0-315-24886-6

National Library of Canada — Bibliothèque nationale du Canada
Canadian Theses Division — Division des thèses canadiennes
Ottawa, Canada
K1A 0N4

PERMISSION TO MICROFILM — AUTORISATION DE MICROFILMER

• Please print or type — Ecrire en lettres moulées ou dactylographier

Full Name of Author — Nom complet de l'auteur

BENJAMIN ANDREW SMILLIE

Date of Birth — Date de naissance: MAY 9 1959

Country of Birth — Lieu de naissance: CANADA

Permanent Address — Résidence fixe

3005 HANOVER AVE
SASKATOON, SASK.
S7J 1E3

Title of Thesis — Titre de la thèse

AN EVALUATION OF INTERNAL MODEL CONTROL

University — Université

UNIVERSITY OF ALBERTA

Degree for which thesis was presented — Grade pour lequel cette thèse fut présentée

MASTERS OF SCIENCE

Year this degree conferred — Année d'obtention de ce grade: 1984

Name of Supervisor — Nom du directeur de thèse: Dr. R.K. WOOD

Date: September 25 1984

Signature: Andrew Smillie

NL-91 (4/77)

THE UNIVERSITY OF ALBERTA

AN EVALUATION OF INTERNAL MODEL CONTROL

by

© B. ANDREW SMILLIE

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF MASTERS OF SCIENCE

IN

PROCESS CONTROL

DEPARTMENT OF CHEMICAL ENGINEERING

EDMONTON, ALBERTA

FALL, 1984

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR B. ANDREW SMILLIE

TITLE OF THESIS AN EVALUATION OF INTERNAL MODEL CONTROL

DEGREE FOR WHICH THESIS WAS PRESENTED MASTERS OF SCIENCE

YEAR THIS DEGREE GRANTED FALL, 1984

(SIGNED) Andrew Smillie .....

PERMANENT ADDRESS:

2005 Hanover Avenue

Saskatoon, Saskatchewan

S7J 1E3

DATED September 25 1984

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled AN EVALUATION OF INTERNAL MODEL CONTROL submitted by B. ANDREW SMILLIE in partial fulfilment of the requirements for the degree of MASTERS OF SCIENCE in PROCESS CONTROL.

....

Supervisor

....R. E. Rink....

....D. G. Fisher....

Date....SEPT. 5, 1984....

To my parents

ABSTRACT

In the past few years, several digital control algorithms have been proposed that incorporate a nonparametric description of the process into the control loop. One such algorithm is internal model control(IMC); applicable to single input-single output(SISO) as well as multiple input–multiple output(MIMO) applications. A review of IMC is carried out to investigate the merits of this algorithm.

The evaluation of IMC begins with a thorough examination of design and stability properties for the single variable IMC predictive controller. The results of simulation studies, using second order linear differential equations that model minimum phase, nonminimum phase(NMP) or unstable process behavior, are presented to illustrate the single variable IMC tuning procedure.

The single variable IMC development is followed by examination of the multivariable IMC properties. In this section special emphasis is given to the design and implementation of the IMC time delay precompensator. Analogous to the single variable development, results of simulation studies are presented to demonstrate the application of the IMC time delay precompensator and MIMO controller. These studies use a transfer function matrix model of a distillation column to represent process behavior.

The evaluation of IMC concludes with a comparison of IMC to the Smith predictor, the inferential Smith predictor, dynamic matrix control, model algorithmic control, the self-tuning controller and the multivariable Ogunnaike and Ray time delay compensator from the viewpoint of theory and implementation. Simulation results using both linear and nonlinear stochastic process models are presented to support the comparison. Also, a discussion of the options available to identify nonparametric process models in deterministic or stochastic environments is included.

The examination of IMC reveals truncation error in the nonparametric process representation often leads to severe deterioration in control performance. In addition the comparison demonstrates the weaknesses of the trial and error IMC tuning procedure for NMP processes or multivariable processes described by a transfer function matrix with the minimum time delay in an off-diagonal position.

ACKNOWLEDGEMENT

The author would like to thank his thesis supervisor, Dr. R.K. Wood, for the assistance and encouragement he has provided during the preparation of this thesis. Appreciation is also extended to Bob Barton and the other DAC centre staff in the Chemical Engineering Department for their assistance.

The intangible but very important contribution made by the Chemical Engineering graduate students through their humor, sarcasm and ever present support is valued and will be fondly remembered.

Lastly the financial support provided by the Department of Chemical Engineering U. of A., and the Natural Sciences and Engineering Research Council is gratefully acknowledged.

Table of Contents

List of Figures

Figure		Page

Figure | Page

Figure Page

NOMENCLATURE

a_i....................denominator coefficients of ARMA model

A.....matrix of coefficients describing DMC process dynamics

A'...........................diagonal elements of matrix A'A

A,B,C.....polynomials of order n corresponding to the system output, control input, and disturbance input, respectively

b_i.....................numerator coefficients of ARMA model

C(z)......................conventional PI or PID controller

$\underline{C}$(z)......conventional multivariable PI controller, diagonal matrix

d(k)..........disturbance input to process at time interval k

$\hat{d}$(k)...........................IMC error signal $(y(k)-y_m(k))$

$\underline{d}$(k)...........................vector of load disturbances

Dc(z).....polynomial containing roots to controller transfer function.

dia..............indicates the diagonal elements of a matrix

e(k)...............feedback error signal at time interval k

E(k)........vector of projected errors for DMC, IMC, and MAC algorithms

E,F,G.general polynomials in the backward shift operator z^{-1} used in the STC control law

F(z)......................single variable exponential filter

$\underline{F}$(z).............diagonal matrix of SISO exponential filters

G(z)...............single variable process transfer function

$\hat{G}_+$(z)...process transfer function containing time delays and NMP characteristics

$\hat{G}_-$(z)...process transfer function without time delays or NMP characteristics

$\underline{G}(z)$.........multivariable process transfer function matrix

$\acute{G}(z)$........single variable process transfer function model

$\underline{\acute{G}}$......multivariable process model transfer function matrix

$\underline{\acute{G}}_{+1}(z)$...precompensator matrix, contains time delays and may contain process NMP characteristics

$\underline{\acute{G}}_{+2}(z)$.........precompensator matrix, used to stabilize the realizable process model inverse, $\underline{\acute{G}}_{-}(z)^{-1}$

$Gc(z)$........................single variable IMC controller

$G\underline{c}$...........................multivariable IMC controller

$\underline{G}^{+}$........delay free multivariable process transfer function

$G_i(s)$.....................................ISP controller

$\underline{\acute{G}}_{I}(z)$.....multivariable steady state decoupler for OR system controller

$H(z)$.......process impulse response with time delays removed

$\acute{H}$....process model impulse response with time delays removed

$\underline{H}(z)$.....process impulse response matrix equivalent to $\underline{\acute{G}}_{-}(z)$

$\underline{I}$...................................the identity matrix.

ITD...............time delay in units of sampling intervals

Kc.....................................PI controller gain

M...........input suppression parameter for IMC control law

$m(k)$.....................output signal from SISO controller

$\underline{m}(k)$.........................output from MIMO controller

N..........number of coefficients in impulse response series

$Nc(z)$....polynomial containing numerator coefficients to IMC controller

$\underline{O}$.........................multivariable offset compensator

P....................optimization horizon in IMC control law

P,Q,R ....costing transfer functions acting on the system output, input and setpoint, respectively

s(k)....................setpoint for single variable process

$\underline{s}(k)$...........vector of setpoints for multivariable process

T...sampling rate

u(k)..........manipulated variable signal at time interval k

U(k).......vector of predicted control inputs for the single variable process

$\underline{u}(k)$.............vector of control inputs at time interval k

$\underline{U}(k)$....matrix of predicted control inputs for multivariable process

V(k)........vector of past control inputs at time interval k

$\underline{V}(k)$ ....matrix of past control inputs for multivariable process

w(k)........................single variable setpoint for STC

y(k)..output from single variable process at time interval k

$y_m(k)$...........output from process model at time interval k

yr(k).......reference trajectory used in the MAC control law

$\underline{y}(k)$....vector of outputs from multivariable process at time interval k

$\underline{y}^{+}(k)$..................time delay compensated process output

$y_d(k)$.........................setpoint or desired trajectory

α..............................filter weighting parameter

β_j..................input weighting parameter at interval k+j

$\underline{\beta}_j(k)$ ..diagonal matrix of weighting factors, for all inputs at time k+j

B..........matrix of input weighting parameters for M steps

$\underline{B}$..matrix of input weighting parameters over M steps for the multivariable process

$\underline{\Delta}(j)$..matrix of weighted predicted output errors at time k+j

$\underline{\nabla}(j)$.........matrix of weighted control actions at time k+j

$\xi(k)$....................................noise input signal

γ_j.......output weighting parameter for IMC at interval k+j

$\underline{\gamma}_j$..........vector of output weightings at interval k+j for multivariable process

$\underline{\Gamma}$....matrix of output weightings for entire single variable horizon of P steps

$\underline{\underline{\Gamma}}$......matrix of output weighting for multivariable process

ϕ................................PI controller reset value

$\phi_y(k|k-\tau)$..prediction of Py(k) at time interval $k-\tau$, used in STC control law

τ................time delay in number of sampling intervals

τ_i^* ......time delay on diagonal element i, of precompensator matrix

τ_{ij}.......minimum time delay in each row of transfer matrix

τ_o........measure of imbalance in process transfer function

1. INTRODUCTION

New methods to optimally control process operations are continually evolving and changing. In the past five years several digital control algorithms have been proposed which incorporate a nonparametric description of the process dynamics into the control loop. Internal model control(IMC) introduced by Garcia and Morari[1,2,3], exemplifies this type of control policy. The fundamental principles that govern IMC as well as the implementation of IMC for single and multivariable processes will be reviewed in this work. In addition a comparison of IMC to other nonparametric control algorithms and to other optimal control algorithms is used to weigh the relative advantages and disadvantages of IMC.

Internal model control is a comprehensive control policy for single input, single output (SISO) or multiple input, multiple output (MIMO) systems. The authors refer to related work carried out by Brosilow[4], Cutler and Ramaker[5,6] and Smith[7] because these control algorithms are very similar to IMC.

The investigation of IMC is split into three chapters. In Chapter 2 the theory and application of IMC for single variable processes are developed. The special features that IMC incorporates to control multivariable processes are examined in Chapter 3. Both chapters begin by deriving the fundamental properties followed by simulation studies that demonstrate the tuning procedures for the single or multivariable IMC controller and illustrate, how the controller parameters are applied to achieve desirable servo and regulatory behavior.

The information provided in Chapters 2 and 3 is used in Chapter 4 to compare IMC to several single and one multivariable control algorithm. Chapter 4 effectively highlights the strengths and weaknesses of the IMC control policy. Chapter 5 summarizes the important results of the investigation and draws conclusions from these results.

2. INTERNAL MODEL CONTROL FOR SISO SYSTEMS

2.1 IMC for SISO Systems

The following material will deal exclusively with SISO systems but it should be recognized that many of the design objectives, tuning procedures, stability properties etc. are applicable to MIMO systems.

2.1.1 Criteria used to evaluate the performance of a control system

Garcia and Morari[1] state four criteria used to evaluate the performance of any control system. These criteria are

i. Regulatory behavior:

   The output variables are to be kept at their setpoint despite unmeasured disturbances affecting the process.

ii. Servo behavior:

   Changes in setpoint should be tracked quickly and smoothly.

iii. Robustness:

   Stability and control performance should be maintained in the face of structural and parametric changes in the underlying process model. This is equivalent to requiring that it be possible to design the controller with minimal a priori process information.

iv. Ability to deal with constraints:

For optimizing control applications where the operation is close to process constraints the controller should be able to guarantee safe operation.

These criteria do not include the response of the controller to stochastic disturbances since these effects are considered to be of secondary importance.

2.2 IMC structure and properties

The conventional feedback controller and the IMC controller block diagrams are shown in Figures 2.1 and 2.2. The IMC controller is essentially a feedforward controller with the feedback loop used to correct for process/model mismatch and unmeasured disturbances. The IMC controller structure can be made equivalent to conventional controllers in the following manner. If the process input-output relationship is represented by the equation

$$y(z) = G(z)m(z) + d(z) \qquad 2.2.1$$

then for the control scheme shown in Figure 2.1, the controller output is given by

$$m(z) = C(z)(s(z)-y(z))$$

The IMC controller, as can be seen from Figure 2.2 calculates the controller output based upon the equation

$$m(z) = \frac{Gc(z)(s(z)-y(z))}{1-Gc(z)\hat{G}(z)} \qquad 2.2.2$$

so it follows that the two control schemes are equivalent if C(z) is defined as

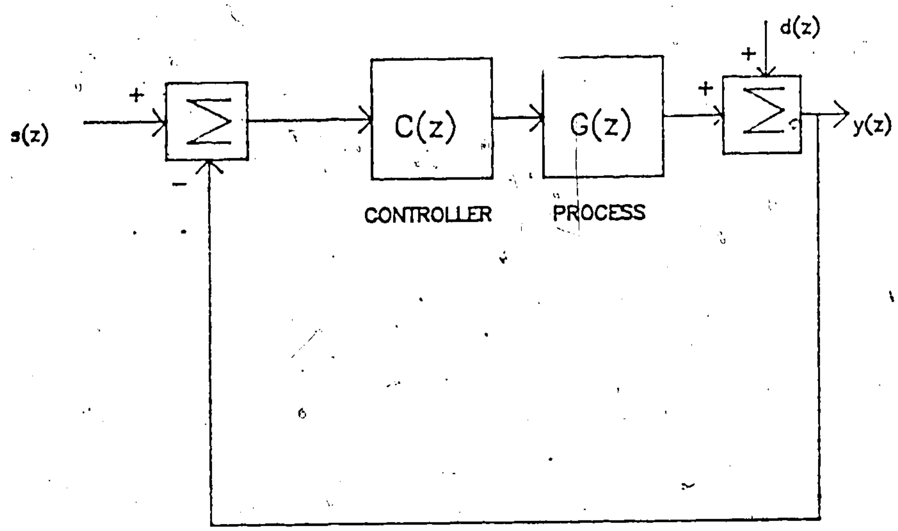

Figure 2.1 Block diagram of conventional feedback control

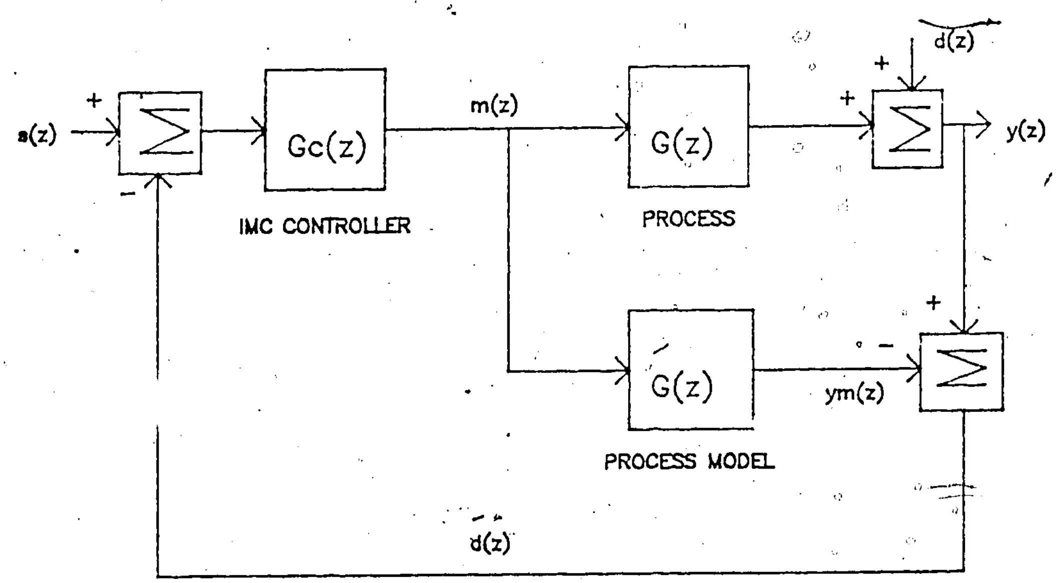

Figure 2.2 Block diagram of basic IMC structure

$$C(z)=\frac{Gc(z)}{1-Gc(z)\hat{G}(z)} \qquad 2.2.3$$

The IMC configuration is purported to be superior to the conventional feedback control scheme since the IMC controller is easier to design and includes robustness as a design objective due to the nature of the feedback signal. The feedback signal $\hat{d}(z)$ represents the difference between the process output and the model output. For a correct process model, the signal $\hat{d}(z)=d(z)$. If no disturbance exists and the model is correct, $\hat{d}(z)=0$. This feedback signal adds robustness to the control loop because it corrects for plant/model mismatch and unmeasured disturbances.

Combination of equations 2.2.1 and 2.2.2 to eliminate m(z) allows the closed loop transfer function to be written as

$$y(z) = \frac{G(z)Gc(z)(s(z)-d(z))}{1+Gc(z)(G(z)-\hat{G}(z))} + d(z) \qquad 2.2.4$$

For bounded setpoint or disturbance effects, stability is ensured as long as the roots of the characteristic equation

$$1+Gc(z)(G(z)-\hat{G}(z)) = 0 \qquad 2.2.5$$

remain inside the unit circle. From the closed loop characteristic equation it is easily demonstrated, for an open loop stable process and stable controller, that if $\hat{G}(z)=G(z)$, then the closed loop system will always be stable. A process that exhibits open loop unstable characteristics (characteristic roots outside the unit circle) should first be stabilized using conventional controllers.

Furthermore if the process model satisfies the condition $\acute{G}(1)=G(1)$, it can be shown, as demonstrated in Appendix A, that zero offset will result.

2.3 Tuning the IMC controller

From equation 2.2.4 it would appear that an obvious choice for $Gc(z)$ is $1/\acute{G}(z)$. Even if $\acute{G}(z)\neq G(z)$, with this choice for $Gc(z)$ equation 2.2.4 reduces to

$$y(z) = s(z) \tag{2.3.1}$$

However this choice for $Gc(z)$ has serious drawbacks. By reference to equation 2.2.2 using $Gc(z)=1/\acute{G}(z)$, it follows that this choice is equivalent to advocating an infinite gain controller since the denominator in this equation will equal zero for this choice of $Gc(z)$. Also from equation 2.2.4 the best controller results in equation 2.3.1 even if the model is incorrect. Obviously the definition of the best controller must be qualified.

The aforementioned problems exist since the best controller is not realizable due to the inherent one step sampling delay introduced by the discrete representation of a continuous process, and due to transport delay. Inverting a process model which contains time delay results in a noncausal controller. Garcia and Morari[1] resolve this problem using two approaches. The first approach factorizes $\acute{G}(z)$ into two parts

$$\acute{G}(z) = \acute{G}_{+}(z)\acute{G}_{-}(z) \tag{2.3.2}$$

$$Gc(z) = 1/\acute{G}_{-}(z) \qquad 2.3.3$$

$\acute{G}_{+}(z)$ will contain all of the time delays and nonminimum phase (NMP) characteristics of $\acute{G}(z)$. The NMP characteristic refers to a transfer function with zeros outside of the unit circle. If these zeros were left in $\acute{G}_{-}(z)$, the resulting controller would have poles outside the unit circle. An example illustrating the design of $\acute{G}_{+}(z)$ is shown in Appendix B. The factorization is optimal since the controller given by equation 2.3.3, results in a controller that drives the discrete process, to the setpoint, in the shortest possible time

if $\acute{G}(z) = G(z)$ and $d(z)=0$ then

$y(z) = \acute{G}_{+}(z)s(z)$

Garcia and Morari[1] have denoted this controller, the 'perfect' controller. It is equivalent to a controller that cancels all process transfer function poles and zeros, inside the unit circle. This type of controller is also commonly called the deadbeat controller[8]. For practical purposes the perfect controller is often unacceptable because the controller drives the process output to the setpoint without considering the other process states. As a consequence, unacceptable oscillations between samples occur.

Since a high order process would be difficult to factorize, and since computer implementation of even simple factorizations is difficult, a second approach is discussed in the next section.

2.4 The predictive control problem

It is necessary to revise the IMC design approach not only for the reasons already mentioned but also to facilitate on-line tuning and to include input and output constraints as a design objective. Garcia and Morari present the predictive control law as a satisfactory solution to these requirements. The process, previously described by an autoregressive model, is now described by an impulse response model which is written as

$$y(z) = z^{-t-1}H(z)u(z) + d(z) \qquad 2.4.1$$

and allows a prediction of the process output based entirely upon past inputs. This model representation is not easily factored to remove NMP characteristics, instead the tuning parameters essentially manipulate the original model until an approximation of the process results that provides satisfactory control dynamics. As mentioned in Section 2.3, the previous design method will often result in a controller with very poor dynamics. By reformulating the process model and including tuning parameters, this new method will allow on-line tuning to remove undesirable effects.

The IMC control policy calculates control outputs m, over a horizion of P samples into the future. These outputs mimimize the sum of the squared errors in the costing function given by

$$J = \min_{m(k)}\{ \sum_{j=1}^{P} [\gamma_j^2 (y_d(k+\tau+j) - y(k+\tau+j|k))^2 + \sum_{j=1}^{M} \beta_j^2 m(k+j-1)^2]\} \quad 2.4.2$$

.
.
.

subject to:

$$y(k+\tau+j|k) = y_m(k+\tau+j|k) + d(k+\tau+j|k) \quad 2.4.3$$

$$y_m(k+\tau+j|k) = h_1 m(k+j-1) + h_2 m(k+j-2) + \ldots + h_N m(k+j-N) \quad 2.4.4$$

As can be seen, equation 2.4.3 requires a prediction of the process disturbance $d(k+\tau+j|k)$, over the horizon. The IMC policy instead of employing a prediction, assumes that the best estimate of this disturbance is given by the present feedback signal $\hat{d}(z)$. Use of the costing function expressed by equation 2.4.2 penalizes excursions of both the controlled and manipulated variables so the trajectory followed by either variable can be influenced to conform to operating constraints. The control law that results from the solution to this minimization is defined in Appendix C.

2.5 Tuning parameters

The five tuning parameters used with IMC to alter the process output and or controller action are the following

i. Input suppression parameter (M):

The input suppression parameter specifies the number of time steps, M, over which the control input is allowed to vary. For example, choosing M equal to three and P equal to ten, implies the first three control outputs m, can change but the remaining seven must be one constant value. As demonstrated in Appendix D, any NMP process can be stabilized by choosing M sufficiently small and P sufficiently large. Reducing M will reduce the extreme excursions of the manipulated variables giving, in general, a more desirable response.

ii. Input penalty parameter (β_j):

This penalty or weighting parameter influences the manipulations of the controller output. It is demonstrated in Appendix D, that by choosing β_j sufficiently large, any NMP process can be stablized. Increasing the magnitude of β_j will decrease the action taken by the manipulated variables making the system more sluggish. Also using a $\beta_j \neq 0$ will lead to offset since

$$Nc(1)/Dc(1) \neq 1/H(1)$$

This is corrected by the use of an offset compensator shown in Figure 2.3.

iii. Output penalty parameter (γ_j):

The output penalty or weighting parameter affects the excursions of the process output. Since the IMC costing function uses only two weighting parameters, it is the ratio of β_j/γ_j that determines which variable is penalized the greatest. Increasing the magnitude of γ_j for constant β_j will reduce the excursions of the control variable.

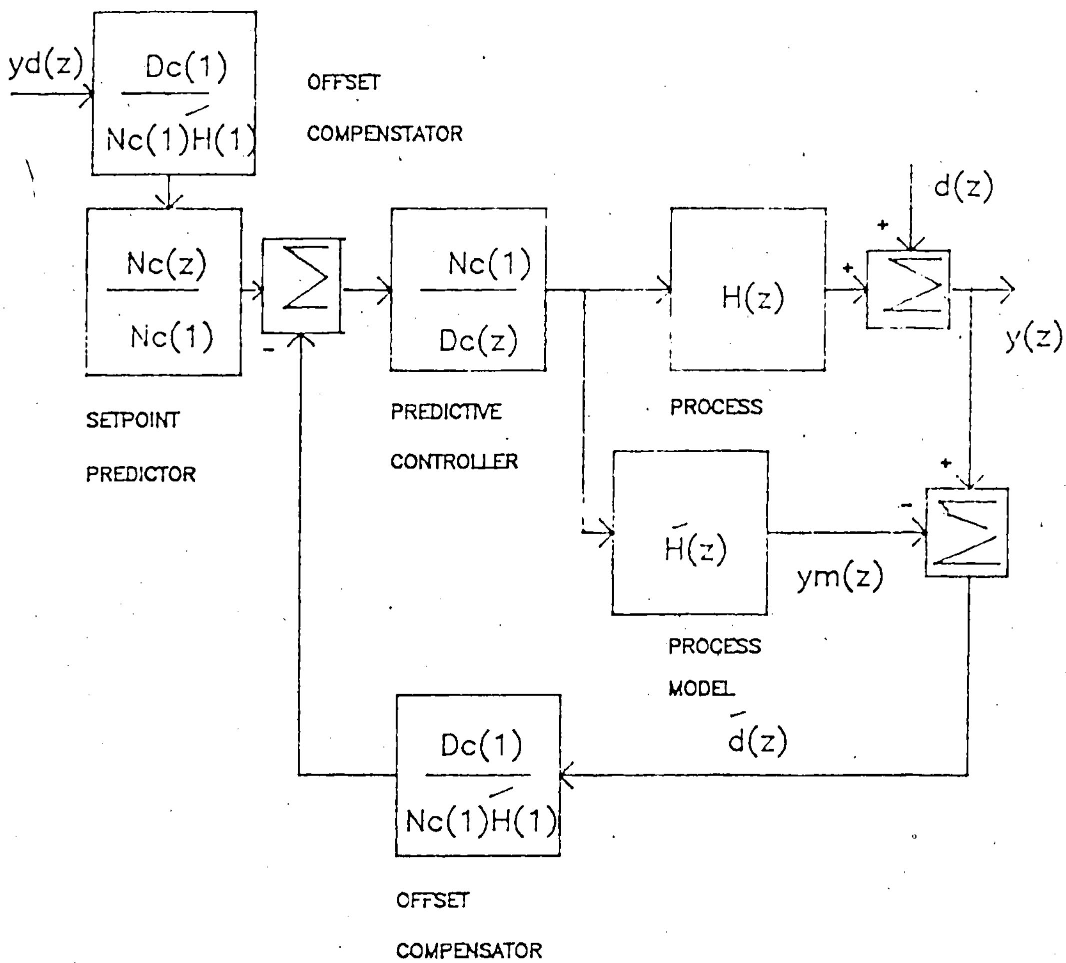

Figure 2.3 Block diagram of IMC predictive control

iv. Sampling time (T):

By making the sampling time large enough the NMP characteristics of the system model can be eliminated.

v. Optimization horizon (P):

As mentioned in the section on the input suppression parameter, any NMP process can be stabilized by choosing P sufficiently large and M sufficiently small. For the minimum phase process the length of the optimization horizon has no effect on the control performance.

2.6 Tuning the predictive controller

In this section guidelines will be presented illustrating the IMC tuning procedure. The first step in this procedure is to build a mathematical model of the process. In most cases sufficient information about the process can be derived from analysis of the open loop response to a step input. Although the true process may be nonlinear and of high order, it is common to approximate the process by a linear first or second order function with dead time. Tuning guidelines will be presented for a minimum phase process, a nonminimum phase process and an unstable process.

i. Tuning the IMC controller when the transfer function is minimum phase:

In this example the process is described by the analog transfer function

$$y(s)=.1e^{-4s}u(s)/[(s+.1)(s+1)] \qquad 2.6.1$$

The open loop response to a step in input is shown in Figure 2.4. A minimum phase process has all poles and zeros inside the unit circle. If equation 2.3.3 is used to calculate $Gc(z)$ (where $\hat{G}(z)=G(z)$), it has been demonstrated this controller will be stable (see Appendix C). From the open loop response shown in Figure 2.4, qualitatively it is reasonable to approximate this process by a second order transfer function and a time delay of four minutes. For the minimum phase case, the sampling rate has little effect on control performance but it is convenient to choose a sampling rate that gives an integer number of time delays. For many recursive or batch identification routines[9], this is all the information the routine requires to find an autoregressive moving average(ARMA) model of the process. In this example the sampling rate was chosen as 4 minutes and the parameters of the second order transfer function determined using a recursive identification routine[10]. The resulting discrete process model and the continuous process model are given in Appendix E. If an identification algorithm is not avaliable, simple graphical analysis[11] should provide a reasonable second order ARMA model. The ARMA model is converted to an impulse response model by dividing the model's denominator into the numerator.

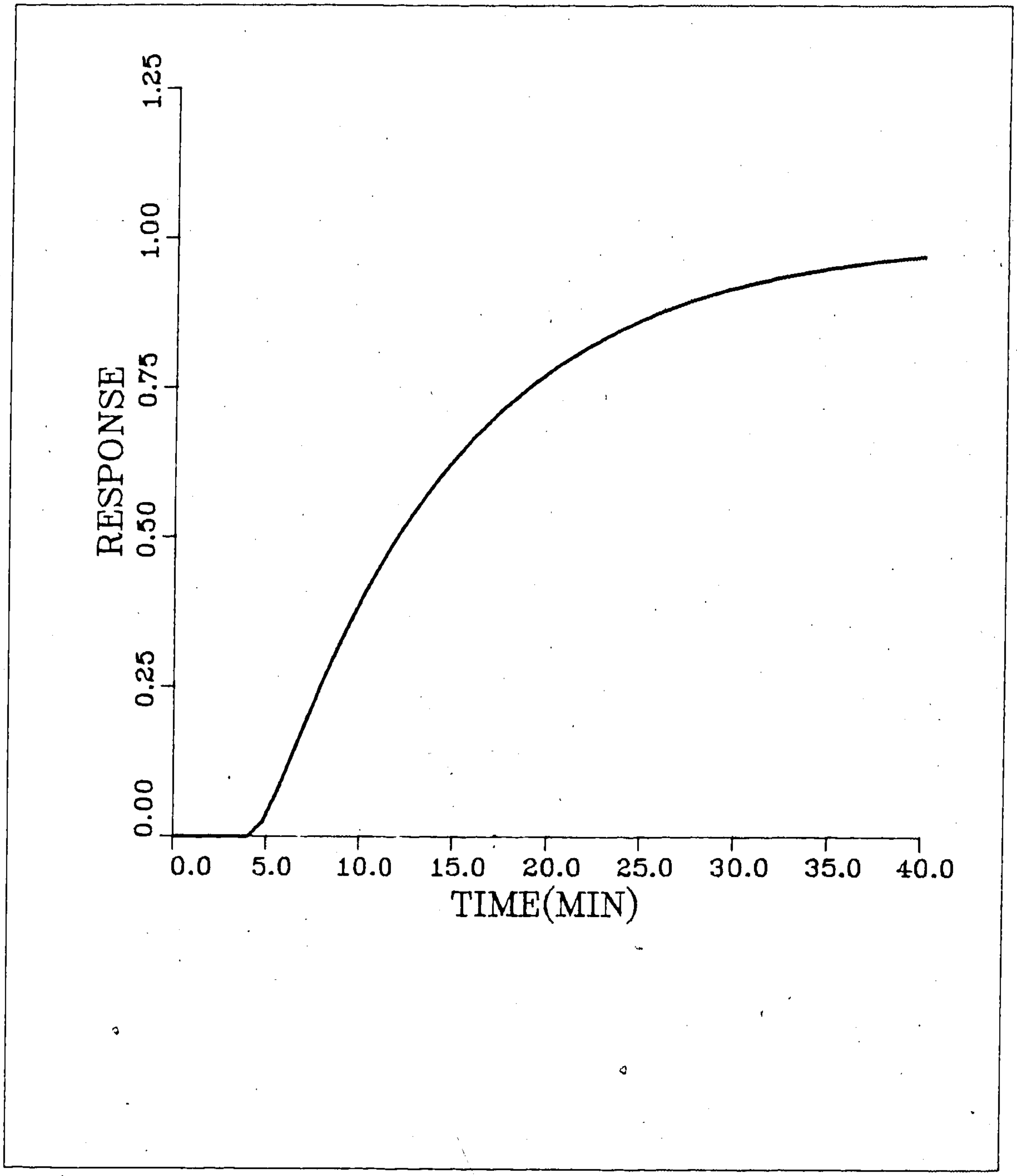

Figure 2.4 Open loop response for a minimum phase system

Since the process is minimum phase, the perfect controller or what is commonly called a deadbeat controller, is realizable. To achieve the perfect controller, the tuning parameters are as follows

$P=N=M=10$, $\beta_j=0$ for $j=1,M$, $\gamma_j=1$ for $j=1,P$.

These parameters plus the process model parameters are inputs to the IMC algorithm. For this and subsequent illustrations, FORTRAN programs are developed that use linear and nonlinear differential equations to simulate the processes. An example that illustrates the operation of these programs, is provided with the FORTRAN code in Appendix O. The response of the closed loop system to a unit step in setpoint is plotted in Figure 2.5. Empirical measures of the control performance are given for both the control and manipulated variables. Output Error is the integral of the absolute error between process output and setpoint. Control effort is the integral of the absolute difference between the control variable at sucsessive sampling intervals. Although the response is stable, oscillations in process output and control action occur. Also note the bounce in the output and subsequently the input after N sampling intervals due to truncation error in the impulse response series. In Figure 2.6 the number of terms in the impulse response series has been increased to 15. The truncation error in this figure has been reduced and postponed another 5

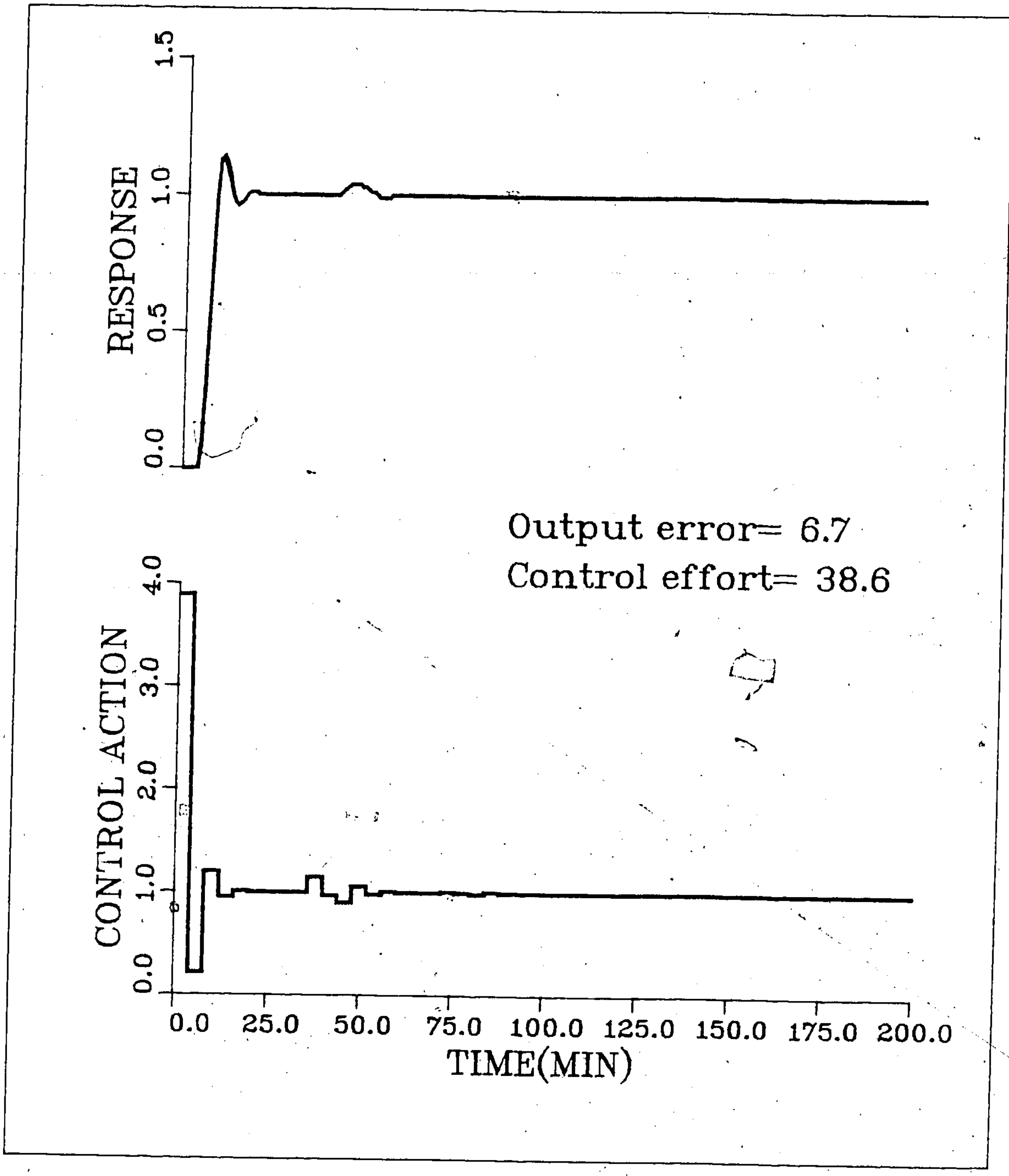

Figure 2.5 Response of the MP system to a positive setpoint change under deadbeat IMC control

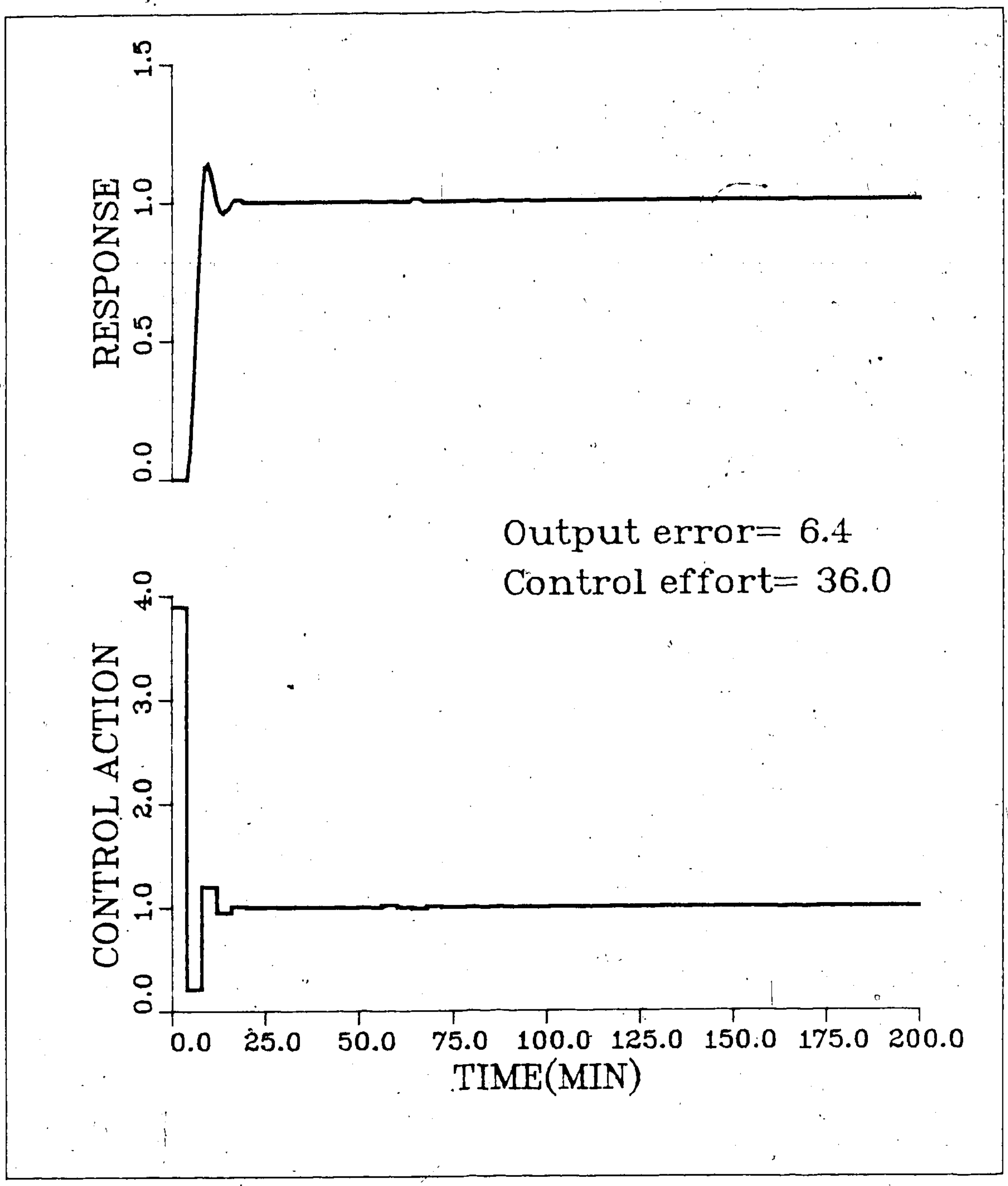

Figure 2.6 Response of the minimum phase system to a unit step change in setpoint under IMC deadbeat control, N=15

sample steps or equivalently 20 minutes. By reducing the input suppression parameter, M, to 3 and penalizing control inputs for 3 steps the excursions of the manipulated variable are reduced, as shown in Figure 2.7. That is

$N=P=10;\ M=3;\ \beta_j=3$ for $j=1,M;\ \gamma_j=1$ for $j=1,P$

The value of 3 for β_j is chosen arbitrarily. In general a larger β_j will cause a more sluggish response and reduce the impact of truncation error on the output response because the magnitude of the initial control step is reduced. In Appendix F it is demonstrated, that for an nth order process, the time series of inputs cannot drive process states to the setpoint until n time intervals have occurred. The second set of tuning parameters are chosen for this reason. The inputs to the process are allowed to vary for n+1 time intervals but after are held constant. The second set of tuning parameters have sacrificed setpoint tracking to provide better control dynamics, they also make the closed loop system more robust since the closed loop poles are shifted closer to the origin.

ii. Tuning the IMC controller when the process has NMP behavior:

The NMP process is described by

$$y(s)=-1e^{-8s}(s-.1)u(s)/[(s+.1)(s+1)] \qquad 2.6.2$$

The open loop response to a step in input is shown in Figure 2.8. The continuous process has a zero in the

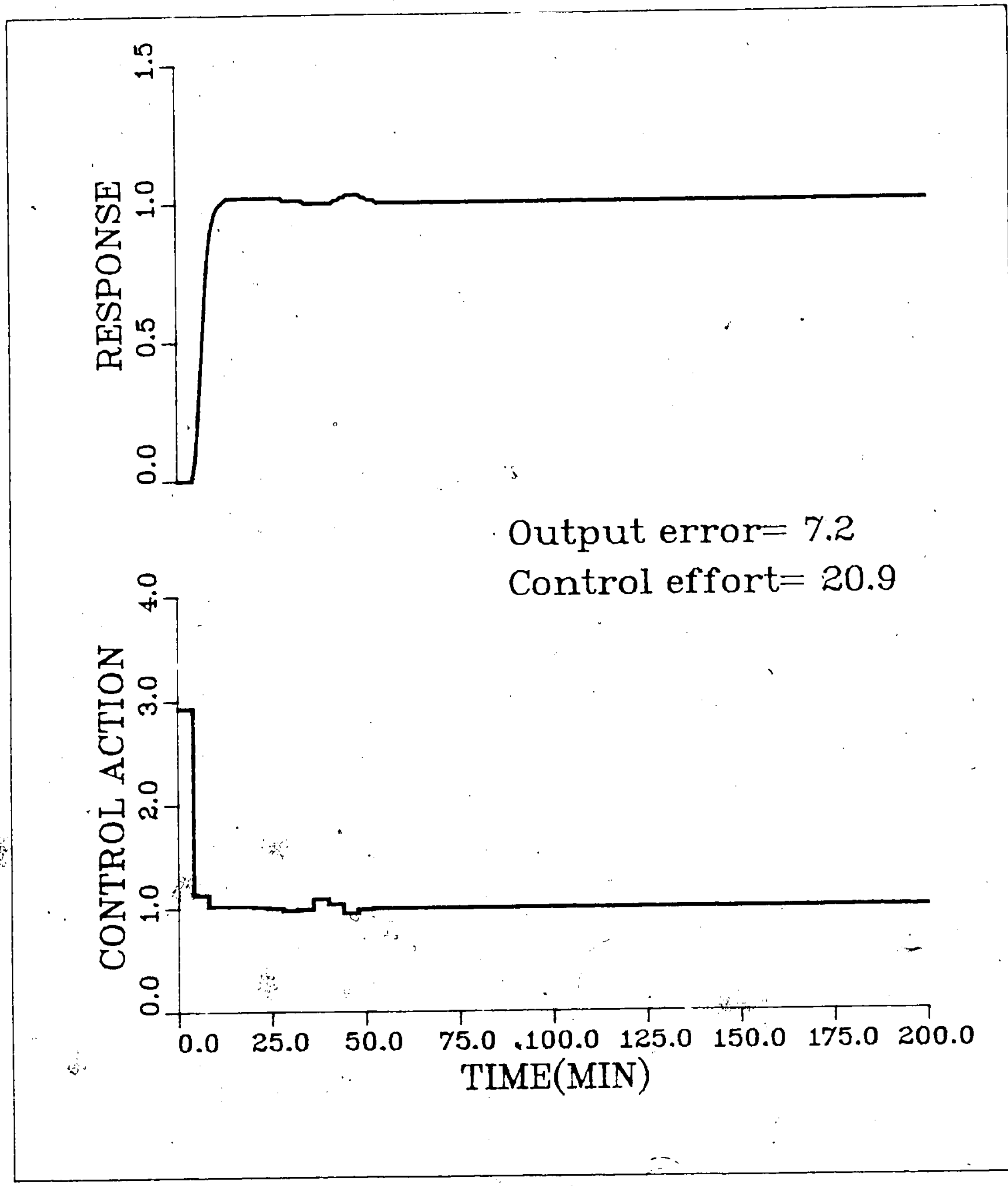

Figure 2.7 Response of the MP system to a positive setpoint change under IMC control

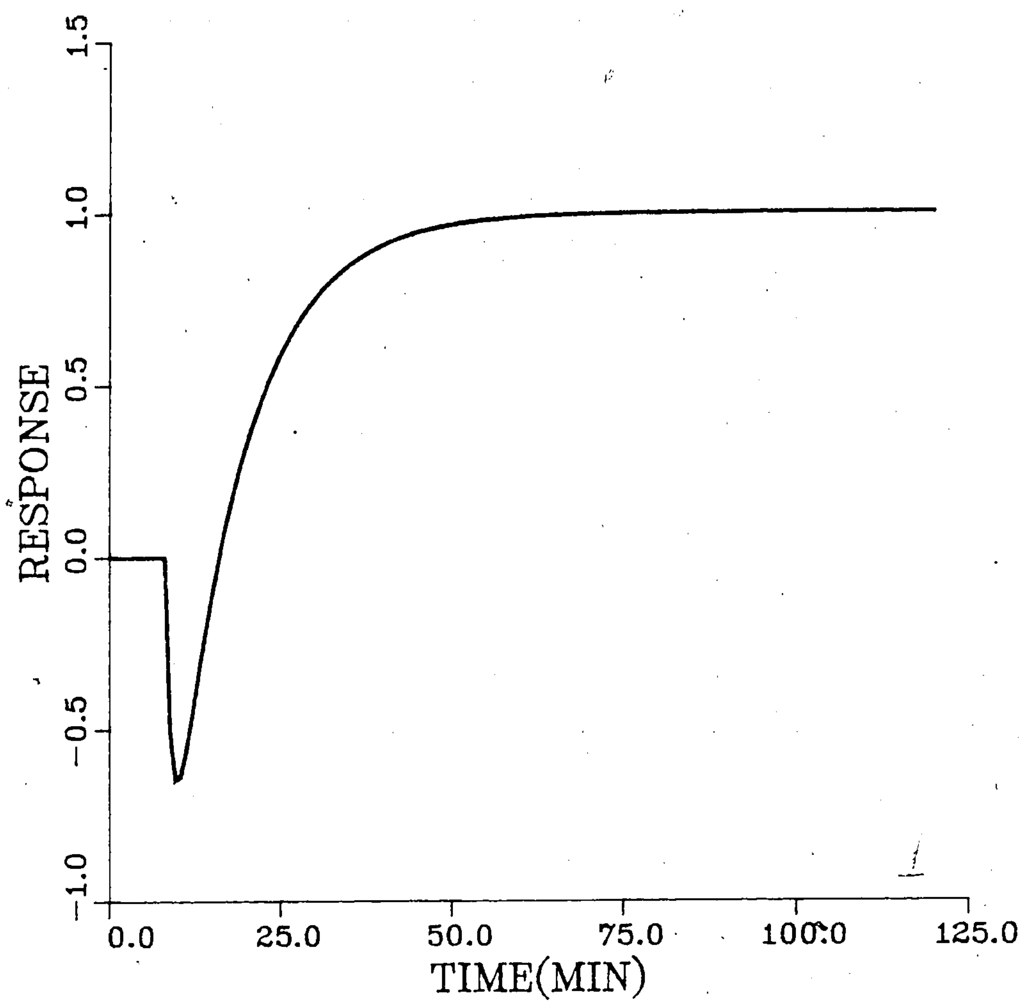

Figure 2.8 Open loop response for a nonminimum phase system

right half plane of the Laplace domain. For the controller to be stable this zero must be compensated. Choosing a sampling interval that avoids the NMP behavior is one method of achieving this end. Factorizations of the NMP process zero often leads to undesirable oscillation in the closed loop response. Alternatively, a stable controller will result if sufficient penalties are applied to the control law formulation. In Figure 2.9 the discrete open loop response for the NMP continuous process sampled at eight minutes is shown. Since the discrete process model is minimum phase, the closed loop system can be tuned as if the process were minimum phase.

In Figure 2.10 the discrete open loop response for the process sampled at four minutes is illustrated. The sampling interval was chosen to give an integer number of time delays. The discrete model is still nonminimum phase at this sampling rate. Figure 2.11 shows the closed loop response for the well tuned NMP process sampled at four minutes. The controller parameters for this response are

$T=4$ min $P=N=10$ $M=3$ $\beta_j=0$ for $j=1,M$ $\gamma_j=1$ for $j=1,P$

The value for the input suppression parameter, M, was chosen arbitrarily. Decreasing, M, reduces excursions of both the manipulated and controlled variables and increases the closed loop stability.

iii. Tuning the IMC controller when the process is unstable:

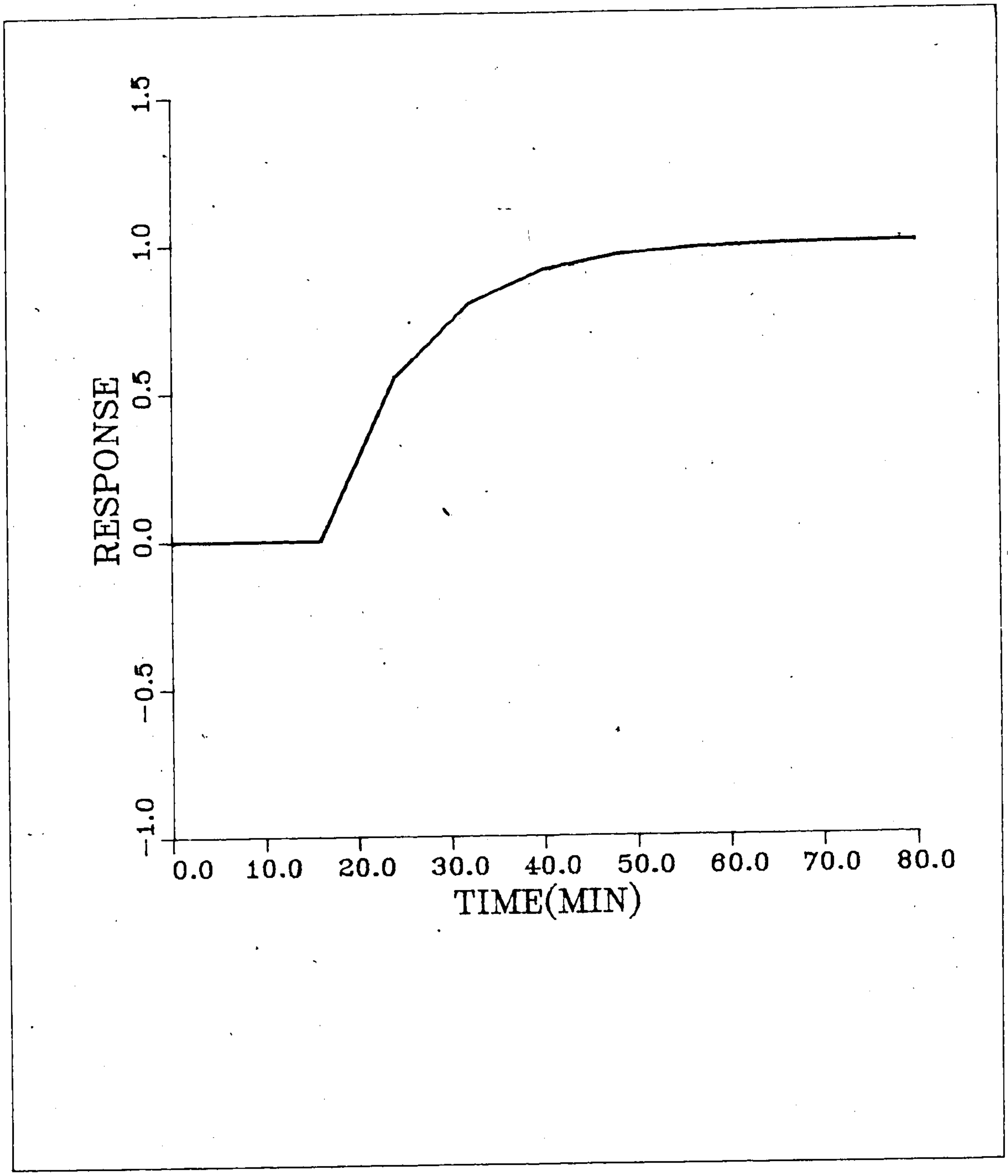

Figure 2.9 Discrete open loop response for the continuous NMP system, sample interval of eight minutes

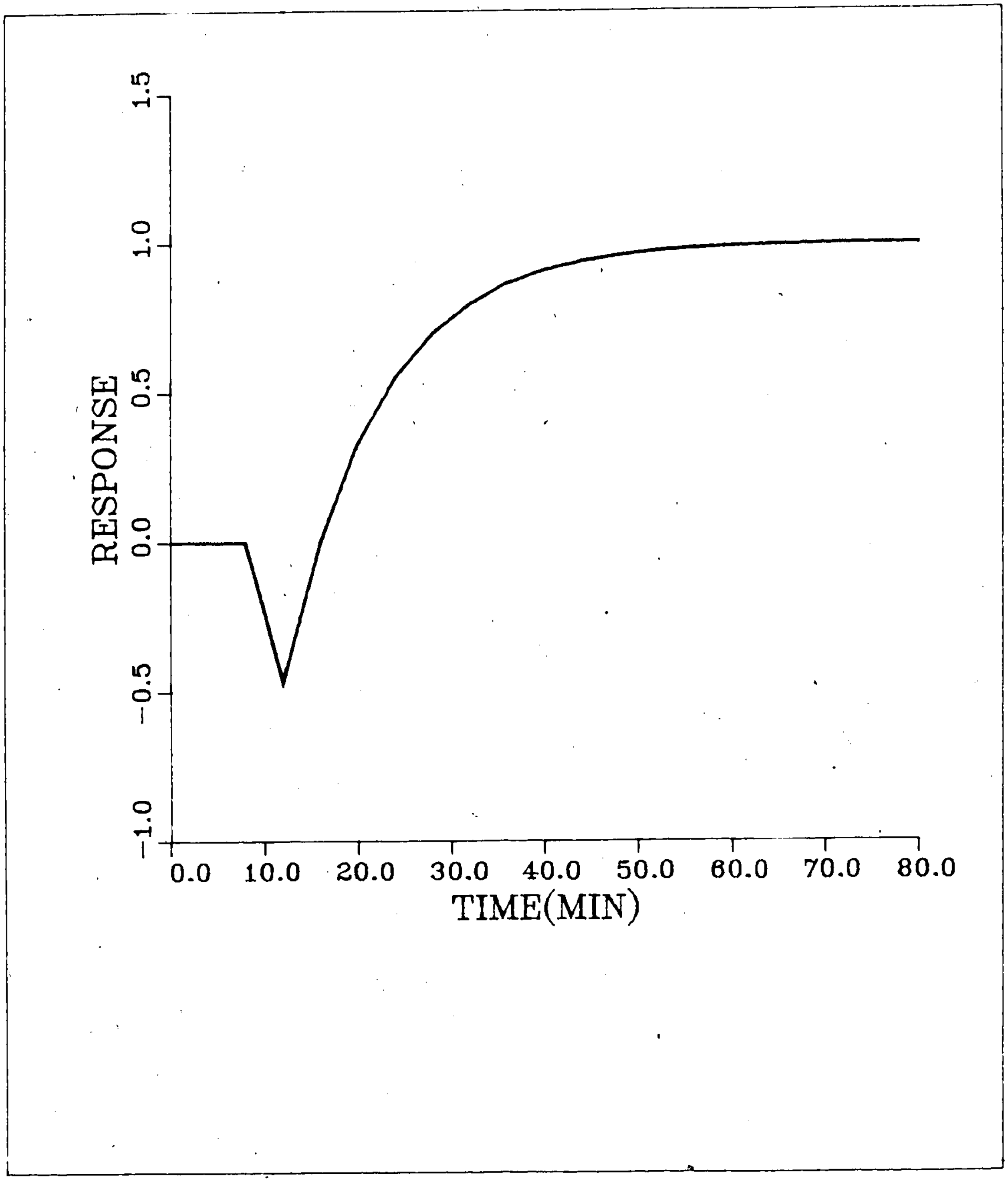

Figure 2.10 Discrete open loop response for the continuous NMP system, sample interval of four minutes

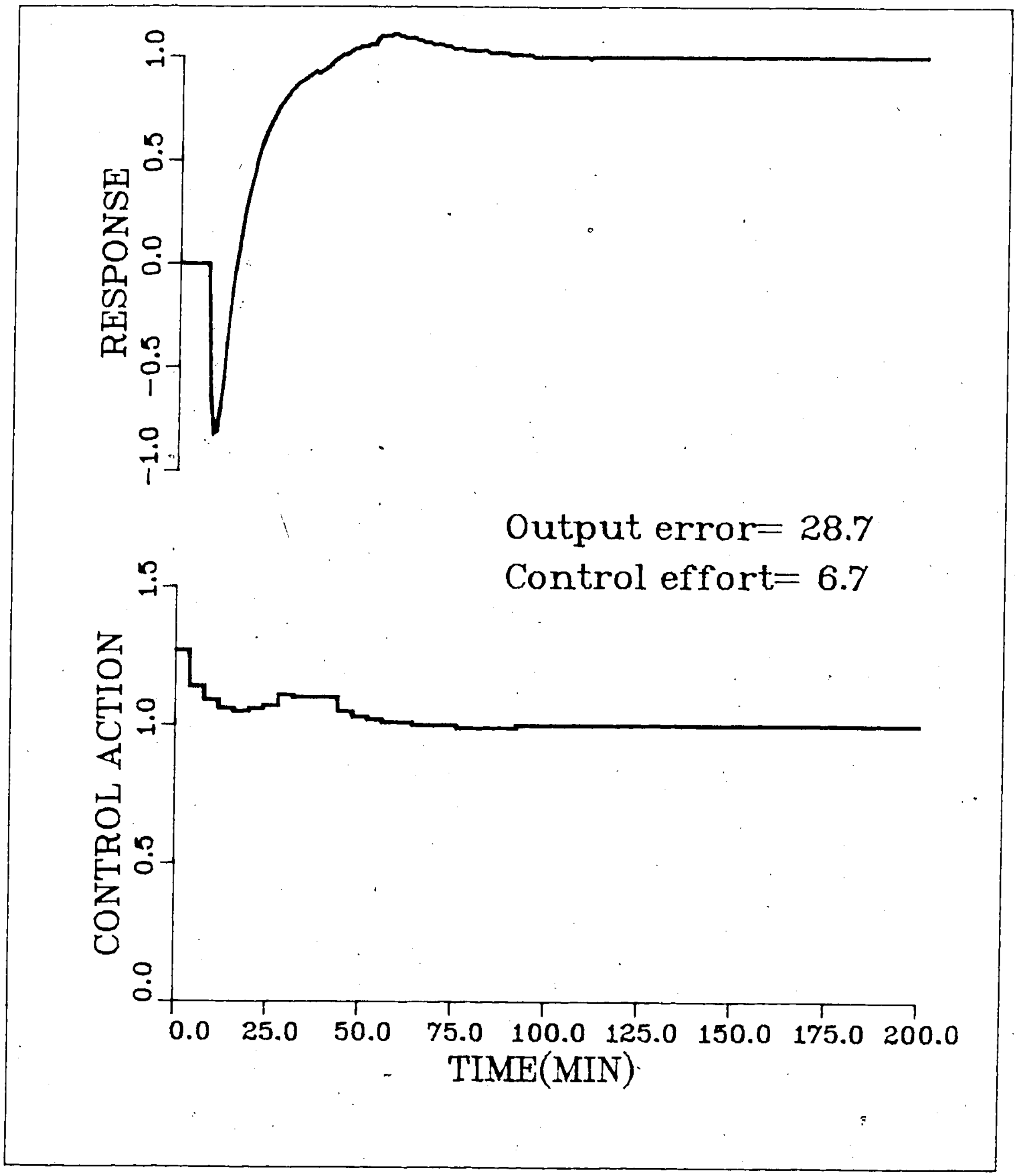

Figure 2.11 Response of the NMP system to a positive setpoint change under IMC control

The unstable process is illustrated in Figure 2.12 and described by

$$y(s)=.2e^{-4s}u(s)/[(s-.1)(s+2)] \qquad 2.6.3$$

For the closed loop response to be stable, all the roots of equation 2.2.5 must be inside the unit circle. If the process is characterized by poles outside the unit circle, then some other control scheme must be used to stabilize the process before IMC is implemented. One approach stabilizes the process using analog or digital proportional feedback control. Luyben[11] provides guidelines for tuning the analog controller using root locus techniques. In the root locus routine a 2nd order Pade polynomial representation of the dead time is used. A proportional gain of 2.0 was sufficient to stabilize the process. In Figure 2.13 the stabilized closed loop response to a unit step in setpoint is shown.

The analog controller is cascaded to the IMC system to optimize the sequence of control steps. The process model provided to the IMC routine, corresponds to the analog controlled, closed loop system. A block diagram of how the system links together is shown in Figure 2.14. The analog controller has stabilized the process so it can be tuned via the minimum phase tuning procedures. In Figure 2.15 the IMC-analog cascaded system response is shown for the IMC parameters of

T=4min P=N=10; M=2; β_j=20 for j=1,M; γ_j=1 for j=1,P

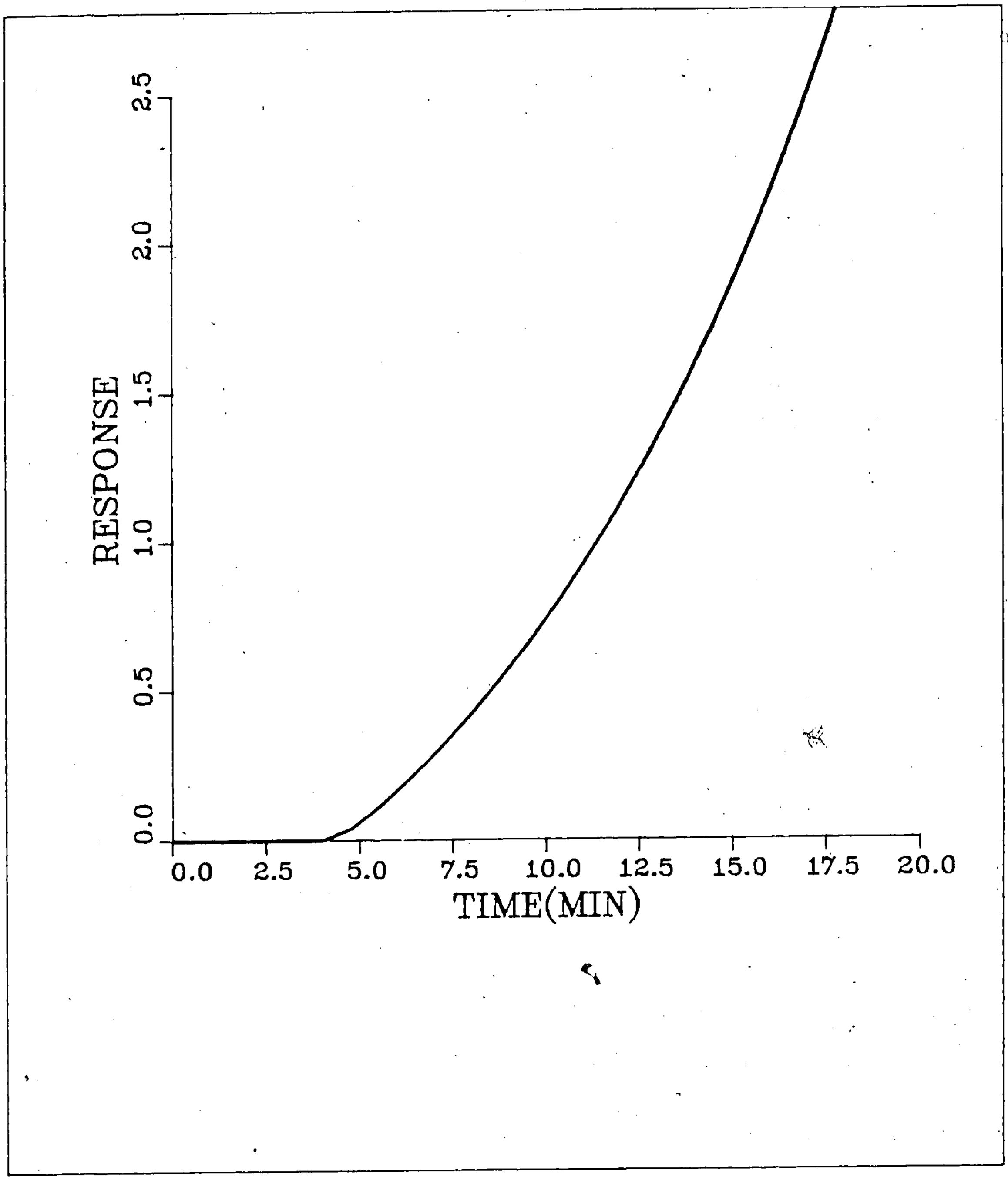

Figure 2.12 Open loop response for an unstable system

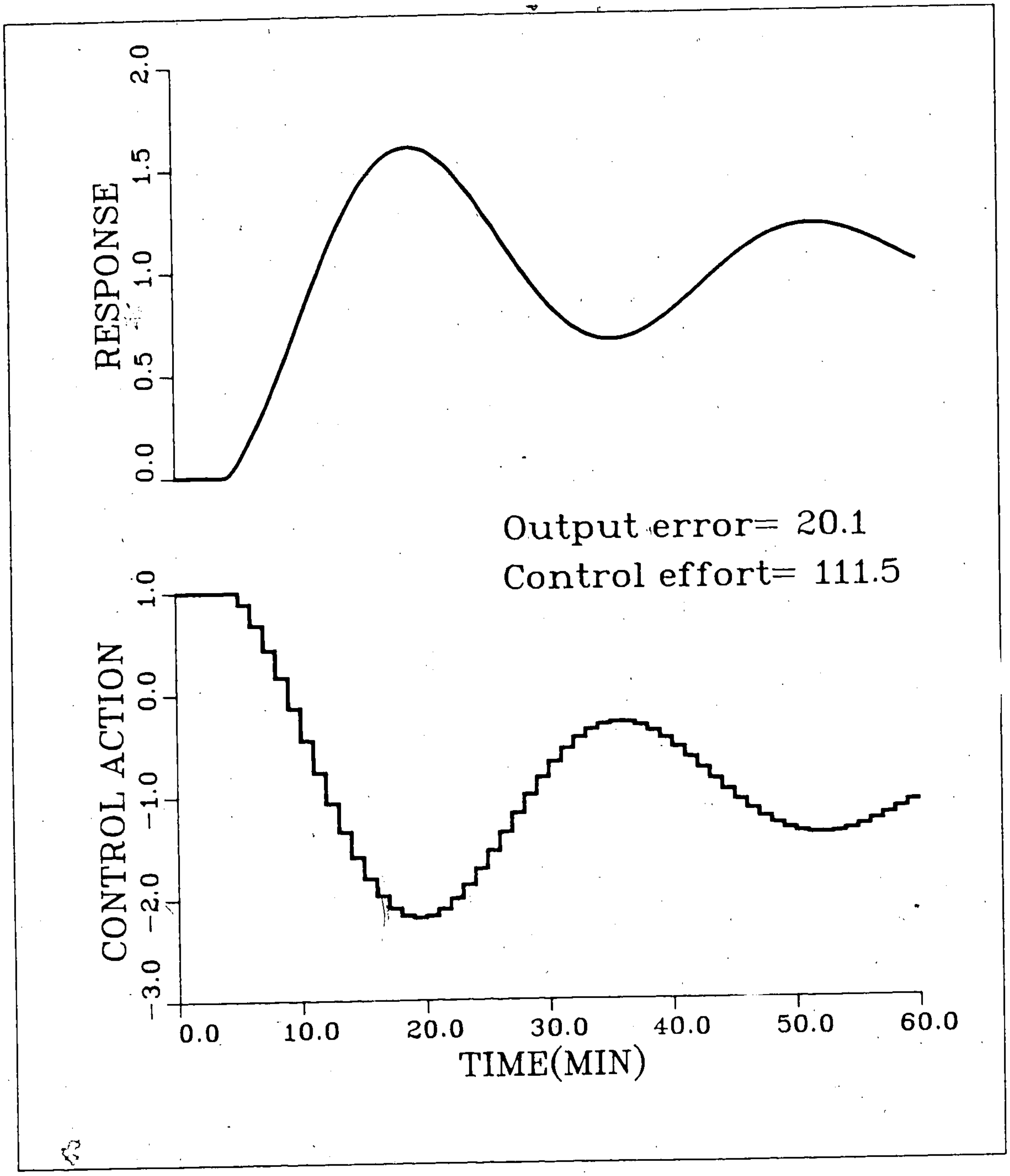

Figure 2.13 Response of the analog controlled system to a positive change in setpoint

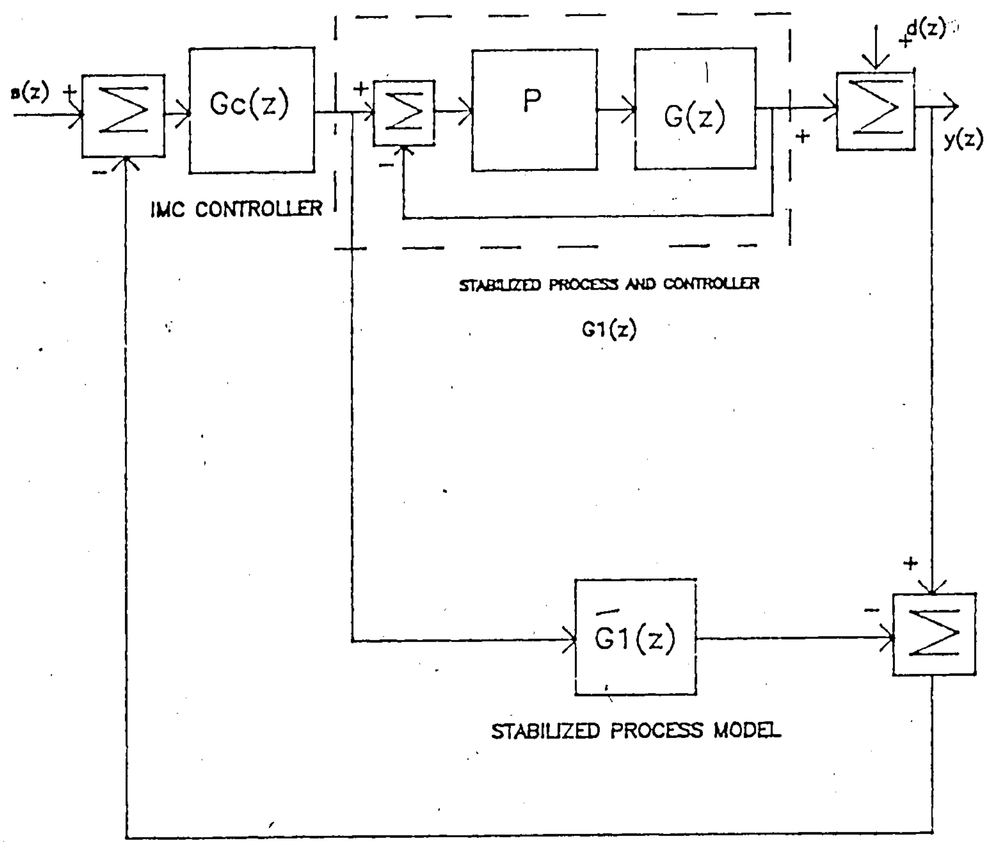

Figure 2.14 Block diagram of hybrid IMC-proportional control

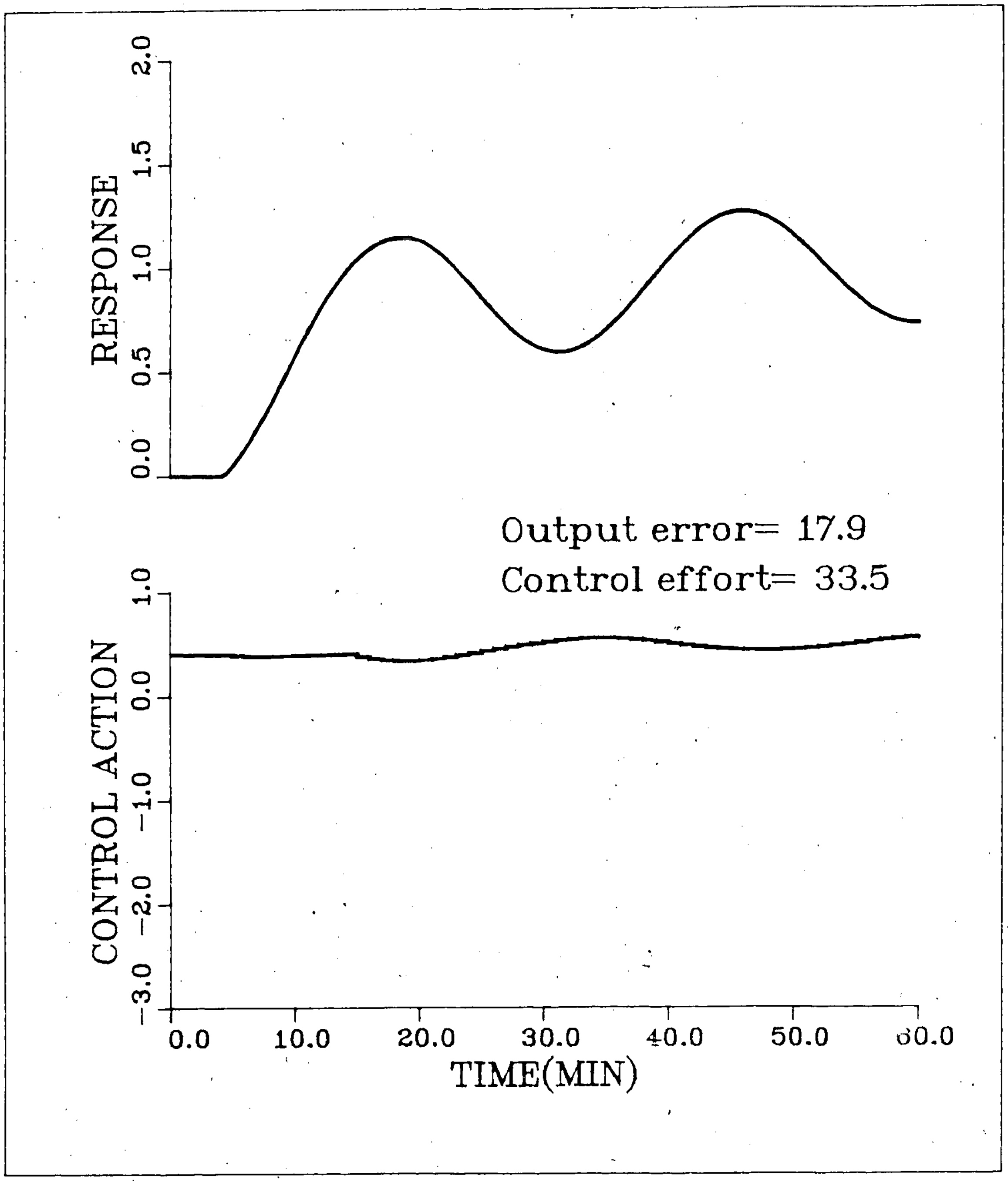

Figure 2.15 Response of the unstable system to a positive setpoint change under hybrid IMC-analog control

These controller parameter choices heavily penalize any excursions of the manipulated variable. The large value of β_j and small value for M, are indicative of a marginally stable process.

2.7 The Filter, F(z)

The final form of the control law includes an exponential filter in the feedforward path as shown in Figure 2.16. An exponential filter is included in the work of Richalet, Rault, Testud and Papon[12]. Development of this control theory and how it relates to IMC is included in the chapter devoted to the literature review, but for the moment it is sufficient to state the filter structure and to explain how it effects the response dynamics.

$$F(z)=\frac{1-\alpha}{1-\alpha z^{-1}} \qquad 0<\alpha<1$$

The closed loop response dynamics improve since the filter effects the speed at which the output reacts to changes in setpoint. To illustrate this point, consider the closed loop transfer function for y(z) when the process model is correct and for no disturbances, that is $\hat{G}(z)=G(z)$ and $d(z)=0$

$$y(z)=\frac{(1-\alpha\ \ )}{1-\alpha z^{-1}}(s(z)-(y(z)-y_m(z)))$$

$$(1-\alpha z^{-1})y(z)=(1-\alpha)s(z)$$

Expressing y(z) as a difference equation in terms of k, it follows that

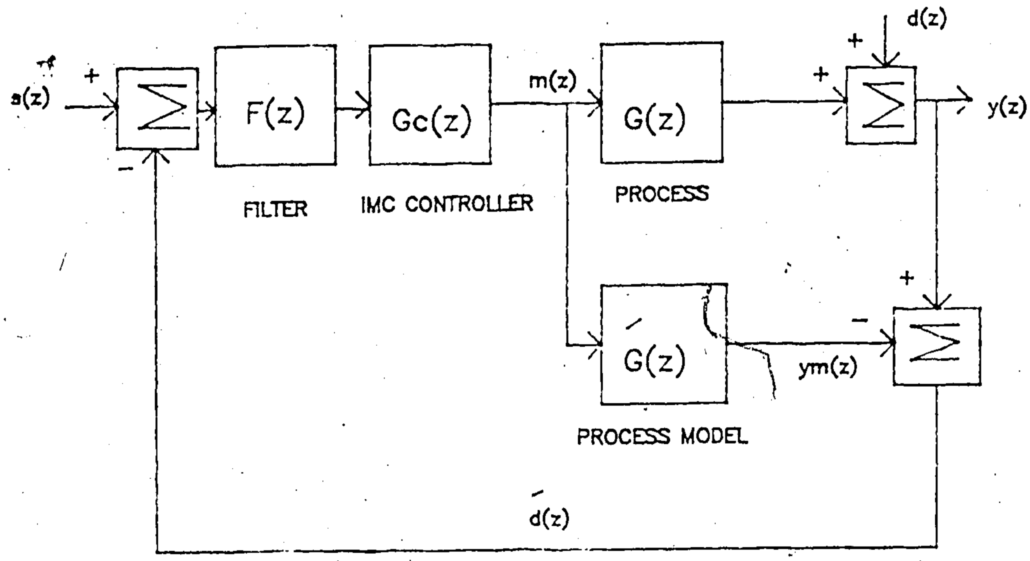

Figure 2.16 Block diagram illustrating the position of the filter

$$y(k)=\alpha y(k-1)+(1-\alpha)s(k)$$

As α is increased from zero to one, the response of the process to a change in setpoint becomes increasingly sluggish. The effect of α can be related to conventional controller design procedures, increasing α improves the robustness of the IMC controller, decreasing the gain improves the robustness of the conventional controller. In both control schemes the changes essentially detune the respective controllers. Detuning in many examples will improve the output error and controller effort measures because it reduces undesirable oscillatory responses.

3. INTERNAL MODEL CONTROL FOR MIMO SYSTEMS

3.1 Introduction

The concept of SISO internal model control is easily extended to embrace MIMO control applications[2,3]. In the single variable case the process output was related to the process input by equation 2.2.1

$$y(z) = G(z)m(z) + d(z) \qquad 2.2.1$$

This equation can also describe the multivariable input output relationship if the notation is embellished to differentiate between scalar, vector and matrix quantities. By denoting vectors with a single underscore and matrices with a double underscore, equation 2.2.1 written for a multivariable system is

$$\underline{y}(z) = \underline{\underline{G}}(z)\underline{m}(z) + \underline{d}(z) \qquad 3.1.1$$

Because the structure of IMC suggests that the best controller for the multivariable process is a realizable inverse of the process transfer function matrix $\underline{\underline{G}}(z)$, the design procedure for the MIMO system is very similar to that for the SISO system.

Time delay compensation techniques for multivariable control applications require greater sophistication than their single variable counterparts. The multivariable self-tuning controller[13] only compensates for time delays on the diagonal elements of the transfer function matrix. Ogunnaike and Ray[14] have proposed a MIMO time delay compensator, that is essentially a multivariable Smith predictor. Garcia and Morari[2] depart from these, and other similar time delay compensation techniques by proposing a factorization of the time delays, from $\acute{\underline{G}}(z)$ using a diagonal precompensator.

The single variable filter briefly discussed in Section 2.7, plays a more important role in the multivariable system. The multivariable filter is a diagonal matrix of SISO filters. The weighting parameter α_i, becomes the most convenient tuning parameter to eliminate undesirable control dynamics and to compensate for plant/model mismatch. In essence the filter may be thought of as a robustness knob. Design and use of the filter, like the time delay compensator, are developed in this chapter. The MIMO discussion concludes with several examples illustrating the multivariable IMC tuning procedure.

3.2 The Optimal Time Delay Compensator

Due to the nature of the IMC design policy the MIMO process transfer function matrix should be open loop stable. If an unstable transfer function exists it must be stabilized, using conventional control as in Section 2.6, before IMC is implemented.

As in the SISO case, the MIMO design requires that the controller $\underline{G}_c(z)$ be based upon a factorization of $\acute{\underline{G}}(z)$

$$\underline{G}_c(z) = \underline{G}_-(z)^{-1} \qquad 3.2.1$$

where

$$\acute{\underline{G}}(z) = \acute{\underline{G}}_{+1}(z)\acute{\underline{G}}_{+2}(z)\underline{G}_-(z) \qquad 3.2.2$$

with $\acute{\underline{G}}_{+1}(z)$ selected to make 3.2.1 realizable, so it may contain time delays and can contain NMP characteristics. If the time delays are not compensated the controller, $G\underline{c}(z)$ will be noncausal. If the NMP characteristics are left in $\acute{\underline{G}}_{-}(z)$ the inverse will be unstable due to zeros outside the unit circle in the process model. The design procedure also requires that $\acute{\underline{G}}_{+2}(z)$ be selected to make $\acute{\underline{G}}_{-}(z)^{-1}\acute{\underline{G}}_{+2}(z)$ stable. To avoid the additional factorization necessary to find $\acute{\underline{G}}_{+2}(z)$, the predictive least squares control law with weighting matrices, is used to find a stable approximation of $\acute{\underline{G}}_{-}(z)$. Additional stability/robustness is provided by the filter $\underline{F}(z)$.

The time delay compensator, $\acute{\underline{G}}_{+1}(z)$, is restricted to a diagonal matrix, this implies the compensator will be optimal in terms of settling time if the minimum delay occurs on the diagonal elements of $\acute{\underline{G}}(z)$. Inclusion of off-diagonal elements in $\acute{\underline{G}}_{+1}(z)$ would cause the formulation to become unnecessarily complicated. The optimal diagonal factorization is found by the relation

$$\tau_j^* = \max_j(\tau_{ij}) \qquad 3.2.3$$

where τ_{ij} are the units of delay in the 'ij'th element of the matrix $\acute{\underline{G}}(z)^{-1}$. The variable τ_j^* will become the j th diagonal element of the inverse time delay compensator matrix $\acute{\underline{G}}_{+1}(z)^{-1}$. To clarify how equation 3.2.3 is used to find $\acute{\underline{G}}_{+1}(z)$ an example is presented.

Example 3.2.1 Designing the time delay compensator for a minimum phase transfer function.

In this example a two by two transfer function describes the input-output relationship

$$\underline{G}(z) = \frac{1.0}{(1-0.5z^{-1})}\begin{vmatrix} 2z^{-3} & z^{-5} \\ z^{-2} & z^{-3} \end{vmatrix} \qquad 3.2.4$$

The inverse transfer function is given by

$$\underline{G}(z)^{-1} = 0.5\begin{vmatrix} z^{3} & -z \\ -z^{4} & 2z^{3} \end{vmatrix} \qquad 3.2.5$$

Using equation 3.2.3 the inverse precompensator can be determined

$$\underline{G}_{+1}(z)^{-1} = \begin{vmatrix} z^{3} & 0 \\ 0 & z^{4} \end{vmatrix} \qquad 3.2.6$$

After taking the inverse of equation 3.2.6 the precompensator is found

$$\underline{G}_{+1}(z) = \begin{vmatrix} z^{-4} & 0 \\ 0 & z^{-3} \end{vmatrix} \qquad 3.2.7$$

In example 3.2.1 the minimum time delay occurs at off-diagonal element 2,1 thus the diagonal factorization is not optimal in terms of settling or response time. For the factorization to be optimal it would be necessary to include terms in the off-diagonal elements of the precompensator matrix. This consequence will be discussed later in this chapter.

The diagonal precompensator or large sampling interval, can be used to remove any NMP characteristics from the process model. An example of how $\underline{G}_{+1}(z)$ can be used to remove NMP behavior from $\underline{\acute{G}}_{-}(z)$ is demonstrated in Appendix G.

3.3 The multivariable predictive control problem

The basic IMC, multivariable closed loop structure is illustrated in Figure 3.1. From this diagram the closed loop transfer function can be established. The control signal vector $\underline{m}(z)$ is determined from

$$\underline{m}(z)=[\underline{I}+G\underline{c}(z)\underline{F}(z)(\underline{G}(z)-\underline{\acute{G}}(z))]^{-1}G\underline{c}(z)\underline{F}(z)(\underline{s}(z)-\underline{d}(z)) \quad 3.3.1$$

and the process output vector $\underline{y}(z)$ from

$$\underline{y}(z)=\underline{G}(z)[\underline{I}+G\underline{c}(z)\underline{F}(z)(\underline{G}(z)-\underline{\acute{G}}(z))]^{-1}G\underline{c}(z)\underline{F}(z)(\underline{s}(z)-\underline{d}(z))+\underline{d}(z) \quad 3.3.2$$

The next section will look at one specific choice for $G\underline{c}(z)$, the completely decoupling controller. The single variable analogy to this multivariable controller is the deadbeat or perfect controller.

3.3.1 The decoupling controller

After $\underline{\acute{G}}_{+1}(z)$ is factored from $\underline{\acute{G}}(z)$, it should be possible to find the controller $G\underline{c}(z)$, given by equation 3.2.1 which results in a completely decoupled response. This is demonstrated in the following example.

Example 3.3.1 The decoupling controller

In this example the IMC controller is designed on the

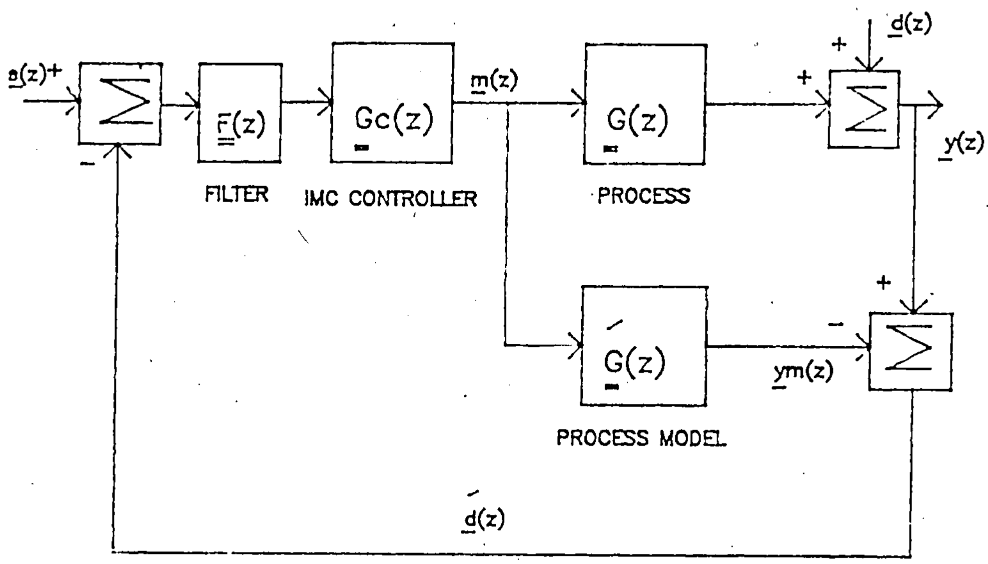

Figure 3.1 Block diagram of basic multivariable IMC structure

basis of equation 3.2.1. In addition the process model, $\underline{\acute{G}}(z)$ is a correct description of the true process and the filter is set to unity. That is $\underline{\acute{G}}(z)=\underline{G}(z)$, $\underline{F}(z)=\underline{I}$. Given these conditions the closed loop input output relationship, based on equation 3.3.2, can be developed in the following manner

Because $\underline{\acute{G}}(z)-\underline{G}(z)=0$

$$\underline{y}(z)=\underline{G}(z)G\underline{c}(z)(\underline{s}(z)-\underline{d}(z)) + \underline{d}(z) \qquad 3.3.3$$

Because $G\underline{c}(z)=\underline{\acute{G}}_{-}(z)^{-1}$, equation 3.3.3 can be further reduced to

$$\underline{y}(z)=\underline{\acute{G}}_{+1}(z)(\underline{s}(z)-\underline{d}(z)) + \underline{d}(z) \qquad 3.3.4$$

The input-output relationship,in equation 3.3.4, will remain stable for bounded $\underline{s}(z)$ and $\underline{d}(z)$. Furthermore, as a consequence of the choice for the controller, the input-output relationship has become completely decoupled.

In practice the controller specified by equation 3.2.1 will be undesirable for several reasons. First, because this controller drives the process very hard, it is sensitive to modelling errors. Secondly process zeros on or near the unit circle will cause undesirable oscillatory behavior. Finally because this controller minimizes the process output without considering the other process states, severe intersample ripple occurs. This will always be the case when the process model order is second or higher.

The problems associated with the controller given by equation 3.2.1 also occurred in the development of the SISO

IMC controller. Analogous to the SISO case the IMC design procedure for MIMO systems introduces the multivariable predictive control law to compensate for problems in the present design policy. As demonstrated in Appendix H, one special case of the predictive control law, results in a decoupling controller, but in general the resulting controller will only approximate the completely decoupled controller. How closely the controller found by the predictive control law resembles the total decoupling controller is influenced by the choice of tuning parameters and time delay compensator. Before examining the predictive control law, the model used to describe the process must be redefined. The multivariable input-output relationship is given by

$$\underline{y}(z)=\underline{\acute{G}}_{+,1}(z)\underline{\acute{G}}_{-}(z)\underline{m}(z)+\underline{d}(z) \qquad 3.3.5$$

Let $\underline{y}^{+}(z)$ denote the output from the time delay compensated process, that is

$$\underline{y}^{+}(z)=\underline{\acute{G}}_{-}(z)\underline{m}(z)+\underline{d}(z) \qquad 3.3.6$$

The process description is further modified by replacing the transfer function matrix $\underline{\acute{G}}_{-}(z)$ with the impulse response matrix $\underline{H}(z)$ to give

$$\underline{y}^{+}(z)=\underline{H}(z)\underline{m}(z)+\underline{d}(z) \qquad 3.3.7$$

Garcia and Morari[1] hypothesize because the impulse response model is a nonparametric or nonparsimonuous description of the process, it contains more information about possible nonlinearities in underlying process, than an ARMA representation. This possibility will be discussed in

Chapter 4. For the moment it is important to note, this representation, unlike an ARMA model, facilitates prediction of future outputs based entirely on the time series of inputs, eliminating the need to store past output history.

3.3.2 "Balance" in the context of the multivariable transfer function

It has been mentioned that $\underline{\acute{G}}_{+1}(z)$ is only optimal if the minimum time delay occurs on the diagonal elements of the transfer function matrix $\underline{\acute{G}}(z)$. A system that meets this requirement is denoted by Garcia and Morari as 'balanced'. Conversely, a system that fails to meet this requirement is termed 'unbalanced'. The measure of 'imbalance' is found as follows. If τ_i^+ is the delay in the ii th element of the precompensator matrix, and τ_{ij} is the minimum time delay in each row of the transfer function matrix, then the measure of imbalance τ_0 is found from

$$\tau_0 = \max_{\text{for all } i} [\tau_i^+ - \tau_{ij}] \qquad 3.3.8$$

The example that follows will help illustrate this concept of balance.

Example 3.3.2 Determining the measure of imbalance in a process transfer function.

For the same process model as used in example 3.3.1, from $\underline{\acute{G}}(z)$ it follows that the minimum time delay per row is

$\tau_1=3$, $\qquad \tau_2=2$

From $\underline{\acute{G}}_{+1}(z)$, it can be seen that the diagonal element time delays are

$\tau_1^{+}=4, \qquad \tau_2^{+}=3$

so using this information with equation 3.3.8, the measure of imbalance is calculated as

$\tau_0=\max[(4-3), (3-2)]=1$

By calculating the resulting $\underline{H}(z)$ from $\underline{\acute{G}}_-(z)$, the implication of $\tau_0 \neq 0$ is apparent. The realizable process transfer function matrix in example 3.2.1 is given by

$$\underline{\acute{G}}_-(z)=\frac{1}{1-0.5z^{-1}}\begin{vmatrix} 2z & z^{-1} \\ z & 1 \end{vmatrix}$$

After carrying out long division on each element of $\underline{\acute{G}}_-(z)$, the equivalent impulse response model is determined to be

$$\underline{H}(z)=\begin{vmatrix} 2 & 0 \\ 1 & 0 \end{vmatrix} z^{1} + \begin{vmatrix} 1 & 0 \\ .5 & 1 \end{vmatrix} z^{0} + \begin{vmatrix} .5 & 1 \\ .25 & .5 \end{vmatrix} z^{-1}+\ldots \qquad 3.3.9$$

Which for simplicity is written using the following notation

$$\underline{H}(z)=\underline{H}_{\tau_0} z^{1}+\underline{H}_0 z^{0}+\underline{H}_{-1} z^{-1}+\ldots \qquad 3.3.10$$

If the decoupling controller, given by equation 3.2.1, is used to calculate the control signal $\underline{m}(z)$, equation 3.3.6 can be rewritten to give the deadbeat control signal

$$\underline{m}(z)=\underline{H}(z)^{-1}(\underline{s}(z)-\underline{d}(z)) \qquad 3.3.11$$

where $\underline{s}(z)$ has replaced $\underline{y}^{+}(z)$. Because the matrix corresponding to τ_0 is singular, inverting this matrix is not possible. Although it is only illustrated for $\tau_0=1$, in general all matrices $\underline{H}_j$, where $j>0$ will be singular. Thus τ_0 determines the number of singular matrices in the impulse response model.

In order to obtain the completely decoupled response it is necessary to set all singular matrices to zero. If the decoupling restraint is relaxed by the use of input and output weightings, it is possible to find an approximation of $\underline{H}(z)^{-1}$ that is nonsingular.

At this point two of the four steps necessary to design the multivariable controller have been completed. These steps are

i. The process model has been factored, via equation 3.2.3, to remove time delays and, if desired, NMP characteristics.

ii. The resulting realizable transfer function $\hat{\underline{G}}_-(z)$ has been transformed into the impulse response model and singular matrices are removed if the decoupling controller is desired.

The next two steps, formulation of the predictive control law and use of the multivariable filter, will now be considered. The predictive control law is used to design a coupled multivariable controller in those cases where the decoupling controller is:

i. not realizable or $\tau_0 \neq 0$

ii. not desirable due to severe intersample ripple

It will be shown that the filter with its exponential weighting can be used not only to compensate, for some of the undesirable aspects of the decoupling controller but also add, robustness to the closed loop system.

3.3.3 The multivariable control law

Due to the similarities between the single variable and multivariable control problem the IMC multivariable predictive control law is almost identical to its SISO counterpart. The matrix of control inputs $\underline{U}(z)$ is calculated over the horizon $P+\tau_0$ to minimize the weighted error between the setpoint and the process output

$$J = \min_{\underline{m}(k)} \sum_{j=1}^{P+\tau_0} [\,||\underline{\Delta}(j)||^2 + ||\underline{\nabla}(j)||^2\,] \qquad 3.3.12$$

$\underline{m}(k)$

$\vdots$

subject to

$$\underline{\Delta}(j) = [\underline{s}^{+}(k+j-\tau_0) - \underline{y}^{+}(k+j-\tau_0|k)]\gamma_j^{t}$$

$$\underline{\nabla}(j) = \underline{m}(k+j)\beta_j^{t}$$

$$\underline{y}^{+}(k+j-\tau_0|k) = \underline{y}_m^{+}(k+j-\tau_0) + \underline{d}^{+}(k+j-\tau_0|k)$$

$$= \sum_{i=1}^{N+\tau_0} \underline{H}_i \underline{m}(k-i) + \underline{d}^{+}(k+j-\tau_0|j)$$

where

$$||x||^2 = x^t x$$

The prediction of $\underline{d}^{+}(k+j-\tau_0|j)$ is given by the difference between the process output, $\underline{y}(k)$, and the output from the process model, $\underline{y}_m(k)$. The control law that results from the solution to this minimization is given in Appendix H.

The tuning parameters T, β, γ, P and M, effect the MIMO closed loop properties in a manner analogous to the SISO closed loop properties (cf Section 2.5). The closed loop stability properties of the multivariable predictive control law are demonstrated in Appendix I.

3.4 The multivariable filter, F(z)

The multivariable, diagonal, exponential filter, $\underline{F}(z)$

$$\underline{F}(z) = \begin{vmatrix} \frac{1-\alpha_1}{1-\alpha_1 z^{-1}} & & & \\ & \frac{1-\alpha_2}{1-\alpha_2 z^{-1}} & & \\ & & \ddots & \\ & & & \frac{1-\alpha_i}{1-\alpha_i z^{-1}} \end{vmatrix}$$

can be used as a tuning parameter to shape the closed loop response of both the control and output variables[2].

The filter detunes the controller, $G\underline{c}(z)$, to dampen input fluctuations leading to a more desirable response. Like the SISO filter, described in Section 2.7, the multivariable filter weighting parameters, α_i, can vary between 0 and 1 with further detuning of the controller for larger values of α_i. Garçia and Morari[3] encourage a design procedure that utilizes only the decoupling controller, given by equation 3.2.1 and the filter. The other parameters β, γ, M, P and T are only utilized when the inverse process model is unstable.

The other aspect of the filter, is its ability to stabilize the closed loop response when plant/model mismatch occurs. Garcia and Morari demonstrated this stabilizing feature for one specialized case which is presented in Appendix J. In general the stabilizing effect of an exponential filter will not be exclusive to IMC, any control algorithm will demonstrate greater robustness, at the expense of control performance, by increasing α_i. The next section will illustrate the effect of the filter on multivariable closed loop performance.

3.5 Tuning the Multivariable Predictive Controller

In Section 2.6 the performance of IMC for different systems was studied by simulation. The three systems were represented by discrete second order linear transfer functions. In this section multivariable IMC is used to control simulated multivariable systems when a unit step in the setpoint of y_1 occurs. Like the systems in Section 2.6, the multivariable system is represented by discrete linear first and second order transfer functions. To illustrate the performance of multivariable IMC, three different systems are studied by simulation.

3.5.1 Tuning a balanced, minimum phase transfer function matrix

The model used is a simplification of that presented for a binary ethanol-water, distillation column model[15]. To represent a balanced minimum phase system, only a two by two matrix is used. From the complete distillation column model given in Appendix K, simplification is carried out to arrive at a two by two matrix

$$\begin{vmatrix} y_1(s) \\ y_2(s) \end{vmatrix} = \begin{vmatrix} \dfrac{0.66e^{-2.6s}}{(6.7s+1)} & \dfrac{-0.61e^{-3.5s}}{(8.64s+1)} \\ \dfrac{1.11e^{-6.5s}}{(3.25s+1)} & \dfrac{-2.36e^{-3.0s}}{(5.0s+1)} \end{vmatrix} \begin{vmatrix} u_1(s) \\ u_2(s) \end{vmatrix}$$

y_1=overhead ethanol mole fraction

y_2=side stream ethanol mole fraction

u_1=reflux flow rate, gpm

u_2=side stream product flow rate, gpm

The transfer function matrix is classified as "balanced" because the minimum time delays occur on the diagonal elements of the matrix. Had a transfer function description not been available correlation analysis could be carried out to identify the impulse response series from which the time delays could be determined.

To begin the controller design procedure, the time delays are factored from the transfer function matrix using equation 3.2.3. Because an impulse response model is used in the control law and yet equation 3.2.3 requires an ARMA model to represent $\underline{G}(z)$, some simplification must be made in the time delay factorization procedure to accommodate those cases where an ARMA model is not available. One solution uses only the time delays to represent the ARMA process transfer function matrix. With this simplification and a sampling rate of 0.5 minutes, the transfer function shown above is represented by

$$\underline{G}(z) \simeq \begin{vmatrix} z^{-5} & z^{-7} \\ z^{-13} & z^{-6} \end{vmatrix}$$

The time delay in element 1,1 of the matrix has been rounded down to an integer number of samples. This approximation avoids the extra algebra associated with modified z-transforms. The inverse transfer function matrix of this $\underline{G}(z)$ is

$$\underline{G}(z)^{-1} = \frac{\begin{vmatrix} z^5 & -z^4 \\ -z^{-2} & z^6 \end{vmatrix}}{1-z^{-9}}$$

Applying equation 3.2.3 reveals the inverse time delay compensator

$$\underline{G}_{+1}(z)^{-1} = \begin{vmatrix} z^5 & \\ & z^6 \end{vmatrix}$$

Taking the inverse of $\underline{G}_{+1}(z)^{-1}$ gives

$$\underline{G}_{+1}(z) = \begin{vmatrix} z^{-5} & \\ & z^{-6} \end{vmatrix}$$

Good starting choices for the IMC tuning parameters, P, M, N, β and γ, are those values that result in the decoupling controller. As demonstrated in Appendix H, a stable and realizable controller is possible for a system described by a balanced minimum phase transfer function. By choosing the above precompensator and the following tuning parameters

$P=M=1 \quad N=30 \quad \underline{B}=0 \quad \underline{\Gamma}=I \quad \alpha_i=0.0$ for all i

the decoupling controller results. The closed loop response for the decoupling controller is shown in Figure 3.2. Absolute measures of 'Control Effort' and 'Output Error' are included in the figure to quantify the control performance. These measures are computed from the following equations

$$\text{Control Effort} = \Sigma|\underline{m}(k+i) - \underline{m}(k+i-1)|T_1$$

$$\text{Output Error} = \Sigma|\underline{y}(k+i) - \underline{s}(k+i)|T_2$$

where T_1 is the time between changes in the control signal

T_2 is $0.2T_1$

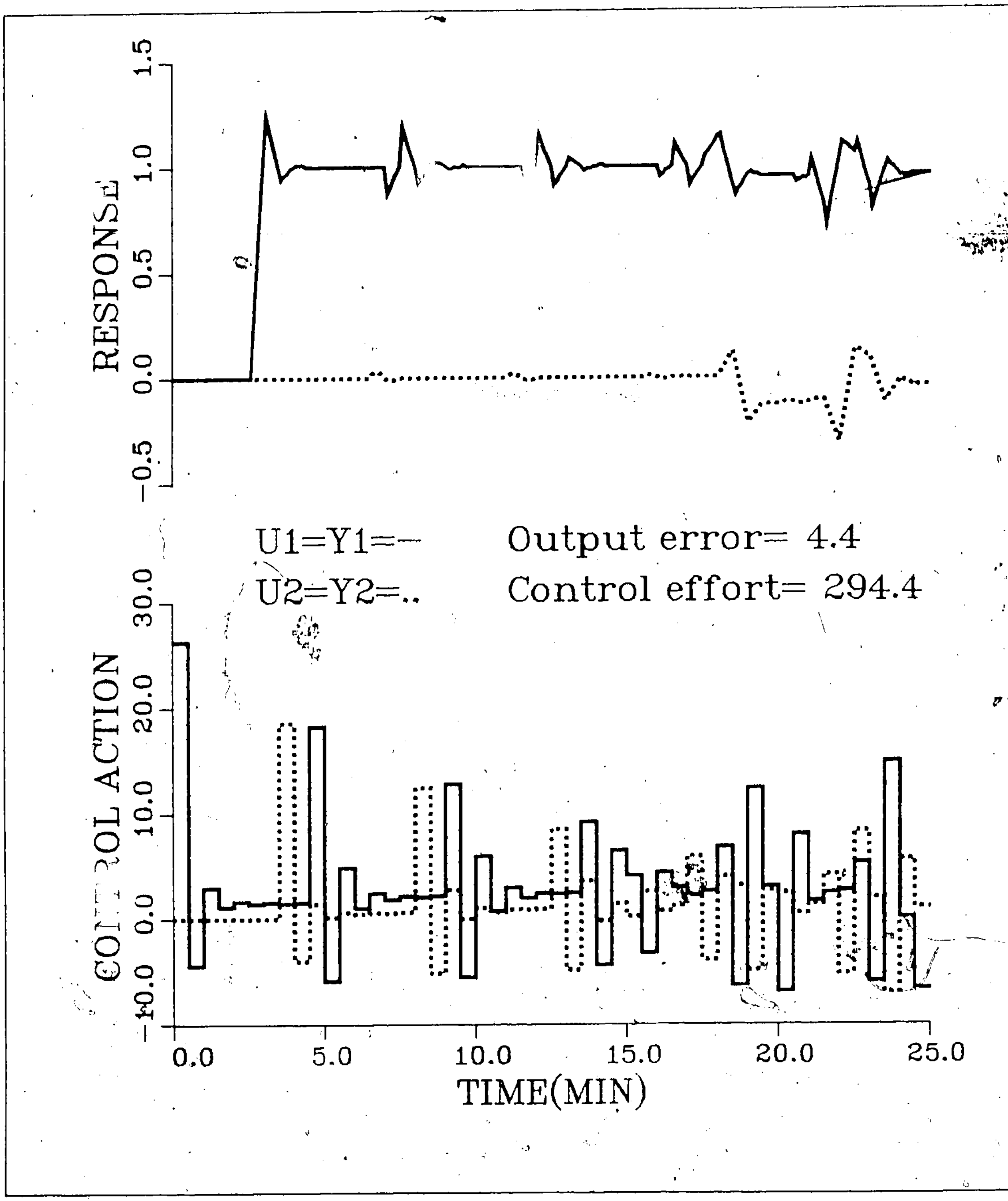

Figure 3.2 Response of column to a positive setpoint change of 1.0 in overhead composition, y_1, under IMC decoupling control, with the time delays of element 1,1 underestimated

In Figure 3.2 the time delay for element 1,1 was underestimated as 5 units of delay, in Figure 3.3 the time delay is overestimated as 6 units of delay. It is evident by comparison of the responses in these two figures that overestimation of the time delay has reduced the undesirable oscillatory behavior in the manipulated variable. However undesirable spikes occur in both trajectories due to the mismatch between the estimate and actual time delay and the truncation error associated with the impulse response series. These two figures also show how extremely sensitive the decoupling controller is to this form of plant/model mismatching. To affirm the ability of IMC to completely decouple the output response when no time delay mismatch occurs, Figure 3.4 shows the closed loop response for the distillation column model when the delay in element 1,1 is reduced to 2.5 minutes. Now the time delay compensator has been able to completely factor the delay giving a perfect deadbeat response for the first N samples. After this time a large departure from the setpoint occurs due to the truncation error. Truncation error occurs because the series length, N, does not encompass a sufficient number of terms. For example, The feedback signal at time k+N, will be zero for a correct model and no disturbance

$$y(k+N)-y_m(k+N)=0$$

however at time k+N+1, the input occurring at time m(k) is no longer included in the vector of past inputs. If the impulse response series coefficient corresponding to N+1 is

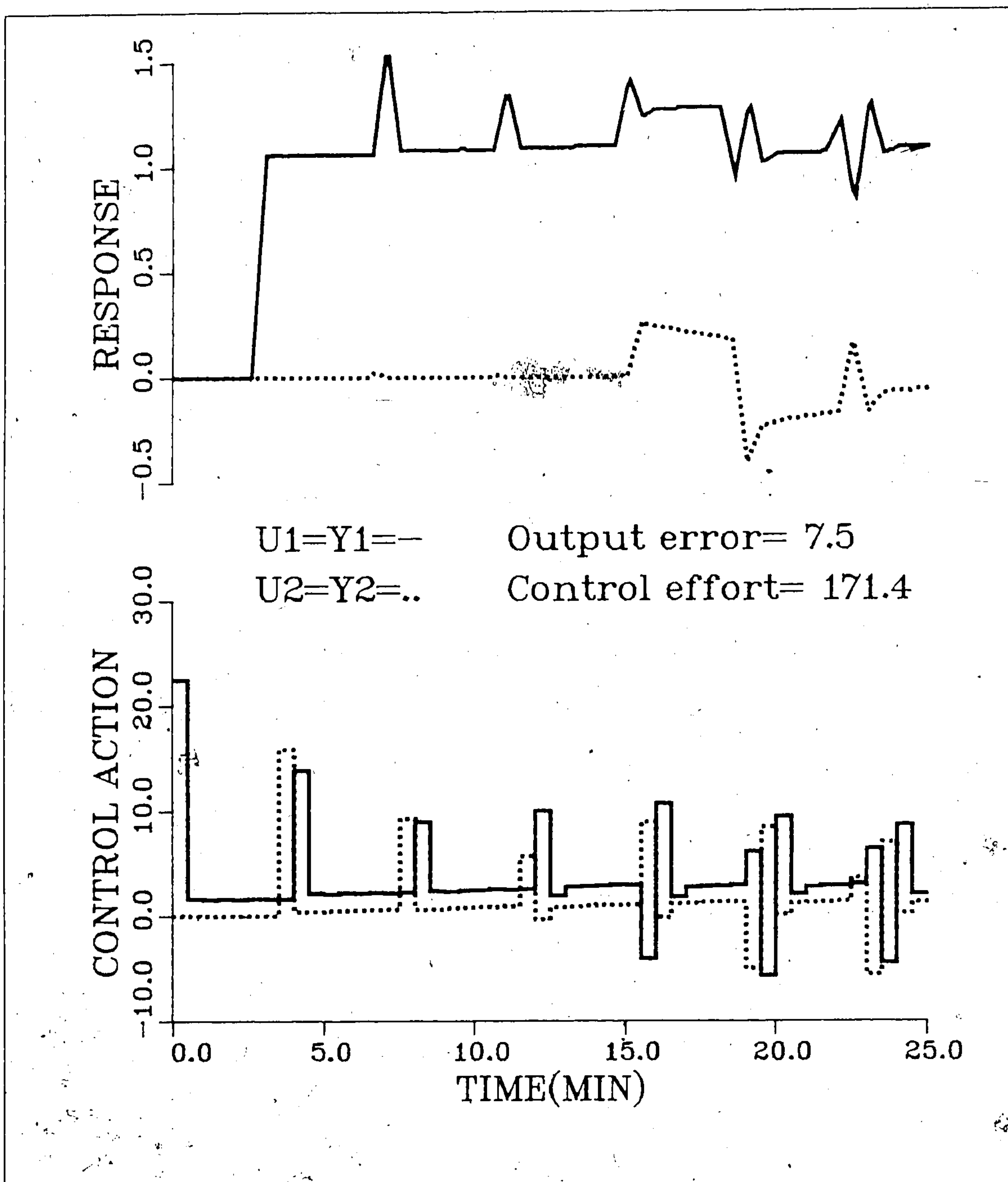

Figure 3.3 Response of column to a positive setpoint change of 1.0 in overhead composition, y_1, under IMC decoupling control, with the time delays of element 1,1 overestimated

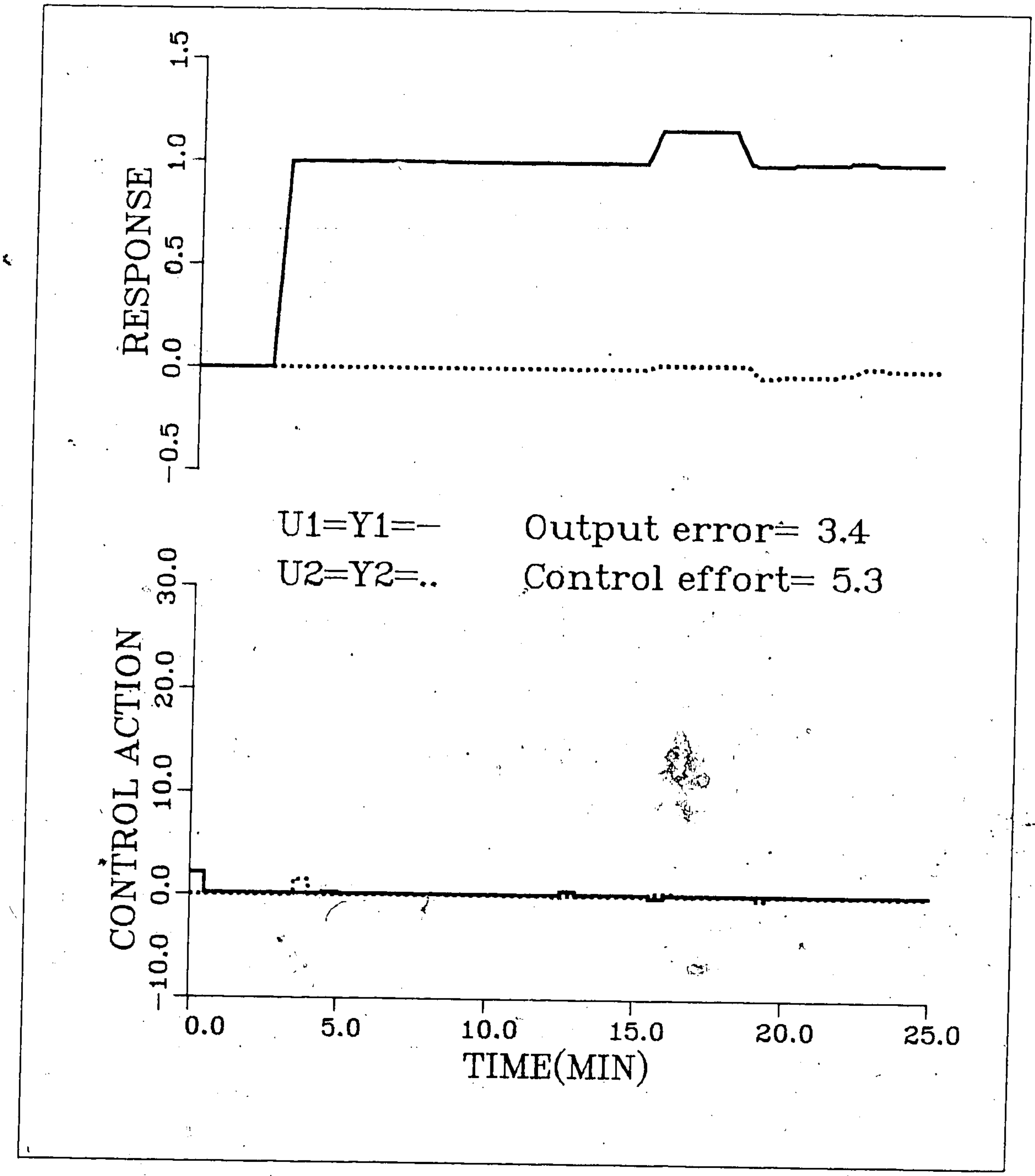

Figure 3.4 Response of column to a positive setpoint change of 1.0 in overhead composition, y_1, under IMC decoupling control, with no time delay mismatch in element 1,1

nonzero, then at time k+N+1

$$y(k+N+1)-y_m(k+N+1)=h_{N+1}\,m(k)$$

Since the feedback signal is no longer zero, the controller begins to take action on what appears to be an unmeasured disturbance, which in turn leads to the bounce in the process output. Theoretically by increasing the series length, N, this error can be reduced. However the large storage requirements of the multivariable system model, place a bound on the maximum size of N. As a result, significant truncation error must be tolerated in many cases.

Coupling effects caused by reducing the input suppression parameter or increasing the ratio of $\underline{B}/\underline{\Gamma}$, are illustrated by the responses in Figures 3.5 and 3.6. In Figure 3.5, the input suppression parameter, M, is reduced from 30 to 2. In Figure 3.6, the ratio of input weighting to output weighting is changed from 0 to .5. Relaxing the decoupling constraint has reduced the extreme excursions of the manipulated variable and reduced the effect of truncation error on the process output. In Figure 3.7, the decoupling controller from Figure 3.3 is used again except this time the filter weighting parameters α_i are both set to .8 instead of 0. The response and control dynamics have improved as a result of the nonzero α_i. This result affirms the design procedure based solely on the decoupling controller and filter extolled by Garcia and Morari.

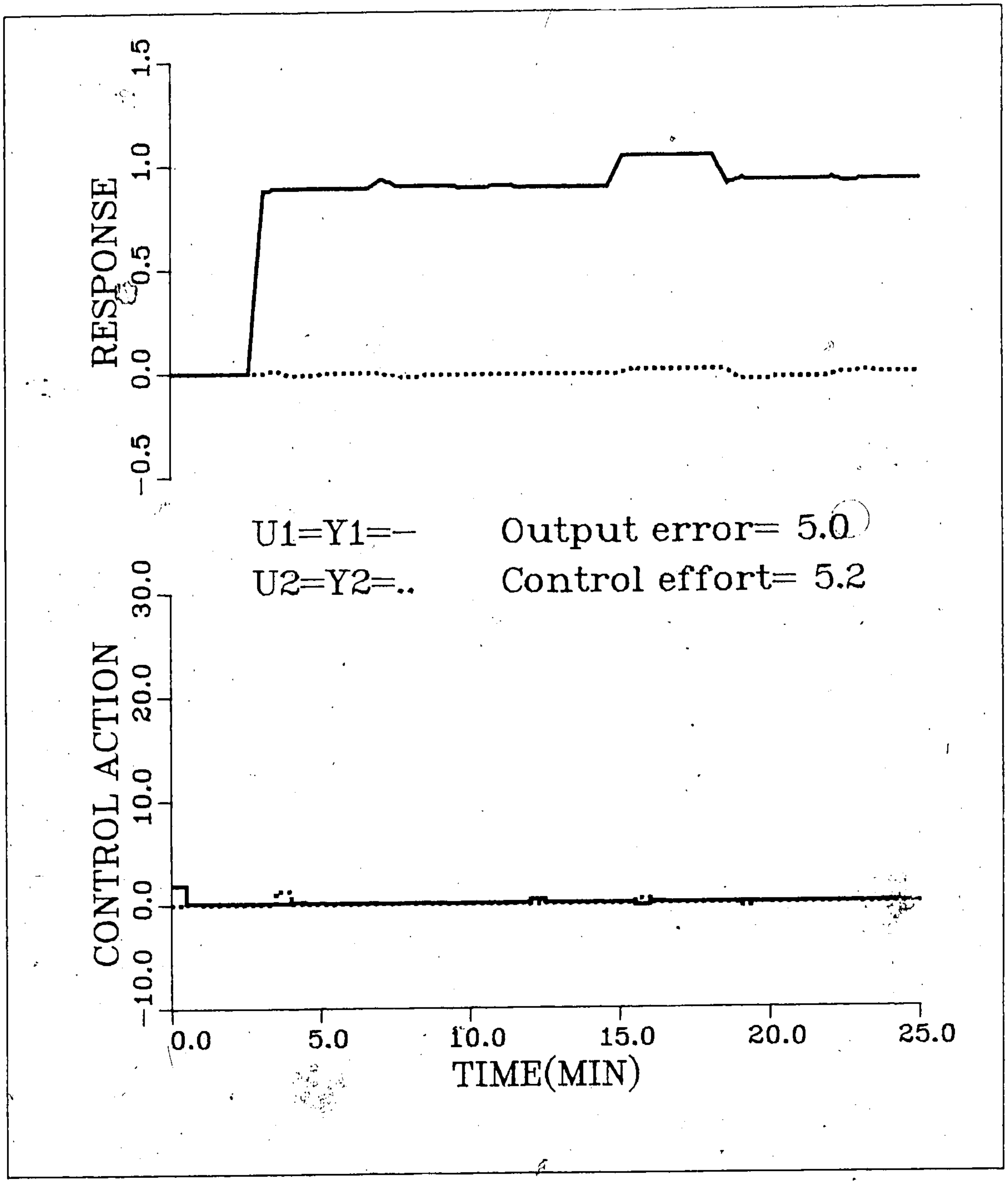

Figure 3.5 Response of column to a positive setpoint change of 1.0 in overhead composition, y_1, under IMC control for; M=2 P=N=30

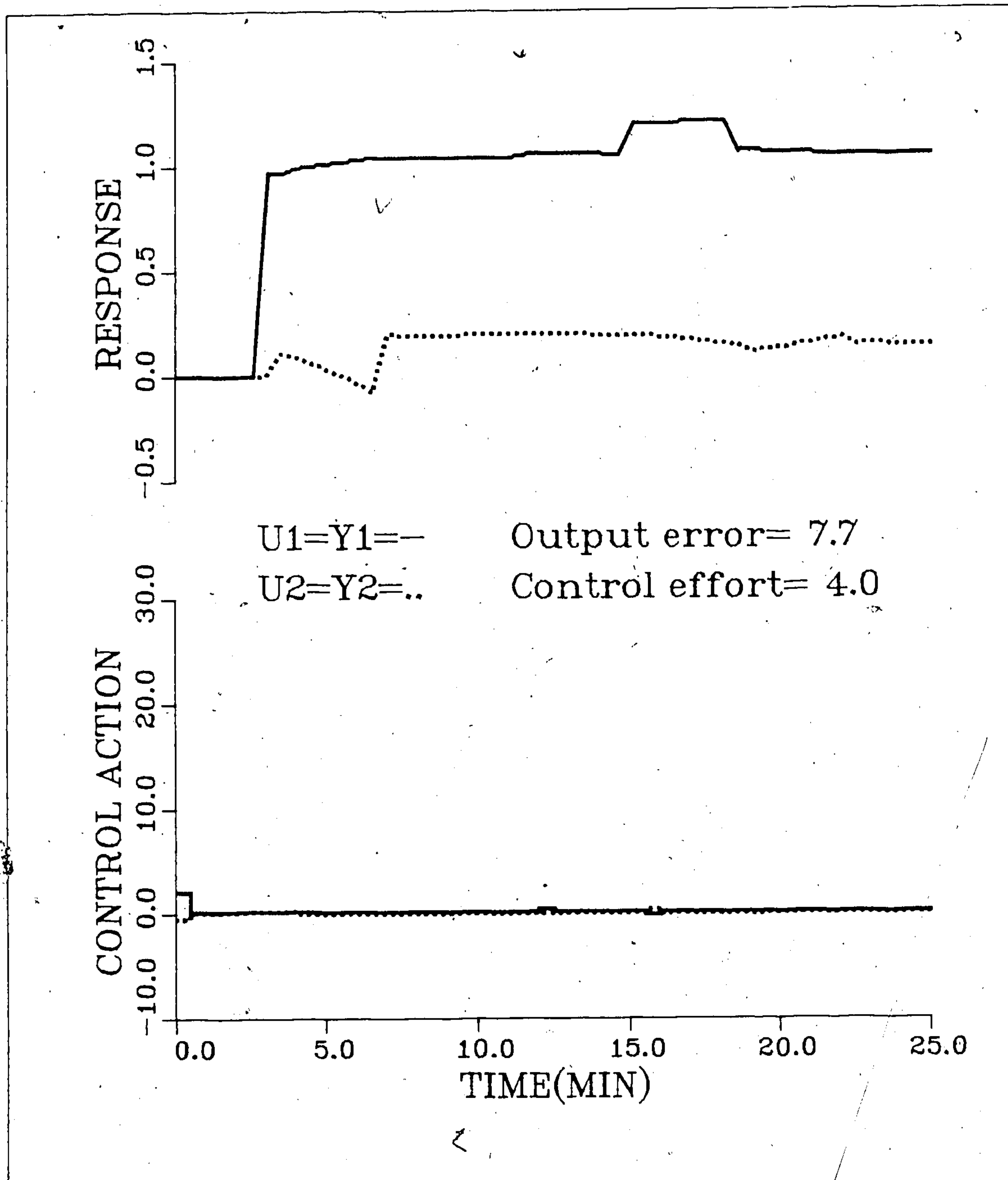

Figure 3.6 Response of column to a positive setpoint change of 1.0 in overhead composition, y_1, under IMC control with β_j=dia(0.5,0.5) for j=1,M

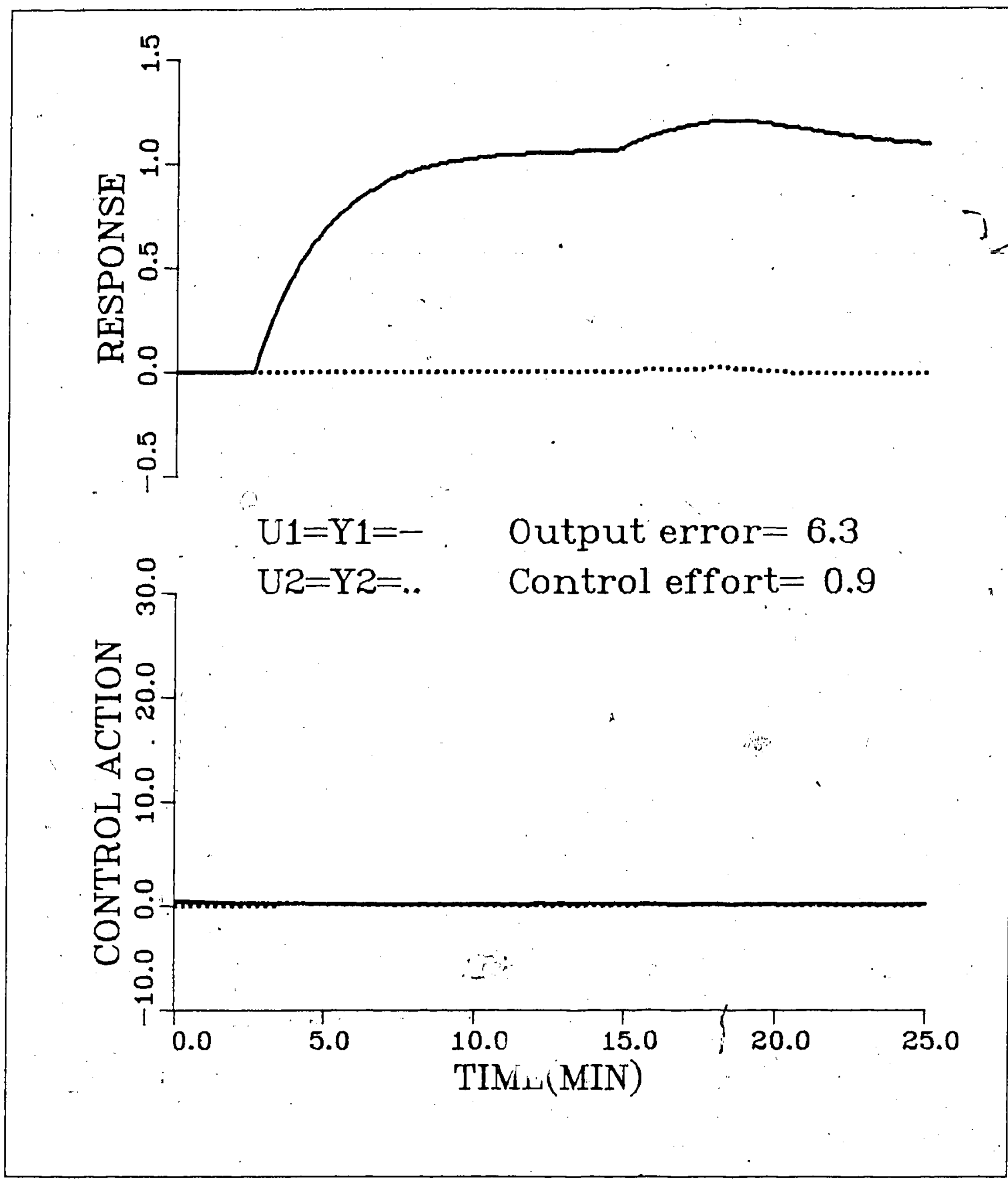

Figure 3.7 Response of column to a positive setpoint change of 1.0 in overhead composition, y_1, under IMC decoupling control with $\alpha_1=\alpha_2=0.8$

Returning to the problem of truncation error in the impulse response model the distillation column model when sampled at 0.5 minutes, needed all 30 terms to achieve satisfactory control. If 30 was insufficient, a slower sampling rate would have been necessary. The performance of the controller when the sampling rate is one minute is illustrated in Figure 3.8. The performance is comparable to the response in Figure 3.7 however the overshoot evident in Figure 3.7 does not appear in this figure because the time series horizon of $\underline{H}(z)$ is doubled. That is NT=30 min in Figure 3.8 versus NT=15 min in Figure 3.7. Increasing the time horizon, NT, has only postponed, not eliminated the truncation error.

3.5.2 Tuning the IMC controller for a process described by a balanced NMP transfer function matrix

Nonminimum phase characteristics or inverse response behavior are frequently modelled by a second order transfer function with a process zero outside the unit circle. As explained in Sections 2.3 and 3.2, the NMP zeros must be compensated to achieve a realizable approximation of $\underline{G}(z)$. Appendix G contains an example of how NMP characteristics can be factored from $\underline{\acute{G}}_{-}(z)$ into $\underline{\acute{G}}_{+1}(z)$. This method of handling NMP characteristics is undesirable because

i. The factorization requires an ARMA model of the process and in most cases an ARMA model will not be available.

ii. A multivariable factorization is an analytical

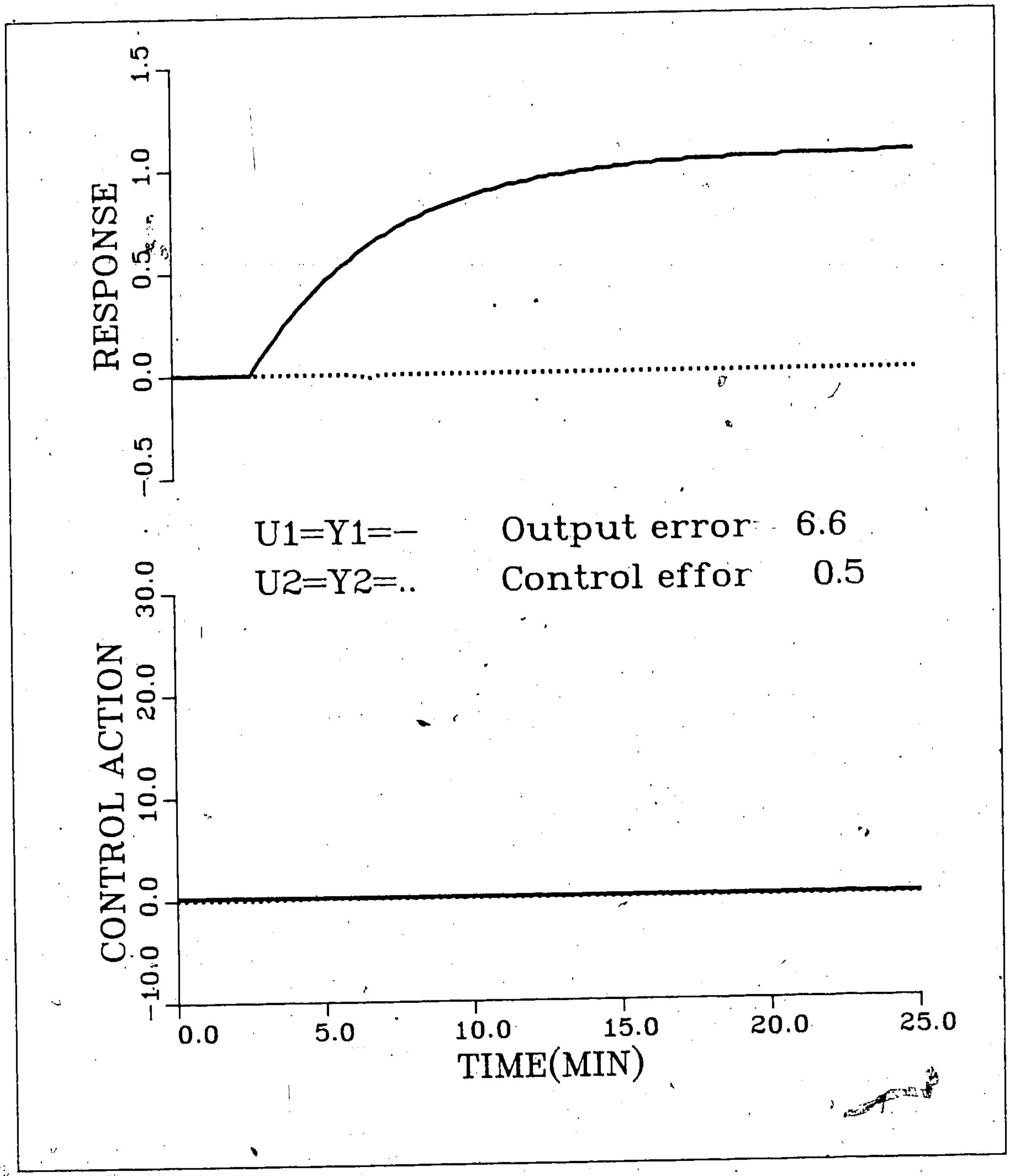

Figure 3.8 Response of column to a positive setpoint change of 1.0 in overhead composition, y_1, under IMC decoupling control for a one minute sample rate

technique not easily programmed into the computer.

iii. The closed loop response becomes very oscillatory.

Instead of removing the NMP zeros from $\hat{G}_-(z)$ two other approaches are suggested. It is demonstrated in Section 2.6, that NMP behavior can be removed from the discrete process representation by choosing a sampling rate that avoids any inverse response behavior in the open loop response.

If the sampling rate cannot be adjusted to avoid the NMP region due to time delay or hardware restrictions, penalties upon the inputs can be applied in the control law formulation to achieve a stable $G_{\underline{C}}(z)$.

The latter method is used in this example to control a nonminimum phase distillation column model when a unit step in the setpoint of y_1 occurs. With three exceptions, the model is taken directly from the model given in Appendix K. One exception is the SISO transfer function used to represent the response of tray #19 temperature to a change in reboiler steam pressure. This second order minimum phase relation is replaced by a second order NMP relation found by Luyben[16]. The other changes effect the time delays in the elements relating steam pressure to overhead composition and steam pressure to side stream composition. These time delays are increased from one to four minutes and 1.2 to four minutes respectively to make the overall transfer function matrix balanced. With these changes the transfer function matrix in Appendix K becomes

$$\underset{\sim}{G}(s)=\begin{vmatrix} \frac{0.66e^{-2.6s}}{(6.7s+1)} & \frac{-0.61e^{-3.5s}}{(8.64s+1)} & \frac{-0.0049e^{-4.0s}}{(9.06s+1)} \\ \frac{1.11e^{-6.5s}}{(3.25s+1)} & \frac{-2.36e^{-3.0s}}{(5.0s+1)} & \frac{-0.012e^{-4.0s}}{(7.09s+1)} \\ \frac{-34.68e^{-9.2s}}{(8.15s+1)} & \frac{46.2e^{-9.4s}}{(10.9s+1)} & \frac{-1.0(-2.50s+1)e^{-s}}{(s+1)(12.5s+1)} \end{vmatrix}$$

y_1=overhead ethanol mole fraction

y_2=side stream ethanol mole fraction

y_3=tray #19 temperature, °C

(corresponding to bottoms composition)

u_1=reflux flow rate, gpm

u_2=side stream product flow rate, gpm

u_3=reboiler steam pressure, psig

The process is sampled every minute. The precompensator $\underset{\sim}{G}_{+1}(z)$=diag(3,3,1) is found using equation 3.2.3. Time delays that are not integer multiples of the sampling rate are overestimated because overestimation has demonstrated better control performance. Before proceeding any further the many different factors that combined to make the IMC controller very difficult to tune should be mentioned.

i. The multivariable transfer function matrix has a wide range of element gains. The SISO transfer function gains in elements 3,1 and 3,2 are a factor of ten greater than the other matrix gains.

ii. The offset compensator was needed to bring the output, y_3 back to its setpoint. However due to the mathematical operations required to compute the offset compensator, large numerical errors were noticed. These errors became intolerable once the model series length, N, became greater than 30. To understand how these errors propagate as N increases consider the multivariable control law

$$\underline{m}(k) = \underline{b}'\underline{K}^{-1}[\underline{\Lambda}'\underline{\Gamma}'\underline{\Gamma}(\underline{E}-\underline{\Psi}\underline{V})] \quad H.4$$

where $\underline{K}=[\underline{\Lambda}'\underline{\Gamma}'\underline{\Gamma}\underline{\Lambda}+\underline{B}'\underline{B}]$

For a 3 by 3 transfer matrix, a series length, N of 30, an optimization horizon, P of 10, and an input suppresion parameter, M of 10, the matrix Ψ has 2700 elements, the matrix Λ has 900 elements. With huge matrices like these going through multiplication, addition and inversion operations, round-o error is inevitable. The offset compensator, $\underline{O}$, uted to make

$$\underline{H}(1)G\underline{c}(1)\underline{O}=\underline{I}$$

or $\underline{O}=[\underline{H}(1)G\underline{c}(1)]^{-1}$

In calculating $\underline{O}$, all of the errors present in the formulation of $G\underline{c}(z)$ are summed into $\underline{O}$.

iii. Because the diagonal element time delays are not integer multiples of the sampling rate a certain amount of plant/model mismatch must be tolerated in the closed loop response.

Keeping these points in mind, the number of terms N, in the impulse response model was chosen as 30 to minimize the truncation error but still permit use of the offset compensator. The optimization horizon P, was 10. Increasing P provided negligible gain in control performance but increased the computation time. From Appendix I, it is known the NMP process can be stabilized by choosing M/P sufficiently small or $\underline{B}/\underline{\Gamma}$ sufficiently large. In this case the choice of M/P was the crucial stability factor. The closed-loop system was only stable for M<5. The best choices for $\underline{B}$ and $\underline{\Gamma}$ were their default values of 0 and $\underline{I}$ respectively. Otherwise the presence of $\underline{B}$>0 or $\underline{\Gamma}$>$\underline{I}$, caused the offset compensator, $\underline{O}$, to have unacceptably large error. Figure 3.9 illustrates the response of the process for

$N=30 \quad P=10 \quad M=3 \quad \tau_0=0$

$\underline{\beta}_j=(0.0,0.0,0.0)$ for $j=1,M$

$\underline{\gamma}_j=(1.0,1.0,1.0)$ for $j=1,P$

$\alpha_i=0.8$ for $i=1,2 \quad \alpha_3=0.0$

The filter weighting parameters, α_i, were chosen to detune or slowdown the responses of y_1 and y_2 but keep y_3, the difficult output to tune, responding as fast as possible to any setpoint changes or disturbance signals. Increasing one or all of the α_i, would have only a slight, detrimental effect. However, decreasing α_1 or α_2, would cause the response to deteriorate very quickly. Sampling the process at a faster rate was impossible because the truncation error caused an unacceptable plant/model mismatch. Sampling the process at a slower rate minimizes the truncation error problems but has the disadvantage of introducing time delay mismatching that eventually will cause an unstable response.

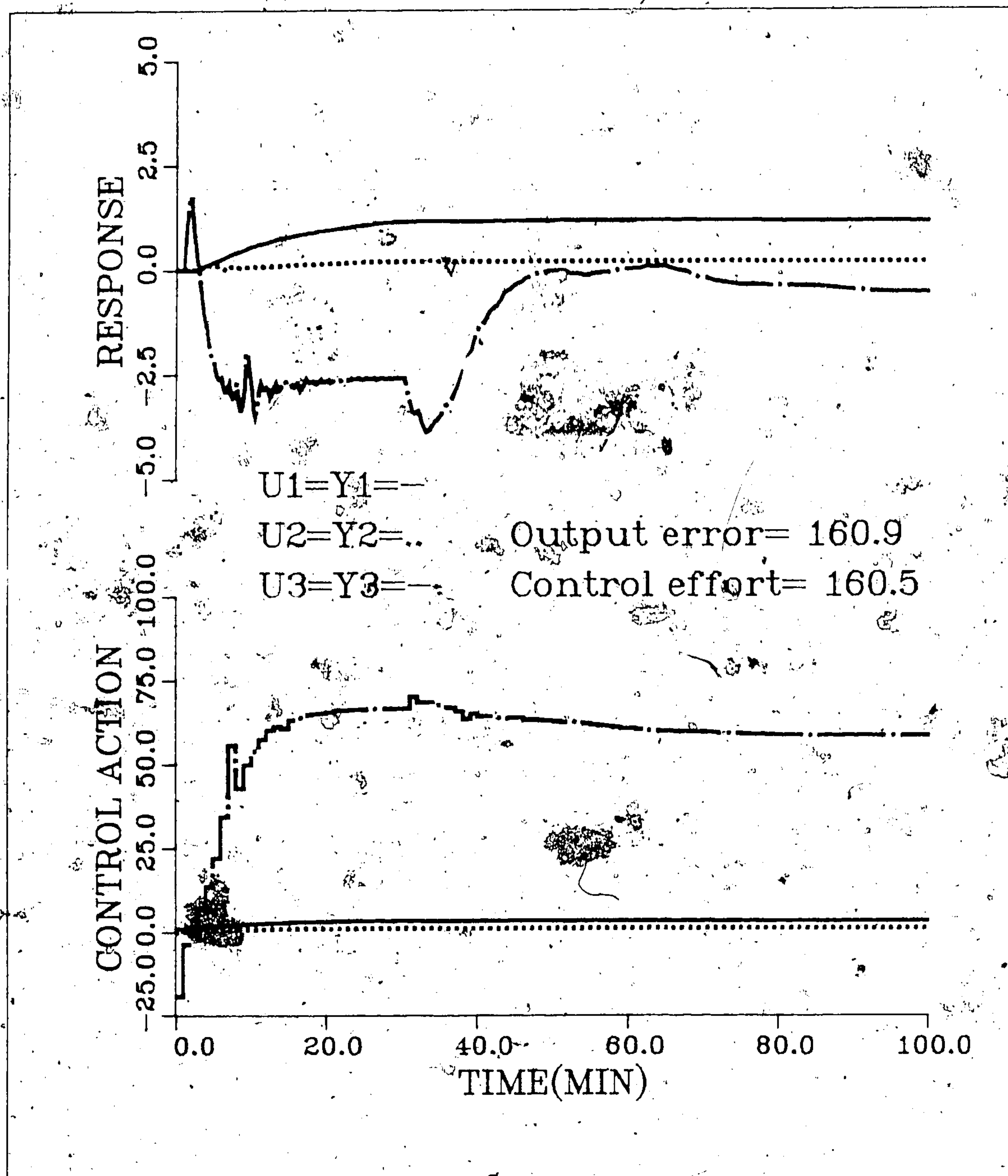

Figure 3.9 Response of NMP column to a positive setpoint change of 1.0 in overhead composition, y_1, under IMC control for; M=3 P=10 N=30

3.5.3 Tuning an unbalanced, minimum phase transfer function matrix

A transfer function matrix is termed unbalanced, when the minimum time delays are located in the off-diagonal positions. As a consequence of imbalance the decoupling controller will be nonrealizable due to singularities in the impulse response series. Several methods IMC uses to control unbalanced process transfer functions, discussed in Section 3.3 are

i. Perform Guassian elimination on the elements of the control law to eliminate the rank deficiencies of $\underline{\Gamma\Delta}$. After the singularities are removed, the decoupling controller will be realizable.

ii. Simply remove the first τ_0 singular matrices from the impulse response series. Although this provides a decoupling controller the control dynamics can be poor or unstable due to plant/model mismatch.

iii. Relax the decoupling constraint to allow input and output weighting. By sufficiently penalizing the control law a stable controller results.

The first approach will not be considered because it requires a considerable amount of extra programming and provides only marginally better performance than the coupled controller. The second method is also not considered because of the undesirable performance characteristics already stated.

A two by two matrix with the minimum time delay in position 1,2 is chosen from the distillation column model given in Appendix K so that it represents an unbalanced process transfer function. The two by two transfer function matrix selected for study is

$$\begin{vmatrix} y_1(s) \\ \\ y_2(s) \end{vmatrix} = \begin{vmatrix} \frac{0.66e^{-2.6s}}{(6.7s+1)} & \frac{-0.0049e^{-1.0s}}{(9.06s+1)} \\ \\ \frac{1.11e^{-6.5s}}{(3.25s+1)} & \frac{-0.012e^{-1.2s}}{(7.09s+1)} \end{vmatrix} \begin{vmatrix} u_1(s) \\ \\ u_2(s) \end{vmatrix}$$

y_1=overhead ethanol mole fraction

y_2=side steam product flow rate, gpm

u_1=reflux flow rate, gpm

u_2=reboiler steam pressure, psig

As in the previous two examples, an impulse response model that correctly describes the process is provided to the IMC controller. For a sampling rate of one minute, the time delay factorization gives $\underset{\sim}{G}_{\cdot 1}=(3,4)$ with the measure of imbalance τ_0 found from equation 3.3.8 to be two. The IMC controller is used to control the unbalanced process when a unit step in the setpoint of y_1 is desired. Again the IMC controller was very difficult to tune and provided very poor control. The reasons for the poor performance are

i. The process model has time delays that are not integer multiples of the sampling rate. This causes plant/model mismatching to occur. The effect of this mismatch, is reduced not eliminated, by overestimating the time delays.

ii. The SISO elements that make up $\underset{\sim}{G}(z)$, have very different steady state gains. Elements 1,1 and 2,1 have gains approximately 100 times greater than the gains in elements 1,2 and 2,2.

iii. The first τ_0 matrices of the impulse response series describing this process will be singular. To stabilize the response input weighting on at least these first two matrices will be necessary to eliminate the singularities.

iv. To reduce the steady state offset caused by the input weighting, the offset compensator is used. Associated with the offset compensator, $\underline{Q}$, are numerical errors, arising from round-off error and plant/model mismatch due to the truncation error.

Simulation results from two trials are presented to illustrate the effect of $\underline{B}$ and $\underline{\Gamma}$ on the closed loop response. In Figure 3.10 the main objective was to eliminate the steady state offset. Because the offset compensator was very prone to error for $\underline{B}\neq 0$, it was important to keep the input weighting small. Futhermore, $\underline{Q}$, was also susceptible to large values of $\underline{\Gamma}$. Bound by these limitations, the steady state offset was minimized for

$N=30 \quad P=8 \quad M=3 \quad \alpha_i=0.8$ for $i=1, 2$

$\beta_j=\text{diag}(0.0,0.05)$ for $j=1,M$

$\gamma_j=\text{dia}(1,1)$ for $j=1,2$ and $\gamma_j=\text{dia}(1,0)$ for $j=3,P$

The value for N, was the maximum possible before the offset compensator became useless. It was found the value for P made very little difference for $P\geq 8$. Being the smallest, eight minimizes the computation time. Reducing M, increases

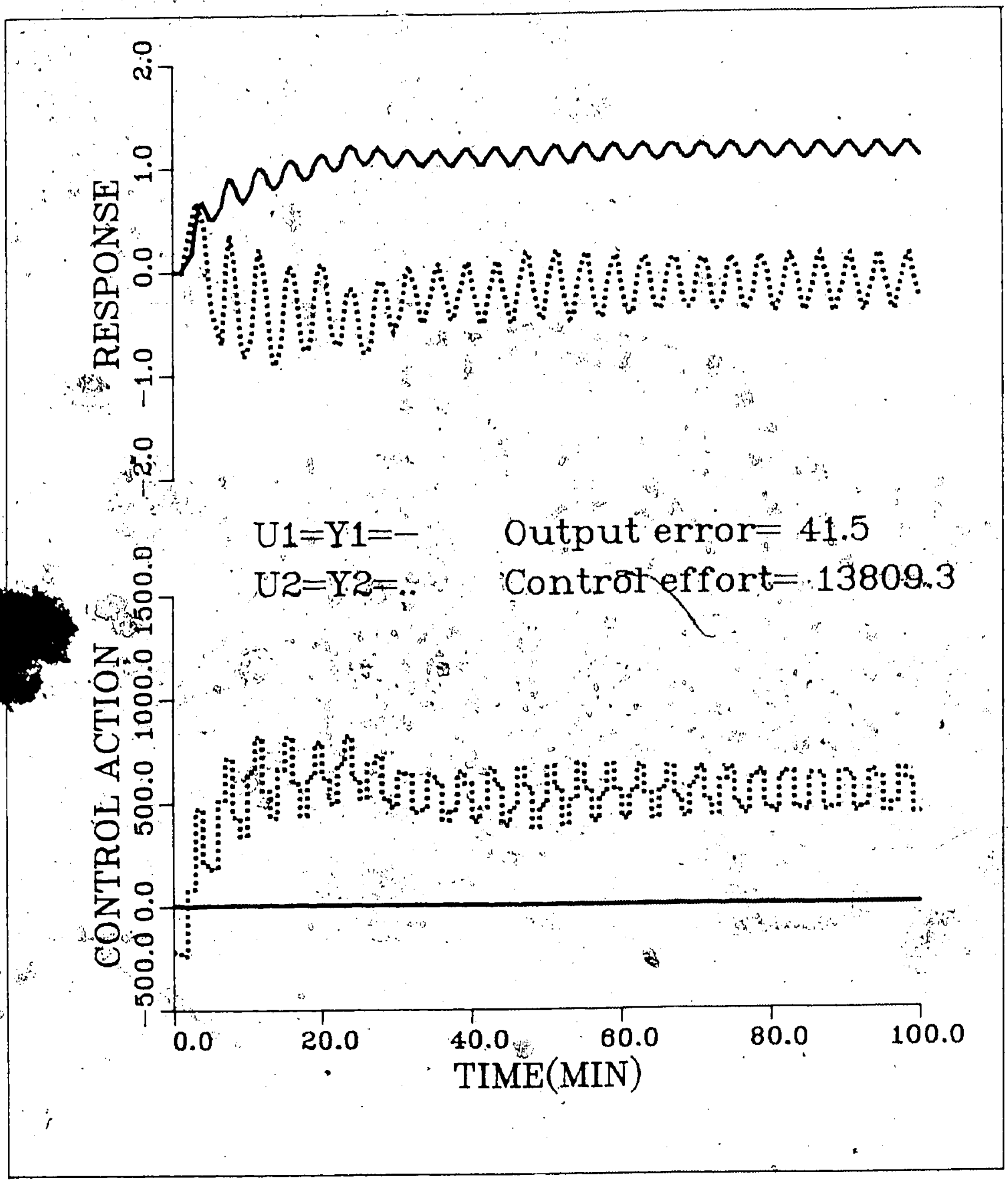

Figure 3.10 Response of unbalanced column to a positive setpoint change of 1.0 in overhead composition, y_1, under IMC control with; M=3 P=8 N=30 γ_j=dia(1,1) for j=1,2 and γ_j=dia(1,0) for j=3,P β_j=dia(0.0,0.05) for j=1,M

the closed loop stability, but it also, in effect, detunes the closed loop system. Everytime the controller is further detuned, the job of the offset compensator becomes more vital. In this example, M, was set to its maximum value. Increasing M further would cause the system to become unstable. The filter weighting parameters, α_i were set very high, to prevent oscillation as much as possible. Using the maximum possible value for M, means the closed loop response is only marginally stable. Appling large α_i, enhances the robustness. The input weightings, β_j, were chosen to ensure all singularities were removed, yet penalize the response as little as possible. Whether, β_j was 0.0,0.05 or 0.0,0.01 made very little difference and furthermore there is no input weighting on u_1. The output penalty parameters, γ_j, were $\underline{I}$ for the first two horizon steps. The remaining six steps only penalized excursions of y_1 not y_2. Because the input and output weighting are only meaningful relative to each other, the first two output weightings are unity. If the penalty on y_2 was zero for all P steps, the input penalty of 0.05 would be meaningless and the system would become unstable. If γ_j=1.0,0.0 for j=2,P, the system is unstable. A balance was struck between stability and setpoint tracking when $\gamma_j=\underline{I}$ for j=1,2 and γ_j=1.0,0.0 for j=3,P. By setting γ_j, to 1.0,0.0 for j=3,P, an infintely greater weighting is placed upon excursions of y_1, than y_2. This sacrifices the quality of the side stream product, y_2, but it provides better control on the overhead product composition, y_1. The response in Figure 3.10 is undesirable because of insufficient penalties on process inputs. Closed loop robustness has been sacrificed to provide reasonable setpoint tracking.

In Figure 3.11 the input and output penalties have been increased to ensure a robust response. The tuning parameters used in Figure 3.11 are

$N=30 \quad P=8 \quad M=3 \quad \alpha_i=0.8$ for $i=1,2$

$\beta_j=\mathrm{dia}(0.1,1.0)$ for $j=1,M$

and $\gamma_j=\mathrm{dia}(1,1)$ for $j=1,P$

The changes have eliminated most of the undesirable oscillations found in Figure 3.10. However, because the input and output penalties have further detuned the IMC controller, the offset compensator's deficiences become more noticeable.

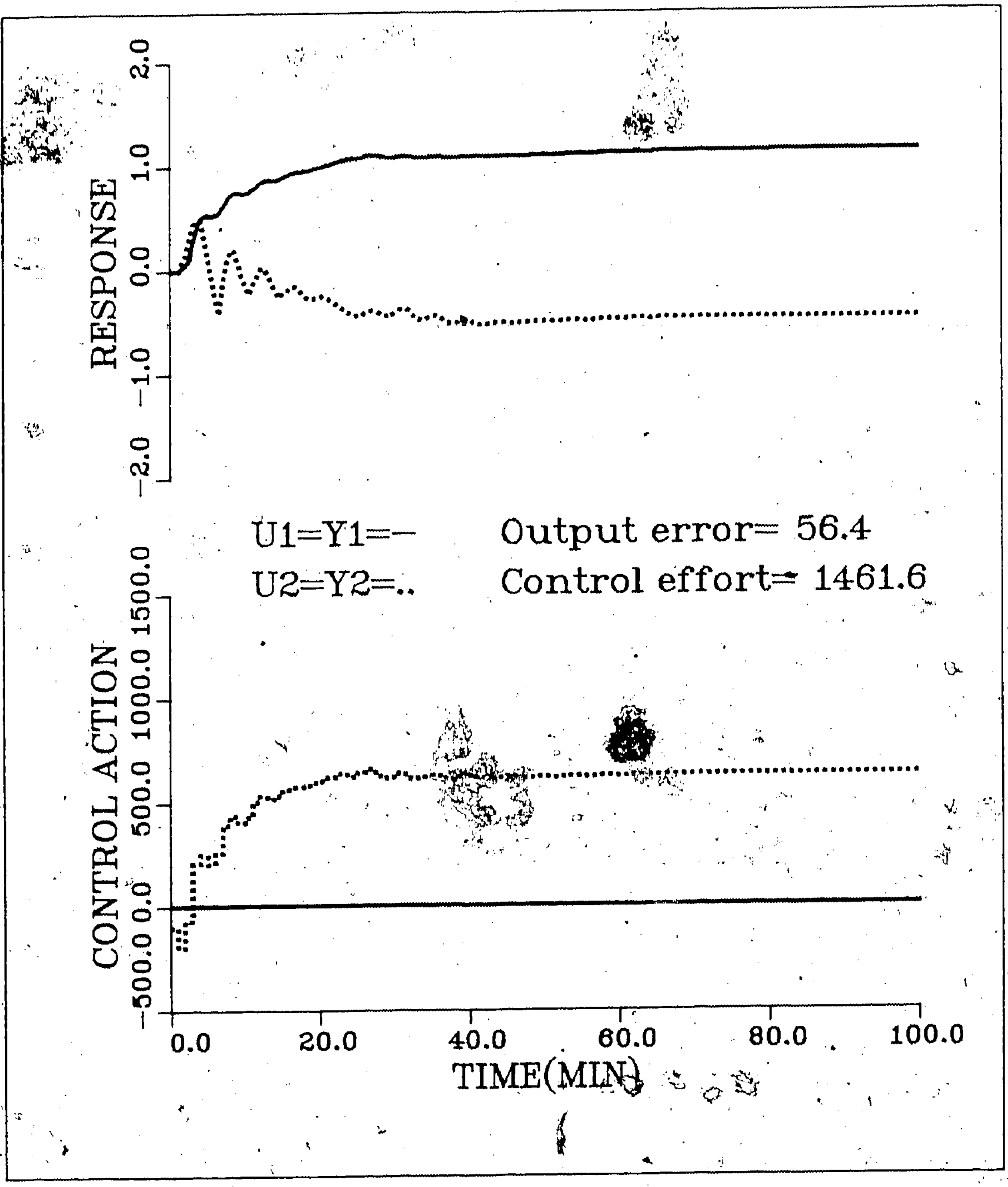

Figure 3.11 Response of unbalanced MP column to a positive setpoint change of 1.0 in overhead composition, y_1, under IMC control with; M=3 P=8 N=30 γ_j=dia(1,1) for j=1,P β_j=dia(.1,1) for j=1,M

4. A COMPARISON OF IMC TO RELATED ALGORITHMS

4.1 Introduction

In this chapter the concepts of IMC will be compared to other control strategies to illustrate the tradeoffs or comprimises necessary in the application of the IMC control law. This material will relate IMC to strategies with an obvious resemblance as well as those with only a subtle similarity. Single and multivariable schemes will be considered with the first comparison the single variable Smith predictor versus IMC.

4.2 IMC vs the Smith Predictor

The Smith predictor(SP) was developed by Smith[7] to remove time delays from the closed loop characteristic equation enabling higher controller gains. The original formulation was developed for implementation using an analog system. Since that time discrete, formulations of the SP have been developed. The discrete representation of the SP as given by Doss and Moore[17] is shown in Figure 4.1. Garcia and Morari[1] have demonstrated that the predictor block of Figure 4.1 can be split into two equivalent blocks, one containing the delay free process model, the other containing the delayed process model. This new equivalent structure is illustrated in Figure 4.2. The similarities between this diagram and the IMC block diagram in Figure 2.2, occur because both algorithms use the same time delay

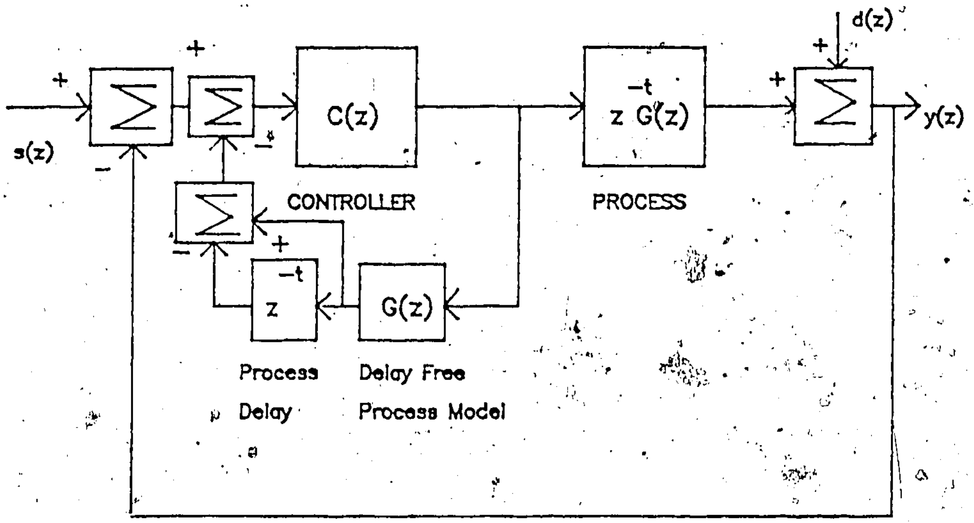

Figure 4.1 Block diagram of the discrete Smith predictor

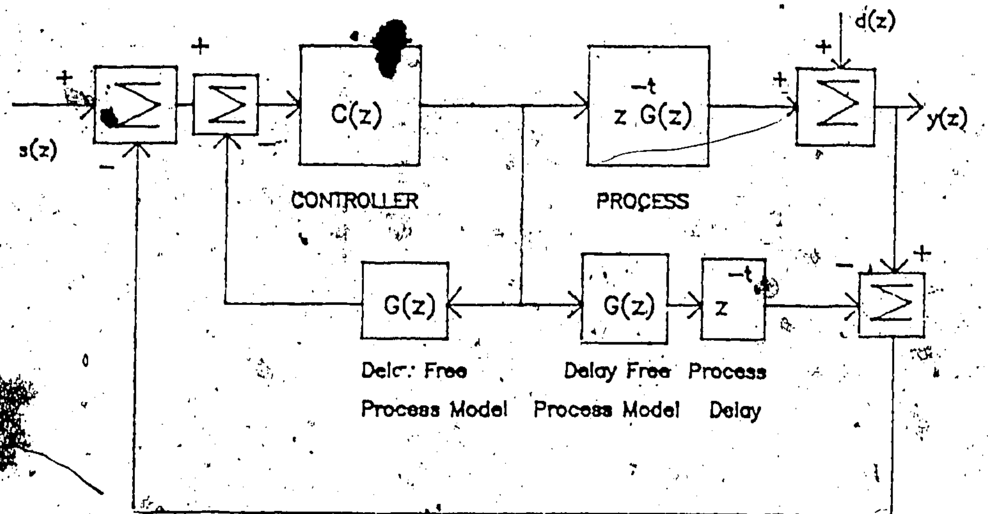

Figure 4.2 Block diagram of the Smith predictor where the predictor block is split into two parts.

compensation technique. When the process model is correct, the closed loop characteristic equation reduces to the delay free characteristic equation. When the plant model is incorrect, the difference between $y(k-\tau)$ and $y_m(k-\tau)$ is used to compensate for errors in the model.

The algorithm of Garcia and Morari departs from the SP policy with regards to design and implementation of the controller. The IMC philosophy encourages the use of a minimum variance controller. Robustness is added to the closed loop system by the use of a first order filter in the forward path. The filter location is shown in Figure 3.1. This approach supposedly leads to the best tradeoff between robustness and optimal performance.

Conversely, the SP technique relies upon discrete forms of the traditional, three mode, PID feedback controller. The PID structure limits the ability of the controller to cancel process zeros and poles which, in turn, limits the ability of the controller to provide optimal performance. Suboptimal performance of the SP algorithm is further encouraged by the use of tuning procedures like Ziegler Nichols or Coon Cohen, which implicitly sacrifice setpoint tracking and disturbance rejection for the sake of closed loop robustness.

The relative merits of IMC and a Smith predictor with a PI controller are highlighted in two example simulations. In the first example, the minimum phase second order transfer function, used to illustrate the IMC tuning procedure in Section 2.6, is used to demonstrate the ability of IMC and the SP-PI combination to provide regulatory control when an unmeasured disturbance occurs.

The continuous process is sampled every four minutes, resulting in the discrete transfer function given by equation E.1.2 in Appendix E. Both control laws are provided with perfect process models. A disturbance of unit magnitude occurs when the process is operating at steady state, that is $y(0^-)=0$, $u(0^-)=0$, $d(0^-)=0$ but $d(0^+)=1$ giving a $y(0^+)=1$. The IMC controller is tuned to provide deadbeat response, with a filter constant α of .2 to prevent intersample ripple. The deadbeat controller is used because it will reject the unmeasured load disturabance faster than other IMC controllers and because its design, $Gc(z)=G_-(z)^{-1}$, makes it easy to compare to other strategies. The deadbeat controller parameters are

$$N=30 \quad M=P=1 \quad \beta_j=0.0 \text{ for } j=1,M \quad \gamma_j=1.0 \text{ for } j=1,P$$

The IMC closed loop response is shown in Figure 4.3.

The SP-PI combination is tuned to best approximate the deadbeat response of the IMC controller. A velocity form of the PI structure given by

$$u(k)=u(k-1)+Kce(k)-Kc\phi e(k-1)$$

is implemented to prevent saturation of the control signal. To choose the Kc and ϕ values the second order process model is approximated by a first order model. Once the first order approximation is found, the PI tuning parameters, Kc and ϕ can be chosen so that the first order pole and gain

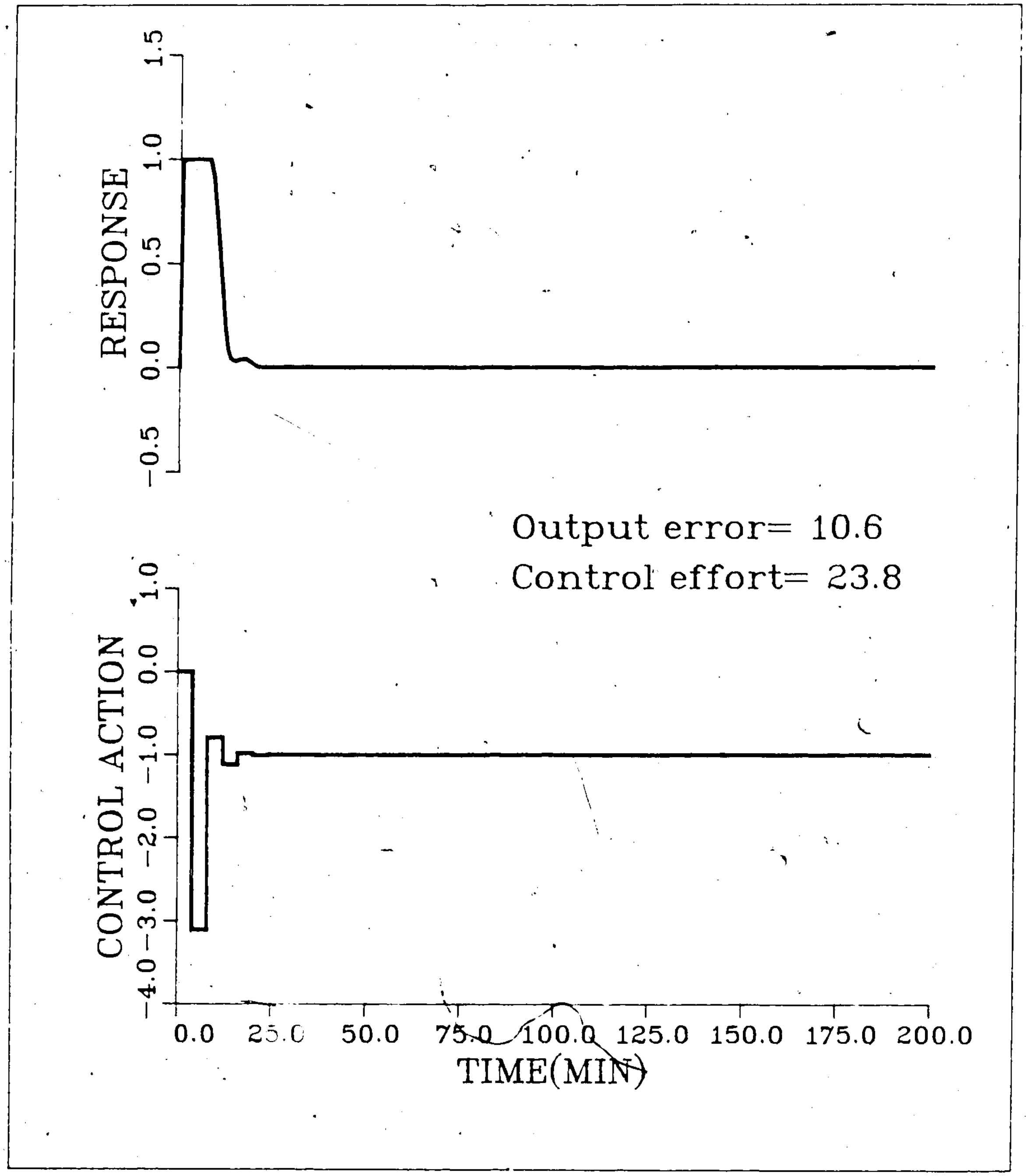

Figure 4.3 Response of the MP system to an unmeasured disturbance of 1.0 under IMC deadbeat control

are cancelled. This type of PI controller, will closely resemble the minimum variance or deadbeat controller. The discrete second order transfer function is given by

$$y(k)=\frac{(.109z^{-2}+.0729z^{-3})}{1-1.12z^{-1}+.301z^{-2}}u(k)+d(k) \qquad \text{E.1.2}$$

The poles of this transfer function, found from the roots of the denominator polynomial, are located at .448 and .672. To tune the PI controller the dominant pole, .672 is chosen for ø. The controller gain Kc, is chosen such that

$$\frac{1}{Kc}=\frac{b_1+b_2+\ldots}{1-.672} \qquad \text{to give } Kc=3.03$$

The SP-PI response for these Kc and ø values is illustrated in Figure 4.4 and is almost identical to the IMC deadbeat controller.

The second example looks at the ability of the two control strategies to perform regulatory control on a nonminimum phase(NMP) process. The single variable continuous NMP transfer function, given by equation 2.6.2, is used to represent the process. The continuous process is sampled every four minutes to give the discrete representation

$$y(k)=\frac{(-.467z^{-3}+.791z^{-4})}{1-.689z^{-1}+0.0123z^{-2}}u(k)+d(k) \qquad \text{E.2.2}$$

As in the previous example the process is at steady state when the unit step in d(k) occurs. That is, $y(0^-)=0$, $u(0^-)=0$, $d(0^-)=0$ and then $d(0^+)=1$ giving $y(0^+)=1$. The IMC deadbeat controller will be unstable due to the presence of a process zero which is outside the unit circle. However, from the tuning guidelines presented in Section 2.6 by

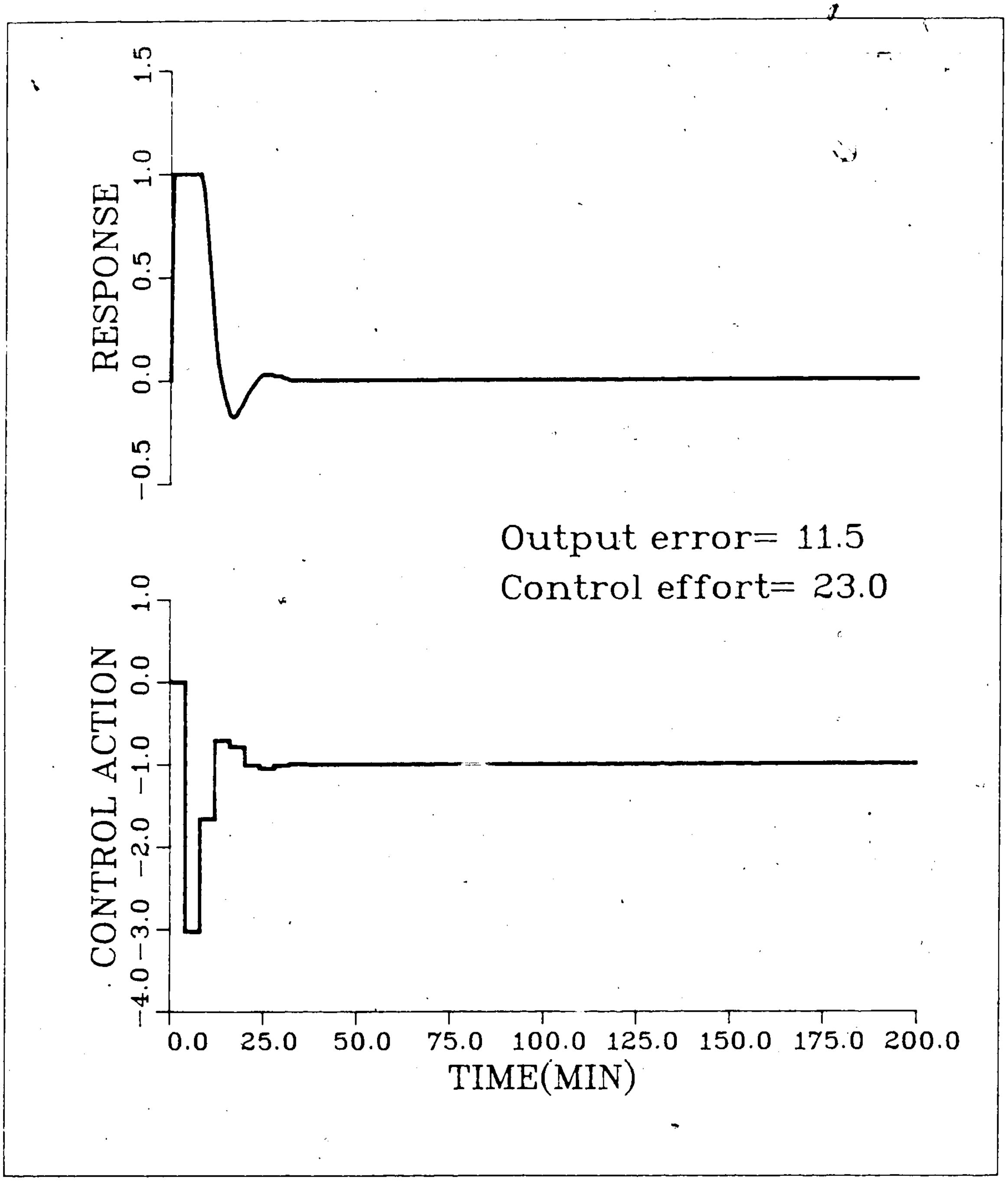

Figure 4.4 Response of the MP system to an unmeasured disturbance of 1.0 under SP-PI control

selecting the input suppression parameter M, to be less than the the optimization horizon, P, a stable controller will result. The IMC controller performed the best when

$P=10$ $M=8$ $N=30$ $\gamma_j=1$ for all j

$\beta_j=0.0$ for all j $\alpha=0.6$

Figure 4.5 shows the closed loop response for this system.

The SP-PI system was tuned to best approximate the deadbeat closed loop response. The controller constants, Kc and ø, were chosen using the same method as for the minimum phase example. The resulting Kc and ø were 3.03 and .67 respectively. However these parameters resulted in an unstable closed loop response. To stabilize the closed loop system the controller gain was reduced until the least output error occurred. This corresponded to a gain of .6. The closed loop response for Kc=0.6 and ø=.67 is shown in Figure 4.6. Surprisingly the SP-PI combination shows better regulatory behavior than the IMC technique. This result can be related to the fact that IMC is essentially a process pole-zero cancelling controller. In this example, the IMC controller was penalized until a stable approximation of the pole-zero cancelling controller resulted. The method used to tune the PI controller did not attempt to cancel the NMP process zero and as a result this controller performed better.

The IMC controller showed better disturbance rejection than the Smith predictor with a PI controller in the case where the deadbeat controller was stable. However when the

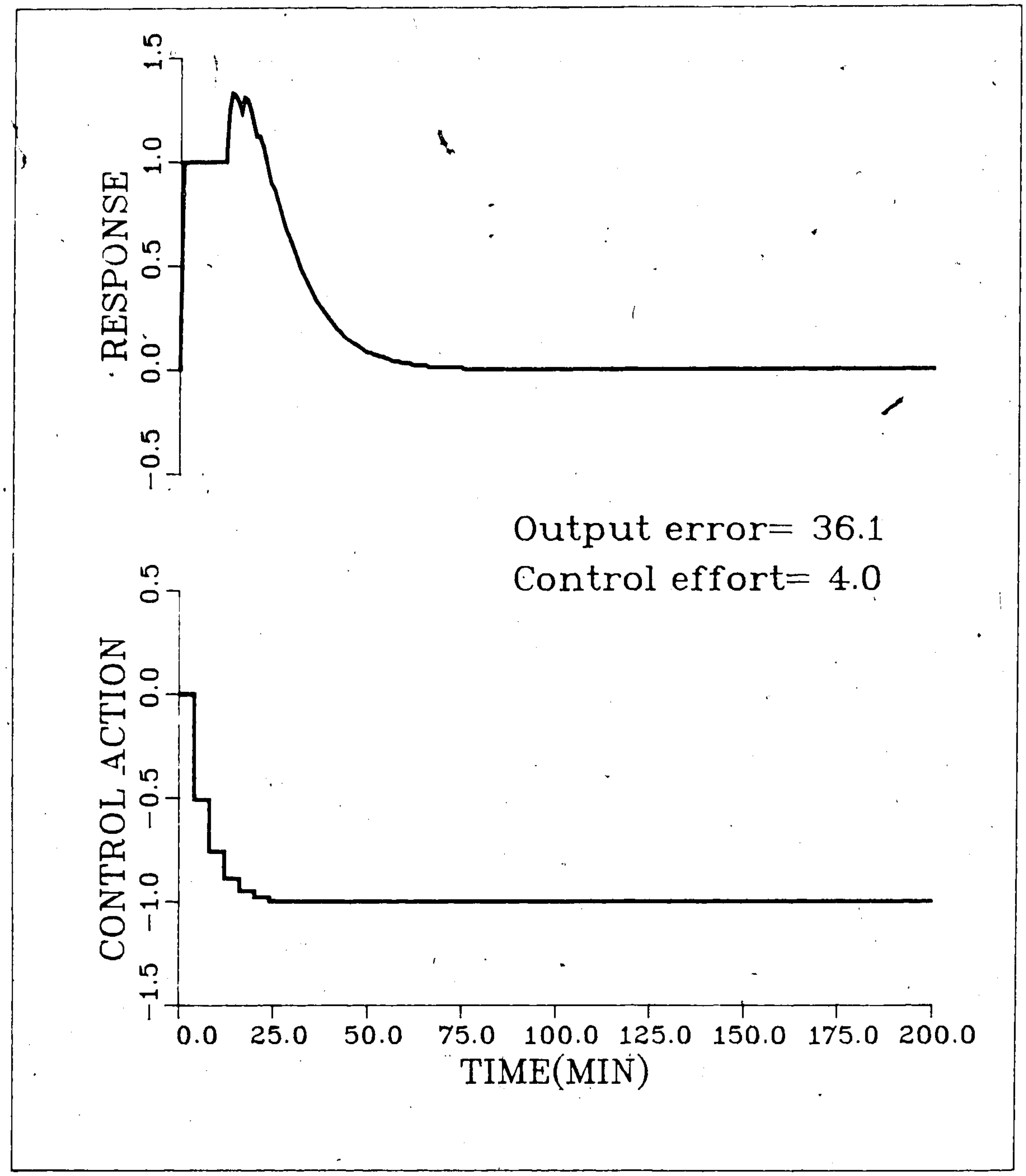

Figure 4.5 Response of the NMP system to an unmeasured disturbance of 1.0 under IMC control

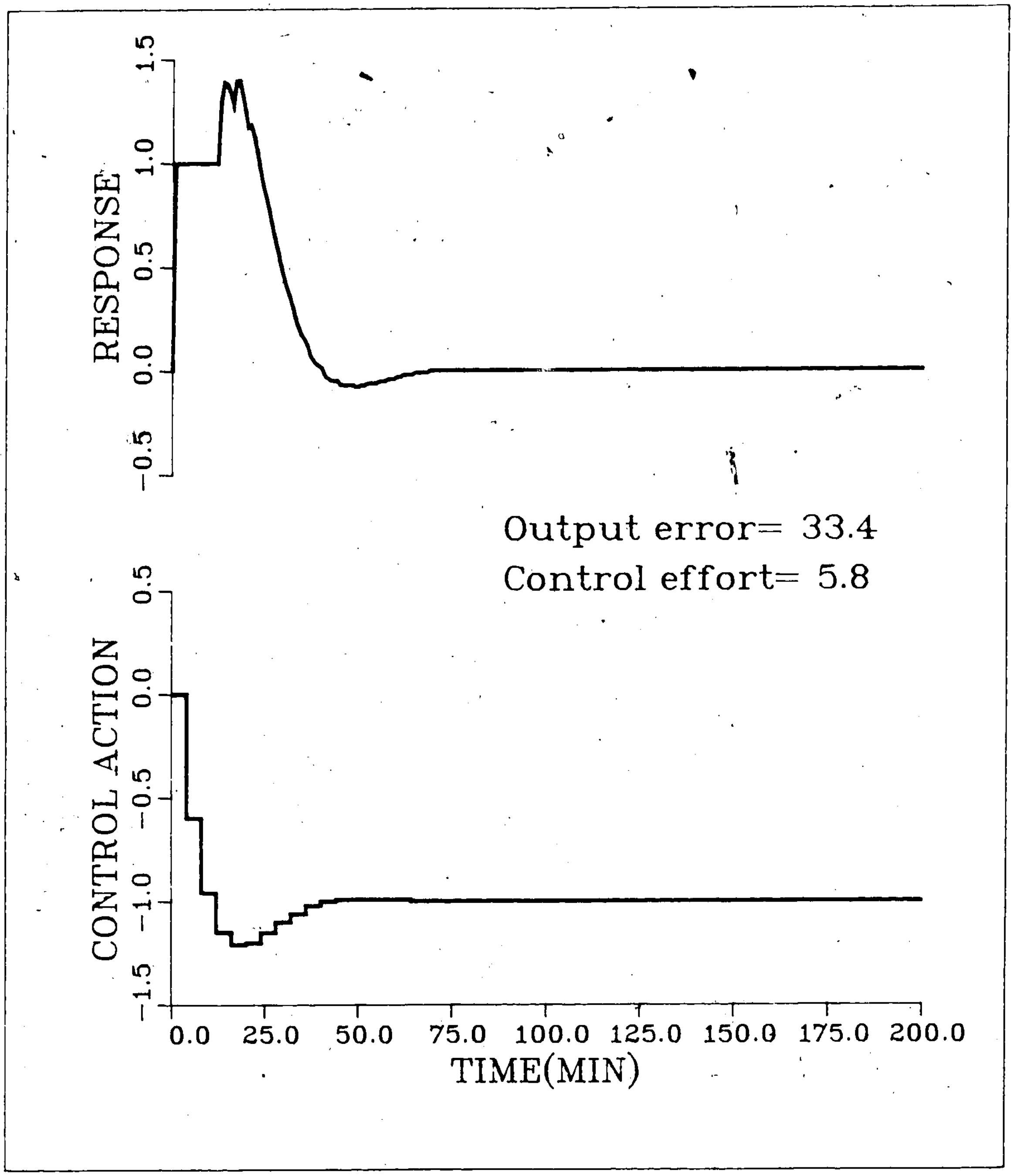

Figure 4.6 Response of the NMP system to an unmeasured disturbance of 1.0 under SP-PI control

deadbeat controller is not realizable, it is difficult to achieve a desirable response trajectory with IMC.

4.3 IMC vs the Inferential Smith Predictor

In Chapter 2 the IMC control law was defined through a two phase derivation. In the first phase the concept of the perfect controller was introduced. It was shown this controller was given by the realizable inverse of the process transfer function. Theory predicts the perfect controller should provide high performance servo and regulatory control. In practice this type of controller has poor robustness because it is susceptible to modelling errors and process zeros near the unit circle. Another more serious drawback of this first design concept concerned the implementation of the perfect controller when the process exhibited NMP behavior. As a result a second method, the predictive control strategy, given by equation 2.4.2 was introduced.

The predictive controller contained many of the desirable features of the perfect controller, and many others not included in the first phase. Shaping of the closed loop dynamics to meet operating constraints or to enhance robustness, was not possible with the first design method.

On the basis of an examination of control theory articles published over the last few years IMC can be considered to be a hybrid of the control philosophies, of perfect control and predictive control. The concept of a perfect controller was discussed in 1979 by Brosilow[4] in connection with his development of the inferential Smith predictor(ISP). Developments similar to the IMC predictive controller can be found in the MAC and DMC algorithms discussed in the next section.

The block diagram for the ISP of Brosilow is shown in Figure 4.7. If the G_i is replaced by Gc, only the discrete representaion of Garcia and Morari[1] differentiates IMC from ISP. Brosilow, as do Garcia and Morari, selects G_i based upon a realizable inverse of the process model, $\acute{G}$. Similarly, F(s) is used to enhance the robustness of the closed loop system. Both authors emphasis the importance of the closed loop structure because it ensures closed loop stability when the process, process model and controller are all stable and all inputs are bounded. Brosilow did not consider an application of the perfect controller to a nonminimum phase process.

4.4 IMC vs Dynamic Matrix Control and Model Algorithmic Control

A strong similarity exists between dynamic matrix control[5], model predictive heuristic control[12] and the predictive IMC controller. Dynamic matrix control(DMC) was developed by Cutler and Ramaker in the late seventies. At the same time Richalet, Rault, Testud and Papon as well as Rouhani and Mehra[18] were developing model predictive

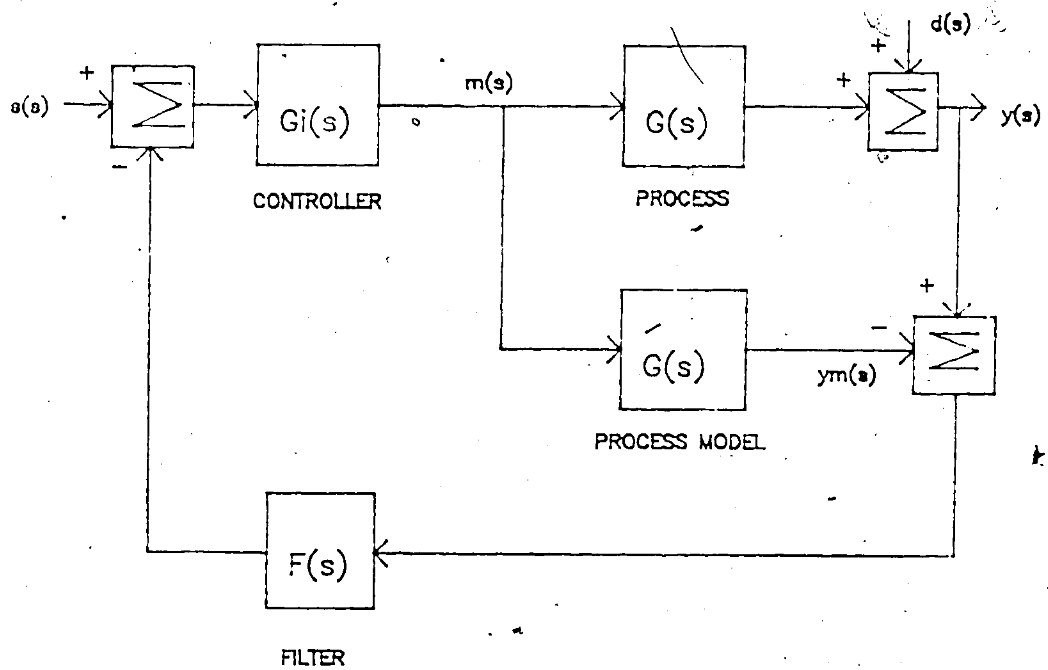

Figure 4.7 Block diagram of the inferential Smith predictor

heuristic control or as it has now been designated model algorithmic control(MAC).

All three strategies attempt to design a high performance linear optimal controller that also achieves significant closed loop robustness. The important unifying feature of these techniques is the use of nonparametric models to describe the process dynamics. Rouhani and Mehra emphasize the importance of the redundancy found in the nonminimal process description because it ensures accurate modelling of any nonlinearities in the process. The overparameterization also reduces the need for recursive on line identification. The disadvantages of the nonparametric description will be considered later in this chapter.

The MAC policy, like IMC, uses an impulse response description of the process dynamics. In DMC, the open loop response to a unit step input is used to represent the dynamics. Because all three algorithms approach controller design through least squares minimization of the weighted error between predicted output and setpoint the control laws are very similar. From equation L.5, in Appendix L, the DMC control law can be expressed as

$$\Delta m(k)=-[100\ldots0][A^tA+\alpha A']^{-1}A^tE \qquad \text{L.5}$$

and the equivalent IMC control law from equation C.6 is

$$m(k)=[1\ 0\ 0\ldots0_m][\Lambda^t\Gamma^t\Gamma\Lambda+B^tB]^{-1}\Lambda^t\Gamma^t\Gamma[E-\Psi V] \qquad \text{C.6}$$

The differences in these two equations are:

i. The positional form of the IMC control law versus the incremental form of the DMC control law. In IMC the signal sent to the controller is calculated from

$$m = m(k) + m(0)$$

where m(0) is the steady state control signal

In DMC the equivalent control signal is given by

$$m = \Delta m(k) + m(k-1) + m(0)$$

The positional form of IMC accounts for the presence of the ΨV term in equation C.6.

ii. The method of constraining the control signal.

Both algorithms constrain the control behavior through an ad hoc application of scalar weighting in the control laws. The DMC policy weights the diagonal elements of A^tA by a nonzero weighting factor, α. In equation L.5 the diagonal elements of A^tA are denoted by A'. In IMC, the ratio of β/γ is used to influence the control and output trajectories. Removal of the constraints from each formulation yields the minimum variance controller. The resulting expressions corresponding to DMC and IMC are

$$\Delta m(k)=-[100...0][A^tA]^{-1}A^tE \qquad 4.3.1$$

and

$$m(k)=[10...0][\Lambda^t\Lambda]^{-1}\Lambda^t[E-\Psi V] \qquad 4.3.2$$

These differences are very superficial since if DMC were to use an impulse response series instead of the step response series the two strategies would be identical.

The IMC technique differs significantly from DMC because of its comprehensive approach to process control. The IMC technique is applicable to not only minimum phase systems but also systems with time delay, with NMP behavior and with unstable dynamics. In addition, procedures are developed to handle all of the complexities accompanying the multivariable control problem. For example Garcia and Morari[2] clearly outline the procedure to control a multivariable process when the time delays are unbalanced. The authors of DMC have yet to develop policies to cope with time delay or NMP behavior. The multivariable DMC development considers only the trivial control problem of a minimum phase delay free multivariable process.

Although the examination of DMC has shown many similarities exist between DMC and IMC, the next material will show an even greater resemblance exists between MAC and IMC. The MAC control law, as derived in Appendix L, is

$$m(k)=[100\ldots0][\Lambda^t\Lambda]^{-1}\Lambda^t[E-\Psi V] \qquad \text{L.9}$$

where Λ, Ψ, and V are as defined for IMC and

$$E=|C(1-\alpha^{2j-1})-(y(k)(1-\alpha)-y_m(k))|$$

With the exception of the $(1-\alpha^{2j-1})$ factor that appears in the definition for E, this equation is identical to the IMC control law given by equation 4.3.2. For MAC, with α equal to zero the two equations are identical.

The MAC weighting parameter α shapes the closed loop response in a manner similar to the IMC filter, F(z). A difference however exists because the α weighting of MAC is explicitly included in the control law minimization but for IMC the α weighting is applied to the controller after minimization occurs.

In Chapters 2 and 3 in applying the IMC technique use of β and γ to shape the closed loop response for minimum phase systems was discouraged. Instead, the combination of minimum variance control or perfect control, with exponential filtering in the feedforward path, was promoted as the best design approach. This is virtually the same policy as MAC.

To control a process with NMP behavior, Rouhani and Mehra have incorporated linear quadratic control(LQC) theory into the MAC design. The control law for the NMP process, as derived in Appendix L, uses the LQC performance criterion, equation L.13, to choose a minimum phase approximation of the dynamics. This procedure is unique among the three nonparametric algorithms. The disadvantage associated with a larger, more demanding MAC-LQC computer algorithm is weighed against the advantage of a stable controller that provides optimal control.

A simple NMP system, under either MAC-LQC control or IMC control, is simulated to compare the ability of each controller to reject an unmeasured disturbance. The continuous process represented by the transfer function

$$y(s) = \frac{-1(s-.1)u(s)}{(s+.1)(s+1)} + d(s) \qquad 4.3.3$$

when sampled every four minutes gives the following discrete NMP model with a process zero at 1.69

$$y(k) = \frac{(-.467z^{-1}+.791^{-2})u(k)}{1-.689z^{-1}+.0123z^{-2}} + d(k) \qquad 4.3.4$$

Like the previous examples of this chapter, the closed loop response to an unmeasured disturbance, $d(k^{+})=1$, is examined. In Figure 4.8 the regulatory performance of the well tuned IMC controller is shown. The tuning parameters are

$N=30 \quad P=10 \quad M=8 \quad \alpha=0.6 \quad \Gamma=1.0 \quad B=0.0$

Figure 4.9 shows the response for the MAC-LQC controlled NMP system. To tune the MAC-LQC controller values for the scalar quanties, α, Q_2 and N are specified. The value for N, as in the IMC case, should be sufficient to minimize the truncation error. Increasing either α or Q_2 will reduce the speed at which the controller responds to setpoint changes or unmeasured disturbances. Correspondingly the dynamics of the manipulated variable usually improve as α or Q_2 are increased. In this situation, the closest, stable approximation to the minimum variance controller was desired. To realize this controller only a small value of Q_2 was used. The final choices were

$N=25 \quad \alpha=0.0 \quad Q_2=0.02$

The value for α was zero to minimize the output error measure. When Q_2 was reduced to 0.01 the closed loop response remained stable but the output error was greater. For this simple system the MAC-LQC strategy provides significantly better regulatory control with an output error measure of 23.7 than IMC with an output error measure of 28.1.

However, IMC unlike MAC, can be applied to the control of systems with time delays or to multivariable systems. Application of MAC to control multivariable systems has not

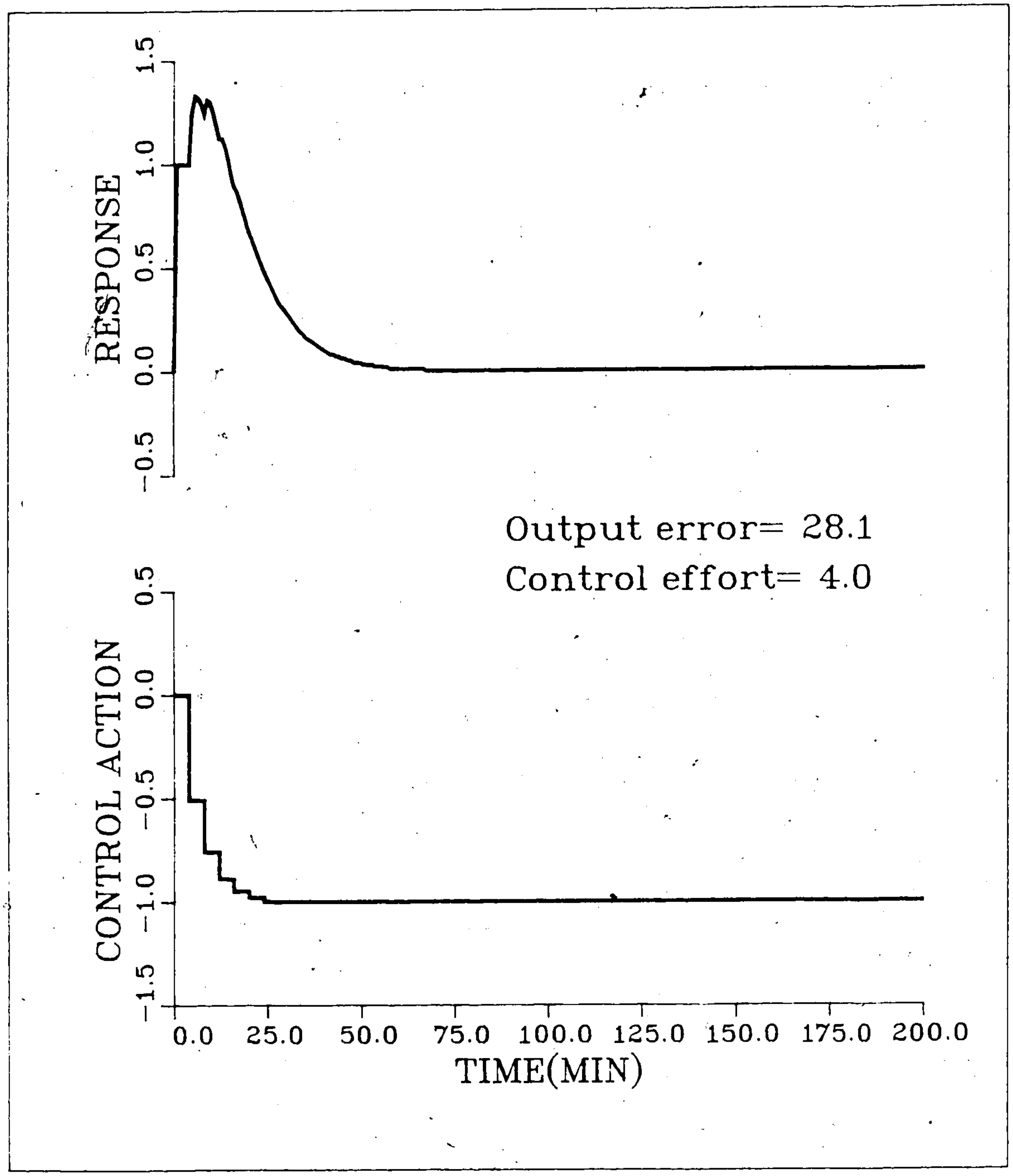

Figure 4.8 Response of the NMP system to an unmeasured disturbance of 1.0 under IMC control

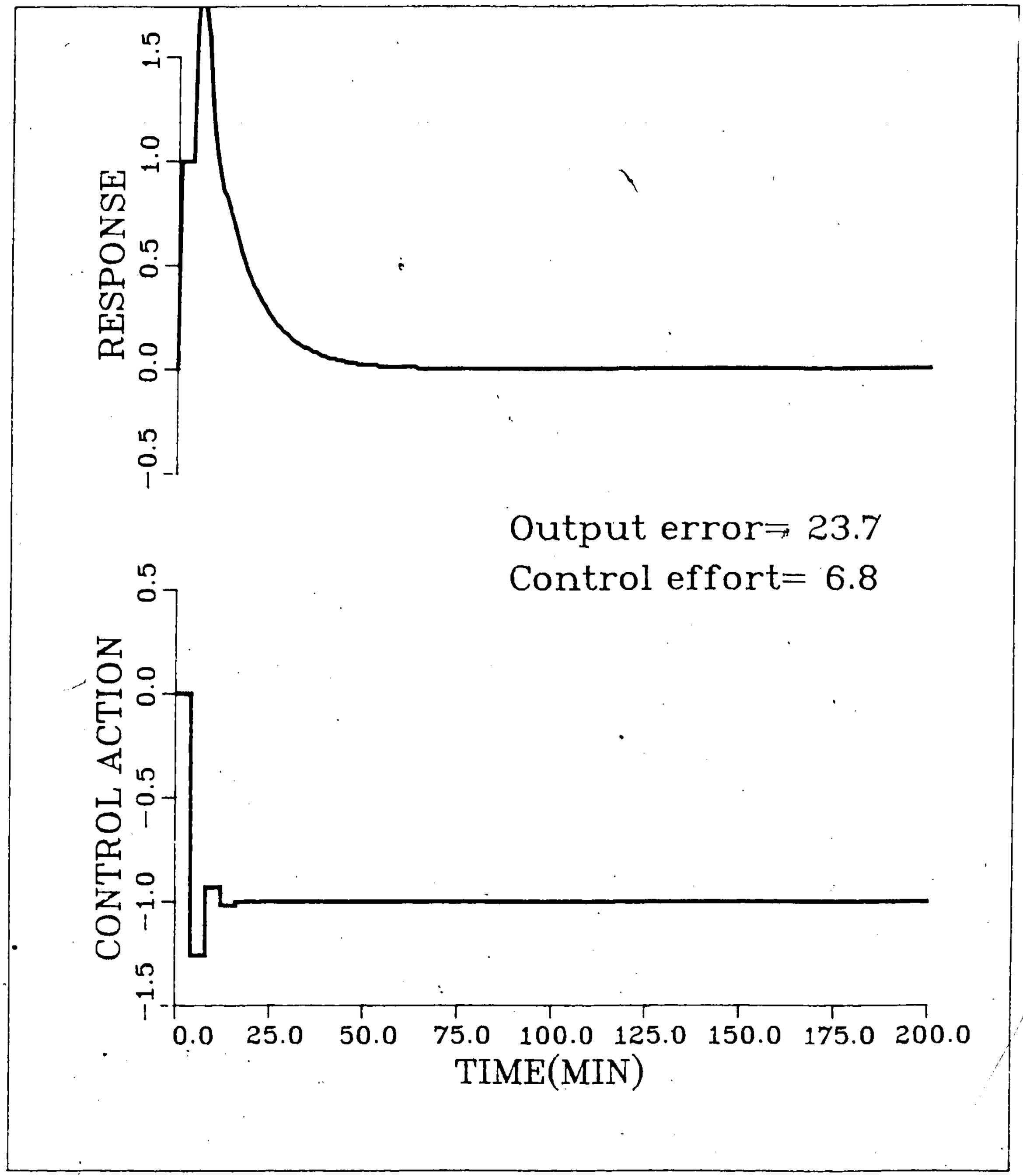

Figure 4.9 Response of the NMP system to an unmeasured disturbance of 1.0 under MAC-LQC control

been considered, nor has the additional complexity that time delay introduces to the formulation of single or multivariable LQC controllers.

4.5 IMC vs single variable self-tuning control

A considerable amount of study has been devoted to the control of systems using self-tuning controllers(STC). Although the adaptive nature of the self-tuning algorithm would suggest such controllers would have little in common h IMC, closer examiniation reveals many similarities. erent variations of STC are present in the literature so for ease of comparison the work of Clarke and Gawthrop[19] will be used to exemplify the SISO self tuning controller. A block diagram of this control strategy is illustrated in Figure 4.10.

The comparison of IMC to STC will focus on the design approach, model representation and performance criteria.

4.5.1 Design Approach

The self-tuning controller is designed to function in an adaptive manner. It has a simple control law to ensure compact computer implementation. Conversely, IMC is designed to operate as a model reference controller. Its control law can require extensive matrix operations not suited to the recursive structure of an adaptive algorithm. Both algorithms design feedforward controllers with some form of feedback to compensate for modelling errors and/or

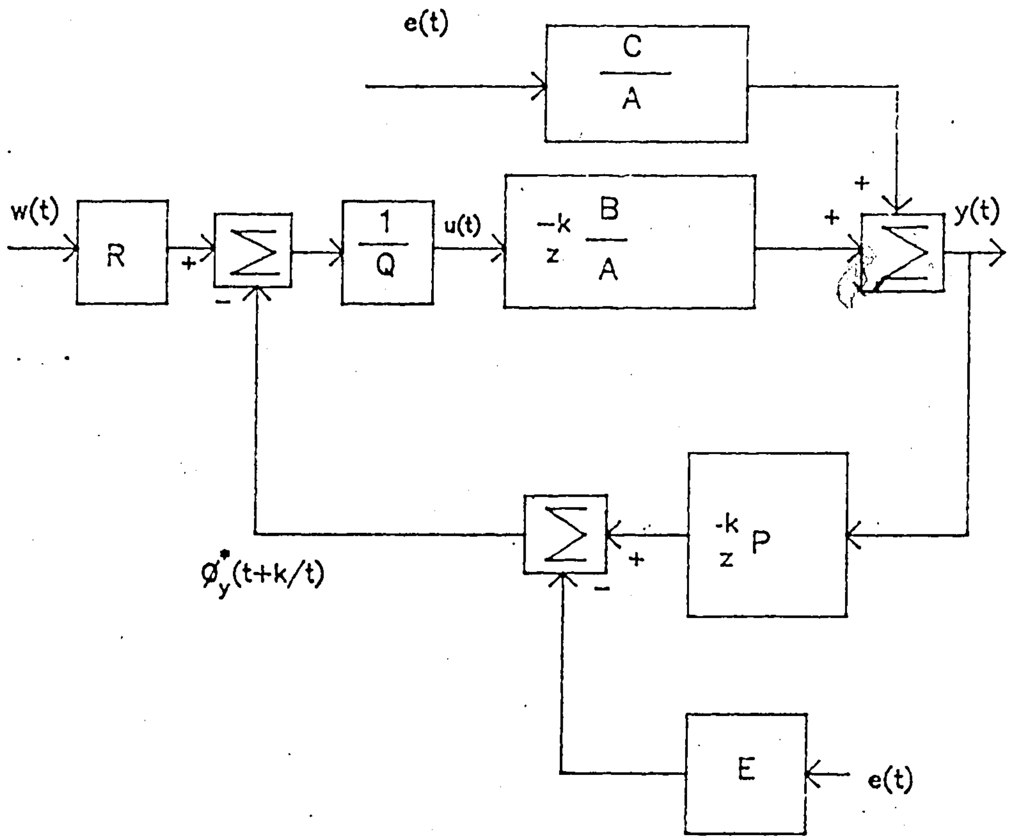

Figure 4.10 Block diagram of single variable self-tuning control

disturbances.

4.5.2 The Process Model

The method of representation of the process dynamics differs significantly in these two algorithms. In IMC an impulse response series is used to model the process dynamics while the STC technique uses an ARMA model to represent the process dynamics. ARMA models are minimal representations, requiring few parameters to model the input-output relationship. The process model is given by

$$A(z^{-1})y(k+\tau)=B(z^{-1})u(k)+C(z^{-1})\xi(k+\tau) \qquad 4.4.1$$

The poles of the process transfer function are given by the roots of polynomial $A(z)$. The zeros of the deterministic transfer function are given by the roots of polynomial $B(z)$. The polynomial $C(z)$ is used to represent any colouring of the process noise. When the noise is unbiased or white, $C(z)$ is set to unity. For the parameter identification to work $C(z)$ must have all roots inside the unit circle. In Chapter 2, it was noted that IMC considers noise modelling to be of secondary importance and as as result, no provisions are made to model these stochastic effects.

4.5.3 The Performance Criteria

The STC control law formulation is very different from the algorithms presented so far, because of this the control law derivation will be included in this section rather than being relegated to the appendix.

To incorporate output, control and setpoint dynamics into the controller design the cost function is chosen to minimize the variance in an auxiliary output $\phi(k)$, given by

$$\phi(k)=P(z^{-\pi})y(k)+Q(z^{-1})m(k-\tau)-R(z^{-1})w(k-\tau) \qquad 4.4.2$$

where the P, Q, and R are weighting transfer functions used to modify the output and setpoint trajectories. The (z^{-1}) notation will now be dropped for convenience. At time interval $k-\tau$, $Qm(k-\tau)$ and $Rw(k-\tau)$ are known. However at time k, $Py(k)$ must be predicted. By defining $\phi_y(k)=Py(k)$ and $\phi_y^*(k|k-\tau)$ as the prediction of $\phi_y(k)$ based upon information known at time $k-\tau$, the problem is reduced to finding the prediction of ϕ_y^*. Because the self-tuning controller uses both past inputs and past outputs in this prediction some extensive substitution operations are required to arrive at

$$C\phi_y^*(k|k-\tau)=\frac{F}{P_d}y(k-\tau)+Gm(k-\tau) \qquad 4.4.3$$

It should be noted that this equation does not include any stochastic effects in the prediction of ϕ_y^*. As a result the actual value of ϕ_y differs from ϕ_y^* by a noise correction term

$$\phi_y(k)=\phi_y^*(k|k-\tau)+E\xi(k) \qquad 4.4.4$$

To predict the output based upon past inputs and outputs the following identities have been used

$$\frac{CP_n}{AP_d}=E+z^{-k}\frac{F}{AP_d} \qquad 4.4.5$$

$$G=EB \qquad 4.4.6$$

The polynomials E, F and G are found from equations 4.4.5

and 4.4.6. Substituting equation 4.4.3 into equation 4.4.2 allows the value of ϕ_y^* to be predicted in terms of known quantities

$$\phi^*(k)=\frac{F}{CP_d}y(k-\tau)+\frac{G}{C}m(k-\tau)+Qm(k-\tau)-Rw(k-\tau) \qquad 4.4.7$$

The control law is found by choosing the control signal m(k) to set the τ step ahead prediction of $\phi^*(k+\tau|k)$ to zero. That is

$$m(k)=\frac{-\frac{F}{P_d}y(k)+CRw(k)}{G+CQ} \qquad 4.4.8$$

This control algorithm is implicit in nature because only the parameters of the control law polynomials F, G and C are recursively identified. An explicit algorithm would identify the parameters of the ARMA model given by equation 4.4.1 and then solve the identities found in equations 4.4.5 and 4.4.6 to arrive at the F and G polynomials. By identifying the control law parameters the number of computations at each sample interval is reduced.

Several simulation results are now presented to illustrate the advantages and disadvantages of each algorithm. The results in, Figures 4.11 and 4.12, show the MP process response when the deadbeat IMC and STC controllers are used to control the minimum phase process described by equation E.1.2 in Appendix E. These results are for IMC tuning parameters of

M=P=10 N=10 B=0.0 Γ=1.0

and deadbeat STC tuning parameters of

P=1 R=1 Q=0.0

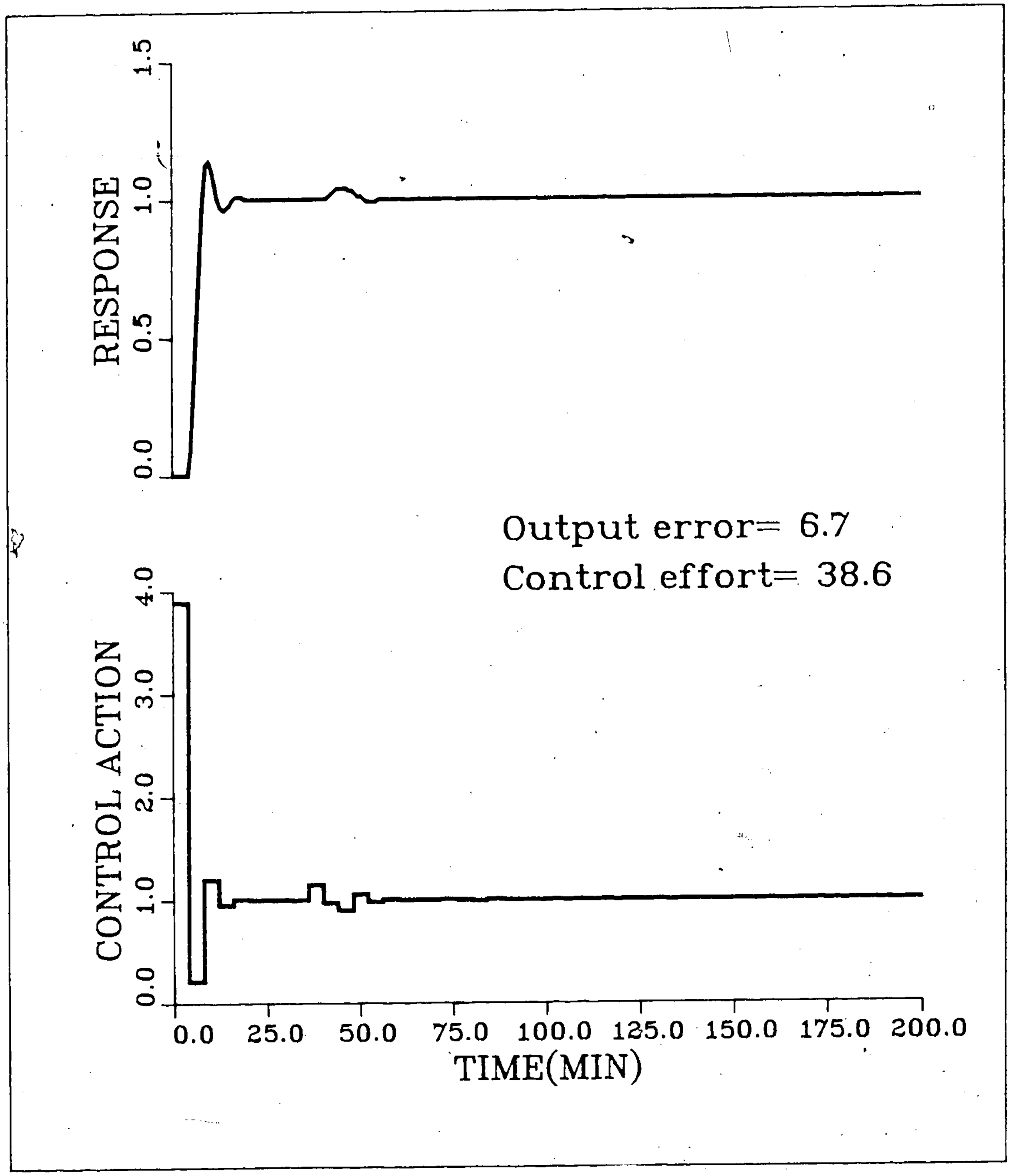

Figure 4.11 Response of the MP system to a positive step change in setpoint under IMC deadbeat control

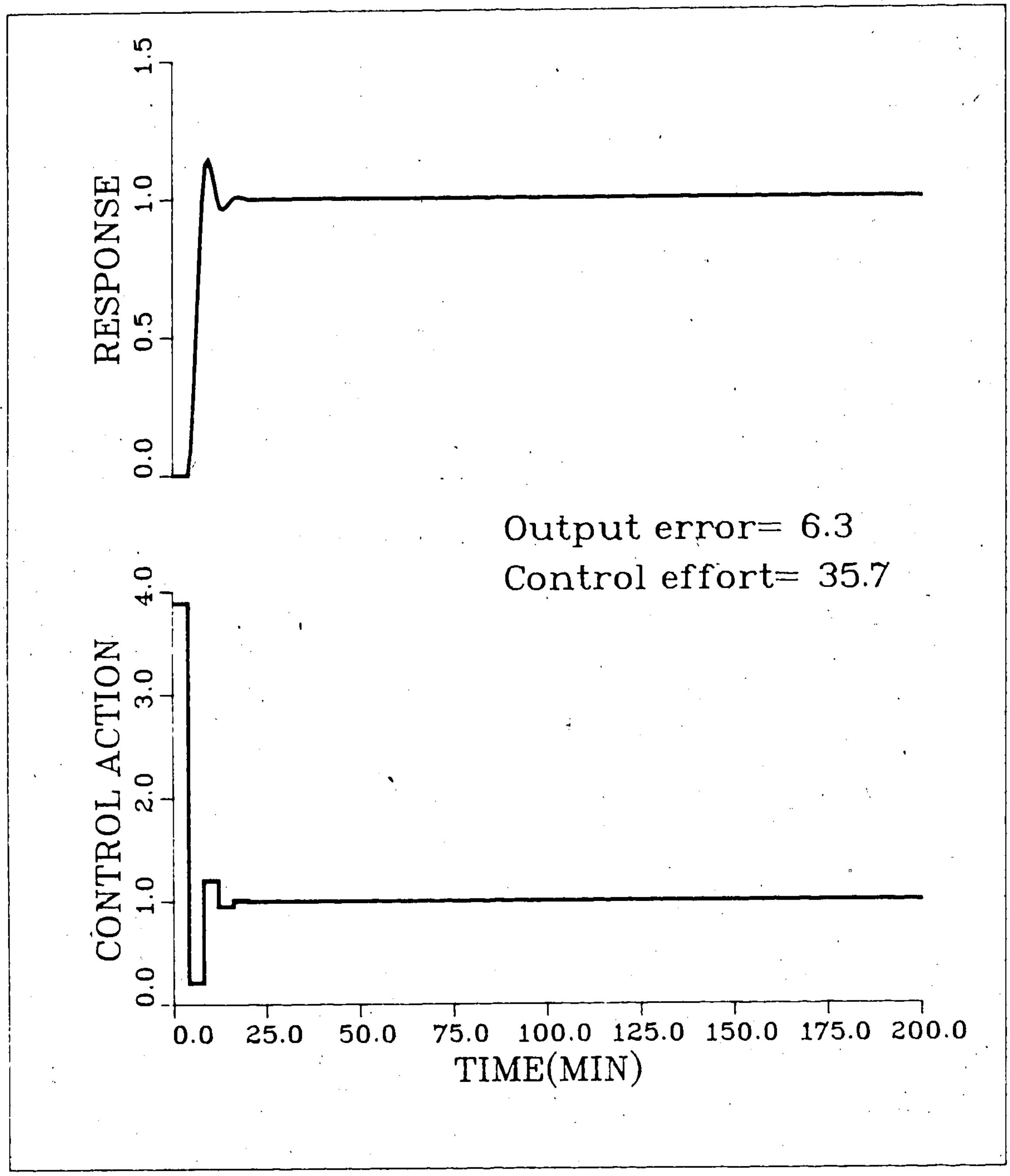

Figure 4.12 Response of the MP system to a positive step change in setpoint under STC deadbeat control

Both controllers are provided with an exact description of the process. The two responses are almost identical because both controllers have placed their poles and zeros to cancel the poles and zeros of the process. Unlike the STC response, a slight bounce occurs in the IMC response due to the truncation error.

The stability or robustness of such an adaptive algorithm may be difficult to ensure because the parameter estimation scheme may estimate unreliable parameter values due to an insufficiently exciting input signal. To prevent parameter blow-up, identification routines may be turned off when the output prediction agrees closely with the actual output. Use of a model reference controller strategy like IMC, removes the need for on-line parameter adaption as well as the extra coding required to ensure stability of the adaptive algorithm. The next results illustrate the disturbance rejection feature of the STC algorithm for the same MP system when an unmeasured disturbance, $d(0^+)=1.0$, occurs giving, $y(0^+)=1.0$. In Figure 4.13 because identification was turned off prior to the disturbance, the algorithm is unable to prevent offset. In the next figure, Figure 4.14, indentification is on when the disturbance occurs. The resulting response shows some initial oscillation while the disturbance is identified, after which the disturbance is completely eliminated. Compare Figure 4.14 where the output error is 22.5 and control effort 43.8 to Figure 4.3 where the feedback structure of IMC has

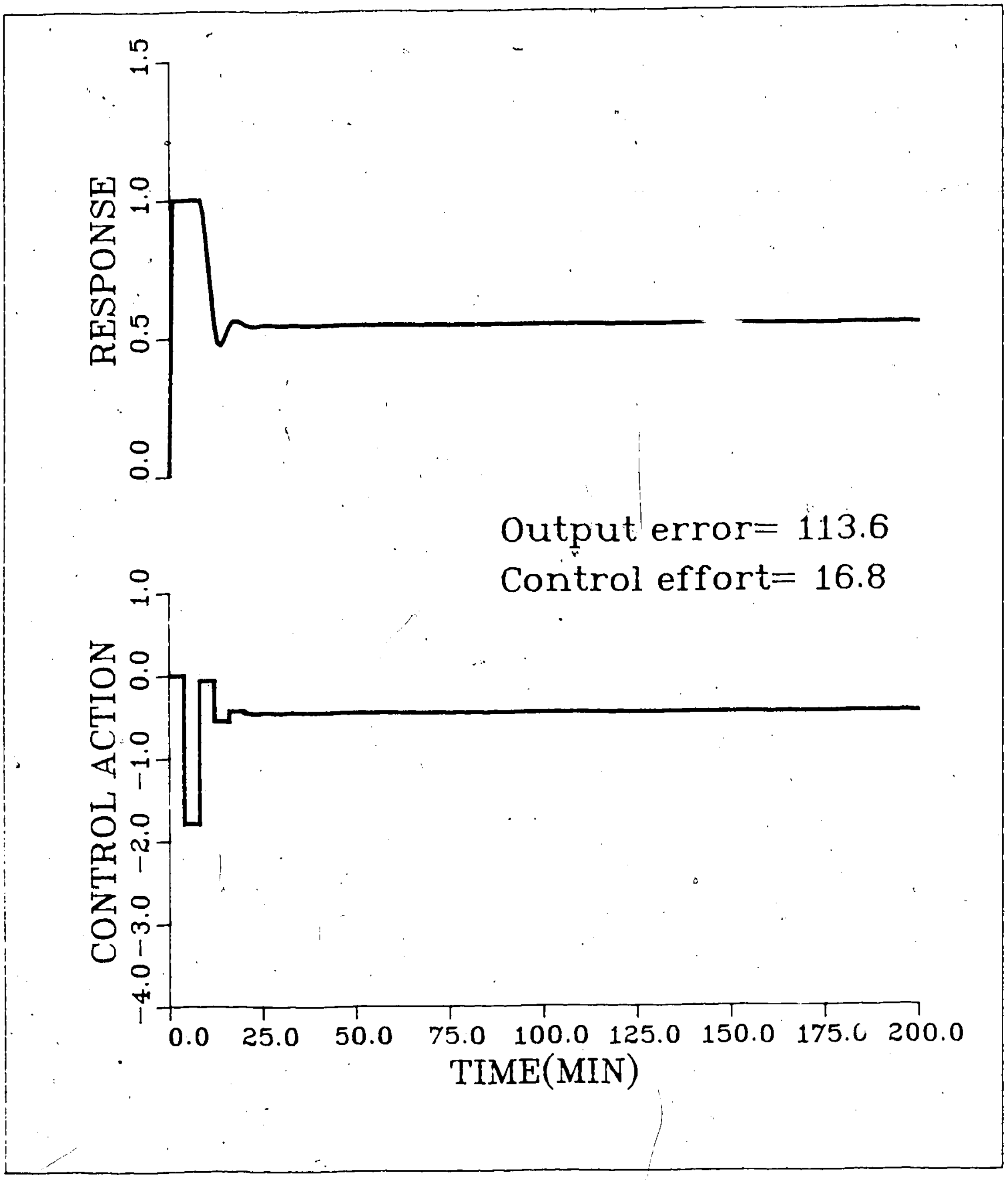

Figure 4.13 Response of the MP system to an unmeasured disturbance of 1.0 under STC control with no identification

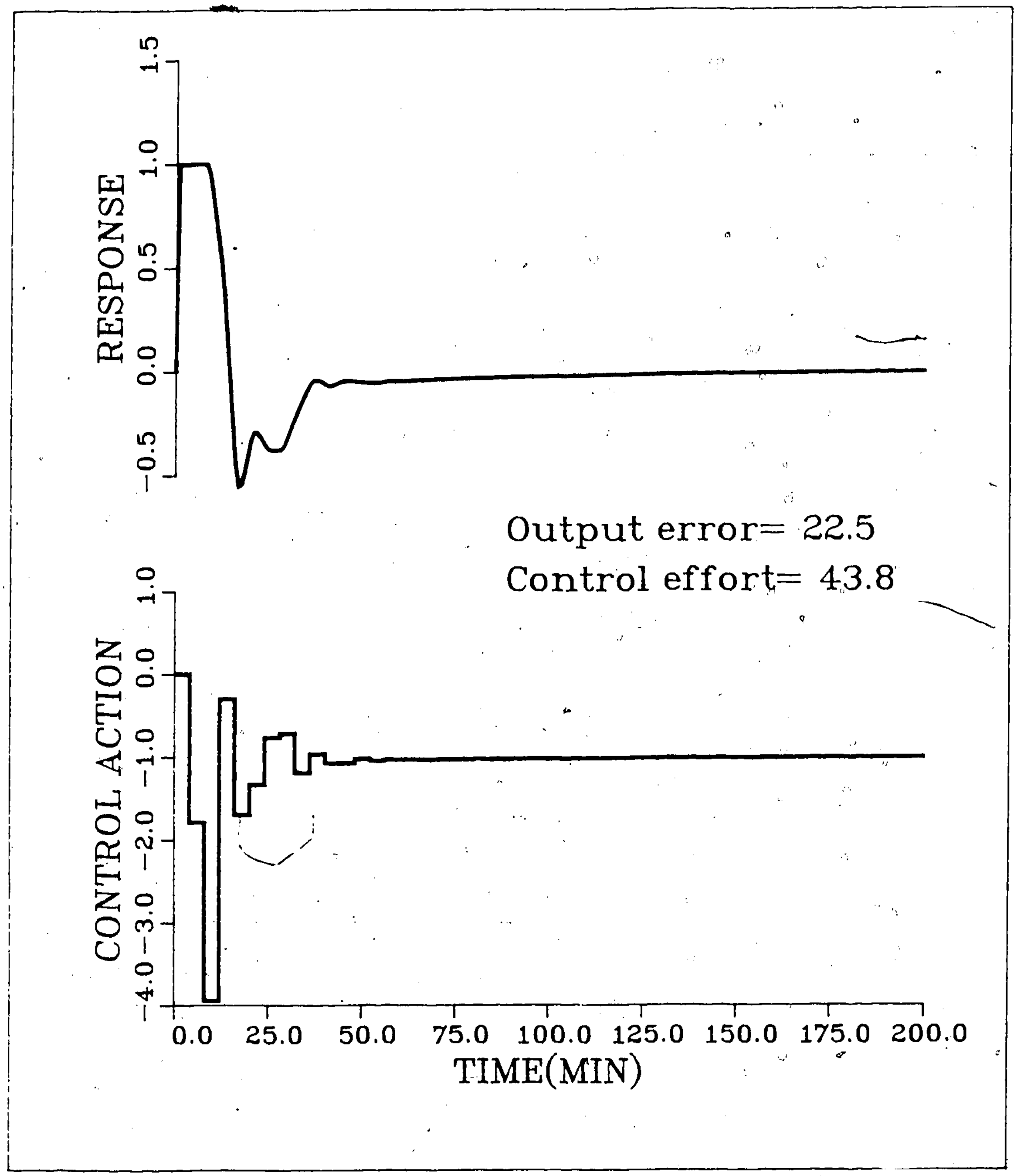

Figure 4.14 Response of the MP system to an unmeasured disturbance of 1.0 under STC control with identification

resulted in an output error of 10.6 and a control effort of 23.8.

Simulation of the NMP process, given by equation E.2.2, in Appendix E, demonstrates the advantage of the STC weighting technique over IMC. Figure 4.15 traces the response of the system to a unit step in setpoint using the STC controller with

P=1 R=2 Q=1

The process zero outside the unit circle is compensated by the Q weighting of 1.0. To prevent offset, the setpoint weighting, normally at one is increased by an amount equal to Q. Increasing R by a scalar amount in effect increases the value of the setpoint by a factor R. Trial and error showed, the best value for R, was 1+Q. Because IMC uses only β and γ to weight the control law the three parameter STC tuning procedure is not applicable. Figure 4.16 shows the respones of the well tuned NMP system to a positive step in setpoint using the IMC controller with the following param ters

M=[illegible] P=10 B=0.0 Γ=1.0

The ratio M/P was used to compensate the process zero outside the unit circle because it proved to be a better tuning combination than β/γ. As in the previous simulations where IMC has been applied to the NMP system, the controller was difficult to tune and provided questionable 'optimal' performance.

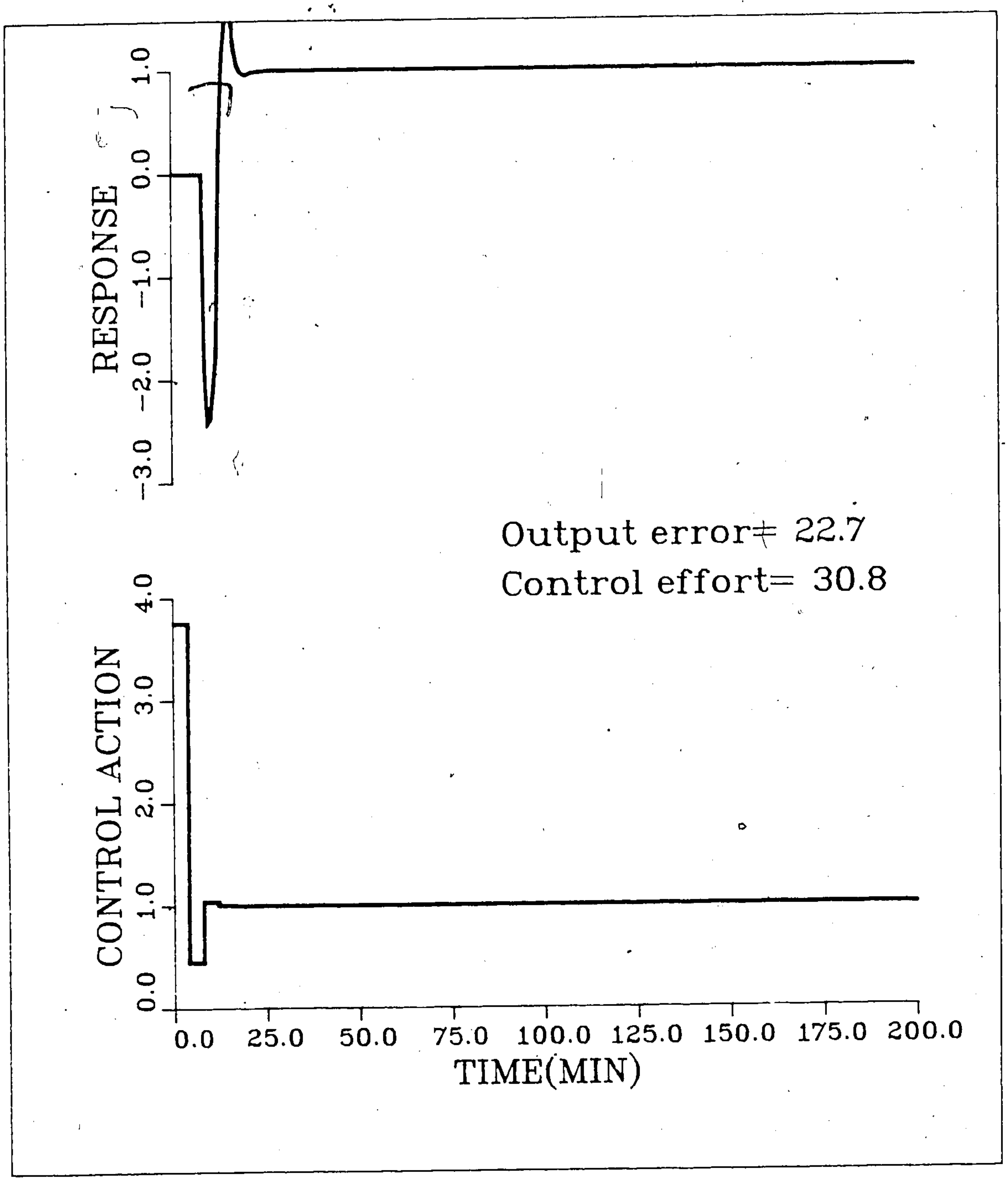

Figure 4.15 Response of the NMP system to a positive step change in setpoint under self-tuning control

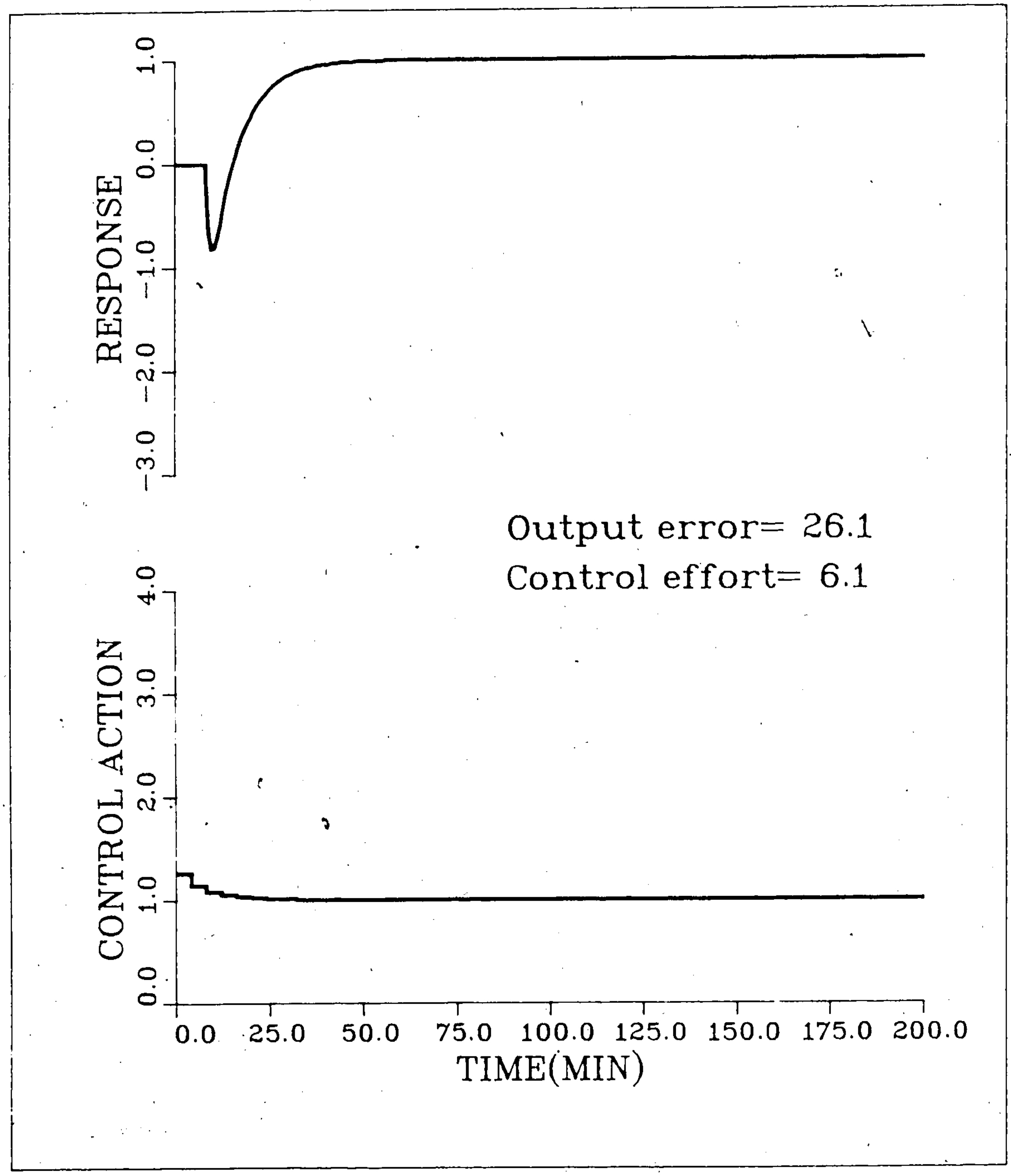

Figure 4.16 Response of the NMP system to a positive step change in setpoint under IMC control

All the process simulation results so far have used linear deterministic systems described by perfect models. To examine the effect of noise and plant/model mismatch on the IMC and STC controllers a nonlinear stochastic process simulator is used. A block diagram of the blending tank shown in Figure 4.17, and described in Appendix M, has nonlinear dynamics and a variable time delay. White noise of variance 0.01 is added to the process output to give a noise to signal ratio of roughly ten percent. A cross correlation procedure as outlined in Box and Jenkins[20] is used to develop an impulse response model of the relationship between inlet valve position and product concentration. The STC routine uses the recursive least squares(RLS), upper-diagonal factoization routine of Bierman[10] to identify the controller parameters.

The first set of results are for control of solute concentration using the IMC and STC algorithms when the tank steady state level is in the conical portion of the tank. Figures 4.18 and 4.19 show the response of the nonlinear process to a unit step in setpoint using the deadbeat IMC and STC controller respectively. Before the step, the solute concentration is 30.1 kg/m^3 and the height of solution in the tank is 0.4m but when the concentration has reached 31.1 kg/m^3, the solution level is 0.44m. These results were achieved using the IMC controller parameters of

$M=P=1 \quad N=20 \quad \alpha=0.0 \quad \gamma=1 \quad \beta=0$

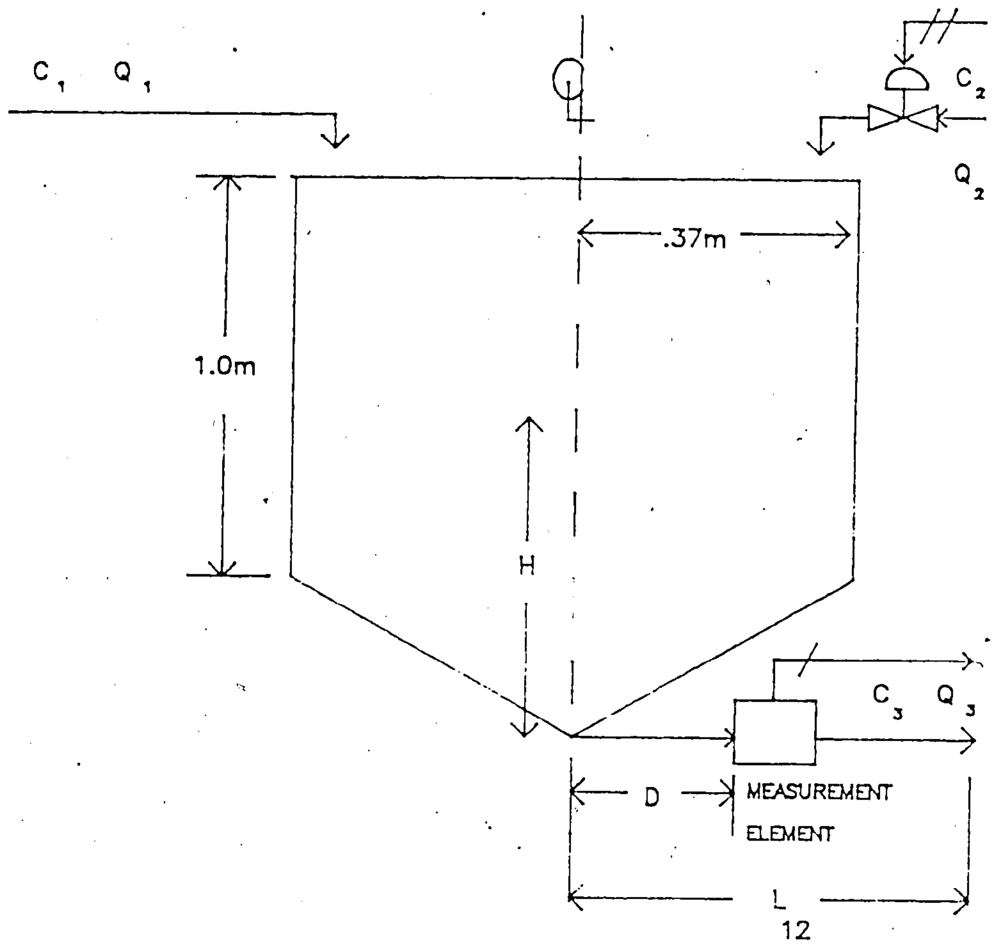

Figure 4.17 Block diagram of the nonlinear blending tank

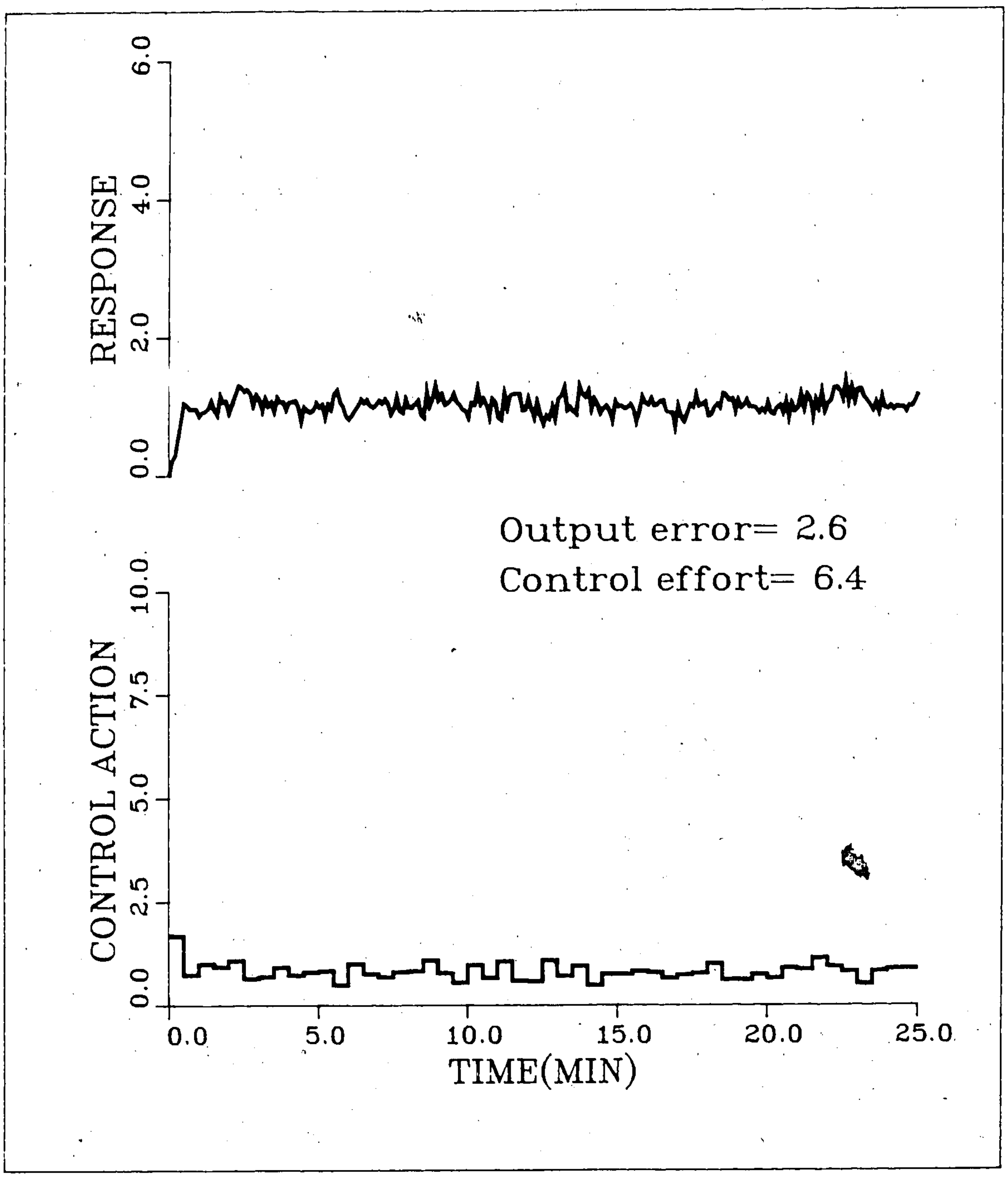

Figure 4.18 Response of the concentration and inlet flow rate for a positive step change in setpoint under IMC deadbeat control

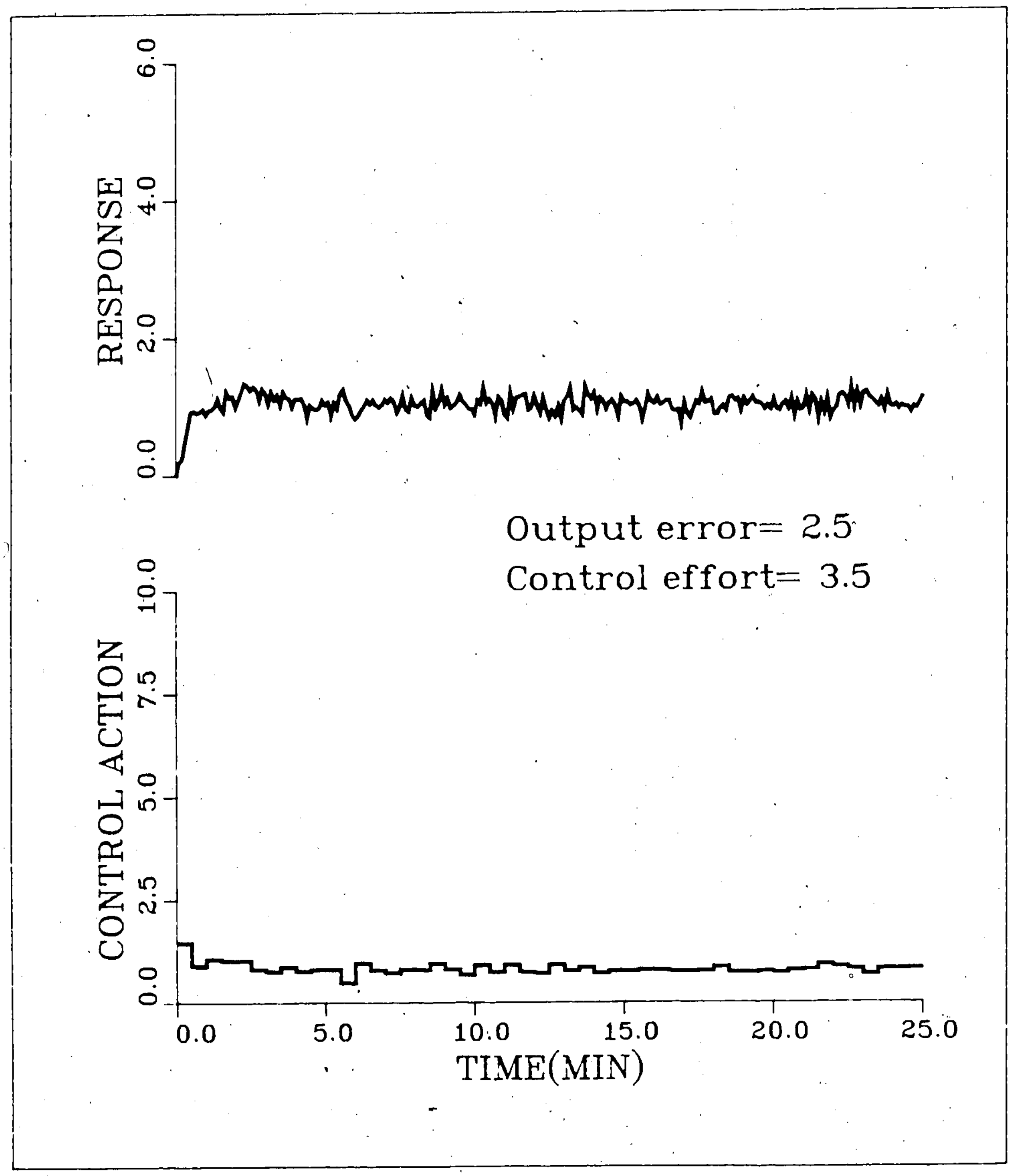

Figure 4.19 Response of the concentration and inlet flow rate for a positive step change in setpoint under STC deadbeat control

and the STC tuning parameters of

P=R=1 Q=0

Both algorithms exhibited the best performance when the transport delay was ignored even though the actual delay varied between .36 and .44 of a sampling unit. Also the self-tuning controller identified the control parameters on the basis of a first order linear model. The output error measure was almost the same for the two control algorithms, however control using the STC technique showed less oscillation in the manipulated variable, consequently the control effort measure for STC was much less than the corresponding IMC measure. For this small change in operating conditions the cross correlation procedure has identified a linear model of the nonlinear process that allows the IMC algorithm to control the blending tank as well as the STC algorithm.

When a large positive change in operating conditions occurs the IMC controlled process shows better transient dynamics than the STC controlled process. The responses in Figures 4.20 and 4.21 result for the same initial process conditions and controller parameters as in the previous example except that the process is responding to a change in setpoint of 4.0kg/m^3 rather than 1.0kg/m^3. The response of the adaptive STC controller shows overshoot promptly followed by tight regulatory control about the new setpoint. The IMC algorithm moves the process to the new setpoint using a stationary model to predict the process output and

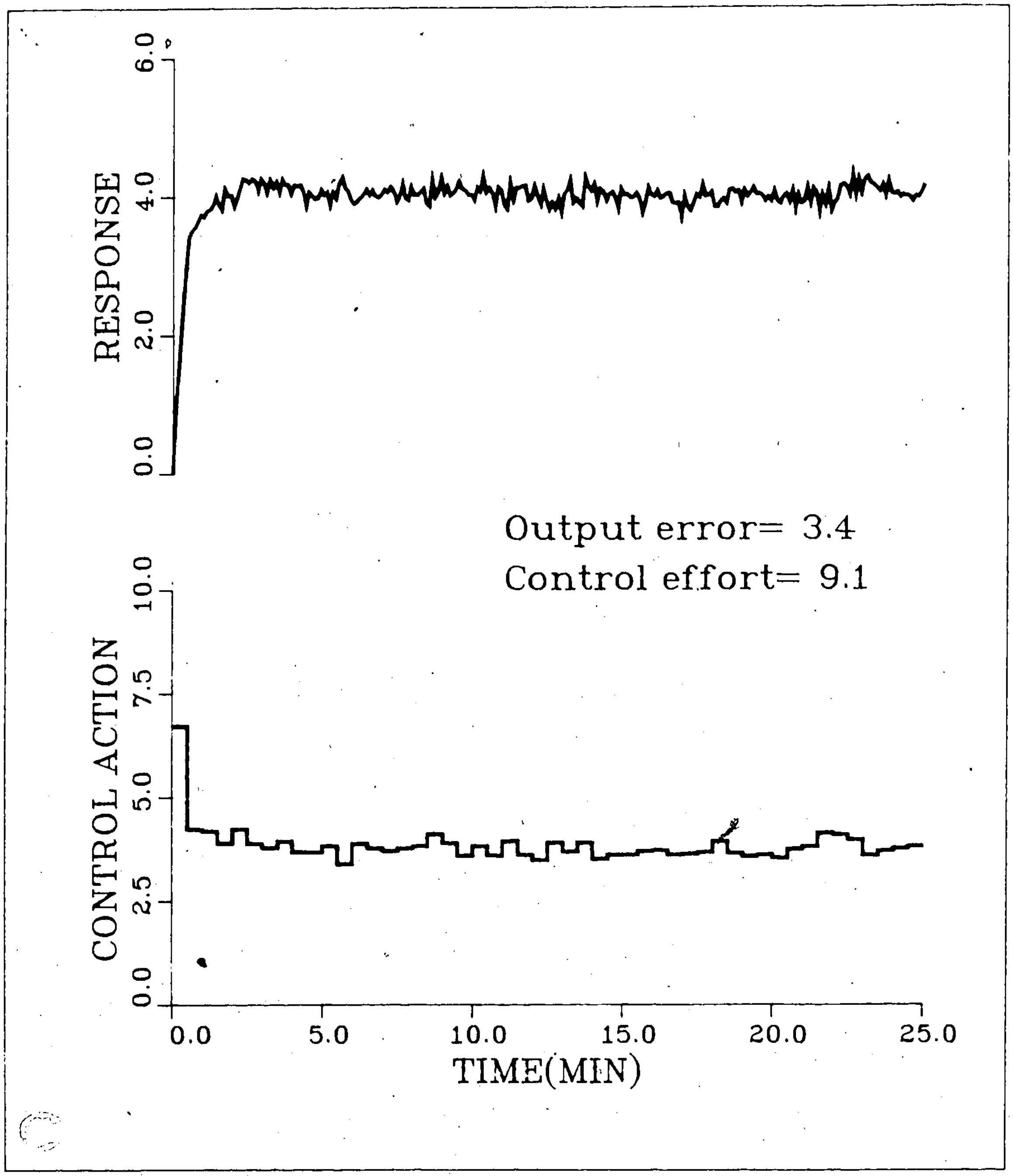

Figure 4.20 Response of the concentration and inlet flow rate to a positive step change in setpoint under deadbeat IMC control

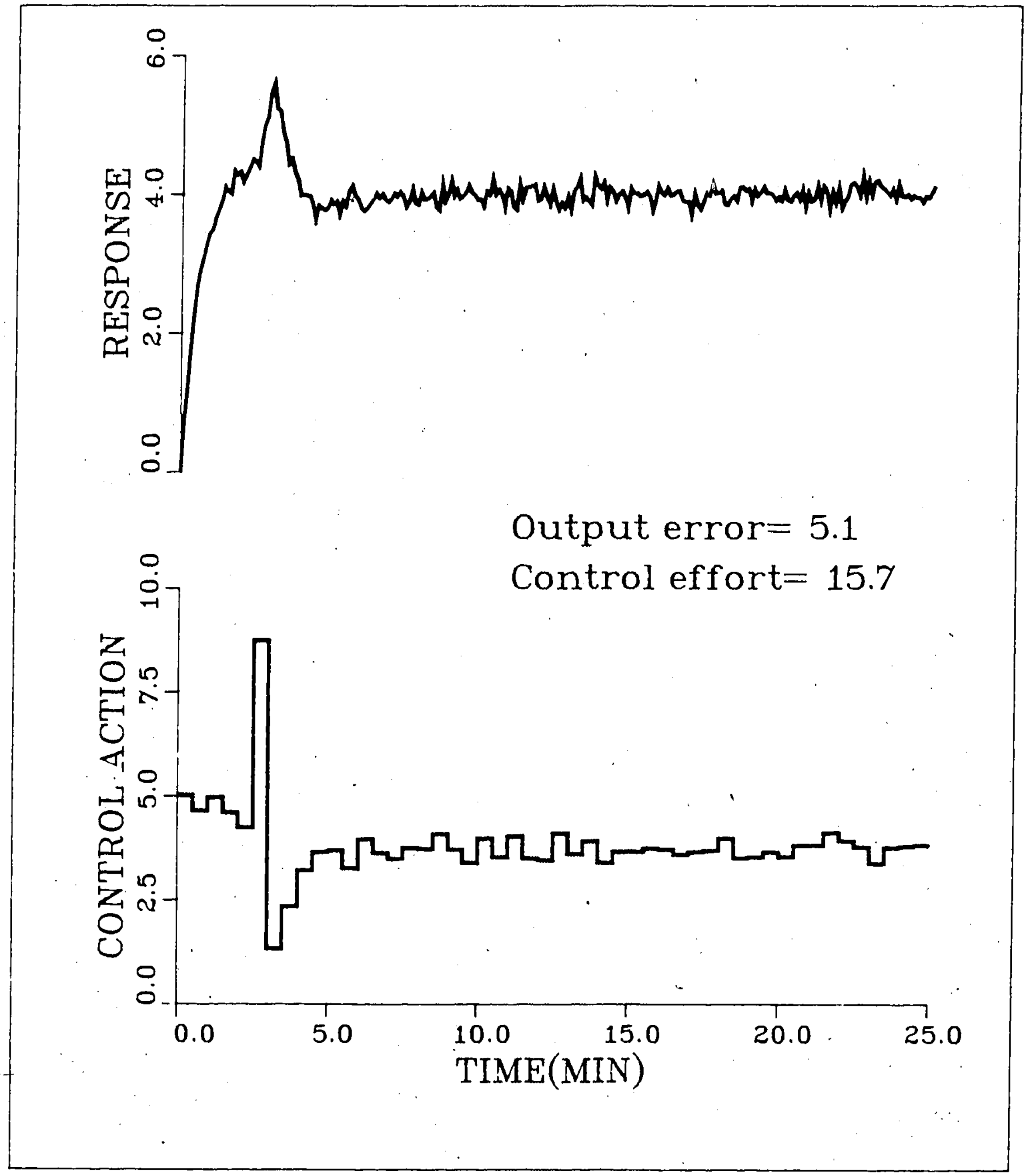

Figure 4.21 Response of the concentration and inlet flow rate to a positive step change in setpoint under STC deadbeat control

as a result the undesirable overshoot caused by parameter identification in the STC algorithm is gone, resulting in lower output error and control effort measures.

To further substantiate these observations, negative steps in process operating conditions are simulated when the initial steady state liquid level is 0.6m and the product concentration is 33.8 kg/m³. These initial conditions correspond to the final conditions for the positive step of 4.0 kg/m³ in setpoint. The controllers are tuned to the same setting as above. Cross correlation is carried out to identify an impulse response model for the IMC program at these operating conditions. Figures 4.22 and 4.23 show the process response to a setpoint change from 33.7 kg/m³ to 32.7kg/m³ using the IMC and STC controllers respectively. This magnitude of setpoint change brings the tank liquid level to the very bottom of the cylindrical section. The response of the nonlinear process to a setpoint change from 33.7 kg/m³ to 29.8 kg/m³ using the IMC and STC controllers is shown in Figures 4.24 and 4.25. The negative steps in operating conditions have affected the ability of IMC to control the nonlinear process. For the small negative step, the dynamics of the manipulated variable under IMC control are very undesirable. The closed loop response of the IMC controlled process becomes unstable for the case where a large negative step in operating conditions occurs. The larger error between the process and the model for the negative changes in operating conditions is responsible for

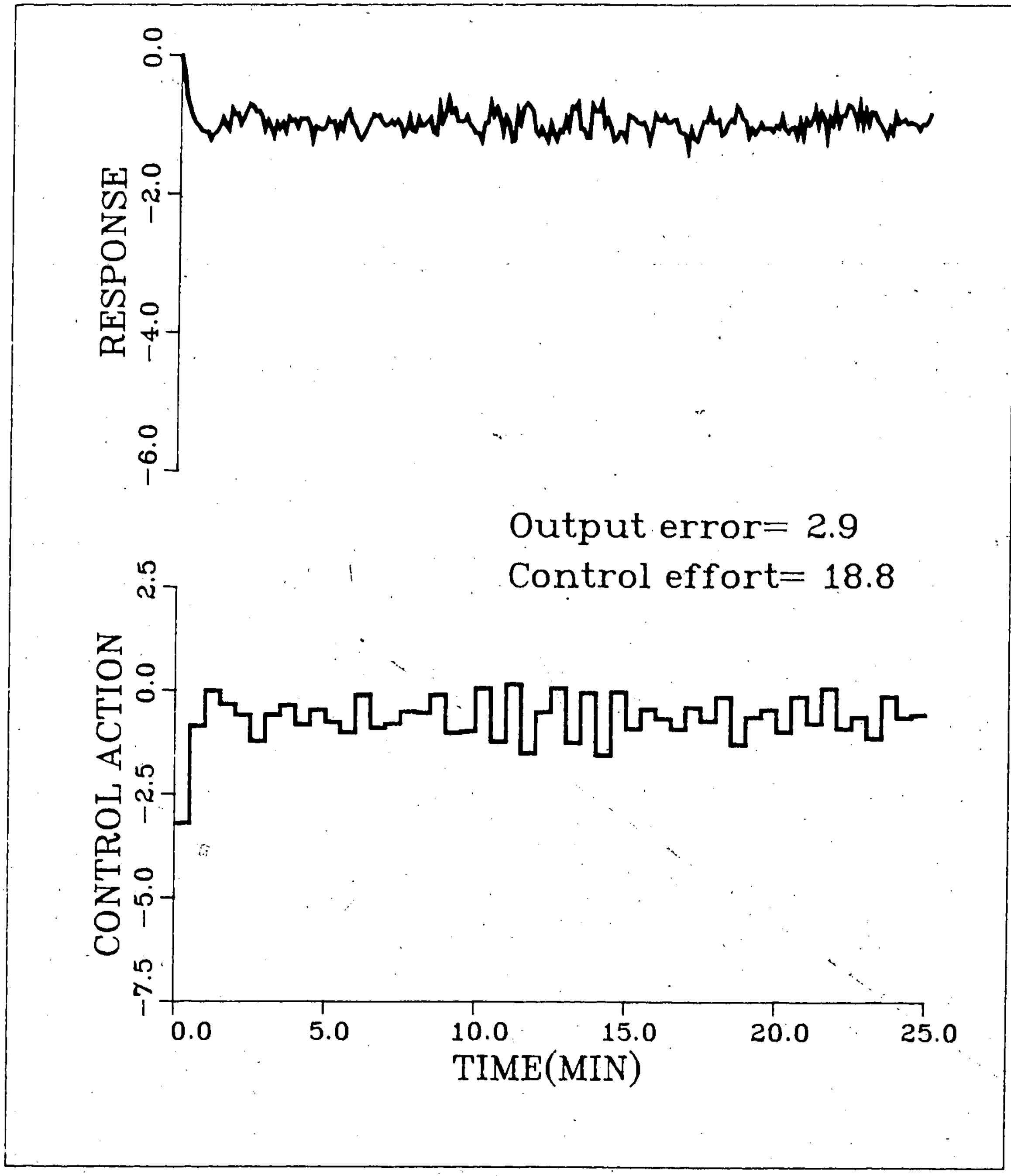

Figure 4.22 Response of the concentration and inlet flow rate to a negative step change in setpoint under IMC deadbeat control

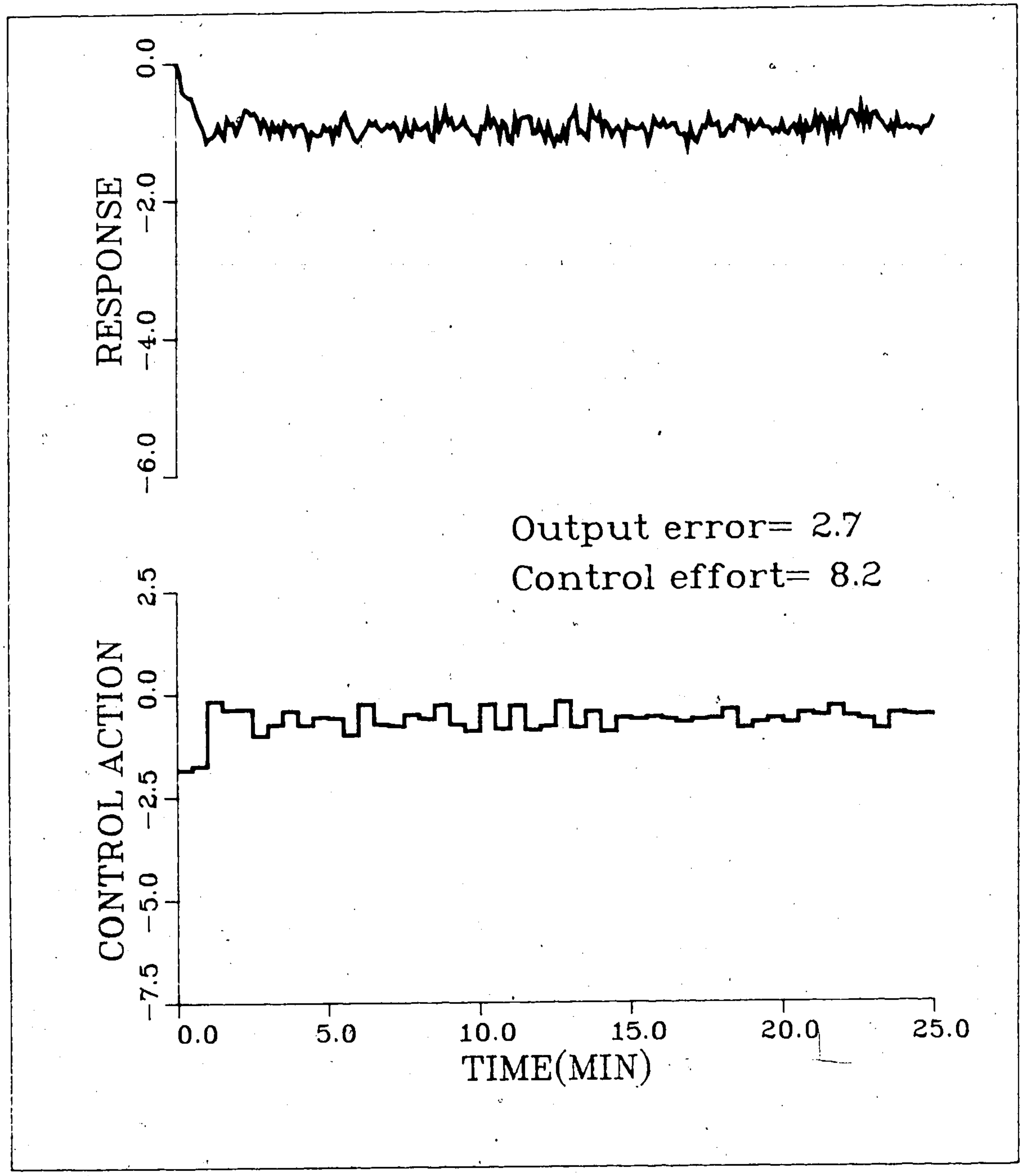

Figure 4.23 Response of the concentration and inlet flow rate to a negative step change in setpoint under STC deadbeat control

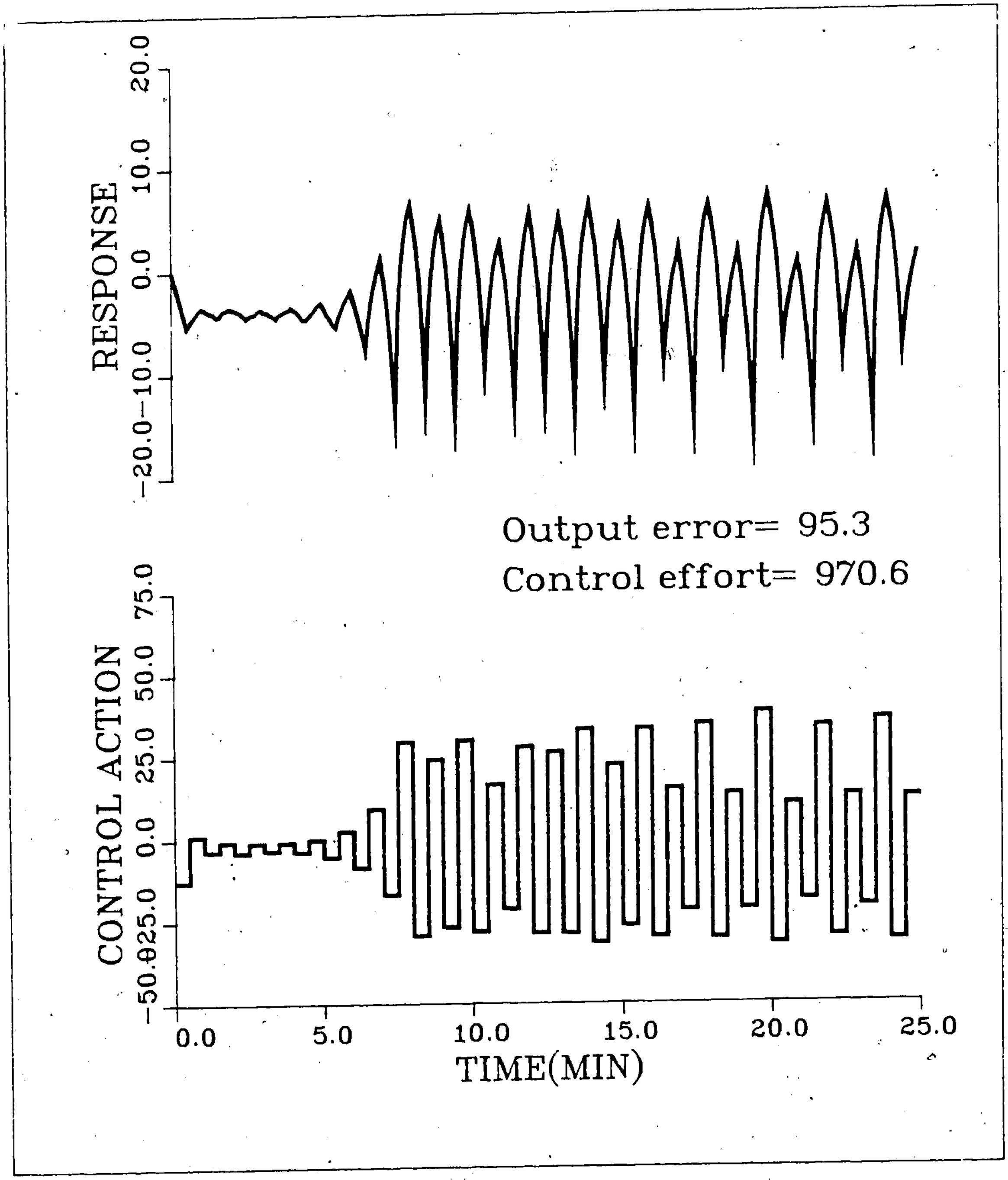

Figure 4.24 Response of the concentration and inlet flow rate to a negative step change in setpoint under deadbeat IMC control

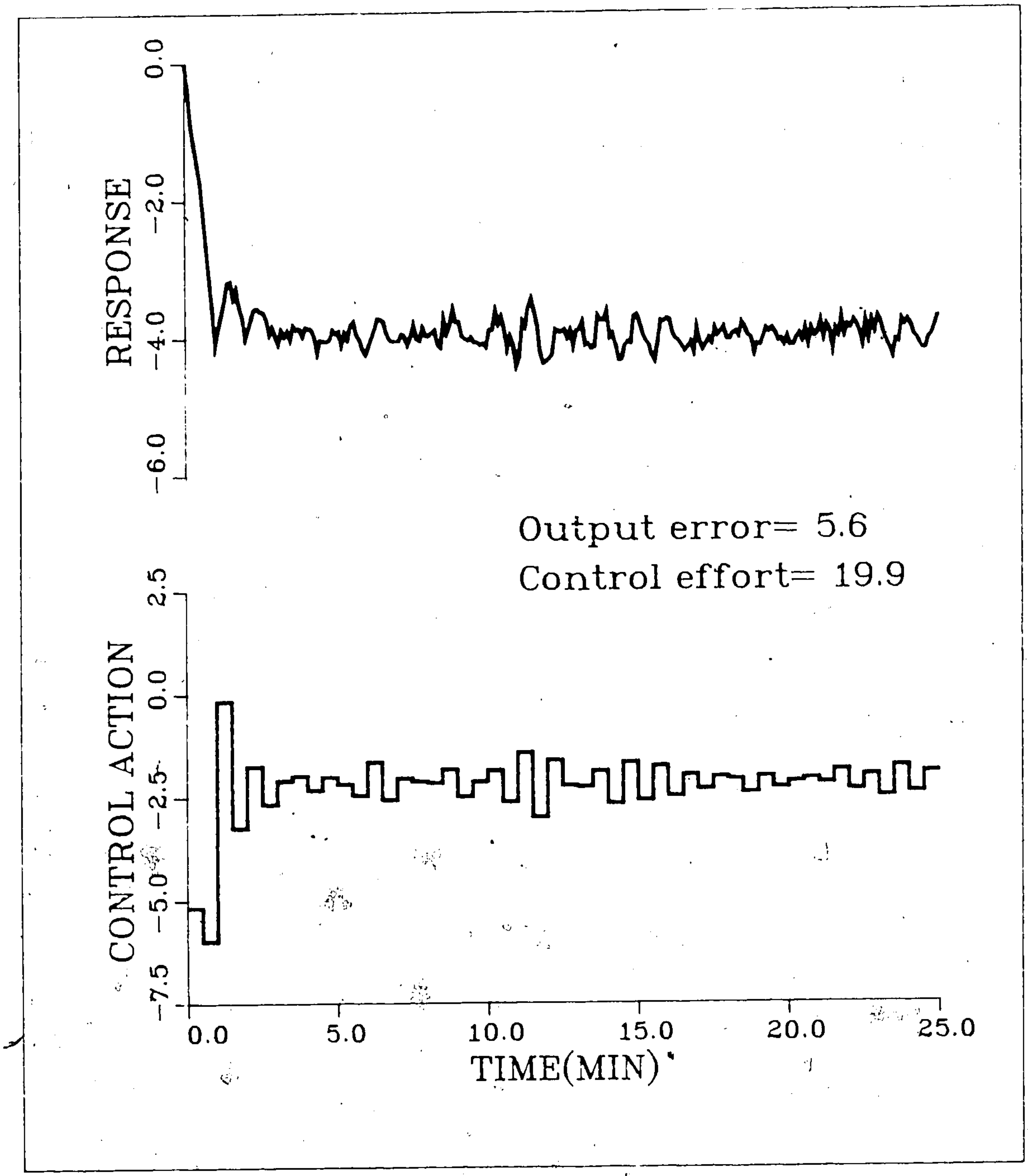

Figure 4.25 Response of the concentration and inlet flow rate to a negative step change in setpoint under STC deadbeat control

the poor responses.

It should be noted that the performance of the IMC controller did not deteriorate due to stochastic effects. Futhermore, application of the cross correlation procedure lead to the identification of an unbiased linear model approximation of the underlying deterministic process.

4.6 IMC vs The multivariable Smith predictors

Many multivariable process operations, such as distillation, cannot be satisfactorily controlled by use of multiple, single variable control loops. For these processes it is desirable to use multivariable controllers to achieve good regulatory and servo control.

Analogous to the single variable control problem, multivariable time delay compensation schemes are desirable because such strategies remove time delay from the closed loop characteristic equation facilitating higher controller gains. In 1973 Alevisakis and Seborg[21] introduced a multivariable single delay Smith predictor to eliminate a single delay from the multivariable closed loop characteristic equation. Like the original Smith predictor, the Alevisakis and Seborg predictor combines the time delay compensator with conventional two and three mode controllers.

In 1979 Ogunnaike and Ray[14] designed a multivariable, multidelay, Smith predictor. This predictor extended the work of Alevisakis and Seborg from a single delay to a multiple delay compensator. A block diagram representation of the Ogunnaike and Ray(OR) compensator is shown in Figure 4.26. From this figure the OR compensator closed loop transfer function is

$$\underline{y}(k)/\underline{s}(k)=\underline{G}(z)\underline{C}(z)[\underline{I}+\underline{C}(z)(\underline{G}(z)+\acute{\underline{G}}^{+}(z)-\acute{\underline{G}}(z))]^{-1} \quad 4.5.1$$

The characteristic roots are found from the determinant of the return difference operator

$$\det|\underline{I}+\underline{C}(z)[\underline{G}(z)+\acute{\underline{G}}^{+}(z)-\acute{\underline{G}}(z)]|=0 \quad 4.5.2$$

For a correct process model equation 4.5.2 reduces to

$$\det|\underline{I}+\underline{C}(z)\acute{\underline{G}}^{+}(z)|=0 \quad 4.5.3$$

Multivariable controller design techniques that require a delay free process description, can now be applied. A controller $\underline{C}(z)$ could be designed by assuming the closed loop transfer function was given by

$$\underline{y}(z)/\underline{s}(z)=\acute{\underline{G}}^{+}\underline{C}(z)[\underline{I}+\acute{\underline{G}}^{+}(z)\underline{C}(z)]^{-1} \quad 4.5.4$$

Garcia and Morari[2] show that this form of time delay compensation is nonoptimal. It was mentioned in Section 3.2 that the optimal multivariable controller or the multivariable controller that minimizes the least-square error was equivalent to the inverse of the realizable process transfer function matrix. As a result the IMC algorithm does not remove all the delays from the characteristic equation unless the delays in each row of the transfer function matrix are equal. By basing the controller design upon limits set by $\acute{\underline{G}}_{+1}(z)$, optimal performance can be achieved when $\acute{\underline{G}}_{+1}(z)$ is optimal.

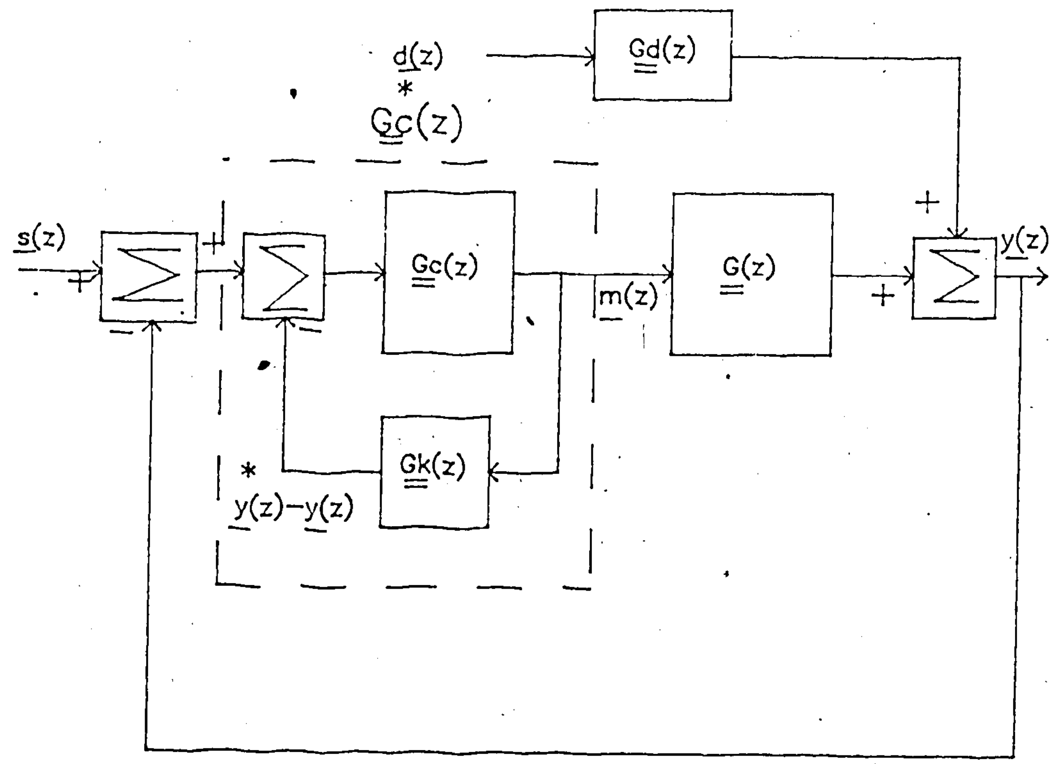

Figure 4.26 Block diagram of the Ogunnaike and Ray multivariable time delay compensator

To demonstrate the performance of IMC compared to the OR compensator with diagonal PI controllers(OR-PI) results from the simulation of two different examples are given. Both examples use the distillation column model of Wood and Berry[22], as given in Appendix N. Ogunnaike and Ray[14] have demonstrated the superior performance of this model under OR-PI control relative to the column model under multiple, single variable PI controllers without any time delay compensation. In the IMC and OR-PI experiments, the techniques are evaluated with respect to servo control for a step in the setpoint of the overhead product from 96.25 to 97.0 mass percent methanol when the column is initially operating at steady state conditions. The distillation column transfer function model when sampled at one minute gives rise to the following discrete representation

$$\underline{G}(z)=\begin{vmatrix} \frac{0.744z^{-2}}{1-.942z^{-1}}, & \frac{-0.879z^{-4}}{1-.954z^{-1}} \\ \frac{0.579z^{-8}}{1-.912z^{-1}}, & \frac{-1.301z^{-4}}{1-.933z^{-1}} \end{vmatrix}$$

Since $\underline{G}(z)$ is a balanced minimum phase transfer function matrix, the IMC time delay compensator will be optimal. The controller parameters, chosen to give complete decoupling, without filtering are

$N=30 \quad P=M=1 \quad \underline{\acute{G}}_{+1}(z)=\text{diag}(z^{-1},z^{-3}) \quad \gamma_j=0.0 \text{ for } j=1,P$

$\beta_j=0.0 \text{ for } j=1,M \quad \alpha=0.0\ 0.0$

The control performance of the column using this controller is shown in Figure 4.27. The perfect process model without time delay mismatch gives complete decoupling with deadbeat

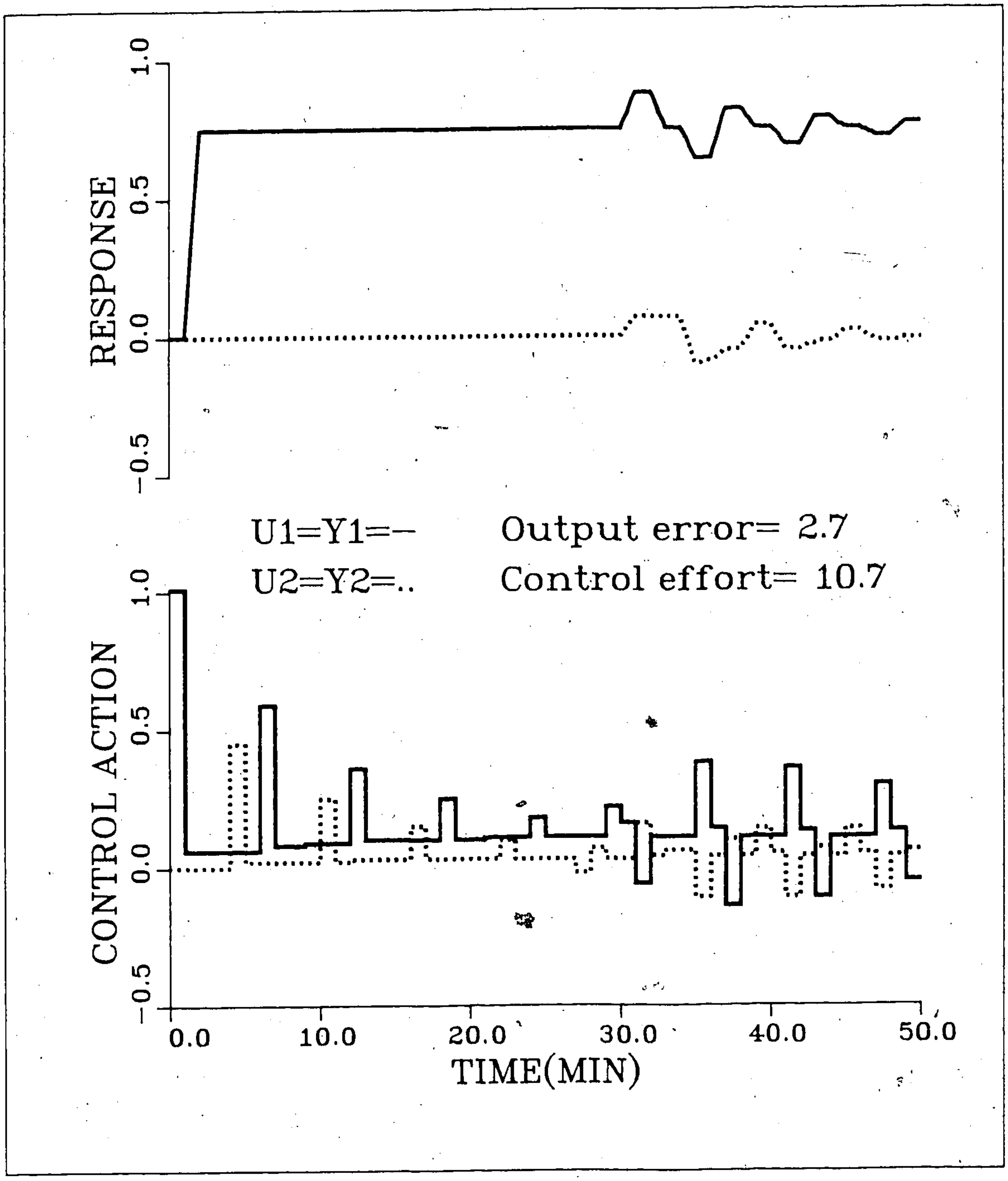

Figure 4.27 Response of both product compositions, with balanced time delays, to a setpoint change of 0.75 in overhead composition, y_1, under IMC

response over the first N sampling but then truncation error causes some oscillation in both output variables.

The 'velocity' form of the PI controller expressed as

$$m(k)=m(k-1)+Kce(k)-Kc\phi e(k-1) \qquad 4.5.5$$

was implemented in evaluation of the OR compensator in order to eliminate the need to use reset windup protection. This controller was used for each of the PI controllers on the diagonal elements of $\underline{C}(z)$. These controllers were tuned for the delay free process to provide deadbeat response of the diagonal elements of the open loop transfer function $\acute{\underline{G}}(z)\underline{C}(z)$. Following the guidelines given by Ogunnaike and Ray[14] the deadbeat tuning parameters were reduced in magnitude to compensate for the effect of time delay and off-diagonal element interaction. It was found, reducing the deadbeat gain by a factor of .5, produced the least output error. The final choices for Kc and ø were

element	Kc	ø
1,1	0.670	0.942
2,2	0.385	0.933

It should be noted that the OR compensator also includes provisions for a multivariable steady state offset compensator. This compensator $\acute{\underline{G}}_i(1)$, is designed to satisfy $\acute{\underline{G}}(1)\acute{\underline{G}}_i(1)=\underline{I}$. When this compensator was implemented it had no observable effect on the simulation result. The column response for the OR-PI combination is shown in Figure 4.28. The obvious difference between the control performance under IMC and with the OR-PI control strategies is the inability

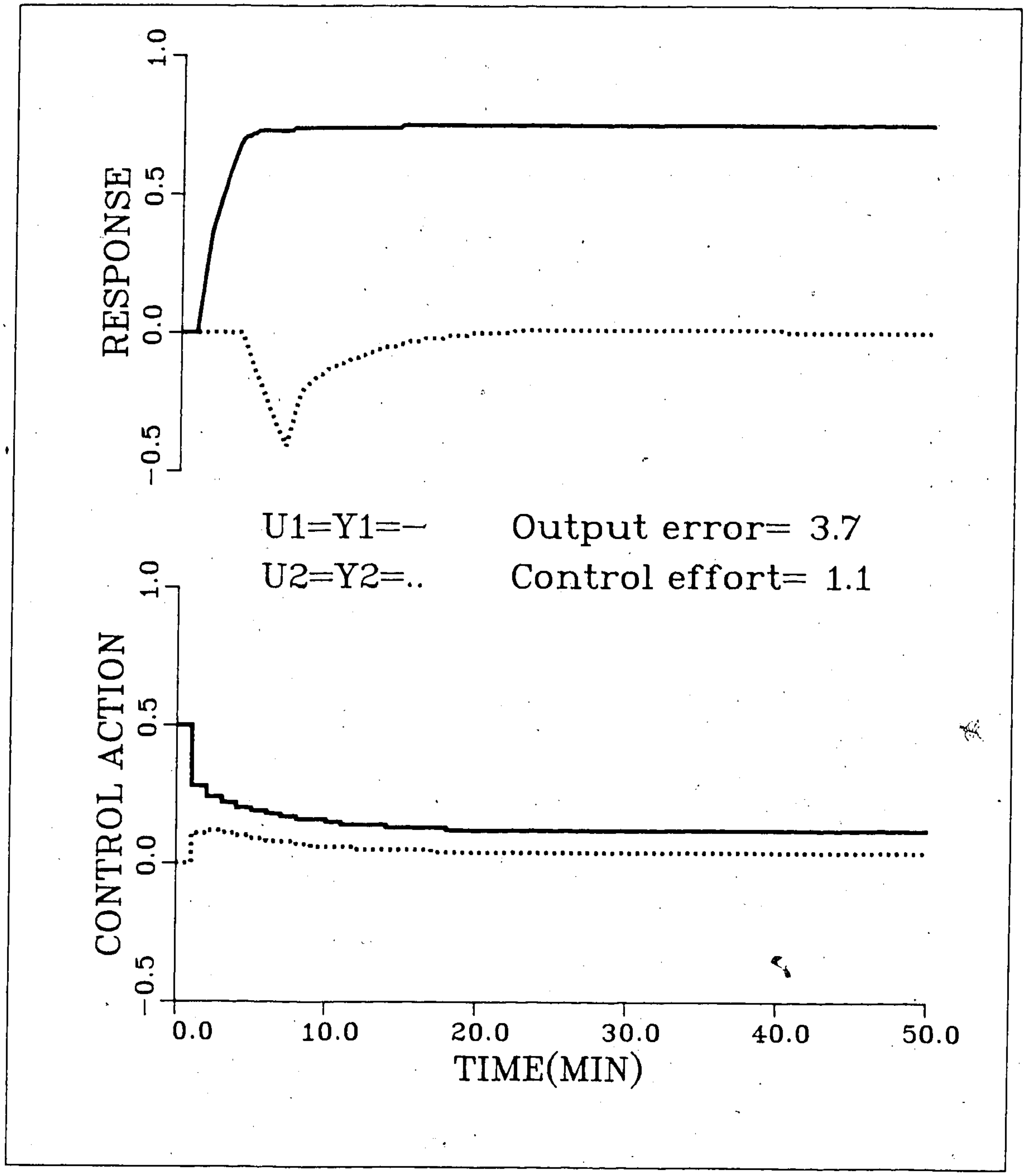

Figure 4.28 Response of both product compositions, with balanced time delays, to a setpoint change of 0.75 in overhead composition, y_1, under OR-PI control

of the latter technique to provide complete decoupling because it is unable to completely cancel the process dynamics with PI controllers.

To further investigate controller performance, the distillation column transfer function is modified to put the minimum time delay, in the first row at element 1,2 rather than 1,1. The process simulator is sampled at one minute to give

$$\underline{G}(z) = \begin{vmatrix} \dfrac{0.744z^{-4}}{1-.942z^{-1}} & \dfrac{-0.879z^{-2}}{1-.954z^{-1}} \\ \dfrac{0.579z^{-8}}{1-.912z^{-1}} & \dfrac{-1.301z^{-4}}{1-.933z^{-1}} \end{vmatrix}$$

Because the transfer function delays are now unbalanced, tuning the IMC controller to find the best response is tedious. The time delay compensator with the offset compensator operative was found to be

$$\underline{\acute{G}}_{+1}(z) = \mathrm{diag}(z^{-3}, z^{-5})$$

and the control law tuning parameters adjusted to

$$N=30 \quad P=20 \quad M=5 \quad \alpha=0.8\ 0.8 \quad \tau_0=2$$

$$\gamma_j=1.0,\ 1.0 \text{ for } j=1,P \quad \beta_j=.01,\ .01 \text{ for } j=1,2$$

$$0.0,\ 0.0 \text{ for } j=3,M$$

These weighting choices prevent singularities in the first τ_0 impulse response matrices and at the same time sufficiently penalize the output error, to achieve good setpoint tracking. The performance of this controller is shown in Figure 4.29. Complete decoupling is not achieved, but the trajectory of the manipulated variables, u_1 and u_2, show less oscillatory behavior due to the relaxation of the

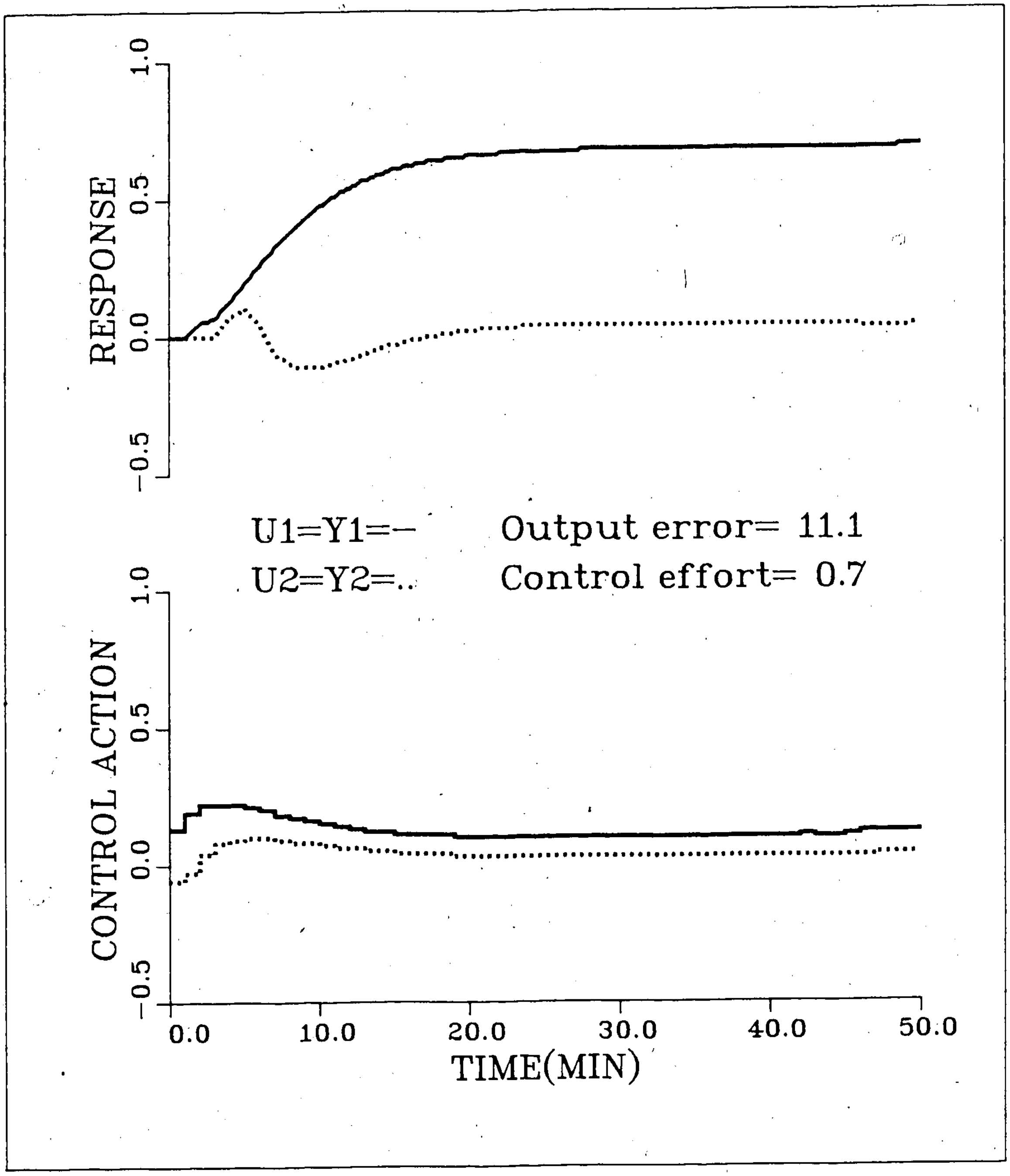

Figure 4.29 Response of both product compositions, with unbalanced time delays, to a setpoint change of 0.75 in overhead composition, y_1, under coupled IMC control

decoupling constraint.

In contrast to operation under IMC which required parameter tuning, the OR compensator with PI controllers, uses the same PI parameters in both simulations. A slight deterioration in the setpoint tracking has occurred, but the response, shown in Figure 4.30, has a trajectory very similar to the balanced process response in Figure 4.28.

On the basis of these simulation results, comparing IMC to the multivariable Smith predictor of Ogunnaike and Ray with PI controllers, two observations are appropriate

i. The IMC controller, via its incorporation of process dynamics into the control law design, plus the feedforward nature of the controller, provides better setpoint tracking when the time delay factorization is optimal. However it is difficult to tune the IMC controller when the time delay factorization is nonoptimal due to singular matrices appearing in the process model impulse response series.

ii. The OR-PI combination is unable to achieve high performance control due to the inherent limitations imposed by PI controllers. The advantage of this combination is a simple tuning procedure that provides a robust closed loop system.

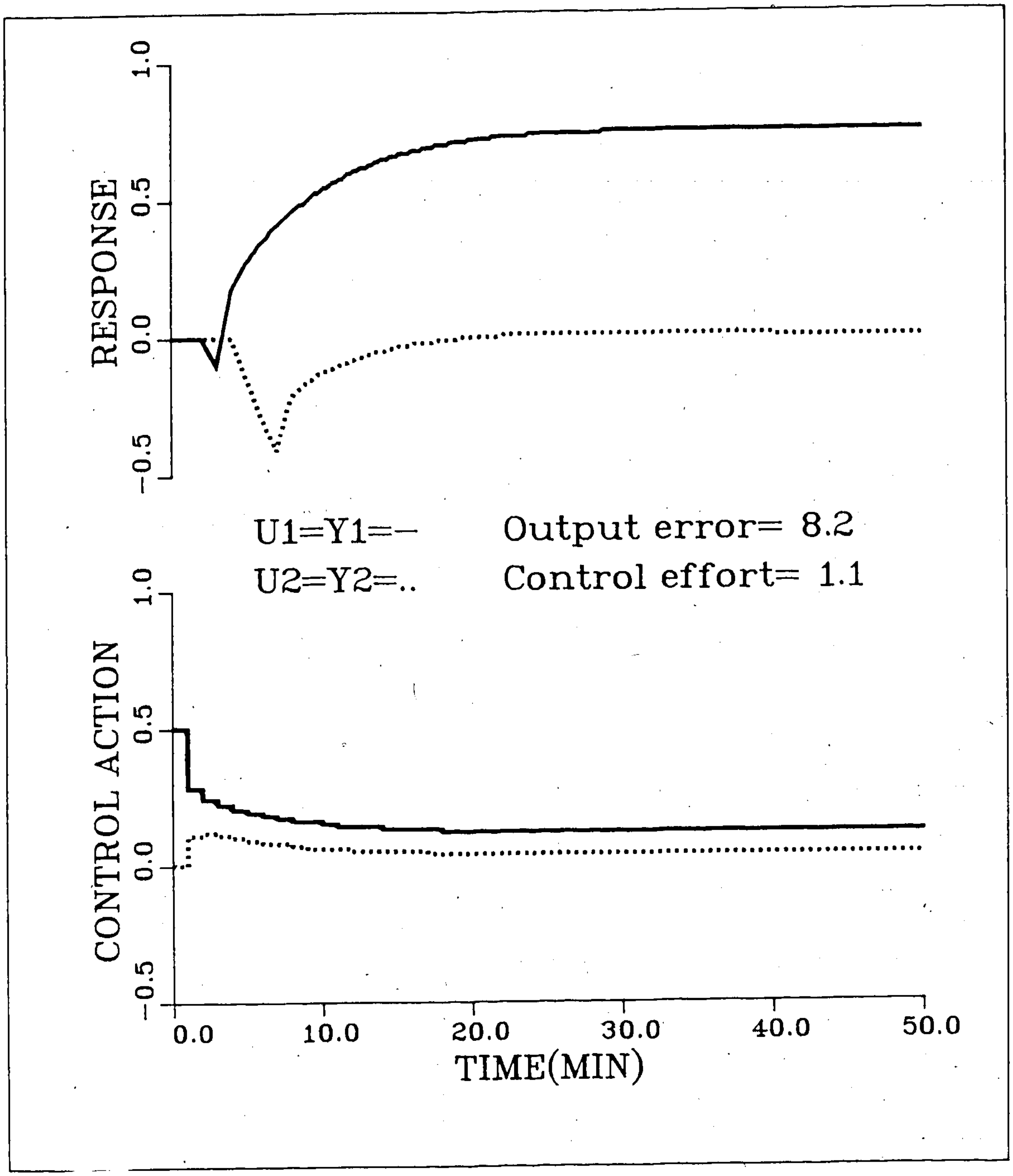

Figure 4.30 Response of both product compositions, with unbalanced time delays, to a setpoint change of 0.75 in overhead composition, y_1, under OR-PI control

4.7 Identification of nonparametric process models

A significant ommision in the development of IMC, is a satisfactory procedure to identify single and multivariable impulse response models. It is always assumed a perfect model of the single or multivariable system exists, when the theory and implementation of IMC is developed by Garcia and Morari. A review of DMC and MAC, which also use nonparametric or nonparsimonious models, shows these methods like IMC treat the identification problem inadequately.

One identification procedure is developed by Richalet et al[12] to identify the impulse response series used by model predictive heuristic control. This identification method uses a recursive least squares algorithm to identify the thirty to fifty parameters of the model series. The feasibility of such a scheme has to be questioned because the computational effort to determine such a large number of parameters is enormous. For example, to identify just one SISO weighting function would require a covariance matrix of at least 2500 elements. Add to this the number of observations necessary t achieve convergence, the possibility of multiple SISO weighting functions and a colossal programming problem is created.

In Section 4.5 the SISO impulse response model for the blending tank was identified using a cross correlation procedure outlined by Box and Jenkins[20]. Box and Jenkins explain that for direct identification by cross correlation of a nonparametric weighting function to be sucsessful, the noise to signal ratio must be small. If this ratio is not small, direct identification becomes unsatisfactory. An identification procedure that is endorsed by Box and Jenkins and more recently by Inouye[23], makes initial estimates of the weighting function through cross correlation, after which an autoregressive model is used to obtain an accurate process model. From an examination of the initial weighting function and the covariance of each term in this function, a good estimate of the system time delay and the approximate linear model order can be determined. With the time delay and model order estimated, an ARMA model can be used to accurately identify the best linear model of the system. By carring out long division on the ARMA model parameters, the final form of the impulse response model is found.

In all of the control theories, using nonparametric descriptions of the process dynamics, it has been suggested the impulse response or step response series improves robustness because it better identifies any nonlinearities in the underlying process. With the procedures suggested by Inouye or by Box and Jenkins, the nonparametric description cannot contain any more information about the process dynamics than the equivalent ARMA model.

5. CONCLUSIONS

The previous three chapters have scrutinized the concept of internal model control to evaluate its applicability to single and multivariable process control. Chapters 2 and 3 examined the development and implementation of IMC for single variable and multivariable processes. In Chapter 4 other algorithms were examined to highlight the relative strengths and weaknesses of IMC. From the material presented in these chapters the following problems associated with applying the IMC technique can be identified:

i. Numerical errors arising from the use of an impulse response series, was found to have a significant impact on the control performance. The IMC technique uses a nonparametric description of the process to formulate the predictive controller. This type of description is used because it allows a prediction of the process output based entirely upon past inputs. However the truncation error due to the finite approximation of an infinite series causes control performance to deteriorate. This characteristic although minor for single variable control, can still be noticed. For example the responses shown in Figures 2.5 and 2.6, exhibit an obvious bounce in the output response due to truncation error. The problem becomes exacerbated in the multivariable control problem because the storage requirements for a nonparametric description are far greater. To meet the storage limitations, truncation error becomes an intrinsic problem. The responses in Figure 3.4 depict how serious the truncation error can become with a multivariable process model.

A second problem caused by the process model is the large errors present in the calculation of the multivariable offset compensator. It was found these errors became unacceptable for a model with a series length of N>30. In Figures 3.9, 3.10 and 3.11 the offset compensator was necessary to achieve even the limited success shown in these figures. As a consequence, the truncation error associated with a series length, N=30, had to be tolerated.

ii. The IMC controller is difficult to tune for those processes where the time delay compensated inverse process model is not realizable. An equivalent categorization would be any process for which the deadbeat (single variable), or completely decoupling (multivariable) IMC controller was unstable. In Chapters 2 and 3, this category of process was controlled using IMC with only limited success. In Chapter 4, IMC demonstrated superior performance whenever the deadbeat or multivariable decoupling controllers were stable. Otherwise the IMC controller did not compare favourably with the other control algorithms. Figure 4.6 shows the disturbance rejection behavior of a NMP single variable system under Smith predictor control. The IMC controlled response for the same system is shown in Figure 4.5. Furthermore, comparison of Figure 4.30 and 4.31 shows that the multivariable IMC controller is unable to provide better setpoint tracking than the multivariable OR compensator with diagonal PI controllers when the simulated multivariable process has unbalanced time delays.

Equivalently, the IMC controller is difficult to tune whenever the default tuning parameter choices

$$\underline{B}=0 \quad \underline{\Gamma}=\underline{I} \quad M=P\leq N$$

result in an unsatisfactory closed loop response. The trial and error procedure suggested by Garcia and Morari[1,3] is inadequate because of the large number of parameters involved.

Although problems do exist in the implementation of IMC, there are many attractive features that deserve note:

i. Whenever the deadbeat or decoupling controller was realizable, the IMC tuning policy of implementing the perfect controller followed by exponential filtering compared favourably to the SP or OR compensator algorithms. This is illustrated for the SISO minimum phase process in Figure 4.3 where the IMC controlled disturbance rejection is better than the rejection using the SP with PI control, shown in Figure 4.4. The multivariable IMC decoupling controller, even with truncation error present in the model representation, provides better setpoint tracking on the Wood-Berry distillation column model in Figure 4.27, than the OR compensator with diagonal PI controllers as shown in Figure 4.28.

ii. The IMC multivariable time delay compensator is developed to provide optimal control given a diagonal precompensator structure. The discussion in section 3.2 argued the diagonal IMC time delay compensator is not optimal when the time delays are unbalanced but this drawback was justified by Garcia and Morari because the diagonal precompensator is easier to formulate. Given the diagonal restriction, Garcia and Morari provide procedures to optimally factor the unbalanced time delays. Other multivariable control policies[13] continue to factor the diagonal element time delays when this is nonoptimal.

iii. It has been demonstrated for small changes in operating conditions, IMC can perform as well as the adaptive STC algorithm on a stochastic, nonlinear control application. Figures 4.18 and 4.19 show the responses of the concentration and inlet flow rate to a positive step in setpoint for the nonlinear blending tank with measurement noise, under IMC and STC respectively. The output error measures are almost identical in these two figures. In addition these figures also illustrate that IMC is not biased by the presence of white measurement noise.

To conclude, Garcia and Morari have suggested an innovative and thought provoking approach to process control. Their use of a nonparametric process model is a recent innovation that is receiving considerable attention. The multivariable IMC development is comprehensive and contains features, like their method of time delay factorization, that are unique. It has been pointed out that problems exist with the implementation of IMC, however the fundamental properties of IMC are still very attractive and it is likely that further work will be carried out to improve the performance of IMC.

6. RECOMMENDATIONS

The results of this study have not only shown where faults exist in the implementation of IMC, but have also indicated directions that future investigations might follow. Suggested studies recommended to improve IMC or aspects of IMC that deserve further investigation are:

i. The trial and error tuning procedure suggested by Garcia and Morari is unsatisfactory because a large number of tuning parameters are involved. One avenue that may offer a solution to this tuning problem, is to incorporate LQC theory into the IMC controller design for those cases where the default tuning parameter choices result in an unstable controller. The success of such a scheme is shown in Figure 4.9 where the MAC algorithm has used LQC theory to choose a minimum phase approximation of the underlying NMP process transfer function. With the minimum phase approximation it is possible to apply the default tuning parameters.

ii. Whenever the SISO or MIMO process model is minimum phase and the input suppresssion parameter, M=P, the length of the optimization horizon, P, will not effect control performance. An optimization horizon of 1, which is equivalent to a single step ahead predictor versus the multistep ahead predictor when P>1, is used for the results in Figures 4.3 and 4.27. Although an ARMA model makes the multistep predictor difficult to formulate relative to an impulse response model, the

ARMA formulation eliminates the problem of truncation error and is simple to formulate for the single step predictor. The recent SISO control work of Defaye, et al.[24] relates closely to the work of Garcia and Morari but the former use an ARMA model for both the single step and multistep prediction problem. Although this procedure is attractive for the minimum phase SISO control problem, its applicability to the more complex problem where NMP characteristics exist needs to be investigated.

iii. Garcia and Morari have developed a diagonal time delay precompensator that is straight forward to implement and is optimal for those cases where the time delays are balanced. This form of time delay compensation is not optimal when the transfer function matrix is unbalanced, however the authors argue designing the optimal time delay factorization for the unbalanced case is difficult. The position of Garcia and Morari should be examined by comparing the performance of the IMC time delay precompensator to techniques that allow a precompensator with off-diagonal elements. Here at the University of Alberta, Rao Sripada, is investigating the possibilities of a less restrictive time delay compensator using the work of Wolovich and Falb[25].

7. REFERENCES

1. Garcia, C.E., Morari, M. ;'Internal Model Control. 1. A Unifying Review and Some New Results', *I &EC Proc. Des. &Dev.* 21,308-323 (1982)
2. Garcia, C.E., Morari, M. ;'Internal Model Control. 2. Design Procedure for Multivariable Systems', Submitted to *I &EC Proc. Des. &Dev.* (August 1982)
3. Garcia, C.E., Morari, M.; ' Internal Model Control. 3. Multivariable Control Law Computation and Tuning Guidelines ', Submitted to *I &EC Proc. Des. &Dev.* (April 1983)
4. Brosilow, C. B.; ' The Structure and design of Smith Predictors from the Viewpoint of Inferential Control'; *Proc. J.A.C.C.*, Denver CO. (1979)
5. Cutler, C.R., Ramaker, B.L. ;' Dynamic Matrix Control A Computer Control Algorithm', *Proc. J.A.C.C.*, 1, WP5-B (1980)
6. Cutler, R., Ramaker, B.; 'Dynamic Matrix ControlA Computer Control Algorithm', *A.I.Ch.E. 86th National Meeting*, Paper No. 51b, (1979)
7. Smith, O.J.M.; ' Closer Control of Control Loops with Dead Time ', *Chem. Eng. Prog.*, 53, 217-219 (1957)
8. Stephanopoulos, G.; *CHEMICAL PROCESS CONTROL An Introduction to Theory and Practice*, Prentice-Hall(Englewood Cliffs, N.J.,1984)
9. Eykhoff, P. (ed);*Trends and Progress in System Identificiation*, Pergamon, (Oxford, 1981)

10. Bierman, G.J.; ' Measurement Updating using the U-D Factorization ', *Automatica*,12, 4, 375-382 (1976)

11. Luyben, W.L.; *Process Modeling, Simulation, and Control for Chemical Engineers*, McGraw Hill, (New York, N.Y., 1973)

12. Richalet, J.A., Rault, A., Testud, J.D., Papon, J.; 'Model predictive heuristic control: Applications to Industrial Processes', *Automatica*, 14, 413-428 (1978)

13. Morris, A.J., Nazer, Y., Wood, R.K.; 'Multivariate Self-Tuning Process Control', *Opt. Con. App.&Meth.*, 3,363-387 (1982)

14. Ogunnaike, B.A., Ray, W.H.; 'Multivariable Controller Design for Linear Systems Having Multiple Time Delays', *AIChE Journal*, 25, 6, 1043-1057 (1979)

15. Ogunnaike, B.A., Lemaire, J.P., Morari, M., Ray, W.H.;' Advanced Multivariable Control of a Pilot-Plant Distillation Column', *AIChE Journal*,29, 4, 632-640(1983)

16. Luyben, W.L.; 'Feedback and Feedforward Control of Distillation Columns with Inverse Response', *Instn. Chem. Eng. Symp. Ser.*, 6, 32, (1969)

17. Doss, J.E., Moore, C.F.; 'The Discrete Analytical Predictor - A Generalized Dead Time Compensation Technique', *Inst. Chem. Pet. Ind.* 16, 135-143 (1980)

18. Rouhani, R.,Mehra, R.; 'Model Algorithmic Control(MAC); Basic Theoretical Properties', *Automatica*, 18, 4, 401-414 (1982)

19. Clarke, D.W., Gawthrop, P.J.; 'Self-tuning control', *Proc. IEE*, 126, 6, 633-640 (1979)

20. Box, G.E.P., Jenkins, G.M.; *Time Series Analysis: forecasting and control*, Rev. Ed., Holden Day, 370-383 (San Francisco, Cal., 1976)

21. Alevisakis, G., Seborg, D.E.; ' An extension of the Smith Predictor method to multivariable linear systems containing time delays', *Int. J. Con.*, 3, 17, 541-551 (1973)

22. Wood, R.K., Berry, M.W.; 'Terminal composition control of a binary distillation column', *Chem. Eng. Sci.*, 28, 1707-1717 (1973)

23. Inouye, Yujiro; 'Approximation of Multivariable Linear Systems with Impulse Response and Autocorrelation Sequences', *Automatica*, 19, 3, 265-277(1983)

24. Defaye, G., Caralp, L., Jouve, P.; 'A Simple Deterministic Predictive Control Algorithm and its Application to an Industrial Chemical Process: a Distillation Column', *Chem. Eng. J.*, 27, 161-166(1983)

25. Wolovich, W.A., Falb, P.L.; 'INVARIANTS AND CANONICAL FORMS UNDER DYNAMIC COMPENSATION', *SIAM J. Con. and Opt.*, 14, 6, 996-1008(1976)

26. Jury, E.I.; *Theory and Application of the z-Transform Method*, Wiley, (Huntington, N.Y., 1973)

27. Kwakernaak, H., Sivan, R.; *Linear Optimal Control Systems*, Wiley Interscience, (New York, N.Y., 1972)

28. Gantmacher, F.R; *Matrix Theory*, Vol 1, Chelsea,(New York, N.Y.,1959)

29. Postlethwaite, I., MacFarlane, A.G.J.; *A Complex Variable Approach to the Analysis of Linear Multivariable Feedback Systems*, Springer Verlag (Berlin, 1979)

30. Franklin, G.F., Powell, J.D.;*Digital Control of Dynamic Systems*,Addison-Wesley(Reading, Mass., 1980)

31. Bennett, C.O., Myers, J.E[illegible]*um, Heat, and Mass Transfer*, 2nd Ed., McGraw-H[illegible] York, N.Y., 1974)

8. APPENDICES

A. Zero offset results for any model that satisfies $\acute{G}(1)=G(1)$

Offset is defined as the difference between the output and the setpoint, that is

$$e(z) = s(z) - y(z) \tag{A.1}$$

If $y(1)$ is calculated from 2.2.4, when $\acute{G}(1)=G(1)$, it is found that

$$y(1) = s(1)-d(1)+d(1) \tag{A.2}$$

so substitution of this result into equation A.1 gives

$$e(1) = 0 \tag{A.3}$$

B. Factorization to remove NMP characteristics from the process model

Suppose a discrete NMP second order transfer function is given by

$$\acute{G}(z)=\frac{z-1.2}{(z-.4)(z-.5)}$$

Choose $\acute{G}_+(z)$ so that the process zero is removed from $\acute{G}_-(z)$ and yet $\acute{G}_+(1)=1$. One possible factorization is

$$\acute{G}_+(z)=\frac{-z+1.2}{1.2z-1}$$

With this factorization $\acute{G}_-(z)$ becomes

$$\acute{G}_-(z)=\frac{-(1.2z-1)}{(z-.4)(z-.5)}$$

so that the NMP characteristic has been removed to provide a minimum phase $\acute{G}_-(z)$.

C. **Solution to the performance index minimization.**

The control objective is minimization to reduce the error between the setpoint and the process output. Defining the error between the setpoint and feedback signal as

$$\epsilon_{k+j} = y_d(k+\tau+j) - (y(k)-y_m(k)) \qquad \text{C.1}$$

By the use of equation C.1 and the impulse response model for y_m it is possible to write the set of linear equations that give the error between the setpoint and output, in terms of future and past control actions as

$$y_d(k+\tau+1)-y(k+\tau+1|k)=\epsilon_{k+1}-[h_1m(k)] -[h_2m(k-1)+h_3m(k-2)\ldots+h_N m(k-N+1)]$$

$$y_d(k+\tau+2)-y(k+\tau+2|k)=\epsilon_{k+2}-[h_1m(k+1) +h_2m(k)] -[h_3m(k-1)+\ldots+h_N m(k-N+2)]$$

$$y_d(k+\tau+3)-y(k+\tau+3|k)=\epsilon_{k+3}-[h_1m(k+2)+ h_2m(k+1)+h_3m(k)] -[h_4m(k-1)+\ldots+h_N m(k-N+3)]$$

$$\vdots$$

$$y_d(k+\tau+P)-y(k+\tau+P)=\epsilon_{k+P} - [h_1m(k+P-1)+h_2m(k+P-2)+\ldots+h_P m(k)] -[h_{P+1} m(k-1)+\ldots+h_N m(k-N+P)]$$

This set of P linear equations can be written in matrix form by defining the following matrices:

$$E = \begin{vmatrix} y_d(k+\tau+1)-(y(k)-y_m(k)) \\ y_d(k+\tau+2)-(y(k)-y_m(k)) \\ y_d(k+\tau+3)-(y(k)-y_m(k)) \\ \cdot \\ \cdot \\ \cdot \\ y_d(k+\tau+P)-(y(k)-y_m(k)) \end{vmatrix}$$

$$\Lambda = \begin{vmatrix} h_1 & & & & \\ h_2 & h_1 & & & \\ h_3 & h_2 & h_3 & & \\ h_4 & h_3 & h_2 & h_1 & \\ h_m & h_4 & h_3 & h_2 & h_1 \\ h_6 & h_5 & h_4 & h_3 & h_2+h_1 \\ & \cdot & \cdot & \cdot & \cdot \\ & & \cdot & \cdot & \cdot \\ & & & & \Sigma h_i \end{vmatrix}$$

$$\Gamma = \mathrm{dia}[\gamma_1, \gamma_2, \ldots, \gamma_{P-1}, \gamma_P]$$

$$B = \mathrm{dia}[\beta_1, \beta_2, \ldots, \beta_{M-1}, \beta_M]$$

$$U = \begin{vmatrix} m(k) \\ m(k+1) \\ m(k+3) \\ \cdot \\ \cdot \\ \cdot \\ m(k+M-1) \end{vmatrix}$$

$$V = \begin{vmatrix} m(k-1) \\ m(k-2) \\ m(k-3) \\ \cdot \\ \cdot \\ \cdot \\ m(k-M) \\ \cdot \\ \cdot \\ \cdot \\ m(k-N+1) \end{vmatrix}$$

$$\Psi = \begin{vmatrix} h_2 & h_3 & \cdot & \cdot & \cdot & h_N \\ h_3 & h_4 & \cdot & \cdot & h_N & \\ h_4 & \cdot & h_N & & & \\ \cdot & \cdot & & & & \\ h_N & & & & & \end{vmatrix}$$

With these matrices equation 2.4.2 can be written as:

$$J=\min(\Gamma E-\Gamma\Lambda U-\Gamma\Psi V)'(\Gamma E-\Gamma\Lambda U-\Gamma\Psi V)+(BU)'(BU) \qquad C.4$$

$$dJ/dU=0=2(-\Gamma\Lambda)'(\Gamma E-\Gamma\Lambda U-\Gamma\Psi V)+2(BU)'(BU) \qquad C.5$$

The above equation is solved for U but because the control law is solved at every control interval, only the control input, m(k), is used. Therefore:

$$m(k)=[1\ 0\ 0 \ldots 0_M][\Lambda'\Gamma'\Gamma\Lambda+B'B]^{-1}\Lambda'\Gamma'\Gamma[E-\Psi V] \qquad C.6$$

The perfect controller will result from C.6 when the tuning parameters are

$P=M\leq N$, $\beta_j=0$ for $j=1,M$; $\gamma_j=1$ for $j=1,P$

so it follows that

$\Gamma=I \qquad B=0$

$$\Lambda = \begin{vmatrix} h_1 & & & & & & \\ h_2 & h_1 & & & & & \\ h_3 & h_2 & h_1 & & & & \\ \cdot & \cdot & \cdot & \cdot & & & \\ \cdot & \cdot & \cdot & \cdot & \cdot & & \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \\ h_M & \cdot & \cdot & \cdot & \cdot & \cdot & h_1 \end{vmatrix}$$

Equation C.6 is simplified in the following manner

$$m(k)=[1\ 0\ 0 \ldots 0_M][\Lambda'\Lambda]^{-1}\Lambda'[E-\Psi V] \qquad C.7$$

$$m(k)=[1\ 0\ 0 \ldots 0_M]\Lambda^{-1}[E-\Psi V] \qquad C.8$$

$$m(k)=[\ 1/h_1\ \ 0\ \ 0\ \dots\ \underset{M}{0}\ ][E-\Psi V] \qquad C.9$$

$$h_1 m(k)=[y_d(k+\tau+1)-(y(k)-y_m(k))- (h_2 m(k-1)+h_3 m(k-2)+\dots+h_N m(k-N+1))] \qquad C.10$$

$$h_1 m(k)+h_2 m(k-1)+\dots+h_N m(k-N+1)=[y_d(k+\tau+1)-\hat{d}(k)] \qquad C.11$$

If the feedback signal $\hat{d}$ is zero, which corresponds to a correct model and no disturbance, d(k), taking the z-transform of equation C.11 clearly shows the delay free process model to be

$$m(z)=\frac{1}{H(z)} z^{\tau} y_d(z) \qquad C.12$$

D. **Stability properties of the SISO controller**

The SISO stability properties, are based on the following important restrictions,

i. The process input output relationship, given by equation 2.2.1 is stable for bounded inputs and bounded disturbances. This implies G(z) must be stable.

ii. The process model used in the control law calculation is correct. That is $\hat{G}(z)=G(z)$.

iii. Because of these restrictions and the nature of the closed loop characteristic equation (cf equation 2.2.5), to prove stability of the closed loop system, it is sufficient to demonstrate that the roots of the controller Gc(z)=Nc(z)/Dc(z) are inside the unit circle. When the backward shift operator z^{-1} is used, all roots of the polynomial $Dc(z^{-1})$ must lie outside the unit circle for closed loop stability. From equation C.6, it can be shown that $Dc(z^{-1})$ is given by

$$Dc(z^{-1})=1+[1\ 0\ldots0][\Lambda^t\Gamma^t\Gamma\Lambda+B^tB]^{-1}\Lambda^t\Gamma^t\Gamma\Psi Z \qquad D.1$$

where $Z^t=[z^{-1}\ z^{-2}\ z^{-3}\ \ldots\ z^{-N+1}]$

D.1 Any minimum phase, or nonminimum phase system can be stabilized by making the ratio of M/P sufficiently small

For the purposes of illustration consider the case where $\beta_j=0$ for $j=1,M$ and $\gamma_j=1$ for $j=1,P$. In addition let $M=1$ and $P>>M$. With these simplifications equation D.1 becomes

$$Dc(z^{-1})=1+[\Lambda^t\Lambda]^{-1}\Lambda^t\Psi Z \qquad D.2$$

Jury[26] has demonstrated that all of the roots of the polynomial $Dc(z^{-1})$ will lie outside the unit circle provided the following expression is satisfied

$$1>\sum_{i=1}^{N-1}|\delta_i| \qquad D.3$$

where $[\delta_1\ \delta_2\ .\ .\ .\ \delta_{N-1}]=[\Lambda^t\Lambda]\Lambda^t\Psi$

$\delta_0=1$

Now for $M=1$ Λ^t becomes

$$\Lambda^t=[a_1\ a_2\ \ldots\ a_N\ a_N\ \ldots\ a_N]$$

$$\longleftarrow\ P\ \longrightarrow$$

where the vector elements, a_i are calculated from

$$a_i=\sum_{k=1} h_k$$

so $$\Lambda^t\Lambda=\sum_{i=1}^{N-1} a_i^2+\sum_{i=N}^{P} a_n^2$$

thus, the roots of $Dc(z^{-1})$ are given by

$$Dc(z^{-1})=0=1+\Big[\sum_{i=1}^{N-1} a_i^2 + (P-N)a_N^2\Big]^{-1}\Lambda^t\Psi Z \qquad \text{D.4}$$

$$\Lambda^t\Psi=\Big[\sum_{i=1}^{N-1} a_i h_{i+1} \;\; \sum_{i=1}^{N-2} a_i h_{i+2} \;\; \ldots \;\; \sum_{i=1}^{2} a_i h_{N+i-2} \;\; a_1 h_{N-1}\Big]$$

With these definitions the criterion for stability is

$$\Big|\sum_{i=1}^{N-1} a_i^2+(P-N)a_N^2\Big|>\sum_{k=1}^{N-1}\Big|\sum_{i=1}^{k} a_i h_{N+i-k}\Big| \qquad \text{D.5}$$

So it follows that by selecting P to be sufficiently large, equation D.5, will be satisfied.

D.2 Any minimum phase, or nonminimum phase system can be stabilized by choosing a finite $\beta^+>0$

For a $\beta^+>0$ such that $\beta_j\geq\beta^+$ $(j=1,2,\ldots,M)$ the control law given by equation B.6 can be shown to be stable for all $M\geq1$, $P\geq1$ and $\gamma_j>0$. To simplify the derivation β will be considered to be constant over the optimzation horizon P. That is

$\beta=\beta_j$ for all j

The characteristic roots of the controller transfer function are the roots of the polynomial $Dc(z)$, that is

$$Dc(z)=\delta_0+\delta_1 z^{-1}+\ldots+\delta_j z^{-j} \qquad \text{D.6}$$

the polynomial $Dc(z)$ is of order N-1

$${}_0=1$$

$$\delta_i=b[\Delta^t\Delta+\beta^2 I]^{-1}\Delta^t\Psi \qquad \text{D.7}$$

where

$$b=[1\ 0\ 0 \ldots 0_M\ ]$$

$$\Delta=\Gamma\Lambda$$

As noted previously, for the controller to be stable all of the roots of $Dc(z^{-1})$ must be outside the unit circle. This condition exists if equation D.3 is satisified. From equation D.7 it is possible to write

$$|\delta_i|=\leq 1/\beta^2\ ||b||\ ||\Delta^t\Psi||\ ||[\Delta^t\Delta/\beta^2+I]^{-1}||$$

where the norm is defined as

$$||a||\equiv\sqrt{a^t a}$$

By approximation of the norm of $||[\Delta^t\Delta/\beta^2+I]^{-1}||$ with the binomial expansion

$$||[\Delta^t\Delta/\beta^2+I]^{-1}||=||I-\Delta^t\Delta/\beta^2+(\Delta^t\Delta/\beta^2)^2-(\Delta^t\Delta/\beta^2)^3+\ldots||$$

it follows that if β^2 is chosen such that $\beta^2>||\Delta^t\Delta||$, the binomial series will converge. Therefore it should be possible to find a β, such that the series converges and satisfies the requirement

$$1>\sum_{i=1}^{N-1}|\delta_i|$$

E. The continuous and discrete transfer functions used in Section 2.6 to illustrate the IMC tuning procedure.

The parameters of the transfer functions used in Section 2.6 were chosen because the steady state gains and time constants illustrate the effect of the IMC tuning parameters.

i. The minimum phase process.

The continuous transfer function is given by

$$y(s)=.1e^{-4s}u(s)/[(s+.1)(s+1)] \quad \text{E.1.1}$$

Transform of equation E.1.1 from the continuous domain to the discrete domain by the inclusion of a ZOH and a sampling rate of four minutes yields

$$y(z)=z^{-2}[0.109+0.0729z^{-1}]u(z)/[1-1.12z^{-1}+0.301z^{-2}] \quad \text{E.1.2}$$

ii. The nonminimum phase process.

The process will exhibit nonminimum phase behavior because a process zero is located at .1 in the continuous model

$$y(s)=-1e^{-8s}(s-.1)u(s)/[(s+.1)(s+1)] \quad \text{E.2.1}$$

When this transfer function is combined with a ZOH and sampled at an interval of eight minutes, a discrete representation that does not exhibit NMP behavior results

$$y(z)=z^{-2}[0.00190+0.549z^{-1}]u(z)/[1-0.450z^{-1}+0.00015z^{-2}] \quad \text{E.2.2}$$

When the sampling rate is reduced from eight to four minutes, the discrete representation includes the NMP behavior of the underlying continuous process

$$y(z)=z^{-3}[-.467+.791z^{-1}]u(z)/[1-.689z^{-1}+0.0123z^{-2}] \quad \text{E.2.3}$$

iii. The unstable process

The process is unstable due to a pole at .1 in the continuous model

$$y(s)=.2e^{-4s}u(s)/[(s-.1)(s+2)] \qquad \text{E.3.1}$$

This unstable process is stabilized using a proportional feedback controller with a gain of 2.0. The discrete representation of the stabilized process, sampled at one minute is

$$y(z)=z^{-5}[0.059+0.0320z^{-1}]u(z)/[1-1.240z^{-1}+.150z^{-2}] \qquad \text{E.3.2}$$

F. Why most nth order linear processes require n sampling intervals to arrive at steady state

Consider a second order linear process (n=2) with constant coefficients.

$$y(k)=f(y,u)$$

$$dy(k)/dt=f(dy/dt,u)$$

The variable $dy(k)/dt$ represents the first derivative of y with respect to time. The initial conditions are:

$$y(0)=0$$

$$dy(0)/dt=0$$

for a change in setpoint from 0 to 1, the system will arrive at the new steady state when:

$$y(k+j)=1$$

$$dy(k+j)/dt=0$$

The system has two initial conditions and two boundary conditions. Therefore it will require four equations to determine the sequence of control inputs. The four equations are:

$$y(k+1)=f(y,u)$$

$dy(k+1)/dt=f(dy/dt,u)$

$y(k+2)=f(y,u)$

$dy(k+2)/dt=f(dy/dt,u)$

Two sampling periods are required to bring all process states to the new steady state conditions. If the control variable is added to the initial and boundary conditions, it will require six equations or n+1 sampling intervals because there are now three initial conditions and three boundary conditions to satisfy.

G. Factorization to remove NMP characteristics from the process model

To demonstrate the ability of the diagonal factorization to remove NMP characteristics from the process model, consider a two input, two output linear process described by equation 3.1.1, where the transfer function $\underline{G}(z)$ is given by the following relation

$$\underline{G}(z) = \begin{vmatrix} \frac{0.6}{z-0.4} & \frac{0.5}{z-0.5} \\ \frac{0.6}{z-0.5} & \frac{0.6}{z-0.4} \end{vmatrix} \qquad \text{G.1}$$

The process zeros of this transfer function are found from the denominator roots of the matrix determinant. That is

$$|\underline{G}(z)| = \frac{0.06(z^2-2z+0.7)}{(z-0.4)^2(z-0.5)^2} = 0$$

From the denominator roots, the system zeros are located at

$z = 1 \pm \sqrt{0.3}$

Because a zero occurs at $z-(1+\sqrt{0.3})$ the inverse process model will be unstable. A precompensator, $\underline{G}_{+1}(z)$, is chosen to cancel the unstable zero and because, $\underline{G}_{+1}(z)$ is diagonal, each diagonal element must contain the zero outside the unit circle and to ensure zero offset, the factorization must satisfy $\underline{G}_{+1}(1)=\underline{I}$. Kwakernaak and Sivan[27] have demonstrated one form of factorization that satisfies the decoupling constraint and minimizes the sum of squared errors in the control law is

$$\underline{G}_{+1}(z)=\begin{vmatrix} \dfrac{-z+1.547}{1.547z-1} & 0 \\ 0 & \dfrac{-z+1.547}{1.547z-1} \end{vmatrix} z^{-1}$$

Because this factorization leads to severe closed loop inverse response behavior, or NMP behavior, in both input and output some other less restrictive factorization should be used to factor NMP characteristics. However a precompensator with off diagonal terms is mathematically cumbersome to manipulate, so a sampling rate that removes the NMP behavior from the process model is recommended.

H. Solution to the multivariable predictive control law minimization.

The matrix of control inputs, $\underline{U}(z)$, is chosen to minimize the costing function given by equation 3.3.12. The multivariable solution is almost identical to the single variable solution, shown in Appendix C. The multivariable costing function is

$$J=\min_{\underline{m}(k)} \sum_{j=1}^{P+\tau_0} [\,||\underline{\Delta}(j)||^2 + ||\underline{\nabla}(j)||^2\,] \qquad 3.3.12$$

$$\vdots$$

Subject to

$$\underline{\Delta}(j)=[\underline{s}^+(k+j-\tau_0)-\underline{y}^+(k+j-\tau_0|k)]\underline{\gamma}_j^t$$

$$\underline{\nabla}(j)=\underline{m}(k+j)\underline{\beta}_j^t$$

$$\underline{y}^+(k+j-\tau_0|k)=\underline{y}_m^+(k+j-\tau_0)+\underline{d}^+(k+j-\tau_0|k)$$

$$=\sum_{i=1}^{N+\tau_0} \underline{H}_i\underline{m}(k-i) + \underline{d}^+(k+j-\tau_0|j)$$

where

$$||x||^2=x^tx$$

By defining the following additional matrices, the minimization can be rewritten entirely in matrix notation. Define $\underline{E}(k-\tau_0)$ to be the matrix of differences between the vectors of setpoints and disturbance estimates over the horizon of $P+\tau_0$. That is

$$\underline{E}(k-\tau_0)=\begin{vmatrix} \underline{s}^+(k+1-\tau_0)-\underline{d}^+(k+1-\tau_0|k) \\ \underline{s}^+(k+2-\tau_0)-\underline{d}^+(k+2-\tau_0|k) \\ \vdots \\ \underline{s}^+(k+P-\tau_0)-\underline{d}^+(k+P-\tau_0|k) \\ \underline{s}^+(k+P+1-\tau_0)-\underline{d}^+(k+P+1-\tau_0|k) \\ \vdots \\ \underline{s}^+(k+P)-\underline{d}^+(k+P|k) \end{vmatrix}$$

The series of linear equations describing the output from the process model, $y_m(k+j-\tau_o)$ for $j=1,P+\tau_o$, can be written in terms of future and past control actions

$$\underline{Y}_m(k-\tau_o) = \underline{\Lambda}\underline{U} + \underline{\Psi}\underline{V} \qquad \text{H.1}$$

Where $\underline{Y}_m$, $\underline{\Lambda}$, $\underline{U}$, $\underline{\Psi}$ and $\underline{V}$ are defined as follows

$$\underline{Y}_m(k-\tau_o) = \left|\begin{array}{c} y_m(k+1-\tau_o) \\ y_m(k+2-\tau_o) \\ \cdot \\ \cdot \\ \cdot \\ y_m(k+P-\tau_o) \\ \cdot \\ \cdot \\ y_m(k+P) \end{array}\right|$$

$$\underline{\Lambda} = \left|\begin{array}{cccc} \underline{H}_1 & & & \\ \underline{H}_2 & \underline{H}_1 & & \\ \underline{H}_3 & \underline{H}_2 & \underline{H}_1 & \\ \underline{H}_4 & \underline{H}_3 & \underline{H}_2 & \underline{H}_1 \\ \underline{H}_5 & \underline{H}_4 & \underline{H}_3 & \underline{H}_2+\underline{H}_1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \underline{H}_N & \cdot & \cdot & \underline{H}_4+\ldots+\underline{H}_1 \\ \cdot & \cdot & \cdot & \cdot \\ \underline{H}_{N+\tau_o} & \cdot & \cdot & \underline{H}_N+\ldots+\underline{H}_1 \\ & \cdot & \cdot & \cdot \\ & & \cdot & \cdot \\ & & & \underline{H}_{N+\tau_o}+\ldots+\underline{H}_1 \end{array}\right| \; P+\tau_o$$

$$\longleftarrow \; M \; \longrightarrow$$

The matrix, Λ, has $(P+\tau_o)$x(number of outputs) rows and Mx(number of inputs) columns.

$$\underline{U}= \begin{vmatrix} \underline{m}(k) \\ \underline{m}(k+1) \\ \underline{m}(k+2) \\ \vdots \\ \underline{m}(k+M-1) \end{vmatrix}$$

$$\underline{\Psi}= \begin{vmatrix} \underline{H}_2 & \underline{H}_3 & \cdots & \underline{H}_{N+\tau_0} \\ \underline{H}_3 & \underline{H}_4 & \cdots & \underline{H}_{N+\tau_0} \\ \cdots & \underline{H}_{N+\tau_0} & & \\ \cdots & \underline{H}_{N+\tau_0} & & \\ \cdot\ \cdot & \underline{H}_{N+\tau_0} & & \\ \underline{H}_{N+\tau_0} & & & \end{vmatrix}$$

$$\underline{V}= \begin{vmatrix} \underline{m}(k-1) \\ \underline{m}(k-2) \\ \underline{m}(k-3) \\ \vdots \\ \underline{m}(k-N-\tau_0) \end{vmatrix}$$

Using the definitions for $\underline{Y}_m(k-\tau_0)$ and $\underline{E}(k-\tau_0)$, equation 3.3.12 can be rewritten as

$$J = \min_{\underline{U}(k)} ||\underline{\Gamma}(\underline{E}(k-\tau_0)-\underline{Y}_m(k-\tau_0))|| + ||\underline{B}\underline{U}(k)|| \qquad \text{H.2}$$

where

$$\underline{\Gamma}^{t}=\text{dia}(\gamma_1\ \gamma_2\ \cdots\ \gamma_P)$$

$$\underline{B}=\text{dia}(\beta_1\ \beta_2\ \beta_3\ \cdots\ \beta_M)$$

In the following steps, equation H.2 is expanded, differentiated with respect to $\underline{U}(k)$, and then solved to find a solution for $\underline{U}(k)$.

$$0=\frac{d}{d\underline{U}}[(\underline{\Gamma}\underline{E}-\underline{\Gamma}\underline{\Lambda}\underline{U}-\underline{\Gamma}\underline{\Psi}\underline{V})'(\underline{\Gamma}\underline{E}-\underline{\Gamma}\underline{\Lambda}\underline{U}-\underline{\Gamma}\underline{\Psi}\underline{V}) + (\underline{B}\underline{U})'(\underline{B}\underline{U})]$$

$$0 = [\ 2(\underline{\Gamma}\underline{E}-\underline{\Gamma}\underline{\Lambda}\underline{U}-\underline{\Gamma}\underline{\Psi}\underline{V})'(-\underline{\Gamma}\underline{\Lambda}) + 2(\underline{B}\underline{U})'\underline{B}]$$

$$0 = -(\underline{\Gamma}\underline{E})'(\underline{\Gamma}\underline{\Lambda}) + (\underline{\Gamma}\underline{\Lambda}\underline{U})'(\underline{\Gamma}\underline{\Lambda}) + (\underline{\Gamma}\underline{\Psi}\underline{V})'(\underline{\Gamma}\underline{\Lambda}) + (\underline{B}\underline{U})'\underline{B}$$

$$[\underline{\Lambda}'\underline{\Gamma}'\underline{\Gamma}\underline{\Lambda}+\underline{B}'\underline{B}]\underline{U} = \underline{\Lambda}'\underline{\Gamma}'\underline{\Gamma}\underline{E}-\underline{\Lambda}'\underline{\Gamma}'\underline{\Gamma}\underline{\Psi}\underline{V}$$

$$\underline{K}\underline{U} = \underline{\Lambda}'\underline{\Gamma}'\underline{\Gamma}(\underline{E}-\underline{\Psi}\underline{V}) \qquad \text{H.3}$$

where $\underline{K}=[\underline{\Lambda}'\underline{\Gamma}'\underline{\Gamma}\underline{\Lambda}+\underline{B}'\underline{B}]$

As in the SISO case, only the vector of inputs at time k, $\underline{m}(k)$, is implemented before the problem is reformulated at the next sampling interval, k+1. Therefore inverting $\underline{K}$ and solving for $\underline{m}(k)$ gives the multivariable control law

$$\underline{m}(k) = \underline{b}'\underline{K}^{-1}[\underline{\Lambda}'\underline{\Gamma}'\underline{\Gamma}(\underline{E}-\underline{\Psi}\underline{V})] \qquad \text{H.4}$$

with $\underline{b}$ defined by the following matrix

$$\underline{b}=\begin{vmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ . & . & . & \dots & 0 \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \\ . & . & . & \dots & 0 \\ 0 & . & . & \dots & 0 \end{vmatrix} \updownarrow \text{M(number of outputs)}$$

← number of inputs →

For the case where $\tau_0 \neq 0$, the matrix $\underline{K}$ will be rank deficient for some values of $\underline{\Gamma}$, $\underline{B}$ and M so inversion is impossible due to singularities. The IMC technique compensates for the rank deficiency by either changing the tuning parameters or augmenting the elements of the control law until a nonsingular matrix results[3].

For the case where $\underline{\Gamma}=\underline{I}$, $\underline{B}=0$, $M=P\leq N$ and $\tau_0=0$, equation H.4 simplifies to the decoupling controller. This can be demonstrated as follows. Since $\underline{\Lambda}$ becomes a lower triangular matrix equation H.4 reduces to

$$\underline{m}(k) = [\underline{H}_1^{-1}\ 0\ 0\ \ldots\ 0](\underline{E}-\underline{\Psi}\underline{V})$$

$$\underline{m}(k) = \underline{H}_1^{-1}[\underline{s}^+(k)-\underline{d}^+(k) - \sum_{i=2}^{N+\tau_0} \underline{H}_i\underline{m}(k-i)] \qquad \text{H.5}$$

By multiplying both sides of H.5 by $\underline{H}_1$ and then taking the z transform of the resulting difference equation, the result is the decoupling controller

$$\underline{H}(z)\underline{m}(z) = \underline{s}^+(z)-\underline{d}^+(z)$$

$$\underline{m}(z) = \underline{H}(z)^{-1}(\underline{s}^+(z)-\underline{d}^+(z)) \qquad \text{H.6}$$

I. Closed loop properties of the multivariable predictive controller

The multivariable stability properties are based on the following important restrictions:

i. The process input-output relationship, given by equation 3.1.1, is stable for bounded inputs and bounded disturbances. This implies $\underline{G}(z)$ is stable.

ii. The process model used in the control law calculation is correct, that is $\hat{\underline{G}}(z)=\underline{G}(z)$.

iii. Because of these restrictions and the nature of the closed loop system given by equation 3.3.2, to prove stability of the closed loop system it is sufficient to prove that the roots of the determinant $\det\{\underline{D}(z)\}$ are inside the unit circle[3] where $\underline{D}(z)$ is calculated from the closed loop transfer function by the following method.

If equation H.4 is expressed with all control terms on one side of the equality sign and all error terms on the other side, the equation can be rewritten as

$$\underline{m}(k)+\sum_{j=2}^{N+\tau_0} \underline{D}_j \underline{m}(k-j) = \sum_{j=1}^{P+\tau_0} \underline{E}_j (\underline{s}^*(k+j-\tau_0)-\underline{d}^*(k+j-\tau_0))$$

By comparing this form, to the matrix notation form of equation H.4, it can be shown that

$$[\underbrace{\underline{I}\ \underline{D}_2\ \underline{D}_3\ \ldots\ \underline{D}_j}_{N+\tau_0}]=[\underline{I}\ \underline{b}'\underline{K}^{-1}\underline{\Delta}'\underline{\Gamma}'\underline{\Gamma}\underline{\Psi}] \qquad \text{I.1}$$

$$[\underbrace{\underline{E}_1\ \underline{E}_2\ \ldots\ \underline{E}_j}_{P+\tau_0}] =\underline{b}'\underline{K}^{-1}\underline{\Delta}'\underline{\Gamma}'\underline{\Gamma} \qquad \text{2}$$

Using equations I.1 and I.2 ~~to~~ write the controller transfer function gives

$$G\underline{c}(z) =[\underline{D}(z)]^{-1}\underline{E}(z) \qquad \text{I.3}$$

I.1 Effect of the ratio of input to output penalty on closed loop stability

This proof will demonstrate there exists a finite $\beta^* 0$ such that if $||\underline{B}'\underline{B}||>\beta^*$ the controller given by equation I.3 is stable.

From equation I.3 stability is determined from roots of the determinant of the return difference operator

$$\det\{\underline{I}+\underline{b}'\underline{K}^{-1}[\underbrace{\underline{A}_1 z^{-1}+\underline{A}_2 z^{-2}+\underline{A}_3 z^{-3}+\ldots+\underline{A}_j z^{-j}}_{N+\tau_0-1}]\}=1+\sum_{j=1}^{N+\tau_0-1} f_j z^{-j}=0. \qquad \text{I.4}$$

where $\underline{\Delta}'\underline{\Gamma}'\underline{\Gamma}\underline{\Psi}=[\underbrace{\underline{A}_1\ \underline{A}_2\ \ldots\ \underline{A}_j}_{N+\tau_0-1}]$

Because the backward shift operator is used all roots of equation I.4 must be outside the unit circle for stability. Using Schur's formula[28] enables equation I.4 to be written in the more convenient form

$$\det\begin{vmatrix} \underline{I} & \underline{A}_1 z^{-1} + \ldots + \underline{A}_j z^{-j} \\ -\underline{b}^t \underline{K}^{-1} & \underline{I} \end{vmatrix} \qquad \text{I.5}$$

If $\underline{K}^{-1}$ is expressed as

$$\underline{K}^{-1} = [(\underline{B}^t\underline{B})^{-1}\underline{\Gamma}^t\underline{\Delta}^t\underline{\Delta\Gamma} + \underline{I}]^{-1}(\underline{B}^t\underline{B})^{-1} \qquad \text{I.6}$$

and if $\beta*$ is selected sufficiently large

$$||\underline{B}^t\underline{B}|| > ||\underline{\Gamma}^t\underline{\Delta}^t\underline{\Delta\Gamma}||$$

then, equation I.6 can be written in terms of a convergent binomial series

$$||\underline{K}^{-1}|| \le$$

$$||(\underline{B}^t\underline{B})^{-1}|| + ||\underline{\Gamma}^t\underline{\Delta}^t\underline{\Delta\Gamma}||^1\,||(\underline{B}^t\underline{B})^{-1}||^2$$

$$+ ||\underline{\Gamma}^t\underline{\Delta}^t\underline{\Delta\Gamma}||^2\,||(\underline{B}^t\underline{B})^{-1}||^3 + \ldots \qquad \text{I.7}$$

Since $\underline{B}=\text{dia}[\beta*\ \beta*\ \ldots]$ and $||(\underline{B}^t\underline{B})^{-1}|| = 1/\beta*$, equation I.7 can be reduced to

$$||\underline{K}^{-1}|| \le$$

$$1/\beta* + ||\underline{\Gamma}^t\underline{\Delta}^t\underline{\Delta\Gamma}^t||\,(1/\beta*)^2$$

$$+ ||\underline{\Gamma}^t\underline{\Delta}^t\underline{\Delta\Gamma}||^2 (1/\beta*)^3 + \ldots \qquad \text{I.8}$$

It follows from the series expansion, that as $\beta*\to\infty$ $||\underline{K}^{-1}||\to 0$ and because $\underline{K}$ is nonsingular, the matrix must vanish as $\beta*\to\infty$. With this knowledge the determinant given by equation I.4 is evaluated from the expression

$$1+\sum_{j=1}^{N+\tau_0-1} f_j z^{-j}=0 \qquad \text{I.9}$$

For a $\beta*$ sufficiently large, the sum $\Sigma f_i<1$. This satisfies the stability criterion as demonstrated by Jury[26].

I.2 The stabilizing effect of the ratio M/P

If it can be assumed that $\underline{\Gamma}=\underline{I}$, $\underline{B}=0$, then for a sufficiently small M, and a sufficiently large P>N, the control law given by equation H.4 is stable. The proof, for simplicity is carried out for M=1 and [illegible] large. For this choice of tuning parameters, equation H.4 is reduced to

$$\underline{m}(k)=(\underline{\Delta}'\underline{\Delta})^{-1}\underline{\Delta}'[\underline{E}-\underline{\Psi}\underline{V}] \qquad \text{I.10}$$

where $\underline{\Delta}'=[\underline{H}_1 \;\; \underline{H}_1+\underline{H}_2 \;\ldots\; \sum_{i=1}^{N+\tau_0}\underline{H}_i \ldots \sum_{i=1}^{N+\tau_0}\underline{H}_i]'$

and the product $\underline{\Delta}'\underline{\Psi}$ is

$$\underline{\Delta}'\underline{\Psi}=[\underbrace{\underline{A}_1 \;\; \underline{A}_2 \;\ldots\; \underline{A}_j}_{N+\tau_0-1}]$$

$$\underline{A}_i=\sum_{j=1}^{N+\tau_0-i}\Big(\sum_{k=1}^{j}\underline{H}_k\Big)\underline{H}_{j+i}$$

With these variable definitions, the determinant of the characteristic equation, equation I.4, can be written as

$$\det\{\underline{I}+(\underline{\Delta}'\underline{\Delta})^{-1}[\underline{A}_1z^{-1}+\underline{A}_2z^{-2}+\ldots+\underline{A}_jz^{-j}]\} \qquad \text{I.11}$$

Using Schur's formula, equation I.11 can be written as

$$\det\begin{vmatrix} \underline{I} & \underline{A}_1z^{-1}+\ldots+\underline{A}_jz^{-j} \\ -(\underline{\Delta}'\underline{\Delta})^{-1} & \underline{I} \end{vmatrix} \qquad \text{I.12}$$

The determinant given by equation I.5 was evaluated by demonstrating that the element $-\underline{b}'(\underline{K})^{-1}$ converged to the zero matrix. Similarly, the determinant given by I.12 is evaluated by showing that the element $-(\underline{\Delta}'\underline{\Delta})^{-1}$ becomes the null matrix as the horizon P is increased

$$\underline{\Delta}'\underline{\Delta}=\underline{H}_1'\underline{H}_1+\ldots+\sum_{j=1}^{N+\tau_0-1}\underline{H}_j'\sum_{j=1}^{N+\tau_0-1}\underline{H}_j+(P-N)\sum_{j=1}^{N+\tau_0}\underline{H}_j'\sum_{j=1}^{N+\tau_0}\underline{H}_j \qquad \text{I.13}$$

Equation I.13 is equivalent to

$$\underline{\Delta}'\underline{\Delta}=\underline{H}_1'\underline{H}_1+\ldots+\sum_{j=1}^{N+\tau_0-1}\underline{H}_j'\sum_{j=1}^{N+\tau_0-1}\underline{H}_j+(P-N)\underline{\acute{G}}'(1)\underline{\acute{G}}(1) \qquad \text{I.14}$$

Because the process transfer function is assumed to be stable, the steady state transfer function, $\underline{\acute{G}}(1)$, will be of full rank. Therefore the inverse of equation I.14 can be written as

$$(\underline{\Delta}'\underline{\Delta})^{-1}=$$
$$\left[\frac{1}{P-N}(\underline{\acute{G}}'(1)\underline{\acute{G}}(1))^{-1}\left[\underline{H}_1'\underline{H}_1+\ldots+\sum_{j=1}^{N+\tau_0-1}\underline{H}_j'\sum_{j=1}^{N+\tau_0-1}\underline{H}_j\right]+\underline{I}\right]^{-1}$$
$$\frac{1}{P-N}(\underline{\acute{G}}'(1)\underline{\acute{G}}(1))^{-1} \qquad \text{I.15}$$

From equation I.15, as $P\to\infty$, $(\underline{\Delta}'\underline{\Delta})^{-1}\to 0$. It should be possible to find a P>>M to satisfy the stability condition established for equation I.3.

J. Ability of the filter to provide robustness to the closed loop transfer function

This stability proof is restricted to situations where the plant/model mismatch satisfies

$$\mathrm{Re}\{\lambda_j[\underline{G}(1)\underline{\acute{G}}(1)^{-1}]\} > 0 \quad j=1,r \qquad \text{J.1}$$

where $\lambda_j[A]$ denotes the jth eigenvalue of A. The filter will be used to stabilize the closed loop transfer function when plant/model mismatch occurs. The proof uses a $\underline{\acute{G}}_-(z)$, based upon equation 3.2.1. From Figure 3.1, the open loop transfer function is given by

$$\underline{y}(z)=\underline{G}(z)\underline{\acute{G}}_-(z)^{-1}[\underline{I}-\underline{F}(z)\underline{\acute{G}}_{+1}]^{-1}\underline{F}(z)\underline{s}(z) \qquad J.2$$

Let $\underline{X}$ denote the open loop transfer function and $\omega(\underline{X})$ denote the open loop characteristic polynomial containing all the poles of $\underline{X}$. Similarly let $\psi(\underline{X})$ denote the closed loop characteristic polynomial. For the closed loop system to be stable, all the roots of $\psi(\underline{X})$ must be inside the unit circle. The closed loop transfer function is calculated from

$$\underline{y}(z)=\underline{X}(z)[\underline{I}+\underline{X}(z)]^{-1}\underline{s}(z) \qquad J.3$$

From equation J.3, the closed loop characteristic roots are calculated from

$$\psi(\underline{X})=\omega(\underline{X})\ \det(\underline{I}+\underline{X}) = 0 \qquad J.4$$

Since the proof requires that the process be stable, $\underline{G}(z)$ and $\underline{\acute{G}}(z)$ will have all poles inside the unit circle. For a stable process and process model, all roots of $\omega(\underline{X})$ are inside the unit circle with the exception of poles introduced at the unit circle by the integral controller. Because stability is determined by the unstable poles of the closed loop transfer function; to prove stability it is sufficient to demonstrate that the zeros of the return difference operator are stable. Given that the filter tuning parameter $\alpha_j \neq 0$ for all j, it is possible to write

$$\det\{\underline{I}+\underline{X}\}= =\det\left| \underline{I}+\underline{G}\underline{\acute{G}}^{-1}\underline{\acute{G}}_{+1}\ \{\underline{I}-\frac{1-\alpha *}{1-\alpha * z^{-1}}\underline{\acute{G}}_{+1}\}^{-1}\ \frac{1-\alpha *}{1-\alpha * z^{-1}}\ \underline{I}\right|$$

In the steps that follow, this equation is modified to produce a relation with a numerator and denominator expression. First, the second term in the determinant is multiplied by the factor

$$\frac{1-\alpha * z^{-1}}{1-\alpha * z^{-1}}$$

giving

$$=\det\{\underline{I}+\underline{G}\underline{\acute{G}}^{-1}\underline{\acute{G}}_{+1}z[(z-1)\underline{I}+(1-\alpha *)(\underline{I}-z\underline{\acute{G}}_{+1})]^{-1}(1-\alpha *)\underline{I}\}$$

In the next step, (z-1) is factored from the group of terms subjected to the inversion operation to yield

$$=\det\left| \underline{I}+\underline{G}\underline{\acute{G}}^{-1}\underline{\acute{G}}_{+1}(1-\alpha *)\frac{z}{z-1}\ [\underline{I}+\frac{1-\alpha *}{z-1}(\underline{I}-z\underline{\acute{G}}_{+1})]^{-1}\right|$$

The first element, $\underline{I}$, is multiplied by the terms enclosed by the inversion operation to give

$$=\det[\{\underline{I}+\frac{1-\alpha *}{z-1}(\underline{I}-z\underline{\acute{G}}_{+1})+\underline{G}\underline{\acute{G}}^{-1}\underline{\acute{G}}_{+1}(1-\alpha *)\frac{z}{z-1}\}\ \{\underline{I}+\frac{1-\alpha *}{z-1}(\underline{I}-z\underline{\acute{G}}_{+1})\}^{-1}]$$

The common $(1-\alpha *)\underline{I}$ is removed, resulting in

$$=\det\left|[(1-\alpha *)\underline{I}\ ][\frac{1}{1-\alpha *}\underline{I}+\ \{(\underline{I}-z\underline{\acute{G}}_{+1})+\underline{G}\underline{\acute{G}}^{-1}\underline{\acute{G}}_{+1}z\ \}\frac{1}{z-1}\underline{I}]\ [\underline{I}+\frac{1-\alpha *}{z-1}(\underline{I}-z\underline{\acute{G}}_{+1})]^{-1}\right|$$

It is now possible to write the determinant of the return difference operator as the quotient of two determinants

$$\frac{\det\left|\frac{1}{1-\alpha*}\underline{I}+[(\underline{I}-z\underline{\acute{G}}_{+1})+\underline{G}\underline{\acute{G}}^{-1}\underline{\acute{G}}_{+1}\}\frac{1}{z-1}\underline{I}\right|}{\det\left|\underline{I}+\frac{1-\alpha*}{z-1}\underline{I}(\underline{I}-z\underline{\acute{G}}_{+1})\right|}=0 \qquad J.5$$

so the denominator determinant is calculated entirely from diagonal matrices. The resulting polynomial will be finite everywhere except possibly at $z=1$. However, $\underline{\acute{G}}_{+1}(1)=\underline{I}$, so using L'Hopitals rule

$$\lim_{z\to 1} (\underline{I}-z\underline{\acute{G}}_{+1})/(z-1) \text{ is finite}$$

Theref[illegible]enominator of equation J.5 has no roots. To deter[illegible]ability of this equation it is sufficient to examine the roots of the numerator polynomial. Using the Nyquist stability criterion and the characteristic loci (CL) technique[29] the eigenvalues of

$$[\frac{1}{1-\alpha*}\underline{I}+\{(\underline{I}-z\underline{\acute{G}}_{+1})+\underline{G}\underline{\acute{G}}^{-1}\underline{\acute{G}}_{+1}\}\frac{1}{z-1}\underline{I}]$$

must not encircle the origin as z traverses the exterior of the unit circle. An equivalent criteria requires the eigenvalues of

$$[\{(\underline{I}-z\underline{\acute{G}}_{+1})+\underline{G}\underline{\acute{G}}^{-1}\underline{\acute{G}}_{+1}\}\frac{1}{z-1}\underline{I}] \qquad J.6$$

must not encircle $-1/(1-\alpha*)$ as z traverses the exterior of the unit circle. Because $\underline{G}$ and $\underline{\acute{G}}$ are both stable, the matrices will be finite over this contour. The only singularity of equation J.6 occurs at $z=1$. To examine the stability of the eigenvalues at z=1, the contour is indented

$$z=1+\epsilon e^{i\theta}, \quad -\frac{\pi}{2}<\theta<\frac{\pi}{2}$$

Thus as $\epsilon\to 0$

$$(\underline{I}-z\acute{\underline{G}}_{+1})\to 0$$

$$(\underline{G}\acute{\underline{G}}^{-1}\acute{\underline{G}}_{+1})\to(\underline{G}(1)\acute{\underline{G}}(1)^{-1})$$

so the eigenvalues of equation J.6 can now be written as

$$\lambda_j\{\underline{G}(1)\acute{\underline{G}}(1)^{-1}\}\frac{1}{\epsilon}e^{-i\theta} \qquad \text{J.7}$$

If $\lambda_j=|\lambda_j|\exp(-i\phi_j)$ then equation J.7 is equivalent to

$$|\lambda_j|e^{i\phi_j}\frac{1}{\epsilon}e^{-i\theta}$$

or

$$\frac{|\lambda_j|}{\epsilon}e^{-i(\phi_j-\theta)}$$

so provided equation J.1 is satisfied, the eigenvalues $\lambda_j\to+\infty$ as $z\to 1$. To summarize, all eigenvalues, λ_j, of equation J.6, are finite or tend to $+\infty$ as $z\to 1$. Thus an α^* exists such that the characteristic loci do not cross the real axis to the left of $1/(1-\alpha^*)$.

K. A Binary Ethanol-Water Distillation Column Model

The model for a 19 plate, 12 inch diameter column, relating outputs to inputs and disturbances was determined through pulse testing. The column had variable feed and side steam draw off locations. Further details are given by Ogunnaike et al[15]. The equation relating input to outputs was presented as

$$\underline{y}_m(s)=\underline{G}(s)\underline{m}(s)+\underline{G}_d(s)\underline{d}(s)$$

where

y_m is a vector of outputs

$\underline{m}$ is a vector of control inputs

$\underline{d}$ is a vector of measured disturbances

with the transfer functions $\underline{G}$, $\underline{G}_d$ given as

$$\underline{G}(s)=\begin{vmatrix} \dfrac{0.66e^{-2.6s}}{(6.7s+1)} & \dfrac{-0.61e^{-3.5s}}{(8.64s+1)} & \dfrac{-0.0049e^{-1.0s}}{(9.06s+1)} \\ \dfrac{1.11e^{-6.5s}}{(3.25s+1)} & \dfrac{-2.36e^{-3.0s}}{(5.0s+1)} & \dfrac{-0.012e^{-1.2s}}{(7.09s+1)} \\ \dfrac{-34.68e^{-9.2s}}{(8.15s+1)} & \dfrac{46.2e^{-9.4s}}{(10.9s+1)} & \dfrac{0.87(11.61s+1)e^{-s}}{(3.89s+1)(18.8s+1)} \end{vmatrix}$$

$$\underline{G}_d(s)=\begin{vmatrix} \dfrac{0.14e^{-12s}}{(6.2s+1)} & \dfrac{-0.0011(26.32s+1)e^{-2.66s}}{(7.85s+1)(14.63s+1)} \\ \dfrac{0.53s^{-10.5s}}{(6.9s+1)} & \dfrac{-0.0032(19.62s+1)e^{-3.44s}}{(7.29s+1)(8.94s+1)} \\ \dfrac{-11.54e^{-0.6s}}{(7.01s+1)} & \dfrac{0.32e^{-2.6s}}{(7.76s+1)} \end{vmatrix}$$

for

y_1=overhead ethanol mole fraction

y_2=side stream ethanol mole fraction

y_3=tray #19 temperature, °C (corresponding to bottoms composition)

u_1=reflux flow rate, gpm

u_2=side stream product flow rate, gpm

u_3=reboiler steam pressure, psig

d_1=feed flow rate, gpm

d_2=feed temperature, °C

L. Formulation of the DMC and MAC control laws

Dynamic Matrix Control

The dynamic matrix control technique is applicable to any stable system that can be described by or approximated by a series of linear differential equations. The DMC control law will be formulated for a two input, single output system. Let the inputs to this process be denoted I_1 and I_2. The output response, ø, generated by unit steps in these inputs is

$a_1\ a_2\ .\ .\ a_i$ for a step in I_1

$b_1\ b_2\ .\ .\ b_i$ for a step in I_2

By defining $\delta\phi_i$ as the change in output ø between its initial value and the value at interval i and ΔI_i^j as the change in input i between the intervals j-1 and j, the effect of any series of future inputs on the output ø can be computed. $\delta\phi_i$ can be written as

$$\delta\phi_1 = a_1\Delta I_1^1 + b_1\Delta I_2^1$$

$$\delta\phi_2 = a_2\Delta I_1^1 + a_1\Delta I_1^2 + b_2\Delta I_2^1 + b_1\Delta I_2^2$$

.

.

If input one is considered the manipulated input and input two considered a measurable disturbance, DMC reformulates the input-output relationship to provide for feedforward control on the disturbance input.

The disturbance input now appears on the left hand side of the equality sign. Because a good estimate of future disturbance inputs is the present disturbance value, all ΔI_2

for intervals after one, are set to zero. The set of equations, L.1, can now be written as

$$\delta\phi_1+b_1\Delta I_2=a_1\Delta I_1^1$$

$$\delta\phi_2+b_2\Delta I_2=a_2\Delta I_1^1+a_1\Delta I_1^2$$

$$\vdots \qquad \text{L.2}$$

Note the sign of the b series has been switched arbitrarily. A vector I of optimal future control inputs are determined to minimize the error between the desired and predicted outputs. That is

$$\frac{d}{dI}(E-AI)^t(E-AI) = 0 \qquad \text{L.3}$$

In equation L.3, E, I, and A are defined by

$$E=\begin{vmatrix} \delta\phi_{d1}+b_1\Delta I_2 \\ \delta\phi_{d2}+b_2\Delta I_2 \\ \vdots \end{vmatrix}$$

$$I^t=\begin{vmatrix} \Delta I_1^1 & \Delta I_1^2 & \Delta I_1^3 & \cdots \end{vmatrix}$$

$$A=\begin{vmatrix} a_1 & & & \\ a_2 & a_1 & & \\ a_3 & a_2 & a_1 & \\ & & \vdots & \end{vmatrix}$$

The solution to equation L.3 gives

$$I= -(A^tA)^{-1}A^tE \qquad \text{L.4}$$

Implementing this control law, leads to the unconstrained minimum variance controller. By multiplying the diagonal elements of the square matrix A^tA by a scaling factor greater than one, constraints are placed upon future control moves. And to minimize the error due to plant/model mismatch only the first element in the vector I is implemented before the control problem is reformulated. With these changes L.4 becomes

$$\Delta I_1 = -[1\ 0\ 0\ \ldots]'(A'A + \alpha A')^{-1}A'E \qquad \text{L.5}$$

$$A' = \mathrm{dia}[a_1^2\ \ a_1^2 + a_2^2\ \ + \ldots + \sum_{i=1}^{P} a_i^2]$$

Note, the diagonal matrix A' contains the diagonal elements of the matrix product, A'A.

Formulation of the MAC control law

The MAC technique is applicable to any stable process that can be described or approximated by a series of linear differential equations. The process behavior is modelled by an impulse response series. A block diagram representation of the MAC configuration is shown in Figure L.1. For a correct process model this algorithm is optimal because it minimizes the least square error between the predicted output and the setpoint reference trajectory. The reference trajectory is defined by

$$yr(k+j)=\alpha^{j}yr(k)+(1-\alpha^{j})C \qquad \text{L.6}$$

where C= change in setpoint

k=time interval when the change in setpoint occurs

α=tuning parameter between 0 and 1

The reference trajectory gives the control engineer the ability to shape the closed loop process response. In this case a first order path to the new setpoint has been selected. This path could be of higher order if desired.

The predicted process output is calculated from

$$yp(k+j)=y_{m}(k+j)+y(k)(1-\alpha)-y_{m}(k) \qquad \text{L.7}$$

Equation L.7 includes the feedback signal, $y(k)(1-\alpha)-ym(k)$, to prevent steady state offset when an unmeasured disturabnce or modelling error occurs. Again the parameter, α, is used to weight the feedback signal.

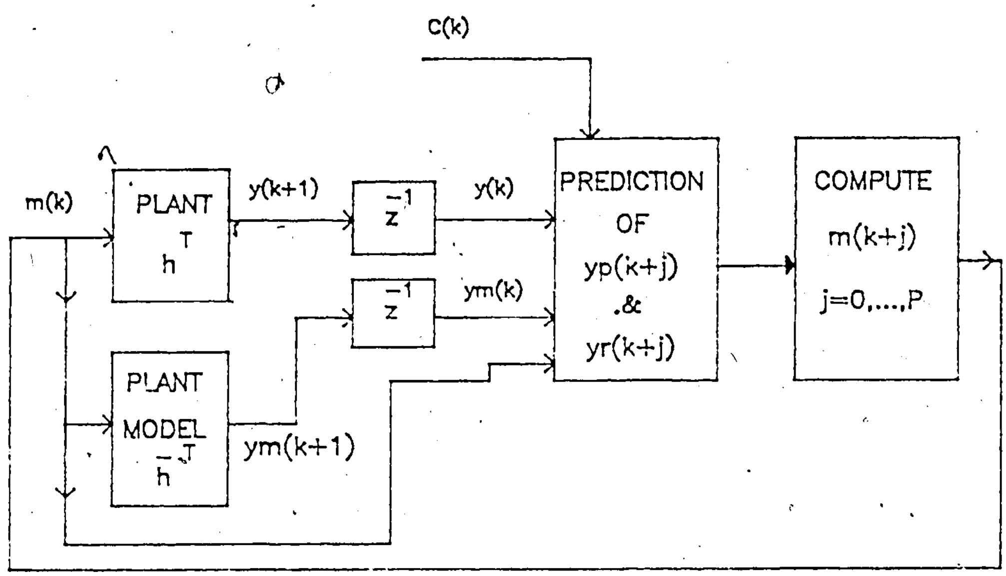

Figure L.1 Block diagram of the MAC feedback structure

The controller is formulated through least squares minimization of the cost function, J, defined by

$$J=\min\sum_{j=1}^{P}[yp(k+j)-yr(k+j)]^2 \qquad \text{L.8}$$

In this algorithm, like IMC, P is the optimization horizon.

For P=N, the solution to the minimization of equation L.8 is

$$m(k)=[1\ 0\ 0\ 0\ \ldots][\Lambda^t\Lambda]^{-1}\Lambda^t[E-\Psi V] \qquad \text{L.9}$$

The definitions for Λ, Ψ, and V are the same as the IMC definitions for these symbols. The exception, E, is given by

$$E=|C(1-\alpha^{2j-1})-(y(k)(1-\alpha)-y_m(k))|$$

When the horizon length is one, the minimization results in

$$0=\sum_{i=0}^{N}\hat{h}_i m(k-i)+\sum_{i=0}^{N}h_i(1-\alpha)m(k-1-i)-\sum_{i=0}^{N}\hat{h}_i m(k-1-i)-(1-\alpha)C$$

Solving for m(k), gives

$$m(k)=\frac{1}{\hat{h}_o}\Big\{\sum_{i=0}^{N}\hat{h}_i m(k-1-i)-\sum_{i=1}^{N}\hat{h}_i m(k-i)$$
$$+(1-\alpha)[C-\sum_{i=0}^{N}h_i m(k-1-i)]\Big\} \qquad \text{L.10}$$

By consideration of this form of control law, it can be noted that:

i. The speed with which the controller responds to changes in disturbances or setpoints is influenced by the choice of α. Furthermore, Rouhani and Mehra[18] have demonstrated that increasing α, increases the robustness of the closed loop system.

ii. Constraining of the input signal m(k) will lead to steady state offset.

iii. This formulation is unable to handle processes exhibiting NMP behavior because the transfer function zero, outside the unit circle, will cause the controller to become unstable.

For the minimum phase process, the MAC controller was designed based upon a model that best describes the true process dynamics. Optimal control, in the least square sense, occurs when $\hat{h}=h$. When the process exhibits NMP behavior, using $\hat{h}=h$, with equation L.9 or L.10, will result in an unstable controller. For the NMP case, $\hat{h}$ must be chosen to ensure stable minimization of the least square error. By redefining the cost function and vector of control inputs, linear quadratic control(LQC) theory can be applied to find the best $\hat{h}$. The vector of control inputs is defined in terms of a recursive state equation

$$M(k)=UM(k-1)+b_1[L^t \; 0]M(k-1) \qquad \text{L.11}$$

where $M(k)=[m(k) \; m(k-1) \; \ldots \; m(k-N-1)]$

$$U=\begin{vmatrix} 0 & 0 & \ldots & 0 \\ 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ 0 & 0 & 1 & \ldots \\ 0 & \ldots & 1 & 0 \end{vmatrix} \; \updownarrow N+2 \qquad b_1^t=\begin{vmatrix} 1 \\ 0 \\ 0 \\ \cdot \\ 0 \end{vmatrix} \; \updownarrow N+2$$

$$L^t=\frac{1}{\hat{h}_0}[\hat{h}_0-\hat{h}_1+(C-h_0)(1-\alpha), \ldots, \hat{h}_{n-1}-\hat{h}_n+(C-h_{n-1})(1-\alpha), \; \hat{h}_n+(C-h_n)] \qquad \text{L.12}$$

Selecting the cost function, J, to conform to

$$J=\frac{1}{2}\sum_{j=1}^{\infty} M'Q_1M + m(j)Q_2m(j)$$

permits the use of LQC algorithms. The new MAC cost function is given by

$$J=\frac{1}{2}\sum_{j=1}^{\infty}[y(i)-yr(i)]^2+m(j)Q_2m(j) \qquad \text{L.13}$$

$$=\frac{1}{2}\sum_{j=1}^{\infty}\ [\sum_{i=0}^{N}h_i m(j-i)-\{\alpha\sum_{i=0}^{N}h_i m(j-1-i)+(1-\alpha)C\}\ +m(j)Q_2m(j)] \qquad \text{L.14}$$

This cost function can be written in terms of the input vector M(k)

$$J=\frac{1}{2}\sum_{j=1}^{\infty}M(k)qq'M'(k)+(1-\alpha)C+m(j)Q_2m(j) \qquad \text{L.15}$$

where $q'=[h_0,h_1-\alpha h_0,\ldots,h_n-\alpha h_{n-1},-\alpha h_n]$ L.16

Q_2 is a scalar control weighting factor

m(j) is the optimal control input at interval k+j.

The LQC problem is now completely defined. Using equations L.15 and L.16, the matrix Ricatti equations or Hamilton's equations may be solved to find the steady state optimal control vector, L. Using the computed values for L, and equation L.12, the optimal, minimum phase impulse response series, h_i, can be calculated. An algorithm, outlined in Franklin and Powell[30], was used to solve the matrix Ricatti equations.

M. Description of the Solute Blending Tank

A solute blending tank, shown in Figure M.1, is used to represent a process with nonlinear dynamics. The blending tank has a total volume of 0.506m^3(approximately 134 us.gal.). The tank was sized to ensure the bulk flow rate of the exit stream never exceeded 2m/s.

Product concentration, c_3, is controlled by manipulating the flow rate of the second input stream, q_2. The first stream is considered to have a constant flow rate, q_1, and solute concentration, c_1.

On the basis of this information, assuming perfect mixing, the tank dynamics can be represented by the following differential equations

Overall mass balance

$$q_1\rho_1+q_2\rho_2=q_3\rho_3+\frac{d}{dt}(\rho_3 V) \quad \text{M.1}$$

Component balance

$$c_1q_1+c_2q_2=c_3q_3+\frac{d}{dt}(c_3V) \quad \text{M.2}$$

Because the solute concentrations are small, the first equation can be further simplified by assuming all stream densities are equal

$$q_1+q_2=q_3+\frac{d}{dt}V \quad \text{M.3}$$

The relationship between the volume of solution in the tank and the exit flow rate causes the nonlinear process dynamics. To relate exit flow rate to tank volume, the mechanical energy balance equation[31] is applied to a differential element of solution, flowing between the tank and the point of pipe discharge. The mechanical energy

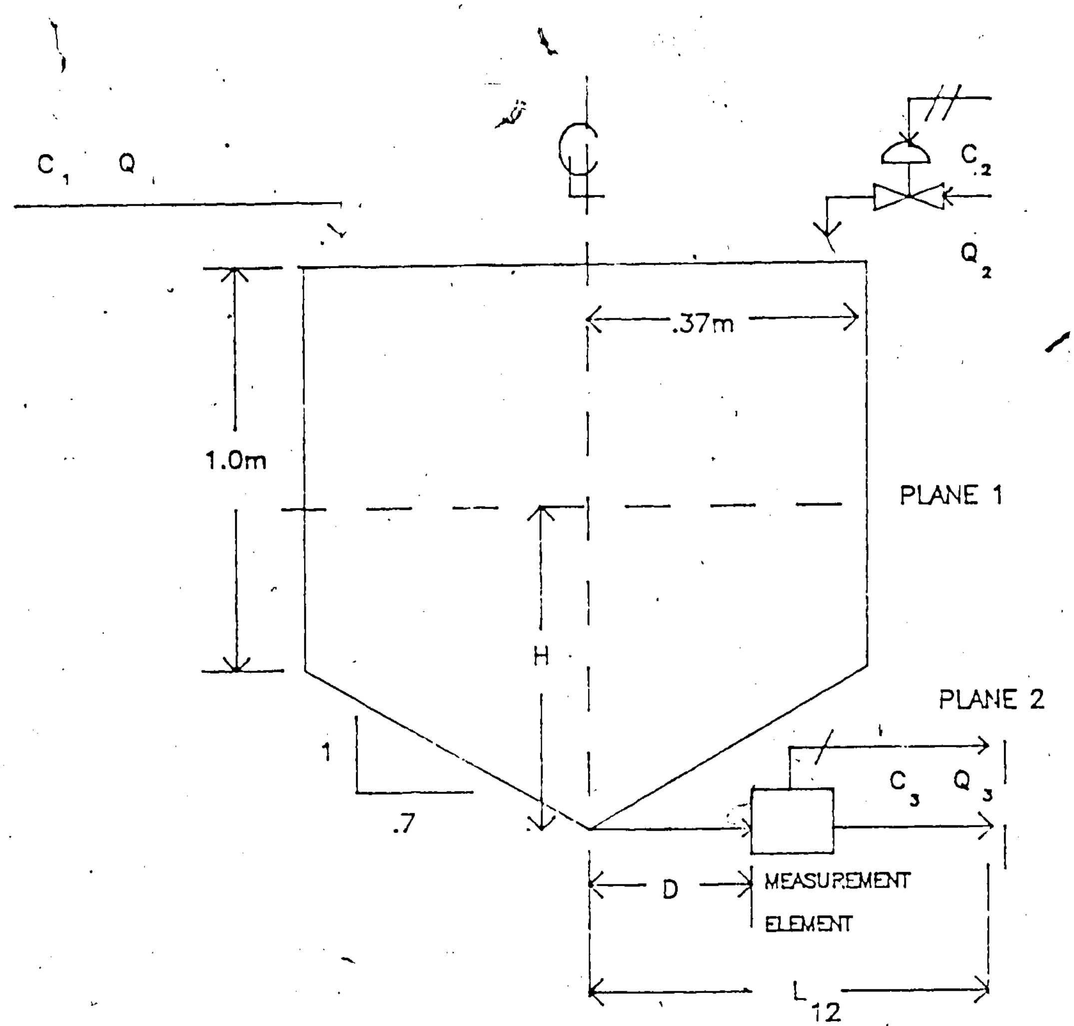

Figure M.1 Block diagram of blending tank. Measurement planes are indicated.

balance equation describes the pressure, kinetic, potential and frictional forces acting upon this differential mass.

$$v\,dP + U\,dU + g\,dz + d\,lwf = 0 \qquad \text{M.4}$$

Through integration, this equation is expanded to include all the forces acting on a unit mass of solution as it flows from the blending tank to the pipe exit to yield

$$v(P_2 - P_1) + \frac{U_2^2}{2} - \frac{U_1^2}{2} + g(z_2 - z_1) + lwf_{12} = 0 \qquad \text{M.5}$$

Where the subscripts 1 and 2 denote process variables at the tank and point of discharge respectively. Equation M.5 is further simplified by requiring:

i. Atmospheric pressure at the blending tank and at the point of discharge.

ii. The bulk flowrate, U_1 is negligible compared to U_2.

iii. The vertical distance between the point of discharge and the solution level, is equivalent to the height of solution in the blending tank.

With these assumptions, equation M.5 becomes

$$\frac{U_2^2}{2} - gh + lwf_{12} = 0 \qquad \text{M.6}$$

The Fanning friction factor, f, is used in the empirical equation

$$lwf_{12} = \frac{f U_2^2 L_{12}}{R} \qquad \text{M.7}$$

to determine the lost work due to friction. With an estimate of the Reynolds number, Re, a Moody diagram[31] can be used to specify a value for, f. Below is a table of parameters that specify the size and initial operation conditions for the blending tank

parameter	value [units]
Radius of discharge pipe	0.02m
Radius of cylindrical tank	0.35m
Height of cone section	0.5m
Distance of measuring element from tank	8.0m
Slope of cone=radius/height	0.7
Average fluid density	$1000.kg/m^3$
Equivalent length of pipe	18.m
Flow rate stream 1	$0.0005m^3/s$
Flow rate stream 3	0.0008 – $0.0009m^3/s$
Concentration stream 1	$20.kg/m^3$
Concentration stream 2	$50.kg/m^3$
Concentration stream 3	30.– $34.kg/m^3$
Bulk velocity of stream 3	0.60 – 0.73m/s
Sampling rate	0.5min/sample

The operating conditions for this tank correspond to a $Re \simeq 8000$, giving a $f=0.008$.

Because equation M.7 only computes lost work due to straight length of pipe, an additional pipe length or 'equivalent' length, Leq, is added to the actual length of pipe, L_{12} to account for the additional friction losses arising as the fluid enters the pipe entrance. Equation M.7 can be written as

$$lwf_{12}=\frac{fU_2^2(Leq+L_{12})}{R} \qquad M.8$$

Equation M.8 is substituted into M.6 to give

$$U_2=\sqrt{\frac{ghR}{R+f(Leq+L_{12})}} \qquad M.9$$

which is used to solve for U_2.

A FORTRAN subroutine was written to compute the tank response to changes in input flow rate. The differential equations, M.2 and M.3 were solved using finite differencing. The relationship between tank volume, V and output flow rate, q_3 was found using equation M.9.

N. The Wood-Berry Distillation Column Model

Based upon open loop pulse testing[22] a linear model was developed for a nine inch diameter, eight tray, methanol-water distillation column. The continuous transfer function model is reported as

$$\begin{vmatrix} y_1(s) \\ \\ y_2(s) \end{vmatrix} = \begin{vmatrix} \dfrac{12.8e^{-s}}{16.7s+1} & \dfrac{-18.9e^{-3s}}{21.0s+1} \\ \\ \dfrac{6.6e^{-7s}}{10.9s+1} & \dfrac{-19.4e^{-3s}}{14.4s+1} \end{vmatrix} \begin{vmatrix} x_1(s) \\ \\ x_2(s) \end{vmatrix}$$

where y_1=weight percent methanol in overhead product

y_2=weight percent methanol in bottoms product

x_1=reflux flow rate (lb/min)

x_2=steam flowrate (lb/min)

O. **Computer programs**

This section contains the FORTRAN code for the process simulator and the control algorithms. A copy of the terminal listing produced by the simulation results shown in Figure 2.5 is included to assist the reader in understanding the operation of the suite of programs.

The function of the different parts of the code as well as the definitions of the variables is provided by comments contained within the source code of each routine. In addition wherever possible, appropriate references have been included to facilitate understanding of program functions. At present, the simulator is set up to operate on the University of Alberta, Amdahl 5860 computer with its MTS operating system. However with the exception of the free read statements, all programming is carried out in FORTRAN IV coding.

The programs have been arranged such that the main program appears first followed by the various subroutines ordered alphabetically.

The first section of this appendix is the terminal listing for the simulation study that produced the response curves in Figure 2.5. To achieve these results, the command file MRUN1 was sourced twice. On the first pass, the pulse generating routine was used to prepare the impulse response series from the ARMA model and on the second pass, the IMC programs were used to generate the response curves. The same results could have been prepared by sourcing the command file MRUN1 just once but this, for reasons left unsaid, makes the output file -9 very long. Throughout the listing, annotations have been added to help the reader. These notes are provided in *italics* to differentiate them from the terminal output, or user input.

INDEX

The file MIMC.DAT7 is edited to ensure the beta and gamma weightings are correct. The format for this file can be found in the source code listing for MIMC1.FOR

```
#ed mimc.dat7
:p1 20
:        1      0.0 1.0
:        2      0.0 1.0
:        3      0.0 1.0
:        4      0.0 1.0
:        5      0.0 1.0
:        6      0.0 1.0
:        7      0.0 1.0
:        8      0.0 1.0
:        9      0.0 1.0
:       10      0.0 1.0
:       11      0.0 1.0
:       12      0.0 1.0
:       13      0.0 1.0
:       14      0.0 1.0
:       15      0.0 1.0
:       16      0.0 1.0
:       17      0.0 1.0
:       18      0.0 1.0
:       19      0.0 1.0
:       20      0.0 1.0
:st
#    $.05,    $.05T
```

The command file MRUN1 is sourced to determine the impulse response series. The sampling rate, the number of rows, columns and highest order of the transfer function matrix, along with the number of integer time delays and analog model parameters must be provided.

Two time delays are required, one for the control sampling rate, T(1), and one for the intersample sampling rate, T(2)=.2T(1). For example, if the analog dead time is 6 min, T(1)=10min, giving T(2)=2min, then the dead time as observed by a sampling rate of T(1) can be approximated as either 0 or 1. If 0, is chosen then the number of integer delays corresponding to T(2) will be 3. On the other hand, *if the number of integer delays corresponding to T(1) were chosen as 1, then the number of integer delays corresponding to T(2) would be, -2.*

```
#so mrun1
#    $.00,     $.06T
# $em -9
# "-9" does not exist.
#    $.03,     $.09T
# $em -11
# "-11" does not exist.
#    $.03,     $.12T
# $em -l
# "-L" does not exist.
#    $.03,     $.16T
# $copy multi.obj to -l(*L+1)
#    $.10,     $.26T
# $copy mztr.obj to -l(*L+1)
#    $.01,     $.27T
# $copy mol.obj to -l(*L+1)
#    $.01,     $.28T
# $copy mpul.obj to -l(*L+1)
#    $.01,     $.29T
# $copy mdat.obj to -l(*L+1)
#    $.01,     $.30T
# $copy mimc1.obj to -l(*L+1)
#    $.16,     $.46T
# $r -l+*imsllib+minit.obj+mimc2.obj+morc.obj+...
# 22:47:34
:
  enter the sampling rate, T
?4.0
  enter the simulation type ,linear=1, nonlinear=2
?1

  enter the number of rows, IR where IR<4
  enter the number of columns, IC where IC<4
  enter the highest model order, IO where IO =1 or 2
  enter these three values on the next line
  if ISIM=2 IR=IC=IO=1
?1,1,2

  below are the eight transfer function types
  1. G(s) = K/s
  2. G(s) = K/(s+r)
  3. G(s) = K*omega**2/( s*s + 2*zeta*omega*s +omega**2)
  4. G(s) = K/( (s+r)(s+p) )
  5. G(s) = K/( s(s+r) )
  6. G(s) = K (s+q)/( (s+r)(s+p) )
  7. G(s) = K (s+q)/(s(s+r))
  8. G(s) = K (s+q)/(s+r)
  enter the element time delay for rate T(1)
  the extra delay required for for rate T(2)
```

```
  and the transfer function type(1-8)
       element        1     1
?1,0,4
  enter: K, R, P
?0.1,.1,1.

  menu of program options
  block 1-open loop response for step response
  block 2- open loop response to a pulse test
  block 3- IMC control law
  block 4-  RC control law, delay free case
  block 5- ORC control law, delay included
  block 6 - ORC - control law, delay free and delayed
  block 7 - MAC -NMP control law
  block 8 - cross correlation
  block 15 - exit program
  enter block number on next line
?2

  input the pulse size for input       1
?2.

  menu of program options
  block 1-open loop response for step response
  block 2- open loop response to a pulse test
  block 3- IMC control law
  block 4- ORC control law, delay free case
  block 5- ORC control law, delay included
  block 6 - ORC - control law, delay free and delayed
  block 7 - MAC -NMP control law
  block 8 - cross correlation
  block 15 - exit program
  enter block number on next line
?15
# 22:48:25  T=0.73  RC=0
#    $.28,    $.74T
```

File MIMC.DAT10 is viewed to demonstrate format of the impulse response series coefficients. Details on the structure of this file can be found in the source code listing for MULTI.FOR.

```
#ed mimc.dat10
:p1 20
:       1         0.0
:       2         0.2572
:       3         0.2435
:       4         0.1646
:       5         0.1103
:       6         0.0740
:       7         0.0496
:       8         0.0332
```

```
:       9             0.0223
:      10             0.0149
:      11             0.0100
:      12             0.0067
:      13             0.0045
:      14             0.0030
:      15             0.0020
:      16             0.0014
:      17             0.0009
:      18             0.0006
:      19             0.0004
:      20             0.0003
:st
#    $.01,     $.75T
```

The file -11 is viewed to illustrate the structure. Additional information on the structure of this file can be found in the source code for MULTI.FOR and MZTR.FOR.

```
#ed -11
:p/f
:       1        4.00000
:       2        1
:       3        1    1    2
:       4          1         0         4
:       5        0.10000    0.10000    1.00000
:       6          2
:       7           2.0000
:       8         15
:st
#    $.01,     $.75T
```

The command file MRUN1 is sourced again to evaluate the performance of the IMC algorithm. The control parameters M=P=N=10 are required along with setpoints, time delay estimates and filter parameters.

```
#so mrun1
#    $.00,     $.75T
# $em -9
# Done.
#    $.00,     $.76T
# $em -11
# Done.
#    $.00,     $.76T
# $em -l
# Done.
#    $.00,     $.76T
# $copy multi.obj to -l(*L+1)
#    $.05,     $.82T
# $copy mztr.obj to -l(*L+1)
```

```
#    $.01,     $.83T
# $copy mol.obj to -l(*L+1)
#    $.01,     $.84T
# $copy mpul.obj to -l(*L+1)
#    $.01,     $.84T
# $copy mdat.obj to -l(*L+1)
#    $.01,     $.85T
# $copy mimc1.obj to -l(*L+1)
#    $.14,     $.99T
# $r -l+*imsllib+minit.obj+mimc2.obj+morc.obj+...
# 22:49:25

  enter the sampling rate, T
?4.
  enter the simulation type ,linear=1, nonlinear=2
?1

  enter the number of rows, IR where IR<4
  enter the number of columns, IC where IC<4
  enter the highest model order, IO where IO =1 or 2
  enter these three values on the next line
  if ISIM=2 IR=IC=IO=1
?1,1,2

  below are the eight transfer function types
  1. G(s) = K/s
  2. G(s) = K/(s+r)
  3. G(s) = K*omega**2/( s*s + 2*zeta*omega*s +omega**2)
  4. G(s) = K/( (s+r)(s+p) )
  5. G(s) = K/( s(s+r) )
  6. G(s) = K (s+q)/( (s+r)(s+p) )
  7. G(s) = K (s+q)/(s(s+r))
  8. G(s) = K (s+q)/(s+r)
  enter the element time delay for rate T(1)
  the extra delay required for for rate T(2)
  and the transfer function type(1-8)
       element       1     1
?1,0,4
  enter: K, R, P
?.1,.1,1.

  menu of program options
  block 1-open loop response for step response
  block 2- open loop response to a pulse test
  block 3- IMC control law
  block 4- ORC control law, delay free case
  block 5- ORC control law, delay included
  block 6 - ORC - control law, delay free and delayed
  block 7 - MAC -NMP control law
  block 8 - cross corrrelation
  block 15 - exit program
  enter block number on next line
?3
  enter the setpoints, one per line
```

```
?1.
  enter IMC tuning  parameters
  P- optimization horizon
      where P<or=N and (P+T0)*outputs<90
  M- input suppression  parameter M*inputs<or=P<90
  N- number of terms in impulse response model
     where N+Tdelay<70 and (N+T0)*inputs<120
  TO- measure of imbalance in TF
  enter these four integer values on the next line
?10,10,10,0
  enter the time delay diagonal elements, 1 per line
?1
  enter filter coefficients, one per line
?0.0
  do you want offset compensation yes(1), no(2)
?2

  menu of program options
  block 1-open loop response for step response
  block 2- open loop response to a pulse test
  block 3- IMC control law
  block 4- ORC control law, delay free case
  block 5- ORC control law, delay included
  block 6 - ORC - control law, delay free and delayed
  block 7 - MAC -NMP control law
  block 8 - cross corrrelation
  block 15 - exit program
  enter block number on next line
?15
# 22:50:29  T=0.827  RC=0
#    $.29,   $1.28T
```

The file -11 is viewed to illustrate the structure. The response shown in Figure 2.6 could be generated by editing this file and changing the third 10 in line 8 to a 15, after which command file MRUN2 is sourced.

```
#ed -11
:p/f
:      1        4.00000
:      2        1
:      3        1    1    2
:      4          1      0       4
:      5        0.10000   0.10000   1.00000
:      6          3
:      7            1.0000
:      8        10    10    10    0
:      9         1
:     10            0.0
:     11         2
:     12          15
:st
```

```
#    $.01,    $1.29T
```

Finally, the file -9, is viewed to illustrate the structure of the output data.

```
#ed -9
:p1 20
:        1          0.0          0.0          3.89
:        2          0.80         0.0          3.89
:        3          1.60         0.0          3.89
:        4          2.40         0.0          3.89
:        5          3.20         0.0          3.89
:   6    4.00  0.0    3.89  4.00    15.55
:        7          4.00         0.0          0.21
:        8          4.80         0.09         0.21
:        9          5.60         0.29         0.21
:       10          6.40         0.53         0.21
:       11          7.20         0.77         0.21
    12    8.00   1.00  0.21  5.85    30.28
       13           8.00         1.00         1.20
       14           8.80         1.13         1.20
       15           9.60         1.14         1.20
:      16          10.40         1.11         1.20
:      17          11.20         1.06         1.20
:   18      12.00  1.00   1.20   6.20    34.26
:      19          12.00         1.00         0.95
:      20          12.80         0.97         0.95
:st
#    $.01,    $1.30T
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                     C
C   Mainline program for multivariable simulator      C
C    MULTI.FOR                                        C
C   Subroutines  Note: in most cases the              C
C      title of the subroutine file name, will be the C
C      name of the subroutine as it is called in the  C
C      main program, prefaced by the letter M. For    C
C      example the mainline program calls subroutine  C
C      IMC1. The title given to this subprogram file  C
C      is MIMC1.FOR. There are exceptions.            C
C   ZTRANS - program MZTR.FOR, converts analog model  C
C            to discrete model                        C
C   STEP - program MOL.FOR, initializes input vector  C
C          for a step response                        C
C   PULSE - program MPUL.FOR, initializeds input vectorC
C          for a pulse test                           C
C   DATA - program MDAT.FOR, computes the process     C
C          output given the inputs                    C
C          For ISIM=1 a linear function is assumed,   C
C          and a three by three transfer function is  C
C          permissible                                C
C          For ISIM=2 the SISO blending tank is       C
C          assumed, therfore only a 1 input 1 output  C
C          relationship is possible                   C
C   IMC1 - program MIMC1.FOR, computes the return     C
C         difference operator RDO, in the IMC control C
C         law.                                        C
C   IMC2 - program MIMC2.FOR, computes the control    C
C         signal u(k)                                 C
C   ORC - program MORC.FOR, computes the control signalC
C        for the Ogunnaike and Ray compensator        C
C   INIT-MINIT.FOR initializes the control parameters C
C           for the IMC, ORC and MAC algorithms       C
C   CROSS-MCROSS.FOR carries out the cross correlation C
C   MACNMP - MMACNMP.FOR computes the best stable,    C
C            minimum phase approximation of the NMP   C
C            impulse response series using LQC        C
C            theory                                   C
C   MACCL - MMACCL.FOR computes the control signal for C
C          the MAC algorithm (SISO only)              C
C   NLSS - MNLSS.FOR computes the steady state        C
C          operating conditions for the blending tank C
C   The IMSL routines include;                        C
C       LINV1F- carries out matrix inversion (in IMC1) C
C       GGNML- generates a normally distributed,      C
C              (variance=mean=1) signal.(in DATA)     C
C                                                     C
C   Operation of Program                              C
C   All of the programs included in this package are  C
C   written in FORTRAN IV with the exception of free  C
C   read statements used to input data from the       C
C   operator's console. In addition the IMSL library  C
C   of subroutines is accessed to carry out some      C
```

```
C   linear algebra operations.                            C
C                                                         C
C  To run the program                                     C
C   SO MRUN1                                              C
C     This file contains commands that initialize the    C
C     the various resources needed by the main           C
C     program. Since the programs are all menu driven    C
C     the operator will be requested to enter the        C
C     required parameters. Note: all inputs are          C
C     written into the temporary file -11 attached to    C
C     LU 11                                               C
C     All of the calculated output values are written    C
C     to the temporary file -9 attached to LU 9 .        C
C     The operators terminal is LU 5 and LU 6 for read   C
C     and writes respectvely                              C
C     After the first run is completed, make changes     C
C     to  -11 which will be used by another command      C
C     file MRUN2 for subsequent runs.                     C
C    SO MRUN2                                             C
C     Use MRUN2 for all runs following the                C
C     first run. It is easier to make any changes in     C
C     the run parameters, by editing file -11 than       C
C     going through the tedious task of reentering all   C
C     parameters interactively.                           C
C                                                         C
C    READ or WRITE files that must be available           C
C     MIMC.DAT7-contains the beta and gamma weightings   C
C                see MIMC1.FOR for explaination of stru- C
C                ture.                                    C
C     MIMC.DAT10- contains the output from either step,  C
C                 pulse, or cross correlation testing.   C
C                 ie it contains the nonparametric modelC
C     MIMC.DAT13- contains the initial steady state      C
C                 operating conditions for the blending  C
C                 tank as well as the other computed     C
C                 initial steady state conditions. See   C
C                 MNLSS.FOR for the structure of this    C
C                 file.                                   C
C      -9 - contains the calculated output and input. SeeC
C           MDAT.FOR for the structure of this file      C
C -12,-11 -contains the parameters entered by the        C
C            process operator. See ZTRANS.FOR and        C
C            MULTI.FOR for the structure of this file    C
C                                                         C
C    Variable ID                                          C
C   T(n) - vector of sampling rates                       C
C        T(1) - control sampling rat[illegible]           C
C        T(2) - .2T(1) records in[illegible]   behavior   C
C   GN(n,i,j,k) - matrix containing n[illegible]or ARMA   C
C        coefficents                                      C
C        n - corresponds to sampling ra[illegible]        C
C        i - corresponds to row                           C
C        j - corresponds to column                        C
C        k - corresponds to SISO para[illegible]          C
```

```
C  GD(n,i,j,k) - matrix containing denominator ARMA    C
C       coefficients                                   C
C       indices are same as above                      C
C  GP(j) - vector containing diagonal elements of      C
C         optimal time delay compensator               C
C  Y(i,l) - matrix of process outputs                  C
C       l - refers to 1 of 5 intersample times         C
C  U(j,m) - matrix of control inputs                   C
C       m - refers to 1 of 30 possible inputs          C
C         - where 1 is the present input               C
C  ITD(i,j) - matrix of time delays                    C
C  H(i,j,k) - impulse reponse matrix                   C
C         k - corresponds to term in response series   C
C             subroutine OL                            C
C                                                      C
C  IR - number of rows in TF                           C
C  IC - number of columns in TF                        C
C  IO - order of ARMA model                            C
C  ISIM - determines whether the simulation is linear  C
C         or nonlinear 2. Note if ISIM=2, IR=1, IC=1,  C
C         IO=1, otherwise the program will not execute C
C         properly                                     C
C  ICASE - determines the transfer of control through  C
C          the main program as well as the             C
C          initialization program INIT.                C
C                                                      C
C   Additional information about any of the           C
C   subroutines called by the main program can be foundC
C   in the source code listings for those subprograms  C
C                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      DIMENSION GN(2,3,3,2), GD(2,3,3,2), T(2), ITD(3,3),
     1      U(3,70), Y(3,5), USTEP(3), H(3,3,70), OFF(3,3)
     2      , GAMMA(180), CONST(90,90), GP(3), PSI(60,120)
     3      , FILTER(3), ITD2(3,3)
     4      , IOR(3,3), IDF(3,3), CK(3,3), CI(3,3)
     5      , GI(3,3), HK(50),A(7),YD(3)
      INTEGER P, GP
      REAL PSI
C
C initialization
      DATA GN/36*0.0/, GD/36*0.0/, T/2*0.0/, ITD/9*0/,
     1      U/210*0.0/, Y/15*0.0/, USTEP/3*0.0/,
     2      H/630*0.0/, GAMMA/180*0.0/, CONST/8100*0.0/
     3      , PSI/7200*0.0/, FILTER/3*0.0/,GP/3*0/
     4      , ITD2/9*0/, OFF/1.0,3*0.0,1.0,3*0.0,1.0/
     5      , IDF/9*0/, CK/9*0.0/, CI/9*0.0/, IOR/9*0/
     6      , GI/9*0.0/, ITEST/0/, HK/50*0.0/
     7      , A/1.0,-1.0,1.0,1.0,1.0,-1.0,-1.0/, ISIM/0/
     8      , YD/3*0.0/
C
C call subroutine ZTRANS to determine discrete
C representation
```

```
      CALL ZTRANS(IR,IC,IO,ITD,GN,GD,T, ITD2,ISIM)
C
    5 CONTINUE
C determine which function the program is to perform
      WRITE(6,100)
      CALL FREAD(5,'(I):', ICASE)
      WRITE(11,101)ICASE
      IF(ICASE .EQ. 15) GOTO 99
C
C transfer control via the implied goto
      GOTO(20,20,27,39,39,39,50,60), ICASE
C
   20 CONTINUE
C ..........................................................
C . Block 1 and Block 2                                    .
C . this block calculates the open loop response           .
C .                                                        .
C ..........................................................
      DO 25 J=1, IC
        IF(ICASE .EQ. 1) CALL STEP(J,U)
        IF(ICASE .EQ. 2) CALL PULSE(J,U)
        DO 24 K=1, 70
          CALL DATA(IR,IC,IO,ITD,GN,GD,U,T,Y,K,ITD2,ISIM,YD)
          DO 21 M=1, IR
            H(M,J,K)=Y(M,5)/U(J,K)
   21     CONTINUE
   22     CONTINUE
          DO 23 L=1, 69
            U(J,71-L)=U(J,70-L)
   23     CONTINUE
          IF(ICASE .EQ. 2) U(J,1)=0.0
   24   CONTINUE
   25 CONTINUE
C write step or impulse model ro LU 10
      DO 26 K=1, 70
        DO 26 I=1, IR
          WRITE(10,102) (H(I,J,K),J=1, IC)
   26 CONTINUE
      GOTO 5
   27 CONTINUE
C ..........................................................
C . Block 3                                                .
C . This block computes the IMC control signal             .
C .                                                        .
C ..........................................................
C call INIT to initial the IMC tuning parameters
      CALL INIT(ICASE,IC,P,M,N,TO,GP, FILTER,IOFF,
     1      IOR,CK,CI,IR,GI,YD)
C initialize U and Y to zero
      DO 29 I=1, IR
        DO 28 J=1, 5
          Y(I,J)=0.0
   28   CONTINUE
   29 CONTINUE
```

```
      DO 30 I=1, IC
        DO 30 J=1,70
          U(I,J)=0.0
   30 CONTINUE
C read the impulse response  coefficents from
C logical unit 10(MIMC.DAT10)
      REWIND10
      REWIND9
      NPTO=N+TO
      DO 32 K=1, NPTO
        DO 31 I=1, IR
          READ(10,102)(H(I,J,K),J=1,IC)
   31   CONTINUE
   32 CONTINUE
C call IMC1 to calculate CONST, PSI and GAMMA
      CALL IMC1(IR,IC,N,M,P,TO,H,GP,CONST,GAMMA,
     1      PSI, OFF,IOFF)
C calculate the IMC control performance
      DO 37 L=1, 100
C compute u(k)
      CALL IMC2(IR,IC,N,M,P,TO,CONST,GAMMA,PSI,Y,U,
     1     H, OFF, FILTER,YD)
C call data to calculate y(k+1)
      CALL DATA(IR,IC,IO,ITD,GN,GD,U,T,Y,L,ITD2,ISIM,YD)

   37 CONTINUE
      GOTO 5
C
   39 CONTINUE
C ..................................................
C . Block 4,5 and 6                                .
C . This block computes the ORC control signal     .
C .                                                .
C ..................................................
C call INIT to compute the ORC tuning parameters
      CALL INIT(ICASE,IC,P,M,N,TO,GP,FILTER,IOFF,
     1      IOR,CK,CI,IR,GI,YD)
   40 CONTINUE
C initialize U and Y to zero
      DO 42 I=1, IR
        DO 41 J=1,5
          Y(I,J)=0.0
   41   CONTINUE
   42 CONTINUE
      DO 44 I=1,IC
        DO 43 J=1,70
          U(I,J)=0.0
   43   CONTINUE
   44 CONTINUE
C compute the performance of controller  ning parameters
C if ICASE equals 4 - delay free case
      IF(ICASE-5)45,47,45
C delay free case
   45 CONTINUE
```

```
      DO 46 L=1, 50
C compute u(k)
      CALL ORC(IR,IC,IO,IDF,GN,GD,Y,U,CK,CI,IDF,GI,L,YD)
C call DATA to calculate y(k+1)
      CALL DATA(IR,IC,IO,IDF,GN,GD,U,T,Y,L,IDF,ISIM,YD)
   46 CONTINUE
      IF(ICASE .EQ. 4) GOTO 5
      ICASE=5
      GOTO 40
C response for delayed process
   47 CONTINUE
      DO 48 L=1, 50
C compute u(k)
      CALL ORC(IR,IC,IO,ITD,GN,GD,Y,U,CK,CI,IOR,GI,L,YD)
C compute y(k+1)
      CALL DATA(IR,IC,IO,ITD,GN,GD,U,T,Y,L,ITD2,ISIM,YD)
   48 CONTINUE
      GOTO 5
C
   50 CONTINUE
C ..................................................
C . Block 7                                          .
C . This block computes the MAC control signal for.
C . a NMP system                                     .
C .                                                  .
C ..................................................
C call INIT to determine the MAC contol parameters
      CALL INIT(ICASE,IC,P,M,N,TO,GP,FILTER,IOFF,
     1      IOR,CK,CI,IR,GI,YD)
C initialize U and Y to zero
      DO 54 I=1, IR
        DO 53 J=1, 5
          Y(I,J)=0.0
   53   CONTINUE
   54 CONTINUE
      DO 56 I=1, IC
        DO 55 J=1,70
          U(I,J)=0.0
   55   CONTINUE
   56 CONTINUE
C read the impulse response coefficents from logical
C unit 10(MIMC.DAT10)
      REWIND10
      DO 58 K=1, 60
        DO 57 I=1, IR
          READ(10,102)(H(I,J,K),J=1,IC)
   57   CONTINUE
   58 CONTINUE
C call MACNMP to compute the MAC contoller gain HK
      CALL MACNMP(N,FILTER,H,GP,HK)
C calculate the MAC-NMP control performance
      DO 59 L=1, 50
C compute u(k)
      CALL MACCL(N,HK,U,Y,FILTER,GP,YD)
```

```
C call DATA to calculate y(k+1)
      CALL DATA(IR,IC,IO,ITD,GN,GD,U,T,Y,L,ITD2,ISIM,YD)
   59 CONTINUE
      GOTO 5
   60 CONTINUE
C ..........................................................
C . Block 8                                                .
C . This block identifies the process model                .
C . through cross correlation                              .
C ..........................................................
      REWIND10
      DO 69 J=1,IC
C begin by initializing the output vector to zero
        DO 62 I=1,IR
          A(1)=1.0
          A(2)=-1.0
          A(3)=1.0
          A(4)=1.0
          A(5)=1.0
          A(6)=-1.0
          A(7)=-1.0
          DO 61 IJ=1,5
            Y(I,IJ)=0.0
   61     CONTINUE
   62   CONTINUE
C initialize the input vector to zero
        DO 64 L=1, 70
          DO 63 IJ=1,IC
            U(IJ,L)=0.0
   63     CONTINUE
   64   CONTINUE
        DO 68 L=1,400
          U(J,1)=A(1)
C store u(k),y(k) in logical unit 10 attached to MIMC.DAT10
          DO 65 I=1,IR
            WRITE(10,103)A(1),(Y(I,5),I=1,IR)
   65     CONTINUE
C call DATA to generate y(k+1)
          CALL DATA(IR,IC,IO,ITD,GN,GD,U,T,Y,L,ITD2,ISIM,YD)
C update the maximum length function
          TEMP=A(6)*A(7)
          DO 66 I=1,6
            A(8-I)=A(7-I)
   66     CONTINUE
          A(1)=TEMP
C update the control vector
          DO 67 IJ=1,69
            U(J,71-IJ)=U(J,70-IJ)
   67    CONTINUE
   68   CONTINUE
   69 CONTINUE
C call CROSS to calculate the cross correlation function
C and the best estimate of H
      CALL CROSS(IR,IC,T)
```

```
      GOTO 5
C terminate program
   99 CONTINUE
      STOP
C
C format lines
  100 FORMAT(/' menu of program options ',/,
     1 ' block 1-open loop response for step response ',/,
     1 ' block 2- open loop response to a pulse test ',/,
     2 ' block 3- IMC control law ',/,
     3 ' block 4- ORC control law, delay free case ',/,
     4 ' block 5- ORC control law, delay included ',/,
     5 ' block 6 - ORC - control law, delay free and',
     5 ' delayed',/,
     6 ' block 7 - MAC -NMP control law',/,
     7 ' block 8 - cross corrrelation ',/,
     8 ' block 15 - exit program ',/,
     9 ' enter block number on next line')
  101 FORMAT(2X,2I4)
  102 FORMAT(2X,4(2X,F8.4))
  103 FORMAT(2X,4(2X,F10.6))
 END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C CROSS                                                C
C   Subroutine MCROSS.FOR, May 7 1984                  C
C    - subroutine calculates the cross correlation     C
C      function and the corresponding best estimate of C
C      the impulse response series                     C
C   Reference: Box, G.E.P., Jenkins, G.M.; 'Time SeriesC
C   Analysis forecasting and control' Rev Ed.          C
C   Holden Day ,377-387,(1976)                         C
C                                                      C
C                                                      C
C   Variable ID                                        C
C   U-matrix of inputs, present and past               C
C   H- matrix of impulse response series coefficents   C
C   PHI-matrix of cross covariance coefficients        C
C   T- the sampling rate                               C
C   H1 - mean of process output                        C
C   U1 - mean of process input                         C
C   PHIX - cross covariance of the input u             C
C   PHIY - cross covariance of the output y            C
C   RXY - cross correlation coefficient estimate       C
C                                                      C
C   IR -number of rows in transfer function matrix     C
C   IC - number of columns in transfer function matrix C
C                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE CROSS(IR,IC,T)
      DIMENSION T(3)
      REAL *8 PHI(3,3,50),H(3,3,400),U(400)
     1    , RXY(3,3,50),H1(3,3),PHIY(3,3),U1,PHIX
      DATA PHI/450*0.0D0/,RXY/450*0.0D0/,U1/0.0D0/
     1     ,U/400*0.0D0/,H/3600*0.0D0/,PHIX/0.0D0/
     2     , PHIY/9*0.0D0/, H1/9*0.0D0/
C calculate the coefficents of U and read the process
C outputs from logical unit 10 attached to MIMC.DAT10
      REWIND10
      DO 13 J=1,IC
        U1=0.0D0
        DO 12 L=1,400
          READ(10,100)U(L),(H(I,J,L),I=1,IR)
          U1=U1+U(L)
          DO 11 I=1,IR
            H1(I,J)=H1(I,J)+H(I,J,L)
   11     CONTINUE
   12   CONTINUE
        U1=U1/400.0D0
        DO 13 I=1,IR
          H1(I,J)=H1(I,J)/400.0D0
   13 CONTINUE
      REWIND10
C calculate the cross covariance coefficent
      DO 25 I=1,IR
        DO 24 J=1,IC
          DO 18 K=1,50
```

```
            NMK=400-K
            DO 17 L=1,NMK
              PHI(I,J,K)=PHI(I,J,K)+(U(L)-U1)*(H(I,J,L+K)-
     1        H1(I,J))
   17       CONTINUE
            PHI(I,J,K)=PHI(I,J,K)/400.0D0
   18     CONTINUE
   24   CONTINUE
   25 CONTINUE
C calculate the variance of the input
      DO 26 L=1,400
        PHIX=PHIX+(U(L)-U1)*(U(L)-U1)
   26 CONTINUE
      PHIX=PHIX/400.0D0
C calculate the vairiance of the output
      DO 30 I=1, IR
        DO 29 J=1,IC
          DO 28 L=1,400
            PHIY(I,J)=PHIY(I,J)+(H(I,J,L)*(H(I,J,L)-H1(I,J))
     1      -H1(I,J))
   28     CONTINUE
          PHIY(I,J)=PHIY(I,J)/400.0D0
   29   CONTINUE
   30 CONTINUE
C calculate the cross correlation coefficent estimate
      DO 35 I=1,IR
        DO 34 J=1,IC
          DO 33 L=1,50
            RXY(I,J,L)=PHI(I,J,L)/(DSQRT(PHIX)*
     1                 DSQRT(PHIY(I,J)))
              H(I,J,L)=RXY(I,J,L)*DSQRT(PHIY(I,J))/(
     1                 DSQRT(PHIX))
   33     CONTINUE
   34   CONTINUE
   35 CONTINUE
C write impulse response series coefficents to LU 10
C LU [illegible] (MIMC[illegible]AT10)
      DO 37 L=1,50
        DO 36 I=1,IR
          WRITE(10,100)(H(I,J,L),J=1,IC)
   36   CONTINUE
   37 CONTINUE
      RETURN
  100 FORMAT(2X,3(2X,F10.6))
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C DATA                                                  C
C  MDAT.FOR   process simulator                         C
C  Inputs consist of                                    C
C   - history of past inputs                            C
C   - transfer functions and time delays                C
C  Calculates new process output                        C
C                                                       C
C  T(n) - vector of sampling rates                      C
C       T(1) - control sampling rate                    C
C       T(2) - .2T(1) records intersample behavior      C
C  GN(n,i,j,k) - matrix containing numerator ARMA       C
C       coefficents                                     C
C       n - corresponds to sampling rate                C
C       i - corresponds to row                          C
C       j - corresponds to column                       C
C       k - corresponds to SISO parameter               C
C  GD(n,i,j,k) - matrix containing denominator ARMA     C
C       coefficients                                    C
C       indices are same as above                       C
C  Y(i,l) - matrix of process outputs                   C
C       l - refers to 1 of 5 intersample times          C
C  U(j,m) - matrix of control inputs                    C
C       m - refers to 1 of 30 possible inputs           C
C         - where 1 is the present input                C
C  ITD(i,j) - matrix of time delays                     C
C  X(i,j,k) - matrix of SISO process outputs            C
C       k - refers to time of output                    C
C         - 1, present    2, one past  3, two past      C
C  XNOISE(i,j,k) - matrix of past SISO noise inputs     C
C  C(i,j,k) - coefficients of noise polynomial          C
C  TIME - counter used to keep track of elapsed time    C
C  YERR - integral output error                         C
C  UERR - integral control effort                       C
C  IR - number of rows in TF                            C
C  IC - number of columns in TF                         C
C  IO - order of ARMA model                             C
C  ISIM - ISIM determines transfer of control ISIM=1    C
C         for linear simulations                        C
C                                                       C
C   Variable ID for the Non Linear simulator            C
C   The nonlinear simulator is a blending tank made up  C
C   of a cylinder and a cone                            C
C   The radius of the tank is given by R=R1+S*H.        C
C   The tank has three streams denoted 1,2,3.           C
C   The measuring element is situated a distance D fromC
C   the exit of the blending tank, on stream 3.         C
C   The flow rate of stream 2 is controlled via a zero  C
C   order valve.                                        C
C   QS-vector of steady state flow rates                C
C   CS-vector of steady state concentrations            C
C   Q1,Q2,Q3 - flow rates of respective streams         C
C   C1,C2,C3 - concentrations of respective streams     C
C   R1 -radius of blending tank at exit                 C
```

```
C    S-slope of line relating height to radius                C
C    VS-steady state volume of blending tank                  C
C    HS-steady state height of tank                           C
C    D-distance of measuring element from exit of tank        C
C    UB-bulk velocity of fluid leaving tank                   C
C    RHO-density of fluid in tank, all steams are             C
C    assumed dilute                                           C
C    DELV-time derivative of tank volume                      C
C    DELC-time derivative of C3                               C
C    CONC-vector containg present and past process            C
C    outputs                                                  C
C    TDELAY-used to estimate the transport delay from         C
C    tank exit to measuring element                           C
C    HCONE - maximum height of cone section                   C
C    A-difference in HS-HCONE for HS .GT. HCONE               C
C     when HS .LT. HCONE  A=0.0                               C
C    VCONE=maximum volume of cone                             C
C    ACYL=cross sectional area of cylinder                    C
C    ISIM -ISIM=1 linear simulator,ISIM=2 nonlinear sim.C
C                                                             C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE DATA(IR,IC,IO,ITD,GN,GD,U,T,Y,IN,ITD2
     1       ,ISIM,YD)
      DIMENSION ITD(3,3), GN(2,3,3,2), GD(2,3,3,2), U(3,70),
     1       T(2), X(3,3,3), XNOISE(3,3,2), Y(3,5), ITD2(3,3)
     2       ,C(3,3,2), YOLD(3), YD(3), U1(3,3,20)
     3       , CONC(20), CS(3),QS(3)
      REAL NOISE
      REAL *8 DSEED
C explicit functions
      F1(E,GG)=PI*(R1**2+S*R1*E+S**2*E**2/3.0)*E-GG
      F2(E)=PI*(R1**2+2.0*S*R1*E+S**2*E**2)
C initialization
      DATA C/0.0,17*0.0/, YERR/0.0/, UERR/0.0/,
     1       X/27*0.0/, XNOISE/18*0.0/, TIME/0.0/,
     2       INDEX/0/, NOISE/0.0/, DSEED/9999./, YOLD/3*0.0/
     3       , U1/180*0.0/, UB/0.0/
     4       , VS/0.0/, HS/0.0/, RHO/0.0/, R1/0.0/, S/0.0/
     5       , TDELAY/0.0/, D/0.0/, DELV/0.0/, DELC/0.0/
     6       , CONC/20*0.0/, CS/3*0.0/, QS/3*0.0/, PI/3.1415/
     7       , ACYL/0.0/, VCONE/0.0/, A/0.0/
C
      IF(IN-1)5,5,10
C
    5 CONTINUE
      IF(ISIM .EQ. 2) CALL NLSS(CS,QS,RHO,R1,S,VS,HS,D
     1       ,EQL,FRIC,HCONE)
      TIME=0.0
      INDEX=0
      YERR=0.0
      UERR=0.0
      NOISE=0.0
      DSEED=9999.
```

```
      A=0.0
      IF(HS .GT. HCONE) A=HS-HCONE
      ACYL=PI*(R1+S*HCONE)**2
      VCONE=PI*(R1*R1+S*R1*HCONE+S*S*HCONE**2/3.0)*HCONE
      V=VS
      H=HS
      C1=CS(1)
      C2=CS(2)
      C3=CS(3)
      Q1=QS(1)
      Q2=QS(2)
      Q3=QS(3)
      DO 8 I=1, IR
        DO 7 J=1, IC
          DO 6 L=1,IO
            XNOISE(I,J,L)=0.0
    6     CONTINUE
          DO 7 L=1, 3
          X(I,J,L)=0.0
    7   CONTINUE
        YOLD(I)=0.0
    8 CONTINUE
      DO 9 I=1, 20
        CALL GGNML(DSEED,1,NOISE)
        CONC(I)=CS(3)+NOISE*C(IR,IC,IO)
    9 CONTINUE
C
   10 CONTINUE
C jump to the nonlinear simulator if ISIM=2
      IF(ISIM .EQ.2) GOTO 50
C ...........................................
C . linear simulator                          .
C ...........................................
C update the intersample control vector U1
      DO 14 I=1,IR
        DO 13 J=1, IC
          DO 12 K=1,5
            DO 11 L=1,3
              U1(I,J,(L)*5+K)=U(J,ITD(I,J)+L-1)
   11       CONTINUE
   12     CONTINUE
          DO 13 K=1,6
          U1(I,J,K)=U1(I,J,ITD2(I,J)+10+K)
   13   CONTINUE
   14 CONTINUE
C
C begin calculations
C
      DO 30 I=1, IR
        DO 25 K=1, 5
          Y(I,K)=0.0
          DO 20 J=1, IC
            X(I,J,1)=0.0
            CALL GGNML(DSEED,1,NOISE)
```

```
            XNOISE(I,J,1)=NOISE
            DO 15 L=1, IO
              X(I,J,1)=X(I,J,1)+GN(2,I,J,L)*
     1        U1(I,J,5+L-K)
     2        +GD(2,I,J,L)*X(I,J,L+1)+C(I,J,L)*XNOISE(I,J,L)
   15       CONTINUE
            X(I,J,3)=X(I,J,2)
            X(I,J,2)=X(I,J,1)
            XNOISE(I,J,2)=XNOISE(I,J,1)
            Y(I,K)=Y(I,K)+X(I,J,1)
   20     CONTINUE
C Note if a disturabance were to be simulated, the magnitude
C would be added to the right hand side of the next line
C and the program recompiled
          Y(I,K)=Y(I,K)
   25   CONTINUE
   30   CONTINUE
C
C compute control effort and output error
      DO 31 J=1, IC
        UERR=UERR+ABS(U(J,1)-U(J,2))*T(1)
   31 CONTINUE
      DO 33 I=1, IR
        DO 32 J=1, 5
          YERR=YERR+ABS(Y(I,J)-YD(I))*T(2)
   32   CONTINUE
   33 CONTINUE
C output - logical unit 9
      WRITE(9,100)(TIME,(YOLD(I),I=1,IR),(U(J,1),J=1, IC))
      DO 35 K=1, 4
        TIME=TIME + T(2)
        WRITE(9,100)(TIME,(Y(I,K),I=1,IR),(U(J,1),J=1,IC))
   35 CONTINUE
      TIME=TIME+T(2)
      WRITE(9,100)(TIME,(Y(I,5),I=1,IR),(U(J,1),J=1,IC))
     1   , YERR, UERR
      DO 40 I=1,IR
        YOLD(I)=Y(I,5)
   40 CONTINUE
      RETURN
C
   50 CONTINUE
C ..........................................
C . nonlinear simulator                    .
C ..........................................
C start by adding the control signal pertubation U(1,1)
C to Q2
      Q2=QS(2)+U(1,1)*.2*QS(2)
      TDELAY=0.0
C begin loop to calculate C3 and Q3 over the intersample
      DO 60 I=1,5
        CALL GGNML(DSEED,1,NOISE)
        DELV=Q1+Q2-Q3
        V=T(2)*60.*DELV+V
```

```
      DELC=(C1*Q1+C2*Q2-C3*Q3-C3*DELV)/V
      C3=T(2)*60.*DELC+C3
      CONC(6-I)=C3+NOISE*C(IR,IC,IO)
      IF(V .GT. VCONE) H=HCONE+(V-VCONE)/ACYL
      IF(V .GT. VCONE) GOTO 56
C use Newtons method,to solve the cubic relationship
C between V and H
      DO 55 J=1,5
        TEMP1=F1(H,V)
        TEMP2=F2(H)
        H=H-TEMP1/TEMP2
 55   CONTINUE
 56   CONTINUE
      UB=SQRT((2.0*9.81*H*R1)/(R1+2.0*FRIC*(EQL+D)))
      Q3=PI*R1**2*UB
 60 CONTINUE
C store the delayed pertubation C3's in YOUT
    TDELAY=D/(UB*T(1)*60.)
    IF(TDELAY .GT. 15) WRITE(6,102)
    DO 65 I=1,5
      Y(1,I)=(CONC(6-I+IFIX(TDELAY))-CS(3))
 65 CONTINUE
C update the vector of past histories
    DO 66 I=1,15
      CONC(21-I)=CONC(16-I)
 66 CONTINUE
C
C compute control effort and output error
    DO 67 J=1, IC
      UERR=UERR+ABS(U(J,1)-U(J,2))*T(1)
 67 CONTINUE
    DO 69 I=1, IR
      DO 68 J=1, 5
        YERR=YERR+ABS(Y(I,J)-YD(I))*T(2)
 68   CONTINUE
 69 CONTINUE
C output - logical unit 9
    WRITE(9,100)(TIME,(YOLD(I),I=1,IR),(U(J,1),J=1,IC))
    DO 71 K=1, 4
      TIME=TIME + T(2)
      WRITE(9,100)(TIME,(Y(I,K),I=1,IR),(U(J,1),J=1,IC))
 71 CONTINUE
    TIME=TIME+T(2)
    WRITE(9,100)(TIME,(Y(I,5),I=1,IR),(U(J,1),J=1,IC))
   1      ,YERR,UERR,H
    DO 72 I=1,IR
      YOLD(I)=Y(I,5)
 72 CONTINUE
    RETURN
C format lines
100 FORMAT(2X,F6.2,2X,8(F8.2,2X))
102 FORMAT(' trouble , the variable time delay exceeds',
   1      ' the dimensions of CONC ')
    END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C IMC1                                                 C
C  Subroutine MIMC1.FOR  February 2 1984               C
C   - program builds the LAMDA, GAMMA, BETA, EPSILOM   C
C     and PSI matrices required by the control law     C
C                                                      C
C   Reference: Garcia and Morari; ' Internal Model     C
C      Control. 3. Multivariable Control Law           C
C      Computation and Tuning Guidelines ', Submitted  C
C      I &EC Proc. Des. and Dev. (April 1983)          C
C                                                      C
C   Inputs - N, M, P, TO, IR, IC, H and the optimal    C
C          time delay compensator                      C
C          - must read input and output weighting from C
C            logical unit 7(MIMC.DAT7)                 C
C            The beta and gamma weighting are written  C
C            so that all the input weightings at time K C
C            and all the output weightings at time K   C
C            occur on the same line. For each new time C
C            interval a new line of input and output   C
C            weightings are recorded. Further infor-   C
C            mation can be found from the read statementC
C                                                      C
C   Variable ID                                        C
C   H(i,j,k) - impulse response matrix                 C
C            - i,j identifies position of series in    C
C              transfer function matrix                C
C            - k denotes term in impulse response      C
C              series                                  C
C   GP(j) - optimal time delay compenstaor             C
C   GAMMA - matrix of output costing                   C
C   LAMDA - matrix of future impulse response coeff.   C
C   LAMDAT - matrix equivalent to transpose of LAMDA   C
C   PSI   - matrix of past impulse response coeff.     C
C   BETA  - matrix of input costing                    C
C   RDO - matrix containing the inverted return        C
C       difference operator                            C
C   CONST - matrix equal to the product of RDO and     C
C         LAMDAT                                       C
C   NC1 -  matrix= sum of controller numerator         C
C        coefficents                                   C
C   H1 - matrix= sum of impulse response series        C
C       coefficients                                   C
C   DC1 - matrix= sum of controller denominator        C
C        coefficents                                   C
C   OFF - matrix offset compensator                    C
C                                                      C
C   N - number of terms in impulse response series     C
C   M - input suppression parameter                    C
C   P - optimization horizon                           C
C   IC - number of columns in transfer function matrix C
C   IR - number of rows in transfer function matrix    C
C   TO - measure of imbalance in process transfer      C
C       function                                       C
```

```
C   IOFF - determines whether the offset compensator   C
C          is operative.  If IOFF=1, compensator is     C
C          operative. If IOFF=2, compensator is not     C
C          operative.                                   C
C                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE IMC1(IR,IC,N,M,P,TO,H,GP,CONST,GAMMA,
     1      PSI, OFF,IOFF)
      DIMENSION GP(3), GAMMA(180),BETA(180), LAMDA(90,90),
     1      PSI(60,120), LAMDAT(90,90), WKARE1(90,90)
     2      , WKARE2(91), H(3,3,70), CONST(90,90)
     3      , OFF(3,3), NC1(3,3), DC1(3,3), WKARE3(3,3),
     4      WKARE4(3,3),H1(3,3),WKARE5(3,3),WKARE6(60,120)
     5      , RDO(90,90)
      REAL PSI, LAMDA, LAMDAT, NC1, NCMAX
      INTEGER P, TO, PPTO, PPTOP1, PIR, MIC, GP
C
C initialization
      DATA BETA/180*0.0/, LAMDA/8100*0.0/,RDO/8100*0.0/
     1      , WKARE1/8100*0.0/,LAMDAT/8100*0.0/, NCMAX/0.0/
     2      , WKARE3/9*0.0/, WKARE4/9*0.0/, NC1/9*0.0/,
     3      DC1/9*0.0/, H1/9*0.0/, WKARE5/9*0.0/,
     4      WKARE6/7200*0.0/
C read in values for BETA and GAMMA from logical unit 7
      PPTO=P+TO
      IF(IC-2)1,3,5
    1 CONTINUE
      DO 2 I=1, PPTO
        CALL FREAD(7,'(2R):',BETA(I),GAMMA(I))
    2 CONTINUE
      GOTO 6
    3 CONTINUE
      DO 4 I=1, PPTO
        CALL FREAD(7,'(4R):',BETA((I-1)*IC+1),
     1        BETA((I-1)*IC+2)
     2        ,GAMMA((I-1)*IR+1),GAMMA((I-1)*IR+2))
    4 CONTINUE
      GOTO 6
    5 CONTINUE
      DO 6 I=1, PPTO
        CALL FREAD(7,'(6R):',BETA((I-1)*IC+1),
     1      BETA((I-1)*IC+2),BETA((I-1)*IC+3)
     2      ,GAMMA((I-1)*IR+1),GAMMA((I-1)*IR+2),
     3      ,GAMMA((I-1)*IR+3))
    6 CONTINUE
C build the LAMDA matrix
      NPTO=N+TO
      NPTOP1=N+TO+1
      MIC=M*IC
      PIR=PPTO*IR
      NPTOM1=N+TO-1
C
      DO 10 K=1, PPTO
```

```
      DO 9 L=1, K
        IM=(K-1)*IR
        IN=(L-1)*IC
        DO 8 J=1, IC
          DO 7 I=1, IR
            IMI=IM+I
            INJ=IN+J
            IF(INJ .GT. MIC) INJ=MIC
            LAMDA(IMI,INJ)=LAMDA(IMI,INJ)+
     1      H(I,J,GP(I)+K+1-TO-L)*GAMMA((K-1)*IR+I)
    7     CONTINUE
    8   CONTINUE
    9 CONTINUE
   10 CONTINUE
C
      IF(P-N)25,25,15
C
   15 CONTINUE
C
      DO 24 K=NPTOP1, PPTO
        DO 23 L=1, NPTO
          IL=K-NPTO+L
          IM=(K-1)*IR
          IN=(IL-1)*IC
          DO 22 J=1, IC
            DO 21 I=1, IR
              IMI=IM+I
              INJ=IN+J
              IF(INJ .GT. MIC) INJ=MIC
              LAMDA(IMI,INJ)=LAMDA(IMI,INJ)+H(I,J,GP(I)-TO
     1        +NPTOP1-L)*GAMMA((K-1)*IR+I)
   21       CONTINUE
   22     CONTINUE
   23   CONTINUE
   24 CONTINUE
C
C build the PSI matrix
C
   25 CONTINUE
      DO 30 K=1,PPTO
        DO 29 I=1,IR
          IEND=NPTO-K
          DO 28 L=1, IEND
            DO 27 J=1,IC
              PSI((K-1)*IR+I,(L-1)*IC+J)=H(I,J,GP(I)+L+K)*
     1           GAMMA((K-1)*IR+I)
   27       CONTINUE
   28     CONTINUE
   29   CONTINUE
   30 CONTINUE
      DO 31 I=1,6
   31 CONTINUE
C build the return difference operator
      CALL TRANS(LAMDA,90,90,PIR, MIC, LAMDAT, 90,90)
```

```
      CALL MULT(LAMDAT,90,90,MIC,PIR,LAMDA,90,90,MIC,WKARE1,
     1      90,90)
      DO 40 I=1, MIC
        DO 39 J=1, M
          DO 38 JJ=1, IC
            JJJ=(J-1)*IC+JJ
            WKARE1(I,JJJ)=WKARE1(I,JJJ)+BETA(JJJ)*BETA(JJJ)
   38     CONTINUE
   39   CONTINUE
   40 CONTINUE
      CALL LINV1F(WKARE1,MIC,90,RDO,0,WKARE2,IER)
C
C compute RDO*LAMDAT
      CALL MULT(RDO,90,90,MIC,MIC,LAMDAT,90,90,PIR,CONST,90,
     1      90)
C if offset compensation is not required RETURN
C if IOFF=1 offset, if IOFF=2 no offset
      IF(IOFF.GT.1)RETURN
C compute the NC1
      DO 45 K=1, PPTO
        DO 44 I=1, IR
          DO 43 J=1, IR
            NC1(I,J)=NC1(I,J)+CONST(I,(K-1)*IC+J)
   43     CONTINUE
   44   CONTINUE
   45 CONTINUE
C compute H1
      DO 50 K=1, NPTO
        DO 49 I=1, IR
          DO 48 J=1, IC
            H1(I,J)=H1(I,J)+H(I,J,K+GP(I)-TO)
   48     CONTINUE
   49   CONTINUE
   50 CONTINUE
C compute DC1
      CALL MULT(CONST,90,90,MIC,PIR,PSI,60,120,NPTOM1,WKARE6
     1      ,60,120)
      DO 55 K=1, NPTOM1
        DO 54 I=1, IR
          DO 53 J=1, IR
            DC1(I,J)=DC1(I,J)+WKARE6(I,(K-1)*IC+J)
   53     CONTINUE
   54   CONTINUE
   55 CONTINUE
      DO 57 I=1,IR
        DC1(I,I)=DC1(I,I)+1.0
   57 CONTINUE
C compute DC1 inverse  store inverse in WKARE3
      CALL LINV1F(DC1,IR,3,WKARE3,0,WKARE2,IER)
C multiply NC1 by WKARE3 store result in WKARE4
      CALL MULT(WKARE3,3,3,IR,IR,NC1,3,3,IR,WKARE4,3,3)
C multiply WKARE4 by H1 store the result in WKARE5
      CALL MULT(H1,3,3,IR,IR,WKARE4,3,3,IR,WKARE5,3,3)
C compute the inverse of NC1*WKARE3*H1 and store in OFF
```

```
      CALL LINV1F(WKARE5,IR,3,OFF,0,WKARE2,IER)
  100 FORMAT(6F10.4)
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C SUBROUTINE MULT                                          C
C                                                          C
C MULTIPLIES TWO MATRICES TOGETHER                         C
C                                                          C
C EX. Y = A*B                                              C
C     DIMENSION OF A IN THE CALLING PROGRAM IS IA,IB       C
C     DIMENSION OF B IN THE CALLING PROGRAM IS IC,ID       C
C     DIMENSION OF Y IN THE CALLING PROGRAM IS IE,IF       C
C     THE ACTUAL NUMBER OF ROWS AND COLUMNS IN USE         C
C       IN MATRIX A ARE N,M                                C
C     THE ACTUAL NUMBER OF COLUMNS IN B ARE L              C
C                                                          C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE MULT(A,IA,IB,N,M,B,IC,ID,L,Y,IE,IF)
      DIMENSION A(IA,IB), B(IC,ID), Y(IE,IF)

C INITIALIZATION
      DO 5 I = 1, IE
        DO 5 J = 1, IF
          Y(I,J) = 0.0
    5 CONTINUE

C MULTIPLY THE MATRICES
      DO 10 I = 1, N
        DO 10 J = 1, L
          DO 10 K = 1, M
            Y(I,J) = Y(I,J) + A(I,K)*B(K,J)
   10 CONTINUE

      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                          C
C SUBROUTINE TRANS                                         C
C                                                          C
C TRANSPOSES A MATRIX                                      C
C                                                          C
C EX. YT(I,J) = Y(J,I)                                     C
C  WHERE:                                                  C
C    DIMENSION OF Y IN CALLING PROGRAM IS IA,IB            C
C    DIMENSION OF YT IN CALLING PROGRAM IS IC,ID           C
C    THE ROWS AND COLUMNS IN USE IN Y ARE N,M              C
C                                                          C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE TRANS(Y, IA, IB, N, M, YT, IC, ID)
      DIMENSION Y(IA,IB), YT(IC,ID)

C INITIALIZATION
      DO 10 I = 1, IC
```

```
        DO 10 J = 1, ID
          YT(I,J) = 0.0
   10 CONTINUE

C FIND THE TRANSPOSE OF Y
      DO 15 I = 1, N
        DO 15 J = 1, M
          YT(J,I) = Y(I,J)
   15 CONTINUE

      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C IMC2                                                 C
C   Subroutine MIMC2.FOR February 12 1984              C
C    - programs computes the IMC control signal u(k)   C
C     on the basis of y(k) and information calculated C
C     from MIMC1.FOR                                   C
C                                                      C
C   Reference: Garcia and Morari; ' Internal Model     C
C      Control. 3. Multivariable Control Law           C
C      Computation and Tuning Guidelines ', Submitted  C
C      I &EC Proc. Des. and Dev. (April 1983)          C
C                                                      C
C   Variable ID                                        C
C    YM - vector containing model prediction of ym(k)  C
C    Y - vector containing true y(k)                   C
C    E - vector containing difference between setpoint C
C       and feedback signal y(k)-ym(k)                 C
C    DIFF - difference of E-PSIU                       C
C    UNEW - prediction of future control inputs        C
C    U - matrix of control history                     C
C    CONST - product of RDO*LAMDAT                     C
C    GAMMA - vector of input weightings                C
C    PSI - matrix of historical response coefficents   C
C    PSIU-  product of PSI*U                           C
C    H - matrix containing impulse response coefficentsC
C                                                      C
C    N - order of truncation                           C
C    M - input suppression parameter                   C
C    P - optimization horizon                          C
C    IR - number of rows in TF matrix                  C
C    IC - number of columns in TF matrix               C
C    TO - measure of imbalance in TF matrix            C
C                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE IMC2(IR,IC,N,M,P,TO,CONST,GAMMA,PSI,Y,U,H
     1       , OFF, FILTER,YD)
      DIMENSION CONST(90,90), GAMMA(180), PSI(60,120)
     1       , Y(3,5), U(3,70), H(3,3,70), E(90,2)
     2       , YM(3), PSIU(90), YD(3), DIFF(90,1), UNEW(90,1)
     3       , FILTER(3), OFF(3,3), WK1(3,1), WK2(3,1)
      REAL PSI, PSIU
      INTEGER P, TO, PPTO, PIR
C initialization
      DATA E/180*0.0/, YM/3*0.0/
     1       , WK1/3*0.0/, WK2/3*0.0/
      NPTO=N+TO
      PPTO=P+TO
      MIC=M*IC
      PIR=PPTO*IR
      NIR=NPTO*IR
      DO 2 K=1, PIR
        PSIU(K)=0.0
        E(K,1)=0.0
        DIFF(K,1)=0.0
```

```
      UNEW(K,1)=0.0
    2 CONTINUE
C calculate the predicted output ym(k)
      DO 5 I=1, IR
        YM(I)=0.0
        DO 4 J=1, IC
          DO 3 K=1, NPTO
            YM(I)=YM(I)+H(I,J,K)*U(J,K)
    3     CONTINUE
    4   CONTINUE
    5 CONTINUE
C calculate the offset compensated feedback signal and
C setpoint
      DO 6 I=1, IR
        WK1(I,1)=YD(I)-(Y(I,5)-YM(I))
    6 CONTINUE
      CALL MULT(OFF,3,3,IR,IR,WK1,3,1,1,WK2,3,1)
C calculate E(k) over the horizon P
      DO 8 K=1, PPTO
        DO 7 I=1, IR
          IK=(K-1)*IR+I
          E(IK,1)=FILTER(I)*E(IK,2)+(1-FILTER(I))*WK2(I,1)*
     1          GAMMA((K-1)*IR+I)
          E(IK,2)=E(IK,1)
    7   CONTINUE
    8 CONTINUE
C compute the product of PSI*U store this result in PSIU
      DO 12 K=1, PIR
        PSIU(K)=0.0
        DO 11 L=1,NPTO
          DO 10 J=1, IC
            IL=(L-1)*IC+J
            PSIU(K)=PSIU(K)+PSI(K,IL)*U(J,L)
   10     CONTINUE
   11   CONTINUE
   12 CONTINUE
C compute the difference between E and PSIU
      DO 14 I=1, PIR
        DIFF(I,1)=E(I,1)-PSIU(I)
   14 CONTINUE
C compute the horizon of new control actions
      CALL MULT(CONST,90,90,MIC,PIR,DIFF,90,1,1,UNEW,90,1)
C implement only the first IC terms in UNEW
      DO 16 K=1, 69
        DO 15 J=1, IC
          U(J,71-K)=U(J,70-K)
   15   CONTINUE
   16 CONTINUE
      DO 17 J=1, IC
        U(J,1)=UNEW(J,1)
   17 CONTINUE
C return
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C INIT                                                C
C  Subroutine MINIT.FOR February 11 1984              C
C  - program initializes the various control laws     C
C                                                     C
C  Inputs - enter the control law case number.ie ICASE C
C                                                     C
C  Variable ID                                        C
C  IMC                                                C
C   P - optimization horizon                          C
C   M - input supppression parameter                  C
C   N - number of terms in truncation model           C
C   TO - measure of imbalance in process transfer     C
C        function                                     C
C   GP - diagonal elements of optimal time delay      C
C       compensator                                   C
C   IC - number of inputs in transfer function matrix C
C   FILTER - vector containing IMC filter parameters  C
C   IOFF - determines whether or not offset           C
C        is desired                                   C
C                                                     C
C  ORC                                                C
C   IOR - ORC time delay estimates                    C
C   CK  - matrix of controller gains                  C
C   CI  - matric of controller reset values           C
C   GI - multivariable steady state decoupler         C
C                                                     C
C  MAC-NMP                                            C
C  FILTER -alpha weighting in setpoint trajectory     C
C  GP- optimal time delay compensator                 C
C  N - number of terms in impulse response series     C
C                                                     C
C                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE INIT(ICASE, IC, P, M, N, TO, GP, FILTER
     1      , IOFF,IOR,CK,CI,IR,GI,YD)
      DIMENSION GP(3), FILTER(3),IOR(3,3),CK(3,3),CI(3,3)
     1      ,GI(3,3), YD(3)
      INTEGER GP, P, TO
      REAL GI
C enter the desired setpoint
      WRITE(6,112)
      DO 5 I=1,IR
        CALL FREAD(5,'(1R):',YD(I))
        WRITE(11,104)YD(I)
    5 CONTINUE
C transfer control via ICASE
      GOTO(10,10,10,21,21,21,25), ICASE
   10 CONTINUE
C initialize IMC control law
C enter P, M, N, TO
      WRITE(6,100)
      CALL FREAD(5, '(4I):', P, M, N, TO)
      WRITE(11,101)P,M,N,TO
```

```
C
C enter the optimal time compensator diagonal elements
      WRITE(6,102)
      DO 15 I=1, IC
        CALL FREAD(5,'(I):',GP(I))
        WRITE(11,101)GP(I)
   15 CONTINUE
C enter the filter coefficent values
      WRITE(6,103)
      DO 20 I=1, IC
        CALL FREAD(5,'(R):',FILTER(I))
        WRITE(11,104)FILTER(I)
   20 CONTINUE
C ask for offset compensation
      WRITE(6,105)
      CALL FREAD(5,'(I):',IOFF)
      WRITE(11,101)IOFF
C
      RETURN
   21 CONTINUE
C initialize the ORC tuning parameters
C enter the time delay compensator estimates for each
C element
      WRITE(6,106)
      DO 22 I=1, IR
        DO 22 J=1, IC
          WRITE(6,107)I,J
          CALL FREAD(5,'(I,3R):',IOR(I,J),CK(I,J),CI(I,J),
     1    GI(I,J))
          WRITE(11,108)IOR(I,J),CK(I,J),CI(I,J),GI(I,J)
   22 CONTINUE
      RETURN
   25 CONTINUE
C initialize MAC control law
      WRITE(6,109)
      CALL FREAD(5,'(2I,2R):',N,GP(1),FILTER(1),FILTER(2))
      WRITE(11,110)N,GP(1),FILTER(1),FILTER(2)
      RETURN
C format lines
  100 FORMAT(' enter IMC tuning parameters',/,
     1       ' P- optimization horizon',/,4X,' where P<or=N'
     1       ,' and (P+T0)*outputs<90',/,
     2       ' M- input suppression  parameter',
     2       ' M*inputs<or=P<90',/,
     3       ' N- number of terms in impulse response model',
     3       /,3X,' where N+Tdelay<70 and (N+T0)*inputs<120'
     4       ,/,' TO- measure of imbalance in TF',/,
     5       ' enter these four integer values on the',
     6       ' next line')
  101 FORMAT(2X,4(I3,2X))
  102 FORMAT(' enter the optimal time delay diagonal',
     1       ' elements, one per line ')
  103 FORMAT(' enter filter coefficients, one per line ')
  104 FORMAT(2X,F10.4)
```

```
105 FORMAT(' do you want offset compensation yes(1),'
   1       ,' no(2)')
106 FORMAT(' ORC control law ',/,
   1       ' enter the time delay estimate, controller',
   2     ' gain, controller reset value and multivariable',
   3       ' S.S. decoupler')
107 FORMAT(' element ',2X,I4,2X,I4)
108 FORMAT(2X,I4,2X,4(F10.4,2X))
109 FORMAT(' enter the number of terms in the impulse',
   1       ' response series, N<=50',/,
   1       ' the time delay estimate GP(1)',/,
   2       ' the alpha weighting, FILTER(1), 0<=FILTER<1'
   3       ,/,' and the input weighting R(FILTER(2) ',/,
   4       ,' enter these four values on the next line ')
110 FORMAT(2X,2(I4,2X),2(F10.4,2X))
112 FORMAT(' enter the setpoints, one per line ')
    END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C MACCL                                                 C
C  Subroutine MMACCL.FOR, April 28 1984                 C
C  MAC control law, used in conjunction with MACNMP     C
C  This control law is developed using an optimization C
C   horizon P of 1. Also subroutine MACNMP must be      C
C   called first to generate HK                         C
C                                                       C
C   The MAC paper is Rouhani and Mehra 1982, AutomaticaC
C   Vol. 18, No. 4, 401-414(1982)                       C
C                                                       C
C  Variable ID                                          C
C  U - matrix of past control inputs                    C
C  HK- matrix of model coefficents                      C
C  N number of terms in HK series                       C
C                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE MACCL(N,HK,U,Y,FILTER,GP,YD)
      DIMENSION HK(50),U(3,70),FILTER(3),GP(3),Y(3,5)
     1      ,YD(3)
      NM1=N-1
      UNEW=0.0
      DO 10 I=1,NM1
        UNEW=UNEW+(HK(I)-HK(I+1))*U(1,I)/HK(1)
   10 CONTINUE
      UNEW=UNEW+HK(N)*U(1,N)/HK(1)-(1.0-FILTER(1))*Y(1,5)
     1      /HK(1)+(1.0-FILTER(1))*YD(1)/HK(1)
C update the input vector with unew
      DO 15 I=1,69
        U(1,71-I)=U(1,70-I)
   15 CONTINUE
      U(1,1)=UNEW
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C MACNMP                                                C
C   Subroutine MMACNMP.for  April 28 1984               C
C     - subroutine solves the matrix Ricatti equations C
C       to find the optimum gain K in the MAC control  C
C       law when the process is NMP phase               C
C                                                       C
C   The MAC paper is Rouhani and Mehra 1982, AutomaticaC
C   The matrix Ricatti equations are taken from         C
C        Franklin and Powell, Digital Control of        C
C        Dynamic Systems, Addison-Wesley 1980 p254      C
C                                                       C
C   Inputs - N, FILTER, H and GP                        C
C   Outputs - K the optimum gain for the controller     C
C                                                       C
C   Variable ID                                         C
C   H - a  matrix (must be 1,1,N) containing impulse    C
C       response series coefficents                     C
C   HA - the minimum phase approximation of the NMP     C
C        transfer function.                             C
C   GP  - optimal time delay compensator (must be 1 by C
C       1 for this program to work)                     C
C   FILTER - vector containing alpha weighting (must beC
C       of dimension 1 for this program to work)        C
C   GAMMA - matrix of dimension (N,1) contains a 1 and C
C       the rest are zeros                              C
C        GAMMA corresponds to b in the MAC formulation C
C   Q  - matrix of dimension N,N. Contains the norm of C
C       the vector h0, h1-alpha*h0, h2-alpha*h1,...    C
C   PHI - matrix of dimension N,N. Contains all zeros  C
C       except for the lower diagonal of ones.          C
C       Corresponds to U in the MAC formulation         C
C   M - used to solve the matrix Ricatti equations      C
C       dimension of N,N                                C
C   S - used to solve the matrix Ricatti equations      C
C       dimension N,N                                   C
C   TEMP - is a temporary workspace of dimension N,N    C
C                                                       C
C  N is the number of terms in the impulse response     C
C    in the impulse response series                     C
C                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE MACNMP(N,FILTER,H,GP,HA)
      DIMENSION Q(50,50), PHI(50,50), GAMMA(50), K(50)
     1      , S(50,50), M(50,50), H(3,3,70), FILTER(3),
     2      GP(3), TEMP(50,50), HA(50), PHIT(50,50)
      REAL M, K
      INTEGER GP
C initialization
      DATA Q/2500*0.0/, PHI/2500*0.0/, GAMMA/50*0.0/,
     1      S/2500*0.0/, M/2500*0.0/, TEMP/2500*0.0/
     2      ,K/50*0.0/, SUMK/0.0/, SUMH/0.0/, PHIT/2500*0.0/
     3      ,R/0.000/
```

```
C begin by building the Q matrix, use G as temporary storage
      R=FILTER(2)
      NM1=N-1
      NM2=N-2
      NP1=N+1
      GAMMA(1)=H(1,1,GP(1)+1)
      DO 10 I=2,N
        GAMMA(I)=H(1,1,GP(1)+I)-FILTER(1)*H(1,1,GP(1)+I-1)
   10 CONTINUE
      GAMMA(NP1)=-FILTER(1)*H(1,1,GP(1)+N)
      DO 15 I=1,NP1
        DO 14 J=1,NP1
          Q(I,J)=GAMMA(I)*GAMMA(J)
          S(I,J)=Q(I,J)
   14   CONTINUE
   15 CONTINUE
C build the PHI matrix
      DO 17 I=2,NP1
        PHI(I,I-1)=1.0
        PHIT(I-1,I)=1.0
   17 CONTINUE
C begin the recursive calculations
      DO 30 L=1, 30
        DO 20 I=1,N
          TEMP(1,I)=-S(1,I+1)/(S(1,1)+R)
          K(I)=-S(1,I+1)/(S(1,1)+R)
   20   CONTINUE
        TEMP(1,NP1)=0.0
        K(NP1)=0.0
        DO 25 I=2,NP1
          DO 23 J=1,NP1
            TEMP(I,J)=PHI(I,J)
   23     CONTINUE
   25   CONTINUE
        CALL MULT(S,50,50,NP1,NP1,TEMP,50,50,NP1,M,50,50)
        CALL MULT(PHIT,50,50,NP1,NP1,M,50,50,NP1,S,50,50)
        DO 29 I=1,NP1
          DO 27 J=1,NP1
            S(I,J)=S(I,J)+Q(I,J)
   27     CONTINUE
   29   CONTINUE
   30 CONTINUE
C calculate the optimum impulse response series
C from K
      DO 35 I=1,N
        SUMH=SUMH+H(1,1,GP(1)+I)
        SUMK=SUMK+K(I)
   35 CONTINUE
      HA(1)=-(1.0-FILTER(1))*SUMH/(SUMK-1.0)
      HA(N)=(1.0-FILTER(1))*H(1,1,GP(1)+N)+K(N)*HA(1)
      DO 40 I=1,NM2
        HA(N-I)=K(N-I)*HA(N-I)+(1.0-FILTER(1))*
     1        H(1,1,GP(1)+N-I)+HA(N-I+1)
   40 CONTINUE
```

```
      RETURN
100 FORMAT(2X,7(F8.4,2X))
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C NLSS                                                C
C   Subroutine MNLSS.FOR,  May 10 1984                C
C   - subroutine calculates the steady state          C
C   operating conditions for the blending tank        C
C                                                     C
C   Variable ID                                       C
C   CS-vector containing the concentrations of solute C
C   in each stream                                    C
C   QS-vector containing the steady state flow rates  C
C   in each stream                                    C
C   VS-steady state volume of solution in tank        C
C   HS-steady state height of solution in tank        C
C   R1-radius of bottom outlet from blending tank     C
C   FRIC-friction factor                              C
C   EQL-equivalent lenth of friction loss due to elbow C
C   S-slope of line relating height to tank radius    C
C   Radius=R1+S*H where  H is measured from the bottom C
C   RHO-density of fluid, it is assumed all solutions C
C      are dilute so stream densities are equal       C
C   D-distance of measuring element from outlet of    C
C      tank                                           C
C   HCONE-maximum height of cone shape                C
C   A-for HS .LE. HCONE   A=0.0                       C
C     for HS .GT. HCONE   A=HS-HCONE                  C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE NLSS(CS,QS,RHO,R1,S,VS,HS,D,EQL,FRIC
     1       , HCONE)
      DIMENSION CS(3),QS(3)
      REWIND13
C read initial conditions set by the operator from logical
C unit 13 attached to MIMC.DAT13
      CALL FREAD(13,'(12R):',R1,S,RHO,QS(1),HS,CS(1),CS(2)
     1      ,D,EQL,FRIC,HCONE)
C begin calculations
      UB=SQRT((2.0*9.81*HS*R1)/(R1+2.0*FRIC*(EQL+D)))
      QS(3)=3.1416*R1**2*UB
      QS(2)=QS(3)-QS(1)
      CS(3)=(QS(1)*CS(1)+QS(2)*CS(2))/QS(3)
      A=0.0
      IF(HS .GT. HCONE) A=HS-HCONE
      VS=3.141*(R1**2+S*R1*(HS-A)+S*S*(HS-A)*(HS-A)/3.0)
     1      *(HS-A)
      VS=VS+3.141*(R1+S*HCONE)**2*A
C write the steady state values to LU 13
      WRITE(13,100)QS(3),CS(3),VS,UB
      RETURN
  100 FORMAT(2X,4(F10.4,2X))
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C  PULSE                                                    C
C  Subroutine: MPUL.FOR Jan 31 1984                         C
C  initializes the pulse vector U for a pulse response      C
C                                                           C
C  Variable ID                                              C
C  U initial value of input matrix                          C
C                                                           C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE PULSE(N, U)
      DIMENSION U(3,70)
C initialization
      IN=N-1
      IF(IN .LT. 1) IN=1
      DO 10 J=IN, N
        DO 9 K=1, 50
          U(J,K)=0.0
    9   CONTINUE
   10 CONTINUE
C
C input initial values
      WRITE(6,100) N
      CALL FREAD(5,'(R):', U(N,1))
      WRITE(11,101)U(N,1)
      RETURN
C format lines
  100 FORMAT(/' input the pulse size for input ',2X,I4)
  101 FORMAT(2X,F10.4)
C
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C ORC                                                C
C  Subroutine MORC.FOR, April 13 1984                C
C  This subroutine calculates the control action     C
C  using the multivariable control law of            C
C  Ogunnaike and Ray  AIChE Journal (Vol 25, No.6,   C
C  1043-1057, 1979)                                  C
C                                                    C
C  Variable ID                                       C
C  GN matrix of SISO ARMA numerator coefficents      C
C  GD matrix of SISO ARMA denominator coeeficents    C
C  Y - matrix of current output signals              C
C  U - matrix of past control actions                C
C  CK - matrix of PI controller gains                C
C  CI - matrix of PI controller reset values         C
C  YF - predicted delay free output                  C
C  YDE - predicted delayed output                    C
C  E - matix of present and past errors between the  C
C      setpoint - actual output - YF + YD            C
C  M - vector of calculated control inputs           C
C  MD - vector of decoupled control inputs           C
C  YD - vector of setpoints                          C
C                                                    C
C  IOR - time delay estimates for the transfer functionC
C      matrix                                        C
C  ITD - actual time delays for the transfer function C
C      elements                                      C
C  IC - number of columns in transfer function matrix C
C  IR - number of rows in transfer function matrix   C
C  IO - order of highest SISO transfer function      C
C                                                    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE ORC(IR,IC,IO,ITD,GN,GD,Y,U,CK,CI,IOR,
     1      GI,L,YD)
      DIMENSION ITD(3,3), GN(2,3,3,2), GD(2,3,3,2), U(3,70)
     1      , Y(3,5), CK(3,3), CI(3,3), IOR(3,3)
     2      , E(3,2), M(3), YD(3), GI(3,3), MD(3)
     3      , YDE(3,3,3), YF(3,3,3)
      REAL M, MD
C initialization
      DATA E/6*0.0/, M/3*0.0/
     1      , MD/3*0.0/, YDE/27*0.0/, YF/27*0.0/
C if l equals one set M,E,YDE,YF to zero
      IF(L .GT. 1) GOTO 3
      DO 2 J=1,IC
        E(J,2)=0.0
        M(J)=0.0
        DO 1 I=1, IR
          YDE(I,J,2)=0.0
          YDE(I,J,3)=0.0
          YF(I,J,2)=0.0
          YF(I,J,3)=0.0
    1   CONTINUE
```

```
    2 CONTINUE
    3 CONTINUE
C compute E(I,1)
      DO 6 I=1, IR
        E(I,1)=0.0
        DO 5 J=1, IC
          YDE(I,J,1)=0.0
          YF(I,J,1)=0.0
          DO 4 K=1, IO
            YF(I,J,1)=GN(1,I,J,K)*U(J,ITD(I,J)-IOR(I,J)+K)+
     1      GD(1,I,J,K)*YF(I,J,K+1)
     2      +YF(I,J,1)
            YDE(I,J,1)=GN(1,I,J,K)*U(J,ITD(I,J)+K)+
     1      GD(1,I,J,K)*YDE(I,J,K+1)
     2      +YDE(I,J,1)
    4     CONTINUE
          YF(I,J,3)=YF(I,J,2)
          YF(I,J,2)=YF(I,J,1)
          YDE(I,J,3)=YDE(I,J,2)
          YDE(I,J,2)=YDE(I,J,1)
          E(I,1)=YF(I,J,1)-YDE(I,J,1)+E(I,1)
    5   CONTINUE
        E(I,1)=YD(I)-Y(I,5)-E(I,1)
    6 CONTINUE
C compute the new control signal m(k)
      DO 10 J=1, IC
        DO 9 I=1, IR
          M(J)=CK(J,I)*E(I,1)-CK(J,I)*CI(J,I)*E(I,2)+M(J)
    9   CONTINUE
   10 CONTINUE
C compute the steady state decoupled control output
      DO 13 J=1,IC
        MD(J)=0.0
        DO 12 I=1,IR
          MD(J)=GI(J,I)*M(I)+MD(J)
   12   CONTINUE
   13 CONTINUE
C update the error matrix
      DO 14 I=1, IR
        E(I,2)=E(I,1)
   14 CONTINUE
C update the control vector
      DO 16 K=1, 69
        DO 15 J=1, IC
          U(J,71-K)=U(J,70-K)
   15   CONTINUE
   16 CONTINUE
      DO 17 J=1,IC
        U(J,1)=M(J)
   17 CONTINUE
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C  STEP                                                    C
C  Subroutine: MOL.FOR  Jan 31 1984                        C
C  initializes the step vector U for a step response       C
C                                                          C
C  Variable ID                                             C
C  U initial value of input matrix                         C
C                                                          C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE STEP(N, U)
      DIMENSION U(3,70)
C initialization
      IN=N-1
      IF(IN .LT. 1) IN=1
      DO 10 J=IN, N
        DO 9 K=1, 70
          U(J,K)=0.0
    9   CONTINUE
   10 CONTINUE
C
C input initial values
      WRITE(6,100) N
      CALL FREAD(5,'(R):', U(N,1))
      WRITE(11,101)U(N,1)
      RETURN
C format lines
  100 FORMAT(/' input the step size for input ',2X,I4)
  101 FORMAT(2X,F10.4)
C
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C ZTRANS                                                   C
C   Subroutine: MZTR.FOR  Jan 31 1984                      C
C    : digital difference equations (z domain) are derivedC
C      from the analog (Laplace domain models)             C
C                                                          C
C    Ref: Neuman and Baradello IEEE Transactions           C
C         Vol SMC 9 No 12      1979                        C
C                                                          C
C    Program Requirements:                                 C
C      This is an interactive program. The user is         C
C      required to enter                                   C
C            - the type of transfer function               C
C            - sampling rate                               C
C            - pertinent analog model parameters           C
C            - time delay                                  C
C                                                          C
C    Variable ID **                                        C
C                                                          C
C    ICON(I,J) - type of transfer function                 C
C                                                          C
C    analog model parameters:                              C
C    G(s) - transfer function                              C
C    K - steady state gain                                 C
C    r,p,q - model parameters                              C
C    zeta - damping factor                                 C
C    omega - frequency                                     C
C                                                          C
C    discrete model parameters                             C
C    GD(1,I,J,K) - parameters corresponding to model with C
C     the sampling rate equal to the control action rate  C
C    GN(1,I,J,K)- parameters corresponding to the control C
C       model                                              C
C    T(1) - sampling rate for the control model            C
C    GD(2,I,J,K) - parameters corresponding to the outputsC
C         from the systems where the sampling rate T(2)   C
C                      is .2T(1)                           C
C    GN(2,I,J,K) - parameters corresponding to the inputs C
C            where the sampling rate is T(2)               C
C    T(2) - .2T(1)                                         C
C                                                          C
C       NOTE: two sampling rates are used to record the   C
C             sample behavior                              C
C                                                          C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                          C
      SUBROUTINE ZTRANS(IR,IC,IO,ITD,GN,GD,T,ITD2,ISIM)
      DIMENSION GN(2,3,3,2), GD(2,3,3,2), T(2), ITD(3,3),
     1      ICON(3,3), ITD2(3,3)
      REAL K

C initialization                                           C
      DATA K/0.0/, OMEGA/0.0/
     1      , R/0.0/, P/0.0/, Q/0.0/
```

```
     2      , ICON/9*0/, IREP/2/

C input sampling rate, T(1)                                    C
      WRITE(6,98)
      CALL FREAD(5,'(R):',T(1))
      WRITE(11,109)T(1)
      T(2) = .20*T(1)

C determine if the simultaion is for a linear or
C nonlinear system
      WRITE(6,116)
      CALL FREAD(5,'(I):',ISIM)
      WRITE(11,110)ISIM
C input number of rows, IR, number of columns, IC and order
C of highest model
      WRITE(6,97)
      CALL FREAD(5,'(3I):', IR, IC, IO)
      WRITE(11,110) IR, IC, IO

      IF(ISIM .EQ. 2)RETURN
C input model type and time delay in each loop                 C
      WRITE(6,100)
      DO 50 L=1, IR
        DO 49 M=1, IC
          WRITE(6,114)L,M
          CALL FREAD(5,'3(I):',ITD(L,M),ITD2(L,M),ICON(L,M))
          WRITE(11,115)ITD(L,M), ITD2(L,M), ICON(L,M)
C
C transfer control to appropriate block                        C
      IFUNC=ICON(L,M)
      GOTO(5,10,15,20,25,30,35,40), IFUNC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                              C
C    ..........................................................C
C    : block 1                                                 C
C    : G(s) = K/s                                              C
C    ..........................................................C
C                                                              C
C  input K, the gain                                           C
    5 CONTINUE
      WRITE(6,102)
      CALL FREAD(5,'(R):',K)
      WRITE(11,109)K

      DO 6 I = 1, IREP
        GD(I,L,M,1)= 1.0
        GN(I,L,M,1)= K*T(I)
    6 CONTINUE

      GOTO 48

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                              C
C   ...........................................................C
```

```
C    . block 2                                                        C
C    . G(s) = K/( s + r )                                             C
C    ..............................................................  C
C                                                                     C
C input K, the gain and R                                             C
   10 CONTINUE
      WRITE(6,103)
      CALL FREAD(5,'2(R):', K, R)
      WRITE(11,109)K,R
      DO 11 I = 1, IREP
       GD(I,L,M,1) = EXP(-R*T(I))
       GN(I,L,M,1) = K/R*(1.0 - EXP(-R*T(I)))
   11 CONTINUE

      GOTO 48

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C     ..............................................................C
C     . block 3                                                       C
C     . G(s)= omega**2 / (s*s + 2*zeta*omega*s + omega**2            C
C     ..............................................................C
C                                                                     C
C input zeta, the damping coefficent, omega and the gain, K
   15 CONTINUE
      WRITE(6,104)
      CALL FREAD(5,'3(R):', ZETA, OMEGA, K)
      WRITE(11,109)ZETA,OMEGA,K
      CHECK= ZETA-1.0
      I=0
      J = 0
      IF( CHECK )16,17,18

C zeta .lt. 1  ( underdamped system  )                                C
   16 CONTINUE
      I = I + 1
        DEL = ZETA*OMEGA
        OMEGA1=OMEGA*SQRT(1-ZETA*ZETA)
        GAMMA = ZETA/SQRT(1-ZETA*ZETA)

        GD(I,L,M,1) = 2.0*EXP(-DEL*T(I))*COS(OMEGA1*T(I))
        GD(I,L,M,2) = -EXP(-2*DEL*T(I))
        GN(I,L,M,1) = 1.0 - EXP(-DEL*T(I))*(COS(OMEGA1*T(I))
     1        +GAMMA*SIN(OMEGA1*T(I)))
        GN(I,L,M,2) = EXP(-DEL*T(I))*(EXP(-DEL*T(I))-COS(
     1        OMEGA1*T(I))+ GAMMA*SIN(OMEGA1*T(I)))

      IF(I-IREP)16,19,19

C zeta .eq. 1 (critically damped system  )                            C
   17 CONTINUE
      I = I + 1
        GD(I,L,M,1) = 2.0*EXP(-OMEGA*T(I))
        GD(I,L,M,2) = -EXP(-2.0*OMEGA*T(I))
```

```
      GN(I,L,M,1) =  1.0-EXP(-OMEGA*T(I))*(1.0+OMEGA*T(I))
      GN(I,L,M,2) = EXP(-OMEGA*T(I))*(EXP(-OMEGA*T(I))
     1     +OMEGA*T(I))- 1.0)

     IF(I - 1)17,19,19
                                                              C
C zeta .gt. 1 (overdamped system )
   18 CONTINUE
     I = I + 1
      OMEGA1 = OMEGA*SQRT(ZETA*ZETA -1.0)
      GAMMA = ZETA/SQRT(ZETA*ZETA - 1.0)

      GD(I,L,M,1) = 2*EXP(-DEL*T(I))*COSH(OMEGA1*T(I))
      GD(I,L,M,2) = -EXP(-2.0*DEL*T(I))
      GN(I,L,M,1) =1.0 - EXP(-DEL*T(I))*(COSH(OMEGA1*T(I))
     1      + GAMMA*SINH(OMEGA1*T(I)))
      GN(I,L,M,2) = EXP(-DEL*T(I))*(EXP(-DEL*T(I)) - COSH(
     1      OMEGA1*T(I)) + GAMMA*SINH(OMEGA1*T(I)))

      IF(I - IREP)18,19,19

   19 CONTINUE
      J = J + 1

      GN(J,L,M,1) = K*GN(J,L,M,1)
      GN(J,L,M,2) = K*GN(J,L,M,2)
      IF(J - IREP)19,40,40
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                            C
C                                                            C
C   .........................................................C
C   . block 4                                                C
C   . G(s) = K/( (s+r)*(s+p) )                               C
C   .........................................................C
C                                                            C
C
C input K, r, p
   20 CONTINUE
      WRITE(6,105)
      CALL FREAD(5,'3(R):', K, R, P)
      WRITE(11,109)K,R,P

      DO 21 I = 1, IREP
        GD(I,L,M,1) = EXP(-R*T(I))+ EXP(-P*T(I))
        GD(I,L,M,2)   -EXP(-(R+P)*T(I))
        GN(I,L,M,1)= (K/(R*P*(R-P)))*((R-P)-R*EXP(-P*T(I))+
     1      [illegible]P*EX[illegible]-R*T(I)))
        GN(I,L,M,[illegible] = [illegible] (R*P*(R-P)))*((R-P)*EXP(-(R+P)*T(I))
     1      -P[illegible] (-P*T(I[illegible] P*EXP(-R*T(I)))
   21 CONTINUE

      GOTO 48
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                            C
```

```
C     ..........................................................C
C     . block 5                                                 C
C     . G(s) = K/(s(s+r))                                       C
C     ..........................................................C
C                                                               C
C input K and r                                                 C
   25 CONTINUE
      WRITE(6,106)
      CALL FREAD(5,'2(R):', K, R)
      WRITE(11,109)K,R

      DO 26 I = 1, IREP
        GD(I,L,M,1) = 1 + EXP(-R*T(I))
        GD(I,L,M,2) = -EXP(-R*T(I))
        GN(I,L,M,1) = (K/(R*R))*(1-R*T(I)-EXP(-R*T(I)))
        GN(I,L,M,2) = (K/(R*R))*(1-EXP(-R*T(I))-R*T(I)*
     1          EXP(-R*T(I)))
   26 CONTINUE

      GOTO 48

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                         C
C   .......................................................C
C   . block 6                                             C
C   . G(s) = K*(s+q)/( (s+r)*(s+p)                        C
C   .......................................................C
C                                                         C
C input K, q, r, p                                        C
   30 CONTINUE
      WRITE(6,107)
      CALL FREAD(5,'4(R):', K, Q, R, P)
      WRITE(11,109)K,Q,R,P

      DO 31 I = 1, IREP
        GD(I,L,M,1) = EXP(-P*T(I)) + EXP(-R*T(I))
        GD(I,L,M,2) = -EXP(-(R+P)*T(I))
        GN(I,L,M,1) =-K/(P-R)*(EXP(-P*T(I))-EXP(-R*T(I))
     1   +(Q/P)*(1-EXP(-P*T(I))) - (Q/R)*(1-EXP(-R*T(I))))
        GN(I,L,M,2) = K*(Q/(R*P)*EXP(-(R+P)*T(I))+((P-Q)/
     1          (P*(R-P))*
     1          EXP(-R*T(I))+((Q-R)/(R*( R-P)))*EXP(-P*T(I)))
   31 CONTINUE

      GOTO 48

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                         C
C    ......................................................C
C    . block 7                                            C
C    . G(s) = K*(s+q)/(s*(s+r))                           C
C    ......................................................C
C                                                         C
C input K, q, r                                           C
```

```
   35 CONTINUE
      WRITE(6,108)
      CALL FREAD(5,'3(R):', K, Q, R)
      WRITE(11,109)K,Q,R

      DO 36 I = 1, IREP
        GD(I,L,M,1) = 1.0 + EXP(-R*T(I))
        GD(I,L,M,2) = -EXP(-R*T(I))
        GN(I,L,M,1) = -K/R*(Q*T(I)+((R-Q)/R)*(1-EXP(-R*T(I
     1      ))))
        GN(I,L,M,2) = -K/R*(Q*T(I)*EXP(-R*T(I)) + ((R-Q)/R)
     1      *(1-EXP(-R*T(I))))

      GOTO 48
   36 CONTINUE

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                    C
C   ..................................................C
C   . block 8                                        C
C   . G(s) = K*(s+q)/(s+r)                           C
C   ..................................................C
C                                                    C
C input K,Q,R
   40 CONTINUE
      WRITE(6,108)
      CALL FREAD(5,'3(R):', K, Q, R)

      DO 46 I=1,3
        GD(I,L,M,1) = EXP(-R*T(I))
        GN(I,L,M,1)= K
        GN(I,L,M,2) = -K*(1+Q/R*(GD(I,L,M,1)-1))
   46 CONTINUE

   48 CONTINUE

   49   CONTINUE
   50 CONTINUE
      RETURN
C  FORMAT LINE **                                       C
   97 FORMAT(/' enter the number of rows, IR where IR<4',/
     1 ' enter the number of columns, IC where IC<4',/,
     2 ' enter the highest model order, IO where IO =1 or'
     2 ' 2',/,
     3 ' enter these three values on the next line',/,
     4 ' if ISIM=2 IR=IC=IO=1 ')
   98 FORMAT(/' enter the sampling rate, T ')
  100 FORMAT(/' below are the eight transfer function types'
     1       ,/,' 1. G(s) = K/s ',/,
     2       ' 2. G(s) = K/(s+r) ',/,
     3       ' 3. G(s) = K*omega**2/( s*s + 2*zeta*omega*s',
     3       ' +omega**2)'
     4       ,/,' 4. G(s) = K/( (s+r)(s+p) )',/,
     5       ' 5. G(s) = K/( s(s+r) ) ',/,
```

```
  6      ' 6. G(s) = K (s+q)/( (s+r)(s+p) ) ',/,
  7      ' 7. G(s) = K (s+q)/(s(s+r))',/,
  8      ' 8. G(s) = K (s+q)/(s+r)',/,
  9      ' enter the element time delay for rate T(1)',/,
  1      ' the extra delay required for for rate T(2)',/,
  2      ' and the transfer function type(1-8)')
102 FORMAT(' enter: K')
103 FORMAT(' enter: K and R')
104 FORMAT(' enter: zeta, omega and K ')
105 FORMAT(' enter: K, R, P')
106 FORMAT(' enter: K, R')
107 FORMAT(' enter: K, Q, R , P')
108 FORMAT(' enter: K, Q, R')
109 FORMAT(4F10.5)
110 FORMAT(4I4)
111 FORMAT(//' TRUE PARAMETERS ')
112 FORMAT(6X,4(F10.5,3X))
113 FORMAT(//' PARAMETER ESTIMATES ',/,2X,' ITER ',4X,
  1      ' THETA(I)')
114 FORMAT(5X,' element ',3X,I4,2X,I4)
115 FORMAT(2X,I4,3X,I4,3X,I4)
116 FORMAT(' enter the simulation type ,linear=1,'
  1      ,' nonlinear=2')
    END
```