System Identification and Model-Based Control of a Filter Cake Drying Process

by

Johannes Jacobus Wiese

Thesis submitted in partial fulfilment
of the requirements for the degree

Master of Science in Engineering

(Chemical Engineering)

in the Faculty of Engineering

at the Stellenbosch University

Supervised by

Professor Chris Aldrich

March 2011

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

…….. ……

Signature Date

Abstract

A mineral concentrate drying process consisting of a hot gas generator, a flash dryer and a feeding section is found to be the bottleneck in the platinum concentrate smelting process. This operation is used as a case study for system identification and model-based control of dryers. Based on the availability of a month's worth of dryer data obtained from a historian, a third party modelling and control software vendor is interested in the use of this data for data driven model construction and options for dryer control. The aimed contribution of this research is to use only data driven techniques and attempt an SID experiment and use of this model in a controller found in literature to be applicable to the dryer process. No first principle model was available for simulation or interpretation of results. Data were obtained for the operation from the plant historian, reduced, cleaned and investigated for deterministic information through surrogate data comparison – resulting in usable timeseries from the plant data. The best datasets were used for modelling of the flash dryer and hot gas generator operations individually, with the hot gas generator providing usable results.

The dynamic, nonlinear autoregressive models with exogenous inputs were identified by means of a genetic programming with orthogonal least squares toolbox. The timeseries were reconstructed as a latent variable set, or "pseudo-embedding", using the delay parameters as identified by average mutual information, autocorrelation and false nearest neighbours. The latent variable reconstruction resulted in a large solution space, which need to be investigated for an unknown model structure. Genetic Programming is capable of identifying unknown structures. Freerun prediction stability and sensitivity analysis were used to assess the identified best models for use in model based control. The best two models for the hot gas generator were used in a basic model predictive controller in an attempt to only track set point changes.

One step ahead modelling of the flash dryer outlet air temperature was unsuccessful with the best model obtaining a validation R^2 = 43%. The lack of process information

contained in the available process variables are to blame for the poor model identification. One-step ahead prediction of the hot gas generator resulted in a top model with validation R^2 = 77.1%.

The best two hot gas generator models were implemented in a model predictive controller constructed in a real time plant data flow simulation. This controller's performance was measured against set point tracking ability. The MPC implementation was unsuccessful due to the poor freerun prediction ability of the models. The controller was found to be unable to optimise the control moves using the model. This is assigned to poor model freerun prediction ability in one of the models and a too complex freerun model structure required. It is expected that the number of degrees of freedom in the freerun model is too much for the optimiser to handle. A successful real time simulation architecture for the plant dataflow could however be constructed in the supplied software.

It is recommended that further process measurements, specifically feed moisture content, feed temperature and air humidity, be included for the flash dryer; closed loop system identification be investigated for the hot gas generator; and a simpler model structure with smaller reconstructed latent variable regressor set be used for the model predictive controller.

Opsomming

‘n Drogings proses vir mineraal konsentraat bestaan uit drie eenhede: ‘n lug verwarmer-, ‘n blitsdroeër- en konsentraat toevoer eenheid. Hierdie droeër is geïdentifiseer as die bottelnek in die platinum konsentraat smeltingsproses. Die droeër word gebruik as ‘n gevallestudie vir sisteem identifikasie asook model-gebasseerder beheer van droeërs. ‘n Maand se data verkry vanaf die proses databasis, het gelei tot ‘n derde party industriële sagteware en beheerstelsel maatskappy se belangstelling in data gedrewe modelering en beheer opsies vir die drogings proses. Die doelwit van hierdie studie is om data gedrewe modeleringstegnieke te gebruik en die model in ‘n droeër-literatuur relevante beheerder te gebruik. Geen eerste beginsel model is beskikbaar vir simulasie of interpretasie van resultate nie. Die verkrygde data is gereduseer, skoon gemaak en bestudeer om te identifiseer of die tydreeks deterministiese inligting bevat. Dit is gedoen deur die tydreeks met stochastiese surrogaat data te vergelyk. Die mees gepaste datastelle is gebruik vir modellering van die blitsdroeër en lugverwarmer afsonderlik.

Die nie-liniêre, dinamiese nie-linieêre outeregressie modelle met eksogene insette was deur ‘n genetiese programmering algoritme, met ortogonale minimum kwadrate, identifiseer. Die betrokke tydreeks is omskep in ‘n hulp-veranderlike stel deur gebruik te maak van vertragings-parameters wat deur gemiddelde gemeenskaplike inligting, outokorrelasie en vals naaste buurman metodes verkry is. Die GP algoritme is daartoe in staat om the groot oplossings ruimte wat deur hierdie hulp-veranderlike rekonstruksie geskep word, te bestudeer vir ‘n onbekende model struktuur. Die vrye vooruitskattings vermoë, asook die model sensitiwiteit is inag geneem tydens die analiese van die resultate. Die beste modelle se gepastheid tot model voorspellende beheer is gemeet deur die uitkomste van ‘n sensitiwiteits analise, asook ‘n vrylopende voorspelling, in oënskou te neem.

Die een-stap vooruit voorspellende model van die droeër was onsusksesvol met die beste model wat slegs ‘n validasie R^2 = 43% kon behaal. Die gebrekkige meet

instrumente in die droeër is te blameer vir die swak resultate. Die een-stap vooruit voorspellende model van die lug verwarmer wat die beste gevaar het, het ‘n validasie R^2 = 77.1% gehad.

‘n Basiese model voorspellende beheerder is gebou deur die 2 beste modelle van slegs die lugverwarmer te gebruik in ‘n intydse simulasie van die raffinadery data vloei struktuur. Hierdie beheerder se vermoë om toepaslike beheer uit te oefen, is gemeet deur die slegs die stelpunt te verander. Die beheerder was egter nie daartoe in staat om die insette te optimeer, en so die stelpunt te volg nie. Hierdie onvermoë is as gevolg van die kompleks vrylopende model struktuur wat oor die voorspellingsvenster optimeer moet word, asook die onstabiele vryvooruitspellings vermoë van die modelle. Die vermoede is dat die loslopende voorspelling te veel vryheids grade het om die insette maklik genoeg te optimeer. Die intydse simulasie van die raffinadery se datavloei struktuur was egter suksesvol.

Beter meting van noodsaaklike veranderlikes vir die droër, o.a. voginhoud van die voer, voer temperatuur, asook lug humiditeit; geslotelus sisteem identifikasie vir die lugverwarmer; asook meer eenvoudige model struktuur vir gebruik in voorspellende beheer moontlik vermag deur ‘n kleiner hulp veranderlike rekonstruksie te gebruik.

Dedications

The masters study is dedicated to Hans Heunis, Con Blom, Dumbo van der Westhuizen and Stian Swart who spent many a mile on the bike saddle with me. Without the escape to the mountains I would surely have gone mad.

Acknowledgements

Professor Chris Aldrich for his guidance, knowledge and patience.

CSense Systems and Department of Process Engineering for the bursary and fellowship opportunity allowing me to pay for the studies as well as equip me with experience.

God, because who and what he made me is His gift to me; who and what I become is my gift to Him.

Glossary

Concentrate	PGM mineral containing substance received from the concentrator process where the PGM rich substance is separated from the non PGM containing ore. This separation is generally done by a flotation process. The feed from the concentrators to the smelting furnace is known as concentrate.
Advanced Process Control	APC is s term with a loose definition defined by exclusion rather than definition. APC refers to any control strategy, or applied algorithm or computer logic, which is not seen as standard PID feedback control.
Historian	A data repository used to store and allow access to historic data. Commonly it is a server on a network with specific software installed to manage data collection, storage and retrieval over a network.
Latent Variable	A variable constructed from the process variables for use by the system identification algorithm. The latent variables assist with the identification of which process delays to be included in the model structures. A set of latent variables are constructed for each process variables, with each latent variable being a different process delay.
Matte	Metal as obtained from the smelting process resulting from the separation of the metal (matte) and slag during smelting. It has a PGM concentration higher than concentrate.
OPC	Ole DB for Process Control. A communication standard used to communicate data between devices in the control network of an operation.
SCADA	Supervisory Control and Data Acquisition. It specifically refers to the computer hardware and software architecture required for collecting measurement data online and controlling processes according to predetermined logic based on these measurements.
Six Sigma	A methodology followed to identifying areas of high variation and the causes of these. Ultimately it encompasses an entire philosophy followed in an attempt to drive continuous improvement.

Abbreviations

AMI	Average Mutual Information
APC	Advanced Process Control
DCS	Distributed Control System
DMC	Dynamic Matrix Control
FD	Flash Dryer
FDFeed	Concentrate Feed to the Flash Dryer (used in emp
FDTemp	Flash Dryer Air Outlet Air Temperature
FNN	False Nearest Neighbours
GA	Genetic Algorithm
GP	Genetic Programming
GPOls	Genetic Programming with Orthoginal Least Squares
HGG	Hot Gas Generator
HGGTemp	Hot Gas Generator Outlet Air Temperature
IMC	Internal Model Controller
MPC	Model-based Predictive Control / Model Predictive Control
MSE	Mean Square Error
NARX	Nonlinear Autoregressive Model with eXogenous inputs
NN	Neural Network
OLS	Orthogonal Least Squares
OPC	OLE DB for Process Control
PGM	Platinum Group Metals
PLC	Programmable Logic Controller
RMS	Root Mean Square
SCADA	Supervisory Control and Data Acquisition
SID	System Identification
SISO	Single-Input Single-Output
SP	Set Point
SSE	Sum Square Error
SST	Total Sum of Squares
VSD	Variable Speed Drive

Table of Contents

Chapter 1 Introduction

1.1 Background

The South African economy was built on mining through the past century and has become a leading role player in the world with regards to metal and mineral production. This position developed from the gold rush in the Transvaal in the late 1800's to the situation today where South Africa not only directly supplies a major constituency of platinum, coal, gold and steel to the market, but also produces 304 mining engineers per annum compared to the 130 from Australia, 127 from Canada and 35 from the United States (Landelahni, 2010). The dependence of the South African people and economy on the minerals and mining sectors accentuates the need to keep these industries alive and running at optimum throughput producing world class products. This will help ensure South Africa's future in a market threatened by the rise of market share by the BRIC countries (Brazil, Russia, India and China).

The minerals and mining industry contributed 7.7% of the South African gross domestic product (GDP) in 2007 at R223.9 billion in sales. (South African Department of Minerals and Energy, 2007). This amount is mainly divided between platinum group metals – PGM's - (35%), coal (19%) and gold (17%) for 2007 of which the market has shifted to be largely dependent on PGM's. The growth in PGM sales is evident in Figure 1 and Figure 2.

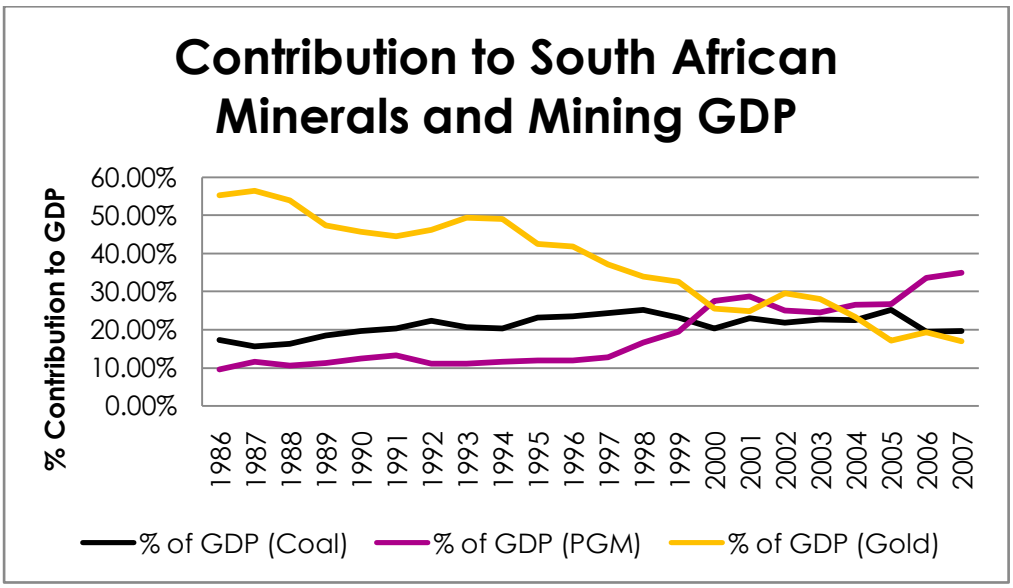


Figure 1: Contribution of PGM's, Gold and Coal to the mining segment of the South African GDP (after South African Department of Minerals and Energy, 2007).

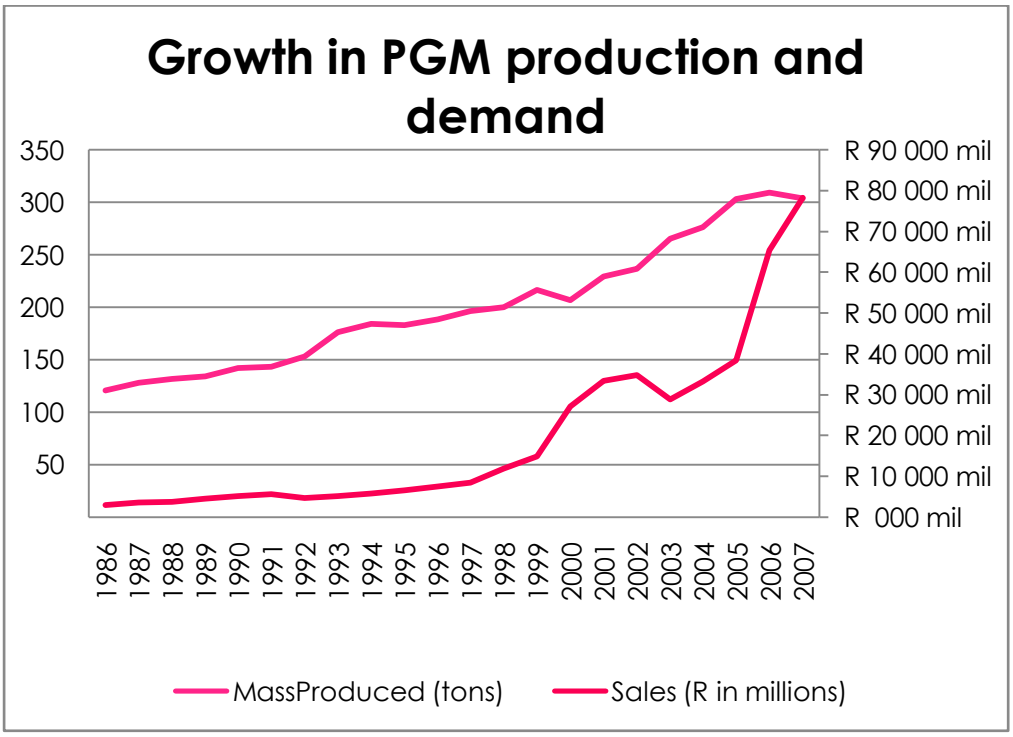


Figure 2: Growth in demand and sales of platinum group metals for the period 1986-2007 (after South African Department of Minerals and Energy, 2007).

The contribution gold made to the minerals and mining portion of the South African GDP has decreased to the point that it is equivalent to the contribution of coal in sales value. PGM demand has however overtaken both coal and gold as the single major contributor to the

minerals and mining sector. The shift is largely contributed to the steep increase in PGM prices since 2004 as is visible in Figure 2.

The South African gold economy has thus been upgraded to platinum.

South Africa produces 57% of the world's PGM's but holds 87.7% of the estimate global PGM reserves (South African Department of Minerals and Energy, 2008). This, together with the trends discussed earlier, places PGM producers, and South Africa, in an extremely favourable position for the future. The focus of this research will be on one of the major platinum producers in South Africa.

The value chain for the PGM extraction process under study is included in Figure 3 below. The focus of this research is on the smelting operations of the PGM value chain.

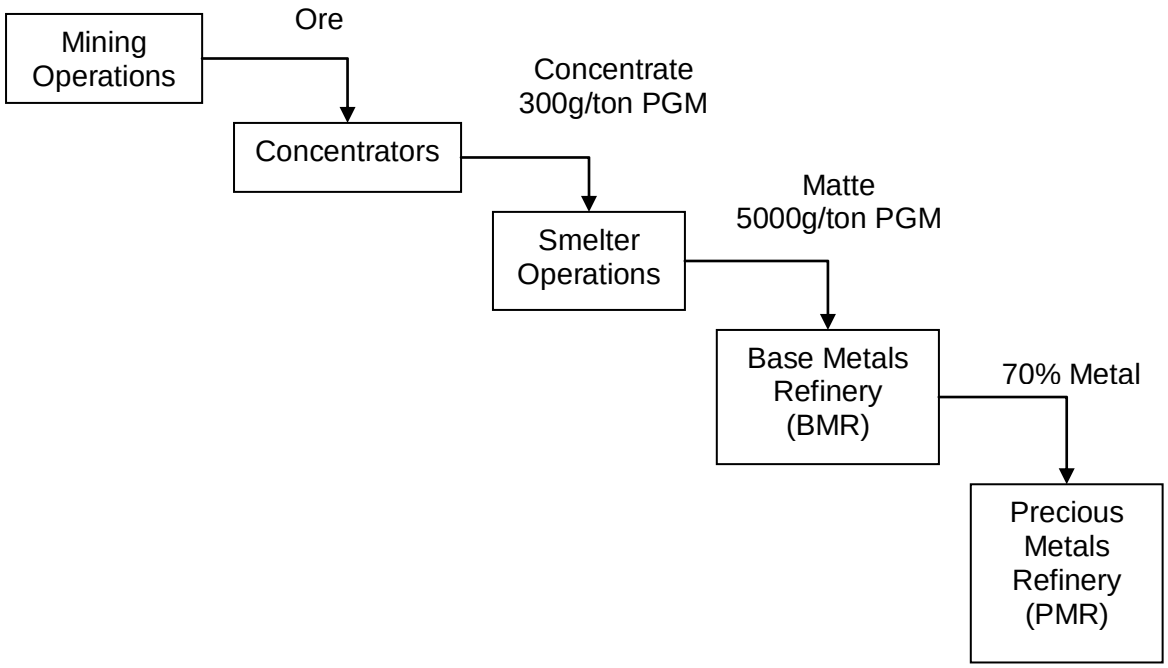


Figure 3: The PGM value chain and corresponding operations. The focus of this research is on the Smelter Operations.

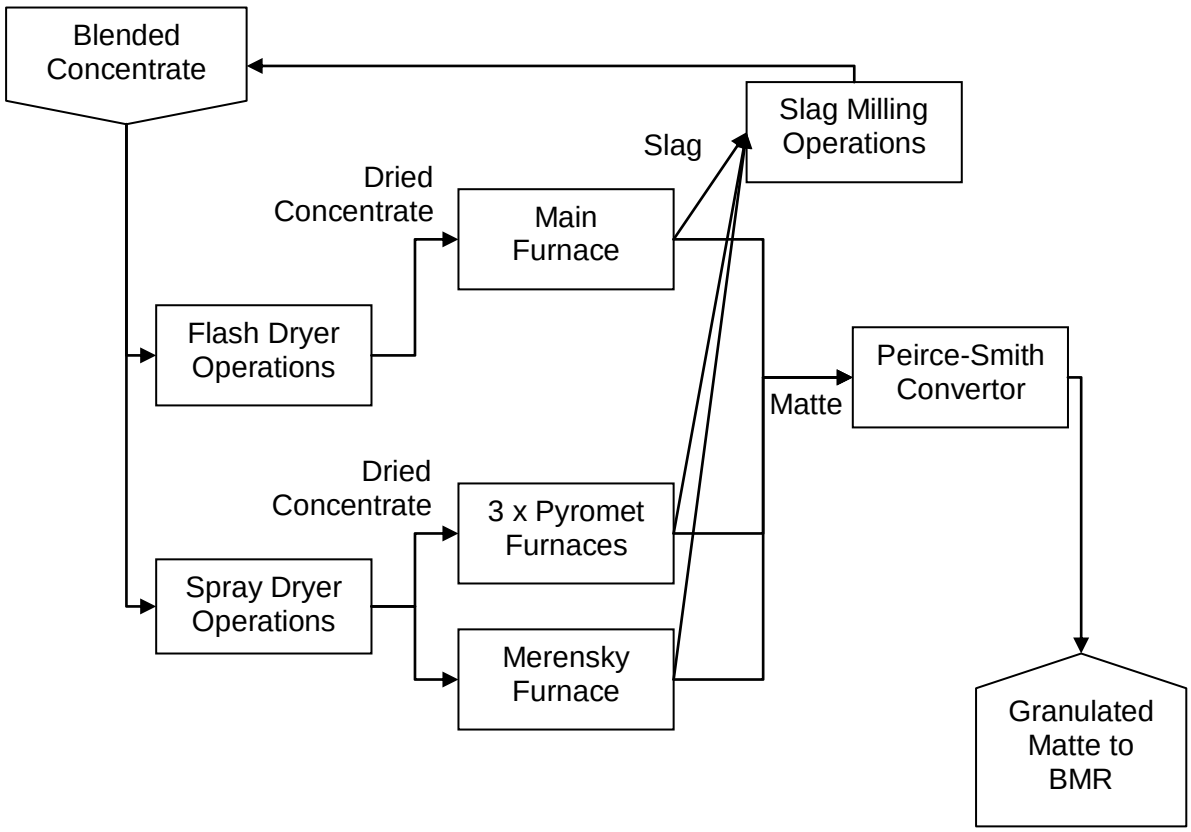


Figure 4: Smelting Operations at the Base Metals Refinery (BMR) complex under study. The focus of this study is on the Flash Dryer Operations.

The main steps in the smelting operations are portrayed in Figure 4. The concentrate blend consisting of a low, medium and high chromite content blend is fed to the drying operations. The flash dryer operations feed only the main furnace, a 28 MW arc furnace. The spray dryer operations feed the remaining three pyrometallurgical furnaces and the Merensky furnace. The slag produced by the five furnaces is recycled through the slag mill and fed into the concentrate blend. The matte is fed to the convertor, whereafter it is granulated and sent to the base metal refining operations. The focus of this research will be on the flash dryer operations preceding the main furnace.

As depicted earlier, the second highest contributor to South African minerals and mining is coal. South African domestic coal prices were 3 times lower than export prices. However domestic sales, and thus use, was 3 times as high as exports (South African Department of Minerals and Energy, 2007). The abundance of cheap coal as energy source does not help the drive toward reduced emissions and optimised use of energy resources. However, with

carbon penalties in the future, the drive to use fuel more efficiently will acquire more attention.

1.2 Problem Statement

As a third party solution vendor, CSense Systems was responsible for scoping and delivering a control solution for the flash dryer operations. The industry interest in the dryer market triggered the exploration of modelling and control options for dryers, resulting in this specific research focussing on model predictive control of this dryer setup.

The problem statement is:

> Investigate the control method preferred in literature for dryers; identify a system identification strategy which handles the challenges in dryer modelling, use flash dryer operation data available in a historian for system identification with the aim of developing a controller for the operation.

The area of research is

- system identification of a dynamic dryer model from historic plant data; and
- Investigation of applying a model predictive controller by making use of the nonlinear model found in the system identification step, together with the required data preparation in a real time plant data flow environment.

The aim of this research is to:

- Identify from literature the preferred control technique for dryer operations;
- Identify challenges in system identification of dryer models and address these shortcomings by making use of a SID methodology and algorithm able to handle the challenges;
- Investigate the applicability of a data preparation and system identification methodology in a real life case study and recommend any alterations to the methodology for the problem investigated;

- Measure the system identification technique against similar techniques, or more basic model structures to identify if the technique is comparable;
- Investigate alternate SID parameters and additions to the technique used;
- Review the process measurements available against process variables used in literature, stating the additional measurements required and recommended for research in future dryer studies;
- Identify the section of the filter cake drying circuit most suitable for system identification and control given the available plant measurements;
- Compare the controller recommended from literature to the current plant controller, to see whether the recommended control is able to track the set point better, and with a tighter variation.
- Investigate the application of the identified model for the required controller in the CSense software package in a simulation of plant data flow;

1.3 Thesis Structure

This thesis is divided into the background- , the methodology- and the results sections. The background section is incorporated into chapters 2 and 3. Chapters 4, 5 and 6 set out the methodology used and background surrounding it, with the results included in chapters 7, 8 and 9. Chapter 10 ties down the thesis with the conclusions and recommendations resulting from the case study.

The thesis structure is set out below.

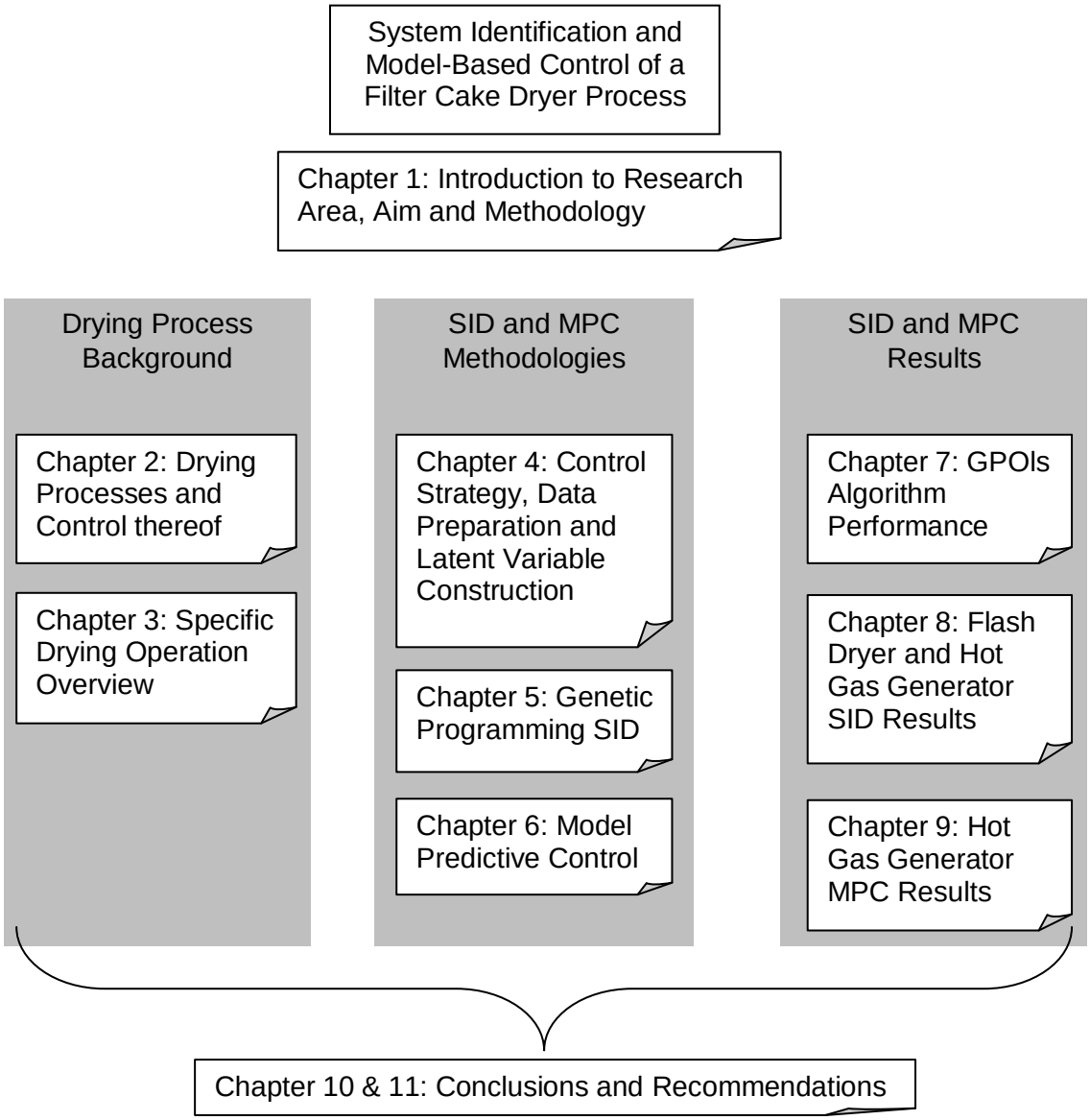


Figure 5: The Thesis Structure is divided into 3 main sections for the Background, Methodology and Results, with the Introduction and Conclusions section before and after the main body.

Chapter 2 provides details with regards to drying dynamics and operations. The business case for control is introduced with a discussion surrounding control strategies investigated in literature. Chapter 3 shifts the general dryer focus to the specific dryer operation being investigated. The dryer is put into context of the PGM smelting operation and the problem statement is set out. Control is introduced as an answer to these problems.

Chapter 4 introduces the researches methodology structure and then shifts the focus to the control strategy with the data preparation required. Data preparation makes up a large part of this case study as the plant historian data posed various challenges. The choice of dataset to use and the latent variable reconstruction of the timeseries end off the chapter.

The GP algorithm and adjustable parameters are discussed in chapter 5. Various additions to the experimentation technique are mentioned with the chapter concluding with the GP-based SID approach followed for the specific case study. Chapter 6 discusses model-based predictive control and the experimental setup created for the MPC experiments. The various choices in MPC design parameters are set out in the chapter.

The results section starts of in Chapter 7 by looking at the choice of GP parameters and toolbox additions and how each of these influenced the experiment results. The specific GPOls algorithm is compared to another commercially available GP toolbox, Discipulus ®. The choice of GP approach as SID method is also compared against linear modelling results for the process.

The system identification results, in chapter 8, are discussed for each of the flash dryer and hot gas generator. The choice of timeseries and data preparation results also make out a large portion of this chapter, seeing as this step was crucial to using the historian data. Chapter 9 investigates the applicability of the models for MPC and compares the control to a random choice in manipulated variable. This is only investigated for the hot gas generator seeing as no model could be identified for the flash dryer.

The concluding remarks, main findings and recommendations are included in chapter 10.

DRYING PROCESS BACKGROUND

Chapter 2 Literature Survey of the Control of Drying Processes

2.1 The Drying Process

2.1.1 Introduction to Drying Operations

Dryers in industry have their origin in mechanical design based on process needs and experience, with little regard to theoretical knowledge. The bridge between practice and research have caused a lot of process knowledge to be locked in practice and not accessible to researchers, as manufacturers seldom disclose specifics. This separation of knowledge and specific designs according to specific needs have caused dryer operations to evolve into a thicket of types, setups, control methods and approaches (Mujumdar, 1995). This inaccessibility of information, high variety of speciality drying setups and low regard of modelling research by the dryer industry through the decades have caused theoretical knowledge and modelling to be unattended by researchers. There has however been a collection of research in the past 2 decades allowing better understanding of dryers, although the link between these hard, rugged machines and the theory and research remain somewhat separated (Wang et al., 2007).

2.1.2 Drying Process Dynamics

The drying process is the removal of a liquid, either bound in the microstructure or pores of the solid or located on the solid surface. Moisture is removed from a solid by means of evaporation or vaporisation through heat exchange heating the liquid and mass flow to remove the vapour. The humidity, temperature and pressure of the air used for drying; the enthalpy and vapour pressure of the moisture in the solid; as well as the solid characteristics and temperature all influence the drying dynamics.

Seeing as a dryer will remove moisture up to the point of equilibrium, it is a self regulating process. This equilibrium point is however dependant on the air humidity and temperature of both the air and the solid. The efficiency of the dryer is also largely dependent on the residence time of the particles in the dryer with longer residence times equalling drier

products due to more time for heat transfer and moisture removal. The following diagram illustrates the profile and how the conditions change through the profile of a generic dryer with a horizontal material and air flow in a direct concurrent drying process.

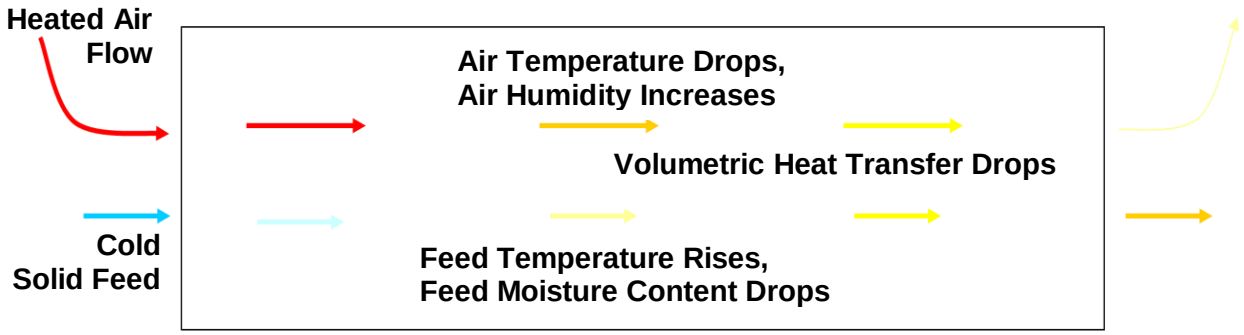


Figure 6: Dryer dynamics along the profile of a direct, co-current dryer

As the heated air moves through the dryer, it cools down and its humidity increases. The solid particles heat up and lose moisture. It is clear that the residence time, together with the air heat and humidity as well as solids moisture, will determine the if the point of equilibrium will be reached.

Through the process of drying, as the states change and liquid is removed, the composition of the feed changes and the surface exposure of water, to heat and air flow, is reduced. This causes a drop in the moisture removal rate. The drying rate changes from a constant moisture removal rate, generally referred to as stage 1, to a decreased rate, stage 2. A second drop occurs when the surface of the solid is dry, thus isolating heat from the moisture and decreasing the liquid removal rate during the 3rd drying stage. This is illustrated in the rate-of-drying curve under constant drying conditions (Mujumdar, 1995).

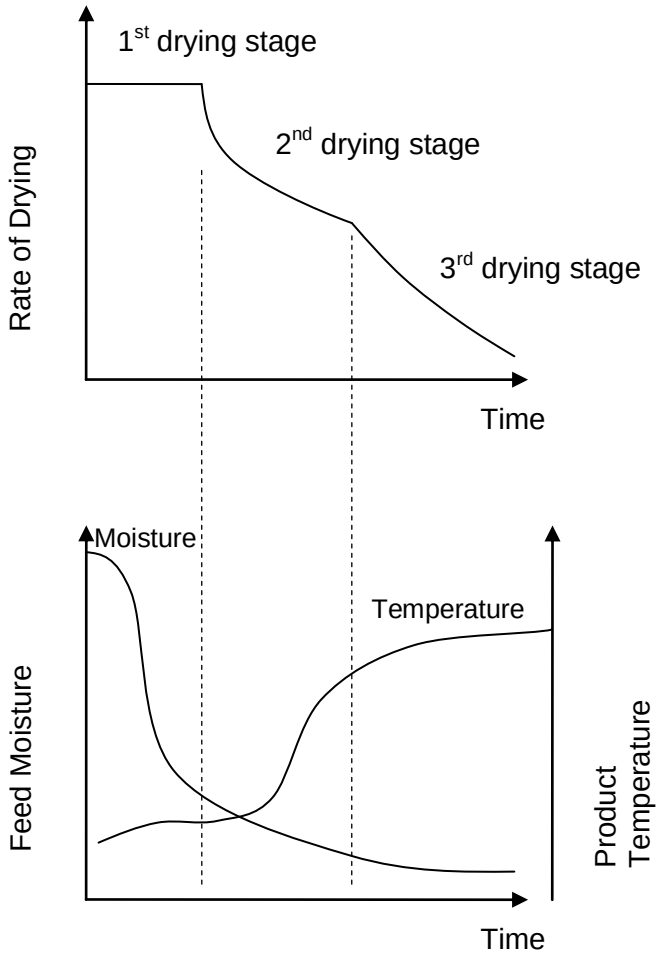


Figure 7: Typical rate of drying curve for a solid condition inserted on a graph of typical feed moisture content and feed temperature indicates conceptually that the most moisture is removed with the least effort (after Mujumdar, 1995). More moisture removal requries an exponential increase in temperature and energy admitted.

The typical residence time of feedstock in a dryer depends on the material under discussion, the dryer operation and the amount of moisture removal required. As indicated in Figure 7 the unbound surface moisture is removed quickly during stage 1, but longer drying times are needed to remove increased amounts of the bound moisture. During these longer drying times the product temperature increases, which may influence the end material composition and product quality in the case of heat sensitive products. This trade off between residence time, product moisture, and product temperature depends on the feed characteristics, product requirements and influence of high temperatures on the feedstock. This influences the choice of which dryer to use.

2.1.3 Dynamics of the Flash Dryer

Dryers are classified according to the method of heat admission, the characteristics of the solids bed, the methods of material handling and the direction of air flow. The decision in each of these depends on material types and the required residence time of the feedstock in the dryer.

Seeing as a dryer will remove moisture up to the point of equilibrium, drying is a self regulating process. This equilibrium point is however dependant on the air humidity and temperature of both the air and the solid. The efficiency of the dryer is also largely dependent on the residence time of the particles in the dryer with longer residence times resulting in a dryer product as is apparent from Figure 7. The following table indicates the time material is exposed to heat conditions with regards to the type of dryer being used.

Table 1: Heat Exposure Times of Solids (after Mujumdar, 1995)

		Typical Residence Time in Dryer			
	Dryers	0-10 sec	10-30 sec	10-60 min	1-6 h
Convection	Belt Conveyor			✓	
	Flash	✓			
	Fluid Bed			✓	
	Rotary			✓	
	Spray		✓		
	Tray (Batch)				✓
	Tray (Continuous)			✓	
Conduction	Drum		✓		
	Steam Jacket Rotary			✓	
	Steam Tube Rotary			✓	
	Tray (Batch)				✓
	Tray (Continuous)			✓	

In the filter cake drying process the mineral concentrate only needs to be exposed to the drying air for a short period as only the unbound surface moisture needs to be removed and the throughput needs to be high, hence the choice of a flash dryer.

Flash dryers fall in the category of suspended bed dryers together with fluid bed and spray dryers and are the simplest form of pneumatic dryers. Pneumatic dryers comprise all dryers where the drying air also serves for conveying the feed from points A to B (Korn, 2001). As

in fluid bed dryers, this is enabled by an air flow greater than the terminal velocity of the particles causing the particles to be lifted and eventually transported by the airflow.

Wet feed is introduced into the dryer by a screw feeder or conveyor at the bottom of a vertical tube. The heated air is introduced as a swirling airflow accelerating the wet particles upward. As these particles travel with the heated air, they are heated and moisture is removed. At the top of the tube the dry product is separated from the air flow by means of cyclones and normally a filter before the air is emitted into the atmosphere. A flash dryer generally consists of a feed section, a vertical dryer section and a particle separator section. A further section could include the air heating section, but this can also be part of another process or a separate burner. The diagram below, Figure 8, is a basic depiction of how a flash dryer operates.

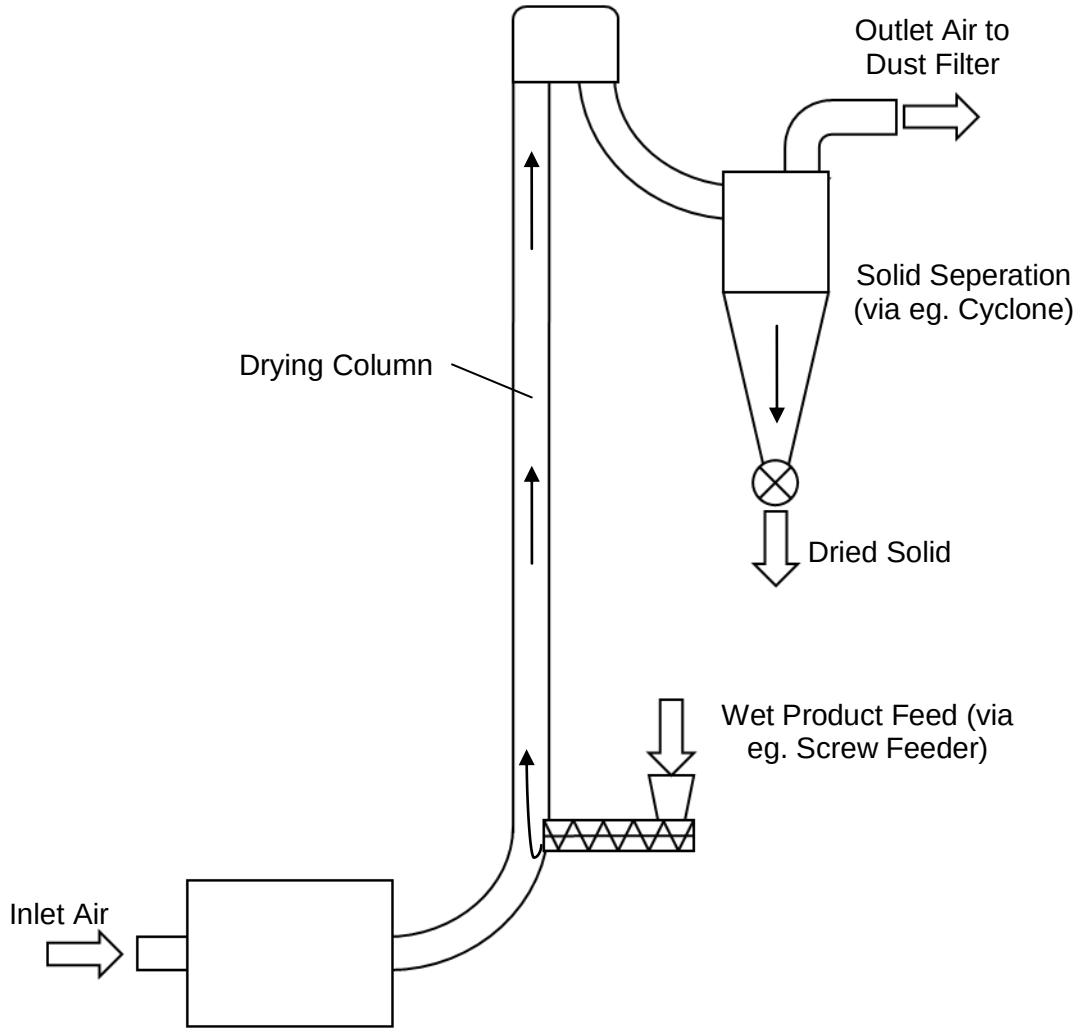


Figure 8: Diagrammatic depiction of a flash (pneumatic) dryer (after Korn, 2001)

Flash dryers are focused on removing the unbound surface moisture of the solid with some initial removal of bound moisture possible, but not always incorporated. There are fluid bed dryers which can remove bound moisture, but the residence time and the solid temperature increases drastically with the decreasing drying rate as indicated earlier in Figure 7. The residence time of the feed in a flash dryer is round 1 to 3 seconds, hence the name *flash*. This short residence time, relatively cheap capital costs and efficient design, makes it a very popular dryer technically and economically.

Advantages of flash drying include effective solids mixing, ensuring a uniform temperature and moisture spread among the various particles. The aeration of the solid bed allows a large surface area to be exposed to heated air, same as in the fluid bed dryer

(Korn, 2001; Perry & Green, 1997).

2.2 Drying Process Control

The focus of dryer control research has shifted from regulatory PID control (Dufour, 2006) to model based control approaches with the ability to predict and optimise control outputs (Dufour et al., 2003; De Temmerman et al., 2009; Didriksen, 2002; Liu & Bakker-Arkema, 2001; Abudkhalifeh et al., 2005). The need for predictive ability and optimal control is prominent in dryer processes as the contribution made by a small optimisation based on disturbances will result in significant changes in energy consumption and product quality (Dufour, 2006). PID control may be enough for regulatory control in drying, but the influence of external disturbances has such a great effect on drying process outputs, together with the dominant time delays and nonlinearities of the process that closed loop, optimising predictive control strategies are needed to ensure product quality.

2.2.1 Classification of Process Variables Applied in Dryer Control

In this section it is important to differentiate between the dryer and heat source control. The heat source controller controls a burner, or other type of heat generation unit, by typically manipulating fuel source; whereas the dryer controller controls product moisture by manipulating feed and the required heat. This may vary in different dryer setups and under different control strategies.

2.2.1.1 Measured Variables and Sensors Available

Effective drying control is defined as having the ability to dry a product to desired moisture within a limited variance despite changes in the feed moisture and feed mass flow rate, together with a variation in inlet air humidity and inlet air temperature (Abdel-Jabbar et al., 2002). To measure and control according to this goal, five real time values are needed.

There is however a level of concern in dryer control as the anecdote “you cannot control

what you cannot measure" rings true. Sensors are the main shortfall of drying process control. The problem here lies in both the cost and availability of the needed sensors together with the harsh conditions in which these sensors should operate (Arjona et al., 2005; Young, 2008).

The following table lists the most common variables in the drying process and a view on the ease of measurement of each variable based on the flash dryer studied in this research. The distinction between the control, manipulated and load variables are developed based on common literature, but may vary according to the dryer setup and control strategy developed.

Table 2: Common manipulated, control and disturbance variables and the ease of measurement of each.

Variable	Readily Measured	Control Variable	Manipulated Variable	Disturbance
Solid Feed Inflow Moisture	NO			✓
Solid Feed Inlet Feed rate	YES		✓	
Air Inflow Humidity	NO			✓
Air Inflow Temperature	YES		✓	✓
Air Inflow Flow rate	YES		✓	
Fuel Consumption/Heating Rate	YES		✓	
Solid Feed Outflow Moisture	NO	✓		
Solid Feed Outlet Temperature	NO	✓		
Air Outflow Humidity	NO	✓		
Air Outflow Temperature	YES	✓		

The difficulty of measuring solids moisture and air humidity in real time hampers controllability of the drying process (Arjona et al., 2005). Solid product moisture is the main product quality measurement and thus the ultimate control variable; whereas measurement of feed moisture and inlet air humidity can serve as measuring the disturbances and promote better control.

Air humidity can be measured by means of wet and dry bulb temperatures (Arjona et al., 2005; Holmberg & Athila, 2006), but the sensors used for this are either inaccurate or have a short life span due to the harsh conditions present in a dryer. Capacitive measurement of product moisture is used in a corn drying process (Trelea et al., 1997), but this measurement generates around 13.3% noise variance around the real value. Air humidity is rarely measured in real time.

Various researchers have simulated the control of a first principle dryer model with the assumption that solids moisture or air humidity variables can be used in control (Abdel-Jabbar et al., 2002; Abudkhalifeh et al., 2005; De Temmerman et al., 2009; Duchesne et al., 1997; Holmberg & Athila, 2006; Liu & Bakker-Arkema, 2001). The research makes no mention of the fact that these variables are difficult to measure (Arjona et al., 2005).

It is common in industry (Arjona et al., 2005), and also in literature (Hjalmarsson et al., 1996), to use the dryer outlet air temperature as an indication of drying efficiency and thus product moisture. Although this link is stated as highly nonlinear and dependant on many variables, it is the only feasible path to follow currently and is adopted by this research case study as well.

The drying process will be modelled using the available data streams with the knowledge that there will be lost dynamics due to some fundamental measurements being unavailable.

2.2.1.2 Control, Manipulated and Load Variables

Various different drying processes make use of different heat sources, different drying mechanisms and thus have different process variables. The common process variables for a continuous dryer process making use of direct heating of the feed, by a separately heated secondary air flow, are depicted in Table 2. The classification of each variable as a manipulated, control or load variable is also indicated in the table.

2.2.2 The Importance and Benefits of Dryer Control

The aim of any industrial drying operation is to produce product at a desired quality at maximum throughput, but at minimum cost (Holmberg & Athila, 2006). The main expenses

linked to drying processes is not so much in the initial capital costs but rather in the daily running of the process (Dufour, 2006), it is thus important to manage both the energy usage, product quality and overall efficiency to increase yield and lower costs.

Drying operations are responsible for 10-25% of energy usage in the developed world and dryer energy efficiency ranging between, a dismal 10%, to an average 60% (Dufour, 2006). These figures, together with the greener processes required by society and enforced by stricter emission regulations (Mujumdar & Huang, 2007), indicate that a drying process, amongst other processes, should make every energy unit count and every emission worthwhile. This means increased throughput and increased energy efficiency. The energy intensity of the drying process is repeatedly stated as a good enough reason for improving the control methods of dryers. Instead of heating to maximum temperatures to ensure maximum drying, steady control will cut energy costs and environmental impacts (Abudkhalifeh et al., 2005; Mujumdar, 2004).

Dufour (2006) assigns savings in energy consumption, maintenance costs and increased yield, due to less off spec product and faster drying times, to better control. In four separate cases - two grain dryers, a beet sugar dryer and a rotary dryer - optimised control strategies have decreased energy consumption by between 1.2% - 15% and cost by average 1.3% , and increased product throughput in two of the reported cases by 0.86% and 1.4%. Furthermore, in the case of the beet dryer, the off spec product decreased from 11% to 4% and downstream energy costs decreased by £14000/annum. The return on investment of a model based predictive controller in the sugar beet dryer and a PI controller in the rotary dryer case was 17 and 9 months respectively. Dufour further states that the expected payback time for a complex control system implementation including, first principle modelling and software development and roll out, at a dryer complex is around 18 months. The cost of implementing a control system is mainly situated in SCADA development and creating a trusted distributed control system (DCS) network. The cost of developing and implementing advanced control and online optimisation techniques is usually small in comparison to this DCS capital expense (Perry & Green, 1997).

A further motivation for dryer control implementation rests on the fact that a dryer process unit has a significantly long lifetime and most physical dryer innovations discovered from research recently cannot be introduced unless a whole new dryer setup is commissioned (Mujumdar & Huang, 2007). The innovations in control however can be implemented more readily on the existing dryer setup than a physical change.

2.2.3 Current Control Solutions Used

Seeing as drying is a self regulating process it is convenient to manually control the process according to the philosophy “run-until-dry”. With increased conscience surrounding energy efficiency and stricter demands on product specifications, from especially the food industry, the interest in dryer control has entered the research spotlight the past 20 years. The largest portion of research in drying processes is still mainly involved in the comprehension and modelling of the drying process (Dufour, 2006; Mujumdar, 2004). As far as control research goes, classic PID based controllers are the most researched dryer control method the decade before 2006 (Dufour, 2006). Feed forward control was introduced to drying processes to overcome the long time delays in the process and was found to be superior to traditional feedback control in this aspect (Duchesne et al., 1997). The tuning of such controllers, both feedback and feed forward, was however difficult due to the drying process nonlinearity and the saturation of the actuators (Arjona et al., 2005). The ability of model based control strategies to handle these nonlinearities and long time delays introduces a new era in dryer control philosophy (Abudkhalifeh et al., 2005).

Model predictive control (MPC) is the most recent and promising control method researched in dryer processes (Dufour et al., 2003; De Temmerman et al., 2009; Didriksen, 2002; Liu & Bakker-Arkema, 2001; Abudkhalifeh et al., 2005). This is mainly due to the ability of the control strategy to handle multivariable as well as nonlinear processes with long delay times. Other favouring traits of MPC are

- the ability of optimising the model online by means of an online optimiser;
- the ability of adjusting the model;

- changing the control strategy online; and
- use of constraints on variables.

(Dufour et al., 2003; Liu & Bakker-Arkema, 2001)

The increase in energy and raw material costs, together with advancements in low cost computers, has moved the focus of control philosophies toward more efficient multivariable and nonlinear model based controllers (Abdel-Jabbar et al., 2002).

2.2.4 Models Applied in Model Based Dryer Control

A 2006 review of trends in dryer control (Dufour, 2006) states that half of all dryer models in literature are based on first principles, the other half is split into 40% black box models (these are undefined in the literature under discussion) and 10% based on neural networks, fuzzy logic or no model at all. In an overview study on physical drying process modelling done by Wang et al. (2007), it is stated that the first principle mathematical models for the drying process can only be used to solve specific problems, as there is still a large knowledge gap in the understanding of the fundamentals of the process as well as a lack of accurate measurements. The inner workings and dryer modelling is microscopic, whilst the overall performance is macroscopic accentuating the lack of a proper link between these two views (Huang & Mujumdar, 1992). The model based controllers researched by De Temmerman et al. (2009), Didriksen (2002), Dufour et al. (2003), Abukhalifeh et al. (2005), Holmberg and Athila (2006) and Liu and Bakker-Arkema (2001) are all based on partial differential equation models of the mass, energy and momentum conservation principles. All these research pieces were successful in controlling either experimental setups or simulations of the process.

The nonlinearity in dryer models allows model linearisation, to make well known and powerful linear control tools accessible control options (Trelea et al., 1997) (Abudkhalifeh et al., 2005). Nonlinear models can also be accommodated by making use of advanced control methods, able to handle process nonlinearities, such as MPC. The use of nonlinear models do however slow down the optimisation process as the calculation load is too much

to execute effectively. Dufour et al.(2003) made use of offline linearisation of a nonlinear model, thus using a time varying linear model for the online predictive and optimisation algorithms.

With the increased separation between specialised control algorithms and specific processes the control engineer does not necessarily have the first principle process knowledge to model a process. In the case of drying where first principles are still being investigated and where every dryer setup and material type differs, the modelling problem is even more involved. In such cases input-output process data and a robust stochastic system identification method comes in handy. Stochastic mathematical models in drying are mainly used for control and process optimisation.

Mathematical modelling is viewed as a relatively easy method to innovate the drying process to enable not only more cost efficient and better understood drying technologies, but more ecologically friendly and better controlled dryers. The modelling of dryers from fundamental principles to understand the phenomena involved in drying is however necessary, but for control purposes the data driven mathematical modelling can suffice in the regions the model was built for.

Duchesne et al. (1997) compared steady state neural network, dynamic neural network and hybrid neural network (PI – NN) control schemes for the reason of including dryer nonlinearities in the control scheme. Huang and Mujumdar (1992) made use of nonlinear steady state neural networks for dryer modelling. These nonlinear models enable the inclusion of process nonlinearities in the control strategy. The neural network is however a black box model in the true sense of the word that, unless an expert investigates the model internally, only the model inputs and model outputs are visible. Abdel-Jabbar et al. (2002), Trelea et al. (1997) and Arjona et al. (2005) also applied models identified from data by means of system identification methods to various control strategies. It should be noted that the models are normally in discrete form to enable digital control (Abudkhalifeh et al., 2005).

The use of system identified models in advanced control is justified, but the decision of which identification method to use remains a preference based on the needs of the identification problem at hand.

2.2.5 Dryer Control in the Industry

Dryer control research has a very uneven focus among industries. 66% of literature discussing dryer control is focused on the food industry with the first publication dating back to 1983. Painting (8.5%), pharmaceuticals (6.8%), paper (6.8%) and wood (5.1%) industries enjoy the middle tier of attention with the first publications just before and around the turn of the millennium. The mineral and textile industries are at the bottom of the spread with 1.7% of process control research attention going to each (Dufour, 2006).

Duchesne et al. (1997) made use of a feed forward PI controller in a virtual ore drying process simulation study. A decentralised PID controller was developed and implemented in the food industry at a live olive waste dryer plant (Arjona et al., 2005).

Model based controllers have been applied to virtual dryers modelled in the food industry, specifically in the grain drying (Liu & Bakker-Arkema, 2001), pasta drying (De Temmerman et al., 2009) and sugar beet drying (Didriksen, 2002). Furthermore model based controllers are found to be used in an experimental setup of a water based solvent extraction in a paint drying process (Dufour et al., 2003) and is discussed as a possible solution for the bio fuel (tree bark) drying control in the paper and pulp industry (Holmberg & Athila, 2006).

Below is a summary of all the research found for dryer modelling and control, as mentioned in the literature review.

Table 3: Summary of dryer modelling and control as found in literature and discussed in this section.

Process	Control Strategy	Model Used	Reference
Infrared Drying of Paint	Model Predictive Control	Time Varying Linear Model based on Partial Differential Equations	Dufour et al., 2003
Pasta Dryer	Model Predictive Control	First Principle Differential Equations	De Temmerman et al., 2009
Sugar Beet Drying	Model Predictive Control	First Principle Differential Equations	Didriksen, 2002
Grain Dryer	Model Predictive Control	First Principle Differential Equations	Liu & Bakker-Arkema, 2001
Electric Infrared Dryer for Fibre Sheet Drying	Model Predictive Control	Linearised First Principle Differential Equations	Abudkhalifeh et al., 2005
Bio Fuel (tree bark) dryer in the paper and pulp industry	Model Based Control	First Principle Differential Equations	Holmberg & Athila, 2006
Mixed Flow Corn Dryer	PI and LQG Control (Linear-Quadratic Gaussian)	Linearised Model identified from input-output data	Trelea et al., 1997
Rotary Dryer for Ore drying	Feed forward PI control	Steady state-, dynamic and hybrid neural networks	Duchesne et al., 1997
NA	NA	Steady state nonlinear neural network	(Huang & Mujumdar, 1992)
Continuous Fluidised Bed Dryers	Internal Model Control; Model Predictive Control	Data Driven Modelling; Linear State Space and Transfer Function Models	Abdel-Jabbar et al., 2002
Olive Waste Dryer	Decentralised PID	Data Driven Modelling	Arjona et al., 2005

2.3 Conclusions Drawn from the Literature Review

From the literature review it is clear that a number of phenomena is involved in the drying process increasing the effort required for modelling the dryer process. This required first principle knowledge and understanding, together with the variations in dryer setups, results in a much involved modelling process. It is stated that the combination of input-output data, together with a robust system identification technique overcomes some of these issues. That said, it was also found that not all the process variables in drying are readily measured, possibly influencing the success of such a data driven system identification approach.

The business case for dryer control is largely built on energy efficiency and limiting emissions, with product moisture playing a larger role in the regulated industries such as the food industry. Product quality is measured as the moisture content in the exit product. This moisture is not measured online, but it is common to make use of exit air temperature as an indication of drying efficiency. This approach is adopted in this research as well.

The case for MPC is strengthened in literature by the number of research pieces in the field, and the statements that MPC is capable of handling the time delays and nonlinearities contained in drying dynamics. Most of these MPC however make use of linear or linearised models for prediction and control move optimisation. It will be the aim of this research to investigate implementation of a nonlinear model in a basic MPC algorithm to find possible shortcomings or future research requirements. This will approach will require rigorous system identification.

Chapter 3 Characteristics of the Drying Operation Investigated

This section sets out the make-up of the specific drying process investigated. Furthermore it provides insight into the behaviour of the process concluding with the reason for focussing on control as a solution and the control strategy currently followed.

3.1 Concentrate Dryer and Smelting Operations

The focus of research is on a dryer process included in the smelting leg of the PGM value chain as discussed previously (Chapter 1). The smelter process uses wet feed from the concentrators and supplies matte to the base metals refinery, increasing the PGM concentration fifteen fold from about 300 grams per ton of concentrate to an estimated 5000 grams per ton of matte. The drying plant is a sub operation of the smelting plant. It is responsible for drying concentrate, received from the concentrator plant, before it is fed to the smelters. The drying plant consists of 3 filter presses responsible for primary removal of water and 2 dryers responsible for final drying of concentrate. Two of the filters operate at 24t/hr and the other one at 18t/hr. Operations between the filters are alternated (CSense Pty Ltd, 2007). The dryers entail a spray dryer, used for feed going to the Pyromettalurgical furnaces or Merensky furnace, and a flash dryer for feed going to the 28MW main furnace. The focus of this study will be on the flash dryer section of the drying plant depicted in Figure 9. The location of variable measurements and operating points are also included in the diagram.

The following variables are important for further reference and are marked in Figure 9:

a. Coal Conveyor Feed Rate

b. Fluidising Damper Valve

c. Hot Gas Generator Secondary Air Temperature

d. Filter Cake Feeder Rate

e. Flash Dryer Output Air Temperature

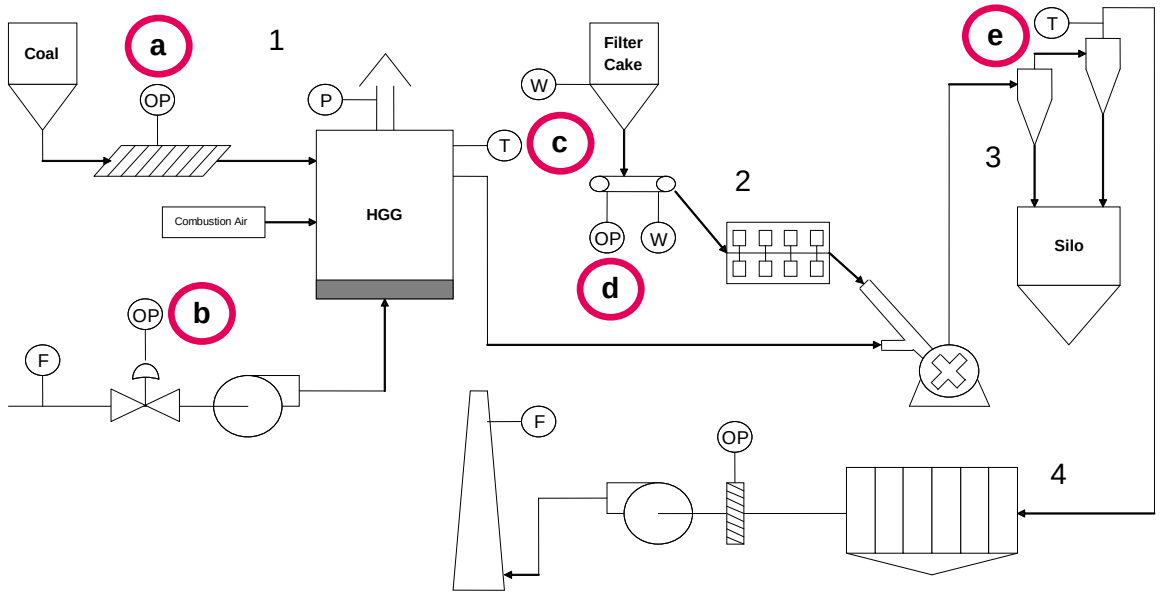


Figure 9: The flash dryer drying plant consisting of the (1) HGG, (2) Feeder, (3) Flash Dryer and (4) the bag house filters. Control points (OP) and measured variables (P-pressure, W-weight, T-temperature and F-flow rate) are indicated. (CSense Pty Ltd, 2007) (Included with permission)

The flash dryer dries moist concentrate obtained from the filters to around 10% moisture before feeding it to the main furnace. A 10 ton concentrate filter cake bin is situated between the flash dryer and the filters serving as a process buffer. There is no official buffer between the flash dryer and the furnace; also there is no communication between the filters and the flash dryer, neither is there any communication between the flash dryer and the furnace. The drying plant consists of the following, numbered correspondingly in Figure 9:

1. hot gas generator;

2. filter cake feeder;

3. flash dryer with cyclone separators; and

4. filtering stage in the bag house before the heated air is released into the atmosphere.

(CSense Pty Ltd, 2007)

These sections will now be discussed in detail individually.

3.1.1 Hot Gas Generator (HGG)

The HGG fuel source is coal which is fed into the burner. Combustion air is added to the burner and the emission stack of the burner is damped. Secondary (heating) air is taken from the atmosphere and heated indirectly in the burner. Secondary airflow into the HGG is controlled by the fluidising damper valve situated before the HGG. It is called the fluidising damper as this airflow influences the fluidisation of particles in the flash dryer further down in the process. The coal feeder conveyor is controlled by a PID feedback loop based on the outlet air following a set point determined by the operator. This PID controller can only set the conveyor as on or off, resulting in what is known as bang-bang control. Outlet secondary air is fed to the flash dryer for the drying process.

3.1.2 Flash Dryer Feeder (FD Feeder)

The FD feeder consists of a conveyor feeding moist filter cake from the filter cake bin to the bottom of the flash dryer where it is mixed with the heated secondary air from the HGG. There is no communication between the filters, upstream from the buffer bin, and the flash dryer feed.

The feeder is stopped on 2 occasions:

1. When the flash dryer exit air temperature drops below the efficient drying temperature boundary of 94°C; and
2. When the buffer bin runs empty.

3.1.3 Flash Dryer (FD)

The flash dryer consists of a vertical cylinder in which the heated air and moist concentrate is mixed and transported upwards to a cyclone separator. It is during this transportation and

aeration where the concentrate is dried. The cyclones separate the concentrate from the air and drop it into a concentrate silo. The air is sent to the bag house for filtering.

3.1.4 Bag House

The bag house filters the air before emitting it into the atmosphere. A damper valve is situated after the bag house which can control the air flow. The exit air flow rate is measured at the stack before releasing the cleaned air to the atmosphere.

The operation of the bag house is assumed as not important for this research.

3.2 Problem Statement for the Dryer Control Solution

Through an ongoing Six Sigma initiative, it was found that the drying plant is a bottleneck in the production operations. The throughput was being limited by:

- Large oscillations in the HGG output temperature causing FD downtime and feed stoppages, as well as temperature spikes to above 200°C causing equipment damage; and
- Lack of synchronisation between the filter plant and flash dryer feeder causing the feed bin to run empty and a resulting concertina effect in concentrate availability and operations;

Feed Stoppages are firstly due to the temperature interlock which shuts off concentrate feed when the flash dryer output temperature goes below 94°C. Secondly stoppages are caused by the feed bin running empty.

3.2.1 Feed Stoppages due to Temperature Interlocks

The following trend, Figure 10, indicates the problem which arises when the temperature interlock is activated. The interlock activates at points indicated ‘A’, ‘B’ and ‘C’ in the figure. The flash dryer output temperature recovers quickly, but within 5 minutes from when the interlock is activated, the output temperature spikes above 200°C. This is above the recommended operating temperature for the drying equipment, especially the bag house filters and can thus shorten the equipment life span. At the same time the hot gas generator

output temperature is not synchronised with the activation of the interlock as there is no communication. Not only can this assist in the temperature spike in the flash dryer, but coal is being fed and wasted when the operations are essentially off.

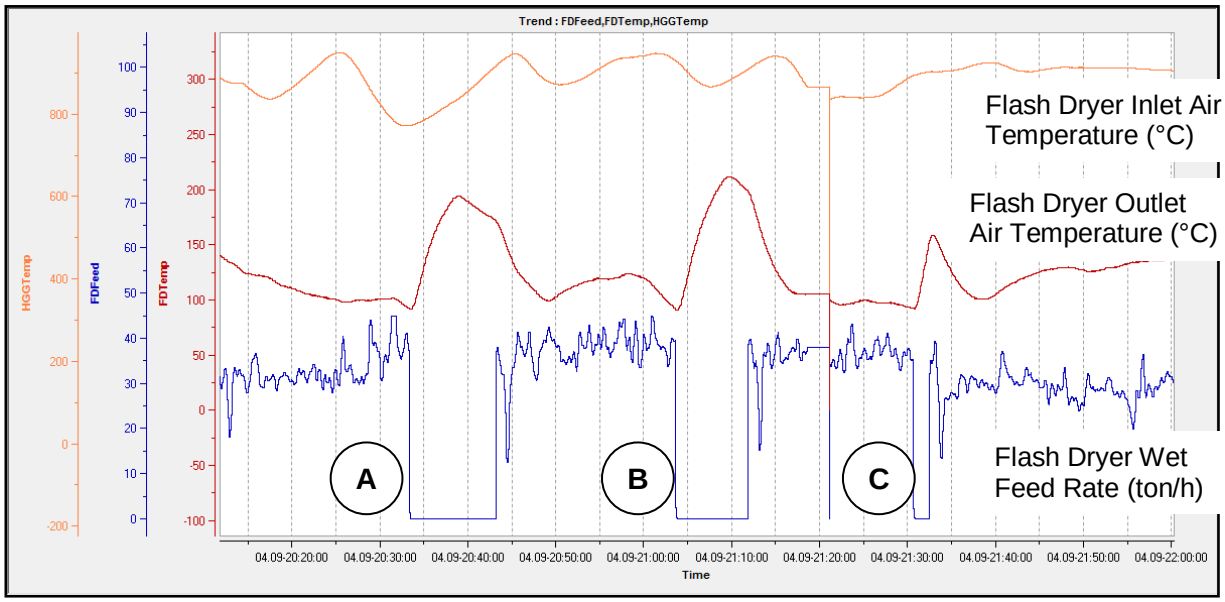


Figure 10: The interlock stops the concentrate feed and causes the temperature to spike to above 200°C damaging the equipment, especially the bag house. The hot gas generator operation is not linked to this interlock becoming active.

3.2.2 Feed Stoppages due to Bin Empties

A secondary influence on concentrate feed stoppages is the availability of concentrate. There is, at the time of this project, no communication between the filters forgoing the flash dryer in the process, and the flash dryer operations. The communication exists only when the operator finds the concentrate bin is running empty and contacts the filter operators to find why this is happening. In Figure 11 it can be seen where the operator stops the feed and waits for the bin to receive more filter cake concentrate.

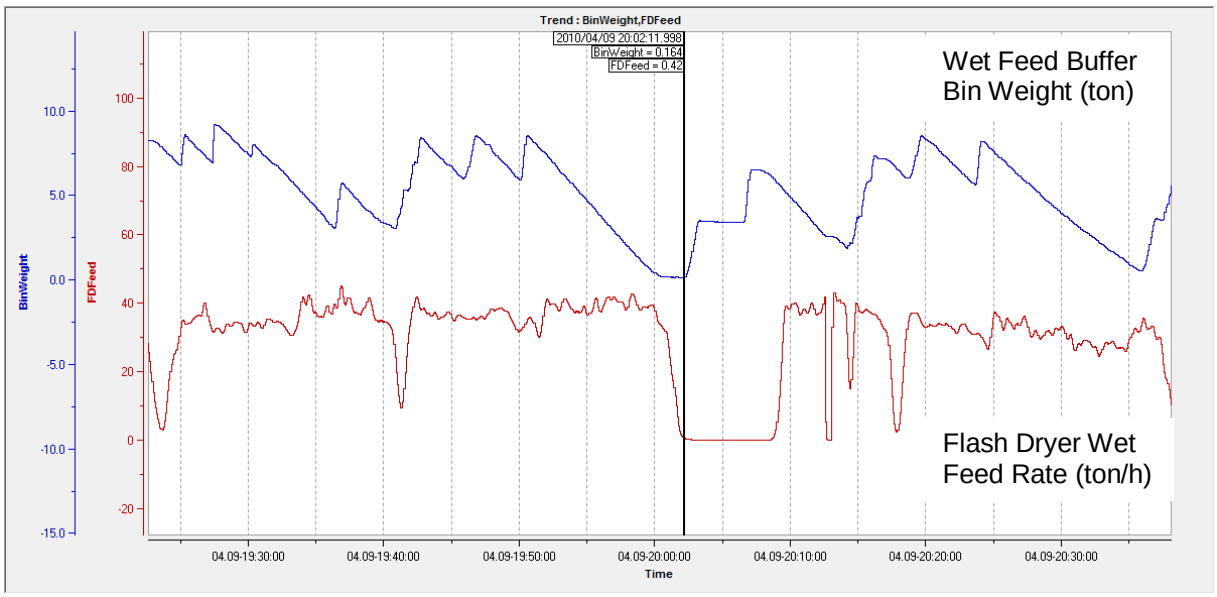


Figure 11: Lack of communication between the filter section and the flash dryer operation is causing feed stoppages due to filter cake shortage.

3.2.3 Hot Gas Generator Oscillations

Oscillations in the hot gas generator output temperature is experienced every 25 to 30 minutes. This is due to the current on-off control strategy followed in controlling the coal feeder as can be seen in the trend, Figure 12. These oscillations are passed on to the flash dryer and it is expected that these contribute to variation in the flash dryer operations.

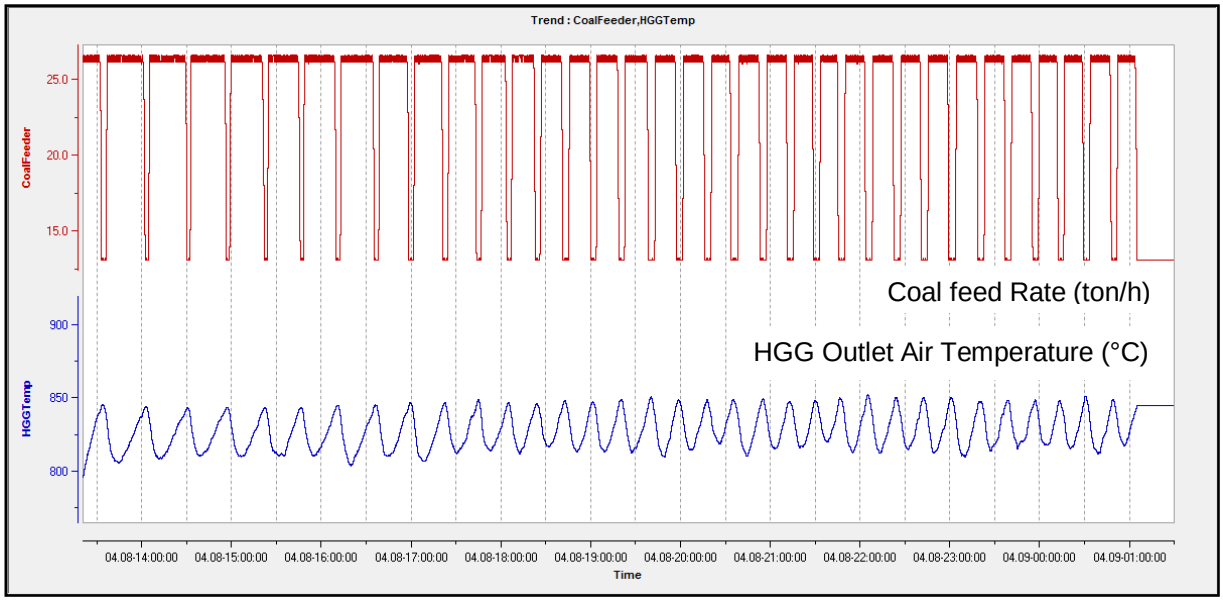


Figure 12: The on-off ("bang-bang") control approach followed for the hot gas generator is causing oscillations in the output temperature. These oscillations are passed on to the flash dryer operations.

A closer look at the dynamics of the coal feed and the output temperature in Figure 13, indicates that the process output continues to decrease and then, around 8 minutes after the step change in the coal feed, the output turns around and starts to increase. This transition in the output temperature during this delay period is smooth compared to the coal feed's sudden change.

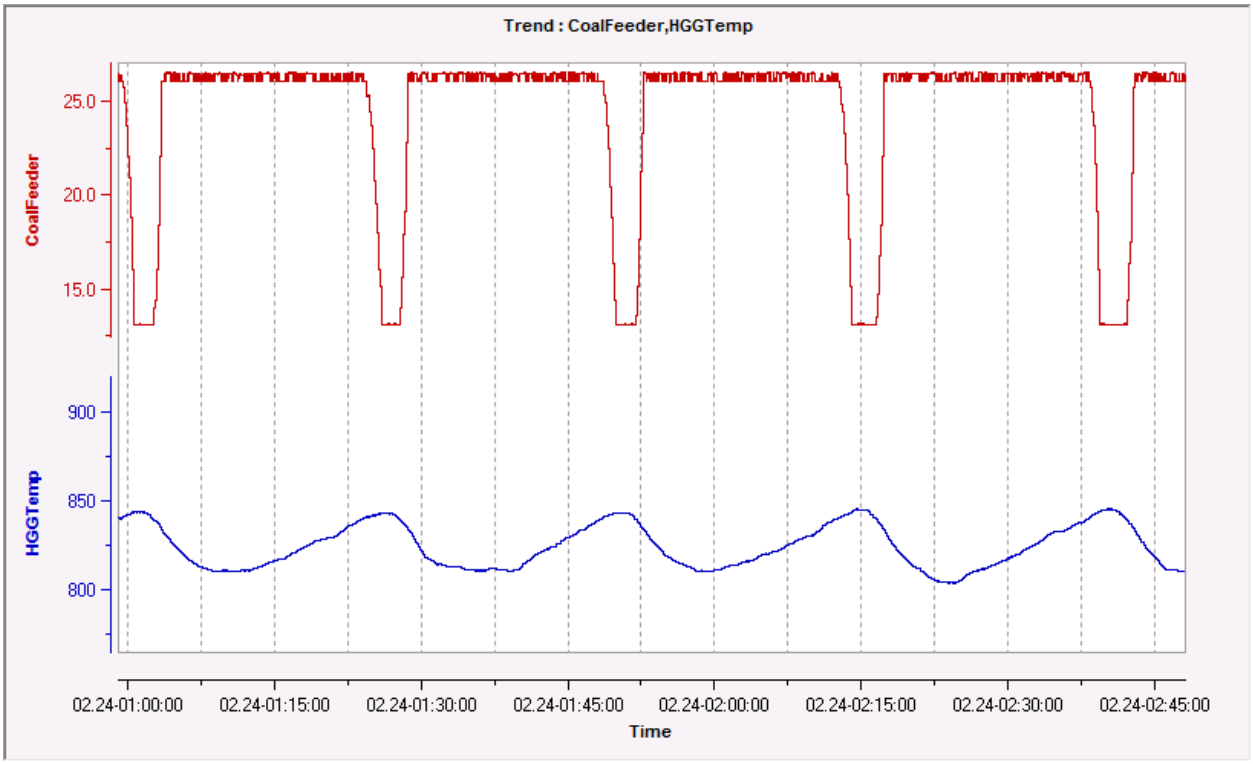


Figure 13: A closer look at the on-off ("bang-bang") control approach indicates large process lags, with smooth transitions in the ouput temperature, compared to the stepped input.

3.3 Control Strategy as a Problem Solution

These aforementioned phenomena were assigned to poor regulatory control (SAIMC, 2008; CSense Pty Ltd, 2007). It is thus an obvious choice to consider a different, or complimenting, control strategy. The aim of such a solution will be, based on the specific problems identified in the previous section:

- Prevent flash dryer outlet temperature spikes through synchronisation between hot gas generator output and flash dryer operations;
- Maximise output of filter cake by preventing feed stoppages by pre-emptive actions to signs of the feed bin running empty, as well as keeping the flash dryer temperature above the interlock temperature; and

- Smooth the hot gas generator output temperature, decreasing oscillations, by replacing the “bang-bang” control approach.

Note that an advanced process control strategy has been designed and commissioned for the flash dryer operation in April 2007. This implementation sparked industry interest in dryer control alternatives as well as identification of possible hurdles to control, and resulted in this research piece investigating model predictive control as an alternative control strategy. It is necessary to understand the implemented APC for the dataset filtering steps, seeing as data for system identification in this project was collected when these APC’s were running; as well as to interpret the comparison between the existing APC and the proposed MPC at the end of this investigation.

It is the objective of this study to further investigate possible modelling and control strategies from literature which can be applied by process control specialists to the abovementioned drying plant. The focus will not be on streamlining material flow between the filter section and the drying section. This problem will be omitted from the research and assumed as not hampering operations.

The advanced process controller currently active on the plant, and also active during data collection, is divided into 3 sections.

1. HGG IMC: The first is an internal model controller (IMC) implemented at the HGG to control the outlet air temperature by manipulating the coal feed.
2. FD PI controller: The second is a feedback-feed forward PI controller controlling the feed bin weight by adjusting the feed set point.
3. Fluid Damper Rules Based controller: The third controls the fluidising damper and thus the inlet air flow rate by means of if-then rule sets. This controller also controls the maximum and minimum allotted ranges for the feed rate set point and thus limits the FF PI controller.

(CSense Pty Ltd, 2007)

The dataflow for the solution is depicted in Figure 14.

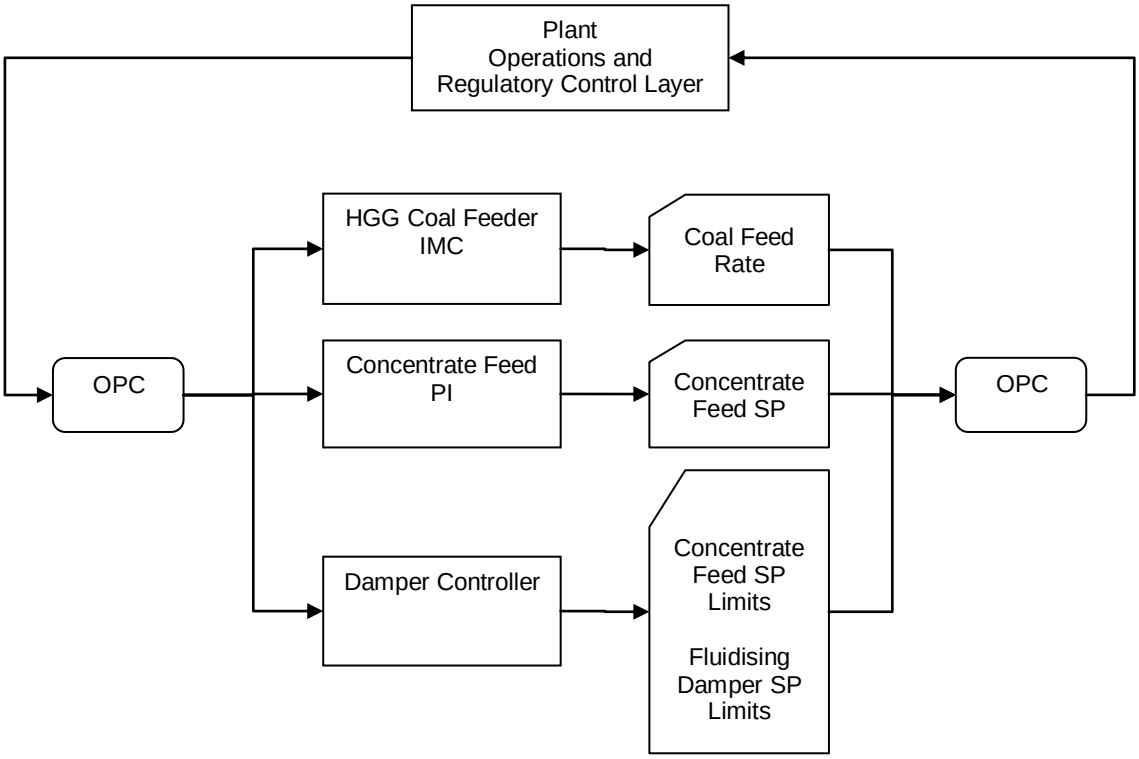


Figure 14: Data Flow of the Current Live Drying Operation APC

This APC is situated on top of the process control network and provides set points to the regulatory control layer. The PLC layer of control for both the coal and concentrate feed is a weak controller. Influences of these closed loop controllers will be neglected during system identification (Van Schalkwyk, 2009).

3.4 Conclusion

The current control strategy is running and is controlling the process according to plant opinion. This will be briefly investigated later when comparing the MPC results to the current live plant control. Given the success of the current controller, it is proven that this dryer can be controlled. It is thus a good base for investigating system identification of a nonlinear model for the various drying operations and attempting a MPC with the best models.

SYSTEM IDENTIFICATION AND MODEL BASED CONTROLLER RESEARCH METHODOLOGIES

Chapter 4 Data Preparation and Analysis as per the Dryer Control Strategy Requirements

4.1 Methodology Overview

This research entails two main sections:

1. System identification (SID) of drying circuit models by means of genetic programming (GP); and
2. Model based predictive control (MPC) of the drying circuit and identification of best control possibilities.

Before the discussion surrounding dataset preparation and modelling commences, it is necessary to understand the opted control strategy and the required models. This will put the handling of the data before and during system identification into context. After this the section continues with the data preparation and analysis; choice of timeseries to use by means of surrogate data comparison; and ending off with the construction of the latent variable set.

The genetic programming system identification discussion and method followed for constructing and simulating the model based predictive control is handled in the next two chapters individually.

The following diagram of the detailed steps followed in the methodology, used for this case study, is included as a guide throughout the rest of this section.

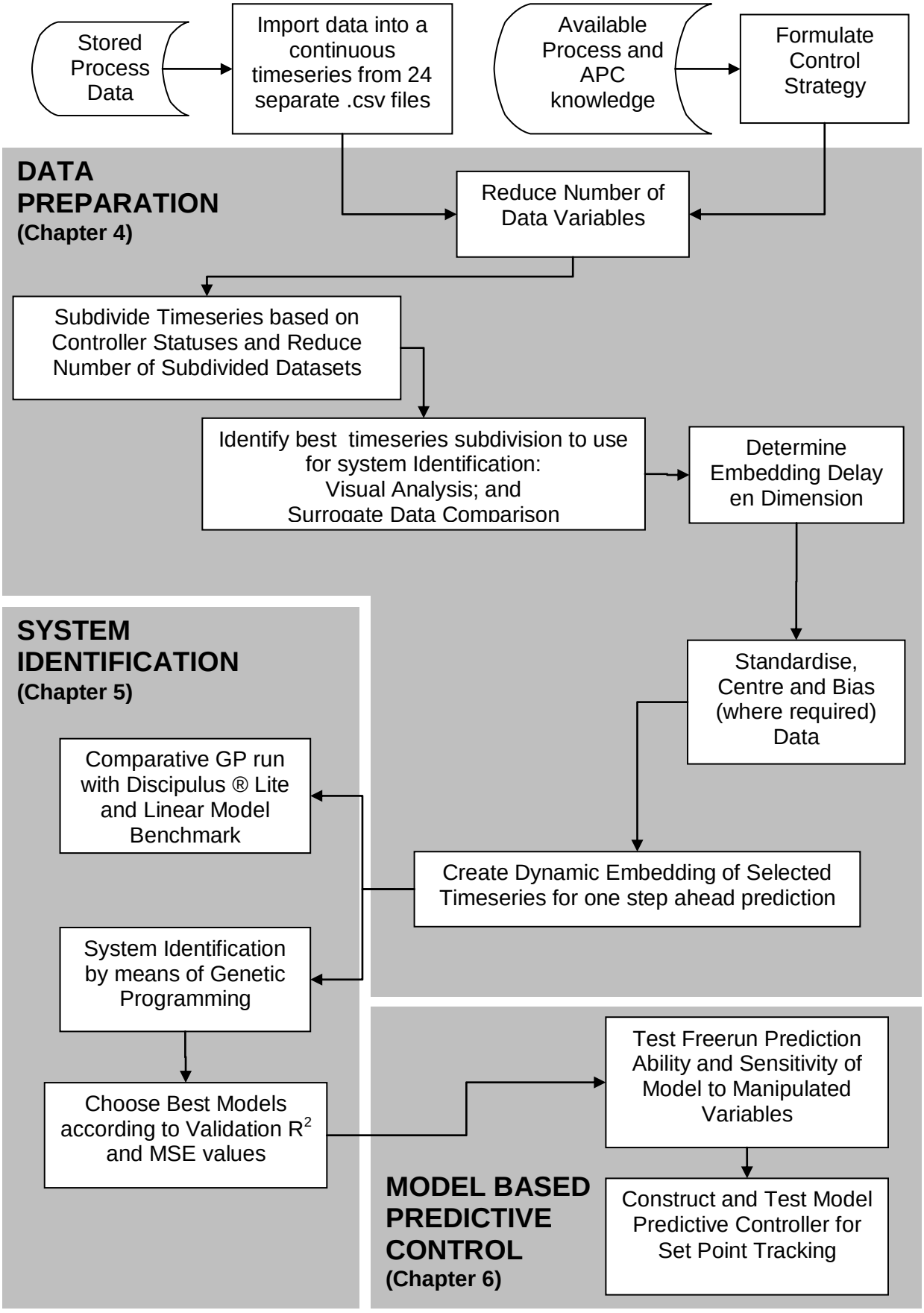


Figure 15: Data Preparation, System Identification Methodology Steps followed to allow the planned Model-Based Controller Strategy to be constructed and tested for the Flash Dryer operations.

4.2 Proposed Control Solution

4.2.1 Control Philosophy

Obtaining desired moisture levels is difficult to monitor in real time due to lack of sensors. The aim will be to control the flash dryer output temperature to follow a desired set point and be within a specified control band. The assumption, used in literature (Hjalmarsson et al., 1996) and practice (Arjona et al., 2005), is that the moisture content in the solid follows the outlet air temperature. Although it is stated in the same literature that the correlation could be very nonlinear and dependant on multiple variables, a strong correlation is assumed for the rest of this research. The flash dryer control philosophy will focus on controlling the air temperature without any focus on moisture levels of either the exit air or the material feed in and out.

In the case of secondary air heating, i.e. drying air, it is common to decouple the control of the heat generator from the operations of the flash dryer. This approach is also assumed applicable, as the controllers investigated for both these two sections of the drying operations, are decoupled in this research.

4.2.2 Control Strategy

The flash dryer (FD) exit air temperature, hence product moisture, is influenced by both the inlet air temperature, received from the hot gas generator (HGG), and the conditions in the flash dryer. Both of these processes need to be considered in a dryer control strategy. Two possible control strategies, in this research, are indicated in the following table.

Table 4: Two possible control strategies for the concentrate drying process

Control Strategy	Manipulated Variables	Control Variables
Flash Dryer	Concentrate Feed HGG Outlet Air temperature	FD Outlet Air Temperature
Hot Gas Generator	Coal Feed Fluid Damper %	HGG Outlet Air Temperature

It should be noted that the investigation will assume that the HGG outlet air temperature can be manipulated by a flash dryer controller. In practice this will be done by means of a set point change for the hot gas generator controller. The lag of the HGG controller is neglected with the aim of investigating the impact of the HGG outlet air temperature as manipulated variable.

Some specifics regarding these two strategies will briefly be discussed in the same order as in the table.

4.2.2.1 Flash Dryer Controller

The FD controller will consist of a model with concentrate feed and HGG outlet air temperature as input variables and the FD outlet air temperature as output variable. The FD outlet air temperature set point is managed manually and will be assumed to stay at 140°C. The area of focus is depicted in the following figure.

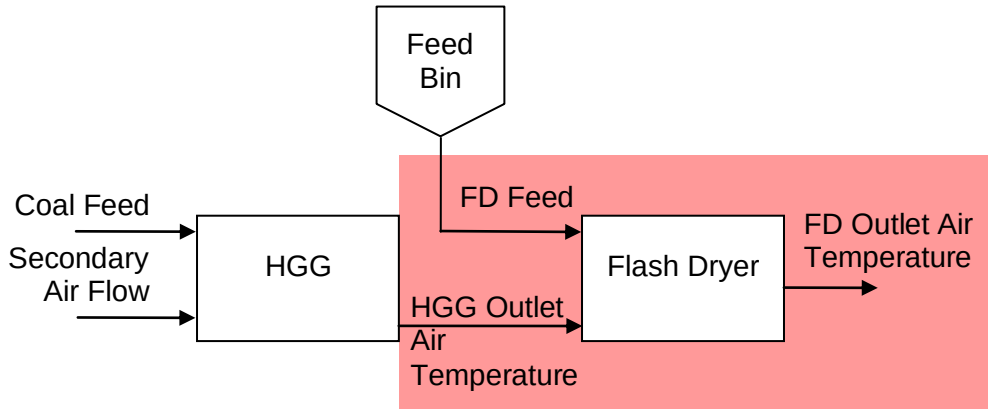


Figure 16: Flash dryer feed controller focus area as part of the whole process

The controller implemented will be a model predictive controller.

Assumption for the FD controller:

- The controller will not take into account the bin level. The source of wet concentrate feed is assumed inexhaustible for this research;

4.2.2.2 Hot Gas Generator Control

Seeing as the focus point of the current APC solution is to control the HGG outlet air temperature, it is decided to investigate control of the same section of the drying plant by means of MPC. The HGG outlet air temperature set point is managed by an external source, either an operator or another controller. The process inputs include the secondary airflow rate influenced by the fluidising damper, as well as the burning coal feed rate. The focus area of this controller is illustrated in the diagram.

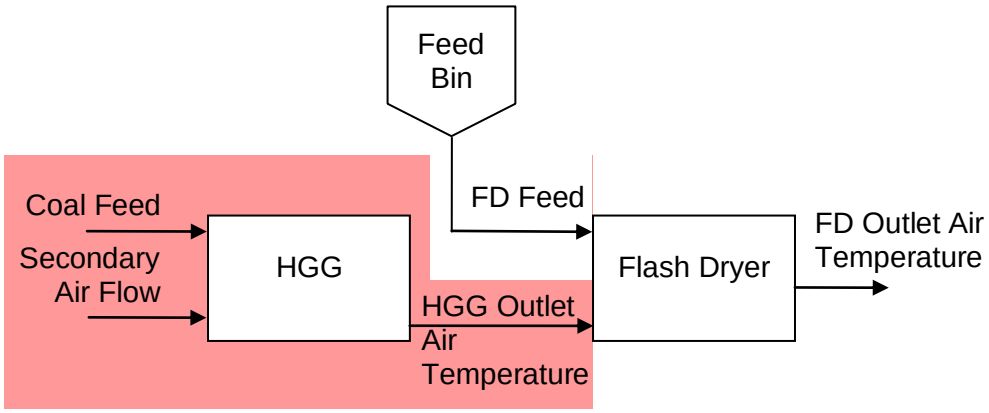


Figure 17: Hot gas generator (HGG) controller focus area as part of the whole process

A MPC will be developed for this controller based on the identified model. The secondary air flow variable is found from theory to have minimal effect on the outlet air temperature. The decision to include or omit this variable will however be left to the genetic programming system identification procedure. The following assumption is made regarding the hot gas generator controller:

- The coal source is not constrained;

4.3 Dataset Preparation and Analysis

It is logical that the data preparation exercise for raw data obtained from the historian will need a great amount of attention, pre-processing and analysis. This section explains the methodology followed to separate the data into subsets according to the models which need to be generated. Furthermore, it explains how the best dataset, of the various created subsets, is chosen for system identification through an analysis of the dynamic information

available in the dataset, by means of surrogate data analysis. The construction of a latent variable input regressor set from the timeseries, concludes the section.

4.3.1 Dataset Background

Seeing as it was not possible to do step testing on the plant and there is no model available to simulate step tests on, the available plant datasets, located on the plant historian, were used for system identification and process simulation.

It is clear that the datasets obtained in such a manner need to be audited and reviewed to ensure process data validity and to remove parts of the timeseries where process anomalies and seemingly out-of-the-ordinary process events occurred. Such events are common on a plant and this “plant misbehaviour” information will be stored in the historian dataset.

To further complicate the data preparation, datasets from the plant are only available from after the implemented advanced process control (APC) strategy, as discussed in 3.3 - Control Strategy as a Problem Solution. This complicates the process dynamics in an captured in the data and requires these APC influenced sections be removed.

The dataset obtained ranges from 16-May-2009 00:00:00 to 08-June-2009 00:00:00 with 5 second intervals, i.e. 17280 data points per day. This data range was randomly selected and supplied by the industry partner. Although there is no information regarding the good operation of the process, or insuring the presence of expected dryer disturbances and dynamics, this was the data received which had to be used. All other influences of process drift, unconventional feeds, different types of ore or maintenance issues were assumed negligible until proven otherwise. This data is thus unaltered and is gathered from the CSense APC historian. This historian holds both the process variables and the APC on/off status. These on/off states are used to filter out the data when the APC solution was on, excluding the influence of the current APC solution.

It is assumed that the historian data collection settings, data compression, averaging, sample-and-hold, etc, under which the data are collected and stored in the historian during collection, have no effect on the dataset's relevance for the research.

All data manipulation was done in Matlab. The data were supplied in 24 .csv files per day for the 24 days in the time range and were imported into Matlab. Only the major variables identified as having an influence (see Classification of Process Variables Applied in Dryer Control discussed earlier) were imported. Further discussion of the reduction of the datasets is included next.

4.3.2 Data Reduction

Data reduction entails reducing the dataset by either

- decreasing the number of records by increasing the sampling time; or
- by reducing the number of process variables included.

The following process variables were identified as important for system identification from the earlier discussion regarding process variables in the dryer process (see 2.2.1). All other variables were removed from the datasets.

Table 5: Process Variables used in the system identification with units of measure and plant tag names

Process Variable	Unit of Measure	Tag Name
Concentrate Feed Bin Weight	tons	Bin_Weight
HGG Coal Feeder Rate	tons/hour	CoalFeeder_OP
Flash Dryer Concentrate Feed Rate	tons/hour	FlashDryerFeed_PV
Flash Dryer Outlet Air Temperature	°C	FlashDryer_Temp
Secondary (heated) Air Flow Valve Opening	% of maximum opening	FluidisingDamper_OP
HGG Outlet Air Temperature (same as Flash Dryer Inlet Air Temperature)	°C	HGG_Temp

4.3.3 Data Cleaning

The current APC was deployed in April 2007. The active controllers create a problem for system identification as process variable correlations are altered by the collective process and advanced controller dynamics. Modelling from a dataset such as this could capture the dynamics of the process under advanced control and not of the process alone. If the model

predictive controller was going to be applied on top of these controllers, the dynamics of the controllers could be included in the model, but seeing as the MPC is aimed at replacing these controllers, datasets incorporating these dynamics need to be either removed, or avoided. There are instances where these controllers are switched off, and the problematic combined dynamics can be avoided by only using these "controller-off" sections for SID. The "controller-on" states are marked by the following the Boolean historian tags individually:

Table 6: Live Flash Dryer APC solution and the tags which indicate if the specified controller is on or off

Controller	Tag Name
HGG internal model controller	ControllerOn_HGG
Flash Dryer Feed Set Point PI controller	ControllerOn_FDryerFeedSP
Fluidising Damper If-Then controller	ControllerOn_FluidDamper

The following figures illustrate the 24 day dataset obtained and the influence of the controllers on the process. Figure 18 illustrates the various APC statuses. From visual inspection it is clear that the HGG Coal Feed APC and Fluidising Damper APC are linked. These two controllers are in the same status 95.5% of the time of this dataset. For further discussion in this section they are assumed to be the same.

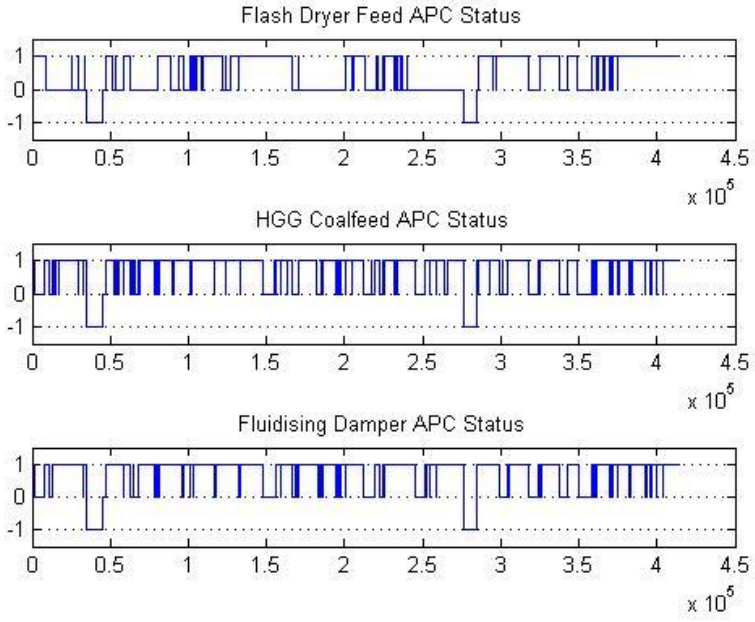


Figure 18: On (1), Off (0) or No Data (-1) Status of the Current APC Controllers. The HGG and Fluidising Damper controllers correspond 95% of the time.

Comparisons of the flash dryer and hot gas generator variables with the controller states are included in Figure 19 and Figure 20. In both figures there are two sections where no data were recorded. These are indicated by the combined drop of all variables to zero. These sections are removed from further investigation.

In Figure 19 very little can be deduced visually from the correlation between the flash dryer dynamics and the controller state. The largest controller-off states are indicated by the orange shaded areas. The controller active state will however be removed from the timeseries for safety.

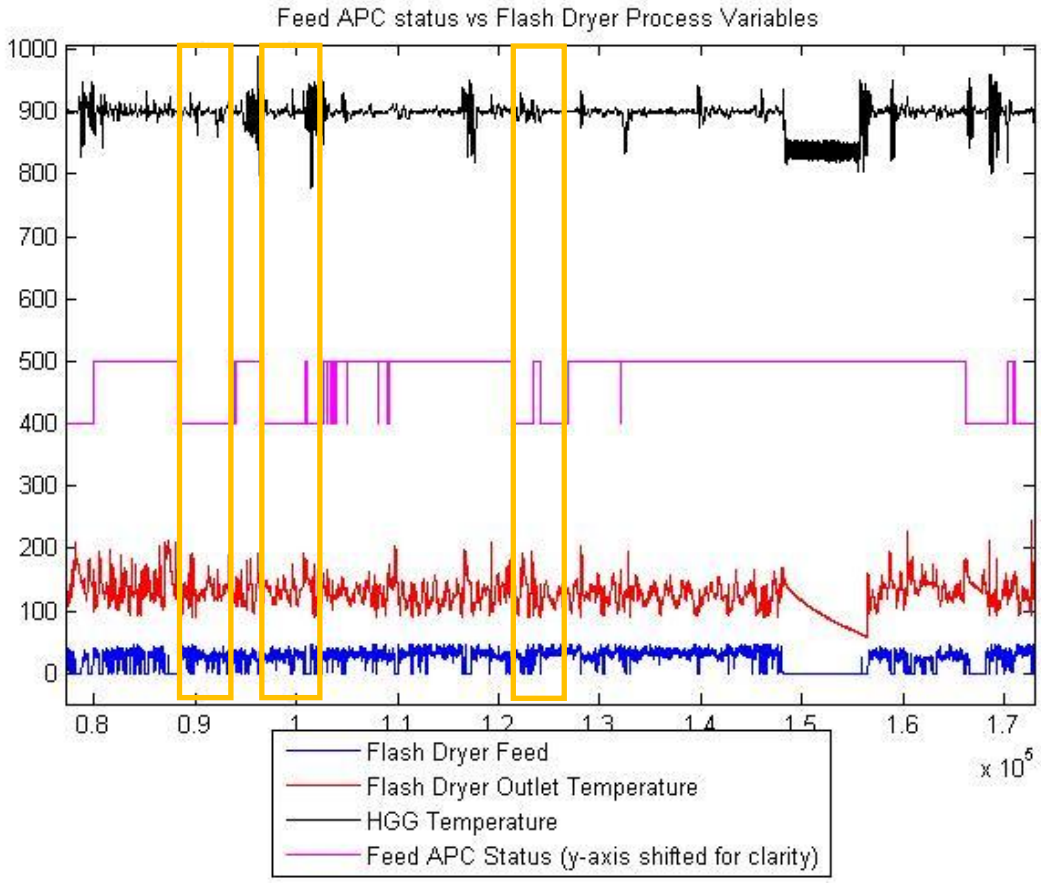


Figure 19: Flash Dryer Process Variables compared to the Feed APC. The Feed APC "off" statuses are highlighted. No real difference from the rest of the process is visually discernable.

The coal feed APC has a significant influence on the hot gas generator output temperature as can be seen in Figure 20. Orange shaded areas indicate major sections where the controller was off. Every time the controller is off a "bang-bang" controller assumes control of the coal feed; whilst the fluidising damper defaults to a value and is adjusted manually by

the operator. The hot gas generator drops from the 900°C to an average of 840°C. This is probably due to the drop in coal feed. The influence of the APC is apparent. Both the sections where the controller is off and the whole dataset will be investigated for system identification.

No plant input was supplied to whether the coal feeder motor is equipped with a variable speed drive (VSD). From the smoother action seen during the controller-on states indicate that the drive is not only on and off, but makes use of a VSD. Throughout this research it is assumed that the coal feeder motor can be manipulated as a VSD.

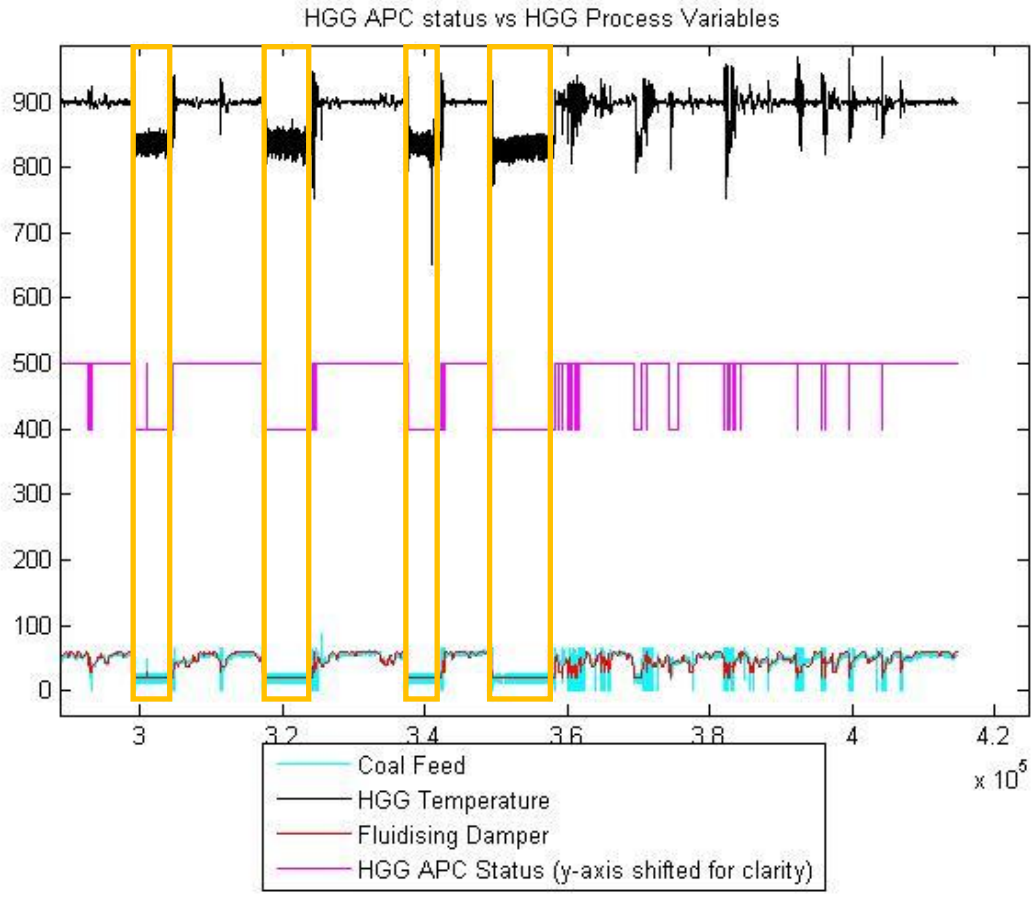


Figure 20: Hot Gas Generator Process Variables compared to the HGG APC. The HGG APC "off" statuses are highlighted. A clear difference can be seen when the controller is on and off.

The goal is to identify a nonlinear dynamic model using a latent variable reconstruction. For this a continuous timeseries is required. The sections where the controllers are off are however breaks this continuous timeseries. The sections will thus have to be isolated and

subdivided into smaller datasets. Each dataset will then be evaluated for individual SID purposes.

The decision of which controller to link to which variable is based on the understanding of the prior APC solution and drying circuit. The following table sets out the states each of the current APC controllers must be in to enable the modelling of the models discussed earlier.

Table 7: Dependencies of process sections on the current APC solution viewed to clarify the required dataset manipulation

Control Strategy	HGG internal model controller	Flash Dryer Feed Set Point PI controller	Fluidising Damper If-Then controller
Flash Dryer	NA	Off	NA
Hot Gas Generator	Off	NA	Off

The assumption is made that the influence of the controllers on the timeseries can be neglected at the point where the controller is switched off. All further delayed influences from the point where the controller is switched off are thus neglected.

The remaining subsets are initially evaluated for usability by looking at timeseries length, and visual inspection of the dynamics of the data. Advanced analysis is done by means of surrogate data comparison discussed later in Chapter 8.

Seeing as process knowledge and plant input to this project is limited, provision for specific process states and consequent dynamics are not made beforehand but will be handled as they are found.

4.3.4 Data Normalisation and Induced Bias

Although this is a general step in system identification, the discussion of data normalisation needs to be included seeing as it is such an important step in system identification. Normalisation of data refers to scaling of the variables to similar orders of magnitude to allow them to be compared. The normalisation step needs to be included in the GP algorithm or any of the modelling steps. The following formula is used:

$$x_{n_i} = \frac{(x_i - \mu)}{\sigma}$$

The mean (μ) of a timeseries for a period is subtracted from all the data points (x_i) one at a time. The result is then divided by the standard deviation (σ) of the timeseries for the same time period. The result (x_{ni}) is a timeseries with zero average and standard deviation of 1. The normalised validation sets are constructed using the statistics of the training dataset.

Furthermore, in some cases in this research, the normalised timeseries is shifted, or biased, by centering the timeseries at a value larger than zero. This is done so all values are positive. This offset is uniform across all the variables in the timeseries, and is introduced in cases where a square root in the functional set requires closure - all variables should preferably be positive to prevent imaginary numbers resulting from the root of a negative value. The specific application will be discussed at a later stage. At this point it should be noted that the bias is determined by the largest negative value across all the already standardised and centered variables. The bias is equal to the absolute of the determined value and rounded up to the nearest integer. This chosen bias value will be adopted by the controller for the model used as the model parameters is be determined based on this bias.

Means and standard deviations used for normalisation are included in Appendix A .

4.3.5 Nature of Process Dynamics in the Timeseries: Surrogate Data Comparison

The process information stored in process data can be extracted in the form of a model by means of system identification methods. Poor extraction of this information may be indicative of timeseries which do not contain the dynamics of the process or from which the dynamics cannot be determined. A good SID method can fail due to misunderstood process dynamics or structures. It is, however, aimless to test a method on a dataset, and specified variables, if one is not sure that the dataset does not contain deterministic process dynamics. Evaluation of data discussed thus far has been mainly heuristic and based on visual inspection.

Barnard and Aldrich (2001) developed a nonlinear SID methodology that enables identification of nonlinear process dynamics from a one-dimensional or multi dimensional

timeseries. This methodology makes use of a culmination of various methods of timeseries analysis to explain the nonlinear dynamics collected in the timeseries.

The use of the above mentioned methodology, contained in a Matlab® based toolbox, and the results of this analysis will be included here as it confirms if a timeseries is apt for modelling and further assists with the choice of datasets.

4.3.5.1 Overview of the Nonlinear System Identification Methodology

The methodology developed by Barnard and Aldrich (2001) is used to determine the amount of information, or dynamics, captured in the timeseries. For each of the models which needs to be generated, the output variable dynamics will be analysed. In each of the models, the output variable only includes a single variable timeseries making the analysis simpler than in the case of multiple outputs.

The methodology followed in this research using the developed toolbox, as well as the outputs aimed at, is displayed in the diagram in Figure 21 below.

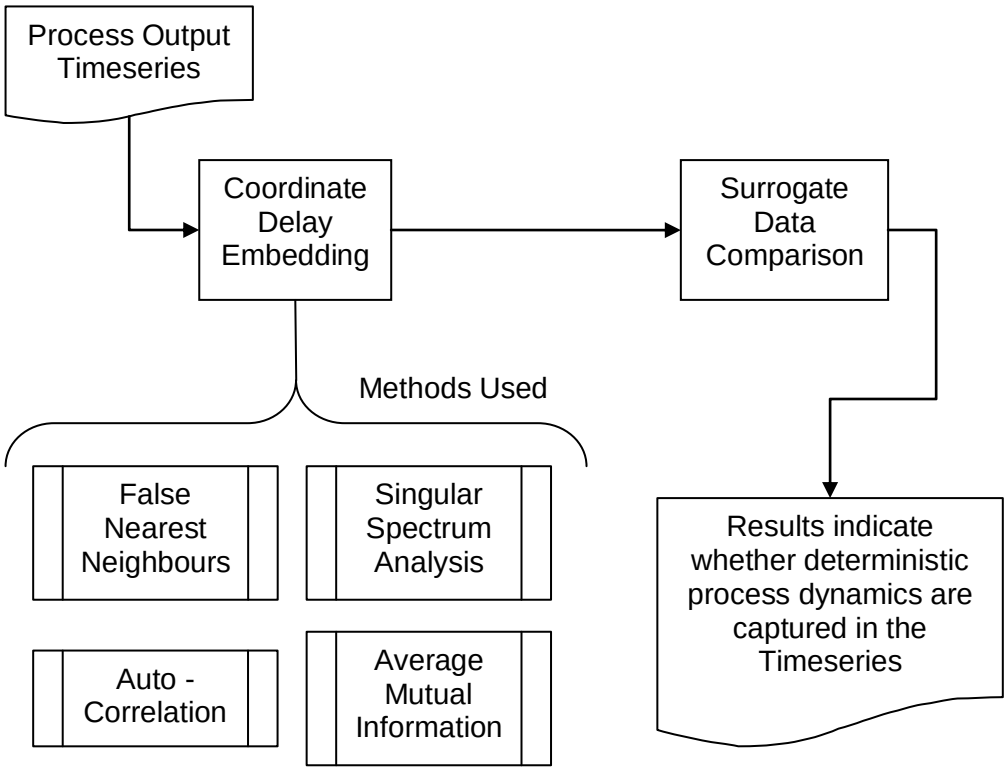


Figure 21: Process flow followed in the use of the methodology used to ascertain whether the process dynamics can be obtained from the timeseries

The techniques used in this research and included in this toolbox are:

- Coordinate Delay Embedding constructing the a state space;
- Autocorrelation or AMI (average mutual information) to obtain the time delay per variable;
- False Nearest Neighbours to determine the embedding dimension; and
- Surrogate data used for classification of data as deterministic or stochastic.

For better understanding of the use of the formulation of the toolbox and the various techniques the reader is referred to Barnard and Aldrich (2001).

4.3.5.2 Analysis of Surrogate Data

The surrogate data comparison indicates whether the process dynamics present in the timeseries could possibly contain deterministic dynamics. This *a priori* knowledge of dynamics present in the system data are used in this research to in selecting the appropriate subdivided time series. The result is not necessarily that the deterministic dynamics in the timeseries can be extracted by the SID technique, but rather assists in filtering the large number of datasets for a better timeseries to use for modelling. It is thus a contributing section included in this case study's SID methodology.

Stochastic data is compared to the time series being investigated. Calculated statistics of both datasets are compared visually. The degree of separation between the surrogate data and real data, allows a conclusion to be drawn regarding if the timeseries contains stochastic or a deterministic dynamics. The larger the separation, the better is the chances of identifying deterministic dynamics and ease of modelling the process. The extraction of the deterministic information is still dependant on the SID algorithm used. The results does not ensure deterministic dynamics in the timeseries, but rather strengthens the probability that the timeseries contains deterministic dynamics.

The visual comparison could be a weakness of the method, but is not a major factor in this research as this method is used to select between various subdivided timeseries. In the

absence of process knowledge and various inexplicable occurrences in the flash dryer output timeseries, this method allows possible better selection of the timeseries to use.

4.3.6 Construction of Latent Input-Output Variables for Dynamic Model Structures Identification

For a dynamic model the system identification procedure needs access to the process history. The drying process has no internal process states which are being measured, however a variation on this theme is possible by creating a state space, where the states consist of the previous output variable values. A variation on the theme is used in this research, seeing as the state space is reconstructed, but the process inputs are delayed and supplied to the GP algorithm for model structure selection. Based on the method of dynamic modelling used by Madar et al. (2005) using a GP algorithm to identify model structure both the delayed influence of process input and output variables are investigated. It is noted that this is not a theoretically correct coordinate delay embedding, as the input variables are also "embedded" in a sense. The logic behind this approach lies in the ability of the GP to identify a model structure from various possible solutions, allowing non-contributing variables to be rejected by not including these in the model structure.

In genetic programming it is common to make use of a time shift operator to select dynamic inputs and delayed versions of variables (Hinchliffe & Willis, 2003). The GPOls toolbox is not constructed in this way, and requires various latent variables to be constructed representing the delayed versions of the variables. The selection procedure will thus not include time shift operators but rather choose between a large number of variables, in this case constructed similar to the coordinate delay embedding approach as presented by Barnard and Aldrich (2001). This pseudo-embedding includes the process outputs, effectively the reconstructed state space, as well as various delays of the process input for selection by the GPOls algorithm.

4.3.6.1 Overview of Latent Variable Construction

The latent variable construction was used in a genetic programming dynamic modelling exercise by Madar, Abonyi and Szeifert (2005). Madar et al. included all the lagged versions

of each variable in the latent variable construction. In so doing a whole new collection of variables was created. As far as the GP algorithm is concerned, these variables are all independent of each other and can be viewed separately as contributing to the process outputs. The independence of these variables is ensured during the identification of the delay parameters.

The construction allows both investigation of the unknown process delays, as well as construction of a dynamic model. The GP is capable to reject delays which do not contribute.

4.3.6.2 Determining the Latent Variable Delays and Parameters

The construction of the latent variable matrix as well as the identification of the delays and dimensions used are discussed in this section. To start off with it is helpful to portray each column of the latent variable matrix (each represents a latent variable) as

$$x(t - mk)$$

where *k* is the time delay and *m* the number of delayed versions of the variable. Each column will thus be a 'new' variable or delayed version of a variable.

These parameters *k* and *m* need to be determined to ensure that the latent variables are statistically independent.

Barnard and Aldrich (2002) states that *k* is determined by either using

- Autocorrelation; or
- Average mutual information (AMI);

and *m* is determined by means of

- a False Nearest Neighbourhood algorithm following on the identification of *k*.

These three methods are all included in the toolbox developed by Barnard and Aldrich (2001). These parameters will thus be obtained by making use of this specific functionality in the mentioned toolbox and will assist in construction of independent latent variables.

Note that although in theory the GP algorithm is capable of selecting between a sequence of single time interval lagged variables, the correlation between these variables puts strain on the GP algorithm to differentiate between these single lagged variables. In this research the GP is assisted by predetermining which delays will contribute the most process information to the model.

4.3.6.3 Constructing the Latent Variable Matrix

The construction results in an interpretable matrix of variables where the values of the lagged data can be clearly identified with the process, if they are not normalised. The matrix size, per one dimensional timeseries, is *(n-k)* by *m*. The construction for each one dimension/variable is done independently and then each construction is concatenated with the rest. The varying values of *k* for each variable will cause varying matrix dimensions. The sections at the bottom of the latent variable reconstruction are deleted resulting in an overall matrix of size (*n-K*) by *M* where

$$K = \max\{k_1, k_2, k_3, \dots, k_i\}$$

$$M = \sum_{1}^{i} m_i$$

$$i = number\ of\ variables$$

The resulting matrix for 2-input-single-output process, with inputs u_1 and u_2 and output y, will look as shown below. Delays for all the example variables are 2 and number of latent variables for all are 2. Note that the current value of the process output is not included in the latent variable reconstruction, as it only results in selecting the previous process output as the one-step ahead prediction.

Time	u1(t-2)	u1(t)	u2(t-2)	u2(t)	y(t-2)	One step ahead y(t+1)
t-6	1	3	90	92	50	53
t-5	2	4	91	93	51	54
t-4	3	5	92	94	52	55
t-2	4	6	93	95	53	56
t-1	5	7	94	96	54	57
t	6	8	95	97	55	58

Figure 22: Latent variable matrix used for identification of a one-step ahead prediction model

The GP algorithm is expected to extract the information for the one step ahead prediction at each time step according to the values in that row of the matrix as included above. At time *t* the expected output is *58* and the available input values to include in the model structure are *6, 8, 95, 97* and *55*. The GP is capable of nonlinearly combining whichever of these values to obtain the model output for a one-step ahead prediction. This is further discussed in the next section.

This discussed approach is followed in this research by means of the created function *gpols_gen_dataset*. This additional function is discussed in Appendix E.4.5.

4.3.6.4 Use of the Latent Variable Matrix by the SID Algorithm

This latent variables construction step foregoes system identification. The resulting matrix columns, consisting of these delayed process variables, are then used as a multivariate input to the GP algorithm. Each matrix column is viewed as a separate variable which could possibly influence the process outputs. The GP algorithm tests the variables to find the best suited combination of delays and variables. The selection of model structure and degree is left to the GP algorithm. Practically the GP sees many variables, as depicted in the conceptual diagram below. The input, *x*, enters the latent variable construction and results in *k* latent variables, *x(t-kd)*. The GP uses each of these as a model input and identifies the relationship with the expected output. The result is a model in terms of all the *x(t-kd)* values.

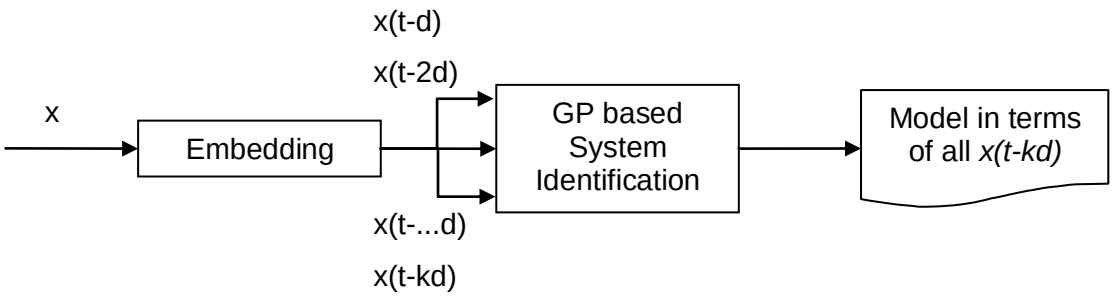


Figure 23: A single process input is re-constructed as a latent variable set, creating *k* latent variables, all seen by the GP based SID method as a separate variable used for system identification.

Note that for this drying circuit it was decided to omit the first instance, *k*=0, of the process output variable as a process output variable lagged one time step will only result in the

previous process output chosen as model input. This high correlation exists due to the relatively slow variation in the process outputs every 5 seconds. Such a self feeding model, using only the previous output, will not be helpful for prediction or control.

The delay parameters for selected datasets are included in Appendix D and discussed further in Chapter 7. These parameters will be used for the latent variable construction for the training of the model. Only the training dataset delay parameters will be used, and will be adopted by the validation timeseries, irrespective of which dataset is used for validation. The parameters trained with will be adopted by the controller during online data preparation, as this reconstruction of latent variables is required by the identified model structure.

4.4 Summary of the Data Preparation Methodology

The control strategy for the HGG and FD sections were established and the data reduced accordingly. The data were filtered for times where the APC controllers were active during data collection. These areas were removed from the modelling datasets. The data bias was introduced as a requirement for the square root function used during system identification.

The remaining subdivided timeseries were investigated for deterministic information by means of the surrogate data comparison. Finally the construction and use of the latent variable input regressor set was explained.

Chapter 5 System Identification with Genetic Programming

The nonlinear system identification procedure used for identification of the drying circuit dynamics is Genetic Programming with orthogonal least squares (GPOls). The toolbox developed by Madar et al. (2005) was analysed and found to follow trusted GP principles. The same approach was used by Coelho and Pessôa (2009) for nonlinear identification of a mechanical experimental setup.

This section starts off with the use of GP as SID technique and moves onto the specific GP algorithm, GPOls toolbox, used. The GPOls parameters and fitness function are discussed. The GPOls toolbox is compared to Discipulus ® as a benchmarking exercise. The adjustments and additions to the toolbox and experimentation process are set out. This section ends off with the specific experimentation logic and approach followed for identification of FD and HGG models.

5.1 Genetic Programming as System Identification Technique

A nonlinear autoregressive with exogenous inputs (NARX) model makes use of past process outputs and the process inputs in a nonlinear empirical structure to represent the process. The nonlinearity suits the point raised previously that dryer dynamics are nonlinear. The use of process history in terms of past process inputs and outputs allows for the delays in the drying process. This is also the structure which will be investigated in this research. The NARX model structure is discussed in Appendix E.1.

Coelho (2009) identifies six steps in the procedure for identifying a NARX model by means of genetic programming. These six steps coincide with other more general discussions in literature. (Ljung, 1999).. These steps are depicted in the diagram below.

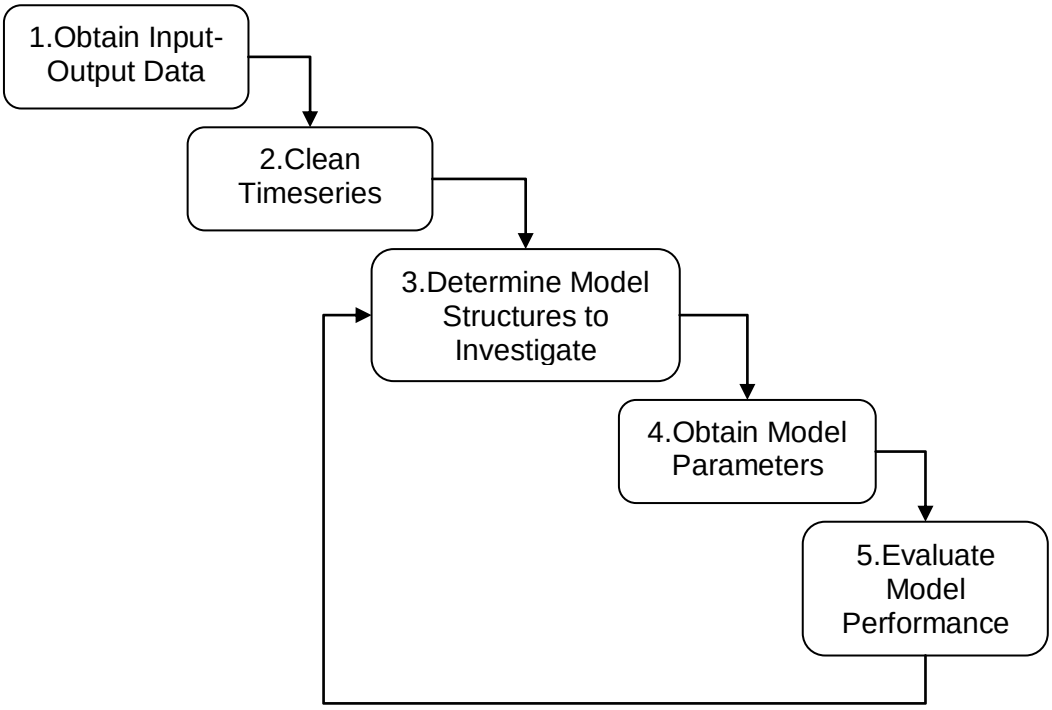


Figure 24: General System Identification Steps. Steps 3 and 4 are dealt with by the GP.

Steps 1 and 2 have been discussed in the previous sections (see Dataset Preparation and Analysis). During the next section steps 3, 4 and 5 will be covered by the discussion of the GP algorithm which will be used.

The power of GP lies in handling step 3 with its ability to choose a model structure. Most modelling approaches, even GP, require inputs determining or stating the model structure. However, genetic programming can search a whole solution space, or part thereof, for best fits. This solution space may include various model structures, functions, variables, combinations of variables and orders/delays.

As stated earlier, it was found that the dynamics of drying consist of highly nonlinear and sometimes misunderstood combinations of phenomena (Mujumdar & Zhonghua, 2008). The GP allows, through the computer processing power, a stochastically driven search method for exploring the solution space defined by the user. It is thus able to explore various model structures and the influences and combinations of these variables on the process output.

The trade off is time and cost of finding a solution, where cost is equal to computing power required. Seeing as computers and software are very powerful and easily accessible, this

cost is not a large price to pay for offline modelling. This however depends on the size of the search space defined by the user. The GP allows the dryer dynamics to be explored without predetermining or defining solution structures and preconceived ideas, which was stated in literature as the main problem in the modelling of dryer dynamics.

Note that in some of the experiments previous solutions were loaded into the population. This allowed evolution to commence from previous best solutions found in other experiments, or to force the experiment to search in a particular direction. This is seen as guidance to searching the solution space, rather than limiting it.

5.2 Genetic Programming with Orthogonal Least Squares Toolbox

As genetic programming (GP) is similar in operation to a genetic algorithm (GA), an algorithm generally well known; a thorough generalised discussion of GP is not discussed here, but only included in Appendix E – Genetic Programming. The reader is referred to this appendix for more detail regarding GP. The general workflow of genetic programming is presented in Appendix E.2.2. The GPOls toolbox follows the same basic process flow except for the following

- Tree pruning by means of orthogonal least squares;
- Direct Reproduction by means of a Generation Gap; and
- No early termination of the run when a pre-established fitness level has been reached.

The following diagram displays the high level logic of the GPOls toolbox. The three bullets above are portrayed in this diagram, as well as extra information regarding “parameter calculation” and “displaying the answer”. This aspect are included seeing as adjustments were made to the display of the results.

A more comprehensive discussion of the algorithm and the pseudo code are included in Appendix E.3 GPOls Toolbox.

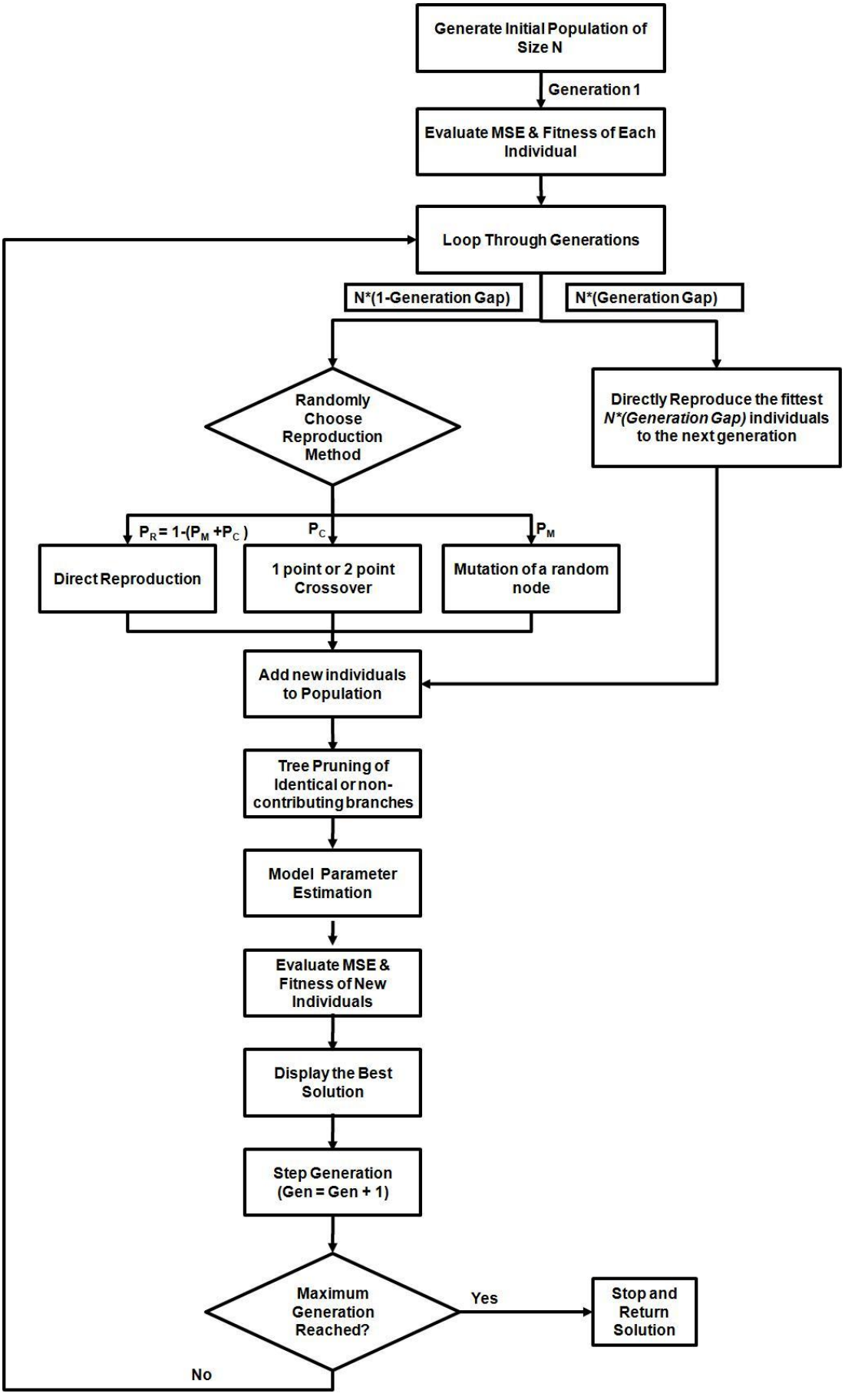


Figure 25: Genetic Programming with Orthogonal Least Squares Algorithm Workflow

Madar et al. (2005) developed a freeware GPOls toolbox in Matlab. This toolbox consists of a number of loose files located in the same directory and is available for download at www.fmt.veim.hu/softcomp. It however does not include any user manual or precise description of how this toolbox works, except for the inclusion of two demonstrations which can be run. It is thus necessary to spend time in working through the code and the workflow to harness the functionality of the toolbox.

Some features of the toolbox used will be briefly discussed here with further information available in Appendix E.

5.2.1 Orthogonal Least Squares and Over Fitting

Over fitting occurs when an individual's tree structure becomes too specific to the training dataset and the ability to fit unseen data from the process is compromised. Too many degrees of freedom are added into the model, causing it to lose its ability to generalise. Three methods in literature are noted as a possible measure and solution for over fitting in GP's.

1. Penalise an individual in the fitness function based on tree length. Grosman and Lewin (2002), McKay (1997) and Madar (2005) make use of the same method by means of a sigmoidal penalty function based on the tree size.
2. During the training run each individual through the "unseen" validation data as well to measure its fitness. At a point the validation fit reaches a minimum. Further development of a solution results in over fitting (Willis et al., 1997).
3. Madar et al(2005) introduces a tree pruning algorithm based on orthogonal least squares to establish the contribution of each tree to the variance in the output, calculating the "energy" of each term and measuring this quantified "energy" against a threshold value.

The 1st and 3rd measures are followed in this research, where the 3rd measure, tree pruning, is discussed here. The 1st measure, penalising overly complex models, is discussed in the next section. The OLS algorithm is discussed here.

The OLS algorithm enables the selection of the most significant subset of regressors from the original set based on the contribution of each regressor to the end goal. This contribution, or energy, is measured against a minimum threshold value to select the subset from the original regressor set. The aim is to create more parsimonious solutions and prevent overfitting. OLS has been compared to support vector machine techniques and found to create more concise accurate solutions (Chen, 2006). The theory of OLS is included in Appendix E - E.3.5 Orthogonal Least Squares Theory.

In some cases it is necessary to makes use of a very low OLS threshold to assist evolution in the population to start, and later to increase the threshold as evolution continued to prevent over fitting. For this reason an adjustable OLS threshold was implemented. As the fitness increases the OLS threshold will increase. This was only implemented where evolution struggled to occur and is noted in

5.2.2 Fitness Function

The fitness function provides a value measuring each individual's ability to solve the problem at hand, or model fit. There are various definitions for how to calculate the fitness. For a system identification exercise this will more often than not be based on a goodness-of-fit type calculation. A discussion on the evolution of fitness functions and also the use of variations used in literature is included in Appendix E - E.2.4 Fitness.

In this research it was decided to stay with the definition used by Madar et al. in the original toolbox. This definition coincides with popular belief in literature that the correlation coefficient between the model output and the training dataset is a better measurement for system identification (McKay, 1997; Madar et al., 2005; Willis et al., 1997). It is also popular in literature to incorporate a penalty function in the fitness calculation. This penalty function reduces the fitness based on the length of the solution above a pre defined limit. A weighting for this penalty is also included. The penalty ensures that the more basic solution will be rated above a more complex one, unless the latter is a much better model fit.

The fitness function used by McKay(1997) and Madar et al. (2005) following sigmoidal penalty fitness function is used, with one alteration. A penalty is included setting the fitness to zero if any imaginary numbers are produced by the identified solution. The altered fitness function is:

$$f_i = \frac{\alpha_3 r_i}{1 + e^{[\alpha_1(L_i - \alpha_2)]}}$$

Where f_i is the individual's fitness; r_i is the correlation coefficient between the model output and the process output; L_i is the length of the tree in number of nodes; α_1 is the penalty weighting; α_2 is the maximum tree length at which the penalty is activated; and α_3 is a factor set to zero during the run if any imaginary answer is produced by the solution. It is defaulted to 1 otherwise.

This fitness value is calculated for each individual and ranking of individuals take place in terms of this value.

Note that although the mean square error (MSE) and R^2 values are calculated during the experiments they are not used during the search for a solution but rather as a universal indication of goodness-of-fit in the post-identification analysis of the solution.

5.2.3 GPOls Toolbox Parameters

GP in general has a number of generic parameters which can be adjusted. Changing these will alter the way the search space is investigated, the computational intensity and time required, the solution found and, eventually, the usefulness of the GP run results. The parameters adjusted may and will differ for various GP algorithms seeing as various different techniques are incorporated. Furthermore, different combinations of these parameters will result in different sections of the search space to be investigated for solutions. In the GPOls toolbox the parameters required by the toolbox are tabulated in Table 8 below. A discussion around each of these parameters, and their function, is included in Appendix E - E.3.3 GPOls Parameters.

Madar et al. (2005) provide a default combination of these parameters which are found to be the best general starting point. Various combinations will be attempted in identification

experiments and documented. These will be discussed on a per experiment basis. The default settings used for the initial evolution of solutions are also included in the table.

Table 8: GP parameters for the GPOIs toolbox and their recommended default values

GP Adjustable Parameter	Abbreviated Name in Code	Value Range	Recommended Default Value
Generation Gap	GenGap	{0,1}	0.8
Probability for Crossover	PC	{0,1}	0.7
Probability for Mutation	PM	{0,1}	0.3
Selection Type	SelTyp	[0,1,2...Population size]	2
One or Two Point Crossover	CrossOver	[1,2]	2
Tree Size Penalty: Weighting	TreePen	{0,1}	0.2
Tree Size Penalty: Location	TreePenLoc	[0,1,2...Max Tree Size]	25
Orthogonal Least Squares Tree Pruning Threshold	OLSThres	{0,1}	0.7
Polynomial Evaluation on/off	PolyEval	[0,1]	1 (on)
Evaluate all or only new individuals	EvalInd	0=new; 1=all	0 (new only)

The population size, number of generations and maximum tree depth are also adjustable parameters in the GPOIs toolbox, but are handled differently by the GPOIs algorithms. They are thus separated from the table above. Default values have not been supplied for these either. These three parameters are however also adjusted in the first section of the experiment logic and discussed, together with all the GPOIs parameters in the aforementioned appendix, Appendix E - E.3.3 GPOIs Parameters.

It should be noted that the choice of functional and terminal sets can also be seen as parameters to the GP algorithm, although they do not explicitly influence the GP logic and operations. The terminal set contains the list of all the variables which can be used in evolution of a solution. The functional set contains all the mathematical operators which can be used and will alter the size of the search space. These sets are included in Appendix E - E.3.4.

5.2.4 Dryer Experiment Parameters

Aside from the GPOls parameters there are a number of parameters specific to the dryer SID experiments developed for this study. These include the experiment name, process to model and subdivided timeseries to use, among others. A full list of these parameters and the meaning of each is included and discussed in Appendix E E.3.2 Drying Experiment Parameters.

These parameters do not influence the GP algorithm as such, but rather the administration of the experiment and the specifics regarding dataset details and validation methods.

5.2.5 Benchmarking of the GPOls Toolbox

The GPOls toolbox is used and referred to in literature (Madar et al., 2005). Coelho and Pessoa (2009) used the same method. The toolbox is also adapted by the Matlab Community. Nonetheless, it was felt necessary to measure the efficiency and ability of the GPOls toolbox measured against proven GP software. The comparison will be done on model fit ability for the flash dryer data. Discipulus Lite ® was used as alternative GP software. Furthermore a linear ARMA model was also fitted to the timeseries and compared to the GPOls results.

Discipulus ® is a Linear Genetic Programming software tool for creating programmes based on datasets and a fitness function. In this research it will be used to measure the accuracy of the GPOls toolbox being used. Discipulus ® is programmed to run directly on machine code, making it superior in execution speed. It has been found to be between 60-200 times faster than other interpreting systems(Francone, 2001). Discipulus ® makes use of the standard GP method as discussed previously. It uses the standard mean square error (MSE) value as a fitness measurement during training and validation. The software used is Discipulus Lite ® and the resulting solutions are provided in C/C++ or assembler code. The result from Discipulus ® will not be used.

The linear ARMA models is fit using the CSense Linear Model. This is a basic linear MISO model fit.

Four experiments are attempted per benchmark modelling technique: one for the flash dryer and one for the hot gas generator. Identification of models for these two processes is completed using both the average mutual information (AMI) and autocorrelation methods of determining delay parameters. This results in four experiments. The delay parameters mentioned are discussed at a later stage.

This comparison is based on the validation set MSE and R^2 values of the best identified model.

5.3 Additions and Adjustments to the GPOls Toolbox

The GPOls toolbox was investigated for possible required upgrades or changes to the logic. The toolbox was found to follow the normal GP workflow and that no major adjustments in the workflow and logic is required. Three smaller areas were located where the toolbox was adjusted. Any other work was additions to the toolbox and did not influence the engine of the GPOls algorithm but rather the investigation of the resulting individuals and preparation of the dataset.

5.3.1 Adjustments

Three changes to the GPOls engine were made. The first was the removal of the weighting matrix, which weights the importance of each data point in the training dataset. This matrix was found to be too memory intensive. This functionality was not needed and the piece of code was removed from *gpols_evaluate.m*.

The second adjustment entails incorrect reference to the calculation of mean square error. It was found that the toolbox was actually calculating the squared sum of errors and not the mean. This basic error was alleviated by dividing the SSE by the number of data points and calculating the MSE.

The third adjustment entails the alteration of the fitness function to include a penalty if any imaginary number is produced by the solution.

The toolbox structures were also manipulated for the following:

- the calculation of R^2 during experimentation;

- extraction of a formula executable in Matlab® if the latent variable reconstruction is available;
- self adjusting OLS threshold; and
- the use of predefined populations.

These are not adjustments, neither are they additions to the toolbox, but rather additional functionality external from the toolbox based on the toolbox results.

5.3.2 Additions

The functionality added to the GPOls toolbox and the Matlab® files where the functions are located, are included in the following table, Table 9.

Table 9: Functionality added to the GPOls toolbox and the names of the functions created.

Functionality	Matlab® Function	System Identification Function	Re-Usable in other GPOls Experiments
Obtain the executable formula string for a specific individual for the population index *ix*.	*gpols_any_result.m*	*Analyse GP Output*	*Yes*
Display the fitness, MSE and formula for a specific individual for the population index *ix*.	*gpols_any_result.m*	*Analyse GP Output*	*Yes*
Display the fitness, MSE and symbolic formula for the *n* best results	*gpols_best_results.m*	*Analyse GP Output*	*Yes*
Obtain the symbolic and executable formulas, MSE, fitness and population index of the *n* best individuals	*gpols_best_result.m*	*Analyse GP Output*	*Yes*
In some cases predefined populations were used as a starting point for a GP run. In such cases it is necessary to evaluate if the population coincides with the chosen terminal and functional sets. This function tests the population and provides a solution if it is incorrect.	*gpols_testpopulation.m*	*Data Preparation*	*Yes*

Functionality	Matlab® Function	System Identification Function	Re-Usable in other GPOls Experiments
Track the evolution of the population through all the generations. A landscape of the fitness and MSE values are plotted for the whole population for each GP run.	*gpols_trackevo.m*	*Analyse GP Output*	*Yes*
Calculate the validation MSE, fitness and residuals for the chosen individual. These results are plotted to compare to the training dataset.	*gpols_validate.m*	*Analyse GP Output*	*Yes*
Select the delay parameters relevant to the chosen training dataset.	*gpols_embedparameters.m*	*Data Preparation*	*No*
Create the latent variable set, with corresponding symbols for an *n*-step prediction model.	*gpols_gendataset.m*	*Data Preparation*	*Yes, provided that a latent variable set is constructed.*

These functions can be divided into the functions used for data preparation and analysis of the outputs of the GP runs. The functionality included, divided into Data Preparation and GP Output Analysis, is discussed below. A complete explanation of the functions in Table 9, and how it works, is included in Appendix E- E.4 GPOls Additions.

5.3.3 Data Preparation: Latent Variable Reconstruction

The dataset was pseudo-embedded as discussed previously. As this latent variable reconstruction procedure can be a long section of code, it was separated from the main GP experiment logic by creating a separate function. The separate function *gpols_gendataset* was included to enable cleaner experiment code and allow the function to be used again for embedding of the validation dataset.

The delay parameters, determined according to the discussion in 04.3.6.2, differ depending on the training dataset being used. To allow easier transition between training datasets, a function was created which serve as a database for the delay parameters. This function, *gpols_embedparameters*, is called before the latent variable set is constructed.

5.3.4 Analysis of Experiment Outputs

In the process of choosing the best model obtained it is necessary to investigate the residual analysis, mean square error (MSE), R^2, fitness and trends of the original dataset versus the model outputs. To enable this it is first necessary to obtain the formula from the GPOls programme memory. The GPOls memory structure needs to be compiled to an interpretable string executable for both the latent variables training as well as the validation sets. This executable string of either a specific, or the best individual is then used to calculate the training and validation residuals, as well as the validation MSE, R^2 and fitness. This string is constructed and extracted by any of the two procedures *gpols_any_result* or *gpols_best_result.*

During the search for the best solution the population could get stuck in a local optimum. A three dimensional plot of the population landscape over all the generations provides a visualisation tool to evaluate the evolution over time. This plot trends the fitness or MSE per individual for each generation. A slow evolving run, too random evolution or a very narrow search can be recognised from this landscape. This assists in guiding the setup of GP parameters for consequent experiments. This is a visual aid to analysis.

A further requirement is validation of the data with alternate datasets during post analysis. A stored experiment output, created during the experiment, is accessed, the best model obtained and the validation statistics and graphs are provided for the given dataset. This provides a better view into the solution obtained and better comparison between results.

The presentation and analysis of data are discussed further and in more detail in 5.4.2 Analysis and Presentation of Results.

5.4 Genetic Programming System Identification Experiments

5.4.1 Dryer SID Experiment Logic

The GPOls toolbox provides the functionality and the engine to enable the GP system identification exercise. The toolbox does however not have a graphical user interface of sorts, but relies on the user to access the GPOls functionality and control the GP run by

means of experiment logic set out in a Matlab® *.m file. The drying circuit GP experiment logic, or system identification logic, is located in a single Matlab® file. This logic is developed specifically for this filter cake drying circuit, but it provides a reusable backbone for any future use of this toolbox. There are also various experiment parameters specific to this case which can be adjusted in the code, as well as the inclusion of a function used to reconstruct the timeseries as a set of latent variables. These, and other details to the GPOls toolbox, are discussed in E.3 GPOls Toolbox.

For every experiment, the SID experiment as well as GPOls parameters are adjusted. A copy of the logic file, the dataset used and results are all stored in a folder according to the experiment name, which is defined in the experiment parameters. The experiment logic is set up to control the following:

- The process to be modelled (flash dryer or hot gas generator);
- The dataset to use;
- Validation Type (as discussed in 5.4.2 Analysis and Presentation of Results);
- Logic of latent variable reconstruction of the timeseries;
- Defining the functional and terminal sets, Population size, Generation size;
- Loading a predefined population or randomly generating a new population (See Section 7.3.1, as well asAppendix E - E.2.3 Initial Population);
- Setting up the GPOls parameters;
- Displaying and saving the results during and after each GP experiment;
- Repeating the GP experiment and storing the results with a unique experiment name in a specified file; and
- Constructing a report of the experiments to enable easy post run comparison of various experiments.

Seeing as a GP is a stochastic modelling technique, it will result in different outcomes each time, requiring repetition of an experiment. It should be noted that the logic was written to allow various repetitions of the same experiment to be run. To enable efficiency in the experimentation step of research, this repetition functionality was included in the logic. The results of each repetition were stored separately for possible analysis at a later stage. The best individual, as well as the training and validation statistics were stored in a report. This report would contain the information for all the repetitions of a single experiment. Each repetition is independent.

The experiment logic is discussed in further detail in Appendix E.3.1 GP Experiment Logic.

5.4.2 Analysis and Presentation of Results

The experiments, and number of repetitions, brings about the task of analysing a large amount of results in search of a best solution. To assist in this task various methods have been employed in the experiment and in the additions to the toolbox. The methods developed, or adopted, and employed are:

- Trend of Population Evolution across the Generations;
- Comparison Goodness-of-Fit Statistics for each Run;
- Trends of the Model Output;
- Residual Analysis; and
- Interpretation of the Model Empirical Formula.

These methods are discussed in more detail in Appendix E - E.5 Methods for Analysis and Presentation of System Identification Results.

5.4.3 Dryer SID Experimentation Approach

The experimentation process started with the default parameters from literature, and then commenced to other parameters based on the experiment outcomes. However, as in any experimentation process a step-wise adjustment is required to ensure valid comparisons between results.

Apart from adjusting the GPOls parameters, the inclusion of the following were also found to influence modelling efficiency and accuracy:

- Choice between AMI or autocorrelation delay and number of latent variables, i.e. defining the terminal set;
- Dataset used for training and validation;
- Considering the inclusion of process anomalies in the data sets; and
- Functional set to use.

A number of modelling experiments will be done with the aim of both finding the best model for control, but also commenting on the best choice, or combination thereof, for the topics listed above. The findings for each of these are discussed in the results sections.

5.5 Summary of the GP Methodology

The GP approach is preferred due to the unknown model structure of the models required. The latent variable reconstruction results in a large solution space to be explored for model structures and models, which the GP is capable of exploring.

The GPOls toolbox is applied, after some adjustments and additions. Two sets of parameters are highlighted:

- GPOls parameters, and
- Dryer SID experiment parameters.

The GPOls parameters adjust the search, whereas the experiment parameters indicate dataset to use and location to store results, amongst others.

The GP approach is benchmarked against linear ARMA models identified. The GPOls algorithm is benchmarked against the commercial GP package Discipulus ®.

Chapter 6 Model Based Predictive Control

The aim is not to build a fully functioning controller, but rather to investigate the suitability of the models identified, together with required online data preparation, for a basic predictive control approach. For this, the best models found will be investigated for use in control. A controller will be developed accordingly.

A general discussion surrounding the MPC theory and terminology is included in Appendix H. The reader is referred to this section for a background on MPC.

6.1 MPC Solution Architecture and Dataflow

The controller solutions will be developed and tested using CSense 4.3 Architect and CSense Server Manager. This software allows direct communication with the OPC (OLE-DB for Process Control) and enables real time simulations similar to which will be experienced on site. This also allows easy integration into the existing site software architecture, although this is not the aim of this research. The Architect is the development environment and the Server Manager is the real time solution deployment environment.

The simulations of the controller are run in real time. A modular approach, same as would be experienced on a plant site, is created using the following stand-alone entities:

- OPC, or process state memory;
- Process simulation to calculate process reactions to control steps;
- Latent Variable Reconstruction;
- Model Predictive Controller; and
- Process initialisation

The solution data flow between these entities is depicted in the following diagram:

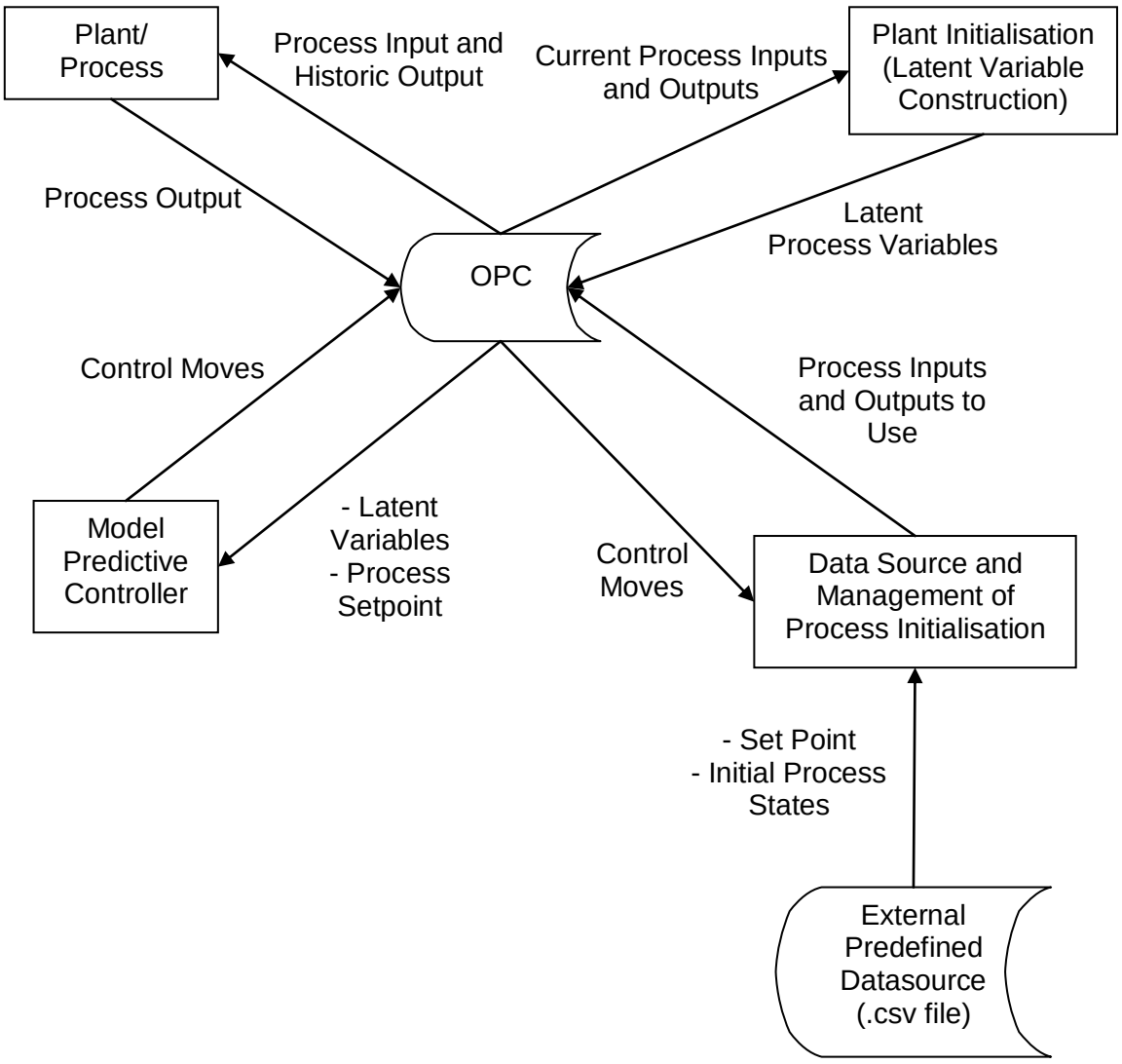


Figure 26: Dataflow for Simulation of Controller. Each entity stands alone with data flowing between the entities, with the plant memory (OPC) being the mutual read and write memory link

The logic behind this dataflow is the modular representation of the plant and controller, with the OPC, or SCADA, being the point of connection. The plant in the diagram is a simulation of the process. The simulation makes use of the process model identified during the system identification exercise. Various process models were investigated for use as the plant representation, but it was decided to use the same model as used by the controller. The lack of model mismatch is noted, but not seen as a problem as the aim is not to create a 100% usable controller, but rather investigate the possibilities of building such a controller based on existing process data and identified models. Once this is proven as efficient, the

next step would be to investigate robustness under model mismatch. This is not done in this research.

The initialisation, i.e. historic values for the latent variable construction, is handled by the data source. The training dataset was used for initialisation to ensure that valid process states are stored in the history of the process. The initialisation entails storing all the relevant past process inputs and outputs in the latent variable construction for use in the model. The plant and controller models each require initialisation. The plant model's initialisation is handled in the same module as the plant. The latent variable construction used by the controller is located in a separate module due to the different execution rates required for latent variable construction and optimisation. The latent variable reconstruction is done every time interval, or 5 seconds in this case. The optimisation requires more time, due to model nonlinearity and, and needs to be separated from the construction of the latent variables set.

Three information variables are included in the data flow to establish when the models, in each of the controller and plant modules, have initialised as well as when the optimisation is complete and the calculated control moves should be implemented. These information variables are passed between the modules through the OPC module. These information variables are manipulated in the modules which they represent.

Once the models have been initialised, a freerun state commences with the data source only used for identifying set point changes, or noise additions to either the process inputs or output. The process output from this point onwards is obtained from the plant model and the control moves, obtained from the controller.

The data streams being passed through the "OPC" module are included and explained in the table below:

Table 10: Explanation of variables passed between the various modules used in the control simulation. The OPC module (plant memory) is the linking module.

Variable Description	Origin and Use of Variable
Control Moves	Optimised by the Controller module every optimisation step. Implemented for the control window by the Process Initialisation and Management Module.
Input Variable	Current Process Input
Output Variable	Current Process Output as calculated by the Process/Plant Module
Latent Input Variables	Normalised latent variable reconstruction of Input variable history. This is used by the controller during each optimisation step.
Latent Output Variables	Normalised latent variable reconstruction of Output variable history. This is used by the controller during each optimisation step.
Output Set Point	Produced by the external .csv file and introduced to the controller.
Additional Input and Output Noise	Included as an option. Input variable noise is added before it is implemented in the Process/Plant module. Output noise is implemented after the real output is calculated by Process/Plant module.
Plant Model Initialised	Information variable indicating when the Process module's plant model latent variable reconstruction is complete and the model is ready for simulation.
Controller Model Initialised	Information variable indicating when the Controller module's plant model latent variable reconstruction is complete and the model is ready for simulation.
Control Moves Optimiser Complete	Information variable indicating each time the optimiser has calculated new control moves. The control variables are implemented from the first until the last control move in the MPC control window. Hereafter the optimiser would have completed the next set of optimised control moves.

6.2 Data Preparation and Constraints

The input and process output data introduced at each time step need to be normalised and biased, if required. The data also need to be included in the latent variable reconstruction for use by the model. These data preparation steps are executed for each run of each module. The latent variable construction is initialised by an external input. The information variables are used to indicate when the latent variable reconstruction is complete and when

the simulation can start. Insufficient data in the latent variables will result in bad quality model outputs. The delay parameters correspond with the model used.

Note that independent latent variable reconstruction and normalisation steps are required for the Process/Plant module model and the Controller plant-model to cater for any further studies which allow model mismatch.

Limiting the plant model inputs and outputs is handled indirectly by the constraints implemented in the optimisation step of the control moves. Control moves constraints are implemented to be within the data range used for training. The plant model outputs are however not limited in any way. No control move size penalties, as noted in literature, are included in the goal function for the controller either.

6.3 Process Prediction and Control Move Optimisation

The choice of prediction window size (N_p) is dependent on the process dynamics, whereas the number of control moves implemented (N_c) was decided to fill the time between optimisations. This is decided due to the long optimisation time required.

The models identified are one-step ahead prediction models. The option is thus to allow a prediction window of 1 time step (N_p=1) and then optimise a control window of 1 time step (N_c=1). The slow dryer process dynamics mentioned in literature however require a longer prediction window and thus freerun prediction. This produces a larger window to optimise control moves over, but requires more computing time. Consequently a larger control window is required to fill the gaps between the optimisation steps.

The prediction window was chosen based on:

- Longest lagged process variable included in the model;
- Time required for the process to settle, based on the outcome of the sensitivity analysis trend in section 9.2 Model Sensitivity Analysis; and
- The freerun prediction stability of the identified process.

The control window was chosen based on:

- The time required for the optimiser to complete an optimisation batch run.

Prediction entailed a recursive freerun prediction using the previous reconstructed latent process variables, previous predicted outputs from the current prediction step and variables which need to be optimised. The optimised variables were initialised prior to optimisation and comprised of the input variables, and possible control moves, across the prediction window. The following figure explains the information flow of the recursive prediction.

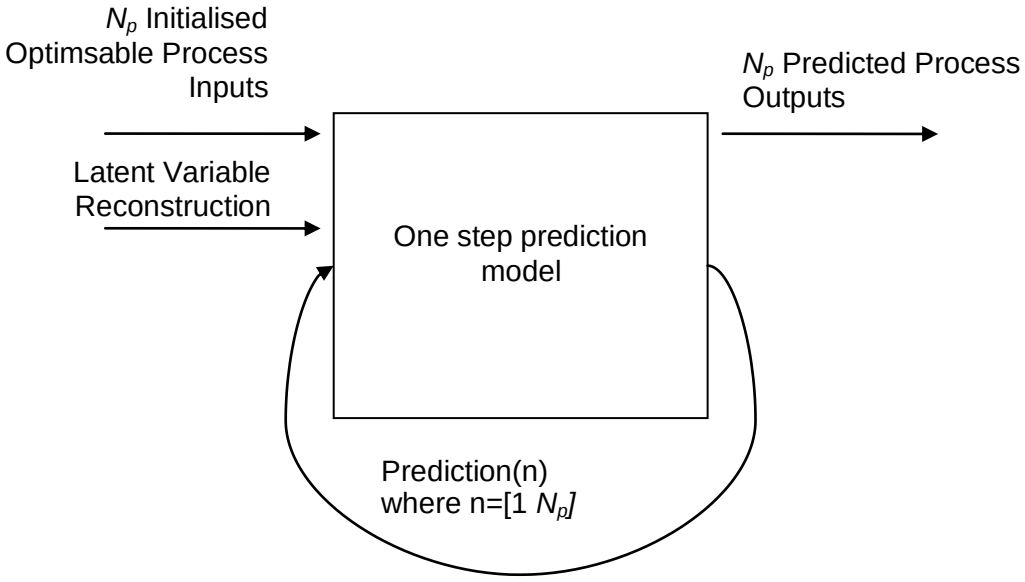


Figure 27: Recursive prediction implemented in the model based controller.

The predicted process outputs are used for calculation of the optimisation goal function. The goal function used is the sum square of errors between the predicted process outputs and the set point calculated across the prediction window.

$$goal = \sum_{n=1}^{N_p} (Set\ Point - Output(n))^2$$

All the inputs for the whole prediction window are optimised, but only N_c control moves of these are implemented. As the process actually commenced from time k to $k+N_c$, the control moves from time $k+N_c+1$ to $k+2N_c$ are implemented. This assumes that the freerun prediction is stable for the whole prediction window.

The optimiser used is a commercially used genetic algorithm (GA) of which the parameters are included in Table 11. The GA implemented is a straightforward GA as is well known in practice. The parameters are fixed and not accessible or adjustable by the user. The GA re-initialises for every execution of the MPC, and keeps no memory of the previous MPC results.

Table 11: Genetic Algorithm optimiser parameters

GA Parameter	Parameter Value
Population Size	1000
Number of Generations	1000
Mutation Probability	0.1
Crossover Probability	0.7
Selection Strategy	Tournament Selection
Termination	Either 1000 generations or when the whole population consists of the same individual

It is assumed that this optimiser is able to optimise well enough. It will be compared against a controller which selects only random versions of the control variable as well as the current live plant control solution. The outputs will then be compared.

6.4 Experiments Investigated and Models Used

It was found that only the hot gas generator could be identified successfully. This resulted in only a hot gas generator model predictive controller being investigated. Only the best process model, as found in system identification, was used for the controller. This same model was used to represent the process. This assumes that a perfect model was identified. This is further discussed in the results of the system identification step. The precise experiments are tabulated in Appendix I and discussed in the results section.

The choice of models used will be discussed in Chapter 9 Results: Model Based Control.

6.5 Summary of MPC Approach

A real time simulation environment is constructed using a modular approach for the controller, plant, model initialisation, process memory and external data input.

The MPC approach is adjusted from convention due to the long time required for optimisation. The optimisation is executed offline and then all control moves are implemented from this optimised batch until the next optimisation is complete. The MPC optimisation makes use of a Genetic Algorithm based optimiser.

The latent variable set is constructed using historic values, and thereafter values produced by the plant model and controller.

GENETIC PROGRAMMING APPROACH, DRYER SYSTEM IDENTIFICATION AND MODEL BASED CONTROLLER RESULTS

Chapter 7 Results: Genetic Programming with Orthogonal Least Squares Algorithm

This section sets out the evolution of the choice in GP parameters and functional sets. The successful use of additions to the GPOls SID strategy is discussed. The GPOls tool benchmarked against Discipulus ® Lite GP toolbox, and linear ARMA models end off the section.

The aim with this section is not to prescribe fixed parameters, seeing as the exploration of parameters and functional sets was not the main purpose, but rather to explain the logic followed during evolution of the experimentation process.

7.1 Algorithm Parameters

Initial parameters were chosen using defaults provided by Madar et al(2005). Parameters as identified by Coelho and Pessôa (2009) were also explored. These parameters were explored during identification of the FD model and adopted by the HGG system identification. The default GP parameters, as proposed by Madar et al(2005) resulted in 2 solutions. The results are presented below:

Table 12: Results for the Flash Dryer System Identification Exercise using the default GPOls parameters prescribed by Madar et al. (2005)

Model	Train MSE	Train R^2
1	0.650	35%
2	0.822	17.9%

From the limited and relatively poor result it is clear that a wider search is required and also a larger number of repetitions of each experiment. Note that the validation statistics are not presented, as a dataset not comparable to the training set was attempted as validation.

Exploration of alternative GPOls parameter settings resulted in a more complex individual being permitted, by setting softer tree size penalty parameters. This, together with a sequential increase in mutation, population size and generation gap, to allow wider

exploration of the solution space, resulted in single versions of a lagged flash dryer feed - the same as Model 2 in Table 12 above.

Eventually a decrease in OLS threshold, a tree pruning parameter, as well as change in selection from random selection to tournament selection (tournament size = 2) resulted in the inclusion of more branches to the trees. Although this could result in overfitting due to more complex and longer model structures, the decreased OLS threshold and softer tree size penalties seem like the only adjustment available to allow further exploration of the solution space. The best parameter set identified and used as base for all SID experiments are tabulated below:

Table 13: GPOIs parameters identified by experimentation and used as base set for all SID experiments for the flash dryer model.

GPOIs Parameter	Value
Generation Gap	0.95
Probability for Crossover	0.3
Probability for Mutation	0.7
Selection Type	2
One or Two Point Crossover	2
Tree Size Penalty: Weighting	0.1
Tree Size Penalty: Location	100
Orthogonal Least Squares Tree Pruning Threshold	0.5
Polynomial Evaluation on/off	0
Evaluate all or only new individuals	0
Population Size	100
Maximum Tree Depth	8
Number of Generations	120
Functional Set	+,*,/

The experiments, using these parameter sets to explore the solution space, were run for both AMI and autocorrelation delay parameters and resulted in the following solutions – the top solution per experiment is presented:

Table 14: Results using the GPOls search parameters as defined in Table 13.

Model	Latent Variable Delay Parameters	Train MSE	Train R^2
3	AMI	0.589	33.4%
4	Autocorrelation	0.600	40.3%

The AMI model resulted in a more basic model, comparable to model 1 in Table 12 with a lower R^2 of 33.4% versus 35%, but a better MSE of 0.589 versus 0.650. The autocorrelation delay model resulted in a slightly nonlinear combination of flash dryer process inputs, concentrate feed and inlet air temperature, as well as a lagged version of the process output. Compared to the model identified from the AMI delay parameters, this model is an improvement based on training R^2. The slight increase in R^2 obtained for this model in the training set to 40% is not significant enough to accept the model as representative of the process dynamics.

Note that at this stage of experimentation the focus was partially, as mentioned earlier, to explore various validation sets as well, however, the validation sets used in these specific experiments were found to differ too much from the training set. Validation statistics are thus omitted at this early stage of the SID experiments. This is not seen as a problem as the later, more final experiments, are compared based solely on validation data.

The main adjustments to the initial parameters allowed more complex individuals by shifting the fitness criteria, OLS threshold, and also increasing the mutation rate to allow a wider search. Further experiments were based on these parameter sets with adjustments in population size, mutation rates, OLS thresholds and tournament selection size adjusted on a per experiment basis. A global optimum was attempted by doing a wide search by means of higher mutation rates, but still enough crossover to allow the current population to develop.

7.2 Functional Set

The choice of functional set contributes to the size of the solution space. A smaller functional set will cause a smaller solution space and accordingly easier exploration of the solution space by the algorithm, but with the risk of limiting the search. Additionally, more complex

functions will result in a more complex solution space, requiring a more robust search algorithm and experiment parameter set combination. This said, the most basic functional set consists of addition (+) resulting in a linear ARMA model. The literature states that the dryer dynamics are expected to be nonlinear, thus necessitating the use of a functional set consisting of nonlinear functions. The inclusion of basic nonlinear functions multiplication (*) and division (/) and more extreme nonlinear functions such an exponent (e^x) and square root ($\sqrt{x}$) functions were also investigated.

7.2.1 Functional Sets: Flash Dryer

The most basic functional set experimented with includes only the linear functions plus and minus [+,-]. This functional set together with the most complex functional set [+, *, /, -, *e^x, *$\sqrt{x}$] produced the best results as can be seen in the table below.

Comparative Grouping	Latent Variable Delay Parameters	Functional Set	Validation MSE	Validation R^2
A	Autocorrelation	[+,-]	0.674	38.20%
A	AMI		0.701	35.46%
B	Autocorrelation	[+,*,/,*-,/-]	0.868	20.43%
B	AMI		0.715	34.14%
C	Autocorrelation	[+,-,*,/,*-,/-]	0.728	33.30%
C	AMI		0.727	32.97%
D	Autocorrelation	[+, *, /, -, *e^x, *$\sqrt{\lvert x \rvert}$]	0.951	12.83%
D	AMI		0.784	27.79%
E	Autocorrelation	[+, *, /, -, *e^x, *$\sqrt{x}$] with bias	0.697	36.10%
E	AMI		0.668	38.48%

Table 15: Comparison of experiments based on the functional set used for FD modelling. Comparable experiments are grouped accordingly

The inclusion of more nonlinear functions [*,/,*-,/-] did not contribute to the statistic and seems to have largely overcomplicated the solution space, causing the GPOls algorithm to struggle to identify a better model.

The inclusion of advanced nonlinear functions includes the use of both the exponent and square root functions. Only the square root function is noted in literature to have been used with the GPOls toolbox. The problem with these functions is that they only have one input

and not two. The GPOls toolbox is however built that each functional node has 2 terminal nodes. The function will thus not have closure as an unused branch will be left, which the GPOls toolbox does not know how to handle. It was decided to make use of a function which will have closure by adding a multiplication sign in front of both these functions resulting in the following:

$$* \; sqrt$$

$$* \; exp$$

Although this approach may be limiting and the decision of which function to put in front each of these may be more complex, the decision was made to use only the multiplication function. This method is specified in the website discussing the toolbox (Madar, 2005).

The use of these complex functions in the functional set resulted in models with imaginary outputs. This is ascribed to the handling of negative numbers by the square root. The imaginary outputs resulted in imaginary correlation coefficients and eventually fitness values. The toolbox was not created to handle these, and the algorithm could not order the solutions and evolve in a direction of higher fitness. This is visible in a 3 dimensional plot of the fitness values per individual per generation noted during the run of one such experiment.

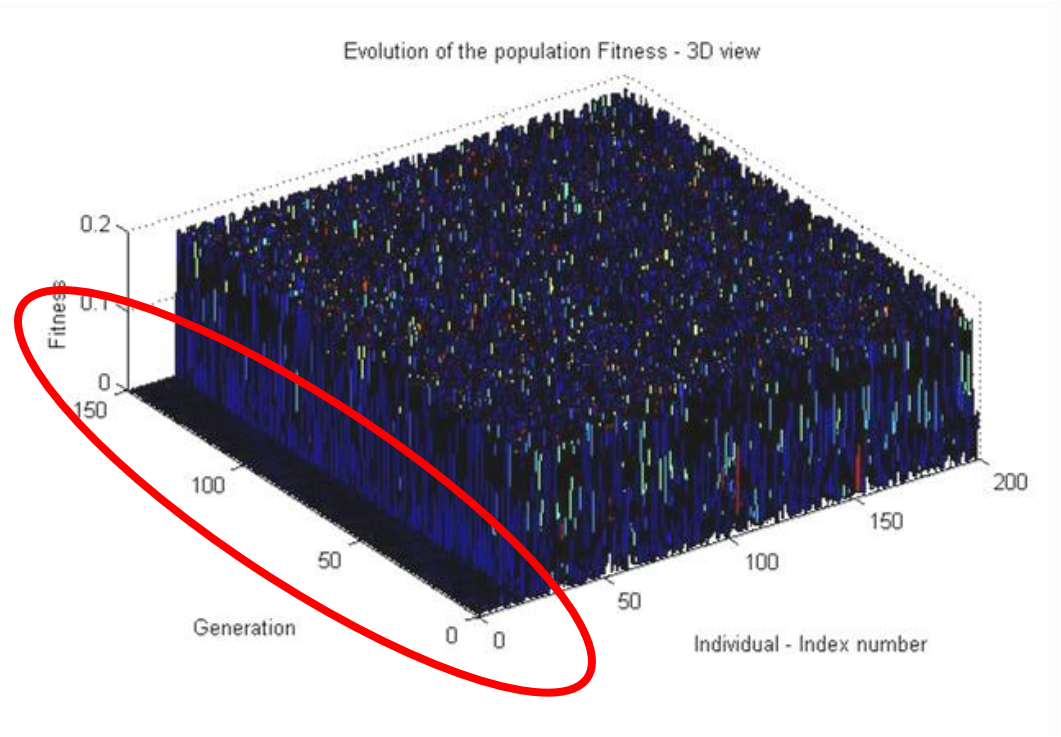


Figure 28: The landscape of the fitness values for all individuals across all the generations indicate a flat stretch, circled in red, where the fittest individuals are supposed to be situated. These individuals are incorrectly rated as fit due to large imaginary number contingents increasing the absolute value of the individual's fitness.

The flat strip on the left in the population throughout the generations results from direct reproduction of the fittest individuals. The algorithm sorts in terms of the absolute value of the imaginary and real numbers and copied individuals with higher imaginary values over as the fittest solutions. The GPOls algorithm was accordingly adjusted to sort according to the real values only.

Two possible solutions were investigated:

i. Take the absolute value of the input ($\sqrt{|x|}$); or

ii. Shift the whole normalised timeseries with a positive bias larger than the largest negative in all variables in the timeseries ($\sqrt{(x+bias)}$).

It was found that a positive bias shift in the timeseries will not ensure real value closure for the square root function, as the GP is allowed freedom with regards to model structure and can subtract values from each other inside the square root function, resulting in a negative. This was handled by altering the fitness function to include a penalty when an imaginary

number is produced by an individual. This penalty function adjustment was mentioned in section 5.3 Additions and Adjustments to the GPOls Toolbox. Results are of these adjustments are discussed in section 7.3 Successful Use of Additions to the GPOls Approach.

For the flash dryer it is concluded from results thus far that

- the choice of functional set can overcomplicate the solution space and cause the search to struggle to initialise;
- the more complex functional set did not necessarily enhance the models identified for the flash dryer;
- the use of the positive bias offset in conjunction with the square root function is preferred to the use of the square root of an absolute value;

7.2.2 Functional Sets: Hot Gas Generator

The hot gas generator SID experiments indicates a slight improvement with the use of the advanced mathematical functions. This is indicated in the following table.

Table 16: Comparison of experiments based on the functional set used for HGG modelling. Comparable experiments are grouped accordingly

Comparative Grouping	Latent Variable Delay Parameters	Functional Set	Validation MSE	Validation R^2
A	Autocorrelation	[+,*,/,*-,-,/-]	0.133	77.1%
A	AMI		0.241	58.5%
B	Autocorrelation	[+, *, /, -, *ex, *√|x|]	0.181	68.9%
B	AMI		0.192	66.8%
C	Autocorrelation	[+, *, /, -, *ex, *√x] with bias	0.133	77.1%
C	AMI		0.167	71.2%

For use of the basic mathematical functions, a large difference between the performance of the autocorrelation delay parameters outperforming the AMI delay parameters. Other than this, the improvements in validation statistics improve as the functional set complexity increases.

The inclusion of e^x and $\sqrt{|x|}$ improves the model performance for both delay parameter sets for experiment grouping B. The use of the bias together with the square root function (no absolute value) provides the best solution. This result correlates with the same finding for the flash dryer SID exercise. The bias with the advanced functional set is thus noted as the most successful functional set.

7.3 Successful Use of Additions to the GPOls Approach

Additions and adjustments made to the GPOls approach and toolbox are noted in the methodology section 5.3 Additions and Adjustments to the GPOls Toolbox. The contributions of these will be briefly highlighted for inclusion in future research.

The contribution of the automatic calculation of the R^2, correct calculation of MSE, automatic calculation of the validation statistics, repetition of the experimental setup and the summary report is self explanatory as these save time and improve experimentation efficiency. The result however remains untouched. The introduction of the latent variable reconstruction still needs to be discussed and will be handled at a later stage. The focus now will be in the use of a predefined population, adjustments in the fitness function and the use of the trend of fitness values for the evolution of an answer.

7.3.1 Predefined Population

The first adjustment worth noting is the use of a predefined population. As seen with the exploration of the functional set [+,*,/,*-,/-] in the previous section, section 7.2, the search was unable to initialise a successful search in a specific direction. This was also noted during the exploration of the functional set including $\sqrt{|x|}$. Both these experiments included the same functions as previous experiments, but could not find even those solutions.

The possibility of predefining a population provides a jumpstart for a solution. It also allows a search to be initiated where it stopped last time, using new search parameters or an expanded functional set. It is noted that this might nudge the search into a suboptimal direction. This is however a risk worth taking, although the modeller should be aware of this.

As an example, when comparing the AMI delay solution noted in Table 17 to the solution using the same functional set, but starting with the predefined solution noted in Table 14 for AMI delay (repeated here in the table below), the benefit is clear.

Table 17: The use of a predefined population (listed first) resulting in an improved solution (listed second)

Latent Variable Delay Parameters	Train MSE	Train R^2	Validation MSE	Validation R^2
AMI	0.623	43.73%	0.715	34.14%
AMI	0.430	47.56%	0.622	42.74%

The solution from the population with no predefined population is listed first in Table 17, whereas the second solution listed results from a predefined population. The predefined population supplied an advanced point to initiate various searches from, whereas each repetition of the experiment starting from a random population possibly seldom found a more advanced point to search from. This resulted in fewer searches breaking through to a point where an improved model could be identified.

As mentioned, this functionality supplied the possibility of using various different functional sets and GP parameters in a series of experiments. This was done when exploring the use of the square root and exponent functions, as well as negative division and multiplication.

7.3.2 Fitness Function Adjusted

The square root function could produce imaginary numbers, resulting in an unstable model. Although the sorting algorithm for the GPOls toolbox was adjusted to only look at the real outputs of the model when sorting, it is still possible for a high real output to have an imaginary number contingent, and so be included in the next generation. This needs to be handled by removing such individuals.

The fitness function was adjusted by adding a parameter penalising the fitness if it contained an imaginary number. The result can be seen indirectly in a 3 dimensional trend, Figure 29, of the MSE values for each individual across all generations. The white spots are

where the model output contained imaginary numbers. The spikes in MSE values are due to unstable model output brought about by the nonlinear functions.

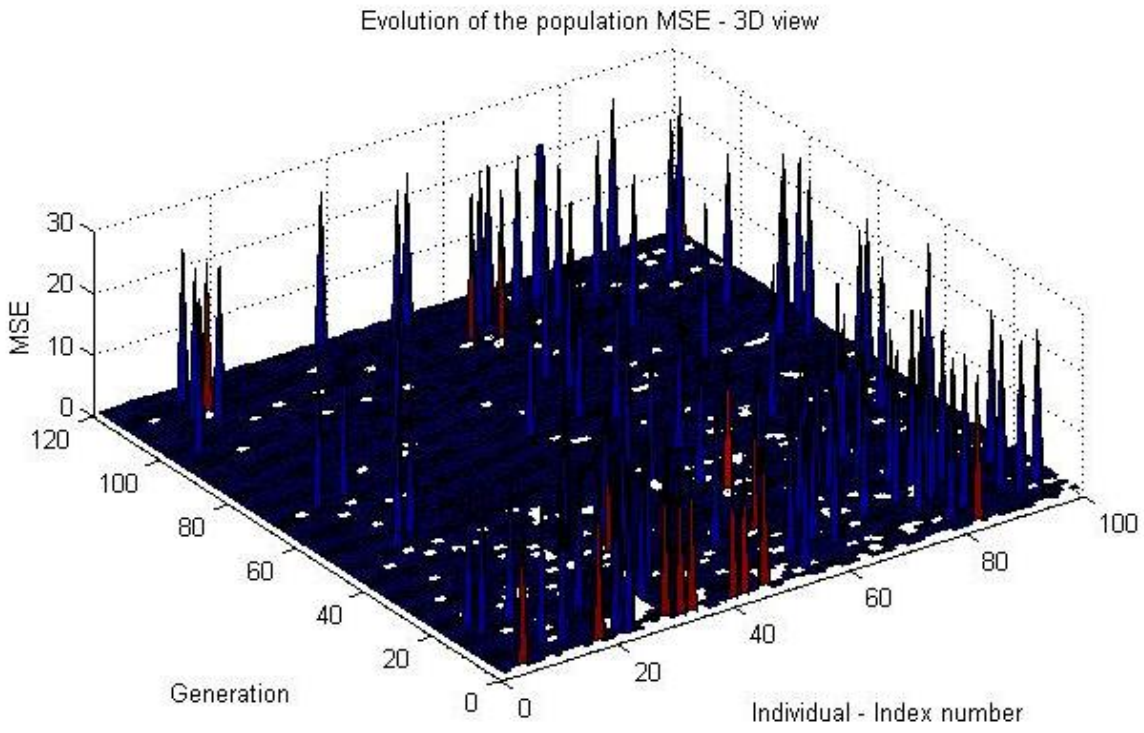


Figure 29: MSE values across the evolution to illustrate the handling of individuals with imaginary outputs by the adjusted penalty function.

As can be seen from the disappearance of white areas, individuals with imaginary number outputs are removed. It can only be due to the penalty function. It is thus concluded that the penalty function works. Without this successful penalty function, the population could evolve into a set of models producing non-real outputs, which would not assist in the modelling of the dryer.

7.3.3 Trend of Fitness Landscape Evolution

The trend of the fitness landscape assists in visualisation of the workings of the GP. It provides a view in how wide the search was as is illustrated when comparing the 2 trends in Figure 30 and Figure 31.

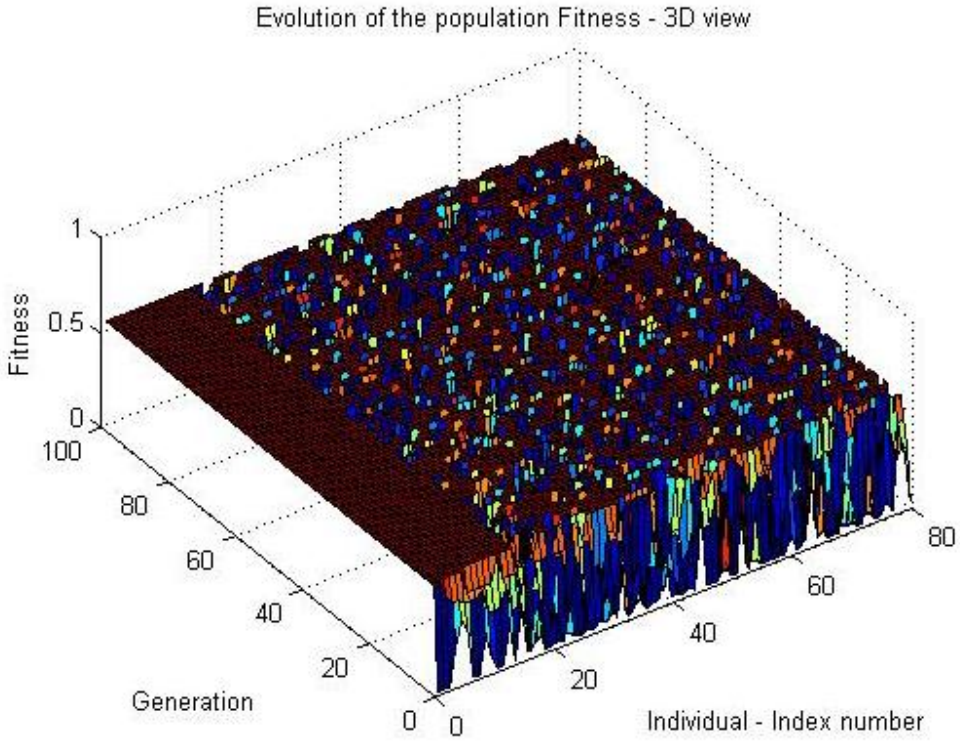


Figure 30: Fitness landscape of an experiment found to search too narrow.

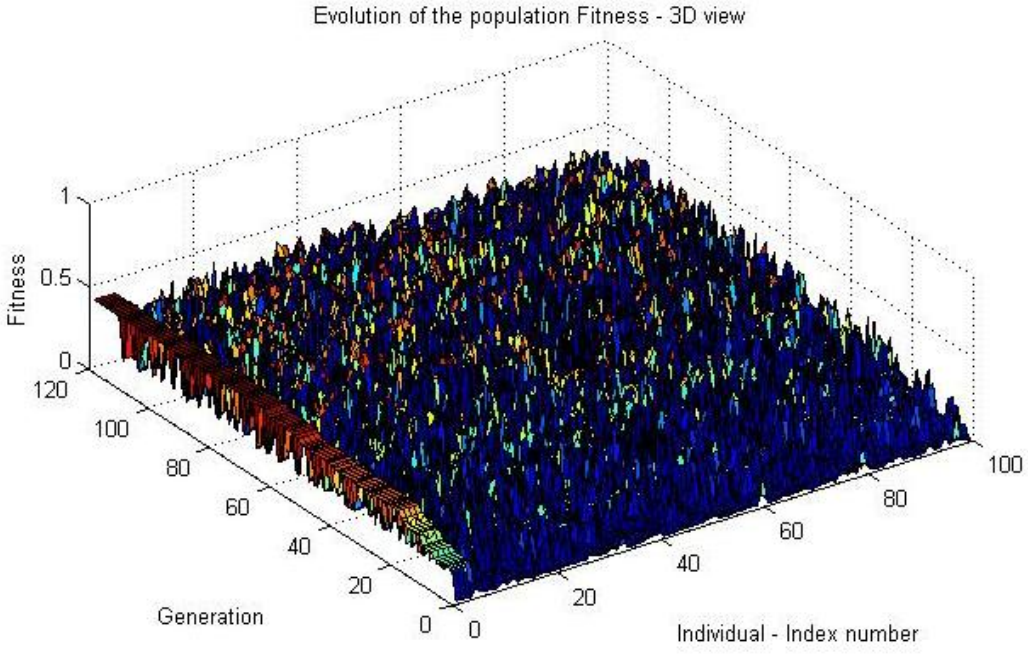


Figure 31: Fitness landscape of a population showing gradual increase in the general fitness as well as a relatively wide search.

This trend also provides a view into how early or late an increase in fitness occurred. Early increase in fitness followed by a flat plateau of fitness values through the generations, could

signify that the population got stuck in a local optimum, requiring alternate search parameters. Late identification of a model could indicate that more time could have produced a better solution in the newly identified direction in the solution space. An example of a late peak in evolution of the solution is depicted in Figure 32 below. By using the predefined population functionality, this population can be explored further. This trend was used to navigate the experimentation process, especially for the FD SID experiments.

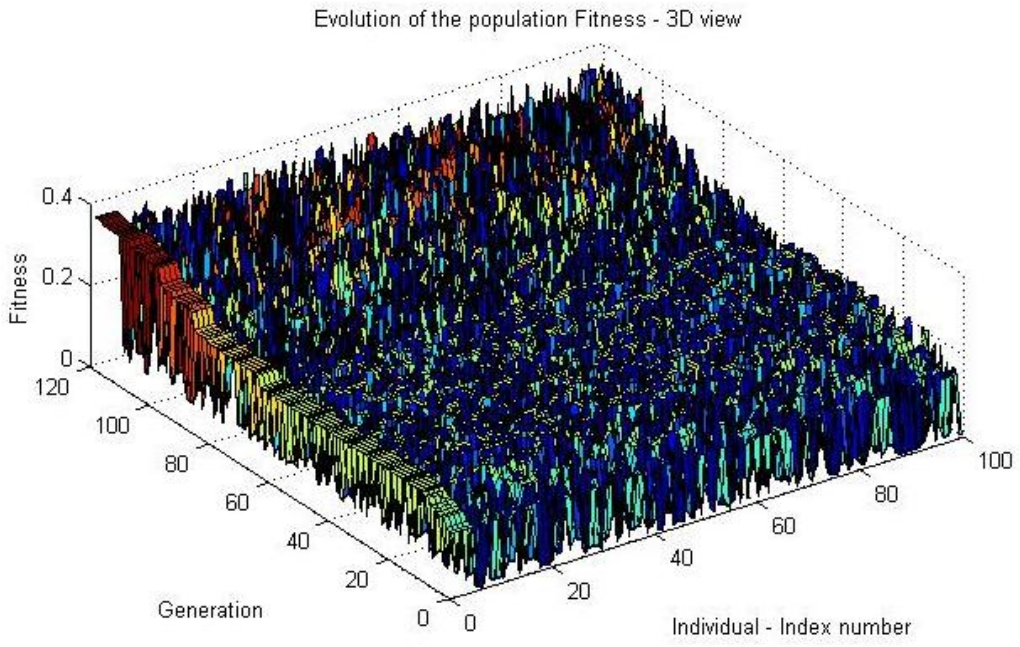


Figure 32: Fitness landscape for an experiment where the algorithm found a more successful direction in the solution space shortly before terminating the run.

7.4 GPOls Algorithm Benchmarking

Performance of the GPOls toolbox against another GP algorithm is measured. The choice of using the GP SID method is also reviewed by comparison to linear ARMA models identified. Both methods make use of the reconstructed latent variable timeseries.

7.4.1 Benchmark against Discipulus ® GP

The accuracy and ability of the GPOls toolbox was proved in previous research (Madar et al., 2005) and is also accepted and used by the Matlab® community . The same approach was used successfully by Coelho and Pessôa (2009). Nonetheless the toolbox performance is measured against Discipulus ® GP software as a benchmark. Discipulus ® is a

commercially available GP SID package and thus seen as a good tool to benchmark against.

The pre-processing of the dataset is the same as in the GPOIs experiments in that the same latent variable reconstruction was used. This was done for both the hot gas generator and the flash dryer units, for both AMI and autocorrelation delay parameters. Four experiments, one for each process and each latent variable set, were completed with Discipulus. Each experiment was repeated 40 times due to the stochastic behaviour of a GP. The default Discipulus ® GP parameters were used. The experiments and the outcomes of the best models, chosen according to the validation MSE, are presented in Table 18 below. The best models found by the GPOIs algorithm, discussed in the next chapter, are included for comparison.

Process	Latent Variable Delay Parameters	Discipulus Validation MSE	Discipulus Validation R^2	GPOIs Validation MSE	GPOIs Validation R^2
Flash Dryer	AMI	0.617	49.2%	0.620	42.3%
	Autocorrelation	0.640	47.3%	0.622	43.6%
Hot Gas Generator	AMI	0.078	86.6%	0.151	81.7%
	Autocorrelation	0.052	91.0%	0.133	76.7%

Table 18: Discipulus Lite ® GP modelling results compared to GPOIs algorithm results as a benchmark of the GPOIs algorithm performance

Direct comparison of the MSE values in the table above, indicate that the Discipulus ® software performance is better than the GPOIs algorithms, for the HGG unit, for both AMI and autocorrelation delays. Analyses of the validation MSE values obtained across all the repetitions of the experiments indicate weaker performance of the GPOIs algorithm, with a weaker best model, as well as a wider variation of poorer performing models across all the repetitions of the experiments for the HGG. This is indicated in Figure 33 below.

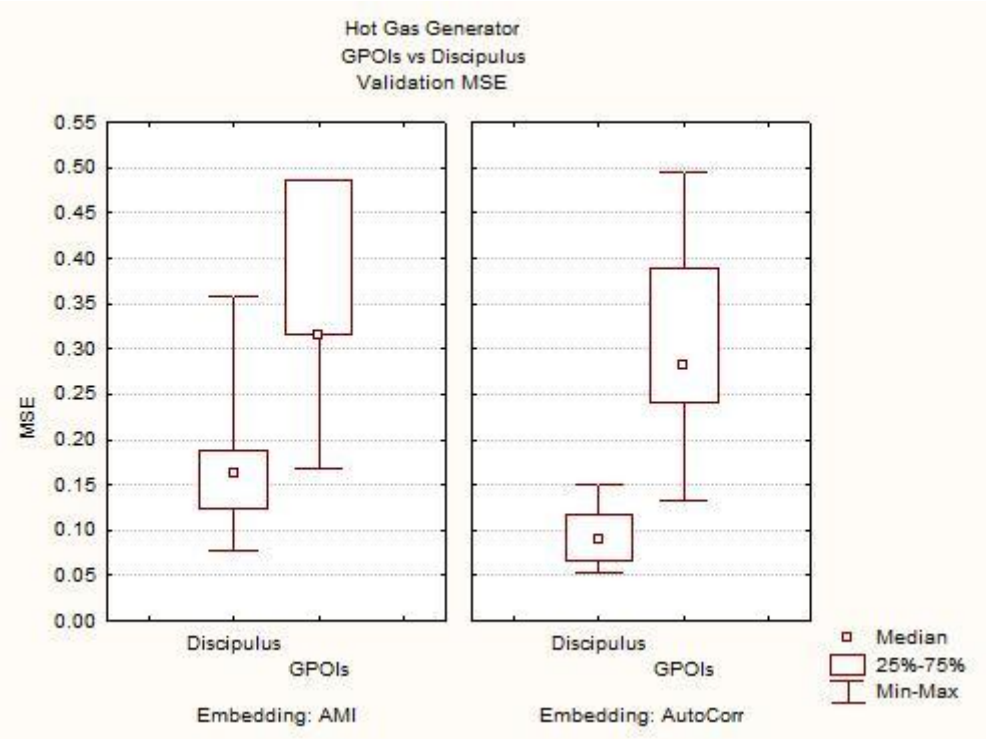


Figure 33: Comparison of Discipulus ® and GPOIs GP algorithm performance for the Hot Gas Generator according to the validation MSE. The GPOIs algorithm performed weaker according to solution spread over the experiments, as well as best model identified.

The flash dryer indicates a larger degree of overlapping performance, with the lowest MSE values being close together and obtained by either one of the algorithms. The wider range of MSE values for the GPOIs algorithm indicates a less efficient algorithm finding fewer good solutions than Discipulus ® . This is indicated by the wider spread of the 2nd and 3rd quartiles, as well as the larger range of values obtained by the GPOIs algorithm in Figure 34 below. The poor overall performance by both algorithms for flash dryer models identification, indicate possible missing process information.

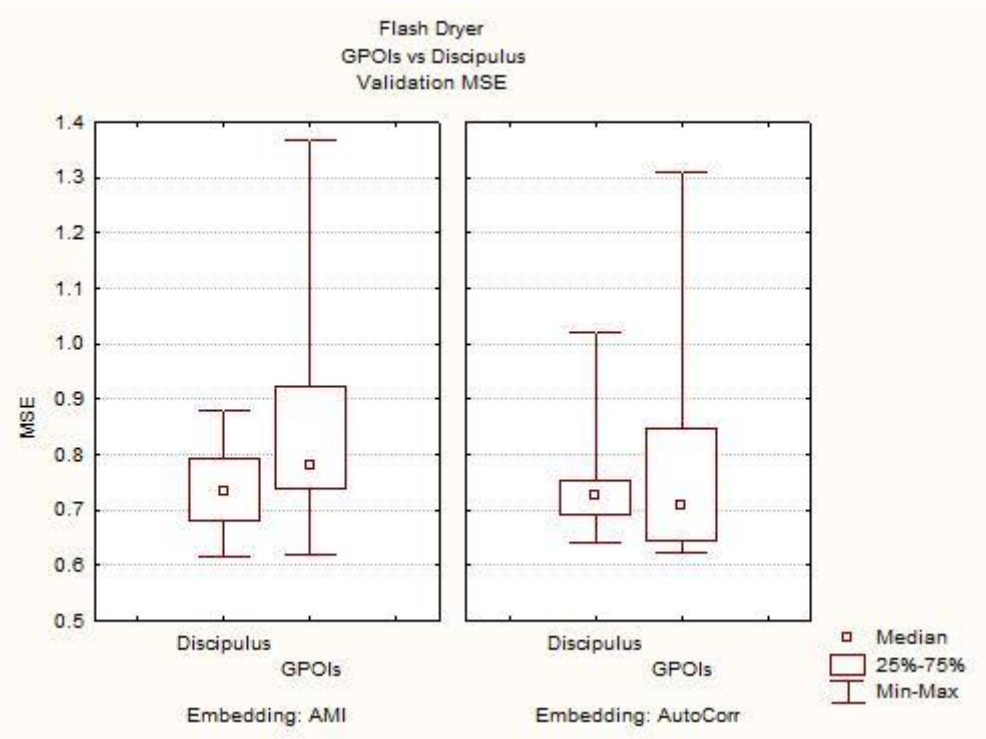


Figure 34:Comparison of Discipulus ® and GPOIs GP algorithm performance for the Flash Dryer according to the validation MSE. The GPOIs algorithm indicates a larger variation in performance of solution obtained than the narrow spread of the Discipulus ® results.

The conclusion is made that the GPOIs algorithm found solutions for the HGG operation, comparative to the solutions found by Discipulus®. The spread of solutions identified by Discipulus is tighter, but the best models identified by GPOIs is comparative to the Discipulus results.

The poor performance of both algorithms in identifying the flash dryer model, indicates that there is some information missing in the model inputs space. This could be either due to incorrect latent variable construction, or missing process variables. From literature, the latter is expected to be the reason.

7.4.2 Benchmark against Linear (ARMA) Models

One of the cheapest models to construct is a linear model. It is thus a good benchmark to measure if the intensive calculations required for any SID experiment are worth the effort given the modelling results and information extracted from the timeseries. Linear multiple input single output (MISO) models were constructed for both the FD and HGG units using

both the AMI and autocorrelation delay parameters. The validation statistics are included in Table 19.

Table 19: The GPOIs identified models are s better, according to validation statistics, when compared to the identified ARMA models.

Process	Latent Variable Delay Parameters	ARMA Validation MSE	ARMA Validation R^2	GPOIs Validation MSE	GPOIs Validation R^2
Flash Dryer	AMI	0.781	22.0%	0.620	42.3%
	Autocorrelation	0.625	37.8%	0.622	43.6%
Hot Gas Generator	AMI	0.153	74.9%	0.151	81.7%
	Autocorrelation	0.183	69.9%	0.133	76.7%

The models identified by the GPOIs algorithms are superior to all the linear models identified. This difference is clear when looking at the FD model identified with AMI delay, as well as the HGG model identified with autocorrelation delays. In these cases both the MSE and R^2 values indicate the advantage of using the GP method. The difference is however small when looking at the FD model with autocorrelation delays and the HGG model with AMI delays. The MSE values slow a very small improvement. The R^2 values however indicate a larger improvement.

It is concluded that the GP algorithm, and resulting search in the nonlinear solution space of unknown model structures, is preferred as it results in better models according to validation MSE and R^2 values.

The coefficients and variables, used in each of the ARMA models fitted, are included in Appendix F.

7.4.3 Benchmarking Conclusion

Although Discipulus ® performance is better for the FD operations and similar for the HGG, it was decided to continue with the models identified by this toolbox, seeing as the models in Discipulus® Lite is not easily accessible and cannot be incorporated into the current CSense simulation environment. Closer inspection as to why Discipulus ® performs better was not done because the focus is on the concentrate drying process on not the GP modelling toolbox and technique. It is recommended that this comparison be done in future

research to contribute to upgrading of the GPOIs approach resulting in more successful modelling of both processes.

The results from GPOIs outperformed the ARMA models identified. The comparison indicates clearly that the GPOIs results are better in two of the four cases. The other two cases are much closer, but with the GP approach still in performing better. It is however expected that the benefit will be much more clear cut when using a nonlinear approach with adaptive model structure, given the expected nonlinearities expected in the drying processes.

Preliminary conclusion can be made with regards to the datasets available. The poor performance of both GP algorithms and the ARMA modelling, for the flash dryer modelling exercises indicate possible lack of information in the flash dryer datasets to successfully model the process, whereas the hot gas generator modelling results indicate more representative datasets and results. This conclusion will be discussed further in the next chapter.

7.5 Section Conclusions

The following conclusions were made for this section:

- The default parameters supplied in literature resulted in very poor models. An adjusted set of specific GP parameters, resulting in the best initial solutions, were identified and used as the base for all experiments;
- The square root function with bias is preferred over square root of an absolute value based on model fit results;
- Despite the fact that the most basic functional set [+,-] resulted in one of the best models for the HGG, the more complex functional sets resulted in improved models for the HGG and FD;
- No functional set could find a valid flash dryer model.

- The addition of defining the predefined population was crucial for investigating the flash dryer operations. The predefined population was also crucial to the success of the HGG model, but to a lesser degree;
- The adjusted fitness function enabled the use of the square root operator with bias;, and was successful;
- The landscape of population fitness values assisted in identifying the limits in GP experiments by providing a visual tool for assessing the GP run;
- The GPOls results are better than the identified linear ARMA models, indicating that using the GP approach will result in added model accuracy. This comparison does not take into account the effort required in terms of time, or the size of the improvement in model accuracy;
- The GPOls toolbox performed poorer than the Discipulus ® GP tool. It was however decided that the difference is acceptable and that the results from the GPOls toolbox can be used for the remainder of this research. Investigation into reasons for the Discipulus algorithm to perform better, was not done; although the functional set used inspired inclusion of advanced functions into the GPOls functional sets.
- A possible lack of process information in the flash dryer timeseries is identified seeing as both linear ARMA and Discipulus resulted in poor flash dryer models, together with the initial GPOls. This is a preliminary conclusion which should be verified.

Chapter 8 Results: Modelling of a Filter Cake Drying Process with Genetic Programming

Two modelling exercises were initiated, one for each of the two process units, in an attempt to find the process section best suited to modelling and control. As experiments continued and the nature of the datasets were exposed by the system identification results experimentation parameters were adjusted, validation and training data changed and various further experiments were attempted in order to obtain a model or set of models which can be investigated for use in MPC. The results of the data preparation and SID steps are discussed in this section. At the end of this section it will be clear if a model could be identified from historic process data using the GPOls algorithm together with the identified delay parameters. It will also be clear which section of the process is better suited for modelling and, hence, control.

Note that during this discussion the fitness values are not used to compare solutions from different experiments, seeing as it is GPOls specific. The R^2 and MSE values are included for discussion.

8.1 Flash Dryer

This section will specifically discuss the results for the modelling exercises for the flash dryer unit. The logical progression of how the investigation of the datasets developed is set ou in this section.

8.1.1 Dataset Analysis

From the collection of subdivided datasets, resulting from the data cleanup and reduction exercise, three were chosen for the FD model identification. The three datasets are named “dataset 1”, “dataset 2” and “dataset 3”. The numbering was arbitrarily assigned during data preparation according to the length of the datasets, with dataset 1 being the longest. The choice and discussion regarding the filtering of all the datasets, and eventual choice of these three datasets, is discussed in Appendix B.1. The choice between these three

remaining datasets are discussed in this section, focussed on surrogate data comparison results.

The flash dryer dataset also indicated various process states in the data. These were isolated and investigated. Results are included in this section.

8.1.1.1 Process States Identified

From the trends of the data and initial flash dryer modelling results, three process states could be identified. These were named, for purposes of this study, as the following:

i. Idle state;

ii. Anomaly state; and

iii. Normal Operation state

Although the anomaly-state might be confusing seeing as an anomaly is normally an unforeseen and unrepeated occurrence, it was named such through this study from the start and then it was decided to keep the naming used initially.

These states are present in FD dataset 2 and displayed in the following trend.

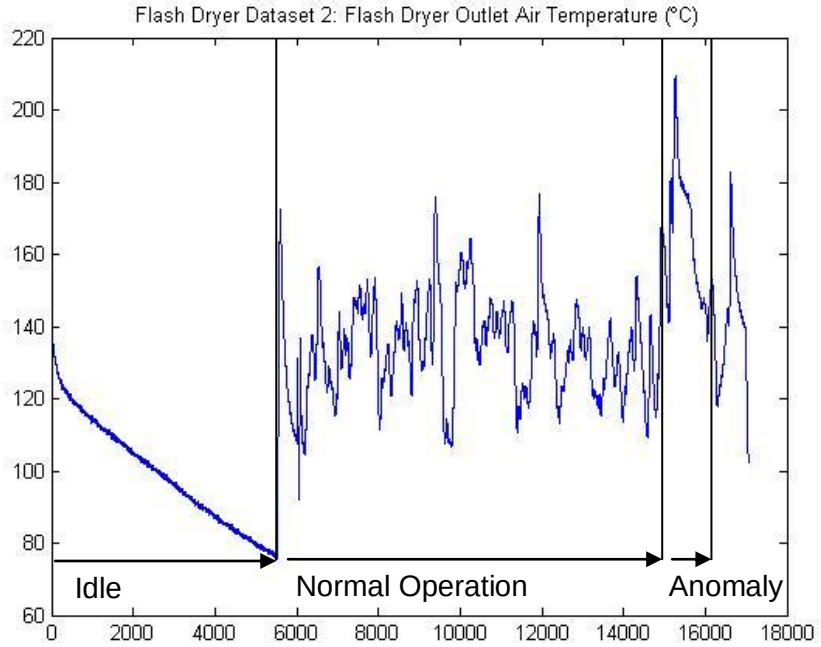


Figure 35: The 3 Identified Process States in Flash Dryer Data, Dataset 2, from left to right: Idle state, Normal Operations and Anomaly states.

It is clear that the idle-state consists of a gradual, smooth decline in the process output. This drop, first noted in Figure 19, seems suspicious and could be ascribed to a moving average not receiving inputs, or another unforeseen data collection issue. No clarity on the situation is available and it will henceforth be seen as a process idle state.

The normal process is, from process knowledge, supposed to run around the 120°C set point. It can be seen, in Figure 35, that the output is running above this most of the time, but contains many process spikes and drops, indicating on dynamics which could possibly be captured in a model.

The anomaly-state is recognised by the steep spike and gradual, but semi stepwise decline in outlet temperature. The process does not seem to be out of control, but the reason for the spike should be recognised for the model and controller to be able to predict it. The anomaly state was distinguished during initial system identification exercises where the model output indicated inability to track this section, compared to the rest of the timeseries.

The following table indicates the presence of these states in each of the three chosen datasets.

Table 20: Occurrences of identified flash dryer process states

Flash Dryer Dataset	Idle State	Anomaly State	Normal Operation
Dataset 1		x	x
Dataset 2	x	x	x
Dataset 3	x		x

Experiments for the identification of the FD model will commence with the inclusion and exclusion of these process states in the training data. The attempt is made to see if these states add to model dynamics.

8.1.1.2 Visual Inspection of dataset

Datasets 1 and 3 for the flash dryer are investigated visually. These two datasets were used during system identification.

Figure 36 below contains the two input, one process output and two controller state variables for dataset 1 of the flash dryer. Prominent in this dataset is the lack of variation in hot gas generator outlet air temperature. This is due to the HGG outlet air temperature being under IMC control, limiting the variation available, the system identification effort continues, due to lack of alternative data. It is understood that the model will only be valid for this area. In the fact that a model is identified, this model should be able to be used for control, as long as the IMC for the HGG is operational and within the same operating set point. The lack of variation in the HGG outlet air temperature also the ability to extract information. except during the two occasions, highlighted, when the current HGG APC controller is off. During these times the HGG outlet air temperature starts to oscillate. It is also during these regions where the anomaly state occurs. However, closer inspection to the case marked *Anomaly 1*, shaded yellow, in the figure, indicates that the increase in the flash dryer outlet air temperature is before the controller is switched off, and coincides closer with the stoppage in concentrate feed than with the change in controller status. This is the same in the case marked *Anomaly 2*, yellow shaded area on the right of the figure. It is preconceived that the controller state does not cause this anomaly, but rather the drop in the concentrate feed to zero. It is unclear if the coal feed caused the anomaly, or if the anomaly occurred due to an unmeasured variable or external input and the operator stopped the coal feed accordingly.

Due to lack of any physical on-site process input, the investigation will continue assuming that there is enough information in the input variables to predict these dynamics. This will however have to be investigated by the GP procedure.

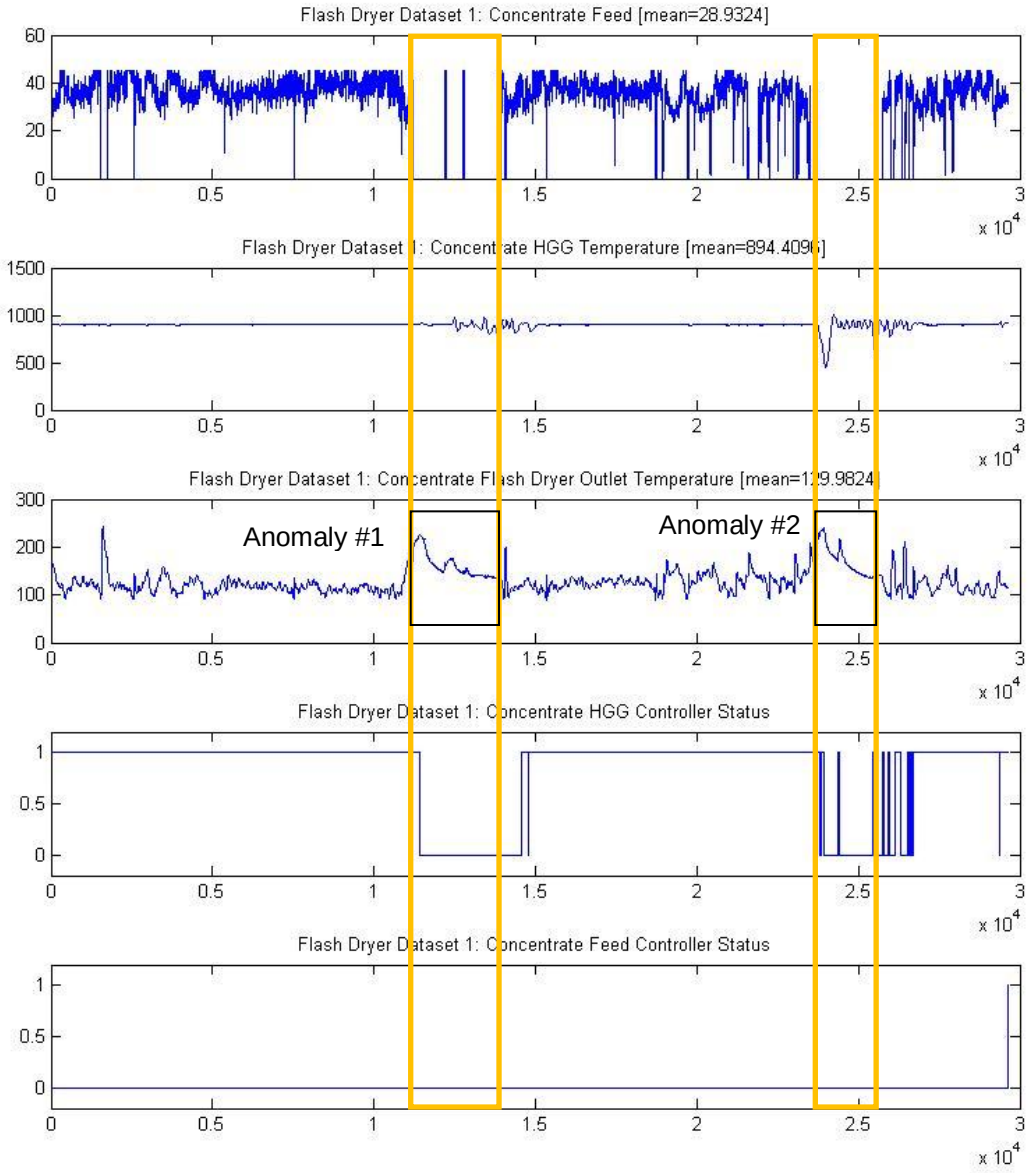


Figure 36: Trends of variables for flash dryer dataset 1. The Anomaly states are indicated with the yellow shaded areas. It is unclear what is causing this spike in flash dryer outlet air temperature.

Dataset 3, in Figure 37 below, contains the largest normal operating conditions with feed stoppages. The HGG outlet air temperature contains larger variation as well. Prominent in dataset 3, is the presence of an idle state, shaded in the figure. The idle state is recognised by the declining outlet air temperature. During this time, the flash dryer feed is zero and some increased activity in the HGG outlet air temperature is visible.

This visual inspection indicates a possible problem with including the idle state in the system identification data: The feed shows no variation, and the inlet air temperature shows

highly oscillating behaviour. Visually these inputs do not seem viable for a gradual decreasing model output. It is possible that the system identification procedure might only choose the previous process out as input. Including the idle state in system identification might thus be a superfluous exercise. These sections of idle and normal process states from this dataset should however be investigated by the GP system identification procedure.

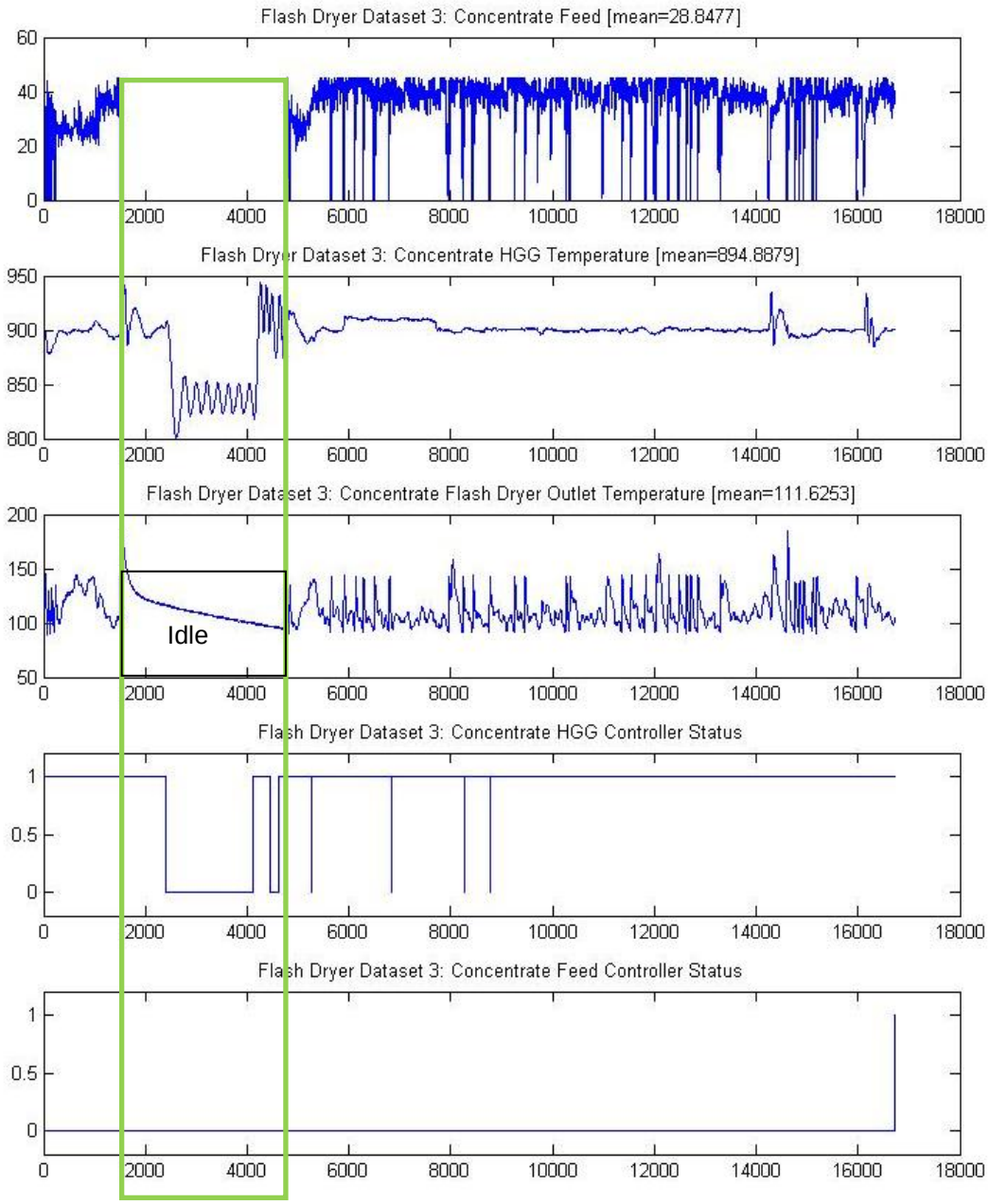


Figure 37: Trends of variables for flash dryer dataset 3. The shaded area identifies the process idle state. No clear reason for the idle state can be recognised and seems more like a controller state or data collection rule.

The next section will distinguish between the datasets based on the nature of the information available in the dataset. The aim is to identify the dataset best suited for modelling.

8.1.1.3 Choosing the Most Representative Dataset by Means of Surrogate Data Comparison

The surrogate data analysis indicated that only dataset 3 displayed the presence of deterministic information which could be easily extracted by a SID procedure. The investigation indicates similar separation from the surrogate data for the flash dryer outlet air temperature for dataset 3 using delay parameters identified by AMI and autocorrelation. This is seen in Figure 38 and Figure 39. This is done with the process idle-states present in the dataset.

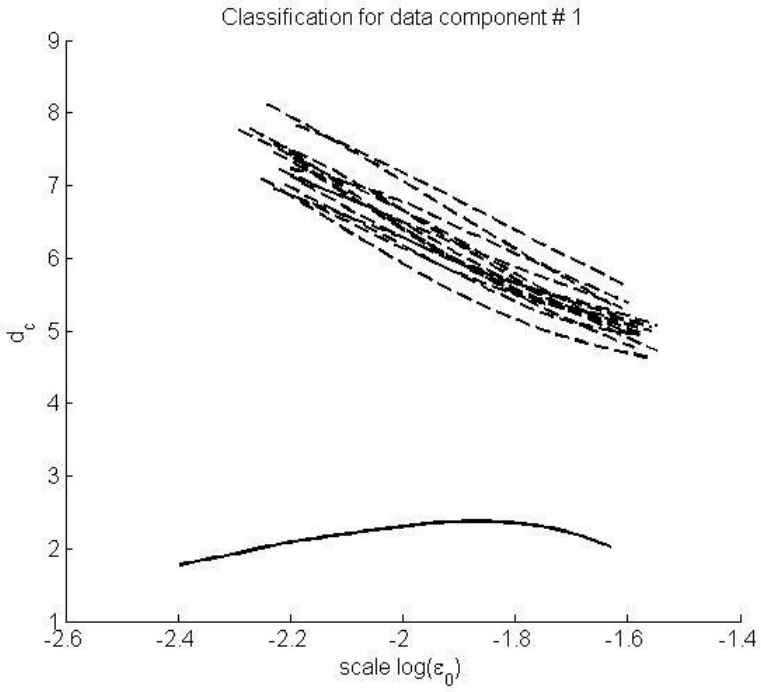


Figure 38: Surrogate Data Classification of the FD outlet air temperature for , with IDLE state present, for delay parameters identified by autocorrelation- 168x5 (*delay* x *embedding dimension*)

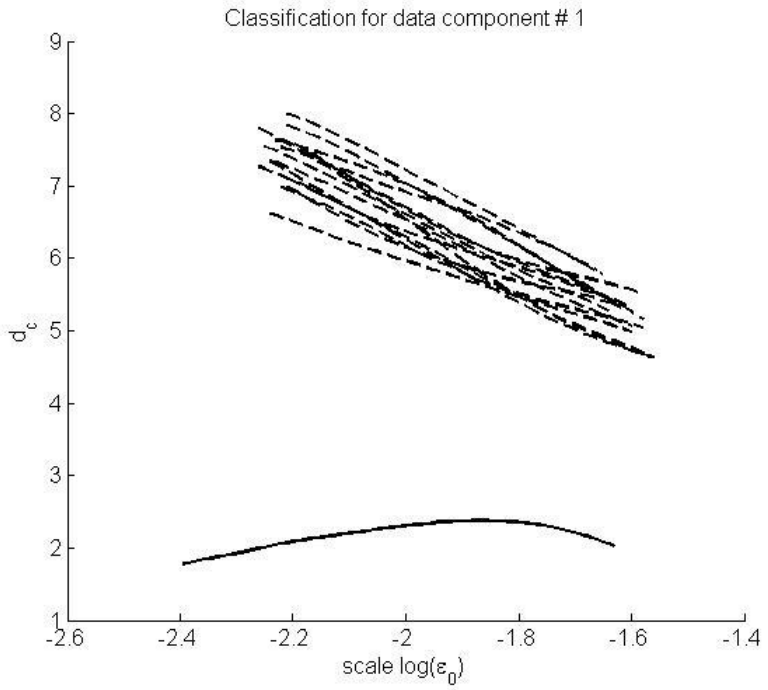


Figure 39: Surrogate Data Classification of the FD outlet air temperature in dataset 3, with IDLE state present, for delay parameters identified by AMI - 65x5 (*delay* x *embedding dimension*)

The inclusion of the idle state in modelling could however be superfluous resulting from the discussion in the previous section – the idle-state might not be modelled accurately due to possible lack in process information. Inclusion of the idle-state in modelling will thus only result in stronger inclusion of the least lagged version of the process outlet variable, as indicated in the previous discussion. Dataset 3 with the idle-states removed from the timeseries, and new delay parameters calculated accordingly, indicate weaker separation from the surrogate data, although the separation is still strong enough to distinguish deterministic information from the timeseries as seen in Figure 40 and Figure 41 below. Both the included and excluded idle state datasets should be investigated for system identification based on the clear surrogate data separation.

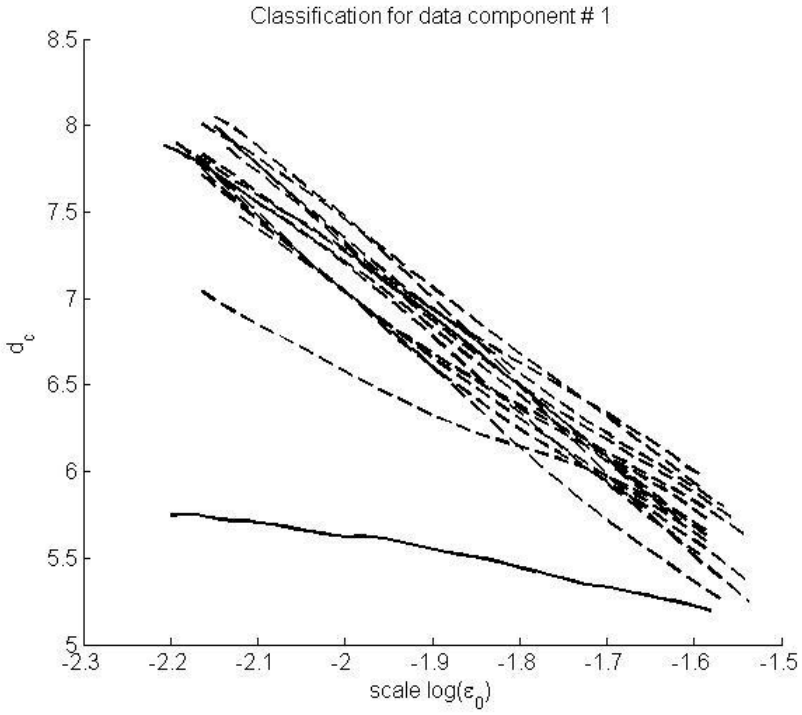


Figure 40: Surrogate Data Classification of the FD outlet air temperature in dataset 3, with IDLE states removed, for delay parameters identified by autocorrelation - 109x5 (*delay* x *embedding dimension*)

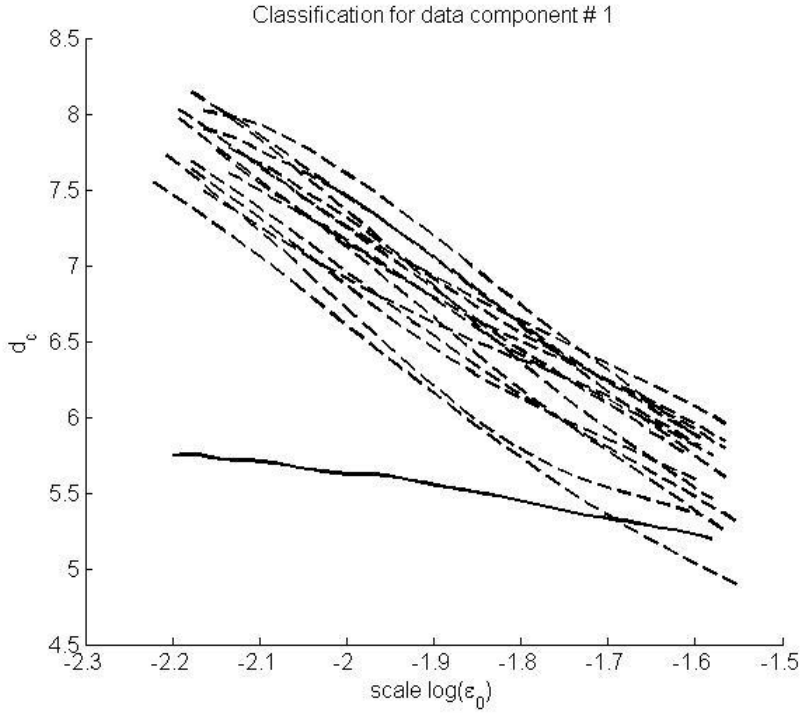


Figure 41: Surrogate Data Classification of the FD outlet air temperature with IDLE states removed for delay parameters identified by AMI - 65x5 (*delay* x *embedding dimension*)

Datasets 1 and 2 indicated poor separation from the surrogate data. (See Appendix C – Process Output Timeseries Analysis for a detailed discussion.) The use of these datasets for training would probably not result in good models. Seeing as datasets 1 and 2 contain the anomaly state, and are found to contain only stochastic information, it could be deduced that the presence of the anomaly state is causing this inability to differentiate between the stochastic and deterministic information in the dataset. However, a surrogate data analysis done on dataset 2 with the idle and anomaly-states removed indicates that this is not true and that even with only the normal process state isolated in dataset 2, it still contains mainly stochastic information.

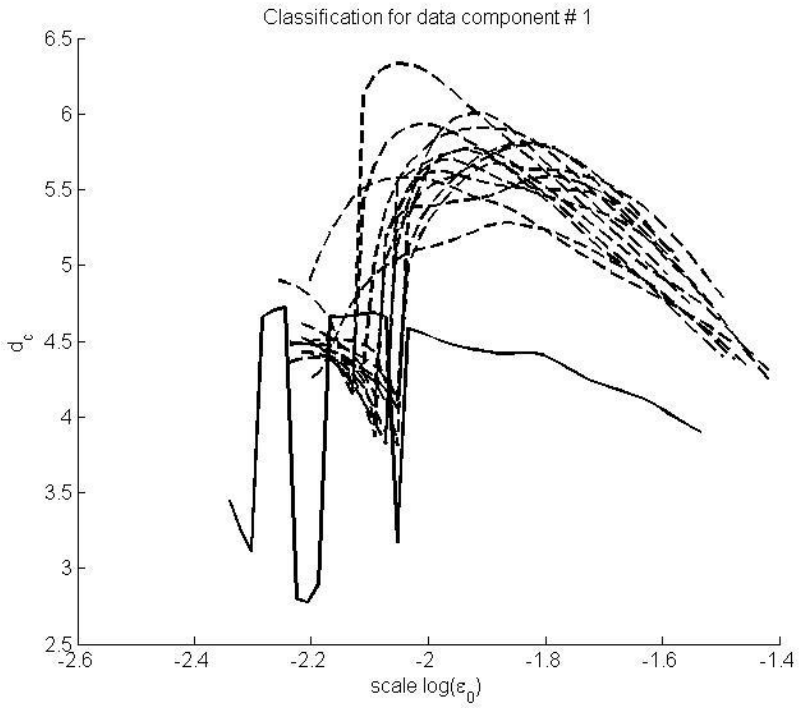


Figure 42: Surrogate Data Classification of the FD outlet air temperature for delay parameters 42x4

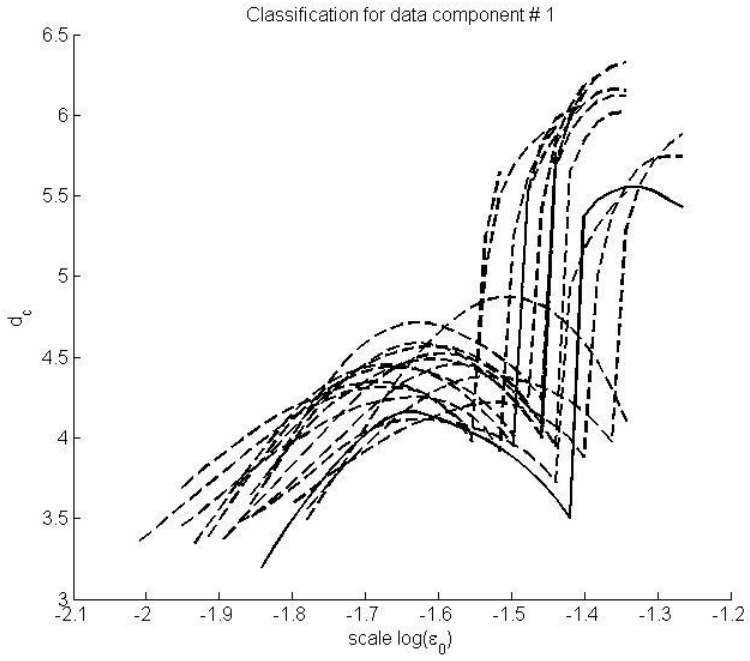


Figure 43: Surrogate Data Classification of the FD outlet air temperature for delay parameters 273x3

It is thus clear that only dataset 3 is suited for process modelling, although the presence of the idle-state is favoured for modelling, which is contrary to what is expected. The need to include the prediction of the anomaly state in the model necessitates the use of datasets 1 and 2, although it is expected, from the surrogate data comparison, that no result will be found from these datasets.

These findings, of including and excluding the anomaly states and using datasets 1 and 2 will be confirmed by the GPOIs algorithm.

8.1.1.4 Correlation with the Least Lagged Process Output

Correlation with the least lagged process output is an indication of how easily the system identification technique will be able to supply a model better than the least lagged process output. This resulting model would thus contain a combination of process inputs and outputs, with inputs required for a control model. The dataset with the weakest correlation to the least lagged process output should obtain a model easier than with the other datasets, depending on the information available in the data.

The following table sets out the datasets, lags per latent variable reconstruction, and the correlation of the one-step ahead process output with the least lagged process output present in the latent variable regressor set.

Table 21: Correlation coefficients of closest lagged process output. Obtained models are expected to have better correlation figures than these, otherwise the least lagged process output will be the best model.

Process	Dataset Number	Output Lag as per AMI (5sec increments)	Correlation Coefficient (AMI)	Output Lag as per Autocorrelation (5sec increments)	Correlation Coefficient (Autocorrelation)
Flash Dryer - Idle States Present	1	82	0.735	623	0.347
	2	41	0.943	405	0.608
	3	65	0.352	168	0.153
Flash Dryer – Idle States Removed	1	82	0.735	623	0.347
	2	56	0.779	365	0.059
	3	65	0.1102	109	0.049

Dataset 3 with the Idle state present, as well as with the idle state removed shows the weakest correlation between the process output and least lagged process output. This is the case for both AMI and autocorrelation delay parameters.

The correlation of 5.9% for dataset 2 with idle-state removed could be ascribed to the high delay of 365 time increments (30minutes and 25seconds) and is misleading as a previous output loses relevance as time goes by.

Comparing the correlation coefficients determined for delays identified by AMI and autocorrelation, indicate that the delays identified by autocorrelation should result in easier identification of models. The longer delays might however be synthetic and not make practical sense, which could be misleading. The choice of delay parameters will be discussed in the next section 8.1.2 Delay Parameters.

From the visual inspection of the datasets it was mentioned that the inclusion of the idle-state will result in stronger influence of the least lagged process output, seeing as the input variables do not have any visual resemblance to the idle-state, leaving the process output to be the only with values which could predict the idle-state output. This comment is reinforced

by the higher correlation between the least lagged output variable and the process output for idle-state present datasets. This holds true for datasets 2 with 94.3% < 77.9% and dataset 3 with 35.2% < 11%. It is thus expected that the idle-state will be modelled with inclusion of the least lagged variable when included in the SID training set. This will not contribute to a model suited for control.

These qualitative conclusions will however need to be investigated by the GP during modelling.

8.1.1.5 Conclusion

The following conclusions are drawn from the analysis of the datasets for flash dryer system identification:

- Three process states could be distinguished visually and will be investigated during system identification;
- The idle and the anomaly-state indicate similar input variable dynamics, but different process output reaction, possibly indicating a lack of process measurements to distinguish the states;
- Dataset 3 is the best suited for modelling purposes as it indicates best separation from the stochastic surrogate data and lowest correlation to the least lagged process output;
- Although the presence of the idle-state increases the deterministic information, the idle-state might favour inclusion of the least lagged process output over manipulated process inputs. This will however need to be proven during system identification using a timeseries with the idle-state present as training dataset; and
- Removal of the anomaly-state from datasets 1 and 2 does not contribute to better separation between deterministic and stochastic timeseries. Modelling with these datasets with the anomaly-state removed will not be investigated.

8.1.2 Delay Parameters used for Latent

This section sets out the most important delay parameters used in the study and indicates the preference between AMI and autocorrelation delay parameters for FD modelling. No real distinction could be drawn, due to the poor SID results for the flash dryer.

8.1.2.1 Delay Parameters Identified

The delay parameters identified for dataset 3, with the idle-state removed, are included and discussed below. This is found to be the best dataset for training and trained the best models, thus it is chosen for discussion over the other datasets. Delay parameters for the other datasets are included in Appendix D.

The delays, identified by autocorrelation, are much higher for the flash dryer outlet air temperature as well as the hot gas generator outlet air temperature.

Flash Dryer Dataset 3 – IDLE states removed:

Table 22: Delay parameters for process variables for Flash Dryer Dataset 3 with IDLE states removed

Process Variables	Method to Determine Delay k	Delay (k)	Number of Delayed Variables (m)
Filter Cake Feed Rate	AMI	16	10
	Autocorrelation	16	10
Hot Gas Generator Output Temperature	AMI	56	6
	Autocorrelation	363	5
Flash Dryer Outlet Air Temperature	AMI	65	5
	Autocorrelation	109	5

The choice of which *m* and *k*, as identified by AMI or by autocorrelation, to use, is not clear. The distinction between the various identified parameters and the inclusion of these variables is investigated by the GP and will be discussed next.

8.1.2.2 AMI vs. Autocorrelation as Identified by the GP

The choice of which set of delay parameters to use was investigated by the GPOIs algorithm in the experiments are grouped (Alphabetically) and compared in Figure 44.

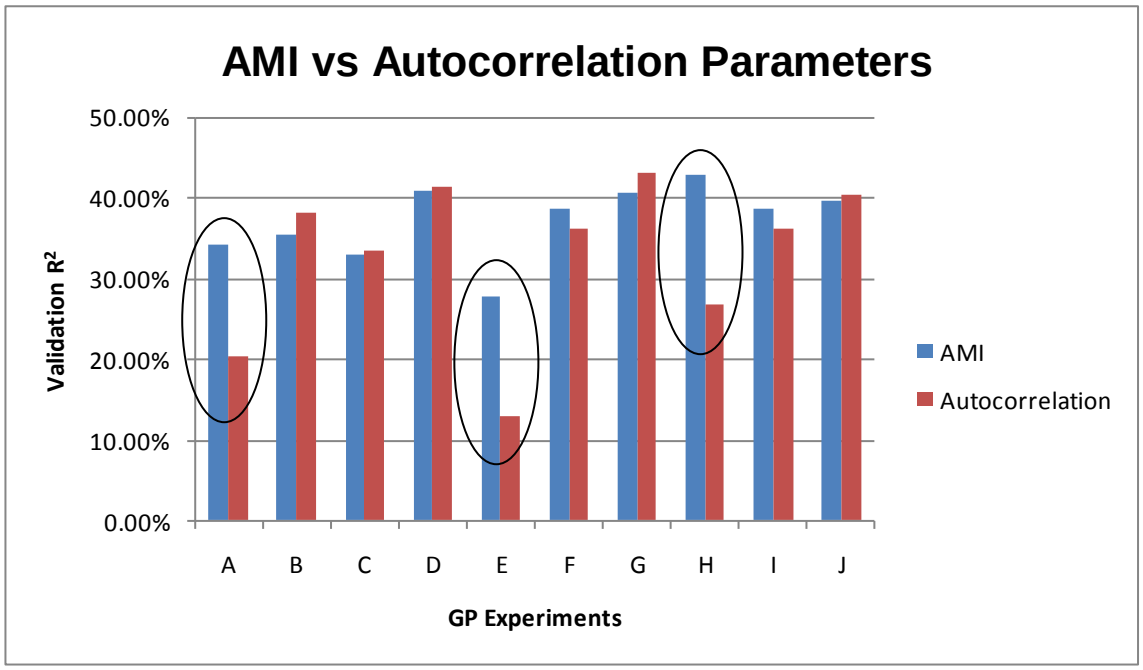


Figure 44: Comparison of Validation R^2 values for comparable experiments with different delay parameters. AMI delay parameters result in more consistent results, with autocorrelation results visibly underperforming on three occasions.

Throughout the trend the AMI and autocorrelation system identification results are similar. There are however three occasions where the autocorrelation parameters visibly underperformed. Deciding which parameter identification method to use for the flash dryer is not clear cut, although the more consistent AMI parameter set performance favours this method. Delays from both methods will be investigated further.

The comparison results for the figure above are tabulated in Appendix G.1.1

8.1.3 Handling of Anomalies and Idle States

The inclusion of all process states in the identified flash dryer model requires the trained models to be exposed to such states. Furthermore it is necessary to validate the solutions against data containing the same process states. A brief discussion regarding the idle state is included in this section. Apart from that section the idle state is omitted from any future discussions or modelling. Only the normal process state and anomaly-state is investigated.

The three strategies followed, for investigating these two states, are:

1. Train models on normal process dynamics and investigate the ability of the model to extrapolate to the anomaly-state;

2. Train models on a timeseries which includes both the normal and anomaly-states to be modelled; and
3. Train models on the normal process dynamics and then retrain the identified models on a second timeseries which includes the “anomaly” process state. In so doing the models are exposed to all dynamics.

The specific datasets used for each of these strategies are set out in Table 23 below.

Table 23: Training strategies, and corresponding datasets, used in an attempt to include the identified process state dynamics in the models

Strategy number	Training Dataset	Validation Dataset	Strategy
1	Dataset 3 (idle-state removed)	Dataset 1	Measure of models identified can extrapolate to include the anomalies
2	Dataset 1	Dataset 3 (idle-state removed)	Train with the anomalies; see if a model can be identified and measure if this model represents normal process operations
3	Dataset 3 (idle-state removed); and then Dataset 1 in a second GP experiment	Dataset 1; and then Dataset 3 (idle-state removed) in a second GP Experiment	Models are trained on normal process dynamics. The identified population is then used as a predefined population for a second, narrower identification exercise where dataset 1 is used for further training.

It was found that none of the approaches was able to include the process dynamics for both the process states. The results for the three approaches are briefly discussed below in the same order as mentioned above.

The fourth subsection below investigates if the inclusion of the process idle-state contributes to the representative process dynamics included in the identified model.

8.1.3.1 Extrapolate Normal Process State Model to the Anomaly State

11 experiments were done using dataset 3 as training set, with validation against dataset 1, as a measure of how well the model can extrapolate to include the unseen anomaly state. The statistics for the experiments, in degrading order of fit according to validation R^2 and MSE, are:

Table 24: Fit statistics for models trained on dataset 3 and extrapolated to validation on dataset 1 as measure of representing the anomaly-state

Model Number	Latent Variable Delay Parameters	Train MSE	Train R^2	Validation MSE	Validation R^2
1	Autocorrelation	0.673	32.9%	2.149	40.6%
2	AMI	0.650	35.0%	2.204	40.4%
3	AMI	0.666	33.4%	2.215	40.0%
4	Autocorrelation	0.784	21.9%	2.269	37.3%
5	AMI	0.630	37.1%	2.350	36.4%
6	Autocorrelation	0.637	36.5%	2.744	24.2%
7	Autocorrelation	0.697	46.2%	3.203	11.5%
8	AMI	0.822	17.9%	3.492	5.4%
9	AMI	0.822	17.9%	3.492	5.4%
10	Autocorrelation	0.600	40.3%	3.878	-7.1%
11	AMI	0.573	42.8%	4.101	-1107.0%

The 3 main findings are discussed by looking at models 1, 4 and 7 in Table 24 above.

Model 1, portrayed in Figure 45, indicates that the anomalies are estimated by a flat lined increase in outlet air temperature. This flat line corresponds to the concentrate feed stationary at zero during this period. The poor fit statistics for training together with the very high MSE value for the validation set indicates that this model is not properly representative of the process.

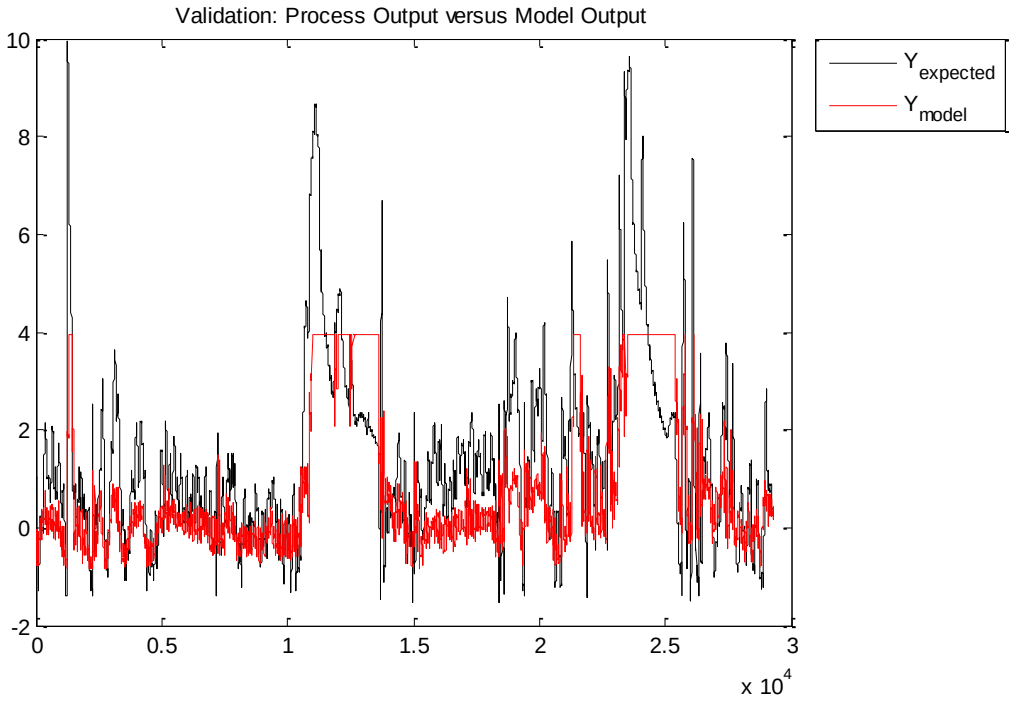


Figure 45: The extrapolation ability of the model trained on dataset 3 to represent the anomaly state results in a flat lines estimation of the process anomaly.

Model 4, Figure 46, represents the anomalies in the process very well, but does not indicate any presence of normal process operating conditions. The lack of representing the normal process conditions is indicated by the poor training R^2 for this model.

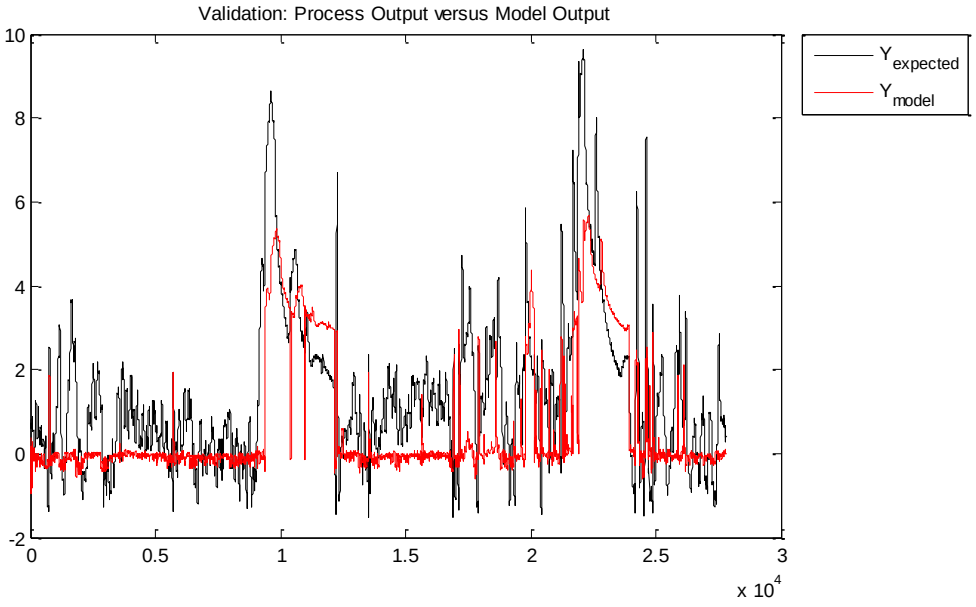


Figure 46: The shape of the anomaly is well represented, largely due to the presence of the historic process output variables. The normal process operating conditions are poorly represented.

Model 7, Figure 47 below, indicates some variation of the output in the anomaly state. This variation is assigned to a stronger presence of the HGG outlet air temperature. The deviation of the HGG outlet air temperature was identified in the visual inspection, in section 8.1.1.2, as another possible cause of the anomaly. The inclusion of this variation due to the HGG outlet air temperature presence results in a very poor R^2 of 11.5%, making the rejection of this model obvious.

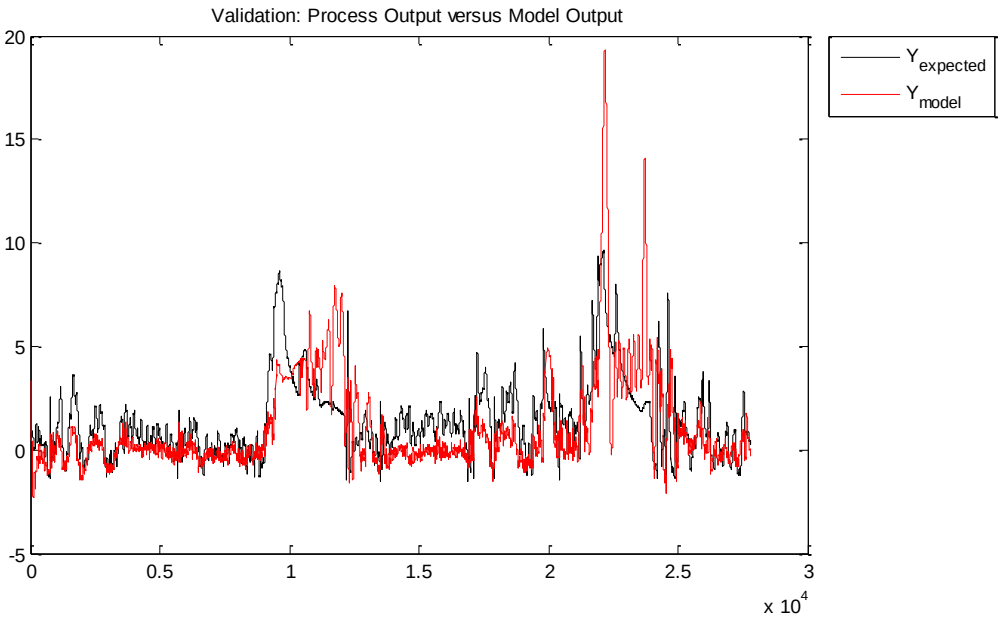


Figure 47: The stronger presence of the hot gas generator outlet air temperature, does not contribute to a more accurate representation of the anomaly.

It is concluded that the anomalies are identified by the sudden drop in concentrate feed, although the drop in feed might be due to the operator's process experience and awareness of the anomaly. The presence of the HGG outlet air temperature does not contribute to the model fit.

The normal process conditions, given the available measurements, is not able to extrapolate to the anomaly state.

8.1.3.2 Train on All Process States

Two experiments were done on dataset 1 as a training set despite the surrogate data comparison indicating a lack of deterministic dynamics. Both experiments resulted in choosing the least lagged process output as model, indicating that the surrogate data

comparison was correct. This approach and resulting models are not discussed further as a least lagged process output model cannot be used as a predictive model used for optimisation of future process inputs.

8.1.3.3 Sequential Training on Different Dynamics

These experiments used models identified from training on dataset 3 as the initial population for the new GP experiment. The new experiment used these populations and exposed them to the anomaly state contained in dataset 1; as depicted in the following diagram:

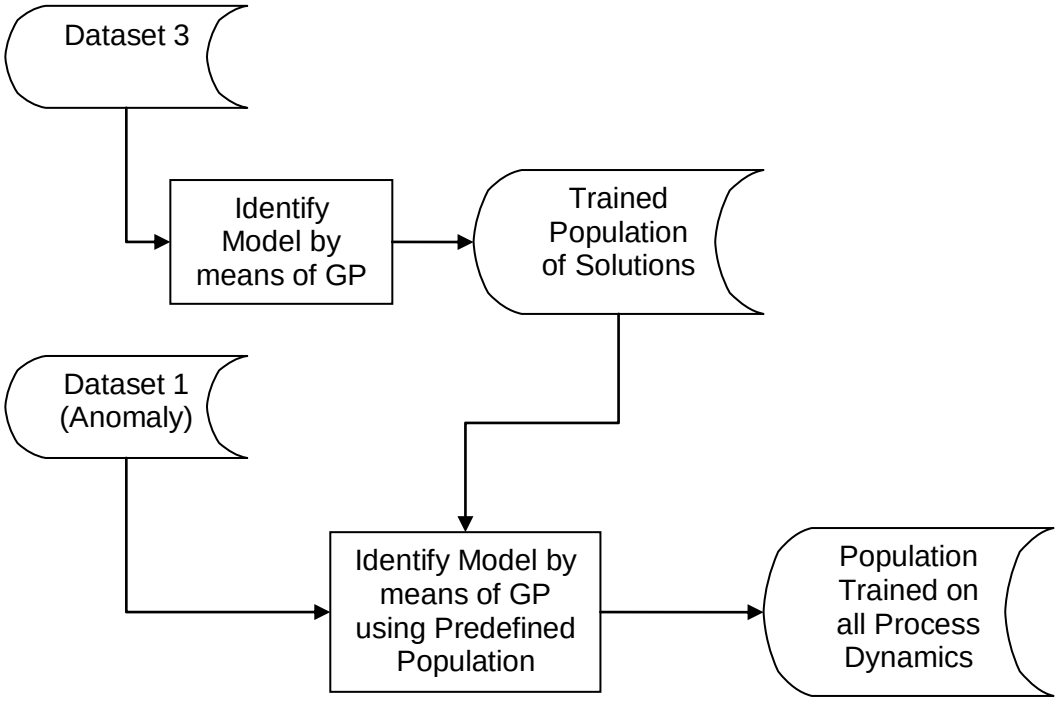


Figure 48: Sequential training method used. The population from the one experiment is used as initial population for the next experiment, where a different dataset is used with the anomaly state dynamics.

Less mutation was allowed, as well as a small tournament selection strategy in an attempt to allow only small adjustments to the population. This was done to prevent the population from evolving into a least lagged process output model or being over fitted on dataset 1 dynamics, losing the dynamics instilled during the training on dataset 3.

The best solutions from the populations resulting from step 1 of the sequential training were:

Latent Variable Delay Parameters	Train MSE	Train R^2	Validation MSE	Validation R^2
Autocorrelation	0.637	36.5%	2.744	24.2%
AMI	0.666	33.4%	2.215	40%

Table 25: Models from the initial populations for the second GP system identification exercise.

Although the fit statistics for the predefined population are poor, it was decided to continue with the next step in the sequential training to gauge the possibility of using this method. The models were trained on dataset 1 and validated using dataset 3. This is done to see how much information the population has lost in the second training. The results were:

Model Number	Latent Variable Delay Parameters	Train MSE	Train R^2	Validation MSE	Validation R^2
3	Autocorrelation	0.310	54.4%	0.314	-14.8%
4	AMI	0.395	60.7%	0.418	-53.5%

Table 26: A sequence of training on normal process dynamics and then process anomalies resulted in similar results for both the AMI and autocorrelation delay parameters

The final models in Table 26 indicate that both parameter sets result in similar results according to training and validation statistics. Investigation of the model structure however indicates that model 4, for AMI delay parameters, resulted in the least lagged process output. This result was obtained across all 40 repetitions of the experiment. The negative R^2 is ascribed to the parameter estimation step which overfits the model to the training data.

The autocorrelation parameter set resulted in model 3 - a model structure other than only the least lagged process variable. This model did however produce a negative validation R^2 on the normal process state represented by dataset 3. Dataset 2 was used as an unseen validation dataset. The fit R^2 is lower than the training statistics, but the MSE value is much better.

Table 27: MSE and R^2 values for validation of model 3 with dataset 2. The R^2 is weaker, but the MSE values are better than the training data.

MSE	R^2
0.274	36.8%

Although the anomaly state is followed (greyed out area in Figure 49) it is noted that the overall fit is still not good enough to represent the normal process dynamics. Furthermore, it is important that the freerun prediction of the anomaly-state be investigated. The model identified contained various lags of the concentrate feed and a lagged version of the flash dryer outlet air temperature. Due to the feeds being zero during the anomaly-state, it is possible that the model structure evolved in such a way that only the lagged process output is taken into account for the anomaly-state, and no accurate prediction during this state will be possible. This is not investigated in this research, as these models are not representative enough for the purpose of a control model.

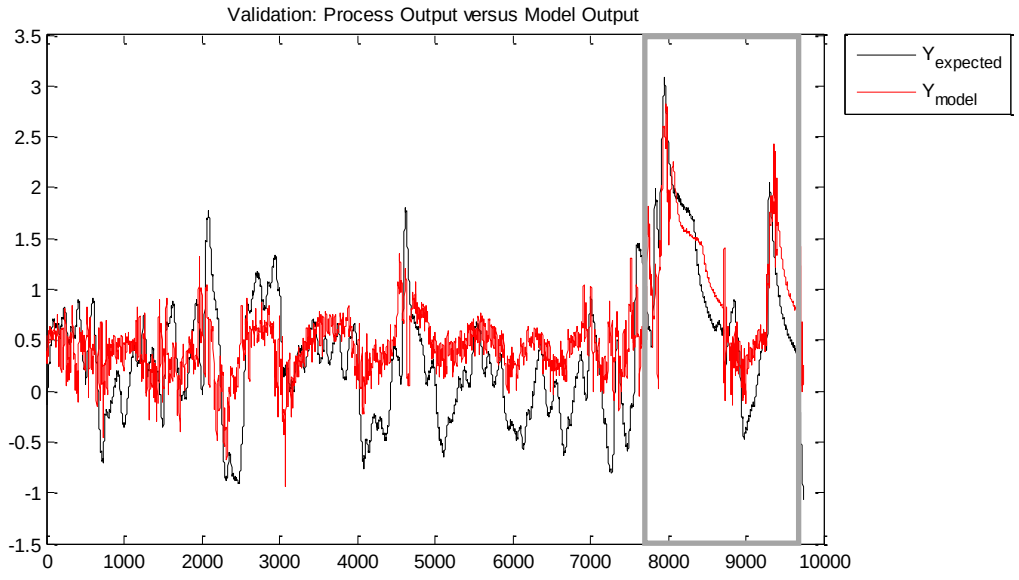


Figure 49: Validation on dataset 2 indicates comparable handling anomaly process dynamics (in the greyed area), but still indicates poor tracking of the normal process dynamics foregoing the anomaly.

8.1.3.4 Idle State

The hypothesis that the idle-state includes process dynamics which could contribute to the model fit, was tested by training on a dataset which includes the idle-state and testing model validity on normal process operation.

Dataset 3 with idle-state included is used for both training and validation, with the first 60% of the timeseries, which includes the idle-state, used for training. This is done for each of the AMI and autocorrelation delay parameters. The resulting model fits for 2 such experiments are:

Table 28: Models, identified when including the idle-state dynamics in the training dataset, does not result in models able to represent the process well enough.

Model Number	Latent Variable Delay Parameters	Train MSE	Train R^2	Validation MSE	Validation R^2
1	AMI	0.456	54.3%	0.897	13.0%
2	Autocorrelation	0.351	44.5%	0.839	22.7%

From a comparison of the validation and training R^2 values, it is clear that the dynamics identified from the idle process state together with the normal process state does not represent the normal process dynamics well.

The conclusion is that the idle-state does not contribute to the dynamics of the model in representing normal process dynamics. It is recommended that the occurrence of the idle-state should be investigated in terms of causes of the idle-state, unmeasured process variables and data collection procedures used.

8.1.3.5 Conclusion

It is clear that training on dataset 1 only, provided no usable results. This corresponds with the findings of the surrogate data comparison hypothesis earlier.

The models trained on normal process dynamics are able to extrapolate and identify the anomalies, although the anomaly is not tracked very well. It is also unclear if the concentrate feed drop causes the anomaly or is caused by an operator due to the occurrence of the anomaly, as the same input variable dynamics are seen during the process idle-state. More information regarding the occurrence of both these states, as well as additional measurements, is required to accurately include these states in a model.

The strategy of sequential training: first on dataset 3, then using the resulting population to train further on dataset 1 provided no usable results, although the training did represent the anomaly and process dynamics at accuracies similar to other models found. Two problems are however raised, although not further investigated:

1. The anomaly-state might only be represented by the least lagged process output; and
2. The normal states for the various datasets differ by a certain bias.

From these it can be seen that some process information is missing, which could include some unmeasured process disturbances. Further investigation into identification of missing information and possible disturbances for the flash dryer is required and recommended. This could be enabled through inferential sensors ('soft sensors') predicting process measurements in the absence of measurement instrumentation, or by inclusion of missing measurement instrumentation.

8.1.4 The Best Model Obtained: Flash Dryer

Contrasted to the strategies discussed in the previous section is the one displayed in Table 29 below, where only the normal process dynamics are being investigated. This follows the findings that no representative model could be obtained for inclusion of the anomaly-state; as well as the inability of the idle-state to contribute to model accuracy.

Table 29: Training strategy for models not including the Idle or Anomaly process states

Training Dataset	Validation Dataset	Strategy
Dataset 3 (first 60%)	Dataset 3 (last 40%)	Models for normal process dynamics are identified and validated using separate sections of dataset

This exercise resulted in the best models based on fit statistics for both the training and validation sets.

The assumption is made that the process states can be identified and the use of a controller can rely on only the modelling and identification of the normal process conditions. Control

strategies during the process states other than normal operation are thus ignored for further research, and the focus is only on modelling and controlling the normal process dynamics.

The top 4 models trained on the first 60%, and validated on the second 40%, are discussed further. All these models contain nonlinear model structures. The models are selected based on validation MSE values. The reader is referred to Appendix G - – System Identification Experiment Outputs for the model structures and all the results. The fit statistics are reproduced in Table 30 below.

Table 30: Results for the top 4 models using only dataset 3 as training and validation sets

Model Number	Latent Variable Delay Parameters	Train MSE	Train R^2	Validation MSE	Validation R^2
1	AMI	0.403	49%	0.620	42.9%
2	AMI	0.430	47.5%	0.622	42.7%
3	Autocorrelation	0.468	37.4%	0.622	43%
4	AMI	0.391	52.3%	0.633	41.7%

Across all four models the validation statistics are similar, with varying training statistics. It is however clear that the statistics are still too weak to confidently say that a representative model has been found as only 40-50% of the variation in the process can be explained with the models identified. Model 1 was investigated in an effort to identify any possible shortcomings or any clue to why the modelling failed.

Investigation of the residuals for model 1 indicates that there is no strong correlation between the residuals and any variable in the terminal set. There is thus limited linear information to be extracted from the possible model inputs, and no clear indication of overfitting. The nonlinear combinations are not investigated explicitly and it is assumed that the most relevant of these combinations would be included by the GP search algorithm.

Table 31: Correlation coefficients between the terminal set and the residuals as a measure of the information remaining in the terminal set

Latent Variable	Correlation with Residuals
FDFeed(k-0)	-11%
FDFeed(k-16)	0%
FDFeed(k-32)	-9%
FDFeed(k-48)	-14%
FDFeed(k-64)	-9%
FDFeed(k-80)	2%
FDFeed(k-96)	-6%
FDFeed(k-112)	1%
FDFeed(k-128)	-1%
FDFeed(k-144)	-10%
FDFeed(k-160)	-7%
HGGTemp(k-0)	2%
HGGTemp(k-56)	0%
HGGTemp(k-112)	5%
HGGTemp(k-168)	5%
HGGTemp(k-224)	7%
HGGTemp(k-280)	8%
HGGTemp(k-336)	4%
FDTemp(k-65)	-2%
FDTemp(k-130)	-2%
FDTemp(k-195)	-1%
FDTemp(k-260)	-10%
FDTemp(k-325)	-1%

Visual analysis of the trend of this model indicates some dynamics are missed. From understanding of drying dynamics, the stoppages in the flash dryer feed would result in the temperature spikes, which can be clearly seen in the trend. Examples of these are marked on the trend as “1”. The model follows the general trend of the flash dryer outlet temperature, but some drops in outlet temperature is missed, as indicated on the trend by “2”. It is possible that these could be dynamics introduced by unmeasured disturbance variables, such a feed moisture variation which would introduce drops in the outlet air temperature. These drops could also be introduced by peaks in air humidity, although it is unlikely that air humidity would change so rapidly. Change in air humidity is expected to rather introduce a slower dynamic such as a process drift. Inlet air temperature is expected to have a similar influence. Inlet product temperature may be a higher frequency

disturbance, such as inlet moisture, and may be worth investigating as a cause for the drops in the flash dryer outlet air temperature.

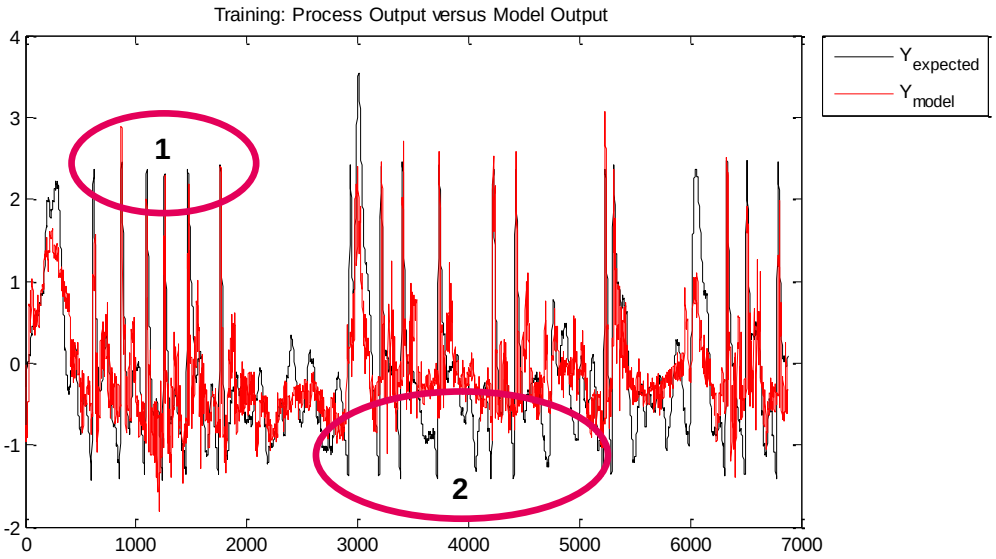


Figure 50: Model flash dryer outlet air temperature compared to the expected process values. Peaks in outlet temperature due to feed stoppages are identified well (1), whereas some drops in the outlet air temperature is missed (1). This could possibly due to lack of feed moisture and temperature measurements.

The system identification of the flash dryer, given the measured variables and rigorous data preparation, was not successful.

8.1.5 Summary: Flash Dryer System Identification

Three process states were identified based on visual inspection of the data: idle-, anomaly- and normal process states. These process states and the possible different dynamics were taken into account during the SID exercise. The idle and anomaly states could not be modelled successfully.

Surrogate data comparison indicated that only dataset 3 is suited for system identification. This was confirmed by the system identification results which could not identify models using dataset 1, but could find better models using dataset 3.

The correlation to the least lagged process variable indicated that dataset 3 should obtain a model other than the least lagged process variable easier than the other datasets.

The decision between which parameter set to use was unclear, although AMI provided flash dryer model results more consistently than the autocorrelation delay parameters.

The best model identified for the flash dryer was based on the normal process state only. This model was however very weak, and it was decided that no model could be found. This failure to identify a model was not due to over fitting, neither could any indication be found that there are any process information still available in the residuals which are located in the model inputs. Only the linear correlation was inspected.

The inability to obtain a model is possibly due to unmeasured disturbance variables, such as feed moisture and temperature, as well as inlet air humidity and temperature. The product moisture and outlet air humidity could also supply a more truthful view of the process efficiency and process model accuracy. The size of influence of these variables need to be researched further to establish which sensors are the most beneficial for inclusion in dryer operations.

8.2 Hot Gas Generator

This section will specifically look at the datasets, delay parameters and system identification experiments focussed on the hot gas generator. The reason for investigating the hot gas generator is twofold:

- this is the focus process unit for the control strategy currently implemented on site; and
- the lack of a flash dryer process model necessitates investigation of other sections as possible areas to focus modelling and control on.

The same discussion structure as for the flash dryer is used, although this discussion is much shorter, due to more definite results.

8.2.1 Dataset Analysis

The datasets for the hot gas generator are visually much less complicated than the dataset obtained for the flash dryer. This is largely due to the presence of only two inputs and one

output variable, as well as the “bang-bang” control philosophy followed by the hot gas generator.

From the collection of subdivided datasets, resulting from the data cleanup and reduction exercise, the longest timeseries was chosen for further investigation. This dataset is arbitrarily named “dataset 1” based on the length of the dataset compared to the remaining subdivided HGG datasets. The discussion surrounding the choice of this dataset is included in Appendix B.2.

8.2.1.1 Process States Identified

No process states were identified for the hot gas generator process. The removal of the controller on-off states, as discussed in 0

Data Cleaning, removed the only visual difference in the process states. The cleaned datasets where the controllers were “off, are thus investigated further without any expected variation in process states.

8.2.1.2 Visual Inspection of the Dataset

As mentioned, the reduced and cleaned datasets for the HGG are visually much less complex than the flash dryer datasets. This is seen in the second and third trends in Figure 51 below where the coal feed shows a stepped control and the outlet air temperature follows the oscillations of the input.

The fluidising damper shows zero variation, as seen in the first trend of the figure. This is the case in all the feasible datasets investigated. This variable is removed from further research due to this lack of variation.

The data cleaning exercise removed the active controller states as is confirmed by the last two trends in the figure below.

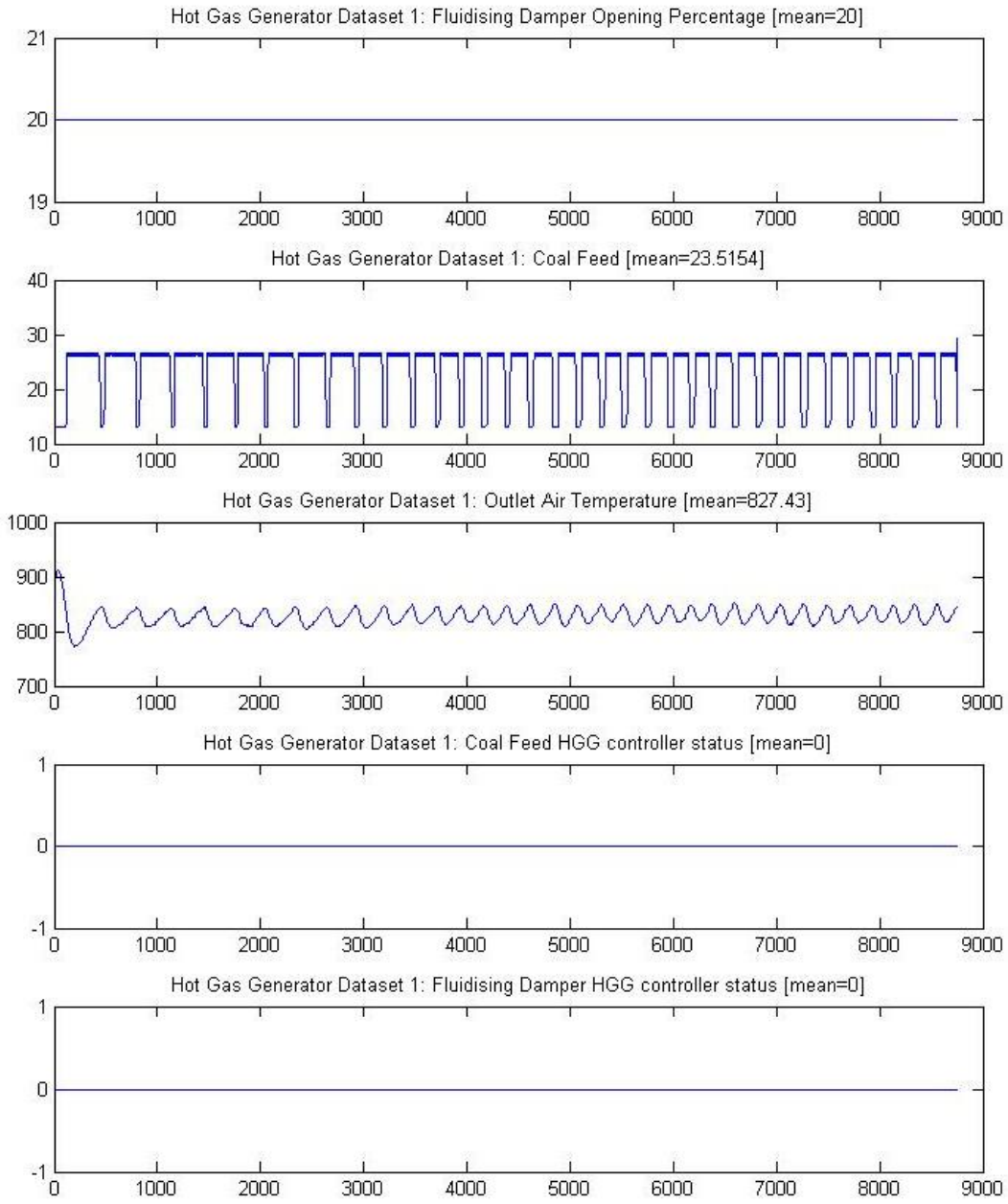


Figure 51: Trends of variables for hot gas generator dataset 1. The fluidising damper shows no variation; controllers are all off; and the coal feed and outlet temperature is clearly linked based on the oscillations.

This dataset will thus be used for system identification without any alterations or recognition of process states. Only the coal feed and outlet air temperatures will be used as input and output to a dynamic, nonlinear single-input-single-output (SISO) model.

8.2.1.3 Correlation with the Least Lagged Process Output

From the similar discussion for the flash dryer in 8.1.1.4, the table of correlation of the least lagged variable and the one-step-ahead model output is included below. It can be seen that

a poor correlation between the one-step ahead predicted output and the least lagged output exists for both delays identified by AMI and autocorrelation. The population of solutions only needs to overcome these relatively low correlations to enable identification of a model other than a least lagged process output. From the weak correlations, the system identification is expected to be successful in identifying a model other than the least lagged process output.

Table 32: Correlation coefficients between closest lagged process output and 1-step ahead value. Obtained models are expected to have better correlation figures than the least lagged output, otherwise the least lagged process output will be the best model.

Process	Dataset Number	Output Lag as per AMI (5sec increments)	Correlation Coefficient (AMI)	Output Lag as per Autocorrelation (5sec increments)	Correlation Coefficient (Autocorrelation)
Hot Gas Generator	1	65	0.128	76	0.081

8.2.1.4 Choosing the Most Representative Dataset

The surrogate data analysis indicated that dataset 1 contains sufficient deterministic information to separate it from the surrogate data. This holds true for both parameter sets identified by AMI and autocorrelation. The trends for the surrogate data comparison are visually very similar for both parameter sets. The result from AMI is displayed below.

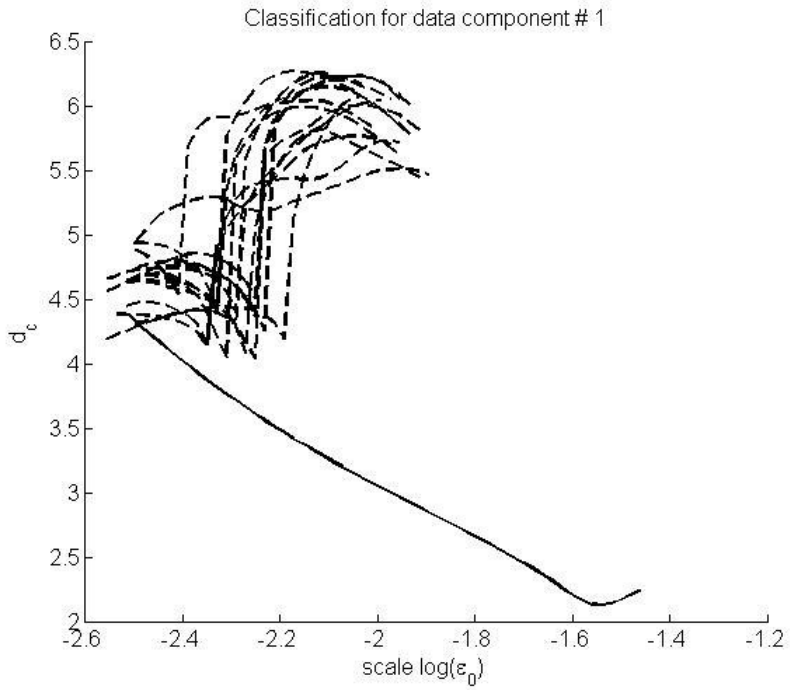


Figure 52: Surrogate Data Classification of the HGG outlet air temperature for the AMI delay parameters. This comparison shows clear separation, indicating that the process dynamics are deterministic.

The complete set of analyses are included in Appendix C.9 Analysis 9: Hot Gas Generator Dataset 1.

8.2.1.5 Conclusion

The following conclusions are drawn from the analysis of the datasets for the hot gas generator system identification:

- No process states could be identified from the reduced and cleaned subdivided datasets; and
- Dataset 1 is well suited for modelling purposes as it indicates good separation from the stochastic surrogate data and low correlation compared to the least lagged process output

8.2.2 Delay parameters

The delay parameters for dataset 1 are displayed and discussed in an attempt to identify if either of AMI or autocorrelation methods of identifying the parameters are preferred by the HGG timeseries used.

8.2.2.1 Delay parameters Identified

The delay parameters identified for dataset 1 show very similar delays identified for both AMI and autocorrelation methods. As seen in Table 33 below, the autocorrelation delays for the outlet air temperature is 11 time intervals (55 seconds) larger than AMI; and coal feed delay is only 1 time interval difference.

Hot Gas Generator Dataset 1:

Table 33: Delay parameters for process variables for Hot Gas Generator Dataset 1. The fluidising damper data has no variation and is henceforth omitted from the study.

Process Variables	Method to Determine Delay *k*	Delay (*k*)	Number of Delayed Variables (m)
Coal Feed Rate	AMI	44	10
	Autocorrelation	45	10
Fluidising Damper	AMI	Zero variation	Zero variation
	Autocorrelation	Zero variation	Zero variation
Hot Gas Generator Output Temperature	AMI	65	4
	Autocorrelation	76	4

The choice of which parameter set to use is not clear and will be investigated by the GP.

8.2.2.2 AMI vs. Autocorrelation as Identified by the GP

The validation R^2 values in Figure 53 indicate that for each set of comparable experiments, i.e. experiments using the same experiment parameters, the autocorrelation delay parameters outperformed the parameters identified by the AMI. The modelling differences for experiment set A are noticeably favoured to the autocorrelation delay parameters with an 18.6% difference in validation R^2. The rest of the experiments indicate only slight, an almost negligible differences in performance.

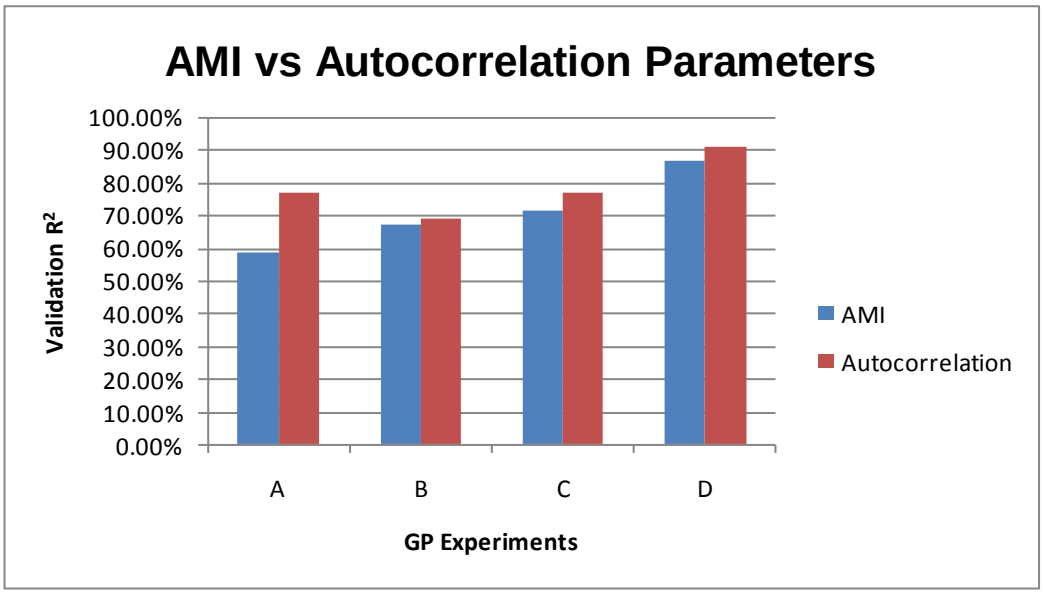


Figure 53: Comparison of Validation R^2 values for comparable experiments with different delay parameters. The results are similar, but autocorrelation seems to consistently perform better than AMI

The difference is not significant enough to warrant further investigation of just one parameter set; hence both AMI and autocorrelation models will be investigated further.

The comparison results for the figure above are tabulated in Appendix 238G.1.2.

8.2.3 The Best Model Obtained: Hot Gas Generator

As mentioned previously, a smaller number of SID experiments were required for the hot gas generator than for the flash dryer exercise. This can be largely ascribed to the less complex timeseries being investigated as well as the stronger availability of deterministic information. Due to the success of this system identification exercise, these models are carried over to the MPC experiments. Some time will thus be spent in this section to compare the models and fully understand the results obtained.

The top 4 models chosen for further investigation are included in Table 34.

Table 34: Top 4 Hot Gas Generator system identification results according to validation MSE.

Model Number	Latent Variable Delay Parameters	Train MSE	Train R^2	Validation MSE	Validation R^2
1	Autocorrelation	0.165	72.8%	0.133	77.1%
2	AMI	0.103	83.1%	0.167	71.2%
3	Autocorrelation	0.093	84.7%	0.133	77.1%
4	Autocorrelation	0.094	84.6%	0.151	74.1%

Model 1 is the best model obtained with the most basic functional set used. A model of corresponding simplicity was obtained with validation statistics comparable to the other models.

Model 2 uses the advanced functional set with the timeseries biased after normalisation to assist with closure on the square root function. These experiments provided the weakest result according to validation statistics, but the strongest model for AMI delay parameters.

The models 3 and 4 were both obtained from the same experiment. This experiment uses the same method as for model 2, but only using autocorrelation delay parameters. Model 3 shows the best validation and training statistics. Model 4 is however investigated due to the unstable freerun prediction of model 3. This is discussed in Chapter 9. All three of models 2, 3 and 4 make use of the bias induced in the timeseries after normalisation. Comparison and discussion of these four models are now discussed further.

The validation data output for Model 1 shows spikes on the model output at the lower extremes of the oscillations. There are also step wise increases, with unexpected decreases in outlet air temperature, compared to the smoother temperature increases of the expected temperature.

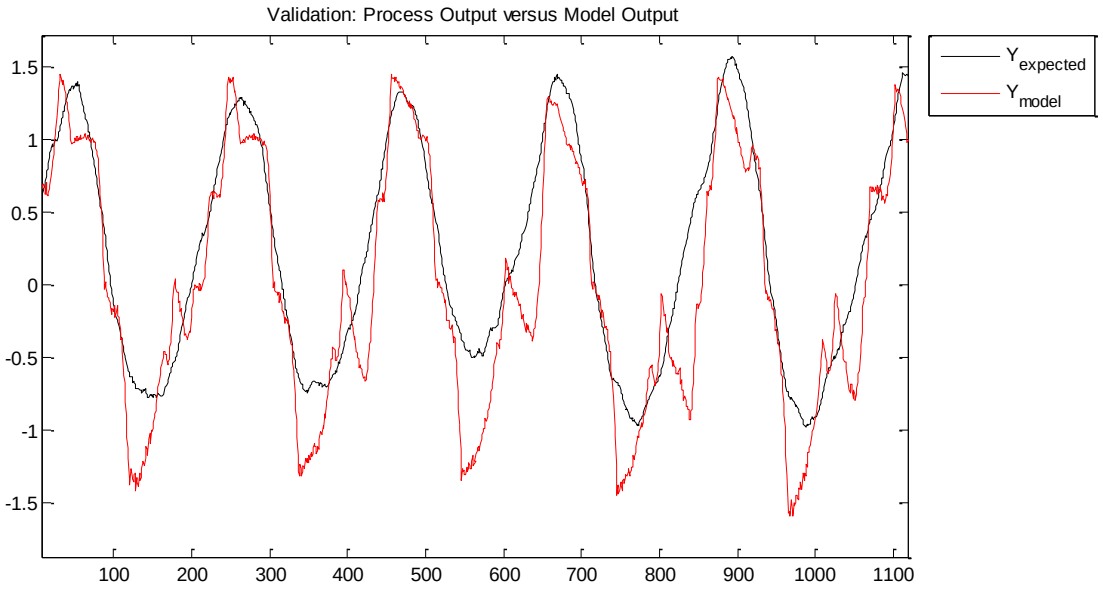


Figure 54: Model output plot for model 1, validation data, indicates some spikes at the dips in the output and spikes after the dips.

Model 2, Figure 55, performed better than model 1 at the lower ends of the temperature oscillations, but with some high frequency noise at the lower extremes of the model output. The output at the lower temperature outputs are also more lagged for the model output, shifting the output to the right of the expected values. Furthermore the peaks are predicted lower and flatter than expected. When used in control this model ,might result in more coal being fed to the process than necessary, as the model indicates a lower temperature than the actual process.

No clear distinction can be made visually between models 1 and 2.

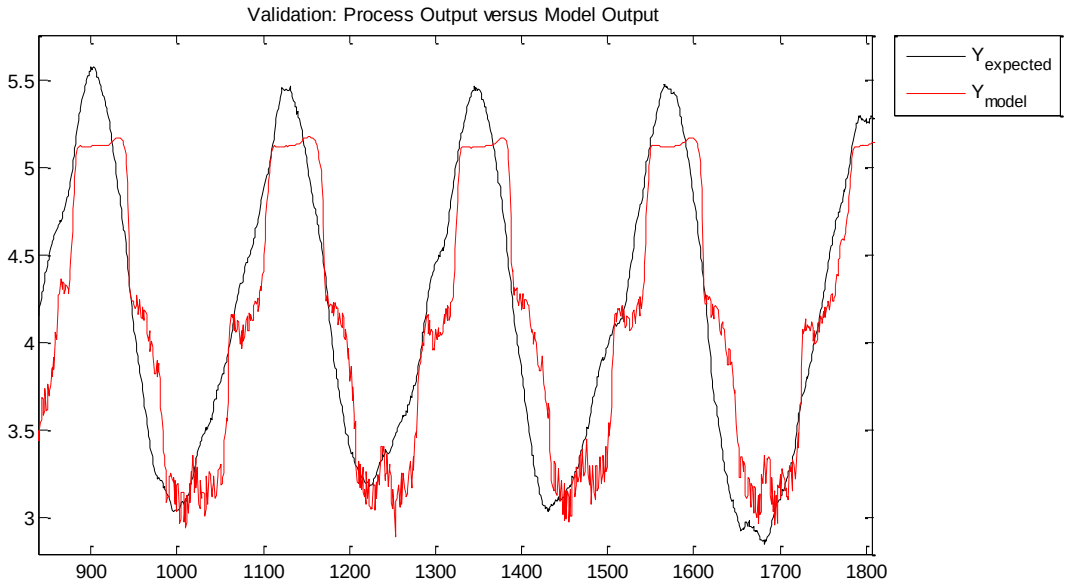


Figure 55: Model output plot for model 2, for the validation data, indicates noise in the dips of the data and flat peaks lower than expected.

Model 3 is the best fit model identified for both the training and validation datasets. The output for the validation set, Figure 56, indicates similar findings to model 2, with the exception that the model output at the lower ends contains more noise. Comparison to model 2 visually is difficult and left to the fit statistics, which indicate model 3 performs better.

Model 3 is thus preferred over model 2. Model 1 shows the same fit statistics, thus no distinction can be made between models 1 and 3.

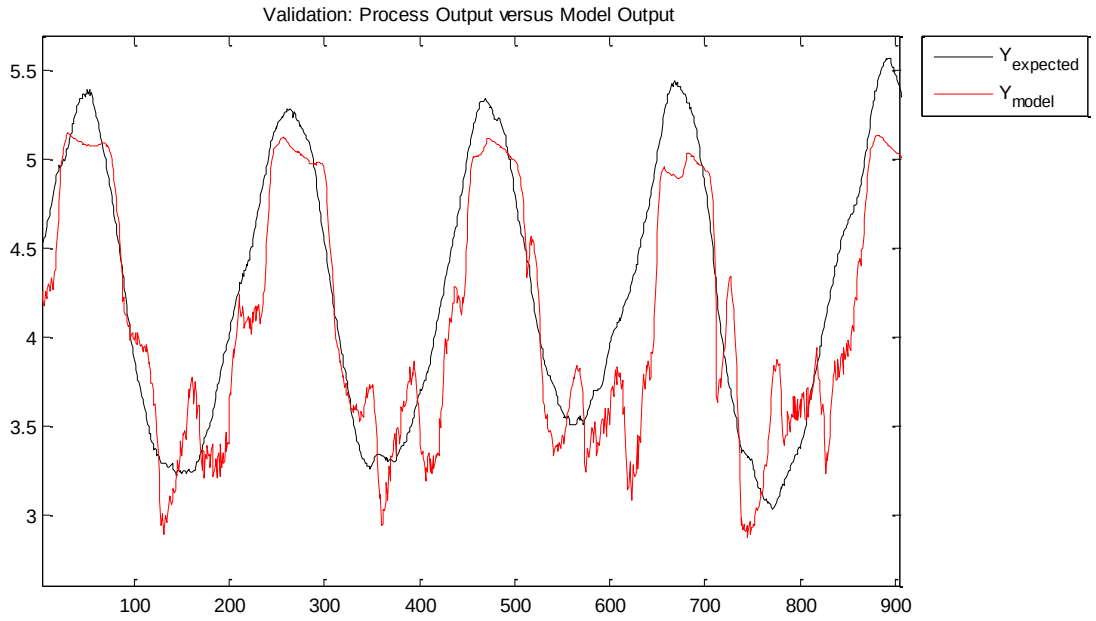


Figure 56: Model output plot for model 3, for the validation data. The peaks are predicted lower than expected, with some large noise oscillations in the dips of the oscillations.

Model 4 compared to model 2 indicate better estimation of the peaks, but lower estimation of the outlet temperature lows. The fit statistics indicate that model 4 is better than model 2. The decision between model 1, 3 and 4 is left to the fit statistics, which indicate models 1 and 3 are the better representations of the process as seen in Table 34.

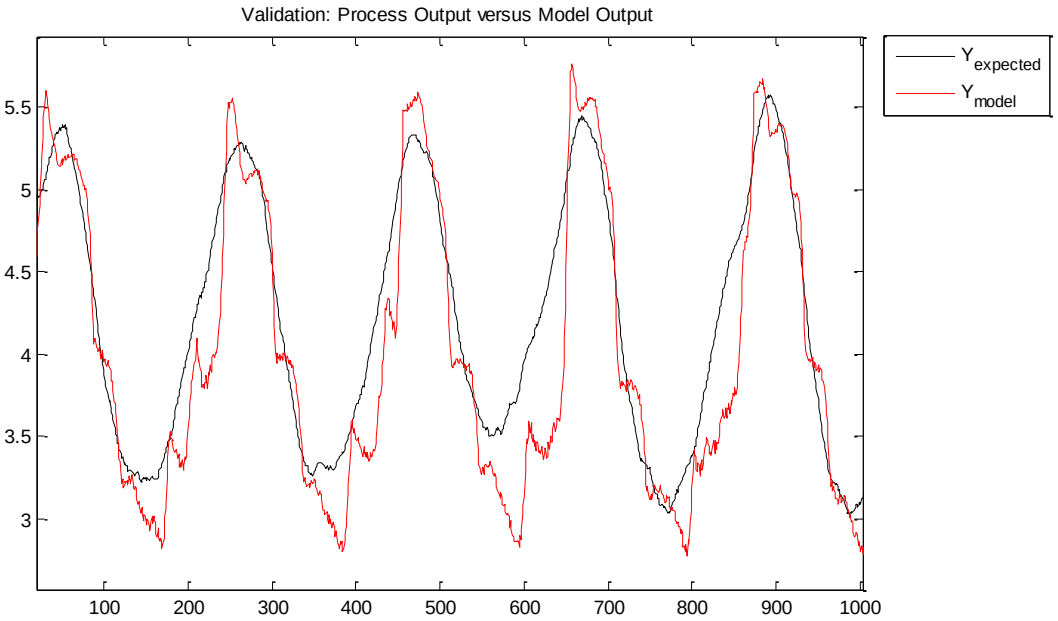


Figure 57: Model output plot for model 4, for the validation data. The dips are predicted lower than expected with some delayed dips to the right of the original dip.

Models 1 and 3 are the preferred models based on the fit statistics. Model 1 is noted to be less complex due to the more basic functional set and simpler model structure, which could assist in easier optimisation of the control moves by the MPC algorithm. During the sensitivity analysis in the next chapter it will be shown that model 1 is rejected. Model 2 will be investigated as a representative of the AMI delay parameters used. The inclusion of model 4 is discussed 9.1 Model Freerun Prediction Ability, where becomes clear that model 3 is not stable enough for application to a MPC solution.

8.2.4 Summary

The longest hot gas generator dataset was selected as the dataset to use. The surrogate data comparison indicated sufficient deterministic information in the data. The fluidising damper was removed from the research seeing as it contained no variation in the available subdivided datasets.

The delay parameters indicated that autocorrelation consistently outperformed the AMI parameter set. This contradicts the findings in the flash dryer, possibly indicating that neither delay parameter identification method can be favoured.

Models were identified for the hot gas generator with less effort than for the flash dryer, and four of these were investigated further. The models identified will be investigated for freerun prediction ability; sensitivity and reaction to stepped process inputs; and the remaining best models then applied in a MPC solution. This is discussed in the next section.

8.3 System Identification Summary and Conclusions

The major conclusion made is that the flash dryer is not suited for modelling, and hence not suited for model based control either.

The hot gas generator however is much better suited for modelling and control.

Despite the rigorous data preparation and pre-modelling data analyses, no representative model could be identified for the flash dryer. The flash dryer is thus removed from further MPC research for this study.

The following conclusions were drawn from the flash dryer system identification:

- Three process states were identified from the data, but neither one could be modelled successfully, or add information for successful modelling of the normal process state;
- Surrogate data comparison results were confirmed by the system identification exercises. Better models were created using dataset 3, compared to dataset 1, as was predicted by the surrogate data comparison;
- AMI delay parameters resulted in more consistent results than the autocorrelation delay parameters;
- There is missing information in the flash dryer inputs. Feed moisture, airflow humidity and feed temperatures could add this missing disturbance information. Analysis of the best flash dryer model indicated no strong correlation between the residuals and the model inputs, indicating no clear information overlap due to poor model fitting.

Models were identified for the hot gas generator. These are seen as representative and used in further MPC studies in this research.

The following concluding remarks need attention with regards to the hot gas generator:

- No prevalent process states can be identified from the subdivided timeseries after data reduction and cleaning;
- The fluidising damper is removed from the modelling and control strategy due to the lack of variation in the process variable in the timeseries;
- As found by the surrogate data comparison, dataset 1 for the hot gas generator contains sufficient deterministic dynamics for system identification;
- Delay parameters as identified by autocorrelation allows for better model identification than the AMI delay parameters for the timeseries studied; and
- The best 4 models identified will all be investigated further for application in an MPC solution.

Chapter 9 Results: Model Based Control

A MPC solution is developed using the nonlinear HGG models identified and constructing a freerun prediction over an established prediction window and optimising the control moves. Resulting from the latent variable construction used in the system identification methodology, a real time latent variable construction is built into the algorithm. This latent variable construction is extended by the freerun prediction, which requires predicted values and optimised control moves to be incorporated in a real time latent variable construction. This amalgamation of methods for control using this nonlinear model is tested in an attempt to answer the following questions:

- Can the current CSense Architect development environment be used, as-is, for MPC development?
- Can the data preparation strategy, specifically the latent variable construction, be used in the environment for an online solution?
- Is the nonlinear model developed suited for freerun prediction?
- Can a freerun prediction be constructed in the CSense development environment?
- Is the freerun prediction of the nonlinear model developed optimisable using the given CSense optimisation tools?
- Can a model based predictive controller be used to control the dryer setup investigated, given the specifics of the system identification methodology and the software environment available?

The aim is thus not to create a fully functioning model based predictive controller (MPC), but rather to investigate the questions stated.

The flow of the chapter follows the MPC development methodology used and is summarised in the following diagram.

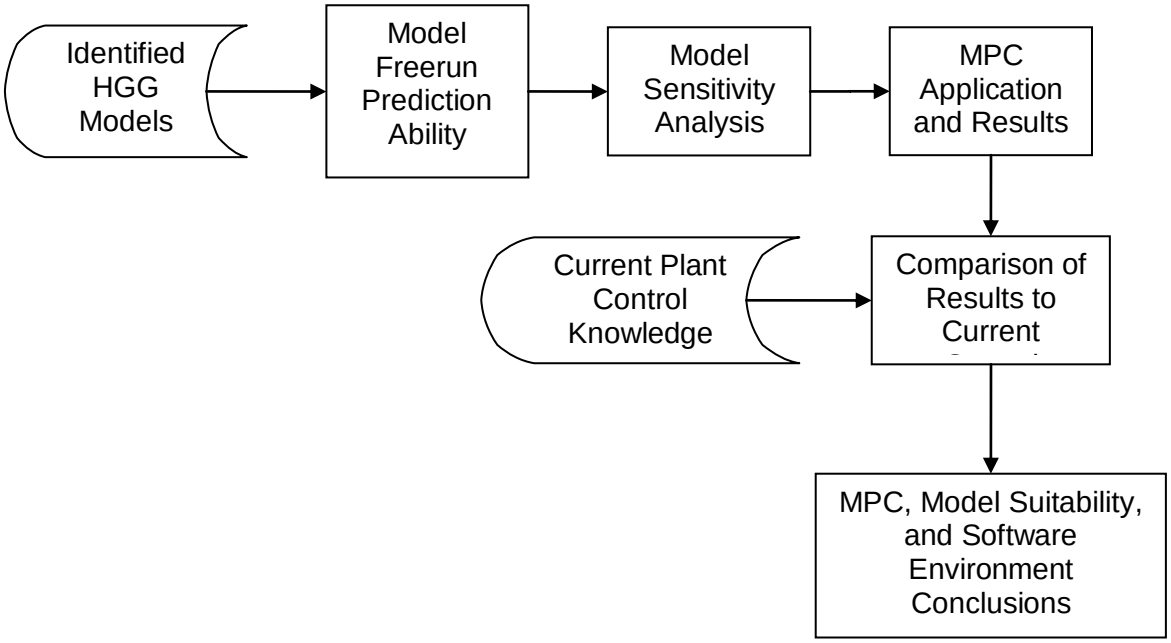


Figure 58: Flow of the MPC experimentation process with the aim of finding i) Are the models identified suited for MPC and control? ii) Is the available software environment able to host the control strategy and algorithm? and iii) How do the results compare with current control?

The HGG models identified are analysed looking at the sensitivity to the manipulated variable (coal feed) and stability over a freerun prediction. The MPC is developed and results obtained. Results are compared against current control. Conclusion are then drawn regarding the models used, software requirements foreseen and MPC as a dryer solution.

The conceptual investigation is focussed only on the hot gas generator as no accurate model could be identified for the flash dryer section.

9.1 Model Freerun Prediction Ability

The freerun ability of the identified models is investigated for two reasons:

i. As a measurement of the prediction ability of the model; and

ii. To ensure that the model is stable in predicting the required time window necessary for predictive control modelling.

The table of the top 4 HGG models investigated is repeated here from section 8.2.3.

Table 35: Hot Gas Generator models investigated for use in a MPC solution

Model Number	Latent Variable Delay Parameters	Train MSE	Train R^2	Validation MSE	Validation R^2
1	Autocorrelation	0.165	72.8%	0.133	77.1%
2	AMI	0.103	83.1%	0.167	71.2%
3	Autocorrelation	0.093	84.7%	0.133	77.1%
4	Autocorrelation	0.094	84.6%	0.151	74.1%

The best model identified, as discussed earlier, is model 3. This model however becomes unstable during freerun prediction as can be seen in Figure 59 below. After 77 time intervals the freerun prediction suddenly deviates from the expected value. The model output oscillates and crosses the expected output twice after the first deviation, but then steers off and does not return to the expected value for the remainder of the freerun prediction.

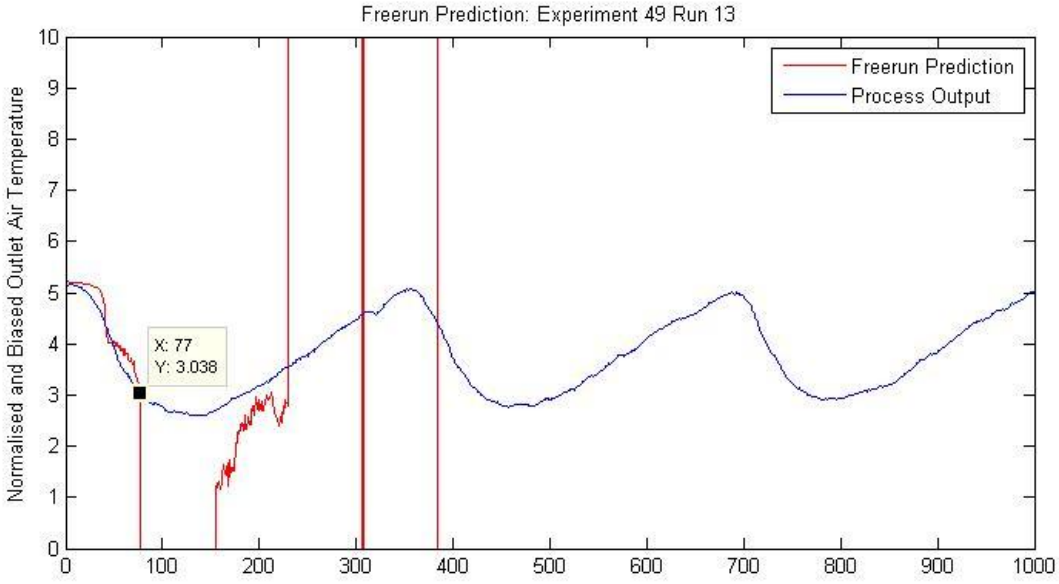


Figure 59: Freerun prediction ability of the model 3 shows poor freerun prediction ability with the prediction growing exponentially and ultimately leaving the expected values completely..

For this reason the next best model for autocorrelation from the same experiment, model 4 in the foregoing table. The freerun prediction of this model shows a visual good fit to the oscillations in the data, although the highs and lows are over and under estimated, as can be seen in Figure 60 below. The model shows considerable stability in identifying the oscillations in a timely manner.

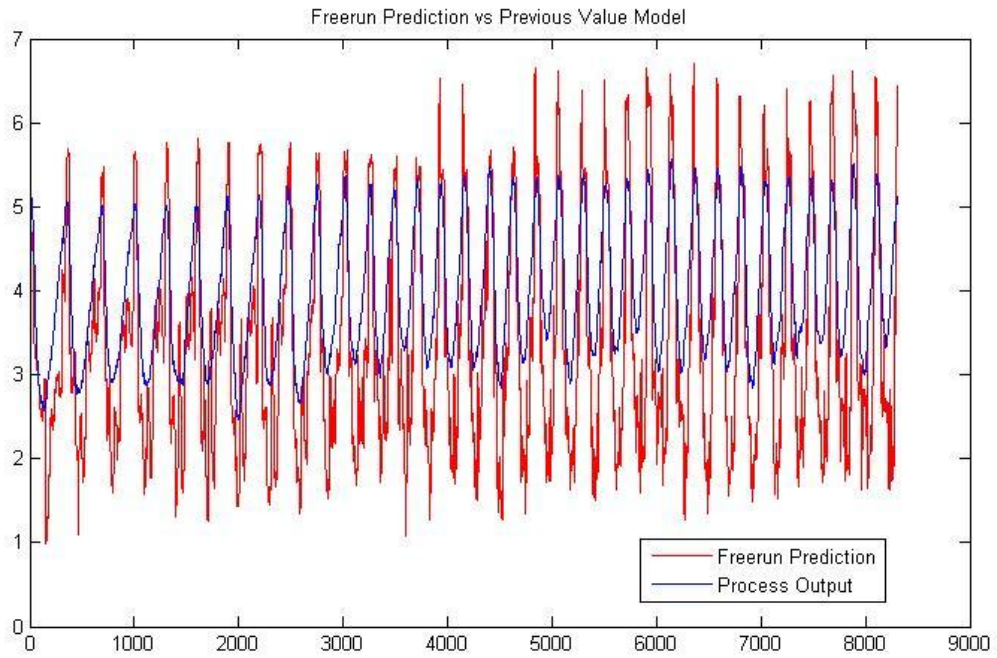


Figure 60: Freerun prediction ability of the model 4 shows stability and good identification of the process oscillations.

Closer investigation of the initial stages of the freerun prediction indicates that the process output is less smooth than expected. The sudden drops and rises in the model output could not be traced to a specific model input at the specific time step. This trend does not supply a thorough view on the freerun prediction ability of the model.

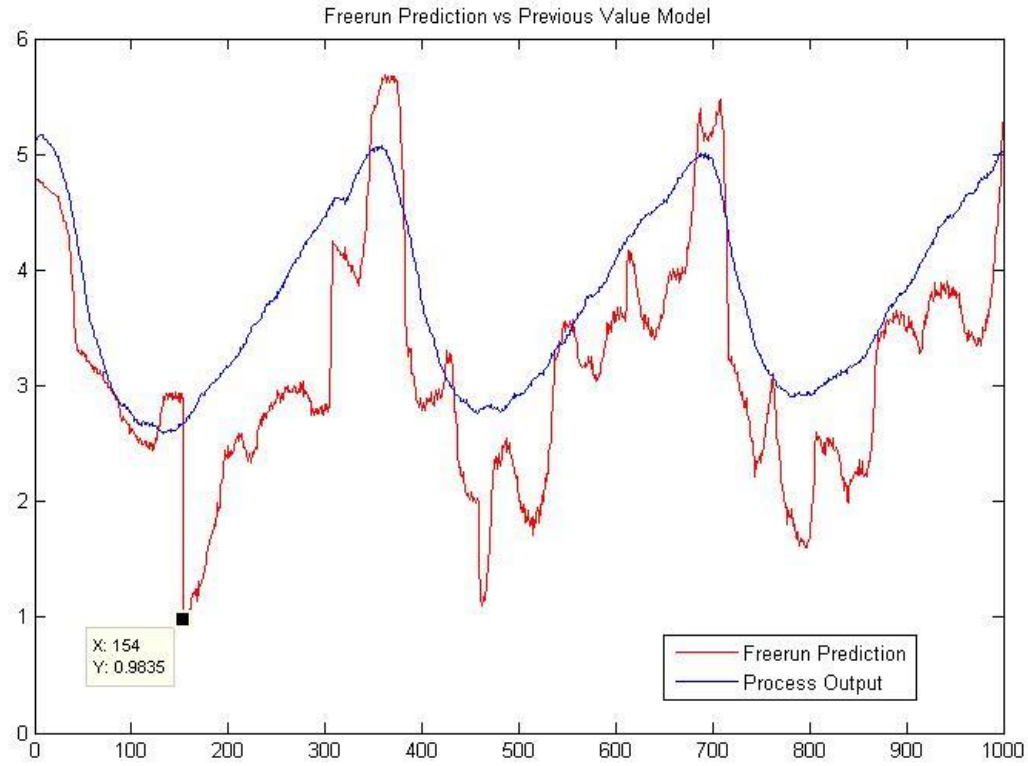


Figure 61: Closer view of the freerun prediction ability of the model 4 indicating sudden deviations in model output

A plot of the development of the mean square error and R^2 values per prediction step was introduced, included in Figure 62 below. This figure indicates that the model freerun prediction performance peaks at 153 time intervals. This corresponds with the model output in Figure 61. From 153 time intervals onwards, the model statistics in the freerun prediction indicate increasingly poor model performance for the remainder of the prediction with the R^2 value becoming negative. Small oscillations on the R^2 and MSE fit statistics in Figure 62 correspond with the oscillations of the model output, providing evidence that the poorer model output at the extremities are causing the poor fit. Furthermore the freerun prediction loses accuracy as the prediction window increases. This might hamper accuracy of the controller depending on the prediction window required.

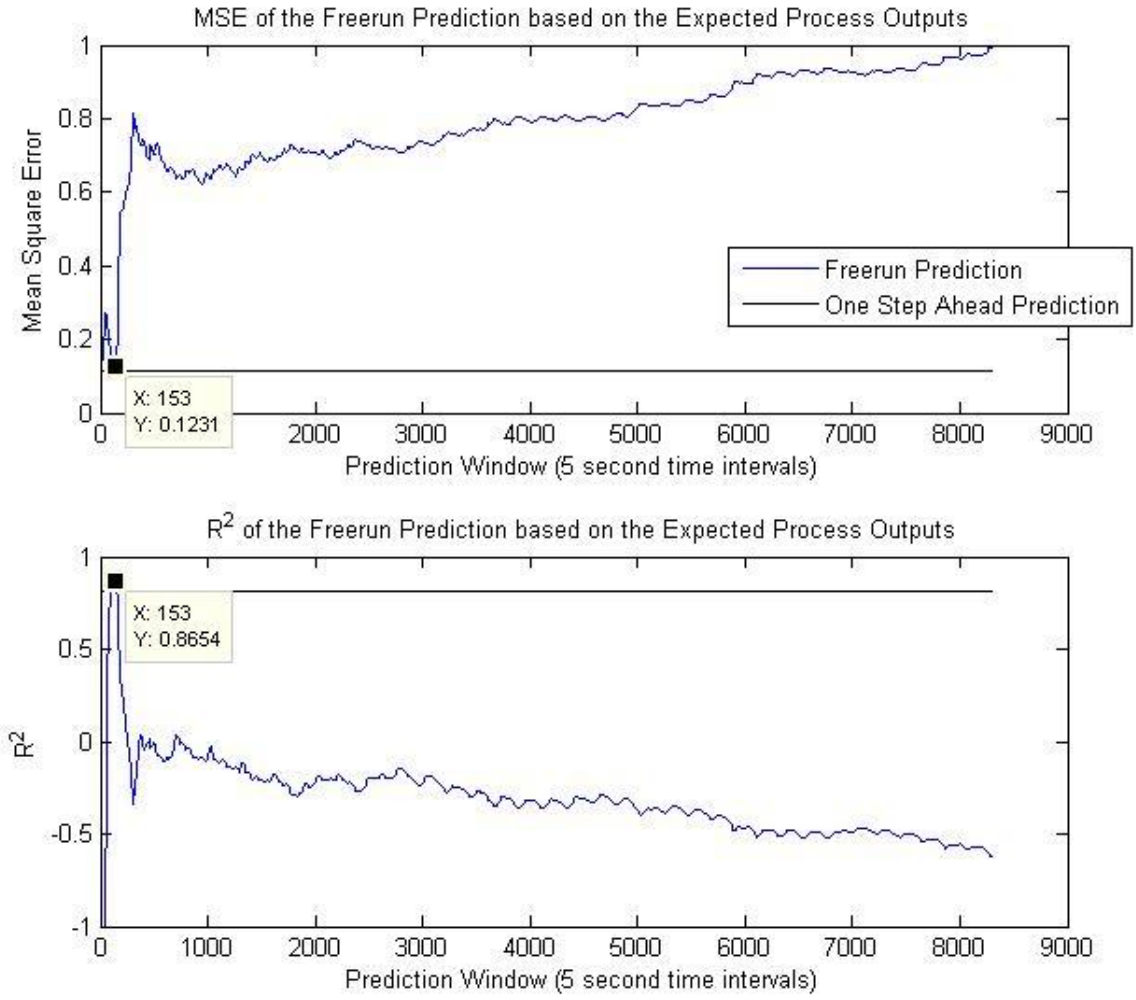


Figure 62: MSE and R^2 values per prediction step, compared to real process data, for the freerun prediction of the model 4. The freerun prediction loses accuracy after 153 time steps and keeps losing accuracy, quickly ending with an R^2 below zero.

The third model investigated is model 2. This model has the worst fitness statistics of the three models investigated thus far, but indicates more stable freerun predictability than the previous two models with less deviation at the highs and lows. From visual inspection it is hard to see why the model has poorer fit statistics than the previous models. The initial stages of the model output deviates from the expected value in the area between the lows and highs. The model output however loses that instability after four oscillations and seems to stabilise.

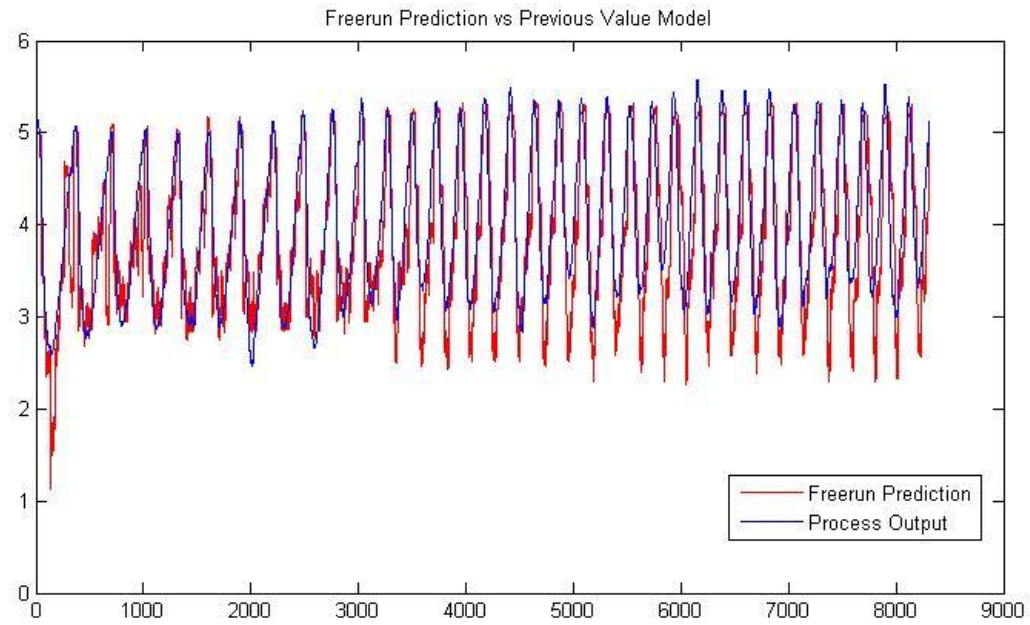


Figure 63: Freerun prediction ability of the model 2 indicates good tracking of highs, lows and oscillations.

The growth of the R^2 and MSE values during the freerun prediction indicate that there is a decrease in fit after 133 time intervals of freerun prediction. However, the R^2 and MSE values recover and move closer to the statistics identified during the SID step. This is seen in Figure 64, where the black line represents the fit statistic found during SID and the blue line is the evolution of the statistic during the progression of the freerun prediction. Despite the worse fit statistics this model seems the best combination of process representation and freerun stability. The long range freerun stability could assist in the accuracy of the controller.

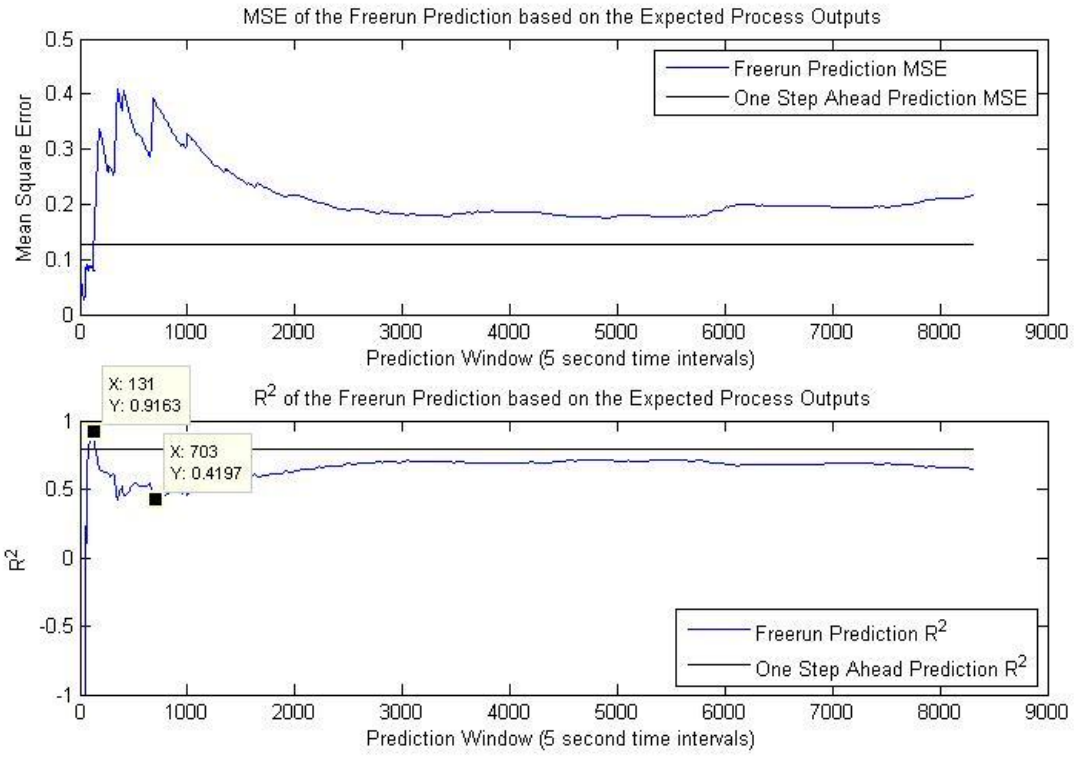


Figure 64: MSE and R^2 values per prediction step, compared to real process data, for the freerun prediction of model 2 indicate that the freerun prediction loses accuracy after 131 time steps, but then recovers after 703 time steps. On the long run it strives to the accuracy of the initial model fitting.

Based on the freerun prediction ability the last model discussed, model 2, should be used due to better freerun stability. However, the better validation fit statistics of the model 4 indicates possibly better capturing of the process dynamics, although the freerun prediction seems more unstable. Both these models should be investigated further. It is however expected that model 4 is too unstable on the freerun prediction.

Due to the poor results from the sensitivity analysis of model 1 the freerun prediction of model 1 is not investigated. These results are investigated next.

9.2 Model Sensitivity Analysis

The same model numbers as set out in Table 34 in the previous section are kept for discussion in this section.

The reaction of the identified models to steps in the input variables was determined for the following reasons:

i. To see if the output reacts as expected and the process settles at the expected output;

ii. To estimate the settling time, and thus prediction time window required for model predictive control

Three of the identified models are investigated. These are model 1, model 2 and model 4 as indicated and discussed in the previous section and set out in Table 35. Model 3 is removed according to the finding that it is too unstable during freerun prediction.

For this sensitivity analysis, the model latent variable construction is initialised using the initial timeseries used for training. Hereafter the model is left to freerun prediction and the stepped input, in Figure 65 to the SISO model. The stepped input is selected as the minimum, midrange and maximum points of the coal feed time series used for training. These steps were selected 13, 20 and 26 tons/hour. These minimum and maximum values were also applied as the input constraints in the MPC, discussed in the next chapter.

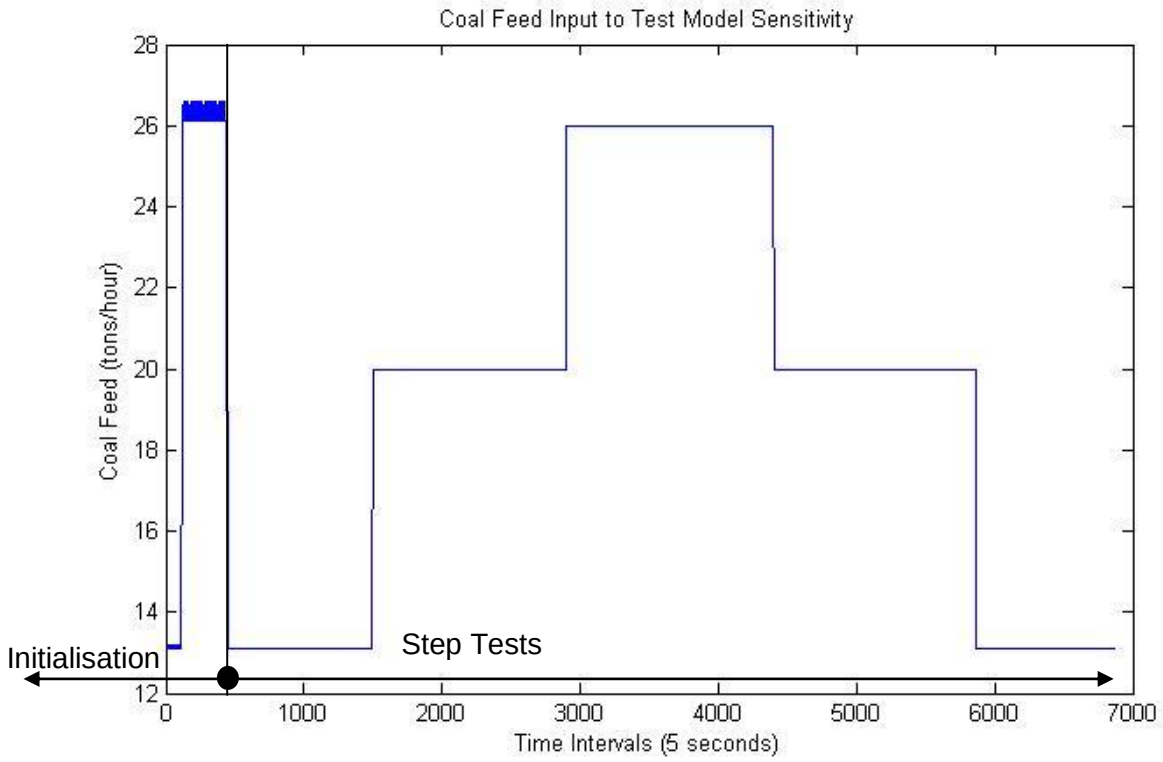


Figure 65: Coal Feed input used for sensitivity analysis of models. The section on the left, as indicated, is for process valid initialisation of the latent variable construction. The steps on the right are to test the sensitivity to the model to the coal feed.

Model 1, in Figure 66 below, is included in the discussion although the process did not settle on a higher or lower state as would be expected. The model was stable across the whole sensitivity analysis and reacted to the steps in input as indicated in the figure. Interestingly, the coal feed increase initially indicates a drop in output temperature, then a small increase after a few time intervals, followed by a few oscillations before it comes to rest at the same temperature as before the step input. Investigation of the model structure indicates that only one lagged version of the process output is included in the model structure, thus having a very weak influence on the model output. The lack of being able to change the output permanently deems this model useless. It is removed from further investigation.

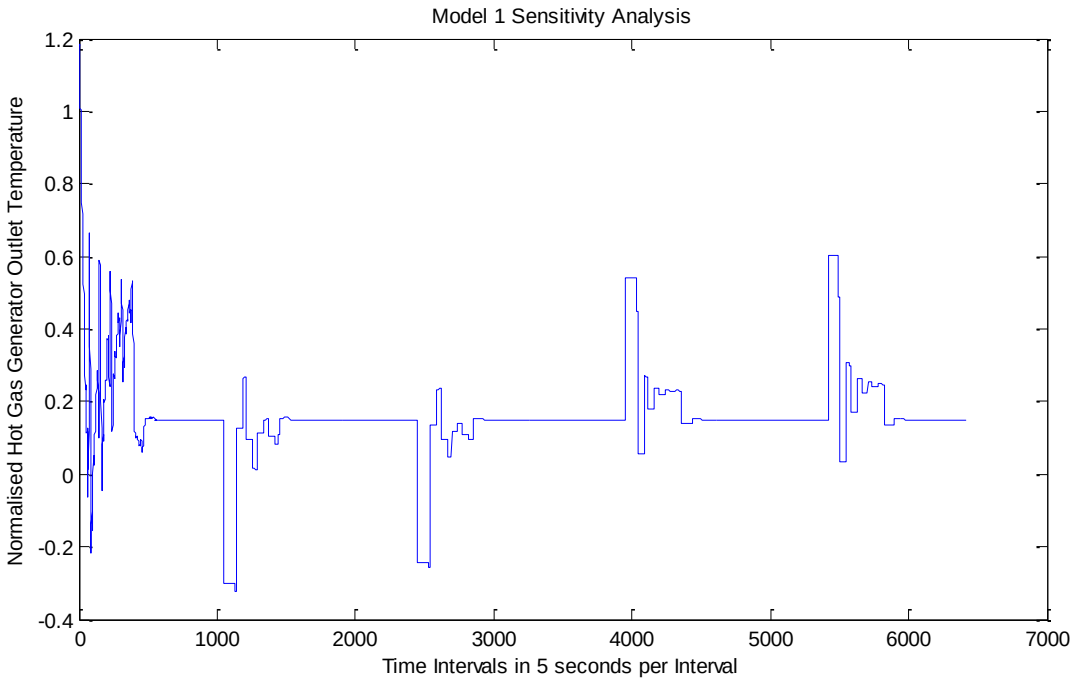


Figure 66: Sensitivity Analysis output for the model 1 shows that the model cannot shift the process permanently according to the feed.

The sensitivity output from the model 4, in Figure 67 below, indicated much better reaction to the stepped input. Once again the outlet temperature drops with the initial step input in coal, followed by the slight increase. However, the process settles at a lower temperature. This is not dynamics expected from any burner and indicates that the model is incorrect. It is in actual fact inverted. A few of the weaker fitted models, from the same experiments as where

these models originate, were analysed as well and indicated either unstable behaviour or the similar dynamics, despite comparable fit statistics.

The “settling time” was visually estimated to a point where the remainder of the process is not oscillating far from the settling temperature. The number of time intervals for model 4 are 410, 453, 457 and 452 from the point of the coal feed step and the point where it “settles”. This “settling time” will be used in the prediction window for the MPC.

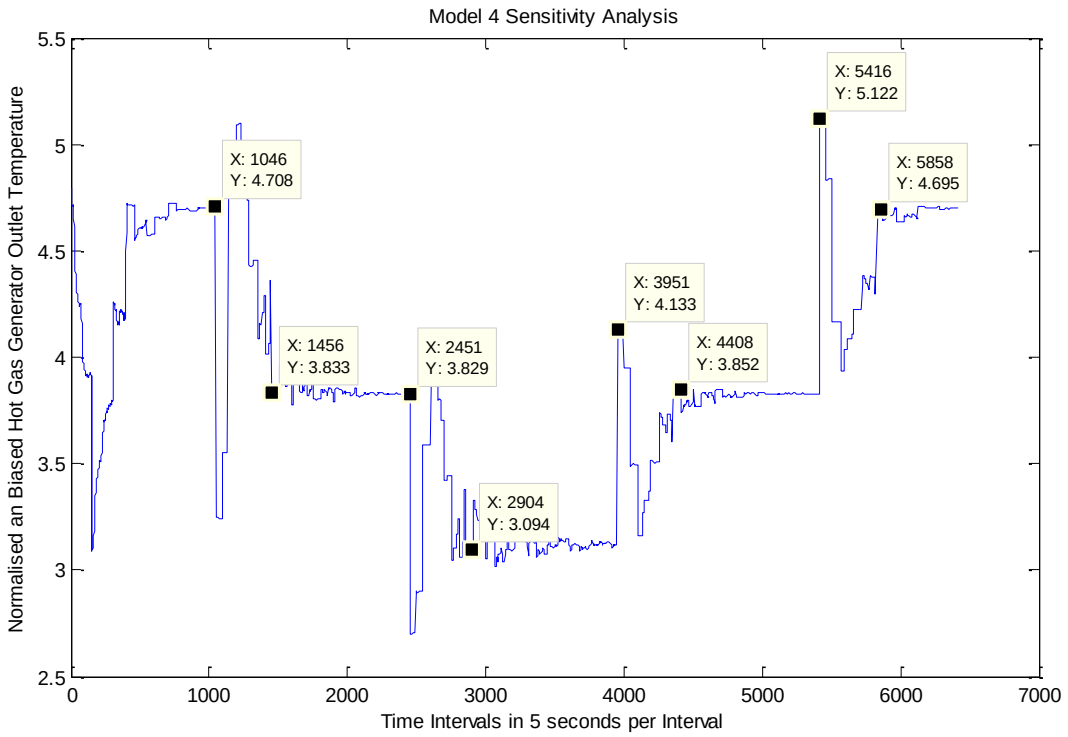


Figure 67: Sensitivity Analysis output for the model 4 with data labels to estimate time lags. The process output shifts, but not in the direction expected.

Model 2, in Figure 68 below, indicates the same macro reaction by settling at a lower process state instead of higher state. This model’s output is also much more basic than the previous model, with less noise and a much lower overshoot in outlet temperature after the step was initiated. The more basic output might allow easier optimisation. This will be investigated later by the MPC experiments.

The “settling time”, in number of time intervals for model 2, are 440, 520, 527 and 390 from the point of the coal feed step and the point where it “settles”. The “settling time” is much more varying than for model 4. A single time will be selected for the MPC in the next section.

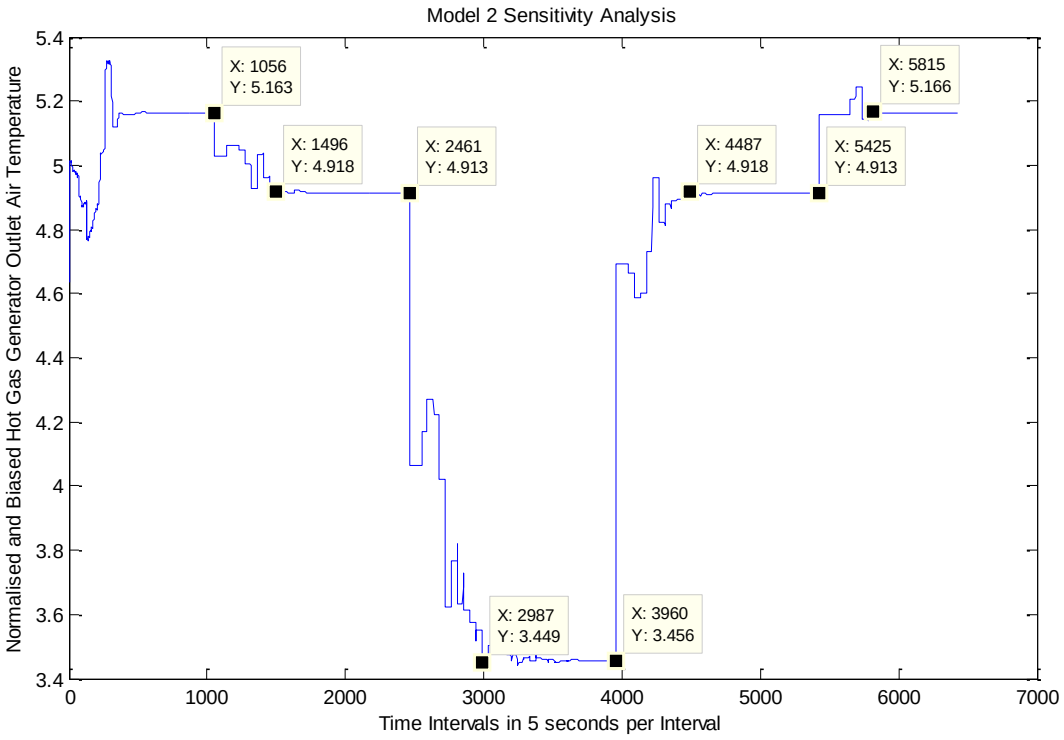


Figure 68: Sensitivity Analysis output for the model 2 with data labels to estimate time lags. The same process shift reaction than model 4 is identified.

The reason for inverted dynamics might be due to incorrect delays incorporated in the model structure. The visual investigation of the dataset included earlier in section 3.2.3 Hot Gas Generator Oscillations and Figure 13, the coal feed delays included in the models are briefly investigated. The visual inspection in section 3.2.3 indicated an expected delay of between 7 – 9 minutes from when the coal feed is increased, until the outlet air temperature starts to rise. The delays included in the model structure for model 2 are:

Table 36: Coal feed delays included in the model structure for model 2

Coal feed delays included in model 2 structure (minutes)
0
7.3
14.7
18.3
22
25.7
29.3
33

Looking at model 4, it is found that this structure as well included more than one delay for the coal feed.

Table 37: Coal feed delays included in the model structure for model 4

Coal feed delays included in model 4 structure (minutes)
0
3.75
7.5
15
30
33.75

Both models included a coal feed delay of just over 7 minutes, corresponding with the visual inspection of the data. The other delays could however be included by the GP algorithm based on some events in the output corresponding to steps in these latent variable inputs. The contributions of these additions could possibly be artificial and not a true representation of the process. It is concluded that the inclusion of more delays then initially estimated in the model structure could have caused the inverse reaction of the model, although this cannot be proven as fact. It is recommended that the influence of the various coal feed delays be investigated further in future research.

Although these last two models, models 2 and 4, are inverted in terms of expected burner dynamics, it was decided to use these models to investigate the ability of applying one step-ahead prediction models, resulting from the identification methodology followed, in a model based predictive controller. The investigation will focus on

- The complexity required to construct the controller from such a long time delay dynamic model using a real time latent variable construction; and
- Ability to optimise the control moves using the dynamic model and latent variable construction.

The aim will not be to construct a usable controller, seeing as a representative model of the process could not be identified, although the fitness statistics indicate a relatively good fit.

The inverted actions may be assigned to closed loop data dynamics. This needs to be confirmed in future studies.

9.3 Model Predictive Controller Outputs

The methodology discussed in Chapter 6 was implemented without any feedback in the control strategy. Models 2 and 4 were used. From the results of the sensitivity analyses and the freerun prediction, it was decided that there will be no mismatch between the plant model and the controller model used for the controller simulations, thus only investigating the usability of the complex latent variable construction required for freerun prediction; as well as the nonlinear model structure for freerun stability and optimisation of manipulated variable control moves.

Two experiments for each of the two models were investigated. The aim is measuring the ability of using the culmination of the identified nonlinear model, real time latent variable construction, model predictive control strategy, identified prediction and control windows and an industry used genetic algorithm optimiser. The reader is referred back to the diagram in Figure 26, and the corresponding discussion, for review of this approach.

The two experiments (runs 1 and 2) include the developed model predictive controllers with prediction and control windows. The prediction window is selected from visual analysis of the settling time during the sensitivity analysis, as discussed previously in section 9.2 Model Sensitivity Analysis. It is noted that these prediction windows are larger than the freerun prediction stability of model 2 and model 4. These were however selected as an initial starting point for prediction window selection. Selecting a shorter prediction window will result in skewed optimisation due to the overshoot spikes in the model output. This might require an altered goal function catering for overshoot and oscillations in the output. Alternate goal functions and prediction window selection is not investigated in this research and it is recommended that these be investigated in a follow up research focussing solely on MPC.

The control window is selected based on the time required for optimisation of the prediction window. This is discussed in the methodology, section 6.3 Process Prediction and Control Move Optimisation.

Table 38: Controller Setups investigated for the control of the hot gas generator

Simulation Number	Model	Prediction Window (Np)	Control Window (Nc)	Set Point Change	Goal Function
Run 1	Model 2	440	96	Yes	Optimised across 440 prediction steps
Run 2	Model 4	450	96	Yes	Optimised across 450 prediction steps

It should be noted for the research conclusions that setting up a real time freerun prediction in the CSense Architect development environment is not efficient. The whole sequence of freerun prediction model formulas were hardcoded repeatedly using various variables for each freerun prediction step. This could have influenced optimisation efficiency.

The output of run 1, Figure 69, indicates that the constructed MPC could not control the process for the setpoint (red line) at set points of 830°C and 820°C. The goal function output, at the bottom of same figure, indicates that the optimiser could not optimise the model for both setpoints. This is especially visible for set point 820°C. At the set point of 840°C the model could be better optimised and the model was better controlled than the other operating sections of 820°C and 830°C. Although the set point of 840°C was not tracked 100%, the MPC application shows some ability to optimise the model in this region.

The noise in the model output indicates a weakness of the model used, as the process will not able to jump up and down so quickly in practice. It is expected that the various lags used in the latent variable construction would contribute to the inclusion of a transient response component, together with the various lags introduced by the latent variable construction. From the results it is however clear that there is no smooth transient response component included in the model. Inclusion of such a component should be investigated.

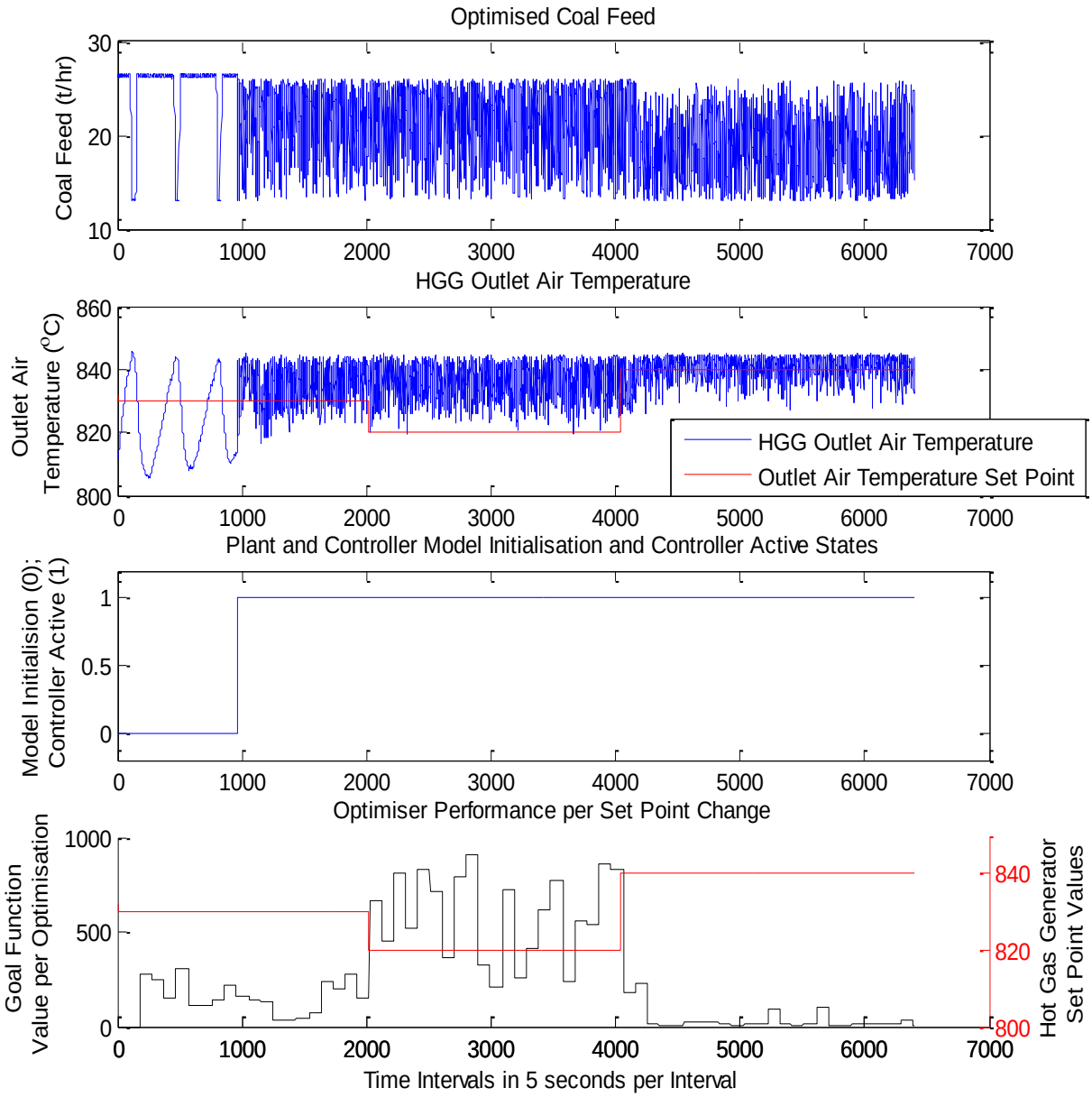


Figure 69: MPC output for run 1 indicate poor control of the process for the lower output, but better control at the higher set point.

Run 2, Figure 70, indicates similar step results for the goal function output compared to the change in set point. However the value of the goal function is visually higher than run 1 in all cases, indicating worse optimiser performance for the recursive model script used. This could be due to the unstable freerun prediction, which is to be expected. The current prediction window is 450 time steps. From the freerun prediction stability analysis in

section 9.1, the model freerun prediction started to deteriorate at 153 time steps. The model freerun prediction ability is thus too unstable for MPC, as can be seen from the MPC results.

Once again the lack of a transient response component is notable from the sudden peaks and drops in the outlet air temperature.

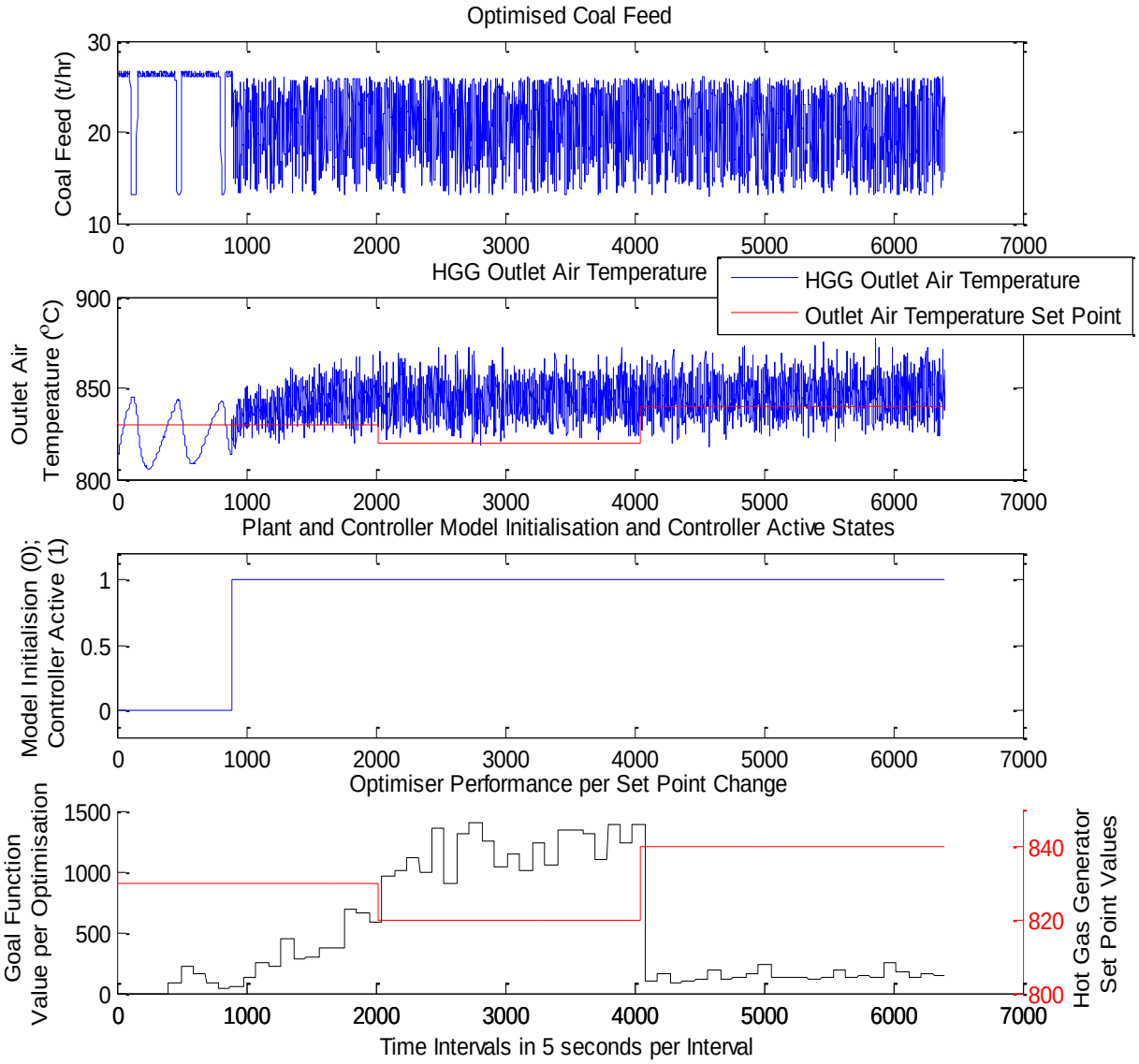


Figure 70: MPC output for run 2 indicates poor ability to control the process and higher goal function outputs.

Comparison of the average outlet air temperature for each set point across all the runs, Figure 71, indicate that both runs 1 and 2 could not control the process at the set points 820°

and 830°C. Run 2 was not able to optimise the process at the 840°C set point either. The figure however indicates that run 1 was able to control the process at the set point for 840°C.

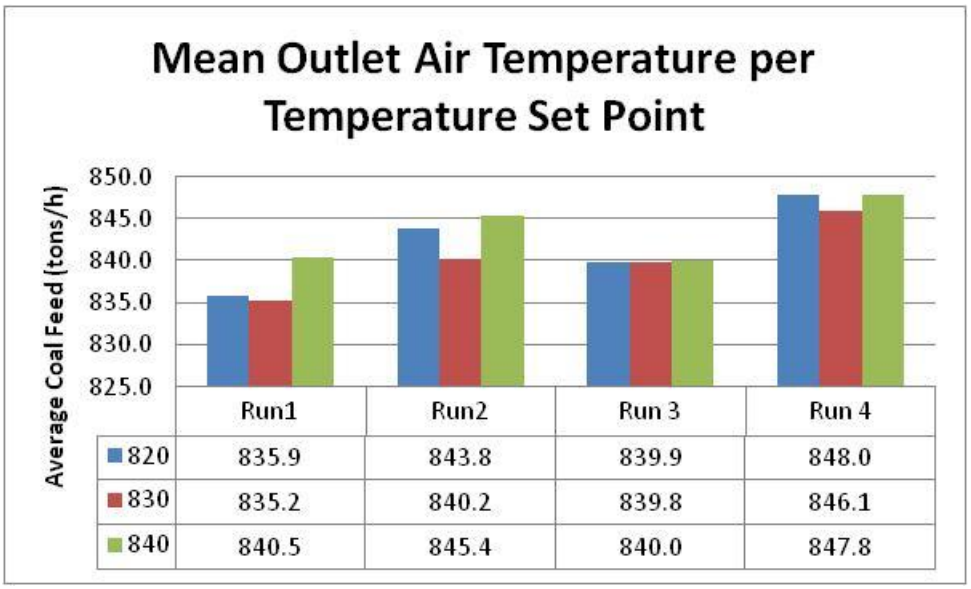
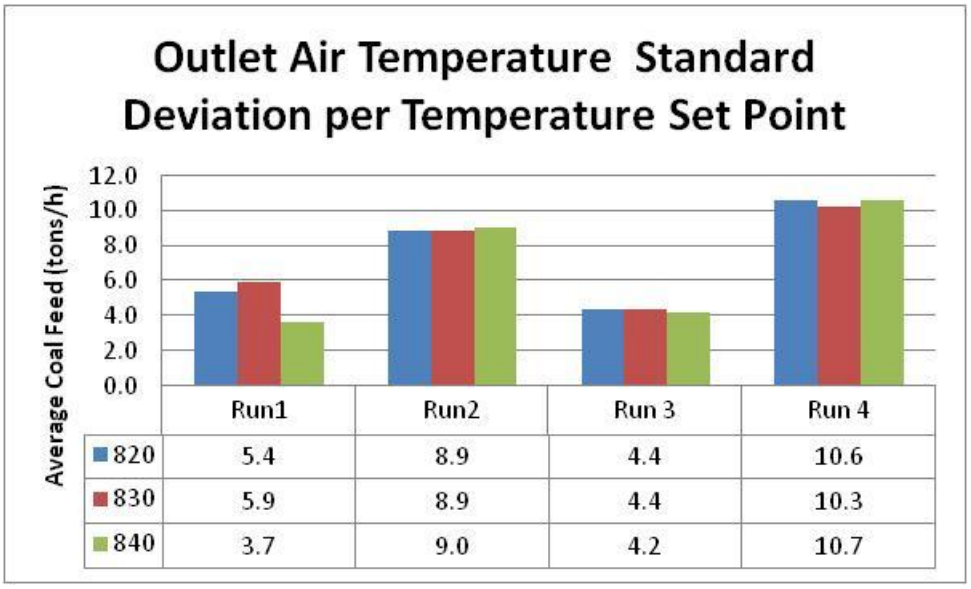


Figure 71: Average outlet air temperature for each set point across all the runs

A selection of random control moves were generated for the constructions of runs 1 and 2. These were named run 3 and run 4, with the first being a replication of run 1, and the latter, of run 2. The randomly generated control moves indicate that the model in run 1 and 3 has a lower normal output than the model used for runs 2 and 4. This could indicate that the model in run 2 could not be optimised at the lower set points due to the models natural inclination to predict higher values. The same argument could explain why the run 1 could optimise good at 840°C as the model's natural inclination is to settle at 840°C. Figure 72 below shows that the standard deviation for the section best controlled, run 1 at 840°C, has the lowest standard deviation. The poor controller performance for run 2 is further indicated by the high standard deviation of the outlet air temperature for all the set points investigated.



	Run1	Run2	Run 3	Run 4
820	5.4	8.9	4.4	10.6
830	5.9	8.9	4.4	10.3
840	3.7	9.0	4.2	10.7

Figure 72: Standard deviation of the outlet air temperature for each set point across all the runs

It is concluded that the recursive nonlinear model structure is too complex with too many degrees of freedom to optimise and the optimiser used is too weak to optimise the given problem. Another optimiser should be investigated. This requires the MPC architecture and solution used to be reconstructed on another platform.

The lack of a smooth transient response component also makes the HGG process representation by the models questionable. The models are anyway rendered obsolete due to the inverse process output actions identified during the sensitivity analysis. The inclusion of a transient response component should be investigated.

9.4 Comparison to Current Plant Controller

The current live plant controller is the IMC setup discussed earlier. The controller is expected to follow a set point constantly at 900°C. The mean and standard deviation of the best performing section of the MPC simulation above is compared to the results from the IMC. MPC Run1 controlled at 840°C is compared to the live IMC on site. It is clear from the comparison of the mean values and standard deviations, in Figure 73, that the current live IMC outperforms the best of the MPC simulations.

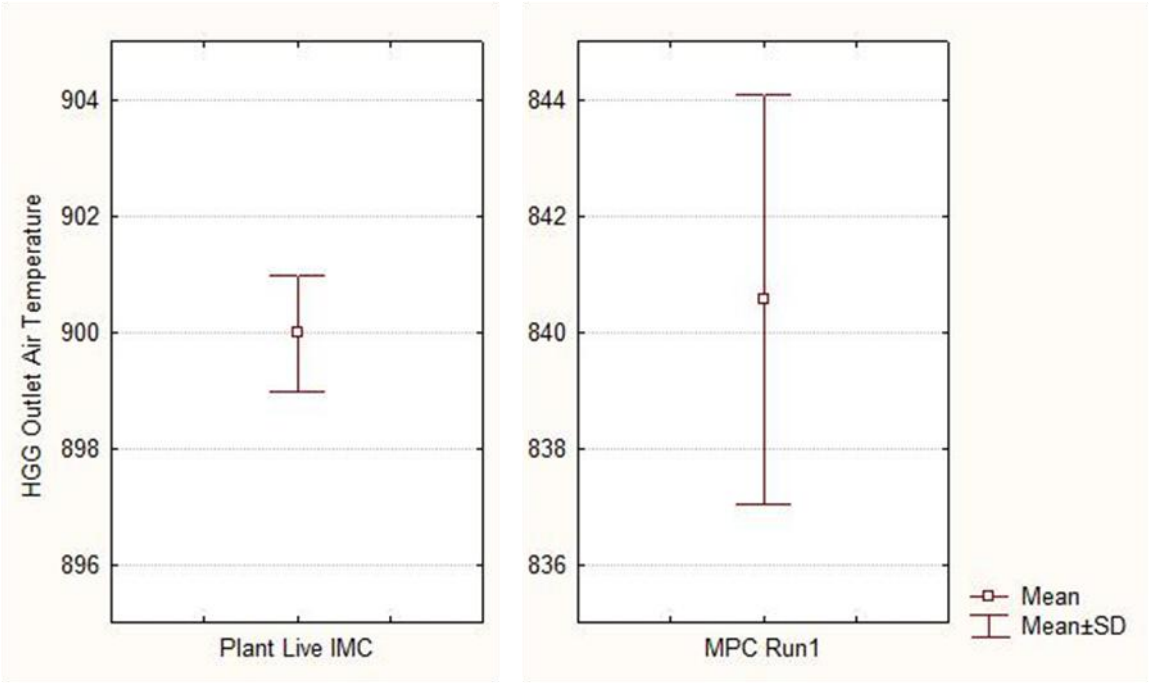


Figure 73: The current IMC controller for the HGG operation live on site, performs better than the best control found during the MPC simulations.

Table 39: Comparison of means and standard deviations indicate that the current live IMC outperforms the best of the MPC simulations.

Controller	Set Point	Mean	Standard Deviation
Live IMC	900	899.98	1.005
MPC Simulation Run1	840	840.56	3.536

9.5 MPC Conclusions

The hot gas generator resulted in promising models from the SID exercise. These models were investigated for stable freerun prediction, sensitivity reaction to stepped inputs and the final chosen models were developed into an MPC solution.

The following conclusions are made regarding the MPC approach:

- Sensitivity Analysis is crucial to identification of the models for MPC. The sensitivity analysis indicated model 1's inability to adjust the process settling temperature, despite having the best model fit statistics;
- The models identified contained inverted process dynamics. This could be due to the inclusion of various delayed versions of the coal feed or lack of closed loop identification methods in the approach used;

- Testing of model freerun prediction ability is crucial in choosing a model for MPC. Model 3 was rejected based on freerun prediction analysis. The freerun prediction performance possibly also identifies why model 4 could not be optimised, as it was unstable after 153 time intervals into the freerun prediction, but was applied in a MPC with a prediction window of 450 time intervals;
- The use of the latent variable construction was easily configurable and is found appropriate for real time implementation. Future modelling using latent variable construction can definitely be considered;
- No true conclusion can be made with regards to using the GA in the controller, seeing as the model freerun ability is questionable. It is noted that the complex model structure together with the long prediction windows might have contributed to the controllers inability to optimise the more stable model 2;
- The CSense development environment does not allow efficient freerun prediction for the model structure used. This should be investigated for future real time implementations if MPC is considered with such a formula based model.
  (It is noted that a solution can be constructed in various coding languages for the software. The user friendliness for any process engineer is the trade off in such a case);
- The current IMC controller performs better than the developed MPC;

The solution is thus found infeasible and overly complex for a real time solution. It is recommended that linear models, or piece-wise linearisation of models be investigated for the models identified. It is also recommended that open loop step test data be used to insure the data are valid for system identification.

Chapter 10 Conclusion

The aim of this research was to investigate a concentrate dryer operation for possible control strategies. This investigation entailed research of current dryer literature to identify suitable control strategies, as well as a system identification methodology. This SID methodology was used to extract information from the historic dryer operation data in an attempt to identify a nonlinear dynamic model to be used in an advanced process control strategy, specifically a model predictive controller.

The contribution of this study is thus aimed at system identification of the unknown model structures for dryers using a conglomeration of system identification techniques, to identify a process model for the flash dryer operations. The resulting model indicates which area of the dryer is best suited for modelling and control. This model is also used in developing, constructing and testing the use of this model in an advanced control strategy.

The main conclusion based on the problem statement is:

> The historian data could not be used for system identification of the flash dryer model. The hot gas generator could not be modelled, despite the fact the R^2 model fit was above 70%. Inverted dynamics were recognised rendering the HGG model inaccurate. The lack of a model hampered proper testing of an MPC controller, although the data preparation and use the empirical NARX model in an MPC application for a implementation in CSense could be tested. This approach was found to be limited by the optimisability of the unstable freerun prediction ability of the model by the supplied GP.

The main case study results with regards to the dryer study include the following:

1. **Model based control is the preferred dryer control strategy**

   Model based control, and specifically MPC, is identified from literature as being the control strategy best suited to the dryer operation's nonlinear behaviour and long dead times. MPC is also capable in handling multivariate problems common in dryer operations.

2. **Dryer dynamics are complex and misunderstood, requiring a SID strategy and algorithm capable of identifying unknown model structures**

   GP is a possible solution, amongst others, capable of identifying both model structures and parameters if the correct data are provided, and is a favoured approach for unknown model structures. This technique was combined with a latent variable construction method, allowing investigation of unknown process lags and construction of a nonlinear dynamic model to be identified.

3. **Process measurements investigated in literature are not available in practice**

   The lack of process instrumentation in practice for accurately measuring concentrate temperature and moisture, in and out of the dryer, as well as the drying air humidity, hampers both the accuracy of the system identification results, as well as the controller efficiency. Absence of these variables result in lack of influential dynamics as well as the construction of a noise model.

4. **Latent variable construction of the subdivided timeseries allowed identification of dynamic models**

   The creation of various lagged variables using delay parameters identified, allowed the time delays in dryer dynamics to be included in the model. The structured identification of the delay parameters allowed weak inter-correlated lagged variables to be investigated for process dynamics. Poor selection of these embedded variables would have resulted in possible strongly correlated input variables to compete for the same position, wasting algorithm energy. The latent variable construction is also easily implemented in a real time solution in the CSense development environment.

5. **No representative flash dryer model could be identified from the data**

   The GPOls method, with latent variable construction of the input and output variables, could not identify a model with good enough fit statistics to represent the flash dryer process. Temperature spikes due to feed stoppages could be identified, however the other dynamics were predominantly missed.

6. **Models for the hot gas generator could be identified and validated well, but were found to lack the proper dynamics during the sensitivity analysis**

   SISO models, due to lack of variation in the fluidised damper variable, were identified successfully for the hot gas generator. These models validated well against the historic data and showed stable freerun prediction. However, inverse reactions to coal feed occurred with outlet air settling at a lower temperature when coal was fed; and vice verse. The models could thus not be accepted to represent the process. Suitability of these models for use in a MPC strategy was however investigated, despite the misrepresentation of the process.

7. **GA optimiser could not optimise the freerun HGG model**

   GA optimiser could not optimise over the prediction window seeing as the freerun prediction stability was poor. The long prediction window required to include settling time was longer than the freerun prediction ability of the model. This proved detrimental in that the model could not be optimised. The more stable freerun model, model #2, indicated some ability to be optimised in a specific region, but the complex model structure is expected to have hampered the optimisation.

As system identification entails the largest part of this study, with efforts and adjustments to the approach occurring frequently in an attempt to identify flash dryer models, some of the conclusions with regards to the system identification approach are presented here:

1. **The GPOls toolbox obtains results comparable with Discipulus Lite®**

   The benchmarking exercise between the GPOls algorithm and the commercially available Discipulus Lite®, indicates comparable results with Discipulus Lite® outperforming according to fit statistics. The GPOls results are however still within a comparable order of magnitude, and can thus be used with confidence.

2. **The GPOls toolbox outperformed the Linear ARMA Model**

   The GP approach performed better than linear ARMA models constructed based on the latent variable reconstruction. These models are cheaper, but performed worse

than the GP approach, indicating the nonlinear system identification methods required for dryer modelling.

3. **The advanced nonlinear functional set is favoured for SID with the GP**

   The advanced functional set, which contain the square root and exponent functions, proved to be favoured over the more basic functional set. This is ascribed to the expected nonlinearities present in the dynamics of the dryer.

4. **The induced bias is preferred over the square root of the absolute value given a fitness function adjustment**

   The biased dataset, in an attempt to prevent non-real numbers, produced models with better fit statistics when used with the square root, than the absolute value. It was found imperative to still adjust the fitness function to penalise non-real numbers due to the square root of a conglomeration of functions. This combination of bias and adjusted penalty function proved to work well with the GPOls algorithm.

5. **The use of a predefined population contributed to system identification**

   The use of previous populations as a base to start a search from allowed the search to be both directed and make use of earlier breakthroughs by previous GP experiments. The GP algorithm was not consistently able to reach a global optimum, but departure of the search from this “heightened” population allowed earlier breakthroughs to be harnessed. This functionality was only used during system identification of the flash dryer model, seeing as it provided difficult dynamics to identify.

Chapter 11 Recommendations

Based on the conclusion discussed previously as well as the inability of the system identification step to identify a representative model, according to both validation fit statistics and expected response, the following recommendations are made for future experiments for dryer modelling and control.

1. **Investigate the SID and MPC methodology on a known model**

   The methodology, application of the method of latent variable construction with GP, should be tested against a known nonlinear model. The result should be implemented as is done in this study to establish the validity of the method.

2. **Investigate better process measurements through either instrumentation or the use of "soft sensors"**

   The lack of feed and product temperatures and moistures hamper modelling and accurate control. The measurements of air humidity should also be investigated. The inclusion of measurement could be investigated by comparison to a fundamental dryer model for the specific dryer and comparing the results of the data and system identification from the data. Literature indicates these sensors are unable to handle the harsh dryer conditions, thus implying the investigation of air humidity model based ("soft") sensors as an option.

3. **Investigate the inexplicable Flash Dryer dynamics noted in this research**

   Investigate the occurrences of the misunderstood sections of dynamics, named "idle" and "anomaly" in this research, in an attempt to understand what caused these dynamics in the data.

4. **Investigate whether closed loop system identification would result in NON-inverted models**

   The identified models showed inverse process dynamics. It could possibly be ascribed to the data being collected under closed loop control. The incorporation of closed loop SID techniques should be investigated for inclusion in identification of the HGG process model.

5. **Investigate inclusion of a transient response component**

   The steps in the model outputs are very sudden for the empirical model identified by the GP. The incorporation of a transient response component for the output should be investigated as it might better represent the steps in the process.

6. **Investigate the use of the ARMA linear models for use in a MPC**

   The linear ARMA models identified should be investigated for use in a MPC, as it does not carry the baggage of complex optimisations, but still allow representation of the model. This should also result in a more basic freerun prediction structure in the software environment used.

7. **Investigate a GP fitness function which measures the freerun ability of the identified model**

   Investigate an alternate fitness function for the GP which includes the freerun prediction ability of an individual. This would allow identification of a model best fit for freerun prediction and can be penalised based on the length of freerun required. This would however require longer modelling times, but might be beneficial if a representative model could be identified able of stable freerun prediction.

References

Abdel-Jabbar, N.M., Jumah, R.Y. & Al-Haj Ali, M.Q. (2002) Multivariable Process Identification and Control of Continuous Fluidized Bed Dryers. *Drying Technology*, Vol. 20(7), pp.1347-77.

Abudkhalifeh, H., Dhib, R. & Fayed, M.E. (2005) Model Predictive Control of an Infrared-Convective Dryer. *Drying Technology*, Vol. 23, pp.497-511.

Arjona, R., Olilero, P. & Vidal, B.F. (2005) Automation of an Olive Waste Industrial Dryer. *Journal of Food Engineering*, (68), pp.239-47.

Barnard, J.P. & Aldrich, C. (2001) A Systematic Methodology for Empirical Modeling of Nonlinear State Space Systems. *Computer Aided Chemical Engineering*, Vol. 9, pp.75-80.

Barnard, J.P. & Aldrich, C. (2002) Chapter 10 Embedding of Multivariate Dynamic Process Ssytems. *Process Metallurgy*, Vol. 12, pp.299-312.

Chen, S. (2006) Local Regularization Assisted Orthogonal Least Squares Regression. *Neurocomputing*, Vol. 69, pp.559-85.

Coelho, L.S. & Pessôa, M.W. (2009) Nonlinear Model Identification of an Experimental Ball-and-Tube System Using a Genetic Programming Approach. *Mechanical systems and signal processing*, Vol. 23, pp.1434-46.

CSense Pty Ltd (2007) **Advanced Process Control for HGG and Flash Drying Plant**. Pretoria: CSense Systems (Pty) Ltd CSense Systems (Pty) Ltd.

CSense Pty Ltd (2007) **Advanced Process Control on Flash Drying Plant: Scoping Study**. Pretoria: CSense Systems (Pty) Ltd CSense Systems (Pty) Ltd.

De Falco, I., Iazzetta, A., Tarantino, E. & Della Cioppa, A. (2000) An Evolutionary System for Automatic Explicit Rule Extraction. *2000 Congress on Evolutionary Computation*, pp.450-57.

De Temmerman, J., Dufour, P. & Nicola (2009) MPC as Control Strategy for Pasta Drying Processes. *Computers and Chemical Engineering*, Vol. 33, pp.50-57.

Didriksen, H. (2002) Model Based Predictive Control of a Rotary Drier. *Chemical Engineering Journal*, (86), pp.53-60.

Duchesne, C., Thibault, J. & Bazin, C. (1997) Dynamics and Assessment of Some Control Strategies of a Simulated Industrial Rotary Dryer. *Drying Technology*, Vol. 15(2), pp.477-510.

Dufour, P. (2006) Control Engineering in Drying Technology: Review and Trends. *Drying Technology*, (24:7), pp.883-904.

Dufour, P., Toure, Y., Blanc, D. & Laurent, P. (2003) On Nonlinear Distributed Parameter Model Predictive Control Strategy: On-line Calculation time reduction and application to an experimental drying process. *Computers and Chemical Engineering*, (27), pp.1533-42.

Elsey, J. et al. (1997) Modelling and Control of a Food Extrusion Process. *Computers and Chemical Engineering*, (21), pp.S361-66.

Francone, F.D. (2001) *Discipulus Linear Genetic Programming Software: How it Works*. [Online] Available at: http://www.rmltech.com/Discipulus%20How%20It%20Works.pdf [Accessed January 2009].

Francone, F.D. (2001) **Discipulus: Owner's Manual**. Littleton, Colorado, USA: Register Machine Learning Technologies, Inc.

Greeff, D.J. & Aldrich, C. (1998) Empirical Modelling of Chemical Process Systems with Evolutionary Programming. *Computers in Chemcial Engineering*, Vol. 22(7-8), pp.905-1005.

Grosman, B. & Lewin, D.R. (2002) Automated Nonlinear Model Predictive Control using Genetic Programming. *Conputers and Chemical Engineering*, (26), pp.631-40.

Grosman, B. & Lewin, R. (2004) Adaptive Genetic Programming for Steady-State Process Modeling. *Computers and Chemical Engineering*, (28), pp.2779-90.

Hinchliffe, M.P. & Willis, J.W. (2003) Dynamic Systems Modelling using Genetic Programming. *Computers and Chemical Engineering*, (27), pp.1841-54.

Hjalmarsson, H., Gevers, M. & De Bruyne, F. (1996) For Model-based Control Design, Closed-Loop Identification. *Automatica*, Vol. 32(12), pp.1659-73.

Holmberg, H. & Athila, P. (2006) Simulation Model for the Model-Based Control of a Biofuel Dryer at an Industrial Combined Heat and Power Plant. *Drying Technology*, Vol. 24, pp.1547-57.

Huang, B. & Mujumdar, A.S. (1992) **Prediction of Industrial Dryer Performance using Neural Networks**. In *Proceedings of the 8th International Drying Symposium Part B*. Montreal.

Hussain, A.E. et al. (2000) Modeling of a Winding Machine using Genetic Programming. *2000 Congress on Evolutionary Conputation*, pp.398-402.

Korn, O. (2001) Cyclone Dryer: A Pneumatic Dryer with Increased Solid Residence Time. *Drying Technology*, Vol. 19(8), pp.1925-37.

Landelahni (2010) **Landelahni Mining Survey 2010**. Annual Survey. Johannesburg: Landelahni.

Liu, Q. & Bakker-Arkema, F.W. (2001) A mocel predictive control of grain drying. *Journal of Food Engineering*, (49), pp.321-26.

Ljung, L. (1999) **System Identification - Theory for the user Second Edition**. Upper Saddle River, New Jersey: Prentice Hall.

Ljung, L. (2006) **Identification of Nonlinear Systems**. In *Proceedings of the 9th International Conference on Control, Automation, Robotics and Vision (ICARCV 2006)*. Singapore IEEE.

Madar, J. (2005) **GP-OLS**. [Online] Available at: http://www.fmt.vein.hu/softcomp/gp/gpols.html [Accessed 11 October 2010].

Madar, J., Abonyi, J. & Szeifert, F. (2005) Genetic Programming for the Identification of Nonlinear Input-Ouput Models. *Industrial & Engineering Chemistry Research*, Vol. 44, pp.3178-86.

McKay, B., Willis, M. & Barton, G. (1997) Steady State Modelling of Chemical Process Systems using Genetic Programming. *Computers in Chemical Engineering*, Vol. 21(9), pp.981-96.

Morari, M. & Zafiriou, E. (1989) **Robust Process Control**. Englewood Cliffs, New Jersey, United States of America: Prentice-Hall Inc.

Mujumdar, A.S., ed. (1995) **Handbook of Industrial Drying**. 2nd ed. New York: Marcel Dekker.

Mujumdar, A.S. (2004) Research and Development in Drying: Recent Trends and Future Prospects. *Drying Technology*, Vol. 22(1&2), pp.1-26.

Mujumdar, A.S. & Huang, L.X. (2007) Global R&D Needs in Drying. *Drying Technology*, Vol. 25, pp.647-58.

Mujumdar, A.S. & Zhonghua, W. (2008) Thermal Drying Technologies - Cost Effective Innovation Aided by Mathematical Modeling Approach. *Drying Technology*, Vol. 26, pp.145-53.

Perry, R.H. & Green, D.W. (1997) **Perry's Chemical Engineering Handbook**. McGraw-Hill. pp.Sections 8,12.

Ramesh, K., Abd Shukor, S.R. & Aziz, N. (2009) **Nonlinear Model Predictive Control of a Distillation Column Using NARX Model**. In *10th International Symposium on Process Systems Engineering*. Sau Paulo Elsevier.

Rossiter, J.A. (2003) **Model Based Predictive Control: a Practical Approach**. Boca Raton, Florida, United States of America: CRC Press.

SAIMC (2008) *Advanced Process Control - APC: CSENSE Intelligence Improves Smelter Efficiency at Lonmin*. [Online] Available at: http://instrumentation.co.za/article.aspx?pklArticleId=5288&pklCategoryId=71 [Accessed July 2009].

Sjöberg, J. et al. (1995) Nonlinear Black-box Modelling in System Identification: a Unified Overview. *Automatica*, Vol. 31(12), pp.1691-724.

South African Department of Minerals and Energy (2007) *Department of Minerals and Energy:Minerals*. [Online] Available at: http://www.dme.gov.za/pdfs/minerals/B1%20Stat%20Tables%202008.xls [Accessed 18 March 2010].

South African Department of Minerals and Energy (2008) *South Africa's Mineral Industry 2007/08*. [Online] Available at: http://www.dme.gov.za/minerals/mineral_stats.stm [Accessed 18 March 2010].

Trelea, I., Courtois, F. & Trystram, G. (1997) Dynamics Analysis and Control Strategy for a Mixed Flow Corn Dryer. *Journal of Process Control*, (7), pp.57-64.

Van Schalkwyk, T. (2009) CSense APC Implimentation at Lonmin Flashdryer. *Interview*.

Wang, W., Chen, G. & Mujumdar, A.S. (2007) Physical Interpretation of Solids Drying: An Overview on Mathematical Modeling Research. *Drying Technology*, Vol. 25, pp.659-68.

Willis, M.J. et al. (1997) Systems Modelling using Genetic Programming. *Conputers and Chemical Engineering*, (21), pp.S1161-66.

Winkler, S., Affenzeller, M. & Wagner, S. (2004) Identifying Nonlinear Model Structures Using Genetic Programming Techniques. *Cybernetics and Systems*, pp.689-94.

Young, B.R. (2008) **Food Drier Process Control**. In Chen, X.D. & Mujumdar, A.S. *Drying Technologies in Process Control*. 1st ed. Oxford: Blackwell Publishers. pp.270-98.

Zulkeflee, S.A. & Aziz, N. (2009) **NARX-Model-Based Control (NARX-MBC) for Citronellyl Laurate Esterification Reactor**. In *10th International Symposium on Process Systems Engineering*. Sao Paulo Elsevier.

Appendix A – Normalisation Parameters

The means and standard deviations used for normalisation of each dataset is included in this appendix:

B.1. Flash Dryer

Idle States Removed

Flash Dryer Variable	Dataset Number	Mean	Standard Deviation
Concentrate Feed (ton/hour)	1	28.932	15.359
	2	23.146	10.958
	3	36.659	10.257
Flash Dryer Inlet Temperature (°C)	1	894.408	41.233
	2	900.564	11.480
	3	901.551	5.806
Flash Dryer Outlet Temperature (°C)	1	129.979	25.795
	2	137.467	17.380
	3	110.503	13.457

Idle States Present

Flash Dryer Variable	Dataset Number	Mean	Standard Deviation
Concentrate Feed (ton/hour)	1	28.932	15.359
	3	28.848	16.929
Flash Dryer Inlet Temperature (°C)	1	894.408	41.233
	3	894.888	14.183
Flash Dryer Outlet Temperature (°C)	1	129.979	25.795
	3	111.625	21.775

B.2. Hot Gas Generator

No distinction was made regarding states

Hot Gas Generator Variable	Dataset Number	Mean	Standard Deviation
Coal Feed (ton/hour)	1	23.515	5.194
Hot Gas Generator Outlet Temperature (°C)	1	827.430	15.634

Appendix B – Dataset Reduction

This addendum contains the detailed procedure and results for dataset reduction.

The basis on which datasets will be filtered will initially be by visual inspection. This is humanly possible due to the few variables involved and knowledge of the expected dynamics. To accomplish this the shortest timeseries need to be deleted and the remaining timeseries should be displayed visually. 3000 data points was the cut off for hot gas generator timeseries and 7000 data points was the cut off for flash dryer timeseries. These values were selected based on visual inspection of the available subdivided dataset lengths.

If the visual inspection could not distinguish the best timeseries, then the nonlinear system identification methodology will be used. Distinguishing between usable datasets can be done on the grounds of these findings.

Two data preparation approaches result in two different divisions of the timeseries. As discussed in the previous section the one approach divides the timeseries according to the APC status and the data gaps. The second approach only takes the data gaps into account, assuming the controllers will not significantly alter the process dynamics.

B.1. Flash Dryer Datasets

B.1.1. Timeseries Subdivided According to APC Status and Data Gaps

After initial inspection of the sub-datasets it was decided to only use datasets with more than 1000 data points. The remaining datasets will be analysed visually. This was found to be sufficient as, what is perceived by the researcher as a manageable amount of datasets, were left over. For the FD model 6 datasets remained. The resulting datasets are included in Figure 74 below.

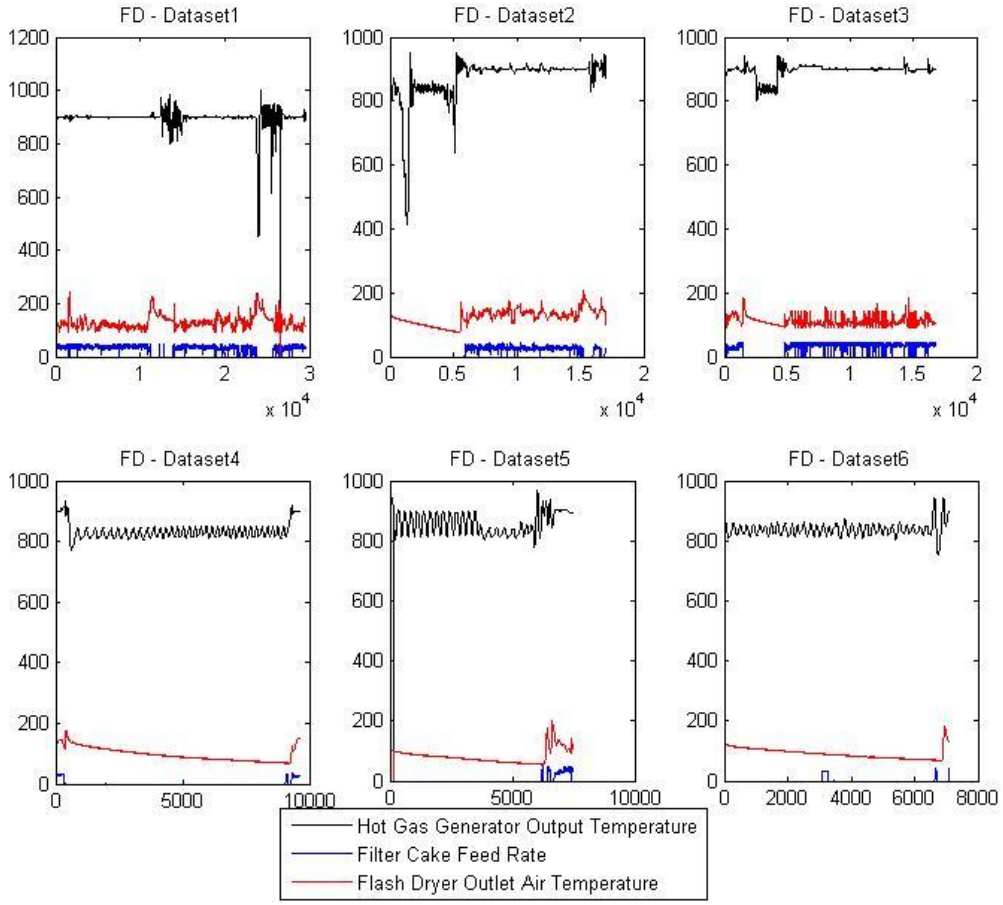


Figure 74: Flash Dryer datasets identified for modelling based on APC status and timeseries length

The flash dryer datasets chosen for possible modelling includes six timeseries datasets, but with less consistent and determinable dynamics than the hot gas generator datasets. The number of data points per dataset is:

Table 40: Length of subdivided timeseries datasets for the flash dryer; measured in number of data points

Dataset Number	Number of Data Points
1	29628
2	17055
3	16728
4	9589
5	7479
6	7071

There is thus a dramatic drop in the length of continuous timeseries when looking at the subdivided datasets. It is apparent that the first three, and longest, datasets will be chosen based on length. Further visual inspection of the process dynamics in Figure 74 indicate

that datasets 4, 5 and 6 have periods where the flash dryer temperature gradually degrades and contains very little oscillating dynamics as is seen in datasets 1, 2 and 3. From visual inspection of the trend of the full flash dryer dataset, Section 4.3.3 - Figure 19, combined with the trends above, it is clear that these sections of gradual degradation occur when the hot gas generator APC was switched off and when the process was in idle. At this time the filter cake feed was also off. Although this information of slow temperature drop during idling might contribute to the process model, it is of no use only modelling this, as will be done if dataset 4, 5 and 6 are used. Furthermore dataset 2 starts off with a idle state. This situation will need definite starting conditions which cannot be assumed to exist at the start of the dataset. The conclusion is made that datasets 1 and 3 will be used for modelling.

Note that these gradual degradation sections were found to hamper the model fitting to general drying circuit operations. This is discussed next in section B.1.2.

B.1.2. Timeseries Without Dryer Process IDLE state

The flash dryer timeseries identified in the previous section include sections where the flash dryer outlet air temperature decreases gradually. During this time the hot gas generator operation is stepped down, but still operational. The flash dryer feed is also stopped. Initially it was expected that including these degradation dynamics will assist in creating a more representative model. During system identification it was however found that separating this idle state allows better modelling for both the active and the idle state.

The resulting datasets for the active dynamics, only looking at datasets 1, 2 and 3 as illustrated in Figure 74, are portrayed in the following figure, Figure 75. Only datasets 2 and 3 were reduced by deleting the first 5500 and 4700 data points, identified visually, respectively.

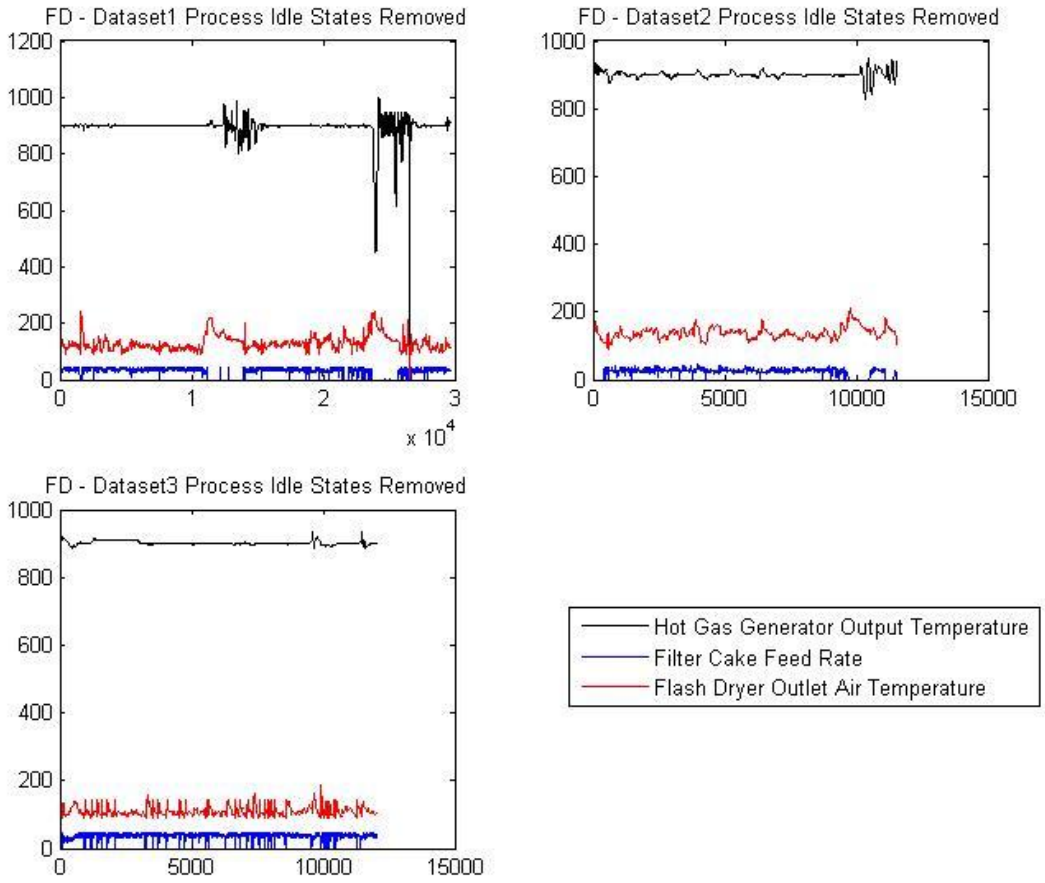


Figure 75: Flash Dryer datasets identified for modelling based on APC status, Process Idle States and timeseries length

The dataset lengths have been reduced as indicated below.

Table 41: Number of data points per dataset for flash dryer datasets with the process idle states removed

Dataset Number	Number of Data Points
1	29628
2	11556
3	12029

With the gradual degradation of the flash dryer outlet air temperature during the idle states removed, dataset 2 can also be investigated for use in system identification.

B.2. Hot Gas Generator Datasets

B.2.1. Timeseries Subdivided According to APC Status and Data Gaps

After initial inspection of the subdivided datasets it was decided to only use datasets with more than 1000 data points. The remaining datasets will be analysed visually. This was found to be sufficient as, what is perceived by the researcher as a manageable amount of datasets, were left over. For the HGG model 8 datasets remained. The resulting datasets are included in Figure 76 below.

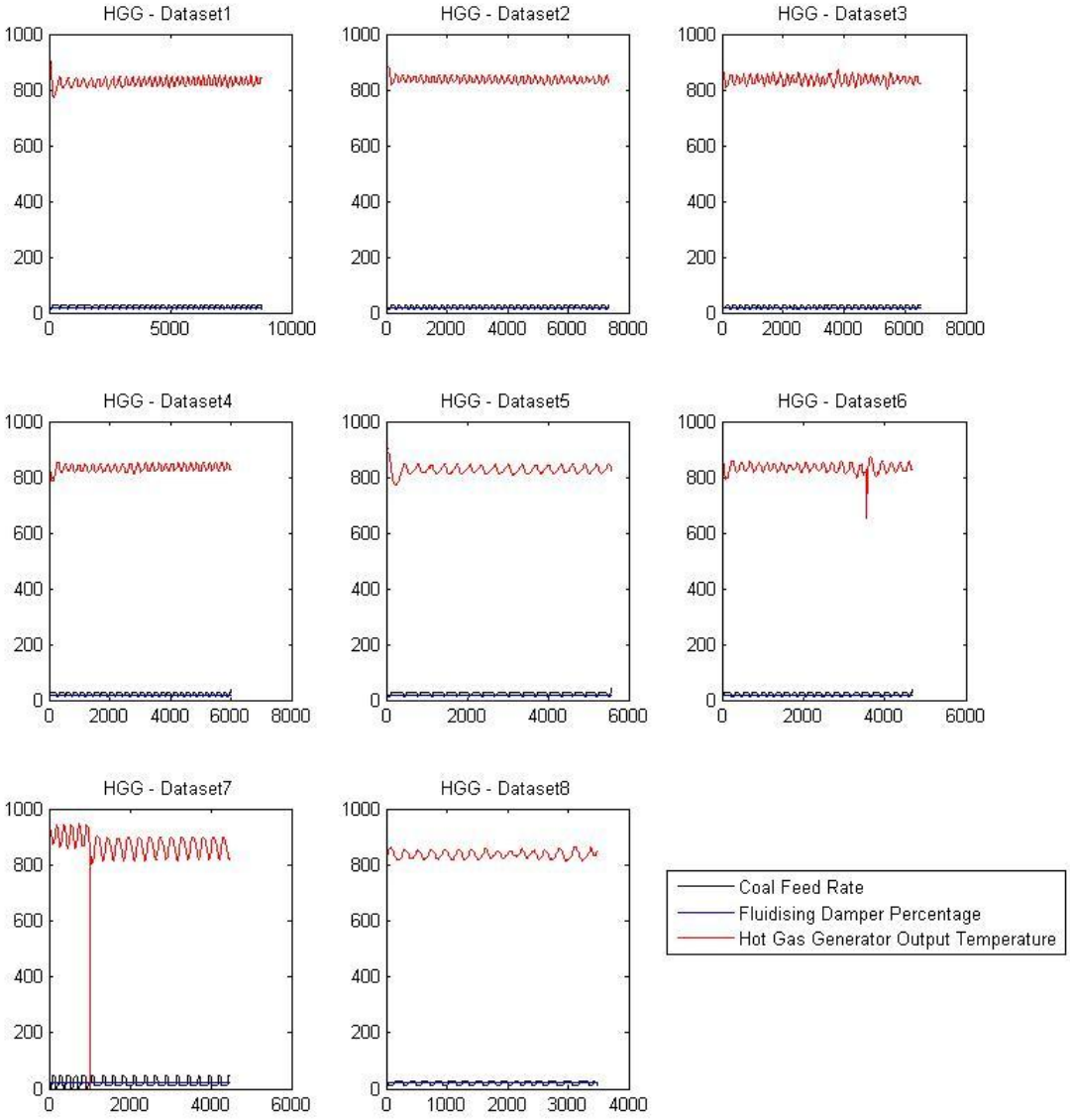


Figure 76: Hot Gas Generator datasets identified for modelling based on APC status and timeseries length

The chosen eight hot gas generator datasets clearly indicate the “bang-bang” control approach used when the APC is off. This should allow easy system identification seeing as it represents data similar to what is expected during step testing. The number of data points included in each timeseries dataset is:

Table 42: Length of subdivided timeseries datasets for the hot gas generator; measured in number of data points

Dataset Number	Number of Data Points
1	8752
2	7335
3	6492
4	6010
5	5579
6	4666
7	4470
8	3492

The average operating range is however very limited. The hot gas generator outlet temperature range is between 800 and 880°C with the exception of dataset 7 which contains a section ranging between 890 and 940°C, but then drops to the same dynamics as the other datasets for the remainder of the time. The operating range for the fluidising damper is however problematic, as it was constant throughout all the datasets.

The conclusion is made that any of the datasets can be used for modelling purposes, but the fluidising damper will need to be omitted from modelling. The longest dataset will thus be used: Dataset 1.

Further analyses would commence on these remaining datasets alone. Note that the datasets will be tested for availability of process information and dynamics by means of the surrogate data comparison. This will provide an analysis of the possibility of identifying a model from the datasets.

Appendix C – Process Output Timeseries Analysis

The complete surrogate data comparison results for all the datasets specified is included in this section. The meaning and discussion of these trends are included in sections 8.1.1.3 for the flash dryer and 0 for the hot as generator.

The analysis was done to obtain *a priori* information regarding the dynamics available in the timeseries by only looking at the output dataset. This was done by viewing the dynamic attractor in the phase space as well as the surrogate data plot against the real timeseries to determine if the process dynamics are deterministic or stochastic. This was done for each chosen dataset for the flash dryer and the hot gas generator setups separately.

C.1. Analysis 1: Flash Dryer Dataset 1

Variable under investigation: Flash Dryer Outlet Air Temperature

Process Variables	Method to Determine Delay *k*	Delay (*k*)	Number of Delayed Variables (m)
Flash Dryer Outlet Air Temperature	AMI	82	5
	Autocorrelation	623	4

Trend of the dataset under study:

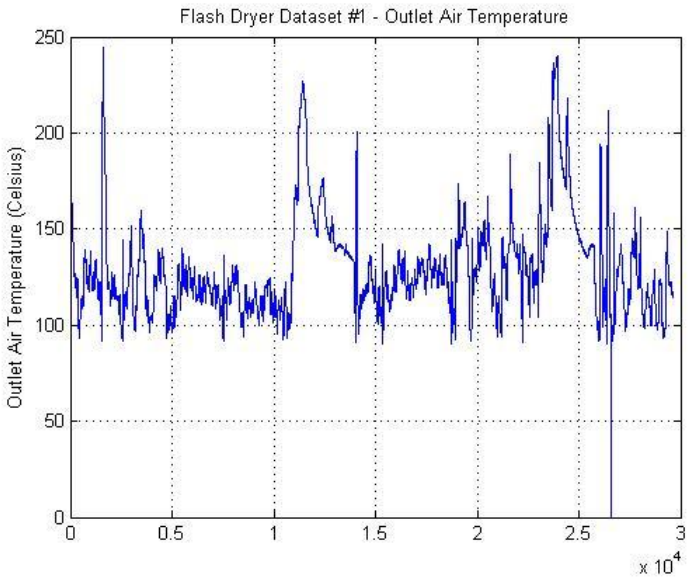


Figure 77: Flash Dryer Outlet Temperature for Dataset 1

Both sets of delay parameter pairs were analysed separately.

Trend of the surrogate data comparison:

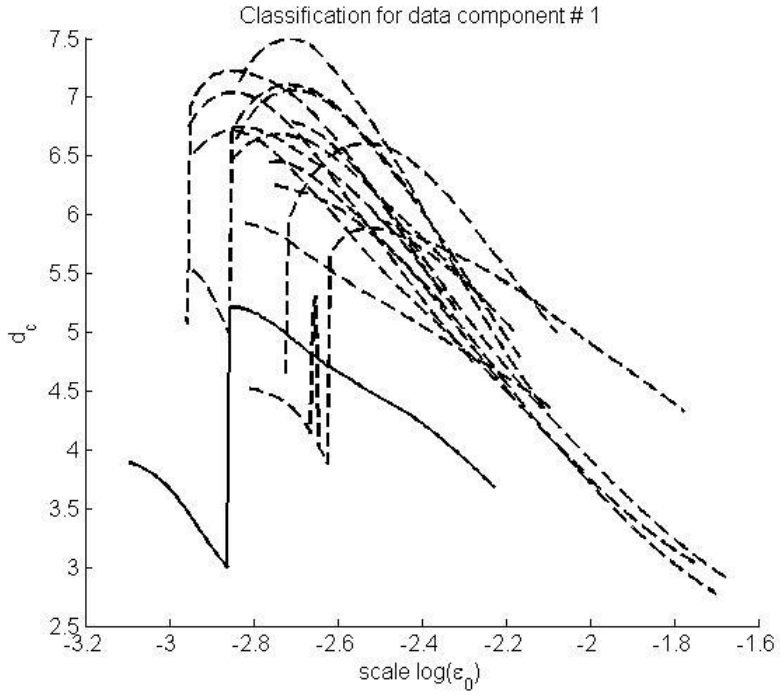


Figure 78: Surrogate Data Classification of the HGG outlet air temperature for delay parameters 82x4

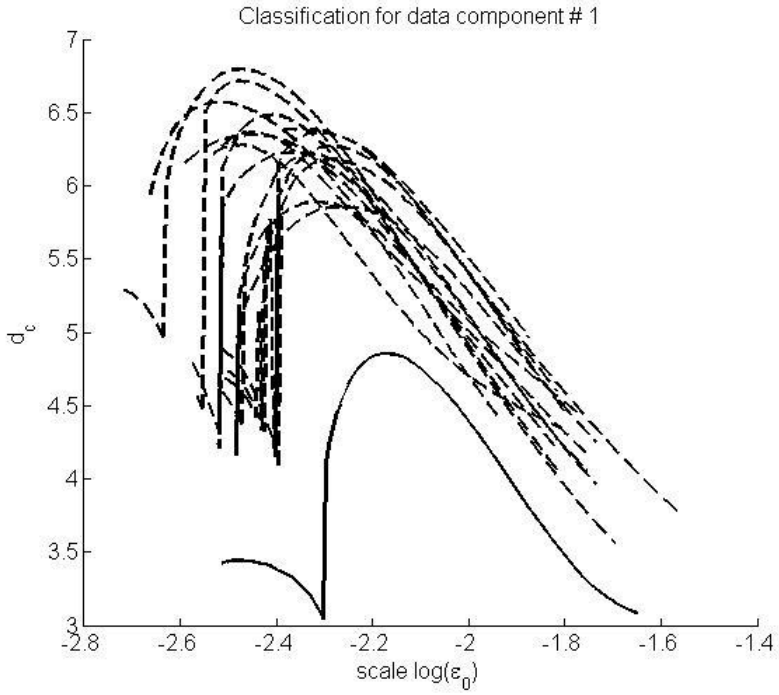


Figure 79: Surrogate Data Classification of the HGG outlet air temperature for delay parameters 623x4

The delay parameter set 623x4 is better separated from the surrogate data, although better separation should be required.

C.2. Analysis 2: Flash Dryer Dataset 1 Pre Anomaly Data

The timeseries before the anomaly is investigated separately. This is a section of dataset 1 ranging from data point 1922 to data point 10830. This interval is established visually.

Variable under investigation: Flash Dryer Outlet Air Temperature

Process Variables	Method to Determine Delay *k*	Delay (*k*)	Number of Delayed Variables (m)
Flash Dryer Outlet Air Temperature	AMI	47	4
	Autocorrelation	273	Too few data points in dataset to calculate this dimension

Trend of the dataset under study: (Boxed area is investigated)

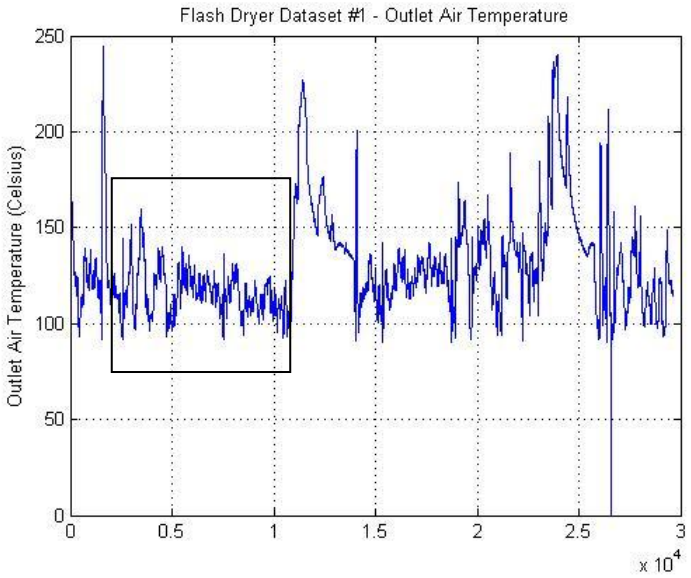


Figure 80: Flash Dryer Outlet Temperature for Dataset 1 before anomaly 1

Only the set of delay parameters identified by AMI and FNN were analysed.

Trend of the surrogate data comparison:

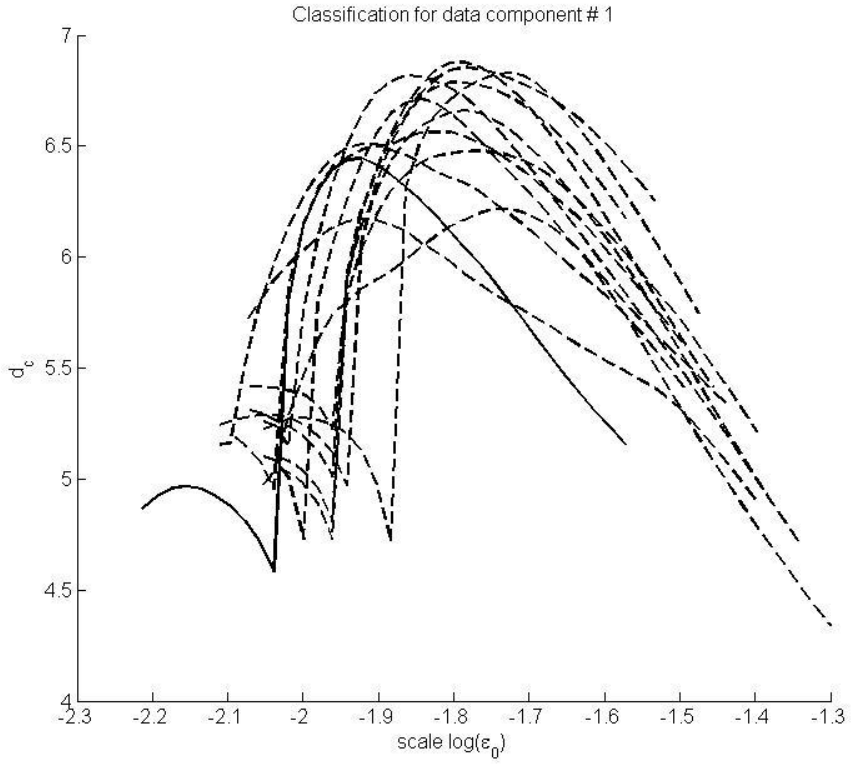


Figure 81: Surrogate Data Classification of the FD outlet air temperature for delay parameters 82x4

The surrogate data analysis indicates that the timeseries could not be differentiated from stochastically driven data and is thus not best suited for system identification.

C.3. Analysis 3: Flash Dryer Dataset 1 Post Anomaly Data

The timeseries before the anomaly is investigated separately. This is a section of dataset 1 ranging from data point 14290 to data point 22950. This interval is established visually.

Variable under investigation: Flash Dryer Outlet Air Temperature

Process Variables	Method to Determine Delay *k*	Delay (*k*)	Number of Delayed Variables (m)
Flash Dryer Outlet Air Temperature	AMI	52	4
	Autocorrelation	368	Too few data points in dataset to calculate this dimension

Trend of the dataset under study: (Boxed area is investigated)

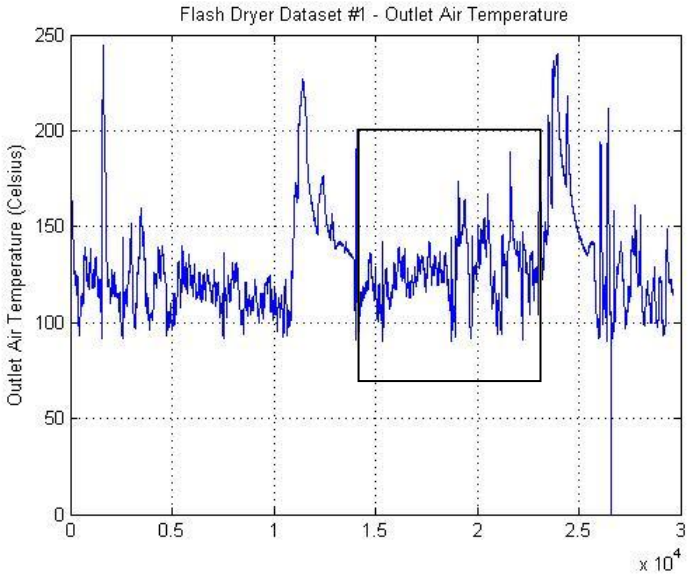


Figure 82: Flash Dryer Outlet Temperature for Dataset 1 after anomaly 1

Only the set of delay parameters identified by AMI and FNN were analysed.

Trend of the surrogate data comparison:

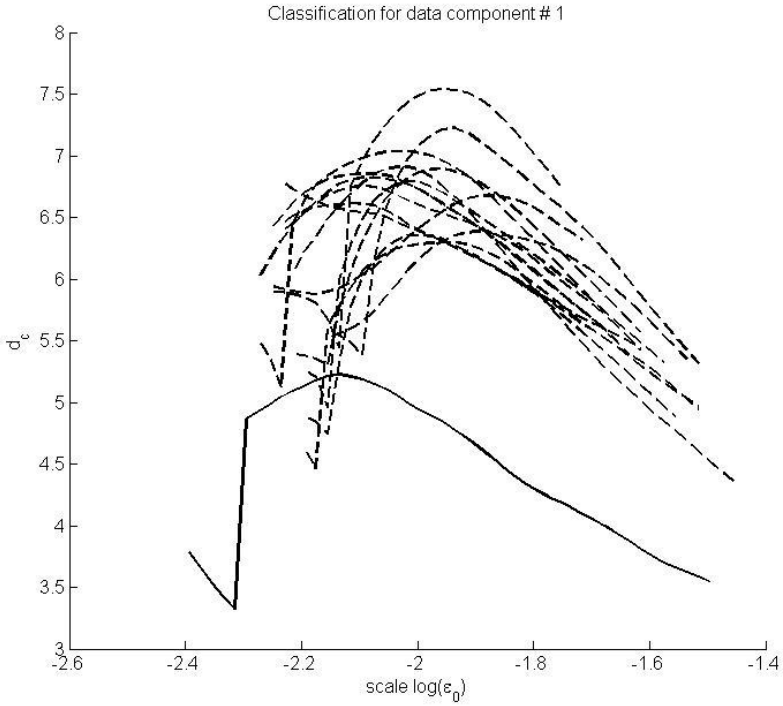


Figure 83: Surrogate Data Classification of the FD outlet air temperature for delay parameters 52x4

The surrogate data analysis could identify some deterministic data in the datasets.

C.4. Analysis 4: Flash Dryer Dataset 2

Variable under investigation: Flash Dryer Outlet Air Temperature

Process Variables	Method to Determine Delay *k*	Delay (*k*)	Number of Delayed Variables (m)
Flash Dryer Outlet Air Temperature	AMI	41	4
	Autocorrelation	405	3

Trend of the dataset under study:

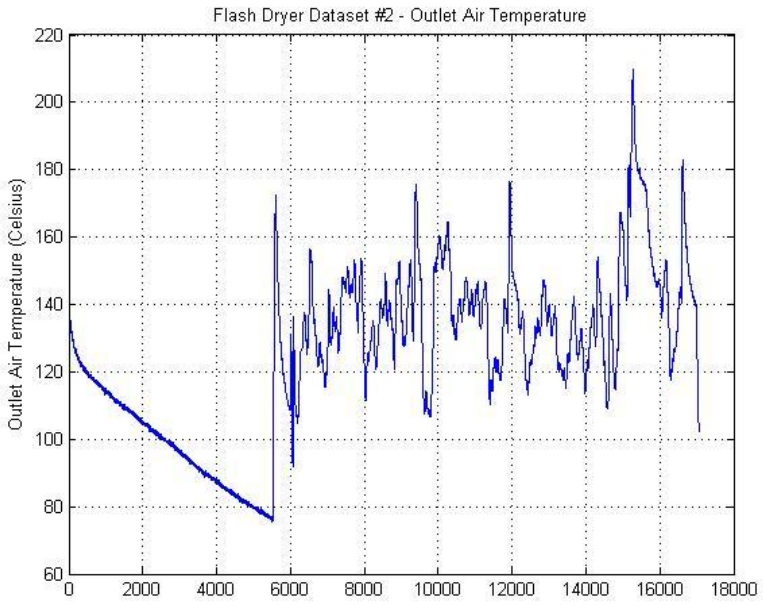


Figure 84: Flash Dryer Outlet Temperature for Dataset 2

Both sets of delay parameter pairs were analysed separately. Results are

Trend of the surrogate data comparison:

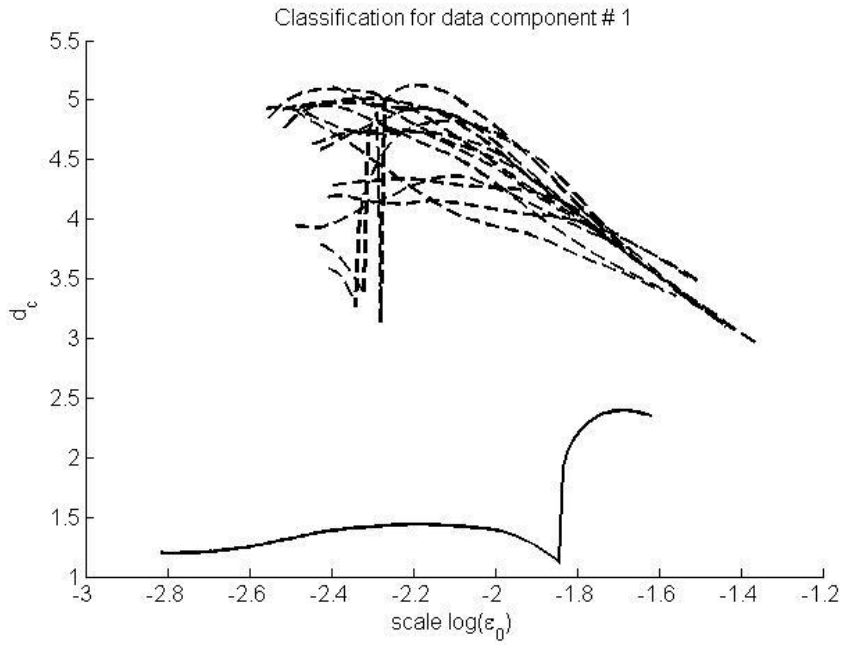


Figure 85: Surrogate Data Classification of the FD outlet air temperature for delay parameters 41x4

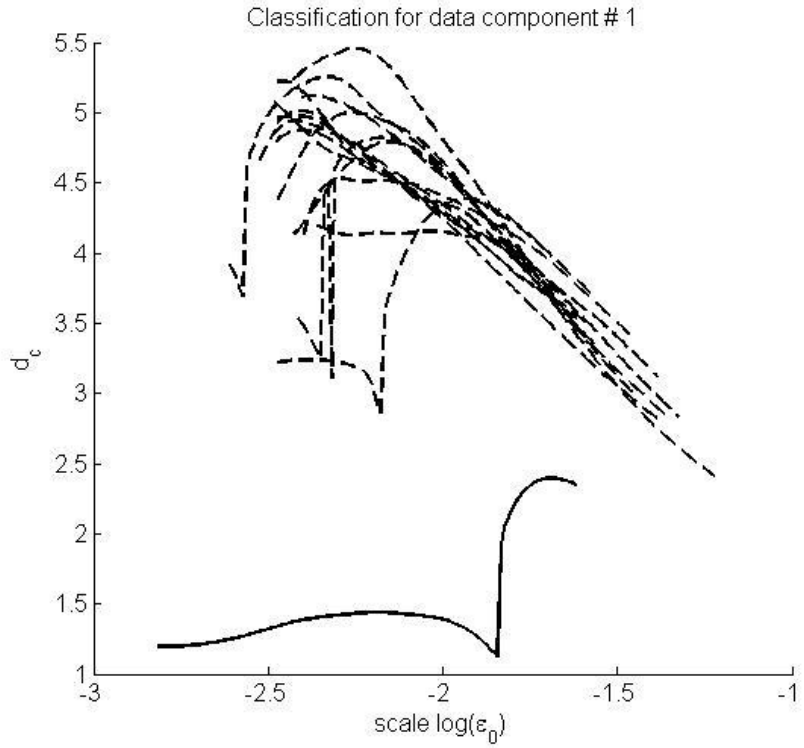


Figure 86: Surrogate Data Classification of the FD outlet air temperature for delay parameters 405x3

Both surrogate data classifications indicate very good separation from the surrogate data.

C.5. Analysis 6: Flash Dryer Dataset 2 Idle State Removed

The idle process state was removed for this analysis.

Variable under investigation: Flash Dryer Outlet Air Temperature

Process Variables	Method to Determine Delay *k*	Delay (*k*)	Number of Delayed Variables (m)
Flash Dryer Outlet Air Temperature	AMI	58	4
	Autocorrelation	365	3

Trend of the dataset under study:

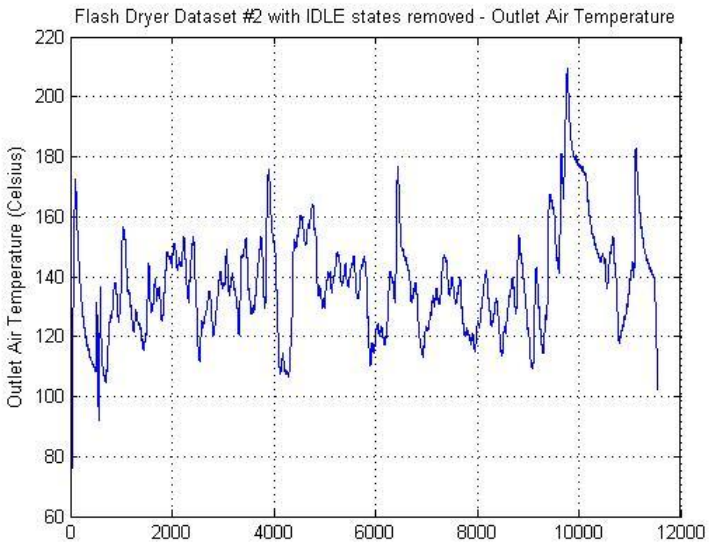


Figure 87: Flash Dryer Outlet Temperature for Dataset 2 with IDLE states removed

Both sets of delay parameter pairs were analysed separately. Results are

Trend of the surrogate data comparison:

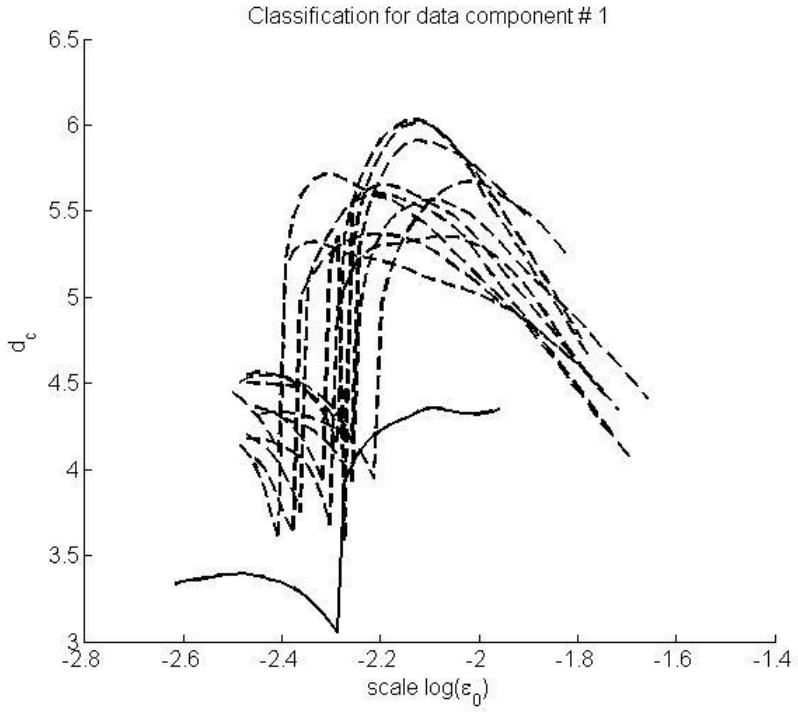


Figure 88: Surrogate Data Classification of the FD outlet air temperature with IDLE states removed for delay parameters 65x5

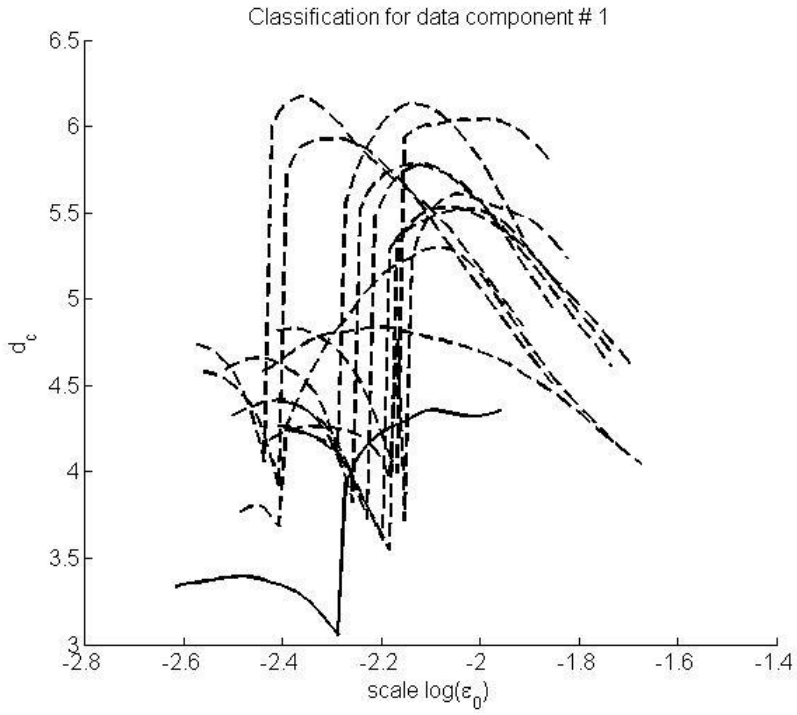


Figure 89: Surrogate Data Classification of the FD outlet air temperature with IDLE states removed for delay parameters 365x3

Both surrogate data classifications indicate that the process dynamics are not very strongly deterministic and might be difficult to model. The removal of the idle-state decreased the deterministic information in the timeseries.

C.6. Analysis 7: Flash Dryer Dataset 2 Anomaly and Idle Data Removed

The timeseries before the anomaly is investigated separately. This is a section of dataset 1 ranging from data point 1 to data point 9617. This interval is established visually.

Variable under investigation: Flash Dryer Outlet Air Temperature

Process Variables	Method to Determine Delay *k*	Delay (*k*)	Number of Delayed Variables (m)
Flash Dryer Outlet Air Temperature	AMI	42	4
	Autocorrelation	273	3

Trend of the dataset under study: (Boxed area is investigated)

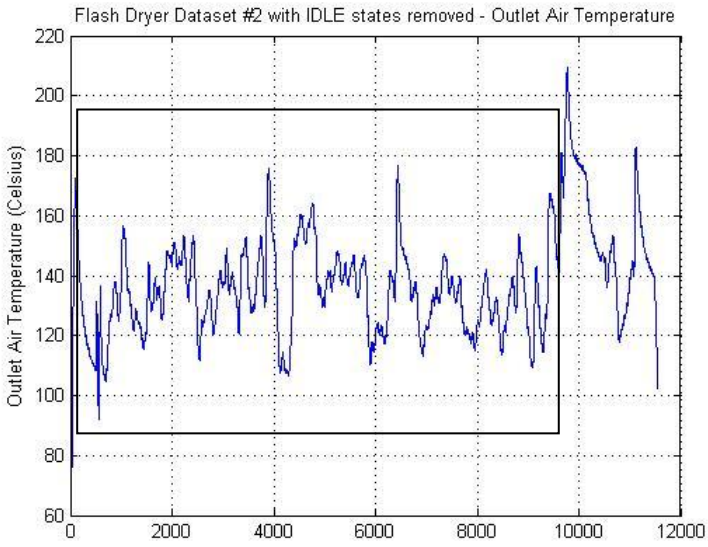


Figure 90: Flash Dryer Outlet Temperature for Dataset 2 before Anomaly with IDLE state removed

Trend of the surrogate data comparison:

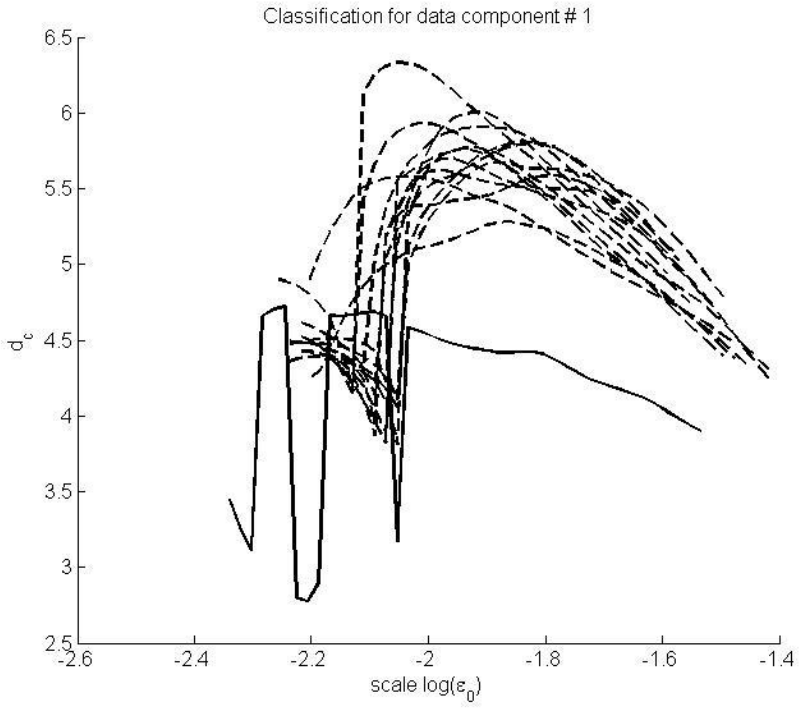


Figure 91: Surrogate Data Classification of the FD outlet air temperature for delay parameters 42x4

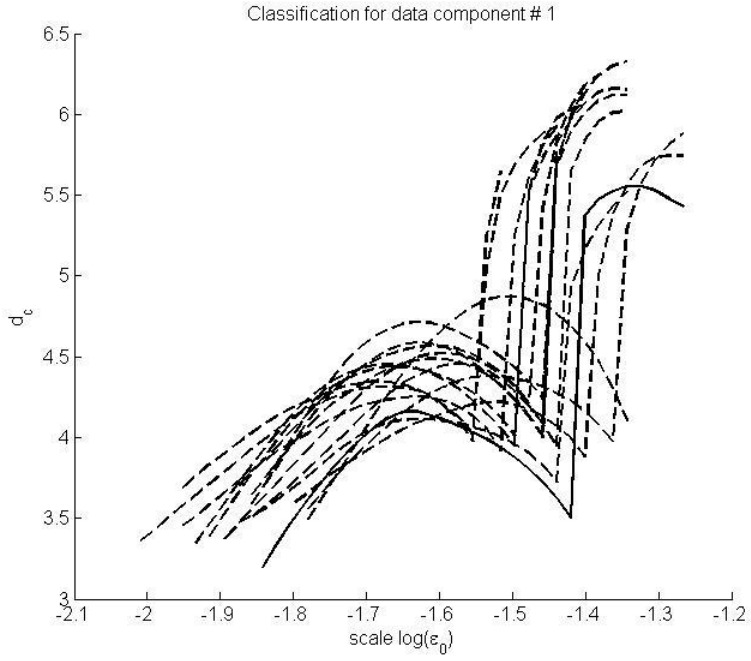


Figure 92: Surrogate Data Classification of the FD outlet air temperature for delay parameters 273x3

The surrogate data indicates poor separation from the stochastic data.

C.7. Analysis 7: Flash Dryer Dataset 3

Variable under investigation: Flash Dryer Outlet Air Temperature

Process Variables	Method to Determine Delay *k*	Delay (*k*)	Number of Delayed Variables (m)
Flash Dryer Outlet Air Temperature	AMI	65	5
	Autocorrelation	168	5

Trend of the dataset under study:

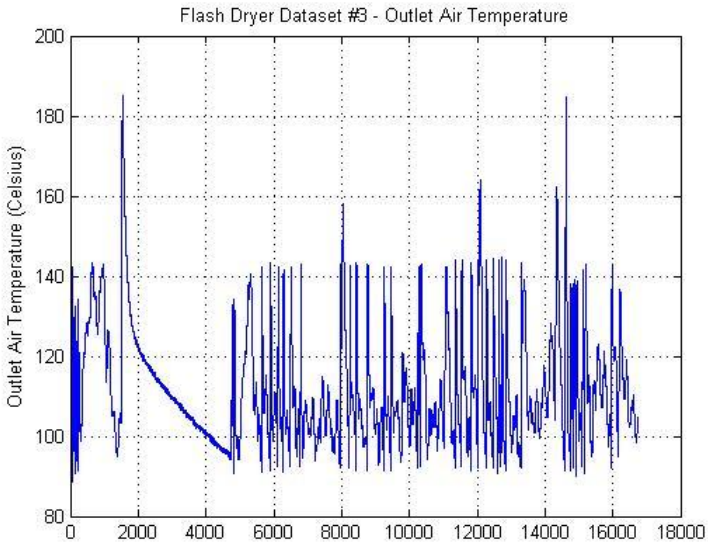


Figure 93: Flash Dryer Outlet Temperature for Dataset 3

Both sets of delay parameter dimension pairs were analysed separately. Results are

Trend of the surrogate data comparison:

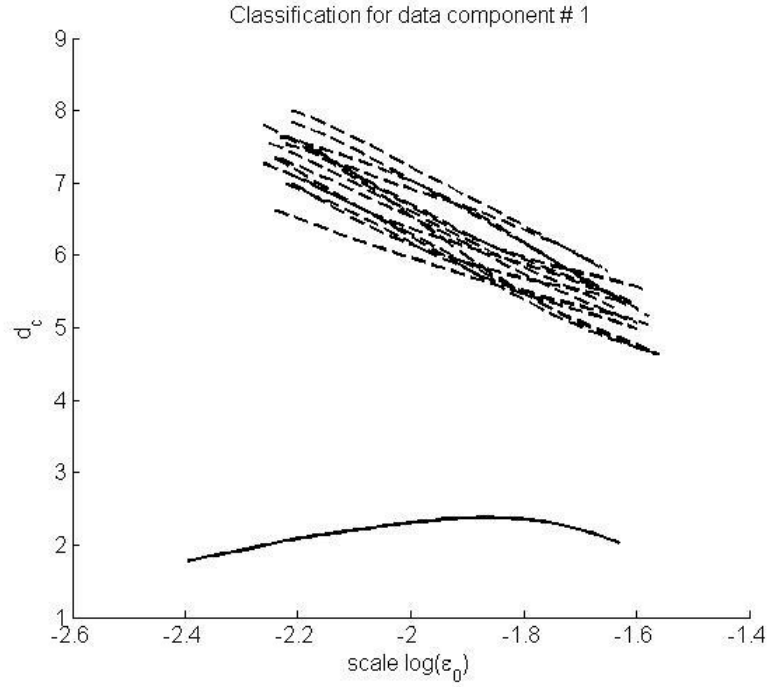


Figure 94: Surrogate Data Classification of the HGG outlet air temperature for delay parameters 65x5

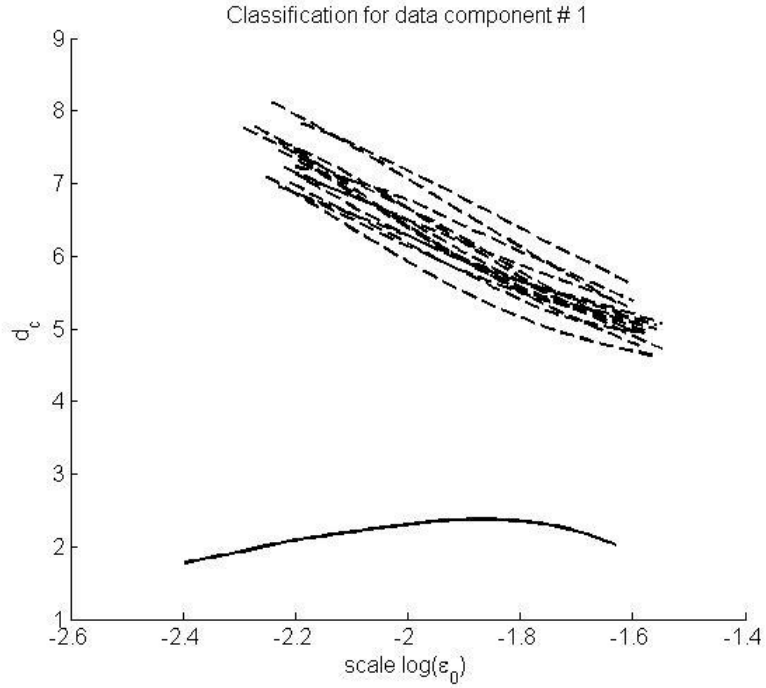


Figure 95: Surrogate Data Classification of the HGG outlet air temperature for delay parameters 168x5

Both surrogate data classifications indicate that the process dynamics are deterministic, and thus suitable for system identification.

C.8. Analysis 8: Flash Dryer Dataset 3 Idle State Removed

The idle process state was removed for this analysis.

Variable under investigation: Flash Dryer Outlet Air Temperature

Process Variables	Method to Determine Delay *k*	Delay (*k*)	Number of Delayed Variables (m)
Flash Dryer Outlet Air Temperature	AMI	65	5
	Autocorrelation	109	5

Trend of the dataset under study:

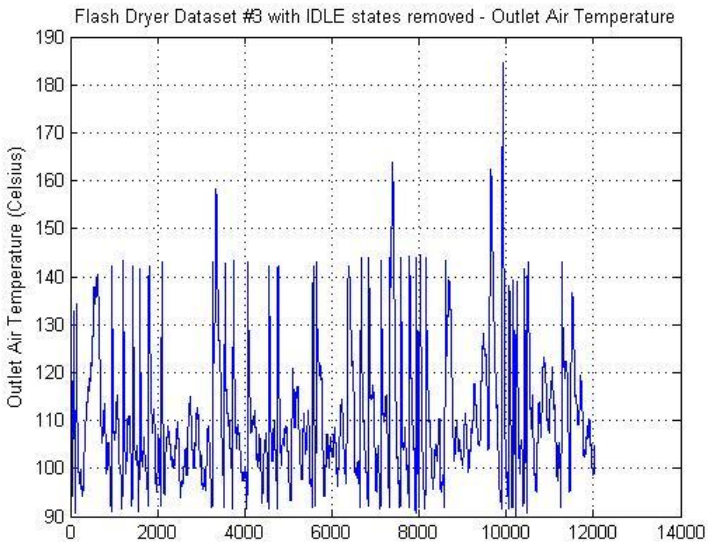


Figure 96: Flash Dryer Outlet Temperature for Dataset 3 with IDLE states removed

Both sets of delay parameter pairs were analysed separately. Results are

Trend of the surrogate data comparison:

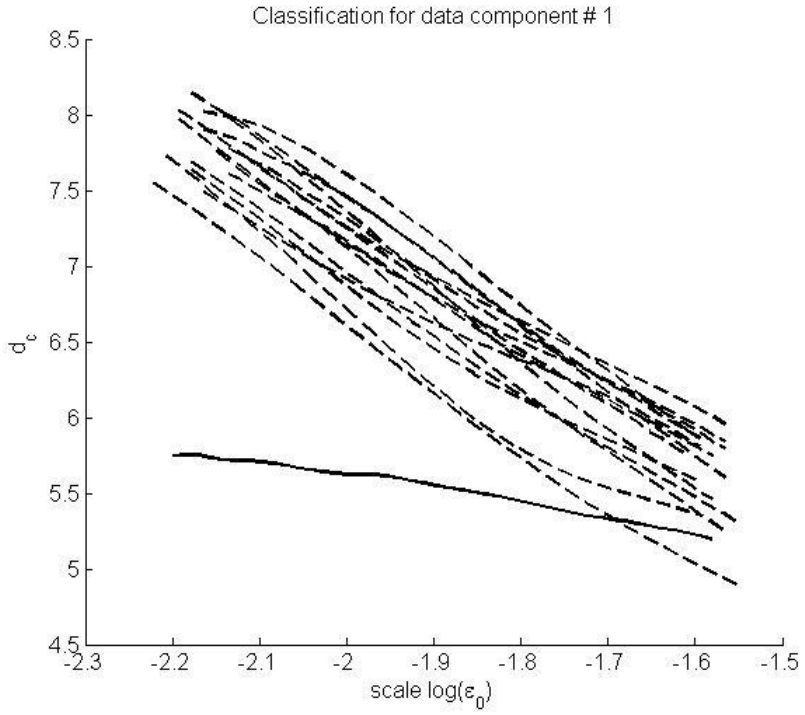


Figure 97: Surrogate Data Classification of the FD outlet air temperature with IDLE states removed for delay parameters 65x5

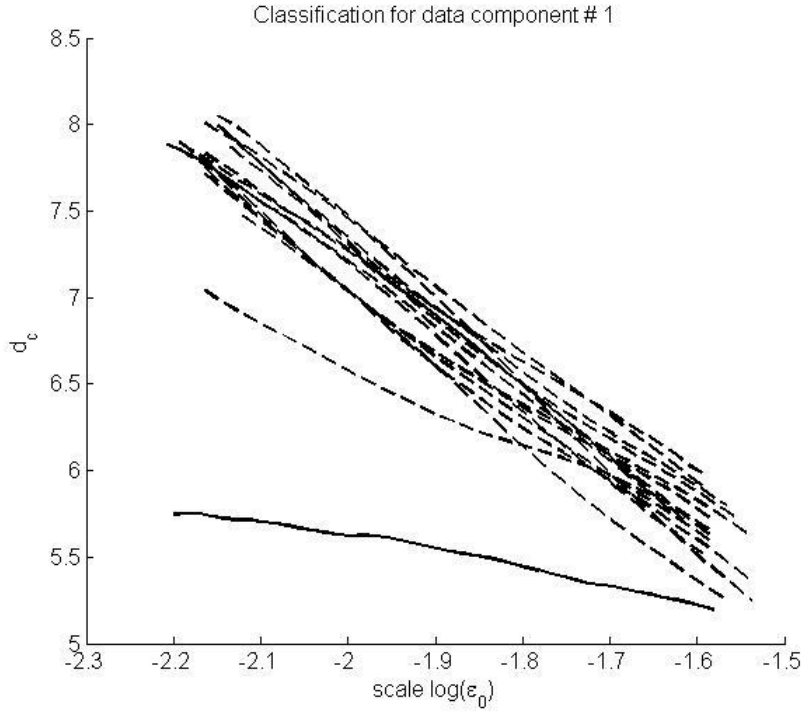


Figure 98: Surrogate Data Classification of the FD outlet air temperature with IDLE states removed for delay parameters 109x5

The surrogate data analysis indicates more stochastic behaviour than when the idle states were included. The removal of these easily modelled areas may explain the less deterministic behaviour of the surrogate data classification.

C.9. Analysis 9: Hot Gas Generator Dataset 1

Variable under investigation: Hot Gas Generator Outlet Air Temperature

Process Variables	Method to Determine Delay *k*	Delay (*k*)	Number of Delayed Variables (m)
Flash Dryer Outlet Air Temperature	AMI	65	4
	Autocorrelation	76	4

Trend of the dataset under study:

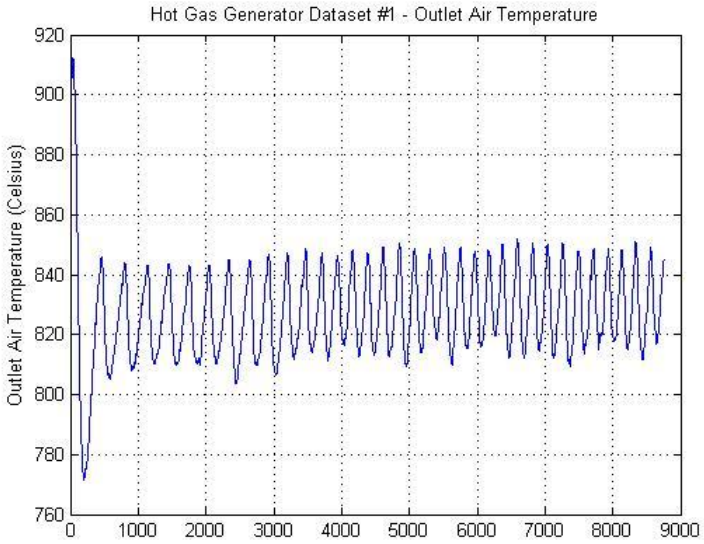


Figure 99: Hot Gas Generator Outlet Temperature for Dataset 1

Both sets of delay parameter pairs were analysed separately. Results are

Trend of the surrogate data comparison:

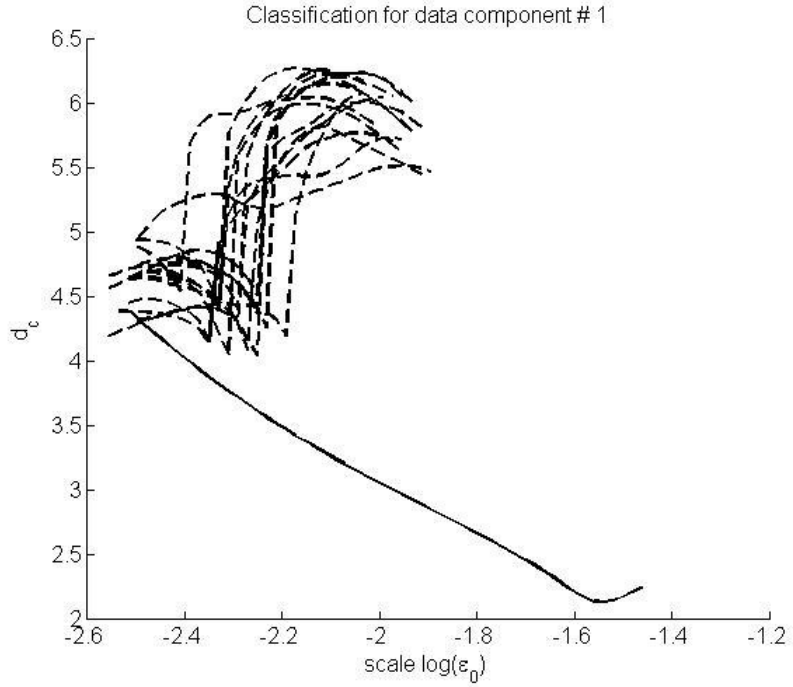


Figure 100: Surrogate Data Classification of the HGG outlet air temperature for delay parameters 65x4

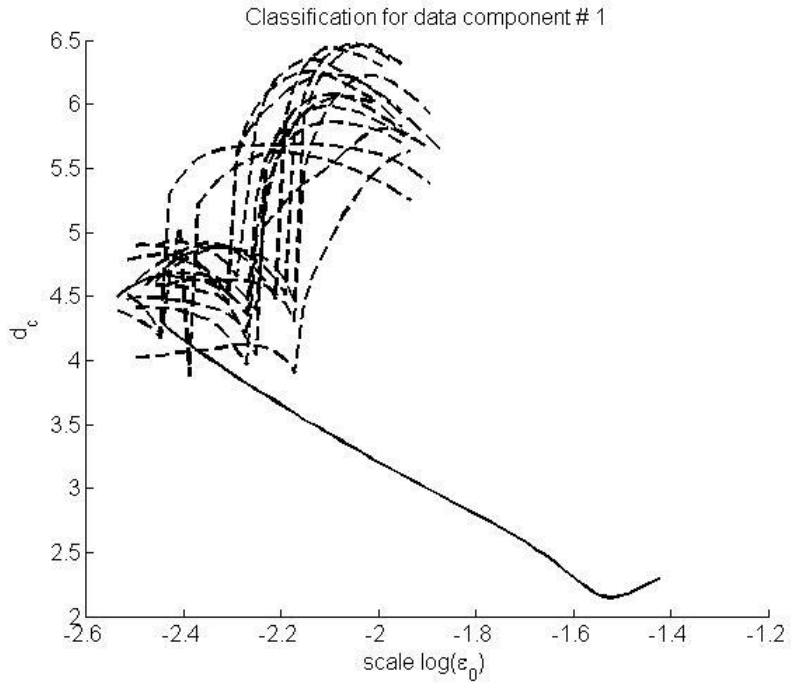


Figure 101: Surrogate Data Classification of the HGG outlet air temperature for delay parameters 76x4

The dynamic attractors indicate very strong cyclic behaviour with practically no noise in the dynamics. The surrogate data comparison indicates good separation from the stochastic data.

C.10. Conclusions

It is apparent that the removal of the idle states in the flash dryer datasets has caused the dynamics to be less deterministic. The inclusion of the idle state is however not preferred as it is not fully understood in terms of dynamics and method of data collection. It is possible that this is caused by a data collection error. There is no information to back this assumption.

The hot gas generator dataset surrogate data analysis indicates that the process is largely deterministic and thus the dynamic behaviour should be more easily extracted by the system identification parameters.

Appendix D – Delay parameters

The delay parameters, as identified in the methodology, make use of both the AMI as well as autocorrelation methods to determine delay. From this the number of delayed variables is determined. The following parameters were obtained for the flash dryer datasets 1 and 3 and hot gas generator dataset 1. Flash dryer dataset 3 is presented with and without the presence of the process idle-state. The removal of anomalies for the flash dryer datasets is not investigated.

Flash Dryer Dataset 1:

Table 43: Delay parameters for process variables for Flash Dryer Dataset 1

Process Variables	Method to Determine Delay *k*	Delay (*k*)	Number of Delayed Variables (m)
Filter Cake Feed Rate	AMI	62	8
	Autocorrelation	315	10
Hot Gas Generator Output Temperature	AMI	74	6
	Autocorrelation	260	6
Flash Dryer Outlet Air Temperature	AMI	82	5
	Autocorrelation	623	4

Flash Dryer Dataset 3:

Table 44: Delay parameters for process variables for Flash Dryer Dataset 3

Process Variables	Method to Determine Delay *k*	Delay (*k*)	Number of Delayed Variables (m)
Filter Cake Feed Rate	AMI	17	10
	Autocorrelation	16	10
Hot Gas Generator Output Temperature	AMI	80	6
	Autocorrelation	338	5
Flash Dryer Outlet Air Temperature	AMI	65	5
	Autocorrelation	168	5

Flash Dryer Dataset 3 – IDLE states removed:

Table 45: Delay parameters for process variables for Flash Dryer Dataset 3 with IDLE states removed

Process Variables	Method to Determine Delay *k*	Delay (*k*)	Number of Delayed Variables (m)
Filter Cake Feed Rate	AMI	16	10
	Autocorrelation	16	10
Hot Gas Generator Output Temperature	AMI	56	6
	Autocorrelation	363	5
Flash Dryer Outlet Air Temperature	AMI	65	5
	Autocorrelation	109	5

Hot Gas Generator Dataset 1:

Table 46: Delay parameters for process variables for Hot Gas Generator Dataset 1

Process Variables	Method to Determine Delay *k*	Delay (*k*)	Number of Delayed Variables (m)
Coal Feed	AMI	44	10
	Autocorrelation	45	4
Hot Gas Generator Output Temperature	AMI	65	10
	Autocorrelation	76	4

Appendix E – Genetic Programming

This section is an overview of system identification, genetic programming in general, the GPOls toolbox used and all additions and adjustments incorporated for data preparation or analysis of results.

E.1. System Identification Overview

The understanding of any process, system, dynamic or chemistry rests on the availability of information and knowledge. The usefulness of this information depends on the problem which needs to be solved and the format the information, or knowledge, is available in. Engineering and science, mathematics and mathematical models are embedded with large amounts of information and knowledge, thus enabling one to harness these mathematical equations and methods to further the understanding of the subject under study. (McKay et al., 1997) Mathematical models allow better supervision, fault detection, prediction, estimation of immeasurable variables, optimisation and model based control. (Coelho & Pessôa, 2009)

In control theory, the model of a process or system, essentially contains the knowledge of the relationship between inputs and outputs in a mathematical format. This allows harnessing mathematical based control, via computer, to attain the desired process outcomes. The benefit from better controlling a process is self explanatory.

The model is only a created window into the workings of the process and not the process itself. It is a fact that the process or system will continue existing or operating whether there is a model or not one at all. Nature is not truly susceptible to mathematical modelling, thus the focus should shift from obtaining model perfectly representing the whole process to obtaining a useful model.(Ljung, 1999)

The need is thus to find a best representative model for the process concerned, an accurate representation in the range of operating conditions, created under valid assumptions and in a useful format for the problem at hand. In this case the problem is finding a model useful for control be it from first principles or based on data.

System identification entails locating a representative model *G(x)* for a given input-output dataset containing the values for the chosen regressor set *x*. The basic idea is finding this unknown model $G(x_k)$ that the model's outputs are as close as possible to the expected output value *y(k)* for the given input value.

$$y(k) = G(x_k) + e(k)$$

e(k) represents the model error; *k* denotes the current time of the data point.

The model can be nonlinear or linear and may be parametric or nonparametric depending on the system identification method applied. The main questions during system identification include

- choosing the regressor set;
- finding the model structure;
- establishing model parameters; and
- deciding on the best model generated.

The theoretical background of choice of regressor set and model structure will be discussed below. The identification of parameters is left to the used toolbox. Choice of the best model is discussed during the methodology discussion.

E.1.1. Regressor Set and Model Fundamentals

In a single-input single-output (SISO) system a set of input data *u* and output data *y* have been observed over the time *t*.

$$u^t = [u(1)u(2) \ldots . u(t)]$$

$$y^t = [y(1)y(2) \ldots . y(t)]$$

System identification entails identifying the relationship between these input and output datasets. Assume the real process is represented by a transfer function in past input and output values plus all noise and disturbances:

$$y(t) = G(q)u(t) + v(t)$$

Where *u(t)* is the input at time *t*, *y(t)* is the output at time *t*, *G* is the transfer function from *u* to *y* and *v(t)* is the noise at time *t; q* is the discrete time shift operator. The time shift operator enables historic values to be incorporated to create a dynamic model. The noise is assumed as a Gaussian white noise distribution of *e(t)* as follows:

$$v(t) = H(q)e(t)$$

Where *H* is the filter transfer function used for the white noise input and *e(t)* is the white noise at time *t*. {e(t)} is assumed to be Gaussian - a sequence of independent random variables with a zero mean and a variation λ.

Ljung (1999) states that in control theory the noise may be excluded from the model to create a deterministic model. The resulting “noise-free” model will then be:

$$y(t) = G(q)u(t)$$

This noise-free solution can be obtained practically by means of a low pass filter in control with the assumption that the white noise is a high frequency disturbance. The *e(t)* term can however also be interpreted as model error or unknown disturbance. (Morari & Zafiriou, 1989) In such cases the *e(t)* term is modelled separately and introduced to the control structure separately. It is thus important to evaluate whether the error term should be included in the control structure, and consequently modelled, or if it can be assumed as negligible. The resulting process, the error assumed as significant to the control problem, can be diagrammatically portrayed as seen in Figure 102.

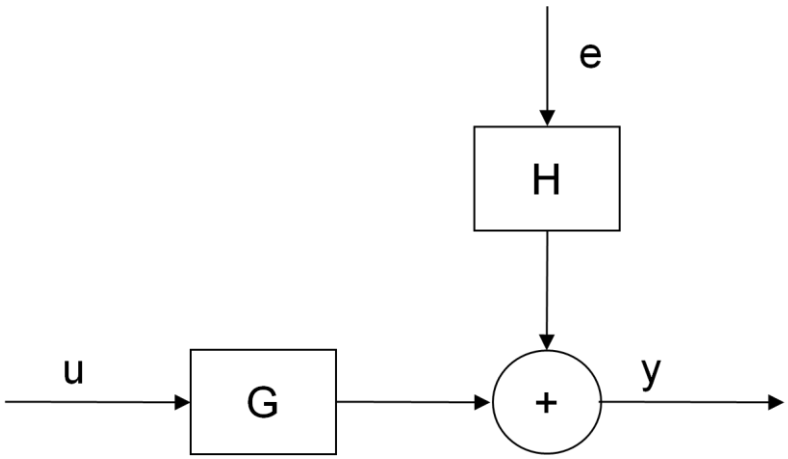


Figure 102: Input-Output system with disturbances/white noise

From process knowledge it is apparent that a current system state may be influenced by historic values. This was until now represented in the formulas in text by the discrete time shift operator *q*. The time shift operator will be brought back into the syntax when the genetic programming algorithm is discussed, but for the time being the historic values will be represented as below.

$$y(t) = G(u^{t-1}, y^{t-1}) + v^t$$

$u^{(t-1)}$ and $y^{(t-1)}$ are called the regressors of the model and are included in the model during the model structure selection process. These are defined as:

$$u^{t-1} = [u(1)u(2) \ldots . u(t-1)]$$

$$y^{t-1} = [y(1)y(2) \ldots . y(t-1)]$$

If $\varphi(t)$ is the regressor set included in the model structure, a family of models have been defined as $G(\varphi)$. The resulting model output ŷ is defined as

$$\hat{y}(t) = G(\varphi)$$

The model *G*, as written above, contains the structure but as yet no parameters. Assume the chosen model parameters are defined by θ, the model output given the parameters selected is:

$$\hat{y}(t \mid \theta) = G(\varphi(t), \theta)$$

The model *G* generated by the chosen modelling technique is thus a function of the chosen regressors from *u(t)* and *y(t)* combined with the 'best' parameters *θ*. Parameter estimation is normally either a batch or recursive optimisation process comparing the model outputs $\hat{y}$ to the expected *y*. (Ljung, 2006) The parameter estimation methods that have been used in literature include Batch Least Squares (Willis et al., 1997) , Orthogonal Least Squares (Coelho & Pessôa, 2009) and Recursive Least Squares.

E.1.2. Model structure

The choice of model structure influences both the quality and price of the model thus influencing the usefulness of the model. Quality refers to the accuracy of the model outputs normally measured as a mean square error. The price of the model amounts to the effort required in obtaining the useable model algorithm and method complexity versus the intended use of the model. It is thus clear that some form of experience and involvement is required to measure this subjective criterion.

The general model structure in system identification is given as

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

The models possible, by combining the polynomials A(q) to F(q), are described in the following table.

Table 47: Common Model Structures

Polynomials Used	Name of Model Structure
B(q)	Finite Impulse Response
A(q), B(q)	Auto-Regressive with External/Exogenous Inputs
A(q), B(q), C(q)	Auto-Regressive Moving Average with External Input
B(q), F(q)	Output Error
B(q), C(q), D(q), F(q)	Box-Jenkins

These linear model types can be expanded to nonlinear models by expanding the polynomials.

The choice of regressors included in the model structure will vary depending on the process dynamics being modelled and the resulting model structure required. There are essentially four sets of regressors which are used in the modelling techniques mentioned:

- Process Inputs *u(t-k);*
- Process Outputs *y(t-k);*
- Simulated Model Outputs $\hat{y}_s(t-k|\theta)$*; and*
- Predicted Model Outputs $\hat{y}_p(t-k|\theta)$

The different nonlinear model structures, with the regressors used in each, include (Sjöberg et al., 1995), (Ljung, 2006):

- Nonlinear Finite Impulse Response (NFIR) – *u(t-k)* as regressor
- Nonlinear Autoregressive Model with eXogenous inputs (NARX) – *u(t-k)* and *y(t-k)* as regressors
- Nonlinear Output Error (NOE) – *u(t-k)* and $\hat{y}(t-k|\theta)$ as regressors
- Nonlinear Auto-Regressive Moving Average with eXogeneous input (NARMAX) – *u(t-k)*, $\hat{y}(t-k|\theta)$ and $\varepsilon(t-k|\theta)$ as regressors
- Nonlinear Box Jenkins (NBJ) - *u(t-k)*, $\hat{y}(t-k|\theta)$, $\varepsilon(t-k|\theta)$ and $\varepsilon_u(t-k|\theta)$ as regressors

The NARX model structure is defined as:

$$y(k) = f\left(y(k-1), \dots, y\left(k-n_y\right), u(k-t_d-1), u(k-t_d-2), \dots, u(k-t_d-n_u)\right) + e(k)$$

(Coelho & Pessôa, 2009)

Where *y(k)* is the output at time *k*; t_d is the process dead time or time-delay; n_y and n_u are the output and maximum time shifts.

The reason for choosing the NARX model structure includes:

- It enables previous process outputs (interpreted as current process states) to have an influence on the current process output;

- The model structure has been explored succesfully by both Madar et al. (2005) and Coelho et al(2009) using the GP algorithm applied in this research;
- Furthermore, this structure has been used in model based predictive control algorithms, although these applications were all neural network NARX structures. (Zulkeflee & Aziz, 2009; Ramesh, 2009);
- The structure easily expands from the linear ARX structure to the nonlinear NARX model under discussion; and
- The linear-in-parameters formulation of the NARX model allows analytical parameter estimation methods to be used.

Rossiter (2003) supplies a practical view on the use of CARIMA models in Model Based Predictive Control. The use of NARX models will be similar in approach. NARX modelling has recently been used for an experimental mechanical control setup (Coelho & Pessôa, 2009). This NARX model was generated by means of a genetic programming algorithm. A Matlab® toolbox for the creation of NARX models (Madar et al., 2005) by means of genetic programming, which is the focus of this study, has been created.

E.2. Genetic Programming Information

This section of the appendix provides general GP background for the reader unfamiliar with genetic programming. Except for the second GPOls workflow, which is specific to the GPOls toolbox, the rest of this section of Appendix E - E.2 focuses on a general discussion surrounding genetic programming.

E.2.1. Genetic Programming Overview

Evolutionary algorithms (EA's) are a family of stochastically driven computation methods which aim at finding a solution to a problem by making use of the Darwinian principle of evolution commonly stated as "*survival of the fittest*". (Grosman & Lewin, 2004) A population of possible solutions, in whichever format, will randomly be compared according to certain fitness criteria and the "fittest" individual will survive and reproduce, creating a stronger population of possible solutions. The algorithm will thus strive at finding a near optimal

solution. As it is not a mathematical, step-wise solution to a problem, but rather a stochastic optimisation process (Madar et al., 2005), numerous runs of the algorithm to find a solution will not provide the same answer every time. It can be expected that the average solution to the runs will be similar in nature.

There are various methods in this family of algorithms:

- Genetic Algorithms
- Evolutionary Strategies
- Genetic Programming
- Evolutionary Programming

(Willis et al., 1997)

These methods have seen growing application in the modelling of nonlinear systems. (Hussain et al., 2000; Coelho & Pessôa, 2009). EA's have been used to build rule based models (De Falco et al., 2000), empirical models (Coelho & Pessôa, 2009; McKay, 1997), parameter estimation (Hussain et al., 2000) to name a few applications.

The main reason for using such algorithms is to simplify the modelling process. The aim is to find simpler model structures using less time and less money to create these models. It also allows modelling without expert knowledge of the first principles of the process. (Hussain et al., 2000).

Genetic programming (GP) is a popular evolutionary algorithm method first accredited to JR Koza (McKay et al., 1997) who, in 1992, made use of evolutionary principles to evolve tree structures to generate computer programmes. In GP based system identification a population of individual possible solutions is created, compared and the population evolved in the direction of increasing fitness to establish an empirical formula that best represents the input-output relationship in the datasets. This relationship is captured in the hierarchical tree structure representing an empirical formula. The evolution process is enabled by

reproduction methods, each with its own aim, increasing the fitness while still trying to search as large part of the search space as possible.

GP is chosen as the system identification procedure in this research due to its capability with regards to the following:

- Identifying the model structure from the family of possible model structures with limited prior process knowledge;
- Producing a linear-in-parameters NARX model proved to be usable in model based control;
- Adjustable goal function to decide between possible solutions to the modelling problem.

GP has been used to identify a model for control a food extrusion process (Elsey et al., 1997); closed loop identification of an experimental floating ball setup (Coelho & Pessôa, 2009); identification of a nonlinear model for model based control in mixing tank setup and a liquid-liquid extraction process (Grosman & Lewin, 2002); and steady state modelling of a vacuum distillation column and a stirring tank reactor operation. (McKay et al., 1997)

E.2.2. Workflow of Genetic Programming

Genetic programming is a recursive optimisation process starting with an initial population of possible solutions each with its own fitness. These fitness values are compared and individuals compete against each other, by means of a selection procedure, to reproduce. The offspring are expected to have similar or higher fitness values than their parents. These offspring are then added back to the population replacing the parents to form the next generation of the population. There is no assurance that a global optimum is reached, apart from thorough exploration by adjusting the mutation, crossover and direct reproduction parameters. Higher mutation rates assists the GP to not get stuck in a local optimum.

The following diagram sets out the generalised workflow to explain the concept followed by evolutionary algorithms. (McKay et al., 1997) (Coelho & Pessôa, 2009)

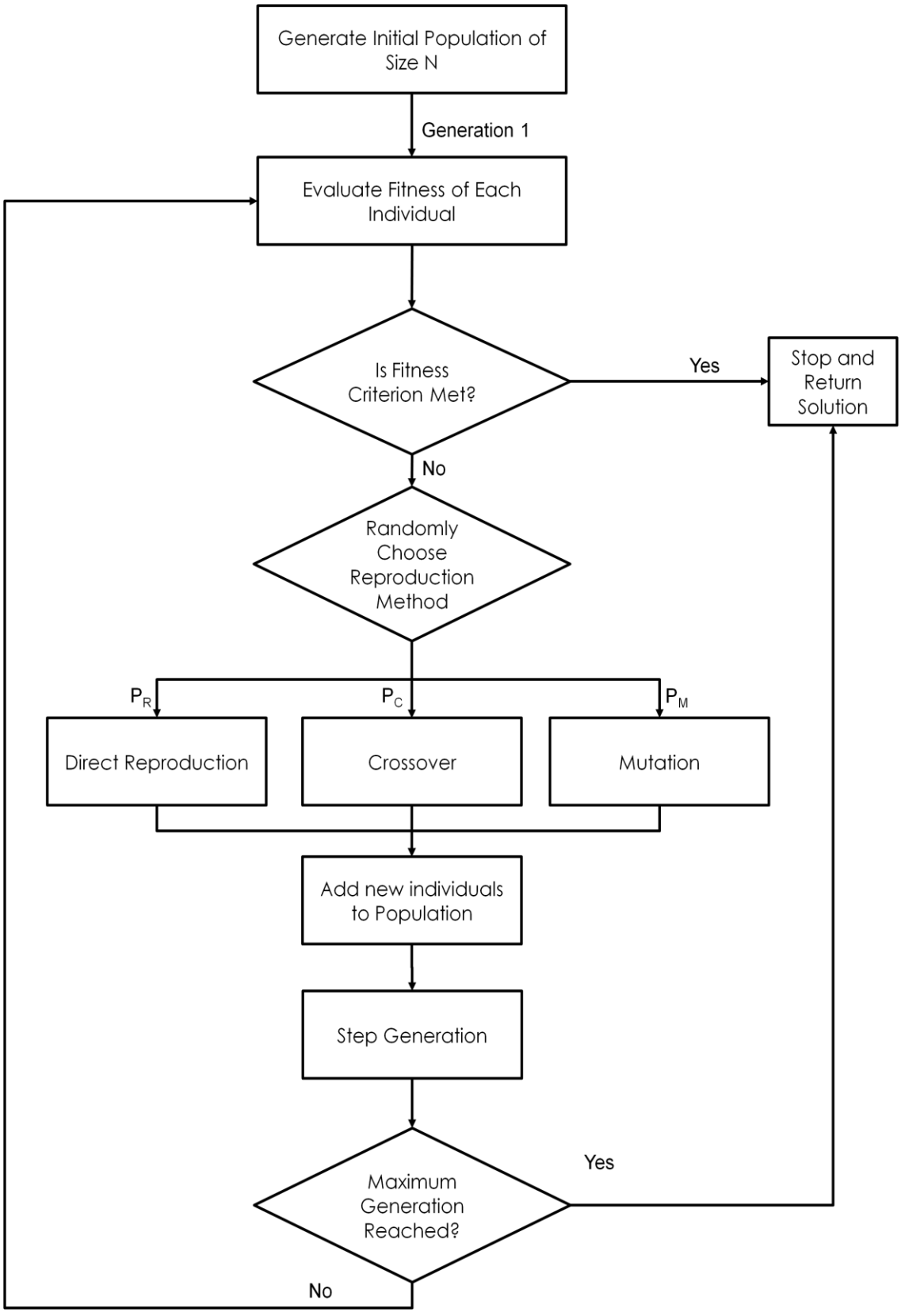


Figure 103: Genetic Programming Algorithm Workflow

E.2.3. Initial Population

The initial population of individual solutions can be generated in various ways. Three of the most common methods are:

- Random generation;
- Pre-Selected Function; and
- Previous solution functions

A GP is normally set up to execute a definite number of runs. In the last population generating method above, the solutions from previous runs are included in the population.

It should be noted that the initial population can be built up from all three these methods to increase the model search space. Sjöberg et al. (1995) states that prior knowledge of the process and physical insight should be incorporated into limiting the search space for the model structure. This makes sense when taking into account that a finite sample of process data will not contain all the knowledge about the mechanics and internal workings of the process and some guidance can be provided. (McKay et al., 1997) it is however clear that limiting the structure of the programmes prematurely eliminates various possible model structures from the model space, so this should be done wilfully and cautiously.

In this experiment, the choice exists to load an existing, or predefined population as generation 1, instead of randomly generating a population. This predefined population is either the result of a previous experiment or the user's manually defined population. Defining such a population is not discussed, unless it makes out a very important part of a specific experiment. In such a case the source of the predefined population will be discussed.

E.2.4. Fitness Functions

This section discusses the various fitness functions and the decision surrounding the parameters of the fitness function used in this study.

The fitness of an individual in a GP is a numeric value and a function of the desired model outcome, which is focussed on accuracy. Accuracy, or fitness, is normally measured by making use of error based criteria such as least square errors (LSE), with lowest error indicating the fittest individual (Greeff & Aldrich, 1998). In some cases correlation coefficients of the target output value compared to the model predicted value are used as model fitness value. (McKay et al., 1997) Whichever method is chosen, or combination of methods constructed to represent fitness, the fitness value is used by the algorithm during the search for a solution. Any other statistics calculated, not contributing to the fitness function, do not attribute to the search for a solution, unless incorporated elsewhere into the algorithm. Such statistics can be used for choosing the optimal model in post analysis work.

The fitness may be altered depending on the required outcome of the model. In MIMO modelling the fitness function is built up by means of weighted values per output variable. Another MIMO fitness method makes use of the Pareto fitness. (Hinchliffe & Willis, 2003) The fitness function can also be harnessed to penalise an individual for complexity, over fitting and severity (Grosman & Lewin, 2002; McKay, 1997; Madar et al., 2005). Madar et al. (2005) tried the same penalty function, but also presented a parsimonious tree pruning algorithm incorporated in the parameter estimation step. Willis et al. (1997) combines the use of root mean square (RMS) fitness with the correlation coefficient by weighting the importance of each criterion. Discipulus ® software makes use of minimising the average of the square of raw errors over the dataset (Francone, 2001). Winkler et al. (2004) and Greef et al. (1998) propose the same approach can be used for nonlinear model structures. The point at hand being that the fitness function is very flexible depending on the software being used.

The following fitness functions discussed above were proposed:

- R^2 values are used by Coelho & Pessôa (2009).
- Greeff (1998) – Sum squared error function (note that the error is minimised). This is the base method of measuring fitness in literature.(Winkler et al., 2004)

$$SSE = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

- Francone (2001) makes use of a standard mean square error with a separated algorithm for enforcing parsimony to ensure simpler solutions prevail.

$$MSE = \frac{1}{N} \sum_{i=1}^{N} |\hat{y}_i - y_i|^2$$

- Grosman (2004) and Grosman (2002) – Using the standard deviation of the data (*SSY*) as well as the standard deviation of the sum square errors between predicted and expected output values (*SSE*); combined with a penalty for the tree complexity:

$$Fitness = \frac{\alpha F_M + (1 - \alpha) F_P}{1 + \exp(\gamma_1 (n_b - (\gamma_2 + \beta)))}$$

$$F_M, F_P = \frac{SSY}{SSY + SSE}$$

α applies the importance of the validation set in the fitness formula, implying the measure to which over fitting should be taken into account; γ_1 is the severity to which a complex tree structure should be penalised and γ_2 is the level of branches of the best solution thus far; β increases the search space. This fitness formula is a trade off between model complexity and fitness.

- McKay (1997) and Madar et al. (2005) – Combination of correlation of output values, instead of minimum error methods, combined with a penalty for the complexity of the solutions.

$$Fitness = \frac{r(i)}{1 + \exp(a_1 (S_L - a_2))}$$

a_1 and a_2 are parameters adjusting the softness with which the complexity rule is enforced; S_L is the size of the tree under investigation expressed as string length; *r(i)* is the correlation coefficient measuring the variation between the predicted and expected values.

$$r(i) = \left| \frac{\left(\sum \frac{y_P(i,j) y_T}{R}\right) - \left(\sum \frac{y_P(i,j)}{R}\right)\left(\sum \frac{y_T}{R}\right)}{\sigma_P(i)\sigma_T} \right|$$

i denotes the individual tree under inspection; *j* denotes a specific data record; *R* is the number of data records; σ_P and σ_T is the standard deviation of the predicted and target values; $y_P(i,j)$ refers to the predicted value for record *j* by tree *i*; y_T is the expected target value.

Madar et al. included a orthogonal least squares tree pruning algorithm to enforce parsimony.

- Willis et al. (1997) – A multiple objective fitness function built up by the linear sum combination of a RMS error and correlation measure. The correlation is worked in to incorporate non-correlated function generation.:

$$Fitness = \alpha f + (1-\alpha)(1-\gamma)$$

$$f = \frac{E\{y\}}{E\{y\} + RMS}$$

α is a weighting parameter setting the importance of the RMS and correlation fitness measurements; γ is the average of the standard correlation coefficient between functional groups; *E{.}* is the expectation operator.

The following table compares the fitness functions presented:

Table 48: Comparison of fitness functions from literature. The sections indicate what the focus of each function is.

Source	Prediction Error	Correlation	Tree complexity
Francone (2001) and Greeff (1998)	✓		
Grosman (2004) and Grosman (2002)	✓		✓
McKay (1997)		✓	✓
Willis et al. (1997)	✓	✓	
Madar et al. (2005)	✓	✓	✓

E.3. GPOls Toolbox

The GPOls toolbox is built up of 16 *.m files. 7 of these are main function files and the other 9 are called on to manipulate the tree structures. Within some of the *.m files smaller sub function are written for the main purpose of cleaner code. These 16 Matlab® files, together with the main GP user file, will be discussed.

The general structure in calling the function, explanation of inputs and outputs, a brief description of its workings and where it fits in the larger GP algorithm will be included. The user file is discussed first and the GPOLS toolbox files are handled thereafter in alphabetical order.

Note that this discussion does not include the latent variable reconstruction used to construct the terminal set. The latent variable reconstruction is discussed separately from the normal GPOls algorithm.

E.3.1. GP Experiment Logic

The discussion of the logic needed to manage and run the GP will be introduced and presented in increases levels of complexity and involvement:

i. General pseudo code for using the GPOls toolbox;

ii. Pseudo code for the drying circuit GP experiment; and

General pseudo code for using the GPOls toolbox:

Load/define input (X) and output (Y) variables
Adjust Dataset to expected time shifts and time delays
*Define the functional** and terminal sets.*
Set the population size
Set the maximum tree size
Initialise the population using gpols_init function
Set the GP Parameters/Options
Evaluate the population using the gpols_evaluate function
Start the GP loop for a defined number of generations – termination criteria being the number of generations.
    Evaluate and manipulate the population using the gpols_mainloop function
    Display the result per loop iteration
End the loop
Display the best result based on fitness

**The Functional Set MUST be defined with a '+' first and a '*' second in the functional set array. This is due to the hardcoded search for terms in the polynomial generating and tree pruning functions.

The pseudo code for using the drying circuit experiment:

Set up the Matlab® Environment for the experiment by closing all current trends and clearing the workspace
Load the GP Experiment Parameters, including the experiment name, process to model and dataset to use
Load the GPOls parameters, including the terminal set and number of repetitions of the experiment required
Set up the experiment directory and experiment results report
Load the prepared and normalised dataset into the workspace from a specified directory
Load the process information for the process being modelled, including the delay parameters
Set up the validation dataset according to the validation type specified
Normalise and construct latent variable set for the validation dataset
Construct latent variable set for the training dataset and define the functional set
Save the latent variable construction of the training dataset in csv format
Start the loop of the predefined number of experiments (n=1 to number of experiments)
    Initialise population or load an existing population
    Load the GPOls parameters
    Evaluate the first generation of the population
    Display the first generations best solution
    Populate the fitness and MSE values for the first generation for evolution tracking
    Start the loop of Generations with the number of generations as the termination rule

        Evaluate and manipulate the population using the gpols_mainloop function
        Display the result per loop iteration
        Populate the fitness and MSE values for the first generation for evolution tracking
        End the generation Loop

    Plot the landscape of the evolution of the population for the MSE and fitness values
    Save the landscape to the experiment directory
    Display the best result of the experiment
    Extract an executable version of the result from the population by using the function gpols_best_results.
    Calculate the model output by means of the executable model
    Trend the training data versus the model output
    Calculate and display the residuals of the training set
    Calculate the MSE, fitness, model output and residuals of the validation set
    Display the model output and residuals of the validation set
    Populate the Experiment report
    Save the experiment result and report in the experiment directory
    Repeat the Experiment Loop until the number of experiments are reached

Display the experiment report to compare all the experiments
End the GP run

E.3.2. Drying Experiment Parameters

The GP logic discussed has various parameters of its own. These parameters are specific to the drying circuit GP run and indicate variations in the experiment, such as dataset to use

and process to model. The following parameters, with the parameter variable name in italic brackets, are worth mentioning:

Training Dataset Contains Process Idle States ("*Idle*"): This indicates if the dataset used for training contains process idle states. *Idle*=1 means the dataset does contain idle states and *Idle*=0 means there are no idle states present. The need to distinguish is due to distinguish between the dataset to import and the delay parameters to load.

Identification Method for Delay Parameters Preferred ("*AMI*"): Delay parameters were calculated using both the AMI or Autocorrelation methods. *AMI*=1 indicates that the delay parameters generated by AMI should be used. *AMI*=0 indicates the use of delay parameters generated by means of Autocorrelation. Both methods use the false nearest neighbour method to identify the number of latent variables.

Experiment Name ("*ExpName*"): The experiment name is used when saving the results. Each repetition of the experiment is saved in the same directory under the specified name with the number of the specific repetition concatenated to the name. This parameter is a string.

Process to Control ("*Controller*"): This could be either the flash dryer process or the hot gas generator process. This parameter is a number "1" or" 2", where "1" refers to the hot gas generator and "2" refers to the flash dryer.

Dataset to load ("*datasetNumber*"): The number of the dataset to be used for training. The datasets are stored with numbers in a Matlab® structure. Retrieving the correct data requires pointing to the correct dataset number. This parameter is determined by studying the datasets to determine the best option.

Prediction Step Size ("*nstep*"): The step size of prediction steps required for the predictive model is set. By default this is set to one, unless specifically chosen otherwise.

Validation Type ("*type_val*"): Specifies if a second dataset will be used for validation (1) or a percentage of the training dataset be moved to the validation set (2). This parameter is either "1", or "2".

Validation Dataset or Percentage ("valdatasetNumber" and "*Val_percentage*"): The use of these parameters depends on the choice of the validation type.

The validation dataset number is used in the same way as the training dataset number discussed above.

The validation percentage indicates the percentage of the training dataset which should be used for validation rather than training. This is a fraction between 0 and 1.

Number of Experiment Repetitions ("*NumRuns*"): The number of repetitions of independent system identification experiments required. The result of every loop is saved separately and the collective results are included in the experiment report.

Generating a new or using a predetermined population ("*New_Pop*"): This indicates if the GP run should randomly generate a new population from the functional and terminal set (*New_Pop*=1). The alternative is to load an old, previously generated population making use of population already evolved in a specific direction (*New_Pop*=0). See Appendix E - E.2.3 Initial Population.

E.3.3. GPOIs Parameters

The GPOIs parameters are displayed in section 5.2.3 in Table 8. These parameters are repeated and discussed here.

Generation Gap refers to the size of the previous population which should be replaced in the new population. This is a fraction between 0 and 1 where 0 places the old population as-is into the new population, and 1 replaces all of the old population. The remainder of the population either undergoes direct reproduction, crossover or mutation. Replacing all of the population will result in a very dynamic evolution of the population, but with much less consistency.

Crossover probability controls the chance of an individual being submitted to crossover. It is a fraction between 0 and 1, where 0 will result in less crossover and 1 will result in all the individuals being submitted to crossover.

Probability of mutation controls the chance of an individual being submitted to mutation. It is a fraction between 0 and 1, where 0 will result in less mutation and 1 will result in all the individuals being submitted to mutation.

Note: The crossover probability plus mutation probability must be equal or less than 1. The remainder less than 1 is calculated and represents the direct reproduction probability. The probability for direct reproduction is not explicitly chosen, but calculated as *1-(probability for crossover + probability for mutation)*

Selection Type refers to the method by which individuals are chosen for reproduction. Reproduction refers to either direct reproduction, crossover or mutation. There are three methods of selection, random selection, roulette and tournament selection. 0 indicates random selection; 1 indicates roulette selection; and 2 or greater than 2 refer to tournament selection. In the case of tournament selection the number greater than or equal to 2 indicates the number of individuals involved in the tournament.

One- or two-point crossover indicates the type of crossover. 1 indicates one-point – 2, two-point cross over.

Tree size penalty weighting is a fraction between 0 and 1 indicating the severity of the tree size penalty. A lower fraction results in a softer penalty. See E.2.4 Fitness Functions for a detailed discussion on choosing this parameter.

Tree size penalty location is a number indicating the maximum number of nodes which may be present in an individual tree before the penalty starts taking effect. The more nodes the individual tree contains above the tree size penalty location number, the harsher the penalty will be.

OLS threshold is a value between0 and 1 indicating the minimum energy contribution of a branch before it should be pruned. A higher fraction will result in more branches being pruned and a lower fraction (e.g. 0.05) will allow branches with a small contribution to be preserved.

Polynomial evaluation is either switched “on” (1) or “off” (0). If switched on, all individuals are interpreted as polynomials, in which case all plus signs (‘+’) are replaced by multiplication signs (‘*’). For this researched this was always chosen as “off”.

Evaluation of individuals can either occur by evaluating all individuals (EvalInd = 1) during every generation loop, or only evaluating all new individuals (EvalInd = 0) at the end of each generation loop. This is only to stop unnecessary computation and makes the algorithm faster.

The following 4 parameters are also GP parameters, although they are handled separately from the other parameters by the GPOls algorithm.

Terminal set is discussed in E.3.4.

Population size defines the number of individuals in the population. This number stays fixed throughout the GP run.

Maximum tree depth indicates the maximum number of nodes which an individual may consist of. The value defined is the power of 2, i.e.

$$maximum\ number\ of\ nodes = 2^{maxtreedept\ \ h}$$

The individual lengths are determined by the node type and may be shorter in length. The tree stops when the leave ends of the tree are all of the functional set and no more mathematical operators are available to expand the tree.

Number of Generations: The length of the GP run is defined as the number of generations which should be completely evolved before the run terminates. The number of generations is the only termination criterion in the GPOls toolbox.

These parameters are all set at the beginning of the experiment logic.

E.3.4. GP Functional and Terminal Sets

The functional set contains the mathematical operators which can be used in the tree nodes of the individuals. Tree nodes containing these operators are also referred to as functional

nodes. In the GPOls algorithm this set is defined in the main experiment logic by the user. The operators included varied between experiments, but could be any of the following:

- Addition (+);
- Subtraction (-);
- Division (/);
- Multiplication (*);
- Negative Division (./-);
- Negative Multiplication (.*-); and
- Square Root (√)

Note that the square root function is tested with both the absolute value or without, when data is biased.

The terminal set is the list of process variables which would make up the terminal nodes of the individual trees, or the variables in the equation. The terminal set is the result of interpreting the process knowledge and preparing the data for system identification. In the GPOls toolbox this set is the columns of *X*, where *X* denotes the matrix of the constructed latent variable set - each column representing a variable, or lagged version of a variable. The size of the terminal set corresponds to the number delayed versions of each variable included in the latent variable contruction.

This terminal set is built up during generation of the dataset by the Matlab function built for this experiment specifically. This is discussed in more detail in the discussion of the function *gpols_gendataset*.

E.3.5. Orthogonal Least Squares Theory

Given the linear-in-parameters model

$$y(k) = \hat{y}(k) + e(k) = \sum_{i=1}^{M;N} \theta_i F_i(k) + e(k)$$

$$F = \begin{bmatrix} F_1[x_1] & \dots & F_M[x_1] \\ \dots & \dots & \dots \\ F_1[x_{1N}] & \dots & F_M[x_N] \end{bmatrix}$$

Where y is the expected output vector; ŷ is the predicted ouput; e is the white noise; F is the regressor matrix; θ is the parameter vector, or model weights, for the terms identified in the linear-in-parameters model for M regressors and N data points. This regression model can be rewritten in matrix format as

$$\boldsymbol{y = F\theta + e}$$

The orthogonal decomposition of the regression matrix F is

$$\boldsymbol{F = WA}$$

Where A is a MxM upper triangular matrix and W is a NxM matrix with orthogonal columns such that

$$W^T W = D$$

Where D is a diagonal matrix.

The output variance is explained by

$$y^T y = \sum_{i=1}^{M} g_i^2 w_i^T \, w_i + e^T e$$

Where g is auxiliary parameter vector given by

$$g = D^{-1} W^T y$$

The err term indicating the relative variance contribution of each term, Fi, to the output value is given by

$$[err]_i = \frac{g_i^2 w_i^T w_i}{y^T y}$$

(Chen, 2006)(Madar et al., 2005)

The OLS algorithm transforms the regression matrix F into a set of orthogonal basis vectors. This allows the influence of each term Fi to be investigated. Thus it investigates the influence of each term on the model output.

E.3.6. Adjusting OLS threshold

For the GPOls toolbox an adjusting OLS threshold was implemented. This was implemented where the evolution could not start. The OLS threshold begins low too include branches with small contributions, thus possibly over fitting the model. However, as fitness increased the OLS threshold is increased according to the size of the jump in the fitness value.

An upper and lower OLS threshold is defined in the experiment. The lower bound is used as initial OLS threshold. The OLs threshold is then increased according to the following implemented formula:

$$Active\ OLS\ Threshold = \Delta fitness\ of\ fittest\ individual * (Upper\ OLS\ bound - active\ OLS\ threshold)$$

If no changing OLS threshold is required, the upper and lower limits are chosen to be equal during experimentation setup.

E.4. GPOls Additions

The extra functions included in the toolbox include the following functions, including a brief discussion of how the function works and the detailed code. The table in the original text is repeated here. Each of these added functions are discussed after the table.

Table 49: Functionality added to the GPOls toolbox and the names of the functions created.

Functionality	Matlab® Function	System Identification Function
Obtain the executable formula string for a specific individual for the population index *ix*.	*gpols_any_result.m*	*Analyse GP Output*

Functionality	Matlab® Function	System Identification Function
Display the fitness, MSE and formula for a specific individual for the population index *ix*.	*gpols_any_result.m*	*Analyse GP Output*
Calculates the validation statistics of the best individual in a given population according for any dataset. Also constructs the graphs for model output and residual analysis.	*gpols_alternative_val.m*	*Analyse GP Output*
Display the fitness, MSE and symbolic formula for the *n* best results	*gpols_best_results.m*	*Analyse GP Output*
Obtain the symbolic and executable formulas, MSE, fitness and population index of the *n* best individuals	*gpols_best_result.m*	*Analyse GP Output*
In some cases predefined populations were used as a starting point for a GP run. In such cases it is necessary to evaluate if the population coincides with the chosen terminal and functional sets. This function tests the population and provides a solution if it is incorrect.	*gpols_testpopulation.m*	*Data Preparation*
Track the evolution of the population through all the generations. A landscape of the fitness and MSE values are plotted for the whole population for each GP run.	*gpols_trackevo.m*	*Analyse GP Output*
Calculate the validation MSE, fitness and residuals for the chosen individual. These results are plotted to compare to the training dataset.	*gpols_validate.m*	*Analyse GP Output*
Select the delay parameters relevant to the chosen training dataset.	*gpols_embedparameters.m*	*Data Preparation*
Create the latent variable reconstruction, with corresponding symbols for an *n*-step prediction model.	*gpols_gendataset.m*	*Data Preparation*

E.4.1. GPOLS_ANY_RESULT

This function is used to find the details for any individual in the population after the GP experiment is complete. Once a desired saved population is loaded into the workspace, the executable formula, fitness and MSE of an individual can be obtained and displayed. Only the executable formula is returned as an output. The latent varible regressor set, population and specific location, or index, of the individual is required. The "*display*" variable indicates if the results should be displayed in Matlab.

This function assists with post-experiment evaluation of individuals. It is developed solely to assist the user.

This function may be used with any experiment using the current version of the GPOls toolbox.

E.4.2. GPOLS_ALTERNATIVE_VAL

This function uses a given saved experiment output and calculates the MSE and fitness of the model according to any given dataset. The dataset is signified by the dataset number as established during data preparation. This function requires insight into the dataset used for training as well as if AMI or autocorrelation settings were used to calculate the delay parameters. The latent variable set is constructed and the residuals calculated according to the executable model string.

E.4.3. GPOLS_BEST_RESULTS

This function identifies the top "*x*" individuals in terms of fitness and returns their executable function, interpretable function, MSE and fitness values as well as their position in the population as outputs. An option is included to display the results in Matlab.

This function is used during the experiment in the validation step. In this case only the top individual is identified. The user may also use choose to analyse more than one individual, in which case this function will assist.

This function may be used with any experiment using the current version of the GPOls toolbox.

E.4.4. GPOLS_EMBEDPARAMETERS

The delay parameters for each dataset which may be used in the drying experiment, are stored in and obtained from this function. These parameters are used in the *gpols_gen dataset* function. It serves as a database for the experiment, and assists in making the experiment parameters less complex to adjust. The delay parameters are chosen and returned when the dataset number, process being modelled, process idle state inclusion and preferred method of obtaining the process orders are provided.

This function is designed specifically for the drying circuit experiment and should be adjusted if it is to be used in future applications of the GPOls toolbox

E.4.5. GPOLS_GEN_DATASET

This function makes use of the delay parameters obtained by the user and constructs the latent variable set for an *n*-step prediction model. It also constructs the terminal set for the GP algorithm. The terminal set is used to display an interpretable version of the solutions. See E.3.5. The results from this function entail:

- The latent variable regressor set;
- An n-step time shifted version of the output regressor; and
- The collection of names of the terminal set.

The inputs required by this function are the

- input timeseries, with a variable per column;
- output timeseries, single variable;
- process delays per variable in a column array;
- number of delayed versions per variable in a column array;
- number of time steps to shift the output variable (n-step prediction);
- input variable names with a name per cell (optional); and
- output variable name in a cell (optional).

If the variable names are omitted the function only constructs and returns the latent variable reconstruction.

This function may be used with any experiment using the current version of the GPOls toolbox. The delay parameters should be defined elsewhere.

E.4.6. GPOLS_TESTPOPULATION

As discussed in Appendix E - E.2.3 Initial Population, it is possible to load a predefined population as the first generation of a GP run. This population should relate to the terminal and functional sets. Furthermore, the population size should be tested to see if it corresponds to the GP run parameter ‘popusize’. This function does all three tests and adjusts, or stops, the experiment accordingly.

If the functional or terminal sets differ in any way, the function will prompt the user and end the experiment. The experiment needs to be adjusted before it can commence.

If the population size of the predefined population differs from the population size defined for the current GP run, then the population size of the predefined population is adopted. The user is prompted that the population size for the experiment was adjusted.

E.4.7. GPOLS_TRACKEVO

This function creates a three dimensional plot of the landscape of the evolution. This assists in identifying the search getting stuck in a local optimum; visually investigating the evolution of individuals and comparing various results from different GPOls parameters.

The population, the number of generations, current generation counter and historic fitness and MSE values are required as inputs. An array of historic and current fitness and MSE values are supplied as outputs. These arrays are used as inputs in the next loop, until the final generation is reached.

This function may be used with any experiment using the current version of the GPOls toolbox. It should be included inside the generation loop, to allow it to obtain every generation’s population fitness and MSE. Note that the first generation’s fitness and MSE should be populated outside of this function.

E.4.8. GPOLS_VALIDATE

The residuals, fitness and MSE of the validation set, for a specific individual, are all calculated by this function. The latent variable regressors of the validation dataset is required as inputs. Furthermore the executable model string, GPOls parameters, population and index of the chosen individual needs to be supplied.

This function may be used with any experiment using the current version of the GPOls toolbox, provided that the *gpols_best_results* function is used to obtain an executable model.

E.5. Methods for Analysis and Presentation of System Identification Results

E.5.1. Trend of Population Evolution

The search for a solution runs the risk of getting stuck in a local optimum. The only way to adjust this is by either repeating the experiment, or adjusting the evolution parameters, such as mutation and cross over probabilities. Identifying whether the population is stuck in a local optimum requires a view of the whole population for each generation.

This was accomplished by making use of a 3D plot of the fitness and MSE, separately, for each individuals and generation. The result is a landscape of the aptitude of each individual in the population over all the generations. Investigation of this landscape assists in qualitatively identifying when individuals all became the same (local optima) or the influences of variations in parameters on the evolution of the population.

The function and discussion included in Appendix E - E.4.6 describes the procedure followed to obtain and store these trends.

E.5.2. Comparison to the Least Lag Outputs

If not enough information is available in the timeseries, then the previous process output value will be the best prediction for the next process output. It is expected that the GP algorithm will test this possibility by including such a model in the search space and removing it once a better model is generated. Nonetheless, as the GP is a stochastic

method and not certain to test this, any model identified should be compared to a model consisting of the least lagged process output. This is done by comparing the fitness of a model (calculated from the correlation between model output and expected output) to the correlation coefficient between the process output and the least lagged process output. The delay parameters and correlation coefficient of the training set is used.

This approach is followed for every latent variable reconstruction, as the correlation coefficient will differ depending on the lag chosen. The following correlation coefficients are used as comparison for the models identified according to the training set used. The correlation coefficients for lags identified by both the AMI and autocorrelation methods need to be assessed.

Table 50: Correlation coefficients of closest lagged process output. Obtained models are expected to have better correlation figures than these; otherwise the least lagged process output will be the best model.

Process	Dataset Number	Output Lag as per AMI (5sec increments)	Correlation Coefficient (AMI)	Output Lag as per Autocorrelation (5sec increments)	Correlation Coefficient (Autocorrelation)
Hot Gas Generator	1	65	0.085	76	0.306
Flash Dryer - Idle States Present	1	82	0.735	623	0.347
	2	41	0.943	405	0.608
	3	65	0.352	168	0.153
Flash Dryer – Idle States Removed	1	82	0.735	623	0.347
	2	56	0.779	365	0.059
	3	65	0.1102	109	0.049

Note that the fitness values of models are compared to these values. Although the fitness values are adjusted correlation coefficients, they will be used for comparison seeing as the fitness will never increase to above the correlation coefficients in the table above.

E.5.3. Comparison of Experiment Run Fit Statistics

An amount of repetitions of the GP experiments are required as GP is a stochastic system identification process. Coelho et al. (2009) used 50 repetitions for each of the GP experiments. In some cases in this research 40 repetitions were used.

For each repetition of a run the best model, according to training fitness, is saved during the SID run for further analysis, resulting in a collection of 40 to 50 individual solutions which need comparison and analysis to find the most representative model and population. The various solutions are compared based on the validation R^2 and thereafter the validation MSE values.

The comparison of various experiments requires information to be extracted from each population with regards to training and validation fitness and MSE's. This is done by building an experiment report which is consulted after the experiment is complete.

The report consists to the following headers and contains the information for every run of the specific experiment:

- Loop Number: the experiment repetition;
- Process Formula: interpretable process model;
- Executable Formula: Matlab® executable model;
- Train MSE;
- Train Fitness;
- Training R^2;
- Validation MSE;
- Validation Fitness;
- Validation R^2; and
- GP Options: Array of GPOls parameters.

This comparison allows

- identification of the best solution in this experimental setup;
- evaluation of the variables constituting this solution;
- variations to the solution;

- degree of convergence to a single solution for this experimental setup; and
- access to the model in executable and interpretable form.

From this knowledge it is possible to delve deeper into a specific experimental run's population. The experiment report is populated in the main experiment logic discussed in Appendix E - E.3.1 GP Experiment Logic.

E.5.4. R^2 and Nonlinear Modelling – Negative R^2

R^2 is a goodness of fit measurement, commonly referred to as the coefficient of determination and calculated as the square of the correlation coefficient between the model output and the desired output. In modelling in general R^2 is not the best measure of fit for the evolution of models, as it will increase as the decrease of freedom of the error decrease, i.e. when another variable is added to the model. R^2 will thus not be used for nonlinear goodness of fit measurement.

However, there is another use for R^2 which could assist in nonlinear modelling. R^2 is also calculated as one minus the square sum of errors (SSE) divided by the total corrected sum of squares (SST). This formulas are standard in statistics and ANOVA and are presented here:

$$SST = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

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

$$R^2 = 1 - \frac{SSE}{SST}$$

The benefit of this calculation of R^2 is in the fact that it will indicate whether the model is doing better than the mean. If the R^2 is negative, then the mean is a better model than the model used. The confusion that might exist is in the fact that a square of something can be negative, but from the equation it is apparent that this is possible. It results in the conclusion

that R^2 is not necessarily the square of the correlation coefficient, unless the variables correlated are jointly distributed random variables.

Furthermore, the fact that R^2 will increase with the addition of every term to the model limits the use of R^2. Furthermore the fact that nonlinear modelling do not necessarily add terms, but rather create more complex ones, makes the $R^2_{adjusted}$ calculation commonly used in statistics possibly unusable.

E.5.5. Residual and Model Output Trends

The trend of the model output for both the training and validation phases allows visual inspection of the best solution of the experiment. This may assist in identifying anomalies in the data which are not being identified by the model, as well as an idea to whether the model may be useful given the operating ranges and the anomalies which are captured in the model.

The normal plot of the residuals measures the amount of information still available in the residuals. Model error can be accounted as white noise if they are distributed normally. If not, then the error contains information, or is red noise. The usefulness of the model and success of the system identification experiment can be deduced from this.

E.5.6. Residual Analysis

The residual analysis includes the normal probability plot testing the hypothesis that the residual are white noise and thus normally distributed, if not the residuals still contain information.

Correlation and autocorrelation test between residuals and process inputs are also recommended as a residual analysis method for nonlinear dynamic systems (Hinchliffe & Willis, 2003). The approaches adopted in this research are visual inspection of the normal probability plot of the residuals, as well as the correlation coefficient between residuals and all the lagged inputs. This calculation will only be used to determine which variable is most likely to contribute to the accuracy of the model, and not as a comparative number to distinguish between models.

All residual analyses are done after experimentation and are not part of the GP search algorithm. These were also used on an ad-hoc basis.

E.5.7. Interpretation of the Model Empirical Formula

The GPOls toolbox provides the empirical formula of the model. Throughout the latent variable reconstruction of the dataset, the symbols were adjusted as to include the various lags of process variables. Hence the influence of a precise lagged version of a process input can be interpreted partially from the empirical formula. “Partially”, because the models are usually too complex to precisely see the influence.

For instance, large variation could be ascribed to the presence of a specific variable. Comparison between various solutions might identify that only one model contains this variable with high variation. Any unique behaviour could be due to this variables presence. This could guide further research or modelling exercises

Appendix F GPOls Toolbox Benchmarking

The results of benchmarking the GPOls toolbox re included in this appendix. The GPOls algorithm was compared against

i) a GP algorithm – Discipulus ®; and

ii) linear ARMA models.

F.1. Discipulus ® Benchmark Results

The following tables include the training and validation results for each of the 40 repitions of each of the 4 experiments.

F.1.1. Flash Dryer AMI Latent Variable Reconstruction

Table 51: Fit Statistics for each of the Discipulus ® GP Runs for Flash Dryer modelling with AMI Latent Variable Reconstruction

Run	Train MSE	Train R^2	Validation MSE	Validation R^2
1	0.395	51.9%	0.675	44.4%
2	0.400	51.2%	0.724	40.4%
3	0.464	43.4%	0.803	33.9%
4	0.433	47.2%	0.617	49.2%
5	0.623	24.0%	0.831	31.6%
6	0.568	30.7%	0.828	31.9%
7	0.569	30.6%	0.785	35.4%
8	0.391	52.3%	0.772	36.4%
9	0.395	51.8%	0.711	41.4%
10	0.322	60.7%	0.659	45.8%
11	0.341	58.5%	0.677	44.2%
12	0.477	41.9%	0.784	35.4%
13	0.488	40.6%	0.733	39.6%
14	0.517	37.0%	0.787	35.2%
15	0.460	43.9%	0.799	34.2%
16	0.521	36.4%	0.751	38.1%
17	0.470	42.7%	0.726	40.2%
18	0.504	38.5%	0.778	36.0%
19	0.344	58.1%	0.680	44.0%
20	0.409	50.2%	0.736	39.4%
21	0.405	50.7%	0.731	39.8%
22	0.399	51.3%	0.814	33.0%
23	0.586	28.6%	0.806	33.6%
24	0.489	40.4%	0.781	35.7%
25	0.390	52.5%	0.683	43.8%
26	0.430	47.5%	0.618	49.1%
27	0.369	55.1%	0.688	43.4%
28	0.585	28.7%	0.880	27.5%
29	0.405	50.6%	0.734	39.6%
30	0.395	51.9%	0.675	44.4%
31	0.395	51.9%	0.675	44.4%
32	0.400	51.2%	0.724	40.4%
33	0.464	43.4%	0.803	33.9%
34	0.433	47.2%	0.617	49.2%
35	0.623	24.0%	0.831	31.6%
36	0.568	30.7%	0.828	31.9%
37	0.569	30.6%	0.785	35.4%
38	0.391	52.3%	0.772	36.4%
39	0.395	51.8%	0.711	41.4%
40	0.322	60.7%	0.659	45.8%

F.1.2. Flash Dryer Autocorrelation Latent Variable Reconstruction

Table 52: Fit Statistics for each of the Discipulus ® GP Runs for Flash Dryer modelling with Autocorrelation Latent Variable Reconstruction

Run	Train MSE	Train R^2	Validation MSE	Validation R^2
1	0.486	36.2%	0.824	32.1%
2	0.393	48.4%	0.735	39.5%
3	0.371	51.3%	0.756	37.7%
4	0.363	52.4%	0.691	43.1%
5	0.392	48.5%	0.698	42.6%
6	0.412	46.0%	0.674	44.5%
7	0.378	50.4%	0.824	32.1%
8	0.396	48.0%	0.720	40.7%
9	0.382	49.9%	0.736	39.4%
10	0.948	-24.3%	1.022	15.8%
11	0.379	50.3%	0.728	40.1%
12	0.424	44.4%	0.729	40.0%
13	0.687	9.8%	0.929	23.5%
14	0.343	55.0%	0.640	47.3%
15	0.405	46.9%	0.778	36.0%
16	0.414	45.6%	0.702	42.2%
17	0.364	52.2%	0.679	44.1%
18	0.355	53.4%	0.747	38.5%
19	0.376	50.7%	0.724	40.4%
20	0.393	48.5%	0.699	42.4%
21	0.404	47.0%	0.743	38.8%
22	0.466	38.8%	0.741	39.0%
23	0.442	42.1%	0.693	42.9%
24	0.443	41.9%	0.772	36.4%
25	0.494	35.2%	0.850	30.0%
26	0.444	41.8%	0.694	42.9%
27	0.407	46.6%	0.676	44.3%
28	0.409	46.4%	0.708	41.7%
29	0.491	35.6%	0.706	41.8%
30	0.419	45.0%	0.780	35.8%
31	0.321	57.9%	0.675	44.4%
32	0.416	45.4%	0.676	44.3%
33	0.578	24.2%	0.740	39.1%
34	0.428	43.9%	0.644	47.0%
35	0.506	33.7%	0.749	38.4%
36	0.370	51.4%	0.734	39.6%
37	0.355	53.4%	0.651	46.4%
38	0.380	50.2%	0.661	45.5%
39	0.466	38.9%	0.892	26.5%
40	0.410	46.2%	0.720	40.7%

F.1.3. Hot Gas Generator AMI Latent Variable Reconstruction

Table 53: Fit Statistics for each of the Discipulus ® GP Runs for Hot Gas Generator modelling with AMI Latent Variable Reconstruction

Run	Train MSE	Train R^2	Validation MSE	Validation R^2
1	0.313	48.5%	0.159	72.7%
2	0.052	91.4%	0.105	81.9%
3	0.430	29.4%	0.144	75.1%
4	0.074	87.9%	0.163	72.0%
5	0.445	26.9%	0.141	75.8%
6	0.077	87.4%	0.188	67.6%
7	0.215	64.8%	0.275	52.7%
8	0.116	81.0%	0.117	79.8%
9	0.147	75.9%	0.129	77.8%
10	0.062	89.8%	0.166	71.5%
11	0.079	87.1%	0.125	78.5%
12	0.136	77.6%	0.207	64.3%
13	0.397	34.7%	0.169	71.0%
14	0.089	85.4%	0.124	78.7%
15	0.329	46.0%	0.234	59.7%
16	0.329	46.0%	0.234	59.7%
17	0.329	46.0%	0.234	59.7%
18	0.073	88.0%	0.168	71.1%
19	0.240	60.5%	0.189	67.5%
20	0.279	54.2%	0.164	71.8%
21	0.090	85.1%	0.187	67.8%
22	0.397	34.7%	0.357	38.6%
23	0.138	77.3%	0.175	69.8%
24	0.168	72.4%	0.164	71.8%
25	0.053	91.3%	0.089	84.6%
26	0.062	89.8%	0.153	73.7%
27	0.253	58.4%	0.118	79.6%
28	0.397	34.7%	0.169	71.0%
29	0.064	89.5%	0.088	84.8%
30	0.074	87.8%	0.201	65.4%
31	0.293	51.8%	0.187	67.9%
32	0.053	91.3%	0.162	72.1%
33	0.099	83.8%	0.122	78.9%
34	0.089	85.3%	0.265	54.4%
35	0.128	79.0%	0.131	77.5%
36	0.091	85.0%	0.078	86.6%
37	0.143	76.4%	0.144	75.3%
38	0.179	70.5%	0.203	65.0%
39	0.052	91.5%	0.093	83.9%
40	0.070	88.5%	0.087	85.1%

F.1.4. Hot Gas Generator Autocorrelation Latent Variable Reconstruction

Table 54: Fit Statistics for each of the Discipulus ® GP Runs for Hot Gas Generator modelling with Autocorrelation Latent Variable Reconstruction

Run	Train MSE	Train R^2	Validation MSE	Validation R^2
1	0.074	87.8%	0.061	89.5%
2	0.096	84.2%	0.070	88.0%
3	0.082	86.5%	0.076	86.8%
4	0.110	81.8%	0.100	82.7%
5	0.077	87.4%	0.059	89.8%
6	0.124	79.6%	0.092	84.2%
7	0.073	88.0%	0.112	80.8%
8	0.186	69.4%	0.143	75.3%
9	0.074	87.8%	0.088	84.8%
10	0.179	70.5%	0.121	79.2%
11	0.048	92.0%	0.053	90.9%
12	0.164	73.0%	0.121	79.3%
13	0.224	63.1%	0.138	76.2%
14	0.112	81.5%	0.066	88.6%
15	0.068	88.9%	0.061	89.6%
16	0.129	78.8%	0.123	78.9%
17	0.223	63.3%	0.111	81.0%
18	0.255	58.0%	0.146	74.9%
19	0.286	53.0%	0.077	86.8%
20	0.183	69.9%	0.104	82.0%
21	0.116	80.8%	0.063	89.1%
22	0.187	69.2%	0.095	83.7%
23	0.131	78.4%	0.112	80.8%
24	0.160	73.7%	0.100	82.8%
25	0.065	89.3%	0.052	91.0%
26	0.075	87.7%	0.076	87.0%
27	0.243	59.9%	0.114	80.4%
28	0.056	90.7%	0.070	88.0%
29	0.073	88.0%	0.067	88.5%
30	0.127	79.0%	0.058	90.1%
31	0.265	56.3%	0.128	78.0%
32	0.089	85.4%	0.060	89.7%
33	0.268	55.8%	0.086	85.2%
34	0.247	59.4%	0.150	74.1%
35	0.080	86.9%	0.055	90.5%
36	0.102	83.2%	0.090	84.5%
37	0.249	58.9%	0.127	78.2%
38	0.136	77.5%	0.121	79.2%
39	0.152	74.9%	0.090	84.6%
40	0.091	85.1%	0.070	88.0%

F.2. ARMA Model Comparison

The coefficients for each of the variables for the identified ARMA models are included in this appendix. This is included for each of the four experiments done. The CSense Linear Model Tool was used together with the latent variable reconstruction.

The general model structure is as follows:

$$\hat{y} = \sum c_i x_i + C$$

The number of variables is equal to the number of reconstructed latent variables.

F.2.1. Flash Dryer AMI Latent Variable Reconstruction

Table 55:Linear Model Coefficients for the Flash Dryer model using the AMI Latent Variable Reconstruction

Latent Variable (x_i)	Coefficient (c_i)
FDFeed(k-0)	-0.059
FDFeed(k-16)	-0.450
FDFeed(k-32)	-0.202
FDFeed(k-48)	-0.093
FDFeed(k-64)	-0.095
FDFeed(k-80)	-0.157
FDFeed(k-96)	-0.218
FDFeed(k-112)	-0.209
FDFeed(k-128)	-0.167
FDFeed(k-144)	-0.115
FDFeed(k-160)	-0.071
HGGTemp(k-0)	0.243
HGGTemp(k-56)	-0.243
HGGTemp(k-112)	0.135
HGGTemp(k-168)	-0.170
HGGTemp(k-224)	-0.002
HGGTemp(k-280)	-0.347
HGGTemp(k-336)	0.202
FDTemp(k-65)	-0.001
FDTemp(k-130)	-0.026
FDTemp(k-195)	0.074
FDTemp(k-260)	-0.043
FDTemp(k-325)	-0.131
Constant (C)	0.122

F.2.2. Flash Dryer Autocorrelation Latent Variable Reconstruction

Table 56:Linear Model Coefficients for the Flash Dryer model using the Autocorrelation Latent Variable Reconstruction

Latent Variable (x_i)	Coefficient (c_i)
FDFeed(k-0)	-0.059
FDFeed(k-16)	-0.433
FDFeed(k-32)	-0.203
FDFeed(k-48)	-0.128
FDFeed(k-64)	-0.131
FDFeed(k-80)	-0.193
FDFeed(k-96)	-0.265
FDFeed(k-112)	-0.262
FDFeed(k-128)	-0.217
FDFeed(k-144)	-0.145
FDFeed(k-160)	-0.091
HGGTemp(k-0)	-0.319
HGGTemp(k-363)	0.000
HGGTemp(k-726)	0.049
HGGTemp(k-1089)	0.094
HGGTemp(k-1452)	-0.077
HGGTemp(k-1815)	-0.111
FDTemp(k-109)	-0.015
FDTemp(k-218)	0.075
FDTemp(k-327)	-0.110
FDTemp(k-436)	0.005
FDTemp(k-545)	-0.053
Constant (C)	0.157

F.2.3. Hot Gas Generator AMI Latent Variable Reconstruction

Table 57:Linear Model Coefficients for the Hot Gas Generator model using the AMI Latent Variable Reconstruction

Latent Variable (x_i)	Coefficient (c_i)
CoalFeed(k-0)	-0.225
CoalFeed(k-44)	0.234
CoalFeed(k-88)	0.280
CoalFeed(k-132)	-0.036
CoalFeed(k-176)	-0.044
CoalFeed(k-220)	-0.079
CoalFeed(k-264)	-0.053
CoalFeed(k-308)	-0.089
CoalFeed(k-352)	-0.059
CoalFeed(k-396)	-0.080
CoalFeed(k-440)	-0.067
HGGTemp(k-65)	0.303
HGGTemp(k-130)	-0.439
HGGTemp(k-195)	-0.183
HGGTemp(k-260)	0.176
Constant (C)	-0.170

F.2.4. Hot Gas Generator Autocorrelation Latent Variable Reconstruction

Table 58:Linear Model Coefficients for the Hot Gas Generator model using the Autocorrelation Latent Variable Reconstruction

Latent Variable (x_i)	Coefficient (c_i)
CoalFeed(k-0)	-0.245
CoalFeed(k-45)	0.203
CoalFeed(k-90)	0.291
CoalFeed(k-135)	0.035
CoalFeed(k-180)	-0.064
CoalFeed(k-225)	-0.096
CoalFeed(k-270)	-0.103
CoalFeed(k-315)	-0.042
CoalFeed(k-360)	-0.057
CoalFeed(k-405)	-0.073
CoalFeed(k-450)	-0.091
HGGTemp(k-76)	0.099
HGGTemp(k-152)	-0.447
HGGTemp(k-228)	-0.027
HGGTemp(k-304)	0.161
Constant (C)	-0.172

Appendix G – System Identification Experiment Outputs

G.1. AMI versus Autocorrelation Latent Variable Reconstruction Results

G.1.1. Flash Dryer

For the Flash Dryer the following model fit statistics were obtained for the two delay parameter sets. The groups are compared.

Table 59: List of experiments used to identify which of the AMI or Autocorrelation delay parameters result in the best model. Comparative experiments are grouped.

Comparative Grouping	Latent Variable Delay Parameters	Train MSE	Train R^2	Validation MSE	Validation R^2
A	AMI	0.666	33.4%	NA	NA
A	Autocorrelation	0.784	21.9%	NA	NA
B	AMI	0.623	43.7%	0.715	34.1%
B	Autocorrelation	0.656	13.9%	0.868	20.4%
C	AMI	0.526	35.9%	0.701	35.5%
C	Autocorrelation	0.512	32.9%	0.674	38.2%
D	AMI	0.451	45.1%	0.728	33.0%
D	Autocorrelation	0.447	41.3%	0.728	33.3%
E	AMI	0.450	45.2%	0.642	40.8%
E	Autocorrelation	0.485	36.4%	0.641	41.3%
F	AMI	0.529	35.5%	0.784	27.8%
F	Autocorrelation	0.665	12.7%	0.951	12.8%
G	AMI	0.488	40.5%	0.668	38.5%
G	Autocorrelation	0.500	34.4%	0.697	36.1%
H	AMI	0.422	51.7%	0.646	40.5%
H	Autocorrelation	0.468	38.6%	0.622	43.0%
I	AMI	0.403	50.9%	0.620	42.9%
I	Autocorrelation	0.481	36.9%	0.800	26.7%
J	AMI	0.488	40.5%	0.668	38.5%
J	Autocorrelation	0.500	34.4%	0.697	36.1%
K	AMI	0.468	43.0%	0.655	39.7%
K	Autocorrelation	0.434	43.1%	0.651	40.3%

G.1.2. Hot Gas Generator

For the Hot Gas Generator the following model fit statistics were obtained for the two delay parameter sets. The groups are compared.

Table 60: The HGG system identification experiments indicate that the delay parameters identified by autocorrelation are preferred. Comparative experiments are grouped.

Comparative Grouping	Latent Variable Delay Parameters	Validation MSE	Validation R^2
A	Autocorrelation	0.052	91.0%
A	AMI	0.078	86.6%
B	Autocorrelation	0.133	77.1%
B	AMI	0.241	58.5%
C	Autocorrelation	0.181	68.9%
C	AMI	0.192	66.8%
D	Autocorrelation	0.133	77.1%
D	AMI	0.167	71.2%

G.2. Flash Dryer

45 different experiments were attempted to identify a flash dryer model. The tables below contains the experiments (1) ordered from lowest validation mean square error to highest and (2) numerical order. The most important characteristics of each experiments is included, namely

- Delay parameters used;
- Presence of process idle state;
- Training and validation datasets;
- Training fitness, MSE and R^2; and
- Validation fitness, MSE and R^2.

Fitness values for only the GPOls tool is included, seeing as the MSE value was used by Discipulus®.

G.2.1. Ordered According to Descending MSE

Table 61: All the flash dryer models identified per experiment. Results are ordered in descending order of validation MSE.

Experiment	Latent Variable Delay Parameters	Idle State	Train Dataset	Train MSE	Train Fitness	Train R^2	Validation Dataset	Validation MSE	Validation Fitness	Validation R^2
11	Autocorrelation	NA	3	0.310	54.4%	68.6%	3	0.314	18.5%	-14.8%
12	AMI	NA	3	0.395	55.9%	22.9%	3	0.418	1.1%	-58.7%
30	AMI	NA	1	0.395	55.9%	60.7%	3	0.418	1.1%	-53.5%
Discipulus	AMI	NA	3	0.433	NA	47.2%	3	0.617	NA	49.2%
40	AMI	NA	3	0.403	49.0%	50.9%	3	0.620	42.3%	42.9%
17	AMI	NA	3	0.430	47.5%	47.6%	3	0.622	43.6%	42.7%
39	Autocorrelation	NA	3	0.468	37.4%	38.6%	3	0.622	42.9%	43.0%
19	AMI	NA	3	0.391	52.3%	52.3%	3	0.633	43.8%	41.7%
Discipulus	Autocorrelation	NA	3	0.343		55.0%	3	0.640	NA	47.3%
10	Autocorrelation	NA	3	0.485	22.2%	36.4%	3	0.641	25.7%	41.3%
24	Autocorrelation	NA	1	0.485	22.2%	36.4%	3	0.641	25.7%	41.3%
25	Autocorrelation	NA	1	0.485	22.2%	36.4%	3	0.641	25.7%	41.3%
27	Autocorrelation	NA	3	0.485	31.9%	36.4%	3	0.641	36.9%	41.3%
26	AMI	NA	3	0.450	39.5%	45.2%	3	0.642	37.2%	40.8%
38	AMI	NA	3	0.422	49.6%	51.7%	3	0.646	40.7%	40.5%
45	Autocorrelation	NA	3	0.434	41.1%	43.1%	3	0.651	39.0%	40.3%
44	AMI	NA	3	0.468	41.6%	43.0%	3	0.655	38.8%	39.7%
16	AMI	NA	3	0.491	40.1%	40.1%	3	0.662	39.4%	39.0%
36	AMI	NA	3	0.488	33.4%	40.5%	3	0.668	32.0%	38.5%
42	AMI	NA	3	0.488	39.8%	40.5%	3	0.668	38.2%	38.5%

Experiment	Latent Variable Delay Parameters	Idle State	Train Dataset	Train MSE	Train Fitness	Train R^2	Validation Dataset	Validation MSE	Validation Fitness	Validation R^2
20	AMI	NA	3	0.512	25.8%	32.9%	3	0.674	30.5%	38.2%
37	Autocorrelation	Idle	3	0.500	29.3%	34.4%	3	0.697	31.5%	36.1%
43	Autocorrelation	Idle	3	0.500	29.3%	34.4%	3	0.697	31.5%	36.1%
21	AMI	NA	3	0.526	29.3%	35.9%	3	0.701	29.8%	35.5%
15	AMI	NA	3	0.623	43.7%	43.7%	3	0.715	38.1%	34.1%
29	Autocorrelation	NA	3	0.447	36.6%	41.3%	3	0.728	32.2%	33.3%
28	AMI	NA	3	0.451	40.8%	45.1%	3	0.728	31.5%	33.0%
34	AMI	NA	3	0.529	30.7%	35.5%	3	0.784	24.9%	27.8%
41	Autocorrelation	NA	3	0.481	36.3%	36.9%	3	0.800	29.3%	26.7%
23	Autocorrelation	NA	1	0.351	44.5%	44.5%	3	0.839	26.6%	22.7%
18	Autocorrelation	NA	3	0.656	20.7%	13.9%	3	0.868	21.8%	20.4%
33	AMI	NA	3	0.671	16.8%	18.2%	3	0.869	21.4%	20.0%
22	AMI	NA	3	0.456	54.3%	54.3%	3	0.897	14.8%	13.0%
35	Autocorrelation	NA	3	0.665	11.7%	12.7%	3	0.951	19.5%	12.8%
3	AMI	NA	3	0.463	53.9%	54.1%	3	1.413	9.1%	-55.2%
9	AMI	NA	3	0.463	53.9%	54.0%	3	1.413	9.1%	-55.2%
13	Autocorrelation	NA	3	0.673	21.6%	32.9%	1	2.149	32.4%	40.6%
1	AMI	NA	3	0.650	33.4%	35.0%	1	2.204	47.0%	40.4%
6	AMI	NA	3	0.666	33.4%	33.4%	1	2.215	49.2%	40.0%
8	Autocorrelation	NA	3	0.784	21.9%	21.9%	1	2.269	52.9%	37.3%
2	AMI	NA	3	0.630	29.1%	37.1%	1	2.350	37.9%	36.4%
8	Autocorrelation	NA	3	0.637	36.5%	36.5%	1	2.744	52.8%	24.2%
14	Autocorrelation	NA	3	0.697	45.0%	46.2%	1	3.203	42.2%	11.5%
4	AMI	NA	3	0.822	17.9%	17.9%	1	3.492	48.4%	5.4%
5	AMI	NA	3	0.822	17.9%	17.9%	1	3.492	48.4%	5.4%

G.2.1. Ordered According to Experiment Number

Table 62:All the flash dryer models identified per experiment. Results are ordered in order of ascending experiment number.

Experiment	Latent Variable Delay Parameters	Idle	Train Dataset	Train MSE	Train Fitness	Train R^2	Validation Dataset	Validation MSE	Validation Fitness	Validation R^2
1	AMI	NA	3	0.650	33.4%	35.0%	1	2.204	47.0%	40.4%
2	AMI	NA	3	0.630	29.1%	37.1%	1	2.350	37.9%	36.4%
3	AMI	NA	1	0.463	53.9%	54.1%	3	1.413	9.1%	-55.2%
4	AMI	NA	3	0.822	17.9%	17.9%	1	3.492	48.4%	5.4%
5	AMI	NA	3	0.822	17.9%	17.9%	1	3.492	48.4%	5.4%
6	AMI	NA	3	0.666	33.4%	33.4%	1	2.215	49.2%	40.0%
7	AMI	NA	3	0.573	42.7%	42.8%	1	4.101	21.0%	-1107.0%
8	Autocorrelation	NA	3	0.784	21.9%	21.9%	1	2.269	52.9%	37.3%
9	AMI	NA	1	0.637	36.5%	36.5%	3	2.744	52.8%	24.2%
10	Autocorrelation	NA	3	0.600	40.2%	40.3%	3	3.878	39.1%	-7.1%
11	Autocorrelation	NA	1	0.463	53.9%	54.0%	3	1.413	9.1%	-55.2%
12	AMI	NA	1	0.485	22.2%	36.4%	3	0.641	25.7%	41.3%
13	Autocorrelation	NA	3	0.310	54.4%	68.6%	1	0.314	18.5%	-14.8%
14	Autocorrelation	NA	3	0.395	55.9%	22.9%	1	0.418	1.1%	-58.7%
15	AMI	NA	3	0.673	21.6%	32.9%	3	2.149	32.4%	40.6%
16	AMI	NA	3	0.697	45.0%	46.2%	3	3.203	42.2%	11.5%
17	AMI	NA	3	0.623	43.7%	43.7%	3	0.715	38.1%	34.1%
18	Autocorrelation	NA	3	0.491	40.1%	40.1%	3	0.662	39.4%	39.0%
19	AMI	NA	3	0.430	47.5%	47.6%	3	0.622	43.6%	42.7%
20	AMI	NA	3	0.656	20.7%	13.9%	3	0.868	21.8%	20.4%
21	AMI	NA	3	0.391	52.3%	52.3%	3	0.633	43.8%	41.7%
22	AMI	Idle	3	0.512	25.8%	32.9%	3	0.674	30.5%	38.2%
23	Autocorrelation	Idle	3	0.526	29.3%	35.9%	3	0.701	29.8%	35.5%

Experiment	Latent Variable Delay Parameters	Idle	Train Dataset	Train MSE	Train Fitness	Train R^2	Validation Dataset	Validation MSE	Validation Fitness	Validation R^2
24	Autocorrelation	NA	3	0.456	54.3%	54.3%	3	0.897	14.8%	13.0%
25	Autocorrelation	NA	3	0.351	44.5%	44.5%	3	0.839	26.6%	22.7%
26	AMI	NA	3	0.485	22.2%	36.4%	3	0.641	25.7%	41.3%
27	Autocorrelation	NA	3	0.485	22.2%	36.4%	3	0.641	25.7%	41.3%
28	AMI	NA	3	0.450	39.5%	45.2%	3	0.642	37.2%	40.8%
29	Autocorrelation	NA	3	0.485	31.9%	36.4%	3	0.641	36.9%	41.3%
30	AMI	NA	1	0.451	40.8%	45.1%	3	0.728	31.5%	33.0%
33	AMI	NA	3	0.447	36.6%	41.3%	3	0.728	32.2%	33.3%
34	AMI	NA	3	0.395	55.9%	60.7%	3	0.418	1.1%	-53.5%
35	Autocorrelation	NA	3	0.671	16.8%	18.2%	3	0.869	21.4%	20.0%
36	AMI	NA	3	0.529	30.7%	35.5%	3	0.784	24.9%	27.8%
37	Autocorrelation	NA	3	0.665	11.7%	12.7%	3	0.951	19.5%	12.8%
38	AMI	NA	3	0.488	33.4%	40.5%	3	0.668	32.0%	38.5%
39	Autocorrelation	NA	3	0.500	29.3%	34.4%	3	0.697	31.5%	36.1%
40	AMI	NA	3	0.422	49.6%	51.7%	3	0.646	40.7%	40.5%
41	Autocorrelation	NA	3	0.468	37.4%	38.6%	3	0.622	42.9%	43.0%
42	AMI	NA	3	0.403	49.0%	50.9%	3	0.620	42.3%	42.9%
43	Autocorrelation	NA	3	0.481	36.3%	36.9%	3	0.800	29.3%	26.7%
44	AMI	NA	3	0.488	39.8%	40.5%	3	0.668	38.2%	38.5%
45	Autocorrelation	NA	3	0.500	29.3%	34.4%	3	0.697	31.5%	36.1%
Discipulus	AMI	NA	3	0.468	41.6%	43.0%	3	0.655	38.8%	39.7%
Discipulus	Autocorrelation	NA	3	0.434	41.1%	43.1%	3	0.651	39.0%	40.3%

G.3. Hot Gas Generator

45 different experiments were attempted with the aim to identify a hot gas generator model. The tables below contain the experiments in 1) ordered from lowest validation mean square error to highest and 2) numerical order. The most important characteristics of each experiments is included, namely

- Delay parameters used;
- Presence of process idle state;
- Training and validation datasets;
- Training fitness, MSE and R^2; and
- Validation fitness, MSE and R^2.

Fitness values for only the GPOls tool is included, seeing as the MSE value was used by Discipulus®.

G.3.1. Ordered According to Descending MSE

Table 63: All the hot gas generator models identified per experiment. Results are ordered in descending order of validation MSE.

Experiment	Latent Variable Delay Parameters	Idle	Train Dataset	Train MSE	Train Fitness	Train R^2	Validation Dataset	Validation MSE	Validation Fitness	Validation R^2
Discipulus	Autocorrelation	NA	3	0.065	NA	89.3%	3	0.052	NA	91.0%
Discipulus	AMI	NA	3	0.091	NA	85.0%	3	0.078	NA	86.6%
32	Autocorrelation	NA	1	0.165	52.5%	72.8%	3	0.133	61.8%	77.1%
49	Autocorrelation	NA	3	0.093	81.8%	84.7%	3	0.133	76.7%	77.1%
48	AMI	NA	3	0.103	80.8%	83.1%	3	0.167	71.5%	71.2%
47	Autocorrelation	NA	3	0.223	62.7%	63.2%	3	0.181	79.7%	68.9%
46	AMI	NA	3	0.245	59.3%	59.8%	3	0.192	80.5%	66.8%
31	AMI	NA	3	0.159	57.4%	73.9%	3	0.241	65.1%	58.5%

G.3.2. Ordered According to Experiment Number

Table 64: All the hot gas generator models identified per experiment. Results are ordered according to experiment number.

Experiment	Latent Variable Delay Parameters	Idle	Train Dataset	Train MSE	Train Fitness	Train R^2	Validation Dataset	Validation MSE	Validation Fitness	Validation R^2
31	AMI	NA	1	0.159	57.4%	73.9%	1	0.241	65.1%	58.5%
32	Autocorrelation	NA	1	0.165	52.5%	72.8%	1	0.133	61.8%	77.1%
46	AMI	NA	1	0.245	59.3%	59.8%	1	0.192	80.5%	66.8%
47	Autocorrelation	NA	1	0.223	62.7%	63.2%	1	0.181	79.7%	68.9%
48	AMI	NA	1	0.103	80.8%	83.1%	1	0.167	71.5%	71.2%
49	Autocorrelation	NA	1	0.093	81.8%	84.7%	1	0.133	76.7%	77.1%
Discipulus	AMI	NA	1	0.091		85.0%	1	0.078		86.6%
Discipulus	Autocorrelation	NA	1	0.065		89.3%	1	0.052		91.0%

Appendix H – Model-Based Predictive Control Theory

This appendix provides an overview of basic MPC theory as found in literature. This is included for the reader not familiar with the technique.

H.1. Model-based Predictive Control

What is a APC? Advanced process control (APC) refers loosely to any control algorithm, which is not traditional PID based control. An APC algorithm can thus include predictive methods, conditions, rules, “what-if”-simulations or optimisation, operator guidance systems or any method that allows a decision or control move to be executed. There is thus no formal standard, although various methods have been formally researched and is recognised. One of these is model-based predictive control.

Model-based Predictive Control (MPC) is one of various model based control strategies available. MPC was first coined by 2 separate industrial research groups in the late 1970’s. Shell Oil developed their dynamic matrix control (DMC) method which focused on multivariate constrained control problems; and a French company, ADERSA, used their IDCOM method which was similar to the DMC method. The general MPC concept has evolved from these and is today widely used in especially the petrochemical industry. (De Temmerman et al., 2009)(Perry & Green, 1997)

MPC is a preferred model based control strategy, and also preferred over classical control strategies due to its ability to:

- Handle MIMO models;
- Incorporate difficult dynamic behaviour such as time-delays;
- Include input and output variable constraints;
- Integrate with various optimisation schemes;
- Update easily online;
- Adaptable cost function which may include economic and energy considerations;

- Track a set point trajectory;

(Dufour et al., 2003) (Perry & Green, 1997)

There are however some disadvantages a control engineer should be weary of when developing or recommending a MPC strategy.

- It is unfamiliar to plant personnel, whereas classic PID control is proven and trusted on site;
- Demanding of computer resources due to the intense and reiterative optimisation procedures;
- Model development may be time consuming and difficult;
- Models may be limited in the process operating range they can represent;

(Dufour et al., 2003) (Perry & Green, 1997)

These issues can be answered by focussing on change management, computing power and algorithm efficiency for the first two disadvantages. The model limitations and development workload should be brought into consideration when planning the timelines of the project and using proven methods and experienced people.

H.2. Workings of MPC

The idea of MPC is to apply a dynamic model for predicting process output (control variable) values; comparing these predictions to the required set point value and then calculating and optimising the required manipulated variable adjustments needed to drive the control variable towards the required set point.

Consider a process with an input variable set u; output variable set y; model G. MPC predicts the output values y_j for a future discrete time window with a maximum prediction window N_p discrete time events forward from current time k. From these predicted values the control algorithm generates the control moves $\tilde{u}$ for a future control time window with a maximum control window N_c discrete time events forward from current time k. Only the first

control move for time *k+1* is implemented. The following diagram illustrates the prediction, control move generation and implementation over a future timeline from current time *k*.

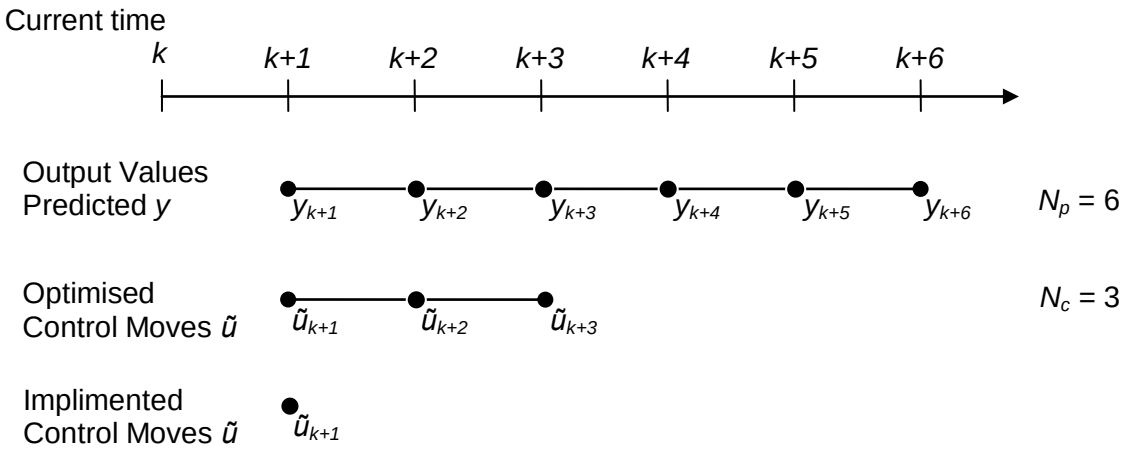


Figure 104: Illustration of MPC prediction and control move implementation at time *k*

The optimisation procedure solves an iterative open loop optimisation problem aimed at finding the best manipulated variable values. Often this needs to be executed within an allotted optimisation time. The goal function for the optimisation procedure is commonly made up of a set point tracking error together with a second penalty criteria for either too large control moves (Abudkhalifeh et al., 2005) (Abdel-Jabbar et al., 2002)(Perry & Green, 1997) or for enforcing output constraints (Dufour et al., 2003) (De Temmerman et al., 2009). The goal function may be linear, quadratic or a higher order nonlinear depending on the criteria of the control output. The order of the goal function implies which optimisation procedure to use – linear programming (LP), quadratic programming (QP), or a nonlinear programming method. (Perry & Green, 1997) This optimisation procedure is repeated at every sample time. The goal function for set point tracking with penalties for too large control moves is

$$minJ(\tilde{u}) = \sum_{i=1}^{N_p}\left[\gamma(i)\left(y_{ref}(k+i) - y(k+i)\right)^2\right] + \sum_{j=1}^{N_c}\left[\lambda(j)\left(\Delta\tilde{u}(k+j-1)\right)^2\right]$$

(Abdel-Jabbar et al., 2002)

Where γ and λ are weights for the square prediction errors for time *i* and control moves for time *j*. In the case of multiple manipulated variables, the off diagonals of the matrix λ indicate the independence of the manipulated variables. (Abudkhalifeh et al., 2005)

Over and above control move size penalties, MPC can incorporate control variable constraints into the control algorithm. These constraints may be either on the variable size, velocity or acceleration and is denoted as follows:

$$u_{min} \leq \tilde{u} \leq u_{max}$$

$$\dot{u}_{min} \leq \dot{\tilde{u}} \leq \dot{u}_{max}$$

$$\ddot{u}_{min} \leq \ddot{\tilde{u}} \leq \ddot{u}_{max}$$

(Dufour et al., 2003)(De Temmerman et al., 2009)

MATLAB's Model Predictive Control toolbox follows the abovementioned procedure (Abdel-Jabbar et al., 2002), but only supports linear optimisation. The fundamentals discussed here will be used to construct MPC logic and investigating its efficiency and effectiveness.

Appendix I – Model Predictive Control Experiments

The results for the randomly chosen control moves are included here. These trends are included for reference to the trends in Chapter 9.

I.1. Random Control Model 2

The outlet air temperature tends to hit a ceiling and deviate downwards. This is seen as the natural reaction of the model.

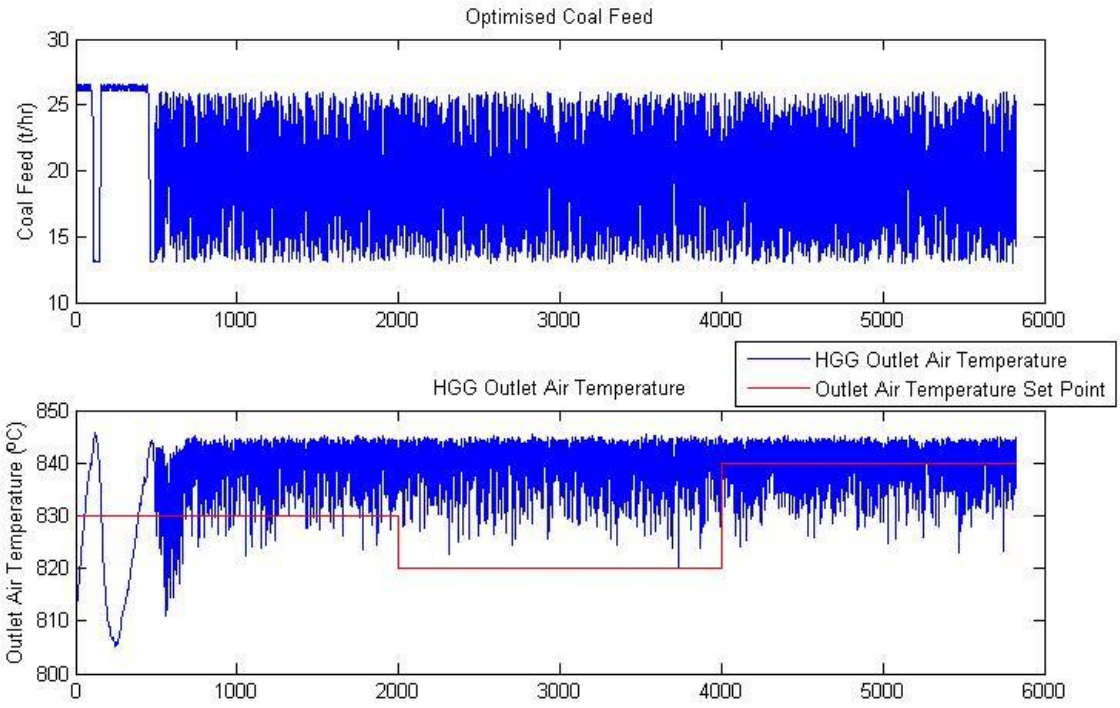


Figure 105: Outlet air temperature resulting from a random choice of coal feed – Run 3

I.2. Random Control Model 4

The model used results in an outlet air temperature oscillating around a more central area, indicating a more normal distribution in the actions.

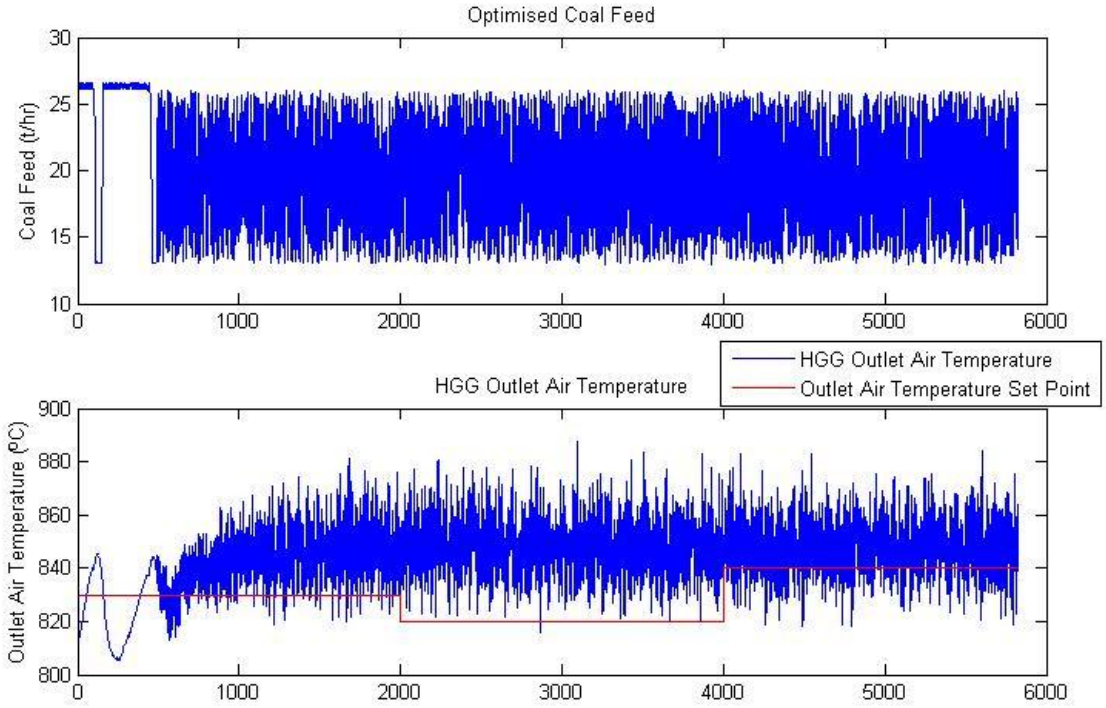


Figure 106: Outlet air temperature resulting from a random choice of coal feed – Run 4