THE UNIVERSITY OF ALBERTA

PERFORMANCE AND TUNING OF ADAPTIVE GENERALIZED PREDICTIVE CONTROL

BY

ANDREW R. McINTOSH

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

PROCESS CONTROL

DEPARTMENT OF CHEMICAL ENGINEERING

EDMONTON, ALBERTA

FALL, 1988

ISBN 0-315-45765-1

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR: Andrew R. McIntosh

TITLE OF THESIS: Performance and Tuning of Adaptive Generalized Predictive Control

DEGREE: Master of Science

YEAR THIS DEGREE GRANTED: Fall, 1988

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

(SIGNED) Andrew R. McIntosh

PERMANENT ADDRESS:

312 2624 Millwoods Rd. E.

Edmonton, Alberta

T6L 5K7

DATE: Oct 1?, 1988

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 PERFORMANCE AND TUNING OF ADAPTIVE GENERALIZED PREDICTIVE CONTROL submitted by ANDREW R. McINTOSH in partial fulfilment of the requirements for the degree of MASTER OF SCIENCE in PROCESS CONTROL.

....

Dr. D.G. Fisher,
Supervisor

....

Dr. S.L. Shah

....

Dr. V. Gourishankar

DATE: October 6, 1988

To Laurie and to Maria

ABSTRACT

The objective of this thesis was to evaluate the performance and recommend strategies for commissioning and tuning adaptive Generalized Predictive Control (GPC). Experimental and analytical studies demonstrated that GPC can be successfully applied to processes ranging from simple low-order, stable plants to complex, open-loop unstable, nonminimum phase plants with unknown, variable time-delays and nonstationary disturbances. GPC is a flexible algorithm which, for particular settings of the design parameters, reduces to most of the well-known adaptive algorithms including Generalized Minimum Variance and Pole Placement.

The key analytical development was the transformation of the standard GPC control law into an equivalent linear form which made it possible to derive the closed-loop transfer function. This provided a means of comparing different GPC design alternatives and allowed the effect of the various tuning parameters to be determined. It was demonstrated that the majority of the controller parameters can be assigned appropriate fixed values and the speed of response varied by adjusting either the maximum output horizon, control weighting or reference model pole location.

The closed-loop output response was shown to be invariant to changes in process gain and insensitive to changes in dynamics or dead-time, given an accurate process model. For practical applications, where model-plant mismatch is inevitable, a design polynomial introduced into the feedback path is critical for improving robustness to modelling errors and tailoring the rejection of disturbances. This polynomial has little effect on the setpoint-tracking properties of the closed-loop and therefore allows GPC to meet independent servo and regulatory control objectives. A bandpass filter should also be used to focus parameter estimation on frequencies around the closed-loop bandwidth and reduce the deleterious impact of unmodeled dynamics and measurement noise. In addition, it was demonstrated that a simple supervisory system could be used to adjust tuning parameters such that the actual servo and regulatory performance approaches user-specifications.

Finally, a comparison was made between GPC and a Pole Placement controller obtained via the traditional Sylvester equation approach. In most experimental runs, Pole Placement was almost as effective but was sensitive to overspecification of the model order. GPC is recommended for process control applications.

ACKNOWLEDGEMENTS

I would like to thank Dr. D.G. Fisher and Dr. S.L. Shah for their technical assistance and supervision during the course of this research. Special thanks are also due to Robert Barton, of the DACS centre, for his never-ending patience in answering questions related to computer system hardware and software.

The work on this thesis was made more enjoyable by the other graduate students in Process Control. In particular, I owe much to discussions with Brendan Minter, Dave Shook, Mario Fortuna, Brenda Mounce and Andre Vien.

The financial support from the Natural Sciences and Engineering Research Council of Canada, the Department of Chemical Engineering and Imperial Oil Limited is gratefully acknowledged.

Finally, I take the opportunity to thank my parents for their constant support and encouragement over so many years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

General Nomenclature

A(.)	Polynomial in the argument "."
$\hat{A}$(.)	Estimated polynomial
A_0(.)	True Polynomial
A'(.)	Polynomial corresponding to C=1 or K_p=1
δA, na	Degree of polynomial A
a_i	i^{th} coefficient of polynomial A
A_j(.)	Polynomial corresponding to prediction interval j
$a_{j,i}$	i^{th} coefficient of polynomial A_j
A(1)	Steady state gain of A
A(0)	Initial coefficient of A
a(t)	Value of sampled signal at time t
a_{ss}	Steady state value of a(t)
$\hat{a}(t+j\|t)$	j-step ahead prediction of a(t) given data available at the current time t
a	Vector a
A	Matrix A
E{A}	Expected value of A given data available at time t
tr(A)	Trace of matrix A
A^T	Transpose of matrix A
$A_n(.)/A_d(.)$	Transfer function A

Alphabetic Symbols

$A(q^{-1})$	Process model denominator polynomial

$\overline{A}(q^{-1})$	Product of A and Δ
$B(q^{-1})$	Process model numerator polynomial
$C(q^{-1})$	Noise model polynomial
$C_c(q^{-1})$	Controller regulatory design polynomial
c_1	Repeated root of C_c polynomial
$C_e(q^{-1})$	Estimator design filter polynomial
c_e	Repeated root of C_e polynomial
$D(q^{-1})$	Feedforward model polynomial
d	Physical dead-time of process in sampling intervals, not including ZOH ($d \geq 0$)
d_{max}	Maximum expected physical dead-time
$d_u(t)$	step "input" type disturbance
$d_y(t)$	step "output" type disturbance
e(t)	Error between setpoint and process output
$E_j(q^{-1})$	Polynomial from j-step ahead Diophantine identity
f(t+j)	Open-loop prediction of y(t+j) assuming all present and future control increments are zero
f	Vector of future predicted outputs
$F(q^{-1})$	Polynomial in PP control law
$F_j(q^{-1})$	Polynomial from j-step ahead Diophantine identity
F_{sp}	Transfer function setpoint prefilter
$G(q^{-1})$	Polynomial in PP control law
$G_j(q^{-1})$	Product of E_j and B
$\tilde{G}_j(q^{-1})$	Polynomial obtained from G_j and C
$\tilde{g}_j$	Coefficient of the polynomial $\tilde{G}_j$
$\overline{G}_j(q^{-1})$	Polynomial obtained from G_j and C
$G_p(s)$	Process transfer function
G	Matrix of elements from $\tilde{G}_j$

G_r	Matrix of elements from $\hat{G}_j$ reduced using control horizon
h	Prediction horizon for EHAC
h	Row vector of elements from the 1st row of the matrix $(G_r^T G_r + \lambda I)^{-1} G_r^T$
h_j	Element of the vector **h**
$H(q^{-1})$	Polynomial in PP control law
$H(q^{-1})$	Impulse response model polynomial
I	Identity matrix
J	Value of a quadratic cost function
k	Process dead-time in sampling intervals, including ZOH ($k \geq 1$)
K_c	Proportional gain
K_d	Disturbance transfer function static gain
K_p	Process static gain
$M(q^{-1})$	Closed-loop reference model
$M_r(q^{-1})$	Regulatory closed-loop reference model
$M_s(q^{-1})$	Servo closed-loop reference model
n	maximum value of na+1 and nb
na	order of process (model); degree of $A(q^{-1})$
nb	degree of $B(q^{-1})$
N_1	Minimum output horizon
N_2	Maximum output horizon
NU	Control horizon
N_{min}	Minimum value of N_2 for which $\text{sign}\left[\sum_{j=1}^{N_2} g_{j-1}\right] = \text{sign}(K_p)$
o_v	Fractional overshoot
$P(q^{-1})$	Model-following polynomial (GPC); Desired characteristic polynomial (PP)
p_1	Root of 1st order P polynomial
$\overline{P}(q^{-1})$	Product of P and C

P_d	Desired closed-loop performance specification
$P_m(t)$	Measured closed-loop performance
$P_r(q^{-1})$	Desired regulatory characteristic polynomial
$P_s(q^{-1})$	Desired servo characteristic polynomial
q^{-1}	Backward shift operator, $q^{-1}a(t) = a(t-1)$
$Q(q^{-1})$	Dynamic control weighting transfer function
$r(t)$	Residual (Difference between actual and predicted output)
$R(q^{-1})$	Polynomial in the general linear form of the GPC control law
s	Laplace transform variable
$S(q^{-1})$	Polynomial in the general linear form of the GPC control law
$S_u^2(t)$	Control signal variance
S_v	Sylvester matrix
t_d	Physical dead-time of process
t_p	Tuning parameter setting
t_r	Rise time of process in sampling intervals
t_s	Settling time of process in sampling intervals
$t_{.63}$	63 % rise time of process
$T(q^{-1})$	Polynomial in the general linear form of the GPC control law
T_s	Sampling interval (time)
$u(t)$	Manipulated process input (control signal)
$\overline{u}(t)$	Mean value of control signal
$u^e(t)$	u(t) filtered by $1/C_e$
$u^f(t)$	u(t) filtered by $1/C_c$
$\tilde{u}$	Vector of present and future projected control increments
$v(t)$	Feedforward signal; Gaussian noise
$w(t)$	(Filtered) setpoint for controlled variable
w	Future setpoint sequence vector

$x(t)$	Nonstationary disturbance, $\xi(t)/\Delta$
$y(t)$	Measured process output (controlled variable)
$y_{sp}(t)$	Raw setpoint for controlled variable
y[illegible](t)	$y(t)$ filtered by $1/C_e$
y[illegible]	$y(t)$ filtered by $1/C_c$
$\hat{y}_{OL}(\infty)$	Predicted open-loop settled value of the output
z	Z-transform variable

Greek Symbols

Δ	Differencing operator, $1-q^{-1}$
$\epsilon(t)$	*A priori* estimation error
$\phi(t)$	I/O regressor vector
$\psi(t)$	Auxiliary process output
θ	Process model parameter vector
λ	(Absolute) control weighting
λ_{rel}	Relative control weighting
λ_{um}	Forgetting factor for control signal mean value calculation
λ_{uv}	Forgetting factor for control signal variance calculation
σ^2	Variance of the noise
τ	Natural period
τ_c	Continuous time (repeated) root of the C_c polynomial
τ_{CL}	Closed-loop time constant
τ_D	Derivative time
τ_I	Integral time
τ_P	Process time constant
ω	Frequency

$\xi(t)$	Uncorrelated random sequence of zero mean
ζ	Damping factor

Abbreviations

ARMA	Autoregressive Moving-Average
CARMA	Controlled Autoregressive Moving-Average
CARIMA	Controlled Autoregressive Integrated Moving-Average
CL	Closed-Loop
DMC	Dynamic Matrix Control
DMF	Detuned Model-Following configuration of GPC
EHAC	Extended Horizon Adaptive Control
ELS	Extended Least Squares
EPSAC	Extended Prediction Self-Adaptive Control
EXACT	Expert Adaptive Controller Tuning
FOM	Full Order Model
I on SP	PI(D) controller with only Integral action on Setpoint changes
IDCOM	Identification/Command
I/O	Input/Output
GMV	Generalized Minimum Variance
GPC	Generalized Predictive Control
GPP	Generalized Pole Placement
LQ	Linear Quadratic
LW	Lambda Weighting configuration of GPC
MIMO	Multi-Input Multi-Output
MOCCA	Multivarible Optimal Constrained Control Algorithm
MPM	Model-Plant Mismatch

MUSMAR	Multistep Multivariable Adaptive Regulator
MV	Minimum Variance
NMP	Nonminimum Phase
OH	Output Horizon configuration of GPC
OL	Open-Loop
PCA	Predictive Control Algorithm
PI(D)	Proportional Integral (Derivative)
PI(D) on SP	PI(D) controller with Proportional, Integral (and Derivative) action on Setpoint changes
PP	Pole Placement
PRBS	Psuedo Random Binary Sequence
RLS	Recursive Least Squares
RML	Recursive Maximum Likelihood
ROM	Reduced Order Model
SISO	Single-Input Single-Output
SP	Setpoint
T/C	Thermocouple
ZOH	Zero Order Hold

1. INTRODUCTION

An industrial chemical process represents a challenging environment for automatic control due to the presence of nonlinearities, large time-delays, a wide range of unmeasurable disturbances, and interactions from other control loops. Measurements are typically corrupted by noise and quantization errors; actuators are nonlinear and reach saturation limits during normal operation. Furthermore, the plant itself may be unstable and/or exhibit unusual dynamics such as an inverse response.

The three-term PID controller has gained strong acceptance in the chemical process industry as a result of its simplicity and robustness. There are, however, fundamental limitations of the fixed parameter PID structure: the controller tuning procedure maybe time consuming and must be repeated if there are significant changes in the process. In addition, time delays can only be handled by detuning the loop which leads to more sluggish behavior.

Adaptive controllers, which adjust their parameters automatically to compensate for variations in the characteristics of the process, could theoretically compensate for actual time variations in the process (e.g. as a result of catalyst decay, or heat exchanger fouling) and/or changes in operating conditions for nonlinear plants. Early efforts at adaptive control (Kalman, 1958) were hampered by lack of theory and suitable equipment for implementation of the more sophisticated control laws. The unprecedented advances in microprocessor technology in the 1970's sparked renewed interest in adaptive control theory which in turn led to a large number of reported experimental applications (Seborg et al., 1986).

Despite a large diversity in approaches to adaptive control, the various designs may be classified into two principal categories:

1) Model-based techniques where the parameters of a process model are identified (directly or indirectly) and used to derive a suitable control law.

2) Expert system or pattern recognition methods where the parameters in a control law are adjusted based on measured closed-loop performance (i.e. overshoot measured once per underdamped transient response) combined with heuristic rules obtained from experience or process knowledge.

Model-based adaptive controllers are often referred to as "self-tuning" in the literature since they were originally conceived for systems with constant but unknown parameters (Clarke and Gawthrop, 1975). The term "parameter-adaptive" is also used (Isermann and Lachmann, 1985) to emphasize that the model parameters are adjusted on-line. In this thesis the terms "self-tuning" and "adaptive" are used interchangeably. A block diagram of a typical self-tuning control system is shown in Figure 1.1. (The figures throughout the thesis are located at the end of each chapter.)

1.1 Self-Tuning Control

Time delays are frequently encountered in chemical process control as a result of, for example, transportation of products over long distances or the finite amount of time required to sample and complete the analysis of a stream. In the former case, if the flowrate of material through the process is not constant, the time delay (or dead-time) will vary. Large time delays are the main source of difficulty for PID controllers (Stephanopoulos, 1984). Consideration of this problem led to the development of predictive control strategies; control is based not on the current output but rather a value of the output predicted at some time in the future based on knowledge of past control actions and a process model. The original Smith Predictor (Smith, 1957) provided time-delay compensation for a conventional fixed-parameter controller. Two early self-tuning algorithms, the Minimum Variance (Åström and Wittenmark, 1973) and Generalized Minumum Variance (Clarke and Gawthrop, 1975) controllers are based on k-step ahead prediction of the output (where k is the known time delay of the process, including the unit sampling delay). The certainty-equivalence principle is envoked whereby parameter estimates are used in the design of the control law assuming they are the true parameters of the process. These algorithms are known to be sensitive to an incorrect specification of the time delay as well as nonminimum phase process characteristics.

Two alternative strategies for model-based adaptive control of industrial chemical processes with (variable) dead-time appear promising; long-range prediction and pole placement.

1.1.1 Long-range Multistep Predictive Control

The essential idea in long-range multistep predictive control is to consider a trajectory of future predictions beyond the largest expected pure time-delay. Making the prediction horizon significantly greater than the time delay normally yields a detuned controller which is more robust to modelling errors and poor measurements. Early long-range predictive control algorithms such as IDCOM (Richalet et al., 1978) and DMC (Cutler and Ramaker, 1980) were based on deterministic impulse or step response models, respectively. As a consequence of the large number of parameters in the process model representation, these controllers are not suitable for adaptive implementation.

Long-range predictive controllers based on autoregressive moving-average (ARMA) models have the potential to become self-tuning if combined with a recursive parameter estimation routine. Peterka's infinite-stage predictive controller (1984), the EPSAC algorithm (DeKeyser and Van Cauwenberghe, 1982) and the MUSMAR approach (Mosca et al., 1984) fall in this category. All of these algorithms require extensive computations; the former relies on matrix factorization and decomposition while the latter two involve a bank of self-tuning predictors for each prediction horizon. Generalized Predictive Control (Clarke et al., 1987a,b) is a recent generalization of the previous multistep predictive algorithms as well as being the natural long-range extension of GMV. It involves explicit identification of a process model and the minimization of a multistep quadratic cost function of future predicted errors and projected control increments. The algorithm represents a very flexible approach to self-tuning control as a consequence of the significant number of design parameters which may be specified by the user.

The Generalized Predictive Controller and its closed-loop transfer function are derived in Chapter 2. The theoretical analysis in Chapter 4 indicates that despite the large number of design parameters, the algorithm is easy to commission and tune. Based on the simulated and experimental applications in Chapters 5 and 7, it is concluded that GPC normally provides excellent control of chemical processes.

1.1.2 Pole Placement

Pole placement approaches to model-based adaptive control do not involve cancellation of open-loop zeros and are therefore inherently suitable for nonminimum phase plants. It is important to note that pole placement controllers, although not based on prediction, do take into account process dead-time. Compensation for variable time delays is achieved simply by overparameterizing the numerator of the estimated model. A discrete-time self-tuning pole placement regulator with explicit identification of the process model is described by Wellstead et al. (1979). It has its roots in classical control methods, wherein the control objective is to move the closed-loop poles to prespecified positions which define the desired transient response. Åström and Wittenmark (1980) derive several alternative pole/zero placement controllers which deal strictly with the servo problem. Extensions of the pole placement concept to include both setpoint following and regulation were produced independently by Wellstead and Sanoff (1981) and Clarke (1982). These algorithms are based on a model which assumes zero-mean noise and hence the resulting controllers do not have integral action. More recently, natural integral action has been introduced into the pole placement approach through the assumption of a model in which the noise term is nonstationary (discussed in the next section) so that offset between the output and setpoint approaches zero even for persistent load disturbances (Tuffs and Clarke, 1985).

The derivation of a Pole Placement controller with integral action is presented in Chapter 3. In the absence of near common factors in the model polynomials, the performance of this Pole Placement controller presented in Chapters 6 and 8 is comparable to that achieved using Generalized Predictive Control.

1.1.3 Process Models

Until recently, the most frequently assumed model structure for derivation of self-tuning controllers was the CARMA (Controlled Autoregressive Moving Average) model (Åström and Wittenmark, 1973):

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + C(q^{-1})\xi(t) \qquad (1.1.1)$$

where A, B and C are polynomials in the backward shift operator, q^{-1}:

$$A(q^{-1}) = 1 + a_1 q^{-1} + \dots + a_{na} q^{-na}$$
$$B(q^{-1}) = b_0 + b_1 q^{-1} + \dots + b_{nb} q^{-nb}$$
$$C(q^{-1}) = 1 + c_1 q^{-1} + \dots + c_{nc} q^{-nc}$$

u(t) is the control input, y(t) is the measured process output and $\xi(t)$ is an uncorrelated random sequence with zero mean. If the process has physical dead-time (not including the unit sampling delay) the leading elements of $B(q^{-1})$ are zero. Although this model has been commonly utilized, it would seem to be inappropriate for industrial chemical processes for which disturbances are nonstationary. Removal of offset for controllers based on the CARMA model is frequently accomplished by the *ad hoc* insertion of an integrator. The CARMA model may be reformulated to include a DC offset term:

$$Aq^{-1})y(t) = B(q^{-1})u(t-1) + C(q^{-1})\xi(t) + d' \qquad (1.1.2)$$

which takes into account nonzero mean values of the input and output. However, estimation of d' using the "1 in the data vector" method is reported to be unsuccessful (Tuffs, 1984) due to a lack of persistent excitation.

The use of a CARIMA (Controlled Autoregressive Integrated Moving Average) model:

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + C(q^{-1})\xi(t)/\Delta \qquad (1.1.3)$$

where Δ is the differencing operator, $1-q^{-1}$, has been suggested or recommended by many authors (Harris et al., 1982; Belanger, 1983; Peterka, 1984; McDermott and Mellichamp, 1984; Ydstie et al., 1985; Lelic and Wellstead, 1987). By various assumptions relative to the form of $C(q^{-1})$ and $\xi(t)$ the disturbance model can be interpreted as either stochastic Brownian motion or representing random steps at random times. Tuffs and Clarke (1985) have demonstrated that both predictive and pole placement controllers based on the CARIMA model contain inherent integral action. Equation (1.1.3) may be written in the equivalent incremental form:

$$A(q^{-1})\Delta y(t) = B(q^{-1})\Delta u(t-1) + C(q^{-1})\xi(t) \qquad (1.1.4)$$

where the desirable zero-mean nature of the data vector used in parameter estimation is apparent.

The CARIMA process model is used exclusively in this thesis.

1.2 Performance Adaptive Control

Industrial application of adaptive control has been limited primarily because the benefit of improved control is outweighed by the risk of poor behavior. Self-tuning controllers are susceptable to instability if there exists a large amount of model-plant mismatch or the design parameters are selected improperly. Since the actual (as apposed to predicted) performance is not monitored directly, under these circumstances, unacceptable closed-loop behavior may persist until an operator identifies and rectifies the situation. A suggested alternative approach to adaptive control involves evaluating the actual performance and adjusting controller parameters to insure that the actual closed-loop performance approaches user specifications.

1.2.1 Pattern Recognition Approach

Foxboro's Expert Adaptive Controller Tuning (EXACT) approach represents the most successful, commercially available, pattern recognition based adaptive controller. The algorithm monitors the closed-loop response following a load disturbance or setpoint change and automatically calculates P, I and D constants to minimize recovery time subject to user-specified damping and overshoot constraints (Kraus and Myron, 1984; Myron, 1986). While the EXACT controller is reported to work well in many applications, since it is based on a PID control structure, there is no compensation for time-delays (Minter and Fisher, 1988). The potential exists to achieve better control by combining the pattern recognition approach with a self-tuning controller.

1.2.2 Supervisor for Performance Adaptive Control

A performance supervisor would monitor the closed-loop behavior and adjust tuning parameters of the underlying self-tuning controller in order to achieve and maintain desired performance. The proposed "performance adaptive" control strategy removes the need for the user to specify the controller tuning parameters directly. The performance supervisor could not be expected to tune the closed-loop system if the controller is beyond its range of capabilities or if there existed gross modelling errors. Therefore, it represents only one element of a complete supervisory shell for a self-tuning controller.

In Chapter 10, a prototype performance supervisor is presented which adjusts GPC or PP controller tuning parameters in order to meet user specifications on actual servo and regulatory performance.

1.3 Objectives of Study

This thesis is concerned with single-input single-output (SISO) adaptive control of chemical processes. The objectives of the study, as originally conceived, were as follows:

a) theoretically analyze and interpret the Generalized Predictive and Pole Placement algorithms in order to provide practical guidelines on the selection of their design and tuning parameters

b) identify conditions under which closed-loop performance is or is not maintained when there are changes in process gain or dynamics

c) experimentally evaluate and compare self-tuning Generalized Predictive and Pole Placement control of chemical processes involving nonlinearities, measurement noise, variable time delays, etc.

d) develop a prototype performance supervisor which monitors closed-loop behavior and adjusts controller tuning parameters to meet user specifications on both servo and regulatory performance.

1.3.1 Structure of Thesis

The structure of the thesis is outlined below with emphasis on the contributions of this work. A more detailed introduction is given at the start of each individual chapter.

Chapter 2 reviews the development of the basic Generalized Predictive Control (GPC) algorithm and existing extensions which incorporate several design polynomials. The closed-loop transfer function is derived by expressing the control law in a general linear form. This serves as a basis for the design of self-tuning PID controllers based on GPC. For particular settings of the controller tuning parameters, it is shown that GPC reduces to many well-known control techniques.

Chapter 3 contains the derivation of a Pole Placement (PP) controller based on the CARIMA process model. The design of self-tuning PID controllers based on PP is presented.

Chapter 4 describes three alternative strategies available to a control engineer for selecting design parameters during the commissioning of GPC. The key to each strategy is the identification of a single active tuning parameter which may be adjusted by the user during operation to vary the overall speed of response. Requirements for maintaining output performance in spite of changes in process gain and dynamics are outlined. Finally, alternatives are discussed for including a noise model polynomial which, in the ideal case where the process model is exact, does not influence servo control.

Chapter 5 reports the results of an extensive simulation study of GPC, first with exact process models and then in the presence of model-plant mismatch. A root locus analysis provides insight into the effect of the tuning parameters. The importance of using controller (C_c) and estimator (C_e) design polynomials (representing prior knowledge of the noise model) for tailoring the rejection of disturbances and achieving robustness to modelling errors is emphasized.

Chapter 6 deals with the performance and tuning of Pole Placement controllers. A simulation study, which parallels that in Chapter 5, is presented and discussed.

Chapter 7 includes the results of the experimental application of self-tuning GPC to a stirred tank heater and to a set of interacting tanks. Runs with the former pilot-scale plant demonstrate the necessity of using the C_c and C_e polynomials so that GPC is robust to measurement noise.

Trials with the nonlinear interacting tanks demonstrate maintenance of performance for large changes in process dynamics.

Chapter 8 contains experimental runs for self-tuning PP control which are directly comparable to those in the previous chapter.

Chapter 9 is a comparison between Generalized Predictive and Pole Placement control. The comparison is based on analytical and experimental results presented in the previous chapters.

Chapter 10 begins with the description of a proposed hierarchical supervisory system for a (self-tuning) controller. Some of the functions which should be incorporated into such a system are listed. A significant contribution of the chapter is the development of a performance supervisory loop which gives self-tuning controllers the ability to acheive and maintain user-specified servo and regulatory closed-loop performance. The performance supervisor yields excellent results for GPC and PP control of the stirred tank heater.

Chapter 11 draws conclusions from the results presented in earlier chapters and provides suggestions for future work.

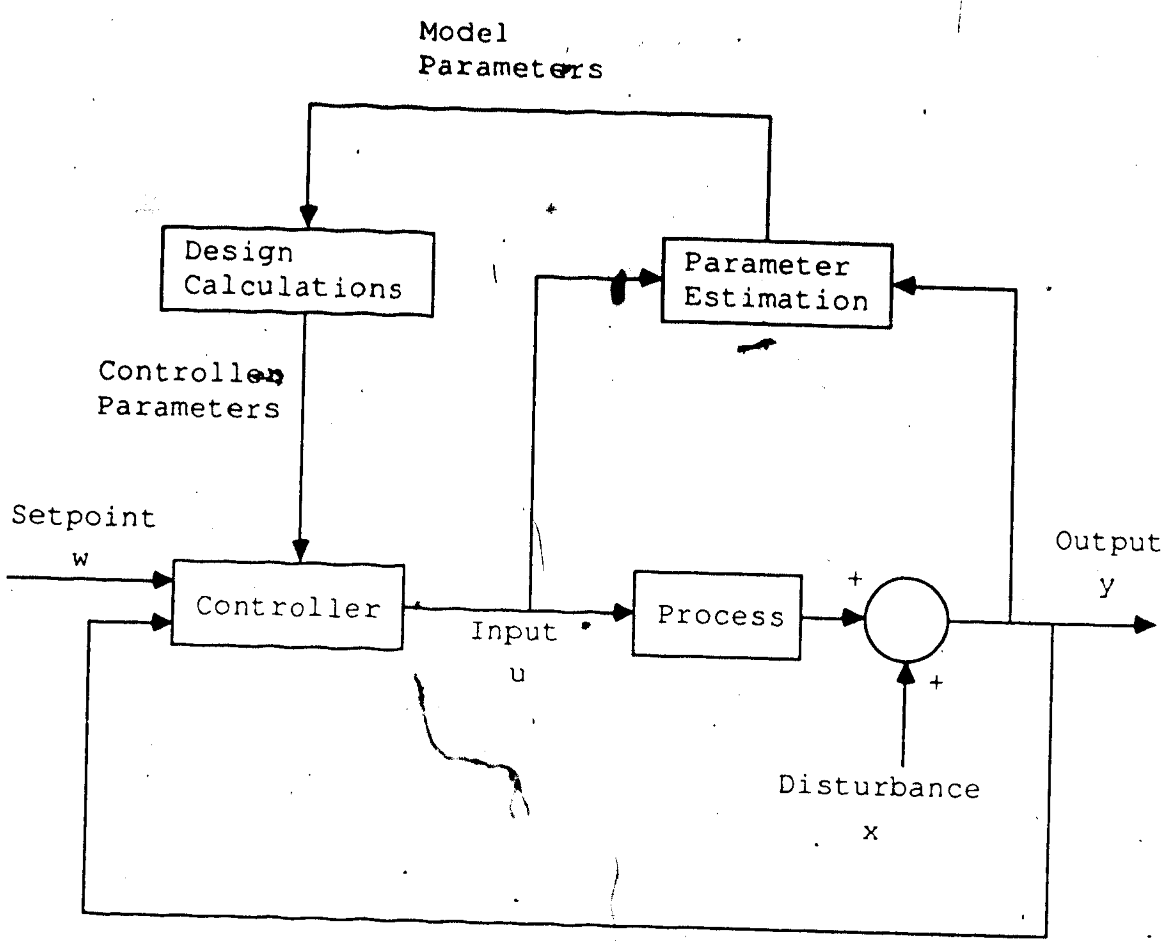


Figure 1.1 Block Diagram of an Explicit Self-Tuning Controller.

2. GENERALIZED PREDICTIVE CONTROL

Generalized Predictive Control (Clarke et al., 1987a,b) represents a unification of many earlier long-range, multistep predictive control techniques. Compared with these earlier approaches, GPC is more flexible and requires less computational effort. The ability of the strategy to overcome many of the problems in fixed gain or self-tuning control depends upon the integration of five key ideas:

1) the use of long-range prediction over a finite multistep horizon which extends beyond the dead-time of the process enables compensation for variable and unknown time delays

2) the assumption of a CARIMA process model and

3) the (optional) weighting of control *increments* in the cost function ensures offset-free rejection of nonstationary disturbances as a result of inherent integral action

4) recursion of the Diophantine equation is a computationally efficient approach which allows for the inclusion of weighting polynomials to modify the predicted output trajectory

5) the choice of a control horizon (first introduced by Cutler and Ramaker, 1980) after which projected control increments are set to zero significantly reduces computational effort and simplifies control of nonminimum phase plants (since control weighting is no longer *necessary*).

GPC is capable of stable control of processes which are simultaneously nonminimum phase and open-loop unstable and whose models may be overparameterized by the identification scheme (Clarke et al., 1985).

The basic Generalized Predictive Control algorithm (Clarke et al., 1987a) is derived in Section 2.1 and extended to incorporate polynomial weightings for modifying the closed-loop response (Clarke et al., 1987b) in section 2.2. The large number of design parameters gives GPC great flexibilty but, in general, makes the algorithm difficult to tune. (It is difficult to obtain a useful closed-loop expression directly in terms of these design parameters.) However, in Section 2.3, the standard control law is rearranged into an equivalent general linear form which makes it possible to derive the closed-loop transfer function. The effect of the various design parameters can then be evaluated by analyzing this closed-loop expression. The linear form of the control law is also useful for demonstrating that GPC reduces to a number of well-known strategies (Section 2.5) for limiting cases of the design parameters. Restrictions which must be placed on the process model in order for GPC to reduce to a PI or PID controller are outlined in Section 2.4. Design equations are given for the proportional, integral and derivative constants in terms of coefficients in the general linear form of the control law.

2.1 Derivation of the Basic Algorithm

The development of the Generalized Predictive Control (GPC) algorithm depends upon long-range prediction of the process output using a CARIMA model. The Diophantine identity, which is combined with the CARIMA model, may be solved recursively such that the prediction of the output over a multistep horizon is computationally efficient. Minimization of a cost function consisting of a combination of future predicted errors and control increments yields the control law. The introduction of a "control horizon" makes implementation of the algorithm practical for self-tuning applications.

The derivation of the basic GPC algorithm is based on the simplified CARIMA model where $C(q^{-1})$ is assumed to be 1:

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + \xi(t)/\Delta \qquad (2.1.1)$$

(The extension for the case of the general CARIMA model with $C(q^{-1}) \neq 1$ will be considered in a later section).

2.1.1 Output Prediction

To derive a j-step ahead predictor of $y(t+j)$ based upon eqn. (2.1.1) consider the identity:

$$1 = E_j(q^{-1})A(q^{-1})\Delta + q^{-j}F_j(q^{-1}) \qquad (2.1.2)$$

which uniquely defines the polynomials $E_j(q^{-1})$ and $F_j(q^{-1})$ given $A(q^{-1})$ and the prediction interval j. Multiplying eqn. (2.1.1) by $E_j\Delta q^j$ (and dropping the argument (q^{-1}) of the polynomials for notational simplification) gives:

$$E_jA\Delta y(t+j) = E_jB\Delta u(t+j-1) + E_j\xi(t+j)$$

so that after substituting for $E_jA\Delta$ from eqn. (2.1.2) we have:

$$y(t+j) = E_jB\Delta(t+j-1) + F_jy(t) + E_j\xi(t+j) \qquad (2.1.3)$$

Since the degree of E_j is j-1 the noise components are all in the future and since ξ is assumed to be white the optimal output predictor, given measured outputs up to time t and control signals u(t+i) for $i \le j-1$, is:

$$\hat{y}(t+j|t) = G_j\Delta u(t+j-1) + F_jy(t) \qquad (2.1.4)$$

Note that from eqn. (2.1.2) $G_j = BE_j = B[1-q^{-j}F_j]/A\Delta$ so that the first j coefficients of G_j are the step response coefficients (i.e. coefficients of $B/A\Delta$).

GPC is based on the minimization of a multistep cost function (section 2.1.3) where j ranges from the minimum output horizon (N_1) to the maximum output horizon (N_2). For j>d (where d is the physical dead-time of the process expressed as an integer multiple of the sampling interval) the predicted outputs depend upon present and future control actions.

2.1.2 Recursion of the Diophantine Equation

In order to obtain the range of output predictions from N_1 to N_2, the Diophantine equation (2.1.2) could be solved to give E_j and F_j for each j. A simplier and far more computationally efficient scheme is to use recursion of the Diophantine equation (Clarke et al., 1987a) so that the polynomials E_{j+1} and F_{j+1} are obtained from E_j and F_j. To show this recursion procedure consider the two Diophantine equations with $\overline{A}$ defined as $A\Delta$:

$$1 = E_j\overline{A} + q^{-j}F_j \qquad (2.1.5a)$$

$$1 = E_{j+1}\overline{A} + q^{-(j+1)}F_{j+1} \qquad (2.1.5b)$$

Subtracting (2.1.5a) from (2.1.5b) gives:

$$0 = (E_{j+1} - E_j)\overline{A} + q^{-j}(q^{-1}F_{j+1} - F_j)$$

The polynomial $E_{j+1}-E_j$ is of degree j and therefore may be split into two parts:

$$E_{j+1} - E_j = \tilde{E}_j + e_{j+1,j}q^{-j}$$

so that:

$$\tilde{E}_j\overline{A} + q^{-j}(q^{-1}F_{j+1} - F_j + \overline{A}e_{j+1,j}) = 0$$

Since the coefficients of the first j terms must be zero, $\tilde{E}_j=0$ and F_{j+1} is given by $F_{j+1} = q(F_j - \overline{A}e_{j+1,j})$.

As $\overline{A}$ is monic we have:

$$e_{j+1,j} = f_{j,0} \tag{2.1.6a}$$

$$f_{j+1,i} = f_{j,\,i+1} - \overline{a}_{i+1}e_{j+1,j} \qquad i = 0,..,na \tag{2.1.6b}$$

Also,

$$E_{j+1} = E_j + e_{j+1,j}q^{-j} \tag{2.1.7}$$

and,

$$G_{j+1} = BE_{j+1} \tag{2.1.8}$$

Hence given the process polynomials A and B and the solution for prediction interval j (i.e. E_j and F_j), (2.1.6) and (2.1.7) may be used to obtain E_{j+1} and F_{j+1} and then (2.1.8) used to calculate G_{j+1} with little computational effort. To initialize the iterations note that for j=1 the Diophantine equation is:

$$1 = E_1\overline{A} + q^{-1}F_1$$

and as $\overline{A}$ is a monic polynomial:

$$E_1 = 1, \quad F_1 = q(1-\overline{A}) \tag{2.1.9}$$

For implementation purposes it is possible to develop similar equations to compute G_{j+1} recursively from G_j and F_j in order to avoid calculating and storing the E polynomials.

2.1.3 The Predictive Control Law

Suppose a future setpoint sequence $[w(t+j);\ j=N_1,\ldots,N_2]$ is available at time t. In most cases the elements of this sequence will be equal to the current setpoint w(t) (which may be changed by the user at any time), although for batch process control or robotics a future "preprogrammed" setpoint sequence may be known. Optionally, w(t) may be obtained from a raw setpoint $y_{sp}(t)$ by prefiltering:

$$w(t) = F_{sp} y_{sp}(t) \qquad (2.1.10)$$

where F_{sp} is a transfer function filter.

The objective of the controller is to drive future process outputs y(t+j) "close" to w(t+j) bearing in mind the control activity required to do so. A receding horizon approach is adopted where at each sampling time t:

a) the future setpoint sequence is obtained,

b) the predictive model (2.1.4) is used to generate a set of predicted outputs $\hat{y}(t+j|t)$ and corresponding output errors, $e(t+j) = w(t+j) - \hat{y}(t+j|t)$ for $j=N_1,\ldots,N_2$ (noting that $\hat{y}(t+j|t)$ depends in part upon present and future control increments which have yet to be determined),

c) a quadratic cost function composed of the future predicted errors and control increments is minimized to provide the optimal sequence of controls, $\Delta u(t+j)$; $j=0,1,\ldots,N_2-1$

d) the first control increment, $\Delta u(t)$, of the sequence is implemented.

The cost function considered for the basic GPC algorithm is of the form:

$$J(N_1,N_2) = E\left\{ \sum_{j=N_1}^{N_2} [y(t+j) - w(t+j)]^2 + \sum_{j=1}^{N_2} \lambda(j)[\Delta u(t+j-1)]^2 \right\} \qquad (2.1.11)$$

where:

N_1 is the minimum output horizon ($N_1 \geq 1$)

N_2 is the maximum output horizon ($N_2 \geq N_1$)

$\lambda(j)$ is a control weighting sequence frequently chosen to be the constant λ.

The expectation in (2.1.11) is conditioned on data up to time t assuming that no future measurements are available. Recall that the optimal prediction of y(t+j) (for the process described by (2.1.1)) is given by equation (2.1.4):

$$\hat{y}(t+j|t) = G_j\Delta(t+j-1) + F_jy(t)$$

$\hat{y}(t+j|t)$, in general, depends upon

i) past and present measured outputs as well as past known control increments

ii) present and future control increments yet to be determined.

Let f(t+j) be that component of y(t+j) composed of signals known at time t (i.e. f(t+j) is the open-loop prediction of y(t+j) assuming all present and future control increments are zero). In order to determine f(t+j), G_j may be split into two polynomials as follows:

$$G_j = \tilde{G}_j + q^{-j}\overline{G}_j \qquad (2.1.12)$$

where the polynomial degrees are: δG_j = nb+j-1, $\delta\tilde{G}_j$ = j-1, $\delta\overline{G}_j$ = nb-1

The coefficients of $\tilde{G}_j$ are step response coefficients of the process so that we may drop the prediction interval (1st) subscript:

$$\tilde{G}_j = g_o + g_1q^{-1} + \dots + g_{j-1}q^{-(j-1)}$$

while,

$$\overline{G}_j = g_{j,j} + \dots + g_{j,j+nb-1}q^{-(nb-1)}$$

The optimal prediction becomes:

$$\hat{y}(t+j|t) = \tilde{G}_j\Delta u(t+j-1) + f(t+j) \qquad (2.1.13)$$

where,

$$f(t+j) = \overline{G}_j\Delta u(t-1) + F_jy(t) \qquad (2.1.14)$$

Equation (2.1.13) may be written for j = $N_1,\dots,N_2$ as follows:

$$\hat{y}(t+N_1|t) = \tilde{G}_{N_1}\Delta u(t+N_1-1) + f(t+N_1)$$

$$\vdots \qquad\qquad \vdots$$

$$\hat{y}(t+N_2|t) = \tilde{G}_{N_2}\Delta u(t+N_2-1) + f(t+N_2)$$

or in vector form:

$$\hat{y} = G\tilde{u} + f \tag{2.1.15}$$

where,

$$\hat{y} = [y(t+N_1)\ y(t+N_1+1)\ \dots\ \hat{y}(t+N_2)]^T$$

$$\tilde{u} = [\Delta u(t)\ \Delta u(t+1)\ \dots\ \Delta u(t+N_1-1)\ \dots\ \Delta u(t+N_2-1)]^T$$

$$f = [f(t+N_1)\ f(t+N_1+1)\ \dots\ f(t+N_2)]^T$$

and the matrix G is of dimension $(N_2-N_1+1) \times N_2$:

$$G = \begin{bmatrix} g_{N_1-1} & \dots & g_0 & 0 \\ \vdots & & & \\ & & & \\ g_{N_2-1} & \dots & & g_0 \end{bmatrix} \tag{2.1.16}$$

Again note that the elements of this matrix are the step response coefficients of the process so that if the plant has physical dead-time $d \geq N_1$ the first $d+1-N_1$ rows and last d columns of G will be entirely zeros.

With,

$$w = [w(t+N_1)\ w(t+N_1+1)\ \dots\ w(t+N_2)]^T$$

the expectation of the cost function (2.1.11) may be written:

$$J(N_1,N_2) = \left\{(w - G\tilde{u} - f)^T(w - G\tilde{u} - f) + \lambda\tilde{u}^T\tilde{u}\right\} \tag{2.1.17}$$

The minimization of J assuming no constraints on future controls results in the projected control increment vector:

$$\tilde{u} = \left[G^TG + \lambda I\right]^{-1}G^T(w - f) \tag{2.1.18}$$

Note that the first element of $\tilde{u}$ is $\Delta u(t)$ so the current control increment is obtained when the first row of $(G^TG + \lambda I)^{-1}G^T$ is multiplied by the error vector (w-f).

2.1.4 The Control Horizon

The matrix to be inverted in eqn. (2.1.18) when computing the current control increment is of dimension $N_2 x N_2$. Although in the non-adaptive case the inversion need only be performed once and may be done off-line, in the case of self-tuning the computational load at each sampling interval would be excessive since N_2 is typically large ($N_2 \simeq 10$ is commonly used). Moreover, if the plant has physical dead-time such that $d \geq N_1$ the matrix G^TG is singular and a finite nonzero value of control weighting, λ, is required for a realizable control law (i.e., for $(G^TG + \lambda I)^{-1}$ to exist).

The practicality of the GPC algorithm results from borrowing the idea of a control horizon from the Dynamic Matrix Control (DMC) method of Cutler and Ramaker (1980). After an interval $NU < N_2$ projected future control moves are constrained to be zero:

$$\Delta u(t+j-1) = 0 \qquad j > NU \tag{2.1.19}$$

The control horizon, NU, represents the number of nonzero control increments the algorithm is free to select in order to minimize the cost function. Control changes further in the future have an effective infinite weight placed on them. The use of the control horizon significantly reduces the computational effort since the vector $\tilde{u}$ is then of dimension NU and the prediction equations reduce to

$$\hat{y} = G_r\tilde{u} + f$$

where,

$$G_r = \begin{bmatrix} g_{N_1-1} & \cdots & g_0 & 0 \\ \vdots & & & \ddots \\ & & & g_0 \\ & & & \vdots \\ g_{N_2-1} & \cdots & & g_{N_2-NU} \end{bmatrix}_{(N_2-N_1+1)\times NU} \tag{2.1.20}$$

The control law is now:

$$\tilde{u} = \left[G_r^T G_r + \lambda I\right]^{-1} G_r^T (w - f) \tag{2.1.21}$$

and the matrix involved in the inversion is of dimension NUxNU. If NU=1, the calculation of $\Delta u(t)$ involves a simple scalar inversion.

The basic GPC algorithm is represented in block diagram form in Figure 2.1.

2.2 Extensions to the Basic Algorithm

The applicability and flexibility of the original Minimum Variance (MV) strategy (Åström and Wittenmark, 1973) was extended by the addition of design transfer functions $P(q^{-1})$ and $Q(q^{-1})$ to yield the Generalized Minimum Variance (GMV) controller (Clarke and Gawthrop, 1975, 1979). This section introduces similar transfer functions to the basic GPC algorithm. In addition, the derivation is extended to incorporate the general CARIMA model and feedforward compensation is also outlined.

2.2.1 Auxiliary Output; P Weighting

The basic GPC algorithm was based on the minimization of a set of predicted errors between the setpoint trajectory and the predicted output trajectory. Instead of predictions of the actual output, an auxiliary output,

$$\psi(t) = P(q^{-1})y(t) \tag{2.2.1}$$

may be considered, where $P(q^{-1})$ is a transfer function with numerator and denominator polynomials P_n and P_d selected by either the user or a supervisory system:

$$P(q^{-1}) = P_n(q^{-1})/P_d(q^{-1})$$

$P(q^{-1})$ should be selected such that $P(1)=1$ so that at steady state there is no offset between the auxiliary and actual outputs.

The appropriate cost function to be minimized is:

$$J(N_1,N_2,NU,P) = E\left\{ \sum_{j=N_1}^{N_2} [\psi(t+j) - w(t+j)]^2 + \sum_{j=1}^{NU} \lambda(j)[\Delta u(t+j-1)]^2 \right\} \qquad (2.2.2)$$

The prediction equations given in the previous section must be modified to forecast $\psi(t+j)$ rather than $y(t+j)$. Recall the simplified CARIMA model with $C(q^{-1}) = 1$:

$$Ay(t) = Bu(t-1) + \xi(t)/\Delta \qquad (2.2.3)$$

but now consider the Diophantine Identity:

$$P_n/P_d = E_jA\Delta + q^{-j}F_j/P_d \qquad (2.2.4)$$

Following the same procedure as in the previous section we obtain:

$$\hat{\psi}(t+j|t) = G_j\Delta u(t+j-1) + F_jy(t)/P_d \qquad (2.2.5)$$

where $G_j = E_jB$.

The Diophantine recursion equations developed earlier are identical to those involved here except the starting point is different:

$$E_1 = P_n(0)/P_d(0), \qquad F_1 = q(P_n - E_1\bar{A}P_d) \qquad (2.2.6)$$

where $\bar{A} = A\Delta$.

The selection of the $P(q^{-1})$ transfer function to achieve "model-following" will be detailed in later sections. There does not appear to be any evidence in the literature indicating that transfer function P weighting yields better results than simple polynomial P weighting (i.e. $P_d=1$). Consequently, in the analysis and applications which follow polynomial P weighting will be assumed. In this case the polynomials G_j and F_j are of the same degree as without P weighting (P=1) and the structure of the prediction equation remains unchanged.

2.2.2 Dynamic Control (Q) Weighting

Instead of constant (λ) or variable (λ(j)) costing on control increments it is possible to use dynamic costing $\lambda Q(q^{-1})$ on the control signal with cost function:

$$J(N_1,N_2,NU,Q) = E\left\{ \sum_{j=N_1}^{N_2} [y(t+j) - w(t+j)]^2 + \sum_{j=1}^{NU} \lambda[Q(q^{-1})u(t+j-1)]^2 \right\} \quad (2.2.7)$$

In order to avoid the problem of λ-offset (Clarke and Gawthrop, 1979) $Q(q^{-1})$ must have a factor of $(1-q^{-1})$. The role of Q in GPC is much reduced compared to GMV where Q is critical for stabilization of nonminimum phase processes. For GPC, Q may be viewed as a fine tuning parameter which may be selected as the transfer function:

$$Q(q^{-1}) = Q_n(q^{-1})/Q_d(q^{-1}) \qquad \text{with } Q(1) = 0$$

Details on the inclusion of Q weighting are provided in (Mohtadi, 1986).

The selection of the Q transfer function for GPC, as with GMV, is difficult *a priori*. The GMV control law may be written in the conceptual feedback form:

$$u(t) = \frac{1}{\lambda Q}\left[\hat{\psi}(t+k|t) - w(t)\right] \quad (2.2.8)$$

The choice of,

$$\lambda Q(q^{-1}) = \lambda(1-q^{-1})/(1-\alpha q^{-1}) \quad (2.2.9)$$

corresponds to a PI regulator acting on the predicted error $\hat{\psi}(t+k|t)-w(t)$ (Morris et al., 1981) with gain K_c and integral time τ_I related to the constants α, λ and the sampling time T_s as follows:

$$\lambda = 1/K_c, \qquad \alpha = (\tau_I - T_s)/\tau_I \quad (2.2.10)$$

It was suggested that suitable values of K_c and τ_I be found from conventional tuning rules for PI regulators acting on the process without time delay (which has been effectively removed by the k step ahead prediction). This is inconvenient (at best) and forms the basis of statements that GMV with Q weighting is non-adaptive (Gawthrop, 1986) since periodic retuning is required to accomodate process variations (Minter and Fisher, 1988). The same difficulties are encountered when attempts are made to specify Q for GPC. Since better performance can be achieved with much less design effort by using other tuning parameters, the use of dynamic control costing, $Q(q^{-1})$, is not recommended and will not be considered further in this thesis.

2.2.3 Feedforward Compensation

The potential exists for self-tuning to make a contribution in industrial process control in the area of dynamic feedforward compensation. Feedforward signals may be either measured disturbances or control signals from other loops in a multiloop environment. The feedforward signal may be added to the simplified CARIMA model to yield:

$$A(q^{-1})\Delta y(t) = B(q^{-1})\Delta u(t-1) + D(q^{-1})\Delta v(t-1) + \xi(t) \qquad (2.2.11)$$

The feedforward signal adds a term to the predicted output:

$$\hat{y}(t+j|t) = G_j\Delta u(t+j-1) + F_j y(t) + E_j D\Delta v(t+j-1) \qquad (2.2.12)$$

It is evident that, in general, future values of v must be known or estimated. A model for the disturbance is therefore required, the simplest of which assumes that there are no future changes in the level of the disturbance:

$$E\left\{\Delta v(t+j-1)\right\} = 0 \qquad j=2,3,... \qquad (2.2.13)$$

or that v(t) consists of random steps at random times. More generally it may be assumed that:

$$\Delta v(t) = \left[V_n(q^{-1})/V_d(q^{-1})\right]\Delta\nu(t) \qquad (2.2.14)$$

where $\nu(t)$ is an uncorrelated random sequence. Since this model is not known *a priori*, a recursive extended least squares or equivalent estimator must be employed to identify the V_n and V_d polynomials. In practice, the model of eqn. (2.2.13) is usually adequate. For minimum phase systems it is possible to obtain exact rejection if the disturbance transfer function (feedforward) delay is greater than or equal to the control signal delay. Note that when the feedback controller is detuned (i.e. by using λ control weighting) feedforward compensation is also detuned. Details for one method of recovering the exact rejection property are provided in (Tuffs, 1984).

2.2.4 Colored Noise; the C Polynomial

At this point we will extend the derivation of the GPC algorithm to the general CARIMA model where $C \neq 1$:

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + C(q^{-1})\xi(t)/\Delta \qquad (2.2.15)$$

To make the derivation reasonably general we will include polynomial P weighting. Consider the Diophantine Identity:

$$CP = A\Delta E_j + q^{-j}F_j \qquad (2.2.16)$$

Multiplying (2.2.15) by $E_j\Delta q^j$ and substituting for $E_jA\Delta$ from eqn. (2.2.16) yields:

$$C\psi(t+j) = F_j y(t) + G_j\Delta u(t+j-1) + CE_j\xi(t+j) \qquad (2.2.17)$$

where $\psi(t+j) = Py(t+j)$ and $G_j = E_jB$, or

$$\psi(t+j) = F_j y^f(t) + G_j\Delta u^f(t+j-1) + E_j\xi(t+j) \qquad (2.2.18)$$

where the superscript f denotes a signal filtered by 1/C. Since $\delta E_j = j-1$ the optimal predictor is:

$$\hat{\psi}(t+j|t) = F_j y^f(t) + G_j\Delta u^f(t+j-1) \qquad (2.2.19)$$

The constraints and cost are in terms of $\Delta u(t+j-1)$, j=1,..,NU rather than $\Delta u^f(t+j-1)$ and therefore the prediction equation must be modified. Consider the identity:

$$G_j = \tilde{G}_jC + q^{-j}\overline{G}_j \qquad (2.2.20)$$

where $\delta\tilde{G}_j = j-1$ and $\delta\overline{G}_j = \max(nb-1, nc-1)$.

Substituting gives:

$$\hat{\psi}(t+j|t) = F_j y^f(t) + \tilde{G}_j \Delta u(t+j-1) + \overline{G}_j \Delta u^f(t-1) \tag{2.2.21}$$

As with the development of the basic algorithm, let f(t+j) be that component of $\hat{\psi}(t+j|t)$ composed of signals known at time t:

$$f(t+j) = F_j y^f(t) + \overline{G}_j \Delta u^f(t-1) \tag{2.2.22}$$

so that,

$$\hat{\psi}(t+j|t) = \tilde{G}_j \Delta u(t+j-1) + f(t+j) \tag{2.2.23}$$

Writing this equation for $j = N_1,..,N_2$ in vector form and introducing the control horizon, NU:

$$\hat{\psi} = G_r \tilde{u} + f \tag{2.2.24}$$

where,

$$\hat{\psi} = [\hat{\psi}(t+N_1)\ \hat{\psi}(t+N_1+1)\ \ldots\ \hat{\psi}(t+N_2)]^T$$

$$\tilde{u} = [\Delta u(t)\ \Delta u(t+1)\ \ldots\ \Delta u(t+NU-1)]^T$$

$$f = [f(t+N_1)\ f(t+N_1+1)\ \ldots\ f(t+N_2)]^T$$

$$G_r = \begin{bmatrix} \tilde{g}_{N_1-1} & \cdots & \tilde{g}_0 & 0 \\ & & & \ddots \\ \vdots & & & \tilde{g}_0 \\ & & & \vdots \\ \tilde{g}_{N_2-1} & \cdots & & \tilde{g}_{N_2-NU} \end{bmatrix} \tag{2.2.25}$$

The minimization of the cost function then yields the control law:

$$\tilde{u} = \left[G_r^T G_r + \lambda I\right]^{-1} G_r^T (w - f) \tag{2.2.26}$$

Remark: The C polynomial which has been incorporated into GPC may be either a controller design polynomial, C_c, or an estimate, $\hat{C}$, of the "true" C obtained by recursive identification. These alternative strategies will be explored further in later chapters.

2.2.4.1 Recursive Implementation

Once again the Diophantine recursion equations for computing E_j and F_j are identical except for the initialization:

$$E_1 = C(0)P(0), \qquad F_1 = q(CP - E_1\bar{A}) \tag{2.2.27}$$

where $\bar{A} = A\Delta$.

However, now the identity of eqn. (2.2.20) must also be solved to determine $\tilde{G}_j$ and $\bar{G}_j$. While it is possible to develop another set of recursion equations (as in Mohtadi, 1986) the following derivation shows that it is possible to make use of the standard set of equations developed thus far and relate the solution to the case where C=1.

Using primes to denote polynomials for the case when C=1, we many rewrite eqn (2.2.4) (with polynomial weighting) as:

$$P = E_j' \text{[illegible]} + q^{-j}F_j' \tag{2.2.28}$$

also rewriting equation (2.1.12):

$$G_j' = \tilde{G}_j' + q^{-j}\bar{G}_j' \tag{2.2.29}$$

Multiplying (2.2.28) by CB and substituting for $G_j' = E_j'B$:

$$CBP = C\left[\tilde{G}_j'A\Delta + q^{-j}(\bar{G}_j'A\Delta + BF_j')\right] \tag{2.2.30}$$

Now consider the case where C≠1. Multiplying the Diophantine Identity (2.2.16) by B and substituting for $G_j = E_jB$ from eqn. (2.2.20) yields:

$$CBP = \tilde{G}_jCA\Delta + q^{-j}(\bar{G}_jA\Delta + BF_j) \tag{2.2.31}$$

Equating the last two expressions and dividing by $CA\Delta$ gives:

$$\tilde{G}_j + q^{-j}(\bar{G}_jA\Delta + BF_j)/CA\Delta = \tilde{G}_j' + q^{-j}(\bar{G}_j'A\Delta + BF_j')/A\Delta \tag{2.2.32}$$

Therefore $\tilde{G}_j = \tilde{G}_j'$ and the elements of the matrix G_r involved in the control calculation are independent of C. (Ultimately, this leads to the property that servo control is unaffected by the C polynomial, when there is no model-plant mismatch. Details are provided in section 4.5.4). For the case where $P(q^{-1})=1$, these elements are step response coefficients of the process.

The previous observation leads to a simple method for computing $\overline{G}_j$. The Diophantine recursion subroutine can be called with C=1 giving $\tilde{G}_j' = \tilde{G}_j$ and again with C providing G_j. Then eqn. (2.2.20) may be used to compute $\overline{G}_j$ directly.

2.3 General Linear Form of Control Law

In the previous sections, the basic GPC algorithm was derived and subsequently extended to incorporate polynomial weightings. For the case where the algorithm is based on the simplified CARIMA model with C=1, the control law was written in the form:

$$\tilde{u} = \left[G_r^T G_r + \lambda I\right]^{-1} G_r^T (w - f) \tag{2.3.1}$$

where the open-loop j-step ahead prediction of the output,

$$f(t+j) = F_j y(t) + \overline{G}_j \Delta u(t-1) \tag{2.3.2}$$

and the polynomials are of degree: $\delta F_j = na$, $\delta\overline{G}_j = nb-1$

While the control law is easy to implement written in this way, this representation is not convienient for analysis. In this section, the control law is rearranged into a general linear form such that the closed-loop transfer function can be determined and the effect of the design and tuning parameters analyzed.

In order to carry out the derivation, the assumption is made that the elements of the future setpoint sequence supplied to the controller are simply equal to the current setpoint:

$$w = [w(t) \;\; \ldots \;\; w(t)]^T \tag{2.3.3}$$

With this assumption, note that the calculation of $\Delta u(t)$ involves the following terms:

$$w(t),\; y(t),\; y(t-1),\ldots,y(t-na),\; \Delta u(t-1),\ldots,\Delta u(t-nb)$$

Since the terms appear linearly, the control law may be expressed in the general linear form:

$$T(q^{-1})\Delta u(t) = R(q^{-1})w(t) - S(q^{-1})y(t) \tag{2.3.4}$$

where $\delta T=nb$, $\delta R=0$ and $\delta S=na$.

It remains only to determine the coefficients of the polynomials $T(q^{-1})$, $R(q^{-1})$ and $S(q^{-1})$. To this end, let the elements of the first row of the matrix $(G_r^T G_r + \lambda I)^{-1} G_r^T$ be denoted by the row vector:

$$h^T = \begin{bmatrix} h_{N_1} & h_{N_1+1} & \dots & h_{N_2} \end{bmatrix}^T$$

So the calculation of $\Delta u(t)$ is,

$$\Delta u(t) = \begin{bmatrix} h_{N_1} & \dots & h_{N_2} \end{bmatrix} \begin{bmatrix} w(t) - F_{N_1} y(t) - \overline{G}_{N_1} \Delta u(t-1) \\ \vdots \\ w(t) - F_{N_2} y(t) - \overline{G}_{N_2} \Delta u(t-1) \end{bmatrix} \tag{2.3.5}$$

or,

$$\Delta u(t) = \sum_{j=N_1}^{N_2} h_j w(t) - \sum_{j=N_1}^{N_2} h_j F_j y(t) - \sum_{j=N_1}^{N_2} h_j \overline{G}_j \Delta u(t-1) \tag{2.3.6}$$

And by comparison with eqn (2.3.4):

$$T = 1 + q^{-1}\left[\sum_{j=N_1}^{N_2} h_j \overline{G}_j\right], \quad R = \sum_{j=N_1}^{N_2} h_j \quad \text{and} \quad S = \sum_{j=N_1}^{N_2} h_j F_j \tag{2.3.7}$$

Or expanding the polynomials in q^{-1}:

$$T = 1 + \left[\sum_{j=N_1}^{N_2} h_j g_{j,j}\right] q^{-1} + \dots + \left[\sum_{j=N_1}^{N_2} h_j g_{j,j+nb-1}\right] q^{-nb} \tag{2.3.8a}$$

$$S = \left[\sum_{j=N_1}^{N_2} h_j f_{j,0}\right] + \dots + \left[\sum_{j=N_1}^{N_2} h_j f_{j,na}\right] q^{-na} \qquad (2.3.8b)$$

In later sections it will be shown that the polynomials T, R and S simplify for limiting cases of the GPC tuning parameters.

2.3.1 Closed-Loop Transfer Function

When the control law is expressed in the general linear form (shown in Figure 2.2):

$$T\Delta u(t) = Rw(t) - Sy(t)$$

it is easy to combine it with the simplified CARIMA model:

$$Ay(t) = Bq^{-1}u(t) + x(t)$$

where $x(t)=\xi(t)/\Delta$ represents a nonstationary disturbance, to obtain the closed-loop expressions for u(t) and y(t):

$$y(t) = \frac{BRq^{-1}w(t) + T\Delta x(t)}{TA\Delta + q^{-1}BS} \qquad (2.3.9)$$

$$u(t) = \frac{ARw(t) - Sx(t)}{TA\Delta + q^{-1}BS} \qquad (2.3.10)$$

Note that at steady state (q=1):

$$y_{ss} = \left[R(1)/S(1)\right] w_{ss} \qquad (2.3.11)$$

where from eqn (2.3.7):

$$R(1) = \sum_{j=N_1}^{N_2} h_j\,, \qquad S(1) = \sum_{j=N_1}^{N_2} h_j F_j(1)$$

But F_j was obtained from the Diophantine Identity (with, for example, polynomial P weighting):

$$P(q^{-1}) = E_j(q^{-1})A(q^{-1})\Delta + q^{-j}F_j(q^{-1})$$

so that,

$$F_j(1) = P(1) = 1 \quad \text{and} \quad S(1) = R(1).$$

Equation (2.3.11) becomes:

$$y_{ss} = w_{ss} \tag{2.3.12}$$

which indicates that GPC provides offset-free control in the presence of nonstationary step-like disturbances. Note that this is true even if the control law polynomials T, R and S are derived from inexact estimates of A and B obtained from a recursive identification algorithm as long as the closed-loop system is stable.

2.3.2 Extension for Colored Noise

The GPC algorithm was extended in section 2.2.4 to incorporate a C polynomial. Even with colored noise, the control law can be rearranged into the general linear form of eqn. (2.3.4). For this case, the coefficients of T, R and S are determined as follows.

The control law, equation (2.2.26) is:

$$\tilde{u} = (G_r^T G_r + \lambda I)^{-1} G_r^T (w - f)$$

Only the vector f, representing the open-loop predicted output trajectory, is different from the case where C=1:

$$f(t+j) = F_j y(t)/C + \overline{G}_j \Delta u(t-1)/C$$

Note also that the polynomials F_j and $\overline{G}_j$ originate from the Diophantine Identity involving C, eqn. (2.2.16). Under the assumption that elements of the future setpoint sequence supplied to the controller at time t are equal to the current setpoint w(t) (i.e. no preprogrammed setpoints are known) the equation for the current control increment $\Delta u(t)$ is:

$$\Delta u(t) = \sum_{j=N_1}^{N_2} h_j w(t) - \left[\sum_{j=N_1}^{N_2} h_j F_j\right] y(t)/C - \left[\sum_{j=N_1}^{N_2} h_j \overline{G}_j\right] \Delta u(t-1)/C \tag{2.3.13}$$

where $\delta\bar{G}_j$=max(nb-1,nc-1) and the vector $\underline{h}$ again represents the first row of the matrix $(G_r^T G_r + \lambda I)^{-1} G_r^T$.

Multiplying by C and rearranging into the general linear form, $T\Delta u(t) = Rw(t) - Sy(t)$ we have:

$$T = C + q^{-1}\left[\sum_{j=N_1}^{N_2} h_j \bar{G}_j\right], \qquad R = C\left[\sum_{j=N_1}^{N_2} h_j\right] \quad \text{and} \quad S = \sum_{j=N_1}^{N_2} h_j F_j \tag{2.3.14}$$

with polynomial degrees: δT=max(nc,nb), δR=nc, δS=na.

Combining the control law expression with the general CARIMA model:

$$Ay(t) = Bq^{-1}u(t) + Cx(t) \qquad \text{with } x(t) = \xi(t)/\Delta$$

gives the closed-loop transfer function equations:

$$y(t) = \frac{BRq^{-1}w(t) + TC\Delta x(t)}{TA\Delta + q^{-1}BS} \tag{2.3.15}$$

$$u(t) = \frac{ARw(t) - SCx(t)}{TA\Delta + q^{-1}BS} \tag{2.3.16}$$

Again, at steady state,

$$y_{ss} = \left[R(1)/S(1)\right] w_{ss} \tag{2.3.17}$$

where from eqn. (2.3.14):

$$R(1) = C(1)\left[\sum_{j=N_1}^{N_2} h_j\right], \qquad S(1) = \sum_{j=N_1}^{N_2} h_j F_j(1)$$

but from the Diophantine Identity eqn. (2.2.16):

$$F_j(1) = C(1)P(1) = C(1)$$

Therefore, $S(1)=R(1)$ and $y_{ss}=w_{ss}$. Again it is shown that GPC asymptotically eliminates offset in the presence of nonstationary disturbances.

2.4 PID Structure

There has been considerable interest in recent years in obtaining self-tuning PID controllers (Vermeer, 1987; Song et al., 1984). Self-tuning PID controllers have been designed based on the GMV controller (Gawthrop, 1986) as well as the Pole Placement algorithm (Tjokro and Shah, 1985). In this section, the conditions under which the Generalized Predictive Control law is structurally equivalent to a PID controller are outlined. Expressions for the proportional, integral and derivative constants allow the design of PID controllers based on a multistep cost function criterion.

In the previous section, the control law for GPC was rearranged into the form of eqn. (2.3.4):

$$T\Delta u(t) = Rw(t) - Sy(t)$$

where in the most general case with a C polynomial the degrees of T, R and S are:

$$\delta T = \max(nb,nc),\ \delta R = nc \text{ and } \delta S = na$$

A typical discrete PID control law may be written in the velocity form (Stephanopoulos, 1984):

$$\Delta u(t) = K_c\left[1 + \frac{T_s}{\tau_I} + \frac{\tau_D}{T_s}\right]e(t) - K_c\left[1 + 2\frac{\tau_D}{T_s}\right]e(t-1) + K_c\left[\frac{\tau_D}{T_s}\right]e(t-2) \qquad (2.4.1)$$

where K_c is the proportional constant, τ_I is the integral time, τ_D is the derivative time, T_s is the sampling interval and $e(t) = w(t) - y(t)$.

In practice, it is desirable to remove the setpoint from the derivative action to avoid control signal "spikes" or "kicks" due to step setpoint changes. It is also fairly common to remove the setpoint from the proportional term (Isermann, 1981). This leads to the modified PID control law:

$$\Delta u(t) = K_c\left[\frac{T_s}{\tau_I}\right]w(t) - K_c\left[1 + \frac{T_s}{\tau_I} + \frac{\tau_D}{T_s}\right]y(t) + K_c\left[1 + 2\frac{\tau_D}{T_s}\right]y(t-1) - K_c\left[\frac{\tau_D}{T_s}\right]y(t-2) \qquad ...(2.4.2)$$

By comparison with eqn. (2.3.4) it is obvious that the GPC control law has the same structure as this modified PID controller if the polynomials in the

CARIMA model are restricted to:

$$A(q^{-1}) = 1 + a_1 q^{-1} + a_2 q^{-2}$$

$$B(q^{-1}) = b_0$$

$$C(q^{-1}) = 1$$

In other words, in order to get a PID structure, we must assume a 2nd order model (na=2) with no physical delay or open-loop zeros (nb=0) and without colored noise (nc=0). With this assumption eqn. (2.3.4) may be expanded to:

$$t_0 \Delta u(t) = r_0 w(t) - s_0 y(t) - s_1 y(t-1) - s_2 y(t-2) \qquad (2.4.3)$$

where the coefficients are functions of the GPC tuning parameters and model parameters as given by eqn. (2.3.8):

$$t_0 = 1 , \quad r_0 = \sum_{j=N_1}^{N_2} h_j \quad \text{and} \quad s_i = \sum_{j=N_1}^{N_2} h_j f_{j,i} \quad \text{for } i=0,1,2 \qquad (2.4.4)$$

Comparing eqn. (2.4.3) with the modified PID controller eqn. (2.4.2) yields the following four equations:

$$r_0 = K_c T_s / \tau_I \qquad (2.4.5)$$

$$s_0 = K_c \left[1 + T_s/\tau_I + \tau_D/T_s \right] \qquad (2.4.6)$$

$$s_1 = -K_c \left[1 + 2\tau_D/T_s \right] \qquad (2.4.7)$$

$$s_2 = K_c \tau_D / T_s \qquad (2.4.8)$$

Note that only 3 of these equations are independent since from the Diophantine Identity it was shown in section 2.3.1 that S(1)=R(1). In this particular case:

$$r_o = s_o + s_1 + s_2 \qquad (2.4.9)$$

Equations (2.4.6) through (2.4.8) may be solved for K_c, τ_I and τ_D:

$$K_c = -(s_1 + 2s_2) , \quad \tau_I = \frac{-T_s(s_1 + 2s_2)}{s_0 + s_1 + s_2} \quad \text{and} \quad \tau_D = \frac{-s_2 T_s}{(s_1 + 2s_2)} \qquad (2.4.10)$$

2.4.1 PI Structure

The GPC control law assumes the structure of a PI controller if the model is restricted to:

$$A(q^{-1}) = 1 + a_1 q^{-1}$$
$$B(q^{-1}) = b_0$$
$$C(q^{-1}) = 1$$

In other words, to design a PI controller based on GPC a lst order model (na=1) without a time delay (nb=0) or a noise term (nc=0) must be assumed. The constants in the PI control law with no proportional action on setpoint changes:

$$\Delta u(t) = K_c \left[\frac{T_s}{\tau_I} \right] w(t) - K_c \left[1 + \frac{T_s}{\tau_I} \right] y(t) + K_c y(t-1) \qquad (2.4.11)$$

are given by:

$$K_c = -s_1 , \qquad \tau_I = \frac{-s_1 T_s}{s_0 + s_1} \qquad (2.4.12)$$

where s_1 and s_2 are (complex) functions of the tuning and model parameters as follows:

$$s_i = \sum_{j=N_1}^{N_2} k_j f_{j,i} \quad \text{for } i=0,1 \qquad (2.4.13)$$

It is important to point out that no restrictions have been placed on the settings of the basic tuning parameters of GPC (the minimum and maximum output horizons N_1 and N_2, the control horizon NU, and control weighting λ) in order to arrive at the PID/PI structure. In addition, polynomial P weighting may be used without restriction. Therefore, the three tuning strategies proposed in sections 4.2 through 4.4, which allow the user to adjust the closed-loop speed of response by varying a single tuning parameter, apply equally well to the GPC based PID/PI controllers as they do to the general GPC algorithm. The remaining "active" tuning parameter of GPC is easier to specify than PID controller constants and may in fact be automatically adjusted by a supervisory system as demonstrated in Chapter 10. For example, following the recommendations in section 4.2, N_2 may be used to give the desired response with NU=1, N_1=1, P=1 and λ=0.

2.5 Special Cases of GPC

The Generalized Predictive Control algorithm is an extremely flexible approach to control which contains, as special cases for particular settings of its design and tuning parameters, many of the most successful methods employed to date. In Table 2.1, the required settings of the design/tuning parameters are listed which reduce GPC to some of these other strategies. After briefly surveying the algorithms listed, a select few, which are important for the development in subsequent chapters, will be discussed in detail.

Table 2.1 Special Cases of GPC

Algorithm	NU	N_1	N_2	P	λ
Default GPC*	1	1	10	1	0
Pole Placement	N_2-n	n	$\rightarrow\infty$	P	0
Extended Horizon	1 or h	h>d	h>d	1	0
Generalized Minimum Variance	1	d+1	d+1	P	Q
Dynamic Matrix Control†	<N_2	1	>d	1	0 or λ
Mean-Level	1	1	$\rightarrow\infty$	1	0
Exact Model-Following	N_2-d	1	>d	P	0
Detuned Model-Following	<N_2-d	1	>d	P	0 or λ
Deadbeat	n	n	≥2n-1	1	0

Notes: n = max(na+1,nb)
d = integer physical delay not including ZOH (d≥0)
* as suggested by Clarke et al. (1987a)
† step response model

Clarke et al. (1987a) provide general guidelines regarding the choice of the output and control horizons for the basic GPC algorithm. They conclude that the default settings N_1=1, N_2=10 and NU=1 give robust performance for a large number of relatively simple processes. Their self-tuning GPC controller with these settings provides excellent control of a simulated process with changing order and time delay which gives difficulty for self-tuning GMV and PP algorithms (Clarke et al., 1987a).

Pole Placement

GPC is equivalent to a Pole Placement (PP) controller if $N_1=n$, $N_2\to\infty$, $\lambda=0$ and $NU=N_2-n$ where n is the number of states of the system. The closed-loop poles are placed at the zeros of the $P(q^{-1})$ polynomial (Mohtadi, 1986). Although this is of theoretical interest, these settings are not practical as the matrix which must be inverted in the control law is of dimension NUxNU. For implementation purposes, Mohtadi and Clarke (1986) present an alternative method of achieving Pole Placement by performing deadbeat control on the augmented plant, $A\Delta Py(t) = B\Delta Pu(t-1)$ with minor modifications to the standard algorithm.

Extended Horizon

Ydstie et al. (1985) proposed an Extended Horizon Adaptive Controller (EHAC) which predicts the output at a single point in time, h steps ahead, where h is selected greater than the process deadtime. The predicted output is then set equal to the setpoint to calculate the projected control sequence. As with GPC, assumptions must be made regarding future control actions. Two such strategies outlined were to:

a) choose the constant control $u(t) = u(t+1) = \ldots = u(t+h-1)$. This is equivalent to a control horizon of 1 in GPC where $\Delta u(t)$ is freely choosen but $\Delta u(t+1) = \ldots = \Delta u(t+h-1) = 0$.

b) choose the set of controls $u(t)$, $u(t+1)$, ..., $u(t+h-1)$ which minimizes the control effort:

$$\sum_{i=1}^{h} [u(t+h-i)]^2$$

This scheme is realized in GPC with NU = h (Mohtadi, 1986).

The EHAC, as with GPC, is implemented in a receding horizon fashion where u(t) is determined and applied to the plant with the calculations being repeated at each sampling instant. An incremental model is suggested as a means of eliminating offset. Although the algorithm is reported to work well for stable, damped processes, it is not suitable for open-loop unstable or oscillatory plants where the performance is sensitive to the choice of h (Mohtadi, 1986). In particular, large values of h (normally the most conservative) may lead to oscillatory behavior. For these types of processes, more than one point of the predicted output trajectory is usually necessary to achieve satisfactory control.

Generalized Minimum Variance

The Generalized Minimum Variance (GMV) controller of Clarke and Gawthrop (1975, 1979) is one of the most well-known self-tuning schemes. It minimizes the single stage cost:

$$J = E\left\{ [Py(t+k) - Rw(t)]^2 + \lambda[Qu(t)]^2 \right\} \tag{2.5.1}$$

where k = delay including ZOH = d+1

While originally derived based on a CARMA model the method has been extended for a CARIMA model which introduces integral action to eliminate offset in a natural way (Tuffs and Clarke, 1985). The GMV single step optimization may be accomplished in GPC by setting both the minimum and maximum output horizons to d+1 and the control horizon to 1.

The GMV controller is known to be robust against overspecification of the model order but it performs very poorly with variable time delays and is sensitive to nonminimum phase processes. In both of these situations, λ or Q weighting must be selected carefully to avoid instability. This is difficult in practice since the correct range of Q values is not known *a priori*. Even for minimum phase processes, the selection of Q is subject to the difficulties discussed in section 2.2.2.

Controllers based on Weighting Sequence Models

A large number of long-range multi-stage predictive controllers based on weighting-sequence (impulse or step response) models have been proposed in the literature. The methods include Identification/Command (IDCOM), the Predictive Control Algorithm (PCA), and Dynamic Matrix Control (DMC) reviewed in (Clarke and Zhang, 1987) as well as the Multivariable Optimal Constrained Control Algorithm (MOCCA) (Sripada and Fisher, 1985; Lim, 1988).

Consider the CARIMA model with $C(q^{-1}) = A(q^{-1})$:

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + A(q^{-1})x(t) \tag{2.5.2}$$

Expanding the B/A transfer function as an infinite series we obtain the impulse or weighting-sequence model:

$$y(t) = H'(q^{-1})u(t-1) + x(t) \tag{2.5.3}$$

For open-loop stable processes the magnitude of the coefficient of q^{-n}, h_n', decreases as $n \to \infty$. It is therefore possible to make an approximation by truncating H' to the finite polynomial $H(q^{-1})$ of order N for sufficiently large N:

$$y(t) = (h_0 + h_1q^{-1} + \ldots + h_Nq^{-N})u(t-1) + x(t) \tag{2.5.4}$$

The advantage of such a model is that no knowledge is needed of the order of the process. However this type of model cannot be used for open-loop unstable processes and, due to the large number of parameters in the model (typically 20-40), cannot practically be used for adaptive control. Note also that in deriving weighting-sequence controllers the implicit assumption is made that disturbances are governed by $C(q^{-1})=A(q^{-1})$ (unless a separate disturbance model is incorporated into the particular algorithm).

Identification/Command

The IDCOM algorithm (Richalet et al., 1978) is related to GPC with $N_1=1$, $NU=N_2$ and $\lambda=0$ acting on a prefiltered setpoint. However, it uses as iterative approach to calculate explicitly all future controls and must not be allowed to converge! Convergence produces pure model-following control which is undesirable as $H(q^{-1})$ is very likely to have unstable roots, leading to a control signal which grows without bound.

Predictive Control Algorithm

The PCA (Bruijn et al., 1980) is similiar to GPC with $N_1=1$, $NU=N_2$ and nonzero control weighting λ, which trades model-following performance against control effort. Integral action is introduced in an *ad hoc* manner (as with IDCOM) unlike the straightforward method used by GPC.

Dynamic Matrix Control

Dynamic Matrix Control, developed and used by Shell on a number of refinery applications (Cutler and Ramaker, 1980), is based on a step response model of the plant, $S(q^{-1})$ where:

$$\Delta S(q^{-1}) = H(q^{-1}) \tag{2.5.5}$$

DMC originated the key idea of the control horizon (i.e. constraining future control increments to be zero for j>NU) and thus is closest in formulation to GPC.

MOCCA is very similiar to the most recent versions of DMC which accommodate a constrained optimization using linear or quadratic programming (Garcia and Morshedi, 1986). It is important to note that these numerical optimization techniques can also be applied in an adaptive environment using GPC.

2.5.1 Mean-Level Control

One of the important limiting cases of GPC is the steady-state model inverse or "mean-level" controller. A mean-level controller provides a single step in control following a step change in the setpoint which will drive the process output exactly to the new setpoint at steady state. GPC tends to a mean-level controller as $N_2 \rightarrow \infty$ if NU=1, N_1=1, λ=0 and P=1. The derivation here follows along the lines of Mohtadi et al. (1986):

With N_1=1, NU=1 and λ=0 the basic GPC control law from eqn. (2.4.21) becomes:

$$\Delta u(t) = \left[G_r^T G_r\right]^{-1} G_r^T (w - f) \tag{2.5.6}$$

where

$$G_r = \left[g_0 \; g_1 \; \cdots \; g_{N_2-1}\right]^T \tag{2.5.7}$$

$$G_r^T G_r = \sum_{i=0}^{N_2-1} g_i^2 \quad \text{and} \quad f = \left[f(t+1) \; f(t+2) \; \ldots \; f(t+N_2)\right]^T$$

With the elements of the setpoint trajectory equal to the current setpoint

$$w = [w(t) \ldots \; w(t)]^T$$

eqn. (2.5.6), after multiplying by $G_r^T G_r$, becomes:

$$\left[\sum_{i=0}^{N_2-1} g_i^2\right]\Delta u(t) = \sum_{i=0}^{N_2-1} g_i\left[w(t)-f(t+i+1)\right] \tag{2.5.8}$$

We can also write the control law for a maximum output horizon of N_2+1:

$$\left[\sum_{i=0}^{N_2} g_i^2\right]\Delta u(t) = \sum_{i=0}^{N_2} g_i\left[w(t)-f(t+i+1)\right] \tag{2.5.9}$$

Subtracting (2.5.8) from (2.5.9) for large N_2 yields:

$$\Delta u(t) = \left[1/g_{N_2}\right]\left[w(t) - f(t+N_2+1)\right] \tag{2.5.10}$$

Now $f(t+N_2+1)$ is the open-loop prediction of $y(t+N_2+1)$ assuming all present and future control increments are zero:

$$f(t+N_2+1) = \hat{y}_{OL}(t+N_2+1)$$

and g_{N_2} is the (N_2+1)th step response coefficient, which in the limit as $N_2 \to \infty$ for an OL stable process is:

$$\lim_{N_2 \to \infty} g_{N_2} = K_p, \text{ the steady state gain of the process}$$

and,

$$\lim_{N_2 \to \infty} \left\{\hat{y}_{OL}(t+N_2+1)\right\} = \hat{y}_{OL}(\infty), \text{ the predicted OL settled value of the output}$$

Therefore, taking the limit of eqn. (2.5.10) as $N_2 \to \infty$ gives:

$$\Delta u(t) = \left[1/K_p\right]\left[w(t) - \hat{y}_{OL}(\infty)\right] \tag{2.5.11}$$

which is the equation for a mean-level algorithm where the controller is simply the inverse of the steady-state gain of the model.

A mean-level controller places the closed-loop poles in the same locations as the open-loop poles of a stable process. This can be shown by considering the general linear form of the GPC control law. For an output horizon of N_2, the elements of the first (and only) row of $(G_r^T G_r)^{-1} G_r^T$ are:

$$h_j = \frac{g_{j-1}}{\sum_{i=0}^{N_2-1} g_i^2} \qquad j=1,..,N_2 \tag{2.5.12}$$

The polynomials of the control law $T\Delta u(t) = Rw(t) - Sy(t)$ are, from eqn. (2.3.7):

$$T = 1 + q^{-1}\frac{\left[\sum_{j=1}^{N_2} g_{j-1}\overline{G}_j\right]}{\sum_{i=0}^{N_2-1} g_i^2}, \qquad R = \frac{\sum_{j=1}^{N_2} g_{j-1}}{\sum_{i=0}^{N_2-1} g_i^2} \qquad \text{and} \qquad S = \frac{\sum_{j=1}^{N_2} g_{j-1}F_j}{\sum_{i=0}^{N_2-1} g_i^2} \tag{2.5.13}$$

We can write a similar set of equations for an output horizon of N_2+1 and take the difference (for large N_2) to arrive at:

$$T = \left[1/g_{N_2}\right]\left[g_{N_2} + q^{-1}\overline{G}_{N_2+1}\right], \qquad R = 1/g_{N_2} \qquad \text{and} \qquad S = F_{N_2+1}/g_{N_2} \tag{2.5.14}$$

The characteristic polynomial for the closed-loop system (section 2.3.1) is:

$$TA\Delta + q^{-1}BS \tag{2.5.15}$$

Substituting for T and S the characteristic polynomial becomes:

$$\frac{g_{N_2}A\Delta + q^{-1}\left[\overline{G}_{N_2+1}A\Delta + BF_{N_2+1}\right]}{g_{N_2}} \tag{2.5.16}$$

The Diophantine Identity, eqn. (2.1.2), may be written for $j=N_2+1$:

$$1 = E_{N_2+1}A\Delta + q^{-(N_2+1)}F_{N_2+1} \tag{2.5.17}$$

Multiplying through by B and substituting for $G_{N_2+1}=E_{N_2+1}B$ using eqn. (2.1.12) gives:

$$B = \tilde{G}_{N_2+1}A\Delta + q^{-(N_2+1)}\left[\bar{G}_{N_2+1}A\Delta + BF\right] \qquad (2.5.18)$$

So that the characteristic polynomial, eqn. (2.5.16), after rearrangement becomes:

$$\frac{A\Delta\left[g_{N_2} + q^{N_2}\left[B/A\Delta - \tilde{G}_{N_2+1}\right]\right]}{g_{N_2}} \qquad (2.5.19)$$

Since the coefficients of both $B/A\Delta$ and

$$\tilde{G}_{N_2+1} = g_0 + g_1q^{-1} + \dots + g_{N_2}q^{-N_2}$$

are step response coefficients, the characteristic polynomial may be written in the form:

$$\frac{A\Delta\left[g_{N_2} + g_{N_2+1}q^{-1} + \dots\right]}{g_{N_2}} \qquad (2.5.20)$$

Taking the limit as $N_2 \to \infty$ for an open-loop stable process:

$$\lim_{N_2 \to \infty} \left\{\frac{A}{g_{N_2}}\left[g_{N_2} + \left[g_{N_2+1} - g_{N_2}\right]q^{-1} + \left[g_{N_2+2} - g_{N_2+1}\right]q^{-2} + \dots\right]\right\} = A$$

Which verifies that for a mean-level controller, acting on a stable process, the closed-loop poles remain in the same locations as the open-loop poles.

2.5.2 Exact Model-Following

In section 2.2.1, the basic GPC algorithm was extended to include polynomial $P(q^{-1})$ weighting. The use of an auxiliary output allows GPC to achieve model-following, where $M(q^{-1})=1/P(q^{-1})$ is the user-specified closed-loop reference model. For **exact** model-following the controller attempts to cancel the process zeros and therefore this approach is unsuitable for processes which are nonminimum phase (NMP). Åström et al.

(1984) demonstrate that it is more a rule than an exception for sampled-data systems to have unstable zeros. For example, continuous time systems with pole (over zero) excess larger than two will always give discrete systems with zeros outside the unit circle for sufficiently small sampling periods. Fractional time delays, unavoidable in practice, also frequently lead to NMP discrete systems. Since process zeros must obviously not be cancelled we are forced to "detune" the model-following properties of GPC for practical applications. It is therefore important to have a knowledge of the tuning parameter settings which lead to exact and detuned model-following for GPC.

Exact model-following is achieved for the Generalized Minimum Variance (GMV) controller with Q=0 (Gawthrop, 1977). Therefore, it is expected that GPC would exhibit the same behavior for NU=1, $N_1=N_2=d+1$ and $\lambda=0$. Note that we can relax the specification on the minimum output horizon to $N_1 \leq d+1$ since outputs prior to y(t+d+1) cannot be influenced by $\Delta u(t)$. The following development demonstrates that exact model-following is obtained for these special settings of the GPC tuning parameters.

A delay of d units implies that the first d terms of both the B and $G_{d+1}(=E_{d+1}B)$ polynomials will be zero:

$$\begin{aligned} G_{d+1} &= \tilde{G}_{d+1} + q^{-(d+1)}\overline{G}_{d+1} \\ &= g_d q^{-d} + g_{d+1,d+1}q^{-(d+1)} + \dots + g_{d+1,d+nb}q^{-(d+nb)} \end{aligned} \qquad (2.5.21)$$

And (taking $N_1=1$ for simplicity):

$$\left[G_r^T G_r\right]^{-1} G_r^T = \left[0 \ 0 \ \dots \ 0 \ g_d\right] / g_d^2 \qquad (2.5.22)$$

So the only nonzero element of the vector **h**, introduced in section 2.3, is:

$$h_{d+1} = 1/g_d \qquad (2.5.23)$$

The polynomials in the general linear form of the control law are then:

$$T = 1 + q^{-1}\overline{G}_{d+1}/g_d = q^d G_{d+1}/g_d \ , \quad R = 1/g_d \quad \text{and} \quad S = F_{d+1}/g_d \qquad (2.5.24)$$

Recall that the closed-loop transfer function is:

$$y(t) = \frac{BRq^{-1}w(t) + T\Delta x(t)}{TA\Delta + q^{-1}BS}$$

Substituting for T, R and S and rearranging we get:

$$y(t) = \frac{Bq^{-(d+1)}w(t) + G_{d+1}\Delta x(t)}{G_{d+1}A\Delta + q^{-(d+1)}BF_{d+1}}, \quad (2.5.25)$$

However G_{d+1} and F_{d+1} originate from the Diophantine Identity (eqn. (2.2.4)) which written for j=d+1 and multiplied by B is:

$$BP = G_{d+1}A\Delta + q^{-(d+1)}BF_{d+1} \quad (2.5.26)$$

Thus,

$$y(t) = \frac{q^{-(d+1)}w(t) + E_{d+1}\Delta x(t)}{P} \quad (2.5.27)$$

Which verifies that exact model-following is achieved after the inevitable delay of d+1 samples. Substituting for T, R and S, the control law ($T\Delta u(t) = Rw(t) - Sy(t)$) may be written as:

$$\Delta u(t) = \frac{w(t) - F_{d+1}y(t)}{q^{d}E_{d+1}B} \quad (2.5.28)$$

It is apparent from this equation that the controller cancels the process zeros.

If $P(q^{-1})$ is specified to be unity, exact model-following provides minimum variance, (MV) or "d+1 step ahead" control.

The reader may suspect that NU=1 and N_2=d+1 are not the only GPC settings which lead to exact model-following. Intuitively, if N_2 is increased beyond d+1 as long as NU is incremented by the same amount the controller retains sufficient degrees of freedom so that the cost function minimization still results in exact model-following. Simulation runs have shown that setting N_2>d and NU=N_2-d does give exact model-following, though it is difficult to prove this for the most general case.

Mohtadi (1986) suggests that NU=N_2 results in exact model-following.

While intuitively the larger value of NU should not effect the cost minimization, the singularity of the $G_r^TG_r$ matrix makes it impossible to realize the control law for this case. (If $NU>N_2-d$ the last column of G_r is composed of zeros and $G_r^TG_r$ is singular.)

In summary, $N_2>d$, $NU=N_2-d$, $N_1=1$ and $\lambda=0$ leads to exact model-following behavior for GPC. For practical applications model-following must be "detuned" by either increasing the maximum output horizon relative to the control horizon (i.e. $N_2-d>NU$) or by adding control weighting ($\lambda>0$).

2.5.3 Deadbeat Control

The final special case of GPC which will be considered is that of deadbeat control. Mohtadi and Clarke (1986) show that GPC is equivalent to a stable state deadbeat controller if the system is observable and controllable and $NU=n$, $N_1=n$, $N_2\geq 2n-1$, $P=1$ and $\lambda=0$. The number of states of the system, $n = \max(na+1,nb)$. State deadbeat control places all of the closed-loop poles at the origin so that the closed-loop transfer function becomes:

$$y(t) = \frac{Bq^{-1}}{B(1)}w(t) + T\Delta x(t) \tag{2.5.29}$$

For a step setpoint change, the controlled variable reaches the new setpoint in at most nb+1 steps.

Experience indicates that for deadbeat control the specifications for NU, N_1 and N_2 may be relaxed to:

$$NU \geq na+1 \;, \qquad N_1 \geq nb+1 \qquad \text{and} \qquad N_2 \geq NU+N_1^{\circ}-1$$

and GPC still delivers deadbeat control.

2.6 Summary

The Generalized Predictive Control approach incorporates a significant number of design and tuning parameters: N_1, N_2, NU, λ, $P(q^{-1})$. Rearranging the control law into a general linear form, has made it easier to show that for specific settings of these parameters, GPC reduces to many well-known control algorithms (see Table 2.1). It is important to recognize that the increased complexity of GPC is warranted by the fact that other combinations of parameter settings allow GPC to overcome the limitations of these earlier techniques and yield better overall performance. In Chapter 4, simple strategies for commissioning GPC are devised which reduce the number of active tuning parameters while still retaining the performance capabilities.

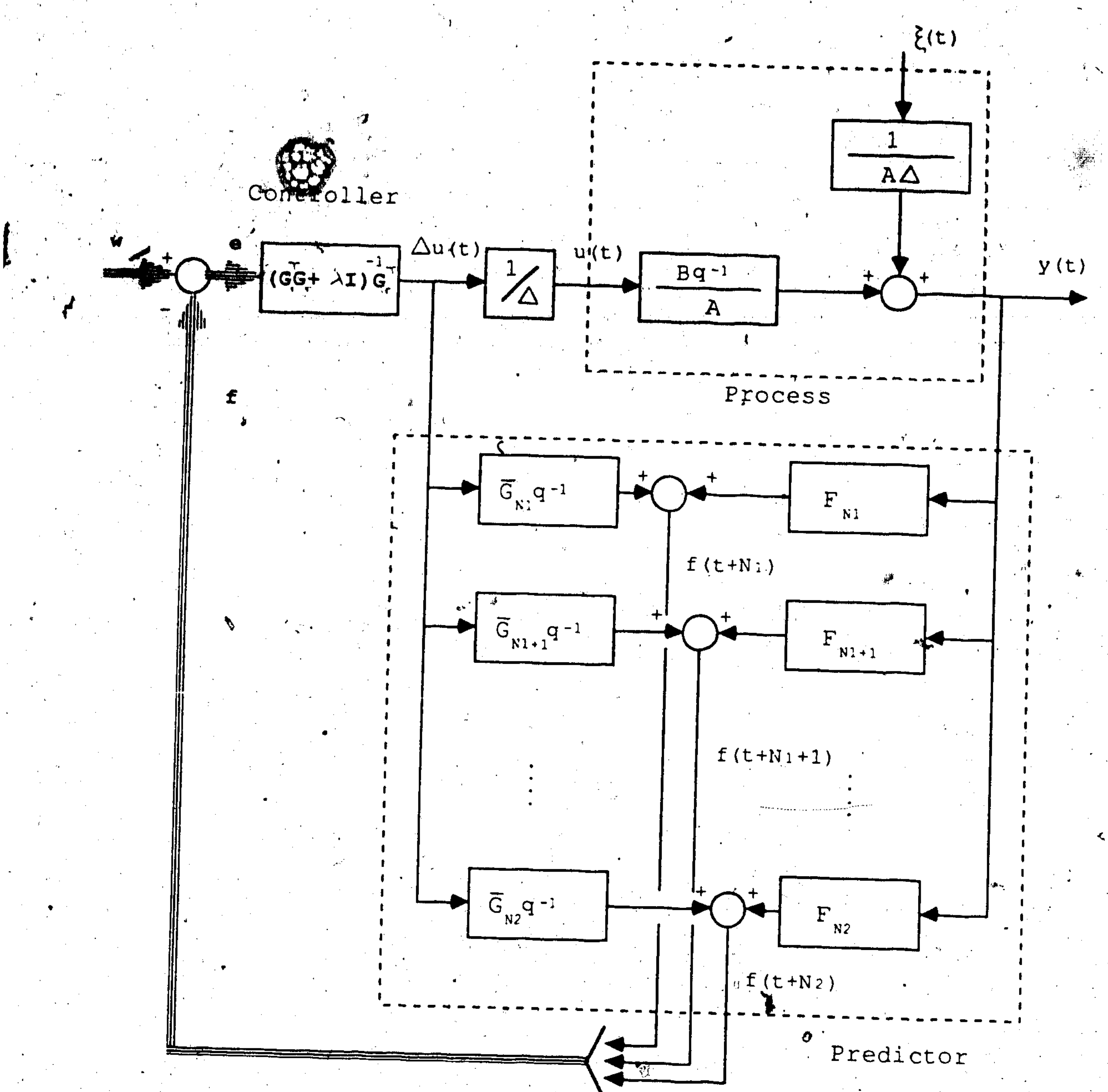


Figure 2.1 Block Diagram Representation of Basic GPC Algorithm

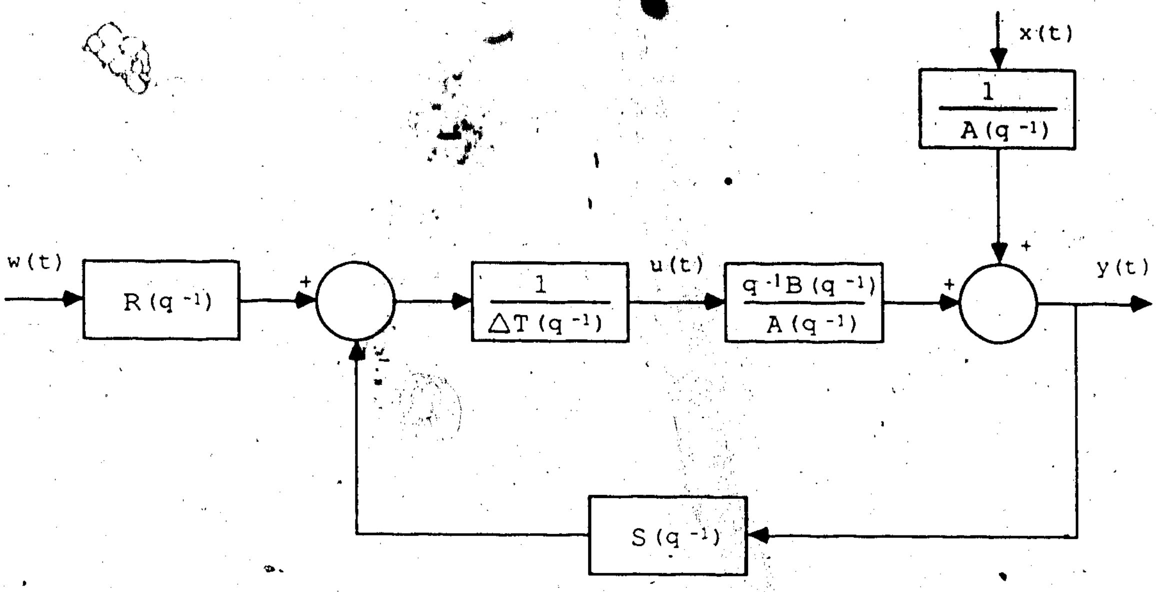


Figure 2.2 Block Diagram For General Linear Form Of GPC

3. POLE PLACEMENT CONTROL

Many authors have attempted to remove the barrier between the "optimal" predictive strategies involving minimization of a cost function and the "suboptimal" pole assignment technique. Allidina and Hughes (1980) present an algorithm for adjusting the P and Q weighting polynomials in the self-tuning GMV controller to locate the closed-loop poles at prespecified locations thus demonstrating that pole locations can be uniquely translated into a cost function performance criteria. McDermott and Mellichamp (1984) follow the same approach except the closed-loop pole position(s) are not user-specified but rather are obtained periodically by minimizing a multistep objective function involving the predicted error for a simulated setpoint change. A more elegant method of achieving Pole Placement within a long-range multistep predictive control environment has been described by Lelic and Zarrop (1987) and Lelic and Wellstead (1987). The only significant advantage of this Generalized Pole Placement (GPP) formulation over standard PP is the ability to use known future (preprogrammed) setpoints. The authors were dissatisfied with their account of how to select the various horizons, which, in trials without preprogrammed setpoints had little effect upon the closed-loop response. Intuitively this makes sense; once the pole locations are specified the system response has largely been determined. As mentioned in the previous chapter, Mohtadi (1986) has shown that GPC reduces to a PP controller for particular choices of the control and output horizons. This is accomplished without having to adjust the P and Q polynomials in response to process model changes. In robotics or batch control applications these predictive Pole Placement controllers may yield improved performance over that of a standard PP algorithm (McDermott and Mellichamp, 1984). However, in the majority of chemical process control applications, the increase in complexity does not warrant their use. For this reason the Pole Placement work carried out in this thesis is based on the straight forward incremental formulation of Tuffs (1984).

The derivation of the incremental PP controller presented in section 3.1 includes the important extension for colored noise. Next, the formation of the Sylvester equation and numerical problems associated with its solution are considered (section 3.2). Finally, design equations are given for two different Pole Placement based self-tuning PID controllers (section 3.3).

3.1 Derivation of the Basic Algorithm

The basic idea of Pole Placement control is to select the parameters of a general linear control equation in such a way as to place the closed-loop poles at prespecified positions. The derivation which follows is based on the CARIMA model which leads to a controller with inherent integral action. Consider the control law:

$$G\Delta u(t) = Hw(t) - Fy(t) \tag{3.1.1}$$

which when applied to the simplified CARIMA model ($C(q^{-1}) = 1$):

$$Ay(t) = Bq^{-1}u(t) + \xi(t)/\Delta \tag{3.1.2}$$

yields the closed-loop equation:

$$y(t) = \frac{BHq^{-1}w(t) + G\xi(t)}{GA\Delta + q^{-1}BF} \tag{3.1.3}$$

The closed-loop poles are located at the roots of a polynomial $P(q^{-1})$ if we solve for the controller parameters F and G from the Diophantine equation:

$$GA\Delta + q^{-1}BF = P \tag{3.1.4}$$

so that:

$$y(t) = \frac{BHq^{-1}w(t) + G\xi(t)}{P} \tag{3.1.5}$$

To obtain offset-free behavior we must have:

$$B(1)H(1)/P(1) = 1 \tag{3.1.6}$$

The simplest choice for $H(q^{-1})$ which satisfies this criterion is the constant:

$$H = P(1)/B(1) \tag{3.1.7}$$

which results in the closed-loop expression:

$$y(t) = \frac{P(1)Bq^{-1}w(t)}{PB(1)} + \frac{G\xi(t)}{P} \tag{3.1.8}$$

Note that the open-loop zeros are retained in the closed-loop.

3.1.1 Extension for Colored Noise

The previous derivation may be readily extended to the general CARIMA model with $C(q^{-1}) \neq 1$ (Tuffs, 1984):

$$Ay(t) = Bq^{-1}u(t) + C\xi(t)/\Delta \tag{3.1.9}$$

Applying the same form of control law as eqn. (3.1.1):

$$G\Delta u(t) = Hw(t) - Fy(t)$$

the closed-loop expression may be written:

$$y(t) = \frac{BHq^{-1}w(t) + CG\xi(t)}{GA\Delta + q^{-1}BF} \tag{3.1.10}$$

If the controller polynomials F and G are obtained by solving the extended Diophantine equation:

$$GA\Delta + q^{-1}BF = PC \tag{3.1.11}$$

the closed-loop system is

$$y(t) = \frac{BHq^{-1}w(t)}{PC} + \frac{G\xi(t)}{P} \tag{3.1.12}$$

Where for no offset $H(q^{-1})$ must be selected to satisfy:

$$B(1)H(1)/(P(1)C(1)) = 1 \tag{3.1.13}$$

A good choice for $H(q^{-1})$ is that which eliminates the noise structure polynomial $C(q^{-1})$ from the servo response:

$$H = P(1)C/B(1) \tag{3.1.14}$$

The closed-loop equation is then:

$$y(t) = \frac{P(1)Bq^{-1}w(t)}{PB(1)} + \frac{G\xi(t)}{P} \quad (3.1.15)$$

Remark: As for the Generalized Predictive Controller, the C polynomial which has been incorporated into the PP control law may be either a design polynomial, C_c, or an estimate of the "true" noise model, $\hat{C}$, obtained by recursive parameter identification (e.g. using ELS). The former case, where C is selected as a surrogate noise model, was first considered by Wellstead and Sanoff (1981).

3.1.2 Closed-loop Properties

In the previous section we obtained the closed-loop expression for the controlled variable assuming that the C polynomial in the Diophantine equation was identical to that in the CARIMA model. To be more general, the appropriate equation for the case where the "true" noise polynomial, C_0, is different from the C polynomial used in the Diophantine equation is:

$$y(t) = \frac{P(1)Bq^{-1}w(t)}{PB(1)} + \frac{GC_0\Delta x(t)}{PC}$$

where $x(t)=\xi(t)/\Delta$ represents a nonstationary disturbance. The corresponding expression for the manipulated variable is:

$$u(t) = \frac{P(1)Aw(t)}{PB(1)} - \frac{FC_0x(t)}{PC} \quad (3.1.16)$$

Since no open-loop zeros are cancelled by the controller, nonmimimum phase processes are easily handled by the PP algorithm.

It is important to verify that integral action ensures offset-free behavior in spite of incorrect model parameter estimates. The open-loop control law for this case may be written:

$$u(t) = \left[1/\hat{G}\Delta\right]\left[\hat{H}w(t) - \hat{F}y(t)\right] \quad (3.1.17)$$

where $\hat{G}$, $\hat{F}$ and $\hat{H}$ are obtained from the polynomials $\hat{A}$ and $\hat{B}$ estimated, for example, by recursive least squares. From the Diophantine equation at steady state:

$$\hat{F}(1) = P(1)C(1)/\hat{B}(1) \tag{3.1.18}$$

Therefore $\hat{F}$ may be split into two components:

$$\hat{F} = P(1)C(1)/\hat{B}(1) + \Delta\hat{F}' \tag{3.1.19}$$

Recall also that $\hat{H} = P(1)C/\hat{B}(1)$ so that eqn. (3.1.17) may be written:

$$u(t) = \frac{1}{\hat{G}\Delta}\left[\frac{P(1)C}{\hat{B}(1)}w(t) - \frac{P(1)C(1)}{\hat{B}(1)}y(t) - \hat{F}'\Delta y(t)\right] \tag{3.1.20}$$

At steady state, an error term $(w_{ss}-y_{ss})$ independent of the parameter accuracy can be isolated, and the denominator has zero gain. This confirms that integral action is obtained inspite of any model-plant mismatch.

3.1.3 Feedforward Compensation

Feedforward signals, including measured disturbances and control signals from other loops, may be accomodated by using the modified control equation:

$$G\Delta u(t) = Hw(t) - Fy(t) - S\Delta v(t) \tag{3.1.21}$$

which when applied to the full CARIMA model:

$$Ay(t) = Bq^{-1}u(t) + C\xi(t)/\Delta + Dq^{-1}v(t) \tag{3.1.22}$$

produces the closed-loop system,

$$y(t) = \frac{BHq^{-1}w(t) + (DG-BS)q^{-1}\Delta v(t) + CG\xi(t)}{GA\Delta + q^{-1}BF} \tag{3.1.23}$$

With the Diophantine equation defined as in eqn. (3.1.11) and with H selected according to eqn. (3.1.14) this becomes:

$$y(t) = \frac{P(1)Bq^{-1}w(t)}{PB(1)} + \frac{(DG-BS)q^{-1}\Delta v(t)}{PC} + \frac{G\xi(t)}{P} \tag{3.1.24}$$

For perfect compensation we must have

$$DG - BS = 0 \tag{3.1.25}$$

A polynomial solution for S does not exist unless D is proportional to B. For the general case, S must be a transfer function:

$$S = DG/B \tag{3.1.26}$$

Obviously this type of feedforward compensation is unsuitable for nonminimum phase processes. Alternatively we may be satisified with the "steady state" feedforward controller:

$$S = D(1)G/B(1) \tag{3.1.27}$$

which is not restricted to minimum phase processes and is therefore more suited for practical applications.

3.2 Solution of the Diophantine Equation

The polynomial Diophantine Equation plays an important role in the application of a Pole Placement controller. For the case of the general CARIMA model, the Diophantine Equation which must be solved was given by eqn. (3.1.11):

$$GA\Delta + q^{-1}BF = PC$$

A minimal degree solution for F and G is obtained by choosing the polynomial degrees:

$$ng = nb \ , \quad nf = \max(na, np+nc-nb-1) \tag{3.2.1}$$

where,

$$F = f_0 + f_1 q^{-1} + \dots + f_{nf} q^{-nf}$$

$$G = g_0 + g_1 q^{-1} + \dots + g_{ng} q^{-ng}$$

Letting $A\Delta = \overline{A}$ and $PC = \overline{P}$ the Diophantine Equation may be written in the form:

$$G\overline{A} + q^{-1}BF = \overline{P} \tag{3.2.2}$$

with,

$$\overline{A} = 1 + \overline{a}_1 q^{-1} + \dots + \overline{a}_{na+1} q^{-(na+1)} \quad \text{and,}$$

$$\overline{P} = \overline{p}_0 + \overline{p}_1 q^{-1} + \dots + \overline{p}_{np+nc} q^{-(np+nc)}$$

Equating coefficients of like powers of q^{-1} we obtain nf+ng+2 simultaneous equations which may be written in matrix form:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ \bar{a}_1 & \ddots & b_0 & \ddots \\ \vdots & & \vdots & 0 \\ \bar{a}_{na+1} & 1 & & b_0 \\ & \bar{a}_1 & b_{nb} & \vdots \\ & \vdots & & \\ 0 & \bar{a}_{na+1} & 0 & b_{nb} \end{bmatrix} \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{ng} \\ f_0 \\ f_1 \\ \vdots \\ f_{nf} \end{bmatrix} = \begin{bmatrix} \bar{p}_0 \\ \bar{p}_1 \\ \vdots \\ \bar{p}_{np+nc} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\text{or } S_v x = t \qquad (3.2.3)$$

where S_v is known as the Sylvester Matrix.

It is well known that a solution to this equation exists (i.e. S_v^{-1} exists) only if the polynomials $\bar{A}$ and B do not contain a common factor unless this common factor is also a factor of $\bar{P}$. Since B is normally overparameterized to incorporate the delay, the degree of A must be selected larger than the actual order of the process. Specification of the exact order to avoid the common factor problem is difficult in practice and thus represents a short-coming of the Pole Placement method.

Note also that for plants with large dead-time, the dimension of the Sylvester Matrix is high; the computational effort involved in determining the controller parameters may be considerable.

3.2.1 Numerical Problems

While an (identical) common factor in the polynomials $\bar{A}$ and B manifests itself as a singularity of the Sylvester Matrix, roots of $\bar{A}$ and B which lie close to each other produce an ill-conditioned system of equations. Small perturbations in the model parameters will then cause large variations in the controller polynomials F and G.

The only complete solution to this problem is to isolate the common factors in $\bar{A}$ and B and remove them from the polynomialss before constructing the simultaneous equations (Kučera, 1979). However this method relies heavily on having identical cancelling factors which will almost never be the case in practice.

One set of techniques for resolving the difficulty relies on approximating the solution of the Sylvester equation in a numerically robust way. For example, a routine based on the Householder orthogonality transform examines the S_v matrix to determine its "psuedo-rank" from a user supplied tolerance, and if a rank deficiency is detected it calculates a "minimum-norm" solution (Lawson and Hanson, 1974). The approximate solution implies that the closed-loop poles will not coincide exactly with the user-specified set. This is turn may lead to undesirable control behavior.

Lozano-Leal and Goodwin (1985) discuss another strategy whereby the parameter estimates from a recursive least squares algorithm with data normalization are modified to ensure that the process model polynomials are always relatively prime. This technique requires a search for a vector, α, which when multiplied by the covariance matrix and added to the parameter estimates, ensures the nonsingularity of the Sylvester Matrix. It is used to prove global stability without a persistency of excitation requirement but is not demonstrated to be feasible for practical applications.

A non-theoretical but simple potential solution to the numerical ill-conditioning problem is based on the assumption that the true system polynomials are relatively prime. Under these very common circumstances, the controller polynomials F and G are updated only when the condition number of the Sylvester Matrix is below a user specified bound. The last numerically robust solution is held until the parameter estimates migrate from transient locations for which common factors in $\bar{A}$ and B exist (Tzouanas and Shah, 1985).

3.3 PID Structure

As was done for Generalized Predictive Control, it is useful to outline the conditions under which the Pole Placement algorithm is structurally equivalent to a PID controller. Equations relating the proportional gain, integral time and derivative time to model parameters and the user selected characteristic polynomial allow the design of self-tuning PID controllers based on PP.

The polynomials in the control law:

$$G\Delta u(t) = Hw(t) - Fy(t)$$

based on the general CARIMA model, are of degree:

$$\delta G = nb \qquad \delta F = na \qquad \delta H = nc$$

as long as the characteristic polynomial is choosen such that:

$$\delta P \le na+nb-nc+1$$

Clearly for a PID structure we must have na=2 and nb=0:

$$A(q^{-1}) = 1 + a_1 q^{-1} + a_2 q^{-2} \qquad B(q^{-1}) = b_0 \qquad (3.3.1)$$

in which case the limit on the degree of P becomes:

$$\delta P \le 3-nc$$

For any PID control structure, there is the requirement that H(1)=F(1). Recall the Diophantine Equation (3.1.11):

$$GA\Delta + q^{-1}BF = PC$$

so that

$$F(1) = P(1)C(1)/B(1) \qquad (3.3.2)$$

Normally H is selected as:

$$H = P(1)C/B(1) \qquad (3.3.3)$$

so the requirement is naturally met.

Different variations of the PID controller, where the setpoint is removed from the derivative or proportional and derivative terms, impose additional restrictions on the relationship between H and F. This is easiest to see from the following (velocity) discrete PID equations. The standard PID controller was given in Chapter 2 as:

$$\Delta u(t) = K_c\left[1 + \frac{T_s}{\tau_I} + \frac{\tau_D}{T_s}\right]e(t) - K_c\left[1 + \frac{2\tau_D}{T_s}\right]e(t-1) + K_c\left[\frac{\tau_D}{T_s}\right]e(t-2) \qquad (3.3.4)$$

The "PI on SP" form, where the setpoint is removed from the derivative term is:

$$\Delta u(t) = K_c\left[1 + \frac{T_s}{\tau_I}\right]w(t) - K_c w(t-1) - K_c\left[1 + \frac{T_s}{\tau_I} + \frac{\tau_D}{T_s}\right]y(t) + K_c\left[1 + \frac{2\tau_D}{T_s}\right]y(t-1) - K_c\left[\frac{\tau_D}{T_s}\right]y(t-2) \qquad (3.3.5)$$

while the "I on SP" version with only integral action on setpoint changes was also given in Chapter 2 as:

$$\Delta u(t) = K_c\left[\frac{T_s}{\tau_I}\right]w(t) - K_c\left[1 + \frac{T_s}{\tau_I} + \frac{\tau_D}{T_s}\right]y(t) + K_c\left[1 + \frac{2\tau_D}{T_s}\right]y(t-1) - K_c\left[\frac{\tau_D}{T_s}\right]y(t-2) \qquad (3.3.6)$$

For the "PID on SP", "PI on SP" and "I on SP" controllers,

$$H = F, \qquad (3.3.7a)$$

$$h_0 = f_0 - f_2,\ h_1 = f_1 + 2f_2,\ h_2 = 0 \text{ and} \qquad (3.3.7b)$$

$$H = F(1) = f_0 + f_1 + f_2, \qquad (3.3.7c)$$

respectively.

With H given by eqn. (3.3.3) and F obtained from the solution of the Diophantine Equation (3.1.11) it is evident that the above relationships will not hold for a general C polynomial. However, two special cases, where the form of C is restricted, will be considered next. (The implications of these restrictions on the robustness and regulatory properties of the controller will be considered at length in subsequent chapters.)

3.3.1 "I on SP" Controller

If the condition is imposed that C=1, then

$$H = P(1)/B(1) = F(1)$$

and the PP algorithm reduces to a "I on SP" PID controller. The user is free to choose the closed-loop poles via $P(q^{-1})$ as long as $\delta P \leq 3$. With a 2nd order A polynomial, zeroth order B polynomial, unity C polynomial and $P = 1 + p_1 q^{-1} + p_2 q^{-2} + p_3 q^{-3}$, the solution of the Diophantine Equation (3.1.11) is:

$$G = g_0 = 1 \quad \text{and} \quad F = q(P - A\Delta)/B \tag{3.3.8}$$

so that the coefficients of F are:

$$f_0 = \frac{1+p_1-a_1}{b_0}, \quad f_1 = \frac{p_2+a_1-a_2}{b_0} \quad \text{and} \quad f_2 = \frac{p_3+a_2}{b_0} \tag{3.3.9}$$

Comparing coefficients of the "I on SP" PID controller with the PP control law yields three equations:

$$f_0 = K_c\left[1 + \frac{T_s}{\tau_I} + \frac{\tau_D}{T_s}\right], \quad f_1 = -K_c\left[1 + \frac{2\tau_D}{T_s}\right] \quad \text{and} \quad f_2 = K_c\left[\frac{\tau_D}{T_s}\right] \tag{3.3.10}$$

which may be solved for the PID controller constants:

$$K_c = -(f_1+2f_2), \quad \tau_I = \frac{-T_s(f_1+2f_2)}{f_0+f_1+f_2} \quad \text{and} \quad \tau_D = \frac{-T_s f_2}{f_1+2f_2} \tag{3.3.11}$$

Substituting for f_0, f_1, and f_2 using (3.3.9) we arrive at expressions for K_c, τ_I and τ_D as functions of A, B and P:

$$K_c = \frac{-(a_1+a_2+p_2+2p_3)}{b_0}, \quad \tau_I = \frac{-T_s(a_1+a_2+p_2+2p_3)}{1+p_1+p_2+p_3}, \quad \text{and} \quad \tau_D = \frac{-T_s(a_2+p_3)}{(a_1+a_2+p_2+2p_3)}$$

...(3.3.12)

3.3.1.1 PI Structure

Following a similar line of reasoning as for the previous section, the conclusion is reached that the PP algorithm will reduce to a "I on SP" PI controller:

$$\Delta u(t) = K_c\left[\frac{T_s}{\tau_I}\right]w(t) - K_c\left[1 + \frac{T_s}{\tau_I}\right]y(t) + K_c y(t-1) \tag{3.3.13}$$

if na=1, nb=0, and nc=0:

$$A(q^{-1}) = 1 + a_1 q^{-1}$$

$$B(q^{-1}) = b_0$$

$$C(q^{-1}) = 1$$

The PI controller constants in terms of A, B and the user specified characteristic polynomial, $P(q^{-1}) = 1+p_1q^{-1}+p_2q^{-2}$, are:

$$K_c = \frac{-(a_1+p_2)}{b_0} \quad \text{and} \quad \tau_I = \frac{-T_s(a_1+p_2)}{1+p_1+p_2} \tag{3.3.14}$$

3.3.2 "PID on SP" Controller

If it is assumed that C=A (for an open-loop stable process), the desired characteristic polynomial should be selected such that [illegible] With $P=1+p_1q^{-1}$, $A=1+a_1q^{-1}+a_2q^{-2}$ and $B=b_0$ the solution of the Diophantine Equation (3.1.11) is:

$$[illegible] = 1, \quad F = [illegible]/B = (1+p_1)A/b_0 \tag{3.3.15}$$

and since

$$H = P(1)C/B \; [illegible] \; A/b_0 = F \tag{3.3.16}$$

the PP algorithm reduces to a [illegible] on SP" PID controller. Equations (3.3.10) and (3.3.11) still hold and with the coefficients of F substituted from (3.3.15), the PID controller gains are given by:

$$K_c = \frac{-(1+p_1)(a_1+2a_2)}{b_0}, \quad \tau_I = \frac{-T_s(a_1+2a_2)}{1+a_1+a_2}, \quad \tau_D = \frac{-T_s a_2}{a_1+2a_2} \tag{3.3.17}$$

3.3.2.1 PI Structure

If the degree of A is limited to 1 (rather than 2):

$$A(q^{-1}) = 1 + a_1q^{-1}$$

$$B(q^{-1}) = b_0$$

Then with $C(q^{-1}) = A(q^{-1})$, a "PI on SP" PI controller can be derived from the PP algorithm. The controller gains in terms of A, B and the user specified characteristic polynomial, $P(q^{-1}) = 1+ p_1q^{-1}$, are:

$$K_c = \frac{-(1+p_1)a_1}{b_0} \quad \text{and} \quad \tau_I = \frac{-T_s a_1}{1+a_1} \tag{3.3.18}$$

3.4 Summary

The derivation of a conceptually-simple incremental Pole Placement controller (Tuffs, 1984) has been presented. Recommendations for selecting the desired characteristic polynomial and noise model polynomial will be given in subsequent chapters. By limiting the structure of the process and disturbance models, the PP algorithm has been shown to be equivalent to PID (and PI) controllers. Pole Placement control may also be obtained as a special case of the Generalized Predictive Controller. While conceptually more complex, such a predictive Pole Placement controller is capable of incorporating preprogrammed setpoints for robotics or batch control applications.

4. PERFORMANCE AND TUNING OF GENERALIZED PREDICTIVE CONTROL

In Chapter 2, the basic GPC algorithm was derived and extended to incorporate weighting polynomials (P, etc.) and colored noise (C). Rearrangement of the control law into a general linear form allowed the determination of a closed-loop transfer function and assisted in demonstrating that GPC includes many well-known control techniques as special cases. The flexibility of GPC arises from the substantial number of design parameters built into the approach. The purpose of this chapter is to provide insight into the effect of these parameters on the performance of the controller in order to provide guidelines as to their selection for chemical process control.

The Generalized Predictive Control law is based on the minimization of a cost function consisting of future predicted errors and weighted control increments. With control weighting enabled, the optimal sequence of projected controls represents a trade-off between output performance (i.e. the servo and regulatory response of the controlled variable alone) and control effort. Unless special precautions are taken, the characteristics of the closed-loop output response will vary with changes in the process even with perfect modelling. Since for most applications this variation is unacceptable, it is important to identify the conditions under which GPC will maintain a consistent output performance. This is the topic of section 4.1.

As mentioned the GPC algorithm incorporates a significant number of design or tuning parameters. Even in the case where we do not specifically take into account the nature of the disturbances (i.e. ignoring C), the closed-loop behavior depends upon the choice of the following five controller parameters:

a) minimum output horizon, N_1

b) maximum output horizon, N_2

c) control horizon, NU

d) control weighting, λ

e) output weighting, $P(q^{-1})$.

Although, the gross effect of each individual parameter has been considered elsewhere (Clarke et al., 1987a; Lambert,E.P., 1987; Lambert,M., 1987), from a user standpoint, the specification of these parameters is still difficult, since they are highly interacting. Strategies are needed for reducing the number of "active" tuning parameters (by assigning proper default values to the remainder during commissioning) while retaining the ability of GPC to handle a wide range of processes from simple low order, stable plants, to complex, nonminimum phase, unstable plants with variable time delays. Three tuning strategies (based on the Output Horizon, Lambda Weighting and Detuned Model-Following configurations of GPC) which involve adjusting only a single tuning parameter are described and analyzed in sections 4.2 to 4.4.

The addition of two more potential tuning parameters:

f) the controller design polynomial, C_c

g) the setpoint prefilter, F_{sp}

makes it possible to significantly improve performance in noisy environments and meet independent servo and regulatory objectives. Sections 4.5 and 4.6 are concerned with different interpretations of C and methods of achieving control with "two degrees of freedom" (Stephanopoulos and Huang, 1986).

4.1 Maintenance of Output Performance

The Generalized Predictive Controller with $P(q^{-1})$ and λ weighting minimizes the cost function:

$$J = E\left\{ \sum_{j=N_1}^{N_2} [Py(t+j) - w(t+j)]^2 + \sum_{j=1}^{NU} \lambda[\Delta u(t+j-1)]^2 \right\} \qquad (4.1.1)$$

As the process input/output behavior changes, the tradeoff between the output performance and control activity, represented by the first and second terms, shifts in order to minimize J. Therefore, as long as fixed control weighting is employed, the user should not expect the response of the output to remain invariant, even when an exact model is available. On the other hand, if control weighting is not used (or if it adjusted when the process changes as described in section 4.3.2), the output performance remains unchanged, at least for some types of process changes.

4.1.1 Process Gain Changes

It is straightforward to show that the output performance is maintained in spite of gain changes in the process if no control weighting is used (λ=0) and there are no modelling errors.

Process gain changes are reflected entirely by the B polynomial in the CARIMA model, the coefficients of which are proportional to the process gain, K_p (see, for example, Neuman and Baradello, 1979). Factoring out the gain we have

$$B = K_p B' \tag{4.1.2}$$

where the prime denotes a quantity corresponding to the case where K_p=1. As a result, the E_j and F_j polynomials obtained from the Diophantine Identity:

$$P = E_j A\Delta + q^{-j}F_j$$

are independent of K_p:

$$F_j = F_j' \qquad G_j = E_j B = K_p E_j' B' = K_p G_j' \tag{4.1.3}$$

Therefore, the elements of the matrix G_r are proportional to K_p:

$$G_r = K_p G_r'$$

so with λ=0,

$$\left[G_r^T G_r\right]^{-1} G_r^T = \left[G_r^{T'} G_r'\right]^{-1} G_r^{T'}/K_p \tag{4.1.4}$$

The elements of the vector **h** are given by:

$$h_j = h_j'/K_p \qquad j=N_1,..,N_2$$

so that the polynomials in the general linear form of the control law (section 2.3):

$$T\Delta u(t) = Rw(t) - Sy(t)$$

are functions of K_p as follows:

$$T = T' \qquad R = R'/K_p \qquad S = S'/K_p \tag{4.1.5}$$

Substituting the above values of T, R and S into the closed-loop transfer function (derived in section 2.3.1) gives:

$$y(t) = \frac{BRq^{-1}w(t) + T\Delta x(t)}{TA\Delta + q^{-1}BS} = \frac{B'R'q^{-1}w(t) + T'\Delta x(t)}{T'A'\Delta + q^{-1}B'S'} \tag{4.1.6}$$

Which indicates that the closed-loop output response is invariant of process gain changes.

4.1.2 Changes in Process Dynamics

Changes in the dynamics of the process imply changes in the location of the open-loop zeros (as well as the open-loop poles). In order to be suitable for nonminimum phase processes, the design and tuning parameters must be selected such that GPC does not attempt to cancel the zeros of the process. Obviously, with the open-loop zeros retained in the closed-loop, the output performance cannot be made completely invariant of the changes in dynamics. In many applications, however, the open-loop zeros do not have a significant influence on the closed-loop response. With good model identification and with the design/tuning parameters selected according to one of the configurations recommended in the following sections, the output performance will be relatively insensitive to changing dynamics.

4.2 Tuning Strategy #1 based on Output Horizon Configuration

A useful strategy for implementation of the Generalized Predictive Controller is one wherein:

1) the majority of the design parameters are fixed during the commissioning stage when the controller is first installed

2) a small number of "active" tuning parameters (preferably one) are made available for adjusting the closed-loop speed of response in a predictable manner, during operation.

It should be possible for the user to obtain control action which ranges from detuned (corresponding to a conservative control objective) to strong (giving "tight" control of the measured variable).

Note: the settings of the design parameters fixed during the commissioning stage are said to define a *configuration* of GPC.

The first proposed tuning strategy is based on the "Output Horizon" configuration of GPC. For this configuration, the design parameters are assigned the following fixed values during commissioning:

$$NU=1, \quad N_1=1, \quad P=1, \quad \lambda=0 \qquad (4.2.1a)$$

The active tuning parameter, N_2, is then used to vary the speed of response with:

$$d+1 \leq N_2 \leq \infty$$

where d is the integer time delay of the process.

As $N_2 \rightarrow \infty$ these are the settings for the mean-level controller described in section 2.5.1. Such a controller is very conservative; at each time step, the control increment expected to reduce the error to zero at steady state is implemented. On the other hand, for $N_2=d+1$ we have a minimum variance (MV) type controller which attempts to eliminate the "d+1" step ahead prediction error. The control adjustments generated by MV control are unacceptably large in most applications. This type of controller is also very sensitive to underestimation of the dead-time, d, and may only be used for minimum phase processes.

For practical application of this Output Horizon configuration of GPC to real processes it is possible to achieve the useful range of responses with :

$$d_{max}+1 < N_2 \leq t_s \qquad (4.2.1b)$$

where d_{max} is the maximum expected time delay and t_s is the settling time of the process in sampling intervals (including the delay). Note that there exists a good intuitive "feel" for the selection of N_2 which is related to the desired closed-loop response time. For start-up in an uncertain environment, a large value of N_2 is recommended. In Appendix A, it is shown that, even in the presence of model-plant mismatch, the closed-loop system using GPC with NU=1, $N_1=1$ and $\lambda=0$ is guaranteed to be stable for large enough N_2 as long as the open-loop system and the model [illegible] stable.

The Output Horizon configuration has been recommended by Maurath et al. (1985) for predictive control of unconstrained SISO systems using controllers based on impulse or step response models (i.e. DMC). It may be used in the manner described for all processes with the exception of continuous-time NMP plants which exhibit on initial "inverse" response. The selection of N_2 for such plants requires more careful consideration as discussed in the next section.

4.2.1 Inverse Response Plants

The initial response of a continuous-time nonminimum phase process to a step input is in the opposite direction as the final steady state value. In order to provide stable control over such a process, the maximum output horizon, N_2, must be large enough to "look beyond" the inverse portion of the response. For the Output Horizon configuration:

$$G_r = \left[g_0 \; g_1 \; \cdots \; g_{N_2-1}\right]^T$$

and

$$\left[G_r^T G_r\right]^{-1} G_r^T = \left[g_0 \; g_1 \; \cdots \; g_{N_2-1}\right] \Big/ \sum_{i=0}^{N_2-1} g_i^2 \tag{4.2.2}$$

where g_i, $i=0,..,N_2-1$ are step response coefficients of the model. In terms of the notation of section 2.3:

$$h_j = g_{j-1} \Big/ \sum_{i=0}^{N_2-1} g_i^2 \qquad j=1,..,N_2 \tag{4.2.3}$$

and the polynomial R in the general linear form of the control law:

$$R = \sum_{j=1}^{N_2} h_j = \sum_{j=1}^{N_2} g_{j-1} \Big/ \sum_{i=0}^{N_2-1} g_i^2 \tag{4.2.4}$$

Assume that initially the system is at steady state (with $y(t)=0$ for simplicity) when there is a step change in the setpoint $w(t)$. The control action taken is:

$$\Delta u(t) = Rw(t) \tag{4.2.5}$$

In order for the controller to take action in the proper direction we must have:

$$\text{sign}(R) = \text{sign}\left[\sum_{j=1}^{N_2} g_{j-1}\right] = \text{sign}(K_p) \qquad (4.2.6)$$

where K_p=[B(1)/A(1)], is the steady state gain of the process.

If N_{min} is the lowest value of N_2 for which this criterion if satisfied then we must have $N_2 \geq N_{min}$ for stable control. Note that it is possible to check on-line if N_2 is large enough to "look beyond" the inverse response and satisfy this criteria.

4.3 Tuning Strategy # 2 based on Lambda Weighting Configuration

A second strategy for simplifying the task of tuning GPC is based on what will be referred to as the "Lambda Weighting" configuration. The following design parameters are fixed during commissioning:

$$NU = na+1, \quad N_1 = nb+1, \quad N_2 \geq NU+N_1-1, \quad P=1 \qquad (4.3.1a)$$

where na and nb are the orders of the model polynomials $A(q^{-1})$ and $B(q^{-1})$ (nb includes the delay).

The scalar control weighting, λ, is designated as the active tuning parameter with

$$0 \leq \lambda \leq \infty$$

Normally if N_2 is set roughly equal to the rise time of the process (defined as the time elapsed from a step change in the input until the output reaches 63% of the final steady state value) in sampling intervals,

$$N_2 \simeq t_r \qquad (4.3.1b)$$

the above inequality for N_2 is satisfied. N_2 set in this way will also normally be larger than N_{min} (defined in the previous section) so that there will be no problem controlling inverse response plants.

For this configuration, λ=0 yields deadbeat control. Placing all of the poles at the origin in this manner drives the controlled variable to the setpoint in at most nb+1 steps. This rapid response requires stronger control actions than are desirable in most circumstances. Nonzero values of

λ may be used to achieve the desired speed of response; as λ is increased the controller is progressively detuned. In the following section it will be shown that one closed-loop pole approaches z=1 while the remainder converge to the open-loop poles as $\lambda \to \infty$. This implies that for open-loop unstable processes, the closed-loop system is stable only if λ is below some critical value, λ_{crit}, which is not known *a priori*. For this reason the Lambda Weighting configuration is not recommended for unstable plants.

4.3.1 Infinite Control Weighting

Recall the GPC control law is:

$$\tilde{u} = \left[G_r^T G_r + \lambda I\right]^{-1} G_r^T (w - f)$$

For sufficiently large values of λ we have

$$\left[G_r^T G_r + \lambda I\right]^{-1} G_r^T \simeq G_r^T/\lambda \tag{4.3.2}$$

so that the elements of the first row of this matrix are:

$$h_j = g_{j-1}/\lambda \qquad j=N_1,..,N_2 \tag{4.3.3}$$

In the general linear form of the control law, the polynomials T, R and S, from eqn. (2.3.7) become:

$$T = 1 + q^{-1}\left[\sum_{j=N_1}^{N_2} g_{j-1}\bar{G}_j/\lambda\right] , \quad R = \sum_{j=N_1}^{N_2} g_{j-1}/\lambda \quad \text{and} \quad S = \sum_{j=N_1}^{N_2} g_{j-1}F_j/\lambda \tag{4.3.4}$$

In the limit as $\lambda \to \infty$,

$$\underset{\lambda\to\infty}{\text{limit}}\ (T) = 1 , \quad \underset{\lambda\to\infty}{\text{limit}}\ (R) = 0 \quad \text{and} \quad \underset{\lambda\to\infty}{\text{limit}}\ (S) = 0 \tag{4.3.5}$$

the closed-loop characteristic polynomial becomes

$$\underset{\lambda\to\infty}{\text{limit}}\ (TA\Delta + q^{-1}BS) = A\Delta \tag{4.3.6}$$

This verifies that in the limit as $\lambda \to \infty$, the closed-loop poles approach the locations of the open-loop poles and an additional pole at z=1. The pole at z=1 is a consequence of weighting control **increments** in the GPC cost function.

4.3.2 Maintenance of Performance

In section 4.1.1 we showed that, with $\lambda=0$, the response of the controlled variable is invariant to process gain changes as long as we have exact modeling. Experience confirms that fixed λ weighting implies a tradeoff between output performance and control effort. However, if λ is scaled such that it is proportional to the gain of the process squared, the output performance is independent of gain changes once again. To see this, reconsider the development in section 4.1.1 where the process gain, K_p, was factored out:

$$B = K_p B' \qquad F_j = F_j' \qquad G_j = K_p G_j'$$

(The prime denotes a quantity for a unity process gain.) The elements of the matrix G_r are proportional to the gain:

$$G_r = K_p G_r'$$

It λ is scaled according to,

$$\lambda = \bar{\lambda} K_p^2 = \bar{\lambda}\left[B(1)/A(1)\right]^2 \qquad \bar{\lambda} = \text{a constant} \tag{4.3.7}$$

then,

$$\left[G_r^T G_r + \lambda I\right]^{-1} G_r^T = \left[G_r^{T'} G_r' + \bar{\lambda} I\right]^{-1} G_r^{T'}/K_p \tag{4.3.8}$$

The elements of the first row of this matrix are inversely proportional to the process gain and proceeding in the manner of section 4.1.1 we find again that:

$$T = T' \qquad R = R'/K_p \qquad S = S'/K_p$$

Since $T=T'$ and R and S appear in the closed-loop transfer frunction as products with B, the response of the output is independent of the actual process gain.

The output performance cannot be made completely independent of changes in process dynamics. However, it is possible to desensitize the closed-loop system to these changes if instead of using (4.3.7), the actual λ used in the control calculations is scaled such that it is proportional to $[B(1)]^2$:

$$\lambda = \lambda_{rel}[B(1)]^2 \qquad \lambda_{rel} = \text{a constant} \qquad (4.3.9)$$

where the "relative" control weighting, λ_{rel}, is chosen by the user.

To explain intuitively why this should work, suppose that the dynamics of a process become slower. If the sample time does not change, the coefficients of the B polynomial (estimated on-line) and the elements of the G_r matrix (which are step response coefficients) will decrease in a similar fashion. The "balance" between the two terms in the matrix

$$\left[G_r^T G_r + \lambda I\right] = \left[G_r^T G_r + \lambda_{rel}[B(1)]^2 I\right]$$

which is inverted during the control calculation is maintained. In this situation, the changing dynamics will have less influence on the output response. (Note: earlier in this section it was shown that maintaining the "balance" between the two terms is necessary for the output response to be invariant of gain changes.)

There are also numerical reasons for preferring the scaling of eqn. (4.3.9) to that of eqn. (4.3.7). Reliable estimates of the process gain are difficult to obtain on-line; A(1) is often quite small and, for self-tuning applications where estimated parameters fluctuate, $\hat{A}(1)$ may approach zero for transient periods.

The scaling of λ by $[B(1)]^2$ was first suggested by Lam (1980) for use with N-stage LQ based state-space self-tuning controllers.

4.3.3 Selection of Control Weighting

The range of values of λ_{rel} (or λ) which provide reasonable detuning of the controller are not immediately obvious. Based on simulation and experimental trials, λI must be within an "order of magnitude" of $G_r^T G_r$ to have some effect. If λI is "approximately equal" to $G_r^T G_r$, control increments will be roughly halved relative to deadbeat control (corresponding to $\lambda=0$). Using the trace of $G_r^T G_r$ as a measure of its magnitude, a good starting value of λ which is expected to reduce the deadbeat control increments by a factor of approximately $1/(m+1)$ is

$$\lambda_{start} = \frac{m|tr(G_r^T G_r)|}{NU} \qquad (4.3.10)$$

This value of λ may be used to calculate a starting value for λ_{rel} via eqn. (4.3.9) which may then be adjusted during operation to fine tune the response.

4.4 Tuning Strategy # 3 based on Detuned Model-Following Configuration

GPC can be set up to follow the closed-loop model $M(q^{-1})=1/P(q^{-1})$ exactly, in which case the process zeros are cancelled. In order to have a practical control scheme for plants with arbitrary zeros we must "detune" the model following capabilities. As discussed in section 2.5.2, this may be accomplished by either increasing the output horizon relative to the control horizon or by adding control weighting. The former method is easier to carry out as the amount of control weighting required to achieve stability for NMP processes without detuning the control loop more than necessary is difficult to determine *a priori*. "Detuned Model-Following" may be obtained by commissioning GPC with:

$$NU = na+1, \quad N_1 = 1, \quad N_2 > d+NU, \quad \lambda = 0 \qquad (4.4.1a)$$

and using the inverse closed-loop model, $P(q^{-1}) = 1/M(q^{-1})$, as the active tuning parameter. (Note that $N_2 = d+NU$ would yield exact model following.) By setting $NU = na+1$ we are allowing the controller sufficient degrees of freedom to follow the model sufficiently closely but not exactly. Simulation runs indicate that incrementing N_2 by only a small amount over that required for exact model-following ensures that no attempt is made to cancel open-loop zeros. For processes with variable time delays, N_2 should be set large enough so that it exceeds d_{max}+NU where d_{max} is the largest expected dead-time. To be on the safe side, a rather larger value of N_2 is recommended, corresponding more closely with the rise time of the process, (defined previously to include the dead-time and any inverse response time) in sampling intervals:

$$N_2 \simeq t_{rs} \qquad (4.4.1b)$$

Recommended methods for selecting the closed-loop model, $M(q^{-1})$, to achieve a desired response will now be considered.

4.4.1 Specification of Desired Closed-Loop Models

A good way of selecting M is to consider an appropriate continuous-time model and find its equivalent pulse transfer function (Clarke, 1983). The simpliest continuous model for an overdamped response is a 1st order process with a gain of one:

$$M(s) = 1/(1 + \tau_{CL} s) \tag{4.4.2}$$

The pulse transfer function for a sample interval T_s,

$$M(z^{-1}) = \frac{(1 - p_1)z^{-1}}{1 - p_1 z^{-1}} \tag{4.4.3}$$

has a pole, $p_1 = e^{-T_s/\tau_{CL}}$, where τ_{CL} is the desired closed-loop time constant.

Since it is of no interest to increase the dead-time in the process, the polynomial $P(z^{-1})$ should be selected as

$$P(z^{-1}) = \frac{1 - p_1 z^{-1}}{1 - p_1} \tag{4.4.4}$$

Note that $P(1)=1$ to ensure offset-free behavior. This first order closed-loop model is primarily applicable to plants which have 1st order characteristics. Experience indicates that excessive control action is required to make many higher order "industrial-type" processes respond in a 1st order manner. For these cases a more appropriate model is 2nd order:

$$M(s) = \frac{1}{\tau^2 s^2 + 2\zeta\tau s + 1} \tag{4.4.5}$$

where τ is the natural period and ζ is the damping factor.

The z-transform of this model (including ZOH) for underdamped systems ($\zeta<1$) is given by Neuman and Baradello (1979):

$$M(z^{-1}) = \frac{n_1 z^{-1} + n_2 z^{-2}}{1 + p_1 z^{-1} + p_2 z^{-2}} \tag{4.4.6}$$

with,

$$p_1 = -2e^{-\zeta T_s/\tau}\cos\left[(T_s/\tau)\sqrt{1-\zeta^2}\,\right], \qquad p_2 = e^{-2\zeta T_s/\tau},$$

$$\text{and} \qquad n_1 + n_2 = 1 + p_1 + p_2$$

(n_1 and n_2 as functions of T_s, τ and ζ are given in the above reference.)

Since we are in a detuned mode where we do not follow the model precisely, it makes little difference if the numerator dynamics are ignored. A polynomial P may then be used:

$$P(z^{-1}) = \frac{1 + p_1 z^{-1} + p_2 z^{-2}}{1 + p_1 + p_2} \tag{4.4.7}$$

For step setpoint changes, the 63% rise time (dominant time constant) is approximately 2τ as long as $0.5<\zeta<1.0$. The fractional overshoot (for a 2nd order system given by eqn. (4.4.5)) is solely a function of the damping factor (Stephanopoulos, 1984):

$$o_v = e^{-\left\{\frac{\pi\zeta}{\sqrt{1-\zeta^2}}\right\}} \tag{4.4.8}$$

Rearranging this equation gives the damping factor as an explicit function of the overshoot:

$$\zeta = \sqrt{\frac{(\ln o_v)^2}{\pi^2+(\ln o_v)^2}} \tag{4.4.9}$$

For convenience, the damping factor is given for different values of the percentage overshoot in the following table:

% Overshoot, o_v	Damping factor, ζ
5	0.69
10	0.59
15	0.52
20	0.46
25	0.40

4.5 Interpretations of the C polynomial in the CARIMA Model

The Generalized Predictive Controller was extended to incorporate the C polynomial in section 2.2.4. In that derivation, C represented the CARIMA noise model polynomial. It follows immediately that if an on-line estimate, $\hat{C}$, can be obtained using an appropriate recursive identification algorithm (e.g. Extended Least Squares or Recursive Maximum Likelihood) better disturbance rejection may be possible. However, an alternative engineering approach is to interprete C as a design polynomial, C_c, which may be selected in such a way as to achieve desired performance. The important observation that C does not affect the setpoint tracking properties provides guidance for its specification and allows GPC to meet independent servo and regulatory objectives.

4.5.1 Estimating $\hat{C}$ using Extended Least Squares

The general CARIMA model,

$$A(q^{-1})\Delta y(t) = B(q^{-1})\Delta u(t-1) + C(q^{-1})\xi(t)$$

can be rearranged into the extended least squares form:

$$\Delta y(t) = -q(A-1)\Delta y(t-1) + B\Delta u(t-1) + q(C-1)\xi(t-1) + \xi(t) \tag{4.5.1}$$

or,

$$\Delta y(t) = \theta^T\phi(t) + \xi(t) \tag{4.5.2}$$

where

$$\theta = \begin{bmatrix} a_1 \dots a_{na} & b_0 \dots b_{nb} & c_1 \dots c_{nc} \end{bmatrix}^T$$

and

$$\phi(t) = \begin{bmatrix} -\Delta y(t-1) \dots -\Delta y(t-na) & \Delta u(t-1) \dots \Delta u(t-1-nb) & \xi(t-1) \dots \xi(t-nc) \end{bmatrix}^T$$

Since the noise terms $\xi(t-1),..,\xi(t-nc)$ are unmeasureable they must be proxied by either the *a priori* estimation error, $\epsilon(t)$, or the *a posteriori* residual, e(t),

$$\epsilon(t) = \Delta y(t) - \hat{\theta}^T(t-1)\phi(t) \tag{4.5.3}$$

$$e(t) = \Delta y(t) - \hat{\theta}^T(t)\phi(t) \tag{4.5.4}$$

using the parameter estimates,

$$\hat{\theta} = \left[\hat{a}_1 \dots \hat{a}_{na}\ \hat{b}_0 \dots \hat{b}_{nb}\ \hat{c}_1 \dots \hat{c}_{nc}\right]^T$$

An ordinary, recursive least squares algorithm employing the above "extended" regressor and parameter vector forms the Extended Least Squares (ELS) method. Certain positive-real conditions must be satisfied for convergence of $\hat{\theta}$ (Ljung and Söderström, 1983). If they are violated, a more sophisticated routine such as Recursive Maximum Likelihood (RML) must be used. When either ELS or RML is used, it is important that $\hat{C}(q^{-1})$ remain inverse-stable as the estimators are themselves systems with poles given by $\hat{C}(q^{-1})$. This condition may be ensured by performing spectral factorization to give a stable representation with identical statistical properties (Åström and Wittenmark, 1984). This, however, adds considerably to the on-line computational burden (Tuffs, 1984).

The use of *a posteriori* residuals instead of *a priori* prediction errors results in faster transient convergence of the $\hat{C}$ parameters (Ljung and Söderström, 1983). Experience indicates that, even in this case, the $\hat{C}$ parameters converge at a rate which is an order of magnitude slower than the $\hat{A}$ and $\hat{B}$ coefficients (Isermann, 1981).

Most real processes have more than one disturbance or noise source acting on them. For example, ξ_1 may represent a disturbance which enters at the process input, ξ_2 may be measurement noise acting directly on the process output, etc. This can be modelled as:

$$A\Delta y(t) = B\Delta u(t-1) + C_1\xi_1(t) + C_2\xi_2(t) + \dots \qquad (4.5.5)$$

The noise components can be combined into a single random sequence $C\xi(t)$ by spectral factorization. Only if the individual noise variances σ_i^2 remain constant will $C(q^{-1})$ be a time invariant polynomial. In practical applications this does not hold and therefore we expect $C(q^{-1})$ to vary with time. Coupled with the slow convergence of the $\hat{C}$ parameters this implies that successful identification of the noise model $C(q^{-1})$ is unlikely. Since $\hat{C}(q^{-1})$ may have little correspondence to the actual noise structure, the use of a design polynomial (which may represent prior knowledge of the process noise) has been suggested by several authors.

4.5.2 Design Polynomial, C_c

Wellstead and Sanoff (1981) introduce an arbitrary filter with fixed coefficients forming a surrogate noise model into the Pole Placement algorithm. Clarke (1983) uses an *a priori* estimate of $C(q^{-1})$ denoted by $T(q^{-1})$ (not the same as the T polynomial appearing in the general linear form of the control law (2.3.4)) in the GMV controller for tailoring the response to disturbances. The choice of $1/T(q^{-1})$ as a low pass filter detunes the control action for high frequency transient disturbances (Tuffs, 1984). In particular, the use of $T(q^{-1})$ prevents overcompensation when controlling industrial processes known to be affected by rapid variations in load but which respond relatively slowly to the control signal.

In the majority of the following sections we will consider the use of a design polynomial , C_c, in the Generalized Predictive Control algorithm.

4.5.3 One Degree of Freedom Structure

It is instructive to consider the special case where the polynomial representing the disturbance term in the CARIMA model, $C(q^{-1})$ is equal to $A(q^{-1})$:

$$y(t) = [B/A]u(t-1) + \xi(t)/\Delta$$

(i.e. the disturbances consist of random steps acting directly at the output of the process). The solution of the Diophantine Identity,

$$CP = A\Delta E_j + q^{-j}F_j$$

is then,

$$F_j = A \qquad E_j = (P - q^{-j})/\Delta \tag{4.5.6}$$

so that,

$$G_j = E_jB = (P - q^{-j})B/\Delta$$

Substituting for F_j, the open-loop prediction j steps ahead (eqn. (2.2.22)) becomes:

$$f(t+j) = y(t) + \overline{G}_j\Delta u(t-1)/C \tag{4.5.7}$$

The open-loop prediction of the auxiliary output is simply the current measurement plus the expected influence of past controls. This implies that the predicted future effect of a disturbance on the output is equal to the current output effect of the disturbance. Note that this is the default type of disturbance prediction used by long-range predictive control algorithms based on impulse or step response models (e.g. DMC) without a single series forecaster (Man, 1984) or closed-loop observer to predict the future effect of residuals (Navratil et al., 1988).

The polynomials S and R in the general linear form of the GPC law with C=A are, from section 2.3.2:

$$S = R = A\left[\sum_{j=N_1}^{N_2} h_j\right] \tag{4.5.8}$$

Recalling the closed-loop expression for y(t), eqn. (2.3.15), the error e(t) may be written:

$$e(t) = w(t) - y(t) = \frac{\left[TA\Delta + q^{-1}B(S-R)\right]w(t) - TC\Delta x(t)}{TA\Delta + q^{-1}BS} \tag{4.5.9}$$

With C=A and S=R,

$$e(t) = \frac{TA\Delta\left[w(t) - x(t)\right]}{TA\Delta + q^{-1}BS} \tag{4.5.10}$$

Therefore the response of the control error to disturbances and setpoint changes is identical apart from the sign. If disturbances (as observed at the process output) and setpoint changes have the same form (e.g. steps) the controller can be designed "optimally" for both. If, on the other hand, disturbances have a different form (i.e. C≠A), then it is generally not possible to tune this "one degree of freedom" controller for optimal servo and regulatory response. It is necessary to consider a structure with "two degrees of freedom" (Stephanopoulos and Huang, 1986; Zafiriou and Morari, 1987). We will consider several ways of designing GPC for independent servo and regulatory control in section 4.6. First, however, we will show that in the absence of modelling errors (in A and B) the C polynomial does not affect servo control.

4.5.4 Independence of Servo Control and C

To show that the C polynomial incorporated into the GPC control law does not affect setpoint tracking properties, recall equation (2.2.32) where the primes denoted polynomials for the case where C=1:

$$\tilde{G}_j + q^{-j}(\overline{G}_j A\Delta + BF_j)/CA\Delta = \tilde{G}_j' + q^{-j}(\overline{G}_j' A\Delta + BF_j')/A\Delta$$

Therefore,

$$\overline{G}_j A\Delta + BF_j = C(\overline{G}_j' A\Delta + BF_j') \qquad (4.5.11)$$

Writing this equation for $j=N_1,\ldots,N_2$, multiplying by the set of real numbers, h_j, and summing yields:

$$\sum_{j=N_1}^{N_2} h_j\overline{G}_j A\Delta + \sum_{j=N_1}^{N_2} h_j BF_j = C\left[\sum_{j=N_1}^{N_2} h_j\overline{G}_j' A\Delta + \sum_{j=N_1}^{N_2} h_j BF_j'\right] \qquad (4.5.12)$$

The summation terms are related to the polynomials in the general linear form of the control law as may be seen by rearranging eqn. (2.3.7) (for C=1) and (2.3.14):

$$\sum_{j=N_1}^{N_2} h_j\overline{G}_j' = q(T'-1) \qquad \sum_{j=N_1}^{N_2} h_j F_j' = S' \qquad (4.5.13a)$$

$$\sum_{j=N_1}^{N_2} h_j\overline{G}_j = q(T-C) \qquad \sum_{j=N_1}^{N_2} h_j F_j = S \qquad (4.5.13b)$$

So that eqn. (4.5.12) may be written:

$$q(T-C)A\Delta + BS = C[q(T'-1)A\Delta + BS'] \qquad (4.5.14)$$

which after rearrangement yields:

$$TA\Delta + q^{-1}BS = C[T'A\Delta + q^{-1}BS'] \qquad (4.5.15)$$

This equation indicates that the characteristic polynomial for a general C is formed as the product of the characteristic polynomial for the case C=1 and the polynomial C (i.e. the roots of C become closed-loop poles). Noting also that R = CR', the closed-loop transfer function given by eqn. (2.3.15)

$$y(t) = \frac{BRq^{-1}w(t) + TC_0\Delta x(t)}{TA\Delta + q^{-1}BS} = \frac{CBR'q^{-1}w(t) + TC_0\Delta x(t)}{C\left[T'A\Delta + q^{-1}BS'\right]} \qquad (4.5.16)$$

where the "true" noise structure polynomial C_0 is distinguished from C used in the control law. Thus in the absence of process modelling errors (i.e. $A=A_0$, $B=B_0$) the C polynomial incorporated into the control law does not affect the response to setpoint changes. However, since in general, $T \neq T'C$, the disturbance rejection properties are modified (as is the servo response when there is MPM). If in addition to the process model, the noise model is exact ($C=C_0$) the disturbance predictions are optimal and the output variance can be minimized, if desired.

4.5.5 Prediction of Residuals

The C polynomial can be interpreted as modifying the prediction of the residuals into the future (Foley, 1988). The residual is defined as the difference between the actual output and the model predicted output at the current time. Recall the open-loop prediction of the output j steps ahead was given by eqn. (2.2.22):

$$f(t+j) = \left[F_j/C\right]y(t) + \left[\bar{G}_j/C\right]\Delta u(t-1)$$

An expression for $\bar{G}_j$ may be obtained by multiplying the Diophantine Identity (2.2.16) by B and substituting for G_j from eqn. (2.2.20):

$$\bar{G}_j = q^j C\left[PB/A\Delta - \tilde{G}_j\right] - BF_j/A\Delta \qquad (4.5.17)$$

Substituting $\bar{G}_j$ into eqn. (2.2.22) gives:

$$f(t+j) = \left[\frac{F_j}{C}\right]\left[y(t) - \left\{\frac{B}{A\Delta}\right\}\Delta u(t-1)\right] + q^j\left[\frac{PB}{A\Delta} - \tilde{G}_j\right]\Delta u(t-1) \qquad (4.5.18)$$

The coefficients of $\tilde{G}_j$ are the first j coefficients of $PB/A\Delta$, such that,

$$f(t+j) = \left[\frac{F_j}{C}\right]\left[y(t) - \left\{\frac{B}{A}\right\}u(t-1)\right] + G_j^*\Delta u(t-1) \qquad (4.5.19)$$

where $G_j^* = g_j + g_{j+1}q^{-1} + g_{j+2}q^{-2} + \ldots$

(For the case where P=1, these are step response coefficients of the process.) Since the residual,

$$r(t) = y(t) - (B/A)u(t-1)$$

the open-loop output prediction may be written:

$$f(t+j) = \left[F_j/C\right]r(t) + G_j\Delta u(t-1) \qquad (4.5.20)$$

Thus, the transfer function F_j/C determines how the past and present residuals are predicted into the future. Since both disturbances and model-plant mismatch contribute toward the residuals, it should be clear that the C polynomial influences both the robustness and disturbance rejection properties of the closed-loop. The second term in the previous equation represents the predicted effect on the output of past control increments based on the I/O model of the process.

The explicit prediction of residuals is emphasized in the block diagram of Figure 4.1.

4.6 Independent Servo and Regulatory Objectives

In order to have the capability of achieving independent servo and regulatory performance objectives, a control law must have at least two degrees of freedom. If, in addition, the two "control elements" are partially or completely decoupled the initial tuning of the controller will be simplified. For GPC we know that the horizons N_1, N_2 and NU, the control weighting, λ, and the P weighting polynomial influence the overall behavior of the closed-loop (servo and regulatory). A setpoint transfer function prefilter,

$$w(t) = F_{sp}y_{sp}(t) \quad \text{with} \quad F_{sp}(1) = 1$$

may be used to modify the closed-loop servo response alone. The controller design polynomial, C_c, tailors the rejection of disturbances and behavior of the loop with MPM. Clearly, several possibilities exist for designing a controller with two degrees of freedom. By way of introduction, consider the Exact Model-Following configuration of GPC ($NU=N_2-d$, $N_1=1$, $N_2>d_1$, $\lambda=0$) with a setpoint prefilter and C_c design polynomial. The closed-loop transfer function may be written:

$$y(t) = F_{sp}\frac{q^{-(d+1)}}{P}\,y_{sp}(t) + \frac{T^{*}C_0\Delta}{C_c P}\,x(t) \qquad (4.6.1)$$

where $T^{*} = q^{-d}T/BR'$

F_{sp}, P and C_c should be selected in such a way as to transform the closed-loop expression into:

$$y(t) = M_s q^{-(d+1)} y_{sp}(t) + M_r T^{*} C_0 \Delta x(t) \qquad (4.6.2)$$

where M_s is the closed-loop servo model and M_r is the closed-loop "regulatory model".

The three possibilities listed in Table 4.1 will be considered in turn.

Table 4.1 GPC Alternative Settings for Independent Servo and Regulatory Control

Case	F_{sp}	P	C_c
1	1	$1/M_s$	M_s/M_r
2	M_s	1	$1/M_r$
3	M_s/M_r	$1/M_r$	1

In the following chapters it will be shown that it is possible to achieve very similar closed-loop behavior with any of the three methods. However, the interpretation of each alternative is slightly different.

4.6.1 Two Degrees of Freedom, Case 1: C_c as a Disturbance Model

With the setpoint prefilter disabled, P can be set equal to the inverse servo model and C_c used to tailor the response to disturbances. A typical example will clarify this method. Assume that a 2nd order servo model has been choosen according to section 4.4.1:

$$M_s = 1/P = \frac{1 + p_1 + p_2}{1 + p_1 q^{-1} + p_2 q^{-2}}$$

If we select a "regulatory model" with the additional pole $-c_1$:

$$M_r = M_s/C_c = \left[\frac{1 + p_1 + p_2}{1 + p_1 q^{-1} + p_2 q^{-2}}\right]\left[\frac{1}{1 + c_1 q^{-1}}\right]$$

then we are using $C_c = 1+c_1q^{-1}$ to represent knowledge of the disturbances or model-plant mismatch characteristics likely to be encountered. If $1/C_c$ is a low-pass filter, high frequency disturbances are filtered and the controller, tuned for servo control, will also reject disturbances in a desirable manner. The importance of using C_c in this way to provide robustness in the presence of model-plant mismatch will be demonstrated in the following chapter.

A few remarks are in order. The servo and regulatory modes are not completely decoupled. (Although the C_c polynomial may be adjusted to modify the rejection of disturbances without affecting the response to setpoint changes significantly, given a reasonable process model). In by far the majority of applications this is not a disadvantage since the user will not have sufficient knowledge or reason to specify completely different servo and regulatory modes.

For practical applications, exact model following must be "detuned". In this case, the closed-loop will only approximately follow the desired models.

Rather than employing P, the control engineer may select N_1, N_2, NU and λ (according to Output Horizon or Lambda Weighting configurations) to achieve the desired servo response. In these situations, the engineer will be less certain of the locations of the closed-loop poles, but better performance may be achieved.

The alternative presented in this section (Case 1) is recommended for control with two degrees of freedom unless it is desired to decouple the servo and regulatory modes completely. It is used extensively in this thesis.

4.6.2 Two Degrees of Freedom, Case 2: C_c as a Regulatory Model

The second method for providing two degrees of freedom involves prefiltering the setpoint , y_{sp}, and selecting P=1 to give "d+1 step ahead" control. The C_c polynomial may then be used to specify the regulatory response independently.

For example, if the desired servo and regulatory models are 2nd order, then one possibility is:

$$F_{sp} = M_s = \frac{n_1 + n_2 q^{-1}}{1 + m_1 q^{-1} + m_2 q^{-2}}, \qquad F_{sp}(1) = 1$$

$$C_c = 1/M_r = 1 + c_1 q^{-1} + c_2 q^{-2}$$

An all-pole regulatory model is required since in GPC, $C_c(q^{-1})$ is restricted in form to a polynomial. However, the numerator dynamics resulting from z-transformation of a continuous 2nd order model may be retained in the servo model, if desired. (The discrete model coefficients are related to the natural period and damping factor for an underdamped system by equations given in section 4.4.1).

Note again that for Detuned Model-Following the closed-loop behavior will not correspond to the specified models exactly. If the Output Horizon or Lambda Weighting configurations are used (rather than Detuned Model-Following) the appropriate tuning parameters should be selected to give "d+1 step ahead" or deadbeat control to the filtered setpoint, respectively.

The design polynomial C_c is used to specify the desired closed-loop regulatory performance. With this interpretation, it does not just represent a model of the disturbances. However, as an alternative, C_c may be selected as the product of a noise model and the inverse closed-loop regulatory model.

4.6.3 Two Degrees of Freedom, Case 3: C_c Disabled

As a third alternative for making use of two degrees of freedom, P may be chosen as the inverse regulatory model and the setpoint prefiltered by the transfer funtion M_s/M_r to achieve the required servo response.

For example, if the user wants to obtain a 1st order respoonse to setpoint changes but a 2nd order regulatory model, then:

$$P = 1/M_r = \frac{1 + p_1 q^{-1} + p_2 q^{-2}}{1 + p_1 + p_2}, \quad C_c = 1$$

and $$F_{sp} = M_s/M_r = \left[\frac{1 + m_1}{1 + m_1 q^{-1}}\right]\left[\frac{1 + p_1 q^{-1} + p_2 q^{-2}}{1 + p_1 + p_2}\right]$$

could be specified. Substitution of P' and F_{sp} into the closed-loop equation (4.6.1) shows that the desired performance can be obtained. However, when model-following is "detuned" the setpoint prefilter will not be able to cancel the regulatory modes exactly; consequently there may be some interaction when the regulatory model is changed.

At an increase in complexity, C_c may be selected as a noise model rather than fixed at unity. In this circumstance, the product $C_c P$ determines the response of the closed-loop to load changes.

4.7 Parameter Estimation

The difficulty of successfully identifying a noise model, $\hat{C}$, suitable for inclusion in the control law has already been pointed out. The question still remains as to how parameter estimation should be structured to give good estimates of the A and B polynomials in the process model. With ordinary RLS, the prediction error must be uncorrelated with the elements of the regressor in order for the parameter estimates to be unbiased. This will only be the case when the true noise polynomial, $C_0=1$. Since this is rarely the case, other methods must be considered. Two alternatives initially look promising.

The first idea consists of using ELS to identify $\hat{A}$, $\hat{B}$ and $\hat{C}$, as described in detail in section 4.5.1. The difference is that now the estimated polynomial, $\hat{C}$, is not employed in the control law. (A design polynomial C_c may be used as discussed in previous sections). In fact, the sole purpose of using ELS is to eliminate bias in the $\hat{A}$ and $\hat{B}$ parameters.

This method has been used by Walgama (1986) to guarantee convergence of the process parameters for an adaptive Kalman Filter Predictor.

The second method involves filtering the regressor by $1/C_e$, normally selected as a low pass filter to remove high frequency components (Tuffs, 1984). Note that the estimator design polynomial, C_e, may or may not be equal to the design polynomial C_c used in the control law. The incremental model used for parameter estimation is then:

$$\Delta y^e(t) = -q(A-1)\Delta y^e(t-1) + B\Delta u^e(t-1) + \left[C_0/C_e\right]\xi(t) \qquad (4.7.1)$$

where the superscript e denotes a signal filtered by $1/C_e$.

If C_e is equal to the true noise term C_0 an RLS estimator with filtered data will give unbiased $\hat{A}$ and $\hat{B}$ parameters. While an inappropriate C_e filter will result in biased parameters, the low pass filtering of the data vector still limits the frequency band over which the linear model is expected to match the process.

Figure 4.2 emphasizes the use of the controller and estimator design polynomials, C_c and C_e, in the application of self-tuning GPC.

4.8 Summary

Three tuning strategies for GPC, which allow the user to vary the closed-loop speed of response over a full range using a single active (variable) tuning parameter were devised. The remaining tuning parameters (defining the configuration) are fixed during commissioning at the values given in the following table:

Configuration	N_1	N_2	NU	λ	$P(q^{-1})$
Output Horizon	1	variable	1	0	1
Lambda Weighting	nb+1	$\geq NU+N_1-1$	na+1	variable	1
Detuned Model-Following	1	>d+NU	na+1	0	variable

Controllers based on each of these configurations of GPC may be applied to chemical processes without restriction. The Lambda Weighting configuration is, however, not recommended for open-loop unstable plants. The Output Horizon and Detuned Model-Following configurations automatically guarantee that the closed-loop response will be invariant to process gain changes (and relatively unaffected by changes in process dynamics) given an accurate process model. The control weighting, λ, must be proportional to $[B(1)]^2$ to ensure the output performance is maintained when using the Lambda Weighting configuration.

The use of a design polynomial, $C_c(q^{-1})$, in the GPC control law is recommended, rather than attempting to estimate the noise model on-line. In the ideal case of no process-model mismatch, it has been shown that C_c does not affect the setpoint tracking properties of the closed-loop. In practical applications it should be used to tailor the rejection of disturbances and improve robustness to modelling errors, as demonstrated in Chapter 5.

The inclusion of the C_c design polynomial or a setpoint prefilter gives the Generalized Predictive Controller the capability of meeting independent servo and regulatory objectives. Three alternatives for utilizing two degrees of freedom were discussed in this chapter.

A related design polynomial, $C_e(q^{-1})$, was introduced to filter the incremental regressor before parameter estimation. In the following chapter, it will be shown that this filter can be specified in order to focus attention on lower frequencies and allow identification of process models suitable for generating long-range predictions.

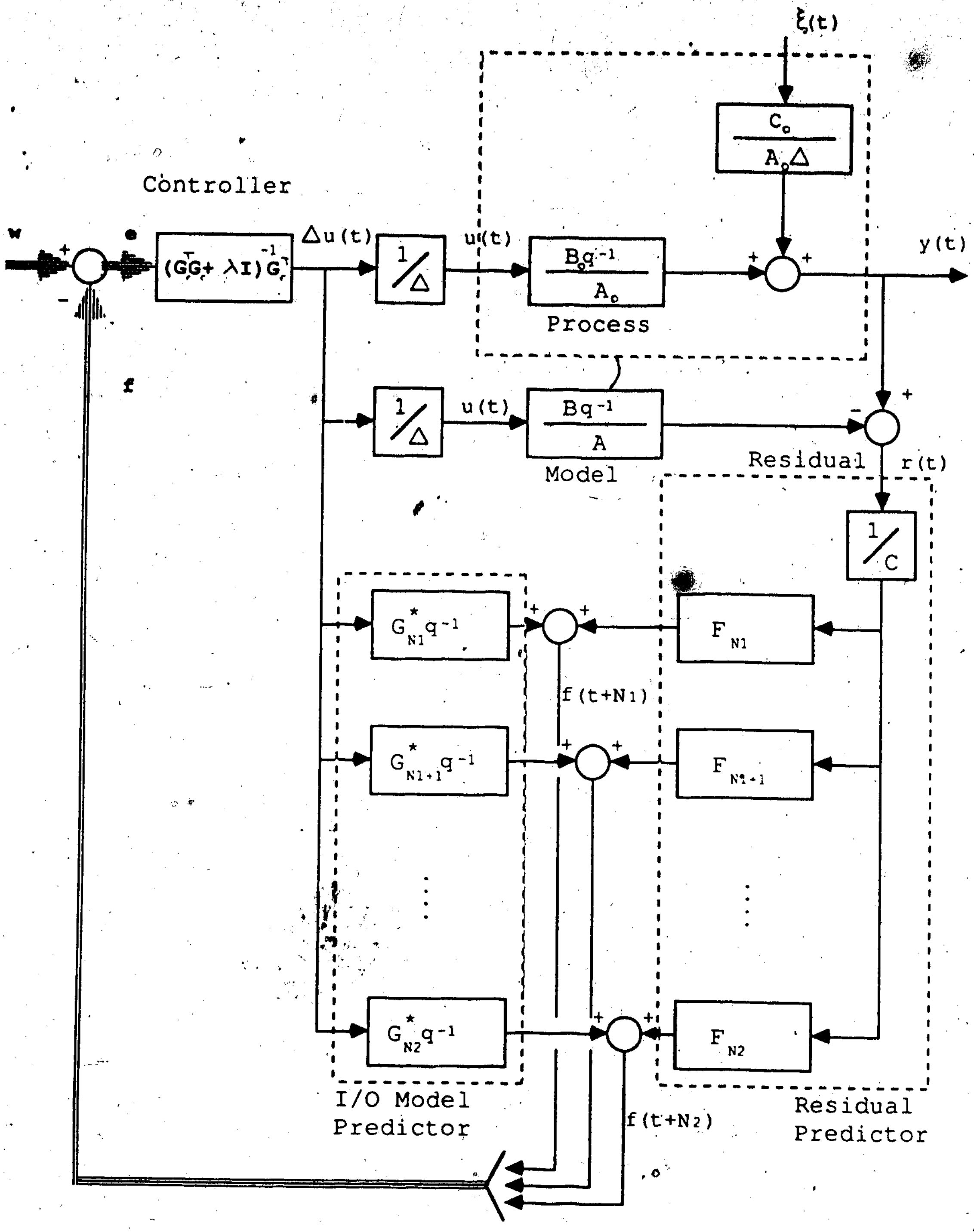


Figure 4.1 Block Diagram of GPC with Explicit Prediction of Residuals

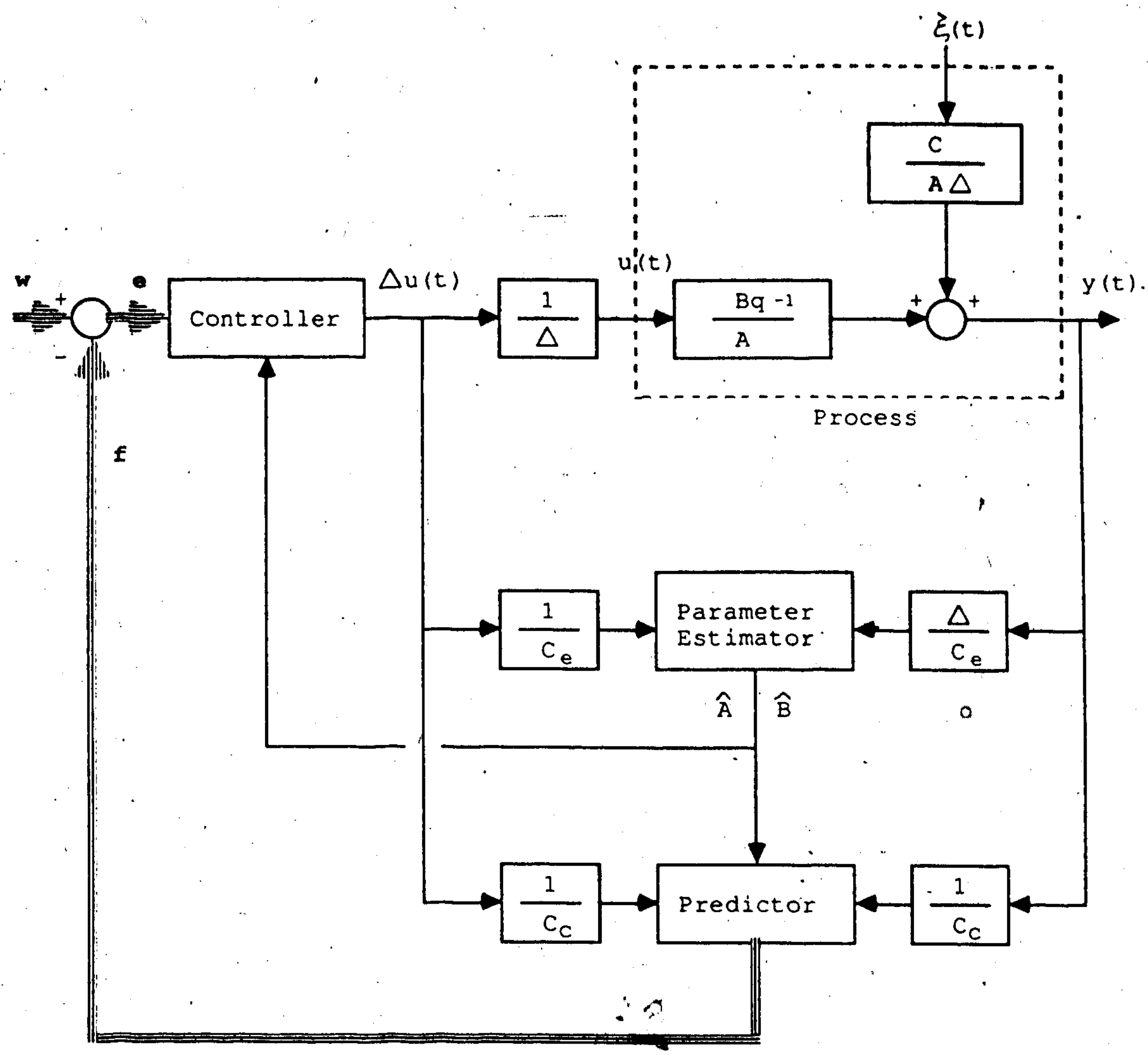


Figure 4.2 Block Diagram of Self-Tuning GPC with Controller Design Polynomial, C_c, and Estimator Filter Polynomial, Ce

5. EVALUATION OF THE GENERALIZED PREDICTIVE CONTROL TUNING STRATEGIES

Random and unmeasurable events make it difficult to interpret the behavior of a self-tuning controller during an experimental application. Better understanding is achieved by carrying out simulations under carefully chosen conditions. However, it is important to make such simulations as realistic as possible to avoid difficulties when the controller is later used on a real process. For example, the evaluation of the regulatory behavior of a controller must be based on simulations where more than one type of disturbance is introduced.

The simulations which follow are intended to:

a) verify the theoretical analysis of the previous chapters,
b) demonstrate the role of the tuning parameters in modifying the closed-loop behavior,
c) provide a basis for specific recommendations for tuning parameter settings, and
d) evaluate the performance of GPC.

Section 5.1 contains details regarding the organization of the simulations, methods of analysis and the identification routine employed for the self-tuning and open-loop identification runs. The simulations carried out with exact model parameters in section 5.2 indicate the level of performance which can be expected under ideal conditions. Parameter estimation under more realistic conditions, when there are disturbances and unmodelled dynamics, is considered next in section 5.3. The reduced order models identified in section 5.3 are used to evaluate the performance of GPC in the presence of model-plant mismatch in section 5.4.

5.1 Details for Simulation Study

The process transfer functions used to study the behavior of GPC are presented in section 5.1.1. For evaluation of regulatory control performance, two types of disturbances are considered (section 5.1.2). The root locus and frequency response analysis techniques, which augment the time-domain simulations in this chapter, are described in sections 5.1.3 and

5.1.4, respectively. For identification purposes, the variable forgetting factor - constant trace RLS algorithm referred to in section 5.1.5, was used exclusively in this thesis.

5.1.1 Process Transfer Functions

The linear SISO processes listed in Table 5.1 were choosen for the simulation study to represent plants which might be encountered in the chemical process industries. Several of the processes have characteristics which make them difficult to control (for well established techniques).

Table 5.1 Simulated Process Transfer Functions

Process	Continuous Transfer Function	Discrete Pulse Transfer Fuction	Zeros	Poles
A	$\frac{1}{(1+10s)}$	$\frac{.0952z^{-1}}{1-.9048z^{-1}}$	–	.9048
B	$\frac{1}{(1+3s)(1+5s)}$	$\frac{.0280z^{-1}+.0234z^{-2}}{1-1.5353z^{-1}+.5866z^{-2}}$	-.8371	.7165 .8187
C	$\frac{1}{(1+s)(1+3s)(1+5s)}$	$\frac{.00768z^{-1}+.02123z^{-2}+.00357z^{-3}}{1-1.9031z^{-1}+1.1514z^{-2}-.2158z^{-3}}$	-.1798 -2.586	.3679 .7165 .8187
D	$\frac{-1}{(1+5s)(1-3s)}$	$\frac{.03506z^{-1}+.03665z^{-2}}{1-2.2143z^{-1}+1.1426z^{-2}}$	-1.045	.8187 1.396
E	$\frac{-2s+1}{(1+6s)(1+2s)}$	$\frac{-.0864z^{-1}+.1468z^{-2}}{1-1.4530z^{-1}+.5134z^{-2}}$	1.699	.6065 .8465
F	$\frac{2(229)}{(s+1)(s^2+30s+229)}$	$\frac{.03700z^{-1}+.07172z^{-2}+.00785z^{-3}}{1-1.3422z^{-1}+.4455z^{-2}-.0450z^{-3}}$	-.1164 -1.822	.9048 .2187± .0443i

Note: Sampling Interval: T_s=1 for Processes A to E, T_s=0.1 for Process F

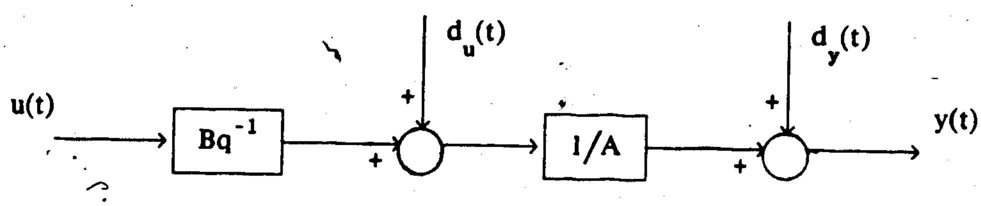
Processes A, B and C are 1st order, 2nd order overdamped and 3rd order overdamped stable processes, respectively. Process B is a benchmark used by several authors in the literature including Vermeer (1987) and Tjokro and Shah (1985). Process C has a nonminimum phase (NMP) zero in the discrete-time domain. The 2nd order Process D is both open-loop unstable and NMP and therefore represents a considerable challenge. Process E is a 2nd order continuous-time NMP plant where the initial response to an input step change is opposite in direction from the steady state response (i.e. "inverse" response). The final process is Rohrs' (1984) benchmark example. This third order plant which includes second order, high frequency dynamics has been used by several authors to study the robustness of self-tuning controllers in the presence of unmodeled dynamics (Cluett et al., 1987).

5.1.2 Types of Disturbances

Although chemical processes are subjected to a wide variety of unmeasurable disturbances, the range of effects is adequately spanned by considering disturbances which enter at the process input and others which manifest themselves directly at the process output. For the purpose of this thesis, isolated, nonzero-mean, step-type disturbances are considered. Letting $d_u(t)$ and $d_y(t)$ represent "input" and "output" step-type disturbances, the difference equation used for the simulations involving regulatory control was

$$Ay(t) = Bu(t-1) + d_u(t) + Ad_y(t) \tag{5.1.1}$$

corresponding to the following block diagram:



Note that the "true" noise polynomial, $C_0(q^{-1})$, in the CARIMA model is equal to 1 for the "input" type disturbance and $A(q^{-1})$ for the "output" type disturbance.

5.1.3 Root Locus Analysis

Rearrangement of the GPC control law into a general linear form, as discussed in Chapter 2, resulted in the characteristic equation:

$$TA\Delta + q^{-1}BS = 0 \tag{5.1.2a}$$

The polynomials $T(q^{-1})$ and $S(q^{-1})$ are functions of all of the tuning parameters of GPC as well as the process model polynomials $A(q^{-1})$ and $B(q^{-1})$. Root loci are plots of the roots of the characteristic equation in the complex z plane as one tuning parameter is varied over a specified range. If the tuning parameter is continuous (e.g. control weighting, λ) the loci are continuous. For tuning parameters which take on only integer values (eg. the maximum output horizon, N_2) the locations of the roots are represented by discrete points.

On the root locus plots of this chapter, open-loop zeros are denoted by an "o" and open-loop poles by a large "□". For discrete plots, a small "□" is used to plot closed-loop pole locations whereas solid lines are used for continuous loci. The unit circle represents the stability boundary. For the smaller root locus plots, only the closed-loop poles are indicated.

When there exists model-plant mismatch, the characteristic equation is written:

$$\hat{T}A\Delta + q^{-1}B\hat{S} = 0 \tag{5.1.2b}$$

where now $\hat{T}$ and $\hat{S}$ are obtained from the estimated model polynomials $\hat{A}$ and $\hat{B}$. (The controller is based on the model $(\hat{A},\hat{B})$ which may differ from the actual process (A,B).)

5.1.4 Frequency Response Analysis

The frequency response of a discrete time system is obtained by setting $z=e^{j\omega T_s}$ in the transfer function description of that system. With the GPC control law in its general linear form, the closed-loop transfer function may be obtained for disturbances entering the loop at arbritrary locations. For example, considering the $d_u(t)$ and $d_y(t)$ disturbances of section 5.1.2, the transfer function description is:

$$y(z) = \frac{BRz^{-1}w(z) + T\Delta d_u(z) + TA\Delta d_y(z)}{TA\Delta + z^{-1}BS} \qquad (5.1.3)$$

where the argument z is used rather than q for the model and control law polynomials. The frequency response is evaluated for ω between 0 and the Nyquist frequency, π/T_s. For Bode plots, it is convenient to consider the independent variable to be the normalized frequency, ωT_s, in radians.

Isermann (1981) refers to the transfer function relating the controlled variable, y(z), to disturbances entering the loop directly at the process output, $d_y(z)$, as the "dynamic control factor" (i.e. $G_{yd}(z^{-1}) = TA\Delta / TA\Delta + z^{-1}BS$). The magnitude (gain) of the dynamic control factor $|G_{yd}|$ indicates how much disturbance are attenuated by the controller. For most controllers, a plot of $|G_{yd}|$ versus ωT_s may be divided into three regions (Isermann, 1981):

Region I: low frequencies; $|G_{yd}| < 1$; disturbances are reduced
Region II: medium frequencies; $|G_{yd}| > 1$; disturbances are amplified
Region III: high frequencies; $|G_{yd}| \simeq 1$; disturbances are unaffected

The effectiveness of the controller is restricted to region I. Isermann also points out that tuning parameter changes of a controller which reduce $|G_{yd}|$ in one region invariably cause the magnitude of the dynamic control factor to increase in another region. (i.e. optimal tuning parameter settings depend on the frequency content of the disturbances encountered).

5.1.5 Recursive Least Squares

The basic RLS algorithm has a number of deficiencies which limit its usefulness for practical identification problems. Many variants of the basic algorithm have been proposed (Shah and Cluett, 1987). For simplicity, in this thesis, parameter estimation was carried out using a RLS algorithm with variable forgetting factor maintaining a constant trace of the covariance matrix (Sripada and Fisher, 1987). (The on/off criteria discussed in the referenced paper were not employed).

5.2 Evaluation of GPC with Exact Process Models

In this section we are interested in the ideal behavior of GPC; the control algorithm is given the exact parameters of the simulated process at each sampling instant. The evaluation of the three configurations (tuning strategies) of GPC detailed in Chapter 4 is based on experience with all of the processes listed in Table 5.1. For the sake of brevity, only those simulations demonstrating the most important characteristics of each configuration, will be presented. The set of simulations with exact process models included in this chapter are summarized in Table 5.2.

Table 5.2 GPC Simulations with Exact Process Models (nonadaptive)

Fig. No.	Process	Time-Varying	Servo	Regul-atory	GPC Config.	Purpose
5.1	C	no	yes	no	OH	effect of N_2
5.2	D	"	"	"	"	"
5.3	C	"	"	"	LW	effect of λ
5.4	D	"	"	"	"	"
5.5	C	"	"	"	DMF	effect of $P(q^{-1})$
5.6	C	yes	yes	no	OH	main. of performance
5.7a	"	"	"	"	LW,λ=const	"
5.7b	"	"	"	"	LW,$\lambda \propto [B(1)]^2$	"
5.8	"	"	"	"	DMF	"
5.9,10	C	no	yes	yes	DMF	effect of $C_c(q^{-1})$
5.11	"	"	"	"	3 cases	2 degrees of freedom

Abbreviations: OH = Output Horizon, LW = Lambda Weighting, DMF = Detuned Model-Following

5.2.1 Output Horizon Configuration

For the Output Horizon configuration of GPC, the active tuning parameter used to alter the speed of response is the maximum output horizon, N_2. Figures 5.1a and 5.1b show the position of the closed-loop poles and corresponding time domain response for the third order Process C. For $N_2=1$, the controller attempts to cancel the open-loop zeros (-.1798, -2.5862); instability results after a finite period of time (not shown). When N_2 is set to 2, three nonzero CL poles close to the origin yield a rapid response. As N_2 is increased, the CL poles tend to the OL poles (.3679, .7165, .8187) resulting in mean-level control. For $N_2 \geq 2$, the OL zeros are retained in the closed-loop transfer function.

It is interesting to consider the same configuration of GPC applied to OL unstable and NMP Process D. For $N_2=1$ we again have "1 atep ahead" control which is not stable due to the presence of the NMP zero (-1.0453). If N_2 is increased to 2 (see Figure 5.2a and b) the CL poles are complex and relatively close to the origin. For larger values of N_2 the response of the system is progressively detuned. As $N_2 \rightarrow \infty$, one CL pole approaches the stable OL pole (.8187) while the other tends to the intersection of the positive real axis and the unit circle. This results in the very sluggish "integrating" response for the last setpoint change of Figure 5.2b.

For an inverse response process it was stated in section 4.2.1 that N_2 must be selected large enough to "look beyond" the inverse response. Runs with Process E indicated that stable control could only be achieved for $N_2 \geq 5$. For this process the step response coefficients which make up the G_r matrix are as follows:

-.0864, -.0652, .0101, .1085, .2129, .3140, ...

Looking at the sum of the first N_2 elements, we find that for $N_2 \geq 5$ this sum is positive whereas for $N_2<4$ it is negative. For this positive gain process, this verifies that for stable control N_2 must be large enough to satisfy:

$$\text{sign}\left[\sum_{j=1}^{N_2} g_{j-1}\right] = \text{sign}(K_p)$$

In summary, for the processes considered, it was straightforward to vary the speed of response using N_2 as the active tuning parameter. Large values of N_2 resulted in stable control in all cases, and such values are recommended for start up under uncertain conditions. Note that the effect of N_2 upon the closed-loop speed of response is nonlinear. For OL stable processes, the effect decreases as N_2 becomes large. For an OL unstable process, the opposite tendency may be observed.

5.2.2 Lambda Weighting Configuration

For this configuration of GPC, the tuning parameters are set up to achieve deadbeat control which is then detuned for practical application using λ control weighting. Figure 5.3a shows root loci as λ is varied from 0 to ∞ while servo responses for four values of λ in this range are plotted in Figure 5.3b, for Process C. No control weighting results in all CL poles being assigned to the origin and a corresponding deadbeat response. As $\lambda \rightarrow \infty$ three CL poles tend toward the OL poles (.3679, .7165, .8187) while a fourth tends toward z=1 giving an "integrating" response. The fourth pole is a consequence of weighting control **increments** in the multistep cost function (necessary so that there will be no offset at steady state). For all values of λ the OL zeros are retained in the closed-loop.

Figures 5.4a and 5.4b show simulation results for OL unstable Process D. Once again λ=0 yields deadbeat control with all CL poles at the origin. For $\lambda \rightarrow \infty$ the CL poles approach the OL poles (.8187, 1.3956) and the point z=1 and, as a result, the system is stable only for values of λ<299. This represents a limitation of the Lambda Weighting configuration; for OL unstable processes a large amount of control weighting, normally considered to be conservative, may lead to instability.

Stable control was achieved for all values of λ applied to Process E since the maximum output horizon N_2, set to 10, is sufficient to "look beyond" the inverse response (results not shown).

5.2.3 Detuned Model-Following

To achieve a practical model reference controller that can be applied to NMP plants, the cancellation of process zeros must be avoided by "detuning" in some manner. For a process without physical delay (d=0), setting $NU=N_2$ results in exact model-following. The first entry in Table 5.3 (located at the end of the chapter together with corresponding Figure 5.5) demonstrates that with $NU=N_2=4$ applied to Process C, two CL poles attempt to cancel the OL zeros (-.1798, -2.5862) one of which is NMP. Note that the other CL pole is exactly equal to that of the desired reference model (0.9). One method of "detuning" is to increase N_2 relative to NU. When N_2 is incremented by 1, the CL pole previously outside the unit circle is "pulled" inside and placed near the origin.

An alternative method of "detuning" exact model-following is to add a small amount of control weighting. Consider the middle two entries of Table 5.3. It is evident that for stable control to be achieved λ must be $> \lambda_0$ where $\lambda_0 \simeq 9\times10^{-6}$. Too large a value for λ would lead to poor tracking of the reference model. Unfortunately, the range of suitable values of λ which prevent cancellation of NMP process zeros without unnecessarily slowing the response of the system are not known *a priori*. It is therefore recommended that exact model-following be "detuned" using N_2 rather than λ.

To account for variable time delays or "inverse response" processes a larger value of N_2 is normally used. With $N_2=10$, the final block of results in Table 5.3 lists the closed-loop pole locations when the reference model is changed. The corresponding time domain servo responses are shown in Figure 5.5. The fairly high level of control activity is due to the fact that we are asking a 3rd order process to behave in a 1st order manner. Specifying a 2nd order desired closed-loop model with a similar rise time results in a substantial reduction in control signal variance.

The Detuned Model-Following configuration of GPC gave stable control for all processes in Table 5.1. By altering the reference model it was possible to obtain a full range of response times. The locations of the CL poles not associated with the reference model depend upon the other tuning parameter settings (i.e. N_1, N_2, NU and λ). These poles are normally close to the origin and have relatively little influence on the CL response.

5.2.4 Maintenance of Servo Performance

For many applications it is desirable that the output response of the closed-loop remain invariant to changing process characteristics. For others, a trade-off between control effort and output performance may be acceptable provided that the response of the controlled variable does not undergo serious degradation. To investigate maintenance of performance for each configuration of GPC, control was exercised over a time-varying third order process. Just prior to each upward-going setpoint change the characteristics of Process C were changed according to Table 5.4, the control algorithm receiving the exact process parameters (including the time delay) at each sampling instant.

Table 5.4 Characteristics of Time-Varying Process C

Transfer Function: $$G_p(s) = \frac{K_p e^{-t_d s}}{(1+s\tau_{p1})(1+s\tau_{p2})(1+s\tau_{p3})}$$

Sampling Interval	K_p	t_d	τ_{p1}	τ_{p2}	τ_{p3}	Description
0-149	1	0	1	3	5	nominal values
150-249	1	3	1	3	5	time delay added
250-349	100	0	1	3	5	gain change
350-449	1	0	1	30	5	dynamics change
450-549	1	0	1	-3	5	unstable process

Note: $T_s=1\ \forall t$ $(d=t_d/T_s)$

5.2.4.1 Output Horizon Configuration

The time domain response for Output Horizon control is shown in Figure 5.6. The response is identical for the third setpoint change compared to the first (involving a process gain change) and very similar to the 2nd and 4th (time delay added and a change in one of the time constants, respectively). Only when the process becomes OL unstable does the output response become significantly more sluggish. Adjustment of N_2 would be required in this case to restore the speed of response to that obtained for the nominal process conditions.

5.2.4.2 Lambda Weighting Configuration

Moving on to the Lambda Weighting configuration consider first the case where constant control weighting is used (Figure 5.7a). $\lambda=1$ gives a "nice" response for the nominal Process C but some overshoot when time delay is added. The higher process gain for the third setpoint change allows the controller to achieve a near deadbeat response since the small control effort required does not weigh in the multistep cost function minimization (eqn. 4.1.1). Since the cost function penalizes control activity, when the time constant of the process increases, larger control actions (required in order to maintain output response time) are not allowed and a sluggish response with large overshoot is obtained.

The problem of maintaining performance in the face of gain and dynamics changes can be largely alleviated by employing the scaling of λ shown to be necessary in section 4.3.2. For nominal Process C, B(1)=.03248 and therefore selecting $\lambda_{rel}=948$ gives $\lambda=\lambda_{rel}[B(1)]^2=1$ and the same response, as for the no scaling case, to the first setpoint change (Figure 5.7b). With this scaling, the output response is completely invariant of the process gain change and almost invariant of all the other process variations. Note that λ itself ranges from .01338 to 10000 in order to achieve this. For nominal values of Process C, trace($G_r^TG_r$) is on the order of 4 so that the value $\lambda=1$ used to start is that which roughly halves the control increments which would have been implemented for deadbeat control (see section 4.3.3).

5.2.4.3 Detuned Model-Following

The servo response for Detuned Model-Following control of the time-varying Process C is shown as Figure 5.8. A critically damped second order reference model was selected with natural period, $\tau=3$ ($P = 12.407 - 17.820q^{-1} + 6.413q^{-2}$). This control strategy results in the output performance being maintained almost perfectly unchanged in spite of the large variations in the process gain and dynamics. (The output response is actually only identical for the case of the process gain change). However, the large control actions taken are the penalty for such excellent maintenance of performance.

In summary, all three configurations of GPC are capable of rendering the output servo response of the closed-loop invariant to process gain changes and insensitive to changing dynamics given an accurate process model at all sampling instances.

5.2.5 Disturbance Tailoring using C_c

The previous simulations dealt with tuning parameters which affect the overall (i.e. servo and regulatory) behavior of the closed-loop (although only servo responses were shown). Here the ability of the controller design polynomial C_c to modify the rejection of disturbances is demonstrated.

Assume, for convenience, that the third order Process C is at steady state with output equal to zero when an unit change in measurement occurs (due to a disturbance of unknown type). Figure 5.9a shows the predicted output trajectory f(t+j), j=1,35 for three different settings of C_c (with P=1). (Note that at this point the type of disturbance has not been specified and therefore the **actual** effect of the disturbance upon the output in the future is unknown. Figure 5.9a simply shows how C_c modifies the **predicted** output trajectory.)

Consider also Figure 5.9b which shows the closed-loop response to setpoint changes and deterministic unmeasured disturbances where the overall GPC tuning parameters have been set for Detuned Model-Following. Thirty sampling intervals after each upward-going setpoint change a step "input" type disturbance ($d_u(t)$) of size 0.1 is applied and removed 40 samples later. This is followed by a step "output" type disturbance ($d_y(t)$) of magnitude 0.2 applied for 30 interations.

C_c=1 is frequently assumed in applications of (self-tuning) control reported in the literature. With this assumption, the model employed by GPC is:

$$y(t) = [B/A]u(t-1) + x(t)/A$$

where $x(t) = \xi(t)/\Delta$ represents a nonstationary load.

The controller expects that all disturbances will have an increasingly large effect on the output in the future (Figure 5.9a). This expectation leads to excellent rejection of the "input" type loads which are perfectly modelled (Figure 5.9b, samples 60 to 120).. However, these strong predictions are unsuitable for disturbances which have fully manifested themselves at the output in a single time step and the controller over-reacts strongly in this case (Figure 5.9b, sample 140 to 180).

With $C_c = A$, the controller is actually making use of the model:

$$y(t) = [B/A]u(t-1) + x(t)$$

with the expectation that the future effect of disturbances on the output will be the same as the current effect (Figure 5.9a). The step response equivalent of this model is used in the basic formulation of Dynamic Matrix Control (Cutler and Ramaker, 1980). As anticipated, the rejection of "input" type disturbances is very sluggish (Figure 5.9b, samples 250 to 330).

As a compromise, $1/C_c$ may be selected as a low pass filter. For example, with $C_c=1-.8q^{-1}$, the controller rejects both types of disturbances in a manner intermediate to that of the two previous cases (Figure 5.9b).

Note that while C_c was used to modify the regulatory response, with no process modelling errors, the servo response remained unaffected by this design polynomial.

It is interesting to also look at the closed-loop frequency response for different settings of the design polynomial C_c. Magnitude Bode plots for the transfer functions,

$$G_{ud}(z^{-1}) = y(z)/d_u(z) = T\Delta/(TA\Delta+z^{-1}BS)$$

$$G_{yd}(z^{-1}) = y(z)/d_y(z) = TA\Delta/(TA\Delta+z^{-1}BS)$$

are shown as Figures 5.10a and 5.10b. It is apparent that $C_c=1$ gives better rejection of both types of disturbances at low frequencies. However, $C_c=1-.8z^{-1}$ or $C_c=A$ detunes the controller and provides better attenuation of high frequency loads. (Note that $d_u(t)$ and $d_y(t)$ are not restricted in form to step disturbances for this frequency response analysis).

5.2.6 Two Degreees of Freedom

In section 4.6, discussion centered on different ways of using "two degrees of freedom" to allow independent servo and regulatory closed-loop behavior. In the absense of modelling errors, each of the three alternatives presented allows specification of servo and regulatory models which are followed closely in the closed-loop. The simulation in Figure 5.11 was carried out with Process C and the same sequence of setpoint changes and disturbances used previously. The polynomials P and C_c and the transfer function F_{sp} followed the schedule:

Sampling Interval	F_{sp}	P	C_c
0-229	1	$(1-.8q^{-1})/.2$	$1-.9q^{-1}$
230-429	$.2/(1-.8q^{-1})$	1	$(1-.8q^{-1})(1-.9q^{-1})$
430-630	$(1-.9q^{-1})/.1$	$(1-.9q^{-1})(1-.8q^{-1})/(.1)(.2)$	1

with the other tuning parameters set for Detuned Model-Following. Note that in each case the servo and regulatory reference "models" are:

$$M_s = .2/(1-.8q^{-1}) \; ; \qquad M_r = (.2)(.1)/(1-.8q^{-1})(1-.9q^{-1})$$

The response to setpoint changes and the rejection of the disturbances is almost identical for the different methods. This simulation will be repeated when there is model-plant mismatch to evaluate the robustness of each method under more realistic conditions.

5.3 Parameter Estimation

In this section the emphasis is shifted to on-line parameter estimation for self-tuning control. In particular, we are concerned with how parameter estimation should be carried out in the presence of disturbances and unmodeled dynamics to yield models suitable for GPC.

5.3.1 Effect of Disturbances

Figures 5.12a and 5.12b present the results of self-tuning control of Process C (with gain K_p=20, so that the B polynomial coefficients are significantly greater than zero) with C_c=C_e=1. The basic tuning parameters of GPC were set for Detuned Model-Following. Steps in the disturbance, $d_u(t)$, of magnitude 0.1 were applied to the process and removed at sample times 160, 200, 240 and 280. Similarly steps in $d_y(t)$ of size 0.2 were added to the output over the period from 400 to 520 iterations. The parameter estimates of the full order model were initialized with $\hat{b}_0$=1 and the rest equal to zero; the covariance matrix was set to 100 times identity. With a constant trace RLS algorithm this implies a large parameter estimation gain at all times. For clarity, only the parameter estimates $\hat{a}_1$ and $\hat{b}_1$ are plotted (with true values a_1=-1.9031, b_1=.4246).

During the initial sequence of setpoint changes, the parameters converge to their true values leading to excellent rejection of the "input" type disturbances. However when the "output" loads hit (for which the true noise polynomial, C_0=A) the parameters become biased resulting in a poor response for the following setpoint change.

For the sake of contrast, Figure 5.13 shows results for the case where both the controller design polynomial, C_c, and estimator filter, C_e, are set equal to A. In this case, the parameters become biased when the $d_u(t)$ disturbances occur (for which C_0=1).

If C_c is selected to be $1-.8q^{-1}$ and used both in the control law and for regressor filtering (i.e. C_e=C_c) the parameters are slightly biased for both types of disturbances but control is, on average, better (see Figure 5.14). Note also that the regulatory performance compares favourably with the equivalent simulation where GPC is given an exact process model (cf. last part of Figure 5.9).

Now consider the use of Extended Least Squares (ELS) identification with *a posteriori* residuals as proxies for the noise and the order of the estimated noise term, nc=3(=na). The results shown graphically in Figure 5.15 apply to the case where the estimated polynomial $\hat{C}$ is used in the control law. For the $d_u(t)$ and $d_y(t)$ disturbances the true noise filters are C_0=1 and C_0=A, respectively. During the first half of the simulation, the plotted estimated parameter, $\hat{c}_1$, hovers around zero. When the steps in

$d_y(t)$ occur, $\hat{c}_1$ tends toward a_1. As frequently noted in the literature, the convergence rate for the $\hat{C}$ parameters is significantly slower than for $\hat{A}$ and $\hat{B}$. It is important to note that the $\hat{A}$ and $\hat{B}$ parameters are unbiased throughout the simulation inspite of the different step disturbances. The $\hat{C}$ polynomial proves to be unsuitable for inclusion into the control law as the resulting closed-loop behavior is very poor.

One might expect that of ELS were used for identification but a fixed polynomial, C_c, was employed in the control law (i.e. in place of $\hat{C}$) the combination would be very effective. This speculation turns out to be the case for the simulation shown as Figure 5.16, with $C_c=1-.8q^{-1}$. Using ELS, the $\hat{A}$ and $\hat{B}$ parameters are unbiased by the $d_u(t)$ and $d_y(t)$ disturbances and, as a result, the servo performance is excellent throughout. The selection of C_c represents a compromise so that both "input" and "output" disturbances are rejected in a satisfactory manner. Note that this run is almost indistinguishable from the corresponding non-adaptive simulation with no MPM (cf. last part of Figure 5.9).

5.3.2 Reduced Order Models

All real processes are of high order but, for practical reasons, must be represented by low order models. Parameter estimation must be conducted in the presence of unmodelled dynamics in such a way as to yield good predictions over the long-range prediction horizon. In general, this means that it is important for the model to match the process closely at relatively low frequencies (frequencies around the desired closed-loop bandwidth).

Consider Rohrs' third order process with a first order model. To obtain converged or "tuned" parameter estimates an open-loop identification test was performed using a PRBS input (sequence length = 511, amplitude = 1). A PRBS signal was selected to ensure all modes of the process were excited as would typically be the case under closed-loop control. Tuned parameter sets are listed in Table 5.5 for RLS with the regressor filtered by $1/C_e$. They are also tabulated for ELS with different specifications for the order of the estimated noise polynomial, nc.

Table 5.5 Estimated First Order Models for Rohrs' Process (T_s=0.1)

Algorithm	Settings	Parameter Estimates $\hat{a}_1$	$\hat{b}_0$	Notes
RLS	$C_e=1$	-.65	-.012	
"	$C_e=1-.9q^{-1}$	-.94	-.033	
"	$C_e=(1-.9q^{-1})^2$	-.93	.145	
ELS	nc=1	-.31	-.003	large variance
"	nc=2	-.18	-.002	in parameters
"	nc=3	-.20	-.002	(esp. $\hat{C}$)

The parameter estimates obtained using ordinary RLS (C_e=1) and ELS are such that the sign of the estimated process gain is incorrect! Stable control would not be possible if the controller employed any of these models. The Extended Least Squares method does not, by itself, ensure that a reduced order model with a good low frequency fit will be obtained.

The frequency response of the full order process and each of the reduced order models identified using RLS with regressor filtering is shown in Figure 5.17a. Since an incremental model is used for parameter estimation, the input and output measurements are actually filtered by Δ/C_e, the frequency response of which is plotted in Figure 5.17b. (Note that it is only the relative gain at different frequencies that is important; the frequency responses shown are actually for $C_e(1)\Delta/C_e$ so that the curves coincide at low frequencies). With C_e=1, the differencing operator, Δ, amplifies high frequencies. This explains why the corresponding first order model fits the third order process best at the Nyquist frequency. With $C_e=1-.9q^{-1}$, Δ/C_e is a high pass filter and again the estimated model provides the best match with the true process at relatively high frequencies. For the 2nd order design polynomial $C_e=(1-.9q^{-1})^2$, Δ/C_e is a bandpass filter and the model obtained closely matches the true process at all but the highest frequencies. This model is clearly the "best" for long-range predictive control and will be used in the next section for control with a reduced order model. (Note that the dominant OL pole of Rohrs' process is 0.9048. The roots of the first and second order C_e polynomials were selected such that the cut-off frequency of the filters coincided with that of the process.)

Rohrs' process is predominantly 1st order with only high frequency 2nd order dynamics. As such, regressor filtering allows one to almost completely mask out the unmodeled dynamics. For a more difficult problem consider estimating a first order model for the third order Process C with the same PRBS input sequence. Tuned parameter sets may be found in Table 5.6 for various regressor filter polynomials.

Table 5.6 Estimated First Order Models for Process C

Algorithm	Settings	Parameter Estimates $\hat{a}_1$	$\hat{b}_0$
RLS	$C_e=1$	-.80	-.0075
"	$C_e=1-.9q^{-1}$	-.97	.009
"	$C_e=(1-.9q^{-1})^2$	-.94	.080

The unit step response for each of the identified reduced order models may be compared with that of the true process in Figure 5.18a. Note that a reasonable low frequency model is obtained only with the 2nd order C_e polynomial.

While the step response of Process C is clearly not that of a first order process with no delay, it could be interpreted as originating from a first order process with a small time delay (e.g. in a noisy industrial environment by a practicing control engineer). Under these conditions an overparameterized B polynomial together with a first order A polynomial is a reasonable choice for the model structure. Table 5.7 contains tuned parameter estimate for nb=2 and na=1 with corresponding unit step responses shown in Figure 5.18b.

Table 5.7 Estimated First Order plus Time Delay Models for Process C

Algorithm	Settings	Parameter Estimates			
		$\hat{a}_1$	$\hat{b}_0$	$\hat{b}_1$	$\hat{b}_2$
RLS	$C_e=1$	-1.10	.0035	.022	.017
"	$C_e=1-.9q^{-1}$	-.97	.005	.027	.029
"	$C_e=(1-.9q^{-1})^2$	-.91	.012	.023	.065

The models identified with ELS and different C polynomial orders were unsuitable. In all cases results were similar to those obtained using RLS with no regressor filtering ($C_e=1$) and therefore are not tabulated.

The first order + time delay models (Figure 5.18b) fit the true step response better (at least for the first 10 sampling intervals) than the corresponding models obtained with only b_0 estimated (Figure 5.18a). For example, with no regressor filtering the sign of the gain for the first order model is wrong; stable control is not possible for GPC using this model. However, when the $\hat{B}$ polynomial is overparameterized an unstable model is identified which provides adequate predictions for small output horizons. Simulations indicated that for most GPC controller settings this model allowed stable, although oscillatory, control.

5.3.3 Specification of C_e

The role of the estimator filter, C_e, is to attenuate high frequency signals, including measurement noise and unmodeled dynamics. The differencing operator, Δ, serves to eliminate d.c. offsets and low frequency loads. Together, Δ/C_e focuses parameter estimation on the relevant frequency range. There are two aspects to the choice of C_e: selecting the order of C_e and specifying the root(s) of C_e.

An extensive theoretical analysis of RLS estimation with regressor filtering (Mohtadi, 1988) leads to the conclusion that the overall weighting on the prediction error is given by:

$$\left|\hat{A}\Delta/C_e\right|^2,$$

This indicates that if the weighting is to "roll-off" at high frequencies, C_e should be of degree $\geq$ na+1. However, the simulation and experimental results presented in this work and elsewhere (Lambert,E.P., 1987; Lambert,M., 1987) do not indicate the need for a C_e filter of higher degree than two.

With $\delta C_e=2$, Δ/C_e constitutes a bandpass filter (see Figure 5.17b). Bandpass filtering of the I/O data has been recommended by several authors including Sripada and Fisher (1987), Middleton et al. (1988) and Wittenmark (1988). The filter should pass frequencies around the desired closed-loop bandwidth and attenuate higher and lower frequencies (Wittenmark, 1988). One rule of thumb governing design of the estimator bandpass filter is that the upper break frequency should be about twice the desired closed-loop bandwidth and the lower break frequency should be about one tenth the desired bandwidth of the closed-loop system (Middleton et al., 1988).

However, if C_e is restricted in form to a 2nd order polynomial with repeated roots (i.e. $C_e=(1-c_e q^{-1})^2$), the number of parameters which must be specified by the user is minimized. The double breakpoint of this filter is at $-\ln(c_e)/T_s$. Many authors recommend sampling a process at an interval of approximately 1/10 of the dominant open-loop time constant. Typically the closed-loop time constant is specified to be approximately one half of the open-loop time constant. The desired closed-loop bandwidth is then in the range of around $1/5T_s$ rad/s. A value of $c_e \simeq 0.8$ places the double breakpoint of $1/C_e$ at the desired closed-loop bandwidth. This value has been suggested as a default in the literature (Lambert,M., 1987).

5.4 Evaluation of GPC in the Presence of Model-Plant Mismatch

For practical applications of non-adaptive or self-tuning GPC the model used in the design of the controller will not generally correspond exactly to the process being controlled. It is important, therefore, that the control system be insensitive or robust with respect to these modelling errors. In this section the simulations are concerned with demonstrating the effect of the basic tuning parameters, and in particular the controller design polynomial C_c, on the robustness of Generalized Predictive Control. The "best" reduced order models identified in the previous section will be employed. The set of simulations with model-plant mismatch are summarized in Table 5.8.

Table 5.8 GPC Simulations with Model-Plant Mismatch (nonadaptive)

Fig. No.	Process	Servo	Regulatory	GPC Config.	Purpose
5.19	F	yes	no	OH	Effect of C_c
5.20	"	"	"	LW	Effect of λ
5.21	"	"	"	LW	Effect of C_c
5.22	"	"	"	DMF	"
5.23,24	C	yes	no	OH	Effect of C_c
5.25	C	yes	yes	3 cases	2 degrees of freedom

Abbreviations: OH = Output Horizon, LW = Lambda Weighting, DMF = Detuned Model-Following

Notes: time invariant 3rd order processes
reduced (1st) order models identified off-line (section 5.3.2)

5.4.1 Output Horizon Configuration

For Rohrs' process the "best" reduced order model (ROM) (identified using RLS with regressor filtering) was:

$$\hat{A} = 1-.93q^{-1} \qquad \hat{B} = .145$$

Using this model and with the basic tuning parameters set for Output Horizon control the closed-loop system is stable only for $N_2 \geq 34$. However, even for values of N_2 tending to infinity the closed-loop response to setpoint changes or disturbances is very oscillatory. The addition of the design polynomial, $C_c = 1-.9q^{-1}$ or $C_c = (1-.9q^{-1})^2$ stabilizes the system for smaller values of N_2 and yields satisfactory closed-loop behavior. The servo response for N_2=10 is shown in Figure 5.19 along with z-plane graphs indicating locations of the nonzero closed-loop poles. For comparison, the response of the system when the controller is given the exact full order model (FOM) is included. Note that the control signal was clamped to lie within ± 5 units.

5.4.2 Lambda Weighting Configuration

Figure 5.20 applies to the case where the basic tuning parameters are given values appropriate for Lambda Weighting control. The closed-loop system using the same ROM is stable for $\lambda \geq .16$ with a smooth servo response when $\lambda \geq 10$. Note that trace($G_r^T G_r$)/NU = 4.54; lambda values of the same order of magnitude, recommended earlier for startup, would lead to a stable closed-loop.

Lambda was set to the relatively low value of unity to demonstrate the use of the design polynomial C_c in improving the behavior of the closed-loop. As shown in Figure 5.21, adding a first or second order C_c polynomial with roots at 0.9 damped the output oscillations. For completeness the corresponding response with no MPM and C_c=1 is included.

5.4.3 Detuned Model-Following

In the presence of model-plant mismatch, using a first order P polynomial [$P = (1-p_1q^{-1})/(1-p_1)$] the closed-loop is stable provided that $p_1 \geq 0.91$. However, for this and other desired closed-loop models with slower poles, high frequency oscillations remain. With p_1=0.8, Figure 5.22 again demonstrates the use of C_c for the purpose of making GPC robust to the unmodeled dynamics. Note that the choice of $C_c=1-.9q^{-1}$ yields better performance than $C_c=(1-.9q^{-1})^2$.

5.4.4 Use of C_c in Achieving Robustness

The previous simulations with Rohrs' process demonstrated the use of C_c in providing robustness for high frequency unmodeled dynamics. A second example, of this important property, for the more difficult case of low frequency MPM, would be beneficial. Recall the first order model (identified using RLS with regressor filtering) which best fit the step response of Process C was

$$\hat{A} = 1-.94q^{-1} \qquad \hat{B} = .080$$

In this case there is more severe model-plant mismatch than for Rohrs' example. Output Horizon control using this ROM and with $N_2=10$ proves to be unstable. However, either $C_c=1-.9q^{-1}$ or $C_c=(1-.9q^{-1})^2$ yields a stable system as shown in Figure 5.23. With the second order selection of C_c the closed-loop response is slightly more detuned. Note that when GPC receives the full order model an excellent response is possible without the use of C_c in the control law.

A first order model which better matches the step response of Process C was identified with B "overparameterized":

$$\hat{A} = 1-.91q^{-1} \qquad \hat{B} = .012+.023q^{-1}+.065q^{-2}$$

Although the closed-loop is stable for $C_c=1$ using this ROM, when a first or second order C_c polynomial is added the response is improved considerably (Figure 5.24).

5.4.5 Specification of C_c

The controller design polynomial, C_c, influences both the robustness and disturbance rejection properties of the closed-loop system. As a results of its dual role, the specification of C_c represents a trade-off between rapid elimination of disturbances and sensitivity to measurement noise and unmodeled dynamics.

Combining eqn. (2.3.16) with (4.5.15) the following closed-loop expression for u(t) may be obtained:

$$u(t) = \frac{R'Aw(t)}{T'A\Delta + q^{-1}BS'} - \frac{SC_0x(t)}{C_c[T'A\Delta + q^{-1}BS']} \tag{5.4.1}$$

(where the primes denote polynomials corresponding to $C_c=1$)

The transfer function S/C_c affects the rejection of disturbances and MPM.

Since δS=na, this analysis leads to speculation that the degree of C_c should be $\geq$ na to ensure reasonable high frequency properties (Mohtadi, 1988).

Several factors seem to point to the specifiction of the degree of C_c equal to the order of the process model, na. First, recall from section 5.2.5, that with C_c=A the future effect of residuals is predicted to be the same as the current effect. The Dynamic Matrix Controller, making use of this disturbance model, has proven to be very robust in industrial applications (Cutler and Ramaker, 1980). GPC takes on similar robust characteristics if δC_c=na and the root(s) of C_c are selected to coincide with the dominant open-loop time constant. Secondly, with δC_c=na, the predicted output trajectory for a change in the residual (ie f(t+j, j=1,...) is normally a monotonically increasing function of the prediction horizon, j. (This may not be the case when δC_c>na.) Thus, the form of the prediction of the future effect of a disturbance meets intuitive expectations. Finally, for some of the simulations and experimental trials conducted in this thesis, better results were obtained with δC_c=na than with a C_c polynomial of higher or lower degree (see section 5.4.3, for example).

Much of the discussion in section 5.3.3 (regarding the specification of the root(s) of C_e) also applies to the specification of the root(s) of $C_c(q^{-1})$. For simplicity, C_c can be choosen as a stable polynomial with repeated roots:

$$C_c = (1-c_1q^{-1})^{na}$$

and c_1 set to 0.8 or 0.9 to start with. Once the controller has been commissioned, c_1 may be adjusted to fine tune the regulatory response.

Remark: Although C_c has been specified as a monic polynomial (ie $C_c(0)$=1) this is not a requirement. It is the location of the root(s) of C_c rather than the steady state gain, $C_c(1)$, that is of importance. To see this, note that if C_c is multiplied by an arbitrary scalar, k, the polynomials G_j=E_jB and F_j determined from the Diophantine identity (2.2.16):

$$kC_cP = kE_jA\Delta + q^{-j}kF_j$$

are also scaled by k. From eqn. (2.2.20),

$$kG_j = \tilde{G}_j kC_c + q^{-j} k\overline{G}_j$$

it is evident that $\tilde{G}_j$ is unaffected by k whereas $\overline{G}_j$ is proportional to k. Since $\overline{G}_j$ and F_j enter the control law through the open-loop j-step ahead prediction of the output,

$$f(t+j) = \frac{kF_j\, y(t)}{kC_c} + \frac{k\overline{G}_j \Delta u(t-1)}{kC_c} = \frac{F_j\, y(t)}{C_c} + \frac{\overline{G}_j \Delta u(t-1)}{C_c}$$

only as ratios with C_c, the predictions are independent of the scaling, k.

5.4.6 Independent Servo and Regulatory Objectives

While the importance of using the design polynomial C_c for robustness has been demonstrated, the observant reader may wonder whether it is possible to achieve similar results using different settings of the other tuning parameters, in particular P and F_{sp}. The product of C_c and P enters the control law through the Diophantine equation (2.2.16), the essential difference being that $1/C_c$ filters the past control increments and past and present measurements in the prediction (eqn. 2.2.22) and thus ends up in the servo numerator of the closed-loop transfer function based on the equivalent general linear form of the control law (eqn. 4.5.16). Thus by augmenting P with a term which is also added to the setpoint prefilter, F_{sp}, a similar effect to using C_c may be achieved. This is in fact the third alternative for using "two degrees of freedom" to achieve independent servo and regulatory objectives.

The simulation conducted in section 5.2.6 was repeated with the controller based on the first order + time delay model of Process C:

$$\hat{A} = 1-.91q^{-1} \qquad \hat{B} = .012+.023q^{-1}+.065q^{-2}$$

so that now there is MPM. Figure 5.25 shows the response to the same sequence of step setpoint changes and disturbances with F_{sp}, P and C_c taking on the same values for each portion of the simulation. Note that the time scale has been adjusted and the control horizon, NU, is different from the simulation with no MPM.

The most important observation is that similar closed-loop performance is achieved for the alternative tuning parameter settings as suggested by the analysis in section 4.6. The closed-loop pole locations (not shown) are almost the same. If the basic tuning parameters were set for Exact rather than Detuned Model-Following, the CL poles (and zeros) and time domain response would be identical (although the system is unstable since the zeros of the ROM are NMP).

To recap, each of the alternatives discussed earlier for utilizing two degrees of freedom yield similar results for the same regulatory and servo reference "models". Whatever alternative is used, it is **strongly** recommended that the regulatory model contain an additional term used to tailor the response to disturbances and provide robustness to MPM. It is convenient, but not necessary, that C_c be used for this purpose, corresponding to the first alternative presented in section 4.6.

5.5 Summary

The time-domain simulations, root locus analysis and frequency response analysis enhanced the understanding of the GPC algorithm. The following conclusions and recommendations are derived from the results presented in this chapter:

a) In the absence of MPM, the three configurations (tuning strategies) of GPC yield excellent control over a wide variety of processes, including those that are nonminimum phase and/or open-loop unstable. The active tuning parameter can be used to vary the speed of response over a wide range.

b) If fixed λ control weighting is used, the output performance may be seriously degraded if there are large changes in the process gain or dynamics. However, if λ is proportional to $[B(1)]^2$ or if the Output Horizon or Detuned Model-Following configurations are used, the closed-loop response is almost completely invariant of these changes.

c) The controller design polynomial $C_c(q^{-1})$ can be used to achieve acceptable regulatory behavior for the range of disturbances acting on the process under consideration. Even more importantly, C_c dramatically improves the robustness of the controller to model-plant mismatch without detuning the setpoint tracking properties of the system. It is recommended that C_c be selected as a polynomial of order na (i.e. $C_c=(1-c_1q^{-1})^{na}$, where na is the degree of $\hat{A}(q^{-1})$. For reasonable sample times, a value of c_1 equal to 0.9 represents a conservative setting while 0.8 often yields an acceptable trade-off between robustness and performance. As the root(s) of C_c tend toward zero, disturbances are rejected more rapidly with stronger control action.

d) The estimator filter $C_e(q^{-1})$ should be utilized to focus the estimator on frequencies around the closed-loop bandwidth and reduce parameter drift for unmeasured disturbances. It is strongly recommended that C_e be selected as a 2nd order polynomial (i.e. $C_e=(1-c_eq^{-1})^2$) which leads to bandpass filtering of the I/O data in the incremental regressor. In this case, inspite of nonzero mean disturbances and unmodelled dynamics, it is possible to identify low frequency models, which provide good long-range predictions. For most applications the roots of C_e may be selected as 0.8 or 0.9.

Note: the reader interested only in GPC may wish to skip to Chapter 7 where an experimental evaluation is presented.

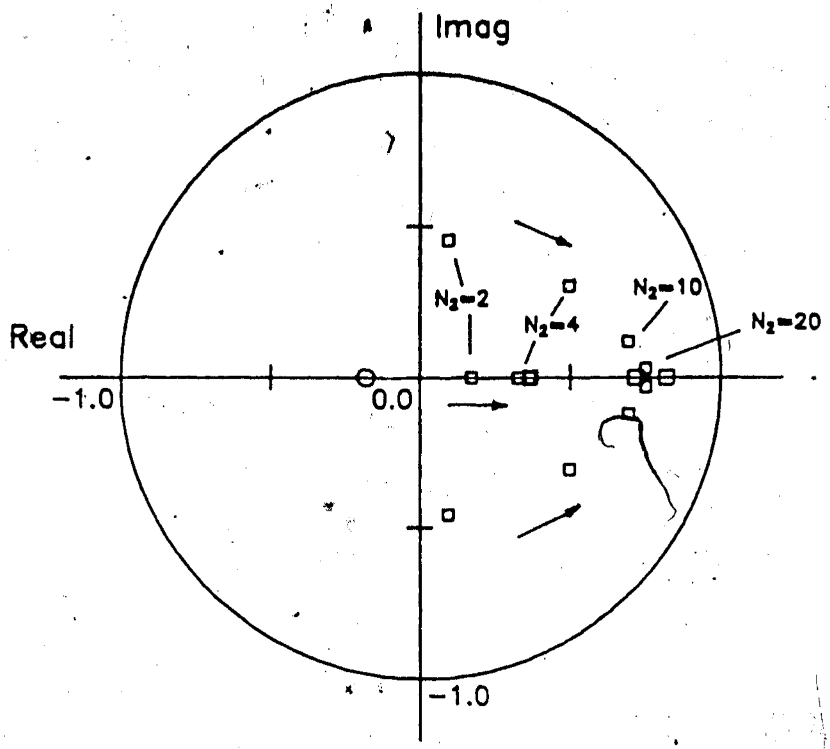


Figure 5.1a Root Loci for Output Horizon Control of Process C

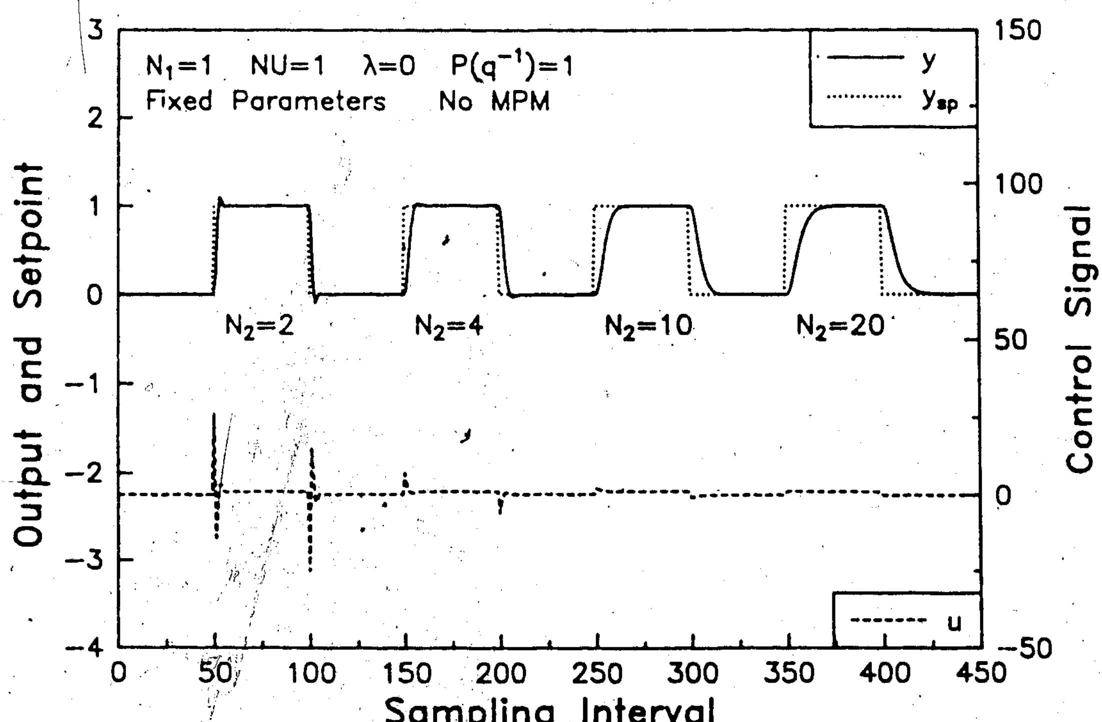


Figure 5.1b Servo Responses for Output Horizon Control of Process C

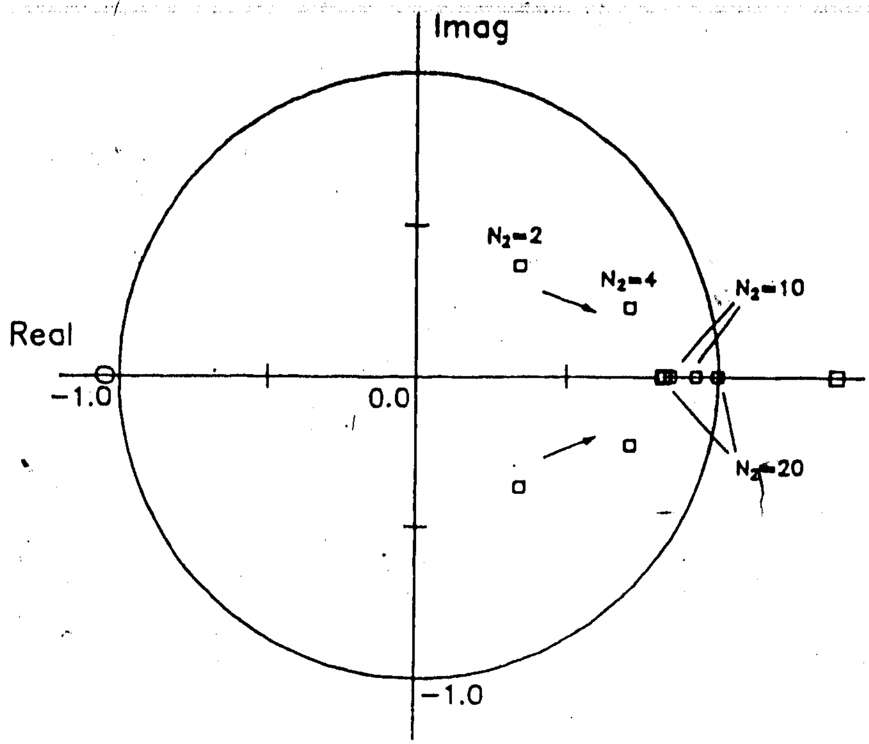


Figure 5.2a Root Loci for Output Horizon Control of Process D

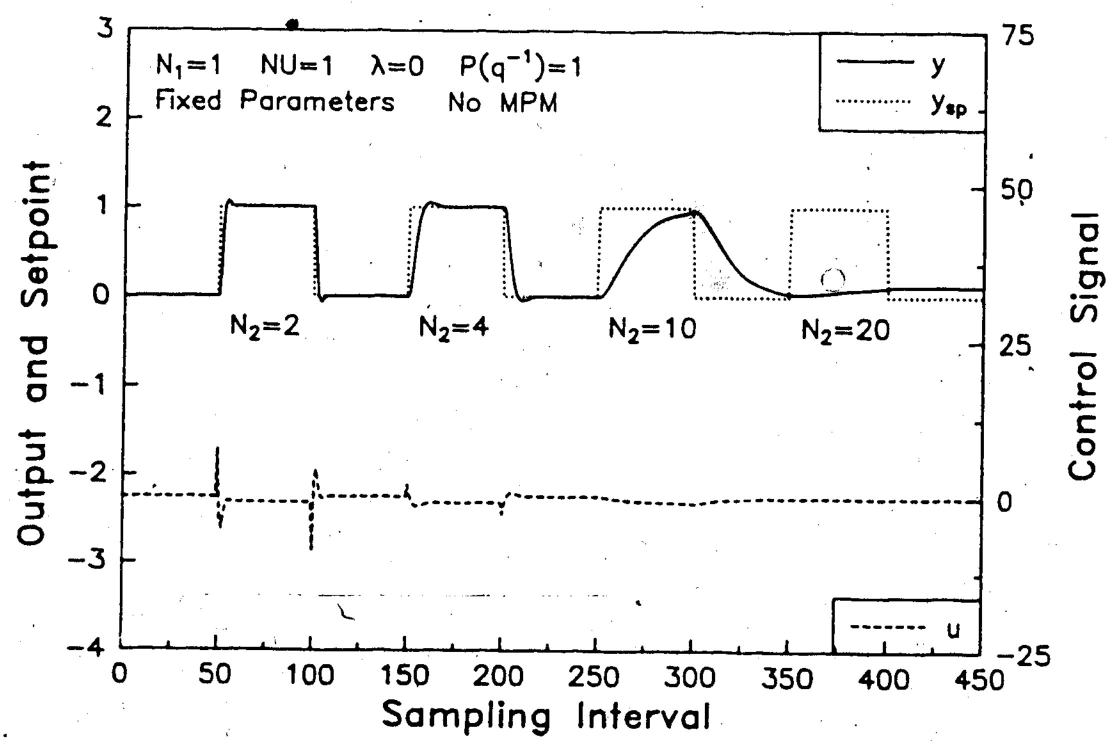


Figure 5.2b Servo Responses for Output Horizon Control of Process D

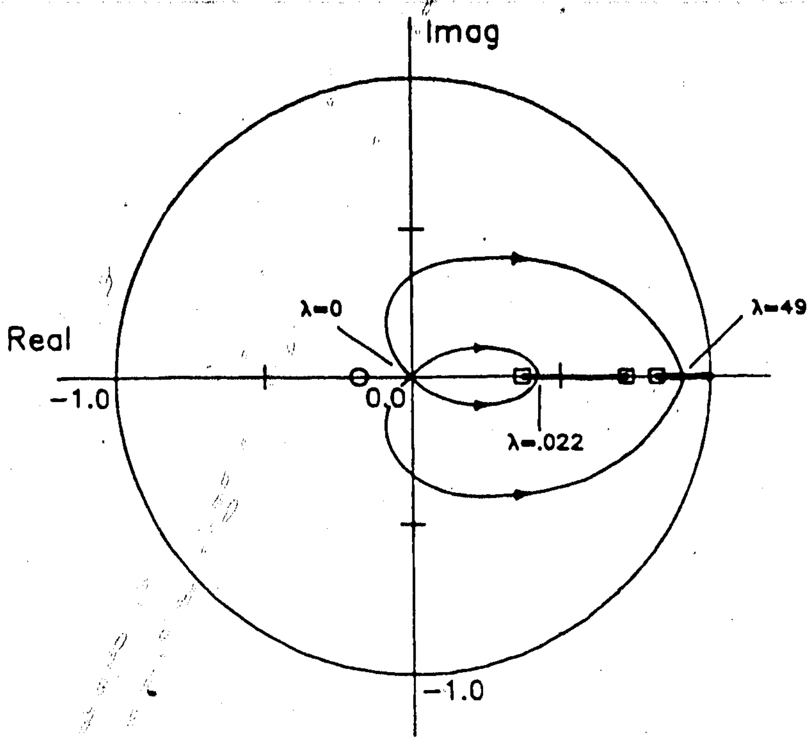


Figure 5.3a Root Loci for Lambda Weighting Control of Process C

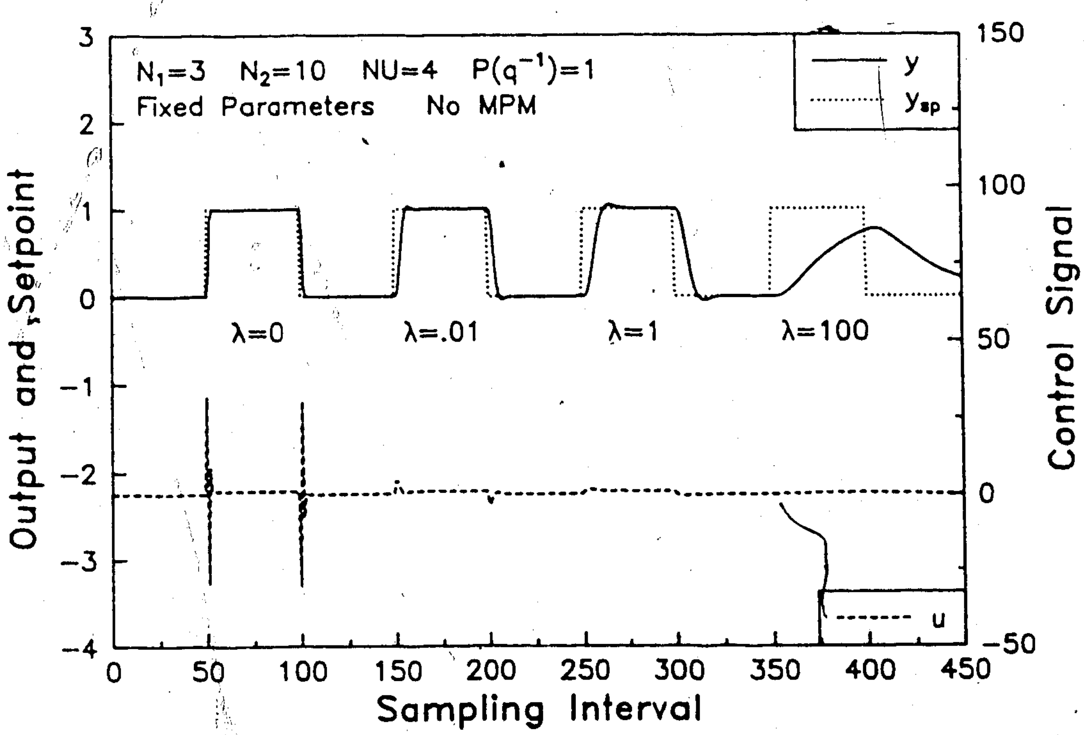


Figure 5.3b Servo Responses for Lambda Weighting Control of Process C

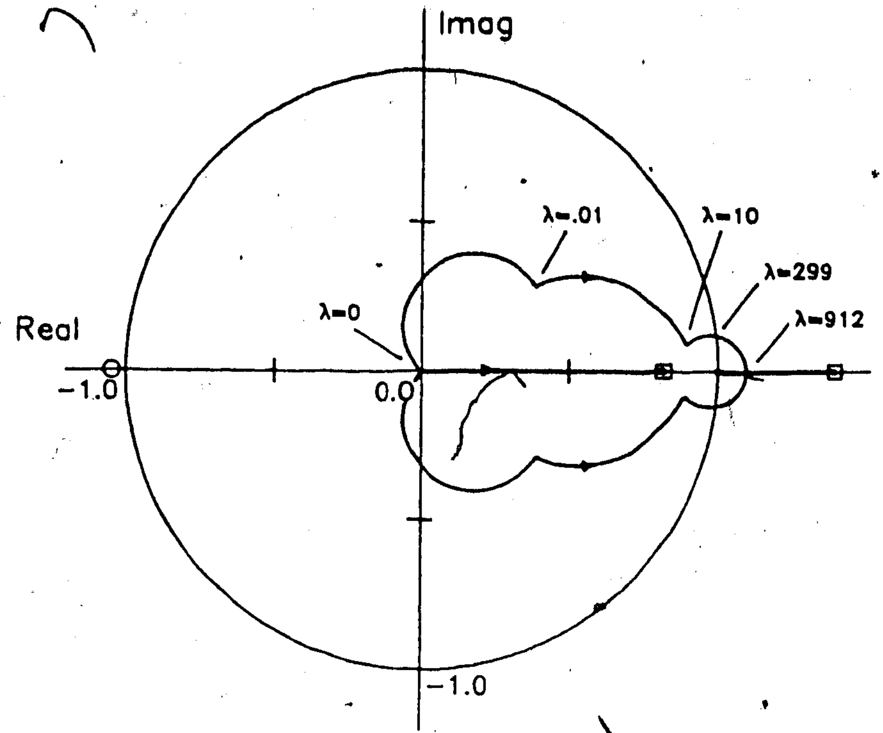


Figure 5.4a Root Loci for Lambda Weighting Control of Process D

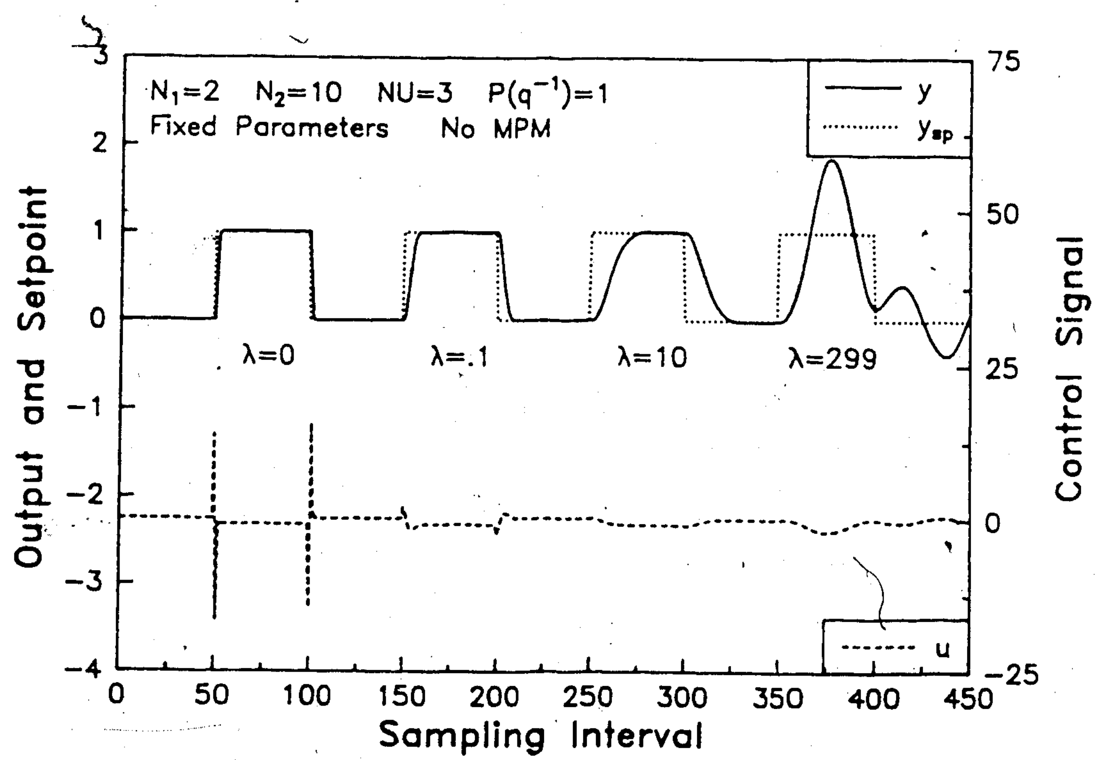


Figure 5.4b Servo Responses for Lambda Weighting Control of Process D

Table 5.3 Closed-loop Poles for Detuned Model-Following Control of Process C

N_2	NU	λ	p_1	Closed-loop Poles
4	4	0	0.9	0.9000, -2.5862, -0.1798
5	4	0	0.9	0.9000, -0.3821, -0.1799
4	4	10^{-6}	0.9	0.9000, -2.0574, -0.1803
4	4	10^{-5}	0.9	0.9000, -0.8820, -0.1853
10	4	0	0.0	0.0000, -0.2252 ± 0.0399i
"	"	"	0.6	0.6004, -0.3458, -0.1835
"	"	"	0.8	0.7999, -0.3521, -0.1827
"	"	"	0.9	0.9004, -0.3545, -0.1824

Notes: $P = 1 - p_1 q^{-1}/(1-p_1)$, $N_1 = 1$

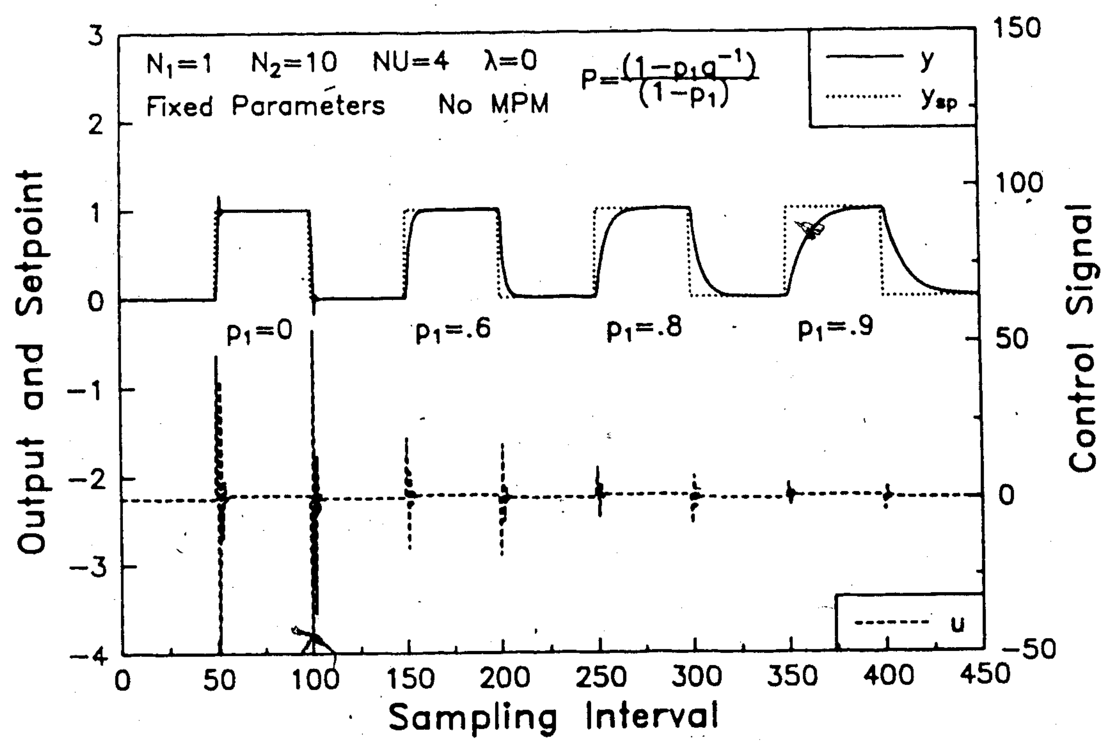


Figure 5.5 Servo Responses for Detuned Model-Following Control of Process C

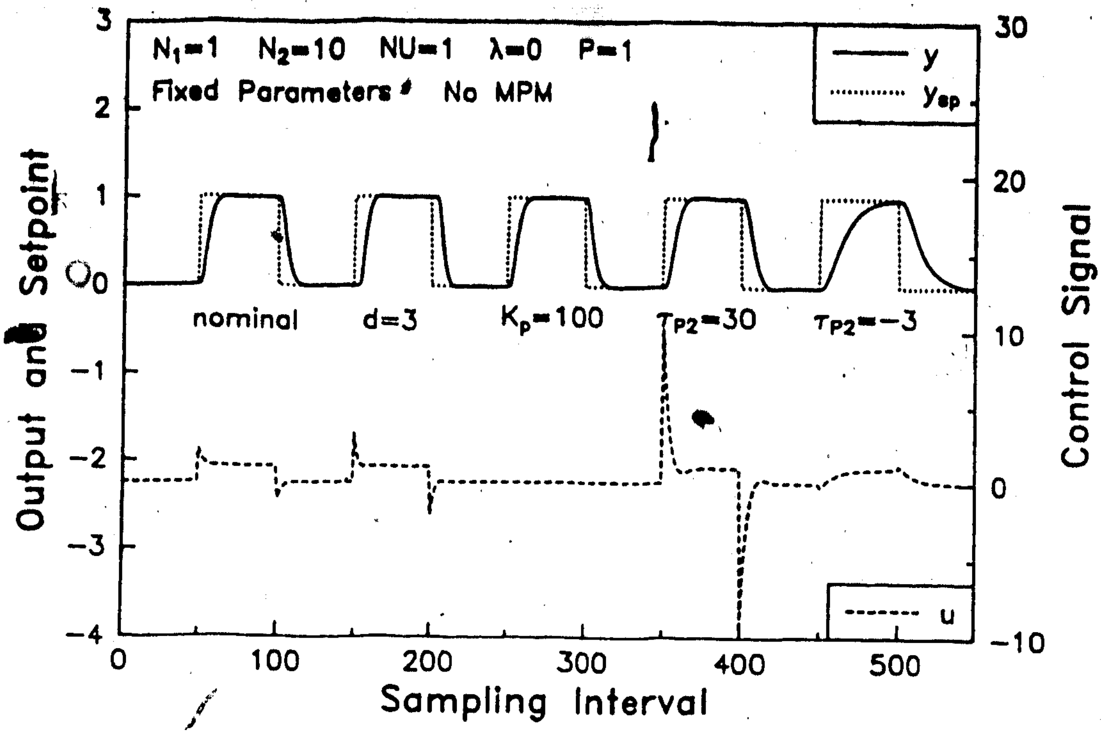


Figure 5.6 Servo Performance for Output Horizon Control of Time–Varying Process C

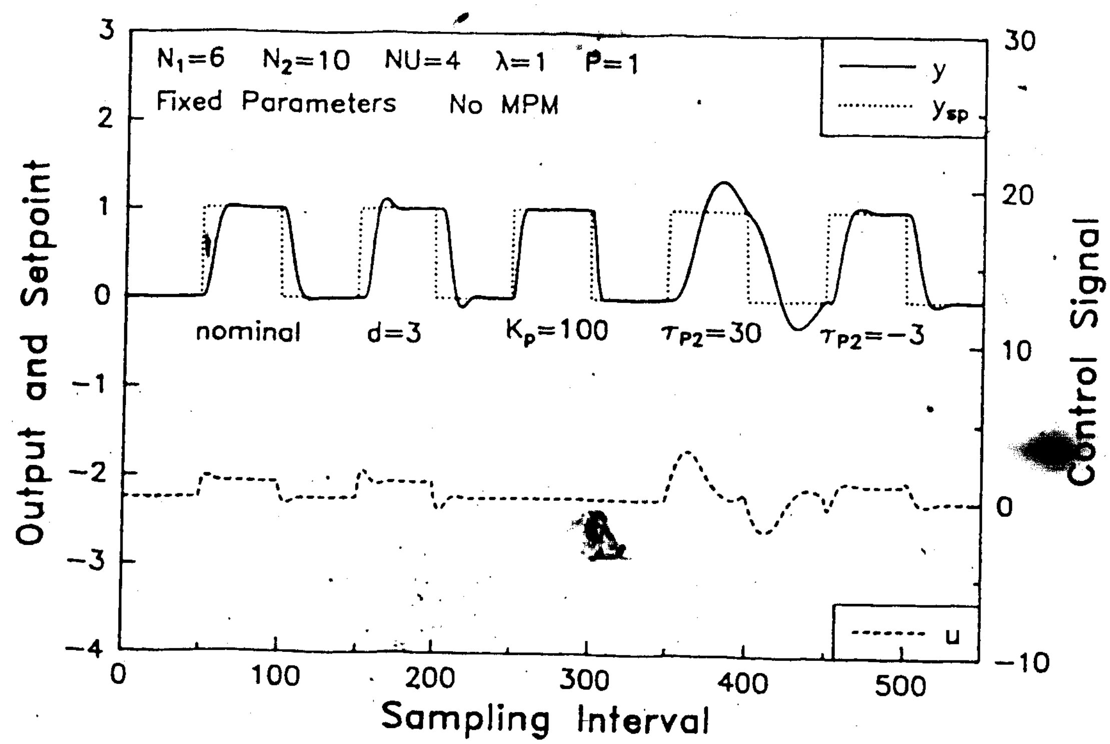


Figure 5.7a Servo Performance for Lambda Weighting Control of Time–Varying Process C (No Scaling)

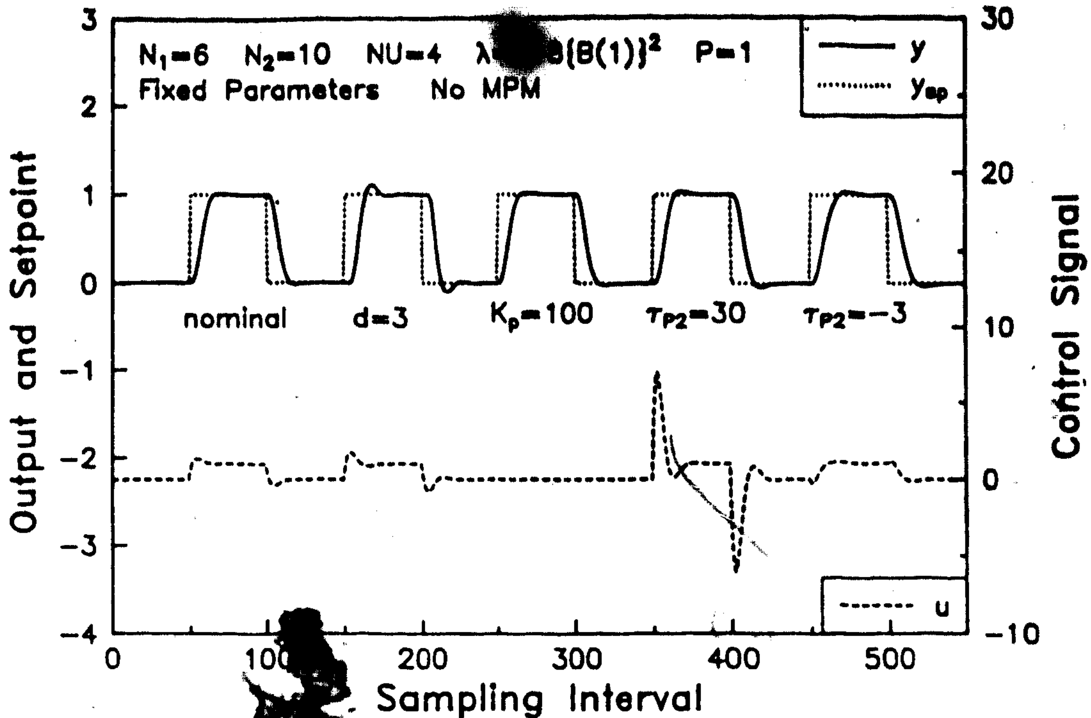


Figure 5.7b Servo Performance for Lambda Weighting Control of Time–Varying Process C (With Scaling)

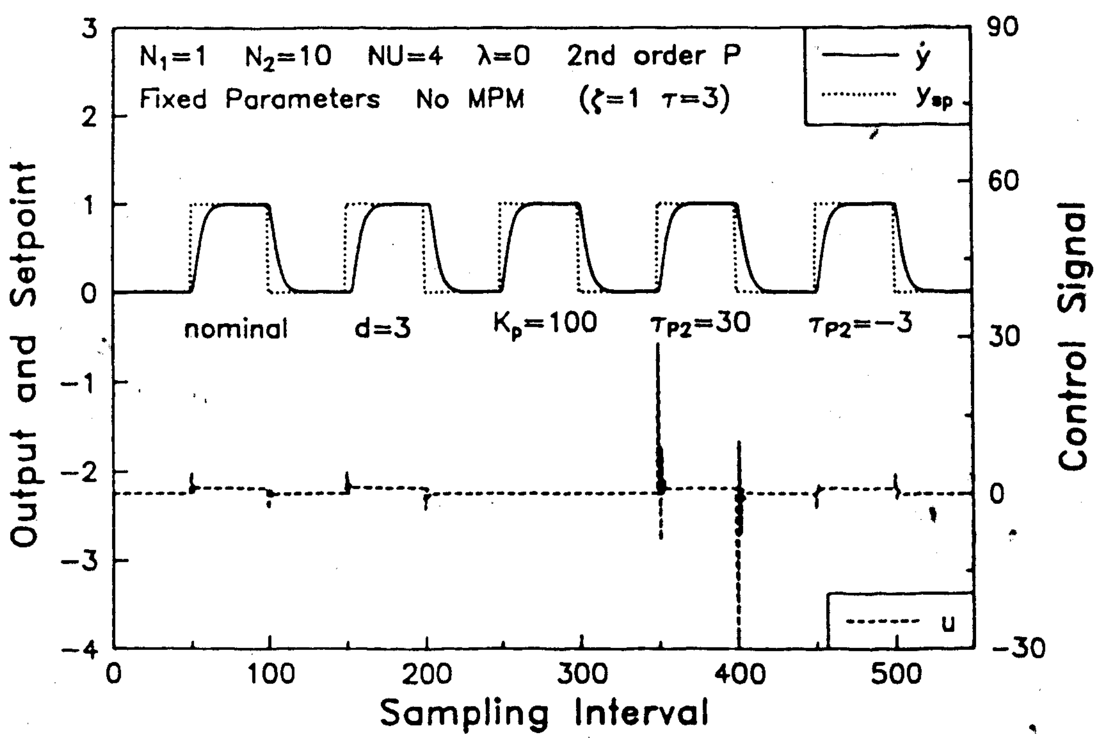


Figure 5.8 Servo Performance for Detuned Model–Following Control of Time–Varying Process C

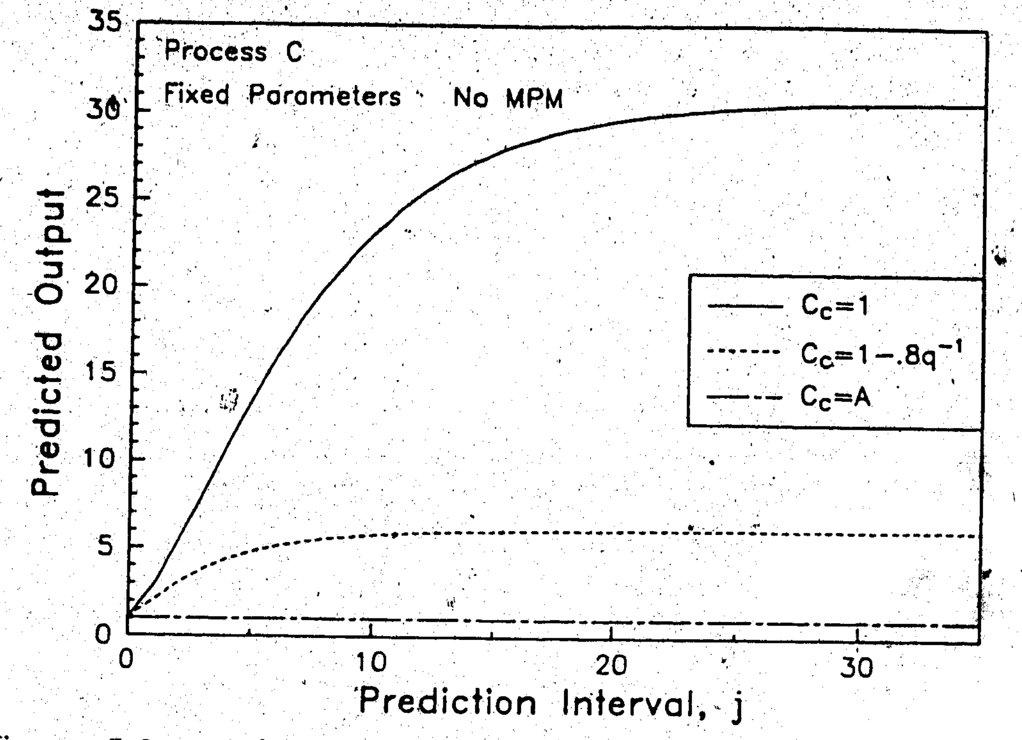


Figure 5.9a Influence of C_c Design Polynomial on Predicted Output Trajectory for a Unit Disturbance

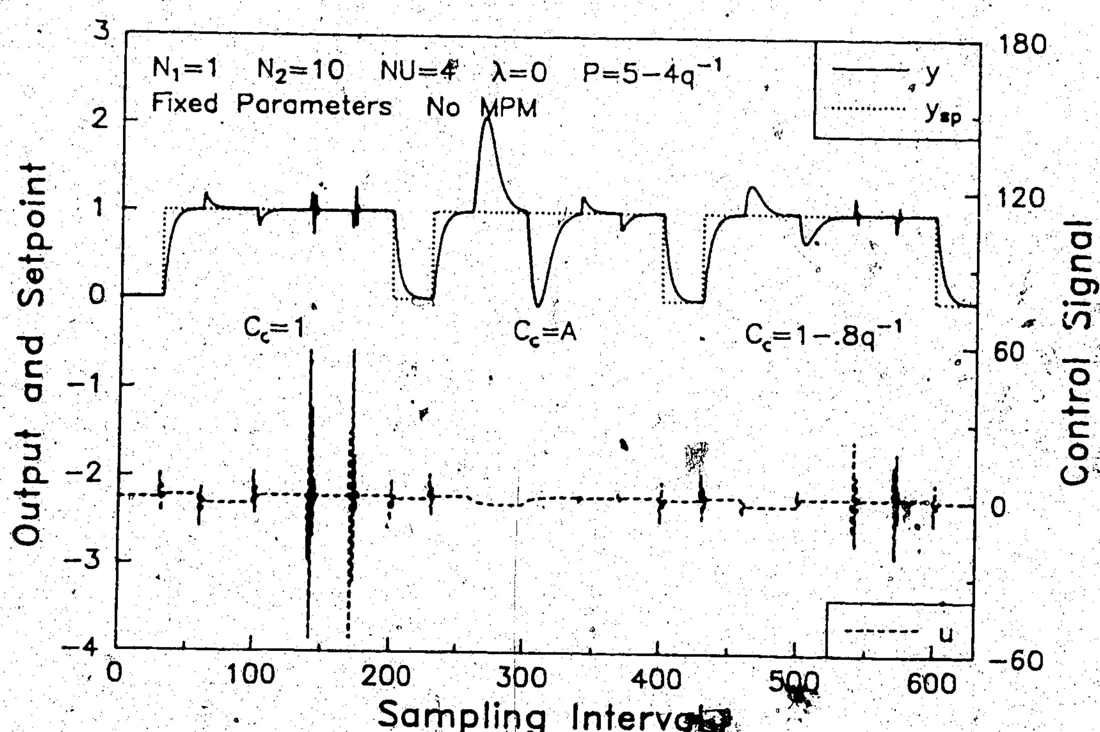


Figure 5.9b Effect of C_c Design Polynomial on Disturbance Rejection for GPC Control of Process C

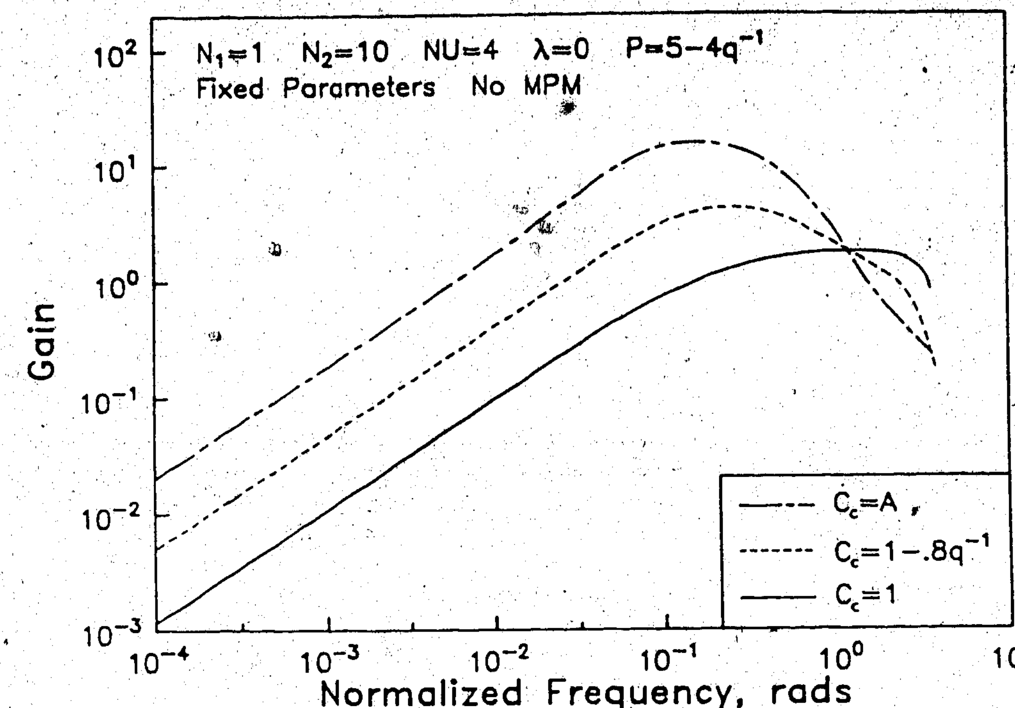


Figure 5.10a Frequency Response of Transfer Function $G_{ud}(z^{-1})$ arising from GPC Control of Process C

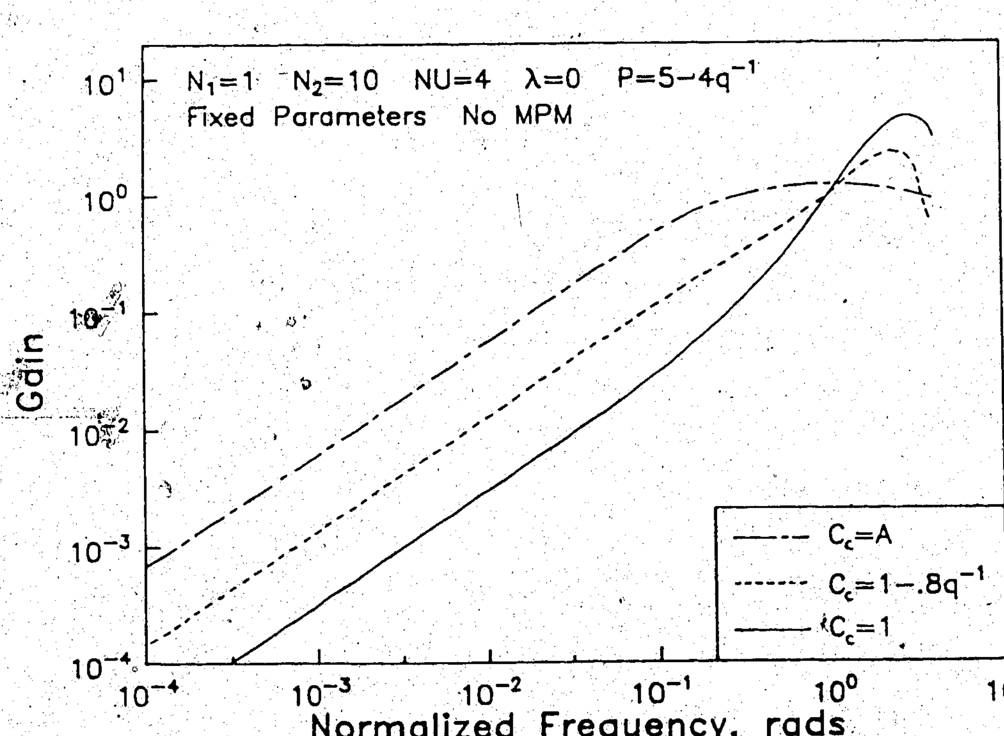


Figure 5.10b Frequency Response of Transfer Function $G_{yd}(z^{-1})$ arising from GPC Control of Process C

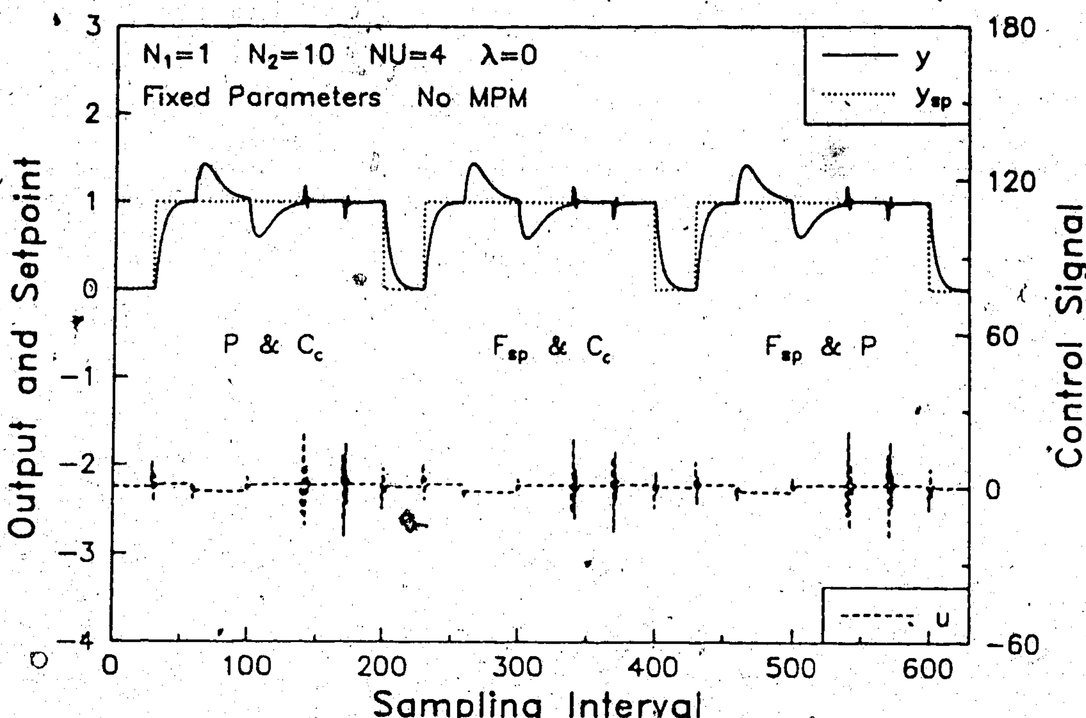


Figure 5.11 Different Methods of Specifying Servo and Regulatory Behavior

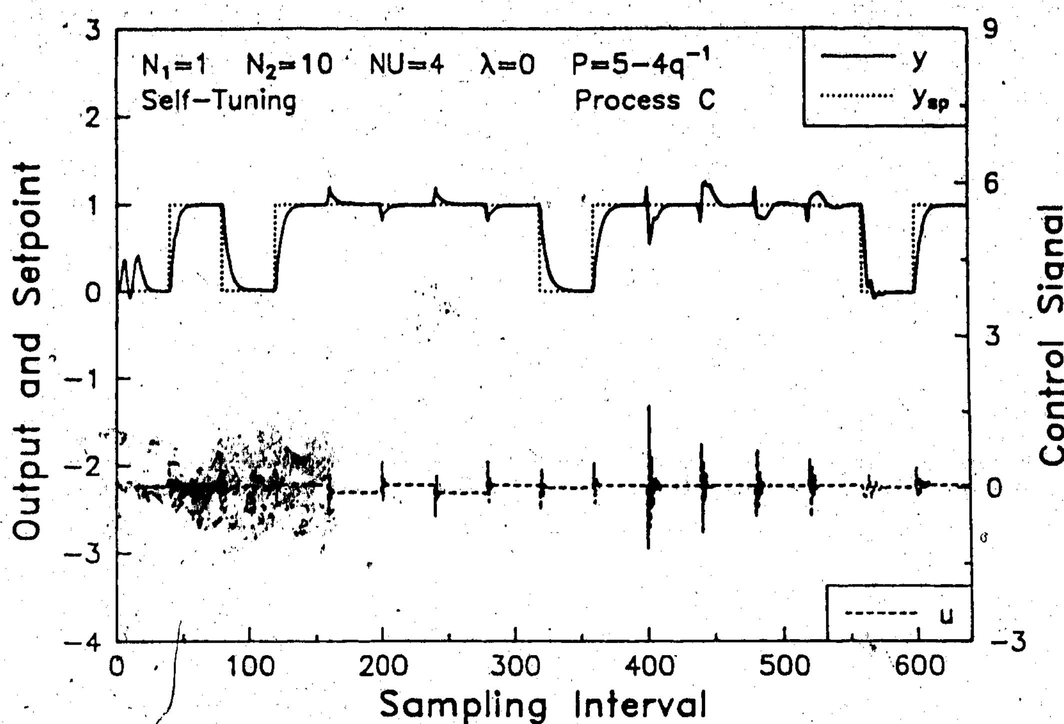


Figure 5.12a Generalized Predictive Control with $C_c=1$ in the Presence of Various Step Disturbances

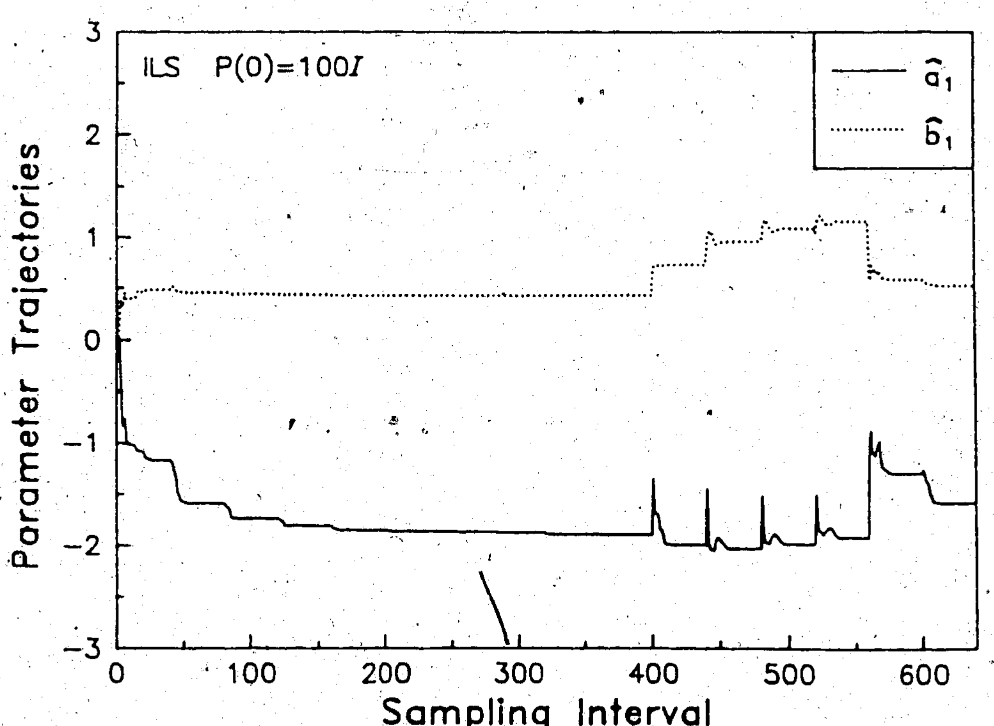


Figure 5.12b Corresponding Parameter Estimation using RLS

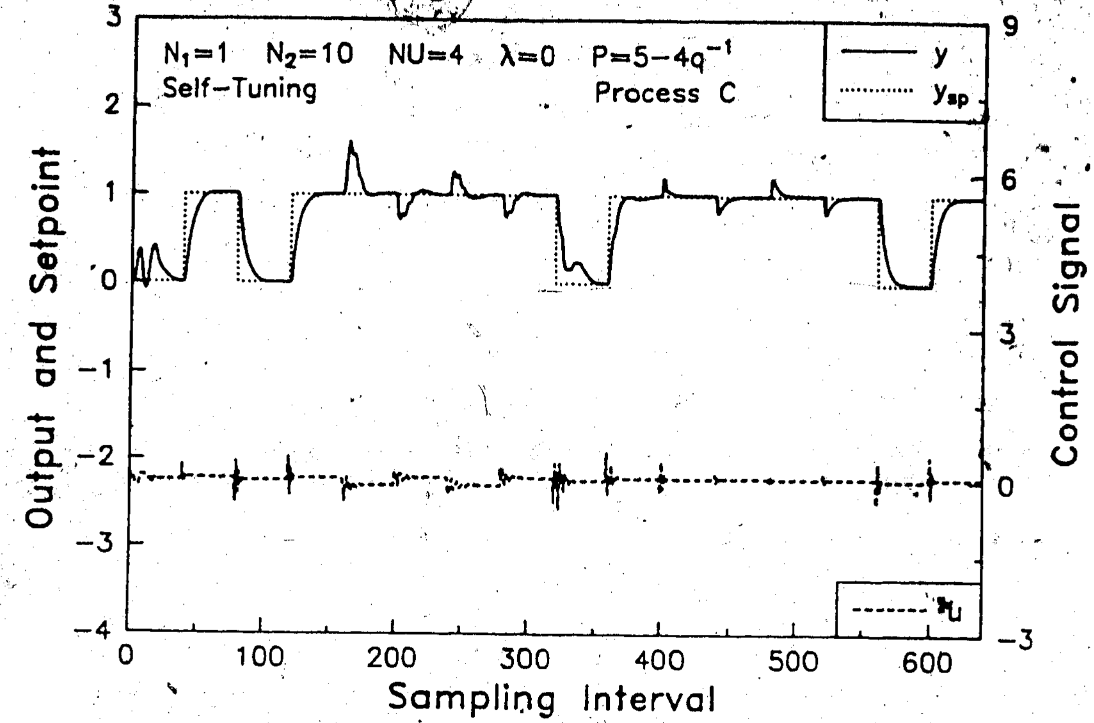


Figure 5.13a Generalized Predictive Control with $C_c = A$ in the Presence of Various Step Disturbances

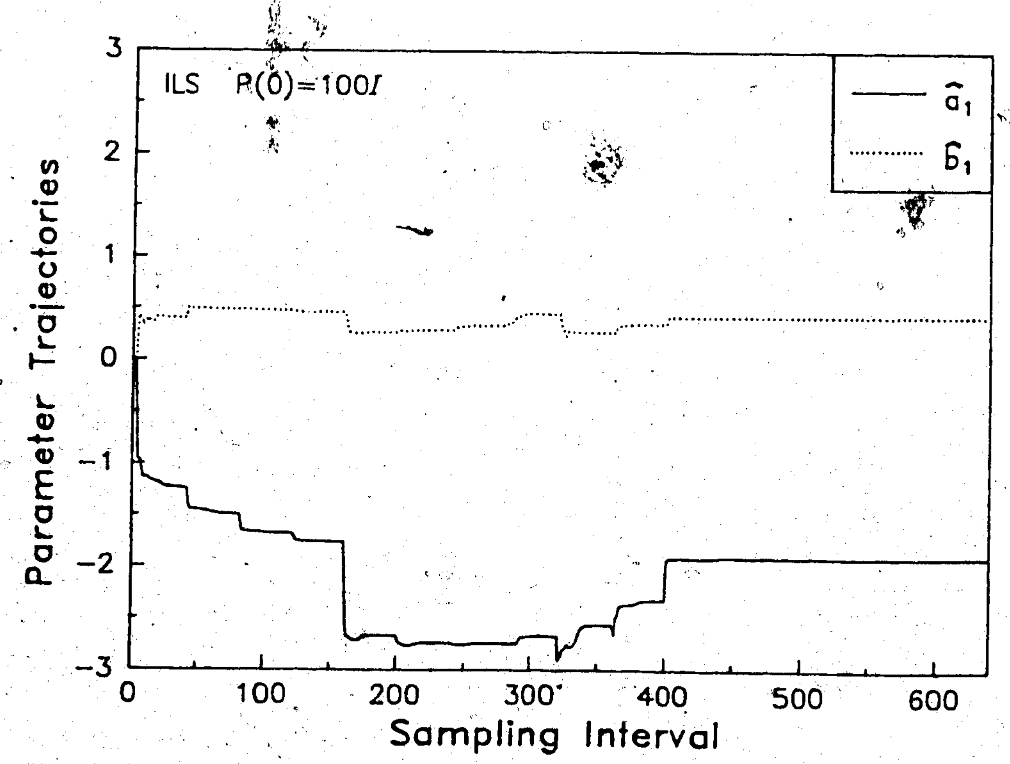


Figure 5.13b Corresponding Parameter Estimation using RLS with Regressor Filtering

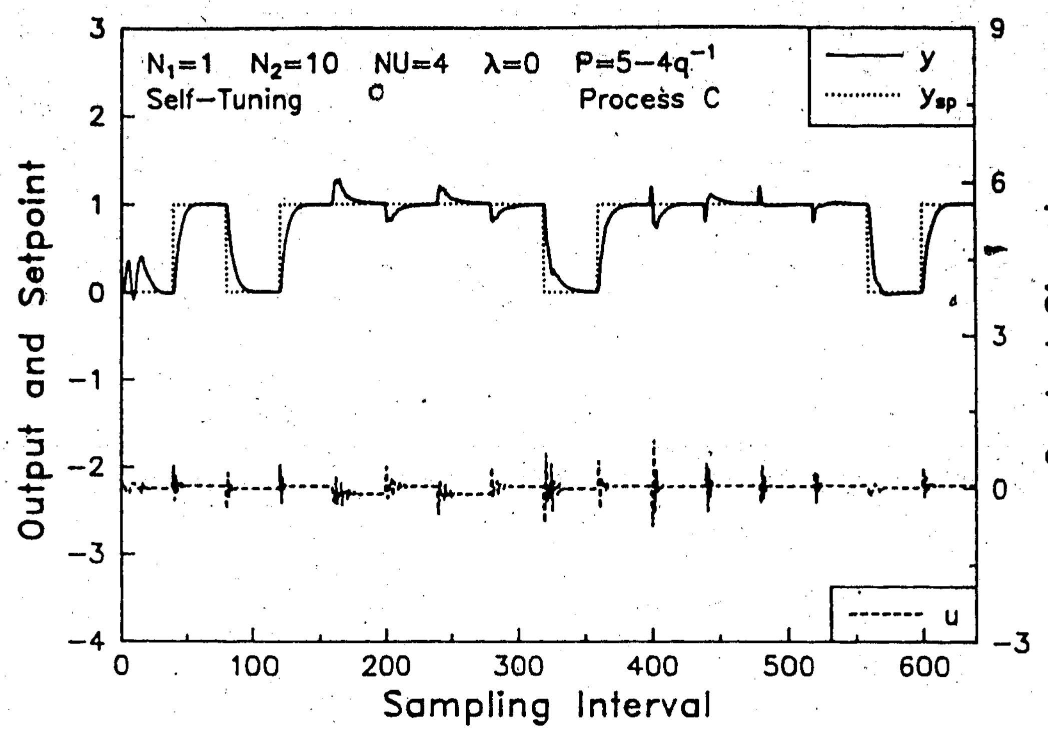


Figure 5.14a Generalized Predictive Control with $C_c=1-.8q^{-1}$ in the Presence of Various Step Disturbances

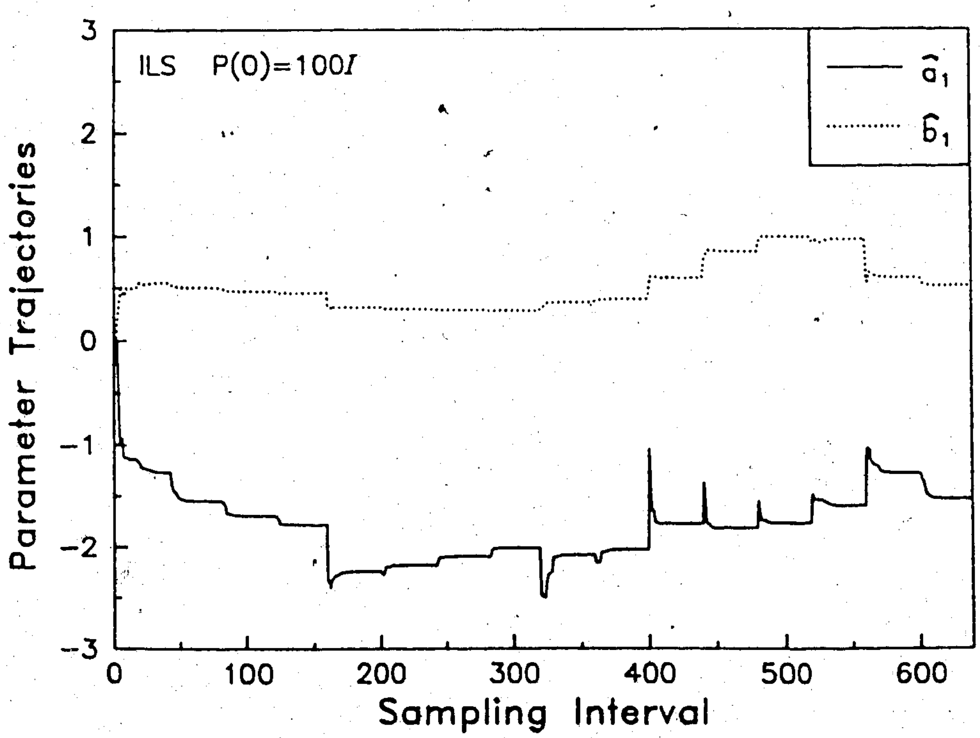


Figure 5.14b Corresponding Parameter Estimation using RLS with Regressor Filtering

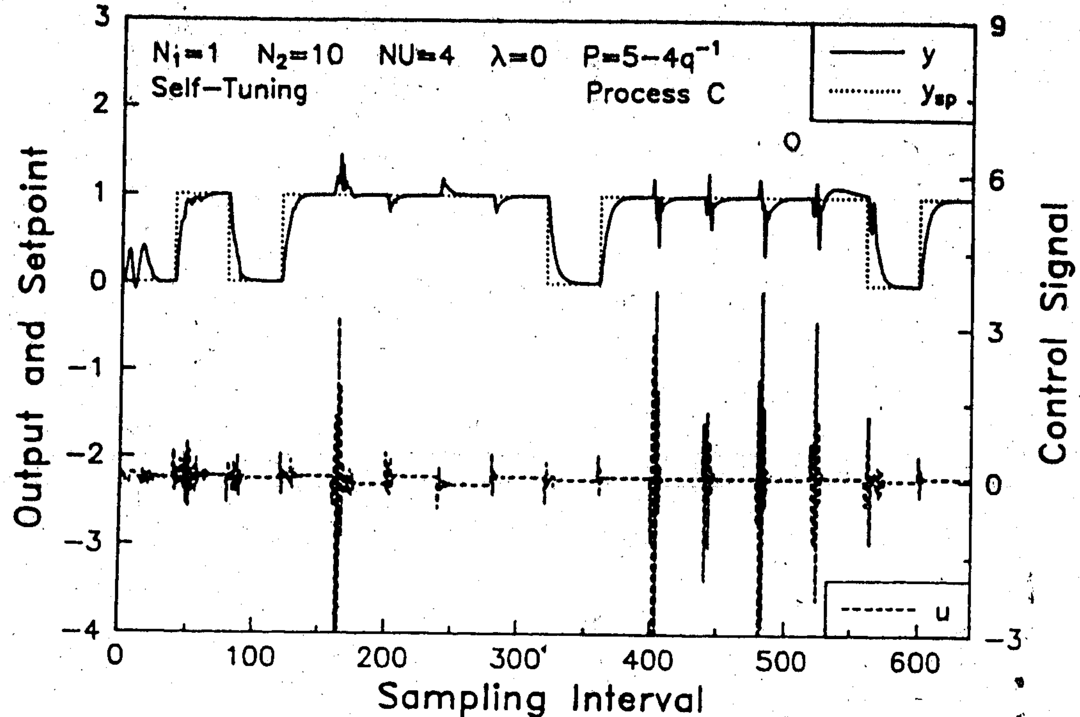


Figure 5.15a Generalized Predictive Control using $\hat{C}$ from ELS in the Presence of Step Disturbances

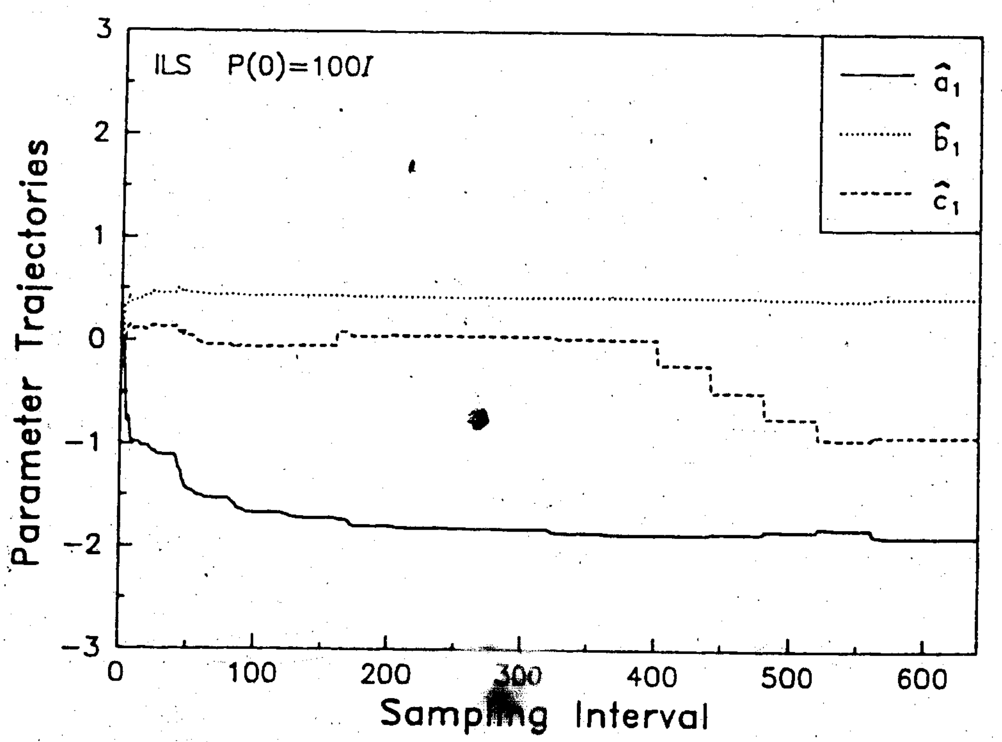


Figure 5.15b Corresponding Parameter Estimation using ELS

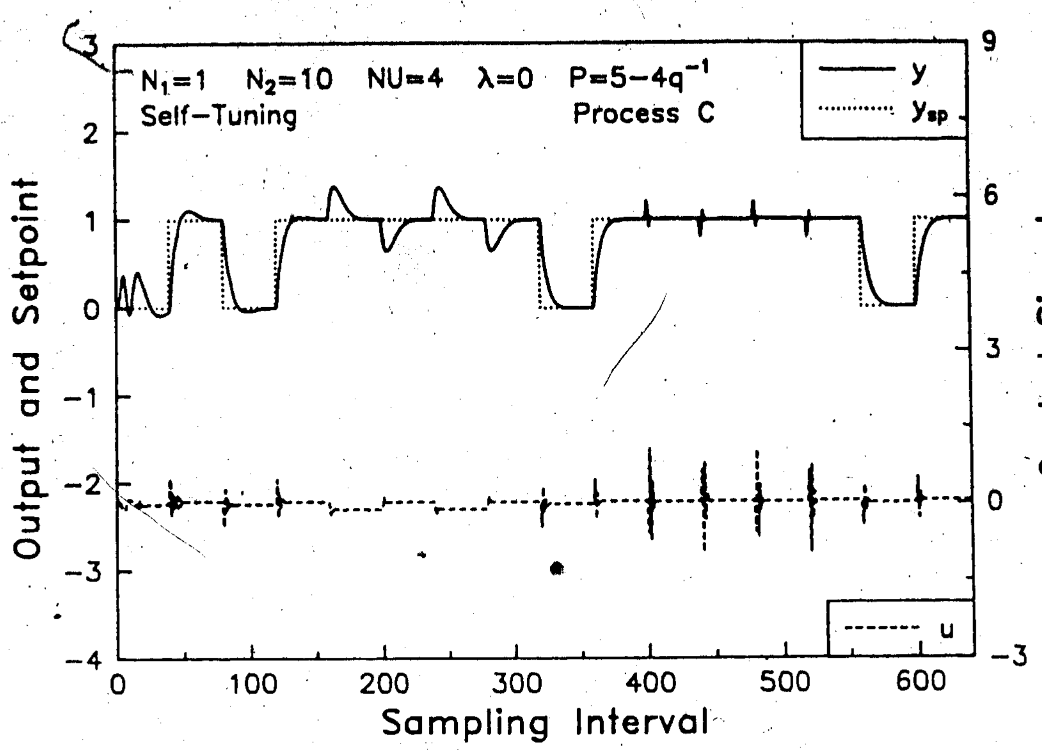


Figure 5.16a Generalized Predictive Control with $C_c = 1 - .8q^{-1}$ in the Presence of Various Step Disturbances

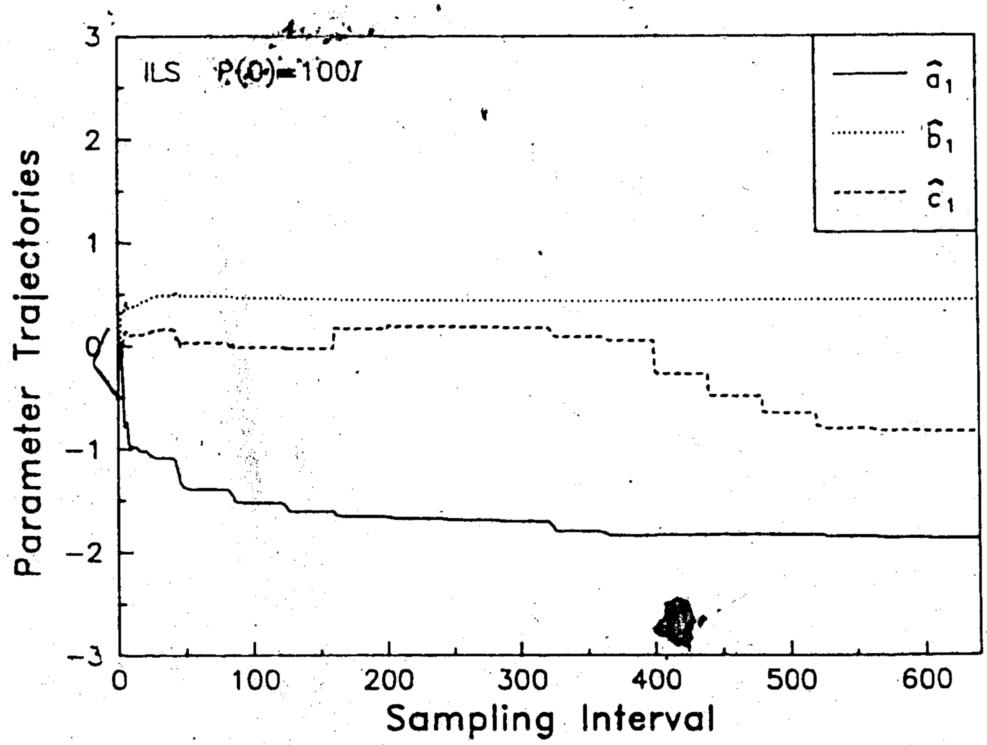


Figure 5.16b Corresponding Parameter Estimation using ELS

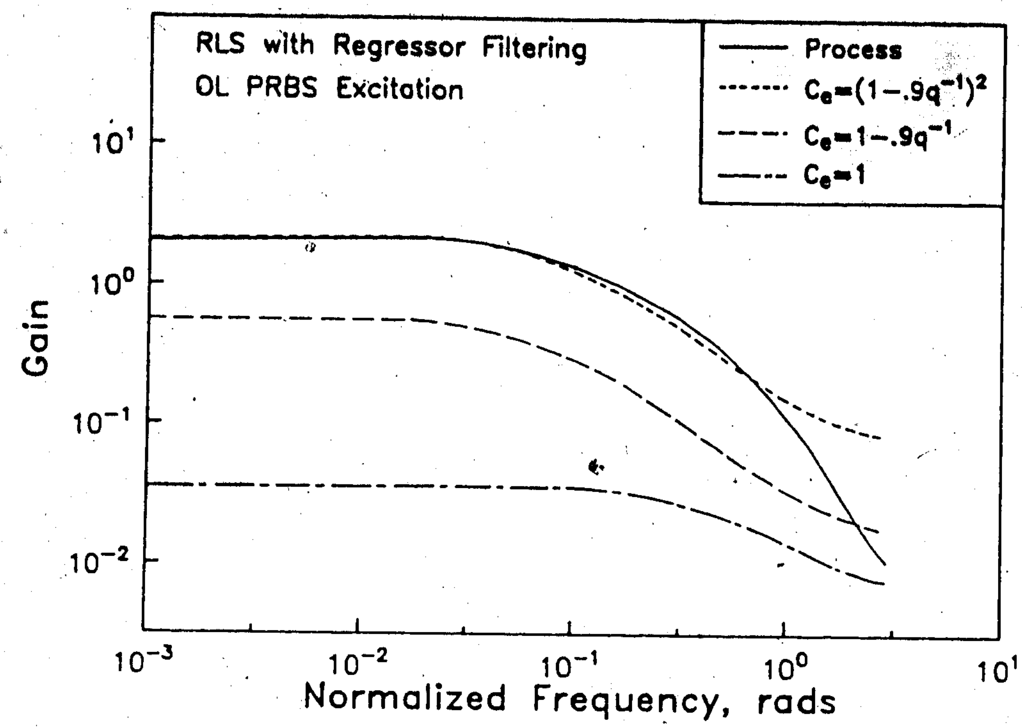


Figure 5.17a Frequency Response of Rohrs' Process and Identified First Order Models

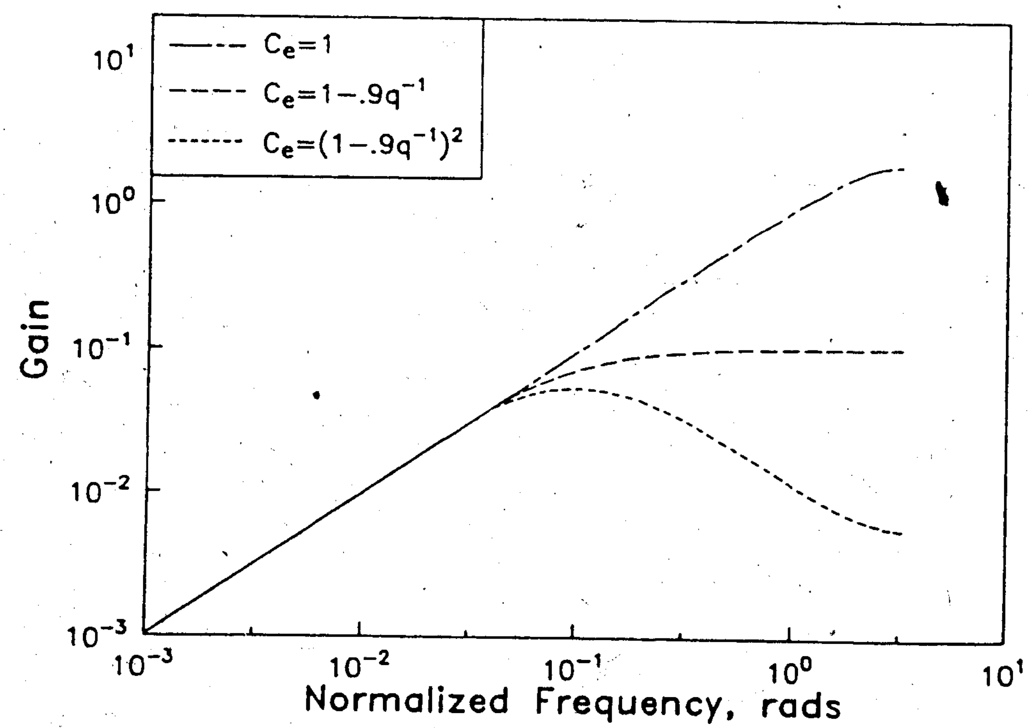


Figure 5.17b Frequency Response of Filter Δ/C_e

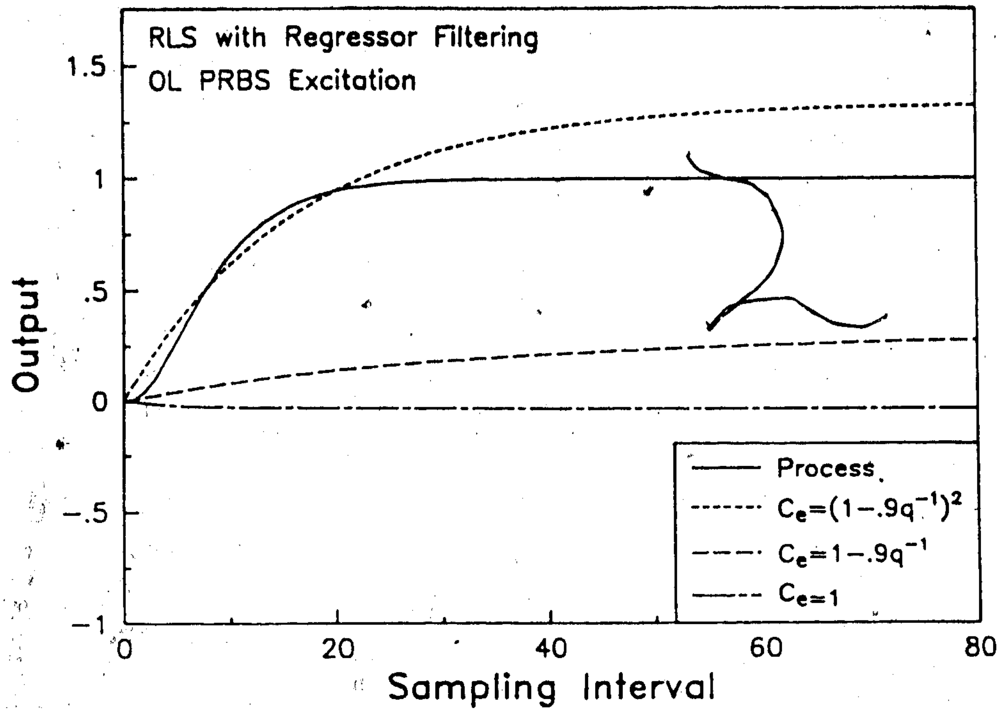


Figure 5.18a Unit Step Response of Process C and Identified First Order Models

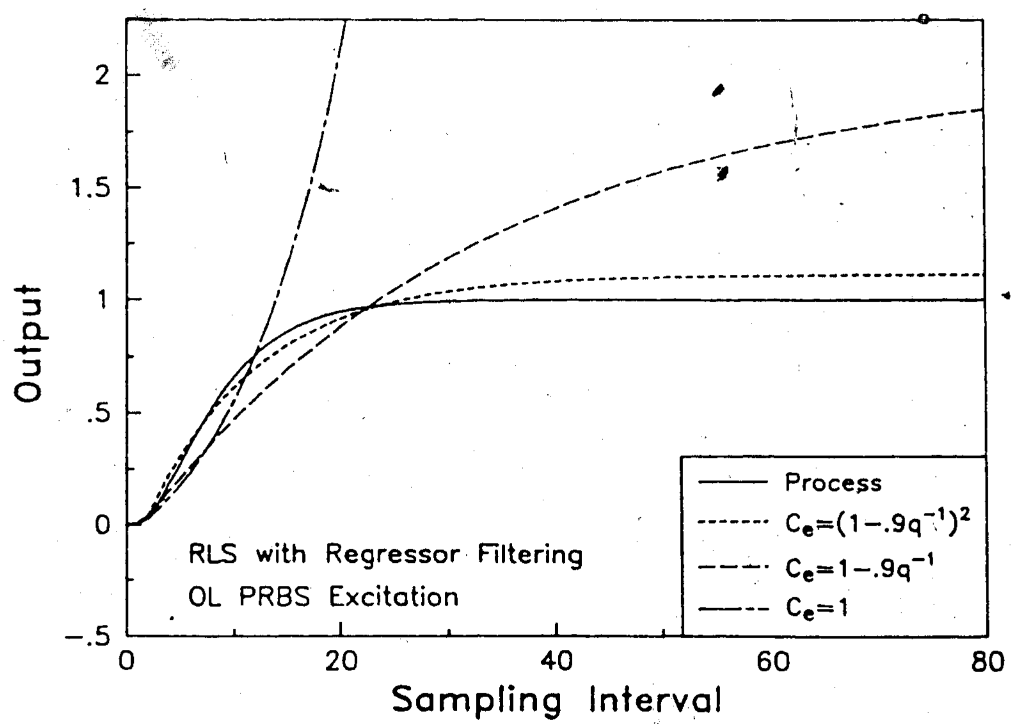


Figure 5.18b Unit Step Response of Process C and Identified First Order + Time Delay Models

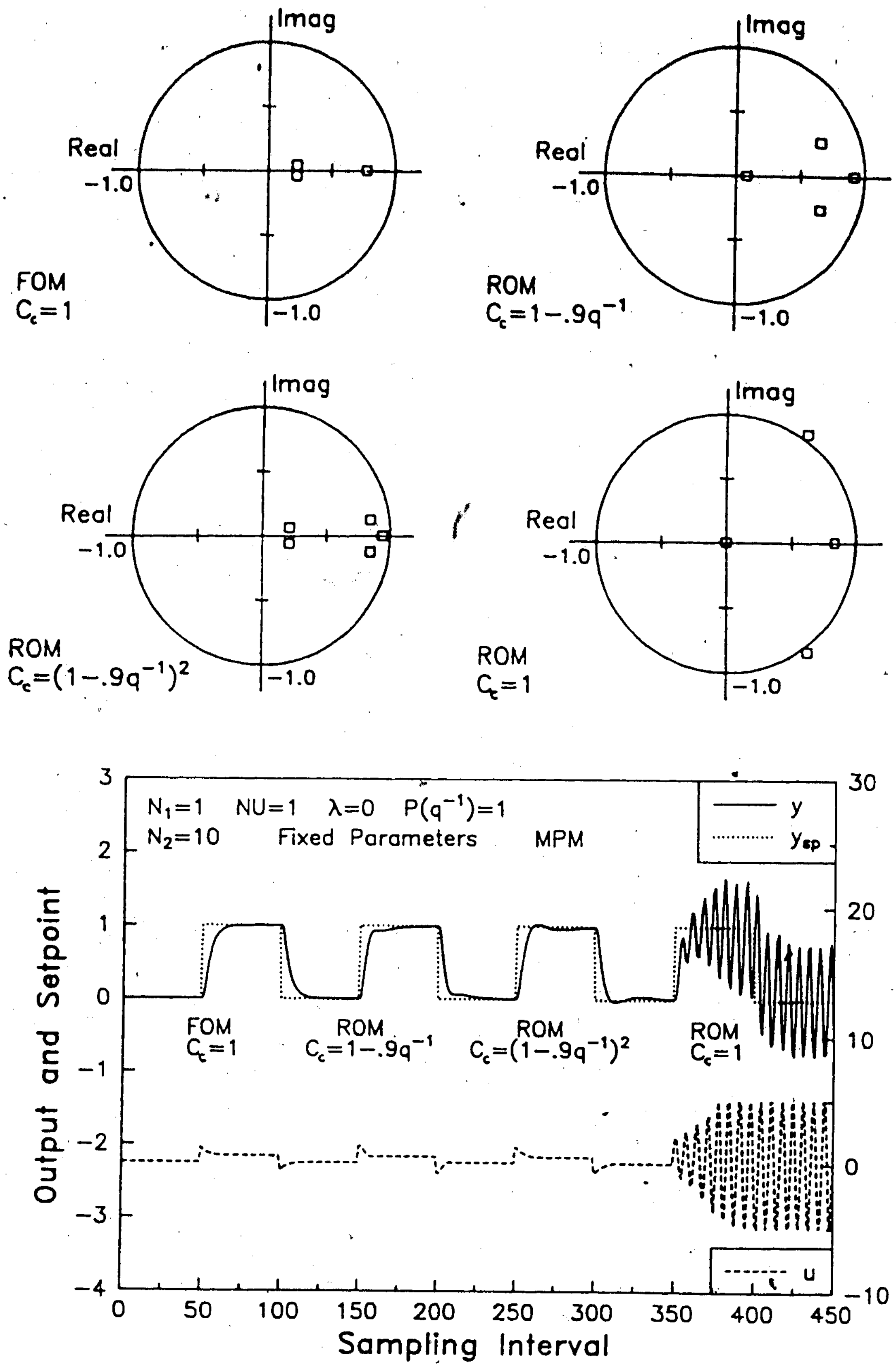


Figure 5.19 Output Horizon Control of Rohr's Process with Unmodeled Dynamics

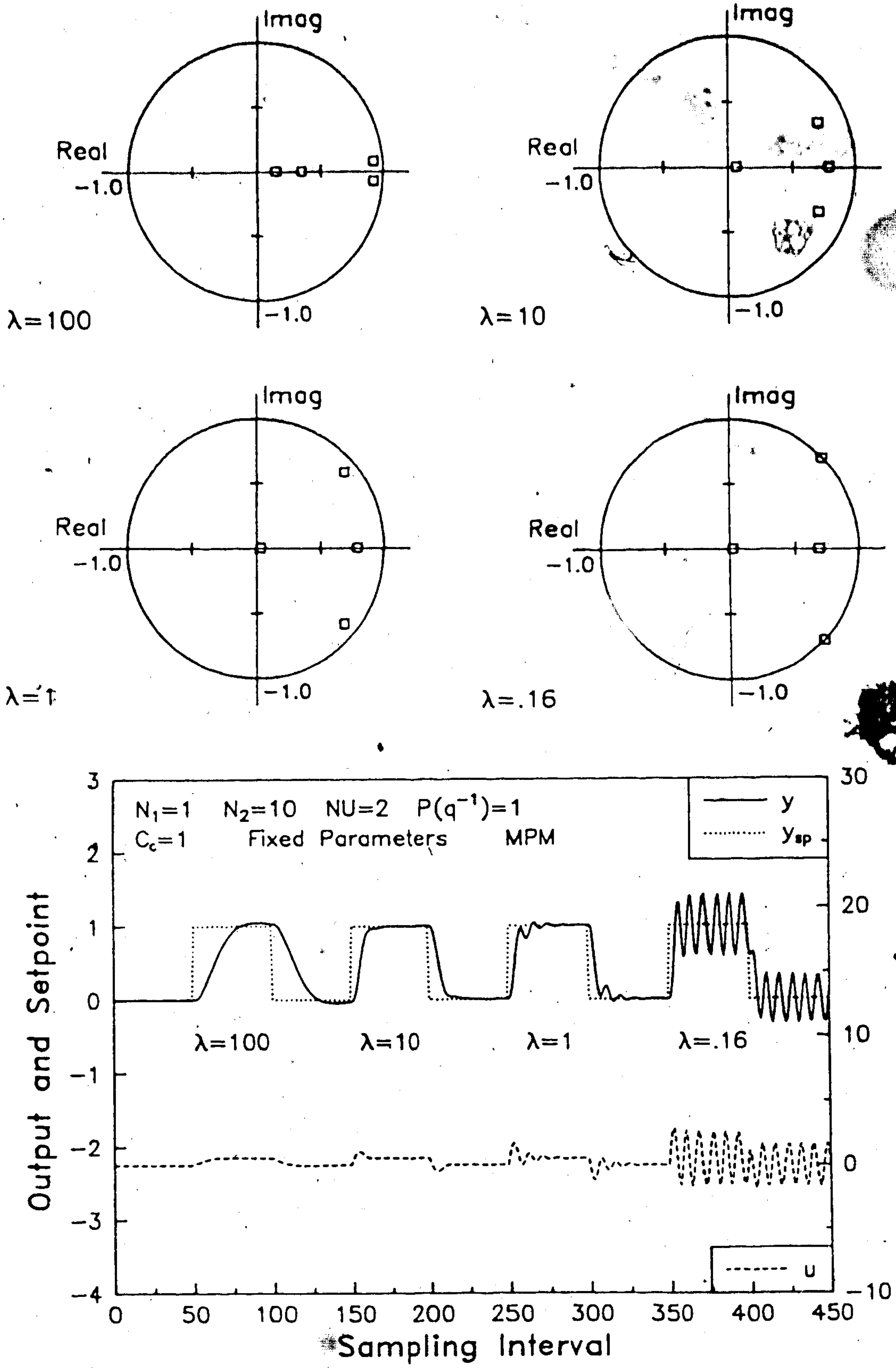


Figure 5.20 Lambda Weighting Control of Rohr's Process with a Reduced Order Model

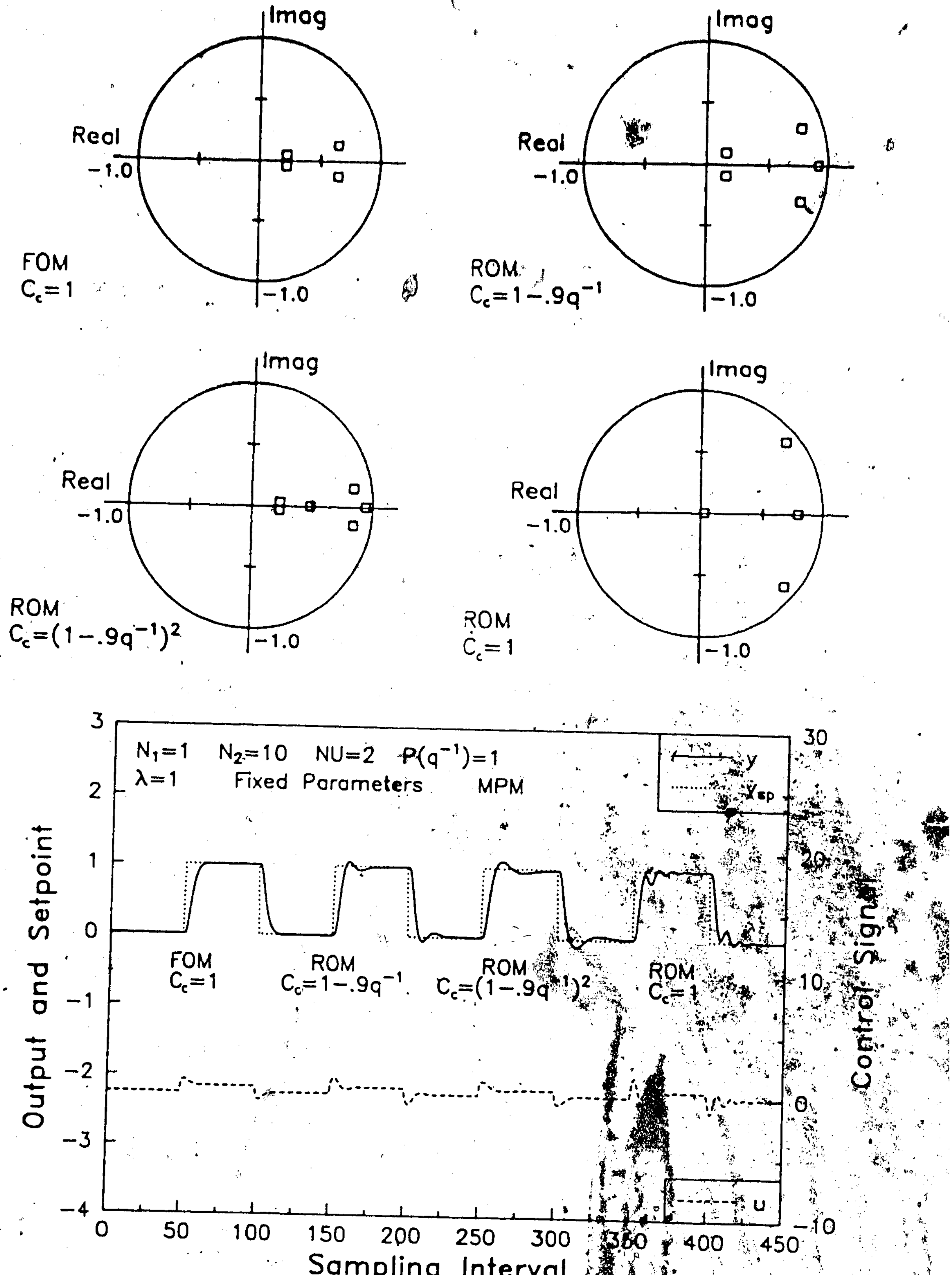


Figure 5.21 Effect of C_c Design Polynomial on Lambda Weighting Control of Rohr's Process

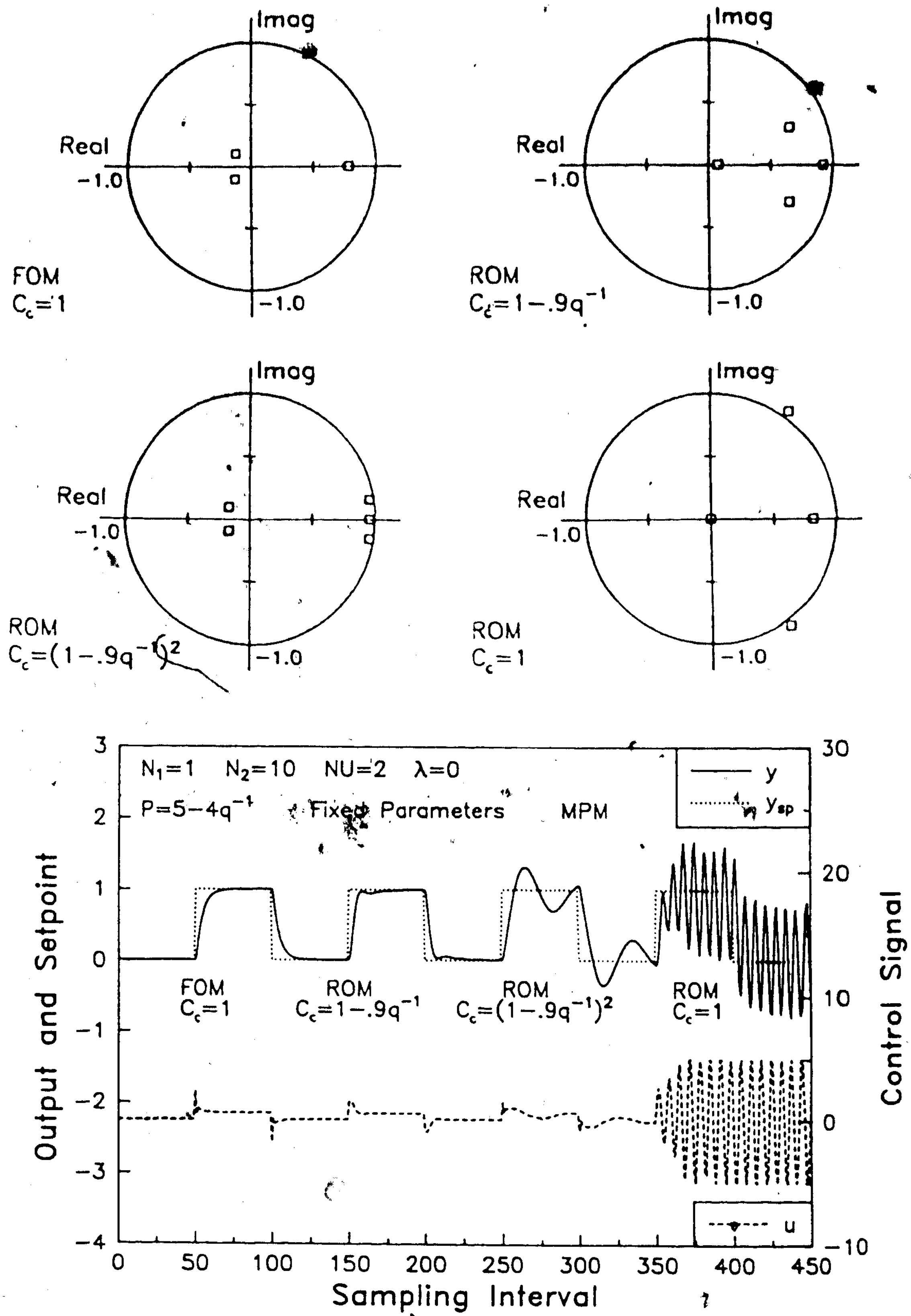


Figure 5.22 Detuned Model–Following Control of Rohr's Process with Unmodeled Dynamics

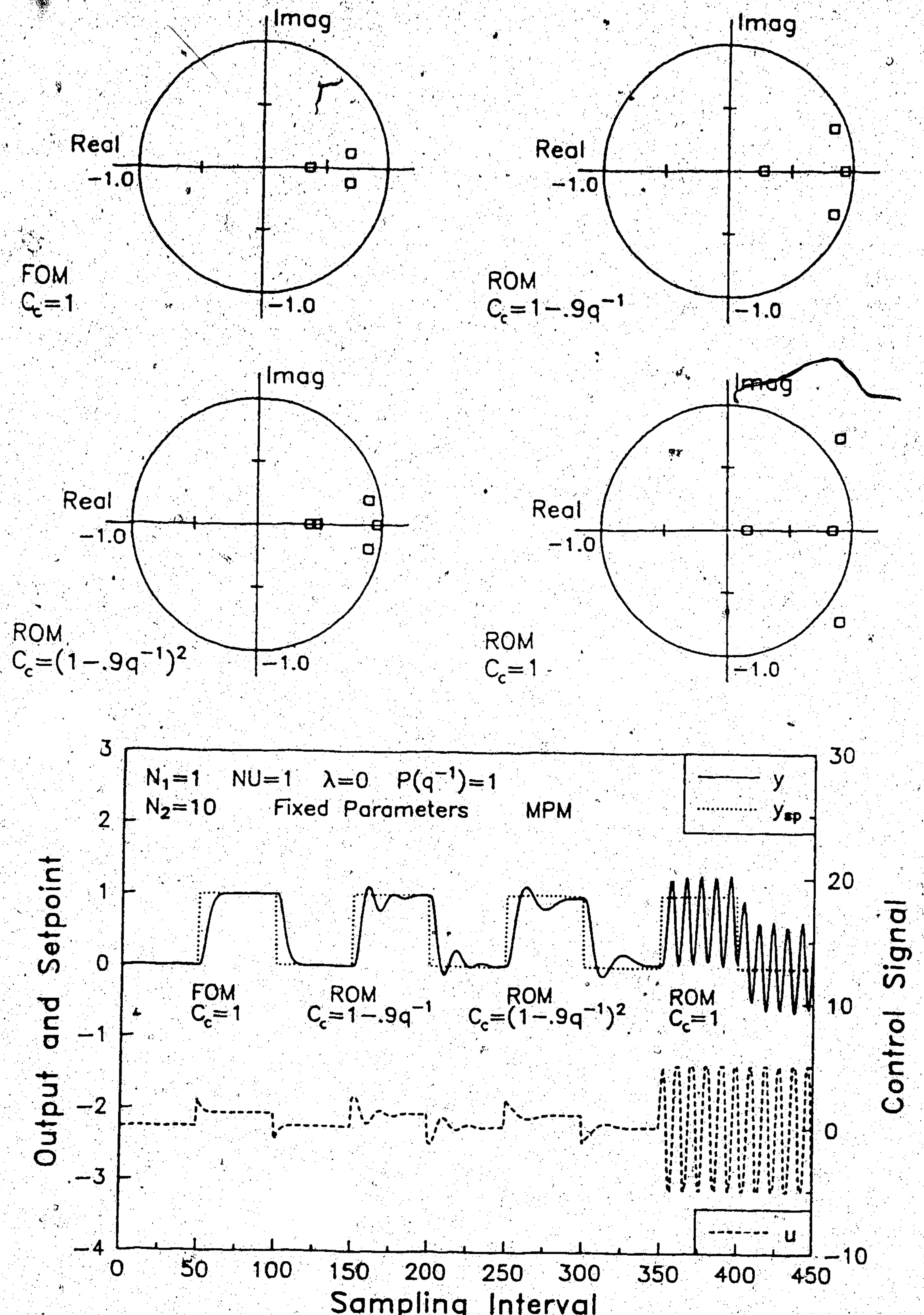


Figure 5.23 Output Horizon Control of Process C with a First Order Model

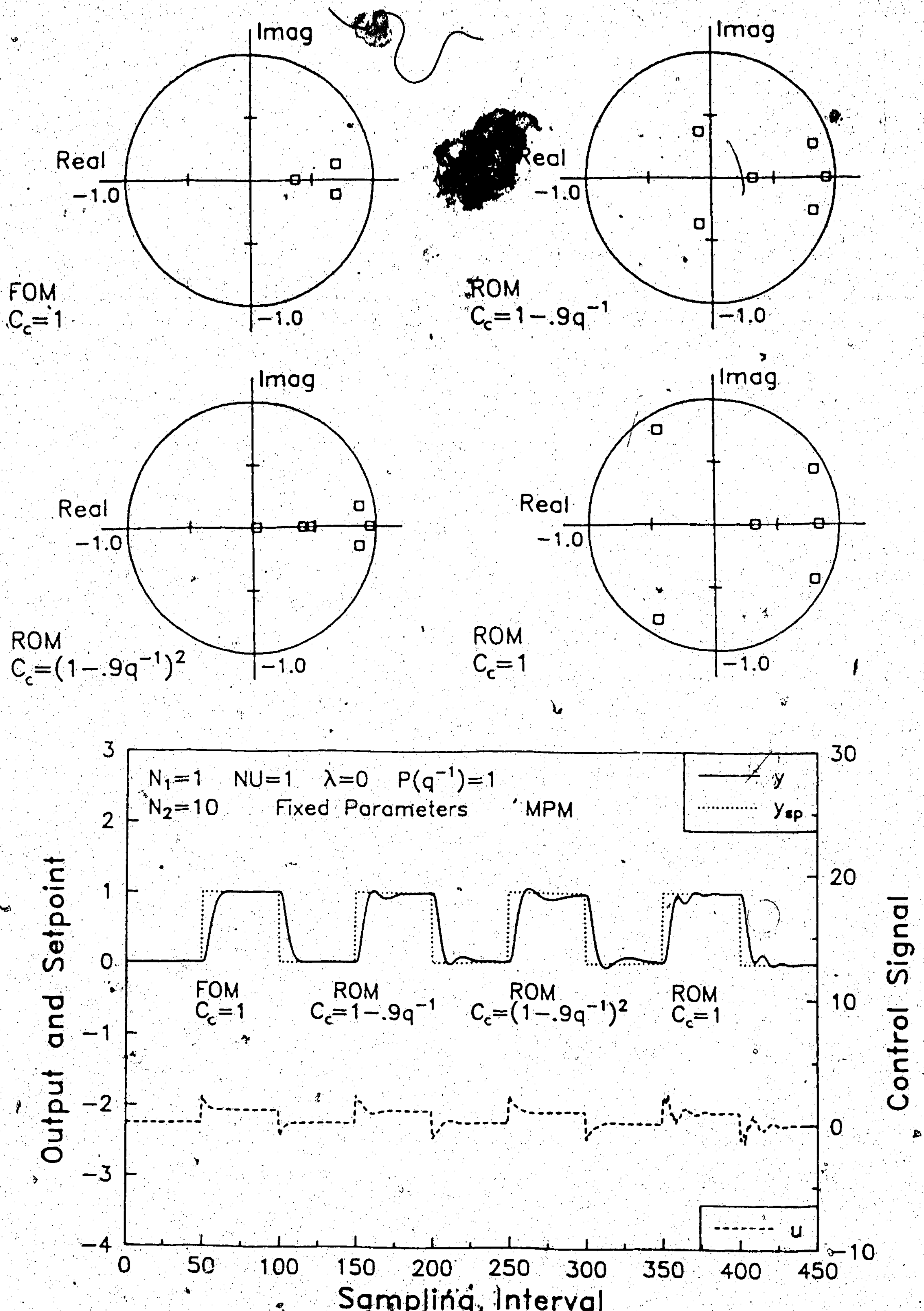


Figure 5.24 Output Horizon Control of Process C with a First Order + Time Delay Model

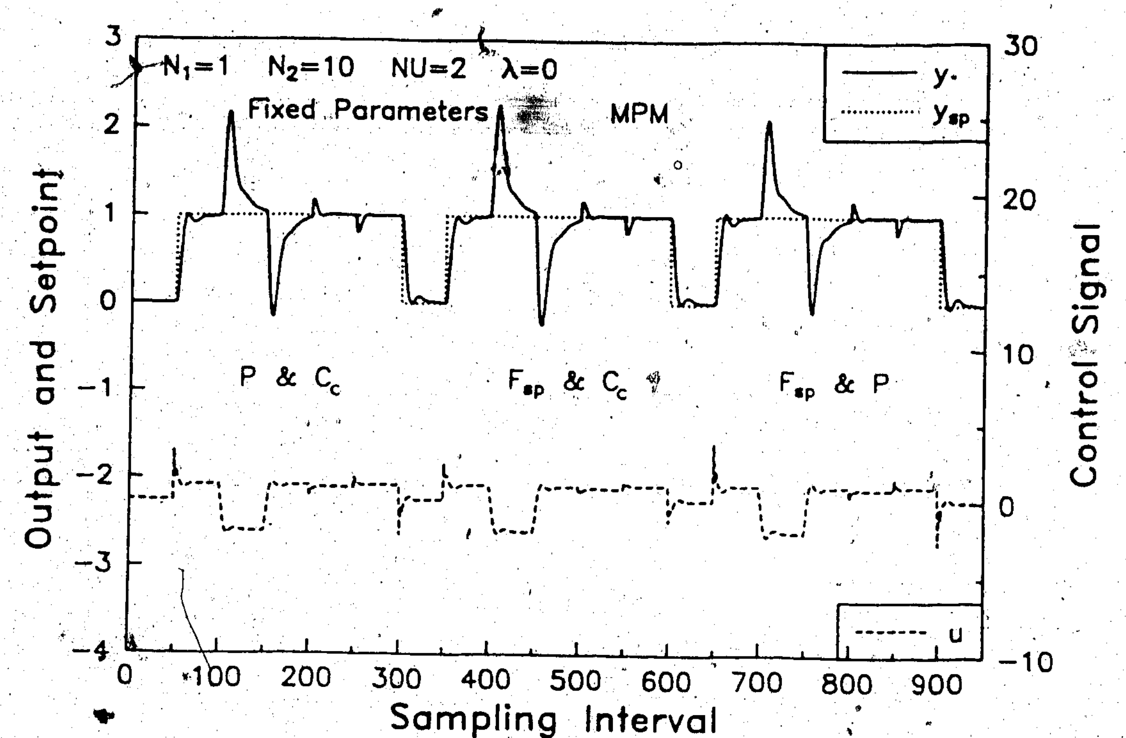


Figure 5.25 Different Methods of Specifying Servo and Regulatory Behavior in the Presence of Unmodeled Dynamics

6. PERFORMANCE AND TUNING OF POLE PLACEMENT CONTROL

This chapter is concerned with the selection of the tuning parameters and resulting closed-loop behavior of the Pole Placement (PP) algorithm, derived in Chapter 3. Placing the poles, at specified locations implies that the output response of the system will be invariant to process gain changes as long as the model reflects the actual process (section 6.1). Although the PP control strategy does not have a large number of tuning parameters, compared with GPC, the selection of the desired closed-loop poles requires consideration (section 6.2). The extension of the scheme to include a setpoint prefilter and a polynomial for tailoring disturbance rejection, allows Pole Placement to meet independent servo and regulatory objectives (section 6.3). Finally, the performance and properties of the Pole Placement controller will be demonstrated through a simulation study (section 6.4).

6.1 Maintenance of Output Performance for Process Gain Changes

Consider the case where there are gain changes in a process but there is no model-plant mismatch. (The estimated parameters track the true values exactly). Process gain changes are reflected entirely in the B parameters such that the gain may be factored out:

$$B = K_p B' \tag{6.1.1}$$

The prime denotes a polynomial corresponding to a gain of unity. The closed loop expression for the output was given by eqn. (3.1.15):

$$y(t) = \frac{P(1)B}{PB(1)} q^{-1} w(t) + \frac{G}{P} \xi(t)$$

Clearly the servo response of the system will be invariant of process gain changes. Note that G is obtained from the solution of the Diophantine equation (3.1.11):

$$GA\Delta + q^{-1}BF = PC$$

With B replaced by $K_p B'$ the polynomials G and F for an arbitrary gain may be expressed in terms of the solution for a unit gain:

$$G = G', \qquad F = F'/K_p \tag{6.1.2}$$

Substituting for B and G, the closed-loop transfer function becomes:

$$y(t) = \frac{P(1)B'}{PB'(1)} q^{-1} w(t) + \frac{G'}{P} \xi(t) \tag{6.1.3}$$

and it is apparent that the response of the output will be unaffected by process gain changes.

6.1.1 Changes in Process Dynamics

For this PP controller, the open-loop process zeros are retained in the closed-loop. Therefore, changing process dynamics will influence the behavior of the closed-loop, even with perfect modelling. However, in many cases, the zeros do not have a strong impact on the output response and hence the closed-loop performance will be **almost** invariant.

In contrast, the output performance will be seriously degraded if the order of the process decreases such that near common factors arise in the estimated model polynomials, $\hat{A}$ and $\hat{B}$ (Clarke et al., 1987a).

6.2 Specification of Desired Closed-Loop Poles

The desired time-domain response is specified using the desired characteristic polynomial, $P(q^{-1})$. For example, a first order closed-loop response with a time constant τ_{CL} may be sought by specifying:

$$P(q^{-1}) = 1 - p_1 q^{-1} \qquad \text{with } p_1 = e^{-T/\tau_{CL}} \tag{6.2.1}$$

A more realistic user-specified characteristic polynomial for typical (relatively high order) industrial processes is 2nd order. Significantly less control action is normally required for a 2nd order type response with the same rise time. The denominator of the pulse transfer function of an underdamped 2nd order system with natural period, τ, and damping factor, ς, was given in section 4.4.1:

$$P(q^{-1}) = 1 + p_1 q^{-1} + p_2 q^{-2} \tag{6.2.2}$$

with $p_1 = -2e^{-\zeta T_s/\tau} \cos\left[(T_s/\tau)\sqrt{1-\zeta^2}\right]$ and $p_2 = e^{-2\zeta T_s/\tau}$

Note that since the open-loop zeros of the process are retained in the closed-loop, the actual damping factor and natural period will differ from the desired values, even with no MPM. It is intuitively easier to specify the desired overshoot and rise time as opposed to τ and ζ. The damping factor for a 2nd order model is an explicit function of the fractional overshoot given by eqn. (4.4.9):

$$\zeta = \sqrt{\frac{(\ln o_v)^2}{\pi^2 + (\ln o_v)^2}}$$

Unfortunately, the 63% (or any other) rise time is a complex function of both τ and ζ. It is approximately equal to 2τ as long as $0.5<\zeta<1.0$. For a critically damped 2nd order system ($\zeta=1.0$) the precise relationship is

$$\tau = t_{.63}/2.15 \tag{6.2.3}$$

6.3 Independent Servo and Regulatory Control

The difficulty of obtaining a recursive estimate of the noise model, $\hat{C}$, useful for control purposes (discussed in section 4.5.1) implies that the use of a design polynomial, C_c, is a more effective approach. This design polynomial is eliminated from the servo response, as long as eqn. (3.1.14) is satisfied; therefore only unmodelled dynamics and disturbances are affected by C_c. The servo response alone may be modified by introducing a setpoint prefilter ($w(t) = F_{sp} y_{sp}(t)$; $F_{sp}(1) = 1$). Then, as with GPC, there exists several different ways of utilizing "two degrees of freedom" to meet independent servo and regulatory objectives. The closed-loop expression with a setpoint prefilter as derived in section 3.1.2 is:

$$y(t) = \frac{F_{sp} P(1) B}{PB(1)} q^{-1} y_{sp}(t) + \frac{GC\Delta}{PC_c} x(t) \tag{6.3.1}$$

The three methods outlined in Table 6.1 may be used to select F_{sp}, P and C_c so that the closed-loop transfer function becomes:

$$y(t) = \frac{P_s(1)B}{P_sB(1)} q^{-1} y_{sp}(t) + \frac{GC\Delta}{P_r} x(t) \qquad (6.3.2)$$

where P_s is the desired servo "characteristic polynomial"
P_r is the desired regulatory "characteristic polynomial"

Table 6.1 PP Alternative Settings for Independent Servo and Regulatory Control

Case	F_{sp}	P	C_c
1	1	P_s	P_r/P_s
2	$P_s(1)/P_s$	1	P_r
3	$\frac{P_rP_s(1)}{P_sP_r(1)}$	P_r	1

Note that since $PC_c = P_r$ for all three cases, the polynomials F and G obtained by solution of the Diophantine equation (3.1.11):

$$GA\Delta + q^{-1}BF = PC_c$$

are the same. Also, the product HF_{sp} in the control law,

$$G\Delta u(t) = HF_{sp}y_{sp}(t) - \sigma Fy(t)$$

works out to be $P_s(1)P_r/P_sB(1)$ in each case. Thus, while each of the three alternatives discussed in the following sections is conceptually different, the final control law obtained is identical, for given polynomials P_r and P_s.

6.3.1 Two Degrees of Freedom, Case 1: Using P and C_c

If the P polynomial is used to form the desired pole set for control, the design polynomial C_c may be used to add additional po[illegible] disturbance rejection. The regulatory characteristic equation must[illegible] the servo characteristic equation as a factor to satisfy the restriction that C_c be a polynomial. With this scheme, the servo and regulatory modes are not completely decoupled; altering the desired response to setpoint changes will also affect the behavior for load variations. C_c may be interpreted as a fixed noise model and used to filter the regressor to improve parameter estimation as discussed in section 4.7. This alternative is by far the most commonly referred to in the literature; it may be used for all applications except those where the user wishes the servo and regulatory modes to be completely different.

6.3.2 Two Degrees of Freedom, Case 2: Using F_{sp} and C_c

With P=1 and $F_{sp}=P_s(1)/P_s$, the controller with provide a deadbeat response to filtered setpoint changes. The C_c polynomial can then be used to independently specify the regulatory response; the two modes are completly decoupled. For this strategy, C_c is the desired regulatory characteristic polynomial (as opposed to the noise model of case 1) and should not be used to filter the regressor for identification.

6.3.3 Two Degrees of Freedom, Case 3: Using F_{sp} and P

Without using the design polynomial C_c, the desired regulatory response may be specified using P and the setpoint filter transfer function selected to cancel the desired regulatory poles and introduce a different set of poles for servo control. At an increase in complexity, C_c, representing information about the type of disturbances usually encountered, may be added. Together, C_c and P, then determine how disturbances are to be handled.

6.4 Evaluation of Pole Placement Control

A simulation study, paralleling that completed for GPC, was undertaken to evaluate the PP controller. The transfer functions for the processes considered are listed in Table 5.1. Details on the types of disturbances studied may be found in section 5.1.2.

6.4.1 Evaluation of PP Control with Exact Process Models

While the evaluation of the PP controller was carried out with each of the processes listed in Table 5.1, for the sake of brevity, only simulations with Process C will be presented. This third order discrete-time NMP plant is considered to be typical of many processes in the chemical industry. The simulations involving control with an exact process model are summarized in Table 6.2. Figure numbers for GPC runs which are directly comparable are listed in the last column of this table.

Table 6.2 PP Simulations with Exact Process Models (nonadaptive)

Fig. No.	Process	Time-Varying	Servo	Regul-atory	Purpose	GPC Comparison Fig. No.
6.1	C	no	yes	no	effect of $P(q^{-1})$	5.1b,3b,5
6.2	C	yes	yes	no	main. of performance	5.6-8
6.3	C	no	yes	yes	effect of $C_c(q^{-1})$	5.9b
6.4	"	"	"	"	2 degrees of freedom	5.11

6.4.1.1 Varying the Speed of Response

Figure 6.1 shows the servo response of the system to step setpoint changes as the desired closed-loop pole is varied from 0 to 0.9. Significantly less control action was required for the case of a second order critically damped desired characteristic polynomial with the same 63% rise time (results not shown).

6.4.1.2 Maintenance of Servo Performance

The Pole Placement control strategy inherently gives a closed-loop response which is invariant of process gain variations (in the absence of MPM). However, no guarantee of maintenance of output performance can be given for changing process dynamics as the OL zeros are retained in the CL transfer function. To examine the behavior of PP control for a time-varying process with precisely known parameters, the simulation described in section 5.2.4 was repeated. A second order desired characteristic polynomial was specified corresponding to a critically damped process with natural period, $\tau=3$. As Figure 6.2 illustrates, the servo response of the system is almost perfectly invariant inspite of the large changes in dead-time, gain, and process dynamics. The results compare very favorably with those of GPC in the Detuned Model-Following mode (cf. Figure 5.8).

6.4.1.3 Disturbance Tailoring using C_c

The controller design polynomial C_c may be used to alter the regulatory response of the closed-loop, without affecting the servo behavior, in a similar manner as for GPC. For the simulation shown in Figure 6.3, unmeasured step disturbances $d_u(t)$ and $d_y(t)$ of magnitudes 0.1 and 0.2, respectively, were applied and removed after each upward-going setpoint change. With $C_c=1$, the controller provides optimal rejection of the "input" type disturbance but overreacts to the steps in $d_y(t)$. With $C_c=A$ the situation is reversed; the steps in $d_u(t)$ are rejected very slowly while the "output" type disturbance is handled in an optimal manner. The third portion of the simulation serves to denomstrate that $C_c=1-.8q^{-1}$ provides a reasonable compromise.

6.4.1.4 Two Degrees of Freedom

Earlier discussion pointed out that there are several alternatives open to the user in terms of specifying the desired servo and regulatory closed-loop poles. The simulation in Figure 6.4 was carried out with Process C and the same sequence of setpoints and disturbances used previously. The tuning parameters P, C and F followed the schedule:

Sampling Interval	F_{sp}	P	C_c
0-229	1	$(1-.8q^{-1})$	$1-.9q^{-1}$
230-429	$.2/(1-.8q^{-1})$	1	$(1-.8q^{-1})(1-.9q^{-1})$
430-630	$(1-.9q^{-1})/.1$	$(1-.9q^{-1})(1-.8q^{-1})$	1

similar to that for the analogous GPC run in section 5.2.6. The desired servo and regulatory characteristic polynomials are:

$$P_s = 1-.8q^{-1}$$
$$P_r = (1-.8q^{-1})(1-.9q^{-1})$$

in each case. Identical responses are obtained as expected. The same would be true even if there existed model-plant mismatch, since the control law for each case is identical (see section 6.3). Note that the actual regulatory response obtained depends upon the type of disturbance entering the loop.

6.4.2 Evaluation of PP Control in the Presence of Model-Plant Mismatch

The simulations in this section are based on the reduced order models identified off-line in section 5.3.2. Table 6.3 provides a summary. The last column of this table indicates the figure number of the comparable GPC simulation presented in Chapter 5.

Table 6.3 PP Simulations with Model-Plant Mismatch (nonadaptive)

Fig. No.	Process	Servo	Regulatory	Purpose	GPC Comparison Fig. No.
6.5	F	yes	no	Effect of C_c	5.19,21,22
6.6,7	C	yes	no	Effect of C_c	5.23,24

Notes: time invariant 3rd order processes
reduced (1st) order models identified off-line (section 5.3.2)

6.4.2.1 Rohrs' Process

In Chapter 5, the following reduced order model (ROM) of Rohrs' process was identified using RLS with regressor filtering:

$$\hat{A} = 1-.93q^{-1} \quad \hat{B} = .145$$

To examine the robustness of Pole Placement control with respect to unmodelled dynamics, this ROM was used along with a first order desired characteristic polynomial, $P=1-p_1q^{-1}$ and the design polynomial, $C_c=1$. The closed-loop was observed to be stable only for $p_1 \geq 0.91$. However, as Figure 6.5 illustrates, the system may be stabilized with $p_1=0.8$ using a first or second order C_c polynomial. The response to setpoint changes compares favorably to that using the full order model (FOM) and $C_c=1$. The closed-loop poles shown graphically were obtained as roots of the characteristic equation:

$$\hat{G}A\Delta + q^{-1}B\hat{F} = 0$$

where $\hat{F}$ and $\hat{G}$ are computed from the Diophantine equation:

$$\hat{G}\hat{A}\Delta + q^{-1}\hat{B}\hat{F} = PC_c$$

(A,B represent the true process, $\hat{A}$,$\hat{B}$ represent the estimated model).

6.4.2.2 Use of C_c in Achieving Robustness

Recall in section 5.3.2, two first order models of Process C were identified, one with nb=0:

$$\hat{A} = 1-.94q^{-1} \qquad \hat{B} = .080$$

and the other with nb=2:

$$\hat{A} = 1-.91q^{-1} \qquad \hat{B} = .012+.023q^{-1}+.065q^{-2}$$

There is a large amount of model-plant mismatch, particularly with the former model. Figures 6.6 and 6.7 demonstrate the use of the design polynomial C_c in providing robustness to this MPM. For the case of the "first order + time delay" model (i.e. that with nb=2) the system is stable with $C_c=1$ but the response improves considerably when first or second order polynomials are used instead.

6.5 Summary

In general, the simulation results with Pole Placement control compare favorably with those obtained in the previous chapter for the three proposed GPC tuning strategies. In particular:

a) Excellent closed-loop behavior was observed for chemical processes which are nonminimum phase and/or open-loop unstable. A first order desired characteristic polynomial, $P(q^{-1})$, may be used to vary the closed-loop response for a first order plant. For higher order processes, a 2nd order desired characteristic polynomial is recommended.

b) The Pole Placement strategy inherently yields a closed-loop response which is invariant of process gain changes given an accurate model. Changes in process dynamics normally have only a minor influence on the output performance. (Changes in process order, on the other hand, may result in serious deterioration of the response).

c) The role of the controller design polynomial, $C_c(q^{-1})$ is the same as for Generalized Predictive Control. Therefore, the recommendations given in the previous chapter are also valid for Pole Placement.

d) As with GPC, the PP strategy benefits by having a model which matches the true process at relatively low frequencies. $C_e(q^{-1})$ should be selected as a 2nd order polynomial corresponding to bandpass filtering of the regressor.

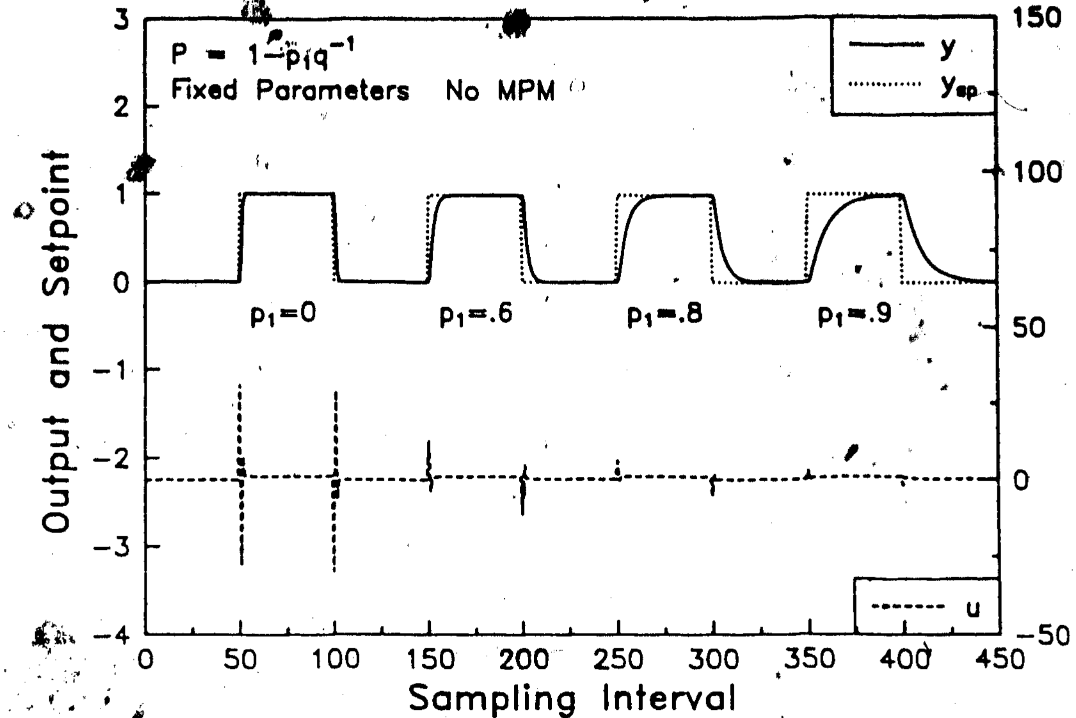


Figure 6.1 Servo Responses for Pole Placement Control of Process C

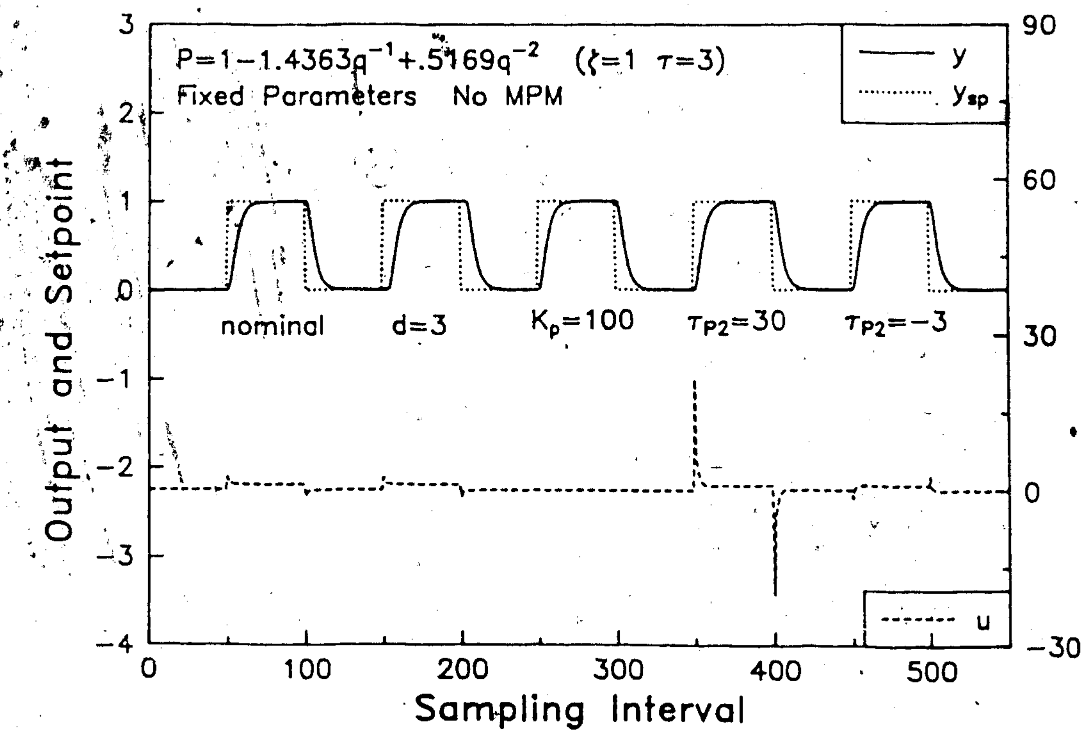


Figure 6.2 Servo Performance for Pole Placement Control of Time-Varying Process C

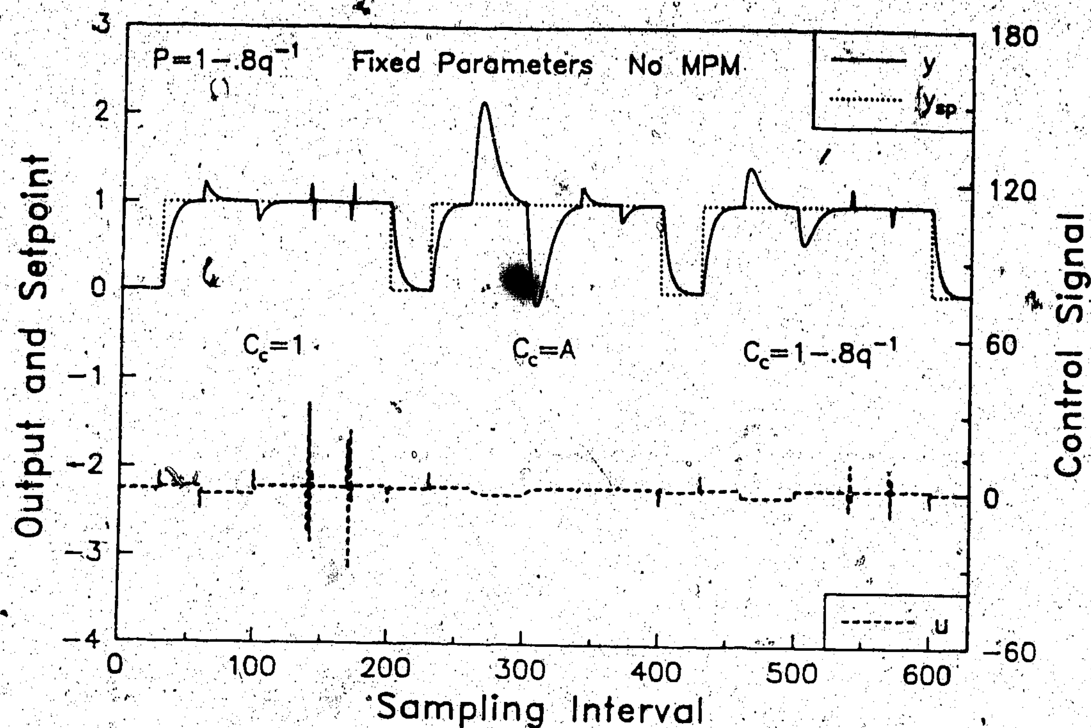


Figure 6.3 Effect of C_c Design Polynomial on Disturbance Rejection for PP Control of Process C

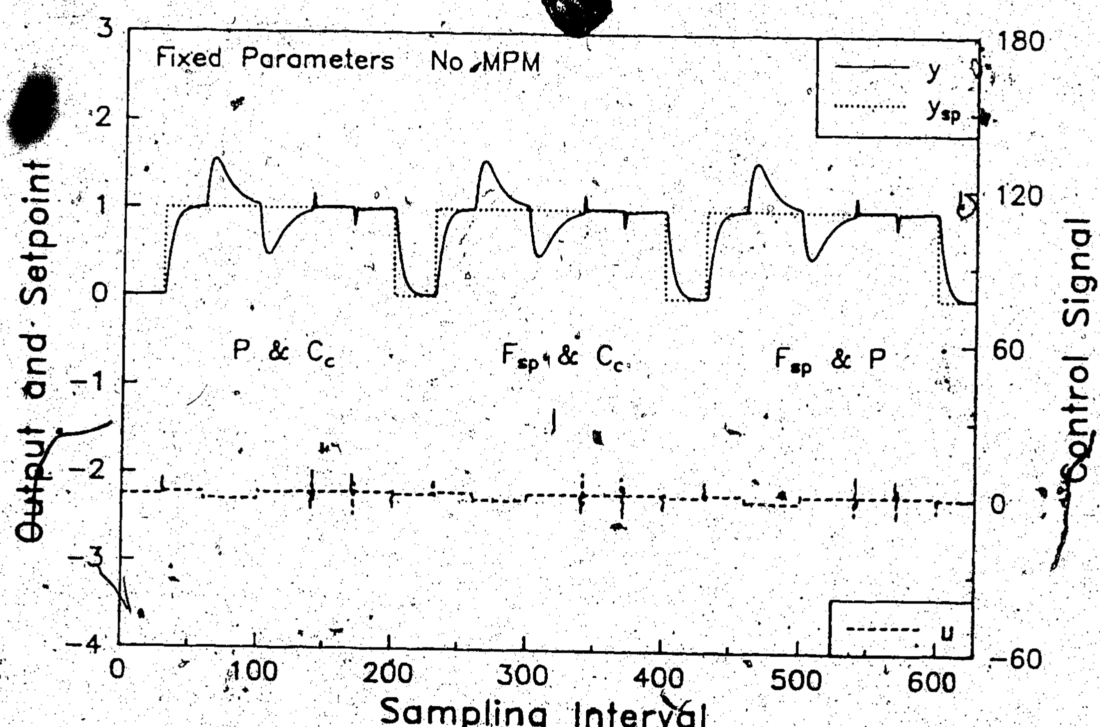


Figure 6.4 Different PP Tuning Parameter Settings Yielding Identical Servo and Regulatory Behavior

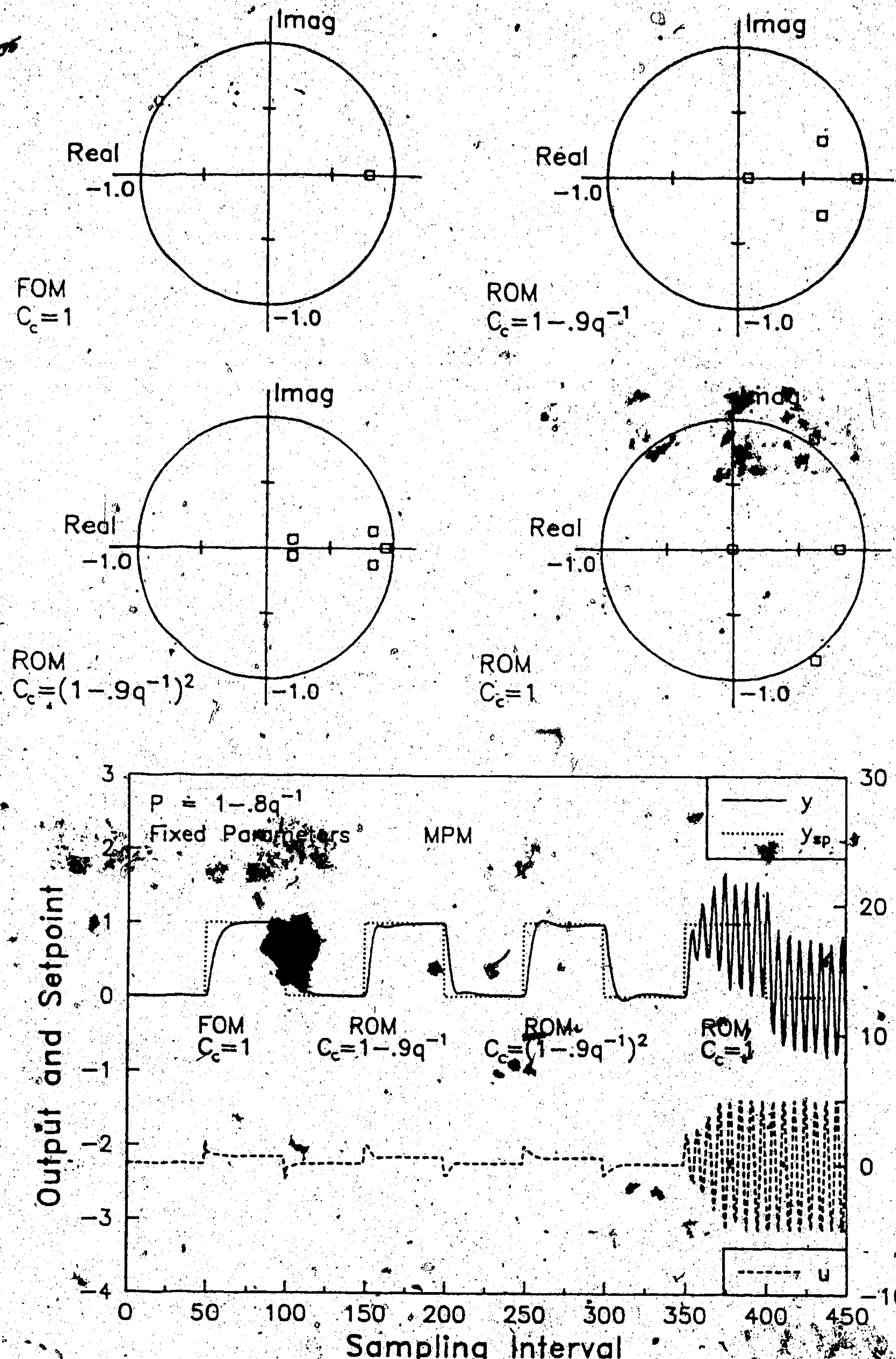


Figure 6.5 PP Control of Rohrs' Process with Unmodeled Dynamics.

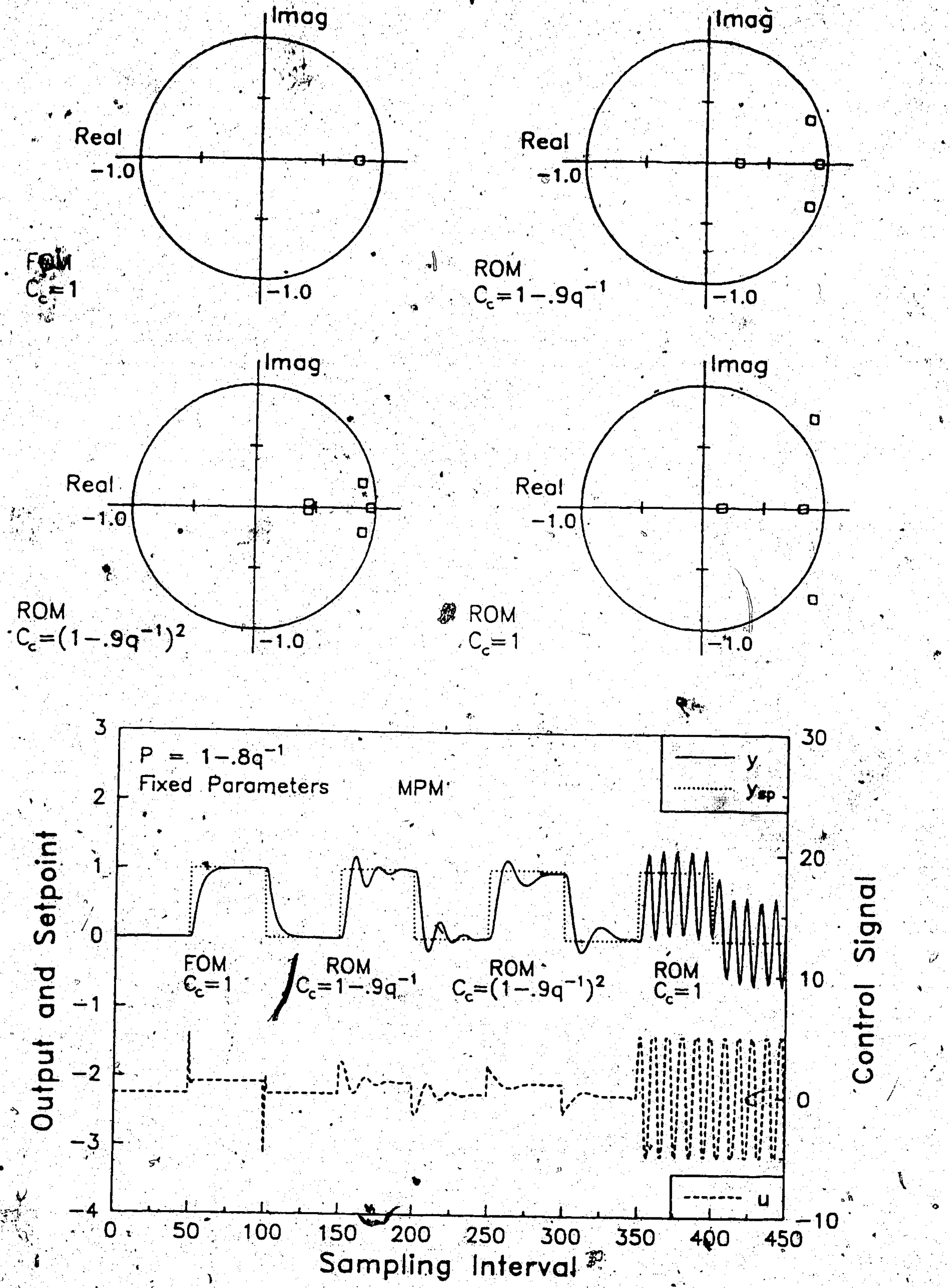


Figure 6.6 PP Control of Process C with a First Order Model

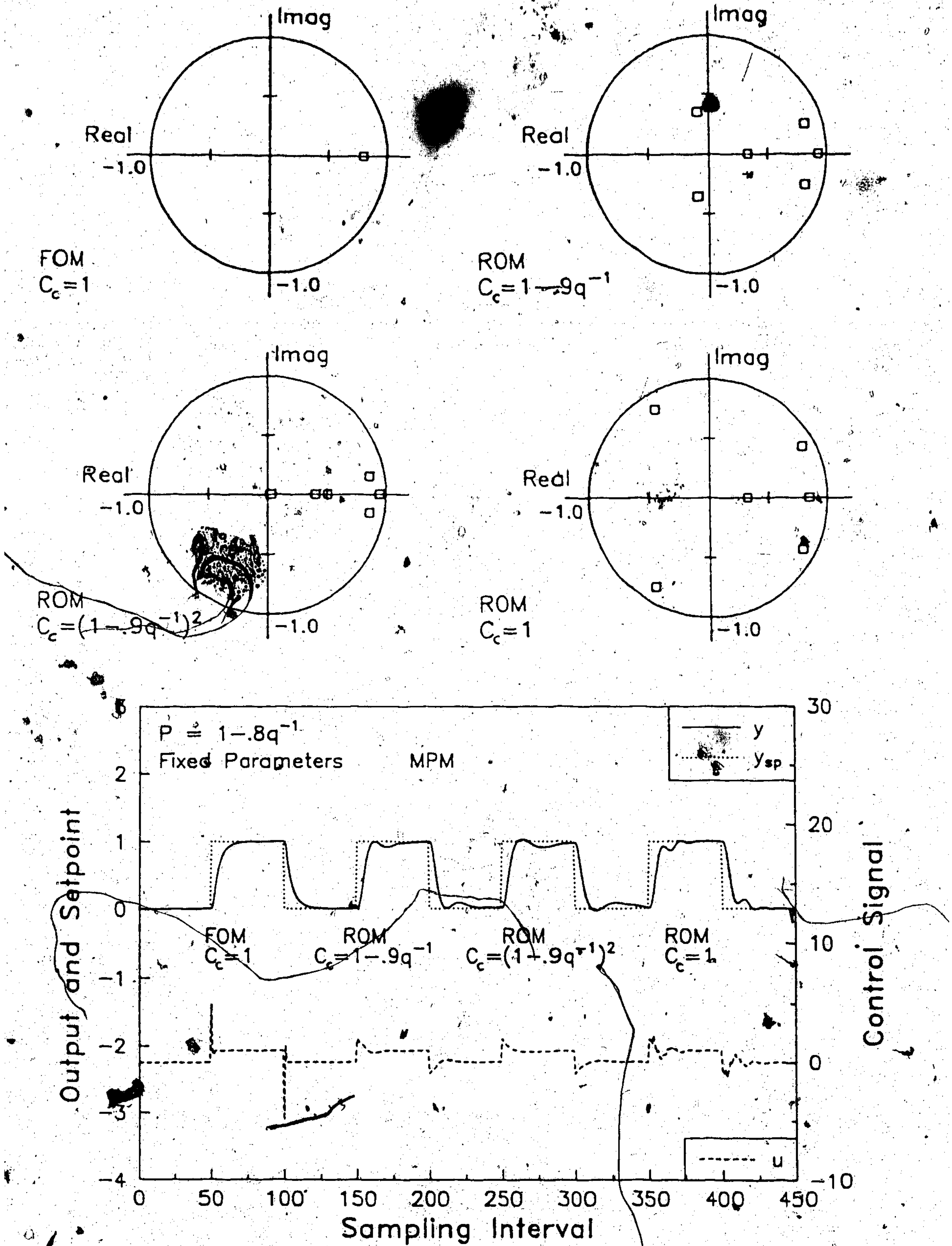


Figure 6.7 PP Control of Process C with a First Order + Time Delay Model

7. EXPERIMENTAL APPLICATION OF SELF-TUNING GENERALIZED PREDICTIVE CONTROL

While simulations provide insight into the operation of a control algorithm and the selection of its tuning parameters, the true test of an algorithm remains with its application on a real process. During experimental runs the behavior of the controller is influenced by unmeasurable events as well as unknown elements of the plant. While this implies that no two tests are precisely comparable, in a pilot plant environment care can be taken to ensure that all runs in a set take place under similar conditions.

Experimental work with two pilot plants is included in this Chapter. The Stirred Tank Heater apparatus allows evaluation of GPC on a relatively linear process with predominately first order dynamics. However, as described in section 7.1, depending upon the method used to introduce cold water, the process can be far from ideal with a large amount of measurement noise. The time delay can be varied either by measuring the outlet water temperature at different distances from the tank or by changing the water flowrate.

The Interacting Tanks, detailed in section 7.4, may be arranged to form a 2nd order nonlinear process better suited for evaluating the ability of a self-tuning controller to maintain desired performance inspite of changing process conditions. While known nonlinearities can, in principle, be handled by a form of gain-scheduling, from the point of view of the controller, a nonlinear process under changing operating conditions behaves in a manner similar to a time-varying process.

The real-time computer system and other implementation details are described briefly in section 7.2. The experimental work in sections 7.3 and 7.5 includes both servo and regulatory control. This is in recognition of the fact that setpoint changes are a relatively rare occurrance in the chemical industries yet much of self-tuning control literature deals exclusively with the servo problem.

7.1 Stirred Tank Heater

The Stirred Tank Heater equipment is illustrated schematically in Figure 7.1. Cold water enters the glass tank, is heated with steam and flows out through a long copper tube with thermocouples located at varying distances from the tank. Several different feedback and feedforward configurations are possible using the apparatus. For the SISO computer-controlled loop, the temperature of the outlet water and the steam flowrate were selected as the controlled and manipulated variables, respectively. A pneumatic proportional regulator is in place to maintain constant liquid level in the tank by altering the outlet flow of water. Manually changing the inlet cold water flowrate is a convenient method of introducing a disturbance. Note that the offset due to proportional control results in (small) variations in level when disturbances of this type are introduced. Note also that the time delay present in the system is inversely proportional to the inlet cold water flowrate. Standard operating conditions for the Stirred Tank Heater are listed in Table 7.1.

Table 7.1 Stirred Tank Heater Nominal Operating Conditions

Inlet Water Temperature	5 °C
Inlet Water Flowrate	53 cm^3/s
Water Level	24 cm
Steam Valve Opening	50 %
(Resulting) Outlet Water Temperature	35 °C

Each of the four thermocouples operate over the range 22-54 °C.

While the tank is equipped with a mixer, it cannot be considered "well-mixed"; the inlet water plunges directly into the tank on the same side as the outlet tubing connection. However, when a deflector is put in place to direct the water to the wall of the tank away from the outlet tubing connection, the resulting temperature measurement is free of noise.

7.1.1 Open-Loop Characterization

Figure 7.2 shows that open-loop response of the outlet temperature measured using T/C #1 to step changes in steam valve position (50-57%) and inlet water flowrate (53-66 cm^3/s) occurring at 200 s intervals. Note that the inlet water flowrate is not plotted in the figure. The magnitude of the disturbance is that used throughout this thesis. For the first half of the run the inlet water deflector was in place; the first order response to the manipulated and disturbance variables is expected from a simple mathematical model of the precess (Stephanopoulos, 1984). The time constant of the process, τ_p, is approximately 48 s while the process and disturbance static gains are $K_p \simeq 0.85$ °C/% and $K_d \simeq -0.50$ °C/(cm³/s), respectively. At steady state the standard deviation of the measurement signal is 0.08 °C. For the second half of the test no inlet water deflector was used; the response of the system no longer approximates first order and the standard deviation of the temperature signal has increased dramatically to 0.6 °C.

With the first thermocouple (T/C #1) the physical dead-time is relatively small. Measuring the temperature using T/C #2 located further from the tank exit introduces a large transportation time delay as indicated in Table 7.2.

Table 7.2 Time Delays associated with the Stirred Tank Heater

Thermocouple	Inlet Water Flowrate	
	53 cm³/s	66 cm³/s
#1	4 s	3 s
#2	40 s	31 s

7.1.2 Selection of Sampling Interval

Discrete sampling at a rate of 1/10 to 1/4 of the dominant time constant or pure time delay (whichever is larger) is frequently recommended in the literature (Stephanopoulos, 1984; Isermann, 1981). A sample time, T_s, of 8 s was selected for all trials with the Stirred Tank Heater. This sampling interval is approximately 1/6th of the process time constant and 1/5th of the time delay when using T/C #2.

7.2 Implementation Details

Implementation of self-tuning control requires a more sophisticated approach than conventional process control algorithms. In either case, a real-time computer system capable of multi-tasking is a prerequisite. After describing the hardware and software used for the experimental work a few other practical aspects of self-tuning control will be mentioned.

7.2.1 Computer System and Applications Software

The control algorithms studied were programmed in FORTRAN 77 on the Hewlett Packard 1000 21MX E-Series computers in the University of Alberta's Data Acquisition Control and Simulation Centre. The RTE-6/VM operating system provides a full multi-tasking environment. A typical application involving real-time data acquisition and computing, file I/O, a graphical operator display and operator interrupts enabling parameter changes has been described (McIntosh and Yegneswaran, 1986). The experimental runs presented in this and subsequent chapers were obtained using a simplified scheme. An independent operator communications program is used to modify a configuration disk file containing tuning parameters and desired operating conditions. The real-time data acquisition and control task reads the configuration file when the operator changes have been completed.

Approximately 17 pages of FORTRAN source code was required to program the Generalized Predictive Control algorithm. (The real-time Pole Placement program used in the next chapter consisted of 11 pages of source code.) In contrast, 4 pages of code made up the fixed gain PID controller used for comparison.

7.2.2 Initialization

The same procedure was followed for startup of both the self-tuning GPC and PP controllers. During the first 25 sampling intervals the control signal was manually switched between user-specified values (every 5 iterations). This was sufficient for the parameter estimation to arrive at a reasonable model. Bumpless transfer to closed-loop control is automatically achieved due to the incremental nature of the control laws.

The controllers are protected from "integral windup" simply by limiting the control signal to the range between saturation limits (normally 0-100%). The incremental control signal actually applied to the plant is used in the calculations at the next sampling instant.

7.2.3 Parameter Estimation

The constant trace Recursive Least Squares algorithm, described in section 5.1.5, was used for parameter estimation during all experimental runs. To provide rapid initialization, the trace of the covariance matrix was inflated by a factor of 10 for the first 25 sampling intervals during which time the manual step changes in the control signal provided excitation. The covariance matrix was also inflated for 10 iterations following every setpoint change (with the expectation that the data would be persistently exciting during this period).

7.3 Generalized Predictive Control of the Stirred Tank Heater

The experimental runs with the Stirred Tank Heater were conducted with three objectives in mind. The first set are intended to demonstrate the effect of the regressor filtering polynomial, C_e, and the regulatory design polynomial, C_e, on the robustness and behavior of GPC in the presence of model-plant mismatch, disturbances and measurement noise. Next the ability of the three overall GPC tuning strategies to provide a full range of desired response times (closed-loop bandwidths) is evaluated. Finally, maintenance of performance is demonstrated for a trial with variable dead-time. The complete set of experimental runs with the Stirred Tank Heater are summarized in Table 7.3.

Table 7.3 GPC Experimental Runs with the Stirred Tank Heater (Self-tuning)

Fig. No.	With/Without Deflector	Time Delay	GPC Config.	$C_e(q^{-1})$	$C_c(q^{-1})$	Purpose
7.3	with	constant	OH	1	1	effect of C_e,C_c
7.4	"	"	"	1	$1-.8q^{-1}$	"
7.5	"	"	"	$(1-.8q^{-1})^2$	$1-.8q^{-1}$	"
7.6	without	constant	OH	1	1	effect of C_e,C_c
7.7	"	"	"	1	$1-.8q^{-1}$	"
7.8	"	"	"	$(1-.8q^{-1})^2$	1	"
7.9	"	"	"	$(1-.8q^{-1})^2$	$1-.8q^{-1}$	"
7.10	"	"	"	$(1-.8q^{-1})^2$	$(1-.8q^{-1})^2$	"
7.11	with	constant	OH	$(1-.8q^{-1})^2$	$1-.8q^{-1}$	effect of N_2
7.12	"	"	LW	"	"	effect of λ
7.13	"	"	DMF	"	"	effect of $P(q^{-1})$
7.14	with	variable	PID*	-	-	time delay comp.
7.15	"	"	DMF	$(1-.8q^{-1})^2$	$1-.8q^{-1}$	"

Abbreviations: OH = Output Horizon, LW = Lambda Weighting, DMF = Detuned Model-Following

* manually tuned

7.3.1 Stabilization using Polynomials C_e and C_c

In the simulations of Chapter 5, the regressor filtering and disturbance tailoring polynomials were employed to stabilize closed-loop systems in the presence of unmodelled dynamics and used to improve the rejection of various disturbances. Here we will deal with the impact of these polynomials on self-tuning servo and regulatory control of the Stirred Tank Heater temperature measured using T/C #1. For all of the trials in this subsection, the inlet water flowrate was changed stepwise from 53 cm³/s to 66 cm^3/s at time 750 s. At 900 s the flowrate was returned to 53 cm^3/s. Output Horizon tuning parameter settings were used with N_2 set to the default value of 10. The variable forgetting factor identification algorithm maintained the trace of the covariance matrix at 0.2 (except for the 10 iterations following a setpoint change when the trace was inflated and maintained at 2.0).

With the inlet water deflector in place the process is well-mixed and relatively noise-free; a first order model is expected to provide adequate predictions in this case.

7.3.1.1 With Inlet Water Deflector

If C_e and C_c are both unity, the response of the closed-loop, shown in Figure 7.3a, is oscillatory with relatively strong control adjustments. An integrating model (i.e. $\hat{a}_1 \simeq -1$) was identified for the last half of the run as indicated by the parameter trajectories in Figure 7.3b. If the regulatory action of the controller is detun[illegible] C_c=1-.8q^{-1} the rapid oscillations in the output are almost eliminate[illegible]e 7.4). Selection of a 2nd order regressor filtering polynomial was shown earlier to correspond to bandpass filtering of the I/O data. With C_e=(1-.8q^{-1})^2 the control performance in Figure 7.5 is slightly improved compared with Figure 7.4. This is true even though the "true" noise model polynomial corresponding to the inlet water flowrate disturbances is closer to unity (since this is an "input" type disturbance).

7.3.1.2 Without Inlet Water Deflector

If the Stirred Tank Heater is operated without the inlet water deflector the control task is made much more difficult; a large amount of noise obscures the character of the process. Consider again GPC with C_e=C_c=1; results are shown in Figure 7.6. The identification algorithm fails to come up with a model suitable for long-range prediction. At 620 s the sign of the estimated parameter, $\hat{b}_o$, becomes negative and the controller saturates permanently. Even if the rejection of disturbances is detuned, by setting C_c to 1-.8q^{-1}, the problem of a poor low frequency model persists (Figure 7.7). The estimated process gain is consistently far too low (compared with the process gain determined earlier based on open-loop step tests). For example, after the startup sequence, $\hat{K}_p = \hat{B}(1)/\hat{A}(1) \simeq$ 0.[illegible] °C/% whereas the open-loop step tests (section 7.1.1) gave $K_p \simeq 0.85$ °C/%.

If the incremental regressor is filtered by $1/C_e = 1/(1-.8q^{-1})^2$ the initial parameter estimates shown in Figure 7.8b provide a reasonable low frequency model. After 600 s, the RLS algorithm identifies an unstable model ($\hat{a}_1 \approx -1.1$). The controller is very active throughout (with $C_c = 1$) and eventually drives $\hat{b}_o$ to a negative value. Detuning the (regulatory) controller with $C_c = 1-.8q^{-1}$ rectifies the problem as demonstrated in Figure 7.9. The identified model gives good long-range predictions and is not corrupted by the inlet water flowrate disturbances (occurring at 750 and 900 s). If the same polynomial is used in the control law as for regressor filtering ($C_c = (1-.8q^{-1})^2$) the controller's rejection of disturbances is further detuned (Figure 7.10). The output performance is degraded slightly in comparison to the previous run.

It is important to note that it was necessary to select both the estimator filter, $C_e(q^{-1})$, and controller design polynomial, $C_c(q^{-1})$, as outlined earlier in section 5.5, to achieve good results.

7.3.2 Altering the Speed of Response

Once stability of the closed-loop system has been obtained by properly selecting any parameters associated with the identification algorithm and by using the design polynomials $C_e(q^{-1})$ and $C_c(q^{-1})$, attention can be turned to adjusting the overall (servo and regulatory) speed of response of the closed-loop. Each of the three tuning strategies conveniently allow GPC to achieve a full range of response times. For the experimental runs in this subsection, step changes in the inlet water flowrate (from 53 cm^3/s to 66 cm^3/s and back again) were introduced during the period when the setpoint was 40°C. Just prior to each upward-going SP change, the active tuning parameter was adjusted to increase the [illegible] response. The variable forgetting factor RLS algorithm maintained the trace of the covariance matrix at 0.02 with bandpass filtering of the regressor using [illegible]. The initialization sequence and the parameter trajectories are not pr[illegible] (as they add little information).

The results for the tuning strategy based on the Output Horizon configuration are shown in Figure 7.11. Note that the smallest recommended value for N_2 is 2 for this process with physical time delay less than one sampling interval.

For Lambda Weighting control it was recommended in section [illegible] that the trace of $G_r^T G_r$ be used as a guide in the selection of λ. [illegible]tial parameter convergence, $|tr(G_r^T G_r|/NU \simeq 3.0$. The servo and [illegible]atory response of the system is shown in Figure 7.12 for λ = 3.0, 0.3 [illegible] 0.03. The response is slightly more oscillatory, (not as "tight") compared with Output Horizon control.

Finally, the results for Detuned Model-Following are presented in Figure 7.13. The selected values of p_1 (0.8, 0.7 and 0.4) correspond to desired closed-loop time constants of 35s, 22s and 9s, respectively. Note that a first order closed-loop model is particularly suitable for this predominantly first order process.

7.3.3 Handling Variable Dead-time

A change in time delay may be introduced by switching between thermocouples during an experimental run. At a water flowrate of 53 cm^3/s, switching between T/C #1 and #2 changes the dead-time from approximately 4 s to 40 s. Such a large variation in time delay destabilizes fixed parameter as well as some self-tuning controllers (GMV, for example). As a simple demonstration, a PID controller was tuned by trial and error to give an overdamped response to setpoint changes and inlet water flowrate disturances (occurring at times 250 s and 500 s) using T/C #1, as shown in Figure 7.14. When the larger dead-time is introduced, the servo and regulatory response is very oscillatory (disturbances entered at 1400 and 1700 s).

Compensation for the variable time delay with GPC results from over-parameterizing the $\hat{B}(q^{-1})$ polynomial. The maximum expected physical time delay is 5 sampling intervals ($d_{max} = t_d/T_s$ = 40s/8s = ~5). Estimating 7 b parameters ($\hat{b}_0$ through $\hat{b}_6$) allows for any remaining fractional dead-time. Any of the three configurations of GPC may be applied. Consider Detuned Model-Following with $P=5-4q^{-1}$, (corresponding to a desired closed-loop time constant of 35 s). When the larger time delay is introduced a transient of approximately 40 sampling intervals (Figure 7.15a) is required to arrive at an updated parameter set. (For clarity, in Figure

7.15b only the parameters $\hat{b}_0$ and $\hat{b}_s$ are plotted. As expected, $\hat{b}_s$ hovers near zero when there is little time delay and $\hat{b}_0$ tends to zero when the time delay increases.) The inlet water flowrate step disturbances introduced at 750, 900, 1750 and 2050 s are rejected smoothly in the presence or absence of the additional transportation delay.

7.4 Interacting Tanks

The interacting tank apparatus consists of a pair of glass vessels connected by a manually adjustable valve (resistance), as shown schematically in Figure 7.16. The level in the first "conical" tank and the inlet water flowrate were selected as the controlled and manipulated variables, respectively. For operation with the level above 18cm the diameter is constant; a simple mass balance (Stephanopoulos, 1984) indicates that the transfer function between the inlet water flowrate and the level in the first tank is second order (overdamped) with a zero in the continuous-time domain (i.e. pseudo first order). For levels in the first vessel below 18cm, the diameter decreases linearly. Highly nonlinear behavior is observed for operation in this conical section. Process variations were introduced by changing the resistance between the tanks. Since altering the resistance between the tanks in open-loop causes an immediate change in liquid level, with a slight abuse of terminology, this change will be referred to as a "disturbance". Standard operating conditions for the interacting tanks may be found in Table 7.4.

Table 7.4 Nominal Operating Conditions for the Interacting Tanks

Inlet Water Pump Speed	85 %
Inlet Water Valve Position	52 %
(Resulting) Conical Tank Level	18 cm

7.4.1 Nonlinearity

The open-loop response of the conical tank level to step changes in the control signal to the inlet water valve is plotted in Figure 7.17. The control valve is air-to-close resulting in a negative overall process gain. The control signal followed the sequence 52%, 54%, 60%, 75%, 60%. At the 1900 second mark, the standard disturbance was introduced by increasing the resistance between the tanks.

The highly nonlinear nature of the system is evident form Figure 7.17. Based on this and other open-loop experiments, approximate process gains and dominant time constants are listed as functions of liquid level changes in Table 7.5.

Table 7:5 Nonlinearity of the Interacting Tanks

Conical Tank Level	Process Gain, K_p	Dominant Time Constant, τ_p
12-18 cm	-5 cm/%	200 s
9-12 cm	-.5 cm/%	40 s
6-9 cm	-.15 cm/%	15 s

The pure time delay of the process was determined to be under 2 s. Note that the conical shape of the first vessel gives rise to directionally-varying dynamics.

7.4.2 Sampling Interval

It is obviously difficult to follow recommendations in the literature for selection of the sampling interval when there exists large variations in the response time of a process. A sample time of 8 s was felt to be a reasonable compromise. This interval represents 1/25th of the largest expected dominant time constant and one half of the smallest.

7.5 Generalized Predictive Control of the Interacting Tanks

In Chapters 4 and 5, it was shown theoretically and demonstrated via siumulations that each of the three configurations of GPC is capable of maintaining output performance; the response of the controlled variable is invariant to process gain changes and insensitive to changes in dynamics provided there are no modelling errors. The experimental trials with the Interacting Tanks were conducted in order to evaluate the ability of self-tuning GPC to maintain output performance on a time-varying or nonlinear process. [illegible] parameter estimation algorithm must remain alert at all times to follow changing conditions.

For conciseness, Detuned Model-Following control of the Interacting Tanks is not discussed; results were similar to that of Output Horizon control. The experimental runs with the Interacting Tanks are summarized in Table 7.6.

Table 7.6 GPC Experimental Runs with the Interacting Tanks (Self-tuning)

Fig. No.	GPC Configuration	$C_e(q^{-1})$	$C_c(q^{-1})$	Purpose
7.18	OH	$(1-.8q^{-1})^2$	$1-.8q^{-1}$	maintenance of performance
7.19*	LW,λ=constant	"	"	"
7.20	LW,$\lambda \propto [\hat{B}(1)]^2$	"	"	"
7.21	PID	-	-	"

Abbreviations: OH = Output Horizon, LW = Lambda Weighting

* manually tuned

7.5.1 Output Horizon Configuration

For this configuration no explicit weighting is placed on control activity and therefore it is expected that output performance will be maintained. The closed-loop response to setpoint changes (between 18cm, 12cm and 6cm) and the disturbance entering at 2000 s is shown in Figure 7.18a. The default value of the controller regulatory design polynomial, $C_c=1-.8q^{-1}$, was employed for all GPC runs with the Interacting Tanks. A second order model was identified with the trace of the covariance matrix held constant at 80 and bandpass filtering of the regressor ($C_e=(1-.8q^{-1})^2$).

Parameter trajectories are indicated in Figure 7.18b with estimates initialized at zero. (Startup consisted of four OL step changes in the control signal followed by one CL setpoint change during the first 500 s of the run).

Comparing the servo responses it is evident that the self-tuning controller is able to maintain the closed-loop bandwidth approximately constant. Most importantly, this is being accomplished during single large setpoint changes (i.e. as opposed to over several small setpoint changes within the region of nonlinearity).

7.5.2 Lambda Weighting Configuration

The previous experimental design was duplicated with the basic tuning parameters set for Lambda Weighting control. At intermediate liquid levels ($\simeq$12cm), tuned model parameter estimates resulted in $|tr(G_r^T G_r| \simeq 6$. A value of λ=1.0, therfore, is expected to yield control actions roughly 2/3 as large as for deadbeat control. Figure 7.19a confirms the analysis in Chapters 4 and 5 which indicated that a constant value of λ is unsuitable for maintaining output performance, when there are large gain and dynamic changes.

The estimated process gain, $\hat{K}_p = \hat{B}(1)/\hat{A}(1)$, and the sum of the $\hat{B}(q^{-1})$ polynomial parameters, $\hat{B}(1)$, are plotted in Figure 7.19b. The identification algorithm is remaining alert and providing reasonable static gain estimates at the different operating levels. For example, with the setpoint at 18cm during the time interval from 700 to 1000 s, the gain was estimated to be -6.5 cm/%. This is relatively close to the value -5 cm/% computed from open-loop tests, listed in Table 7.5.

If the control weighting is proportional to $[\hat{B}(1)]^2$, the output performance can be made insensitive to process changes. From the previous figure, at intermediate liquid levels, parameter estimates were such that $\hat{B}(1) \simeq -0.06$. Thus $\lambda = 300(\hat{B}(1))^2$ should yield approximately the same speed of response for operation around 12cm. When the setpoint is 18cm or 6cm, λ will be automatically adjusted to compensate for the process nonlinearity. The resulting output performance is very consistent as indicated in Figure 7.20a. Again note that this approach involves a time-varying weighting on control action.

7.5.3 Comparison with PID Control

It is interesting to compare the previous results with that which can be achieved using a fixed gain PID controller. Figure 7.21 is illustrative of the problems a PID controller has in maintaining output performance. The controller was initially tuned using the Ziegler-Nichols technique (Ziegler and Nichols, 1942) with the gains adjusted by trial and error to improve the servo response at intermediate liquid levels. For the setpoint changes between 12cm and 6cm a large amount of integral action is needed in order to automatically "reset" the output (by roughly 20%). In contrast, when moving between 18cm and 12cm very little integral action is required; too much leads to the overshoot shown in Figure 7.21.

7.6 Summary

The self-tuning Generalized Predictive Controller gave very good results when applied first to a pilot scale Stirred Tank Heater with a large amount of measurement noise and a variable time delay, and second to a set of highly nonlinear Interacting Tanks. The incremental algorithm was easy to commission with automatic bumpless transfer and anti-reset windup protection.

Specification of the estimator and controller design polynomials, $C_e(q^{-1})$ and $C_c(q^{-1})$, according to the recommendations given in Chapter 5, was shown to be absolutely vital for satisfactory control of the Stirred Tank Heater. Excellent compensation for sudden changes in time delay was obtained simply by overparameterizing the $\hat{B}(q^{-1})$ polynomial. Each of the three tuning strategies of GPC could be used to vary the closed-loop speed of response.

The experimental runs with the nonlinear Interacting Tanks demonstrated that self-tuning GPC is capable of maintaining output performance despite large changes in process gain and dynamics. Therefore, it is not necessary for the user (or a supervisory system) to adjust the tuning parameters as long as the identified model is always a reasonable representation of the process.

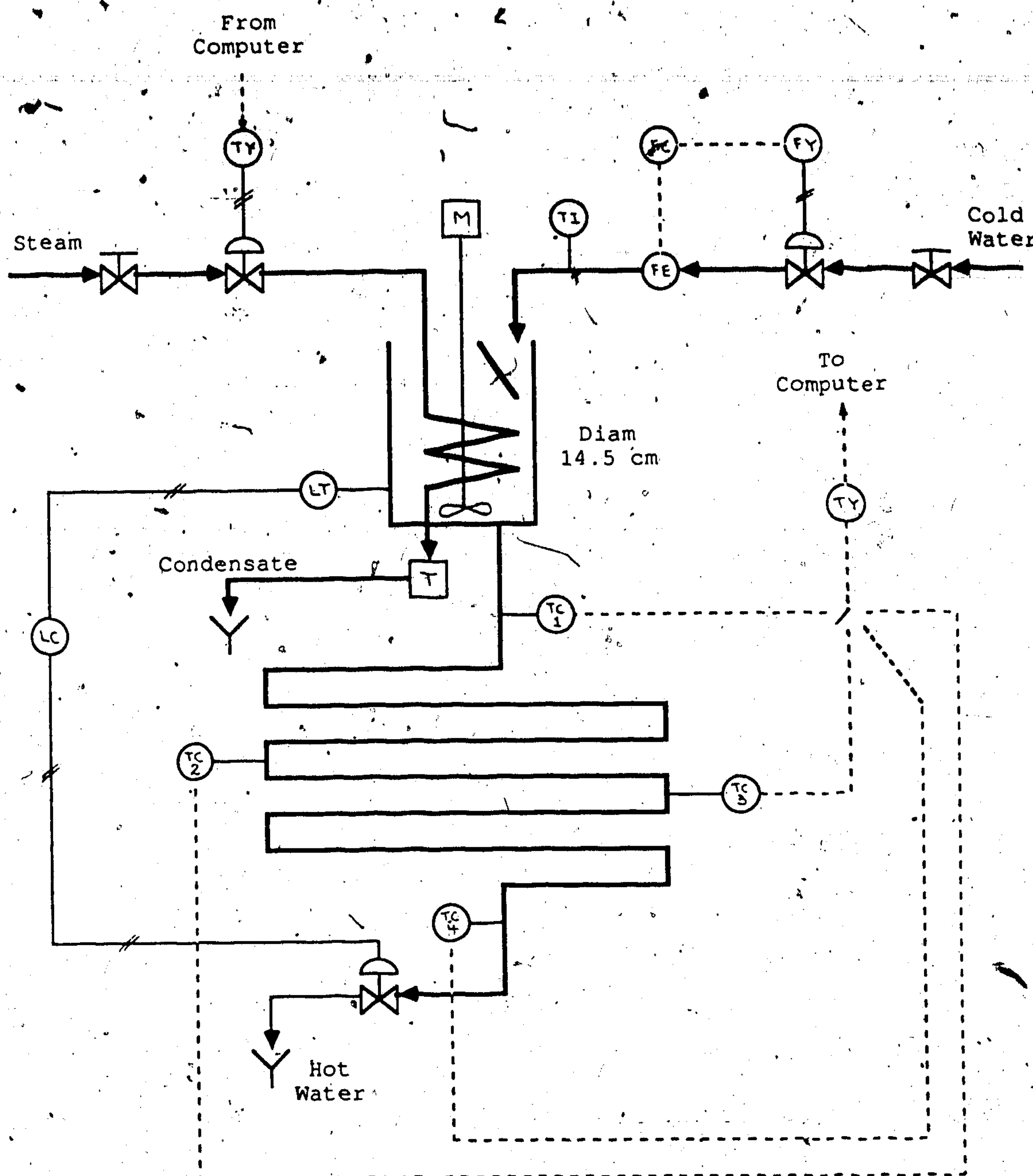


Figure 7.1 Schematic Diagram of Stirred Tank Heater

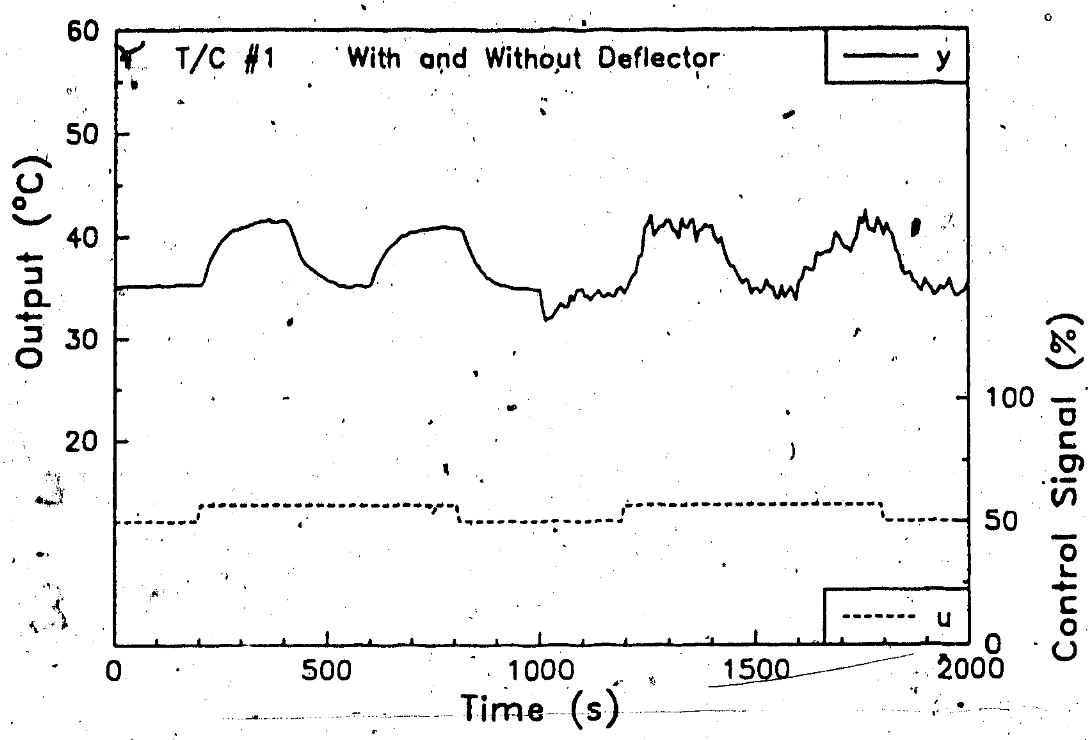


Figure 7.2 Open Loop Response of Stirred Tank Heater

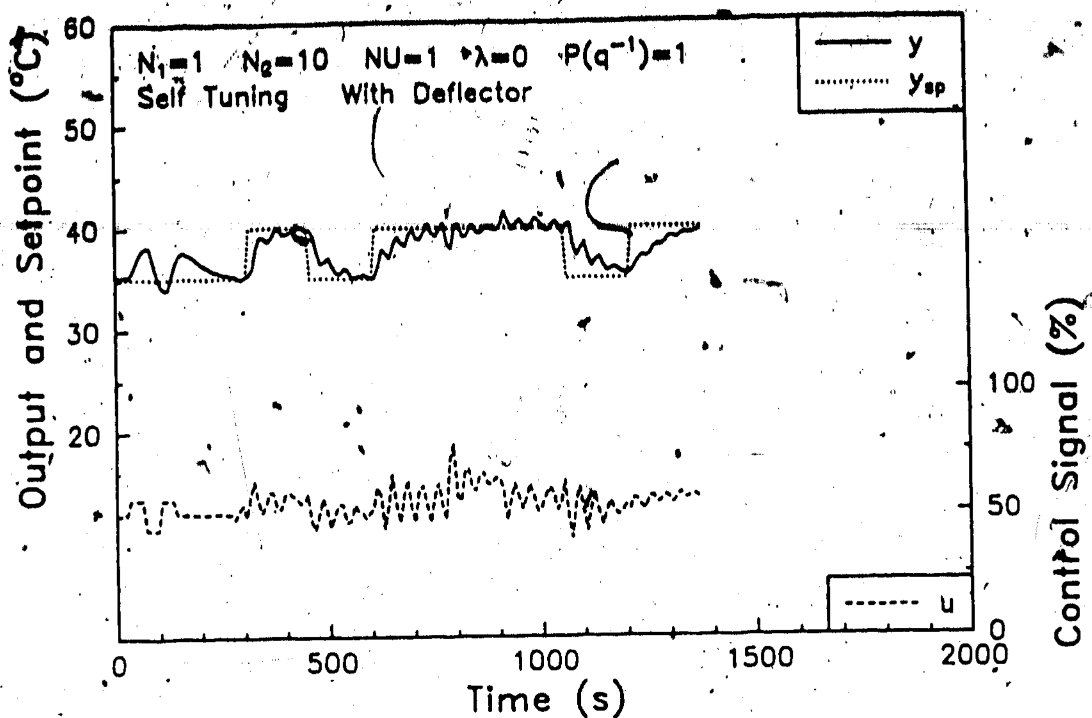


Figure 7.3a Control of the Stirred Tank Heater using GPC with $C_c=1$

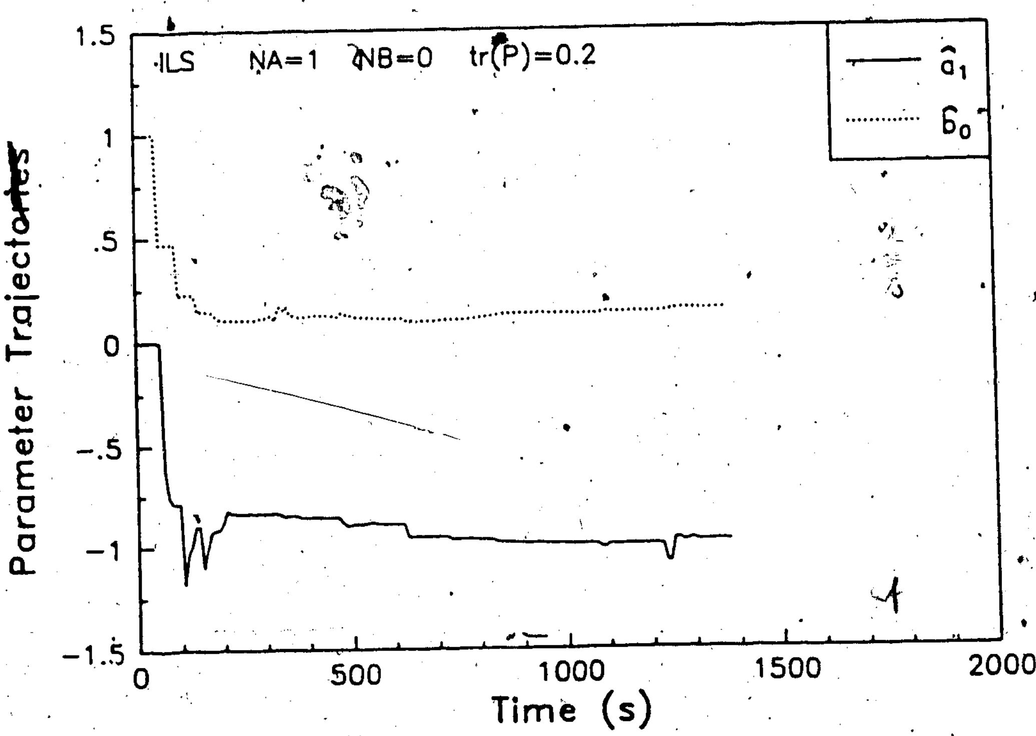


Figure 7.3b Parameter Estimation with $C_e=1$

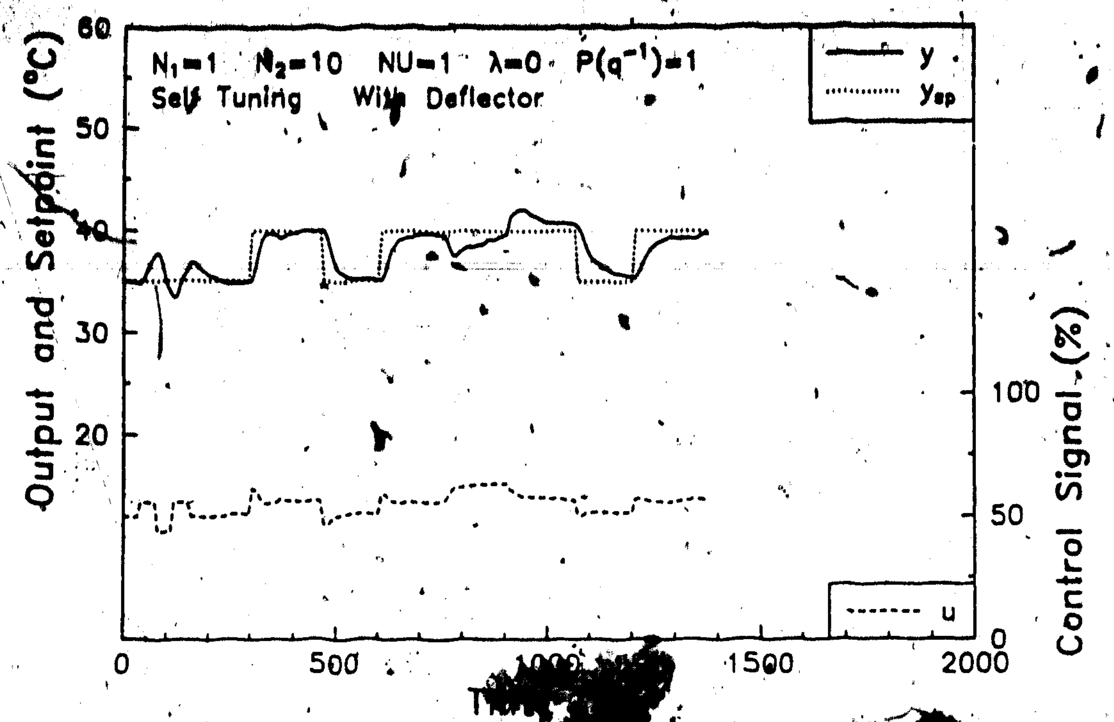


Figure 7.4a Control of the Stirred Tank Heater u[illegible]
with $C_c=1-.8q^{-1}$

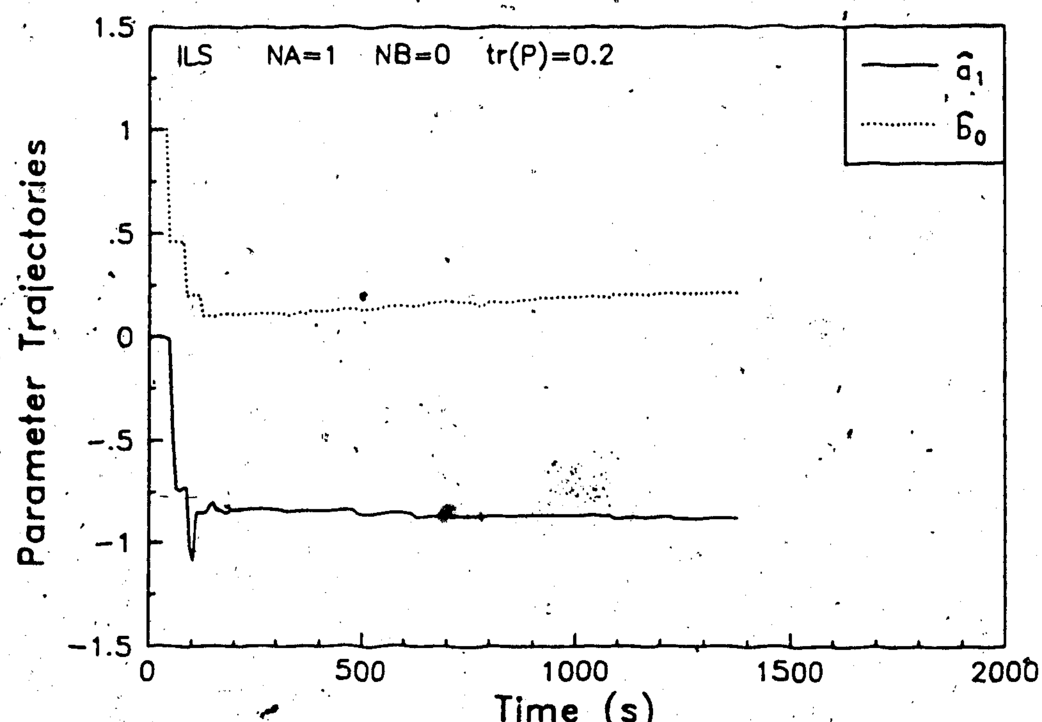


Figure 7.4b Parameter Estimation with $C_e=1$

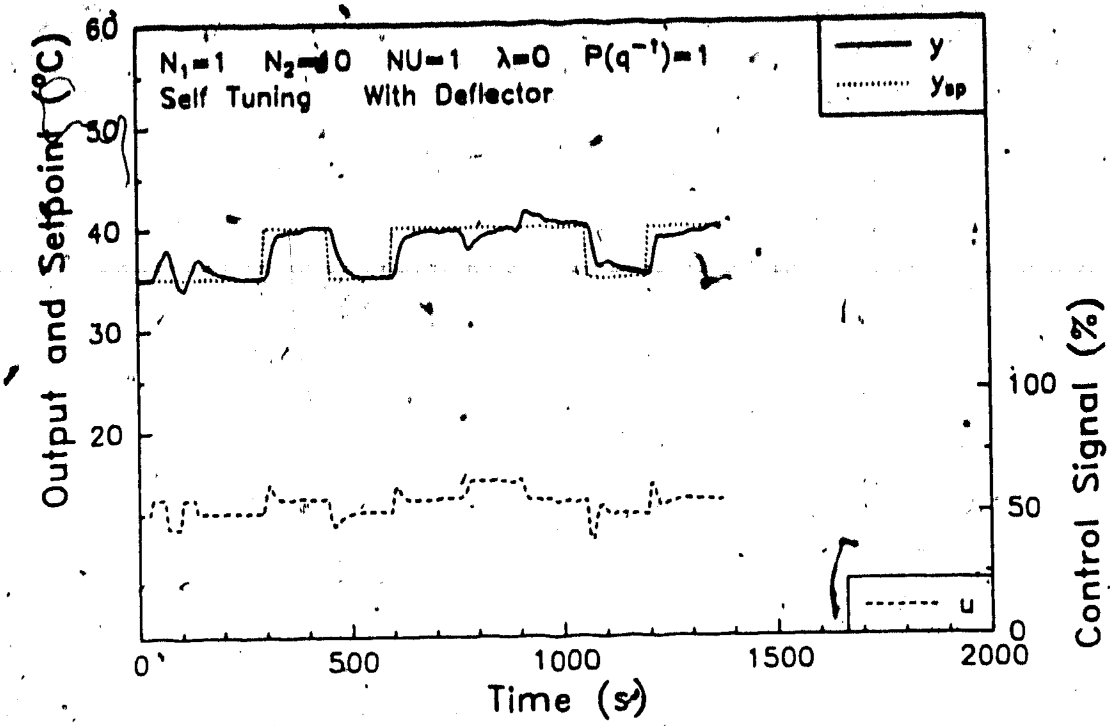


Figure 7.5a Control of the Stirred Tank Heater using GPC with $C_c=1-.8q^{-1}$

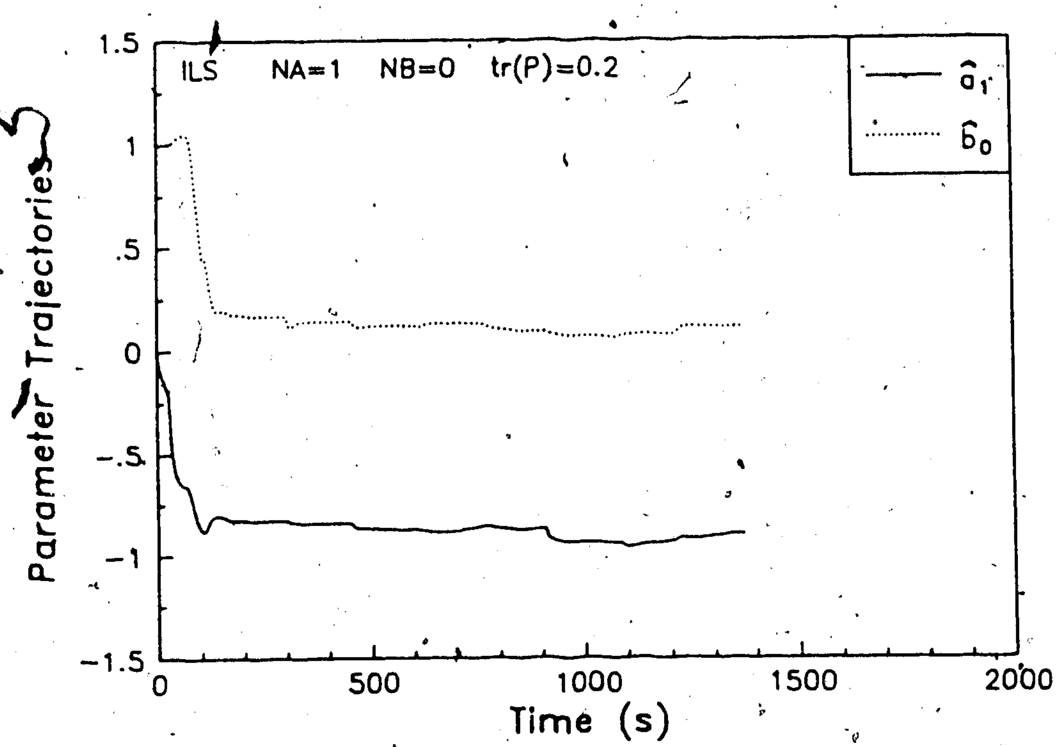


Figure 7.5b Parameter Estimation with $C_e=(1-.8q^{-1})^2$

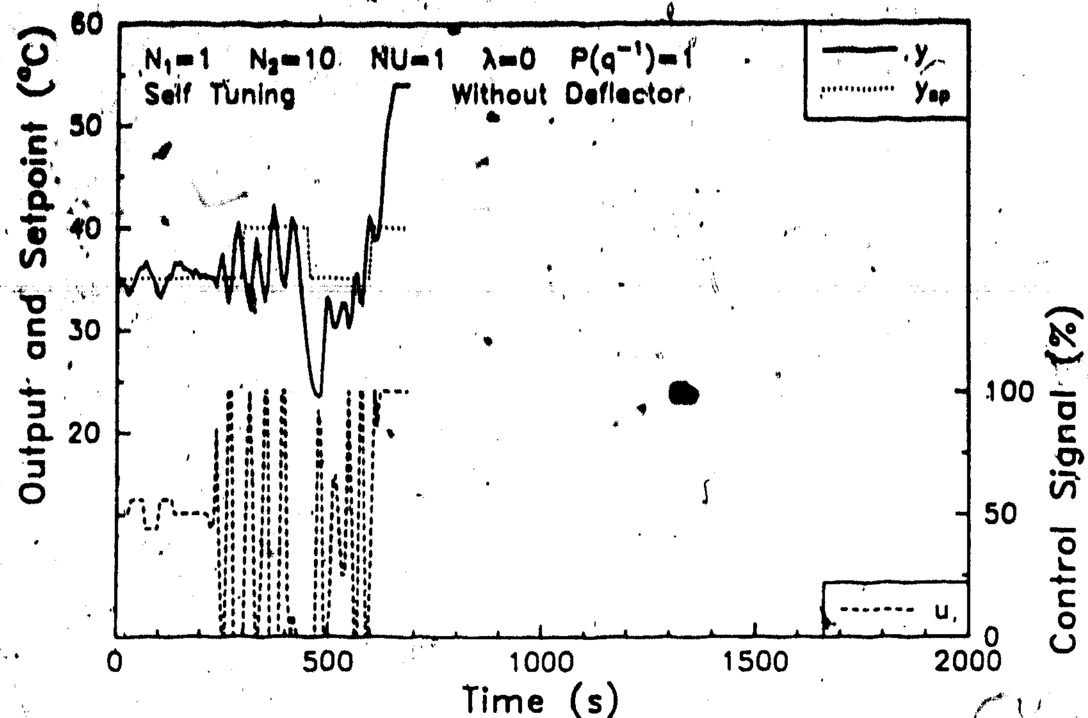


Figure 7.6a Control of the Stirred Tank Heater using GPC with $C_c=1$

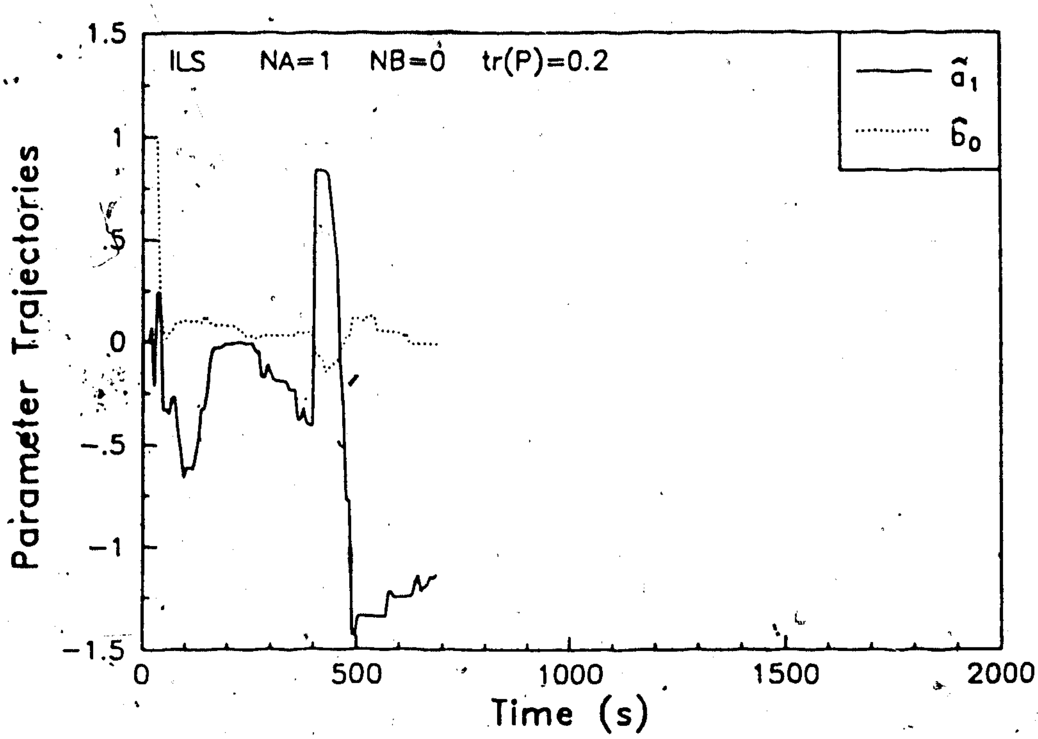


Figure 7.6b Parameter Estimation with $C_e=1$

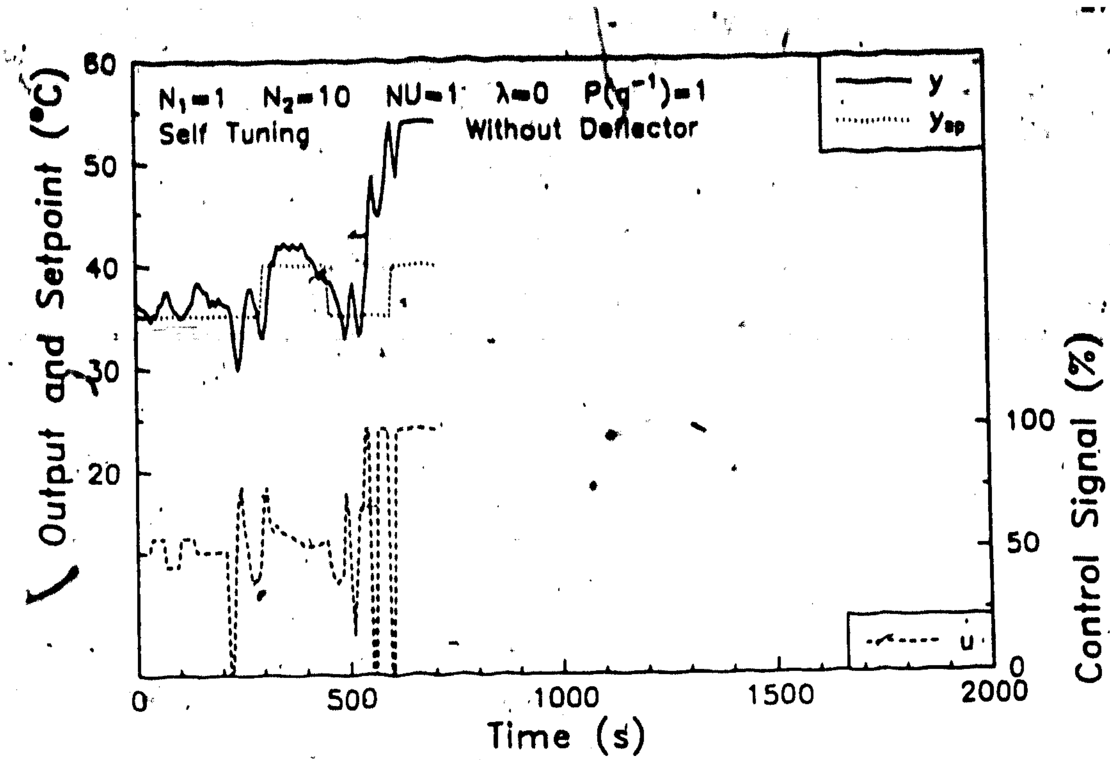


Figure 7.7a Control of the Stirred Tank Heater using GPC with $C_c=1-.8q^{-1}$

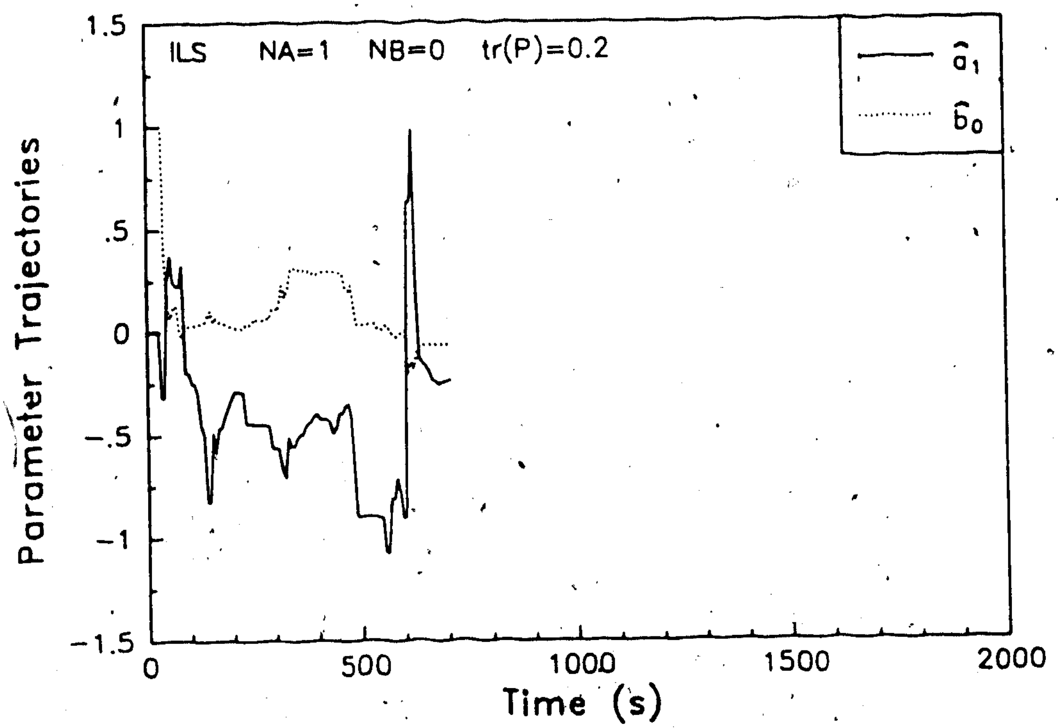


Figure 7.7b Parameter Estimation with $C_e=1$

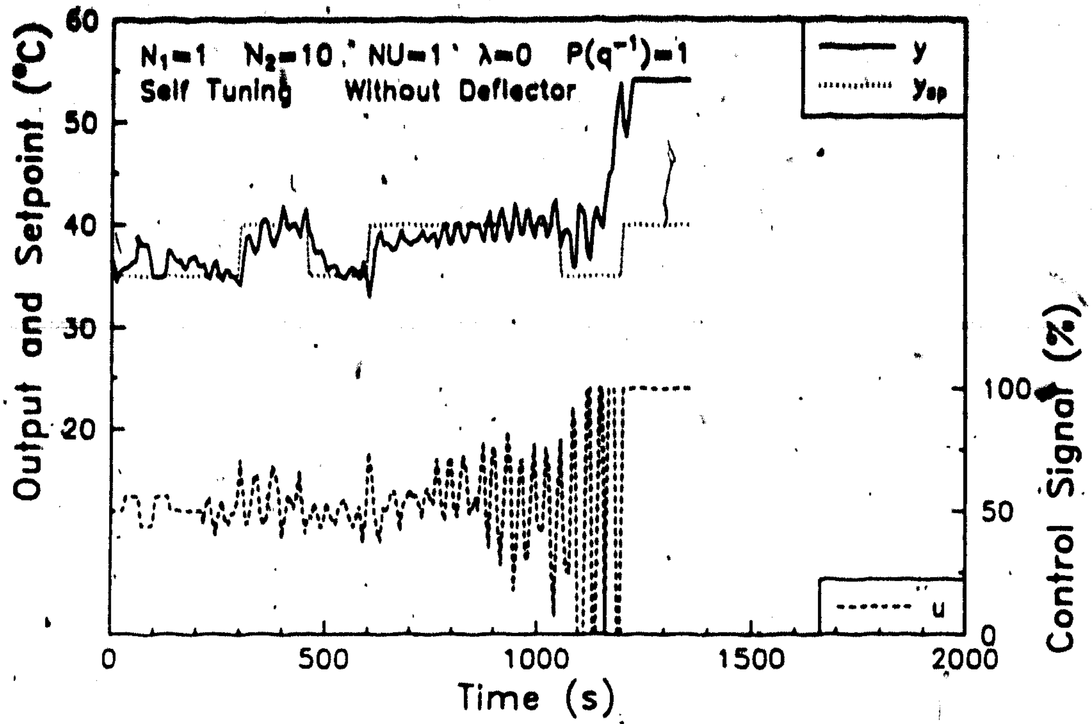


Figure 7.8a Control of the Stirred Tank Heater using GPC with $C_c=1$

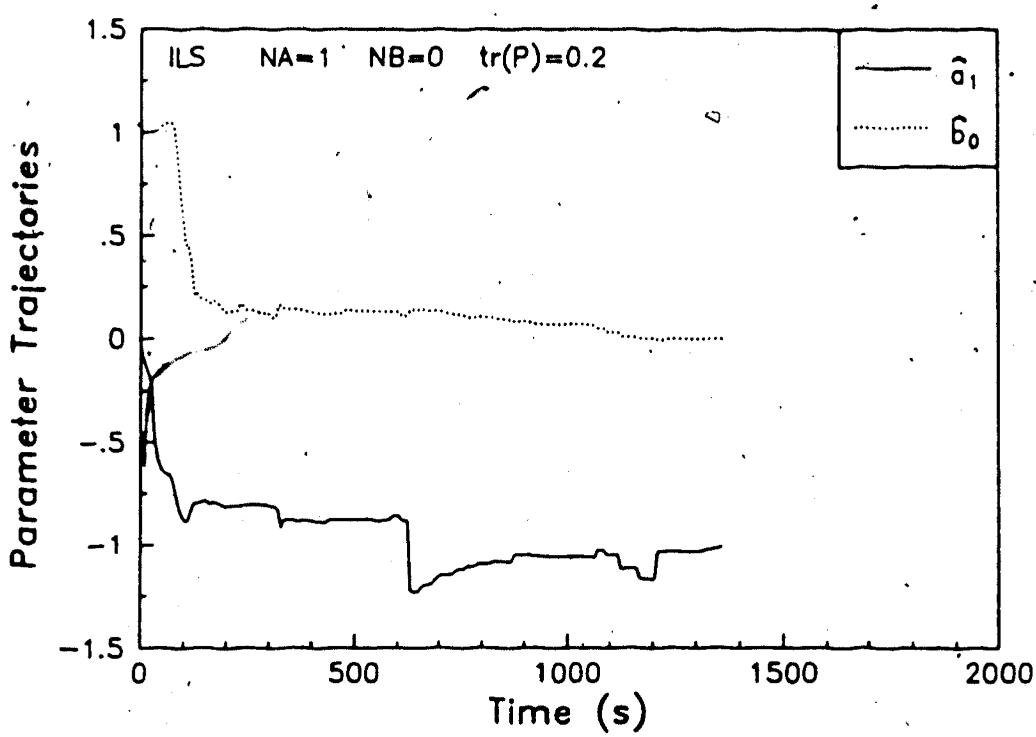


Figure 7.8b Parameter Estimation with $C_e=(1-.8q^{-1})^2$

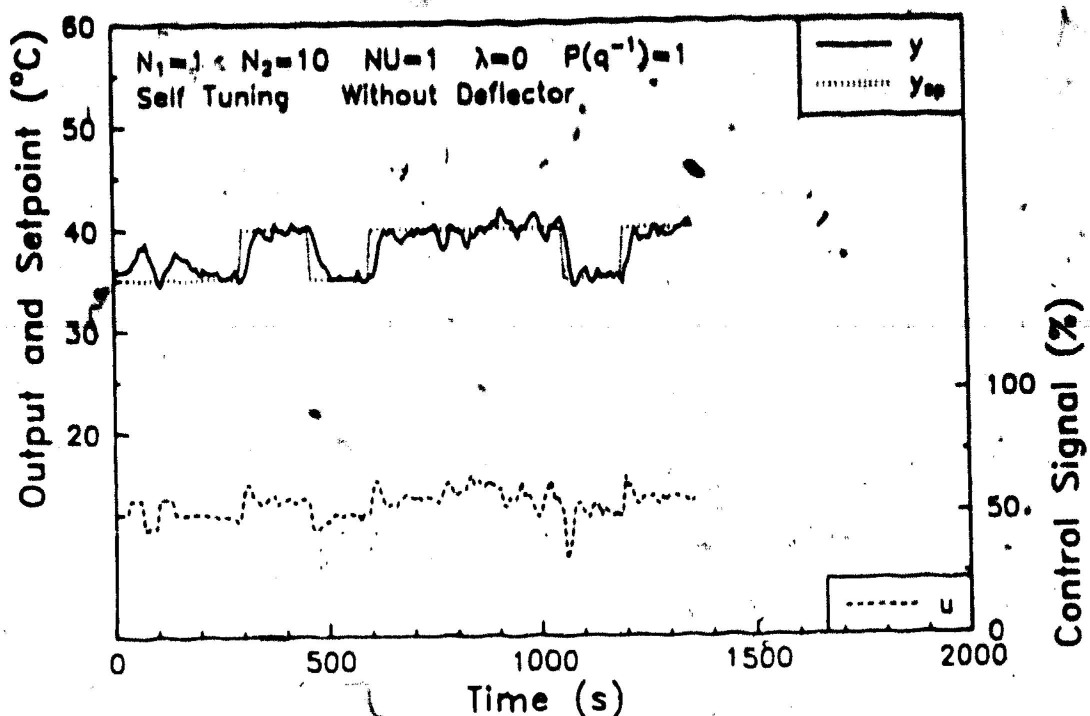


Figure 7.9a Control of the Stirred Tank Heater using GPC with $C_c=1-.8q^{-1}$

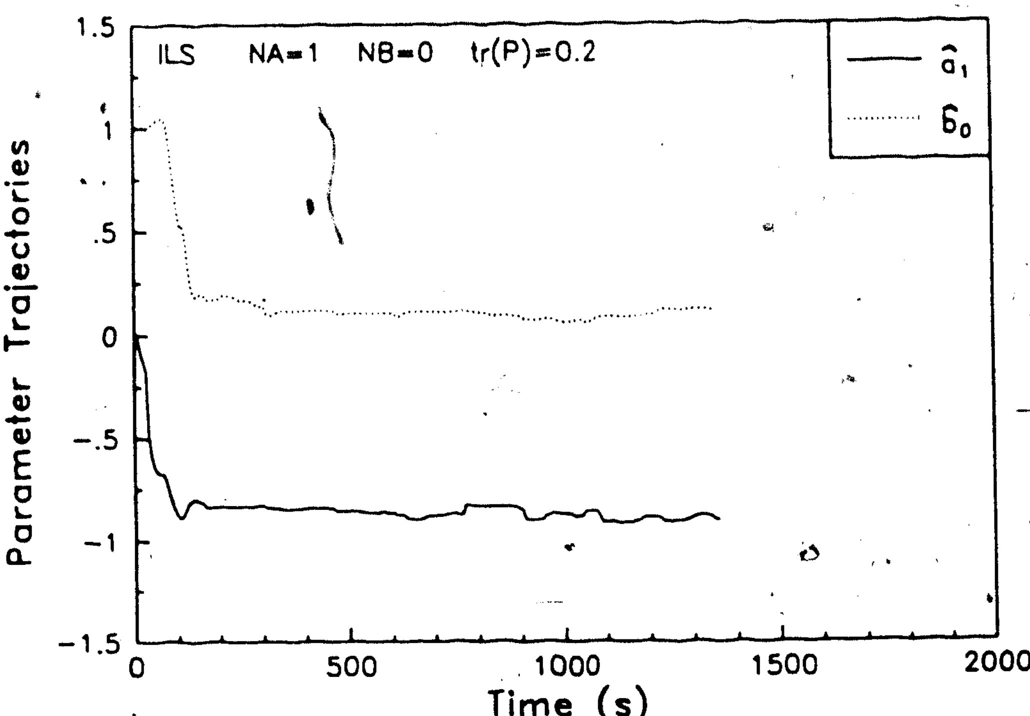


Figure 7.9b Parameter Estimation with $C_e=(1-.8q^{-1})^2$

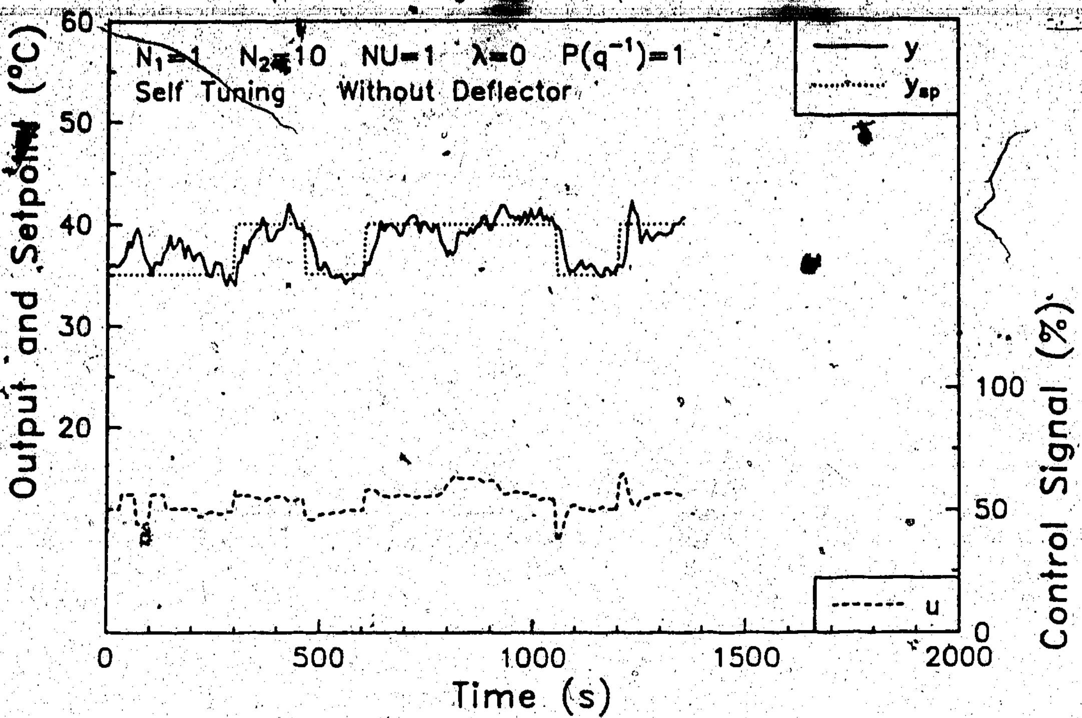


Figure 7.10a Control of the Stirred Tank Heater using GPC with $C_c=(1-.8q^{-1})^2$

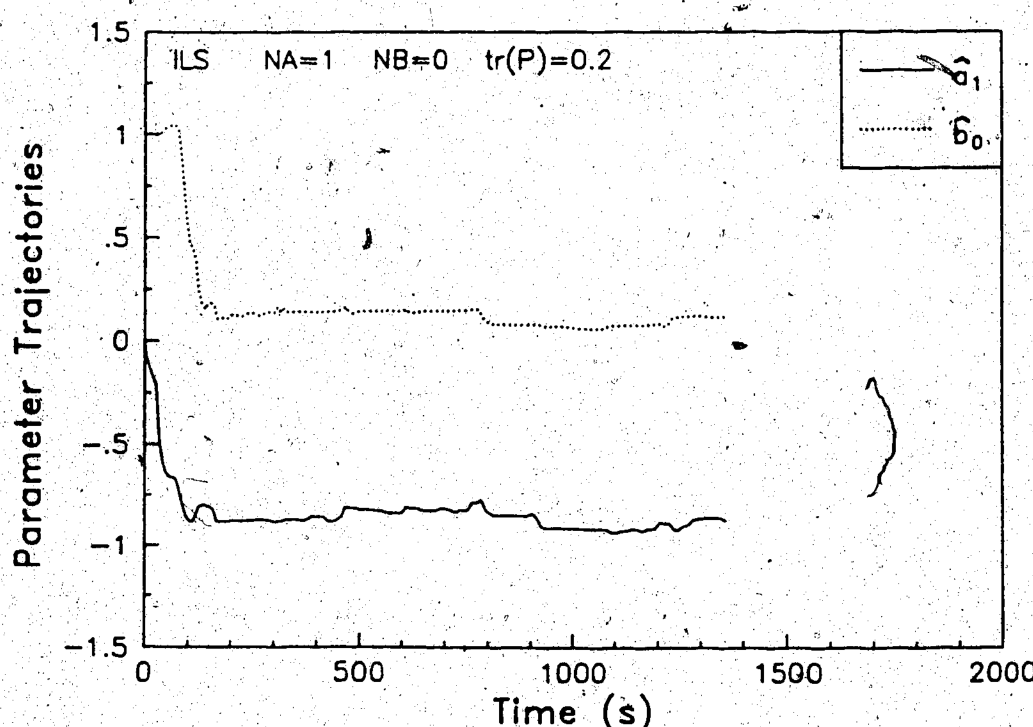


Figure 7.10b Parameter Estimation with $C_e=(1-.8q^{-1})^2$

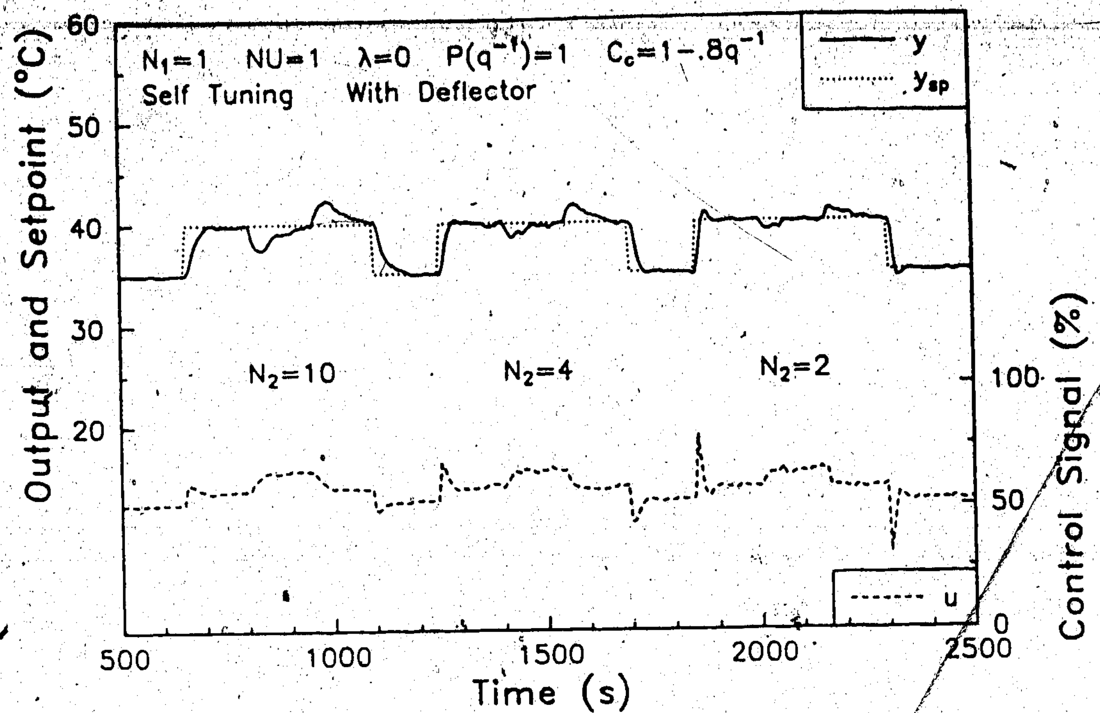


Figure 7.11 Output Horizon Control of the Stirred Tank Heater

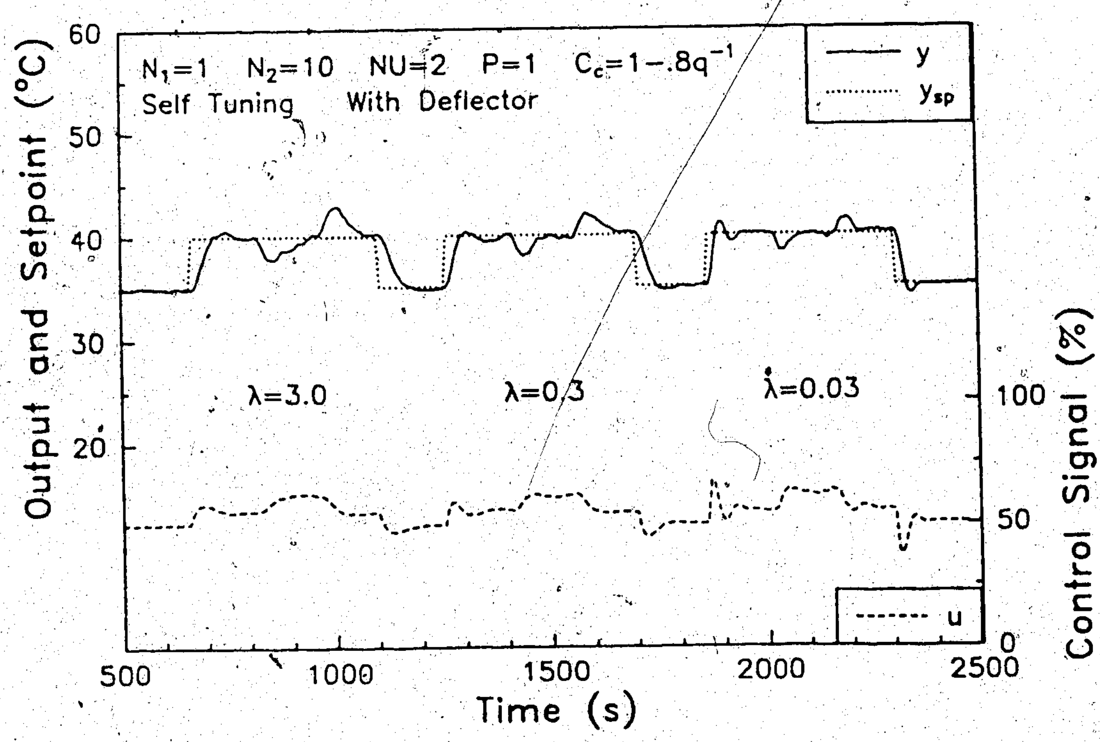


Figure 7.12 Lambda Weighting Control of the Stirred Tank Heater

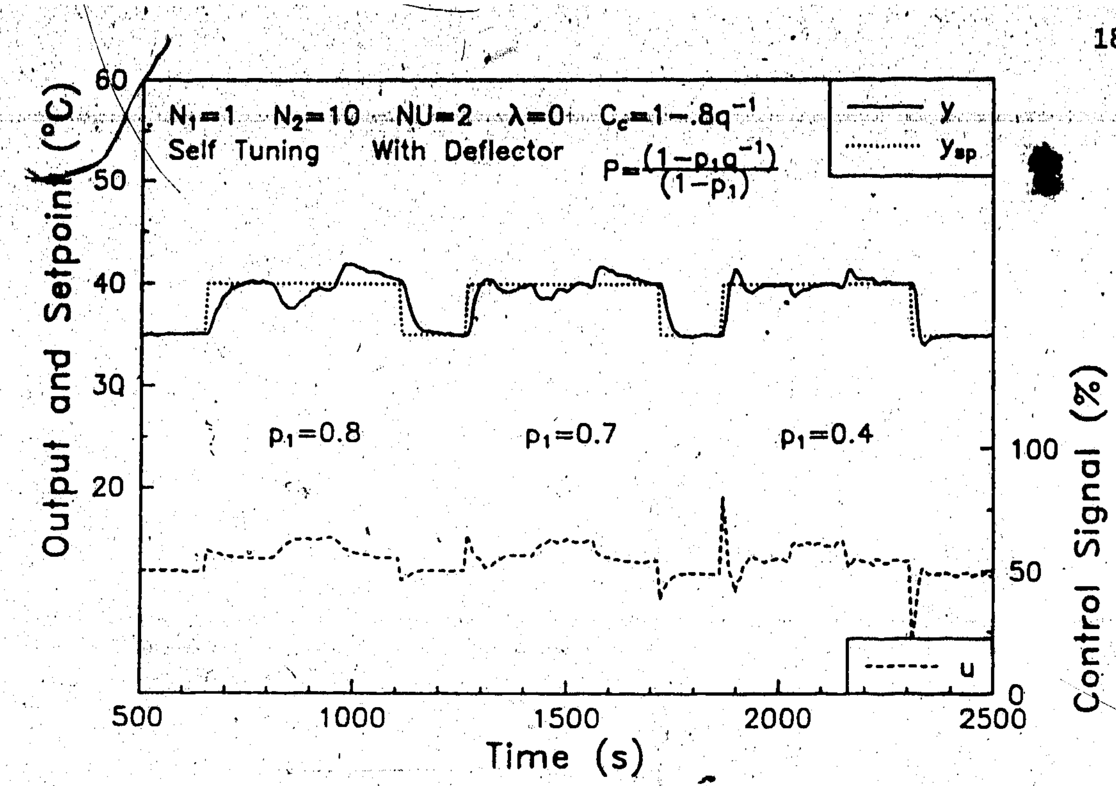


Figure 7.13 Detuned Model–Following Control of the Stirred Tank Heater

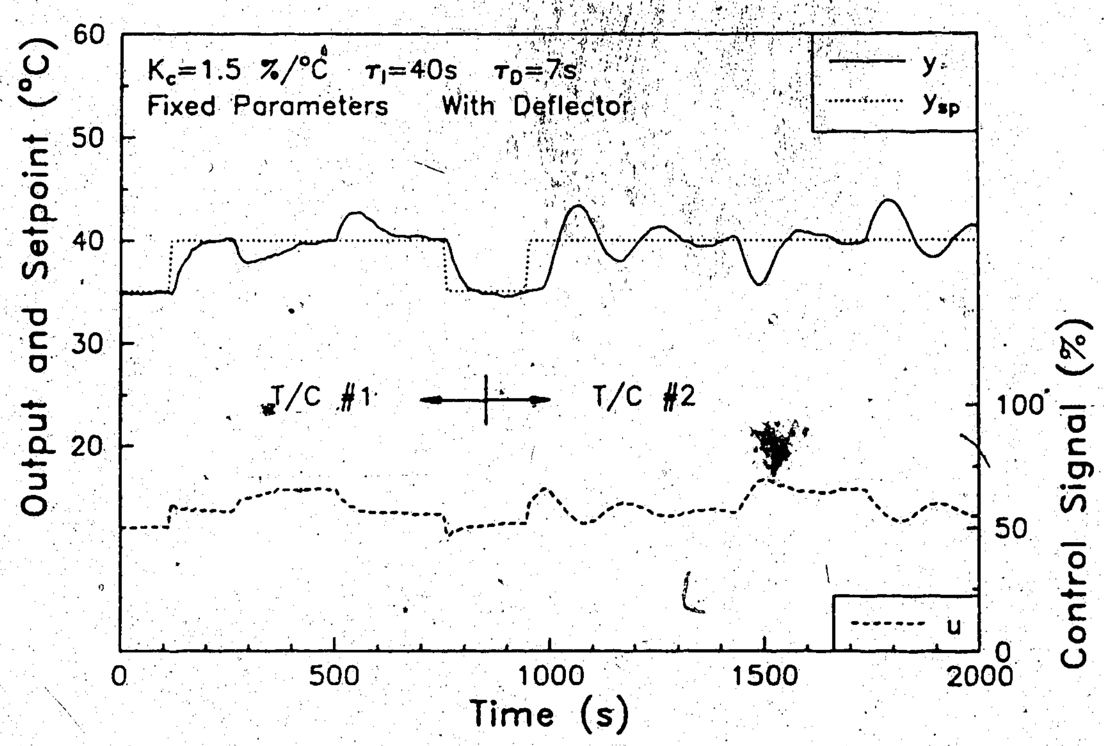


Figure 7.14 PID Control of the Stirred Tank Heater with a Variable Time Delay

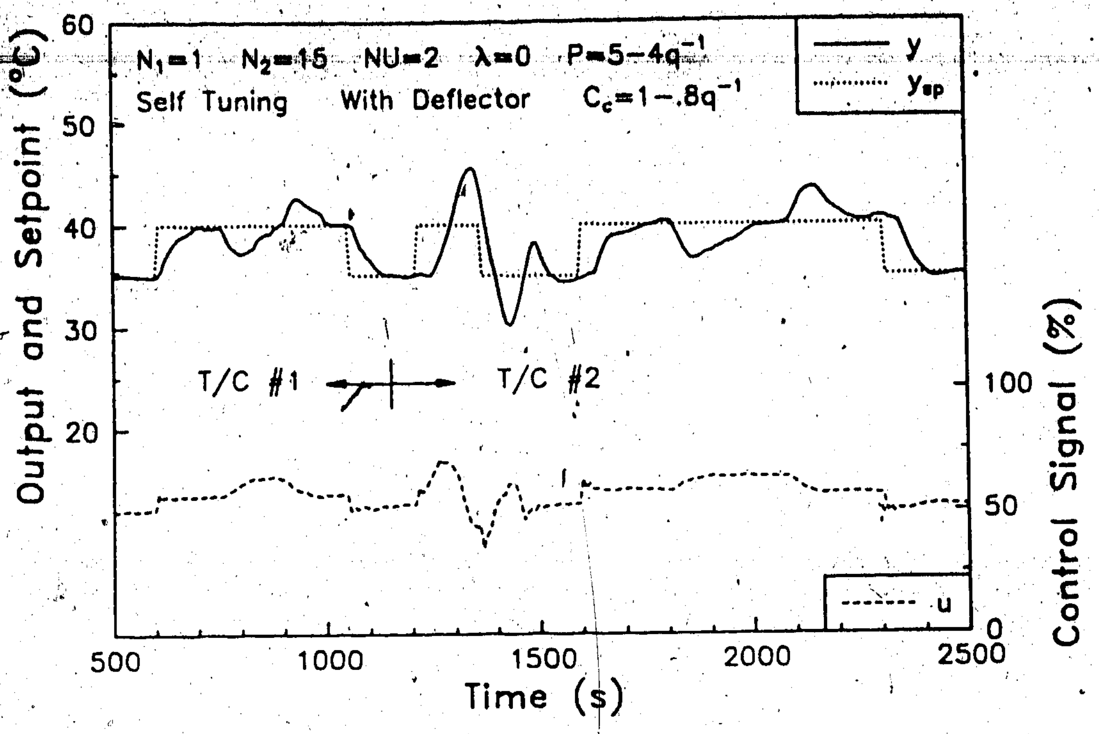


Figure 7.15a Generalized Predictive Control of the Stirred Tank Heater with a Variable Time Delay

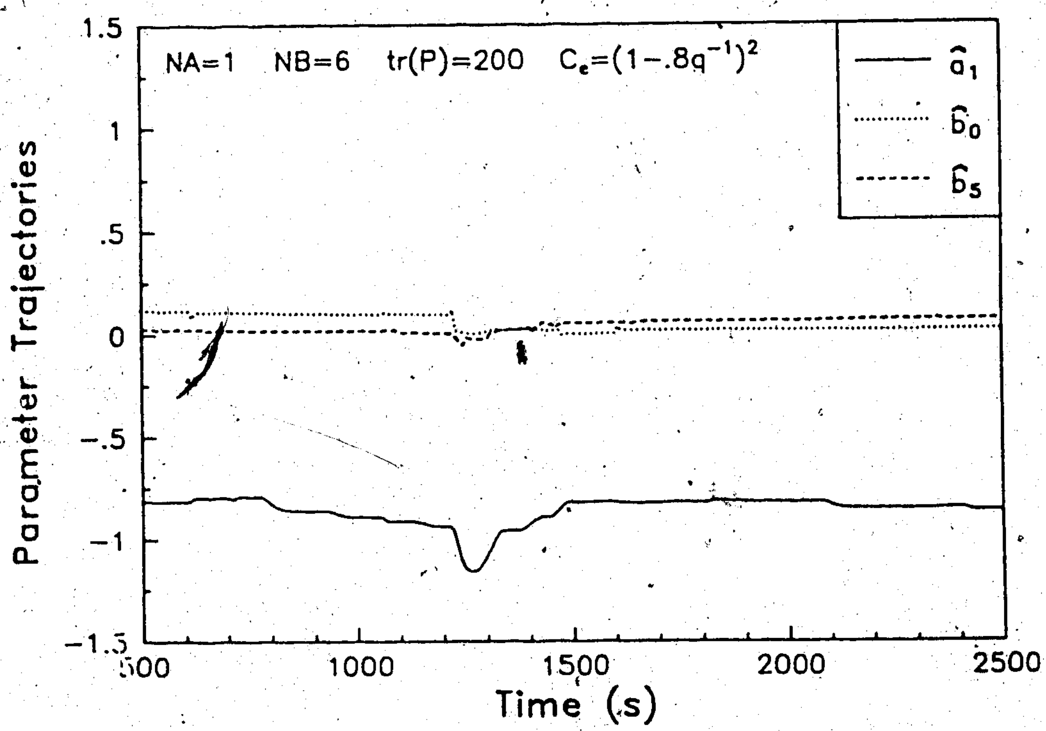


Figure 7.15b Selected Parameter Estimates

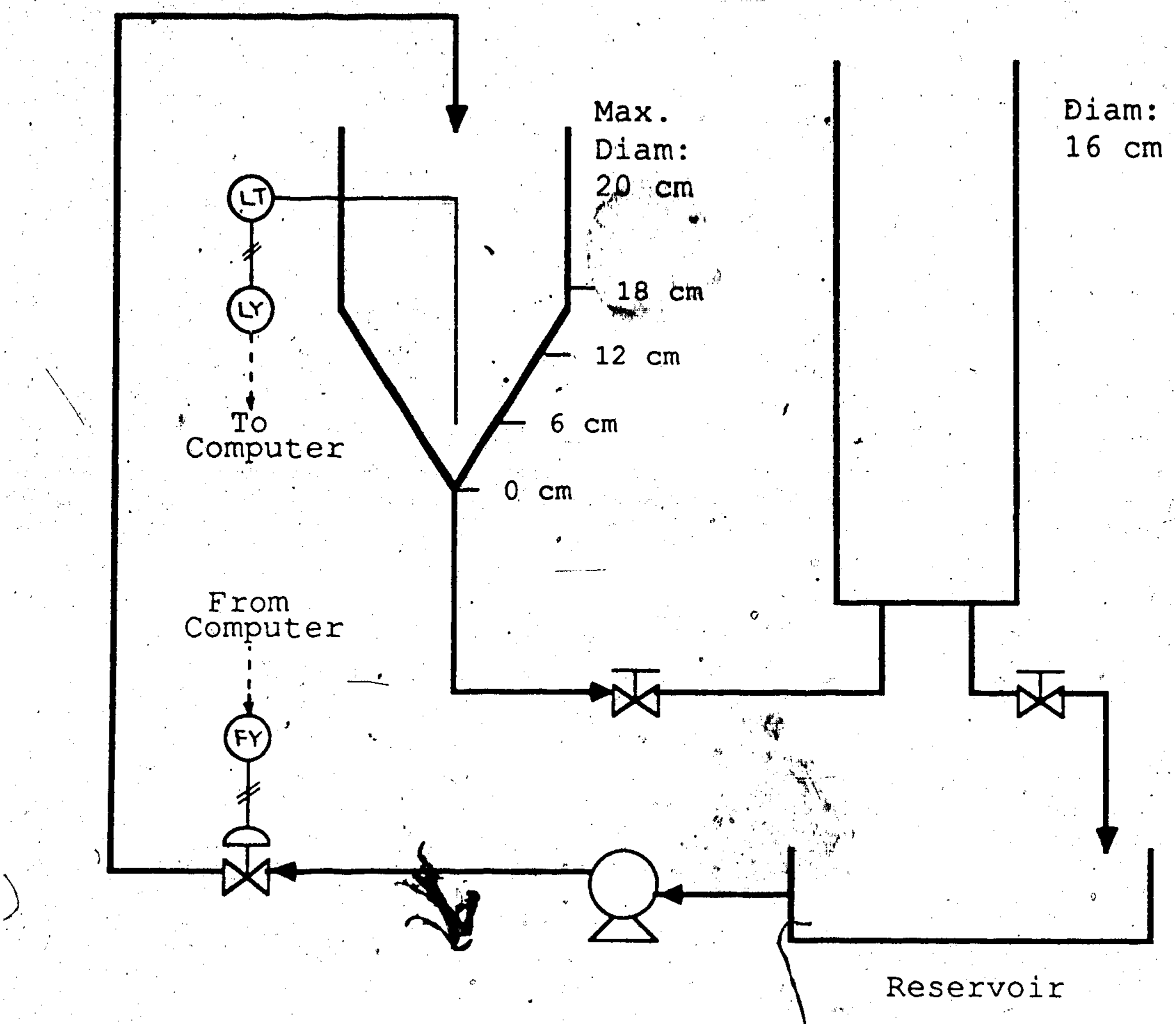


Figure 7.16 Schematic Diagram of Conical and Cylindrical Interacting Tanks

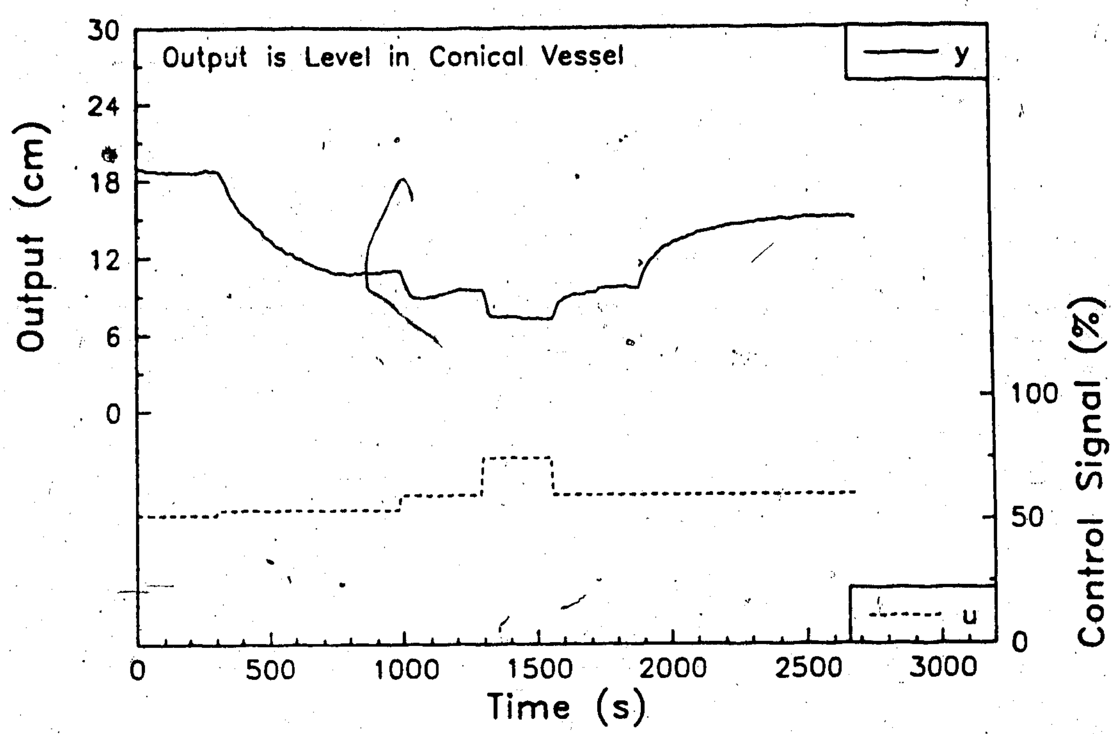


Figure 7.17 Open Loop Response of the Interacting Tanks

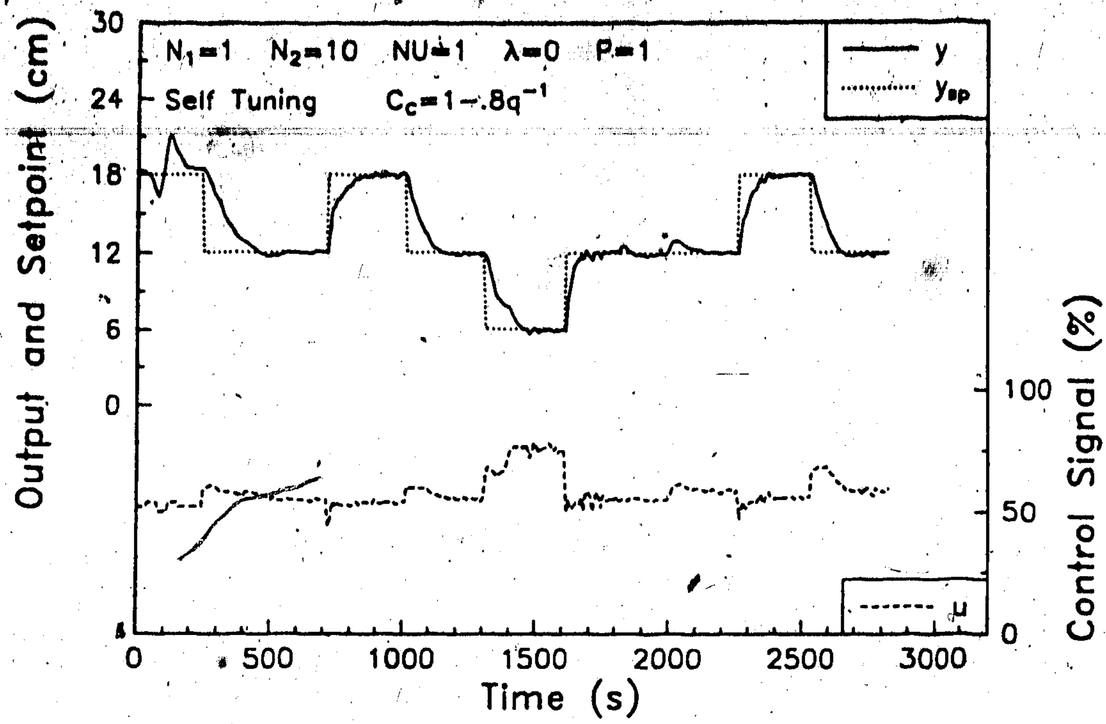


Figure 7.18a Output Horizon Control of the Interacting Tanks

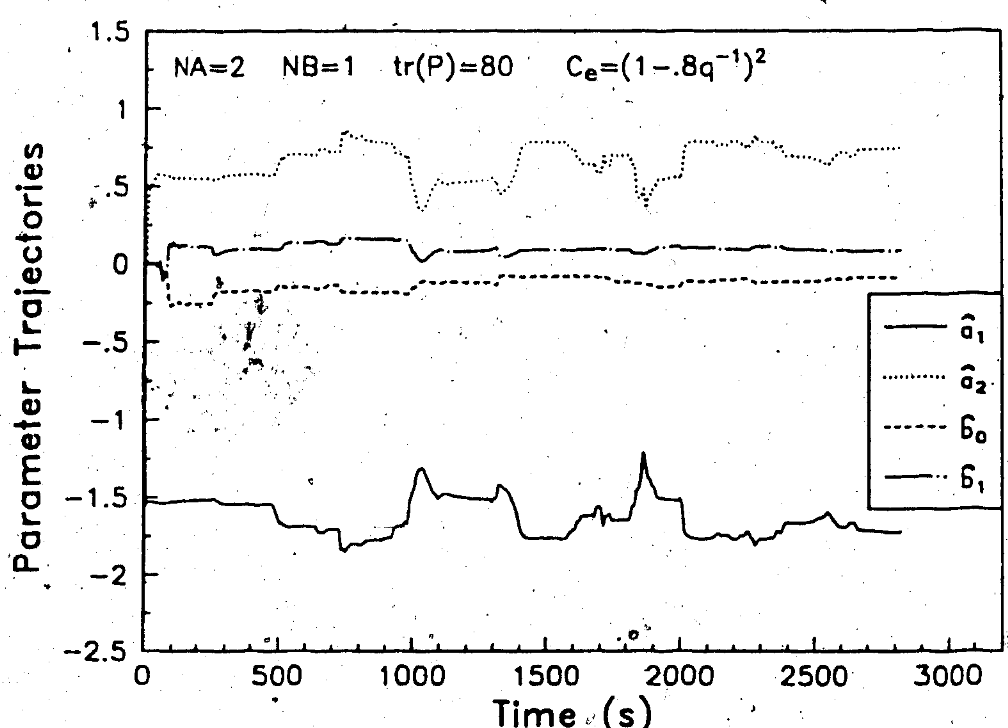


Figure 7.18b Corresponding Parameter Estimates

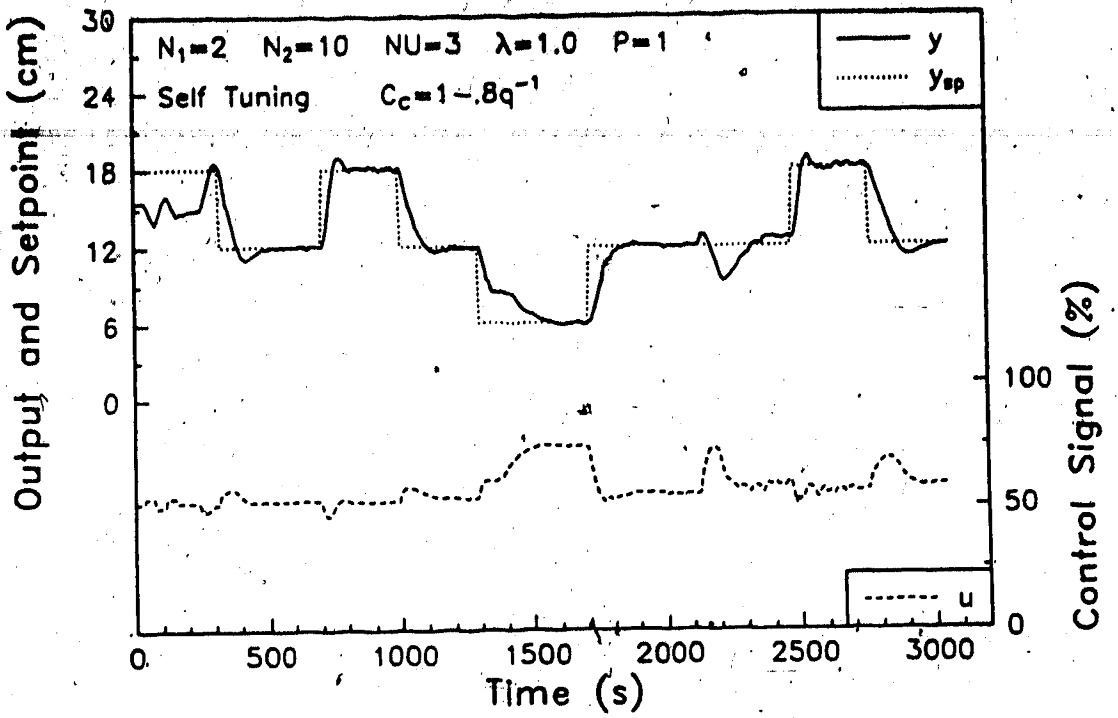


Figure 7.19a Lambda Weighting Control of the Interacting Tanks (no scaling)

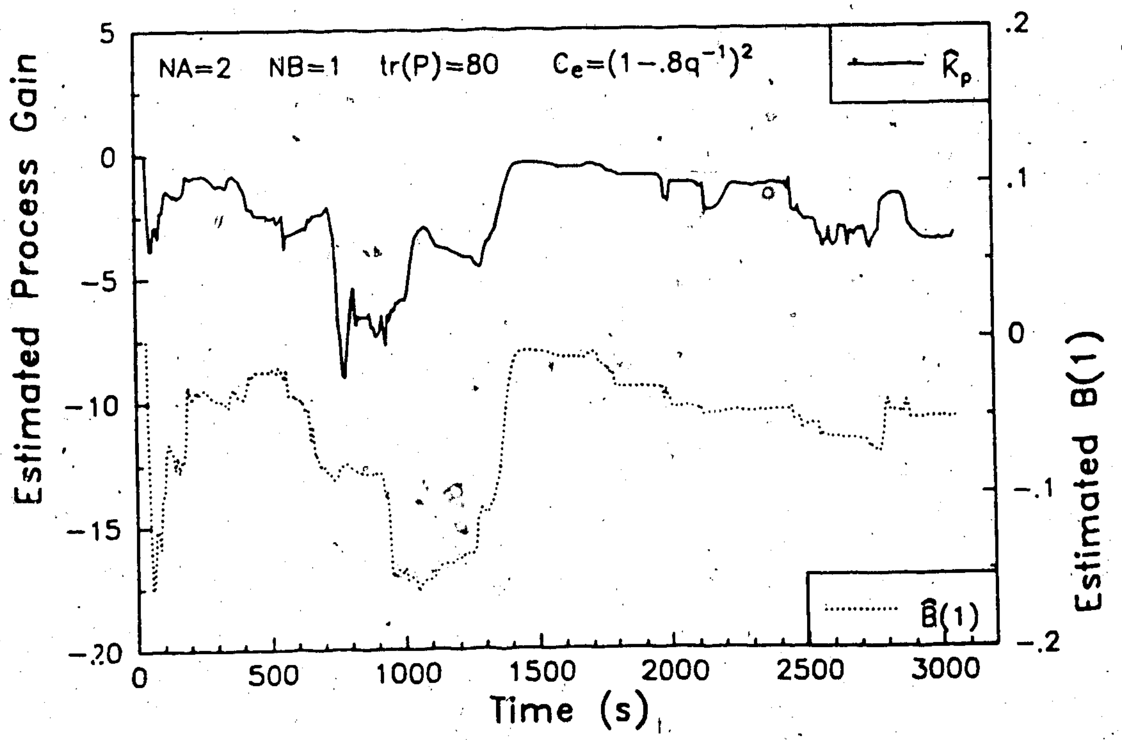


Figure 7.19b Characteristics of Identified Process Model

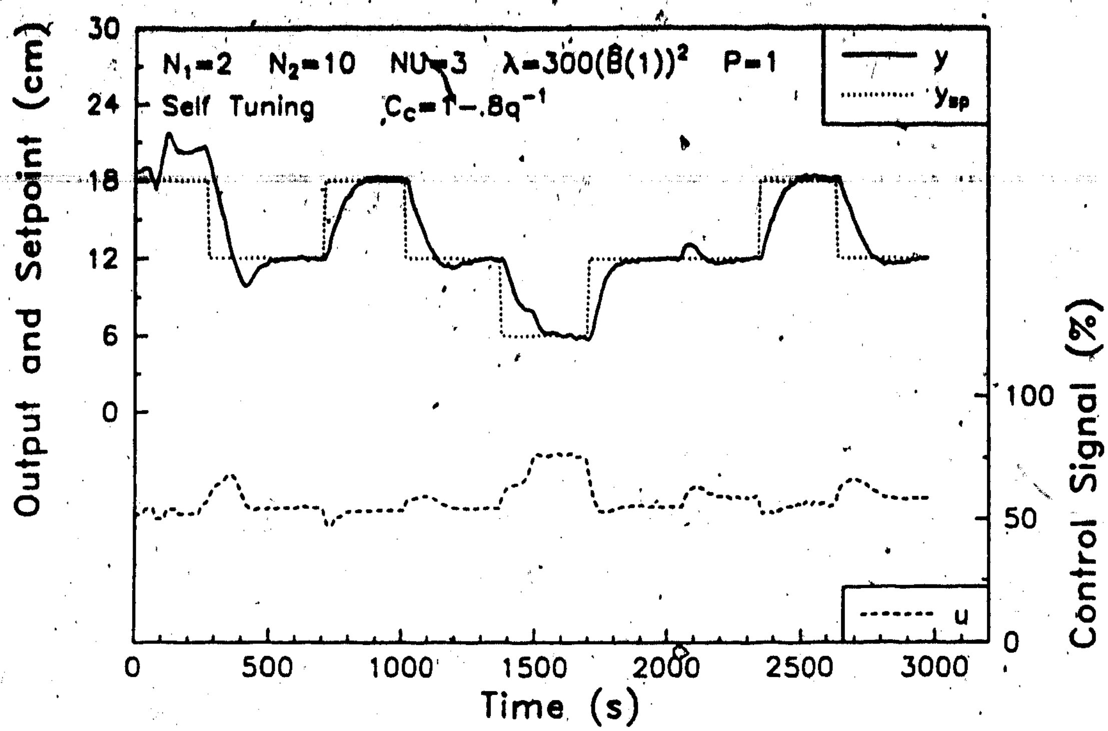


Figure 7.20a Lambda Weighting Control of the Interacting Tanks (with scaling)

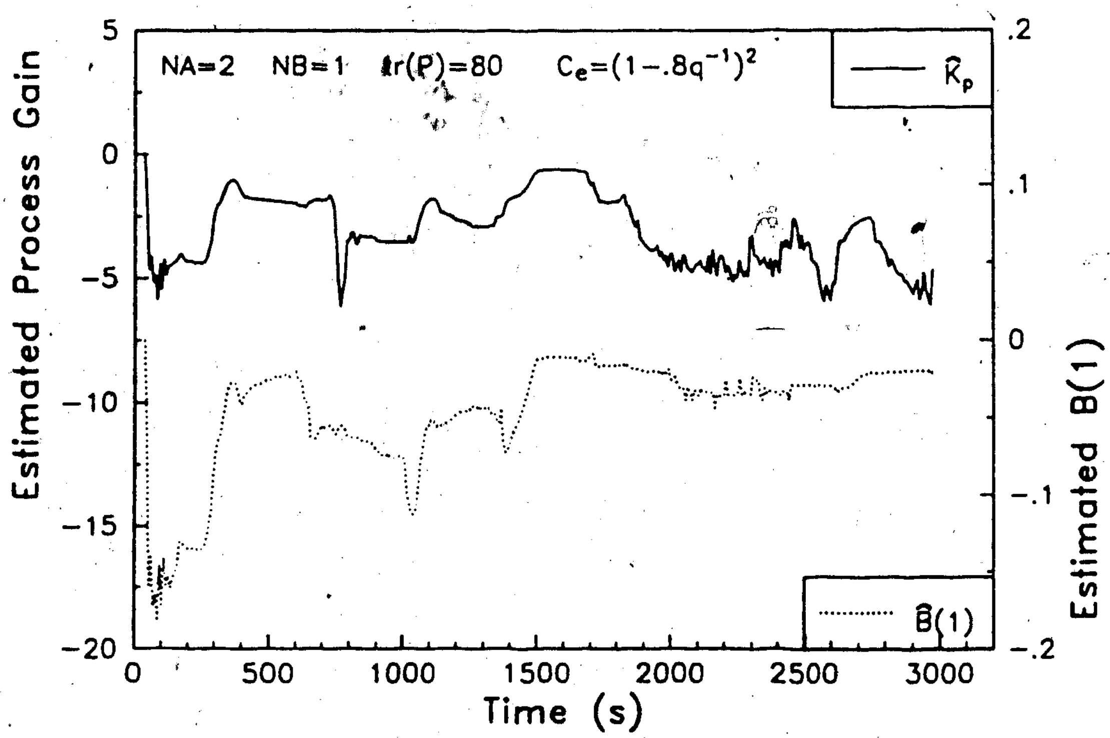


Figure 7.20b Characteristics of Identified Process Model

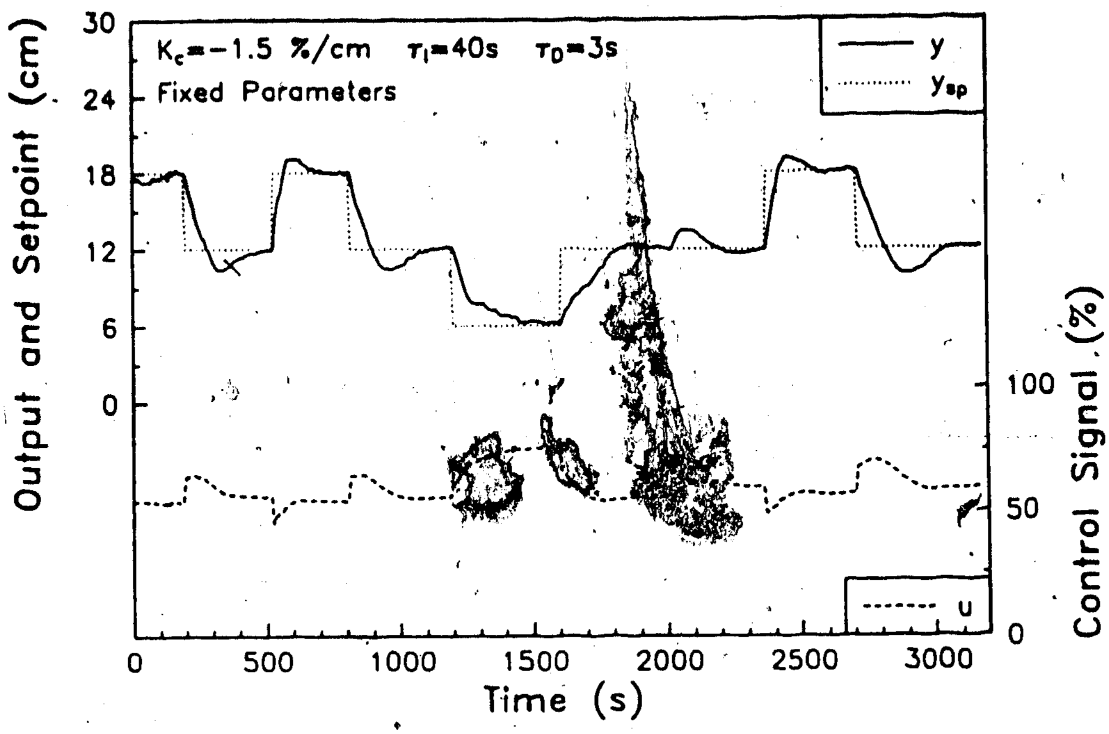


Figure 7.21 PID Control of the Interacting Tanks

8. EXPERIMENTAL APPLICATION OF SELF-TUNING POLE PLACEMENT CONTROL

Consistent with the parallel development of the GPC and PP controllers, the experimental runs in this chapter are directly comparable to GPC trials in the previous chapter. The behavior of Pole Placement was in most respects found to be similar to that of GPC, particularly [illegible] GPC tuning parameters were selected for Detuned Model-Following. It is therefore possible to reach similar conclusions regarding the influence of the controller noise model polynomial, $C_c(q^{-1})$, and the desired characteristic polynomial, $P(q^{-1})$. To avoid repetition, only a subset of the runs actually carried out with PP control are included in the following sections on the Stirred Tank Heater and Interacting Tanks.

8.1 Control of the Stirred Tank Heater

The pilot scale Stirred Tank Heater was described in section 7.1; recall that the open-loop response of the system depends strongly on whether an inlet water deflector is used or not. The implementation details discussed in section 7.2 apply equally well to the Pole Placement algorithm. The Pole Placement experimental runs with the Stirred Tank Heater are summarized in Table 8.1.

Table 8.1 PP Experimental Runs with the Stirred Tank Heater (Self-tuning)

Fig. No.	With/Without Deflector	Time Delay	$C_e(q^{-1})$	$C_c(q^{-1})$	Purpose	GPC Comp. Fig. No.
8.1	with	constant	1	1	eff. of C_e,C_c	7.3
8.2	"	"	$(1-.8q^{-1})^2$	$1-.8q^{-1}$	"	7.5
8.3	without	constant	1	1	eff. of C_e,C_c	7.6
8.4	"	"	$(1-.8q^{-1})^2$	$1-.8q^{-1}$	"	7.9
8.5	with	constant	$(1-.8q^{-1})^2$	$1-.8q^{-1}$	eff. of P	7.11-13
8.6	with	variable	$(1-.8q^{-1})^2$	$1-.8q^{-1}$	deadtime comp.	7.15

8.1.1 Stabilization using Polynomials C_e and C_c

The sequence of step setpoint changes and disturbances for the analogous GPC runs were replicated here for PP control; at time 750 s the inlet water flowrate was increased from 53 cm^3/s to 66 cm^3/s, returning to the former flowrate at 900 s. The constant trace for the RLS identification algorithm was specified to be 0.2. A first order desired characteristic polynomial with pole at 0.8 was used throughout.

8.1.1.1 With Inlet Water Deflector

In the absence of a large amount of measurement noise the closed-loop is stable for C_e and C_c equal to one as demonstrated in Figure 8.1. At times, however, the identified model gives poor predictions of the ouput (the model is unstable during the period from 600 to 1100 s). With C_c=1, the controller reacts strongly to what is perceived to be a series of disturbances with the net result that sustained oscillations are present in the response of the input and output.

If bandpass filtering of the regressor is introduced, a better low frequency model is identified (In Figure 8.2, the model is stable with a "reasonable" steady state gain). At the same time if the disturbance model polynomial used in the controller is selected as $C_c=1-.8q^{-1}$, the regulatory action of the controller is detuned and the output response is smooth. The inlet water flowrate disturbance does cause some parameter bias resulting in overshoot for the following setpoint change.

8.1.1.2 Without Inlet Water Deflector

Removing the inlet water deflector from the Stirred Tank Heater reduces the signal/noise ratio considerably. It is also expected that unmodelled dynamics will be present as the process no longer appears first order (since it is not well-mixed). Without regressor filtering a suitable low frequency model cannot be obtained (Figure 8.3) and it is impossible to stabilize the closed-loop.

On the other hand, very good results are achieved using $C_e=(1-.8q^{-1})^2$ and $C_c=1-.8q^{-1}$ as illustrated in Figure 8.4. For the majority of the run the estimated parameters hovered around $\hat{a}_1$=-0.85 and $\hat{b}_0$=0.14 (with corresponding gain estimate, $\hat{K}_p$=0.93 °C/%). When $\hat{a}_1$ approached -1.0 at 1000 s the control performance was only slightly degraded. These settings for $C_e(q^{-1})$ and $C_e(q^{-1})$ also gave the best results using GPC.

8.1.2 Altering the Closed-Loop Response

A first order desired characteristic polynomial is a suitable selection for this predominantly first order process. Figure 8.5 shows results for a trial where P was changed from $1-.8q^{-1}$ to $1-.7q^{-1}$ and finally to $1-.4q^{-1}$ before each increase in setpoint. (The corresponding desired closed-loop time constants are 35, 22 and 9 s, respectively). The standard inlet water flowrate load changes were introduced during operation with each tuning parameter setting. The trace of the covariance matrix was maintained at 0.02 and the regressor filtered using $C_e=(1-.8q^{-1})^2$. The initialization sequence is not shown in Figure 8.5. The results are very close to those obtained with GPC in its Detuned Model-Following configuration.

8.1.3 Variable Dead-time Compensation

The Pole Placement algorithm's ability to adapt to changing time delay is a consequence of overparameterizing the process model numerator. Estimating seven B parameters is sufficient for the maximum expected dead-time of 5 sampling intervals. The temperature measurement was switched from T/C #1 to T/C #2 at time 1150 s as indicated in Figure 8.6. After the retuning transient, the response to setpoint changes is delayed by 40 s but otherwise basically unchanged. As expected, the time taken to eliminate the water flowrate step disturbances has increased with the additional dead-time present.

8.2 Control of the Nonlinear Interacting Tanks

The conical and cylindrical Interacting Tanks described in section 7.4 allow evaluation of Pole Placement control on a second order highly nonlinear process. The primary objective is to confirm that the Pole Placement algorithm will maintain a consistent output response given reasonable model parameter estimates. The experimental runs with the

Interacting Tanks are summarized in Table 8.2.

Table 8.2 PP Experimental Runs with the Interacting Tanks (Self-tuning)

Figure No.	Model Orders na	Model Orders nb	Purpose	GPC Comparison Figure No.
8.7	2	1	maintenance of performance	7.18-20
8.8	2	0	"	"

8.2.1 Full Second Order Model

Figure 8.7 displays the closed-loop behavior and parameter trajectories for self-tuning PP control of the level in the conical vessel. A full second order model was identified with all other controller and estimator parameters set the same as for GPC in the previous chapter. The lack of stability is thought to be due to near common factors in the estimated model polynomials $\hat{B}(q^{-1})$ and $\hat{A}(q^{-1})$. At one sampling instant (toward the end of the run) the identified process model was:

$$\frac{q^{-1}\hat{B}(q^{-1})}{\hat{A}(q^{-1})} = \frac{-.0734q^{-1}+.0519q^{-2}}{1-1.6678q^{-1}+.6796q^{-2}} = \frac{-.0734q^{-1}(1-.7071q^{-1})}{(1-.9596q^{-1})(1-.7082q^{-1})}$$

The OL zero (.7071) lies close to one OL pole (.7082) and is expected to have produced an ill-conditioned Sylvester Matrix.

One possible way to rectify the problem is to identify a lower order model.

8.2.2 Reduced Second Order Model

The same experimental run was repeated with 1 B parameter and 2 A parameters estimated (since $\hat{b}_1$ for the previous run was observed to hover near zero). The results are shown in Figure 8.8. With the reduced 2nd order model the speed of response is almost invariant of the process changes. Increasing the resistance between the tanks at 2100 s causes an abrupt change in the parameter estimates. It is perhaps for this reason that the rejection of this disturbance is very sluggish. Note that the small "blip" in the output at time 2650 s was due to a short power surge which momentarily increased the inlet water pump speed.

The OL step response of this process appears predominantly 1st order (Figure 7.17). Based on this observation, a 1st order model would be more reasonable for this application. Results obtained with 2 B and 1 A parameter estimates (not included) were similar to Figure 8.8. Note that using a 1st order model allows overparameterization of B to account for variable time delays without causing difficulties with the solution of the Sylvester equation.

8.3 Summary

The Pole Placement algorithm, when applied to the Stirred Tank Heater, gave results very similar to those obtained in the previous chapter using GPC. Overparameterization of the $\hat{B}(q^{-1})$ polynomial allowed excellent compensation for variable time delays.

When control of the nonlinear Interacting Tanks was attempted, near common factors in the estimated polynomials of the full second order model destabilized the closed-loop. Although in this particular case, the problem was rectified by reducing the number of estimated parameters, the sensitivity of Pole Placement controllers to overparameterization of the $\hat{A}(q^{-1})$ polynomial (when $\hat{B}(q^{-1})$ is also overparameterized) represents a significant drawback.

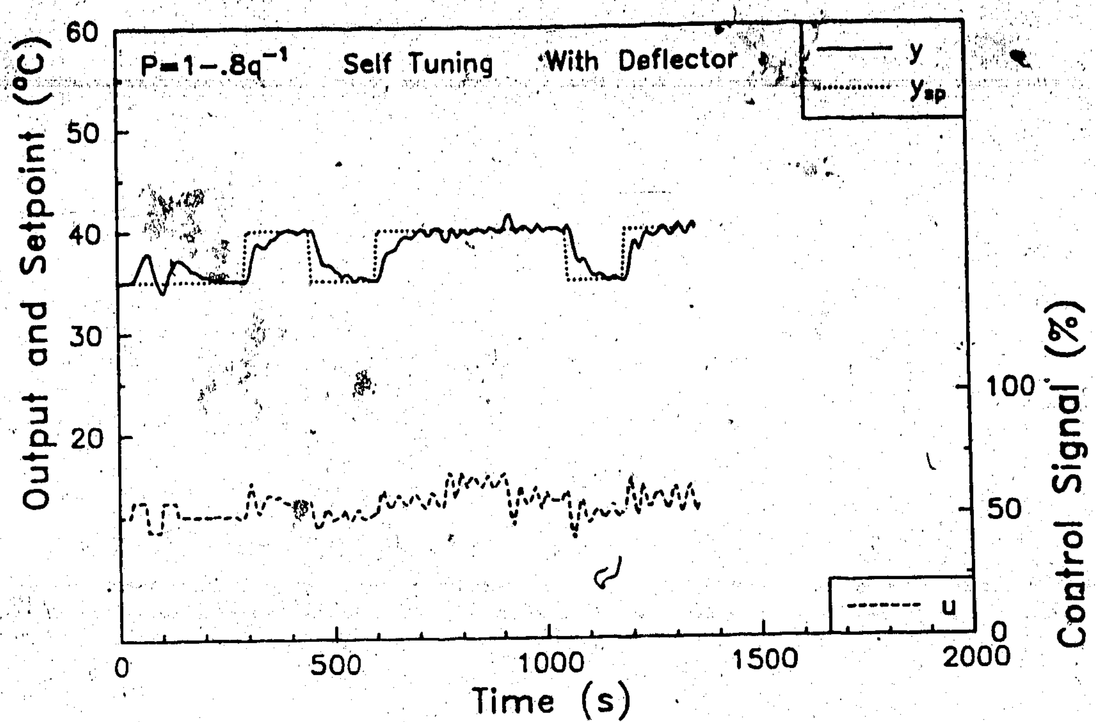


Figure 8.1a Control of the Stirred Tank Heater using Pole Placement with $C_c=1$

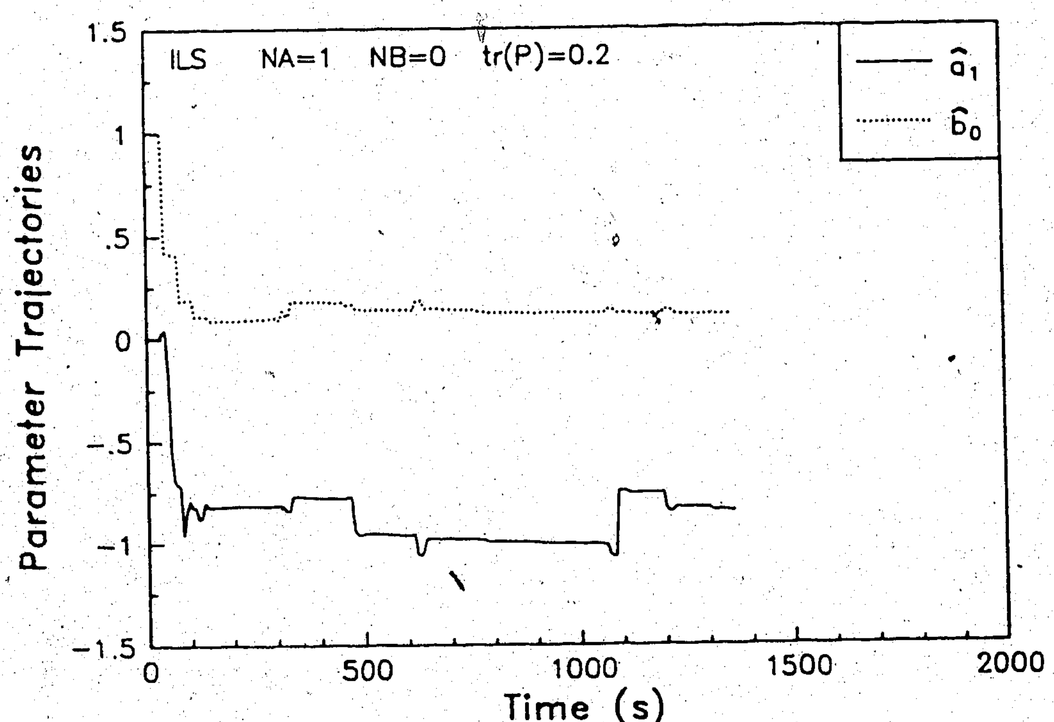


Figure 8.1b Parameter Estimation with $C_e=1$

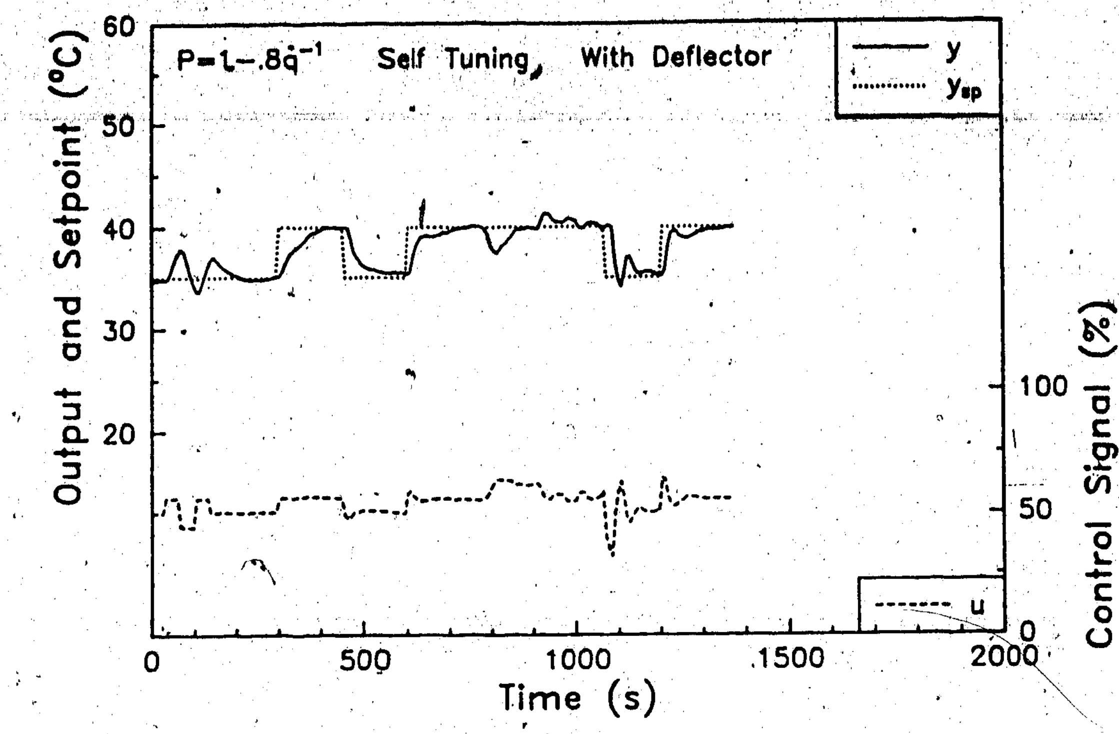


Figure 8.2a Control of the Stirred Tank Heater using Pole Placement with $C_c=1-.8q^{-1}$

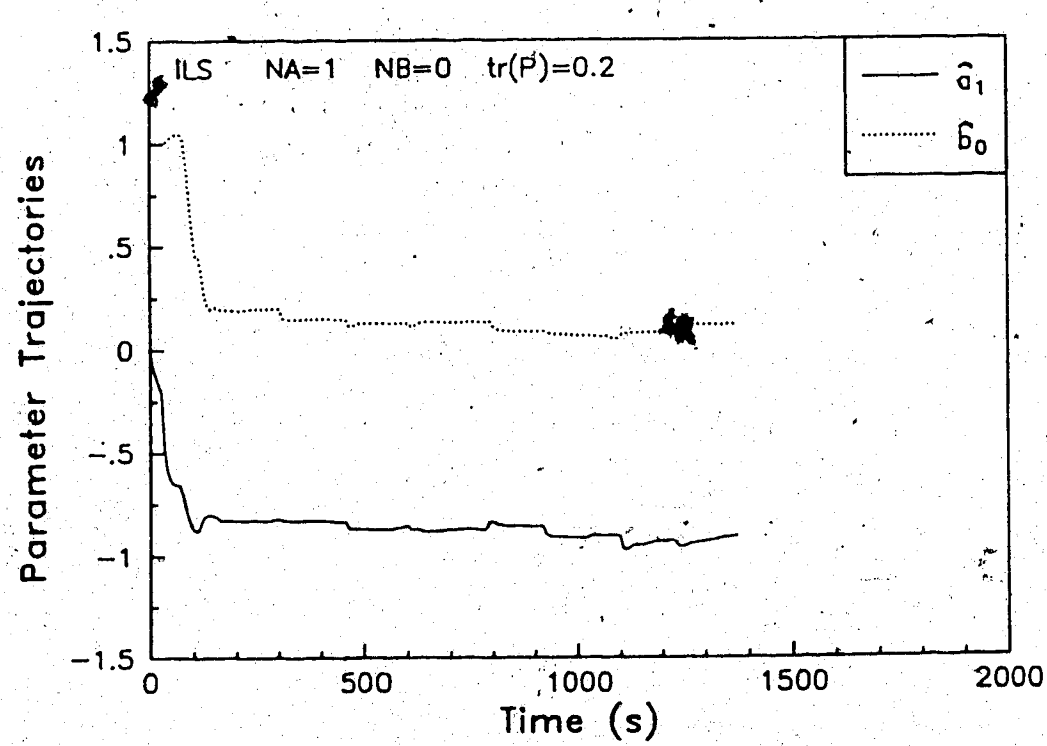


Figure 8.2b Parameter Estimation with $C_e=(1-.8q^{-1})^2$

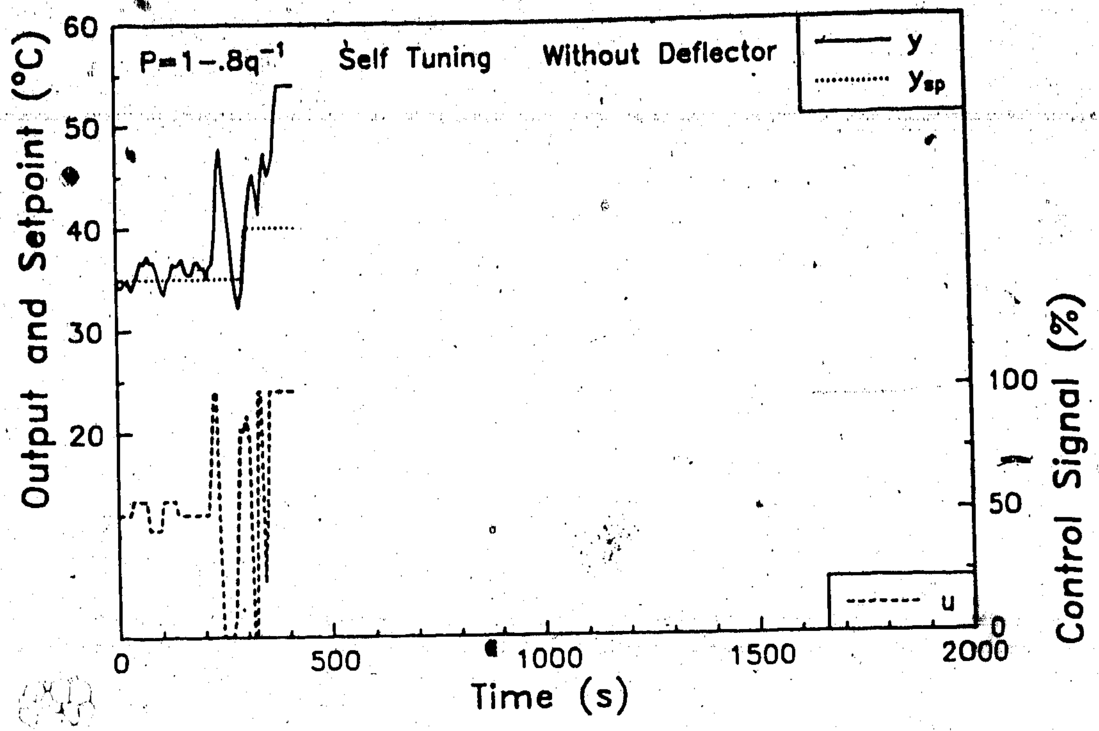


Figure 8.3a Control of the Stirred Tank Heater using Pole Placement with $C_c=1$

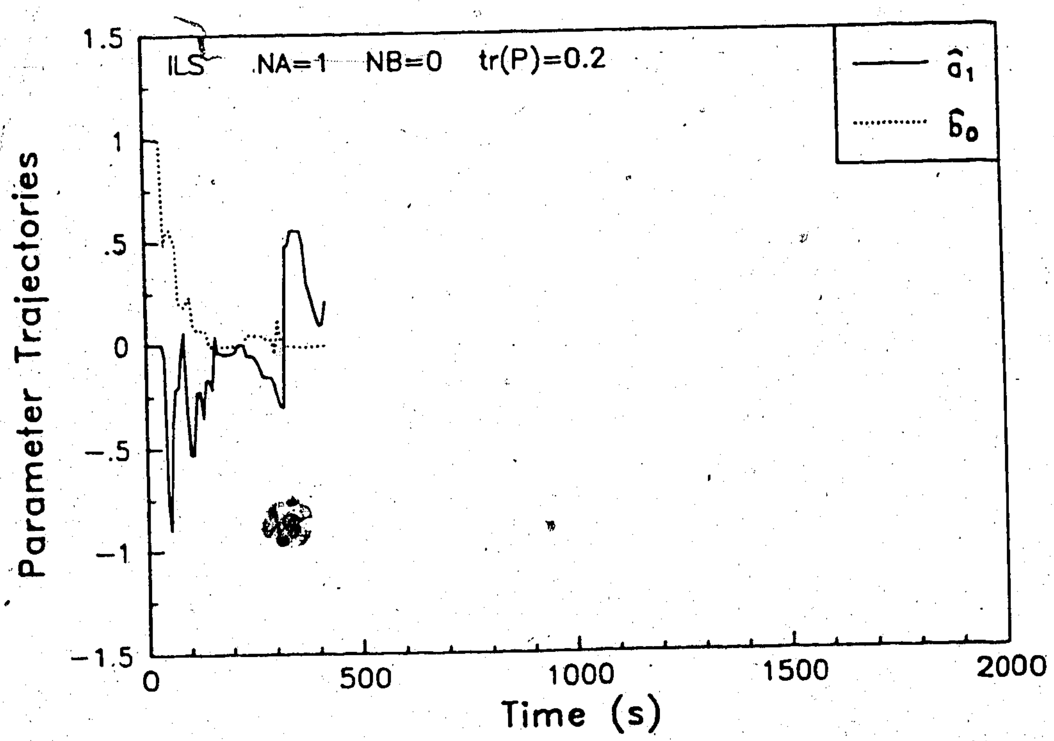


Figure 8.3b Parameter Estimation with $C_e=1$

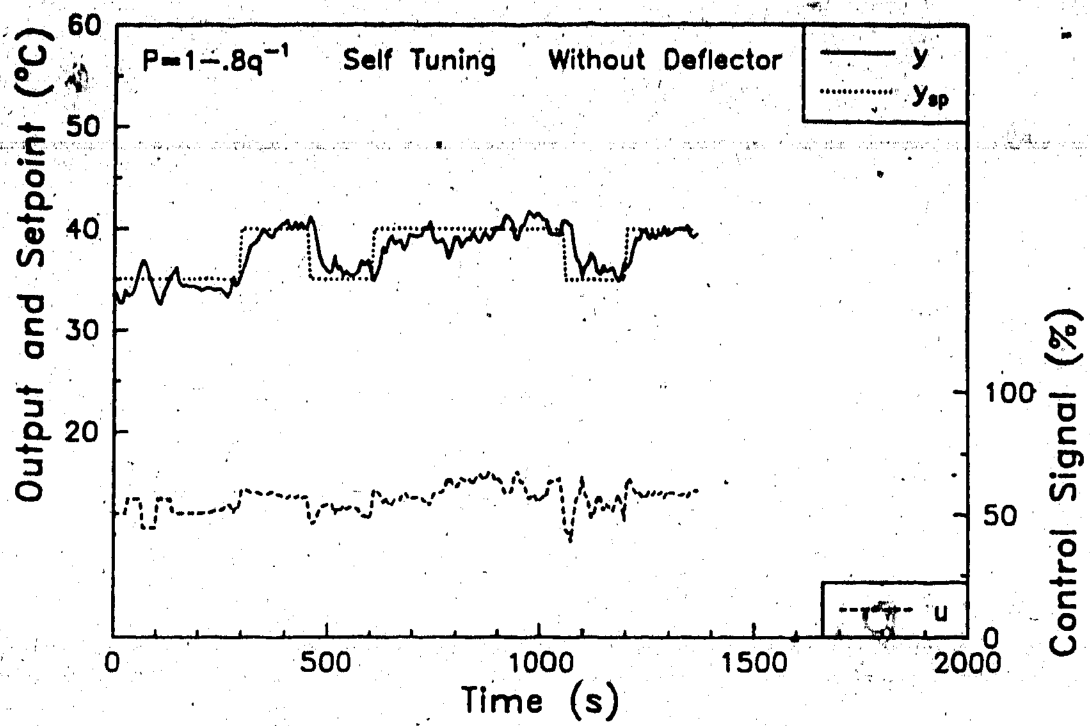


Figure 8.4a Control of the Stirred Tank Heater using Pole Placement with $C_c=1-.8q^{-1}$

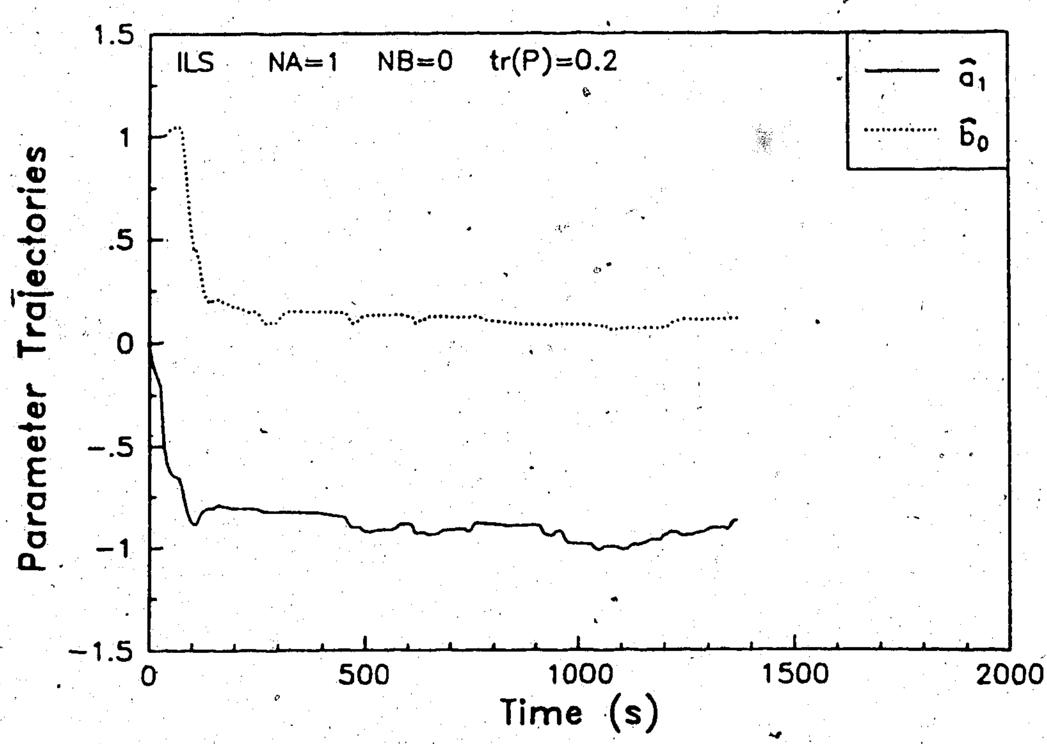


Figure 8.4b Parameter Estimation with $C_e=(1-.8q^{-1})^2$

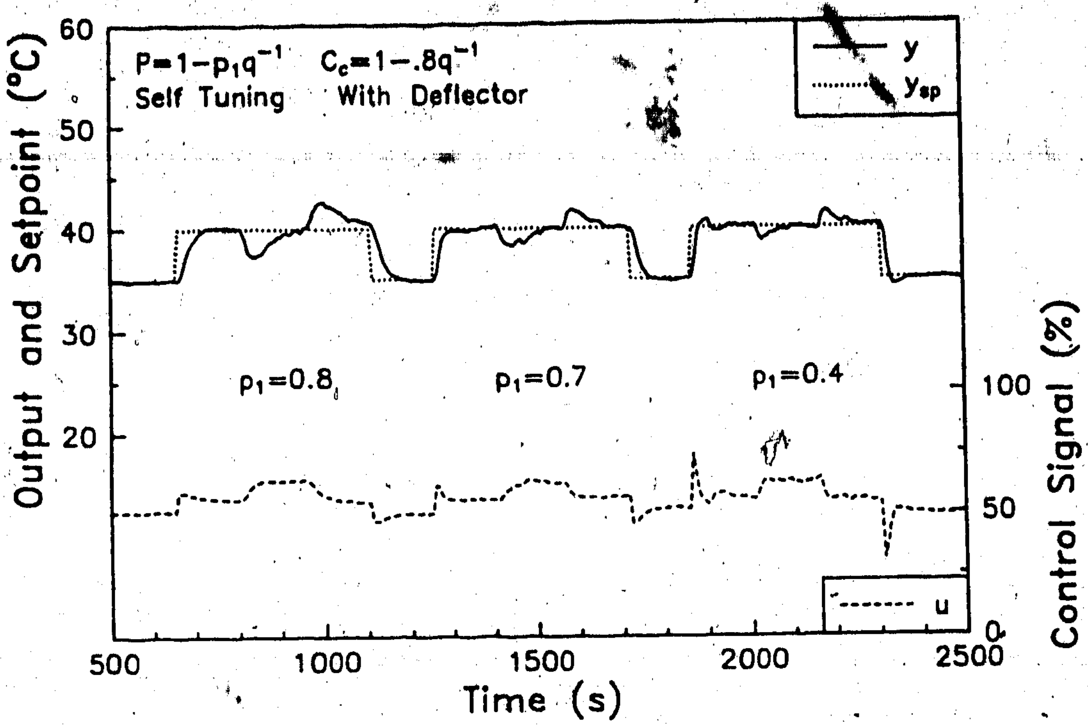


Figure 8.5a Pole Placement Control of the Stirred Tank Heater.

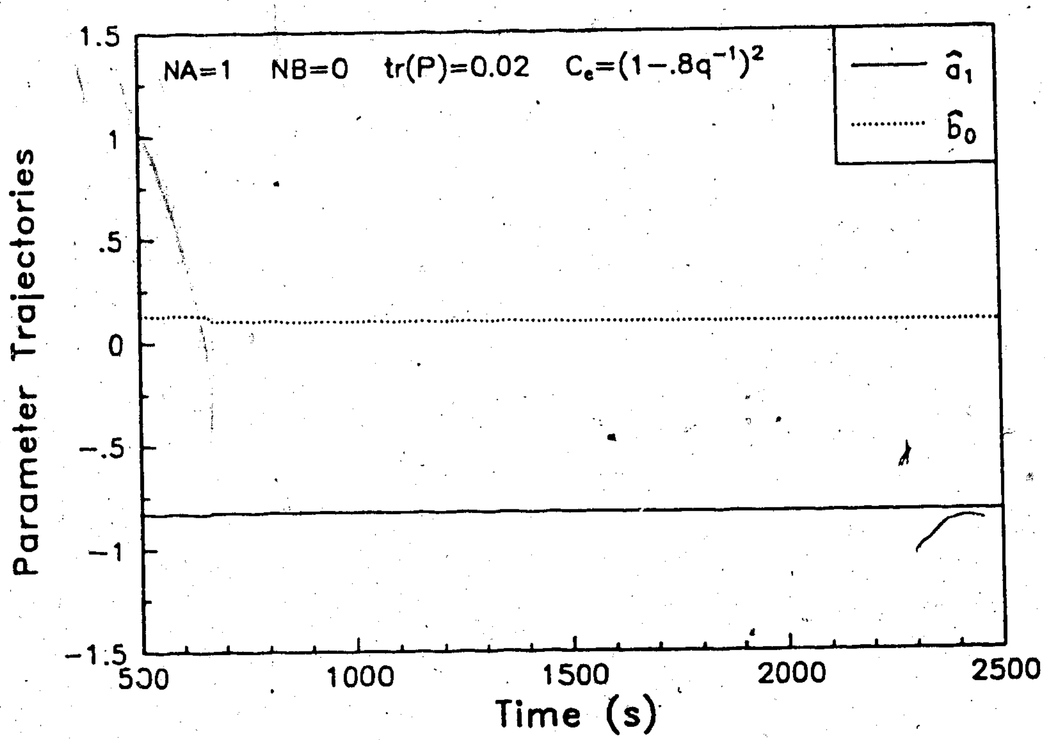


Figure 8.5b Corresponding Parameter Estimation

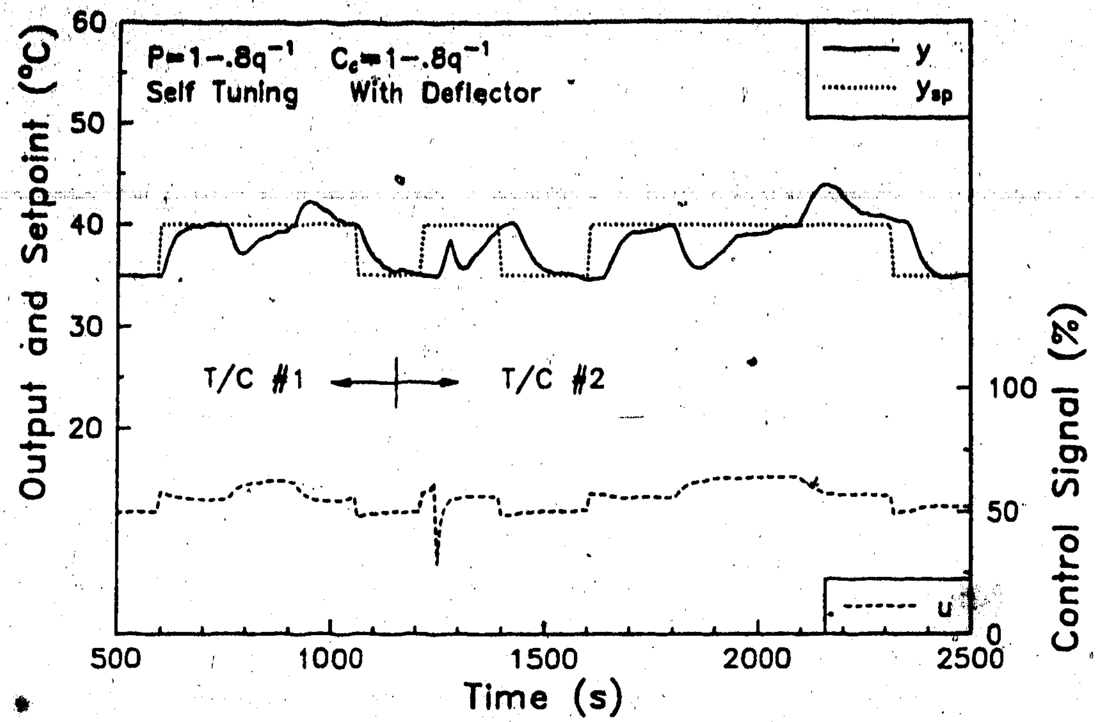


Figure 8.6a Pole Placement Control of the Stirred Tank Heater with a Variable Time Delay

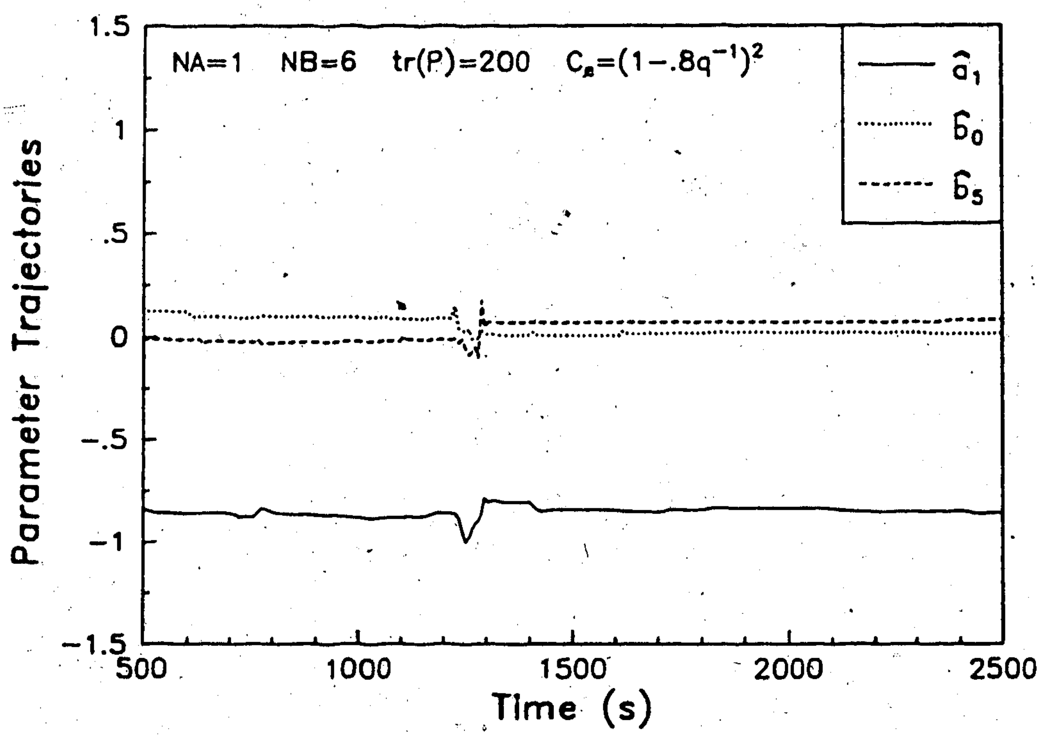


Figure 8.6b Selected Parameter Estimates

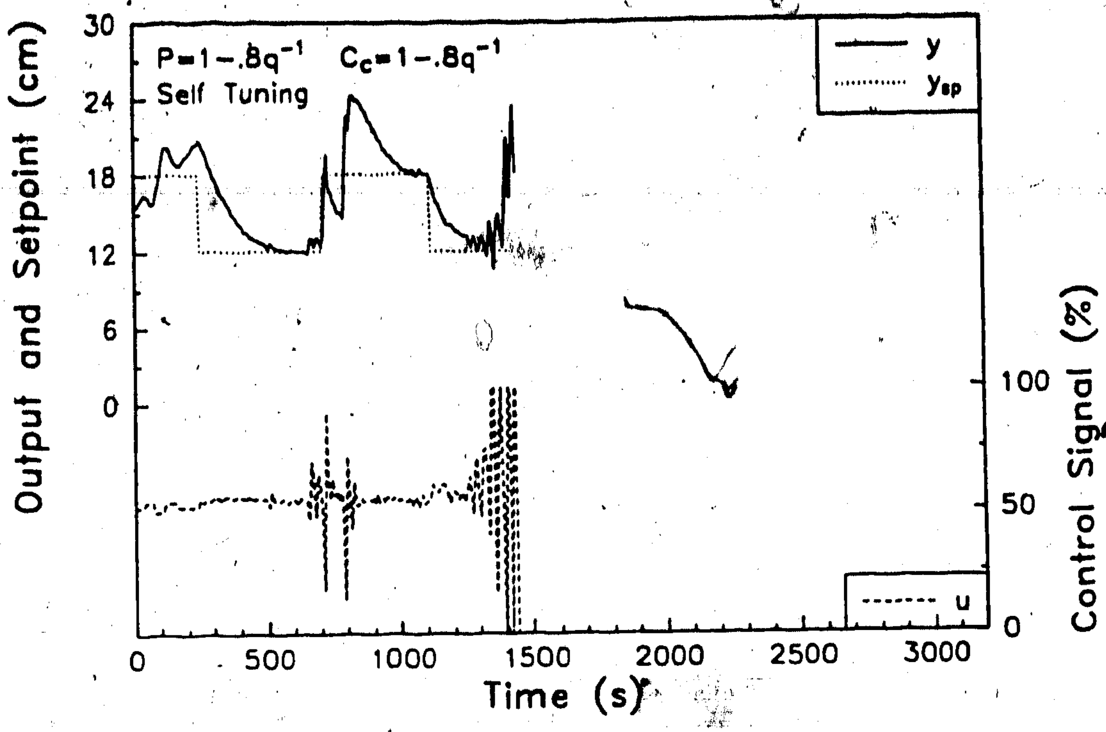


Figure 8.7a Pole Placement Control of the Interacting Tanks

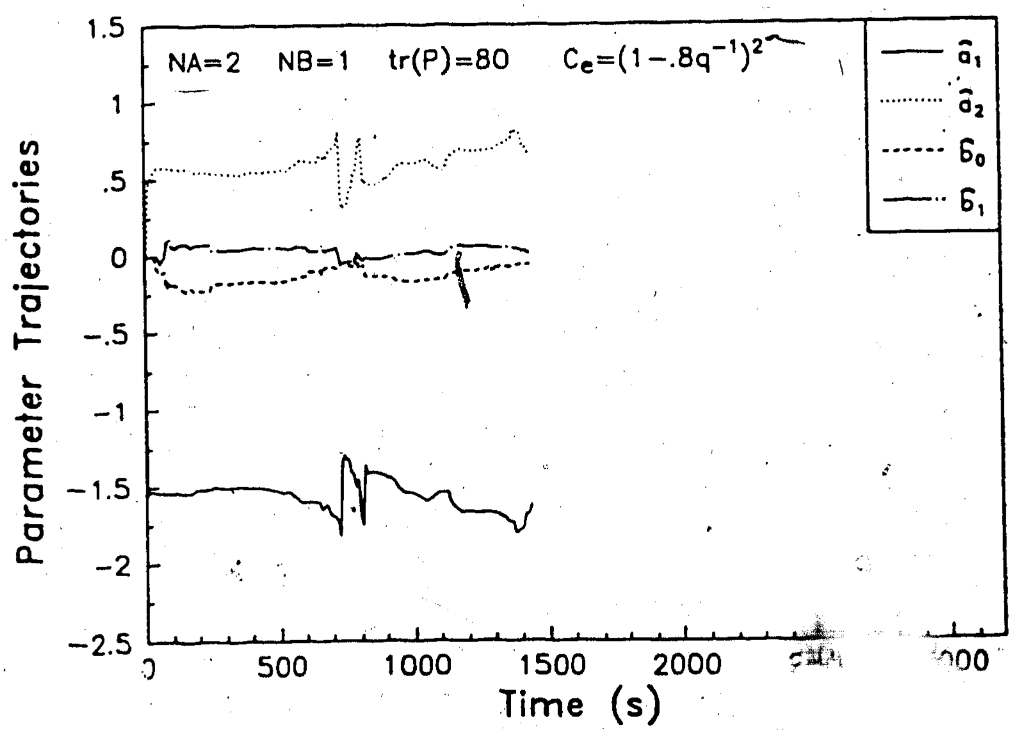


Figure 8.7b Parameter Estimates for Full [illegible]del

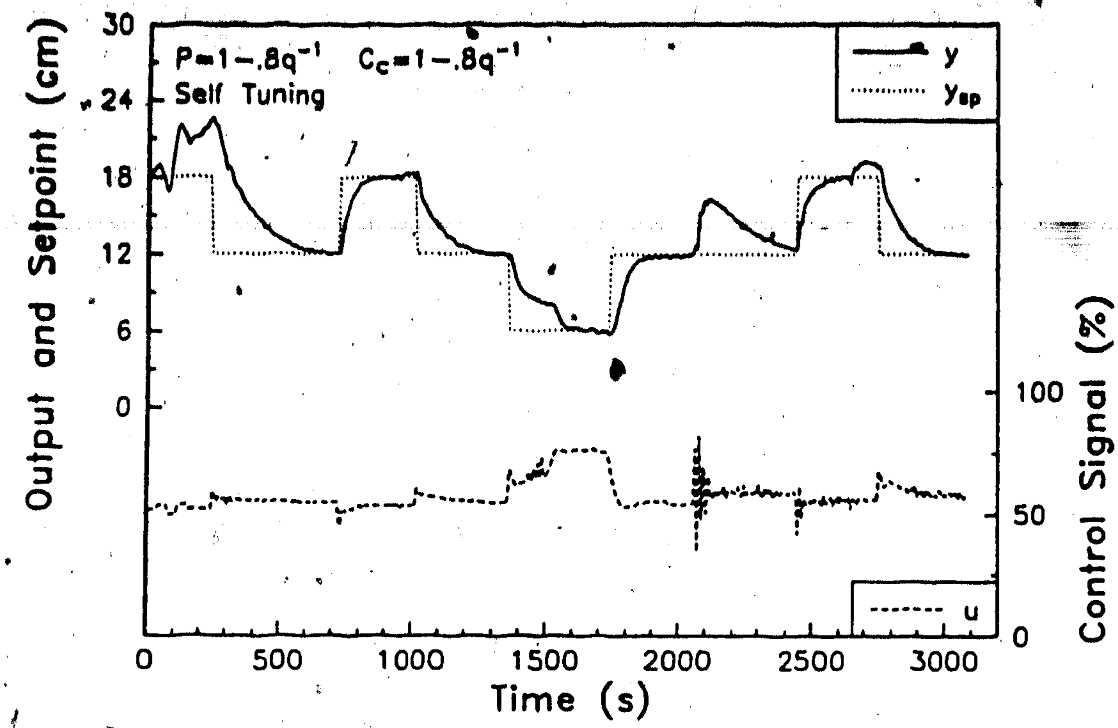


Figure 8.8a Pole Placement Control of the Interacting Tanks

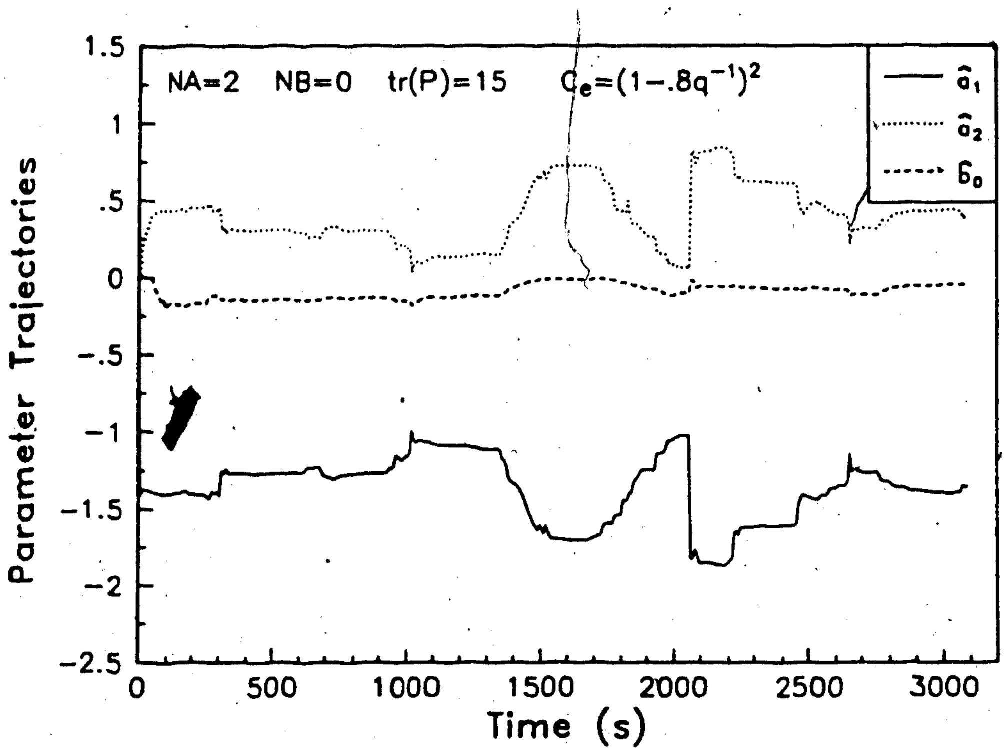


Figure 8.8b Parameter Estimates for Reduced 2nd Order Model

9. COMPARISON OF GENERALIZED PREDICTIVE AND POLE PLACEMENT CONTROL

In the previous four chapters, simulation and experimental results were used to evaluate and demonstrate the capabilities of the Generalized Predictive and Pole Placement techniques. Since the behavior of GPC depends strongly on the settings of its many design and tuning parameters, three configurations of GPC were studied. When each of these configurations is implemented, only one tuning parameter remains available for the user to adjust in order to achieve the desired response time. Note that while the potential exists to achieve better performance with GPC by adjusting all of the tuning parameters during operation, in practice this is extremely difficult to do. The possibility of accidently selecting a very poor tuning parameter set is quite high. In this chapter, a comparison is made between each of the three configurations of GPC and the PP controller, based on results presented in Chapters 5 through 8.

In section 9.1, the closed-loop transfer function for Generalized Predictive Control is compared to that for Pole Placement control and the relationship between the two control strategies is commented upon. In sections 9.2 through 9.5, the algorithms are compared based on the following criteria:

1) ease of application
2) computational effort
3) ability to maintain performance despite variations in
    a) process gain and dynamics
    b) dead-time
    c) process order
4) character of the closed-loop response

Section 9.6 provides a brief summary of the results of the comparison.

9.1 Closed-Loop Transfer Functions

In section 4.5.4, it was shown that the GPC characteristic polynomial contains the controller design polynomial, $C_c(q^{-1})$, as a factor. The closed-loop transfer function for GPC may thus be written:

$$y(t) = \frac{C_c B R' q^{-1} w(t) + T' C_0 \Delta x(t)}{C_c [T' A \Delta + q^{-1} B S']} \tag{9.1.1}$$

where C_0 represents the "true" noise model.

R', S', and T' are polynomials in the general linear form of the control law (2.3.4) with $C_c(q^{-1})=1$, given by eqn. (2.3.7). These polynomials are indirect functions of the design and tuning parameters, N_1, N_2, NU, λ, $P(q^{-1})$ as well as the model polynomials $A(q^{-1})$ and $B(q^{-1})$. Thus, the closed-loop poles depend upon all these variables.

A Pole Placement controller can be obtained as a special case of GPC by several different combinations of the design parameter settings (Mohtadi and Clarke, 1986; see also section 2.5). In this case, the closed-loop transfer function becomes:

$$y(t) = \frac{C_c B R' q^{-1} w(t) + T' C_0 \Delta x(t)}{C_c P} \tag{9.1.2}$$

and the user specifies the closed-loop poles directly through the choice of $P(q^{-1})$. Aside from a difference in notation, this is the same closed-loop transfer function that was obtained for the standard Pole Placement formulation, (see section 3.1.2):

$$y(t) = \frac{C_c B[P(1)/B(1)]q^{-1}w(t) + G C_0 \Delta x(t)}{C_c P} \tag{9.1.3}$$

(with the following equivalences: $P(1)/B(1) = R'$, $T = G$)

A few remarks are in order.

The Pole Placement controller can be thought of as a "fourth" configuration of Generalized Predictive Control. For the Pole Placement configuration, the desired characteristic polynomial, $P(q^{-1})$, serves as the active tuning parameter.

The roots of the controller design polynomial, $C_c(q^{-1})$, become closed-loop poles when using Pole Placement or Generalized Predictive Control with arbitrary design and tuning parameter settings. In either case, $C_c(q^{-1})$ does not affect servo control in the absence of MPM.

In general, the closed-loop poles for GPC (ie. roots of $[T\dot{A}\Delta + q^{-1}\dot{B}S]$) move to different locations if $A(q^{-1})$ and $B(q^{-1})$ change. Simulations indicate that these movements are often quite small for typical process variations. In contrast, when using the Pole Placement strategy, the closed-loop poles are invariant to changes in the model polynomials $A(q^{-1})$ and $B(q^{-1})$.

9.2 Ease of Application

Past criticism of the Generalized Predictive Controller often centered on the difficulty of specifying the numerous design and tuning parameters. Based on experience with the three configurations of GPC devised in this thesis, this criticism is no longer valid. With only one active tuning parameter used to vary the overall speed of response, the Generalized Predictive Controller can be as easy to apply as the Pole Placement strategy.

9.3 Computational Effort

The computational effort involved in using GPC depends primarily upon the control horizon, NU, and to a lesser extent on the maximum output horizon, N_2.

The matrix which is inverted in the control calculation:

$$(G_r^T G_r + \lambda I)^{-1}$$

is of dimension NUxNU and therefore, larger values of NU increase computing requirements. For the Output Horizon configuration, NU=1 and only a scalor division is necessary. The computational effort involved for the Lambda Weighting or Detuned Model-Following configurations is usually somewhat higher, since NU=na+1. However, a 2nd order model (possibly with B overparameterized) is of sufficiently high order for most chemical processes. Therefore, frequently NU≤3, and the matrix inversion is not burdensome.

The number of Diophantine recursions is proportional to N_2. In most cases, after tuning the closed-loop response, N_2 is somewhat smaller for the Output Horizon configuration.

standard Pole Placement formulation, the solution for the polynomials F and G requires the inversion of the Sylvester For a minimal degree solution, this square matrix is of dimension (na, np+nc-nb-1) + nb + 2. When B is overparameterized to compensate for a large (or variable) time delay, the computational effort involved in this inversion may be considerable. On the other hand, if Pole Placement is obtained as a special case of GPC, the number of operations is comparable to the Detuned Model-Following or Lambda Weighting configurations.

9.4 Maintenance of Performance

The goal of an adaptive control system is to maintain the closed-loop performance at user specifications despite variations in the gain, dynamics, dead-time and/or order of the process. The control algorithm itself should be capable of maintaining performance for these variations, given a reasonable model of the process at all times.

9.4.1 Variations in Process Parameters

Placing the closed-loop poles at prespecified locations implies that the output response will be invariant to process gain changes using Pole Placement. The same will be true for Generalized Predictive Control if no control weighting is used (Output Horizon and Detuned Model-Following configurations) or if $\lambda \propto [B(1)]^2$ (Lambda Weighting configuration). Since the open-loop zeros are retained in the closed-loop for both GPC (eqn. (9.1.1)) and PP (eqn. (9.1.3)), the output response will vary slightly for changes in process dynamics. The output performance is more consistently maintained using the Detuned Model-Following configuration or Pole Placement strategy compared with the Output Horizon or Lambda Weighting configurations (compare Figures 5.8 and 6.2 with 5.6 and 5.7b).

9.4.2 Variations in Dead-time

The ability of both the GPC and PP controllers to compensate for variable dead-time is a consequence of overparameterization of the B polynomial. Excellent performance was observed using either algorithm for large changes in the time delay (compare Figures 7.15 and 8.6).

9.4.3 Variations in Process Order

GPC is known to be robust against variations in the order of the process which result in an overparameterized model (ie. an overparameterized A polynomial) (Clarke et al., 1987a). This is not true for PP control; instability frequently results when near common factors arise in the estimated model polynomials $\hat{A}$ and $\hat{B}$ (see Figure 8.7). The GPC and PP controllers appear to be equally robust to model-plant mismatch when the model is underparameterized (compare Figures 5.19, 5.21 and 5.22 with 6.5).

9.5 Closed-Loop Response Characteristics

Some qualitiative comments can be made regarding the closed-loop response obtained using PP or GPC control. Less control effort is expended (not suprisingly) when the Lambda Weighting configuration is employed. However, for many applications, a slightly more oscillatory response is obtained compared with the other two configurations of GPC and with PP control (compare Figure 5.7b with 5.6, 5.8 and 6.2; note the different scales for the control signal).

9.6 Summary

The primary limitation of the Pole Placement algorithm is the inability to provide acceptable control performance when the model (i.e. A polynomial) is of higher order than the process. This limitation is overcome by the prediction-based GPC algorithm. The GPC approach requires less computational effort than the standard PP strategy. Both self-tuning algorithms are highly effective in maintaining the output performance despite variations in process gain, dynamics or dead-time.

10. PERFORMANCE ADAPTIVE CONTROL

In the ideal case where an accurate dynamic model of the process is known at all times, the closed-loop response can be made invariant to process gain changes using either Generalized Predictive Control or Pole Placement. However, this assurance cannot be given under more realistic conditions when there is

a) model-plant mismatch, or
b) changes in process dynamics

And even if no model-plant mismatch existed, the appropriate settings of the controller tuning parameters yielding the desired closed-loop behavior are not always known *a priori*.

Therefore, consider the introduction of an "outer loop" which monitors the closed-loop system and adjusts controller parameters so that the performance meets user specifications. Such a "performance supervisor" (or "performance tuner") would carry out the initial selection of tuning parameters to **achieve** desired performance and thereafter make minor alterations as necessary to **maintain** consistent behavior in the presence of modelling errors and variations in process dynamics. Since the underlying self-tuning GPC or PP controller has a structure with "two degrees of freedom", it should be possible to meet specifications on both the response to setpoint changes and the rejection of disturbances.

It must be recognized that a performance supervisor cannot be expected to tune the closed-loop system when the underlying controller is beyond its range of capabilites. The performance tuner simply carries out a local optimization of the controller tuning parameters once the control algorithm and structural parameters (e.g. model order, sample time, etc.) have been selected. The selection of the "best" controller and determination of these structural parameters would be the responsibility of a higher level "global off-line supervisor". The supervisory shell for an (adaptive) controller is envisioned to consist of the two hierarchical layers shown in the following figure:

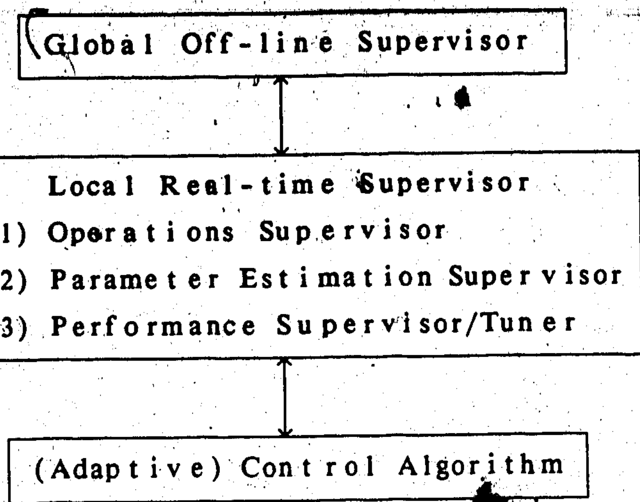


Figure 10.1 Hierarchical Structure for Adaptive Control Supervisory System

The development of a complete control supervisory system is beyond the scope of this work. However, in order to give the reader a perspective on where the performance supervisor would fit into the overall scheme, some of the functions of a supervisory shell will be outlined in section 10.1. The emphasis of the chapter is on a prototype performance tuner suitable for either Generalized Predictive or Pole Placement control. Two aspects of the performance loop are dealt with in section 10.2; what constitutes a reasonable set of user specifications for servo and regulatory control and how should the controller tuning parameters be adjusted to meet the desired performance. Sections 10.3 and 10.4 contain simulations and experimental runs which demonstrate the operation of the performance tuner.

10.1 Supervisory Shell for an Adaptive Controller

There is a large discrepancy between simulation algorithms and practical adaptive controllers which must handle nonlinearities, unmodelled dynamics, measurement noise, etc. Reported successful applications almost always include "safety-nets" and the use of "special tricks" to overcome limitations due to the violation of assumptions made in the theoretical development of the controller (Wittenmark and Åström, 1984). The idea of incorporating the "special tricks" into a supervisory shell for an adaptive controller is not new; an early paper on the subject (Schumann et al., 1981) proposed a startup phase for determination of structural parameters followed by supervision of parameter estimation during closed-loop operation. Isermann and Lachmann (1985) expanded the functions of the on-line supervisor to include monitoring closed-loop stability with appropriate modifications to the controller design.

More recently, the use of an expert system has been suggested to assist in the task of commissioning and operating self-tuning controllers (Åström, 1985; Sanoff and Wellstead, 1985). Expert systems are computer programs which contain knowledge in a specific area of human expertise, sufficient to perform as consultants. The body of knowledge is organized (often as a set of production rules) in a knowledge base distinct from the inference engine which applies the expertise to solve the problem at hand. As a consequence, the systems knowledge can be easily expanded and modified. Expert systems have traditionally been applied for off-line problem solving, such as diagnostics, but for supervision of adaptive control a real-time system is necessary (Arzen, 1986). It is also important that the system have a well-developed user-interface to allow the operator or control engineer to add or change "rules" and thereby build the system incrementally, based on experience.

In this thesis it is proposed that the supervisory system consist of a global off-line supervisor and a local real-time supervisor as indicated in Figure 10.1. The global supervisor is responsible for the selection and configuration of the set of control, estimation and other routines which are best suited to the control problem at hand. At the present time this task is best performed off-line by control engineers, with the assistance of computers for computation, simulation, etc. The automation of this high-level task is beyond current capabilities.

With the selection and configuration of the control system complete, a real-time supervisor may be used to monitor closed-loop performance and locally optimize controller and estimator tuning parameters. The real-time local supervisor would also be responsible for monitoring the operations which take place within a process control environment (e.g. alarms, errors, etc.).

In the following subsections, the functions of the global and local supervisors will be outlined. A **brief** review of the current state of technology with regards to implementation of some of these functions will also be given.

10.1.1 Global Off-Line Supervisor

Some of the functions involved in the selection and design of a control system which would be carried out by the global off-line supervisor include:

- identification of process characteristics
- identification of appropriate model structure(s), model order(s) and dead-time (as a function of operating conditions)
- selection of sample time.
- selection of the "most suitable" control algorithm from a set of available routines
- selection of an appropriate parameter estimation algorithm

At the present time, a pre-tune test is often used to obtain initial process knowledge. For stable overdamped processes, a open-loop step change in the control signal giving rise to a "process reaction curve" (Stephanopoulos, 1984) may be used to estimate the dead-time, process gain and dominant time constant. An indication of the amount of measurement noise present may be obtained by calculating the variance of the output once steady state has been reached. As an alternative, a relay signal may be used to obtain process information in terms of the critical gain and period (Åström, 1985). The sample time may be computed from the dominant time constant and time delay using rules available in the literature (e.g. Isermann, 1981). Sela and Ray (1986) discuss implications of the choice of sample time on parameter estimation and control.

Schumann et al. (1981) describe a search technique for identification of model order and dead-time which involves performing parameter estimation for a number of assumed structures and comparing how well each model predicts the plant output. A better criteria for selection of model structure would be how well a controller designed based on a model behaves in closed-loop control of the actual process. Liu and Gertler (1987b) introduce a closed-loop performance index by designing controllers for each low order candidate model and comparing closed-loop transfer functions derived with a high order reference model representing the plant. The procedure is, however, quite lengthy and relies heavily on the quality of the reference model.

Currently there is much disagreement in the literature on which control and estimation algorithms are most suitable for various applications. Each research group has their favorite methods which, unfortunately, are not always evaluated in an objective manner. This thesis evaluates only the Generalized Predictive and Pole Placement controllers.

One of the criticisms of adaptive control is concerned with the large number of controller and identification parameters which must be specified by the user. The global off-line supervisor would also be responsible for configuration and startup functions including:

- initial configuration of the controller and estimator
- selection of conservative initial settings for the controller and estimator design parameters
- design of a simple, safe, backup controller (e.g. PI)
- ensuring that the user performance specifications are realistic

For example, if the Generalized Predictive Controller was selected for an application, the global supervisor would choose the particular configuration (Output Horizon, Lambda Weighting, or Detuned Model-Following) to be implemented. If in turn, the Output Horizon configuration was selected, the global supervisor would set NU, N_1, P and λ to the appropriate default values given in section 4.2. A conservative value of the active tuning parameter, N_2, (which affects both servo and regulatory control) would be selected (i.e. N_2 = settling time of process in sampling intervals) along with a detuned setting for the controller design polynomial, C_c, (which affects stability and disturbance rejection) (ie. $C_c=(1-.9q^{-1})^{na}$). Assuming use of a constant trace RLS identification algorithm for parameter estimation, the global supervisor would recommend conservative values for the trace of the covariance matrix (gain of the estimator) and the estimator filter polynomial (ie. $C_e=(1-.9q^{-1})^2$). All of these initial choices would be subject to later adjustment by the local real-time supervisor.

Finally, the global supervisor must ensure that the user specifications on the desired closed-loop performance are realistic and may be achieved given the characteristics of the process, the operating constraints (i.e. actuator limits) and the control and estimation strategies selected.

10.1.2 Local Real-Time Supervisor

The local supervisor carries out a number of real-time tasks to monitor and optimize operation of the adaptive control system. It is proposed that the local supervisor consist of the following modules:

1) Operations Supervisor (or Executive)

2) Parameter Estimation Supervisor

3) Performance Supervisor/Tuner

10.1.2.1 Operations Supervisor

The operations supervisor is concerned with the tasks of the local on-line supervisor not specifically associated with the parameter estimation or control routines. This includes, but is not restricted to:

- monitoring plant alarms related to the control loop
- choosing the active performance specifications and constraints from the user-supplied set
- detecting instabilities by monitoring closed-loop behavior
- performing fault diagnosis to identify the source of severe problems with the adaptive controller
- identifying whether the adaptive controller is beyond its range of capabilities (requiring attention of global supervisor or operator)
- recommending "recovery procedures" to restore safe operation of the loop (e.g. use of backup controller)
- detecting and eliminating erroneous measurements, (due to instrument failure, etc.) data handling errors, "outliers", etc. (e.g. by using a velocity limiting filter)

- **performing operator communications including:**
  - **advising operator of current status**
  - **recommending operator actions**
  - **allowing modifications to knowledge-base**

Ensuring that the closed-loop system remains stable is the most important consideration of the operations supervisor. Practical experience indicates that adaptive controllers may go unstable if there exits a large amount of model-plant mismatch (relative to the desired closed-loop bandwidth) due to a poor model structure or large process change. For any corrective action to be taken, the monitor must detect if the system is becoming unstable or moving towards the limit of stability. Several heuristic instability monitors have been proposed in the literature (e.g. Isermann and Lachmann, 1985). Gertler and Chang (1986) provide a review of the subject and suggest the combination of filters, rectifiers and trend analyzers to form an instability indicator. Liu and Gertler (1987b) suggest a scheme incorporating filtering the absolute value of the error which makes possible a uniform handling of oscillatory and non-oscillatory signals and reduces the effects of noise.

Once a trend towards instability is detected, action must be taken to restore safe operation of the loop either by detuning the adaptive controller or switching to a safe fixed-gain backup controller. Liu and Gertler (1987a,b) demonstrate the former alternative for a Pole Placement controller. Using Pole Placement, closed-loop stability is guaranteed if both the open-loop plant and identified model are stable and the desired closed-loop poles are specified to be sufficiently close to the poles of the model. Thus they suggest moving the desired closed-loop poles towards the model poles incrementally until stability is restored. For the Output Horizon configuration of Generalized Predictive Control, Appendix A contains a proof indicating that the closed-loop will be stable for large values of N_2 as the algorithm tends toward a mean-level controller. For the Lambda Weighting and Detuned Model-Following configurations of GPC, stability can also be brought about if the active tuning parameter is used to detune the controller sufficiently.

With stability restored, the next logical step is to carry out fault diagnosis to identify the source of the problem. It is important to determine if there has been a large change in the process such that the adaptive controller is beyond its range of operation. The operator or global supervisor should be notified in this case.

10.1.2.2 Parameter Estimation Supervisor

The main objective in supervising the parameter estimation procedure is to ensure that the model identified is a "reasonable" representation of the process. It is important not to update the controller based on parameter estimates that are grossly in error (Seborg et al., 1986). The parameter estimation supervisor should:

- check if the gain and poles of the discrete-time process model are realistic
- monitor the prediction error, parameter estimates, etc. for abrupt changes, ringing, etc.
- adjust design parameters of the estimation algorithm (e.g. trace of covariance matrix, etc.) in response to changes in operation (e.g. setpoint changes) or indications of model acceptability
- carry out parameter "scheduling" (resetting) if parameters are known functions of operating conditions

Robust control requires a model which is accurate around the cross-over frequency (Wittenmark and Åström, 1984). To estimate a good model it is necessary to monitor the excitation in the relevant frequency band; when the control signal generated by the controller is not persistently exciting extra perturbation signals must be added or the estimation algorithm must be switched off. The parameter estimation supervisor would be responsible for,

- monitoring excitation
- detecting disturbances (versus a change in the process)
- deciding whether the parameter estimator should be on or off

Several of the modifications to the basic RLS algorithm are intended to avoid difficulties when there is insufficient excitation (Shah and Cluett, 1987). However, the selection of the on/off "thresholds" and other estimator tuning parameters are often application specific. Much work remains to be done to create a universal parameter identification system.

10.1.2.3 Performance Supervisor/Tuner

Industrial implementation of adaptive control has been hindered by the fact that the benefit of improved control is outweighed by the risk of poor behavior as a result of modelling and controller design errors. Self-tuning controllers are equipped with tuning parameters which compensate for model-plant mismatch, usually by slowing the closed-loop speed of response and trading-off performance for robustness. Fixed tuning parameter settings do not take into consideration that the accuracy of the process model changes over time. A supervisor could be used to monitor the actual closed-loop performance and modify tuning parameters to reflect changes in model accuracy in order that the behavior meet user specifications.

In the following section, the performance supervisor, shown in block diagram form in Figure 10.2, will be discussed at greater length.

10.2 Performance Supervisor/Tuner

The role of the performance supervisor is to adjust controller tuning parameters so that the actual measured closed-loop performance meets user-specifications. The vision of an ideal adaptive controller which requires no user input is unrealistic; it is at least necessary to tell the controller what it is expected to do (Wittenmark and Åström, 1984). However, the specifications given to the adaptive controller should be in terms of performance expected; the user should not be required to set values for tuning parameters which are abstract or otherwise unrelated to closed-loop behavior.

10.2.1 Closed-Loop Performance Specifications

Universal agreement on what constitutes a reasonable and complete set of specifications for closed-loop performance has not been reached. The following closed-loop specifications have been suggested or used for demonstration purposes:

- **Bandwidth and damping (Wittenmark and Åström, 1984)**
- **Dominant pole response (damping ratio and oscillation frequency) (Jiang, 1987)**
- **Error damping or control signal variance (Pollard and Brosilow, 1986)**
- Overshoot **and damping of error (Myron, 1986; Bristol and Kraus, 1984)**
- Overshoot for setpoint changes (Minter and Fisher, 1988)

Methods for altering controller parameters such that actual performance meets specifications have also been proposed. However, these authors have either ignored regulatory control altogether or attempt to tune parameters to reach a comprise between the response to setpoint changes and rejection of disturbances. Clearly, a better approach is to consider separate specifications for servo and regulatory performance.

10.2.1.1 Servo Response Specifications

The set of servo performance criteria which should be specified by the user depends upon the objectives of the particular application. Although there are many different setpoint following objectives, two situations are commonly encountered.

In the first situation the application calls for control as "tight" as possible. An appropriate strategy would be to minimize the response time subject to constraints on overshoot and/or damping. The user would be required to specify the manimum allowed overshoot (and/or damping) to step setpoint changes (Since damping and overshoot are not independent, if both are specified the algorithm would act on whichever is closest to the constraint). Note that when the supervisor has finished tuning the loop, the actual measured overshoot (or damping) should be near its target value. This approach is used by the Foxboro EXACT to tune PID controller constants (Kraus and Myrom, 1984). Minimization of the response time with an overshoot constraint may be an unrealistic performance specification for highly damped process with little dead-time which cannot easily be made to respond in a oscillatory manner. The supervisor will continue to increase controller "gains" in an effort to force the output to overshoot following setpoint changes; this results in excessive control action (Minter and

Fisher, 1988)

For many chemical process control applications a very rapid resonse to setpoint changes is not desirable; the user is looking for overdamped behaviour. In this case, it is more appropreate for the user to specify the desired rise or settling time. While any defined rise or settling time may be used, the 63% rise time (closed-loop time constant) has been assumed in the following. Approximate knowledge of the dominant open-loop time constant at one operating point (determined from a pre-tune test) may be used as a reference to specify the desired closed-loop time constant; a good default is simply the open-loop time constant. In the event that the specified 63% rise time does lead to a underdamped response, the user should also provide limits on the allowable overshoot (and/or damping). In contrast to the previous strategy, in the majority of cases the actual overshoot and damping will be below the limits specified when the target 63% rise time is met.

10.2.1.2 Regulatory Response Specifications

It is much more difficult to provide reasonable specifications on the regulatory performance of a controller. This is a direct consequence of the unknown nature of the types of disturbances encountered. If measurement noise or other stochastic disturbances are continually present, the control signal variance (measured when the setpoint is constant) may be used as a performance criterion. Control engineers frequently tune loops based on keeping the average level of control output variations below an acceptable level. If on the other hand, the process is subject to infrequent, sudden load changes the error damping (i.e. decay ratio) is a better performance index. The estimate of the amount of measurement noise present, obtained during the pretune phase, may be used as a basis for selecting the most appropriate measure of regulatory performance. *A priori* knowledge of the anticipated disturbances would also weigh in the decision as to which specification is most appropriate.

10.2.2 Controller Adjustment Mechanisms

Even in the absence of model-plant mismatch, noise, and other uncertainties, a functional relationship between the controller tuning parameter settings and the resulting closed-loop behavior is not known explicitly. The settings of the tuning parameters yielding the desired performance must be found by an optimization or search procedure. Past researchers have proposed mechanisms to adjust controller tuning parameters such that the measured closed-loop performance is equal to the specified target value. For example, Pollard and Brosilow (1986) modify the filter time constant, λ_f, of an Internal Model Controller according to:

$$\lambda_f(t) = \lambda_f(t-1) + w(t)\ (P_m(t)-P_d) \tag{10.2.1}$$

where $P_m(t)$ = measured performance, and P_d = desired performance

However, the choice of the weighting factor, w(t), was not given in the reference.

For the purpose of this thesis, the following simple general tuning parameter adjustment equation will be used:

$$t_p(t) = t_p(t-1) \pm \frac{k\left[P_m(t) - P_d\right]}{P_d}\ t_p(t-1) \tag{10.2.2}$$

where t_p is the tuning parameter setting and k is an adjustment gain factor frequently assigned the value of 1. Rearrangement of (10.2.2) into the form:

$$\frac{t_p(t) - t_p(t-1)}{t_p(t-1)} = \pm \frac{k\left[P_m(t) - P_d\right]}{P_d} \tag{10.2.3}$$

makes it apparent that the percentage change in the tuning parameter setting is proportional to the percentage difference between the measured and desired performance. Thus, both the performance and tuning parameter values are "normalized". Note also that $t_p(t-1)$ is the tuning parameter value in use prior to the current adjustment. The tuning parameter is not normally adjusted every sampling interval (but rather after a setpoint change (servo) or after a number of sampling intervals (regulatory)).

Since both Generalized Predictive and Pole Placement controllers have "two degrees of fredom" it is possible to adjust appropriate tuning parameters to satisfy specifications on servo and regulatory peformance.

10.2.2.1 Maintaining Servo Performance

For Generalized Predictive Control, the active tuning parameter which should be adjusted to maintain servo performance at user specifications depends upon which of the three configurations of GPC outlined in Chapter 4 has been selected:

Configuration	Active Tuning Parameter
Output Horizon	maximum output horizon, N_2
Lambda Weighting	relative control weighting, λ_{rel}
Detuned Model-Following	inverse closed-loop model, $P(q^{-1})$

Since we are interested in maintaining a consistent output response, for the Lambda Weighting configuration the actual control weighting should be scaled with $[\hat{B}(1)]^2$ (i.e. $\lambda=[\hat{B}(1)]^2\lambda_{rel}$). Recommendations regarding the structure of $P(q^{-1})$ were given in section 4.4.1: for a process adequately represented by a first order model, $P(q^{-1})$ may be selected as a first order polynomial. For higher order plants, a 2nd order $P(q^{-1})$ polynomial results in less control activity. In situations where little overshoot can be tolerated, ζ (the damping factor) should be taken as ≥ 1 and τ (the natural period) used to vary the speed of response. For Pole Placement Control, the desired closed-loop characteristic polynomial, $P(q^{-1})$, is used to modify the servo response. It may be selected in the same manner as the inverse closed-loop model of (Detuned Model-Following) GPC.

10.2.2.2 Maintaining Regulatory Performance

The controller design polynomial, $C_c(q^{-1})$, has been demonstrated to be very effective for tailoring the rejection of disturbances and reducing the negative effects of model-plant mismatch, when using Generalized Predictive or Pole Placement control. This tuning parameter has very little impact on the servo response provided that the closed-loop is stable. The simulation and experimental runs of previous chapters have also demonstrated that

$C_e(q^{-1})$ has a strong influence on the variance of the control signal. For both GPC and PP, this controller design polynomial should be selected as

$$C_e = (1-c_1q^{-1})^{na}$$

where c_1 is the tuning parameter adjusted by the supervisor (within limits of 0 to 1) to meet regulatory performance specifications and na is the model order. For practical applications, it may be desirable to further limit the range of acceptable values for c_1 (e.g. $0.5 \leq c_1 \leq 0.95$).

10.3 Performance Adaptive Control Simulations

In this section, the operation of the performance supervisor/tuner is demonstrated for self-tuning Generalized Predictive and Pole Placement control of the third order process,

$$G_p(s) = \frac{1}{(1+s)(1+3s)(1+5s)}$$

referred to earlier as Process C (Table 5.1). A first order model is identified using RLS with the trace of the covariance matrix equal to 0.02. The regressor is bandpass filtered ($C_e=(1-.9q^{-1})^2$) in order to obtain a good low frequency model inspite of the unmodelled dynamics. The difference equation

$$Ay(t) = Bu(t-1) + d_u(t) + Av(t) \tag{10.3.1}$$

was used to simulate the process where steps in $d_u(t)$ of magnitude 0.02 were introduced 40 samples after each upward-going setpoint change and removed 40 samples later. Gaussian noise, v(t), with a standard deviation of 0.05 was added to the output continuously.

The overshoot and closed-loop time constant were calculated by observing the response for 20 iterations after each setpoint change. The regulatory performance was evaluated by measuring the control signal variance using the recursive equations:

$$\bar{u}(t) = (1-\lambda_{um})u(t) + \lambda_{um}\bar{u}(t-1) \tag{10.3.2}$$

$$S_u^2(t) = (1-\lambda_{uv})\left[u(t) - \bar{u}(t)\right]^2 + \lambda_{uv}S_u^2(t-1) \tag{10.3.3}$$

where the forgetting factors λ_{um} and λ_{uv} were choosen to be relatively small (i.e. < 0.9) so that only high frequency variations in u(t) contribute to the variance. The variance was not updated during the transient following each setpoint change.

The performance tuner was started 100 iterations from the start of a run. The tuning parameter which affects servo control was adjusted once following each setpoint change. The regulatory tuning parameter was adjusted every 30 samples.

10.3.1 Generalized Predictive Control

Two simulation runs were conducted based on the Output Horizon configuration of GPC. For the first of these, the servo objective was to meet the user-specified 63% rise time of 6 sample intervals. The regulatory performance objective was to keep the variance of the control signal at the desired value of 0.01. Results are shown in the four parts (a to d) of Figure 10.3. The regulatory tuning parameter adjusted using equation (10.1.2) was actually the equivalent continuous time root, τ_c, of $C_c=1-c_1q^{-1}$ ($\tau_c=-T_s/\ln c_1$) which is expected to be more linearly related to the variance of the control signal than the discrete-time root, c_1. The controller was initialized with Output Horizon, $N_2=5$ and the design polynomial $C_c=1-.8q^{-1}$. Even after the open-loop startup, during which time the model parameters converge, these tuning parameter settings yield oscillatory behavior as a consequence of the model-plant mismatch and measurement noise. By increasing N_2 and c_1 (Figure 10.3d) the performance supervisor is able to improve the response dramatically such that the closed-loop system meets both the servo and regulatory performance specifications (Figure 10.3c).

For the second run, the servo objective was to minimize the response time subject to the constraint that the overshoot not exceed 20%. The closed-loop response and measured performance are illustrated in Figures 10.4a and b. Model parameter trajectories (which were almost identical to those of Figure 10.3b) and the tuning parameters adjusted by the performance supervisor are not plotted. As the run progresses, both the overshoot and control signal variance approach the desired target values.

10.3.2 Pole Placement Control

In parallel with the GPC runs, Pole Placement runs were conducted for the two different servo objectives. In both cases, a first order characteristic polynomial ($P=1-p_1q^{-1}$) and first order regulatory design polynomial ($C_c=1-c_1q^{-1}$) were selected. The performance supervisor adjusted the continuous time roots of these polynomials.

The four parts of Figure 10.5 contain results for the case where the desired performance is expressed in terms of a closed-loop time constant of 6 samples (for setpoint changes) and regulatory control signal variance of 0.01. The initial settings of $p_1=0.8$ and $c_1=0.8$ give rise to underdamped behavior. The performance tuner progressively detunes the loop by incrementing both p_1 and c_1 (Figure 10.5d) until the performance specifications are reached (Figure 10.5c). Note that without the performance supervisor, the pole of the desired characteristic polynomial would normally have been set by the designer to be $p_1 = e^{-T/\tau_{CL}} = e^{-1/6} =$ 0.85. The performance tuner finds that as a result of modelling errors a value of $p_1=0.87$ is necessary such that the **actual** closed-loop time constant is 6 sampling intervals.

The performance tuner works equally well when the setpoint tracking objective is to minimize the response time while keeping the overshoot below 20% as shown in Figure 10.6a,b. However, the observed overshoot is quite sensitive to measurement noise, since it is determined from only one point on the response curve (the first peak). As a consequence, the performance tuner may adjust the servo tuning parameter frequently when there is little reason to do so. A more robust technique would involve "fitting a curve" to the data such that the measurement of the overshoot is less susceptible to noise.

10.4 Performance Adaptive Control of the Stirred Tank Heater

In Chapter 7 and 8, it was shown that proper selection of the regulatory design polynomial, C_c, is absolutely necessary for stable control of the Stirred Tank Heater, when operated without the inlet water deflector. The following experimental runs demonstrate how the performance supervisor automatically tunes this polynomial to yield the desired regulatory performance. The outer performance loop also allows the controller to meet independent specifications on the setpoint tracking properties of the closed-loop.

Standard operating conditions for the Stirred Tank Heater, including sample time, were described in section 7.1. Disturbances were introduced by increasing the inlet water flowrate from 53 cm^3/s to 66 cm^3/s for a period of approximately 200s during the time the setpoint was at 40 °C. A first order model was identified with tr(P)=0.1, and the estimator design polynomial, $C_e=(1-.8q^{-1})^2$. The performance supervisor was commissioned 500s from the start of each run. The response to setpoint changes was monitored for 100s before adjusting the servo tuning parameter. The regulatory tuning knob (C_c) was adjusted every 120s of regulatory control (control during any time except the 100s following a setpoint change).

10.4.1 Generalized Predictive Control

The runs with GPC were conducted based on the Output Horizon configuration although similar results can be achieved for Lambda Weighting or Detuned Model-Following control. Consider the situation where the user specifies a 63% rise time in response to setpoint changes of 30s and a regulatory control signal variance of 0.25. Figure 10.7 (a to d) contains results when the initial controller tuning parameters are intentionally initialized very poorly ($N_2=5$, $C_c=1$) for this system. The performance supervisor detunes the controller by incrementing both N_2 and c_1 (where $C_c=1-c_1q^{-1}$) (Figure 10.7d) until both the servo and regulatory performance specifications are met (Figure 10.7c).

Next consider the case where the objective for setpoint tracking is to maximize the speed of response while keeping the overshoot at or below 20%. To make the demonstration more realistic, the controller tuning parameters were initialized very conservatively as is normally recommended for startup ($N_2=20$, $C_c=1-.95q^{-1}$). As indicated by the results of Figure 10.8 (a & b), the performance supervisor "tightens" the servo response while keeping the control signal variance at the acceptable level. (The model and controller tuning parameter trajectories are not shown).

10.4.2 Pole Placement Control

The results for performance tuning of Pole Placement control are equally impressive. For the experimental run shown in Figure 10.9 (a to d), the controller tuning parameters were initialized poorly ($P=1-.5q^{-1}$, $C_c=1$)

and the closed-loop system appears unstable. The performance supervisor detunes both the regulatory and servo elements of the controller until the targets for the 63% rise time (30s) and control signal variance (0.25) are reached (Figure 10.9c). In the absence of model-plant mismatch, one would expect that with $p_1 = e^{-T/\tau_{CL}} = e^{-s/30} = 0.77$ the desired closed-loop time constant would be met. However, from Figure 10.9d, it is evident that the performance tuner finds a value of $p_1=0.88$ to be necessary.

The results for the final run with the performance supervisor are shown in Figure 10.10 (a & b). The servo objective of this run was to minimize the rise time with an overshoot constraint of 20%, given conservative initial tuning parameter settings ($P=1-.9q^{-1}$, $C_c=1-.95q^{-1}$). As pointed out earlier, the measurement of the overshoot is quite sensitive to noise; the performance tuner over-reacts in attempting to correct for these variations (Figure 10.10b).

10.5 Summary

A hierarchical structure for a computer control supervisory system consisting of a global off-line level and a local real-time level has been proposed. Some of the functions of the modules comprising the system have been outlined, and a prototype of the performance supervisor module developed. The performance supervisor (a module within the local real-time supervisor) monitors the actual response of the closed-loop system and adjusts controller tuning parameters to achieve and maintain user-specifications on both servo and regulatory performance, despite the inevitable presence of modelling errors. Experimental runs with the Stirred Tank Heater demonstrated the ability of the performance supervisor to minimize the response time to step setpoint changes subject to an overshoot constraint or achieve a desired 63% rise time. During regulatory control, the variance of the control signal was maintained at a user-specified level. The performance supervisor manipulated the active tuning parameter of the GPC or PP controller (i.e. N_2, λ or $P(q^{-1})$) as well as the regulatory control design polynomial $C_c(q^{-1})$ to accomplish its goal.

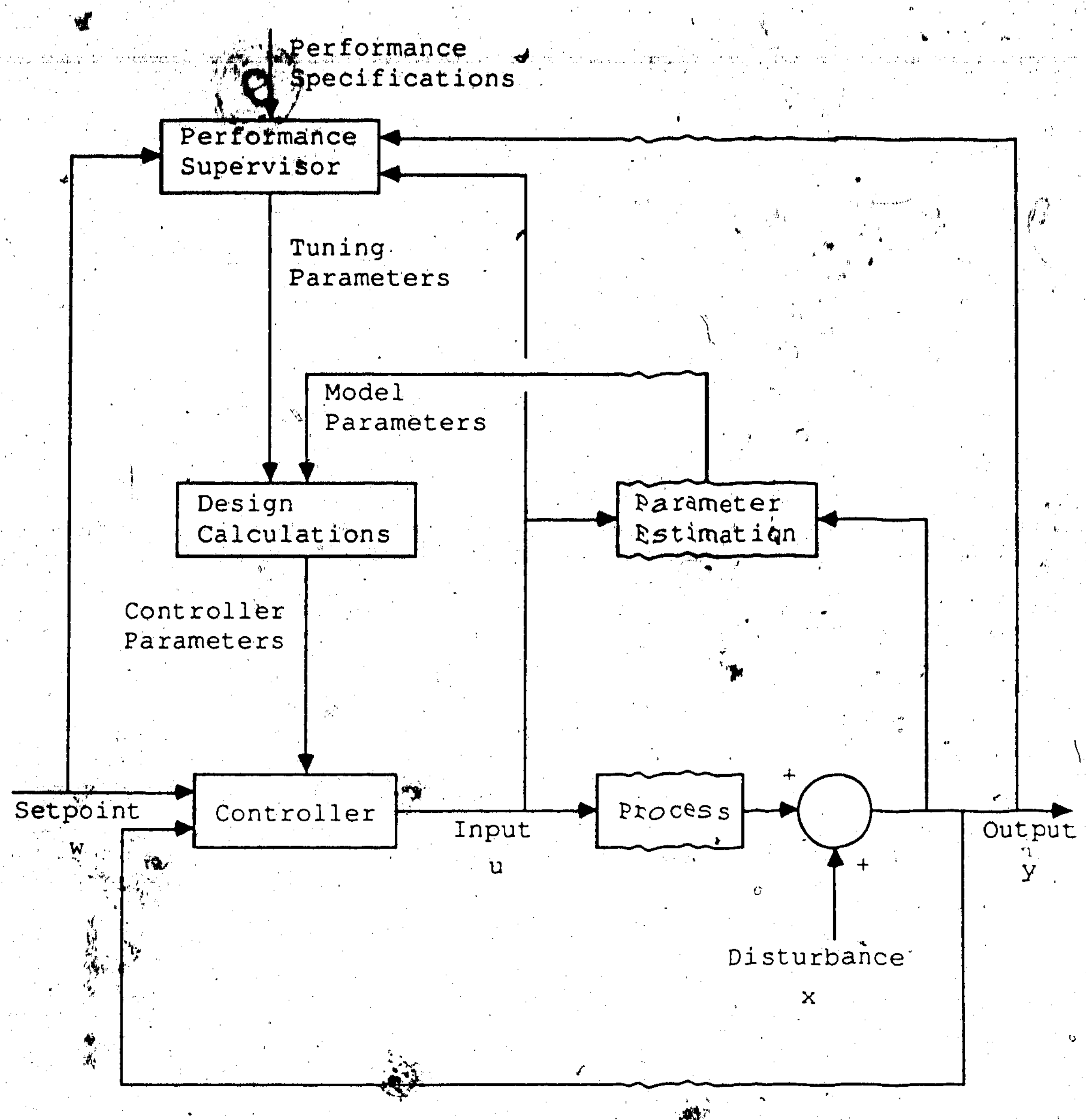


Figure 10.2 Block Diagram of Self-Tuning Controller with Performance Supervisor

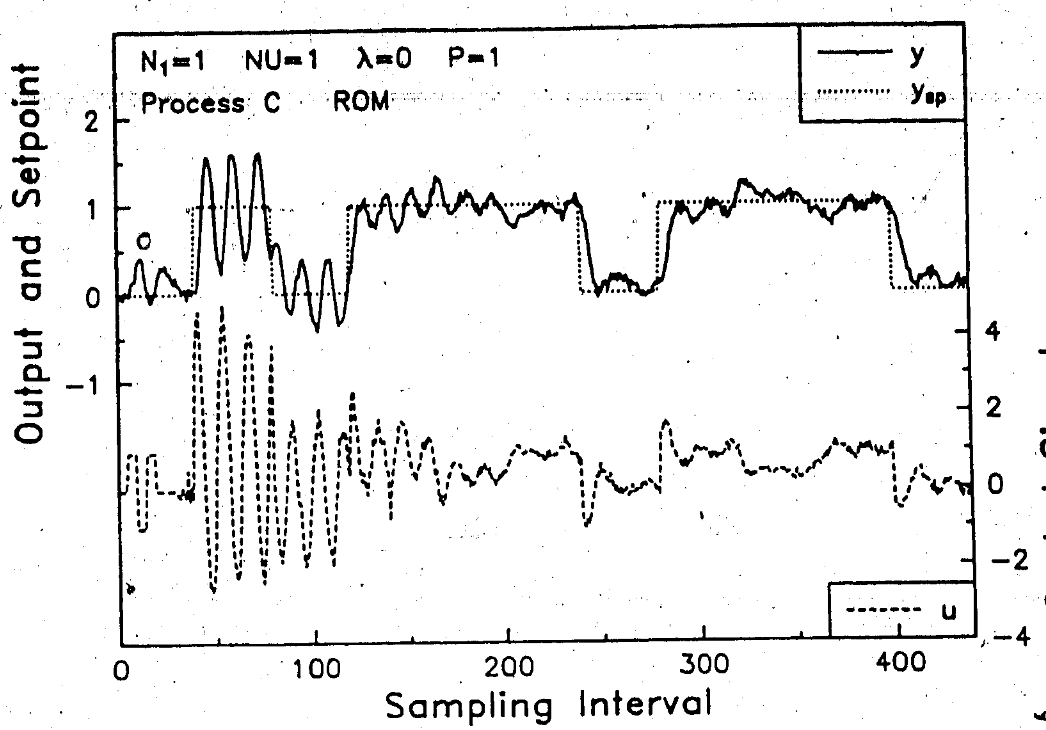


Figure 10.3a Performance Adaptive Control based on Output Horizon Configuration of GPC

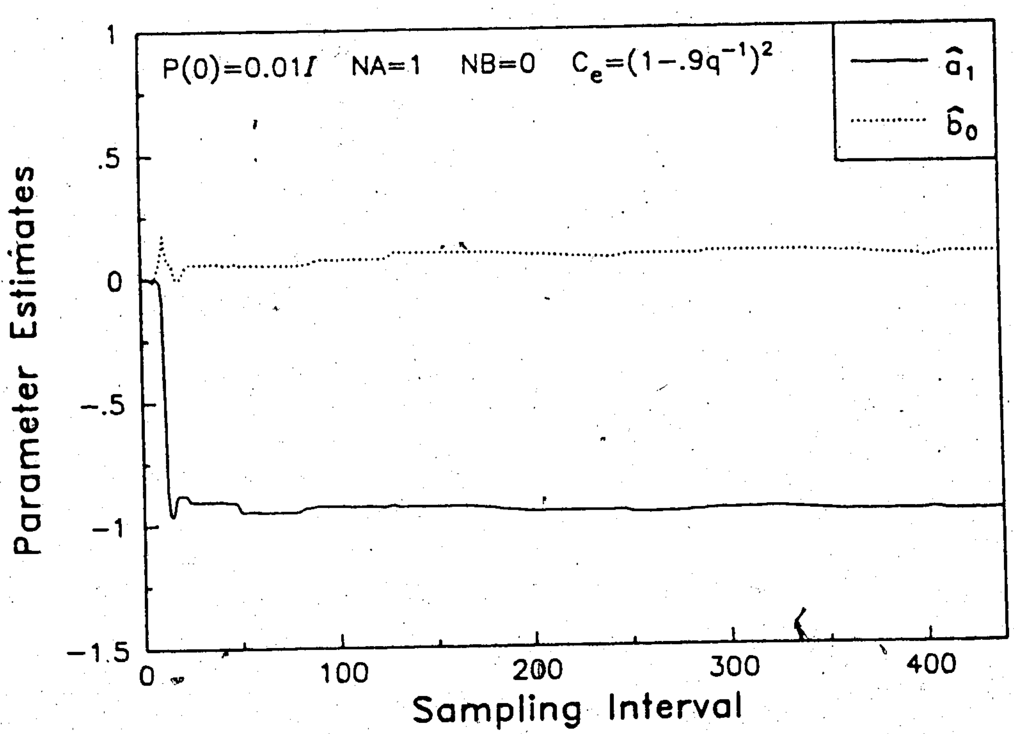


Figure 10.3b Model Parameter Trajectories

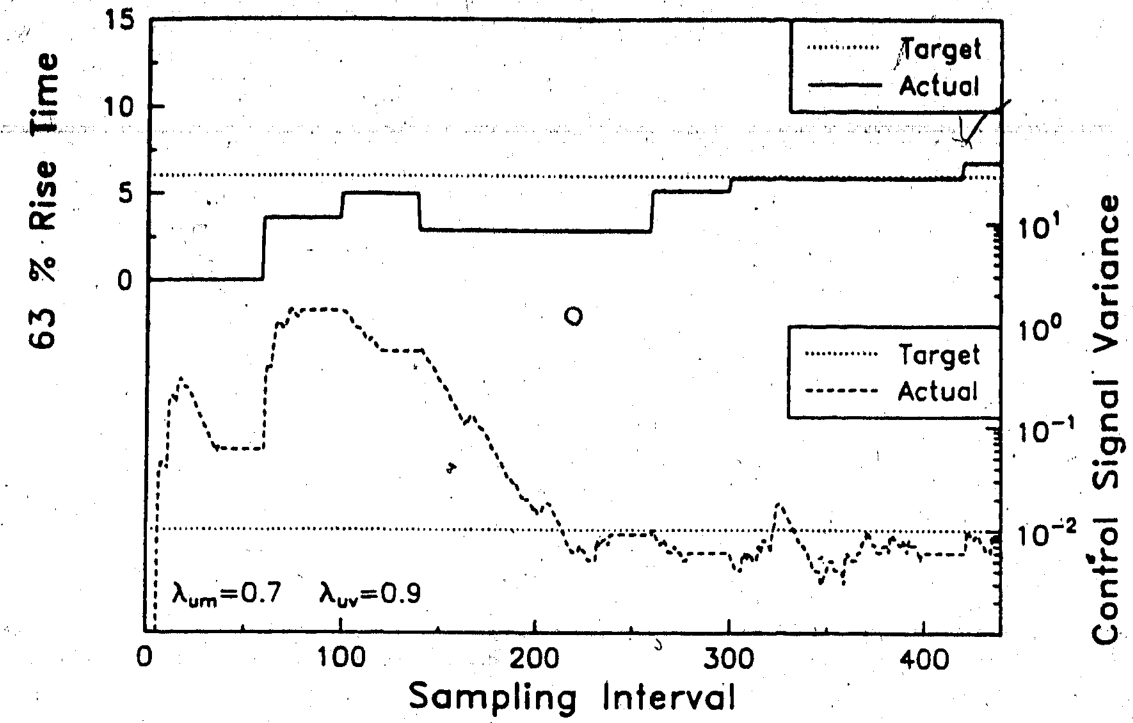


Figure 10.3c Actual and Specified Closed Loop Performance (Servo Objective: Desired 63% Rise Time)

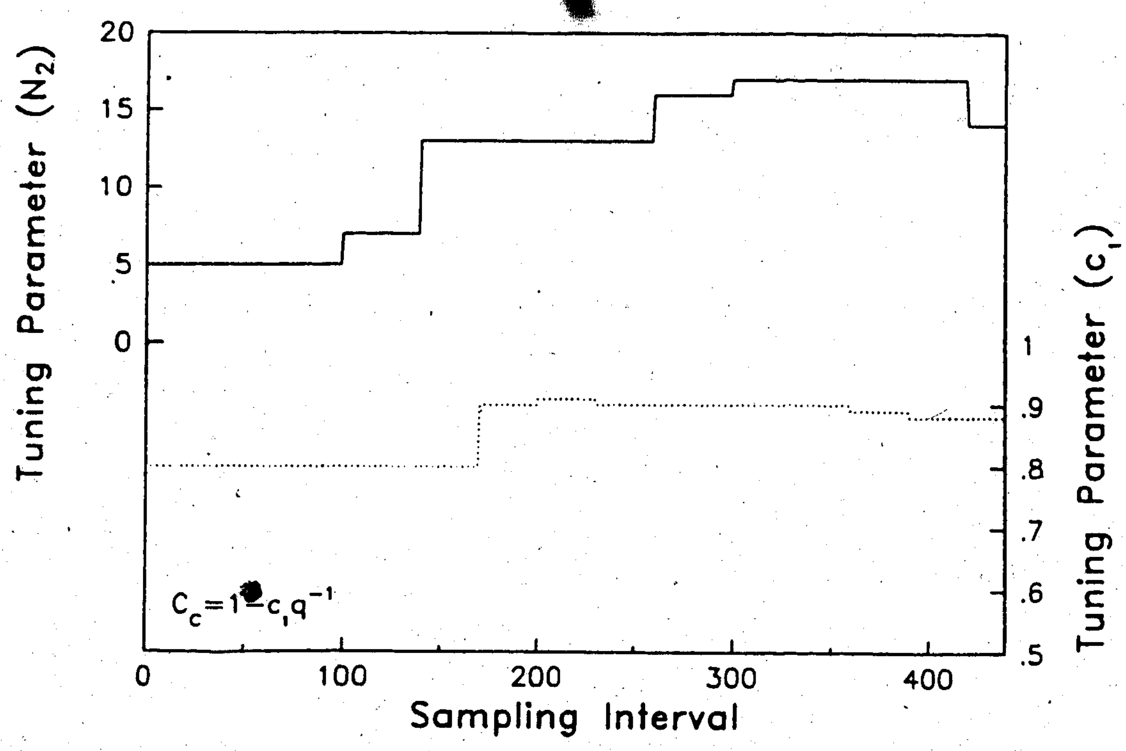


Figure 10.3d Controller Tuning Parameter Trajectories

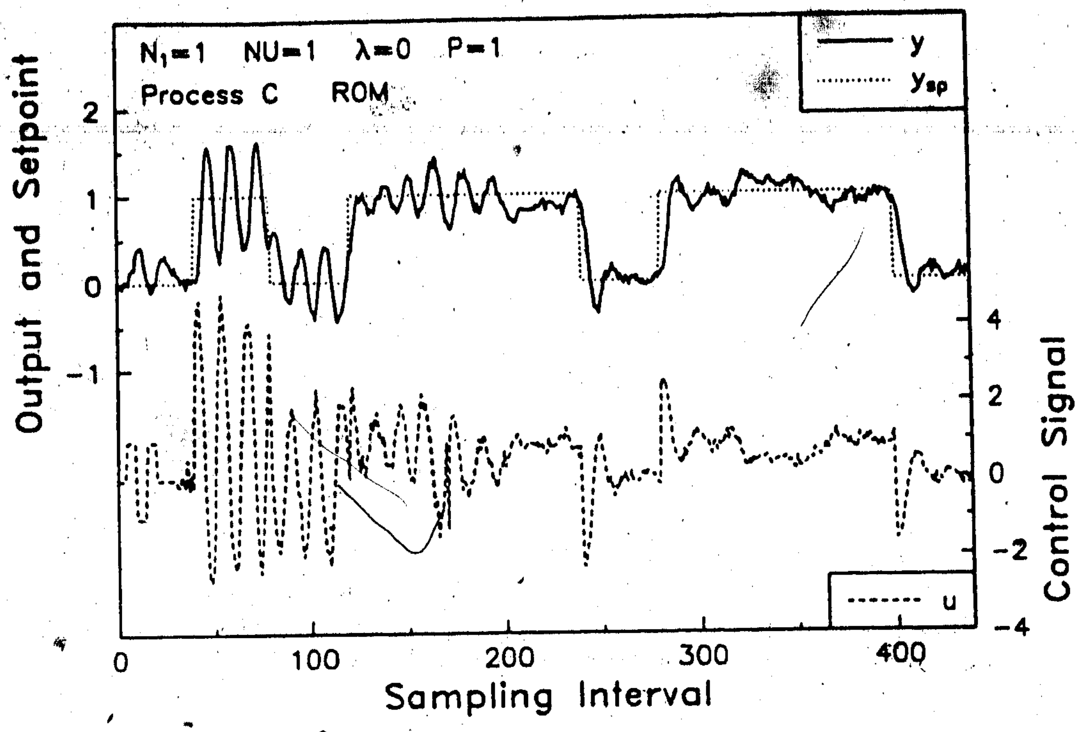


Figure 10.4a Performance Adaptive Control based on Output Horizon Configuration of GPC

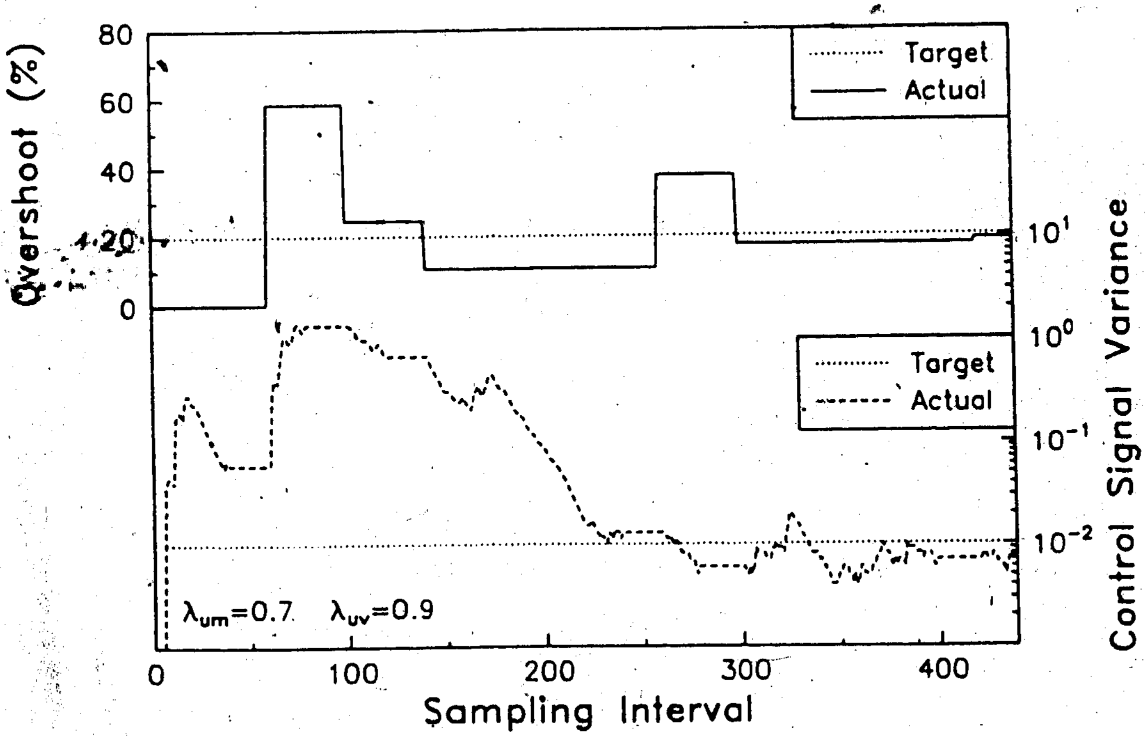


Figure 10.4b Actual and Specified Closed Loop Performance (Minimize Response Time with Overshoot Constraint)

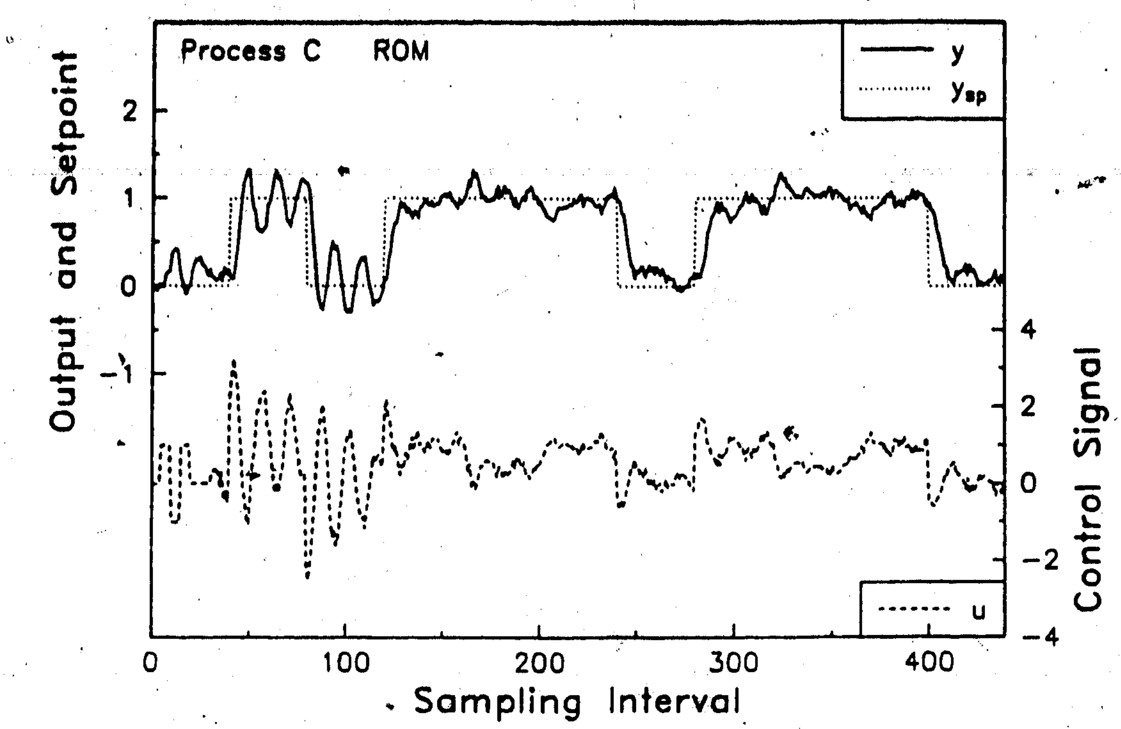


Figure 10.5a Performance Adaptive Control based on Pole Placement Algorithm

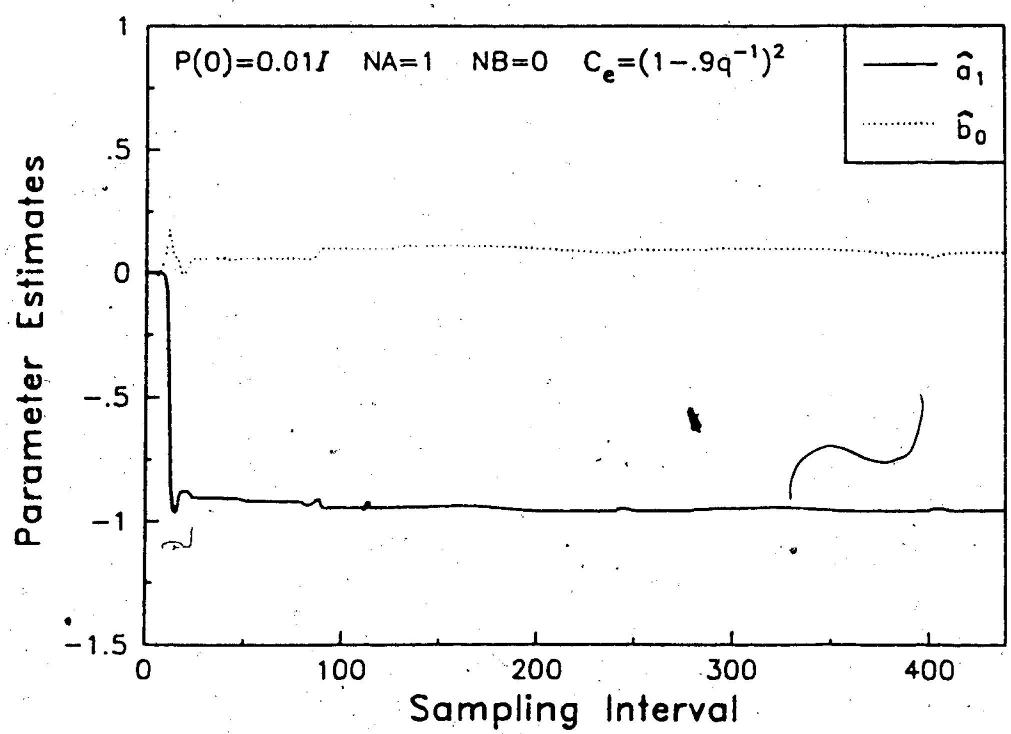


Figure 10.5b Model Parameter Trajectories

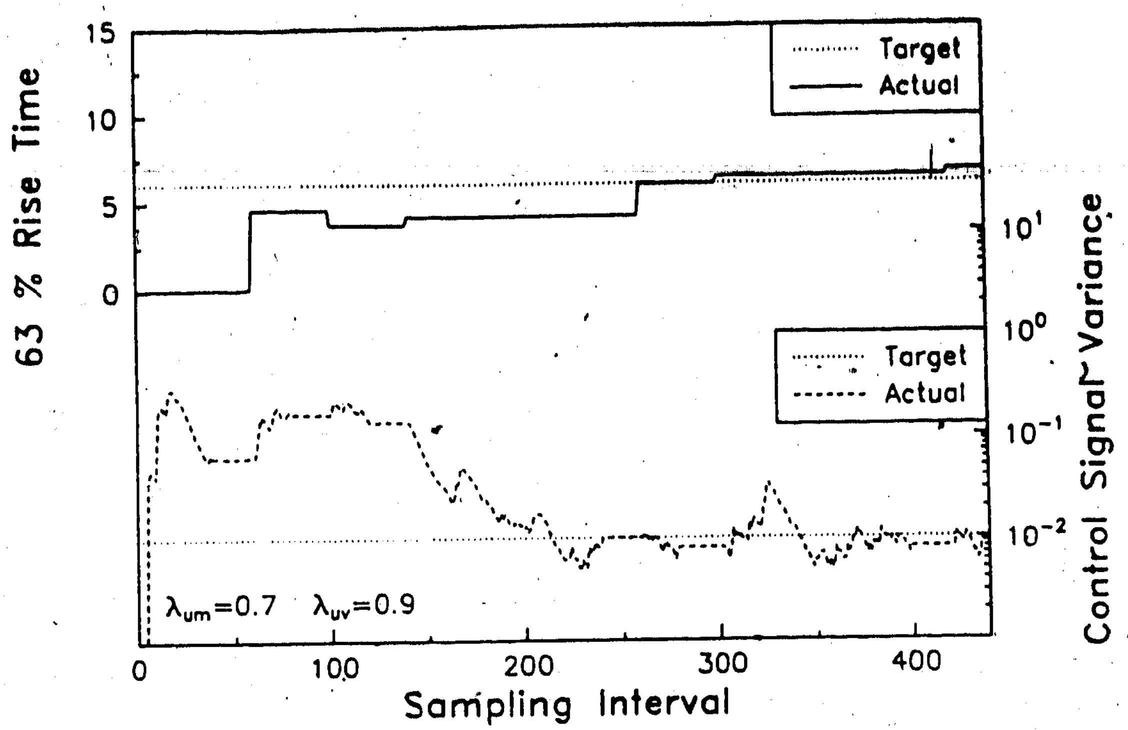


Figure 10.5c Actual and Specified Closed Loop Performance (Servo Objective: Desired 63% Rise Time)

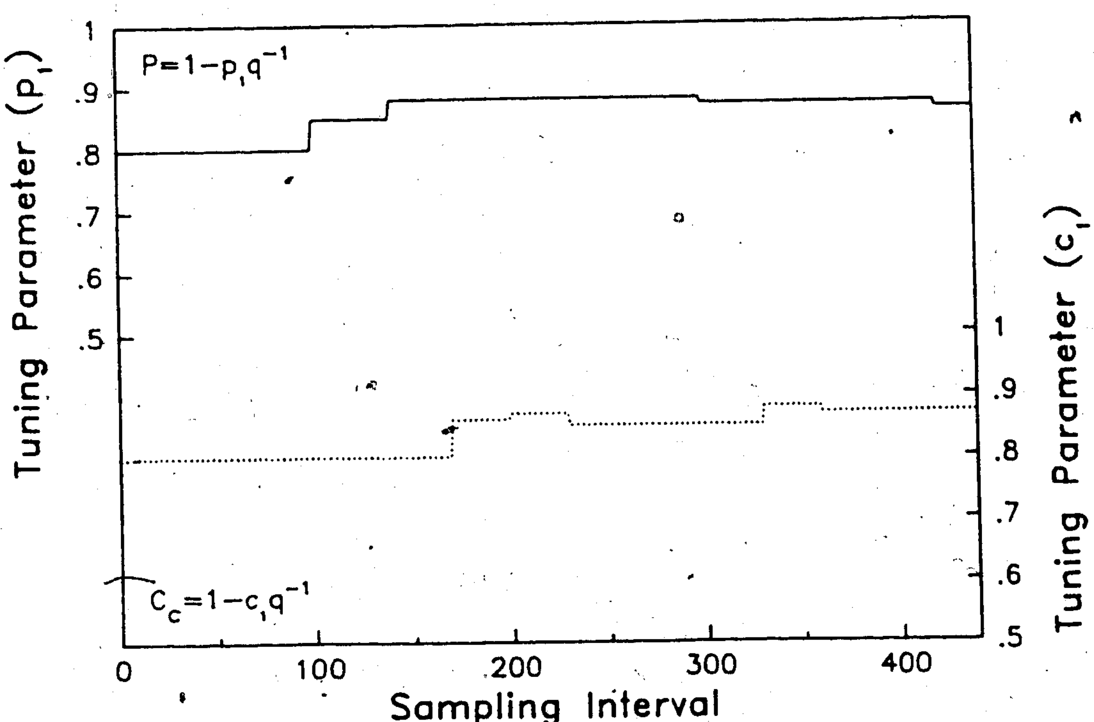


Figure 10.5d Controller Tuning Parameter Trajectories

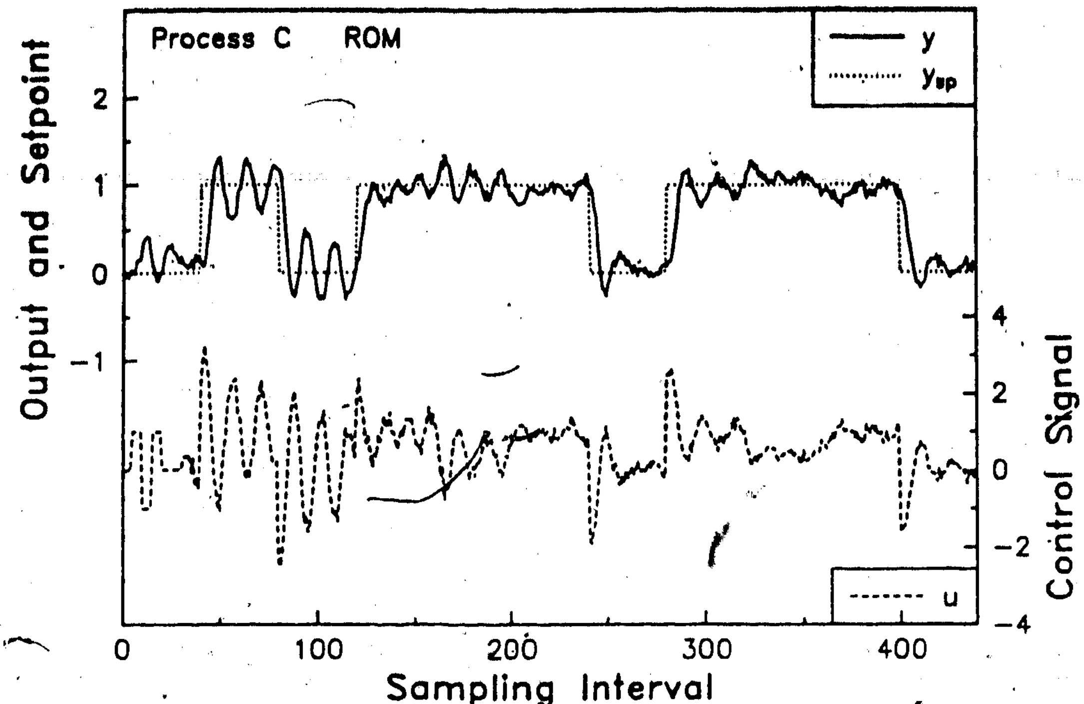


Figure 10.6a Performance Adaptive Control based on Pole Placement Algorithm

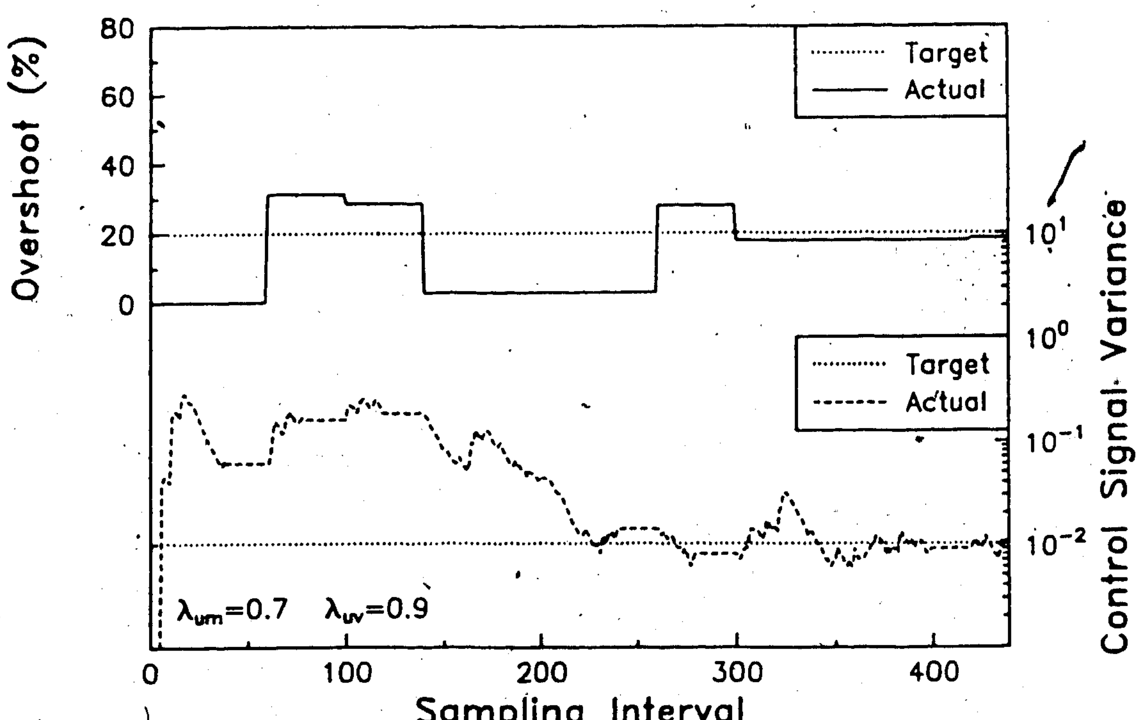


Figure 10.6b Actual and Specified Closed Loop Performance (Minimize Response Time with Overshoot Constraint)

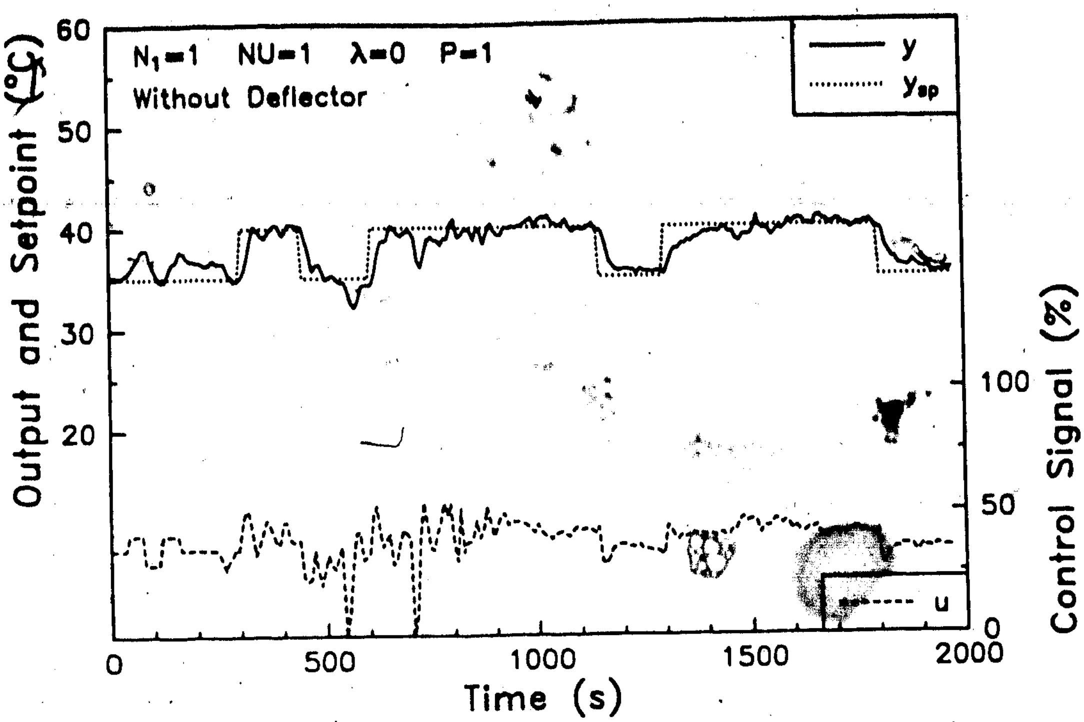


Figure 10.7a Performance Adaptive Generalized Predictive Control of the Stirred Tank Heater

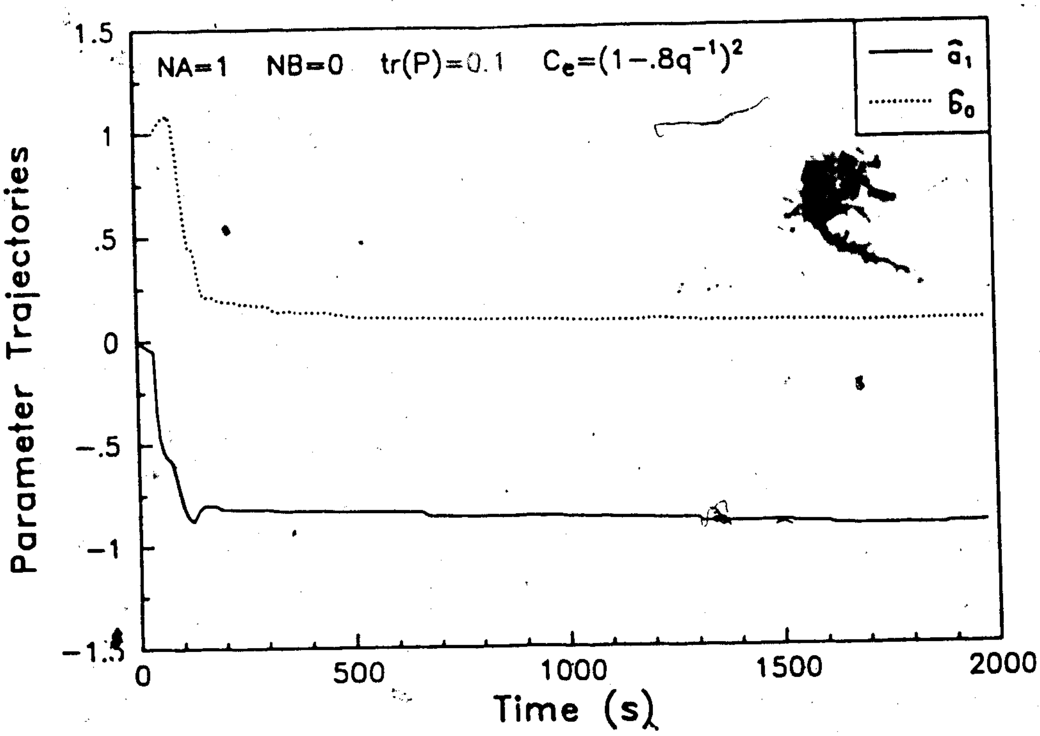


Figure 10.7b Model Parameter Estimates

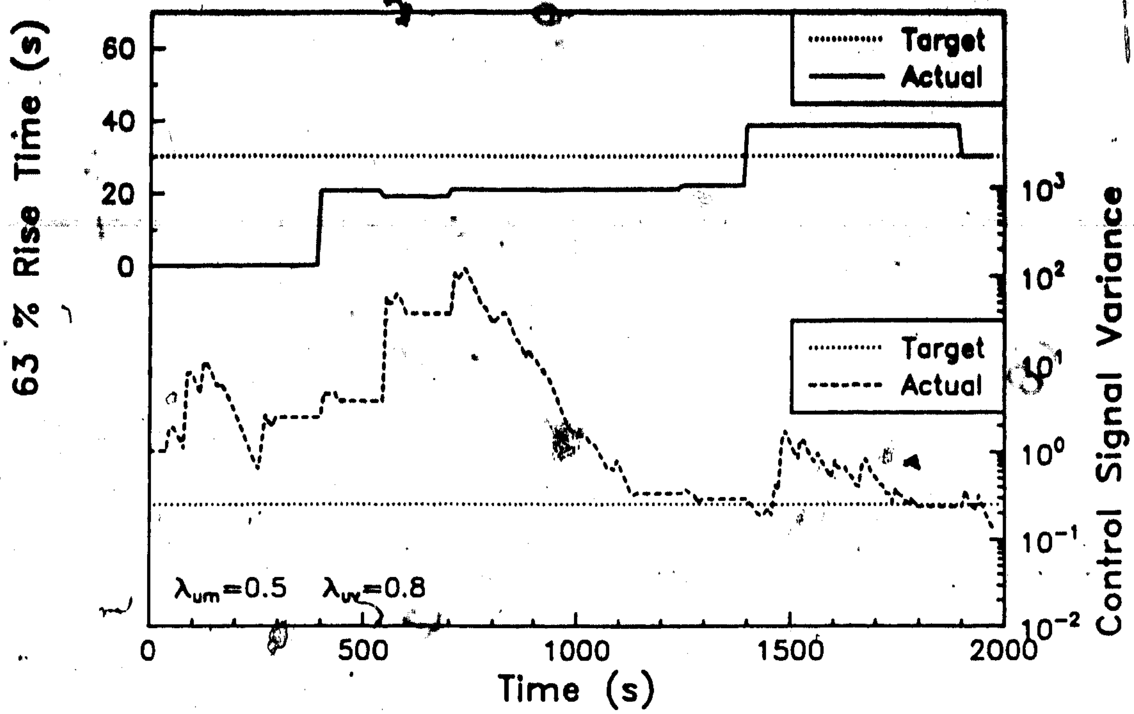


Figure 10.7c Actual and Specified Closed Loop Performance (Servo Objective: Desired 63% Rise Time)

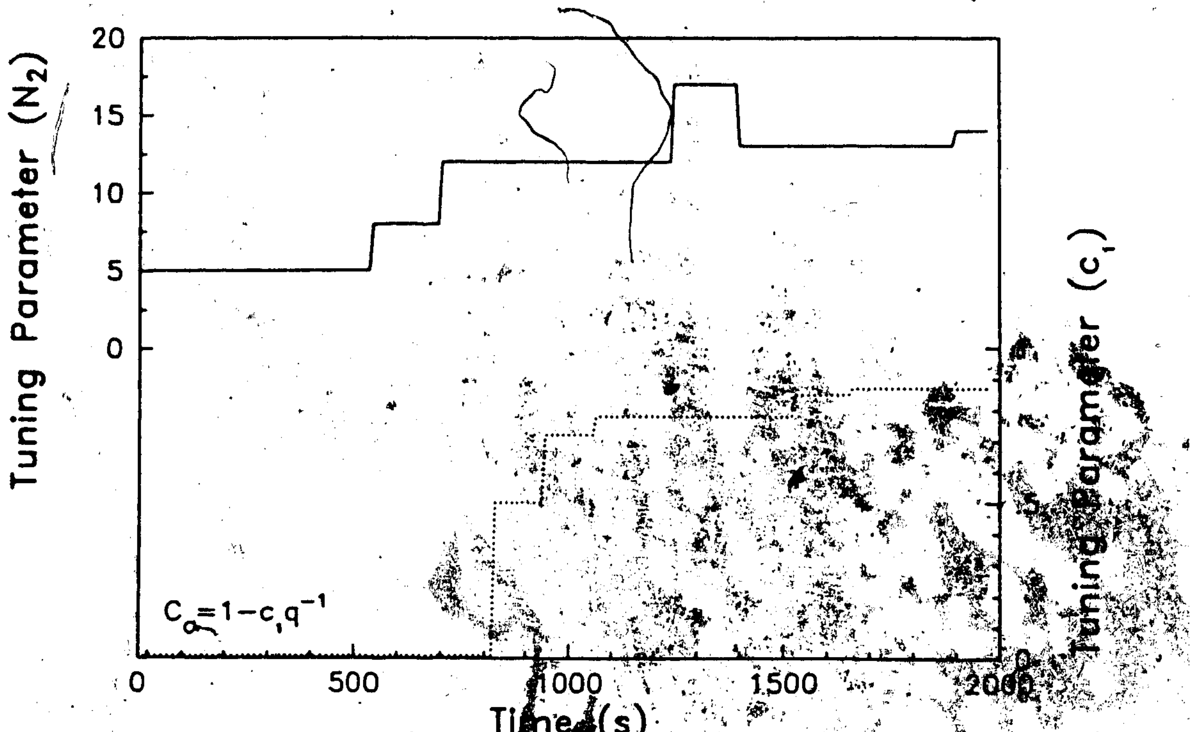


Figure 10.7d Controller Tuning Parameter Trajectories

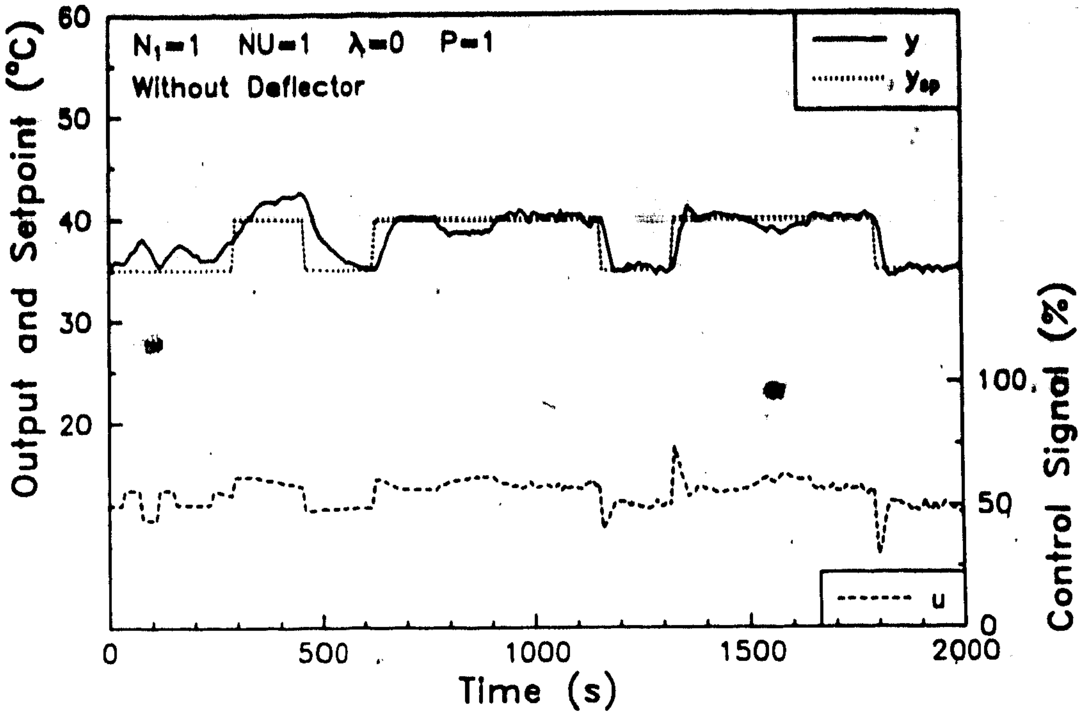


Figure 10.8a Performance Adaptive Generalized Predictive Control of the Stirred Tank Heater

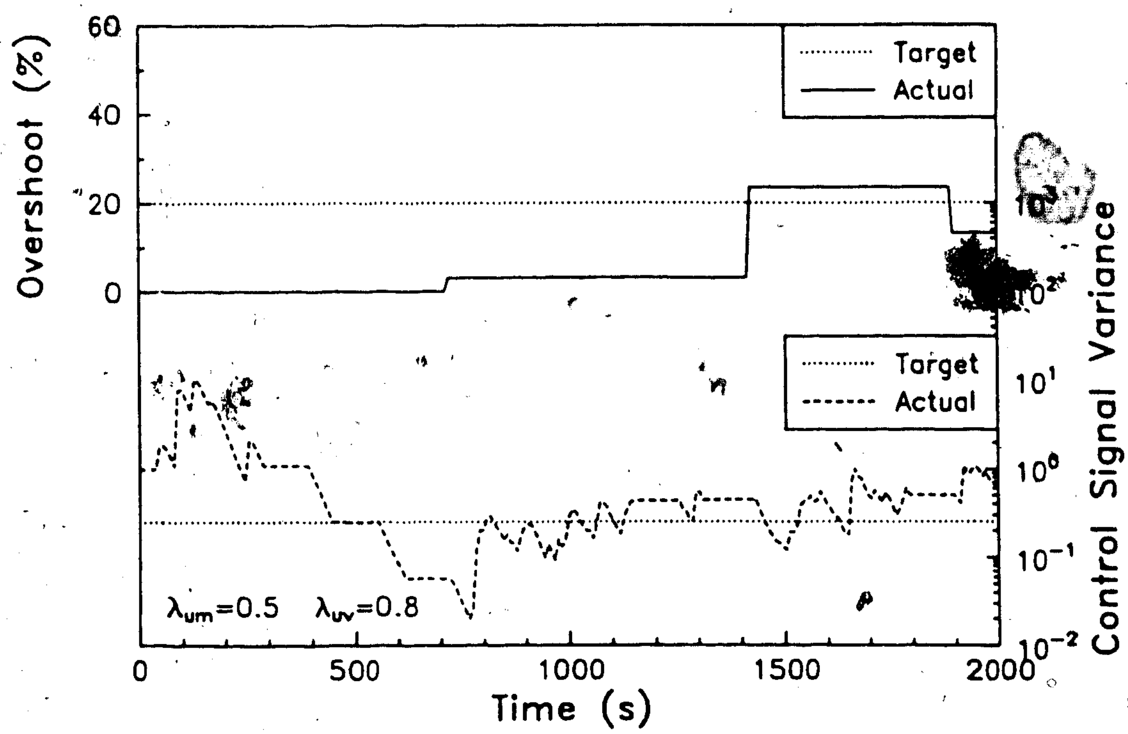


Figure 10.8b Actual and Specified Closed Loop Performance (Minimize Response Time with Overshoot Constraint)

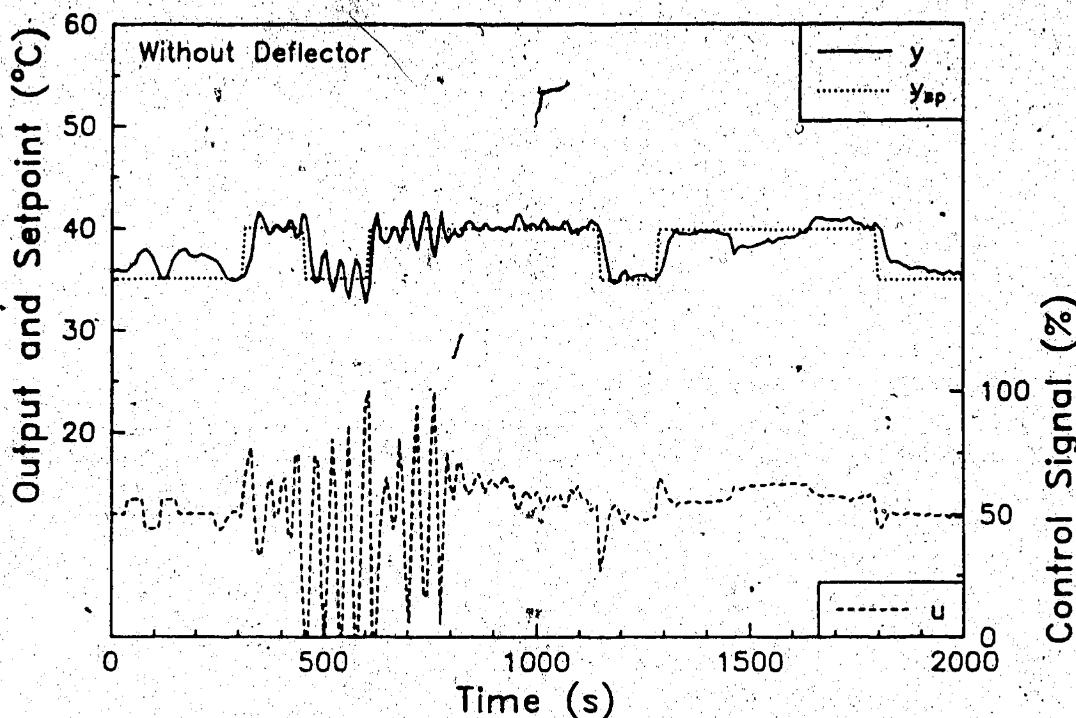


Figure 10.9a Performance Adaptive Pole Placement Control of the Stirred Tank Heater

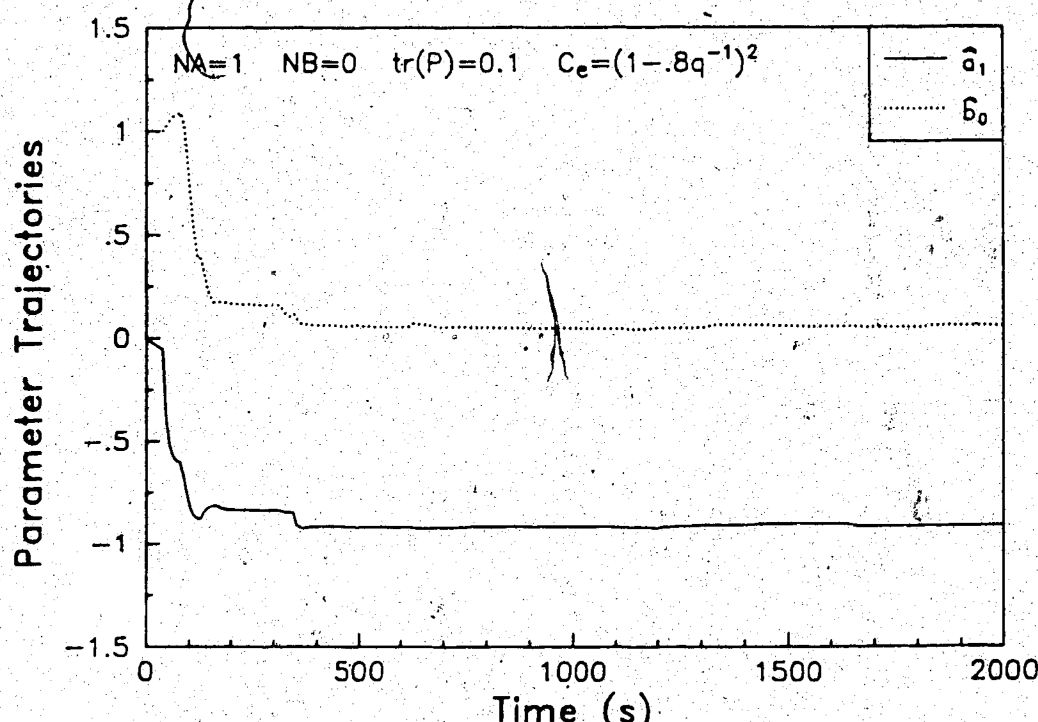


Figure 10.9b Model Parameter Estimates

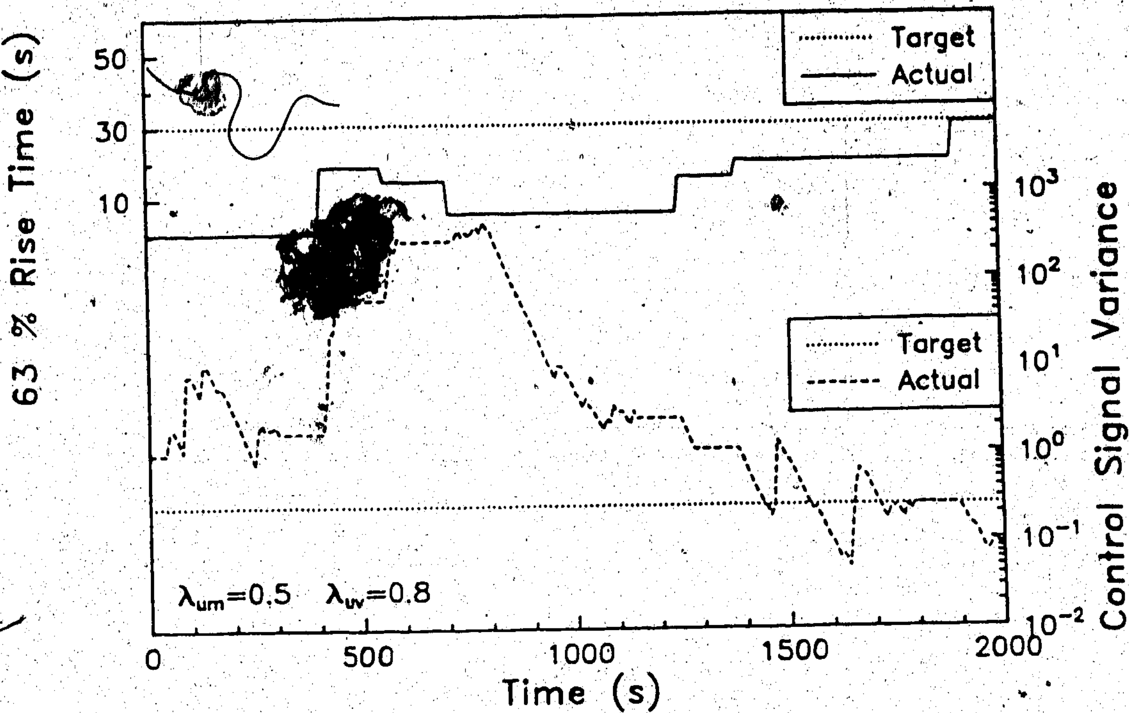


Figure 10.9c Actual and Specified Closed Loop Performance (Servo Objective: Desired 63% Rise Time)

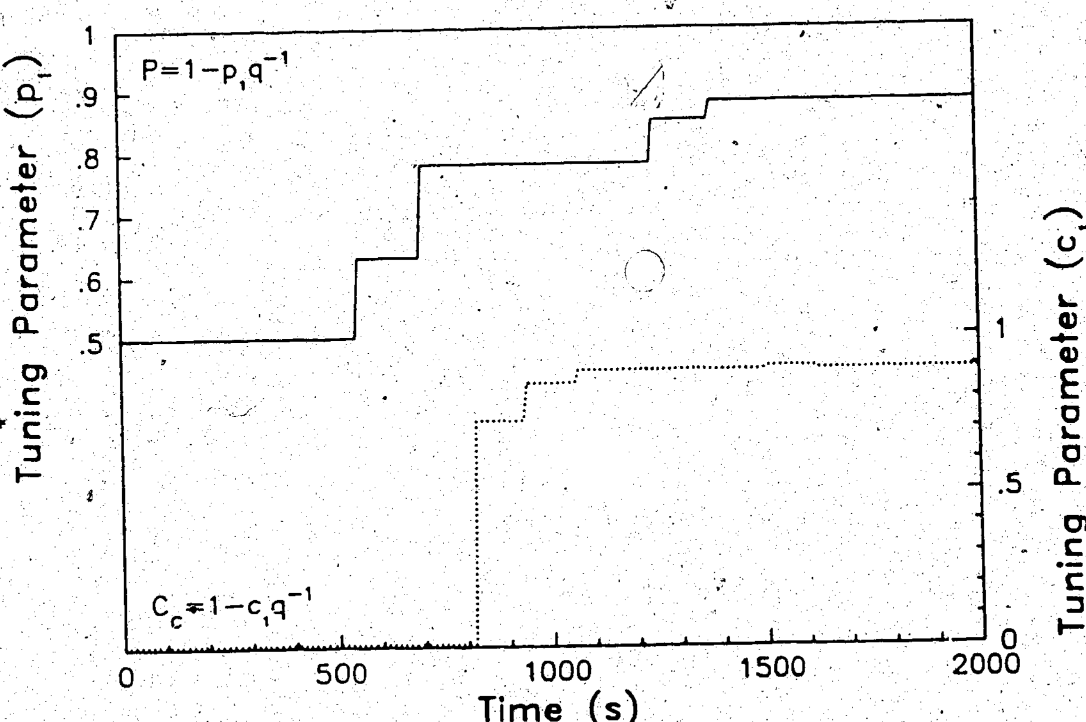


Figure 10.9d Controller Tuning Parameter Trajectories

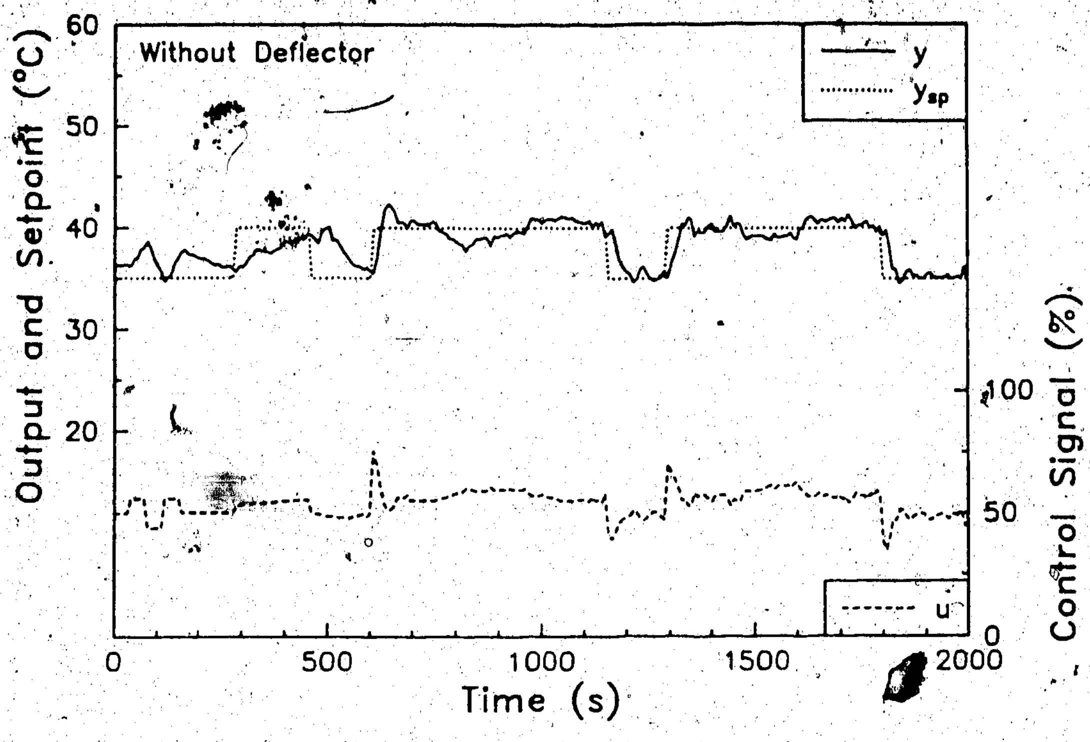


Figure 10.10a Performance Adaptive Pole Placement Control of the Stirred Tank Heater

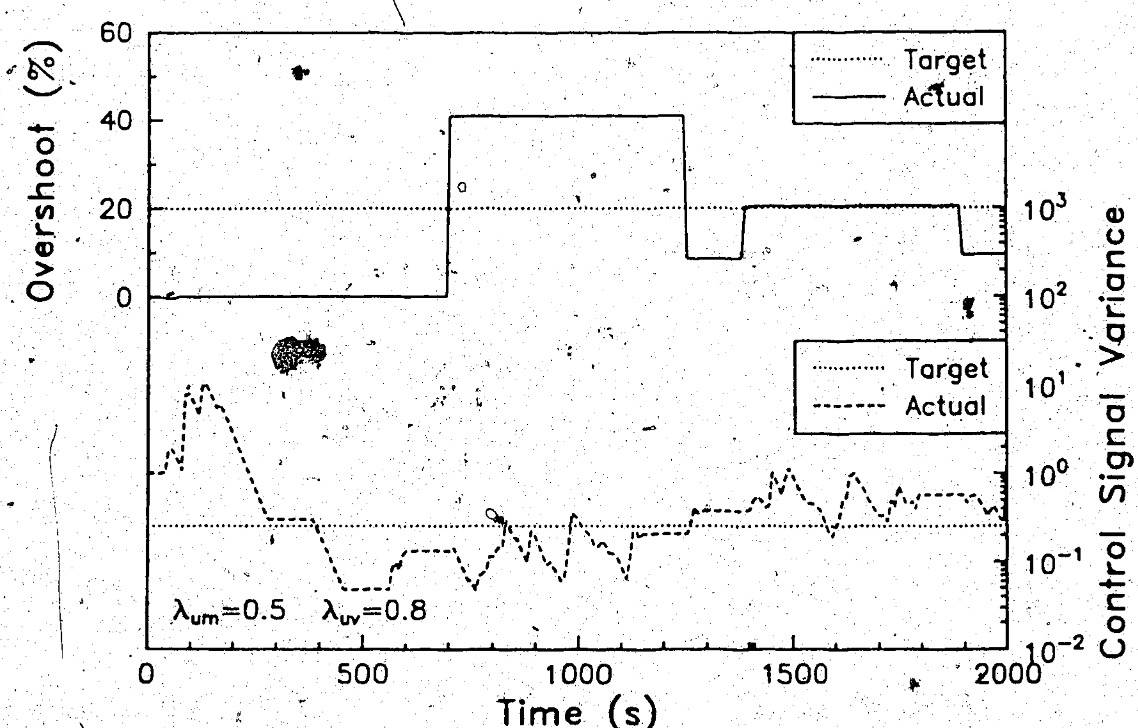


Figure 10.10b Actual and Specified Closed Loop Performance (Minimize Response Time with Overshoot Constraint)

11. CONCLUSIONS AND RECOMMENDATIONS

Self-tuning Generalized Predictive and Pole Placement controllers are capable of effectively controlling simple overdamped plants, as well as complex processes which are simultaneously nonminimum phase and open-loop unstable with variable dead-time. The assumption of a CARIMA process model gives rise to integral action required for the elimination of offset due to nonstationary disturbances frequently encountered in the chemical process industries. The Generalized Predictive Controller reduces to numerous well-known algorithms (e.g. Deadbeat, Generalized Minimum Variance, Extended Horizon, etc.) for particular settings of its design and tuning parameters. More importantly, for different settings of these parameters, the GPC algorithm overcomes many of the limitations of these earlier methods and yields much improved performance. If restrictions are placed on the form of the CARIMA model, the GPC algorithm can be made structurally equivalent to a PID/PI controller. This opens up the possibility of developing self-tuning PID controllers based on the minimization of a multistep cost function.

11.1 Controller Tuning

The large number of tuning parameters associated with the Generalized Predictive Controller gives the algorithm great flexibility but at the same time makes implementation more difficult. Strong interactions between the parameters implies that the most practical tuning strategies are those that assign constant values to reduce the number of active tuning parameters while still retaining the ablity to control a broad range of processes. The three configurations of GPC devised enable the user to vary the closed-loop response over a wide range by adjusting a single tuning parameter. A root locus analysis of GPC can be performed by deriving the closed-loop transfer function based on an equivalent general linear form of the control law. The following conclusions are also based on time-domain simulations and experimental runs.

1) As the active tuning parameter, N_2, of the Output Horizon configuration is increased from d+1 to ∞ the controller is progressively detuned. For an open-loop stable process, the closed-loop poles move from the open-loop zeros (minimum variance control) to the open-loop poles

(mean-level control). This configuration of GPC requires the least computational effort.

2) For the Lambda Weighting configuration, the control weighting, λ, is used to vary the speed of response. As λ is adjusted from 0 to ∞ the closed-loop poles move from the origin (deadbeat control) to the open-loop poles plus the point $z=1$ on the unit circle. This configuration is recommended for applications where strong control action cannot be tolerated. However, the response may be slightly oscillatory.

3) The inverse closed-loop model, $P(q^{-1})$ is the primary tuning parameter for the Detuned Model-Following configuration of GPC. The closed-loop poles will tend to be "close" to the user-specified roots of P. This strategy is recommended for applications where accurate following of a reference model is desired. The penalty incurred is a more active control signal.

Pole Placement can be obtained within the framework of GPC by specific settings of the tuning parameters. Alternatively, the "traditional" formulation where the coefficients of the controller parameters are solved for directly may be used. In either case, the PP controller cannot be used when both the numerator and denominator of the model are overparameterized. The sensitivity of the approach to numerical ill-conditioning due to near common factors in the model was observed during experimental runs with a set of Interacting Tanks. For this reason, the three configurations of GPC previously evaluated are recommended over the Pole Placement technique.

11.2 Maintenance of Performance

For the majority of industrial applications, the output behavior should remain consistent despite variations in the process. If fixed control weighting is employed, the output response may be seriously degraded if there are large changes in the process, even if the model is accurate at all times. However, if λ is proportional to $[B(1)]^2$ or if the Output Horizon, Detuned Model-Following or Pole Placement configurations are utilized, the closed-loop response will be invariant of process gain changes and relatively unaffected by changing dynamics. This was demonstrated for adaptive control of the nonlinear Interacting Tanks.

11.3 Robustness to Model-Plant Mismatch

In realistic applications, the identified model will never be a true representation of the process. Therefore, it is crucial that the controller be insensitive to model-plant mismatch. A controller design polynomial, $C_c(q^{-1})$ representing prior knowledge of the disturbances affecting a plant, should be used to provide robustness to this MPM and tailor the rejection of disturbances. It was found that $C_c(q^{-1})$ was essential for satisfactory control of the pilot-scale Stirred Tank Heater. For GPC, C_c determines how residuals (disturbances plus the effects of modelling errors) influence the predicted future output trajectory. Thus, in the absence of MPM, the setpoint tracking properties of the closed-loop are independent of the C_c polynomial. This controller design polynomial gives both GPC and PP controllers "two degrees of freedom" to meet independent servo and regulatory objectives.

11.4 Low Frequency Model Identification

An estimator filter, $C_e(q^{-1})$ should be used to focus the parameter estimation algorithm on frequencies around the closed-loop bandwidth. This filter allows identification of a model which is suitable for both GPC (i.e. the model gives accurate long-range predictions) and PP, inspite of nonzero mean disturbances, unmodelled dynamics and measurement noise. Bandpass filtering of the regressor using $C_e(q^{-1})$ was vital for stable control of the Stirred Tank Heater.

An Extended Least Squares algorithm gave unbiased model parameter estimates ($\hat{A}(q^{-1})$ and $\hat{B}(q^{-1})$) in the presence of various "colored" disturbances, but the model identified did not provide a good low frequency fit when there were unmodelled dynamics. Very poor control performance was observed when the estimate of the noise model, $\hat{C}(q^{-1})$, was used in the control law. The use of a user-specified design polynomial, $C_c(q^{-1})$, in the control calculation is the recommended approach.

11.5 Performance Tuning

With model-plant mismatch, no guarantee can be given that the output will respond according to expectations. In addition, the appropriate settings of the active tuning parameter and the C_c polynomial yielding desired behavior are not known *a priori*. A performance supervisor was developed to monitor the response of the closed-loop system and adjust both the active tuning parameter (for the particular configuration of GPC implemented) and $C_c(q^{-1})$ to achieve and maintain user-specified servo and regulatory performance. The performance tuner is only one element of a proposed hierarchical supervisory system consisting of a global off-line level and a local real-time level. The performance supervisor carries out a local optimization of the controller parameters once the structural parameters (e.g. model order, sample time, etc.) have been selected and a reasonable model identified. Experimental trials with the Stirred Tank Heater demonstrated the ability of the performance supervisor to meet overshoot constraints or give a desired closed-loop time constant for setpoint changes while keeping the regulatory control signal variance at a user-specified target value.

11.6 General Comments

One of the objectives of this work was to provide strategies and guidelines to simplify the task of selecting the controller tuning parameters to yield desired performance. The selection of parameters for a realistic parameter estimation algorithm was not the prime focus of this work. However, based on experience with the constant trace RLS routine used throughout this thesis, the selection of estimator parameters (e.g. the trace of the covariance matrix) is much more difficult than the selection of the controller parameters. Identification remains the "weak link" in adaptive control.

11.7 Future Work

The control algorithms studied in this thesis represent the current state-of-the-art in self-tuning or adaptive technology. While it has been demonstrated that these strategies can provide improved control performance, in order for adaptive techniques to gain widespread acceptance in industry they must be well understood, easy to implement and maintenance free. Much progress has been made through realistic experimental trials, but a number of areas [illegible]t for future work:

1) The self-tuning PI and PID controllers designed based on the Generalized Predictive Control and Pole Placement algorithms require detailed evaluation and comparison with other similar strategies reported in the literature. In particular the PP "PID on SP" controller derived based on the assumption that $C(q^{-1})=A(q^{-1})$ should be quite robust to model-plant mismatch and measurement noise (see Chapter 6). The possibility of identifying several B parameters and replacing b_o by $\sum b_i$ in the equations used to calculate the PID controller gains, in order to handle small time delays, should be investigated.

2) The Extended Least Squares algorithm yields an unbiased model which, unfortunately, may not match the process at low frequencies. In general, the opposite is true for Recursive Least Squares with (bandpass) filtering of the regressor. The potential exists to obtain an unbiased low frequency model by combining filtering with the ELS approach. Further work should also be directed toward developing better guidlines for specification of the estimator filter, $C_e(q^{-1})$.

3) Only a prototype of the performance tuner module of the local real-time supervisor was developed and tested in this thesis. This module requires refinement and the remaining modules in the supervisory system require development in order to produce a complete adaptive control package suitable for industrial applications.

4) The formulation of the Generalized Predictive Controller has been extended to the multivariable case (Mohtadi, et al., 1986) and work is underway to incorporate constraints using linear or quadratic programming. The question as to whether the tuning strategies which have been proposed for the SISO GPC algorithm are applicable to the MIMO case should be answered.

In very general terms, good control performance can be obtained using either GPC or PP as long as the model is a "reasonable" representation of the process. The challenge for the future is to develop equally easy to use identification routines which track process variations and provide an accurate model for extended periods of time.

12. REFERENCES

Allidina, A.Y. and Hughes, F.M. (1980) "Generalized Self-Tuning Controller with Pole Assignment", *Proc IEE*, Vol. 127, Pt. D, No. 1, pp. 13-18.

Arzen, K. (1986) "Use of Expert Systems in Closed Loop Feedback Control", *ACC*, Seattle, WA, pp. 140-145.

Åström, K.J. (1985) "Auto-tuning, Adaptation and Expert Control", *ACC*, Boston, MA, pp. 1514-1519.

Åström, K.J., Hagander, P. and Sternby, J. (1984) "Zeros of Sampled Systems", *Automatica*, Vol. 20, No. 1, pp. 31-38.

Åström, K.J. and Wittenmark, B. (1973) "On Self Tuning Regulators", *Automatica*, Vol. 9, No. 2, pp. 185-199.

Åström, K.J. and Wittenmark, B. (1980) "Self-Tuning Controllers based on Pole-Zero Placement", *Proc. IEE*, Vol. 127, Part D, No. 3, pp. 120-130.

Åström, K.J. and Wittenmark, B. (1984) Computer Controlled Systems - Theory and Design, Prentice-Hall Inc., Englewood Cliffs, N.J.

Belanger, P.R. (1983) "On Type 1 Systems and the Clarke-Gawthrop Regulator", *Automatica*, Vol. 19, No. 1, pp. 91-94.

Bristol, E.H. and Kraus, T.W. (1984) "Life with Pattern Adaptation", *ACC*, San Diego, CA. Vol. 2, pp. 888-892.

Bruijn, P.M., Bootsma, L.J. and Verbruggen, H.B. (1980) "Predictive Control using Impulse Response Models", *IFAC Symposium on Digital Computer Applications to Process Control*, Dusseldorf.

Clarke, D.W. (1982) "Model Following and Pole-Placement Self-Tuners", *Optimal Control Applications and Methods*, Vol. 3, pp. 323-335.

Clarke, D.W. (1983) "The Application of Self-Tuning Control", *Trans. Inst. of Meas. and Control*, Vol. 5, No. 2, pp. 59-69.

Clarke, D.W. (1988) "Self-Tuning Multistep Optimisation Controllers", *Workshop on Adaptive Control Strategies for Industrial Use*, Kananaskis, Alberta, June 20-22, pp. 1-25.

Clarke, D.W. and Gawthrop, P.J. (1975) "Self-Tuning Controller", *Proc. IEE*, Vol. 122, No. 9, pp. 929-934.

Clarke, D.W. and Gawthrop, P.J. (1979) "Self-Tuning Control", *Proc. IEE*, Vol. 126, No. 6, pp. 633-640.

Clarke, D.W. and Mohtadi, C. (1987) "Properties of Generalized Predictive Control", *IFAC 10th World Congress on Automatic Control*, Munich. W.G., Vol. 10, pp. 63-74.

Clarke, D.W., Mohtadi, C. and Tuffs, P.S. (1987a) "Generalized Predictive Control - Part I: The Basic Algorithm", *Automatica*, Vol. 23, No. 2, pp. 137-148.

Clarke, D.W., Mohtadi, C. and Tuffs, P.S. (1987b) "Generalized Predictive Control - Part II: Extensions and Interpretations", *Automatica*, Vol. 23, No. 2, pp. 149-160.

Clarke, D.W., Tuffs, P.S. and Mohtadi, C. (1985) "Self-Tuning Control of a Difficult Process", *IFAC Identification and System Parameter Estimation*, York, U.K.

Clarke, D.W. and Zhang, L. (1987) "Long-Range Predictive Control Using Weighting-Sequence Models", *Proc. IEE*, Vol. 134, Pt. D, No. 3, pp. 187-195.

Cluett, W.R., Shah, S.L. and Fisher, D.G. (1987) "Robust Design of Adaptive Control Systems using Conic Sector Theory", *Automatica*, Vol. 23, pp. 221-224.

Cutler, C.R. and Ramaker, B.L. (1980) "Dynamic Matrix Control - A Computer Control Algorithm", *JACC*, San Francisco.

DeKeyser, R.M.C. and Van Cauwenberghe, A.R. (1982) "Typical Application Possibilities for Self-Tuning Predictive Control", *IFAC Symp. Ident. Syst. Param. Est.*, Washington, D.C. pp. 975-980.

Foley, M. (1988) "Adaptive Control using a Kalman Filter", MSc. Thesis, University of Alberta.

Garcia, C.E. and Morshedi, A.M. (1986) "Quadratic Programming Solution of Dynamic Matrix Control (QDMC)", *Chem. Eng. Commun.*, Vol. 46, pp. 73-87.

Gawthrop, P.J. (1977) "Some Interpretations of the Self-Tuning Controller", *Proc. IEE*, Vol. 124, No. 10, pp. 889-894.

Gawthrop, P.J. (1986) "Self-Tuning PID Controllers: Algorithms and Implementation", *IEEE Trans. on Aut. Control*, Vol AC-31, No. 3, pp. 201-209.

Gertler, J. and Chang, H. (1986) "An Instability Indicator for Expert Control", *IEEE Control Systems*, Vol. 6, No. 4, pp. 14-17.

Harris, T.J. MacGregor, J.F. and Wright, J.D. (1982) "An Overview of Discrete Stochastic Controllers: Generalized PID Algorithms with Dead-Time Compensation", *Can. J. of Chem. Eng.*, Vol. 60, pp. 425-432.

Isermann, R. (1981) Digital Control Systems, Springer-Verlag, Berlin, W.G.

Isermann, R. and Lachmann, K.H. (1985) "Parameter-adaptive Control with Configuration Aids and Supervision Functions", *Automatica*, Vol. 21, No. 6, pp. 625-638.

Jiang, J. (1987) "Performance Monitoring in Expert Control Systems", *IFAC 10th World Congress on Automatic Control*, Munich, W.G., Vol. 6, pp. 303-307.

Kalman, R.E. (1958) "Design of a Self-Optimizing Control System", *Trans. ASME*, Vol. 80, pp. 468-478.

Kraus, T.W. and Myron, T.J. (1984) "Self-Tuning PID Controller Uses Pattern Recognition Approach", *Control Engineering*, Vol. 31, No. 6, pp. 106-111.

Kucera, V. (1979) Discrete Linear Control - The Polynomial Equation Approach, John Wiley, Toronto.

Lam, K.P. (1980) "Implicit and Explicit Self-Tuning Controllers", D.Phil Thesis, Oxford University (OUEL Report No. 1334/80).

Lambert, E.P. (1987) "Process Control Applications of Long-Range Prediction", OUEL Report No. 1715/87, University of Oxford.

Lambert, M. (1987) "Adaptive Control of Flexible Systems", OUEL Report No. 1707/87, University of Oxford.

Lawson, C.L. and Hanson, R.J. (1974) Solving Least-Squares Problems, Prentice-Hall Inc., Englewood Cliffs, N.J.

Lelic, M.A. and Wellstead, P.E. (1987) "Generalized Pole-Placement Self-Tuning Controller Part 2. Application to Robot Manipulator Control", *Int. J. Control*, Vol. 46, No. 2, pp. 569-601.

Lelic, M.A. and Zarrop, M.B. (1987) "Generalized Pole-Placement Self-Tuning Controller Part 1. Basic Algorithm", *Int. J. Control*, Vol. 46, No. 2, pp. 547-568.

Lim, K. (1988) "Multivariable Optimal Constrained Control Algorithm (MOCCA)" MSc. Thesis, University of Alberta.

Liu, K. and Gertler, J. (1987a) "On-Line Stabilization of Adaptive Controllers by Detuning in a Supervisory Framework", *ACC*, Minneapolis, MN, pp. 194-200.

Liu, K. and Gertler, J. (1987b) "A Supervisory (Expert) Adaptive Control Scheme", *IFAC 10th World Congress on Automatic Control*, Munich, W.G., Vol. 6, pp. 375-380.

Ljung, L. and Söderström, T. (1983) Theory and Practice of Recursive Identification, MIT Press, Cambridge, MA.

Lozano-Leal, R. and Goodwin, G.C. (1985) "A Globally Convergent Adaptive Pole Placement Algorithm Without a Persistency of Excitation Requirement", *IEEE Trans. on Aut. Control*, Vol. AC-30, No. 8, pp. 795-798.

Man, D. (1984) "Process Control Using Single Series Forecasting", MSc. Thesis, University of Alberta.

Maurath, P.R., Mellichamp, D.A. and Seborg, D.E. (1985) "Predictive Controller Design for SISO Systems", *ACC*, Boston, MA, Vol. 3, pp. 1546-1552.

McDermott, P.E. and Mellichamp, D.A. (1984) "An Auto-Pole-Placement Self-Tuning Controller", *Int. J. Control*, Vol. 40, No. 6, pp. 1131-1147.

McIntosh, A.R. and Yegneswaran, P.K. (1986) "Real-Time Multitasking Kalman Filter with Continuous Graphics Support", Chemical Engineering Internal Department Report, University of Alberta.

Mellichamp, D.A. (ed.) (1983) "Real-Time Computing with Applications to Data Acquistion and Control", Van Nostrand Reinhold, New York.

Middleton, R.H., Goodwin, G.C., Hill, D.J. and Mayne, D.Q. (1988) "Design Issues in Adaptive Control", *IEEE Trans. on Aut. Control*, Vol. 33, No. 1, pp. 50-58.

Minter, B.J. and Fisher, D.G. (1988) "A Comparison of Adaptive Controllers: Academic vs Industrial", *ACC*, Atlanta, Vol. 2, pp. 1653-1658.

Mohtadi, C. (1986) "Studies in Advanced Self-Tuning Algorithms", D. Phil. Thesis, Oxford University.

Mohtadi, C. (1988) "On the Role of Prefiltering in Parameter Estimatin and Control" *Workshop on Adaptive Control Strategies for Industrial Use*, Kananaskis, Alberta, June 20-22, pp. 261-282.

Mohtadi, C. and Clarke, D.W. (1986) "Generalized Predictive Control, LQ or Pole Placement: A Unified Approach", *Proc. 25th CDC*, Athens, Greece, pp. 1536-1541.

Mohtadi, C., Shah, S.L., and Clarke, D.W. (1986) "Generalized Predictive Control of Multivariable Systems", *OUEL Report* No. 1640/86, University of Oxford, also in *Proc. of 5th Yale Workshop in Adaptive Systems Theory* (May 20-22,1987), pp. 54-59.

Morris, A.J., Nazer, Y. and Wood, R.K. (1981) "Single and Multivariable Applications of Self-Tuning Controllers" in Self-Tuning and Adaptive Control, (Harris and Billings, ed.) Perter Perigrinus Ltd., Stevenage, U.K.

Mosca, E., Zappa, G. and Manfredi, C. (1984) "Multistep Horizon Self-Tuning Controllers: The MUSMAR Approach", *IFAC 9th World Congress on Automatic Control*, Budapest, Vol. 2, pp. 935-939.

Myron, T.J. (1986) "Self-tuning PID Control--An Expert System Approach", *IFAC Auto. Control in Petroleum, Petrochemical and Desalination Industries*, Kuwait, pp. 77-80.

Navratil, J.P., Lim, K.Y. and Fisher, D.G. (1988) "Disturbance Feedback in Model Predictive Control Systems", *IFAC International Workshop on Model Based Process Control*, Atlanta.

Neuman, C.P. and Baradello, C.S. (1979) "Digital Transfer Functions for Microcomputer Control", *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. SMC-9, No. 12, pp. 856-860.

Peterka, V. (1984) "Predictor-based Self-Tuning Control", *Automatica*, Vol. 20, No. 1, pp. 39-50.

Pollard, J.F. and Brosilow, C.B. (1986), "Adaptive Inferential Control", *ACC*, Seattle, WA, pp. 701-706.

Richalet, J., Rault, A., Testud, J.L. and Papon, J. (1978) "Model Predictive Heuristic Control: Appplications to Industrial Processes", *Automatica*, Vol. 14, No. 5, pp. 413-428.

Rohrs, C.E., Athans, M., Valavani, L. and Stein, G. (1984) "Some Design Guidelines for Discrete-time Adaptive Controllers", *Automatica*, Vol. 20, No. 5, pp. 653-660.

Sanoff, S.P. and Wellstead, P.E. (1985) "Expert Identification and Control", *IFAC Ident. and Syst. Param. Est.*, York, U.K. pp. 1273-1278.

Schumann, R., Lachmann, K.H. and Isermann, R. (1981) "Towards Applicability of Parameter- Adaptive Control Algorithms", *IFAC 8th World Congress on Automatic Control*, Kyoto, Japan, Vol. 2, pp. 903-910.

Seborg, D.E., Edgar, T.F. and Shah, S.L. (1986) "Adaptive Control Strategies for Process Control: A Survey", *A.I.Ch.E. Journal*, Vol. 32, No. 6, pp. 881-913.

Sela, R. and Ray, W.H. (1986) "Toward the Development of Prudent and Robust Self-Tuning Adaptive Controllers for Process Control", *ACC*, Seattle, WA., pp. 17-24.

Shah, S.L. and Cluett, W.R. (1987) "RLS Based Estimation Schemes for Self-Tuning Control", *1987 Annual A.I.Ch.E. Meeting*, New York, N.Y.

Smith, O.J.M. (1957) "Closer Control of Loops with Dead Time", *Chem. Eng. Prog.*, Vol. 53, No. 5, pp. 217-219.

Song, H.K., Fisher, D.G. and Shah, S.L. (1984) "Experimental Evaluation of a Robust Self-Tuning PID Controller", *Can. J. of Chem. Eng.*, Vol. 62, pp. 755-763.

Sripada, N.R. and Fisher, D.G. (1985) "Multivariable Optimal Constrained Control Algorithm. Part 1: Formulation and Application", *Proc. Int. Conf. on Ind. Process Modeling and Control*, Hangzhou, China, Vol. 1.

Sripada, N.R. and Fisher, D.G. (1987) "Improved Least Squares Identification", *Int. J. Control*, Vol. 46, No. 6, pp. 1889-1913 also *ACC*, Minneapolis, MN, June 10-12,87, Vol. 3, pp. 2027-2037.

Stephanopoulos, [illegible] (1984) Chemical Process Control, Prentice-Hall Inc., New Jersey.

Stephanopoulos, G. and Huang, H. (1986) "The 2-Port Control System", *Chem. Eng. Science*, Vol. 41, No. 6, pp. 1611-1630.

Tjokro, S. and Shah, S.L. (1985) "Adaptive PID Control", *ACC*, Boston, MA., Vol. 3, pp. 1528-1534.

Tuffs, P.S. (1984) "Self-Tuning Control: Algorithms and Applications", D. Phil. Thesis, Oxford University (OUEL Report 1567/85).

Tuffs, P.S. and Clarke, D.W. (1985) "Self-Tuning Control of Offset: A Unified Approach", *Proc. IEE*, Vol. 132, Pt. D, No. 3, pp. 100-110.

Tzouanas, V. and Shah, S.L. (1985) "Adaptive Pole-Placement Control of A Batch Polymer Reactor", in IFAC Workshop on Adaptive Control of Chemical [illegible]ocesses (Unbehaven, ed.) Pergamon Press, Toronto, pp. 114-11[illegible]

Vermeer, P.J. (19[illegible]) "Design and Evaluation of Practical Self-Tuning PID Controllers", MSc. Thesis, University of Alberta.

Walgama, K. (1986) "Multivariable Adaptive Predictive Control of Stochastic Systems with Time Delays", MSc. Thesis, University of Alberta.

Wellstead, P.E., Prager, D. and Zanker, P. (1979) "Pole Assignment Self-Tuning Regulator", *Proc. IEE*, Vol. 126, No. 8, pp. 781-787.

Wellstead, P.E. and Sanoff, S.P. (1981) "Extended Self-Tuning Algorithm" *Int. J. Control*, Vol. 34, No. 3, pp. 433-455.

Wittenmark, B. (1988) "Adaptive Control: Implementation and Application Issues", *Workshop on Adaptive Control Strategies for Industrial Use*, Kananasis, Alberta, June 20-22, pp. 245-260.

Wittenmark, B and Åström, K.J. (1984) "Practical Issues in the Implementation of Self-Tuning Control", *Automatica*, Vol. 20, No. 5, pp. 595-605.

Ydstie, B.E., Kershenbaum, L.S. and Sargent, R.W.H. (1985) "Theory and Application of an Extended Horizon Self-Tuning Controller", *A.I.Ch.E. Journal*, Vol. 31, No. 11, pp. 1771-1780.

Zafiriou, E. and Morari, M. (1987) "Setpoint Tracking vs Disturbance Rejection for Stable and Unstable Processes", *ACC*, Minneapolis, MN., pp. 649-651.

Ziegler, J.G. and Nichols, N.B. (1942) "Optimum Settings for Automatic Controllers", *Trans. ASME*, Vol. 64, p. 759.

APPENDIX A STABILITY LEMMA FOR OUTPUT HORIZON GPC CONTROLLER

The proof of Theorem 1 is given in (Liu and Gertler, 1987a).

Theorem 1: Given a plant model $\hat{G}_p(q^{-1})$ with denominator $\hat{A}(q^{-1}) = 1 + \hat{a}_1 q^{-1} + \ldots + \hat{a}_{na} q^{-na}$ and poles α_i, $|\alpha_i|<1$, i=1,..,na. Given also a controller designed to yield a closed-loop transfer function $G_{CL}(q^{-1})$ with denominator $P_{CL}(q^{-1}) = \hat{A}(q^{-1}) + \Delta_{CL}(q^{-1})$ (where $\Delta_{CL}(q^{-1}) = \delta_1 q^{-1} + \ldots + \delta_h q^{-h}$) and poles ρ_j, $|\rho_j|<1$, j=1,..,h. For any stable non-integrating process $G_p(q^{-1})$ and model $\hat{G}_p(q^{-1})$ there is a limit value $\delta_\ell>0$, so that the closed-loop is guaranteed to be stable for $|\delta_j|<\delta_\ell$, j=1,..,h.

Based on this theorem consider Lemma 1.

Lemma 1: For the basic GPC algorithm with NU=1, N_1=1, λ=0, and setpoint w(t), and for any stable non-integrating process $G_p(q^{-1})$ and model $\hat{G}_p(q^{-1}) = q^{-1}\hat{B}(q^{-1})/\hat{A}(q^{-1})$, there exists a limit value $N_\ell<\infty$, such that the system is guaranteed to be stable if $N_2>N_\ell$.

The proof of this Lemma follows directly from the results of sections 2.3.1 and 2.5.1. In section 2.3.1, it was shown that GPC places the poles of the closed-loop system at the roots of:

$$\hat{T}\hat{A}\Delta + q^{-1}\hat{B}\hat{S} \tag{A.1}$$

In section 2.5.1, under the assumptions:

a) N_1=1, NU=1, λ=0
b) $\hat{A}(q^{-1})$ is a stable polynomial with all roots inside the unit circle
c) the setpoint sequence is equal to the actual setpoint (ie no preprogrammed setpoints are used)

it was demonstrated that the characteristic polynomial (A.1) may be written for large N_2 in the form:

$$\hat{A} + \left[\hat{A}/g_{N_2}\right]\left[(g_{N_2+1}-g_{N_2})q^{-1} + (g_{N_2+2}-g_{N_2+1})q^{-2} + \ldots\right] \tag{A.2}$$

where g_i is a step response coefficient for the model:

$$\hat{B}/\hat{A}\Delta = g_0 + g_1 q^{-1} + \dots + g_{N_2} q^{-N_2} + g_{N_2+1} q^{-(N_2+1)} + \dots \qquad (A.3)$$

Equation (A.2) indicates that GPC in this configuration places the poles of the closed-loop transfer function at the roots of

$$P_{CL}(q^{-1}) = \hat{A}(q^{-1}) + \Delta_{CL}(q^{-1}) \qquad (A.4)$$

where,

$$\Delta_{CL}(q^{-1}) = \left[\hat{A}/g_{N_2}\right]\left[(g_{N_2+1}-g_{N_2})q^{-1} + (g_{N_2+2}-g_{N_2+1})q^{-2} + \dots\right]$$

$$= \delta_1 q^{-1} + \delta_2 q^{-2} + \dots. \qquad (A.5)$$

The coefficents $\delta_j \to 0$, $j=1,\dots$ as $N_2 \to \infty$. Therefore, a limit value, N_ℓ, exists such that for $N_2 > N_\ell$, $|\delta_j| < \delta_\ell \ \forall j$. This completes the proof.

■