1-1-2006

Minimum variance tuning of PI controllers using hybrid genetic algorithms

Joseph Hanna
Ryerson University

MINIMUM VARIANCE TUNING OF PI CONTROLLERS USING HYBRID GENETIC ALGORITHMS

by

Joseph Hanna, B. Sc.

Toronto, Ontario, Canada

A dissertation

presented to Ryerson University

in partial fulfillment for the degree of

Master of Engineering

in the Program of Chemical Engineering

Faculty of Engineering

Ryerson University

Toronto, Ontario, Canada, 2006

Author's Declaration

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

\-----   ---------

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

\-----------

Borrower's Page

Ryerson University requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Name	Signature	Address	Date

Minimum Variance Tuning of PI Controllers using Hybrid Genetic Algorithms

Master Degree Dissertation 2006, © Joseph Hanna
Master of Chemical Engineering, Ryerson University

Abstract

One of the main confronts in control engineering is the assessment of closed loop performance. Harris ascertains a performance index where the best performance is assumed to be attained by a minimum variance controller.

This research spotlights on the tuning of the illustrious and most frequently used PI controller to achieve minimum variance conditions. The optimization problem is embarked upon two different approaches. The first approach uses enumerative search optimization for its simplicity. The second approach applies an exploited hybrid genetic algorithm that is developed to generate vigorous and premium results. The algorithm amalgamates the genetic operations of selection, crossover, and mutation with Newton's search inside successively expanding and contracting parameter domains using alternating logarithmic and linear mappings. Finally, the obtained PI parameters are tested and simulated with data from three control loops at Falconbridge Smelter in Sudbury and compared with the existing tuning parameters. The new parameters yield optimal results.

ACKNOWLEDGEMENT

I would like to thank Dr. Simant R. Upreti for his technical assistance during my master studies. Dr. Upreti was of a great help in my research. Furthermore, he has always provided me with useful suggestions while leaving me a remarkable degree of freedom to pursue my own direction of study on my thesis topic. In addition, I would like to thank Dr. Ali Lohi for his guidance and mentorship.

Joseph Hanna

December, 2005

DEDICATION

I would like to dedicate this work to my wife and my parents for their love and care in every step of my life.

Table of Contents

List of Tables

List of Figures

Nomenclature, Abbreviations and Acronyms

D	Derivative time
$E\|_t$	'Expectation' provisional upon data available up to and including current time, t
I	Reset rate, repeats / seconds, Hertz
$\hat{I}_1$, I_1	Objective Function
I_2	Constrained Objective Function
J_{ac}	Jacobian Matrix
J_{MV}	Minimum Variance
K	The process gain
K_i	Controller integral gain, repeats / seconds, Hertz
$K_{i,max}$	Controller integral gain lower bound
$K_{i,min}$	Controller integral gain higher bound
K_p	The controller proportional gain
$K_{p,max}$	Controller proportional gain lower bound
$K_{p,min}$	Controller proportional gain higher bound
M_{se}	Mean square error
N_{gen}	Number of generations
n_i, N_i	Input process disturbance.
n_o, N_o	Output disturbance.
N_{pop}	Number of solutions or chromosomes in the population
N_x	Number of optimization parameters
N_{xsites}	Number of cross-over sites
P	Proportional Gain
p_{cros}	Crossover Probability
p_{mut}	Mutation Probability
Q	Noise covariance matrix

R_i	Pseudo-random number
t	Time, seconds
T	Sampling Time, seconds
U(t)	The control effort
W(t)	Set point
$\bar{x}$	Mean Value of Optimization Parameters
$\hat{x}$	Optimal Vector of Optimization Parameters
Y	Process Output
Y_{max}	Process Output lower bound
Y_{min}	Process Output higher bound
x	Optimization Parameters
z	Discrete Operator

Greek Symbols

$\xi(t)$	Random zero-mean sequence
σ^2	Variance
Δ	Deviation
λ	System Time Delay
Σ	Output covariance matrix

Abbreviations

FL	Falconbridge Loop
H_{PI}	Harris Performance Index
H_{NPI}	Harris Normalized Performance Index
P_i	Performance Index
PI	Proportional Integral Controller
N_{Pi}	Normalized Performance Index

Part I

INTRODUCTION

Chapter 1

Introduction

The design and implementation of different process control strategies have been largely pondering to the controller designer's hard work. The underlying principle has been that systems, which are difficult to control, need strategies that are more robust in order to achieve the optimal performance. Although there are a variety of control strategies, proportional-integral-derivative (PID) controllers are still the most widely used for their simplicity and technological maturity. It is nearly impossible to monitor the performance of all PID control loops in any manufacturing facility. However, it is more realistic to monitor most critical control loops with some formalized assessment tools. When a poorly performing control loop is identified, it becomes necessary to diagnose the underlying cause. For example, the poor performance could be due to improper tuning, improper controller structure, changing process dynamics, or an excursion due to process disturbance. The emerging area of performance assessment provides a means of diagnosing control loop performance using time series and digital signal processing techniques - Huang et al. (1997). The benefits of continual loop monitoring and performance assessment are becoming well known to the industry. Controller monitoring can successfully find problem areas within the plant.

In most modern industrial process plants, a sophisticated and powerful Distributed Control System (DCS) or Programmable Logic Controllers (PLC) exist. These systems have allowed an explosive growth in the amount of process data that are available to analyze and benchmarking of PID tuning parameters.

The focal point of this project is the solution of a stochastic regulatory control problem. The minimum variance controller is a benchmark for this control problem. Harris (1989) established a new benchmark for performance assessment of control systems as the best achievable control is measured by mean square error. Astrom (1971) had this same idea; however, it was Harris that made a simple calculation technique using time series data.

Harris (1992) et al. commenced working on a normalized performance index to characterize the performance of feedback control schemes against a benchmark of minimum variance control.

In the present study, a PI tuning technique based on minimum variance control is introduced, modeled, and tested. The tuning technique maximizes a Normalized Performance Index (N_{Pi}) by two optimization techniques. The first technique is an enumerative search technique for the N_{PI} maximum. The second technique involves the utilization of Hybrid Genetic Algorithms (GA), to ensure the achievement of a global solution of optimal minimum variance tuning parameters. Both approaches are modeled, tested, simulated, and compared.

The focus of most of the previously conducted research stressed on the effect of output stochastic disturbance. In a typical industrial application, noisy measurement is one of the main sources of output stochastic disturbance, which can be easily filtered through a variety of digital filters such as first order low-pass, band-pass, band-stop and other higher order filter. A PI controller itself can act as a first order low-pass filter.

The main objective of this project is to design a PI feedback regulatory controller that can minimize the output variance and to reduce the effect of this stochastic input noise. Stochastic input disturbance is selected because it is more realistic.

The optimization problem is non-linear, discontinuous, and not necessarily multi-model. In addition, the constraints for this optimization problem are mainly implicit. Genetic Algorithms are perfect candidates to solve such problems. Although genetic algorithms are computationally efficient in seeking the optimum point precisely, they cannot locate the exact global optimum reliably. Hybridization is used in order to eliminate the weakness of GA by incorporating Newton's method to enhance its local optimizing ability.

1.1 Literature Review

Astrom (1970), in his book "Introduction to Stochastic Control Theory", has developed the linear stochastic control theory where he stated that "The performance of the system depends critically on the information available at the time the value of the control signal should be determined. For example, a delay of the measured signal will lead directly to deterioration of the performance". In addition, Astrom started the minimum variance control theory which articulates that "For a linear time invariant system with one input and one output, the regulation problem can be solved assuming that the disturbance acting on the system can be described as a realization of a normal stationary stochastic process with a rational spectral density, and the purpose of control is to minimize the variance of the output. An optimal predictor can be thought to be consisting of two parts: one predictor which predicts the effect of the process disturbance on the output and one dead beat regulator which computes the control signal required to make the predicted output equal to the desired value". This theory was the heart of minimum variance control.

Harris (1989) further explored Astrom's work; and initiated the concept of the use of minimum variance as a benchmark to measure controller's performance. His work comprehended that the theoretical best achievable control as measured by the mean square error is the benchmark for controller efficiency. If the theoretical best achievable control represents a significant improvement over the current performance, alternate controller tuning or feedback control strategies can be considered if this improved performance is acceptable. However, the best achievable performance itself may not be adequate. In these cases, alternate approaches such as feedforward control, reduction of dead time, and different loop pairings must be used to achieve the reduction in variance. In order to estimate the theoretically achievable minimum variance performance, it is necessary to predict the integer number of the delay samples plus one ahead of the calculating time, which is exactly what was suggested by Astrom.

The strength in Harris' work is his simple approach to calculate the best achievable control, which is the minimum variance control, by fitting univariant time series to process data collected under routine control. Harris illustrated that the lower bound on the closed loop variance could be obtained by analyzing the closed loop data. The lower bound performance is calculated by solving a system of linear equations (The Diophantine equation). A priori knowledge about the system time delay must be identified.

Shinskey (1988) introduced an alternative approach, which is widely used in PID controller tuning. He called it the absolute performance index, (API), which assesses the performance of PID-type controller in terms of integrated absolute error (IAE).

Astrom et al. (1990) presented a means for assessing achievable performance using PID control. Achievable performance is characterized in terms of bandwidth and dimensionless numbers such as normalized peak error for set point and load disturbance and rise time.

Desborough et al. (1992) introduced the normalized performance index to characterize the performance of feedback control schemes against a benchmark of minimum variance control. This normalized performance index is a number between zero and one; with zero indicating that the process is operating under minimum variance control, i.e. its optimal bound. Spectral density analysis was used in counter part with recursive least squares to estimate the performance index online which enables the use of control charts to monitor changes in performance.

Harris (1993) expanded his work to include feedforward/feedback systems. He developed a performance index to assess the performance of the whole control scheme. The paper was concerned mainly about multi-input single-output (MISO) systems where there is not any cross correlation among the unmeasured and measured disturbances.

Stanfeli et al. (1993) commenced some research about monitoring and diagnosing of process control performance. The paper introduced a hierarchical method for monitoring and diagnosing the performance of a single cascade control loop that is composed of a feedback and feedforward control loops. The analysis was done based on typical operating plant data. They used the IAE (Integrated Absolute Error), maximum deviation and decay ratio. Subsequently they designed the best achievable performance with the existing control structure, and identified the required steps to improve the current performance. The strength of their work was in the methodology that they have used to achieve the above-mentioned goals in addition to the ability to isolate whether the poor performance is due to the feedforward loop or the feedback loop.

Harris et al. (1996) extended their previous work to include Multi-input Multi-output (MIMO) systems. They introduced a technique for controller performance assessment of MIMO systems. A minimum variance controller was used as a lower bound against the current performance. The process involved the estimation of the process interactor matrix, which is characterizing the dead-time structure of the process. They introduced a quick non-parametric correlation test for the closeness of the current control system performance to the theoretical lower bound. Finally, they made a quantitative analysis that provides an estimation of the theoretical lower bound on variance-covariance matrix of the controlled variables, and on the quadratic performance objective of the minimum variance controller.

In the same year, Huang et al. introduced some development work in the same area of performance assessment of MIMO systems. Afterwards, they presented a research (1997) on a new approach in performance assessment, which is based on filtering and correlation analysis of the process output and filtered data. They used a whitening filter analogous to time series modeling, where the final test of adequacy of the model consists of checking if the residuals are white. In contrast to time series modeling where the estimation of the model is of interest, the residual or the innovation sequence was the main item of interest in this whitening process. By knowing this residual white noise, the minimum variance, which is the lower bound for performance assessment, can be estimated easily.

Horch et al. (1999) have modified Harris performance index. The minimum variance control is based on cancellation of the model dynamics and so placing all closed poles in the origin. This makes the minimum variance controller non applicable in practical life because of its low tracking capability. They suggested that one pole can be placed using either control design guideline or additionally available process knowledge. The lower bound for process output minimum variance was changed and therefore reflected on performance index calculations.

Seppala et al. (2002) changed the direction of performance monitoring for multivariable systems from estimating the control invariant component of the closed loop output covariance to the assessment of dynamic analysis side of multivariable process and the assessment of the dynamic interactions between loops. They simplified the multi-output dynamic analysis problem by treating the tracking error trends as a vector process of endogenous stochastic variables and using vector autoregressive structure to model dynamic relationships. Once such a model has been estimated, a host post-estimation diagnostic, such as a multi impulse response analysis, can be used to interpret the dynamic interactions between the tracking errors.

Ko et al. (2004) presented a research about the utilization of the same principle to tune a PI control loop, which is similar to the work presented here. However, their approach assumed a pre-defined transfer function for the output stochastic disturbance, which is almost impossible to determine in an industrial application. In addition, they based their model on minimization of process output that is subjected to output disturbance that is different from our case, which considers input stochastic disturbance.

The potential of minimum variance control is clear from what is shown above. The directions that different researchers have taken show the different possible applications of this approach. These directions are further explored and improved to offer means by which the mostly commonly used PI controller can be utilized to provide minimum variance performance.

1.2 Thesis Overview

This thesis is mainly divided into three parts besides the introduction and conclusion. The first main topic reveals the theoretical background behind this research. The second part is the main core contribution of the research, which details the mathematical derivations and the different approaches that were used to outline the objective function and constraints into an explicit mathematical form. The third part demonstrates the results that were achieved by the two different optimization techniques, which are Hybrid Genetic Algorithms and Enumerative search. In addition, a detailed sensitivity analysis was carried out to examine the robustness of the calculated performance index based on changes in different process parameters.

Chapter One describes the research objective and gives the reader some knowledge about basic concepts and terminology that are being used through this thesis. In addition, this chapter documents the investigations that were done by other researchers in the same field.

Chapter Two is intended to detail minimum variance control, and discusses the actual implementation of this controller on process models.

Chapter Three presents genetic algorithm and its different operators. It details the coding and programming of such technique. In addition, it describes the optimization algorithm that is being used in this research.

Chapter Four presents the mathematical modelings for the performance index as well as the output variance that is used as objective functions in the optimization processes. This is a core part of the research that elaborates different mathematical and control principles such as discrete state space, optimization, and statistics.

Chapter Five is considered to be a sensitivity analysis for the solution robustness in an industrial application. In any industrial application, the process parameters change with time, i.e., heat exchanger fouling or control valve sticking. This chapter examines the effect of the changes in these process parameters on the performance index.

Chapter Six is dedicated to the PI tuning approach, which is based on output minimization or performance index maximization. It compares three cases from an industrial application at Falconbridge's., Sudbury smelter.

Chapter Seven concludes the thesis by comparing the main results that have been reached. Moreover, it includes proposals for future work.

Part II

Theoretical Exploration and Programming Description

Chapter 2

Minimum Variance Controller and Harris Performance Index Calculations

2.1 Definition

The key issue in many control systems devise is to design a controller that reduces disturbance, which is stochastic in many cases. For linear stochastic systems, the process can be described by a pulse function and the disturbances acting on the system are filtered white noise. Minimum variance control refers to a specific optimal control problem for these linear stochastic systems described by input-output relations.

The minimum variance control can be designed by fitting univariant time series to process data collected under routine control. The lower bound on the closed loop variance could be obtained by analyzing the closed loop data. The lower bound performance is calculated by solving a system of linear equations [The Diophantine

Equation – Harris (1989)]. Prior knowledge about the system time delay must be known.

The minimum variance controllers are not recommended for all control applications. They can be used when

1. Only a rough guideline is needed and simplicity is a premium
2. Variations of control signal and actuator movements are not important
3. Very tight control of the process output is demanded without concerns for power or energy used

Inventory control loops such as pressure, where a tight control is required and many stochastic disturbances exist, are good applications for minimum variance control.

On the other hand, minimum variance control is not recommended when

1. There are tight limits on actuator movements and control signal variations
2. The controller is in practice restricted to being low order and of a particular form
3. The cost involved is not simply an output variance or tracking error

Cascade control loops are good examples of where minimum variance controls should not be used. The master controller output shall ascertain changes on the slave loop set point, which should not be moving aggressively. In the mean time, the slave loop should be designed mainly for tracking rather than disturbance rejection.

A limitation of minimum variance controller is that its properties are dependent on sampling time and its implementation in control systems is not necessarily straightforward. In addition, time delay and process model should be known prior to the controller's design.

2.2 Derivation of Minimum Variance Control Law

The minimum variance controller provides a control signal $u(t)$ that minimizes the following performance objective function:

$$J_{MV} = E[w(t) - y(t+k)]^2\Big|_t \tag{2.1}$$

The notation $E|_t$ denotes an 'expectation' provisional upon data available up to and including current time, t. In addition, at time equal to zero all data are equal to zero. Taking the expected value of a variable squared gives the variance of that variable. In this case, J_{MV} therefore refers to the variance of the error between set-point $w(t)$ and the controlled output k-time steps in the future, $y(t+k)$. The desired controller is thus one that minimizes this variance, hence the name Minimum Variance control.

In order to enable minimization of Eq. (2.1) with respect to $u(t)$, which is the controller effort, first we need to be able to relate the controlled output y to the manipulated input u. This is made available via a process model, the simplest of which is the CARMA (Controlled Auto-Regressive Moving Average) model:

$$Ay(t) = z^{-k}Bu(t) + C\xi(t) \tag{2.2}$$

Equation (2.2) is an ARMAX or a CARMA model, where k is the time delay of the process, expressed as an integer multiple of the sampling interval T and A (z), B (z) and C (z) are polynomials in z^{-1}. That is:

$$A(z)=1+a_1z^{-1}+a_2z^{-2}+.....+a_{N_A}z^{-N_A}, N_A=\deg(A(z))$$
$$B(z)=1+b_1z^{-1}+b_2z^{-2}+.....+b_{N_B}z^{-N_B}, N_B=\deg(B(z))$$
$$C(z)=1+c_1z^{-1}+c_2z^{-2}+.....+c_{N_C}z^{-N_C}, N_C=\deg(C(z))$$

2.3

where $\xi(t)$ is a random zero-mean sequence with finite variance σ^2. That is,

$$E\{\xi(t)\}=0$$

and

$$E\{\xi(t)^2\}=\sigma^2$$

2.4

However, the objective function involves a term in the future, namely $y(t+k)$, which is not available at time t. Therefore the minimization cannot be performed unless we can replace $y(t+k)$ with a realizable estimate. This can be achieved via the use of the following identity:

$$C = EA + z^{-k}F$$

2.5 a

where E and F are also polynomials in z^{-1}. This identity, which is known in mathematics as the polynomial division identity, gives essentially the quotient and the remainder of the division of two polynomials. In this case

$$\frac{C}{A} = E + z^{-k}\frac{F}{A}$$

2.5 b

with $\deg(E) = k-1$.

Equation (2.5) assumes further significance in that it enables the separation of current and past values from future values. As a result, Equation (2.5 b) is also known as the 'Separation Identity'. To see how this is accomplished, multiply E into Equation (2.2). This gives

$$EAy(t) = z^{-k}EBu(t) + CE\xi(t) \tag{2.6}$$

Using Equation (2.5a) to substitute for EA in Equation (2.6), we get

$$\left(C - z^{-k}F\right)y(t) = z^{-k}EBu(t) + CE\xi(t) \tag{2.7a}$$

Time shift Equation (2.7a) k-steps into the future by multiplying z^{k} to give

$$\left(C - z^{-k}F\right)y(t+k) = EBu(t) + CE\xi(t+k) \tag{2.7b}$$

Next, separate out terms involving future values to the right-hand-side, and terms involving past and current values to the left-hand-side:

$$Cy(t+k) - CE\xi(t+k) = EBu(t) + Fy(t) \tag{2.8}$$

Defining

$$y^{*}(t+k \mid t) = y(t+k) - E\xi(t+k) \tag{2.9}$$

Then obtain the "*k*-step-ahead" predictor of y(t) as

$$Cy^*(t+k \mid t) = EBu(t) + Fy(t) \tag{2.10}$$

Now we can use $y^*(t+k|t)$ in place of $y\ (t+k)$ in the objective function, since it is a function of past and current values of y and u only, as signified by the index $(t+k|t)$. Since only $y^*(t+k|t)$ is needed, we re-arrange Equation (2.10) into a more suitable form, as

$$y^*(t+k \mid t) = EBu(t) + Fy(t) + Hy^*(t+k-1 \mid t-1) \tag{2.11}$$

Where H is another polynomial in z^{-1} defined as

$$H = (1-C)z \tag{2.12}$$

Substituting $y^*(t+k|t)$ for $y\ (t+k)$ in the objective function gives

$$\begin{aligned} J_{MV} &\cong E\left[w(t) - y^*(t+k \mid t)\right]^2\Big|_t \\ &= E\left[w(t) - EBu(t) - Fy(t) - Hy^*(t+k-1 \mid t-1)\right]^2\Big|_t \end{aligned} \tag{2.13}$$

When minimizing J_{MV} w.r.t. $u\ (t)$, we are seeking a $u\ (t)$ that will set

$$\frac{\partial J_{MV}}{\partial u(t)} = -2e_o b_o \left[w(t) - EBu(t) - Fy(t) - Hy^*(t+k-1 \mid t-1)\right] = 0$$

2.14

where e_o and b_o are the initial conditions for E and B

From equation (2.14), it is clear that the required control is

$$u(t) = \left[w(t) - Fy(t) - Hy^*(t+k-1 \mid t-1)\right] / EB$$

2.15

2.3 Implementation of minimum Variance Control

In calculating $u(t)$ from Equation (2.15), we require the coefficients of the *F, H, E* and *B* polynomials. To simplify matters, we define the product *EB* to be *G*, thus reducing the number of polynomials from 4 to 3. Using this new nomenclature, the *k*-step-ahead predictor $y^*(t+k|t)$ becomes:

$$y^*(t+k \mid t) = Gu(t) + Fy(t) + Hy^*(t+k-1 \mid t-1)$$

2.16

The control signal is calculated as:

$$u(t) = \frac{1}{g_o}\left[w(t) - \sum_{i=1}^{N_o} g_i u(t-i) - Fy(t) - Hy^*(t+k-1 \mid t-1) \right]$$

2.17

If we do not know the coefficients of *F*, *H*, *E* and *B* polynomials, they will have to be estimated from process input-output data. Time-shifting Equation (2.16) k-time-steps back, gives

$$y^*(t \mid t-k) = y(t) - E\xi(t) = Gu(t-k) + Fy(t-k) + Hy^*(t-1 \mid t--k1)$$

2.18

Or

$$y(t) = Gu(t-k) + Fy(t-k) + Hy^*(t-1 \mid t--k1) + \eta(t), \eta(t) = -E\xi(t)$$

2.19

Equation (2.19) thus provides the regression expression for estimating the coefficients of G, F, and H. If estimation and control is carried out every sampling instant, then we have a self-tuning minimum variance control strategy.

The minimum variance controller has several interesting properties. Re-arrangement of Equation (2.15) gives

$$w(t) = EBu(t) + Fy(t) + Hy^*(t+k-1 \mid t-1) = y^*(t+k \mid t)$$

2.20

This is known as the 'control law', and tells us that the control signal calculated from Equation (2.15) will drive the k-step-ahead predictor $y^*(t+k|t)$ to the set-point $w(t)$. Using the definition of Equation (2.9),

$$w(t) = y(t+k) - E\xi(t+k) \Rightarrow y(t+k) = w(t-k) + E\xi(t)$$

2.21

Thus, if the process model is precise, then the controlled output will track the set point after the time delay period. The only error will be that due to a weighted sum of process stochastic disturbance (noise). The controller offers dead-time compensation and the response is as good as possible. Further, if there is no process noise, i.e. $\xi(t)$

=0, then it can be noticed that the minimum variance controller is equivalent to a deadbeat controller.

Equation (2.21) also represents the closed loop relationship, and we can see that there are no poles or zeros. This indicates that the minimum variance controller achieves its performance objective by canceling process dynamics. Therefore, it cannot be applied to non-minimum phased systems. Another limitation is that the minimum variance strategy is often observed to exert excessive control effort, which may not be tolerated from the operational point of view.

2.4 History, derivation and explanation of Harris Performance Index

Both Astrom (1970) and Harris (1989) have proved the use of minimum variance control as a benchmark standard against which control loop performance can be assessed. The most distinguished work is that Harris showed how simple time series analysis technique could be used to find an appropriate expression for the feedback controller-invariant term from routine operating data. This contribution was significant in the sense that a distinct and new direction and framework for the control loop performance monitoring area was revealed.

Harris work has exposed that the best achievable performance, when measured by the mean square error, can be estimated from a time series to closed loop process output data alone. Harris proved that the lower bound of the output variance is the output variance, which is produced by a minimum variance controller.

More recently, another related performance assessment statistics defined as the normalized performance index has been proposed by Desborough and Harris (1992).

Harris normalized performance index (H_{NPI}) is a scalar between zero and one; with zero indicating that, the process is operating under minimum variance control i.e. at its optimal performance bound.

In a general case where the controller is not a minimum variance,

$$\sigma_y^2 = \sigma_{mv}^2 + \sigma_{\hat{y}}^2$$

2.22

where σ_y^2 is the variance of the output

σ_{mv}^2 is the variance of the output under minimum variance condition

$\sigma_{\hat{y}}^2$ is the variance of the forecast output

From Equation (2.21), the variance of the output under minimum variance conditions when the set point deviation is zero (Regulatory and not Tracking problem) is equal to the variance in the process noise ξ(t), and this is the best achievable variance.

The mean square error (M_{se}) of the output can be calculated as follows

$$M_{se}(Y) = \sigma_y^2 + \mu_{\hat{y}}^2$$

2.23

where μ is the mean deviation from the setpoint.

It is assumed that the output is unbiased therefore; the mean square error is equal to variance of the output. Finally,

$$H_{NPI} = 1 - [\sigma^2_{mv} / M_{se}(Y)] = 1 - [\sigma^2_{mv} / \sigma^2_y]$$

2.24

As it was stated before, In a pure regulatory problem the minimum variance is the variance that is produced by the process noise, therefore,

$$H_{NPI} = 1 - Var(N_o) / \sigma^2_y$$

2.25

where N_o is the output noise

The performance index (P_i) that is being used in this research is calculated in a similar way to the Harris Normalized Performance Index. However, the ratio was between the input noise variance and the output variance.

$$P_i = Var(N_i) / \sigma^2_y$$

2.26

The best performance can be achieved under minimum variance conditions by maximizing this performance index. In order to scale this index between zero and one, the following normalized performance (N_{Pi}) index is used for comparisons and benchmarking

$$N_{Pi} = Var(N_i) / (\sigma^2_y + Var(N_i))$$

2.27

This shall have a value of one as a maximum, which is achieved at minimum variance conditions. It shall also have a value of zero when the output variance is too high that the value of stochastic input disturbance variance diminishes beside it.

Chapter 3

Genetic Algorithms

3.1 Introduction

Genetic Algorithms (GAs) are stochastic global optimization techniques that replicate natural evolution on the solution space of the optimization problems [Goldberg (1989)]. Unlike most conventional optimization techniques that sequentially evaluate a single point in the search region, GAs operate on a population of prospective solutions (individuals) in each iteration (generation). By merging some individuals of the present population in accordance with predefined regulations or operations, a new population that have better solutions or fitter individuals, is created as the next generation. Individuals for reproduction are elected based on their objective function values (fitness value) and the Darwinian theory of the survival of the fittest. Supported by the Schema theory, GAs are proven to yield better solutions along the evolving process since good characters in the individuals of a generation are always passed to the next generation.

An imperative stride in a GA is the encoding of variables, x_i in a string structure (known as genetic code) to symbolize a point in the solution space. Three coding schemes, which are named binary, gray, and floating-point, are available. Traditionally, binary coding is used, and it works best when the variables are discrete. Since some engineering problems deal with continuous variables, a floating-point number is more appropriate for representing a continuous variable since it allows representation to the machine internal precision and it requires less memory. Binary coding is used in this research.

The first step after choosing the coding scheme is to generate an initial population of solutions in the feasible region. This is often done in a random fashion. Therefore, GA's do not require initial estimates. Number of solutions or chromosomes in the population (N_{pop}) affects both the reliability and efficiency of the GA. A small N_{pop} is inadequate to investigate the solution space meticulously, whereas a large population would prevent convergence to local minima but causes slow convergence. Therefore, an optimal N_{pop} exists for locating the global solution within a reasonable computational time. It was suggested that N_{pop} equal to (4–40) times the number of binary bits used to represent a chromosome. Selection scheme in the GA refers to how the individuals in a population are elected for reproduction. In order to imitate the survival of the fittest theory, individuals having better fitness values must have a higher chance of being selected. An effective selection scheme should guarantee that a certain minimum number of individuals would be selected for reproduction. Selection schemes based straightforwardly on objective function values as fitness values may lead to premature convergence due to the presence of "super-individuals", which are greatly fitter than others. Two techniques, scaling and ranking, are used by researchers to overcome this problem. Scaling maps objective function values to some positive fitness values whereas in the ranking technique, individuals of the population are ordered in ascending order of objective function values. Linear and exponential ranking are commonly used. Selection of individuals for reproduction is based on scaled values or ranking instead of actual objective function values.

Genetic operators used to create new individuals for the next population from those selected individuals of the current population, serve as searching mechanisms in GA. Crossover and mutation operators are commonly used by programmers and researchers.

Crossover forms two new individuals by first choosing two individuals from the mating pool, and then exchange different parts of genetic information between them. This merging operation is only to take place with a user-defined crossover probability (p_{cros}) so that some parents remain unaffected even if they are chosen for reproduction. Three types of crossover, namely, single point, multipoint, and uniform, are possible for binary coding.

Mutation is a unary operator that creates a new solution by a random change on an individual. It provides a guarantee that the probability of searching any given string will never be zero and acting as a safety net to recover good genetic objects which may be vanished through the action of selection and crossover. Just like crossover, mutation proceeds with a small probability p_{mut}. In binary coding, mutation is the flipping of the binary bits from 0 to 1 or vice versa. Non-binary mutation is usually achieved by either disturbing parameter values or random selection of new values within the feasible range. This is known as uniform mutation.

A GA terminates when a user-specified criterion is satisfied. Usually, it stops after evolving for a specified number of generations (N_{gen}), when the best fitness value has reached an expected target or when the fitness values in the population are close.

3.2 Mapping

For any optimization parameter, a mapping relates the binary-coded deviation ($\Delta x_{i,2}$) and the mean parameter value ($\bar{x}_i$) to the parameter value (x_i). Therefore, a mapping presents a vector (x) equivalent to each binary-coded deviation vector (Δx_2) in its population. Two mapping techniques are normally used to tackle this problem – Upreti (2004).

The first mapping technique is what is called Logarithmic Mapping. The purpose of logarithmic mapping is to improve the relative precision within the elements of x. For any optimization parameter, the logarithmic mapping provides the value, $x_i = b^{z_i}$ where

$$b = x_{i,\max} - x_{i,\min} \tag{3.1}$$

$$z_i = \log_b \bar{x}_i + \frac{\log_b D_i}{2^{N_{bit,i}} - 1} \Delta x_{i,2} \tag{3.2}$$

In Equation (3.1), b is the logarithmic base, and $x_{i,max}$ and $x_{i,min}$ are the maximum and minimum values of the parameter, x_i. In Equation (3.2), D_i is the value of the domain between the limits of $D_{min} > 0$ and b, and $N_{bit,i}$ is the number of representative bits for any i-th element of Δx_2, i. e. $\Delta x_{i,2}$.

The second technique is what is called Linear Mapping. The linear mapping is simpler, and it is specified by the following equation:

$$x_i = \bar{x}_i + \frac{D_i}{2^{N_{bit,i}} - 1} \Delta x_{i,2} \tag{3.3}$$

3.3 GA Optimization Algorithm

The optimization technique is composed of two consecutive steps. The first step is applying genetic algorithms until we reach the global solution zone. Then, the second step utilizes gradient search to locate the exact global solution. It was proposed in Upreti (2004), where the optimization algorithm that is used here engages Newton's search with genetic operations. This shall enable the location of the exact global minimum in a more opportune and efficient manner. Given N_x number of optimization parameters, the presented optimization algorithm randomly initializes a mean value $\bar{x}_i$ for each optimization parameter between the limits $x_{i,min}$ and $x_{i,max}$, $i = 0, 1, \ldots, N_x - 1$. The value of any i-th parameter, x_i, can be calculated from $\bar{x}_i$, and a binary-coded deviation $\Delta x_{i,2}$ based on some mapping. The N_x values of each of x_i, $\bar{x}_i$ and $\Delta x_{i,2}$ form vectors, x, $\bar{x}$ and Δx_2, respectively. In addition to $\bar{x}$, a population of Δx_2 is also randomly generated.

The mappings to calculate x from x, $\bar{x}$ and Δx_2 in its population are described in Section 3.2. In order to produce the optimal vector $\hat{x}$, the genetic operations of selection, crossover, and mutation are consecutively applied to the population of binary-coded deviation vector Δx_2. A value of objective function is associated with each Δx_2 by using the x (as calculated from the mapping) to solve the mathematical model of a problem. These objective function estimations are done before selection. The value of each objective function is scaled up by raising it to a specified power, $n > 1$, to favor the optimally better members of the population during selection. If any constraint is violated for any Δx_2, its objective function is set to zero so that the deviating Δx_2 is eliminated in the following iteration of selection after participating in crossover and mutation.

After a certain number of genetic operations (N_{gen} generations) occur, the optimal vector that was achieved is improved by Newton's search. Next, the domain of each x_i is constricted, and $\bar{x}$ is replaced by $\hat{x}$. This is considered a complete iteration of the algorithm. The logarithmic mapping is alternated in the following iterations with a linear mapping. Each of the domains is reduced until it accomplishes its minimum size, when it is stretched successively. This stretch helps in finding a better optimum in bigger domains. When the maximum size of a domain is reached, its consecutive contraction is resumed to let the improvement of an optimistically new optimal vector.

The programming of the algorithm follows the following steps:

1. Initialize,

   a. $\bar{x}$, the vector of mean values of optimization parameters using,

$$\bar{x}_i = x_{i,\min} + R_i\left(x_{i,\max} - x_{i,\min}\right); \qquad 0 \le R_i \le 1; \quad i = 0,1,\ldots,N_{x-1} \tag{3.4}$$

   where R_i in equation (3.4) is the i-th pseudo-random number obtained from a pseudo-random number generator;

   b. a population of N_{pop} binary-coded deviation vectors Δx_2 using the pseudo-random number generator; and

   c. the parameter domain, $D_i = (x_{i,\max} - x_{i,\min})/2$ for each optimization parameter

2. Set logarithmic mapping for the genetic operations of selection, crossover, and mutation.

3. Generate an optimal vector by repeating the following consecutive operations on the population of Δx_2 for N_{gen} generations:

   a. objective function (J) evaluation for each Δx_2,
   b. selection based on the scaled objective function (J^n),
   c. crossover with probability p_c, and
   d. mutation with probability p_m

4. Improve the optimal vector ($\hat{x}$) obtained this far using Newton's Method (Section 3.4)

5. Store the resulting optimal value of objective function ($\hat{I}_1$), and corresponding optimal vector ($\hat{x}$).

6. Replace $\bar{x}$ by $\hat{x}$.

7. Repeat Steps 3–6 once with linear mapping.

8. For each optimization parameter,

   a. if D_i is equal to either $D_{i,min}$ or $D_{i,max}$, set the size-variation factor for control domain, $C = C^{-1}$. (This step allows the alternation of the successive contraction of D with its successive expansion.)
   b. set $D_i = CD_i$. If $D_i < D_{i,min}$, set $D_i = D_{i,min}$. If $D_i > D_{i,max}$, set $D_i = D_{i,max}$. (This step allows the variation of D_i within its limits.)

9. Go to Step 2 until the iterative change in $\hat{I}_1$ is negligible.

3.4 Gradient Search Algorithm

Newton's method is an efficient algorithm for finding approximations to optimization of a real-valued function.

The scheme of this technique can be described as: an initial value is selected to start with a value which is logically close to the true zero of the first derivative, then substitutes the function by its tangent which can be derived from an explicit function by the basic principles of calculus. Then the zero of the second derivative can be computed (which is simply done with basic algebra). This zero of the second derivative will typically be a better approximation to the first derivative zero, and the method can be iterated.

Suppose f: [a,b] $\rightarrow$ R is a differentiable function defined on the interval [a,b] with values in the real numbers R. We start with an arbitrary value x_0 (the closer to the zero of the first derivative the better) and then define for each natural number n:

$$\bar{x}_{n+1} = \bar{x}_n - \frac{\bar{f}'(\bar{x}_n)}{\bar{\bar{f}}''(\bar{x}_n)}$$

3.5

where, f' denotes the first derivative and f'' denotes the second derivative of the function f respectively. .

We can prove that, if f' is continuous, and if the unknown zero x is isolated, then there exists a neighborhood of x such that for all starting values x_0 in that neighborhood, the sequence $\{x_n\}$ will converge towards x. Furthermore, if $f'(x) \neq 0$, then the convergence is quadratic, which intuitively means that the number of correct digits roughly doubles in every step.

In general, the convergence is quadratic: the error is squared at each step (that is, the number of exact digits doubles in each step). There are some cautions to be considered when programming this method. Initially, Newton's method requires that the derivative be calculated directly. If instead the derivative is approximated by the slope of the line through two points on the function's graph, the secant method results — although based on how computational effort is measured, the secant method may be more efficient. In addition, if the initial value is too far removed from the true zero, Newton's method can fail to converge at all to a global minimum or maximum. Because of this, all practical implementations of Newton's method put an upper limit on the number of iterations and perhaps on each iteration size. Moreover, if the root being sought has multiplicity greater than one, the convergence rate is reduced to linear (errors reduced by a constant factor at each step) unless special steps are taken. Finally, a penalty function should be used to avoid convergence to zero, which is another disadvantage of Newton's method convergence.

To carry out Newton's Method along the steepest descent in Step 4 of the algorithm, the interior penalty function - Upreti (2004) - method was used. It incorporates the inequality constraints, into the augmented objective function given by

$$I_2 = \lim_{r \to o} \left\{ I_1 - \sum_{i=1}^{N} \frac{1}{r} \left[\frac{1}{g_i} \right] \right\}; \; g_i < 0 \quad \forall i \tag{3.6}$$

where, g_i is inequality constraint and r is the interior penalty function coefficient.

The programming of the algorithm follows the following steps:

1. Obtain the vector, $\hat{x}$, and corresponding $\hat{I}_1$ generated so far using genetic operators. Set the counter, i = 0; and the penalty, r = 1.

2. Set the Newton's search counter, j = 0. Set $x^{(j)} = \hat{x}$, and calculate $I_2^{(j)}$

3. Calculate the vector of the partial derivatives of $I_2^{(j)}$, i.e. $x'^{(j)}$ using the derivatives' equations in section 4.2. If ($\|x'\| = 0$) then set $x^{(j+1)} = x^{(j)}$, and go to Step 8.

4. Calculate $x^{(j+1)}$ along the steepest descent direction as follows:

$$\bar{x}^{(j+1)} = \bar{x}^{(j)} - \frac{\left\|\bar{x}'^{(j)}\right\|}{\left\|\bar{\bar{x}}''^{(j)}\right\|} \tag{3.7}$$

Calculate the corresponding $I_2^{(j+1)}$

5. If $I_2^{(j+1)} > I_2^{(j)}$ then set $x^{(i+1)} = x^{(j)}$, and go to Step 8.

6. If ($|1 - I_2^{(j)} / I_2^{(j+1)}| < \varepsilon$) then set $x^{(i+1)} = x^{(j+1)}$ go to Step 8.

7. Set j = j + 1 and go to Step 3.

8. Calculate $I_1^{(i+1)}$ corresponding to $x^{(i+1)}$. If ($I_1^{(i+1)} > I_1^{(i)}$ then set $\hat{I}_1 = I_1^{(i)}$ $\hat{x}$ = $x^{(i)}$, and exit.

9. Set $\hat{I}_1 = I_1^{(i+1)}$ and $\hat{x} = x^{(i+1)}$. If $(|1 - I_2^{(j)}/I_2^{(j+1)}| < \varepsilon)$ or $(r < \varepsilon)$ then exit;

10. Reduce the penalty to r = Cr, set i = i + 1, and go to Step 2.

3.5 Model Inputs

1. The mathematical model and its parameters for the calculation of objective function;

2. The number of optimization parameters (N_x), and constraints;

3. The minimum value ($D_{i,min}$) of control domain, its maximum value ($D_{i,max}$, i = 0, 1, . . .N_{x-1}), and a factor (C) to vary the size of control domain;

4. A seed number to generate pseudo-random numbers;

5. The following parameters for the genetic operations of selection, crossover, and mutation:

   a. the number of bits ($N_{bit,i}$) for each optimization parameter
   b. the number of cross-over sites (N_{xsites}) for each $\Delta x_{i,2}$
   c. the probability of cross-over (p_c)
   d. the probability of mutation (p_m)
   e. the power index (n) to scale objective function
   f. the number of genetic generations (N_{gen}) every iteration

Part III

Mathematical Modeling

Chapter 4

Mathematical Modeling

4.1 Normalized Performance Index Calculations

The calculations of the process model are performed in a discrete state space model. This was chosen because state space model can be further used for MIMO systems easily. In addition, digital control is more practicable and widely used in industrial control systems. This choice also allows us to interpret the time delay and the sampling time to the dimension of the system.

The following system is under study

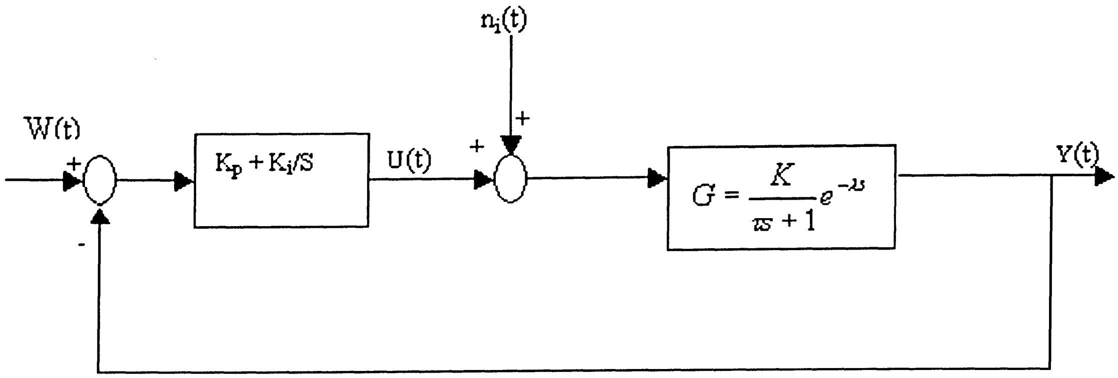

Figure 4.1 Closed Loop Block Diagram

where

W(t) is the change in set point (equals zero for regulatory problem).

U(t) is the control effort.

K is the process gain.

τ is the process time constant.

λ is the system time delay.

n_i is the process input disturbance.

n_o is the output disturbance.

K_p is the controller proportional gain.

K_i is the controller integral gain.

The following equations describe the continuous open loop state space model for the above-mentioned system

$$x^{\cdot}(t) = Ax(t) + B[(U(t-\lambda) + n_i(t)] \quad 4.1$$

$$Y(t) = Cx(t) + n_o(t) \quad 4.2$$

If the sampling Time is equal to T, therefore the system can be written in the discrete form as follows

$$x[(k+1)T] = e^{AT} x(kT) + \int_{kT}^{(k+1)T} e^{A(kT+T-L)} B\big[(U(L-\lambda) + n_i(kT))\big]dL$$

4.3

Let us introduce a new variable D where τ=kT+T-D and substitute τ in terms of D, therefore:

$$x_{k+1} = e^{AT} x_k + \int_0^T e^{AD} B\big[(U(kT+T-\lambda-D) + n_{ik})\big]dD$$

4.4

The time delay can be factorized into two parts as follows

$$\lambda = (l+1)T - m$$

4.5

where $l+1$ is the multiple integers part of the time delay plus one
It is always more than or equal to zeros and m is the fractional extra part of the sampling time. Equation 4.4 can be expressed after eliminating λ and substitute it in terms of l and m as follows:

$$x_{k+1} = e^{AT} x_k + \int_0^m e^{AD} B(U(kT-(l+1)T+T) + n_{ik})dD + \int_m^T e^{AD} B(U(kT-(l+1)T)dD$$

4.6

This can be simplified to be

$$x_{k+1} = F_1 x_k + G_1(U_{k+1-l} + n_{ik}) + G_2 U_{k-l}$$

4.7

where

$$F_1 = e^{-T/\tau}$$
$$G_1 = K(1 - e^{-m/\tau})$$
$$G_2 = K(e^{-m/\tau} - e^{-T/\tau})$$

The system can be re-written in a matrix form noticing that it shall be a variable dimensions matrix system. The size shall depend on the time delay and the sampling time. It can be written as follows:

$$\begin{bmatrix} x_{k+1} \\ U_{k-5} \\ U_{k-4} \\ \downarrow \\ U_k + n_i \end{bmatrix} = \begin{bmatrix} F_1 & G_1 & G_2 & 0 & \rightarrow \\ 0 & 0 & I & 0 & \rightarrow \\ \downarrow & \downarrow & 0 & I & 0 \\ \downarrow & \downarrow & \downarrow & 0 & I \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ U_{k+1-l} \\ U_{k-l} \\ \downarrow \\ U_{k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ \downarrow \\ \downarrow \\ \downarrow \\ 1 \end{bmatrix} (U_k + n_{i_k})$$

4.8

The system can be further simplified to be in the following form

$$X_{k+1} = FX_k + GU_{2,k}$$

4.9

$$Y_k = CX_k + n_{ok}$$

4.10

where

$$X_{k+1} = \begin{bmatrix} x_{k+1} \\ U_{k-5} \\ U_{k-4} \\ \downarrow \\ U_k + n_i \end{bmatrix} \qquad X_k = \begin{bmatrix} x_k \\ U_{k+1-l} \\ U_{k-l} \\ \downarrow \\ U_{k-1} \end{bmatrix} \qquad G = \begin{bmatrix} 0 \\ \downarrow \\ \downarrow \\ \downarrow \\ 1 \end{bmatrix}$$

$$F = \begin{bmatrix} F_1 & G_1 & G_2 & 0 & \rightarrow \\ 0 & 0 & I & 0 & \rightarrow \\ \downarrow & \downarrow & 0 & I & 0 \\ \downarrow & \downarrow & \downarrow & 0 & I \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad \& \qquad U_{2,k} = (U_k + n_{i_k})$$

The closed loop system can be obtained by substituting for the control efforts in above-mentioned equation from the PI controller equation as follows:

$$U_{2,k} = -K_P Y_k - K_I \int Y_k dT \tag{4.11}$$

The integral mode can be introduced as $\alpha_k = \int Y_k dT$ and therefore, $Y_k = \frac{\partial \alpha_k}{\partial T}$, $Y_k = \frac{\alpha_{k+1} - \alpha_k}{T}$ and finally

$$\alpha_{k+1} = TY_k + \alpha_k \tag{4.12}$$

Substitute from 4.10 into 4.12 the following equation is obtained

$$\alpha_{k+1} = TCX_k + \alpha_k + Tn_{o_k}$$

4.13

In addition, with substitution from 4.11 into 4.9 and substitute for the integral mode from 4.13, the following equation is obtained

$$X_{k+1} = FX_k - GK_P Y_k - GK_I \alpha_k$$

4.14

Finally substitute for Y_k from 4.10 and with some equation re-arrangement, the following equation is obtained

$$X_{k+1} = (F - GK_P C)X_k - GK_I \alpha_k - GK_P n_{ok}$$

4.15

The closed loop state space model can be written in the following form

$$X_{c_{k+1}} = F_c\, X_{c_k} + G_c n_{o_k}$$

4.16

$$Y_k = C_c X_{c_k}$$

4.17

where:

$$X_{c_{k+1}} = \begin{bmatrix} X_{k+1} \\ \alpha_{k+1} \end{bmatrix}, \qquad X_{c_k} = \begin{bmatrix} X_k \\ \alpha_k \end{bmatrix}, \qquad F_c = \begin{bmatrix} F - GK_P C & -GK_I \\ TC & 1 \end{bmatrix}$$

$$G_c = \begin{bmatrix} -GK_P \\ T \end{bmatrix} \qquad \& \qquad C_c = [C \quad 0]$$

It is clear that to calculate the performance index, output variance should be calculated first. The covariance matrix for the output was calculated by solving the following discrete Lyapunov equation:

$$\Sigma_{k+1} = F_c \Sigma_k F_c^{\ T} + G_c Q G_c^{\ T}$$

4.18

where Σ is the output covariance matrix and Q is the noise covariance matrix, which is

$$Q = \begin{bmatrix} Variance(n_i) & 0 \\ 0 & Variance(n_o) \end{bmatrix}$$

4.19

The variance of the output for performance index evaluation is calculated as follows:

$$\sigma_y = C_c \Sigma_k C_c^T$$

4.20

The performance index that is being used in this research is calculated in a similar way to the Harris Normalized Index (HPI). However, the ratio was between the input stochastic noise variance and the output variance as follows

$$P_i = \mathrm{Var}\ (N_i) \,/\, \sigma_y^2$$

4.21

The best performance can be achieved under minimum variance conditions by maximizing this performance index. In order to scale this index between zero and one, the following normalized index is used for comparison and benchmarking

$$N_{Pi} = \mathrm{Var}\ (N_i)\ /(\sigma_y^2 + \mathrm{Var}\ (N_i)) \qquad 4.22$$

This index has a value of one as a maximum, which is achieved at minimum variance conditions. It has also a value of zero when the output variance is too high that the value of noise variance diminishes beside it.

An enumerative search to maximize the performance index was initially used and then the results were scaled to the Normalized Performance Index for comparison. The technique is based on changing the values of the tuning parameters, which are the optimization variables (K_p and K_i), and store them only if the following constraints are not violated:

- The closed loop should be stable
- The closed loop should be controllable and observable
- The covariance matrix should be positive definite
- The covariance matrix should be symmetric

It is very clear that the above-mentioned constraints are non-linear and implicit. In addition, the objective function itself is implicit as well. Therefore, the programming is very challenging and the optimization process is quite complicated.

Another important factor is the time that is used to carry out this enumerative search. The matrix sizes are based on the time delay and the sampling time. With long time delay processes, which are good examples where minimum variance control will be useful, these matrix sizes shall be too large and the required time to solve this problem is too long. This is one of the main reasons to find another optimization search technique that is more timely efficient and can handle this degree of non-linearity. Genetic Algorithm is a perfect candidate.

4.2 Normalized Performance Index calculations using Hybrid Genetic Algorithms

The optimization problem shall be handled in a different way. An explicit format for the output variance is developed in order to carry out the optimization process.

The open loop can be expressed in a discrete format as follows:

$$G(z) = \frac{z^{-(l+1)}(G_1 z + G_2)}{z - F} \tag{4.23}$$

where

$$\begin{aligned} F_1 &= e^{-T/\tau} \\ G_1 &= K(1 - e^{-m/\tau}) \\ G_2 &= K(e^{-m/\tau} - e^{-T/\tau}) \end{aligned}$$

In addition, the controller discrete transfer function can be expressed as

$$C(z) = K_P + \frac{K_i T}{z - 1} \tag{4.24}$$

Equation 4.24 can be rearranged to be written as follows;

$$C(z) = \frac{K_P z + (-K_p + K_i T)}{z - 1}$$ 4.25

The closed loop discrete transfer function (T) can be written as:

$$T = \frac{Y(z)}{N_i(z)} = \frac{G}{1 + GC}$$ 4.26

By substituting equations 4.23 and 4.25 into 4.26, the following equation is obtained:

$$T = \frac{Y(z)}{N_i(z)} = \frac{\left[\dfrac{z^{-(l+1)}(G_1 z + G_2)}{z - F}\right]}{1 + \left[\dfrac{z^{-(l+1)}(G_1 z + G_2)}{z - F}\right]\left[\dfrac{K_P z + (-K_p + K_i T)}{z - 1}\right]}$$ 4.27

Then by multiplying equation 4.27 by $\frac{z-F}{z-F}$ and rearranging the equation;

$$T = \frac{Y(z)}{N_i(z)} = \frac{z^{-(l+1)}(G_1 z + G_2)}{z - F + \left[z^{-(l+1)}(G_1 z + G_2)\right]\left[\dfrac{K_P z + (-K_p + K_i T)}{z - 1}\right]}$$ 4.28

Then by multiplying equation 4.28 by $\frac{z-1}{z-1}$ and rearranging the equation;

$$T = \frac{Y(z)}{N_i(z)} = \frac{z^{-(l+1)}(G_1 z + G_2)(z - 1)}{(z - F)(z - 1) + z^{-(l+1)}(G_1 z + G_2)(K_P z + (-K_p + K_i T))}$$ 4.29

$$T = \frac{Y(z)}{N_i(z)} = \frac{(z^{-l} - z^{-(l+1)})(G_1 z + G_2)}{z^2 - (F+1)z + F + z^{-(l+1)}(G_1 z + G_2)(K_p z + (-K_p + K_i T))}$$

4.30

Then by multiplying equation 4.30 by $\frac{z^{-2}}{z^{-2}}$ and rearranging the equation;

$$T = \frac{Y(z)}{N_i(z)} =$$

$$\frac{(z^{-(l+2)} - z^{-(l+3)})(G_1 z + G_2)}{1 - (F+1)z^{-1} + Fz^{-2} + z^{-(l+3)}\left(G_1 K_p z^2 + \left(G_1(-K_p + K_i T) + G_2 K_p\right)z + G_2(-K_p + K_i T)\right)}$$

4.31

This can be rearranged as follows;

$$T = \frac{Y(z)}{N_i(z)} =$$

$$\frac{G_1 z^{-(l+1)} + (G_2 - G_1)z^{-(l+2)} - G_2 z^{-(l+3)}}{1 - (F+1)z^{-1} + Fz^{-2} + G_1 K_p z^{-(l+1)} + \left(G_1(-K_p + K_i T) + G_2 K_p\right)z^{-(l+2)} + \left(G_2(-K_p + K_i T)\right)z^{-(l+3)}}$$

4.32

By carrying out the inversion of (z) transform on the above equation, we can obtain Y(k) in an explicit format;

$$Y(k) = f_1 Y_{k-1} + f_2 Y_{k-2} + f_3 Y_{k-(l+1)} + f_4 Y_{k-(l+2)} + f_5 Y_{k-(l+3)} + g_1 N_{ik-(l+1)} + g_2 N_{ik-(l+2)} + g_3 N_{ik-(l+3)}$$

4.33

where

$$
\begin{aligned}
f_1 &= F + 1 \\
f_2 &= -F \\
f_3 &= -G_1 K_p \\
f_4 &= -\left(G_1\left(-K_p + K_i T\right) + G_2 K_p\right) \\
f_5 &= -G_2\left(-K_p + K_i T\right) \\
g_1 &= G_1 \\
g_2 &= G_2 - G_1 \\
g_3 &= -G_2
\end{aligned}
$$

The output variance can be calculated assuming a simple regulatory problem where the change in set point is equal to zero.

$$J_\infty = \lim_{N \to \infty} E\left\{\frac{1}{N}\sum_{k=1}^{N} y^2(k)\right\}$$

4.34

The performance index can be written as

$$I_1 = \frac{\sum_{i=0}^{i=N_s-1}\left(N_{i_i}\right)^2}{\sum_{i=0}^{i=N_s-1}\left(Y_i\right)^2} = \frac{V_{N_i}}{V_Y}$$

4.35a

The only constraint in this optimization problem is locating the closed loop poles inside the unit circle of the stability diagram. In other words the roots of the Diophantine equation, which is the denominator of equation 4.32, should be less than one.

This could be a problem to solve especially with high order equation. Alternative approach was suggested that has shown great potential.

This different approach clamps the process output at a certain high enough value that would be only violated if the closed loop poles are positive. In other words, the output values shall be higher than these clamps only if the loop is unstable. This value was selected to be 10 times higher than the process gain knowing that the input noise ranges only between 1 and -1. The output variations within these limits are acceptable in normal industrial practice.

As the constraint is handled differently, the constrained optimization can be initially handled as non-constrained optimization problem in the programming. Then the augmented objective function I_2 is formed after adding the following constraints;

$$-10 < Y_i < 10 \tag{4.35b}$$

$$0.001 < K_p < 10 \tag{4.35c}$$

$$0.001 < K_i < 10 \tag{4.35d}$$

In order to carry out the search using Newton's method, the first derivative of the objective function should be obtained with respect to each optimization variable.

- Objective Function First Derivative with respect to K_p

$$\frac{\partial Y_i}{\partial K_p} = -G_1 Y_{k-(l+1)} + (G_1 - G_2)Y_{k-(l+2)} + G_2 Y_{k-(l+3)}$$
$$+ f_1 \frac{\partial Y_{k-1}}{\partial K_p} + f_2 \frac{\partial Y_{k-2}}{\partial K_p} + f_3 \frac{\partial Y_{k-(l+1)}}{\partial K_p} + f_4 \frac{\partial Y_{k-(l+2)}}{\partial K_p} + f_5 \frac{\partial Y_{k-(l+3)}}{\partial K_p} \tag{4.36}$$

And

$$\frac{\partial I_1}{\partial K_p} = -2\frac{\sum_{i=0}^{i=N_s-1}\left(N_{i_i}\right)^2}{\sum_{i=0}^{i=N_s-1}\left(Y_i\right)^3}\frac{\partial Y_i}{\partial K_p}$$

4.37

- Objective Function First Derivative with respect to K_i

$$\frac{\partial Y_i}{\partial K_i} = -G_1 T Y_{k-(l+2)} - G_2 T Y_{k-(l+3)}$$
$$+ f_1 \frac{\partial Y_{k-1}}{\partial K_i} + f_2 \frac{\partial Y_{k-2}}{\partial K_i} + f_3 \frac{\partial Y_{k-(l+1)}}{\partial K_i} + f_4 \frac{\partial Y_{k-(l+2)}}{\partial K_i} + f_5 \frac{\partial Y_{k-(l+3)}}{\partial K_i}$$

4.38

And

$$\frac{\partial I_1}{\partial K_i} = -2\frac{\sum_{i=0}^{i=N_s-1}\left(N_{i_i}\right)^2}{\sum_{i=0}^{i=N_s-1}\left(Y_i\right)^3}\frac{\partial Y_i}{\partial K_i}$$

4.39

In addition, the second derivative, which is the Jacobian Matrix, is given by

$$\bar{\bar{J}}_{ac} = \begin{bmatrix} \dfrac{\partial^2 I_2}{\partial K_p^2} & \dfrac{\partial^2 I_2}{\partial K_p \partial K_i} \\ \dfrac{\partial^2 I_2}{\partial K_i \partial K_p} & \dfrac{\partial^2 I_2}{\partial K_i^2} \end{bmatrix}$$

4.40

It can be easily shown that $\dfrac{\partial^2 I_2}{\partial K_i \partial K_p} = \dfrac{\partial^2 I_2}{\partial K_p \partial K_i}$ because the function is invariant.

- Objective Function Second Derivative with respect to K_p

$$\frac{\partial^2 Y_i}{\partial K_p^2} = -2G_1 \frac{\partial Y_{k-(l+1)}}{\partial K_p} + 2(G_1 - G_2)\frac{\partial Y_{k-(l+2)}}{\partial K_p} + 2G_2 \frac{\partial Y_{k-(l+3)}}{\partial K_p}$$
$$+ f_1 \frac{\partial^2 Y_{k-1}}{\partial K_p^2} + f_2 \frac{\partial^2 Y_{k-2}}{\partial K_p^2} + f_3 \frac{\partial^2 Y_{k-(l+1)}}{\partial K_p^2} + f_4 \frac{\partial^2 Y_{k-(l+2)}}{\partial K_p^2} + f_5 \frac{\partial^2 Y_{k-(l+3)}}{\partial K_p^2}$$

4.41

And

$$\frac{\partial^2 I_1}{\partial K_p^2} = 6\frac{\sum_{i=0}^{i=N_s-1} (N_{i_i})^2}{\sum_{i=0}^{i=N_s-1} (Y_i)^4} \left(\frac{\partial Y_i}{\partial K_p}\right)^2 - 2\frac{\sum_{i=0}^{i=N_s-1} (N_{i_i})^2}{\sum_{i=0}^{i=N_s-1} (Y_i)^3} \frac{\partial^2 Y_i}{\partial K_p^2}$$

4.42

- Objective Function Second Derivative with respect to K_i

$$\frac{\partial^2 Y_i}{\partial K_i^2} = -2G_1 T \frac{\partial Y_{k-(l+2)}}{\partial K_i} - 2G_2 T \frac{\partial Y_{k-(l+3)}}{\partial K_i}$$
$$+ f_1 \frac{\partial^2 Y_{k-1}}{\partial K_i^2} + f_2 \frac{\partial^2 Y_{k-2}}{\partial K_i^2} + f_3 \frac{\partial^2 Y_{k-(l+1)}}{\partial K_i^2} + f_4 \frac{\partial^2 Y_{k-(l+2)}}{\partial K_i^2} + f_5 \frac{\partial^2 Y_{k-(l+3)}}{\partial K_i^2}$$

4.43

and

$$\frac{\partial^2 I_1}{\partial K_i^2} = 6\frac{\sum_{i=0}^{i=N_s-1}\left(N_{i_i}\right)^2}{\sum_{i=0}^{i=N_s-1}(Y_i)^4}\left(\frac{\partial Y_i}{\partial K_i}\right)^2 - 2\frac{\sum_{i=0}^{i=N_s-1}\left(N_{i_i}\right)^2}{\sum_{i=0}^{i=N_s-1}(Y_i)^3}\frac{\partial^2 Y_i}{\partial K_i^2}$$

4.44

- Objective Function Second Derivative with respect to K_p then K_i

$$\frac{\partial^2 Y_i}{\partial K_i \partial K_p} = -G_1 T\frac{\partial Y_{k-(l+2)}}{\partial K_p} - G_2 T\frac{\partial Y_{k-(l+3)}}{\partial K_p} - G_1\frac{\partial Y_{k-(l+1)}}{\partial K_i} + (G_1 - G_2)\frac{\partial Y_{k-(l+2)}}{\partial K_i} + G_2\frac{\partial Y_{k-(l+3)}}{\partial K_i} +$$

$$f_1\frac{\partial^2 Y_{k-1}}{\partial K_i \partial K_p} + f_2\frac{\partial^2 Y_{k-2}}{\partial K_i \partial K_p} + f_3\frac{\partial^2 Y_{k-(l+1)}}{\partial K_i \partial K_p} + f_4\frac{\partial^2 Y_{k-(l+2)}}{\partial K_i \partial K_p} + f_5\frac{\partial^2 Y_{k-(l+3)}}{\partial K_i \partial K_p}$$

4.45

And

$$\frac{\partial^2 I_1}{\partial K_i \partial K_p} = 6\frac{\sum_{i=0}^{i=N_s-1}\left(N_{i_i}\right)^2}{\sum_{i=0}^{i=N_s-1}(Y_i)^4}\frac{\partial Y_i}{\partial K_i}\frac{\partial Y_i}{\partial K_p} - 2\frac{\sum_{i=0}^{i=N_s-1}\left(N_{i_i}\right)^2}{\sum_{i=0}^{i=N_s-1}(Y_i)^3}\frac{\partial^2 Y_i}{\partial K_i \partial Kp}$$

4.46

After calculating the non-constrained function (I_1) and derivatives, the following inequality constraints are added to maintain Y, K_p and K_i within the pre-specified limits,

$$I_2 = I_1 -$$

$$\lim_{r\to 0}\left\{ r\left[\sum_{i=1}^{N}\left(\frac{1}{Y_{\min} - Y_i} + \frac{1}{Y_i - Y_{\max}}\right) - \frac{1}{K_p - K_{p_{\min}}} - \frac{1}{K_i - K_{i_{\min}}} + \frac{1}{K_p - K_{p_{\max}}} + \frac{1}{K_i - K_{i_{\max}}}\right]\right\}$$

4.47

where, Y_{min}, Y_{max}, K_{pmin}, K_{pmax}, K_{imin}, K_{imax} are the limits for Y, K_p and K_i respectively and r is the interior penalty coefficient that will be minimized.

Then the derivatives of the augmented objective function are expressed as follows,

$$\frac{\partial I_2}{\partial K_p} = \frac{\partial I_1}{\partial K_p} - \lim_{r \to 0} r \left[\sum_{i=1}^{N} \left(\frac{1}{(Y_{\min} - Y_i)^2} + \frac{-1}{(Y_i - Y_{\max})^2} \right) \cdot \frac{\partial Y_i}{\partial K_p} + \frac{1}{\left(K_p - K_{p_{\min}}\right)^2} - \frac{1}{\left(K_p - K_{p_{\max}}\right)^2} \right]$$

4.48

$$\frac{\partial^2 I_2}{\partial K_p^2} = \frac{\partial^2 I_1}{\partial K_p^2} -$$

$$\lim_{r \to 0} r \left[\begin{array}{l} \sum_{i=1}^{N} \left(\frac{2}{(Y_{\min} - Y_i)^3} + \frac{2}{(Y_i - Y_{\max})^3} \right) \left(\frac{\partial Y_i}{\partial K_p} \right)^2 + \sum_{i=1}^{N} \left(\frac{1}{(Y_{\min} - Y_i)^2} + \frac{-1}{(Y_i - Y_{\max})^2} \right) \cdot \frac{\partial^2 Y_i}{\partial K_p^2} \\ - \frac{2}{\left(K_p - K_{p_{\min}}\right)^3} + \frac{2}{\left(K_p - K_{p_{\max}}\right)^2} \end{array} \right]$$

4.49

Similarly;

$$\frac{\partial I_2}{\partial K_i} = \frac{\partial I_1}{\partial K_i} -$$

$$\lim_{r \to 0} r \left[\sum_{i=1}^{N} \left(\frac{1}{(Y_{\min} - Y_i)^2} + \frac{-1}{(Y_i - Y_{\max})^2} \right) \cdot \frac{\partial Y_i}{\partial K_i} + \frac{1}{\left(K_i - K_{i_{\min}}\right)^2} - \frac{1}{\left(K_i - K_{i_{\max}}\right)^2} \right]$$

4.50

$$\frac{\partial^2 I_2}{\partial K_i^2}=\frac{\partial^2 I_1}{\partial K_i^2}-r\left[\begin{array}{l}\sum_{i=1}^{N}\left(\frac{2}{(Y_{\min}-Y_i)^3}+\frac{2}{(Y_i-Y_{\max})^3}\right)\cdot\left(\frac{\partial Y_i}{\partial K_i}\right)^2+\\ \sum_{i=1}^{N}\left(\frac{1}{(Y_{\min}-Y_i)^2}+\frac{-1}{(Y_i-Y_{\max})^2}\right)\cdot\frac{\partial^2 Y_i}{\partial K_i^2}-\frac{2}{\left(K_i-K_{i_{\min}}\right)^3}+\frac{2}{\left(K_i-K_{i_{\max}}\right)^2}\end{array}\right]$$

4.51

Finally;

$$\frac{\partial^2 I_2}{\partial K_i \partial K_p}=\frac{\partial^2 I_1}{\partial K_i \partial K_p}-$$

$$\lim_{r\to 0} r\left[\sum_{i=1}^{N}\begin{pmatrix}\left(\frac{2}{(Y_{\min}-Y_i)^3}+\frac{2}{(Y_i-Y_{\max})^3}\right)\cdot\left(\frac{\partial Y_i}{\partial K_i}\right)\cdot\left(\frac{\partial Y_i}{\partial K_p}\right)+\\ \left(\frac{1}{(Y_{\min}-Y_i)^2}+\frac{-1}{(Y_i-Y_{\max})^2}\right)\cdot\frac{\partial^2 Y_i}{\partial K_i \partial K_p}\end{pmatrix}\right]$$

4.52

The objective function and its derivatives are programmed into the Hybrid Genetic Algorithms code. The value of the function and the derivatives were checked numerically and were confirmed to be correct. Finally, the results are collected and explained in the next two chapters.

Part IV

RESULTS AND DISCUSSION

Chapter 5

Effect of Process Parameters on the Performance Index

5.1 Introduction

We begin by examining the effect of change in process parameters on the normalized performance index. The following system was taken as an example for this purpose:

$$G = \frac{5}{10s+1}e^{-5s} \tag{5.1}$$

It is a first order plus time delay system, which is commonly used in research. In addition, it is the most widely used in industry. It has a process gain of five, a time constant of ten seconds, and a time delay of five seconds. This system shall be referred as the standard system in the following text.

The system was simulated than optimized using enumerative approach. The following parameters were achieved. The best normalized performance index was 0.78 and it was achieved at a controller proportional gain value of 0.4 and an integral value of 0.01. A sampling time of 1 second was used.

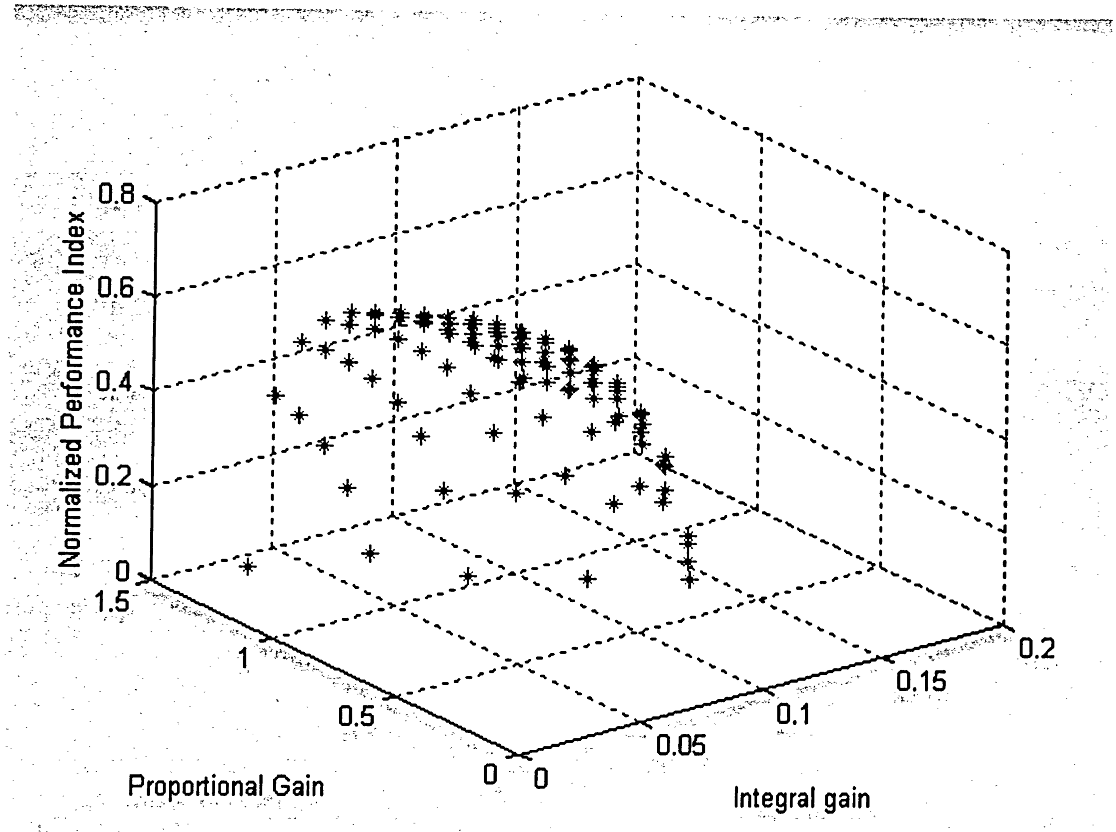

Figure 5.1 Normalized Performance Index 3-D Graph for the Standard System using Enumerative Search

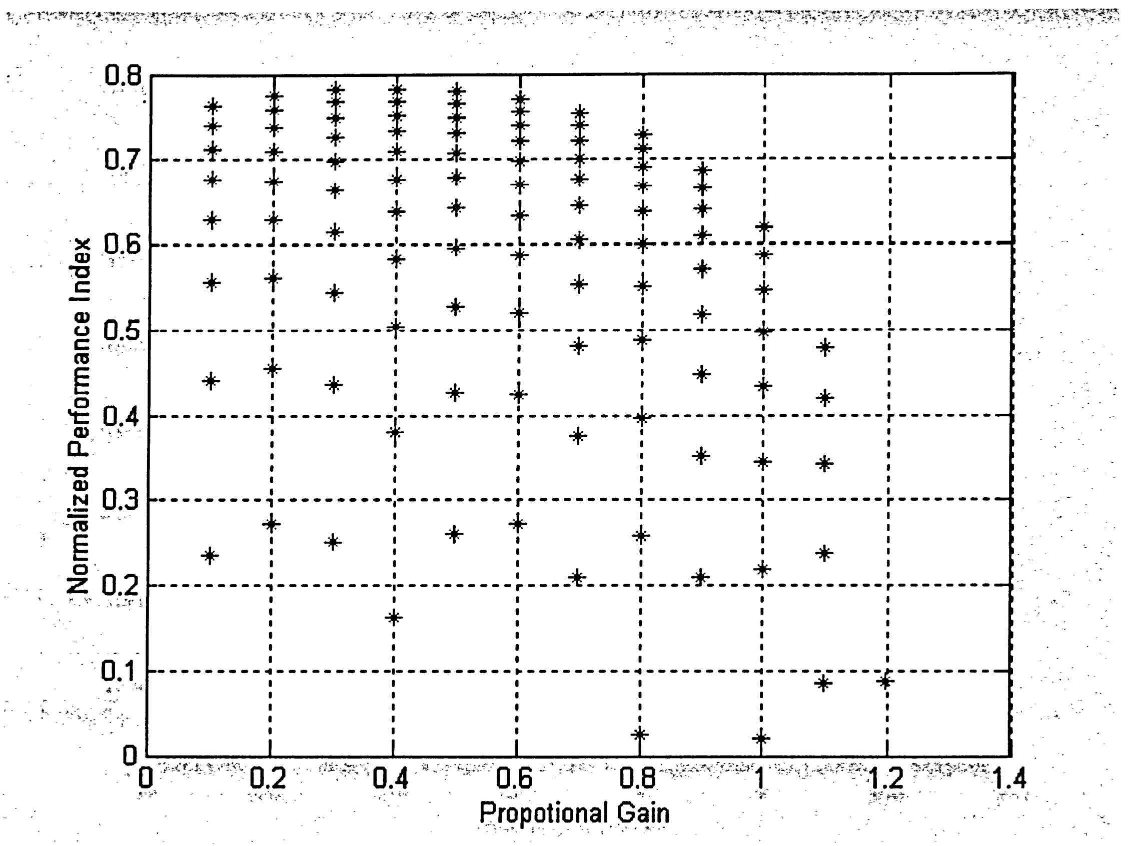

Figure 5.2 Effect of Change in Proportional Gain on the Normalized Performance Index for the Standard System using Enumerative Search

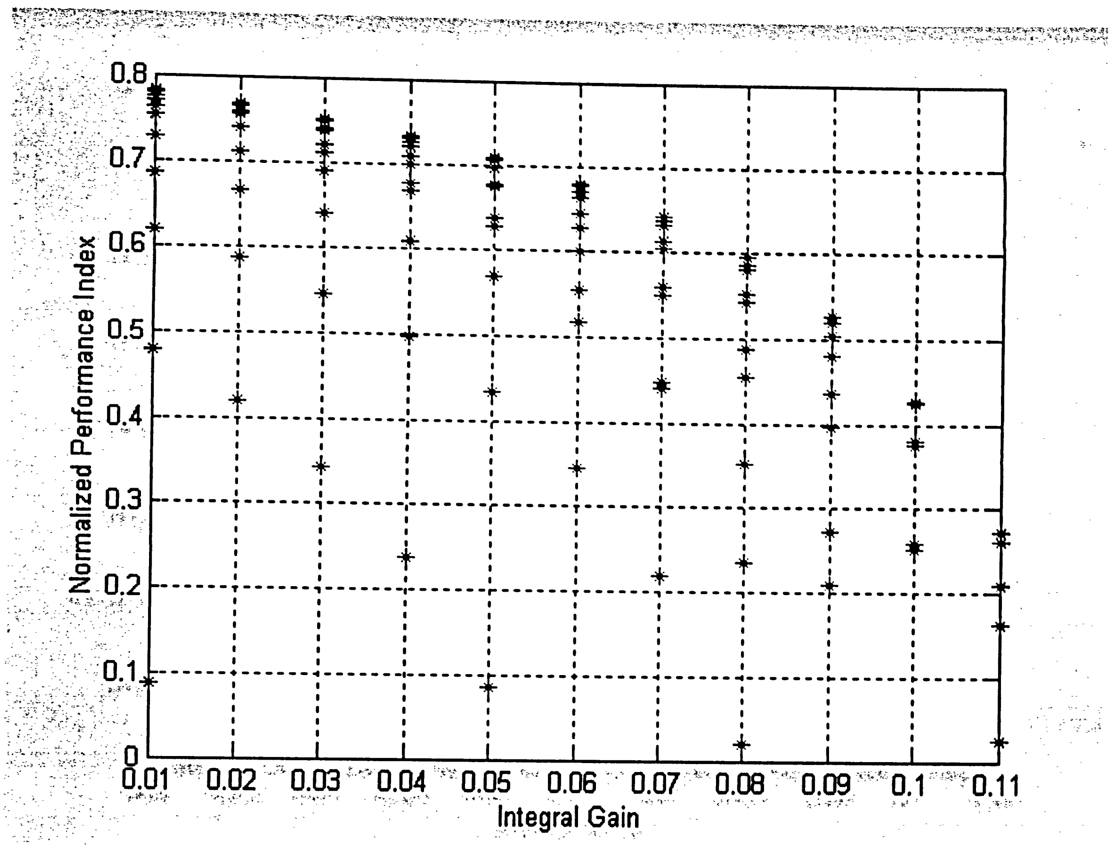

Figure 5.3 Effect of Change in Integral Gain on the Normalized Performance Index for the Standard System using Enumerative Search

It is obvious from Figure 5.1 and Figures 5.3 that the best performance index has a tendency to be achieved at a lower integral gain value. The reasoning behind this phenomenon is that a lower integral gain means higher integral time. When a control loop has a low integral time, it moves quickly to mitigate the existing error. This shall increase the variance of the output. Slower loops are better for minimum variance control as the high integral time acts as a first order low-pass digital filter for the stochastic disturbance. In Figure 5.2, it is clear that the maximum performance index does not exist at any of the proportional gain constraint limits.

One of the main objectives of this research is to determine the sensitivity of the solution. The process parameters are not always constant. For example, a heat exchanger gain and time constant shall be changed if any fouling occurs. In addition, a change in a flow rate shall change the time delay. Therefore, a sensitivity analysis is carried out in order to determine the stability of the minimum variance solution. Each control loop shall have an upper bound at which a higher variance shall not be tolerated.

5.2 Effect of Change in Time Constant

The process gain and the time delay were kept constant and the time constant is changed among a range of its possible values for a stable closed loop system until we reached a value where the performance index does not change significantly. The result is shown in Figure 5.4.

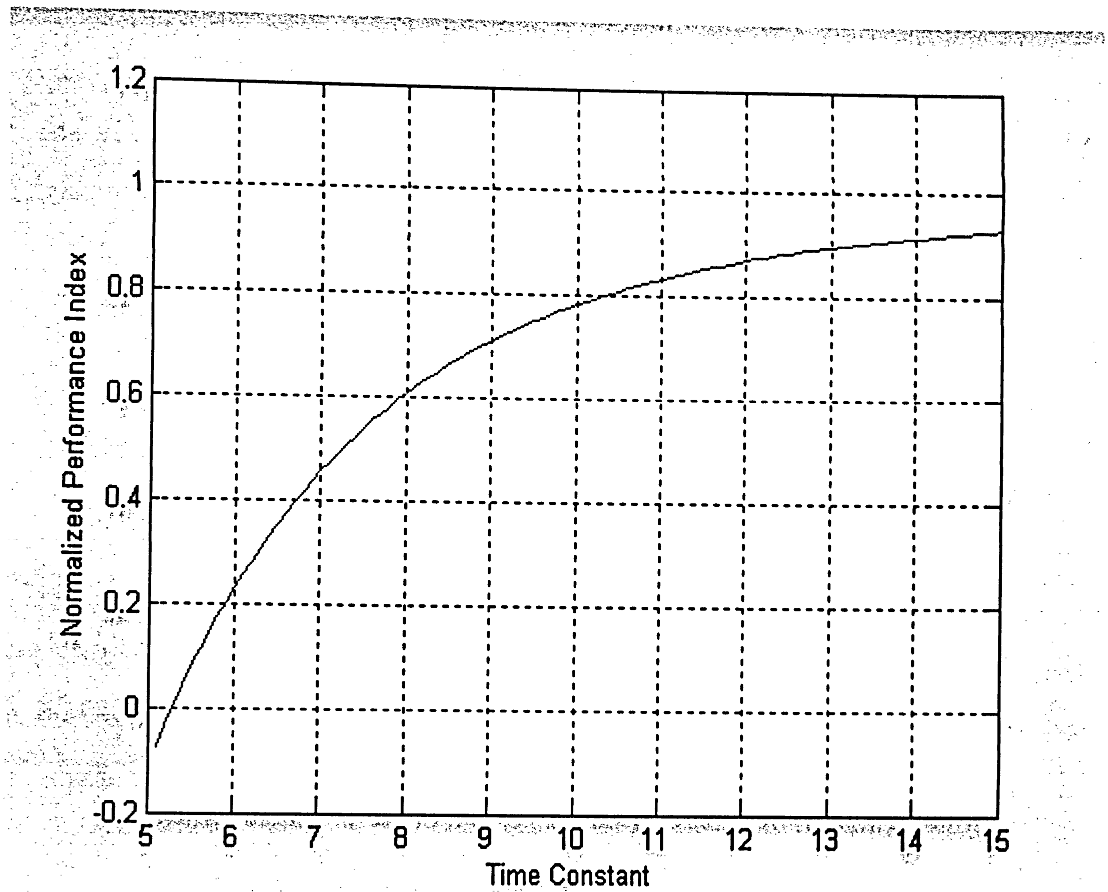

Figure 5.4 Effect of Change in Time Constant "seconds" on the Normalized Performance Index for the Standard System using Enumerative Search

The system was tested in the range of 50% to 150% of the original time constant. The result is very interesting, as it is normally known that the increase in time constant will move the process towards instability. However, these tuning parameters are more robust with higher time constant. There are two reasons behind this behavior. The first reason is related to the process as the slower the process is, the more robust to absorb stochastic disturbance. In other words, the system does not react quick enough to be affected. The second reason is related to the controller, as by using such high integral time, the controller movement is so limited that it mainly reacts on the proportional part.

Another point that could be noticed from Figure 5.4 is the slope, which is steeper when the time constant is low and less steep when the time constant is high. In other words, the performance index is more sensitive to reduction in time constant than the increase.

5.3 Effect of Change in Process Gain

The process time constant and the time delay were kept constant and the process gain is changed among a range of its possible values for a stable closed loop system until we reached a value where the performance index does not change significantly. The result is shown in Figure 5.5.

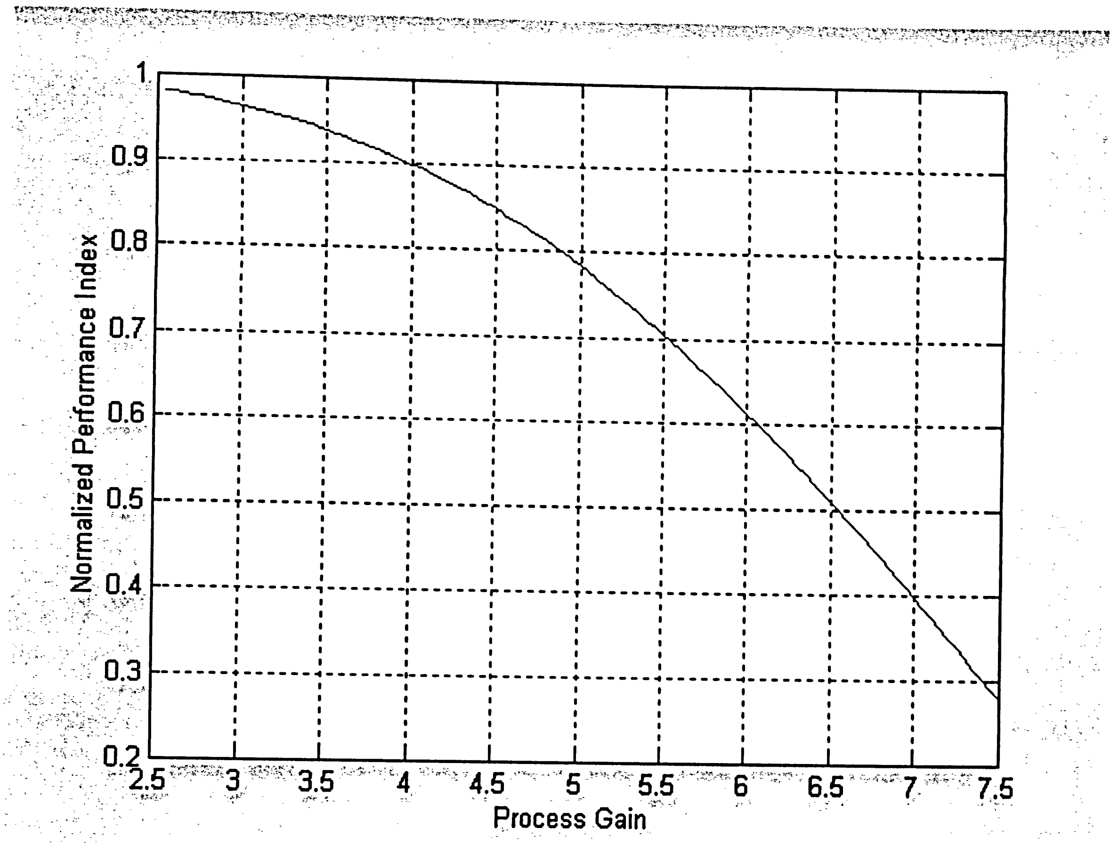

Figure 5.5 Effect of Change in Process Gain on the Normalized Performance Index for the Standard System using Enumerative Search

The system was tested in the range of 50% to 150% of the original process gain as shown in Figure 5.5. The results are excepted, as it is normally known that the increase in process gain will move the process towards instability. These tuning parameters are less robust with higher process gain. There is one reason behind this behavior. This reason is related to the process, as the lower the process gain, the less magnification of the stochastic noise or disturbance. In other words, the system does not amplify the noise with lower process gain.

In addition, it is clear that for the above-mentioned closed loop system that the performance index is very sensitive to process gain. It is noticeable that around the value of the process gain of 5, which is the original gain of the system, the performance index can be a good measurement for process gain change and as the gain increases, the performance index decreases.

5.4 Effect of Change in Time Delay

The process time constant and the gain were kept constant and the process time delay is changed among a range of its possible values for a stable closed loop system. It was noticed that the performance index does not change significantly. The result is shown in Figure 5.6.

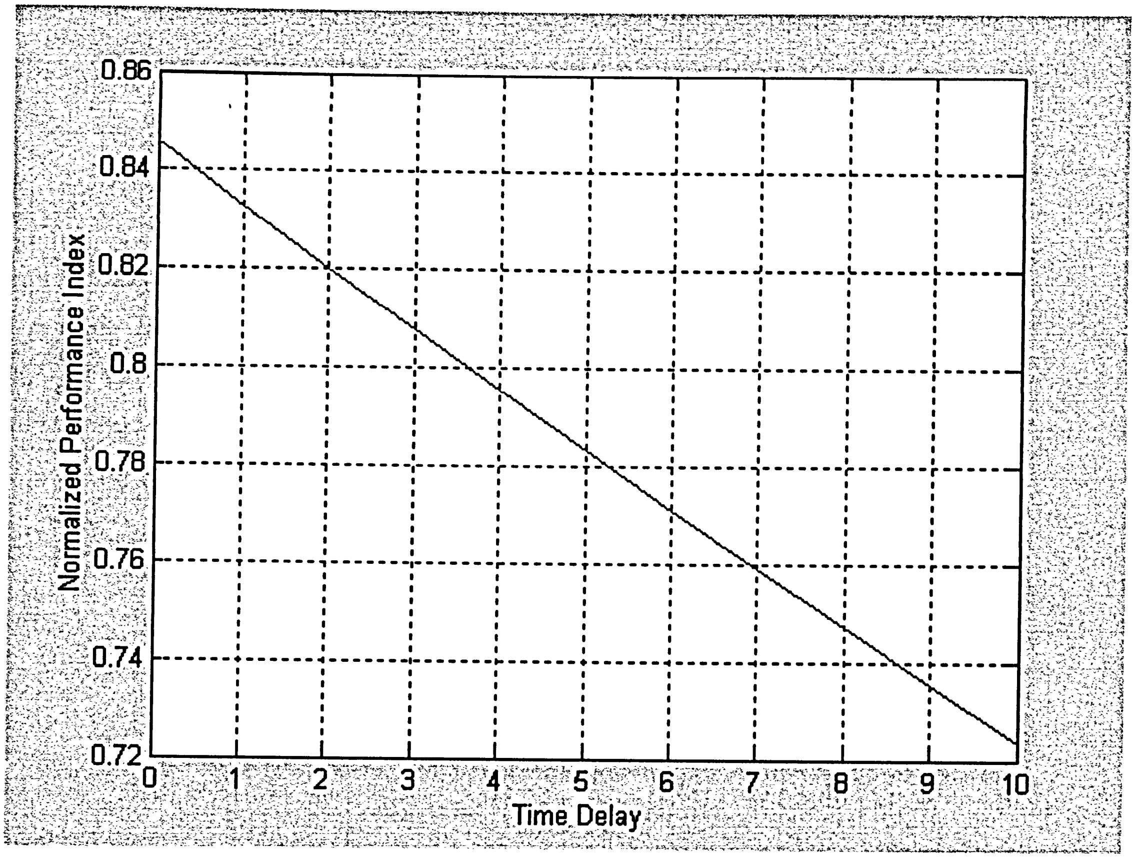

Figure 5.6 Effect of Change in Process Time Delay on the Normalized Performance Index for the Standard System using Enumerative Search

The system was tested in the range of 0% to 200% of the original process time delay. The results are excepted, as it is normally known that the increase in process time delay will move the process towards instability. There is one reason behind this behavior. This reason is related to the controller, as the minimum variance controller is very robust against time delay. In fact, it is recommended with loops that have high time delay. That is why the change in time delay has a less significant effect on the normalized performance index. Moreover, vice versa; normalized performance index cannot be used as an indication for time delay changes. In addition, it should be noticed that the relation is almost linear. However, this behavior was not noticed when the test was expanded to a higher time delay range as per Figure 5.7.

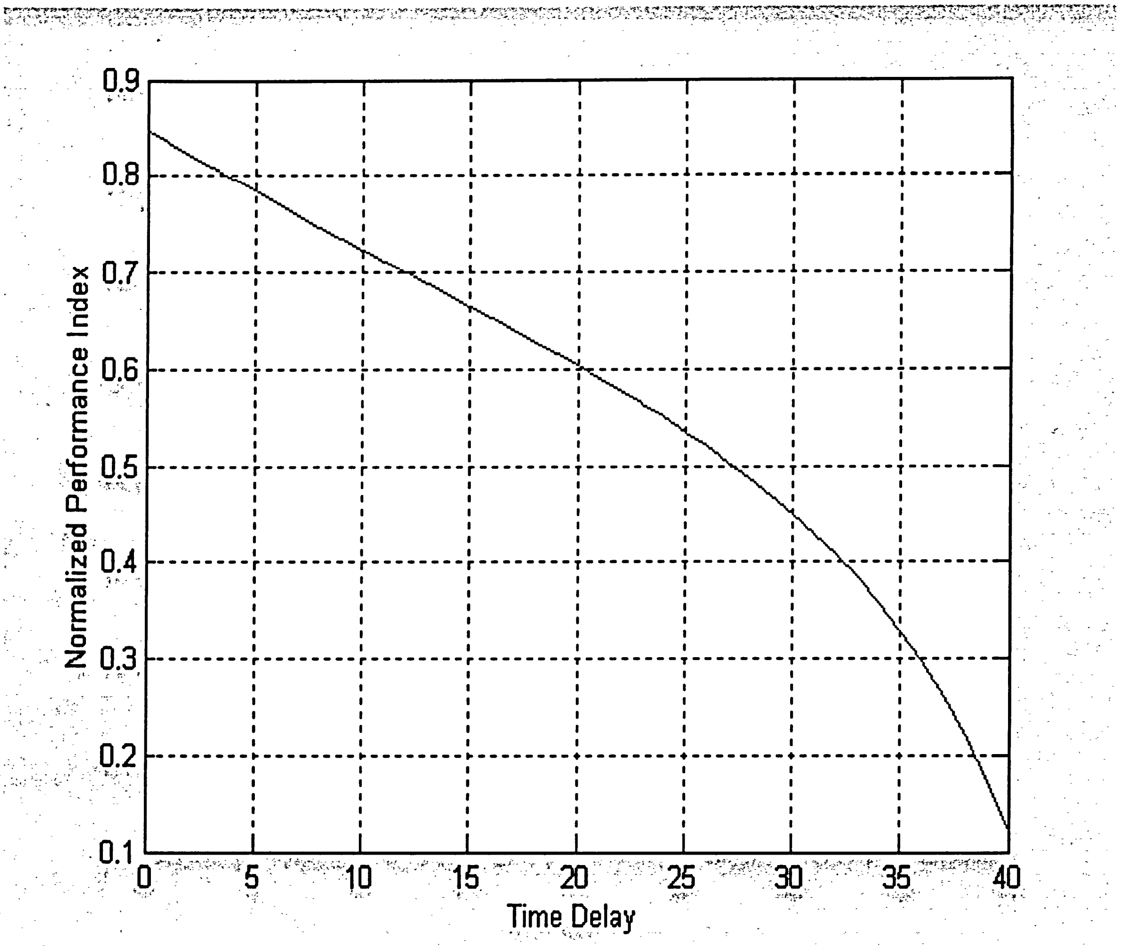

Figure 5.7 Effect of Large Change in Process Time Delay "seconds" on the Normalized Performance Index for the Standard System using Enumerative Search

Chapter 6

Proportional - Integral Controller (PI) Tuning by Performance Index Maximization

6.1 Introduction

Three different models are used for experimental testing. The three models came from three control loops that are located at the Falconbridge - Sudbury smelter. The process parameters were estimated prior to the optimization by performing a step test on each control loop. Then, identification and tuning software "ExperTune®" was used to identify the process parameters and find the best tuning values. The rules that were used by ExperTune® to estimate the initial tuning parameters were different but they are most commonly used in industry. The main algorithms were Lambda λ Tuning and IAE tuning [Shinskey (1988)].

Subsequently, the process data, which were estimated by ExperTune®, were used in two different optimization algorithms as described in Chapter 4. The optimization variables, which are the proportional gain and the integral gain, in addition to the objective function, which is the Normalized Performance Index, were calculated. The resultant values of the proportional and integral gain were compared against the existing tuning parameters that are used at Falconbridge using the following Simulink model. The new values have shown superior results in almost all cases.

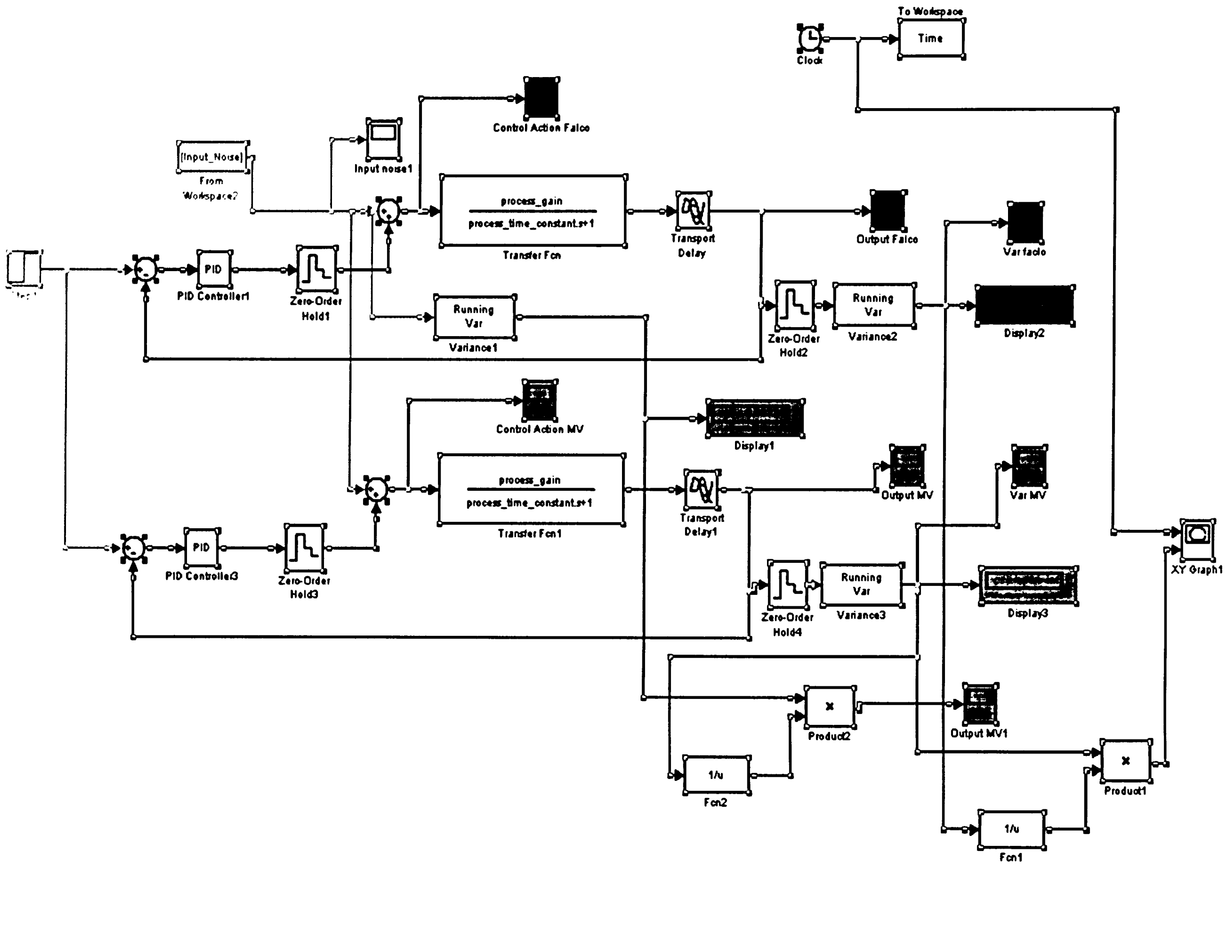

Figure 6.1 Schematic showing the Simulink model that is used to compare performances

The Simulink model is simply a model that subjects two different controllers for the same process model into two types of changes. The first change is stochastic process input disturbance, which is applied to both loops under comparison simultaneously. The second change is a change in the set point with a magnitude of one, which is also applied to both loops simultaneously. Although the minimum variance controller is not the best choice as a tracking controller but this test is important as it measures the controller robustness in a typical industrial application.

The running variance ratio was compared between both controllers. The ratio has the variance of the optimal controller in the numerator and the existing controller in the denominator. Therefore, if the ratio is more than one, the existing controller is better than the optimal controller. Moreover, if the ratio is less than one, the optimal controller is better than the existing controller.

6.2 Optimization using Enumerative Search

Roaster Air Blower Flow Loop

This is an essential control loop to control the draft in the roaster. The air is measured downstream the air blower and manipulates the dampers on the blower air inlet. This loop is a perfect candidate for minimum variance control application with input disturbance. The air temperature and wind speed are different types of stochastic input disturbance. Table 6.1 Table 6.4 shows the process parameters and the existing tuning parameters of this control loop.

Description	Roaster
	Air Blower Flow
Process Gain	1
Time Constant (s)	3.5
Time Delay (s)	8.5
Sampling Time (s)	1
FL - PB	2000
FL - Gain	0.05
FL - Integral Time (minutes/repeat)	0.21
FL - Integral Gain (repeats/second)	0.079

Table 6.1 Roaster Air Blower Control Loop Summary

where

FL	Falconbridge Loop
PB	Proportional band which is the unit that is used for gain in the smelter Foxboro control system
Integral Time	This is the unit that describes the integral action in the smelter Foxboro control system in minutes per repeat

The search for this system was done using enumerative search and the following results in Table 6.2 were obtained. The change in the value of the normalized performance index is sketched with changes in the value of the proportional gain as per Figure 6.2 and Figure 6.3. The same was done with respect to integral gain as per Figure 6.2 and Figure 6.4.

Normalized Performance Index	0.99
MV - Gain	0.4
MV - PB	250
MV - Integral Gain (repeats/second)	0.05
MV – Integral Time (minutes/repeat)	0.33

Table 6.2 Roaster Air Blower Enumerative Search Optimal Results

It should be mentioned that the value of 0.05 for the integral gain was used as the lower bound in the search algorithm. In other words, the result can be better by lowering this limit. The reason is mentioned before as higher performance index value normally obtained at low integral gain values. However, there is a practical limit should be followed in the selection of this value; otherwise the controller will loose its tracking properties.

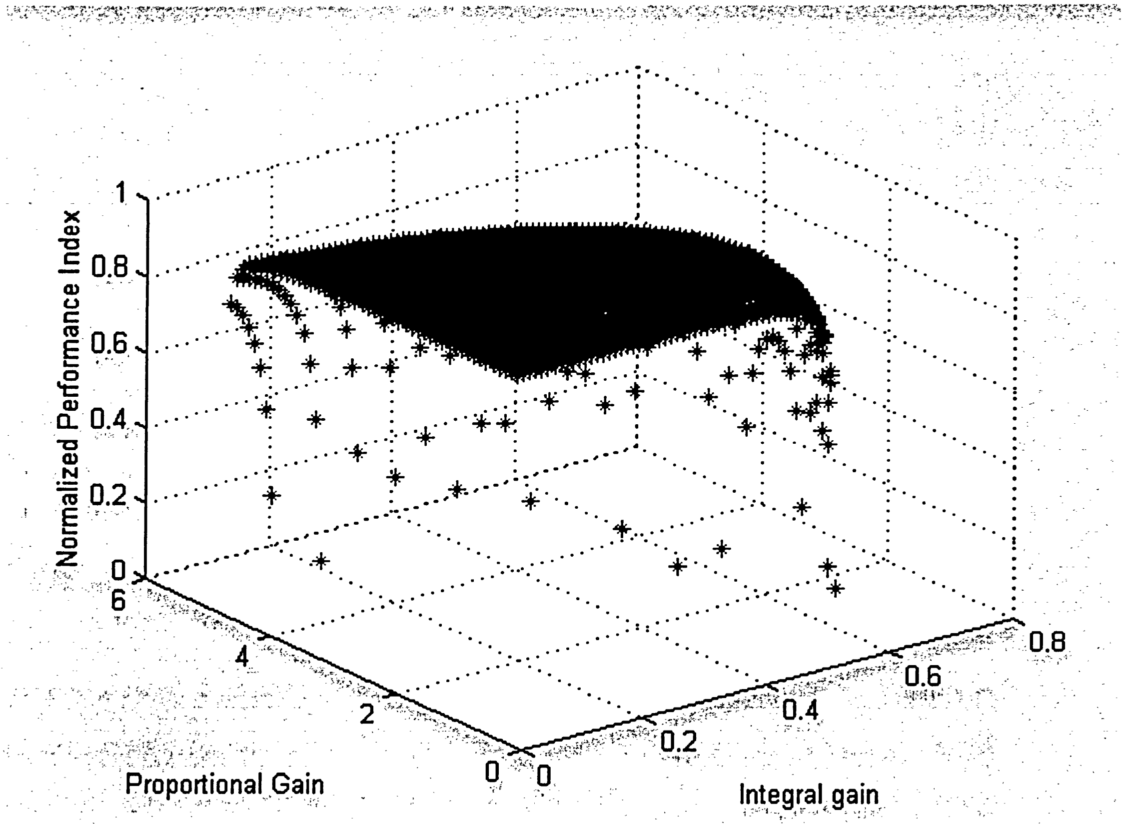

Figure 6.2 3-D Diagram showing the change in objective function with different optimization variables for the roaster air blower flow control loop using Enumerative Search

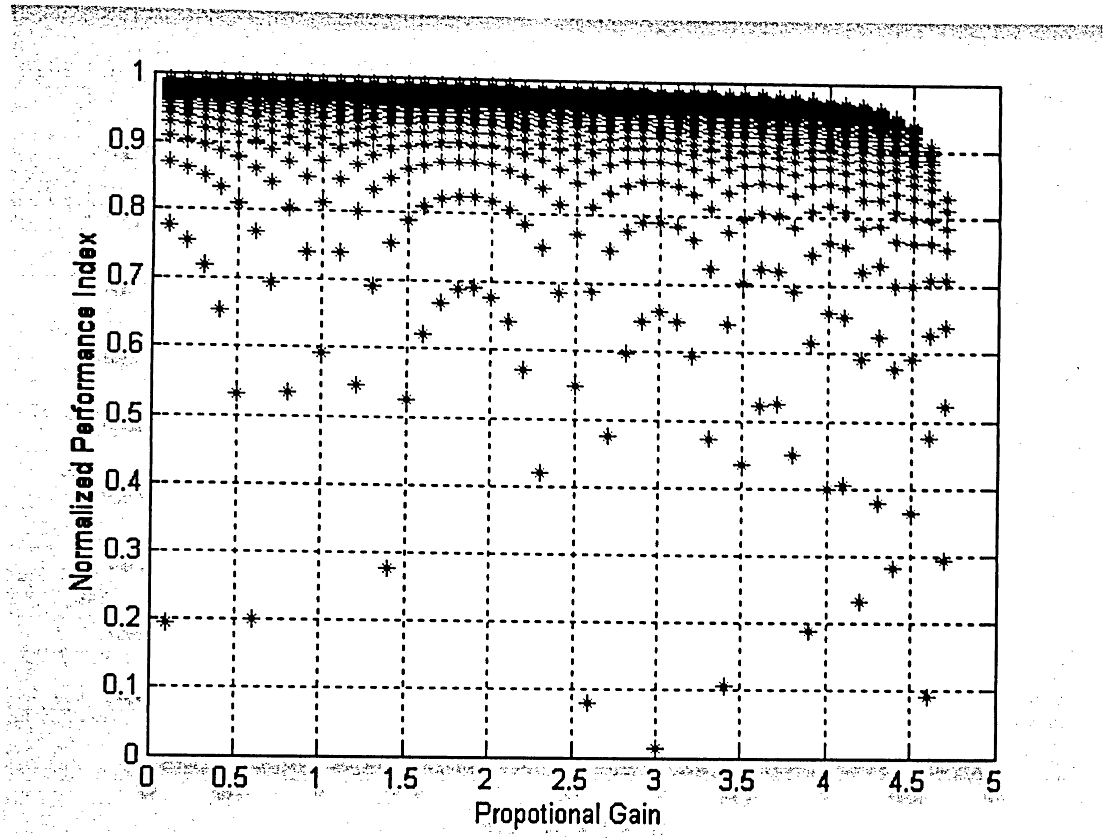

Figure 6.3 Change in objective function with different proportional gains for the roaster air blower flow control loop using Enumerative Search

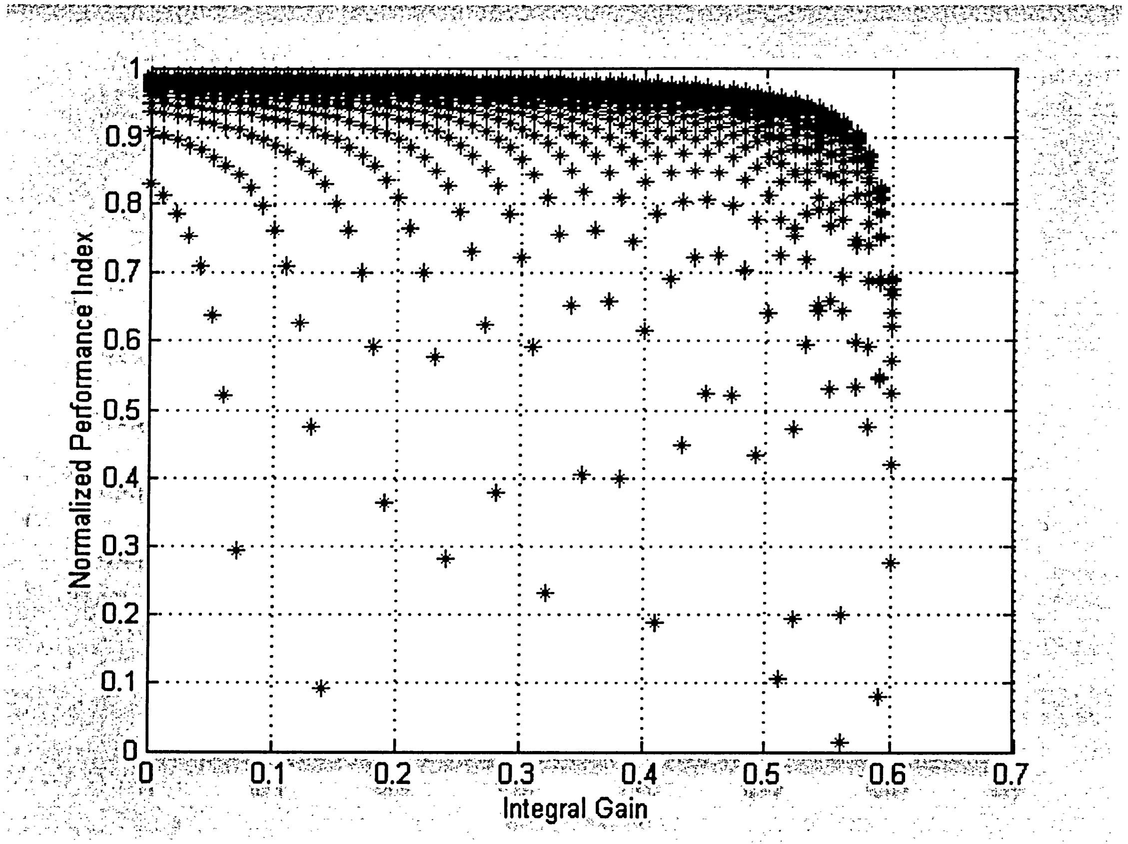

Figure 6.4 Change in objective function with different integral gains for the roaster air blower flow control loop using Enumerative Search

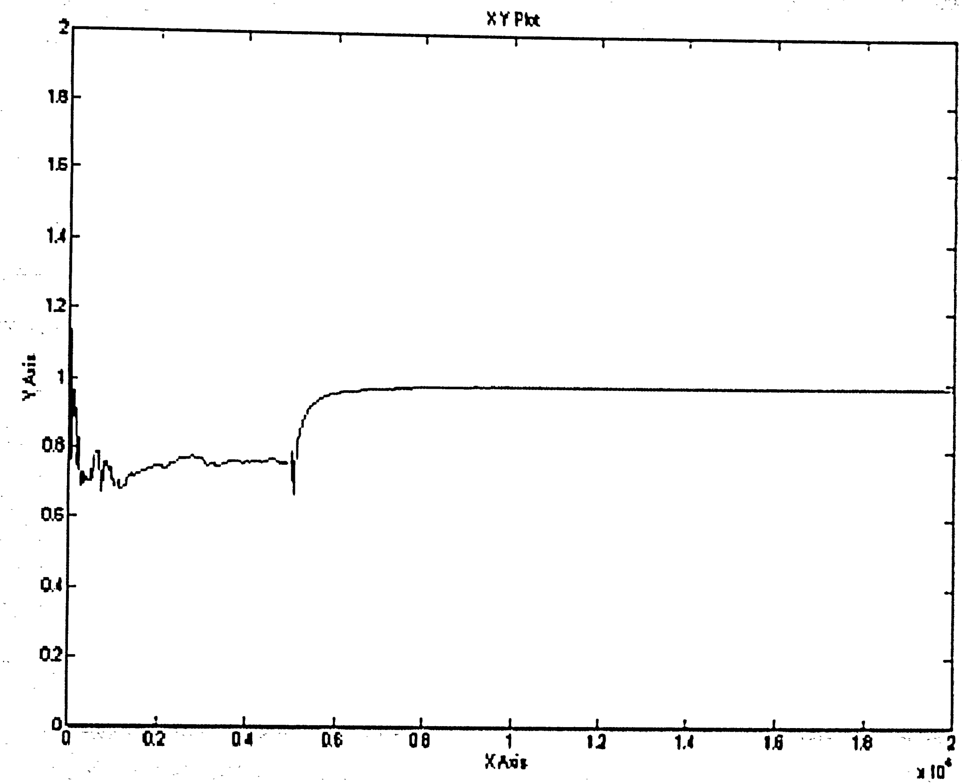

Figure 6.5 Running Variance Ratio (Y – Axis) between the optimal controller variance using Enumerative Search and the existing controller variance with time (X - Axis) for the roaster air blower flow control loop - Set point change at time 5000

It is clear from Figure 6.5 that the enumerative search has yielded better results than the existing controller.

When the lower bound for the integral gain was lowered to 0.001, the performance index increased slightly as per Table 6.3. Although it has shown greater results as per Figure 6.6 from the minimum variance point of view, it is not superior enough to jeopardize the other tracking properties.

Normalized PI	0.99
MV - Gain	0.3
MV - PB	333.33
MV - Integral Gain (repeats/second)	0.001
MV – Integral Time (minutes/repeat)	16.67

Table 6.3 Roaster Air Blower Enumerative Search Optimal Results with lower integral gain bound

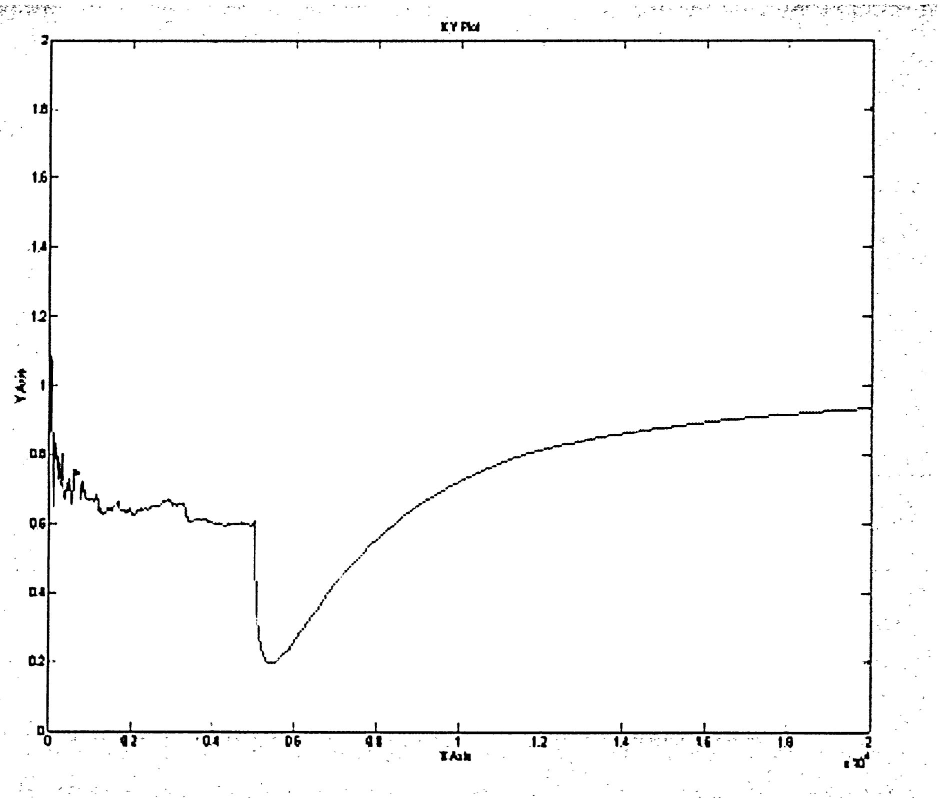

Figure 6.6 Running Variance Ratio (Y – Axis) between the optimal controller variance using Enumerative Search and the existing controller variance with time (X - Axis) for the roaster air blower flow control loop with a lower bound of the integral gain equal to 0.001- Set point change at time 5000

Montcalm Repulper Feed Flow Rate

This control loop is one of the loops that are subject to many different kinds of disturbance. The slurry flows from an eight-leg header, called the octopus, into two repulpers, two disc filters, and one return line that is equipped with a pressure control valve. Every line is equipped with a set of valves with different gains and actuators. Opening and closing the other valves disturbs the flow in the header. This normally happens with a random effect.

Another challenge in this loop is to minimize the flow variance, which may disturb the level in the small repulper if not tightly controlled. Overflow of the repulper can lead to an expensive cleaning job and low level can damage the repulper discharge valve. Table 6.4 shows the process parameters and the existing tuning parameters of this control loop.

Description	Montcalm Repulper Feed Flow Rate
Process Gain	12.9
Time Constant (s)	24.75
Time Delay (s)	28
Sampling Time (s)	1
FL - PB	400
FL - Gain	0.25
FL - Integral Time (minutes/repeat)	0.625
FL - Integral Gain (repeats/second)	0.0267

Table 6.4 Montcalm Repulper Feed Flow Rate Control Loop Summary

The search for this system was done using enumerative search first and the following results in Table 6.5 were obtained. The change in the value of the normalized performance index is sketched with changes in the value of the proportional gain as per Figure 6.7 and Figure 6.8. The same was done with respect to integral gain as per Figure 6.7 and Figure 6.9.

Normalized Performance Index	0.51
MV - Gain	0.07
MV - PB	1428.57
MV - Integral Gain (repeats/second)	0.001
MV – Integral Time (minutes/repeat)	16.67

Table 6.5 Montcalm Repulper Feed Flow Rate Enumerative Search Optimal Results

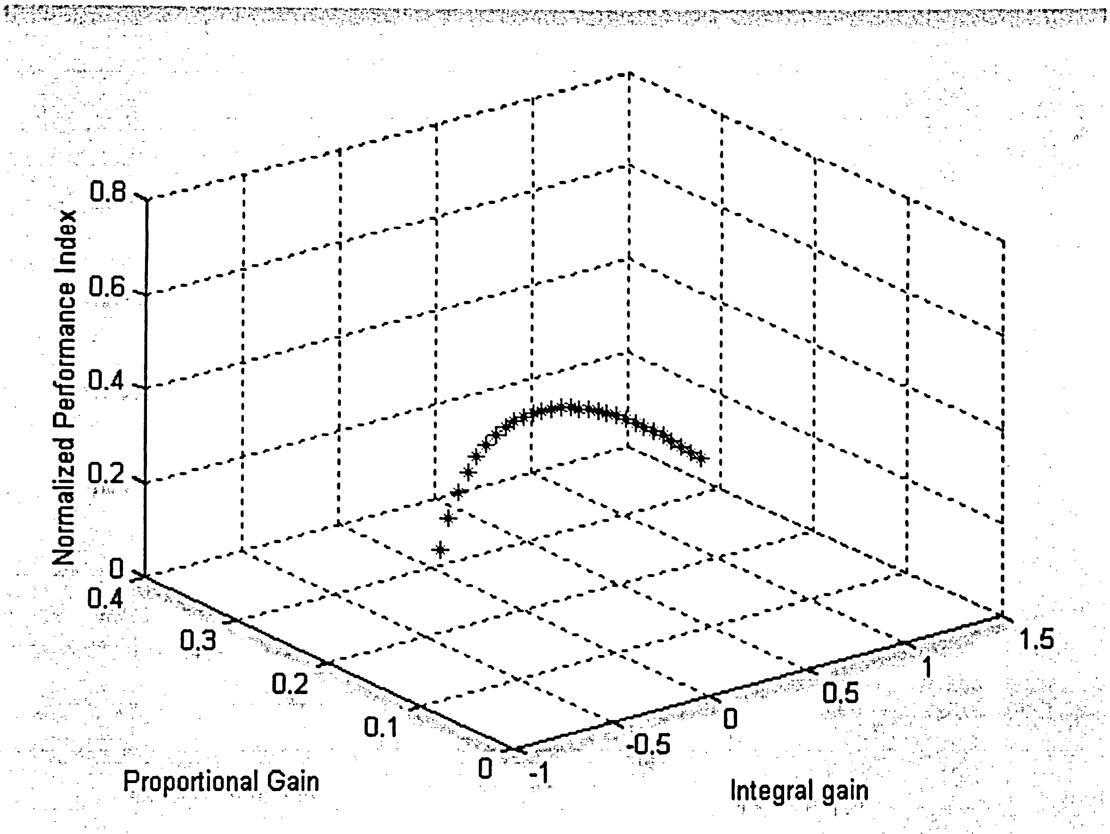

Figure 6.7 3-D Diagram showing the change in objective function with different optimization variables for Montcalm Repulper feed flow rate control loop using Enumerative Search

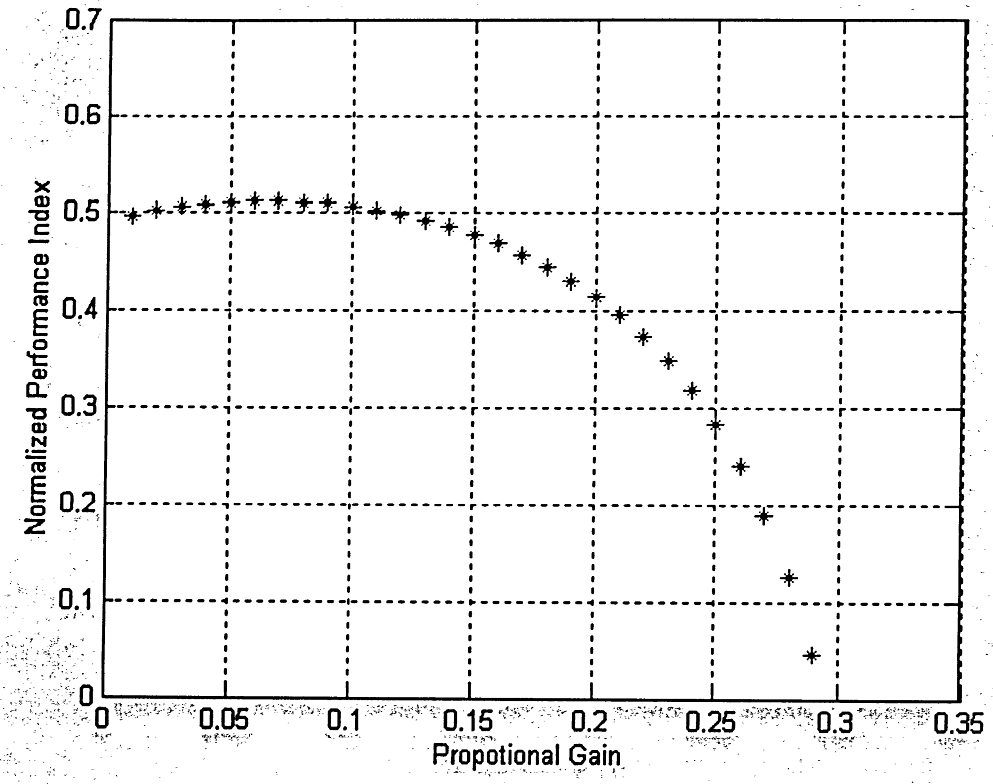

Figure 6.8 Change in objective function with different proportional gains for the Montcalm Repulper flow rate control loop using Enumerative Search

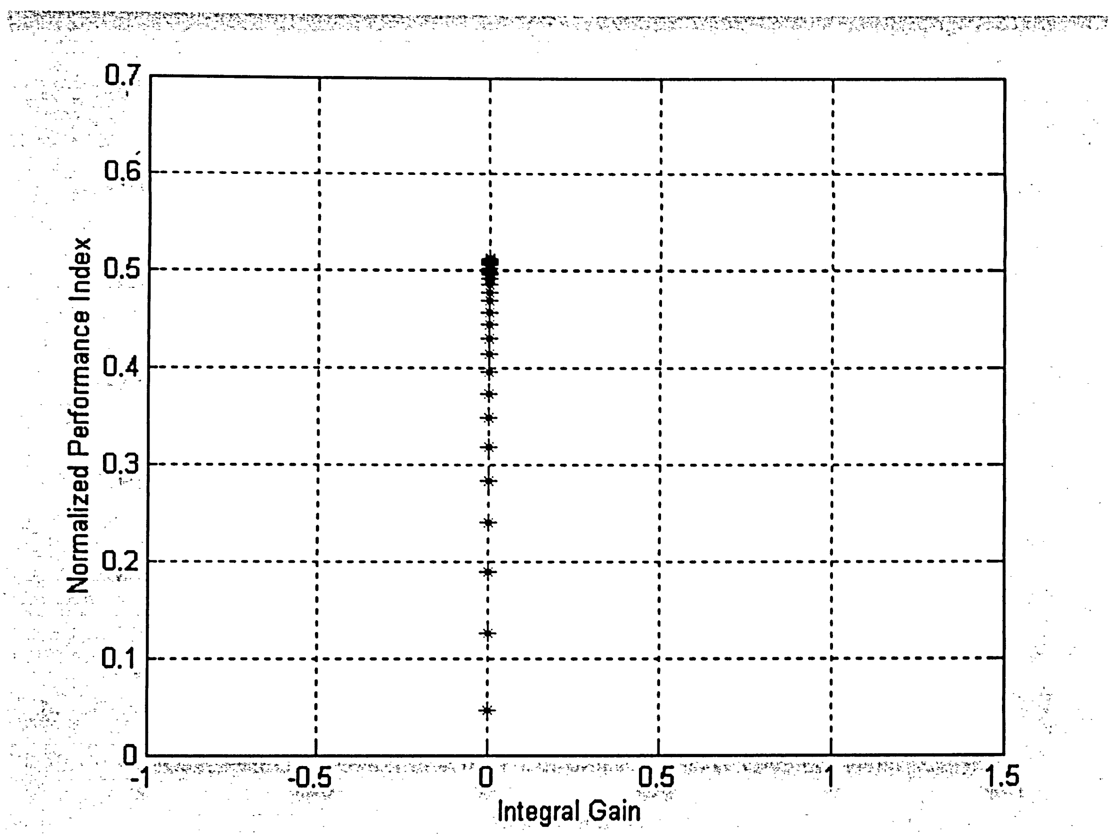

Figure 6.9 Change in objective function with different integral gains for Montcalm Repulper feed flow rate control loop using Enumerative Search

It should be noticed in Figure 6.7 and Figure 6.9 that this system is hard to tune on minimum variance basis. On the other hand, the enumerative search may not be successful enough to tackle this problem. It has shown only one integral gain value, which is the lower bound. This data can be only verified if another search algorithm is used.

However, when the loop compared against the existing tuning parameters, it has shown better performance as per Figure 6.10a and Figure 6.10b. Moreover, the existing tuning parameters have led to unstable behavior and the ratio between the variance of the optimal parameters and the existing parameters is almost zero.

Further investigation has shown that the existing loop is currently in manual mode because of the existing controller behavior. This means that the existing controller cannot control the process. The used tuning rules were not good enough to tune this loop.

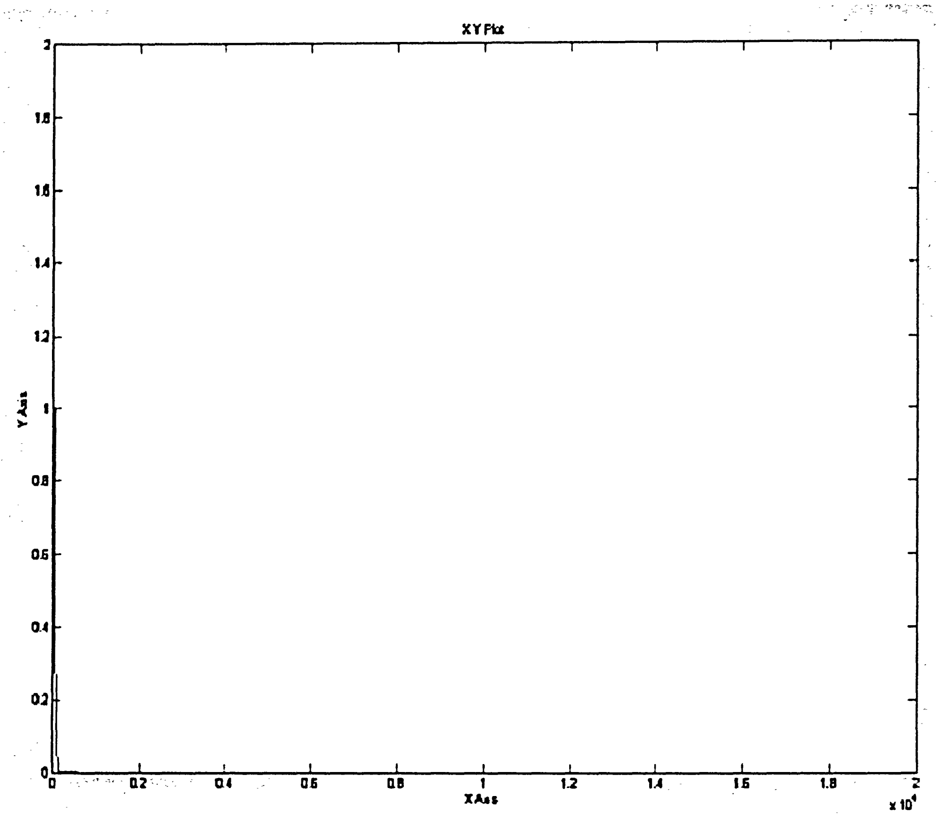

Figure 6.10a Running Variance Ratio (Y – Axis) between the optimal controller variance using Enumerative Search and the existing controller variance with time (X - Axis) for Montcalm Repulper feed flow rate control loop - Set point change at time 5000

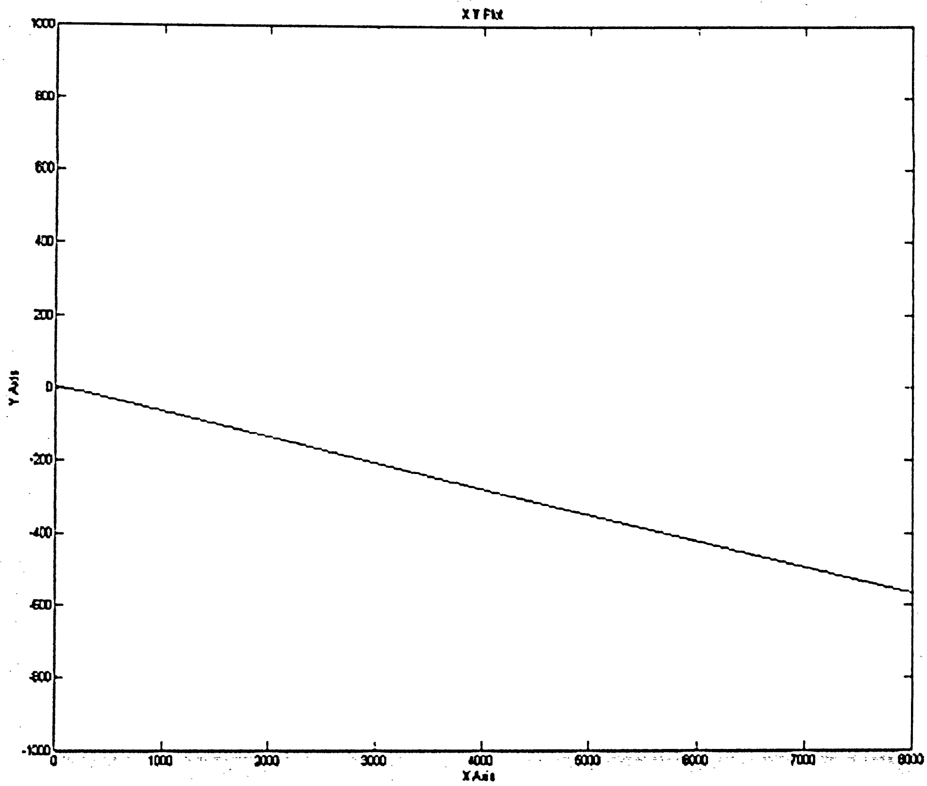

Figure 6.10 –b Running Variance Logarithmic Ratio (Y – Axis) between the optimal controller variance using Enumerative Search and the existing controller variance with time (X - Axis) for Montcalm Repulper feed flow rate control loop - Set point change at time 5000

Roaster Feed Flow Rate

This control loop is one of the most important control loops in the smelter as it determines the smelter hourly production. The flow is measured using a corilois mass flow meter and it adjusts the reciprocating pump speed. Plugging and different foreign material in the feed can definitely disturb this loop in a random behavior. The challenge in this loop is to minimize the flow variance, which will disturb the roaster operation if not tightly controlled. Table 6.7 shows the process parameters and the existing tuning parameters of this control loop.

Description	Roaster
	Feed Flow Rate
Process Gain	0.67
Time Constant (s)	7
Time Delay (s)	11
Sampling Time (s)	1
FL - PB	380
FL - Gain	0.26
FL - Integral Time (minutes/repeat)	0.23
FL - Integral Gain (repeats/second)	0.072

Table 6.6 Roaster Feed Flow Rate Control Loop Summary

The search for this system was done using enumerative search and the following results in Table 6.7 were obtained. The change in the value of the normalized performance index is sketched with changes in the value of the proportional gain as per Figure 6.11 and Figure 6.12. The same was done with respect to integral gain as per Figure 6.11 and Figure 6.13.

Normalized Performance Index	0.99
MV - Gain	64.31
MV - PB	1.55
MV - Integral Gain (repeats/second)	0.078
MV – Integral Time (minutes/repeat)	0.21

Table 6.7 Roaster Feed Flow Rate Enumerative Search Optimal Results

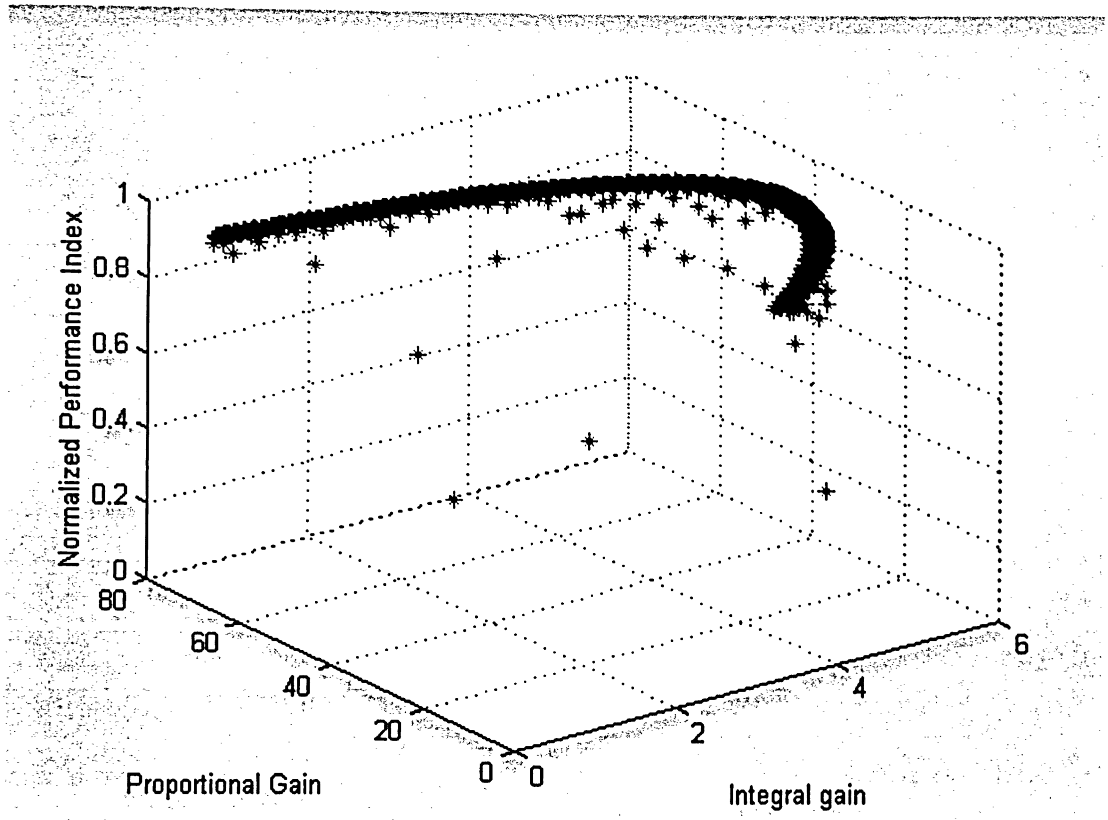

Figure 6.11 3-D Diagram showing the change in objective function with different optimization variables for the roaster feed flow rate control loop using Enumerative Search

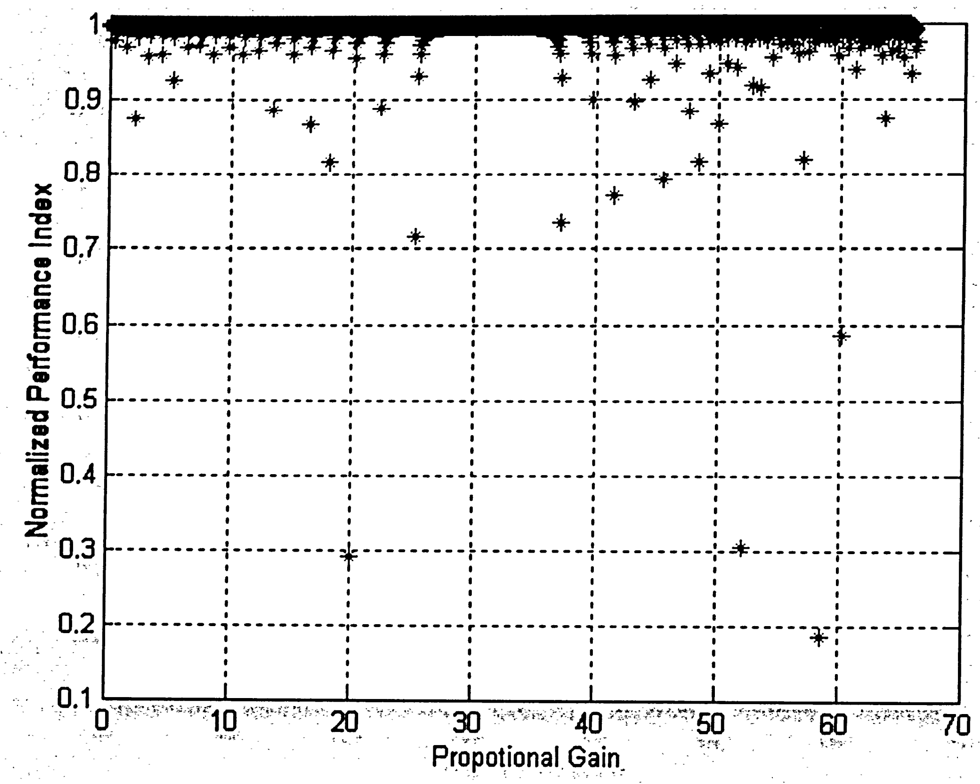

Figure 6.12 Change in objective function with different proportional gains for the roaster feed flow rate control loop using Enumerative Search

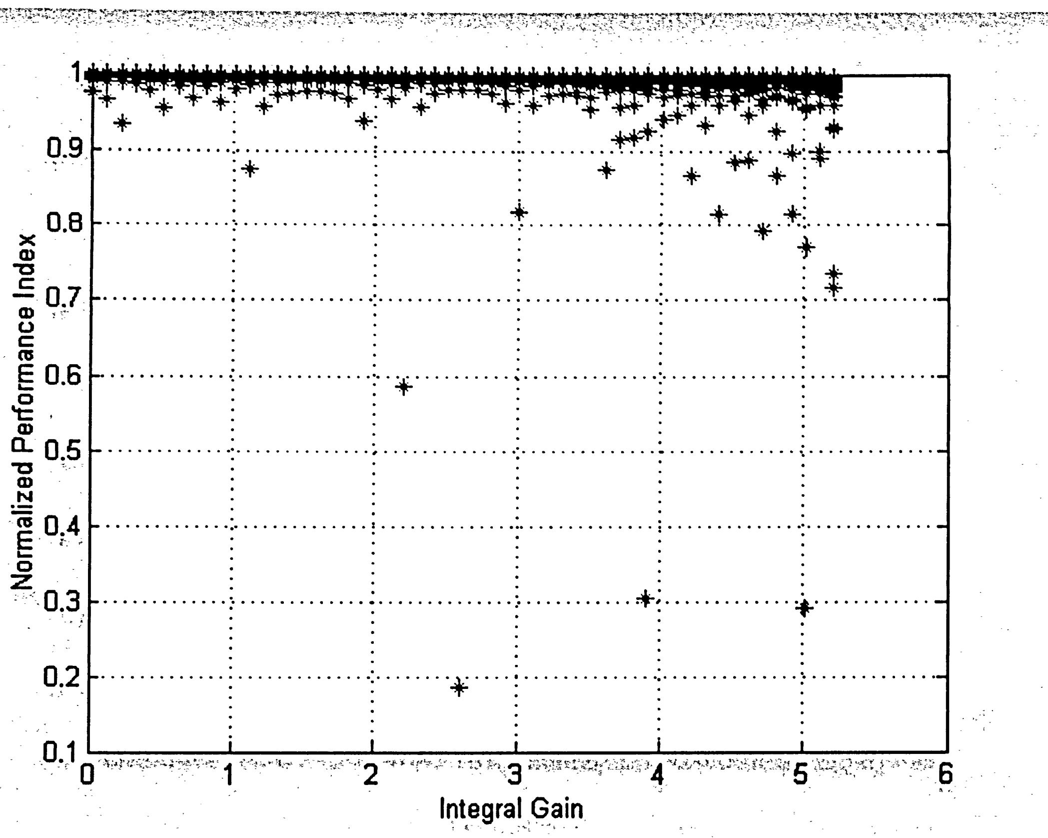

Figure 6.13 Change in objective function with different integral gains for the roaster feed flow rate control loop using Enumerative Search

The enumerative search does not seem to be successful enough to tackle this problem. The reason is the location of local minimum values. The optimum point is located among other good values and the results can be refined with smaller step size. When a smaller step size was used, the program calculation time took approximately 64 hours.

6.3 Optimization using Hybrid Genetic Algorithms

Roaster Air Blower Flow Loop

The same optimization was done using hybrid genetic algorithms, which has proven itself quick and reliable in finding the optimal parameters.

Normalized Performance Index	0.89
MV – Gain	0.069
MV – PB	1459.51
MV - Integral Gain (repeats/second)	0.0014
MV – Integral Time (minutes/repeat)	11.81

Table 6.8 Roaster Air Blower GA Search Optimal Results

Although the normalized performance index value shown above in Table 6.8 is lower than the one that was obtained by the enumerative search, but in fact it yielded better performance when it was compared with the enumerative search controller as shown in Figure 6.14.

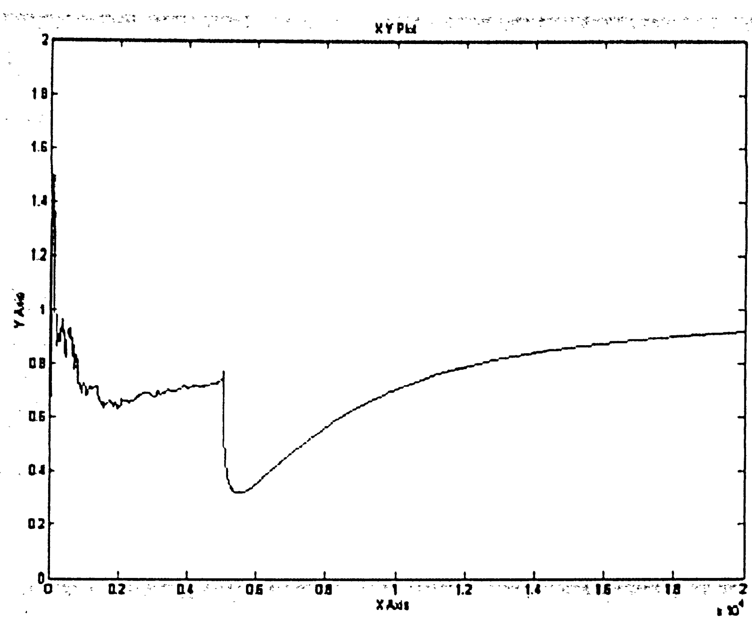

Figure 6.14 Running Variance Ratio (Y – Axis) between the hybrid GA optimal controller variance and the enumerative search optimal controller variance with time (X - Axis) for the roaster air blower flow control loop - Set point change at time 5000

The reason for this behavior is related to the bounds that were added during the optimization process. The output was clamped between 1 and –1 to confirm stability. On the other hand, the stability for the enumerative search was checked by the location of the closed loop poles, which gave it more freedom to move in the search space. This is the reason to explain why the GA has yielded lower dynamic variance output when both methods were compared against each other.

Another sensitivity analysis was done to check the robustness of the solution. The stochastic input disturbance random generator seed was changed and the results did not change significantly. These results in Table 6.9 have shown the real strength of the normalized performance index as a ratio instead of the output variance on its own.

Random Seed	Seed 1	Seed 2	Seed 3
Normalized Performance Index	0.889	0.891	0.886
MV - Gain	0.0685	0.0675	0.0680
MV - PB	1459.51	1482.12	1469.72
MV - Integral Gain (repeats/second)	0.0014	0.0014	0.0011
MV – Integral Time (minutes/repeat)	11.8	11.8	16.7

Table 6.9 Roaster Air Blower GA Sensitivity Analysis for Random Seeds

Another comparison was performed to demonstrate another powerful characteristic of hybrid GA, which is the optimization time. From Table 6.10, it is clear that hybrid genetic algorithm takes significantly shorter time compared with enumerative search.

Loop	Enumerative Search	Hybrid Optimization
Roaster Blower	23 min	13.9 s
Roaster Feed	2.1 h	46.3 s
Repulper Feed	64 h	43.2 s

Table 6.10 Time comparison between Enumerative Search and Hybrid GA

Montcalm Repulper Feed Flow Rate

The same optimization was done using genetic algorithms, which again has proven itself quick and reliable in finding the optimal parameters. The results are shown in Table 6.11.

Normalized Performance Index	0.97
MV – Gain	0.26
MV – PB	380.23
MV - Integral Gain (repeats/second)	0.0015
MV – Integral Time (minutes/repeat)	10.82

Table 6.11 Montcalm Repulper feed flow rate GA Search Optimal Results

It is clear from Figure 6.15 that the yielded performance index is close to the maximum and much better than the enumerative search optimization

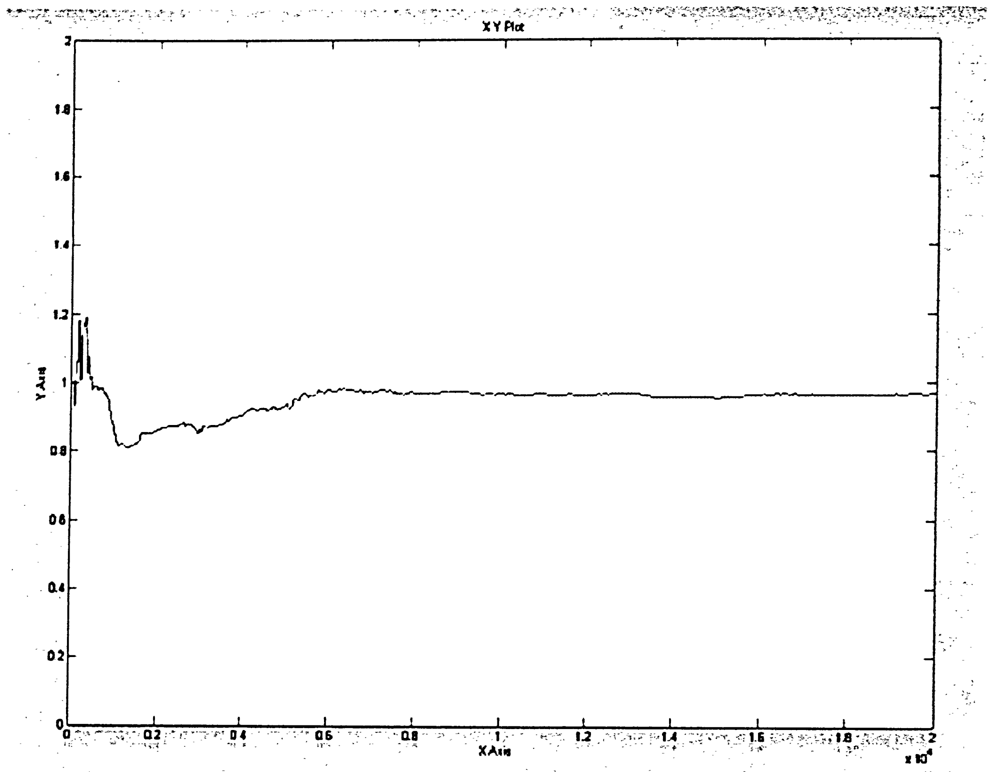

Figure 6.15 Running Variance Ratio (Y – Axis) between the optimal hybrid GA controller variance and the optimal enumerative search controller variance with time (X - Axis) for Montcalm Repluper Feed flow control loop - Set point change at time 5000

Roaster Feed Flow Rate

The same optimization was done using genetic algorithms, which again has proven itself quick and reliable in finding the optimal parameters. The results are shown in Table 6.12.

Normalized PI	0.22
MV – Gain	0.043
MV – PB	2346.7
MV - Integral Gain (repeats/second)	0.0015
MV – Integral Time (minutes/repeat)	11.47

Table 6.12 Roaster feed flow rate GA Search Optimal Results

The enumerative search failed to achieve a practical lower value that is why the comparison this time is done between the existing controller and the optimal controller that was designed by hybrid GA. The results are shown in Figure 6.16. These results reassured again that GA yields better results than enumerative search.

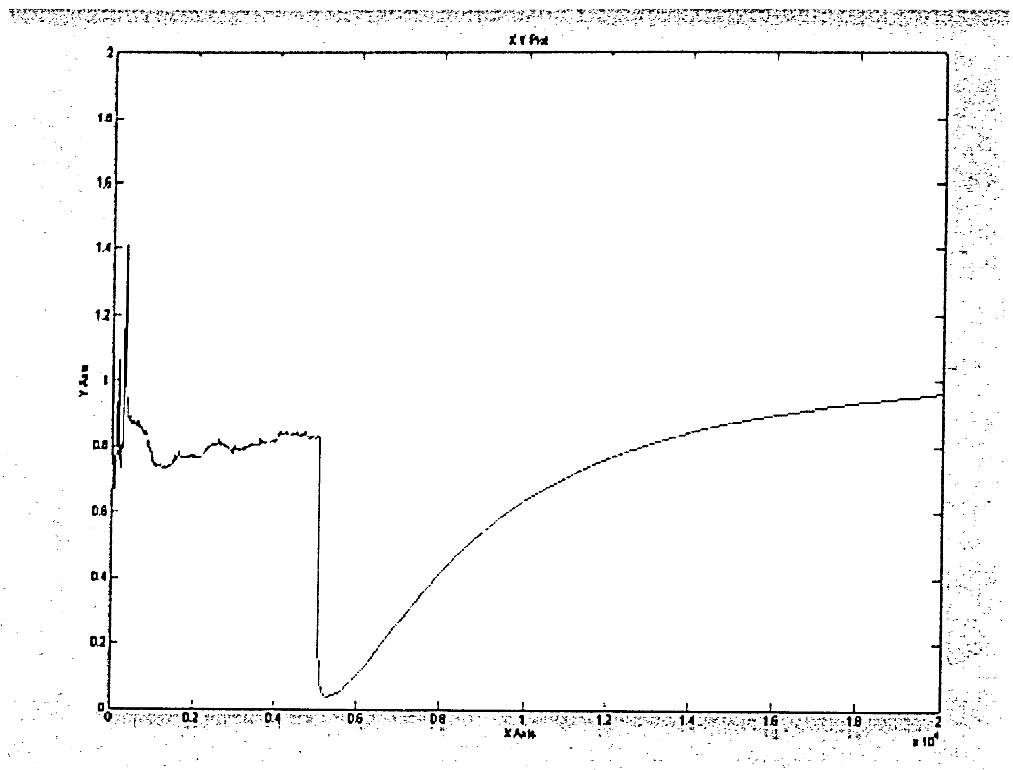

Figure 6.16 Running Variance Ratio (Y – Axis) between the optimal hybrid GA controller variance and the existing controller variance with time (X - Axis) for Roaster Feed flow control loop - Set point change at time 5000

Table 6.13 summarizes the results obtained with the three loops.

Parameter	Roaster Air Blower	Montcalm Feed Flow Rate	Roaster Feed Flow Rate
Gain	1	12.9	0.67
Time Constant (s)	3.5	24.75	7
Time Delay (s)	8.5	28	11
Sampling Time (s)	1	1	1
Integral Gain - FL	0.05	0.25	0.263
Proportional Gain - FL	0.079	0.027	0.072
Integral Gain - Enumerative search	0.05	0.001	No Solution
Proportional Gain - Enumerative search	0.4	0.07	No Solution
% Actual Improvement - Enumerative search	**26.8%**	**100%**	**0%**
Integral Gain - Hybrid GA	0.0014	0.0015	0.001
Proportional Gain - Hybrid GA	0.068	0.04	0.21
% Actual Improvement - Hybrid GA	**35.7%**	**109.1%**	**21%**

Table 6.13 Results Summary

The percentage increase in the performance improvement was calculated from the ratio of the two loops variances in the Simluink model over a time span of 5000 seconds. The following Equation is used

% Actual Improvement = (Var (FL) – Var (Opt)) X 100 / Var (FL)

where Var (FL) Variance of Falconbridge loop

Var (Opt) Variance of the optimized loop (either Enumerative pr Hybrid GA)

The performance improvement for Montcalm loop is considered 100%, as the existing loop is unstable. The benchmarking for the GA optimization method is based on comparison with the enumerative search method.

Table 6.13 summarizes all results. It has shown the weakness of the enumerative search when it is compared with Hybrid Genetic Algorithms, as GA was able to find a feasible solution in all cases. In addition, the Hybrid GA has always yielded better solution than the existing tuning methods or the enumerative search.

Part V

Conclusion

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This research focused on the tuning of the proportional - integral (PI) controller in order to achieve a closed loop performance that is as close as possible to minimum variance controller. The work was directed towards minimum variance control to mitigate the effect of input stochastic disturbance.

The criterion of minimum variance tuning, as it was established by Harris, was based on minimizing the ratio between the variance in the stochastic input noise and the variance of the output. In order to scale this index between zero and one, the following normalized index was introduced for comparison and benchmarking.

$$N_{Pi} = \mathrm{Var}\,(N_i)\,/(\sigma_y^2 + \mathrm{Var}\,(N_i)) \qquad 7.1$$

An objective function, which is the performance index, was formulated. The closed loop output was mathematically driven into an explicit format, which is considered an immense contribution in research. Afterwards, the maximum performance index was calculated by two different methods, which were enumerative search and hybrid Genetic Algorithms. The constraints for this optimization problem were non-linear and implicit. Therefore, an alternative approach was used to put them in an explicit format for the Hybrid Genetic Algorithm optimization; and to clamp the process output at a certain high enough value that will be only violated if the closed loops poles are positive.

Three different models were used for experimental testing. The three models came from three control loops that are located at Falconbridge - Sudbury smelter. Subsequently, the process data were used in the two different optimization algorithms. The optimum values of the proportional and integral gains were compared against the existing tuning parameters at Falconbridge using Simulink model. The optimized values have shown superior results in almost all cases.

Hybrid genetic algorithms demonstrated very promising distinctiveness. It provided a global solution that had better minimum variance characteristics than the enumerative search technique, IAE and Lambda tuning. In addition, it was very quick in locating the optimum values. While the improvement in the performance index was 28.6%, 100% and 0% using enumerative search, the improvement was higher using Hybrid Genetic Algorithms (35.7%, 109.1% and 21%).

Finally, by analyzing the normalized performance index sensitivity, the following results were noticed. Process gain changes affected the normalized performance index considerably. Time constant changes affected the normalized performance index considerably. Time delay changes did not affect the normalized performance to a great extend

7.2 Recommendations for Future Work

1. The first recommendation for future work can be based on Horch (1999) work. They have modified Harris performance index. The minimum variance control is based on cancellation of the model dynamics and so placing all closed poles in the origin. This makes the minimum variance controller non applicable in practical life because of its low tracking capability. They suggested the placement of one pole using either control design guideline or additionally available process knowledge.

The above-mentioned approach can be as promising as it can help the controller servo capability. Hybrid genetic algorithms can be used to tackle this problem after implying an extra constraint, which is the location of the new pole.

2. The second recommendation can be the utilization of the great potential of Hybrid Genetic Algorithms for Multi-input and Multi-output (MIMO) problems. Huang (1997) approach can be fostered and expanded by using Hybrid Genetic Algorithms optimization.

3. Finally, the explicit function for the closed loop output can be used extensively for other methods in the controller design. The objective function and the relevant derivatives are in an explicit formula that can be used in any single or multi-object optimization processes.

Bibliography

Astrom, K.J. and Wittenmark, B., *Adaptive Control*, Second Edition, Addison-Wesley Publishing Company, Inc., 1995.

Astrom, K.J. and Wittenmark, B., *Computer Controller Systems Theory and Design*, Prentice Hall Information and System Sciences Series, 1997.

Astrom, K.J. and Wittenmark, B., *On Self Tuning Regulators*, Automatica, Vol. 9, pp. 185-199, 1973.

Astrom, K.J., *Introduction to Stochastic Control Theory*, Academic Press Inc, 1989.

Coley, D.A. *An Introduction to Genetic Algorithms for Scientists and Engineers*, 2nd edition, World Scientific, 1999 p. 78.

Desborough L., *Performance Assessment Measures for Univariate Feedforward / Feedback Control*", The Canadian Journal of Chemical Engineering, Vol.71, pp. 605-616, Dec.1993.

Eveleigh, V.W., *Adaptive Control and optimization Techniques*, McGraw-Hill, Inc., 1967.

Goldberg, D.E., *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison-Wesley, p. 124, 1989.

Goldberg, D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Kluwer Academic Publishers, 1989.

Goldberg, D.E., *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*, Addison-Wesley, 2002.

Harris, T.J. and Desborough L., *Performance Assessment Measures for Univariate Feedback Control*, The Canadian Journal of Chemical Engineering, Vol.70, pp. 1186-1197, Dec.1992.

Harris, T.J., *Assessment of Control Loop Performance*, The Canadian Journal of Chemical Engineering, Vol.67, pp. 856-861, Oct.1989.

Harris, T.J., Bourdeau, F. and MacGregor, J.F., *Performance Assessment of Multivariable Feedback Controllers*, Automatica, Vol.32, pp. 1505-1518,1996.

Horch, A. and Alf J., *Modified Index for Control Performance Assessment*, Journal of Process Control, Vol.9, pp. 475-483, 1999.

Huang, B. and Shah, S. L., *Performance Assessment of Control Loops*, Springer, 1999.

Huang, B., Shah, S.L., Kwok, E.K., *Good, Bad or Optimal? Performance Assessment of Multivariable Processes*, Automatica, Vol.33, pp. 1175-1183, 1997.

Ko, B. and Edgar, T., *PID Control Performance Assessment: The Single Loop Case*, AIChE Journal, Vol. 50, No. 6, pp. 1211-1218, June 2004.

Marlin, T.J., *Process Control Designing Processes and Control Systems For Dynamic Performance*, First Edition, McGraw-Hill, Inc., 1995.

Nocedal, J. & Wright, S. J. *Numerical Optimization*, Springer-Verlag., 1999.

Press, W.H., Teukolsky, S.A., Vellerling, T.A. and Flannery B.P., *Numerical Recipes in C*, 2nd Edition, Cambridge University Press, 1995.

Rangaiah, G.P., *Evaluation of genetic algorithms and simulated annealing for phase equilibrium and stability problems*, Fluid Phase Equilibria, pp. 187–188, 2001.

Seppala, C.T. Harris, T.J. Bacon, D.W., *Time Series Methods for Dynamic Analysis of Multiple Controlled Variables*, Journal of Process Control, Vol.12, pp. 257-276, Feb. 2002.

Shinskey, F. G., *Process Control systems*, Third Edition, McGraw-Hill, Inc.

Stanfelj, N., Marlin T.E., and MacGregor J.F., *Monitoring and Diagnosing process Control performance: The Single Loop Case*, American Chemical Society, pp. 301-314, 1993.

Tham, M.T., *Minimum Variance And Generalized Minimum Variance Control Algorithms*, Department of Chemical and Process Engineering, University of Newcastle upon Tyne, pp. 1-18, 1999.

Upreti, S.R., *A New Robust Technique for Optimal Control of Chemical Engineering Processes*. Comp. Chem. Eng., 28 (8), pp. 1325-1336, 2004.

Watanabe, K., *Adaptive Estimation And Control*, Prentice hall, 1952.

Ypma, T.J., *Historical development of the Newton-Raphson method*, SIAM Review 37 (4), pp. 531–551, 1995.

Appendix

A-1 Basic Concepts and Definition

PID It is the most common control method in process control. It is a continuous feedback loop that maintains the process sinuous normally by taking corrective action whenever there is any deviation from the desired value ("setpoint") of the process variable (rate of flow, temperature, voltage, etc.). An "error" occurs when an operator manually changes the setpoint or when an event (valve opened, closed, etc.) or a disturbance changes the load, thus causing a change in the process variable.

The PID controller receives signals from sensors and computes corrective action to the actuators from a computation based on the error (proportional), the sum of all previous errors (integral) and the rate of change of the error (derivative).

P Proportional Gain

I Reset rate repeats / seconds

D Derivative time, seconds

Controller Tuning There are several methods for tuning a PID loop. The choice of method will depend largely on whether the loop can be taken "offline" for tuning, and the response time of the system. If the system can be taken offline, the best tuning method often involves subjecting the system to a step change in input, measuring the output as a function of time, and using this response to determine the control parameters.

If the system must remain online, one tuning method is to first set the I and D values to zero. Increase the P until the output of the loop oscillates. Then increase I until oscillation stops. Finally, increase D until the loop is acceptably quick to reach its setpoint. The best PID loop tuning usually overshoots slightly to reach the set point more quickly, however some systems cannot accept overshoot.

Effects of changes "increase" in parameters				
Parameter	**Rise Time**	**Overshoot**	**Settling Time**	**S.S. Error**
P	Decrease	Increase	Small Change	Decrease
I	Decrease	Increase	Increase	Eliminate
D	Small Change	Decrease	Decrease	Small Change

Table A1.1 Controller Tuning Rules

Process Model Transfer Function It is a mathematical representation of the relation between the input and output of a linear time-invariant system. The transfer function is commonly used in the analysis of single-input single-output (SISO) relationship. It is mainly used in linear, time-invariant system theory, signal processing, communication theory, and control theory.

In its simplest form for continuous-time signals, the function is often written as

$$H(s) = \frac{Y(s)}{X(s)} \tag{1.1}$$

where H(s) is the symbol for the transfer function, Y(s) is the output function, and X(s) is the input function. In discrete-time systems, the function is similarly written as H(z) = Y(z) / X(z).

Process White Noise It is a random signal with a flat power spectral density. In other words, the signal's power spectral density has equal power in any band, at any centre frequency, having a given bandwidth.

An infinite-bandwidth white noise signal is purely a theoretical construct. By having power at all frequencies, the total power of such a signal is infinite. In practice, a signal can be "white" with a flat spectrum over a defined frequency band.

Process Optimization It is the problem of determining the inputs of a function that minimizes or maximizes its value. Sometimes constraints are imposed on the values that the inputs can take; this problem is known as constrained optimization.

Genetic Algorithms A genetic algorithm (GA) is a method used to find approximate solutions to difficult-to-solve optimization problems through application of the principles of evolutionary biology to computer science. Genetic algorithms use biologically-derived techniques such as inheritance, mutation, natural selection, and recombination (or crossover). Genetic algorithms are a particular class of evolutionary algorithms.

Genetic Operator It is a process used in genetic algorithms to maintain genetic diversity. Genetic variation is a necessity for the process of evolution. Genetic operators used in genetic algorithms are analogous to those which occur in the natural world: survival of the fittest, or selection; asexual or sexual reproduction (crossover, or recombination); and mutation.

Jacobian matrix The Jacobian matrix is the matrix of all first-order partial derivatives of a vector-valued function. Its importance lies in the fact that it represents the best linear approximation to a differentiable function near a given point. In this sense, the Jacobian is akin to a derivative of a multivariate function. It is widely used in optimization as it refers to a local minimum or maximum when its determinant is equal to zero.

Newton's Method It is a well-known algorithm for finding roots of equations in one or more dimensions. It can be used to find local maxima and local minima of functions by noticing that if a real number x^* is a stationary point of a function $f(x)$, then x^* is a root of the derivative $f'(x)$, and therefore one can apply Newton's method to $f'(x)$.