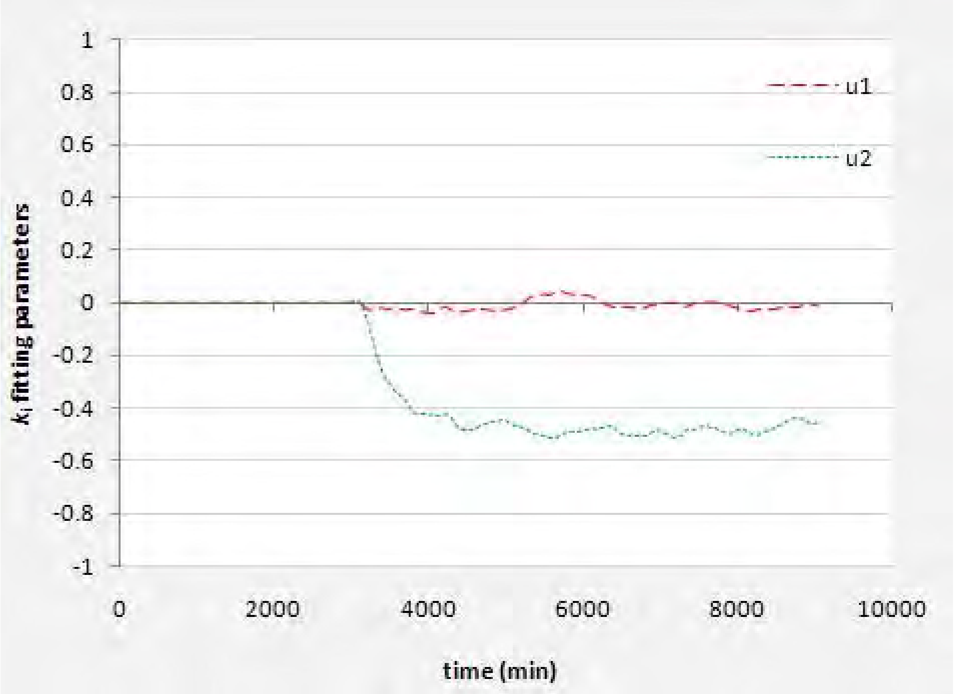


Figure 7.40: Kalman Filter parameter estimation plot – denominator coefficient mismatch

Table 7.9:
Summary of results for denominator coefficient mismatch

	k_1	k_2
EXPECTED k_i FOR UNDERESTIMATED GAIN	0.00	-1.00
MOVING WINDOW REGRESSION	0.0041	-0.9361
KALMAN FILTER	0.0421	-0.5124

The correlations plots reveal the expected result i.e. a mismatch in channel MV2 – CV2. The regression plots on the other hand reveal the results that one would expect for a gain underestimated by 50%. The fitting parameter obtained for k_2 is fairly close to the expected result for such a gain error for the moving window form of regression, whereas the fitting parameter obtained by the Kalman filter regression is approximately half of the moving window value.

The closeness of the fitting parameter obtained for a higher order coefficient found in the denominator which is twice the plant value compared, to that which is obtained for a gain underestimated by 50%, can be explained by considering the following model form:

$$y(s) = \frac{K(cs+1)}{as^2 + bs + 1}u \tag{7.6}$$

In this 2nd order model coupled with a 1st order lead (same form as model 2 in equation 7.5), coefficients a, b and c are arbitrary constants. Neglecting any time delay mismatch, and substituting a **forward difference approximation** (simplest form of a discrete approximation; see table 5.1) for the Laplace operator, *s*, and assuming a **unit sampling interval**, one obtains the following form for equation (7.6):

$$y(q) = \frac{K\left[c\left(1-q^{-1}\right)+1\right]}{a\left(1-q^{-1}\right)^2 + b(1-q^{-1}) + 1}u \tag{7.7}$$

Expanding all polynomials within the model, the following is obtained:

$$y(q) = \frac{K + Kc + Kcq^{-1}}{1 + b - bq^{-1} + a(1 - 2q^{-1} + q^{-2})}\, u$$

$$= \frac{K(c+1) + Kcq^{-1}}{1 + b - bq^{-1} + a - 2aq^{-1} + aq^{-2}}\, u$$

$$= \frac{K(c+1) + Kcq^{-1}}{(1 + a + b) - (b + 2a)q^{-1} + aq^{-1}}\, u \qquad (7.8)$$

The appropriate discrete model form is obtained by dividing throughout by (1 + *a* + *b*):

$$y(q) = \frac{\dfrac{K(c+1)}{(1+a+b)} - \dfrac{Kc}{(1+a+b)} q^{-1}}{1 - \dfrac{(b+2a)}{(1+a+b)} q^{-1} + \dfrac{a}{(1+a+b)} q^{-2}}\, u \qquad (7.9)$$

The following assumptions can be made:

$(1 + a + b) \approx a$ if $a >> (b+1)$

$K + Kc \approx K$ if $K >> Kc$ i.e. if c is very small

$b + 2a \approx 2a$ if a $>> b$

Equation (7.9) thus simplifies to:

$$y(q) = \frac{\dfrac{K}{a} - \dfrac{Kc}{a} q^{-1}}{1 - \dfrac{2a}{a} q^{-1} + \dfrac{a}{a} q^{-2}}\, u \qquad (7.10)$$

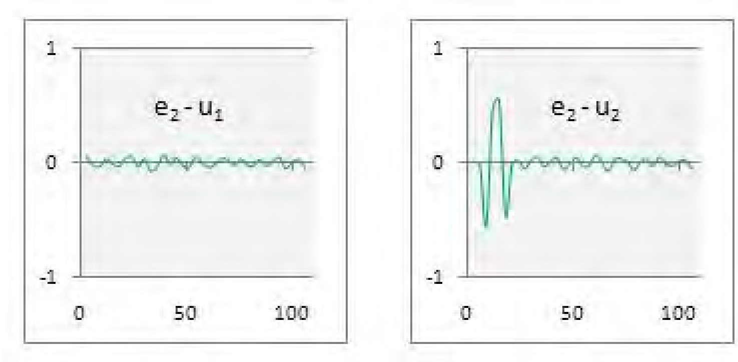
$$y(q) = \frac{\frac{K}{a} - \frac{Kc}{a} q^{-1}}{1 - 2q^{-1} + q^{-2}} u$$

$$y(q) = \frac{\frac{K}{a}\left[1 - cq^{-1}\right]}{1 - 2q^{-1} + q^{-2}} u \tag{7.11}$$

From equation (7.11), it can be seen that if the coefficient a is significantly large, it becomes directly linked to the gain of the system. Equation (7.5) shows that a = 240.4 is significantly larger than that of the other coefficients.

7.2.2. Mismatch in coefficient in numerator

Suppose that mismatch was added in such a way that coefficient c (c = 0.0295) in channel MV2 – CV2 from equation (7.5) is underestimated by 50%. As a result of this factor being much smaller than the actual gain (500 times smaller), one would expect very small fitting coefficients. The results pertaining to this scenario are shown below, in figures 7.41 – 7.43:



(b) (b)

Figure 7.41: Correlation plots for numerator coefficient mismatch

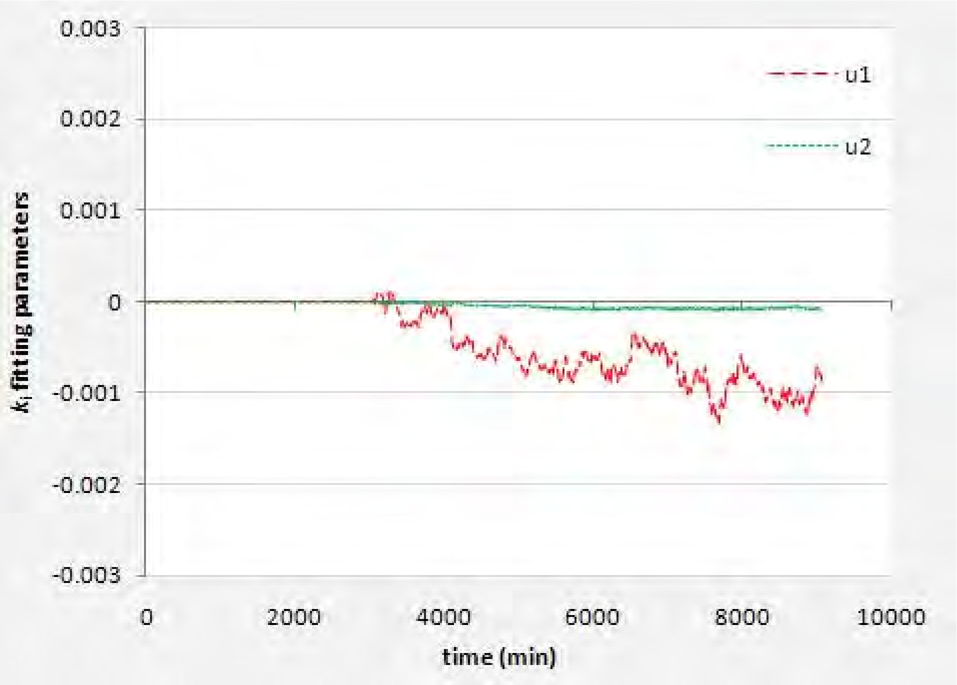


Figure 7.42: Moving Window Regression plot – numerator coefficient mismatch

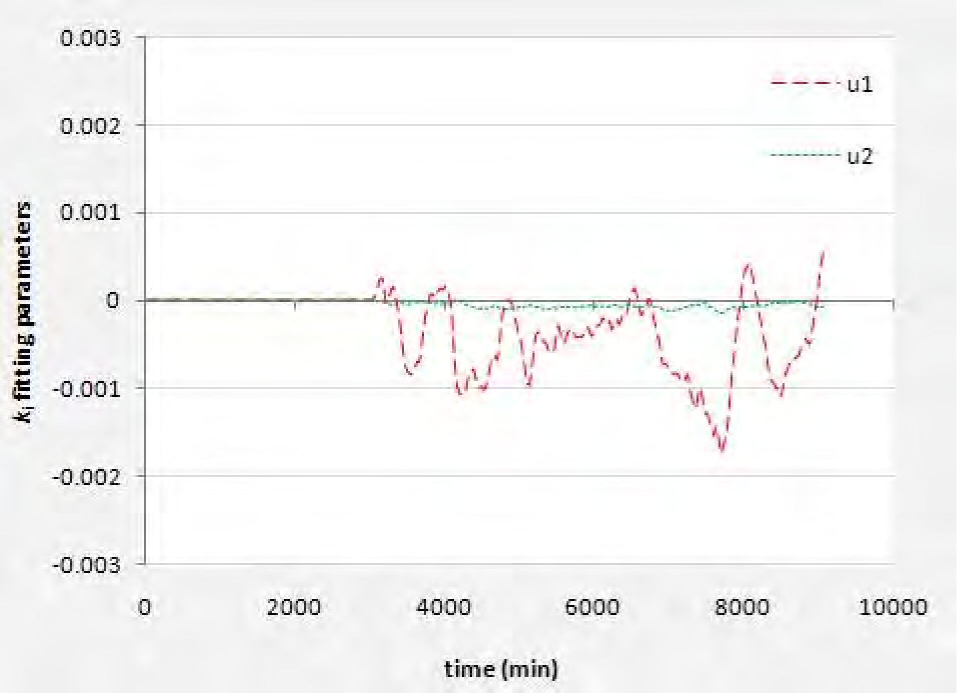
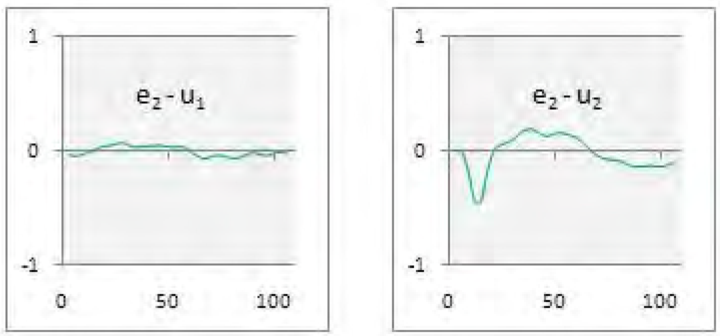


Figure 7.43: Kalman Filter parameter estimation plot – numerator coefficient mismatch

7.2.3. Mismatch in lower order coefficient in denominator

Coefficient b (b = 3.6) in channel MV2 – CV2 is underestimated by 50%. One would expect negligible fitting parameters for this scenario as equation (7.11) shows that coefficient is not present in the models discrete form for large a coefficients. This is shown by figures 7.45 and 7.46.



(c) (b)

Figure 7.44: Correlation plots for lower order denominator coefficient mismatch

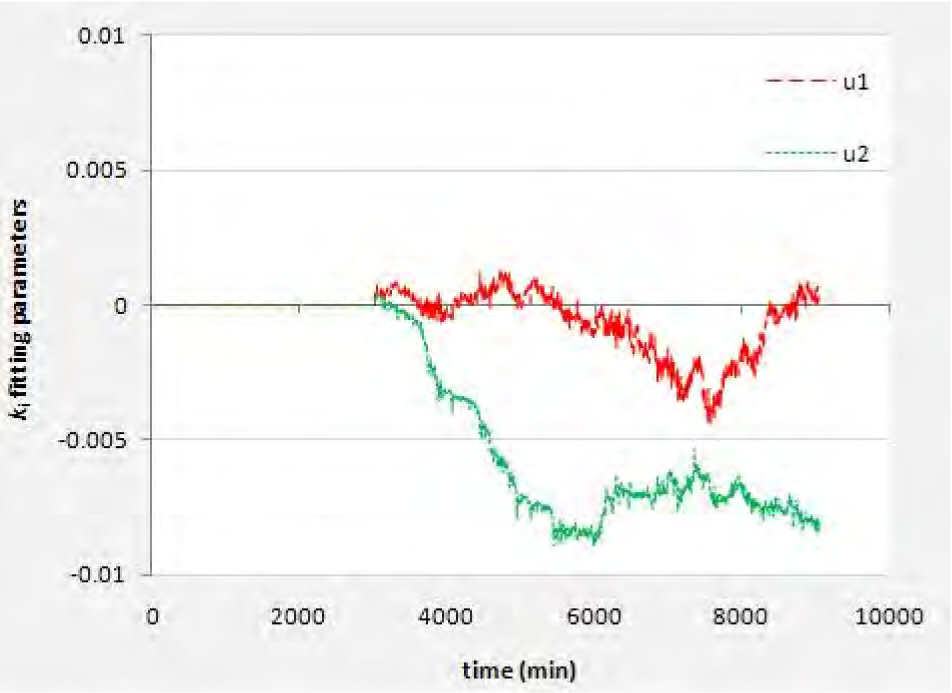


Figure 7.45: Moving Window Regression plot- lower order denominator coefficient

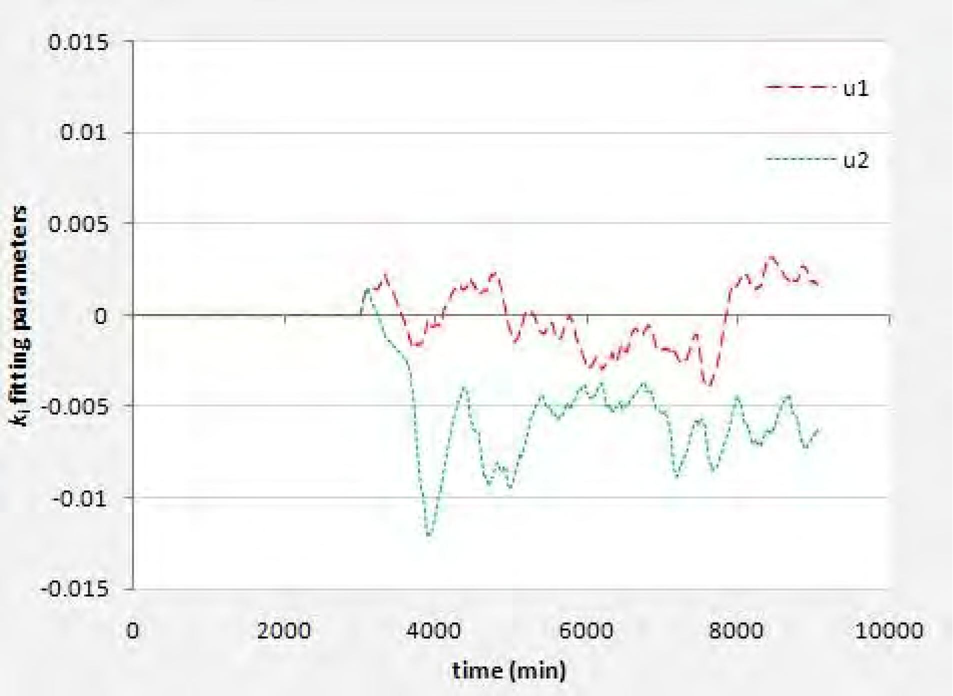


Figure 7.46: Kalman Filter parameter estimation plot- lower order denominator coefficient

7.3. RULES FOR DETECTING SIGNIFICANT MISMATCH

In order to make the provision of model diagnostic results better suited for interpretation by a maintenance engineer monitoring a Model Predictive Controller, a number of strategic warning mechanisms are implemented within the MATLAB ® program. These warning mechanisms provide information when certain limits are violated. These warning mechanisms include increased error variance, correlation coefficients violating certain confidence bounds as well as regression coefficients violating set confidence levels.

7.3.1. Error Variance

The benchmark for error variance depends on the variance set for the description of the noise model in an ARX model equation; see equation (3.28). Badwe et al., (2009) describes the noise model in their simulation studies as having a variance of 0.0075. For this work, the noise model is given a variance of 0.01. In the absence of any MPM, the noise model is shown to be equal to the model error obtained:

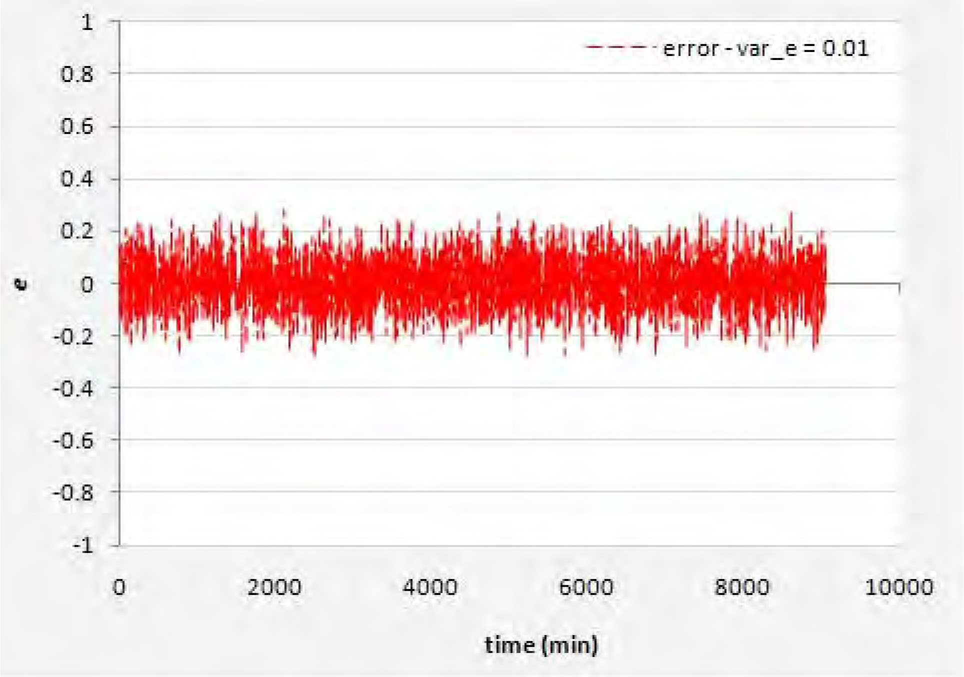


Figure 7.47: Error plot in the absence of MPM

This is regarded as the benchmark for optimal control. If for any reason, the variance of this error is shown to be greater than the benchmark standard variance, a first warning is issued, informing the user that they may be a modeling mismatch present, or other factors which may cause the error variance to increase:

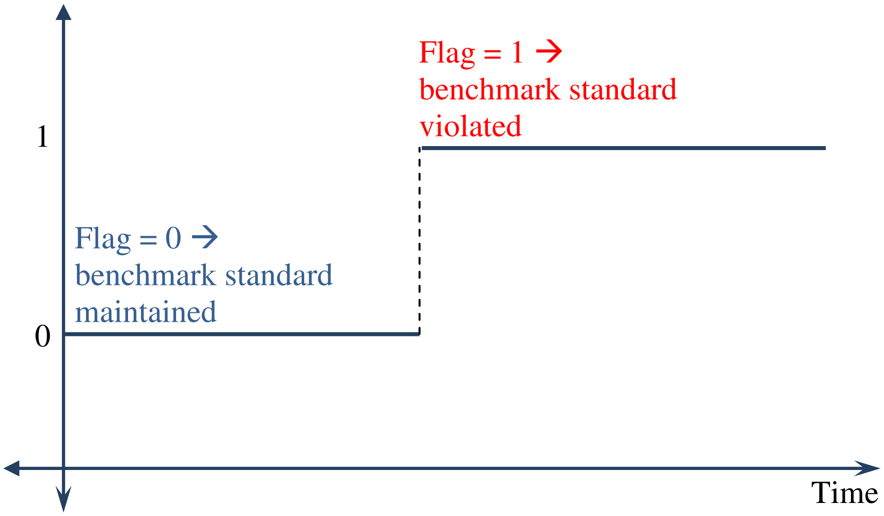


Figure 7.48: Warning Flag mechanism for increased variance

7.3.2. Cross – Correlation Confidence Bounds

Webber and Gupta (2008) suggest that 95% confidence bounds (range of -0.05 to 0.05 for correlation coefficients) are suitable for the detection of significant MPM in Cross – Correlation analyses. For this work, larger confidence bounds are used as significant mismatches are found to be in a correlation coefficient range outside -0.5 to 0.5. If it is shown that a correlation plot reveals that the coefficients are outside the range of -0.5 to 0.5, a second warning is issued which implies the presence of MPM. These flags are issued for each input – output pairing. Coefficients within range do not necessarily imply that no MPM is present. As shown in section 7.1.4, the influence of noise signals reduce the magnitude of the correlation coefficients notably even in the presence of MPM.

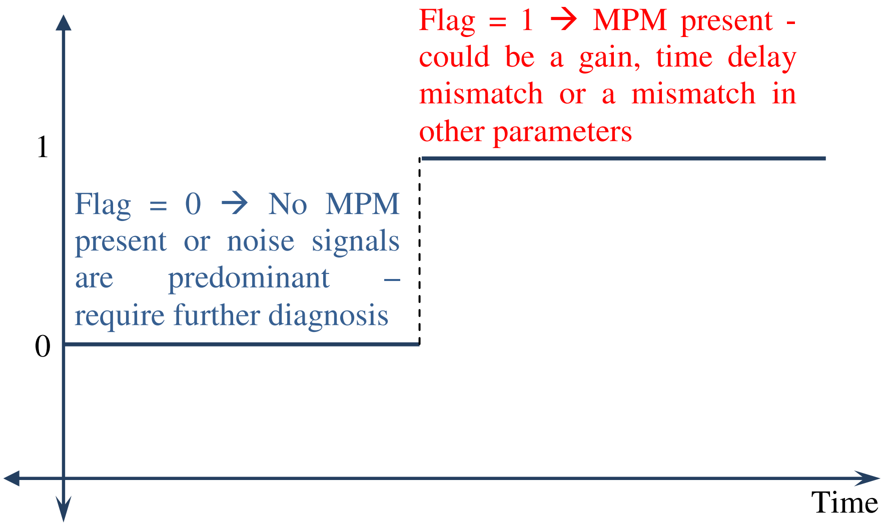

Flag = 1 → MPM present - could be a gain, time delay mismatch or a mismatch in other parameters
1
Flag = 0 → No MPM present or noise signals are predominant – require further diagnosis
0
Time

Figure 7.49: Warning Flag mechanism for the violation of correlation bounds

7.3.3. Regression fitting parameters confidence bounds

These bounds can be set by the user depending on how much a gain may be mismatched in a specific application. For example small bounds are required to be set for a system that requires a temperature to be maintained in a tight range. In the case of maintaining the level in a tank, larger bounds may be used as the tank may be operating such that it does not overflow or run dry. For this work confidence levels, for the ratio of equation error to plant value of the RHS of equation (5.25), ranging from -1 to 1 is used. If a parameter is shown to be outside this range, a third warning is issued for a gain mismatch. Another range outside of -1 to 1 is incorporated in this mechanism to issue a warning for the presence of a time delay mismatch (see section 7.1.5).

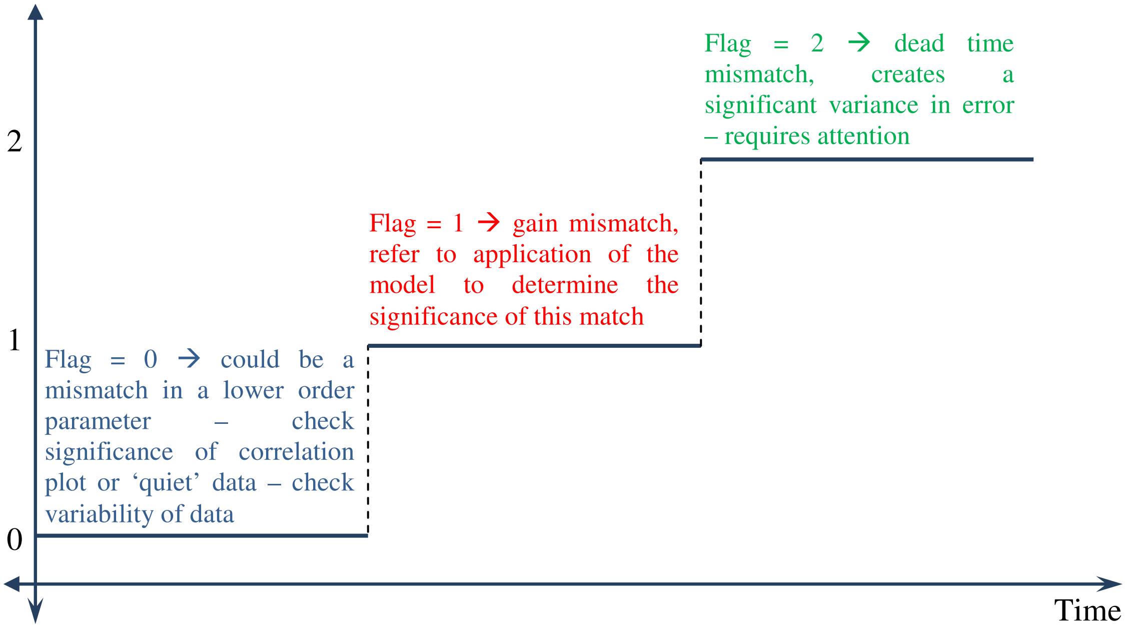

Flag = 2 → dead time mismatch, creates a significant variance in error – requires attention
2
Flag = 1 → gain mismatch, refer to application of the model to determine the significance of this match
1
Flag = 0 → could be a mismatch in a lower order parameter – check significance of correlation plot or 'quiet' data – check variability of data
0
Time

Figure 7.50: Warning Flag mechanism for the violation of regression parameter bounds

If the above warning mechanism issues a warning for a gain mismatch, then another mechanism is employed on the multiplicative factors to determine the significance of the mismatch. These bounds may be set by the user:

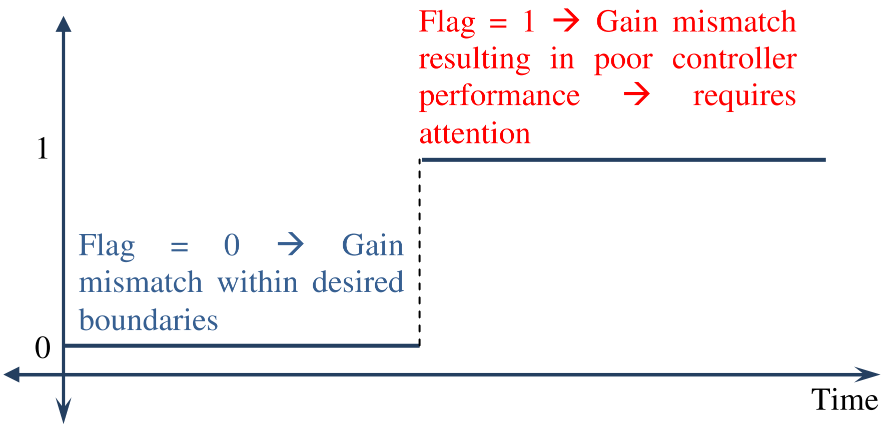

Flag = 1 → Gain mismatch resulting in poor controller performance → requires attention
1
Flag = 0 → Gain mismatch within desired boundaries
0
Time

Figure 7.51: Warning Flag mechanism for the violation gain bounds

These bounds depend on the application of the model as well as whether the controller is performing well under the circumstances wherein a mismatch is present. The typical bounds applied in this thesis ranges from 0.7 to 1.5.

7.3.4. Variability of data

High variability of the input data is paramount to the efficacy of this methodology. In the case where variability is low, the resulting fitting parameters are shown to be less significant. Thus another warning mechanism is developed which issues a warning when the variability of the data is low. This informs the user that he should switch to a non – differential, open – loop prediction form in order to obtain a reasonable result (see figure 7.50).

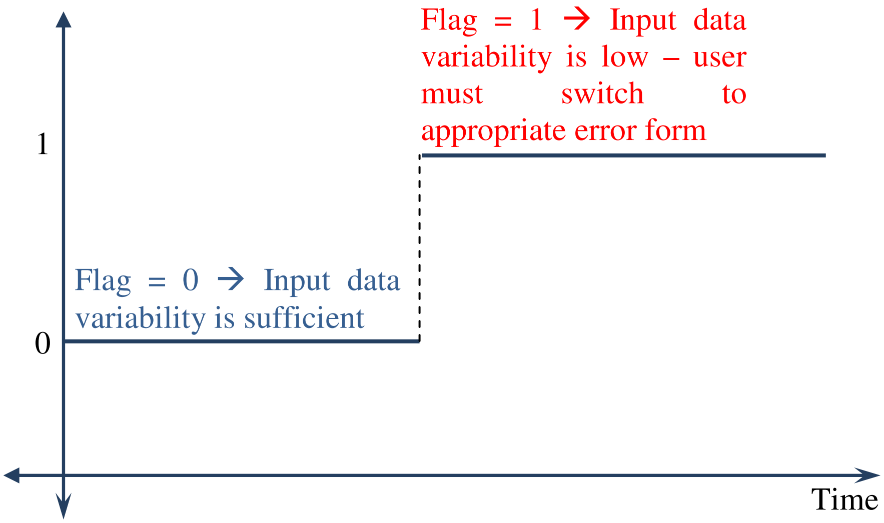

Flag = 1 → Input data variability is low – user must switch to appropriate error form
1
Flag = 0 → Input data variability is sufficient
0
Time

Figure 7.52: Warning Flag mechanism for the variability of input data

7.3.5. Noise levels

Noise levels disrupt the efficacy of the methodology tremendously. In the presence of significant noise levels, figure 7.48 may exhibit random switching between flags, making it difficult to distinguish the source of MPM. If, however the MPM dominates over the noise signal over a range of time, one may obtain a reasonable diagnosis.

7.3.6. Diagnosis Table

Each of the warning mechanisms shown above are combined to provide an overall diagnosis displayed in the table below:

Table 7.10:
Diagnosis table for Model validation methodology

Error Variance	**Cross – Correlation**	**Regression parameter bounds**	**Significant factor bounds**	**DIAGNOSIS**
0	0	0	0	• Model Quality indication → good • No MPM present
1	1	1	0	• MPM – gain mismatch detected • Not significant (user discretion)
1	1	1	1	• MPM – gain mismatch detected • Controller performance affected
0	1	0	0	• MPM – lower order polynomial • Coefficients mismatch
1	1	2	X	• MPM – time delay mismatch • Model requires attention
1	0	0, 1, 2	X	• Influence of noise • Difficult to distinguish model error

*x – user is required analyze the results carefully in order to make a diagnosis

The maintenance engineer may use this table when the models used in a Model Predictive Controller are being tested. In the case where the input data has low variability, a warning is issued to the user to change the error detection form to a form which is suitable for cases of ‘quiet’ data i.e. non - differential equation error with fixed offset.

7.3.6.1. Application of the diagnosis table

The validity of the above diagnosis table is tested using the results obtained for section 7.1.4.2. This scenario is representative of a typical industrial process where the influence of noise on a plant signal is inevitable. Suppose that one does not know that a mismatch relating to channel MV1 – CV1 for the Shell Heavy Oil Fractionator. In this case, careful observations for each warning mechanism are needed in order to provide a diagnosis.

The first step is to view the warning mechanism related to the error variance:

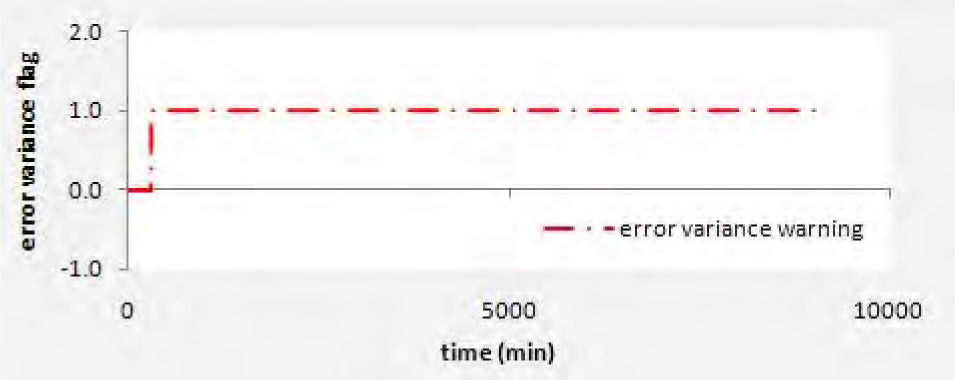


Figure 7.53: Error variance warning flag

This flag indicates that the variance of the error obtained is higher than the benchmark in the case of no model mismatch or excessive noise signals. The next step is to view the correlation warning flags which provide an indication of which input – output pairing may be responsible for the error obtained.

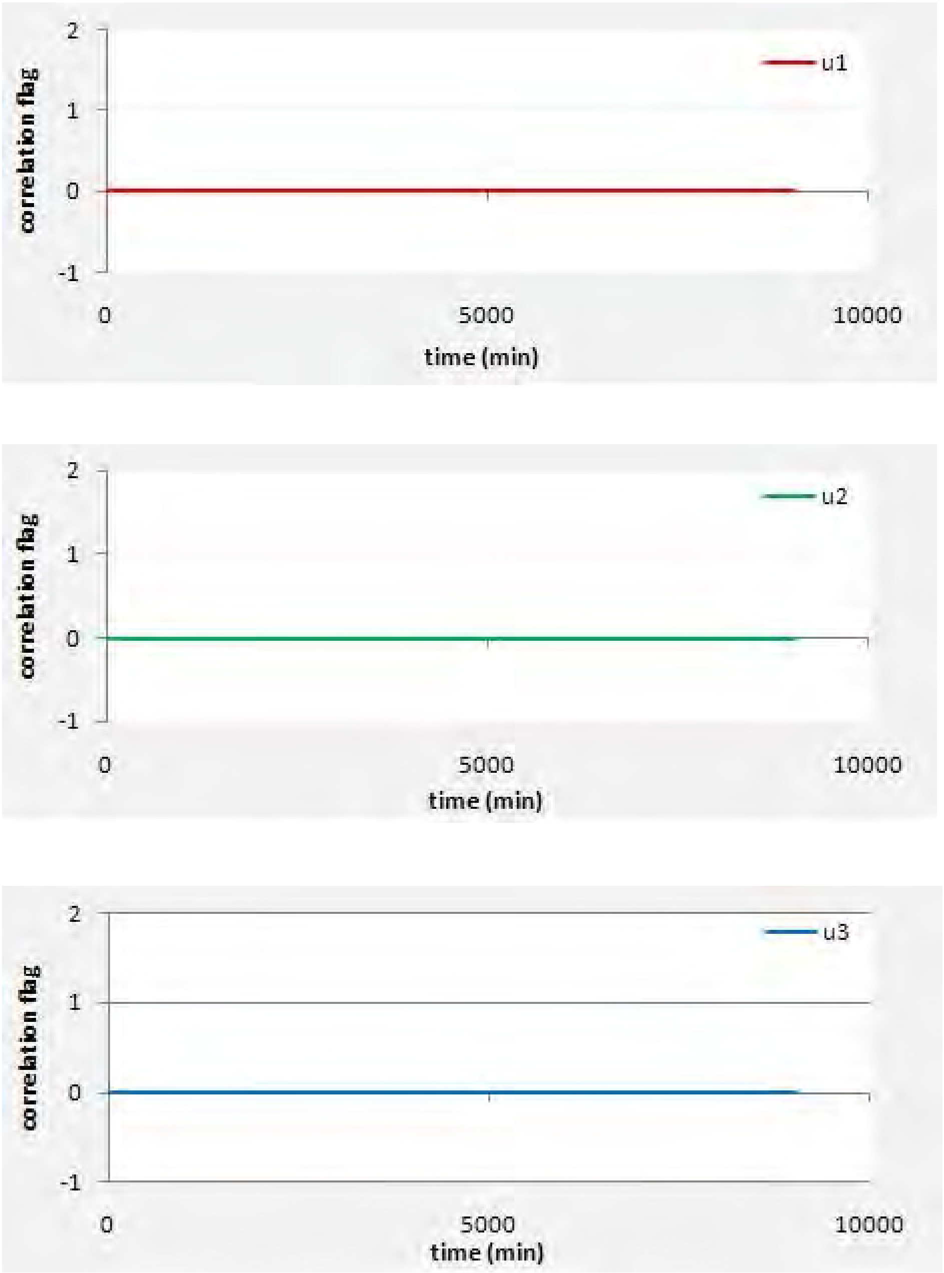


Figure 7.54: Correlation coefficients bounds violation flags

No flag is issued for any of the input – output pairings involved (figure 7.54). One may deduce that the error obtained is influenced by the presence of noise, which subsequently reduces the magnitude of the correlation coefficients. The next step is to determine whether the regression fitting parameters are within the -1 to 1 range. Flag mechanisms are set up for both moving window regression and the Kalman filter:

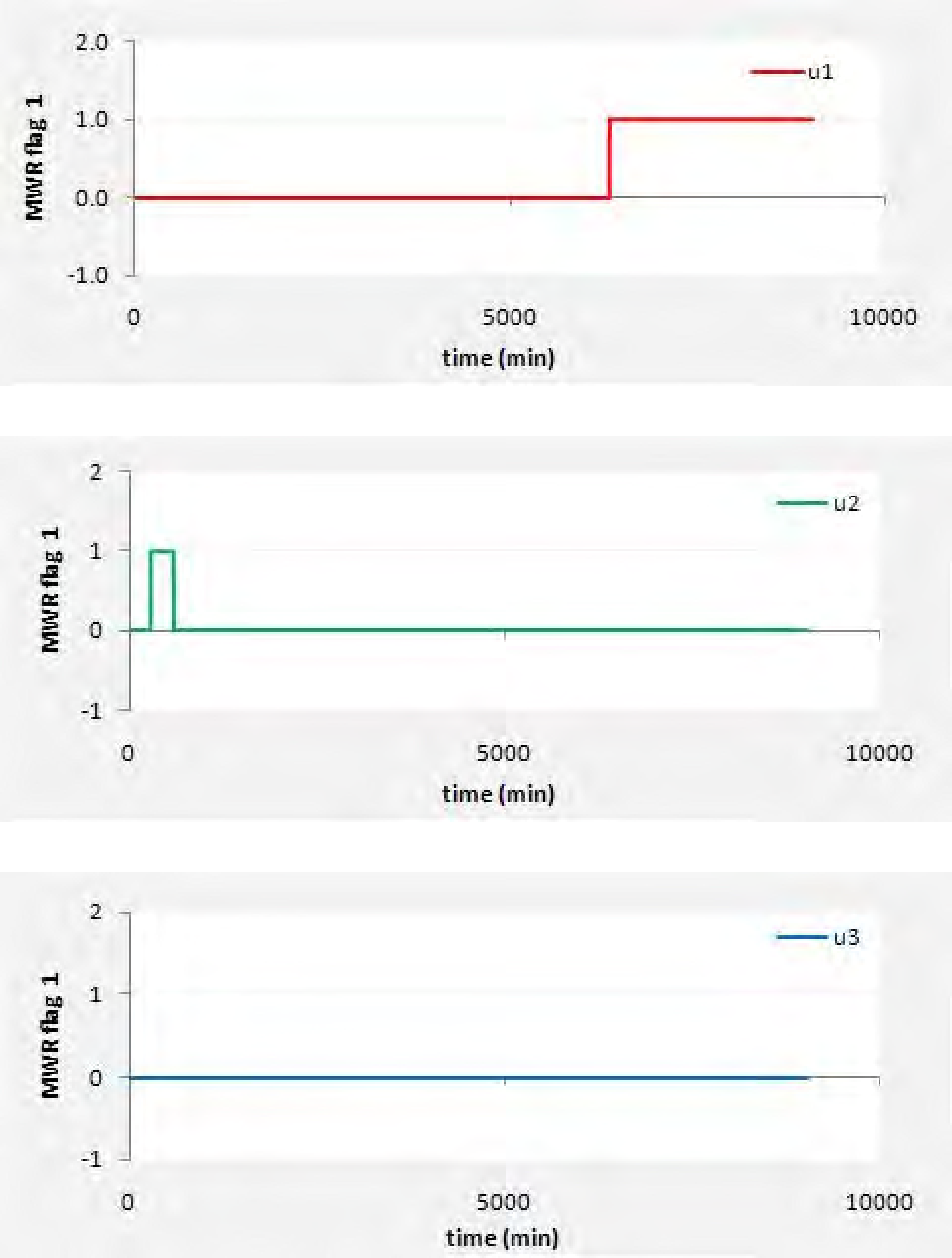


Figure 7.55: MWR parameter flag 1

A flag warning of 0 implies that the parameters are in the desired range. A flag warning of 1 implies that the parameters lie outside the desired bounds. In this case, k_1 is found to be within the -1 to +1 range but outside its desired bounds of -0.3 to +0.3.

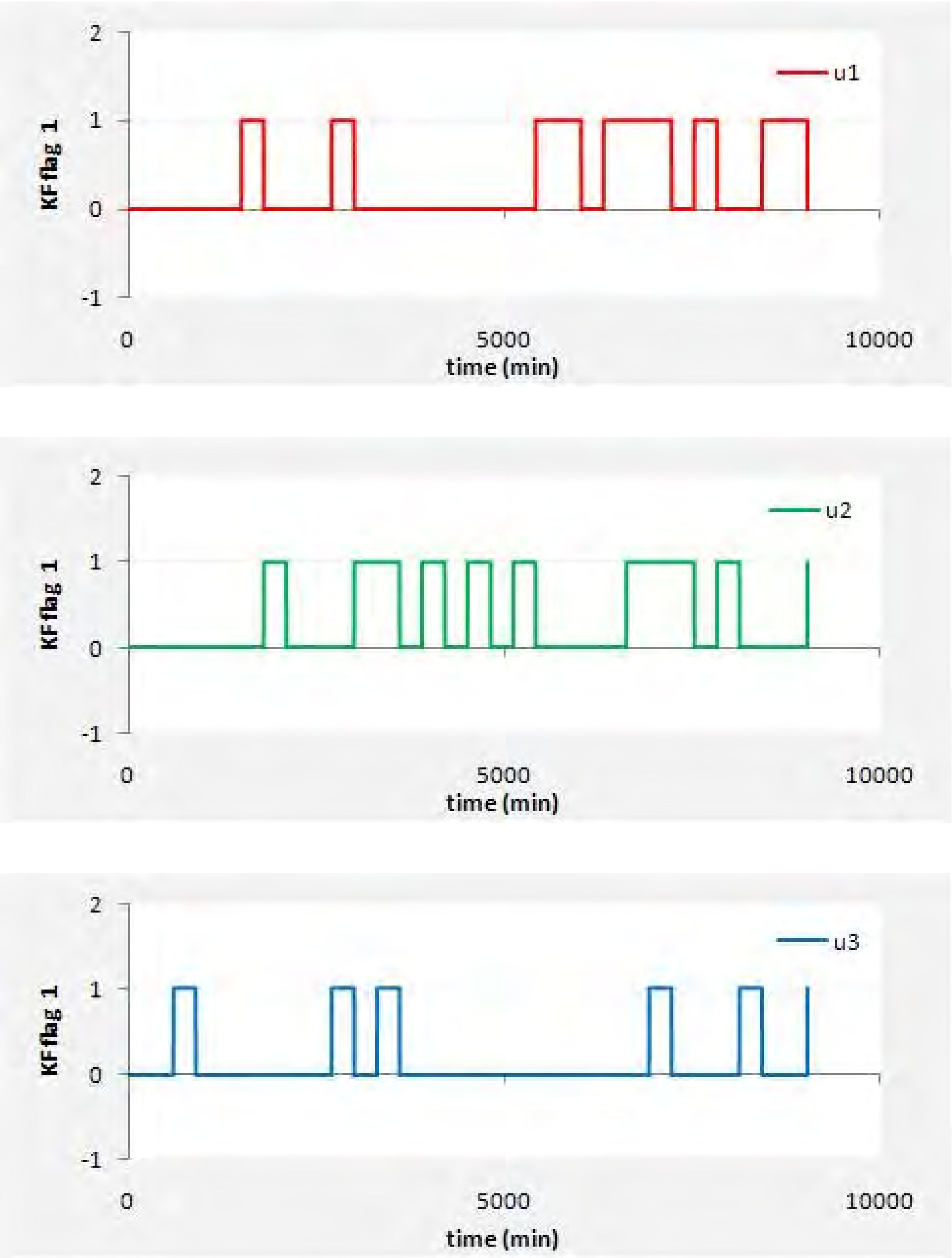


Figure 7.56: Kalman Filter parameter flag 1

In the case of the Kalman filter, haphazard switches between flag options are due to the susceptibility of the Kalman filter to disruption by noise. Nevertheless, all flags above indicate that the parameters all lie in the range -1 to +1, ruling out the possibility of a time delay mismatch. Also due to the fact that a flag warning of 1 implies a parameter fitting outside the range of -0.3 to +0.3 a mismatch in lower order parameters is subsequently ruled out. It can thus safely be assumed that the error is due to a gain mismatch or purely the influence of noise.

The next mechanism focuses on the multiplicative factors. A range of 0.5 (the model gain is 50% smaller than the plant gain) to 1.3 (the model gain is 30% higher than the plant gain). A flag warning of 1 would imply a significant mismatch in the gain, if and only if the flag remains constant for long period of time:

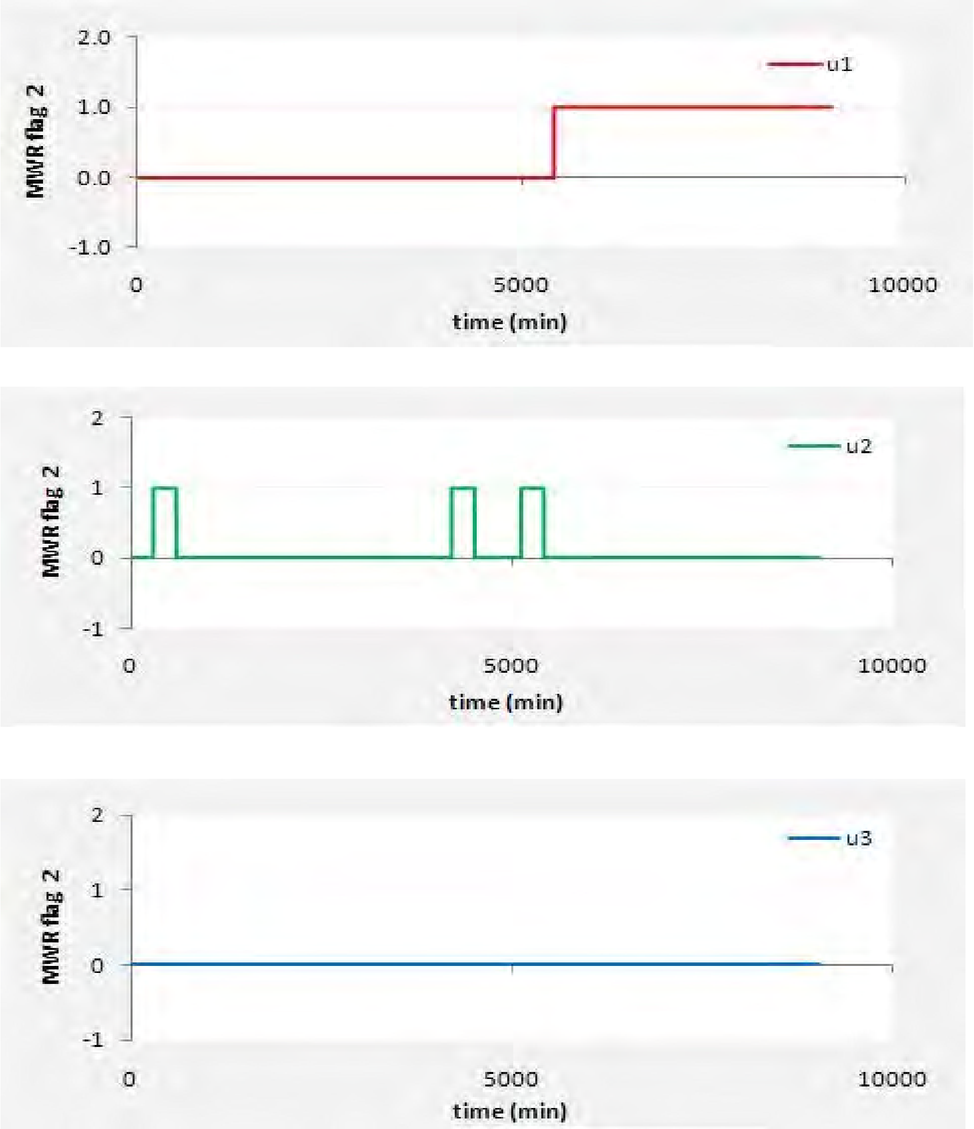


Figure 7.57: MWR parameter flag 2

From the figure 7.57 above one can deduce that there exists a significant gain mismatch due to the persistent flag warning of 1 issued for k_1. The random switch in flag warning depicted for k_2 is primarily due to the influence of noise.

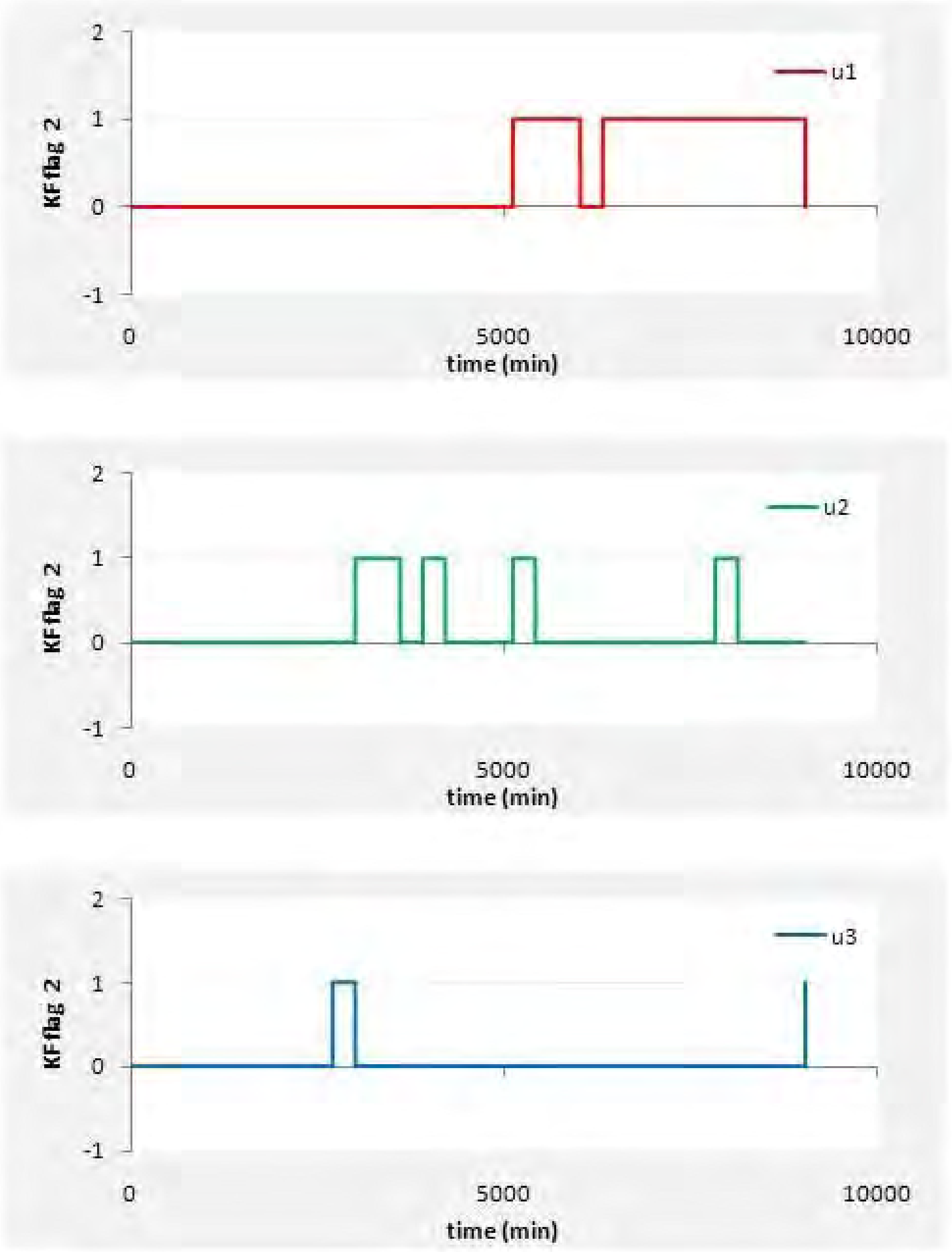


Figure 7.58: Kalman Filter parameter flag 2

The Kalman filter also lies in agreement with the flags issued for the moving window regression. The complete diagnosis in this case is as follows: a significant gain mismatch exists in channel MV1 – CV1 (see figure 7.55 and 7.56); there exist some level of noise signal influence due to small correlation coefficients (see figure 7.52); thus the error formed is due to a gain mismatch in channel MV1 – CV1 as well as due to the influence of an unmeasured noise signal.

7.3.7. Model Set reduction

The technique of reducing model sets in order to determine true and significant model mismatches is an important one especially in the case where plant input and output signals are disrupted by the influence of noise. It is important to clarify that the purpose of reducing model sets is to look for the situation where one of the sets produces no correlated error, and merely brings more noise that could mask correlated error in the other set. Those inputs that are disrupted by noise compound the error that forms and results in spurious mismatch detections. Reducing model sets simply means holding an input constant. In differential mode this means the differential input is zero, effectively leaving out the model related to that specific input. When one decides to reduce the model set, the open – loop prediction form for the model output needs to be employed as one may not ‘borrow’ previous plant output values, see section 5.2, equation 5.24. The differential mode eliminates the effect of the offset. Model set reduction minimizes the effect of noise signals and subsequently provides adequate results (see section 8.2.1).

CHAPTER 8 INDUSTRIAL CASE STUDY

In the previous chapter, known model mismatches and their expected simulated responses allowed for the investigation of various scenarios that are applicable in industrial applications. This chapter seeks to extend the results and rules for significant mismatch detection to a real plant system. A petrol Debutanizer model, supplied by SASOL together with industrial data, is used to test the validity of the methodology developed in this work. This system contains several Process Variables and the requirement to focus on smaller subsets of models established in chapter 7 is further addressed. For this purpose CV2 is considered initially to develop this requirement. The results pertaining to the remaining CVs are documented and pertinent observations are discussed.

8.1. PETROL DEBUTANIZER

From an operating point of view, the Petrol Debutanizer serves a dual purpose. In petrol mode, it is used as an additional feed debutanizer to the next phase of operation. In diesel mode, it is used to remove light components (C_4 and lighter) from the petrol/diesel product sent to the next phase of operation. Any light components in this product will be flared, since the next phase of operation has no facility to remove or process excess light material.

The petrol debutanizer has several Process Variables. These include 6 CVs, 6 MVs and 6 measured DVs. A simplified Process and Instrumentation Diagram of the system, for which the petrol debutanizer controller is employed, is shown below. All process variables are highlighted within the diagram and listed in the table below. Process variables that are not depicted in the PID below are found in upstream processes.

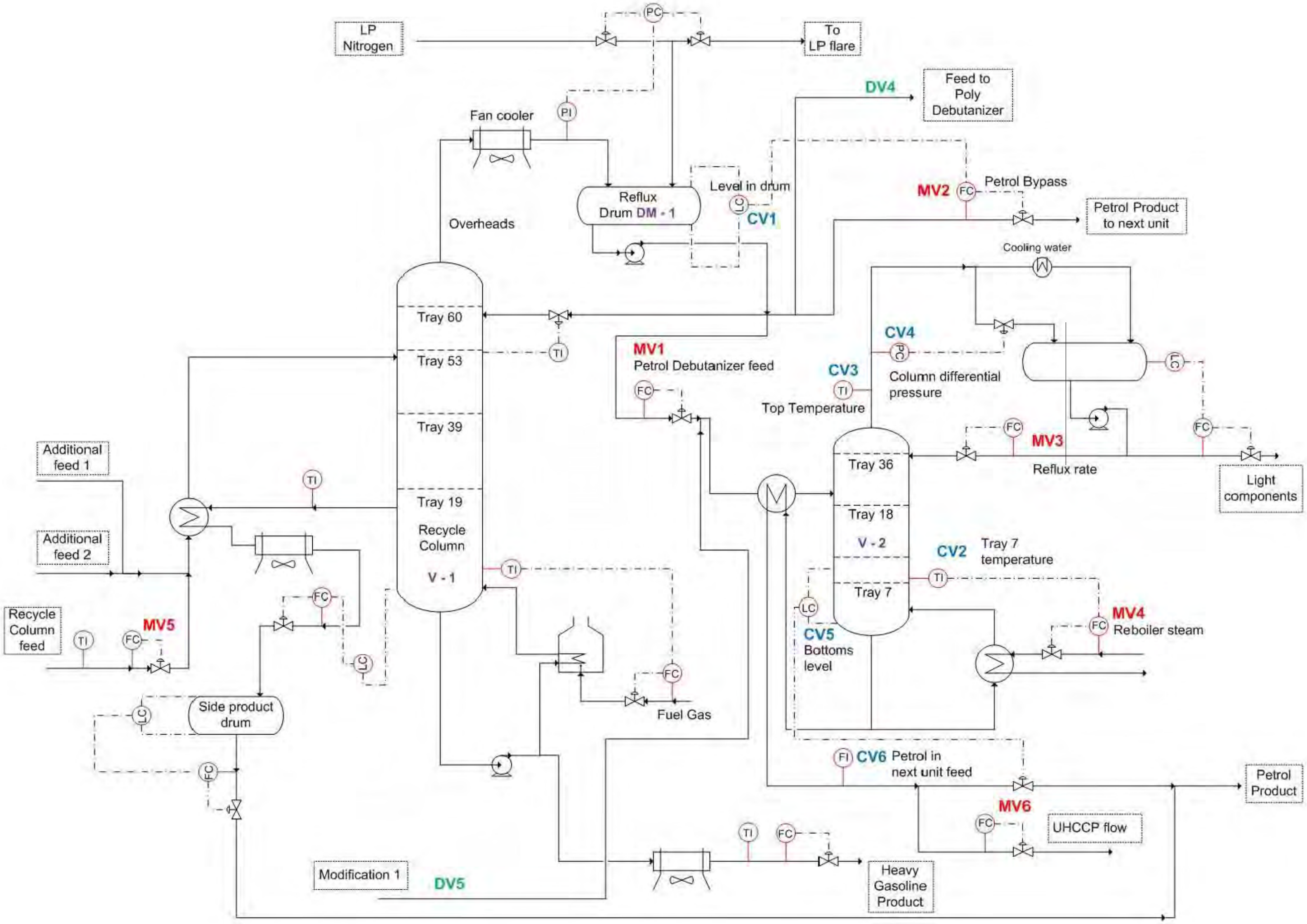


Figure 8.1: Simplified PID of Petrol Debutanizer unit

Table 8.1:
Summary of Process Variables for Petrol Debutanizer unit

CV	Description	MV	Description
CV1	Level in Reflux drum DM -1	MV1	V-2 Column feed
CV2	V-2 Tray 7 temperature	MV2	Petrol Bypass
CV3	V-2 Top temperature	MV3	Reboiler Steam
CV4	V-2 Column Differential Pressure	MV4	V-2 reflux
CV5	V-2 Bottoms level	MV5	V-1 feed flow
CV6	Petrol feed flow to next unit	MV6	UHCCP flow from DVs

DV	Description
DV1	Rectifier 1 bottoms product
DV2	Rectifier 2 bottoms product
DV3	Petrol Recycle 1
DV4	Petrol Recycle 2
DV5	Mod 1
DV6	Mod 2

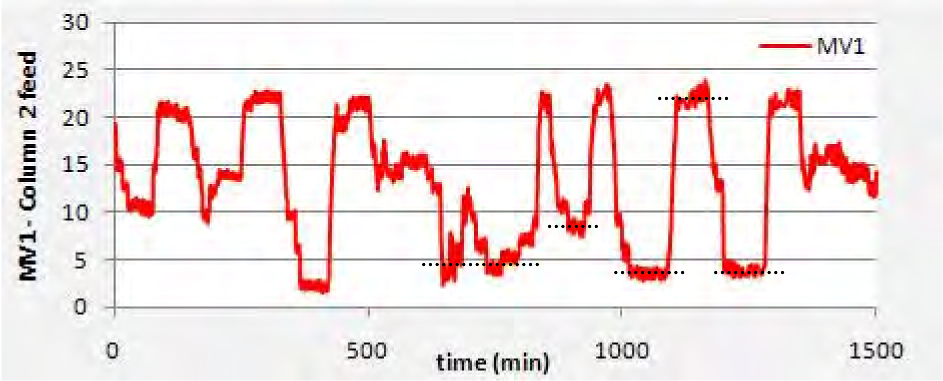
Each Process Variable is labeled in one of three colours in order to differentiate between inputs (MVs and DVs) and outputs (CVs).

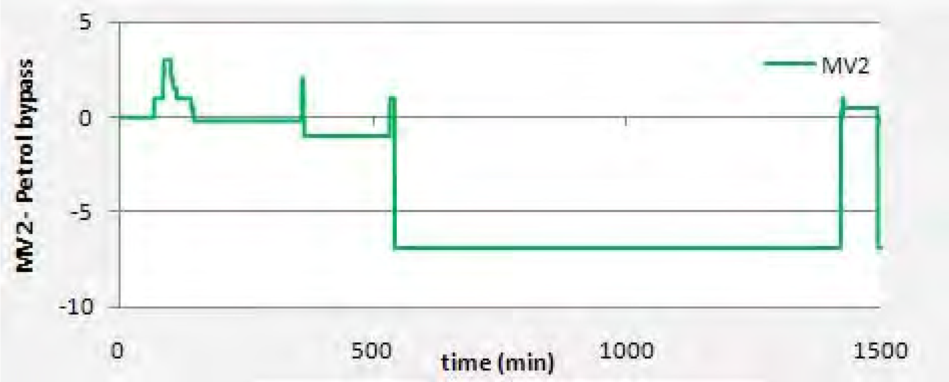
The Model Predictive Controller is primarily employed for the operations of column V-2. The feed to the V-2 column is drawn from DM-1, the reflux drum of V-1. This drum also provides the petrol recycle to the reactors in the next phase of operation through a poly-debutanizer. The feed to V-2 is normally running on flow control, and the level is controlled with a petrol bypass line around V-2 routed directly to the next phase feed tank. The control objectives of the petrol debutanizer controller are to stabilize the level in DM-1, and obtain the desired petrol fraction in bottoms product of column V-2. The optimization objective is to minimize the petrol bypass flow labeled as MV2 in the diagram above.

The model matrix for this particular controller can be found in Appendix B, figures B.1 to B.3. There are no models in channels that are left blank since that specific MV or DV does not affect the corresponding CV. 8264 samples of data were provided by SASOL with a sampling interval of 0.5 minutes (approximately 3 days worth of data). A suitable subset of data (a days worth of data ~ 3000 data points) was chosen from the provided data which encompasses the various scenarios covered in chapter 7.

8.1.1. Manipulated Variables

Each of the 6 MVs present in this system is imperative in achieving the controller's optimization objectives and control objectives. Each MV is represented below. N.B. dotted lines (····) indicate periods of time in which the data is relatively flat.





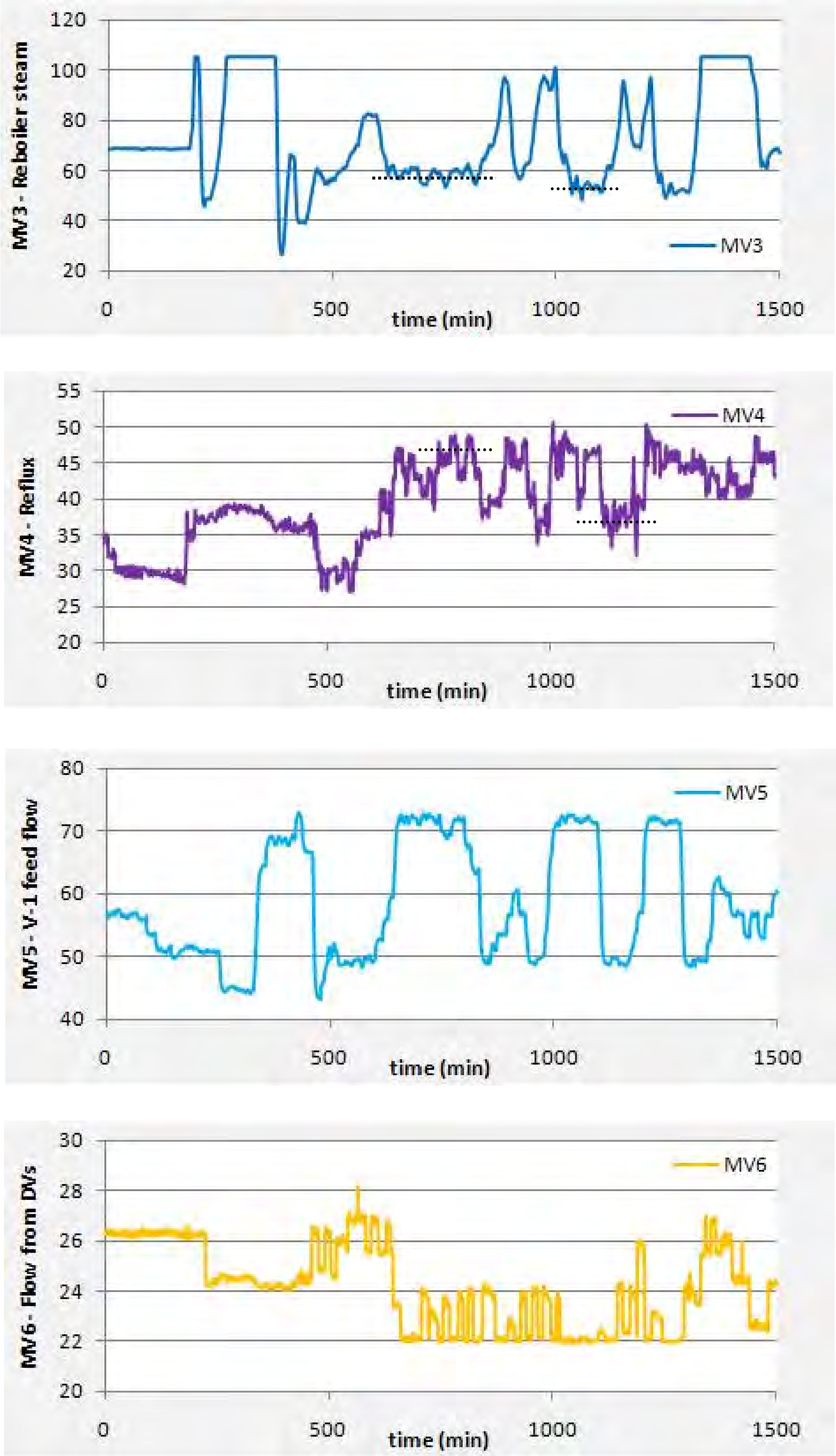


Figure 8.2: Manipulated Variables

Due to the nature of the operation of a MPC controller, these manipulated variables need to be tested in order to determine which (if any) inputs are correlated amongst each other. Correlation plots are provided below:

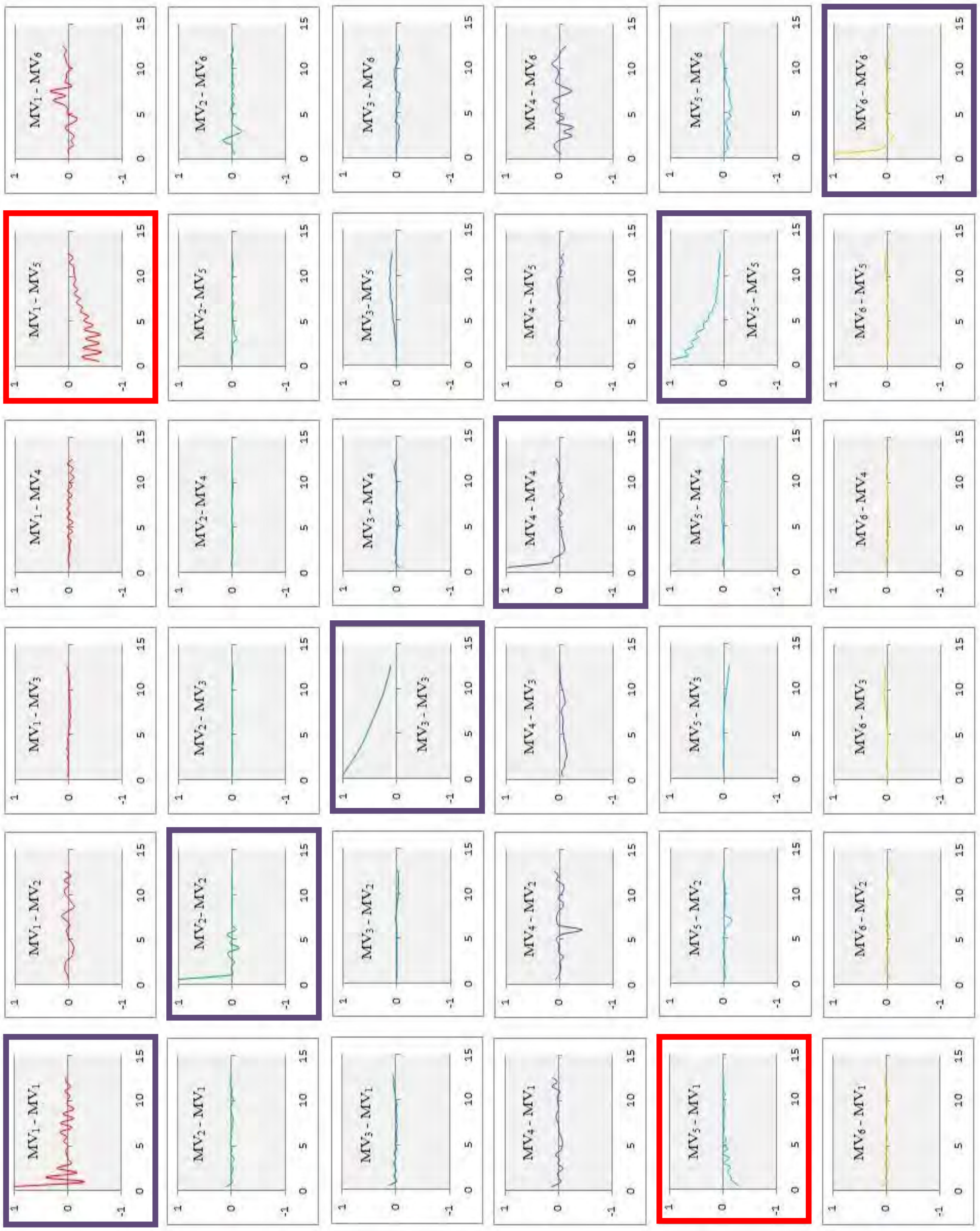
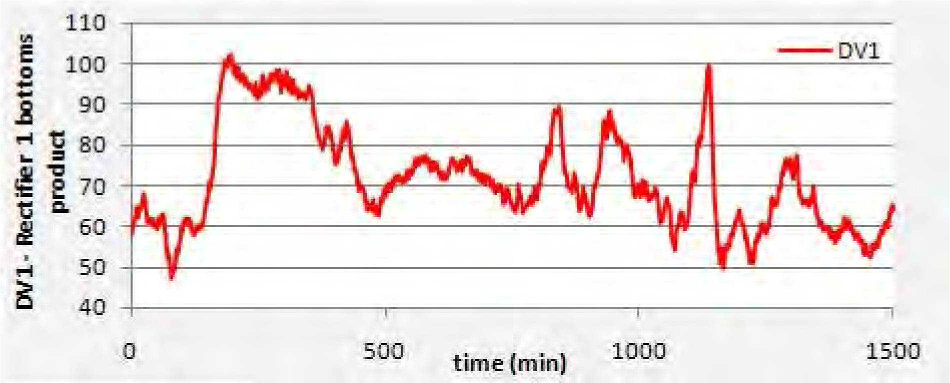


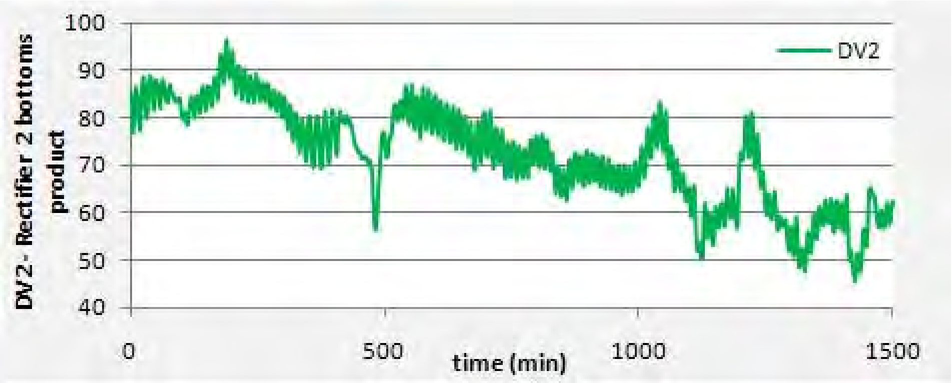
Figure 8.3: Correlation plots for all MVs

Each of the inputs is correlated with itself as expected. The only two inputs that are found to be correlated with each other are MV1 (V-2 column feed) and MV5 (V-1 column feed). These correlation plots are highlighted in red in the figure above. There is also an indication that the correlation between the two variables is negative implying that they exhibit an inverse relationship. This inverse relationship is justified by the following argument: when the flow to column V-2 gets too high (which is obtained from the reflux drum in column V-1), the feed to column V-1 is decreased in order to maintain control of the level in DM -1 (reflux drum in column V-1).

8.1.2. Measured Disturbance Variables

Controlled variables are influenced by a large number of disturbance variables present in the system. Although some of the disturbance variables are found in upstream processes, their influence on each CV is tantamount to the influence of the MVs. The variables are regarded as feed-forward variables, thus they will not exhibit any correlation amongst each other as they are uncontrolled variables. These DVs are displayed below:





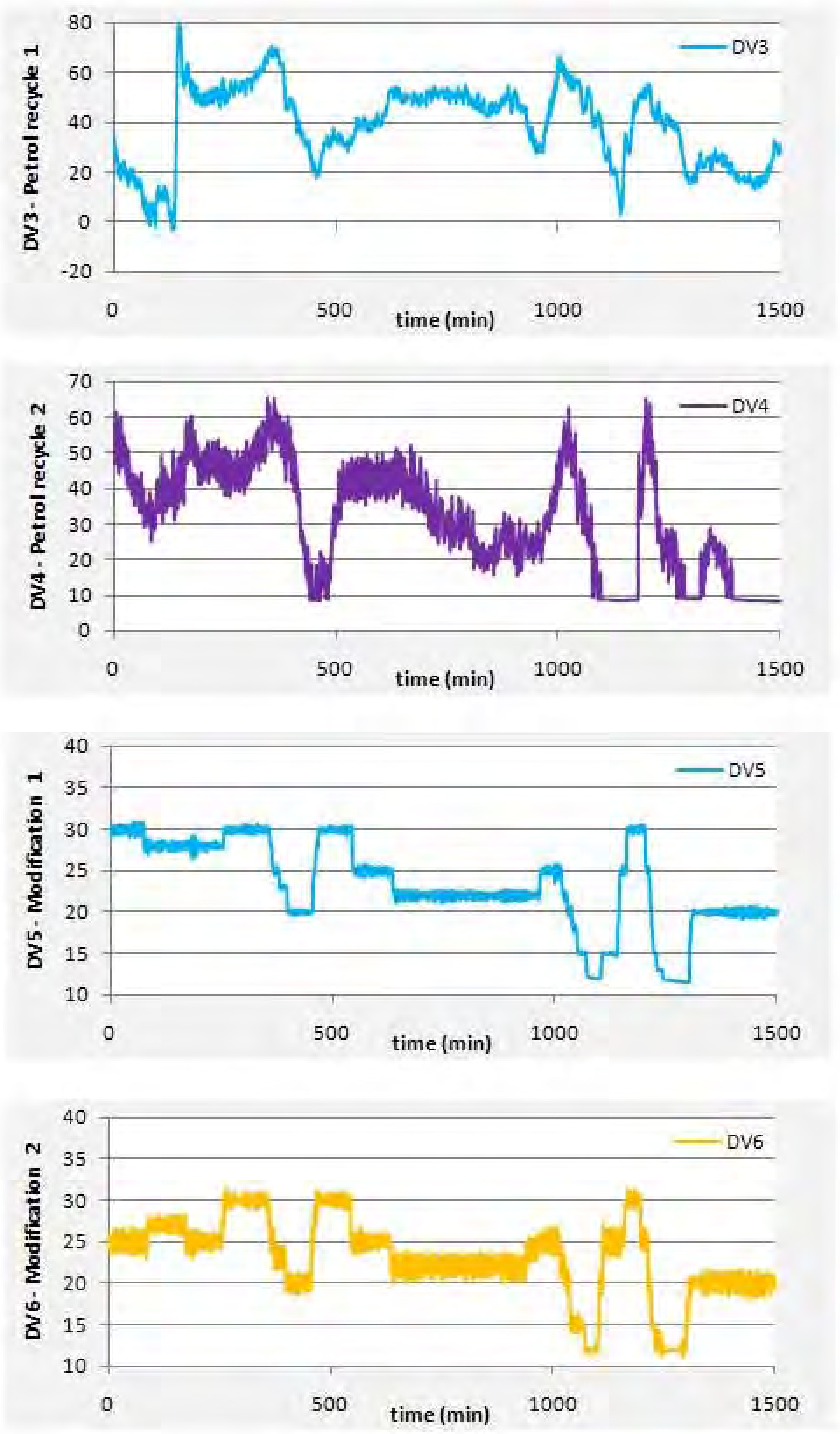
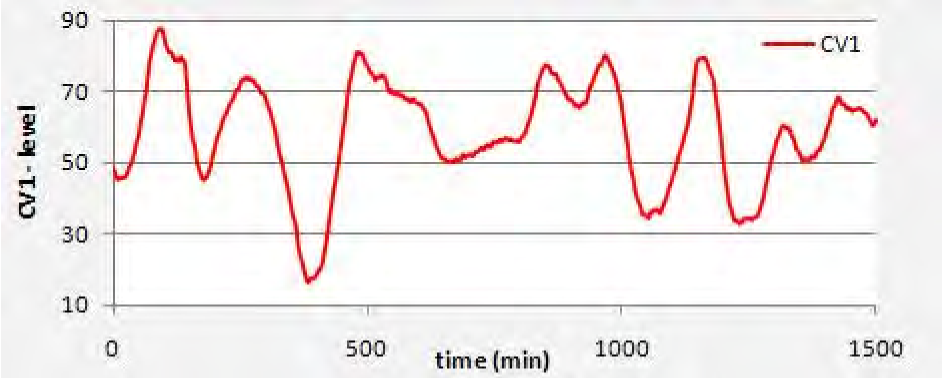
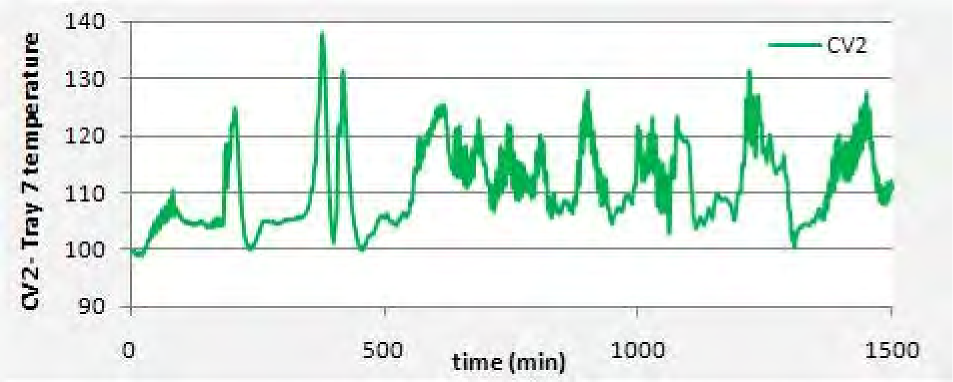


Figure 8.4: Disturbance Variables

8.1.3. Controlled Variables

This system has 6 CVs, each equally important in the operation of the petrol debutanizer. One of the primary objectives of the controller is to maintain the level in DM-1, which is CV1 (highlighted in blue in figure 8.1). Maintaining this level within its desired limits minimizes the amount of light carbon compounds being flared (MV2) as well as provides column V-1 with a suitable feed flow. CV2, which is the temperature of tray 7 in column V-2, is responsible for yielding the desired petrol/diesel ratio in the bottoms product. The desired separation in the column is formed as a result of the top column temperature (CV3) and the differential column pressure (CV4). CV5 represents the bottoms level in V-2. CV6 represents the petrol feed to the next phase of processing. These controlled variables are shown below:





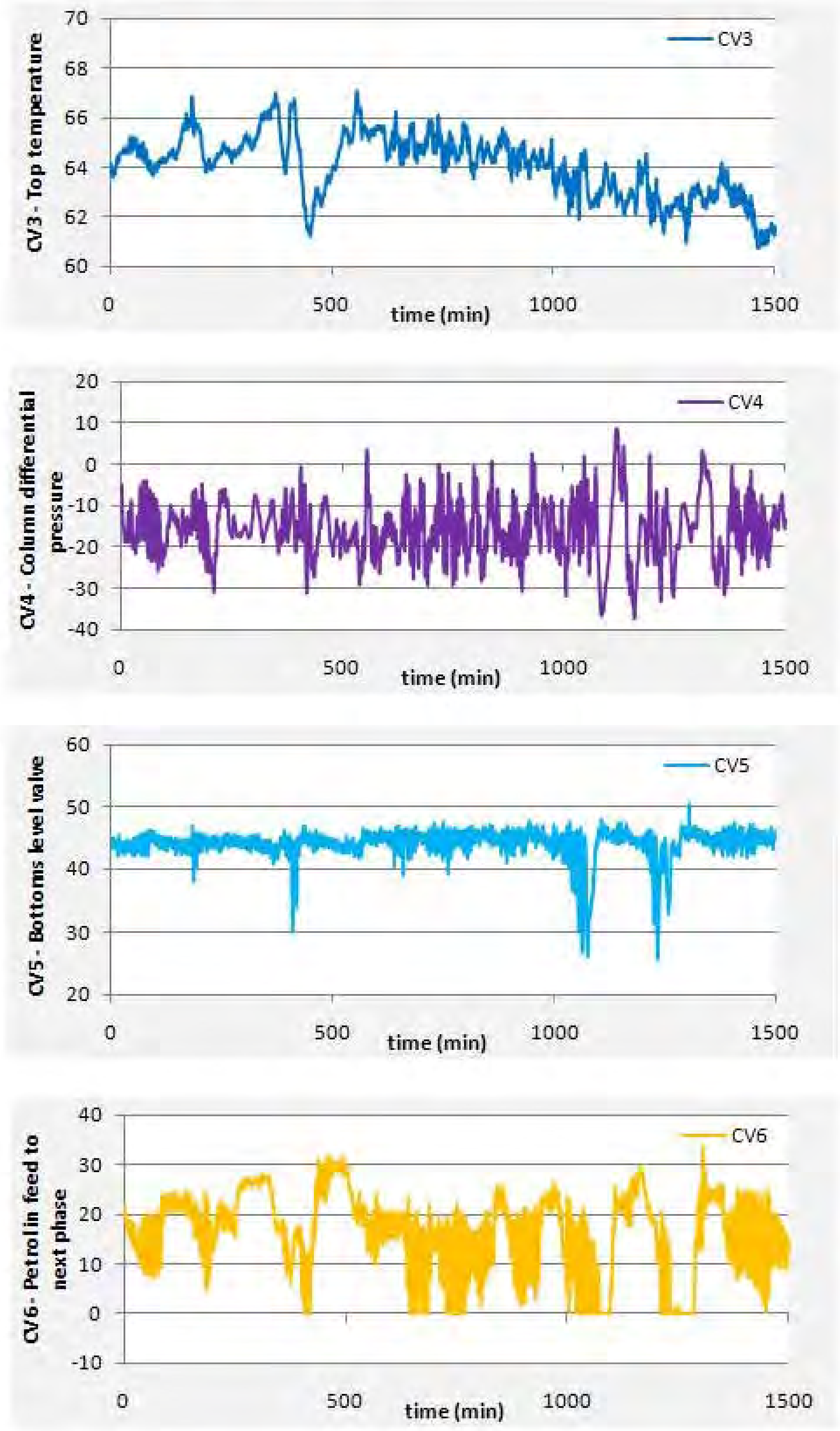


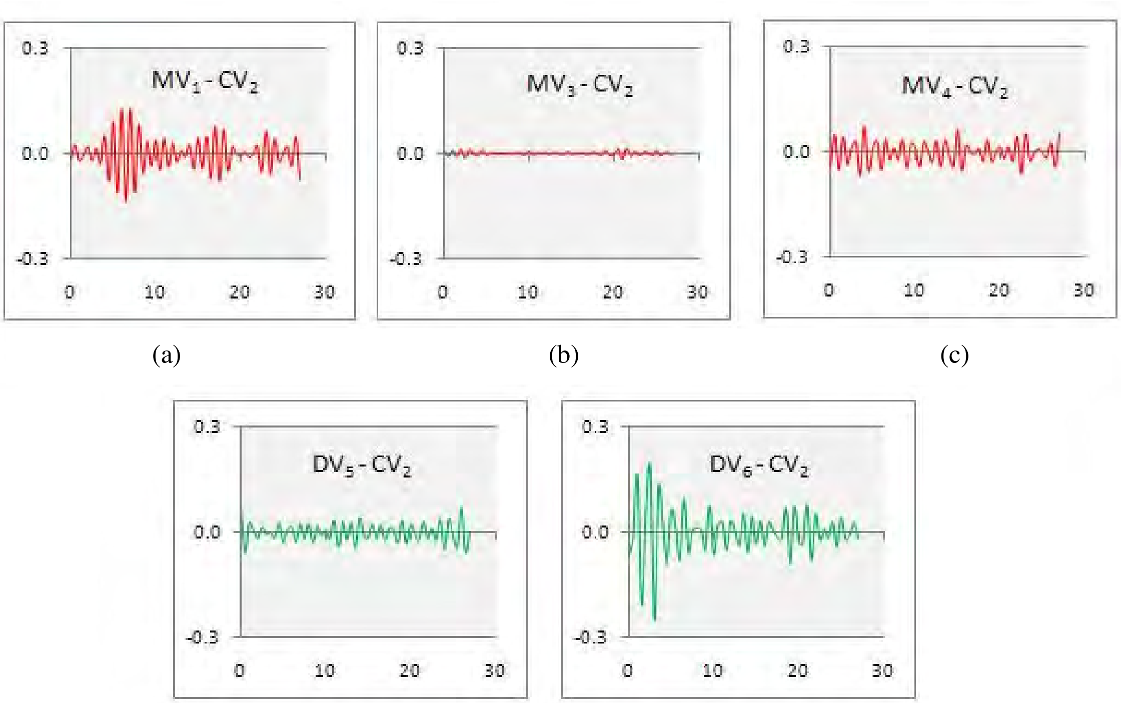
Figure 8.5: Controlled Variables

8.2. Industrial Results

This section presents and discusses the model validation results obtained for the models employed in the petrol debutanizer controller. Prior to the presentation of all the results, the models related to CV2 are first tested to demonstrate the concept of reducing the model set. Reducing the model set implies reducing the number of parameters that contribute to the error as well as reducing the impact of those signals that are influenced by the presence of unmeasured disturbances. The results for the remaining model sets are thereafter presented. The reader is referred to Appendix C for the full model set results.

8.2.1. Tray 7 Temperature in column V-2 – CV2

This temperature is influenced by MV1, MV2, MV4, DV5 and DV6. The models related to each of these pairings can be found in Appendix B, the second row of the model matrix. Equation (5.25) is used to define the error form. In the first instance, all models in CV2 are incorporated in the error formed and the results obtained are displayed below:



(a) (b) (c)

(d) (e)

Figure 8.6: Correlation plots for CV2

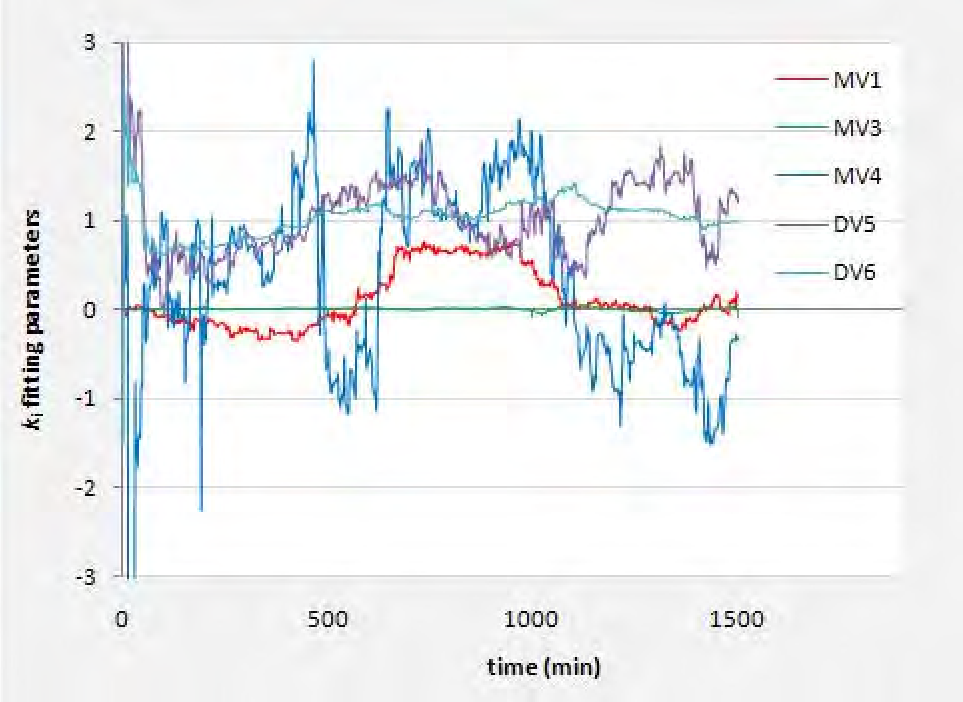


Figure 8.7: Moving Window regression plot for CV2

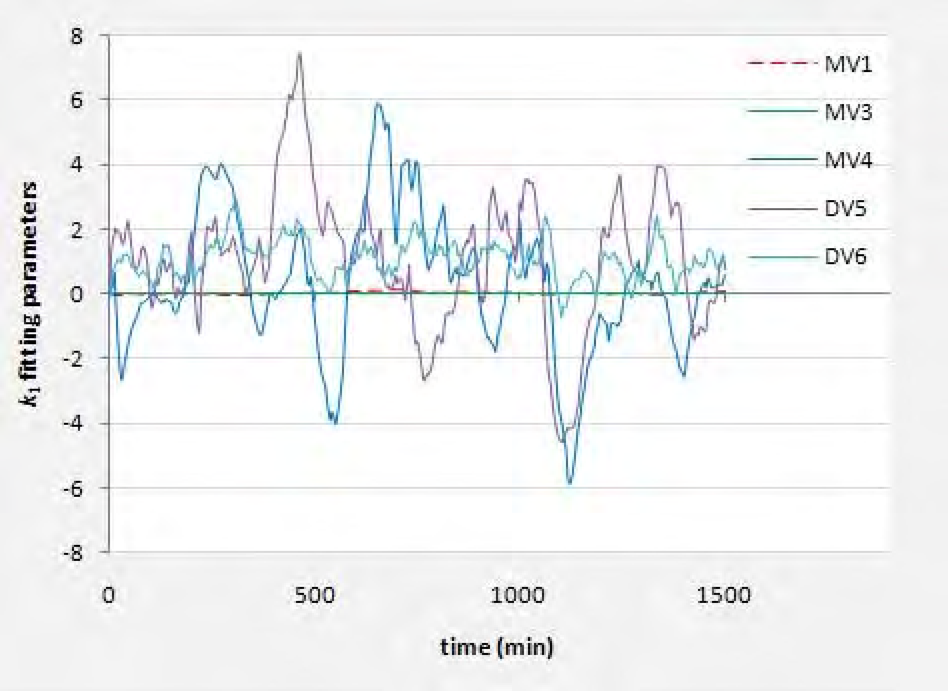


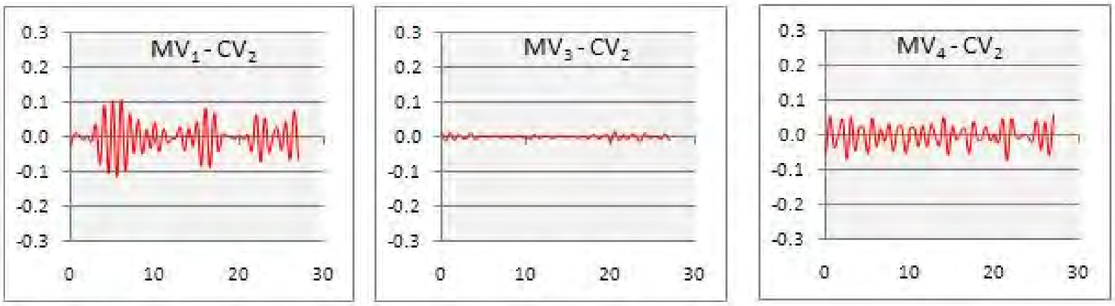
Figure 8.8: Kalman Filter plot for CV2

The correlation plots imply that a mismatch exists in channels relating CV2 to MV1, MV4, DV5 and DV6. These plots do not display strong correlations which could be a result of the influence of noise on the plant signals.

The magnitude of the k_i values obtained by the respective regressions in figures 8.5 and 8.6 are dissimilar due to the moving window applying equal weighting to each point within its window and thus allowing for a smoother response and the Kalman filter's reliance on more recent values. Models relating MV4-CV2, DV5-CV2 and DV6-CV2 exhibit large k_i values. From the model matrix presented by SASOL, the models in DV5-CV2 and DV6-CV2 contain a delay term of zero. Pragmatically, a delay term of zero is not possible, as the output cannot occur at the same time as the input. In this case, the diagnosis would be a time delay mismatch in the aforementioned channels. However, in the case of a large model set, noise signals can compound the error due to the large number of model coefficients. This subsequently makes the fitting parameters larger than one anticipates. One may consider using smaller subsets of models in order to find significant modeling errors. Smaller subsets will reduce the influence of plant noise. Webber and Gupta (2008) employed this tactic by holding certain MVs constant with the intention of reducing the subset of models that contain a mismatch.

8.2.1.1. Effect of MVs on CV2

Consider the results obtained for a reduced model set containing the models relating to the MVs:



(a) (b) (c)

Figure 8.9: Correlation plots for CV2 relating to each MV

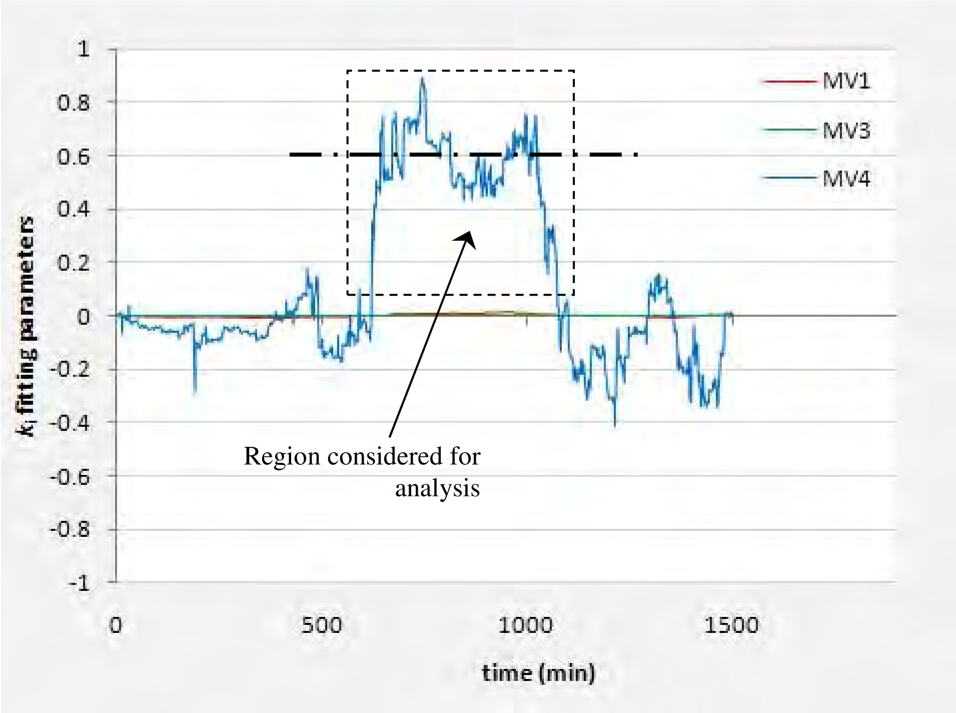


Figure 8.10: Moving Window regression plot for CV2 and each MV

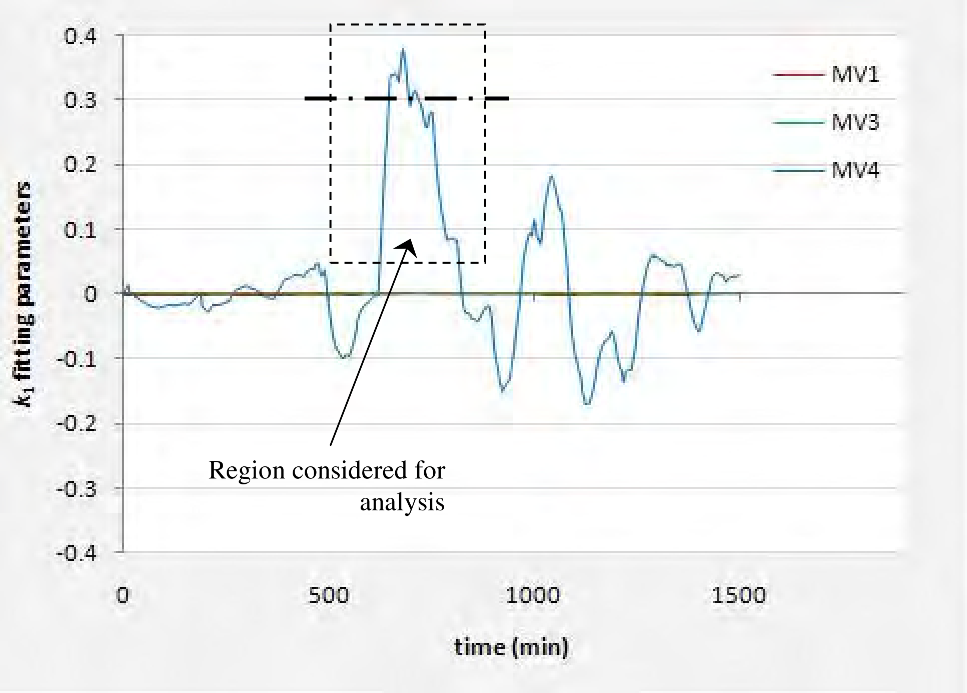


Figure 8.11: Kalman Filter plot for CV2 and each MV

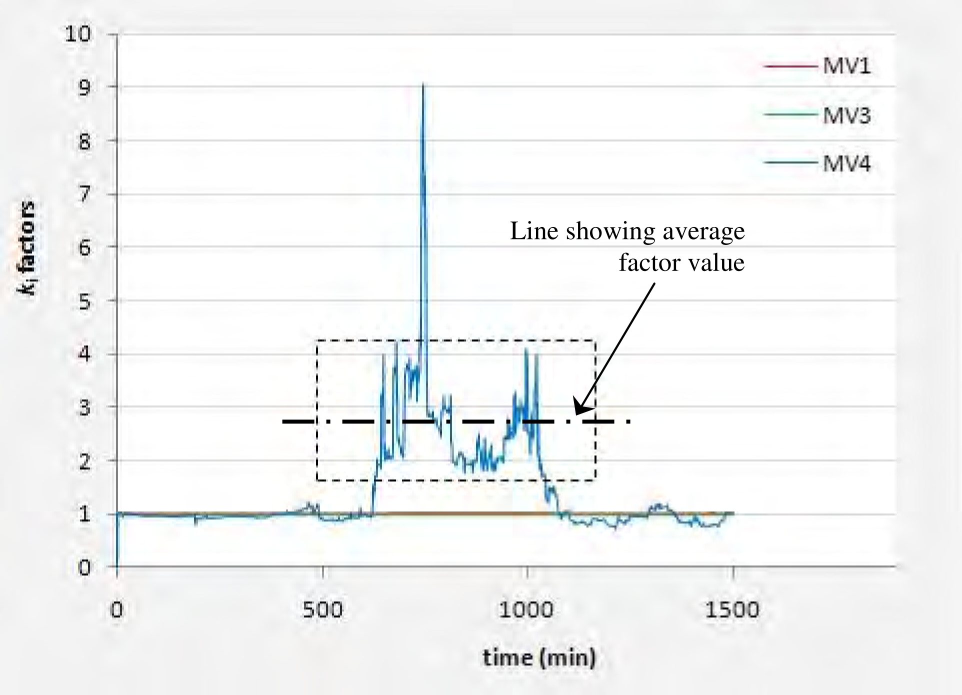


Figure 8.12: Moving Window regression factor plot for CV2 and each MV

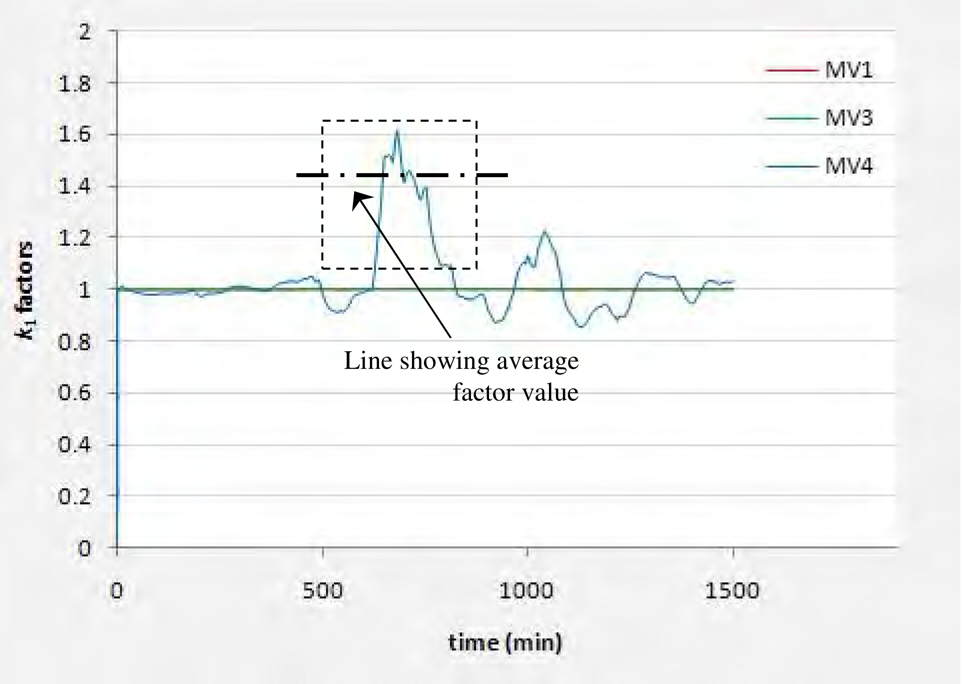


Figure 8.13: Kalman Filter factor plot for CV2 and each MV

It should be reiterated that the fitting parameters, k_i, in figures 8.10 to 8.13, represent the way the error relates to a particular u_i expressed as a fraction of the gain used in the model for that u_i term. Although the correlation plots display the same results obtained in the case where all variables were considered, the regression plots are now more distinct. It can also be seen that parameter in channel MV4-CV2 is strongly influenced by noise in figure 8.7. However, its variance is lowered as a result of reducing the model set. A diagnosis can now be made by considering the warning mechanisms developed in chapter 7.

The results for each warning mechanism are displayed below. This enables the maintenance engineer to locate the source of error.

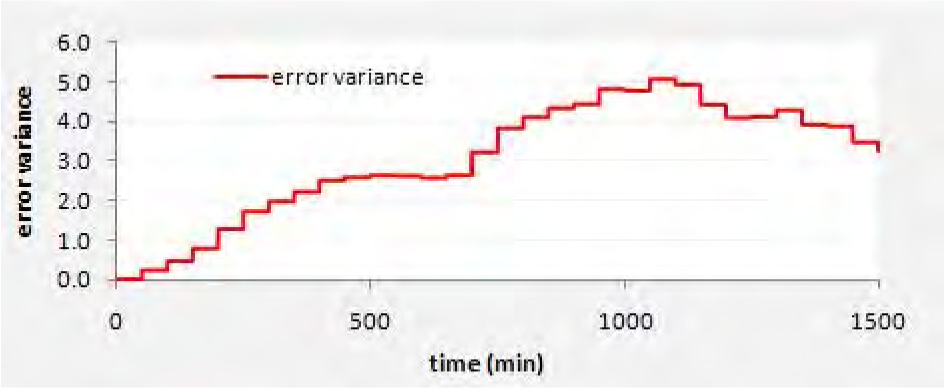


Figure 8.14: Error Variance for CV2 related to its MVs

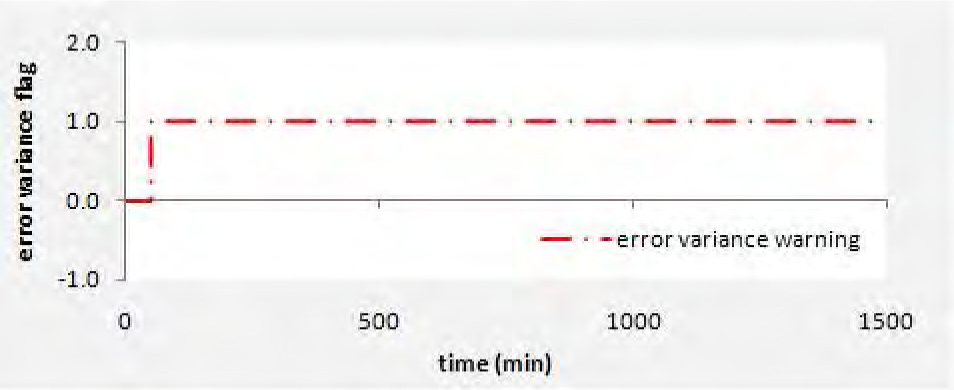


Figure 8.15: Warning Flag for high error variance

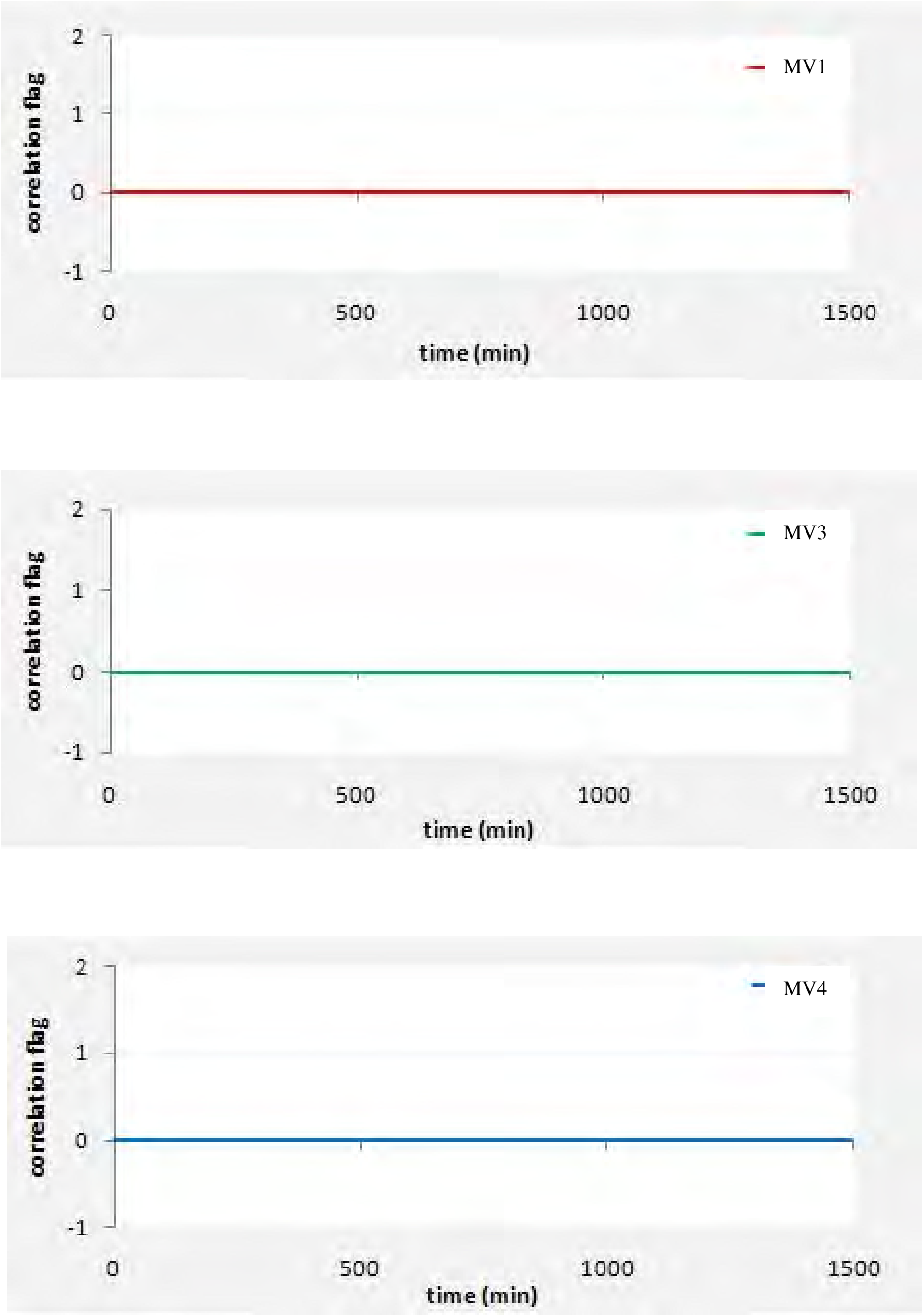


Figure 8.16: Warning Flag for correlation bounds violation

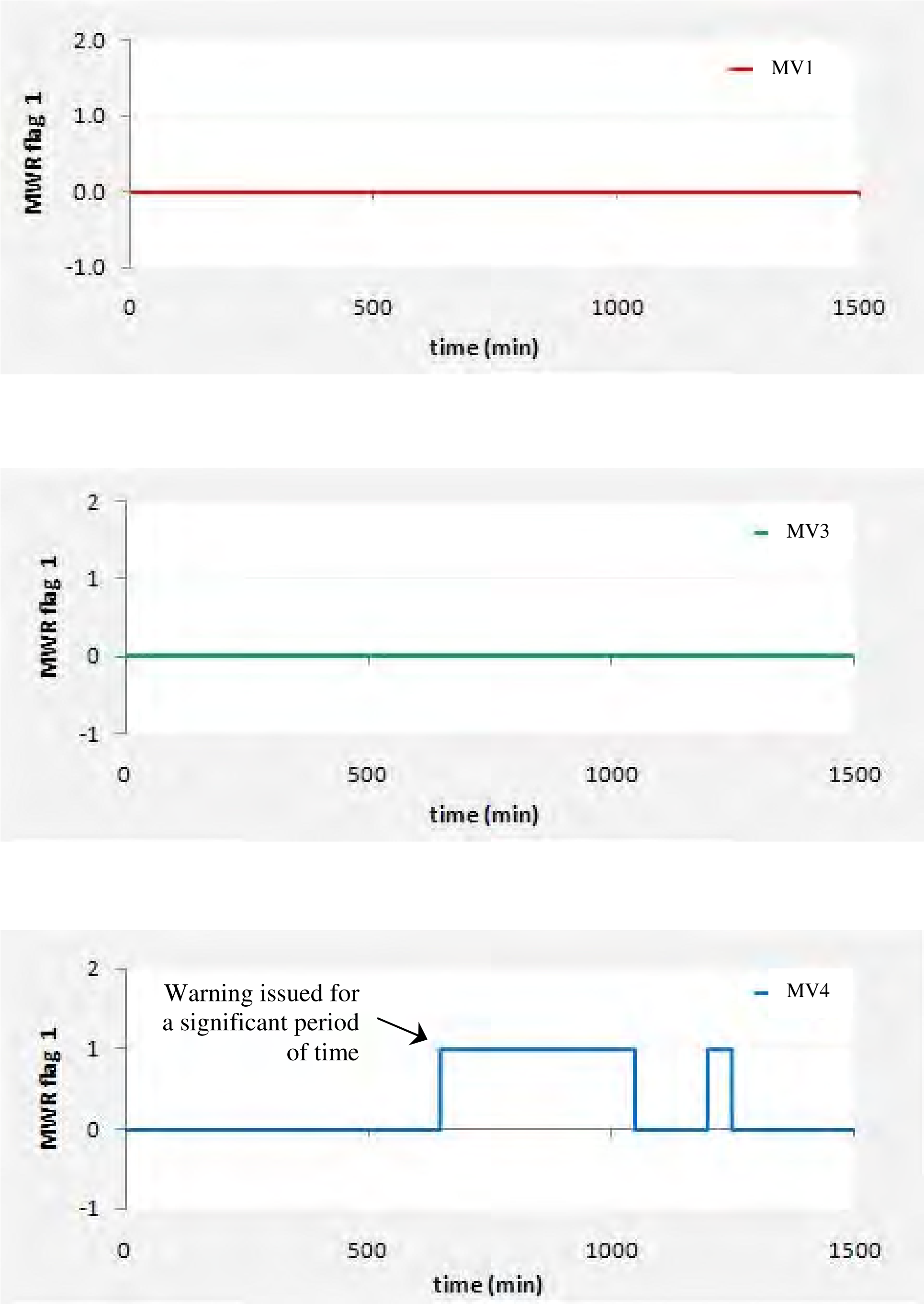


Figure 8.17: MWR Warning Flag for gain or time delay mismatch

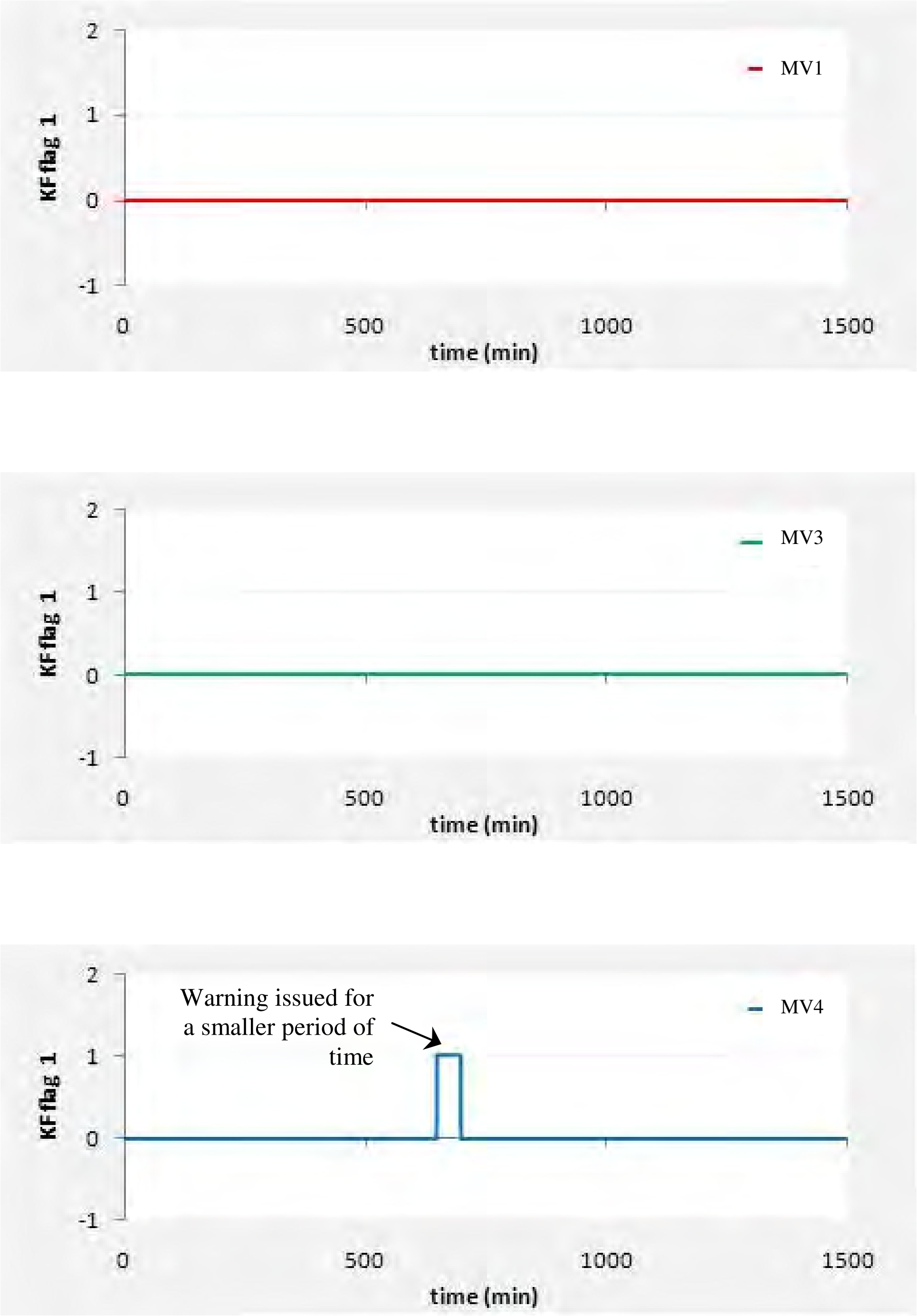
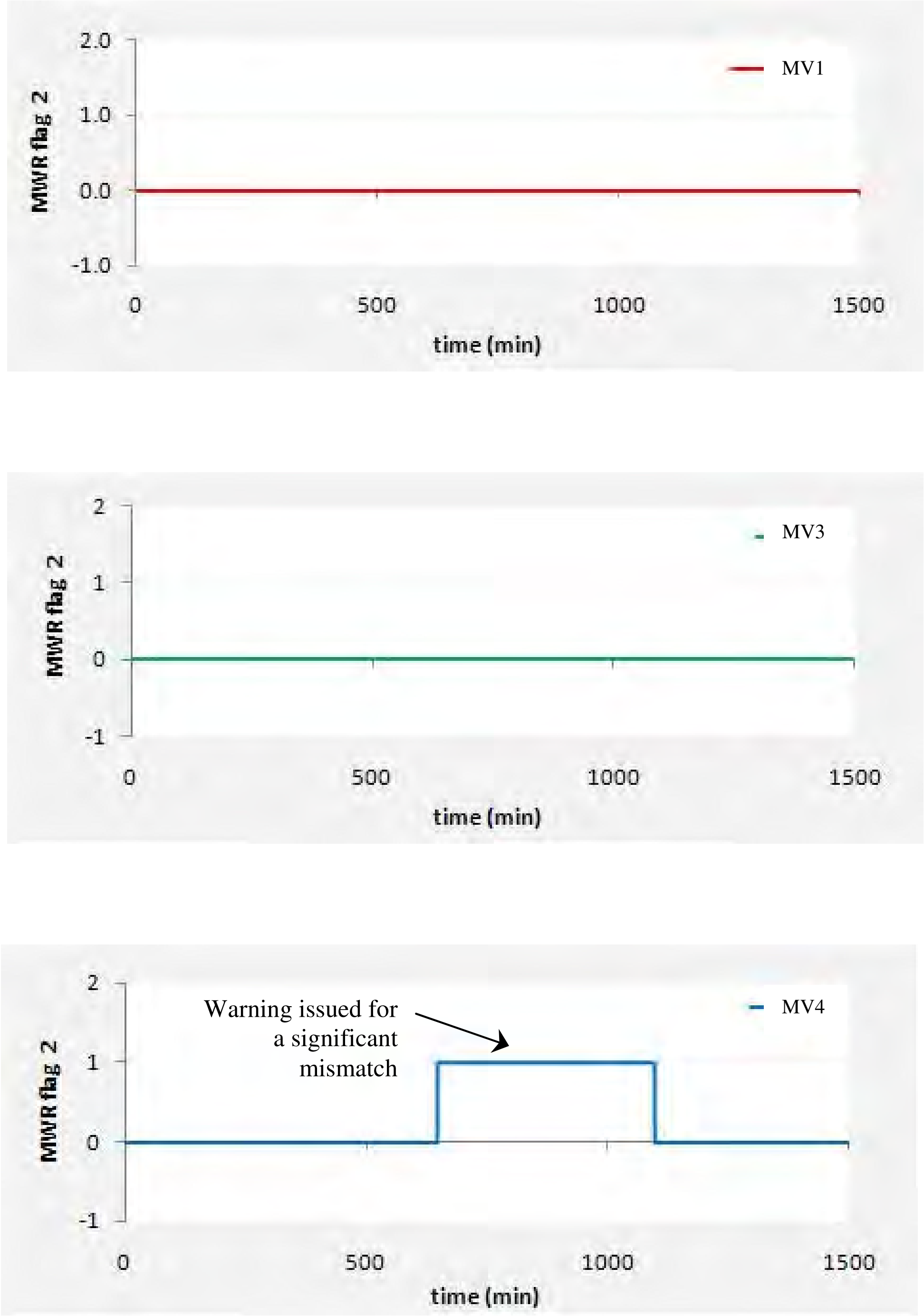
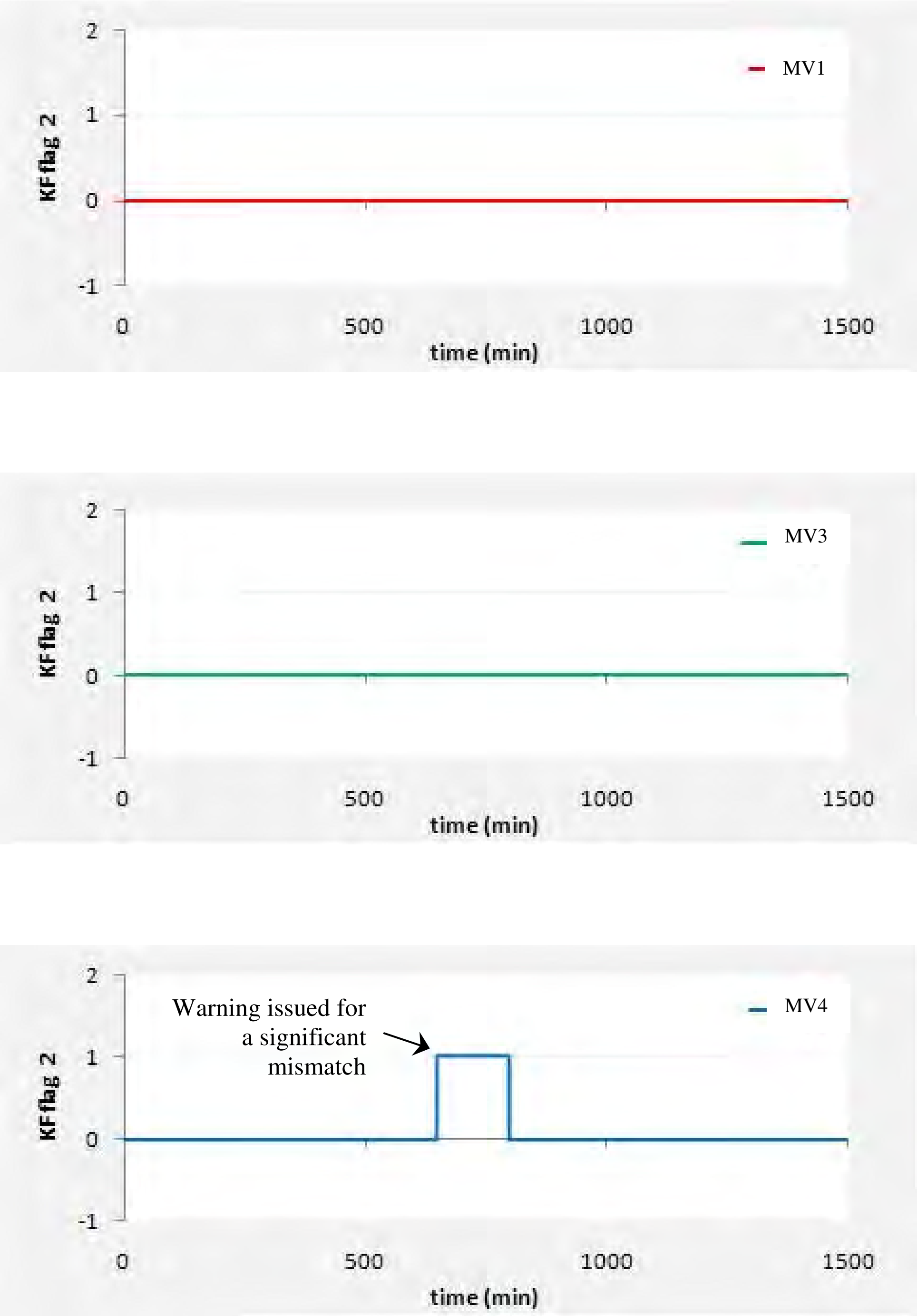


Figure 8.18: KF Warning Flag gain or time delay mismatch for CV2 and its MVs



Warning issued for a significant mismatch

Figure 8.19: Significance of model error: MWR



Warning issued for a significant mismatch

Figure 8.20: Significance of model error: KF

Figure 8.14 displays the increase of the error variance over the sampled data time period. This variance is shown to be significantly higher than the benchmark established in chapter 7. A warning is therefore issued (figure 8.15). The correlation flags issued in figure 8.16 are all zero over the time frame indicating that there is no MPM or that there is an influence of noise in the system, which reduces the ability of the correlation analyses to detect MPM. The next set of flags is used to determine the form of parameter mismatch present (if it exists). Figure 8.17 (MWR flag) shows that for MV4, a flag warning of 1 is issued for a long period of time. This implies that the parameter obtained is in the range -1 to 1, ruling out a mismatch possibility due to a time delay. In figure 8.18 (Kalman Filter flag), a flag of 1 is issued for MV4-CV2 but for a shorter period of time. This is due to the fact that the input data variability is not sufficiently high. The next warning mechanism determines how significant the model mismatch is by considering it as a model gain change (or a higher order polynomial coefficient). Figures 8.19 and 8.20 reveal a flag warning of 1 for the channel related to MV4. This is to due to a significant gain mismatch in this channel. Hence the diagnosis for these MVs is as follows: there exists no MPM in channels relating CV2 to MV1 and MV3. A significant gain mismatch is found for MV4-CV4. The actual regression fitting parameters and factors are displayed in the tables below for the period of time in which the flag warnings were issued:

Table 8.2:
Overall Regression parameters for CV2 in relation to MVs

	k_1	k_3	k_4
MOVING WINDOW REGRESSION	$1.511\text{x}10^{-3}$	$1.531\text{x}10^{-4}$	0.5921
KALMAN FILTER	$2.214\text{x}10^{-4}$	$3.450\text{x}10^{-6}$	0.2988

Table 8.3:
Overall Regression multiplicative factors for CV2 in relation to MVs

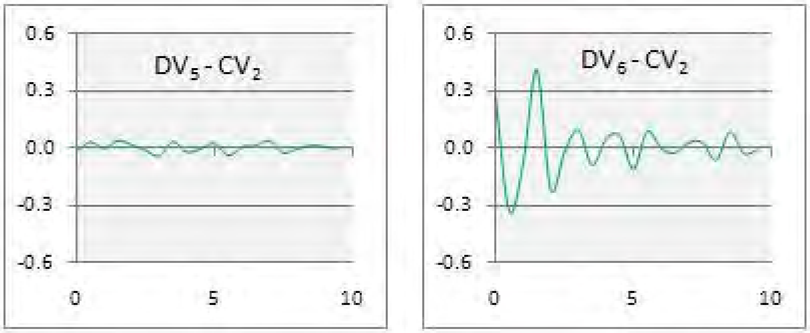
	η_1	η_3	η_4
MOVING WINDOW REGRESSION	1.0016	1.0001	2.4517
KALMAN FILTER	1.0002	1.0000	1.4262

The results displayed in the tables above were obtained by averaging the data in the periods where the flags were issued. The difference in the values obtained for the moving window regression (a model gain being ~2.45 times higher than the plant gain) and the Kalman Filter (a model gain being ~1.42 times higher than the plant gain) is due to the properties of each method and its ability to deal with low variance data and the influence of noise. The reader is to refer to figure 8.2 which illustrate periods of low variance input data.

It should also be noted that the mismatch displayed for the model MV4-CV2 could possibly be as a result of a mismatch in the higher order polynomial coefficient in the denominator as it is shown to be significantly larger than the gain itself (refer to appendix B, figure B.1).

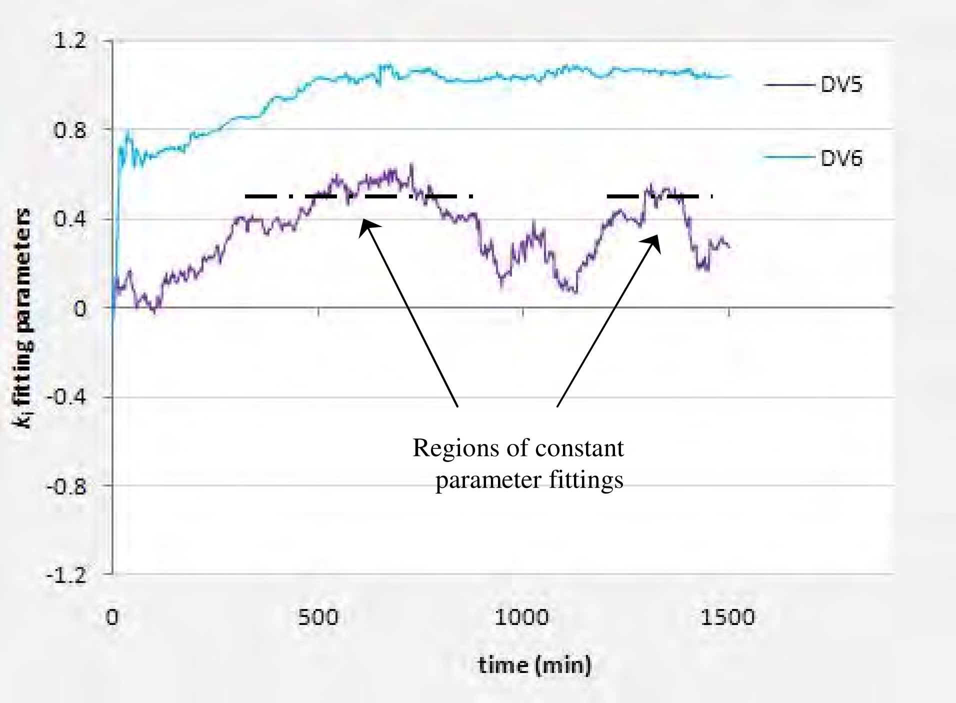
8.2.1.2. Effect of DVs on CV2

In this instance, the respective DVs were considered separately from the MVs. The results are illustrated below:



(a) (b)

Figure 8.21: Correlation plots for CV2 related to each DV



Regions of constant parameter fittings

Figure 8.22: Moving Window regression plot for CV2 and its DVs

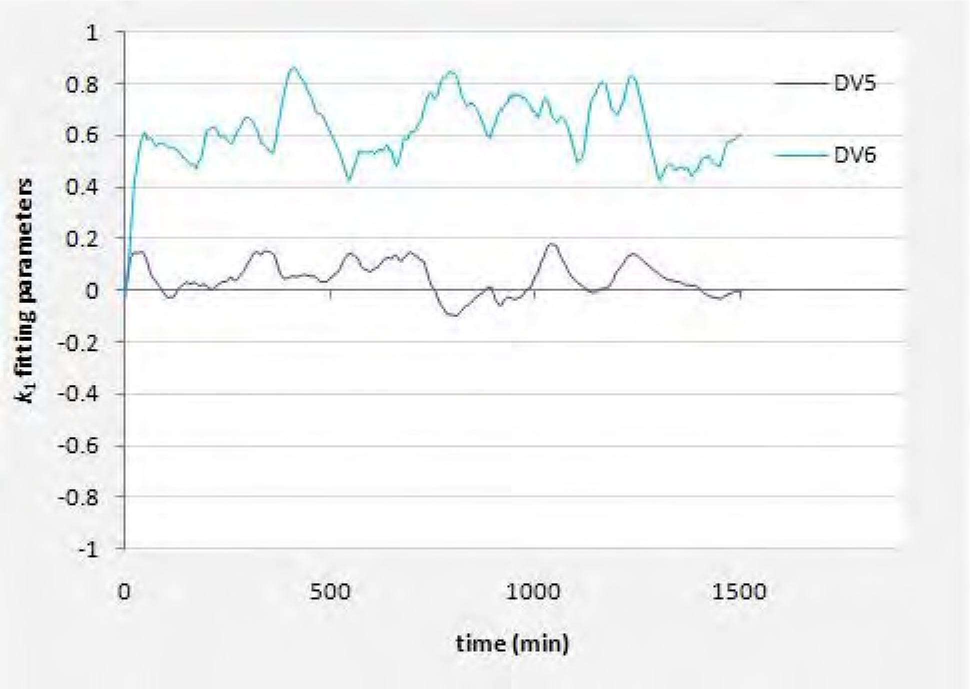


Figure 8.23: Kalman Filter plot for CV2 and its DVs

The magnitude of the coefficients in the correlation plot in figure 8.21(b) is shown to range between -0.3 to +0.3 in comparison to the plot displayed in figure 8.6(e) which has a range of -0.2 to +0.2.

The regression plot obtained in figure 8.22 is in agreement with the results obtained in figure 8.7 as the fitting parameter related to DV6-CV2 is shown to have a steady reading greater than 1 for a extended period of time. In the case of the parameter obtained for DV5-CV2, figure 8.7 illustrates that it exceeds a value of 1 over a period of time. However, due to the fact that the model set was reduced to handle MVs and DVs separately, the magnitude of this parameter is shown to fall into the region of -1 to +1. The Kalman filter also shows a drop in the magnitude of the parameters related to each DV when figure 8.23 is compared to figure 8.8.

These results together with the warning mechanisms below were used to diagnose the modeling error:

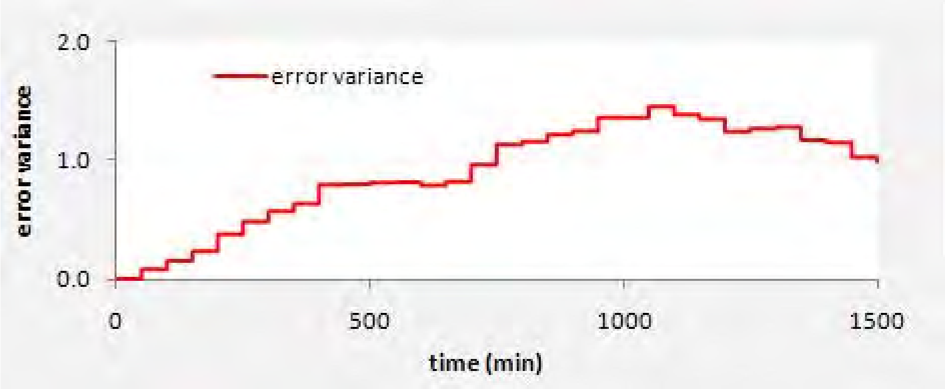


Figure 8.24: Error variance due to the influence of the DVs

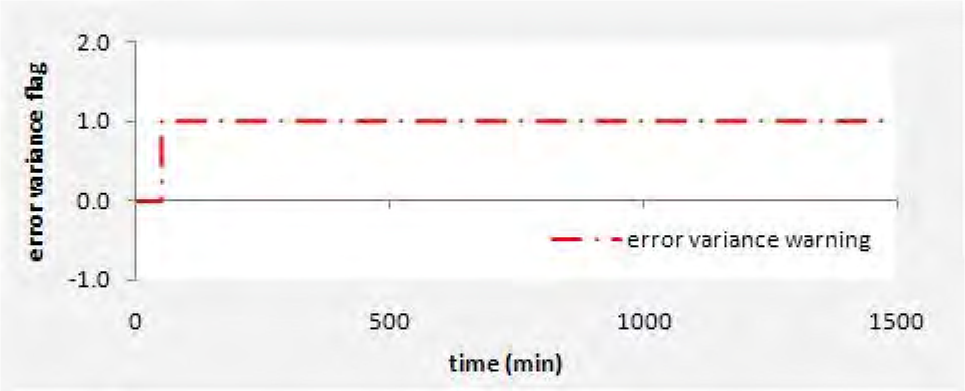


Figure 8.25: Error variance warning flag due to the influence of the DVs

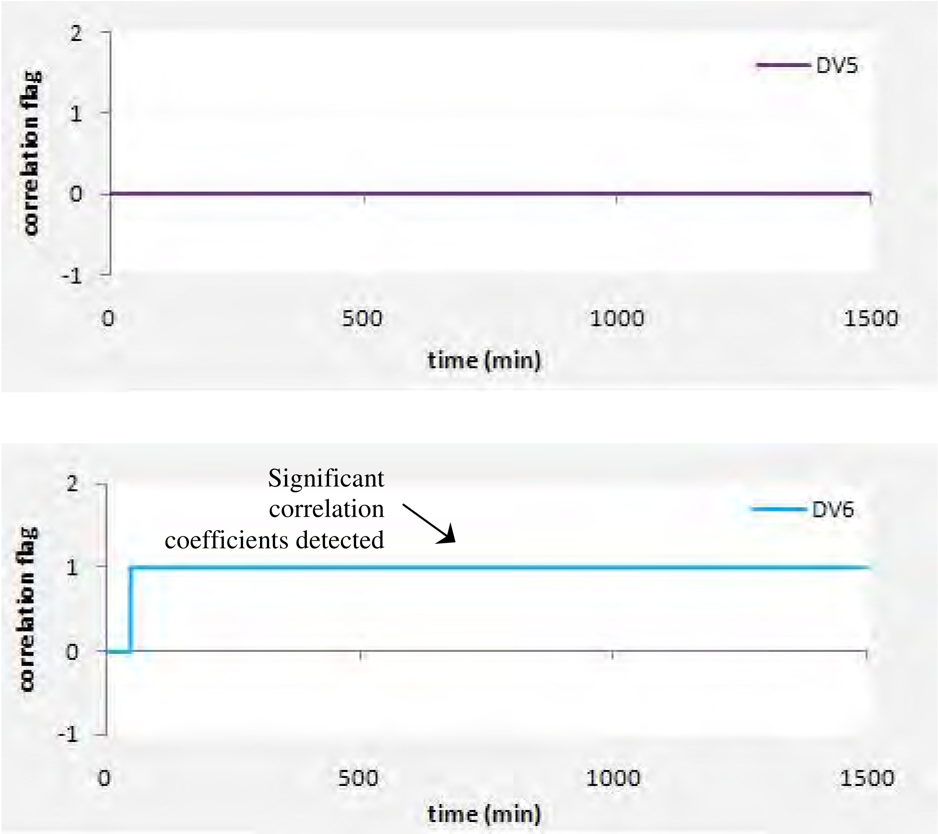
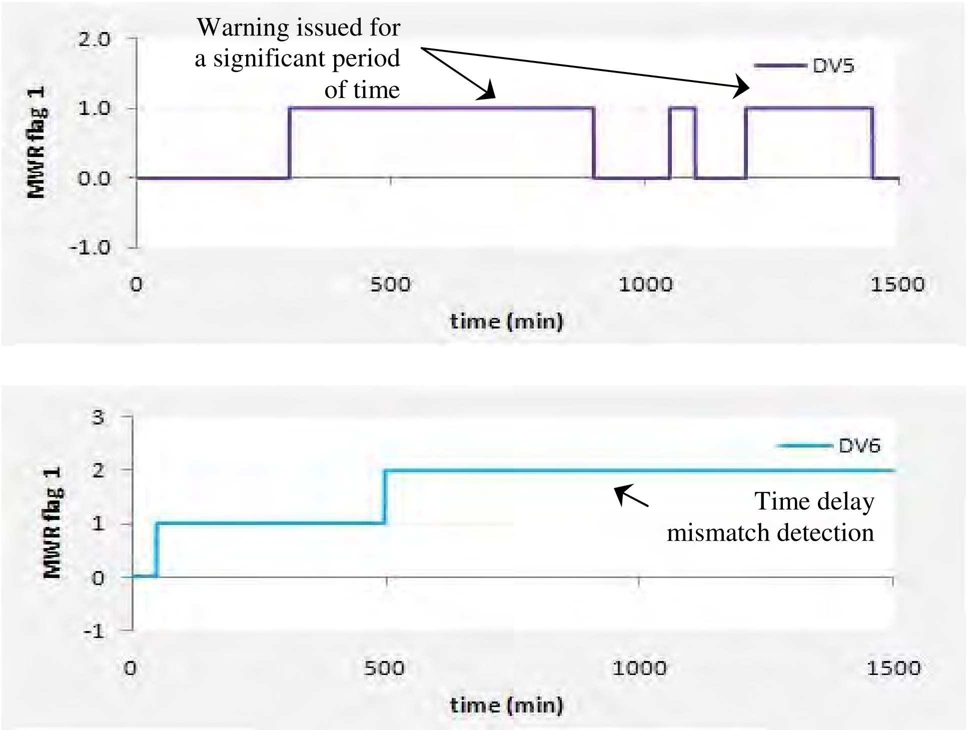


Figure 8.26: Correlation confidence bounds violation for DVs



Warning issued for a significant period of time

Time delay mismatch detection

Figure 8.27: MWR Warning Flag gain or time delay mismatch

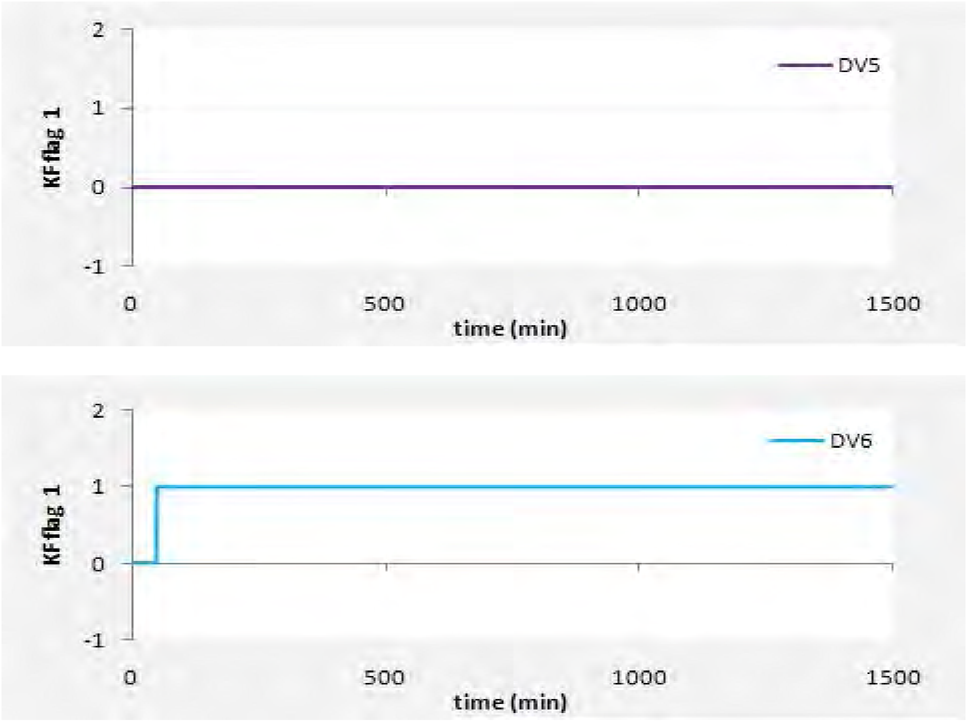


Figure 8.28: KF Warning Flag gain or time delay mismatch

The variance of the error formed due to the disturbance variables in figure 8.24 is much lower than the variance obtained in comparison to figure 8.15. Thus is can be stated that the bulk of the error formed when all the variables were considered can be owed to the models linked to the respective manipulated variables. Nonetheless, the variance obtained in figure 8.24 is still significant (see figure 8.25). The correlation warning flags show that DV6-CV2 exceeds the bounds set for significant mismatch. The subsequent warning mechanism (employed for the MWR fitting parameters) show that the fitting parameter for DV5-CV2 lies in the range of -1 to +1 implying the presence of a gain error, but the parameter for DV6-CV2 lies outside these bounds illustrating that a time delay mismatch is present in this model. The very same means is applied to the Kalman Filter parameters and yields conflicting results reiterating the fact that the Kalman Filter is most suitable when the data is highly informative. Channel DV5-CV2 was tested for the significance of the mismatch present with the factor plot displayed below, followed by the next warning mechanism and summary of the overall parameter results:

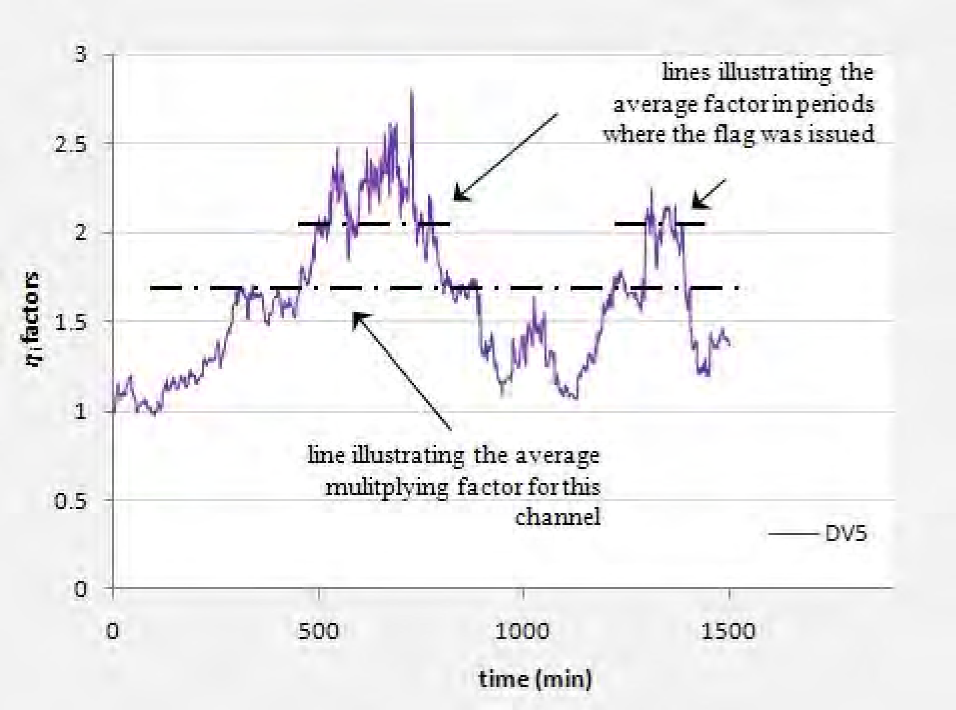


Figure 8.29: Moving Window regression factor plot for CV2 and DV5

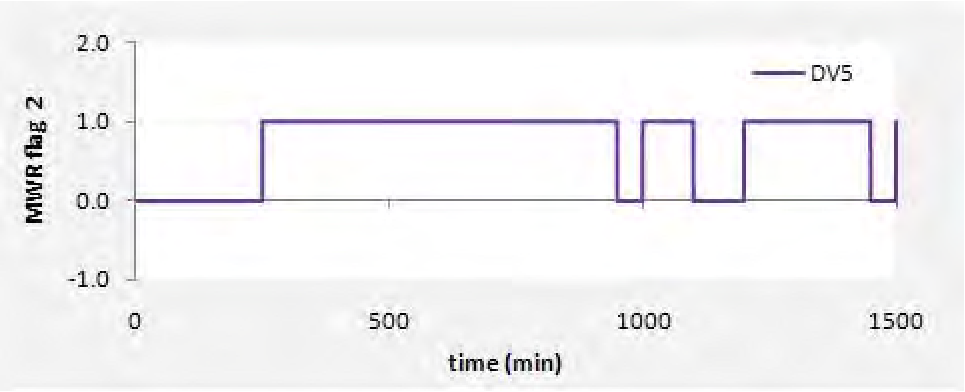


Figure 8.30: Significance of model error: MWR

Table 8.4:
Overall Regression parameter and multiplicative factor for DV5-CV2

	k_1
MWR fitting parameter	0.5305
	η_1
MWR multiplying factor	2.1299

The gain error is significant in DV5-CV2 and the model gain is found to be ~2 times higher than that of the plant. Careful consideration of the parameters present in the model in channel DV5-CV2 shows that the coefficients in the numerator and denominator are significantly large in magnitude. This suggests that this mismatch could be due to a mismatch of ~2 times the higher order coefficient in the numerator or ~0.5 times the higher order coefficient in the denominator. A mismatch in the numerator directly implies that a mismatch exists in the gain due to the fact that the gain is extracted as a common factor from the coefficients in the numerator.

The overall diagnosis for the disturbance variables is as follows: the warning mechanisms together with the regression results reveal that channel DV6-CV2 contains a time delay mismatch and channel DV5-CV2 shows the possibility of a gain mismatch or the mismatches mentioned above.

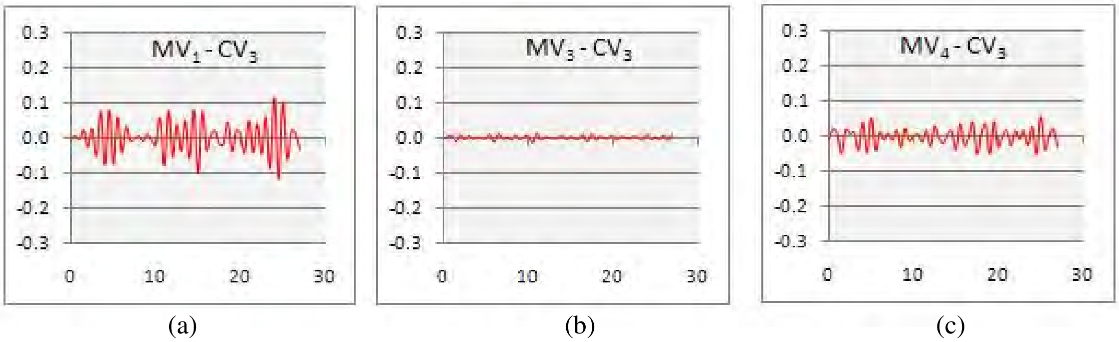
Mismatch in the models shown for both those related to the MVs and DVs could result in improved controller performance (depending on the controller tuning) or cause the temperature in tray 7 (CV2) in column V-2 to shift away from its desired range. The latter point could be detrimental when the column is operated in diesel mode. If this temperature is found to be out of its desired range, then the possibility of light components entering the bottoms product increases. The phase to which this bottoms product is sent next does not have a facility to process these LPG compounds and they are thus flared. This is a regarded as a loss as the LPG compounds are normally removed in the distillate and sent for further processing into marketable products.

8.2.2. Top Column Temperature in column V-2 – CV3

The top column temperature is controlled by MV1, MV3, and MV4 and affected by DV5 and DV6. The results displayed below involve reducing the models related to CV3 into two sets as shown for CV2. The reader is referred to Appendix C for the results of the full model set. It should be noted henceforth that warning mechanisms will be shown only in the instance where a significant mismatch is detected.

8.2.2.1. Effect of MVs on CV3

The results relating the models of CV3 to MV1, MV3 and MV4 are shown below. These models are responsible for maintaining the top column temperature within its desired limits in order to sustain the purity of the top product.



(a) (b) (c)

Figure 8.31: Correlation Plots for CV3 related to MVs

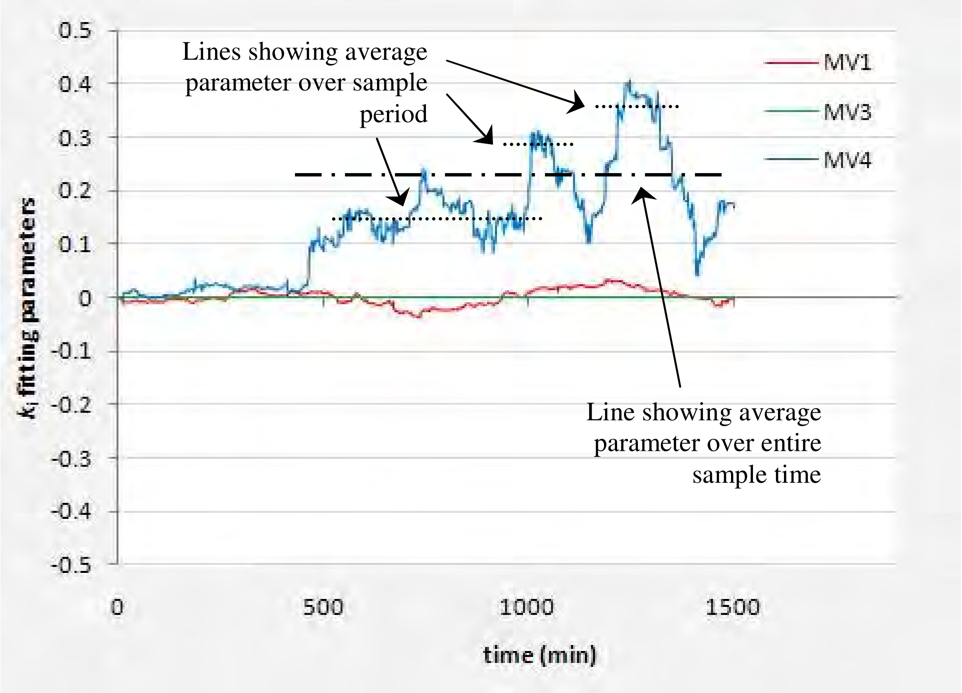


Figure 8.32: Moving Window regression plot for CV3 and MVs

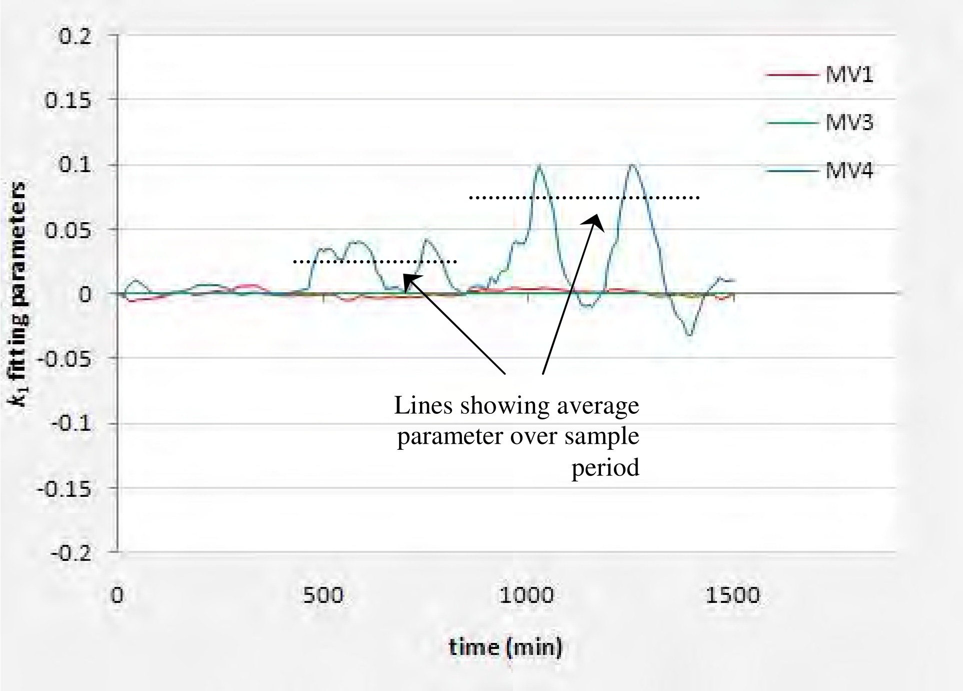


Figure 8.33: Kalman Filter plot for CV3 and MVs

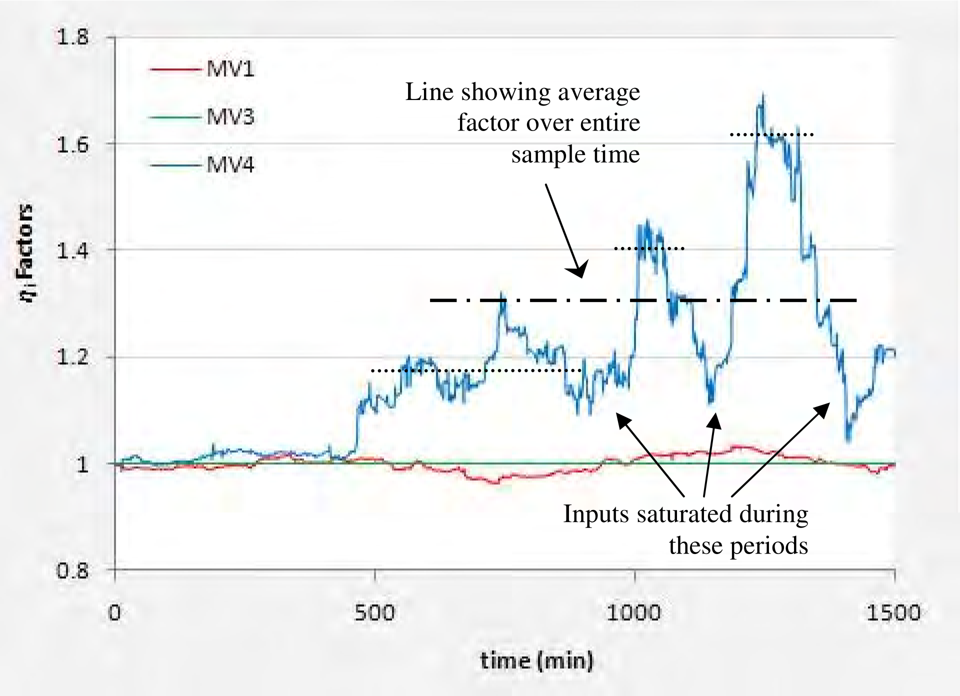


Figure 8.34: MWR factor plot for CV3 and MVs

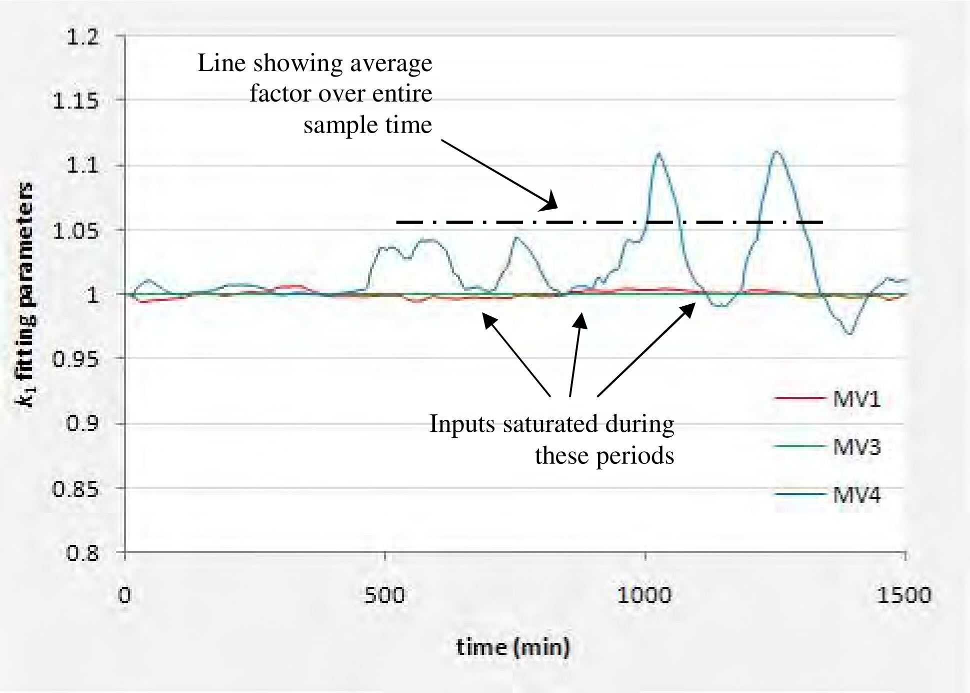


Figure 8.35: KF factor plot for CV3 and MVs

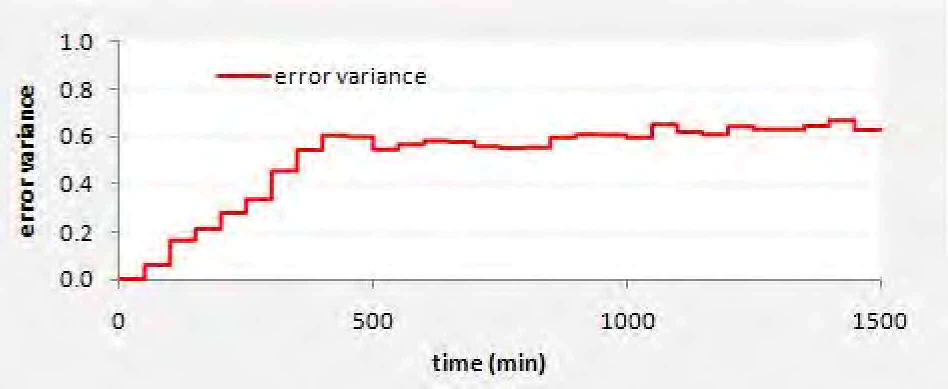


Figure 8.36: Error variance for CV3 due to the influence of the MVs

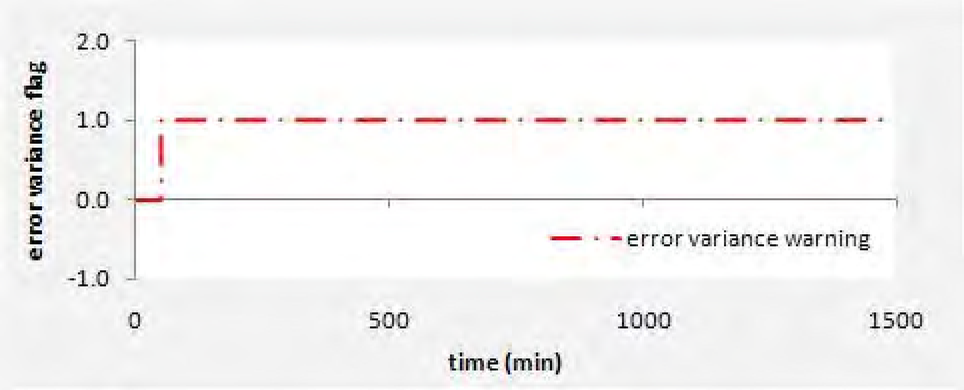
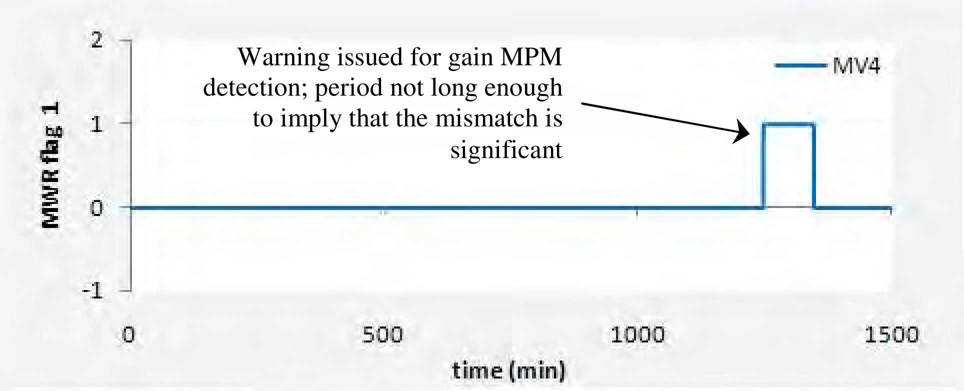


Figure 8.37: Error variance flag for CV3 due to the influence of the MVs



Warning issued for gain MPM detection; period not long enough to imply that the mismatch is significant

Figure 8.38: Warning flag for significant gain or time delay mismatch detection

The correlation plots reveal that possible mismatches were found in channels MV1-CV3 and MV4-CV3. The range of the correlation coefficients lie within the confidence bounds set and no warning flags were issued accordingly. The error variance is shown in figure 8.36 to be greater than the benchmark. The extent of the mismatches is illustrated in figures 8.32 and 8.33. The unsteady response of these variables is owed to the fact that during certain periods of time the inputs involved were saturated at their limits. The difference in the range of the parameters obtained for the Moving window regression and the Kalman Filter is again owed to the variance of the input data. These plots suggest that channel MV4-CV3 contains a gain mismatch. Careful analysis of figure 8.32 suggests that this parameter is not in violation of the confidence bounds set i.e. within -0.3 to +0.3. As a result, no warning was issued for a significant gain mismatch apart from a warning issued for a short period of time in figure 8.38. This time period is regarded as too short to provide evidence of a significant gain mismatch. The overall fitting parameters and multiplicative factors are displayed in the table below:

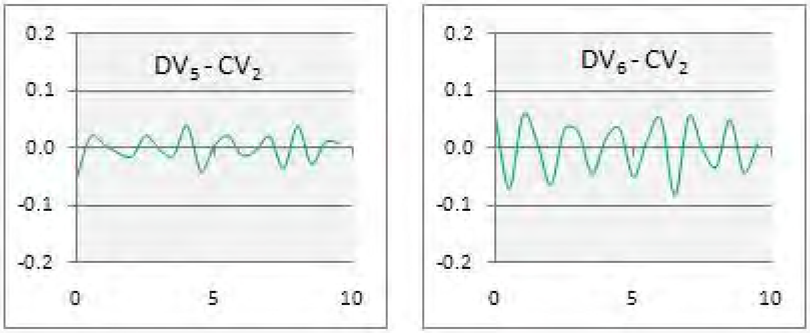
Table 8.5:
Overall Regression parameters and multiplicative factors for MV_i-CV3

	k_1	k_3	k_4
MWR fitting parameters	1.58×10^{-3}	-8.27×10^{-6}	0.2301
Kalman Filter parameters	-1.44×10^{-4}	-1.03×10^{-6}	0.0623
	η_1	η_3	η_4
MWR multiplying factors	1.00	1.00	1.2989
Kalman Filter multiplying factors	1.00	1.00	1.0664

The overall diagnosis for this model set is as follows: although the gain in channel MV4-CV3 is shown to be ~1.3 times higher than the plant gain, all the models in this model set (MV1-CV3, MV3-CV3, MV4-CV3) are shown to represent the plant fairly well.

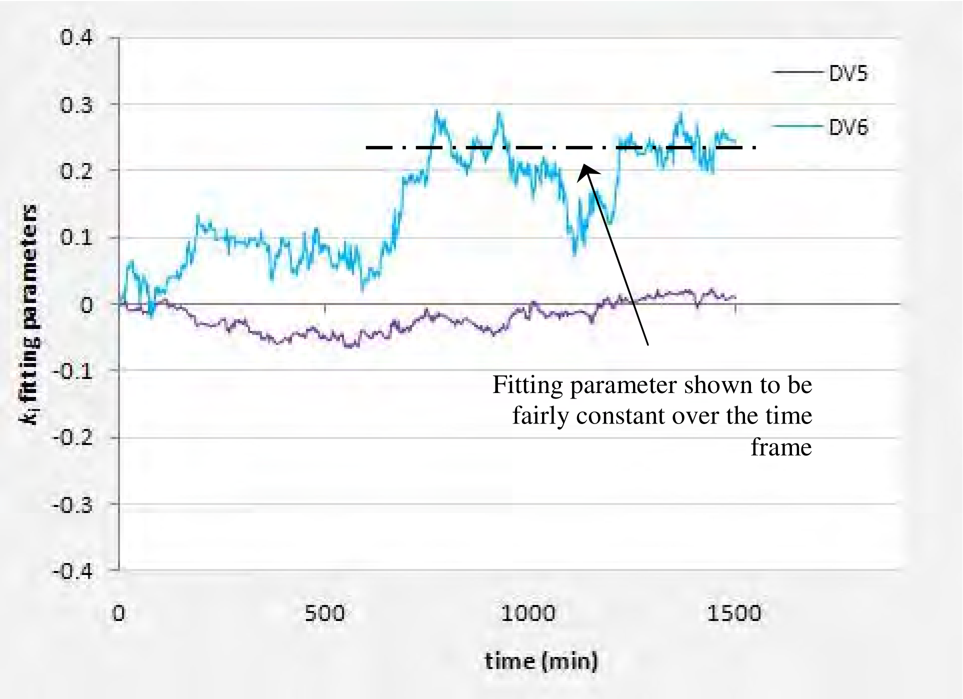
8.2.2.2. Effect of DVs on CV3

The results pertaining to the respective DVs involved are shown below:



(a) (b)

Figure 8.39: Correlation Plots for CV3 related to DVs



Fitting parameter shown to be fairly constant over the time frame

Figure 8.40: Moving Window regression plot for CV3 and DVs

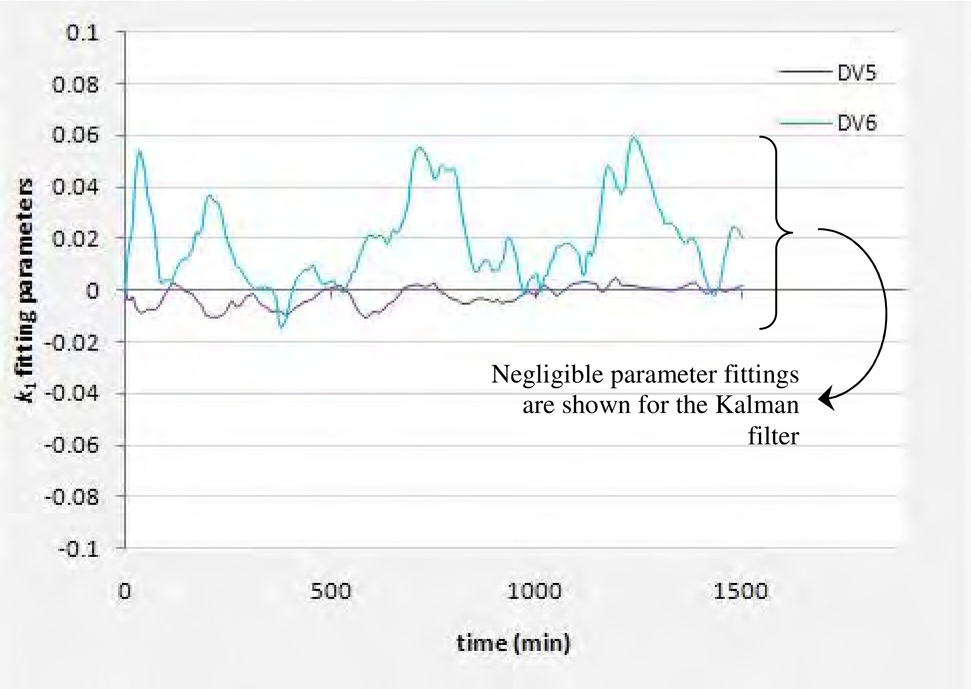


Figure 8.41: Kalman Filter plot for CV3 and DVs

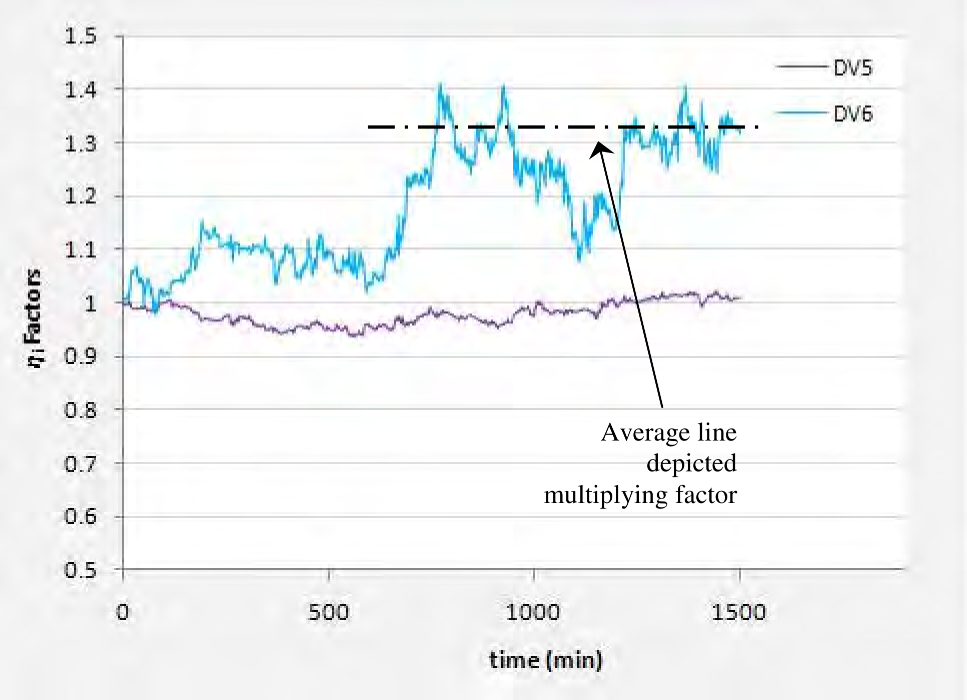


Figure 8.42: MWR factor plot for CV3 and DVs

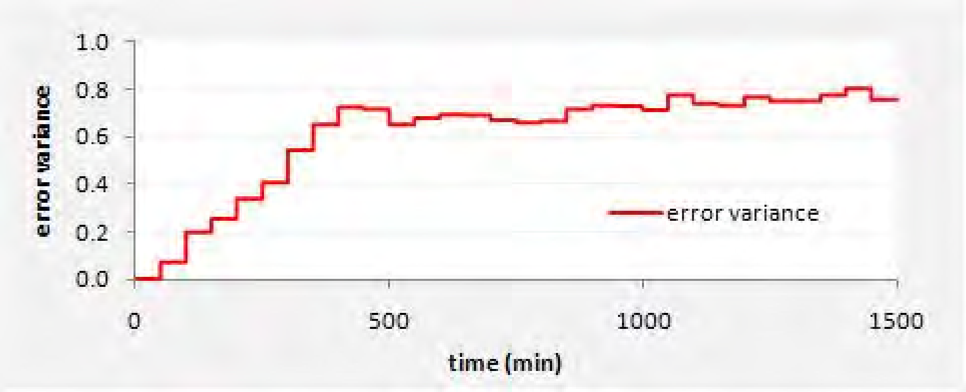


Figure 8.43: Error variance for CV3 due to the influence of the DVs

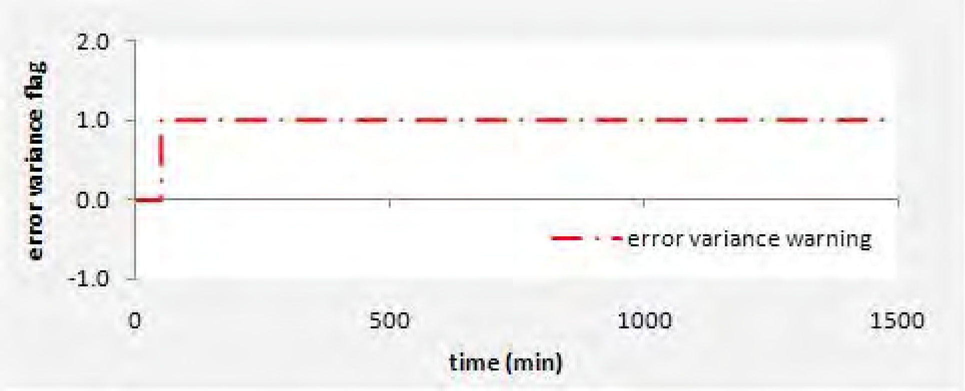


Figure 8.44: Error variance flag for CV3 due to the influence of the DVs

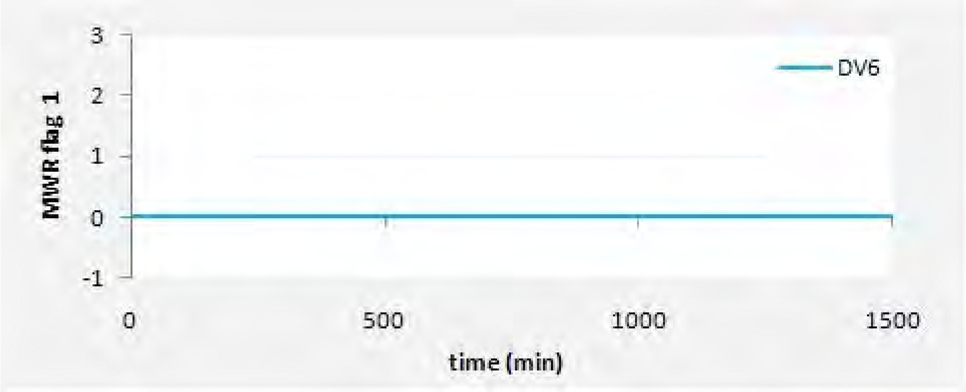


Figure 8.45: Warning flag for significant gain or time delay mismatch detection

Table 8.6:
Overall Regression parameters and multiplicative factors for DV_i-CV3

	k_5	k_6
MWR fitting parameters	-0.0208	0.2511
Kalman Filter parameters	-0.0024	0.01960
	η_5	η_6
MWR multiplying factors	0.9796	1.3353
Kalman Filter multiplying factors	0.9979	1.0120

The results obtained suggest that channel DV6-CV3 contains a gain mismatch which is within the desired bounds. The model in channel DV5-CV3 represents its portion of the plant fairly accurately.

The top column temperature manipulates the amount of reflux returned to the column. Reflux represents cooled, condensed top product returned to the tower top and, as such it is being reprocessed. If the top temperature is increased the reflux return will be increased. If the models were found to be inaccurate, this top column temperature may possibly rise to values outside the columns mode of operation. This could possibly result in a too high reflux ratio which may cause flooding in the tower resulting in poor separation and causing 'off-spec' products throughout the system.

8.2.3. Column Differential Pressure in column V-2 – CV4

The column pressure is important as it affects the boiling point temperature of the overhead liquid products. It is controlled by MV1, MV2 and MV3. The disturbance variables acting on this CV are DV5 and DV6. The models in the channels related to CV4 are divided into 2 model sets as shown before.

8.2.3.1. Effect of MVs on CV4

The column feed (MV1), petrol bypass (MV2) as well as the reboiler steam (MV3) serve to regulate the column pressure in V-2. The results pertaining to these model channels are shown below:

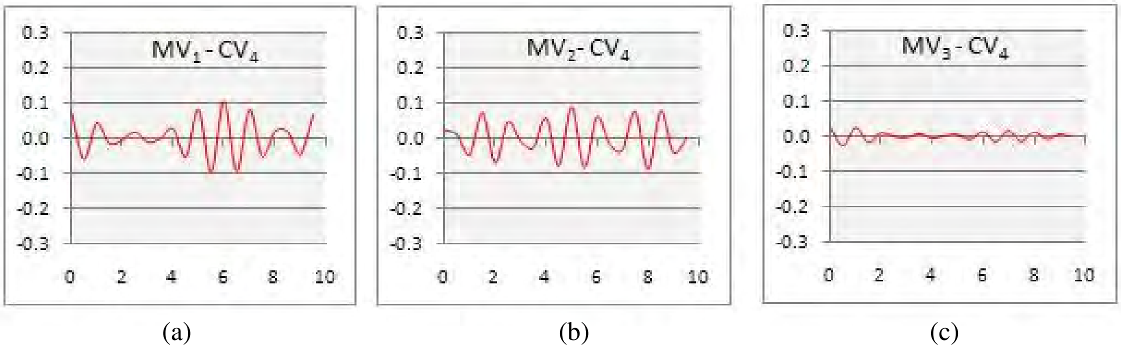


Figure 8.46: Correlation plots for CV4 related to MVs

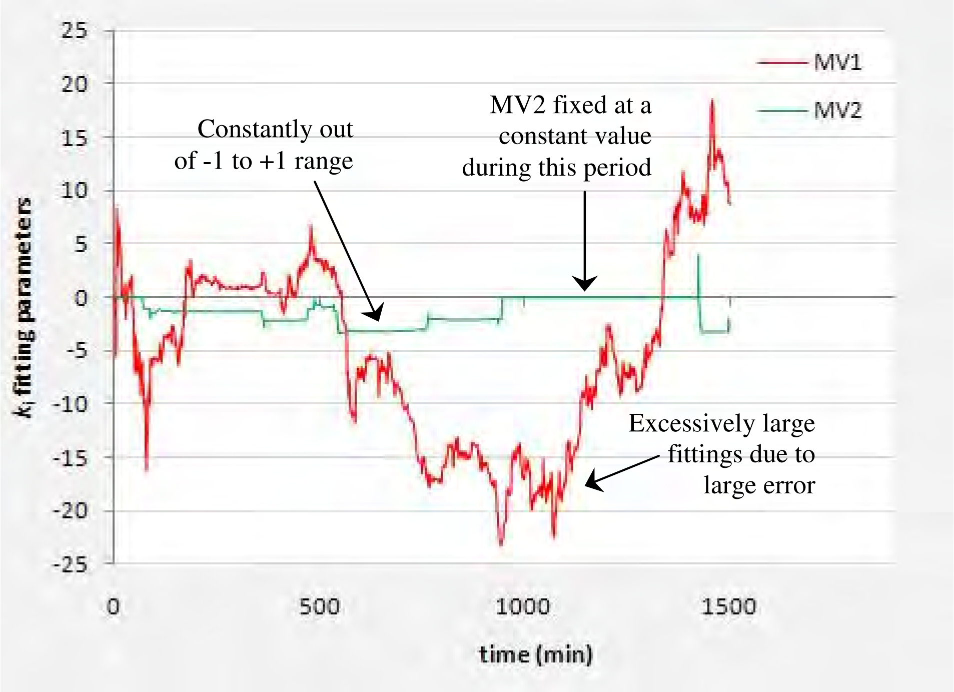


Figure 8.47: Moving Window Regression plot for CV4 related to MV1 and MV2

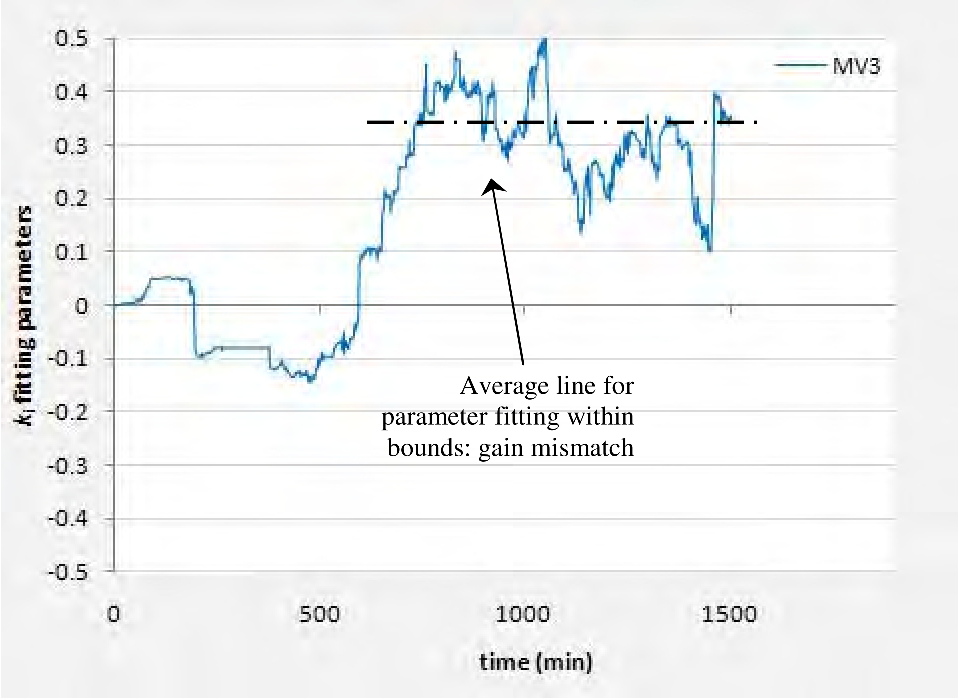


Figure 8.48: Moving Window Regression plot for CV4 related to MV3

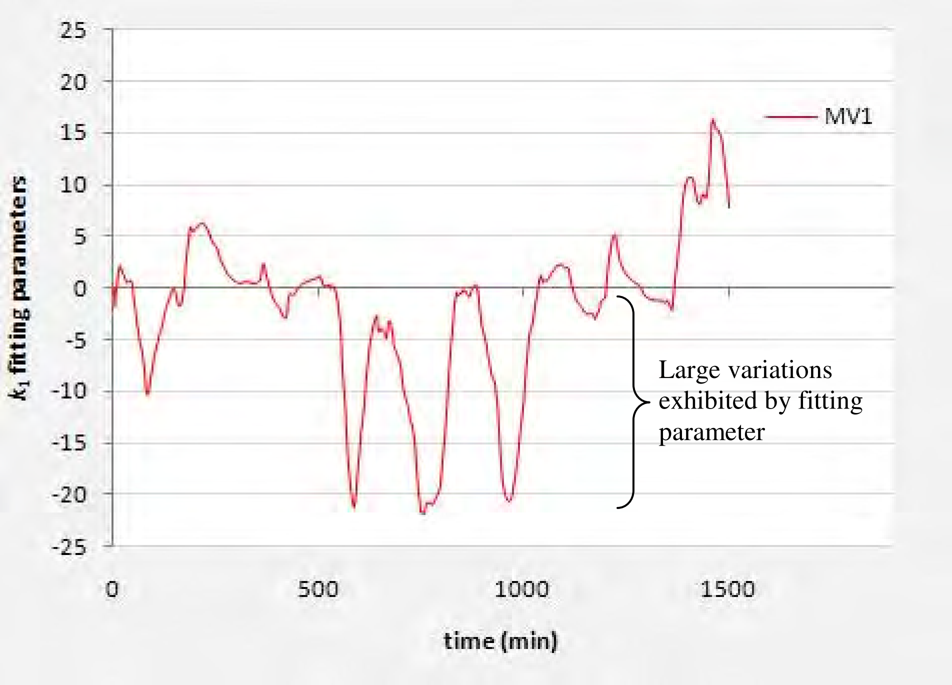


Figure 8.49: Kalman Filter plot for CV4 related to MV1

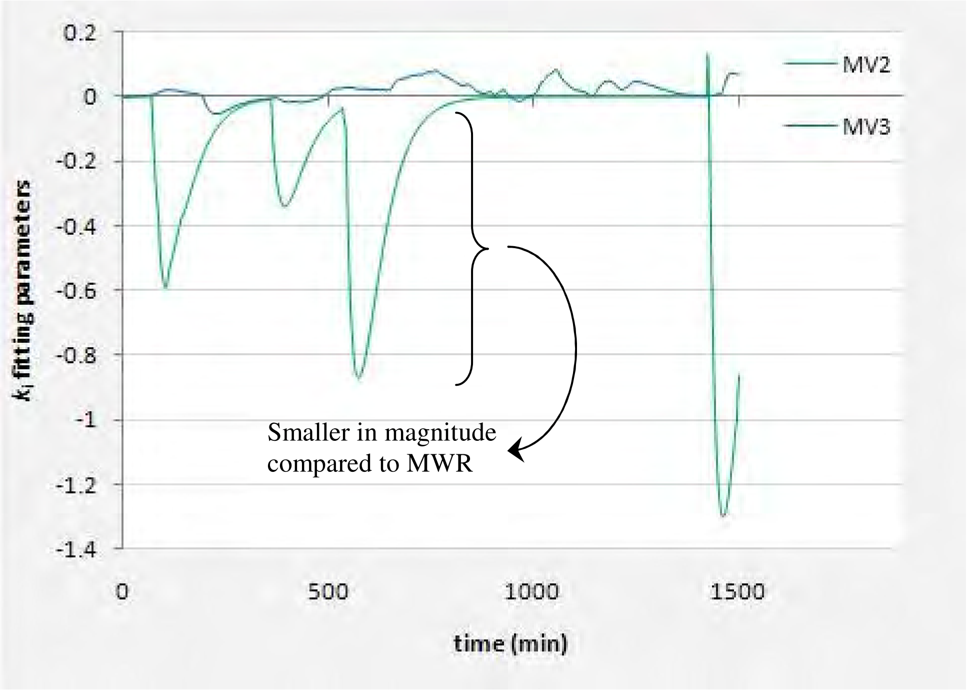


Figure 8.50: Kalman Filter plot for CV4 related to MV2 and MV3

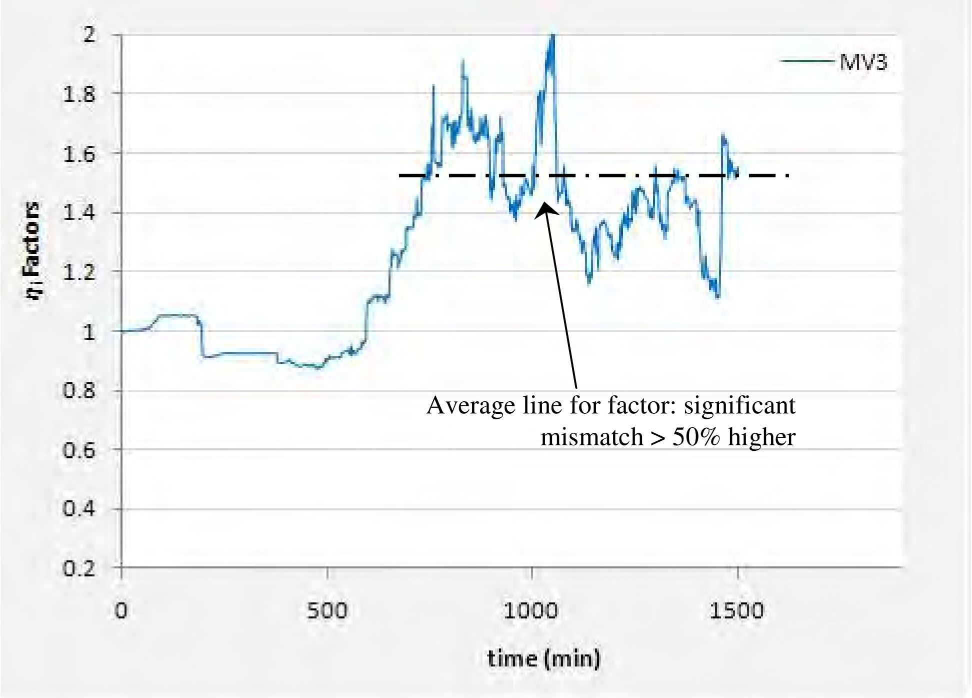


Figure 8.51: MWR factor plot for CV4 related to MV3

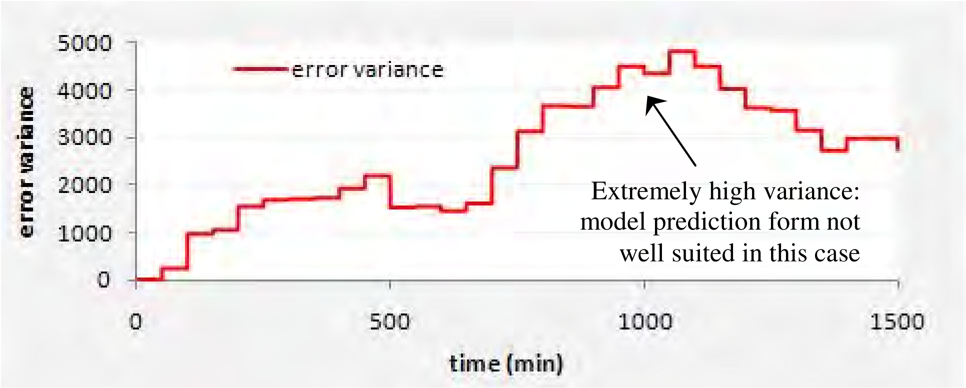
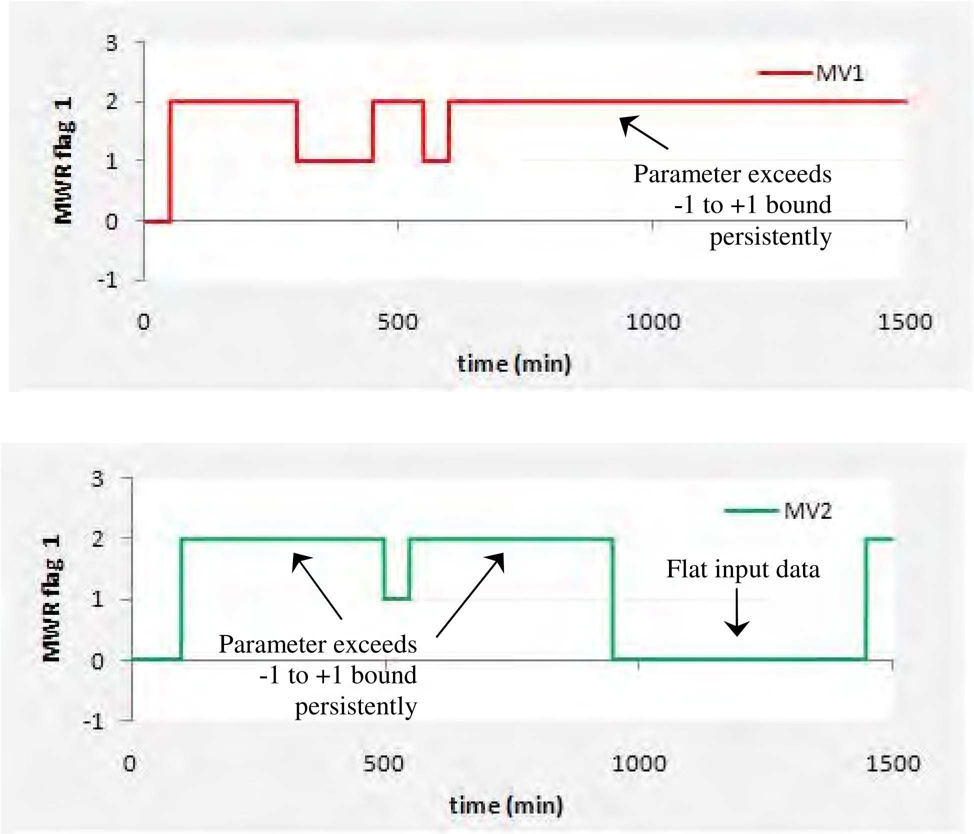
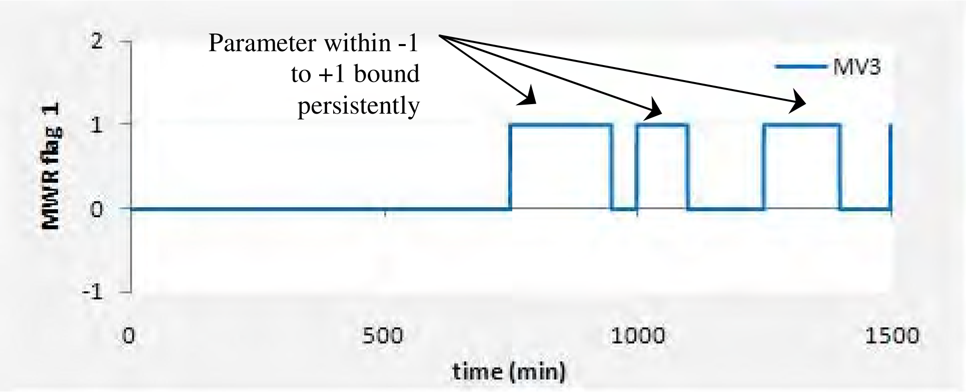


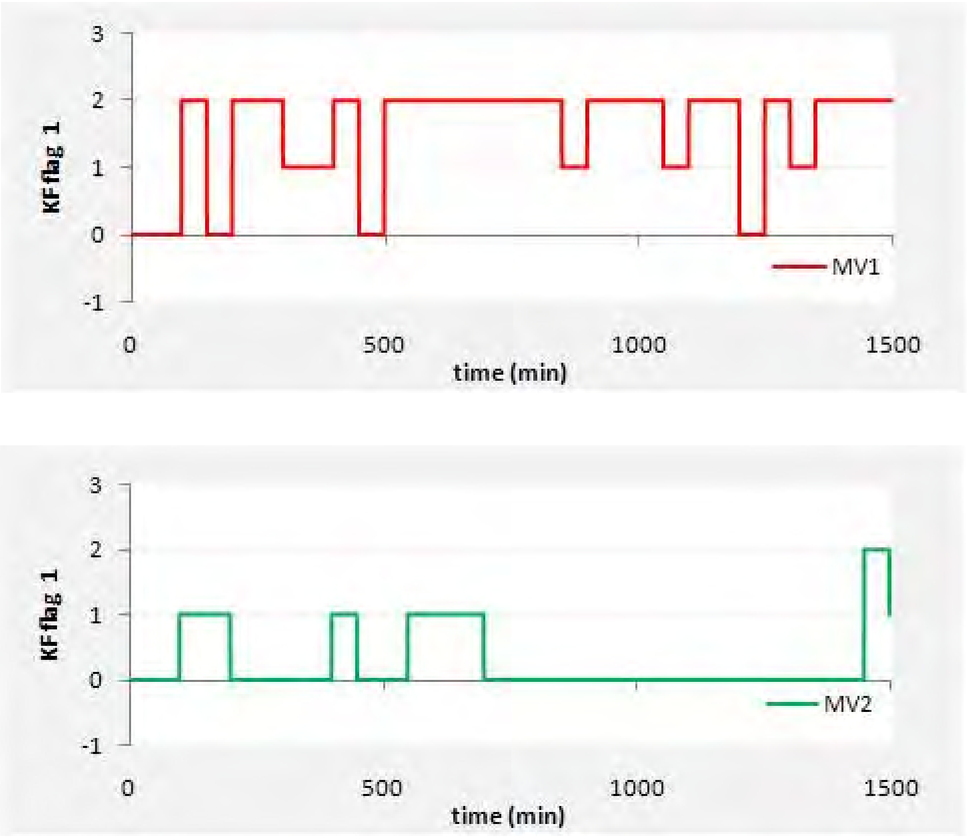
Figure 8.52: Error variance for CV4 due to the influence of the MVs





Parameter within -1 to +1 bound persistently

Figure 8.53: Warning flags for CV4 for gain or time delay mismatch: MWR



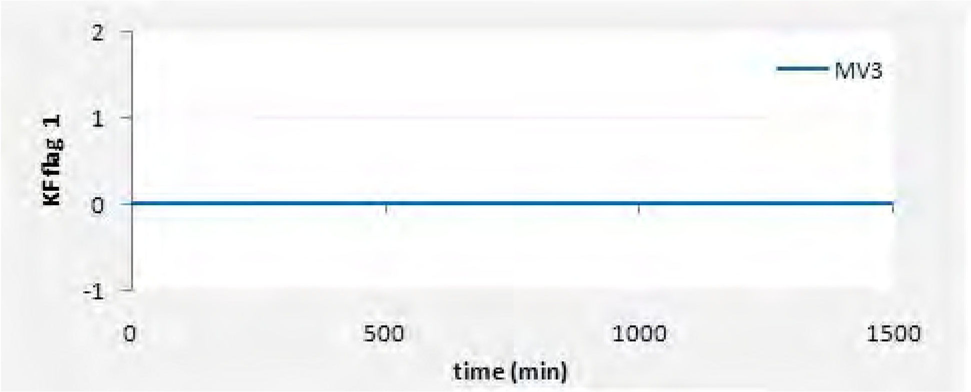


Figure 8.54: Warning flags for CV4 for error gain or time delay mismatch: KF

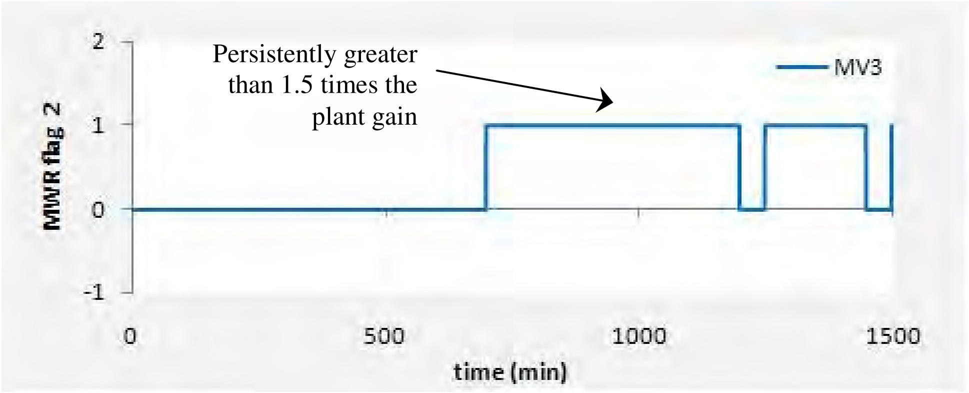


Figure 8.55: Warning flag for gain greater than 50% of plant gain

The correlation plots in figure 8.46 suggest that channels MV1-CV4 and MV2-CV4 contain a mismatch. The regression plots illustrate that the parameters fitted for these models are excessively large. This is due to the fact that the error variance (figure 8.52) is exceptionally high. This suggests that the current model output prediction form is not well suited in the case where the output covers a range of negative and positive values as it fails to track the plant output adequately. This significantly large variance in the error obtained could also be owed to the fact that the input MV2 shows very little movement over large periods of time. Each model relating to CV4 contains time delay terms of zero (refer to row 4 in the model matrix found in Appendix B). It can be stated that CV4 reacts much faster to changes in the reboiler steam flow rate (MV3) than to changes in the column feed (MV1) or the petrol bypass flow (MV2). As a result, the parameter fitting in channel MV3-CV4 is not formed as a consequence of a time delay mismatch (lies within the range of -1 to +1), whereas the fitting parameters obtained for MV1-CV4 and MV2-CV4 can be owed to a significant time delay mismatch (refer to figure 8.53). Figure 8.54 presents the warning mechanism issued for the parameters obtained through the Kalman Filter computation. This agrees with the results obtained for the moving window regression, apart from the warning issued for channel MV3-CV4. The Kalman Filter does not recognize that a mismatch is present in this channel as the bulk of the error formed is due to the models in channels MV1-CV4 and MV2-CV4.

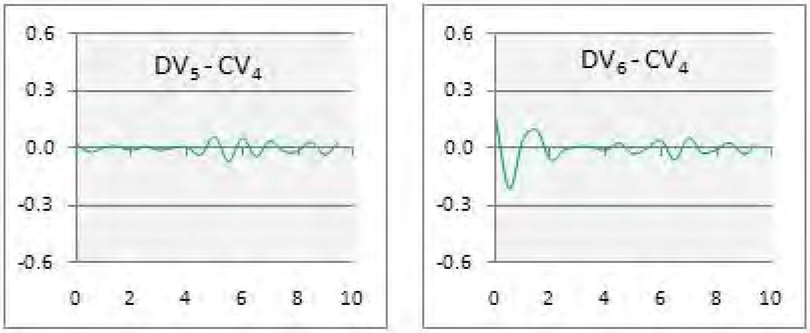
The mismatch in channel MV3-CV4 is tested for its significance and it is found to be greater than 1.5 times the plant gain. The overall fitting parameter and multiplicative factor for this channel is displayed in the table below:

Table 8.7:
Overall Regression parameter and multiplicative factor for MV3-CV4

	k_3
MWR fitting parameter	0.3452
	η_3
MWR multiplying factor	1.5271

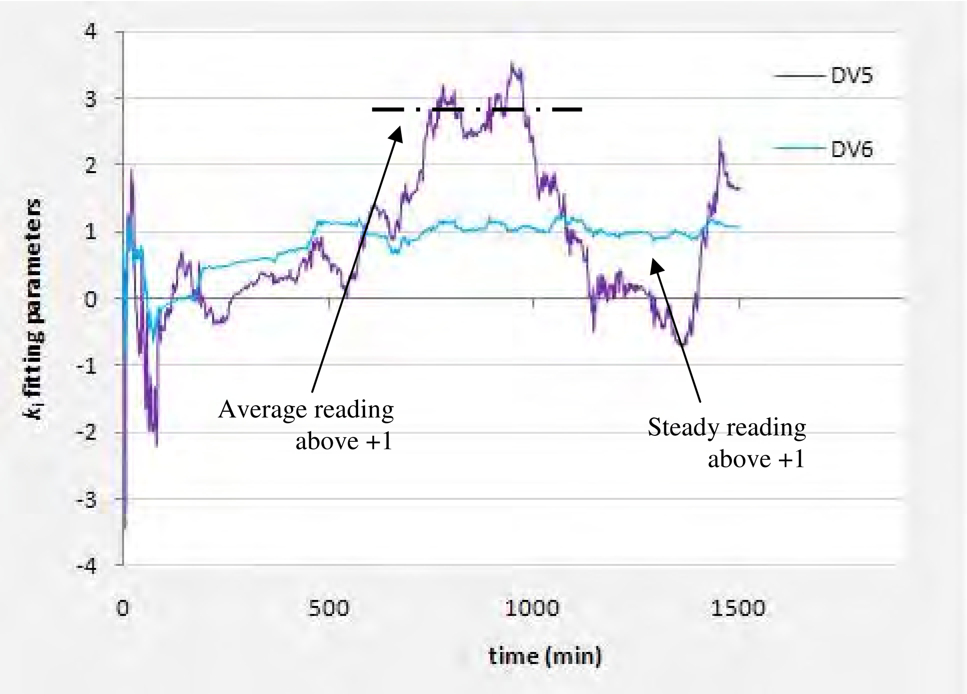
8.2.3.2. Effect of DVs on CV4

The results for each DV related to CV4 are shown below:



(a) (b)

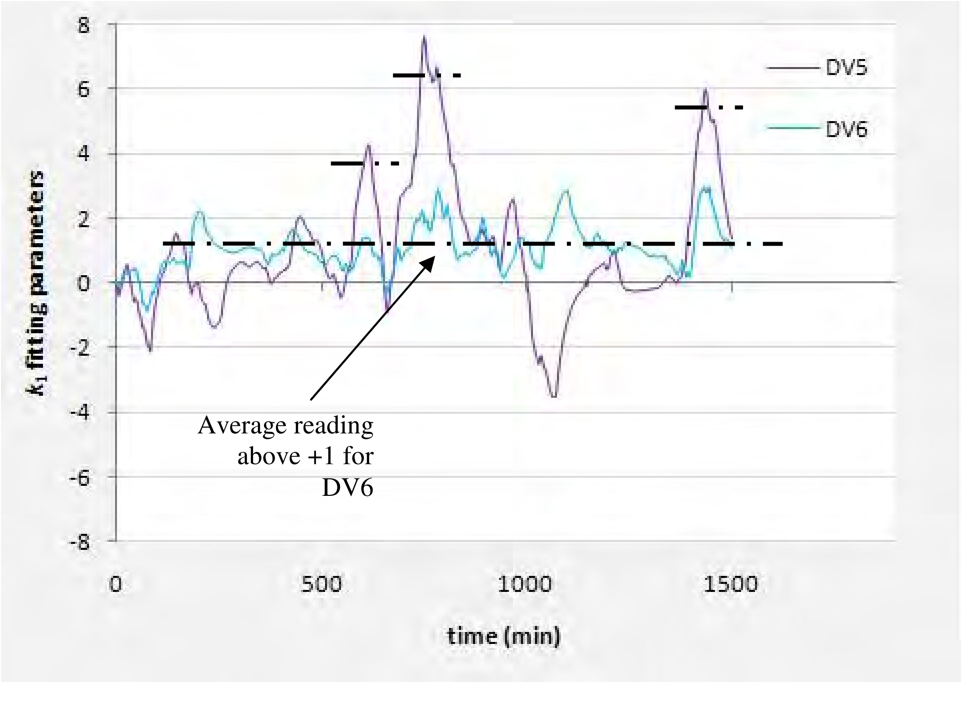
Figure 8.56: Correlation Plots for CV4 related to its DVs



Average reading above +1

Steady reading above +1

Figure 8.57: Moving Window Regression plot for CV4 related to its DVs



Average reading above +1 for DV6

Figure 8.58: Kalman Filter plot for CV4 related to its DVs

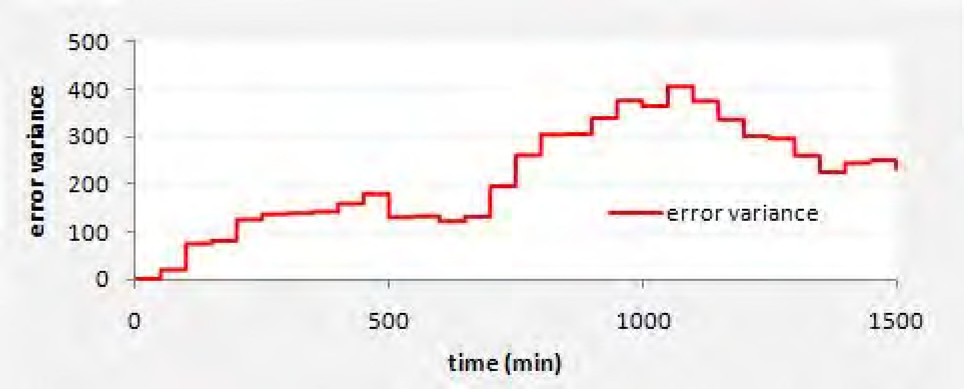


Figure 8.59: Error variance for CV4 due to the influence of the DVs

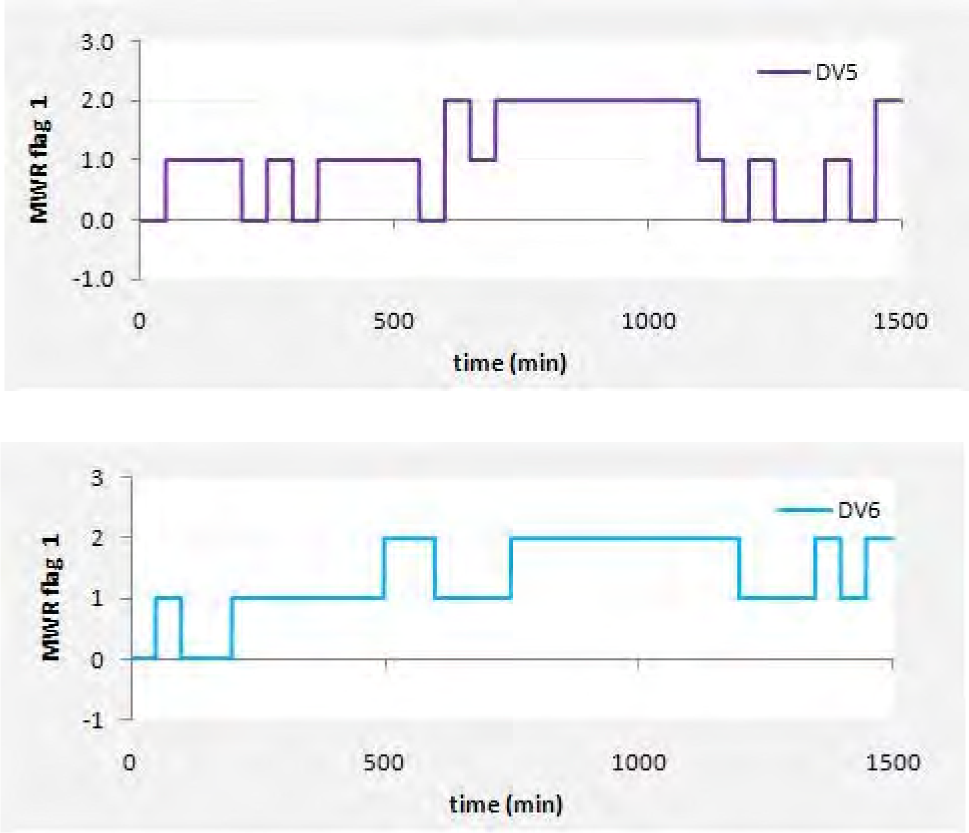


Figure 8.60: MWR Warning flag for gain of time delay mismatch

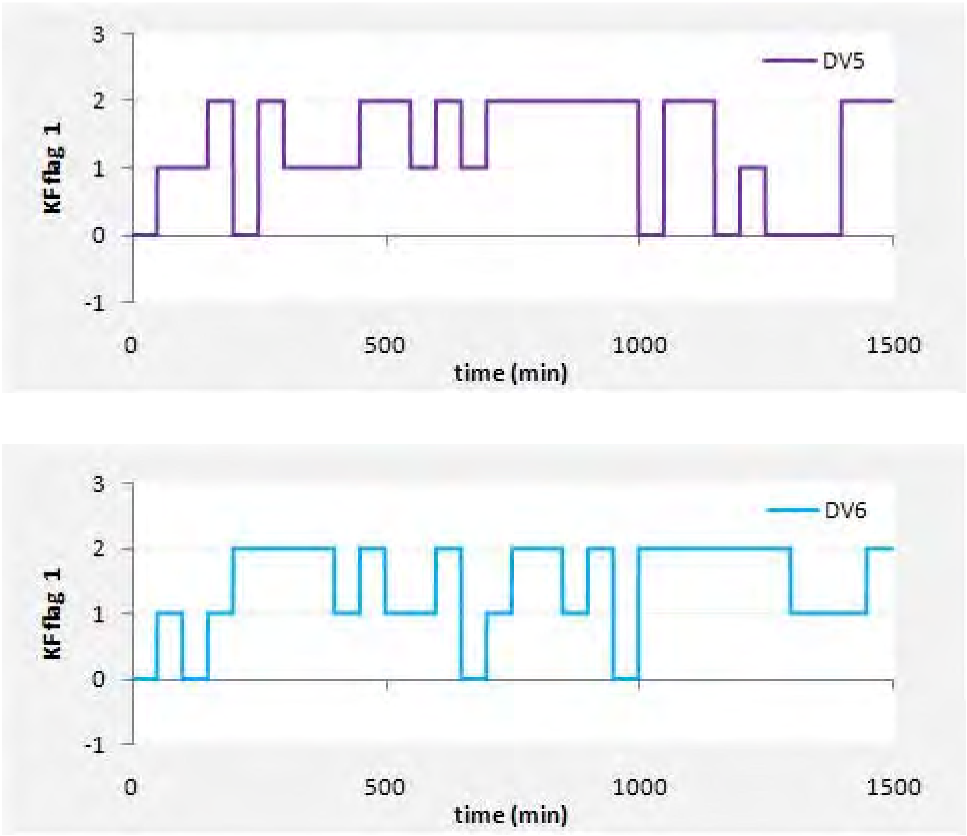


Figure 8.61: KF Warning flag for gain of time delay mismatch for CV4 related to its DVs

The error variance is again shown to be extremely high (figure 8.59). This again emphasizes the trouble that the prediction model output form (equation error differential form) would have in tracking the plant output, which covers both a negative and positive range. In this case, both regression methods suggest that the mismatch present in channels DV5-CV4 and DV6-CV4 are both time delay mismatches. This result is further emphasized by the persistent warning flag value of 2 being issued (refer to figures 8.60 and 8.61).

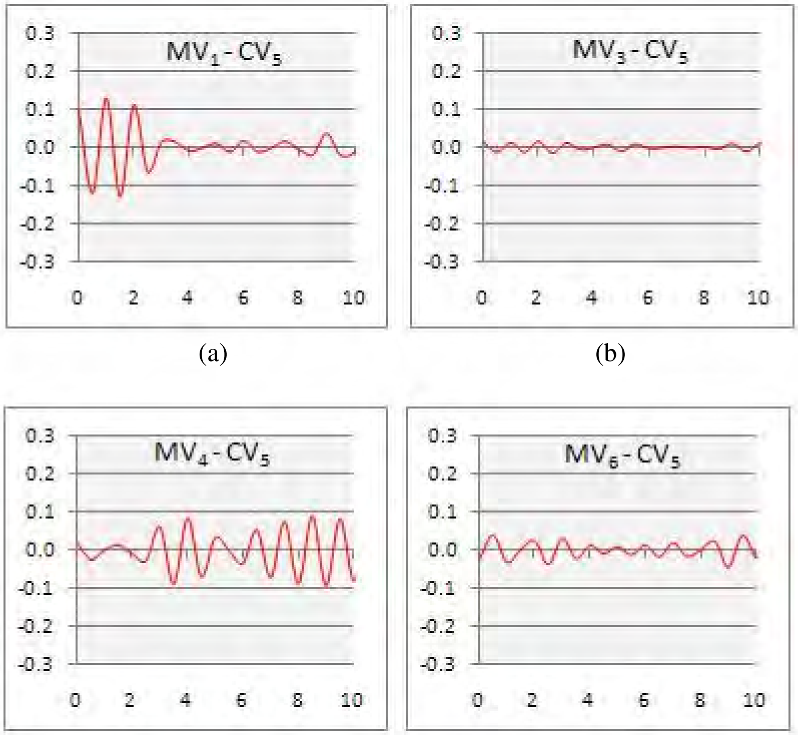
If these model mismatches shown for all the models related to CV4 cause the column pressure to increase beyond its upper limit, the consequence is that the liquid top product will start to boil at higher temperatures (an increase in the column pressure causes an increase in the boiling point temperature of the top product constituents).

8.2.4. Bottoms level in column V-2 – CV5

The bottoms level in the petrol debutanizer column is controlled by MV1, MV2, MV4 and MV6. It is also affected by DV5 and DV6.

8.2.4.1. Effect of MVs on CV5

The results related to these model channels are shown below. The bottoms level is required to be operated adequately (by the control of the aforementioned MVs) as the bottoms level ensures that vapor does not exit through the bottom of the column.



(a)

(b)

(c)

(d)

Figure 8.62: Correlation Plots for CV5 related to its MVs

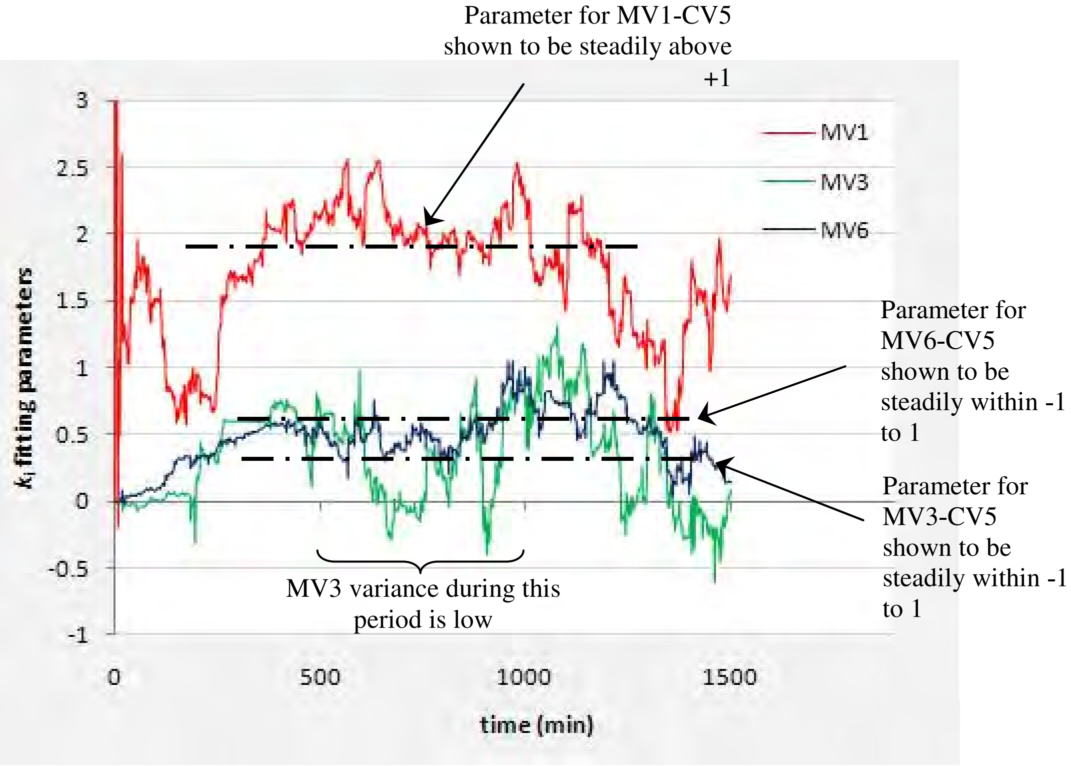


Figure 8.63: Moving Window Regression plot for CV5 related to its MVs

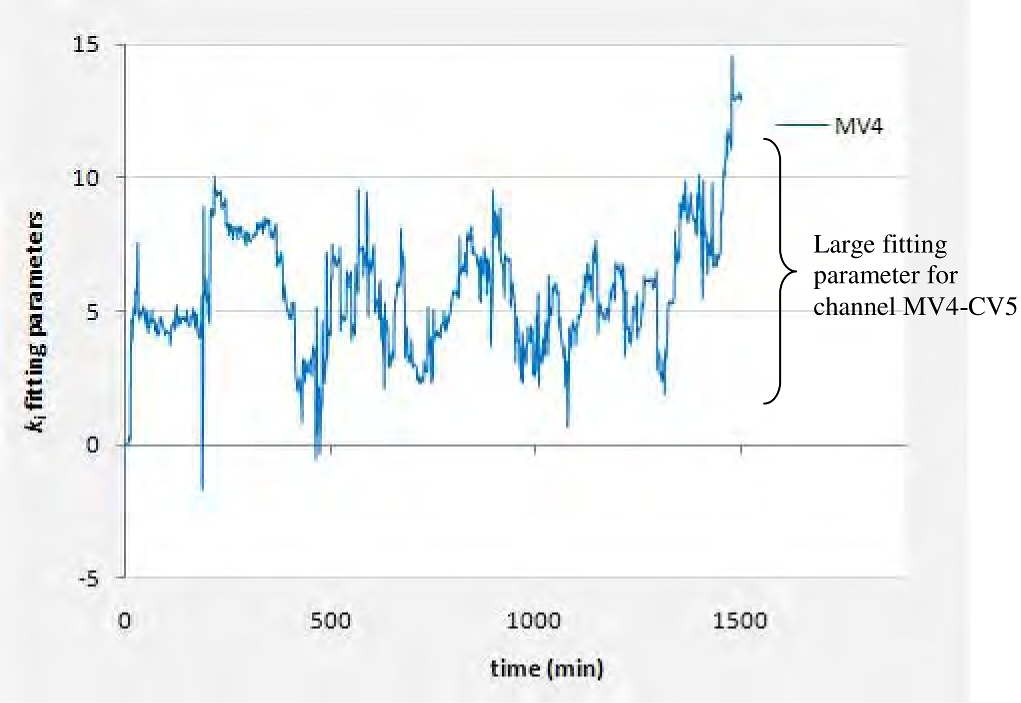


Figure 8.64: Moving Window Regression plot for CV5 related to MV4

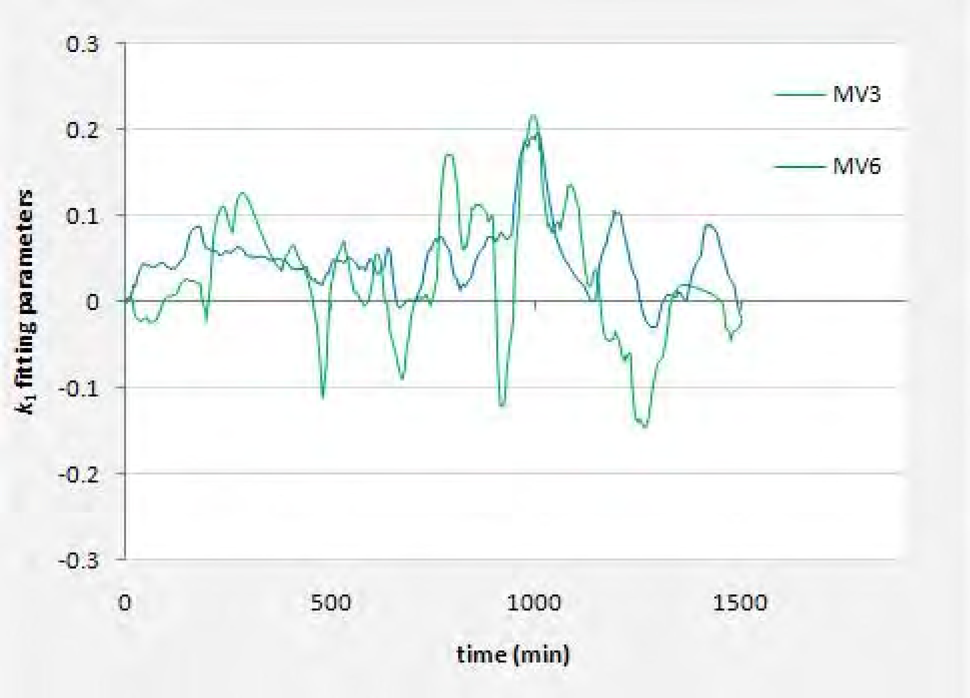


Figure 8.65: Kalman Filter plot for CV5 related to MV3 and MV6

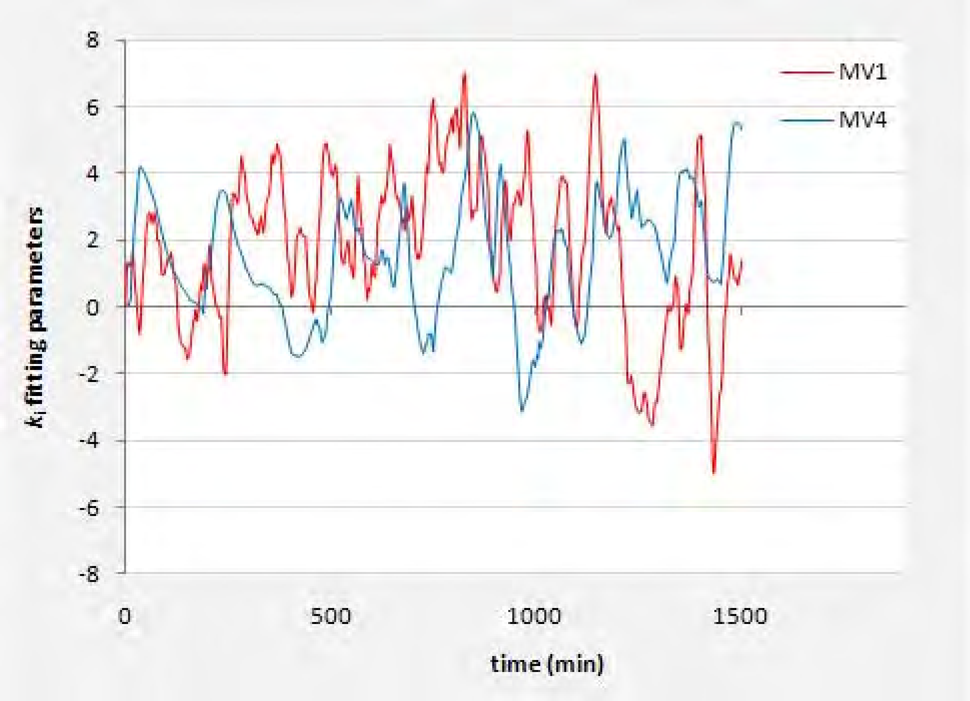


Figure 8.66: Kalman Filter plot for CV5 related to MV1 and MV4

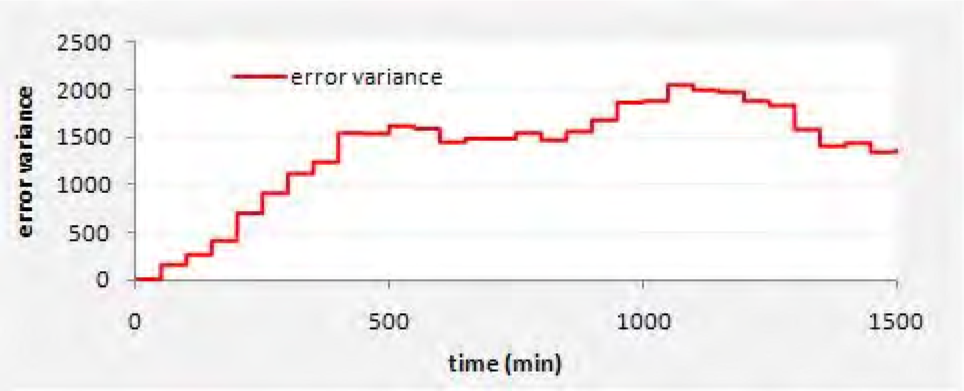
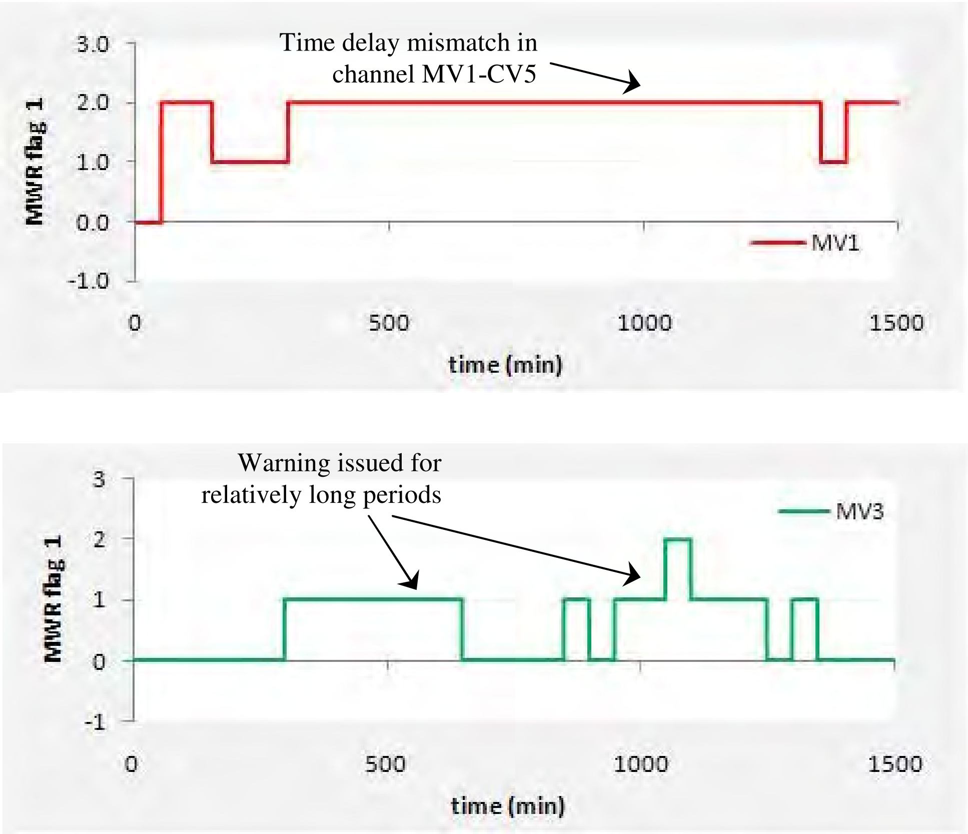
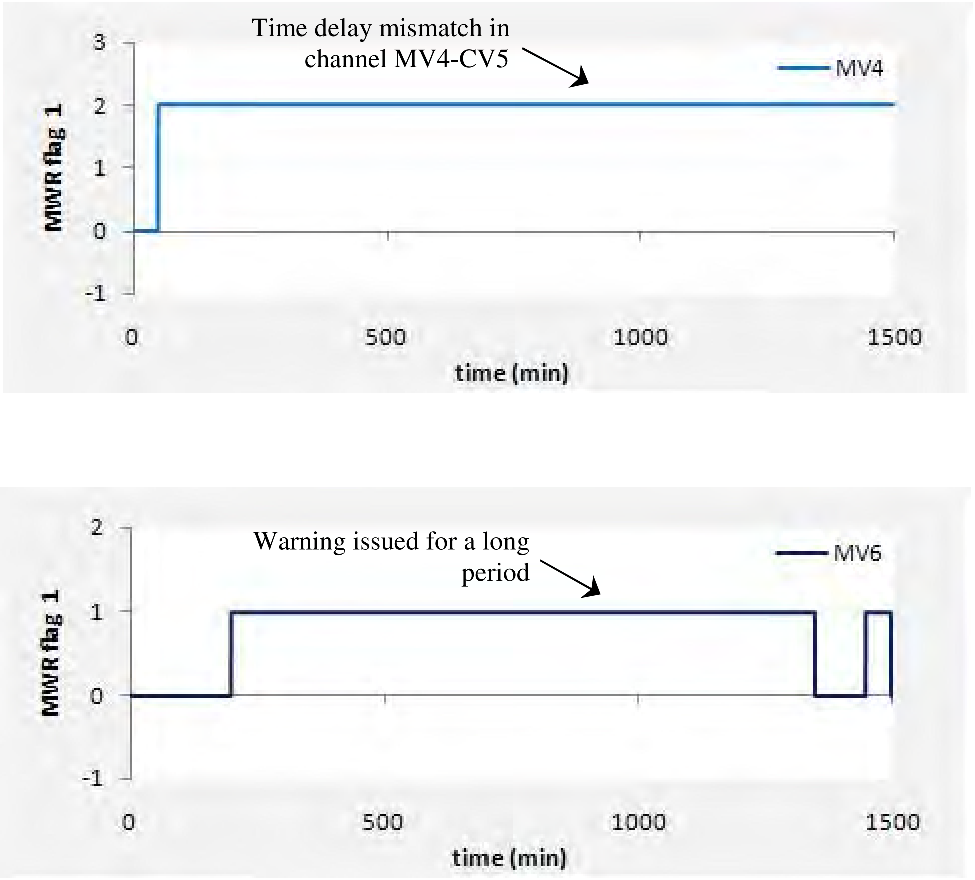


Figure 8.67: Error Variance for CV5 related to its MVs


Time delay mismatch in channel MV1-CV5

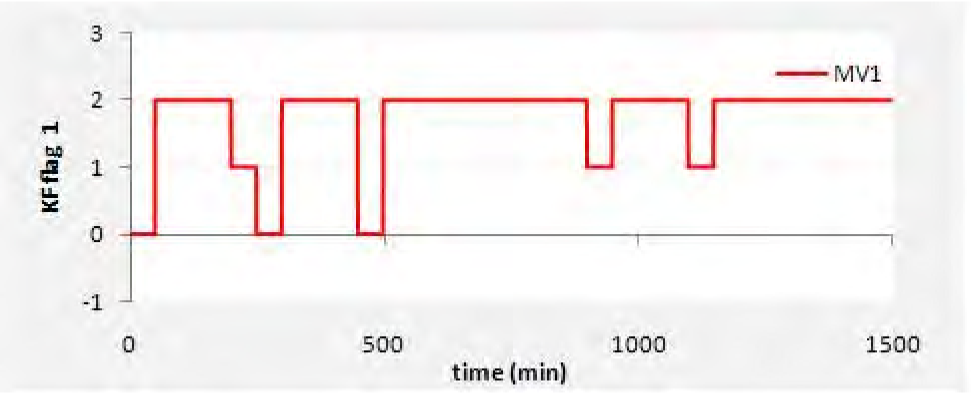
Warning issued for relatively long periods



Time delay mismatch in channel MV4-CV5

Warning issued for a long period

Figure 8.68: MWR Warning Flag for gain or time delay mismatch for CV5



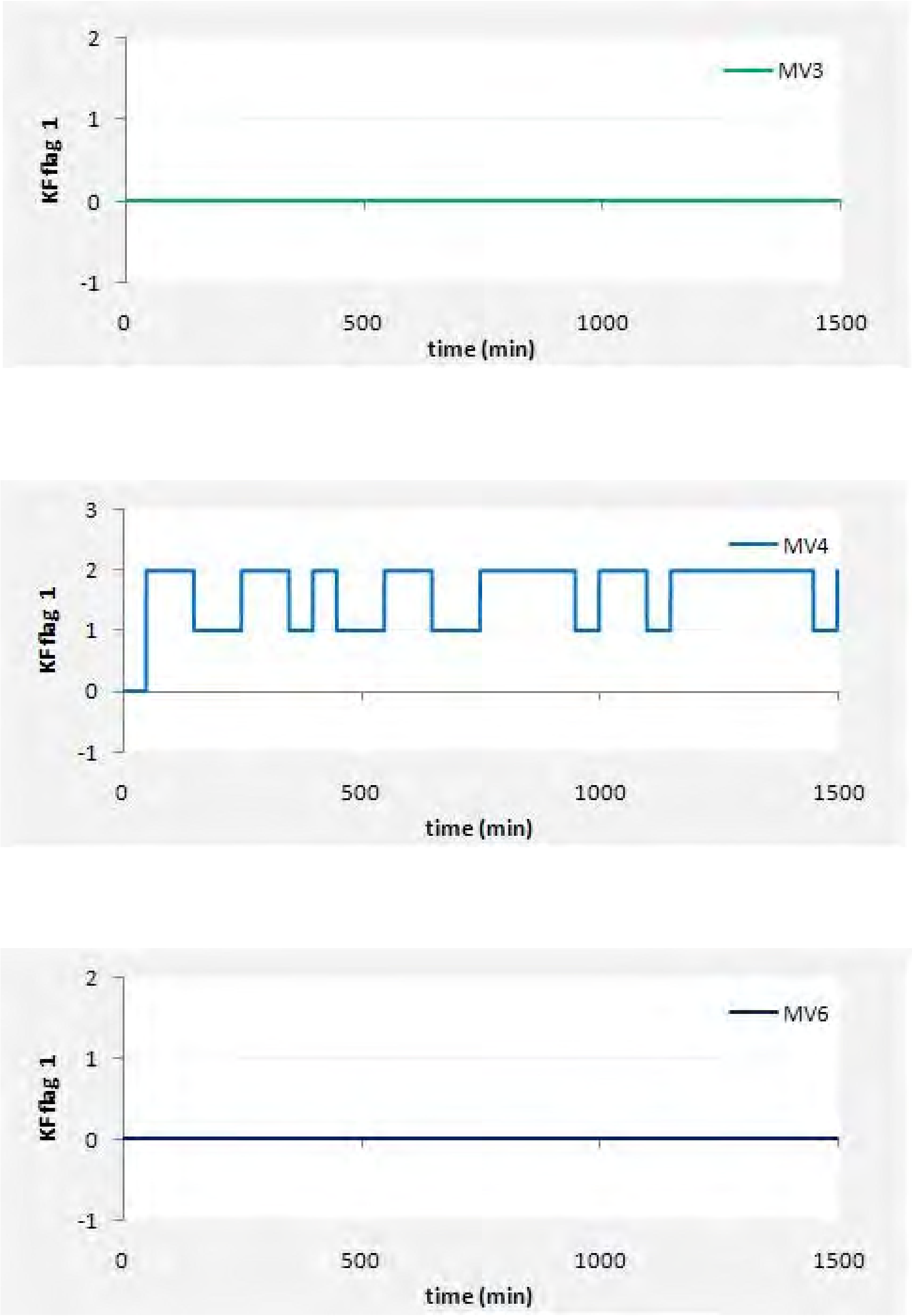


Figure 8.69: KF Warning Flag for gain or time delay mismatch for CV5

From figure 8.62, strong correlations were exhibited by channels MV1-CV5 and MV4-CV5. These strong correlations are attributed to those sources which create the bulk of the error. From figure 8.63 and 8.64, the parameters obtained for MV1-CV5 and MV4-CV5 were found to be much larger in comparison to the parameters acquired for the other models. The bulk of the large error variance in figure 8.67 is thus owed to these two channels. The large fitting parameters have been proven to be synonymous with large time delay mismatches (see figure 8.68). The Kalman filter plots in figure 8.65 and 8.66 are in agreement with these results (see figure 8.69). The results obtained for the parameters in those channels related to MV3 and MV6 suggest that the gain in each model is matched. The gains in these channels were tested for the extent of the gain mismatch and the following results were obtained:

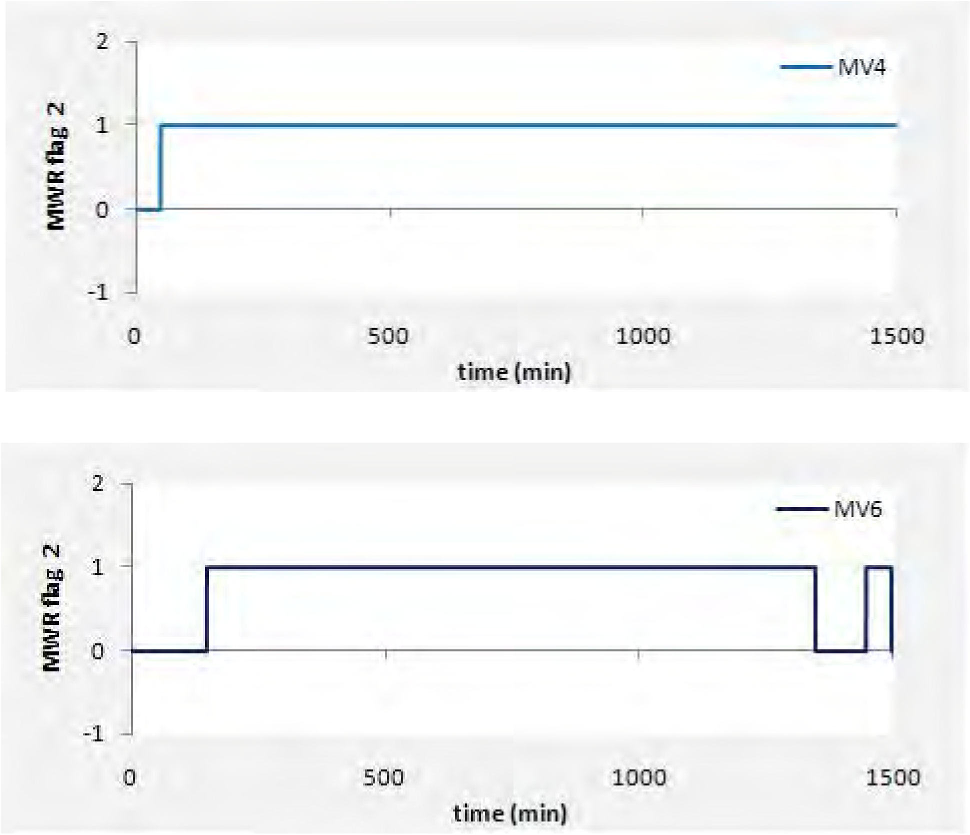


Figure 8.70: MWR Warning Flag for significant gain mismatch

These warning mechanisms suggest that the gain is higher than 1.5 times the plant gain. The actual fitting parameters and the multiplicative factors are displayed in the table below:

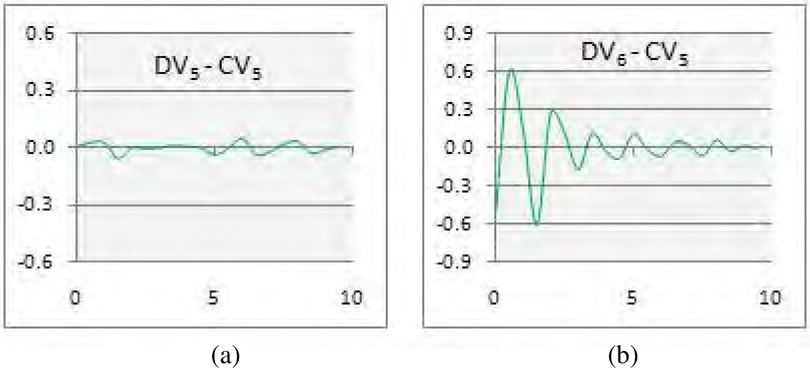
Table 8.8:
Overall Regression parameters and multiplicative factors for MV_i-CV5

	k_4	k_6
MWR fitting parameter	0.6151	0.3527
	η_4	η_6
MWR multiplying factor	2.5981	1.5449

The overall diagnosis of these models is as follows: channels MV1-CV5 and MV4-CV5 exhibited significantly large time delay mismatches. The remaining channels MV3-CV5 and MV6-CV5 both show that the gain is mismatched significantly. All the models relating the MVs to CV5 were found to be significantly mismatched. This may adversely affect the regulation of the level in column V-2.

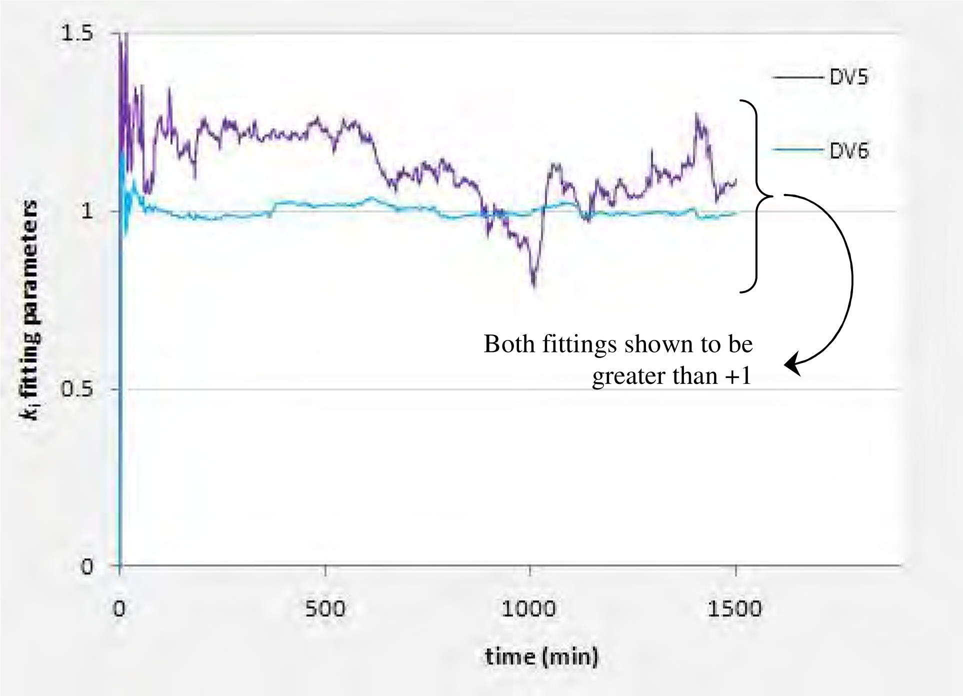
8.2.4.2. Effect of DVs on CV5

The results for these DVs are as follows:



(a) (b)

Figure 8.71: Correlation Plots for CV5 related to its DVs



Both fittings shown to be greater than +1

Figure 8.72: Moving Window Regression plot for CV5 related to its DVs

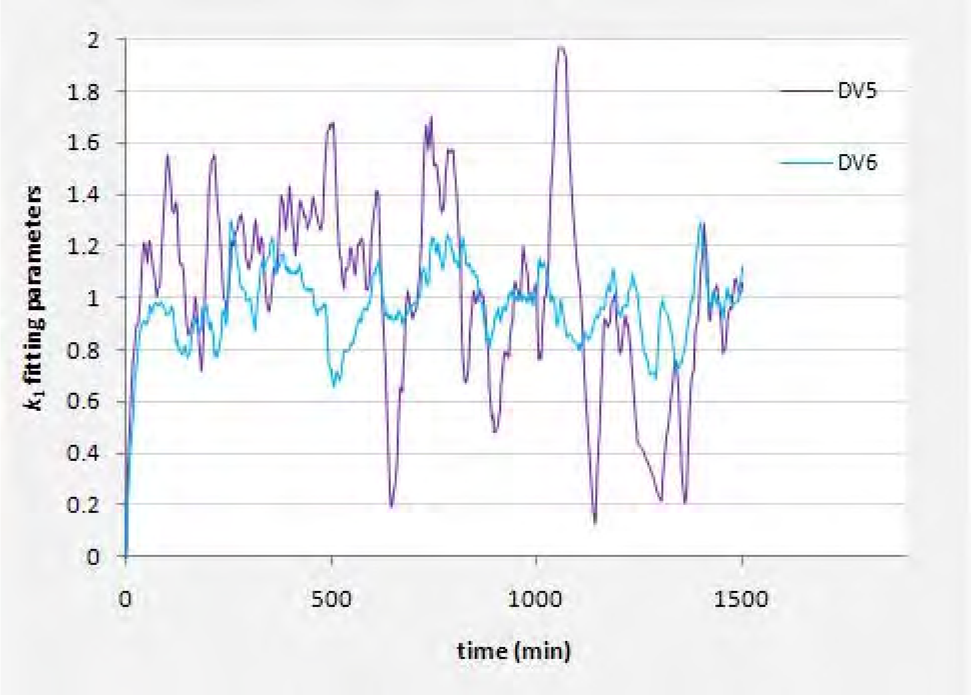


Figure 8.73: Kalman Filter plot for CV5 related to its DVs

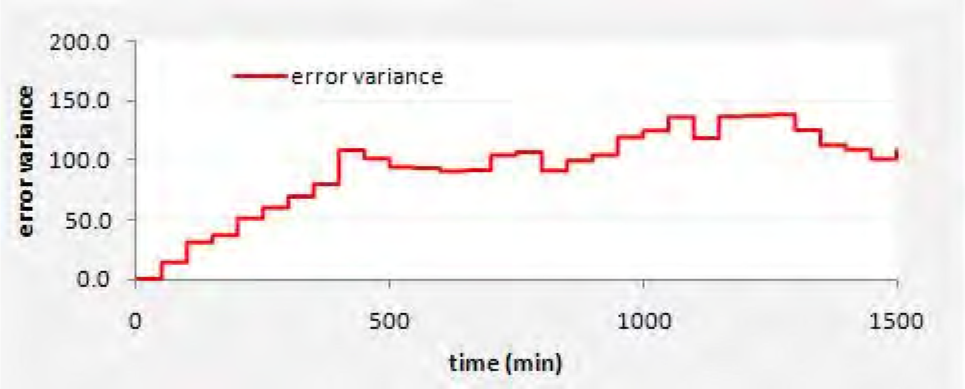


Figure 8.74: Error Variance for CV5 related to its DVs

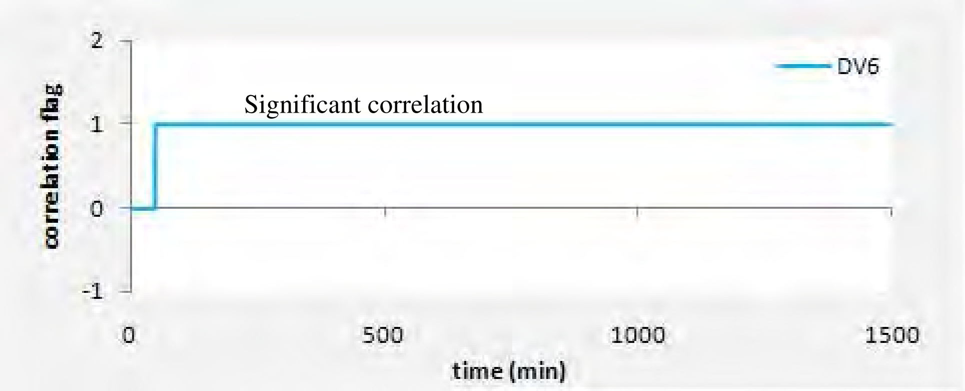
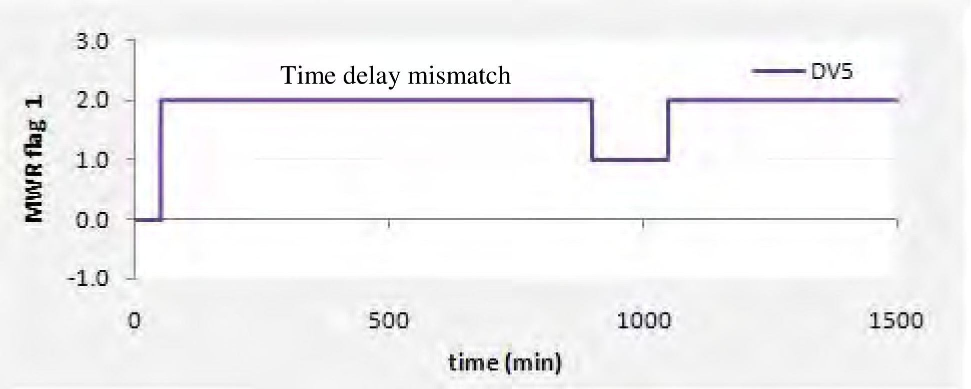


Figure 8.75: Warning flag for correlation bounds violation for DV6



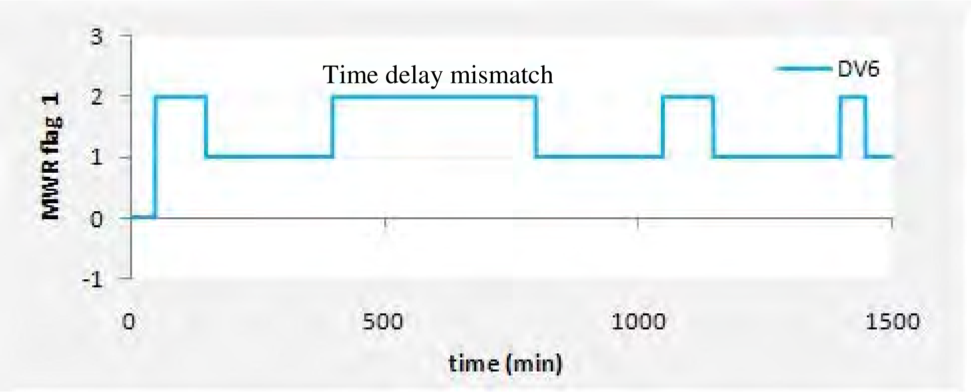


Figure 8.76: Warning flag for gain or time delay mismatch related to DVs

A strong correlation is shown in figure 8.71(b) for channel DV6-CV5. This correlation is shown to be in violation of the bounds set for the correlation plots (figure 8.75). A smaller correlation is shown for channel DV5-CV5, which may be assumed to be due to the influence of noise. The moving window regression plot illustrates that both fitting parameters are greater than +1 with the fitting parameter related to DV5-CV5 being the larger of the two. This confirms the assumption that the unmeasured disturbances influence DV5 to a greater extent (in comparison to DV6) resulting in the error variance (figure 8.74) owing largely to the output contribution of model DV5-CV5. The warning mechanisms in figure 8.76 confirm that a time delay mismatch is persistent over a long period for each DV.

All the models related to CV5 contain a degree of mismatch. This could result in high liquid levels in the column sump or low liquid levels. In the case of high liquid levels, the column will consequently become flooded.

8.2.5. Petrol feed to next phase of operation – CV6

The bottoms product from column V-2 is controlled by MV1, MV2 and MV6. DV5 and DV6 serve as the measured disturbance variables. One of the control objectives for the petrol debutanizer is to obtain the desired petrol fraction in the bottoms product.

8.2.5.1. Effect of the MVs on CV6

The column feed, the petrol bypass and the stream that is split from CV6 regulates this controlled variable. The modeling error results in relation to the models linked to each of the above mentioned MVs are shown below:

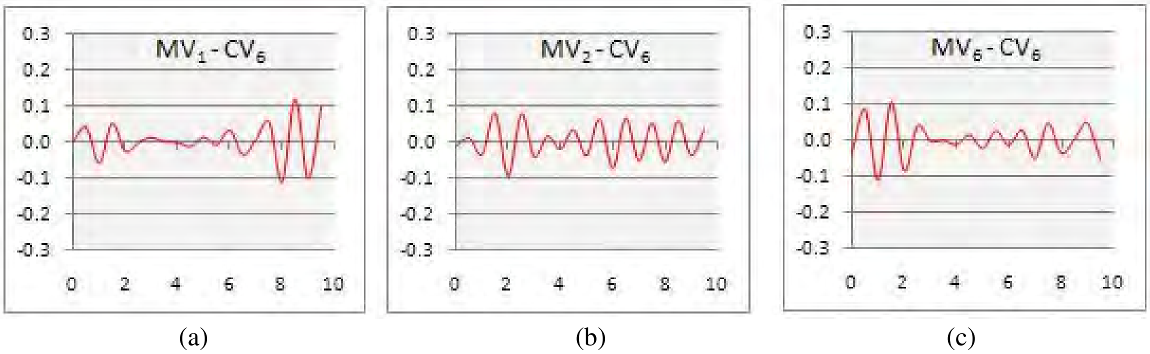


Figure 8.77: Correlation Plots for CV6 related to its MVs

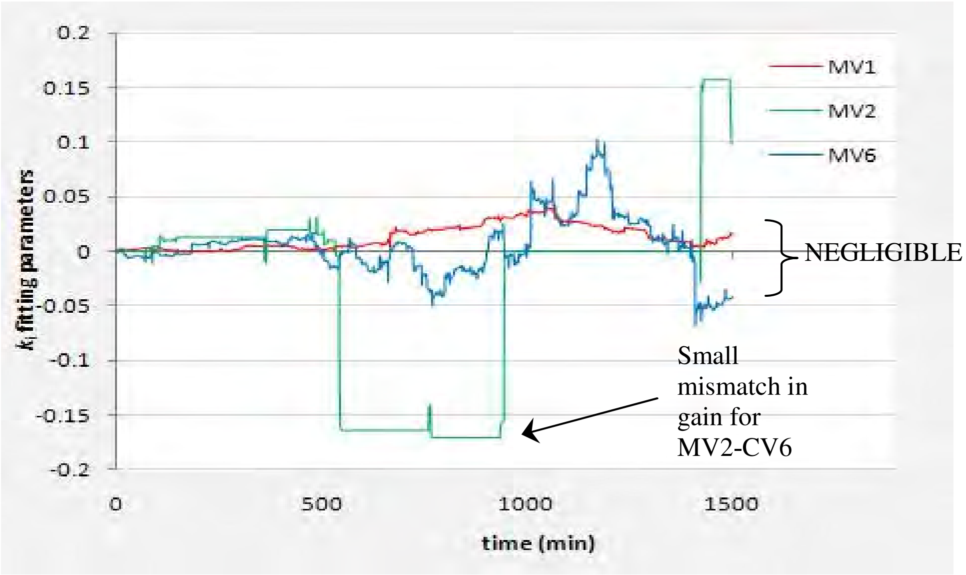


Figure 8.78: Moving Window Regression plot for CV6 related to its MVs

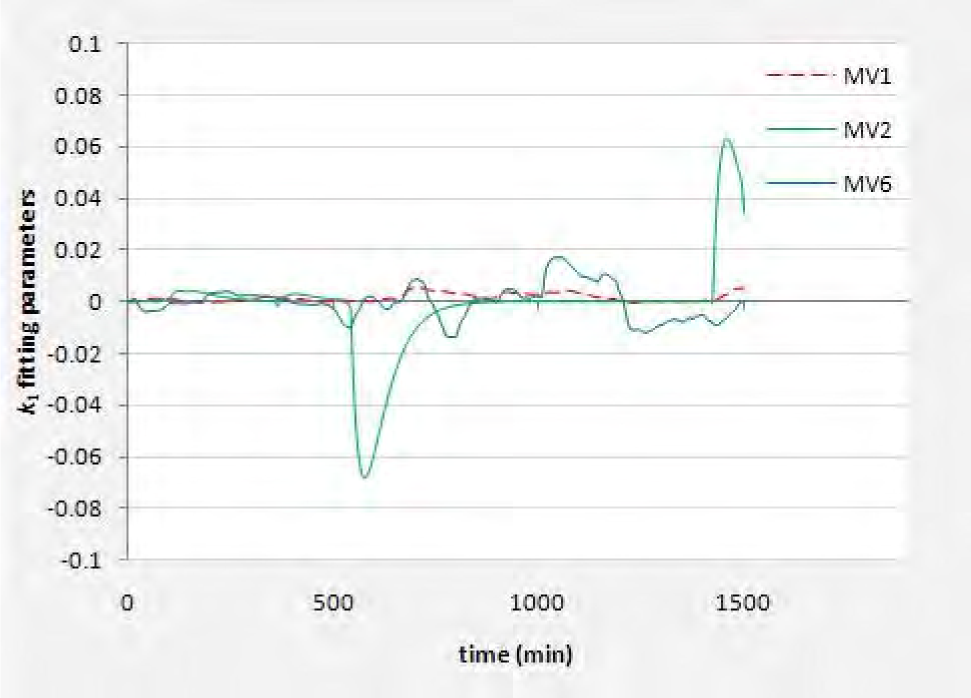


Figure 8.79: Kalman Filter plot for CV6 related to its MVs

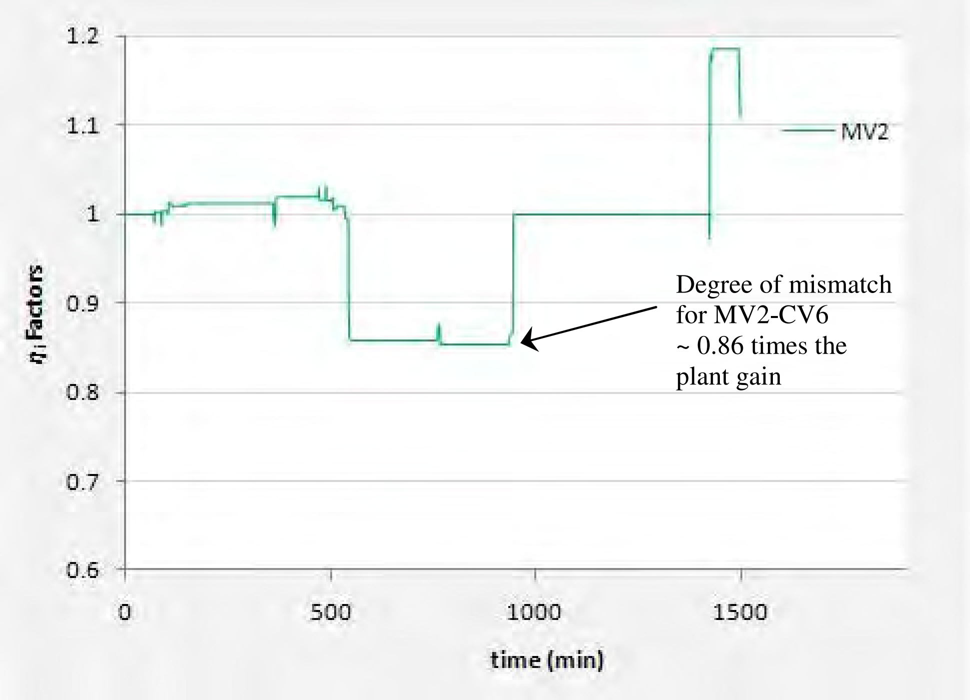


Figure 8.80: MWR factor plot for CV6 related to its MVs

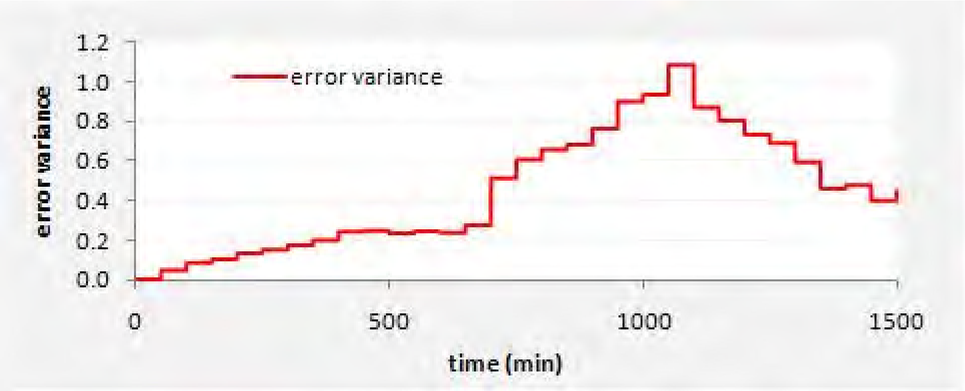


Figure 8.81: Error Variance for CV6 related to its MVs

Due to the fact that the input data for MV2 and MV6 are shown to be flat for long periods of time, the non-differential, open-loop prediction error formulation is employed; refer to equation 5.24. The correlation plots in figure 8.77 suggest that there is a mismatch in all channels tested. On the other hand the deviation away from zero for the fitting parameters in figure 8.78 and 8.79 is negligible apart from the mismatch found in channel MV2-CV6. Hence it can be deduced that the error is due to lower order coefficients present in the models. No warning is issued for the mismatch detected in channel MV2-CV6 as the model gain is found to be only ~ 0.86 times that of the actual plant gain (figure 8.80). These models represent the plant system reasonably accurately.

8.2.5.2. Effect of the DVs on CV6

The relevant results related to DV5-CV6 and DV6-CV6 is shown below:

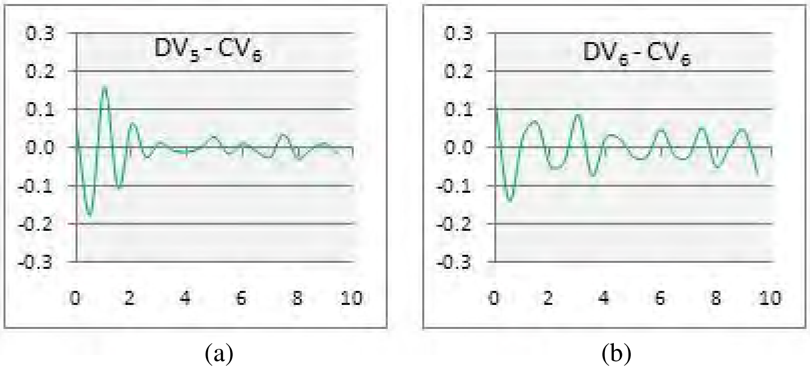


Figure 8.82: Correlation Plots for CV6 related to its DVs

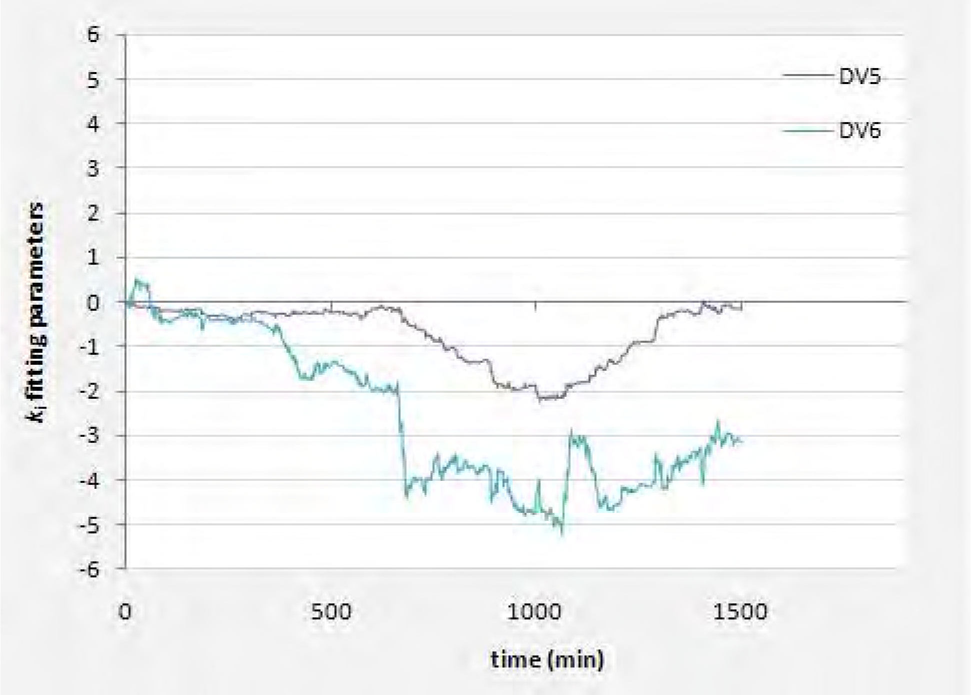


Figure 8.83: Moving Window Regression plot for CV6 related to its DVs

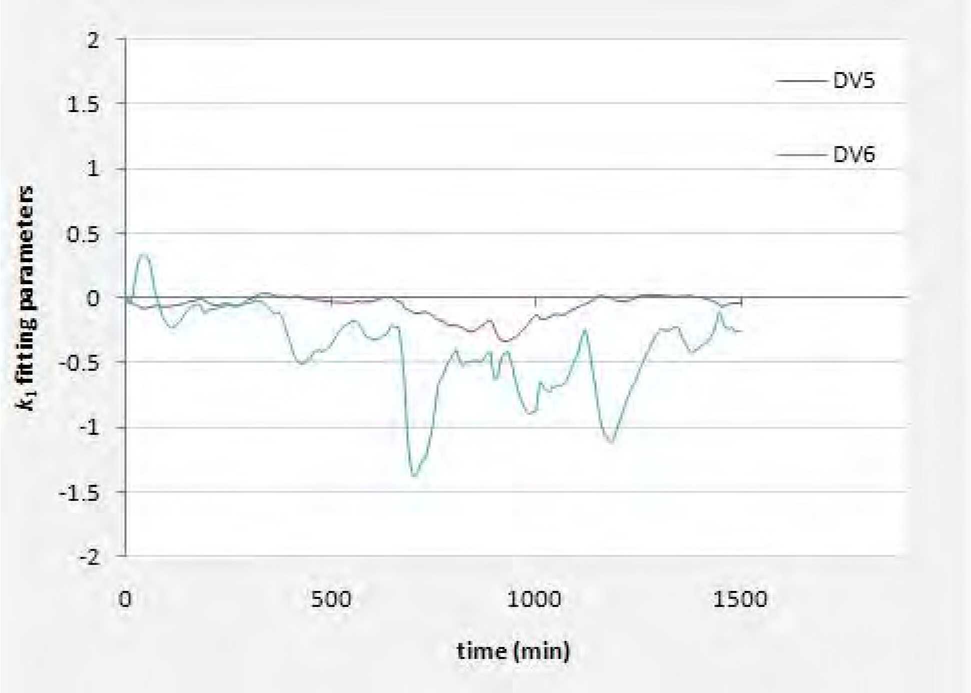


Figure 8.84: Kalman Filter plot for CV6 related to its MVs

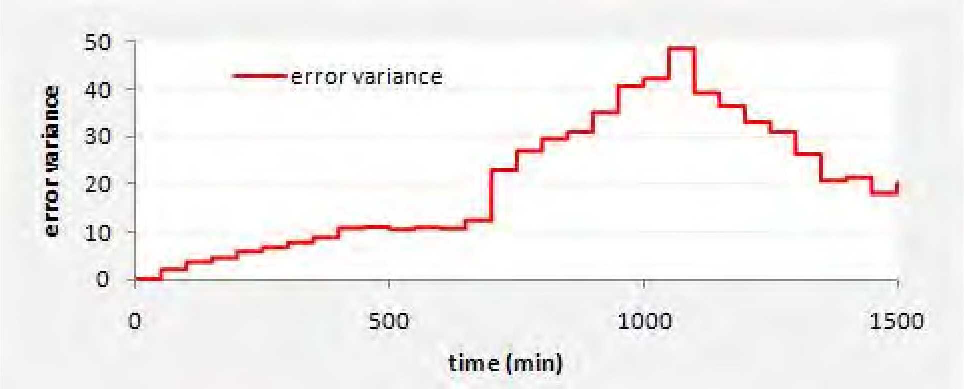
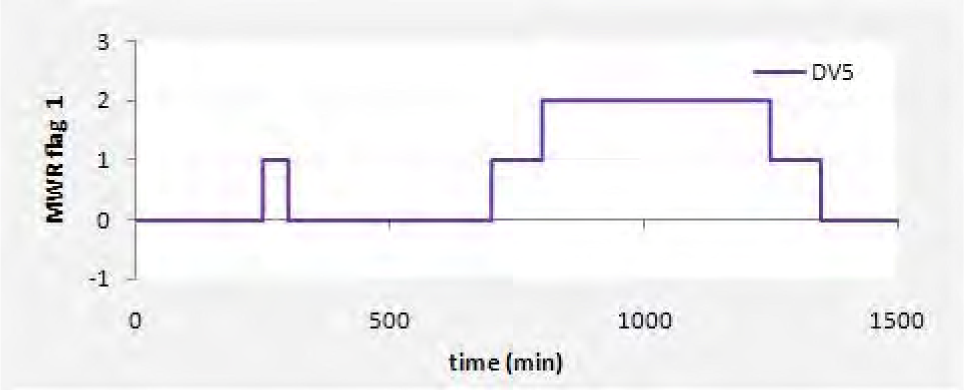


Figure 8.85: Error Variance for CV6 related to its DVs



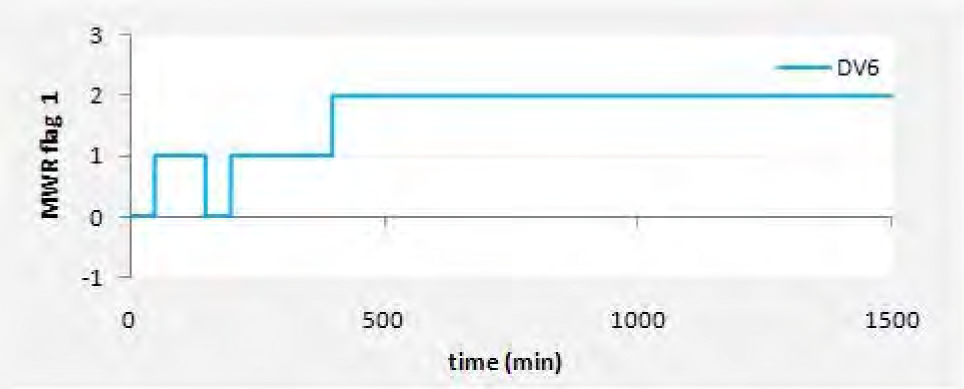


Figure 8.86: Warning flag for gain or time delay mismatch related to DVs

The correlation plot for DV5, figure 8.82(a), exhibits a stronger correlation in comparison to figure 8.82(b). However, the regression fitting parameters in figure 8.83 show that the parameter related to DV6-CV6 deviates further away from zero than the parameter obtained for DV5-CV6. It is suggested that the signal DV6 is influenced largely by unmeasured disturbances compared to DV5. Thus it can be deduced that the large error variance in figure 8.85 is primarily due to the influence of DV6 (and the unmeasured noise acting upon it). Figure 8.84 illustrates the response for the Kalman filter which does not produce reasonable results when the input data is not sufficiently rich. The fitting parameters obtained by the moving window regression are found to be outside the range of -1 to +1 prompting the warning flags to suggest that these channels are subject to time delay mismatches.

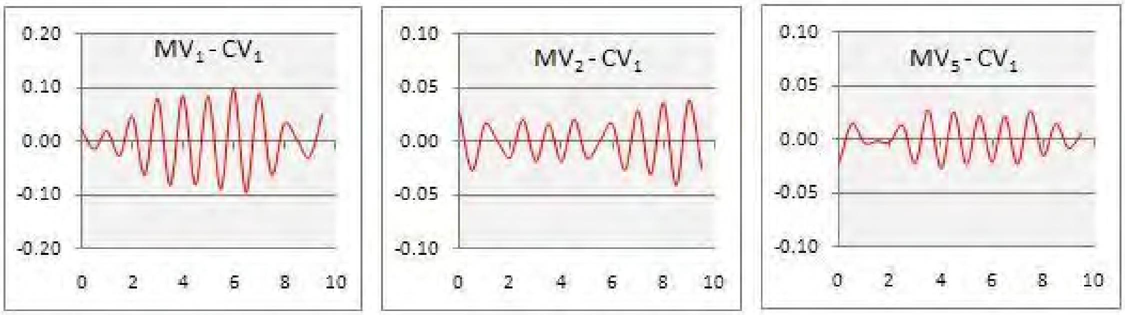
CV6, the petrol product is sent to the next phase for further processing. Mismatches in the models related to CV6 will imply that the next phase of operation will receive a too high flow rate or a too low flow rate which could result in production losses.

8.2.6. Level in DM-1 – CV1

One of the major control objectives of this controller is to maintain the level in this reflux drum. Maintaining the level in this drum implies that the petrol bypass flow and subsequent production loss is minimized. This level is influenced by MV1, MV2, MV5, DV1, DV2, DV3 and DV4. All of these models are integrator type models (see Appendix B, row 1 in figures B.1 through to B.3). These models produce very large prediction outputs (on a direct basis) making the use of differential inputs and outputs vital in the detection of model mismatches in these channels (see section 5.3.1). As a consequence, these models are divided into 3 model sets to avoid large outputs. Large model sets containing integrator terms in all models make it difficult to detect any model mismatch as the fitting parameters obtained by regression become extremely large in order to fit the extremely high error formed (refer Appendix C, figures C.14 and C.15).

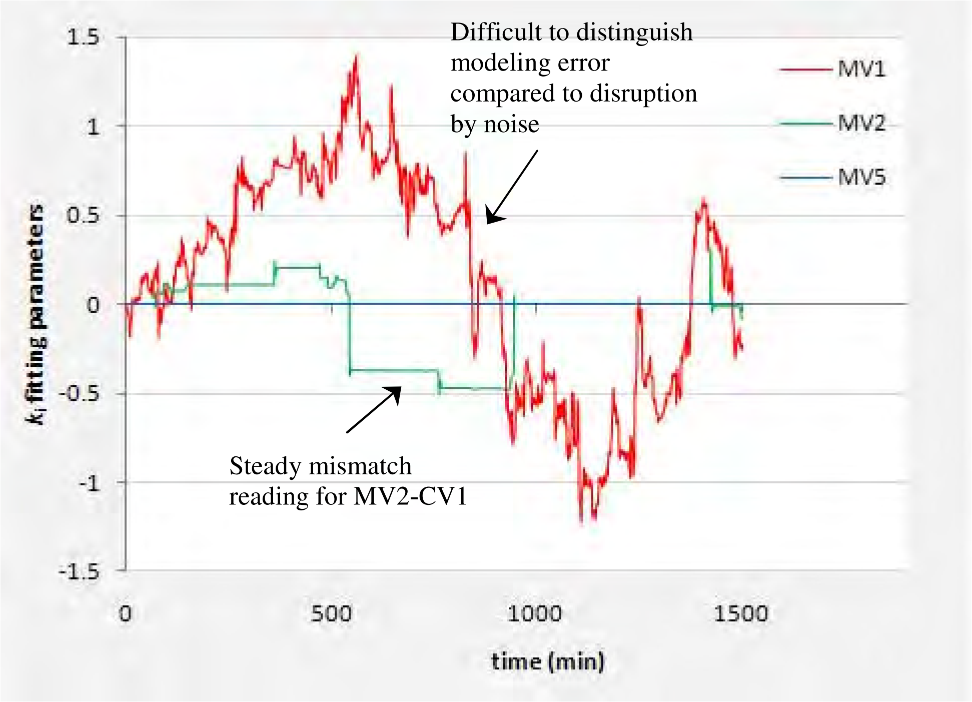
8.2.6.1. Effect of MVs on CV1

The first of the 3 models sets contains all the manipulated variables related to CV1. The results for these models are illustrated below:



(a) (b) (c)

Figure 8.87: Correlation Plots for CV1 related to its MVs



Difficult to distinguish modeling error compared to disruption by noise

Steady mismatch reading for MV2-CV1

Figure 8.88: Moving Window Regression plot for CV1 related to its MVs

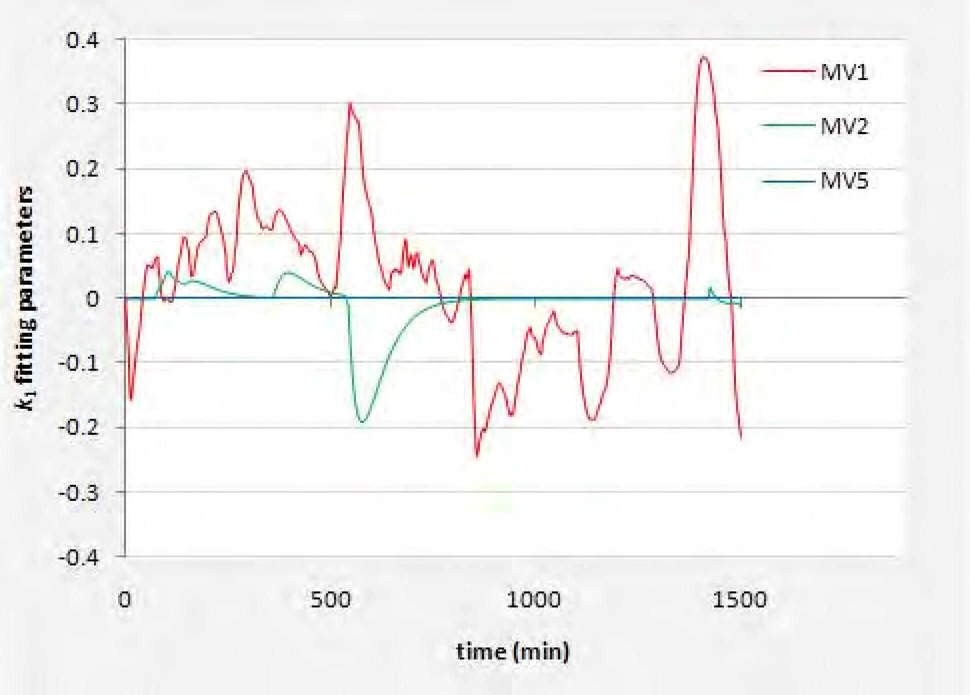


Figure 8.89: Kalman Filter plot for CV1 related to its DVs

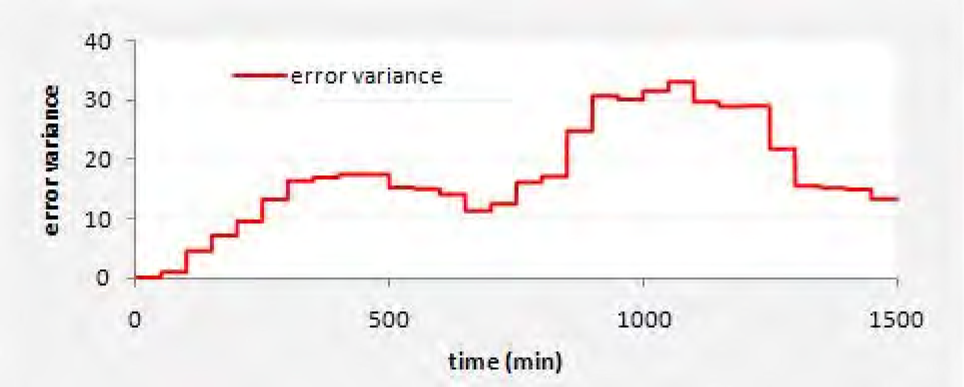


Figure 8.90: Error Variance for CV1 related to its MVs

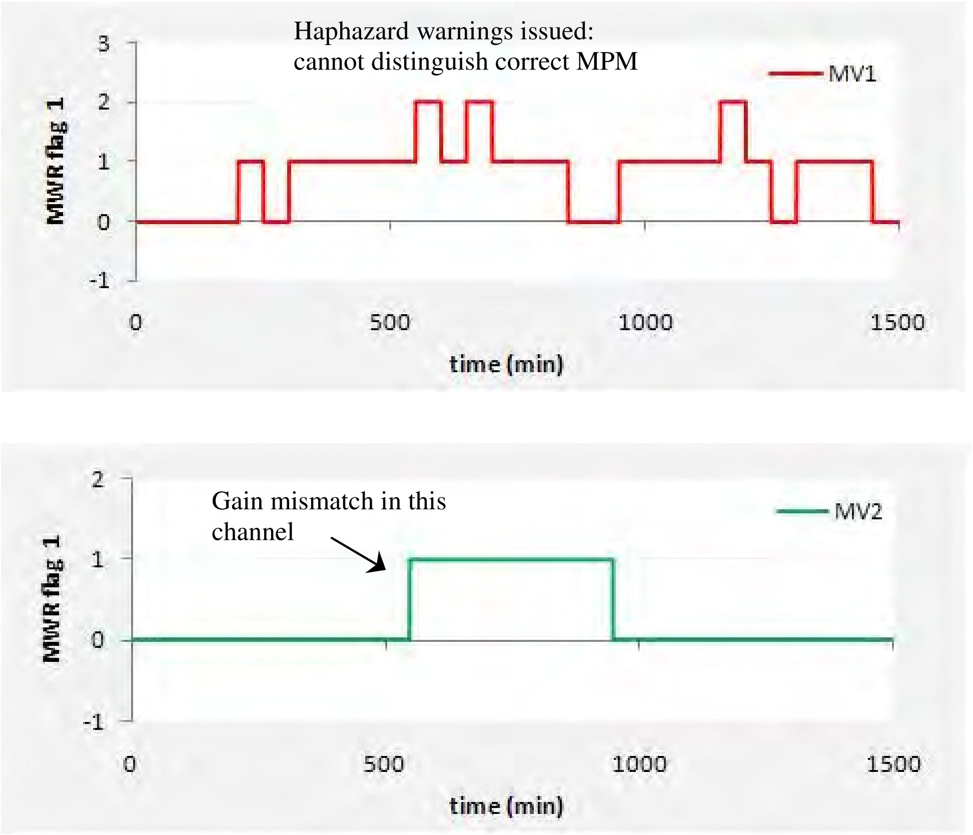


Figure 8.91: Warning flag for gain or time delay mismatch related to MVs

One may recall that in figure 8.3 it was deduced that MV1 and MV5 are correlated with each other. The effect of this correlation amongst these inputs can be seen in figure 8.87. The correlation plot in figure 8.87(c) has the same local variations and shapes as that obtained in figure 8.87(a) albeit with a smaller magnitude. These plots suggest there is a mismatches in channels MV1-CV1 and MV5-CV1. However, the Moving window regression plot and the Kalman Filter plot in figures 8.88 and 8.89 respectively suggest the there is no mismatch present channel MV5-CV1. The correlation plot revealed a false indication of a mismatch due to the correlation amongst the inputs. Low variance data results in smaller magnitudes of fitting parameters obtained by Kalman filter estimations (figure 8.89). Distinguishing the source of model error for MV1-CV1 was seen to be impossible as the influence of noise signals produced a fitting parameter that has a somewhat oscillatory behavior. The methodology fails to recognize true model mismatches in the presence of the large amounts of noise disruption. The large error variance in figure 8.90 suggests that the error is compounded by noise signals. The fitting parameter for channel MV2-CV1 produces a steady reading which lies within the range of -1 to +1, thus it is regarded as a gain mismatch (Figure 8.91). The multiplicative factor for this fitting parameter is shown below in figure 8.92:

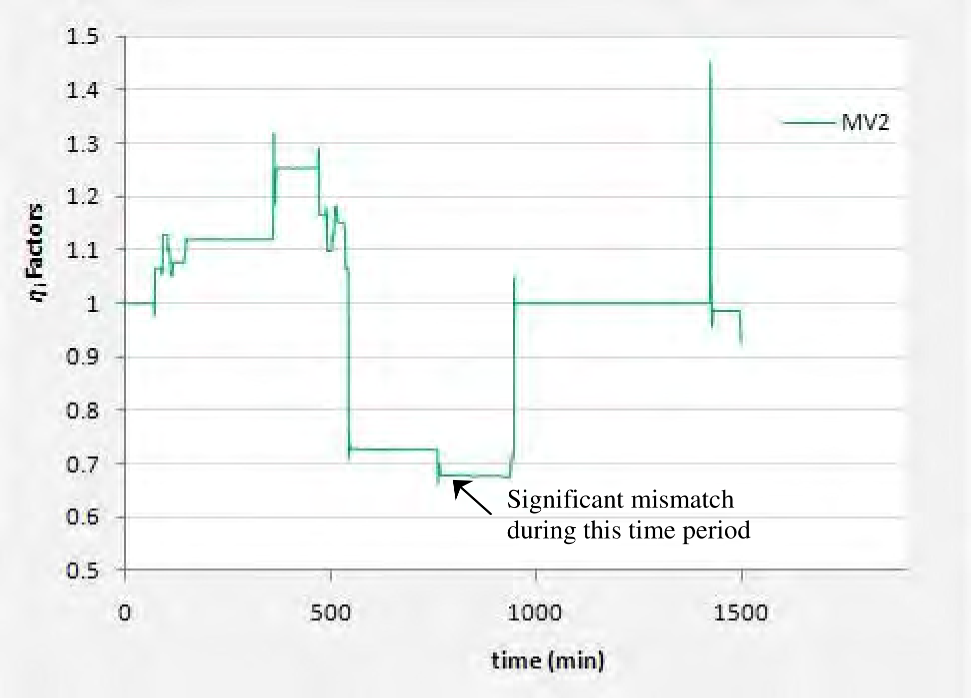


Figure 8.92: MWR factor plot for CV1 related to its MV2

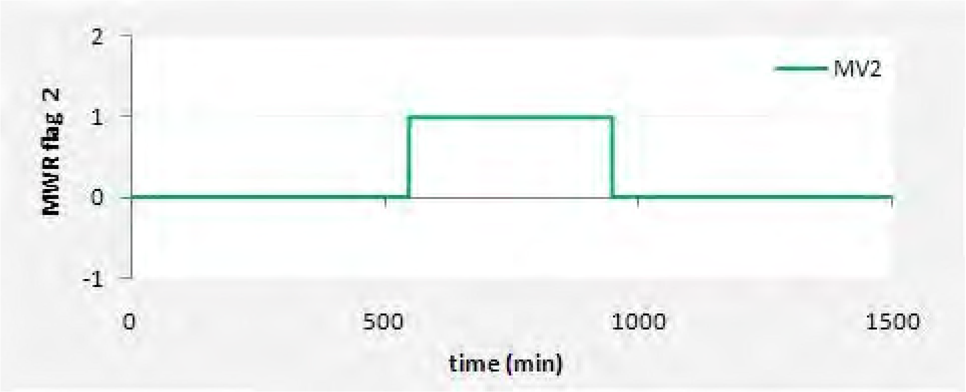


Figure 8.93: Warning flag issued for significant gain mismatch

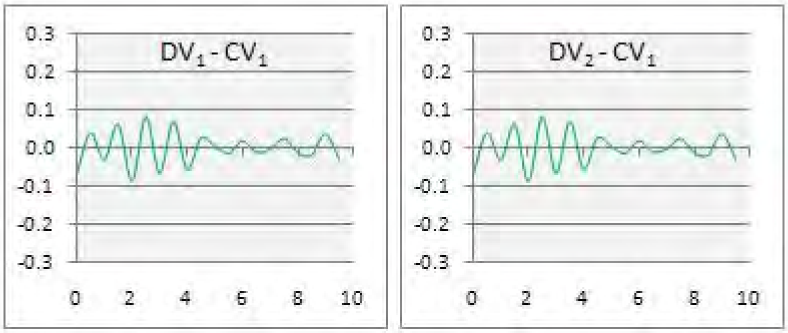
Table 8.9:
Overall Regression parameter and multiplicative factor for MV2-CV1

	k_2
MWR fitting parameter	0.4856
	η_2
MWR multiplying factor	0.6731

This gain mismatch was found to be significant in the time period between 500 and 1000 minutes. The fitting parameter obtained was ~ 0.5 and as a result the multiplying factor was found to be ~ 0.67.

8.2.6.2. Effect of DV1 and DV2 on CV1

The second model set related to CV1 contains the models related to DV1 and DV2. The corresponding results are given below:



(a) (b)

Figure 8.94: Correlation Plots for CV1 related to DV1 and DV2

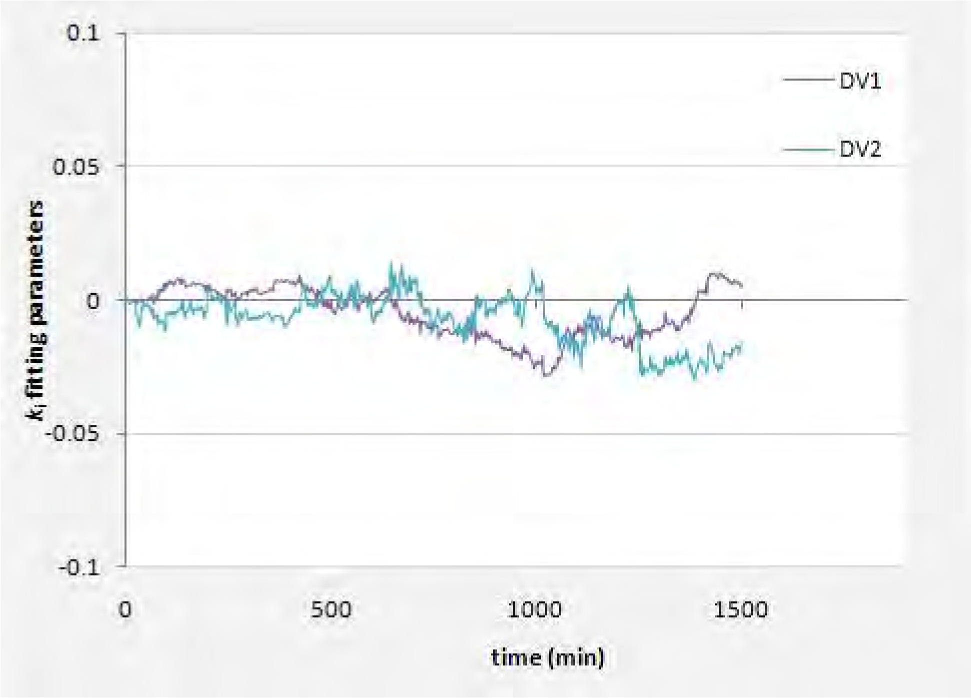


Figure 8.95: Moving Window Regression plot for CV1 related to DV1 and DV2

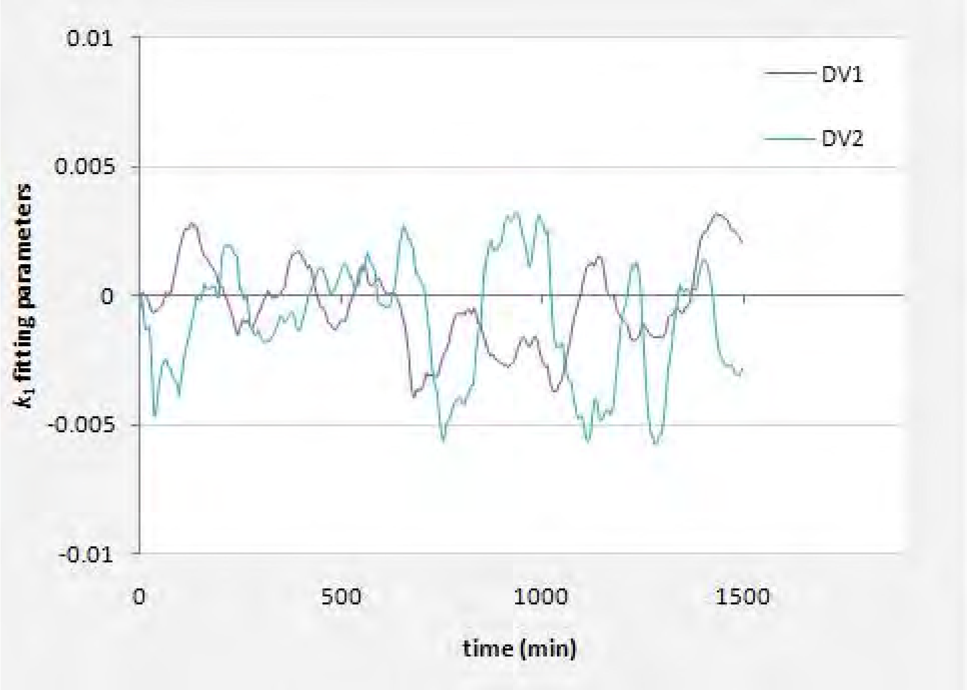


Figure 8.96: Kalman Filter plot for CV1 related to DV1 and DV2

Negligible deviations away from zero for the regression plots, in figure 8.95 and 8.96 suggest that these models represent the plant accurately. The small correlation deviations imply that mismatches may be present in the lower order polynomial coefficients.

8.2.6.3. Effect of DV3 and DV4 on CV1

The third model set is composed of channels DV3-CV1 and DV4-CV1. The results are as follows:

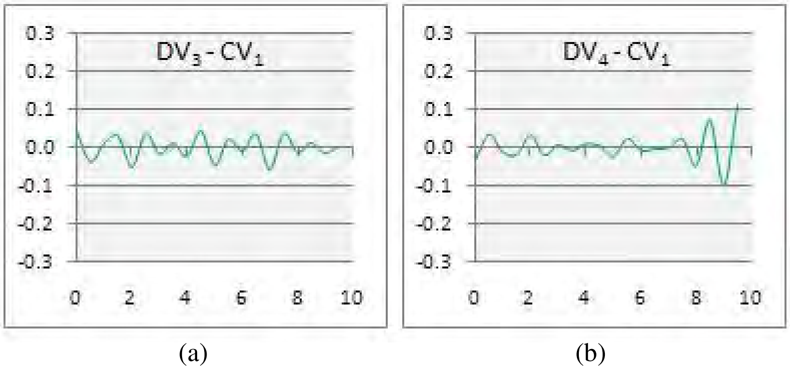


Figure 8.97: Correlation Plots for CV1 related to DV3 and DV4

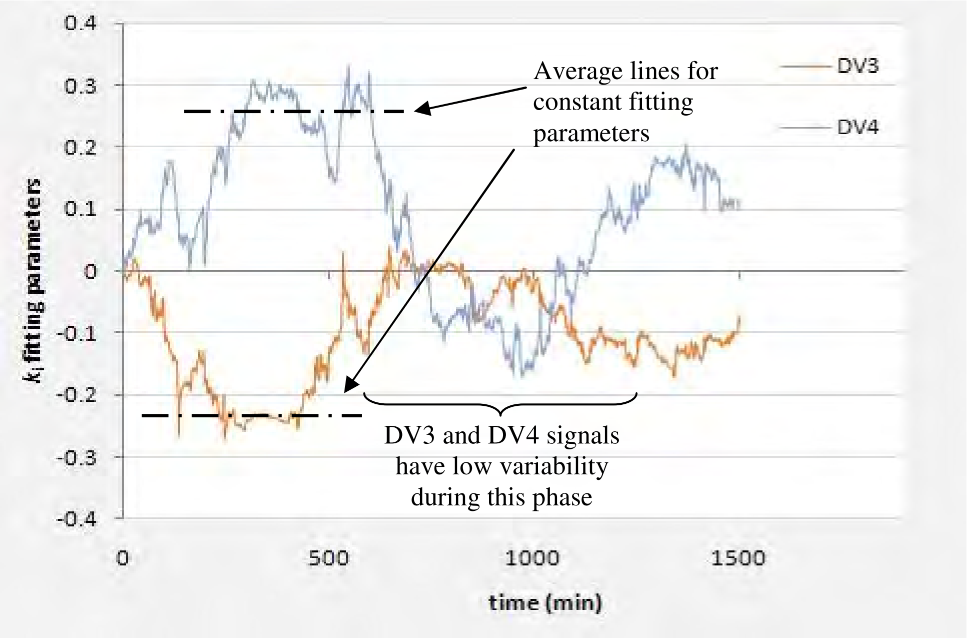


Figure 8.98: MWR plot for CV1 related to DV3 and DV4

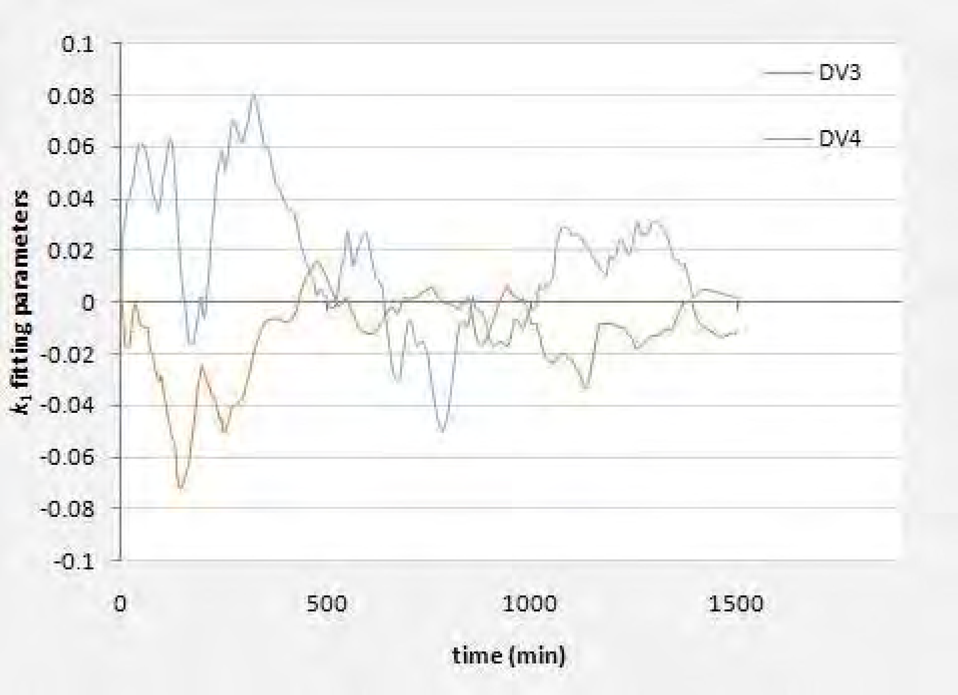


Figure 8.99: Kalman Filter plot for CV1 related to DV3 and DV4

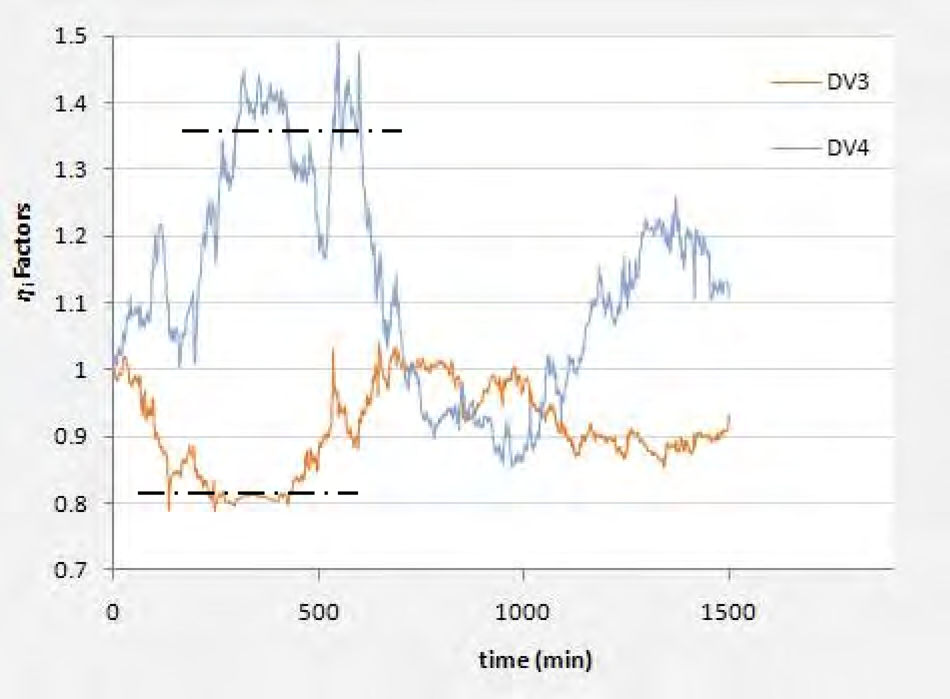


Figure 8.100: MWR factor plot for CV1 related to DV3 and DV4

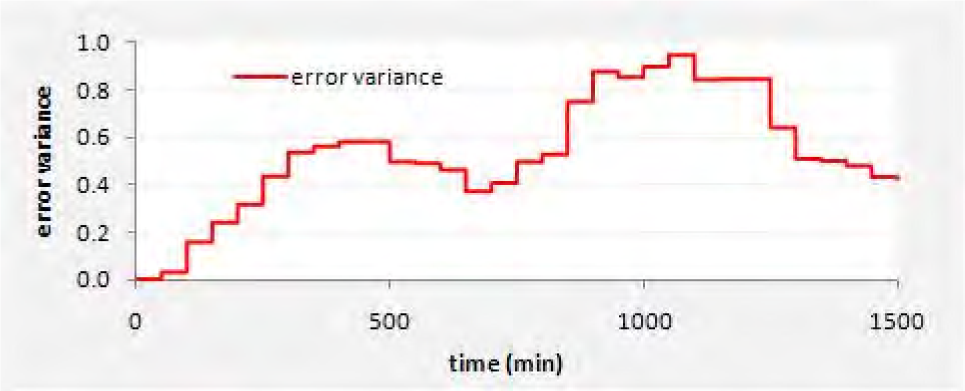


Figure 8.101: Error Variance for CV1 related to DV3 and DV4

Table 8.10:
Overall Regression parameters and multiplicative factors for DV$_i$-CV1

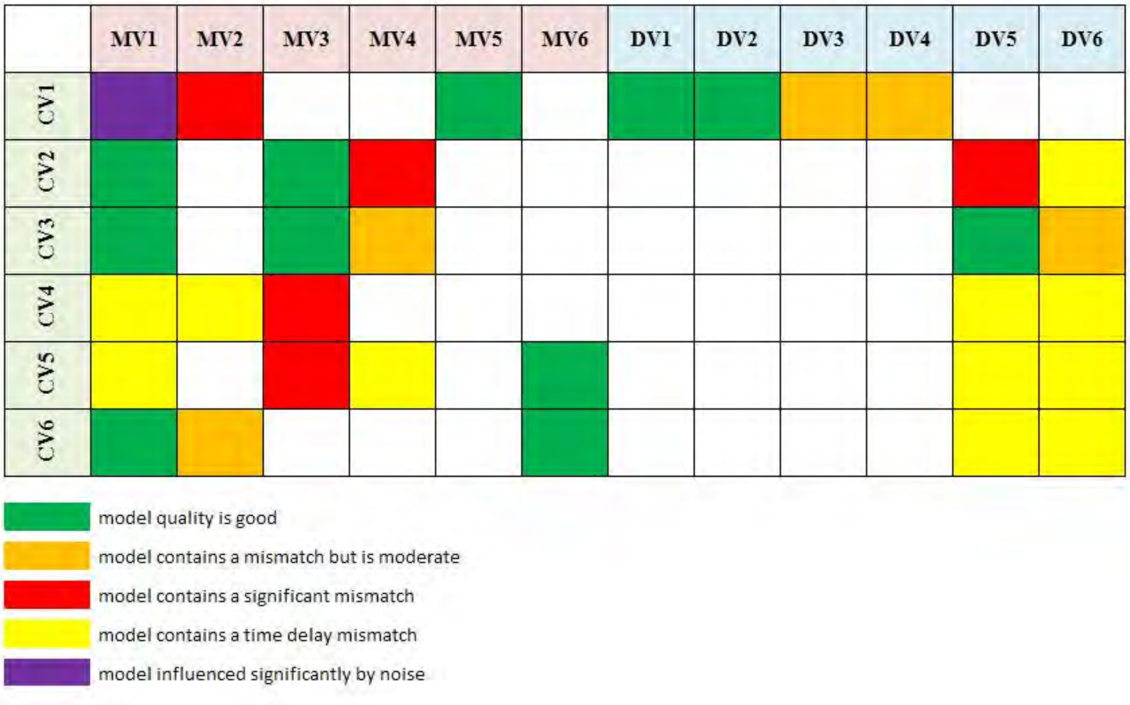
	k_3	k_4
MWR fitting parameter	-0.2251	0.2736
	η_3	η_4
MWR multiplying factor	0.8163	1.3767

In this instance, the each model in this set is shown to contain a gain mismatch. This gain mismatch is not regarded as significant.

The overall diagnosis for CV1 is as follows: channels MV5-CV1, DV1-CV1 and DV2-CV2 show no significant mismatch and thus resemble the plant accurately in that aspect. A diagnosis could not be provided for MV1-CV1 as the behavior of the fitting parameter obtained is disrupted by noise signals. Channels DV3-CV1 and DV4-CV1 were both detected for gain mismatches, which were subsequently found to be within the desired range.

8.3. DIAGNOSIS CHART

Once all the models in a model matrix are tested and a model quality indication is provided, the following chart is provided to the maintenance engineer that illustrates the specific input-output pairings that exhibit the different forms of mismatch:



	MV1	MV2	MV3	MV4	MV5	MV6	DV1	DV2	DV3	DV4	DV5	DV6
CV1	🟪	🟥			🟩		🟩	🟩	🟧	🟧		
CV2	🟩		🟩	🟥							🟥	🟨
CV3	🟩		🟩	🟧							🟩	🟧
CV4	🟨	🟨	🟥								🟨	🟨
CV5	🟨		🟥	🟨		🟩					🟨	🟨
CV6	🟩	🟧				🟩					🟨	🟨

🟩 model quality is good

🟧 model contains a mismatch but is moderate

🟥 model contains a significant mismatch

🟨 model contains a time delay mismatch

🟪 model influenced significantly by noise

Figure 8.102: Diagnosis chart for the Petrol debutanizer

It provides a simple yet informative tool that shows the precise location(s) that contain a mismatch. If the controller is performing badly, the maintenance engineer would know where the mismatch lies and subsequently which inputs to perturb in the re-identification process. Furthermore, when this chart is viewed as a whole, the user can determine if a certain input is causing a certain mismatch. For example, if all the models related to a specific input contain some degree of a mismatch then it can safely be assumed that there could be a fault with the equipment used to acquire that specific input. Such is the case for the disturbance variables DV5 and DV6. Conversely, if all of the models relating to a specific output are in error, the measurement of that output, presently or in the commissioning stage, could be in error.

CHAPTER 9 CONCLUSIONS

The conclusions gathered from the research covered, simulation studies and the industrial case study are presented. The general outcomes satisfying the required objectives are initially highlighted. A brief view of the advantages and shortfalls of the methodology is given by the explanation of the simulation results obtained in chapter 7. The efficacy of the methods is shown in the diagnosis of the results obtained for the industrial case study.

9.1. GENERAL REMARKS

The ever increasing competition between industrial companies is an incentive to push the limits of the performance of industrial plants in which large sums of money are invested. As a consequence, a number of industries have turned to Advanced Process Control systems such as the widely applied Model Predictive Controller. Model Predictive Control has been widely used throughout the petroleum, chemical, metallurgical and pulp and paper industry over the years. It is industrially attractive because it handles hard constraints, usually on input and output signals. MPC technology has progressed steadily since the first MPC applications 34 years ago. Despite the numerous strides made in these applications throughout the years, the ‘Achilles heel’ of these applications, however, still remains with the accuracy of the process model. Performance monitoring and diagnosis of any MPM are necessary to assure effectiveness of process control and consequently safe and profitable plant operation. The primary application of the methodology developed in this work is envisaged in the provision of a quality indication for the models employed for MPC systems under closed-loop conditions. The methods are shown to be capable of locating the precise position of MPM and subsequently defining the extent of the mismatch.

The following were successfully achieved as a means to satisfy the objectives of this work:

- Different model validation approaches, specifically those that are capable of being implemented in an online manner or making use of historical data, were surveyed.

- The methodology developed was based on the ability to provide a qualitative view of a model mismatch as well as present the extent of the mismatch quantitatively. For these purposes, methods based on correlation analyses and regression analyses were developed based on the error.

- The regression techniques included recursive regression by the moving window approach and the Kalman Filter adapted for parameter estimations.

- Validation simulations were performed on two simulation units and were designed to mirror industrial occurrences. This aided in establishing the effectiveness of the methodology in industrial applications.

- Simulation results revealed a number of positive attributes of the developed methodology under industrial situations as well its shortfalls.

- The methods developed were tested on an industrial application provided by SASOL, a petrol debutanizer, and were shown to provide reasonable results coupled with the reduction of model sets.

The following sections detail some of the key conclusions drawn considering the simulated results obtained as well as those obtained for the industrial application.

9.2. Proposed Methodology

The methodology developed involved several intricate concepts. These concepts included model form, discretization, optimal sampling intervals and suitable error formulations. The model for the predicted output was an ARX model type. The models presented by SASOL were in a continuous-time form (Laplace transforms). These models were required to be converted to a form that can be employed in real-time. A Tustin approximation for the Laplace operator was used for this purpose. In the case of simulation studies, the optimal sampling interval was chosen to be a fraction of the shortest time constant in the system. Several error forms were defined, each with their advantages and shortfalls. The proposed methods were designed to solve for gain mismatches. The regression methods were developed with the idea of reducing computation complexity making them suitable for online implementation. These methods were modified to handle certain industrial occurrences such as the absence of disturbances.

9.3. MATLAB ® Software

Although the methods developed were easily implemented in order to mirror an online environment, several built-in MATLAB ® functions were employed. Nevertheless, each method is capable of running efficiently due to the ease of computation of the methods developed apart from the application of the partial correlations method and the computation of the correlation amongst inputs. This method requires three regression stages. The correlation amongst N inputs requires the computation of N^2 correlation plots. The program is set in a manner such that each method may be skipped if it is not required.

9.4. Simulation Studies

The method provided accurate and expected results for simple scenarios such as a gain mismatch. It was also shown to detect mismatches for small changes as well as larger ones. In the case of a system exhibiting correlations amongst some inputs, the proposed methodology was shown to separate the interactive relationship between these variables and subsequently provided the expected result. Most model validation methods, if not all, have a need for persistent signal excitation. It is shown that if the excitation reduces, the input term (β) in equation (5.32) will tend to get small and the default k_{exp} values will be given. The methods in this work provided reasonable results to an extent in a situation were data remained relatively flat. Noise levels present the biggest threat to the efficacy of the methodology. Simulation results reveal that the methodology fails to distinguish underlying modeling errors as the noise signals begin to dominate the overall error that forms. Although the proposed methodology primarily focused on relative gain mismatches, simulations reveal the ability to detect changes in time delay, and other parameters present in a model. Simple warning mechanisms were developed to provide an indication of the type of parameter mismatch and subsequently determine its significance.

9.5. INDUSTRIAL CASE STUDY

Several pertinent observations were noted in this case. The model matrix provided by SASOL contained 33 models. The proposed methodology focused on each output independently. It was found that large model sets related to each output resulted in errors that are compounded by a combination of noise signals and MPM. The idea of reducing the model sets to smaller sets produced reasonable results. Coupled with the warning schemes as well as the analysis of the regression plots model diagnostics were provided for each model. The fitting parameters are shown to vary due to the nature of the input data and the non-linearity present in the system. Several models were found to be mismatched significantly and key consequences of mismatch related to each CV were documented. In some instances (for example in channel MV1-CV1), the methodology failed to provide a model mismatch diagnosis. In comparison to the simulation results obtained for levels of noise, this failure could be owed to large levels of noise disruption. A simple diagnosis chart was developed which serves as a quick means of providing mismatch model locations if a controller is found to be performing poorly.

CHAPTER 10 RECOMMENDATIONS

In this chapter recommendations are made with respect the model validation approaches based on the results and conclusions presented in this work. These include the ability to deal with noise sensitivity. Additionally suggestions into further studies are made in terms of developing the proposed method online.

10.1. RECOMMENDATIONS

Given the conclusions stated in the previous chapter, an understanding towards providing model quality indications is evident. The following points, however, deserve attention in related research projects in future.

10.1.1. Noise Reduction

The topic of model output sensitivity to noise is acknowledged as being reasonably important and worthy of further discussions. The impact of noise signals is emphasized in chapter 7. Although this thesis is limited to the detection of models that represent plant dynamics (hence the choice of an ARX model), it is suggested that during intrusive plant testing that noise models be identified and introduced into the model matrix (ARMAX model). In this instance, the effect of noise will be considered within the models and as a result it will inhibit the disruption of the proposed methodology.

Another means of handling the effect of noise is by implementing a data pre-filtering system which allows plant data to pass through and subsequently have the effect of noise reduced. Ljung (1987) emphasizes a filter termed the ‘anti-aliasing filter’, which removes that portion of a signal which occupies the high frequency band. Developing such a means of noise reduction will go a long way in ensuring the success of the proposed methodology in various types of plants. It would be desirable to determine the impact of introducing such a filtering system on maintaining the plant dynamics within the plant output.

10.1.2. Adaptive Control

A more in depth recommendation would be to investigate Adaptive Model Predictive Control, where system identification and model validation techniques are implemented online in efforts to continuously update the model being used. There is a strong market for self-tuning MPC controllers. Validating the different models being generated by a continuous stream of data and subsequently have the controller vary the model parameters is an interesting problem.

10.2. Impact of MPM on Controller Performance

Little work has been done in the area of quantifying the impact of MPM on controller performance. Even though the poor models in the controller are detected, a more pertinent question to ask is: what is the quantitative effect of MPM on MPC performance? Is it significant in which case one can apply the proposed technique? Otherwise one should look at other causes for poor MPC performance. A poor model may not necessarily lead to degradation in the controller performance. Hence it is highly desirable to isolate the role of MPM in poor control and quantify its impact.

10.3. Future Work

The future work will involve further investigation into data pre-filtering in order to reduce the noise. The next step would be to develop this tool online. This would imply that the MATLAB ® built-in functions would have to be removed and programmed generically. At this stage one expects the periods of best excitation to give the best estimates. They can be expected to vary though due to non-linearity. One may examine the possibility of investigating reduced Λ diagonal terms in equation (5.31) to see if the k_i's go more constant.

APPENDIX A SOFTWARE DESCRIPTION

All programs written and used in this work are provided in the attached CD in a folder entitled 'SOFTWARE'. This folder contains two subsequent folders entitled 'FUNCTION FILES' and 'MAIN ERROR DETECTOR PROGRAM'. The folder 'FUNCTION FILES' contains all the scripted files used to develop the preliminary concepts explained in section 6.2. The latter folder contains the main simulation program which is briefly described below. A copy of this thesis is also found on the accompanying disc together with a 'turnitin' report.

A.1. FUNCTION FILES

These files are essential in the running of the model validation tool developed in this work. These files include model selection, obtaining a suitable sampling interval, discretization, coefficient handling and window data handling.

A.1.1. Model selection

The function files cv_parameters_plant.m and cv_parameters_model.m store the coefficients related to the models in the Laplace domain. A specific value for 'cv' allows for the choice of models to be analyzed. Currently 'cv' values of 0.1, 0.2 and 0.3 provide the models related to each output of the Shell Heavy Oil Fractionator and values from 1 through to 6 correspond to the models for each output of the Petrol Debutanizer controller.

A.1.2. Sampling interval

Once the model set is chosen, the function file get_dt.m is used to find the suitable sampling interval based on the dominant time constant found in the model set. In the case of the models used for the Petrol Debutanizer, the sampling interval is found and compared to the set sampling time provided by SASOL. A warning is issued when the sampling interval obtained using this function file differs by a marginal limit set within the main program, deeming the sampling interval provided by SASOL as inadequate.

A.1.3. Discretization

The function file s_to_z.m takes the model coefficients from the model set and converts them into a form that can be used in real time. The discrete approximation is set by assigning a value to 'm', either 1, 2 or 3 for the desired choice. The discrete form coefficients are obtained by using the built-in MATLAB ® function *numden.* This function enables the coefficients to be found in numerical form. This script also makes use of the function *simple* to cancel off any common factors when the discrete approximation is substituted for the Laplace operator.

A.1.4. Coefficient handling

These function files handle the discrete coefficients by displaying them into a form as depicted by figure 6.9. The c_coeff.m file produces the coefficients related to polynomial for the lagged output. It is obtained by using the built-in function *conv* which takes the product of all model lag polynomials and subsequently arranges the resulting coefficients from the highest order to the lowest. The d_coeff.m file produces the coefficients related to the lag polynomials for the inputs. In this case, division of polynomials is required for which the *deconv* function is used.

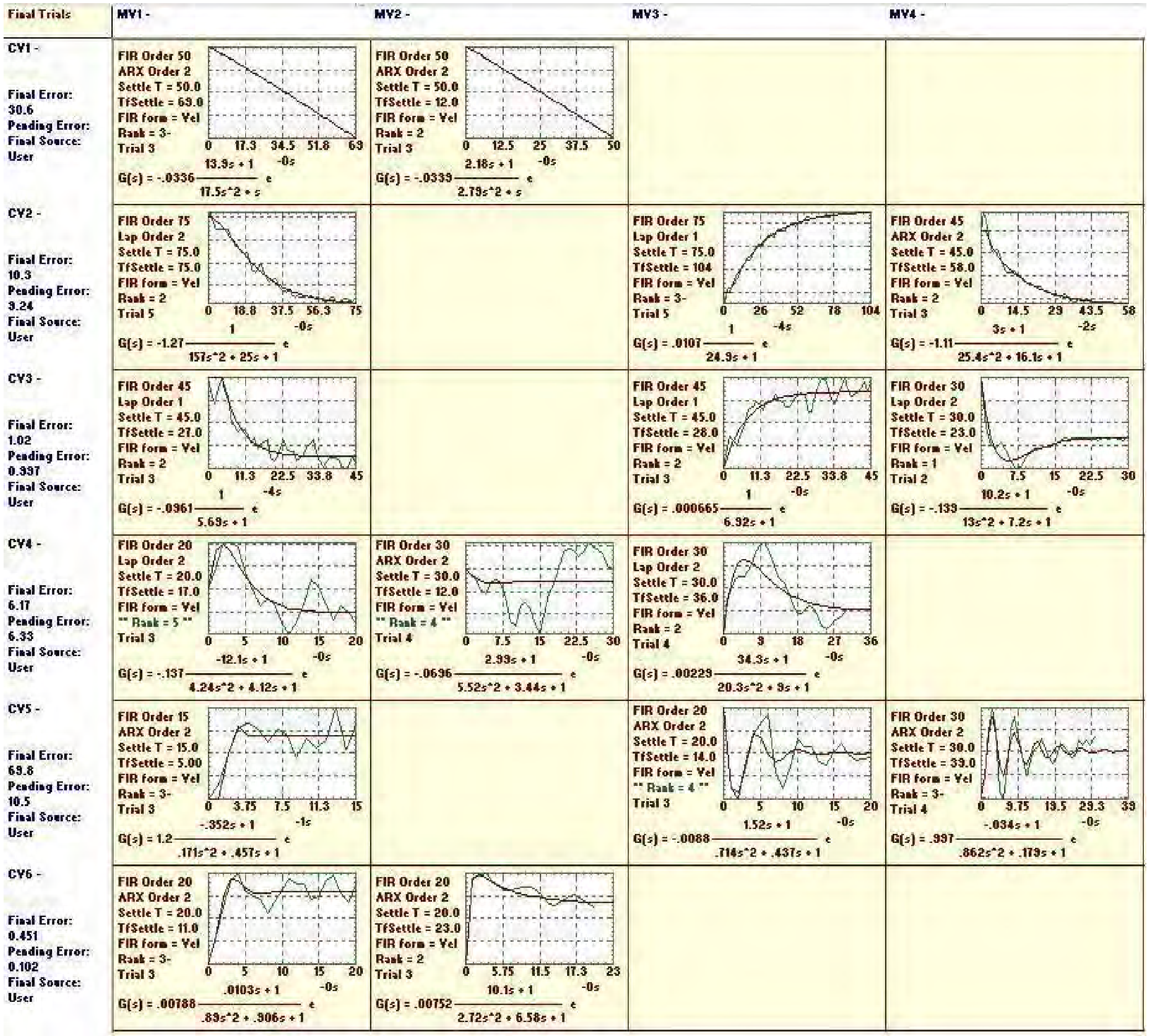
A.1.5. Window data handling

The moving window concept is maintained by the implementation of these two function files; ewg.m and uwg.m. The ewg.m file allows for each error value computed at each time step to be placed in its correct position in the moving window. Similarly, the uwg.m file positions each input value in relation to the time it entered.

A.2. MAIN ERROR DETECTION PROGRAM

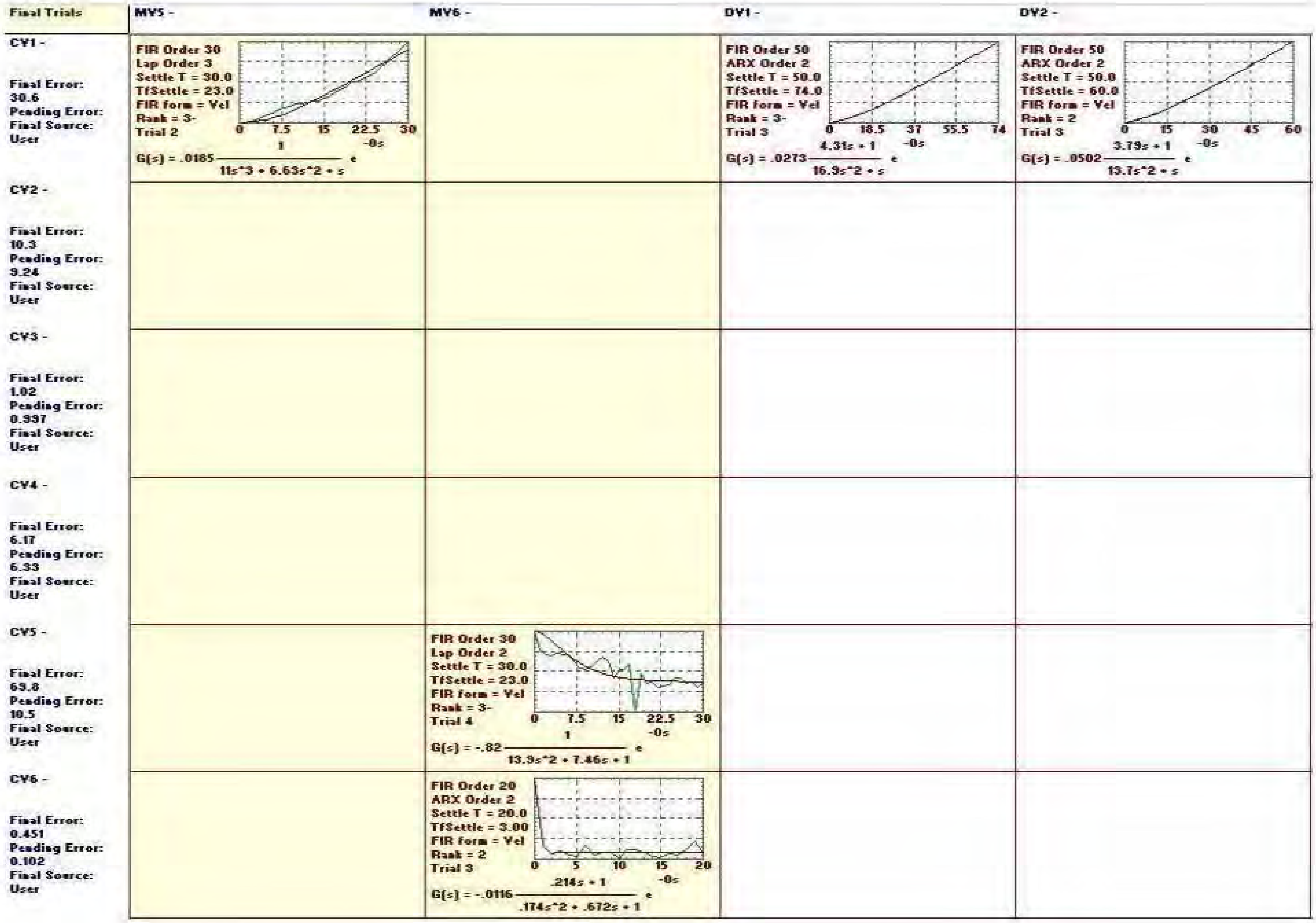
The methods developed in this work are all implemented within this program under a single time loop in order to envisage an online environment. Prior to the execution of the main time loop, a number of decisions which are set by the user, are required to be made. Among these decisions are the choice of error definition as well as which method to skip or run. A binary logic value of 1 is assigned to cases which are required to be implemented and a value of 0 is assigned in cases where certain methods are negated. Each method operates in an online manner, computing results at each time step. For the correlation analysis, correlation plots are obtained at set intervals within the designated time frame, defined by the user. On the other hand, the regression fitting parameters are computed at each time step and are stored for plotting and analysis.

APPENDIX B
DEBUTANIZER MODELS AND PID



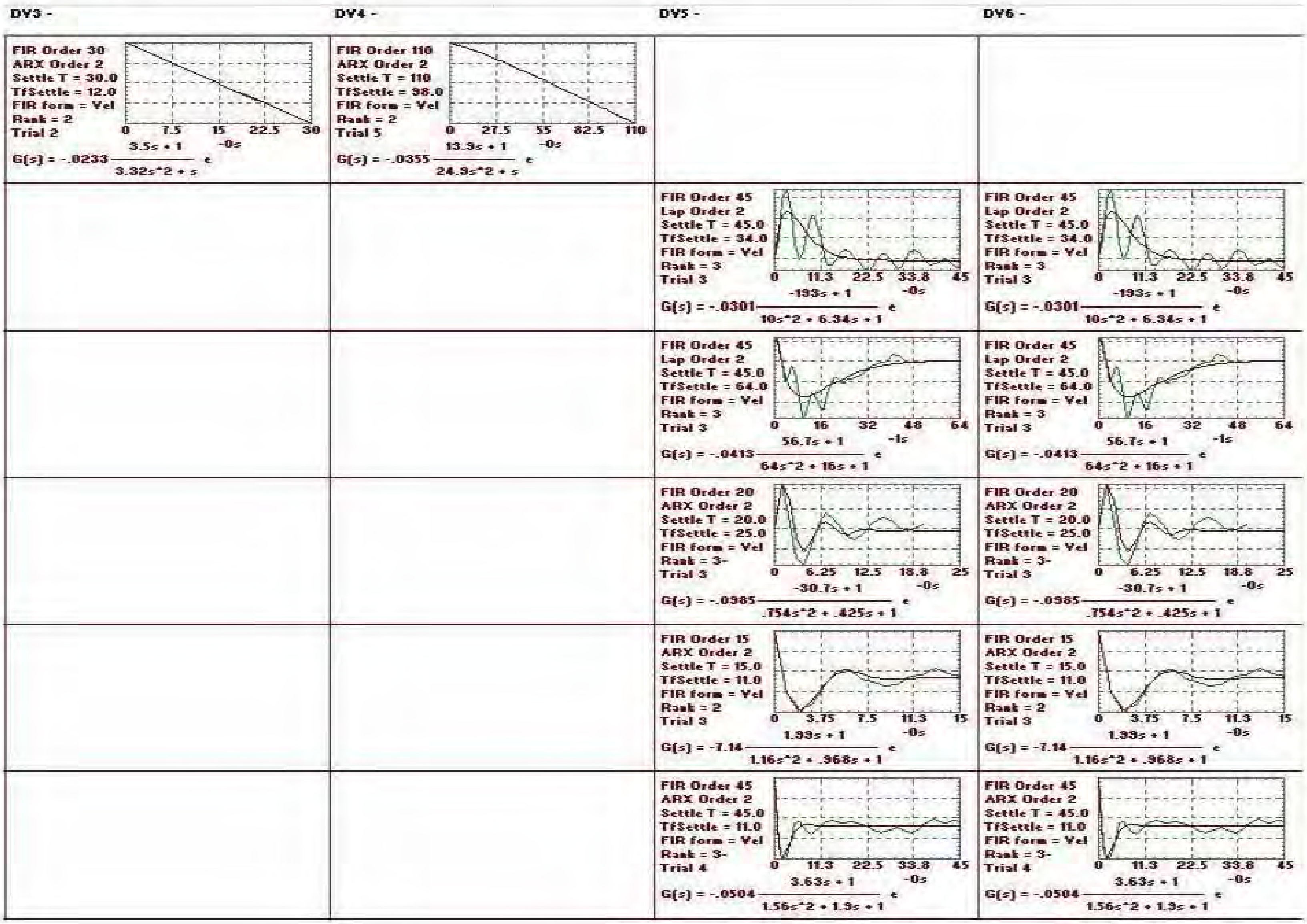
Final Trials	MV1 -	MV2 -	MV3 -	MV4 -
CV1 - Final Error: 30.6 Pending Error: Final Source: User	FIR Order 50 ARX Order 2 Settle T = 50.0 TfSettle = 69.0 FIR form = Vel Rank = 3- Trial 3 $G(s) = -.0336 \frac{13.9s + 1}{17.5s^2 + s} e^{-0s}$	FIR Order 50 ARX Order 2 Settle T = 50.0 TfSettle = 12.0 FIR form = Vel Rank = 2 Trial 3 $G(s) = -.0339 \frac{2.18s + 1}{2.79s^2 + s} e^{-0s}$		
CV2 - Final Error: 10.3 Pending Error: 9.24 Final Source: User	FIR Order 75 Lap Order 2 Settle T = 75.0 TfSettle = 75.0 FIR form = Vel Rank = 2 Trial 5 $G(s) = -1.27 \frac{1}{157s^2 + 25s + 1} e^{-0s}$		FIR Order 75 Lap Order 1 Settle T = 75.0 TfSettle = 104 FIR form = Vel Rank = 3- Trial 5 $G(s) = .0107 \frac{1}{24.9s + 1} e^{-4s}$	FIR Order 45 ARX Order 2 Settle T = 45.0 TfSettle = 58.0 FIR form = Vel Rank = 2 Trial 3 $G(s) = -1.11 \frac{3s + 1}{25.4s^2 + 16.1s + 1} e^{-2s}$
CV3 - Final Error: 1.02 Pending Error: 0.997 Final Source: User	FIR Order 45 Lap Order 1 Settle T = 45.0 TfSettle = 27.0 FIR form = Vel Rank = 2 Trial 3 $G(s) = -.0961 \frac{1}{5.69s + 1} e^{-4s}$		FIR Order 45 Lap Order 1 Settle T = 45.0 TfSettle = 28.0 FIR form = Vel Rank = 2 Trial 3 $G(s) = .000665 \frac{1}{6.92s + 1} e^{-0s}$	FIR Order 30 Lap Order 2 Settle T = 30.0 TfSettle = 23.0 FIR form = Vel Rank = 1 Trial 2 $G(s) = -.139 \frac{10.2s + 1}{13s^2 + 7.2s + 1} e^{-0s}$
CV4 - Final Error: 6.17 Pending Error: 6.33 Final Source: User	FIR Order 20 Lap Order 2 Settle T = 20.0 TfSettle = 17.0 FIR form = Vel ** Rank = 5 ** Trial 3 $G(s) = -.137 \frac{-12.1s + 1}{4.24s^2 + 4.12s + 1} e^{-0s}$	FIR Order 30 ARX Order 2 Settle T = 30.0 TfSettle = 12.0 FIR form = Vel ** Rank = 4 ** Trial 4 $G(s) = -.0696 \frac{2.99s + 1}{5.52s^2 + 3.44s + 1} e^{-0s}$	FIR Order 30 Lap Order 2 Settle T = 30.0 TfSettle = 36.0 FIR form = Vel Rank = 2 Trial 4 $G(s) = .00229 \frac{34.3s + 1}{20.3s^2 + 9s + 1} e^{-0s}$	
CV5 - Final Error: 69.8 Pending Error: 10.5 Final Source: User	FIR Order 15 ARX Order 2 Settle T = 15.0 TfSettle = 5.00 FIR form = Vel Rank = 3- Trial 3 $G(s) = 1.2 \frac{-.352s + 1}{.171s^2 + .457s + 1} e^{-1s}$		FIR Order 20 ARX Order 2 Settle T = 20.0 TfSettle = 14.0 FIR form = Vel ** Rank = 4 ** Trial 3 $G(s) = -.0088 \frac{1.52s + 1}{.714s^2 + .437s + 1} e^{-0s}$	FIR Order 30 ARX Order 2 Settle T = 30.0 TfSettle = 39.0 FIR form = Vel Rank = 3- Trial 4 $G(s) = .997 \frac{-.034s + 1}{.862s^2 + .179s + 1} e^{-0s}$
CV6 - Final Error: 0.451 Pending Error: 0.102 Final Source: User	FIR Order 20 ARX Order 2 Settle T = 20.0 TfSettle = 11.0 FIR form = Vel Rank = 3- Trial 3 $G(s) = .00788 \frac{.0103s + 1}{.89s^2 + .906s + 1} e^{-0s}$	FIR Order 20 ARX Order 2 Settle T = 20.0 TfSettle = 23.0 FIR form = Vel Rank = 2 Trial 3 $G(s) = .00752 \frac{10.1s + 1}{2.72s^2 + 6.58s + 1} e^{-0s}$		

Figure B.1: Debutanizer model matrix (inputs MV1 – MV4)



Final Trials	MV5 -	MV6 -	DV1 -	DV2 -
CV1 - Final Error: 30.6 Pending Error: Final Source: User	FIR Order 30 Lap Order 3 Settle T = 30.0 TfSettle = 23.0 FIR form = Vel Rank = 3- Trial 2 $G(s) = .0185 \frac{1}{11s^3 + 6.63s^2 + s} e^{-0s}$		FIR Order 50 ARX Order 2 Settle T = 50.0 TfSettle = 74.0 FIR form = Vel Rank = 3- Trial 3 $G(s) = .0273 \frac{4.31s + 1}{16.9s^2 + s} e^{-0s}$	FIR Order 50 ARX Order 2 Settle T = 50.0 TfSettle = 60.0 FIR form = Vel Rank = 2 Trial 3 $G(s) = .0502 \frac{3.79s + 1}{13.7s^2 + s} e^{-0s}$
CV2 - Final Error: 10.3 Pending Error: 9.24 Final Source: User				
CV3 - Final Error: 1.02 Pending Error: 0.997 Final Source: User				
CV4 - Final Error: 6.17 Pending Error: 6.33 Final Source: User				
CV5 - Final Error: 65.8 Pending Error: 10.5 Final Source: User		FIR Order 30 Lap Order 2 Settle T = 30.0 TfSettle = 23.0 FIR form = Vel Rank = 3- Trial 4 $G(s) = -.82 \frac{1}{13.9s^2 + 7.46s + 1} e^{-0s}$		
CV6 - Final Error: 0.451 Pending Error: 0.102 Final Source: User		FIR Order 20 ARX Order 2 Settle T = 20.0 TfSettle = 3.00 FIR form = Vel Rank = 2 Trial 3 $G(s) = -.0116 \frac{.214s + 1}{.174s^2 + .672s + 1} e^{-0s}$		

Figure B.2: Debutanizer model matrix (inputs MV5 – MV6, DV1 – DV2)



DV3 -	DV4 -	DV5 -	DV6 -
FIR Order 30 ARX Order 2 Settle T = 30.0 TfSettle = 12.0 FIR form = Vel Rank = 2 Trial 2 $G(s) = -.0233 \frac{3.5s + 1}{3.32s^2 + s} e^{-0s}$	FIR Order 110 ARX Order 2 Settle T = 110 TfSettle = 98.0 FIR form = Vel Rank = 2 Trial 5 $G(s) = -.0355 \frac{13.9s + 1}{24.9s^2 + s} e^{-0s}$		
		FIR Order 45 Lap Order 2 Settle T = 45.0 TfSettle = 34.0 FIR form = Vel Rank = 3 Trial 3 $G(s) = -.0301 \frac{-193s + 1}{10s^2 + 6.34s + 1} e^{-0s}$	FIR Order 45 Lap Order 2 Settle T = 45.0 TfSettle = 34.0 FIR form = Vel Rank = 3 Trial 3 $G(s) = -.0301 \frac{-193s + 1}{10s^2 + 6.34s + 1} e^{-0s}$
		FIR Order 45 Lap Order 2 Settle T = 45.0 TfSettle = 64.0 FIR form = Vel Rank = 3 Trial 3 $G(s) = -.0413 \frac{56.7s + 1}{64s^2 + 16s + 1} e^{-1s}$	FIR Order 45 Lap Order 2 Settle T = 45.0 TfSettle = 64.0 FIR form = Vel Rank = 3 Trial 3 $G(s) = -.0413 \frac{56.7s + 1}{64s^2 + 16s + 1} e^{-1s}$
		FIR Order 20 ARX Order 2 Settle T = 20.0 TfSettle = 25.0 FIR form = Vel Rank = 3- Trial 3 $G(s) = -.0985 \frac{-30.7s + 1}{.754s^2 + .425s + 1} e^{-0s}$	FIR Order 20 ARX Order 2 Settle T = 20.0 TfSettle = 25.0 FIR form = Vel Rank = 3- Trial 3 $G(s) = -.0985 \frac{-30.7s + 1}{.754s^2 + .425s + 1} e^{-0s}$
		FIR Order 15 ARX Order 2 Settle T = 15.0 TfSettle = 11.0 FIR form = Vel Rank = 2 Trial 3 $G(s) = -7.14 \frac{1.99s + 1}{1.16s^2 + .968s + 1} e^{-0s}$	FIR Order 15 ARX Order 2 Settle T = 15.0 TfSettle = 11.0 FIR form = Vel Rank = 2 Trial 3 $G(s) = -7.14 \frac{1.99s + 1}{1.16s^2 + .968s + 1} e^{-0s}$
		FIR Order 45 ARX Order 2 Settle T = 45.0 TfSettle = 11.0 FIR form = Vel Rank = 3- Trial 4 $G(s) = -.0504 \frac{3.63s + 1}{1.56s^2 + 1.9s + 1} e^{-0s}$	FIR Order 45 ARX Order 2 Settle T = 45.0 TfSettle = 11.0 FIR form = Vel Rank = 3- Trial 4 $G(s) = -.0504 \frac{3.63s + 1}{1.56s^2 + 1.9s + 1} e^{-0s}$

Figure B.3: Debutanizer model matrix (inputs DV3 – DV6)

APPENDIX C
FULL INDUSTRIAL RESULTS

The full results pertaining to the studies performed in chapter 8 are shown here. Chapter 8 demonstrates the concept of reducing the model set in order to obtain suitable results. This Appendix illustrates the effect of including large model sets for analysis. Common features of performing the designed methodology on large model sets are the small magnitude of the correlation coefficients as well as the excessively large regression fitting parameters. These results seek to further establish the benefit of handling smaller model sets.

C.1. TOP COLUMN TEMPERATURE (CV3)

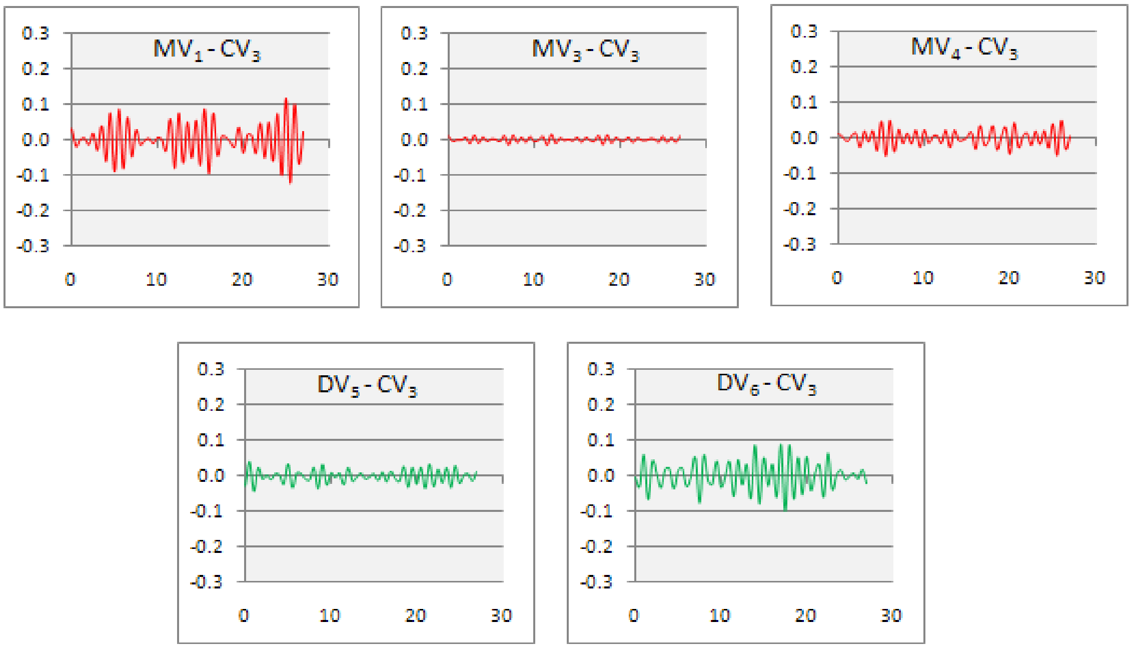


Figure C.1: correlation plots for CV3

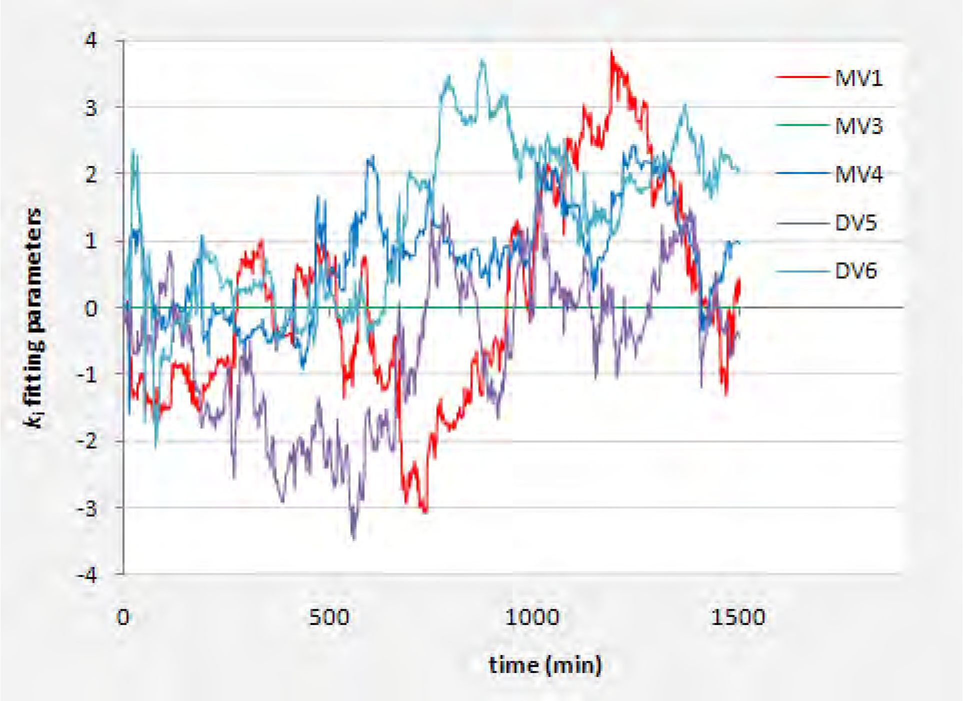


Figure C.2: Moving window regression plot for CV3

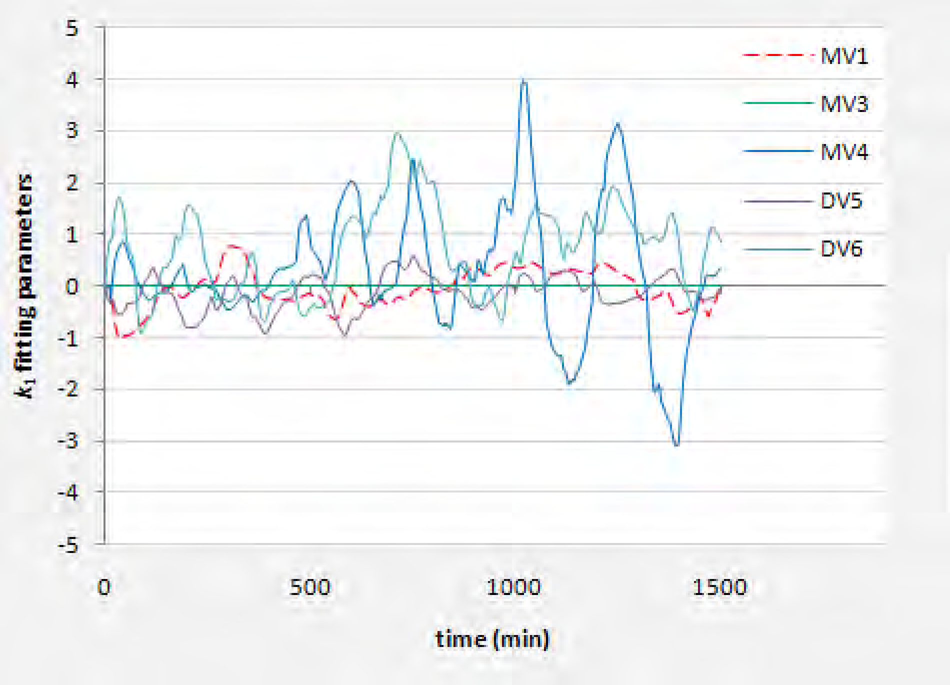


Figure C.3: Kalman Filter plot for CV3

C.2. Column Differential Pressure (CV4)

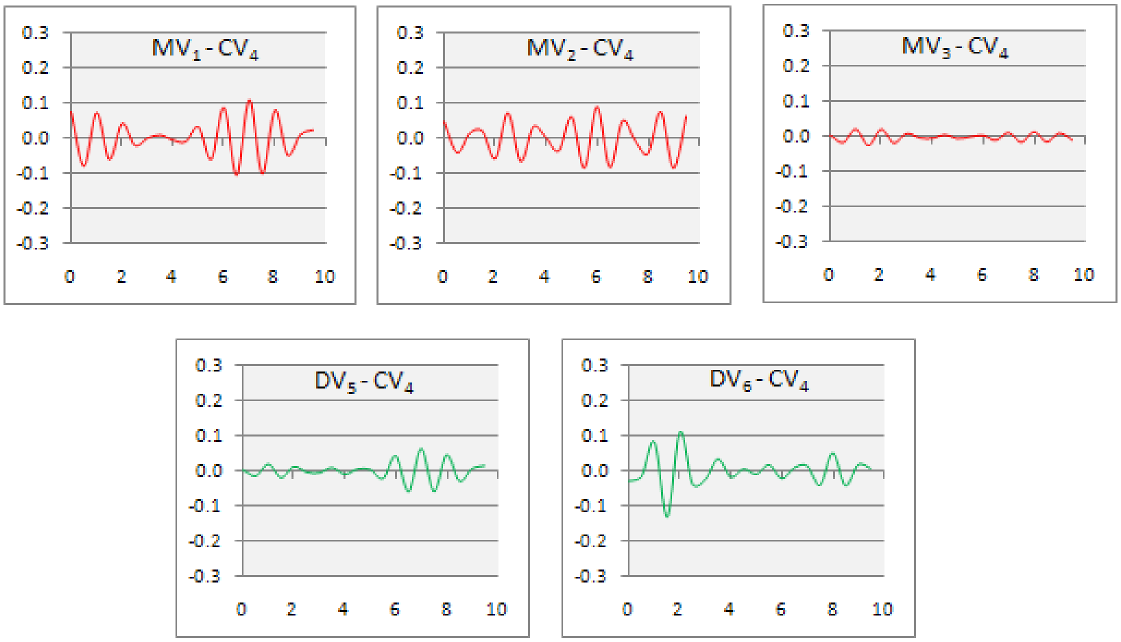


Figure C.4: Correlation plots for CV4

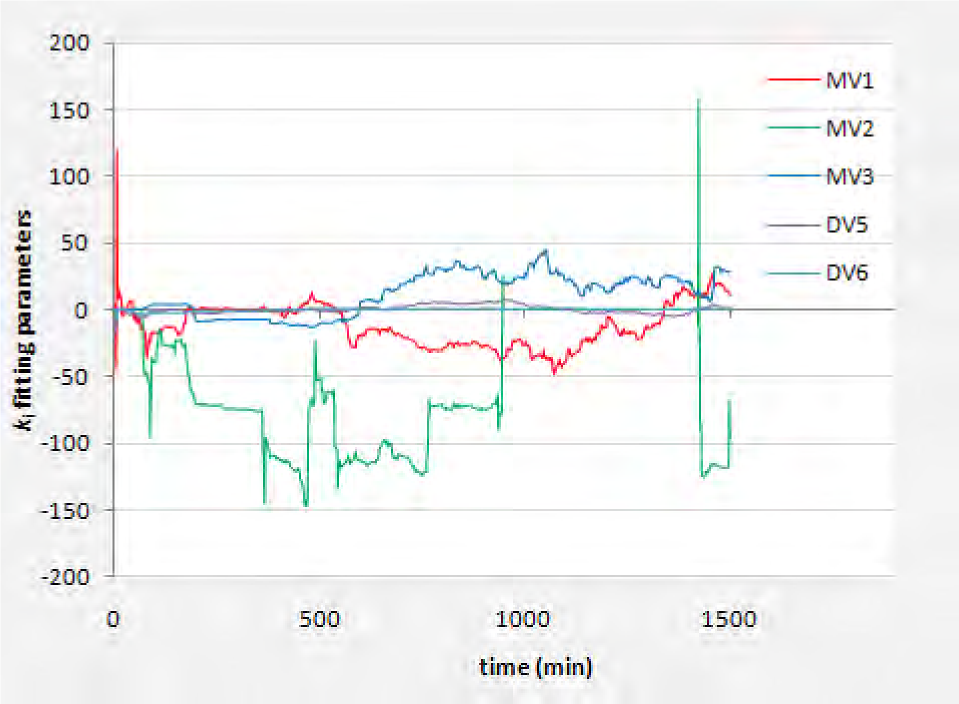


Figure C.5: Moving window regression plot for CV4

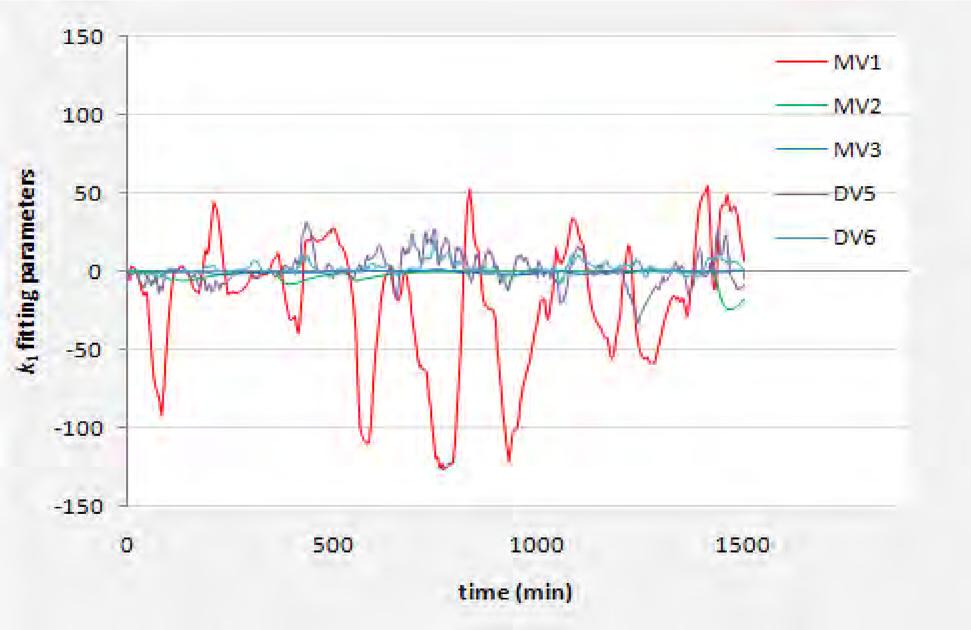


Figure C.6: Kalman Filter plot for CV4

C.3. Bottoms Level (CV5)

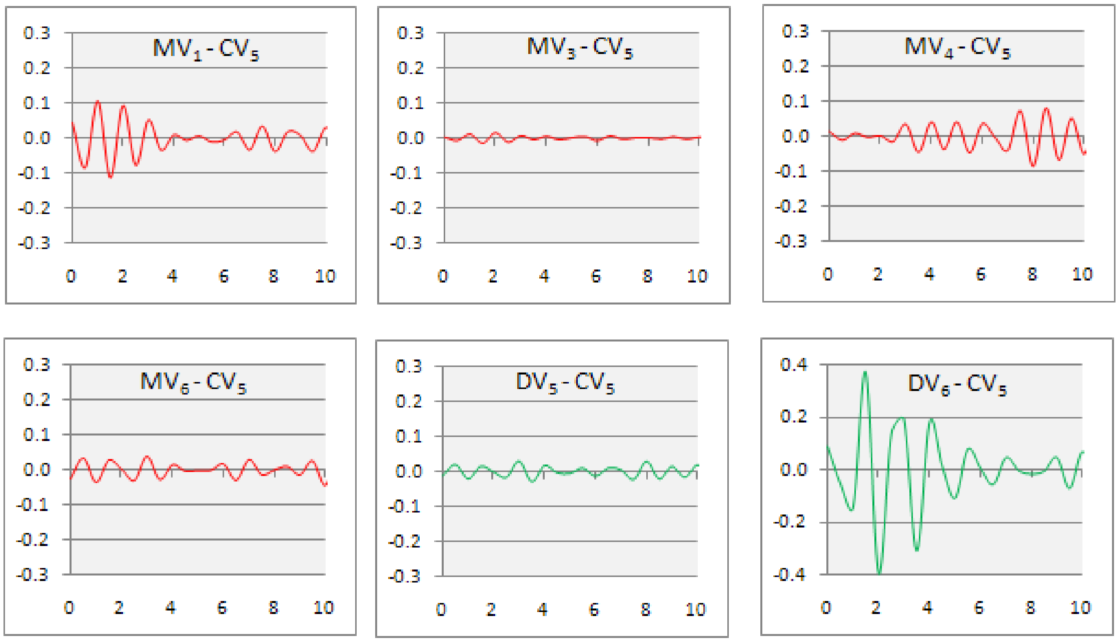


Figure C.7: Correlation plots for CV5

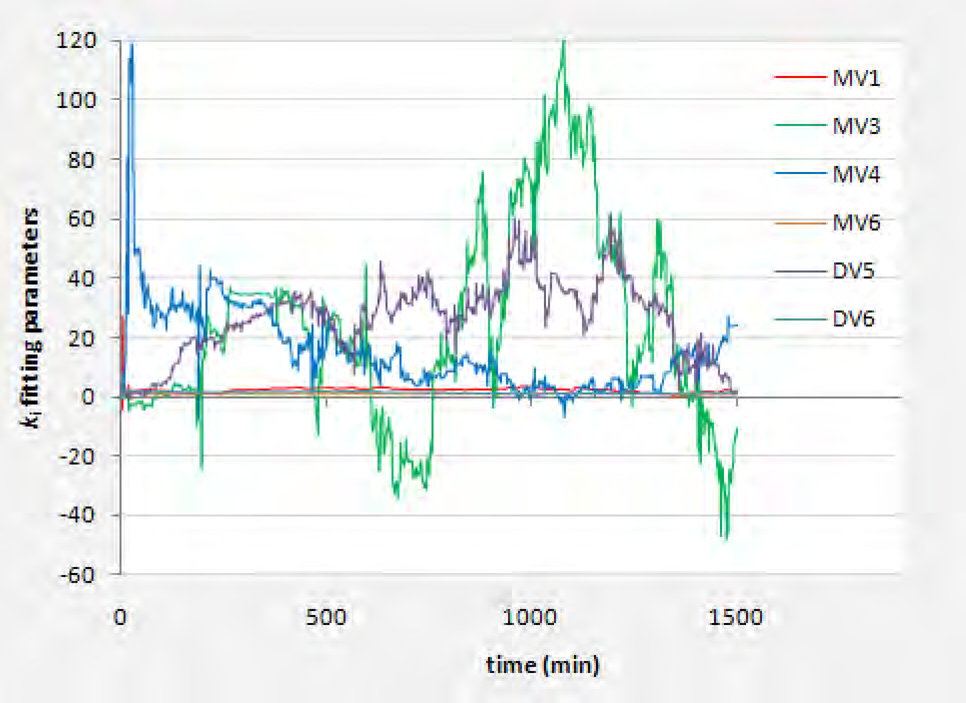


Figure C.8: Moving window regression plot for CV5

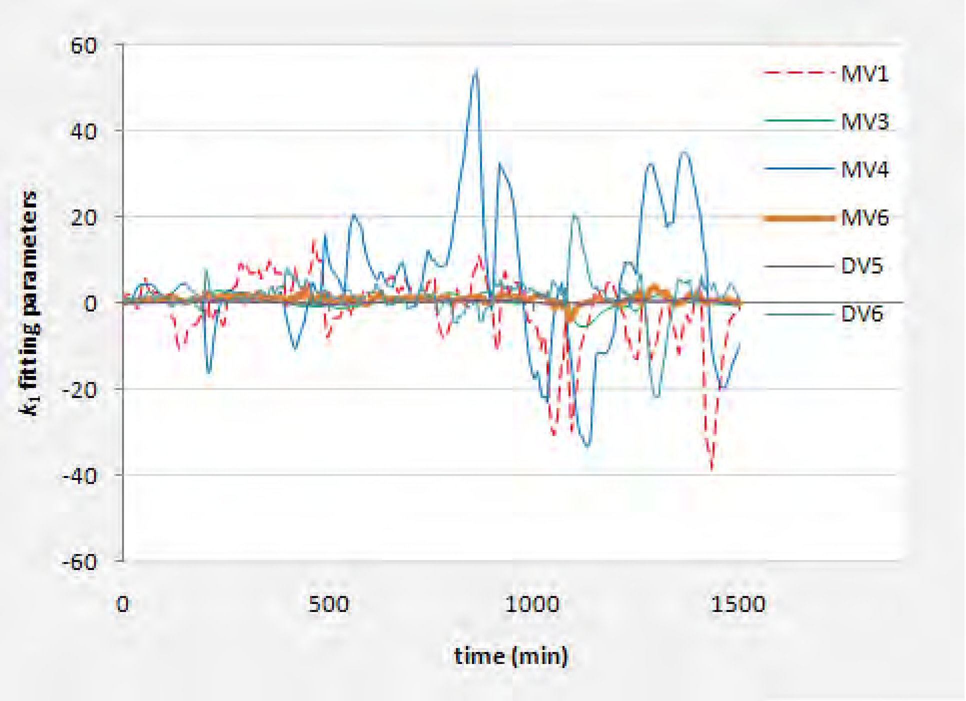


Figure C.9: Kalman Filter plot for CV5

C.4. Petrol Bottoms Product (CV6)

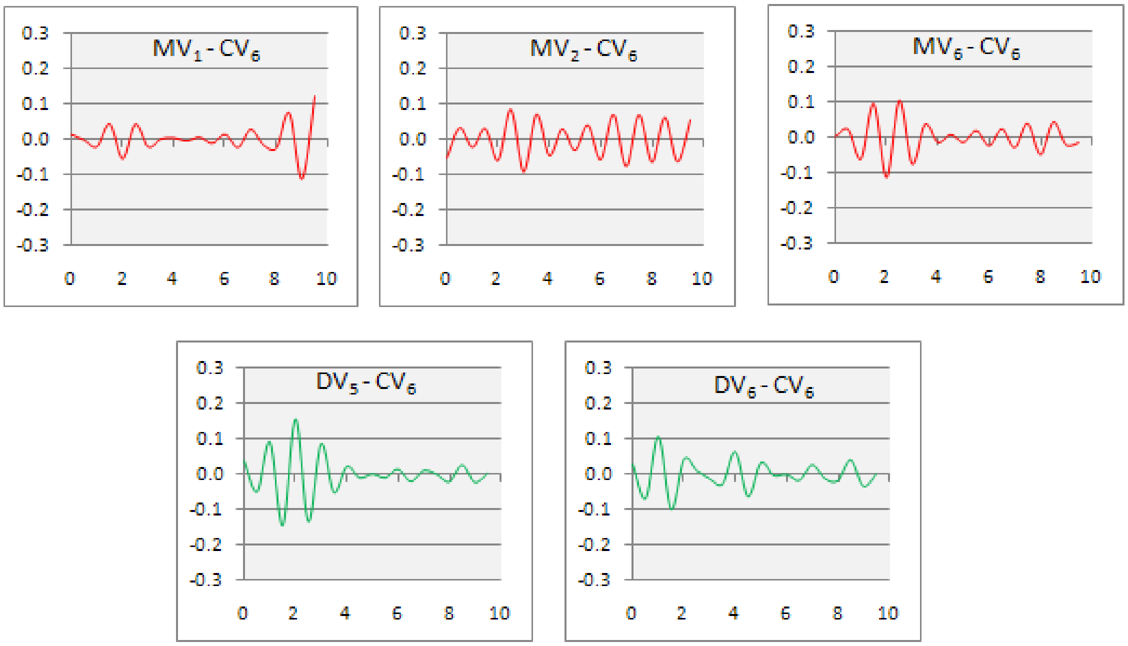


Figure C.10: Correlation plots for CV6

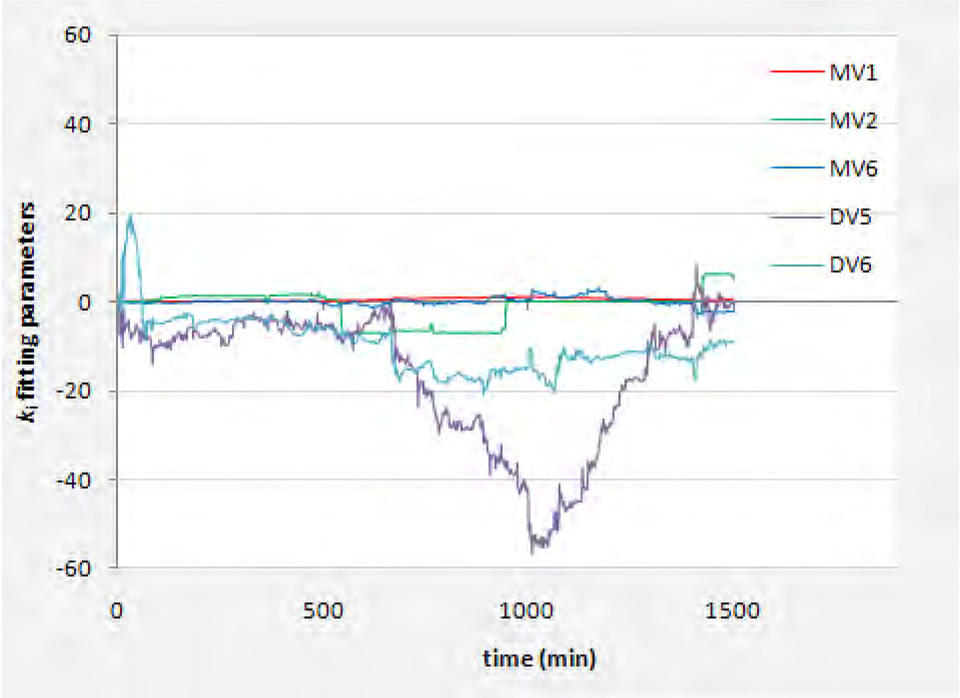


Figure C.11: Moving window regression plot for CV6

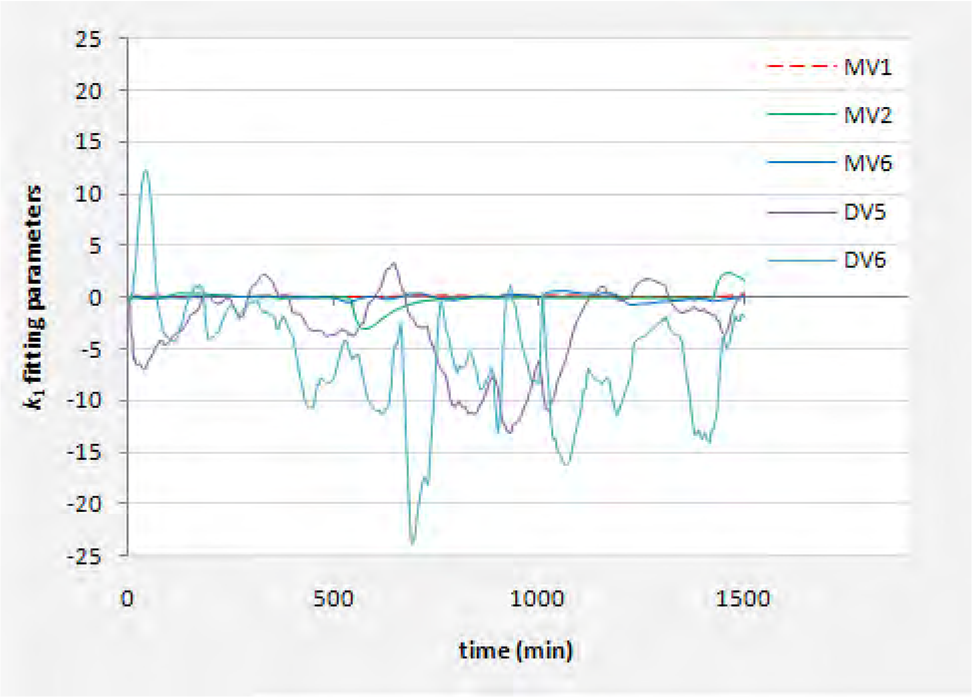


Figure C.12: Kalman Filter plot for CV6

C.5. Level in DM-1 (CV1)

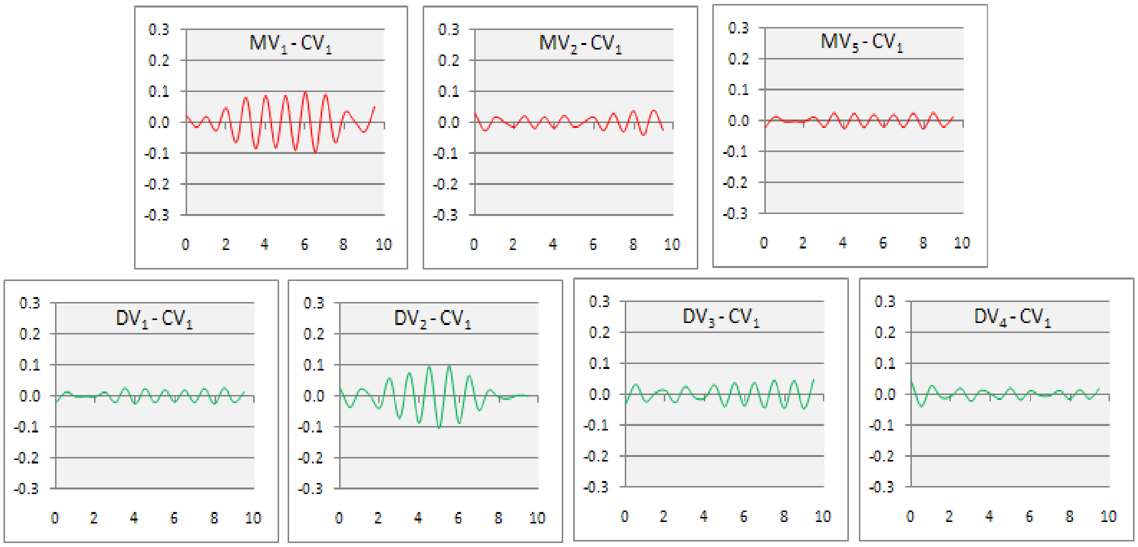


Figure C.13: Correlation Plots for CV1

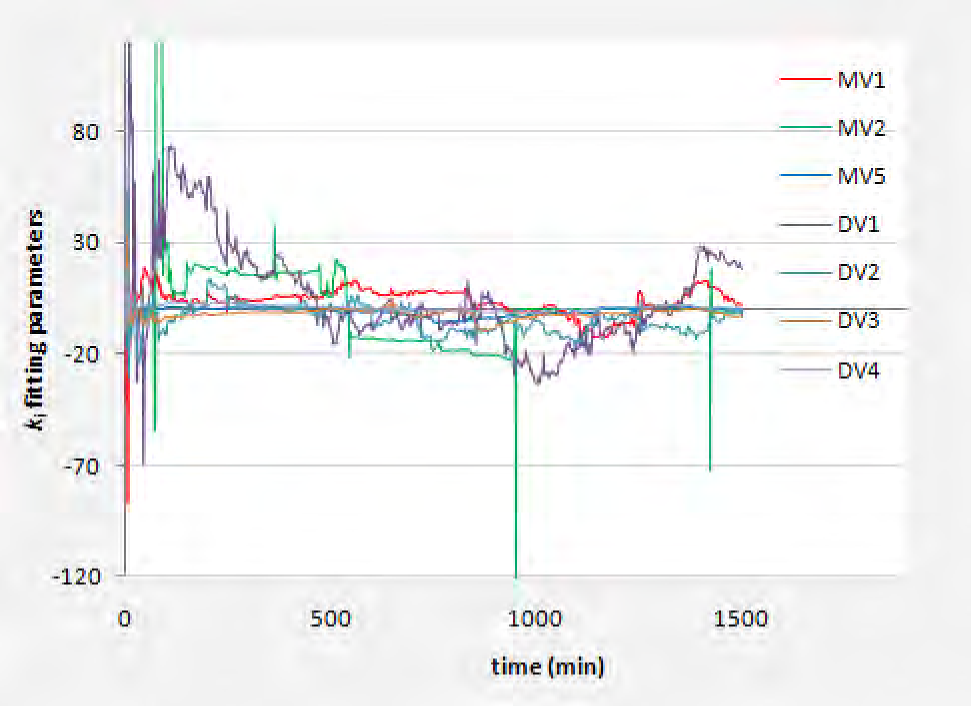


Figure C.14: Moving window regression plot for CV1

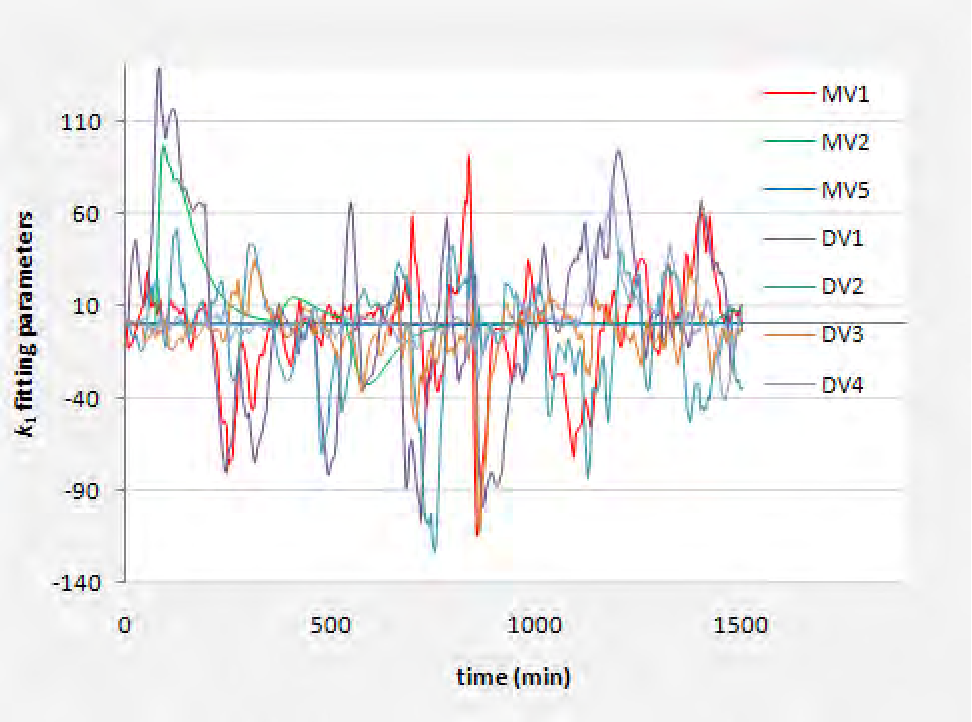


Figure C.15: Kalman Filter plot for CV1

BIBLIOGRAPHY

Astrom, K. 1967. “Computer control of a paper machine – an application of linear stochastic control theory”, *IBM J.*, 7: 389-396.

Badwe, A., Gudi, R., Patwardhan, R., Shah, S., Patwardhan, S. 2009. “Detection of model-plant mismatch in MPC applications” *Journal of process control*, 19: 1305-1313.

Basseville, M. 1998. “On-board component fault detection and isolation using the statistical local approach”, *Automatica*, 34(11): 1391-1415.

Bequette, B.W. 1991. “Nonlinear control of chemical processes: a review”, *Ind. Eng. Chem. Res*, 30(7): 1391-1413.

Brisk, M.L. 2004. “Process Control: Potential benefits and wasted opportunities”, *in the 5th Asian Control Conference*.

Corriou, J.P. 2004. “Process Control: Theory and Applications”, Springer-Verlag London Limited.

Cutler, C., Morshedi, A., Haydel, J. 1983. “An industrial perspective on advanced control”, *In AICHE annual meeting*, Washington, DC.

Cutler, C. R., Ramaker, B. L. 1979. “Dynamic matrix control-a computer control algorithm”, *AICHE national meeting*, Houston, TX.

Deshpande, P.B., Caldwell, J.A., Yerrapragada, S.S., Bhalodia, M.A. 1995. “Should You Use Constrained Model Predictive Control?”, *chem. Eng. Prog*.

Ender, D. 1993. “Process control performance: Not as good as you think”, *Control Engineering*, 9: 180-190.

Harris, T.J., Boudreau, F., Macgregor, J.F. 1996. “Performance assessment of multivariable feedback controllers”, *Automatica*, 32: 1505-1518.

Harris, T.J., Seppala, C.T., Desborough, L.D. 1999. “A review of performance monitoring and assessment techniques for univariate and multivariate control systems”, *J. Proc Control*, 9: 1-17.

Hasseloff, V., Friedman, Y.Z., Goodhart, S.G. 2007. “Implementing coker advanced process control”, *Hydrocarbon Processing*, 99-103.

Heeger, D. 2000. “Signals, linear systems and convolution”, New York University (NYU), 26th September.

Huang, B. 2002. “Minimum variance control and performance assessment of time-variant processes”, *Journal of Process Control*, 12:707-719.

Huang, B., Tamayo, E.C. 2000. “Model validation of industrial model predictive control systems”, *Pergamon*, 55: 2315-2327.

Huang, B., Shah, S. 1999. “Performance Assessment of Control Loops”, Advances in Industrial Control, Springer, London, 1999.

Huang, B., Shah, S.L., Kwok, K.E. 1997. “Good, bad or optimal? Performance assessment of multivariable processes”, *Automatica*, 6: 1175-1183.

Hugo, A.J. 2001. “Process Controller Performance Monitoring and Assessment”, *Control Arts inc*.

Jelali, M. 2005. “An overview of control performance assessment technology and industrial applications”, *Control Eng. Pract.* 14: 441–466.

Jiang, H., Shah. S.L., Huang, B. 2008. “Control relevant on-line model validation criterion based on robust stability conditions”, *Can. J. Chem. Eng*., 86: 893-904.

Kalman, R.E. 1960. “A new approach to linear filtering and prediction problems”, Transactions of ASME, *Journal of basic engineering*, 87: 35-45.

Kocijan, J., Murray-Smith, R., Rasmussen, C.E., Girard. 2003. “Gaussian Process Model Based Predictive Control”.

Kruger, U., Zhou, Y., Irwin, G.W. 2004. “Improved principal component monitoring of large-scale processes”, *J. Proc. Control*, 879-888.

Kwakermaak, H., Siven. 1972. “Linear optimal control systems”, New York, John Wiley & Sons.

Larimore, W. 1990. “Canonical Variate Analysis in Identification, Filtering and Adaptive Control”, *IEEE Conf. Decision Control*.

Ljung, L. 1985. “Asymptotic variance expressions for identified black-box transfer function models”, *IEEE Trans. Automatic Control*, 834-844.

Ljung, L. 1989. “ System Identification: theory for the user”, Prentice Hall, Englewood, Cliffs, NJ.

Ljung, L. 1999. “System identification theory for the user”, 2nd ed. Saddle River (NJ), PTR Prentice Hall.

Ljung, L. 2002. “Prediction error estimation methods”, *Circuits, Systems, and Signal Processing*, 21(1): 11-21.

Loquasto, F., Seborg, D.E. 2003. “Monitoring Model Predictive Control Systems using Pattern Classification and Neural Networks”, *Ind. Eng. Chem. Res*, 42(20): 4689-4701.

Loquasto, F., Seborg, D.E. 2003. “Model Predictive Controller Monitoring based on Pattern Classification and PCA”, *Proceedings of the American Control Conference*.

Maciejowski, J.M. 2002. “Predictive control with constraints” Harlow (England), Prentice Hall.

Mathworks. 2010. “Residual Analysis”, available from www.mathworks.com (Accessed: 25 September 2010)

Moradkhani, H., Soorooshian, S., Gupta, H.V., Houser, P.R. 2005. “Dual state-parameter estimation of hydrological models using ensemble Kalman Filter”, A*dvances in water resources*, 28: 138-147.

Morari, M., Garcia, C.E., Prett, D.M. 1989. “Model predictive control: Theory and practice – a survey”, *Automatica*, 25: 335-348.

Miller, R., Desborough, L. 2001. “Increasing customer value of industrial control performance monitoring- Honeywell’s experience”, *CPS VI meeting*.

Patwardhan, R.S., Shah, S.L. 2002. “Assessing the performance of model predictive controllers”, *Can. J. Chem. Eng*, 80: 954-966.

Patwardhan, R.S., Shah, S.L. 2002. “Issues in performance diagnostics of model-based controllers”, *Journal of Process Control*, 12: 413-427.

Prett, D., Morari, M., “Shell control problem”, *Shell Process Control Workshop,* Butterworth Publishers, Stoneham, MA, 355-360.

Qiang, Z., Shaoyuan, L. 2006. “Performance Monitoring and Diagnosis of Multivariable Model Predictive Control using Statistical Analysis”, *Chinese J. Chem Eng*., 14(2): 207-215.

Qin, S., Badgwell, T.A. 2003. “A survey of industrial model predictive control technology”, *Control Engineering Practice*, 11: 733-764.

Riaz, R.A., Butt, M.F.U., Chen, S., Hanzo. 2008. “Generic z-domain discrete-time transfer function estimation for ultra-wideband systems”, *ELECTRONIC LETTERS*, 44(25).

Richalet, J., Rault, A., Testud, J., Papon, J. 1976. “Algorithmic control of industrial processes”, *In Proceedings of the 4th IFAC symposium on identification and system parameter estimation*, 1119-1167.

Richalet, J., Rault, A., Testud, J., Papon, J. 1976. “Model Predictive Heuristic Control: Applications to industrial processes”, *Automatica*, 14: 413-428.

SASOL PTY LTD. 2010. “Explore SASOL”, available from www.sasol.com (Accessed: 15 September 2010)

Schafer, J., Cinar, A. 2004. “Multivariable MPC system performance assessment, monitoring, and diagnosis”, *J. Proc. Control*, 14: 113-129.

Seborg, D., Edgar, T., Mellichamp. 2004. “Process Dynamics and Control”, John Wiley & Sons.

Sekara, T.B., Stojic, M.R. 2005. “Application of the α-approximation for Discretization of Analogue systems”, ELEC. ENERG., 18(3): 571-586.

Selvanathan, S., Tangirala, A.K. 2010. “Diagnosis of Poor Control Loop Performance due to Model-Plant Mismatch”, *Ind. Eng. Chem. Res*, 49(9): 4210:4229.

Shridhar, R., Cooper, D.J. 1998. “A tuning strategy for unconstrained multivariable model predictive control”, *Ind. Eng. Chem. Res*., 37: 4003-4016.

Soderstrom, T., Staica, P. 1993. “System Identification”, Prentice-Hall, international, U.K.

Stoorvogel, A.A., 2010. “Model Predictive Control”, [lecture notes], *Disc course, Lecture notes*. University of Twente.

Thornhill, N.F., Oettinger, M. 1999. “Refinery-wide control loop performance assessment”, *J. Proc Control*, 9: 109-124.

Van Gerpen, J., Shanks, B., Pruszko, R. 2004. “Biodiesel Analytical Methods”, National Renewable Energy Laboratory, Colorado, U.S.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N., Yin, K. 2003. “A review of process fault detection and diagnosis Part III: Process history based methods”, *Computers and Chemical Engineering,* 27: 327-346.

Webber, J., Gupta, Y. 2008. “A closed-loop cross-correlation method for detecting model mismatch in MIMO model-based controllers”, *ISA transactions*. 47: 395-400.

Weyer, E., Bell, G., Lee, P.L. 1999. “System identification for generic model control”, *Journal of Process Control*, 9: 357-364.

Zhang, Q., Basseville, M., Benveniste, A. 1994. “Early warning of slight changes in systems”, Automatica, 30(1): 95-113.

Zhu, Y.C., Van Wijck, M., Janssen, E., Graaf, A.J.M., Van Aalst, C.H., Kievt, L. 1997. “Crude unit identification for MPC using ASYM method”, *Proceedings of American Control Conference*, 3395-3399.