An Optimizing Control Strategy for Milling Circuit Operation

by

Jarrod Hart

Submitted to the University of Cape Town in fulfillment of the requirements for the degree of Master of Science in Engineering

December 1997

Acknowledgments

I would like to extend my sincerest gratitude to Dr Chris Swartz for his supervision, especially for his amazing ability to keep things in perspective. I would also like to thank him for all the opportunities he has afforded me and the patience he has shown.

Thanks also to Roderick Ross who shared the lab with me for the duration of my studies. What I learnt from you cannot be simply expressed.

I also owe much gratitude to Rolf Poser who taught me everything I know about UNIX and who graciously rescued me from many a computing disaster.

The following people also served to make my time at UCT a happy one: Bill Randall, Craig Balfour, Nellie Ndili, Clifford Meyer, Caleb Hattingh, Paul Waugh, and Arno Oprins.

Thanks also to my co-habitants during this thesis: Allan Hanbury, Greg Torr, Tim Hawkins and Kerry-Anne Hare.

Special gratitude goes to my girlfriend Bronwyn Butcher for her unerring support throughout my studies.

Abstract

It has been recognized that significantly more benefit can be derived by operating processes at the economically optimal process conditions than by solely improving the regulatory control performance. This has been the motivation for an increased focus on optimizing control strategies where the setpoints are adjusted on-line in accordance with an economic criterion. Since plant conditions change, this adjustment needs to be made on a continual basis, but at a frequency lower than that at which inputs are manipulated at the regulatory control level.

In the operation of tumbling mills, the fractional filling of the mill with slurry has a marked effect on the efficiency of the breakage process within the mill. The selection of the setpoint for this variable is critical in the optimization of throughput and energy efficiency in a grinding circuit.

Unfortunately, the optimal value for this setpoint is a function of both circuit and ore characteristics, which means that it varies as the feed and plant characteristics change. This means that periodic re-evaluation of this optimal value needs to be performed in order to fully maximize circuit throughput.

The mechanism for seeking this optimum on-line forms the basis for this thesis.

Continuous evaluation of the function relating throughput to mill loading (if that were possible) would reveal the optimal setpoint for the load which could be regularly updated.

A recent strategy tackled this issue by assuming power draw from the mill motors is related to circuit throughput. With the aim to optimize throughput, power consumption can be used as a yardstick by which the load setpoint (setpoint for fractional filling in mill) can be selected. The load setpoint is simply adjusted periodically to optimize the power consumption and correspondingly, the circuit throughput.

The key proposal in this thesis is a strategy that employs a model of the plant to assist and improve the selection of the load setpoint.

The model uses throughput data to build a relationship between mill load, throughput and slurry characteristics. This relationship is designed to utilize plant data to modify itself online, supplying an accurate 'prediction' of the optimal setpoint at any particular time.

This strategy is then demonstrated using a computer simulation of a simple milling circuit. A comparison with previous control strategies (with process disturbances and

measurement noise simulated) shows potential in several circumstances. The final proposal utilizes a combination of old and new to take advantage of the benefits of both the previous 'power optimizing' strategy and the new model based optimizing control strategy.

Table of Contents

List of Figures

Chapter 1

Introduction

As the result of a increasingly competitive environment, there has been an increased focus on optimizing control strategies where controller setpoints are adjusted on-line in accordance with an economic criterion.

This thesis considers the case of a simple autogenous grinding circuit (see figure 1.1). The figure shows the components considered, including the mill, a sump, a pump and a hydrocyclone. A typical set of available measurements is also shown to demonstrate the tools available to the controller to allow monitoring and manipulation.

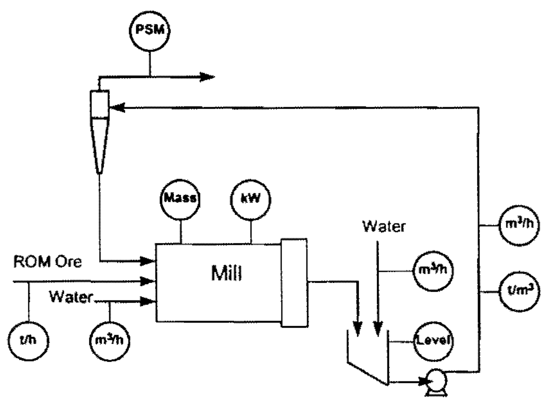


Figure 1-1 Typical autogenous grinding circuit, showing points where measurements may typically be made.

In such a circuit, run of mine (ROM) ore is fed via a conveyor into the mill. The ore is then tumbled within the mill until it passes through the exit grate, whereupon it enters the sump. Water is added to the sump to aid the hydrocyclone, which separates the particles based on size (and unfortunately, density), returning large particles to the mill for further treatment.

As a result of the large sizes and complex nature of these circuits, "dead times" as well as other nonlinearities, make the process difficult to model, and therefore difficult to control optimally. The process of comminution is also notoriously inefficient (as little as 5% of energy spent is used in the generation of new surface area) making it important to understand the processes involved, thereby paving the way to more effective control.

Improved control of milling circuits not only ensures a consistent product but may improve throughput and also reduce costs. In the case of autogenous grinding circuits, although power consumption is major cost, the most rewarding aspect to optimize is the circuit throughput. When comminution is the rate determining step, improving its throughput directly improves the daily production of the entire circuit.

Over the last 25 years much improvement has been made in the control of grinding circuits, starting with simple monitors which prevented mill overloads or sump overflows, and leading to multivariable control strategies that monitor several aspects of circuit operation simultaneously and calculate the corresponding input actions required to ensure accurate setpoint tracking (Hulbert *et al*, 1981).

Autogenous mills have the added difficulty that ROM material serves as the feed. As little prior processing is performed on this feed it exhibits extreme irregularities in size and content. The breakage required of the circuit therefore varies with time making accurate control more difficult. Besides the standard "random" type disturbances, there are slow permanent changes in the feed as new seams are mined and new characteristics emerge. Disturbances of this nature render control setpoints obsolete and new control strategies are required.

One such strategy is termed optimizing control. Optimizing controllers re-evaluate optimum setpoints for the multivariable controllers during operation. There are several variations in the implementation; one that has proved successful for autogenous grinding circuits is described by Craig *et al* (1992).

In their strategy, power draw from the mill motors is assumed to relate to circuit throughput. This premise is based on the assumption that a consistent fraction of each joule of energy is used up in the breakage of bonds in the ore, resulting in new surface area and reduced particle size.

With the aim to optimize throughput, power consumption can therefore be used as a yardstick by which the load setpoint (setpoint for fractional filling in mill) can be selected.

The throughput-load relationship exhibits a central peak as is shown in figure 1.2.

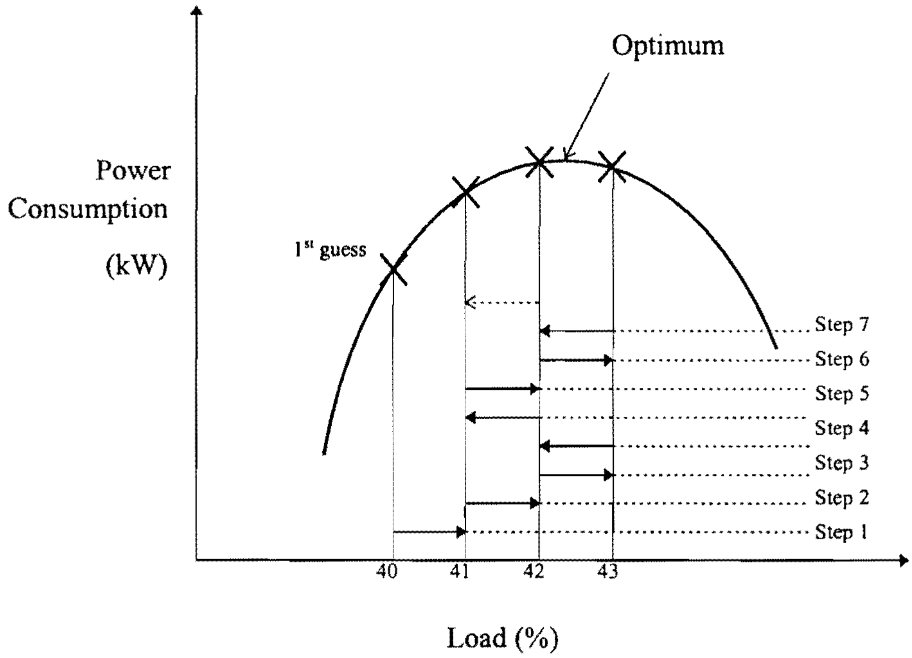


Figure 1-2 Power consumption vs. load relationship for a tumbling mill. Load setpoint changes are indicated.

The load setpoint is altered periodically to optimize the power consumption and correspondingly, the circuit throughput. If the load setpoint change results in an increase in power consumption, then the next step is made in the same direction and vice versa. This ensures that the optimum is tracked even if the underlying relationship should shift.

Craig *et al* report improvements of up to 10% in throughput which is a great success. It was the premise of this thesis that the strategy holds further potential; since the action of the optimizing controller is slow and is not completely accurate due to the assumption that power and throughput correspond directly and it also tends to oscillate about the optimum load setpoint. All of these factors lead to perceptible losses in throughput.

It was therefore the intention of this the thesis to examine the load setpoint with respect to throughput to allow the development of a control strategy that can more accurately estimate the optimum load setpoint.

The primary investigation looked at established grinding circuit models ranging from the breakage models of Bond (1962) to more advanced simulation models of Herbst (1980-1991). New models were also developed at this time.

Particular interest was paid to the effects of particle size, particle hardness, slurry density and slurry viscosity within the circuit, as these were established as representations or "monitors" for variations in circuit operation. Attempts were also made to model power consumption to develop a relationship between power consumption and circuit throughput.

Other models were examined and developed to relate throughput to system states such as mill load, load density, etc. These models allowed a broader understanding of the complexities of milling circuit operation.

The next step was to apply this understanding in the form of an intelligent control system. Several variations were tried but the strategy that allowed for both simple implementation and yielded promising results was eventually selected as the central candidate. The strategy was then refined and tested using a modified computer simulation program originally developed at MINTEK.

The proposed strategy is model-based, involving the extraction of a 3-dimensional surface relating throughput to load setpoint and average slurry viscosity within the mill. The surface is designed to reflect the throughput history of a circuit with respect to load setpoints and measured viscosities. The 3-D surface is then used as a plant model to assist in selection of the load setpoint.

The benefits of using a model are (i) no steady state requirement, allowing a higher frequency (ii) no oscillation and (iii) is more accurate. These factors ensure faster tracking of the optimal setpoint and therefore, improved throughput.

Chapter 2 provides the general background and theory, while a proposed control strategy is presented in Chapter 3.

The strategy is further refined by means of a strategy for on-line adaptation of the 3-D surface to improve the accuracy while removing the need to make an accurately initialize the 3-D surface. It also creates possibilities for a combined control strategy, where other optimizing strategies can be utilized during startup or during periods of extreme change, to allow the model to adapt to new conditions.

Simulation tests are used to demonstrate the strategy in operation. Chapter 3 concludes with a step by step example of the control process for a single iteration of the controller.

Chapter 4 demonstrates the performance of the strategy over an extended period, with repeated setpoint evaluations. Tests are performed to allow comparison with previous optimizing control strategies with regards to the quality of these proposed load setpoints. Further tests are naturally required (and recommended) to determine the full practicality and efficiency of the strategy when applied to a real milling circuit.

After the testing is described in Chapter 4, other possibilities for improved control of grinding circuits are discussed. Conclusions are presented in Chapter 5.

Chapter 2

Background

This chapter is designed to provide information on several of the tools and strategies referred to and utilized during the completion of this thesis.

Before it was possible to tackle the subject of optimizing control it was necessary to develop the subject of process control further. As a simulator was to be developed to assist in the derivation and testing of the proposed control strategy, information on some aspects of milling circuit modelling and simulation was also required.

This chapter is divided in to nine sections:

- *Sections 1-5* Control
- *Section 6* Optimizing Control
- *Section 7* Parameter Estimation
- *Section 8* On-line Optimization
- *Section 9* Grinding Circuit Theory

Several of the topics described here are referred to directly in later chapters while others provide background information, used to provide a setting for the thesis. At the end of some sections, mention is made of where that theory was applied during the thesis.

2.1 Process Control

The initial objective of the control system and control loops selected for a process are to provide stabilizing control of the major process variables so that physical, economic and management objectives can be met (Hales & Vanderbeek,1988).

Contrary to what we may expect, automatic control devices were in use as early as 250 BC by the Greeks who used water level controllers very similar to the level regulators in modern day flush toilets (Mayr, 1970). Early steam engines incorporated the fly-ball governor which was instrumental in the success of steam power (Seborg,1989).

It was in the 1930's that 'three mode' controllers, proportional, integral and derivative (PID) became commercially available (Ziegler, 1975). The first papers were published in the same period (Grebe *et al*, 1933, Ivanoff, 1934). Pneumatic PID controllers gained acceptance during the 1940's while their electronic counterparts became available in the 1950's. In the late 1950's and the 1960's computers were first applied to process control which paved the way for still greater improvements.

2.2 Feedback Control

An open loop controller is the simplest form of control; control actions are made with no knowledge of system states "downstream" of the manipulated variable. This is however of limited application and when downstream information is available it often possible to utilize this information beneficially, by means of feedback control.

One may wish to maintain a certain level within a tank which is fed from one side and emptied through a valve at the other. To do this, the level must be monitored and the valve set accordingly to return the level to the desired value. This forms the basis of feedback, or closed loop control. Information is passed from the system back to the controller.

There are several basic components in a feedback control loop

- the *process* (reaction vessel, heat exchanger, etc.)
- the *sensor/transmitter* (thermocouple, pressure sensor, etc.)
- the *controller*
- the *final control element* (valve, heater, etc.)
- and the *transmission lines* (electric wires, pneumatic pipes, etc.)

It is common to model each component with a relation between input and outputs. Process engineers commonly use *transfer functions* which examine the output/input relationship in the s domain (using the Laplace transform). Transfer functions are flexible models for dynamic systems since several models can be combined very simply while complex integration is avoided.

Block diagrams are then used to represent the control system. Each block contains the function that relates the output to the input. This allows relationships between any two variables to be calculated by simply adding or multiplying the transfer functions that lie on the adjoining path.

A simple feedback structure is shown in figure 2.1. Here the output is measured (G_m relates the output to the measured signal). The measured signal is subtracted from the desired value (the setpoint) via K_m which translates the setpoint into equivalent units as the measured signal.

The difference thus acquired (the "error") is passed through the controller, G_c, which determines the control action required which G_v then translates to an action (e.g. open a valve), which alters the manipulated variable. The system is thus modified to return the controlled variable to the desired value.

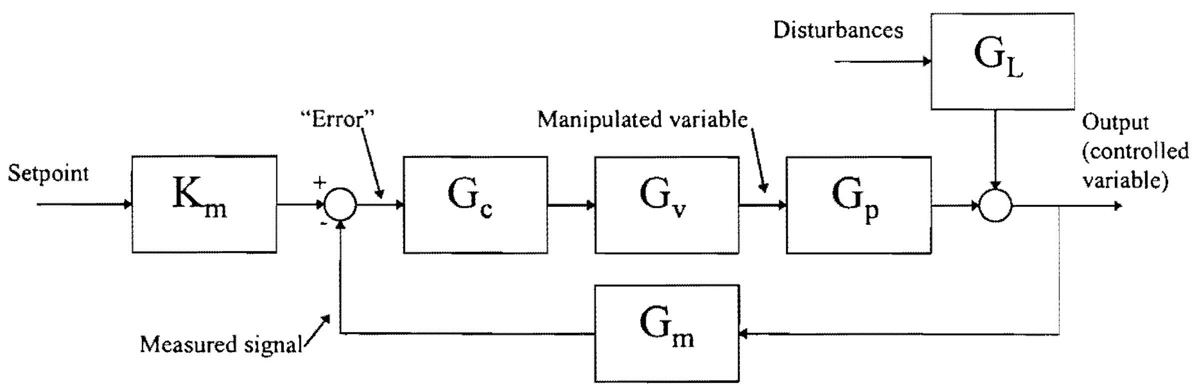


Figure 2-1 Standard block diagram of a feedback (closed-loop) control system. Each block contains transfer functions relating outputs to inputs (Seborg *et al*, 1989).

The controller is now in a position to perform two tasks; the first may be to negate the effects of disturbances (the regulator problem) while second is to perform setpoint tracking (the servo problem).

The Modes of Control

In feedback control the aim is to therefore to reduce the error signal, $e(t)$, to zero where

$$e(t) = R(t) - B(t) \tag{2.1}$$

where $R(t)$ = set point

$B(t)$ = measured value of the controlled variable

A simple linear relationship between the measured error and the control action is termed "proportional" control. The controller output is given by:

$$p(t) = \bar{p} + K_c e(t) \tag{2.2}$$

where $p(t)$ = the controller output

$\bar{p}$ = bias value

K_C = controller gain

The key is that the gain can be adjusted to determine the sensitivity of the output to the deviation of the controlled variable from the setpoint. High sensitivity may reduce the offset (residual error) but may result in excessive input action; it may even saturate or become unstable. The sign of the gain K determines the direction of the corrective action.

This type is control may be sufficient for some systems where high accuracy is not essential, but when accurate setpoint tracking will result in significant economic gain, offset may not be tolerable. The controller may then use a time integral of the error to make a further adjust to the control action. In this way, persistent error can be countered. This is termed proportional-integral control and is equivalent to adjusting the bias ($\bar{p}$). The controller output is now given by:

$$p(t) = \bar{p} + K_c \left[e(t) + \frac{1}{\tau_I} \int_0^t e(t^*) dt^* \right] \tag{2.3}$$

where τ_I is referred to as the integral time or the reset time and can be varied to improve system response. Integral control is also referred to as *reset* or *floating control.*

It is also possible to use the rate of change of the output to determine an improved control action. This is termed derivative control, and can be used in conjunction with proportional and integral actions. A PID controller would have the form:

$$p(t) = \bar{p} + K_c \left[e(t) + \frac{1}{\tau_I} \int_0^t e(t^*) dt^* + \tau_D \frac{de}{dt} \right] \tag{2.4}$$

where τ_D is the derivative time and can also be varied to suit the system.

These controllers are typically designed (Kc, τ_I and τ_D are chosen) by modelling the system (via step tests or frequency response methods) and then following well documented procedures (Seborg 1989) to determine controller parameters that result in the fastest possible response while still maintaining a sufficient margin of stability and preventing excessive control action.

In grinding circuit control, PI (proportional-integral) and PID (proportional-integral-derivative) feedback controllers have been used to regulate the sump level, the mill load, the pressure in the hydrocyclone feed stream and even the particle size in the product stream.

2.3 Specialized Control Strategies

Many variations on standard feedback control exist, such as cascade control, feedforward control and adaptive control. These methods exploit the structure of the process to allow either faster responses or better stability.

It is possible to model many processes extremely well using combinations of Newtonian mechanics, thermodynamics etc. With the advent of computers even stochastic modelling techniques have successfully been applied to complex systems. Model predictive control exploits these models to further improve the quality of control possible.

2.4 Multivariable Control

When a system has more than one output that requires control, the designer has to consider several possibilities. If the outputs are largely independent of one another and each can be manipulated using independent input actions, then completely separate feedback controllers may suffice. Some interaction can even be tolerated.

When interaction is greater, it may be necessary to decouple the system. This can be done by acting on each system to negate the effects of the respective control actions on the other outputs.

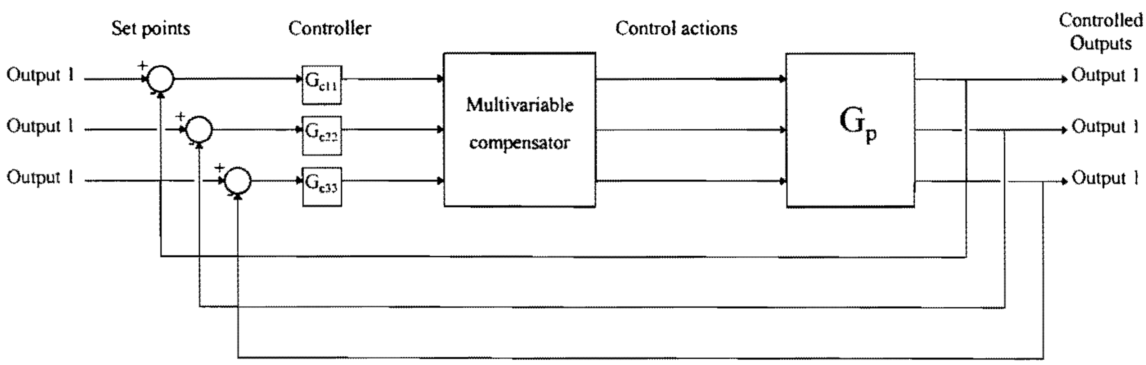


Figure 2-2 Structure of a 3x3 multivariable controller.

A third possibility is to design a truly multivariable controller at the outset. In this case the parameters used in the feedback loops are chosen via a complex design process to ensure both stability and improved control (Seborg, 1989, Skogestad, 1996). The structure of a 3x3 multivariable controller is shown in figure 2.2.

Grinding circuits exhibit a high degree of interaction between control loops and as a result, many new grinding circuits are now controlled with 3 by 3 multivariable controllers. The mill load is typically manipulated using the solids feed rate, while the PSD (particle size distribution) may be controlled by manipulating either the sump addition water or the pressure in the hydrocyclone feed stream. The third controlled state is the sump level, which may be manipulated using the pump speed or the water addition rate.

The last two sets are highly interactive and require a thorough multivariable design to ensure optimal operation with robustness. There are several ways of identifying parameters for multivariable controllers; some rely upon the Nyquist array which describes the interactions between each input and each output. The design technique is to select controller parameters that reduce off-diagonal elements of the array and hence reduce interaction between the individual control loops.

2.4.1 Multivariable Controller Design

Multivariable controllers aim to reduce interaction between individual control loops. Simple controllers merely decouple the loops with feedforward arrangements while true multivariable controllers actually select controller gains and reset times that minimize the interaction. Nyquist Array techniques such as the INA (Inverse Nyquist Array) and the DNA (Direct Nyquist Array) are popular and have been applied to milling circuits in the past (Braae, 1994). Other formulations include the LQG (linear quadratic Guassian) and H_∞ Optimal Control. INA was chosen for this project as it has proven successful for milling circuits (Hulbert *et al*, 1983, Craig *et al*, 1992).

A brief a description of the INA design technique follows (Maciejowski, 1989). Open loop tests are performed to determine transfer functions relating each output to each of the inputs. Inverse Nyquist plots of each of these are placed in an array (the INA). The INA can therefore be used to provide a graphic representation of the open loop response of the combined controller and plant model. The rows of the INA correspond to the setpoints while the columns correspond to outputs. The aim is therefore to reduce the effects of each setpoint on the other outputs (i.e. reduce the off-diagonal elements or obtain *diagonal dominance*). The effect of off-diagonal elements is represented on the diagonal plots in the form of Gershgorin bands. A pre-compensator is then designed to ensure diagonal dominance. Diagonal dominance is achieved when Gershgorin bands no longer enclose the origin.

Rosenbrock (1974) recommends the technique of pseudo-diagonalization for the construction of a pre-compensator. It is however possible to write a computer program to solve for the dynamic compensator by a combination of Guassian elimination and the optimum scaling of rows and columns to maximize diagonal dominance (Hulbert *et al*, 1983).

A more detailed description of the INA technique is given by Rosenbrock (1974). The underlying theory is further described by Rosenbrock and Storey (1970).

2.5 Supervisory Control

A supervisory controller uses process and economic models of the plant to optimize plant operation by maximizing an objective function (which may be based on profit, throughput, etc.). A computer program monitors the process and periodically computes the setpoints which would result in the greatest value for the objective function. These improved conditions are then implemented. A schematic is shown in figure 2.3

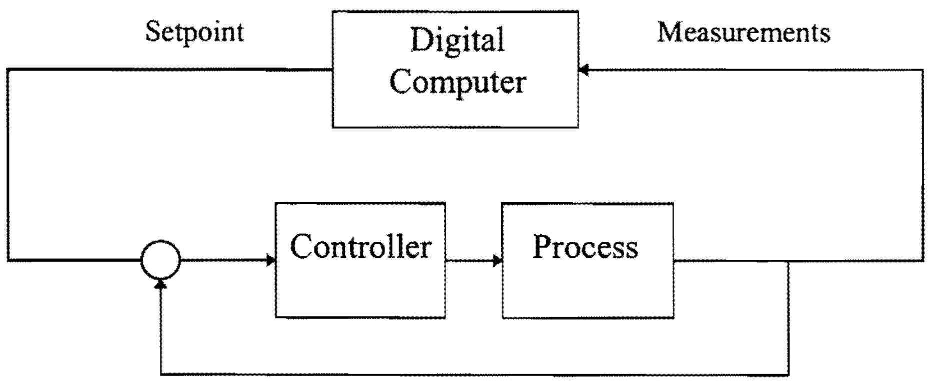


Figure 2-3 Supervisory control framework.

The model of the process used is typically a steady-state model based on either fundamental knowledge of the process or on experimental data, while the economic model relates the costs of raw materials, products and day to day operation. The optimization task faced by the controller may involve anything from a set of linear equations to complex nonlinear mixed integer (MINLP) type optimization.

Supervisory control is well suited to processes where a parameter requires optimization. In the case of autogenous grinding circuits, it is common to define an objective of this nature, for example, it may be desired to optimize the mass throughput of the circuit. This optimization is to be performed while conforming to the required product quality.

2.6 Optimizing Control

Optimizing control is a specific type of supervisory control concerned with the optimization based supervisory control problems described above (Lee & Weekman 1976, Garcia & Morari 1981, 1984, Chen & Joseph 1987, Darby & White 1988). In the case of grinding circuits, where the aim is to maximize throughput, optimizing control has found special application. Optimizing control defines the type of control strategy developed in this thesis.

Modern milling circuits feature many of the controller types mentioned in previous sections: open-loop control, feedback control (PI and PID) as well as multivariable control. In the case of optimizing control there are therefore several setpoints available for optimization. This thesis considers only the mill load setpoint in the development of an optimizing control strategy. Other setpoints (such as the product particle size) are identified in Chapter 4 as possible areas for further development

Craig *et al* (1992), applied optimizing control to a grinding circuit. Their technique is described in the next section.

2.6.1 Optimizing Control of Grinding Circuits - Craig *et al*

In their 1992 paper, Craig *et al* described a strategy for optimizing throughput in an autogenous grinding circuit. The circuit was as shown in figure 1.1 in the first chapter. A 3 by 3 multivariable controller was first designed and tested. Many tests were performed to determine the effect of mill load on both mill power draw and circuit throughput.

It was noticed that there was a distinct correlation between the power draw and the load setpoint, and a curve was put to the relationship. It was however also noted that the peak of this curve varied as conditions within the circuit varied. It was also observed that there was a strong linear relationship between power draw and circuit throughput, which meant the desired operation point would be at (or near) the peak of the power-load curve.

An on-line "hill-climbing" optimization strategy was therefore applied to keep the load setpoint near the peak. This strategy is described next.

Hill Climbing Technique

The first step is to select a starting load setpoint. After a set period of operation the power consumption is measured. The setpoint is then changed by a predefined amount (from say 40 to 41%). After the second period the power is measured again and this is compared with the previous value. If the consumption has increased, then it is deemed that the previous load setpoint change was "successful" and another step is made in the same direction (to 42%).

If, however, the power consumption decreases the setpoint change is deemed unsuccessful and the following setpoint change is made in the opposite direction (Back to 40%). These comparisons are then made periodically and the setpoint adjusted accordingly.

If it were true that the power-load curve were unchanging, then this algorithm would climb to the top and oscillate about the optimum as shown in figure 2.4.

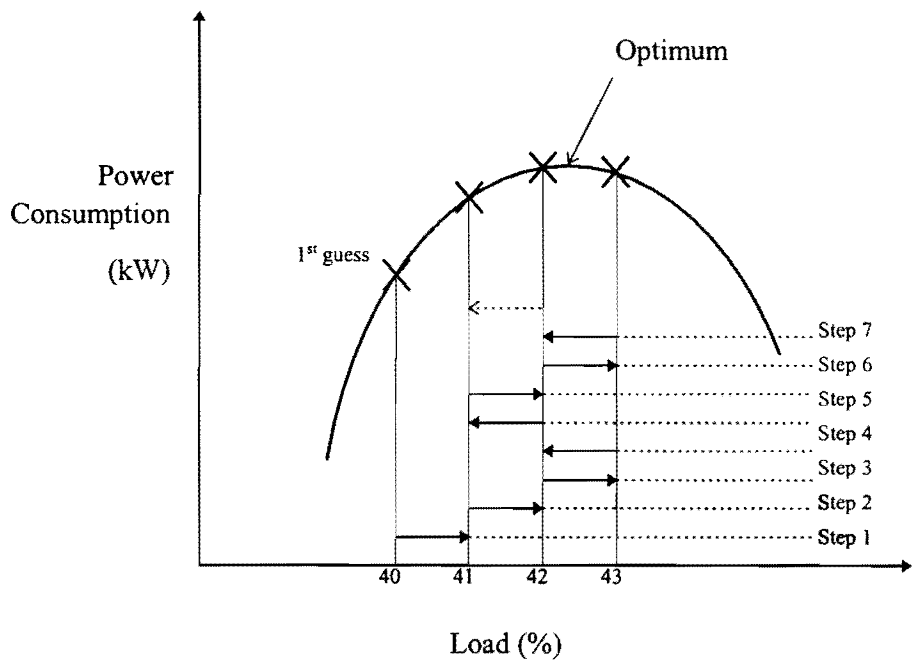


Figure 2-4 Peak searching algorithm. The first guess is made. The second guess is then made and the power consumption is seen to increase. The second alteration is therefore performed in the same direction - 42%. Again the power draw is seen to increase, so 43 is chosen, whereupon the power consumption decreases, so the setpoint is returned to 42. This continues, resulting in oscillation about the optimum.

This strategy will still find the optimum even if the curve were to shift or even change shape. This method caters for slow parameter changes (such a feed characteristics, spigot wear, etc.) since the optimal value will be tracked as it varies. The choice of step size will influence how fast a change in the curve can be followed, but will restrict the resolution of the result (i.e. the algorithm will oscillate further from the optimum).

2.7 Parameter Estimation

In order for a supervisory controller to make decisions based on the current conditions on the plant, information needs to flow from the plant to the controller. This is usually by means of some sort of measuring element that translates a physical quantity into a pressure or electrical impulse.

When the value of the measurement is stationary or changing slowly, then the signal being received by the controller is meaningful and representative of the truth (systematic errors aside). However, when the value one is trying to read varies during measurement as is often the case, then the controller requires a method of extracting a single number that is 'representative' of the value of the state.

The science of determining useful values (usually parameters in a model of the plant in question) from noisy or varying signals is called parameter estimation♦.

A simple example of parameter estimation is the running average. If a stream of noisy measurements is received, it is possible to reduce the effect of noise by generating an output stream where each number represents the average of the previous n incoming numbers. When n is 1, the original stream is returned, but as n is increased an increasingly smoothed dataset is returned. The art is in selecting n such that noise is removed (*filtered*) without hiding genuine trends in the data stream.

2.7.1 The Kalman Filter

A Kalman filter is a popular tool for estimating information that may not be directly accessible.

It is often desired to estimate the state of a system given a set of measurements taken over an interval of time. The state of the system refers to a set of variables that describe the inherent properties of the system at a specific instant of time. Kalman filtering is a useful technique that may be used for estimating or updating the previous estimate of a system's state by: (1) using indirect measurements of the state variables, and (2) using the covariance information of both the state variables and the indirect measurements. Intuitively, the idea is to use information about how measurements of a particular aspect of a system are correlated to the actual state of the system.

To illustrate this concept (adapted from Pichumani *et al*, 1994), suppose we know that the average height and weight for males in S.A. is 178 cm and 71 kg respectively. That means that in the absence of any other information, a randomly selected group of

♦ More can be read in "Optimal Estimation for Engineers", Eduard Eitelberg (1991).

males will most likely have a mean height of 178 cm and weight of 71 kg. But suppose we were told that the height of one of the males is 150 cm. How would the estimate of his weight change? Suppose we also knew that the length of his forearm is 25 cm. Would that provide an even better estimate of his weight? The answer is "yes" if we know how the various measurements are correlated to one another. In human anatomy, we can exploit the correlations among various anthropometric quantities to make estimates using limited data. Kalman filtering is one such technique for using the correlation information to derive better estimates of unknown quantities.

Pichumani *et al* (1994) provide discussion on several aspects of the filter:

The Linear Dynamic Model

The Kalman filter was originally developed to estimate time varying quantities (state variables) from a set of noisy measurements (signals). The set of state variables is usually represented by a vector denoted as $\mathbf{x}(t)$. For example, we could define the state of a satellite to be its position in Cartesian coordinates relative to the center of the earth. In this case, the state vector would look like the following:

$$\mathbf{x}(t) = [x \quad y \quad z]^T \qquad (2.5)$$

where the $(x\ y\ z)$ triplet represents the time-varying coordinates of the satellite's position (note the distinction between **x**, the state vector, and x the x component of the state vector) . In general, it is not possible to determine through direct measurements the precise state of such a system. Instead, we must rely on measurements that provide information about the state variables. In this example, we may have to rely on external radar signals or telemetry signals sent by the satellite itself. In other words, we must infer the state of the system based on the set of measurements we have about the system. Of course, in an ideal world, the state of the system can be determined precisely given an appropriate set of measurements. In practice, this is difficult because these measurements are corrupted by noise. In this example, if we had an onboard instrument that could measure the satellite's (x,y,z) coordinates using a gyroscopic system, the relationship between the state vector could be written as,

$$\mathbf{z}(t) = \mathbf{Hx} + \mathbf{v} \qquad (2.6)$$

where **z** is the measured position vector and $\boldsymbol{v}$ is the noise vector associated with the measurement. The matrix **H** is referred to as the transfer function because it transfers the input, which is the actual state, to the output, which is the set of measurements. Equation 2.6 expresses the measurement vector as a function of the state vector. What we really need is the state vector expressed as a function of the measurement vector. If we know the value of **v**, this would reduce to a simple matrix inversion:

$$\mathbf{x} = \mathbf{H}^{-1}(\mathbf{z} - \mathbf{v}) \tag{2.7}$$

However, the addition of the unknown **v** term implies that we can no longer determine **z** precisely. Instead, we must find an estimate of **z** based on our knowledge of how the state variables are correlated to one another, and on the prior distribution of **v**. A Kalman filter is a function that provides an optimal estimate of the state vector **x** given such knowledge. This is an example of a linear dynamic system because the input and output are described by a linear equation.

The Nonlinear Dynamic Model

Suppose the position of the satellite were measured by a ground-based tracking station which outputs the position in spherical coordinates relative the station's tracking antenna (spherical coordinates are specified by a distance from an origin ρ and two angles of rotation ϕ and θ). This would be an example of a nonlinear dynamic system because the satellite's state vector (which is represented in Cartesian coordinates) is no longer a linear function of the measurement vector. This relationship can now be expressed as,

$$\mathbf{z}(t) = \mathbf{H}(\mathbf{x}) + \mathbf{v} \tag{2.8}$$

where,

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} \dfrac{\arccos z}{\sqrt{x^2 + y^2 + z^2}} \\ \arctan \frac{y}{x} \\ \sqrt{x^2 + y^2 + z^2} \end{bmatrix} \tag{2.9}$$

H is now a general vector valued transfer function of **x**. In both the linear and non-linear cases, the noise is assumed to be a property of the *measurement* process and not a property of the system's state vector.

Without delving into the mathematical derivation of the Kalman filter, a more general description is now presented.

Definition and Example Solution

Consider the time-varying discrete time system:

$$\mathbf{x}_{i+1} = \mathbf{A}_i\mathbf{x}_i + \mathbf{B}_i\mathbf{u}_i + \mathbf{v}_i \tag{2.10}$$

$$\mathbf{y}_i = \mathbf{C}_i\mathbf{x}_i + \mathbf{D}_i\mathbf{u}_i + \mathbf{w}_i \tag{2.11}$$

Where v_i and w_i are zero mean white state and measurement noise respectively. Their covariance (correlation) matrices are

$$E\{\mathbf{v}_i\mathbf{v}_j^T\} = \mathbf{Q}_i\delta_{ij} \tag{2.12}$$

$$E\{\mathbf{w}_i\mathbf{w}_j^T\} = \mathbf{R}_i\delta_{ij} \tag{2.13}$$

respectively.

It is assumed that the initial condition $\mathbf{x}_0$ is a random variable with a given mean

$$E\{\mathbf{x}_0\} = \hat{\mathbf{x}}_0 \tag{2.14}$$

and with a given covariance

$$E\{(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T\} = \mathbf{P}_0 \tag{2.15}$$

Now the optimal filtering problem can be worded:

Find a linear, unbiased estimate $\hat{\mathbf{x}}_{i+1}$ of $\mathbf{x}_{i+1}$ based on the knowledge of the values of u_k, y_k for $k = 0,\ldots,i$, such that the mean square error is minimized:

$$E\{(\hat{\mathbf{x}}_{i+1} - \mathbf{x}_{i+1})^T(\hat{\mathbf{x}}_{i+1} - \mathbf{x}_{i+1})\} \Rightarrow \min \tag{2.16}$$

The complete Kalman filter is given as (from Pichumani *et al*, 1994):

$$\begin{aligned} \text{gain:} \quad & \mathbf{K}_i = \mathbf{A}_i\mathbf{P}_i\mathbf{C}_i^T(\mathbf{R}_i + \mathbf{C}_i\mathbf{P}_i\mathbf{C}_i^T)^{-1} \\ \text{estimate:} \quad & \hat{\mathbf{x}}_{i+1} = \mathbf{A}_i\hat{\mathbf{x}}_i + \mathbf{B}_i\mathbf{u}_i + \mathbf{K}_i[\mathbf{y}_i - \mathbf{D}_i\mathbf{u}_i - \mathbf{C}_i\mathbf{x}_i] \\ \text{error covariance:} \quad & \mathbf{P}_{i+1} = \mathbf{A}_i\mathbf{P}_i\mathbf{A}_i^T - \mathbf{K}_i\mathbf{C}_i\mathbf{P}_i\mathbf{A}_i^T + \mathbf{Q}_i \\ & \text{with} \quad \hat{\mathbf{x}}_0 = E\{\mathbf{x}_0\}, \quad \mathbf{P}_0 = E\{(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T\} \\ & \text{for} \quad i = 0,1,\ldots \end{aligned} \tag{2.17}$$

In fact, this is called the one-step-ahead prediction algorithm, because the measurements, all made at i are used to predict $\mathbf{x}$ at $i + 1$.

This prediction algorithm is used to estimate plant throughput and mill load in Chapter 3. More information on Kalman filters can be obtained from Kalman (1960) Maybeck (1979) and Eitelberg (1991).

2.8 On-line Optimization

Before delving into the subject of milling circuits, it is instructive to examine several aspects and philosophies of on-line optimization and how work in this field pertains to the optimizing control strategies in this thesis.

2.8.1 Requirements

To ensure successful implementation, an on-line optimization scheme requires the following (Young, 1997):

- The strategy should account for effects of uncertain and unmeasureable disturbances and be robust to nonlinearities in plant behaviour.
- Convergence to the economic optimum should be faster than the period of the disturbances of major economic impact.
- Noisy measurements and measurement errors must be handled.
- Any process models used must be robust and solve quickly.
- Process operation should move from one operating point to the next without violating any constraints.

2.8.2 Methods of Locating the Optimum

Young (1997), divides these methods into two types, *on-line* methods and *off-line* methods. A summary of his descriptions follows.

On-line Methods

These may involve the introduction of test signals into a process to assist in the identification of locally valid models. Another online method takes advantage of a good understanding of the process in order to structure the underlying control philosophy so as to track the economic optimum in real-time.

Pattern Search Methods

Standard optimization techniques such as 'hill climbing' are applied directly to the process to determine the steady state optimum. Garcia and Morari (1981) point out that these methods must ensure that the process achieves steady state at all of the trial points, before being able to evaluate the objective function and implement the next optimization move. This means that systems with slow settling times (or indeed where settling times are comparable to the disturbance frequency) may exhibit poor performance.

Empirical Model Based Methods

This involves the application of simple recursive techniques to generate simple dynamic models based on the observed transient behaviour of the respective process. Unlike pattern searches, there is no requirement that the process must achieve steady state, since steady state information can be extracted from the dynamic model. This is attractive since it allows quicker tracking of the economic optimum.

Constraint Control Philosophy

When the economic optimum is defined by a set of constraints, it may be possible to determine the active constraints and observe how they change, and drive the process to track these constraints. This method reduces the computational load by eliminating inactive constraints while adding robustness to the controller.

Before these are discussed in the light of milling, off-line methods will be described.

The Off-line Methods

Off-line methods generally use mechanistic process models to predict he location of the economic optimum of the process, before actually implementing any steady state operating point shifts. Mechanistic models have the advantage that they are meaningful from a physical point of view. This implies that their predictive range can be expected to be wider than that of empirically based models.

The approach does however have several drawbacks:

- Many states, which form part of mechanistic models, cannot be directly measured.
- Models often lack the accuracy required to safely keep the plant in close proximity to limiting constraints.

As a result, off-line implementations need to infer the values of unmeasurable states, and match predictions with measurements using parameter estimation and data reconciliation techniques.

It is important to select a model with a suitable degree of complexity. The more complex, the more maintenance will be required. The model must also allow sufficiently rapid solution to allow implementations to occur at the desired frequency.

It is also beneficial to screen the recommendations of the optimizer before they are implemented. For example, it must be ensured that no significant changes occurred while solving the optimization problem that may render the result obsolete. Given the measurement uncertainty, it is also useful to check that the predicted value for the objective function for the proposed change is indeed an improvement over current conditions.

The pertinence of these strategies to milling is discussed next.

2.8.3 On-line Optimization of Milling Circuits

Milling circuits are complex and nonlinear in nature and are therefore difficult to model in a meaningful way. Parameter estimation is difficult as very little data is available, while the data that is available tends to be noisy (Zeng and Forsberg, 1994).

On-line optimization has up till the present been confined to the on-line type, using the pattern search method. The control strategy of Craig *et al* uses the hill climbing technique, which recognizes that the objective is a monotonic function of the load setpoint.

As is described in the preceding section, this technique is confined to operate using steady state data and is as a result slow to track process changes.

As far as optimization of milling circuits is concerned, the logical step in improving the optimization is to include some model information in the strategy. The framework would therefore change from the simple hill-climbing algorithm:

i) measure power

ii) compare with previous measurement

iii)implement one of two new setpoints, depending on the result of the comparison.

to a model based strategy:

i) measure states

ii) compute point of optimum operation

iii)implement setpoint changes to drive system to predicted optimum.

This forms the basis for the new control strategy proposed as the central theme of this thesis. The principles and guidelines provided by Young (1997) as given above were used during the derivation of the new strategy.

2.9 Milling Circuit Theory

The process of comminution is a complex interaction of many fields of science. Conditions inside a mill present the scholar a formidable task. Standard material balances are insufficient, as the properties of the solids are not necessarily dependent on their composition but rather on their size and crystal structure.

The liquid phase exhibits extremely complex apparent viscosities, which vary widely with the PSD of the solids and the temperature within the mill. There is a constant state of turbulence and high-energy impacts occur at random resulting in a variety of effects.

When one considers then that this environment is only a part of larger circuit in which separations occur and other streams may be introduced, it can be understood why little success has been achieved in theoretical modelling of the process.

Fortunately, with the advent of computers with increased numerical power, mill modelling has been able to go the next step.

2.9.1 Circuit Simulation

As computing power has increased, it has become possible to write computer programs that consider large numbers of particles and apply both known scientific phenomena and statistical analysis to provide a good representation of the comminution process.

Circuits are typically simulated with a model for each sub-process in the circuit. There will be a model for residence times, a model for breakage, a model for power consumption, a separation model for the cyclone and so on.

The central model describes the breakage within the mill. Researchers have tried several ways of modelling breakage and the most popular thus far has been the breakage and selection function that has been refined over the years by Lynch (1975), Austin (1984), Fuerstenau (1985), Herbst (1982-1991), Gao (1990), Petterson *et al* (1992) and several others. Refinements include modifications for particularly large particles which may act as media as well as modifications to allow modelling of semi autogenous and autogenous type mills.

In the development of the new optimizing controller, it has been necessary to use a computer simulator for testing. The model used is property of Mintek and can therefore not be described in detail. Circuit modelling will therefore be discussed in a more general manner. (For the benefit of the reader, a simple simulator, utilizing many of the methods outlined here, is supplied with a few comments in Appendix 1).

Model Description

There are three main sections within the circuit that require models. These models are then combined using simple mass balance type relationships within the simulator, which iterates from model to model cyclically to ensure a successful balance for each time step.

Several particle size classes are usually used to characterize each stream following a geometric $\sqrt{2}$ series. Large particles behave differently and may be treated separately, while the a mill discharge grate may be used to release only particles smaller than some chosen value.

The Mill Model

Within the framework of the simulator, the mill model's purpose is to accept a well defined set of inputs (feed rate, size distribution, etc.) and calculate outputs based on this information. The main task is therefore to calculate the PSD of the product stream based on current and previous inputs.

A common first assumption is that mixing is perfect which means the product stream has the same PSD as the material inside the mill (barring the effect of the grate which retains all particles above a certain size). This allows a primary function to calculate the changes in particle size that occur at each time step and a secondary function to determine flowrates and loads (a population balance over each size class). The particle size changes are popularly modelled with two functions called the *breakage* and *selection* functions. The selection function determines which particles will be broken while the breakage function determine the sizes of the resulting fragments. All constants are estimated in an extensive identification campaign when the model is matched to a particular circuit with a particular feed.

The Selection Function

The selection function S is an array (s_i) that defines the rate at which material is selected from each size class. In the simulation it is related to the specific power input to the mill by the equation

$$s_i = s_i^E\left(\frac{P}{H}\right) \tag{2.18}$$

where P = net power input to the mill (kW)

H = total mass of material being ground (kg)

s_i^E = constant of proportionality [specific selection function] (kg.kW^{-1}hr^{-1})

The specific selection function (for fines) is given by

$$s_i^E = \kappa.d_m(i)^\alpha \tag{2.19}$$

where $d_m(i)$ = geometric mean particle size for particles in size class i (mm)

κ = breakage rate constant of proportionality for fines

α = slope of selection function

The Breakage Function

The breakage function B is also an array ($b_{i,j}$) that defines for each class i, the destination classes j for the progeny particles. As particles are decreasing in size $j \geq i$.

The array is calculated from curves that describe the size distribution of products of breakage which is obtained experimentally.

The Population Balance

A standard mass balance has the form

rate of accumulation = input rate - output rate + production rate - consumption rate (2.20)

This balance is applied to each size class, with the input rate defined by the mill feed, the output rate by the mill discharge, the production rate by the breakage function and the consumption rate by the selection function. The generalized balance is also applied to the water.

It is important to note that there is a well defined mass of each size class in the mill at any one time. This means the output is a function of the input over a long time. This allows the simulator to model the dynamic effects of disturbances by introducing a residence time distribution model. It can be set up to model dead times and time constants making it amenable to process control considerations.

The Sump and Pump Model

In order to make the cyclone feed rate independent of the mill discharge rate a sump is required. The purpose is therefore to contain surges to ensure that the cyclone can be kept at the desired pressure.

The simulator therefore also needs to perform this task. A mass balance is again used - the input is the output from the mill and the output is defined by the pump. Again the sump is assumed to be perfectly mixed and the output is assumed to have the same characteristics as the sump load.

The Cyclone Model

The cyclone serves to return material that is insufficiently reduced in size. It works on the principle that larger (and denser) particles settle faster through the liquid medium for a set force. The slurry is fed tangentially into a conical device with openings at the top center and the bottom. Larger particles are moved to the outside by centripetal forces while finer particles remain near the axis and are allowed out the hole at the top. As more slurry enters it forces the larger particles to move down the cylinder and out the bottom orifice (spigot).

The dynamic modelling of cyclones, like that of milling, is still in its infancy. At present the best models are still largely empirical. A common strategy for simulating cyclones behaviour is described next.

Estimating the Cut Size

The first step is to estimate the particle size which has an equal chance of reporting to either the overflow or the underflow (d_{50c}) (Flintoff *et al*, 1987).

$$d_{50c} = F_1 \frac{39.7 D_c^{0.46} D_i^{0.6} D_o^{1.21} \eta^{0.5} e^{0.063\Phi}}{D_u^{0.71} h^{0.38} Q^{0.45} \left(\frac{\rho - 1}{1.6}\right)^k} \qquad (2.21)$$

where Q = the volumetric flow rate into the cyclone ($m^3.s^{-1}$)

η = the viscosity (cP)

ρ = density of slurry ($kg.m^{-3}$)

F_1 = Plitt model calibration parameter

k = exponential for the effect of solids density on the corrected cut size

Φ = volumetric fraction of solids in cyclone feed

Default values for F_1 and k of 1 and 0.5 respectively were found by Cilliers and Hinde (1991). Refer to figure 2.5 for cyclone measurements.

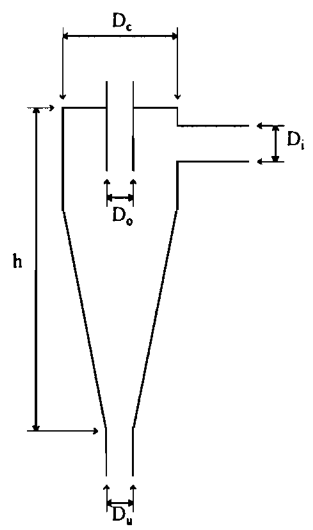


Figure 2-5 Hydrocyclone. Dimensions used in Plitt model are illustrated.

Volumetric Split

The ratio of underflow to overflow (volumetric) is given by

$$s = F_2 \frac{18.62 \rho_s^{0.24} \left(\frac{D_u}{D_o}\right)^{3.31} h^{0.24} \left(D_u^2 + D_o^2\right)^{0.36} e^{\rho_f \Phi}}{D_c^{1.11} P_{drop}^{0.24}} \qquad (2.22)$$

where ρ_f = density factor (to be fitted to specific plant)

F_2 = split flow calibration factor (to be fitted to specific plant)

Q = volumetric flow rate into cyclone

P_{drop} = pressure drop across cyclone

Pressure Drop

The pressure drop is calculated using this equation

$$P_{drop} = \frac{1.88Q^{1.78}\rho e^{0.0055\Phi}}{D_c^{0.37}D_i^{0.94}h^{0.28}\left(D_u^2 + D_o^2\right)^{0.87}} \tag{2.23}$$

Classification

The split of each size class to the overflow and underflow is usually estimated with a Rosin-Rammler type expression which generates a classification curve that is centered on the d_{50c} calculated above.

$$c_i = \frac{\exp\left(\alpha_{lynch}\frac{d_m(i)}{d_{50c}}\right) - 1}{\exp\left(\alpha_{lynch}\frac{d_m(i)}{d_{50c}}\right) + \exp\left(\alpha_{lynch}\right) - 2} \tag{2.24}$$

where c_i is the classification function for particles in size class i. Plitt gives an estimation for α_{lynch}:

$$\alpha_{lynch} = 1.54m_{Plitt} - 0.47 \tag{2.25}$$

where m_{Plitt} is given by:

$$m_{Plitt} = F_3\left\{1.94\left[\frac{D_c^2 h}{Q}\right]^{0.5} e^{-1.58V_u}\right\} \tag{2.26}$$

where F_3 = sharpness of classification (to be calibrated for specific plant- default =1)

V_u = volumetric recovery of slurry to the underflow

V_u is calculated directly from the volumetric split as follows:

$$V_u = \frac{S}{1+S} \tag{2.27}$$

The solids reporting to the underflow can then be calculated from c_i which shows what fraction of the ith size class reports to the underflow.

2.9.2 Power Consumption

Another important aspect of milling is the power consumption. Not only is the continuous consumption a major plant expense, but peaks in power consumption are also highly costly. Power suppliers charge for electricity depending on several parameters:

- overall power consumption, i.e. the integral of power consumption over entire period.
- hours of use. The cost of electricity varies depending on time of day (for practical reasons, power supply companies generally prefer to produce a standard amount of power, rather than to fluctuate as the demand varies).
- peaks in consumption. For the same reason, periods of extreme consumption are heavily penalized.

These considerations make power management an important issue in milling circuit control.

Power Consumption Modelling

In the computer simulation Austin's equation (1990) was used to estimate the power consumption.

$$P = 10.6D^{2.5}L(1 - 1.03J_t)\left[(1-\in)\frac{\rho_s}{W_r}J_t + 0.6J_{steel}\left(\rho_b - \frac{\rho_s}{W_r}\right)\right]\phi_c\left(1 - \frac{0.1}{2^{9-10\phi_c}}\right) \quad (2.28)$$

where D = mill diameter (m, inside liners)

L = mill length (m, inside liners)

$\in$ = voidage of mill charge (≈0.3)

J_t = Total fractional filling of mill with charge

J_{steel} = fractional filling of mill with balls (0 for autogenous milling)

ρ_s = density of solids (kg.m^{-3})

ρ_b = density of balls (kg.m^{-3})

W_r = mass ratio of (solids) to (solids + water) in the mill

ϕ_c = fraction of critical speed of mill

Other models for power consumption may consider liner design or models of flow patterns within the mill (Kapur *et al*, 1991). In Chapter 4 another power model is introduced. This model considers the mass of the mill and the distribution of material in the mill. These are translated in torque estimates and ultimately power consumption estimates.

This new model is required to ensure the fair comparison of all the control strategies tested in the course of this project. The power model described above (equation 2.28) is utilized by the simulator in the process of calculating the selection function. This means that the two variables (*throughput* and *power consumption*) are closely related. This compromises the assumption made in the derivation of the hill-climbing control strategy proposed by Craig *et al* (1992) that power consumption is proportional to throughput. In other words, the simulator and the control strategy are based on the same premise which may lead to a biased comparison. For this reason, another power model is used by the controller for its power estimate. This model is described in Chapter 4.

2.9.3 Viscosity

The solids density can be extremely high in the milling environment. At densities greater than 50% (by mass) solids, slurries of irregularly shaped mineral particles tend to exhibit Bingham-plastic non-Newtonian behaviour (Kelly *et al*, 1982). The viscosity for such slurries may be estimated with the following equation (Heikanen *et al*, 1979):

$$\frac{\mu}{\mu_{H_2O}} = 1 + 2.5F + 14.1F^2 + 0.00273e^{(16.0F)} \tag{2.29}$$

Where F is the volumetric fraction of solids in the slurry. An alternative formulation is given by Zenz *et al* (1960):

$$\frac{\mu}{\mu_{H_2O}} = e^{\left[\frac{4.1(1-\epsilon)}{0.64+\epsilon}\right]} \tag{2.30}$$

Where ϵ is the voidage in the slurry (volume fraction not occupied by solids).

$$\epsilon = (1 - F) \tag{2.31}$$

As a result of the large energy expenditure in milling, the temperatures within the mill may be significantly warmer than the surrounding environment. These temperatures have a significant effect on the viscosity. This effect is modelled with this equation (T in Kelvin):

$$\frac{\mu(T)}{\mu(298.15)} = \frac{1154}{2.1482\left(T - 8.435 + \left(8078.4 + (T - 8.435)^2\right)^{0.5}\right) - 120} \tag{2.32}$$

The actual particle size distribution can have an significant effect on the viscosity. Thomas (1965) gives a correlation in figure 2.6.

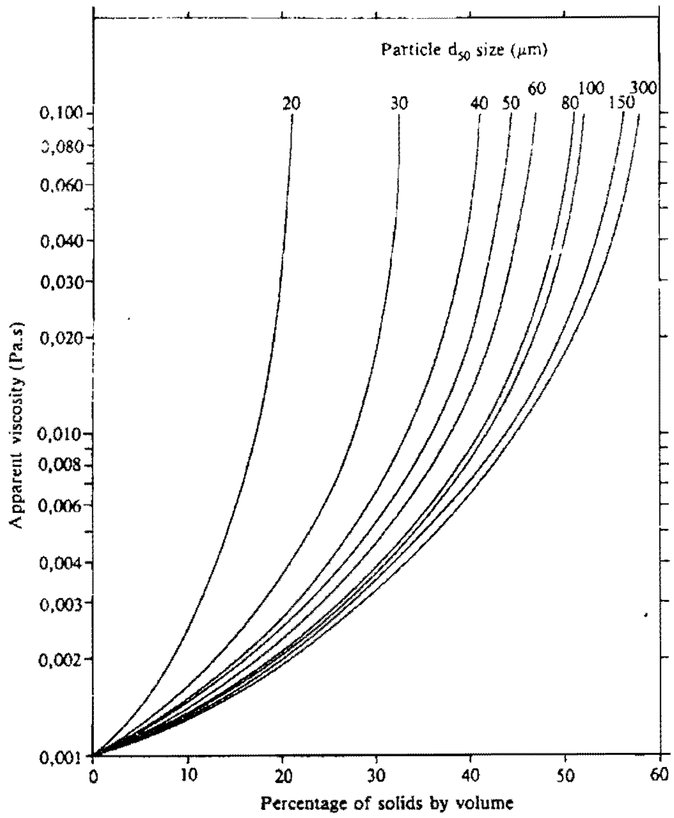


Figure 2-6 Apparent pulp viscosity as a function of solids concentration for various particle size distributions (Thomas, 1965).

With these correlations it is possible to estimate the viscosity of slurry based on its temperature, % solids and particle size distribution. Unfortunately, this information is often difficult to extract on-line, and filtering and estimation are also susceptible to error. In order to get a good representation of the flow in the mill it may be necessary to make a more direct measurement.

On-line Measurement

Shi and Napier-Munn (1996b) present a procedure for the on-line estimation of the rheological properties of unstable (settling) slurries. They point out that certain slurry types exhibit different viscosities at the same % solids and PSD as the result of complex shape and surface interactions.

Their procedure utilizes a Debex viscometer (described by Reeves, 1985, and Napier-Munn, 1990) which is run parallel to a process stream, requiring 2-5 l/s of slurry.

During simulations in the following chapters, Thomas' model will be implemented since more sophisticated measurement or estimation is not available in the simulation environment. In practice Thomas' model could be replaced by either an improved model (for example that of Shi and Napier-Munn, 1996a) or a method of direct measurement as described above.

2.9.4 Mill Critical Speed

The critical speed for a tumbling mill is defined as the rotation rate that results in a centripetal force of g m.s^{-2}. At this point material in the mill will no longer tumble but will remain attached to the mill lining (Kelly *et al*, 1982).

General formula

$$\text{Critical speed} = \pm\sqrt{\frac{g}{2\pi D}} \quad \text{Hz} \tag{2.33}$$

where $g = 9.80665 \text{m.s}^{-2}$

D = diameter in meters

Specific formula

$$\text{Critical speed} = \sim 42.29 D^{-1/2} \quad \text{r.p.m} \tag{2.34}$$

2.9.5 Modelling of Controllers

An important aspect of circuit modelling involves the accurate representation of circuit controllers. Fortunately, computer simulations are well suited to control since the circuit is totally identified - that is, everything is known; the densities in all the streams, the size distributions, flowrates etc. If the task were to derive a controller based only on a simulator a theoretically ideal controller could be implemented, since the models obviously could be "modelled" precisely.

It is therefore necessary to be aware of what is practical for a real plant; for example, it is only possible to control outputs which are practical to measure on a real plant.

For the purpose of designing a supervisory type controller using a simulator it is necessary to simulate not only the circuit, but the feedback controllers and their multivariable structure too.

To make the simulator more realistic it was decided that the control configuration for an existing plant should be used. In this case a plant with a 3x3 multivariable controller was chosen. The three loops were:

- solids feed rate to control mill load
- pump speed to control sump level

- sump water addition to control product particle size

Each loop was set up with a PI feedback controller (explained earlier in this chapter). A multivariable controller design was used to determine the values of K_C and τ_I. The INA was used for this purpose (this technique is described earlier in the chapter).

2.9.6 Liberation

The aim of mineral processing is to extract the valuable fraction from the ore. There are many ways to effect the separation, from simple jigs to complex ion-exchange and electrolysis techniques. Many of these require the valuable mineral to be either exposed or actually separate from the gangue.

The process of comminution aims to *liberate* the valuable fraction by means of size reduction. It is therefore important to describe the efficiency of comminution in these terms. The quality of a product is determined by the degree of liberation attained, not merely by the PSD of the product.

This means energy efficiency can be redefined; no longer is an efficient mill one which generates the most surface area per joule of energy, but the one that liberates the most valuable mineral per joule of energy. For this definition to have any application, liberation needs to be strictly defined.

The simplest definition is *the fraction of valuable mineral that is fully liberated.* This returns a single fraction. Unfortunately, when the valuable mineral has a very low concentration, this definition may become meaningless.

Another definition describes the distribution of valuable mineral in several classes, ranging from fully liberated to completely surrounded by a gangue matrix.

Since the actual requirement in comminution is the liberation of value minerals, the purpose of a milling circuit should not be to reduce particle size to a certain preselected value, but rather to achieve a preselected liberation.

Both 2 and 3 dimensional models have been derived to predict liberation as a function of particle size. Such a model is described next.

Relationship between particle size and degree of liberation.

The aim of a comminution circuit is to liberate valuable minerals from a gangue matrix. The liberation of a mineral is driven up by reducing the particle size of the ore. To answer the question of what PSD is required to effect a certain degree of liberation the relationship between particle size and liberation needs to be examined.

There are several treatments of this problem available (Wiegel, 1967, King, 1979, Belardi *et al*, 1986, etc.). Some use statistical analysis in both two and three dimensions which vary in their accuracy and applicability. Computers have made it possible to exhaustively calculate liberation based not only on particle size distributions, but also shape factors and the effect of preferential breakage on mineral interfaces.

For the purpose of this analysis all that is required is an extremely simple relationship to help demonstrate which variables are related to the degree of liberation. This section therefore derives a few relationships based on simple arguments.

Assuming that the mineral grain size* in the samples varies with time, a function is required to determine the degree of liberation at any one time.

$$\lambda = f(\sigma, \omega) \tag{2.35}$$

where λ is the degree of liberation (a fraction), σ is the particle size distribution and ω is the grain size distribution (σ and ω are vectors) in any particular sample.

In order to derive a suitable function, boundary conditions are needed. If the mineral grain sizes (ω) become very large, then the degree of liberation λ, would tend to 1, and vise versa. If the actual particle sizes became very small, λ would also tend to 1. Again the opposite also applies. A simple version of the function could be:

$$\lambda = \frac{\omega \cdot (\sigma + k_1)}{(\omega + k_2) \cdot \sigma} \tag{2.36}$$

(where k_1 and k_2 are constant vectors). However, when the sizes distributions are equal, this function will return a liberation of 100%. The likelihood that a solid matrix of rock, comprised of mixed minerals could be broken perfectly along the mineral interfaces is vanishingly small. As it is essentially a random process, stochastic methods are necessary to make this function more realistic.

Another model is proposed which represents the desired and undesired minerals as colours; the aim is to get as much completely black string by cutting up a long strand of string made of alternating black and white strips. If each strip is of length l and the strand is cut into segments of length l, then the resulting degree of liberation would be 0 (unless the cuts coincided exactly with the colour interfaces, giving a liberation of 1 - the likelihood of which is statistically negligible). If it were cut into strips of length

* the grain size is the average size of discrete mineral types in a mixed rock matrix. If a rock were broken along all mineral boundaries, the PSD of the product would correspond to the distribution of grain sizes.

$l/2$, then the "degree of liberation" would be ½. If the strips were of length $l/3$, the liberation achieved would be 2/3.

Extending this to three dimensions, if a matrix of black and white cubes, each with side of length l, is cut into cubes of length $l/2$, the liberation achieved would be $(½)^3$. Similarly, if it were cut into cubes of length $l/3$, the liberation achieved would be $(2/3)^3$. The logic can be extended such that:

$$\lambda = \left(\frac{\omega_m - \sigma_m}{\omega_m} \right)^3, \text{for } \omega > \sigma \tag{2.37}$$
$$\lambda = 0, \text{for } \omega > \sigma$$

This model does not, however, account for the *distribution* of sizes experienced in reality which allow smaller particles to be generated from bigger grains and vice-versa.

While the above models have ranges in which their application will be satisfactory, a more general model may be required. Assume a volume, V consists of n_i sub-volumes (cubic with edge size distribution ω), m_i black and n_i-m_i white. This volume, V, is then broken into n_0 cubes (with size distribution σ). The question to be answered is: how many of the n_0 cubes are completely white? Let m_0 represent this number. We require:

$$\lambda n_o = m_o = F\left(\frac{m_i}{n_i}, \omega, \sigma \right) \tag{2.38}$$

and as the breakage process is large random, experiments (simulations or real tests) could be used to generate F. As mentioned in the introduction to this section, this function has been derived by several authors: Wiegel, King and Belardi.

2.9.7 Some Mechanical Considerations

There are several aspects of mill operation that need to be considered before operational decisions can be made. Each of these considerations tends to require a compromise between two conflicting requirements.

The Grinding Action

As attrition acts on the outside surface of each particle, it is logical that a finer feed would promote efficient breakage (more surface area). However, it must be noted that grinding action and throughput will suffer when there is a lack of rocks to act as grinding media. This is one of several balances that determine energy efficiency and product quality.

The effectiveness of the grinding action also depends strongly on the slurry rheology. Attrition may be retarded as low viscosities where the transfer of energy from the shell to the charge is retarded; however the reverse may also be true, since high viscosities prevent slippage, the basis for the abrasive action of attrition. A suitable apparent viscosity is therefore also required. Viscosity is a function of the % solids, the PSD (particle size distribution), the temperature and the shape of the particles.

It seems intuitive that cascading[*] action will result in breakage that favours attrition, while cataracting[*] will result in breakage that favours crack generation and propagation.

In must also be noted that the two actions result in a different product. Not only will attrition result in shrinking cores with progressively exposed exteriors, but the fines generated may have different properties. For example they may be more "chip" shaped, as they were in fact "chipped" from the surface. This may mean higher surface area and strange flow characteristics (poor performance in cyclones - Cilliers *et al*, 1991). In some cases this may even be desired (graphite milling). Attrition may also result in excessive fines (attrition is undesirable when an extremely coarse product is desired). In general a combination of the two breakage modes is preferred.

Circulating load

Higher circulating loads result in lower single-pass residence times. This allows smaller particles to escape the circuit and avoid overgrinding. Higher circulating loads however do require greater capacity from the cyclones and pumps, and also cause greater wear.

Higher circuit loads also result in a slightly larger average particle size within the circuit, promoting the grinding action.

Mill aspect ratio

As with circulating load, the aspect ratio must be chosen as to reduce overgrinding (large *L/D*) (Rowland, 1988). Large diameters also result in a coarser charge and promote efficient breakage.

Another merit of low ratio mills is that they can be more closely modelled by perfect mixing models. Longer mills result in different slurry rheology depending on the

[*] cascading material is the material that tumbles in the main body of the mill load (see figure 4.6).

[*] cataracting material leaves the surface of the main ore boby and travels through the air for a period (see figure 4.6)

distance through the mill. The better mixed charge found in wider mills therefore allows more uniform conditions which promote efficient grinding.

Power draw is closely related to the diameter. It is popular to relate power draw to $D^{2.5}$.

Critical Size Build-up

These occur often as a result of density effects in the cyclones. Often heavier minerals (including gold), if they manage to escape the mill, continually pass through the underflow of the cyclone. This results not only in overgrinding of denser materials, but in the waste of energy and of useful mill volume.

Such build-ups also occur when streams are returned from further upstream for further grinding. This may result in certain mineral types cycling through the loop continually.

Chapter 3

Proposed Control Strategy

In order to control a plant optimally, it is necessary to examine all the information available to the controller and to determine which is most useful. On a milling circuit power consumption data, pressure data, load data and many more are all available continuously and the challenge is to extract benefit from the information.

Physics theory may be applied to predict interactions within the grinding environment, however, due to the complex nature of fracture and particle behaviour, a truly useful model is still a long way away. For the present it is necessary to simply make correlations, examine relationships and build empirical models.

In this chapter, one such model relating throughput to load and viscosity is proposed. The motivation is that once such a model is fitted to a plant, it can be used to determine the load that would result in the greatest throughput for a given viscosity, which it would need to 'read' from the plant in real-time.

This forms the basis for the model-based optimizing controller which is described in this chapter.

The chapter begins by describing the circuit chosen for examination. It then continues to describe the fundamental steps that lead to the final control strategy followed by an example of the strategy in action (in its simplest form) with step by step discussion, concluding the chapter.

3.1 Defining the Grinding Circuit

Many configurations are used for autogenous mills, but for simplicity and generality a simple reaction-separation type circuit will be used (see figure 3.1). It is assumed that the mill is controlled with a 3x3 multivariable controller as described in the previous chapter.

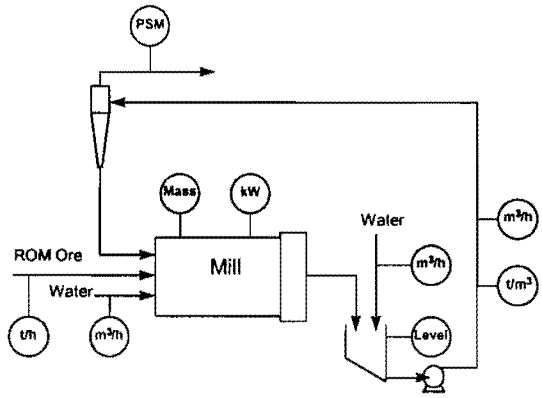


Figure 3-1 Autogenous milling circuit used for controller design.

It is worth noting some of the measurements made:

Controlled variables are:

- mill mass,
- sump level, and
- particle size distribution in product.

Other variables include:

- mill power consumption,
- cyclone feed density,
- cyclone feed volumetric flow rate, and
- mill water addition rate*.

Manipulated elements include:

- the solids feed rate,
- the cyclone feed pump speed (i.e. the volumetric flow rate to the cyclone),
- the sump water addition rate.

* This is assumed constant in this chapter. The possibilities for inclusion in the controller framework are discussed in Chapter 4.

The multivariable controller guides the three "controlled" variables by manipulating these elements. This controller can be designed in several ways; for the purpose of this thesis the Inverse Nyquist Array (INA) method was used‡. This design effectively compensates for complex interactions by making the plant controller "appear" to be three independent single loop systems, one for each output. PI (proportional integral) controllers are then used for each of these loops.

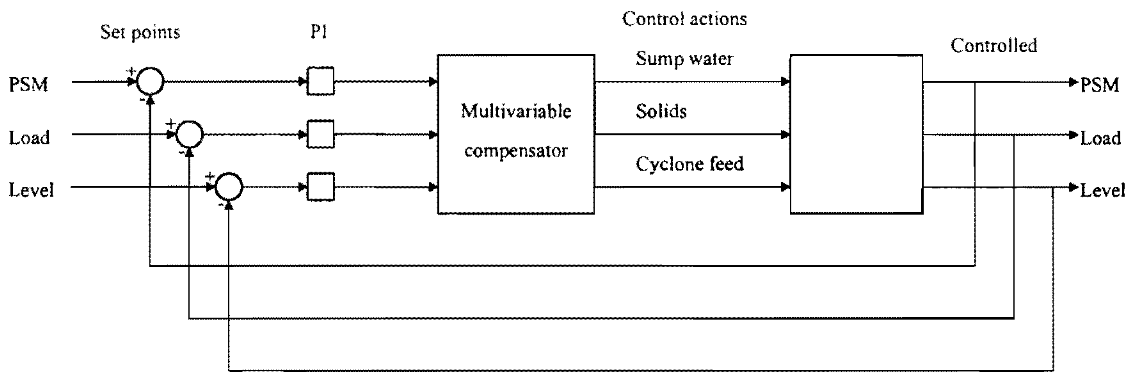


Figure 3-2 The multivariable control scheme (from Craig *et al*).

It is worth noting at this point that there are three additional "free" variables, mill water addition, product size setpoint and mill speed (mill speed is usually set at a predetermined fraction of the mill's critical speed). Further possibilities for these are discussed in Chapter 4.

3.2 Evolution of the Proposed Control Strategy

The optimizing control strategy of Craig *et al* (1992) re-evaluates the load setpoint used by the multivariable controller periodically (see Chapter 2). The setpoint prediction is based upon power consumption, which is assumed to be proportional to throughput.

It is logical that there will exist a load that will result in the greatest power consumption and it also logical that there will be a load that will result in an optimum throughput. Craig *et al*'s assumption requires that these points coincide. One of the early aims of the new strategy was to remove this assumption.

It was also hoped that optimum setpoints could be selected faster and without the oscillation described in Chapter 2.

‡ Described in Chapter 2.

From the brief discussion of on-line optimization in the previous chapter, it is apparent that model information may be used to improve the optimization procedure. The following three subsections describe some of the thinking behind a new control strategy that aims to do use such model information to achieve better results:

- **first section:** this describes how certain variables interact, allowing the reader to appreciate how certain actions would effect system states. A 'chain' of logic shows how throughput depends directly on the breakage rate within the mill, which subsequently depends on conditions inside the mill. This exposes the possibility for correlating throughput to certain individual states that reflect the 'conditions' within the tumbling slurry. *Viscosity* within the mill is identified as having the greatest potential as a 'representative' state and is developed further. This correlation between throughput and viscosity is then proposed as a model that could be used by a controller to seek the throughput 'peak'.
- **second section:** this describes how such a model can be developed to be utilized by a model based optimizing controller.
- **third section:** this demonstrates how the model used by the optimizing controller can be modified to allow on-line adaptation. The model would then 'fit' itself to the plant during operation.

3.2.1 Circuit Modelling

Very little is understood about the relationships between circuit throughput* and plant characteristics. In this section some models and interactions are presented to aid the development of a largely *empirical* model to perform the task of relating circuit throughput to system states within the controller framework.

The basic aim is to develop a model relating circuit throughput to mill load. Parameters used in the model need to be measurable on-line and the simpler the model the better. The aim is therefore to develop a 3-D surface relating circuit throughput to load on one axis and to another state on the second axis. The aim of this section is to identify a state that could fit such a model.

As the primary aim was to maximize circuit throughput, a logical first step was to examine the features of the circuit that impact directly upon throughput. Factors that relate to throughput are examined in the next few sections.

* circuit throughput is the rate at which product leaves via the cyclone overflow - not to be confused with the *mill* throughput which is actually the circulating load in the circuit, and is typically greater than the circuit throughput.

Circuit Throughput - Ore Characteristics Relationship

It is possible to redefine circuit throughput such that many important relationships may become clearer. As the aim of the circuit is to reduce particle size, circuit throughput can be equated to a "particle size flux" within the mill. In other words, the mass of material whose size passes a certain value is considered successfully milled. In the following text circuit throughput is related to particle size, particle hardness and apparent viscosity within the mill.

Seven variables have been selected to describe the circuit:

- M: Product rate ($kg.s^{-1}$) - material leaving he circuit
- L: Mill load (fractional filling, unitless)
- P: Power consumption (kW)
- d_i: 50% passing particle size in feed (μm)
- d_o: 50% passing particle size in product (μm)
- h: Feed hardness parameter (normalized bond work index, dimensionless)

Development

Logic dictates that the larger the initial particles, the smaller the possible production rate - after all a mill can only perform a finite amount of breakage. One can however take this further by relating the starting and finishing particle sizes to a surface area change, which in turn relates to actual breakage performed.

If it is initially assumed that a milling circuit is capable of generating a constant surface area per unit time (and for a constant feed hardness), the following analysis can be used:

Assuming spherical particles means that a particle of diameter d_i has a surface area of πd_i^2 and the volume $\pi/_6 d_i^3$. If this particle is then crushed it will form n particles - with average size d_o with the same total volume - and an increased surface area. The volumetric equivalence gives:

$$\frac{\pi}{6}d_i^3 = n\frac{\pi}{6}d_o^3 \tag{3.1}$$

rewriting,

$$n = \frac{d_i^3}{d_o^3} \tag{3.2}$$

while the new surface area is:

$$n\pi d_o^2 \tag{3.3}$$

so the newly generated surface area (per particle with initial diameter d_i) is

$$S = n\pi d_o^2 - \pi d_i^2 \tag{3.4}$$

$$S = \pi \frac{d_i^3}{d_o} - \pi d_i^2 \tag{3.5}$$

For a milling circuit capable of B m^2/hr, B/S particles can be crushed each hour. This relates directly to the throughput:

$$M = \frac{B}{6S} \rho_s \pi d_i^3 \tag{3.6}$$

which can be written in full (ρ_s is the density of the particle):

$$M = \frac{B\rho_s}{6\left(\frac{1}{d_o} - \frac{1}{d_i}\right)} \tag{3.7}$$

This equation relates throughput to the particle sizes entering and leaving the circuit. It is worth noting that for a constant size reduction ratio, the larger the feed particles, the larger the throughput; this may not be intuitively obvious.

The capacity (B) of a mill to generate surface area will depend directly on the hardness of the feed and may be better written as

$$B = \frac{B_m}{h} \tag{3.8}$$

where B_m is the intrinsic capacity of the mill and h is a measure of hardness (normalized bond work index).

Throughput - Ore Characteristics Relationship: Conclusion

The models given above show how throughput may depend on ore characteristics, such as the particle size and the hardness as well as the mysterious 'breakage capacity'. The next section shows how this parameter can be ascribed to the conditions within the mill - the fractional filling and the way the slurry moves within the milling environment. This argument can be developed to identify a single parameter that reflects the ore character.

A Parameter to Reflect Ore Character?

In the above model, the breakage capacity of a mill was assumed to be intrinsic and constant. This allows the effects of feed and product particle size to be evaluated very simply. However, the breakage rate achieved in a mill is not constant and depends upon the conditions within the mill. Throughput therefore depends not only on particle size and hardness but by how material moves within the mill.

Internal Conditions

As material tumbles in the mill it suffers forces of many types. Some result in fracture, some in attrition while others are absorbed in plastic deformation, noise etc. Optimum breakage occurs when the greatest surface area is generated per unit of time.

For constant energy input, this can be restated: optimum breakage occurs when the greatest surface area is generated per joule of energy. As there is a set requirement of energy for the generation of surface area (to break bonds), this requirement equates to minimizing the energy losses.

In other words, optimum breakage occurs when conditions are such that the greatest possible fraction of the energy is spent on surface area generation (Tkácová *et al*, 1994).

Proposal: the factors that determine these conditions are the *load* and the *apparent viscosity*.

The aim of this section was to develop a 3-D model relating throughput to two parameters, load and one other. This proposal suggests slurry viscosity as the second parameter.

Motivation: Particle size, particle shape and percent solids influence the apparent viscosity, which in turn influences the way material moves within the mill. It is all of these factors that influence energy transfer within the milling environment. This makes viscosity central to the efficiency of breakage.

Measuring Apparent Viscosity

Obtaining real time viscosity estimates from the heart of a mill presents a genuine challenge. It is possible to obtain a rough estimate the viscosity on-line via several steps. The controller has readings for the product PSM (particle size measurement) as well the cyclone feed rate and density. This allows the estimation of the viscosity within this mill using both a model of a cyclone and one for viscosity of heterogeneous mixtures based on PSD, percent solids and temperature (see Chapter 2).

This method relies heavily on the method of back calculation and is therefore not considered accurate, but as far as the correlation is concerned, accuracy is not as important as consistency.

If this method proves inconsistent, it may be necessary to utilize an on-line viscometer to aid in the estimation of viscosity. This is discussed in more detail later in this chapter.

Empirical Model

As stated earlier, how the breakage rate is affected by slurry behaviour within the circuit is still poorly understood. It could be assumed that ideal conditions consist of a mixture of several subtle effects that combine to provide several types of breakage and result in the highest energy transfer efficiency, making modelling exceedingly difficult. Work is being done on computationally intensive "finite element" type modelling where large numbers of particles are all defined and are each given defined properties and are then introduced into a milling environment where each collision is modelled separately (Mishra, 1994a & 1994b). This type of modelling may be extremely useful in improving the understanding of milling dynamics.

For the present it is still necessary to utilize an empirical model to relate throughput to slurry rheology.

Structure

The aim is to develop a 3-D surface that relates circuit throughput to load and slurry conditions. The parameter proposed to be most representative of "slurry conditions" is the viscosity which reflects both the PSD and the % solids, as well as several other more subtle factors.

From the modelling described earlier in this chapter, several "boundary conditions" can be established. Firstly, with respect to load, two effects are apparent. At a load of zero, throughput must necessarily be zero, while at a load of 100%, the throughput must also be zero. This is so since the movement required for breakage will not be possible if the mill were 100% full.

Logically, since the throughput is by definition positive, there must exist a load between 0 and 100% that results in the optimum throughput, and since there are no discontinuities in the process of changing the load, the curve is likely to be continuous and differentiable.

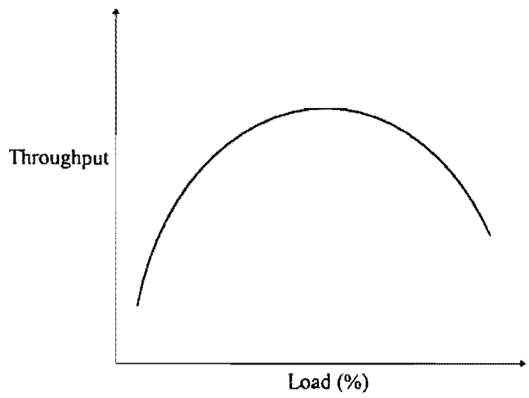


Figure 3-3 The relationship between throughput and load must exhibit a maximum and is likely to be a smooth function.

In order to ensure simplicity and efficiency, a simple quadratic was selected to model this relationship.

A relationship between circuit throughput and viscosity depends on the individual effects of particle size and percent solids as well as the direct effect of viscosity.

Equation 3.7 shows that circuit throughput exhibits an inverse relationship to the particle size of the feed (that is, fresh feed into the circuit). If it is assumed that a larger particle size in the feed will result in a larger particle size in the mill (this is commonly true since the circuit contains material 'between' the states of the feed and product), then this relationship can be presented:

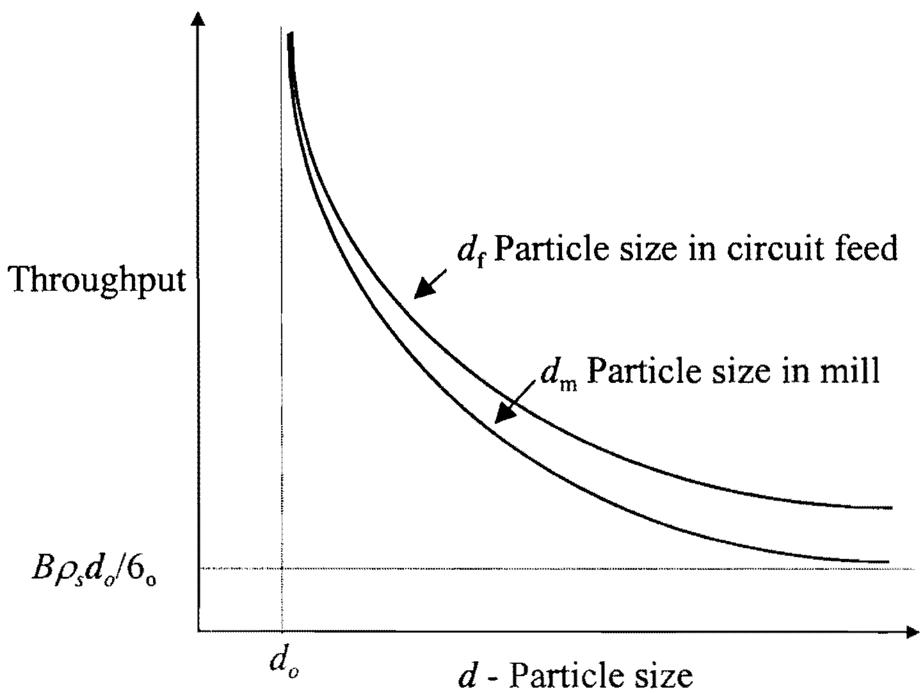


Figure 3-4 Relationship between throughput and particle size as predicted by equation 3.7. The asymptotes show the boundary conditions.

The relationship between throughput and viscosity is the result of several interactions. Primarily, viscosity affects breakage by altering flow patterns and energy transfer within the load. It must however be noted that the physical phenomena that 'cause' the viscosity have their own marked affect on throughput. Particle size is central in this issue - finer particles result in higher viscosities but may mean improved throughput.

The same is true for the % solids; while viscosity depends strongly on % solids as far as rheology is concerned, % solids also affects throughput directly. Water is essential to lubricate the material and allow it to be pumped through the circuit while in the case of autogenous milling, where the ore is the grinding medium, grinding efficiency is greatly retarded at low percent solids.

All of these factors ensure for a fairly complex relationship between throughput and viscosity. Very little specific information about the shape of the curve can be predicted other than at boundary conditions which are of little meaning in this context. Generally, it can be predicted that since all competing influences are continuous in the region of interest (10 to 50% loading) and have consistent concavities, the same must be true of the resulting relationship.

Since the aim is to fit a stochastic model to the relationship based on plant data, these points merely serve to i) ensure that the model's 'form' captures the salient features of the load-throughput-viscosity relationship and ii) to assist in the initial estimates for the model.

The 3-D surface is therefore proposed to have a quadratic shape in both variables:

$$t = x_1 + x_2 v + x_3 l + x_4 v^2 + x_5 l^2 + x_6 vl + x_7 vl^2 + x_8 v^2 l + x_9 v^2 l^2 \tag{3.9}$$

where v is the viscosity and l is the load. The vector $\mathbf{x}$ is the set of parameters available for plant-model identification.

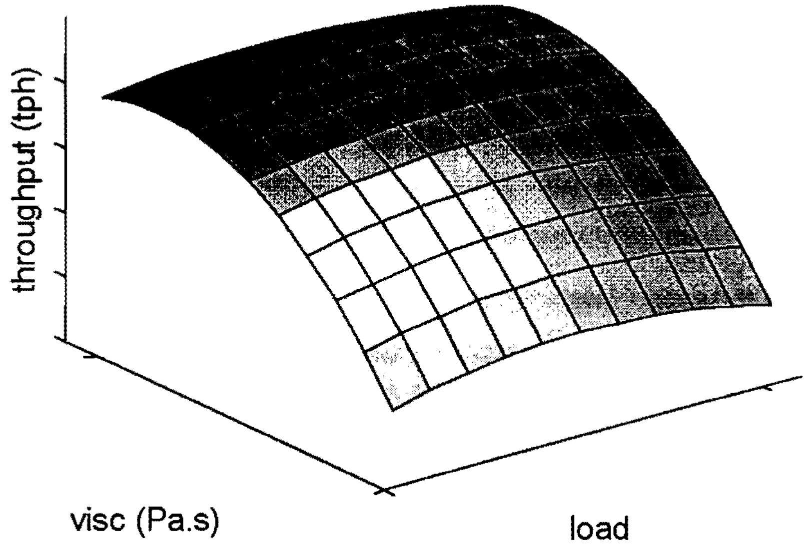


Figure 3-5 A possible shape of the throughput-load-viscosity surface.

An explanation of how this model is used by the controller is given in next section.

3.2.2 Using the Model in the Control Strategy

Supervisory controllers and specifically optimizing controllers aim to select setpoints that maximize a predefined objective function. In the preceding sections, a model that relates throughput to two system states is proposed. This model, once refined for a specific circuit ("identified") allows for precisely this type of control.

The model relates throughput to slurry rheology (as reflected by viscosity) and mill load. Once the viscosity of the slurry is estimated, the controller is in a position to select the load setpoint that optimized throughput according to the model. This setpoint can then be implemented. The procedure can then be repeated periodically.

It is possible to update the model during operation. A procedure for on-line model adaptation is now presented.

3.2.3 Controller Modification: On-line Adaptation

Earlier in this chapter it is shown how a plant model can be obtained and used to predict optimal load setpoints. However, this leaves the task of determining the model accurately prior to implementation.

To counter this a self adapting model has been devised such that a simple 3-D model can be used to start with and the controller will adapt it as it receives data.

The model, a 3-D surface, quadratic in two variables is split into n x m sections, for m values of load and n values of viscosity. A m x n matrix **T** (the throughput matrix), contains estimates for the throughputs at each of the corresponding sets of conditions (i.e. for each viscosity and load).

At the beginning of a run, these values are estimated as well as possible (throughputs for each load-viscosity set). Then, as the plant operates, load, viscosity and throughput data is filtered using a Kalman filter and recorded.

Based on the measured load and viscosity it is then possible to locate the area 'of operation' within the matrix **T** and update the throughput value found in that cell based on the throughput achieved while the plant operated in that region. Each cell's contents are therefore a function of a large amount of throughput data (say over the previous 24 hours), with more recent data being more strongly weighted. In this way the throughput matrix **T** is updated as plant operation wanders through the various elements.

As a further improvement, interpolation is used so that values in *four* cells are effected by each incoming datapoint, since the load and viscosity estimates will never precisely coincide, and will in fact lie somewhere between 4 adjacent cells in the matrix **T**.

Two dimensional interpolation is required to ensure that every new datapoint has a realistic effect on the model. When a two dimensional region is split into equal sized rectangles with certain properties defined at each corner, then property information about a coordinate away from one of the corners can be set to 'belong' to the nearest corner *or* can be assumed to belong to the *nearest four corners* with a weighting depending on the proximity.

The computation for this strategy involved two steps of interpolation, one in each dimension. Assume the x dimension is represented at positions $\{x_1, x_2, x_3, ..., x_n\}$ while the y dimension is represented by $\{y_1, y_2, y_3, ..., y_n\}$. Data corresponding to a point (a,b) between x_i, x_{i+1}, y_i and y_{i+1} would therefore be shared amongst the four intersections: $\{x_i, y_i\}$, $\{x_{i+1}, y_i\}$, $\{x_i, y_{i+1}\}$ and $\{x_{i+1}, y_{i+1}\}$. The split can be calculated using these formulae:

Fraction to $\{x_i, y_i\}$: $\dfrac{x_2 - a}{x_2 - x_1}\dfrac{y_2 - b}{y_2 - y_1}$ (3.10)

Fraction to $\{x_{i+1}, y_i\}$: $\dfrac{a - x_1}{x_2 - x_1}\dfrac{y_2 - b}{y_2 - y_1}$ (3.11)

Fraction to $\{x_i, y_{i+1}\}$: $\dfrac{x_2 - a}{x_2 - x_1}\dfrac{b - y_1}{y_2 - y_1}$ (3.12)

Fraction to $\{x_{i+1}, y_{i+1}\}$: $\dfrac{a - x_1}{x_2 - x_1} \dfrac{b - y_1}{y_2 - y_1}$ (3.13)

It is worth noting that the four formulae for two dimensional interpolation can be simply derived by multiplying the two one-dimension formulae by one another.

The 3-D surface used by the optimizing controller to select the load setpoint is still a simple quadratic in two variables that is fitted to the matrix **T** by means of a least squares algorithm:

The equation for the 3-D surface is given as:

$$t = x_1 + x_2 v + x_3 l + x_4 v^2 + x_5 l^2 + x_6 vl + x_7 vl^2 + x_8 v^2 l + x_9 v^2 l^2 \tag{3.14}$$

where t is throughput (tph) for a specific value of v (viscosity - Pa.s) and l (load). This can be rewritten as

$$a\mathbf{x} = t \tag{3.15}$$

If the estimates for throughput (for m values of load and n values of viscosity) supplied in the matrix **T**:

$$\mathbf{T} = \begin{pmatrix} t_{11} & t_{12} & \cdots & t_{1n} \\ t_{21} & t_{11} & & t_{2n} \\ \vdots & & \ddots & \vdots \\ t_{m1} & \cdots & & t_{mn} \end{pmatrix} \text{tph} \tag{3.16}$$

are rewritten as the vector:

$$\mathbf{b} = \begin{pmatrix} t_{11} \\ t_{21} \\ \vdots \\ t_{m1} \\ \hdashline t_{12} \\ t_{22} \\ \vdots \\ t_{m2} \\ \hdashline \vdots \\ \hdashline t_{1n} \\ t_{2n} \\ \vdots \\ t_{mn} \end{pmatrix} \text{tph} \tag{3.17}$$

we can write

$$\mathbf{Ax} = \mathbf{b} \tag{3.18}$$

where **A** is:

$$\mathbf{A} = \begin{pmatrix} 1 & v_1 & l_1 & v_1^2 & l_1^2 & v_1 l_1 & v_1 l_1^2 & v_1^2 l_1 & v_1^2 l_1^2 \\ \vdots & & & & \vdots & & & & \vdots \\ 1 & v_n & l_m & v_n^2 & l_m^2 & v_n l_m & v_n l_m^2 & v_n^2 l_m & v_n^2 l_m^2 \end{pmatrix} \tag{3.19}$$

with m x n rows; one for each permutation of v_i (1 to n) and l_i (1 to m).

It is now possible to attain the best estimate for **x** using least squares:

If $\mathbf{Ax} = \mathbf{b}$ is overdetermined then the least squares solution $\bar{\mathbf{x}}$ is given by
$\bar{\mathbf{x}} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$ (3.20)

This ensures that the model is still smooth and is equally influenced by each element of the matrix.

Now the plant will adapt the 3-D surface to match the plant automatically. This has the benefit that the initial guess for the surface need not be accurate, since the adaptation mechanism will eventually attain the true shape once sufficient data has been gathered. For example, if an initial guess at the throughput matrix T were:

$$T = \begin{pmatrix} 134 & 134 & 134 & 134 & 134 \\ 134 & 134 & 134 & 134 & 134 \\ 134 & 134 & 134 & 134 & 134 \end{pmatrix} \text{tph} \tag{3.21}$$

the model may adapt as shown in figure 3.7. This particular adaptation is for operation as described in Chapter 4 section 3 (*Tests*). Nominal operation was perturbed by deliberately driving the load setpoint from 0.3 to 0.4 over a 20 hour period., to ensure that several regions of operation where represented.

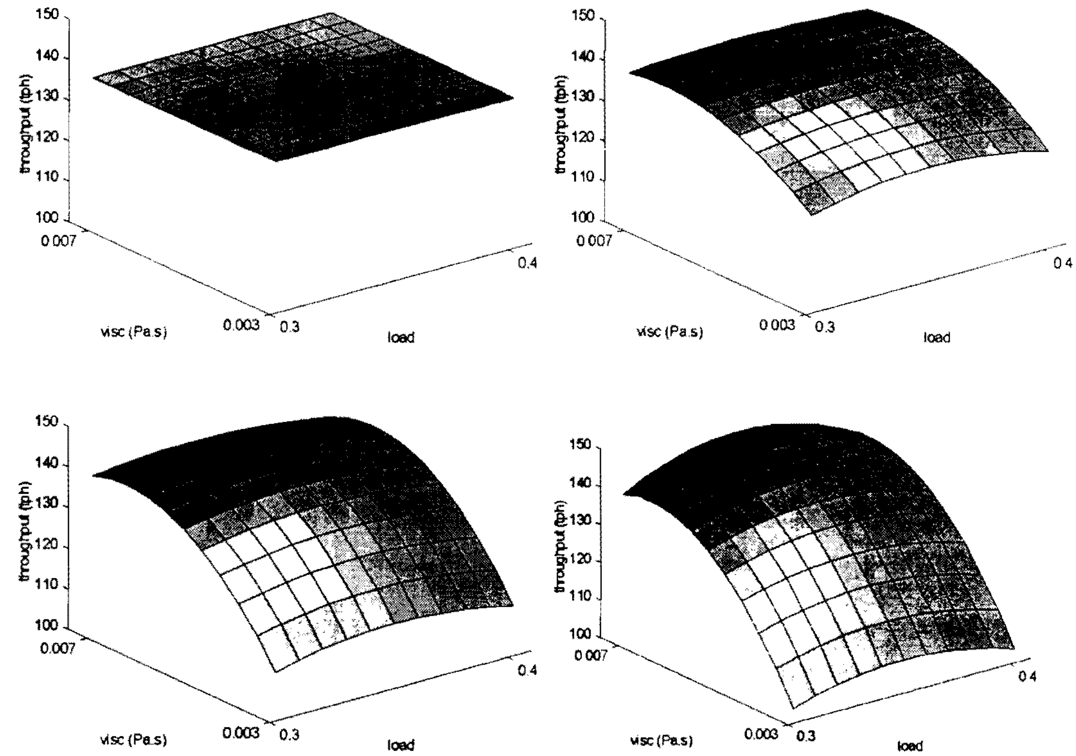


Figure 3-6 Evolution of 3-D surface from a flat plane to a surface fitting plant behaviour.

The throughput matrix has been altered and now appears as follows:

$$\mathbf{T}=\begin{pmatrix}100.4 & 119.8 & 132.5 & 136.8 & 136.8\\ 105.3 & 125.0 & 135.2 & 145.5 & 144.3\\ 100.0 & 120.1 & 131.2 & 138.1 & 137.7\end{pmatrix}\text{tph} \tag{3.22}$$

During the first few hours of operation, while the model adapts to match the plant, operation may not be optimal and it may be worth implementing an alternative optimization strategy to perform setpoint adjustment while adaptation occurs (such as the hill climbing strategy of Craig *et al* (1992). Of course, the better the initial guess at **T** the sooner the model will match the plant. The algorithm is extremely robust and has yet to fail to converge to a satisfactory result.

It must be pointed out that the speed of evolution depends strongly on how much the plant 'wanders'. If the plant stays in a single region of operation (constant viscosity) then the matrix **T** has only one value changed. Although this value receives a higher weighting due to the 'recentness' of the data, it is not possible to fit a satisfactory surface with only one reliable datapoint.

For this reason it may be necessary to set a 'flag' that prevents use of the model until several regions (each element in **T** cell represents a region of operation - a certain viscosity and load) have reliable data. This would only be necessary during start-up when a 'bad' guess had been made at **T**. After the plant has utilized the model for sufficiently long, operators will become familiar with typical values for the matrix **T** (the throughput matrix) and will be capable of making improved guesses.

Once the model fits the plant and is in operation, there are two forms of disturbance possible:

- modelled changes
- unmodelled changes

Modelled changes refer to changes in the plant that move the point of operation around *on the 3-D surface.* A change of viscosity, for example would be a modelled change. In this case the controller can easily re-evaluate the optimal load setpoint as described in the previous section.

Unmodelled changes include serious changes in the plant that render the 3-D model obsolete. In this case recommended load setpoints may be incorrect. This is where the adaptation scheme once again applies; it serves to observe the plant and 'fix' the model by refitting it to the plant data. Once this is done, load setpoints can once again be recommended by the controller.

Most disturbances ought to be of the first kind (such as changes in feed PSD or hardness) but certain events, like feed stoppages or pump malfunctions may alter the plant's fundamental character, requiring the 3-D surface to alter in shape or position.

Discussion

The self adapting modification not only improves the model in terms of accuracy and ease of use, but also allows the operator to switch between different strategies, depending on which has the greater potential for a given circumstance.

It is appreciated that the model based controller performs poorly when the model is poorly identified; this occurs either during start-up (as the result of a poor initial estimate of **T**) or during major plant alterations. Now the operator can disengage the model based controller in favour of the hill climbing technique (for example), until the self adapting model has adapted, and then re-engage the model based strategy. This possibility is discussed further in the next chapter.

3.2.4 Summary

In this section, an optimizing control strategy for milling circuit operation is proposed. The setup and operation of this strategy can be summarized as follows:

- An on-line viscometer is installed on the mill discharge of the existing circuit (or any other means of estimating the viscosity of the mill discharge).
- Readings are taking for an extended period of operation during which a correlation between slurry rheology, load and throughput is generated. Another optimizing control strategy such as the hill-climbing method is applied at this time to seek the optimal load setpoint.

Once a simple 3-D model is fitted to the relationship it is possible to implement the optimizing controller (fully automated):

- Viscosity reading made.
- 3-D curve cut at selected viscosity to expose 2-D curve.
- Optimization performed to select load at peak of 2-D curve.
- Load setpoint set to this value.
- This is repeated at a frequency best suited to capturing effects of gradual process changes.
- During continued operation, the 3-D curve is modified by incorporation of new process data. This ensures greater accuracy (model is permanently being improved) and allows for adaptation when the plant or ore characteristics change.

The theory behind this strategy can be summarized as follows:

- The optimum load setpoint varies depending on several subtle effects attributable to the ore and the plant characteristics, specifically within the grinding environment.
- It is argued that although slurry viscosity is not the main cause for these effects, it may serve to represent them in a general manner. Viscosity is therefore chosen as variable in the optimization since it represents several factors that may influence breakage and hence the position of the optimum load; i.e. % solids, PSD and particle shape, temperature, etc.
- The idea is therefore to examine how viscosity effects the optimum load setpoint and then to use this information at a later stage to select load setpoints based on measured viscosity.

With the theory in place it is now necessary to examine the strategy in operation. For this purpose a computer simulation program supplied by Mintek is used. This simulator uses many of the modelling strategies described in Chapter 2 as well as many refinements and innovations that improve the reliability of plant performance predictions.

Discussion

A few aspects of the strategy that may require clarification are now discussed.

Firstly, the premise that throughput is as strong function of load and viscosity needs to be probed. If load and viscosity effects were masked by the influences of, for example, cyclone performance, the value of the relationship would be greatly reduced.

Fortunately, many of the variables that influence circuit efficiency are also related to viscosity - the PSD, the % solids, the particle shape, the ore surface characteristics and even temperature. Other factors, such as cyclone split or circulating load are often just reflections of these changes.

This ensures that viscosity is indeed related to throughput and in fact shows that there are several links between them.

The next issue concerns the design of the model. The model has the form indicated in figure 3.7.

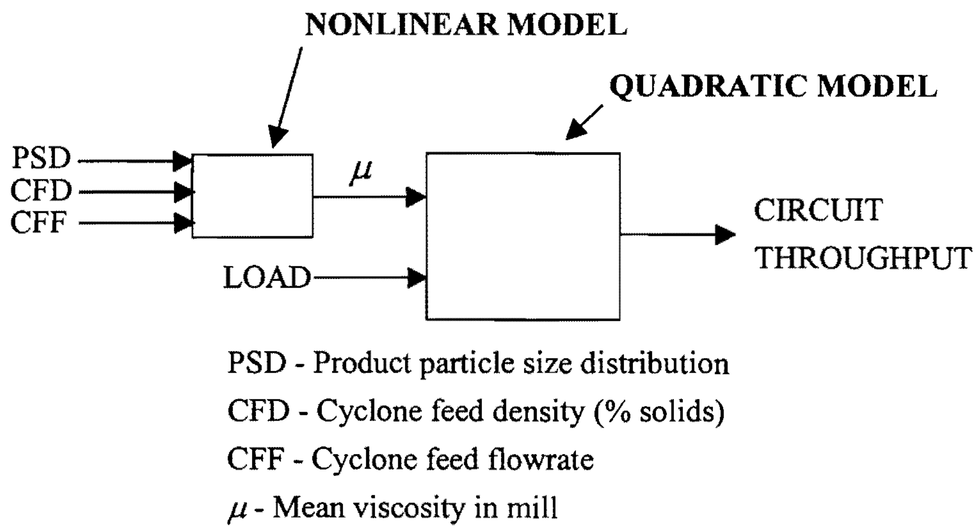


Figure 3-7 Two-model design used by model based optimizing controller.

Before such a model layout can be justified, it must be ensured that a more general, single model would not be better. The alternative design shown in figure 3.8.

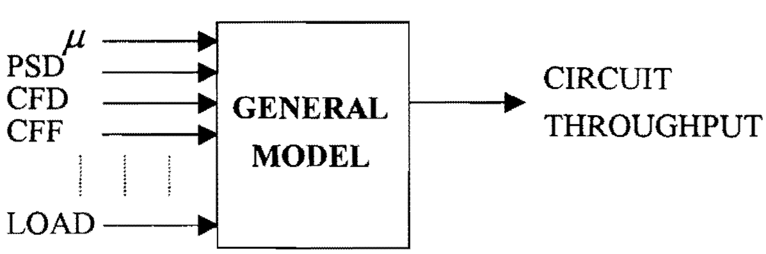


Figure 3-8 Alternative model design utilizing a single general model.

There are many motivations for the more general approach where all measurements are logged and a multi-linear regression or neural network type algorithm is used to generate a relationship between the inputs and outputs. Firstly, this method allows for the addition of further data and secondly, several setpoints could be optimized concurrently.

There are however several reasons why this method was not chosen for the course of this project:

1. General models tend to be complex and any intuitive understanding of the process is lost. The primary aim of this project was to gain insight to the process, and to use this insight to develop a control scheme. Such a controller requires little understanding of the processes involved and degrades to a mathematical exercise. (This does not mean that such strategies are inferior; they do have application and are certain to play a central role in the future of milling circuit control).
2. Simpler models are typically easier to tune, since they use fewer parameters.
3. The two-model design has the advantage that it can be made far simpler by obtaining a direct measurement of the viscosity exiting the mill. The nonlinear model would then fall away, leaving just a simple quadratic to relate the two remaining parameters to the throughput.
4. The two-model design will improve as more is understood about the relationships within a milling circuit. This control strategy is meant to be a step along the road to a controller that utilizes many highly specialized physical models in order to better control and optimize milling circuit operation.

3.3 Implementation of Strategy

To determine the workability of the strategy extensive testing was performed on a computer simulation program developed by Mintek. The first phase was identification, during which the plant response to changes in viscosity and load setpoint were observed. This is necessary to obtain a respectable first estimate of the throughput matrix **T**. The controller was then allowed to select a load setpoint based on this model. This setpoint was then compared with the actual optimum[†]. This

[†] The true optimum *is* known since computer simulation experiments are 100% reproducible; so a separate run may be performed in each case to seek the optimum using another optimization technique (bisection method in this case) to maximize throughput directly, which is not possible on a plant.

process is described now. A description of more thorough testing and comparisons with other controllers are given in the next chapter.

3.3.1 Identification

For this example, it is assumed that the model is unknown to start with and therefore needs to be identified. In examples in the next chapter, the on-line adaptation of the model is in place and the this identification process is only required for commissioning of the strategy, after which the model maintains itself.

During continued operation the controller actually performs this task automatically using a least squares fit of the **T** matrix as described earlier in this chapter. The commissioning would therefore involve an estimate of **T** (i.e. provide predictions of throughput for m x n viscosity-load pairs). After a while, the experienced operator may be able to make this estimate, but for this section plant data will be used to provide a good estimate.

The simulator was run continuously while load setpoints were varied through a wide range from 28 to 47% volumetric filling. This was repeated for several sets of feed characteristics. A graph demonstrating the throughput achieved for each of these conditions was constructed (see figure 3.9).

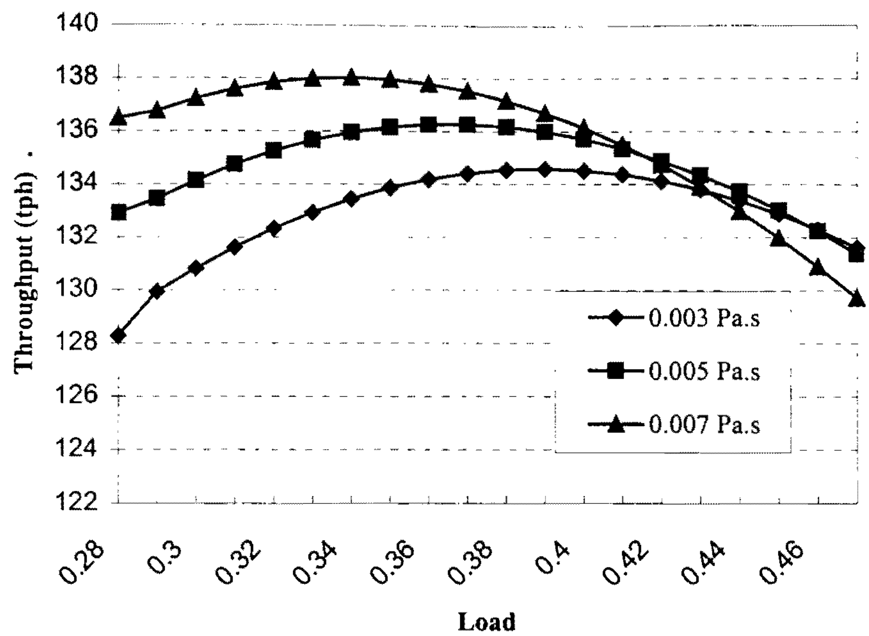


Figure 3-9 The effect of varying feed on throughput - load curve. Feed changes are reflected in 10% changes in viscosity in the mill load[†].

Figure 3.9 shows several aspects of grinding that may be anticipated. Firstly, it is seen how an optimum load setpoint exists for any particular feed. Secondly, it can be seen for this particular ore, how throughput decreases as viscosity decreases. These interactions are discussed in detail in section 3.2.1 (*Circuit Modelling*). Whatever the effect, the model is set to match the data, and is free to adjust to any possible quadratic shape.

The third effect that may be noted is that the position of the maxima of each load-throughput curve is different. This demonstrates that the use of a single load setpoint may result in a loss of efficiency for a variable feed. This dependence is a sensitive function of many factors and will differ from plant to plant, as well as the range of viscosities present.

Figure 3.9 can be used to obtain estimates for the matrix **T** which requires throughputs for three different loads (0.30, 0.35 and 0.40) and 5 different viscosities (0.003, 0.004,0.005,0.006 and 0.007 Pa.s). Interpolation between the curves supplies the required data (see table 3.1).

† Viscosity is a useful parameter for representation of the characteristics of the material within the mill since it relates closely to the PSD, percent solids and general rheology within the mill. Viscosity therefore reflects changes in particle size, changes in temperature and changes in density.

Table 3-1 Table of throughput achieved for various conditions.

	Load		
Viscosity (Pa.s)	0.30	0.35	0.40
0.003	130.8	133.9	134.4
0.004	132.8	135.0	135.0
0.005	134.1	136.1	135.7
0.006	135.9	137.0	135.9
0.007	136.1	137.9	136.1

This data can then be used to fit the quadratic using the method of least squares (described earlier in this chapter). The resulting equation is:

$$t = -7.00e1 + 1.05e3L + 6.03e4\mu - 1.37e3L^2 - 6.09e6\mu^2 - 3.14e5L\mu + 3.32e7L\mu^2 + 4.17e5L^2\mu - 4.57e7L^2\mu^2 \quad (3.23)$$

The resulting surface is shown in figure 3.10

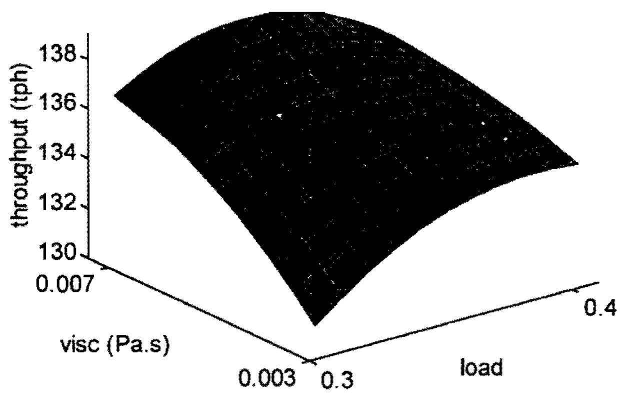


Figure 3-10 Plot of equation 3.23 demonstrating the shape of the relationship between throughput, load and viscosity for the computer simulated milling circuit.

Now a viscosity estimate would allow the computer to cut the surface to generate a simple quadratic relationship between throughput and load. The optimum of this function then corresponds with the desired load setpoint.

Viscosity Estimation

This would ideally be done using an on-line viscometer (see Shi and Napier-Munn, 1996b). Such a viscometer would supply an up to date estimate for of flow characteristics of material leaving the mill.

Alternative viscosity estimation

If this process equipment is not available, the viscosity of the slurry may also be estimated on-line as follows:

- slurry density and flowrate are measured from cyclone inlet stream
- the PSD of the product stream may be measured with an on-line particle size estimator
- a cyclone model (for example Plitt, 1976, see Chapter 2) may be used to back-calculate the PSD of the material *entering* the cyclone
- another model (e.g. Thomas, 1965, also described in Chapter 2) may be used to estimate the viscosity based on the PSD and density

An example of this process is now shown. The density of the stream was measured to be 1710 $kg.m^{-3}$ with a flowrate of 219.3 $m^3.hr$. The PSD of the product stream and the calculated PSD of the cyclone feed stream are shown in table 3.2.

Table 3-2 Size distribution information in the product stream and cyclone feed stream. The d_{50} for these streams (the particle size which divides the mass in half) is shown at the bottom.

Mesh Size (mm)	Cyclone Feed Stream (cumulative % passing)	Product Stream (cumulative % passing)
19.000	100.0000	100.0000
13.200	100.0000	100.0000
9.500	100.0000	98.98228
6.700	100.0000	98.21496
4.750	100.0000	97.52803
3.350	100.0000	96.78073
2.360	100.0000	95.89639
1.700	100.0000	94.86562
1.180	100.0000	93.41377
0.850	100.0000	91.69069
0.600	100.0000	89.31706
0.425	100.0000	86.22410
0.300	99.99865	82.14354
0.212	99.83871	72.15289
0.150	95.37971	52.35212
0.106	83.35769	40.13710
0.075	69.93187	32.26393
0.053	57.66500	26.18379
0.038	47.48365	21.41446
d_{50}	0.0417 mm	0.1415 mm

The d_{50} of the feed stream is used in conjunction with the measured density to estimate the viscosity of the material leaving the mill.

$$\begin{aligned} &\text{solids density} = 2730 \text{ kg.m}^{-3} \\ &\text{slurry density} = 1710 \text{ kg.m}^{-3} \\ &\qquad \therefore \% \text{ solids} = 41.1\% \text{ (by volume)} \\ &d_{50} = 141.5\mu\text{m} \\ &\qquad \therefore \mu = 9.0 \times 10^{-3} \text{ Pa.s (from figure 2.6)} \end{aligned} \tag{3.24}$$

Figure 2.6 is used here to demonstrate the procedure. The controller actually determines μ by cross referencing a matrix which contains the equivalent data. An improved version would contain a mathematical model to return viscosity.

It is worth noting that although this method of estimating viscosity is inaccurate and unproven, accuracy is not of great importance to the success of the control strategy. What is important is that the number found bears a direct correlation with the true viscosity (and is consistent). That is all that is required to ensure proper operation of the controller.

This assumption requires that the model is identified using the same relationships as are used to predict optimal setpoints; in the event of inaccurate viscosity estimates the model would identify with a certain bias, which would 'correct' errors made when reading load setpoint recommendations. This is coincidental, but is nevertheless useful to bear in mind.

Once the viscosity is estimated, whether by an on-line viscometer or by the calculation strategy described above, it is possible to move to the next stage.

3.3.2 Execution

The controller uses the modelled surface as follows:

(i) viscosity parameter estimated and substituted into equation reducing it to a quadratic

$$\text{where } \mu = 9\times10^{-3}\,\text{Pa.s}$$
$$M = -1318.7L^2 + 913.2L + 20.6 \qquad (3.25)$$

(ii) the load setpoint that returns the optimum throughput is then found simply from this formula:

$$\text{where } y = ax^2 + bx + c,$$
$$x_{extremum} = -\frac{b}{2a} \qquad (3.26)$$

for the simulation example this gives:

$$L_{opt} = -\frac{913.2}{2\times(-1318.2)} = 0.346 \qquad (3.27)$$

this load setpoint is then implemented.

During continued operation the 2 variable quadratic would be updated continuously to ensure improved accuracy and plant tracking. This is done by means of the throughput matrix **T** which would be modified based on filtered throughput data. The 3-D model would then be set to match **T** as closely as possible using a least squares fitting algorithm. This is described earlier in the chapter.

3.3.3 Conclusion

The load setpoint selected in this example was 0.349. This compares well with the true optimum 0.356 which can easily be determined by trial and error on the simulator (or by more sophisticated searching algorithms, such as the bisection method, or the Newton Raphson method). This process was repeated for several feed conditions; a listing of the results is shown in table 3.3. The controller has ensured throughput very similar to the ideal. The accuracy of this prediction could be improved still further by either using more data to generate the model or by using a higher order model. It must also be noted that the "curvature" of the fitted curve will vary from one plant to the next and the error resulting from a constant setpoint could be far more severe. The simulator's dynamics are actually fairly forgiving and the use of a constant setpoint only causes the loss of a few tons each day. Similar tests performed on a real circuit have shown an increase in throughput of as much as 10% (Craig *et al*, 1992).

Table 3-3 Comparison of throughput with new controller with "perfect" control and with no control. Here 0.35 is chosen as the setpoint for the constant load setpoint case.

	Load Setpoint via New Strategy		Load Setpoint Ideal		Load Setpoint Constant	
Viscosity (Pa.s)	Load Setpoint	Throughput (tph)	Load Setpoint	Throughput (tph)	Load Setpoint	Throughput (tph)
0.008	**0.332**	138.01	**0.328**	138.02	**0.350**	137.79
0.0085	**0.341**	137.10	**0.339**	137.11	**0.350**	137.01
0.009	**0.349**	136.23	**0.356**	136.26	**0.350**	136.25
0.010	**0.366**	135.29	**0.362**	135.28	**0.350**	135.20
0.011	**0.383**	134.55	**0.379**	134.57	**0.350**	134.18

It must also be noted that the benefits of modifying the load setpoint are more extreme the further from the nominal operating conditions the plant wanders. It is for this reason that this strategy may be of particular use to autogenous mills which suffer extreme feed disturbances.

Further experimentation is shown in the next chapter where a dynamic feed is used and states are estimated on-line using filters to counter the effects of noise. This allows the above algorithm to be executed repeatedly and allows for the model adaptation algorithm to be utilized. Comparisons will again be made with the true optimum and the optimum as chosen by the previous optimizing control strategy (the hill climbing technique).

Chapter 4

Implementation and Testing

4.1 Aim

In the previous chapter, it was shown how the proposed control strategy could suggest a load setpoint using a model of the plant. It was also shown that for steady-state operation, the setpoints chosen were very close to the 'real' optimum.

In this chapter, some modifications to the simulator are described first, followed by the controller tests and comparisons. The controller is tested under conditions closer to real operation, with repeated execution, incorporating noisy data, filtering and on-line model adaptation.

4.2 Simulator Modifications

In order to mimic operation on a real circuit, a pseudo-random binary sequence (PRBS) was utilized to add "white" noise to the mass feed rate and properties and to generate trajectories for feed PSD, hardness as well as the product size requirement.

For evaluation purposes, the throughputs attained with the new strategy are to be compared to the throughputs attained both with no control and with the previous optimizing control strategy that maximized *power consumption* with a hill-climbing technique (see Chapter 2).

To make the comparison fair, it was necessary to modify the simulation model computer program extensively; firstly it was necessary to allow for an 'independent' power consumption estimate. Previously the simulator model assumed that throughput was a direct function of power - the same assumption used in the derivation of the previous optimizing control strategy* (that of Craig *et al*, 1992).

This means that any strategy aiming to maximize power consumption *will* be optimizing throughput. To ensure that the hill climbing technique is on level footing, another model for power consumption is required.

It was also necessary to add code to the simulator to simulate a time-varying and noisy feed. The controller is designed to filter data and these modifications will not only improve the reality of the simulations but will test the filtering algorithms. These modifications are explained in the following sections.

4.2.1 Feed Stream Manipulation

In Chapter 3, the new model based optimizing control strategy was tested on a computer simulation of a grinding circuit. In that section, the feed was constant and no disturbances were present. As a result all system states were constant and could simply be used directly by the controller.

On a real plant this is not possible since readings do not reflect the whole truth about states. For example, if you wished to determine the height of the sea tide it is insufficient to simply measure the level of the water at one instant, since waves are constantly changing the level. This does not mean that there is no level but simply that it is masked by noise.

The same is true for many of the states within the mill - the mass of the mill, the flowrates in various streams, the PSD in the product, etc. Therefore, to fully represent a real plant, it is necessary to simulate the noisy behavior of the plant states *and* to apply filtering techniques to the signals to return a true representation of the states to the controller.

For this simulation, it was decided to use a pseudo random binary sequence (PRBS - a sequence of 1's and 0's randomly arranged - see figure 4.1) to add noise to the feed rate. Pseudo random numbers in the range [0,1] are then used to alter the mean feed

* Equation 2.28 was used to estimate power consumption previously. Throughput was then calculated using breakage and selection functions - the selection function (equation 2.18) is *directly proportional* to the power consumption.

properties in the range [-30%, +30%] of the nominal values. The new stream would typically form a function of the old stream as follows:

$$x_i = (Bc_1 + c_2)y_i \tag{4.1}$$

where x_i = new value for ith element of data stream

y_i = previous value ith element of data stream

B = PRBS – either 0 or 1

C_1 *and* C_2 = constants

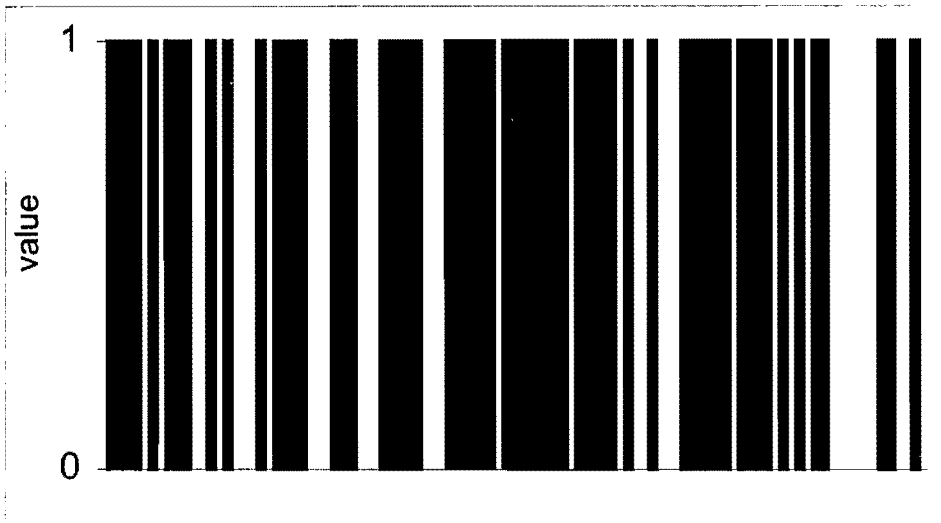


Figure 4-1 A pseudo random binary sequence (PRBS). Such a sequence is used to manipulate inputs to the milling circuit simulation

The feed conditions are changed by altering the PSD and the hardness. The PSD is altered by changing coefficients in the equation used to represent ROM ore:

$$y = 100\left\{\Theta\left(1 - \exp\left(-0.6931\left(\frac{d_i}{x_{50}}\right)^{\varphi}\right)\right) + (1-\Theta)\left(1 - \exp\left(-0.6931\left(\frac{d_i}{d_{50}}\right)^{\beta}\right)\right)\right\} \tag{4.2}$$

where y = mass % rock less than mesh size d_i

Θ = proportion of fines resulting from abrasion

β = distribution modulus for material (excluding material arising from abrasion)

φ = distribution modulus for material arising from abrasion

d_{50} = median size for material (excluding material arising from abrasion)

x_{50} = median size of material arising from abrasion

This equation is a weighted sum of two Rosin-Rammler functions that adequately describes material that has been trammed and hoisted to the surface (Hinde, 1985).

To alter the d_{50} of the feed, the d_{50} and x_{50} in this equation must be varied. The shape can be effected by manipulating the remaining parameters.

The hardness of the ore is modified by altering the selection function. The selection function determines the fraction of each size class that is 'selected' for breakage each iteration of the simulator. *Increasing S* (the selection function, see section 2.9.1) has the equivalent effect on the simulator as a *softer* feed, and vice versa.

A third disturbance that may be considered is a change in the product size setpoint. The setpoint may be changed periodically depending of the downstream requirements†.

Trajectories for these inputs are generated by the simulator during operation. Each input is set as a function of time - steps are triggered using

```
If time = x Then input1 = f(input1)
```

where '*input1*' is the input data-stream in question and *f* is a specially designed function. Periodic events are triggered using

```
If (time Mod x) = 0 Then input1 = g(input1)
```

where g is another specially designed function. Noise is added by repeating this type of event at a high frequency, where *g* utilizes the PRBS described previously.

4.2.2 Parameter Estimation

Once noise has been added to the simulation to ensure that measurements more closely mimic those available on a real plant, new methods of obtaining suitable information from these readings are required. On a real plant the readings typically available are:

- sump level
- mill power draw
- mill mass
- density in cyclone feed stream
- PSD of product stream (or pressure in cyclone feed stream)

† See section "Further possibilities" at the end of this chapter.

- water addition rates

It is assumed these readings are available continuously. To reduce the effects of noise, it is necessary to filter the data prior to use by the controller. The Kalman filter is described in Chapter 2. This algorithm is used in the controller code to obtain smooth values for each parameter. When the controller requires readings, it will be given a single value that is extracted from data recorded over a period of time prior to the call.

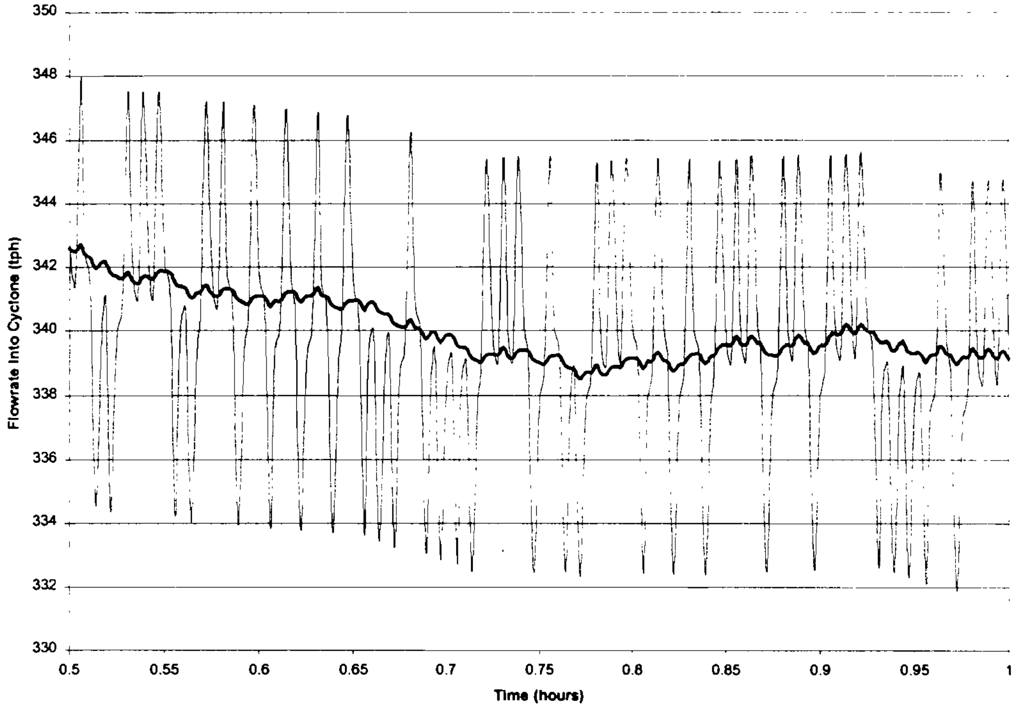


Figure 4-2. Plot demonstrating the effect of the filter on noisy data.

4.2.3 New Model for Power Consumption

The model chosen was proposed by Kapur *et al* (1992). They propose to estimate power consumption based on physical considerations; torque, slurry motion etc.

Relationship Between Mill Load, Mill Throughput and Power Consumption

The power consumption is determined exclusively by what is required to maintain the preselected mill speed. Therefore the factors that would tend to reduce mill speed and bring on the need for more power from the mill motors must be examined. The most direct power requirement is related to the lifting of the ore. This gives potential energy to the load. This energy is then transferred in several ways - friction (to heat) and breakage, deformation (to heat) as well as via momentum transfer - both within the load and back into the mill lining.

Models can be used to illustrate the energy cycle. Assume we have two particles within a rotating cylinder (consider the two dimensional case - two polygons (of finite mass) within a turning circle (see figure 4.3). As the circle rotates the polygons will experience a frictional shear force which will tend to move them up the side of the circle. As the motion is rotational, there is constant acceleration towards the center of the circle. As the objects move up the side the reaction force between the object and the mill wall becomes less (gravity is acting at an angle to the surface). The frictional force lifting the particles is doing work, and hence transferring energy to the particles. Any slippage that occurs results in energy transfer both to heat and the generation of new surface area (attrition). When the particles reach 90 degrees from the bottom, the reaction force is entirely as a result of centripetal forces, which depend on the mill speed. After this point gravity works against the centripetal forces; at some point (assuming the mill is operating at less than critical speed) the objects will fall off the perimeter and across the circle, their potential energy being transferred to kinetic energy as they fall (Davis, 1919).

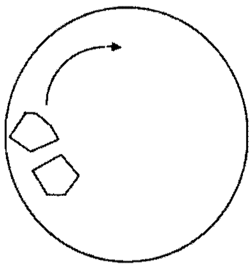


Figure 4-3. Two objects in a rotating circle.

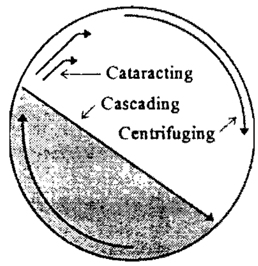


Figure 4-4. Flows within a tumbling mill.

As a result of the random shapes and frictional coefficients, as well as the interaction of the one particle with the other, they may fall at different times and in different directions. Indeed, most particles in a real system are completely surrounded by other particles and their path is a complex function of the paths off all the surrounding particles (Nates *et al* 1994a&b).

When the particles reach the perimeter after their "fall", energy is transferred in several ways. Not only is the energy spent in the propagation of cracks, but on deformation of the polygons, noise, etc. It must also be realized that the particles are not likely to land near the bottom of the mill and they may still possess a certain amount of potential energy. For example, if the particle lands on the far side of the mill, say a third of the way up - it may then transfer its remaining potential energy back to the mill it nears the bottom once more.

In order to relate the load to power draw effectively, many have resorted to complex models and simulations of the process.

The simulator may simply have a certain number of randomly shaped particles being turned in a cylinder, or in the simplified 2-D case, a circle. Parameters that could be varied are the particle size and shape distributions, % solids, mill speed (as a fraction of the critical speed) and particle hardness (Mishra and Rajamani, 1993*). The computational requirements have however proved prohibitive.

The alternative is to treat the load as a continuum and model the mass transfers as such. One such model is described by Kapur *et al* (1992). Their charge flow model for power draft in tumbling mills treats the complex charge motion by coupled cascading cataracting flows.

Power Draw - Multi Torque Model

Kapur *et al* realize that the model of Davis (1919) described earlier, where particles fall through the mill space unhindered by other particles, is highly idealized and restrictive and have developed their model to cater for the effects of interference.

In their approach, the problem is avoided by treating the mill load as a continuum. The compact load is tilted until the dynamic angle of repose (critical angle) is reached and the charge begins to roll, that is cascade down the free surface. It turns out that reasonably accurate expressions for mill power can be derived in the case of the cascading regime as a function of the four primary components which determine the power draft in the first instance, namely, mill size and speed, and it's fitting ratio (essentially a constant representing the combined effects of lifter bar geometry etc.) and charge weight.

There is, however, no reason to expect that the mill charge should invariably move in a cascading mode. Effects that cause the complex and sometimes erratic cataracting include:

- Friction within layers of the tumbling charge, including locking of rotating balls by coarse particles and slippage between charge load and the mill wall.
- Mixing, separation and redistribution of the charge constituents, including sieving and percolation of coarse particles.

* This highly computer-power intensive discrete element method of modelling multi-body collisions has yet to be validated.

- Opening and expansion of the charge bed (by lifter bars or as a result of high speeds).
- Layering of thick pulp to form a more (or less) stable mill lining with or without entrapment of grinding media.
- Centrifuging (material is stuck on the outer perimeter) of the outer layer as a whole at high mill speeds.

The changing character of the pulp rheology as the particle size becomes finer (for a batch run) can give rise to various kinds of highly diverse and highly time dependent mill torque profiles which may even appear random (see Figure 4-5).

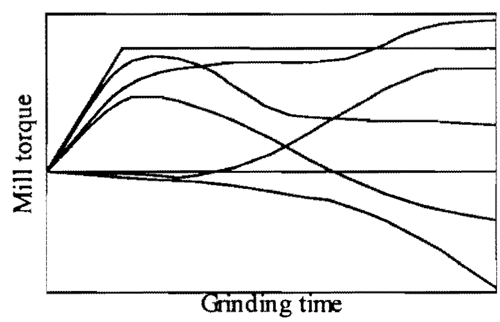


Figure 4-5. Some time-torque profiles reported from the literature (from Kapur *et al*, 1992).

Kapur *et al* quote Pietch (1972), who proposed that the most important of the three distinct terms that comprise the total energy consumption, was the variable sticking effect due to pulp viscosity which causes the media to clump together as well as stick to the mill liner. This, he claimed, was a major cause for the break up of the rotating charge into a cascading pool and a cataracting portion.

Using this research in combination with observations of pulp flow in a glass sided laboratory mill, Kapur *et al* were able to successfully model the power draw (or torque) by separately considering the torque requirements for three different regimes, namely the cascading, cataracting and centrifuging (see Figure 4-4).

The power draw for the cascading fraction is merely calculated from the geometry (see Figure 4-6).

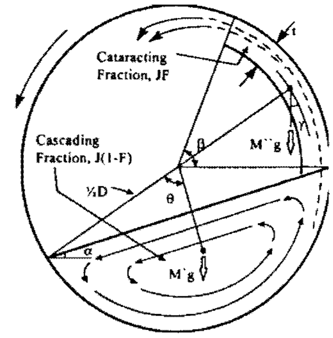


Figure 4-6. Split of tumbling charge into cascading and cataracting fractions (from Kapur *et al*, 1992)

The torque required by the cascading material is given by (Arbiter and Harris, 1982):

$$T_{cas} = M_{cas} gh \sin\alpha \tag{4.3}$$

where M_{cas} is the mass of the charge in the cascading regime ($M_{cas}=M(1-F)$, where M is the total load mass), g is acceleration due to gravity, h is the distance of the center of gravity of the charge from the center of the mill, and a is the dynamic angle of repose of the tumbling load.

From geometric considerations alone,

$$h = \frac{D\sin^3\theta}{3(\theta - \sin\theta\cos\theta)} \tag{4.4}$$

where D is the inner diameter of the mill and θ is half the angle in radians subtended at the mill center by the mill charge at rest. The fraction of the mill volume occupied by charge is:

$$J = \frac{\theta - \sin\theta\cos\theta}{\pi} \tag{4.5}$$

Combining equations 1, 2 and 3 yields an alternative expression for the cascading torque:

$$T_{cas} = \frac{1}{3\pi J} M_{cas} gD \sin^3\theta \sin\alpha \tag{4.6}$$

Using the following approximation by Kiyama *et al* for θ (from Kapur *et al*, 1992),

$$\sin^3\theta = \frac{9\pi}{7} J(1-J) \tag{4.7}$$

the equation can be rewritten in the form:

$$T_{cas} = \frac{3}{7} M_{cas} gD(1-J) K \sin\alpha \tag{4.8}$$

Defining $J_{cas}=J(1-F)$ and using $M_{cas}=M(1-F)$ gives:

$$T_{cas} = \frac{3}{7} MgD(1-F)[1-J(1-F)]K \sin\alpha \tag{4.9}$$

For the cataracting fraction, the torque requirement was calculated by the drag caused by this fraction on the mill lining (the braking action):

$$T_{cat} = M_{cat} g\left(\frac{D-t}{2}\right)\cos\gamma \tag{4.10}$$

where t is the average thickness of the cataracting layer and the cataracting mass is $M_{cat} = MF$. This simplifies to:

$$T_{cat} = MFg\left(\frac{D-t}{2}\right)\sin(\alpha+\theta+\beta/2) \tag{4.11}$$

where t can be estimated from the following procedure. First, consider the case where all charge is uniformly stuck to the mill liner, in which case:

$$M = \frac{\pi}{4} D^2 L d_c \left[1-\left(1-\frac{2t}{D}\right)^2\right] \tag{4.12}$$

where L is the mill length and d_C is the charge density. The charge mass is approximated by:

$$M = \frac{\pi}{4} D^2 L J d_c \tag{4.13}$$

Equating (11) and (12) and then solving for t gives the following:

$$\frac{t}{D} = \frac{1}{2}[1-(1-J)^{1/2}] \tag{4.14}$$

and another result of the uniform layer thickness assumption is:

$$\beta = 2\pi F \tag{4.15}$$

giving, finally:

$$T_{cat} = \frac{1}{4} MFDg[1+(1-J)^{1/2}]\sin(\alpha+\theta+\pi F) \tag{4.16}$$

where θ is given by:

$$\sin^3\theta = \frac{9\pi}{7} J(1-F)[1-J(1-F)] \tag{4.17}$$

Modifications are available to cater for centrifuging include the use of modified mill diameters and load masses.

It is now possible for values of F, K and α to be extracted by fitting the model to data from torque tests (using equations 7 and 14).

Discussion

Kapur *et al* tested the model using data from several sources and the values extracted agreed well with intuition. For example, it was shown that the dynamic angle of repose (α) increases for decreasing particle size. This is intuitive since smaller particles give the slurry a higher superficial viscosity causing greater resistance and therefore a greater lifting effect on the load.

In order for the model to fit the test data, it was also necessary to adjust the fraction of material in the cataracting mode. Cataracting was shown to be greater for more viscous slurries (finer particles) as well as for higher mill velocities.

With regards the power consumption, Kapur *et al* noted the following:

- A peak is observed in the power consumption as a function of F (the fraction in cataracting mode). The power consumption was generally highest at $F \approx 0.2$.
- During batch testing, it was noticed that power consumption reaches a stable peak when there was a mixture of large and small particles in the slurry.
- In further testing it was noticed that power consumption reached a peak as a function of mill speed.

The fraction of the load in the cataracting mode is an important parameter in power draw considerations. The peak at $F \approx 0.2$ comprises the power drawn by each of the flow regimes.

There is no peak in power consumption by the cascading material since as F increases, the mass of material remaining decreases and the energy transfer must therefore decrease.

Naturally, when $F = 0$, the power draw by the cataracting fraction is zero (no cataracting is occurring). As F increases the power draw by the cataracting fraction increases, but this effect is countered by equation 6 which shows that power draw decreases with F. The result is a peak - see figure 4.7.

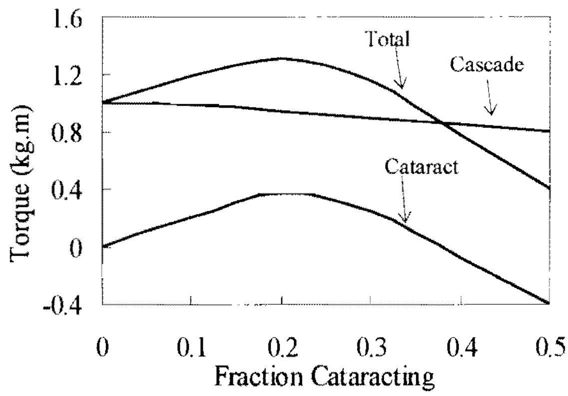


Figure 4-7 The torque requirements as a function of *F* (from Kapur *et al*, 1992).

This model was identified for the simulator and implemented to produce an on-line estimate of the power consumption. The model error was minimized by minimizing the sum of square errors (as compared with existing power estimation model) in a certain range of operation (±10% from nominal for feed d_{50} and feed hardness; 3^2 cases). This power consumption estimate is used by the hill-climbing optimizing controller in the next section.

4.3 Tests

In order to demonstrate the operation of the new optimizing control strategy it is necessary to demonstrate its operation under plant conditions (or at least as close to plant conditions as is possible with a simulator).

The next step is to rate the performance in some way. This is typically done by comparing performance with existing strategies. It is however appreciated that a simulator is not ideal for this task since manifold subtleties are not represented. It does however present an economical (and convenient) alternative.

4.3.1 Aim

The primary aim is merely to demonstrate the strategy in operation in the simulation environment designed for it. Week long plots showing flowrates, control actions, model evolution etc., will be shown.

The most interesting aspect of this stage is the evolution of the 3 dimensional model relating throughput to load and viscosity. As the system operates normally, the controller observes throughput and modifies the model to match the data.

The second aim is to investigate throughput optimization by comparing performance with other optimization methods. This will serve to expose situations where the proposed strategy outperforms existing strategies. This leads to the discussion in which combined strategies are proposed.

4.3.2 Method

The simulator was set up to run for a week (168 hours). This involved the design of week long trajectories for each of the following variables:

- feed hardness
- feed particle size distribution
- product size requirement

Each was designed by a combination of heuristics and then modified by a random factor (a "random walk" variable is added to the suggested value).

It is then possible to examine economic performance relative to other strategies. This is done by operating the same system with the same a set of input trajectories (designed to represent a week of typical operation) using each control strategy in turn.

The total (or cumulative) throughputs would then act as a simple gauge to the relative performances of each strategy. It will also be possible to examine how each strategy coped with various events that were built into the week-long trajectories[¤] .

It must be pointed out that although the final aim is to increase throughput, it is also possible to gauge success in other ways. For example, if the optimal throughput for a certain set of conditions is known (for a simulator this is always possible to determine), then the cumulative loss in throughput would also be a gauge of performance. This gauge is used extensively in the next section.

4.3.3 Results

Nominal operation

Nominal operation is defined here for convenience as operation where the *mean* value of each input to the system is constant. That is to say that the inputs do have variation associated with them, however no permanent changes are made. The conditions are summed up in table 4.1, while size distributions of the feed and product are given in table 4.2.

[¤] If different event-types are better handled by different strategies it opens the way for a "what if" type strategy where different strategies are used at different times depending on the current conditions. This is discussed later.

Table 4-1 Table showing expected values for several parameters during nominal operation.

Parameter	Value
Feed % Solids	96.7 %
Feed Density	2.73 t/m^3
Product % Solids	16.6 %
Product % <70 μm	70.0 %
Mill Load	0.350
Charge Mass	246.1 tons
Mass Throughput	136.3 tph
Mill Discharge Rate	375.9 tph
Mill Discharge % Solids	65.7 %
Cyclone: d_{50c} (cutsize)	119 μm

Table 4-2 Table showing particle size distributions in feed and product for nominal operation.

Size Class (μm)	Feed (% passing)	Product (% passing)
19000	100	100
13200	29.46	100
9500	25.06	100
6700	21.62	100
4750	19.18	100
3350	17.37	100
2360	15.99	100
1700	14.92	100
1180	13.83	100
850	12.85	100
600	11.78	100
425	10.68	100
300	9.54	100
212	8.42	99.84
150	7.35	95.46
106	6.33	83.45
75	5.41	69.99
53	4.58	57.7
38	3.88	47.5
d_{50} (μm)	~14900	40.0

Naturally for the case where there in no permanent change in the system, there is no supervisory control required, other than to get the setpoints to the optimum and to leave them there.

It is also noted that serious changes to the plant, such as a feed stoppage or sump overflow are not directly relevant to supervisory controllers which are non-essential to continued operation. Supervisory control may be put on "hold" until such events are cleared.

It is worth mentioning that events that require postponement of supervisory action may have an effect on the model use in the model based optimizing controller. For example, if supervisory control were postponed, and during this period the plant changed it's mode of operation quite dramatically, the model would be obsolete when the supervisory control is returned, and actions may be counterproductive. It may be worthwhile to allow the controller a period in which to adapt it's model to the new circumstances. It may also be worth reverting to another strategy such as the hill-climbing method (Craig *et al*, 1992) during this period.

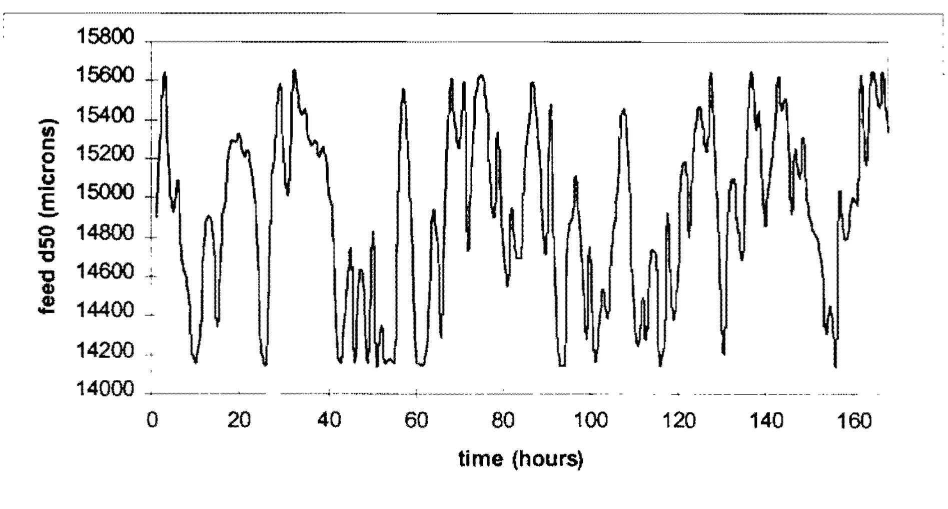
Illustrative Tests

This section is dedicated to the demonstration of the model based optimizing controller and several of its features. Tests are therefore designed to bring certain features to the fore.

The first test is the one week test aimed at demonstrating typical operation.

One Week Test

Plots of input and setpoint trajectories used in the test are given in figure 4.8. It must be noted that although changes in the product size requirements are uncommon, it is likely, in the face of increased competitiveness, that downstream processes may extract benefit from greater control over input stream properties.



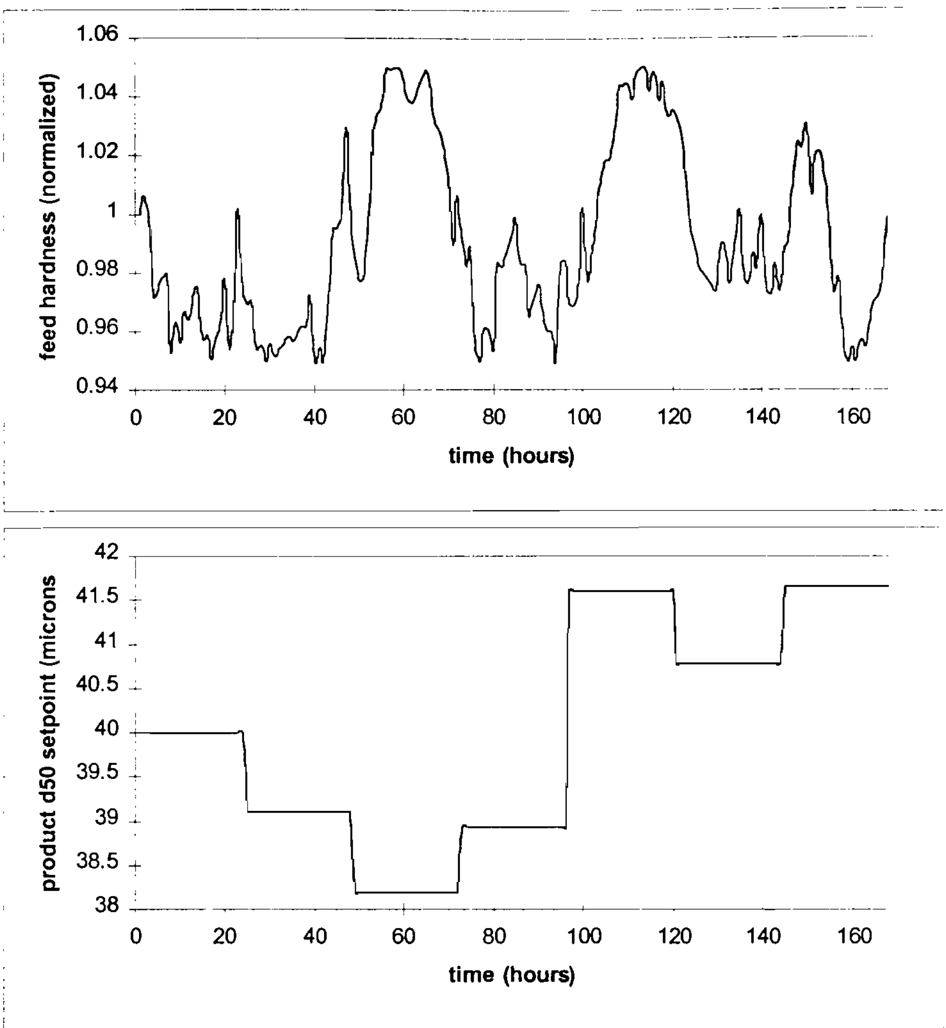


Figure 4-8 Plots of input trajectories for 1-week test.

The actions taken by the controller are shown in figure 4.9. It is worth noting that after the initial learning period, the magnitude of the setpoint changes are considerably smaller than those made by the hill-climbing technique, for which the size needs to be selected to both be large enough to track quickly and small enough to avoid excessive control action during stable action.

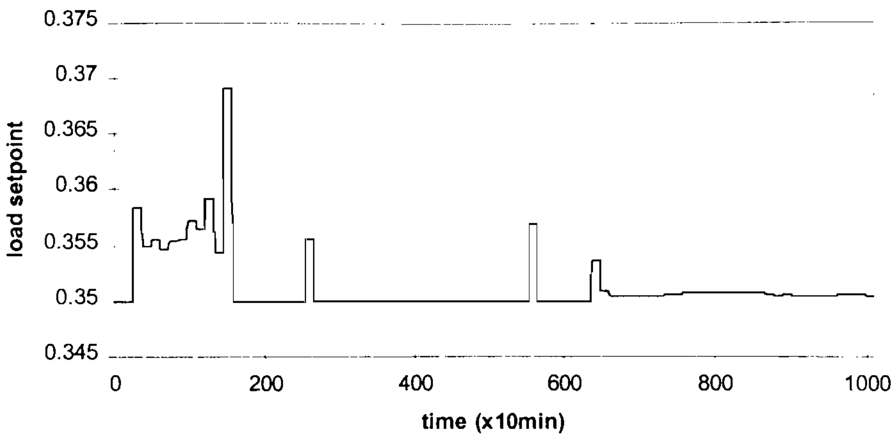


Figure 4-9 Setpoint track for model based optimizing controller for one week of operation.

It is also noticed how, as time passes, the control gets tighter (i.e. operation is more stable and very little control action is required) - this is due to the model evolution which is learning during operation. The plant was started with a guess for the model (a flat plane) which evolved as demonstrated in Chapter 3. The period of strong action coincides with the controllers first experiences of various sets of conditions.

Examining the input trajectories reveals that the plant experiences similar input conditions on several occasions after this period, during which the controller is able to select suitable setpoint with the minimum of control action.

Comparative Tests

The plant was then run three more times using the same set of input trajectories (the 168 hour plots in the previous section); the first time with no control, then with hill climbing optimizing control and then a modified optimizing controller (see description later in this chapter).

Success can be gauged by throughput. For the computer simulation model throughput is not a strong function of load as is true in many 'real' grinding circuits and therefore the fractional improvement is small. For this reason, the size the throughput improvements are therefore not a true representation of what may occur on a real plant. Far more important is the qualitative value of the throughput, which shows that throughput improvements *are* possible and are likely to be far greater on plants where throughput is a strong function of load.

For a simulator it is possible, for each set of conditions, to determine the ideal setpoint by trial and error and it is also possible to calculate the maximum possible throughput, assuming the controller operated ideally♣ . This allows us to calculate the loss in throughput incurred by each strategy by subtracting the throughput attained from the maximum possible value. A plot of cumulative loss in throughput for the week is shown in figure 4.10.

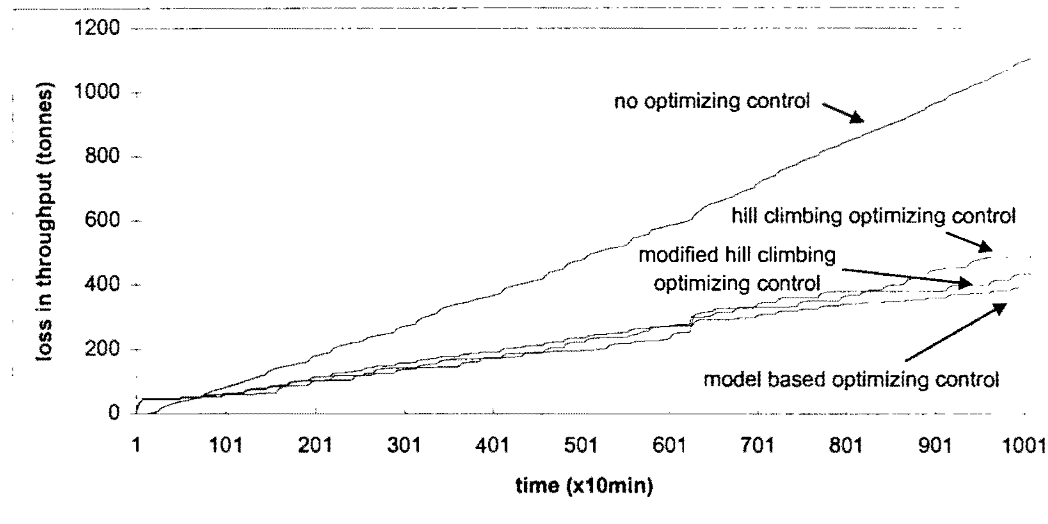


Figure 4-10 Plot of cumulative loss in throughput for 1 week test.

This effectively proves, for the case of the circuit simulator, that the model based strategy, after an initial 'learning' period can compete with the previous strategy and even perform better in some circumstances.

To illustrate what causes the above throughput improvement, it is useful to examine the load setpoint directly.

It is logical, that for any particular set of conditions, there will be a load setpoint that will optimize throughput. If this setpoint were known *a priori* it would then be possible to grade the performance of the optimizing controller be examining how far from this optimum the controller setpoints are placed.

For the case of a simulator, it *is* possible to determine the optimum load setpoint, simply by trial and error. Therefore, it is possible to run the plant at one set of conditions, with the controller in action, and then switch to a second set of conditions, where the true optimal load setpoint is known. It is then possible to examine how the

♣ Based on nominal parameter values with no noise.

optimizing controller alters the load setpoint; how fast does it move to the known optimum, how close does it get, does it oscillate, etc.

This test was performed for 3 cases; with a hill climbing optimizing controller, a modified hill-climbing controller* and the model based controller.

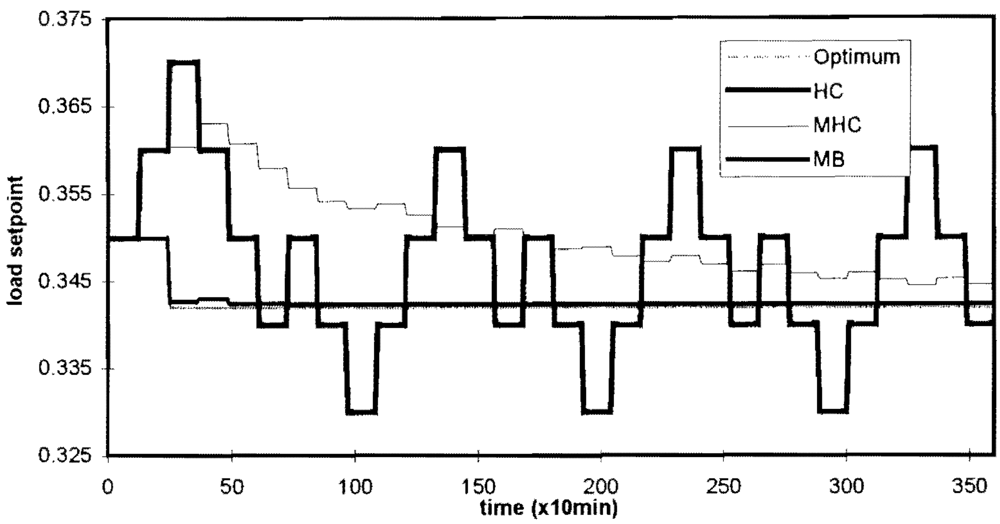


Figure 4.11 Plot of load setpoint after a step change in plant conditions. KEY: HC (Hill climbing strategy), MHC (Modified hill climbing strategy) and MB (Model based strategy). 'Optimum' shows the previously calculated optimum load.

This plot reveals some the reasons why the hill climbing technique loses throughput. Firstly, it is noted that the size of the step changes is large and almost swamps the required change; secondly, the setpoint continues to step continuously, never staying close to the optimum. Although the mean value approximates the optimum to within 0.25%, the true value varies as much as 1.5%.

It is true that the throughput-load curve is fairly flat near the optimum and that errors of less than 1% are practically negligible; however, with errors greater than this, the cumulative loss in throughput can become meaningful.

The modified hill-climbing strategy removes the strong oscillations by making smaller steps (the strategy is described in the next section). This strategy also seeks the optimum very accurately, provided mill operation is at steady state. The price is paid in terms of slower tracking and greater sensitivity to minor disturbances.

The model based optimizing controller is seen to perform remarkably well in this test. This is due to the fact that the model fits the plant very well at the given set of

* see next section (discussion: patches for hill climbing technique) for description of the modifications

conditions (the model had already run at those conditions during model-fitting and therefore could 'remember' the ideal setpoint). Once fully installed and fitted, this ought to be the case for any set of conditions that had been previously experienced.

4.3.4 Discussion

There are several aspects of the results that require comment. Each strategy will be discussed in turn.

No Supervisory Control

Using a single load setpoint results in throughput losses during periods where conditions are far from nominal ("design" conditions), such as when large disturbances hit the system, or when inputs remain far from their design values for extended periods of time.

Hill Climbing Method

Firstly it is noticeable that during periods of very minor input change, the hill climbing technique continued to adjust the load setpoint and hence oscillate about the optimum, resulting in some loss in throughput.

The strategy relies on a steady state assumption: that power consumption at any one time is a direct function of the mill load *setpoint* at a particular time. This is not necessarily true when load conditions are at unsteady state (i.e. the load does not match the load setpoint). If the controller increases the load setpoint, the new feed rate will increase accordingly (PI controller in this case) to increase the load. It may take several hours before the load reaches the setpoint and, to be properly effective, the hill climbing strategy must necessarily wait until this is achieved. In the face of other disturbances this may take a widely variable period of time, or may in fact never occur.

The current strategy just selects a set period of time and assumes that the load and load setpoint are reasonably in agreement.

This hill-climbing strategy is therefore limited in how frequently it can alter the setpoint.

Some possible 'patches' are now described.

Patches for Hill-Climbing Strategy

In order to prevent the controller from perturbing the system unnecessarily, it is possible to weight the change in the setpoint depending on the *magnitude* of the change in power consumption.

At present, the controller makes a step of say 0.01 in the load setpoint each time it executes. Often, however, the power has only changed fractionally, resulting in a basically unnecessary setpoint change.

If the setpoint change were a function of the change in power then the setpoint change could be made small when only small changes are found in the power consumption. For bigger changes in power consumption, bigger changes in setpoint could be made.

A hybrid of a fixed change and a weighed change was designed for testing. The size of the step was determined as follows:

$$\text{change} = 0.001(1 - e^{-abs(power - power\_old)}) + 0.002(abs(power - power\_old))^{0.4} \qquad (4.18)$$

This formulation ensures a minimum limit on the setpoint change of 0.002. This prevents the step decreasing to zero, which would prevent the mechanism from performing normally. The code for implementing this method is supplied in the appendices. The effect may be illustrated diagrammatically:

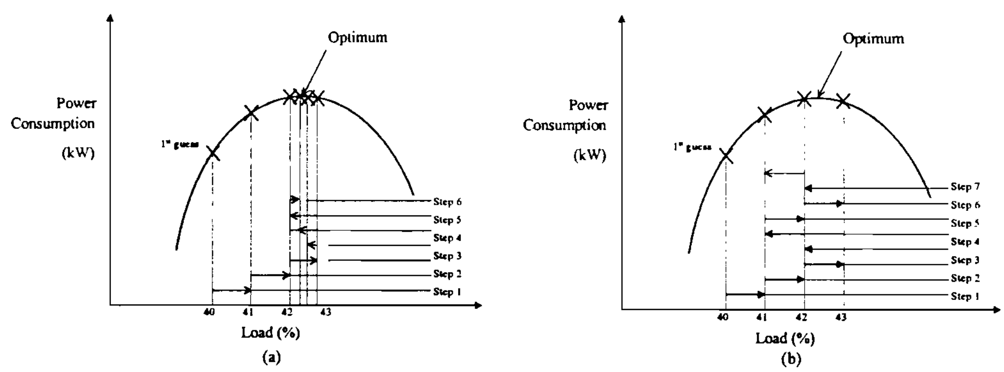


Figure 4-12 Comparison of hill climbing techniques: (a) Modified strategy goes to optimum successfully. (b) Previous strategy takes steps of set size - oscillates about the optimum.

An example of operation is shown in figure 4.12(a). For comparison, the equivalent response from the unmodified controller is also given (figure 4.12b). The modified controller is observed to track quickly to the point of maximum power consumption with a minimum of control action.

NOTE: It must be remembered that the action of the unmodified hill-climbing strategy depends strongly on the size of step chosen. It is possible to select a set stepsize to suit a particular circuit - a large step can be used on a circuit with great variations in feed characteristics, while smaller steps can be used on circuits where the characteristics of operation drift slowly with time.

A compromise does however result; larger steps may mean quicker tracking, however, oscillation about the optimum is increased.

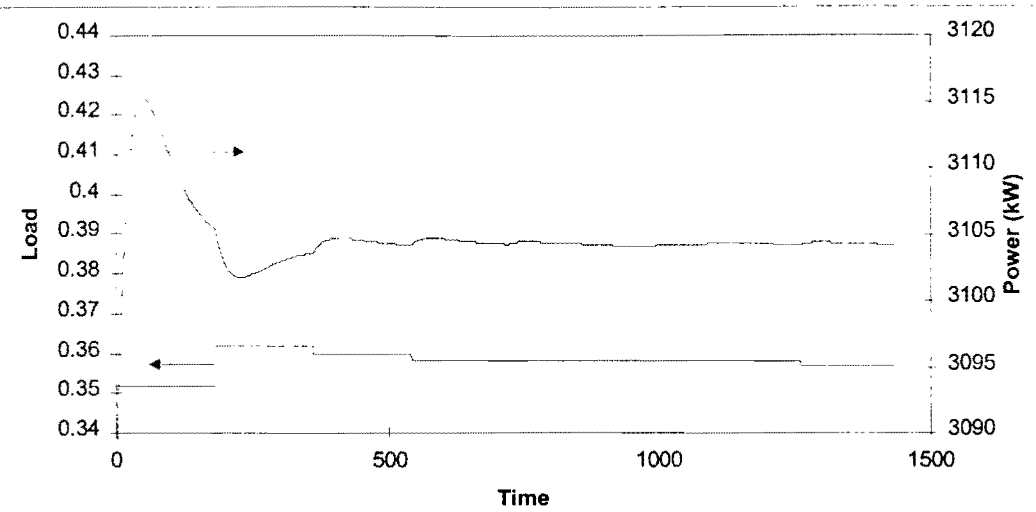


Figure 4-13 Plot of load setpoints and power consumption for the modified power-optimizing, hill-climbing method.

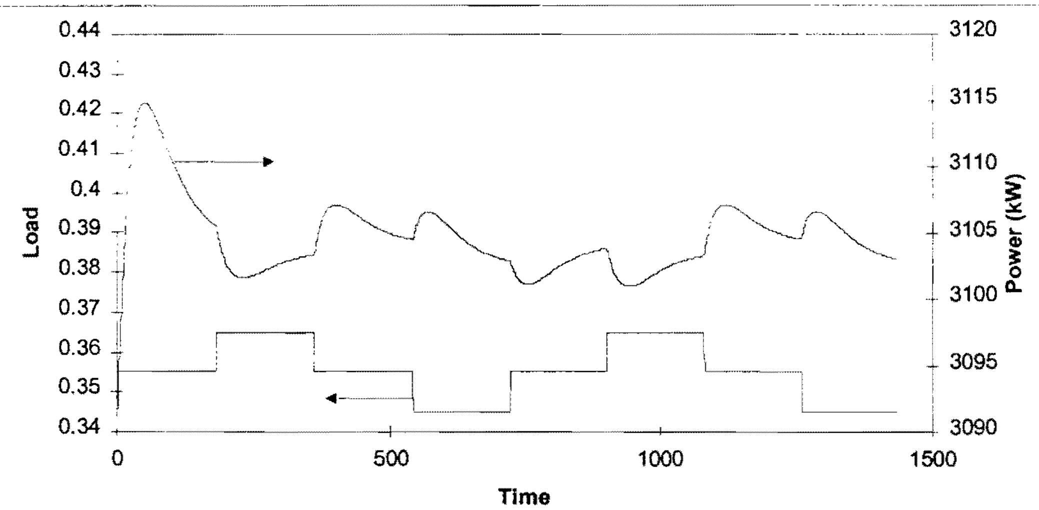


Figure 4-14 Plot of load setpoints and power consumption for unmodified power-optimizing, hill-climbing method.

Another patch that may improve the hill climbing strategy involves monitoring the load. After an adjustment to the load setpoint is made, the controller typically waits a few hour before re-evaluating the situation. The period taken is selected in order to allow the plant to return to steady state, or to put it another way, to allow the power consumption a chance to once more reflect the conditions in the mill. It is in the interest of throughput optimization however, to repeat the evaluations as frequently as possible, resulting in a form of compromise. The questions that arise are as follows:

1. how long does the circuit take to return to steady state?

2. how much benefit is derived from increasing the frequency of the optimizing control actions?
3. how would throughput be effected if setpoints were re-evaluated before steady state was achieved?

The questions are the sort that have definitive answers. A possible compromise strategy may suffice until the processes can be better understood. The controller could be set up to monitor the load and determine when it has settled sufficiently near to the setpoint. It can then use the event as a trigger for control action.

The benefits of such a strategy are immediately apparent:

- frequency adjusts to suit plant; the longer the plant takes to reach steady-state the longer the controller will wait
- when plant is steady the frequency is increased
- the 'degree of steadiness' (how close the load is to the load setpoint) required to trigger control action can be optimized for a particular plant.

This strategy would go a long way towards answering the three questions posed above.

Combined Strategy

The two patches described in the section may have synergistic effects. The first allows for varying load setpoint step sizes while the second allows for a greater execution frequency during steady periods. The first strategy would tend to decrease the time required for steady state to return, while the second could take advantage of this and allow for further evaluations.

This concludes the discussion of the hill climbing technique.

Model Based Method

It is most noticeable that throughput is lost (this can be seen by direct comparison with the hill-climbing strategy) during periods where the model and the plant differ. This is seen to occur during periods of extreme disturbances (both transitory and permanent). In the case where the model is self adaptive, permanent changes are, after a while, captured by the model. Throughput is lost during this operation.

The success of the strategy is therefore seen to depend on the strength of the viscosity-throughput relationship. If a change in throughout occurs with no change in viscosity, operation will necessarily have moved 'off' the 3-D curve. This event is certain to occur to some degree in the face of certain events - particularly those transitory in type. For instance if a the cyclone feed pump malfunctioned or the feed chute choked, the throughput would drop instantly and this would have nothing to do with the viscosity in the mill. This event would constitute a deviation from the model.

If, however, ore characteristics changed, viscosity *would* change, even before the throughput was effected. This event would constitute a modelled disturbance.

Clearly, the controller then relies that all events which effect the viscosity, eventually effect the throughput *in a consistent way*. It is not possible to test this using a simulator and this would form the central theme in further tests on a laboratory or pilot scale mill. If this is proved to be the case, as is assumed in the derivation of this model, then the controller is certain to have unprecedented success. Should the correlation be found to vary, additional parameters may be necessary to improve the fit.

Comparison

The accumulated throughput clearly indicates the success of optimizing control in a general sense. It is also shown that the model based method is capable of outperforming the hill-climbing method.

It must be noted that this is a single result and must be viewed in this context. It is clear that it would be possible to design an input trajectory to favour the one or the other.

It is however clear that the model based method has areas of application. Demonstrating this was the aim of this section of the thesis.

"What if" Type Controllers

In the event that control strategy "A" handles one set of conditions better than control strategy "B" while "B" out performs "A" in another, then they may be room for a special controller with an algorithm that swaps from one strategy to another depending on current conditions.

This 'scheduling approach' may, for example, have application when the model based optimizing control strategy is used for nominal operation but is less during periods where operation ventures far from the nominal conditions. It may then prove more profitable to implement the "hill climbing" optimizing control strategy until either the conditions return to normal, or the model adapts to the new conditions (assuming that it is designed to evolve).

This scheduling approach is now applied to the same set of input trajectories as the individual strategies in the previous section.

The benefits of this method can be illustrated by a cumulative plot of the loss in throughput. This is given in figure 4.15.

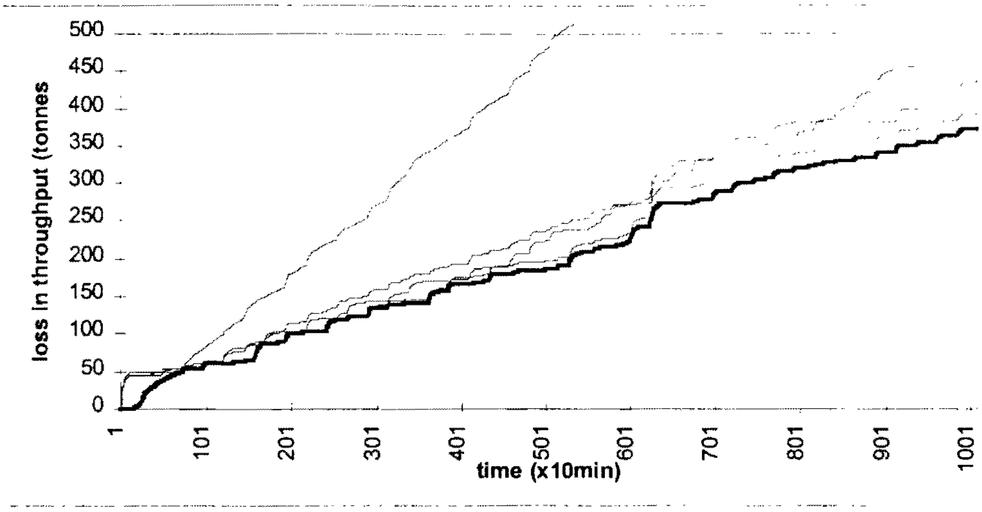


Figure 4-15 Plot of loss in throughput of several strategies as described elsewhere in this thesis. The dark line represents a controller that swaps from one strategy to another depending on which is performing better.

Naturally, much observation would be necessary to determine which circumstances benefit which controller, allowing it to be known when to swap to which strategy.

4.4 Further Possibilities

Complex, all encompassing controllers are becoming more prevalent as process data is becoming more available. High level controllers utilizing algorithms such as neural networks and fuzzy logic are able to determine improved setpoints without any analytical models, based simply on large volumes of process information. Stange *et al* (1997) provide a good example of a hybrid control framework based on agent and blackboard technology, utilizing fuzzy logic to optimized setpoint selection.

A vision for controllers in the future may however take the form of a combination of these controllers combined with plant models. Controllers could then be further improved as the processes become better understood.

Model based optimizing control is a step in that direction; firstly just monitoring one setpoint, such as the load, but eventually growing to take advantage of all available free variables.

In the case of milling, the process states that are available for manipulation are:

- mill speed
- mill water

- product PSD

It is worth examining these setpoints with two questions in mind:

- what determines the "best" value for that setpoint?
- can a competitive advantage be obtained by taking advantage of this variable?
- can the selection procedure be automated?

Each of these is discussed in the following sections.

4.4.1 Effect of Mill Speed

Fairly extensive work has been done on the effect of mill speed on grinding efficiency (Herbst, Rajamani, etc.). Until present it has been found satisfactory to operate mills at set speeds (or fractions of their critical speed) based on physical characteristics of the mill, aspect ratio, type of mill, discharge arrangement, etc.

It is widely accepted that there does exist a speed at which the breakage is favored most. If this point can be shown to vary depending on ore characteristics then there is a strong motivation to apply optimizing control of some sort. Tests of this nature are, however not within the scope of these analyses and are therefore left as a recommendation.

It is certain that data intensive strategies such as fuzzy logic and neural networks would include the manipulation of mill speed as it is simple to implement and would provide an extra degree of freedom to the controller. See for example Stange *et al* (1997).

4.4.2 Effect of Mill Water

Mill water directly effects the % solids in the mill. It therefore impacts on the viscosity and general flow behavior within the mill. It plainly has an important role in the efficiency of the comminution process.

The mill water addition rate therefore has potential as a second setpoint for the optimizing controller to optimize. The model already relates viscosity to throughput - now the controller has the power to drive the viscosity *and* load to achieve the maximum throughput.

It must however be remembered that it is not simply a case of adding water to "improve" the viscosity and therefore "improve" throughput. The addition of water into the mill has other *indirect* effects which may have precisely the opposite effect on throughput. Complex nonlinear systems (of which a milling circuit is an excellent example) often behave counter-intuitively and the manipulation of any one state may effect all the others in *more than one* way.

Therefore, before mill water addition can be utilized in an optimizing controller, a model relating it to throughput would need to be developed.

Also, due to the close link between load setpoint and mill water, it is plain that any strategy would need to consider these setpoints simultaneously to fully realize the potential of the strategy.

4.4.3 Effect of Product PSD

The PSD setpoint can not be used to optimize *mill* performance. In general, the effect PSD setpoint is linear; a higher PSD setpoint allows a greater throughput and vice versa. There is no peak value to seek. As far as the grinding circuit is concerned, an optimal PSD would be the same as the feed PSD (no grinding!).

Optimization the PSD therefore requires feedback from downstream processes to determine the 'highest' suitable PSD setpoint (highest since this is in the interest of the milling circuit).

Therefore, the true optimal value for this setpoint depends on elements downstream in the plant - these elements could be flotation banks, leaching tanks or many other processes. This therefore generates large scope for investigation - work can be done on any of these processes to determine if frequent adjustment or optimizing control of the PSM setpoint is viable.

Until this point, the objective function for the milling process has been defined as follows:

- *maximize throughput*
- *minimize power consumption*

The assumption was made that the product would be of specified quality; i.e. it would have been reduced to a specified size. The value chosen is usually based on the expertise of the plant operators who estimate what size distribution will best compliment the downstream processes.

As the aim is to maximize throughput and minimize power consumption, it is undesirable to mill the ore any more than is required. If the operators were to decide that a d_{50} of 80 μm were as good as 90 μm, then the plant could benefit greatly by operating at the less stringent requirement. Therefore, if the circumstance should ever arise where the downstream process has less stringent requirements these could be translated into a profit. Conversely, if a downstream process could benefit greatly from a reduced particle size, then the milling circuit should be equipped to allow this.

The selection of a particle size is therefore the result of a the competing objective functions of the grinding circuit and the downstream process. The aim of this section is to discuss the aspects of flotation that are influenced by particle size. Flotation is chosen for analysis since it commonly follows milling in the South African context.

For the case of flotation, the actual particle size required depends on the degree of liberation[+] required. The degree of liberation is a useful measure of the quality of the mill discharge and may be a useful guide to the success of the comminution process, complementing the d_{50}.

The flotation process prefers a fully liberated material that is conducive to easy separation. The standard at present is to attempt to comply with downstream requirements with respect to d_{50} only. For the case of flotation there are other desirable qualities that the feed stream may possess:

- highly liberated
- low surface area (related to reagent consumption)
- not too large (particles >120μm are difficult to float)
- not too fine (particles < 30μm are more subject to entrainment)
- constant rate (disturbances tend to reduce efficiency)

Most of these properties are dependent on the PSD chosen. Selecting the PSD setpoint is therefore a compromise between these properties. The choice to this point has been a largely intuitive process on behalf of the operating personnel; in fact some plants have used the same setpoint for years without question.

Discussion

In order to illustrate how benefit can be derived from constant monitoring of the PSM setpoint, it is necessary to simplify the liberation and separation processes.

It is only possible to achieve perfect separation if both the liberation and the separation are perfect. If, after the liberation stage, there are still particles of mixed composition, then no mechanical process can fully separate the mineral types. This can be extended to suggest that the degree of separation is strongly dependent on the degree of liberation achieved. So it would be better to rate the performance of a

[+] this is described in Chapter 2

comminution plant by the degree of liberation achieved rather than the particle size achieved.

There is a lack of understanding of how particle size effects flotation performance - but it is certain that they relate to the degree of liberation of the ore. Many flotation simulations use "fast" and "slow" floating fractions to describe the behaviour of material in the flotation environment. These can easily be related to fractions with high and low degrees of liberation. It is logical therefore that a simulator that calculates flotation performance using this model is likely to calculate improved performance (better grade - faster recovery) for a better liberated feed. The next section attempted to substantiate the above postulate.

For a batch flotation test, the recovery vs. grade relationship yields an interesting form as flotation time increases. Such a curve is given in figure 4.16. In terms of this curve, the objective function should drive the operation (refers to the effective residence time chosen for the ore) to a point on this curve somewhere in the marked region in figure 4.17; this point should take advantage of as high a recovery as possible before the grade begins to drop off too excessively. The precise point depends on the object of the separation - if high grades are necessary, then the residence time will need to be increased until this constraint is met. This will come at the expense of the recovery.

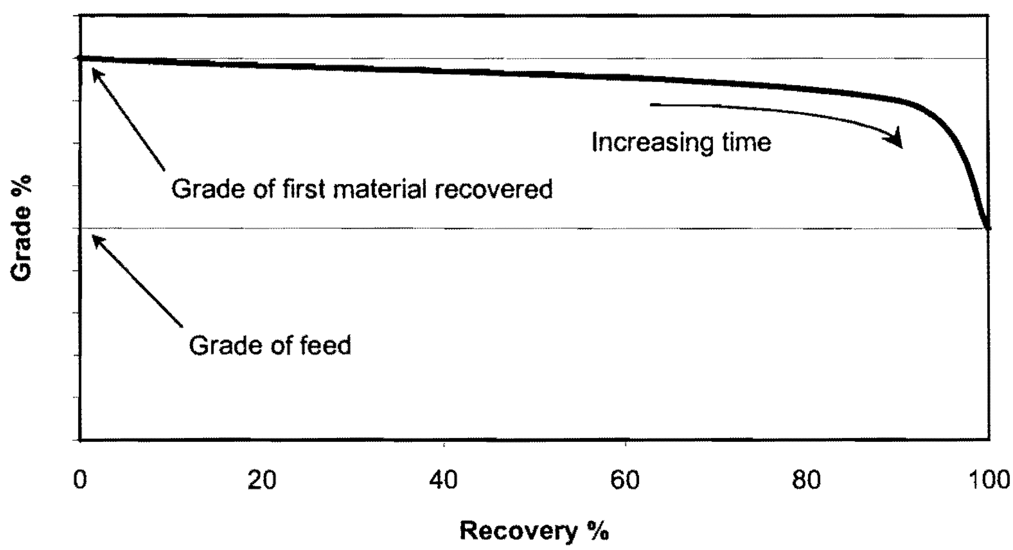


Figure 4-16 The recovery-grade relationship in separation by flotation.

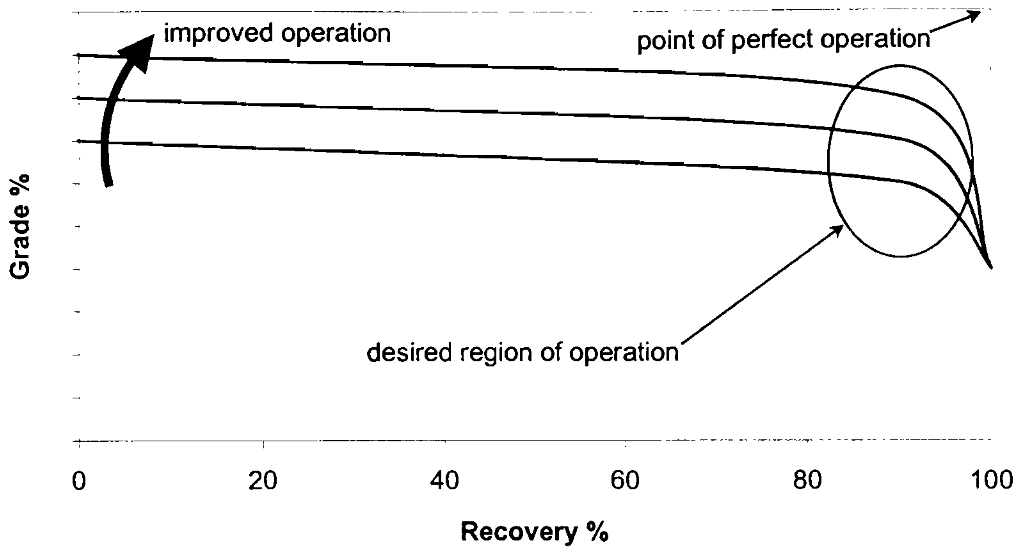


Figure 4-17 A recovery - grade plot demonstrating the region of desired operation. Residence time are selected to force operation into this region.

Although the point of operation chosen may be of importance, it is also important to analyze what forces generated the Grade-Recovery curve in the first place. It is an extremely complex function of mineralogy, surface chemistry and flotation kinetics. Actions that can be taken to move this curve "up", as it were, have been the source of constant research for many decades, and is beyond the scope of this analysis.

What *is* of interest however, is how all these relations fluctuate during operation; and ultimately, how the particle size in the feed can be used to manipulate them actively. The degree of liberation is known to have an effect on the recovery vs. grade plot. Since the settings of a flotation circuit are based on keeping the recovery grade relationship at the desired point on the curve, it must be asked whether a constant degree of *liberation* would better serve the flotation process than merely a constant degree of grinding? To answer this, the influence of the degree of liberation on the above-mentioned relations must be determined.

Effect of Degree of Liberation‡.

Liberation has a major effect on the economic aspects of the separation process, and is therefore of foremost importance when considering the objective function.

The degree of liberation of an ore depends on two factors - the mineral grain size and the actual particle size. Functions that relate their distributions to a fixed liberation (the % liberation for an ore sample is defined as the mass of completely liberated

‡ A model relating liberation to particle size is described in Chapter 2

material as a fraction of the total mass of that mineral in the sample) can be readily calculated.

Completely liberated material usually makes up the fast floating fraction - however, particles are separated as a function of their surface properties - so particles consisting largely of gangue with a coating of the desired mineral will behave precisely the same as completely liberated material (barring density effects). Although this can be combated in some cases using attrition mills, it often causes a decrease in the grade of the fast floating fraction. The fast floating fraction determines the starting position of the recovery vs. grade plot (i.e. at recovery = 0%). The higher this starting position the better the possibilities for high grades *and* recoveries.

Particles with decreasing fractions of the desired mineral on their surface have slower kinetics, and are therefore recovered over a longer period of time (it is the kinetics of these particles that determine the slope of the recovery vs. grade plot in the plateau region). Particles of pure gangue, given enough time, will also float, and the object is therefore to only subject the ore to the flotation environment until a satisfactory recovery is attained - and before excessive gangue is allowed to float (the sharp dip in grade on the plot).

Perfect liberation breaks the material into two distinct types and therefore clearly separates the ore into two distinct kinetic regions making the choice of residence time simpler. - the recovery may reach the desired level faster and fewer gangue particles will therefore have time to be captured. On the recovery vs. grade plot, perfect liberation would manifest itself as a higher and flatter curve - precisely what is desired.

Naturally, perfect liberation is impossible; the flotation circuit needs to cope with what it receives. Air flow rates and water addition rates govern the kinetics of the operation and are set to give residence times that give a suitable grade and recovery.

So the grade recovery relationship depends on liberation; and since the residence times are selected based on the recovery grade plot, it is plain to see that the flowrate setpoints chosen depend on the degree of liberation in the feed. This link suggests that on-line process control of flowrates should take the current degree of liberation into account.

Even if such a controller was not used, benefit can be gained by keeping the degree of liberation at the values which were used to derive the recovery grade plot. This means control of the degree of liberation in the product from the mills *can* benefit flotation performance.

Another point that may be of concern focuses on reagents. Reagents flow into a circuit is often set as a function of the flowrate through that plant section. However, reagent consumption and effectiveness depends strongly on the surface area of the material in the stream. If a plant were to vary dosages based on surface area, ensuring the same degree of surface loading on the minerals, then the particle size is again important.

Controlling the Degree of Liberation

If it were possible to control the degree of liberation directly (which you wish to maximize), rather than via the PSD, a scheme to optimize the milling-flotation pair becomes apparent:

There does exist a range of particle sizes for which flotation is satisfactory. Particles below this range consume large quantities of reagent, have low bubble collision probabilities and are highly susceptible to entrainment, while particles above the range are difficult to suspend and also exhibit highly unstable bubble attachment. There is still however a fairly large range of operation and work is being done to increase this effective range (Mt. Isa Mines Ltd. have a particular interest; as the result of a large grain size in one of their copper mines, high liberation was found possible at sizes as great as 300μm).

This leads to a possible solution: control the grinding circuit based on the grain size to achieve a more consistent degree of liberation. This controller could allow the PSM setpoint to wander within limits as is required to attain the desired liberation. These limits would need to chosen such that the reagent dosages (as dictated by their controller in usual operation) are satisfactory throughout the range.

A Strategy for PSM Setpoint Adjustment

A suitable setpoint can be found from simple tests performed on fresh samples of the cyclone overflow. Samples could be batch floated under standardized conditions (same reagent dosages, residence times, etc.) to return recovery by size and grade by size data plots. This would reveal the selectivity as a function of size, a simple gauge of how the flotation circuit will cope with feeds of various sizes. This strategy is currently impractical since it would require extensive human input; however, as instrumentation becomes better and cheaper, such possibilities may arise.

4.4.4 Optimization of Profit

In the above sections and throughout the thesis, many small aspects of circuit operation are discussed each with a separate aim. One section aims to increase throughput, another to reduce power consumption, while still other sections aim to improve the product quality.

In order to distill these aims to a central theme it is necessary to link each with the central aim, optimizing profit. It is clear that increased throughput will result in an increase in profit since more product may be produced in the same period; throughput can be equated to a rate of 'value addition'.

Power consumption also relates simply to profit, with certain exceptions: power costs vary depending on rate of use and time of day. These do not have any immediate effect on the control strategy *per se*, but does influence startup procedures as well as motor design.

Of interest however, is the effect of product quality as discussed in the previous section. In order to quantify the reward for improved product quality it is necessary to relate all aspects of operation to a single, profit-oriented, objective function.

Designing controllers with this strategy in mind is called the Centralized (or Global) approach. This method ensures that sections of the plant compliment one another but does have some the disadvantage that iterations will necessarily be slower since the speed of the global controller will be limited by the speed of the slowest section of the plant.

To combat this problem the Distributed approach may be used, where smaller plant sections are optimized individually, taking advantage of each units unique characteristics. The overall optimizer is then given the task of coordinating the local optimizers and determining the optimum for a few selected key variables. Examples of key variables would be overall feed rates and product specifications (Darby and White, 1988).

In the case of mineral processing and comminution, the properties of the milling circuit product are key to overall optimization. The particle size distribution affects the operation of the downstream operations. In the case of flotation, the grade and recovery gauge the success and profitability. In order to optimize the milling circuit product, it is necessary to determine the affect of the PSD (particle size distribution) and degree of liberation on the recovery and grade.

Flotation modelling is still fairly simple and is therefore unable to provide robust and accurate predictions of these effects at present. This will hopefully change in the next few years. For the present it will be necessary to utilize these theoretical models in conjunction with stochastic relationships for specific circuits to predict the effects of comminution on flotation.

Chapter 5

Conclusion

A new strategy for optimizing the control of grinding circuits has been presented. The controller aims to increase circuit throughput by manipulating and optimizing the load setpoint which is utilized by the underlying multivariable controller.

The optimal load setpoint is estimated using a simple model of the load-throughput relationship that utilizes slurry viscosity readings to reduce the model to a simple quadratic equation. The load setpoint that optimizes the throughput in this model is then implemented on the plant.

The model, a 3-dimensional surface, is designed to analyze the plant's recent throughput history, in relation to load and viscosity, in order to 'adapt' to the plant as its fundamental character changes. This ensures an accurate model with a minimal of operator input.

The control strategy is referred to as a 'Model-Based Optimizing Control' since it fulfills the role of an optimizing controller by determining setpoints based on the result of an optimization process, and it utilizes a plant model to correlate the setpoint with the optimized variable.

Testing of an optimizing controller ultimately seeks to determine two things:

- Is the controller robust? Is it reliable in the face of various events?

- Is the objective function value improved? I.e. is the controller performing its purpose?

As far as reliability is concerned, optimizing controllers are relatively low-risk, since their continued operation is not essential to plant operation. Should the controller fail to alter setpoints as required the system merely returns to the type of operation that existed before the inclusion of the controller.

The only circumstance where the controller may detrimentally effect operation is if it suggests extremely poor setpoints, and since these setpoints are usually within a very confined region the controller may be set to work only in a very specific range, thereby removing this risk.

The second question can be answered by comparing performance with existing strategies, such as the hill-climbing optimizing controller of Craig *et al.* These tests demonstrate that throughput may be increased in several circumstances, particularly during sustained periods of stable operation.

This result prompts the use of a combined strategy, where different optimization processes are used in different circumstances. As a crude example, it may be beneficial to use the hill-climbing technique during startup and during any periods of disruption, and revert to the model based controller during periods of predictable operation. This could then be refined as more is learnt about which controller is best for certain, more specific, circumstances.

It must however be pointed out that tests can not be considered conclusive as the simulator used, although highly sophisticated at the time of writing, remains limited in its representation of real milling circuits. Although particle size distributions, slurry densities and flows can be modelled to match a real plant extremely well, internal effects, such as the effect of viscosity on breakage are not modelled at all. Therefore the model fitted during tests in this project may bear little relationship to what may occur in a real mill. This was fully appreciated during controller development, therefore the two-variable quadratic model was designed to be fully capable of achieving the shape of the load-viscosity-throughput relationship for many different plant types.

For this reason, further testing of the strategy would be required, either on laboratory or pilot scale, before the potential of the strategy can be accurately assessed. The theory is now in place, and further testing is possible with the minimum of modification to the methods applied in this thesis.

This concludes the presentation of the theory and initial testing of a new, model based optimizing controller for milling circuit operation.

References

Arbiter, N., Harris, C.C., "Scale -up and dynamics of large grinding mills - a case study", In: A.L. Mular and G.V. Jergensen II (Editors), *Design and Installation of Comminution Circuits*, AIME, New York, Ch.26, pp491-508 (1982)

Austin, L.G., Klimpel, R.R., Luckie, P.T., "Process Engineering of Size Reduction: Ball Milling", *SME-AIME*, New York (1984).

Austin, L.G., "A mill power equation for SAG mills", *Minerals and Metallurgical Processing*, pp57-62 (Feb 1990)

Belardi, G., Bonifazi, G., Massacci, P., "Particle Breakage and Mineral Liberation Modelling", *Chem Eng Res Dev*, **64/65**, pp381-391 (1986).

Bond, F.C., *Crushing and Grinding Calculations*, Allis-Chalmers Publ. (1962).

Braae, M., *Control Theory for Electrical Engineers*, UCT Press (1994).

Chen, C.Y., Joseph, B., "On-line Optimization Using a Two-Phase Approach: An Application Study", *Ind. Eng. Chem. Res.*, pp1924-1930 (1987).

Cilliers, J.J., Hinde, A.L., "An improved hydrocyclone model for backfill preparation", *Minerals Engineering*, **4**, nos. 7-11, pp683-693 (1991).

Craig, I.K., Hulbert, D.G., Metzner, G., Moult, S.P., "Optimized multivariable control of an industrial run-of-mine milling circuit.", *Jrnl. S. Afr. Min. Metall.*, **92**, no.6, pp169-176 (1992).

Darby, M.L., White, D.C., "On-Line Optimization of Complex Process Units", *Chemical Engineering Progress*, pp51-59, (1988).

Davis, E.W., "Fine crushing in ball mills", *Trans. AIME*, **61**, pp250-296 (1919).

Eitelberg, E., *Optimal Estimation for Engineers*, NOYB Press (1991).

Fuerstenau, D.W., Venkataraman, K.S., Velamakanni, B.V., "Effect of Chemical Additives on the dynamics of grinding media in wet ball mill grinding", *Int. J. Min. Proc*, **15**, pp251-267 (1985).

Flintoff, B.C., Plitt. L.R., Turak, A.A., "Cyclone modelling: a review of present technology", *CIM Bulletin*. **80**, no. 905, pp39-50 (1987).

Gao, M.W., Forssberg, E.K.S., "Behaviour of the selction function with respect to specific grinding energy", *7th symposium on comminution*, pre-prints, part 2, pp413-428 (1990).

Garcia, C.E., Morari, M., "Optimal Operation of Integrated Processing Systems", *AIChE Jrnl*, pp960-968 (1981).

Garcia, C.E., Morari, M., "Optimal Operation of Integrated Processing Systems: Part II", *AIChE Jrnl.*, pp226-234 (1984).

Grebe, J.J., Boundy, R.H., Cermak, R.W., "The control of Chemical processes", *Trans. AIChE*, **29**, 211 (1933).

Hales, L.B., Vanderbeek, J.L., Herbst, J.A., "Supervisory Control of a Semi-Autogenous Grinding Circuit", *Int. Jrnl. Min. Proc.*, **22**, pp.297-312 (1988).

Harris, C.C., Schnock, E.M., Arbiter, N, "Grinding mill power consumption", *Min. Proc. Tech. Rev.*, **1**, pp297-345 (1985).

Heikanen, K., Laapas, H., "On the Effects of the Fluid Rheological and Flow Properties in the Wet Gravitational Classification", *13th Int. Min. Proc. Congr.*, Warsaw, **1**, Polish Scientific Pub., pp183-204 (1979).

Herbst, J.A., Alba, J.F., Pate, W.T., Oblad, A.E., "Optimal Control of Comminution Operations", *Int. Jrnl. Min. Proc*, **22**, pp275-296 (1988).

Herbst, J.A., Bascur, O.A., "Mineral processing control in the 1980's - realities and dreams", *Control '84. Mineral/Metallurgical Processing*, SME, pp197-214 (1984).

Herbst, J.A., Mular, A.L., "Analysis and control of mineral processing", *Computer methods for the 80's*, SME, Section 5 (1980).

Herbst, J.A., Pate, W.T., Vanderbeek, J.L. and Hales, L.B., "Real time estimation of unmeasured variables in a semiautogenous grinding circuit", *SME-AIME Meeting*, Salt Lake City, Utah (1983).

Herbst, J.A., Rajamani, R.K., "Evaluation of optimising control strategies of closed circuit grinding" *Proceedings of the 13th annual Mineral Processing Congress*, Warsaw (1979).

Herbst, J.A., Siddique, M., Rajamani, K., Sanchez, E., "Population balance approach to ball mill scale up: Bench and pilot scale investigations", *SME/AIME*, vol. 272, pp1945-1954 (1982).

Hinde, A.L., "The size distribution of mined ores and the effects of attrition and abrasion during tramming", Chamber of mines, Technical note No. T124/85 (1985).

Hulbert, D.G., Braae, M., "Multivariable Control of a milling circuit at East Driefontein Gold Mine", NIM report no. 2113 (1981).

Hulbert, D.G., Woodburn, E.T., "Multivariable Control of a Wet-Grinding Circuit", *AiChE Jrnl.*, pp.186-191 (1983).

Ivanoff, A., "Theoretical foundations for the automatic regulation of temperature", *Jrnl. Inst. Fuel*, 7, 117 (1934).

Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," *Trans. ASME Jrnl. of Basic Engineering*, pp. 35-45 (March 1960).

Kapur, P.C., Ranjan, S., Fuerstenau, D.W., "A Cascade-Cataract Charge Flow Model for Power draft of Tumbling Mills", *Int. Jrnl. Min. Proc.*, **36**, pp9-29 (1992).

Kelly, E.G., Spottiswood, D.J., *Introduction to Minerals Processing*, Wiley (1982).

King, R.P., "The Prediction of Mineral Liberation From Mineralogical Texture", *XIV Inter Miner Process Congress*, pp 1-19 (1982).

King, R.P., "A Model for the Quantitative Estimation of Mineral Liberation by Grinding", Int. Jrnl. Min. Proc., **6**, pp207-220 (1979)

Lee, W., Weekman, V.W., "Advanced Control Practice in the Chemical Process Industry: A View from Industry", *AIChE Jrnl.*, **22**, no. 1, pp27-38 (1976).

Lynch, A.J., Rao, T.C., "Modelling and scale-up of hydrocyclone classifiers", *Proceedings of the 11th annual Mineral Processing Congress*, Cagliari, pp.245-269 (1975).

Maciejowski, J.M., *Multivariable feedback design*, Addison Wesley (1989).

Maybeck, P.S., *Stochastic Models, Estimation, and Control*, Volume 1, Academic Press, Inc. (1979).

Mayr, O., *The origins of feedback control*, MIT Press, Cambridge, MA (1970).

Mishra, B.K., Rajamani, R.K., "Numerical simulation of charge motion in ball mills - Lifter bar effect", *Min. & Met. Proc*, pp86-90, (May 1993).

Mishra, B.K., Rajamani, R.K., "Simulation of Charge Motion in Ball Mills: Part I", *Int. Jrnl. Min. Proc.*, **40**, pp171-186 (1994a).

Mishra, B.K., Rajamani, R.K., "Simulation of Charge Motion in Ball Mills: Part II", *Int. Jrnl. Min. Proc.* **40**, pp187-198 (1994b).

Napier-Munn, T.J., "The effect of dense medium viscosity on separation efficiency", *Coal Preparation*, **8** (1990).

Nates, N.B., Nurick, G.N., Reddy, B.N., "The slip of a single particle inside of a rotating cylinder. Part I Theoretical Analysis", *Int. Jrnl Min Proc.*, **38**, pp67-79 (1994a).

Nates, N.B., Nurick, G.N., Reddy, B.N., "The slip of a single particle inside of a rotating cylinder. Part II Experimental Investigation", *Int. Jrnl Min Proc.*, **38**, pp81-92 (1994b).

Petterson, J.K., Sandvik.K.L., "Estimating the Breakage and Selection Functions for a Continuous Mill", *Int Jrnl Min Proc.*, **35**, pp149-158 (1992).

Pietch. W., "Wet grinding experiments in a torque ball mill", In: H. Rumpf and K. Schoenert (editors), *Zerkleinern. Dechema-Monographien* Nr. 1292-1326, Band 69, Part 2, Verlag Chemie, Weinheim, pp. 751-779 (1972).

Pichumani, R., Hemler, P.F., Napel, S., Sumanaweera, T.S., Drace, J., Enzmann, D.R., Adler, J.R., "Segmentation of Spinal Disc Tissue in MR using Kalman Filters and Active Contour Models", *RSNA Annual Conference*, Chicago, November 28 (1994).

Plitt, L.R., "A mathemetical model of the hydrocyclone classifier", *CIM bulletin* **69**, no. 775, pp114-123 (1976).

Rajamani, R.K., Herbst, J.A., "Optimal control of a ball mill grinding circuit-I. Grinding circuit modelling and dynamic simulation", *Chem. Eng. Sci.*, **46**, no. 3, pp861-870 (1991).

Rajamani, R.K., Herbst, J.A., "Optimal control of a ball mill grinding circuit-II. Feedback and Optimal control", *Chem. Eng. Sci.*, **46**, no. 3, pp871-879 (1991).

Reeves, T.J., "On-line viscometer for mineral slurries", *Trans. IMM*, **94**, C201-208 (1985).

Rosenbrock, H.H., Storey, C., *Mathematics of Dynamical Systems*, Nelson (1970).

Rosenbrock, H.H., *Computer-Aided Control System Design*, Academic Press (1974).

Rowland, C.A., "Diameter factors effecting ball mill scale-up", *Int. Jrnl. Min. Proc.*, **22**, pp95-104 (1988).

Seborg, D.E., Edgar, T.F., Mellichamp, D.A., *Process Dynamics and Control*, Wiley (1989).

Shi., F.N., Napier-Munn, T.J., "A model for slurry rheology", *Int Jrnl Min Proc.*, **47**, pp103-123 (1996a).

Shi., F.N., Napier-Munn, T.J., "Measuring the rheology of slurries using an on-line viscometer", *Int. Jrnl. Min. Proc.*, **47**, pp153-176 (1996b).

Stange, W., McInnes, C., Norris, N., van Niekerk, L., Williams, H., "Optimum Control of the Leeudoorn Semi-Autogenous Milling Circuit", *Proceedings - Process Control Workshop '97*, SAIMM Western Cape, pp32-53, (1997)

Skogestad, S., Postlethwaite, I., *Multivariable feedback control - Analysis and design*, Wiley (1996).

Stewart, P.S.B., "The control of wet grinding circuits", *Aust. Chem. Proc. Eng.*, pp22-23 (1970).

Thomas, D.G., "Transport Characteristics of Suspensions VIII - A note on the viscosity of Newtonian suspensions of uniform spherical particles", *Jrnl. Colloid Science*, **20**, pp267-277 (1965).

Tkácová, K., Heegn, H., Števulová, N., "Energy transfer and conversion during comminution and mechanical activation", *Int. Jrnl. Min. Proc.*, **40**, pp17-31 (1994).

Valenzuela, J., Najim, K., Villar, R.D., Bourassa, M., "Learning control of an autogenous grinding circuit", *Int. Jrnl. Min. Proc.*, **40**, pp33-44 (1994).

Wiegel, R.L., Li, K., "A Random Model for Mineral Liberation by Size Reduction", *SME/AIME Trans*, pp179-189 (1967). ",

Young, J.C.C., *Constraint Proximity in Optimizing Control*, PhD Thesis, University of Cape Town (1997).

Ziegler, J.G., "Those magnificent men and their controlling machines", *Jrnl. of Dynamic Systems*, Measurement and Control, Trans. ASME, **97**, 279 (1975).

Zenz, F.A., Othmer, D.F., *Fluidization and Fluid Particle Systems*, Rienhold (1960).

Zeng, Y., Forsberg, E., "Monitoring grinding parameters by signal measurement for an industrial ball mill", *Int Jrnl Min Proc.*, **40**, pp1-16 (1994).

Appendices

A.1 Selected Programmes and Algorithms

The thesis utilizes computer programmes extensively during the development and testing of the proposed control strategy. During the initial phases several FORTRAN programmes were written to assist in the understanding of circuit dynamics and to get a feel for the accuracy (or lack thereof) of mass-balance type models.

Next, a more advanced FORTRAN simulator was kindly donated by Mintek for further study. This programme was however soon replaced with another from Mintek, utilizing many new features, which being coded in Visual Basic (v3.0) proved more user friendly.

This simulator was then used as the testing ground for many strategies, eventually leading to the model based controller described in the thesis. Unfortunately, the coding for the control strategy is interspersed amongst the simulator code which is confidential in nature. A few algorithms of interest are however included for discussion.

A.1.1 Simple Milling Simulation

This section serves to provide some perspective to the programming of mill simulators. A listing of a simple FORTRAN simulator written during the early stages of the thesis is provided, followed by some discussion.

Main Section

```
PROGRAM pcircuit

        !-----------------------------------------------------!
        ! This program simulates a milling circuit            !
        !                                                     !
        !                                                     !
        !      uses 20 particle size classes                  !
        !                                                     !
        !      Jarrod Hart                 April 1996         !
        !-----------------------------------------------------!

        DOUBLE PRECISION M_mf,mmf(20),M_mp,mmp(20),M_sp,msp(20)
        DOUBLE PRECISION Vs,Vm,Qmf,Qmp,W,rhow,Qsp,M_f,mf(20),Qf
        DOUBLE PRECISION Cssp,Csmp,Mm,rhos,Vm,cutsize,tot1
        DOUBLE PRECISION tot2

        INTEGER i,counter

        !EXTERNAL sump
        !EXTERNAL mill

        ! Setup mill, sump and input constants

        rhow=1000                    ! Liquid density (kg/m^3)
        rhos=3500                    ! Solids density (kg/m^3)
        Vs=10                        ! Init vol slur sump (m^3)
        Vm=40                        ! Used vol mill (m^3)
        Mm=27000                     ! Init mass sol mill (kg)
        Ms=4500                      ! Init mass sol sump (kg)
        Cssp=Ms/Vs                   ! Init frac sol sump (kg/m^3)
        Csmp=Mm/Vm                   ! Init frac sol mill (kg/m^3)

        ! Setup circuit feed

        M_f=38
        Qf=0.07
        tot2=0

        mf(1)=0.2
        mf(2)=0.5
        mf(3)=1
        mf(4)=2
        mf(5)=3
        mf(6)=4
        mf(7)=6
        mf(8)=7.5
        mf(9)=9
        mf(10)=10
        mf(11)=12
        mf(12)=16
        mf(13)=20
        mf(14)=25
        mf(15)=30
        mf(16)=40
        mf(17)=50
        mf(18)=52
        mf(19)=48
        mf(20)=46
```

```
      ! Normalize

      DO i=1,20
            tot2=tot2+mf(i)
      END DO
      DO i=1,20
            mf(i)=mf(i)/tot2
      END DO

      ! Setup mill feed

      M_mf=76
      Qmf=0.14
      tot1=0
      DO i=1,20
            mmf(i)=1
            tot1=tot1+mmf(i)
      END DO
      DO i=1,20
            mmf(i)=mmf(i)/tot1
      END DO

      ! Setup mill product

      M_mp=76
      Qmp=0.14
      tot1=0
      DO i=1,20
            mmp(i)=1
            tot1=tot1+mmp(i)
      END DO
      DO i=1,20
            mmp(i)=mmp(i)/tot1
      END DO

      ! Setup sump product

      M_sp=76
      Qsp=0.25
      tot1=0
      DO i=1,20
            msp(i)=1
            tot1=tot1+msp(i)
      END DO
      DO i=1,20
            msp(i)=msp(i)/tot1
      END DO

      ! Settings

      W=0.11                       ! Sump dilution rate (m^3/s)
      cutsize=75                   ! Cutsize in cyclone (microns)

      ! Write size info to file

      OPEN (UNIT=16,FILE='datafiles/cutsz.dat',STATUS='OLD',
     %              IOSTAT=istat)

      write (16,*) cutsize

      CLOSE (UNIT=16)

      OPEN (UNIT=16,FILE='datafiles/cutszerr.dat',STATUS='OLD',
     %              IOSTAT=istat)

      write (16,*) 0

      CLOSE (UNIT=16)

      OPEN (UNIT=16,FILE='datafiles/size.dat',STATUS='OLD',
     %              IOSTAT=istat)
```

```
! Fraction less than x microns

write (16,*) 13
write (16,*) 16
write (16,*) 19
write (16,*) 22
write (16,*) 27
write (16,*) 32
write (16,*) 37
write (16,*) 45
write (16,*) 53
write (16,*) 63
write (16,*) 75
write (16,*) 89
write (16,*) 106
write (16,*) 126
write (16,*) 150
write (16,*) 178
write (16,*) 212
write (16,*) 252
write (16,*) 300
write (16,*) 357

CLOSE (UNIT=16)

! Write to file

 OPEN (UNIT=11,FILE='datafiles/data.dat',STATUS='OLD',
%       IOSTAT=istat)
 OPEN (UNIT=12,FILE='datafiles/mfeed.dat',STATUS='OLD',
%       IOSTAT=istat)
 OPEN (UNIT=13,FILE='datafiles/mprod.dat',STATUS='OLD',
%       IOSTAT=istat)
 OPEN (UNIT=14,FILE='datafiles/sprod.dat',STATUS='OLD',
%       IOSTAT=istat)
 OPEN (UNIT=15,FILE='datafiles/feed.dat',STATUS='OLD',
%       IOSTAT=istat)
 OPEN (UNIT=16,FILE='datafiles/leverr.dat',STATUS='OLD',
%       IOSTAT=istat)


write (11,*) rhow           ! Liquid density (kg/m^3)
write (11,*) rhos           ! Solids density (kg/m^3)
write (11,*) Vs             ! Init vol slur sump (m^3)
write (11,*) Vm             ! Vol mill (m^3)
write (11,*) Mm             ! Init mass sol mill (kg)
write (11,*) Ms             ! Init mass sol sump (kg)
write (11,*) Cssp           ! Init frac sol sump (kg/m^3)
write (11,*) Csmp           ! Init frac sol mill (kg/m^3)
write (11,*) W              ! Sump dilution rate

write (15,*) M_f
write (15,*) Qf
write (16,*) 0
DO i=1,20
      write (15,*) mf(i)
END DO

write (12,*) M_mf
write (12,*) Qmf
DO i=1,20
      write (12,*) mmf(i)
END DO

write (13,*) M_mp
write (13,*) Qmp
DO i=1,20
      write (13,*) mmp(i)
END DO

write (14,*) M_sp
write (14,*) Qsp
```

```
      DO i=1,20
            write (14,*) msp(i)
      END DO

      CLOSE (UNIT=11)
      CLOSE (UNIT=12)
      CLOSE (UNIT=13)
      CLOSE (UNIT=14)
      CLOSE (UNIT=15)
      OPEN (UNIT=16,FILE='datafiles/count.dat',STATUS='OLD',
     %            IOSTAT=istat)
            write (16,*) 0
      CLOSE (UNIT=16)

      ! Simulation


      OPEN (UNIT=41, FILE='datastorage/feed.out',STATUS='OLD',
     &      IOSTAT=istat)
      OPEN (UNIT=42, FILE='datastorage/mfeed.out',STATUS='OLD',
     &      IOSTAT=istat)
      OPEN (UNIT=43, FILE='datastorage/mprod.out',STATUS='OLD',
     &      IOSTAT=istat)
      OPEN (UNIT=44, FILE='datastorage/sprod.out',STATUS='OLD',
     &      IOSTAT=istat)
      OPEN (UNIT=45, FILE='datastorage/cycof.out',STATUS='OLD',
     &      IOSTAT=istat)
      OPEN (UNIT=46, FILE='datastorage/cycuf.out',STATUS='OLD',
     &      IOSTAT=istat)
      OPEN (UNIT=47, FILE='datastorage/data.out',STATUS='OLD',
     &      IOSTAT=istat)
      OPEN (UNIT=48, FILE='datastorage/sizes.out',STATUS='OLD',
     &      IOSTAT=istat)


      DO i=1,4000

            OPEN (UNIT=16,FILE='datafiles/count.dat',STATUS='OLD',
     %            IOSTAT=istat)
            READ (16,*) counter
            CLOSE (UNIT=16)

            CALL pfeed
            CALL pmill
            CALL psump
            CALL pcyclon
            CALL pmixer
            IF (MOD(counter,20).EQ.0) THEN
                  CALL printout
            END IF
            counter=counter+1

            OPEN (UNIT=16,FILE='datafiles/count.dat',STATUS='OLD',
     %            IOSTAT=istat)
            write(16,*) counter
            CLOSE (UNIT=16)

      END DO

      CLOSE (UNIT=21)
      CLOSE (UNIT=22)
      CLOSE (UNIT=23)
      CLOSE (UNIT=34)
      CLOSE (UNIT=35)
      CLOSE (UNIT=36)
      CLOSE (UNIT=37)

      END
```

Subroutines

```
SUBROUTINE pfeed

      !------------------------------------------------------!
      ! This program generates the feed                      !
      !                                                      !
      !       uses 20 particle size classes                  !
      !                                                      !
      !       Jarrod Hart                      April 1996    !
      !------------------------------------------------------!

      DOUBLE PRECISION M_f, mf(20),Qf,R,R2
      INTEGER i, counter,iseed

      OPEN (UNIT=11,FILE='datafiles/feed.dat',STATUS='OLD',
     &          IOSTAT=istat)


      ! Input

      READ (11,*) M_f                   ! old Mass feed (kg/s)
      READ (11,*) Qf                    ! old Vol feed (m^3/s)

      DO i=1,20
             READ  (11,*) mf(i)
      END DO


      CLOSE (UNIT=11)
      OPEN (UNIT=11,FILE='datafiles/feed.dat',STATUS='OLD',
     &          IOSTAT=istat)

      OPEN (UNIT=16,FILE='datafiles/count.dat',STATUS='OLD',
     %       IOSTAT=istat)
      READ (16,*) counter
      CLOSE (UNIT=16)

      ! Manipulations

      !M_f=M_f+1
      OPEN (UNIT=26,FILE='datafiles/iseed.dat',STATUS='OLD',
     %       IOSTAT=istat)
      READ (26,*) iseed
      CLOSE (UNIT=26)

      R=URAND(iseed)
      R2=URAND(iseed)
      M_f=38     !+7*INT(R+0.5)
      Qf=0.07    !98+0.004*INT(R2+0.5)

      OPEN (UNIT=26,FILE='datafiles/iseed.dat',STATUS='OLD',
     %       IOSTAT=istat)
      write (26,*) iseed
      CLOSE (UNIT=26)

      !IF (counter.GT.1000 .AND. counter.LT.2000) THEN
      !      M_f=1.4*M_f
      !END IF

      !IF (mod(counter,200).eq.100) THEN
      !      M_f=1/1.2*M_f
      !END IF
      !IF (counter .EQ. 500) THEN
      !      M_f=1.4*M_f
      !      Qf=1.4*Qf
      !END IF
      IF (counter .GT. 2000) THEN
             M_f=(1.4)*M_f
             !Qf=(1.4)*Qf
      END IF
```

```
      write (11,*) M_f
      write (11,*) Qf

      DO i=1,20
            write(11,*) mf(i)
      END DO

      CLOSE (UNIT=11)

      END

      REAL FUNCTION URAND(IY)
      INTEGER IY,IA,IC,ITWO,M2,M,MIC
      DOUBLE PRECISION HALFM
      REAL s
      DATA M2/0/,ITWO/2/
      IF(M2.EQ.0) THEN
            M=1
10          IF(M.GT.M2)THEN
                  M2=M
                  M=ITWO*M2
                  GOTO 10
            END IF
            HALFM=M2
            IA=8*INT(HALFM*ATAN(1.D0)/8.D0)+5
            IC=2*INT(HALFM*(0.5D0-SQRT(3.D0)/6.D0))+1
            MIC=(M2+IC)-M2
            S=0.5/HALFM
      END IF
      IY=IY*IA
      IY=IY+IC
      IF(IY.LT.0)IY=(IY+M2)+M2
      URAND=FLOAT(IY)*S
      RETURN
END


SUBROUTINE pmill

      !---------------------------------------------------!
      ! This routine estimates the mill output            !
      !                                                   !
      !     uses 20 particle size classes                 !
      !     assumes a constant volume within the mill     !
      !                                                   !
      !      Jarrod Hart                    April 1996    !
      !---------------------------------------------------!

      DOUBLE PRECISION M_mp, mmp(20), M_mf, mmf(20), Ms, Mm, R, totl
      DOUBLE PRECISION Vm, Qmp, rhow, Qmf, Csmf, Csmp,S(20),b(20,20),f
      INTEGER i,j,k

      OPEN (UNIT=11,FILE='datafiles/mprod.dat',STATUS='OLD',
     &         IOSTAT=istat)
      OPEN (UNIT=12, FILE='datafiles/mfeed.dat',STATUS='OLD',
     &         IOSTAT=istat)
      OPEN (UNIT=13, FILE='datafiles/s.dat',STATUS='OLD',
     &         IOSTAT=istat)
      OPEN (UNIT=14, FILE='datafiles/bb.dat',STATUS='OLD',
     &         IOSTAT=istat)
      OPEN (UNIT=15, FILE='datafiles/data.dat',STATUS='OLD',
     &         IOSTAT=istat)

      ! Read data

      READ (15,*) rhow           ! Liquid density (kg/m^3)
      READ (15,*) rhos           ! Solids density (kg/m^3)
      READ (15,*) Vs                   ! Init vol slur sump (m^3)
      READ (15,*) Vm                   ! Vol mill (m^3)
      READ (15,*) Mm                   ! Mass sol mill (kg)
      READ (15,*) Ms                   ! Mass sol sump (kg)
```

```
      READ (15,*) Cssp              ! Frac sol sump (kg/m^3)
      READ (15,*) Csmp               ! Frac sol mill (kg/m^3)
      READ (15,*) W                 ! Sump dilution rate

      ! Input

      READ (12,*) M_mf                   ! Mass rate feed (kg/s)
      READ (12,*) Qmf                    ! Vol rate feed (m^3/s)
      READ (11,*) M_mp                   ! old Mass prod feed (kg/s)
      READ (11,*) Qmp                    ! old Vol prod feed (m^3/s)
      Qmp=Qmf                             ! Mill discharge setting (m^3/s)
      Csmf=M_mf/Qmf                      ! Frctn sols feed (kg/m^3)
      Mm=Mm+M_mf-M_mp                     ! Mass sol in mill (kg)
      Csmp=Mm/Vm                    ! Frac sols in mill
      M_mp=Qmp*Csmp                 ! Mass rate solids dschg (kg/s)


      OPEN (UNIT=16,FILE='datafiles/count.dat',STATUS='OLD',
     %       IOSTAT=istat)
      READ (16,*) counter
      CLOSE (UNIT=16)

      ! Import feed size distribution

      DO i=1,20
             READ  (12,*) mmf(i)
             READ  (11,*) mmp(i)
      END DO

      CLOSE (UNIT=11)
      OPEN (UNIT=11,FILE='datafiles/mprod.dat',STATUS='OLD',
     &           IOSTAT=istat)

      write (11,*) M_mp
      write (11,*) Qmp

      OPEN (UNIT=26,FILE='datafiles/iseed.dat',STATUS='OLD',
     %       IOSTAT=istat)
      READ (26,*) iseed
      CLOSE (UNIT=26)

      R=URAND2(iseed)
      R=1+(R-0.5)/10

      R=R*0.25

      !IF (counter.GT.800) THEN
      !      R=R*0.6
      !END IF

      OPEN (UNIT=26,FILE='datafiles/iseed.dat',STATUS='OLD',
     %       IOSTAT=istat)
      write (26,*) iseed
      CLOSE (UNIT=26)

      DO i=1,20
             READ (14,*) (b(i,j),j=1,20)
             READ (13,*) S(i)
             S(i)=S(i)*R
      END DO

      totl=0
      DO i=1,20
             f=0
             DO k=1,21-i
                    f=f+b(22-k-i,i)*S(k)*Mm*mmp(k)
             END DO
             !write (*,*) f
             mmp(i)=mmp(i)+(M_mf*mmf(i)-M_mp*mmp(i)-S(i)*mmp(i)*Mm+f)/Mm
             totl=totl+mmp(i)
      END DO
```

```
      DO i=1,20
            write(11,*) mmp(i)/tot1
            !write(*,*) mmp(i),tot1,mmp(i)/tot1,Mm,S(i)
      END DO

      CLOSE (UNIT=15)
      OPEN (UNIT=15, FILE='datafiles/data.dat',STATUS='OLD',
     &         IOSTAT=istat)

      WRITE (15,*) rhow          ! Liquid density (kg/m^3)
      WRITE (15,*) rhos          ! Solids density (kg/m^3)
      WRITE (15,*) Vs                  ! Init vol slur sump (m^3)
      WRITE (15,*) Vm                  ! Vol mill (m^3)
      WRITE (15,*) Mm                  ! Mass sol mill (kg)
      WRITE (15,*) Ms                  ! Mass sol sump (kg)
      WRITE (15,*) Cssp          ! Frac sol sump (kg/m^3)
      WRITE (15,*) Csmp           ! Frac sol mill (kg/m^3)
      WRITE (15,*) W                   ! Sump dilution rate

      CLOSE (UNIT=11)
      CLOSE (UNIT=12)
      CLOSE (UNIT=13)
      CLOSE (UNIT=14)
      CLOSE (UNIT=15)

      END

      REAL FUNCTION URAND2(IY)
      INTEGER IY,IA,IC,ITWO,M2,M,MIC
      DOUBLE PRECISION HALFM
      REAL s
      DATA M2/0/,ITWO/2/
      IF(M2.EQ.0) THEN
            M=1
10          IF(M.GT.M2)THEN
                  M2=M
                  M=ITWO*M2
                  GOTO 10
            END IF
            HALFM=M2
            IA=8*INT(HALFM*ATAN(1.D0)/8.D0)+5
            IC=2*INT(HALFM*(0.5D0-SQRT(3.D0)/6.D0))+1
            MIC=(M2+IC)-M2
            S=0.5/HALFM
      END IF
      IY=IY*IA
      IY=IY+IC
      IF(IY.LT.0)IY=(IY+M2)+M2
      URAND2=FLOAT(IY)*S
      RETURN
END


SUBROUTINE psump

      !----------------------------------------------------!
      ! This routine estimates the sump output             !
      ! as a function of the input and dilution rate       !
      !                                                    !
      !      uses 20 particle size classes                 !
      !                                                    !
      !      Jarrod Hart                April 1996         !
      !----------------------------------------------------!

      DOUBLE PRECISION cutsize, M_mp,mmp(20), M_sp,msp(20),Ms, error
      DOUBLE PRECISION tot2,Vs, Qmp, W, rhow, Qsp, Cssp, Csmp
      INTEGER i,counter

      OPEN (UNIT=11,FILE='datafiles/sprod.dat',STATUS='OLD',
     &      IOSTAT=istat)
      OPEN (UNIT=12, FILE='datafiles/mprod.dat',STATUS='OLD',
     &      IOSTAT=istat)
```

```
      OPEN (UNIT=15, FILE='datafiles/data.dat',STATUS='OLD',
     &       IOSTAT=istat)

      ! Read data

      READ (15,*) rhow                    ! Liquid density (kg/m^3)
      READ (15,*) rhos                    ! Solids density (kg/m^3)
      READ (15,*) Vs                      ! Vol slur sump (m^3)
      READ (15,*) Vm                      ! Vol mill (m^3)
      READ (15,*) Mm                      ! Mass sol mill (kg)
      READ (15,*) Ms                      ! Mass sol sump (kg)
      READ (15,*) Cssp                    ! Frac sol sump (kg/m^3)
      READ (15,*) Csmp                    ! Frac sol mill (kg/m^3)
      READ (15,*) W                       ! Sump dilution rate

      OPEN (UNIT=16,FILE='datafiles/count.dat',STATUS='OLD',
     &       IOSTAT=istat)
      READ (16,*) counter
      CLOSE (UNIT=16)

      ! Change W to maintain d50 (PI controller)

      OPEN (UNIT=17, FILE='datafiles/cutsz.dat',STATUS='OLD',
     &       IOSTAT=istat)
      READ (17,*) cutsize
      CLOSE (UNIT=17)
      OPEN (UNIT=18, FILE='datafiles/cutszerr.dat',STATUS='OLD',
     &       IOSTAT=istat)
      READ (18,*) error
      CLOSE (UNIT=18)

      !error=error+cutsize-75
      !IF (counter.GT.50) THEN
      !      W=2-0.0005*(cutsize-75)-0.0002*error
      !END IF

      IF (W.LT.0) THEN
            W=0
      END IF
      !print*, cutsize,error,W

      OPEN (UNIT=18, FILE='datafiles/cutszerr.dat',STATUS='OLD',
     &       IOSTAT=istat)
      write (18,*) error
      CLOSE (UNIT=18)


      ! Input

      READ (12,*) M_mp             ! Mass rate feed (kg/s)
      READ (12,*) Qmp              ! Vol rate feed (m^3/s)
      READ (11,*) M_sp             ! Mass rate feed (kg/s)
      READ (11,*) Qsp              ! Vol rate prod (m^3/s)
      !Qsp=Qmp+W                   ! Set to keep level constant
      Vs=Vs+Qmp+W                  ! Vol in sump (m^3)

      ! Use a PI controller to maintain sump level at 10m^3


      OPEN (UNIT=16, FILE='datafiles/leverr.dat',STATUS='OLD',
     &       IOSTAT=istat)

      READ (16,*) error
      error=error+Vs-10
      Qsp=0.5*(Vs-10)+0.5*error
      IF (Qsp.LT.1) THEN Qsp=1
      IF (Qsp.GT.10) THEN Qsp=10
      CLOSE (UNIT=16)
      OPEN (UNIT=16, FILE='datafiles/leverr.dat',STATUS='OLD',
     &       IOSTAT=istat)
      write (16,*) error
```

```
        CLOSE (UNIT=16)

        Vs=Vs-Qsp
        Csmp=M_mp/Qmp                  ! Frctn sols feed (kg/m^3)
        M_sp=Qsp*Cssp                  ! Mass rate solids dschg (kg/s)

        ! Import feed size distribution

        DO i=1,20
               READ  (12,*) mmp(i)
               !write(*,*)mmp(i)

        END DO

        ! Calculate other variables

        Cssp=(Cssp*Vs+Qmp*Csmp-Qsp*Cssp)/Vs        ! Frac sols in sump
        Ms=Ms+M_mp-M_sp                             ! Mass sol in sump (kg)

        CLOSE (UNIT=11)
        OPEN (UNIT=11,FILE='datafiles/sprod.dat',STATUS='OLD',
     &        IOSTAT=istat)

        write (11,*) M_sp
        write (11,*) Qsp

        tot2=0
        DO i=1, 20
               msp(i)=msp(i)+(M_mp*mmp(i)-M_sp*msp(i))/Ms
               tot2=tot2+msp(i)
        END DO
        DO i=1,20
               write(11,*) msp(i)/tot2
        END DO

        CLOSE (UNIT=15)
        OPEN (UNIT=15, FILE='datafiles/data.dat',STATUS='OLD',
     &           IOSTAT=istat)

        WRITE (15,*) rhow              ! Liquid density (kg/m^3)
        WRITE (15,*) rhos              ! Solids density (kg/m^3)
        WRITE (15,*) Vs                ! Vol slur sump (m^3)
        WRITE (15,*) Vm                ! Vol mill (m^3)
        WRITE (15,*) Mm                ! Mass sol mill (kg)
        WRITE (15,*) Ms                ! Mass sol sump (kg)
        WRITE (15,*) Cssp              ! Frac sol sump (kg/m^3)
        WRITE (15,*) Csmp              ! Frac sol mill (kg/m^3)
        WRITE (15,*) W                 ! Sump dilution rate

        CLOSE (UNIT=12)
        CLOSE (UNIT=11)
        CLOSE (UNIT=15)

END


SUBROUTINE pcyclon

        !------------------------------------------------------------------ !
        ! This routine estimates the selection within a hydrocyclone         !
        !                                                                    !
        !      uses 20 particle size classes                                 !
        !      requires 8 empirical constants                                !
        !                                                                    !
        !      Jarrod Hart                       April 1996                  !
        !------------- ---------------------------------------------------- !

        DOUBLE PRECISION a1,a2,a2,a3,a4,a5,a6,a7
        DOUBLE PRECISION WOF,WF,d50,Qc,fv,Yi(20),Ei(20),di(20),Rf
        DOUBLE PRECISION muf(20),mof(20),msp(20),mu(20),mo(20),mi(20)
        DOUBLE PRECISION M_uf,M_of,Quf,sz(20),cutsize
        INTEGER i
```

```
      ! Read in data

      OPEN (UNIT=12,FILE='datafiles/data.dat',STATUS='OLD',
     &       IOSTAT=istat)

      READ (12,*) rhol          ! Liquid density (kg/m^3)
      READ (12,*) rhos          ! Solids density (kg/m^3)

      CLOSE (UNIT=12)

      ! Read in msp(i)

      OPEN (UNIT=11,FILE='datafiles/sprod.dat',STATUS='OLD',
     &       IOSTAT=istat)


      READ (11,*) M_sp
      READ (11,*) Qsp

      DO i=1,20
             READ (11,*) msp(i)
      END DO

      CLOSE (UNIT=11)

      ! Input other values

      WF=Qsp-M_sp/rhos          ! Water feed rate (m^3/s)
      Qc=Qsp                    ! Volrate:pulp feed (l/s)

      rhom=(rhos*(Qc-WF)+rhol*WF)/Qc    ! Slurry density
      fv=(Qc-WF)/(Qc)                   ! Fractional solids volume

      a1=0.6                    !
      a2=0                      !
      a3=1.3                    !
      a4=9                      ! These constants have been
      a5=5                      ! chosen as a demonstration
      a6=6                      !
      a7=0.1                    !
      a8=0.05                           !

      WOF=a1*WF+a2              ! Water to overflow
      d50=exp(a3+a4*Qc+a5*fv)           ! Cutsize (microns)
      Rf=a7+a8*WOF/WF                   ! Fraction fines -> u-flow

      !write(*,*) Rf,d50,WOF

      ! write (*,*) d50,Rf

      DO i=1,20
             mi(i)=M_sp*msp(i)
      END DO

      OPEN (UNIT=17,FILE='datafiles/size.dat',STATUS='OLD',
     &       IOSTAT=istat)


      M_uf=0
      M_of=0
      DO i=1,20
             READ(17,*) di(i)
             Yi(i)=1-exp(-0.693*(di(i)/d50)*a6) !Fraction->u-flow
             Ei(i)=Yi(i)*(1-Rf)+Rf
             mu(i)=Ei(i)*mi(i)
             M_uf=M_uf+mu(i)
             mo(i)=(1-Ei(i))*mi(i)
             M_of=M_of+mo(i)
             !write (*,*) M_uf,mu(i)
      END DO
```

```
      CLOSE (UNIT=17)

      OPEN (UNIT=16,FILE='datafiles/count.dat',STATUS='OLD',
     &       IOSTAT=istat)
      READ (16,*) counter
      CLOSE (UNIT=16)

      IF (MOD(counter,100).EQ.0) THEN
             OPEN (UNIT=12,FILE='datafiles/size.dat',STATUS='OLD',
     &              IOSTAT=istat)

             DO i=1,19
                    READ (12,*) sz(i)
                    IF (Ei(i) .LT. 0.5) THEN
                           cutsize=sz(i)+(sz(i+1)-sz(i))*(0.5-Ei(i))/
     &                            (Ei(i+1)-Ei(i))
                    END IF
             END DO
             write(*,*) cutsize,' cutsize'
      END IF

      CLOSE (UNIT=12)

      DO i=1,20
             muf(i)=mu(i)/M_uf
             mof(i)=mo(i)/M_of
      END DO

      Qof=1000*WOF/rhol+M_of/rhos
      Quf=Qsp-Qof

      OPEN (UNIT=12,FILE='datafiles/cycuf.dat',STATUS='OLD',
     &       IOSTAT=istat)
      OPEN (UNIT=13, FILE='datafiles/cycof.dat',STATUS='OLD',
     &       IOSTAT=istat)

      write (12,*) M_uf
      write (12,*) Quf

      !OPEN (UNIT=23,FILE='datafiles/cutsz.dat',STATUS='OLD',
      !     &             IOSTAT=istat)

      !write (23,*) cutsize

      !CLOSE (UNIT=23)

      write (13,*) M_of
      write (13,*) Qof

      DO i=1,20
             write (12,*) muf(i)
             write (13,*) mof(i)
      END DO

      CLOSE (UNIT=12)
      CLOSE (UNIT=13)

END


SUBROUTINE pmixer

      !-----------------------------------------------------!
      ! This program mixes streams                          !
      !                                                     !
      !                                                     !
      !      Jarrod Hart                      April 1996    !
      !-----------------------------------------------------!

      DOUBLE PRECISION M_mf,mmf(20),M_f,mf(20),M_uf,muf(20)
      DOUBLE PRECISION Qf,Qmf,Quf,mi,totl
      INTEGER i
```

```
      OPEN (UNIT=11,FILE='datafiles/cycuf.dat',STATUS='OLD',
     &         IOSTAT=istat)
      OPEN (UNIT=12, FILE='datafiles/feed.dat',STATUS='OLD',
     &         IOSTAT=istat)
      OPEN (UNIT=13, FILE='datafiles/mfeed.dat',STATUS='OLD',
     &         IOSTAT=istat)

      ! Input

      READ (11,*) M_uf                    ! Mass rate feed (kg/s)
      READ (11,*) Quf                     ! Vol rate feed (m^3/s)

      READ (12,*) M_f                    ! Mass rate feed (kg/s)
      READ (12,*) Qf                     ! Vol rate feed (m^3/s)

      ! Import feed size distributions

      DO i=1,20
            READ  (11,*) muf(i)
      END DO

      DO i=1,20
            READ  (12,*) mf(i)
      END DO

      ! Calculate mill feed

      M_mf=M_f+M_uf
      Qmf=Qf+Quf

      write (13,*) M_mf
      write (13,*) Qmf

      !OPEN (UNIT=16,FILE='count.dat',STATUS='OLD',
      !     %                IOSTAT=istat)
      !READ (16,*) counter
      !IF (mod(counter,100).eq.0) THEN
      !      write (*,*) M_mf,Qmf,'  Mill feed'
      !END IF
      !CLOSE (UNIT=16)

      tot1=0
      DO i=1,20
            mi=muf(i)*M_uf+mf(i)*M_f
            mmf(i)=mi/M_mf
            tot1=tot1+mmf(i)
      END DO
      DO i=1,20
            write(13,*) mmf(i)/tot1
      END DO

      CLOSE (UNIT=11)
      CLOSE (UNIT=12)
      CLOSE (UNIT=13)

END


SUBROUTINE printout

      !-----------------------------------------------------!
      ! This program reads data from files, and prints      !
      ! them to screen and storage                          !
      !                                                     !
      !                                                     !
      !      Jarrod Hart                   May 1996         !
      !-----------------------------------------------------!

      DOUBLE PRECISION M(6),Q(6),mi(6,20),ave(6),
     &              d,mm,vm,ms,vs,sz(20),tot
      INTEGER i,counter
```

```
      OPEN (UNIT=11,FILE='datafiles/count.dat',STATUS='OLD',
     &       IOSTAT=istat)
      READ (11,*) counter
      CLOSE (UNIT=11)

      OPEN (UNIT=11,FILE='datafiles/size.dat',STATUS='OLD',
     &       IOSTAT=istat)
      DO i=1,20
             READ (11,*) sz(i)
      END DO
      CLOSE (UNIT=11)

      OPEN (UNIT=12, FILE='datafiles/feed.dat',STATUS='OLD',
     &       IOSTAT=istat)

      READ (12,*) M(1)
      READ (12,*) Q(1)
      DO i=1,20
             READ(12,*) mi(1,i)
      END DO
      CLOSE (UNIT=12)

      OPEN (UNIT=12, FILE='datafiles/mfeed.dat',STATUS='OLD',
     &       IOSTAT=istat)

      READ (12,*) M(2)
      READ (12,*) Q(2)
      DO i=1,20
             READ(12,*) mi(2,i)
      END DO
      CLOSE (UNIT=12)

      OPEN (UNIT=12, FILE='datafiles/mprod.dat',STATUS='OLD',
     &       IOSTAT=istat)

      READ (12,*) M(3)
      READ (12,*) Q(3)
      DO i=1,20
             READ(12,*) mi(3,i)
      END DO
      CLOSE (UNIT=12)

      OPEN (UNIT=12, FILE='datafiles/sprod.dat',STATUS='OLD',
     &       IOSTAT=istat)

      READ (12,*) M(4)
      READ (12,*) Q(4)
      DO i=1,20
             READ(12,*) mi(4,i)
      END DO
      CLOSE (UNIT=12)

      OPEN (UNIT=12, FILE='datafiles/cycof.dat',STATUS='OLD',
     &       IOSTAT=istat)

      READ (12,*) M(5)
      READ (12,*) Q(5)
      DO i=1,20
             READ(12,*) mi(5,i)
      END DO
      CLOSE (UNIT=12)

      OPEN (UNIT=12, FILE='datafiles/cycuf.dat',STATUS='OLD',
     &       IOSTAT=istat)

      READ (12,*) M(6)
      READ (12,*) Q(6)
      DO i=1,20
             READ(12,*) mi(6,i)
      END DO
      CLOSE (UNIT=12)
```

```
      OPEN (UNIT=12, FILE='datafiles/data.dat',STATUS='OLD',
     &        IOSTAT=istat)

      READ (12,*) d
      READ (12,*) d
      READ (12,*) vs
      READ (12,*) vm
      READ (12,*) mm
      READ (12,*) ms
      READ (12,*) d
      READ (12,*) d
      READ (12,*) d
      CLOSE (UNIT=12)

      DO j=1,6
             tot=0
             d=1
             DO i=1,19
                    tot=tot+mi(j,i)
                    IF (tot .GT. 0.5 .AND. d .EQ. 1) THEN
                           ave(j)=sz(i-1)+
     &                     (sz(i)-sz(i-1))*(0.5-tot1)/mi(j,i)
                           d=0
                    END IF
             END DO
      END DO

      write (41,*) counter,M(1),Q(1),ave(1)
      write (42,*) counter,M(2),Q(2),ave(2)
      write (43,*) counter,M(3),Q(3),ave(3)
      write (44,*) counter,M(4),Q(4),ave(4)
      write (45,*) counter,M(5),Q(5),ave(5)
      write (46,*) counter,M(6),Q(6),ave(6)
      write (47,*) counter,mm,vm,ms,vs

      IF (mod(counter,100).eq.0) THEN
             !DO i=1,20
                    !write (48,*) sz(i),(mi(j,i),j=1,6)
             !END DO
             write (*,*) counter
             !write (*,*) M(1),Q(1),ave(1),' Feed'
             write (*,*) M(5),Q(5),ave(5),' Prod'
             !write (*,'(6f8.2)') (ave(j),j=1,6)
      END IF

END
```

Discussion

The program assumes the popular modular form where each section of the plant is modelled separately, and is combined by solving each in sequence. This solution is made more interesting in cases where feedback occurs, as is the case in milling circuits.

Here, a module generates the feed (subroutine PFEED) which is then used by the subroutine PMILL to determine the size and mass flow distributions leaving the mill. This output in turn forms the input to the sump, and the sump's output forms the input to the cyclone. The cyclone model then predicts a split and the first estimate for the product is generated. The cyclone underflow, however, is returned and added to the feed that went into the mill subroutine. The mill recalculates its output and the sequence repeats as before, generating a new guess for the product stream. This is repeated until the estimate of the product stream no longer changes (a simple successive substitution type solution). This then forms a steady state model of the plant. For dynamic modelling, the solution procedure needs to be set-up to repeat once for every time step, as inputs are varied. The correct mass flow dynamics are ensured by the cumulative mass terms in each unit, while subtler dynamics such as the effect of load momentum on power consumption are not considered. For this reason such a model is unsuitable for study of plant behaviour in a effort to better understand fundamental processes occurring in the mill. It merely serves as a pseudo-mill upon which the user could 'inflict' process control.

There are several ways to improve upon this model; firstly one could utilize one of many solution algorithms to speed up convergence in the solution step; secondly, better plant models could be utilized for elements like the cyclone, the sump and the mill. Finally, the simulator could be made more user friendly by creating a user interface, where inputs and settings can be altered without the need to recompile (or even restart).

All of these aspects are indeed handled well by the simulator developed by Mintek which was used in the bulk of this project. Since this programme is confidential, this far simpler program was included to give a broad view of milling simulation, from the programming perspective. It may actually be more practical for the reader, since the Mintek code is written in Visual Basic and is therefore difficult to interpret outside of the context of forms and modules. This programme also has the advantage that there are no 'frills' to complicate the understanding.

The other aspect of programming important to this thesis is the coding of controllers. In order to provide some insight, the next section provides some of the control related subroutines written into the Mintek simulator during the course of the project.

A.1.2 Control Related Subroutines

Many control actions are extremely simple to code, since, once the controller is tuned by the proper selection of gains etc., the implementation takes this simple form:

```
error = (setpoint - measured_variable)
cumulative_error = cumulative_error + error
new_input = nominal_input + k*error + (k/tau)*cumulative_error
```

This is the form for a PI (proportional integral) action controller. This type of programming is well documented and therefore, the focus of this section is to describe some of the other methods utilized by the optimizing controller in the programme. Naturally, many routines involve simple tricks which do not warrant discussion in this context, such as the generation of input trajectories via the use of a random number generator. These are well documented elsewhere, and are indeed a built-in function

Visual Basic. Some other routines which may be of interest are however included here. These sections are written in Visual Basic.

Simple filter

```
' We need: d50 of cyclone feed (fmd50)
'          flow of cyclone feed (fmcycf)
'          % solids of cyclone feed (fmcycd)
' and we need the values to change smoothly to reduce
' unnecessary control action. This section od code
' does that job. It provides a variable for each of these
' parameters that represents its 'running' average for the
' last however long (50 steps in this case).

fd50 = 0
fcycf = 0
fcycd = 0
For i = 50 To 2 Step -1
 fmd50(i) = fmd50(i - 1)
 fmcycf(i) = fmcycf(i - 1)
 fmcycd(i) = fmcycd(i - 1)
 fd50 = fd50 + fmd50(i)
 fcycf = fcycf + fmcycf(i)
 fcycd = fcycd + fmcycd(i)
Next i
fd50 = fd50 + fmd50(1)
fcycf = fcycf + fmcycf(1)
fcycd = fcycd + fmcycd(1)
fd50 = fd50 / 50
fcycf = fcycf / 50
fcycd = fcycd / 50
circuit!fd50.Text = Format(fd50, "##0.000")
circuit!fcycf.Text = Format(fcycf, "##0.000")
circuit!fcycd.Text = Format(100 * fcycd, "##0.000")
```

This code simply 'smoothes out' wrinkles in plant data by taking a running average.

Average Size Calculation Code

In the code, flows are calculated for each of 20 size classes. Any particular sample is therefore the sum of the masses in all the size classes. The code describes a sample with a cumulative % passing vector. To calculate the *average* size of a sample it is necessary to step through the vector `cumsizedist(i, j)` until 50% is passed and then interpolate to determine exactly where in the vector the value passes 50 %. This value is then used to read the average size from the size vector `sizc(k)`.

This algorithm calculates the mean size of all the streams. `avesiz(j)` is the resulting vector of average sizes for the 7 streams.

```
Sub avesize()
For j = 1 To 7        ! for streams 1 to 7
 If j = 4 Then j = 5  ! there is no stream 4 in this case
 For i = 1 To nsizes  ! number of size classes
  If cumsizedist(i, j) < 50 Then
   k = i
   i = nsizes
  Else
   k = nsizes
  End If
 Next i
 If k < nsizes Then
  avesiz(j) = sizc(k - 1) - (cumsizedist(k - 1, j) - 50) /
  (cumsizedist(k - 1, j) - cumsizedist(k, j)) * (sizc(k - 1) -
```

```
        sizc(k))
       Else
        avesiz(j) = 50 / cumsizedist(k, j) * sizc(k)
       End If
      Next j
      End Sub
```

Optimizing Control - Hill Climbing Technique

The hill-climbing technique simply alters the load setpoint (`jtotset`) based on whether the power consumption had increased or decreased as a result of the previous step. In this algorithm it is worth noting the care taken to dermine the *sign* as well as the *size* of the step. The section of code that alters the size of the step is optional and is discussed in detail in Chapter 4.

```
Sub powermax()
 'Optimizing contoller optimizes mill power
 If jtotset < 0 Then jtotset = 0.4 ! safety catch
 If circuit!flexistep.Value = 1 And Abs(missastep - 1) < 0.1 Then ! Flexible
                                                                  ! step
                                                                  ! sizes?
  jtchange = (0.001) * (1 - Exp(-Abs(power - powerold))) + 0.002 *
                                     (Abs(power - powerold)) ^ 0.4
 Else
  missastep = 1
  jtchange = 0.01
 End If
 If power > powerold Then
  jtotset = jtotset + action * jtchange
 Else
  jtotset = jtotset - action * jtchange
  action = action * (-1)
 End If
 circuit!jtset.Text = Format(jtotset, "0.0000") ! show change on screen
 powerold = power
End Sub
```

Optimizing Control - Model Based Method

The code that represents the optimizing controller was unfortunately dispersed amongst the simulator code during the development stages. The most important section of code concerns the plant model, a two variable quadratic equation, and the way it adapts to match the plant as the plant changes in the face of long-term feed disturbances as well as wear and tear. The subroutine `adapter` performs this task:

```
Sub adapter()
' This routine takes plant readings (filtered values) and
' supplies coefficients for a 3d function;
'                throughput=f(viscosity, load)
' The function is bi-parabolic and has the form:
'   z=c0+c1x+c2y+c3x^2+c4y^2+c5xy+c6xy^2+c7x^2y+c8x^2y^2
' The task is to return c0-8. These are then used to supply
' a new load setpoint.
' We can determine c0-8 from a finite number of datapoints,
' say 50. But the fifty most recent datapoints are not
' necessarily good points to use since they may represent
' a very small range of possibilities. We need a compromise.
' The first idea is to have a single datapoint from each
' 'region'. This means that while operating in that region
' it is possible to alter that particular region only. The
' curve would then be a best fit of those points only.
' We can identify n regions (we use 15, more may be used
' for a real plant), and easily get c0-8 from these using
```

```
' linear regression. The task is then to take each new
' datapoint, find out which region it lies in, and add it
' to that points "50" and remove a fiftieth of the existing
' value.
'
'

' Part 1: set of ifs to determine region of operation:
'-----------------------------------------------------------------------

If Jtotal > 0.275 And Jtotal < 0.425 And visc > 0.0025 And visc < 0.0075
Then
 i = 3
 j = 5
 If Jtotal < 0.375 Then i = 2
 If Jtotal < 0.325 Then i = 1
 If visc < 0.0065 Then j = 4
 If visc < 0.0055 Then j = 3
 If visc < 0.0045 Then j = 2
 If visc < 0.0035 Then j = 1
 tpdata(3 * (j - 1) + i) = tpdata(3 * (j - 1) + i) * 0.98 + 0.02 * ftp
 ' NB this line incurs a 'stiffness' on the changes
 ' For i = 1 To 15
 '  tpdata(i) = tpdata(i) - 0.0001 * (tpdata(i) - ftp)
 ' Next i
End If

' Part 2: Now we calculate the coefficients of the equation...
'-----------------------------------------------------------------------

' The translating matrix (ataat)was read in in the init section...
' Now to mutiply the matrices:
For i = 1 To 9 ' Matrix Rows
 cdata(i) = 0
 For j = 1 To 15 ' Matrix Columns
  cdata(i) = cdata(i) + ataat(i, j) * tpdata(j)
 Next j
 'If (timel Mod 6 * 3600) = 0 Then
 ' Print #5, cdata(i)
 'End If
Next i

' Part 3: Optimising controller uses the model and
' the viscosity estimate to read off optimal load setpoint.
'-----------------------------------------------------------------------

' Optimizing controller starts here:
' We must use the c values in the equation.
' Viscosity is set, so we are left with a quadratic
' in load.
' From simple calculus, the extremum of a qaudratic
' is given by this formula:
'                       x=-b/2a
'      where            y=ax^2+bx+c
' First we get "a"
' a=c3+c7*visc+c8*visc^2
' Then "b"
' b=c1+c5*visc+c6*visc^2

If (timel Mod 2 * 3600) = 0 Then
 jtotset = -(cdata(2) + cdata(6) * visc + cdata(7) * visc ^ 2) / (2 *
 (cdata(4) + cdata(8) * visc + cdata(9) * visc ^ 2))
 circuit.jtset.Text = Format(jtotset, "0.000")
End If
End Sub
```

Explanations are given in the form of comments in the code.

It must be noted that a translating matrix termed `ataat` is utilized in this section of code. In the least squares problem:

If $\mathbf{Ax} = \mathbf{b}$ is overdetermined then the least squares solution $\bar{\mathbf{x}}$ is given by $\bar{\mathbf{x}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$

`ataat` is simply $(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$.

It is worth noting that $(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$ only needs to be solved once for a particular plant due to the fixed nature of the 'grid' used to generate the model. This greatly improves the computational efficiency of the controller code.

The routine is seeded with an estimate of the plant model the first time it is operated (on subsequent operation the most recent coefficient set may be used). The code is designed to manipulate the model as data becomes available to initially obtain the true form of the surface and then to track changes in the plant.