Development of a Nonlinear Predictive Control Algorithm and its Application to Flotation

BENJAMIN D. H. KNIGHTS

Submitted in fulfillment of the requirements for the degree of

Master of Science in Engineering

Department of Chemical Engineering

University of Cape Town

Cape Town, South Africa.

Supervisors:

Prof. C. L. E. Swartz

Mr W. Langson

October 2001

Synopsis

This study consists of four clearly defined and interlinked objectives. The first is the development of a method of solving nonlinear optimal control problems. This method is then used to solve the underlying optimal control problem in a nonlinear model predictive control (NMPC) strategy. By way of a case study, and to further understanding of the mechanisms of flotation, a dynamic model of a flotation circuit is developed. This nonlinear dynamic model is then used in the NMPC strategy to simulate the nonlinear predictive control of a flotation cell.

An indirect approach is used in the method for solving optimal control problems, applying the Euler-Lagrange equations to transform the optimal control problem into a two-point boundary value problem (BVP). Conventionally these problems have been solved using the multiple shooting differential equation solver, or variants of it. In this work the BVP is solved by orthogonal collocation, using the `FORTRAN` package `COLSYS`.

Strategies are developed to solve a number of different classes of optimal control problems, including problems with constraints on the control trajectory, and terminal state constraints. An important aspect of this solution method is that it produces a continuous control trajectory over the entire solution horizon.

The use of this optimal control problem in the nonlinear model predictive control strategy of Chen and Algöwer (1998) is then shown. This strategy is a quasi-infinite horizon NMPC strategy making use of a stabilising terminal penalty cost and terminal state inequality constraint. Stability is guaranteed if the optimal control problem is feasible.

Conventional MPC, including NMPC, makes use of control moves that are piecewise constant over each sampling period. This is due to conventional solution methods discretising the time axis at each of the sampling points over the control horizon. This results in a set of piecewise constant control moves, of which only the first is implemented and the optimal control sequence then recalculated for the updated system state. Using the optimal control

solution method proposed in this work, the control input determined over the prediction horizon is not limited to being piecewise constant. This continuously varying control signal is then used over each sampling period, resulting in an MPC strategy that uses piecewise *continuously varying* control moves. The use of the continuously varying control input is shown to result in a lower overall objective function cost for the case examined. This is particularly apparent in the input variables, depending on the system and objective function weightings.

A simulation of the application of NMPC to a flotation cell is used as a case study. A dynamic model of a flotation cell is developed, based on a dynamic mass balance of the ore split into floatability classes, and a dynamic water balance. This model manipulates air sparge-rate and valve opening, and the feed to the cell was identified as the primary disturbance variable.

The controller attempts to control concentrate grade and slurry level in the tank. An additional cost is placed on the tailings flow fluctuation to attempt to prevent downstream instability caused by pulsing flowrates. Simulations of the controller performance for set-point changes to the grade (while keeping the tank level constant), and disturbances to the feed stream are shown. The NMPC scheme was able to successfully and efficiently calculate the control moves necessary, and rapidly achieve the setpoints. A comparison with conventional linear MPC (QDMC) was performed, using a step response model identified over the expected operating regime of the system. The NMPC scheme performed considerably better than the linear MPC strategy, with rapid, stable achievement of the setpoints.

Acknowledgements

Although this thesis bears my name, it is the sum of collaborative input from many people and institutions whose help, support and encouragement have been vital to its completion.

To my supervisors, Prof. Chris Swartz and Wilbur Langson, my sincere gratitude. Despite distance and disruption your response has always been rapid, useful and of extremely high quality.

I would like to thank the Council for Mineral Technology (MINTEK) for their financial support. In particular I would like to thank Dr Dave Hulbert for his interest and helpful suggestions during the course of this work. Thanks also to Anglo Platinum and the people at the Bafokeng Rasimone Platinum mine concentrator for the opportunity to view their facility.

No project can be completed without a stimulating environment promoting innovative thought and productive activity. I would like to thank my office mates through the years for providing this - Rhoda Baker, David Seaman, Richard MacRosty, Kevin Dunn, and Andrew Msiza. In particular I would like to thank Andy for his hours of endurance as he bore the brunt of my successes and failures - thanks for the wise words and useful council throughout this project.

For all your love and support, thank you to my family. Knowing that you were always there for me gave me strength through the hardest moments.

These acknowledgements would not be complete without credit being given to the resident [*nix, gnuplot, lyx, LaTeX, FORTRAN, maths] guru, Warwick Duncan.

To my local pseudo-supervisors[1] I would like to express my heartfelt thanks. Your ability to nod your heads at the right time, celebrate my successes, give encouragement during

[1] Anabelle, Christopher, DAvid, Derek, Dr Kunene, Goofy, Grumpy, Lilli, Little-Brother, Liz, Lotti, Mickey-boy, Ruth, Sven, Wes, Wol

my failures, and listen to my unintelligible ramblings without falling asleep was really appreciated.

Finally, my deepest thanks to my best friend. I could not have done it without you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Model predictive control (MPC) is an advanced control strategy that has now been widely adopted by a number of industries. The algorithm has several different implementations used by the various industrial vendors, but they all have a basic structure in common. The changes to manipulable variables of the plant are chosen, based on the behaviour of a process model, to optimise the future behavior of the plant. The adjustments to the manipulated variables are chosen by the solution of an optimisation problem which takes the form of an optimal control problem. Nonlinear model predictive control (NMPC) selects the optimal control sequence based on a nonlinear model of the process. By using a nonlinear model the difficulty of solution of the optimal control problem increases, and only recently have these controllers been seen in industrial applications.

A considerable amount of research has gone into the solution of optimal control problems, and a number of significant advances have been made. Methods have been developed that solve many different classes of these problems, with differing process models, optimisation goals, and constraints.With regular improvements in solution algorithms, and greater and greater computational power, larger and more complex problems can be solved in sufficiently short time to allow for on-line implementation.

1.1 Motivation and Goals

Model predictive control algorithms are typically solved using a discretised optimisation algorithm, resulting in a set of piecewise constant control moves. However it is reasonable

to expect that the use of a piecewise *continuously varying* control sequence is likely to produce improved control, and more accurately satisfy the control objective.

The stability of MPC, particularly NMPC, has received a considerable amount of research attention. Due to the nature of the MPC algorithm, the general form does not guarantee closed-loop stability (Chen and Allgöwer, 1998). A number of significant advances have been made in the stability analysis of linear MPC. However, stability of NMPC is still the focus of considerable research activity. A number of approaches have been considered, with different authors proposing various NMPC schemes guaranteeing stability under certain assumptions. The majority of these methods make use of *stabilising constraints*, requiring the underlying optimal control solution methods being able to accurately satisfy them.

A number of algorithms exist to solve many different classes of optimal control problems. In many of them the time axis is discretised or the input parametrised, resulting in a finite dimensional nonlinear program that can be solved *directly*. These approaches result in a discrete approximation of the optimal solution. The alternate of this approach, known as the *indirect approach*, solves the necessary conditions of optimality, transforming the problem into a set of differential equations to be solved as a boundary value problem (BVP). Numerical methods have been developed to solve this difficult class of differential equations, including the shooting and subsequently the multiple shooting methods. Both of these methods have been used to solve optimal control problems (Maurer and Gillessen, 1975). Orthogonal collocation is another powerful method that has been developed to solve BVPs. However, it does not seem to have gained acceptance as a viable method to indirectly solve optimal control problems, although its use in a direct approach is well known (Meadows and Rawlings, 1997; Cuthrell and Biegler, 1989). The indirect method of solving optimal control problems results in an extremely accurate *continuous* solution of the problem.

One of the key areas of this study is the indirect solution of optimal control problems using orthogonal collocation, with the ability to implement the solution in an NMPC scheme with stability guarantees. This requires that the solution method be able to solve optimal control problems of the form dictated by the NMPC strategy, including any stability constraints.

To further develop and investigate the application of NMPC to industry, a case study is useful. The mineral processing industry makes use of highly complex and nonlinear processes, with system variables that are difficult to measure. The primary refining process of flotation is certainly an example of this. Nonetheless, recent advances in measurement

technology, and better understanding of the relationships governing the flotation process have resulted in it becoming more amenable to dynamic analysis and advanced control. An important part of this study is the development of a dynamic flotation model, and the application of NMPC to flotation using this model.

The basic approach in this work in the development of the flotation model is along fundamental lines, although where necessary empirical relationships are used. The application of NMPC is not intended to be immediately implementable on an industrial scale and many problems that industrial implementation will face are simply ignored (e.g. state estimation, noisy signals). It is hoped that this work will provide an assessment of the potential of this approach and provide a basis for further study in this area, thus increasing interest in the advanced control of flotation circuits.

1.2 Thesis Overview

Chapter 2

The major focus at the start of this study was the solution of optimal control problems using orthogonal collocation. This chapter shows the development of the solution method, with details of the different classes of optimal control problems that have been successfully solved. It starts with a review of the major methods that have been or are being employed to solve these types of problems. The focus then shifts to the indirect method of solution, detailing the necessary conditions of optimality and showing how the boundary value problem is generated. The inclusion of different types of constraint is explained, and the expansion of the method to multi-input, multi-output problems. Finally examples are shown of these different types of problem, comparing the solution by orthogonal collocation with a known exact solution.

The final section in this chapter gives details about the converse HJB method, used to generate optimal control problems with known solution. Unconstrained and constrained optimal control problems can be generated with this method, which can then be used as sample problems.

Chapter 3

The model predictive control method is introduced in detail, and the prominent formulations are reviewed. The major stability results for NMPC are shown, detailing the assumptions used for stability and the changes to the general MPC formulation. An NMPC scheme is then described with a formulation making use of a class of optimal control problem that was solved in the previous chapter. This formulation requires stability constraints and additions to the cost function that can be solved using the method proposed in Chapter 2. A simulated application example of NMPC on a CSTR is then given, and the benefits of piecewise continuous control compared with conventional NMPC using piecewise constant control.

Chapter 4

This chapter outlines the development of the flotation model, beginning with a review of available steady-state models. The key parts of the flotation model are explained, and equations chosen that are most amenable to control. For these equations to be used in dynamic systems, dependence on state and input variables is necessary. Thus wherever possible models were chosen which include, or can be modified under assumptions to include, these characteristics. The expansion of steady-state models to include dynamics was done by performing a mass balance with an accumulation term. Many of the flotation models use kinetic rate equations which can be used directly in this mass balance without modification of the rate constant. However, certain models did require assumptions to be made, as well as modifications to the equation parameters, and the details of these assumptions and their operating regimes are given.

Chapter 5

The final element of this work is the application of NMPC to flotation. Flotation control has evolved through the years, and at present there is considerable interest in the application of advanced control strategies to flotation. The research into this subject is reviewed, to investigate the state of flotation control and reveal trends in the choice of control system description. An NMPC scheme is then applied to the first rougher cell of a simplified flotation system. The model developed in Chapter 3 is used both in the predictive controller and to analyse the closed-loop system response. A number of runs are performed, to show

the response to setpoint changes and the effect of disturbances. The NMPC strategy uses a model that is linearised (affine) in the input variables, although it is highly nonlinear in the state. Most of the runs are performed with exact plant/model matching, although a run is also performed using plant/model mismatch. In this case the plant is simulated using a model that is nonlinear in both the state and input variables, although the controller uses the model affine in the input variables (nonlinear in the state). This test investigates robustness of the controller. Finally, the nonlinear controller is compared to a conventional linear predictive controller (QDMC) to investigate the performance advantages of using a nonlinear controller.

Chapter 2

Optimal Control

2.1 Review of Current Approaches

Optimal control has a history about 360 years long, starting with two problems proposed by Galileo. The first was to calculate the shape of a heavy chain supported at either end by stationary points. The second was to determine the shape of a piece of wire that would cause a bead running along its length under gravity to move from one end to the other in the shortest time. Known as the brachistochrone problem, Galileo's own postulations on its solution were incorrect. In 1662, Fermat also raised the subject of optimal control by proposing that light always chooses the path through different media that will require the shortest amount of time.

Interest in optimal control was heightened when, in 1696, Johan Bernoulli issued a challenge to his contemporaries to solve the brachistochrone within the year. Five other mathematicians took up the challenge (Jakob Bernoulli, Leibnitz, l' Hopital, Tschirnhaus and Newton), and the results were published (including those of Johan Bernoulli) in 1697. Most of the approaches were geometric in nature.

As a result of this competition interest in problems of this form increased considerably, and in 1744 Euler published a book collecting all of the ideas at the time. In 1755 Lagrange wrote to Euler with an analytical approach to solving this type of problem, based on a perturbation (or variational) method. Using Lagrange's perturbation method, Euler's necessary condition was developed which came to be known as the "Euler-Lagrange equation". Euler adopted this approach and it was renamed the "*calculus of variations*".

A long history of development of the theory then occurred, with ideas improved and extended. In 1786 Legendre developed a scalar second order necessary condition of optimality, which was extended to the vector case by Clebsch. The Hamiltonian was developed by Hamilton, and Caratheodory was able to make further advancements using Hamilton's work together with the Legendre-Clebsch condition. In 1838 Jacobi produced a more compact formulation for the Hamiltonian, called the Hamilton-Jacobi equation. Still further work was done by Caratheodory and Weierstrass, until finally Lagrange multipliers were introduced to solve the problem. Pontryagin established the necessary conditions for optimality of this problem in his famous *"maximum principle"*. The details of the history of optimal control are summarised in a review paper by Sargent (Sargent, 2000)

From this time onwards there was an increasingly large number of publications, with different methods to tackle the various forms of constraints and problem formulations. There is still considerable interest and research activity today in optimal control theory.

Optimal control problems have been used in a wide range of applications and environments. In economics they have been applied to the optimal solution of market models, taking into account capital flows, production and the employment sector (Koslik and Breitner, 1997). They have been extensively used in the aerospace industry and the aviation field. Examples include the determination of the time and energy optimal flight paths of space re-entry vehicles, and the abort landing of aircraft in windshear conditions (Pesch, 1991). Optimal control theory has even been applied to the maximisation of the range of a hang glider (Bulirsch et al., 1993). In industry, optimal control problems are of interest as they form the core of the Model Predictive Control (MPC) algorithm (Garcia et al., 1989; Morari, 1994). MPC requires the rapid on-line solution of optimal control problems, often with constraints on the state and control variables. Due to the requirement of rapid problem solution, nonlinear MPC did not find industrial acceptance until recently. With the advent of powerful computers, and novel and efficient methods of solving nonlinear optimal control problems, nonlinear MPC is increasingly being adopted by industry.

2.1.1 Basics of optimal control

Kirk, 1970 describes the objective of optimal control theory as being

> *"to determine the control signals that will cause a process to satisfy the physical constraints and at the same time minimize (or maximize) some performance*

criterion."

The optimal control problems considered in this study consist of the minimisation of an objective function subject to the system dynamics and other constraint equations.

$$V(\mathbf{u}) = \Phi\left(\mathbf{x}(T)\right) + \int_0^T L\left(t, \mathbf{x}(t), \mathbf{u}(t)\right) dt \tag{2.1}$$

subject to :

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left(t, \mathbf{x}(t), \mathbf{u}(t)\right) \tag{2.2}$$

with initial condition :

$$\mathbf{x}(0) = \mathbf{x}_0 \tag{2.3}$$

Constrained by:

$$\mathbf{u}_{min} \leq \mathbf{u}(t) \leq \mathbf{u}_{max} \tag{2.4}$$

$$\psi(\mathbf{x}(T)) = \mathbf{0} \text{ or } \leq \mathbf{0} \tag{2.5}$$

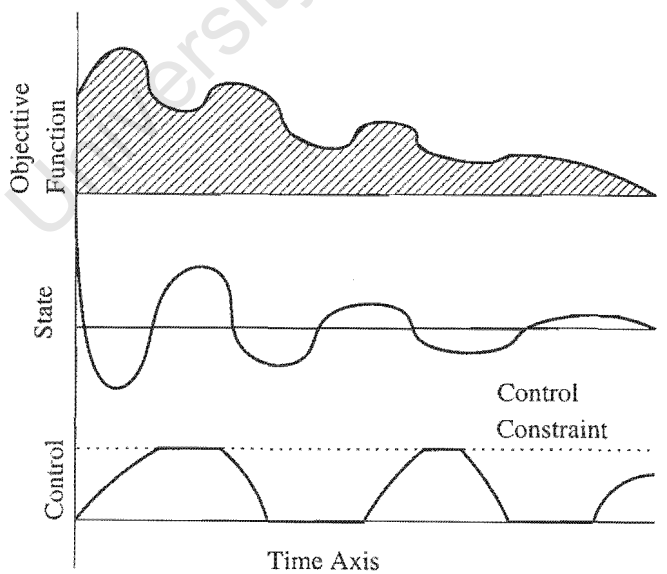

Figure 2.1: Graphical explanation of optimal control

The objective function (Equation 2.1) is an integral over time, with an additional penalty function operating at the terminal time included in the problem. The terminal time for the integral may or may not be set, depending on the type of problem. Only problems with fixed terminal times are considered in this work. The system dynamics (Equation 2.2) must be satisfied during the optimisation. Additional constraints are often included on the control or state trajectory, although only control constraints (in the form of Equation 2.4) are considered in this study. Constraints at the terminal time T may also be included (Equation 2.5). In a physical system, control constraints may represent valves saturating or being fully closed. State constraints could represent maximum temperatures or pressures of a system that must not be exceeded. Terminal state constraints are often included as they have useful stabilising properties (Mayne et al., 2000; Langson et al., 2000). Optimally satisfying these constraints is quite challenging, and has led to a number of different numerical formulations for the solution of these problems.

An important step forward was made with the development of the Euler-Lagrange equations, and Pontryagin's Maximum principle and necessary conditions. By using the Euler-Lagrange equations, an optimal control problem can be reformulated into a set of differential equations consisting of the state equations and a generated set of costate or *adjoint* equations. This boundary value problem (BVP) has a given set of initial values and a set of boundary conditions at the terminal time. A number of formulations make use of these equations in the solution of the optimal control problem (Section 2.1.2).

2.1.2 Methods of solution

Three basic methods of solving optimal control problems have emerged over the years: *complete discretisation* of the problem, *finite parameterisation* of the control trajectory, and the *solution of the two-point BVP* resulting from the Euler-Lagrange equations.

The **complete discretisation** approach is a direct method in which the problem is completely discretised along the time axis, resulting in a finite dimensional problem that can be solved using a NLP[1] solver. It is known as a direct method as the optimisation is achieved by the direct use of a numerical minimiser. The first attempts at this method used finite differences for the discretisation. However, Tsang (Tsang et al., 1975) and later Biegler

[1] A nonlinear program (NLP) is an optimisation problem where a scalar objective function is minimised (or maximised) subject to equality and or inequality constraints, where at least one of the constraints or objective function is nonlinear. A number of different classes of problems exist depending on the type of nonlinearity in the objective function and the constraints

and associates (Cuthrell and Biegler, 1987; Logsdon and Biegler, 1989) introduced the use of direct collocation. This involves fitting polynomials (usually cubic) between the discrete points along the state trajectory. The parameters of the NLP are the state and control values at the grid points.

Biegler's group made a number of advances, including the incorporation of error measures as constraints to decide the number and placing of additional collocation nodes. At the same time Betts and Huffman, and others (Betts and Huffman, 1992; Bauer et al., 1984) formulated the problem as an SQP (sequential quadratic program) and used a trust region algorithm to solve it.

The **finite parameterisation** method also results in a nonlinear program by parameterising the control trajectory. The objective function and constraint equations are evaluated by integrating the system equations, and their gradients by backwards integration of the adjoint equations. A simple example of this was demonstrated by Kelley, using a closely spaced grid of estimated control values. These were used to integrate the system equations, and the adjoint equations were then integrated backwards using the solution of the state equations to calculate a starting value. The results could then be used to find the gradients of the control values and use this to correct the grid of controls (Kelley, 1960). Sargent and associates have continued developing the control parameterisation method, improving on Kelly's method by removing the tie between the control grid spacing and the integration step. This allowed the use of a coarser grid, with piecewise constant or piecewise linear controls, and subsequently a much smaller optimisation problem. They have extended the method to handle multi-stage systems, high index differential algebraic equations and state inequalities (Sargent, 2000).

Solving optimal control problems by **solution of the Euler-Lagrange** equations is an indirect approach, involving the solution of a boundary value problem. The optimisation occurs by solving the necessary conditions for optimality, rather than directly applying a numerical minimiser. Section 2.2.1 details the derivation of the Euler-Lagrange equations, and the generation of the BVP. The indirect approach was one of the first methods used to solve optimal control problems, although at first it was only used to solve problems without constraints. Early attempts with this method used the shooting method, a numerical method to solve BVPs.

Boundary value problems are difficult to solve due to there being a split set of conditions on the differential equations, often with conditions at the start and at the end of the

integration. In linear systems, the equations can be transformed to remove the split in the boundary conditions. However, with nonlinear equations this is not possible. The shooting method was a first attempt at solving problems of this form, by estimating the initial values of those equations not explicitly known, and then integrating the system of equations forwards. Corrections to the initial estimate would be made by comparing the final value to the boundary conditions at that point. The estimation of the initial values is analogous to *"aiming"* and the solution of the equations would be a *"shot"*. The *"target"* is the boundary conditions which must be satisfied. The initial values are re-estimated from the residuals at the end point, analogous to readjusting the aim by examining how close to the target the shot lands. This method has been found to be extremely sensitive to the initial estimate, even to the point of being regarded as unstable. As a result, the "multiple shooting" method was developed by Bulirsch and associates (Stoer and Bulirsch, 1980). This method subdivides the time interval and then estimates the starting values for each subinterval. The starting values for each subinterval are then corrected based on the end point results. However, an initial estimate is needed for each subinterval.

Another indirect method to solve the BVP, by Miele and associates (Miele et al., 1973; Miele and Wang, 1993), is quasilinearisation. In this method a sequence of linear approximations of the system is used to approximate a nonlinear solution. The first step of the algorithm is to linearise the system equations around a guessed starting trajectory. By linearising the equations, both the linearised system and adjoint equations can be integrated forward in time, and this is done for sets of initial values. From a linear combination of these sets a solution that satisfies the boundary conditions is calculated, the system re-linearised around this trajectory, and the process repeated. Stopping criteria for this method include comparing the difference between two successive linearised trajectories to test convergence, and how well the proposed initial values of the approximated solution satisfy the boundary conditions of the nonlinear problem.

Recently hybrid methods, combining two or more of the above methods, have started to play a significant role. An example of this (Bulirsch et al., 1993) is the use of direct collocation on the problem initially, which provides an initial guess for an indirect method using the multiple shooting algorithm. Bock (Bock and Plitt, 1984) also used a hybrid method, combining the use of control parameterisation with a method similar to multiple shooting.

Currently success is being had with the use of slack variables and interior-point methods to convert the problem into a smooth two-point BVP for a differential algebraic (DAE)

system. Constraints are dealt with using barrier functions, which approximate non-smooth inequality constraints (Bell and Sargent, 2000). Powerful solvers for DAE systems are available to solve problems of this form.

2.2 Orthogonal Collocation Based Approach

The approach in this study is an indirect one, solving the first-order necessary conditions of the optimal control problem using orthogonal collocation. Similar to the shooting and multiple shooting methods, orthogonal collocation is a numerical method used to solve the BVP. In this case it is used to solve the BVP resulting from the use of the Euler-Lagrange equations. The method proposed is not to be confused with the direct methods using collocation. These methods discretise the optimal control problem and apply collocation equations to the resulting system, solving using a NLP solver.

A primary point of difference in this method is the use of a generic orthogonal collocation routine to solve the BVP. Specialised optimal control software and algorithms were *not* used, and the BVP was generated using the Euler-Lagrange equations.

Orthogonal collocation differs from the shooting and multiple shooting algorithms in that it attempts to fit splines satisfying the system equations at points corresponding to roots of an orthogonal polynomial within a subdivided mesh throughout the time interval. It does not guess initial values and integrate forwards.

This approach was extended to include optimal control problems with constraints along the control trajectory, and terminal state equality and inequality constraints.

2.2.1 Indirect Method: Euler-Lagrange Equations

As shown in Section 2.1.1, an optimal control problem consists of an objective function to be minimised, constrained by the system dynamics. The initial state of the system is given, and additional constraints on the control and/or state variables may be included.

As shown previously in Equations 2.1 to 2.3, the basic form is as follows:

$$\begin{aligned} V(\mathbf{u}) &= \Phi\left(\mathbf{x}(T)\right) + \int_0^T L\left(t, \mathbf{x}(t), \mathbf{u}(t)\right) dt \\ \text{subject to :} & \\ \dot{\mathbf{x}}(t) &= \mathbf{f}\left(t, \mathbf{x}(t), \mathbf{u}(t)\right) \\ \text{with initial condition :} & \\ \mathbf{x}(0) &= \mathbf{x}_0 \end{aligned}$$

$\mathbf{x}$ and $\mathbf{u}$ are vectors of n state and m control variables respectively. $\mathbf{f}$ is a set of n equations, continuously differentiable in all its variables. The initial condition is a set in $\mathbb{R}^n$ describing the state of the system at $t = 0$. Due to the system being autonomous[2], the time axis can be shifted for processes not starting at $t_0 = 0$. For the present, no additional constraints will be included.

The control $\mathbf{u}$ is chosen to minimise $V(\mathbf{u})$ over the time range $[0, T]$. $\Phi(\mathbf{x}(T))$, the terminal penalty function, and L must be continuously differentiable with respect to all their arguments. Equation 2.1 is regarded as being explicitly a function of $\mathbf{u}$ only, as $\mathbf{x}$ is implicitly determined from $\mathbf{u}$.

The derivation of the Euler-Lagrange equations follows Teo (1991) . Using Lagrange multipliers, the system dynamics (Equation 2.2) can be appended to the cost function:

$$\bar{V}(\mathbf{u}) = \Phi\left(\mathbf{x}(T)\right) + \int_0^T \left\{ L\left(t, \mathbf{x}(t), \mathbf{u}(t)\right) + \boldsymbol{\lambda}(t)' \left[\mathbf{f}\left(t, \mathbf{x}(t), \mathbf{u}(t)\right) - \dot{\mathbf{x}}(t)\right] \right\} dt \quad (2.6)$$

Introducing the Hamiltonian $H(t, \mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t))$ simplifies Equation 2.6, resulting in Equation 2.8.

$$H\left(t, \mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t)\right) = L\left(t, \mathbf{x}(t), \mathbf{u}(t)\right) + \boldsymbol{\lambda}(t)' \mathbf{f}\left(t, \mathbf{x}(t), \mathbf{u}(t)\right) \quad (2.7)$$

$$\bar{V}(\mathbf{u}) = \Phi\left(\mathbf{x}(T)\right) + \int_0^T \left\{ H\left(t, \mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t)\right) - \boldsymbol{\lambda}(t)' \dot{\mathbf{x}}(t) \right\} dt \quad (2.8)$$

[2]independent of time (time invariant)

Equation 2.8 can then be integrated by parts to give:

$$\begin{aligned}\bar{V}(\mathbf{u}) &= \Phi\left(\mathbf{x}(T)\right) - \boldsymbol{\lambda}(T)'\mathbf{x}(T) + \boldsymbol{\lambda}(0)'\mathbf{x}(0) + \\ &\int_0^T \left\{ H\left(t, \mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t)\right) + \dot{\boldsymbol{\lambda}}(t)'\mathbf{x}(t) \right\} dt \end{aligned} \tag{2.9}$$

With a small variation $\delta\mathbf{u}$ in $\mathbf{u}$, the resultant first-order variations in $\mathbf{x}$ and $\bar{V}$ are $\delta\mathbf{x}$ and $\delta\bar{V}$ respectively. $\delta\bar{V}$ is found by the chain rule to be:

$$\begin{aligned}\delta\bar{V} &= \left[\frac{\partial\Phi}{\partial\mathbf{x}} - \boldsymbol{\lambda}(T)'\right]\delta\mathbf{x}(T) + \boldsymbol{\lambda}(0)'\delta\mathbf{x}(0) + \\ &\int_0^T \left\{ \left[\frac{\partial H}{\partial\mathbf{x}} + \dot{\boldsymbol{\lambda}}(t)'\right]\delta\mathbf{x}(t) + \frac{\partial H}{\partial\mathbf{u}}\delta\mathbf{u}(t) \right\} dt \end{aligned} \tag{2.10}$$

Note that in Equation 2.10 the arguments are omitted to simplify the formulation.

A necessary condition for a local minimum is that $\delta\bar{V}$ is zero for a small variation $\delta\mathbf{u}$. Solving Equation 2.10 the following relationships are found:

$$\dot{\boldsymbol{\lambda}}(t)' = -\frac{\partial H\left(t, \mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t)\right)}{\partial\mathbf{x}} \tag{2.11}$$

$$\boldsymbol{\lambda}(T)' = \frac{\partial\Phi\left(\mathbf{x}(T)\right)}{\partial\mathbf{x}} \tag{2.12}$$

Due to the initial condition $\mathbf{x}(0)$ being fixed, $\delta\mathbf{x}(0)$ is zero, and Equation 2.10 becomes:

$$\delta\bar{V} = \int_0^T \left\{ \frac{\partial H\left(t, \mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t)\right)}{\partial\mathbf{u}}\delta\mathbf{u}(t) \right\} dt \tag{2.13}$$

Thus a local minimum is found if

$$\frac{\partial H\left(t, \mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t)\right)}{\partial\mathbf{u}} = 0 \tag{2.14}$$

for the entire time range $t \in [0, T]$. It is important to note that this is only the case if there are no constraints on the input variable. If the input variable is constrained, then the Pontryagin Minimum[3] Principle is used.

[3]The Pontryagin *Maximum* principle and Pontryagin *Minimum* principle are synonymous

The Euler-Lagrange equations consist of the system dynamics (Equation 2.2 and 2.3), the adjoint (or costate) dynamics (Equation 2.11 and 2.12), and Equation 2.14. The result is a set of boundary value differential equations, with initial conditions at $t = 0$ and boundary conditions at $t = T$. In this study these equations are solved by orthogonal collocation, using the `FORTRAN` package `COLSYS`.

2.2.2 Control Constraints

The Euler-Lagrange equations given above show that if $\frac{\partial H}{\partial \mathbf{u}} = 0$ for all of $t \in [0, T]$ then an optimal solution has been found. However, if the problem contains control constraints then a solution for Equation 2.14 may not lie within the constrained region.

For the case where the control is constrained to lie within a given subset $\mathbb{U}$ of $\mathbb{R}^m$, Equation 2.14 is not required to hold if the optimal solution to the Hamiltonian lies on the boundary of $\mathbb{U}$. This is known as the *Pontryagin Minimum Principle* and can be expressed as:

$$H\left(t, \mathbf{x}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t)\right) = \min_{\boldsymbol{\nu} \in \mathbb{U}} H\left(t, \mathbf{x}^*(t), \boldsymbol{\nu}, \boldsymbol{\lambda}^*(t)\right) \tag{2.15}$$

where $\mathbf{u}^*(t)$ is an optimal control solution (that satisfies $\mathbb{U}$) at time t, with $\mathbf{x}^*(t)$ and $\boldsymbol{\lambda}^*(t)$ the corresponding state and costate values.

Using this principle, an optimal *constrained* control value is either within the constrained region (and none of the constraints are active), or it is on the surface of the constrained region (with one or more constraints active).

To solve optimal control problems with control constraints, this principle was used. At every time instant the optimal control input was calculated using Equation 2.14, and tested to ensure that it fell within the constrained region. If a constraint was violated, the optimal input was determined by finding the minimum of the Hamiltonian along the constraint boundary. This was handled as a combinatorial problem, where the Hamiltonian was calculated for every face, edge and vertex of the constrained region. The control value that resulted in the smallest Hamiltonian was used at that time instant.

Figure 2.2 graphically demonstrates three possible optimal control solutions in a two dimensional system, both with active and inactive constraints.

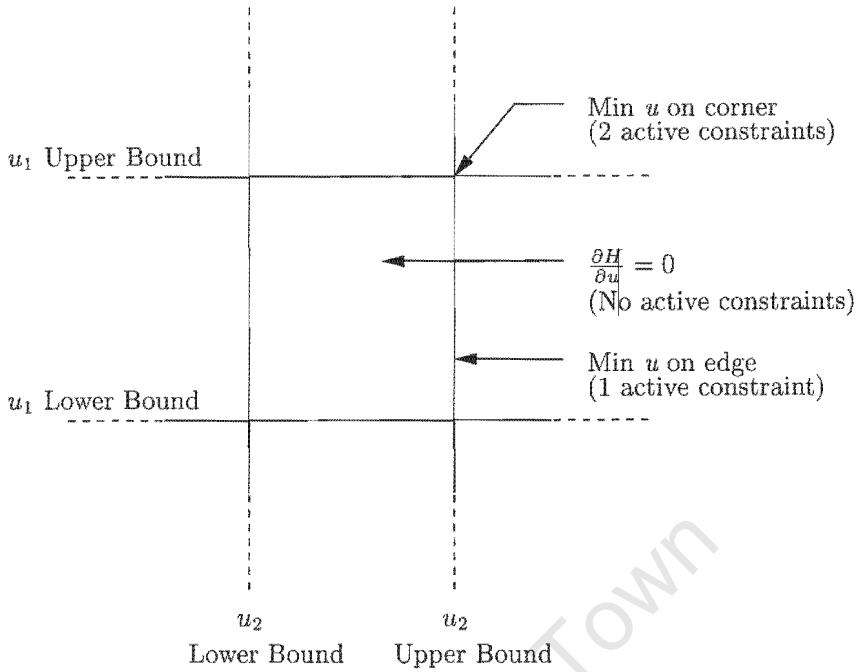

Figure 2.2: Examples of optimal solutions in a two-input system

2.2.3 Terminal State Constraints

An important class of optimal control problems are those that include terminal state constraints, in particular problems with terminal state inequality constraints. These are systems in which the state variables at the terminal time must optimally satisfy some constraint condition. This class of control problems has important stabilising characteristics for Nonlinear Model Predictive Control (Langson et al., 2000; Chen and Allgöwer, 1998).

The constraints considered force the system state variables to lie within a defined region at the problem's terminal time. In problems with the following objective function:

$$V = \Phi\left(\mathbf{x}(T)\right) + \int_0^T L(t, \mathbf{x}(t), \mathbf{u}(t))dt \tag{2.16}$$

these constraints are of the form:

$$\mathbf{g}(\mathbf{x}(T)) - \mathbf{c} \leq \mathbf{0} \tag{2.17}$$

In Equation 2.16, $\Phi(\mathbf{x}(T))$ represents a terminal penalty cost, and $L(t, \mathbf{x}(t), \mathbf{u}(t))$ is the stage cost.

Problems of this form can be solved using a modified set of boundary conditions as described in Kirk (1970). The boundary conditions for a problem with the terminal state constrained to lie on a surface are:

$$\mathbf{x}^*(0) = \mathbf{x}_0 \tag{2.18}$$

$$\frac{\partial \Phi}{\partial \mathbf{x}}(\mathbf{x}^*(T)) - \boldsymbol{\lambda}^*(T) = \sum_{i=1}^{k} d_i \left[\frac{\partial g_i}{\partial \mathbf{x}}(\mathbf{x}^*(T))\right] \tag{2.19}$$

$$\mathbf{g}(\mathbf{x}^*(T)) = 0 \tag{2.20}$$

where:

$\boldsymbol{\lambda}$	Lagrange multiplier
$\mathbf{g}$	Surface equality constraint $\mathbf{g}(\mathbf{x}(t)) = \mathbf{0}$
d_i	Multiplier variable (introduced)
k	Number of constraint equations

To solve the BVP, $2n$ boundary conditions are required due to n state and n adjoint first-order differential equations. Equations 2.18 - 2.20 are $2n + k$ equations in the variables d_i (k of them), $\mathbf{x}^*(0)$, $\boldsymbol{\lambda}^*(T)$ and $\mathbf{x}^*(T)$, providing sufficient conditions for the two-point BVP to be solved.

These boundary conditions only describe the case where the terminal states are forced to lie on a surface (i.e. equality constraint). The inequality constrained problem was solved by testing each iteration of the terminal state values. With each call to the boundary condition function, the constraint equations were solved: if the terminal state values lay outside the constrained region, then the above boundary conditions were used. However, if the boundary value lay within the constrained region, then the boundary conditions of an unconstrained problem were used. This forced the solution to lie on or within the constrained region.

2.2.4 COLSYS program

The FORTRAN package COLSYS (Ascher et al., 1981) was used to solve the boundary value problem resulting from the Euler-Lagrange equations. This is a powerful implementation of the orthogonal collocation routine for solving this form of differential equation. It solves systems of mixed order multipoint BVPs, using spline collocation at Gaussian points.

The differential equations are coded in FORTRAN, with separate subroutines for the equations and the boundary conditions. A Jacobian of each of these functions must also be supplied. The routine makes provision for the supply of an initial guess subroutine, which can speed up problem solution and is necessary in cases where the equations are not defined for certain state values.

COLSYS is able to solve mixed-order systems of multipoint boundary value problems. It does not explicitly convert higher-order equations to first-order. COLSYS is designed to handle up to fourth-order equations, with the set of equations being supplied in increasing order. Thus typically the equation system is a set of n equations of order $1 \leq m_1 \leq m_2 \leq \ldots \leq m_n \leq 4$ where m_i is the order of the i^{th} equation:

$$y_i^{(m_i)} = F_i(x, \mathbf{z}(\mathbf{y})) \tag{2.21}$$

$\mathbf{z}(\mathbf{y})$ is the vector of unknowns that would occur if Equation 2.21 was explicitly converted to a first-order system. Thus if $n = 3$ with equation set $[y_1', y_2'', y_3^{(iv)}]$, $\mathbf{z}(\mathbf{y})$ would be $[y_1, y_2, y_2', y_3, y_3'', y_3''']$.

To solve the system, $m^* = \sum_{i=1}^{n} m_i$ nonlinear multipoint separated boundary conditions are required:

$$G_j(\zeta_j, \mathbf{z}(\mathbf{y})) = 0, \quad j = 1, \ldots m^* \tag{2.22}$$

The system is solved over the range $a \leq x \leq b$, and ζ_j is the location of the j^{th} boundary condition within this range.

Four subroutines are required for COLSYS to solve the BVP. They are the set of differential equations (in subroutine FSUB), the Jacobian of FSUB (in subroutine DFSUB), the set of boundary conditions (in subroutine GSUB), and the Jacobian of these boundary conditions (in subroutine DGSUB). A fifth subroutine can be supplied, GUESS, containing the initial guess (approximation) of the solution.

FSUB and GSUB follow the form of Equations 2.21 and 2.22. FSUB accepts x and $\mathbf{z}(\mathbf{y})$, and returns $\mathbf{F}_i$, whereas GSUB returns G_j using parameters j and $\mathbf{z}(\mathbf{y})$

The full matrix solution of the Jacobian contained in DFSUB is returned to COLSYS in the

following form:

$$DF = \begin{bmatrix} \frac{\partial F_1}{\partial z_1} & \frac{\partial F_1}{\partial z_2} & \cdots & \frac{\partial F_1}{\partial z_{m^*}} \\ \frac{\partial F_2}{\partial z_1} & \frac{\partial F_2}{\partial z_2} & & \\ \vdots & & \ddots & \\ \frac{\partial F_n}{\partial z_1} & & & \frac{\partial F_n}{\partial z_{m^*}} \end{bmatrix} \tag{2.23}$$

GSUB only returns the j^{th} boundary value as requested by the COLSYS subroutine. Similarly, DGSUB only returns the the vector of the j^{th} row of the Jacobian:

$$DG_j = \left[\begin{array}{ccc|c} \frac{\partial G_j}{\partial z_1} & \frac{\partial G_j}{\partial z_2} & \cdots & \frac{\partial G_j}{\partial z_{m^*}} \end{array} \right] \tag{2.24}$$

For the specific case of optimal control problems with n state and n costate equations ($2n$ in total), x represents the time axis t, and the equations are all first-order. This results in $\mathbf{z}(\mathbf{y})$ being the vector of state and costate variables.

Similarly, for this class of problems ζ from Equation 2.22 is a vector in which half the entries are at $t = 0$ and half at $t = T$ for the state and costate equations respectively.

An example of the FORTRAN driver program to solve Case 1 (Section 2.3.1) is included in Appendix B.

An interface for COLSYS was written to allow it to be used in the well known mathematical programming language MATLAB[4]. This interface made use of the MATLAB *Application Programming Interface*, which can be used to allow both C and FORTRAN programs to be used in the MATLAB environment. MATLAB is an interpreted language, and as such is not as fast as a compiled language. Using the MATLAB-COLSYS routine did increase the execution time required to solve problems, however development in the MATLAB environment is easier and simpler than in FORTRAN. This resulted in less lost time due to errors and programming complexity. Many of the MATLAB optimisation and simulation tools were also useful for testing and development. Nonetheless many of the larger problems were written entirely in FORTRAN.

[4]MATLAB is a registered trademark of The MathWorks, Inc

2.3 Application Examples

To test the method of indirectly solving optimal control problems by orthogonal collocation, a number of different sample problems were solved. Problems with different forms and constraints were generated or found from literature, and then solved using the method described in Section 2.2. To test against the correct solution, examples from literature were only taken if an exact solution was included, or problems were generated using the co-HJB and co-CHJB method to be described in Section 2.4.

2.3.1 Case 1: Single input, single output, unconstrained

A simple problem is shown below, with only one state and one input variable. The application of the Euler-Lagrange equations (Section 2.2) results in two differential equations - describing the dynamics of the state and costate (adjoint) variables. The state initial condition at $t = 0$ and the costate boundary condition at $t = T$ are given.

The problem was generated using the co-HJB method (Section 2.4).

$$V(u) = \min_{u(\cdot)} x^2(T) + \int_0^T \left\{x^2 + u^2\right\} dt \tag{2.25}$$

subject to :

$$\dot{x}(t) = 12.5x(1-x)^2 e^{2x} - \frac{1}{2}x + 5(1-x)e^x u \tag{2.26}$$

with initial condition :

$$x(0) = x_0 \tag{2.27}$$

The Hamiltonian is thus:

$$H = x^2 + u^2 + \lambda\left(12.5x(1-x)^2 e^{2x} - \frac{1}{2}x + 5(1-x)e^x u\right) \tag{2.28}$$

Using the Euler-Lagrange equations results in the following additional adjoint system:

$$\begin{aligned}\dot{\lambda} &= 2x + \lambda(12.5(1-x)^2e^{2x} - 25x(1-x)e^{2x} + \\ &\quad 25x(1-x)^2e^{2x} - \frac{1}{2} - 5e^x u + 5(1-x)e^x u) \quad (2.29)\end{aligned}$$

with boundary condition :

$$\lambda(T) = 2x(T) \quad (2.30)$$

The control input is found by solving $\frac{\partial H}{\partial u} = 0$:

$$u = -\frac{5}{2}\lambda e^x(1-x) \quad (2.31)$$

The problem was designed with the following optimal solution as a function of the state variable:

$$u^* = 5(x^2 - x)e^x \quad (2.32)$$

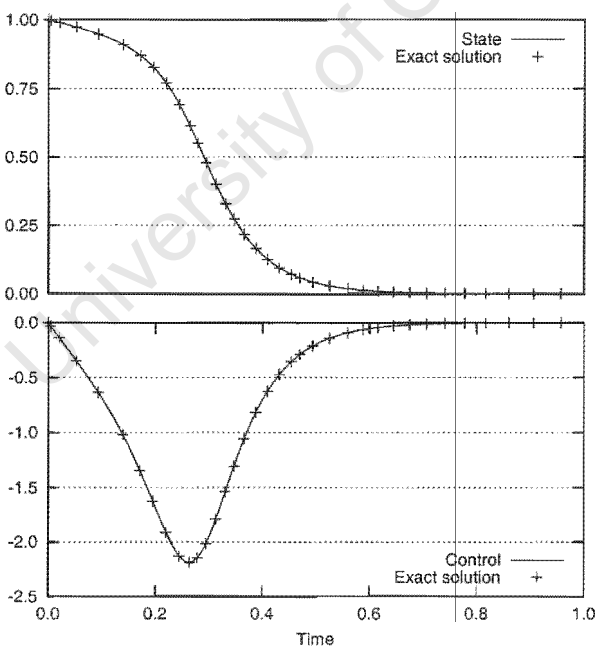

Figure 2.3: Case 1 - Single input, single output optimal control problem example

Figure 2.3 is a plot of the solution of this system, with the exact solution plotted as discrete points. The continuous line represents the solution found by `COLSYS` and the

Euler-Lagrange equations. A highly accurate solution results from this simple system. This example was useful as a first test case which led the way to solving more complex cases and larger problems.

2.3.2 Case 2: Multi input, multi output, unconstrained

A **two state, three input** unconstrained problem was generated using the co-HJB method as shown in Doyle *et al*, 1996.

Minimise:

$$V(\mathbf{u}) = V^*(T) + \int_0^T \left\{\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{u}(t)\right\} dt \tag{2.33}$$

$$V^*(t) = \mathbf{x}^T(t)\mathbf{P}\mathbf{x}(t) \tag{2.34}$$

subject to:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t)) + \mathbf{g}(\mathbf{x}(t))\mathbf{u}(t) \tag{2.35}$$

where:

$$\mathbf{P} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{2.36}$$

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{2.37}$$

$$\mathbf{g}(\mathbf{x}(t)) = \begin{bmatrix} 1+x_1^2 & x_1+x_2 & x_2^2 \\ x_1 & x_1-x_2 & 0 \end{bmatrix} \tag{2.38}$$

$$\boldsymbol{\gamma}(\mathbf{x}(t)) = \frac{1}{2}\begin{bmatrix} x_2 \\ -x_1 \end{bmatrix} \tag{2.39}$$

$\mathbf{f}(\mathbf{x}(t))$ is calculated using the following equation (Doyle et al., 1996):

$$\mathbf{f}(\mathbf{x}(t)) = \frac{1}{2}\left(\mathbf{g}(\mathbf{x}(t))\mathbf{g}^T(\mathbf{x}(t))\mathbf{P} - \mathbf{P}^{-1}\mathbf{Q}\right)\mathbf{x}(t) + \mathbf{P}^{-1}\boldsymbol{\gamma}(\mathbf{x}(t)) \tag{2.40}$$

The optimal input is then:

$$\mathbf{u}^*(t) = -\mathbf{g}^T(\mathbf{x}(t))\mathbf{P}\mathbf{x}(t) \tag{2.41}$$

The equations were chosen simply to ensure that the system is nonlinear in the state variables. The costate equations and boundary conditions were calculated according to the Euler-Lagrange equations (Section 2.2.1), and an analytical Jacobian (for the COLSYS routine) was used. These equations are not included due to the straightforward but tedious nature of the calculations.

Figure 2.4 shows the solution of this unconstrained MIMO problem. The exact solution is represented by the discrete markers, and the calculated solution is shown by the continuous lines.

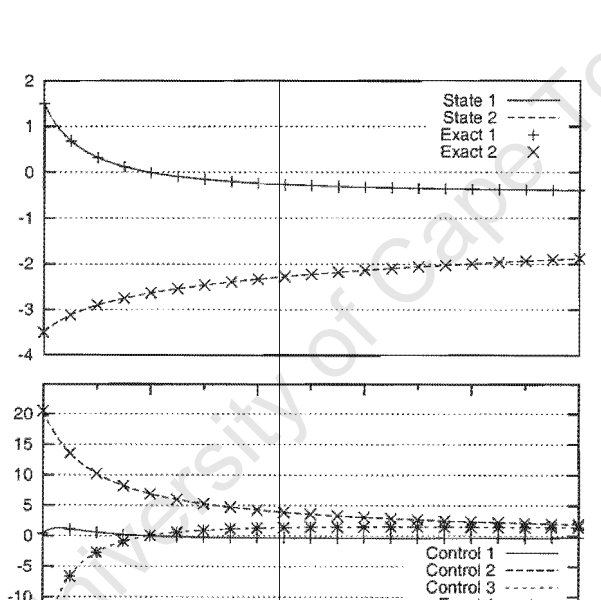

Figure 2.4: Case 2 - Unconstrained MIMO problem

2.3.3 Case 3: MIMO, with input constraints

This problem is of similar complexity to the previous one (Section 2.3.2), but is constrained in the input variables. It is a problem with **two state** and **three input** variables. It was generated using the converse constrained HJB (co-CHJB) method (Nevistic, 1997), as shown in Section 2.4.2. The problem is shown below.

Minimise:

$$V(\mathbf{u}) = V^*(T) + \int_0^T \left\{ q(\mathbf{x}(t)) + \mathbf{u}^T(t)\mathbf{u}(t) \right\} dt \tag{2.42}$$

$$V^*(t) = x_1^2 - x_1 \tag{2.43}$$

$$q(\mathbf{x}(t)) = \left(\frac{\partial V^*(\mathbf{x})}{\partial \mathbf{x}} \right)^T \mathbf{h}(\mathbf{x}(t)) \tag{2.44}$$

subject to:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t)) + \mathbf{g}(\mathbf{x}(t))\mathbf{u}(t) \tag{2.45}$$

$$|\,\mathbf{u}\,| \leq \boldsymbol{\alpha} \tag{2.46}$$

where:

$$\mathbf{f}(\mathbf{x}(t)) = \begin{bmatrix} x_2 \\ -x_1 \end{bmatrix}$$

$$\mathbf{g}(\mathbf{x}(t)) = \begin{bmatrix} -\beta_1 x_2^2 & x_2 - x_1 & -x_1 \\ 0 & -x_1 & \beta_2 x_2 (1 - x_1^2) \end{bmatrix}$$

$$\mathbf{h}(\mathbf{x}) = -\left(\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}'(\mathbf{x}) + \frac{\left(\frac{V^*(\mathbf{x})}{\partial \mathbf{x}} \right)^T \mathbf{u}'^T(\mathbf{x})\mathbf{u}'(\mathbf{x})}{\left\| \left(\frac{V^*(\mathbf{x})}{\partial \mathbf{x}} \right) \right\|^2} \right)$$

$$\mathbf{u}'(\mathbf{x}) = -\frac{1}{2} \left(\frac{V^*(\mathbf{x})}{\partial \mathbf{x}} \right) \mathbf{g}(\mathbf{x})$$

The optimal input ($\mathbf{u}^*(\mathbf{x})$) around which this problem was designed is described using the saturation operator (Section 2.4.2):

$$\mathbf{u}^*(\mathbf{x}) = sat_\alpha \left(-\frac{1}{2} \mathbf{g}(\mathbf{x}(t))^T \left(\frac{V^*(\mathbf{x})}{\partial \mathbf{x}} \right)^T \right) \tag{2.47}$$

As in the above sections, the costate equations and boundary conditions were calculated according to the Euler-Lagrange equations (Section 2.2.1), but are not included due to their complexity and length.

Figure 2.5 shows the solution of this problem. As with the previous cases, the exact

solution is represented by the discrete markers, and the calculated solution is shown by the continuous lines.

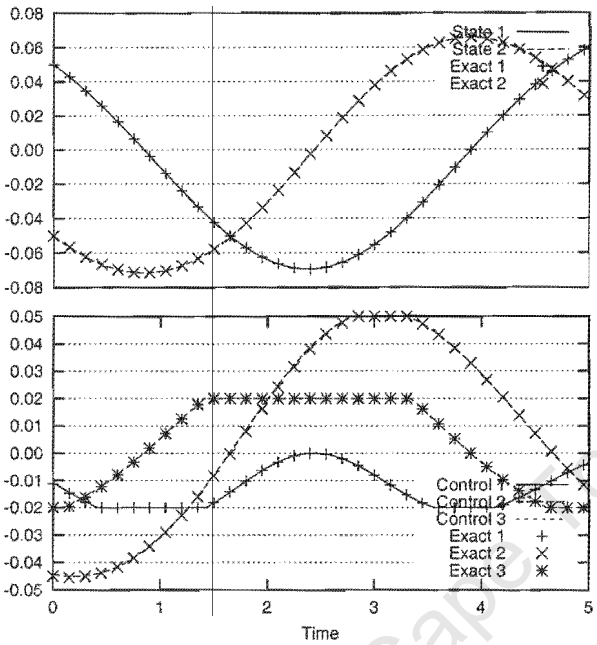

Figure 2.5: Case 3 - Constrained MIMO problem

2.3.4 Case 4: Terminal state constraints

This is a linear example from the literature with known solution, and is used as a test case in which the solution can be checked. It is taken from Kirk (1970), and although linear

and extremely simple, it is useful as a test of the method.

$$V(u) = \min_{u(\cdot)} \int_0^T \left\{ \frac{1}{2} u(t)^2 \right\} dt \tag{2.48}$$

subject to :

$$\dot{x}_1(t) = x_2(t) \tag{2.49}$$

$$\dot{x}_2(t) = -x_2(t) + u(t) \tag{2.50}$$

with initial condition :

$$x(0) = x_0 \tag{2.51}$$

and terminally constrained to lie on :

$$x_1(t) + 5x_2(t) = 15 \tag{2.52}$$

The adjoint system of equations is calculated using the Euler-Lagrange equations, and the boundary conditions as shown in Section 2.2.3:

$$\dot{\lambda}_1 = 0 \tag{2.53}$$

$$\dot{\lambda}_2 = -\lambda_1(t) + \lambda_2(t) \tag{2.54}$$

with boundary conditions :

$$BC_1 = x_1(0) - x_1^0 \tag{2.55}$$

$$BC_2 = x_2(0) - x_2^0 \tag{2.56}$$

$$BC_3 = \lambda_1(T) - \frac{1}{5}\lambda_2(T) \tag{2.57}$$

$$BC_4 = x_1(T) + 5x_2(T) - 15 \tag{2.58}$$

The control input is calculated by solving $\frac{\partial H}{\partial u} = 0$ to give:

$$u = -\lambda_2 \tag{2.59}$$

The analytical solution of this problem is published in Kirk (1970), in terms of the evolution of the state variables as a function of time. Figure 2.6 compares the calculated state variable solution with the published analytical solution. The optimal control input is calculated from the state solution using Equation 2.59. The terminal state satisfies the terminal state equality constraint (Equation 2.52).

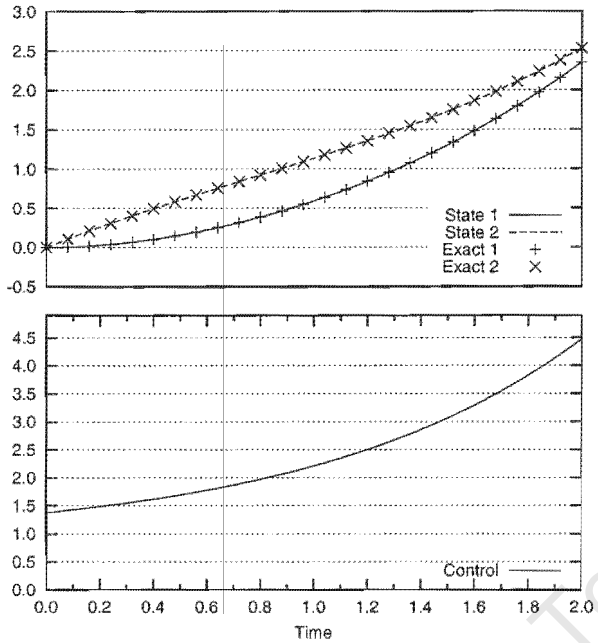

Figure 2.6: Case 4 - Terminal state constrained optimal control problem (Kirk, 1970)

2.4 Sample Problem Generation - co-HJB and co-CHJB method

The co-HJB method, or *converse Hamilton-Jacobi-Bellman* method (Doyle et al., 1996) generates optimal control problems with known solution. The heart of this method is the *dynamic programming approach* to solving optimal control problems. Dynamic programming uses the Hamilton-Jacobi-Bellman (HJB) partial differential equation, which can be solved analytically in certain special cases. In essence, this method solves the HJB equation *backwards*, starting with an optimal solution and designing a dynamic system and optimal control to satisfy the optimal solution. Optimal control problems can thus be generated to test solution methods against a known optimal solution.

The co-CHJB (converse *constrained* HJB) method (Nevistic, 1997) is an extension of the above method to generate optimal control problems constrained in the control variable. The details of both these methods can be found in Doyle *et al* (1996) and Nevistic (1997) but the procedure to generate problems using these methods is summarised below.

2.4.1 Converse Hamilton-Jacobi-Bellman method (unconstrained)

This method makes use of the Hamilton-Jacobi-Bellman (HJB) partial differential equation for solving optimal control problems. Given an optimal value function and performance objective, the HJB equation is used to determine the system dynamics for which the value function corresponds to the solution of the optimal control problem.

Optimal control problems with known solution are generated by specifying the optimal infinite horizon value function $V^*(\mathbf{x})$, a state cost contribution to the performance objective, $q(\mathbf{x})$, and part of the system dynamics $\mathbf{g}(\mathbf{x})$. For the multi-input, multi-output case, an additional function $\boldsymbol{\gamma}(\mathbf{x})$ may need to be specified. The optimal value function $V^*(\mathbf{x})$ returns the optimal infinite horizon cost of the system evolving from $\mathbf{x}$.

The method returns the system dynamics in the form

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{g}(\mathbf{x}(t))\mathbf{u}(t)$$

by determining the function $\mathbf{f}(\mathbf{x})$. The optimal control input is then determined as a function of $\mathbf{x}$ using $V^*(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$.

This method treats the problem given by:

$$\min_{\mathbf{u}(\cdot)} V(\mathbf{u}) = \int_0^\infty \left\{ q(\mathbf{x}(t)) + \mathbf{u}^T(t)\mathbf{u}(t) \right\} dt \tag{2.60}$$

with the following system dynamics:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{g}(\mathbf{x}(t))\mathbf{u}(t) \tag{2.61}$$

$$\mathbf{x}(0) = \mathbf{x_0} \tag{2.62}$$

In this case, $\mathbf{u}(t)$ must be piecewise continuous, and is not constrained. $V^*(\mathbf{x})$ represents the optimal solution of Equation 2.60 evolving from $\mathbf{x}$, subject to the system dynamics of Equation 2.61. Substituting into the Hamilton-Jacobi-Bellman equation (Doyle et al., 1996) results in:

$$V_{\mathbf{x}}^*(\mathbf{x})\mathbf{f}(\mathbf{x}) - \frac{1}{4}V_{\mathbf{x}}^*(\mathbf{x})\mathbf{g}(\mathbf{x})\mathbf{g}^T(\mathbf{x})V_{\mathbf{x}}^{*T}(\mathbf{x}) + q(\mathbf{x}) = 0 \tag{2.63}$$

Note that $V^*(\mathbf{x}) = \min_{\mathbf{u}(\cdot)}[V(\mathbf{u})]$ and $V_{\mathbf{x}}^* = \frac{\partial V^*(\mathbf{x})}{\partial \mathbf{x}}$.

Using Equation 2.63, $\mathbf{f}(\mathbf{x})$ can be found. For the simple case of a system with one state and one control variable, $f(x)$ can be found using:

$$f(x) = \frac{\frac{1}{4}(V_x^*(x)g(x))^2 - q(x)}{V_x^*(x)} \tag{2.64}$$

For MIMO systems, the solution of Equation 2.63 is more difficult. A solution is obtained using $q(\mathbf{x}) = \mathbf{x}^T\mathbf{Q}\mathbf{x}$ and $V(\mathbf{x}) = \mathbf{x}^T\mathbf{P}\mathbf{x}$ (where $\mathbf{P} > \mathbf{0}$). $\mathbf{f}(\mathbf{x})$ is thus (Doyle et al., 1996):

$$\mathbf{f}(\mathbf{x}(t)) = \frac{1}{2}\left(\mathbf{g}(\mathbf{x}(t))\mathbf{g}^T(\mathbf{x}(t))\mathbf{P} - \mathbf{P}^{-1}\mathbf{Q}\right)\mathbf{x}(t) + \mathbf{P}^{-1}\boldsymbol{\gamma}(\mathbf{x}(t)) \tag{2.65}$$

where $\boldsymbol{\gamma}(\mathbf{x})$ is a function in $\mathbb{R}^n$ which satisfies $\boldsymbol{\gamma}(\mathbf{0}) = 0$. $\mathbf{g}(\mathbf{x})$ and $\boldsymbol{\gamma}(\mathbf{x})$ can be chosen to ensure the system dynamics are nonlinear.

The optimal control $\mathbf{u}^*(\mathbf{x})$ is then described by:

$$\mathbf{u}^*(\mathbf{x}) = -\frac{1}{2}V_{\mathbf{x}}^*(\mathbf{x})\mathbf{g}(\mathbf{x}) \tag{2.66}$$

This solves the infinite horizon case, but we are interested in the finite horizon case. To solve this the *principle of optimality* is used. Simply put, this states that if $\mathbf{u}^*(\mathbf{x}(t))$ is optimal for the infinite horizon problem, then $\mathbf{u}^*(\mathbf{x}(t))$ is also optimal over the interval $[0, T]$. Thus the optimal infinite horizon cost can be split into optimal subregions, with the sum of these subregions equal to the infinite horizon cost. Using this fact, a finite horizon cost function can be generated by performing the integral from time $t=0$ to arbitrary terminal time $t=T$, and then adding the optimal infinite horizon value function using the system state at $t=T$. Due to the fact that $V^*(\mathbf{x})$ represents the optimal infinite horizon cost corresponding to initial condition $\mathbf{x}$, the cost function becomes:

$$\min_{\mathbf{u}(\cdot)} V(\mathbf{u}) = V^*(\mathbf{x}(T)) + \int_0^T \left\{q(\mathbf{x}(t)) + \mathbf{u}(t)^T\mathbf{u}(t)\right\} dt \tag{2.67}$$

which is equivalent to Equation 2.60.

For this method to yield an optimal solution, $V^*(\mathbf{x})$ and $q(\mathbf{x})$ must be positive definite, and $\lim_{\mathbf{x}\to 0}\frac{q(\mathbf{x})}{V_{\mathbf{x}}^*(\mathbf{x})} = 0$.

With careful choice of $V^*(\mathbf{x})$, $q(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$, the resulting equations $\mathbf{u}^*(\mathbf{x})$ and $\mathbf{f}(\mathbf{x})$ can be designed to be neat and simple.

The case study from Section 2.3.1 was generated using this method, and is shown below as an example:

Specify the input functions, $V^*(x)$, $q(x)$ and $g(x)$:

$$V^*(x) = x^2 \tag{2.68}$$
$$q(x) = x^2 \tag{2.69}$$
$$g(x) = 5(1-x)e^x \tag{2.70}$$

$f(x)$ is calculated using Equation 2.64:

$$\begin{aligned} f(x) &= \frac{\frac{1}{4}(V_x^*(x)g(x))^2 - q(x)}{V_x^*(x)} \\ &= \frac{\frac{1}{4}\left[2x\left(5(1-x)e^x\right)\right]^2 - x^2}{2x} \\ &= \frac{100}{8}x(1-x)^2e^{2x} - \frac{1}{2}x \end{aligned}$$

The system dynamics from $f(x)$ and $g(x)$ are found using the scalar form of Equation 2.61:

$$\dot{x} = f(x) + g(x)u$$

The optimal control solution for this system is then determined using Equation 2.66:

$$\begin{aligned} u^* &= -\frac{1}{2}V_x^*g(x) \\ &= -5x(1-x)e^x \end{aligned}$$

This solution is for the infinite horizon optimal control problem. The objective function for the finite horizon problem is determined using the modifications shown in Equation 2.67:

$$\begin{aligned} \min_{u(\cdot)} V(u) &= V^*(x(T)) + \int_0^T \left\{q(x(t)) + u(t)^T u(t)\right\} dt \\ &= x^2(T) + \int_0^T \left\{x^2 + u^2\right\} dt \end{aligned}$$

The term $x^2(T)$ accounts for the optimal cost for the state evolution from $t = T$ to $t = \infty$. The choice of system initial condition $x(0) = x_0$ is arbitrary.

The case example in Section 2.3.2 was also generated using this method, although for the MIMO problem.

2.4.2 Converse *Constrained* Hamilton-Jacobi-Bellman method

The same form for the objective function and system dynamics as in Section 2.4.1 are used:

$$\min_{\mathbf{u}(\cdot)} V(\mathbf{u}) = \int_0^\infty \left\{ q(\mathbf{x}(t)) + \mathbf{u}^T(t)\mathbf{u}(t) \right\} dt \tag{2.71}$$

which is equivalent to

$$\min_{\mathbf{u}(\cdot)} V(\mathbf{u}) = V^*(\mathbf{x}(T)) + \int_0^T \left\{ q(\mathbf{x}(t)) + \mathbf{u}^T(t)\mathbf{u}(t) \right\} dt$$

where $V^*(\mathbf{x}(t))$ is the optimal infinite horizon value function (specified). Hence $V^*(\mathbf{x}(T))$ represents the optimal infinite horizon cost from the terminal state of the finite horizon performance objective.

This performance objective is subject to the system dynamics and initial conditions

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{g}(\mathbf{x}(t))\mathbf{u}(t) \tag{2.72}$$

$$\mathbf{x}(0) = \mathbf{x_0} \tag{2.73}$$

However, in this case a set of constraint functions is included in the formulation:

$$\mathbf{c}(\mathbf{x}, \mathbf{u}) \leq \mathbf{0} \tag{2.74}$$

Only examples using saturation constraints were considered:

$$\|\mathbf{u}\|_\infty \leq \boldsymbol{\alpha} \tag{2.75}$$

$V^*(\mathbf{x})$ and $q(\mathbf{x})$ are only admissible in the co-CHJB problem with constraints of the form

of Equation 2.75 if $q(\mathbf{x})$ can be factored into the product

$$q(\mathbf{x}) = V_{\mathbf{x}}^{*}(\mathbf{x})\mathbf{h}(\mathbf{x}) \tag{2.76}$$

$\mathbf{h}(\mathbf{x})$ must be continuous and satisfy $\mathbf{h}(\mathbf{0}) = \mathbf{0}$.

It is shown in Nevistic (1997) that $\mathbf{h}(\mathbf{x})$ can be found using:

$$\mathbf{h}(\mathbf{x}) = -\left(\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}^{*}(\mathbf{x}) + \frac{V_{\mathbf{x}}^{*T}(\mathbf{x})\mathbf{u}^{*T}(\mathbf{x})\mathbf{u}^{*}(\mathbf{x})}{\|V_{\mathbf{x}}^{*}(\mathbf{x})\|^2}\right) \tag{2.77}$$

where $\mathbf{u}^{*}(\mathbf{x}) = -\frac{1}{2}V_{\mathbf{x}}^{*}(\mathbf{x})\mathbf{g}(\mathbf{x})$

In summary, the generation of this form of problem was approached by choosing $V^{*}(\mathbf{x})$, $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$, and then calculating $\mathbf{h}(\mathbf{x})$. This was done in an iterative fashion to ensure all required conditions were met. $q(\mathbf{x})$ can be found using Equation 2.76. The optimal constrained input is then (Nevistic, 1997):

$$\mathbf{u}^{*}(\mathbf{x}) = sat_{\alpha}\left(-\frac{1}{2}\mathbf{g}(\mathbf{x}(t))^{T}V_{\mathbf{x}}^{*T}\right) \tag{2.78}$$

where the function $sat_{\alpha}(z)$ is defined as:

$$\boxed{sat_{\alpha} = \begin{cases} \alpha & z > \alpha \\ z & -\alpha \leq z \leq \alpha \\ -\alpha & z < -\alpha \end{cases}} \tag{2.79}$$

The case example in Section 2.3.3 was generated using this method, as follows:

Choosing

$$V^{*}(t) = x_1^2(t) - x_1 \tag{2.80}$$

$$\mathbf{f}(\mathbf{x}(t)) = \begin{bmatrix} x_2 \\ -x_1 \end{bmatrix} \tag{2.81}$$

$$\mathbf{g}(\mathbf{x}(t)) = \begin{bmatrix} -\beta_1 x_2^2 & x_2 - x_1 & -x_1 \\ 0 & -x_1 & \beta_2 x_2(1 - x_1^2) \end{bmatrix} \tag{2.82}$$

$\mathbf{h}(\mathbf{x})$ can then be found using Equation 2.77:

$$\mathbf{h}(\mathbf{x}) = -\left(\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}^{'}(\mathbf{x}) + \frac{\left(\frac{V^*(\mathbf{x})}{\partial \mathbf{x}}\right)^T \mathbf{u}^{'T}(\mathbf{x})\mathbf{u}^{'}(\mathbf{x})}{\| \left(\frac{V^*(\mathbf{x})}{\partial \mathbf{x}}\right) \|^2} \right)$$

Finally, $q(\mathbf{x})$ can be found by Equation 2.76:

$$q(\mathbf{x}) = V_{\mathbf{x}}^*(\mathbf{x})\mathbf{h}(\mathbf{x})$$

This results in an optimal control problem in the form of Equations 2.71 - 2.73 and 2.75. The optimal feedback control function is

$$\mathbf{u}^*(\mathbf{x}) = sat_{\alpha}\left(-\frac{1}{2}\mathbf{g}(\mathbf{x}(t))^T \left(\frac{V^*(\mathbf{x})}{\partial \mathbf{x}} \right)^T \right)$$

Chapter 3

Nonlinear Predictive Control

3.1 Review

Model Predictive Control (MPC) is an advanced control strategy in which the current control action is determined by the solution of an optimisation problem at each sampling instance. MPC is widely used in the petrochemical and similar industries due to its ability to handle multi-input, multi-output problems with hard constraints on the state and control. Plants can be operated closer to their limits, resulting in high efficiencies and large economic benefits. Although linear MPC is the primary implementation, nonlinear MPC is gaining acceptance for highly nonlinear applications. Improvements in methods of solving nonlinear optimal control problems, and increased computational power have led to this increasing acceptance.

There is some argument as to the origins of MPC. Amoco and IBM had algorithms with some elements of MPC as early as the 1960s (Morari, 1994), but most research started after a seminar paper was published by Cutler and Ramaker in 1980 (Cutler and Ramaker, 1980). At present many vendors offer MPC with identical base structure, but differing in the finer details. These fine details have proven to be critical in whether or not a particular algorithm is applicable.

Details about the solution of optimal control problems are given in Chapter 2, but the basic algorithm is as follows (see Figure 3.1):

Using a known set of past input (u) moves, and the corresponding output history (y) of a process, as well as knowledge of the process dynamics, the present state ($\hat{x}$) of the

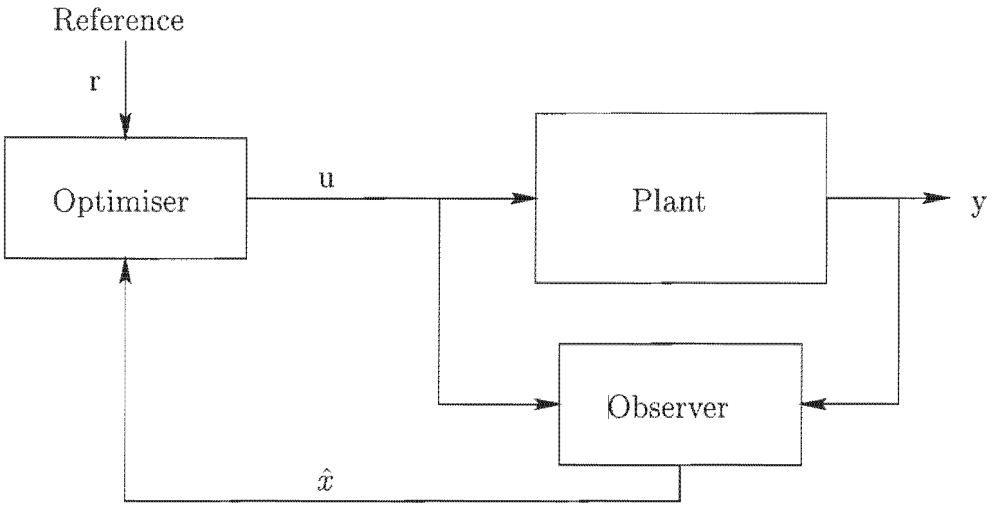

Figure 3.1: General structure of Model Predictive Control (MPC)

process can be estimated. This is the role of the observer which is a mathematical state approximator (e.g. Kalman filter).

With a process model and initial process state ($\hat{x}$), the future state and output of the process can be predicted for any set of input moves. The quality of the prediction is dependent on the accuracy of the model. By applying an optimiser to the model, a set of input moves can be found that will cause the predicted output to optimally follow some desired reference path. The optimisation is performed over a finite future time (the output horizon) using a certain number of input moves (input or control horizon). Depending on the type of optimisation, constraints can be included on the input and output variables. With a sufficiently accurate process model, implementation of this set of input moves can be expected to cause the behaviour of the real process to follow the desired reference path.

Due to expected imperfection in the process model, only the first input move is implemented, and then the input trajectory is recalculated based on the state of the process at the next sampling instance. Thus deviations in the model from the actual process response are corrected at every sampling instance. The horizon is also moved forward - referred to as moving or receding horizon MPC. A formulation of this sort is readily expanded to the multi-input, multi-output case.

The predictor models used in the algorithm can be either linear or nonlinear. Linear models

have the following form:

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)x(t) + D(t)u(t) \end{aligned} \tag{3.1}$$

Often A, B, C and D are independent of time (also know as time invariant or autonomous).

Nonlinear models are described by:

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), t) \\ y(t) &= h(x(t), u(t), t) \end{aligned} \tag{3.2}$$

f and h are usually time invariant, and assumed to be continuously differentiable. A subclass of nonlinear models are those where the system dynamics are linear or *affine* in the control. They have the following form:

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t) \tag{3.3}$$

Although this system is linear in the control, $f(x(t))$ and $g(x(t))$ are nonlinear functions of the state variable.

MPC optimization is subject to equality and inequality constraints. These can occur on either the input or the output. Frequently input constraints are hard constraints, whereas output constraints are allowed some flexibility.

Amongst the many advantages of MPC are that it is designed for constraints, the multi-input, multi-output implementation is straightforward, and it handles time delays. However, it does have some limitations.

One large area of concern, and thus ongoing research, is the stability and robustness of MPC. Determining stability theoretically is difficult due to constraints making the closed-loop system nonlinear, even if the plant and model are linear. The algorithm also does not have an explicit description in the form of a function, which is required for most stability analyses. Thus in general there are few stability guarantees, but results are emerging in this regard. Nonetheless, robust performance analysis remains generally unsolved.

Nonlinear MPC also suffers from the Euler-Lagrange solution of optimal control problems being local. The global optimal solution is, in most cases, unknown. As a result of this, the evaluation of the performance of MPC is difficult (Nevistic, 1997).

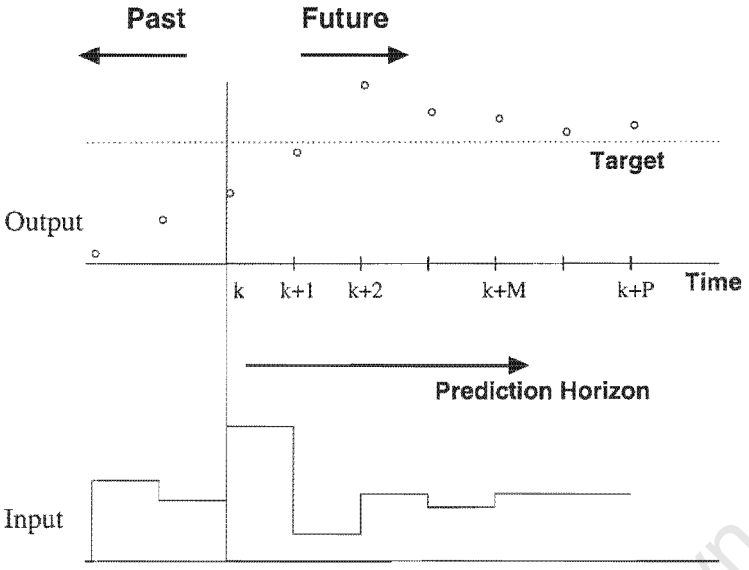

Figure 3.2: Graphical interpretation of model-predictive control

Linear MPC includes quite a large family of algorithms, including dynamic matrix control (DMC) and quadratic dynamic matrix control (QDMC).

DMC relies on a step response model, and operates under the assumptions of linearity and time invariance. It makes use of the fact that, under these assumptions, given any arbitrary change in input a system will respond according to a linear description of the system, which can be described by means of a matrix of step response coefficients. Unmodeled effects can be taken into account by means of additional variables.

Using this fact, given a desired output, a set of inputs can be determined to optimally satisfy the output (as explained above). In the case of DMC this is achieved using a least-squares solution of the equations. Due to the nature of this optimisation, it can be done off-line to generate a matrix of solution coefficients and thus all that is required at each sampling instance is a matrix multiplication.

However, often it is desired to strictly conform to input and output constraints. This is achieved by using quadratic programming methods to solve the optimisation problem. In this case, constraints can be formulated as linear inequalities, thus allowing tighter constraint control (Garcia and Morshedi, 1986). Unlike DMC, the quadratic program must be solved at each sampling instance, which imposes a computational constraint on the method. Nonetheless quadratic programs are rapidly solved, ensuring that this method can be practically implemented.

For systems with mild nonlinearity, linear controllers can be used that are merely detuned. However, in cases where the nonlinearity is strong, there is considerable advantage to using nonlinear controllers.

The main characteristic of *nonlinear MPC* is the solution of an optimal control problem with an underlying nonlinear dynamic model. This is non-trivial, and a number of methods have been considered to achieve this, either in avoiding the nonlinear optimal control problem (e.g. *linearized optimal control*), or solving it.

Linearized optimal control (Garcia, 1984) uses a linearised controller (DMC or QDMC) in which the linear model is updated as the process state changes. The updated linear model is used to obtain the step response coefficients, with the effect of past inputs on the response determined using a nonlinear model. Effectively this is a linear controller with a nonlinear model.

Determination of the optimal nonlinear control input requires the solution of a nonlinear optimal control problem as outlined in Chapter 2. A number of methods have been considered, but the most popular are the direct methods. Conventionally this results in a piecewise constant control trajectory, although this is only an approximation of the optimal solution.

3.1.1 Problem formulation and stability results

Allgöwer *et al* (1999) provide an excellent review on nonlinear predictive control from which much of the discussion in this section is drawn.

As described above, MPC centers around a model of the process. Historically this has been a time-domain or step/impulse response model which is simple to understand and use. However, state-space models are increasingly being used by researchers in the MPC field:

$$x_{k+1} = f(x_k, u_k) \tag{3.4}$$

$$y_k = h(x_k) \tag{3.5}$$

$$u_k \in \mathbb{U} \tag{3.6}$$

$$x_k \in \mathbb{X} \tag{3.7}$$

where:

x is the n-vector of states
y is the p-vector of outputs (measurable)
u is the m-vector of inputs (manipulated variables)
k is the discrete time sample number

In this form, the model describes a discrete NMPC formulation, in which moves are described by piecewise constant steps over the sample time. By using the state-space form, generalisation to multivariable systems is straightforward, closed-loop properties can be easily analysed, and it is amenable to on-line computation.

Finite Horizon NMPC

Consider the case of a system described by Equations 3.4 to 3.7, with the sets $\mathbb{U}$ and $\mathbb{X}$ given by:

$$\mathbb{U} := \{u_k \in \mathbb{R}^m \mid u_{min} \leq u_k \leq u_{max}\} \tag{3.8}$$
$$\mathbb{X} := \{x_k \in \mathbb{R}^n \mid x_{min} \leq x_k \leq x_{max}\} \tag{3.9}$$

To begin only the regulator problem is considered - that of bringing the process states from an initial condition x_0 to the fixed operating setpoint. In this case the open-loop NMPC optimal control problem, to be solved at each sampling instance, is given by:

Solve:

$$\min_{u_k^N} V(x_k, u_k^N) \tag{3.10}$$

subject to:

$$x_{l+1|k} = f(x_{l|k}, u_{l|k}) \tag{3.11}$$
$$x_{0|k} = x_k \tag{3.12}$$
$$u_{l|k} \in \mathbb{U}, \quad l \in [0, N-1] \tag{3.13}$$
$$x_{l|k} \in \mathbb{X}, \quad l \in [0, N] \tag{3.14}$$

with:

$$V(x_k, u_k^N) = \Phi(x_{N|k}) + \sum_{i=0}^{N-1} L(x_{i|k}, u_{i|k}) \tag{3.15}$$

The role of the double index (e.g. $x_{l|k}$) indicates an internal controller variable. The first subscript of the variable is the prediction time in the controller starting at $l = 0$, whereas the second subscript is the sampling time or time instance when the prediction was made. $u_k^N = \{u_{0|k}, u_{1|k}, \ldots, u_{N-1|k},\}$ where the upper index N is the control horizon length. It is assumed that the prediction and control horizon lengths are equal. At each sampling instance k, the first element of the solution sequence u_k^{N*} of the optimal control problem (Equations 3.10 to 3.15) is applied. Thus $u_k = u_{0|k}^*$ and the process can thus be regarded as having closed-loop NMPC control. Due to N having a finite value, this form of NMPC is known as *finite horizon NMPC*.

Useful assumptions (Allgöwer et al., 1999)

- A_1: $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^n$ is twice continuously differentiable on $\mathbb{X} \times \mathbb{U}$.
- A_2: $0 = f(0, 0)$. It is also assumed that the steady state is the desired operating point.
- A_3: $\mathbb{U} \subset \mathbb{R}^m$ is compact and convex, $\mathbb{X} \subseteq \mathbb{R}^n$ is closed and simply connected with (0,0) in the interior of $\mathbb{X} \times \mathbb{U}$.
- A_4: At each time instance k the full state x_k is available for measurement (state feedback).
- A_5: No disturbances (persistent or not) act on the system.
- A_6: There is no mismatch between the model used to predict the dynamic behaviour and the real plant.
- A_7: The stage cost L (Equation 3.15) is a quadratic function in x and u:
$$L(x_{i|k}, u_{i|k}) = x_{i|k}^T Q x_{i|k} + u_{i|k}^T R u_{i|k}$$
with Q, R positive definite.

Depending on the NMPC scheme, not all of the assumptions may be required. Many of these assumptions detract from the practical implementation of MPC (e.g. A_5 and A_6), but this will be dealt with in the section on robust MPC (to follow). In some cases, assumptions such as A_7 can be relaxed as they are only used to simplify some of the results.

Infinite Horizon NMPC

Using an infinite prediction horizon ($N = \infty$) with Assumptions A_4 to A_6 holding, Bellman's Principle of Optimality can be used to show that the open-loop input and state trajectories computed at time k as the solution of the NMPC optimisation problem are

equal to the closed-loop trajectories of the nonlinear system. As a result, closed-loop stability is guaranteed as any feasible predicted trajectory will return to the origin. Known as *infinite horizon NMPC*, this is the most intuitive way to achieve closed-loop stability.

To approximate infinite horizon NMPC, extremely large prediction horizons are used. In linear systems, the use of large prediction horizons is quite feasible and there are a number of efficient ways to solve the resulting huge quadratic program. However, in the case of nonlinear MPC the use of extremely large horizons is generally not possible, leaving infinite horizon NMPC as a conceptual method only. Finite horizons are thus the focus of NMPC research.

The use of a finite prediction horizon results in the actual closed-loop response differing from the open-loop predicted response. This will be the case even if there are no plant/model mismatches, and for short prediction horizons can result in significant differences. Two consequences of this fact are apparent. Firstly, repeatedly minimising the finite horizon performance objective does not result in the optimal solution of the infinite horizon performance objective. This needs to be considered when choosing the objective function for the finite horizon case. Secondly, due to the open-loop and closed-loop trajectories differing, open-loop stable trajectories do not guarantee stability in the closed-loop system. Thus the stability properties of finite horizon NMPC needs to be studied, and methods developed that guarantee closed-loop stability.

Ideally a controller design procedure that determines stabilising prediction and control horizons based on system model and objective function is desired. However, pursuit of this design procedure has proved fruitless and thus the development of methods that are independent of system model and performance objective are the current focus of research. The approach that has been adopted to guarantee closed-loop stable NMPC is the modification of the NMPC setup by the addition of suitable equality and inequality constraints. Known as stability constraints, they are not added as a result of physical restrictions or plant performance enhancements, but simply to guarantee closed-loop stability. Due to their stability guarantee being independent of the system model or performance function, NMPC strategies with these modified formulations are know as NMPC approaches with *guaranteed stability.*

The sections following will describe some of the available formulations.

NMPC with zero state terminal equality constraint

This is the most widely suggested scheme for guaranteed stability of NMPC. Stability is assured by including a terminal equality constraint forcing the states to the origin at the prediction horizon (N):

$$x_{N|k} = 0$$

In linear quadratic control this corresponds to an infinite initial condition of the backwards Ricatti difference equation, the solution of which results in a stabilising state feedback controller for linear controllable systems (Allgöwer et al., 1999).

The use of this constraint results in a solution equivalent to the infinite horizon solution. By forcing the states to the origin at the prediction horizon and setting the input thereafter to zero, the system will remain at the origin forever (barring disturbances). Thus the prediction horizon has been expanded to infinity. Feasible formulations for both discrete and continuous time NMPC strategies have been shown (Meadows et al., 1995; Mayne and Michalska, 1990).

This method is the most popular at present, as it has a clear theoretical framework and no off-line tuning or computation is required. However, the addition of a terminal equality constraint does increase the computational burden of the problem and raises issues of feasibility. In many cases the region of operation is severely restricted to ensure feasible solutions can be found.

For systems with value functions that are not continuous at the origin, it can be shown that if the horizon length of NMPC with zero state terminal equality constraint is increased to infinity, the resulting trajectory does not necessarily approach that of infinite horizon NMPC. As a result the choice of the closed-loop value function is not straightforward.

Practical application of this method must take these issues into consideration, and this method can only be recommended with some reservation.

Dual-mode NMPC

The term dual-mode refers to the use of two different controllers applied to separate regions of the state-space. The reasoning behind this method is to attempt to relax the requirement of an exact solution of the zero-state terminal equality constraint described above. From a

computational point of view it is impossible to exactly satisfy a zero-state constraint; this constraint is only approximately satisfied, forcing the states to lie in a small region around the origin and as a result the guarantee of stability is in general lost. To relax the equality stability constraint while ensuring the stability guarantee is not lost, dual-mode NMPC was introduced (Michalska and Mayne, 1993).

Dual-mode NMPC makes use of two different controllers that are applied in different regions of the state-space depending on whether the present state lies within some terminal region (containing the origin) or not. If the state lies outside the region, NMPC with a variable horizon length is applied. However, if the state lies inside the terminal region, then a linear state feedback controller is used ($u_k = Kx_k$). Closed-loop control is implemented by switching between the two controllers:

$$u_k = \begin{cases} u^*_{0|k}, & \text{if } x_k \notin \Omega \\ Kx_k, & \text{if } x_k \in \Omega \end{cases} \tag{3.16}$$

The state feedback K and the terminal region Ω are determined off-line. Ω is determined such that it is a positive invariant region of attraction for the nonlinear system controlled by the linear state feedback law. The input and state constraints are both satisfied with this linear controller in Ω.

For NMPC, the optimisation problem includes the determination of the horizon length N_k. This is taken as an additional minimiser, resulting in the following cost function:

$$\min_{u_k^N, N_k} V(x_k, u_k^N, N_k)$$

with the additional terminal state constraint:

$$x_{N_k|k} \in \Omega_B \tag{3.17}$$

Ω_B is the boundary of the terminal state region, and Equation 3.17 requires that the terminal states of the optimal control problem solved at sampling instance k lie on the boundary of the terminal region Ω.

It can be shown, under fairly weak conditions, that the nonlinear system can reach the terminal region in a finite time under NMPC. Closed-loop stability is assured by use of the

stabilising local linear controller thereafter (Allgöwer et al., 1999).

This approach has a number of advantages: computationally it is more attractive than methods requiring a terminal state equality constraint as the inequality constraint can be handled more efficiently; a larger region of attraction is realised, due to feasibility problems playing a smaller role.

However, the implementation of this control scheme is more involved as a result of the two modes of operation and the off-line determination of the state feedback law and the terminal state region. Another drawback is that the predicted open-loop and the actual closed-loop responses will in general be different.

Contractive NMPC

Contractive NMPC makes use of the following stability constraint:

$$\| x_{N_k|k} \|^2 \leq \alpha^2 \| x_k \|^2, \quad \alpha \in (0,1) \tag{3.18}$$

This constraint is known as the *contraction stability constraint* and it forces the magnitude of the state vector to contract by a specified factor with each new input calculation. Unlike previously described NMPC formulations, the *entire* input function $u_k^{N_k*}$over the time interval $[k, k+N_k-1]$ is applied to the nonlinear system, and the following optimisation is only solved at time instance $k+N_k$.

de Oliveira (1996) suggests an additional stability constraint

$$\max_{j=0,\ldots,N_k} \| x_{j|k} \|^2 \leq \beta^2 \| x_k \|^2, \quad \beta \in [1,\infty) \tag{3.19}$$

and that the horizon length $N_k \leq N_{max}$ is used as an additional minimiser. Closed-loop asymptotic stability is proven by showing that $\| x_{k+j} \|^2 \leq \alpha^2 \| x_k \|^2 \rightarrow 0$ as $j \in \mathbb{N} \rightarrow \infty$ holds (Allgöwer et al., 1999).

Feasibility for this method must be assumed for all time instances for which the optimisation is performed. This is a strong assumption due to the addition of two constraints and a modified cost function (including final time as with dual-mode NMPC). As a result this approach is not very attractive for practical application due to the feasibility of solution for each time instance not being guaranteed. For the previous methods discussed, feasibility at $k=0$ implies feasibility at all later times. Large values of α might obtain the required

feasibility, but will result in a slow contraction of the state vector. Furthermore, despite the free specification of the desired control performance by the objective function, the actual achieved performance will be influenced by α and β.

This approach can be generalised by the use of a contraction constraint on some positive definite function $J(x_{k|N_k}) \leq \alpha^2 J(x_k), \quad \alpha \in (0,1)$. This enforced contraction guarantees the monotonic decay of J, which allows the use of a Lyapunov based proof of closed-loop stability if J is used as a Lyapunov function (Mayne, 1996).

Quasi-infinite horizon NMPC

The quasi-infinite NMPC scheme makes use of an inequality stability constraint and a terminal penalty term in addition to the standard setup, respectively:

$$x_{N|k} \in \Omega \tag{3.20}$$

$$\Phi\left(x_{N|k}\right) = x_{N|k}^T P x_{N|k} \tag{3.21}$$

The terminal penalty function is not a performance specification, but aims to approximate an infinite horizon cost functional. This allows closed-loop stability to be achieved with a finite horizon optimisation. The matrix P is chosen off-line according to a procedure (to follow) to ensure the infinite horizon cost is adequately approximated.

The infinite horizon cost is defined by

$$V^{\infty}(x_k, u_k^{\infty}) = \sum_{i=0}^{\infty} L\left(x_{i|k}, u_{i|k}\right) \tag{3.22}$$

where u_k^{∞} is a control sequence of infinite length. This cost can be split up into two parts, a finite horizon cost and an infinite horizon cost:

$$\min_{u_k^{\infty}} V^{\infty}(x_k, u_k^{\infty}) = \min_{u_k^{\infty}} \left\{ \sum_{i=0}^{N-1} L\left(x_{i|k}, u_{i|k}\right) + \sum_{i=N}^{\infty} L\left(x_{i|k}, u_{i|k}\right) \right\} \tag{3.23}$$

$\Phi\left(x_{N|k}\right)$ aims to approximate the second term on the right hand side of the above equation. Due to this being a nonlinear system, this is not possible unless the state of the system over the time interval $[k+N, \infty)$ is constrained to lie within some neighborhood of the origin. With such a constraint an upper bound can be determined for the cost functional. Constraint Equation 3.20 performs this role with Ω being constructed to allow the local

linear state feedback law $u_{i|k} = Kx_{i|k}$ to asymptotically stabilise the system in Ω, and to render Ω positively invariant for the closed-loop. Due to Ω being positively invariant, only the terminal (point) inequality constraint (Equation 3.20) is necessary to ensure that the trajectories u_k remain in Ω for all time beyond $k+N$. We now have:

$$\min_{u_k^\infty} V^\infty(x_k, u_k^\infty) \leq \min_{u_k^N} \left\{ \sum_{i=0}^{N-1} L\left(x_{i|k}, u_{i|k}\right) + \sum_{i=N}^{\infty} L\left(x_{i|k}, Kx_{i|k}\right) \right\} \tag{3.24}$$

Terminal region Ω and the terminal penalty matrix P can be chosen using the same methods used in dual-mode NMPC. With the correct choice of P it can be shown (Chen and Allgöwer, 1998) that:

$$\sum_{i=N}^{\infty} L\left(x_{i|k}, u_{i|k}\right) \leq x_{N|k}^T P x_{N|k} \tag{3.25}$$

Substituting this result into Equation 3.24 results in:

$$\min_{u_k^\infty} V^\infty(x_k, u_k^\infty) \leq \min_{u_k^N} V(x_k, u_k^N) \tag{3.26}$$

with V being the finite horizon cost functional described in the original problem formulation (Equation 3.15). This implies that the optimal solution of the finite horizon cost bounds that of the infinite horizon cost. Using this property asymptotic stability for the closed-loop system can be guaranteed independent of the control performance.

This method was inspired by the dual-mode NMPC technique, but simplifies the formulation by removing the need to switch controllers based on the region of operation. Modifications of this method include attempting to approximate the terminal penalty function by solving the infinite horizon cost from N using the local linear feedback control law:

$$\Phi\left(x_{N|k}\right) = \sum_{i=N}^{\infty} L\left(x_{i|k}, Kx_{i|k}\right) \tag{3.27}$$

This is not practically implementable, but it has been suggested to approximate the infinite sum with one of sufficient length. However, this raises a number of questions including how long *"sufficient"* is and how severe the additional computational burden will be.

The quasi-infinite method, like dual-mode NMPC, has a number of computational advantages. It is also able to overcome some of the main drawbacks of the dual-mode method. It

is a simpler implementation as the controller switching is avoided, and the approximation of infinite horizon cost ensures that the finite and infinite horizon costs are at least similar. This allows controller performance to be adjusted by the stage cost L. As with dual-mode NMPC, the optimal solution of Equations 3.10 to 3.15 is not required to guarantee stability; feasibility also implies stability.

Robust NMPC

The NMPC schemes described above require a number of assumptions to hold, including that the system model and actual plant match perfectly. This also implies that no unknown disturbances affect the plant. For practical implementation these requirements are unrealistic, and thus the development of an NMPC strategy that is robust in the face of practical imperfections is extremely important. The main obstacle to be overcome is the ability to handle the differences between predicted open-loop and actual closed-loop responses.

A number of results have been published for linear predictive controllers, but the robustness analysis for NMPC is still considered unsolved although some preliminary results are available. Zero-state terminal equality constraint, dual-mode, contractive and quasi-infinite horizon NMPC inherently possess robustness properties or can be made robust by simple changes. This usually involves conservative choices for the stabilising constraints (e.g. "smaller" terminal inequality constraint regions in dual-mode NMPC). However, stability guarantees cannot be given based on the degree of uncertainty and no design techniques exist to determine, for example, the size of a constrained region as a function of the expected level of plant/model mismatch. Thus this remains an ongoing area of research.

3.1.2 Solution strategies

The NMPC schemes described above require the on-line solution of a nonlinear optimal control problem at each sampling instance to determine the manipulated inputs. As a result of the problem being nonlinear, the optimisation problem is generally nonconvex. This presents a major challenge and is the subject of considerable research. Practical implementation of MPC requires the use of theory and algorithms from the fields of numerical optimisation and numerical linear algebra. Improvements in the field of optimisation theory have important impacts on MPC research, particularly in the field of nonlinear model

predictive control. The discussion to follow considers some of the more widely used solution strategies, as well as mentioning alternative NMPC formulations with improved computational characteristics.

Nonlinear programming

The typical NMPC formulation is based on a discrete-time state-space model of the nonlinear system. The discrete-time model may be obtained by nonlinear input-output system identification, or by fundamental system modeling. If the fundamental modeling approach is adopted, a continuous-time model is usually produced from which a discrete-time model can be explicitly produced. However, the continuous model can often be used without being explicitly discretised, as the NLP solver implicitly forms a discrete model from it (Henson, 1998).

The solution of this NLP problem presents a number of challenges. These include the efficient and reliable solution of the nonlinear system, potentially with a large number of decision variables. Powerful computers are required if the on-line (real-time) solution of this problem is to be at all viable. The nonlinear system dynamics may well form a nonconvex constraint of the NLP problem, with the result that few methods are available that can guarantee a global minimum. Furthermore, feasibility cannot be guaranteed in the presence of nonconvex constraints. Some of the widely used algorithms to solve the NLP problem in the face of these challenges are presented below.

Successive linearisation of model equations

The simplest method to solve the NLP is to approximate the nonlinear system equations using Jacobian linearisation around the nominal operating region. The resulting model can be discretised and, if the objective function is quadratic, solved using a simple quadratic programming method. Unfortunately this approach does not compensate for process nonlinearities.

The linearisation approach can be extended by linearising the system at each sampling instant around the current operating point. This reduces the NMPC problem to a linear MPC problem at each time step. However, this will only compensate partially for process nonlinearities.

A number of variations on the successive linearisation method have been proposed by various researchers. Most of the methods use a linearised model for the NLP problem, and then use the nonlinear dynamics to predict the system behaviour from previous input moves. This leads to methods that update the linear model along the computed trajectory, or several times over the sampling period.

This must still be regarded as an approximation of the system, and the full benefit of using nonlinear system dynamics will not be realised.

Sequential model solution and optimisation

The main aim of using nonlinear system equations is to capture process nonlinearities. Thus improved performance is inevitable if the nonlinear system equations are used in the optimisation. However, NLP solvers do not solve ODEs (the system equations) and to overcome this problem, a separate ODE solver can be used for the system equations, and an NLP solver is used to perform the optimisation. Solved sequentially, iterations of this procedure are repeated until the desired level of accuracy is achieved.

An advantage of this method is that the only decision variables for the NLP solver are the manipulated inputs. However it is difficult to incorporate state (or output) constraints in the formulation, and the problem is known to be unreliable for larger problems (Meadows and Rawlings, 1997).

This method has been used as a basis by many researchers for solving NMPC. The model solution can be simplified by using collocation on finite elements (although other methods can be used), resulting in a finite number of nonlinear algebraic equations.

Simultaneous model solution and optimisation

The discretised model described in the above method can be incorporated into the nonlinear program, resulting in a simultaneous solution of the model and optimisation problem. The decision variables become the system inputs over each of the finite elements, and the state variables at each collocation point. This method increases the number of decision variables as the sampling period is shortened or the prediction horizon lengthened, or if the number of collocation points is increased.

This method handles large NLP problems well, and is suitable for state/output constraints (Meadows and Rawlings, 1997).

Alternative NMPC formulations

Another area to be considered when attempting to improve computational efficiency when solving the NMPC problem is the actual problem formulation. Alternative NMPC problem formulations with better computational properties have been proposed during the last few years. Zheng (1997) proposes a scheme to reduce the number of decision variables in the NLP problem. This is achieved by only allowing the *current* inputs to be manipulable, and all future inputs along the control horizon are decided by linear MPC controllers linearised according to their region of operation in the state-space.

Using an alternative approach in the formulation of the nonlinear model itself can, in some cases, be more computationally efficient when used in NLP problem solvers. Examples include second-order Volterra models with linear auto-regressive terms, and polynomial ARMAX models (Henson, 1998).

3.2 A Hybrid Nonlinear MPC Strategy

3.2.1 Introduction

Conventional MPC (both linear and nonlinear) results in a discretised, piecewise-constant control trajectory (see Figure 3.2). Each control input consists of a step held constant over the sampling period. This is as a result of the method used to solve the underlying optimal control problem, usually a discretised method in itself. This hold action is not as a result of physical limitations of the control equipment, due to the fact that most physical devices operate on an analogue basis and are perfectly capable of handling continuously varying signals. It is also reasonable to expect that a continuously varying control input would lead to improved performance due to the continuous input being a more accurate representation of the optimal control profile. The hybrid NMPC strategy proposed here makes use of a continuously varying control trajectory, computed by the indirect solution of the optimal control problem by orthogonal collocation. Indirect solution of optimal control problems has been shown in Chapter 2 to be highly accurate, as the control and/or state profile is not approximated by a constant or linearly varying grid of points.

Unlike conventional MPC, the optimal control problem is solved by determining a smooth control trajectory over the control horizon. The problem is solved from the state of the

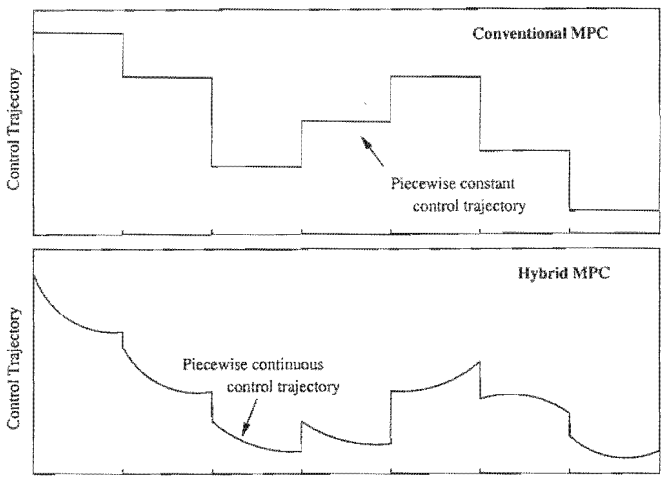

Figure 3.3: Comparison of piecewise constant and piecewise continuous control profiles

system at the sampling instance over the prediction horizon. The control and prediction horizon are of equal length. Instead of implementing the first optimal control move (as with piecewise constant controls), the first *sequence* of the control trajectory is implemented. This sequence is the same length as the sampling length, resulting in a continuously varying control input throughout the entire sampling instance. The optimal control profile is then re-calculated at the next sampling instance using the system state at that time.

The hybrid strategy is implemented using the quasi-infinite horizon NMPC strategy of Chen and Algöwer (1998), with the on-line optimal control problem solved using the indirect collocation method described in Chapter 2. This scheme guarantees asymptotic closed-loop stability of both stable and unstable systems. Their method is a quasi-infinite NMPC scheme with a finite horizon cost including a terminal cost. These cost functions are optimised subject to the system dynamics, control constraints and an additional terminal inequality constraint (a stability constraint). The reasons for the terminal cost and terminal inequality stability constraint are to approximate an infinite horizon cost, with the final system states in a prescribed region at the terminal time. The terminal penalty cost represents an upper bound on the infinite horizon cost within the region specified by the terminal constraint if the system is controlled by a linear feedback controller. This controller is never implemented, as the nonlinear controller is also used within the region, but it is a guarantee of stability and is used in the off-line determination of the terminal penalty matrix and the terminal region. This scheme thus has a *quasi-infinite prediction horizon* but with a finite control horizon.

If the Jacobian linearisation of the nonlinear system to be controlled is stabilisable, the unique positive-definite, symmetric solution of an appropriate Lyapunov equation can serve as the penalty matrix of the terminal cost. The terminal region includes the origin, and the area it includes can be determined off-line. If the open-loop optimal control problem is feasible, then closed-loop asymptotic stability is guaranteed at $t = 0$ (Chen and Allgöwer, 1998).

3.2.2 Problem formulation

This formulation follows the NMPC scheme proposed by Chen and Allgöwer (1998).

The system dynamics and initial condition are described by

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \tag{3.28}$$

$$\mathbf{x}(0) = \mathbf{x}_0 \tag{3.29}$$

with state $\mathbf{x}(t) \in \mathbb{R}^n$ and input $\mathbf{u} \in \mathbb{R}^m$. The system is subject to input constraints:

$$\mathbf{u}(t) \in \mathbb{U} \tag{3.30}$$

The following assumptions are used (Chen and Allgöwer, 1998):

1. $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ is twice continuously differentiable and $\mathbf{f}(\mathbf{0}, \mathbf{0}) = \mathbf{0}$. Thus the system is at steady-state at $\mathbf{0} \in \mathbb{R}^n$ with $\mathbf{u} = \mathbf{0}$.
2. $\mathbb{U} \subset \mathbb{R}^m$ is compact, convex and $\mathbf{0} \in \mathbb{R}^m$ is contained within $\mathbb{U}$.
3. The dynamic system (Equations 3.28 and 3.29) has a unique solution for any initial condition $\mathbf{x}_0 \in \mathbb{R}^n$ and any piecewise continuous and right-hand continuous $\mathbf{u}(\cdot) : [0, \infty) \rightarrow \mathbb{U}$.

It is assumed that the states are all measurable, allowing for state feedback. The notation used by Chen and Allgöwer is used throughout this section. For any vector $\mathbf{x} \in \mathbb{R}^n$, $\| \mathbf{x} \|$ represents the 2-norm, with the weighted norm $\| \mathbf{x} \|_P$ defined by $\| \mathbf{x} \|_P^2 := \mathbf{x}^T P \mathbf{x}$. P is an arbitrary Hermitian, positive-definite matrix. To distinguish between the real time in which the controller is operating, and the simulated "internal" time used in the solution of the optimal control problem, the variables used in the solution of the optimal control

problem are denoted by $(\bar{x}, \bar{u})$. This also highlights the fact that the predicted and actual values will differ.

The optimal control problem to be solved at time t with initial condition $\mathbf{x}(t)$ is formulated as follows:

$$\min_{\bar{\mathbf{u}}(\cdot)} V(\mathbf{x}(t), \bar{\mathbf{u}}(\cdot)) \tag{3.31}$$

with

$$V(\mathbf{x}(t), \bar{\mathbf{u}}(\cdot)) = \| \bar{\mathbf{x}}(t+T_p; \mathbf{x}(t), t) \|_P^2 + \int_t^{t+T_p} \left(\| \bar{\mathbf{x}}(\tau; \mathbf{x}(t), t) \|_Q^2 + \| \bar{\mathbf{u}}(\tau) \|_R^2 \right) d\tau \tag{3.32}$$

subject to

$$\dot{\bar{\mathbf{x}}} = \mathbf{f}(\bar{\mathbf{x}}, \bar{\mathbf{u}}), \quad \bar{\mathbf{x}}(t; \mathbf{x}(t), t) = \mathbf{x}(t) \tag{3.33}$$

$$\bar{\mathbf{u}}(\tau) \in \mathbb{U}, \quad \tau \in [t, t+T_p] \tag{3.34}$$

$$\bar{\mathbf{x}}(t+T_p; \mathbf{x}(t), t) \in \Omega \tag{3.35}$$

where $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{m \times m}$ denote positive-definite, symmetric weighting matrices; T_p is the finite prediction horizon; $\bar{\mathbf{x}}(\cdot; \mathbf{x}(t), t)$ is the state trajectory described by Equation 3.33, as a result of the control trajectory $\bar{\mathbf{u}}(\cdot) : [t, t+T_p] \to \mathbb{U}$. The initial condition is included in Equation 3.33. The controller uses the model equation to predict the system evolution from its initial state at real time t.

The objective function includes a standard finite horizon cost to specify the control performance, and a penalty cost on the states at the terminal point on the horizon. Note that for simplification the control and prediction horizons are taken to be the same length. The terminal inequality constraint ensures that the system states are contained in or on the surface of the terminal region Ω at the end of the prediction horizon. Ω is chosen to ensure that it is invariant for the nonlinear system controlled by local linear state feedback. The weighting P ensures that the finite horizon cost bounds the infinite horizon cost starting

from Ω and controlled by the local linear state feedback controller:

$$\begin{aligned} \| \bar{\mathbf{x}}(t+T_p;\mathbf{x}(t),t) \|_P^2 &\geq \int_{t+T_p}^{\infty} \left(\| \bar{\mathbf{x}}(\tau;\mathbf{x}(t),t) \|_Q^2 + \| \bar{\mathbf{u}}(\tau) \|_R^2 \right) d\tau \\ \bar{\mathbf{u}} &= K\bar{\mathbf{x}}, \quad \forall \bar{\mathbf{x}}(t+T_p;\mathbf{x}(t),t) \in \Omega \end{aligned} \tag{3.36}$$

The terminal penalty matrix P and terminal region Ω are determined off-line, ensuring that the invariance property of Ω holds, and that the input constraints are satisfied in Ω.

This scheme can be thought of as having two controller steps, the finite horizon nonlinear optimal control stage, and the locally linearised feedback control stage. The nonlinear controller scheme drives the system to the terminal region in which the linearised controller operates. This local controller would steer the system to the origin. However, the control profile for the linear feedback controller will never be implemented - in fact it is never even calculated. The closed-loop control is calculated by the nonlinear controller irrespective of the system states. The linearised controller is merely used to determine (off-line) the terminal penalty matrix and the terminal region Ω.

This NMPC scheme is proposed by Chen and Algöwer (1998), and the method of solution of optimal control problems described in Chapter 2 is highly amenable to its implementation, due to its ability to handle control and terminal state inequality constraints.

3.2.3 Implementation

An important aspect of this approach is the determination of P and Ω. The approach from Chen and Algöwer (1998) is adopted, providing a systematic method to determine the terminal penalty matrix and terminal constraint region. This is done off-line, and the notational use of the bar ($\bar{\mathbf{x}}$) is discontinued.

The use of Jacobian linearisation results in a system of the form:

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} \tag{3.37}$$

with A and B found from

$$A = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{0}, \mathbf{0})$$
$$B = \frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\mathbf{0}, \mathbf{0})$$

If this system is stabilisable then a linear state feedback control law of the form $\mathbf{u} = K\mathbf{x}$ can be found such that $A_K = A + BK$ is asymptotically stable. P is found by the solution of the Lyapunov function:

$$(A_K + \kappa I)^T P + P(A_K + \kappa I) = -Q^* \tag{3.38}$$

where $Q^* = Q + K^T RK \in \mathbb{R}^{n \times n}$ and is positive definite and symmetrical.

The solution of Equation 3.38 results in a unique, positive-definite and symmetrical P. $\kappa \in [0, \infty)$ satisfies:

$$\kappa < -\lambda_{max}(A_K) \tag{3.39}$$

The operator λ_{max} demands an explanation: for any Hermitian matrix A, $\lambda_{max}(A)$ and $\lambda_{min}(A)$ denote the largest and smallest real part of the eigenvalues of A respectively. $\| A \|$ denotes the induced 2-norm of A.

Ω is found by determination of the constant α, which specifies a region surrounding the origin Ω_α of the form:

$$\Omega_\alpha := \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x}^T P \mathbf{x} \leq \alpha\} \tag{3.40}$$

The region Ω_α must satisfy the following conditions:

1. $K\mathbf{x} \in \mathbb{U}$ for all of $\mathbf{x} \in \Omega_\alpha$. Thus the linear feedback controller satisfies the input constraints.

2. Ω_α is invariant for the linear system controlled by the local linear feedback controller

3. For any $\mathbf{x}_1 \in \Omega_\alpha$ the infinite horizon cost

$$V^{\infty}(\mathbf{x}_1, \mathbf{u}) = \int_{t_1}^{\infty} \left(\| \mathbf{x}(t) \|_Q^2 + \| \mathbf{u}(t) \|_R^2 \right) dt \tag{3.41}$$

subject to the nonlinear system equations but controlled by the linearised feedback controller is bounded from above by $\| \mathbf{x}_1(t) \|_P^2$, i.e.:

$$V^{\infty}(\mathbf{x}_1, \mathbf{u}) \leq \mathbf{x}_1^T P \mathbf{x}_1 \tag{3.42}$$

Ω_α is found through a four-step procedure, as outlined in Chen and Algöwer (1998).

1. K is found, based on the Jacobian linearisation of the nonlinear system. This is the locally stabilising state-feedback gain array.

2. $\kappa \in [0, \infty)$ is found which satisfies Equation 3.39:

$$\kappa < -\lambda_{max}(A_K)$$

The Lyapunov equation (3.38) is solved to find P. This will be a positive-definite symmetric matrix.

3. A search is then performed to find the largest α_1 that describes the region Ω_{α_1} (Equation 3.40) which satisfies the constraint $K\mathbf{x} \in \mathbb{U}$ for all $\mathbf{x} \in \Omega_{\alpha_1}$.

4. Finally, the largest possible α must be found in the set $(0, \alpha_1]$ which ensures

$$L_\phi \leq \frac{\kappa \lambda_{min}(P)}{\| P \|} \tag{3.43}$$

where $L_\phi := \sup\{ \| \phi(\mathbf{x}) \| / \| \mathbf{x} \| \mid \mathbf{x} \in \Omega_\alpha, \mathbf{x} \neq \mathbf{0} \}$ and $\phi(\mathbf{x}) := \mathbf{f}(\mathbf{x}, K\mathbf{x}) - A_K\mathbf{x}$. L_ϕ is found by a maximisation over Ω_α.

The inequality in the fourth step is difficult to satisfy resulting in an extremely small Ω_α region for some systems (Chen and Allgöwer, 1998). A less conservative, yet valid, terminal region can be found by a different approach. Steps 1 - 3 are followed as above, but Step 4

is replaced by a simple optimisation search. The optimisation problem

$$\max_{\mathbf{x}}\{\mathbf{x}^T P\phi(\mathbf{x}) - \kappa\mathbf{x}^T P\mathbf{x} \mid \mathbf{x}^T P\mathbf{x} \leq \alpha\} \tag{3.44}$$

is solved repeatedly with α being reduced from α_1 until the maximisation becomes non-positive. The region found in this way can then be used as a terminal region (Chen and Allgöwer, 1998).

3.2.4 Conclusion

The method presented in Chapter 2 of solving optimal control problems indirectly using orthogonal collocation is highly amenable to use in the NMPC strategy proposed by Chen and Algöwer (1998). This strategy guarantees closed-loop asymptotic stability under weak assumptions, with the main requirement being feasibility of the optimal control problem solution. This strategy has computational and formulation advantages; it does not impose high computational demands and is straightforward to implement. It does require the use of a terminal inequality constraint, but this is efficiently handled in the optimal control solution method proposed.

The off-line determination of the terminal penalty and terminal region do not impose additional computational demands on the real-time implementation, and thus are not considered a burden.

There are other NMPC strategies which are amenable to the use of the proposed optimal control problem solution, and thus the method proposed above should not be considered as the only option.

3.3 Application Example

The application of NMPC to a continuous stirred tank reactor (CSTR) is used to compare conventional to hybrid NMPC. Figure 3.4 is a schematic illustration of an isothermal CSTR in which the reaction

$$\begin{aligned} A &\xrightarrow{k_1} B \\ B &\xrightarrow{k_2} C \\ 2A &\xrightarrow{k_3} D \end{aligned}$$

occurs. The dynamics and control of this system have been considered in van de Vusse (1964); Sistu and Bequette (1995); and Meadows and Rawlings (1997), amongst others. The results presented here closely follow Meadows and Rawlings, (1997).

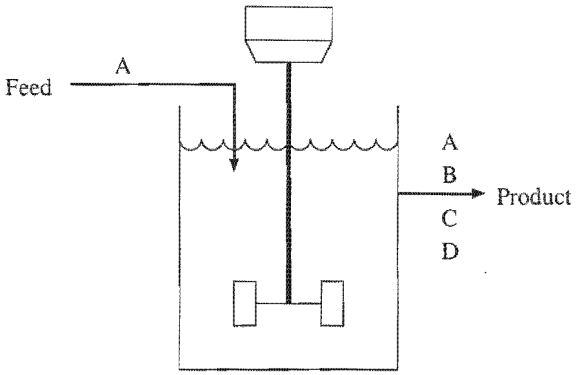

Figure 3.4: Diagram of isothermal CSTR

The system can be described by the following nonlinear two-state system:

$$\dot{C}_A = -k_1 C_A - k_3 C_A^2 + (C_{A,0} - C_A)q \tag{3.45}$$

$$\dot{C}_B = k_1 C_A - k_2 C_B - C_B q \tag{3.46}$$

Here C_A and C_B represent the concentrations of A and B in the tank respectively, and $C_{A,0}$ is the concentration of A in the feed stream. The manipulated variable is q, the dilution rate. Nominal values of $\{k_1, k_2, k_3, C_{A,0}\} = \{50hr^{-1}, 100hr^{-1}, 10l/mol.hr, 10mol/l\}$ were used for the rate constants and feed concentration. The control objective is to maintain the outlet concentration of the desired product B at a setpoint $\overline{C_B} = 1mol/l$. Solutions for this setpoint may be attained at steady-state conditions $\{\overline{C_A}, \overline{C_B}, \overline{q}\} = \{2.5, 1.0, 25\}$ or $\{\overline{C_A}, \overline{C_B}, \overline{q}\} = \{6.6667, 1.0, 233.33\}$.

Due to lower dilution rates achieving higher conversion (Sistu and Bequette, 1995) the first solution is used for steady-state, around which deviation variables are formed:

$$x_1 = C_A - \overline{C_A}$$

$$x_2 = C_B - \overline{C_B}$$

$$u = q - \overline{q}$$

The resulting state-space model in deviation form is:

$$\dot{x}_1 = -k_1(x_1 + \overline{C_A}) - k_3(x_1 + \overline{C_A})^2 + (x_F - (x_1 + \overline{C_A}))(u + \overline{q}) \quad (3.47)$$

$$\dot{x}_2 = k_1(x_1 + \overline{C_A}) - k_2(x_2 + \overline{C_B}) - (x_2 + \overline{C_B})(u + \overline{q}) \quad (3.48)$$

The control variable has the constraint $-25 \leq u \leq 25$ $(0 \leq q \leq 50)$ and the state variables are physically constrained such that the concentrations remain positive:

$$x \in \mathbb{X}\{x \in \mathbb{R}^2 \mid x_1 \geq -2.5, x_2 \geq -1.0\} \quad (3.49)$$

Consideration of the velocity vectors along the boundaries of $\mathbb{X}$ reveals that if the control constraint is satisfied and $\mathbf{x}(0) \in \mathbb{X}$, then the resulting state trajectory never leaves $\mathbb{X}$. This is due to the rates approaching zero as the concentrations tend to zero. As a result, if the initial concentrations are positive, and the input constraint is not violated (always positive), the concentrations will always be positive or zero. Thus the state constraint need not be included in the hybrid control law.

The following parameters are used:

Parameter	Value
$\mathbf{Q}$	$1000I^2$
R	1
Δ[s]	7.2
T [s]	3Δ

where $\mathbf{Q}$ and R are cost function weightings, Δ is the sample time, and T is the prediction horizon length.

To determine the terminal region, the Jacobian linearisation of the system results in the following parameters:

$$A = \begin{bmatrix} -125 & 0 \\ 50 & -125 \end{bmatrix} \quad (3.50)$$

$$B = \begin{bmatrix} 7.5 \\ -1 \end{bmatrix} \quad (3.51)$$

which describes the system $\dot{\mathbf{x}} = A\mathbf{x} + Bu$.

The linear state feedback matrix K was calculated to be

$$K = \begin{bmatrix} -19.726 & -1.2545 \end{bmatrix} \tag{3.52}$$

Using these values, P is found

$$P = \begin{bmatrix} 11.865 & 16.838 \\ 16.838 & 41.282 \end{bmatrix} \tag{3.53}$$

To determine P, κ was taken to be 90% of $-\lambda_{max}(A_K)$.

α_1 was found which ensured $Ku \in \mathbb{U}$ for all $\mathbf{x} \in \Omega_{\alpha_1}$:

$$\alpha_1 = 8.455 \tag{3.54}$$

Applying step 4 described in Section 3.2.3 resulted in a conservative estimate for α:

$$\alpha = 3.787 \tag{3.55}$$

However, solving the optimisation problem of Equation 3.44 showed that α_1 can be used to describe the terminal region, with the optimisation returning a non-positive solution at $\alpha = \alpha_1$.

Thus the following terminal state constraint must be satisfied for the problem:

$$\mathbf{x}^T P \mathbf{x} \leq 8.455 \tag{3.56}$$

The initial condition used for the system is

$$\mathbf{x}(0) = \begin{bmatrix} 0.5 \\ 0.1 \end{bmatrix}$$

This initial condition is already within the terminal region described by Equation 3.56, resulting in a stable controller.

The hybrid NMPC algorithm was compared to a conventional NMPC implementation on

the basis of the approximate infinite horizon cost function

$$V_\infty(u) = \int_0^{T_\infty} \left\{ \mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t) + u^T(t)Ru(t) \right\} dt \tag{3.57}$$

Figure 3.5 shows the performance of the hybrid model predictive controller.

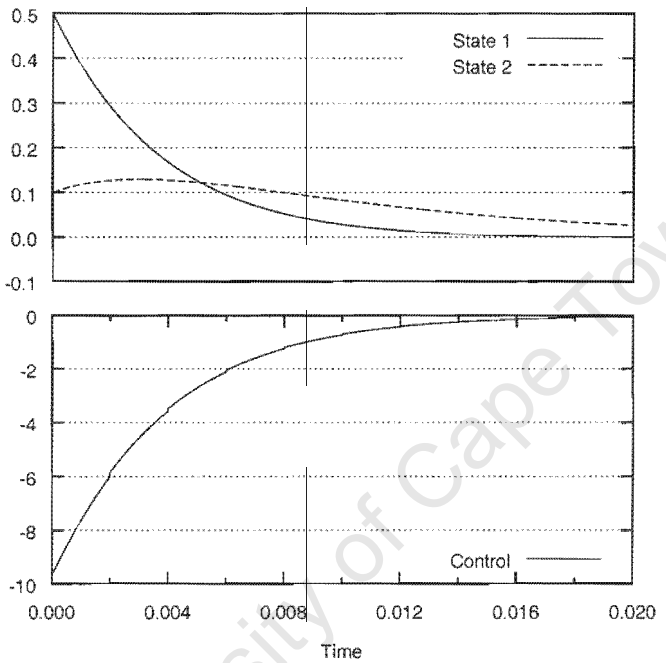

Figure 3.5: NMPC of a CSTR - Example of NMPC using piecewise continuous control

T_∞ was chosen to be sufficiently large that the deviation form of the state and control variables were close to zero. Design specific parameters such as a terminal penalty and constraint were removed to provide a more objective comparison. The contribution of state and control variables to the above integral are shown separately in Figure 3.6, which shows the results of comparisons using the following parameter values:

Parameter	Case 1	Case 2	Case 3	Case 4
Q	$\begin{bmatrix} 1000 & 0 \\ 0 & 1000 \end{bmatrix}$	$\begin{bmatrix} 500 & 0 \\ 0 & 1000 \end{bmatrix}$	$\begin{bmatrix} 1000 & 0 \\ 0 & 500 \end{bmatrix}$	$\begin{bmatrix} 1000 & 0 \\ 0 & 1000 \end{bmatrix}$
R	[1]	[1]	[1]	[5]

Figure 3.6 shows the hybrid MPC law produces a reduced cost, especially in the control contribution.

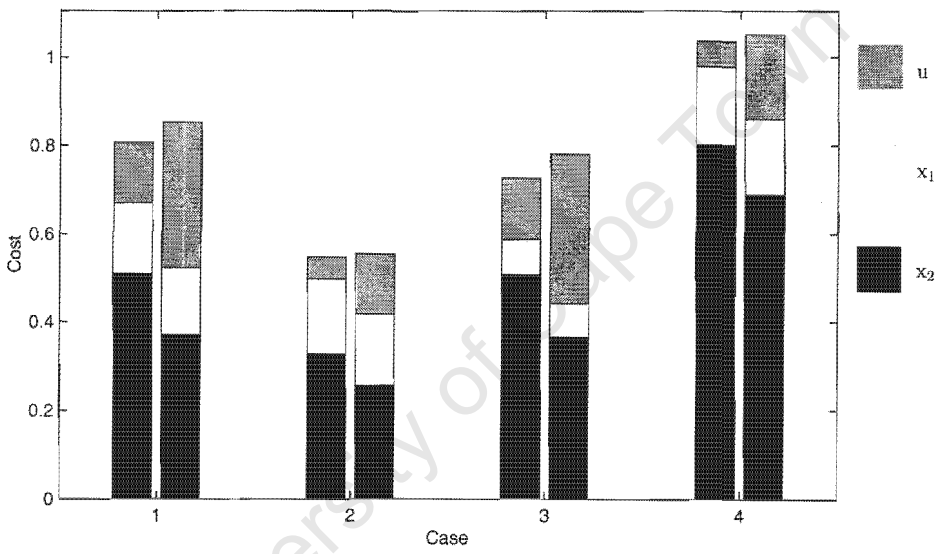

Figure 3.6: NMPC of a CSTR - Comparison of cost contributions of x_1, x_2 and u to the value function

Chapter 4

Flotation Modeling

4.1 Review

This review aims to cover some of the more advanced aspects of flotation modeling, and how it can be used to produce a useful dynamic model with application to control in the minerals processing industry.

A fair amount of progress has been made by the scientific community into the finding of a suitable model that describes the process of flotation. However this must still be considered work in progress, and the models available are constantly improving. The vast majority of models at present only consider flotation at steady-state, barely considering process dynamics. This work attempts to use the available models to produce a dynamic model and makes assumptions about the system dynamics where necessary. These assumptions will be given.

To successfully model flotation, the process has been broken down into what are believed to be its constituent processes. This enables the modeling of the highly complex process to occur in manageable sections. At present the process is modeled as occurring by three mechanisms, namely *true flotation*, *entrainment* in concentrate water, and *entrapment* of particles within other particles. The recovery of water is also modeled as a separate process.

Most research interest has been on developing and understanding the kinetics of true flotation, although models for entrainment and water recovery are available. In the development of a complete flotation model, the ability to accurately predict the behaviour of these constituent subprocesses is vital for a representative overall model.

4.1.1 Basics of Flotation

Flotation is a separation process used extensively in minerals processing, although it has also been used successfully in paper recycling and cleansing of oil-contaminated soils. As a process to separate valuable minerals from non-valuable, it has found application in base-metal, coal, potash and platinum refining operations, to name but a few.

The underlying mechanism of flotation relies on hydrophobic particles in a water-based slurry attaching to air bubbles as they pass through it. When the bubbles reach the surface of the slurry, instead of bursting they are chemically induced to form a froth which can be easily removed. This froth will be rich in the hydrophobic mineral. Thus if the desired mineral is hydrophobic, it will attach to bubbles as they pass through the slurry, and be found in high concentration in the froth on the surface of the slurry. Reverse flotation is a similar process, only in this case the undesirable particles report to the froth, leaving the desired minerals in the slurry. Paper recycling uses reverse flotation to remove the graphite pigment particles from paper.

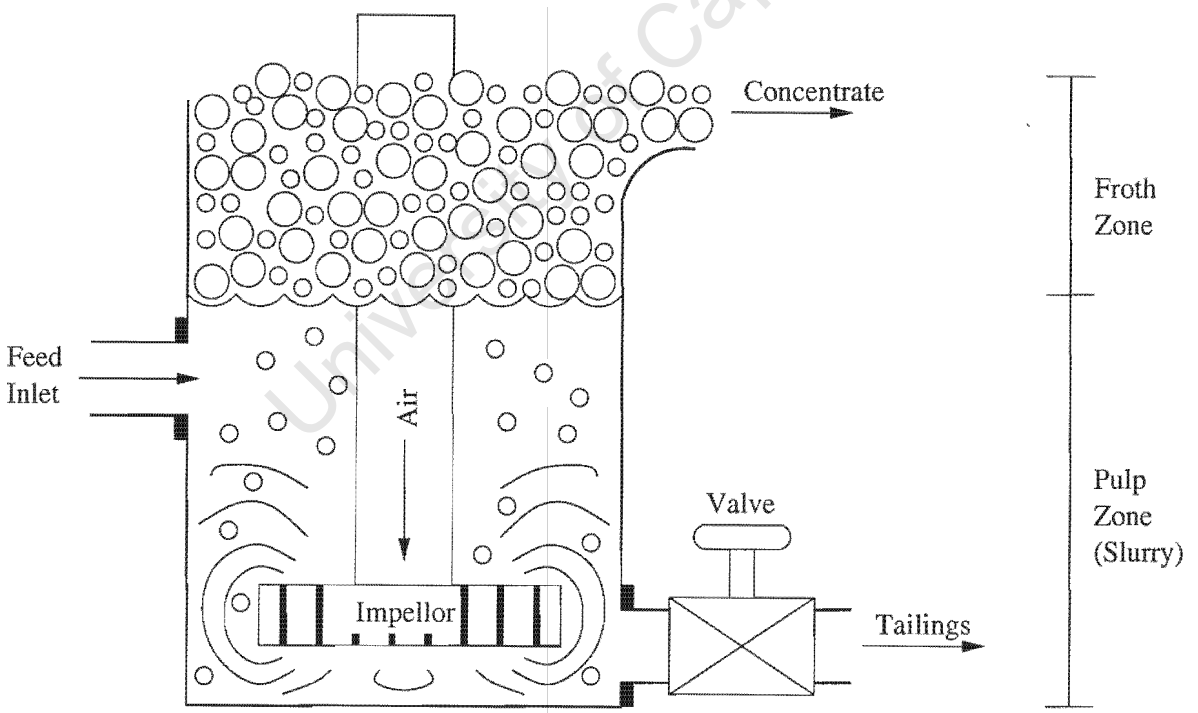

Figure 4.1: Schematic diagram of a simple flotation cell

Chemicals play a significant role in the flotation process, as they are used both to form a

stable froth and to modify surface hydrophobicity of minerals in the slurry. Certain chemicals, known as collectors, are able to selectively bond to the surface of specific minerals, modifying its surface characteristics to make it hydrophobic. Another class of chemicals - depressants - achieve the reverse effect by rendering specific mineral surfaces hydrophilic and thus preventing their attachment to bubbles ensuring they are not floated.

True flotation describes the recovery of minerals to the froth by attachment (due to hydrophobicity) of the mineral particle to a bubble. This attachment requires a large amount of energy, which is supplied by an impeller in the slurry. The impeller also ensures good mixing and even-distribution of bubbles throughout the flotation cell. Minerals exhibit a range of floatability resulting in different rates of flotation. Using chemical additives correctly can ensure that the desired minerals have considerably higher rates of flotation than undesirable minerals, resulting in a highly enriched froth.

Entrainment is a major mechanism operating in parallel with true flotation that causes minerals to report to the froth zone. It is almost completely non-selective, discriminating according to particle size (not mineral type). Minerals are recovered by this mechanism due to their being trapped in the thin water layer surrounding the bubble, and in interstitial pockets of water between bubbles. Particles may also be entrained in the wake of ascending bubbles. The primary factor affecting the rate of entrainment is particle size - smaller particles are more easily entrained than larger particles. As bubbles move through the layer of froth on the surface (froth zone), the pockets and layers of water containing entrained minerals drains back into the slurry lowering the amount of entrained mineral. Thus the deeper the froth on the surface of the slurry (and hence the greater the froth residence time), the lower the effect of entrainment. Typically particles smaller than 50 μm are the most affected by this mechanism.

Entrapment occurs as a result of different minerals occurring in the same particle. Thus if a small piece of gangue is attached to a valuable (and floatable) piece of mineral, it will be carried to the concentrate along with the valuable mineral. This results in a lowering of the grade of the concentrate. Similarly, valuable minerals may be attached to larger gangue particles which are not floated. This results in a loss of the valuable mineral which lowers recovery. Entrapment can be reduced by effective operation of the milling circuit to ensure adequate liberation of the valuable minerals.

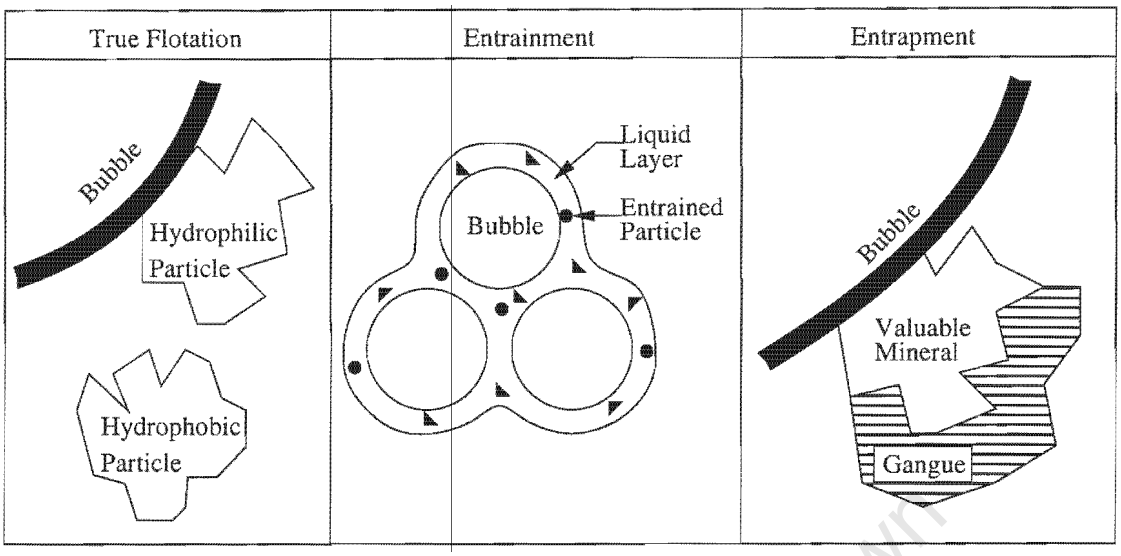

Figure 4.2: Flotation sub-processes: true flotation, entrainment; entrapment

4.1.2 Dynamic Modeling

An early dynamic model of a flotation circuit was proposed by Manlapig (1977), with the dynamic analysis of a chalcopyrite circuit . A dynamic mass balance was performed over a flotation cell, assuming the cell volume remained constant (perfect level control). Using the constant cell volume assumption, the flow to the tailings could be determined from the feed and concentrate flowrates.

The mineral body was divided into a number of mineral classes, based on primary flotation mechanism and rate of flotation. First-order rate equations were used to model the recovery of minerals to the concentrate, using the mass of minerals in the flotation cell as the driving force for the mass transfer. The rate constants were dependent on the type of mineral they described. Valuable (floating) minerals used a rate constant taking into account the mass transfer between the pulp and froth phases. Gangue minerals on the other hand were primarily recovered by entrainment, and so the rate constant was calculated using a classification function and the rate of water recovery.

By describing the mass transfer in a single cell, banks of flotation cells could be built using the tailings from one cell as the feed for the next. A full flotation circuit could be built up from these smaller building blocks.

Another dynamic model of flotation was developed by Smith (1984). This circuit followed

the development of Manlapig, using similar modeling techniques and assumptions. Once again perfect level control was assumed. It was found that flotation rate constants varied throughout the circuit, and this effect was attributed to bubble loading. Thus a model was developed which modified the flotation rate constant to take into account the changes in bubble loading across the circuit.

Both of these dynamic models provide a good starting point for the development of a dynamic flotation model. To extend and improve on these models, the most accurate representation of the flotation subprocesses is required. Thus models describing true flotation, entrainment and water recovery are investigated, with models describing the dynamics of the subprocesses being of particular interest.

4.1.3 Modeling of True Flotation

Most flotation models in the literature that describe true flotation on a macro scale assume it to be a first-order rate process (Harris, 1998). Thus the rate of removal of substance from the cell is proportional to the concentration of that substance in the cell:

$$-\frac{dC}{dt} = kC \tag{4.1}$$

It has been found that the agitation in conventional flotation machines is sufficient for the mixing to be considered ideal, and the cells can be modeled as having perfect mixing. The flotation rate constant, k, is modeled as a function of the characteristics of both the physical flotation unit and the stream being processed. By separating the modeling of k into these two categories, *and considering them to be independent of each other*, the problem is simplified considerably.

(Gorain et al., 1997) in (Harris, 1998) made an important step forward when they showed that, on an industrial flotation plant, the flotation rate constant is linearly dependent on a parameter called the bubble surface area flux (S_b). Effectively this is the rate of bubble surface area generation. Thus

$$k \propto S_b \tag{4.2}$$

The bubble surface area flux can be determined from the superficial gas velocity (J_g) and

the Sauter mean bubble diameter (d_b):

$$S_b = \frac{6J_g}{d_b} \tag{4.3}$$

This relationship is useful as both J_g and d_b can be physically measured. However, the bubble size (d_b) is also a function of air rate, and a number of other physical attributes of the cell. Unless these relationships are determined, Equation 4.3 is of limited use in a predictive model. Fortunately, Gorain (1997) found the following empirical relationship for S_b:

$$S_{b,k} = aN_s^b \left(\frac{Q_{\text{air}}}{A}\right)^c A_s^d P_{80}^e \tag{4.4}$$

where:

$S_{b,k}$	Bubble surface area flux of cell k	
$a-e$	Empirical parameters	
N_s	Impeller peripheral speed	$\left[\frac{L}{t}\right]$
Q_{air}	Air flowrate into flotation cell	$\left[\frac{L^3}{t}\right]$
A	Cell cross sectional area	$[L^2]$
A_s	Impeller aspect ratio (diameter/height)	$\left[\frac{L}{L}\right]$
P_{80}	80% passing feed size	$[L]$

Using Equation 4.4, S_b can be determined as a function of air flowrate alone (assuming the physical parameters of the cell remain constant).

It has been shown (Gorain et al., 1998a; Gorain et al., 1998b; Mathe et al., 1998) that k must be corrected for the recovery in the froth phase (R_f). The mineral floatability (P_i) must also be taken into account. This results in the flotation rate constant being a function of three components:

$$k_i = P_i S_b R_f \tag{4.5}$$

It is clear that although S_b can be determined from direct measurements, the determination of P_i and R_f is nontrivial. Therefore a lot of research has gone into finding ways to determine these parameters.

The mineral floatability P_i can be regarded as the basis for flotation. It is rooted in a bubble's ability to form a stable attachment to the particle surface, and it is not possible

to measure this directly. One approach to solving this problem has been to fit a kinetic equation curve to data obtained from batch flotation tests. Data can also be obtained from pilot or full-scale plants. There are a number of kinetic models that have been proposed, which consider the overall process in terms of different flotation classes. The flotation rate of the process is thus divided into a number of classes, each with their own rate constant (k_i). Table 4.1 shows some of the more common class definitions.

Table 4.1: Commonly used rate parameters (Harris, 1998)

Model	Type	Rate Parameters
Garcia Zuniga	discrete	k
Kellsall	discrete	k_{fast} and k_{slow}
Thomlinson and Fleming	discrete	k by size class
Rectangular (Klimpel)	distribution	k as a floatability distribution parameter (usually applied by size class)
Imaizumi and Inoue	discrete	k by floatability class
Zaidenberg	discrete	k by size, liberation and floatability class

Mineral floatability is a function of a wide-ranging and complex array of factors including physical mineral characteristics and the environment of the mineral (physical and chemical condition of the slurry). Particle size, liberation, degree of oxidation, slime coating and reagent coverage are but a few examples of factors that affect mineral floatability. Thus the more simple models (e.g. Garcia Zuniga, Kendall) are unable to describe the process of flotation with any physical significance. Even if the model appears to achieve a good fit, the parameters extracted have no real physical significance and thus are not useful for predictive modeling or simulation. The models incorporating real, physical parameters in the expression for k_i are likely to be the most useful in formulating relationships with physical significance. The models by Imaizumi and Inoue, and Zaidenberg are in this class of model.

The model proposed by Zaidenberg describes the floatability distribution of the mineral by size class. This model is of interest because it forms a link between comminution and flotation, and thus these two processes could be modeled and optimized simultaneously.

The model by Imaizumi and Inoue expresses floatability as a distribution of floatability classes. This model shows promise due to the importance of floatability on the rate of flotation.

There are two main approaches to determining the actual kinetic response of each floatability class. The first (employing scale-up factors) is not well founded in theory, whereas the second (the FRT approach) has a more theoretical foundation, but is far more complex. The two approaches are often combined.

The scale-up factor approach is not based on physical attributes, but simply uses a factor to scale from batch tests to real plant data. Parameters are fitted using least-squares minimisation. The method is not predictive, and cannot be used to test different configurations unless the changes are extremely small. However, it is useful for characterizing a given circuit.

The second method, known as the froth retention time (FRT) method, determines the kinetic response of each floatability class based on the characterization of performance of the froth phase. This performance is based on the froth retention time, defined as:

$$\tau_{fc,k} = \frac{V_{f_k}}{f_{c_k}} \tag{4.6}$$

where V_{f_k} is the volume of the froth in cell k, and f_{c_k} is the volumetric flow of concentrate from cell k. $\tau_{fc,k}$ represents the froth retention time in unit k.

This is analogous to residence time in a reactor, and provides a means of scaling up and down based on froth volume. It is important to note that it is not the actual residence time of the froth phase, due to air forming a large proportion of the froth volume. Although the FRT does not seem to form a well-defined physical attribute, it has been found to be a useful parameter in flotation modeling (Harris, 1998). It is also useful from the perspective that it is an easily observed factor for those operating flotation plants.

Experimentation has shown that froth recovery (R_f) can be expressed as a function of $\tau_{fc,k}$:

$$R_{f_k} = (1 - \alpha)e^{-\beta.\tau_{fc,k}} + \alpha \tag{4.7}$$

where

R_{f_k}	The froth recovery in unit k
$\tau_{fc,k}$	The froth retention time in unit k
α	A froth parameter related to the amount of non-draining mineral
β	A froth stability parameter

However, Equation 4.7 can be simplified (Franzidis and Manlapig, 1999) to become:

$$R_{f_k} = e^{-\beta.\tau_{fc,k}} \tag{4.8}$$

Using Equations 4.4, 4.8 and 4.5 the flotation rate constant can be determined, and this can then be used to determine the rate of removal of each species from each flotation cell by true flotation. Each class of mineral will have its own flotation rate constant k_i.

Returning to Equation 4.1, C can be considered analogous to the amount of a specific mineral class in the flotation cell. Describing the total mass of mineral class i in a flotation cell as X_i, and reapplying Equation 4.1 results in:

$$C_{tf,i} = k_i X_i \tag{4.9}$$

k_i is the flotation rate constant for a specific mineral floatability class and $C_{tf,i}$ is the contribution of true flotation to the concentrate mass flow of floatability class i. In steady-state modeling, this equation is not used in this form. It is solved for the case of a CSTR and used in a recovery model based on the feed rate of minerals to the cell and the flotation rate constants. However, this equation is a kinetic model and as such can be used without modification to include true flotation in the model of the dynamic behaviour of the cell.

4.1.4 Modeling of Entrainment

It is well known that there exists a strong relationship between the recovery of particles by entrainment, and water recovery. Above a minimum water recovery this relationship has been found to be linear (Kirjavainen, 1992; Savassi et al., 1998). Taking advantage of this linear relationship, the degree of entrainment is defined as the ratio of the recovery of solids to the recovery of water (Franzidis and Manlapig, 1999).

$$E_i = \frac{\text{recovery of entrained particles of the i}^{th}\text{ size interval to the concentrate}}{\text{recovery of water to the concentrate}}$$

Using this relationship, the recovery to the concentrate of size class i can be determined if both the degree of entrainment for that size class, and the water recovery is known:

$$R_i = E_i R_w$$

Entrainment is a function of many variables including mineral specific-gravity, particle size, residence-time of air in the froth, and the froth structure. This is an extremely complex relationship and is difficult to model on a fundamental level. However, entrainment has been found to be a strong function of particle size, and a semi-empirical model of the degree of entrainment is available to take advantage of this behaviour (Savassi et al., 1998):

$$\begin{aligned} E_i &= \frac{2}{e^{\left(2.292\left(\frac{x_i}{\xi}\right)^{\theta}\right)} + e^{\left(-2.292\left(\frac{x_i}{\xi}\right)^{\theta}\right)}} \\ \text{with :} & \\ \theta &= 1 + \frac{ln(\delta)}{e^{\left(\frac{x_i}{\xi}\right)}} \end{aligned} \tag{4.10}$$

In Equation 4.10, x_i is the particle size [μm], and ξ and δ are entrainment and drainage parameters respectively. ξ is defined as being the particle size (in microns) for which E_i is 20%. This equation is a type of partition curve, similar to those used in hydrocyclone models. Typically it has a shape as shown in Figure 4.3.

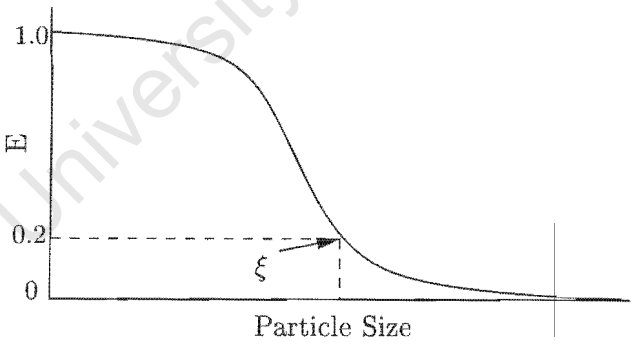

Figure 4.3: Typical entrainment partition curve

It is important to note that this is a steady-state model, based on the feed to the cell and not the mass of minerals within the cell.

Combining true flotation and entrainment

Certain mineral groups (e.g. gangue) are known to be recovered to the concentrate by both true flotation and entrainment. This is caused by the mineral having a slow but non-negligible rate of flotation. To model this phenomenon, another parameter is introduced, ϕ, which represents the fraction of mineral in the feed that floats by true flotation. The recovery of these minerals is thus modeled using:

$$R_k^0 = (1 - \phi) \sum_{j=1}^{d} (M_j R_{ENT,j}) + \phi R_{TF} \tag{4.11}$$

where:

R_k^0	The overall recovery in unit k
ϕ	The fraction of mineral in the feed floating by true flotation
M_j	The mass fraction of size class j
$R_{ENT,j}$	Recovery by entrainment of size class j
R_{TF}	Recovery by true flotation

4.1.5 Modeling of Water Recovery

The models given above (Sections 4.1.3 and 4.1.4) determine the solids recovery, but cannot be used to determine the amount of water recovered to the concentrate. However, the degree of entrainment is dependent on the water recovery, thus an accurate, independent water recovery model is required.

As with entrainment, available water recovery models are entirely empirical. This is a serious drawback, as it forms a vital part of the model. The empirical model is unlikely to give accurate performance at conditions far from the operating conditions over which it was determined.

Water recovery is often modeled assuming the recovery of water is proportional to the solids recovery (Harris, 1998):

$$Q_w = aC_s^b \tag{4.12}$$

where

Q_w The volumetric flowrate of water to the concentrate
C_s The mass flowrate of solids to the concentrate
a, b Empirical parameters to be derived for the circuit

This model is not useful as it depends on the flowrate of solids to the concentrate, which is in turn dependent on the water recovery. A far more useful model is presented in Harris (2000) which determines water recovery based on cell operating conditions. It has the same form as the models of true flotation, but with specific empirical water-recovery parameters including the floatability of water (P_w) and parameters for the froth recovery of water:

$$C_w = P_w S_b R_{f,w} X_w \tag{4.13}$$

where:

C_w	Mass flowrate of water to the concentrate stream	$[\frac{m}{t}]$
X_w	Mass of water in the cell	$[m]$
P_w	Floatability of water	
S_b	Bubble surface area flux of cell (same as flotation)	
$R_{f,w}$	Froth recovery of water	

The froth recovery of water was determined using

$$R_{f,w} = \Omega e^{(\sigma V_f - \chi \tau_{\text{air}})} \tag{4.14}$$

where Ω, σ and χ are empirical parameters (Harris, 2000). V_f is the volume of the froth, and τ_{air} is found using:

$$\tau_{\text{air}} = \frac{\varepsilon_g V_f}{Q_{\text{air}}} \tag{4.15}$$

ε_g is the gas holdup in the froth zone, and is generally assumed to be 1. This model is completely independent of the mass flow of solids, which allows the water recovery to be calculated for use in the entrainment calculation.

4.2 Development of a Dynamic Flotation Model

A major aspect of this project is the dynamic modeling of a flotation system. A dynamic model describes the behaviour of a process as a function of time, based on the state of

the system and inputs or disturbances to the system. With knowledge of the time-based evolution of the system, inputs can be chosen that will not only satisfy conditions at steady-state, but will also ensure that the system reaches steady-state rapidly and with stability.

Most of the available flotation models are not dynamic, and are based on the feed to the system as opposed to the state of the system. However, the fundamental models underlying the steady-state models are often useful for dynamic modeling. For cases where the steady-state model is not based on a dynamic mechanism, assumptions must be made regarding its dynamics.

The basic unit of flotation was taken to be the flotation cell, as it is a well mixed unit with clearly defined input, state and disturbance variables. A mass balance including accumulation was done across the cell for each mineral class to show how the mass of that class within the cell changed over time. This required knowledge of the feed to the cell, as well as the calculation of the flows exiting the cell based on the state of the cell and any input variables.

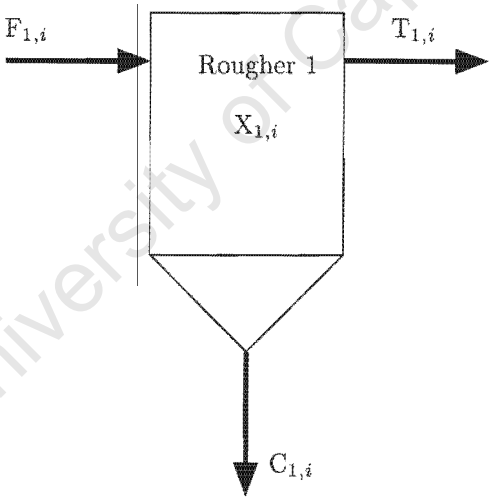

Figure 4.4: Flow description for flotation cell mass balance

Model Equations

The mass balance was done across the cell, resulting in the rate of change in the cell being described by:

$$\frac{dX_i}{dt} = F_i - C_i - T_i \tag{4.16}$$

where:

X_i	Mass of species i in cell	$[m]$
F_i	Mass flow of feed species i into cell	$\left[\frac{m}{t}\right]$
C_i	Mass flow of concentrate species i out of cell	$\left[\frac{m}{t}\right]$
T_i	Mass flow of tailings species i out of cell	$\left[\frac{m}{t}\right]$

This equation shows that the rate of change of mass in each mineral class in the cell is a function of the rate at which it is fed into the cell, the rate of removal from the cell to the concentrate, and the rate of removal to the tailings stream.

F_i term: This is simply the mass feed rate into the cell of each mineral class. The feed can originate from the comminution circuit, or be the tailings from an upstream cell.

T_i term: This is the tailings mass flow term. Constant volume was not assumed, so the manipulation of valve opening will result in a change in tank level. Linear valve dynamics were assumed, with the driving force being cell slurry-head:

$$Q_T = \rho_{cell}.g.K_v.vo.H \tag{4.17}$$

$$T_i = Q_T \frac{X_i}{V} \tag{4.18}$$

where:

Q_T	Total volumetric tailings flowrate	$\left[\frac{L^3}{t}\right]$
T_i	Mass flow of tailings species i out of cell	$\left[\frac{m}{t}\right]$
ρ_{cell}	Overall density of pulp in cell	$\left[\frac{m}{L^3}\right]$
g	Gravitational acceleration constant	$\left[\frac{L}{t^2}\right]$
K_v	Valve discharge coefficient	
vo	Valve opening percentage	$[\%]$
H	Pulp level in cell	$[L]$
V	Tank volume	$[L^3]$

ρ_{cell} is a function of the state of the cell, calculated using:

$$\rho_{cell} = \frac{\sum_{i=1}^{n+1} X_i}{V} \tag{4.19}$$

X_i is the mass of each species in the cell, and n is the number of solid species. The $n+1^{th}$ species is water. The volume of liquid in the cell can be calculated using the mass and density of each species in the cell, including water:

$$V = \sum_{i=1}^{n+1} \frac{X_i}{\rho_i} \tag{4.20}$$

Height is easily determined from the volume, using the cell cross-sectional area A:

$$H = \frac{V}{A} \tag{4.21}$$

C_i term: This is the most complex term in the mass balance equation. Two modes of mass removal are considered, namely true flotation and entrainment (entrapment is ignored). However, a very important aspect of this calculation is the water recovery rate. This is performed first, as the result affects both true flotation and entrainment. Equation 4.13 is used in the given form, as it is a kinetic model describing the rate of change of the mass of water within the cell:

$$C_w = P_w S_b R_{f,w} X_w$$

The calculation of τ_{air} and subsequently $R_{f,w}$ was done using Equation 4.15 and 4.14:

$$\tau_{\text{air}} = \frac{\varepsilon_g V_f}{Q_{\text{air}}}$$

$$R_{f,w} = \Omega e^{(\sigma V_f - \chi \tau_{\text{air}})}$$

The froth volume V_f is determined as being the froth height multiplied by the cell's cross-sectional area. The froth height is the difference in height between the slurry level in the cell and the launder lip. Q_{air} is the volumetric flowrate of air into the flotation cell. The bubble surface area flux is calculated using Equation 4.4.

True flotation:

The first-order rate equation form of true flotation (Equation 4.9) was used:

$$C_{tf,i} = k_i X_i$$

The flotation rate constant, k_i, is determined as shown using Equation 4.5:

$$k_i = P_i S_b R_f$$

P_i is the floatability of the i^{th} fraction of the mineral contained in the cell, and is found by experimentation. Gorain's bubble surface area flux model (Equation 4.4) is used to determine S_b as a function of the air sparge-rate:

$$S_{b,k} = a N_s^b \left(\frac{Q_{\text{air}}}{A} \right)^c A_s^d P_{80}^e$$

The simplified model for froth recovery (Equation 4.8) posed by Franzidis and Manlapig (1999) is used for R_f:

$$R_{f_k} = e^{-\beta . \tau_{fc,k}}$$

The froth retention time, τ_{fc}, is a complex function of air flowrate due to its dependence on the volumetric flow to the concentrate. An implicit relationship results from this dependence, which is resolved by approximating the concentrate flowrate using the water recovery and an estimated concentrate specific-gravity.

Lynch (1981), in a discussion on the use of aeration rate as a manipulated variable, states that the flotation process generally responds extremely rapidly to changes in the aeration rate. Taking this into account, and to simplify the model, it was assumed that the dynamics within the froth are negligible, thus allowing the use of the instantaneous froth recovery instead of a dynamic froth recovery model. Thus changes directly affecting the froth (e.g.

air sparge-rate) are modeled as happening instantly. Although in reality the response will not be instantaneous, it will be considerably faster than the dynamic response of the slurry within the cell.

Entrainment:

Entrainment is the primary mechanism by which non-floating gangue reports to the concentrate. The model for entrainment uses a type of partition curve to determine the mineral entrainability, based on the particle size of a particular species. The recovery of each size fraction is:

$$R_{ent,j} = E_j R_w \tag{4.22}$$

where E_j is the entrainability of size class j. Thus if mineral i is split into d size fractions, the recovery by entrainment for that mineral is:

$$R^o_{ent,i} = \sum_{j=1}^{d} m_{i,j} \left(E_{i,j} R_w\right) \tag{4.23}$$

where $m_{i,j}$ is mineral i mass fraction size class j. Similarly $E_{i,j}$ is the entrainability parameter of mineral i size class j.

These recovery equations are based on the feed into the cell and thus are not dynamic models. To overcome this, it was assumed that the flow by entrainment operated with similar dynamics to the recovery of water (for which a dynamic model is available). This assumption presumes that the rate of entrainment is a function of the cell conditions, in particular the pulp composition. The starting point for the development of a dynamic model is Equation 4.22:

$$\begin{aligned} R_{ent,j} &= E_j R_w \\ \Rightarrow \frac{C_{ent,j}}{F_j} &= E_j \frac{C_w}{F_w} \end{aligned} \tag{4.24}$$

where:

$C_{ent,j}$	Rate of flow by entrainment of mass fraction j to the concentrate	$\left[\frac{m}{t}\right]$
F_j	Feed rate of mass fraction j	$\left[\frac{m}{t}\right]$
C_w	Flowrate of water to the concentrate	$\left[\frac{m}{t}\right]$
F_w	Feed rate of water to the cell	$\left[\frac{m}{t}\right]$

Equation 4.24 can be rearranged as follows:

$$C_{ent,j} = E_j C_w \frac{F_j}{F_w} \tag{4.25}$$

The above equation shows that the rate of entrainment is based on the ratio between the feed rate of the entrained species, and the feed rate of water. This is a steady-state equation that is not fundamentally based on the composition of the cell. To relate the rate of removal by entrainment to the slurry composition within the cell it was assumed that the actual driving force for the equation is the ratio between the amount of size class j and water in the cell (X_j and X_w respectively). However, using this assumption could cause the experimentally-determined degree of entrainment (E_j) to be invalid. Fortunately the linear relationship between the degree of entrainment and the water recovery ensures that the ratio of entrainable minerals to water in the cell remains very close to the ratio in the feed. Proof of this is as follows:

Due to the cell being well mixed:

$$\frac{T_j}{T_w} = \frac{X_j}{X_w}$$

where T_j and T_w are the rates of mass flow to the tailings of mineral size-class j and water respectively. X_j and X_w are the masses of j and water in the cell.

At steady-state, with the feed rate of mineral j and water being F_j and F_w, and the rate of flow to the concentrate of the same being C_j and C_w:

$$\begin{aligned}
\frac{T_j}{T_w} &= \frac{F_j - C_j}{F_w - C_w} \\
\text{However, for } C_j &\approx C_{ent,j} \\
&= \frac{F_j - E_j C_w \frac{F_j}{F_w}}{F_w - C_w} \\
\frac{T_j}{T_w}(F_w - C_w) &= F_j - E_j C_w \frac{F_j}{F_w} \\
\frac{T_j}{T_w}\frac{(F_w - C_w)}{(F_w - E_j C_w)} &= \frac{F_j}{F_w} \\
\frac{F_j}{F_w} &= \frac{(F_w - C_w)}{(F_w - E_j C_w)}\frac{T_j}{T_w}
\end{aligned}$$

$$\frac{F_j}{F_w} = \frac{(F_w - C_w)}{(F_w - E_j C_w)} \frac{X_j}{X_w}$$

$$\therefore \text{ if } F_w \gg C_w :$$

$$\frac{F_j}{F_w} \approx \frac{X_j}{X_w}$$

Thus as long as the feed rate of water is considerably larger than the rate of water recovered to the concentrate, the ratio within the cell of entrained mineral classes to water is very similar to the feed. It is important to note that this proof only applies to those mineral classes which are primarily recovered by entrainment. The relationship does not hold for species in which true flotation or any other recovery mechanism plays a significant role.

In conclusion it can be shown that as long as the flow of water to the concentrate is much less than the flow in the feed:

$$\frac{F_j}{F_w} \approx \frac{X_j}{X_w} \tag{4.26}$$

Those species for which both entrainment and true flotation play significant roles, the empirical entrainability parameters would need to be found based on the state within the cell, not on the cell feed composition. This also applies to cells where the flowrate of water to the concentrate is of similar order of magnitude as that of the feed. This could happen in the final flotation stages (cleaners).

Thus the dynamic equation for entrainment is:

$$C_{ent,j} = E_j C_w \frac{X_j}{X_w} \tag{4.27}$$

The entrainability E_i is found using the partition curve described by Equation 4.10. Although the entrainability parameters ξ and δ in the partition curve are dependent on air flowrate, at present the model assumes their values are constant over the operating ranges used. It has been determined experimentally that these parameters vary linearly with air flowrate (Savassi et al., 1998), but their variation needs to be determined experimentally and cannot at this time be predicted.

Distributed Parameter Model

To model the flotation of an ore, it is split up into a number of flotation classes. The reason for this is that different species in the ore have different flotation parameters, and may have

different dominant mechanisms of flotation . The flotation classes can be based on a number of distinguishing characteristics, including particle size, liberation and floatability. In this model, floatability was used to divide the mineral into different classes. Each of these classes had a dominant mechanism of recovery to the concentrate, either true flotation or entrainment.

Harris (2000) made use of a parameter, ϕ_i, as a means to specify the dominant mechanism driving the recovery of a certain mineral class. As shown in Equation 4.11, ϕ_i represents the fraction of the feed which floats by true flotation. For the mineral classes with flotation as the dominant mechanism ϕ_i is taken as 1, whereas for the non-floating classes it is zero. This was extended for use in the dynamic model by assuming similar ratios hold, and using ϕ_i and $(1 - \phi_i)$ to weight the true flotation and entrainment rates respectively. Mechanistically this is not strictly true, as all the mineral class is available for recovery by entrainment. However, the approximation was used to allow the steady-state empirical rate constants to be used. In all the mineral classes, ϕ_i was either very close to zero or one, and thus its effect was small.

Summary of model

Using the notation and relationships from the sections above, the dynamic flotation model can be written as:

$$\dot{X}_i = F_i - T_i - C_i \tag{4.28}$$

$$\dot{X}_w = F_w - T_w - C_w \tag{4.29}$$

where F_i is the feed to the system, and

$$T_i = \rho_{cell}.g.K_v.H\frac{X_i}{V}vo$$

with T_w equivalent to T_i with $i = w$.

$$C_i = \phi_i \left(P_i S_b R_f X_i\right) + (1 - \phi_i) \sum_{j=1}^{d} \left(E_j C_w \frac{X_{i,j}}{X_w}\right) \tag{4.30}$$

$$C_w = P_w S_b R_{f,w} X_w$$

$X_{i,j}$ is the j^{th} mass fraction of mineral species i:

$$X_{i,j} = m_j X_i \tag{4.31}$$

To show the form of this model, it is rewritten below grouping constants and showing the state and control variables as x_i and u_i respectively. Two basic equations exist, the dynamics of the solid species and the cell water. Constants have been grouped where possible, resulting in an explicit equation system in the state and control variables.

For the recovery of solids, the equation is as follows:

$$\begin{aligned}\dot{x}_i &= \Lambda_i - \\ &\theta_i e^{\left(-\beta\left(\frac{\alpha_1 - \sum_{k=1}^{n+1}\frac{x_k}{\rho_k}}{\alpha_6 e^{\left(\alpha_7 - \alpha_3\sum_{k=1}^{n+1}\frac{x_k}{\rho_k}\right)}}\right)u_2^{-c}x_w^{-1}\right)} x_i u_2^c + \\ &\mu_i \frac{\left(\alpha_2 e^{\left(\alpha_3\left(\alpha_1 - \sum_{k=1}^{n+1}\frac{x_k}{\rho_k}\right)\right)} x_w u_2^c\right)}{x_w} \sum_{j=1}^{n}\{\gamma_{i,j}x_i\} - \\ &\frac{\alpha_8\left(\sum_{k=1}^{n+1} x_k\right)}{\sum_{k=1}^{n+1}\frac{x_k}{\rho_k}} x_i u_1 \quad i = 1 \ldots n\end{aligned} \tag{4.32}$$

The dynamic mass balance for water has slightly different dynamics to the solids, and can be summarised as:

$$\dot{x}_w = \Lambda_w - \left(\alpha_2 e^{\left(\alpha_3\left(\alpha_1 - \sum_{k=1}^{n+1}\frac{x_k}{\rho_k}\right)\right)} x_w u_2^c\right) - \left(\frac{\alpha_8\left(\sum_{k=1}^{n+1} x_k\right)}{\sum_{k=1}^{n+1}\frac{x_k}{\rho_k}} x_w u_1\right) \tag{4.33}$$

where:

x_i	Mass of species x in the cell	$[m]$
x_w	Mass of water in the cell	$[m]$
u_1	Valve opening percentage	$[\%]$
u_2	Air sparge-rate	$\left[\frac{L^3}{t}\right]$
$\alpha_i, \beta, \rho_i, \gamma_{i,j}, \mu_i, \Lambda_i, \theta_i$	Lumped system parameters	
n	Number of solid species	

The nonlinearity of this system is quite apparent, in all the state variables and in u_2. The system is affine in u_1, the valve opening.

4.3 Open-Loop Simulation Results

4.3.1 Model implementation details

The system simulated is a small 30l pilot cell, with dimensions (weir height, impeller size etc) fitting a cell of that volume. The flotation aims to recover platinum group metals from an ore containing chromite and gangue. This ore comes from the UG2 reef in the North-West province of South Africa. Most of the parameters and values were taken from the PhD. thesis by T. Harris (Harris, 2000), and can be found in Appendix A.

Table 4.2 shows how the ore was split up for the model, and shows the dominant mechanism for the recovery of each mineral class.

Table 4.2: Species distribution used for flotation model

Fraction number	Slurry species	Primary mechanism of recovery
1	Fast floating PGM's	True flotation
2	Slow floating PGM's	True flotation
3	Non-floating PGM's	Entrainment
4	Chromite	Entrainment
5	Gangue	Entrainment
6	Water	Water recovery model

One of the reasons chromite is modeled separately from the gangue is that it has unique flotation and entrainment parameters. It is also an important variable as it is vital that the level of chromite reporting to the concentrate be kept as low as possible. Chromite complicates the smelting process, and must be kept below a certain level.

4.3.2 Linearisation of control variables

The model shown above is a highly nonlinear function of the air flowrate. As flowrate was chosen to be an input variable, it was decided to linearise the model around it at a reasonable operating level. By doing this the optimal control problem would be affine in all the input variables allowing the minimisation of the Hamiltonian to be considerably simplified, and thus increase program execution speed. The right hand side of Equations

4.34 and 4.35 represent the nonlinear dynamics of the system:

$$\begin{aligned} f_i(\mathbf{x},\mathbf{u}) \;=\; \Lambda_i - & \\ & \theta_i e^{\left(-\beta\left(\frac{\alpha_1-\sum_{k=1}^{n+1}\frac{x_k}{\rho_k}}{\alpha_6 e^{\left(\alpha_7-\alpha_3\sum_{k=1}^{n+1}\frac{x_k}{\rho_k}\right)}}\right)u_2^{-c}x_w^{-1}\right)} x_i u_2^c + \\ & \mu_i \frac{\left(\alpha_2 e^{\left(\alpha_3\left(\alpha_1-\sum_{k=1}^{n+1}\frac{x_k}{\rho_k}\right)\right)}x_w u_2^c\right)}{x_w} \sum_{j=1}^{n}\{\gamma_{i,j}x_i\} - \\ & \frac{\alpha_8\left(\sum_{k=1}^{n+1}x_k\right)}{\sum_{k=1}^{n+1}\frac{x_k}{\rho_k}} x_i u_1 \quad i = 1\ldots n \end{aligned} \tag{4.34}$$

$$f_w(\mathbf{x},\mathbf{u}) = \Lambda_w - \left(\alpha_2 e^{\left(\alpha_3\left(\alpha_1-\sum_{k=1}^{n+1}\frac{x_k}{\rho_k}\right)\right)}x_w u_2^c\right) - \left(\frac{\alpha_8\left(\sum_{k=1}^{n+1}x_k\right)}{\sum_{k=1}^{n+1}\frac{x_k}{\rho_k}} x_w u_1\right) \tag{4.35}$$

Linearising the input variable u_2 at the nominal operating value Q_{ss} results in the system being represented by:

$$\dot{x}_i = f_i(\mathbf{x},\mathbf{u})|_{u_2=Q_{ss}} + \left.\frac{\partial f_i(\mathbf{x},\mathbf{u})}{\partial u_2}\right|_{u_2=Qss}(u_2 - Q_{ss}) \tag{4.36}$$

$$\dot{x}_w = f_w(\mathbf{x},\mathbf{u})|_{u_2=Q_{ss}} + \left.\frac{\partial f_w(\mathbf{x},\mathbf{u})}{\partial u_2}\right|_{u_2=Qss}(u_2 - Q_{ss}) \tag{4.37}$$

Clearly Equations 4.36 and 4.37 are still nonlinear in the state variables, but have been linearised in the input variables.

Figure 4.5 compares the effect of the linearised input variables to the unmodified nonlinear model, by showing the rate of change of system state (at a nominal state condition) of each of the mass classes within the cell as a function of Q_{air}.

The relationship for the valve opening is already linear, and thus did not need to be linearised.

The effect of this linearisation on the model is shown in the open-loop results, where both linearised and nonlinearised results are shown for step changes to the air flowrate.

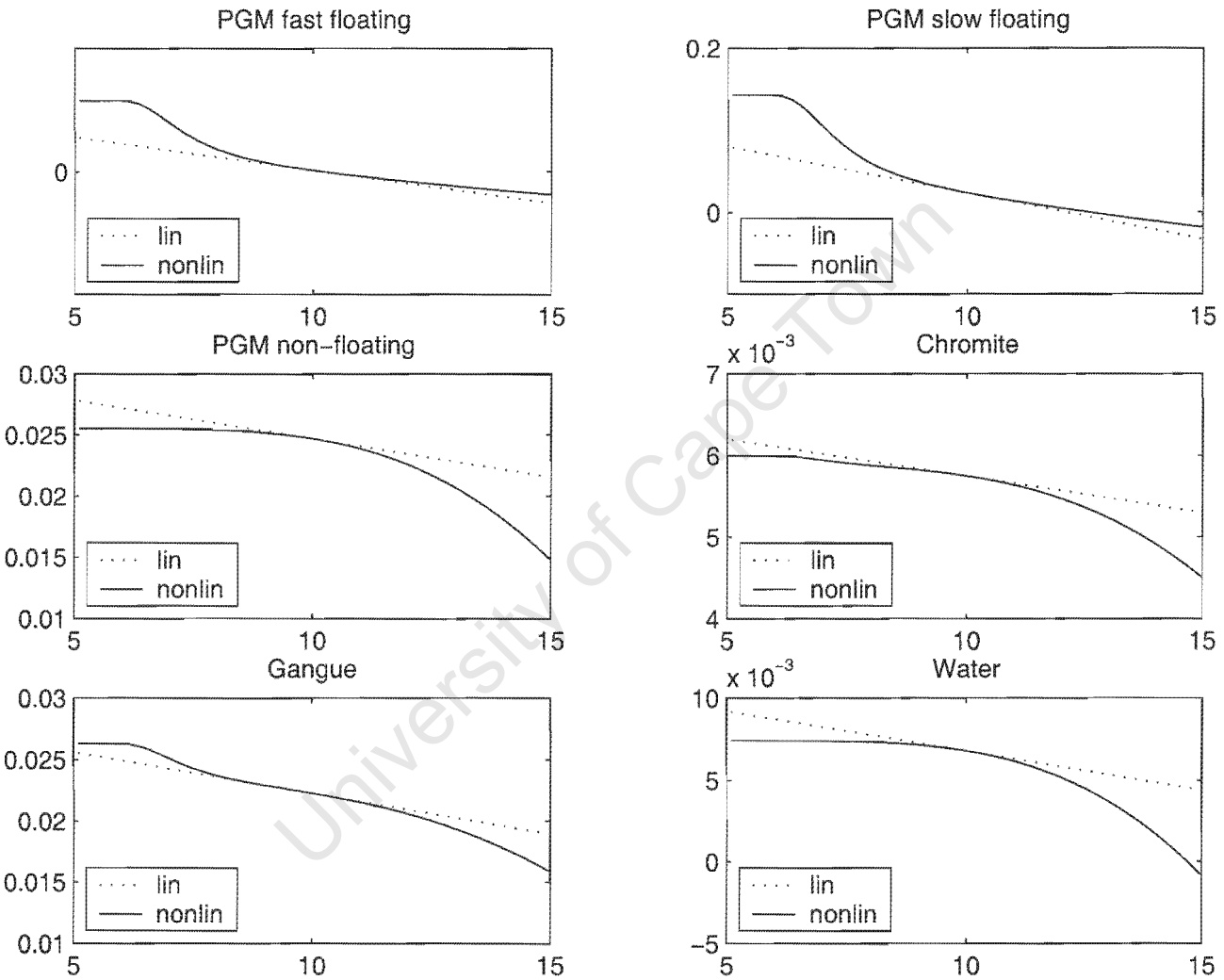

Figure 4.5: Effect of Q_{air} linearisation on rates of flotation of floatability classes

4.3.3 Simulation and testing

A sequence of open-loop tests were performed to highlight the dynamic response of the model to changes in input variables. Figures 4.6, 4.7 and 4.8 show the effect of a 20% increase in air rate on recovery, grade and tailings fluctuation. The tailings fluctuation is the rate of change of tailings flowrate - an important variable as it can affect the stability of downstream cells.

All of these graphs compare the models linearised in the air sparge-rate to the unmodified nonlinear models. In all cases it is clear that the linearisation of the input variable (Q_{air}) does not result in a large deviation from the nonlinear results. In all cases the model is nonlinear in the state variables.

Step change in air sparge-rate (Q_{air})

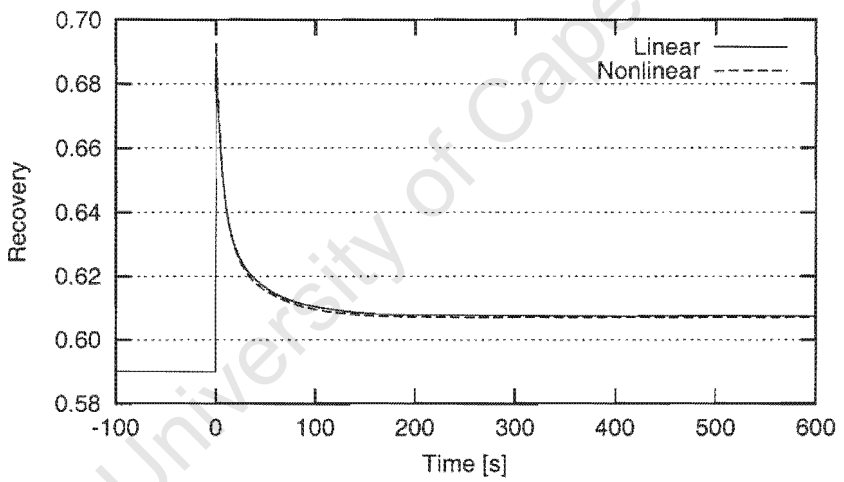

Figure 4.6: Open-loop response of recovery to step change in Q_{air}

As is expected, Figure 4.6 shows a large immediate increase in recovery, and improvement at steady-state. This is due to the increased air rate increasing both the true flotation rate through an increased bubble surface area flux, and the rate of entrainment through increased water recovery rate. The immediate response to the step change in sparge-rate is due to the assumption of negligible dynamics in the froth, resulting in instantaneous recovery through the froth.

The large overshoot is due to the initial high concentration of valuable minerals within the cell, and the increased sparge-rate. With a higher concentration of valuables in the

slurry, the increase in sparge-rate causes a large increase in the recovery. Due to recovery being based on the feed rate of minerals to the cell (which is constant), the large increase in mineral flowrate to the concentrate stream results in a similarly large increase in the recovery. In certain instances (e.g. a steady-state with low recovery with subsequent large increase in sparge-rate) the recovery can exceed 100% (for short periods of time). With the increased sparge-rate, the valuable mineral concentration in the cell rapidly decreases resulting in the recovery dropping to its new steady-state value.

However, increasing the recovery also increases the rate at which gangue and chromite report to the concentrate. This will result in a lowering of the grade, and this result is seen in Figure 4.7. Once again the input-linearised and nonlinear graphs are very similar.

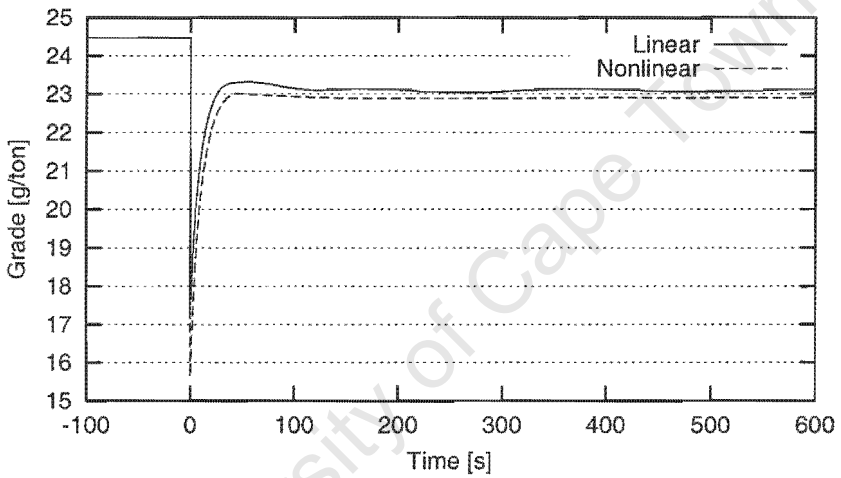

Figure 4.7: Open-loop response of grade to step change in Q_{air}

The sudden increase in air rate results in an increase in the flow to the concentrate. This lowers the level of the cell and causes the pressure head to decrease. As a result the tailings flowrate is expected to decrease, but only for a short period of time, and by a small amount. The fluctuation this causes in the tailings rate would thus be quite small and rapid. This fluctuation is unlikely to cause significant disturbances further downstream. Figure 4.8 shows the short, sharp disturbance that rapidly vanishes.

Step change in valve opening (vo)

Figures 4.9 and 4.10 show the effect of a 20% increase in valve opening on the liquid level within the cells, and the tailings fluctuation rate. Opening the valve leads to a decrease in the cell level. The response is not expected to be very fast, and Figure 4.9 shows that

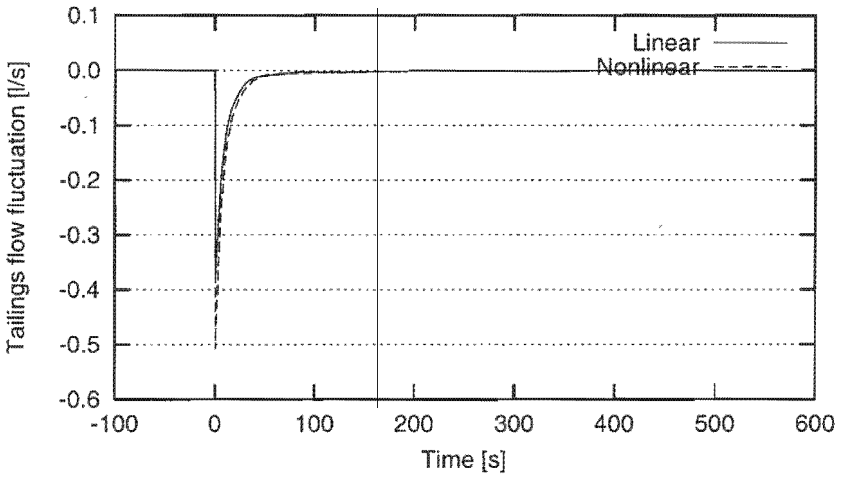

Figure 4.8: Fluctuations in tailings flowrate due to step change in Q_{air}

the effect takes several minutes to steady out. This fluctuation is also expected to have a strong effect on the rate of change of the tailings flow (Figure 4.10). The change in valve opening results in a far larger and longer-lasting effect than the change in air flowrate shown in previous figures.

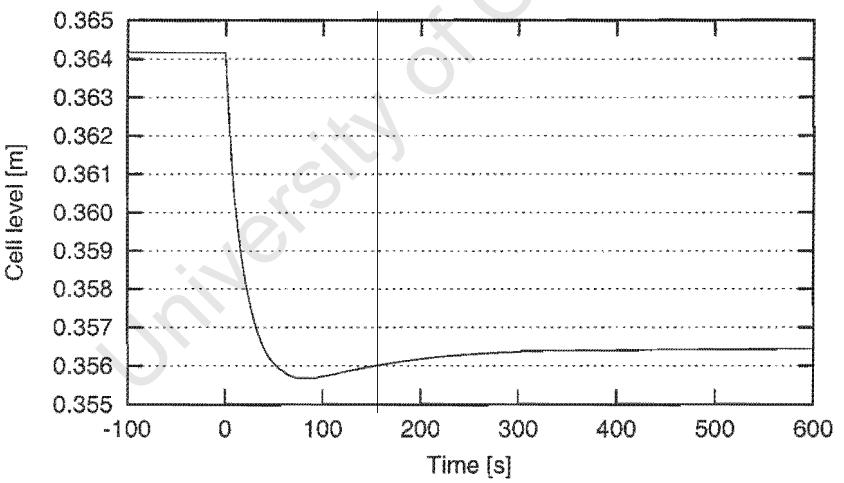

Figure 4.9: Open-loop response of cell level to step change in *vo*

Directly after the change to valve opening, the the tailings flowrate starts to decrease due to the tank level decreasing. This results in a negative rate of change of tailings flowrate, as shown in Figure 4.10. As the level within the tank settles down to its new steady-state, the tailings flow decreases until it too is steady resulting in a zero rate of change.

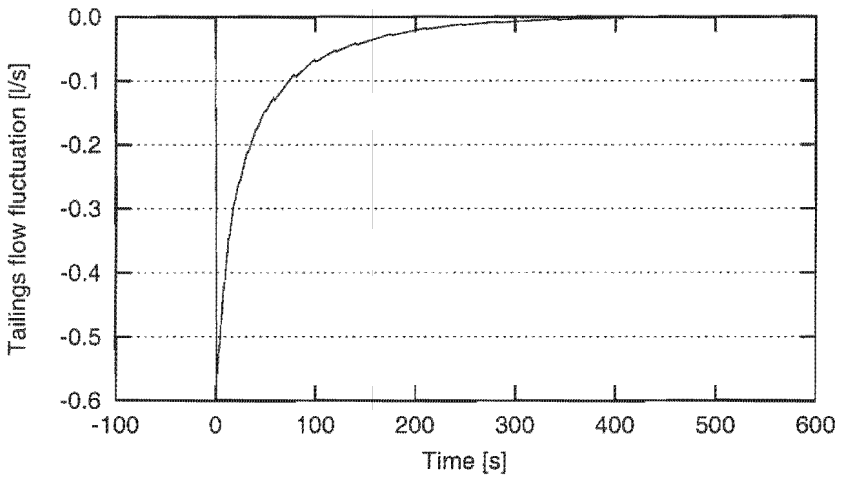

Figure 4.10: Fluctuations in tailings flowrate due to step change in *vo*

Conclusion

A dynamic flotation model has been produced which qualitatively follows expected flotation behaviour. Air sparge-rate and valve opening can be used as input variables, and the model can be expanded to form banks and finally a full flotation circuit. This model was produced using modified steady-state models and parameters, and predicts recoveries and grades very close to the anticipated values.

Chapter 5

Application of NMPC to a Flotation Unit

5.1 Review

Flotation control has a history around 30 years long, starting with the introduction of on-line stream analysis (OSA) instruments. By using on-line assay measurements, insight was gained into the dynamics of flotation, and the types of disturbances being experienced in flotation circuits. It was shown that circuits were experiencing considerable fluctuations (both in magnitude and frequency) of the grade and recovery of concentrate and tailings streams (McKee, 1991). These upsets resulted in poor performance, and consequently an interest in control to improve the situation. Control theory offered the opportunity to stabilise the systems, improving metallurgical performance and increasing profit. However, the complexity of flotation systems has meant that flotation control to this day leaves much to be desired.

Flotation circuits are highly nonlinear, highly interacting systems existing in a harsh environment and governed by variables and parameters that are difficult to measure. The circuits experience a number of process upsets, including changes within the ore body, upstream operational changes (e.g. to the milling circuit) and time dependent changes to the system dynamics (e.g. wear on machinery, oxidation of minerals). These upsets destabilise the system and lead to a degradation of flotation performance.

Flotation systems are also subject to large time delays, especially when considering the effect of a control action on the entire plant. Due to the highly complex nature of flotation,

and the number of system parameters, the response of the system can change with no obvious reason. The result of this is that controller tuning can be extremely difficult, or need to be updated regularly.

Nonetheless, considerable advancement in minerals processing technology has resulted in the process of flotation being far better understood, and produced a number of instruments that can be used to accurately determine the state of the system. This means that advanced flotation control is rapidly becoming a viable option, although there are still a number of issues to be overcome. A considerable amount of literature on the subject is now available.

5.1.1 Control Objectives and Disturbances

When designing a control system for a plant, the objective of the control must be carefully considered. It is generally felt that there are three tiers of control objective that should be sequentially applied to a plant. At each of the different tiers the objective of the control changes, affecting the calculation of the input moves. These three tiers are (Lynch et al., 1981; McKee, 1991):

- Regulatory control - stabilisation of the plant at a setpoint
- Supervisory control - achieve an optimal grade and/or recovery setpoint
- Optimising control - maximise the economic benefit of the plant

Regulatory control of the plant involves attempting to remove process fluctuations, ensuring that the grade and recovery are held constant at a given setpoint. This does not mean that the plant is operating optimally. However, if a plant is able to be stabilised at an arbitrary recovery and grade, then an optimal setpoint can be chosen which the control system can then attempt to satisfy, moving the control objective to the next tier.

Supervisory control uses additional knowledge of the process to determine setpoints for the circuit. The controller then attempts to choose inputs for the process which cause the system to respond in a way that moves it towards these setpoints. The setpoints can be chosen by the plant operators using a grade-recovery curve for the circuit, and is usually based on requirements of smelters or refineries further downstream. With good system stability it may be possible to move to a higher level grade-recovery curve, thus improving system performance. Before this level of control is attempted, the controller must be able

to keep stability in the plant, enabling the controller to seek stability at the desired plant setpoints.

Optimising control is the final tier of control, where the economic benefit of the plant is maximised. At this level the setpoints are chosen to optimally satisfy an objective function that quantifies cost within the plant. This control objective will be a function of the price of the mineral, the cost structures involved with the sale of the mineral and purchasing of plant consumables, and the operating and mining strategies. Other factors may also be considered, and its form will vary from circuit to circuit. Unless the controller is able to satisfy desired setpoints and ensure the plant's operation is stable, this level of control will not succeed. This mode of control is only used if important system assays and flows are within prescribed limits.

However, no matter which type of control objective is employed there are certain factors and physical constraints that may prevent the controller from achieving its goal. Physical circuit constraints can include equipment limitations or upstream sections of the plant performing poorly. Oversized valves may prevent the controller from achieving fine enough flowrate control, pumps that are too large may cause surging. If the grinding circuit supplying feed to the flotation plant is not grinding fine enough the mineral may not be sufficiently liberated, preventing the desired grade being achieved.

Process disturbances will also affect the controller's ability to achieve a desired result. This is especially important when attempting to stabilise the process. A number of process disturbances can occur, including changes to the plant throughput, variations in the ore, or even changes to the operating setpoints.

The grinding circuit can once again play an important role by ensuring that the feed to the circuit is as stable as possible. This requires good control on that section of the plant, to which a lot of research has already been devoted.

One of the major difficulties in flotation is changes within the ore body. Different ore deposits have different characteristics, and knowing whether the ore has changed, and how to adjust the circuit to handle the change, is virtually an art-form requiring extremely experienced operators. Often it calls for changes to both the control objective and adjustments to the flotation chemistry. Control systems at the present time are unable to make changes at this level, although expert systems (rules-based controllers) are finding use in this area.

Plant setpoint changes, although operator induced, do cause disturbances which may have an effect on the grade and recovery of the process. Controllers should be able to take this into account, executing smooth process changes and rapidly returning the plant to a stable state.

5.1.2 Types of Controller

The most basic level of control employs the use of conventional PID controllers. These are often used in small feedback loops to achieve setpoints in a cascade-control type fashion. They have been found to be unsuccessful in any situation where plant interactions may occur.

Another type of controller frequently used in the minerals processing industry is the *"expert system"*. This is a rules based controller, often using fuzzy logic to decide how best to respond to a given situation. Controllers of this sort recognise the importance of operators on plants, and attempt to react to plant disturbances in the same way that an operator would. Decisions are made based on rules developed in consultation with the plant operators, who are assumed to know how to optimally control the plant. Effectively this controller becomes an automatic operator, but has advantages over a human operator in that it is able to ensure that the control decisions on the plant are consistent, and not dependent on which operator is on duty and whether or not a change has been noticed. However the control action suffers from the problem that the optimal action is entirely dependent on the operators experience, which may well not be the mathematical optimum for the process.

Nonetheless expert systems are useful, often giving feedback to operators to correct disturbances quickly and thus stabilise the plant. Some systems even have speech synthesisers to inform operators when a problem is occurring. They can also be used to train inexperienced operators. However, they cannot be considered controllers in the conventional use of the term.

Model-based control includes the more advanced control strategies, but relies on an accurate description of the system being controlled. The system model must be able to reproduce the dynamic behaviour of the circuit. An optimiser is then used to choose the mathematically optimal set of control actions to satisfy the control objective. Adaptive controllers are popular choices of model-based controllers, although there are a number of other types

including model predictive control. The model of the process may be identified by off-line tests of the process, on-line tests, or fundamental models.

Off-line testing involves stepping system input variables and then capturing the time-based response of the system. This method is a well known control technique, but in the flotation context is not ideal as, over time, the system behaviour is known to change.

On-line model determination is more applicable as the model is updated when significant deviation between plant and model is noticed. Adaptive control algorithms have this ability.

Fundamental system modeling is extremely complex, and results in equations requiring a large number of parameters that need to be experimentally determined. However it does provide important insights into the behaviour of the system, and thus more informed decisions can be made regarding the controller tuning. Nonetheless this type of model is usually not feasible for on-line implementation.

5.1.3 Flotation Control

A model-based controller known as Global Predictive Control (Desbiens et al., 1998; Milot et al., 2000) has been used to control simple flotation units. Global Predictive Control (GlobPC) is an adaptive control strategy based on an Internal Model Control (IMC) structure. Tracking, regulation and feedforward dynamics are determined from three independent cost functions. This leads to perfect decoupling of the control modes (Milot et al., 2000).

In Desbiens *et al* (1998), GlobPC was used to simulate the control of a set of four rougher flotation units. The system was a multi-input, single-output configuration where the final system grade (of the combined cells) was controlled by manipulation of the collector addition. The original system included air flowrate in the rate term but it was found that this did not have a large effect on the flotation.

The model included a first-order rate equation, and an entrainment model. Four mineral classes were used:

- Mineral rich particles with fast kinetics
- Mineral poor particles with slow kinetics
- Floating gangue particles with slow kinetics

- Non-floating gangue particles

The gangue particles had small amounts of valuable mineral locked in them. The collector and air rate were related to the rate constants, the proportion of slow-floating gangue, and the entrainment coefficients by sigmoidal functions coupled to transfer functions modeling the conditioning stage. Further details of the model were not disclosed.

GlobPC is only able to handle linear models, but the models for the rougher units were nonlinear. This was overcome by linearising the models in three operating regimes, with interpolation between the regimes as the system moved from one to the next. It was found that the use of linear regimes allowed good operation over a wide range of operating conditions. However, the focus of the investigation was on the GlobPC algorithm and not on flotation control, and the controller was not tested on a real flotation system.

Milot *et al* (2000) tested the use of GlobPC on a flotation column, manipulating the tailings-flow and wash-water setpoints to control the pulp-froth interface height and bias (downward flow of wash water). Both of the output variables were estimated using virtual sensors. The pulp-froth interface height was determined using a conductivity probe, with the point of inflection in the conductivity representing the interface point. The bias was estimated using a neural-network technique. It is important to note that the flotation column was only a two-phase laboratory device which did not contain any solids.

The column model was identified using pseudo-random step changes separately in each of the input variables (holding the other constant). A software package (ADAPTX©) was used to determine the parameters for each model, resulting in a set of nonlinear models (Figure 5.1).

It was found that $G_{22}(q^{-1})$ was dependent on the wash-water setpoint, and so three local models were developed that covered the regime of operation. The output of this model was then interpolated depending on the regime of operation. The influence of $G_{12}(q^{-1})$ was found to be weak and so it was estimated to be zero. Decoupling was performed on the other models, to allow the use of two monovariable GlobPC controllers.

By using local linear models, GlobPC was able to be used on a nonlinear system, and the froth-pulp interface height and system bias were able to be regulated using the GlobPC structure.

A study of the use of Generalised Predictive Control (GPC) and Long-Range Predictive Control was undertaken by Prince (1996). These two strategies are adaptive control algo-

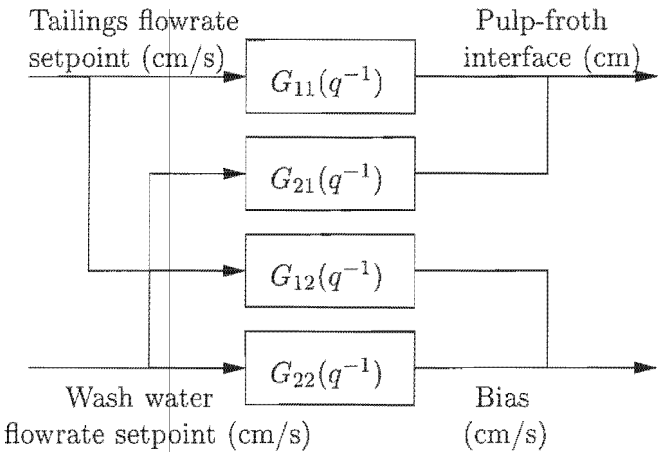

Figure 5.1: Flotation column process model

rithms, with the controller design being based on an on-line determination of the process model. The controller design itself may occur on-line.

LRPC differs from GPC in the method by which the model is determined. GPC uses a recursive least squares method to determine the model parameters, which is incorporated into the model. It has been suggested that this method is not optimal for the GPC strategy, and that the method for determining the model parameters should be linked to the GPC algorithm. This led to the development of long range process identification, with the associated control algorithm being LRPC.

The system used in this study was a simplified flotation circuit, containing four cells with the objective being to control the tank levels. A four-input, four-output system resulted, with the valve openings as the manipulated variables.

It was found that the LRPC algorithm outperformed the GPC method, although the simplified system cannot be regarded as an adequate representation of a flotation circuit.

An industrial implementation of the GPC algorithm was studied by Suiches *et al* (2000). This was a single-input, single-output configuration controlling the tailings grade of the cell by manipulating collector addition rate. The system being studied used reverse flotation, removing pyrite to the concentrate to improve the grade of the tailings stream.

As with other adaptive control methods, the system model was identified on-line avoiding fundamental understanding and modeling of the system. It was found that above a certain collector addition rate the rate of pyrite flotation levelled off but the valuable mineral flotation rates kept increasing. This led to an increased amount of valuable mineral reporting

to the concentrate and consequent loss of recovery in the tailings. To avoid this an upper limit was placed on the collector addition rate.

The installation of this control system led to an improvement in tailings grade from 52.8% to 54.2% (Suichies et al., 2000).

A fundamental approach to flotation modeling, with the aim of implementing a predictive controller was undertaken by Pérez-Correa *et al* (1998). The system was modeled using a mass balance but with empirical relationships where necessary. The basic assumptions were:

- Two phase system: pulp and froth
- Two mineral classes: mineral rich and mineral poor (gangue)
- Within each of the phases the system was perfectly mixed
- Mass transfer can occur in both directions between the two phases
- The density within the pulp and the froth remains constant
- There is one particle size

The controller was able to manipulate the frother addition rate and the cell pulp level. Disturbance variables were taken to be the feed grade and flowrate, the particle size within the feed, and the iron feed content. The outputs that could be measured were the tailings and concentrate copper assay. The concentrate and tailings flowrate were unmeasured outputs.

To simplify the model, the banks of cells were modeled as one cell. Due to the degree of intermixing within the banks, this was regarded as a fair approximation (Pérez-Correa et al., 1998). The final system model consisted of four rougher and three cleaner sections.

Two rules-based controllers were also tested, to simulate normal plant operation. The objective of these controllers was to control general concentrate and tailings grade. The first controller aimed to ensure that the controlled variables remained within a given operating region. The operating region described a minimum general concentrate grade and a maximum tailings grade, achieved by manipulating four sets of pulp levels within the plant.

The second rules-based controller was designed to avoid input saturation. This was achieved by altering the zone of operation of the previous controller, to choose a region of operation which minimised saturations.

Two predictive controllers were tested: dynamic matrix control (DMC) and quadratic dynamic matrix control (QDMC). These are two linear model-based controllers, but differ in the method by which the optimal control trajectory is chosen. For the case of DMC the optimisation is a linear least-squares minimisation, whereas QDMC uses a quadratic programming routine. The DMC algorithm is far less computationally intensive, with the minimisation performed off-line, and each control move calculated by matrix multiplication. However, it is unable to include input constraints, and thus is of limited use. The QDMC method is able to handle constraints, but is far more computationally intensive.

The rules-based controllers were found to achieve the desired control, although the first tended to saturated the inputs. The predictive controllers performed well, with small output deviations and smooth control. QDMC was found to perform better than DMC, and satisfied constraints consistently (Pérez-Correa et al., 1998).

A simple supervisory control scheme is described in Herbst, J. A. and Pate, W. T. (1986). A copper flotation plant consisting of two rougher and one scavenger sections is controlled at a supervisory level by choosing the steady-state input values that maximise a profit-based objective function (also at steady-state).

A Kalman filter is used with a model of the first two rougher cells of the circuit to find changes in the floatability characteristics of the feed. The model for the Kalman filter is a simplified dynamic mass balance of the froth and pulp phases, accounting for four different mineral species. Using the floatability estimate from the Kalman filter and a model of the circuit, the performance of the plant can be predicted. A steady-state control action is then chosen to best satisfy an economic performance function of the plant. This results in a type of feedforward controller which is able to deal with the extremely slow system response. In this way the effect of the disturbance can be minimised by appropriate changes to the input variables before it has propagated through the circuit.

The controller was able to manipulate the frother and collector addition rates, and the pulp level within the cells. The performance equation is a function of the weighted costs of frother, collector and pulp level, and the mass flow of copper to the concentrate. If the grade of the concentrate stream fell below a certain minimum, then this was also included in the performance equation.

The performance of this feedforward scheme was compared to a standard PI system. With periodic changes to the feed floatability, the feedforward scheme was much better at controlling the recovery (Herbst and Pate, 1986).

5.2 System Description

The system configuration chosen to be used as a study of NMPC on flotation was the first rougher flotation cell of a pilot flotation circuit. The circuit concentrates a UG2 platinum-bearing ore, with a fairly high concentration of undesirable chromite. The majority of the parameters from this system were taken from Harris (2000), and can be found in Appendix A.

Two manipulated variables were chosen, constrained to operate in physically realisable regimes. They are the tailings valve opening, and the sparge air flowrate. The controller aimed to satisfy setpoints on the tank level and a concentrate grade. An additional output variable included in the optimisation was the fluctuation in the tailings flowrate. The inclusion of this variable aimed to stabilise the downstream system by preventing large flow fluctuations in the tailings from the controlled rougher cell.

The only disturbance variable considered was changes to the class-specific mineral feedrates. Changes to the feed are well known in industry, due to changes in the ore from the mine, and variations in the upstream milling circuit. Setpoints were chosen based loosely on open-loop results and empirical data.

5.3 Problem Formulation

The optimal control formulation was as follows:

$$\min_{\mathbf{u}(\cdot)} V(\mathbf{u}) = \psi(\mathbf{x}, \mathbf{u}) \tag{5.1}$$

$$V(\mathbf{u}) = w_{T_p} (H_0 - H)^2 \,|_{t=T_p} + \int_0^{T_p} \left\{ w_1 (H_0 - H)^2 + w_2 (G_0 - G)^2 + w_3 \left(\frac{dQ_T}{dt} \right)^2 \right\} dt \tag{5.2}$$

where:

$\mathbf{u}$	$\begin{bmatrix} vo \\ Q_{\text{air}} \end{bmatrix}$	$\begin{bmatrix} \% \\ \frac{l}{min} \end{bmatrix}$
w_i	Output cost weighting	
w_{T_p}	Terminal penalty cost weighting	
H_0	Slurry level setpoint	$[cm]$
H	Cell slurry level	$[cm]$
G_0	Grade setpoint	$[g/ton]$
G	Concentrate grade	$[g/ton]$
Q_T	Tails volumetric flowrate	$\left[\frac{m^3}{s}\right]$
T_p	Prediction horizon	$[s]$

The objective function is constrained by the system dynamics represented by the following equation system, developed in Chapter 4:

$$\begin{aligned} f_i(\mathbf{x},\mathbf{u}) \;=\; & \Lambda_i - \\ & \theta_i e^{\left(-\beta\left(\frac{\alpha_1-\sum_{k=1}^{n+1}\frac{x_k}{\rho_k}}{\alpha_6 e^{\left(\alpha_7-\alpha_3\sum_{k=1}^{n+1}\frac{x_k}{\rho_k}\right)}}\right)u_2^{-c}x_w^{-1}\right)} x_i u_2^c + \\ & \mu_i \frac{\left(\alpha_2 e^{\left(\alpha_3\left(\alpha_1-\sum_{k=1}^{n+1}\frac{x_k}{\rho_k}\right)\right)}x_w u_2^c\right)}{x_w} \sum_{j=1}^{n}\{\gamma_{i,j}x_i\} - \\ & \frac{\alpha_8\left(\sum_{k=1}^{n+1}x_k\right)}{\sum_{k=1}^{n+1}\frac{x_k}{\rho_k}} x_i u_1 \quad i = 1\ldots n \end{aligned} \tag{5.3}$$

$$f_w(\mathbf{x},\mathbf{u}) = \Lambda_w - \left(\alpha_2 e^{\left(\alpha_3\left(\alpha_1-\sum_{k=1}^{n+1}\frac{x_k}{\rho_k}\right)\right)}x_w u_2^c\right) - \left(\frac{\alpha_8\left(\sum_{k=1}^{n+1}x_k\right)}{\sum_{k=1}^{n+1}\frac{x_k}{\rho_k}} x_w u_1\right) \tag{5.4}$$

$$\dot{x}_i = f_i(\mathbf{x},\mathbf{u})|_{u_2=Q_{ss}} + \left.\frac{\partial f_i(\mathbf{x},\mathbf{u})}{\partial u_2}\right|_{u_2=Qss} (u_2 - Q_{ss}) \tag{5.5}$$

$$\dot{x}_w = f_w(\mathbf{x},\mathbf{u})|_{u_2=Q_{ss}} + \left.\frac{\partial f_w(\mathbf{x},\mathbf{u})}{\partial u_2}\right|_{u_2=Qss} (u_2 - Q_{ss}) \tag{5.6}$$

In this set of equations the system constants have been lumped into the parameters α_l, β, ρ_i, $\gamma_{i,j}$, μ_i, Λ_i, θ_i. The system dynamics are represented by Equations 5.5 - 5.6 which are linearised in u_2. These equations are nonlinear in the state.

The following constraints are included on the input variables:

$$vo_{min} \leq vo(t) \leq vo_{max} \tag{5.7}$$

$$Q_{\text{air},min} \leq Q_{\text{air}} \leq Q_{\text{air},max} \tag{5.8}$$

The valve opening constraints were $\{vo_{min}, vo_{max}\} = \{10\%, 90\%\}$, and the air sparge-rates $\{Q_{\text{air},min}, Q_{\text{air},max}\} = \{5[l/min], 15[l/min]\}$. These values were chosen based on recommended physical ranges taken from Harris (2000) and open-loop dynamic model responses.

The slurry level H is found by

$$H = \frac{\sum_{i=1}^{n+1} \frac{x_i}{\rho_i}}{A} \tag{5.9}$$

and the concentrate grade G by

$$G = \frac{\sum_{i=1}^{m} C_i}{\sum_{i=m+1}^{n} C_i} \tag{5.10}$$

where m and n are the number of valuable mineral species and the total number of mineral species respectively. C_i is the concentrate flowrate of species i (as shown in Section 4.2), and is a nonlinear function of the state variable $\mathbf{x}$, but is a linear function of the air sparge-rate (u_2 or Q_{air}) as a result of system linearisation in the input variables. A is the cell cross-sectional area, ρ_i is the density of species i, and x_i is the mass of species i in the cell. The $n+1^{th}$ species is water.

The concentrate grade is an approximation to simplify the formulation. Grade is defined as the mass of valuable mineral in the stream divided by the total mass of the stream. However, due to the mass of valuable mineral being extremely small (parts per million), the grade calculation was simplified by not including the mass of valuable mineral in the total mass of the stream.

The tailings fluctuation was determined using the partial derivative chain-rule:

$$\frac{dQ_T}{dt} = \sum_{i=1}^{n+1} \frac{\partial Q_T}{\partial x_i} \frac{dx_i}{dt} \tag{5.11}$$

State variable scaling

Due to the large ranges in operating values of the state variables (state values ranged from $1e-9$ to $1e+1$), the system equations were scaled to bring the values to within the same order of magnitude. Thus an unscaled nonlinear system equation

$$\frac{dx}{dt} = f(x,u) \tag{5.12}$$

was scaled by substituting $X = Sx \Rightarrow x = \frac{X}{S}$ where S is the scaling factor and X is the "better behaved" state variable:

$$\frac{d\frac{X}{S}}{dt} = f(\frac{X}{S},u) \tag{5.13}$$

$$\Rightarrow \frac{1}{S}\frac{dX}{t} = f(\frac{X}{S},u) \tag{5.14}$$

Appropriate choice of cost weightings

The choice of weightings for the cost function is important to ensure that the system is controlled to a sensible optimum. Due to the cost function not being an actual representation of a tangible cost, heuristic methods of choosing the weightings had to be employed. The procedure chosen was based on ensuring that deviations of 10% or less cause a cost contribution equal to or smaller than unity for each of the terms. Further assumptions had to be made for the flow fluctuation term. The details follow:

Level weighting: The basis for this weighting is that a deviation of 10% from the setpoint will result in an instantaneous cost contribution of 1. Thus:

$$w_1 = \frac{1}{(0.1H_0)^2} \tag{5.15}$$

Grade weighting: This is found in the same manner as the level weighting, with the aim

that a 10% deviation from the setpoint contributes a value of 1 to the cost function:

$$w_2 = \frac{1}{(0.1G_0)^2} \tag{5.16}$$

Flow fluctuation weighting: To determine this cost weight, time must be included in the weighting. It was decided that a 10% change in the tailings volumetric flowrate, Q_T, in a period of two minutes should cause an instantaneous cost contribution of unity. The nominal tailings flowrate was assumed to be approximately the same as the feed flowrate[1]. The weighting is thus found:

$$w_3 = \left(\frac{\Delta t}{0.1Q_F}\right)^2 \tag{5.17}$$

where Δt (taken to be 120 seconds) is the time allowed for the 10% deviation, and Q_F is the volumetric feed flowrate.

Initial guess for the optimal control problem

One of the most crucial elements of the solution of the flotation optimal control problem is correct problem initialisation. A poor initial guess results in the solver attempting to use the equations far from their nominal values. This can result in singular Jacobian approximations, and diverging results. To ensure that the problem was initialised correctly, an initial guess procedure was developed, similar to some of the hybrid methods suggested in Section 2.1.2.

1. Solve the pseudo-optimal control problem over a finite number of discrete time periods, for the entire length of the prediction horizon. This is *not* a solution of the optimal control problem, as the inputs are not optimised over the entire prediction horizon. The inputs are simply chosen as the optimal piecewise constant control over the discrete portion of the prediction length.

2. Use a linear approximation of the control input to remove discontinuities. Initially the piecewise constant control sections were used, but it was found that this resulted in the adjoint trajectory being non-differentiable at each of the transition points.

[1]This assumption is valid, as steady state industrial data shows that the concentrate flowrate from the first rougher cell only comprises approximately 5% of the feed flow (Harris, 2000)

3. Recalculate the state trajectory from this linear control approximation.

4. Calculate the adjoint trajectory by integrating the costate equations in reverse using the smoothed control and corresponding state profiles. If a terminal penalty was included on the cost functional, then this was used as an initial value for the reverse-integration of the adjoint system, calculated using state values at the terminal time.

This method results in an approximate initial guess which allowed fairly robust solution of the optimal control problem. In the NMPC implementation, the solution from the previous iteration also provided an adequate initial guess, which removed the need for the computationally demanding initial optimisation.

5.4 Results and Discussion

A number of simulations were conducted, to test the application of NMPC to flotation. These simulations included setpoint changes on the output variables, feed disturbances, and plant/model mismatch. The aim of these tests was to illustrate the use of NMPC on flotation, with piecewise continuously varying input trajectories. A comparison with conventional linear MPC was also performed.

For all these tests, the terminal penalty function was removed, and the prediction horizon was sufficiently long (900 seconds) to achieve steady operation . The sample time was 45 seconds, allowing 20 sampling periods in the simulation time.

Table 5.1 shows the initial system values used in all the controller tests.

Table 5.1: System initial state

Parameter	Value	Units
G_0	60	[g/ton]
H_0	35	[cm]
Q_{air_0}	9.027	[l/s]
vo_0	59.31	[%]

5.4.1 Test 1: Grade setpoint change

This test shows the response of the nonlinear controller to a change in grade setpoint. The controller attempts to keep the level constant while achieving a new grade setpoint.

The grade setpoint was stepped up at $t = 0$ from 60 [g/ton] to 70 [g/ton]. The slurry level was to be held constant at 35 [cm].

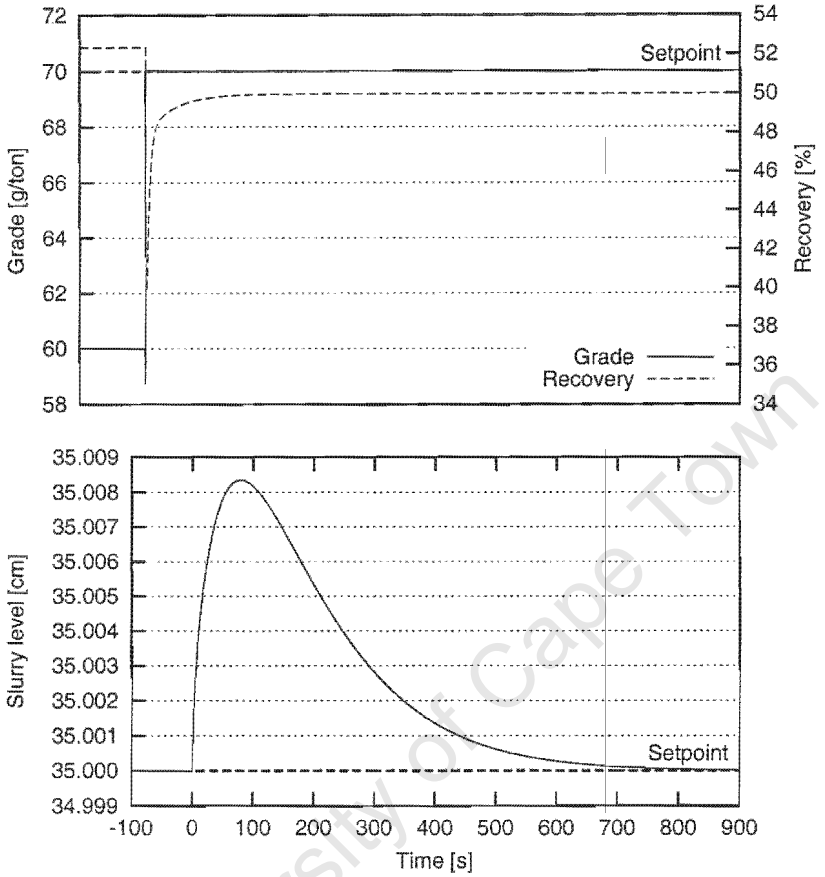

Figure 5.2: Test 1 - Grade setpoint change (output variables)

Figure 5.2 shows the response of the output variables to the setpoint change. The grade is seen to respond extremely quickly to the change. This near-instant response is caused by the assumption that froth recovery has negligible dynamics compared to the dynamics of the slurry.

Mathematically, the grade output equation is an explicit function of air sparge-rate, and can be inverted. Thus the exact air rate can be calculated from the state variables to satisfy any grade setpoint, within constraint bounds. As the state of the slurry changes due to the setpoint change, the air flowrate must be modified to ensure the grade setpoint remains constant. This is seen in Figure 5.3, with the air rate immediately being decreased, and then tending to a constant value as the cell reaches steady-state.

The increased grade is achieved by decreasing the air sparge-rate. This decreases the

mass-pull to the concentrate, and thus with less mass leaving the cell the slurry level rises. Consequently the valve opening must increase. The graphs of the system outputs and inputs show these effects. Clearly the dynamics of slurry level are considerably slower than grade, with it taking almost the full simulation time to reach steady-state. Nonetheless the maximum deviation from the level setpoint is small.

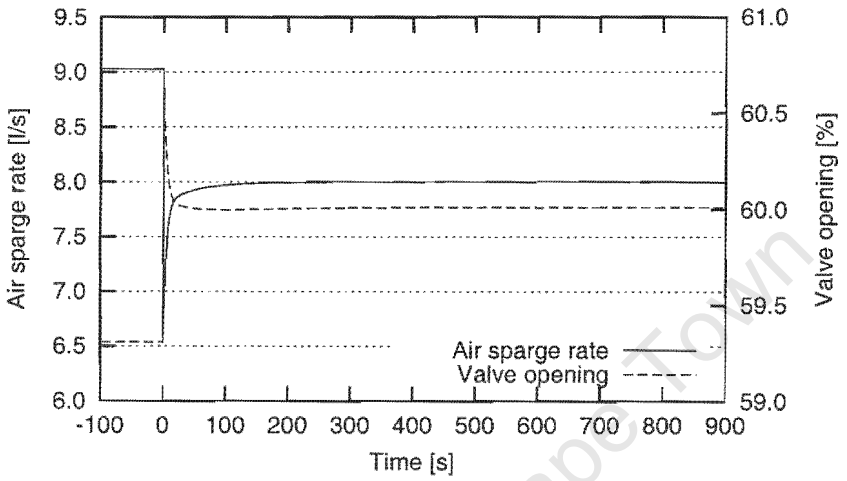

Figure 5.3: Test 1 - Grade setpoint change (input variables)

5.4.2 Test 2: System feed composition change (disturbance)

The system feed was modified by decreasing the concentration of platinum, and increasing the water flowrate. The controller aimed to keep the system at its nominal setpoints as shown in Table 5.1.

The platinum concentration was decreased by 20%, and the feed water increased by 10%. Without control, decreasing the platinum feed would result in a lowering of the concentrate grade. To counter this decrease the air sparge-rate must be decreased. By dropping the air sparge-rate, the effect of entrainment is decreased and the grade of the concentrate increases. However, recovery will decrease as a result of the change.

The increase in water primarily affects the tank level, causing it to rise. The valve opening must therefore increase to keep the level constant.

Figure 5.4 shows the effect on the output variables, with the expected decrease in grade and recovery and increase in tank level.

The grade setpoint is achieved very rapidly, with only a small deviation for a short time period. The input response (Figure 5.5) shows how the air rate is continually adjusted to counter the dynamics within the cell as the modified feed slowly alters the state within the cell.

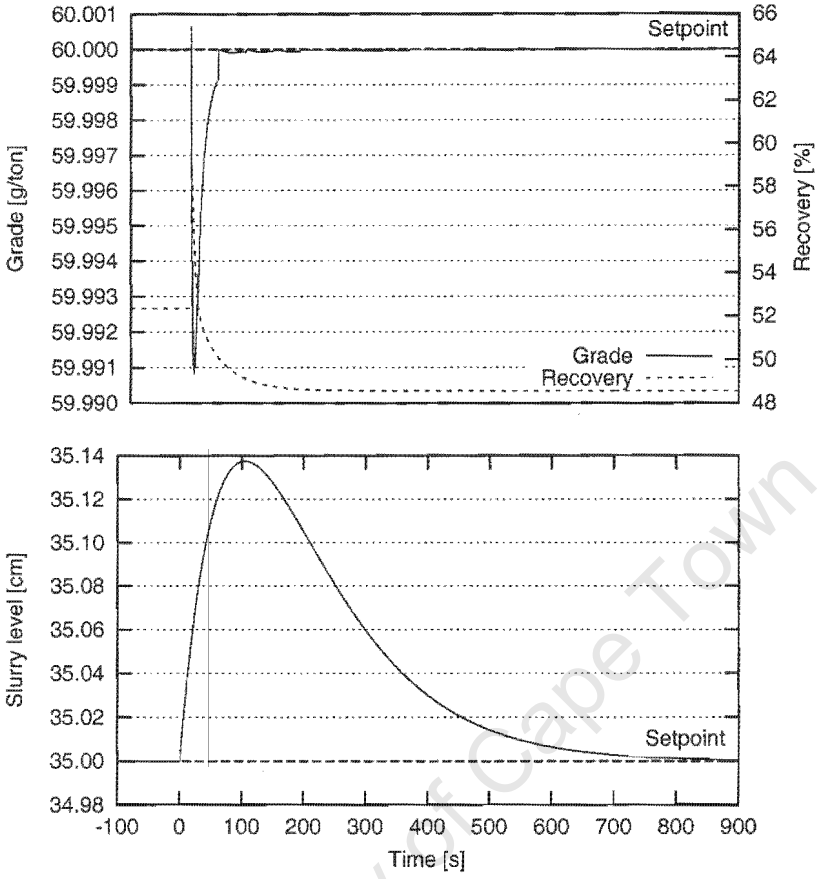

Figure 5.4: Test 2 - Feed change (output variables)

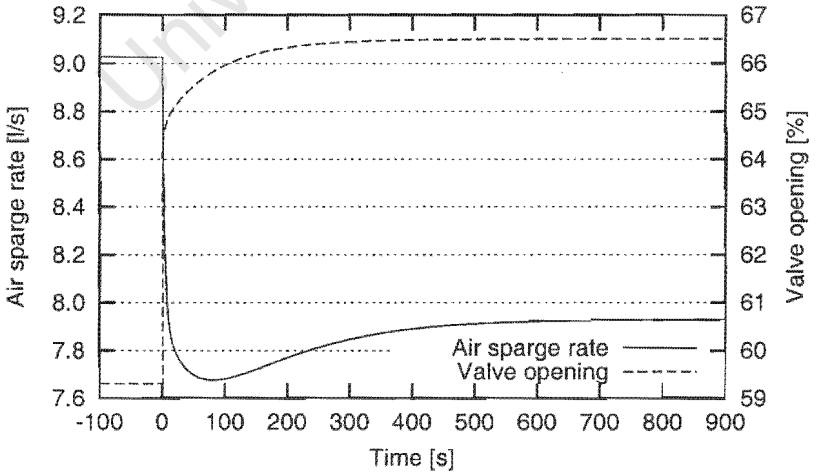

Figure 5.5: Test 2 - Feed change (input variables)

5.4.3 Test 3: Imperfect plant/model matching

As a simple test of robustness, a basic simulation with plant/model mismatch was performed. For this test the controller was exactly the same as all the other tests, but the plant was simulated using a model nonlinear in the controls. With all other tests the plant used the same model as the controller, which was linearised in the control variable (note, this was still nonlinear in the state variables resulting in a nonlinear model).

The controller aimed to keep the plant operating at the same nominal setpoints as shown in Table 5.1.

Figures 5.6 - 5.7 show the plant response represented by the solid lines, and the model upon which the controller is acting by the broken lines. For the concentrate grade, the controller manages to keep the model at the setpoint, but the actual plant response is not perfectly regulated. The sparge-rate fluctuates considerably to control the plant grade (Figure 5.8).

The slurry level shows better control (Figure 5.8), but this is mainly due to valve opening having a linear relationship in both plant and controller model. Nonetheless the fluctuations in sparge-rate cause some variation in level, which was accommodated by manipulation of the valve opening.

Despite the large fluctuations in the output variables, an examination of the magnitude of the variation shows that the deviations from the setpoints of both grade and level are small.

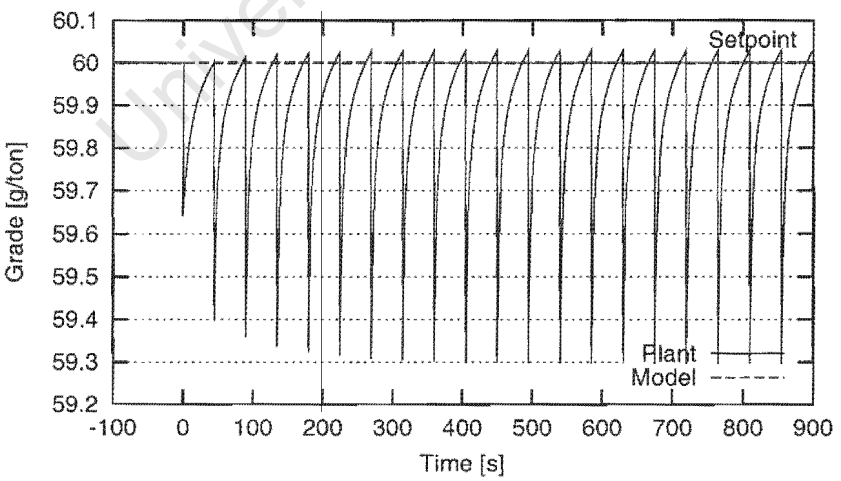

Figure 5.6: Test 3 - Plant/model mismatch (response of grade)

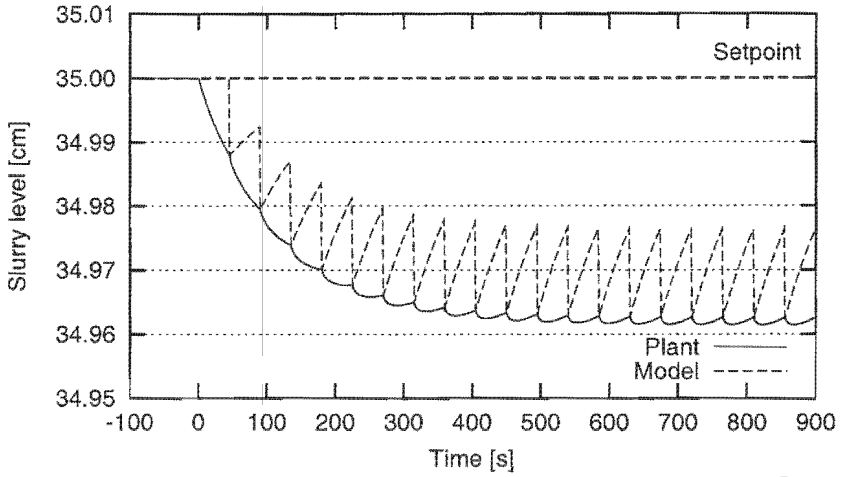

Figure 5.7: Test 3 - Plant/model mismatch (response of slurry level)

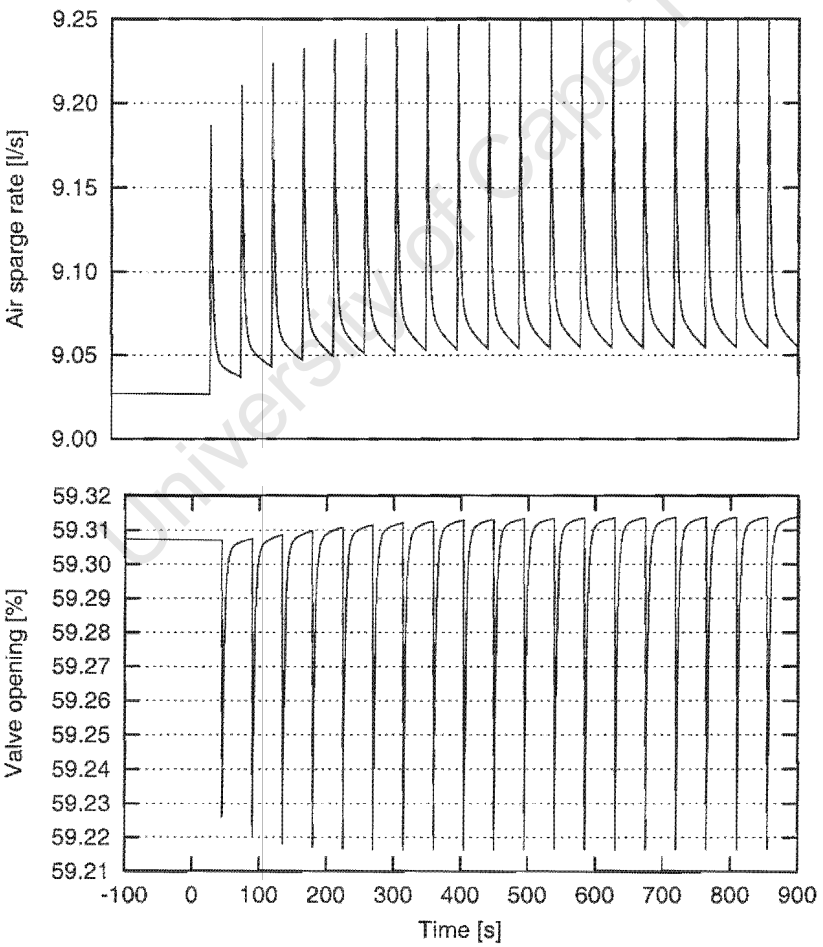

Figure 5.8: Test 3 - Plant/model mismatch (input variables)

5.4.4 Test 4: Comparison of linear and nonlinear MPC

To justify the use of nonlinear MPC on this system, a basic linear MPC controller (QDMC) was designed which attempted to control the nonlinear system. This controller was based on a linear model of the process, and used conventional piecewise constant control input moves. The test is exactly the same as Test 1, with a setpoint change in the concentrate grade from 60 [g/ton] to 70 [g/ton]. The controller attempted to perform this setpoint change while keeping the slurry level constant.

The linear MPC controller was designed using a step response model to identify the system. The model was identified by independent steps up and down of the control variables over the operating regime found by steady-state analysis of the nonlinear system. The objective function, weightings and input constraints were the same for both controllers, although the constraints were not found to be active for this test. Both controllers used a sampling period of 45 seconds, and 20 control moves were simulated resulting in a total simulation time of 900 seconds or 15 minutes.

When identifying the step-response model for the linear controller, it was found that it was vital that the model be identified over the anticipated region of operation of the system. If the regime over which the model was identified differed from the operating regime of the system being controlled, the QDMC controller performance ranged from poor to highly unstable.

As can be seen in Figure 5.9 both controllers are able to satisfy the setpoints, although some cycling occurs with the linear controller. As with Test 1, the nonlinear controller is able to almost instantly satisfy the grade setpoint, and rapidly restores the level to its nominal operating value. The linear controller does not invert the process perfectly, and the grade fluctuates around the setpoint before settling down to a new steady-state. It manages to keep the slurry level close to the setpoint, although it does not stabilise in the 900 second simulation.

Qualitatively the objective cost of the linear controller is far greater than the nonlinear controller, and its performance is dependent on whether it is operating in the region of linearisation or not. The advantage of using a nonlinear controller is the rapid achievement of setpoints, and stable system operation. Clearly the nonlinear controller performs considerably better than the linear MPC controller. Nonetheless the implementation of the nonlinear controller is far more complex and requires a greater degree of understanding of the process dynamics.

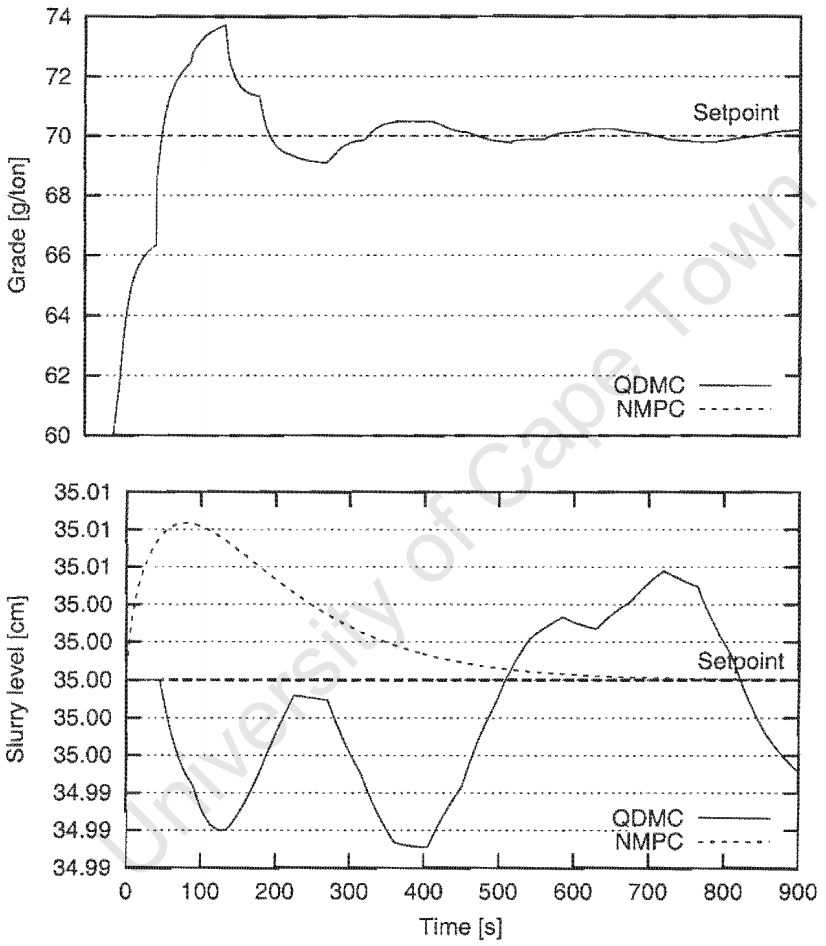

Figure 5.9: Test 4 - QDMC vs NMPC (output variables)

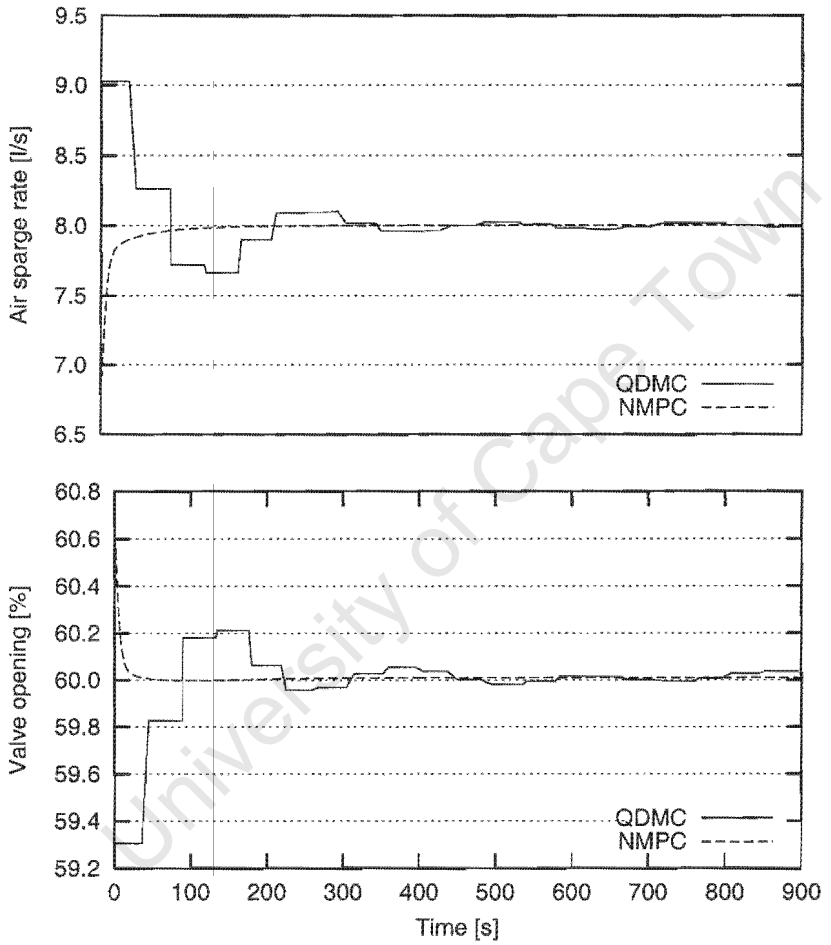

Figure 5.10: Test 4 - QDMC vs NMPC (input variables)

Chapter 6

Conclusions

The first part of this study focussed on the investigation of the use of orthogonal collocation for the indirect solution of optimal control problems. The FORTRAN package COLSYS was used for this purpose. Using the Euler-Lagrange equations, a boundary value problem was produced which could be solved using COLSYS. The method was then extended to a number of different classes of optimal control problem, allowing the inclusion of constraints and the solution of multi-input, multi-output problems. Problems with constraints on the input variables and problems with terminal state constraints were successfully solved.

The method is able to accurately solve optimal control problems, although a fair amount of problem transformation is required. The Hamiltonian function must be produced, and the derivatives with respect to the state, control and adjoint variables taken. COLSYS also requires a Jacobian of the differential and the boundary equations. The inclusion of input constraints increases the complexity of the formulation. Thus although this method is highly efficient and accurate in solving the boundary value problem (BVP), the system transformation to generate the BVP requires some simple problem analysis.

This method is particularly useful for problems affine (linear) in the control variable and with a quadratic objective function as this allows the input to be explicitly calculated. Further analysis of the method is required for the general case. Additional expansions of the method could include other classes of optimal control problems (state trajectory constraints, minimum time problems) and the implementation of improved initial guesses (e.g. hybrid methods).

The method of solving optimal control problems developed in Chapter 2 is capable of being used in a quasi-infinite horizon NMPC strategy (e.g. Chen and Allgöwer, (1998)). This

strategy requires the solution of optimal control problems with a terminal penalty function, and a terminal state inequality constraint. Under certain weak assumptions, this method is able to guarantee closed-loop asymptotic stability. The use of a continuously varying control input, as opposed to the conventional piecewise-constant control input resulted in a lower objective function cost for the case study examined. This was particularly apparent in the cost contribution of the input variable, allowing a more aggressive control to be used, or resulting in considerable savings on the input variable.

A dynamic flotation model was developed which showed qualitatively similar behaviour to real flotation processes. This model was developed by performing a mass balance over a flotation cell with an accumulation term included. Most flotation models are based on first-order rate kinetics, and thus in kinetic form may be used without modification of the rate constants to show rate of change of mass within the flotation cell. A water recovery model of similar form was used to predict the rate at which water is recovered to the concentrate stream. The entrainment model required an assumption that the dynamics of entrainment are linearly dependent on the water recovery. It was shown that the entrainability parameters could be used without modification if the concentrate water flowrate is considerably less than the tailings flowrate, although the entrainment equation needed slight modification. This only holds for those mineral classes primarily recovered by entrainment.

The model produced as a result of this analysis predicted the rate of change of mass within the cell. This rate of change is a function of the mass in the cell, and two input variables - tailings valve opening and air flowrate. A simplified model for use in an optimal control problem was developed by linearising the original model in the input variables. This model was still highly nonlinear in the state variables, but was affine in the control input. Comparing the nonlinear to the input-linearised model showed only small deviations in behaviour.

Finally, a successful simulation of the application of nonlinear model predictive control to a single, simplified flotation cell was performed. Tests included setpoint changes, control of plant disturbances, and control under plant/model mismatch. The controller attempted to optimally satisfy grade and level setpoints, while minimising flow fluctuations in the tailings stream. Grade was primarily controlled using the air sparge-rate, and slurry level with the tailings valve opening, although there was considerable implicit interaction between these two variables. The controller was able to rapidly satisfy both setpoints without offset. An additional test was performed to justify the use of nonlinear model predictive

control (NMPC). The performance of the NMPC controller was compared to a linear MPC controller (quadratic dynamic matrix control). It was shown that the nonlinear controller performed considerably better than the linear controller, with rapid achievement of the setpoints and stable operation.

The case study did not consider typical industrial difficulties such as state estimation and noisy signals, and thus this is a possible area for further research. The control of a larger system is also necessary before this strategy can be proposed for industrial implementation.

References

Allgöwer, F., Badgwell, T. A., Qin, J. S., Rawlings, J. B., and Wright, S. J. (1999). Nonlinear predictive control and moving horizon estimation - an introductory overview. In Frank, P. M., editor, *Advances in Control*, pages 391–449. Springer-Verlag, London.

Ascher, U., Christiansen, J., and Russell, R. D. (1981). Collocation software for boundary-value ODE's. *ACM Transactions on Mathematical Software*, 7(2):209–222.

Bauer, T., Betts, J. T., Hallman, W., Huffman, W. P., and Zondervan, K. (1984). Solving the optimal control problem using a nonlinear programming technique, parts i and ii. *AIAA Papers 84-2037 and 84-2038.*

Bell, M. L. and Sargent, R. W. H. (2000). Optimal control of inequality constrained DAE systems. *Computers and Chemical Engineering*, 24:2385–2404.

Betts, J. T. and Huffman, W. P. (1992). Application of sparse nonlinear programming to trajectory optimization. *Journal of Guidance, Control and Dynamics*, 15(1):198–206.

Bock, H. G. and Plitt, K. J. (1984). A multiple-shooting algorithm for the direct solution of optimal control problems. In *Proceedings of the 9^{th} IFAC World Congress*, Budapest. Pergamon Press.

Bulirsch, R., Nerz, E., Pesch, H. J., and von Stryk, O. (1993). Combining direct and indirect methods in optimal control: range maximization of a hang glider. In Bulirsch, R., Miele, A., Stoer, J., and Well, K. H., editors, *Optimal Control - Calculus of Variations, Optimal Control Theory and Numerical Methods, International Series of Numerical Mathematics*, pages 273–288. Basel: Birkhäuser.

Chen, H. and Allgöwer, F. (1998). A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10):1205–1217.

Cuthrell, J. E. and Biegler, L. T. (1987). On the optimization of differential-algebraic process systems. *AIChE J.*, 33:1257–1270.

Cuthrell, J. E. and Biegler, L. T. (1989). Simultaneous optimization and solution methods for batch reactor control profiles. *Computers chem. Engng*, 13(1/2):49–62.

Cutler, C. R. and Ramaker, B. L. (1980). Dynamic matrix control - a computer control algorithm. In *Proc. Joint Automatic Control Conference*, San Francisco, CA.

de Oliveira, S. L. (1996). *Model predictive control for constrained nonlinear systems.* PhD thesis, Swiss Federal Institute of Technology (ETH), Zürich, Switzerland.

Desbiens, A., Hodouin, D., and Mailloux, M. (1998). Nonlinear predictive control of a rougher flotation unit using local models. In *International Federation of Automatic Control (IFAC) Symposium on Automation in Mining, Mineral and Metal Processing*, pages 297–302, Cologne, Allemagne.

Doyle, J. C., Primbs, J., Shapiro, B., and Nevistić, V. (1996). Nonlinear games: examples and counterexamples. In *Proceedings of the 35th IEEE Conference on Decision and Control*, pages 3915–3920, Kobe, Japan.

Franzidis, J. P. and Manlapig, E. V. (1999). A new, comprehensive, and useful model for flotation. In Parekh, B. K. and Miller, J. D., editors, *Advances in Flotation Technology*, pages 413–423. Society for Mining, Metallurgy, and Exploration, Inc.

Garcia, C. E. (1984). Quadratic dynamic matrix control of nonlinear processes. An application to a batch reactor process. In *AIChE Annual Meeting*, San Francisco, California.

Garcia, C. E. and Morshedi, A. M. (1986). Quadratic programming solution of dynamic matrix control (QDMC). *Chem. Eng. Commun.*, 46:073–087.

Garcia, C. E., Prett, D. M., and Morari, M. (1989). Model predictive control: Theory and practice - a survey. *Automatica*, 25(3):335–348.

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. part 4: Effect of bubble surface area flux on flotation kinetics. *Minerals Engineering*, 10(4):367–379.

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

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

Harris, M. C. (1998). The use of flotation plant data to simulate flotation circuits. In *Minerals Processing Design School*, Technikon SA Conference Centre.

Harris, T. A. (2000). *The Development of a Flotation Simulation Methodology Towards an Optimisation Study of UG2 Platinum Flotation Circuits.* PhD thesis, University of Cape Town.

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

Herbst, J. A. and Pate, W. T. (1986). The power of model based control for mineral processing operations. In *Automation in Mining, Mineral and Metal Processing*, number xv+516 in 5th IFAC symposium, pages 17–33, Tokyo, Japan. Pergamon, Oxford, UK.

Kelley, H. J. (1960). Gradient theory of optimal flight paths. *J. Amer. Rocket Soc.*, 30.

Kirjavainen, V. M. (1992). Mathematical model for the entrainment of hydrophilic particles in froth flotation. *Int. J. Min. Process.*, 35:1–11.

Kirk, D. E. (1970). *Optimal Control Theory - An Introduction.* Prentice Hall, Englewood Cliffs.

Koslik, B. and Breitner, M. H. (1997). In optimal control problem in economics with four linear controls. *Journal of Optimization Theory and Applications*, 94(3):619–634.

Langson, W. S., Knights, B. D. H., and Swartz, C. L. E. (2000). Constrained control of dynamic systems using a hybrid implementation of model predictive control: Theoretical analysis and application to a continuous stirred tank reactor. In *AIChE Conference Proceedings, Unpublished*, Los Angeles. AIChE.

Logsdon, J. S. and Biegler, L. T. (1989). Accurate solution of differential-algebraic optimization problems. *Ind. Eng. Chem. Res.*, 28:1628–1639.

Lynch, A. J., Johnson, N. W., Manlapig, E. V., and Thorne, C. G. (1981). *Mineral and Coal Flotation Circuits - Their Simulation and Control*, volume 3 of *Developments in Minerals Processing Series*. Elsevier Scientific Publishing Company, Amsterdam.

Manlapig, E. V. (1977). *The Dynamic Behaviour and Automatic Control of a Chalcopyrite Flotation Plant*. PhD thesis, University of Queensland.

Mathe, Z. T., Harris, M. C., O'Connor, C. T., and Franzidis, J. P. (1998). Review of froth modelling in steady state systems. *Minerals Engineering*, 11(5):397–421.

Maurer, H. and Gillessen, W. (1975). Applicaton of multiple shooting to the numerical solution of optimal control problems with bounded state variables. *Computing*, 15:105–126.

Mayne, D. Q. (1996). Nonlinear model predictive control: An assessment. In Kantor, J. C., Garcia, C. E., and Carnahan, B., editors, *Fifth International Conference on Chemical Process Control - CPC V*, pages 217–231. American Institute of Chemical Engineers.

Mayne, D. Q. and Michalska, H. (1990). Receding horizon control of nonlinear systems. *IEEE Trans. Automat. Contr.*, 7(35):814–824.

Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Scokaert, P. O. M. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814.

McKee, D. J. (1991). Automatic flotation control - a review of 20 years of effort. *Minerals Engineering*, 4(7-11):653–666.

Meadows, E. S., Henson, M. A., Eaton, J. W., and Rawlings, J. B. (1995). Receding horizon control and discontinuous state feedback stabilization. *Int. J. Contr.*, 5(62):1217–1229.

Meadows, E. S. and Rawlings, J. B. (1997). Model predictive control. In Henson, M. A. and Seborg, D. E., editors, *Nonlinear Process Control*, chapter 5, pages 233–310. Prentice Hall PTR.

Michalska, H. and Mayne, D. (1993). Robust receeding horizon control of constrained nonlinear systems. *IEEE Trans. Aut. Control*, 38(11):1623–1633.

Miele, A. and Wang, T. (1993). Parallel computation of two-point boundary-value problems via particular solutions. *JOTA*, 79:5–29.

Miele, A., Well, K. H., and Tietze, J. L. (1973). Multipoint approach to the two-point boundary-value problem. *J. Math. Anal. Applic.*, 44:625–642.

Milot, M., Desbiens, A., del Villar, R., and Hodouin, D. (2000). Identification and multivariable nonlinear predictive control of a pilot flotation column. In *XXII International Mineral Processing Congress*, pages A3.120–A3.127, Rome, Italie.

Morari, M. (1994). Model predictive control: Multivariable control technique of choice in the 1990s? In *Advances in Model-Based Predictive Control*, pages 22–37. Oxford University Press.

Nevistic, V. (1997). *Constrained Control of Nonlinear Systems.* PhD thesis, ETH, Zurich.

Pérez-Correa, R., González, G., Casali, A., Cipriano, A., Barrera, R., and Zavaa, E. (1998). Dynamic modelling and advanced multivariable control of conventional flotation plants. *Minerals Engineering*, 11(4):333–346.

Pesch, H. J. (1991). Off-line and on-line computation of optimal trajectories in the aerospace field. Course in applied mathematics in the aerospace field, International School of Mathmatics, Ettore Majorana Centre for Scientific Culture.

Prince, K. J. (1996). Generalised predictive control - a study and application. Master's thesis, University of Cape Town.

Sargent, R. W. H. (2000). Optimal control. *Journal of Computational and Applied Mathematics*, 124(1-2):361–371.

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

Sistu, P. B. and Bequette, B. W. (1995). Model predictive control of processes with input multiplicities. *Chemical Engineering Science*, 50(6):921–936.

Stoer, J. and Bulirsch, R. (1980). *Introduction to Numerical Analysis.* Springer-Verlag, New York.

Suichies, M., Leroux, D., Dechert, C., and Trusiak, A. (2000). An implementation of generalized predictive control in a flotation plant. *Control Engineering Practice*, 8:319–325.

Teo, K. L. (1991). *A unified computational approach to optimal control problems.* Wiley.

Tsang, T. H., Himmelblau, D. M., and Edgar, T. F. (1975). Optimal control via collocation and nonlinear programming. *Int. J. Control*, 21:763–768.

van de Vusse, J. G. (1964). Plug-flow type reactor versus tank reactor. *Chemical Engineering Science*, 19:994–997.

Zheng, A. (1997). A computationally efficient nonlinear MPC algorithm. In *Proc. American Control Conf.*, pages 1623–1627, Albuquerque, NM.

Appendix A

Flotation System Parameters

System Description and parameters

Physical system parameters

Parameter	Value	Units
Cell length	0.300	$[m]$
Cell width	0.300	$[m]$
Height of launder lip	0.400	$[m]$
Valve constant (gK_v)	1.2e-6	
Impellor rotation rate	13.87	$[s^{-1}]$
Impellor radius	0.0850	$[m]$
Impellor height	0.0600	$[m]$
Impellor peripheral speed	7.41	$[ms^{-1}]$
Impellor aspect ratio	2.83	

Ore feed parameters

Parameter	Value	Units
Number of mineral classes	5	
Size fraction passing 80% (P_{80})	134.4	$[\mu m]$
Water feed rate	0.282	$[kgs^{-1}]$

Parameter\Species	PGM_1	PGM_2	PGM_3	Chromite	Gangue	Units
Density	16500	16500	16500	4650	3000	$[kgm^{-3}]$
Floatability	2.47e-3	2.55e-4	0	1.86e-4	1.60e-4	
Size distribution:						
$5.0\mu m$	33.4%	33.4%	33.4%	13.9%	17.1%	[%]
$17.5\mu m$	21.6%	21.6%	21.6%	11.0%	9.16%	[%]
$31.5\mu m$	7.40%	7.40%	7.40%	5.94%	5.22%	[%]
$45.5\mu m$	11.1%	11.1%	11.1%	11.9%	11.8%	[%]
$64.0\mu m$	10.8%	10.8%	10.8%	16.5%	14.7%	[%]
Feed rate solids	1.23e-7	2.15e-7	1.01e-7	0.0236	0.0955	$[kgs^{-1}]$

Flotation parameters

Parameter\Species	PGM_1	PGM_2	PGM_3	Chromite	Gangue	Units
Drainage parameter δ	1.26	1.26	1.26	0.635	1.26	
Entrainment parameter ξ	17.42	17.42	17.42	18.33	17.42	$[\mu m]$
True float. frac. ϕ	1	1	0	0.00675	0.05	

Parameter	Value
Froth recovery parameter β	0.07
Gas holdup ε_g	1
S_b constants $\{a, b, c, d, e\}$	{125; 0.514; 0.662; 0; 0.456}
Water recovery parameter χ	14.03
Water recovery parameter σ	0.08
Water recovery parameter Ω	3.00

Appendix B

Sample COLSYS Driver Program

This section includes an example of the FORTRAN code required to solve a simple optimal control problem. Case 1 from Section 2.3.1 is used to demostrate the method.


```
      PROGRAM OC6
C=====================================================================
C     APPLICATION OF COLSYS TO SOLUTION OF OPTIMAL CONTROL PROBLEM
C
C Problem generated by Ben Knights on 18/4/2000 using co-HJB method
C=====================================================================
      implicit double precision (a-h,o-z)
      external fsub, dfsub, gsub, dgsub
      dimension m(2), zeta(2), ipar(11), ltol(2), tol(2), fixpnt(1),
     &          ispace(1900), fspace(32600), z(2)
C
      data ipar/ 1, 0, 0, 2, 32600, 1900, -1, 0, 0, 0, 0/
      data ncomp/2/, m/1, 1/,  aleft/0.d0/, aright/1.d0/,
     &   zeta/0.d0, 1.d0/, ltol/1, 2/, tol/1.d-8, 1.d-8/
C Specify the number of pts to be shown in the output.
      nopts=101;
C
C
C     Solve problem
      call colsys(ncomp, m, aleft, aright, zeta, ipar, ltol,
     &            tol, fixpnt, ispace, fspace, iflag,
     &            fsub, dfsub, gsub, dgsub, guess)
C
```

```
C       Get ouput, and write to file
        open(unit=2, file='COLTEST.LIS', status='UNKNOWN')
        write(*,595) tol(1)
        write(2,*) '# IFLAG = ', iflag
595     format(1h ,' TOL = ', d12.6)
        do i=1, nopts
           t = (i-1)*(aright - aleft)/(nopts-1)
           call appsln(t, z, fspace, ispace)
           Gx = 5.d0*(1.d0 - z(1))*exp(z(1))
           U  = -0.5d0*z(2)*Gx
           write(*,600) t, z(1), z(2), u
           write(2,600) t, z(1), z(2), u
        enddo
        write(*,*) 'IFLAG = ', iflag
        close(unit=2)
600     format(1h ,1x,f7.4,4(2x,f12.8))
C
        stop
        end
C
C
C
C
        subroutine fsub(t, z, f)
        implicit real*8 (a-h,o-z)
        real*8 L
        dimension z(2), f(2)
C
        x=z(1)
        L=z(2)

        Fx  = 12.5d0*x*(1-x)**2*exp(2.d0*x) - 0.5d0*x
        Gx  = 5.d0*(1.d0-x)*exp(x)
        Ux  = -0.5d0*L*Gx

        dFx = 12.5d0*( (1-x)**2*exp(2.d0*x) - 2.d0*x*(1.d0-x)*exp(2.d0*x)
     &       + 2.d0*x*(1.d0-x)**2.d0*exp(2.d0*x)) - 0.5d0
        dGx = -5.d0*x*exp(x)
        dQx = 2.d0*x

        f(1) = Fx + Gx*Ux
        f(2) = -(dQx + L*(dFx + dGx*Ux))
C
        return
        end
```

```
C
C
C
C
      subroutine dfsub(t, z, df)
      implicit real*8 (a-h,o-z)
      dimension z(2), df(2,2)
      real*8 L
C
      x=Z(1)
      L=Z(2)
      Gx  = 5.d0*(1.d0-x)*exp(x)
      Ux  = -0.5d0*L*Gx

      dFx = 12.5d0*( (1.d0-x)**2*exp(2.d0*x) -
     &     2.d0*x*(1.d0-x)*exp(2.d0*x) +
     &     2.d0*x*(1.d0-x)**2*exp(2.d0*x)) - 0.5d0
      dGx = -5.d0*x*exp(x)
      dUx = -0.5d0*L*dGx

      dUl  = -0.5d0*Gx

      ddFx = 12.5d0*exp(2.d0*x)*((-4.d0-8.d0*x)*(1.d0-x) +
     &     4.d0*(1.d0+x)*(1.d0-x)**2 + 2.d0*x)
      ddGx = -5.d0*exp(x)*(x + 1.d0)
      ddQx = 2.d0

      df(1,1) = dFx + ddGx*Ux + Gx*dUx
      df(1,2) = Gx*dUl
C
      df(2,1) = -(ddQx +
     &     L*(ddFx + ddGx*Ux + dGx*dUx))
      df(2,2) = -((dFx + dGx*Ux) + L*dGx*dUl)
      return
      end
C
C
C
C
      subroutine gsub(i, z, g)
      implicit real*8 (a-h,o-z)
      dimension z(2)
c
      if(i .eq. 1) then
         g = z(1) - 1.d0
```

```
      else if (i. eq. 2) then
         g = z(2) - 2*z(1)
      end if
C
      return
      end
C
C
C
C
      subroutine dgsub(i, z, dg)
      implicit real*8 (a-h,o-z)
      dimension z(2), dg(2)
C
      if(i .eq. 1) then
         dg(1) = 1.d0
         dg(2) = 0.d0
      else if (i .eq. 2) then
         dg(1) = -2.d0
         dg(2) = 1.d0
      end if
C
      return
      end
```