DESIGN AND IMPLEMENTATION OF
MODEL BASED CONTROLLER
FOR
SYSTEMS
WITH INPUT SATURATION AND MODEL MISMATCH

BY

JOSEPH P.M. WONG, B.ENG, M.ENG

A Thesis submitted to the Faculty of Graduate Studies
in Partial Fulfillment of the Requirements
for the degree of Doctor of Philosophy

Department of Chemical Engineering
McMaster University, Hamilton, Ontario

MARCH 1992

DESIGN AND IMPLEMENTATION OF
MODEL BASED CONTROLLER
FOR
SYSTEMS
WITH INPUT SATURATION AND MODEL MISMATCH

BY

JOSEPH P.M. WONG

MARCH 1992

DOCTOR OF PHILOSOPHY (1991)
(CHEMICAL ENGINEERING)

McMASTER UNIVERSITY
HAMILTON, ONTARIO

TITLE: DESIGN AND IMPLEMENTATION OF MODEL BASED CONTROLLER FOR SYSTEMS WITH INPUT SATURATION AND MODEL MISMATCH

AUTHOR: JOSEPH P.M. WONG, B.ENG. (McGILL UNIVERSITY, QUEBEC)
M.ENG. (McMASTER UNIVERSITY, ONTARIO)

SUPERVISORS: DR. P.A. TAYLOR
DR. J.D. WRIGHT

NUMBER OF PAGES: xiii,254 pages

ACKNOWLEDGEMENT

The author wishes to express his gratitude to his research supervisors: Dr. P.A. Taylor and Dr. J.D. Wright for their guidance throughout the research work. My thanks also go to these long-time friends - Leo, Dave, Sateesh, Alan, Rob, Dora, for their friendship over the past years. I would also like to thank Dr. Taylor for his invaluable inputs, patience and encouragement during the final stage of this thesis. I am very grateful to the financial support provided by McMaster University. Finally I would like to thank Julia for her constant support during this write up. To my parents, my brother and sisters, I would like to say "Thanks for everything".

ABSTRACT

This research deals primarily with the design and implementation aspects of model based digital control algorithms. The performance of model based digital controllers degrades whenever the process input saturates or the process model is significantly different from the true process. Simulations and experiments have been performed to identify the saturation problems in model based controllers and four algorithms were proposed to compensate for this problem. The saturation problem was generalized to cover all situations where the output computed by the model based controller cannot be applied to the actual process. This includes actuator saturation, operator override, temporary actuator failure and asynchronous control. A generalized implementation scheme of model based controllers was proposed and a dramatic performance improvement was observed by using such an implementation scheme. The second part of the thesis focuses on the robustness aspect of model based controllers. The work reported here takes a time domain, rather than the traditional frequency domain approach. A new stability theorem and corollaries for uncertain systems have been derived and applied to simulated processes. A new analysis method, Error Band Method (EBM), has been developed to analyse the performance of a dynamic uncertain control system. Monte Carlo simulations have been performed to verify the validity of this approach. The method was extended to include controller design and controller comparisons. This methodology provides a different perspective to the analysis and design of uncertain model based digital control system.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

CMV	:	Constrained Minimum Variance
Conv	:	Conventional
EBM	:	Error Band Method
DCLR	:	Desired Closed Loop Response
DCS	:	Distributed Control System
Heur	:	Heuristic Algorithm
IAE	:	Integral Absolute Error
Imp	:	Implementation
LQG	:	Linear Quadratic Gaussian
MVC	:	Minimum Variance Control
MIMO	:	Multi-Input Multi-Output
NAR	:	New Anti-Reset Windup Algorithm
PI	:	Proportional and Integral Controller
PID	:	Proportional Integral and Derivative Controller
QIMC	:	Quadratic Internal Model Control
OS	:	Optimal Switch
SISO	:	Single-Input Single-Output

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

During the past decade computer process control has been gradually accepted by the chemical process industries. Today, most operators rely heavily on the computer. They very often cannot even run the process without it. Computer control has become an indispensable tool which enables the operators to identify a more profitable operating region and move the process to run at that region.

With the advances in computer hardware and operating systems, the computer control algorithm has changed from being heuristic based to model based. Heuristic algorithms such as PID (Proportional, Integral and Derivative) controller have been the main control algorithms for over four decades in the process industry. The only way to make the control system stable is to detune the PID controller. This is due to the limited structure in the PID algorithm which cannot provide adequate dead time compensation. Although detuning can make the control system stable, the result is very often poor performance. In order to improve the controller performance, one option is to make use of a process model. Digital controllers derived from a process model are called model based digital controllers. If the process model is correct and the implementation is proper, one should be able to achieve closed loop performance which corresponds exactly to the controller design specification and this would be the ideal situation. However, when it comes to practical applications, the controller performance is sometimes quite different from the designed specification (see Kusuma et al. (1984)). This thesis addresses some of the problems associated with these circumstances.

In many model based controller design algorithms it is assumed that all of the control actions prescribed by the controller can be implemented. In fact, this assumption is often invalid. For example, when a valve is fully open (saturation), any increase in the prescribed control action will not be possible. Also, an operator override or supervisory control override on the control action would be equivalent to not implementing the prescribed control action. These situations are not rare occasions. In fact, they happen quite often during normal process operation. The effects of saturation or override are seldom addressed in either the controller design or its implementation. A second issue surrounding the model based controller is the quality of the process model, particularly in the processing industry where the process is either poorly known or is too complicated to model except by a simplified model. The discrepancy between the actual process and the simplified model can cause the process performance to deviate from the designed performance. The difficulty in the model mismatch problem is the assessment of the effects of the mismatch between the actual process and the process model.

There are tremendous economic incentives to solve the implementation and model mismatch problems of model based controllers. Accurate implementation can drive the system to operate closer to the designed performance even in face of saturation or override. Proper handling of model mismatch in controller design can increase the time during which the process operates within specification and decrease the amount of product produced outside its specification. Higher production within specification means higher Return of Investment. In the processing industries, these are the key factors in maintaining a competitive edge.

1.2 IMPLEMENTATION OF MODEL BASED DIGITAL CONTROLLER

A model based digital controller is an algorithm designed with the aid of a process model. These controllers can be a piece of computer code written in the host computer or a device function in a DCS (Distributed Computer System). The common basis of these controllers is the use of a process model. Usually a linear first order plus deadtime model is used. The procedure is to take the process model and a design criteria (desired closed loop time constant, desired variance, desired bandwidth, etc.) through some mathematical operations and produce a controller in the form of a difference equation. The implementation of the difference equation is straight forward. However, what is not so obvious is how to handle saturation, operator override, or supervisory control override. The conventional approach is simply to take the control actions which were actually implemented and to feed them back to the difference equation for the next calculation cycle (see Smith (1972), Luyben (1973) and Astrom et al. (1984)). However, this is not the best way to handle these situations. The reasons for this problem and its solutions are given in Chapters 2 and 3. By properly handling the difference equation, one can provide better performance during saturation and override than the conventional way, provided that the process model is adequate.

1.3 MODEL MISMATCH PROBLEM

The model mismatch problem results from using a simplified process model in the designing of the digital controller. Sometimes, one can use a simple linear model to describe the entire behavior of the process within the operating region. But for non-linear processes such as a catalytic reactor, high purity distillation column, treating tank pH, etc, a simple linear model is definitely not adequate. Usually one has to lower the desired controller performance ("detune") of the controller in order to stabilize the control system. If one detunes the controller too much, the controller would not respond to any servo commands or regulatory disturbances. Hence, there should be a balance point where the control system is stable and still responsive to external disturbances. This thesis focuses on a methodology to find this balance point so that the resulting controller can perform well even with a simple process model.

1.4 OBJECTIVE

Given that model based digital control algorithms improve control performance in general, the objective of this thesis is to investigate the issues related to the implementation of model based digital algorithms. Two areas have been identified - saturation and model mismatch. Saturation is a practical issue which causes lost performance (see Gallun et al. (1985), Doyle et al. (1987), Campo et al (1990)). There has been no research investigating why some model based controllers lose performance while some (DMC, IMC, etc.) do not. This thesis provides an answer to this issue. The model mismatch problem is an active research area. Currently, there is no consensus as to the solution. The frequency domain approach initiated in Doyle et al. (1981), Postlethwaithe et al. (1981), Cruz et al. (1981) and Safonov et al. (1981) is very popular, but the investigation is still far from being a useful solution to the problem. This thesis offers a different perspective by approaching the problem from the time domain. The Error Band Method (EBM) has been developed in this work. This is by no means a complete answer to the model mismatch problem. However, by providing a different perspective to the problem and identifying the opportunities with the time domain approach, this work can contribute to a general solution in the future.

1.5 THESIS OUTLINE

Chapter 2 focuses on saturation protection for model based digital algorithms. Four new saturation protection algorithms have been proposed and are described in section 2.2. These algorithms were tested against simulations and a stirred tank pilot plant process. The results are collected in section 2.4. The conclusions for the proposed algorithms are described in section 2.5. Chapter 3 investigates the implementation of model based digital controllers. The inadequacies of the conventional algorithms are discussed, the causes of the inadequacies are analyzed and remeaies are proposed. These are contained in section 3.2. The proposed remedies are extended to MIMO systems. Simulations were performed and the results are discussed in section 3.3. The link between rational and irrational model based digital controllers is explained in section 3.4 together with a conclusion. The model mismatch investigation is covered in chapters 4 and 5. Chapter 4 focuses on the stability aspects of the model mismatch problem. A theorem to test the stability of the uncertain system has been developed and the derivation is described in section 4.2 together with some corollaries of the theorem. The performance aspects of the model mismatch problem are discussed in chapter 5. A novel approach, the Error Band Method (EBM) was developed to measure the performance of an uncertain control system. The rationale and the procedures of the EBM are explained in section 5.3. A step-by-step example is given in section 5.4 to illustrate the procedure of using the EBM to evaluate the performance of a model mismatch system. Section 5.5 exemplifies two applications of EBM - tuning and comparison of digital controllers. Chapter 6 provides a summary and conclusion of this thesis. There are still many areas, particularly in the model mismatch area, which can be further investigated. These research opportunities are identified and described in section 6.2.

Chapter 2

Saturation Protection of Model Based Digital Algorithm

2.1 Introduction

Although the concept of digital control has been around for several decades, the processing industry has been slow to incorporate this technology. With the dramatic increase in hardware and software reliability over the past decade, digital controllers are replacing the old analog controllers. Usually these digital controllers are implemented in a distributed computing environment. The system is usually referred to as a Distributed Control System (DCS). The distributed layout increases system modularity so that redundancy can be incorporated to enhance the system reliability. Inside these distributed computers are preprogrammed functions such as filtering, summation, multiplication, ratioing, PID control, Polynomial digital control, etc. Some systems are flexible enough to allow BASIC or C language programming for more advanced applications. All these tools aim at improving the quality of process control. Unfortunately, in many commercial or even custom-built control systems, one aspect of implementing these digital controllers has been overlooked - Saturation.

Saturation occurs when the final control element, the physical device acting on the manipulated variable, is temporarily unable to respond to the signal from the controller. For the three term controller (PID), many anti-reset windup algorithms have been used to handle the saturation problem (Gallun et al. (1985), Astrom et al. 1984, Luyben (1973), Smith 1972). The typical strategy is to use conditional integration; i.e., integrate the error only when it is appropriate. When the PID controller is expressed explicitly in Proportional, Integral and Derivative terms, it is easy to implement the conditional integration algorithm. However, these terms are usually collapsed together in model based digital controllers and saturation protection is often implemented as a simple clamping of the controller output. By way of an example, Figure 2.1 shows the simulation of a DAHLIN controller (Dahlin (1968)) on a stirred tank process with a simultaneous failure of the final control element for 10 control periods and a setpoint change. Curve (ii) is the response when the conventional clamping algorithm is used for saturation protection, while curve (i) is the response without saturation protection. Although the control signals in curve (i) were greater than 10 volts for some periods, the final control element saturated at 10 volts and control signals greater than this value were of no use in achieving the desired closed loop performance. This figure shows the need for a saturation protection algorithm to prevent windup. However, the consequence of using conventional clamping is that the optimality of the model based digital controller may be lost. This is illustrated in the experiment shown in Figure 2-2a. By comparing the three curves in Figure 2-2a: curve (i) represents the desired closed loop response (DCLR), curve (ii) the closed loop responses produced with the best saturation algorithm and curve (iii) from the conventional clamping method, one can see that the conventional clamping method results in a sluggish output response. One can conclude that two aspects should be considered in dealing with the saturation problem;

controller windup and closed loop optimality. The conventional clamping algorithm solves only the windup problem.

This chapter investigates some of the problems caused by implementing model based digital controllers in applications where the manipulated variable saturates; it then proposes and evaluates four solutions to these problems. The reset windup problem in model based digital controllers can be solved by employing either anti-reset windup or simple clamping methods. However, simulation and experimental results show that these methods produce a much slower process response than the algorithms proposed in this chapter. Four saturation algorithms were studied, ranging from an improved anti-reset windup method to an on-line constrained optimization, with the aim of improving the process response during periods of manipulated variable saturation. The simulations and experiments were performed on a pilot-scale stirred-tank process involving dead-time. These simulations and experiments were designed to show not only the improvement in the closed loop performance by incorporating these saturation algorithms into the model based digital control algorithm, but also the robustness of these saturation algorithms to the severity of the control problem and to model mismatch. They show that it is worthwhile to include these saturation algorithms in model based digital controller designs.

The four proposed saturation algorithms are introduced in the next section. The rationales behind each algorithm are discussed. These algorithms are verified by simulations and experiments of a pilot-scale stirred-tank process. Details of the process are presented in section

2.3. The simulation and the experimental results are described and discussed in section 2.4. A conclusion follows in section 2.5. This chapter deals mainly with SISO systems. The treatment of saturation problem in MIMO systems are discussed in Chapter 3.

2.2 SATURATION ALGORITHMS

Four saturation algorithms are presented in this sections. They differ in the extent to which they make use of process model information. Digital controllers which are designed using a process model and a performance criterion are referred to here as model based digital controllers. Since model based digital controllers already make use of a process model in the controller design equations, the accompanying saturation algorithm can also make use of this information. These algorithms are designed to solve the saturation problem for single input single output systems (SISO) and are arranged in the sequence of increasing computational efforts. A heuristic algorithm represents the idea of conserving the control action prescribed by the model based digital controller. A new anti-reset algorithm freezes the integration of the output error only when the saturation is a result of the integral action. An optimal switch algorithm makes use of the process model to explicitly calculate the optimal period for the control action to stay at the saturation limit. The last algorithm, QIMC, treats the saturation problem as a constrained optimization problem. It will provide the ultimate performance for a system with saturation, but will require the greatest computation.

The DAHLIN digital controller (Dahlin (1968)) was chosen as an example of a model based digital controller. A desired closed loop trajectory is specified in the design of this controller but the actual closed loop trajectory will not follow the desired trajectory when saturation occurs. Therefore, the sum of squared deviations of the actual closed loop trajectory from this desired trajectory can be used as a measure of the effectiveness of the saturation algorithm (see equation 2-1).

$$\sum_{j=i+1+f}^{T'+i+f} \left[Y^{CL}_{ACTUAL}(j) - Y^{CL}_{DESIRED}(j) \right]^2 \qquad (2\text{-}1)$$

where T′ is chosen as six time open loop constants (6τ)
i is the current discrete control interval
f is the number of actual periods of delay in the process model
CL *stands for closed loop*

2.2.1 Conventional Saturation Protection

Many papers have proposed solutions to the windup problem, (Gallun et al.(1985), Astrom et al. (1984), Smith (1972) and Khandria et al.(1976)). These papers focus on the saturation problem in PID controllers. The conventional solution is to freeze the integration of the output error whenever the control action saturates. This is easy to implement in the case of a PID controller because the integration term is explicitly expressed in the controller. However, model based digital controller designs usually collapse the Proportional, Integral and Derivative terms together as polynomials in $\nabla U(i)$ and $E(i)$. The velocity form of a general digital controller is:

$$\begin{aligned}\nabla U(i) = {} & \beta(1)*\nabla U(i-1)+\beta(2)*\nabla U(i-2)+\ldots+\beta(Nb)*\nabla U(i-Nb) \\ & -\alpha(0)*E(i) \quad -\alpha(1)*E(i-1) -\ldots-\alpha(Na)*E(i-Na)\end{aligned} \qquad (2\text{-}2)$$

where the α(z) and β(z) polynomials are the results from Dahlin Control design; Wright et al. (1988)

Harris et al.(1982) show that equation 2-2 can be separated into terms which represent Proportional, Integral, 1st and higher order Derivatives and Dead-time compensation contributions. The velocity form of the general model based digital controller when *Na*=4 is:

$$\begin{aligned}\nabla U(i) = {} & \beta(1)*\nabla U(i-1)+\beta(2)*\nabla U(i-2)+\ldots+\beta(Nb)*\nabla U(i-Nb) \\ & +K1*[E(i)-E(i-1)] \\ & +K2*E(i) \\ & +K3*[E(i)-2E(i-1)+E(i-2)] \\ & +K4*[E(i)-3E(i-1)+3E(i-2)-E(i-3)] \\ & +K5*[E(i)-4E(i-1)+6E(i-2)-4E(i-3)+E(i-4)]\end{aligned} \qquad (2\text{-}3)$$

where the *K* parameters representing the Proportional, Integral and Derivative gains can be related to the α parameters in Equation 2-4 by

$$\begin{aligned}K5 &= -\alpha(4) \ ; \\ K4 &= -4.0*K5 + \alpha(3) \ ; \\ K3 &= -\alpha(2) - 3.0*K4 - 6.0*K5 \ ; \\ K1 &= \alpha(1) - 2.0*K3 - 3.0*K4 - 4.0*K5 \ ; \\ K2 &= -\alpha(0) - K1 - K3 - K4 - K5 \ ;\end{aligned} \qquad (2\text{-}4)$$

and the β parameters represent the Dead-time compensation terms. The analogous Positional Form of the general model based digital controller with Na=4 is:

$$\begin{aligned} U(i) = {} & Uss+\beta(1)*U(i-1)+\beta(2)*U(i-2)+\ldots+\beta(Nb)*U(i-Nb) \\ & +K1*E(i) \\ & +K2* \sum E(i) \\ & +K3*[E(i)-E(i-1)] \\ & +K4*[E(i)-2E(i-1)+E(i-2)] \\ & +K5*[E(i)-3E(i-1)+3E(i-2)-E(i-3)] \end{aligned} \tag{2-5}$$

where *Uss* is the steady state value of the controller output.

Once the model based controller has been rearranged into Proportional, Integral etc. terms, then anti-reset windup can be applied. During saturation, the conventional anti-reset windup algorithm would freeze the summation term in the positional form, equation 2-5, thus stopping the integration. This algorithm can be implemented on digital computer as the following:

```
C AFTER CALCULATING U FROM EQUATION 2-5 WITH E AS THE CURRENT ERROR
C SUM IS THE SUMMATION TERM  K2*Σ E(I)
                IF (U.GT.UMAX) THEN
                        U = UMAX
C       STOP SUMMATION
                        SUM = SUM - K2 * E
                ELSEIF (U.LT.UMIN) THEN
                        U = UMIN
C       STOP SUMMATION
                        SUM = SUM - K2 * E
                ENDIF                                            (2-6)
```

When the velocity form, equation 2-3, of the model based digital controller is used in applications where the actuator requires an absolute signal rather than an incremental one, the controller output U(i) is calculated as:

$$
\begin{aligned}
U(i) = U(i-1)& \\
&+\beta(1)*\nabla U(i-1)+\beta(2)*\nabla U(i-2)+\ldots+\beta(Nb)*\nabla U(i-Nb) \\
&+K1*[E(i)-E(i-1)] + K2*E(i) \\
&+K3*[E(i)-2E(i-1)+E(i-2)] \\
&+K4*[E(i)-3E(i-1)+3E(i-2)-E(i-3)] \\
&+K5*[E(i)-4E(i-1)+6E(i-2)-4E(i-3)+E(i-4)]
\end{aligned}
\qquad (2-7)
$$

and the conventional anti-reset windup algorithm is usually implemented as a simple clamping:

```
C       AFTER CALCULATING U FROM EQUATION 2-7
                IF (U.GT.UMAX) THEN
                        U = UMAX
                ELSEIF (U.LT.UMIN) THEN
                        U = UMIN
                ENDIF                                               (2-8)
```

This combination of a conventional clamping algorithm and a velocity form controller, equations 2-7 and 2-8, is not always equivalent to the anti-reset windup algorithm used for the positional form, equations 2-5 and 2-6. Writing an expression for *U(i-1)* by shifting equation 2-7 backwards by one control interval and using this to substitute for *U(i-1)* in equation 2-7, leads to equation 2-5 upon repeated substitution for *U(i-2), U(i-3) ... U(i-N)*. Segall and Taylor (1986) show that this velocity form will be equivalent to the positional form provided that no terms are discarded during the substitution process. However, the clamping of the control action with the conventional clamping algorithm, equation 2-8, under a velocity form controller simply discards terms; thus the velocity algorithm is no longer equivalent to the positional algorithm. Since output errors are being discarded during saturation, the apparent error seen by the model based digital controller is smaller than the actual error. As a result, the clamping algorithm prevents windup due to error accumulation during saturation, but the process response becomes sluggish. Under these conditions, the process output does not follow the response specified in the controller design procedure and the advantages of using a model based digital controller are lost. The purpose of the subsequent algorithms is to retain the windup protection capability while preserving the optimality of the model based controller.

2.2.2 New Anti-reset Algorithm

This saturation algorithm is due to Segall and Taylor (1986). The idea stems from the observed difference in the operation of the velocity and positional forms of a model based digital controller during saturation, equations 2-7 and 2-5. In order to minimize this difference, Segall and Taylor suggested that the reset windup algorithm should make a distinction between saturation due to integral action and saturation due to Proportional, Derivative and Dead-time compensation terms. The integration should be frozen only when the saturation is due to the integral action since error can only windup through the integral mode. This idea is similar to the technique used in Gallun et al. (1985). Although they focused mainly on PID controllers, their "control path" concept in implementation can be used in model based controllers to prevent windup in cascade and feedforward control configurations. A Fortran listing of the New Anti-reset algorithm applied to model based digital controllers is shown in APPENDIX A.1.

2.2.3 Heuristic Algorithm

In the conventional clamping algorithm, equation 2-8, the digital controller discards excess control action whenever the control action saturates; this results in a slow closed loop response. Intuitively, if one preserves the control action and at the same time avoids the windup problem, then one would expect to obtain a better performance. One way to do this is to augment the current control action with the excess control action from the last control interval. However, if the current control action also saturates, one has to discard the excess control action in order to prevent windup. Hence, only one past excess control action can be preserved. The FORTRAN-77 code for this Heuristic algorithm is listed in APPENDIX A.2. This is a totally intuitive approach and the resulting trajectory is usually underdamped rather than overdamped.

2.2.4 Optimal Switch Algorithm

If one observes the closed loop response of the conventional clamping algorithm in Figure 2-2a, one would notice that the control action comes off the saturation too early. If the control action were to stay at the limit a little longer, the performance would be improved. The Optimal Switch algorithm is based on this observation. The essence of this algorithm is to make use of the process model to determine how long the control action should stay at the saturation limit. When the model based controller prescribes a control action which is outside the physical limit of the manipulated variable, this saturation algorithm overrides the controller and calculates how long the control action should stay at the limit so that the performance criterion in equation 2-9 will be minimized. In effect, this is a dual control strategy in which the saturation algorithm takes over when saturation occurs and the linear digital controller is operative when the manipulated variable is not at one of the saturation limits.

When the manipulated variable saturates at the ith control interval, evaluate the performance index, I, over the future horizon, P, for a saturation period of k control intervals.

$$I(i+1+f;P,k)=\sum_{j=i+1+f}^{i+P+f} \{\, Y_{desired}(j) - Y_{predicted}(j) \,\}^2 \qquad (2\text{-}9)$$

$for\ k = 1,\ 2,\ \ldots.,\ P$

These calculations are repeated for different values of k until the minimum performance index is found. The future horizon was chosen to be six times the open loop time constant, $PT = 6\tau$, in order to allow the predicted output to achieve steady state. $Y_{desired}(j)$ are the elements of the desired closed loop trajectory which is specified during the design of the model based controller. In this study, a DAHLIN controller was used and so the desired trajectory was an overdamped exponential trajectory. $Y_{prediected}(j)$ are the elements of the predicted closed loop trajectory from the following convolution summation:

$$Y_{predicted}(j+f) = \sum_{l=0}^{N} h(l) * U(j-l) \qquad (2\text{-}10)$$

$\quad$ *for* $j=i+1, i+2, \ldots\ i+P$

where $h(l)$ are the process impulse weights of order N. The process inputs, $U(j-l)$, used in equation 2-10 are determined by three different mechanisms. For $(j-l) < i$, we can use the past process inputs since these correspond to the pre-saturation period. During the particular saturation period under consideration, k, the process input must be held at its limit; thus for $i \le (j-l) < i+k$,

$$U(j-l) = \text{saturation value of } U \qquad (2\text{-}11)$$

After the saturation period, we switch back to closed loop operation using the feedback controller to calculate the process input; thus for $i+k \le (j-l) < i+P$

$$U(j-1)=\sum_{m=1}^{Ng} g(m)*\left\{ Y_{desired}(j-1-m+1)-Y_{predicted}(j-1-m+1) \right\} \qquad (2\text{-}12)$$

where *g(m)* are the controller impulse weights of order *Ng.*

This algorithm is operative whenever the control action saturates and, in effect, performs a discrete simulation of the control system in order to search for the optimal saturation period. We used a direct search starting with 1 period of saturation. Equations 2-10, 2-11 and 2-12 are used to predict the output response and equation 2-9 is used to evaluate the performance index. After incrementing the saturation period, the calculations of the predicted output response and performance index are repeated. The saturation period continues to be incremented until the performance index starts to increase. This terminates the search and the optimal saturation period is the one associated with the lower performance index. Note that it is assumed that the objective function is a monotonic function of the saturation period. The experiences from simulations and experiments show that this is in general true for 1st order plus dead time systems. For guaranteed optimal solution, one can calculate the objective function I, in equation 2-9 for the entire horizon P and pick the optimal saturation period. The manipulated variable is then held at its saturation limit for the optimal saturation period before reactivating the feedback controller. Since the calculations involve only linear difference equations, the computation time is not crucial.

2.2.5 Constrained Quadratic Internal Model Control (QIMC)

The motivation for using QIMC comes from the desire to obtain the best possible response from a saturated system. One way to achieve this objective is to formulate the control problem as a constrained optimization. In doing this, QIMC becomes a constrained optimization algorithm rather than a control and saturation algorithm since it calculates the process inputs which will explicitly optimize the performance of the system while satisfying the constraints. The development of QIMC follows from the work of Richalet et al. (1978) and Martin (1981) on predictive, model based control and from the work of Garcia et al. (1981) on Internal Model Control (IMC). The framework proposed by Garcia et al. (1981) was used to define the process model in terms of its impulse weights and the past and future process inputs. This is incorporated into an on-line, constrained optimization procedure which minimizes a quadratic performance index. This differs from the Quadratic Dynamic Matrix Control (QDMC) proposed by Garcia et al. (1984) in the way in which the control problem is posed as an optimization problem and the way in which the on-line optimization with constraints is solved.

For a Single Input Single Output (SISO) process, the following section shows how to transform the problem of a saturating controller into an optimization problem. The objective is to minimize the sum of squared deviations between the predicted and desired output response while satisfying constraints on both the value and rate of change of the manipulated variable and on the value of the predicted process output. At the ith interval we have the measurement of the current process output, $Y_{measured}(i)$, and we want to compute the process input for the current interval, $U(i)$.

2.2.5.1 Process Output Prediction

Define a vector, $Y_{predicted}(i+1+f:i+P+f)$, containing a prediction of the process output from the $i+1+f$ up to the $i+P+f$ interval in terms of : a vector, $U_{future}(i:i+M-1)$, containing the sequence of process inputs from the ith up to the $i+M-1$ interval which will be computed so as to optimize the process performance; a vector, $U_{past}(i-N+1:i-1)$, containing a record of the process inputs from the i-N+1 up to the i-1 interval; a vector, $E_{predicted}(i+1+f:i+P+f)$, containing a prediction of the effects of model uncertainty and external disturbances from the $i+1+f$ up to the $i+P+f$ interval; and the matrices Ψ and Φ containing impulse weight information:

$$\begin{aligned} Y_{predicted}(i+1+f:i+P+f) = {} & \Psi * U_{future}(i:i+M-1) + \\ & + \Phi * U_{past}(i-N+1:i-1) + \\ & + E_{predicted}(i+1:i+P) \end{aligned} \qquad (2\text{-}13)$$

These vectors define the amount of process information available at each interval for the computation of those process inputs which will optimize the process performance. They define a subset of the total process information and so define a local optimization problem at each interval. We need to solve this sequence of local optimizations, rather than one global optimization, because there are unknown external disturbances and the model is not perfect. This is similar to the Moving Window Optimization concept. The forecast for the model uncertainty and external disturbances, $E_{predicted}(i+1+f:i+P+f)$, can be derived from either a stochastic noise model, if one is justified and available, or from a deterministic model. In this application, it was assumed that the process disturbance and model uncertainty were constant over the prediction horizon, P, and were uncorrelated with either past or future values. Therefore, all the elements of $E_{predicted}(i+1+f:i+P+f)$ were assumed to be equal to the value of the prediction error at the ith interval. Thus, the jth element was defined as:

$$E_{predicted}(j+f) = Y_{measured}(i) - Y_{predicted}(i) \qquad (2\text{-}14)$$

and

$$Y_{predicted}(j+f)=\sum_{l=0}^{N} h(l) * U(j-l) \qquad (2\text{-}15)$$

$$\text{for } j=i+1,\ i+2,\ \ldots.\ ,\ i+P$$

where $h(l)$ are the process impulse weights.

2.2.5.2 Optimization Procedure

For the prediction period, $i+1+f$ to $i+P+f$, the deviations of the predicted output from the desired trajectory are used to define an error vector which is to be minimized:

$$E_{trajectory}(i+1+f:i+P+f) = Y_{desired}(i+1+f:i+P+f) - Y_{predicted}(i+1+f:i+P+f) \tag{2-16}$$

In this work an exponential response for the desired output trajectory was chosen so that the results for QIMC could be compared to those from the DAHLIN controller. Then by substitution for $Y_{predicted}(i+1+f:i+P+f)$ from equation 2-13:

$$\begin{aligned} E_{trajectory}(i+1+f:i+P+f) = {} & Y_{desired}(i+1+f:i+P+f) \\ & - \Phi * U_{past}(i-N+1:i-1) \\ & - E_{Predicted}(i+1+f:i+P+f) \\ & - \Psi * U_{future}(i:i+M-1) \end{aligned} \tag{2-17}$$

and defining E_{RR} as those terms in equation 2-17 which are not functions of U_{future}

$$E_{RR} = Y_{desired}(i+1+f:i+P+f) - \Phi * U_{past}(i-N+1:i-1) \qquad (2\text{-}18)$$

$$- E_{predicted}(i+1+f:i+P+f)$$

the trajectory error can be expressed as

$$E_{trajectory}(i+1+f:i+P+f) = E_{RR} - \Psi * U_{future}(i:i+M-1) \qquad (2\text{-}19)$$

The optimization problem is to find the future control actions, U_{future} which will minimize the sum of squared deviations from the desired trajectory. Thus, the optimization problem can be written simply as:

$$\underset{U_{future}}{Min} \left\{ E^{T}_{trajectory} * E_{trajectory} \right\} = \qquad (2\text{-}20)$$

$$= \underset{U_{future}}{Min} \left\{ \left[E_{RR} - \Psi * U_{future} \right]^{T} * \left[E_{RR} - \Psi * U_{future} \right] \right\}$$

where $E^{T}_{trajectory}$ is the transpose of the vector. Upon expansion

$$\underset{U_{future}}{Min}\left\{E^{T}_{trajectory} * E_{trajectory}\right\} = \tag{2-21}$$

$$= \underset{U_{future}}{Min}\left\{ E^{T}_{RR}\, E_{RR} - 2* E^{T}_{RR} * \Psi * U_{future} + U_{future} * \Psi^{T} * \Psi * U_{future}\right\}$$

the scalar quantity $E^{T}_{RR} * E_{RR}$ is a constant which is independent of U_{future}. Therefore, the performance index to be minimized, I, can be formulated as:

$$\underset{U_{future}}{Min}\ I(i+1:i+P;M)=\underset{U_{future}}{Min}\left\{\Gamma*U_{future} + U^{T}_{future}*\Lambda*U_{future}\right\} \tag{2-22}$$

$$\Gamma = -2 * E^{T}_{RR} * \Psi$$

$$\Lambda = \Psi^{T} * \Psi$$

Subject to the constraints on the value of the manipulated variables:

$$U_{low}(i:i+M-1) \le U_{future}(i:i+M-1) \le U_{high}(i:i+M-1)$$

Subject to the rate of change constraints on the manipulated variables

$$| U_{future}(i:i+M-1)-U_{future}(i-1:i+M-2)| \leq U_{difference}(i:i+M-1)$$

Subject to the constraints on the value of the process output variables:

$$Y_{low}(i+1+f:i+P+f) \leq Y_{predicted}(i+1+f:i+P+f) \leq Y_{high}(i+1+f:i+P+f)$$

If constraints on the process outputs are specified, then equation 2-12 will have to be solved at each iteration of the optimization algorithm. For a linear process, Λ and Γ will be constant during the optimization iterations but E_{RR} and hence Γ will need to be evaluated each time the QIMC algorithm is used to compute a new sequence of process inputs. In this application of the QIMC algorithm to provide the optimal performance under saturation conditions; the positional constraints on the manipulated variables were determined from the physical limitations on the heater units, the rate of change constraints on the manipulated variables were set to $\pm\infty$ in all experiments, and the constraints on the process outputs were set to $\pm\infty$ in all the experiments. The constrained optimization problem, equations 2-14, 2-15, 2-17 and 2-22, can be solved by many algorithms. For this work, the constrained Hill algorithm due to Rosenbrock (1960) was chosen because of its simplicity, see Kuest (1973).

One of the key questions in solving optimization problems is the computation time. This is especially important in process control where the computation time has to be less than the control interval. From the form of the objective function, equation 2-22, it is clear that the computational effort will increase as one increases the number of future control actions over which the optimal solution is sought. Fortunately one can focus, in many instances, on the case: Output Horizon (P) equals to 1 and Input Horizon (M) equals to 1. This has been shown in Garcia et al. (1981) to be equivalent to a dead-beat controller, but a smoother output response can be obtained by reshaping the setpoint with a filter. For a system with 20 impulse weights, with both input and prediction horizon set to 1, the computation time is less than 1 second on a VAX 11/750 minicomputer without a floating point accelerator. In the case of a non-minimum phase system, the invertibility problem can be solved by increasing the prediction horizon (P).

Segall et al. (1985) derived an analytical solution to the saturation problem for the one-step ahead Minimum Variance controller of Clark and Hastings-James (1973). This is equivalent to the P=1 and M=1 case in QIMC with positional, but not rate of change, constraints. Segall et al. found that for a first order process, their saturation compensation algorithm possesses a structure similar to the Heuristic algorithm reported here; i.e., the last unimplemented control action is added to the current control action. Their algorithm takes a fraction of the past unimplemented control action instead of the full unimplemented control action used in the Heuristic algorithm. The fraction is determined by the process dynamics, the noise model, and the desired closed loop trajectory.

One advantage of QIMC algorithm is the natural extension to MIMO systems. The QIMC objective function in equation 2-20 can be easily modified to include MIMO systems. The general treatment of MIMO saturation problem are discussed in Chapter 3.3.

2.3 PROCESS DESCRIPTION

The process under investigation was a pilot-scale stirred tank process. A schematic diagram is shown in Figure 2-3. This work focuses on controlling the fluid temperature which, depending upon the application, was measured at the outlet of a 3 meter long pipe or at the outlet of the bottom tank. The inlet flow rate was measured by a rotameter and the nominal flow rate was 3.4 litre/minute. Two sets of manually controlled heaters were used to adjust the equilibrium tank temperature. Another set of heaters was used to implement the control action so that the tank outlet temperature could be controlled. A DEC (Digital Equipment Corporation) DPM23 equipped with the industrial I/O subsystem IPV12 served as the interface between the process equipment and a VAX 11/750. The DPM23 front end was connected to the host VAX 11/750 computer through a DECDataway network. A general application software package was written to enable the communication between the DPM23 and the 11/750 computers. The network supported six pieces of process equipment: a stirred tank process, an extractive distillation column, a solvent extraction column, a catalytic packed-bed reactor, a pH mixing tank and a polymerization reactor train. Details about the hardware and software of this network are contained in Wong (1983) and Segall (1983). All the control algorithms for this stirred-tank process were written in FORTRAN-77 on the host computer and they were incorporated into an interactive real-time control program which was capable of monitoring the process variables, activating different controllers, performing data logging and plotting historical data on-line.

Since a process model is required in order to design a model based digital controller, empirical models of the process were identified using step tests. It was found that the nominal process (Process 1) could be adequately described by a first order plus dead-time transfer function relating the the voltage signal going to the heater in the top tank to the temperature at the end of the dead time pipe:

$$G_p(s) = \frac{0.95\, e^{-45s}}{255s + 1} \tag{2-23}$$

Discretizing with a control interval of T = 45.0 second gave the pulse transfer function for Process 1 :

$$G_p(z^{-1}) = \frac{0.1537 z^{-2}}{1.0 - 0.8382\, z^{-1}} \tag{2-24}$$

In order to check if the proposed saturation algorithms are sensitive to the severity of the control problem, the process was modified so that the ratio of dead-time to time constant was greater than unity. Equations 2-25 and 2-26 represent respectively the continuous and discrete transfer functions of the modified process (Process 2). To modify the physical process, the overflow pipe shown in Figure 2-3 was shortened so that the tank volume and hence the process time constant were reduced and the extra dead-time was added by using a delay vector for the temperature measurement in the real-time computer control program.

$$G_p(s) = \frac{0.95\ e^{-200s}}{180s + 1} \qquad (2\text{-}25)$$

Discretizing with a control interval T = 40.0 second yielded the pulse transfer function for Process 2 :

$$G_p(z^{-1}) = \frac{0.1537z^{-6}}{1.0 - 0.8382\ z^{-1}} \qquad (2\text{-}26)$$

A 2nd order process (Process 3) was created by using the top tank heater to control the bottom tank temperature. Fitting a 1st order with dead time transfer function to the 2nd order relationship between the bottom tank temperature and the top heater voltage gave :

$$G_p(s) = \frac{0.9\ e^{-150s}}{200\ s + 1} \qquad (2\text{-}27)$$

In fact, this is a practice used in many industrial applications whereby the process is modeled by a low order transfer function. A simple process model results in a simplified controller design procedure. Discretizing equation 2-27 with a control interval of T=50.0 second would yield :

$$G_p(z^{-1}) = \frac{0.199\ z^{-4}}{1.0 - 0.7888\ z^{-1}} \qquad (2\text{-}28)$$

In order to illustrate the load disturbance problem, a computer-controlled heater unit in the bottom tank was used to act as a load disturbance to the control system. The bottom tank temperature was controlled by the top heater unit as in the case of Process 3, and the process transfer function was described by equation 2-27. A load disturbance gave a transfer function between the bottom tank temperature and bottom heater voltage of :

$$G_p(s) = \frac{0.98}{110\ s + 1} \qquad (2\text{-}29)$$

The model mismatch problem was tested by reducing the inlet water flow rate to the two-tank system to 40% of full-scale. Empirical step tests showed that the transfer function between the tank top heater and the bottom tank temperature could be adequately described by equations

2-30 and 2-31 for the up and down dynamics respectively.

$$G_p(s) = \frac{2.1\ e^{-150s}}{750\ s + 1} \tag{2-30}$$

$$G_p(s) = \frac{1.9\ e^{-250s}}{700\ s + 1} \tag{2-31}$$

At this low flow rate, the process shows asymmetric dynamics. Notice that the process gain and the process time constant differ by about 100% and 250% respectively from those in equation 2-27. The DAHLIN and QIMC controllers, which were designed using equation 2-27, were applied to this process in order to observe the model mismatch effect on the saturation algorithms.

2.4 EXPERIMENTAL APPLICATION OF THE SATURATION ALGORITHMS

Before any experiments were run, simulations were performed to investigate the proposed saturation algorithms. The performance indices from these simulations appear in Table 2-1 and in the bar charts in Figures 2-4 and 2-6. The simulations show that the proposed saturation algorithms provide an improvement over the conventional clamping algorithm. In order to verify that the improvement can indeed be realized in practice, experiments were performed on a physical stirred tank process. The experimental results are shown in Figures 2-2 and 2-5 with the corresponding performance indices shown in Table 2-1 and on the bar charts in Figures 2-4 and 2-6.

Figures 2-2a and 2-2b show the experimental application of the different saturation algorithms to the stirred tank process configured as Process 1. Curve (i) is the desired closed loop response for which the Dahlin controller was designed using Equation 2-24 as the model of the process. The desired closed loop time constant is chosen to be 1/4 of the open loop time constant; i.e., the closed loop time constant was 64 seconds. Curve (ii) is the control system under QIMC control. For this run there is no constraint on the rate of change of the manipulated variable and the desired set point trajectory is equal to that of DAHLIN controller. An exponential low-pass filter with time constant 64 seconds was installed in the feedback path to reduce the noise in the temperature measurement. These experiments demonstrate that the conventional clamping algorithm with model based digital controllers is inadequate. QIMC, curve (ii), provides the best response (in a least squares sense) and the conventional clamping approach gives the worst response. Curves (iv), (v) and (vi) correspond to DAHLIN control with the Heuristic, New anti-reset and Optimal switch algorithms respectively. It is interesting to note that all the proposed algorithms perform better than the conventional clamping method. The corresponding performance indices (sum of squared deviations from desired trajectory) for Process 1 are listed in Table 2-1 and illustrated in Figure 2-4.

In order to demonstrate that the effectiveness of the proposed saturation algorithms is not significantly affected by the severity of the control problem, the process was deliberately modified so as to increase the dead-time to time constant ratio. The modifications changed the dead-time from 45 to 200 seconds and the time constant from 255 to 180 seconds; this became Process 2. The appropriate process model, equation 2-26, was used to re-design the DAHLIN and QIMC controllers. The feedback filter for the QIMC controller was again used in this experiment. The Performance Indices for this process are shown in Figure 2-6 and listed in Table 2-1. Results of the experiments are plotted on Figures 2-5. The experiments on Process 1 and Process 2 agree very well with the simulation results. The conventional clamping algorithm was consistently the worst saturation algorithm in both the simulations and experiments. The values of the performance indices in Table 2-1 for simulation runs differ slightly from the experimental ones because of measurement noise and inevitable model mismatch effects. The temperature measurement had a precision of ±0.15°C which affected the calculated control action and hence the output performance. Furthermore, the performance index obtained from the simulations assumed that the model used in the controller design was an accurate description of the actual process; this was not always true in the experimental runs. The mismatch between the model used in controller design and the actual process also affected the closed loop output performance and hence the performance index. However, the simulation and experimental results concur that QIMC, Optimal Switch and the New anti-reset algorithm show improvement over the Heuristic algorithm and that all the proposed saturation algorithms show considerable improvement over the conventional clamping algorithm.

2.4.1 Regulatory Control with Saturation Algorithms

The DAHLIN and QIMC controllers used in the previous experiments were designed as servo controllers. It is possible to re-design them for disturbance regulation when the type of disturbance is specified (Mosler et al. (1967) and Smith (1972)). Since explicit model information is used in the Optimal Switch and QIMC saturation algorithms, a model of the disturbance would be needed in order to achieve optimal regulation during periods of saturation. In particular, equations 2-12 and 2-14 would need to be modified to include the effect of the disturbance on the predicted output. However, the purpose here is not to investigate the design of CAHLIN and QIMC controllers for regulation but to demonstrate that the proposed saturation algorithms do not experience numerical instabilities when the process is subjected to load disturbances rather than set point changes.

The disturbance regulation experiments were performed on Process 3. This is actually 2nd order between the bottom tank temperature and the top tank heater voltage but was modeled as 1st order with dead time. The DAHLIN and QIMC controllers were designed using equation 2-26 for setpoint changes with a closed loop tine constant of 50 seconds. A low pass filter with a time constant of 50 seconds was also used in the feedback path of the QIMC controller to reduce measurement noise. The disturbance was a step change in the bottom tank heater voltage from 5 to 1 volt. This affects the bottom tank temperature in a purely 1st order manner as seen in the load transfer function in equation 2-29.

The experimental results from Process 3 are shown in Figures 2-7a and 2-7b. Since the controllers were designed to produce an exponential closed loop response to a step change in the setpoint, the closed loop desired response during a disturbance is a straight line (since there is no setpoint change). This is a very ambitious control objective given that the manipulated variable experiences significant dead time as shown in, equation 2-27, whereas the load variable has a straight forward 1st order affect on the bottom tank temperature, equation 2-29. This, of course, produces severe control actions during the disturbance and causes the manipulated variable to saturate. This tests the saturation algorithms during a disturbance but is not the best way of designing controllers for load regulation. The results show that all the saturation algorithms handled the disturbance without numerical problems.

2.4.2 Model Mismatch Problems with Saturation Algorithms

Model mismatch is a very crucial problem in model-based controllers and many research papers (Palmor et al. (1979), Marshall (1979), Doyle et al. (1981), Lewin et al. (1991), etc.) have discussed this problem. The purpose in this section is not to present a solution to the model mismatch problem but to illustrate this problem with respect to saturation algorithms.

2.4.2.1 Structural Model Mismatch (Process 3)

In this experiment, a process which is physically 2nd order, Process 3, was modeled by the 1st order with dead-time transfer function given in equation 2-28. This low order, approximate model was used in the controller design equations for the DAHLIN and QIMC and in the saturation algorithms and thereby produced controllers which had the wrong structure for this particular process; the controller did not have enough poles and zeros to achieve the desired closed loop response because it was designed using a simplified model. The closed loop time constant was 50 seconds and the QIMC had a feedback filter with a time constant of 50 seconds. The experimental responses after a 3°C set point change are shown in Figures 2-8a and 2-8b and the performance indices are listed in Table 2-1 and illustrated in the bar charts in Figure 2-9. The results show no significant differences between this run and the previous runs even though there is inherent structural mismatch between the real process and the model used in the controller design equations. The fitting of a 1st order plus dead-time model structure to experimental data is usual practice in many industrial applications and this experiment demonstrates that the improvement in performance due to the saturation algorithms can be realized provided that the simplified process model adequately describes the dominant behavior of the real process.

2.4.2.2 Gross Parameter Mismatch (Process 4)

In this experiment the controllers were again designed using equation 2-26 but the inlet water flow rate to the real process was reduced to 40% of full scale. This mimiced a throughput change in an industrial process where the process parameters change by a significant amount. The controllers were not informed of the change in operating point and so there was a gross mismatch between the parameters of the model used in the controller design and the true process model. The process actually had asymmetric dynamics and a comparison of equations 2-30 and 2-31 with the nominal process model, equation 2-27, shows mismatch in all three parameters.

The experimental responses to a 3°C change in setpoint are shown in Figures 2-10a,b. In order to determine whether or not the oscillations were due to model mismatch in the controller design or to mismatch in the saturation algorithms, a further experiment was performed. Instead of the 3°C setpoint change, which caused saturation, a smaller setpoint change (1°C) was designed so that saturation would not occur. If oscillations were generated by this smaller setpoint change then they must come from the mismatched controller and not from the saturation algorithm. Results for the DAHLIN controller for both the 3°C and 1°C setpoint change are shown in Figure 2-11a and the corresponding results for QIMC in Figure 2-11b. These experiments suggest that the oscillation is caused by model mismatch in the controller and not by mismatch in the saturation algorithm.

The simulation and experimental results indicate that QIMC produces more oscillations than the DAHLIN controller under model mismatch conditions. This is due to the fact that QIMC uses more model information than the DAHLIN controller. This result corroborates the observation by Kestenbaum et al. (1976) that model based controllers are more sensitive to model mismatch than controllers, such as PID, which use process information to a lesser extent; the question of the optimality of these two types of controller notwithstanding.

From the simulation and experimental results, it was observed that only the initial stage of the transient response is significantly affected by the saturation algorithm; whereas the overall performance is dictated by the controller and the extent of model mismatch. In any particular experiment, the performance is governed more by the type of controller and its tuning than by the saturation algorithm. There are examples of a poorly tuned controller with a poor saturation algorithm (e.g., the conventional clamp) giving better performance during saturation than the same poorly tuned controller with a better saturation algorithm (e.g., new anti-reset windup). This is because the better saturation algorithm does a better job in preserving the control actions from the poorly tuned controller, hence allowing more of the wrong control actions to affect the process. Of course, the poorly tuned controller degrades the performance when the manipulated variable is not saturating and a better overall performance can be achieved with a properly tuned controller and the best saturation algorithm.

2.5 CONCLUSION

The simulation and experimental runs show that the proposed algorithms not only handle the windup problem but also perform much better than the conventional clamping algorithm. Although more computation is required for these new algorithms, the increase in speed of digital computers means that these algorithms can be applied on-line; this was demonstrated by the experimental work. The Heuristic and New anti-reset algorithms can be implemented easily without too much extra computer memory, therefore they may be incorporated into frontend computer systems for low level control. The Optimal switch algorithm requires a little more memory and computation and is more suitable for implementation on high level control algorithms in the host computer. The QIMC algorithm is a sequence of local, constrained optimizations which is equally applicable to SISO and multivariable processes. It provides the greatest flexibility in specifying the constraints on process input and output and produces the best performance. It requires considerably more computation than the other algorithms and is best suited for higher level, supervisory control. The proposed saturation algorithms show considerable improvement over the conventional clamping algorithm; they give better performance as well as windup protection. It is therefore worthwhile to incorporate these saturation algorithms into model based digital control algorithms.

2.6 NOMENCLATURE

$Y^{CL}(j)$: closed loop response at time j

$Y_{predicted}(i+1+f:i+P+f)$: vector ($Px1$) of predicted outputs composed of $Y_{predicted}(i+1+f)$, $Y_{predicted}(i+2+f),\ldots,$ $Y_{predicted}(i+P+f)$

$U_{future}(i:i+M-1)$: vector ($Mx1$) of future process inputs composed of $U(i)$, $U(i+1)$, ..., $U(i+M-1)$

$U_{past}(i-N+1:i-1)$· Vector (($N-1$) x 1) of past process inputs composed of $U(i-N+1)$, $U(i-2)$, ..., $U(i-1)$

$E_{predicted}(i+1+f:i+P+f)$: vector ($Px1$) of prediction errors composed of the best available forecast of the combined model uncertainty and process disturbances over the $i+1+f$ to $i+P+f$ prediction period

N: system order (# of impulse weights)

M: input horizon (# of intervals into the future over which the process input is to be computed)

P: prediction horizon (# of intervals into the future over which the process output is to be optimized)

Ψ: matrix (PxM) of impulse weights for future inputs

Φ: matrix (P x ($N-1$)) of impulse weights for past inputs.

$E_{trajectory}(i+1+f:i+P+f)$: vector ($Px1$) of future deviations from the desired trajectory composed of $E_{trajectory}(i+1+f)$, $E_{trajectory}(i+2+f)$, ..., $E_{trajectory}(i+P+f)$

$Y_{desired}(i+1+f:i+P+f)$: vector ($Px1$) of the desired output responses over the prediction horizon, composed of $Y_{desired}(i+1+f)$, $Y_{desired}(i+2+f)$, ..., $Y_{desired}(i+P+f)$

T: *Control Interval*

f: *discrete process dead-time*

$Y_{low}(i+i+f+i+P+f)$: *Vector* ($Px1$) of low limits on the output prediction

$Y_{high}(i+i+f+i+P+f)$: *Vector* ($Px1$) of high limits on the output prediction

$U_{low}(1:i+M-1)$: Vector ($Mx1$) of low limits on the future inputs

$U_{high}(1:i+M-1)$: Vector ($Mx1$) of high limits on the future inputs

$U_{difference}(1:i+M-1)$: Vector ($Mx1$) of high limits on the rate of change of future inputs

$\nabla U(i)$ difference input at time i
$u(i) - u(i-1)$

Table 2-1 : Performance Index for Proposed Saturation Algorithm

Algorithm	Process 1		Process 2		Process 3	
	Simulation	Experiment	Simulation	Experiment	Simulation	Experiment
Conventional	6.04	7.14	4.43	5.46	4.30	5.77
Heuristic	2.52	4.12	1.97	2.56	2.25	3.81
Optimal Switch	1.99	2.59	1.56	2.13	1.74	4.53
New Anti-Reset	2.37	3.29	1.56	2.03	1.74	4.42
QIMC	1.97	3.02	1.54	1.73	1.73	3.68

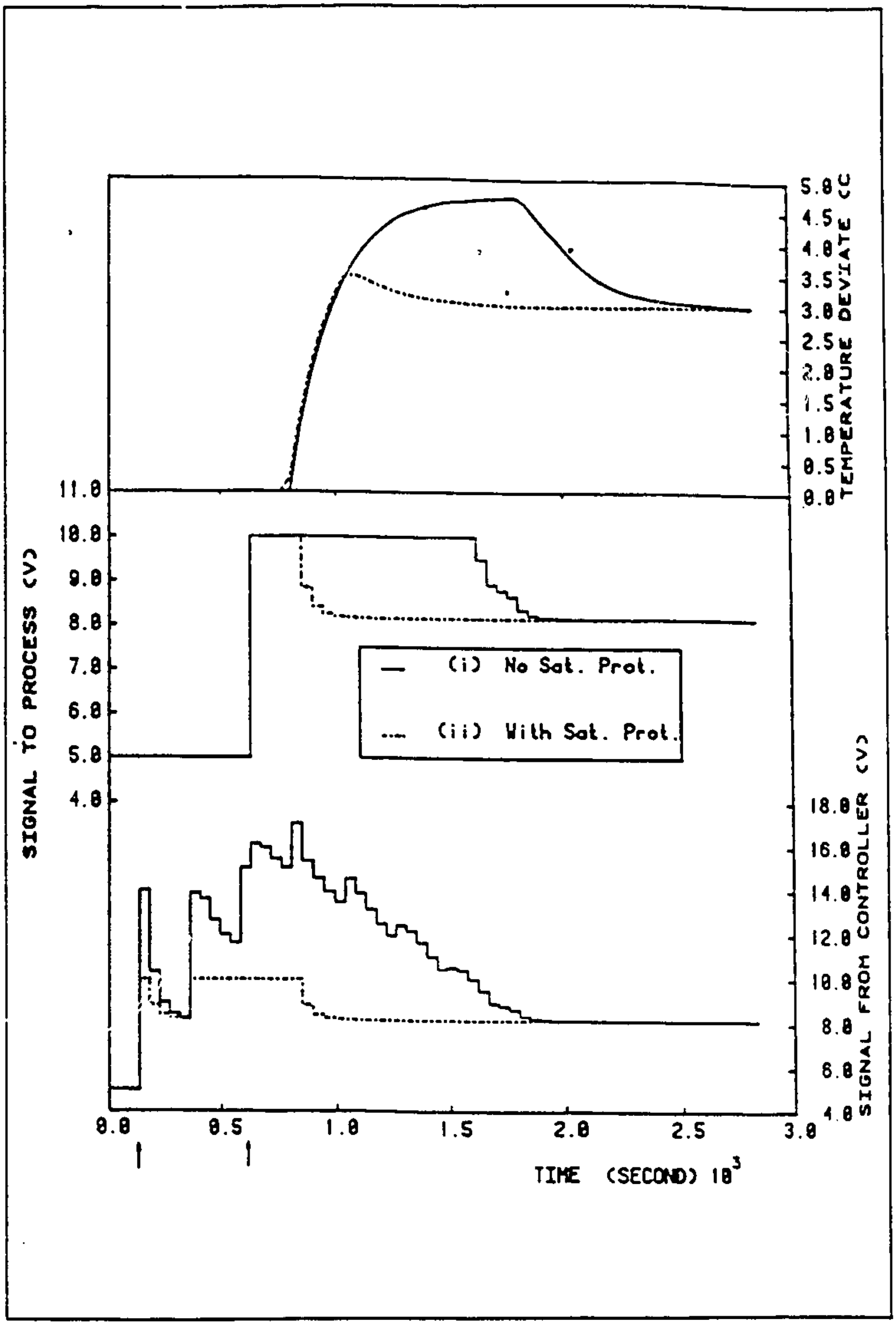


Figure 2.1: Windup Problem in a Digital Controller

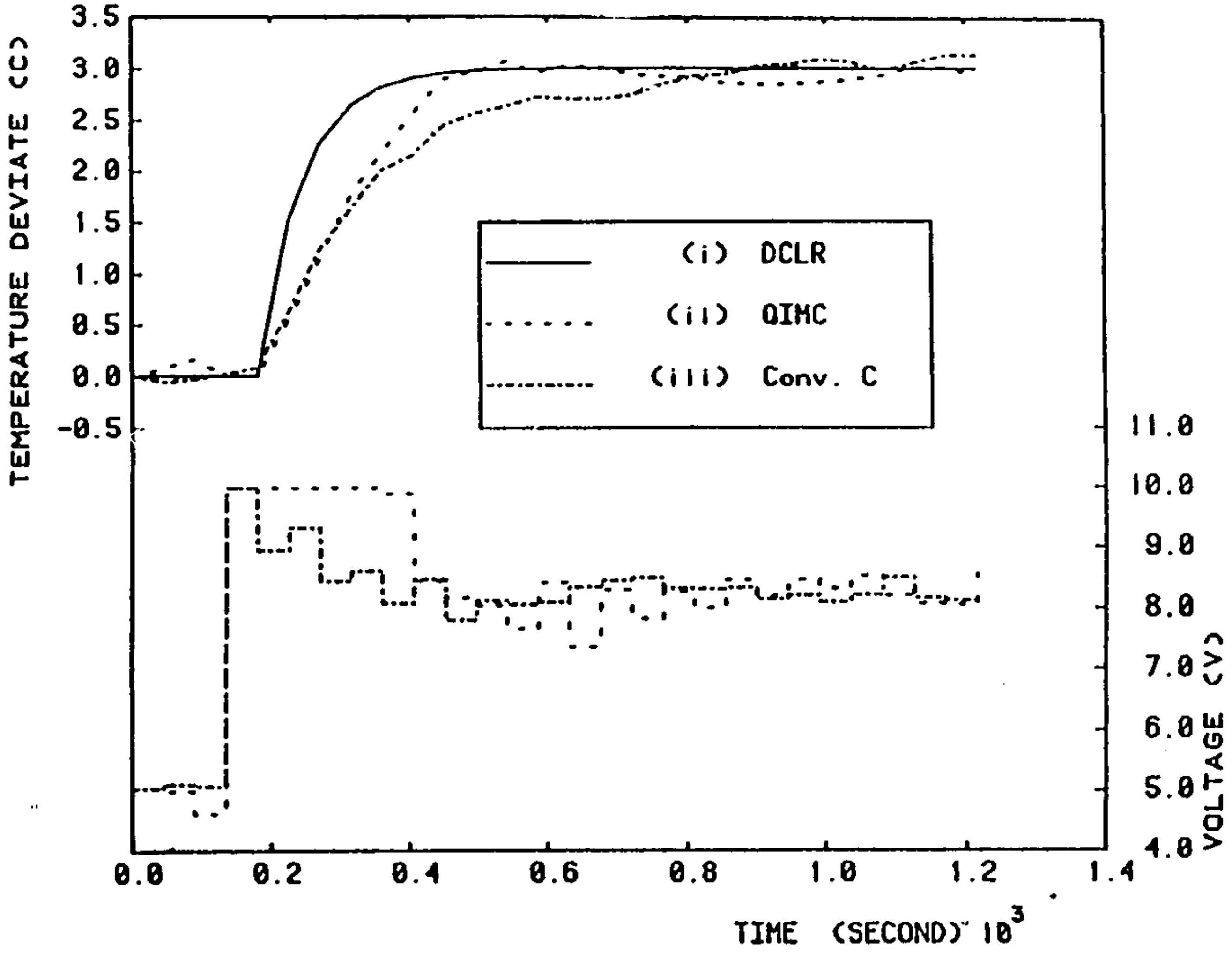


Figure 2-2a: Experiment - Servo responses for nominal process (Process 1)

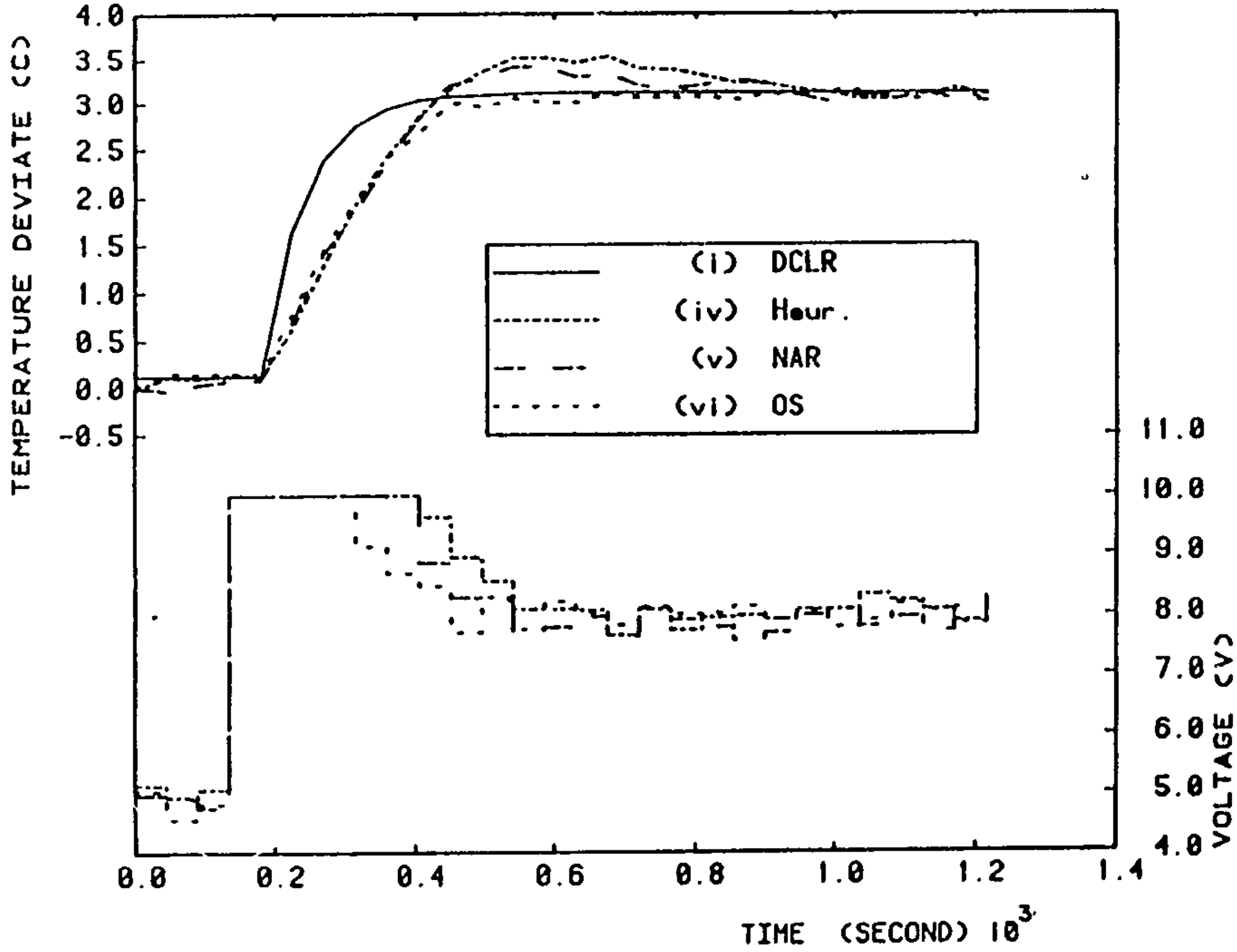


Figure 2-2b: Experiment - Servo Responses for nominal process (Process 1)

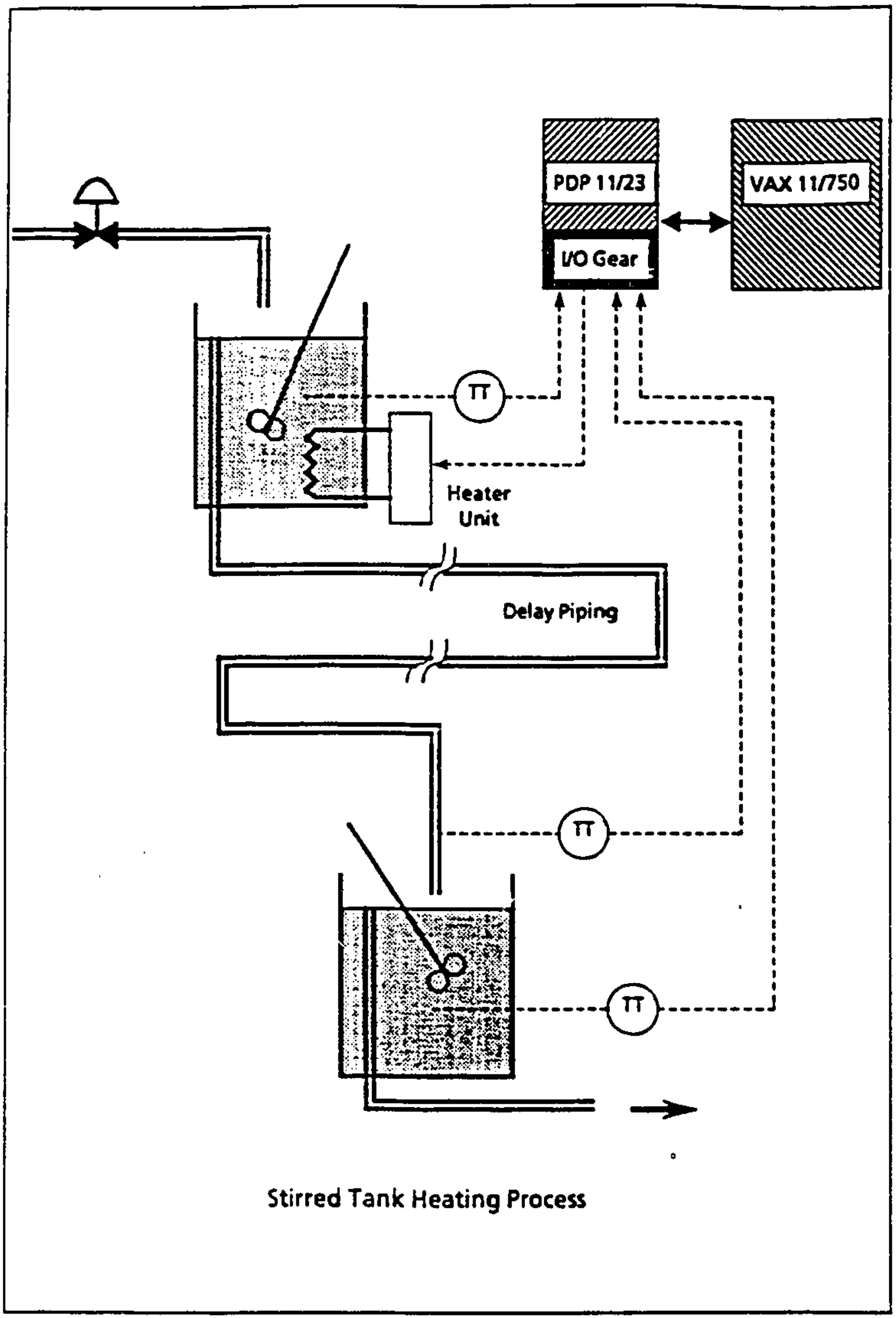


Figure 2-3: Schmatic Diagram of the Pilot-scale Stirred Tank

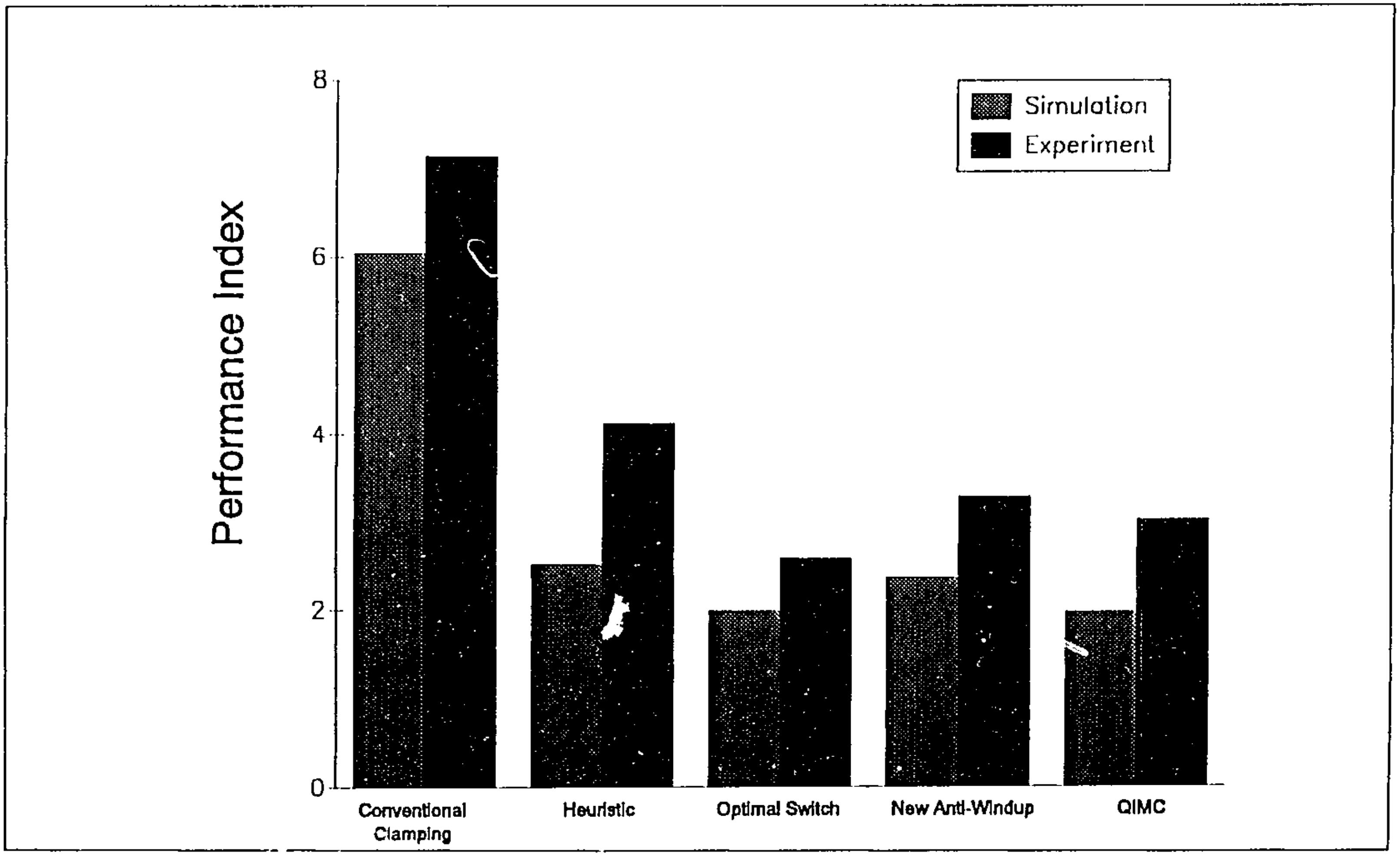


Figure 2-4: Performance Indices for Process 1

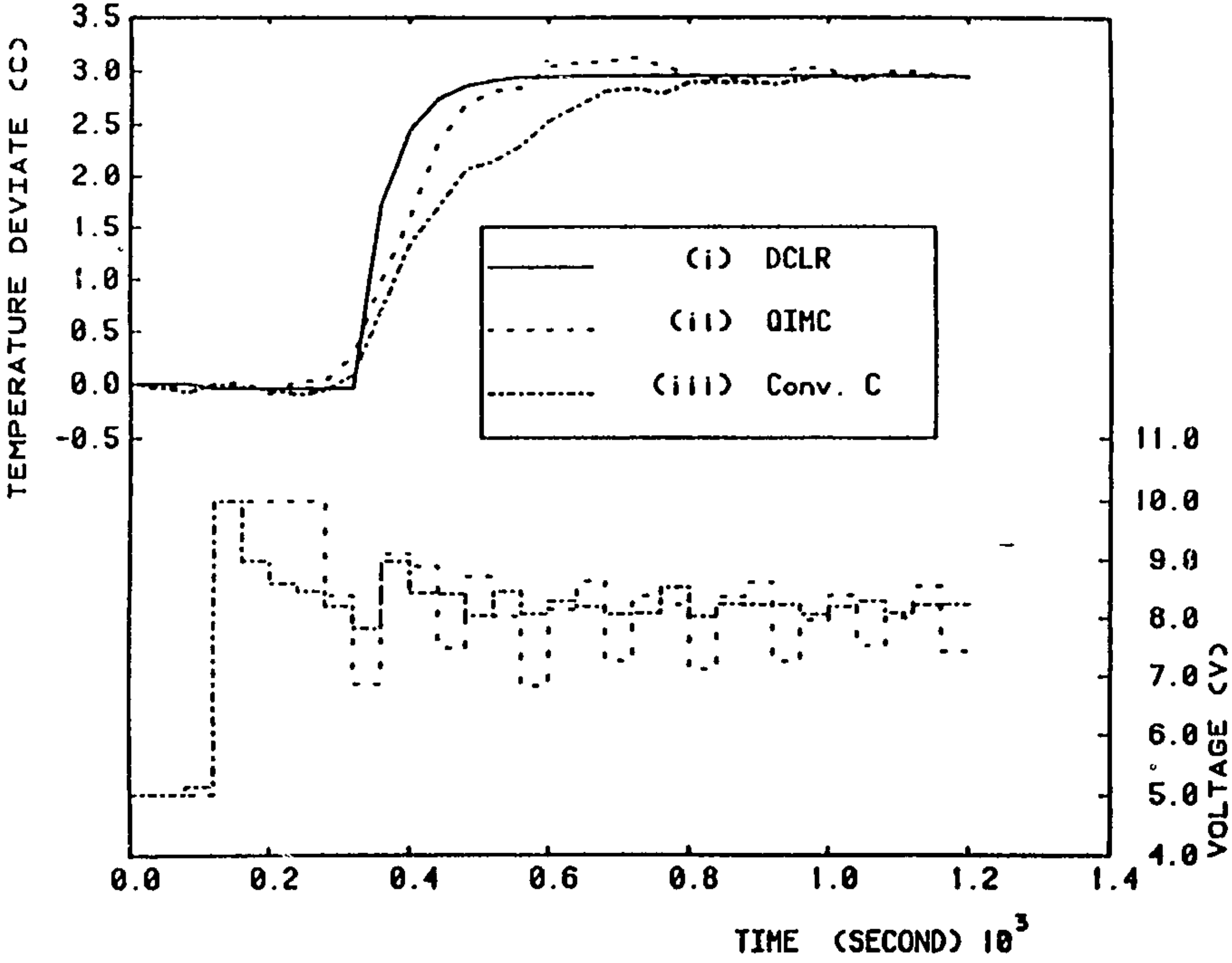


Figure 2-5a: Experiment - Servo Responses for severe control problem (Process 2)

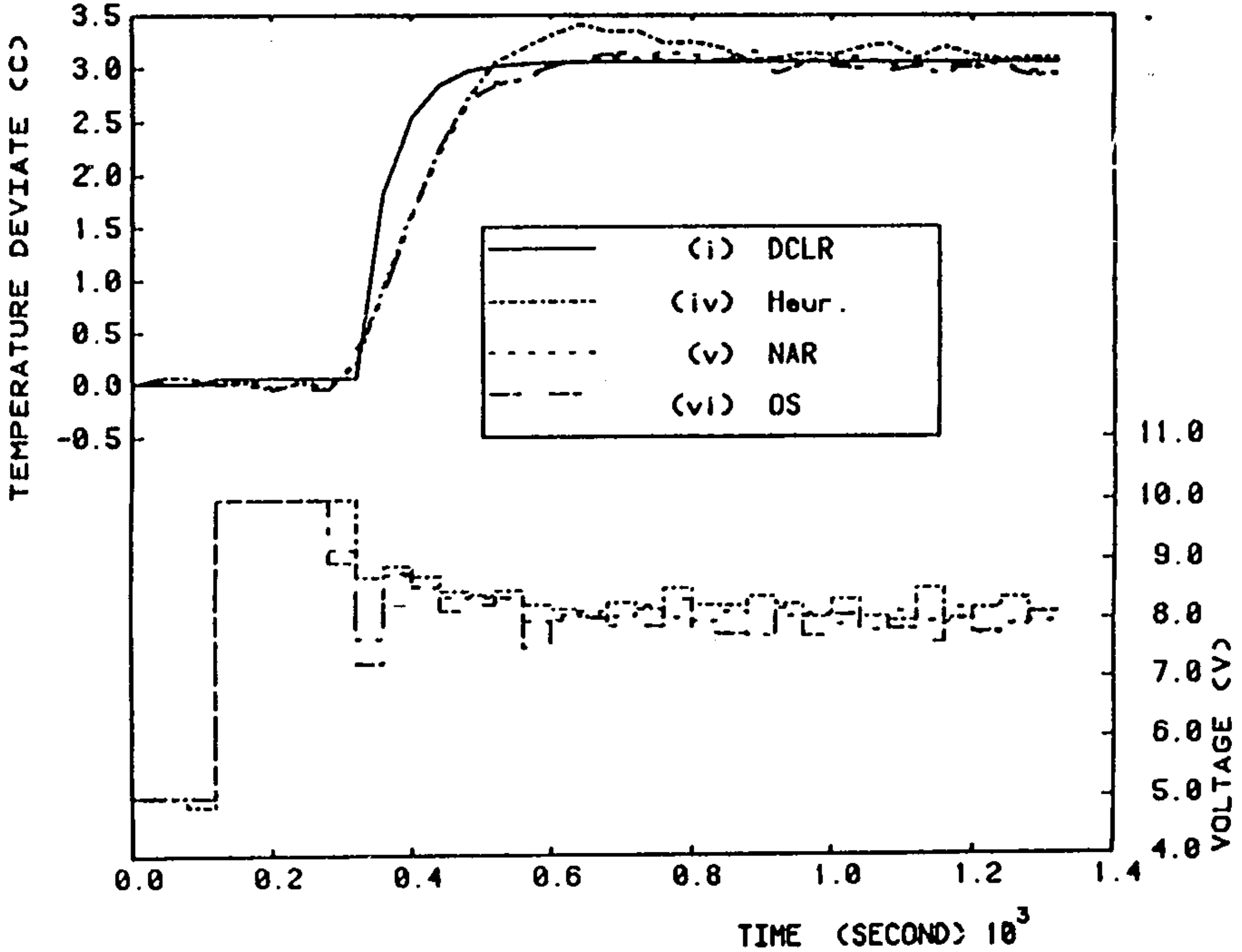


Figure 2-5b: Experiment - Servo Responses for severe control problem (Process 2)

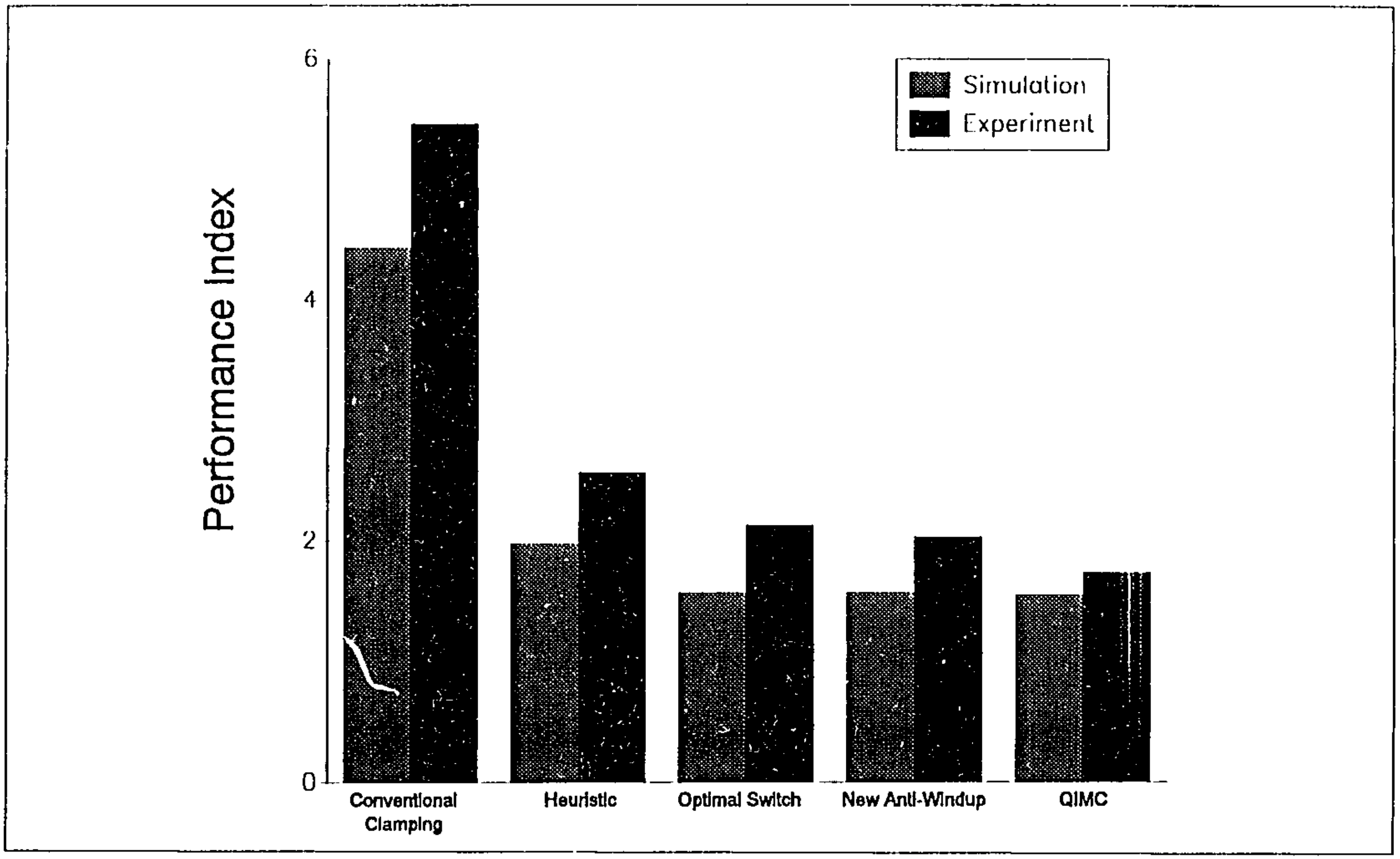


Figure 2-6: Performance Indices for Process 2

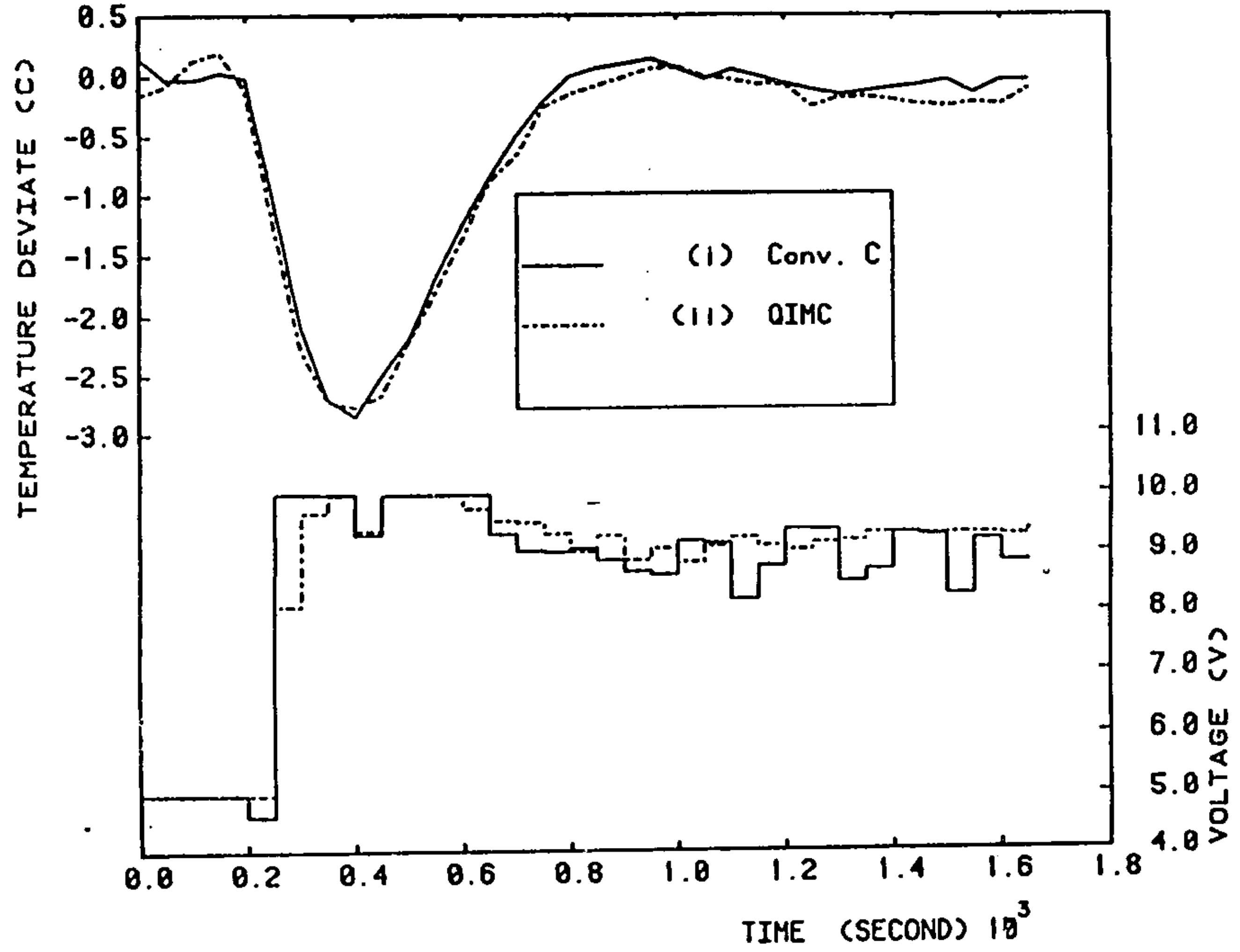


Figure 2-7a: Experiment - Disturbance regulation responses (Process 3)

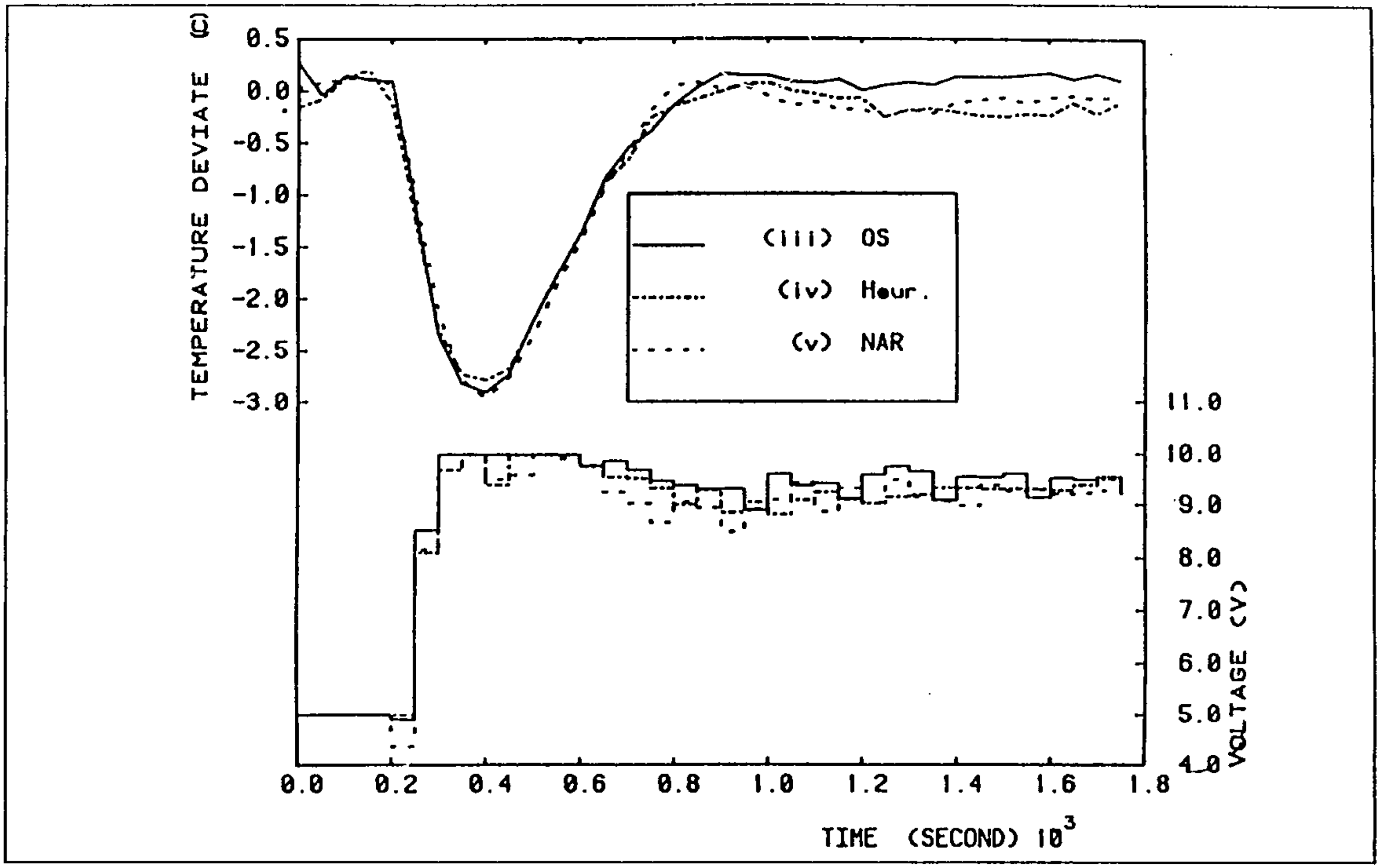


Figure 2-7b: Experiment - Disturbance regulation responses (Process 3)

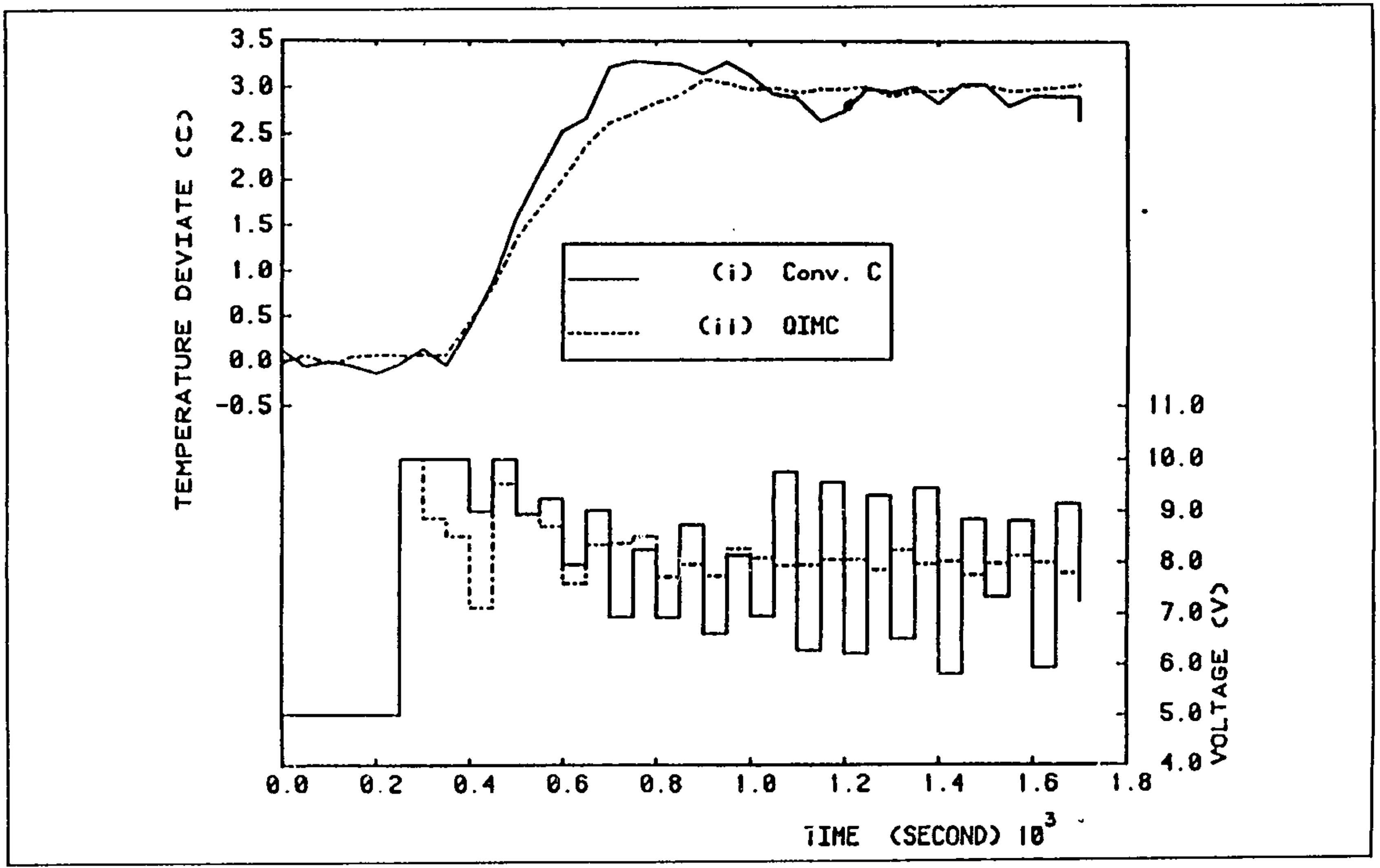


Figure 2-8a: Experiment - Servo Responses with structural model mismatch (Process 3)

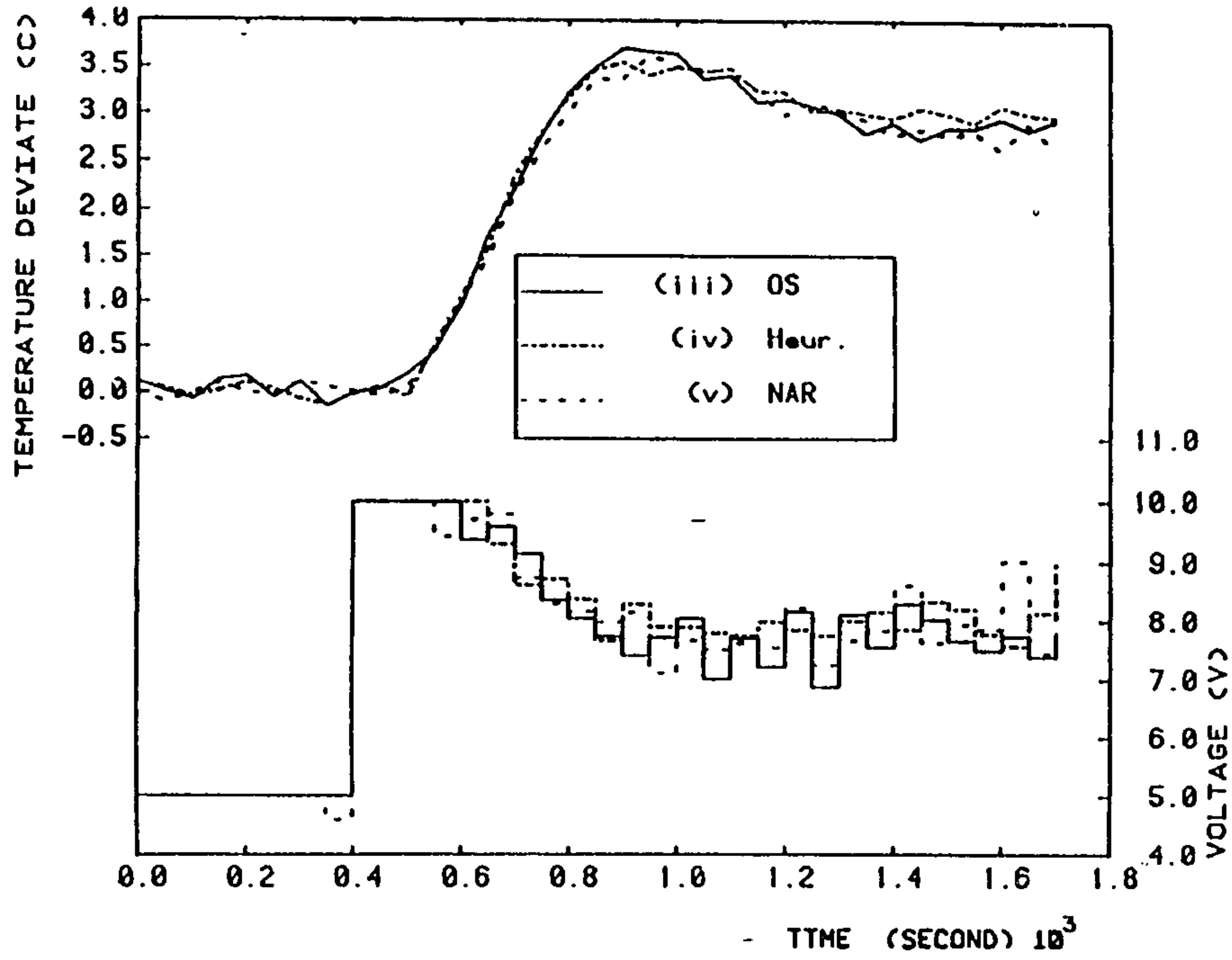


Figure 2-8b: Experiment - Servo Responses with structural model mismatch (Process 3)

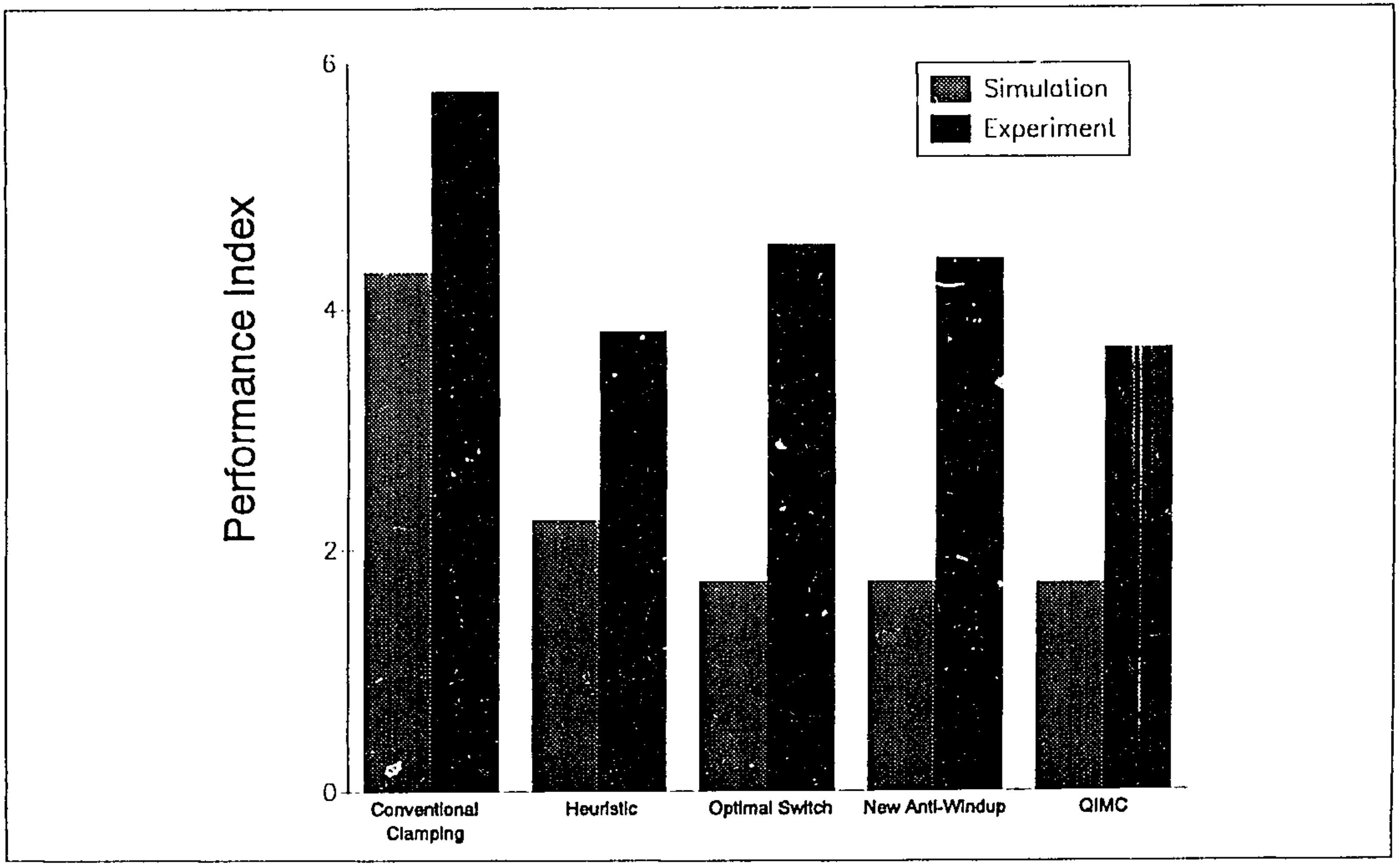


Figure 2-9: Performance Indices for Process 3

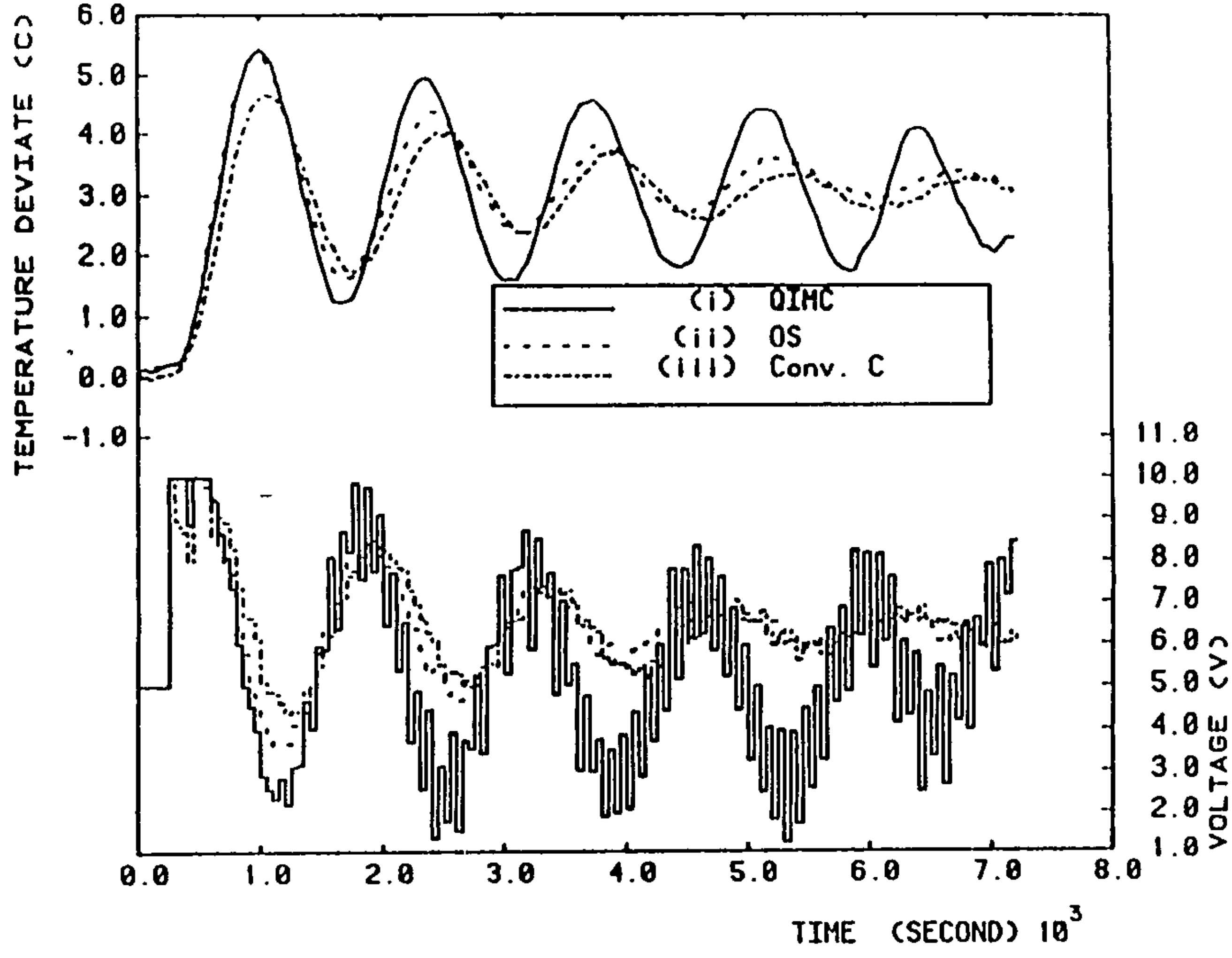


Figure 2-10: Experiment - Servo Responses with gross parameter mismatch (Process 4) for a 3°C setpoing change

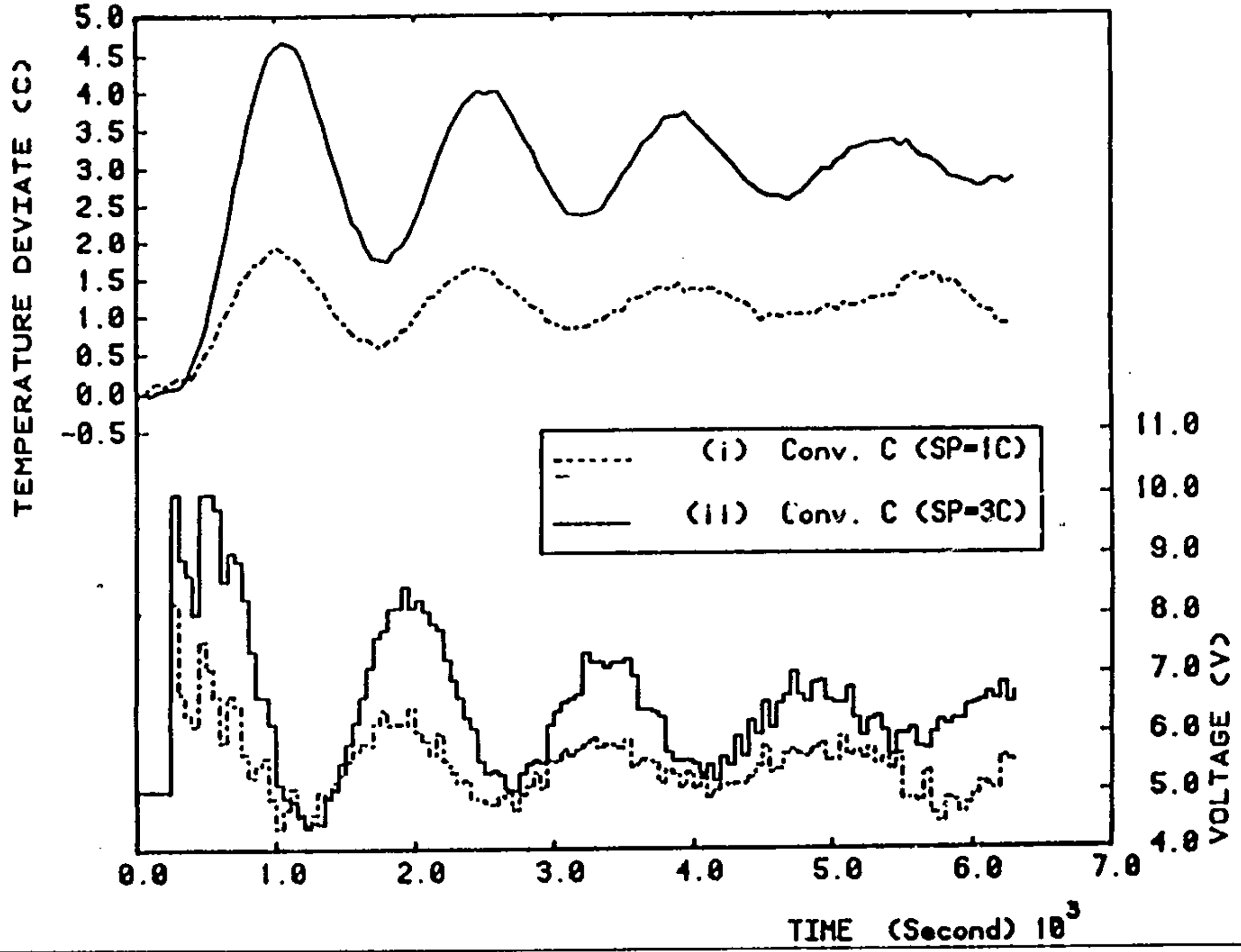


Figure 2-11a: Experiment - Servo Responses with gross parameter mismatch for a 1°C and 3°C setpoint changes using Conventional Clamping

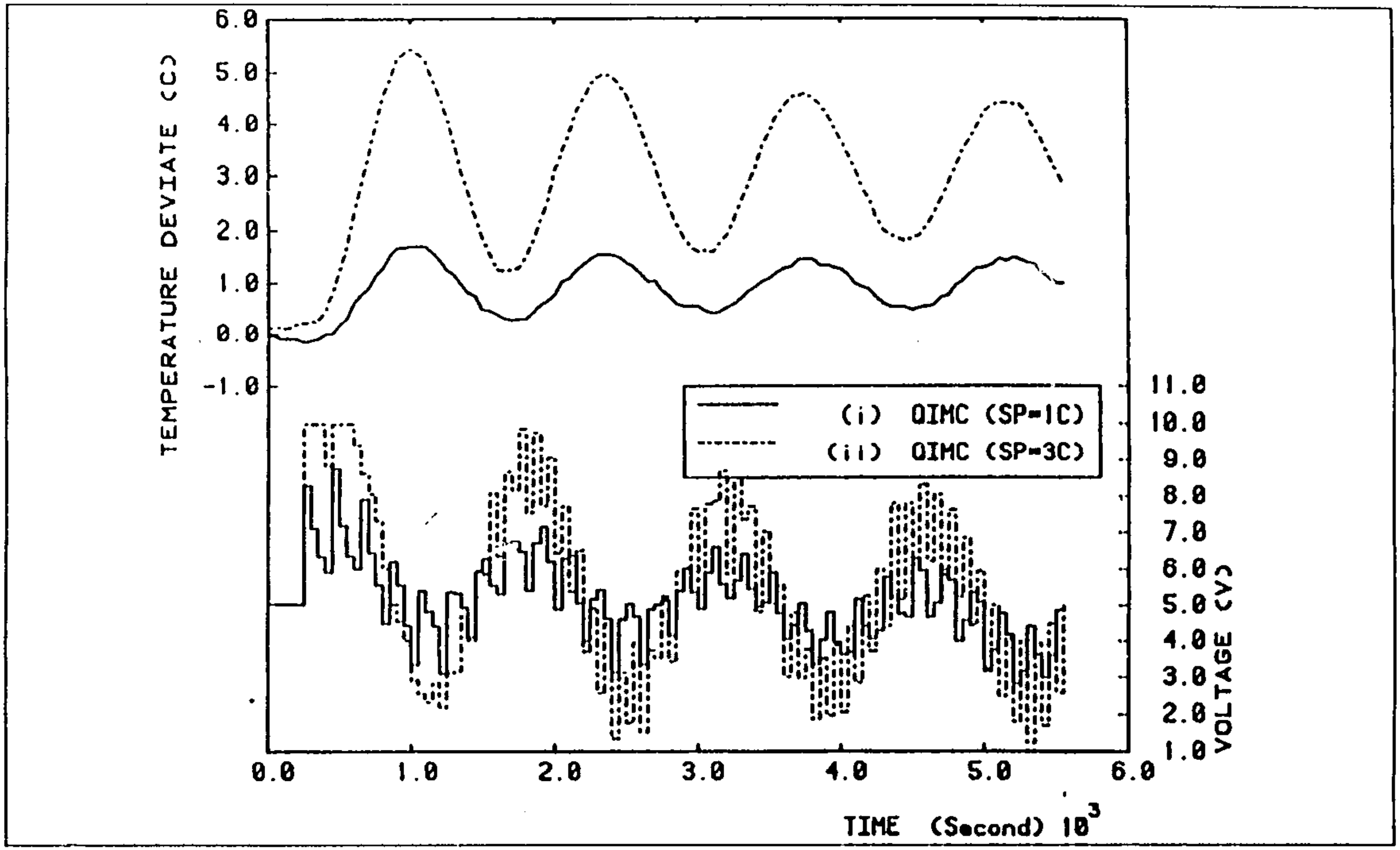


Figure 2-11b: Experiment - Servo Responses with gross parameter mismatch for a 1°C and 3°C setpoint changes using QIMC algorithm

CHAPTER 3

IMPLEMENTATION OF RATIONAL MODEL BASED DIGITAL CONTROLLERS FOR SISO AND MIMO SYSTEMS

3.1 Introduction

A theoretical and experimental investigation of saturation protection for model based digital controllers has been discussed in Chapter 2. However, saturation is only a subset of the more general *"Implementation Problem"*. The implementation problem is here defined as the loss of performance as a result of the method of implementing a digital controller. The performance of a conventionally implemented digital controller can deteriorate when the control action specified by the controller is not fulfilled. This can be due to incidents such as actuator failure, operator override or saturation, all of which occur quite frequently in process control systems. In fact, some control engineers apply model based digital control to high level control loops only to find the controller performance has been ruined by the occasional override by the operators (see Doyle et al. (1987)). As a result, model based digital control algorithms are usually considered to be "not robust enough" for industrial applications.

There are many model based digital control algorithms available in the literature, but their application to industrial processes is scarce. One major factor is the lack of robustness in the algorithms for industrial applications. In some cases, this is due to the fact that even though control engineers have spent a great deal of time on the design of digital controllers, the implementation side is often overlooked. This study identifies the inadequacy of the conventional implementation of model based digital controllers and shows a detailed analysis of the implementation problem which leads to two simple optimal remedies - a *One-Interval Implementation* and a *Multi-Interval Implementation.* These implementations have been verified by computer simulations and the results show that the proposed remedies significantly improve the performance of model based digital controllers. For highly coupled Multi-Input Multi-Output (MIMO) system, the improvement is even more dramatic. Also, the other advantage of these implementation schemes is their simple and efficient computation

Most of the early publication on saturation (Astrom et al. (1984), Khandria et al. (1976), Smith (1972)) limit the investigation to PID controllers. Their approach is mainly to freeze the integration when the control action saturates. However, as demonstrated in Doyle et al. (1987), a controller can windup even without an integrator. It only needs a relatively "slow" controller to produce the undesirable saturation symptoms. Doyle et al. advocated a high gain feedback approach which uses the implemented control actions. However, they also pointed out that naive application of their scheme can lead to instability problem. Their scheme can be extended to MIMO systems. "Directionality" is the key in the MIMO saturation problem. They argue that the saturation scheme should preserve the "direction" of the original control vectors. This direction issue is again emphasized by Campo et al. (1990). Based on a state space framework, they studied the stability and performance of a saturation system. Their solution is similar to the "conditioning controller" proposed by Hanus et al. (1987). The idea is that the controller states should be driven by the implemented control actions, not the "calculated" control actions. It is interesting to note that though this saturation work starts from a totally different basis, the final strategy is very similar to their scheme : using the implemented control action to drive the controller. Both Doyle et al. and Campo et al. pointed out that IMC controller using a rational model implementation can provide a stable saturating system, but has a sluggish performance. This work further shows that a "properly implemented" IMC controller can preserve the design performance even in face of saturation.

Before focusing on the implementation problem, it is important to define the types of model based digital controllers which are being investigated in this chapter. Model based digital controllers in input/output space can be divided into two categories : those which use rational process models[1] (e.g. Linear Quadratic Gaussian(LQG) controller, Dahlin controller, Constrained Minimum Variance(CMV) controllers) - see Harris et al. (1987); and those which use irrational process models[2] - see Cutler and Ramaker (1980) and Richalet et al.

[1] Models expressed in terms of fractional difference polynomials.

$$y(i) = \frac{\omega(z)z^{-b}}{\delta(z)} u(i) + N(i) \quad \text{(Box and Jenkins 1970)}$$

[2] Model expressed in terms of impulse weights or step weights

$$y(i) = \sum_{j=0}^{N} h(j)\, U(i-j) + d(i)$$

(1978). The term "rational" here refers to the fact that the model can be expressed in fractional form. Irrational process models lead to control algorithms like IMC (Internal Model Control), DMC (Dynamic Matrix Control), MAC (Model Algorithmic Control) ; see (Cutler et al. (1980), Garcia et al. (1981), Richalet et al.(1978), Rouhani et al. (1982)) : note that there were studies which implement IMC using a rational model (see Campo et al. (1990), Doyle et al. (1987)). Irrational model based controllers usually perform very well even when the control action cannot be exactly implemented. In fact, this study found that the performance of irrational model based controller corresponds exactly to the constrained optimal solution under certain conditions. However, this characteristic is not inherent in controllers which are derived from a rational process model, Dahlin, LQG or CMV controllers (Wong et al. (1987a), Kczub (1986), Doyle et al. (1987), Campo et al. (1990)) .

The advantage of using a controller derived from rational models is the parsimony in controller parameters. This makes the real time computation of the control actions more efficient than the irrational model based controllers. For a 2 x 2 system, using 40 step weights to describe each transfer function, the computation time for irrational model based controller can be three times longer than the rational model based controller. For higher order and higher dimension systems, this difference will be more dramatic.

The design of most rational and irrational model based controllers is based on the solution of a quadratic optimization with very similar objective functions. The intriguing question is why one type of controller has inherent protection against the implementation problems while the other has none. Because of the similar objective functions, it should be possible to make a rational model based controller match the performance of an irrational model based controller while still retaining the efficiency of the rational implementation. This is the purpose of this chapter. This study does not attempt to develop a new controller. The issue is how to correctly implement a rational model based digital controller so that performance is not lost when the prescribed control action cannot be applied to the process. The major contribution of this work is the analysis of the implementation problem which leads to simple remedies and a clear interpretation of their optimality.

This chapter is organized as follows. Section 3.2 presents the design and implementation of a rational model based controller. The difficulties with the conventional implementation are discussed in section 3.2.3. This is followed by the proposed implementation schemes - *One Interval Implementation* and *Multi-Interval Implementation.* These schemes are used in a simulation study which is described in section 3.2.4. Sections 3.2.5 and 3.3 discuss the extensions of the implementation schemes to long horizon controllers such as LQG and to model based rational MIMO controllers. The conclusions are presented in section 3.4. The nomenclature for this chapter is listed in section

3.5.

3.2 Rational Model Based Digital Controller

The dynamics of a process can be represented in the input/output space by rational backward difference polynomials :

$$\text{Process Model} \quad : \quad y(i) = \frac{\omega(z)z^{-b}}{\delta(z)} U(i) + N(i) \qquad (3\text{-}1)$$

$$\text{Disturbance Model} \quad : \quad N(i) = \frac{\theta(z)}{\phi(z)\nabla^{d}} a(i) \qquad (3\text{-}2)$$

$$\text{Servo Trajectory Model} : y_{sp}(i) = \frac{K_{n}(z)z^{-b}}{K_{d}(z)} \cdot R(i) \qquad (3\text{-}3)$$

The notation used here follows the backward difference operator convention in Box and Jenkins (1970). For example, equation 3-3 means :

$$Y_{sp}(i) = 1/K_{d}(0) \left[\sum_{j=0}^{Or(K_{n}(z))} K_{n}(j)\ R(i-j-b) - \sum_{j=1}^{Or(K_{d}(z))} K_{d}(j) Y_{sp}(i-j) \right] \qquad (3\text{-}3a)$$

If these rational polynomial models are used to derive a digital control law, then the conventional method of implementation (defined below) will perform poorly whenever the prescribed control actions are not exactly applied to the process. A common cause is the saturation of the manipulated variable, but operator override or actuator failure would also cause this problem. This is the "implementation problem" under investigation in this chapter. This study does not result in a "global" optimal solution to the problem, but, the proposed implementation scheme can stop the performance deterioration found in a conventionally implemented rational model based controllers. With the proposed remedy the performance of the control system is always "close" to the performance specified in the controller design.

All model based digital controllers solve an optimization problem. The objective function of many model based digital controllers is quadratic and convex (see equation 3-4). Garcia et al. (1981) offered a detailed discussion on the parameters of this equation.

$$\Omega = \min_{\nabla\underline{U}(i)} \sum_{k=i}^{i+P-1} \alpha \, [Y_{sp}(k+b)-Y(k+b)]^2 + \sum_{k=i}^{i+M-1} \beta \, [\nabla U(k)]^2 \qquad (3\text{-}4)$$

where P and M are the Output and Input Horizons respectively and $\nabla\underline{U}(i) \in \mathcal{R}^M$

For example, if $\alpha=1.0$, $\beta=0.0$, $P=M=1$ and an exponential trajectory is specified for $Y_{sp}(k+b)$, the optimization will result in a Dahlin controller response. The original derivations of Dahlin, CMV and LQG controllers are quite different. However, by properly setting the values of P, M, α and β in the optimization equation 3-4, this same optimization can yield responses similar to those produced by these controllers. Equation 3-4 is therefore used throughout this study so that a general model based digital controller can be studied under the same framework. From common basis, a common implementation problem can be isolated.

The optimization in equation 3-4 requires a model of $Y_{sp}(i+b)$ and $Y(i+b)$ (see Appendix B-1 for details of these models). In the input/output space, these models can be expressed in either *rational* polynomial form :

$$Y(i) = \frac{\omega(z)z^{-b}}{\delta(z)} U(i) + \frac{\theta(z)}{\phi(z)\nabla^{d}} a(i) \tag{3-5}$$

or *irrational* (equation 3-6) polynomial form:

$$Y(i) = Gm(z)\ U(i) \quad + \quad \psi(z)\ a(i) \tag{3-6}$$

where $\omega(z)$, $\delta(z)$ and $Gm(z)$ are themselves irrational polynomials

The rational polynomial model form has many advantages over the irrational form. The rational form has less parameters, thus many rational controller designs result in simpler algorithms and the controller requires less computer memory space. Consequently the real time execution of the algorithm is more efficient. Furthermore, the compact rational polynomial models allow a more efficient stability analysis. These are the incentives to provide rational digital controllers with an optimal implementation scheme.

3.2.1 Optimal Solution

The unconstrained optimal solution for the objective function in equation 3-4 can be obtained using variational calculus. For demonstration purposes, assume that the system is SISO (single input single output) and that the input and output horizons are 1 (*P=M=1*); i.e., using 1 control action to optimize the objective function which penalizes only the first output after the process dead time. By differentiating equation 3-4 with respect to the current change in the control action, $\nabla U(i)$, one obtains :

$$\frac{\delta\Omega}{\delta\nabla U(i)} = -\,2\alpha \left\{ Y_{sp}(i+b) - Y(i+b) \right\} \omega(0) + 2\beta\ \nabla U(i) \qquad (3\text{-}7)$$

where $\omega(0)$ *is the first term of* $\omega(z)$

The optimum can be achieved by setting this derivative to zero.

Optimal Solution :

$$\left\{ Y_{sp}(i+b) - Y(i+b) \right\} = \frac{\beta}{\alpha\ \omega(0)}\ \nabla U(i) \qquad (3\text{-}8)$$

Equation 3-8 is the *Optimal Unconstrained Solution* for the linear quadratic objective function in equation 3-4. The essence of this study is to properly implement this *Optimal Solution.*

3.2.2 Conventional Implementation

The conventional implementation of the optimal solution in equation 3-8 above is illustrated in this section. Substituting the rational models, $Y_{sp}(i+b)$ and $Y(i+b)$ (see Appendix B-1) into equation 3-8, produces another form of the *Optimal Solution* :

$$\frac{K_n(z)}{K_d(z)} R(i) - \frac{\omega(z)}{\delta(z)} U(i) - \frac{T(z)}{\theta(z)} N(i) = \frac{\beta}{\alpha\omega(0)} \nabla U(i) \qquad (3\text{-}9)$$

where $$N(i) = Y_m(i) - \frac{\omega(z)z^{-b}}{\delta(z)} U(i)$$

Feedback enters through the estimates of the current disturbance, $N(i)$, which is related to the measurement, $Y_m(i)$ (see Appendix B-1). Equation 3-9 is different from commonly seen rational digital controllers. However, it can be shown that Equation 3-9 can be degenerated to a Dahlin or a MVC controller (see Appendix B-3). Usually, rational controller design software takes $K_n(z)$, $K_d(z)$, $\omega(z)$, $\delta(z)$, $\theta(z)$, α and β as input parameters (Wright et al. (1988)). It then calculates $T(z)$ and finally produces a numerator polynomial, Nu(z) and a denominator polynomial, De(z) as shown in equation 3-9a where Ym(i) is the measurement of the process output. These polynomials, Nu(z) and De(z) are then used for the implementation of the controller.

$$\frac{Nu(z)}{De(z)} (R(i) - Y_m(i)) = \nabla U(i) \qquad (3\text{-}9a)$$

The conventional implementation of a rational digital controller, equation 3-9, is equivalent to the following steps :

Step 1 - Polynomial Multiplication

Multiply $K_d(z)\delta(z)\theta(z)$ to both sides of equation 3-9, this results in an irrational backward difference expression :

$$\begin{aligned} K_n(z)\delta(z)\theta(z)R(i) &= \omega(z)K_d(z)\theta(z)U(i) \\ &+ T(z)K_d(z)\delta(z)N(i) + K_d(z)\delta(z)\theta(z)\frac{\beta}{\alpha\omega(0)}\nabla U(i) \end{aligned} \tag{3-10}$$

Step 2- Separate the current control action from the past control actions

$$\begin{aligned} \{\omega(0)K_d(0)\theta(0)+\frac{\beta}{\alpha\omega(0)}\ K_d(0)\delta(0)\theta(0)\}\ U(i) &= \\ - \{\sim(\omega(z)K_d(z)\theta(z)) - \sim(\frac{\beta}{\alpha\omega(0)}K_d(z)\delta(z)\theta(z)\nabla)\}U(i) \\ + K_n(z)\delta(z)\theta(z)R(i) - T(z)K_d(z)\delta(z)N(i) \end{aligned} \tag{3-11}$$

where the operator, ~, is defined as following:

$$\omega(z) = \omega(0) + \omega(1)z^{-1} + \omega(2)z^{-2} + \ldots\ldots + \omega(n)z^{-n}$$

$$\sim(\omega(z)) = \omega(1)z^{-1} + \omega(2)z^{-2} + \ldots\ldots + \omega(n)z^{-n}$$

This rearrangement makes $U(i)$ depend explicitly on the $R(i)$'s, the $N(i)$'s and the past control actions, $U(i-j)$'s.

Step 3- To overcome the Windup problem, remember the actual values of the past control actions $U(i-j)_{imp}$, rather than the calculated values

$$\begin{aligned}&\{\omega(0)K_d(0)\theta(0)+\frac{\beta}{\alpha\omega(0)} K_d(0)\delta(0)\theta(0)\}\ U(i) = \\ &-\{\sim(\omega(z)K_d(z)\theta(z))- \sim(\frac{\beta}{\alpha\omega(0)}K_d(z)\delta(z)\theta(z))\nabla\}U_{imp}(i) \\ &+ K_n(z)\delta(z)\theta(z)R(i) - T(z)K_d(z)\delta(z)N(i)\end{aligned} \tag{3-12}$$

The use of past implemented control actions solves the windup problem experienced when using equation 3-11 (Wong et al. (1987a)).

The conventional implementation is to clamp the control action when saturation occurs. This is the same as using the past implemented control actions, $U_{imp}(i)$ and so Equation 3-12 is referred to here as the *Conventional Implementation.* To demonstrate the poor performance of this implementation, a simulation study was performed which was based on an empirical model of an extractive distillation process (see Appendix B). For the SISO case, only the top product (acetone concentration) was controlled in the simulation. Also notice that for the sake of demonstration, only saturation was used to illustrate the implementation problem. The results can be applied to operator override or actuator failure. The saturation protection algorithms introduced in chapter 2 were developed mainly to solve saturation problems which occur in SISO control systems (except the QIMC method) while the general scheme developed here applies to both SISO and MIMO control systems. Different rational model based controller responses were designed and implemented using equation 3-12. Figures 3-1, 3-2 and 3-3 are the responses corresponding to Dahlin, One-Step Optimal and LQG controllers when control action saturation occurs. Since Dahlin controller is based on a servo design, a step response was chosen to test the saturation behavior. However, MVC and LQG controllers are designed for regulatory control, load disturbances were used to test the controllers instead. The Dahlin response was chosen to be 30 minutes; which is half of the process time constant. With α=1.0, β values are specified as 0.0 and 1.0E-7 for One-Step MVC (Clarke et al.1973)) and LQG responses respectively. The true optimal solution can be obtained from an on-line constrained optimization. This allows a direct comparison of the conventional solution with the true optimal solution. One can see from these figures that due to the control action limitations, even the constrained optimal solution cannot reach the non-saturation solution. However, the conventional implementation produces a solution which is far from the feasible optimal solution. Naturally the question raised is "Which steps in the conventional implementation cause this loss of optimality?"

3.2.3 The Problem in the Conventional Rational Digital Controller Implementation

In order to investigate why the conventional implementation of a rational digital controller performs poorly, the *Optimal Solution* in equation 3-8 should be examined. To be truly optimal, the *Optimal Solution* at the current time, i, has to be satisfied. With rational process models, the *Optimal Solution* can be rewritten as shown in equation 3-9. Notice that the "to-be-calculated" current control action, *U(i)*, is grouped in equation 3-9 together with the past control actions *U(i-j)'s*. If the calculated control action can be exactly implemented, there would be no distinction between the nature of *U(i)* *and* the *U(i-j)'s*. Should the implemented control action be different from the calculated one, *U(i)*, then the past *U(i-j)'s* should use the external readback values (the actual implemented values) and thus the past *U(i-j)'s* should be handled differently from the "to-be-calculated" control action, *U(i)*. Instead of performing the polynomial multiplication in the conventional implementation (equation 3-10), one should first separate the past control actions, *U(i-j)'s*, from the "to-be-calculated" control action, *U(i)*. With $K_d(0)=1.0$, $\theta(0)=1.0$ and $\delta(0)=1.0$, Equation 3-9 then becomes :

$$\frac{K_n(z)}{K_d(z)} R(i) - \frac{T(z)}{\theta(z)} N(i) = \{\omega(0)+\frac{\beta}{\alpha\ \omega(0)}\} U(i) \qquad (3\text{-}13)$$

$$+ \{ \frac{\tilde{\omega}(z)}{\delta(z)} - \frac{\beta}{\alpha\omega(0)} \} U(i-1)$$

$$\text{where } \frac{\omega(z)}{\delta(z)} = \omega(0) + \frac{\tilde{\omega}(z)z^{-1}}{\delta(z)} \qquad (3\text{-}14)$$

$$\text{and} \quad \tilde{\omega}(z) = \tilde{\omega}(0) + \tilde{\omega}(1)\ z^{-1} + \tilde{\omega}(2)z^{-2} + \ldots. \qquad (3\text{-}14a)$$

Notice that the second term on the RHS of equation 3-13 contains only past control actions because $\tilde{\omega}(z)$ begins with constant term. From the optimization point of view, one should use the actual implemented control actions, $U_{imp}(i-j)'s$, for the past *U(i-j)'s* in equation 3-13 because the optimal solution, equation 3-8, requires an accurate prediction of the process output. With the rearrangement shown in equation 3-13 and the incorporation of the past implemented control actions, the optimal solution can be rewritten as :

$$\left\{\omega(0)+\frac{\beta}{\alpha\omega(0)}\right\} U(i)= \frac{K_n(z)}{K_d(z)} R(i) - \frac{T(z)}{\theta(z)} N(i) - \left\{ \frac{\tilde{\omega}(z)}{\delta(z)} - \frac{\beta}{\alpha\omega(0)} \right\} U_{imp}(i-1) \tag{3-15}$$

When comparing equation 3-15 with the conventional implementation in equation 3-12, it shows that the difference lies in the treatment of the past control actions, *U(i-j)'s*. Note that the solution here is similar to the strategy used in Campo et al. (1990) and Hanus et al. (1987) : the controller states should be driven by the actual implemented control actions, not the calculated ones. The *polynomial multiplication* step in the conventional solution (see equation 3-10) is the erroneous step. The polynomial multiplication operation implies that the *Optimal Solution* in equation 3-8 can be implemented in all control intervals. If the *Optimal Solution* is not properly compensated by using the implemented control action, the polynomial multiplication will link the past sub-optimality to the future solution and hence result in a solution which is optimal only when the control actions can be exactly implemented. Therefore, one can consider the remedy presented here as correctly using the most recent process information in the optimization. Of course, if the calculated *U(i)* can be exactly implemented, these two solutions (equation 3-12 and equation 3-15) are equivalent.

3.2.4 Remedies for Rational Model Based Controllers

Equation 3-15 is the remedy for the implementation problem. There are two ways to calculate the control action described in equation 3-15 - the *One-Interval Solution* and the *Multi-Interval Solution.*

The *One-Interval* solution can be carried out as follows :

1. $$Y_{SP}(i) = \frac{K_n(z)}{K_d(z)} R(i) \qquad (3\text{-}16)$$

2. $$N(i) = Y_m(i) - \frac{\omega(z) z^{-b}}{\delta(z)} U_{imp}(i) \qquad (3\text{-}17)$$

3. $$Y_N(i) = \frac{T(z)}{\theta(z)} N(i) \qquad (3\text{-}18)$$

4. $$Y_U(i) = \{\frac{\tilde{\omega}(z)}{\delta(z)} - \frac{\beta}{\alpha\ \omega(0)}\} U_{imp}(i-1) \qquad (3\text{-}19)$$

5. $$U(i) = 1/\{\omega(0) + \frac{\beta}{\alpha\ \omega(0)}\}\ [Y_{SP}(i) - Y_N(i) - Y_U(i)] \qquad (3\text{-}20)$$

One can obtain a *Multi-Interval* solution by multiplying both sides of equation 3-15 by $K_d(z)\delta(z)\theta(z)$.

$$\begin{aligned} K_d(z)\delta(z)\theta(z)\{\omega(0) + \frac{\beta}{\alpha\omega(0)}\}\ U(i) &= K_n(z)\theta(z)\delta(z)R(i) \qquad (3\text{-}21) \\ &- T(z)K_d(z)\delta z)Y_m(i) + \{T(z)K_d(z)\omega(z)z^{-b} \\ &- K_d(z)\theta(z)\tilde{\omega}(z) + \frac{\beta}{\alpha\omega(0)} K_d(z)\theta(z)\delta(z)\}\ U_{imp}(i-1) \end{aligned}$$

The solution derived here is different from the optimal saturation compensation solution proposed by Segall et al. (1984). Their solution is based on the One-Step Optimal controller argument of Clarke et al.

(1971) which only accounts for the last unimplemented control action. If the controller action cannct be implemented over a period of time, then their solution would not be "optimal".

The *One-Interval* equations (3-16 to 3-20) and the *Multi-Interval* in equation 3-21 are exactly equivalent. The *Multi-Interval Solution* yields a compact expression while the *One-Interval Solution* separately calculates the disturbance forecasts and the servo trajectory, which is useful information for process monitoring and diagnosis. Notice that both solutions have properly incorporated the past implemented control actions into the *Optimal Solution*. The difference between the conventional solution (equation 3-12) and the remedies (the *One-Interval* solution (equations 3-16 to 3-20) and the *Multi-Interval* solution (equation 3-21)) explains the loss of optimality in the conventional implementation.

As shown in the *Multi-Interval* solution (equation 3-21), the use of the past implemented control action before the polynomial multiplication breaks the connection between past sub-optimality and the future solutions. Therefore, if one can :

1. avoid the polynomial multiplication and
2. use the past implemented control actions,

then the resulting implementation would be optimal. These two criteria for solving the implementation problems correspond exactly to the natural implementation sequence for irrational model based controller

(Cutler et al. (1980)) and explain why irrational predictive controllers do not have implementation problems.

To confirm the optimality of the proposed remedies, computer simulations were again performed on the extractive distillation column model. Both the *One-Interval* and the *Multi-Interval* solutions were applied to the Dahlin and the One-Step Optimal controller. The settings of the controllers were the same as those in the conventional implementation study (see section 3.2.2). In Figure 3-4, the responses from the *One-Interval* and *Multi-Interval* implementations of the Dahlin controller correspond exactly to the constrained optimal solution. Figure 3-5 shows the remedied and the constrained optimal responses of the One-Step Optimal controller; again all responses are identical. These simulation cases show that the *One-Interval* and *Multi-Interval* solutions provide optimal implementations of SISO controls. By comparing Figures 3-4 and 3-5 with Figures 3-1 and 3-2, one can see that these remedies significantly improve the performance of a rational digital controller when compared to their conventional implementations. An index of the relative improvement from the conventional implementation can be calculated based on the integrated squared error :

$$ISE_b = \sum_{i=0}^{25} \alpha \ (\ Y_b(i) - Y \ (i) \ )^2 + \beta \ (\nabla U_b(i) - \nabla U(i))^2 \qquad (3\text{-}22a)$$

where Y (i) and ∇U(i) are the ideal process outputs and inputs when there are no saturation

Y_b and ∇U_b are the process outputs and inputs when there are

saturation

α and β are the weights used in the controller design

The α, β values in Equation 3-22a is to reflect the design criteria in Equation 3-4. Effectively, equation 3-22a is a measure of the deviation from the designed objective function value. The relative improvement is their ratio :

$$ISE_{scaled} = \frac{ISE_O}{ISE_C} \qquad (3\text{-}22b)$$

ISE_O : *ISE based on proposed implementation*

ISE_C : *ISE based on conventional implementation*

The smaller the ISE_{scaled} , the better the implementation. Equation 3-22a is a measure of the deviations of the objective function from the ideal case; the ideal is when all control actions can be exactly implemented. Equation 3-22b gives a relative improvement index, ISE_{scaled}. The indices for the different controller responses are shown in Figure 3-6. For the Dahlin and One-Step Optimal controllers, the ISE_{scaled} of the *One-Interval Implementation* is 61% and 32% respectively. This means a reduction of 39% and 68% in the deviation from the ideal response when the One-Interval Implementation is compared to the conventional implementation.

3.2.5 Remedy for the LQG Controller

The proposed implementation scheme has focused on the case of

P=M=1. This case encompasses a wide class of controllers which uses the exact model inverse (Harris et al. (1987)). For a non-invertible process model, the Dahlin or One-step Optimal controller designs would produce an unstable controller (Smith (1972)). A more sophisticated algorithm is required if one is to obtain a stable controller. LQG is one algorithm which can produce such a stable controller. Alternatively, increasing the input and output horizon (P and M in equation 3-4) also yields a stable controller. With large P and M (relative to process dynamics), the optimal solution would be close to the LQG solution in most cases (Harris et al. (1987)). The Optimal Solution in equation 3-9 shows a close resemblance to the block diagram (see Figure 3-7) of IMC (Garcia et al. (1981)). The most obvious difference is the optimal filter, or disturbance forecast, T(z)/θ(z), instead of a diagonal first order filter. Changing the values of *P* and *M* in equation 3-4, or using spectral factorization in LQG design, is a means of achieving a stable approximate model inverse, $\frac{\delta(z)}{\omega^{\circ}(z)}$. In general, if $\frac{\omega(z)}{\delta(z)}$ is invertible, the optimal inverse of $\frac{\omega(z)}{\delta(z)}$ is $\frac{\delta(z)}{\omega(z)}$. However, if the $\frac{\omega(z)}{\delta(z)}$ is non-invertible, the $\omega^{\circ}(z)$ can be obtained from spectral factorization (see Harris et al. (1987)) or by polynomial inversion technique (see Astrom et al. (1984)). Therefore, the general *optimal solution* for a model based rational digital controller can be written as (Kozub (1986)) :

$$\frac{K_n(z)}{K_d(z)} R(i) - \frac{T(z)}{\theta(z)} N(i) = \frac{\omega^{\circ}(z)}{\delta\ (z)} U(i) \qquad (3\text{-}23)$$

The extension of the *One-Interval Implementation* to this *Optimal Solution* in equation 3-23 is simply to break the term, $\frac{\omega^\circ(z)}{\delta\ (z)}$, into 2 components, $U(i)$ *and* $U(i-j)$*'s* and then to use the $U_{imp}(i-j)$*'s* in place of the past $U(i-j)$*'s*.

$$\omega^\circ(0)U(i) = \frac{K_n(z)}{K_d(z)}\ R(i) - \frac{T(z)}{\theta(z)}\ N(i) - \frac{\tilde{\omega}^\circ(z)}{\delta\ (z)}\ U_{imp}(i) \qquad (3\text{-}24a)$$

$$\textit{where} \quad \frac{\omega^\circ(z)}{\delta\ (z)} = \omega^\circ(0) + \frac{\tilde{\omega}^\circ(z)}{\delta\ (z)} \qquad (3\text{-}24b)$$

and

$$N(i) = Y_m(i) - \frac{\omega(z)z^{-b}}{\delta(z)}\ U_{imp}(i) \qquad (3\text{-}24c)$$

The computation of equation 3-24b is trivial. A polynomial long division can be used to obtain the desired expressions. The *Multi-Interval* solution can be obtained (see equation 3-25) by using the identity in equation 3-24c and then multiplying both sides of the *One-Interval* solution in equation 3-24a by $K_d(z)\delta(z)\theta(z)$.

$$\begin{aligned} K_d(z)\theta(z)\delta(z)\omega^\circ(0)U(i) = {} & K_n(z)\delta(z)\theta(z)R(i) \\ & - T(z)K_d(z)\delta(z)Y_m(i) + \{\ T(z)K_d(z)\omega(z)z^{-b+1} \\ & - \tilde{\omega}^\circ(z)K_d(z)\theta(z)\ \}\ U_{imp}(i-1) \end{aligned} \qquad (3\text{-}25)$$

A simulation was performed using the extraction distillation model (Appendix B-2). The responses of the proposed *One-Interval* and *Multi-Interval* solutions for the LQG implementation are shown in Figure

3-8. The constrained optimal solution in this figure is generated by constraining only the first future control action to the saturation limit. The responses of both implementations are identical, meaning that the constrained optimal solution is equal to the *One-Interval* and the *Multi-Interval* solutions. The ISE_{scaled} for this LQG run is only 51% which means that the remedy cuts the deviation by 49% from the conventional case.

In general, the constrained optimal solution, with constraints on all the future control actions set to the saturation limits will not be the same as the remedy solutions. This is due to the fact that for an N-variable convex quadratic optimization which has simple bound constraints, the constrained optimal solution is not necessarily at the bound. Therefore, one cannot interpret the optimality of the remedies by comparing them to constrained optimization. Instead, the optimality of the remedy should be interpreted using a *Moving Window* argument. In order to be optimal between the interval i to i+P-1, the *Optimal Solution* in equation 3-23 has to be satisfied. The *One-Interval* solution in equations 3-24a,b,c enforces this criterion at every control interval. One can then interpret the remedies as an optimization with a *Moving Window*. If the control action cannot be exactly implemented, then the system will not be optimal, but a fresh optimization with updated process information is started in the next interval. As long as the *Optimal Solution* is properly compensated (i.e., using the past implemented control actions), an optimal solution in the next window is possible. This idea has been used in irrational model based predictive controller algorithms (Cutler et al. (1980), Garcia et al. (1981)). In a predictive controller, one usually calculates *M* control actions and only the first one is implemented. In the next control interval, another *M* control actions are calculated and again only the first one is implemented. This *Moving Window* concept has not been used in the implementation of rational digital controllers. This study provides the link between the irrational predictive controller and the rational digital controller at the implementation level. One may ask: why bother with rational polynomial forms of IMC when the irrational form does not suffer from these implementation problems ? The answer comes from an analysis of the memory space and computer cycles needed to implement these forms. The irrational form stores process models as impulse or step weights and computes the prediction via convolution summations (multiplications and additions). On the contrary, rational forms require very few parameters to characterize the process models and the prediction is a very efficient computation.

It should be noted that using the *Internal Model Control (IMC) Structure* alone will not solve the implementation problem. Kozub (1986) has implemented rational model based controllers (CMV and LQG) using the IMC structure and his results show that the controllers performed poorly during saturation. Similar results were reported by Doyle et al. (1987) and Campo et al. (1990). They all implemented IMC controllers using a rational model. If the polynomial multiplication in equation 3-23 is performed prior to separating the past control actions from the current control action, the system would experience the same performance deterioration as in the conventional implementation. Furthermore, if one performs block-wise calculations as in Kozub (1986), (i.e., calculate $U(i)$ from $e(i) = \dfrac{\omega^{\circ}(z)}{\delta(z)} U(i)$; see figure 3-7), the conventional implementation would again produce poor performance if the calculations are not properly compensated. The discrepancies between the conventional and the proposed remedy are illustrated in the following section.

Consider a simple IMC controller : $e(i) = (\dfrac{\omega^{\circ}(z)}{\delta(z)}) U(i)$

where
$$\omega^{\circ}(z) = \omega^{\circ}(0) + \omega^{\circ}(1) z^{-1}$$
$$\delta(z) = 1 + \delta(1) z^{-1}$$

Conventional Implementation :

The error, e(i), signal is given by -

$$e(i) = \frac{\omega^{\circ}(0)+\omega^{\circ}(1) z^{-1}}{1 + \delta(1)z^{-1}} U(i)$$

Perform polynomial multiplication :

$$\omega^{\circ}(0) U(i) = e(i) + \delta(1) e(i-1) - \omega^{\circ}(1) U(i-1)$$

Mark the implemented control actions for saturation protection:

$$\omega^{\circ}(0) U(i) = e(i) + \delta(1) e(i-1) - \omega^{\circ}(1) U_{imp}(i-1) \qquad (3\text{-}26)$$

Remedy Implementation :

Separate U(i-1) from U(i) and mark the implemented control actions:

$$e(i) = \omega^{\circ}(0) U(i) + \frac{\omega^{\circ}(1)-\omega^{\circ}(0)\delta(1)}{1 + \delta(1)z^{-1}} U_{imp}(i-1)$$

Perform Polynomial Multiplication :

$$e(i) + \delta(1)\ e(i-1) = \omega^{\circ}(0)U(i) + \omega^{\circ}(0)\delta(1)U(i-1) + (\omega^{\circ}(1)-\omega^{\circ}(0)\delta(1))\ U_{imp}(i-1)$$

Rearrange the equation :

$$\omega^{\circ}(0)\ U(i) = e(i) + \delta(1)\ e(i-1) - \omega^{\circ}(0)\ \delta(1)\ U(i-1) + (\omega^{\circ}(0)\delta(1) - \omega^{\circ}(1)\ )\ U_{imp}(i-1) \qquad (3\text{-}27)$$

The comparison above shows that even with IMC block-wise calculation, the conventional implementation will not be equivalent to the proposed remedy. The terms $\omega^{\circ}(0)\delta(1)U_{imp}(i-1)$ and $\omega^{\circ}(0)\delta(1)U(i-1)$ in equation 3-27 are missing in equation 3-26. If the $U_{imp}(i-1)$'s are equal to $U(i-1)$'s, then the two equations are equivalent. This explains Kozub's (1986) saturation problem where a CMV controller is implemented using the IMC block diagram with conventional clamping on the control action.

3.3. Extension to MIMO Rational Digital Controller

Although the implementation schemes proposed have been developed based on SISO systems, the result is applicable to both SISO and MIMO systems. In a MIMO system, the model based digital controller design would be complicated by the dimension of the system, but the basic optimization is the same (see equation 3-28 and compare with 3-4).

$$\Omega = \min_{\nabla \underline{U}(i)} \sum_{q=1}^{O} \sum_{k=i}^{i+P-1} \alpha_q [Y_{q,sp}(k+b) - Y_q(k+b)]^2 + \sum_{n=1}^{I} \sum_{k=i}^{i+M-1} \beta_n [\nabla U_n(k)]^2 \qquad (3\text{-}28)$$

where P and M are the Output and Input Horizons
O and I are the Output and Input Dimensions

$\nabla \underline{U}(i) \in \mathcal{R}^{MxI}$

One observation from the SISO implementation problem is that the success of a conventionally implemented rational model based digital controller relies on the exact execution of the prescribed control actions. For a MIMO system with interaction, the consequence of override, saturation or actuator failure are more serious. If even one control action from a conventionally implemented multivariate controller cannot be applied, the effect will propagate into the entire control system. The implementation problem in a multi-loop control configuration (i.e., a decoupler with multiple SISO controllers), would be even worse since there will be problems originating from implementation of the single loop controllers and also problems stemming from the interaction of the SISO controllers. Intuitively, one would expect the implementation problem to be more pronounced in MIMO systems. There is very little literature addressing the implementation problem of MIMO systems; Doyle et al. (1987), Hanus et al (1987), Campo et al (1990) are among these rare publications. In Doyle et al. and Campo et al., they pointed out that MIMO saturation problem is further complicated by the "directionality" of the process. In MIMO systems, the process gain is a function of the input direction because the effect of one input can be reinforced or counteracted by the other inputs. Therefore, the actual control vector direction should preserve the original "calculated" control vector direction. Though this work did not incorporate this idea, the remedy can be easily modified to include this extra compensation. If the original vector is $[U_1, U_2, U_3]$, the implemented vector should be $\gamma[U_1, U_2, U_3]$ where γ should be chosen such that none of the control actions, U_i, exceed their limits.

To illustrate the implementation problem in a MIMO system, a simulation study was performed on the extraction distillation model (Appendix B-2) ; the results are shown in Figures 3-9. This simulation will be discussed in greater detail in section 3.3.2. A multivariate Dahlin control was used to illustrate the effect of the implementation problem. When comparing the saturated and the unsaturated responses for the simulated acetone and methanol control loops, there is a dramatic difference in performance. The conventional implementation scheme produces 10 hours of "off specification" product. This situation may be tolerable for certain types of process where post-processing can be performed to salvage the "off specification" product (e.g., blending of the distillate in a refinery). However, in other industries (e.g., pulp and paper), excessive moisture in paper simply cannot be used on a printing machine and the product has to go through a whole cycle of reprocessing. Therefore, it is crucial to devise a method to remedy the implementation problem for rational multivariate digital controllers so as to minimize the deviation from the original design specification.

The *Optimal Solution* to the unconstrained MIMO optimization (equation 3-28) is very similar to the SISO solution equation in 3-9. By differentiation and rearrangement of equation 3-28, the unconstrained *Optimal Solution* is :

$$\left[\frac{K_n(z)}{K_d(z)}\right]R(i) - \left[\frac{T(z)}{\theta(z)}\right] N(i) = \left[\frac{\omega^{\circ}(z)}{\delta(z)}\right] U(i) \qquad (3\text{-}29)$$

For a MIMO system, matrix rational polynomials are used to take into account the increase in the system dimension. Note that the effects of P, M, α and β were lumped together into the approximate model, $\left[\frac{\omega^{\circ}(z)}{\delta(z)}\right]$. Rational MIMO model based controller design software can be used to produce the matrix polynomials, $\left[T(z)\right]$ and $\left[\omega^{\circ}(z)\right]$ (see Harris et al. (1987)). Protection against override, saturation or actuator failure can be obtained by applying the *One-Interval* solution to equation 3-29 :

Step 1 : Separate the past control actions from the present control actions

$$\left[\omega^{\circ}(0)\right]U(i) = \left[\frac{K_n(z)}{K_d(z)}\right]R(i) - \left[\frac{T(z)}{\theta(z)}\right]N(i) - \left[\frac{\tilde{\omega}^{\circ}(z)}{\delta(z)}\right]U(i-1) \qquad (3\text{-}30)$$

$$\text{where} \quad \left[\frac{\omega^{\circ}(z)}{\delta(z)}\right] = \left[\omega^{\circ}(0)\right] + \left[\frac{\tilde{\omega}^{\circ}(z)}{\delta(z)}\right] \qquad (3\text{-}30a)$$

For a 2x2 system, equation 3-30a can be expanded as:

$$\begin{bmatrix} \dfrac{\omega^{\circ}_{11}(z)}{\delta_{11}(z)} & \dfrac{\omega^{\circ}_{12}(z)}{\delta_{12}(z)} \\ \dfrac{\omega^{\circ}_{21}}{\delta_{21}(z)} & \dfrac{\omega^{\circ}_{22}}{\delta_{22}(z)} \end{bmatrix} = \begin{bmatrix} \omega^{\circ}_{11}(0) & \omega^{\circ}_{12}(0) \\ \omega^{\circ}_{21}(0) & \omega^{\circ}_{22}(0) \end{bmatrix} + \begin{bmatrix} \dfrac{\tilde{\omega}^{\circ}_{11}(z)}{\delta_{11}(z)} & \dfrac{\tilde{\omega}^{\circ}_{12}(z)}{\delta_{12}(z)} \\ \dfrac{\tilde{\omega}^{\circ}_{21}(z)}{\delta_{21}(z)} & \dfrac{\tilde{\omega}^{\circ}_{22}(z)}{\delta_{22}(z)} \end{bmatrix}$$

The separation of the matrix rational polynomial in equation 3-30a is the same as that in the single polynomial case (see equation 3-24a). However, the separation has to be performed on the matrix elements one by one.

Step 2 : Use the past implemented control actions and use them in place of the past calculated control actions

$$\left[\omega^{\circ}(0)\right]U(i) = \left[\frac{K_n(z)}{K_d(z)}\right]R(i) - \left[\frac{T(z)}{\theta(z)}\right]N(i) - \left[\frac{\tilde{\omega}^{\circ}(z)}{\delta(z)}\right]U_{imp}(i-1) \qquad (3\text{-}31)$$

Equation 3-31 is the proposed implementation scheme for a rational MIMO model based digital controller.

3.3.1 Optimality of Implementation Scheme

The proposed implementation scheme (equation 3-31) looks very simple. The unanswered question is : how "optimal" is the solution? Equation 3-29 is only optimal for an unconstrained system at time i. When the control action(s) cannot be implemented, the solution will not be optimal. However, the attractive feature of the scheme in equation 3-31 is that, similar to the case of a SISO long horizon controller (LQG), the solution in equation 3-31 makes sure that the past unimplemented control action(s) will not affect the optimality of the new solution. As long as the correct information is used at each control interval, and the new control actions can be implemented, the system will be optimal for that window. This is the same *Moving Window concept* which was used to interpret the optimality of LQG implementation scheme. For a MIMO system it is easier to perform the *One-Interval* implementation (equation 3-31) since a *Multi-Interval Implementation* requires matrix polynomial multiplications which can generate very long polynomials. Thus, the *Multi-Interval* solution is not recommended. The advantage of the *One-Interval Implementation* is that it also provides the disturbance forecasts and the servo trajectories, which are useful system information for process monitoring and diagnosis. From the arguments above, one can see that the *Internal Model Structure* and the *Moving Window Optimization* are the links to the "optimal" implementation of rational MIMO model based digital controllers.

3.3.2 Simulation of an Extraction Distillation Control System

The extractive distillation simulation was used again to verify the proposed rational MIMO digital controller implementation scheme.

Figures 3-9a,b show three responses of a multivariate Dahlin Controller: with no saturation, with *One-Interval Implementation* and with conventional implementation. The conventional implementation in this case is simply to clamp the control action(s) at their limits. The multivariate Dahlin controller has been designed with a deadbeat response on both acetone and menthol control loops. The process model of the extraction distillation (see Appendix B-2) is not invertible so the process model has been modified in order to obtain a stable multivariate Dahlin controller (see equations 3-31, 3-32a,b,c,d).

$$\begin{bmatrix} \text{Acetone Conc.} \\ \text{Methanol Conc.} \end{bmatrix} = 1.0E{-}3 \begin{bmatrix} Gp_{11} Gp_{12} \\ Gp_{21} Gp_{22} \end{bmatrix} \begin{bmatrix} \text{Solvent flow} \\ \text{Steam Flow} \end{bmatrix} \tag{3-32}$$

$$Gp_{11} = \frac{0.9059z^{-2} + 0.2635z^{-3}}{1 - 0.627\ z^{-1}} \tag{3-32a}$$

$$Gp_{12} = \frac{-0.8544z^{-1} - 0.4834z^{-2}}{1 - 0.5712z^{-1}} \tag{3-32b}$$

$$Gp_{21} = \frac{-0.6487z^{-1} - 0.8286z^{-2} - 0.03139z^{-3}}{1 + 0.4474z^{-1} + 0.4082z^{-2}} \tag{3-32c}$$

$$Gp_{22} = \frac{2.769\ z^{-1} + 2.533\ z^{-2}}{1 - 0.6703z^{-1}} \tag{3-32d}$$

The unit of concentration deviates is mole percent and the unit of flow rate deviates is ml/min.

The Multivariate Dahlin Controller for this simulation is :

$$\begin{bmatrix} \omega^{\circ}(0) \end{bmatrix} = \begin{bmatrix} 0.000 & -0.854 \\ -0.649 & 2.769 \end{bmatrix}$$

$$\begin{bmatrix} \dfrac{\tilde{\omega}^{\circ}(z)}{\delta\ (z)} \end{bmatrix} = \begin{bmatrix} \dfrac{0.9059+0.2635z^{-1}}{1.-\ 0.627\ z^{-1}} & \dfrac{-0.9710}{1.\ -\ 0.5712\ z^{-2}} \\ \dfrac{-0.5384\ +\ 0.2334\ z^{-1}}{1.\ +\ 0.4474\ ^{-1}\ +\ 0.4082\ z^{-2}} & \dfrac{4.389}{1.\ -\ 0.6703z^{-1}} \end{bmatrix}$$

since the a deadbeat response is required, $K_n(z)/K_d(z) = 1.0$:

$$\begin{bmatrix} \dfrac{K_n(z)}{K_d(z)} \end{bmatrix} = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$$

Dahlin controller is a servo design technique. The optimal filter, T(z)/θ(z), for an equivalent deadbeat response is (see Harris (1982)) :

$$\begin{bmatrix} \dfrac{T(z)}{\theta(z)} \end{bmatrix} = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$$

Figures 3-9a,b show that the proposed remedy is very close to the unsaturated solution, but it takes the conventional implementation a long time to reach the targets. The 0.02 and -0.02 mole percent setpoint changes in the acetone and methanol targets are typical changes for this type of unit. The relative improvement index, ISE_{scaled}(equation 3-22b), for this case was found to be 10% ; i.e., the proposed scheme reduced the deviation generated by the conventional implementation by 90%. Thus, the implementation scheme has had a big impact on the performance of this MIMO control system. Two more simulations were performed to verify the implementation scheme. Figures 3-10a,b show the responses of an unconstrained LQG ($\alpha = I$ *and* $\beta = 0$) controller which was designed using the full non-invertible process model in Appendix B-2. Notice the big swing in the acetone response produced by the conventional implementation. The ISE_{scaled} was 0.15%; i.e., the proposed implementation reduces the deviation by 99.85%. The tuning effect on the proposed implementation scheme is shown in Figures 3-11a,b. The LQG tuning parameters were :

$$\alpha = \begin{bmatrix} 9.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix} \quad \text{and} \quad \beta = \begin{bmatrix} 2.0\text{E-}6 & 0.0 \\ 0.0 & 20.0\text{E-}6 \end{bmatrix}$$

Since larger penalties were imposed on the process inputs relative to the outputs, smooth input responses would be expected. This can be observed in Figure 3-11a,b. The ISE_{scaled} was again calculated and its value is 1.0%. This is equivalent to a reduction in the conventional deviation by 99.0%. One can conclude from these tests that the proposed implementation scheme is not sensitive to the controller tuning. The ISE_{scaled} for all simulated cases are presented in Figure 3-12. The performance improvement is quite significant in both SISO and MIMO systems, ranging from 39% to 99.9% reduction in the deviation caused by the conventional implementation. The improvement is more dramatic in MIMO systems than in SISO systems.

3.4 Conclusion

This chapter has presented a general implementation scheme for SISO and MIMO rational model based digital controllers. A detailed analysis of the implementation problem has been performed which brings out the fact that the different ways of implementing a rational digital controller do make a difference in the controller performance during periods of manipulated variable saturation, operator override or actuator failure. The analysis also in turn leads to simple remedies for the implementation problems. The key is to separate past contrnl actions from the current control action before any polynomial multiplication, and to use the past implemented control actions instead of the past calculated control actions. The optimality of these remedies can be interpreted using the *Moving Window Optimization* concept. Based on the *Internal Model Structure* and the *Moving Window Optimization*, the remedies for the implementation problems are further generalized to cover long horizon controllers, such as LQG and rational MIMO model based controllers. Computer simulations have been performed to verify the proposed remedies. The results demonstrate the superior performance of these remedies. The amount of computation is no more than the conventional solution, thus these remedies could be easily used to implement rational model based digital controllers.

3.5. Nomenclature

$h(i)$	*Process Impulse weight at time i*
$d(i)$	*Process Disturbance weight at time i*
$Y(i)$	*Process Model Output at current time i*
$Y_m(i)$	*Process measurement at current time i*
$U(i)$	*Process Input at current time i*
$e(i)$	*Error Signal in IMC Structure*
$U_{imp}(i)$	*Actual Implemented Input at current time i*
$N(i)$	*Noise Disturbance at current time i*
	$N(i) = Y_m(i) - \dfrac{\omega(z)z^{-b}}{\delta(z)} U_{imp}(i)$
$a(i)$	*Random Shock at current time i*
$Y_{sp}(i)$	*Set point Trajectory at time i*
$Y(i+b)$	*Process Output Forecast at time i+b*
$Y_N(i)$	*Filtered Disturbance at time i*
	$Y_N(i) = \dfrac{T(z)}{\theta(z)} N(i)$
$Y_U(i)$	*Intermediate output*
$N(i+b)$	*Noise Disturbance Forecast at time i+b*
$Y_{sp}(i+b)$	*Set point Trajectory Forecast at time i+b*
$R(i)$	*Set point Input at time i*
$G_R(z)$	*Rational Process Model ;* $G_R(z) = \dfrac{\omega(z)z^{-b}}{\delta(z)}$
$\omega(z)$	*Numerator of the rational process model*
$\delta(z)$	*Denominator of the process model*
	$\delta(z) = 1 + \delta(1)z^{-1} + \delta(2)z^{-2} + \dots$
$\theta(z)$	*Numerator of noise model -Box & Jenkins Form*

$\phi(z)$	Denominator of noise model
∇	Differencing operator, ($1 - z^{-1}$)
$\omega^{\circ}(z)$	Approximate process model numerator
$Or(G(z))$	Order of Polynomial $G(z)$
$\sim(G(z))$	~ implies $G(z)$ with first term removed; $G(z) = G(0) + G(1)z^{-1} + G(2)z^{-2}$ $\sim G(z) = \quad G(1)z^{-1} + G(2)z^{-2}$
α	Output Trajectory Deviation penalty
β	Input Deviation Penalty
d	Differencing operator
P	Output Horizon in Optimization
M	Input Horizon in Optimization
I	Dimension of process input
O	Dimension of process output
$\psi_f(z)$	Forecast error polynomials
$T(z)$	Forecast related polynomials $\dfrac{\theta(z)}{\phi(z)\nabla} = \psi_f(z) + \dfrac{T(z)}{\theta(z)}$
$K_n(z)$	Numerator of Servo Trajectory
$K_d(z)$	Denominator of Servo Trajectory $K_d(z) = 1 + K_d(1)z^{-1} + \ldots.$
$Y_b(i), U_b(i)$	Process Output/Input if control action can be exactly implemented
$Y_c(i), U_c(i)$	Process Output/Input using conventional implementation
$Y_o(i), U_o(i)$	Process Output/Input using One-Interval/ Multi-Interval implementation

$\left[\frac{\omega(z)}{\delta(z)}\right]$ *matrix rational polynomial - Wilson canoical form*

$$\left[\frac{\omega(z)}{\delta(z)}\right] \equiv \begin{bmatrix} \frac{\omega_{11}(z)}{\delta_{11}(z)} & \frac{\omega_{12}(z)}{\delta_{12}(z)} \\ \frac{\omega_{21}(z)}{\delta_{21}(z)} & \frac{\omega_{22}(z)}{\delta_{22}(z)} \end{bmatrix}$$

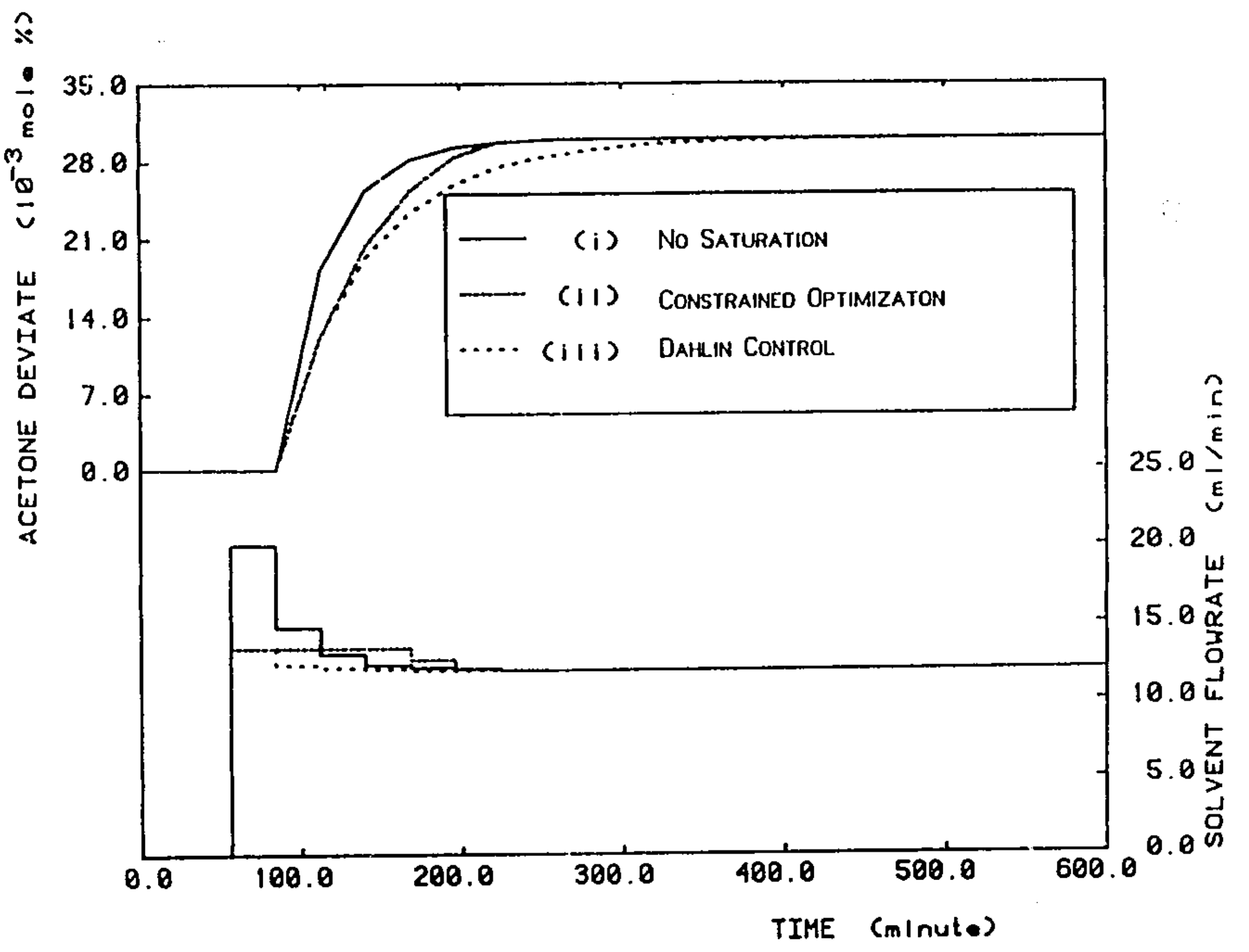


Figure 3-1: Conventional Implementation of Dahlin Controller (SISO system)

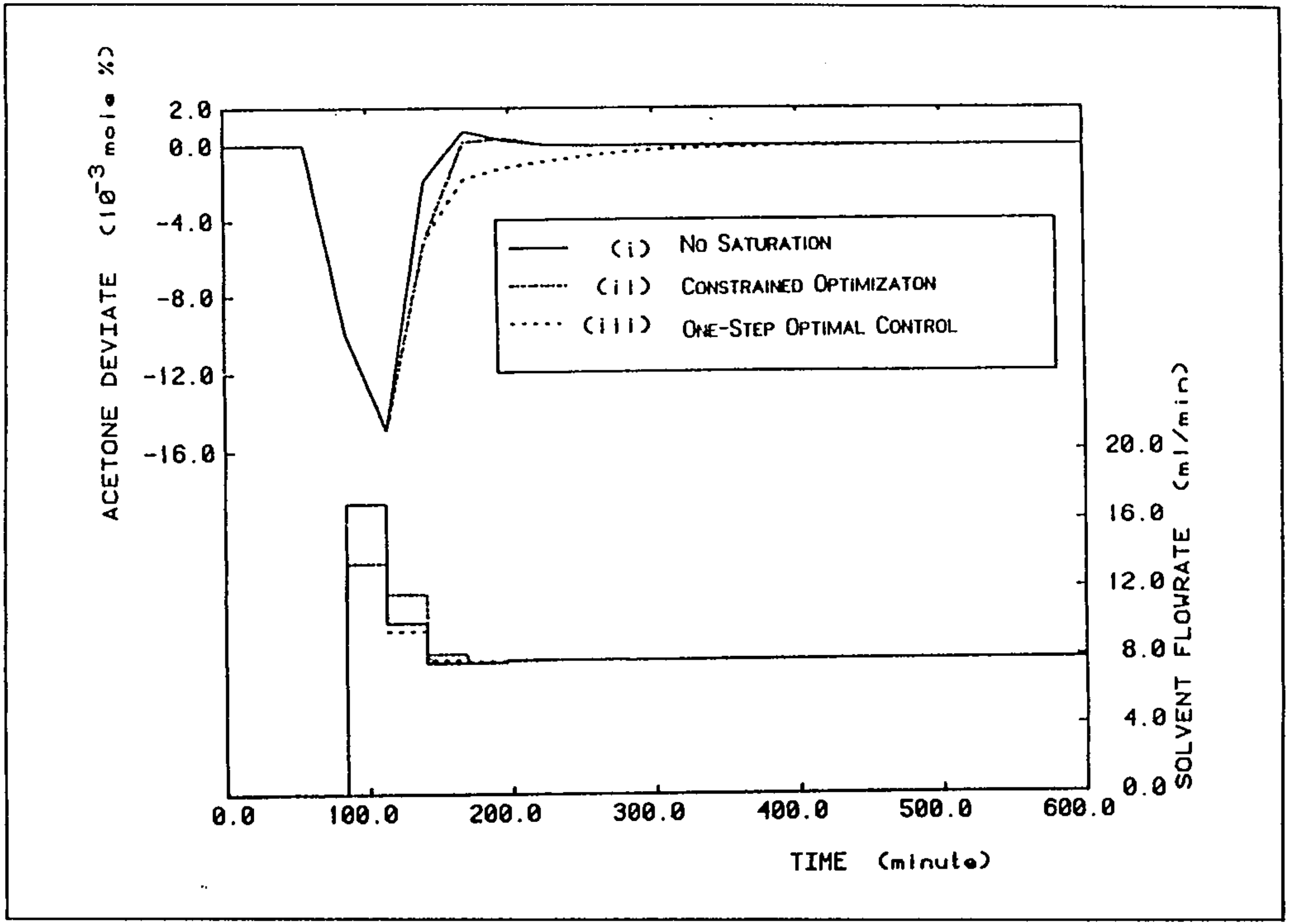


Figure 3-2: Conventional Implementation of One-Step Optimal Controller (SISO system)

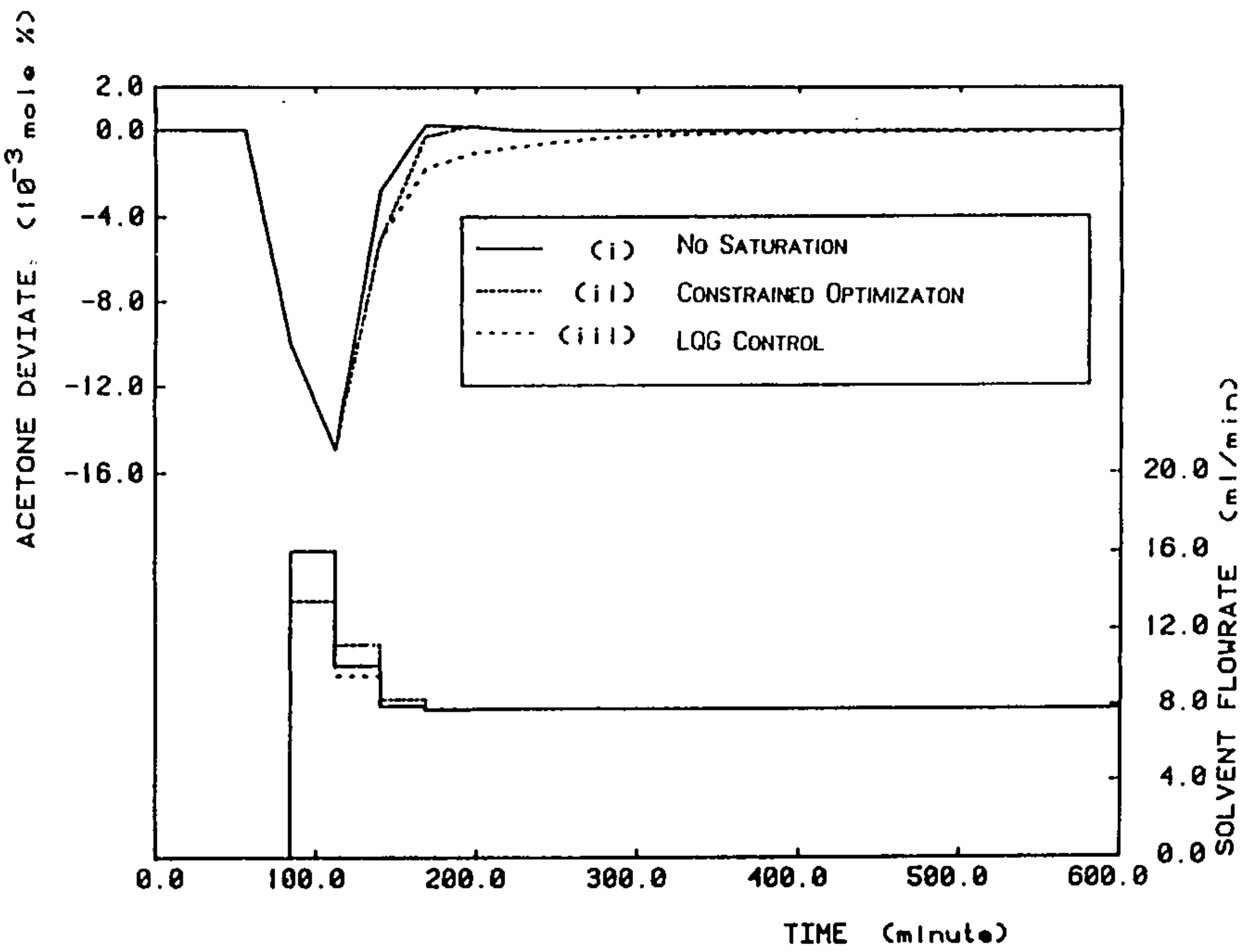


Figure 3-3: Conventional Implementation of LQG Controller (SISO system)

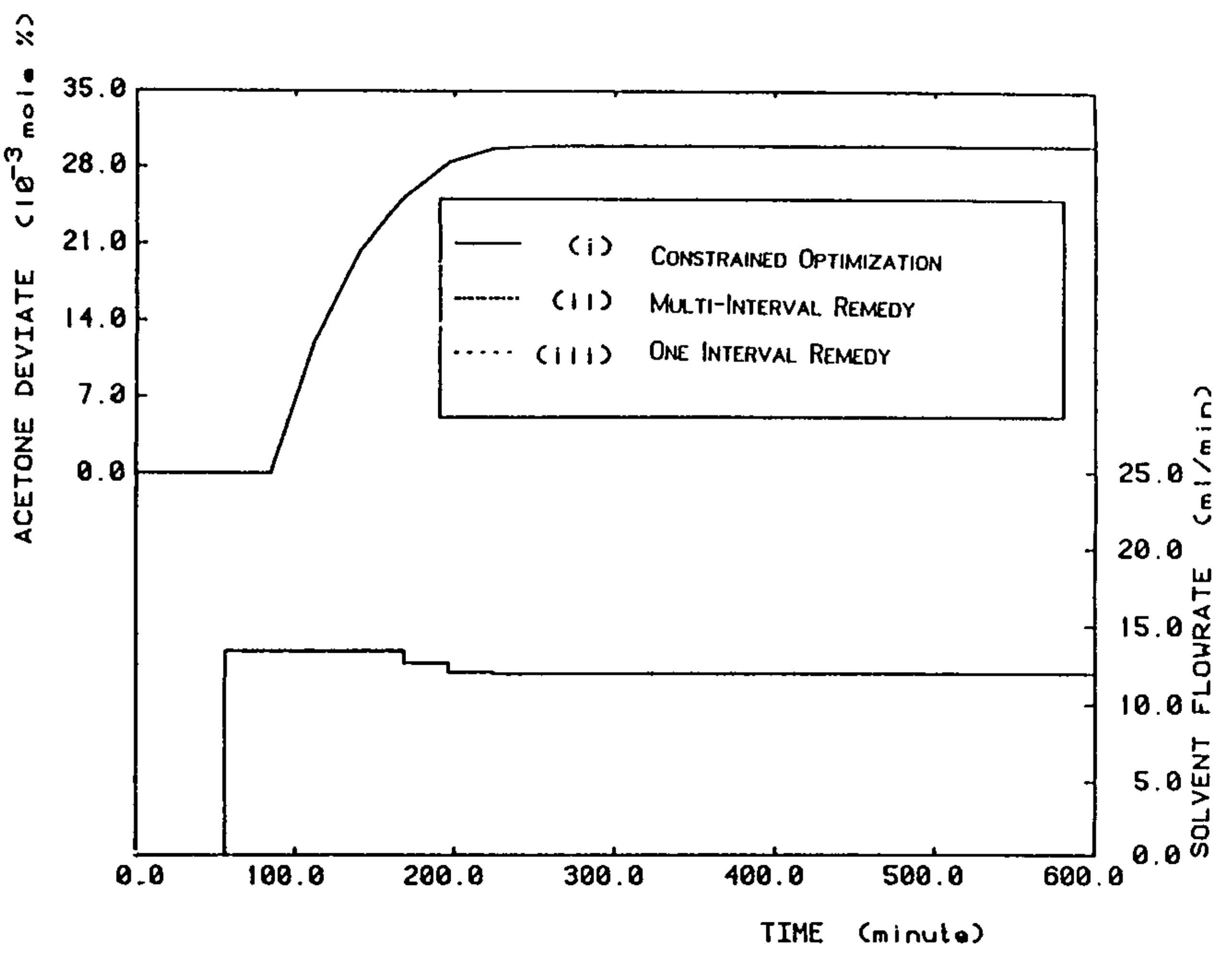


Figure 3-4: Optimal Implementation of Dahlin Controller (SISO system)

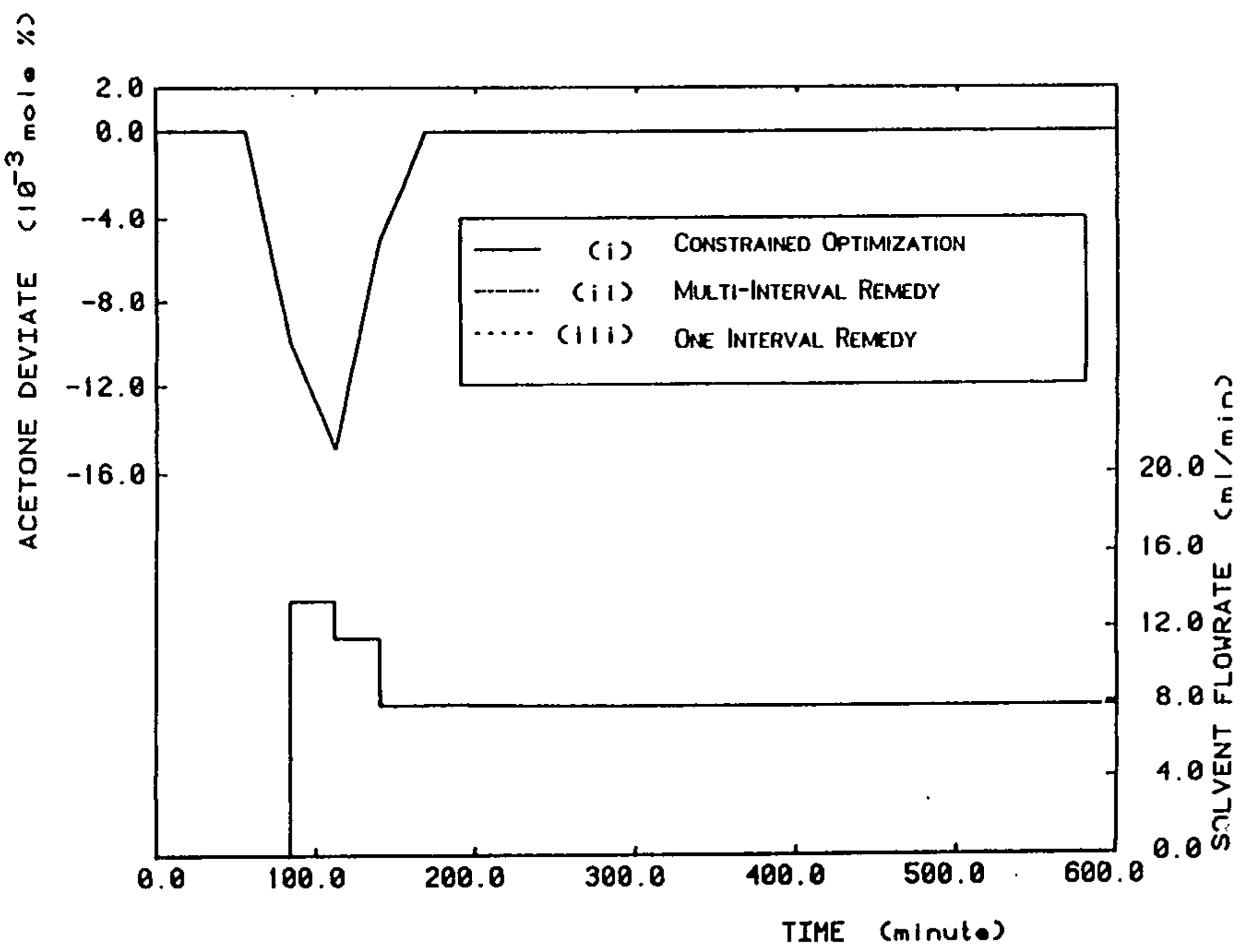


Figure 3-5: Optimal Implementation of One-Step Optimal Controller (SISO system)

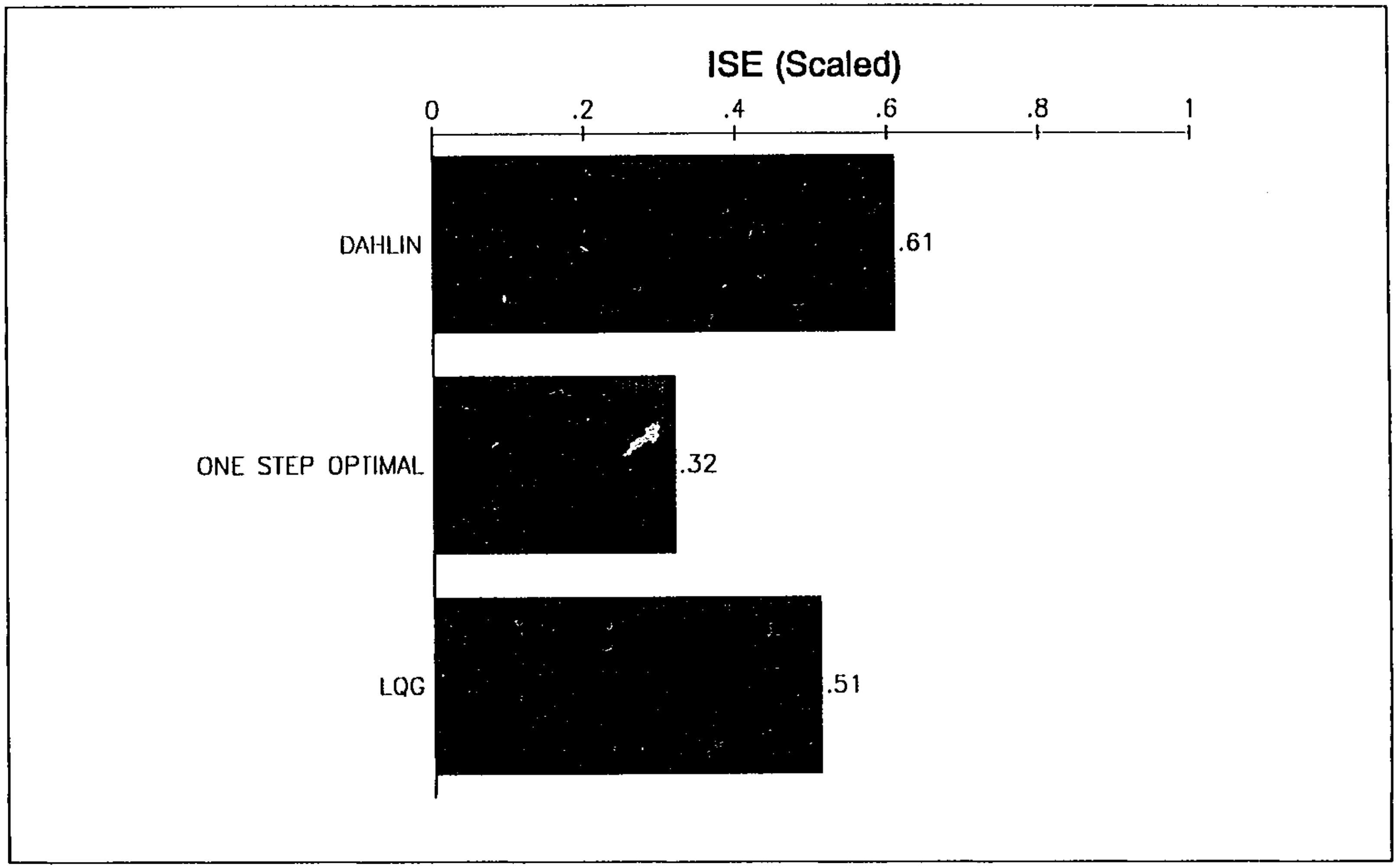


Figure 3-6: Relative Performance of Implementation Schemes (SISO system)

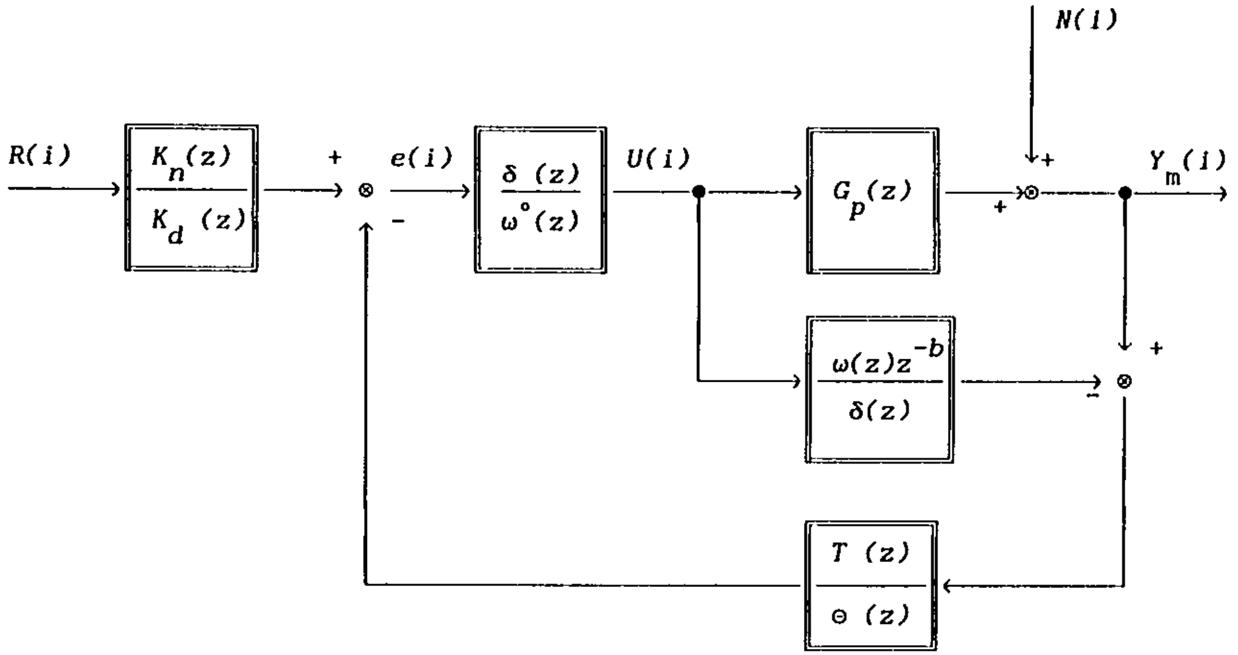


Figure 3-7: Internal Model Structure Block Diagram

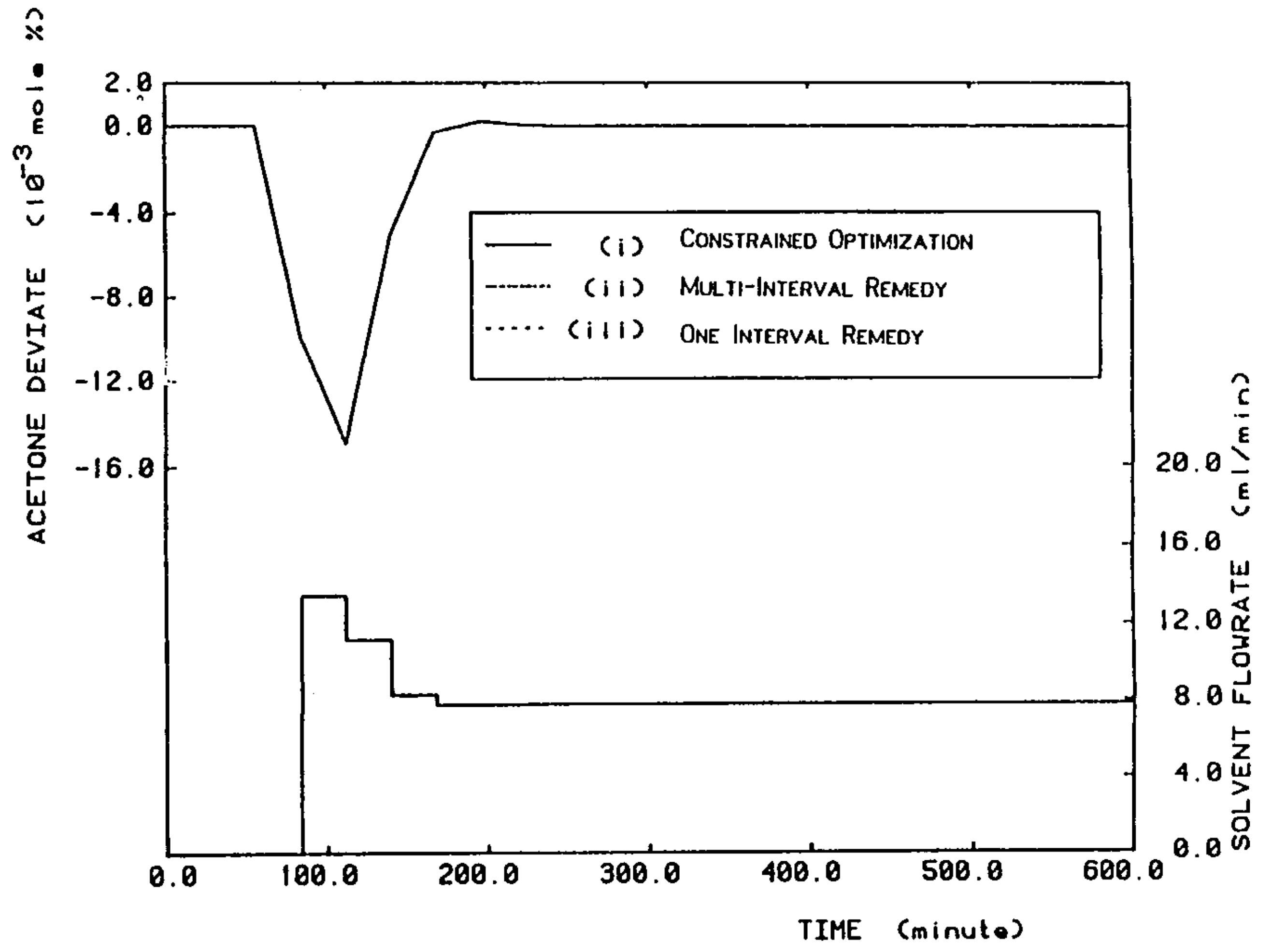


Figure 3-8: Optimal Implementation of LQG Controller (SISO system)

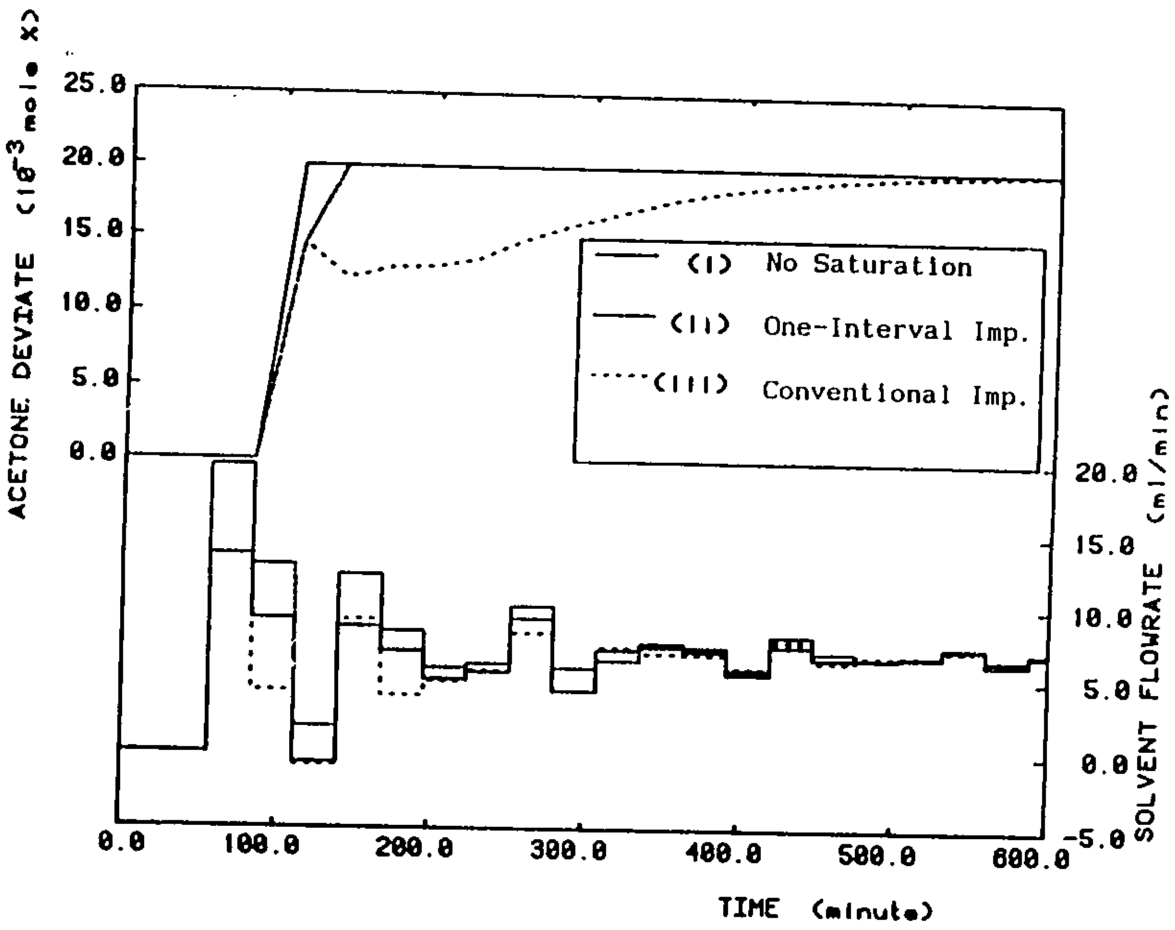


Figure 3-9a: Optimal Implementation of Dahlin Controller (MIMO system)

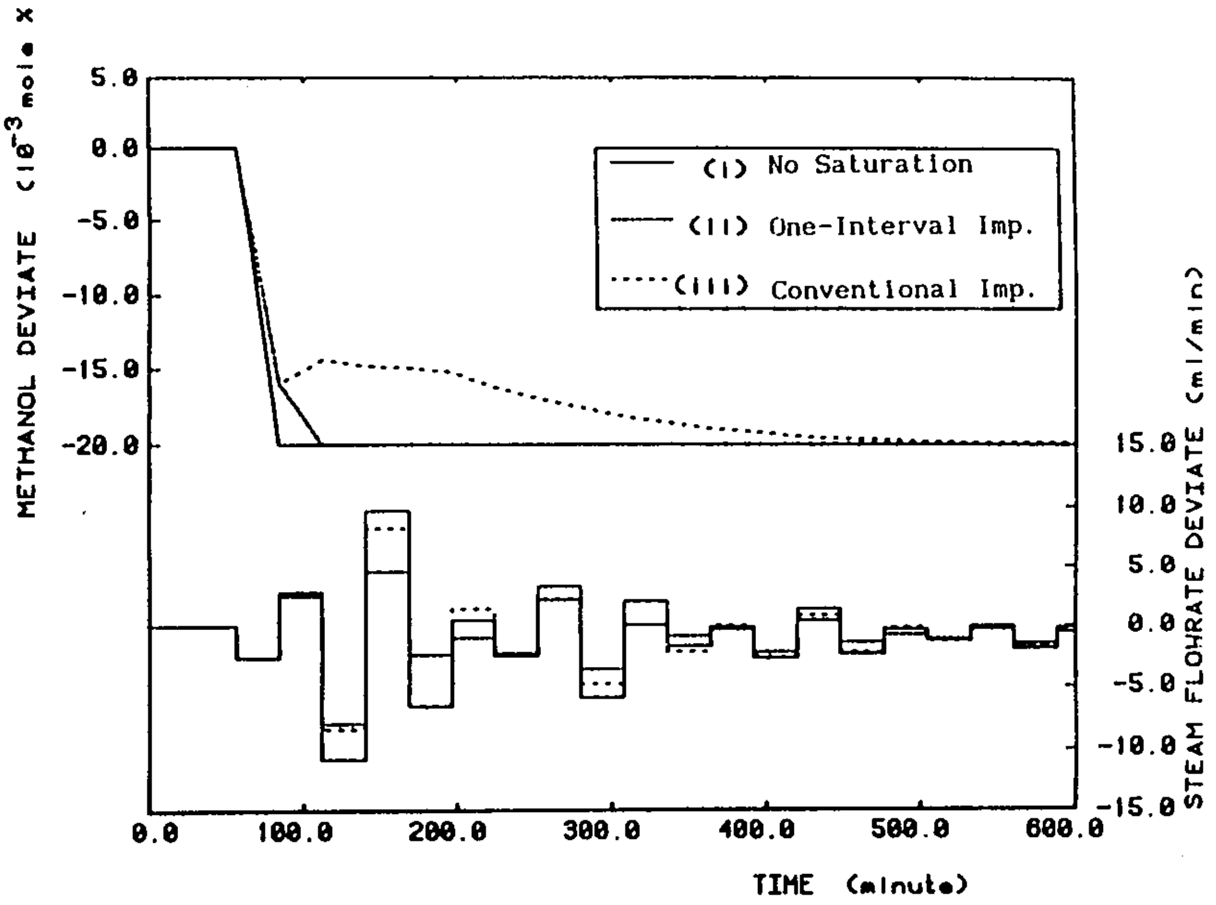


Figure 3-9b: Optimal Implementation of Dahlin Controller (MIMO system)

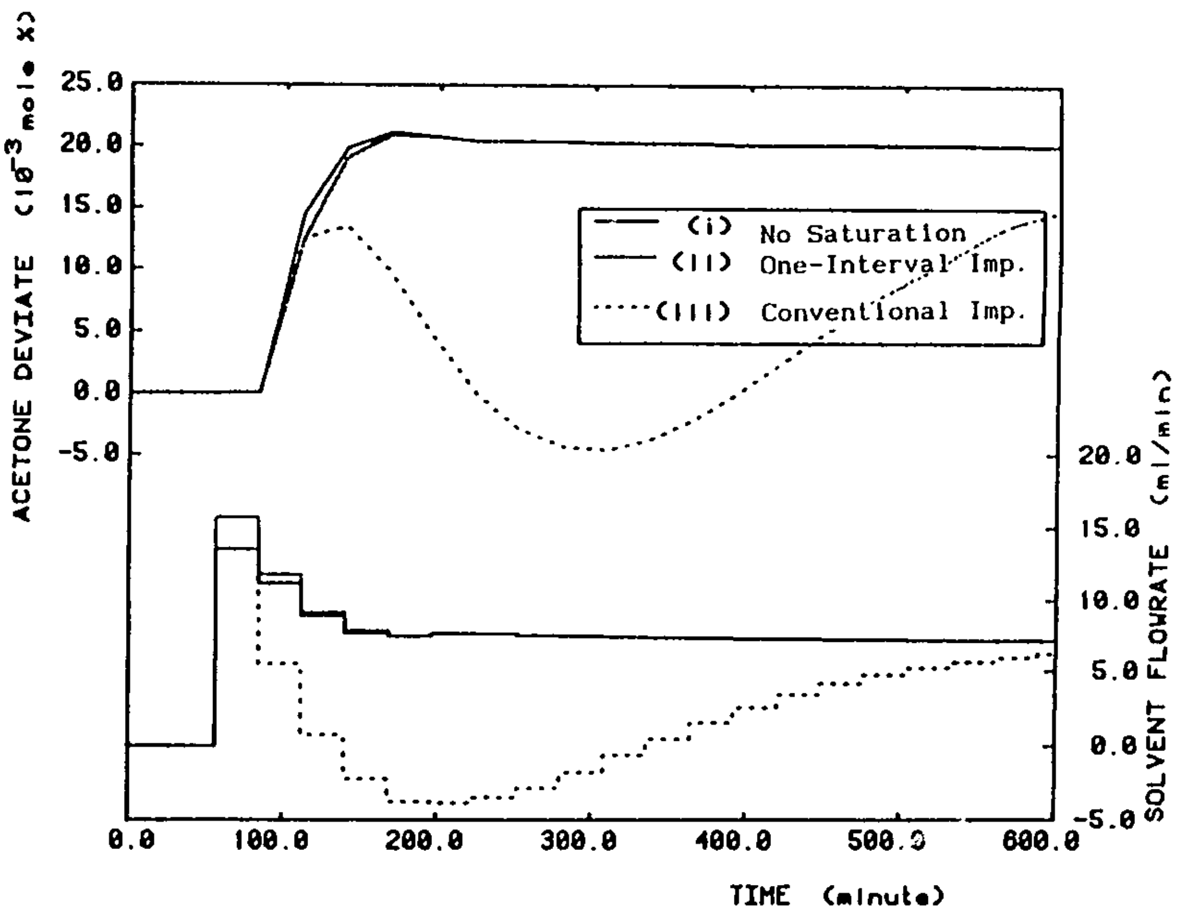


Figure 3-10a: Optimal Implementation of LQG Controller (MIMO system : $\alpha = 1$, $\beta = 0$)

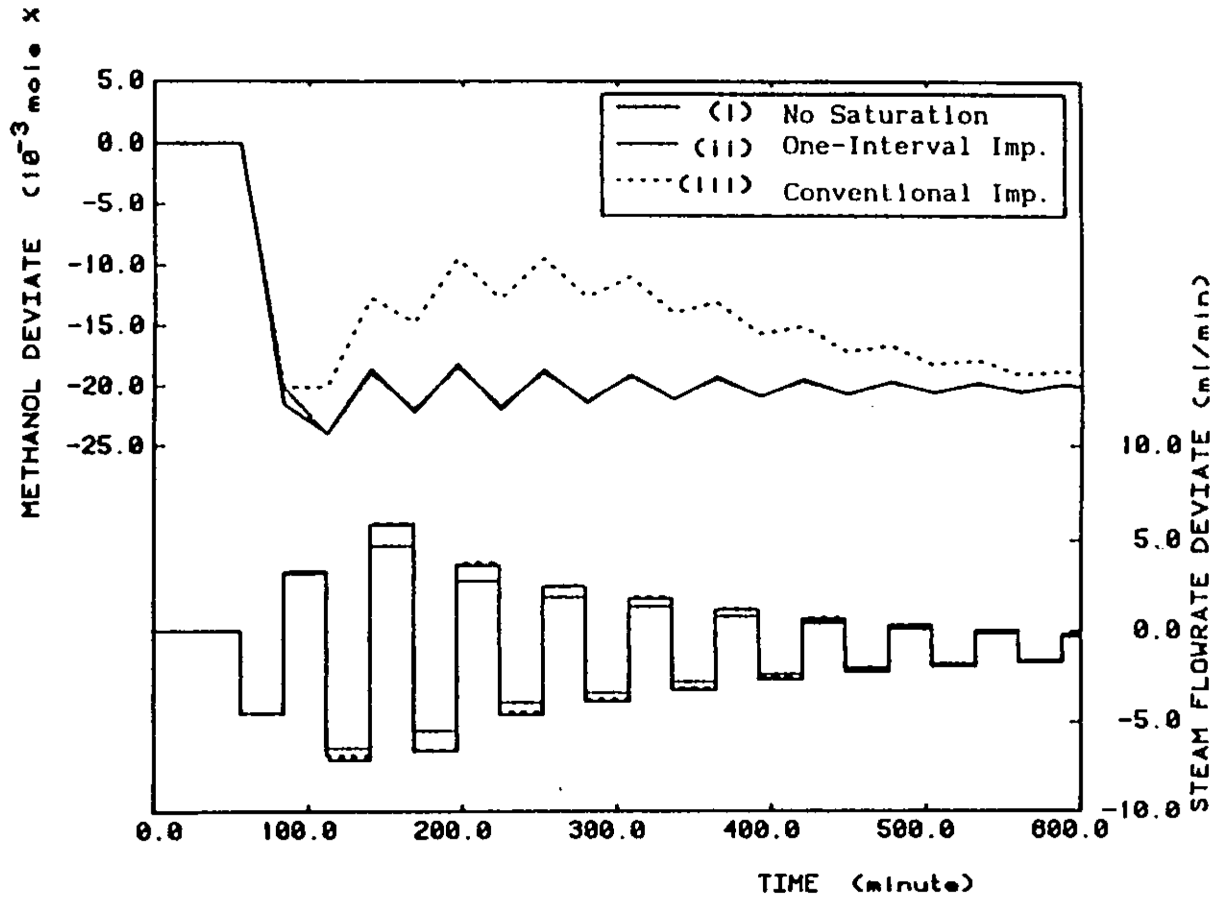


Figure 3-10b: Optimal Implementation of LQG Controller (MIMO system : $\alpha = 1$, $\beta = 0$)

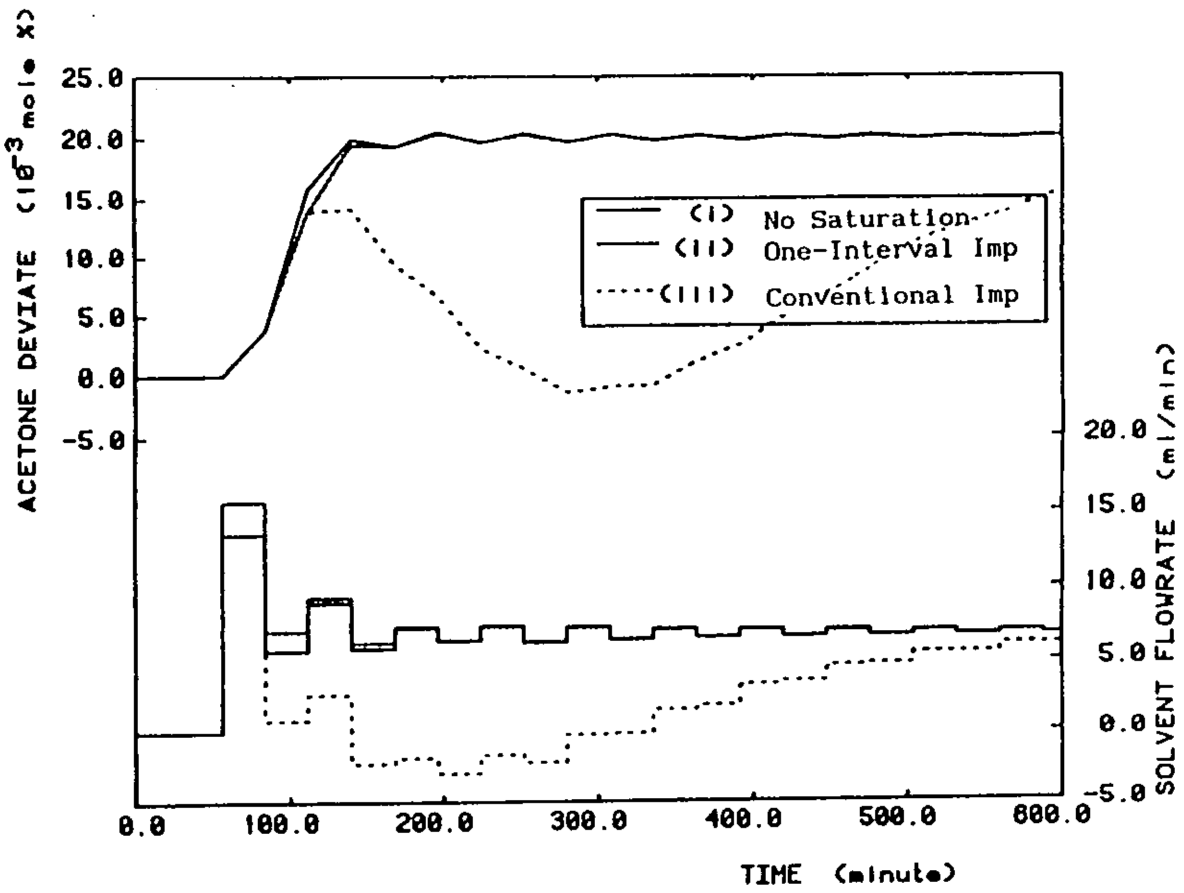


Figure 3-11a: Optimal Implementation of LQG Controller (MIMO system : $\alpha = \begin{bmatrix} 9 & 0 \\ 0 & 1 \end{bmatrix}$, $\beta = \begin{bmatrix} 2e\text{-}6 & 0 \\ 0 & 2e\text{-}6 \end{bmatrix}$

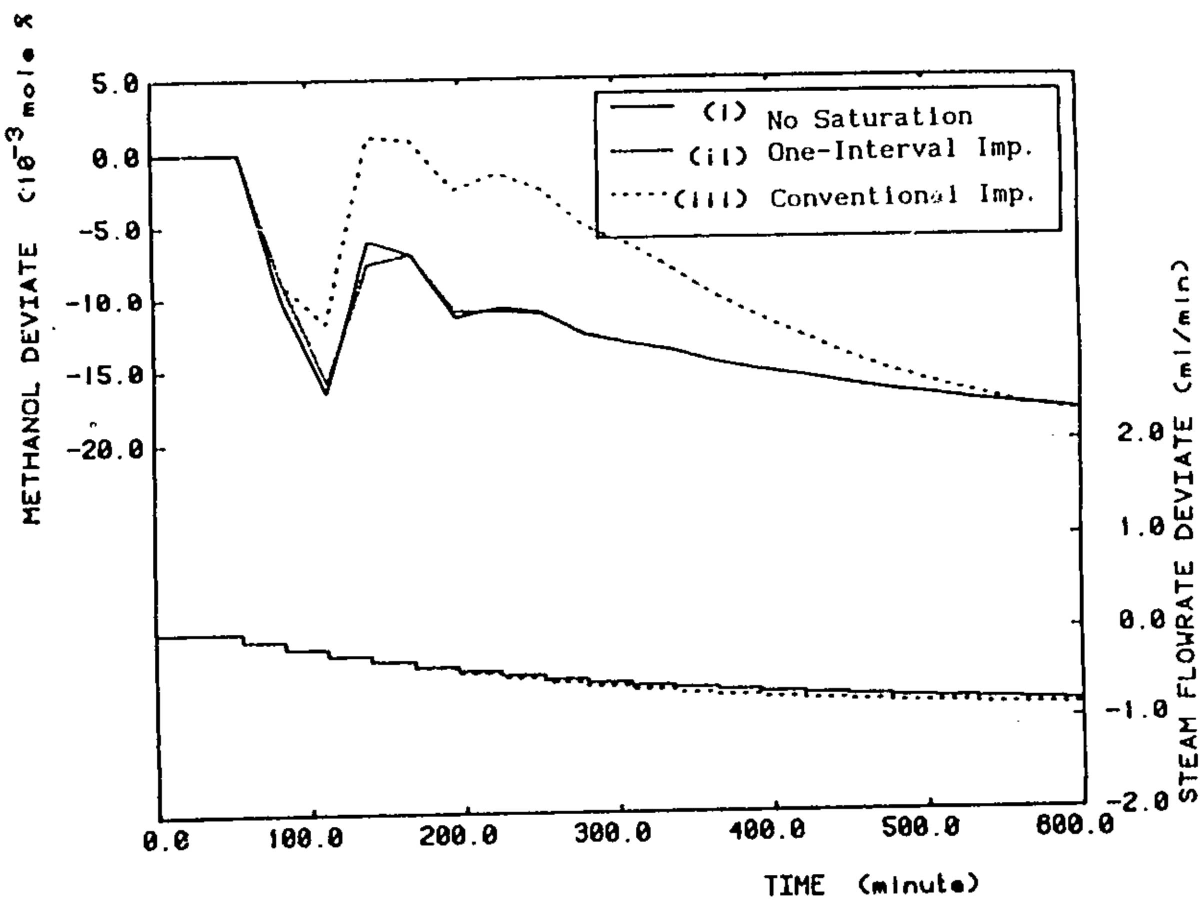


Figure 3-11b: Optimal Implementation of LQG Controller (MIMO system : $\alpha = \begin{bmatrix} 9 & 0 \\ 0 & 1 \end{bmatrix}$, $\beta = \begin{bmatrix} 2e\text{-}6 & 0 \\ 0 & 2e\text{-}6 \end{bmatrix}$

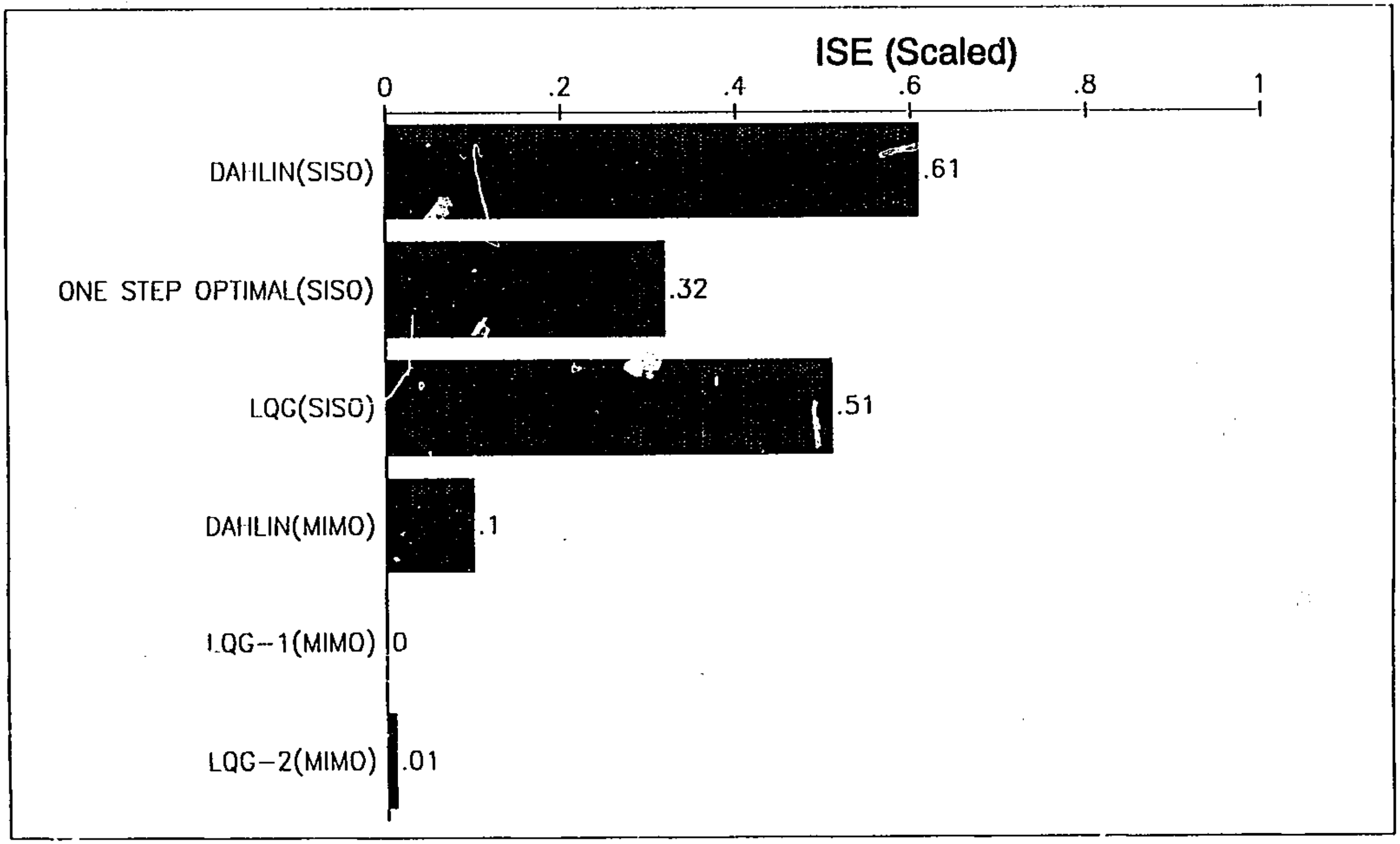


Figure 3-12: Relative Performance of Implementation Schemes (MIMO and SISO systems)

CHAPTER 4

ROBUSTNESS - STABILITY OF UNCERTAIN SYSTEM

4.1 Introduction

In the last two chapters, the implementation of model based controllers has been addressed. The heart of these model based controllers is a model of the process that they are to control. Naturally, the success of these controllers would ultimately depend on the correctness of the process model. If the model is not adequate, the resulting performance can be very different from the design specification as demonstrated in the simulation and experimental studies in Wong et al. (1987a) and Kusuma et al. (1984). This dependency on model adequacy is commonly known as the "robustness problem" or the "model mismatch problem" in the control literature (see Bartlett et al. (1987), Doyle (1982), Huang et al. (1987), Palazoglu et al. (1986), Bequette et al. (1987), Morari et al (1989), Skogestad et al. (1990), Campo et.al (1990), Khambanonda et al (1990)). With the increasing use of model based controllers, this issue has become a very active research area. Although model based control theory promises significant improvement over classical control techniques, the improvement cannot be realized if the process model is not an "adequate" description of the real process.

Stability is one of the important issues in the robustness problem. It dictates the ultimate behavior of a controlled system. Although many measures, such as gain margin, phase margin, etc, have been developed to check the degree of safety designed into a controlled system, these classical techniques are based on the model of an exactly known process. In other words, if the process dynamics are known, these techniques can analyze the closeness of the control system to instability. However, for control systems with varying or uncertain dynamics, these measures cannot be straight forwardly applied and modifications have to be made.

There are several methodologies to assess the stability margin of an uncertain control system. These uncertainties can be attributed to simplification of a complex physio-chemical process, changes in process operating region, etc. In general, the dynamics of an uncertain process can be modeled by a nominal model (Gm(z)) and an uncertainty (Un(z)) about the nominal model (see equations 4.1a,b and Lunze (1982)). The stability of the controlled system can be assessed by analyzing a characteristic equation similar to equation 4.1b.

$$Gp(z) = Gm(z) + Un(z) \tag{4.1a}$$

$$G_{cl}(z) = 1 + Gc(z)Gp(z) = 0 \tag{4.1b}$$

where Gc(z) is the controller

In many model based controller designs, a simplified nominal model, Gm(z), is used in the design. Typically a 1st or 2nd order plus dead time model is used and the model form can be described in a variety of mathematical domains : frequency, input/output transfer function, state space, etc. The parameters in Gm(z) are assumed to be known and fixed and the uncertainty of the controlled system lies in Un(z). There are many ways to describe this uncertainty. Doyle et al. (1981), Horowitz (1982), and others use magnitude uncertainty in the frequency domain. Heinen (1985) and Yedavalli (1986) treat the uncertainty in the state space domain with the uncertainty specified through the parameters in the system matrices. Kharitonov(1978), Bialas et al.(1983), Barmish(1984), Soh et al.(1989) tackle the problem from the Input/Output transfer function space. Each type of uncertainty characterization has its own merit and provides a way to handle the stability investigation for that particular space. It is important to note that transferring uncertainty from one space to another usually introduces conservativeness in the analysis. Therefore, it is better to use a stability analysis which is compatible with the uncertainty characterization; i.e., if the uncertainty is specified in the frequency domain, then one should analyze the uncertain system using a frequency approach. The work here focuses on the Input/Output transfer function space. This takes into account the fact that control engineers in the processing industries tend to be more familiar with input/output data (step test, pulse test, etc.). Therefore, the Input/Output transfer function space approach is easier to understand. Furthermore, the Input/Output space is closely related to the time domain and consequently the uncertain system's performance can be assessed through its time domain response. For a chemical engineer, this approach is more natural than the bandwidth, resonant peak concepts used in the frequency domain.

Since an approximate model is typically used in the design of model based controllers, there will be discrepancies (uncertainty) between the model and the actual process. If one is not careful in assessing this process uncertainty, the more ambitious model based controller designs will probably cause problems rather than delivering the expected high performance. Fortunately, the discrepancy between the actual process and the model is not totally unknown. One can have an idea as to the extent of the process uncertainty by doing repeat step tests or pulse tests. In fact, this uncertainty should be part of the process description and the controller design procedure should incorporate this information so that the resulting controller can perform well even in the face of process uncertainty.

This chapter addresses the stability issue of the robustness problem. A formal problem definition is presented in sections 4.2 together with some background information on the stability of an uncertain control system. Novel stability criteria are discussed in section 4.2.2.1 and examples are shown in section 4.2.2.4. Section 4.3 presents a discussion of the application of these new results to the design of model based controllers. A conclusion of this chapter is given in section 4.4

4.2 Stability of Uncertain Systems

Throughout this chapter, polynomials are used to assess control system stability. In general polynomials are closely related to system stability through the closed loop characteristic equation (equation 4.1b). If one can specify the transfer functions for the controller and the process then a closed loop characteristic equation similar to equation 4.1b can be derived and this dictates the stability of the control system.

In the investigation of the stability of an uncertain system using the input/output transfer function space, there are two main categories of uncertainty - polytopic and interval uncertainty. Although this work focuses on the latter type of uncertainty, both types are discussed so that the nature of the uncertainty description can be appreciated and an appropriate uncertainty analysis can be selected. In this way, the conservativeness of the result can be reduced.

4.2.1 Stability of an Uncertain Polytopic System

Consider the following family of polynomials :

$$P_r(z) = r_1P_1(z) + r_2P_2(z) + \dots + r_nP_n \qquad (4.2)$$

where

$$r = (r_1, r_2, \dots, r_N) \in R^N; \; r_i \geq 0 \text{ for all } i; \; \sum_{i=1}^{N} r_i = 1$$

$P_r(z)$ represents a polytope of polynomials. This type of system usually occurs when the uncertainties in process model parameters are correlated. When r_i sweeps from 0 to 1, it spans an infinite number of polynomials. The objective is to determine the stability of this infinite polynomial set. For a discrete system, this is equivalent to determining whether all the possible roots of equation 4.2 have magnitudes less than unity.

The major breakthrough in the stability analysis of polytopic polynomial systems was presented by Huang et al. (1987) and Bartlett et al. (1987). They gave a rigorous mathematical proof using the theory of topology. Their result shows that the stability of the family of polytopic polynomials can be ascertained by checking the "exposed edges" - pair wise convex combinations of polynomials, $\{(1-r)P_i(z)+rP_j(z);\ r\in[0,1];\ (i,j)\in(1,2,..,N)\}$. This checking only involves one free parameter, $r\in[0,1]$ and hence the computation can be readily performed. Katbab et al.(1990) further generalizes Bartlett et al.'s "Edge Theorem" and presents a more efficient computation method to check polytopic uncertain system. Although Bartlett et al.'s mathematical result seems very interesting, the following example shows that their result is non-trivial to implement in practice. The problem lies in the degree of correlation between the process parameter uncertainties.

EXAMPLE :

Consider a simple 1st order plus dead time process under feedback control (Gc), the characteristic equation is:

$$P(z) = 1 - \delta z^{-1} + Gc\ Kp\ (1-\delta)\ z^{-2} \qquad (4.3)$$

where δ related to the process time constant ($e^{-T/\tau}{}_p$)

T sampling time

τ process time constant

G feedback controller

K process gain

F unit throughput

If the process gain and the process time constant are related to the process throughput, F, in the following manner :

$$Kp = f(F) \quad (4.4a)$$

$$\delta = g(F) \quad (4.4b)$$

Then the characteristic polynomial in equation 4.3 can be written as :

$$P(z) = 1 - g(F)\ z^{-1} + Gc\ f(F)(1-g(F))\ \ z^{-2} \quad (4.5a)$$

Equation 4.5a contains only F and hence is one-dimensional. If the uncertainty of F is :

$$F_1 \quad < \quad F \quad < \quad F_2 \quad (4.5b)$$

The two extreme polynomials are :

$$P_1(z) = 1 - g(F_1)z^{-1} + Gc\ f(F_1)\ (1-g(F_1))\ z^{-2} \quad (4.6a)$$

$$P_2(z) = 1 - g(F_2)z^{-1} + Gc\ f(F_2)\ (1-g(F_2))\ z^{-2} \quad (4.6b)$$

The result of Bartlett et al. (1987) basically proves that the stability of equations 4.5a,b can be assessed by checking the exposed edge; i.e.,

$$P(z) = (1-r)\ P_1(z) + rP_2(z) \qquad \forall \ 0 \le r \le 1 \tag{4.7}$$

Equations 4.5 and 4.6 show that the relations between the process throughput(F) and Kp and δ reduce the uncertain system to a one dimension system. As the throughput changes from F_1 to F_2, the characteristic equation changes from equation 4.6a to 4.6b. In this case, a one-dimensional search (checking the exposed edge) is adequate to assess the stability of this type of uncertain system. The assessment is computationally efficient and more importantly, the result is non-conservative. However, in a real process, it is not trivial to define an uncertainty correlation similar to the one shown in equation 4.4. If there happens to be little correlation between the polynomial coefficients, then a different approach has to be employed.

4.2.2 Stability of an Uncertain Interval System

A more attractive approach for processes with uncorrelated uncertainty is to describe the uncertainty in terms of "interval" polynomials.

Consider a discrete interval polynomial (see equations 4-8,4-9):

$$F(z) = z^n + (\alpha_{n-1}+\beta_{n-1})z^{n-1}+....+(\alpha_1+\beta_1)z^1 + (\alpha_0+\beta_0) = 0 \tag{4-8}$$

where the α_k's are the nominal, fixed coefficients of the polynomial, and the β_k's represent the perturbations in the coefficients. These perturbations can take any values within the symmetric bounds:

$$-\bar{\beta}_k \le \beta_k \le \bar{\beta}_k \qquad \text{and} \qquad \bar{\beta}_k \ge 0 \qquad k = 0,\ 1 \ldots n-1 \qquad (4\text{-}9)$$

where $\bar{\beta}_k$ and $-\bar{\beta}_k$ are the upper and lower bounds of β_k

so that the perturbed system, F(z), contains an infinite number of polynomials. The main concern here is to check the stability of F(z) when the bounds, $\bar{\beta}_k$, are known together with the nominal coefficients, α_k's. Wong et al. (1987b) show that this mathematical problem has a direct application in the study of an uncertain control system: if one can use historical operating records and operational experience to estimate the uncertainty in the discrete impulse response as a bound similar to equation 4-9, then the response of the closed loop control system will have a characteristic equation similar to equations 4-8 and 4-9. Therefore, the stability of the closed loop uncertain system is given by the stability of equation 4-8.

Although there are many stability criteria in the literature (e.g., frequency criteria from the Nyquist theorem, root location from eigenvalue theory), they are not suitable for perturbed coefficients systems because one needs to check the stability of an infinite number of polynomials; this is practically impossible. In the search for necessary and sufficient conditions which guarantee the stability of the interval polynomials, the celebrated work by Kharitonov (1978) was a tremendous step forward for continuous interval polynomials. The necessary and sufficient conditions for the roots of the perturbed system (equations 4-8 and 4-9) to lie on the left hand complex plane, ($\mathrm{Re}(\lambda_k) < 0.0$) requires the checking of four polynomials rather than the infinite set (see Mori and Kokame (1987), Bose and Shi (1987) and Barmish(1984)). This theorem has had a significant impact on the study of continuous uncertain systems and Barmish and De Marco (1987) have provided a perspective of the application of Kharitonov's Theorem to continuous, perturbed coefficient polynomials and matrices. However, these results cannot be directly applied to discrete interval polynomials where stability requires $|\tau_i|<1.0$ (see Hollot and Bartlett (1986) and Bose and Zeheb (1986)). There are many conjectures on the stability characteristics of interval systems. However, many of these conjectures were either proven wrong or shown incorrect with counter-examples. For examples, the works of Bialas (1983), Argon (1986), Juang et al. (1989) were disproved by Karl et al. (1984), Soh C.B.(1990) and Shi (1990) respectively. As Bartlett, the leading investigator in this area, points out in Bartlett (1990) that "In general, non-conservative stability analysis can be extremely difficult to carry out". Huang et al. (1987) used topology arguments to show that the stability of discrete interval polynomials cannot be inferred by extreme polynomials in general. They propose further restrictions on equations 4-8 and 4-9 so that the necessary and sufficient conditions for the stability of the restricted discrete interval polynomial could be obtained. However, these restrictions are too severe to be useful for the study of control systems with uncertain parameters.

Other studies (Berger (1982); Mori (1984); Dabke (1983); Heinen(1985)) have tackled this problem by considering the location of the roots of the perturbed system equation. The roots of F(z)=0 satisfy $|z| \le \delta$ and $\delta > 0$ if

$$\sum_{k=1}^{n} (|\alpha_{n-k}| + \bar{\beta}_{n-k}) / (\delta^{n-k}) \le 1 \qquad (4\text{-}10)$$

For the stability of a discrete system, δ is 1 and equation 4-10 becomes:

$$\sum_{k=1}^{n} |\alpha_{n-k}| + \bar{\beta}_{n-k} \le 1 \qquad (4\text{-}11)$$

Equation 4-11 can also be obtained immediately from Jury's Dominance Coefficients theorem (see Jury (1964)). Equation 4-11 is a conservative stability criteria, in the sense that it becomes inconclusive when the true process is still stable, because equation 4-11 is a necessary but not sufficient condition for stability. Xi and Schmidt(1985) suggested the ratios and signs of α_k's can be used together with the Monotonic and Dominance conditions in Jury (1964) so that a less conservative criteria can be achieved. They proposed two new theorems beyond equations 4-10 and 4-11 based on the ratios of α_k's. However, the perturbed coefficient theorem that they proposed can not be applied to interval type of perturbation.

This study proposes three criteria for the stability of discrete interval polynomials. They are all based on the Rouche's theorem and they make use of particular properties of α_k's and β_k's. The objective is to provide criteria which are less conservative than the other methods. They should not only be computationally feasible, but also straight forward to apply.

In order to avoid checking the stability of an infinite number of polynomials in equations 4-8 and 4-9, Rouche's Theorem (Rosenbrock and Storey (1970)) is used. This theorem provides a comparison technique to count the number of zeros of an analytic function in a certain domain. For this problem, the domain of interest is the unit disk centered at the origin. Three criteria will be derived in the following sections. They are all based on the Rouche's Theorem.

Given the interval polynomial, F(z) in equations 4-8 and 4-9, it can be written as :

$$F(z) = \Phi(z) + \Xi(z) \tag{4-12}$$

where the certain portion is:

$$\Phi(z) = z^n + \alpha_{n-1} z^{n-1} + \ldots + \alpha_1 z^1 + \alpha_0 \tag{4-13}$$

and the uncertain portion is :

$$\Xi(z) = \beta_{n-1} z^{n-1} + \ldots\ldots + \beta_1 z^1 + \beta_0 \tag{4-14}$$

with symmetric bounds on the value of the perturbations

$$-\bar{\beta}_k \leq \beta_k \leq \bar{\beta}_k \qquad \text{and} \qquad \bar{\beta}_k \geq 0 \qquad (4\text{-}15)$$

$$\text{for } k = 0,\ 1 \ldots \text{n-1}$$

4.2.2.1 Theorem 1 -

The discrete interval polynomial, F(z), in equations 4-12 and 4-15 is stable if :

(i) the nominal polynomial, $\Phi(z)$ is stable

(ii) $$|\Phi(e^{j\omega_k T})| > \Psi(\Xi(e^{j\omega_k T})) \qquad (4\text{-}16)$$

$$\forall \quad \omega_k \quad 0 \leq k \leq N-1$$

Proof:

The proof is a consequence of the Rouche's Theorem. The sufficient condition for stability by Rouche's Theorem is :

If (a) $\Phi(z)$ is Schur (i.e., $|\lambda_k| < 1$)

and

(b) $|\Phi(z)| > |\Xi(z)| \qquad \forall$ z on Γ

Then the polynomials, $\Phi(z)$ and $\Phi(z)+\Xi(z)$ have the same number of zeros inside the disk, $|z| < 1$. In other words, if $\Phi(z)$ is stable, then

$\Phi(z)+\Xi(z)$ is also stable. Unfortunately, Rouche's Theorem requires the polynomial coefficients in $\Xi(z)$ to have fixed values. The trick in using Rouche's Theorem for an interval polynomial system is to modify the calculation of $|\Xi(z)|$. Then the same Rouche's Theorem can be applied to check the stability of interval polynomial system.

Condition (i) in Theorem 1 corresponds to condition (a) in Rouche's Theorem. Since condition (a) involves only one polynomial, it can be checked explicitly by either a root finding routine or by Jury Test (Jury 1964); and it poses no problem. To satisfy condition (b) for an interval polynomial system, one must use a magnitude condition.

The Z-transform of $\Xi(z)$ can be calculated by substituting $e^{j\omega_k T}$ for z in equation 4-14 (Neff 1984) :

$$\Xi(e^{j\omega_k T}) = \sum_{l=0}^{n-1} \beta_l e^{j\omega_k T} \tag{4-17}$$

and the magnitude of $\Xi(z)$ can be evaluated through

$$|\Xi(e^{j\omega_k T})| = \left(\Xi(e^{j\omega_k T})\ \Xi(e^{-j\omega_k T}) \right)^{1/2} \tag{4-18}$$

$$\forall \quad \omega_k \qquad 0 < k < N-1$$

The RHS of equation 4-18 can be expanded as

$$= [\{ \sum_{l=0}^{n-1} \sum_{i=0}^{n-1} \beta_i \beta_l e^{j(l-i)\omega_k T} \}]^{1/2} \tag{4-19a}$$

Using the trigonometry equality, $e^{j\theta}=\cos(\theta) + j \sin(\theta)$, equation 4-19a can be rewritten as :

$$= [\ \{ \sum_{l=0}^{n-1} \sum_{i=0}^{n-1} \beta_i \beta_l \cos((l-i)\omega_k T \}]^{1/2} \qquad (4\text{-}19b)$$

$$+ j [\{ \sum_{l=0}^{n-1} \sum_{i=0}^{n-1} \beta_i \beta_l \sin((l-i)\omega_k T \}]^{1/2}$$

Since the magnitude of $\Xi(z)$ is a scalar quantity :

i.e.,

$$|\Xi(e^{j\omega_k T})| = a+bj$$

and

$$b=0.$$

the imaginary part in equation 4-19b must be zero. Thus the RHS of equation 4-18 becomes:

$$= [\ \{ \sum_{l=0}^{n-1} \sum_{i=0}^{n-1} \beta_i \beta_l \cos((l-i)\omega_k T \}]^{1/2} \qquad (4\text{-}20)$$

The trick to get around the infinite possibilities arising from an interval polynomial system is to find the maximum magnitude of $\Xi(z)$ at each discrete frequency, ω_k.

$$|\Xi(e^{j\omega_k T})| \le [\max \{ \sum_{l=0}^{n-1} \sum_{i=0}^{n-1} \beta_i \beta_l \cos((l-i)\omega_k T)\}]^{1/2} \qquad (4\text{-}21)$$

$$\le [\sum_{l=0}^{n-1} \sum_{i=0}^{n-1} \bar{\beta}_i \ \bar{\beta}_l | \cos((l-i)\omega_k T \ | \ ]^{1/2} \qquad (4\text{-}22a)$$

where $\bar{\beta}_i$ and $\bar{\beta}_l$ are the symmetric bounds on the uncertainties.

Defining $\Psi(\Xi(e^{j\omega_k T}))$ as the RHS of equation 4-22a yields:

$$|\,\Xi(e^{j\omega_k T})| \quad \leq \quad \Psi(\Xi(e^{j\omega_k T})) \qquad \forall\ z \text{ on } \Gamma \qquad (z = e^{j\omega_k T}) \tag{4-22b}$$

Equation 4-22b implies that condition (b) of Rouche's Theorem

$$|\ \Phi(z)\ | \quad > \quad |\ \Xi(z)\ | \qquad \forall\ z \text{ on } \Gamma$$

if :

$$|\Phi(e^{j\omega_k T})| \ > \Psi(\Xi(e^{j\omega_k T})) \qquad \forall\ z \text{ on } \Gamma$$

Therefore conditions (i) and (ii) in Theorem 1 extend the conditions (a) and (b) in Rouche's Theorem to an interval polynomial system. Thus the interval polynomial, F(z) in equation 4-12, is stable if conditions (i) and (ii) are satisfied.

Proved

In order to satisfy condition (ii) in Theorem 1, one needs to check all discrete frequencies on the unit circle. To simplify the criteria, a frequency independent criteria is developed in Corollary 1.

4.2.2.2 Corollary 1 -

Corollary 1 :

Given the discrete interval polynomial, F(z), described in equations 4-12,4-13,4-14 and 4-15.

The interval polynomial, F(z) in equation 4-12, 4-13 is stable if:

(i) the nominal polynomial, $\Phi(z)$ is stable

(ii) $$\min_{\omega_k}(|\Phi(e^{j\omega_k T})|) > \sigma(\Xi(z)) \qquad (4\text{-}23)$$

Proof :

The RHS of equation 4-22a defines $\Psi(\Xi(e^{j\omega_k T}))$, setting the cosine term to its maximum value, 1, and noting that

$$\sum_{i=0}^{n-1} \sum_{l=0}^{n-1} \bar{\beta}_i \bar{\beta}_l = \sum_{i=0}^{n-1} \bar{\beta}_i \sum_{l=0}^{n-1} \bar{\beta}_l = (\sum_{i=0}^{n-1} \bar{\beta}_i)^2 \qquad (4\text{-}24)$$

yields

$$\Psi(\Xi(e^{j\omega_k T})) \le \sum_{i=0}^{n-1} |\bar{\beta}_i| \qquad (4\text{-}25)$$

$$\forall \quad \omega_k \qquad 0 \le k < N-1$$

The RHS of equation 4-25 defines the supremal gain, $\sigma(\Xi(z))$, of the uncertain polynomial in equation 4-14.

Since, by definition,

$$|\Phi(e^{j\omega_k T})| \geq \min_{\omega_k} |(\Phi(e^{j\omega_k T})| \qquad \forall \ \omega_k \quad 0 \leq k < N-1$$

and equation 4-25 defines,

$$\sigma(\Xi(z)) \geq \Psi(\Xi(e^{j\omega_k T})) \qquad \forall \ \omega_k \quad 0 \leq k < N-1$$

then if condition (ii) of Corollary 1 is true, the following is also true

$$|\Phi(e^{j\omega_k T})|) > \Psi(\Xi(e^{j\omega_k T})) \qquad \forall \ \omega_k \quad 0 \leq k \leq N-1$$

Conditions (i) and (ii) of Corollary 1 imply the conditions in Theorem 1. This proves Corollary 1.

Corollary 1 is a simpler criteria than Theorem 1. It requires the checking of condition (ii) at a frequency which corresponds to the minimum magnitude of $|\Phi(z)|$. A further simplification to Corollary 1 is to find the lower bound of $|\Phi(z)|$. A criteria using this simplification is developed in Corollary 2.

4.2.2.3 Corollary 2 -

Corollary 2 :

Given the discrete interval polynomial, F(z), described in equations 4-12, 4-13, 4-14 and 4-15. The interval polynomial, F(z) in equations 4-8, 4-9 is stable if :

(i) the nominal polynomial, $\Phi(z)$ is stable

(ii) $1/\mu(D(z)) > \sigma(\Xi(z)) \qquad (D(z) = 1/\Phi(z))$ (4-26)

Proof :

Since, by definition,

$$|\Phi(z)| \qquad \geq \min(|\Phi(z)|)$$

and from equation C-9 in Appendix C, the magnitude of a polynomial can be bounded below by the inverse spectral gain, then

$$|\Phi(z)| \qquad \geq 1/\mu(D(z)) \tag{4-27}$$

Now if condition (ii) of Corollary 2 is true, then the following is true

$$|\Phi(z)| \qquad \geq \sigma(\Xi(z))$$

From equation C-5 and C-3 in Appendix C, the supremal gain of a polynomial is always greater than the magnitude of the polynomial:

$$|\Phi(z)| \quad \geq |\Xi(z)| \quad \forall \ z \text{ on } \Gamma \tag{4-28}$$

Therefore conditions (i) and (ii) in Corollary 2 imply the conditions (a) and (b) in Rouche's Theorem. This completes the proof.

Corollary 2 requires that the polynomial $\Phi(z)$ be inverted (see equation 4-26). This may seem a little clumsy for practical computation. However, in the analysis of uncertain systems (see Wong et al.1987b), the characteristic polynomial usually comes in a form similar to equation 4-29 because of the feedback effect.

$$1 + \Xi(z)\ \Phi(z) = 0 \tag{4-29}$$

$$\text{or} \quad F(z) = \Xi(z) + 1/\ (\Phi(z)) = 0 \quad \text{and } D(z) = \phi(z)$$

One can then directly apply Corollary 2 in equation 4-26 without the need of an inversion. Figure 4.1 illustrates the relationship between Theorem 1, Corollary 1 and Corollary 2. It plots the magnitudes of $|\Phi(z)|$, $\min(\Phi(z))$, $1/(\mu(1/\Phi(z)))$, $\sigma(\Phi(z))$ and $\psi(\Xi(z))$ against normalized discrete frequencies ($\frac{\omega_k T}{\pi}$). The significance of this plot is that it gives a graphical representation of the criteria just derived. The higher the magnitude of $|\Phi(z)|$ over the frequency range, the higher the tolerance to model uncertainty. From the control point of view, the known portion, $\Phi(z)$, relates to the nominal process model as well as the controller. Therefore for a given nominal process model, one can always design a controller which provides a large tolerance to model uncertainty.

4.2.2.4 Examples

Three examples are given in this section to illustrate theorem 1 and its corollaries. Example 1 compares the conservativeness of the criteria developed in this chapter by gradually increasing the uncertainty in the interval polynomial; the criteria are applied to the polynomial and the results are compared. Example 2 shows how closed loop control system stability can be formulated as an interval polynomial. A Dahlin controller (Dahlin (1968)) was used as the controller in this example. The third example uses the distillation control loop simulation described in Appendix B to illustrate the stability criteria.

Example 1:

Consider the interval polynomial below which corresponds to equations 4-8 and 4-9:

$$\Phi(z) = z^2 + 0.2z + 0.1$$

$n = 2$; $\alpha_1 = 0.2$ $\bar{\beta}_1$ is specified between 0.5 and 0.76

$\alpha_0 = 0.1$ $\bar{\beta}_0 = 0.1$

The roots of $\Phi(z)$ are -0.1+0.6j and -0.1-0.6j. Since the roots are inside the unit circle, the nominal system is stable. Table 4-1 shows the effect of the gradual increase of the second uncertain coefficient, $\bar{\beta}_1$, on the proposed stability criteria.

Table 4-1: Comparison of the proposed stability criteria for the discrete interval polynomial in Example 1 at different levels of uncertainty

Criteria (i)-(iv) correspond to equation 4-11, 4-26, 4-23 and 4-16

Case 1 : $\bar{\beta}_1 = 0.5$ $\sigma(\Xi(z)) = 0.6$

(i)	Root location method		criteria = 0.90	(eq 4-11)	Stable
(ii)	Corollary 2		$1/\mu(D(z))= 0.78$	(eq 4-26)	Stable
(iii)	Corollary 1	(N=30)	$\min(\lvert\Phi(w)\rvert)= 0.85$	(eq 4-23)	Stable
(iv)	Theorem 1	(N=30)	satisfied	(eq 4-16)	Stable

Case 2 : $\bar{\beta}_1 = 0.6$ $\sigma(\Xi(z)) = 0.7$

(i)	Root location method		criteria = 1.00	(eq 4-11)	*
(ii)	Corollary 2		$1/\mu(D(z))= 0.78$	(eq 4-26)	Stable
(iii)	Corollary 1	(N=30)	$\min(\lvert\Phi(w)\rvert)=0.85$	(eq 4-23)	Stable
(iv)	Theorem 1	(N=30)	satisfied	(eq 4-16)	Stable

Case 3 : $\bar{\beta}_1 = 0.70$ $\sigma(\Xi(z)) = 0.8$

(i)	Root location method		criteria = 1.10	(eq 4-11)	*
(ii)	Corollary 2		$1/\mu(D(z))= 0.78$	(eq 4-26)	*
(iii)	Corollary 1	(N=30)	$\min(\lvert\Phi(w)\rvert)= 0.85$	(eq 4-23)	Stable
(iv)	Theorem 1	(N=30)	satisfied	(eq 4-16)	Stable

Case 4 : $\bar{\beta}_1 = 0.76$ $\sigma(\Xi(z)) = 0.86$

(i)	Root location method		criteria = 1.16	(eq 4-11)	*
(ii)	Corollary 2		$1/\mu(D(z))= 0.78$	(eq 4-26)	*
(iii)	Corollary 1	(N=30)	$\min(\lvert\Phi(w)\rvert)=0.85$	(eq 4-23)	*
(iv)	Theorem 1	(N=30)	satisfied	(eq 4-16)	Stable

* - Inconclusive : Since the Theorem is a necessary but not sufficient condition, violating the criteria does not imply that the polynomial is unstable.

From the table, the root location method fails to predict a stable polynomial at a much earlier point than the proposed methods. This shows that the proposed criteria are less conservative than the root location criteria.

Example 2:

Consider the characteristic equation of a discrete closed loop control system :

$$1 + Gc(z)\ (\ Gm(z) + Un(z)\ ) = 0 \tag{4-30}$$

Dividing the equation 4-30 by Gc(z):

$$(\frac{1}{Gc(z)}) + (\ Gm(z) \quad + Un(z)\ ) = 0 \tag{4-31}$$

If a Dahlin controller is used (Dahlin 1968),

$$Gc(z) = \frac{1}{Gm(z)} \cdot \frac{K(z)}{1 - K(z)} \tag{4-32}$$

where K(z) is the desired closed loop response in Dahlin Controller.

Equation 4-31 becomes :

$$\frac{Gm(z)}{K(z)} + Un(z) \quad = 0 \tag{4-33}$$

If Gm(z)/K(z) defines the polynomial $\Phi(z)$ in example 1, then the stability results in Example 1 apply directly to this control example. Furthermore, equation 4-33 implies that the tolerable uncertainty is determined by $\frac{Gm(z)}{K(z)}$. Depending on the tuning of the controller, K(z), the term Gm(z)/K(z) will take on different characteristics as shown in Figure 4-2. For a slowly tuned K(z), the magnitude of Gm(z)/K(z) is high over the frequency range. From equation 4-33 this implies that the uncertainty tolerance is high. Comparing with Figure 4-1, it is clear that the higher the magnitude of Gm(z)/K(z) over the frequency range, the higher the uncertainty tolerance. This interesting result on controller robustness corresponds closely with the intuitive way of tuning Dahlin controller - increase the closed loop time constant in K(z) in order to increase robustness.

Example 3:

In this example, the top composition control loop in the extractive distillation tower simulation (see Appendix B-2) is used to illustrated the applicability of the theorem and corollaries developed in this chapter. The process model between the acetone composition and the reflux rate is:

$$Gp_{11}(z) \quad = \frac{0.9059z^{-2} + 0.2635\ z^{-3}}{1 - 0.627\ z^{-1}}$$

If a Dahlin controller is used, the characteristic equation would be similar to equation 4-33 :

$$\frac{Gp_{11}(z)}{K(z)} + Un(z) = 0 \tag{4-34}$$

Case A: closed loop dynamics was tuned to be slower than the open loop

$$K(z) = \frac{0.3z^{-2}}{1-0.7z^{-1}} \tag{4-35}$$

Equation 4-34 becomes :

$$\left(\frac{0.9059z^{-2} + 0.265z^{-3}}{1 - 0.627z^{-1}} \cdot \frac{1-0.7z^{-1}}{0.3z^{-2}}\right) + Un(z) = 0 \tag{4-36}$$

Case B: closed loop dynamics was tuned to be faster than the open loop

$$K(z) = \frac{0.5z^{-2}}{1-0.5z^{-1}} \tag{4-37}$$

Equation 4-34 becomes :

$$\left(\frac{0.9059z^{-2} + 0.265z^{-3}}{1 - 0.627z^{-1}} \cdot \frac{1-0.5z^{-1}}{0.5z^{-2}}\right) + Un(z) = 0 \tag{4-38}$$

Both cases will be considered in the stability analysis.

In a control system containing fast and slow dynamics, one tends to model the system as a slow first order system. In this case, the unmodelled fast dynamics would only affect the first few coefficients in the process model. The uncertainty, Un(z), in this case is:

$$Un(z) = \beta_2\, z^{-2} + \beta_3 z^{-3} \qquad (4\text{-}39)$$

$$\text{where} \quad -\bar{\beta}_2 \le \beta_2 \le \bar{\beta}_2 = 0.2$$

$$\text{and} \quad \beta_3 = 0.2$$

The stability criteria are applied to both systems, equations 4-36 with 4-39 and equations 4-37 with 4-39. The results are collected in Tables 4-2 and 4-3. In the slow tuning case (see Table 4-3), the table shows that when β_2 increases to 1.6, root location criteria failed. The criteria developed in this work shows that β_2 can be increased further to 2.0 before the criteria becomes inconclusive. In the fast tuning case (see Table 4-2), the β_2 tolerable uncertainty for root location criteria and Theorem 1 are 0.3 and 1.0 respectively. These values are lower than the slow tuning case. This agrees with the fact that detuning can tolerate more uncertainty. In general, the proposed criteria perform much better than the root location criteria.

Table 4-2: Comparison of the proposed stability criteria for the closed loop system in Example 2 at different levels of uncertainty (Case A) - Fast Tuning

Criteria (i)-(iv) correspond to equation 4-11, 4-26, 4-23 and 4-16

Case 1 : $\bar{\beta}_2 = 0.2$ $\sigma(\Xi(z)) = 0.4$

(i)	Root location method		criteria = 0.931	(eq 4-11)	Stable
(ii)	Corollary 2		$1/\mu(D(z))= 1.206$	(eq 4-26)	Stable
(iii)	Corollary 1	(N=30)	$\min(\lvert\Phi(w)\rvert)= 1.209$	(eq 4-23)	Stable
(iv)	Theorem 1	(N=30)	satisfied	(eq 4-16)	Stable

Case 2 : $\bar{\beta}_2 = 0.3$ $\sigma(\Xi(z)) = 0.5$

(i)	Root location method		criteria = 1.002	(eq 4-11)	*
(ii)	Corollary 2		$1/\mu(D(z))= 0.723$	(eq 4-26)	Stable
(iii)	Corollary 1	(N=30)	$\min(\lvert\Phi(w)\rvert)=0.739$	(eq 4-23)	Stable
(iv)	Theorem 1	(N=30)	satisfied	(eq 4-16)	Stable

Case 3 : $\bar{\beta}_2 = 1.0$ $\sigma(\Xi(z)) = 1.2$

(i)	Root location method		criteria = 1.37	(eq 4-11)	*
(ii)	Corollary 2		$1/\mu(D(z))= 1.206$	(eq 4-26)	*
(iii)	Corollary 1	(N=30)	$\min(\lvert\Phi(w)\rvert)= 1.209$	(eq 4-23)	*
(iv)	Theorem 1	(N=30)	not satisfied	(eq 4-16)	*

Table 4-3: **Comparison of the proposed stability criteria for the closed loop system in Example 3 at different levels of uncertainty (Case B) - Slow Tuning**

Criteria (i)-(iv) correspond to equation 4-11, 4-26, 4-23 and 4-16

Case 1 : $\bar{\beta}_2 = 0.2$ $\sigma(\Xi(z)) = 0.4$

(i)	Root location method		criteria = 0.533	(eq 4-11)	Stable
(ii)	Corollary 2		$1/\mu(D(z))= 2.178$	(eq 4-26)	Stable
(iii)	Corollary 1	(N=30)	$\min(\|\Phi(w)\|)= 2.22$	(eq 4-23)	Stable
(iv)	Theorem 1	(N=30)	satisfied	(eq 4-16)	Stable

Case 2 : $\bar{\beta}_2 = 1.6$ $\sigma(\Xi(z)) = 1.8$

(i)	Root location method		criteria = 1.000	(eq 4-11)	*
(ii)	Corollary 2		$1/\mu(D(z))= 2.178$	(eq 4-26)	Stable
(iii)	Corollary 1	(N=30)	$\min(\|\Phi(w)\|)=2.22$	(eq 4-23)	Stable
(iv)	Theorem 1	(N=30)	satisfied	(eq 4-16)	Stable

Case 3 : $\bar{\beta}_2 = 2.0$ $\sigma(\Xi(z)) = 2.2$

(i)	Root location method		criteria = 1.13	(eq 4-11)	*
(ii)	Corollary 2		$1/\mu(D(z))= 2.178$	(eq 4-26)	*
(iii)	Corollary 1	(N=30)	$\min(\|\Phi(w)\|)= 2.22$	(eq 4-23)	*
(iv)	Theorem 1	(N=30)	not satisfied	(eq 4-16)	*

4.3 Discussion

In this chapter, three straight forward sufficient conditions for the stability of a discrete interval polynomial are presented. They are all based on the Rouche's Theorem. It is interesting to compare this approach with that of Bartlett et al. (1987). They presented an extension of the work of Huang et al. (1987); they are, in fact, the same researchers. They solve for the location of the roots of the characteristic polynomial when the family of all such polynomials is polytopic in the coefficient space. Using the theory of polytopic sets, they show that the root location of the entire family is determined by examining the roots of the polynomials defining the exposed edges of the polytope. This is a very useful result when it is possible to define the process uncertainty in a polytopic way. Physical examples might be: changes in the operating point of distillation column (McDonald et al. (1987)), or changes in the concentration of an impurity in the feed to a polymer reactor (Kozub and MacGregor (1990)). In these cases, the changes (uncertainty) in a few variables causes corresponding changes in many of the coefficients of the characteristic polynomial. Unfortunately, this approach requires the definition of the "vertex polynomials" which describe the multidimensional polytope. In McDonald's work (1987), she found the family of physically realizable operating points by exhaustive simulation with a model which is only an approximation to the true process. Kozub and MacGregor (1990) used a complex mechanistic model of a polymer reactor to investigate the effect of impurity concentration on the polymer properties. This type of investigation is quite expensive (requiring approximately 2-3 man years of engineering effort to develop the model and establish the physical property data) but there appears to be no other ways of gathering the information required to define the polytopic set for complex physico-chemical processes and still have some confidence that the set is realistic. Simpler processes might be modeled in the least squares fitted sense by low order transfer functions. In this case, a variation in one parameter (say, time constant) will produce a family of polynomials which are polytopic. Unfortunately, experience in the processing industries suggests that the uncertainty often comes from multiple sources and that the transfer functions fitted to plant data show independent variation in many parameters, including the time delay. For a process modeled in this way, the flat faced polytope used by Bartlett et al. (1987) would need to be quite large (and therefore conservative) if it is to encompass the whole family of polynomials, or it would have to be quite complex (and therefore time consuming to define). The advantage of interval polynomials is that they can be easily defined from historical operating records via impulse responses or fitted transfer functions and operational experience. The conservatism that they introduce into the analysis is a serious problem and the work of Bartlett et al. (1987) is an important theoretical contribution when a realistic polytope can be defined. There are, however, a large number of systems in the processing industries which are either so complex, or so understaffed, that the effort to define the polytope is hard to justify. For these systems, interval polynomials are usable, though far from satisfactory. Although the current approach is conservative in nature, this is only an artifice of the mathematics.

The concept of an interval polynomial is a good description of a real control system. If better mathematical tools were available, the conservatism can be further reduced.

4.4 Conclusion

Three criteria have been derived to provide the sufficient conditions for the stability of discrete interval polynomials. They provide stability criteria that are "tighter" than the root location criteria. They can be easily implemented on a digital computer to predict the stability of a characteristic polynomials of the interval type. These criteria can be used in a wide variety of processing industry applications to predict the stability of an uncertain control system.

4.5 Nomenclature

$F(z)$	: interval polynomial
$\Phi(z)$	: certain portion of $F(z)$
$\Xi(z)$	: uncertain portion of $F(z)$
$D(z)$	: inverse of $\Phi(z)$ $(1/\Phi(z))$
N	: total number of discrete frequency
n	: order of $F(z)$
$P(z)$	: Polytope of polynomials
T	: sampling time
ω	: discrete frequency
ω_k	: discrete frequency at frequency $2\pi k/NT$
α_i	: coefficient i of $\Phi(z)$ corresponding to z^i
β_i	: coefficient i of $\Xi(z)$ corresponding to z^i
$\bar{\beta}_i$	: upper bound of β_i
$-\bar{\beta}_i$	: lower bound of β_i
λ_i	: complex eigenvalue i
Γ	: closed path, a unit circle centering at the origin, on the complex plane ; $z=e^{j\omega_k T}$; $j=\sqrt{-1}$
$\Psi(G(e^{j\omega_k T}))$	: maximum magnitude of $G(z)$ at ω_k $= \{ \sum_{i=0}^{n-1} \sum_{l=0}^{n-1} \bar{\beta}_i \; \bar{\beta}_l \; \lvert\cos((-i+l)\omega_k T)\rvert \}^{1/2}$
$\sigma(G(z))$	: supremal gain of $G(z)$ $= \sum_{l=0}^{n-1} \lvert g_k \rvert$

$\mu(G(z))$	: spectral gain of $G(z)$ $= \{ \sum_{i=-n}^{n} \lvert \sum_{j=0}^{n-1} g_{i+j} g_j \rvert \}^{1/2}$
$\min(\Phi(z))$	: minimum magnitude of $\Phi(z)$ over Γ
$\lvert f \rvert$	: magnitude of complex function, f or absolute value of scalar, f
Gm(z)	: nominal process model
Un(z)	: process model uncertainty
Gc(z)	: discrete controller
K(z)	: desired closed loop servo response

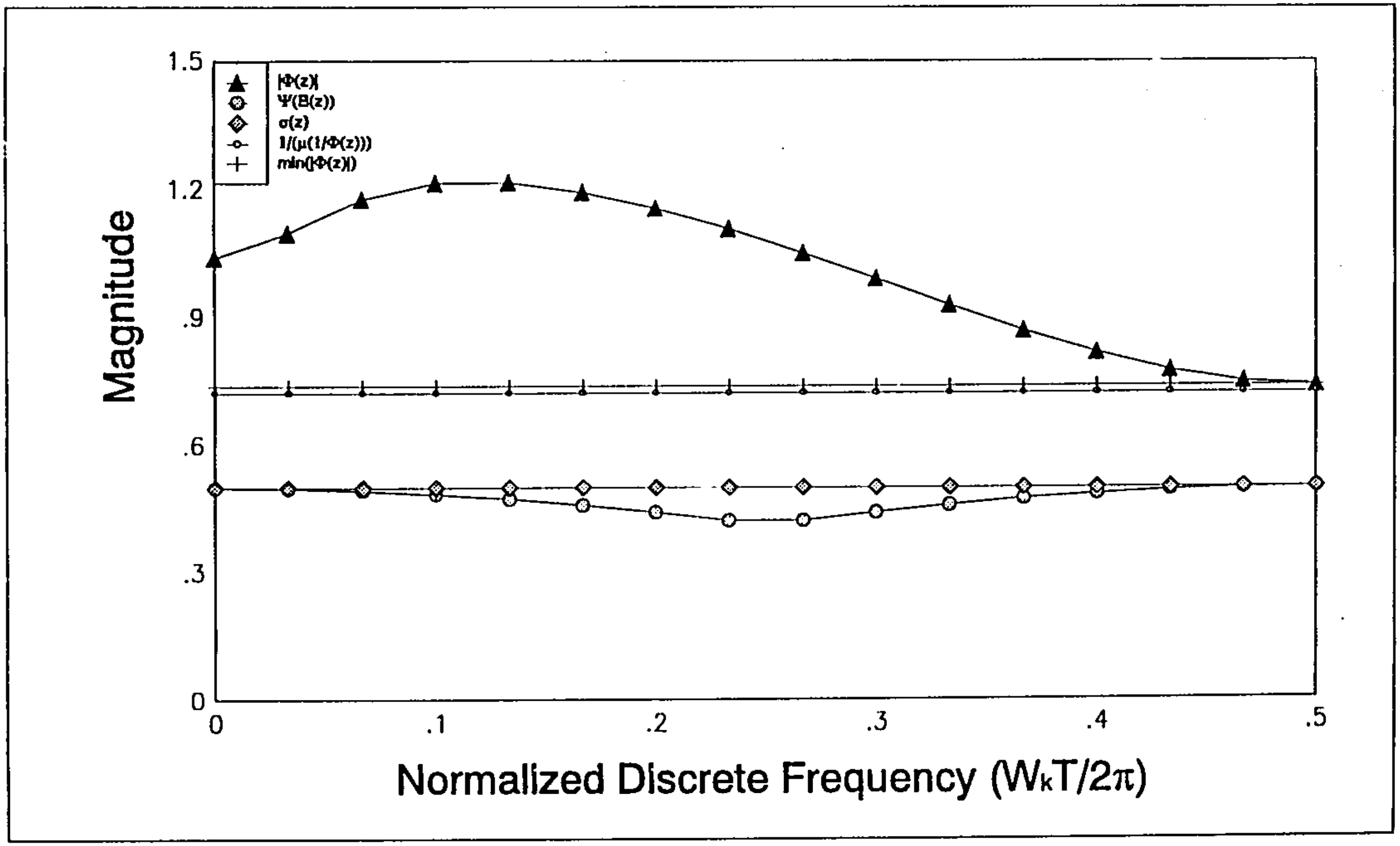


Figure 4-1: Relationship Between Theorem 1, Corollary 1 and Corollary 2

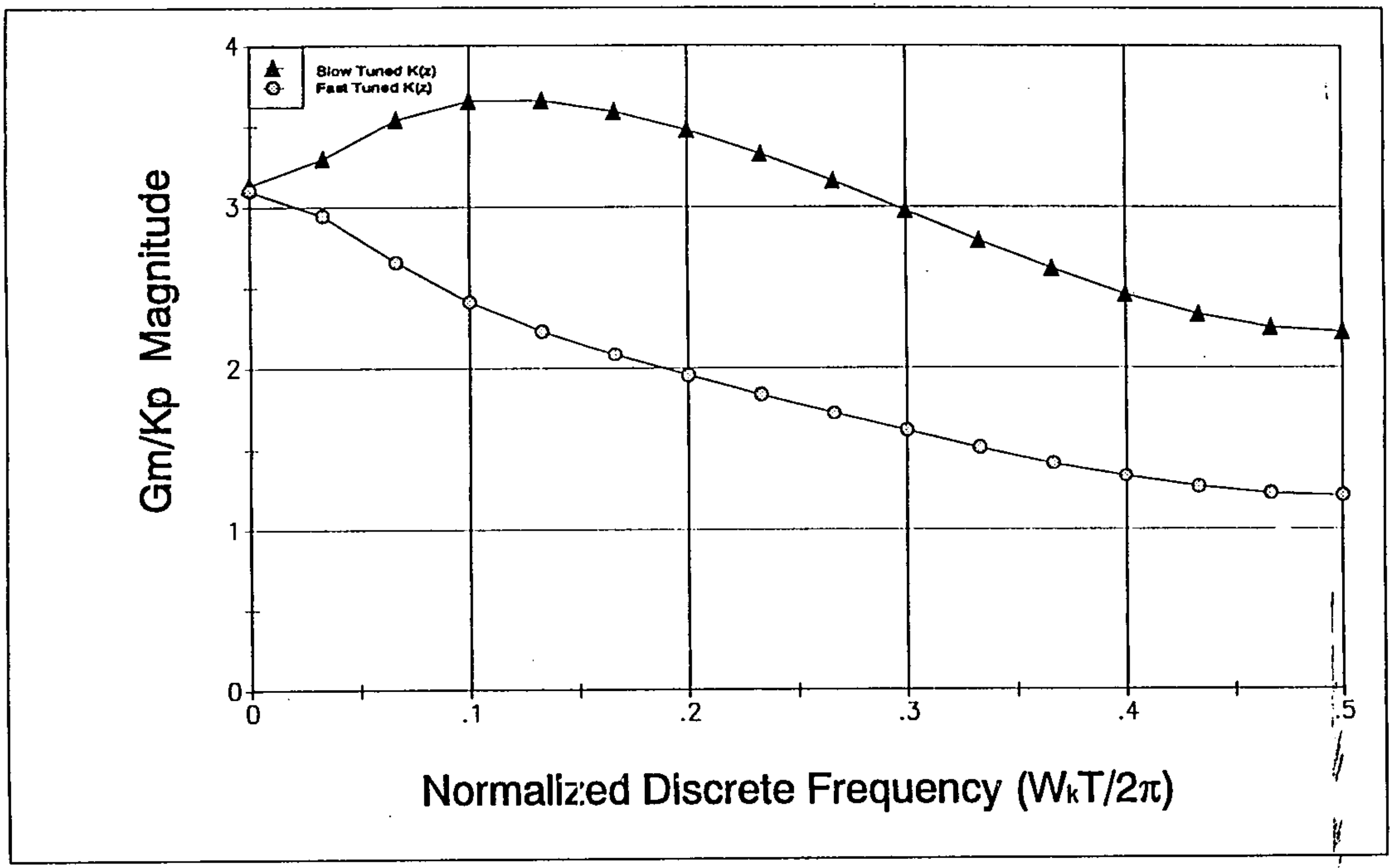


Figure 4-2: Stability Robustness of Dahlin Controller

CHAPTER 5

ROBUSTNESS - PERFORMANCE OF AN UNCERTAIN CONTROL SYSTEM

5.1. Introduction

The overall effectiveness of a control system depends on 2 factors - stability and transient performance. Stability dictates the ultimate behavior of a control system while transient performance measures the closeness of the process variables to their targets. One has to resolve the stability issue before one achieves good transient performance. Chapter 4 provides a methodology to measure the stability of an uncertain control system. The next question is how to assess the "transient performance" of an uncertain control system. Here, the same problem is encountered as in the stability case. Since the actual process is not exactly the same as the process model, the exact closed loop transient behavior cannot be predicted. Being able to predict the closed loop behavior of an uncertain control system can enhance one's confidence in the controller. Therefore, the problem becomes one of how to predict the closed loop behavior of an uncertain control system. This issue is generally categorized as the "performance robustness" aspect of the "Model Mismatch Problem" in the control literature. The work of Wong et al. (1987a) and Kusuma and Elliott (1984) demonstrates that the model mismatch problem is an important issue not only in simulations but also in experimental and industrial applications. Although advanced control theory promises significant performance improvement over classical techniques, the improvement cannot be guaranteed if the process model is not "adequate". This chapter provides a systematic way to analyze the performance of an uncertain process control system.

There have been many studies on the robustness and performance of control system. The pioneer in this area is Horowitz. His approach (Horowitz 1982, Horowitz et al. (1972)) is mainly based on the manipulations of the loop transfer functions on the Nichol's Chart. This approach, though powerful, has not been well received in chemical process control. Doyle and Stein (1981), Palazoglu and Arkun (1986), Skogestad and Lundstrom (1990) advocated the Structured Singular Value (SSV) approach to analyze and design control systems. This is an extension of the Single Input Single Output (SISO) loop gain concept to Multi Input Multi Output (MIMO) systems and it relies heavily on the mathematical properties of Singular Values. To apply this technique, one requires a solid background in advanced mathematical skills such as functional analysis and complex analysis. Leitmann and Barmish (1982) proposed the method of ultimate boundedness of a control system. The process uncertainty is specified through the coefficient uncertainty in the transition and input matrices in a state-space formulation. As a result, they have an interval matrix instead of an interval polynomial. Lunze (1982,1984) developed a time domain bounding method for continuous-time uncertain systems. The uncertainty is specified through a bound in the impulse response of the process. The method mainly deals with continuous system. All these different approaches provide a set of tools to analyze the performance robustness of control systems. The current study was inspired by Lunze's work. This chapter presents a formal approach to the robustness problem by explicitly separating the output uncertainty from the lumped closed loop output response and extends Lunze's method to handle discrete time invariant system. The closed loop response uncertainty band produced in this work is less conservative than that in Lunze's work. The major contribution of this work is to provide a methodology to look at the transient behavior of an uncertain control system. The key to this approach is to use a nominal process model, a process controller and an open loop process uncertainty specification to calculate an envelope (uncertainty band) for the closed loop output response. With the predicted output envelope, one can assess the performance robustness of an uncertain control system by examining the size of the uncertainty band. A model based controller for an uncertain control system can then be tuned from this uncertainty band. Furthermore, the robustness of different types of controllers can be compared based on the size of these bands.

This chapter is organized into 8 sections. Section 5.2 explains the rationale for developing the Error Band Method (EBM), details of which are given in Section 5.3. Examples are described in Section 5.4 to demonstrate the proposed robustness evaluation procedure. Section 5.5 illustrates two potential applications of EBM - to tune a controller and to compare the robustness of controllers. Conclusions and remarks are given in Sections 5.6 and 5.7. The nomenclature for this chapter is collected in Section 5.8.

5.2. Rationale for the Error Band Method

There are many complaints about the performance of model based digital controllers in industrial applications (Kestenbaum et al. 1976). The criticism is mainly focused on the sensitivity of the process output to the process model used in the controller design. The classical safeguard against the "model mismatch problem" is to detune the controller. However, this detuning is contrary to the purpose of high performance, model based, digital controllers. A gross estimation of the process model uncertainty can be obtained from day to day operations or from repeated empirical testing. This information about model uncertainty is part of the process knowledge which should be incorporated into the controller design procedure so that a more "robust" controller can be designed. Although the frequency domain is a very popular approach in analyzing process uncertainty, it is felt that some information cannot be easily recovered from the frequency domain approach, namely the transient behavior of closed loop output response (see Krishnan and Cruickshanks (1977)). Furthermore, data from chemical processes often comes in the form of realtime signals and one needs to convert these signals into a frequency domain uncertainty through a transform (see Laughlin et al. (1986)) in order for the problem definition to fit into the framework of the frequency domain method. If one is not careful with the transformation, conservativeness may be induced because of this mathematical translation process (see Laughlin et al. (1986)). The criteria (e.g., bandwidth, maximum amplification, etc.) obtained from the frequency technique are not intuitive to many chemical engineers. These are the reasons behind the development of the alternate approach presented here. It is meant as a direct time domain criterion for robustness evaluation. This method should complement the powerful frequency domain methods.

The development of the Error Band Method (EBM) consists of four parts :

1) separating closed loop uncertainty from the closed loop output equation
2) specifying open loop process uncertainty
3) predicting the closed loop stability
4) predicting the closed loop performance

5.3. Error Band Method (EBM)

Uncertainty in the EBM is specified non-parametrically in terms of a band in the impulse response. Figure 5-1 gives an overview of the EBM. Notice that the impulse weight is expressed as a "range" at every time lag rather than a fixed value. The collection of the maximal values at each time lag gives the maximal process transfer function, Gpmax(z) while the minimal values give the minimal process transfer function, Gpmin(z). The model based controller, Gc(z), can be designed based on some nominal process transfer function, Gpnom(z). Given the open loop uncertainty, Gpmax(z) and Gpmin(z), and the controller, Gc(z), the main result of the EBM is to generate a closed loop uncertainty band which bounds all the possible closed loop output step responses; see Figure 5-1. This is a very straight forward criterion for an engineer. The bigger the band, the worse the robustness. From the controller design point of view, the goal is to design a controller such that the size of the uncertainty band is reduced and the speed of the closed loop response increased. Of course, the stability issue has to be resolved before the performance can be addressed. The stability of the uncertain system can be assessed through the stability criteria developed in Chapter 4.

The proposed EBM consists of:

a) Separating Closed Loop Uncertainty

b) Describing Open Loop Uncertainty

c) Assessing Closed Loop Stability

d) Assessing Closed Loop Performance

These four elements are discussed in detail in the following sections.

5.3.1 Separating closed loop uncertainty

The classical way of analyzing a control loop is through the control block diagram shown in Figure 5-2a. Notice that the uncertainty Gerr(i) is in an "additive" form. If there is no uncertainty, i.e. Gerr(z)=0, the nominal closed loop response can be calculated from the process model, Gpnom(z), and the process controller, Gc(z), using equation 5-1a :

$$Y(z)= \frac{G_c(z)(\ G_p(z) + G_{err}(z)\ )\ (R(z)-N(z))}{1 + G_c(z)\ (\ G_p(z) + G_{err}(z)\ )} + \frac{D(z)}{1 + G_c(z)\ (\ G_p(z) + G_{err}(z)\ )} \qquad (5\text{-}1a)$$

$$Y(z)= \frac{G_c(z)\ G_p(z)\ (R(z)-N(z))}{1 + G_c(z)\ G_p(z)} + \frac{D(z)}{(1+G(z)\ G_p(z))} + \frac{G_c(z)\ G_{err}(z)\ G_{yf}(z)\ \ (R(z)-N(z)-D(z))}{(1+G_c(z)\ G_p(z))(1-G_{err}(z)G_{uf}(z))} \qquad (5\text{-}1b)$$

However, if uncertainty exists, the closed loop response calculation is not so straight forward. If the problem is stated in terms of parametric uncertainty, (e.g., Berg et al. (1980), Dumont (1981)) the closed loop responses can be expressed in terms of process gain, time constant or dead time. However, as the process order increases, so does the number of parameters and the calculations become intractable. There is a definite need for a methodology that is not limited by the order of the process and can be applied to parametric as well as non-parametric control systems. By manipulating the control block diagram in Figure 5-2a, it is possible to separate the closed loop output into two parts (see equation 5-1b) : one due to the nominal process response and the other due to the process uncertainty. One can show mathematically that the block diagram in Figure 5-2a (equation 5-1a) is equivalent to that in Figure 5-2b (equation 5-1b). (See Appendix D-7). The implication of this separation is that one can now investigate a closed loop control system based on the uncertain part in equation 5-1b.

Physically, Figure 5-2b can be interpreted as follows. A step change in the set point of the control system would generate a nominal control sequence, Unom(iT), which passes through the uncertainty block, Gerr(z), to generate an uncertain disturbance, F(iT). This disturbance, F(iT), feeds back through the feedback loop, Guf(z), and generates additional control actions and hence more disturbances, F(iT). This additional disturbance causes further deviations from the nominal closed loop output response. The advantage of Figure 5-2b is that the overall closed loop response is separated into two parts: the uncertain part, Ydev(iT), and the nominal part, Ynom(iT). Since the process model and the controller are known in the nominal part, the closed loop response calculation, Ynom(iT), is fairly straight forward. The issue here is how to calculate the uncertainty part, Ydev(iT), since it is a function of the open loop uncertainty, Gerr(z). Notice that there is no limitation imposed on Gerr(z) yet. If Gerr(z) is a parametric gain uncertainty :

where $\quad \mathrm{Gerr}(z) = \alpha\ \mathrm{Gpnom}(z). \qquad \alpha_1 \le \alpha \le \alpha_2$

then Ydev(iT) would be a function of α_1 and α_2 only. However, Gerr(z) can be of a non-parametric nature, similar to the open loop uncertainty shown in Figure 5-1. In that case, Ydev(iT) would be a function of $\overline{\mathrm{Gerr}}(i)$, where $\overline{\mathrm{Gerr}}(i)$ is the maximum open loop model error at lag i. Therefore, Figure 5-2b is a flexible way of investigating closed loop system uncertainty. By finding a bound for Ydev(iT), one can then establish the possible deviations from the nominal closed loop response. Therefore, one can focus the robustness analysis on this part of the closed loop response. It should be pointed out that the entire "uncertainty portion" in Figure 5-2b is driven by the nominal control actions, $U_{nom}(iT)$. This makes the analysis relatively simple because servo and regulatory disturbances can be investigated under the same framework.

After this reconfiguration of the problem, the Input/Output properties of an uncertain system can be obtained from the uncertain portion of Figure 5-2b. Note that the analysis is not restricted to the SISO case. It is extendible tc the MIMO system. However, in order to lay the ground for further study, this work will focus on SISO system. In the subsequent sections, a method will be developed to assess the stability and performance of an uncertain control system.

5.3.2 Describing open loop process uncertainty

All processes are basically nonlinear. However, in many occasions, a linear process model is an adequate description of the behavior of a non-linear process within a local region. Even as a nonlinear process moves across its operating region, it is possible to describe the process with a set of linear process models, $\pi(G_p(z))$, (Latosinsky 1988). The type of process uncertainty investigated in this work either comes from process nonlinearity over the operating region or changes in the process parameters over time. Under these conditions, it is not possible to find one linear process model to adequately describe the process. A more realistic description would be a set of linear process mod^ls $\pi(G_p(z))$.

A linear process model can be described in parametric (e.g., Laplace domain transfer function) or non-parametric form (e.g., Frequency or Impulse response). There have been many studies (Dumont (1981), Berg and Edgar (1980)) of process model uncertainty based on the parametric form. A major disadvantage for this approach is the validity of the results for systems of differing structures (i.e., different orders of the parametric model). The extrapolation of robustness results based on low order systems to high order systems has never been demonstrated to be adequate in general. In contrast, the use of a non-parametric model form frees one from the structural restriction inherent to a parametric model. Thus the results of a robustness analysis based upon non-parametric models apply to a wider class of uncertain systems. Of course, if one knows that the process is of fixed order, the parametric methods are more appropriate for a robustness analysis, (Bartlett ^t al. 1987). The non-parametric form (impulse response model) is used here to represent a process so that structural as well as parametric uncertainty can be tackled under the same methodology.

Consider a process, $G_p(z)$, which is fuzzy and can be adequately described by the linear process set, $\pi(G_p(z))$, the impulse responses of the set are distributed within the bound $G_{pmax}(z)$ and $G_{pmin}(z)$ (see

Figure 5-1 and equation 5-2). In fact, the bounds can be obtained either from engineer's experience with the process, from repeated empirical identification or from the confidence interval of impulse weight identification (MacGregor et al. (1986)).

$$G_{pmin}(i) \leq G_p(i) \leq G_{pmax}(i) \qquad \forall \quad i=0,1\ldots N \qquad (5\text{-}2)$$

One way to map out the open loop uncertainty is to perform open loop tests at the extremal operating conditions or to collect impulse responses as the process parameters change with time. The impulse weights are collected from all these experiments and $G_{pmin}(z)$ and $G_{pmax}(z)$ can be obtained by choosing the extremes of the impulse weights at different lags. It is assumed that the process is, for all practical purposes, time invariant. This implies that, in practice, the changes are so slow that the process appears fixed during the time taken to reject disturbances. This time invariant assumption is crucial in the development of a tight bound for the uncertain closed loop response.

Once the upper and lower bounds of the impulse weights ($G_{pmax}(z)$ and $G_{pmin}(z)$) are obtained, the nominal process model, $G_{pnom}(z)$, and the open loop model uncertainty bound, $G_{err}(z)$, can be redefined as :

$$G_{pnom}(z) = (G_{pmax}(z) + G_{pmin}(z)) / 2.0 \qquad (5\text{-}3)$$

$$G_{err}(z) = (G_{pmax}(z) - G_{pmin}(z)) / 2.0 \qquad (5\text{-}4)$$

Notice that with this redefinition, the model uncertainty is symmetrical about the nominal process. In Appendix D-1, it is shown that using a symmetric model error band, Gerr(i), can reduce the size of the output band. However, the drawback is that this approach requires the resulting output bound be symmetric about a nominal response. Bartlett (1990) and the Monte-Carlo simulation in this study show that the true upper and lower bounds of a closed loop uncertainty system are not necessarily symmetric about the nominal closed loop response. Therefore, certain degree of conservatism is introduced as a result of using symmetric model uncertainty. Fortunately, this work also shows that this conservatism is prominent only in certain class of control system (i.e., in tightly tuned control system). Asymmetric error bound has been investigated in this work but with no promising results. This missing part is the bounding procedure which can take the asymmetric error band and produce a tight closed loop uncertainty band. It was decided that symmetric error bound should be used in this study.

5.3.3 Assessing closed loop stability and performance

If the open loop impulse response can be bounded by upper and lower impulse responses, one can find the upper and lower bounds for the closed loop response (see Figure 5-1). This is the principle behind the EBM. Robustness is evaluated through the predicted closed loop output uncertainty band and "robust" controller design can be achieved by tuning the digital controller so as to obtain a small band.

A Triangular Inequality procedure is used to bound the output from the impulse response block. A set of lemmas concerning the bounding of the closed loop process output has been derived and collected in Appendix D. The highlights of these lemmas are:

Lemma D-1: provides the basis for bounding the output f~om an open loop uncertain process. It is shown in this lemma that symmetric open loop uncertainty can produce a "tight" bound. The key point in the bounding procedure is to locate the source of the uncertainty, i.e., whether the uncertainty is in the process or in the input. If the uncertainty is in the inputs or in the "open loop" process, the bounding procedure shown in Lemma D-1 can be used. However, if the uncertainty is in a "feedback" process as in Lemma D-2, the bounding procedure has to be modified to incorporate the effect of feedback.

Lemmas D-2 and D-3: provide the procedure to generate the output uncertainty band for different locations of uncertainty. If the uncertainty is in the input, R(iT), one can view the system as an "open" loop system with $\frac{G_1(z)}{1+ G_1(z)}$ as the process transfer function and R(iT) as the inputs (see Lemma D-3 in Appendix D). The output bound generated would be tighter than the output bound from Lemma D-2 (see Appendix D) where the uncertainty is in a "feedback" process. Because of the feedback effect, uncertainty will be accumulated in the bounding procedure and hence a bigger bound as a result. This is why the location of source of uncertainty is crucial in the bounding procedure.

Lemmas D-4 and D-5: provide a method to calculate a tighter bound for a, practically, time-invariant process. The key lies on the fact that the sequence of convolution is immaterial for a time invariant system. One can group all the process blocks with no uncertainty in front of the blocks with uncertainty. In the case of a closed loop control system, the Gyf(z) block was moved closed to the Unom(iT), i.e., in front of the Gerr(i) block (see Figure 5-2b). Since both Unom(iT) and Gyf(z) have no uncertainty, an exact output can be calculated. With this rearrangement, the accumulation of uncertainty in the early stage can be avoided till later stage. This results in a tighter output bound for the uncertain system.

Lemma D-6: Based on the proofs in Lemmas D-4 and D-5, the calculation procedure for the closed loop uncertainty bound was derived. The results are summarized in equations 5-5 to 5-9.

The actual process output from an uncertain control system is:

$$Y(iT) = Y_{nom}(iT) + Y_{dev}(iT) \qquad (5\text{-}5)$$

The nominal closed loop input response, Unom(iT), and nominal closed loop output response, Ynom(iT) can be calculated from the nominal process model and the controller as follows (see Figure 5-2b):

$$Y_{nom}(z) = \left(\frac{G_c(z)G_{pnom}(z)(R(z)-N(z))}{1 + G_c(z)G_{pnom}(z)} + \frac{D(z)}{1 + G_c(z)G_{pnom}(z)} \right) \qquad (5\text{-}6a)$$

$$U_{nom}(z) = \left(\frac{G_c(z)\ (R(z)-N(z))}{1 + G_c(z)G_{pnom}(z)} + \frac{-G_c(z)D(z)}{1 + G_c(z)G_{pnom}(z)} \right) \qquad (5\text{-}6b)$$

The deviation, Ydev(iT), from the nominal output response due to model uncertainty, Gerr(i), can be calculated using a convolution sum (see Figure 5-2b):

$$G_{yf}(z) = \frac{1}{1 + G_c(z)\ G_{pnom}(z)} \qquad (5\text{-}7a)$$

$$G_{uf}(z) = \frac{-\ G_c(z)}{1 + G_c(z)\ G_{pnom}(z)} \qquad (5\text{-}7b)$$

$$Y_{dev}(iT) = \sum_{j=0}^{N} G_{yf}(j)\ .\ F(iT - jT) \qquad (5\text{-}8)$$

and

$$F(iT)=\sum_{j=0}^{N}Gerr(j)Unom(iT-jT)\ +\sum_{j=0}^{N}Gerr(j)\sum_{l=0}^{N}Guf(l)F((i-j-l)T) \qquad (5-9)$$

where $F(iT)$ in equation 5-9 is the disturbance due to model uncertainty.

If we know the exact value of $Gerr(z)$, then $Ydev(iT)$ and $F(iT)$ can be exactly caiculated from equations (5-8,5-9). However, the $Gerr(z)$ is known only to the extent of a bound $|\bar{G}err(i)|$. Therefore, only a bound, $\bar{Y}dev(iT)$ on $Ydev(iT)$ can be calculated under these circumstances. Based on Lemmas D-1, D-2, D-3, D-4 and D-5, a procedure has been derived (see Appendix D) to bound the closed loop output of an uncertain feedback control system. The bounds for a linear, practically, time-variant closed loop control system are summarized as follows.

Time-Invariant system:

the closed loop output is independent of the sequence of convolution and a bound for the uncertainty is:(see Lemmas D-4 and D-5 in Appendix D)

$$Umod(iT) = \sum_{j=0}^{N} Gyf(j)\ \ Unom(iT-\ jT) \qquad (5-10)$$

$$\bar{Y}dev(iT) = \sum_{j=0}^{N} |\bar{G}err(j)|\,|Umod(iT-jT)| + \sum_{j=0}^{N}|\bar{G}err(j)| \sum_{l=0}^{N} |\bar{G}uf(l)\ \ Ydev(iT-jT-lT)| \qquad (5-11)$$

A comment for equations 5-10 and 5-11 is that the key to calculating a tight bound is to be able to isolate the cause of the uncertainty as much as pcssible. If Gerr(z) only has gain uncertainty, α Gpnom(z), then a tighter bound for this case is:

$$\bar{Y}_{dev}(iT) = |\alpha| \sum_{j=0}^{N} |Gpnom(j)\ Umod(iT-jT)| \qquad (5\text{-}12)$$

$$+ |\alpha| \sum_{j=0}^{N} \left|\sum_{i=0}^{N} Gpnom(i)\ Guf(j-i)\right| \left|\bar{Y}_{dev}(iT-jT-iT)\right|$$

where $\alpha_1 < \alpha < \alpha_2$

The stability for the time-invariant systems (equations 5-10 and 5-11) can be checked by evaluating the criteria developed in Chapter 4. These criteria are:

Characteristic equation of equation 5-11:

$$1 + Gerr(z)\ Guf(z) = 0 \qquad (5\text{-}13)$$

or

$$1/Guf(z) + Gerr(z) = 0 \qquad (5\text{-}14)$$

$$F(z) = \Phi(z) + \Xi(z) = 0 \qquad (5\text{-}15)$$

where

$$\Phi(z) = 1/Guf(z) \qquad (5\text{-}16)$$

and

$$\Xi(z) \cdot Gerr(z) \qquad (5\text{-}17)$$

The interval polynomial, F(z) in equation 5-15 is stable if:

Theorem 1:

(i) the nominal polynomial, $\Phi(z)$ is stable

(ii) $|\Phi(e^{j\omega_k T})| > \Psi(\Xi(e^{j\omega_k T}))$ (4-16)

$\forall \quad \omega_k \quad 0 \leq k \leq N-1$

Corollary 1:

(i) the nominal polynomial, $\Phi(z)$ is stable

(ii) $\min\limits_{\omega_k}(|\Phi(e^{j\omega_k T})|) > \sigma(\Xi(z))$ (4-23)

Corollary 2:

(i) the nominal polynomial, $\Phi(z)$ is stable

(ii) $1/\mu(D(z)) > \sigma(\Xi(z))$ $(D(z) = 1/\Phi(z)) = Guf(z)$ (4-26)

Since Corollary 2 is simple and does not involve extensive frequency calculation, it was used exclusively in the subsequent analysis as a quick stability check.

5.4 Evaluation of the Robustness of Uncertain Control Systems

Having derived the EBM, the robustness of an uncertain control system can be analyzed by the following procedure:

1) Specify the nominal process model (equations 5-2, 5-3)

2) Specify the open loop model uncertainty (equations 5-2,5-3,5-4)

3) Select a tuning factor and design a digital controller (equations 5-6a,b;5-7a,b)

4) Evaluate the properties of closed loop uncertain system for a set point change. ($Y_{dev}(iT)$ from equations 5-6a,b, 5-7a,b, 5-10 and 5-11)

5) Repeat (3) and (4) until the performance specification is satisfied or the stability criteria fail. If the stability criteria fail, it implies that the uncertainty is too great for the controller.

To illustrate this procedure, a Dahlin controller was chosen as the model based controller for the simulation examples. Four cases were simulated:

Case A: first order process with 30% uncertainty (servo)

Case B: first order process with 30% uncertainty (regulatory)

Case C: first order plus dead time process with 30% uncertainty (servo)

Case D: first order plus dead time process with 30% uncertainty and fractional delay uncertainty (servo)

Closed loop step servo responses were generated in all cases except Case B, where a step disturbance was generated.

Case A: First order process with 30% uncertainty (Servo)

The process is a first order time-invariant process with a control interval (T) of 1 minute, and it is controlled by a Dahlin Controller, $G_c(z)$.

$$G_p(z) = \frac{0.1813\ z^{-1}}{1 - 0.8187\ z^{-1}} \qquad (5\text{-}18)$$

The continuous equivalent of $G_p(z)$ in equation 5-18 has a time constant (τ) of 5 minutes and a dead time(τ_d) of 0.0 minutes

a) assume that the open loop uncertainty is within 30% of the nominal process impulse response; i.e., $|\bar{G}_{err}(i)|$ is 30% of $|G_{pnom}(i)|$ (see Figure 5-3a).

$$G_{pnom}(z) = \frac{0.1813\ z^{-1}}{1 - 0.8187\ z^{-1}} \qquad (5\text{-}19)$$

$$|\bar{G}_{err}(i)| = 0.3\ |G_{pnom}(i)| \qquad \forall\ i=0,1,\ldots,N \qquad (5\text{-}20)$$

b) design a Dahlin controller, $G_c(z)$, with the desired closed loop time constant, $\lambda_{desired}$, the same as the open loop time constant, λ_{open},

$$\lambda_{open} = 5.0\ ;\quad \lambda_{desired} = 5.0$$

$$G_c(z) = \frac{1.000 - 0.8187z^{-1}}{1 - z^{-1}} \qquad (5\text{-}21)$$

c) Guf(z) and Gyf(z) are calculated (equations 5-7a,b)

$$Guf(z) = -1.0 \qquad (5\text{-}22)$$

$$Gyf(z) = \frac{1 - z^{-1}}{1 - 0.8187\ z^{-1}} \qquad (5\text{-}23)$$

d) use equations (5-10, 5-11) to generate the Ydev(iT) uncertainty band with a 10 unit set point change at time 2 minute.

The uncertainty band of the closed loop step response for this case is shown in Figure 5-3b. Depending on the designer's preference, the design can stop at this point and the controller, Gc(z), implemented on the actual process or the Dahlin tuning (λ) can be changed and procedure repeated.

Case B: First order process with 30% uncertainty (Regulatory)

This case is the same as Case A except that the Unom(iT) is different from the previous case. The disturbance is injected at D(iT) instead of R(iT). The size of the "step" disturbance is 10 unit and occurs at time 2 minute. Equations 5-10 and 5-11 were used to generate the envelope for Ydev(iT) for a step disturbance change. The uncertainty band is shown in Figure 5-3c. The band generated in this case is similar in shape to that for the set point change case because a step disturbance was used for the band generation. If the disturbance has low order dynamics, the band should be different from the set point case.

Case C: First order plus dead time process with 30% uncertainty (Servo)

The process for this case is a time-invariant, first order plus dead time process with a control interval (T) of 1 minute,

$$Gp(z) = \frac{0.1813\ z^{-2}}{1 - 0.8187\ z^{-1}} \qquad (5\text{-}24)$$

The process in equation 5-24 has a time constant (τ) of 5 minutes and a dead time (τd) of 1 minute. The dead time to time constant ratio ($\tau d/\tau$) in this case is 0.2. This represents a relatively easy control problem. The process is again controlled by a Dahlin Controller, $G_c(z)$.

a) assume that the open loop uncertainty is within 30% of the nominal process impulse response; i.e., $|\bar{G}_{err}(i)|$ is 30% of $|G_{pnom}(i)|$ (see Figure 5-4a).

$$Gpnom(z) = \frac{0.1813\ z^{-2}}{1 - 0.8187\ z^{-1}} \qquad (5\text{-}25)$$

$$|\bar{G}_{err}(i)| = 0.3\ |G_{pnom}(i)| \qquad \forall\ i=0,1,\ldots,N \qquad (5\text{-}26)$$

b) design a Dahlin controller, $G_c(z)$, with the desired closed loop time constant, $\lambda_{desired}$, same as the open loop time constant, λ_{open},

$$\lambda_{open} = 5.0 \; ; \quad \lambda_{desired} = 5.0$$

$$G_c(z) = \frac{1.000 - 0.8187z^{-1}}{1.00 - 0.8187z^{-1} - 0.1813z^{-1}} \tag{5-27}$$

c) $G_{uf}(z)$ and $G_{yf}(z)$ are calculated (equations 5-7a,b)

$$G_{uf}(z) = -1.0 \tag{5-28}$$

$$G_{yf}(z) = \frac{1 - 0.8187\ z^{-1} + 0.1813\ z^{-2}}{1 - 0.8187\ z^{-1}} \tag{5-29}$$

d) Use equations 5-10 and 5-11 to generate the Ydev(iT) uncertainty band with a 10 unit set point change occurring at time 2 minute. The resulting band is shown in Figure 5-4b.

Case D: First order plus dead time process with 30% uncertainty and fractional dead time uncertainty (Servo)

In Case C, it was assumed that there was no uncertainty in the process dead time. In this case, the dead time uncertainty is defined by the impulse weight at lag 2. It is between Gpmax(2) and Gpmin(2) while later the impulse weights are 30% of the nominal process impulse response:

$$Gpmax(2)=2.0 \qquad Gpmin(2) = 0.0 \tag{5-30}$$

and

$$Gerr(i) = 0.3|\ Gpnom(i)| \qquad \forall\ i=3,4...N \tag{5-31}$$

If the actual dead time is less than the nominal dead time, there would be significant value in the second lag in the impulse weight. This is the scenario simulated in Case D. Figure 5-5a shows the open loop uncertainty for this case. The controller was a Dahlin controller with the same tuning as Case C. The closed loop uncertainty band for this case is shown in Figure 5-5b.

Cases A,B,C and D cover a wide class of commonly occurring processes and process uncertainties. Notice that the uncertainty is chosen to be a percentage of the nominal process response in this study. This is mainly for comparison purposes. In order to have a meaningful robustness comparison, it has to be based on the same amount of uncertainty. Using a percentage of the nominal process response as the open loop uncertainty provides a convenient way of quantifying uncertainty. The EBM uncertainty characterization works very well for percentage uncertainty and fractional delay uncertainty. However, if there is pure delay uncertainty, one has to assign Gerr(i)'s throughout the uncertain delay periods. As a result, the uncertainty may become too large for EBM. The characterization of delay uncertainty is an area needed to be improved in the future work.

The closed loop uncertainty band generated in these cases are supposed to include all the possible closed loop responses. The wider the band, the less robust the controller. To quantify the size of the closed loop uncertainty band, a robustness index, IAE (Integral Absolute Error) is defined here as:

$$IAE = \sum_{i=0}^{L} |\max(Ydev(iT))| + |\min(Ydev(iT)| \qquad (5\text{-}32)$$

where L is the simulation time
(100 minutes in this study)

Since IAE is the absolute sum of the range of Ydev(iT) over time, the larger the value of IAE, the bigger the deviations from the nominal response. In the subsequent analysis, this index will be used as an indicator of control system robustness.

In addition to the Ymax and Ymin from EBM, the Monte-Carlo Simulation bounds were also calculated for comparison (see Figures 5-6a,b,c). The band for the Monte-Carlo simulation was generated by running a Monte-Carlo simulation on Gerr. Each Gerr(i)'s was chosen randomly to be either side of the corresponding bound values, $\bar{G}err(i)$. Since the Gerr(i)'s were then known, the closed loop responses, Y(iT), could be calculated using equation 5-1a. In each Monte-Carlo simulation, 100 sets of randomly chosen uncertainties were simulated for each case. The Monte-Carlo bounds represent the maximum and minimum Y(iT)'s at each time step i over all 100 simulations.

5.5. Applications of the EBM

The procedure of getting a closed loop uncertainty band for the process output was given in section 5.4. This section goes on to explore the applicability of the EBM. Two applications have been studied; one in the tuning of digital controllers and the other in the comparison of the robustness of digital controllers.

5.5.1 Tuning of digital controllers

One can improve the robust performance of an uncertain control system by tuning the controller. To illustrate this procedure, Case A was repeated with the Dahlin tuning parameter set to λ=2.5 min. λ=5.0 min and λ=10.0 min. The closed loop uncertainty band calculation was repeated and the results were shown in Figure 5-6a, 5-6b and 5-6c together with the Monte-Carlo simulation bands. The combined band results are plotted in Figure 5-7 to illustrate the effect of tuning on the closed loop uncertainty band. Figure 5-8 shows the uncertainty band index (IAE) versus the Dahlin tuning. Gradually decreasing the desired closed loop time constant will also gradually decrease the size of the envelope until a minimum is reached. This indicates that there is a tuning which is guaranteed stable and provides the best closed loop performance for the specified open loop uncertainty. One can then use this as an initial tuning for the real process. Notice the general shape of the curve in Figure 5-8. The closed loop uncertainty can be reduced with high gain feedback (Fast Dahlin tuning). However, an excessively high gain pushes the control system towards instability and hence the uncertainty band starts to increase after the optimal tuning value.

Figures 5-6a and 5-6b show that the bounds predicted by the EBM and those found by Monte-Carlo simulation bounds are quite close. Theoretically, the EBM bounds should always include the Monte-Carlo bounds because the EBM should include all possible closed loop responses. Figure 5-6c, which corresponds to a Dahlin tuning of λ=2.5 minutes, shows that the Monte-Carlo bounds not only lie within the EBM bounds, but the Monte-Carlo bounds are also much smaller than the EBM bounds. The difference between Figures 5-6a, 5-6b and 5-6c is that the former two cases have a Dahlin tuning slower or equal to the nominal open loop time constant while the latter case is tuned twice as fast as the nominal open loop time constant. A Dahlin controller, which is tuned slower than the nominal open loop time constant, is referred to here as a non-amplifying controller. Conversely, a controller with a faster tuning than the nominal open loop time constant is referred to as an amplifying controller. Notice that the Monte-Carlo simulation used the bound values of Gerr(i). The reasoning behind this choice was that the worst Y(iT) would correspond to the bound values of the uncertainty description. The results shown in Figure 5-6c seem to contradict this assumption. It seems to correspond to the fact that the stability of interval polynomial cannot in general be checked by the vertex polynomial (Huang et al. 1987). This phenomenon was also observed in Bartlett (1990) in which he suggests that "the vital statistics of the transient response of a stable discrete-time uncertainty system cannot be completely determined from the response of the vertex description". A sequence of work was initiated as a result of this finding on the bound.

One attempt was to redesign the Monte-Carlo experiment so that the Gerr(i) did not necessarily take on either the upper and lower bound values. The randomization was designed such that Gerr(i) can take on one of 10 values evenly spaced between the upper and lower bounds. The Monte-Carlo experiments were repeated. However, even with these changes, there is no obvious increase in the size of Monte-Carlo bound in the case of the amplifying controller. This suggests that either the Monte-Carlo simulation does not happen to include the worst case or the Monte-Carlo simulation bounds are correct and the EBM simply does not produce a tight band for an amplifying controller. To further explore this issue, an effort was made to verify the tightness of EBM bounds. An optimization was set up to find the worst Y(iT) at each time step. The problem is formulated as follows:

$$\underset{Gerr(i),\, i \in R^N}{\text{Minimize}} \qquad Ydev(iT) \tag{5-33}$$

Subject to equations 5-1a, 5-2, 5-3, 5-4

At each time step i, the optimization was performed to find the minimum of Ydev(iT). The maximization of Ydev(iT) can be obtained by changing the objective to minimize -Ydev(iT) instead. The optimization package used was NPSOL from Stanford University. During the optimization, it is observed that the solution sometimes converges to local minima. This implies that the resulting optimized bounds may not be global optimum and hence may not include all the possible closed loop responses. However, the optimization result which is plotted in Figure 5-9 can still shed some light on the tightness of the EBM bounds. As noted in Figure 5-9, the Optimization bound is wider than the Monte Carlo simulation bounds but not to a large extent. The IAE's for the Monte Carlo bounds, EBM bounds and Optimization bounds are: 23.5, 86.9 and 32.3 respectively. The EBM bounds are still far from the optimized bounds. This implies that the EBM does not produce a tight bound for an amplifying controller. Although an extensive effort was made to find a tighter bound for an amplifying controller, the issue still has not been resolved. The problem lies on equation 5-11.

$$\bar{Y}_{dev}(iT) = \sum_{j=0}^{N} \left|\bar{G}_{err}(j)\right| \left|U_{mod}(iT-jT)\right| + \sum_{j=0}^{N}\left|\bar{G}_{err}(j)\right| \sum_{l=0}^{N} \left|G_{uf}(l)\ \bar{Y}_{dev}(iT-jT-lT)\right| \qquad (5\text{-}11)$$

The uncertainty portion, Gerr(i), cannot be further isolated in equation 5-11. Therefore, the uncertainty in Ydev(iT) in the early stages of the response will start to accumulate over time and make Ydev(iT) much larger than it should be. This is particularly true for an amplifying controller, where Guf(i) can take on mixed '+' and '-'

signs. However, for a non-amplifying controller, Guf(i), is either all '+' or all '-'. This explains why the EBM bounds are tight for a non-amplifying controller. This provides grounds for further research to explore the bounding procedure for an amplifying controller so that the EBM approach can become more complete. Although the EBM bounds were not tight for amplifying controllers, this should not hinder the progress of the EBM because robust designs produce non-amplifying controller in many instances. Before a complete solution for the EBM is available, one has to be cautious when dealing with amplifying controllers because the EBM bounds for amplifying controllers are conservative.

In Figure 5-10a, the IAE is plotted as a function of the Dahlin tuning parameter and the open loop uncertainty. The process in Case C is used in this analysis. The $\tau d/\tau$ ratio is 0.2 which is an easy control problem. To explore the effect of a more difficult control problem, two new cases are created.

Case E :

The process (see equation 5-34) for this case has a time constant (τ) of 1 minute and a deadtime (τd) of 2 minutes. Therefore the $\tau d/\tau$ ratio is 2.0 which is a more difficult process to be controlled.

$$Gp(z) = \frac{0.632\ z^{-3}}{1 - 0.368\ z^{-1}} \qquad (5\text{-}34)$$

The uncertainty is specified as a percentage of the nominal process.

Case F:

The process is the same as that in Case E. However, the uncertainty contains fractional delay uncertainty and percentage uncertainty. Similar to Case D, the fractional delay uncertainty is :

$$\text{Gpmax}(3) = 0.316 \qquad \text{Gpmin}(3) = 0.0 \tag{5-35}$$

and

$$|\overline{\text{G}}\text{err}(i)| = \alpha\ |\text{Gpnom}(i)| \qquad \forall\ i = 4,5,\ldots N \tag{5-36}$$

The design procedures for Cases C were repeated for these new cases. The IAE's were again calculated for different levels of uncertainties and different values of λ. The results were plotted in Figures 5-10b and 5-10c. Note that the regions in the upper left corner of Figures 5-10a,b,c are the unstable regions according to the stability criteria in Chapter 4. Figures 5-10a,b,c suggest that for open loop uncertainty less than 100% of the nominal process impulse response, there is always a minimum in the IAE. A more interesting observation is that all the minimum points lie to the left of the open loop time constant : in Figure 5-10a, the minimum lies to the left of λ=5.0 minutes, which is the open loop time constant for processes in Cases A,B,C and D and in Figure 5-10b. the minimum points lie to the left of the open loop time constant (λ = 1.0 minute) of the process in Case E. However, the minimum points are not to the left of λ in Figure 5-10c which is the case with fractional delay uncertainty. This observation suggests that, as far as the performance is concerned, for process with no delay uncertainty and the uncertainties less than 100% of the nominal process there is no point in detuning a Dahlin controller more than the open loop time constant. There will be no gain in stability and performance will be lost due to the detuning. This observation has also been confirmed by some Dahlin controller practitioners who found that setting the tuning factor to the open loop time constant is a very robust starting point for tuning.

5.5.2 Comparing the robustness of different controllers

In Figures 5-10a,b,c, there is a minimum IAE associated with each level of open loop uncertainty. Min IAE refers to this minimum IAE value. The value shows the minimum IAE that can be achieved by the tuning procedure described in Section 5.5. Figure 5-11 displays the Min IAE as a function of the level of uncertainty. This curve is a characteristic of a digital controller. Therefore, by comparing the Min IAE at different levels of open loop uncertainty, one is in effect comparing the robustness characteristics of different digital controllers.

In this application study, the process in Case C was used. From Figures 5-10a, if one joins all the points which correspond to the minimum IAE at different levels of uncertainty in Figure 5-10a, one obtains Dahlin Min IAE curve in Figure 5-11. This curve indicates the robustness characteristic of a Dahlin controller. The PI controller Min

IAE curve can be obtained by using the same type of analysis used for the Dahlin controller, except that in this case, a PI controller is used in the analysis. This PI controller is designed based on Dahlin Design approach. A quarter of the process dead time is added to the process time constant. This modified time constant together with the process model gain are used in the normal Dahlin Design procedure. For a first order process with no dead time, the Dahlin Design procedure always results in a PI controller with the appropriate tuning parameters for the specified closed loop response.

As shown in Figure 5-12, the PI Min IAE curve is slightly higher than the Dahlin Min IAE curve at all levels of uncertainty. This suggests that a Dahlin controller can always guarantee a lower IAE than a PI controller. In another word, in face of the "worst case" process, the performance of a Dahlin controller is slightly better than that of a PI controller. It is only slightly better because the PI structure is nearly optimal for this process; the dead time to time constant ratio being only 0.2. The same analysis in Figure 5-12 is performed for the process in Case E. In this case, the $\tau d/\tau$ ratio is 2.0 which represents a more difficult control problem. It is expected that a Dahlin controller should perform much better than a "under-structure" PI controller. Figure 5-13 shows that the IAE for a PI controller is much higher than that for a Dahlin controller, hence confirming the fact that Dahlin controller is better than a PI controller. This observation is not a big surprise. It is expected that an advanced digital controller, with enough structure and properly tuned, should perform better than an "under-structure" controller (PID) (Laughlin et al. 1986). This is the essence of robust controller design : to preserve the optimality of model based controller while still maintaining comparable stability characteristic as a simple PID controller.

5.6. Conclusion

The development of a time domain closed loop uncertainty analysis method, the Error Band Method (EBM), has been presented in this chapter. The closed loop uncertainty band is a very intuitive criterion and it gives a visual indication of the consequence of the open loop uncertainty. The stability criterion is simple; only two gain terms need to be evaluated. The performance is assessed through the closed loop servo output uncertainty band. This uncertainty band is calculated from simple difference equations. The robust performance of the Dahlin Controller has been studied using the EBM. The simulation results show that the EBM gives tight bounds for a non-amplifying controller but not for an amplifying controller. This implies that the EBM gives a conservative result for amplifying controllers. A method for tuning digital controllers has been developed from the generated uncertainty band. Based on the EBM, it was found that for a system with no delay uncertainty and the uncertainty less than 100% of the nominal process impulse response, there is no point in detuning more than the open loop time constant of the process. Furthermore, it was found that a model based digital controller, if properly tuned, can possess a more robust performance than a simple PID controller. As indicated in Section 5.4, EBM works very well for fractional delay uncertainty. However, for systems with integral delay uncertainty, the current uncertainty characterization is not adequate. This warrants future research which should focus on tightening the closed loop uncertainty bands and better characterizing integral delay uncertainty.

5.7. Remarks on Robustness

As a closing comment, it should be emphasized that all "guaranteed" types of robustness analysis are basically "worst case" analyses. It may happen that one may sacrifice the entire analysis for one single "worst case" which may be highly improbable. Therefore, the robustness analysis or design should be treated not as a rule that one must religiously follow but rather as a guideline. Furthermore, should the robustness criteria fail, the controller design, the performance criteria and even the process configuration should be jointly considered in the subsequent analysis and redesign.

5.8. Nomenclature

T : Sampling Interval

N : Length of impulse vector

λ : Dahlin tuning factor (Desired Closed Loop Time Constant

L : Simulation time

Impulse weights -

Gpp(i) : nominal impulse weight at lag i

Gpmax(i): maximum impulse weight at lag i

Gpmin(i): minimum impulse weight at lag i

Guf(i) : nominal process impulse weight at lag i from uncertainty disturbance to controller output

Gyf(i) : nominal process impulse weight at lag i from uncertainty disturbance to process output

Gerr(i) : error bound weight at lag i

$\overline{\text{Gerr}}(i)$: max error bound weight at lag i

Signals -

R(iT) : set point input at time iT

D(iT) : process disturbance at time iT

U(iT) : nominal controller output at time iT

Y(iT) : nominal process output at time iT

F(iT) : uncertainty disturbance generated by nominal control input U(iT) at time iT

Ydev(iT) : output deviation generated by uncertainty disturbance at time iT

Unom(iT) : nominal input closed loop response at time iT

Ynom(iT) : nominal output closed loop response at time iT

- : a "-" on top of signal means the maximum bound of the signal

Operator -

|Guf(z)| : take absolute value of all weights in the impulse response

$\sigma(Guf(z))$: this denotes the supremal gain of $Guf(z)$

$$\sigma(Guf(z)) = \sup_{|z|=1} (Guf(z)) = \sum_{j=0}^{N} |Guf(j)|$$

$|Guf(i)|$: take absolute value of impulse weight $Guf(i)$

$\mu(G(z))$: spectral gain of $G(z)$

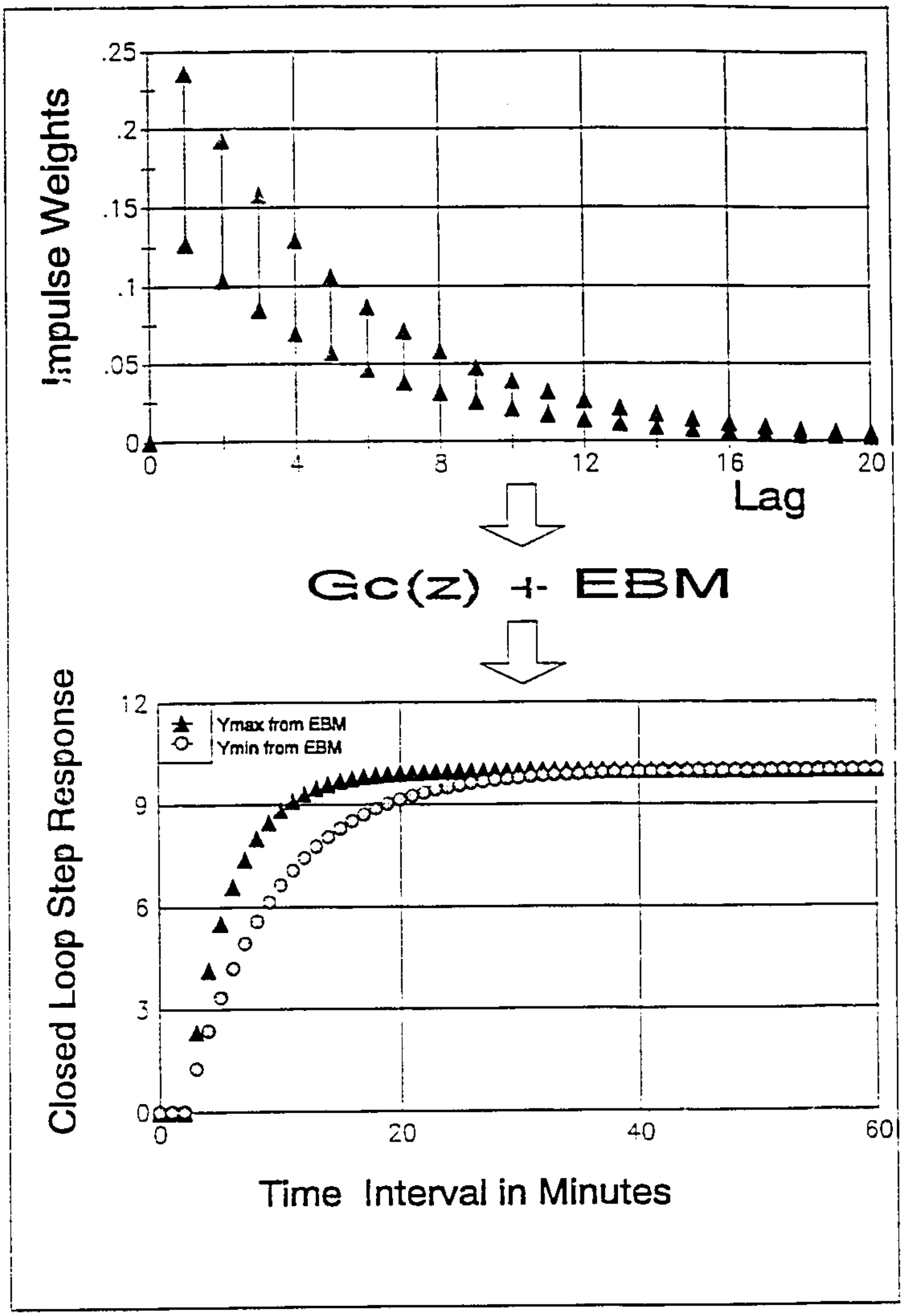


Figure 5-1: Overview of Error Band Method (EBM)

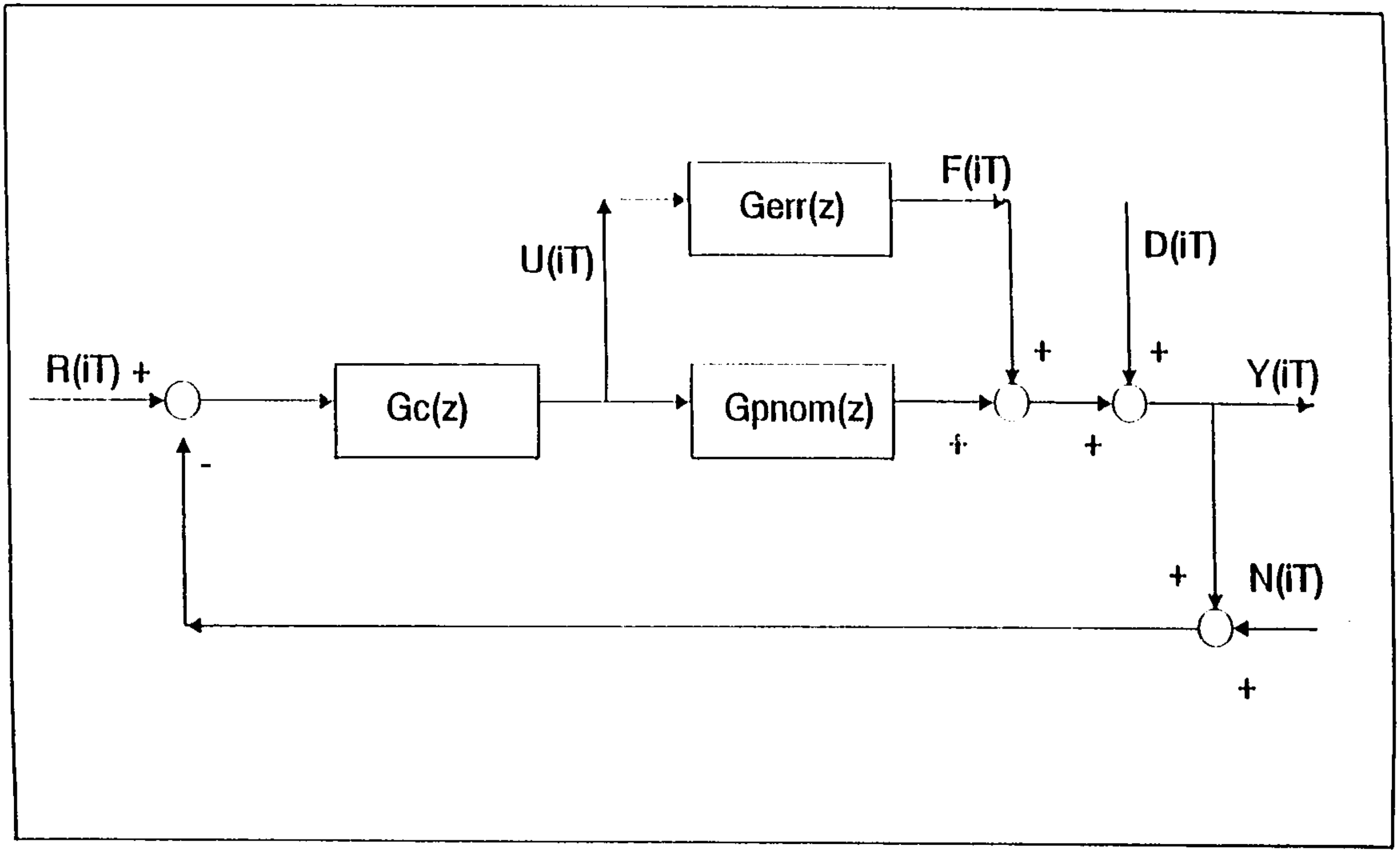


Figure 5-2a: Conventional Control Block Diagram

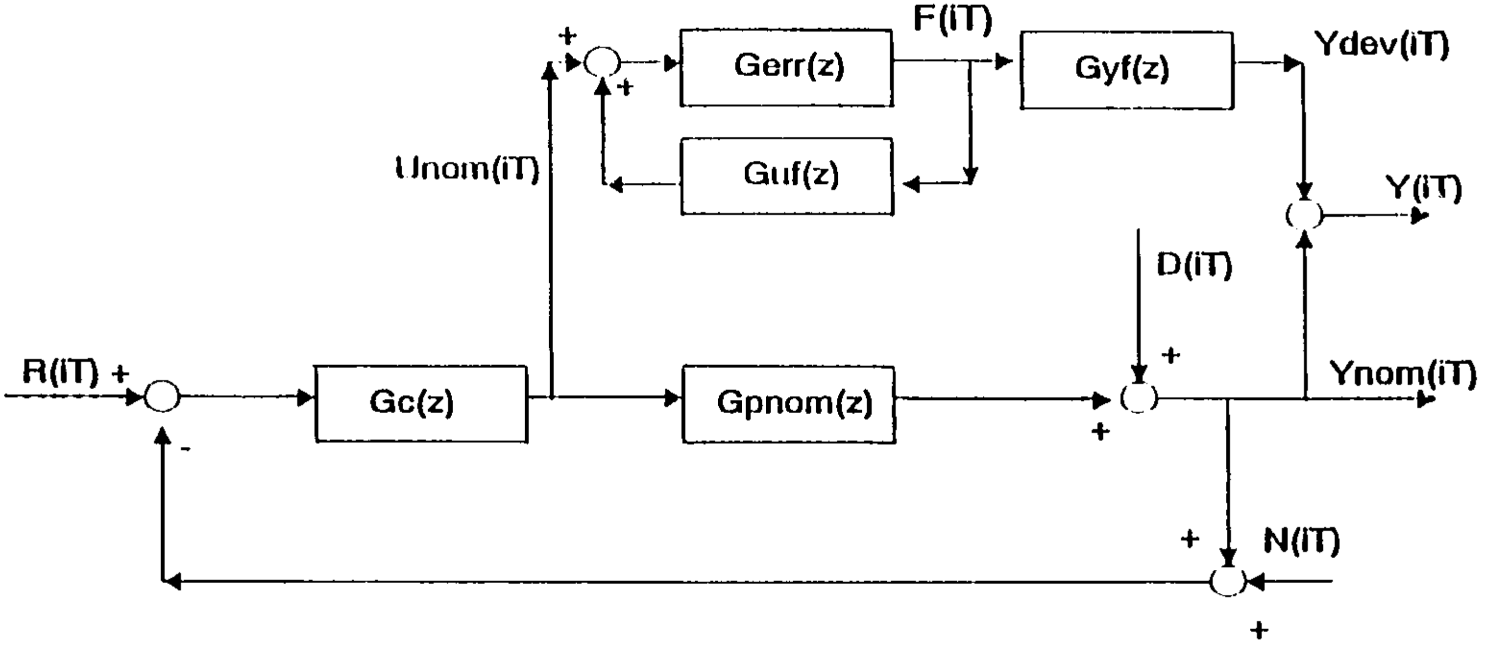


Figure 5-2b: Modified Control Block Diagram

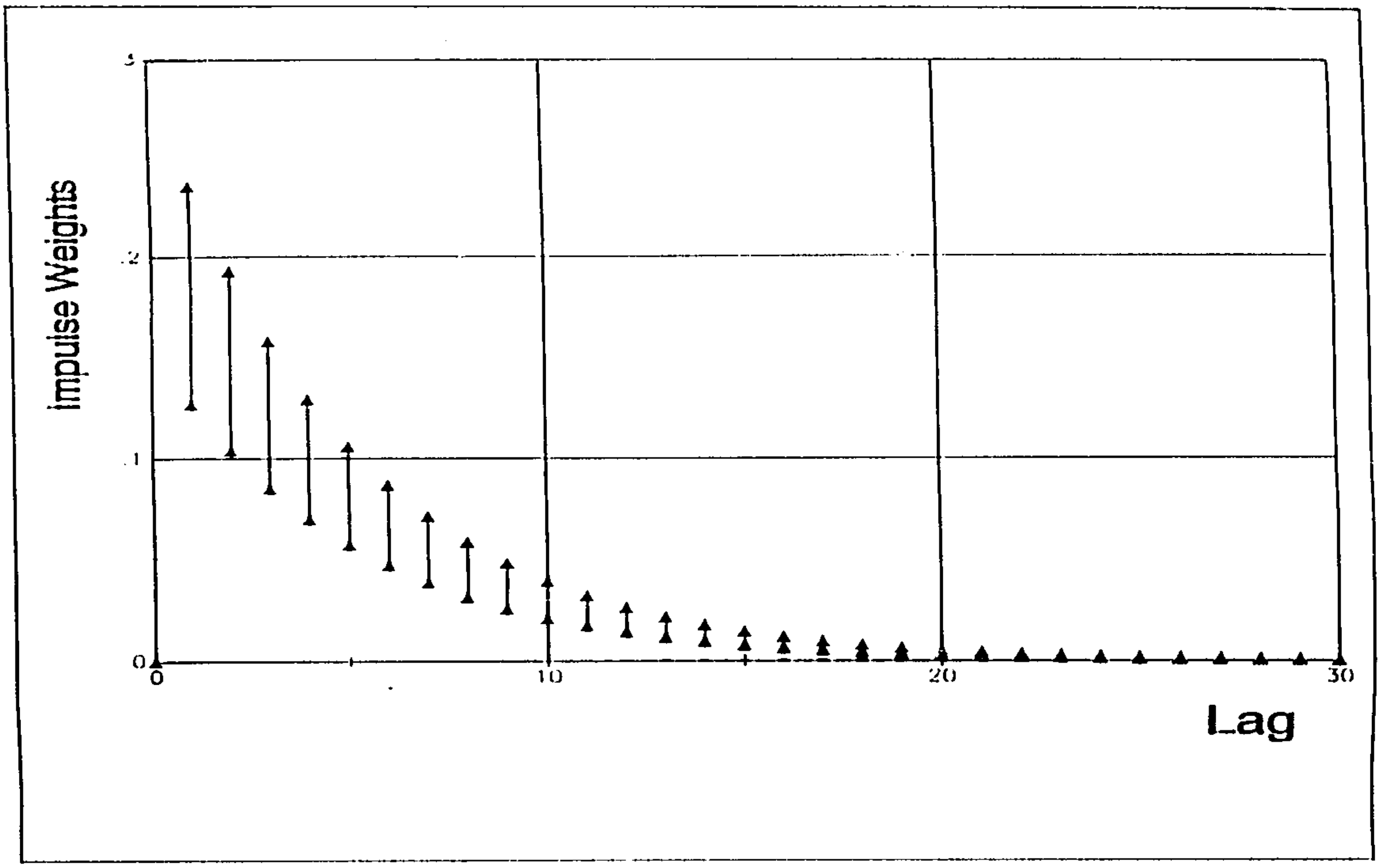


Figure 5-3a: Open Loop Uncertainty ; Uncertainty Level 30%

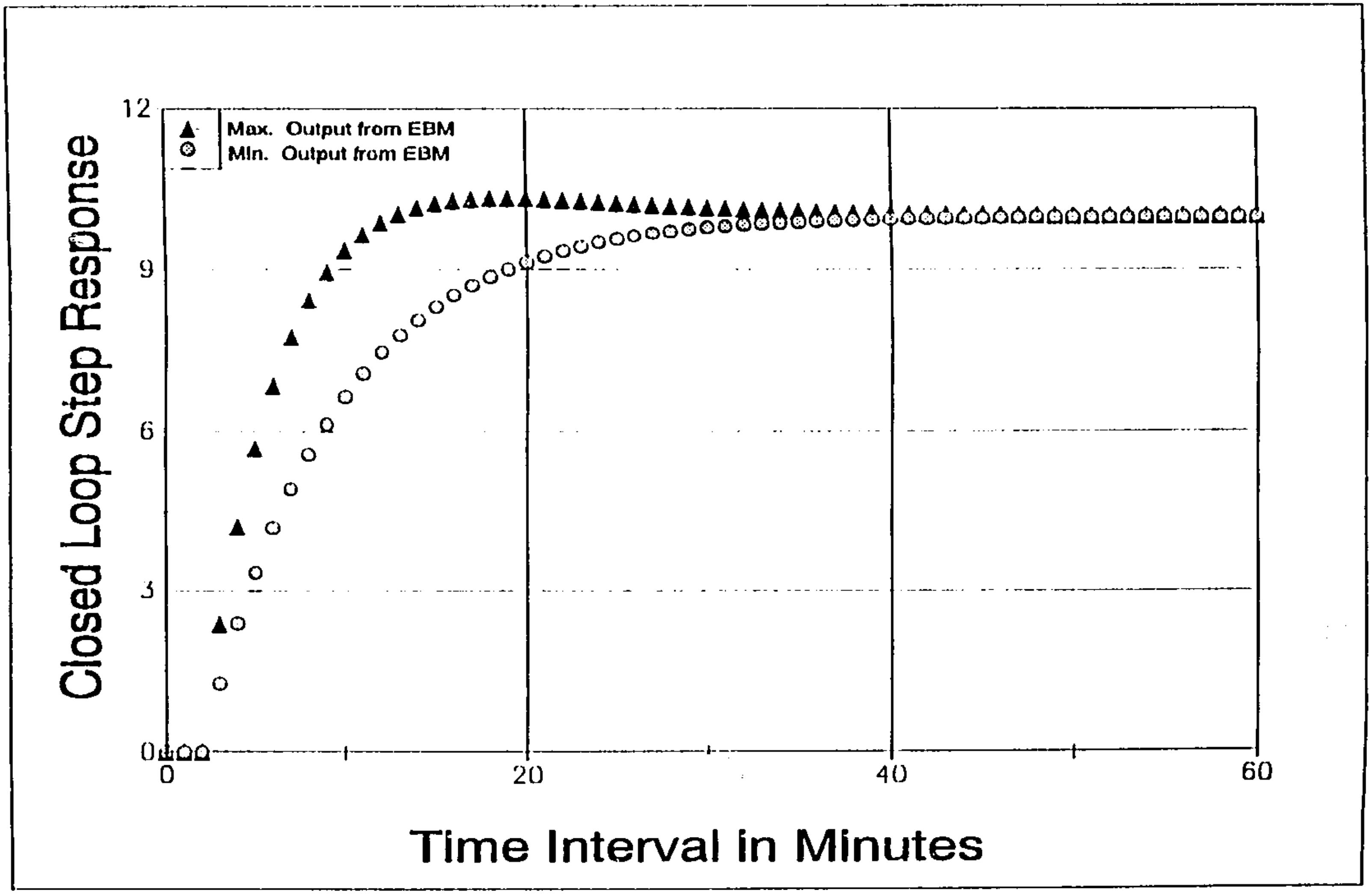


Figure 5-3b: Closed Loop Servo Response Uncertainty Band ; Uncertainty Level 30% λ= 5 minutes

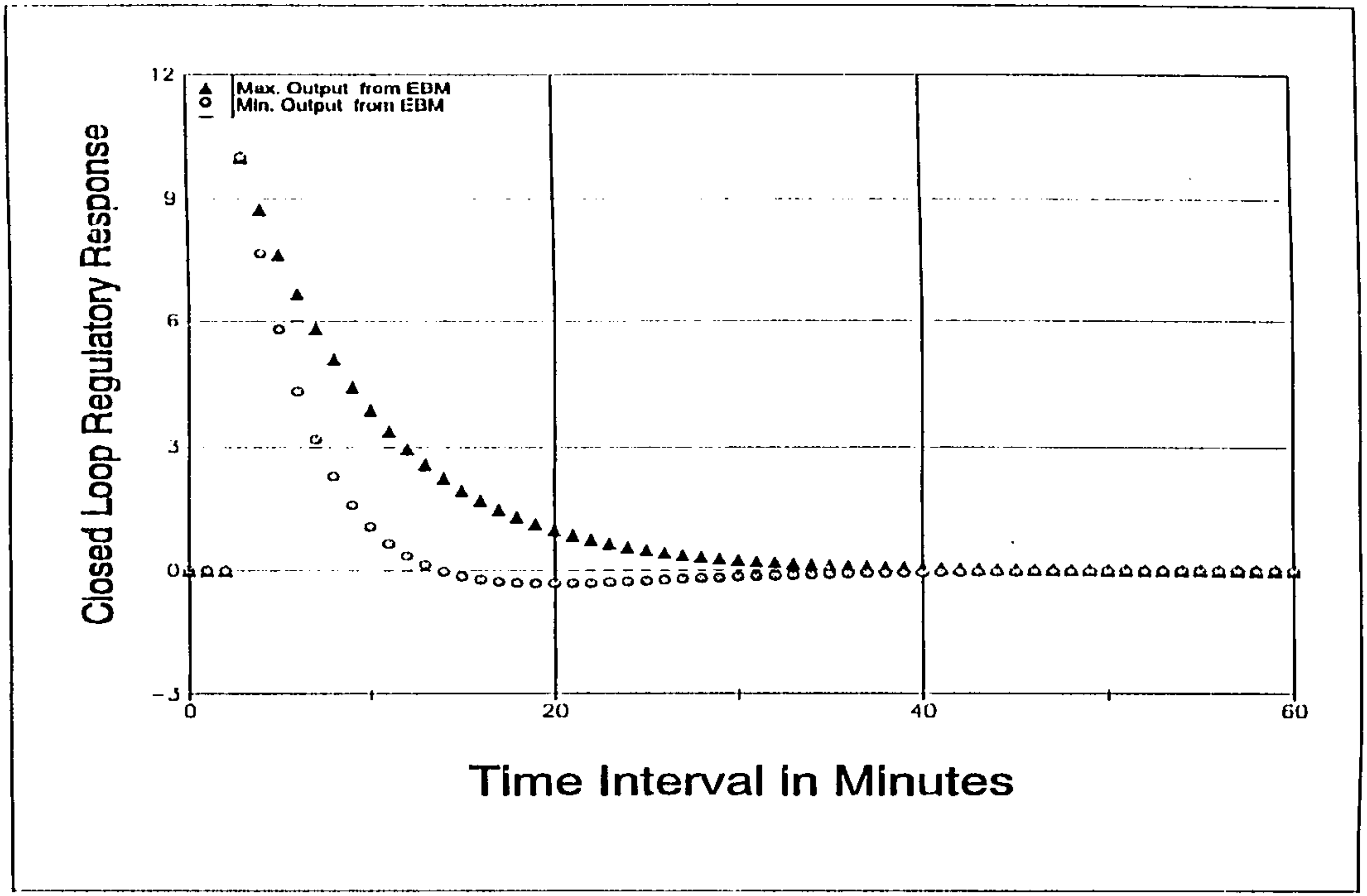


Figure 5-3c: Closed Loop Uncertainty Regulatory Response Band ; Uncertainty Level 30% λ= 5 minutes

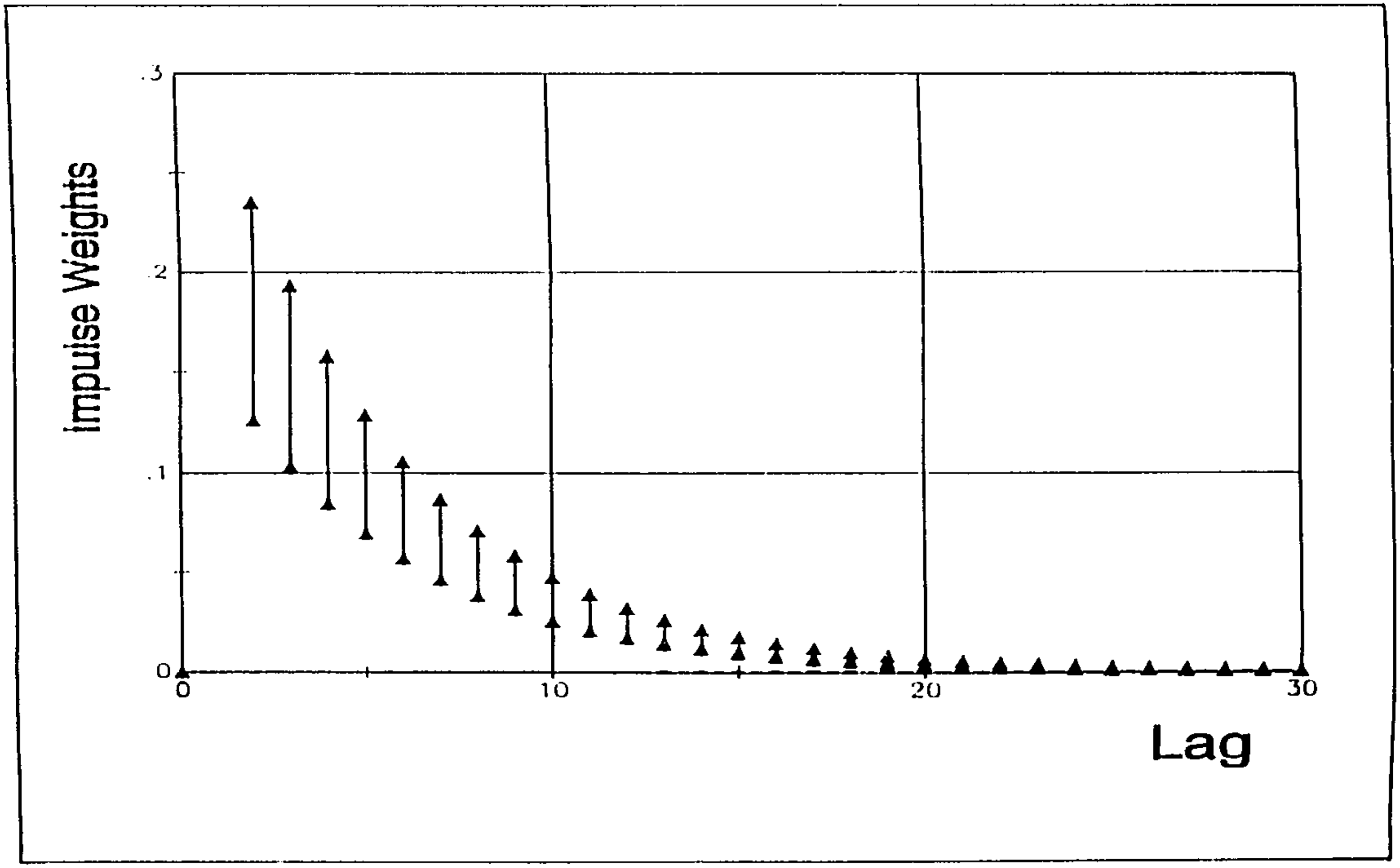


Figure 5-4a: Open Loop Uncertainty for Dead Time Sysytem ; Uncertainty Level 30%

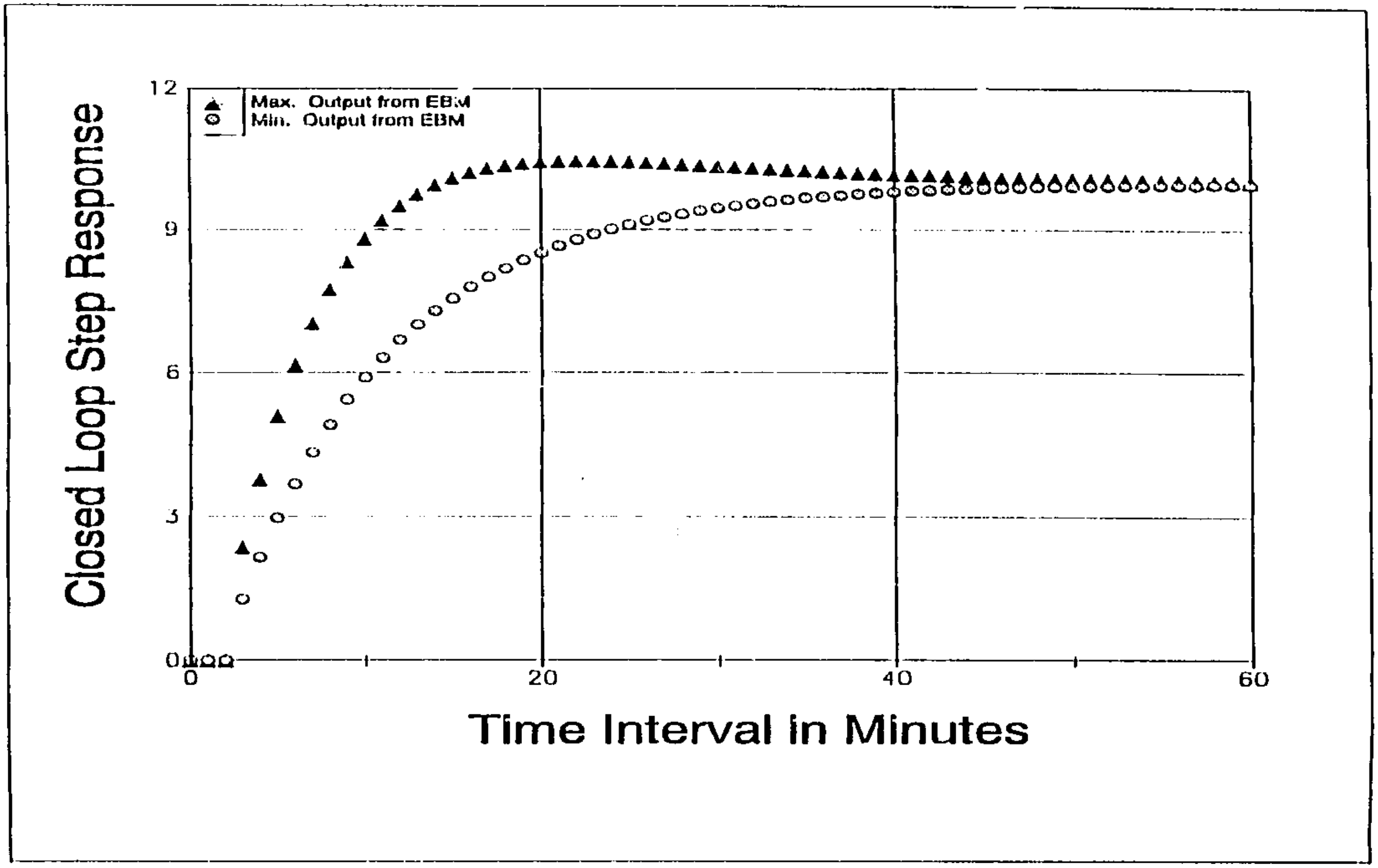


Figure 5-4b: Closed Loop Servo Response Uncertainty Band for Dead Time System Uncertainty Level 30% λ= 5 minutes

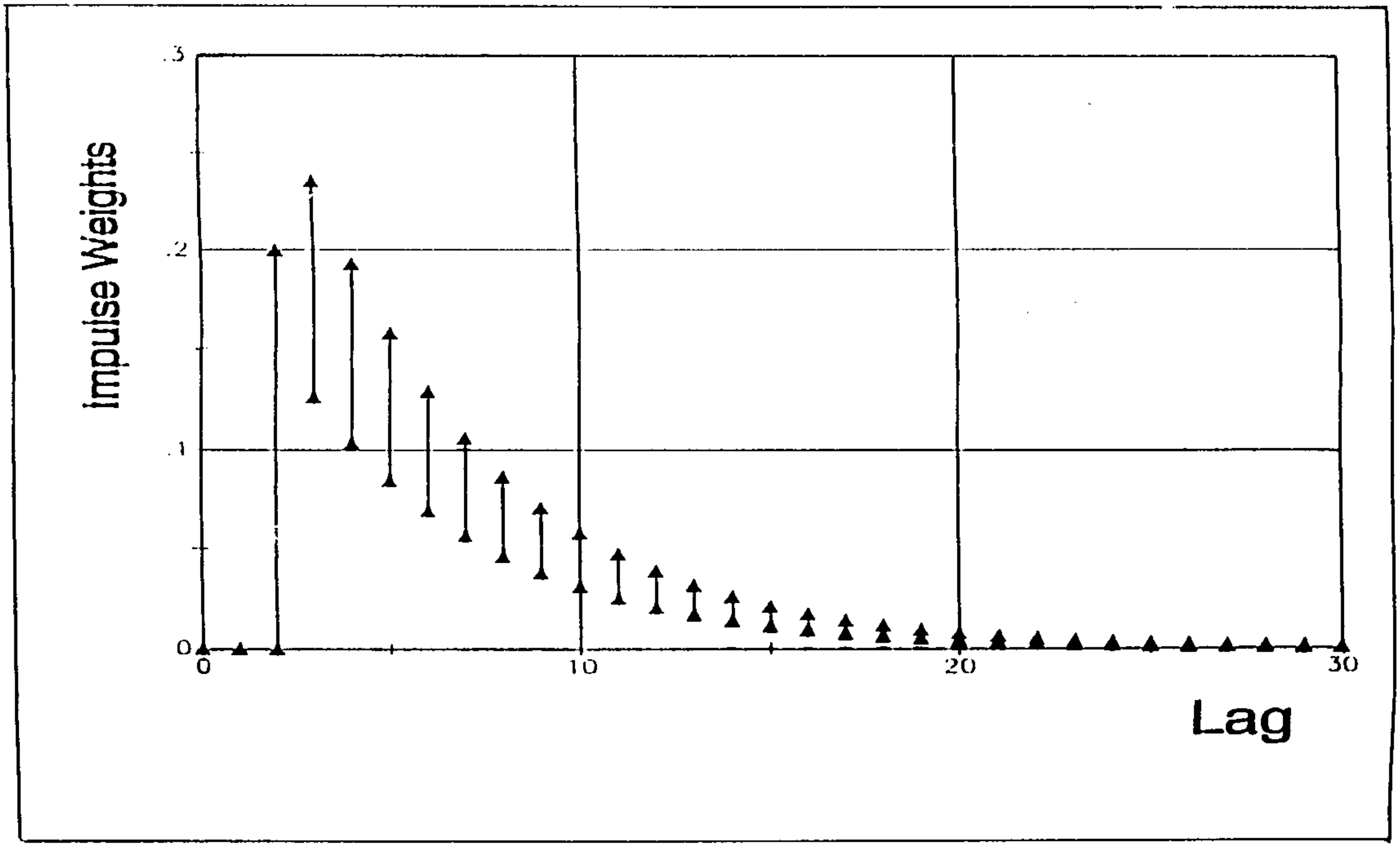


Figure 5-5a: Open Loop Uncertainty for Dead Time System with Deadtime Uncertainty Uncertainty Level 30%

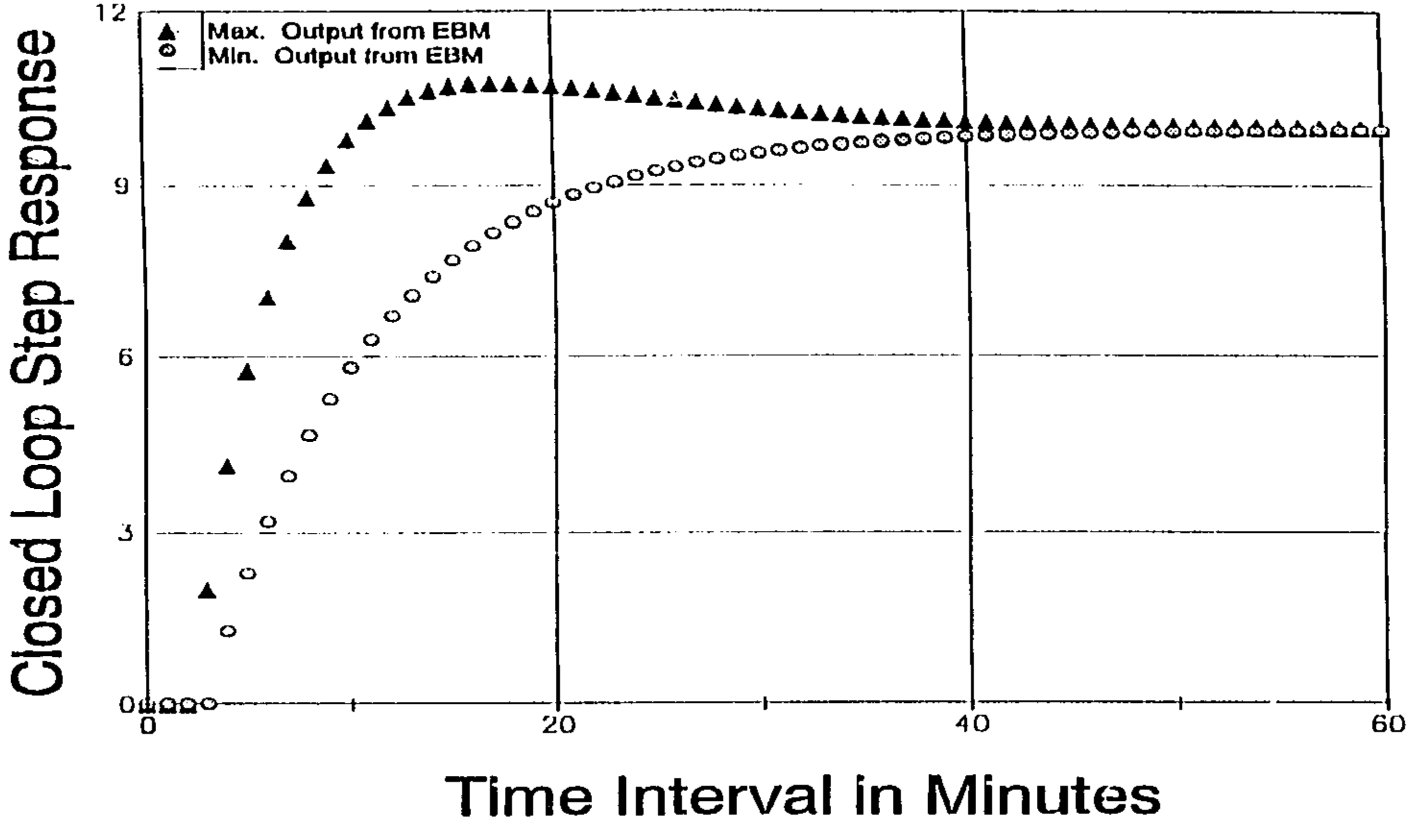


Figure 5-5b: Closed Loop Servo Response Uncertainty Band for Dead Time System with Dead Time Uncertainty ; Uncertainty Level 30% λ= 5 minutes

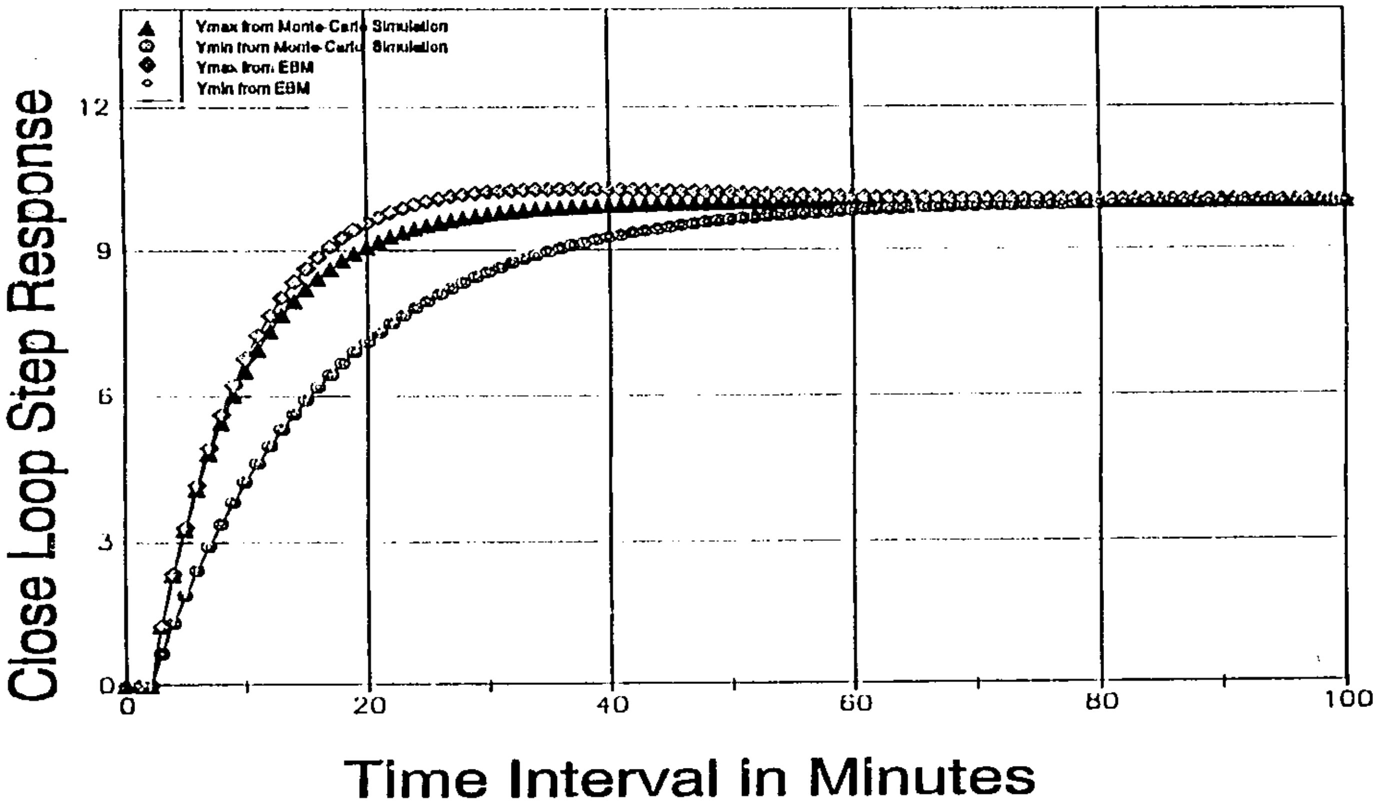


Figure 5-6a: Closed Loop Servo Response Uncertainty Band ; Uncertainty Level 30% λ= 10.0 minutes

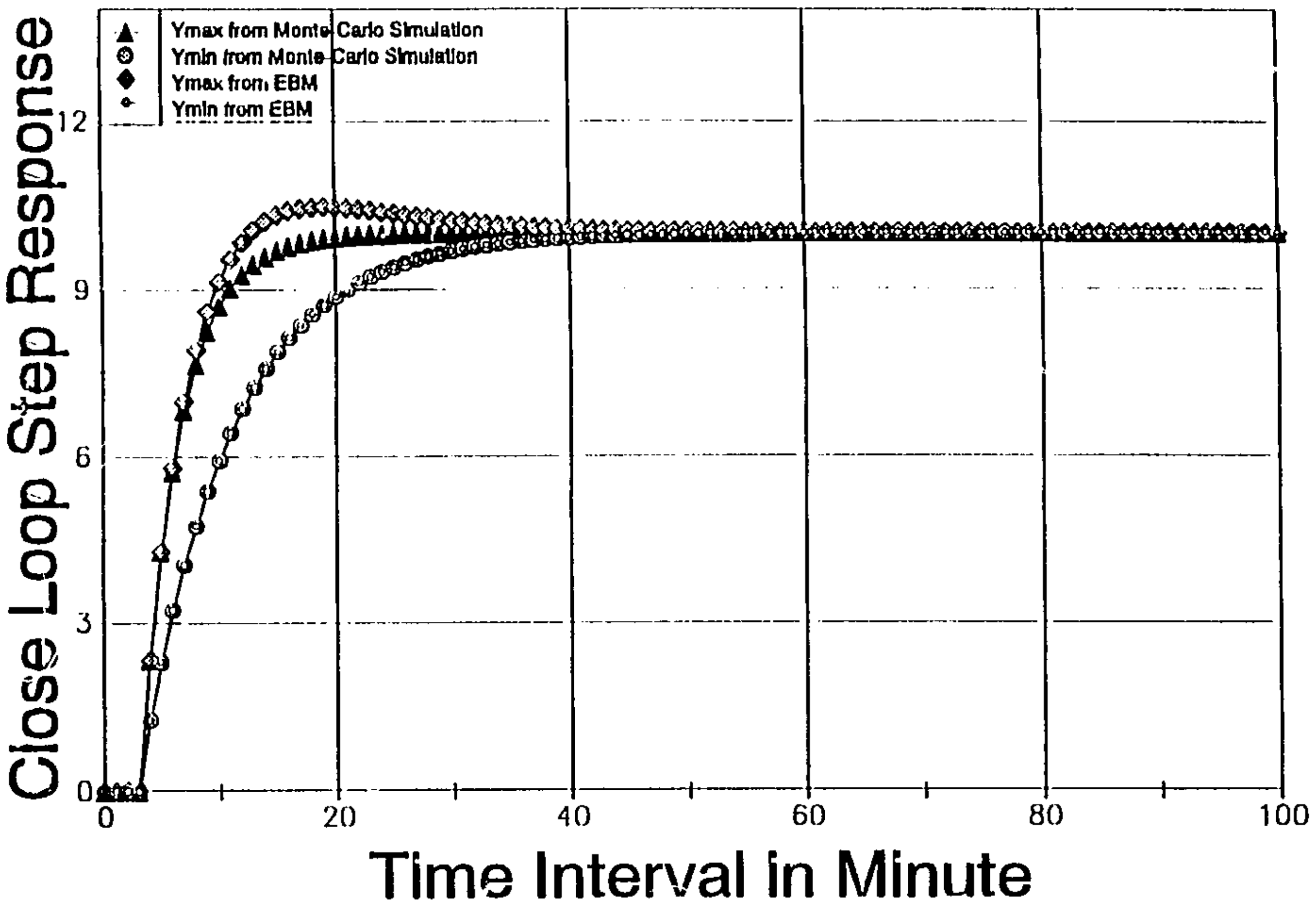


Figure 5-6b: Closed Loop Servo Response Uncertainty Band ; Uncertainty Level 30 % λ=5 minutes

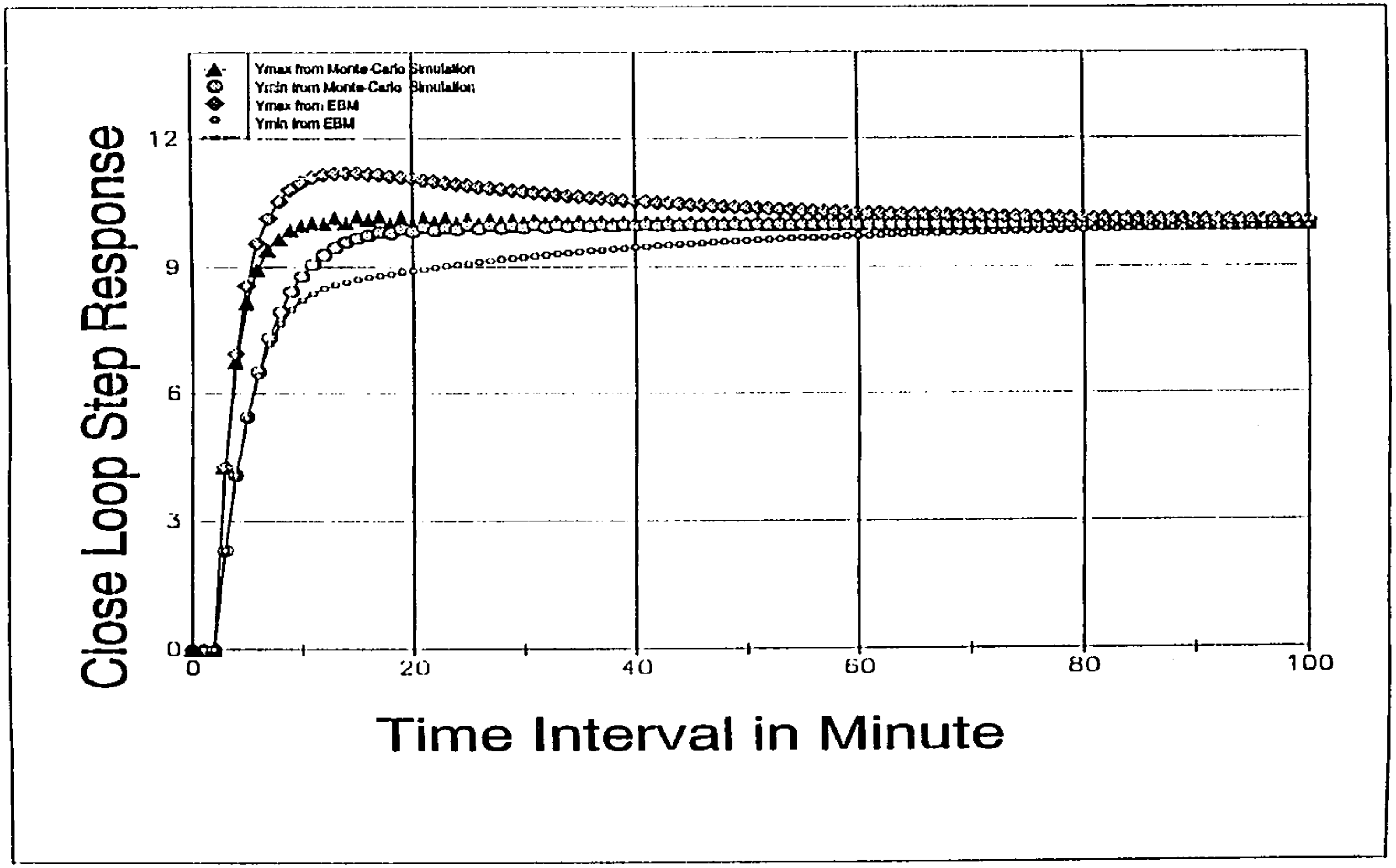


Figure 5-6c: Closed Loop Servo Response Uncertainty Band ; Uncertainty Level 30% λ= 2.5 minutes

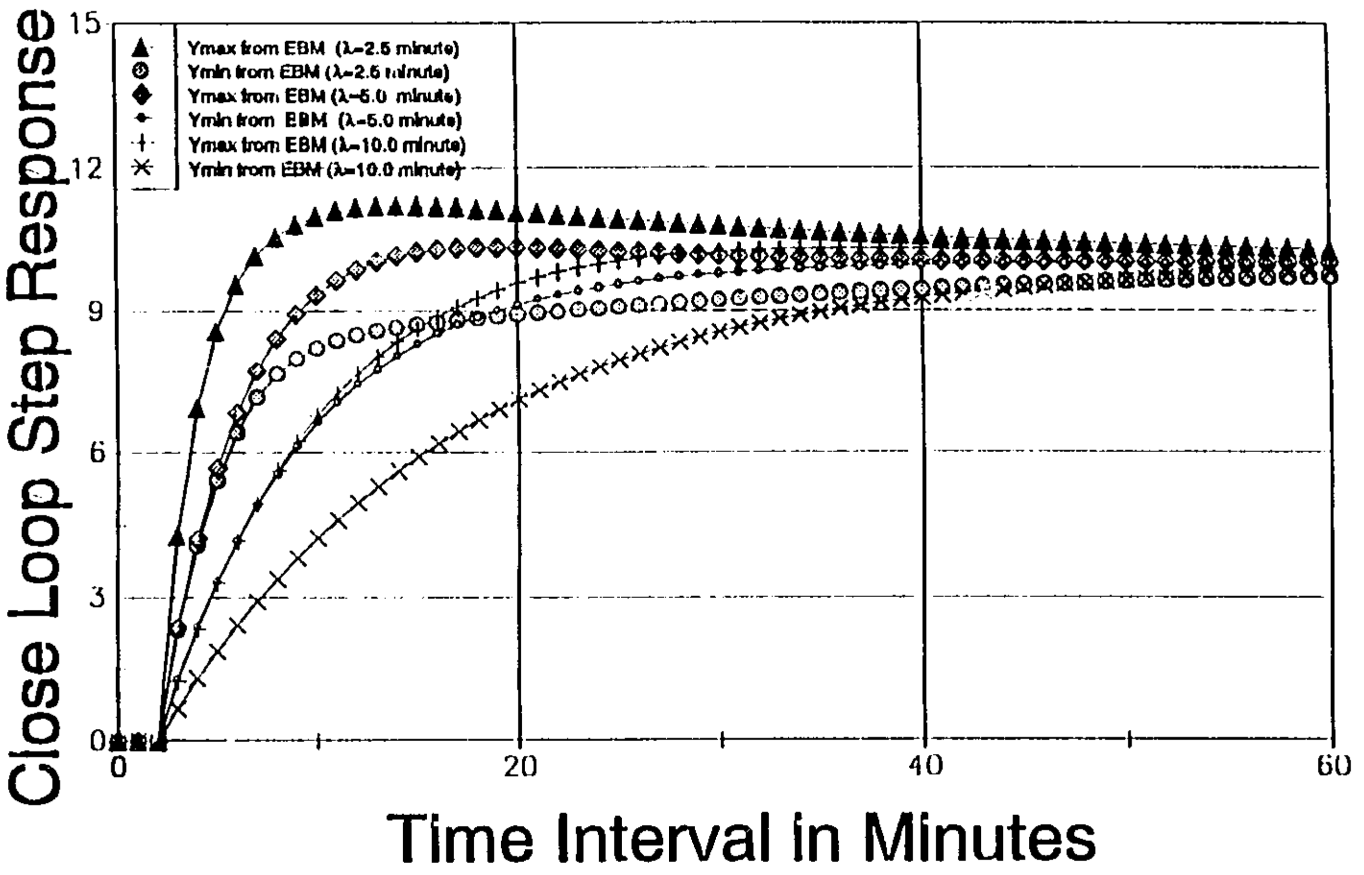


Figure 5-7: Plot of Closed Loop Servo Response Uncertainty Band Versus Dahlin Tuning Uncertainty Level 30%

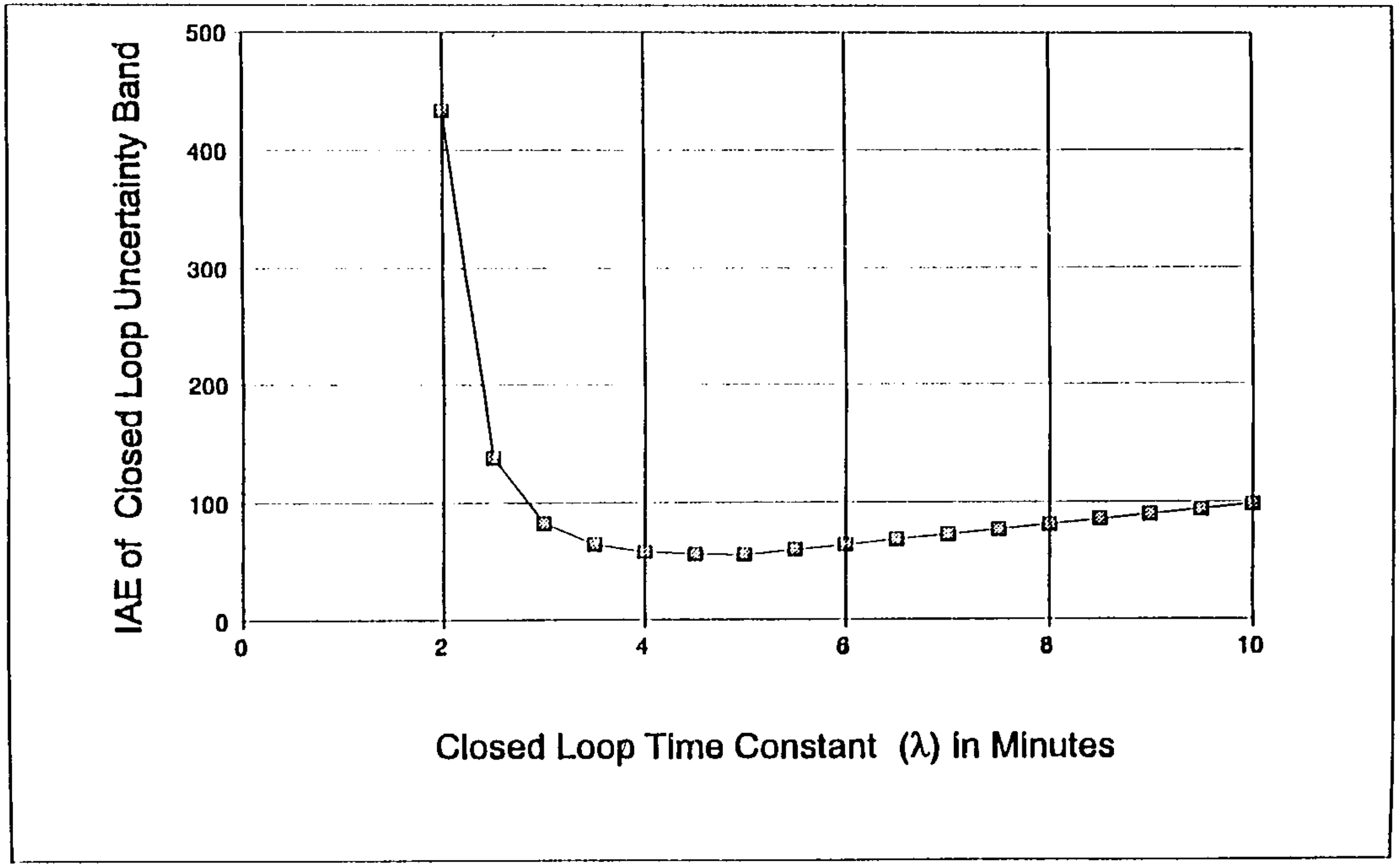


Figure 5-8: Plot of Uncertainty Band IAE versus Dahlin Tuning Uncertainty Level 30%

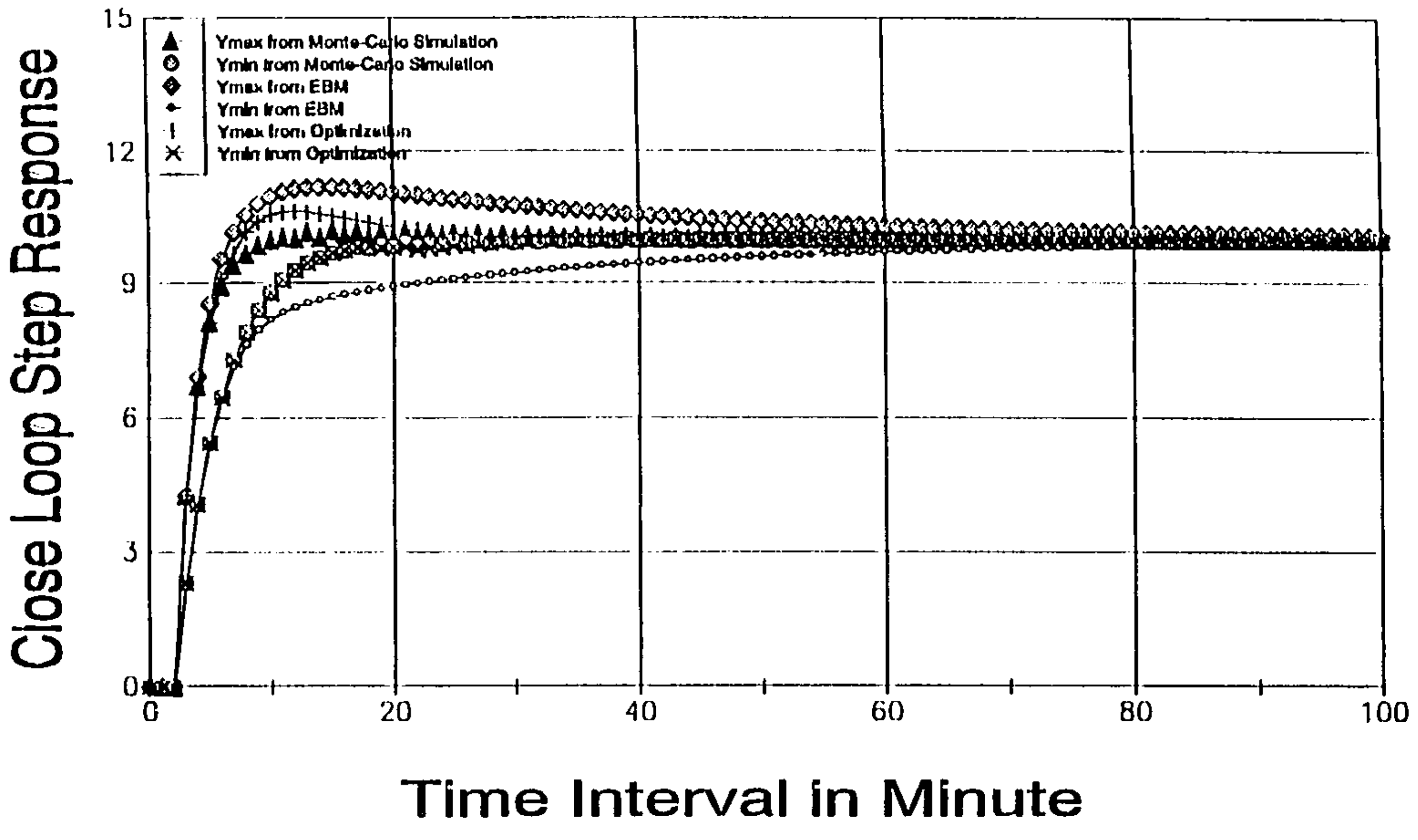


Figure 5-9: Closed Loop Servo Response Uncertainty Band ; Uncertainty Level 30% λ= 2.5 minutes

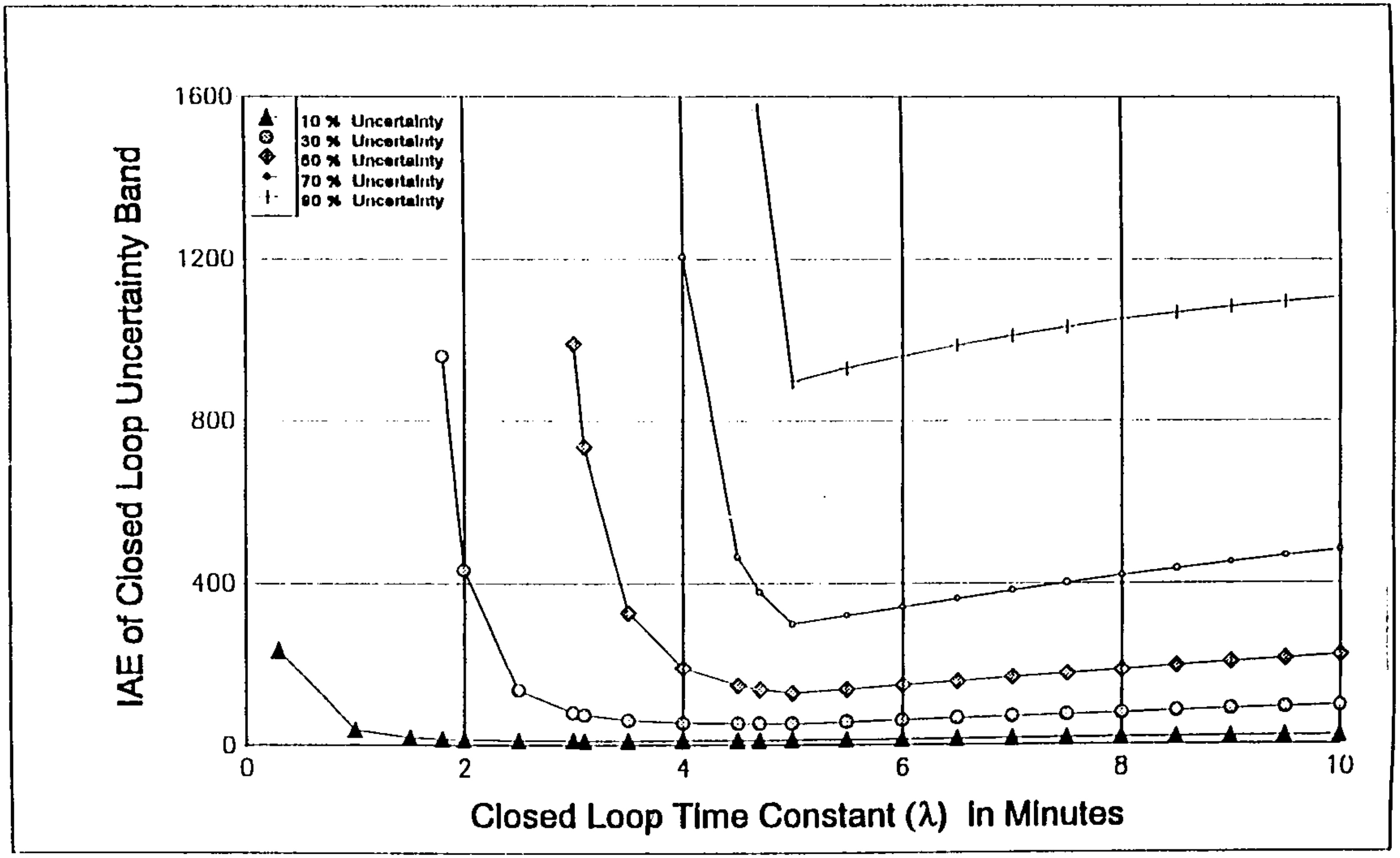


Figure 5-10a: Plot of Uncertainty Band IAE Versus Dahlin Tuning and Uncertainty Level (τd/τ = 0.2)

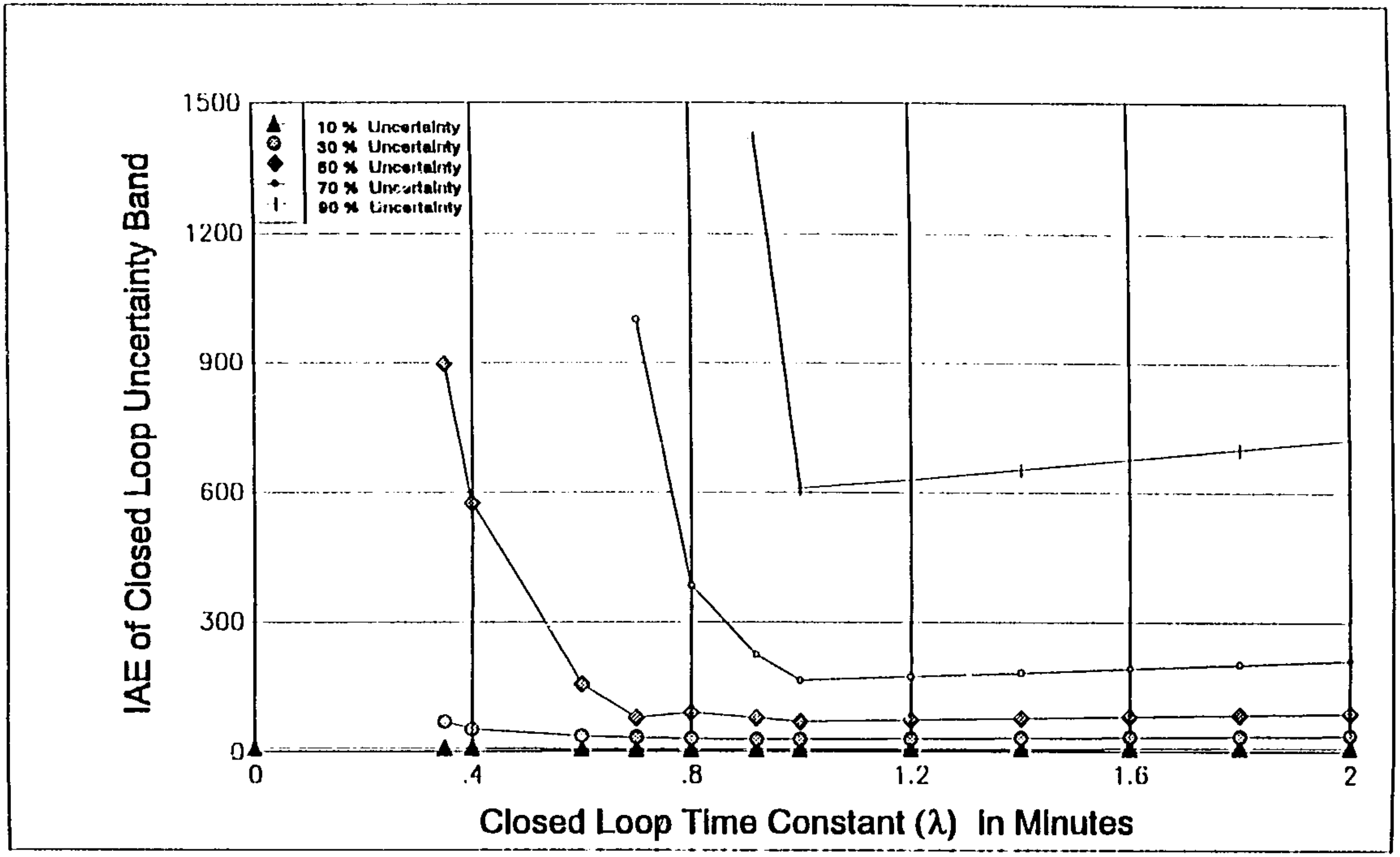


Figure 5-10b: Plot of Uncertainty Band IAE Versus Dahlin Tuning and Uncertainty Level (τd/τ = 2.0)

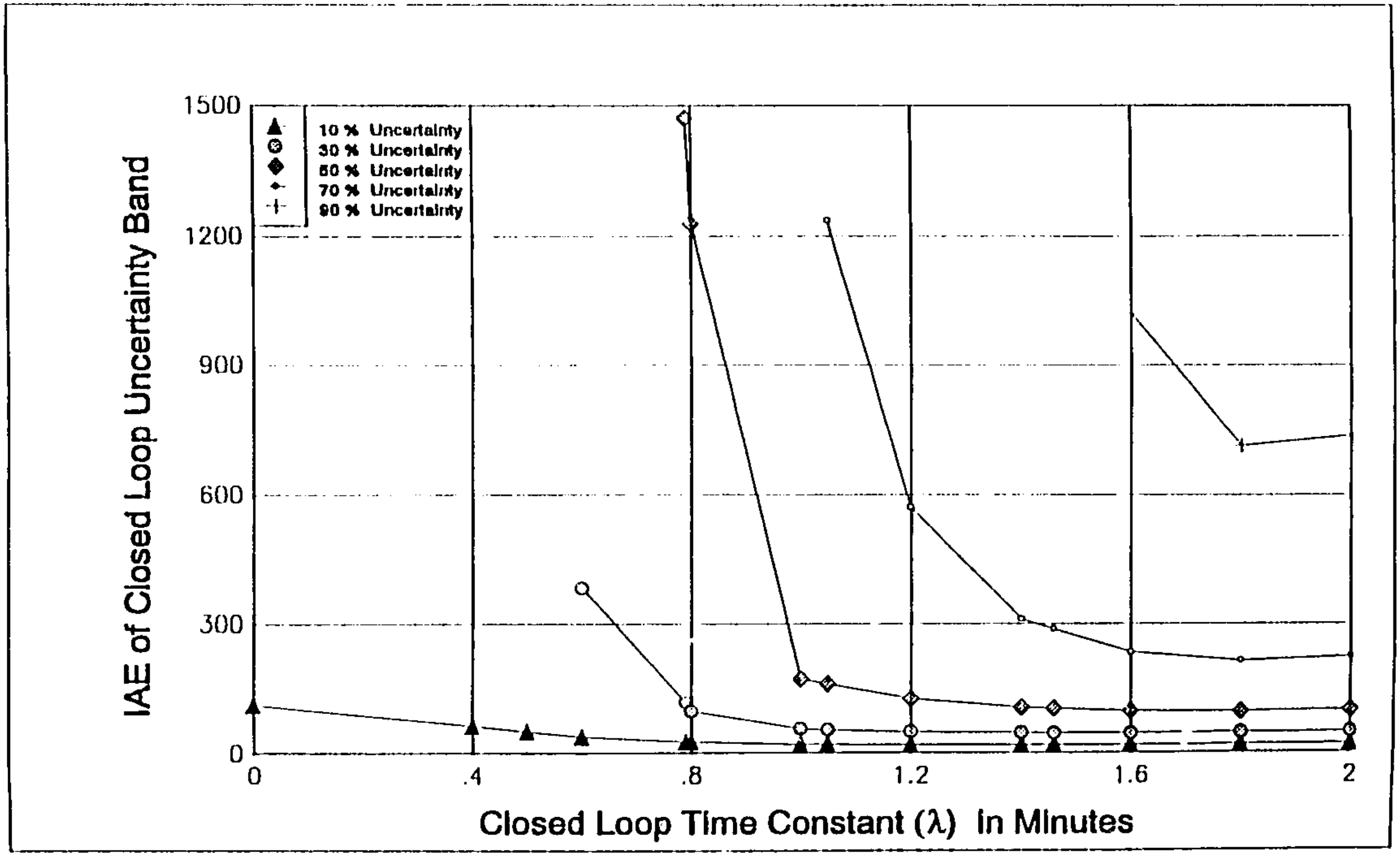


Figure 5-10c: Plot of Uncertainty Band IAE Versus Dahlin Tuning and Uncertainty Level (τd/τ =2.0 and fractional delay uncertainty)

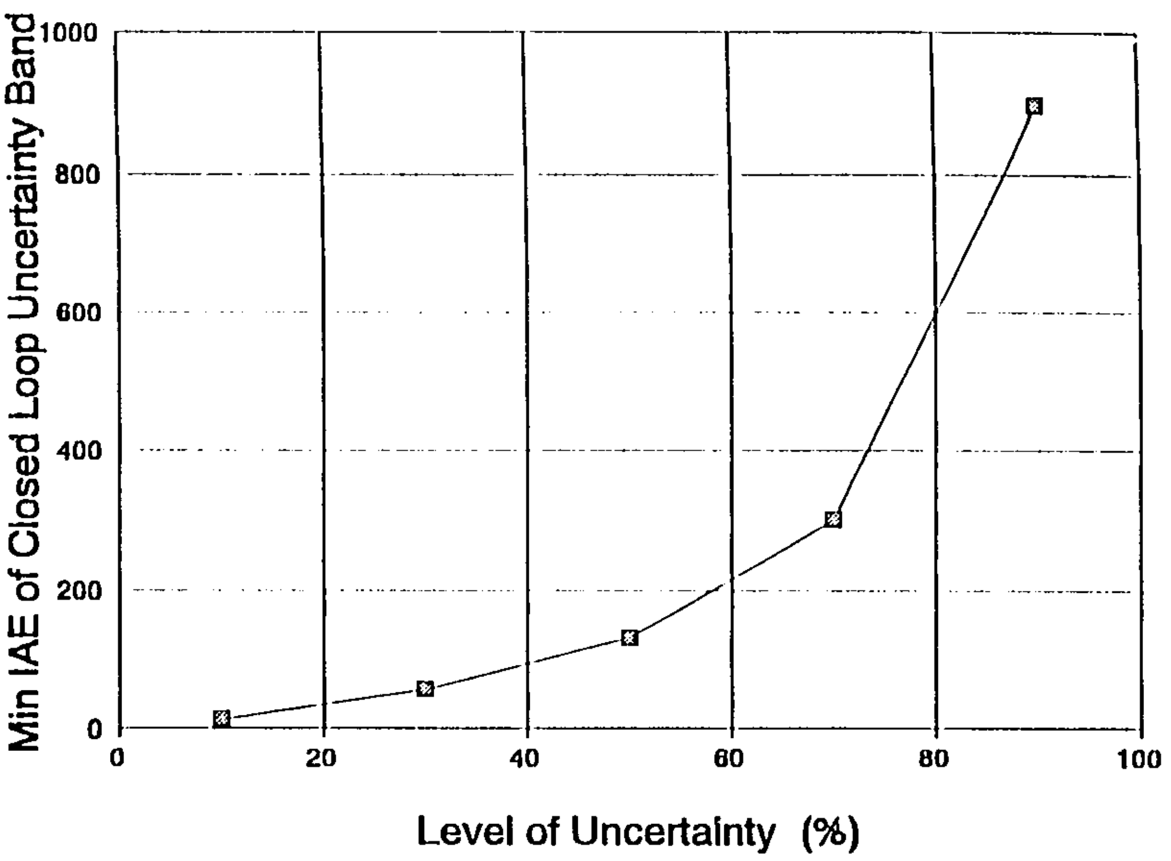


Figure 5-11: Plot of Uncertainty Band Min IAE Versus Uncertainty Level for a Dahlin Controller

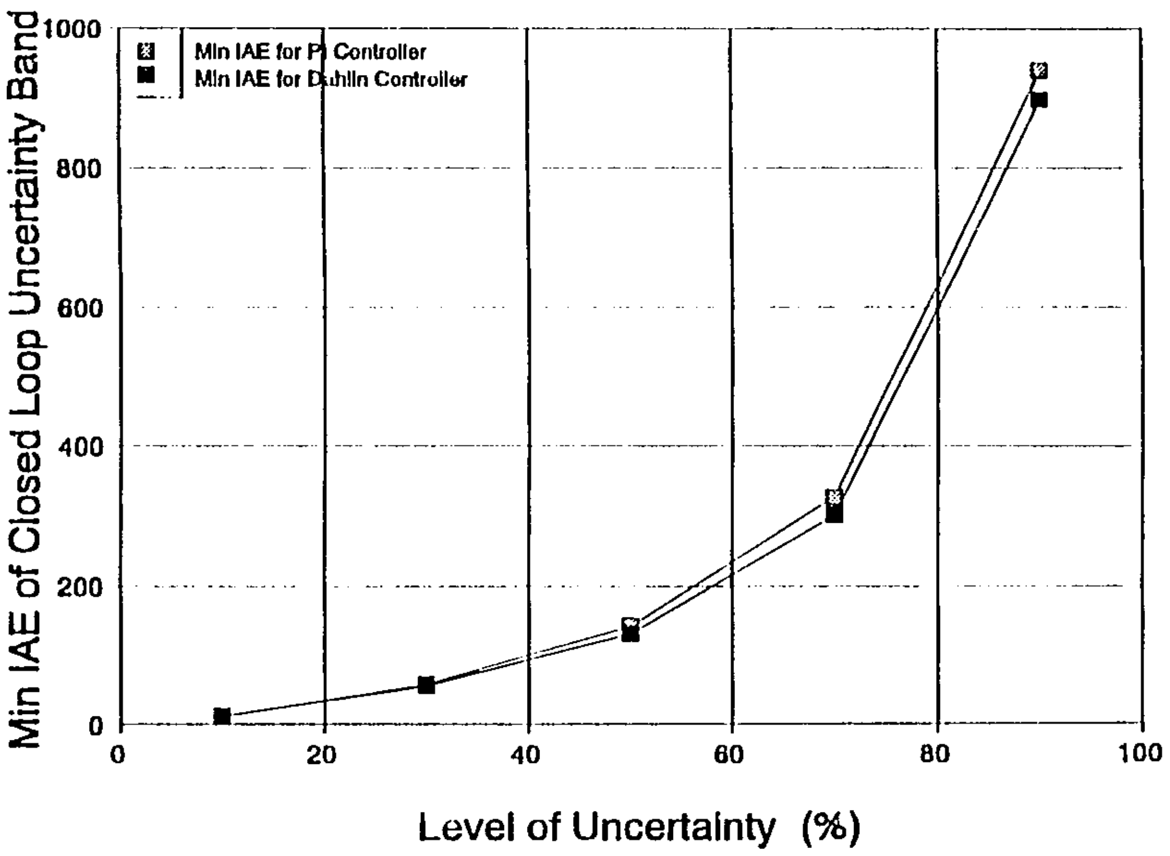


Figure 5-12: Comparison of Dahlin and PI controller Robustness (τd/τ=0.2)

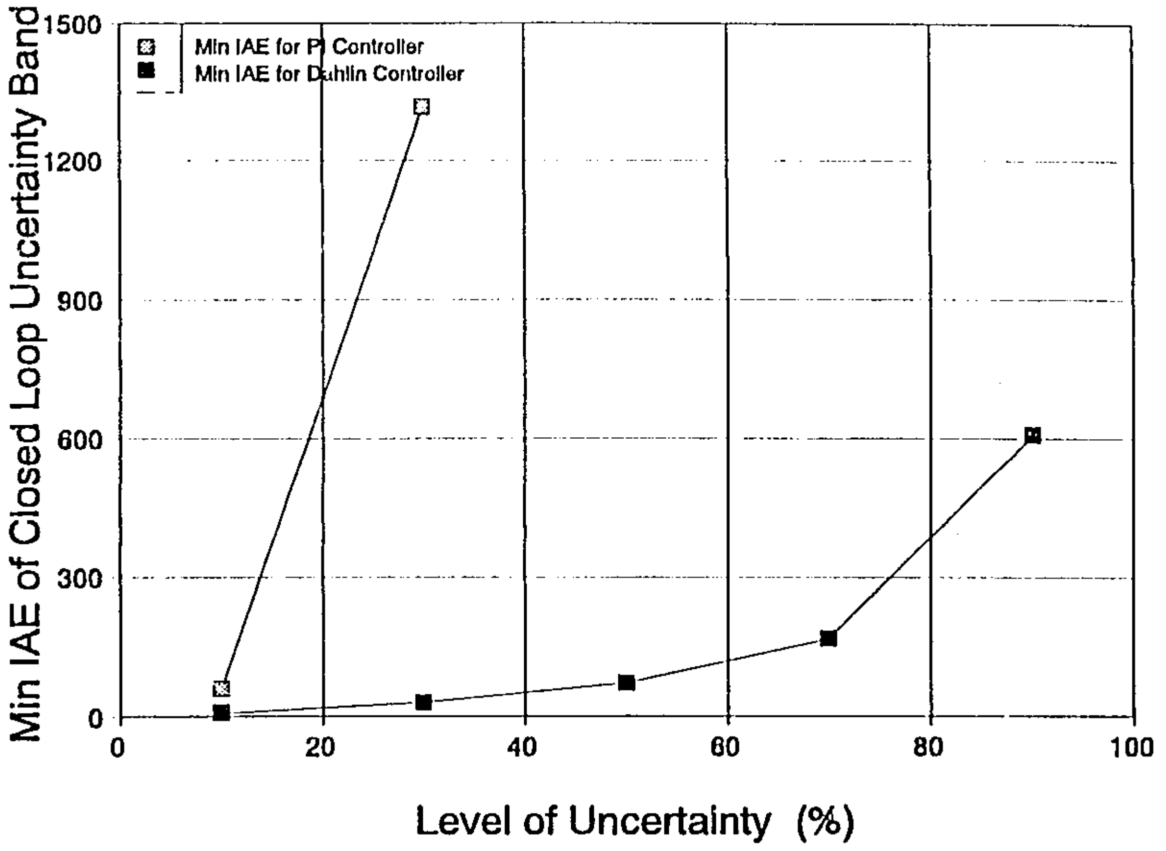


Figure 5-13: Comparison of Dahlin and PI controller Robustness ($\tau d/\tau=2.0$)

CHAPTER 6

RECOMMENDATIONS AND CONCLUSIONS

6.1 GENERAL REMARKS

Automatic control was introduced to the processing industry some forty years ago with the advent of the PID controller. After more than a decade of intense activity on computer process control, and many attempts to introduce model based controllers, the PID is still the dominant, if not exclusive, controller. Model based controllers have not been well accepted as the basic control block in the processing industry because of their sensitivity to model mismatch and other implementation issues. This thesis has addressed some of these implementation issues and the tuning aspects of model based controllers. A properly tuned and implemented model based controller has been shown to be as robust as the PID controller and capable of providing higher performance than PID. Hopefully, this research will have an impact on the wider acceptance of model based controller.

6.2 RECOMMENDATIONS

It has not been possible to explore all the research opportunities resulting from this work. The following section identifies some of the potential research areas.

6.2.1 Implementation Scheme

The moving window idea used in explaining the optimality of long horizon controllers is an interesting concept and worthy of further research. The implementation scheme developed in this work allows a rational model based controller to operate in a moving window fashion. The optimality of the current execution is not affected by previous sub-optimality. This separation is a key element in executing asynchronous control - executing a controller at unequal time intervals. St^tistic Process Control (SPC) preaches that control is required only when disturbances enter the process. This requires an efficient disturbance detection scheme and a model based control which can be executed on a demand basis. Extending the concept of the moving window to this area would provide an important bridge for the gap between SPC and automatic control groups.

6.2.2 Error Band Method (EBM)

The EBM offers a methodology to investigate control system robustness in the time domain. As noted in Chapter 5, the performance evaluation aspect of the EBM is conservative in some cases. This indicates that there are opportunities for future work in improving the EBM. Furthermore, the focus of this work has been on the development of the EBM. One extension of this work could be the exploration of the relationship between the EBM and the frequency domain approach advocated by Doyle et al. (1981). This would provide a bridge between the two different approaches.

6.2.2.1 Interval Polynomial System

The stability criteria developed for interval polynomial systems provides a simple method to study the stability of time domain uncertain control system. The natural extension in this area is to develop stability criteria for MIMO system. The Rouche Theorem, from which the SISO stability criteria are derived, is generally applicable to SISO and MIMO systems. This work laid the ground work for the development of MIMO stability criteria.

On the theoretical side, it is interesting to note the rectangular nature of interval polynomial systems and the linear nature of polytopic polynomial systems. Interval polynomial systems encompass more uncertainty than polytopic polynomial systems and hence the former stability result tends to be conservative. If the uncertainty can be defined as a polytopic polynomial, it would only be proper to develop a new procedure for the closed loop uncertainty band for polytopic polynomial systems. Therefore, one extension of this work is to provide a bounding procedure for polytopic polynomial systems.

6.2.2.2 Band Generation Technique

The conservative nature of the EBM for an amplifying controller is an issue which should not be ignored. Although there have been many attempts to tackle the band generation problem using different techniques, the EBM band is the least conservative result so far. The generation of tighter bounds for all possible closed loop responses remains an issue for future work on the EBM. Furthermore, it was observed that the bounds for the possible closed loop responses are not necessarily symmetric about the nominal responses. If the $Ydev^{max}(iT)$ and $Ydev^{min}(iT)$ can be generated separately, the conservative nature of EBM can be further improved.

6.2.2.3 Integral Delay Characterization

As indicated in Chapter 5, EBM is not adequate in handling integral delay uncertainty. EBM requires that Gerr(i)'s be assigned for all uncertainty delay intervals. If one samples fast enough, the Gerr(i)'s for delay uncertain system would be so large that EBM will not produce any useful result. Of course, one can change the sampling time so that the delay uncertainty is always fractional. However, this is not a solution because sampling time can be dictated by other factors such as disturbances. Future research can examine the uncertainty characterization in the frequency domain (Morari et al. (1986)) to see if parallels can be drawn to better characterize integral delay uncertainty in the time domain. Another alternative is to investigate if step response should be used in characterizing uncertainty instead of the impulse response. Hopefully, the improved EBM can cover a wide class of uncertain processes.

6.3 CONCLUSIONS

Two problems affecting model based control have been extensively studied in this work - implementation and robustness. For the implementation problem, four new anti-reset windup algorithms have been developed. Simulations and experiments were performed and the results indicate that these new algorithms perform much better than the conventional anti-reset windup algorithms. The research on the last algorithm, QIMC, led to the more general implementation problem in rational model based controllers. This general problem is related to saturation, operator override and temporary actuator failure. This work explains why performance is lost in the implementation of rational model based controllers. It also shows the link between rational and irrational model based controllers in terms of their performance during saturation and override. Remedies were developed which enable a rational model based controller perform as well as an irrational model based controller. These remedies work for SISO system and MIMO systems. Simulations were performed and the results show that while the solution is simple to use, the performance improvement is dramatic, especially for MIMO systems.

For the robustness problem, a time domain approach was used to study the effect of model uncertainty. Uncertainty was expressed in terms of the time domain impulse response. A theorem and corollaries were developed to predict the stability of a type of interval polynomial which is closely related to the stability of uncertain control systems when the problem is viewed in the time domain. These stability criteria were shown to be less conservative than the root location criteria. Examples were given to demonstrate the applicability of the newly developed criteria. This time domain approach was carried further to study the performance of the same uncertain closed loop control systems. The Error Band Method (EBM) was developed as a procedure to analyze this uncertain control system. Given an open loop uncertainty and a process controller, the EBM produces a closed loop step response uncertainty band. This band contains all possible closed loop responses. Therefore, the criteria is very simple: the wider the band, the less robust the controller. One can use the band as an aid in tuning a controller which must accommodate a specific open loop uncertainty. A procedure was established to design a robust model based controller and examples were given to illustrate this procedure. The method was further extended to compare the robustness of a Dahlin controller and a PI controller. Although all the results show that the EBM is a sound approach for analyzing control system robustness, the EBM is not without its limitations. From Monte-Carlo simulations, it was found that the closed loop uncertainty band generated by the EBM is tight only for non-amplifying controllers. For amplifying controllers, the EBM bound is conservative. However, this should not be a major setback for the EBM, since most robust controllers are likely to be non-amplifying. The development of EBM in this study laid the groundwork for future opportunities in perfecting the EBM. Furthermore, the EBM offers a totally different perspective to the robustness problem from the frequency domain approach favored by many researchers. Hopefully, a combined time domain and frequency domain approach can be developed in the future.

REFERENCE :

Astrom, K.J.; Wittenmark, B. "Computer Control Systems: Theory and Design", Prentice-Hall, Information and System Science Series, 1984.

Argon M.B., "On Sufficient Condition for the stability of Interval Matrices", Int. J. Control, 1986, 44,1245-1250

Bartlett, A.C. "Vertex Results for the Steady-State Analysis of Uncertain System", Proc. Decision and Control, Dec. 1990

Bartlett, A.C.; Hollot, C.V.; Huang, L. "Root Locations of an Entire Polytope of Polynomials: it suffices to check the edges", Proceeding of American Control Conference, Minneapolis, 1987, 1611-1616.

Barmish, B.R., "Invariance of the Strict Hurwitz Property for Polynomials with Perturbed Coefficients", IEEE Tran. Auto. Control, 1984, AC-29 935-936

Barmish, B.R.; De Marco, C.L "Criteria for Robust Stability of Systems with Structured Uncertainty: A Perspective", Proc of American Control Conference, 1987, 476-481

Barmish B.R.; Leitmann, G."On Ultimate Boundedness Control of Uncertain Systems in the Absence of Matching Assumption", IEEE Tran. Auto. Control, 1982, AC-27, 153-158

Bequette, B.W.; Horton, R.R.; Edgar, T.F. "Resilient and Robust Control of an Energy Integrated Distillation Column", ACC. Proc. 1987, 1027-1033.

Berg, S.; Edgar, T., "Stability of DIgital Control Algorithm subject to Model Errors", JACC Proceedings, San Francisco, 1980.

Berger, C.S., "A simple sufficient condition for the stability of linear discrete time systems", Int. J. Control, 1982, 35, 1073-1080

Bialas, S., "A necessary and sufficient condition for the stability of interval matrices", Int. J. Control, 1983, 37, 717-722

Bose, N.K.; Shi, Y.Q. "A Simple General Proof of Kharitonov's Generalized Stability Criterion", IEEE Trans. Circuits and Systems", 1987, Vol-34, 8, 1233-1237

Bose, N.K.; Zeheb, E. "Kharitonov's theorem and stability test of multidimensional digital filters" IEE Proc. Pt. G., 1986, 133, 187-190

Box, G.; G. Jenkins "Time Series Analysis, Forecasting and Control", Revised Edition, Holden Day, New Jersey, 1970

Campo, P.; Morari M. "Robust Control of Processes subject to Saturation non-linearities", Computer & Chemical Eng., 1990, # 4/5 Vol. 14 pp 343-358.

Clarke, D.; Hasting-James,R. "Design of Digital Controllers for Randomly Disturbed Systems", Proc. Inst. Elect. Eng.,1971, 118, 1503-1506.506.

Cutler, C.; Ramaker, B. "Dynamic Matrix Control - A Computer Control Algorithm", Proc. JACC, San Francisco, California, 1980.

Cruz, J.B.; Freudenburg,J.S.; Looze, D.P. "A Relationship Between Sensitivity and Stability of Multivariable Feedback System", IEEE Trans. Auto. Control, 1981, AC-26, 66-74

Dahlin E.B. "Designing and Tuning Digital Controllers", Instruments Control Systems, Vol. 41, pp. 77-83, June 1968.

Dabke, K.P., "A simple criterion for stability of linear discrete system", Int. J. Control, 1983, 37, 657-659

Djaferis, T.F. "Simple robust stability tests for polynomials with real parameter uncertainty", Int. J. Control, 1991, 53, 907-927

Doyle, J.C. "Analysis of feedback systems with structured uncertainties", IEEE Proc. 1982, Vol. 129, Pt. D., 242-250.

Doyle, J.C.; Smith, R.S.; Enns, D.F. "Control of Plants with Input Saturation Nonlinearities", ACC. Proc. 1987, 1024-1039.

Doyle J.C.; Stein, G.; "Multivariable Feedback Design: Concept for a Classical/Modern Synthesis" IEEE Trans. Auto. Control, 1981, AC-26, 4-16

Dumont, G. "Analysis of the Design and Sensitivity of the Dahlin Regulator, PPRIC Internal Report, 1981.

Gallun, S.E.; Matthews, C.W.; Senyard, C.P.; Slater, B. "Windup protection and initialization for advanced digital control", Hydrocarbon Processing, 1985, June, 63-68

Garcia, C.; Morari, M. "Internal Model Control 1: A UnifyingReview and some new Results", Ind. Eng. Chem. ProcessDes. and Dev., 1981, 21, 308-323.

Garcia, C.E. ; Morshedi, A.M. "Solution of the Dynamic Matrix Control Problem via Quadratic Programming(QDMC)", Proceeding of the Canadian Industrial Computing Society, Ottawa, Ont., 1984.

Glattfelder, A.H.; Schaufelberger, W. "Stability Analysis of Single Loop Control Systems with Saturation and Antireset-Windup Circuits", IEEE Trans. Auto. Control, 1983, AC-23, 1074-1081

Hanus, R.; Kinnaert, M.; Henrotte, J.L. "Conditioning Technique, A General Anti-Windup and Bumpless Transfer Mether", Automatica, 1987, 23, 729-739

Harris, T.J.; MacGregor, J.F.; Wright J.D. "An Overview of Discrete Stochastic Controllers: Generalized PID algorithm with Dead-time Compensation", Can. J. Chem. Eng., Vol. 60, pp. 425-432, 1982.

Harris, T; MacGregor, J. "Design of Discrete Multivariate Linear Quadratic Controller using Transfer Function", To appear in AIChE, 1987.

Heinen, J., "A simple and direct proof of a bound on the zeros of a polynomial", Int. J. Control, 1985, 42, 269-270

Hollot C.V; Barlette A.C. "Some discrete-time counterparts of the Kharitonov stability criteria for uncertain systems", IEEE Tran. Auto. Control, 1986, AC-31, 355-356.

Horowitz I. "Quantitative feedback Theory" IEE Proc. 29, Pt. D, 1982, 215-226

Horowitz I.; Sidi M. "Synthesis of feedback systems with large plant ignorance for prescribed time domain tolerances", Int. J. Control, 1972, 30, 287-309

Huang, L.; Hollot, C.V.; Barlett,A.C., "Stability of families of polynomials: geometric considerations in coefficients space", Int. J. Control, 1987, 45, 649-660

Jiang, C.I. "Sufficient and Necessary Condition for the asymptotic stability of discrete linear interval systems", Int. J. Control, 1888, 47, 1563-1565

Juang, Y.T.; Shao C.S. "Stability Analysis of dynamic interval system", Int. J. Control, 1989, 49, 1401-1408

Jury, E., "Theory and Application of the z-Transform Method", John Wiley Sons, Inc., New York, 1964.

Karl, W.C.; Greschak J.P.; Vesghese G.C. "Comments on 'A necessary and sufficient condition for the stability of interval matrices", Int. J. Control, 1984, 39, 849-851

Katbab, A.; Jury, E.I. "Robust Schur-stability of control systems with interval plants", Int. J. Control, 1990, 51, 1343-1352

Katbab, A.; Jury, E.I. "Generalization and comparision of two recent frequency-domain stability robustness results with interval plants", Int. J. Control, 1991, 53, 463-475

Kestenbaum, A.; Shinnar, R.; Thau, F.E. "Design Concepts for Process Control", Ind. Eng. Chem., Process Des. and Dev., Vol. 15, No. 1, pp. 1-13, 1976.

Khambanonda T.; Palazoglu A. "Robustness studies for multivariate feedback systems using refined eigenvalue inclusion regions

(REIR),Computer & Chem Eng. 1990, Vol 14, No. 4/5, pp 391-400.

Khandria, J.; Luyben, W.L. "Experimental Evaluation of Digital Algorithms for Anti-reset Windup ", Ind. Eng. Chem., Process Des. and Dev., Vol. 15, No.2, pp. 278-284, 1976.

Kharitonov, V.L., "Asymptotic Stability of an equilibrium position of a family of systems of linear differential equations", Differential'n Uranvneiya, 14, 1978, 2086-2088.

Kozub,D., "Advanced Model Based Multivariate Control Design and Application to Packed Bed Reactor Control" M.Eng. Thesis, McMaster University, Hamilton, Ontario, Canada, 1986.

Kozub D.; MacGregor, J. "Feedback Control of Polymer Quality in Semi-Batch Copolymerication Reactor" submitted to Chemical Engineering Science 1990.

Krishnan K.R.; Cruickshanks A. "Frequency-domain design of feedbacks systems for specified insensitivity of time-domain response to parameter variation", Int. J. Control, 1977, 25, 609-620

Kraus, F.; Mansour, M.; Jury, E.I. "Robust Schur-Stability of Intervals Polynomials", Proc. Decision and Control, Dec. 1990.

Kuest, J.L." Optimization Techniques with Fortran", New York: McGraw-Hill, 1973.

Kumar S. "Modelling and Advanced Multivariable Control Strategies for a Pilot Scale Extractive Distillation Column", Ph.D Thesis, McMaster University, Hamilton, Ontario, Canada, 1987.

Kusuma, I ; Elliott, R. "Multivariate Control of a Deisohexanizer Distillation Column", Canadian Industrial Computer Society Conference Proceeding, Ottawa, 1984

Langille, K. M.Eng. Thesis, McMaster University, Hamilton, Ontario, Canada, 1983.

Latosinsky, R. "Analysis, Configuration and Control of an Extractive Distillation Column", M.Eng. Thesis, McMaster

University, Hamilton, Ontario, Canada, 1988.

Laughlin, D. ; Jordan, K.; Morari, M. "Internal model control and process uncertainty: ,apping uncertainty regions for SISO controller design", Int. J. Control, 1986, 44, 1675-1698

Lewin, D.R. "Robustness performance specifications for uncertain stable SISO systems", Int. J. Control, 1991, 53, 1263-1281

Lin, S.H.; Fong, I.K.; Juang, Y.T., Kuo, T.S., Hsu, C.F. "Stability of perturbed polynomials based on the argument principle and Nyquist criterion", Int. J. Control, 1989, 50, 55-63

Lunze, J. "The Design of Robust Feedback Controllers for Partly Unknown Systems by Optimal Control Procedures", Int. J. Control, 1982, 36, 611-630

Lunze, J. "The Design of Robust Feedback Controllers in the Time Domain", Int. J. Control, 1984, 39, 1243-1260.

Luyben, W.L. "Process Modeling, Simulation, and Control for Chemical Engineers", McGraw-Hill Book Company, N.J., 1973

MacGregor, J.; Taylor, P. ; Wright, J. "Advanced Process Control", McMaster University, Hamilton, Canada, 1986.

Marshall,J. "Control of time-delay systems", IEE Control Engineering Series 10, Peter Peregrinus Ltd., 1979.

Martin, G.D. "Long-Range Predictive Control", A.I.Ch.E. Journal, Vol. 27, pp. 748-805, (1981).

Mayer,S., M.Eng. Thesis, McMaster University, Hamilton, Ontario, Canada, 1986.

McDonald K.A.; McAvoy T.J "Application of Dynamic Matrix Control to Moderate and High-Purity Distillation Towers" Ind. Eng. Chem. Res. 1987, 26, 1011-1018

Morari M.; Doyle J.C."A Unifying Framework for Control System Design Under Uncertainty and its Implications for Chemical Process Control", Third International Conference on

Chemical Process Control, Asilomar, CA, 1986

Morari M.; Zafiriou E. "Robust Process Control", Prentice-Hall, Englewood Cliffs, N.J. (1989).

Mori, T., "Note on the Absolute Value of the Roots of a Polynomial", IEEE Tran. Auto. Control, 1984, AC-29, 54-56

Mori, T.; Kokame H. "Convergence property of interval matrices and interval polynomials", Int J. Control, 45, 481-484 , 1987

Mosler, H.; Koppel, L. ; Coughanowr, D. "Process Control by Digital Compensation", A.I.Ch.E. Journal, Vol. 13, No. 4, pp. 768-778, 1967.

Neff, H., Continuous and Discrete Linear System, Harper & Row, Publisher New York, 1984.

Palmor, Z. ; Shinner, R. "Design of Sampled Data Controller", Ind. Eng. Chem., Process Des. and Dev., Vol. 18, No. 1, pp. 8-30, 1979.

Palazoglu, A.; Arkun, Y. "A Multivariate Approach to design chemical plants with robust ynamic operability characteristics", Computer & Chemical Eng., 1986, Vol 10, pp 567-575.

Postlethwaithe, I. Edmunds, J.M.; MacFarlene, A.G. "Principle Gain and Principle Phases in the Analysis of Liner Multivariable Feedback System", IEEE Trans. Auto. Control, 1981, AC-26, 32-46

Richalet, J.; Rault, A.; Testud, J.; Papon, J. "Model Predictive Heuristic Control: application to industrial process", Automatica, 1978, 14, 413-428.

Rosenbrock, H.; STOREY, C., Mathematics of Dynamical Systems, London: Nelson, 1970, 291

Rosenbrock, H.H. "An Automatic Method for finding the Greatest and Least Value of a Function", Computer J., Vol. 3, pp. 175-184, 1960.

Rouhani, R.; Mehra, R. "MAC; Basic Theoretical Properties", Automatica, 1982, 28, 401-414.

Safonov, M.G.; Laub, A.J.; Hartman, G.L. "Feedback Properties of Multivariable System : the Role and the use of the Return Difference Matrix", IEEE Trans. Auto. Control, 1981, AC-26, 47-65

Segall, N.L. "Incorporation of Radial Gradients in Packed Bed Reactor Models", M. Eng. Thesis, McMaster University, 1983.

Segall, N.L.; MacGregor, J.F. ;Wright, J.D., "One-step Optimal Correction for Input Saturation in Discrete Model Based Controllers", Internal Report - McMaster University (1985).

Segall, N.L. ;Taylor, P.A., "Saturation of Single Input Single Output controller written in velocity form: reset windup protection." Ind. Eng. Chem., Process Des. and Dev., in press, 1986.

Segall, N.; MacGregor, J.; Wright, J. "One-Step Optimal Correction for Input Saturation", Proc. CsChE Conference, Sarnia, Ontario, 1984.

Shi Y.Q. "Comments of 'Stability of Entire Polytope cf Polynomials", Int. J. Control, 1990, 51, 495-496

Skogestad S.; Lundstrom P. "μ Optimal LV- control distillation Column", Computer & Chem Eng. 1990, Vol 14, No. 4/5, pp 401-413

Smith, C.L., "Digital Computer Process Control", In Text Education Publishers, 1972

Soh, C.B.; Berger, C.S.; Dabke, K.P., "On the Stability Properties of Polynomials with Perturbed coefficients", IEEE Tran. Auto. Control, 30, 1985, 1033-1036; IEEE Tran. Auto. Control, AC-32, 1986, 239-230

Soh, C.B. "Necessary and sufficient conditions for stability of symmetric interval matrices", Int. J. Control, 1990, 51, 243-248, Soh, C.B. "Correcting Argon's approach for the stability of interval matrices", Int. J. Control, 1990, 51, 1151-1154

Soh, Y.C. "Stability of Entire Polytope of Polynomials", Int. J. Control, 1989, 49, 993-999

Tempo, R. "A Simple Test for Schur Stability of a Diamond of Comples Polynomials", Proc. Decision and Control, Dec. 1989.

Wilson, G.T. "Modelling Linear System for Multivariate Control", Ph. D. Thesis, University of Lancaster, Lancaster, England, 1970.

Wong, P.M. "Multivariate Time Series Modeling and Stochastic Optimal Control of a Packed-Bed Reactor", M.Eng. Thesis, McMaster University, 1983.

Wong, P.M.; Taylor, P; Wright, J. "An Experiment Evaluation of Saturation Algorithm for Advanced Digital Controllers", Ind. Eng. Chem. Process Des. and Dev., 1987a, 26, 1117-1127.

Wong, P.M.; Taylor, P.A. ; Wright, J.D. "Investigation of SISO Digital Controller Robustness in Time Domain", CSChE Conference Proc. Montreal, 1987b.

Wong, P.M.; Taylor, P.A. ; Wright, J.D. "On the Optimal Implementation of Rational Model Based Digital Controllers to SISO systems", Proceeding of Chemical Engineering Graduate Student Conference, Kingston, Ont., 1987c.

Wong, P.M.; Taylor, P.A. ; Wright, J.D. "On the Stability of Discrete Interval Polynomials", Submitted for publication, 1990.

Wright, J.D. "Advanced Process Control - Lab Manual",MacMaster University, 1988.

Xi, Y.; Schmidt, G., "A note on the Location of the Roots of a Polynomial IEEE Tran. Control, 1985, AC-30, 78-80

Yeh, H.H; Banda S.S.; Ridgely, D.B. "Stability Robustness measures utilizing structural information", Int. J. Control, 1985, 41, 365-387.

Yedavalli, R.K., "Stability analysis of interval matrics: another sufficient condition", Int. J. Control, 1986, 43, 767-772

Yeung, K.S., "Linear sytem stability under parameter uncertainties", Int. J. Control, 1983, 38, 459-464

Zames, G. "On the Input-Output Stability of time varying non-linear Feedback System", IEEE Trans. Auto Control, 1966, AC-11, 228-238

Zafiriou E. "Robust Model Predictive Control of Processes with Hard Constaints", Computer & Chem Eng. 1990, Vol 14, No. 4/5, pp 359-372.

APPENDIX A: SAMPLE FORTRAN CODING

APPENDIX A.1: FORTRAN CODING FOR ANTI-RESET WINDUP ALGORITHM

(VELOCITY FORM - ADAPTED FROM SEGALL AND TAYLOR (1986))

```
C
C       Following equation 5, E(i) is the current error
C       We need U(i), the new control action
C       Define Error vector E and control action vector U
C       E vector:  E(i), E(i+1), ..., E(i+4)
C       U vector:  U(i), U(i+1), ..., U(i+NB+1)
C       U(1) is thus the actual control action applied to the
C       process
C
C
C       CALCULATE THE CONTRIBUTION FROM DEAD-TIME COMPENSATION
                DEAD = 0.0
                   DO I=1, NB
                        DEAD = B(I)*(U(I+1) - U(I+2)) + DEAD
                ENDDO
C
C       CALCULATE THE PROPORTIONAL AND DERIVATIVE  CONTRIBUTIONS
C       ASSUMING NA=4
C
                        SUM5 = K(5)*(E(1) - 4.0*E(2) +  6.0*E(3) -
     +                         4.0*E(4) + E(5))
                        SUM4 = K(4)*(E(1) - 3.0*E(2) +  3.0*E(3) -
     +                         E(4))
                        SUM3 = K(3)*(E(1) - 2.0*E(2) +  E(3))
                        SUM1 = K(1) * (E(1) - E(2) )
C
C       ADD THE PROPORTIONAL, DERIVATIVES AND DEAD-TIME
C       COMPENSATION
C       CONTRIBUTION TO THE PAST CONTROL ACTION
        UPD  = UPAST + SUM5 + SUM4 + SUM3 + SUM1 + DEAD
C
C       CALCULATE INTEGRAL CONTRIBUTION
                DELUI = K(2) * E(1)
C
C       SUM TO COMPUTE THE TOTAL CONTROL ACTION
                UPID  = UPD + DELUI
C
C       TEST FOR OUTPUT LIMITS
C       CLAMP UPID ONLY WHEN SATURATION CAUSED BY INTEGRATION
C
                IF (((UPD.GE.UMAX).AND.(UPID.GE.UMAX)) THEN
                        U(1) = UMAX
                ELSEIF (((UPD.LE.UMIN).AND.(UPID.LE.UMIN)) THEN
                         U(1) = UMIN
                ELSEIF (((UPD.LT.UMAX).AND.(UPID.GE.UMAX)) THEN
                         UPID = UMAX
                         U(1) = UMAX
                ELSEIF (((UPD.GT.UMIN).AND.(UPID.LT.UMIN)) THEN
                         UPID = UMIN
                         U(1) = UMIN
                ELSE
                        U(1) = UPID
                ENDIF
C
C       UPDATE THE UPAST
        UPAST = UPID
C
C       Now call a Subroutine to Shuffle backwards the E and U
C       vectors ready for the next control interval
```

APPENDIX A.2: FORTRAN CODING FOR HEURISTC ALGORITHM

```
C       U(i) FROM ANY ADVANCED DIGITAL ALGORITHM
C       THE EXCESS CONTROL ACTION FROM
C       THE PREVIOUS INTERVAL IS COMP
C
C       IF CONTROL ACTION ALREADY SATURATED; DON'T ADD THE
C       COMPENSATION
        IF (U.LT.HILIM .AND. U.GT.LOLIM) U = U + COMP
C
C       SET THE COMPENSATION IF CONTROL ACTION AT UPPER LIMIT
                IF (U.GT.HILIM) THEN
                        COMP = U - HILIM
                        U = U - COMP
C
C       SET THE COMPENSATION IF CONTROL ACTION AT LOWER LIMIT
                ELSEIF (U.LT.LOLIM) THEN
                        COMP = U - LOLIM
                        U = U - COMP
C
C       SET THE COMPENSATION IF CONTROL ACTION NOT SATURATED
                  ELSE
                        COMP = 0.0
                  ENDIF
```

Appendix B : Extractive Distillation Column Models

Appendix B-1 Process/Disturbance Models

For details on the derivation of the model forecast, interested readers are referred to *Box and Jenkins (1970)*.

Process Model :

$$Y(i) = \frac{\omega(z)z^{-b}}{\delta(z)} U(i) + \frac{\theta(z)}{\phi(z)\nabla} a(i) \qquad (B-1)$$

b-Step ahead Process Forecast :

$$Y(i+b) = \frac{\omega(z)}{\delta(z)} U(i) + N(i+b) \qquad (B-2)$$

Disturbance Model:

$$N(i) = \frac{\theta(z)}{\phi(z)\nabla} a(i) \qquad (B-3)$$

b-Step ahead Noise Model Forecast :

$$N(i+b) = \frac{T(z)}{\theta(z)} N(i) \qquad (B-4)$$

Where T(z) can be obtained from the following equation:

$$\frac{\theta(z)}{\Phi(z)\nabla} = \psi_f(z) + \frac{T(z)\ z^{-b}}{\phi(z)\nabla} \qquad (B-5)$$

and the feedback enters through the estimates N(i), which is related to the measurement, $Y_m(i)$ *:*

$$N(i) \;\; = Y_m(i) - \frac{\omega(z)z^{-b}}{\delta(z)} U(i) \tag{B-6}$$

Setpoint Trajectory Model:

$$Y_{sp}(i) = \frac{K_n(z)z^{-b}}{K_d(z)} R(i) \tag{B-7}$$

b-Step ahead Setpoint Trajectory Forecast :

$$Y_{sp}(i+b) = \frac{K_n(z)}{K_d(z)} R(i) \tag{B-8}$$

Appendix B-2 Extractive Distillation Column (EDC)

This is a pilot scale unit in the Chemical Engineering Department at McMaster University. The process performs the separation of the azeotropic mixture of acetone and methanol with the addition of water as the solvent. A sequence of research works can be found in *Langille (1983), Mayer (1986), Latosinsky (1988) and Kumar (1987).* Step tests were performed *(Latosinsky, 1988).* A continuous first order plus deadtime model was found to be an adequate linear fit of the process data. The continuous model was then discretized with a sampling time of 28 minutes to give the following discrete process model:

$$\begin{bmatrix} \text{Acetone Conc.} \\ \text{Methanol Conc.} \end{bmatrix} = 1.0E\text{-}3 \begin{bmatrix} Gp_{11} & Gp_{12} \\ Gp_{21} & Gp_{22} \end{bmatrix} \begin{bmatrix} \text{Solvent Flow} \\ \text{Steam Flow} \end{bmatrix} \qquad (B\text{-}9)$$

Concentrations deviate in mole percent

Flow rate deviate in ml/min

$$Gp_{11} = \frac{0.9059z^{-2} + 0.2635z^{-3}}{1 - 0.627\ z^{-1}} \qquad (B\text{-}10a)$$

$$Gp_{12} = \frac{-0.8544z^{-1} - 0.4834z^{-2}}{1 - 0.5712z^{-1}} \qquad (B\text{-}10b)$$

$$Gp_{21} = \frac{-0.6487z^{-1} - 0.8286z^{-2} - 0.03139z^{-3}}{1 + 0.4474z^{-1} + 0.4082z^{-2}} \qquad (B\text{-}10c)$$

$$Gp_{22} = \frac{2.533\ z^{-1} + 2.768\ z^{-2}}{1 - 0.6703z^{-1}} \qquad (B\text{-}10d)$$

The outputs of the process model are the acetone and methanol concentration deviate (mole%) and the inputs are the solvent and steam flow rate deviate (ml/min). The upper and lower limits for the solvent and steam flow rate deviates are ±13.5 ml/min and ± 20 ml/min respectively. The control interval was chosen to be 28 minutes in this work. One of the process disturbances comes from the feed flow rate. The disturbance is deliberately modeled as a stochastic disturbance rather than a deterministic disturbance for the sake of stochastic controller design. An Integrated Auto-Regressive (IAR) noise model is an adequate description of the disturbance.

$$N(i) = \frac{1}{(1. - 0.488z^{-1})\nabla}\ a(i) \qquad (B\text{-}11)$$

Appendix B-3 Equivalence of Optimal Solution

The following section shows that the optimal controller solution in equation 3-9 can be degenerated to a Dahlin or a MVC controller.

Dahlin Controller :

The difference equation for a Dahlin controller is (Smith (1972)) :

$$\frac{U(z)}{E(z)} = \frac{\delta(z)}{\omega(z)} \cdot \frac{K_n(z)}{K_d(z) - K_n(z)z^{-b}} \qquad (B\text{-}12)$$

The general optimal solution of a model based controller is shown in equation 3-9. For a Dahlin controller, there is no penalty on the control moves ($\beta = 0.0$), hence the equation becomes :

$$\frac{K_n(z)}{K_d(z)} R(i) - \frac{\omega(z)}{\delta(z)} U(i) - \frac{T(z)}{\theta(z)} N(i) = 0.0 \qquad (B\text{-}13)$$

Since Dahlin controller is based on a servo design, the optimal filter, $\frac{T(z)}{\theta(z)}$, which is equivalent to a servo design, is $\frac{K_n(z)}{K_d(z)}$ (see Harris et al. (1982)). Substituting the N(i) relationship in B-6, equation B-13 becomes :

$$\frac{K_n(z)}{K_d(z)} R(i) - \frac{\omega(z)}{\delta(z)} U(i) - \frac{K_n(z)}{K_d(z)} \left[Y_m(i) - \frac{\omega(z)z^{-b}}{\delta(z)} U(i)\right] = 0 \qquad (B\text{-}14)$$

Grouping the R(i) and $Y_m(i)$ terms together in equation B-14 :

$$\frac{K_n(z)}{K_d(z)}\left[R(i)\ -Y_m(i)\right] = \frac{\omega(z)}{\delta(z)}\left[\ 1 - \frac{K_n(z)\ \ z^{-b}}{K_d(z)}\right]\ U(i) \qquad (B\text{-}15)$$

Rearranging equation B-15 :

$$\frac{U(z)}{R(i)-Y_m(i)} = \frac{U(z)}{E(z)} = \frac{\delta(z)}{\omega(z)}\cdot\frac{K_n(z)}{K_d(z)\ -\ K_n(z)z^{-b}} \qquad (B\text{-}16)$$

Equation B-16 is the same as the Dahlin controller in equation B-12.

MVC Controller :

The difference equation for a MVC controller is (see MacGregor et al. (1986)) :

$$U(i) = -\frac{\delta(z)}{\omega(z)} \cdot \frac{T(z)}{\phi(z)\nabla} \cdot \frac{1}{\psi_f(z)} Y_m(i) \tag{B-17}$$

Again, for minimum variance control, the control moves are not penalized (β = 0.0), hence equation 3-9 is :

$$\frac{K_n(z)}{K_d(z)} R(i) - \frac{\omega(z)}{\delta(z)} U(i) - \frac{T(z)}{\theta(z)} N(i) \qquad = 0.0 \tag{B-18}$$

MVC is a regulatory design, the servo trajectory is therefore, $K_n(z)/K_d(z)$ = 0.0. Substituting equation B-6 into equation B-18, one obtains :

$$\frac{\omega(z)}{\delta(z)} U(i) = - \frac{T(z)}{\theta(z)} \left[Y_m(i) - \frac{\omega(z)z^{-b}}{\delta(z)} U(i) \right] \tag{B-19}$$

Rearranging equation B-19 gives :

$$\frac{\omega(z)}{\delta(z)} \left[1 - \frac{T(z)z^{-b}}{\theta(z)} \right] U(i) =- \frac{T(z)}{\theta(z)} Y_m(i) \tag{B-20}$$

Using the relationship in equation B-5, equation B-20 becomes :

$$\frac{\omega(z)}{\delta(z)}\left[\frac{\psi_f(z)\nabla\ \phi(z)}{\theta(z)}\right]U(i) =- \frac{T(z)}{\theta(z)}\ Y_m(i) \qquad (B\text{-}21)$$

Rearranging equation B-21 produces:

$$U(i) =- \frac{\delta(z)}{\omega(z)}\ \frac{T(z)}{\psi_f(z)\nabla\ \phi(z)}\ Y_m(i) \qquad (B\text{-}22)$$

Equation B-22 is the same as the MVC in equation B-17.

APPENDIX C: UPPER AND LOWER BOUND ON THE MAGNITUDE OF A DISCRETE TRANSFER FUNCTION

Given a polynomial, $A(z)$, with z within the unit disk, $|z| < 1$

To prove that : $|A(z)| \leq \mu(A(z))$

(i.e., the magnitude is always bounded above by the spectral gain)

Proof:

The magnitude can be written as the square root of the power spectrum

$$| A(z) | = (A(z) A(z^{-1}))^{1/2} \tag{C-1}$$

Write the z-transformed expression as a convolution sum

$$=\{ \sum_{i=-n}^{n} z^{i} \sum_{l=0}^{n} a_{i+l} a_{l} \}^{1/2} \tag{C-2}$$

Since the maximum magnitude of an analytic function must occur at the boundary $|z|=1$, the following inequality holds :

$$| A(z) | \leq \sum_{i=-n}^{n} \left| \sum_{j=0}^{n} a_{i+j} a_{j} \right|^{1/2} = \mu(A(z)) \tag{C-3}$$

Therefore, the magnitude of $A(z)$ within the unit disk is always bounded by the spectral gain.

To prove that: $\sigma(A(z)) \geq \mu(A(z))$

(i.e., the supremal gain is always greater than the spectral gain)

Proof:

The inequality of $\sigma(A(z)) \geq \mu(A(z))$ can be established from equation C-3 using Triangular Inequality

$$\mu(A(z)) \leq [\sum_{i=-n}^{n} \sum_{j=0}^{n} |a_{i+j}||a_j|^{1/2} \tag{C-4}$$

Equation C-4 can be simplified as :

$$\mu(A(z)) \leq [\sum_{i=0}^{n} |a_i| \sum_{j=0}^{n} |a_j|]^{1/2} = \sigma(A(z)) \tag{C-5}$$

Thus, the supremal gain, σ, must be greater or equal to the spectral gain, μ.

To prove that: $$\frac{1}{\mu(1/A(z))} \lesssim \min(|A(z)|)$$

(i.e., the magnitude of a polynomial is bounded below by the spectral gain of the inverse polynomial)

Proof:

$$\text{Since} \quad |A(z)| = 1/(|1/A(z)|) \tag{C-6}$$

$$\min(|A(z)|) = \min(1/(|1/A(z)|)) \tag{C-7}$$

$$= 1/\ (\max(|1/A(z)|) \tag{C-8}$$

Since the maximum magnitude is bounded above by spectral gain

$$\geq 1/(\ \mu(1/A(z))) \tag{C-9}$$

Therefore, the minimum magnitude of A(z) can be bounded below by the inverse of the spectral gain, $1/\mu$.

Example :

$A(z) = z^2 + 3z - 1$

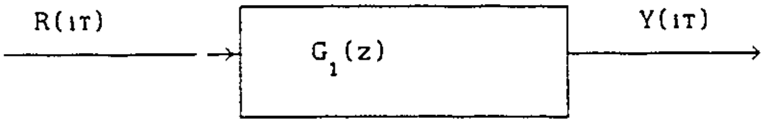
the supremal gain, $\sigma(A(z))$	=	5	(from C-5)
the spectral gain, $\mu(A(z))$	=	3.6	(from C-3)
the maximum gain of A(z)	=	4.4	(equation 4-19; N=20)

This demonstrates that the spectral gain is a tighter bound than the supremal gain.

Appendix D : Bounding the Output from a Pulse Response Block

In order to bound the output of a pulse response block, some lemmata have been developed to aid the bounding process. The key lies in identifying the source of the uncertainty (e.g., uncertainty in input sequence or pulse response block) and the nature of the uncertainty (e.g., time-invariant). It should also be pointed out that there is no unique way to bound the output and the difference between the various methods is in the tightness of the bounds obtained.

Lemma D-1: **To bound the output of an open loop uncertainty process $G_1(z)$**



Given :

If $G_1^{max}(i)$ and $G_1^{min}(i)$ are the upper and lower bound of the process impulse pulse response at time i:

$$G_1^{min}(i) \le G_1(i) \le G_1^{max}(i) \quad \forall \;\; i= 0,1,\ldots,N \tag{D-1}$$

and

$$Y(iT) = G_1 * R(iT) \qquad * : \text{Convolution} \qquad \text{(D-2)}$$

and

$$Gerr(i) = (G_1^{max}(i) - G_1^{min}(i)) / 2 \qquad \text{(D-3)}$$

$$Gpnom(i) = (G_1^{max}(i) + G_1^{min}(i)) / 2 \qquad \text{(D-4)}$$

$$\forall \quad i=0,1\ldots N$$

Then two bounds, $\bar{Y}(iT)$ and $Y'(iT)$, for $Y(iT)$ are :

$$|Y(iT)| \leq Y'(iT) = \sum_{j=0}^{N} \max (|G_1^{max}(j)|, |G_1^{min}(j)|).|R(iT-jT)| \qquad \text{(D-5)}$$

and

$$|Y(iT)| \leq \bar{Y}(iT) = \sum_{j=0}^{N} Gpnom(j)\ R(iT-jT) \qquad \text{(D-6)}$$

$$+ \sum_{j=0}^{N} |Gerr(j)\ R(iT-jT)|$$

and $\bar{Y}(iT)$ is a less conservative bound. Therefore,

$$|Y(iT)| \leq \bar{Y}(iT) \leq Y'(iT) \qquad \text{(D-7)}$$

Proof :

From the convolution equation in D-2, equation D-5 can be established by invoking the Triangular Inequality (see Jury (1964)). Equation D-5 denotes the convolution of the absolute values of the extremal impulse response weights of $G_1(z)$ with the absolute values of input sequence, $R(iT)$'s. This bound on $Y(z)$ can be reduced by transforming the uncertainty in $G_1(z)$ into $\mathrm{Gpnom}(z)$ and a symmetric error band, $\mathrm{Gerr}(z)$, as given in equations D-3 and D-4. The open loop uncertainty specified in D-1 can be rewritten as :

$$\mathrm{Gpnom}(i) - \mathrm{Gerr}(i) \le G_1(i) \le \mathrm{Gpnom}(i) + \mathrm{Gerr}(i) \qquad \text{(D-8)}$$

From D-1, a bound for $Y(iT)$, or $G_1(i) * R(iT)$, can be written as :

$$\begin{aligned}(\mathrm{Gpnom}(i) - \mathrm{Gerr}(i)) * R(iT) &\le G_1(z) * R(iT) \\ &\le (\mathrm{Gpnom}(i) + \mathrm{Gerr}(i)\ ) * R(iT) \qquad \text{(D-9)}\end{aligned}$$

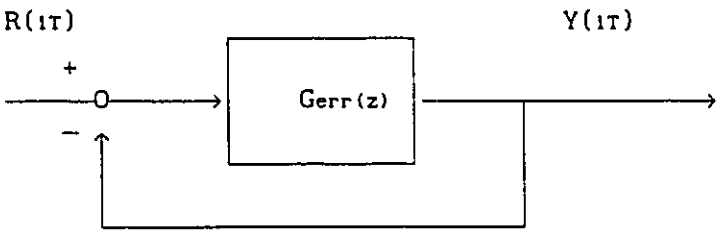
Since the uncertainty is only in $\mathrm{Gerr}(i)$, not in $\mathrm{Gpnom}(i)$ or $R(iT)$, the term $G_1(i) * R(iT)$ or $Y(iT)$ can be bounded above and below by :

$$|Y(iT)| \le \mathrm{Gpnom} * R(iT) + |\mathrm{Gerr}(i)| * |R(iT)| \qquad \text{(D-10)}$$

where the term $|\mathrm{Gerr}(i)| * |R(iT)|$ indicates the bound about the nominal response $\mathrm{Gpnom}(i) * R(iT)$. Since $|\mathrm{Gerr}(i)|$ is always less than or equal to $\max(|G_1(i)|, |G_2(i)|)$ for all i, the bound from D-10 is smaller than that from D-5 and hence

$$|Y(iT)| \le \bar{Y}(iT) \le Y'(iT) \qquad \text{(D-11)}$$

Lemma D-2 : **To bound the output of a FEEDBACK uncertain process (with uncertainty in Gerr(z)**



Given :

If Gerr(i) is bounded symmetrically about zero by $|\bar{G}err(i)|$

$$-|\bar{G}err(i)| \leq Gerr(i) \leq |\bar{G}err(i)|, \qquad \forall \quad i = 0,1,\ldots,N \tag{D-12}$$

Then the bound is :

$$Y(iT) \leq \bar{Y}(iT) = |\bar{G}err(z)| * |R(iT)| + |\bar{G}err(z)| * \bar{Y}(iT) \tag{D-13}$$

Proof :

$G_{err}(z)$ is a pulse transfer function with the impulse weights bounded by $|\bar{G}_{err}(z)|$ and $-|\bar{G}_{err}(z)|$ symmetrically about zero. The process block diagram above can be written as a convolution sum of $R(iT)$ and $G_{err}(i)$ as follows.

$$Y(iT) = G_{err}(i) * (R(iT) - Y(iT)) \qquad \text{(D-14)}$$

By Triangular Inequality, the output magnitude of $Y(iT)$ can be bounded by :

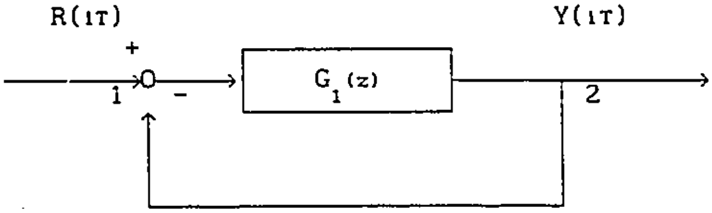
$$|Y(iT)| \le \bar{Y}(iT) = |\bar{G}_{err}| * |R(iT)| + |\bar{G}_{err}| * \bar{Y}(iT) \qquad \text{(D-15)}$$

or in Z-transform space :

$$\bar{Y}(z) = \frac{|\bar{G}_{err}(z)|\,|R(z)|}{1 - |\bar{G}_{err}(z)|} \qquad \text{(D-16)}$$

Equations D-15 and D-16 provide two ways to calculate the bound on $Y(iT)$, either through a convolution sum (D-15) or through a z-transform (D-16).

Lemma D-3 : **To bound the output of a FEEDBACK uncertain process with uncertainty in R(iT)**



Given :

$$-|\bar{R}(iT)| \le R(iT) \le |\bar{R}(iT)| \qquad \forall \qquad i=0,1\ldots N \tag{D-17}$$

Then

$$|Y(z)| \le \bar{Y}(z) = \left| \frac{G_1(z)}{1 + G_1(z)} \right| \cdot |\bar{R}(z)| \tag{D-18}$$

Proof :

R(iT) is the input to the block diagram. The uncertainty occurs only in the input sequence, R(iT)'s, where the magnitude of the R(iT)'s are bounded by |R(iT)|. Since there is no uncertainty inside the feedback loop, the loop transfer from point 1 to point 2 is :

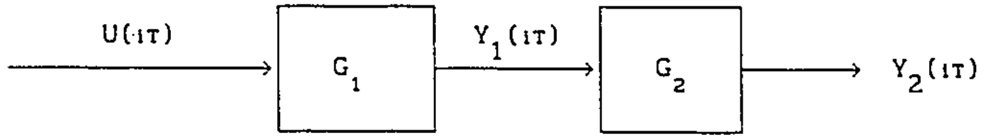
$$Y(z) = \left(\frac{G_1(z)}{1 + G_1(z)} \right) R(z) \tag{D-19}$$

As far as the bounding is concerned, this problem is similar to bounding an open loop uncertain process. Therefore Lemma D-1 can be invoked.

$$|Y(z)| \le \bar{Y}(z) = \left|\frac{G_1(z)}{1 + G_1(z)}\right| \cdot |\bar{R}(z)| \qquad \text{(D-20)}$$

Note the difference between equations D-20 and D-16 in terms of the bounds. The bounds from equation D-20 are always smaller than those in equation D-16. This suggests that one has to be careful in bounding the closed loop output. The key is to pinpoint the location of uncertainty. Failure to do so will result in an unnecessarily large bound on the output.

Lemma D-4 : Implication of Time Invariance to Bounding



If the processes, $G_1(i)$ and $G_2(i)$, are Time Invariant,

Then

$$Y_2(iT) = G_2(i) * G_1(i) * U(iT) = G_1(i) * G_2(i) * U(iT) \qquad \text{(D-21)}$$

Proof :

If a process, $G(z)$, is time invariant, impulse weight, $G(j)$ of $G(z)$ at lag j is not a function of time i. The output, $Y_2(iT)$, in the block diagram can be expressed as a convolution of pulse train as shown in equation D-22.

$$Y_2(iT) = G_2(i) * G_1(i) * U(iT) \qquad \text{(D-22)}$$

or

$$Y_2(iT) = \sum_{j=0}^{N} G_2(j) \sum_{k=0}^{N} G_1(k) \; U((i-k-j)T) \qquad \text{(D-23)}$$

Interchanging the sequence of summation :

$$Y_2(iT) = \sum_{j=0}^{N} \sum_{k=0}^{N} G_2(j) \quad G_1(k) \quad U((i-k-j)T) \tag{D-24}$$

Since $G_1(k)$ is not a function of the time indices (i or j),

$$Y_2(iT) = \sum_{k=0}^{N} G_1(k) \sum_{j=0}^{N} G_2(j) \quad U((i-k-j)T) \tag{D-25}$$

Another way to write D-25 is

$$Y_2(iT) = G_1(i) * G_2(i) * U(iT) \tag{D-26}$$

Combining D-22 and D-26, we give the result in D-21. In another word, Lemma D-4 implies that the sequence of convolution can be exchanged for time invariant systems.

Proved.

Lemma D-5 : IMPLICATION OF Lemma D-4 TO THE BOUNDING PROCEDURE

Given the block diagram in Lemma D-4 :

Assumed that $U(iT)$, G_2 have no uncertainty

The uncertainty in G_1 is specified by symmetric bounds :

$$G_1 : \quad -|\bar{G}_1(i)| \le G_1(i) \le |\bar{G}_1(i)| \qquad \forall \quad i = 0.... N \qquad \text{(D-27)}$$

The convolution sum for the output is :

$$Y_2(iT) = G_2(i) * G_1(i) * U(iT) \qquad \text{(D-28)}$$

If the processes, $G_1(i)$ and $G_2(i)$, are Time Invariant

Then

$$|Y_2(iT)| \quad \le \sum_{j=0}^{N} |G_2(j)| \sum_{k=0}^{N} |\bar{G}_1(k)\ U((i-j-k)T)|$$

and

$$\le \sum_{j=0}^{N} |\bar{G}_1(j)| \left|\sum_{k=0}^{N} G_2(k)\ U((i-j-k)T)\right| \qquad \text{(D-29)}$$

Proof:

By Lemma D-1, the output, $Y_2(iT)$, in the block diagram can be bounded by :

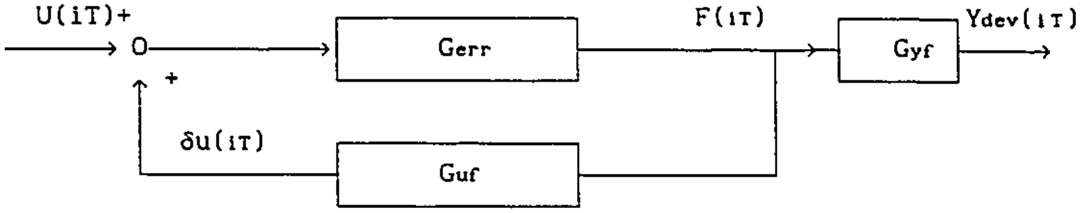
$$|Y_2(iT)| \le \sum_{j=0}^{N} |G_2(j)| \sum_{k=0}^{N} |\bar{G}_1(k)| \, |U((i-j-k)T)| \qquad \text{(D-30)}$$

By the time invariant criteria in Lemma D-4, the pulse transfer function block can be swapped. Since there are no uncertainty in $G_2(i)$ and $U(iT)$, a tighter bound on $Y_2(iT)$ can be obtained.

$$\begin{aligned} |Y_2(iT)| &\le \sum_{j=0}^{N} |\bar{G}_1(j)| \, \Big| \sum_{k=0}^{N} G_2(k) \; U((i-j-k))T \Big| && \text{(D-31)} \\ &\le \sum_{j=0}^{N} |G_2(j)| \sum_{k=0}^{N} |\bar{G}_1(k)| \, |U((i-j-k)T)| \end{aligned}$$

The second expression in the RHS of D-31 is always greater than or equal to the first expression. Therefore, by applying the time invariant criteria, one can get a tighter bound on the output.

Lemma D-6 : Bounding the close loop output response deviation



Given :

If

$$F(iT) = Gerr(i) * U(iT) + Gerr(i) * Guf(i) * F(iT) \tag{D-32}$$

and

$$Y_{dev}(iT) = Gyf(i) * F(iT) \tag{D-33}$$

and

$$-|Gerr(i)| \leq Gerr(i) \leq |Gerr(i)| \;\; \forall \;\; i=0,1,\ldots,N \tag{D-34}$$

Then the Time-Invariant bound for $Y_{dev}(iT)$ is :

$$Y_{dev}(iT) = Gyf(z) \frac{Gerr(z)}{1- Gerr(z)Guf(z)} U(iT) \tag{D-35}$$

$$Umod(iT) = Gyf(i) * U(iT) \tag{D-36}$$

$$\begin{aligned}\overline{Y}_{dev}(iT) =& |\overline{G}err(i)| * |Umod(iT)| \\ &+ |\overline{G}err(i)| * |Guf(i)| * \overline{Y}_{dev}(iT)\end{aligned} \tag{D-37}$$

Proof :

For the Time-Invariant case, by Lemma D-4, one can write equations D-32 and D-33 as :

$$Y_{dev}(iT)=G_{err}(i)*(G_{yf}(i)*U(iT)) + G_{err}(i)*G_{uf}(i)*Y_{dev}(iT) \qquad (D\text{-}38)$$

Since there is no uncertainty in $G_{yf}(i)*U(iT)$, the convolution can be normally operated to obtain $Umod(iT)$ as shown in equation D-36. Applying Lemma D-2 to equation D-10 yields :

$$\begin{aligned}|Y_{dev}(iT)| \le \overline{Y}_{dev}(iT) &= |\overline{G}_{err}(i)|*|Umod(iT)| \\ &\quad +|\overline{G}_{err}(i)|*|G_{uf}(i)|*\overline{Y}_{dev}(iT)\end{aligned} \qquad (D\text{-}39)$$

Proved

Appendix D-7 : Equivalence of Block Diagrams in Figures 5-2a and 5-2b

The z-transformed output in Figures 5-2a and 5-2b can be written as equations D-40 and D-41 :

$$Y(z) = \frac{Gc(z)\ (\ Gp(z) + Gerr(z)\ )(R(z) - N(z))}{1 + Gc(z)\ (\ Gp(z) + Gerr(z))} + \frac{D(z)}{1 + Gc(z)\ (\ Gp(z) + Gerr(z))} \qquad (D\text{-}40)$$

$$Y(z) = \frac{Gc(z)\ Gp(z)(R(z) - N(z))}{1 + Gc(z)\ Gp(z)} + \frac{D(z)}{1 + Gc(z)\ Gp(z)} + \frac{Gc(z)\ Gerr(z)\ Gyf(z)\ (R(z)-N(z)-D(z))}{(1+Gc(z)\ Gp(z))(1-Gerr(z)\ Guf(z))} \qquad (D\text{-}41)$$

where

$$Guf(z) = -\frac{Gc(z)}{1 + Gc(z)Gp(z)} \qquad (D\text{-}42)$$

$$Gyf(z) = \frac{1}{1 + Gc(z)Gp(z)} \qquad (D\text{-}43)$$

The section followed will show the equivalence between equations D-40 and D-41. Substitute Guf(z) and Gyf(z) in D-42, D-43 into D-41:

$$Y(z)= \frac{Gc(z)\ Gp(z)(R(z)-\ N(z))}{1\ +\ Gc(z)\ Gp(z)} + \frac{D(z)}{1\ +\ Gc(z)\ Gp(z)} \qquad (D\text{-}44)$$

$$+ \frac{Gc(z)\ Gerr(z)\ (R(z)-N(z)-D(z))\ \dfrac{1}{(1+Gc(z)Gp(z))}}{(1+Gc(z)\ Gp(z))(1+Gerr(z)\ \dfrac{Gc(z)}{1\ +\ Gc(z)Gp(z)}\ )}$$

Simplify equation D-44,

$$Y(z)= \frac{Gc(z)\ Gp(z)(R(z)-\ N(z))}{1\ +\ Gc(z)\ Gp(z)} + \frac{D(z)}{1\ +\ Gc(z)\ Gp(z)} \qquad (D\text{-}45)$$

$$+ \frac{Gc(z)\ Gerr(z)\ (R(z)-N(z)-D(z))}{(1+Gc(z)\ Gp(z))(1+Gc(z)Gp(z)+Gc(z)Gerr(z))}$$

Group terms in equation D-45 into servo response and disturbance responses :

$$Y(z)\ = \frac{\{(Gc(z)Gp(z)(1+Gc(z)Gp(z)+Gerr(z)Gc(z))+Gc(z)Gerr(z)\}(R(z)-N(z))}{(1+Gc(z)\ Gp(z))\ (\ 1+\ Gc(z)Gp(z)\ +\ Gc(z)Gerr(z))}$$

$$+ \frac{(\ 1+Gc(z)Gp(z))\ D(z)}{(1+Gc(z)Gp(z))\ (1+Gc(z)Gp(z)+Gc(z)Gerr(z)))} \qquad (D\text{-}46)$$

Simplify equation D-46,

$$Y(z) = \frac{(1+Gc(z)Gp(z))\ (Gc(z)Gp(z)+Gc(z)Gerr(z))\ (R(z)-N(z))}{(1+Gc(z)\ Gp(z))(\ 1+\ Gc(z)Gp(z)\ +\ Gc(z)Gerr(z))} + \frac{D(z)}{1\ +\ Gc(z)Gp(z)\ +\ Gc(z)Gerr(z))} \qquad (D\text{-}47)$$

Further simplify equation D-47,

$$Y(z) = \frac{Gc(z)\ (\ Gp(z)\ +\ Gerr(z)\ )(R(z)\ -\ N(z))}{1\ +\ Gc(z)\ (\ Gp(z)\ +\ Gerr(z))} + \frac{D(z)}{1\ +\ Gc(z)\ (\ Gp(z)\ +\ Gerr(z))} \qquad (D\text{-}48)$$

After the rearrangements and simplifications, equation D-40 becomes exactly the same as equation D-43. Thus Figure 5-2a and Figure 5-3b are equivalent mathematically.