Process identification using second order Volterra models for nonlinear model predictive control design of flotation circuits

Ruanne Delport

Process identification using second order Volterra models for nonlinear model predictive control design of flotation circuits

by

Ruanne Delport

A dissertation submitted in partial fulfillment
of the requirements for the degree

Master of Engineering (Control Engineering)

in the

Department of Chemical Engineering
Faculty of Engineering, the Built Environment and Information Technology

University of Pretoria
Pretoria

2nd December 2004

Synopsis

The control of flotation circuits is a complicated problem, since flotation circuits are nonlinear multivariable processes with a significant degree of interaction between the variables. Isolated PID controllers usually do not perform adequately. The application of a nonlinear model predictive algorithm based on second order Volterra models was investigated. Volterra series models are a higher order extension of linear impulse response models. The nonlinear model predictive control algorithm can also be seen as a linear model predictive controller with higher order correction terms.

A dynamic model of a flotation circuit based on the governing continuity equations was developed. The responses obtained represented the qualitative relationships between the model inputs and the controlled variables. This model exhibited strong nonlinearities, including asymmetrical responses to symmetrical inputs and gain sign changes.

This dynamic model was treated as the plant to be identified and from which second order Volterra models were obtained. Full Volterra models required excessively large data sets, but significant reductions in the size of the required data set could be achieved if some of the second order coefficients were constrained to zero. These "pruned" Volterra models represented the plant dynamics significantly better than linear models. In particular, these second order Volterra models were able to model asymmetrical responses including gain sign changes. A special case of "pruned" second order Volterra models are diagonal second order models, where all the off-diagonal coefficients (h_{ij} where $i \neq j$) are constrained to zero. These models required less data than pruned Volterra models containing off-diagonal coefficients, but were less accurate.

The performance of nonlinear model predictive controllers based on a pruned second order and diagonal second order Volterra models was evaluated. The performance of these controllers was also compared to the performance obtained with a first order (linear) Volterra model. All three controllers gave equivalent results for large manipulated variable weights. However, when the controllers were tuned more aggressively, results obtained from the three controllers differed considerably. The pruned nonlinear controller performed well even when tuned aggressively while the performance of the linear controller deteriorated. For the case of disturbance rejection, the linear controller performed slightly better than the nonlinear controllers.

KEYWORDS: Volterra, nonlinear, model predictive, flotation, control

Sinopsis

Die beheer van flotasiestelsels is 'n ingewikkelde probleem omdat flotasie stelsels 'n nie-linieêre, multiveranderlike proses is met 'n noemenswaardige graad van interaksie tussen die veranderlikes. Geisoleerde PID beheerders se werkverrigting is nie bevredigend nie. Die toepassing van 'n nie-linieêre modelgebaseerde beheeralgoritme gebaseer op tweede-orde Volterra modelle is ondersoek. Volterrareeks modelle is 'n hoër orde uitbreiding van lineêre impulsresponsmodelle. Die beheeralgoritme kan ook gesien word as 'n lineêre algoritme met hoër orde korreksieterme.

'n Dinamiese model van 'n flotasiestelsel gebaseer op die fundamentele kontinuïteits vergelykings is ontwikkel. Die model response verkry verteenwoordig die kwalitatiewe verhoudings tussen die model insette en die beheerde veranderlikes. Die model het sterk nie-lineariteite getoon, insluitende asimmetriese response op simmetriese insette. Die teken van die versterking verander ook in sekere gevalle.

Hierdie dinamiese model is gebruik as die proses waarvan tweede orde Volterra modelle verkry moes word. Volledige Volterramodelle vereis baie groot datastelle, maar noemenswaardige verkleinings in die grootte van die vereiste data stelle is verkry deur van die koëffisiënte gelyk aan nul te stel. Hierdie beperkte Volterramodelle kan die proses dinamika beter as linieêre modelle voorstel. Die beperkte Volterra modelle kan byvoorbeeld asimmetriese prosesresponse asook veranderinge in die teken van die versterking voorspel. 'n Spesiale geval van beperkte Volterramodelle is diagonale tweede orde modelle, waar al die nie-diagonalekoëffisiënte (h_{ij} waar $i \neq j$) beperk is tot nul. Hierde modelle vereis minder data as gewone beperkte Volterramodelle, maar is minder akkuraat.

Die werkverrigting van nie-linieêre model gebaseerde beheerders gebaseer op beperkte tweede-orde en diagonale tweede-orde Volterramodelle is geëvalueer. Hierde beheerders is ook vergelyk met 'n lineêre beheerder gebaseer op eerste-orde Volterramodelle. Al drie die beheerders presteer ewe goed wanneer groot gemanipuleerde veranderlike gewigte gebruik word. As die beheerders egter meer aggresief ingestel is, verskil die resultate van die drie beheerders noemenswaardig. Die beperkte nie-lineêre beheerder presteer steeds goed vir aggresiewe instellings, maar die beheer van die lineêre beheerder is swakker in die geval van setpunt veranderings. Vir die geval van versteuringverwerping vaar die lineêre model effens beter.

SLEUTELWOORDE: Volterra, nie-lineêr, modelgebaseer, flotasie, beheer

Acknowledgements

I would like to thank professor Philip de Vaal for his guidance and all my loved ones for their support.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

A	Cell cross sectional area
a	Step response coefficient
As	Aspect ratio
C	Concentration
CF	Classification function
CV	Controlled variable
d	Diameter
E	Expectation operator
FS	Froth stability factor
h	Height
h_n	Volterra coefficient of order n
I	Identity matrix
J	Superficial velocity
J_f	Objective function
k	Sampling instant
k_r	Rate constant
k_w	Rate constant for the recovery of water

K_{flow}	Flow coefficient
K_{valve}	Control valve constant
L	Level
M	Mass mineral
m	Mass flowrate
MGP	Mass gangue in the pulp
MH	Model horizon
MV	Manipulated variable
MWP	Mass water in the pulp
N	Number of data points
N_s	Impeller peripheral speed
np	Number of parameters
P	Mineral floatability
Ps	Sequence period
Q	Volumetric flowrate
R	Recovery
r	Pseudo reaction rate
R_{uu}	Autocorrelation matrix
r_{uy}	Cross-correlation between input u and output y
r_{xx}	Covariance/autocorrelation of sequence x
RR	Rate of recovery
S	Surface area flux
s	Setpoint trajectory
T	Sampling interval
t	Time

T_{uuy}	Cross bi-correlation matrix
t_{uuy}	Cross bi-correlation between input u and output y
U	Input data matrix
u	Input
UH	Prediction horizon
V	Volume in cell
v	Input deviation
VH	Control horizon
W	Weight
w	Output deviation
x	Valve fraction
Y	Output data matrix
y	Output
y	Output
z	Predicted output deviation

Subscripts

b	Bubble
c	Collection zone
f	Froth zone
g	Gas
p	Pulp
s	Sauter
t	Tailings
w	water
as	Air– solid

sw	Solid– water
wa	Water – air

Greek

α	Water drainage coefficient
β	Froth parameter
ϵ	Holdup
γ	Surface tension
κ	Kurtosis
λ	Setpoint filter parameter
μ_i	i^{th} central moment
ω	Drainage parameter
ϕ	Contact angle
ρ	Density
σ	Standard deviation
σ^2	Variance
τ	Retention time
θ	Parameter vector
ξ	Input sequence deviation size
ζ	Skewness

Abbreviations

$ARMAX$	Moving average with exogenous outputs
ARX	Auto regressive with exogenous inputs
DI	Dispersion index
DMC	Dynamic matrix control
IID	Independant identically distributed

$LMPC$ Linear model predictive control

$MIMO$ Multiple input – multiple output

MPC Model predictive control

$NARMAX$ Nonlinear autoregressive moving average with exogenous inputs

$NMPC$ Nonlinear model predictive control

$PRBS$ Pseudo random binary sequence

$QDMC$ Quadratic dynamic matrix control

RBS Random binary sequence

$SISO$ Single input – single output

CHAPTER 1

Introduction

1.1 Background

Flotation is a mineral processing operation that is used to obtain a mineral rich concentrate from finely ground ore. Flotation relies on the difference in physio-chemical surface properties of particles to achieve separation. These surface properties are usually altered by the addition of flotation reagents such as collectors, activators and depressants. The desired particles are rendered hydrophobic and will tend to attach to air bubbles. These air bubbles with attached mineral particles rise to the surface where they are removed as a concentrate froth.

Flotation can either be carried out in columns or flotation cells. In flotation columns, agitation is provided by the air which is sparged into the bottom of the column. Flotation cells are mechanically agitated and the agitator is used to break the air flow into bubbles. Flotation cells are more common than flotation columns. Cells are connected in series to form flotation banks. Several banks are connected in various configurations to form a flotation circuit.

1.2 Problem statement

The control of flotation circuits is a notoriously difficult control problem (Perez-Correa et al., 1998), (Suichies et al., 2002), (Cubillos, 1998). Flotation circuits are nonlinear, multivariable processes with a significant degree of interaction. The process also has a large number of manipulated variables, but essentially two control objectives, namely recovery and concentrate grade. Recovery refers to the fraction valuable mineral entering the system that is recovered in the concentrate while grade is an indication of the purity of the concentrate. The optimum grade and recovery depend on economic objectives.

The performance of PID controllers are degraded by interaction between loops and nonlinearity. This suggests that ordinary PID control is not sufficient for the control of grade and recovery.

The application of nonlinear model predictive control to flotation circuits will be investigated. Nonlinear model predictive control require nonlinear process models. In practice, it is too complex and time consuming to obtain fundamental process models. Due to this, models are usually identified from plant data.

1.3 Method

A dynamic model of a flotation process will be developed. This model should be sufficiently detailed to include the important dynamic relationships between the manipulated and controlled variables. While it is not feasible to develop a fully fundamental model of the flotation process, equations with physical significance (as opposed to purely statistical relationships) will be used as far as possible. This model will then be treated as the plant, and model identification techniques will be used to obtain second order Volterra series models of the plant. The models obtained will be used in a nonlinear model predictive algorithm based on Volterra series models (Maner et al., 1996).

This dissertation is structured as follows. A basic overview of flotation is given in chapter 2 and the control of flotation circuits is reviewed in chapter 3. An overview of the modelling of flotation circuits is given in chapter 4, while the approach and modelling equations used to implement a flotation circuit model in *Simulink* is given in chapter 5. An overview of model identification techniques, with emphasis on the identification of second order Volterra models, is given in chapter 6. The results obtained during the identification of second order Volterra models of the flotation circuit is given in chapter 7. A background of model predictive control techniques is given in chapter 8 and the results obtained with a nonlinear model predictive controller based on the models obtained in chapter 7 is given in chapter 9. Conclusions are made in chapter 10.

CHAPTER 2

Flotation process description

A general description of flotation is given in this chapter. Aspects such as

- the preparation of ore;
- the mechanism of particle bubble attachment;
- flotation cells and
- circuit configuration ;

will be discussed. The concepts of grade and recovery, which play a very important role in the control of flotation circuits, are also introduced in this chapter.

2.1 Froth flotation

Flotation is a solid/solid separation method used to separate valuable mineral particles from undesired particles called gangue. Flotation allows relatively poor ores to be mined since the concentrate from the flotation circuit can be economically processed. The mined ore is ground in a milling circuit before it can be floated. The operation of the milling circuit usually affects the performance of the flotation circuit and acts as a major source of disturbances.

2.2 Milling circuit

Ore is first crushed and then milled under water (Sutherland et al., 1955: 13). From the grinding circuit the pulp is fed to the classifiers, where coarse and fine particles are separated. The particles that are too coarse to be floated are returned to the grinding circuit, while the fine particles are sent to the flotation circuit. The separation in the classifiers is usually based on the weight of the particles, which means that denser particles are usually more finely ground (Sutherland et al., 1955: 13). Ore breakage often occurs along mineral boundaries, yielding particles which contain single minerals. However, some particles will contain more than one mineral (Lynch et al., 1981: 3). Composite particles complicate flotation: if the particles are sent to the tailings, the valuable minerals will be lost and if they are sent to the concentrate, the concentrate will be less pure (Lynch et al., 1981: 3), (Sutherland et al., 1955: 14). Very fine particles also presents problems during flotation, since the conditions for particle – bubble attachment deteriorates (Wills, 1997: 285).

2.3 Particle – bubble attachment

Flotation occurs if a mineral particle attaches to a bubble and is taken to the surface by the buoyancy of the bubble. For particle – bubble attachment to take place, an air – solid interface must be created and a water – air and a water – solid interface must be destroyed. This requires that the sum of the surface tensions of the air – water and solid – water interfaces must be greater than the surface tension of the air – solid interface. This requirement is shown in equation 2.1 (Sutherland et al., 1955: 28). The surface tensions also give an indication of the work required per unit area for the system to change, shown in equation 2.2.

$$\gamma_{as} \leq \gamma_{wa} + \gamma_{sw} \tag{2.1}$$

$$\text{Work} = \gamma_{wa} + \gamma_{sw} - \gamma_{as} \tag{2.2}$$

The air–solid and solid–water interface surface tensions cannot be measured. However, these quantities can be eliminated from equation 2.2 by introducing the contact angle (as indicated in figure 2.1) between the solid and the bubble. The forces in the plane of the solid surface should sum to zero. This requires that $\gamma_{as} = \gamma_{wa}\cos\phi + \gamma_{sw}$. Using this relationship, the work done on the system becomes

$$\text{Work} = \gamma_{wa}\,(1 - \cos\phi) \tag{2.3}$$

Equation 2.3 is also a measure of the work required to break the air solid interface and gives an indication of the strength of the adhesion between the mineral particle and the bubble. If the contact angle is zero, there is no tendency for sticking and if the angle is 90°, the tendency of adhesion reaches the maximum. Both γ_{wa} and the contact angle can be measured experimentally and has been used extensively in the theory of flotation and is a parameter which describes the floatability of a mineral.

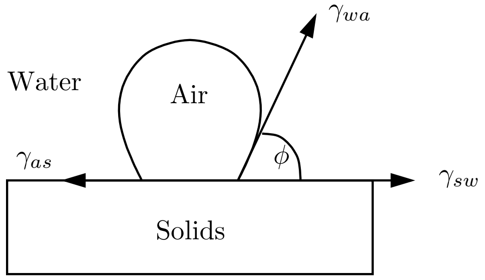

Figure 2.1: The contact angle between the air bubble and the mineral particle and the forces acting on the bubble (Sutherland et al., 1955)

A reagent called a collector is usually added to allow the particle – bubble attachment to take place (Sutherland et al., 1955: 26). The use of flotation reagents is discussed in section 2.4.

2.4 Flotation reagents

Most minerals will not float without some surface modification. Also, the bubbles formed in an air – water – mineral system will usually not form a stable froth capable of supporting mineral particles. Flotation reagents are usually added to modify the system. Flotation reagents can be divided into the following groups.

2.4.1 Frothers

A frother is a substance which is added to water to allow a stable froth to be formed (Sutherland et al., 1955: 9). Flotation cannot occur without a stable froth to carry the valuable mineral particles from the cell.

2.4.2 Collectors

A collector is a substance which enables a mineral to be held at the air– water interface (Sutherland et al., 1955: 9). Minerals can be classified into polar and non-polar minerals. The surfaces of non-polar minerals do not readily attach to water dipoles and are consequently hydrophobic. It is possible to float these minerals without the aid of collectors, but collectors are usually added to improve performance (Wills, 1997: 260). Polar minerals interact strongly with water and are hydrophilic. The surface of these minerals have to be modified before they can be floated.

2.4.3 Regulators

Regulators are reagents that are added to regulate the flotation process (Wills, 1997: 259). These reagents can be subdivided into activators and depressants.

Activator A collector makes a particular mineral floatable in the presence of a collector which would not have had an effect in the absence of the activator (Sutherland et al., 1955: 9).

Depressants A depressant prevents a collector from functioning for a particular mineral (Sutherland et al., 1955: 9).

2.5 Flotation circuits

Flotation is similar to fractional distillation in that the desired purity is not achieved in one separation step (Sutherland et al., 1955: 19). Flotation cells are usually connected in series to form flotation banks. These banks are connected in various configurations to form flotation circuits.

2.5.1 Flotation cells

Figure 2.2 shows a basic flotation cell. Flotation cells are mechanically agitated. The impeller breaks the air stream into bubbles and keeps the mineral particles suspended. Flotation can also be carried out in flotation columns. a Flotation column is shown in figure 2.3. Flotation columns are not mechanically agitated, but air is sparged into the bottom of the column. Wash water is introduced at the top of the flotation column to remove entrained material. Flotation cells are much more common than flotation columns and flotation columns will not be discussed further.

2.5.2 Cell banks

Flotation cells are arranged in series to form cell banks. Pulp (milled ore and water) is fed to the first cell in the bank. Some of the valuable mineral is removed as froth, and the pulp is sent to the next cell, where more mineral is removed (Wills, 1997: 281). The bank may consist of a tank partially separated to form cells, or the bank may consist of individual cells. In some cases, the whole bank is a single tank with agitators in series. Figure 2.4 shows the schematic representation commonly used for flotation banks.

2.5.3 Flotation circuits

Banks of flotation cells are connected to form flotation circuits. A simple flotation circuit is shown in figure 2.5. Most of the valuable mineral is floated in the first few cells (the roughers). The last few cells are called the scavengers. The aim of these cells are to recover the more weakly floating material. This circuit configuration is only successful if the gangue (waste) material is relatively unfloatable (Wills, 1997: 283). It is common to

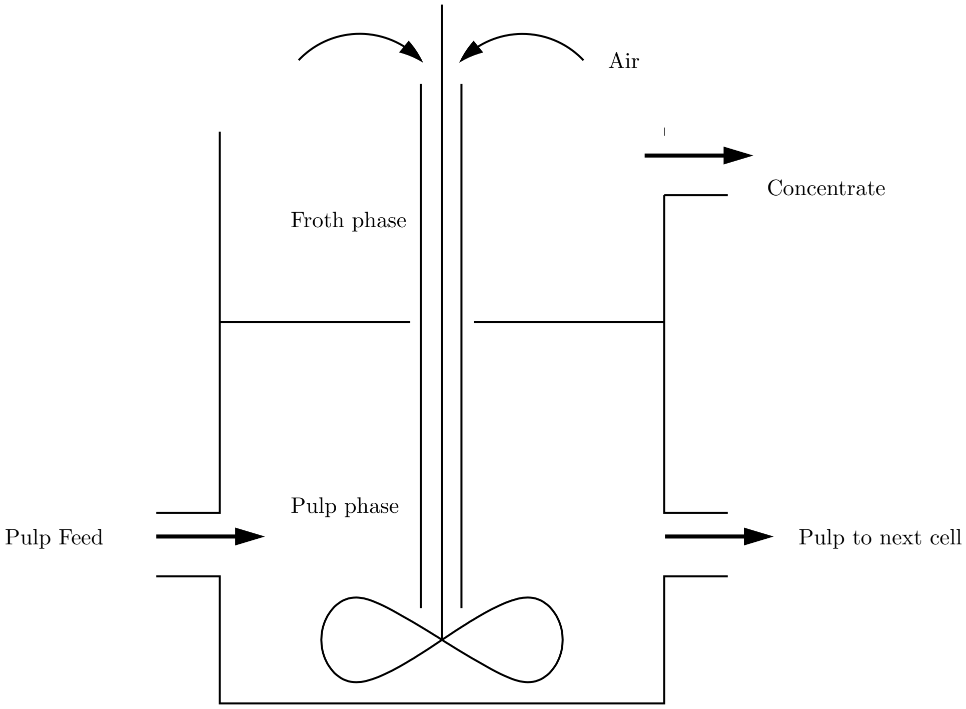

Figure 2.2: Basic flotation cell design

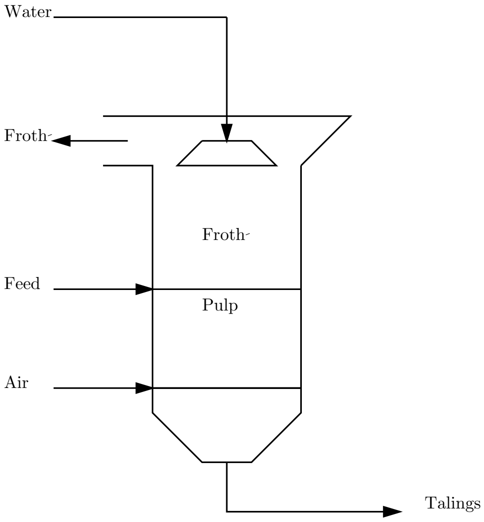

Figure 2.3: A flotation column (Persechini et al., 2000)

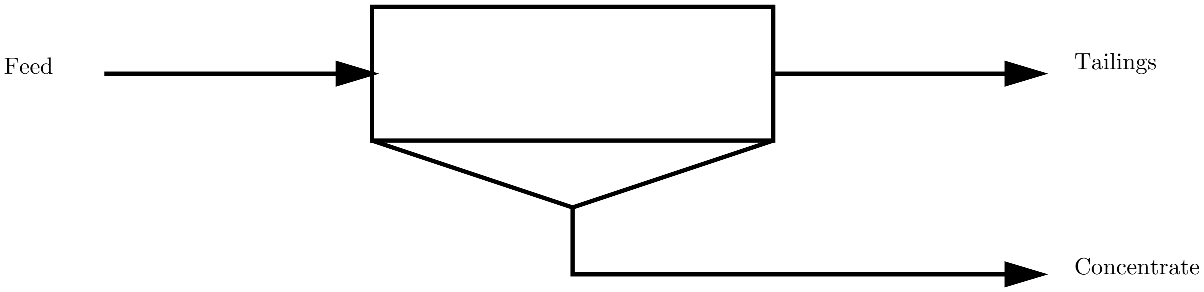

Figure 2.4: Schematic representation of a flotation bank

refloat the concentrate from the roughers to obtain a concentrate with a high grade. This circuit configuration is shown in figure 2.6. Many other possible circuit configurations exist.

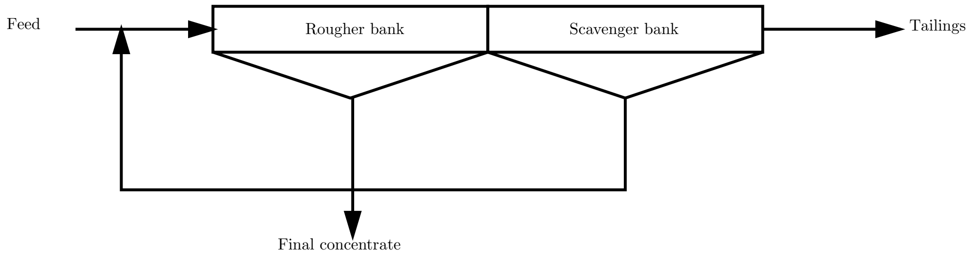

Figure 2.5: A simple flotation circuit with rougher and scavenger banks (Wills, 1997: 283)

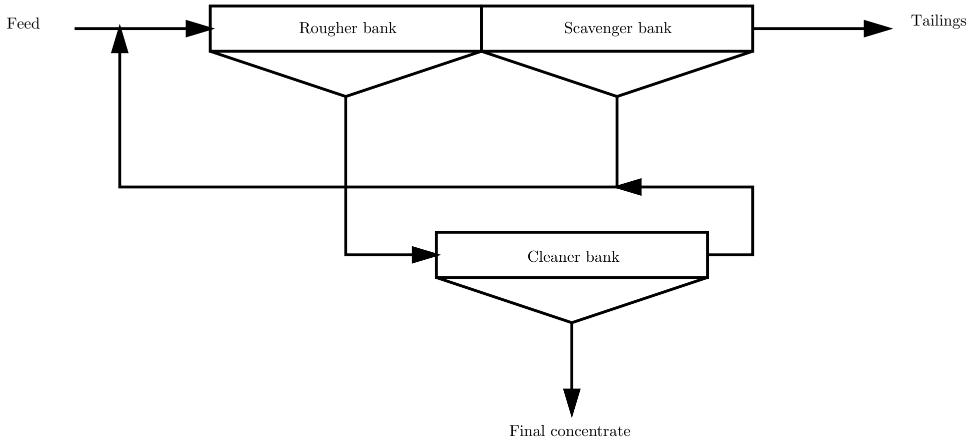

Figure 2.6: A flotation circuit with rougher, cleaner and scavenger banks (Wills, 1997: 283)

2.6 Recovery and Grade

Recovery can be defined as the fraction of minerals present in the feed that is recovered in the concentrate. Grade is an indication of the purity of the concentrate, usually expressed as the concentration of valuable mineral in the concentrate (Sutherland et al., 1955: 20).

Flotation circuits are operated so that a concentrate of a sufficient grade is obtained while the recovery of minerals is also acceptable. Some trade-off between recovery and grade has to be made. Qualitative recovery–time and recovery–concentrate grade curves are shown in figures 2.7 and 2.8. There is no unique separation efficiency for an ore and the grade recovery curve will depend on the operating conditions in the circuit.

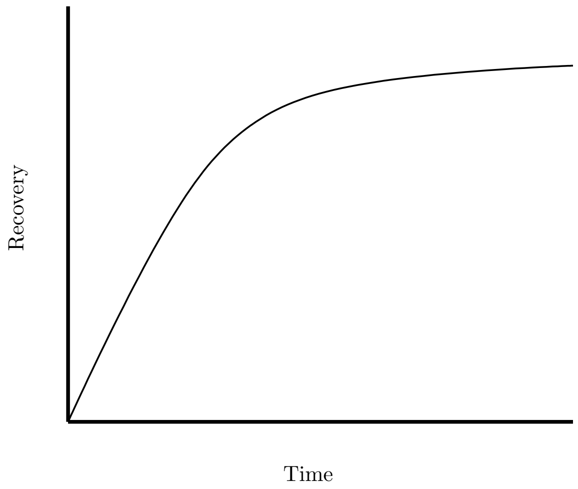

Figure 2.7: The recovery of mineral versus time (Wills, 1997)

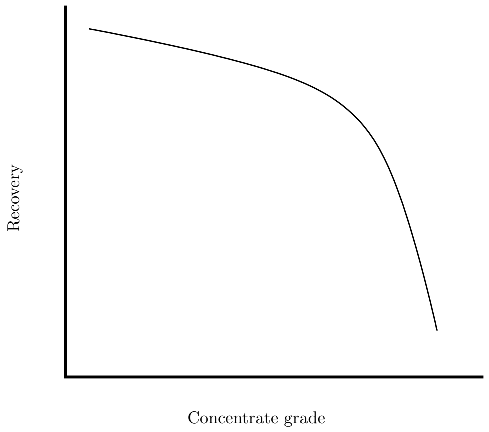

Figure 2.8: The decrease in concentrate grade for higher recoveries (Wills, 1997)

CHAPTER 3

Control of flotation circuits

This chapter discusses important concepts in the control of flotation circuits, with the emphasis on the important variables in flotation. Typical control objectives are discussed in section 3.1.1, while typical manipulated variables are discussed in section 3.1.2. The application of advanced control to flotation circuits is discussed in section 3.2.

3.1 Control Variables

3.1.1 Control objectives

The concentrate grade and the recovery of valuable mineral are the two variables that indicate the actual circuit performance. Ideally, a flotation circuit would give a concentrate containing all the valuable mineral that entered the circuit in the feed, but none of the gangue. However, this separation in not attainable in practice and a trade-off between grade and recovery has to be made.

For a given feed composition and flow, a band of possible operating points can be defined on the grade–recovery plane (see figure 3.1) (Lynch et al., 1981: 15). The aim of a control strategy for a flotation circuit should be to operate the plant on a particular point on the upper bound of this area. The specific operating point is determined by economic factors (Hodouin et al., 2001).

3.1.2 Manipulated variables

The most common manipulated variables that are available for control of recovery and grade on flotation plants are (Perez-Correa et al., 1998), (Lynch et al., 1981: 17):

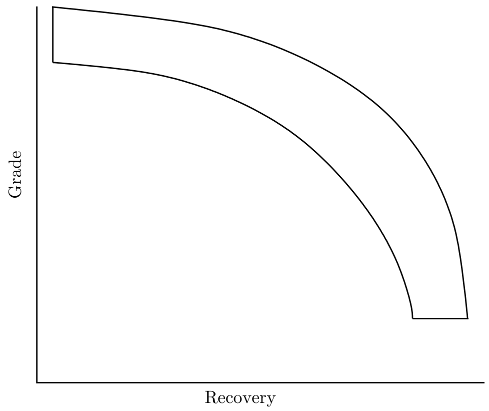

Figure 3.1: The band of possible grade-recovery operating points (Lynch et al., 1981: 15)

- the addition rates of chemicals such as collectors and frothers;
- the air flow rate to the cells and
- the pulp level setpoints in the cells.

Both the recovery and grade are affected by the pulp levels in the cells. Higher pulp levels lead to longer pulp residence times in the cell, which lead to increased mineral recovery. However, higher pulp levels imply lower froth heights, since the froth height is approximately equal to the difference between the cell height and the pulp level (Wills, 1997: 313). Lower froth heights reduce the froth residence time, which leads to poorer gangue drainage. This reduces the concentrate grade (Jamsa-Jounela et al., 2003). The pulp level setpoints are often used as manipulated variables by advanced control strategies, but good control of the levels are sometimes difficult to achieve. The control of the pulp levels is further discussed in section 3.1.4.

Increases in collector addition initially lead to increased valuable mineral recovery, but tend to reach a point where further increases in collector addition no longer increase the recovery, while the recovery of gangue is increased (Wills, 1997: 312). The use of frother addition as a manipulated variable is complicated, since the action of the frother is a complex function of factors such as the water chemistry. Excessive dosage at one part of the circuit may also cause problems downstream (Wills, 1997: 313).

The air flow rate to the cells is a very important manipulated variable. The effects of air flow rate tend to be faster than adjustments in pulp levels and has no residual effects if used in excess (Wills, 1997: 313). Increases in air flow rate lead to reduced froth residence times, which decreases the time gangue particles has to drain from the froth. Higher air flow rates also lead to an increase in bubble surface area flux, which increase the collection of particles in the pulp phase.

3.1.3 Disturbances

Common disturbances in a flotation circuit include changes in the feed rate to the circuit as well as the pulp density, particle size distribution and composition of the feed (Lynch et al., 1981: 16).

3.1.4 Control of pulp levels

Level control of flotation cells can be a very complex task, especially in circuits with a series of interacting levels (Stenlund & Medvedev, 2002) (Kampjarvi & Jamsa-Jounela, 2003). Flotation cells are often controlled with isolated PI controllers that do not handle the interactions from other cells well (Kampjarvi & Jamsa-Jounela, 2003). Stabilisation of the pulp levels in the cells can give improved recoveries (Stenlund & Medvedev, 2002). Good level control is also very important if the pulp level setpoints are to be used in an advanced control system.

Several strategies have been used to improve the control of pulp levels. Stenlund & Medvedev (2002) used both PI controllers with decouplers as well as a linear quadratic optimal controller to control the levels in a cascade coupled flotation circuit. A commercial multivariable flotation control package has been implemented on several plants. This controller controls the levels in the vessels based on the entire upstream inventory.

3.2 Application of advanced control to flotation circuits

Flotation circuits are usually difficult to control. The process tends to be highly nonlinear and has strong interactions between variables (Perez-Correa et al., 1998). This limits the performance attainable by SISO PID controllers and makes the application of advanced control techniques attractive. Advanced control is also attractive from an economic point of view. Due to the high throughput of circuits, even increases in recovery as small as 0.5 % can be economically significant (Ferreira & Loveday, 2000). Various advanced control techniques have been applied to flotation circuits.

Thornton (1991) applied a self-tuning minimum variance adaptive controller to the rougher bank of flotation circuit. The controlled variable was the composition of tailings from the bank and the manipulated variable was the collector flowrate to the bank. The control scheme included a cautiously tuned PID controller in parallel with the minimum variance controller. The final control action is the weighted average of the PID loop and the adaptive controller, with the weight determined by the model error at a particular time. The block diagram for this scheme is shown in figure 3.2. This configuration has the advantage that the controller can adapt to changes caused by factors such as changes in the composition of the ore to be processed.

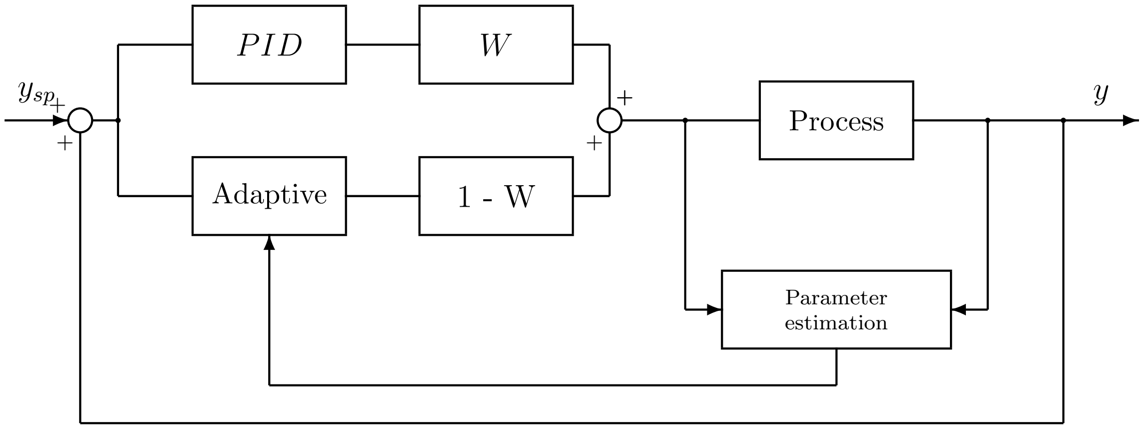

Figure 3.2: Block diagram for a cautious adaptive controller

Suichies et al. (2002) implemented a SISO Generalised predictive controller to control the final grade of an industrial flotation circuit. The ratio of collector addition to feed was used as the manipulated variable by this controller. An ARX model was identified with pseudo random binary sequences. Eight to ten hours were typically required for the tests, which were carried out with the process being controlled by detuned PID loops. This identification procedure has to be repeated from time to time as the models become inaccurate. This controller was able to reduce the variance of the grade significantly.

Both dynamic matrix control (DMC) and quadratic dynamic control (QDMC) (see chapter 8) strategies have been tested on a simulated flotation circuit (Perez-Correa et al., 1998), (Perez-Correa et al., 1995). The levels in the cell banks were used as manipulated variables, and the controllers aimed to keep the concentrate grade above a minimum and the tail grade above a maximum. The QDMC controller performed better than the DMC controller. These control strategies use all the manipulated variables in unison to achieve the control objectives.

The control strategies discussed above rely on linear process models obtained from plant data. Due to the nonlinearity of the flotation process, the accuracy of these models are limited which in turn limits the performance of the controller. Advanced control strategies based on nonlinear process models have also been applied to flotation circuits.

A form of nonlinear predictive control has been applied to a rougher flotation unit (Desbiens et al., 1998). The manipulated variables for this controller were the air flowrate and the rate of collector addition and the controlled variables were the mineral recovery and concentrate grade. Three linear local models were identified for the relationship between the collector flowrate and the concentrate grade. The parameters of the nonlinear model was found by interpolating between the parameters of the three local linear models. This approach requires more plant data than a controller based on a single process model, but gives improved control performance.

Perez-Correa et al. (1998) also applied rule based control strategies to a simulated circuit. These strategies tended to saturate the manipulated variables. The controller constantly aimed to move the process to a better operating point by moving the controller from one operating zone to another. In each operating zone, either recovery or grade was optimised, but the other controlled variable was also affected. This led to the controller causing and correcting deviations in grade and recovery until the manipulated variables were saturated.

Cubillos (1998) applied model based control utilising a hybrid neural network model to a flotation bank. Both the grade and recovery were controlled and the frother and collector flowrates as well as the pulp levels were used as manipulated variables. The model was based on fundamental mass and energy balances with a neural network used to estimate the model parameters. This approach requires less data than a pure hybrid neural network, but requires process knowledge. The model was also adapted on-line at each sampling interval.

3.3 Evaluation of control strategies

Dynamic models of processes may be used to test new control strategies. The use of dynamic models for this purpose has several advantages. New strategies can be tested in a risk free environment and results can be obtained faster than real time. Disturbances can also be introduced as desired. The modelling of flotation circuits will be discussed in the following two chapters.

CHAPTER 4

Modelling of flotation circuits

This chapter provides an overview of the literature of flotation circuit modelling. The techniques used for modelling the important pulp and froth phase phenomena are discussed. The emphasis is placed on two-phase models, where the processes occurring in both the pulp and the froth phase are included in the model.

4.1 Flotation model classification

There are various approaches to modelling flotation circuits. A classification of the most common types of flotation models is shown in figure 4.1. In micro-scale models, flotation is modelled by identifying all the sub-processes and using chemical and physical relationships to predict the process behaviour. This approach is difficult due to the complexity of the system and the interactions between chemical and physical parameters (Polat & Chander, 2000).

Macro-scale models relate the overall response to the operating parameters through a set of mathematical equations. Macro-scale models can be divided into phenomenological models and statistical models. In statistical models, the flotation performance is related to variables through mathematical equations without physical significance. Phenomenological models consist of equations which describe the physics and chemistry of the process. Phenomenological models can use either kinetic, probabilistic or population balance models to describe the particle-bubble behaviour. Probabilistic models are based on the relative occurrence of collision, adhesion and detachment and can link micro and macro-scale models. Kinetic models are similar in form to those used to describe chemical reactions. The rate of flotation is assumed to be related to the concentration of mineral particles in the flotation cell. Population balance models attempt to describe the flotation behaviour by predicting the number of bubbles in the cell as well as their sizes.

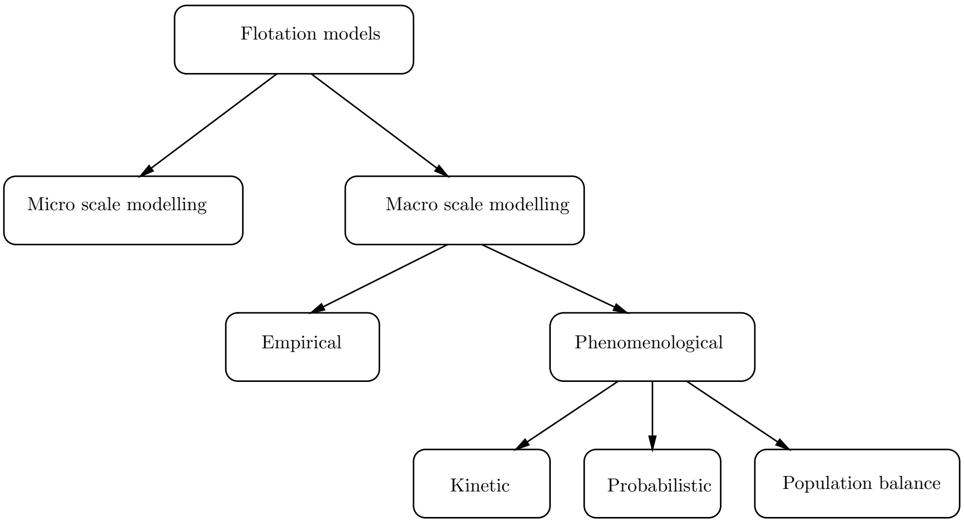

Figure 4.1: Classification of flotation models (Polat & Chander, 2000)

The flotation process can also be described with varying degrees of complexity. Some flotation models only model the levels of pulp in flotation cells and banks. These models are used to test level control strategies in flotation circuits. Other models also take the recovery processes which occur in the cells into account. The pulp and froth can be considered separately, or the system can be modelled as a single, well mixed phase.

4.2 Modelling of levels in the flotation circuit

On an industrial scale, flotation is performed in cells connected in series. Pulp is fed to the first cell and froth (concentrate) is collected. The remaining pulp (tailings) flows to the next cell. The magnitude of the flow to the next cell depends on the pressure difference between cells, the position of the control valves and the viscosity and density of the pulp (Kampjarvi & Jamsa-Jounela, 2003). The pressure difference between the cells depends on the height difference between cells.

4.3 Single phase models

In this type of model, the pulp and froth are described by a single phase of pulp perfectly mixed with froth (Casali et al., 2002). The mathematical complexity of the model is significantly reduced by this assumption (Casali et al., 2002). However, as discussed in section 4.4, these models are less accurate than two phase models.

4.4 Two phase models

While single phase models assume one perfectly mixed phase, many authors (Hemphill & Loveday, 2003), (Lynch et al., 1981), (Vera et al., 2002) claim that the froth phase must also be considered to adequately model flotation. Two phase models include the behaviour of both the froth and the pulp phases.

There are a number of transfer processes between the froth and pulp phases. The transport paths of material in a flotation cell are shown in figure 4.2. The pulp phase is often called the collection zone, since valuable particles are collected through selective particle bubble attachment. Material (both gangue and valuable mineral) may also become entrained between bubbles and enter the froth phase in this way (Savassi et al., 1998) (Mathe et al., 1998).

Several things may happen to a particle attached to a bubble once it enters the froth phase. It may detach from the bubble, through the bubble breaking, from being washed away or being displaced by another particle. Once the particle has detached, the particle may either re-attach to another bubble, or it may return to the pulp (Hemphill & Loveday, 2003). A part of the froth may also collapse, returning the attached particles to the pulp. Entrained particles (which are not attached to a bubble) may either be carried into the concentrate, or may drain back into the pulp. All these processes contribute to the complex behaviour of the flotation process and must be considered in the flotation model.

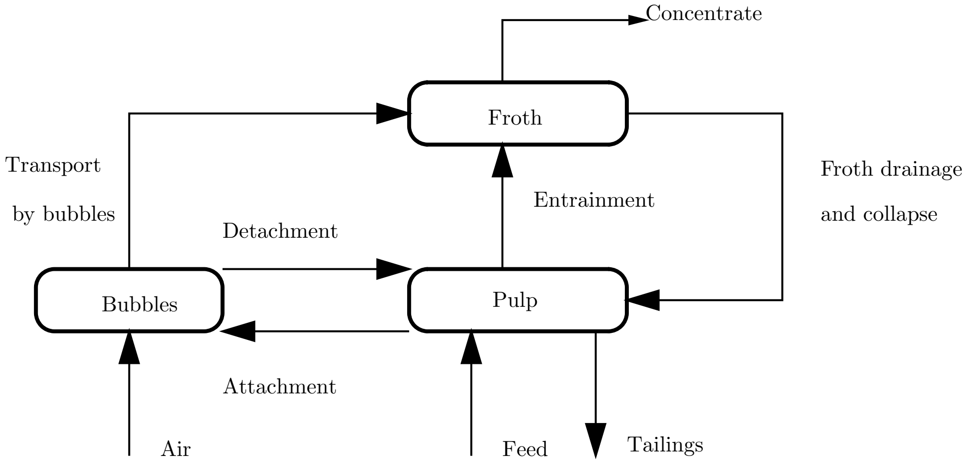

Figure 4.2: Transport paths of material in a flotation cell (Subrahmanyam & Forssberg, 1988)

4.4.1 Froth zone recovery

As discussed above, only a fraction of the particles that enter the froth are collected in the concentrate. The froth zone recovery factor is used to describe the metallurgical performance of the froth phase (Vera et al., 2002). The transfer between the pulp and froth phase (excluding entrainment) is shown in figure 4.3. The effects of entrainment will be discussed in section 4.4.9.

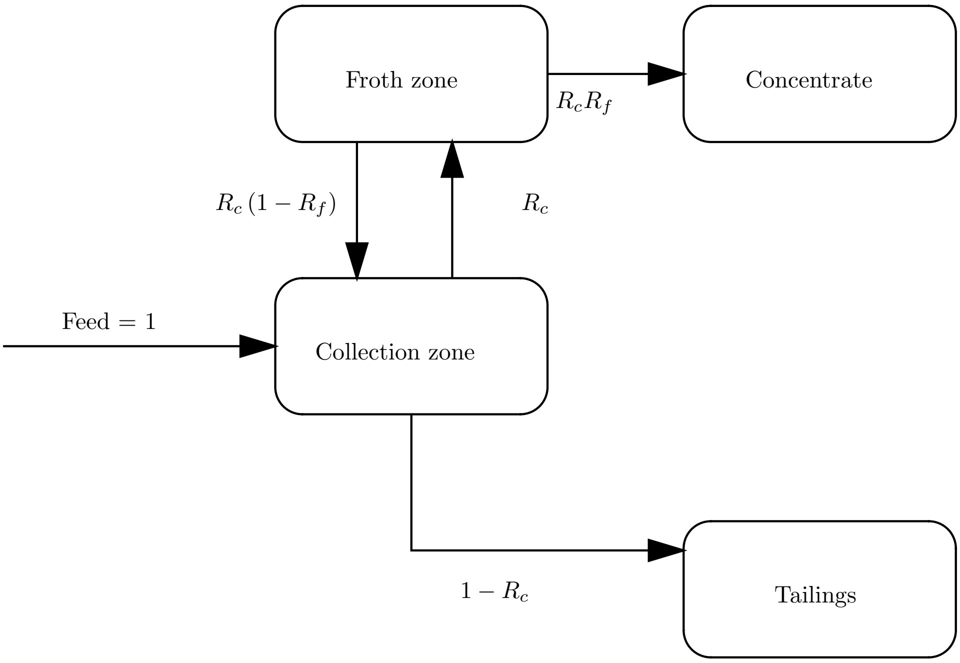

Figure 4.3: The transfer of material between the pulp and froth phases (Vera et al., 2002)

The fraction of material that enters the froth attached to bubbles is the collection zone recovery (R_c). A fraction of these particles returns to the pulp phase. This fraction is called the drop-back and the remaining fraction is the froth zone recovery (R_f) (Finch & Dobby, 1990: 89). In other words, the froth zone recovery is the fraction of particles entering the froth attached to the bubbles that is recovered in the concentrate.

The collection zone recovery usually varies between 60 % to 90 % while the froth zone recovery varies from 10 % and 90 %. A low froth zone recovery can have a significant detrimental effect on the performance of a cell (Vera et al., 2002).

Gorain et al. (1998a) investigated the relationship between froth residence time (τ_f) and the overall rate constant (see section 4.4.7). An exponential relationship of the form given in equation 4.1 was found.

$$R_f = e^{\left(-\beta\tau_f\right)} \tag{4.1}$$

Essentially the same exponential relationship between the froth retention time and the froth zone recovery has been observed by (Vera et al., 2002). Vera et al. (2002) states that this relationship is essentially independent of cell size, cell mechanism, and cell operating conditions. The exponential function (equation 4.1) describes the number of particles which remain attached to bubbles in the froth phase. β is a parameter related to the condition of the froth (Vera et al., 2002).

Particles that have become detached from bubbles may drain back into the pulp or be recovered in the concentrate in the water between bubbles. This mechanism is similar to entrainment, except that only the particles which have initially entered the froth attached to bubbles are considered here. The relationship given in equation 4.2 represents the fraction of particles recovered through entrainment (Vera et al., 2002).

$$\text{fraction entrained} = \frac{1}{1+\omega\tau_f} \tag{4.2}$$

A relationship for the total froth zone recovery can be obtained by combining equations 4.1 and 4.2. The fraction of particles that have detached from bubbles is given by $(1 - e^{(-\beta\tau_f)})$. The fraction of detached particles that are recovered through entrainment is obtained by multiplying with equation 4.2. The resulting relationship is shown in equation 4.3, which is an refinement of equation 4.1.

$$R_f = e^{(-\beta\tau_f)} + \left(1 - e^{(-\beta\tau_f)}\right)\left(\frac{1}{1+\omega\tau_f}\right) \tag{4.3}$$

The froth residence time in the cell affects the froth zone recovery. Various definitions of froth residence time are discussed in section 4.4.8. The relationships given in this section was defined for the froth residence time based on the flow of concentrate, but since the concentrate flowrate cannot be predicted independently, the froth retention time based on the air flowrate may be used (Vera et al., 2002). Hemphill & Loveday (2003) used an effective froth residence time to describe the behaviour of the froth. This parameter was obtained by regression and incorporated effects such as the froth residence time, aeration rate and froth stability.

4.4.2 Hydrodynamic conditions

Flotation equipment factors, such as cell and impeller design, impeller speed and air flow influence the performance of a flotation circuit through the hydrodynamic conditions created by these factors. The hydrodynamic conditions in a flotation cell include factors such as the flow regime, gas dispersion and particle bubble interaction (Finch et al., 2000). Gas dispersion properties include the bubble size, gas flow rate, gas holdup and bubble surface area flux.

As will be discussed in section 4.5.2, the rate at which particles are collected in the pulp phase (collection zone) can be described by a first order rate equation. The pseudo flotation rate constant ($k_{r,c}$), which describes the rate at which particles leave the pulp phase, depends on the hydrodynamic conditions in the cell.

4.4.3 Bubble size

The bubble size in a flotation cell depends on aspects such as impeller type and speed, frother dosage, pulp viscosity and solids concentration. The bubble size distribution tends to vary throughout the cell, with the smallest bubbles closest to the impeller. Bubble size distribution alone does not adequately describe the gas dispersion properties in the flotation cell (Gorain et al., 1995).Gorain et al. (1997) plotted the the flotation rate constant against the average Sauter bubble diameter for several systems. No good correlation between the flotation rate constant and Sauter mean diameter was found. This is not surprising, since bubble size on its own does not describe the volume of gas in the cell.

4.4.4 Gas holdup

When a gas is introduced into a liquid or slurry, a volume of liquid is displaced. The volumetric fraction displaced is the gas holdup (Finch & Dobby, 1990). Gas holdup increases with increasing gas flowrate and decrease in bubble size. Gorain et al. (1997) found no significant relationship between gas holdup and flotation rate constant. This is also not unreasonable, since a system with few large bubbles and another system with many small bubbles may have the same gas holdup, but the flotation behaviour of these systems will be significantly different.

Bubble surface area flux and gas holdup is related, with the one property increasing with the other. This is a consequence of their dependence on gas flowrate and bubble size. Finch et al. (2000) found a linear relationship between gas holdup and bubble surface area flux. This relationship did not hold for the data reported by (Gorain et al., 1997). This discrepancy is likely to be caused by the usage of overall gas holdup measurements by Finch et al. (2000), while Gorain et al. (1997) used the average of local holdup measurements. This relationship is also not likely to hold for cells with a high pulp recirculation, where the gas holdup may be high without the corresponding high bubble surface area flux (see section 4.4.6). Gas holdup is, however, easier to measure than bubble surface area flux.

4.4.5 Superficial gas velocity

The superficial gas velocity is defined as the volumetric flowrate of air to the cell divided by the cell surface area, as shown in equation 4.4.

$$J_g = \frac{Q_g}{A} \tag{4.4}$$

The superficial gas velocity may also be measured. There is often a difference between the measured superficial gas velocity and the gas velocity that is calculated from equation 4.4. Gorain et al. (1996) defined a dispersion index (equation 4.5) based on the difference between a calculated superficial gas velocity, as defined in equation 4.4 and the measured superficial gas velocity. The superficial gas velocity was measured at six points in the cell. Two of these measurements were close to the impeller. The values of $J_{g,\text{calculated}}$ and $J_{g,\text{measured}}$ differ because the measured velocity at six points do not describe the entire surface and the measurements close to the impellers contribute too much compared to those further away from the impeller. Since the air tends to rise close to the impeller when dispersion is poor, relatively large values of $J_{g,\text{measured}}$ indicate poor dispersion. Dispersion index values of close to 100 indicate good dispersion. This index does not have a fundamental physical significance, but was used as an indication of the relative performance of the impellers studied.

$$DI = 100 - \frac{J_{g,\text{measured}} - J_{g,\text{calculated}}}{J_{g,\text{measured}}} \times 100 \tag{4.5}$$

The relationship between the flotation rate constant ($k_{r,c}$) and superficial gas velocity is much stronger than the relationships between flotation rate constant and gas holdup 4.4.4 or bubble size 4.4.3 (Gorain et al., 1997). However, the $k_{r,c} - J_g$ relationship deteriorates at high superficial gas velocities.

4.4.6 Bubble surface area flux

The bubble surface area flux (S_b) is defined in equation 4.6. The Sauter mean diameter used in this calculation is given in equation 4.7.

$$S_b = \frac{6J_g}{d_s} \tag{4.6}$$

$$d_s = \frac{\sum_{i=1}^{i=n} d_i^3}{\sum_{i=1}^{i=n} d_i^2} \tag{4.7}$$

As discussed above, Gorain et al. (1997) conducted studies in which the effect of bubble size, gas holdup and superficial gas velocity on the flotation rate constant was investigated. None of these properties could individually be related to the flotation rate constant. However, when these properties were combined in the bubble surface area flux in the cell, it was found that a linear relationship between bubble surface area flux and flotation rate exists. This relationship was derived for data obtained with several different cell designs.

The $k_{r,c}$ vs S_b relationship can be expressed as shown in equation 4.8. P is the mineral floatability which is dependent on the particle composition and surface conditions.

$$k_{r,c} = PS_b \tag{4.8}$$

As discussed in section 4.4.5, poorly dispersed systems tend to have high superficial velocities close to the impeller. The experimental method used by Gorain et al. (1996)tended to give too high average superficial velocity measurements for poorly dispersed systems. Heiskanen (2000) observed that this effect leads to unrealistically high bubble surface area fluxes for poorly dispersed systems. However, when Gorain et al. (1997) evaluated the $k_{r,c} - S_b$ relationship, the superficial velocity measurements close to the impeller were neglected if undispersed air flow conditions were observed near the impeller. Heiskanen (2000) also observed that Gorain et al. (1997) obtained higher flotation rates for systems with a low dispersion index. Industrial experience shows that poor flotation performance is obtained if the system is poorly dispersed. However, the results (shown in figure 4.4) are not unreasonable. The dispersion index tends to be lower for higher gas flows through the cell. The bubble surface area flux was higher for these systems, which makes higher flotation rates possible even if the system is less dispersed. The dispersion index also does not describe the actual dispersion conditions in all cases.

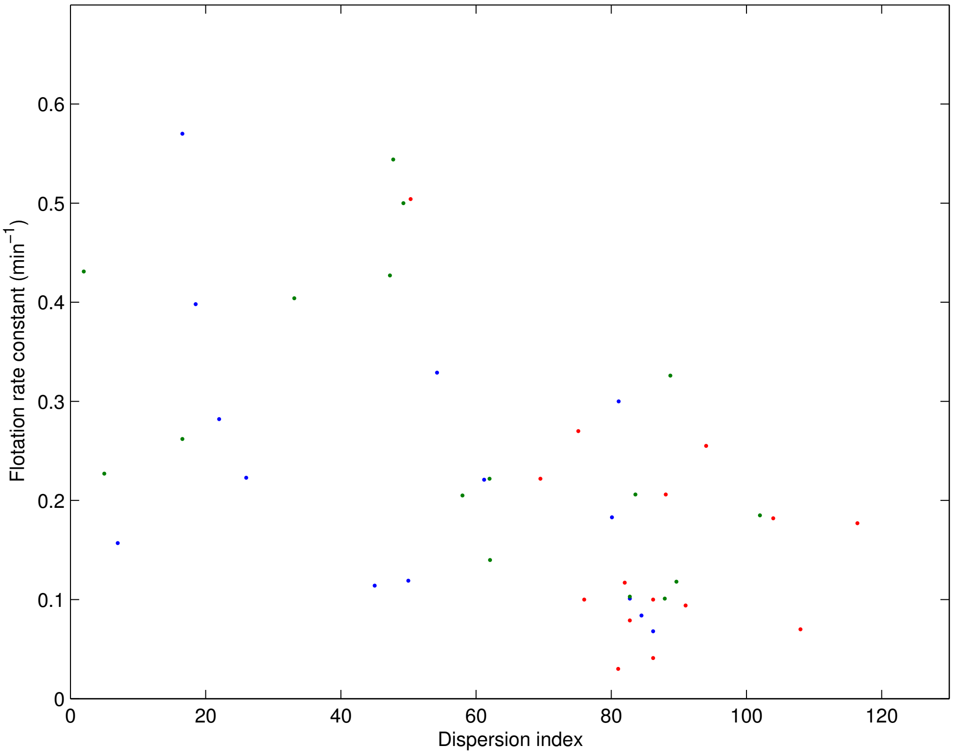

Figure 4.4: The flotation rate constant at various values of the dispersion index

Estimation of bubble surface area flux Bubble surface area flux is a more fundamental parameter to describe flotation than properties like gas holdup, but it is more difficult to measure, since measurements of the bubble sizes in the cell are required (Finch et al., 2000).

In a mechanically agitated flotation cell, the size of the bubbles depends on the impeller design and the impeller speed as well as the air flowrate (Gorain et al., 1999). The frother concentration in the cell also affects the bubble size, but in the range of frother concentrations used in industrial cells, the effect of variations in frother concentration has a minimal influence on bubble size. The mean bubble size tends to decrease with increases in impeller speeds and increases with increases in air flow rate. While aspects such as particle size and slurry density also influence the bubble size and so influence the bubble surface area flux, Gorain et al. (1999) found that these effects were not significant for the ranges of conditions found in industrial cells.

If impeller peripheral speed instead of impeller speed is used, and if air flow rate per unit cell cross-sectional area is used, S_b can be represented independently of cell size and impeller design (Gorain et al., 1999). The impeller can also be characterised with the impeller aspect ratio (As), shown in equation 4.9. The impeller diameter is measured from impeller tip to tip, and the impeller height is the vertical height of the impeller blades Gorain et al. (1999).

$$As = \frac{\text{Impeller diameter}}{\text{Impeller height}} \tag{4.9}$$

To summarise, the surface bubble flux can be assumed to be a function of the peripheral impeller speed, air flow rate per cross sectional area, the impeller aspect ratio and the particle size in the cell. Gorain et al. (1999) used the 80% passing feed (d_{80}) size to represent the particle sizes in the cell. The model shown in equation 4.10 was presented. a, b, c, d and e are the parameters for the model. Gorain et al. (1999) obtained parameters for equation 4.10 with data consisting of S_b values calculated from the superficial gas velocity and measured bubble size. The data included data for a wide variety of ores and cell designs. Data obtained from rougher, cleaner and scavenger circuits were used. The units and parameter values used by Gorain et al. (1999) are shown in appendix A.

$$S_b = aN_s^b\left(\frac{Q_g}{A}\right)^c As^d d_{80}^e \tag{4.10}$$

4.4.7 Overall rate constant

The previous paragraphs described the characterisation of the relationship between the rate at which particles are collected in the pulp phase and enter the froth phase. The rate at which particles are collected in the froth can also be described by a first order kinetic relationship. In this case the rate constant in the rate equation is the overall rate constant, which includes froth effects.

It was found (Gorain et al., 1998b) that the relationship between the overall rate constant deviates from linearity as the froth depth increases. k_r is greater for lower froth depths, due to a higher froth zone recovery. For shallow froth depths, the overall rate constant is essentially equal to the collection zone rate constant $k_{r,c}$. At intermediate froth depths, the k_r values are smaller than for lower froth depths with the same S_b. Further increases in froth depth lead to further decreases in k_r. This suggests that equation 4.8 can be modified to include the froth recovery factor as shown in equation 4.11. This is a non-linear relationship due to the non-linearities contained in R_f. Gorain et al. (1998b) originally proposed the relationship for the froth recovery factor shown in equation 4.12. S_b^* is the threshold bubble surface area flux, defined as the minimum bubble surface area flux where concentrate overflows into the launder. Vera et al. (2002) subsequently investigated the froth zone recovery and formulated a relationship for R_f by describing the particle bubble detachment and drainage in the froth (see section 4.4.1).

$$k_r = PS_bR_f \tag{4.11}$$

$$R_f = \frac{\sigma(S_b - S_b^*)}{1 + \sigma(S_b - S_b^*)} \tag{4.12}$$

Deglon et al. (1999) investigated the $k_r - S_b$ relationship by combining a bubble population model and an attachment-detachment model in simulations. These models predict a near-linear region up to about 50 s^{-1}, after which the $k_r - S_b$ curve goes through a maximum. The shape of the simulated curves are shown in figure 4.5. This effect was described in terms of an attachment-detachment model (discussed in section 4.6.1). Larger bubble surface area fluxes lead to a larger rate of particle bubble detachment, leading to a reduced overall flotation constant. The region of approximate linearity is larger for smaller cell sizes. This is due to the longer retention time in larger flotation cells, which leads to a greater extent of particle-bubble detachment.

A graph of the $k_r - S_b$ relationship for various froth heights was obtained with the empirical relationship for bubble surface area flux given by Gorain et al. (1999) and the froth zone recovery relationship given by Vera et al. (2002). These results differ from those obtained by Deglon et al. (1999), but are similar to the experimental results obtained by Gorain et al. (1998b). Deglon et al. (1999) neglected froth effects and modelled the cells as a single, well mixed phase. This assumption accounts for the deviations from non-linearity at low bubble surface area fluxes. The non-linearity in the $k_r - S_b$ curves shown in figure 4.6 is due to low froth zone recoveries at low froth retention times. This effect is not present in figure 4.5 because the froth zone was not modelled. The data presented by Gorain et al. (1998b) was for S_b values up to about 100 s^{-1}. It is possible that the type of deviations from non-linearity shown in figure 4.5 may occur at high bubble surface area fluxes.

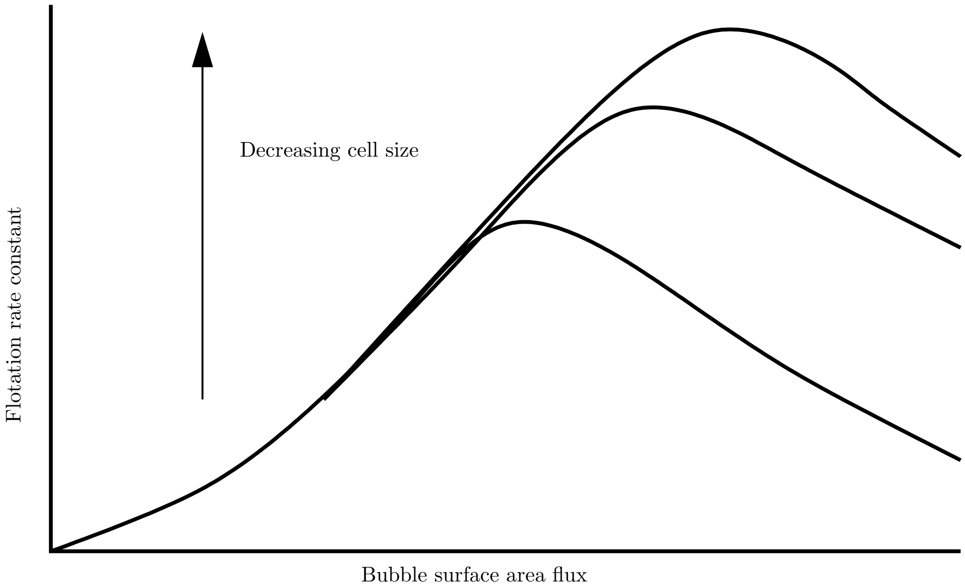

Figure 4.5: The observed deviation from linearity in the k-S_b relationship at high bubble surface area fluxes (Deglon et al., 1999)

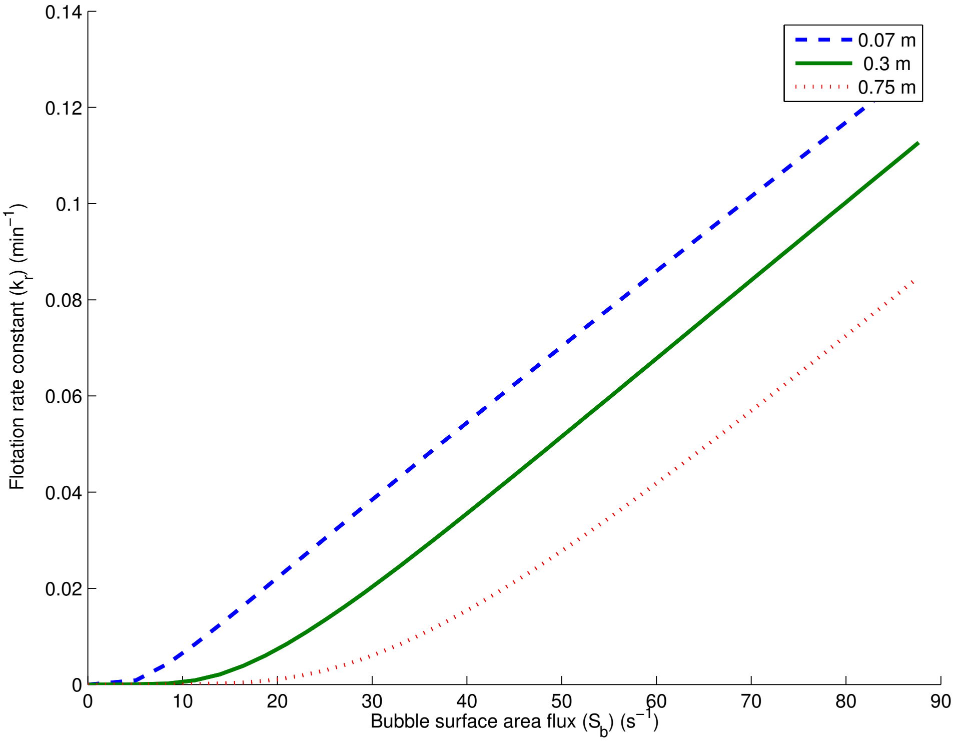

Figure 4.6: The deviation from linearity in the $k_r - S_b$ relationship as predicted by the froth recovery factor

4.4.8 Froth residence time

Froth residence time can be defined in a number of ways. Gorain et al. (1998a) defined the froth residence time based on the residence time of the gas in the froth. The froth residence time can also be defined on the basis of the flowrate of the froth slurry (Lynch et al., 1981: 86) or as the residence time of particles in the froth. These definitions are shown in equations 4.13 to 4.15.

$$\tau_{f,\text{gas}} = \frac{Q_g}{V_f} \tag{4.13}$$

$$\tau_{f,\text{slurry}} = \frac{Q_c}{V_f} \tag{4.14}$$

$$\tau_{f,\text{particles}} = \frac{Q_f}{V_f} \tag{4.15}$$

4.4.9 Entrainment

The attachment of valuable mineral particles to air bubbles is the most important flotation mechanism, with the majority of particles in the concentrate entering the froth through this selective attachment. However, both valuable mineral and gangue particles may enter the froth through entrainment (Savassi et al., 1998).

The net flow of entrained particles depends on the upward velocity of the froth and on the rate of drainage of particles from the froth (Savassi et al., 1998). Particles may drain from the froth via the lamella between bubbles, or the froth may collapse. The particle size and specific gravity of the gangue particles has an important effect on the degree of entrainment (Kirjavainen, 1996), since this determines the drainage rate of the particles. The froth structure and stability also influences entrainment.

There is a strong relationship between the recovery of water and the recovery of material through entrainment. Most of the correlations that have been proposed are based on the recovery of water. Subrahmanyam & Forssberg (1988) suggested that the recovery rate of gangue is proportional to the recovery rate of water.

Entrainment can also be described with a classification function shown in equation 4.16 (Lynch et al., 1981). This factor may be defined for each particle size fraction.

$$CF_i = \frac{\text{Mass particles of } \quad i^{\text{th}} \quad \text{size interval per unit water in concentrate}}{\text{Mass particles of } \quad i^{\text{th}} \quad \text{size interval per unit water in the pulp}} \tag{4.16}$$

The rate of recovery of entrained particles can then be calculated from equation 4.17. It can be assumed that the recovery of water can be described by a first order rate equation, as given in equation 4.18. Using this assumption, the recovery of particles by entrainment can be described by 4.19.

$$RR_g = \frac{RR_w}{MWP} \times CF \times MGP \qquad (4.17)$$

$$RR_w = k_w MWP \qquad (4.18)$$

$$RR_g = k_w \times CF \times MGP \qquad (4.19)$$

The classification function is also dependent on froth residence time. The form of equation shown in equation 4.20 has been proposed to describe this effect (Kirjavainen, 1996). α is a transfer coefficient for water drainage from froth and ω is a transfer coefficient for the drainage of solids from the froth (Kirjavainen, 1996) and τ_f is the froth residence time. α is usually essentially zero and can be neglected. In this case, equation 4.20 becomes 4.21.

$$CF_i = \frac{1 + \alpha_i \tau_f}{1 + \omega \tau_f} \qquad (4.20)$$

$$CF_i = \frac{1}{1 + \omega \tau_f} \qquad (4.21)$$

4.5 Flotation kinetics

The application of kinetics to flotation was originally based on the analogy between chemical reactions (where molecules collide and react) and flotation (where hydrophobic particles collide with bubbles and attach to the bubbles) (Lynch et al., 1981). Like chemical reaction kinetics, the flotation kinetics of an ore are often determined in a semi-batch setup before it is applied to a continuous process.

4.5.1 Batch flotation tests

In batch flotation, a batch of feed is treated with the required chemicals and then aerated. The froth is more or less continuously removed. The composition of the pulp and froth varies continually. At the start of the batch test, the composition of the material in the pulp is that of the feed, while at the end of the test, the pulp is essentially tailings. Water is often added during the tests to keep the volume constant (Gaudin, 1957).

4.5.2 Rate equation

The general rate equation for flotation is shown in equation 4.22 (Polat & Chander, 2000). The pseudo rate constant, k_r, depends on the conditions in the flotation cell and may vary with time.

$$r = -k_r C^m_{\text{particle}}(t)\, C^n_{\text{bubble}}(t) \tag{4.22}$$

There has been some discussion of the actual order of the flotation process, but first order equations (shown in equation 4.23) are most common. First order models assume that the rate of particle-bubble collisions is first order with respect to the number of particles and that the bubble concentration remains constant. Batch flotation test data support first order data under most operating conditions (Polat & Chander, 2000).

$$r = -k_r C_{\text{particle}} \tag{4.23}$$

The first order rate equation can be modified so that data over a wide range of conditions can be described. The value of k_r can be varied to describe the behaviour of different mineral species. k_r can either be described by a continuous or discrete distribution.

4.5.3 Continuous model

This model assumes that the particles to be floated have a continuous distribution of rate constants (Lynch et al., 1981) given by a frequency function $f(k)$ (Ferreira & Loveday, 2000).

The frequency function is usually a well known statistical density function such as gamma and rectangular distributions. The application of these distributions to the feed is often not realistic (Kapur et al., 1991).

The feed can also be divided into a number of classes with all the particles in a class having the same rate constant. This is the approach used in discrete models.

4.5.4 Discrete models

The discrete model form is shown in equation 4.24. Discrete models are often used because calculations are much simpler and these models often give a good fit of batch data (Lynch et al., 1981: 65).

$$r = \sum_{i=1}^{n} k_{r,i} C_{\text{particle}} \tag{4.24}$$

The simplest models divide the mineral into a fast floating and a slow floating class, shown in equation 4.25 (Lynch et al., 1981). A third class of non-floatible material, represented by the recovery at infinite time (R_{inf}) is sometimes added (Villeneuve et al., 1995).

$$r = k_{\text{slow}} C_{\text{particle}} + k_{\text{fast}} C_{\text{particle}} \tag{4.25}$$

4.5.5 The effect of particle size on flotation rate

Particle size has a marked effect on flotation performance, but this effect is often difficult to predict. Particle size plays an important role in the probability of particles colliding with bubbles, attachment to bubbles as well as the probability of the particle staying attached to the bubble. Fine particles have less particle– bubble collisions and tend to become entrained. Small particles may have an excessive absorption of chemicals (Feng & Aldrich, 1999) or some surface oxidation may occur (Wills, 1997: 285). Larger particles are more likely to detach from the bubbles or may be too heavy for particle bubble attachment. The minerals in large particles are also often not sufficiently liberated to float. A qualitative graph of the flotation rate as a function of average particle size is given in figure 4.7.

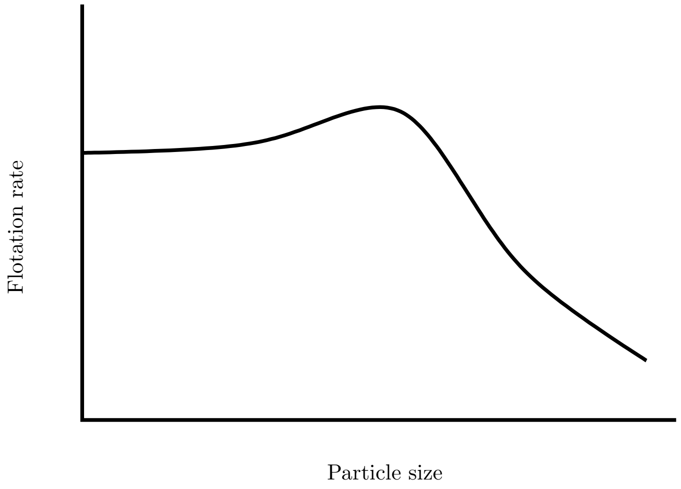

Figure 4.7: The flotation rate as a function of particle size (Wills, 1997)

This effect can be modelled by dividing the feed into classes based on mineral content and particle size and determining a rate constant for each class.

4.6 Population balance models

Population balance models aim to predict the bubble size and number of bubbles in flotation cells. The bubble population is predicted in terms of sub-processes such as bubble break up and coalescence. The flotation cell is usually modelled as two separate zones:

- the impeller zone and
- the bulk zone.

Bubble breakage occurs primarily in the impeller zone, while bubble coalescence occurs mainly in the bulk zone. Air enters the cell near the impeller, where bubble breakage occurs. The bubbles are pumped to the bulk zone, where they can either be recirculated to the impeller zone, or rise to form froth (Deglon et al., 1999).

In the impeller zone, it can be assumed that bubble breakage occurs through collisions between bubbles and in turbulent eddies. The bubble breakage frequency can be defined as the product of the bubble-eddy collision frequency and the breakage efficiency. Calculating the breakage rate requires knowledge of the spectral energy density of the eddies. This makes this calculation infeasible for industrial applications (Deglon et al., 1999).

However, studies into the breakage rate has shown that the bubble breakage rate reduces with decreasing bubble size and drops to zero at the maximum stable bubble size ($d_{b,max}$). The bubble breakage rate also reduces with increasing gas holdup (ϵ).

The bubble population in the bulk zone can be calculated by considering

- the number of bubbles entering the bulk zone from the impeller zone;
- the bubble loss through coalescence;
- the number of bubbles formed from bubble coalescence;
- the number of bubbles leaving the zone through recirculation and
- the number of bubbles leaving the phase as froth.

4.6.1 Particle – bubble attachment and detachment

In flotation kinetics, the sub-processes such as particle – bubble collision, attachment and detachment are usually lumped into a single flotation rate constant (Deglon et al., 1999). However, particle – bubble attachment and detachment rates can also be considered separately. The form of this type of equation is given in 4.26. Since this equation requires the bubble concentration, this model is often combined with a bubble population model to predict the flotation performance.

$$r = \underbrace{-k_{\text{attachment}}C_{\text{free particles}}C_{\text{available bubbles}}}_{\text{attachment rate}} + \underbrace{k_{\text{detachment}}C_{\text{loaded bubbles}}}_{\text{detachment rate}} \tag{4.26}$$

The rate of attachment is the product of the attachment rate constant and the concentration of available bubbles and free particles in the pulp. The detachment rate is the product of the detachment rate constant and the concentration of bubbles with attached particles (Bloom & Heindel, 2002).

The “available” bubble concentration has been interpreted as either the total bubble concentration, the concentration of bubbles without attached particles or the concentrations of bubbles which carry less particles than the maximum number of particles a bubble can support (Bloom & Heindel, 2002).

4.7 Probability models

The rate constant of a mineral particle may be expressed as the probability of success of a series of events. The flotation rate depends on the probability that a particle and bubble will collide ($Pr_{\text{collision}}$) and the probability that the particle will attach to the bubble ($Pr_{\text{attachment}}$). The effect of the froth phase can be included with a froth stability factor, FS. These terms can be combined to calculate the rate constant, as shown in equation 4.27.

$$k = Pr_{\text{collision}} Pr_{\text{attachment}} FS \tag{4.27}$$

The froth stability factor can be divided into $Pr_{\text{not detached}}$, the probability that a particle reaches the froth column without detaching from the bubble and Pr_{drainage}, the probability of drainage from the froth (Lynch et al., 1981: 62).

$$FS = Pr_{\text{not detached}} Pr_{\text{drainage}} \tag{4.28}$$

There are several factors which complicate the use of probability models for the calculation of flotation rates. Assumptions which may not be valid are required to calculate the probabilities from fundamental equations. The equations obtained with this approach also contain parameters which cannot be measured (Lynch et al., 1981: 63).

4.8 Modelling the effect of reagent additions

4.8.1 Modelling the effect of collector addition rates

There are two main approaches to modelling the effect of the addition of collector addition:

- modification of rate constants and
- modification of the fraction of slow-floating particles.

The physical interpretation of the second approach is that the purpose of a collector is to transform valuable mineral particles into fast floating particles (Lynch et al., 1981:

134). This approach requires that the assumption that the mineral can be divided into slow and fast floating fractions is reasonable.

The effect of collector addition rates can also be modelled by modifying the flotation rate constant. Perez-Correa et al. (1998) modified the rate constant as shown in equation 4.29. This empirical equation reproduced experimental behaviour. a, b, c an d are empirical constants.

$$k_r = aQ^3_{\text{collector}} + bQ^2_{\text{collector}} + cQ_{\text{collector}} + d \tag{4.29}$$

4.8.2 Modelling the effect of frother addition rates

The stability of the froth in the cells depends on the frother addition in the cell. At low frother addition rates, the froth is unstable and collapses easily. At higher froth addition rates, the froth becomes more stable and the rate of transfer from the froth to the launder is increased. This should increase the froth zone recovery, as well as the recovery through entrainment. It is usually possible to develop a model for the effect of the addition of frother on water recovery rate. The water recovery rate is then used to calculate the recovery of mineral through entrainment.

CHAPTER 5

Model implementation

In chapter 4, literature on flotation modelling was reviewed. This chapter focuses on the formulation of a model from fundamental equations as well as empirical relationships. The implementation of this model in *Simulink* will also be discussed and the model responses to inputs will be given.

5.1 Model equations

5.1.1 Mass balances

Pulp phase balance

Material enters the cell in the pulp phase as feed and pulp flows out of the cell as tailings. Valuable mineral particles can leave the pulp phase through true flotation (particle/bubble attachment), and both gangue and valuable mineral particles can leave the pulp through entrainment. Mineral and gangue can also re-enter the pulp through froth drainage, bubble breakage and similar mechanisms. The component balances for the minerals in the cell (gangue and valuable mineral) are shown in equation 5.1.

$$\frac{\mathrm{d}M_{i,p}}{\mathrm{d}t} = m_{i,\text{feed}} - m_{i,\text{tailings}} - m_{\text{flotation}} - m_{\text{entrainment}} + m_{\text{drop back}}. \tag{5.1}$$

The mass flow rate of each component out of the cell in the tailings is calculated from the concentration of each component in the pulp and the total volumetric flowrate.

$$m_{i,\text{tailings}} = C_{i,p}Q_{\text{tailings}} \tag{5.2}$$

For cells connected in a bank, the volumetric flow rate from one cell to the next can be estimated from equation 5.3, where L_2 is the pulp level in the next cell and L_1 is the pulp level in the current cell. The total volume of pulp in the cell is the sum of all mineral and water volumes. The pulp level in the cell will be higher than the level calculated from the volume given in equation 5.5 due to air holdup. The total pulp phase volume (pulp and air) is calculated from equation 5.6 and the pulp level in the cell is given by equation 5.7. The pulp phase concentrations in the cell can be calculated from equation 5.4

$$Q_{\text{tailings}} = K_{\text{flow}}\rho g\sqrt{(L_2 - L_1)} \tag{5.3}$$

$$C_{i,p} = \frac{M_{i,p}}{V_p} \tag{5.4}$$

$$V_p = \sum_{i=1}^{n}\frac{M_{i,p}}{\rho_i} + \frac{M_{w,p}}{\rho_w} \tag{5.5}$$

$$V_{p,t} = \frac{V_p}{(1-\epsilon_g)} \tag{5.6}$$

$$L_p = \frac{V_{p,t}}{A_c} \tag{5.7}$$

For purposes of equation 5.3, a constant density is assumed, since the densities in two adjacent cells are approximately the same. The contribution of the mass of air in the pulp phase to the mixture density is also neglected.

The pulp flow from the last cell can be manipulated with a control valve in the line leaving the bank. The flow through a control valve is given by equation 5.8 (Luyben, 1990: 214).

$$Q_{\text{tailings}} = K_{\text{valve}}f(x)\sqrt{\frac{\Delta P}{\rho}} \tag{5.8}$$

If it is assumed that the outlet is at atmospheric pressure, the pressure difference across the control valve is equal to the pressure in the final cell in the bank. This leads to equation 5.9.

$$Q_{\text{tailings}} = K_{\text{valve}}f(x)\sqrt{gL} \tag{5.9}$$

The mass flow from the cell in the concentrate is discussed in sections 5.1.2 and 4.4.9. A mass balance can also be calculated for water as shown in equation 5.10. The mass flow of water in the tailings is given by equation 5.11. Water also leaves the cell in the concentrate. Lynch et al. (1981: 76) modelled the rate of water recovery by assuming that the recovery rate is proportional to the concentration of water in the cell (see equation 5.12). The effect of froth height on the proportionality constant may also be included, but in this case it was assumed that k_w is not a function of froth level.

$$\frac{\mathrm{d}M_w}{\mathrm{d}t} = m_{w,\text{feed}} - m_{w,\text{tailings}} - m_{w,\text{entrainment}} \tag{5.10}$$

$$m_{w,\text{tailings}} = C_{w,p} Q_{\text{tailings}} \tag{5.11}$$

$$r_{w,\text{entrainment}} = k_w M_{w,p} \tag{5.12}$$

$$m_{w\text{entrainment}} = r_{w,\text{entrainment}} V_p \tag{5.13}$$

$$C_{w,p} = \frac{M_{w,p}}{V_p} \tag{5.14}$$

5.1.2 Overall flotation rate

The pulp phase mass balance given above included terms for material leaving the pulp phase through flotation as well as for material re-entering the pulp phase through drop back. It is possible to combine these terms and model the nett rate at which material leaves the pulp phase. This rate is given by equation 5.16.

The collection zone constant (used to model the rate at which particles attach to bubbles in the pulp phase) is the product of the mineral floatibility, P and the bubble surface area flux S_b. Not all of the particles that leave the collection zone (pulp phase) are collected in the concentrate. The fraction of attached particles in the pulp phase that leaves the cell in the concentrate is given by the froth recovery factor. The mineral floatibility, the bubble surface area flux and the froth recovery factor are combined in the overall flotation rate constant, given in equation 5.15, and the overall flotation rate is given by equation 5.16. The mass flow from the cell due to true flotation is then given by equation 5.17

$$k_{\text{overall}} = P S_b R_f \tag{5.15}$$

$$r_{\text{overall}} = k_{\text{overall}} C_{i,p} \tag{5.16}$$

$$m_{i,\text{flotation}} = V_p r_{\text{overall}} \tag{5.17}$$

Mineral floatibility This parameter is a function of mineral type as well as particle size. Vera et al. (2002) measured values of mineral floatibility (P) in the range 1×10^{-3} to

$2{,}7 \times 10^{-3}$ for the floatibility of the valuable mineral. It was assumed that the floatibility of gangue was zero.

Bubble surface area flux The following empirical relationship given by Gorain et al. (1999) was used to calculate the bubble surface area flux.

$$S_b = aN_s^b \left(\frac{Q}{A}\right)^c As^d P_{80}^e \tag{5.18}$$

The values for the constants a to e, were reported by Gorain et al. (1999). They also reported the impeller peripheral speed (N_s) and aspect ratio (As) for several cell designs as well as the 80% passing feed size for several plants. Typical values were selected and are given in Appendix A.

Froth recovery factor The relationship given by Vera et al. (2002) was used to predict the froth recovery factor. This correlation requires the froth retention time, which was calculated from the volumetric air flow rate and the volume taken up by froth in the cell. The froth level is given by the difference between the cell height and the pulp level. The drainage rate parameter, ω, ranges between $7\,{}^{1}\!/_{\text{min}}$ and $75\,{}^{1}\!/_{\text{min}}$ while the froth parameter β has a value of $4\,{}^{1}\!/_{\text{min}}$ (Vera et al., 2002).

$$R_{f,i} = e^{(-\beta\tau_f)} + \left(1 - e^{(-\beta\tau_f)}\right)\left(\frac{1}{1+\omega_i\tau_f}\right) \tag{5.19}$$

$$\tau_f = \frac{AL_f}{Q} \tag{5.20}$$

$$L_f = L_t - L_p \tag{5.21}$$

5.1.3 Entrainment

The entrainment term in the mass balance describes the nett rate at which particles leave the pulp phase through entrainment, which depends on the rate at which the particles enter the froth phase and the rate at which the particles drain back into the pulp.

The rate of recovery of both valuable mineral and gangue particles due to entrainment can be given by equation 5.23, with $m_{\text{entrainment}}$ the water flowrate from the cell expressed as mass per unit time. The rate of recovery of water can be assumed to be proportional to the mass of water in the pulp phase (Lynch et al., 1981: 74). Equation 5.23 can be written in terms of the proportionality constant, K_{rw}. The classification function, CF is given by equation 5.22

$$CF = \frac{1}{1+\omega\tau_f} \tag{5.22}$$

$$m_{\text{entrainment}} = m_{w,\text{entrainment}} CF \frac{M_{i,p}}{M_{w,p}} \tag{5.23}$$

$$m_{\text{entrainment}} = k_w CF M_{i,p} \tag{5.24}$$

5.1.4 Air addition rate

Many of the equations given in the above sections depend on variables related to the flowrate of air. These variables include the froth retention time as well as the air holdup. It is not realistic to assume that a change to the air flowrate to the cell will lead to an instantaneous change in the air holdup or froth retention time. It is more realistic to assume that these properties will change gradually over a few seconds. This effect was introduced by adding a first order lag to describe the effective change in the air flow to the cell.

5.2 Degrees of freedom analysis

The variables used in the flotation model for a system with i mineral components is shown in table 5.1 and the modelling equations for the system are shown in table 5.2. It is assumed that the feed flowrate to the system is specified by upstream conditions. This leaves two variables free to be used for control. The selection of manipulated variables will be discussed in section 5.5.1.

5.3 Feed

The feed to an actual flotation circuit consists of a spectrum of particle sizes and compositions. If the aim is to quantitatively predict the recovery of valuable mineral or the grade of the final concentrate, the feed should be divided into several classes depending on particle size and composition, with each class having its own floatibility. However, the aim of this model is not to make quantitative predictions, but to investigate the dynamics of the flotation process. The feed was divided into two classes: valuable mineral and gangue. The valuable mineral has a relatively high floatibility, while the gangue mineral has zero floatibilty and is only recovered in the concentrate through entrainment. The feed is also only represented by a single particle size.

The feed ore typically contain very low concentrations of valuable mineral. For example, the typical grade of one of the major platinum group metal reefs in the Bushveld Complex is about 4.5 ppm, while other (less valuable) mineral ores contains higher concentrations. The feed to the circuit is in the form of a mineral and water pulp mixture.

Table 5.1: Model variables

Variable	Symbol	Units	Number
Feed flowrate	$m_{i,\text{feed}}$	$^{\text{kg}}/_{\text{s}}$	i
Feed water flowrate	$m_{i,w}$	$^{\text{kg}}/_{\text{s}}$	1
Air flow to cell	Q	$^{\text{m}^3}/_{\text{s}}$	1
Control valve fraction	x		1
Mass of component i in cell	M_i	kg	i
Mass water in cell	M_w	kg	1
Mass flow mineral out in tailings	$m_{i,\text{tailings}}$	$^{\text{kg}}/_{\text{s}}$	i
Mass flow water out in tailings	$m_{i,\text{tailings}}$	$^{\text{kg}}/_{\text{s}}$	1
Volumetric tailings flowrate	Q_{tailings}	$^{\text{m}^3}/_{\text{s}}$	1
Volume pulp in cell	V_p	$^{\text{m}^3}/_{\text{s}}$	1
Pulp phase volume (voidage included)	$V_{p,t}$	m^3	1
Pulp phase mineral concentrations	$C_{i,p}$	$^{\text{kg}}/_{\text{m}^3}$	i
Pulp phase water concentrations	$C_{w,p}$	$^{\text{kg}}/_{\text{m}^3}$	1
Pulp level	L_p	m	1
Froth level	L_f	m	1
Mass flow out in concentrate (particle bubble attachment)	$m_{i,\text{flotation}}$	$^{\text{kg}}/_{\text{s}}$	i
Mass flow out in concentrate (entrainment)	$m_{i,\text{entrainment}}$	$^{\text{kg}}/_{\text{s}}$	i
Overall flotation rate	r_{overall}	$^{\text{kg}}/_{\text{m}^3\text{s}}$	i
Overall flotation rate constant	k_{overall}	s^-1	i
Froth recovery factor	R_f		1
Bubble surface area flux	S_b	$^{\text{m}^2}/_{\text{m}^2\text{s}}$	1
Froth residence time	τ_f	s	1
Classification factor	CF		1
Total		15+ 8i	

Table 5.2: Modelling equations

Relationship	Equation	Number of equations
Mineral mass balance	5.1	i
Water mass balance	5.10	1
Flow relationship	5.3 or 5.9	1
Mineral mass flow from cell in tailings	5.2	i
Water mass flow from cell in tailings	5.11	1
Volume pulp in cell	5.6	1
Total pulp phase volume (voidage included)	5.6	1
Pulp phase mineral concentrations	5.4	i
Pulp phase water concentration	5.14	1
Pulp level	5.7	1
Mass out in concentrate (true flotation)	5.17	i
Mass out in concentrate (entrainment)	5.24	i
Overall flotation rate	5.16	i
Overall rate constant	5.15	i
Froth recovery factor	5.19	1
Froth residence time	5.20	1
Froth level	5.21	1
Bubble surface area flux	5.18	1
Classification function	(5.22)	1
Total		7i + 12

Table 5.3: Degrees of freedom summary

number of variables	15 + 8i
number of equations	-(12 + 7i)
number of variables specified externally (feed)	-(1 + i)
Degrees of freedom for control	2

Table 5.4: Model parameters

Parameter	Symbol	Units
Parameter in relationship for flow between cells	K_{flow}	$\text{m}^{4.5}\text{s}/\text{kg}$
Water recovery rate constant	k_w	$1/\text{m}^3\text{s}$
Mineral floatibility	P	dimensionless
Bubble surface area flux correlation constants	a to e in equation (5.18)	
Cell cross sectional area	A	m^2
Cell height	L_t	m
Impeller peripheral speed	N_s	m/s
Impeller aspect ratio	As	dimensionless
Drainage parameter	ω	$1/\text{s}$
Froth parameter	β	$1/\text{s}$
water drainage parameter	α	$1/\text{s}$

An initial pulp density of 1300 $^{kg}/_{m^3}$ was used. The mineral feed rate is based on a gangue mineral flow rate of 45 $^{kg}/_{s}$. For a mixture density of approximately 1300 $^{kg}/_{m^3}$, a water flow rate of 90 $^{kg}/_{s}$ was used.

5.4 Development environment

The flotation model was developed in *Matlab*'s *Simulink*. This environment is ideal for modular model design. User interfaces can also be created easily.

While it is possible to use *Matlab* m-files in the *Simulink* environment (through the m-file block), this was not done since this slows down the model considerably. The model consists exclusively of native *Simulink* blocks, such as the sum, product and integrator block. Differential equations can be solved with the integrator block.

The system that was used to integrate the mass balance differential equations is shown in figure 5.1. The difference between the inflow and outflow is integrated. The output of the integrator is then the mass in the cell. The initial condition is provided by the user. Since it is more intuitive to specify initial conditions in terms of the cell level and the concentration of each component in the cell, this input is requested from the user and then converted to a total mass in the cell.

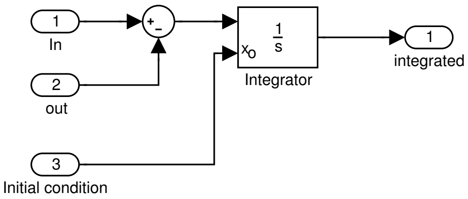

Figure 5.1: Solving differential equations in *Simulink*

Other equations can also be implemented in a similar way. For example, the *Simulink* implementation of the froth recovery factor is shown in figure 5.2.

The modelling equations were programmed as subsystems, and these subsystems were combined to form a model of a flotation cell. Several of these cell models were combined to form a flotation bank, and these banks were combined to form a circuit.

5.5 Circuit configuration

A very simple circuit configuration was used to investigate the control. The circuit consists of a six cell rougher bank and a six cell cleaner bank. The tailings from the rougher bank is the final tailings, while the concentrate from this bank is sent to the cleaning bank. The concentrate from this bank is the final product, while the tailings from this bank (which may still contain significant valuable mineral) is recycled to the first bank. a schematic representation of the circuit is shown in figure 5.3.

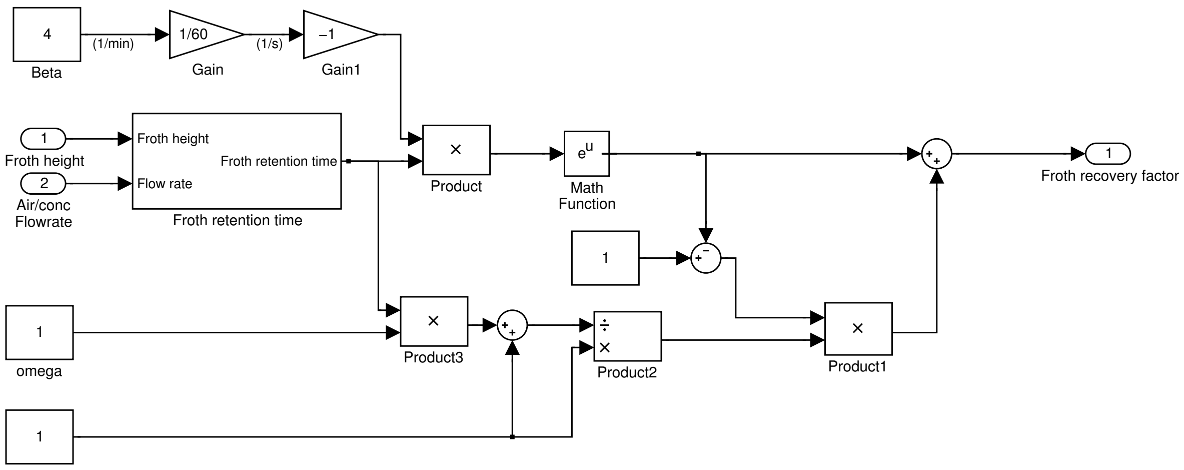

Figure 5.2: The *Simulink* implementation of equation (5.19)

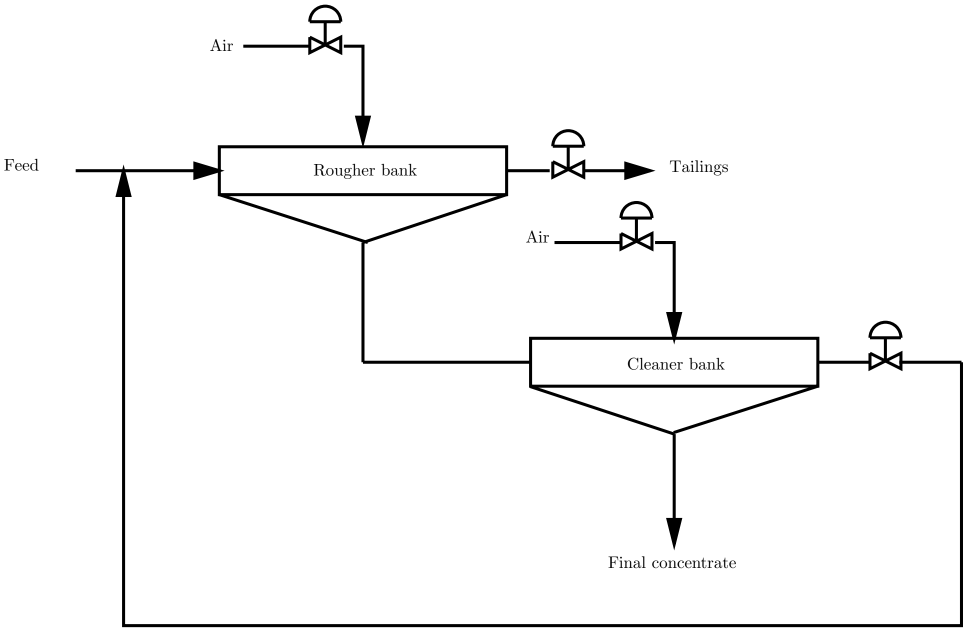

Figure 5.3: The circuit configuration used for identification and control

The cell banks consist of a series of cell models connected in series. The tailings from each cell is sent to the next cell, with the flowrate a function of the relative levels in the cells (see equation (5.3)). The flow from the final cell is manipulated with a control valve. This flow is a function of the level in the cell, as well as the valve fraction.

5.5.1 Manipulated variables

Each cell bank has two manipulated variables. The air flow and the level in the cell bank influence the controlled variables, namely recovery and grade. The air flow to each cell is directly specified by the advanced controller. This is equivalent to assuming that the dynamics of the air flow control base layer loop is negligible.

Most applications of advanced control on flotation circuits use the pulp level setpoints as manipulated variables. The performance of the advanced controller relies on the performance of the PI level control loops. For this system, the two level control loops interact, which will deteriorate the control performance.

The flows from the cell banks may also be used as manipulated variables, which will allow the advanced controller to handle the interactions between the variables. Since the tuning parameters of a PI level controller will not be valid for the entire operating region covered by the nonlinear advanced controller, the flows from the cell banks will be used as manipulated variables.

5.5.2 Controlled variables

The primary controlled variables for this system are the final grade and recovery. For the system where the level setpoints are used as manipulated variables, the level in the final cell of each bank are secondary controlled variables.

5.5.3 Operating point

Like linear step response models, Volterra models are also obtained around an operating point. During identification, the model is initially at steady state at this operating point. The values of the model inputs and outputs at the operating point are given in Appendix A.

5.5.4 Scaling

All the inputs, as well as the outputs, were converted to deviations from the operating point given above scaled to be in the range -1 to 1 by dividing the variables with the maximum expected or allowed changes.

5.6 Process responses

The process responses to various step inputs are shown in figures 5.4 to 5.11. The system is clearly nonlinear, with asymmetrical responses to symmetric inputs. The gain of the system is also dependent on the magnitude of the input. All these factors make linear step responses unsuitable for the modelling of this system and motivates the use of nonlinear modelling techniques such as Volterra series.

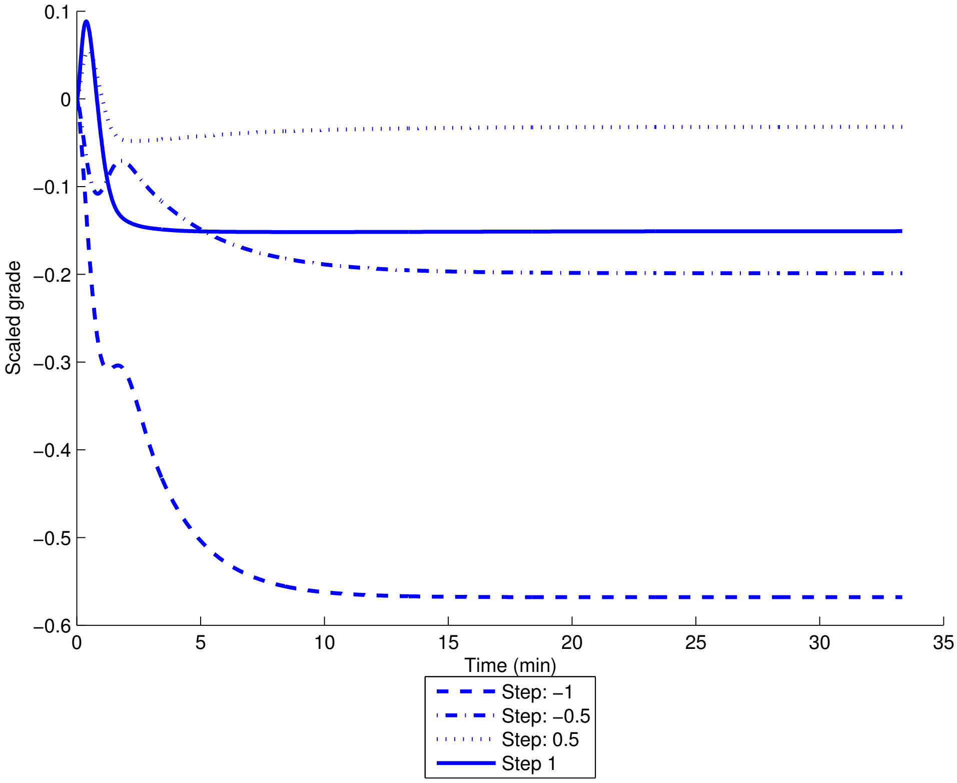

Figure 5.4: The effect of step changes in the air flow rate to the rougher bank on the concentrate grade

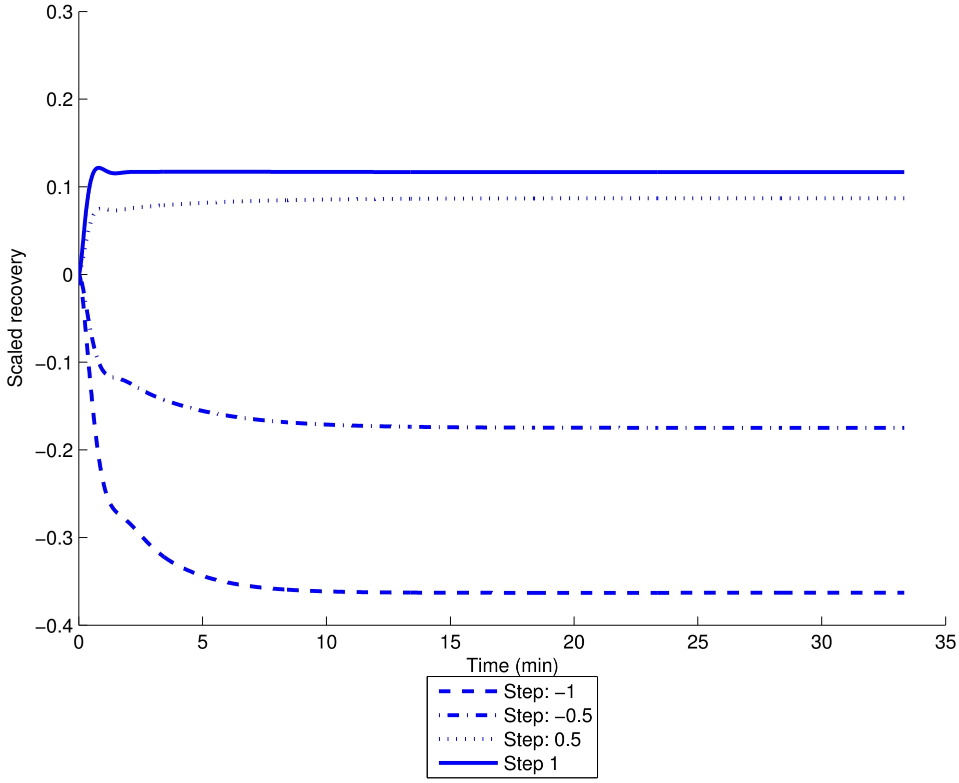

Figure 5.5: The effect of the air flow rate to the rougher bank on the recovery of valuable mineral

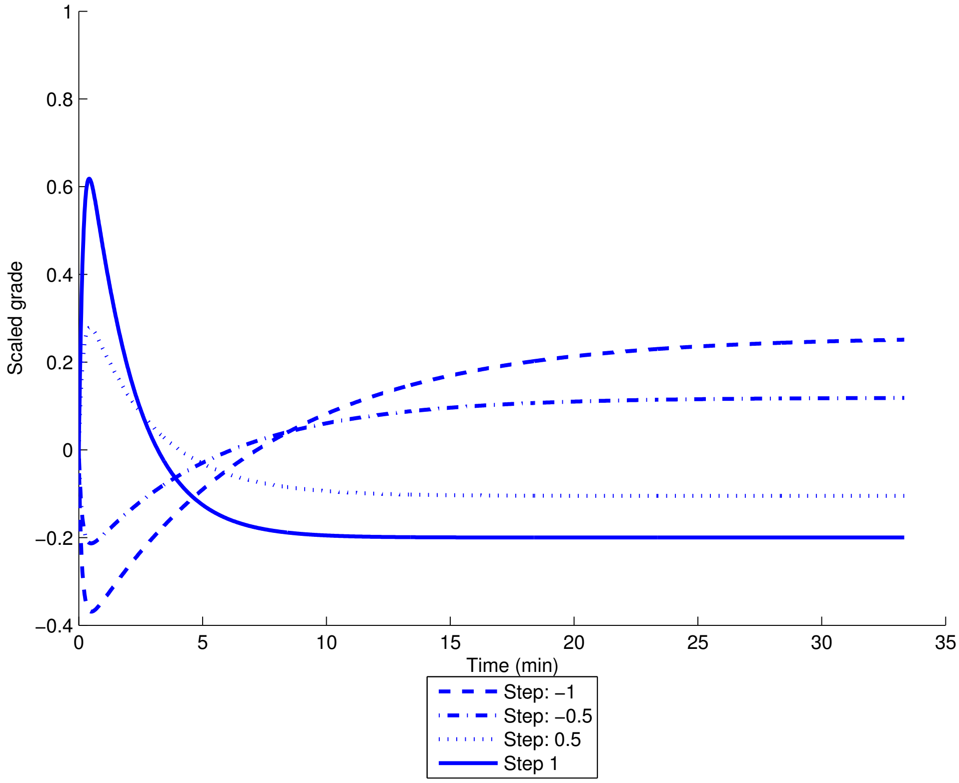

Figure 5.6: The effect of changes in the air flow to the cleaner bank on the concentrate grade

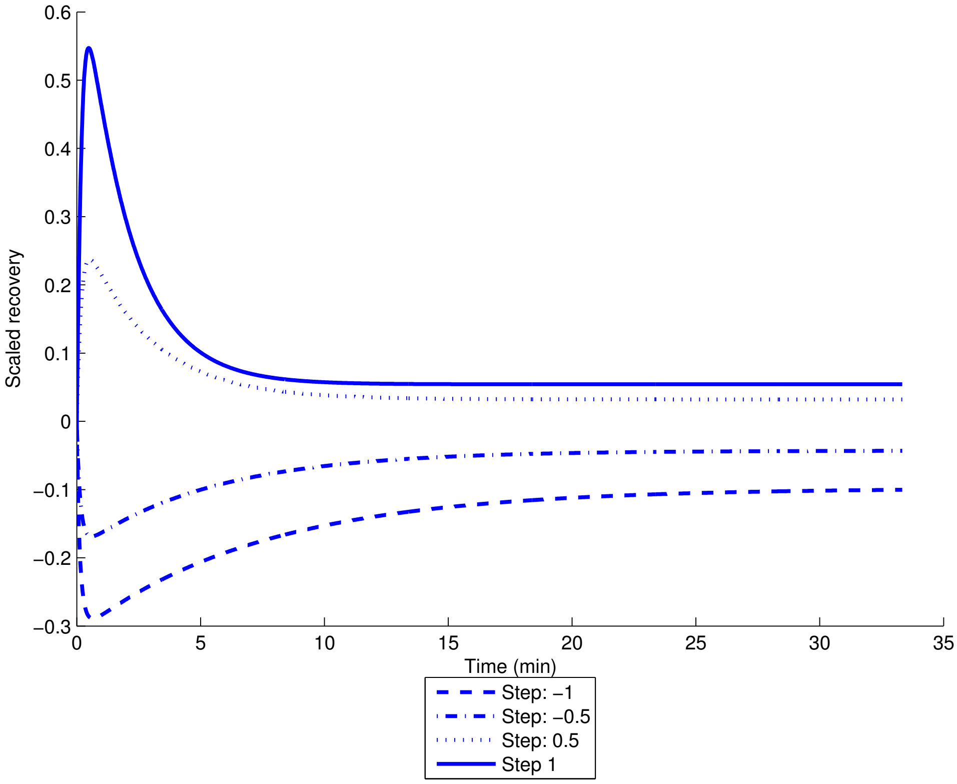

Figure 5.7: The effect of step changes in the air flow to the cleaner bank on the recovery of valuable mineral

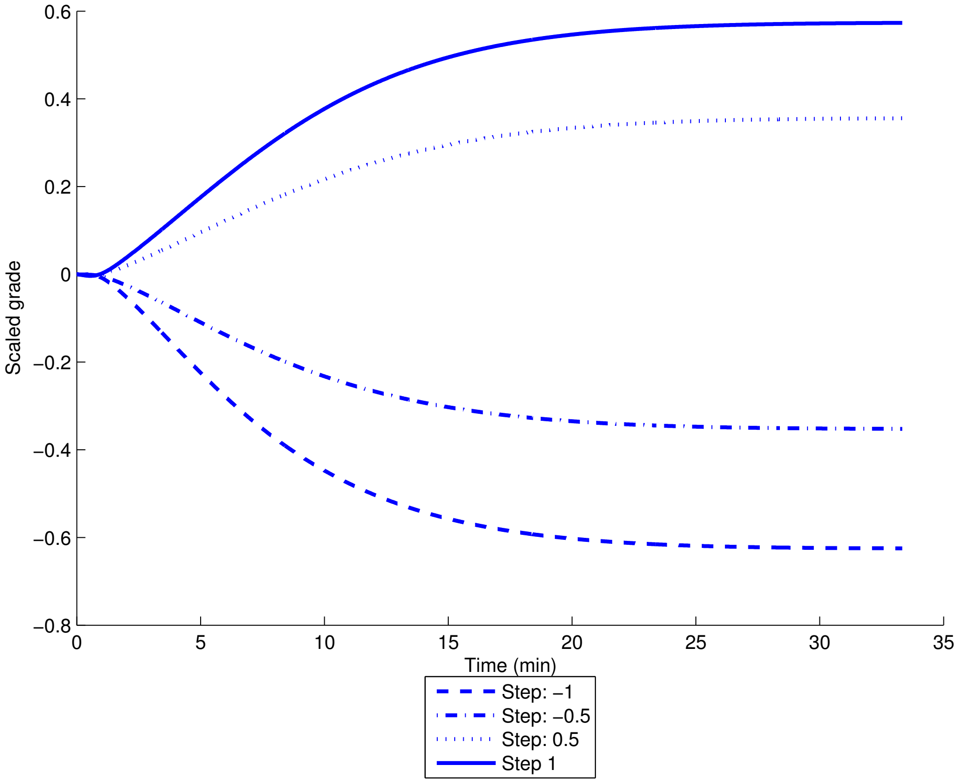

Figure 5.8: The effect of step changes in the setpoint for the flow from the rougher bank on the concentrate grade

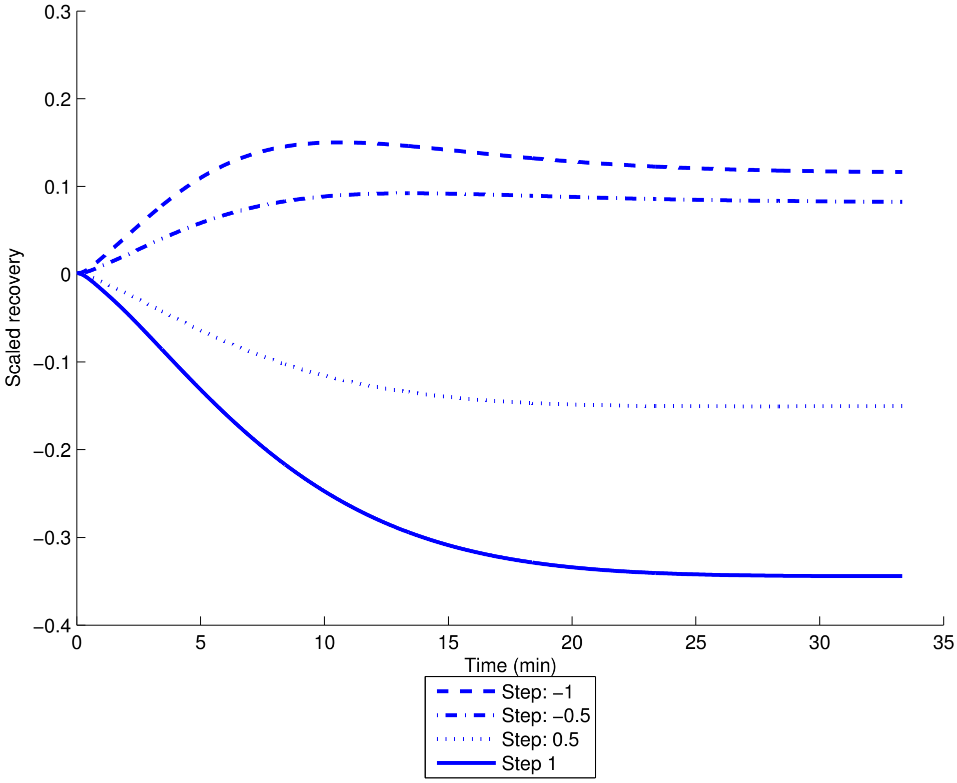

Figure 5.9: The effect of step changes in the setpoint for the flow from the rougher bank on the recovery of valuable mineral

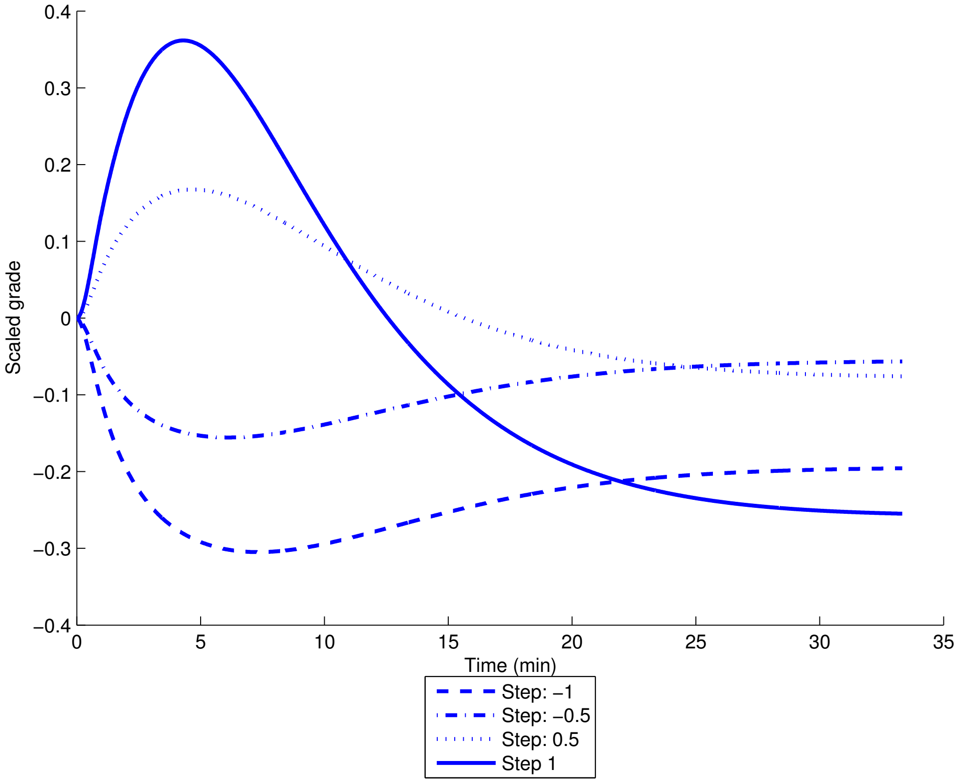

Figure 5.10: The effect of step changes in the setpoint for the flow from the cleaner bank on the concentrate grade

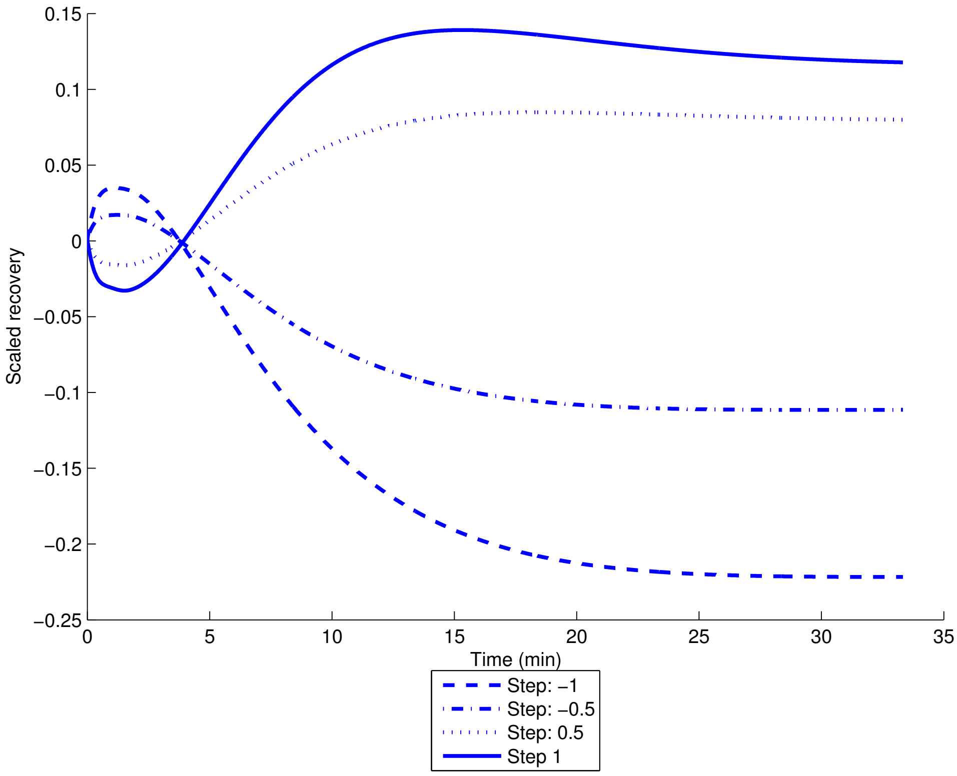

Figure 5.11: The effect of step changes in the setpoint for the flow from the cleaner bank on the recovery of valuable mineral

CHAPTER 6

Model Identification

This chapter describes model identification techniques, with emphasis on the identification of nonlinear models from plant input-output data. The identified nonlinear models are intended for use in nonlinear model predictive control algorithms. Linear models are also discussed, since many of the nonlinear model identification techniques are extensions of linear identification techniques.

The input sequence that is used to generate the input–output data is an important aspect of model identification and an overview of commonly used input sequences is included.

Volterra series is an attractive nonlinear model type, since it is a higher order extension of the linear impulse response model. Later chapters will show that nonlinear model predictive schemes based on second order Volterra models consist of a linear MPC scheme with nonlinear add-on terms. This means that Volterra model predictive controllers can be seen as an enhancement of generally accepted linear model predictive controllers.

6.1 Models for predictive control

Both linear and nonlinear model predictive control require some form of plant model to predict the process behaviour. Linear model predictive control schemes almost exclusively use step or impulse response models (Qin & Badgwell, 2003). These models are obtainable from plant input – output data and require no fundamental process knowledge. The ease with which a model can be obtained contributes to the popularity of linear model predictive controllers.

Nonlinear model predictive control (NMPC) schemes require nonlinear process models. Both fundamental and empirical models have been used for nonlinear model predictive control (Henson, 1998). However, since models obtainable from plant input-output data are desirable from an industrial perspective, this chapter will focus on the identification of models from plant data.

6.2 System properties

6.2.1 Nonlinearity

A system H is linear if the following properties hold for the system:

$$\text{Additive property} \quad H\left[u_1 + u_2\right] = H\left[u_1\right] + H\left[u_2\right] \tag{6.1}$$

$$\text{Homogenous property} \quad H\left[cu\right] = cH\left[u\right] \tag{6.2}$$

A system is nonlinear if it does not satisfy the properties given in equations 6.1 and equation 6.2 (Bendat, 1991: 2).

For a linear system, the system output to a random Gaussian input sequence will also have a Gaussian probability density function while nonlinear systems will tend to have non-Gaussian outputs in response to a Gaussian input (Bendat, 1991: 3).

Most industrial processes are nonlinear to some extent, so the properties given in equations 6.1 and 6.2 will not apply exactly. Some processes are sufficiently linear around a given operating point so that the process can be approximated by a linear model. However, in some cases a process cannot be satisfactorily approximated by a linear model, such as:

- highly nonlinear processes and
- mildly nonlinear processes operated over a wide range.

In these cases, a nonlinear model is required to accurately describe the behaviour of the process.

6.2.2 System memory

The memory of a system refers to the effect that past inputs have on the present outputs of the system. Systems can be classified as: (Bendat, 1991: 3)

- infinite memory;
- finite memory and

- zero memory.

The output of a zero memory system is not affected by past inputs. A system at steady state can be modelled as a zero memory system, since the system output can be predicted from the current input alone.

A system with infinite memory is influenced by all past inputs. However, for most systems, an input has negligible effect a certain time after the input has been applied. Such a system has a finite or fading memory. Most chemical processes are fading memory systems.

6.3 Signal characteristics

Some model identification methods, especially methods which make use of stochastic input signals, are based on statistical analysis of the input and output signals. Some concepts which will be used later in this chapter are discussed in this section.

Random signals may be either stationary or non stationary. For stationary data series, the statistical properties of a data set does not change with time, while the statistical properties of non stationary data does change with translations in time (Bendat, 1991: 8). Most of the properties defined in this section assume that the sequence is stationary.

6.3.1 Mean

For a stationary stochastic process, the mean can be defined as shown in equation 6.3 (Soderstrom & Stoica, 1989: 100), where E is the expectation operator.

$$\overline{u} \triangleq E\left[u(t)\right] \tag{6.3}$$

For a deterministic signal the mean is defined by replacing the expectation operator with the limit of a normalised sum (Soderstrom & Stoica, 1989: 102). The mean can be calculated by evaluating the sum for large N.

$$\overline{u} \triangleq \lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} u(t) \tag{6.4}$$

6.3.2 Central moments

The n^{th} central moment of a random variable u is the moment taken about the mean and is defined in equation 6.5. The first central moment is zero, while the second order central moment of a sample is the variance (see equation 6.6).

$$n^{th} \quad \text{central moment} = E\left[u - E\left[u\right]^{n}\right] \tag{6.5}$$

$$\sigma^2 = E\left[u - E\left[u\right]^2\right] \tag{6.6}$$

6.3.3 Skewness

The skewness of a distribution can be defined as in equation 6.7 (Weisstein, 2004b) where μ_i is the i^{th} central moment. The skewness is a degree of the asymmetry of a distribution. Equation 6.7 can be written in terms of the mean and standard deviation of a random sequence u, shown in equation 6.8.

$$\zeta = \frac{\mu_3}{\mu_2^{3/2}} \tag{6.7}$$

$$\zeta = \frac{E\left[u - \overline{u}\right]}{\sigma^3} \tag{6.8}$$

6.3.4 Kurtosis

The kurtosis is a measure of the degree of peakedness of a distribution, and can be defined as the normalised fourth central moment (Weisstein, 2004a). This can be written in terms of the mean and standard deviation of a random sequence, as shown in equation 6.10. With this definition, the kurtosis of a normal distribution is equal to 3. If it is required that the kurtosis of a normal distribution should be equal to zero, the alternative definition shown in equation 6.11 is used.

$$\kappa = \frac{\mu_2}{\mu_2^2} \tag{6.9}$$

$$\kappa = \frac{E\left[u - \overline{u}\right]}{\sigma^4} \tag{6.10}$$

$$\kappa = \frac{E\left[u - \overline{u}\right]}{\sigma^4} - 3 \tag{6.11}$$

6.3.5 Covariance, cross covariance and cross bi-corellations

The covariance of a stationary random sequence u is defined in equation 6.12 (Soderstrom & Stoica, 1989: 100) and the cross covariance between the input u and the output y can be defined as shown in equation 6.13. For a deterministic signal, the expectation operator is replaced by the limit of the normalised sum (see equation 6.14.

$$r_{uu}(\tau) \triangleq E\left[u(t+\tau) - \overline{u}\right]\left[u(t) - \overline{u}\right]^T \tag{6.12}$$

$$r_{uy} = E\left[y(t+\tau)u(t)^T\right] \tag{6.13}$$

$$r(\tau) \triangleq \lim_{N\to\infty} \frac{1}{N}\sum_{t=1}^{N} \left[u(t+\tau) - \overline{u}\right]\left[u(t) - \overline{u}\right]^T \tag{6.14}$$

The cross bi-correlation between input u and output y is defined by equation 6.15 and may be estimated from data from equation 6.16 (Koh & Powers, 1985), (Pearson et al., 1996).

$$t_{uuy} = E\left[u(k)u(k-i)y(k-j)\right] \tag{6.15}$$

$$t_{uuy}(\tau_1, \tau_2) = \frac{1}{N}\sum_{t=1}^{N-\tau_1} u(t)u(t+\tau_1-\tau_2)y(t+\tau_1) \tag{6.16}$$

6.4 Input signals

Plant input–output data is required for the identification of empirical models. The model that is identified is usually dependent on the input signal that is used to obtain the data (Sung & Lee, 2003). Some identification procedures require a signal that is persistently exciting.

6.4.1 Persistent excitation

The concept of persistent excitation will be illustrated with the identification of an impulse response model from plant data with the least squares technique, which will be discussed in section 6.7.1. The model coefficients are given by equation 6.21 (Sung & Lee, 2003). The symbols are defined in equations 6.17 to 6.20. $\boldsymbol{U}$ is a N by np matrix, where N is the number of measurements and np is the number of parameters to be estimated. In this case the number of parameters is equal to MH, the model horizon.

$$\boldsymbol{u}(k) = \begin{bmatrix} u(k) & u(k-1) & \cdots & u(k-MH+1) \end{bmatrix} \tag{6.17}$$

$$\boldsymbol{U} = \begin{pmatrix} \boldsymbol{u}(1) \\ \vdots \\ \boldsymbol{u}(N) \end{pmatrix} \tag{6.18}$$

$$\boldsymbol{h}_1 = \begin{bmatrix} h(1) & h(2) & \cdots & h(M) \end{bmatrix} \tag{6.19}$$

$$\boldsymbol{Y} = \begin{pmatrix} y(1) \\ \vdots \\ y(N) \end{pmatrix} \tag{6.20}$$

$$\boldsymbol{h} = \left[U^T U\right]^{-1} U^T Y \tag{6.21}$$

For equation 6.21 to have a solution, the matrix U^TU must be invertible. This requirement of non-singularity leads to the concept of persistent excitation. Persistent excitation will now be defined for both stochastic and deterministic input signals.

For a stochastic input signal $\{u(k)\}$, the correlation matrix shown in equation 6.22 can be defined. If this matrix is nonsingular, the signal is persistently exciting of order n. (Nowak & Veen, 1994), (Soderstrom & Stoica, 1989: 120).

$$\boldsymbol{R}_{uu}(n) = E\left[U^T U\right] \tag{6.22}$$

$$\boldsymbol{R}_{uu}(n) = \begin{pmatrix} r_{uu}(0) & r_{uu}(1) & \cdots & r_{uu}(n-1) \\ r_{uu}(-1) & r_{uu}(0) & & r_{uu}(n-2) \\ \vdots & & \ddots & \vdots \\ r_{uu}(1-n) & \cdots & \cdots & r_{uu}(0) \end{pmatrix} \tag{6.23}$$

with

$$r_{uu}(\tau) = E\left[u(t+\tau)u(t)\right] \tag{6.24}$$

If the input sequence is deterministic, the expectation operator can be replaced by the limit of the normalised sum. The sample correlation matrix is then constructed as shown in equations 6.25 and 6.26 (Soderstrom & Stoica, 1989: 120). The input signal is persistently exciting if the sample correlation matrix is nonsingular and the limit given in equation 6.25 exists.

$$r_{uu}(\tau) = \lim_{N\to\infty} \sum_{k=1}^{N} \frac{1}{N} u(k+\tau)u(k) \tag{6.25}$$

$$\boldsymbol{R}_{uu}(n) = \begin{pmatrix} r_{uu}(0) & r_{uu}(1) & \cdots & r_{uu}(n-1) \\ r_{uu}(-1) & r_{uu}(0) & & r_{uu}(n-2) \\ \vdots & & \ddots & \vdots \\ r_{uu}(1-n) & \cdots & \cdots & r_{uu}(0) \end{pmatrix} \tag{6.26}$$

To summarise, in order to identify a model with np parameters with the least squares technique, the input signal must be persistently exciting of order $n = np$. This observation applies to the *consistent* estimation of parameters in *noisy* systems.

Nowak & Veen (1994) investigated the excitement requirements for extended Volterra series. The extended Volterra series is formulated in terms of the input vector defined in equation 6.27. $\otimes$ indicates the Kronecker product. The extended Volterra filters include input interaction terms not included in ordinary Volterra filters. A data matrix (6.28) can then be constructed from these vectors.

$$\boldsymbol{u}(k) = \begin{bmatrix} 1 \\ u(k) \\ u(k)^2 \\ \vdots \\ u(k)^N \end{bmatrix} \otimes \begin{bmatrix} 1 \\ u(k-1) \\ u^2(k-1) \\ \vdots \\ u^N(k-1) \end{bmatrix} \otimes \cdots \otimes \begin{bmatrix} 1 \\ u(k-MH+1) \\ u^2(k-MH+1) \\ \vdots \\ u^N(k-MH+1) \end{bmatrix} \tag{6.27}$$

$$\boldsymbol{V}(k) = [\boldsymbol{u}(k) \cdots \boldsymbol{u}(k+\tau)]^T \tag{6.28}$$

$$\boldsymbol{Y} = \boldsymbol{V\theta} \tag{6.29}$$

The parameters for this extended Volterra series can then also be estimated from the least squares method (equation 6.30). This means that the matrix $(\boldsymbol{V^T V})$ must be invertible for the input signal to be persistently exciting. Nowak & Veen (1994) proved that this matrix is invertible if the input sequence takes on at least $n+1$ discrete values, where n is the order of the Volterra series to be identified. This result is related to the $2n^{th}$ central moment of an independent identically distributed (IID, see section 6.4.2) sequence. The same result applies to ordinary Volterra filters.

$$\boldsymbol{\theta} = \left(\boldsymbol{V}^T\boldsymbol{V}\right)^{-1}\boldsymbol{V}^T\boldsymbol{Y} \tag{6.30}$$

6.4.2 Independent identically distributed sequences

If the random variables in a sequence are independent from each other and all have the same distribution, then the sequence is independent and identically distributed (IID). The covariance matrix for IID sequences is given by equation 6.31. $\boldsymbol{I}$ is a identity matrix.

$$\boldsymbol{R}_{uu} = \sigma^2\boldsymbol{I} \tag{6.31}$$

6.4.3 Step function

A step input is a common input used in the identification of linear models. A step function is defined as shown in equation 6.32 (Soderstrom & Stoica, 1989).

$$u(t) = \begin{cases} 0 & t < 0 \\ \zeta & t \geq 0 \end{cases} \tag{6.32}$$

A step input does give valuable information about the process, including the gain, dead time and dominant time constant. Unfortunately, for nonlinear processes the results obtained depend on the size of the input (Sung & Lee, 2003).

The covariance function, r_{uu}, for a step of size ζ is equal to ζ^2 for all τ. This means that the correlation matrix $R_{uu}(n)$ is nonsingular if and only if $n = 1$. This means that a step input is only persistently exciting of order 1 (Soderstrom & Stoica, 1989: 121) and that consistent estimation of more than one parameter in a noisy environment cannot be obtained with a single step input.

6.4.4 White noise

White noise signals are uncorrelated in time. The autocorrelation function for white noise is given by equation 6.33.

$$E\left[u(k)u(k+\tau)\right] = \begin{cases} 0 & \text{for} \quad \tau \neq 0 \\ \sigma^2 & \text{for} \quad \tau = 0 \end{cases} \tag{6.33}$$

Gaussian white noise is a white noise with a Gaussian probability distribution. An important disadvantage of Gaussian white noise for the identification of nonlinear systems is that the amplitude distribution of the input signal is concentrated about the origin. To accurately identify a nonlinear process the full normal operating range of the input signal should be used (Nowak & Veen, 1994).

Another type of white noise is a random binary sequence (RBS). This signal consists of binary values of a specified amplitude. The signal switches from one level to another with a probability of 0.5. A long sequence is required for the relationship given in equation 6.33 to hold (Sung & Lee, 2003).

The covariance matrix for a Gaussian white noise of variance σ^2 is $R_{uu}(n) = \sigma^2 I_n$ where I_n is the identity matrix of size n. This matrix is invertible for all n, so Gaussian white noise is persistently exciting of all orders (Soderstrom & Stoica, 1989: 121). Similarly, for a random binary sequence taking on levels $\pm\zeta$, the variance is ζ^2 and the covariance matrix $R_{uu} = \zeta^2 I_n$ and the matrix is invertible for all n.

6.4.5 Pseudo random input sequences

Pseudo random sequences are deterministic signals that are designed to have the same covariance function as white noise and this property is independent of signal length. This makes pseudo random signals more practical than random binary signals (Sung & Lee, 2003).

Pseudo random binary input sequences

Pseudo random binary sequences (PRBS) are efficient signals for linear system identification. These input sequences have similar spectral characteristics to white noise (Nowak & Veen, 1994).

A pseudo random binary sequence is usually created by creating a basic pseudo random sequence of length Ps and repeating this sequence as many times as required (Soderstrom & Stoica, 1989: 138). It can be shown that the covariance function for a pseudo random sequence of period P is given by equation 6.34.

$$r_{uu}(\tau) = \begin{cases} \zeta^2 & \text{for} \quad \tau = 0, \pm Ps, \pm 2Ps \ldots \\ -\frac{\zeta^2}{Ps} & \text{elsewhere} \end{cases} \tag{6.34}$$

Since $r_{uu}(\tau) = r_{uu}(1)$ for all $\tau \neq 0, \pm P, \pm 2Ps \ldots$, the covariance matrix for a pseudo random binary sequence is given by equation 6.35.

$$R_{uu} = \begin{pmatrix} \zeta^2 & -\frac{\zeta^2}{Ps} & \cdots & -\frac{\zeta^2}{Ps} \\ \frac{-\zeta^2}{Ps} & \zeta^2 & \cdots & -\frac{\zeta^2}{Ps} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{\zeta^2}{Ps} & -\frac{\zeta^2}{Ps} & \cdots & \zeta^2 \end{pmatrix} \tag{6.35}$$

For large Ps, the off-diagonal elements tend to zero and the covariance matrix for the pseudo random binary signal tends toward the covariance matrix of a random binary sequence. If the total required signal length is fixed, the best pseudo random binary sequence has a period $Ps = N$ (Sung & Lee, 2003).

Pseudo random multilevel input sequences

Pseudo random binary sequences are not persistently exciting for Volterra series of order 2 and higher. This means that pseudo random multilevel sequences are required.

6.5 Linear models

6.5.1 ARMAX

A popular linear model representation is the AutoRegressive Moving Average with eXogenous outputs model (ARMAX) (Pearson, 1995). The disturbance sequence or noise sequence $\{e(k)\}$ represents the effect of noise, modelling errors and unmeasured disturbances.

$$y(k) = \sum_{j=1}^{p} a_j y(k-j) + \sum_{j=0}^{q} b_j u(k-j) + \sum_{j=0}^{r} c_j e(k-j) \tag{6.36}$$

This model can be extended directly to multiple inputs by adding terms corresponding to additional inputs to the second term. This extension can also be applied to nonlinear systems. For the case where $p = 0$ and $q = \infty$, corresponds to the convolution model with $\{b_j\}$ representing the impulse response of the system. The case where $p = \infty$ and $q = 0$ represents and infinite order autoregressive model (Pearson & Pottmann, 2000).

6.5.2 Linear convolution models

Step response models

The original dynamic matrix control (DMC) used a step response model. The model consists of a series of step response coefficients a_i taken at a sample rate Δt. These coefficients represent the response of the process to an unit input change. The definition of the unit input is arbitrary and is chosen as a scaling factor to obtain the response coefficients in a desired range. The process response to a series of changes in the manipulated variable is the sum of the process responses due to the individual manipulated variable changes. This leads to the step response formulation given in equation 6.37 (Cutler, 1982). y_0 is the process initial condition. If the process is in terms of deviation variables and initially at steady state, y_0 will be 0.

$$y(k) = y_0 + \sum_{i=1}^{M} a_i \Delta u(k-i) \tag{6.37}$$

Impulse response model

The impulse response model, shown in equation 6.38, is similar to the step response model. In this case the model coefficients represent process response to an impulse input, although other input signals such as random binary sequences (RBS) and pseudo random binary sequences (PRBS) may be used to obtain the model coefficients. The impulse response coefficients may also be derived from the step response coefficients by taking the backward difference of the step response coefficients (Seborg et al., 1989: 652).

$$y(k) = y_0 + \sum_{i=1}^{MH} h_i u(k-i) \tag{6.38}$$

$$h_i = a_i - a_{i-1} \tag{6.39}$$

$$h_0 = 0 \tag{6.40}$$

The relationship between step and impulse response models is shown in figure 6.1.

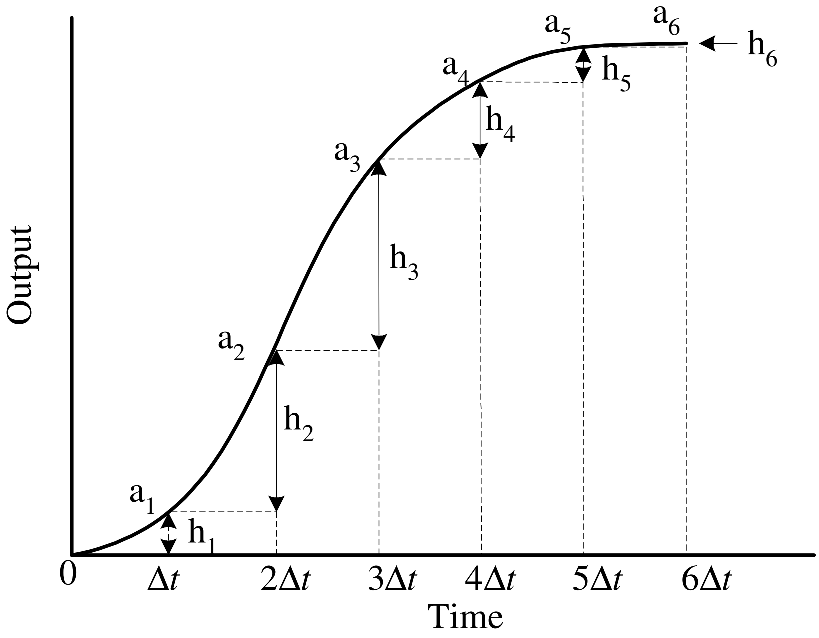

Figure 6.1: The relationship between step and impulse response models

6.6 Nonlinear models

6.6.1 Fundamental models

Fundamental dynamic models can be derived by applying transient mass, energy and momentum balances to the system. The process model will consist of a set of differential and algebraic equations.

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f\left(x, u\right) \tag{6.41}$$

$$0 = g\left(x, u\right) \tag{6.42}$$

$$y = h\left(x, u\right) \tag{6.43}$$

These equations must be discretized since NMPC models are usually formulated in discrete time (Henson, 1998).

Fundamental models have some advantages. These models require less process data than empirical models and plant tests are usually not required. These models are often valid for a very wide range of conditions (Henson, 1998).

Fundamental models are complicated and time consuming to develop and may be too complex for use in NMPC algorithms and will not be discussed further.

6.6.2 Volterra Series

Volterra series can be seen as an higher order extension of the linear impulse response model. The predicted output of a Volterra series at time k is the sum of terms up to the n^{th} order terms (Bendat, 1991: 75).

$$y\left(k\right) = y_0 + y_1\left(k\right) + y_2\left(k\right) + y_3\left(k\right) + \cdots + y_N \tag{6.44}$$

The first, second and third order terms are given in equations 6.45 to 6.47

$$y_1\left(k\right) = \sum_{i=1}^{MH} h_1\left(i\right) u\left(k - i\right) \tag{6.45}$$

$$y_2(k) = \sum_{i=1}^{MH} \sum_{j=1}^{MH} h_2(i, j) u(k - i) u(k - j) \tag{6.46}$$

$$y_3(k) = \sum_{i=1}^{MH} \sum_{j=1}^{MH} \sum_{l=1}^{MH} h_3(i, j, l) u(k - i) u(k - j) u(k - l) \tag{6.47}$$

This work will focus on second order Volterra series. Higher order Volterra series require a large number of parameters. The number of required parameters for a second order Volterra series is shown in Table 6.1 (Pearson et al., 1996). In some cases, even a second order Volterra series require an impractically large data set to identify all the parameters. This leads to the idea of “pruned” Volterra series. In pruned Volterra series, some of the second order coefficients are constrained to zero. This decreases the number of parameters which must be identified (Pearson et al., 1996).

Second order Volterra models can be represented diagrammatically by a linear block

Table 6.1: The number of required parameters for a second order Volterra series

Parameter	Type	Number
y_0	Constant	1
$\{h_1(i)\}$	Linear	MH
$\{h_2(i,j)$	Quadratic	MH^2

and a nonlinear block in parallel. This is shown in figure 6.2.

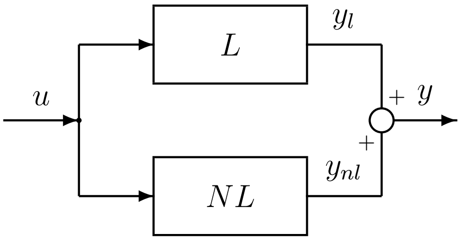

Figure 6.2: A block diagram representation of Volterra series Koh & Powers (1985)

Volterra models can be obtained from a number of sources, including (Doyle et al., 1995)

- nonlinear first principle models;
- NARMAX models;
- neural networks and
- input – output data.

The most attractive method is identification directly from input – output data and this option will be investigated further in section 6.7.2.

6.6.3 Block-oriented models

Block-oriented models can be represented by a combination of static nonlinearities and linear dynamic components. The nonlinear block is static in the sense that it does not depend on past values of the input - i.e. a memoryless block (Rivera, 2001).

Hammerstein models

The general structure of the Hammerstein model is shown in figure 6.3. N is a zero memory non-linear block, while H is a linear dynamic element with a steady state gain of 1 (Pearson & Pottmann, 2000). The steady state behaviour of the process is determined by the static nonlinearity, while the dynamics of the system is a combination of the nonlinear and linear components (Pearson & Pottmann, 2000).

The linear and nonlinear parts of the model may take various forms. For example, the nonlinear part of the model may be a neural network Al-Duwaish & Karim (1997) or a polynomial while the linear part of the model may be an ARMAX or a step response model Al-Duwaish & Karim (1997).

A second order Volterra series with a diagonal second order coefficient matrix is equivalent to a Hammerstein model. The first order term of the Volterra series contains the linear dynamics and the second order term with diagonal coefficients is essentially a static nonlinearity (Rivera, 2001). The Hammerstein reduction of a Volterra series has the form given in equation 6.48 (Koukoulas & Kalouptsidis, 2003).

$$y(k) = \sum_{i=1}^{MH} h_1(i)u(k-i) + \sum_{i=1}^{MH} h_2(i,i)u^2(k-i) \tag{6.48}$$

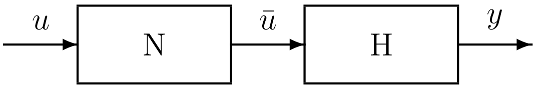

Figure 6.3: The general block diagram representation of Hammerstein models

Wiener models

The structure of a Wiener model is shown in figure 6.4. Like Hammerstein models, Wiener models also consist of a zero memory nonlinear block and a linear dynamic model but with the order of these elements reversed. Hammerstein and Wiener models show the same steady state behaviour, but different dynamic behaviour. For Hammerstein models, the qualitative shape of the curve (for example, damped or underdamped) is independent of the model input magnitude. This is not the case for Wiener models (Pearson & Pottmann, 2000).

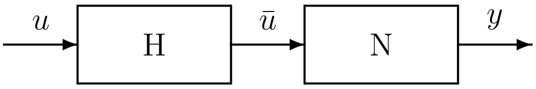

Figure 6.4: The general block diagram format of Wiener models

.

6.7 Model identification from plant data

The aim is to identify the model parameters which minimise the squared prediction error as defined in equation 6.49. The parameters for which the prediction error is a minimum is found by setting the derivative of J_f with respect to the parameter θ equal to zero

(Pearson, 1995).

$$J_f = E\left\{[\hat{y}(k) - y(k)]^2\right\} \tag{6.49}$$

$$\frac{\partial J_f}{\partial \theta} = 0 \tag{6.50}$$

If the model is linear in terms of its parameters, the linear least squares technique can be applied.

6.7.1 Identification of impulse response model

The vectors which have been used in this section were defined in equations 6.18 to 6.21.

$$\begin{aligned} J_f &= [Y - U\boldsymbol{h}]^T [Y - U\boldsymbol{h}] && (6.51)\\ &= \left[Y^T - \boldsymbol{h}^T U^T\right] [Y - U\boldsymbol{h}] && (6.52)\\ &= \left[Y^T Y - \boldsymbol{h}^T U^T Y - Y^T U\boldsymbol{h} - \boldsymbol{h}^T U^T U\boldsymbol{h}\right] && (6.53) \end{aligned}$$

The optimal parameter values can be found by setting the derivative of the objective function to zero, as shown in equation 6.54. Equation 6.54 can be written as equation 6.55, from which the parameter values can be found.

$$\frac{\partial J_f}{\partial \boldsymbol{h}} = -Y^T U + \boldsymbol{h}\left(U^T U\right) = 0 \tag{6.54}$$

$$\left(U^T U\right)\boldsymbol{h} = U^T Y \tag{6.55}$$

$$\boldsymbol{h} = \left(U^T U\right)^{-1} U^T Y \tag{6.56}$$

If u is a random input sequence, equation 6.51 can be written in terms of the expected error (equation 6.57). The parameter vector $\boldsymbol{h}$ is then given by equation 6.58, where R_{uu} is the $N \times N$ autocorrelation matrix defined in equation 6.26, and $\boldsymbol{r}_{uy}$ is the N vector of cross correlation coefficients defined in equation 6.13.

$$J_f = E\,[Y - U\boldsymbol{h}]^T [Y - U\boldsymbol{h}] \tag{6.57}$$

$$\boldsymbol{h} = R_{uu}^{-1} r_{uy} \tag{6.58}$$

6.7.2 Identification of Volterra series

This section considers the identification of second order Volterra models, of the form shown in equation 6.59, with $\hat{y}$ being the predicted output of y.

$$\hat{y}(k) = y_0 + \sum_{i=1}^{MH} h_1(i)u(k-i) + \sum_{i=1}^{MH}\sum_{j=1}^{MH} b(i,j)u(k-i)u(k-j) \tag{6.59}$$

There are several approaches to obtaining Volterra models from plant data. One of the main differences between the various approaches is the type of input sequence that is used. The input sequence must excite the process sufficiently so that the coefficients can be identified.

Identification with random and pseudorandom inputs

The constant term y_0 is chosen so that the the prediction is unbiased. An estimator is biased if the the expected value deviates from the predicted value (Soderstrom & Stoica, 1989: 18).

$$E\left[\hat{y}\right] \neq y \tag{6.60}$$

The expectation of equation 6.59 is given in equation 6.61 (Pearson et al., 1996). y_0 must be chosen so that $E\left[\hat{y}(k)\right] = E\left[y(k)\right] = \overline{y}$. This leads to equation 6.62, which can be substituted into equation 6.59 to give equation 6.63.

$$E\left[\hat{y}\right] = y_0 + \overline{u}\sum_{i=1}^{MH} a(i) + \sum_{i=1}^{MH}\sum_{j=1}^{MH} b(i,j)r_u(i) \tag{6.61}$$

$$y_0 = \overline{y} - \overline{u}\sum_{i=1}^{MH} a(i) - \sum_{i=1}^{MH}\sum_{j=1}^{MH} b(i,j)r_u(i) \tag{6.62}$$

$$\begin{aligned} \hat{y} = \quad & \overline{y} - \overline{u}\sum_{i=1}^{MH} a(i) - \sum_{i=1}^{MH}\sum_{j=1}^{MH} b(i,j)r_u(i) \\ & + \sum_{i=1}^{MH} h_1(i)u(k-i) + \sum_{i=1}^{MH}\sum_{j=1}^{MH} b(i,j)u(k-i)u(k-j) \end{aligned} \tag{6.63}$$

The following analysis will be done in terms of deviation variables, since the input, output and prediction sequences have zero mean in terms of deviation variables, defined below (Pearson et al., 1996).

$$v(k) = u(k) - \bar{u} \tag{6.64}$$

$$w(k) = y(k) - \bar{y} \tag{6.65}$$

$$z(k) = \hat{y}(k) - \bar{y} \tag{6.66}$$

Equation 6.63 can be written in terms of deviation variables and rearranged to give equation 6.67

$$\begin{aligned} z(k) = \quad & \sum_{i=1}^{MH} a(i)v(k-i) + 2\bar{u}\sum_{i=1}^{MH}\sum_{j=1}^{MH} h_2(i,j)v(k-i) + \\ & \sum_{i=1}^{MH}\sum_{j=1}^{MH} h_2(i,j)\left[v(k-i)v(k-j) - r_{uu}(i-j)\right] \end{aligned} \tag{6.67}$$

We seek the parameters $h_1(i)$ and $h_2(i,j)$ so that the following objective function is minimised (Pearson et al., 1996).

$$\begin{aligned} J_f &= E\{[\hat{y}(k) - y(k)]^2\} && (6.68)\\ &= E\{[z(k) - w(k)]\} && (6.69)\\ &= \sigma_w^2 + \sum_{i=1}^{MH} a(i)g(i) + \sum_{i=1}^{MH}\sum_{j=1}^{MH} h_2(i,j)d(i,j) && (6.70) \end{aligned}$$

σ_w^2 is the variance of the zero mean output fluctuation sequence $\{w(k)\}$. $g(i)$ and $d(i,j)$ are defined in equations 6.71 and equations 6.72.

$$\begin{aligned} g(i) = \quad & \sum_{j=1}^{MH} a(j)r_{vv}(i-j) - 2r_{wv}(i) + \\ & \sum_{m=1}^{MH}\sum_{n=1}^{MH} h_2(m,n)\left[4\bar{u}R_{vv}(i-m) + 2E\{v(k-i)v(k-m)v(k-n)\}\right] \end{aligned} \tag{6.71}$$

$$d(i,j) = \sum_{m=1}^{MH}\sum_{n=1}^{MH} h_2(m,n)D(i,j,m,n) - 4\bar{u}E\{v(k-i)v(k-m)v(k-n)\} \tag{6.72}$$

$$D(i,j,m,n) = B(i,j,m,n) + 4\bar{u}^2 R_v v(i-m) + 4\bar{u}E\{v(k-i)v(k-m)v(k-n)\} \tag{6.73}$$

$$B(i,j,m,n) = E\{v(k-i)v(k-j)v(k-m)v(k-n)\} - R_{vv}(i-j)R_{vv}(m-n) \quad (6.74)$$

The identification problem can be simplified considerably by using an input sequence with specific statistical properties (Pearson et al., 1996). In particular, if the input sequence has the properties shown in equations 6.75 and 6.76, equation 6.70 simplifies to equation 6.77. For this case, the identification of the first and second order coefficients decouple and these coefficients can then be identified separately. This is the same results obtained in equation 6.58. The requirement shown in equation 6.76 is valid for any symmetrically distributed input sequence (Pearson et al., 1996).

$$\overline{u} = 0 \quad (6.75)$$

$$E\left[v(k-i)v(k-m)v(k-j)\right] = 0 \quad \text{for all} \quad i, m, n \quad (6.76)$$

$$\begin{aligned} J_f = & \sum_{i=1}^{MH} h_1(i)\left(\sum_{j=1}^{MH} h_1(j)R_{vv}(i-j) - 2r_{wv}(i)\right) \\ & + \sum_{i=1}^{MH}\sum_{j=1}^{MH} h_2(i,j) \\ & \left(\sum_{m=1}^{MH}\sum_{n=1}^{MH} h_2(i,j)E\left[v(k-i)v(k-j)v(k-m)v(k-n)\right] - R_{vv}(i-j)R_{vv}(m-n)\right) \end{aligned} \quad (6.77)$$

The optimal model parameters can be obtained from equation 6.77 by setting the partial derivatives of the objective function to zero. The linear (first order) coefficients can be found from equation 6.78 (Pearson et al., 1996). This is the same results found in equation 6.58. The second order coefficients can be found from equation 6.79, which represents a set of simultaneous linear equations.

$$\boldsymbol{h}_1 = R_{vv}^{-1} r_{wv} \quad (6.78)$$

$$\sum_{m=1}^{M}\sum_{n=1}^{M} h(m,n)B(i,j,m,n) = t_{uuy}(i,j) \quad (6.79)$$

Koh & Powers (1985)obtained a solution for the above equation under the assumption that the input sequence is Gaussian. The second order Volterra coefficients are then given by equation 6.80, with the matrix T_{uuy} defined in equation 6.81 and $\boldsymbol{h_2}$ the second order coefficient matrix given in equation 6.82.

$$\boldsymbol{h_2} = \frac{1}{2} R_{uu}^{-1} T_{uuy} R_{uu}^{-1} \tag{6.80}$$

$$\boldsymbol{T}_{uuy} = \begin{bmatrix} t_{uuy}(1,1) & t_{uuy}(1,2) & t_{uuy}(1,3) & \cdots & t_{uuy}(1,MH) \\ t_{uuy}(2,1) & t_{uuy}(2,2) & & \vdots & \\ t_{uuy}(3,1) & & \ddots & & \\ \vdots & & & & \\ t_{uuy}(MH,1) & \cdots & & & t_{uuy}(MH,MH) \end{bmatrix} \tag{6.81}$$

$$\boldsymbol{h_2} = \begin{bmatrix} h_2(1,1) & h_2(1,2) & h_2(1,3) & \cdots & h(1,MH) \\ h_2(2,1) & h_2(2,2) & & \vdots & \\ h_2(3,1) & & \ddots & & \\ \vdots & & & & \\ h_2(MH,1) & \cdots & & & h_2(MH,MH) \end{bmatrix} \tag{6.82}$$

For an independent, identically distributed input sequence, $B(i,j,m,n)$ is given by equation 6.83, where kurtosis is defined as in equation 6.11. Note that, under this definition, the kurtosis of a Gaussian sequence is 0.

It then follows from equation 6.79 that the second order parameters are given by equation 6.84. No finite solution exists for the diagonal parameters ($i = j$) for a sequence with a kurtosis of -2, which is the kurtosis of a pseudo random binary sequence (Pearson et al., 1996). This means that pseudo random binary sequences cannot be used to identify the diagonal second order coefficients of Volterra series. This confirms the persistent excitation requirement given by Nowak & Veen (1994).

$$B(i,j,m,n) = \begin{cases} (\kappa+2)\sigma^4 & i = j = m = n \\ \sigma^4 & i = m, j = n, m \neq n \\ \sigma^4 & i = n, j = m, m \neq n \\ 0 & \text{otherwise} \end{cases} \tag{6.83}$$

$$h_2(i,j) = \begin{cases} \frac{t_{wvv}(i,j)}{(\kappa+2)\sigma^4} & i = j \\ \frac{t_{wvv}(i,j)}{2\sigma^4} & i \neq j \end{cases} \tag{6.84}$$

Identification with deterministic inputs

As discussed in section 6.7.2, the diagonal ($i = j$) parameters cannot be identified from a pseudo random binary sequence, while it is possible to identify the off-diagonal ($i \neq j$). Parker et al. (2001) developed an identification method based on this property.

A second order Volterra series can be decomposed into constant, linear, diagonal and off-diagonal terms. Equation 6.88 exploits the property $h_2(i,j) = h_2(j,i)$ (Parker et al., 2001).

$$y(k) = h_0 + L(k) + D(k) + O(k) \tag{6.85}$$

where

$$L(K) = \sum_{i=1}^{MH} h(i)u(k-1) \tag{6.86}$$

$$D(k) = \sum_{i=1}^{MH} h_2(i,i)u^2(k-i) \tag{6.87}$$

$$O(k) = 2\sum_{i=1}^{MH}\sum_{j=1}^{i-1} h_2(i,j)u(k-i)u(k-j) \tag{6.88}$$

While the diagonal parameters do not contribute to the identification under pseudo random binary sequences, the off-diagonal parameters do not contribute to the process response for an impulse response, since $u(k-i)u(k-j) = 0$ for $i \neq j$ (Parker et al., 2001). The constant, linear and diagonal parameters can be identified from the input sequence given in equation 6.89. The model response for this input sequence is given in equation 6.90.

$$u(k) = \begin{cases} \xi & k = 0 \\ 0 & 1 \leq k \leq MH \\ -\xi & k = MH+1 \\ 0 & MH+2 \leq k \leq 2MH+1 \end{cases} \tag{6.89}$$

$$y(k) = \begin{cases} h_0 & k = 0 \\ h_0 + h_1(k)\zeta + h_2(k,k)\zeta^2 & 1 \leq k \leq MH \\ h_0 & k = MH+1 \\ h_0 - h_1(k-MH-1)\zeta + h_2(k-MH-1, k-MH-1)\zeta^2 & MH+2 \leq k \leq 2MH+1 \end{cases} \tag{6.90}$$

The aim is to find model parameters which minimise the squared prediction error. To minimise the this error, it is required that $\frac{\partial J_f}{\partial h_1(i)} = 0$. This gives equation 6.92 (Parker et al., 2001) Similarly, the requirement that $\frac{\partial J_f}{\partial h_2(i,j)}$ leads to equation 6.93 (Parker et al., 2001).

$$J_f = \sum_{k=0}^{2MH+1} \left[y(k) - \hat{y}(k)\right]^2 \tag{6.91}$$

$$h_1 = \frac{y(k) - y(k + MH + 1)}{2\zeta} \tag{6.92}$$

$$h_0 + \zeta^2 \hat{h}_2(k,k) = \frac{y(k) - y(k + M + 1)}{2} \tag{6.93}$$

$$h_0 + \frac{\zeta^2}{M+1} \sum_{k+1}^{M} \hat{h}_2(k,k) = \frac{1}{2M+2} \sum_{k=0}^{2M+1} y(k) \tag{6.94}$$

Equations 6.93 and 6.94 can be solved simultaneously to obtain explicit expressions for h_0 and $h_2(k,k)$ given in equations 6.95 and 6.96.

$$h_0 = \frac{y(0) + y(M+1)}{2} \tag{6.95}$$

$$h_2(k,k) = \frac{y(k) - y(0) + y(k + M + 1) - y(M+1)}{2\zeta^2} \tag{6.96}$$

These equations estimate $2MH + 1$ model parameters from $2MH + 2$ observations. This approach should be reasonable in absence of noise and measurement errors. However, it is usually better to repeat the input sequence defined in 6.89 and take an average of the parameters (Parker et al., 2001).

It is possible to estimate the off diagonal parameters by selecting deterministic input sequences designed to excite the plant so that the responses are determined primarily by the off-diagonal parameters (Parker et al., 2001). However, as was seen in section 6.7.2, the off-diagonal parameters can easily be identified from a pseudorandom binary sequence.

CHAPTER 7

Model identification results

The results obtained during the identification of second order Volterra models of the flotation circuit described in chapter 5 are presented in this chapter. The identification of the first order and diagonal second order Volterra coefficients are discussed in section 7.1.3. The identification of the second order off diagonal elements are discussed in section 7.1.4, while the use of pruned Volterra models are discussed in section 7.1.5.

7.1 Volterra models

The identification methods proposed by Parker et al. (2001), described in sections 6.7.2 were applied to the flotation circuit model. In this procedure, the linear and diagonal coefficients of the Volterra model is identified, after which the off-diagonal coefficients are identified.

7.1.1 Sampling interval

The sampling interval for the model must be small enough to model the process dynamics, but smaller sampling intervals require more parameters for a fixed model horizon. A sampling interval of 25 s was selected. This sampling interval was sufficiently small to include the inverse response observed in some of the model responses.

7.1.2 Model horizon

The model horizon should be sufficiently large to include all the relevant dynamics as well as the steady state gain. For this system, the process has reached steady state after 2000 s. With a sampling time of 25 s, this gives a model horizon of 80 intervals.

7.1.3 Identification of the linear and diagonal coefficients

The linear and off-diagonal coefficients are identified with the input sequence shown in equation 7.1 (see section 6.7.2). The value of gamma should be chosen so that the operating region of interest is excited (Parker et al., 2001). For the scaled system, the value of ξ is equal to one.

$$u(k) = \begin{cases} \xi & k = 0 \\ 0 & 1 \leq k \leq MH \\ -\xi & k = MH + 1 \\ 0 & M + 2 \leq k \leq 2MH + 1 \end{cases} \tag{7.1}$$

For a model horizon MH, MH first order (linear) Volterra coefficients will be identified, and MH second order diagonal coefficients will be identified. This means that $2MH$ parameters are identified from $2MH + 1$ data points. This is a valid approach in the absence of noise and measurement errors (Parker et al., 2001).

The results obtained for the model consisting of only the first order model coefficients are shown in figures 7.1 to 7.8. The responses shown are for input steps of -1 and 1. Since the first order Volterra models are linear, the models cannot adequately describe the nonlinear behaviour of the process.

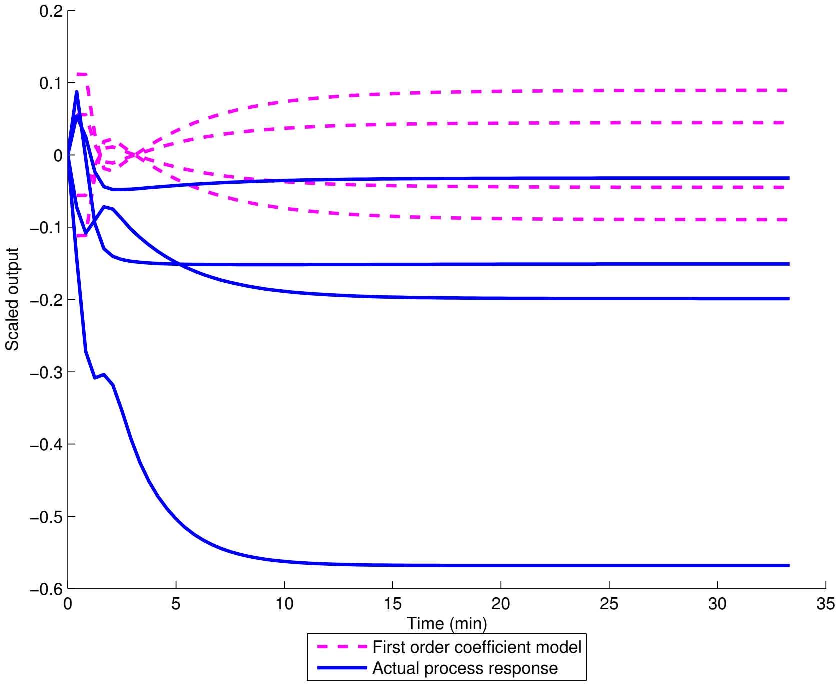

Figure 7.1: Comparison between the actual and response of the product grade to steps in the air flow to the rougher bank and the predicted response for a first order Volterra series

The results obtained for a model consisting of a combination of the first order Volterra coefficients as well as the diagonal second order Volterra coefficients are shown in fig-

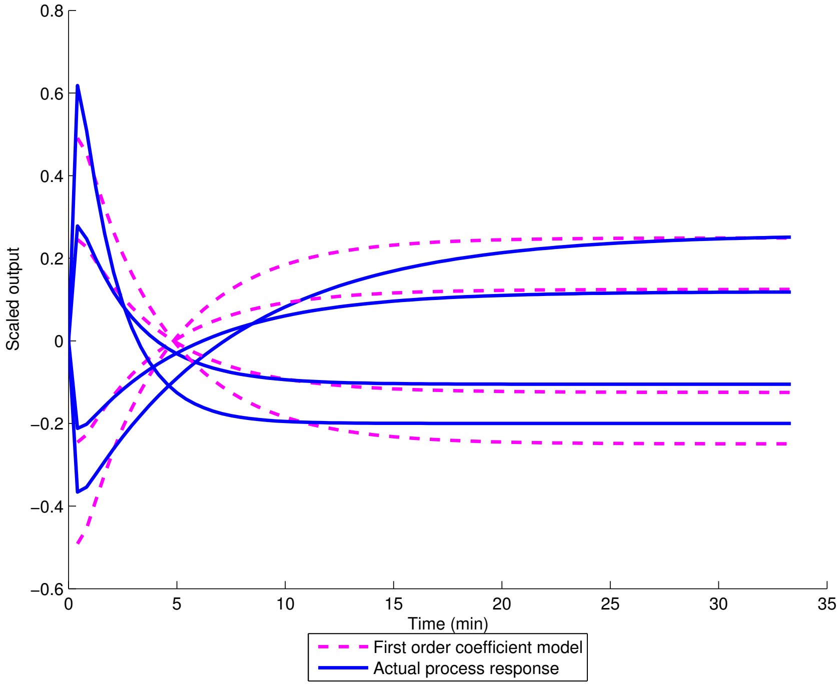

Figure 7.2: Comparison between the actual response of the recovery to steps in the air flow to the rougher bank and the predicted response for a first order Volterra series

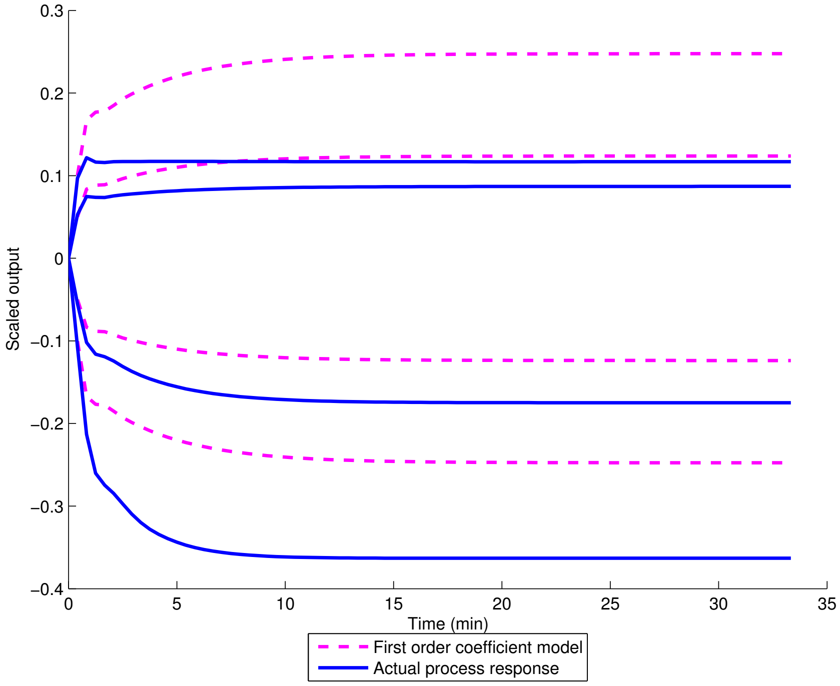

Figure 7.3: Comparison between the actual response of the product grade to steps in the air flow to the cleaner bank and the predicted response for a first order Volterra series

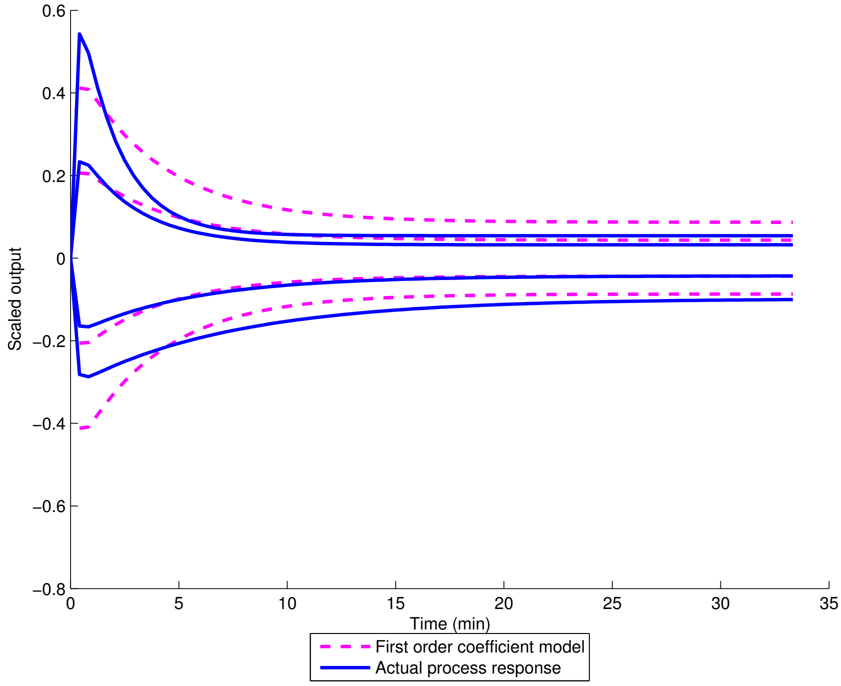

Figure 7.4: Comparison between the actual response of the recovery to steps in the air flow to the cleaner bank and the predicted response for a first order Volterra series

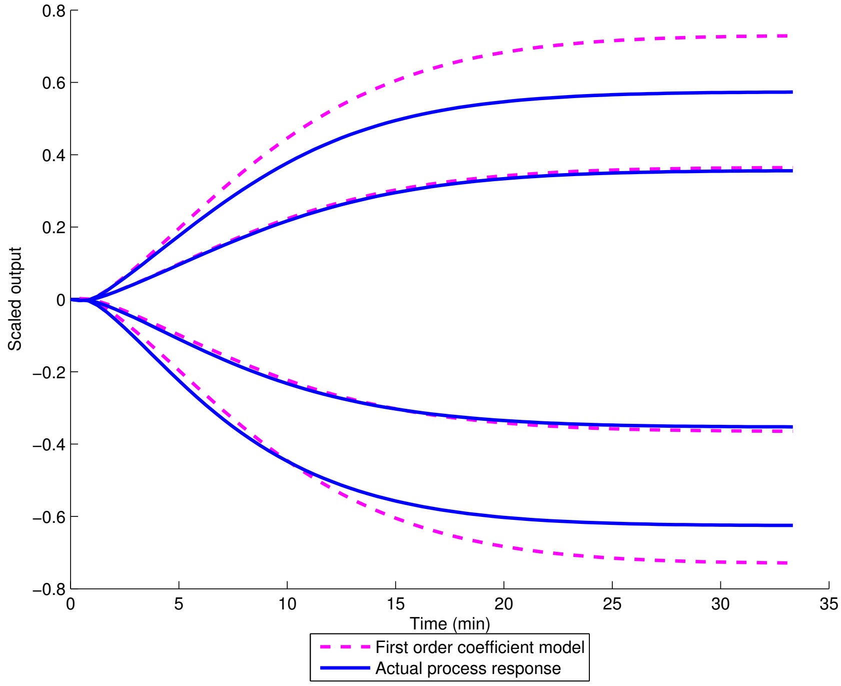

Figure 7.5: Comparison between the actual response of the product grade to steps in the pulp flow from the rougher bank and the predicted response for a first order Volterra series

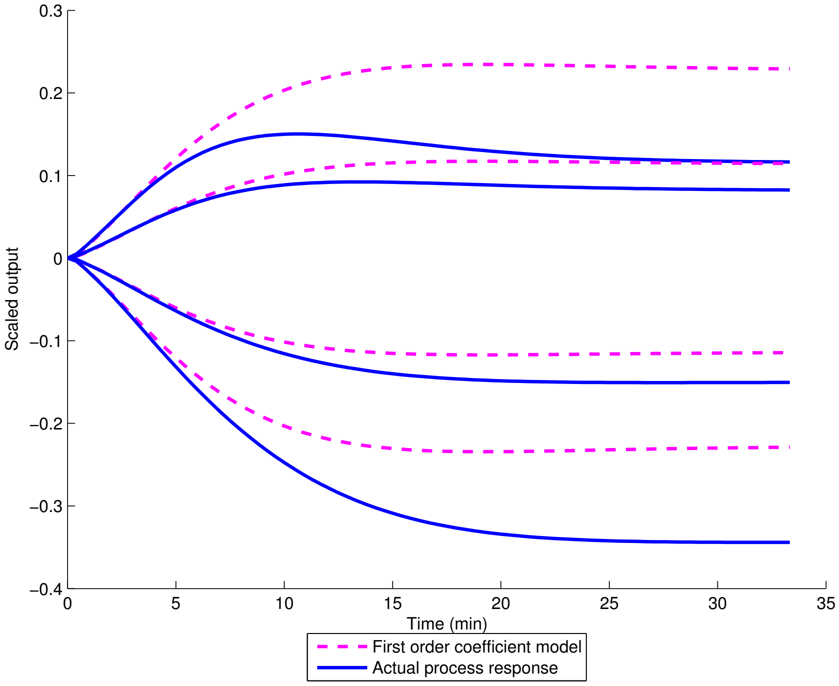

Figure 7.6: Comparison between the actual response of the recovery to steps in the pulp flow from the rougher bank and the predicted response for a first order Volterra series

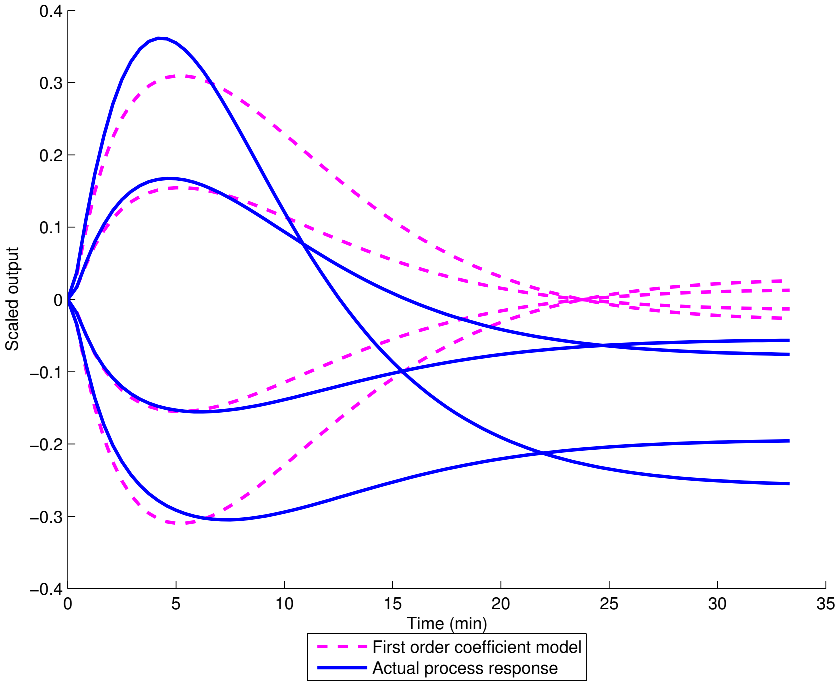

Figure 7.7: Comparison between the actual response of the product grade to steps in the pulp flow from the cleaner bank and the predicted response for a first order Volterra series

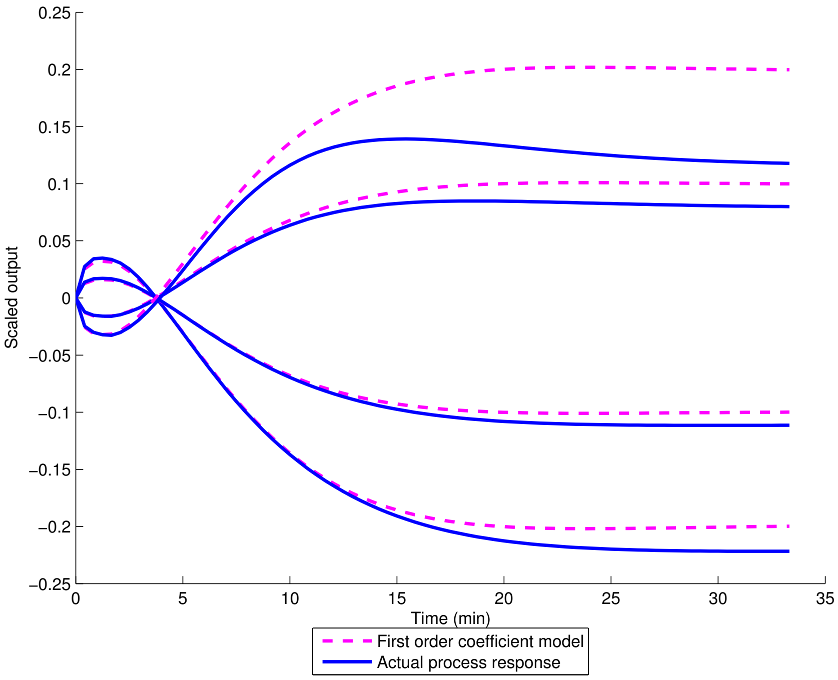

Figure 7.8: Comparison between the actual response of the recovery to steps in the pulp flow from the cleaner bank and the predicted response for a first order Volterra series

ures 7.9 to 7.16. These models contain nonlinear terms and are able to describe some of the system nonlinearity. The performance of the linear and diagonal second order models will be compared in section 7.2.

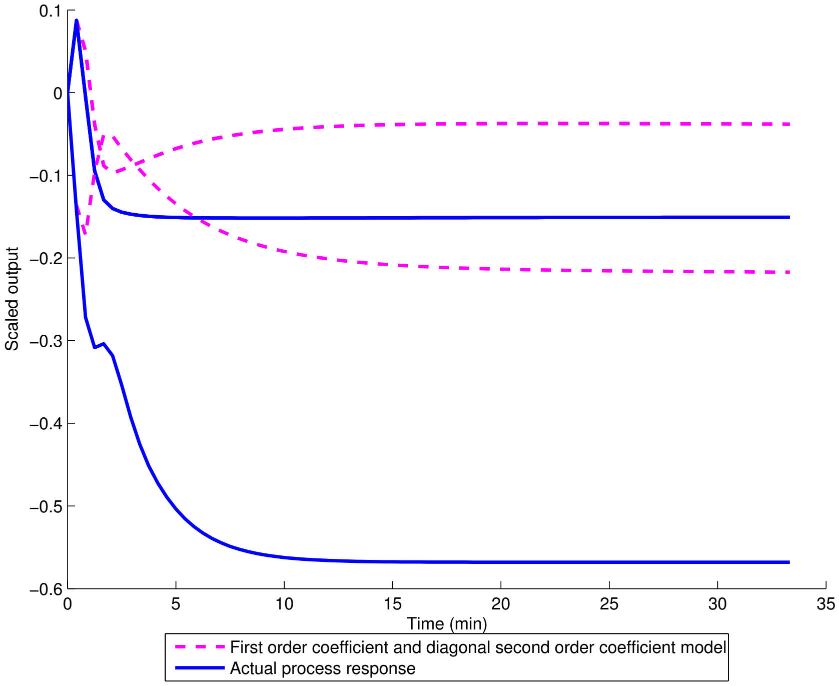

Figure 7.9: Comparison between the actual and response of the product grade to steps in the air flow to the rougher bank and the predicted response for a diagonal second order Volterra series

7.1.4 Identification of the off-diagonal coefficients

The second order off-diagonal coefficients ($h_2(i,j)$ with $i \neq j$) can be calculated with the methods described in section 6.7.2. In this method, the plant is excited with a pseudo random binary input sequence. For this sequence, the contribution of the diagonal and first order coefficients to the process response are minimal (Parker et al., 2001), (Pearson et al., 1996).

The off-diagonal coefficients are calculated from equation 7.2, with the cross bi-correlation t_{yuu} given by equation 7.3 (Koh & Powers, 1985). This correlation requires large data sets to be accurate.

$$h_2(i,j) = \left\{ \begin{array}{ll} \frac{t_{yuu}(i,j)}{2\sigma^4} & i \neq j \end{array} \right. \tag{7.2}$$

$$t_{yuu}(\tau_1, \tau_2) = \frac{1}{N} \sum_{t=1}^{N-\tau_1} u(t)u(t+\tau_1-\tau_2)y(t+\tau_1) \tag{7.3}$$

The off-diagonal second order Volterra coefficients were identified from a data set consisting of 45000 data points. The results are shown in figures 7.17 to 7.24. Despite the large data set used, not all the model has not fully converged. In particular, the steady state behaviour is not described correctly. The number of data points is equivalent to 313 hours of identification per input. These excessive data requirements make the full

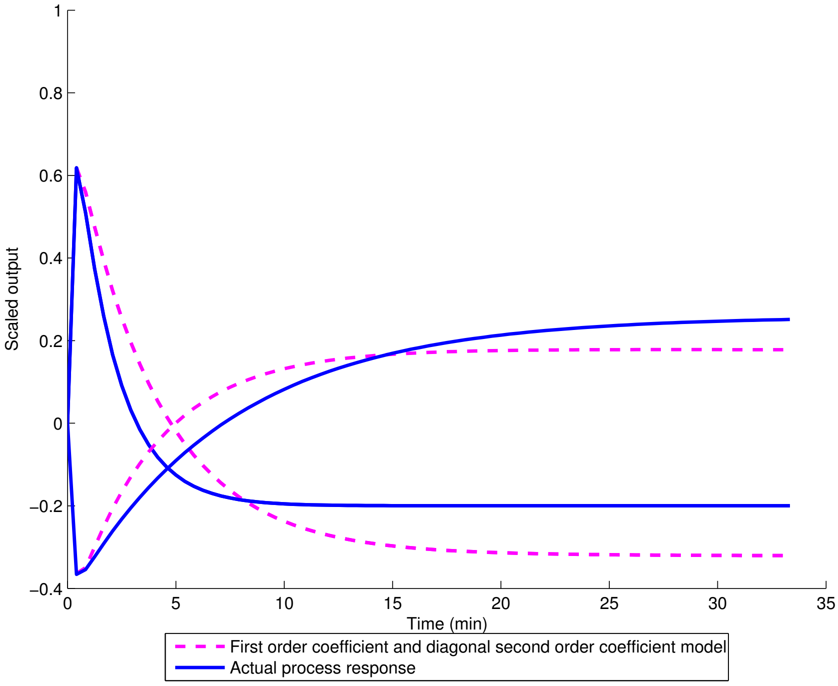

Figure 7.10: Comparison between the actual response of the recovery to steps in the air flow to the rougher bank and the predicted response for a diagonal second order Volterra series

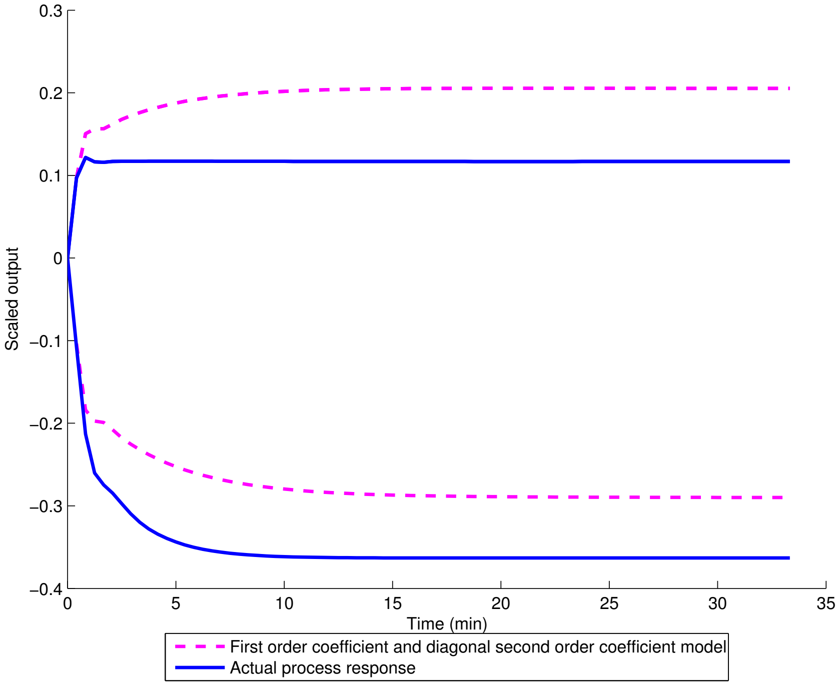

Figure 7.11: Comparison between the actual response of the product grade to steps in the air flow to the cleaner bank and the predicted response for a diagonal second order Volterra series

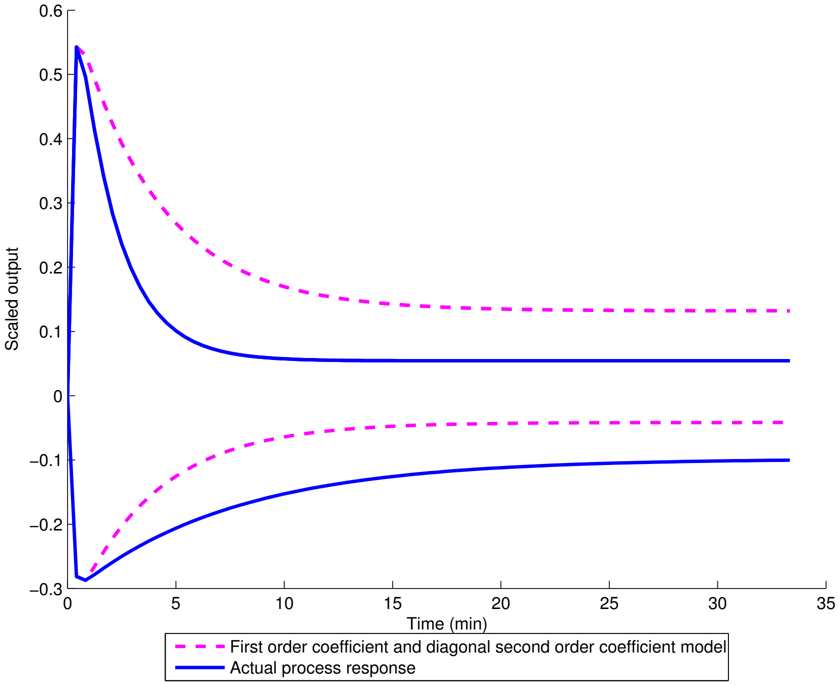

Figure 7.12: Comparison between the actual response of the recovery to steps in the air flow to the cleaner bank and the predicted response for a diagonal second order Volterra series

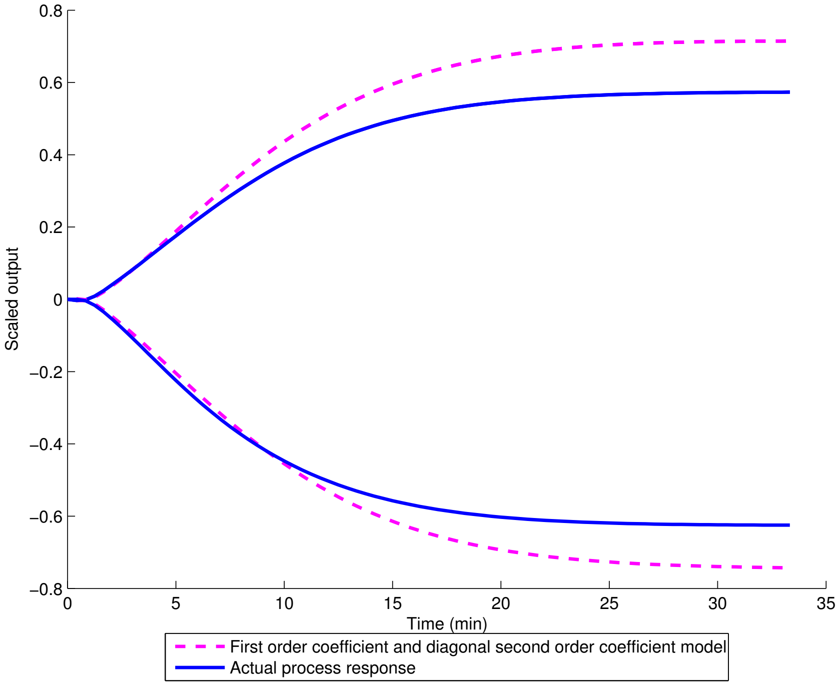

Figure 7.13: Comparison between the actual response of the product grade to steps in the pulp flow from the rougher bank and the predicted response for a diagonal second order Volterra series

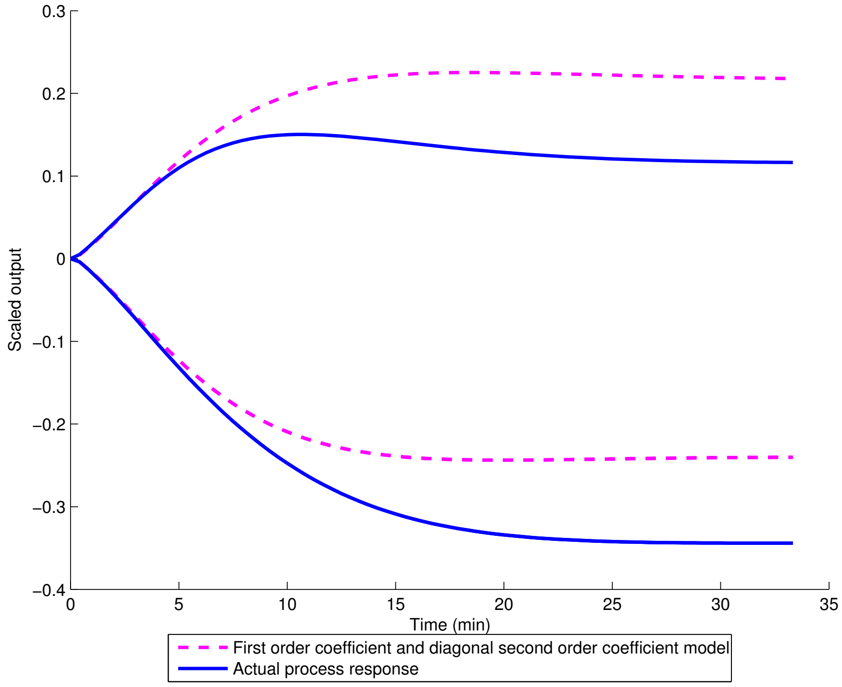

Figure 7.14: Comparison between the actual response of the recovery to steps in the pulp flow from the rougher bank and the predicted response for a diagonal second order Volterra series

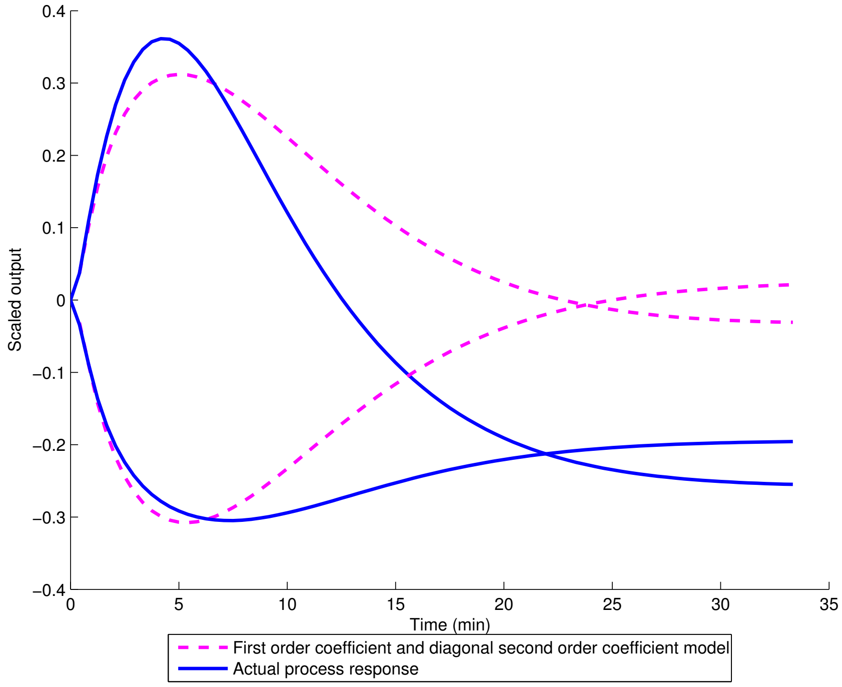

Figure 7.15: Comparison between the actual response of the product grade to steps in the pulp flow from the cleaner bank and the predicted response for a diagonal second order Volterra series

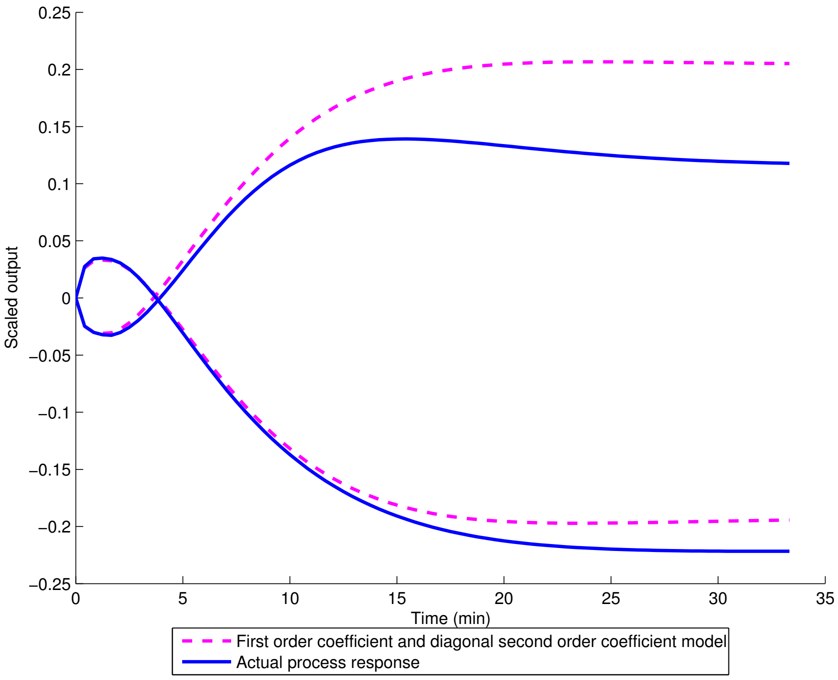

Figure 7.16: Comparison between the actual response of the recovery to steps in the pulp flow from the cleaner bank and the predicted response for a diagonal second order Volterra series

Volterra model impractical. This leads to interest in pruned Volterra models, where some of the second order coefficients are constrained to zero.

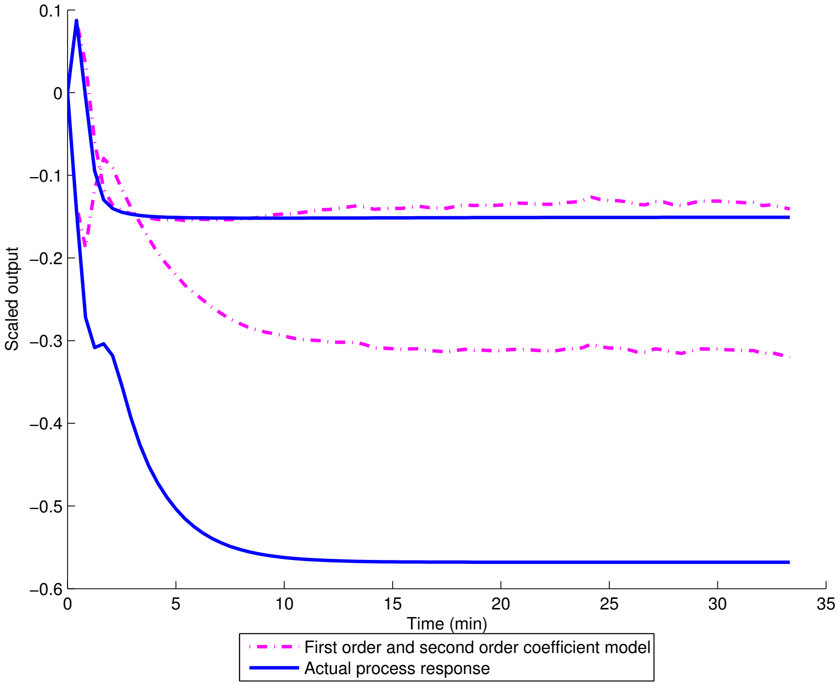

Figure 7.17: Comparison between the actual and response of the product grade to steps in the air flow to the rougher bank and the predicted response for second order Volterra series

7.1.5 Pruned Volterra models

Pearson et al. (1996) gave the following formal definition of a pruned Volterra model. Let $\mathcal{M}$ be the set of integers $\mathcal{M} = \{1, 2, \cdots M\}$ and let $\mathcal{S}$ be a subset of $\mathcal{M}$. A model is pruned with respect to $\mathcal{S}$ if:

$$h_2(i,j) = 0 \qquad \text{if} \quad (i,j) \in \mathcal{S} \tag{7.4}$$

Pearson et al. (1996) proved that, for IID input sequences, constraining some parameters to zero has no influence on the values of the remaining unconstrained parameter estimates. This property was used to prune the second order Volterra models.

The following procedure was used to obtain a pruned Volterra model. Initially, a full Volterra model was identified. One second order coefficient was considered at a time. The predicted response with the coefficient constrained to zero was calculated, and the sum of squared errors and the steady state error was compared to the error values with the coefficient at its calculated value. If the errors are larger for the case where the coefficient is constrained to zero, the coefficient is retained.

The results obtained with pruned models obtained with the procedure described above is shown in figures 7.25 to 7.32. The full second order model used as a starting point was the second order model obtained above.

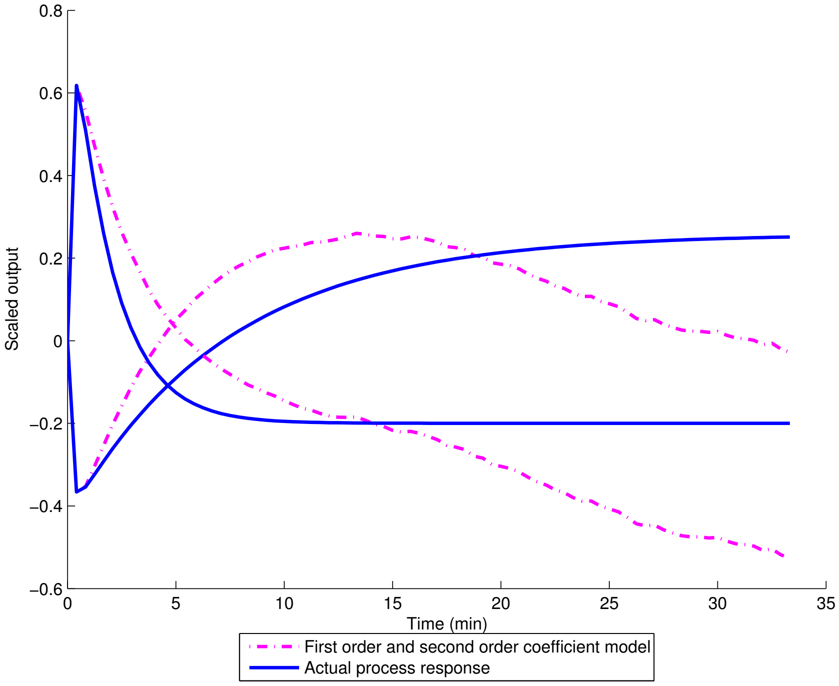

Figure 7.18: Comparison between the actual response of the recovery to steps in the air flow to the rougher bank and the predicted response for a second order Volterra series

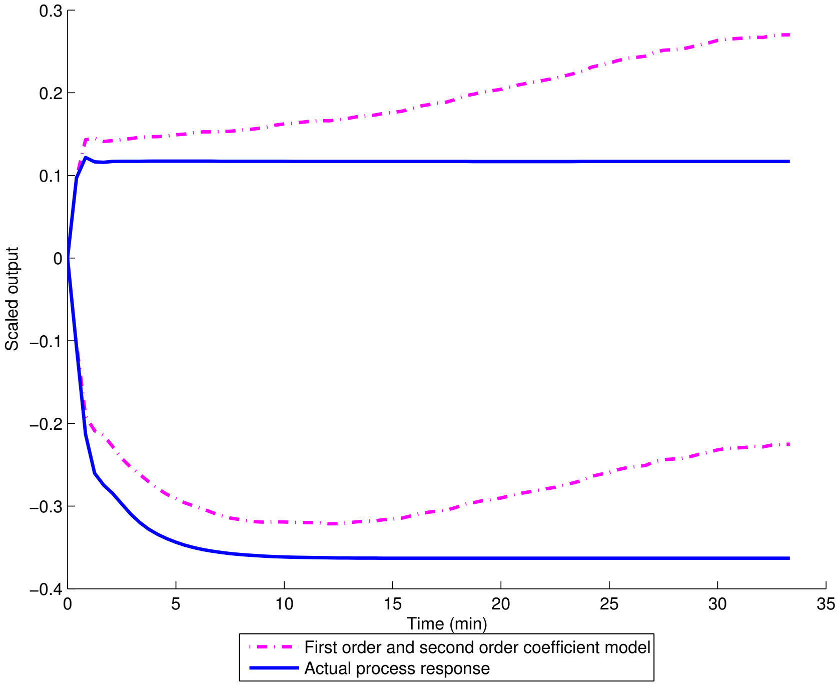

Figure 7.19: Comparison between the actual response of the product grade to steps in the air flow to the cleaner bank and the predicted response for a second order Volterra series

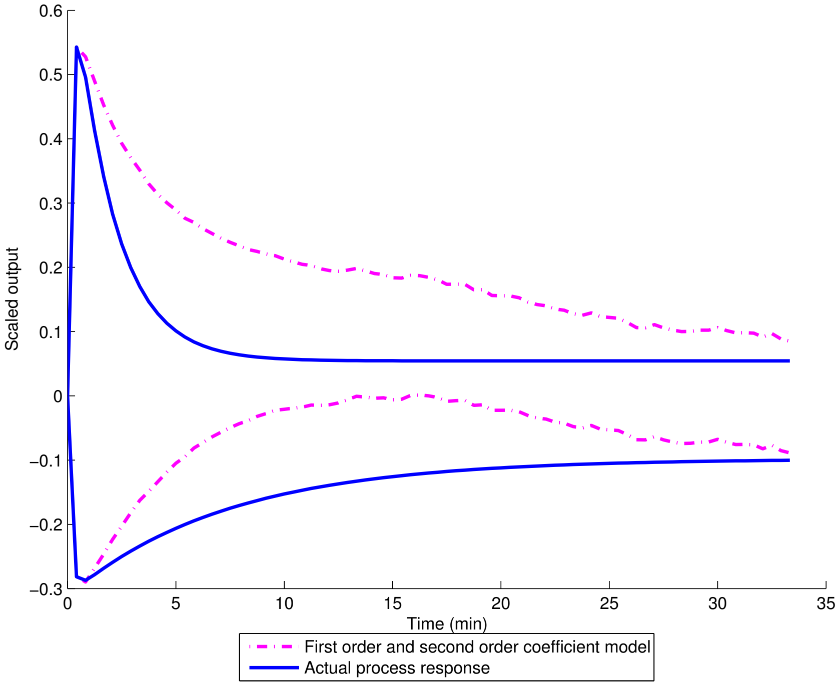

Figure 7.20: Comparison between the actual response of the recovery to steps in the air flow to the cleaner bank and the predicted response for a second order Volterra series

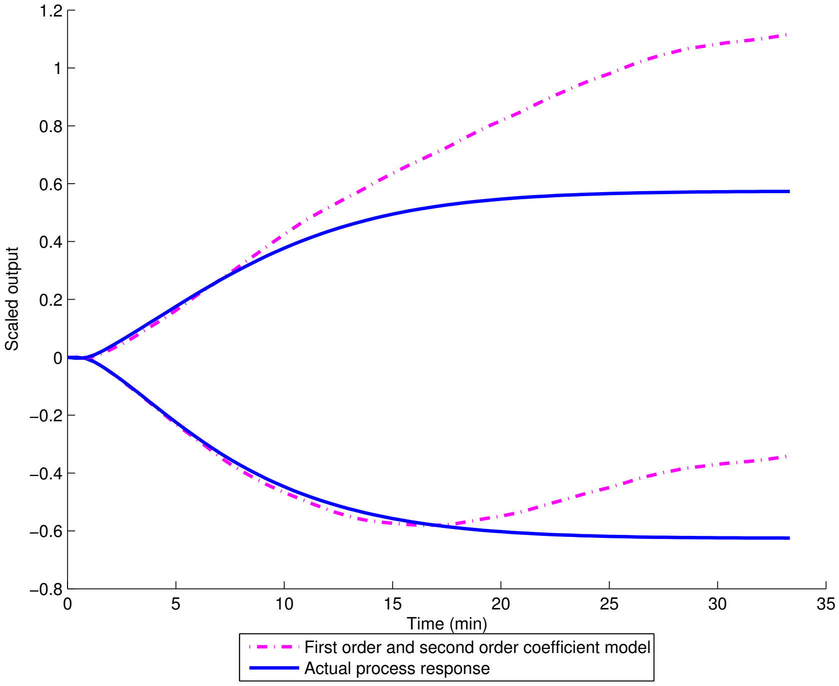

Figure 7.21: Comparison between the actual response of the product grade to steps in the pulp flow from the rougher bank and the predicted response for a second order Volterra series

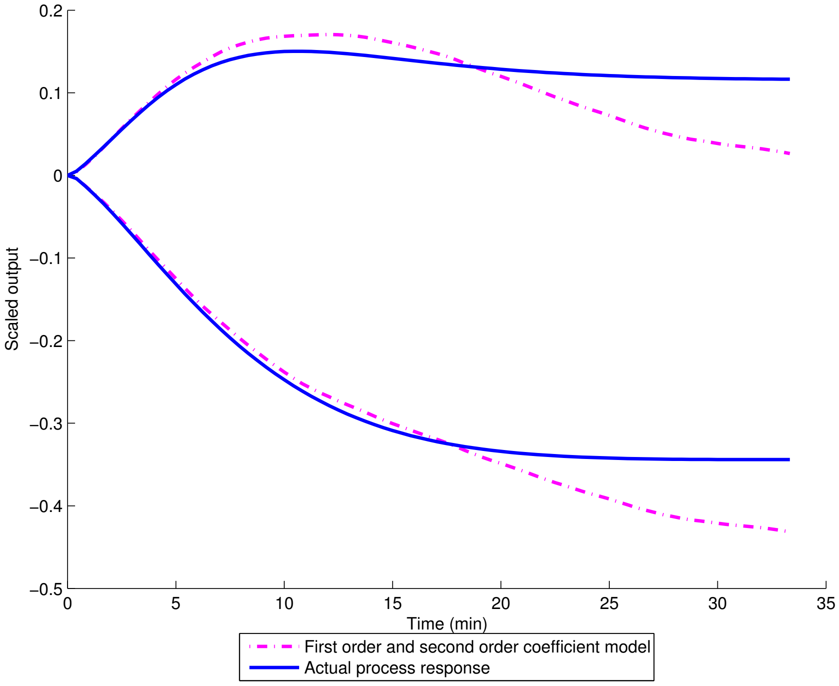

Figure 7.22: Comparison between the actual response of the recovery to steps in the pulp flow from the rougher bank and the predicted response for a second order Volterra series

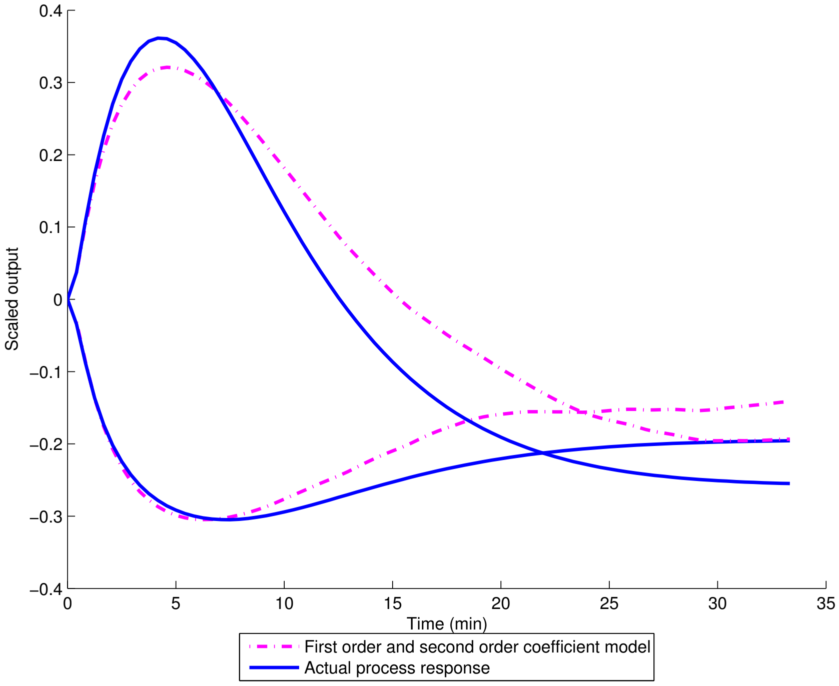

Figure 7.23: Comparison between the actual response of the product grade to steps in the pulp flow from the cleaner bank and the predicted response for a second order Volterra series

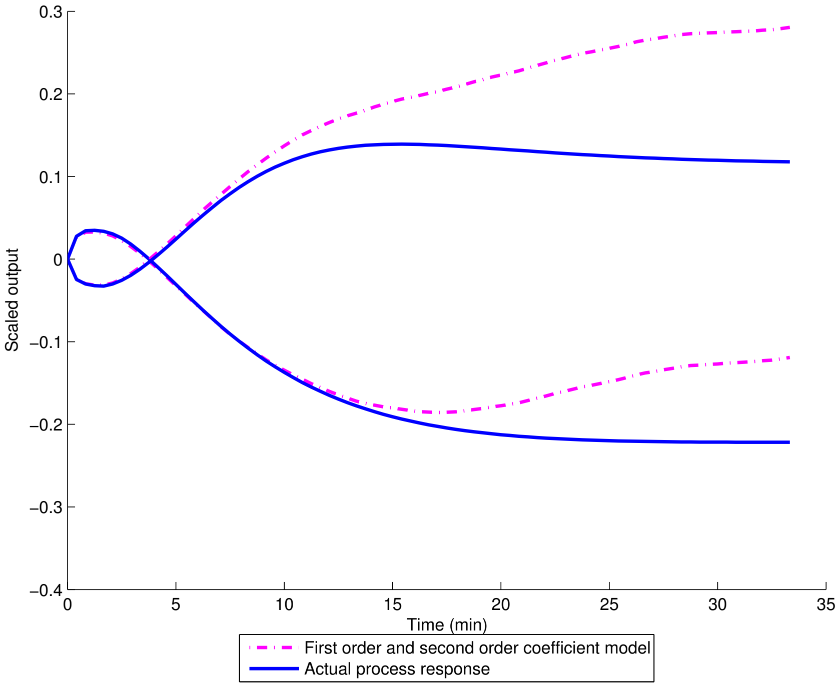

Figure 7.24: Comparison between the actual response of the recovery to steps in the pulp flow from the cleaner bank and the predicted response for a second order Volterra series

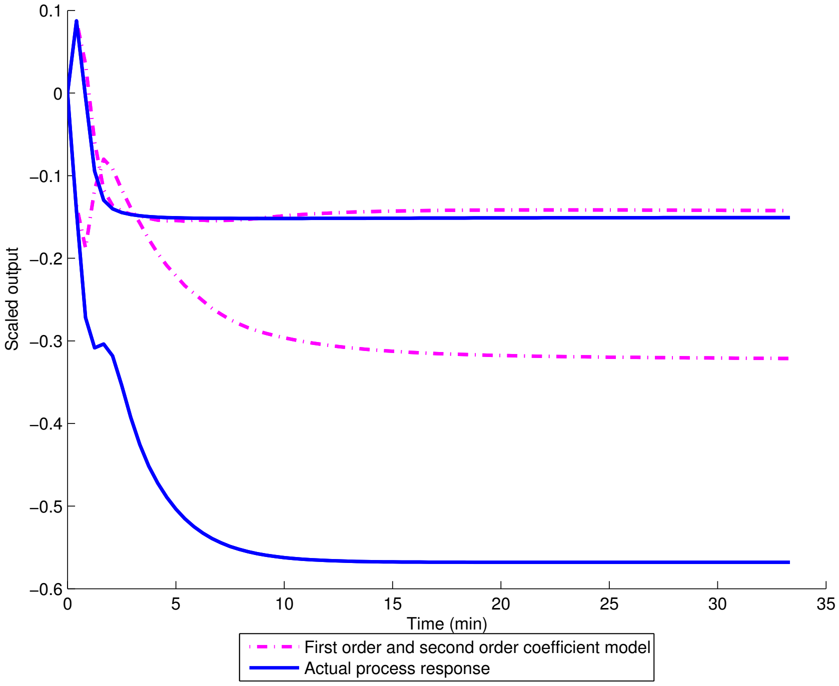

Figure 7.25: Comparison between the actual and response of the product grade to steps in the air flow to the rougher bank and the predicted response for a pruned second order Volterra series obtained from a large data set

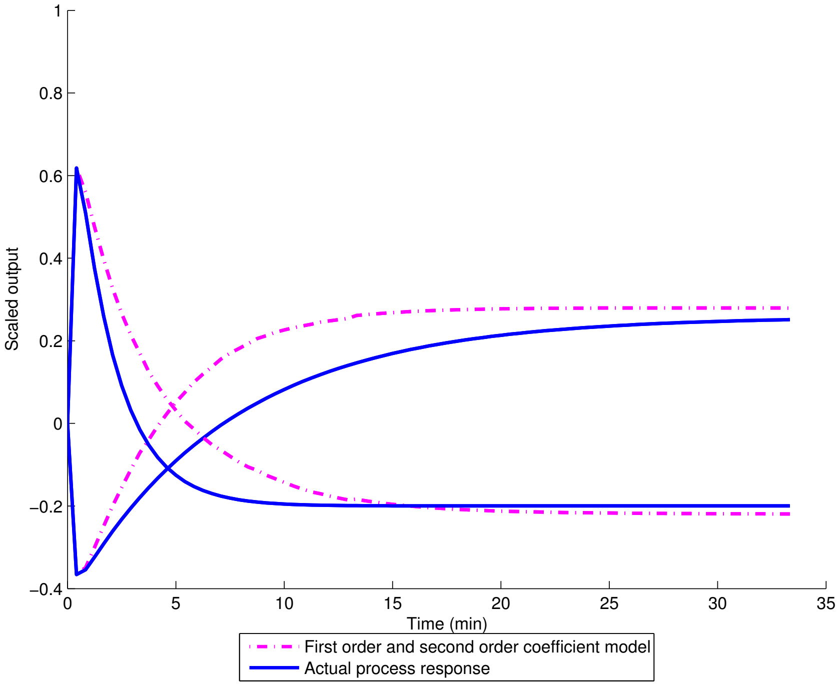

Figure 7.26: Comparison between the actual response of the recovery to steps in the air flow to the rougher bank and the predicted response for a pruned second order Volterra series obtained from a large data set

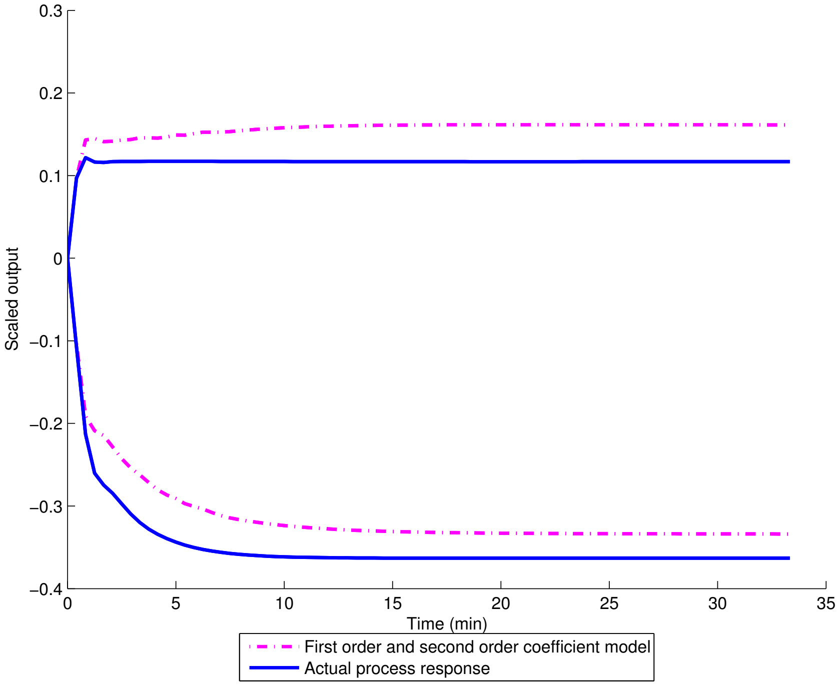

Figure 7.27: Comparison between the actual response of the product grade to steps in the air flow to the cleaner bank and the predicted response for a pruned second order Volterra series obtained from a large data set

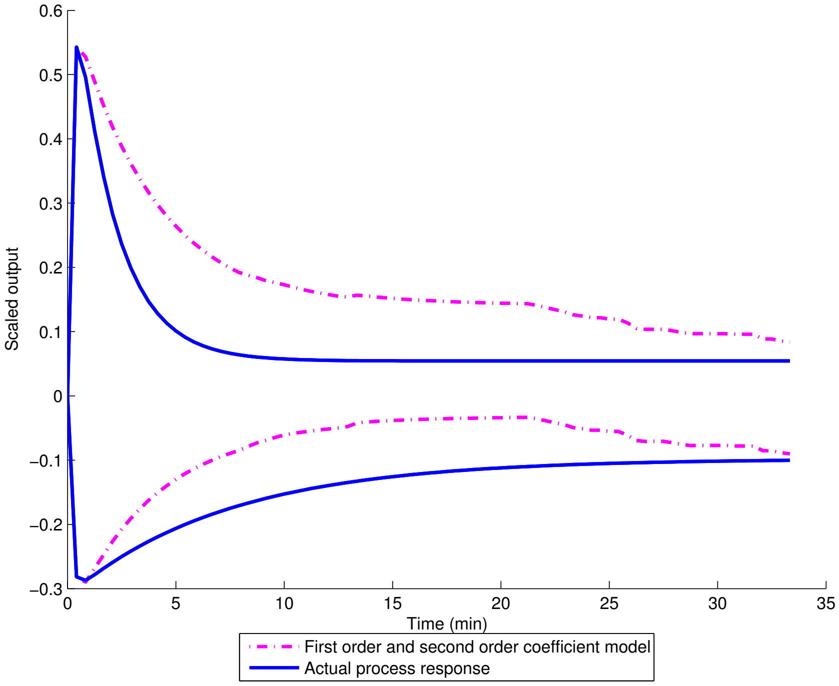

Figure 7.28: Comparison between the actual response of the recovery to steps in the air flow to the cleaner bank and the predicted response for a pruned second order Volterra series obtained from a large data set

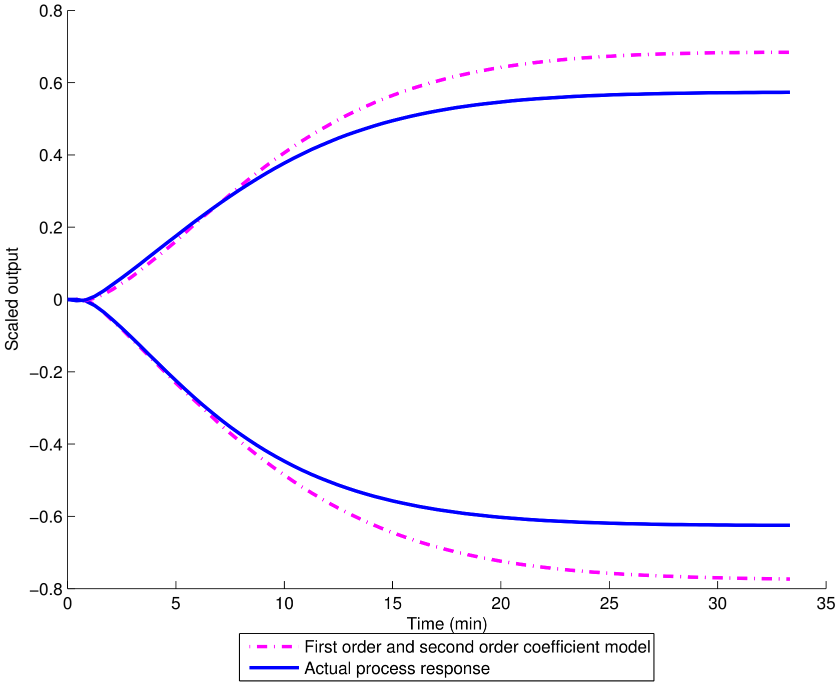

Figure 7.29: Comparison between the actual response of the product grade to steps in the pulp flow from the rougher bank and the predicted response for a pruned second order Volterra series obtained from a large data set

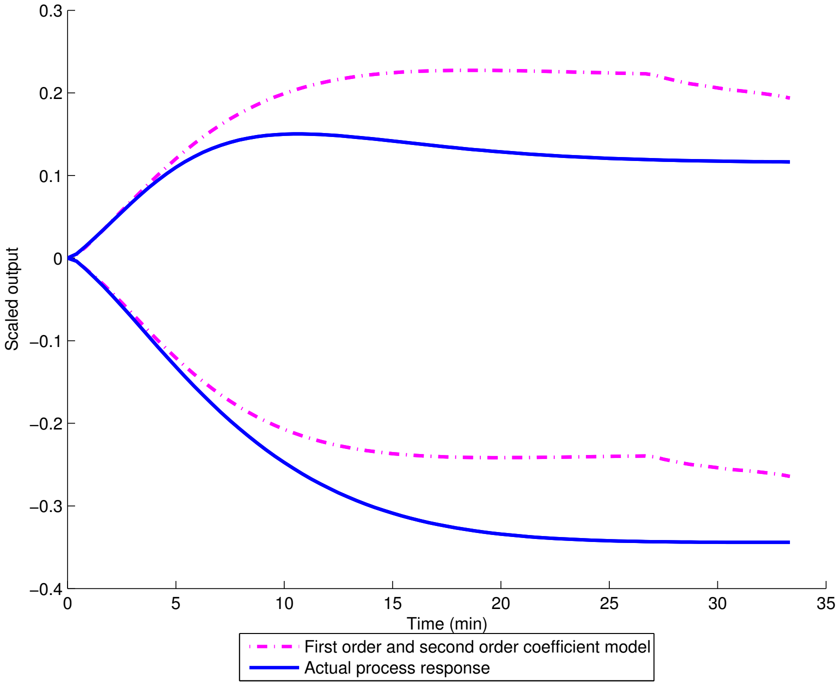

Figure 7.30: Comparison between the actual response of the recovery to steps in the pulp flow from the rougher bank and the predicted response for a pruned second order Volterra series obtained from a large data set

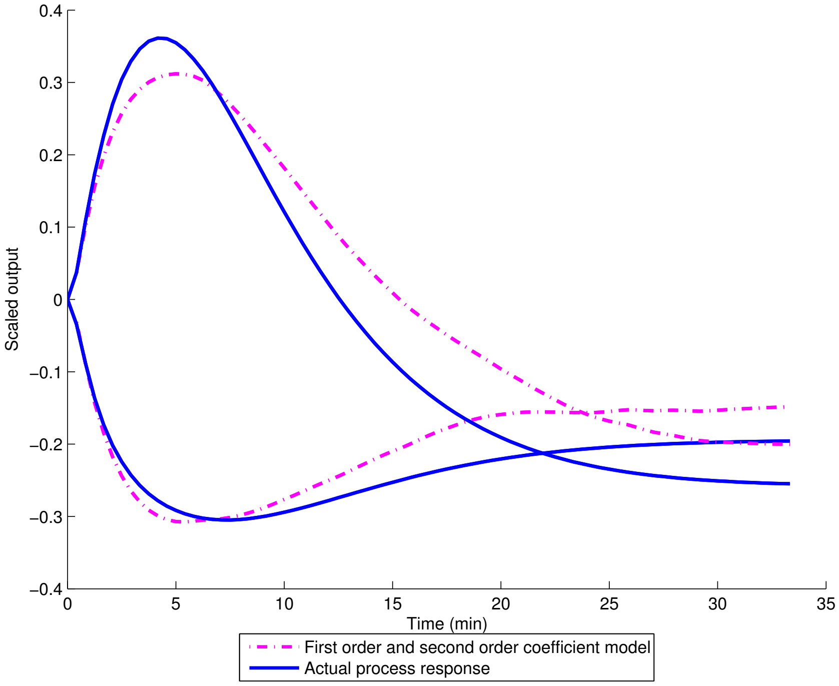

Figure 7.31: Comparison between the actual response of the product grade to steps in the pulp flow from the cleaner bank and the predicted response for a pruned second order Volterra series

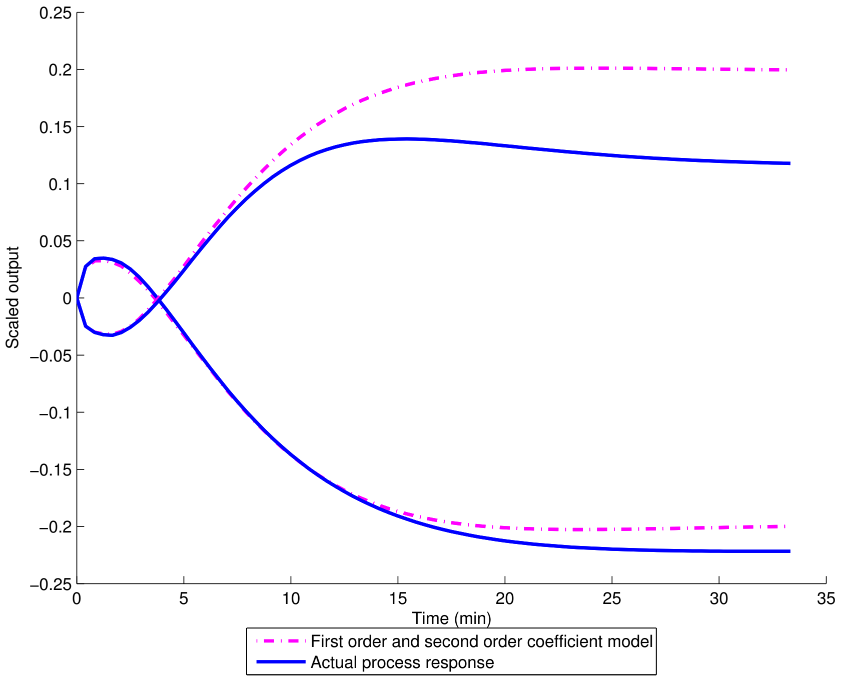

Figure 7.32: Comparison between the actual response of the recovery to steps in the pulp flow from the cleaner bank and the predicted response for a pruned second order Volterra series

7.2 Comparison of results

The sum of squared errors and steady state errors for the first order, diagonal second order and pruned second order Volterra series are shown in figures 7.33 and 7.34. These error values are the average values obtained for steps of -1, -0.5, 0.5 and 1.

The largest improvement in model fit obtained with nonlinear models is for the first input– ouput pair (air flow to the first (rougher) bank and concentrate grade). This response is severely nonlinear, in that the sign of the gain changes. The linear model is unable to describe this phenomenon. The diagonal model is able to describe the gain sign inversion, but exhibits a large steady state offset. The pruned Volterra model reduces this offset as well as the sum of squared errors.

The second largest improvement is for the response between the flow setpoint from the second (rougher) bank and the concentrate grade. This response also exhibits a gain sign change. Both the pruned second order model and the model containing only diagonal second order coefficients were able to predict the sign of the gain correctly.

The differences in the model predictions are less dramatic for the other input– output pairs, but the pruned Volterra model performs better in terms of both steady state offset and sum of squared errors in all cases. The second order diagonal model performs worse than the linear model for the response of both outputs to the air flow to the second (cleaner) bank. For both the outputs the diagonal model is able to predict the size of the peak, but does not settle to the correct steady state value.

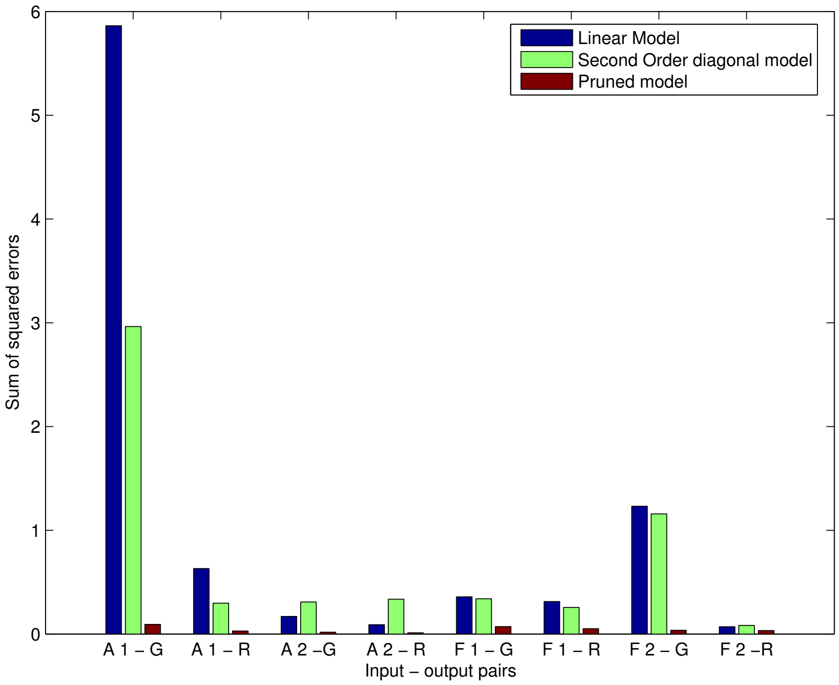

Figure 7.33: The sum of squared errors for the first order, diagonal second order and pruned second order models for all system input output pairs

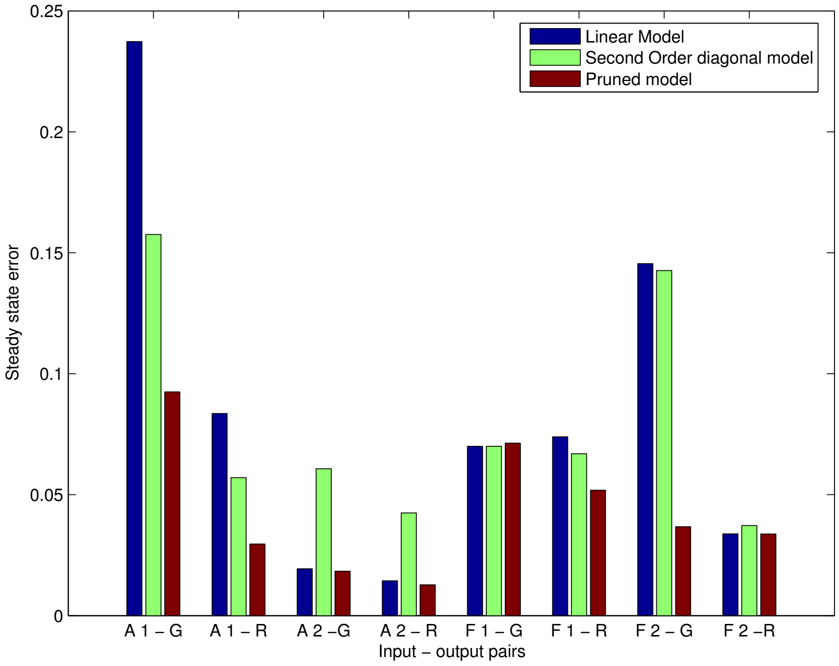

Figure 7.34: The steady state errors for the first order, diagonal second order and pruned second order models for all system input output pairs

While the results shown above were obtained with a large data set, pruned Volterra models make it possible to use less data and still obtain a reasonable improvement in the fit obtained. The data set used for the identification of the full Volterra model used as a starting point was reduced by 25%, 50% and 75% respectively. Figure 7.35 compares the results model obtained from the larger data set with the pruned model obtained from the reduced data sets as well as the second order diagonal model. Even with a data set reduced by 75%, the pruned second models performed better than a second order model containing only diagonal coefficients. The performance obtained with the reduced data sets are comparable to that obtained with the full data set. This shows that pruned Volterra models can be obtained from data sets that are significantly smaller than those required by full Volterra models.

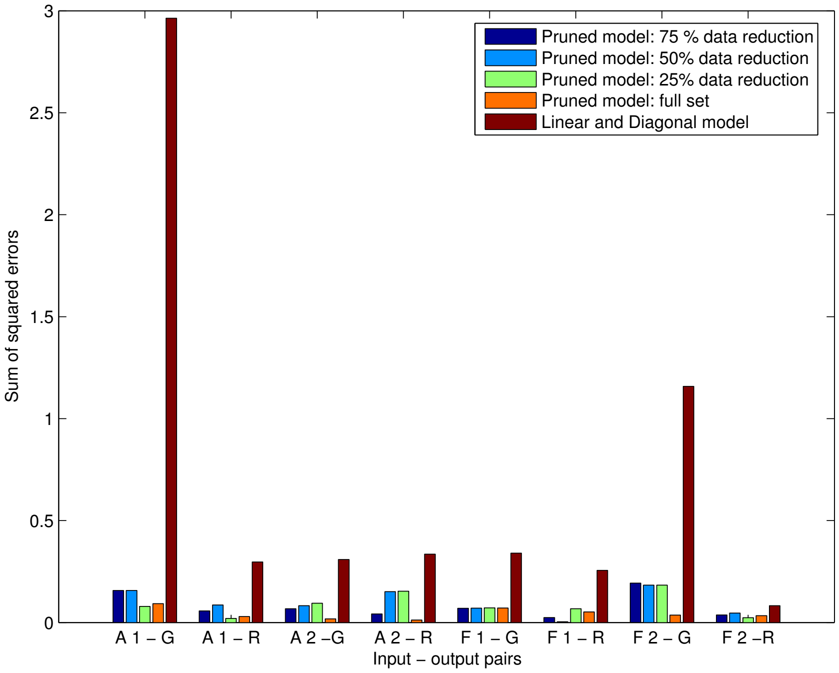

Figure 7.35: Comparison between the sum of squared errors for pruned models obtained with a full data set, a 75%, 50% and 25% reduced data set and the diagonal second order model

Even with a 75 % reduction in the required data points, the data set is still large. For some of the input ouput pairs, the improvement in model predictions is not sufficient to justify such a long testing period, and the linear or diagonal model may be used. For other input – ouput pairs, such as the the air flow to bank 1 – concentrate grade, the improvement in model fit is significant enough to justify the identification of off-diagonal elements.

7.3 Output superposition

For linear systems, the process response to two or more simultaneous inputs is equal to the sum of the responses of the system to the individual inputs (see section 6.2.1). This property does not hold for nonlinear systems.

To model these effects with Volterra series, additional terms will have to be added which will lead to further complications during identification. For this reason, it was assumed that the process response due to two simultaneous inputs can be approximated as the sum of the responses to the inputs acting individually.

CHAPTER 8

Model predictive control

This chapter provides background on model predictive control. Linear model predictive control is discussed first, after which a nonlinear model predictive algorithm based on Volterra models is shown. A matrix formulation for the calculation of the second order terms is presented in section 8.2.4.

8.1 Linear model predictive control

Model predictive control can be defined as ".. a class of control algorithms that utilize an explicit process model to predict the future response of the plant.." (Qin & Badgwell, 2003). Linear model predictive control uses a linear process model to make these predictions. Model predictive control became popular with the publication of papers on dynamic matrix control (DMC) (Cutler et al., 1983) and model predictive heuristic control (Richalet et al., 1978).

8.1.1 Dynamic matrix control

This control algorithm predicts the process response with a step response model, as discussed in section 6.5.2. The predicted process response one step ahead in time is given by equation 8.1 (Cutler et al., 1983).

$$y(k+1) = y_0 + \sum_{i=1}^{k} a(k-i+1)\Delta u(i) \tag{8.1}$$

This equation can now be applied to predict the process response, starting at time k, to CH manipulated variable moves PH intervals into the future, where CH and PH are the control horizon and prediction horizon respectively. The result is given in equation 8.2.

If $i \geq M$, $a(i) = a(MH)$, where MH is the number of step response coefficients (model horizon) (Cutler et al., 1983).

$$y(k + PH) = y_0 + \sum_{i=0}^{k+CH-1} a(k + PH - i)\Delta u(i) \tag{8.2}$$

The prediction includes contributions from past inputs as well as the effect of the predicted inputs. The past and future terms can be separated as shown in equation 8.3 (Cutler et al., 1983).

$$\begin{aligned} y(k + PH) &= y_0 + \sum_{i=0}^{k-1} a(k + PH - i)\Delta u(i) + \sum_{i=k}^{k+CH-1} a(k + PH - i)\Delta u(i) && (8.3) \\ &= y_0 + y_{\text{past}} + \sum_{i=k}^{k+N-1} \Delta u(i) && (8.4) \end{aligned}$$

This series sum can be used to calculate the predicted process output, as illustrated below. This leads to the matrix equation shown in 8.6.

$$\begin{aligned} y(k+1) &= y_{\text{past}}(k+1) + a(1)\Delta u(k) \\ y(k+2) &= y_{\text{past}}(k+2) + a(2)\Delta u(k) + a(1)\Delta u(k+1) \\ \vdots &= \quad \vdots \\ y(k+CH) &= y_{\text{past}}(k+CH) + a(CH)\Delta u(k) + a(CH-1)\Delta u(k+1) + \cdots + a(1)\Delta u(k+CH-1) \\ y(k+MH) &= y_{\text{past}}(k+MH) + a(MH)\Delta u(k) + a(MH-1)\Delta u(k+1) + \cdots \\ &\quad + a(MH-CH+1)\Delta u(k+CH-1) \end{aligned} \tag{8.5}$$

$$\begin{bmatrix} y(k+1) \\ y(k+2) \\ \vdots \\ y(k+MH) \end{bmatrix} = \begin{bmatrix} y_{\text{past}}(k+1) \\ y_{\text{past}}(k+2) \\ \vdots \\ y_{\text{past}}(k+MH) \end{bmatrix} + A \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+CH-1) \end{bmatrix} \tag{8.6}$$

$$A = \begin{bmatrix} a(1) & 0 & 0 & \dots & 0 \\ a(2) & a(1) & 0 & \dots & 0 \\ a(3) & a(2) & a(1) & \dots & 0 \\ \vdots & & & \ddots & \vdots \\ a(PH) & a(PH-1) & \cdots & & a(PH-CH) \end{bmatrix} \tag{8.7}$$

The matrix A is called the Dynamic Matrix of the process. The contribution of unmeasured disturbances as well as model inaccuracies can be estimated based on the difference between the actual and predicted process output. It is usually assumed that the modelling error is equal to the present value of the error.

$$d(k) = y(k)_{\text{measured}} - (y_0 + y(k)_{\text{past}}) \tag{8.8}$$

Equation 8.6 can be rearranged into equation 8.9. The difference between the set-point and the predicted output due to manipulated variable moves is written as E^f in equation 8.10. E^f is the error that would occur if no future adjustments are made to the manipulated variable. An objective function for the model predictive controller is given in equation 8.11. The deviations from set point can be weighed by introducing weighing factors into the objective function, shown in equation 8.13. The manipulated variable moves can also be penalised by introducing manipulated variable weights (W_{MV}).

$$A \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+U-1) \end{bmatrix} = \begin{bmatrix} y_{\text{set}} & - & y_{\text{past}}(k+1) \\ y_{\text{set}} & - & y_{\text{past}}(k+2) \\ \vdots & & \vdots \\ y_{\text{set}} & - & y_{\text{past}}(k+M) \end{bmatrix} \tag{8.9}$$

$$A\Delta u = E^f \tag{8.10}$$

$$J_f = \sum_{i=1}^{V} [y_{\text{set}} - y_{\text{past}} - A\Delta u]^2 \tag{8.11}$$

$$J_f = \sum_{i=1}^{V} \left[E^f - A\Delta u\right] \tag{8.12}$$

$$J_f = \sum_{i=1}^{V} \left[W_{CV}(E^f - A\Delta u)^2\right] + \sum_{i=1}^{M} W_{MV}\left(\Delta u\right) \tag{8.13}$$

Equation 8.13 can be written in terms of matrix operations, as shown in equation 8.14.

$$J_f = \frac{1}{2}\left(A\Delta u - E^f\right)^T W^T W \left(A\Delta u - E^f\right) \tag{8.14}$$

The solution to the least squares problem in the absence of constraints is then given by differentiation of equation 8.14, leading to equation 8.15. If constraints are added, the solution must be found with nonlinear programming techniques.

$$\Delta u = \left(A^T W_{CV}^T W_{CV} A + W_{MV}\right)^{-1} A^T W_{CV}^T W_{CV} E^f \tag{8.15}$$

8.1.2 QDMC

Quadratic dynamic matrix control uses the basic formulation of the dynamic matrix controller (DMC) combined with quadratic programming to solve a constrained multivariable control problem. In other words, the controller finds the least squares solution to equation 8.11 subject to a set of linear constraints (Cutler et al., 1983).

8.1.3 Model predictive heuristic control

Model predictive heuristic control (Richalet et al., 1978) was another one of the earliest applications of model predictive control. In this algorithm, the difference between a reference trajectory and the predicted process response is minimized. An example of a simple first order reference trajectory is shown in equation 8.16 and illustrated in figure 8.1.

$$y_{\text{reference}}(n+i) = \lambda y_{\text{reference}}(n+i-1) + (1-\lambda)y_{\text{set}} \tag{8.16}$$

$$y_{\text{reference}}(n) = y(0) \tag{8.17}$$

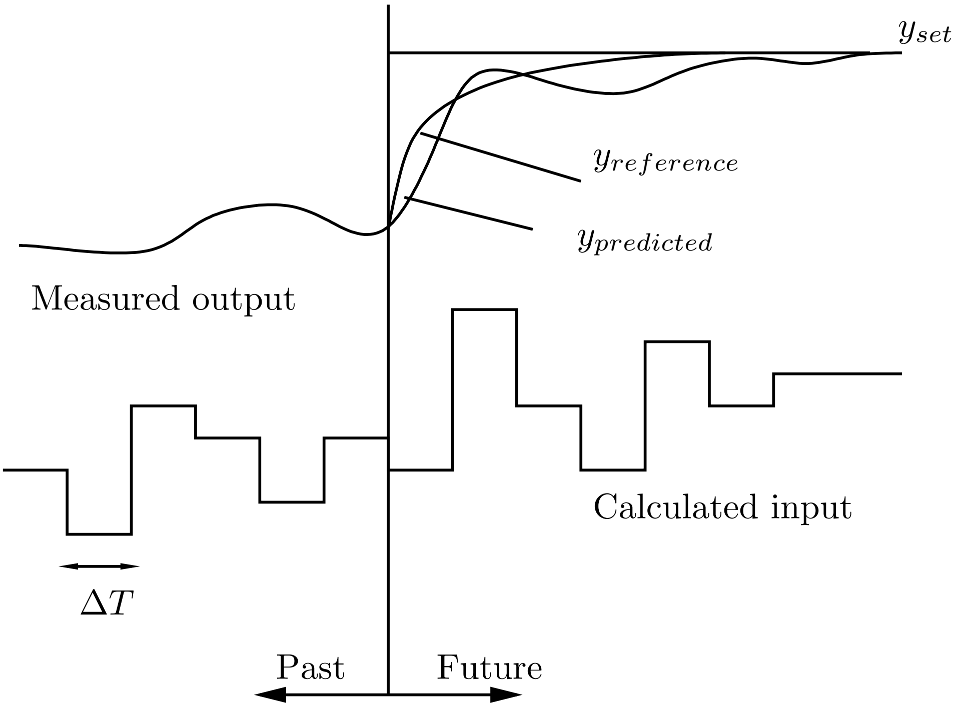

Figure 8.1: The use of a reference trajectory in model predictive heuristic control

8.2 Nonlinear model predictive control

Nonlinear model predictive control uses a nonlinear dynamic model for the prediction and optimisation where linear predictive control uses a linear model. Model predictive control is usually applied either to highly nonlinear processes that operate around a fixed operating point or to moderately nonlinear processes that operate over a wide operating range (Henson, 1998).

A wide variety of models have been used for nonlinear model predictive algorithms. These include:

- fundamental nonlinear differential equations; (Sistu et al., 1993)
- Hammerstein and Wiener models;
- Volterra models; Kashiwagi & Li (2004), (Maner et al., 1996)
- polynomial ARMAX models and
- neural networks.

While fundamental models are valid over a very wide operating region, these models tend to be impractical for application in an industrial environment. The other model types can be obtained from plant data, which is more practical in an industrial environment.

8.2.1 Volterra series nonlinear model predictive control

Maner et al. (1996) proposed a nonlinear model predictive control scheme based on second order Volterra series. This control scheme is attractive since it can be composed into a linear model predictive controller with additional nonlinear terms. This formulation retains the essence of the widely accepted MPC algorithm while promising improved control for nonlinear systems.

The block diagram for the Volterra series nonlinear model predictive controller is shown in figure 8.2. F_{sp} is a setpoint filter used to implement the setpoint trajectory. $\tilde{P}_1$ indicates the linear part of the model, while $\tilde{P}_2$ indicates the nonlinear part of the model. The open loop model prediction for the SISO case is shown in equation 8.18. The term c is expanded in equation 8.19.

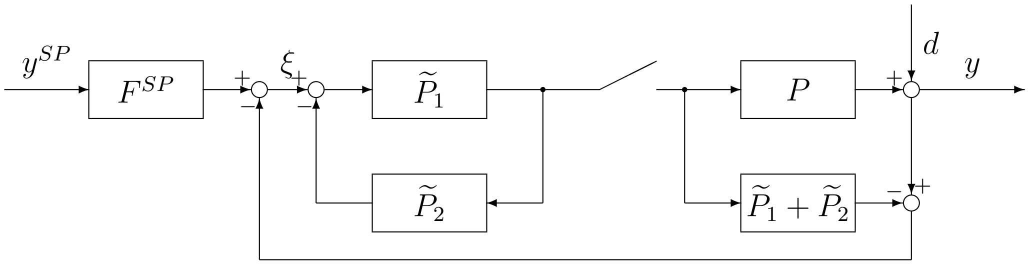

Figure 8.2: Block diagram for a second order Volterra Model predictive controller Maner et al. (1996)

$$\begin{bmatrix} y(k) \\ y(k+2) \\ \vdots \\ y(k+PH) \end{bmatrix} = \begin{bmatrix} h_1(1) & 0 & \cdots & 0 \\ h_1(2) & h_1(1) & \ddots & 0 \\ \vdots & & \ddots & \vdots \\ h_1(PH) & h_2(PH-1) & \cdots & h_1(PH-CH) \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+PH-1) \end{bmatrix} + \begin{bmatrix} c(k+1) \\ c(k+2) \\ \vdots \\ c(k+PH) \end{bmatrix} + \begin{bmatrix} f(k+1) \\ f(k+2) \\ \vdots \\ f(k+PH) \end{bmatrix} \tag{8.18}$$

$$\begin{bmatrix} c(k+1) \\ c(k+2) \\ \vdots \\ c(k+PH) \end{bmatrix} = \begin{bmatrix} h_1(2) & h_1(3) & \cdots & \cdots & h_1(MH) & 0 \\ h_1(3) & h_1(4) & \cdots & h_1(MH-1) & 0 & 0 \\ \vdots & & \ddots & & & \vdots \\ h_1(PH-1) & h_1(PH) & 0 & \ddots & & \vdots \\ h_1(PH) & 0 & 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u(k-1) \\ u(k-2) \\ \vdots \\ u(k-CH) \end{bmatrix} + \begin{bmatrix} d(k+1) \\ d(k+1) \\ \vdots \\ d(k+1) \end{bmatrix} + \begin{bmatrix} g(k+1) \\ g(k+2) \\ \vdots \\ g(k+PH) \end{bmatrix} \tag{8.19}$$

Equations 8.18 and 8.19 may also be written as 8.20 and 8.21. The terms f and g represent the contribution of the second order Volterra coefficients. The calculation of these terms will be discussed in section 8.2.4. $d(k+1)$ is the difference between the measured controlled variable value and the predicted value. This term introduces feedback into the prediction, and compensates for modelling inaccuracies. This SISO formulation can easily be extended to MIMO systems, as shown in equations 8.22 and 8.23.

$$\boldsymbol{y} = \boldsymbol{G}\boldsymbol{u} + \boldsymbol{c} + \boldsymbol{f} \tag{8.20}$$

$$\boldsymbol{c} = \boldsymbol{H}\boldsymbol{u}_{\text{past}} + \boldsymbol{d} + \boldsymbol{g} \tag{8.21}$$

$$\begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \cdots \\ \boldsymbol{y}_{n_{\text{CV}}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{G}_{11} & \boldsymbol{G}_{12} & \cdots & \boldsymbol{G}_{1n_{\text{MV}}} \\ \boldsymbol{G}_{21} & \boldsymbol{G}_{22} & \cdots & \boldsymbol{G}_{2n_{\text{MV}}} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{G}_{n_{\text{CV}1}} & \boldsymbol{G}_{n_{\text{CV}2}} & \cdots & \boldsymbol{G}_{n_{\text{CV}}n_{\text{MV}}} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \vdots \\ \boldsymbol{u}_{n_{\text{MV}}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \\ \vdots \\ \boldsymbol{c}_{n_{\text{CV}}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \\ \vdots \\ \boldsymbol{f}_{n_{\text{CV}}} \end{bmatrix} \tag{8.22}$$

$$\begin{aligned} \begin{bmatrix} \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \\ \vdots \\ \boldsymbol{c}_{n_{\text{CV}}} \end{bmatrix} &= \begin{bmatrix} \boldsymbol{H}_{11} & \boldsymbol{H}_{12} & \cdots & \boldsymbol{H}_{1n_{\text{MV}}} \\ \boldsymbol{H}_{21} & \boldsymbol{H}_{22} & \cdots & \boldsymbol{H}_{2n_{\text{MV}}} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{H}_{n_{\text{CV}1}} & \boldsymbol{H}_{n_{\text{CV}2}} & \cdots & \boldsymbol{H}_{n_{\text{CV}}}n_{\text{MV}} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_{\text{past},1} \\ \boldsymbol{u}_{\text{past},2} \\ \vdots \\ \boldsymbol{u}_{\text{past},n_{\text{MV}}} \end{bmatrix} \\ &+ \begin{bmatrix} \boldsymbol{d}_1 \\ \boldsymbol{d}_2 \\ \vdots \\ \boldsymbol{d}_{n_{\text{MV}}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{g}_1 \\ \boldsymbol{g}_2 \\ \vdots \\ \boldsymbol{g}_{n_{\text{MV}}} \end{bmatrix} \end{aligned} \tag{8.23}$$

8.2.2 Control algorithm

The objective function for the controller is given by equation 8.24. This equation can be differentiated with respect to the manipulated variable u to find the manipulated variable values that will minimize the sum of the squared error over the prediction horizon. The calculation is complicated by the fact that f is also a function of u. Maner et al. (1996) assumed that f was independent of u for the differentiation of 8.24. The controller moves are then calculated from the following iterative procedure.

$$\begin{aligned} J_f &= (\boldsymbol{s} - \boldsymbol{y})^T (\boldsymbol{s} - \boldsymbol{y}) && (8.24) \\ &= (\boldsymbol{s} - (\boldsymbol{G}\boldsymbol{u} + \boldsymbol{c} + \boldsymbol{f}))^T (\boldsymbol{s} - (\boldsymbol{G}\boldsymbol{u} + \boldsymbol{c} + \boldsymbol{f})) && (8.25) \end{aligned}$$

The manipulated variable values are calculated from equation 8.26, which was obtained from equation 8.24 with the assumption that $\boldsymbol{f}$ is not a function of the manipulated variables. For the first iteration, $\boldsymbol{f}$ is calculated from assumed manipulated variable values.

$$\boldsymbol{u} = \left(\boldsymbol{G}^T\boldsymbol{G}\right)^{-1}\left((\boldsymbol{s}-\boldsymbol{c}-\boldsymbol{f})^T G\right)^T \tag{8.26}$$

The difference between the manipulated variable values at the previous iteration and the current iteration is calculated, and if the error is below a specified tolerance, the manipulated variable values are implemented. If the error is too large, new values of $\boldsymbol{f}$ are calculated with the calculated $\boldsymbol{u}$ values and the procedure are repeated.

This formulation will lead to an unstable controller if G^TG is singular or near singular. In the DMC controller algorithm, this situation can be handled by penalising the manipulated variable changes. In this case, the controller is formulated in terms of the manipulated variable values, and penalising the manipulated variable values leads to a steady state offset. If the manipulated variable changes are penalised in the objective function, an analytic solution is no longer possible.

8.2.3 Nonlinear programming

It is also possible to minimize equation 8.27 subject to the constraints given in equations 8.29 and 8.30 to obtain the manipulated variables (Maner et al., 1996).

$$\begin{aligned} J_f &= (\boldsymbol{s}-\boldsymbol{y})^T\boldsymbol{W}_{\mathrm{CV}}(\boldsymbol{s}-\boldsymbol{y}) + W_{\mathrm{MV}}\Delta\boldsymbol{u} && (8.27)\\ &= (\boldsymbol{s}-(\boldsymbol{Gu}+\boldsymbol{c}+\boldsymbol{f}))^T W_{\mathrm{CV}}(\boldsymbol{s}-(\boldsymbol{Gu}+\boldsymbol{c}+\boldsymbol{f})) + W_{\mathrm{MV}}\Delta\boldsymbol{u} && (8.28)\end{aligned}$$

$$\boldsymbol{u}_{\mathrm{low}} \leq \boldsymbol{u} \leq \boldsymbol{u}_{\mathrm{high}} \tag{8.29}$$

$$\Delta\boldsymbol{u}_{\mathrm{low}} \leq \Delta\boldsymbol{u} \leq \Delta\boldsymbol{u}_{\mathrm{high}} \tag{8.30}$$

8.2.4 Calculation of the second order contributions

For the SISO case, the contribution of the second order coefficients can be calculated as follows (Maner et al., 1996). For each input – output pair, the matrix B_{oi} is defined, where o is the output and i is the input. The matrix contains the second order Volterra coefficients. The calculation procedure is illustrated for the case where the model horizon

MH is 4, the prediction horizon PH is 3 and the control horizon CH is 2.

$$B_{oi} = \begin{bmatrix} h_2(1,1) & h_2(1,2) & h_2(1,3) & h_2(1,4) \\ 0 & h_2(2,2) & h_(2,3) & h_2(2,4) \\ 0 & 0 & h_2(3,3) & h_2(3,4) \\ 0 & 0 & 0 & h_2(4,4) \end{bmatrix} \tag{8.31}$$

$$f(k+1) = \begin{bmatrix} u(k) & 0 & 0 & 0 \end{bmatrix} B \begin{bmatrix} u(k) & u(k-1) & u(k-2) & u(k-3) \end{bmatrix}^T \tag{8.32}$$

$$f(k+2) = \begin{bmatrix} u(k+1) & u(k) & 0 & 0 \end{bmatrix} B \begin{bmatrix} u(k+1) & u(k) & u(k-1) & u(k-2) \end{bmatrix}^T \tag{8.33}$$

$$f(k+3) = \begin{bmatrix} u(k+2) & u(k+1) & u(k) & 0 \end{bmatrix} B \begin{bmatrix} u(k+2) & u(k+1) & u(k) & u(k-1) \end{bmatrix}^T \tag{8.34}$$

Similarly, the contribution of the second order coefficients due to past manipulated variable moves is given below.

$$g(k+1) = \begin{bmatrix} 0 & u(k-1) & u(k-2) & u(k-3) \end{bmatrix} B \begin{bmatrix} 0 & 0 & u(k-1) & u(k-2) \end{bmatrix}^T \tag{8.35}$$

$$g(k+2) = \begin{bmatrix} 0 & 0 & u(k-1) & u(k-2) \end{bmatrix} B \begin{bmatrix} 0 & 0 & u(k-1) & u(k-2) \end{bmatrix} \tag{8.36}$$

$$g(k+3) = \begin{bmatrix} 0 & 0 & 0 & u(k-1) \end{bmatrix} B \begin{bmatrix} 0 & 0 & 0 & u(k-1) \end{bmatrix}^T \tag{8.37}$$

A more compact matrix formulation of the above calculation was developed. The calculation procedure described below makes use of matrix manipulations as far as possible. The use of this calculation method was developed for implementation in *Matlab*, since the environment is optimised for matrix manipulations. An implementation of this matrix-based formulation is executes significantly faster than an implementation based on the formulation described above.

The following matrices of manipulated variables may be defined. The matrices in equation 8.38 to 8.40 are all $PH \times MH$ matrices.

$$U_{\text{future}} = \begin{bmatrix} u(k) & 0 & \cdots & 0 \\ u(k+1) & u(k) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ u(k+PH-1) & u(k+PH-2) & \cdots & 0 \end{bmatrix} \tag{8.38}$$

$$U_{\text{future,past}} = \begin{bmatrix} u(k) & u(k-1) & u(k-2) & u(k-3) & \cdots & u(k-MH-1) \\ u(k+1) & u(k) & u(k-1) & u(k-2) & \cdots & u(k-MH-2) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ u(k+PH-1) & u(k+PH-2) & \cdots & \cdots & \cdots & (k-MH-PH) \end{bmatrix} \tag{8.39}$$

$$U_{\text{past}} = \begin{bmatrix} 0 & u(k-1) & u(k-2) & \cdots & u(k-MH-1) \\ 0 & 0 & u(k-1) & \cdots & u(k-MH-2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & u(k-1) \end{bmatrix} \tag{8.40}$$

The second order contribution of future manipulated variable moves (f) can be calculated by adding the columns of the $PH \times MH$ matrix F (equation 8.41). The result should be a column matrix of length PH. The second order contribution of the past manipulated variable moves can be calculated by summing the columns of G, which should also lead to a colum matrix of length PH.

$$F = U_{\text{future}} \times B \cdot U_{\text{future,past}} \tag{8.41}$$

$$G = U_{\text{past}} \times B \cdot U_{\text{past}} \tag{8.42}$$

This formulation can be extended to MIMO systems. The following matrices may be defined. The matrix shown in equation 8.43 consists of the U_{future} matrices (as defined above) for each input. $\mathbf{0}$ is a $PH \times MH$ matrix of zeros. The matrix given in equation 8.44 is a $n_{\text{MV}}PH \times n_{\text{CV}}MH$ matrix. $\boldsymbol{B}$ is a $n_{\text{MV}}MH \times n_{\text{CV}}MH$ matrix.

$$\boldsymbol{U}_{\text{future}} = \begin{bmatrix} U_{1,\text{future}} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & U_{2,\text{future}} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & U_{n_{\text{MV}},\text{future}} \end{bmatrix} \tag{8.43}$$

$$\boldsymbol{U}_{\text{future,past}} = \begin{bmatrix} U_{1,\text{future,past}} & U_{1,\text{future,past}} & \cdots & U_{1,\text{future,past}} \\ U_{2,\text{future,past}} & U_{2,\text{future,past}} & \cdots & U_{2,\text{future,past}} \\ \vdots & \vdots & \ddots & \vdots \\ U_{n_{\text{MV}},\text{future,past}} & U_{n_{\text{MV}},\text{future,past}} & U_{n_{\text{MV}},\text{future,past}} & U_{n_{\text{MV}},\text{future,past}} \end{bmatrix} \tag{8.44}$$

$$\boldsymbol{B} = \begin{bmatrix} B_{11} & B_{21} & \cdots & B_{n_{\text{CV}}1} \\ B_{12} & B_{22} & \cdots & B_{n_{\text{CV}}2} \\ \vdots & \vdots & \ddots & \vdots \\ B_{1n_{\text{MV}}} & \cdots & \cdots & B_{n_{\text{CV}}n_{\text{MV}}} \end{bmatrix} \tag{8.45}$$

$$\boldsymbol{F} = \boldsymbol{U}_{\text{future}} \times \boldsymbol{B} \cdot \boldsymbol{U}_{\text{future,past}} \tag{8.46}$$

$$= \begin{bmatrix} U_{\text{future},1}B_{11}U_{\text{future,past},1} & \cdots & \cdots & U_{\text{future},1}B_{n_{\text{CV}}1}U_{\text{future,past},1} \\ U_{f,2}B_{21}U_{fp,2} & \cdots & \cdots & U_{\text{future},2}B_{n_{CV}2}U_{\text{future,past},2} \\ \vdots & \vdots & \ddots & \vdots \\ U_{\text{future},n_{\text{MV}}}B_{n_{\text{MV}}1}U_{\text{future,past},n_{\text{MV}}} & \cdots & & U_{\text{future},n_{\text{CV}}n_{\text{MV}}}B_{n_{\text{CV}}n_{\text{MV}}}U_{\text{future,past},n_{\text{MV}}} \end{bmatrix} \tag{8.47}$$

$\boldsymbol{F}$, obtained from equation 8.46, is a $n_{\text{MV}}PH \times n_{\text{CV}}MH$ matrix. This matrix can be subdivided into $n_{MV}n_{\text{CV}}$ sub matrices, corresponding to each input output pair. For example, for a two by two system, the matrix $\boldsymbol{F}$ can be written as shown in equation 8.48. $\boldsymbol{f}$ can be obtained by adding the submatrices corresponding to each output, and then adding the columns of the matrices to obtain a column vector of $n_{\text{CV}PH}$.

$$\boldsymbol{F} = \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \tag{8.48}$$

The column vector $\boldsymbol{g}$ can be calculated in a similar manner by defining the matrices U_{past} and $U_{textrmpast,alt}$ and perform the multiplication shown in equation 8.51. The column vector is then obtained by adding the columns of the sum of the submatrices as described above.

$$\boldsymbol{U}_{\text{past}} = \begin{bmatrix} U_{1,\text{past}} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & U_{2,\text{past}} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & U_{n_{\text{MV}},\text{past}} \end{bmatrix} \tag{8.49}$$

$$\boldsymbol{U}_{\text{future,past}} = \begin{bmatrix} U_{1,\text{past}} & U_{1,\text{past}} & \cdots & U_{1,\text{past}} \\ U_{2,\text{past}} & U_{2,\text{past}} & \cdots & U_{2,\text{past}} \\ \vdots & \vdots & \ddots & \vdots \\ U_{n_{\text{MV}},\text{past}} & U_{n_{\text{MV}},\text{past}} & U_{n_{\text{MV}},\text{past}} & U_{n_{\text{MV}},\text{past}} \end{bmatrix} \tag{8.50}$$

$$\boldsymbol{G} = \boldsymbol{U}_{\text{past}} \times \boldsymbol{B} \cdot \boldsymbol{U}_{\text{past,alt}} \tag{8.51}$$

CHAPTER 9

Control results

The results obtained with controllers based on the models obtained in chapter 7 are presented in this chapter. Controllers based on a pruned Volterra model, a diagonal second order and a first order (linear) Volterra model are investigated. The effects of tuning parameters such as the control and prediction horizons as well as manipulated variable weights on the performance of the controllers are also shown.

9.1 Control strategy

The controlled variables for this system are the final concentrate grade and the recovery of valuable mineral in the final concentrate. Typical control strategies would aim to maximise the recovery while keeping the grade above a minimum value, or maximise the grade while maintaining an acceptable recovery. It would be beneficial to incorporate such strategies directly into the controller. Unfortunately, this approach will require solving a nonlinear programming problem with output constraints. This optimisation problem is considerably more complex that a nonlinear optimisation with input constraints, since this will involve nonlinear constrains as well as a nonlinear optimisation objective function (Maner et al., 1996).

Due to the above mentioned complexities, this investigation was limited to nonlinear programming problems with input constraints. The controller is given setpoints for both grade and recovery. However, since grade and recovery are not independent, not all combinations of these setpoints are valid. The steady state attainable region is shown in figure 9.1. These steady state values can be obtained with all the inputs in the range -1 and 1. This figure shows that the maximum scaled recovery for these inputs is approximately 0.125, which translates to an actual recovery of about 90%. The boundaries on grade are less well defined. Setpoints that are clearly within the plane defined by these points were selected for the evaluation of control performance.

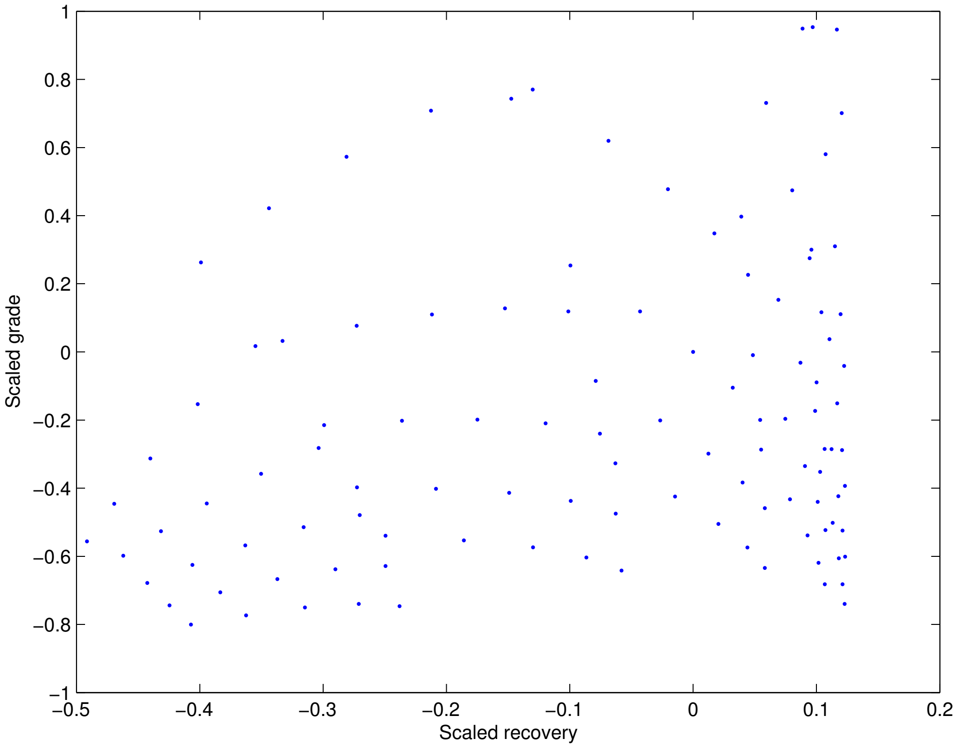

Figure 9.1: The steady state attainable region for the system

9.2 *Simulink* implementation

The controller was implemented in *Simulink* as a s-function. This implementation allows the controller and plant model to be combined in a single Simulink simulation. The plant model can be run continuously, while the controller obtains new measurements and implements control moves at each sampling interval. The Simulink model containing the controller and plant is shown in figure 9.2.

9.2.1 Least squares solution

The unconstrained objective function for the controller can be solved using the least squares technique. Unfortunately, for this system, the matrix $\boldsymbol{G}^T\boldsymbol{G}$ is near singular, and leads to an unstable controller. As discussed in section 8.2.1, the manipulated variable moves, as opposed to the actual manipulated variable values, must be weighted by manipulated variable weights. Due to this, an analytic least squares solution cannot be found if manipulated variable weights are required. However, a solution may be obtained with nonlinear programming methods.

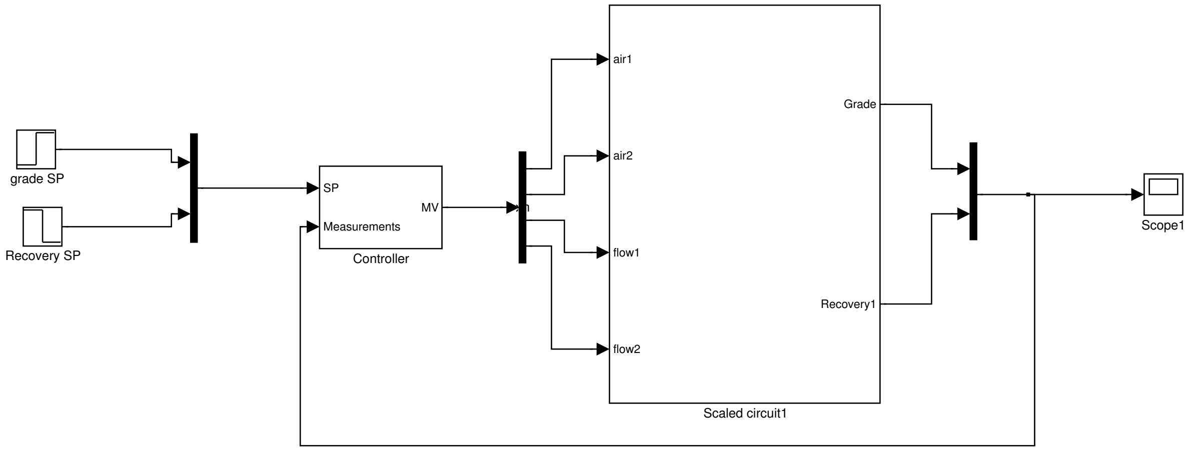

Figure 9.2: The Simulink model containing the controller and plant model

9.3 Nonlinear programming

The *Matlab* nonlinear programming function *fmincon* was used to minimise the objective function given shown in equation 9.1.

$$\begin{aligned} J_f &= (\boldsymbol{s}-\boldsymbol{y})^T \boldsymbol{W}_{CV} (\boldsymbol{s}-\boldsymbol{y}) + W_{MV}\Delta\boldsymbol{u} && (9.1)\\ &= (\boldsymbol{s}-(\boldsymbol{Gu}+\boldsymbol{c}+\boldsymbol{f}))^T W_{CV} (\boldsymbol{s}-(\boldsymbol{Gu}+\boldsymbol{c}+\boldsymbol{f})) + W_{MV}\Delta\boldsymbol{u} && (9.2)\end{aligned}$$

$$-1 \leq u \leq 1 \qquad (9.3)$$

9.4 Controller tuning

The controller has the following tuning parameters:

- move suppression factor (W_{MV});
- controlled variable weights (W_{CV}) and
- setpoint filter (γ).
- control horizon (U);
- prediction horizon (V);

9.4.1 Move suppression factor and controlled variable weights

Move suppression factors are used to penalise manipulated variable changes in the objective function. Move suppression factors slow down the response of the feedback control but increases the robustness of the system to model inaccuracies (Marlin, 2000: 749).

The controlled variable weights can be used to weight the deviations from setpoint of each controlled variable. The setpoint deviations for a weighted controlled variable will decrease at the expense of the performance for the other controlled variables (Marlin, 2000: 749)

Manipulated variable weights were used since the controller was unstable without manipulated variable weights. The magnitude of the manipulated variable weights used will be discussed in section 9.5.

9.4.2 Setpoint filter

A setpoint filter can also be used to prevent large manipulated variable moves by introducing a new setpoint gradually. In this implementation, a setpoint filter was not used since the manipulated variable weights were used and the setpoint filter in not required.

9.4.3 Prediction and control horizon

The controller required large prediction and control horizons to perform well. Both the first order (linear) and second order (nonlinear) Volterra controllers exhibited undesirable behaviour for short prediction and control horizons. Figure 9.3 shows the controlled variables for a setpoint change of 0.8 for grade and figures 9.4 and 9.5 show the manipulated variables for the linear and nonlinear controller. The controlled responses were underdamped. Similar behaviour was observed with a variety of manipulated variable weights. Both the linear and nonlinear controllers performed significantly better for long prediction and control horizons.

9.5 Controller performance

The performance of the first order Volterra and second order (nonlinear) Volterra controllers are compared in this section. Two nonlinear controllers are investigated: a nonlinear controller based on a pruned Volterra model and a controller based on a diagonal second order Voltera model.

In practise, recovery and concentrate grade values higher than the setpoints are not undesirable. However, to be able to compare the relative performance of the nonlinear and linear controllers, positive and negative deviations from setpoint are regarded as equally undesirable.

All the results shown below were generated with a controller with a model horizon of 80, a prediction horizon of 75 and a control horizon of 65.

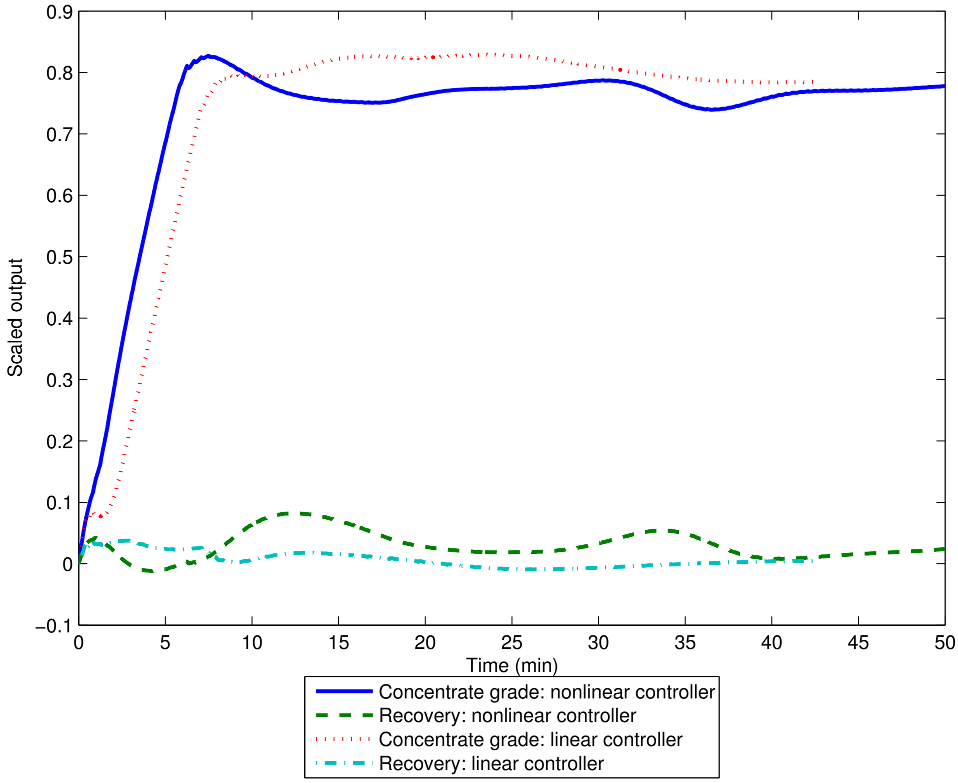

Figure 9.3: The setpoint tracking of the linear and nonlinear controllers for a 0.8 change in concentrate grade setpoint with short prediction and control horizons

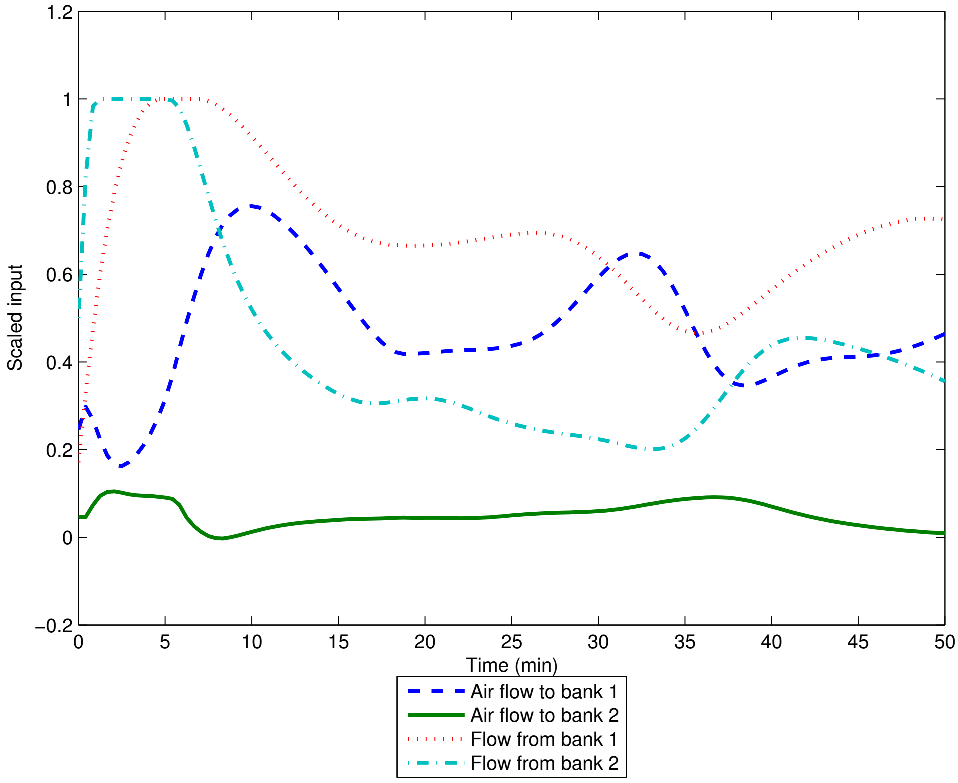

Figure 9.4: The manipulated variable behaviour of the nonlinear controller for a grade setpoint change of 0.8 with short prediction and control horizons

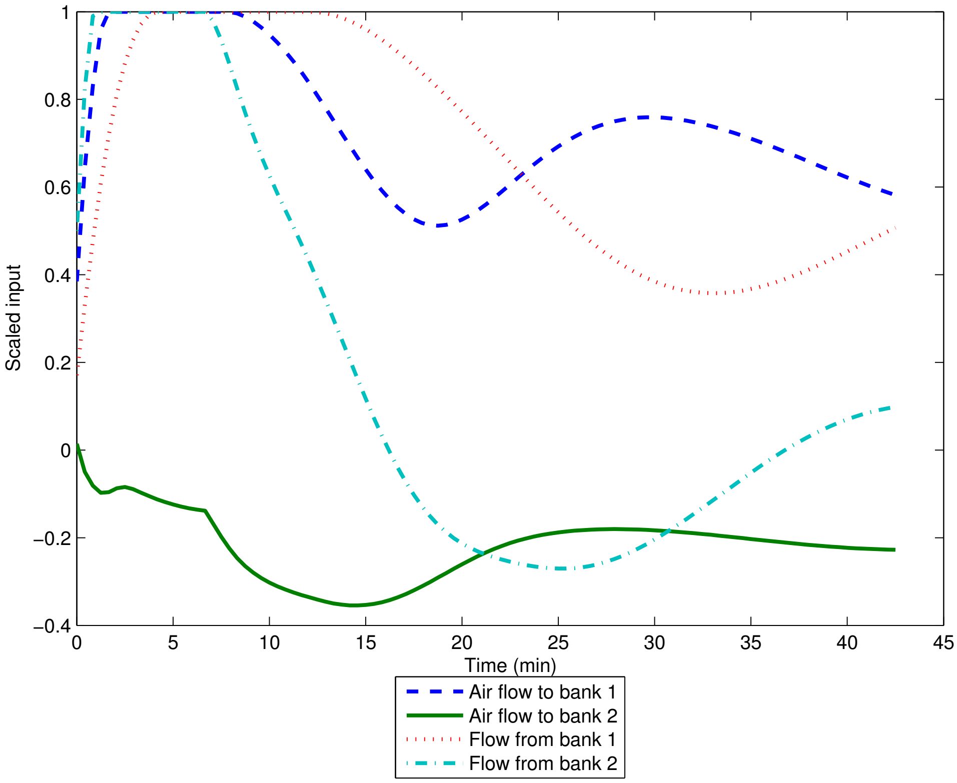

Figure 9.5: The manipulated variable behaviour for a grade setpoint change of 0.8 with short prediction and control horizons

9.5.1 Setpoint changes

The responses for the following setpoint changes are shown below:

- no change in recovery setpoint and a 0.4 step in concentrate grade setpoint
- 0.1 step change in recovery setpoint and -0.3 change in concentrate grade setpoint

These setpoints represent typical behaviour of the system. The 0.1 recovery setpoint change may appear small, but this setpoint is close the boundary of the steady state attainable region shown in figure 9.1.The setpoint tracking obtained by a cautiously tuned nonlinear controller are shown in figures 9.6 and 9.7. The manipulated variable weights used to obtain these responses are shown in table 9.1. For these large manipulated variable weights, the linear, nonlinear and diagonal nonlinear controllers gave similar performances. However, for smaller manipulated variable weights, the controller performances differ significantly. The controlled responses for more aggressively tuned linear and nonlinear controllers for a grade setpoint change of 0.4 are shown in figure 9.8. The pruned Volterra nonlinear controller reached the setpoint faster than the same cautiously tuned controller and had no overshoot. The diagonal nonlinear controller had a slight overshoot, but the linear controller had an considerable overshoot. This shows that the nonlinear controller can be tuned more aggressively than the linear controller and still perform well. The manipulated variable behaviour for the three controllers also differ significantly, as can be seen in figures 9.9 to 9.11. For the pruned Volterra nonlinear controller, the manipulated variables reach their final values after approximately an hour. The diagonal nonlinear controller takes longer, while the linear controller's manipulated variables did not reach their final values during the period shown in the graph.

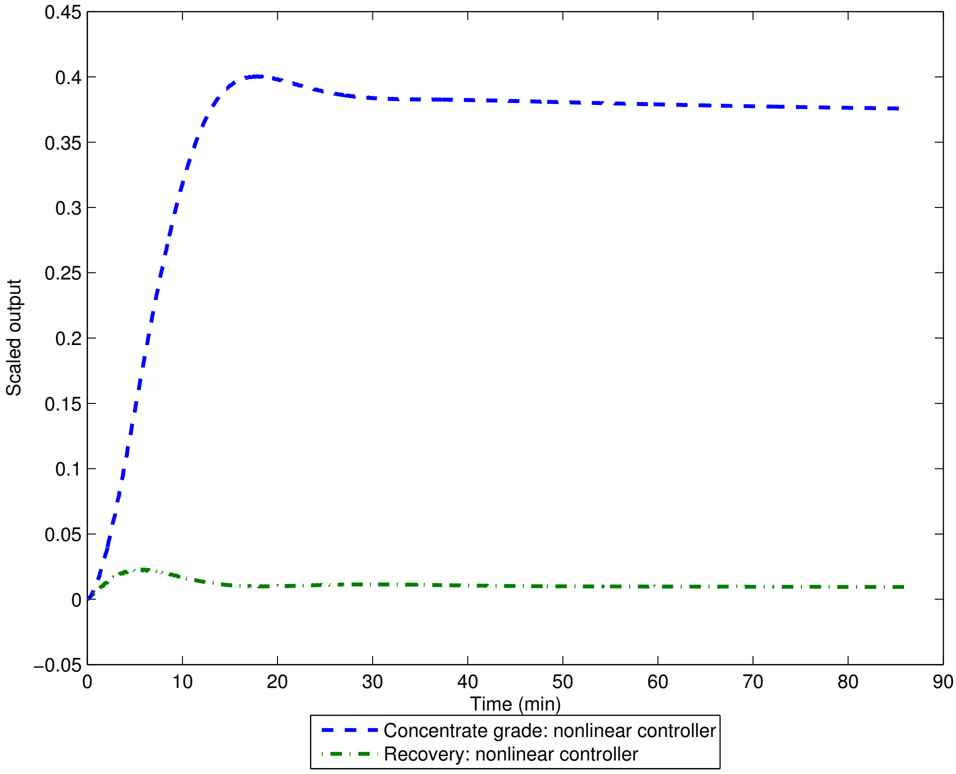

Figure 9.6: The setpoint tracking obtained with a cautiously tuned nonlinear controller for a grade setpoint change of 0.4

9.5.2 Disturbance rejection

The disturbance that was considered is a change in the water flowrate to the circuit. This disturbance alters the residence time of pulp in the cell banks and also affects entrainment. The controlled responses for disturbances of -0.25 and 0.25 in the water flowrate are shown in figures 9.12 and 9.13. The responses obtained with linear and diagonal nonlinear controllers were essentially the same for these cautious tunings. The disturbance rejection of more aggressively tuned linear and nonlinear controllers are shown in figures 9.14 and 9.15. For these disturbances, the linear controller performed slightly better than the nonlinear controllers.

aggressively

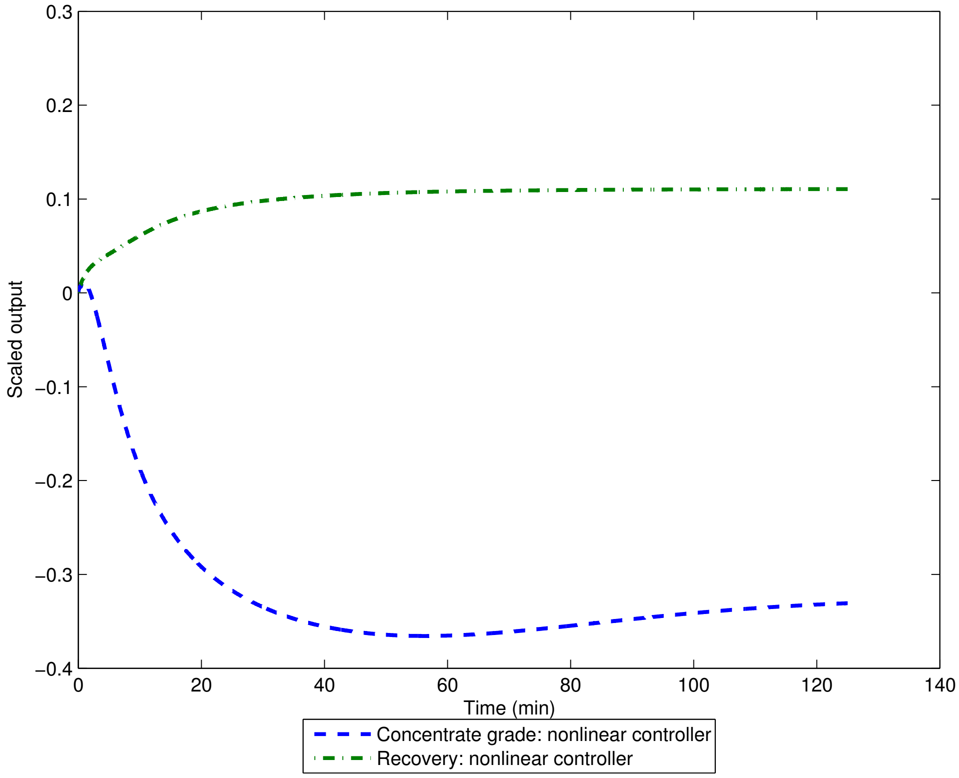

Figure 9.7: The setpoint tracking obtained with a cautiously tuned nonlinear controller for a recovery setpoint change of 0.1 and a grade setpoint change of -0.3

Table 9.1: The manipulated variable weights used to obtain the controlled responses

Response	Air flow to bank 1	Air flow to bank 2	Flow from bank 1	Flow from bank 2
figure 9.3 figure 9.4 figure 9.5	0.5	0	3	1
figure 9.6 figure 9.7 figure 9.12 figure 9.13	4	3	8	7
figure 9.8 figure 9.9 figure 9.10 figure 9.11	0.5	0	3	2
figure 9.14 figure 9.15 figure 9.16 figure 9.17 figure 9.18	1	0.5	4	3

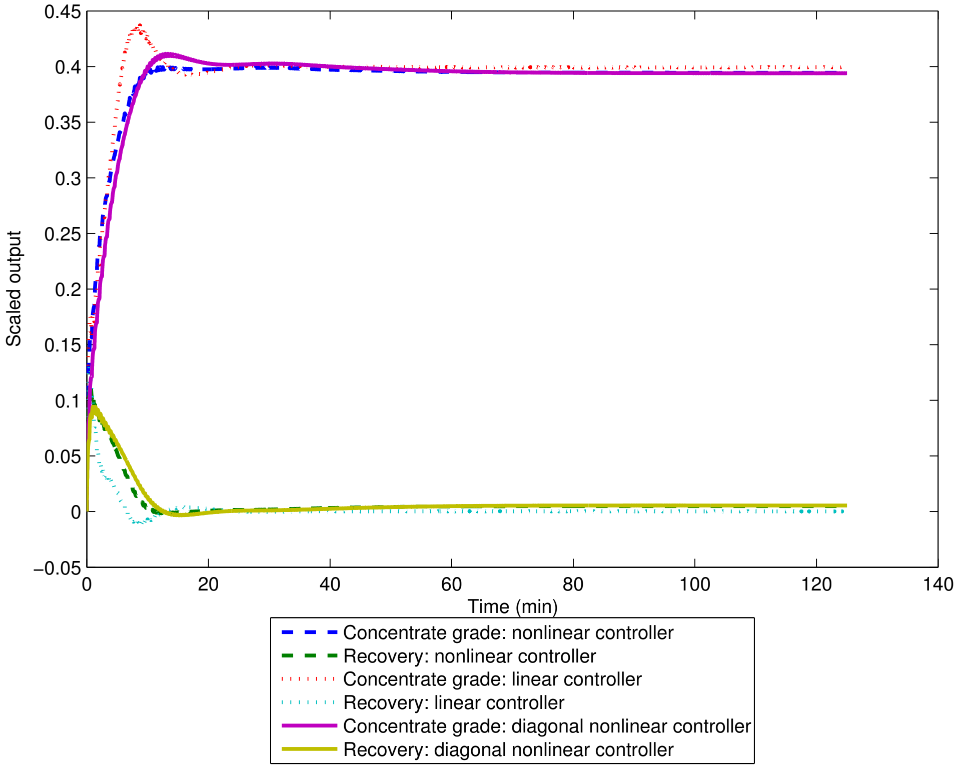

Figure 9.8: The setpoint tracking obtained with aggressively tuned nonlinear and linear controllers for a grade setpoint change of 0.4

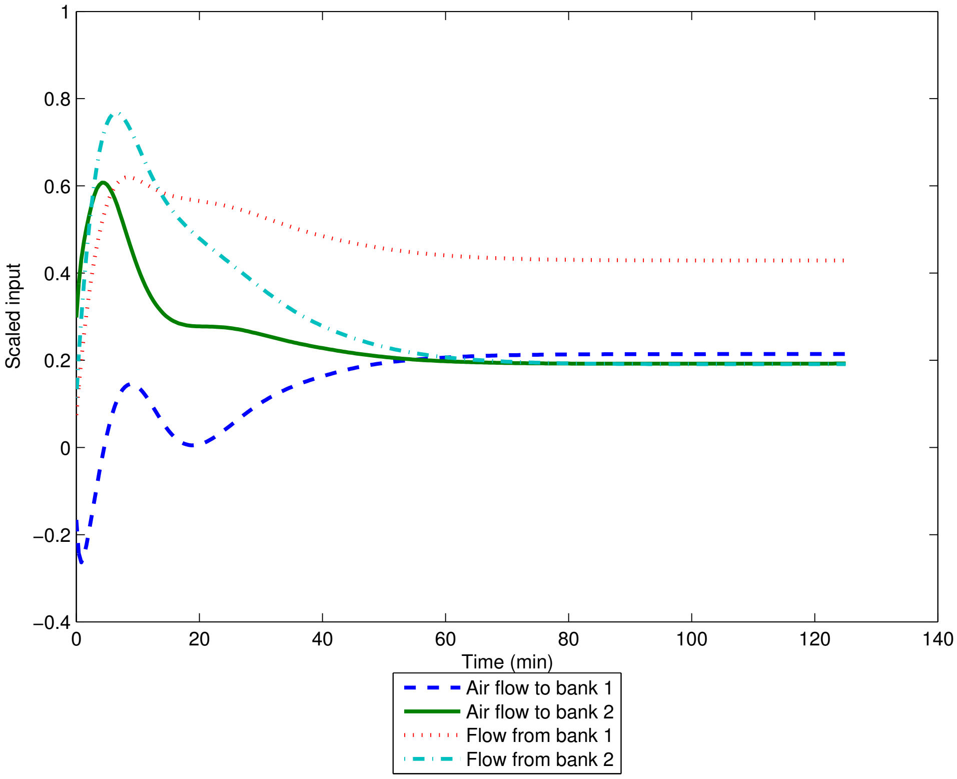

Figure 9.9: The manipulated variable behaviour of an aggressively tuned nonlinear controller for a grade setpoint change of 0.4

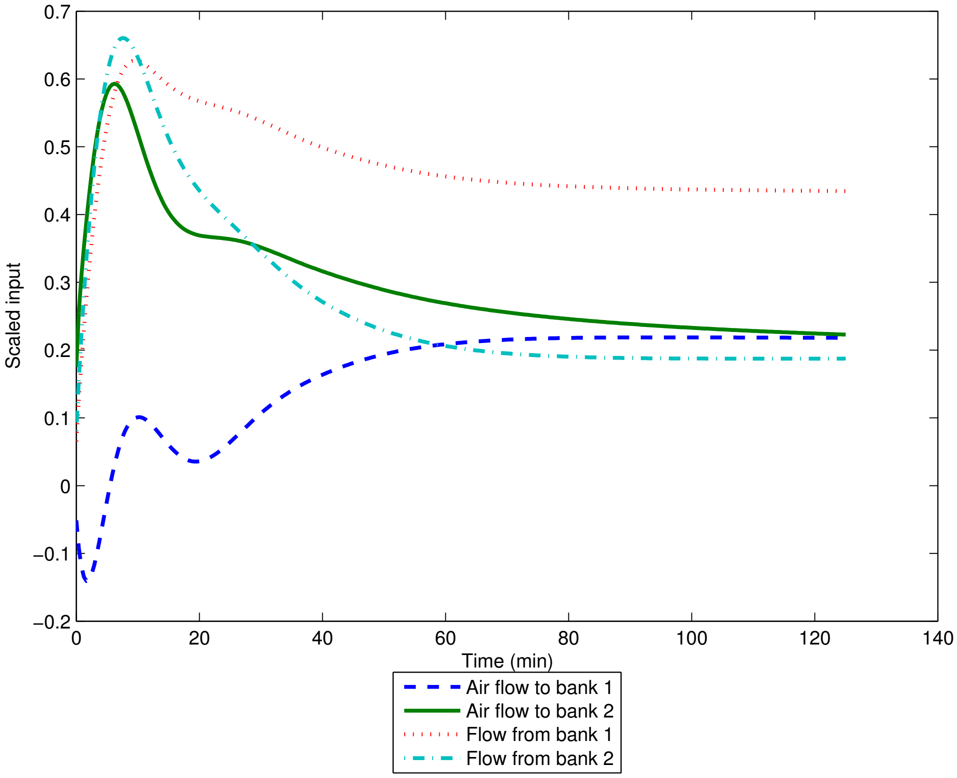

Figure 9.10: The manipulated variable behaviour of an aggressively tuned diagonal nonlinear controller

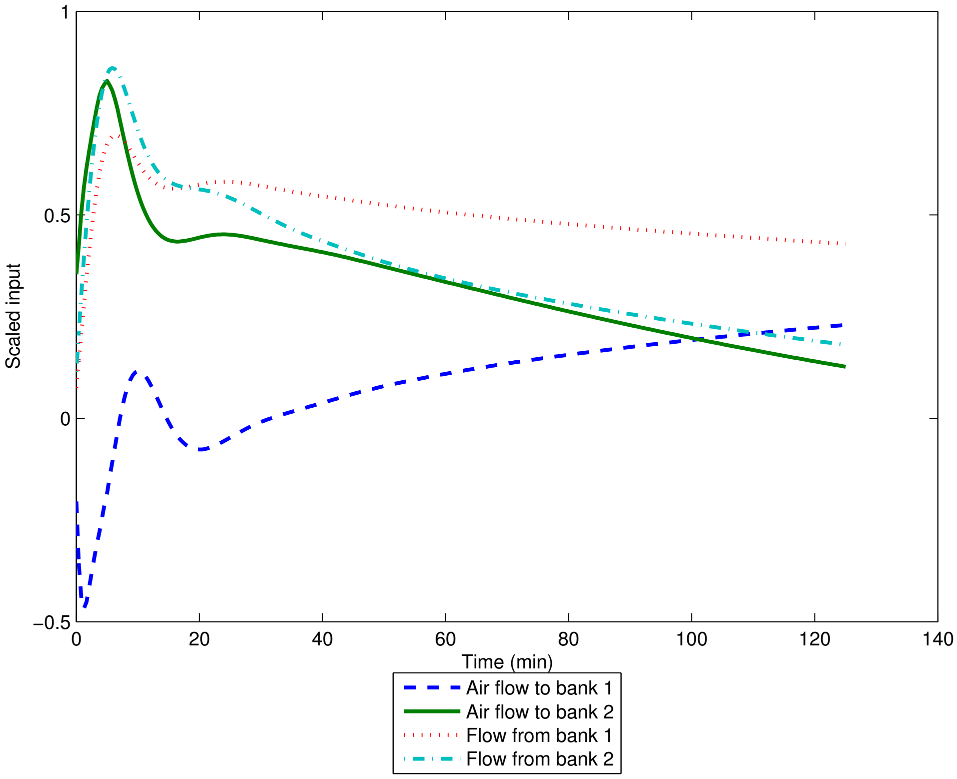

Figure 9.11: The manipulated variable behaviour of an aggressively tuned linear controller for a setpoint change of 0.4

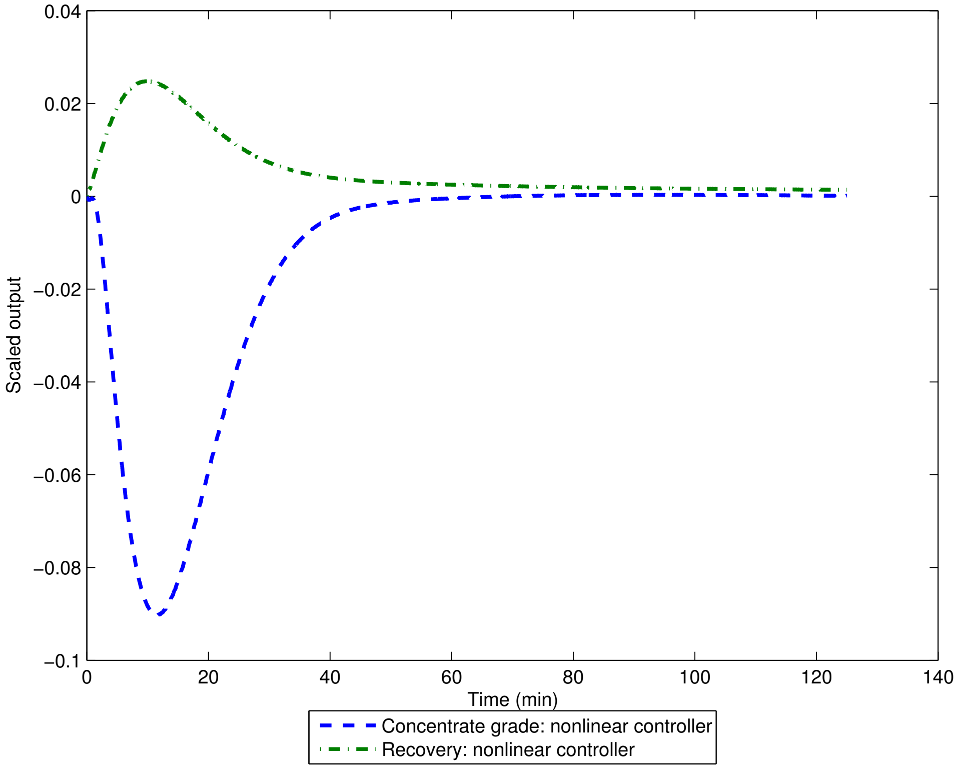

Figure 9.12: The disturbance rejection of a cautiously tuned nonlinear controller for a 0.25 change in water feed rate

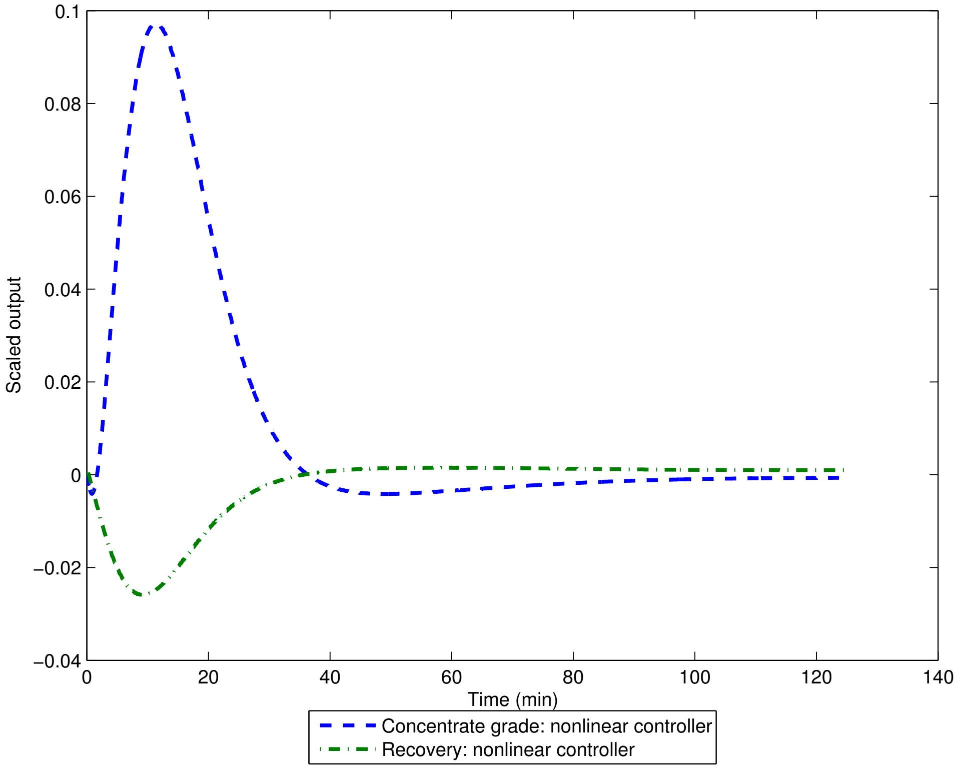

Figure 9.13: The disturbance rejection of cautiously tuned linear and nonlinear controllers for a -0.25 change in the water feed rate

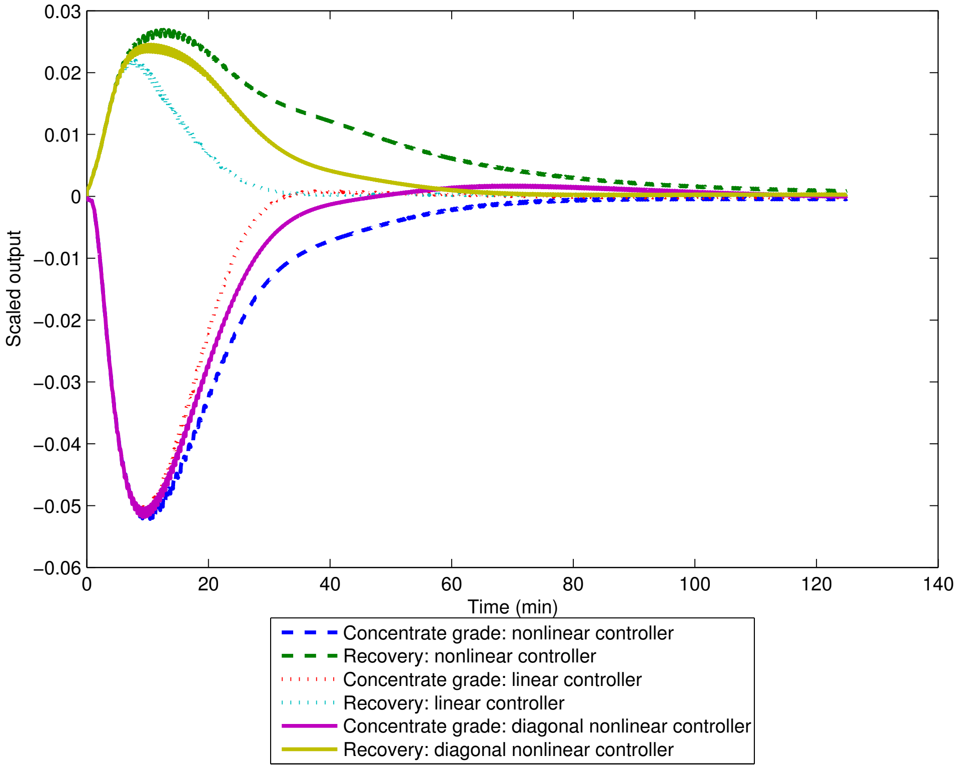

Figure 9.14: The disturbance rejection of aggressively tuned linear and nonlinear controllers for a 0.25 change in water feed rate

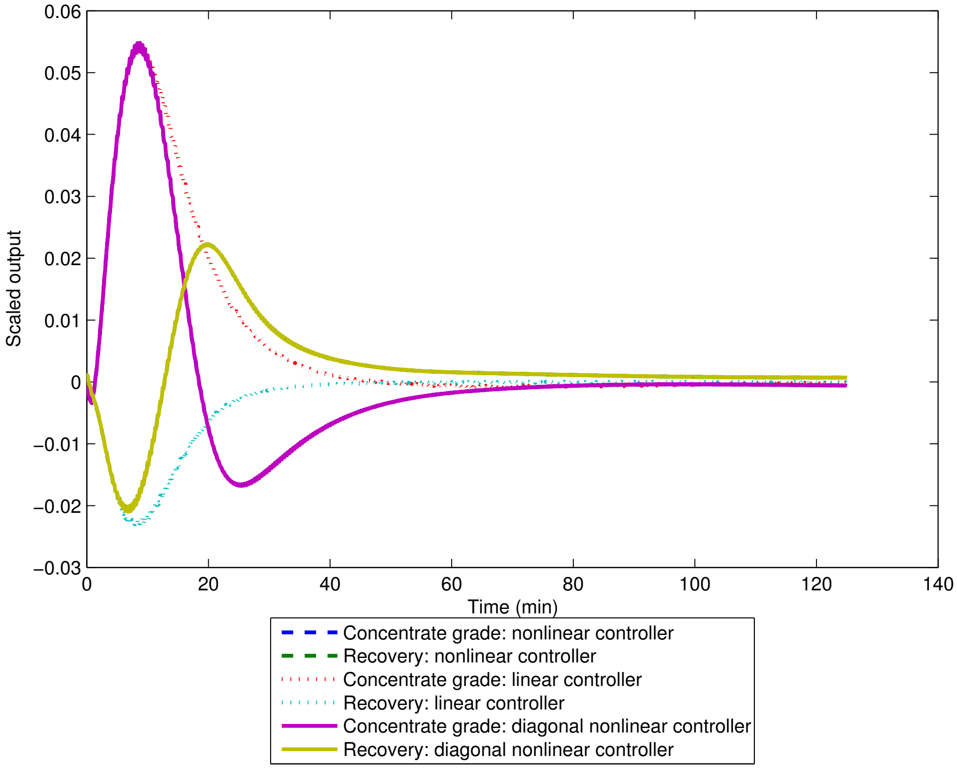

Figure 9.15: The disturbance rejection of aggressively tuned linear and nonlinear controllers for a -0.25 change in water feed rate

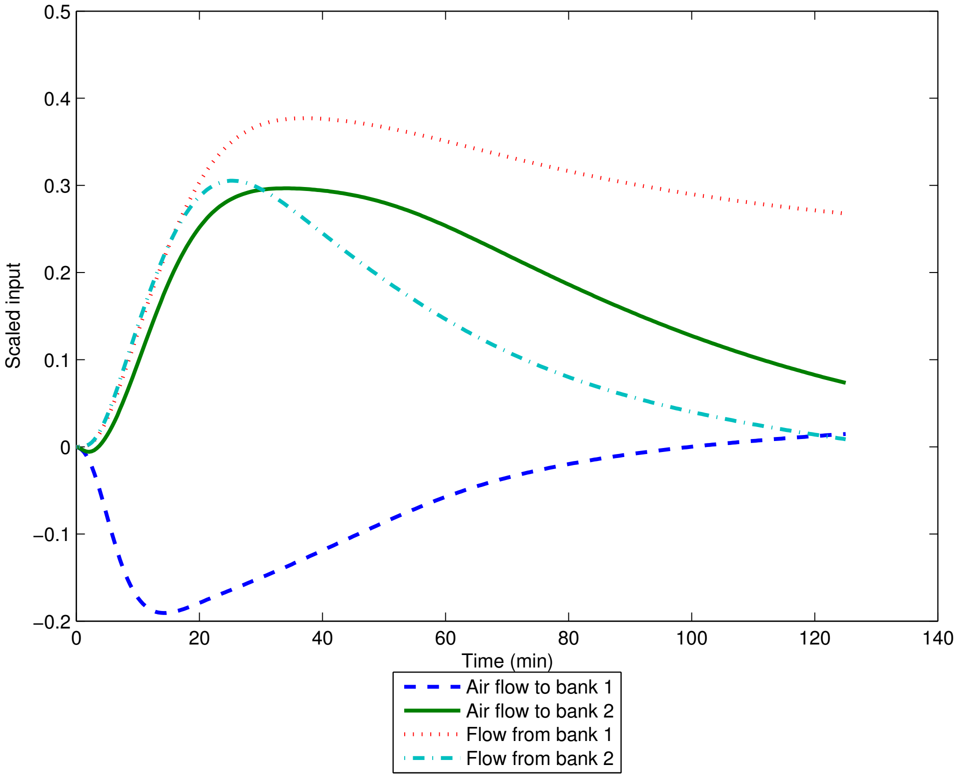

Figure 9.16: The manipulated variable behaviour for a aggressively tuned nonlinear controller for a water feedrate disturbance of 0.25

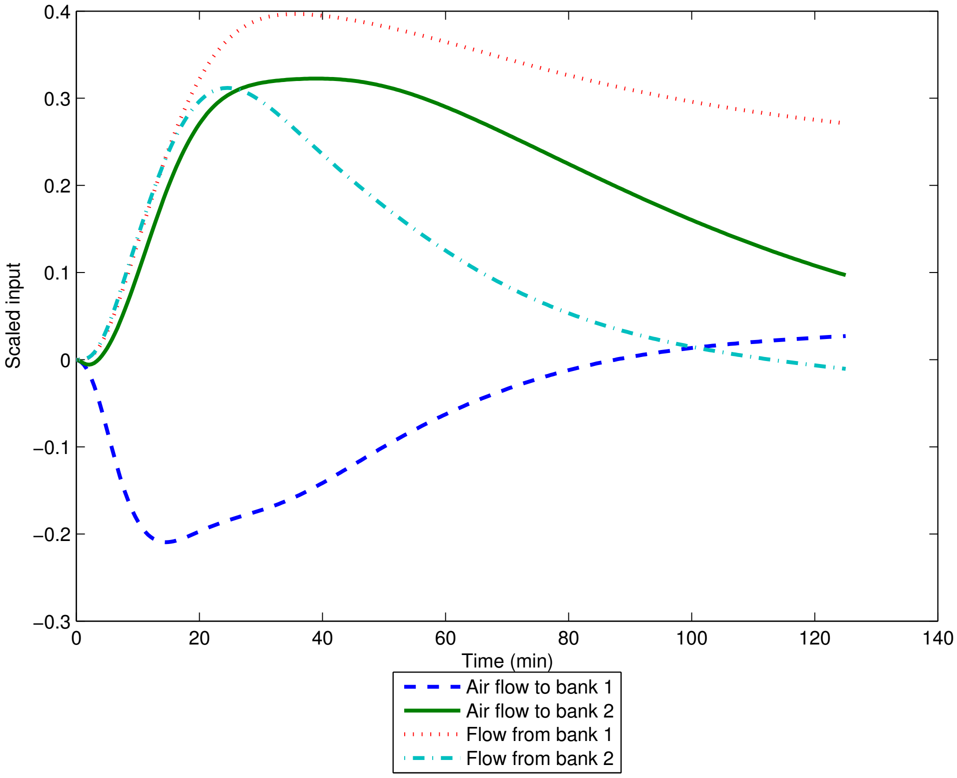

Figure 9.17: The manipulated variable behaviour for a aggressively tuned diagonal nonlinear controller for a water feedrate disturbance of 0.25

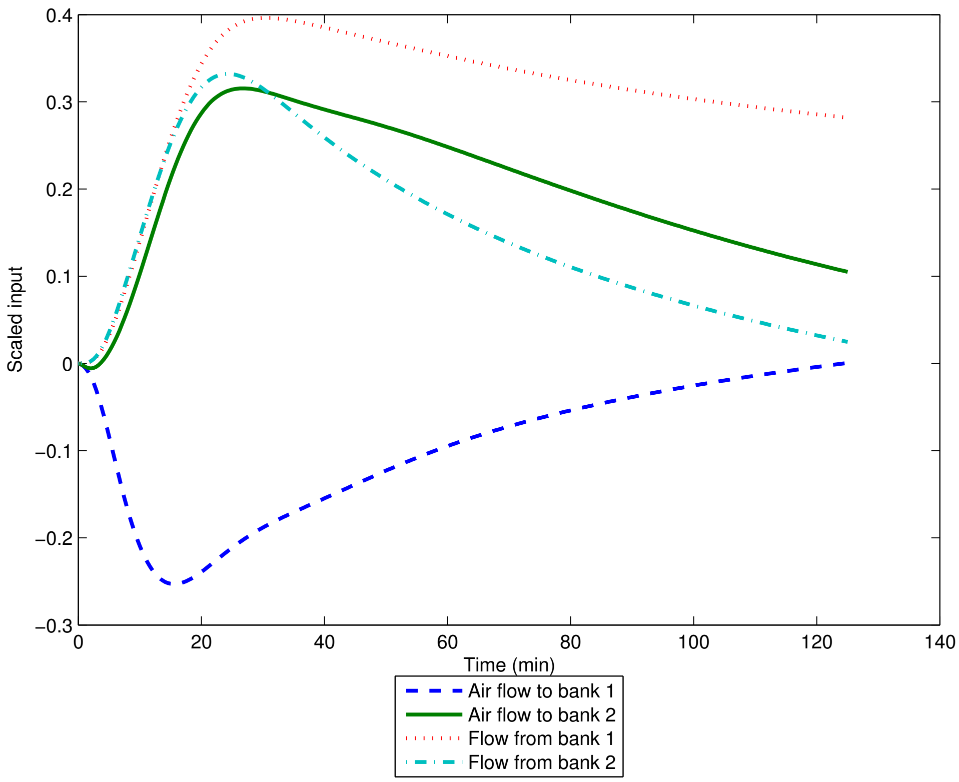

Figure 9.18: The manipulated variable behaviour for a aggressively tuned linear controller for a water feedrate disturbance of 0.25

9.6 Discussion

As described above, three controllers were evaluated in this chapter:

- a nonlinear controller based on a pruned second order Volterra model;
- a nonlinear controller based on a second order Volterra model containing only diagonal coefficients and
- a linear controller based on a first order Volterra model.

All three controllers were able to achieve setpoint tracking as well as disturbance rejection when using sufficiently large prediction and control horizons.

For setpoint tracking as well as disturbance rejection, all three controllers gave essentially the same results when using large manipulated variable weights. Although the control with these settings was acceptable, the responses tended to be sluggish. To obtain faster setpoint tracking, the manipulated variable weights may be reduced. When using the more aggressive tunings, the performance of the linear controller deteriorates, while good performance is obtained with the nonlinear controller based on a pruned second order model. The nonlinear controller based on a diagonal second order model performed better than the linear model, but worse than the controller based on the pruned second order model. For the aggressive manipulated variable tunings, the manipulated variables of the linear controller took much longer to settle at their final values than the nonlinear controllers. This means that the nonlinear controller is able to stabilise the plant faster after a setpoint change.

The nonlinear controller based on the pruned second order model can be tuned more aggressively because the predicted process responses are more accurate, as was seen in chapter 7. The diagonal second order models are not more accurate than the linear model for all input–output pairs, but the improvement in model fit for the input–output pairs exhibiting gain sign changes is significant enough to improve the overall process prediction.

For the case of disturbance rejection, the linear controller performed better than the nonlinear controllers. However, the deviations from setpoint are very small. It is possible that the disturbance excites the plant in a small region where the linear models are more accurate than the nonlinear models.

As discussed in section 7, while pruned Volterra models require less data than full Voltera models, the data sets are still large. This means that long plant tests are required. For some of the input – output pairs, the improvements obtained cannot justify the plant upsets caused by the identification routines. However, it is possible to identify nonlinear models only for those input – output pairs known to be highly nonlinear and to use linear models for the mildly nonlinear input – output pairs.

The nonlinear controller also requires that a nonlinear programming problem must be solved at each sampling interval. For this to occur in real time, significant computational power is required. However, the computational power of commercially available computers is increasing rapidly and a large computational load may be handled easily.

CHAPTER 10

Conclusions

10.1 Flotation modelling

A model of a simple flotation circuit was constructed from semi-empirical relationships as well as fundamental equations. This model describes the qualitative dynamic relationships between the model inputs and the model outputs. The process responses obtained from this model exhibits significantly nonlinear process responses, including asymmetrical responses to symmetric inputs and gain sign changes. Such behaviour cannot be described with linear models.

10.2 Model identification

Full (unconstrained) second order Volterra models require excessively large data sets. Even for a data set of 45000 points, the models did not provide a satisfactory fit. This motivates the use of "pruned" Volterra models, where some of the model coefficients are constrained to zero.

Pruned Volterra models were obtained by evaluating the sum of squared prediction errors and the steady state error for with a coefficient at its calculated value and with the coefficient constrained to zero. Pruned models were obtained from data sets reduced by up to 75%. The models obtained from the smaller data sets gave sum of squared prediction error values comparable to those obtained with a full data set. This shows that pruned Volterra models can be obtained from significantly less data than full Volterra models. It is also possible to use second order Volterra models containing only diagonal coefficients. These diagonal models require even less data than pruned Volterra models containing off-diagonal coefficients but tend to be less accurate.

The pruned models obtained in this way provided significantly better predictions than linear models. The pruned second order model also performed better than the controller based on a diagonal second order model. Both the pruned second order Volterra models and the diagonal second order Volterra models are able to represent asymmetrical responses including gain sign changes. The diagonal second order models performed better than the linear models for responses containing severe nonlinearities, but performed slightly worse than the linear models for mildly nonlinear responses.

10.3 Nonlinear Volterra model predictive control

Volterra model predictive control is an intuitive nonlinear control algorithm, since it is a higher order extension of the familiar linear model predictive algorithm. Models of varying complexity may be used as basis for the controller.

Model predictive controllers based on first order, pruned second order and diagonal second order Volterra models were implemented. For large manipulated variable weights, the performance of the three controllers were essentially the same for both setpoint tracking and disturbance rejection. These tunings resulted in adequate but sluggish responses.

If the controllers are tuned more aggressively, the performance of the respective controllers differ significantly. The nonlinear controller based on a pruned Volterra model gave good performance even when tuned aggressively. For the same tunings, the controller based on a first order Volterra model had an overshoot. The controller based on a diagonal second order model performed slightly worse than the controller based on a pruned Volterra model, but better than the linear model. The nonlinear models may be tuned more aggressively, since the model predictions are more accurate.

The linear controller performed slightly better than the nonlinear controllers for the case of disturbance rejection. The maximum deviation was essentially the same for all three controllers, but the linear controller returned to the steady state value faster than the other two. The deviation from setpoint is very small. It is possible that the disturbances excites the plant in a region where the linear models are more accurate than the nonlinear models.

It is possible to combine linear, nonlinear and diagonal models in the controller formulation. This can be advantageous, since the nonlinear terms give a drastic improvement in model fit if the input – output response is highly nonlinear but a smaller improvement for more mildly nonlinear responses. This approach will reduce the data requirements.

A disadvantage of the nonlinear control algorithm is that a nonlinear programming problem must be solved at each sampling time. This is more computationally intensive than solving a linear program. However, due to the rapidly increasing computational power of commercially available computers, this is not a major issue.

10.4 Recommendations

The following recommendations are made:

10.4.1 Modelling of flotation circuits

Expansion of process model The model of the flotation circuit has not been validated with plant data. The accuracy of the model can be increased if plant data is used to obtain values for the model parameters. The model can also be expanded to include the effects of the addition rates of flotation reagents such as collectors and frothers, since these flows are often used as manipulated variables. Including these inputs will allow control strategies using these manipulated variables to be evaluated.

10.4.2 Model identification

Superposition of outputs In the present work, it was assumed that the response of the system due to two (or more) simultaneous inputs can be approximated with the sum of the responses due to the inputs acting individually. This assumption is valid for linear systems, but is only approximate for nonlinear systems. It is possible that improved performance may be obtained if terms describing the response to simultaneous inputs are added.

Third order Volterra series Third order Volterra series are able to describe higher order nonlinearities that second order Volterra series are not able to describe. However, these models require even more data than second order Volterra models. These models will only be viable for input–output pairs exhibiting severe nonlinearity. However, the it is possible that the improvement in model may justify the added complexity. These issues should be investigated.

10.4.3 Volterra nonlinear control

Nonlinear programming solution with output constraints The current investigation was limited to controllers using nonlinear programming techniques with input constraints. The inclusion of output constraints lead to a more complex nonlinear programming problem. However, the inclusion of output constraints allows one output to be minimised while keeping another output above or below a specified value. For example, the recovery of valuable mineral may be maximised while keeping the grade above a specified level. This allows the controller to optimise the circuit performance. It is recommended that the inclusion of output constraints should be included.

APPENDIX A

Model parameter values

The values for the model parameters are given in table A.1. The value of the floatibility for the valuable mineral is in the range reported by Vera et al. (2002) and it was assumed that the floatibility for the gangue was zero. The parameters a to e for the bubble surface are flux correlations are taken from Gorain et al. (1999). The drainage parameter used in the classification function CF and the froth recovery factor correlation was taken from Vera et al. (2002).

Gorain et al. (1999) reported cell characteristics such as cell and impeller dimensions as well as air flowrates and particle sizes. Typical values were selected and are reported in Table A.2.

The steady state values and maximum deviations that were used in scaling of the circuit are shown in table A.3. The maximum deviations were the maximum values attainable dynamically. For example, the recovery may reach values higher than 100 % for short periods. For example, when the air flow to the cells are increases, more valuable mineral particles are collected. If the concentration of valuable mineral in the cell is high at that moment, the recovery will be high. As the concentration of valuable mineral in the cell is decreases, the recovery will lessen again.

Table A.1: Model parameters

Parameter	Symbol	Value	Units
Mineral floatibility	P	0.0025	dimensionless
Gangue floatibility	P	0	
S_b correltation parameter	a	137.5	
	b	0.2905	
	c	0.7278	
	d	0.0685	
	e	0.3551	
Drainage parameter	ω	1	$^1/_{\text{min}}$
Froth parameter	β	4	$^1/_{\text{min}}$
Constant describing flow between cells	K_{flow}	0.05	
Control valve coefficient for rougher bank	K_{valve}	0.12	
Control valve coefficient for cleaner bank		0.175	

Table A.2: Cell geometry and hydrodynamic characteristics

Quantity	Symbol	Value	Units
Impeller aspect ratio	As	0.75	
Impeller peripheral speed	N_s	0.75	$^{\text{m}}/_{\text{s}}$
80% passing feed size	P_{80}	80	μm
Cell cross sectional area	A	1.875	m^2
Cell height	h	1.75	m

Table A.3: Steady state values and maximum deviations for model inputs and outputs

Variable	Steady state value	Maximum deviation (+ and -)
Air flow	0.03 $^{m^3}/_{\text{s}}$	
Tailings flow from rougher bank	0.1 $^{m^3}/_{\text{s}}$	0.005 $^{m^3}/_{\text{s}}$
Tailings flow from cleaner bank	0.031 $^{m^3}/_{\text{s}}$	0.006 $^{m^3}/_{\text{s}}$
Ore feed flow	45 $^{\text{kg}}/_{\text{s}}$	
Water feed flow	90 $^{\text{kg}}/_{\text{s}}$	5 $^{\text{kg}}/_{\text{s}}$
Recovery	84.5	150
Concentrate grade	172.6	150

APPENDIX B

Program structure and contents

This Appendix discusses the implementation of a flotation circuit model in *Simulink*, the program used for model identification as well as the implementation of the model predictive controller. The files used are listed and discussed and the structure of the programs are illustrated.

B.1 Flotation model

The model of a flotation circuit was implemented in *Simulink*. A model of a flotation cell was created as a collection of *Simulink* subsystem blocks. The flotation cell block can be connected to form a model of a flotation bank and the banks are connected to form a flotation circuit. The structure of the flotation model is shown in figure B.1 and the blocks used to create the model is described below.

Bubble surface area flux The bubble surface area flux for the cell design and the current the gas flowrate into the cell is calculated.

Cell connection This block calculates the flow from the cell based on the level in the current cell and the following cell, as well as the type of cell connection. Available options for the type of cell connections that may be selected are “ Valve” and “Partially connected”. If “Valve” is selected, the flow from the cell is regulated with a control valve. If the “Partially separated” option is selected, it is assumed that the cells are separated with a flange.

Entrainment The recovery of valuable mineral and gangue through entrainment is calculated. The inputs for this block are the mass of mineral in the cell and the froth

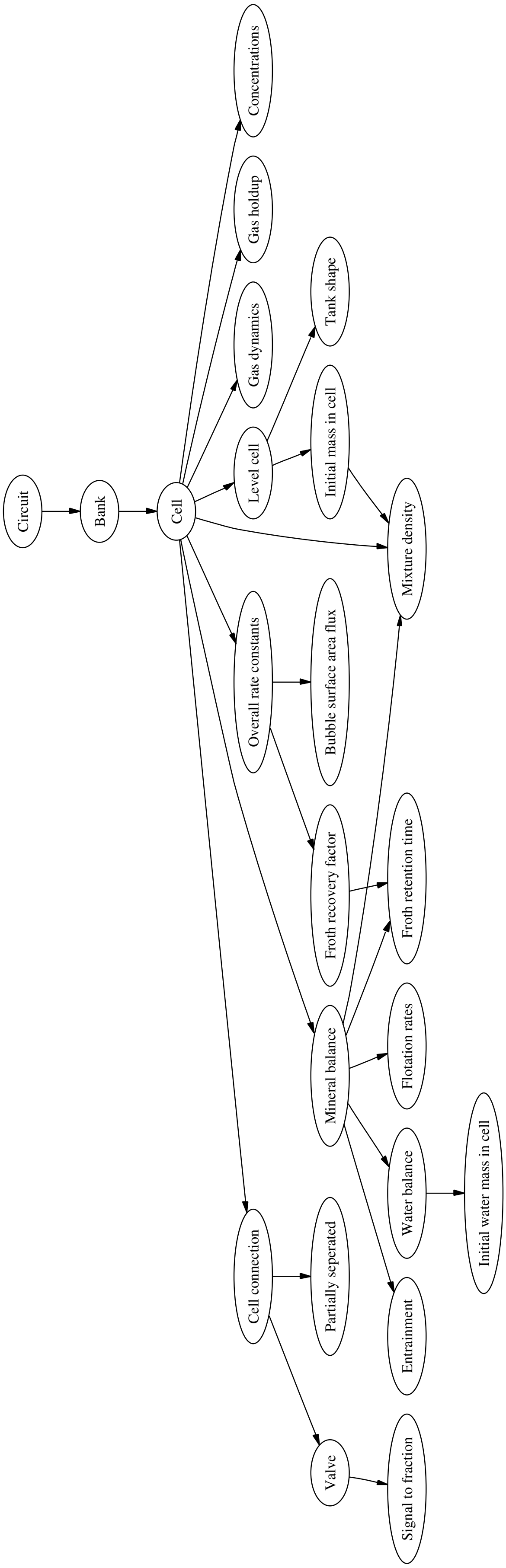

Figure B.1: The structure of the flotation circuit model

residence time.

Froth recovery factor The froth recovery factor (R_f) for the current froth residence time is calculated in this block.

Froth residence time This block calculates the froth residence time based on the flowrate of air into the cell as well as the froth level.

Initial mass in cell This block calculates the initial mass pulp in the cell from quantities supplied by the user through the user interface. The initial mass pulp in the cell is required as an initial condition.

Initial mass water in cell The initial mass water in the cell is calculated from the initial pulp level and water concentration. This quantity is required as an initial condition for solving the differential equations.

Level cell This block calculates the pulp level in the flotation cell.

Mineral balance This block calculates the mineral balance for the cell. The block contains several subsystems.

Mixture density The density of a mixture is calculated from the mass fractions in the cell and the densities of the components.

Partially connected This block calculates the flow between cells connected with a flange.

Tank shape This block calculates the level in the tank for a given volume pulp in the cell.

Valve This block calculates the flow through a control valve.

Water mass balance Calculates the mass balance for water.

B.2 Model identification

The structure of the identification program is shown in figure B.2. The structure of the file “parameterEstimation” is shown in detail in figure B.3. The program consists of .m files (scripts or functions), *Simulink* models, and .mat files. The following files were used.

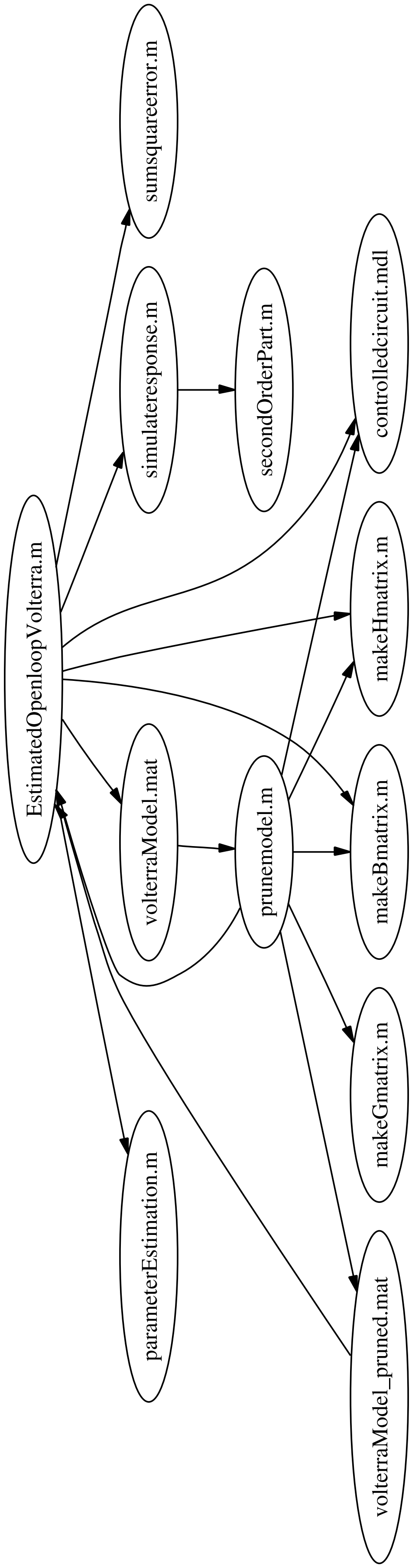

Figure B.2: The structure of the model identification program

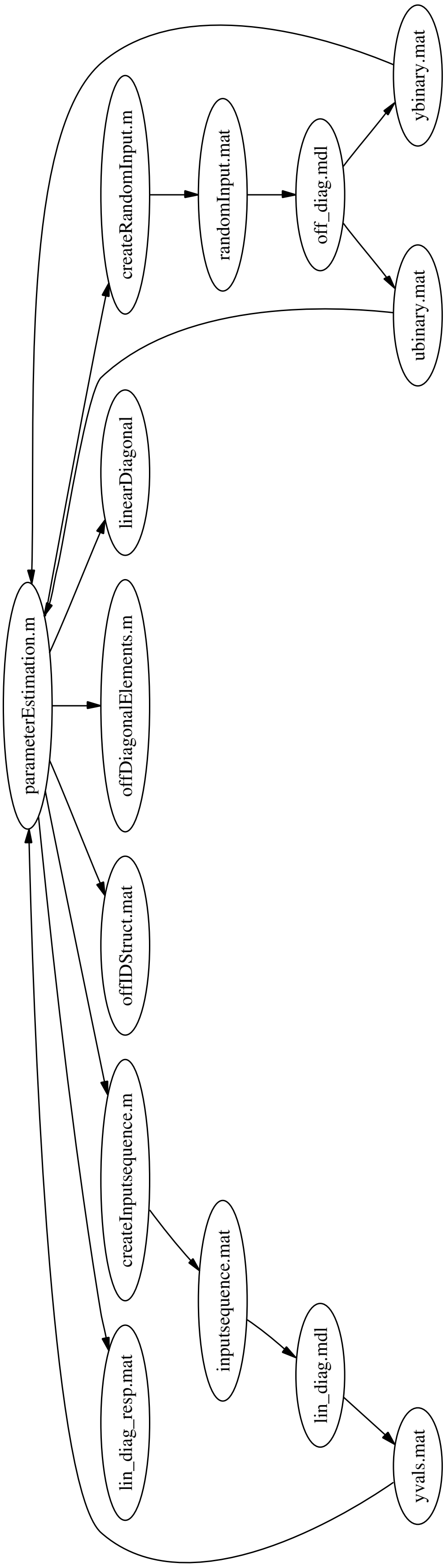

Figure B.3: The structure of the function parameterEstimation

EstimatedOpenloopVolterra.m This file calculates the response as predicted by a Volterra series model and is called by the user. A new Volterra model may be calculated, or a model that has been calculated previously may be loaded. If a new model is to be calculated, new data may be obtained from simulations or existing data (stored in the form of .mat files) may be used.

The function has the following input arguments:

- Tfin, the sampling interval;
- T, the sampling interval;
- calculateNewModel, specifying whether new model coefficients should be calculated, or if a model should be loaded;
- numIn, the number of model inputs;
- numOut, the number of model outputs;
- datareductionfrac, specifying whether the full PRBS system response should be used to calculate a new model;
- simulatenewmodel, specifying whether new data should be obtained and
- pruned, specifying whether a full or pruned Volterra model should be used.

createInputsequence.m The impulse input sequence used for the identification of the first order and diagonal second order coefficients is created in this function. The resulting input sequence is saved in the file "inputsequence.mat".

createRandomInput.m The pseudo random binary input sequence used for the identification of the off-diagonal Volterra coefficients is created in this file.

inputsequence.m This .mat file contains the input sequence used to identify the linear and diagonal second order Volterra coefficients.

lin_diag_resp.mat The data used to identify the first order and diagonal second order coefficients are stored in this file. This is the file that is loaded if new data are not simulated.

linearDiagonal.m The first order (linear) and diagonal second order coefficients are calculated in this function.

makeBmatrix.m The "B" matrix of second order coefficients (see section 8.2.4) is created in this function.

makeGmatrix.m The "G" matrix of first order coefficients (see section 8.2.1) is created in this function.

makeHmatrix.m The "H" matrix of first order coefficients (see section 8.2.1) is created in this function.

off_diag.mdl This *Simulink* model contains the basic flotation circuit model. The process response to a PRBS is obtained from this model.

offDiagonalElements.m The off-diagonal second order Volterra coefficients are calculated in this function.

offIDStruct.mat This .mat file contains the data required to identify the off-diagonal Volterra coefficients. This structure is loaded if a Volterra model is to be obtained from existing data.

parameterEstimation.m This function is called by "EstimatedOpenloopVolterra" and calculates the Volterra model coefficients. This function creates .mat files containing the required input sequences, runs simulations and obtains the simulation results in the form of .mat files. This function also calls the functions which calculates the model coefficients.

prunemodel.m This function obtains a pruned Volterra model from a full Volterra model. One coefficient is considered at a time. The model fit with a coefficient at its calculated value is compared with the model fit with the coefficient constrained to zero. The actual process response may be simulated, or obtained from a .mat file.

randomInput.mat This file contains the pseudo random binary input sequence that is used to identify the off-diagonal second order Volterra coefficients.

secondOrderPart.m The contributions of the second order terms are calculated in this function.

simulateresponse.m The open loop response as predicted by a second order Volterra model is calculated with this function.

sumsquareerror.m The sum of the squared differences between the predicted model response and actual open loop response is calculated with this function.

ubinary.mat This .mat file contains the pseudo random binary input sequence used to identify the off-diagonal Volterra coefficients sampled at the same sampling rate as "ybinary.mat".

volterraModel.mat This file contains a structure with the Volterra model coefficients stored in the fields.

volterraModel_pruned.mat This file contains a structure with the pruned Volterra model coefficients stored in the fields.

ybinary.mat This file contains the process response (as obtained in *Simulink*) to a pseudo random binary sequence.

yvals.mat "yvals" contains the process response (as obtained from the *Simulink* simulation) to the input sequence stored in "inputsequence". This data is used to identify the first order and diagonal second order coefficients.

B.3 Model predictive controller

The model predictive controller was implemented as an s-function in *Simulink*. Implementing the controller in *Simulink* is advantageous, since the controller can easily interface with the process model and issues such a discrete sampling rates are handled easily. The nonlinear programming solution was obtained with the function "fmincon", which can be found in *Matlab*'s Optimisation Toolbox.

The structure of the controller program is shown in figure B.4 and the files used are discussed below.

attainable_region The steady state attainable region is obtained in this function. Each manipulated variable is varied between -1 and 1 in a specified step size. The steady state gain for all possible combinations of manipulated variables is obtained and plotted.

makeBmatrix, makeGmatrix and makeHmatrix . As described above.

mimoSecondOrderPart This function calculates the $\boldsymbol{f}$ term according to the matrix formulation developed in section 8.2.4.

mimoSecondOrderPast This function calculates the $\boldsymbol{g}$ term according to the matrix formulation described in section 8.2.4.

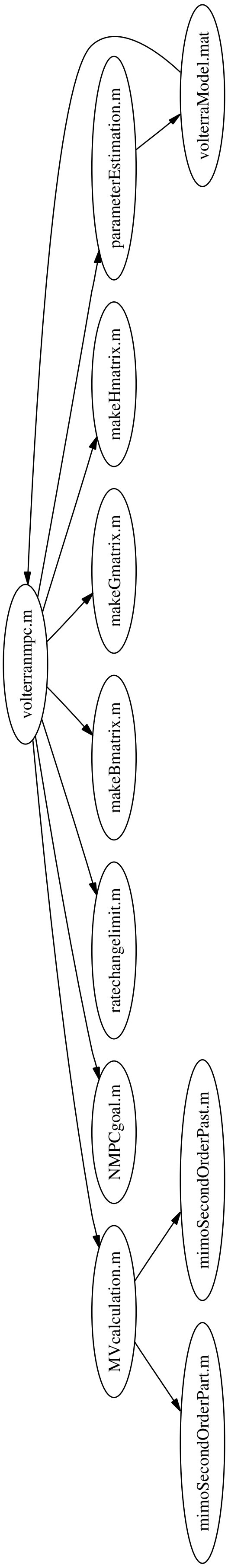

Figure B.4: The structure of the model predictive controller program

MVcalculation This function calculates the manipulated variable moves. Either nonlinear programming or a least squares solution may be used.

NMPCgoal This is the objective function required by "fmincon". The predicted response is calculated in this function and the errors between the predicted response and the required response is calculated. The squared sum of these errors as well as the weighted manipulated variable moves are minimised.

ratechangelimit Constraints on the rate at which manipulated variables may be changes are implemented in this function.

volterranmpc.m This is the main controller function. It is implemented according the the standard *Simulink* s-function template.

BIBLIOGRAPHY

Al-Duwaish, H. and Karim, M. N. (1997) “A new method for the identification of hammerstein model”, *Automatica, 33* (10), 1871–1875.

Bendat, J. S. (1991) *Nonlinear system analysis and identification from random data*, John Wiley and Sons, New York.

Bloom, F. and Heindel, T. J. (2002) “On the structure of collision and detachment frequencies in flotation models”, *Chemical Engineering Science, 57*, 2467–2473.

Casali, A.; Gonzalez, G.; Agusto, H. and Vallebuona, G. (2002) “Dynamic simulator of a rougher simulation circuit for a copper sulphide ore”, *Minerals Engineering, 15*, 253–262.

Cubillos, F. A. (1998) “Adaptive hybrid neural models for process control”, *Computers and Chemical Engineering, 22*, s989 – s992.

Cutler, C.; Morshedi, A. and Haydel, J. “An industrial perspective on advanced control”, Washington DC (1983) AIChe annual meeting.

Cutler, C. R. (1982) “Dynamic matrix control of imbalanced systems”, *ISA transactions, 21* (1).

Deglon, D. A.; Sawyer, F. and O’Connor, C. T. (1999) “A model to relate the flotation rate constant and the bubble surface are flux in mechanical flotation cells”, *Minerals Engineering, 12* (6), 599–608.

Desbiens, A.; Hodouin, D. and Mailloux, M. “Nonlinear predictive control of a rougher flotation circuit”, in *Proceedings of the IFAC symposium: Automation in mining mineral and metal processing*, pages 287 – 292 Cologne, Germany (1998).

Doyle, F. J.; Ogunnaike, B. A. and Pearson, R. K. (1995) "Nonlinear model based control using second order volterra models", *Automatica, 32* (5), 697–714.

Feng, D. and Aldrich, C. (1999) "Effect of particle size on flotation performance of complex sulphide ores", *Minerals Engineering, 12* (7), 721–731.

Ferreira, J. P. and Loveday, B. K. (2000) "An improved model for flotation circuits", *Minerals Engineering, 13*, 1441–1543.

Finch, J. A. and Dobby (1990) *Column flotation*, Pergamon Press, Oxford.

Finch, J. A.; Xiao, J.; Hardie, C. and Gomez, C. O. (2000) "Gas dispersion properties: Bubble surface area flux and gas holdup", *Minerals Engineering, 13* (4), 365–372.

Gaudin, A. M. (1957) *Flotation*, McGraw Hill, New York.

Gorain, B. K.; Franzidis, J. P. and Manlapig, E. V. (1995) "Studies on impeller type, impeller speed and air flow rate in an industrial scale flotation cell: Part 1: Effect on bubble size distribution", *Minerals Engineering, 8* (6), 615–635.

Gorain, B. K.; Franzidis, J. P. and Manlapig, E. V. (1996) "Studies on impeller type, impeller speed and air flow rate in an industrial flotation cell. part 3: Effect of superficial gas velocity", *Minerals Engineering, 9* (6), 639–654.

Gorain, B. K.; Franzidis, J. P. and Manlapig, E. V. (1997) "Studies on impeller type, impeller speed and air flow rate in an industrial scale flotation cell. part4: effect of bubble surface area flux on flotation performance", *Minerals Engineering, 10* (4), 367–379.

Gorain, B. K.; Franzidis, J. P. and Manlapig, E. V. (1999) "The emperical prediction of bubble surface area flux on mechanical flotation cells from cell design and operating data", *Minerals Engineering, 12* (3), 309–322.

Gorain, B. K.; Harris, M. C.; Franzidis, J. P. and Manlapig, E. V. (1998)a "The effect of froth residence time on the kinetics of flotation", *Minerals Engineering, 11* (7), 627–638.

Gorain, B. K.; Napier-Mun, T. J.; Franzidis, J. P. and Manlapig, E. V. (1998)b "Studies on impeller type, impeller speed and air flow rate in an industrial flotation cell: Part 5: Validation of the k/S_b relationship and effect of froth depth", *Minerals Engineering, 11* (7), 615–626.

Heiskanen, K. (2000) "On the relationship between flotation rate and bubble surface area flux", *Minerals Engineering, 13* (2), 141–149.

Hemphill, A. and Loveday, B. "Application of models for optimization of a flotation plant", in *South African Chemical Engineering Congress*, Sun City, South Africa 2-6 September (2003).

Henson, M. A. (1998) "Nonlinear model predictive control: current status and future directions", *Computers and Chemical Engineering, 23*, 187–202.

Hodouin, D.; Jamsa-Jounela, S. L.; Carvalho, M. T. and Berg, L. (2001) "State of the art and challenges in mineral processing control", *Control Engineering Practice, 9*, 995 – 1005.

Jamsa-Jounela, S. L.; Dietrich, M.; Halmevaara, K. and Tilli, O. (2003) "Control of pulp levels in flotation cells", *Control Engineering Practice, 11*, 73–78.

Kampjarvi, P. and Jamsa-Jounela, S. L. (2003) "Level control strategies for flotation cells", *Minerals Engineering, 12*, 1061–1068.

Kapur, P. C.; Dey, A. and Mehrotra, S. P. (1991) "Identification of feed and simulation of industrial flotation circuits", *International Jurnal of Mineral Processing, 31*, 11–35.

Kashiwagi, H. and Li, Y. (2004) "Nonparameteric nonlinear model predicitve control", *Korean Journal of Chemical Engineering, 21* (2), 329–337.

Kirjavainen, V. M. (1996) "Review and analysis of factors controlling the mechanical flotation of gangue minerals", *International journal of mineral processing, 46*, 21–34.

Koh, T. and Powers, E. J. (1985) "Second order volterra filtering and its application to nonlinear system identification", *IEEE transactions on acoustics, speech and signal processing, 33* (6), 1445 – 1455.

Koukoulas, P. and Kalouptsidis, N. (2003) "Blind identification of second order hammerstein series", *Signal Processing, 83* (213-234).

Luyben, W. L. (1990) *Process Modeling, Simulation and Control for Chemical Engineers*, chapter 6 McGraw-Hill New York 2 edition.

Lynch, A. J.; Johnson, N. W.; Manlapig, E. V. and Thorne, C. G. (1981) *Mineral and coal flotation circuits: Their simulation and control*, Elsevier Scientific Publishing Company, .

Maner, B. R.; amd B A Ogunnaike, F. J. D. and Pearson, R. K. (1996) "Non-linear model predictive control of a simulated multivariable polymerization reactor using second order volterra models", *Automatica, 32* (9), 1285–1301.

Marlin, T. E. (2000) *Process Control*, McGraw Hill, New York 2 edition.

Mathe, Z. T.; Harris, M. C.; Connor, C. T. and Franzidis, J. P. (1998) "Review of froth modelling in steady state flotation systems", *Minerals Engineering, 11* (5), 397–421.

Nowak, R. D. and Veen, B. D. V. (1994) "Random and pseudorandom inputs for volterra filter identification", *IEEE transactions on signal processing, 42* (8), 2124–2144.

Parker, R. S.; Heenstra, D.; Pearson, F. J. D. R. K. and Ogunnaike, B. A. (2001) "The identification of nonlinear models for process control using tailored plant friendly input sequences", *Journal of Process control, 11*, 237–250.

Pearson, R. K. (1995) "Nonlinear input output modelling", *Journal of Process control, 5* (4).

Pearson, R. K.; Ogunnaike, B. A. and Doyle, F. J. (1996) "Identification of structurally constrained second order volterra models", *IEEE Transactions on Signal Processing, 44* (11), 2837 – 2847.

Pearson, R. P. and Pottmann, M. (2000) "Gray-box identification of block-oriented nonlinear models", *Journal of Process Control, 10*, 301–315.

Perez-Correa, E. Z. R.; Munoz, C. and Cipriano, A. "Heuristic and model predictive control strategies for a simulated flotation circuit", in *Automation in mining, mineral and metal processing*, Barker, I. J. (Ed.), pages 75–81 Sun City South Africa August (1995).

Perez-Correa, R.; Gonzales, G.; Casali, A.; Cipriano, A.; Barrera, R. and Zavala, E. (1998) "Dynamic simulation and advanced multivariable control in conventional flotation circuits", *Minerals Engineering, 11* (4), 333–345.

Persechini, M. A. M.; Jota, F. G. and Peres, A. E. C. (2000) "Dynamic model of a flotation column", *Minerals Enigineering, 12* (14-15), 1465–1481.

Polat, M. and Chander, S. (2000) "First order flotation kinetics models and methods for estimation of the true distribution of flotation rate constants", *International Journal of Minerals Processing, 58*, 145–166.

Qin, S. J. and Badgwell, T. A. (2003) "A survey of model predictive control technology", *Control Engineering Practice, 11*, 733–764.

Richalet, J.; Rault, A.; Testud, J. L. and Papon, J. (1978) "Model predictive heuristic control: Applications to industrial processes", *Automatica, 14*, 413–428.

Rivera, W. M. L. D. E. (2001) "A methodology for control-relevant nonlinear system identfication using restricted complexity models", *Journal of Process Control, 11*, 209–222.

Savassi, O. N.; Alexander, D. J.; Francis, J. P. and Manlapig, E. (1998) "An empirical model for entrainment in industrial flotation plants", *Minerals Engineering, 11* (3), 243–256.

Seborg, D. E.; Edgar, T. F. and Mellichamp, D. A. (1989) *Process dynamics and control*, Wiley, .

Sistu, P. B.; Gopinath, R. S. and Bequette, B. W. (1993) "Computational issues in nonlinear predictive control", *Computers and Chemical Engineering, 17* (4), 361–366.

Soderstrom, T. and Stoica, P. (1989) *System identification*, Prentice Hall, New York.

Stenlund, B. and Medvedev, A. (2002) "Level control in cascade coupled flotation tanks", *Control engineering practice, 10*, 443–448.

Subrahmanyam, T. V. and Forssberg, E. (1988) "Froth stability, particle entrainment and drainage in flotation: a review", *International Journal of Mineral Processing, 23*, 33–53.

Suichies, M.; Leroux, D.; Dechert, C. and Trusiak, A. (2002) "An implemantation of generalized predictive control in a flotation plant", *Control Engineering Practice, 8*, 319–325.

Sung, S. W. and Lee, J. H. (2003) "Pseudo-random binary sequence design for finite impulse response identification", *Control Engineering Practice, 11*, 935–947.

Sutherland, K. L.; and Wark, I. W. (1955) *Principles of flotation*, Australian Institute of Mining and Metallurgy, Sydney.

Thornton, A. J. (1991) "Cautious adaptive control of an industrial flotation circuit", *Minerals Engineering, 4* (12), 1227–1242.

Vera, M. A.; Mathe, Z. T.; Franzidis, J. P.; Harris, M. C.; Manlapig, E. V. and Connor, C. T. (2002) "The modelling of froth zone recovery in batch and continuously operated laboratory flotation cells", *International Journal of Mineral processing, 64*, 135–151.

Villeneuve, J.; Guillaneau, J. C. and Durance, M. V. (1995) "Flotation modelling: A wide range of solutions for solving industrial problems", *Minerals Engineering,* (4-5), 409–420.

Weisstein, E. "Kurtosis http://www.mathworld.wolfram.com", [June 2004] (2004)a.

Weisstein, E. "Skewness http://www.mathworld.wolfram.com", [June 2004] (2004)b.

Wills, B. A. (1997) *Mineral processing technology*, Butterworth Heinmann, sixth edition.