IMPLEMENTATION OF FEEDFORWARD AND MULTIVARIABLE SELF-TUNING CONTROLLERS IN CHEMICAL PROCESSES

BY

José Ricardo Pérez Correa

A thesis submitted for the degree of
Doctor of Philosophy of the
University of London

Department of Chemical Engineering and Chemical Technology
Imperial College of Science and Technology
London

ABSTRACT

A feedback SISO Robust Self-tuning Regulator was extended to include feedforward and multivariable control.

Several SISO and MIMO self-tuning control algorithms were proposed and tested in simulations and experiments in an absorption/desorption pilot plant. The algorithms were implemented in the pilot plant using an on-line control package called ACS (Advanced Control System).

The proposed controllers employ an extended-horizon minimum-variance control law, and a recursive least squares parameter estimator with variable forgetting factor.

In the particular case of the multivariable self-tuners, a bootstrap scheme estimates both the states and the parameters of an innovation state-space model. The control law in this case was extended to consider different time-horizons for each output.

Simulations and pilot plant experiments have shown that the proposed algorithms work well and they are robust against deterministic disturbances [measurable and unmeasurable], interaction, unknown and varying time-delays, some non-linearities and non-minimum phase systems.

The algorithms are of simple design and require few initial parameters. Among these, the most important are time-horizons, information lengths and model orders. The selection of these parameters has shown to be less easy in feedback/feedforward and multivariable algorithms than in SISO feedback controllers. Nevertheless, the inclusion of feedforward always improve the closed-loop performance. Similarly, MIMO self-tuners have shown much better performance, in interactive systems, than independent SISO self-tuners.

The proposed feedback/feedforward and multivariable controllers retain the detuning property and simplicity that characterize the SISO Robust Extended-horizon Self-tuning Regulator of Ydstie, (Ydstie et al, 1985)

Con mucho cariño para *Ali y Claudio*

ACKNOWLEGMENTS

I am very grateful to Dr. L.S. Kershenbaum for his supervision, patience and encouragement throughout this research. I would also like to thank Professor R.W.H. Sargent for his contribution in setting the objectives of this research.

The successful completion of this project would not have been possible without the active support and help of many friends and colleagues in the Control and Chemical Engineering Departments at Imperial College to whom I am indebted. In particular, I wish to thank G. Stuart for his valuable assisstance in computing, and my dear friends D. Brant, G. Lapidus and V. Limpkin for their valuable comments on grammar and style.

Finally, I acknowledge the financial support provided by The British Council and Odeplan.

"I do not know why, but I have never

seen a machine that, however perfect in

the philosophers' description, is

perfect in its mechanical functioning"

Umberto Eco

CONTENTS PAGE

CHAPTER 1 INTRODUCTION 8

CHAPTER 2 LITERATURE SURVEY 10

2.1 SISO Feedback Algorithms 14
2.2 SISO Feedforward Self-Tuning Regulators 22
2.3 Multivariable Self-Tuning Regulators 24

CHAPTER 3 THEORY 32

3.1 Polynomial Algorithms 32
3.1.1 A Robust Feedback Self-tuner 32
3.1.2 A Feedback-Feedforward Self-tuner 39

3.2 State Space Algorithms 44
3.2.1 The Single-Input Single-Output Case 44
3.2.2 The Multivariable Case 51
3.2.3 A Feedback-Feedforward Multivariable Self-tuner 59

CHAPTER 4 EXPERIMENTAL SYSTEM 63

4.1 The Process 63
4.2 The ACS Package 67
4.3 The User Interface 68

CHAPTER 5 RESULTS AND DISCUSSION 74

5.1 Polynomial Algorithms 74
5.1.1 Feedback Self-tuning Control 75
5.1.2 Feedback-Feedforward Self-tuning Control 96

5.2 Multivariable State-Space Algorithms 126
5.2.1 MIMO State-space Feedback Control 126
5.2.2 An Incremental Feedback State-space Self-tuner 150
5.2.3 MIMO State-space Feedback-feedforward Control 174
5.2.3.a Linear simulations 175
5.2.3.b Quasi-nonlinear simulations 197

CHAPTER 6 CONCLUSIONS 205

LIST OF REFERENCES 209

LIST OF APPENDICES

Appendix A Unexpected behaviour of the extended-horizon control 217

B State-space and input-output representation for a single-input single-output deterministic system 222

C Derivation of the measurement equation from the row companion multivariable state-space canonical form 226

D Noise Generation 233

E Examples of multivariable extended-horizon control 235

F Interaction Measure with Bristol array 243

G Controllability and observability of a state-space discrete-time model 246

H MIMO models used in simulations 248

I Alternative implementations of self-tuning regulators 252

LIST OF MAIN SYMBOLS

A, B,	output and input polynomials of the I/O model equation. Also, the process and control matrices in the state-space equation
C	disturbance polynomial. Also state-space observation matrix
D	reciprocal of the control-gain matrix. Also disturbance polynomial of the predictive I/O model
F	divisor polynomial
G	output polynomial of the predictive I/O model
H	input polyinomial of the predictive I/O model. Also disturbance matrix in the state-space model
K	Kalman gain matrix of the innovations state-space model
S	feedforward state-space gain matrix
T	time-horizon
Y_k	linear combination of known data at time k
a_i	output polynomial coefficients
b_i	input polynomial coefficients
c_i	disturbance polynomial coefficients
d	reciprocal of the controller gain
du	control delay
dw	disturbance delay
e_k	innovation at time k
$\underline{i}(j)$	unitary vector with the one in the position j
m	number of input/output pairs in a MIMO model
n	order of the process
n_j	observability index of the j subsystem
q	order of the disturbance polynomial
u_k	input control at time k
v_k	residual at time k
w_k	input disturbance at time k
y_k	output at time k
$\underline{x}_k$	state vector at time k
z_k	data vector in the state-space algorithm at time k
z^{-1}	delay operator

Δ	backward-difference operator
Σ_o	information length (content)
α_i	coefficients of the output polynomial of the predictive model
β_i	coefficients of the input polynomial of the predictive model
δ_i	coefficients of the disturbance polynomial of the predictive model
$\underline{\phi}_k$	data vector
γ_k	filter gain
$\underline{\theta}_k$	parameter estimates vector
$\hat{\bullet}$	predicted value
$\hat{\bullet}_{k/k-1}$	predicted value at time k given information at time k-1
$\overline{\bullet}$	terms of a positional algorithm
$\underline{\bullet}$	vectors

CHAPTER 1 INTRODUCTION

When Kalman proposed his self-tuning regulator in 1958, the primitive state of the theory and the cost and size of computers made any idea of implementation almost unpractical.

Nowadays, with low-cost microcomputers and an extremely sophisticated theory, implementing self-tuning regulators in real plants can still be a difficult problem.

All the self-tuning algorithms that have been proposed required the selection of a set of initial parameters. These for example are model order, time-delay, forgetting factor, ...etc. The proper selection of these parameters can require a great deal of experience since the self-tuning algorithms are, generally, very sensitive to them.

On the other hand, the theory can only ensure convergence and stability of the algorithms in extremely simple cases [linear, lumped parameters, time invariant].

Most of the research effort in self-tuning control is devoted to design robust algorithms which would not be sensitive to the selection of initial parameters. These algorithms should be robust enough so that they can be applied with confidence in a variety of systems.

The Chem. Eng. Dept. at I.C. have been involved in the development of robust self-tuning regulators to be used in the chemical industry. Chemical plants are characterized by systems with large transportation delays, low noise/signal ratio, non-linearities, interaction between control-loops and by the presence of [measurable and unmeasurable] deterministic disturbances.

Two major achievements have been obtained in this department: the variable forgetting factor concept, which very much simplifies the tuning of the parameter estimator; and the minimum-variance extended-horizon control-law, which improves the robustness of the control algorithm to the selection of initial settings and also to the presence of transportation delays.

So far, the majority of the results in the literature are related to SISO feedback controllers. The problems of interaction and deterministic disturbances have not received the same degree of attention.

It is the aim of this project to extend the sphere of interest of the department to include algorithms robust to [measurable and unmeasurable] deterministic disturbances and to interaction.

Chapter 2 presents a review of the self-tuning literature, covering the aspects of implementing self-tuners, feedforward and multivariable control.

In Chapter 3 several algorithms are proposed which consider the problems of deterministic disturbances and interaction. These algorithms include feedback/feedforward (FB/FF) SISO controllers and FB and FB/FF state-space multivariable controllers.

The experimental system is described in Chapter 4. It includes a diagram of the pilot-plant and a general description of the process and control-loops. There is also a basic description of the IBM/ACS control-package under which the algorithms have been tested.

Chapter 5 presents the results [both simulations and experiments] obtained when implementing the controllers proposed in Chapter 3. During the implementation, the original algorithms suffered some modifications which are described in Appendix I.

Conclusions and ideas for future work are given in Chapter 6.

CHAPTER 2 LITERATURE SURVEY

The problem of controlling a system with uncertain and/or varying parameters has attracted the attention of the control engineering community for many years.

The adaptive control theory is one of the most popular approaches to this problem, with both practitioners and theorists.

The term "adaptive control" is not understood in the same way by all authors. Some of them define an adaptive control system "as one that automatically adjusts the controller settings to accommodate changes in the process to be controlled or its environment" [Seborg, Edgar and Shah, 1986]. Others say that adaptive control "represents one particular approach to designing sub-optimal controllers for non-linear stochastic systems" [Jacobs, 1985].

Goodwin and Sin (1984) consider adaptive those control algorithms that combine an on-line parameter estimation technique with a standard control design for known systems.

This last definition will be used in this work since it clearly defines a series of algorithms that have in common the main advantages, difficulties and implementation problems despite the diversity in their design philosophies.

Among these algorithms, the model reference adaptive controllers and the self-tuning controllers are probably the most developed.

Most of the work done in this area has been concentrated on discrete-time algorithms, so continuous techniques will not be considered.

The model reference scheme [Landau, 1974], as shown in Fig 2.1, consists of a given reference model, an adaptation mechanism, and an adjustable controller.

The reference model gives a desired response. The adaptation mechanism modifies the parameters of the adjustable controller and sometimes generates an auxiliary test signal, in order to minimize a function of the difference between the output of the plant and the output of the reference model. The whole idea is to make the adjustable closed-loop transfer function between the set point y^* and the plant output y asymptotically approach the transfer function between the set point y^* and the output of the reference model y_r.

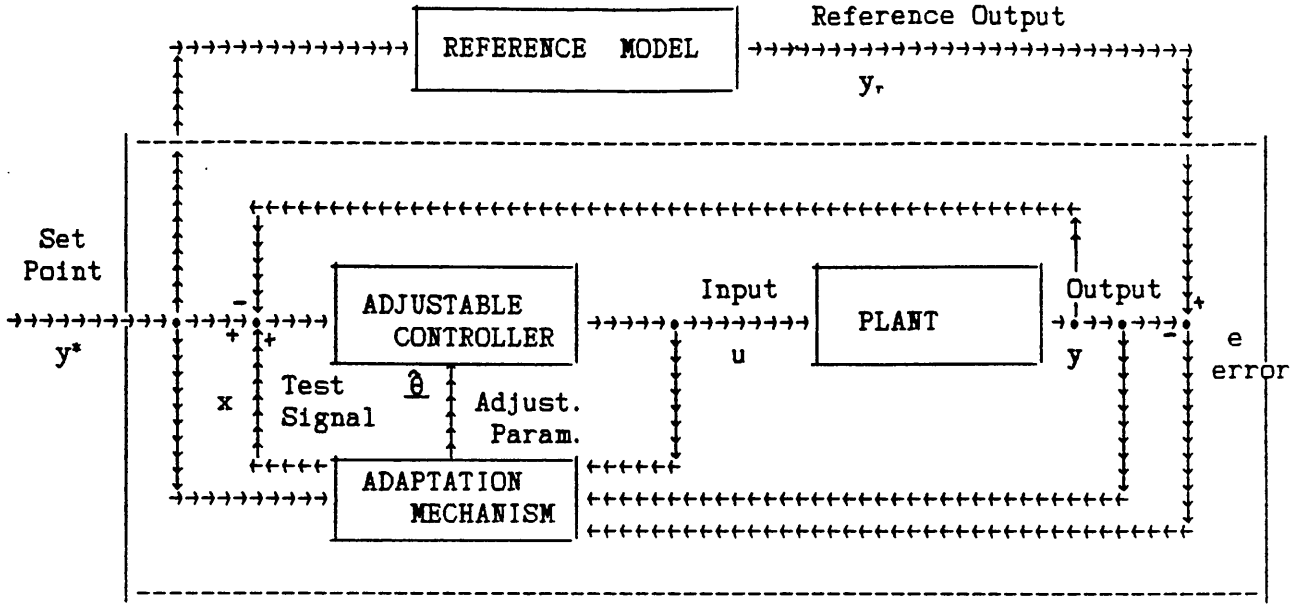


Fig. 2.1: Reference Model Adaptive Control

At the beginning this method was used mainly for servo problems, i.e., for time-varying set points. In regulator problems, where the set point is constant, a test signal must be added to the controller in order to generate enough excitation for parameter convergence.

The main advantages of the model-reference approach are the high speed of adaptation and the detuned control action generated by the smooth reference trajectory. This method also simplifies the stability and convergence analysis of the algorithms, representing an easy way of designing stable adaptive controllers. However, there are few industrial applications of this type of controllers due to the difficulties encountered in its implementation. It is, in general, necessary to know the model structure of the plant. In addition, there is no easy procedure to design the reference model.

This work will focus on the second technique, self-tuning control, which is one of the simplest adaptive schemes to be conceived and designed. Moreover, most of the reported successful industrial applications of adaptive control use the self-tuning approach.

Figure 2.2 gives a diagram of a general self-tuning algorithm.

It is generally understood that a self-tuner simply estimates model parameters of an adjustable system on-line based on input/output data, and then adjusts the settings of a standard control law based on the current parameter estimates [Seborg et al, 1986].

Probably the first self-tuning algorithm was proposed by Kalman in 1958. This algorithm estimates the unknown parameters of a deterministic input/output linear model via an on-line least-squares estimator. It then adjusts the input according to an output dead-beat control law based on the estimated parameters.

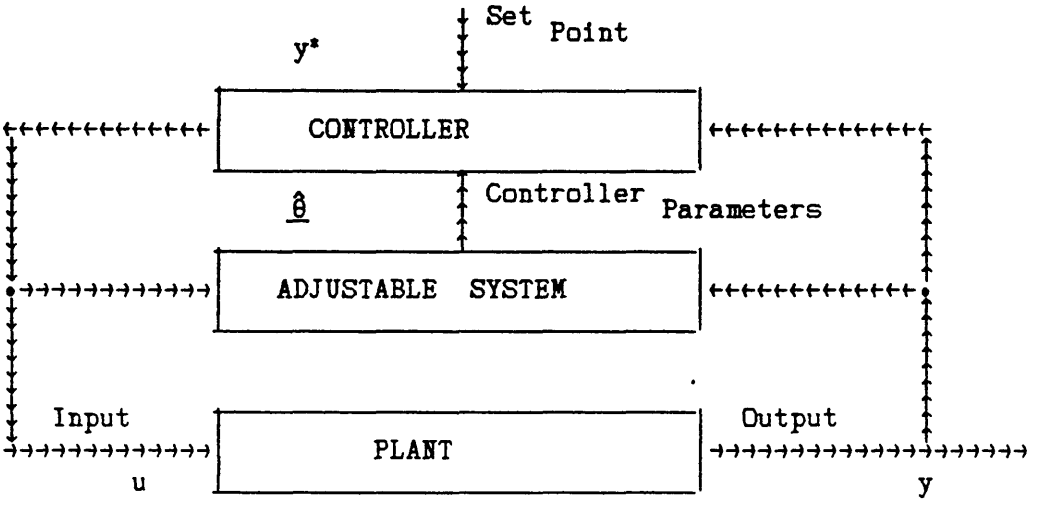


Fig. 2.2: Self-tuning Adaptive Controller

Due to the lack of sufficiently advanced theory and computer technology Kalman's self-tuner went almost unnoticed until 1970, when Peterka presented a more advanced algorithm based on the ideas of Kalman. The main extensions of Peterka were to consider a stochastic model and a minimum-variance control law. However, no theoretical results concerning stability and convergence were presented.

The paper of Åström and Wittenmark (1973), which considered an algorithm very similar to Peterka's self-tuner, was the one that initiated the self-tuning adaptive control as a valid area for research and development. In their work, Åström and Wittenmark showed that if the parameter estimates converge, then the self-tuning minimum variance control law will converge to the optimal control law based on the true model parameters. This is true even if biased model parameters were obtained in the estimation as a consequence of not explicitly estimating the noise parameters. The key point is that under certain conditions it is not necessary to identify the true process, but to adjust the control law to get the best control.

This result encouraged the application of Åström's self-tuning regulator in a variety of industrial processes [Borison & Wittenmark, 1974; Cegrell & Hedqvist, 1975; Borison & Syding, 1976].

Later some problems associated with the self-tuning regulator became evident in practical implementations:

i) the control action tends to be excessive causing oscillatory behaviour and sometimes saturation

ii) the controlled system is unstable for non-minimum phase processes

iii) the performance of the self-tuner is highly sensitive to a correct choice of model order and time-delay

iv) the stability of the algorithm is affected by large and frequent unmeasured deterministic disturbances

v) sometimes the parameter estimator blows-up

The design of new algorithms which include more robust control strategies and more reliable parameter estimators permitted the field to reach a state of maturity so that self-tuning regulators can now be applied with confidence in industry [Isermann, 1985].

In the following, a brief discussion about the different kinds of self-tuners that can be designed will be presented. The first part considers only feedback SISO algorithms, and the next two sections treat the feedforward and the multivariable problem.

2.1.- SISO Feedback Algorithms

The design of a self-tuner starts with the selection of a controller and an adjustable system [see Fig. 2.2]. The adjustable system is formed by a model of the plant and by a parameter estimation algorithm. Hence, in principle, the controller, the model and the parameter estimator define a self-tuning algorithm. However, despite the fact that two algorithms use the same control law, model and parameter estimator, they are not necessarily the same algorithm. This is due to the different ways of implementing an adaptive controller, which are very important in the final performance of the adaptive controller, as is shown in chapter 5.

The two main differences in implementation are;

a) the structure of the adaptive algorithm

b) the type of information used in the data vector of the parameter estimator and control law

a) The structure of a self-tuning controller can be implicit (direct) or explicit (indirect). In the latter case, the control parameters are explicitly obtained from the model parameter estimates through further computations. There is no direct relationship between the control parameters and the parameter estimates.

For example, consider a plant modelled by the following polynomial equation [see section 3.1]:

$$A(z^{-1})\,y_k = B(z^{-1})\,u_{k-du} \qquad (2.1)$$

where A and B are polynomials of order n in the backward difference operator z^{-1}; and y and u are the measurement and the control, respectively. The process delay is given by du.

If the control objective is to drive the output to zero as soon as possible, then it is necessary to do some calculations with the model parameters of A and B [see Appendix I] to obtain the control u_k. In this case, the controller is called explicit.

In general, this approach needs to estimate fewer parameters, and is robust against load disturbances and unknown or varying time-delays. On the other hand, the explicit method is more time consuming, since further computations are needed to get the control parameters.

In the implicit approach, a predictive model of the plant is used. The parameter estimates of the predictive model can be directly related to the control parameters, avoiding further computations.

In this case, the plant model is:

$$y_{k+du} = G(z^{-1}) \cdot y_k + H(z^{-1}) \cdot u_k \qquad (2.2)$$

where the polynomial G is of order n-1 and H is of order n+du-2.

If, as in the previous case, the control objective is to drive the output to zero as soon as possible, then the control u_k can be computed directly from eq. (2.2).

The main advantages of the implicit method are its simplicity and the fact that, in general, it needs less computing time than the explicit method. However, it appears to be more sensitive to unknown and varying time-delays; and it may be less robust against load changes.

b) According to the information used in the data vector of the parameter estimator, a self-tuner can be positional or incremental.

If the estimator is fed with the full value of the variables (inputs, outputs or states), then the self-tuner is called positional.

Incremental algorithms are those that instead of using the full value of the current variables [y_k], they use the difference between the current value and the value of the variable at the previous sample-time [$y_k - y_{k-1}$]. It has been claimed [Clarke et al, 1983] that incremental algorithms are more robust than positional algorithms against step-load changes and for offset elimination [see Chapter 5]. Clarke and collaborators (1983) further generalized the idea to du-incremental controllers, where the data used by the estimator is the difference between the current value and the value of the variable du sample-times earlier [$y_k - y_{k-du}$], where du is the process delay.

After clarifying some basic concepts related to the implementation of self-tuners, a classification of the control laws currently used can be attempted. The control strategies can, in principle, be grouped according to:

a) one step-ahead control
b) pole placement
c) GPC (general predictive controllers)
d) LQG optimal controllers

a) The first self-tuners used one step-ahead control [Kalman, 1958; Peterka, 1970; Åström & Wittenmark, 1973].

Basically, the idea is to minimize a one-step cost function of the input and output of the plant at the earliest time for which the control action will have an effect on the output. This, of course, will depend on the process delay. These types of algorithms commomly use the implicit method.

The self-tuning regulator of Åström & Wittenmark (1973) minimizes the variance of the predicted output at the next time interval for which the control will have an effect. It is probably the simplest self-tuner and was used in many successful industrial applications. However, it became clear that the algorithm needed some modifications in order to make it more acceptable to practitioners. The main disadvantage associated with minimum-variance strategy is the excessive control, which causes oscillations, saturations and in non-minimum phase plants, instability.

The self-tuning controller of Clarke and Gawthrop (1975), uses a generalized minimum-variance (GMV) control strategy, where the one-step cost function includes a penalization of the control effort. In this way, the control is detuned and the self-tuner can avoid saturations and overcome non-minimum phase plants. On the other hand, it is necessary to select another initial parameter, the weight in the control action, which is not always an easy task, although it can be used as a tuning parameter. The algorithm still remains sensitive to a proper selection of time-delay and model order.

b) A pole placement self-tuning regulator was presented by Wellstead et al (1979). The control strategy consists in positioning the closed-loop poles at desired locations, so that the engineer can prespecify the dynamics of the controlled system. Saturations can be avoided and non-minimum phase plants can be controlled without problems. Furthermore, because of the explicit structure used, Wellstead self-tuning regulator is robust against unknown and varying time-delays.

The main drawback of pole placement self-tuners is the amount of computation required. A set of linear equations should be solved at each iteration and the control parameters must be computed from the estimates of the explicit model. In addition, the algorithm is sensitive to the selection of the model order and there is no clear procedure to select the closed-loop poles.

c) Several adaptive algorithms that share the same design philosophy, have been successfully implemented in a variety of systems. These algorithms can be considered as GPC [General Predictive Control] self-tuners, introduced by Clarke (1986).

In the GPC algorithms, a cost function of the input and output (or states) is minimized over a finite time T in the future, larger than the process delay, instead of minimizimg the cost function in the least possible time, as in the one-step ahead control. The GPC's predict future outputs (or states) based on given assumptions about future inputs. These assumptions give an extra degree of freedom that allows the design of a whole class of new adaptive controllers. According to Clarke (1986), these designs can be "considered as versions of LQG" or as "extensions to the GMV predictive controller over a longer time-horizon".

The prediction-horizon and the control-horizon are key design parameters in this type of controllers. Usually they are set at the same value.

One of the simplest self-tuners that can be understood from a GPC point of view, is the robust self-tuning regulator of Ydstie (1982). The cost function to be minimized is the variance of the output error and the prediction-horizon is the same as the control-horizon.

For example, consider the plant modelled by eq. (2.1), where the control objective now is to drive the output to zero in T time intervals, where T is larger than the time delay du. It can be demonstrated [Ydstie, 1982] that the predictive model in this case is:

$$y_{k+T} = G(z^{-1})\,y_k + H(z^{-1})\,u_{k+T-du} \qquad (2.3)$$

here, the orders of the G and H polynomial are the same as in (2.2).

It can be seen that eq. (2.3) includes future values of the control input u [from time k+T-du to time k+1]. The assumptions about these future inputs define different controllers [see section 3.1].

The algorithm has shown in practice [Ydstie et al, 1985; Hiram & Kershenbaum, 1985] to be robust against non-minimum phase plants, unknown and varying time-delays, and model order selection, even though the algorithm retains the simplicity of the original self-tuner.

It appears to be that this self-tuner can be sensitive to deterministc load-changes [Hiram et al, 1986], probably due to the implicit structure of the algorithm [see Chapter5].

More sophisticated GPC's have been proposed. The adaptive predictive control system (APCS) [Martin-Sánchez, 1974; Martin-Sánchez and Shah, 1984] is an implicit GPC algorithm in which the control-horizon and the prediction-horizon are the same. The criteria of minimization is the output error at the time-horizon. The prediction of the future outputs is based on the fact that the controlled system must follow a given trajectory. Then, the assumed future inputs are those which drive the system in the prespecified manner.

The fact that the plant outputs follow a given reference trajectory makes this algorithm very close to an adaptive model reference approach. The APC's algorithm does not possess the simplicity of the original self-tuning, moreover there is no clear guide on how to select the reference trajectory. On the other hand, it has been successfully implemented in different kinds of systems.

The MUSMAR algorithm [Menga and Mosca, 1980; Mosca et al, 1986] is another type of highly sophisticated GPC self-tuner. It is based on "a set of separate predictive models, covering all future behaviour of the plant i/o joint process over the control horizon" [Mosca et al, 1986]. The control strategy used is an extended-horizon generalized minimum-variance control. The authors claim robustness against unmodeled dynamics, selection of model order and unknown time-delays.

The convergence to the optimal LQG solution of the control had also been proved. The major drawback of the MUSMAR self-tuner is the amount of computations involved in the simultaneous on-line estimation of various models, equal in number to the sample-intervals included in the the control-horizon.

A GPC algorithm with multiple predictors has also been proposed by Ydstie (1986), where the time-horizon of an extended-horizon minimum-variance self-tuning controller is automatically adjusted based on pole location criterion.

d) Finally, a self-tuning controller can be designed using optimal LQG (linear quadratic gaussian) control strategy. In this case, a GMV-type cost function is minimized over an infinite time-horizon.

Optimal LQG self-tuners can be designed using state-space or polynomial representation.

In the state-space approach, a Riccati equation must be iterated at each sampling interval [Peterka, 1973], in order to obtain the optimal control. This requires large amount of computations that can be reduced by means of some approximations [Lam, 1980].

In the polynomial [transfer function] case, the computation of the control parameters can be done implicitly [Grimble, 1984] by estimating them directly from input/output data; or the parameters of the control can be explicitly obtained from the model parameters [Peterka, 1982; Trulsson and Ljung, 1985].

No matter which way an LQG self-tuner is implemented, it is much more expensive in computing time than other self-tuners. The selection of the weights in the cost function also pose a problem from an application point of view. It appears that the above are quite serious disadvantages since there are almost no reported implementations in real processes using the LQG approach [Seborg et al, 1986]. On the other hand, LQG self-tuners generally give stable control for unstable and non-minimum phase plants, and it is not necessary to know the time delay a priori.

Before finishing this section it is worth mentioning some robustness problems associated with the parameter estimation in self-tuning algorithms. This step is the key to adaptability since it allows the on-line change of the controller settings.

The discussion will be centred on the recursive least squares estimator (RLS), since this is the most widely used method in practical implementations.

The parameters updating equation in the RLS estimator is of the form:

$$\hat{\underline{\theta}}_k = \hat{\underline{\theta}}_{k-1} + K_k \epsilon \qquad (2.4)$$

where $\hat{\underline{\theta}}$ is the parameter estimates vector, K is the correction or Kalman gain, and ϵ is the innovation defined by the difference between the measured output and the output estimated based on $\hat{\underline{\theta}}_{k-1}$.

The main problem of the standard RLS from an adaptive point of view, is the tendency of the correction-gain to go to zero when the parameters converge; thus when the plant suffers a change in its dynamics, the estimator is slow to correct the parameter estimates and the control performance deteriorates, and can even become unstable. It is said then that the estimator "goes to sleep".

A simple way to avoid the above problem is to give more weight to the most recent data. This is generally done by using a constant forgetting factor in the RLS algorithm, which is equivalent to an exponential weighting of the data. The constant forgetting factor method is currently used, even though it is not very robust.

If the system is persistently excited this approach does not present any problem, but in real plants where there are long periods of calm, this approach can be catastrophic. In this case the algorithm forgets important data, and the new data coming into the estimator does not provide any useful information since the plant is near steady state operating conditions. The correction-gain then starts increasing since it is inversely proportional to the amount of information, denoting a decreasing confidence in the parameter estimates. In this situation, any disturbance entering the system will produce an "explosion" in the parameter estimates.

There are basically two approaches to solve the above mentioned problem.

One is called the "covariance resetting" method [Goodwin & Sin, 1984], in which the covariance of the RLS estimator is reset at various times, and no forgetting is used. The correction-gains are proportional to the trace of the covariance matrix. Thus indirectly the correction gains are periodically adjusted, avoiding the "goes to sleep" phenomenon. However, it should be decided how often and in what amount the covariance will be re-initialized.

A simpler and more systematic approach is to use a variable forgetting factor [Fortescue et al, 1981], which is automatically adjusted according to the information content in the RLS. This approach appears to work very well in real systems [Dumont, 1982; Ydstie et al, 1985]. The main weakness of the variable forgetting scheme is that, theoretically, it does not ensure that the "blow up" phenomenon is avoided. Nevertheless, this phenomenon has not occurred in practical implementations.

Some novel variable forgetting algorithms which theoretically solve the "blow-up" of the covariance matrix have been recently proposed [Saelid and Foss, 1985; Bertin et al, 1986; Kulhavy, 1986]. These new estimators claim to have improved robustness when compared with the Fortescue scheme.

2.2.- SISO Feedforward Self-Tuning Regulators

The majority of the self-tuning algorithms have been formulated as purely feedback controllers, in spite of the appealing characteristics of feedforward control, namely the ability to act in an anticipatory manner and to achieve complete disturbance rejection. Moreover, when a process controlled with feedback self-tuning is subjected to periodic deterministic load disturbances, unpredictable variations of the model parameters can occur. This is because the adaptive algorithm cannot differentiate between a disturbance or an actual parameter change since only the control and the measurement are available to the controller. Furthermore, in this case the estimated gain becomes very large, deteriorating the closed loop system behaviour [Morris, et al, 1982; Hiram et al, 1986].

The main drawback of feedforward control, i.e., the need for a good process model, can be overcome with adaptive feedforward algorithms as model parameters can be estimated on-line.

There are different ways to design feedforward self-tuners. For example, LQG optimal control can be used [Peterka, 1982; Hunt et al, 1986]. This approach, as the feedback counterpart, requires excessive computations making it very unpopular among practitioners. The reported results include only simulations.

Using a polynomial [transfer function] approach with explicit LQG optimization, Sternard (1986) presented some simulations with linear models of order 2 and 3. The results showed a considerable improvement in the performance of the controlled system when adaptive feedforward is used. The LQG feedforward self-tuner also seems robust to model order selection and unknown time delays.

Another approach was taken by Schumann and Christ (1979), who presented several feedforward algorithms of simple design that performed reasonably well under a variety of disturbances. However, the proposed controllers require the knowledge of the model order and time-delays, and considered only feedforward action. Based on simulations with second and third order linear models, Schumann and Christ found that the adaptive feedforward controllers adapt very quickly and, in general, improve the closed-loop performance when compared with pure feedback, especially in processes without time delays.

A GMV multivariable feedback-feedforward self-tuner was tested with non-linear simulations and pilot plant experiments in a distillation column [Morris et al, 1985]. A polynomial matrix [transfer function] approach was used to model the system and the algorithm was implemented in a positional implicit form.

It has been shown that, most of the time, feedforward adaptive control significantly improves the performance of the controlled system, especially in reducing the transient response. The fast adaptation of the feedforward gain may also be observed from the experiments done. The GMV feedback-feedforward self-tuner suffers the same drawbacks of the purely feedback algorithm: sensitivity to selection of the model order, varying and unknown time-delays and selection of the weight in the control action.

Allidina et al (1981) point out that the above algorithm can cause oscillatory behaviour due to the high value of the feedforward gain at certain frequencies.

To detune the feedforward control action, they proposed a new algorithm which minimizes the following cost function:

$$J = E\{\phi^2(k+du)\} \qquad (2.5)$$

where E represents expectation, du is the control delay, and ϕ is given by:

$$\phi(k+du) = p\cdot y_{k+du} + q\cdot u_k + s\cdot w_{k+du-dw} - r\cdot y^* \qquad (2.6)$$

y, u, w, and y* are the output, the control, the measured disturbance and the set point respectively. The scalars p, q, r, and s represent proper weights and dw is the disturbance delay.

The novelty in Allidina's work is to consider the measured disturbance in the cost function. In this way, the selection of the weight s determines the dynamic response of the feedforward term.

The algorithm was tested in the temperature control-loop of a pilot-scale nuclear reactor. The use of adaptive feedforward control greatly improved the closed-loop performance. The proposed controller requires a proper selection of the weight of the feedforward term [s], a fact that detracts from practical implementations [Gawthrop, 1982].

2.3 Multivariable Self-Tuning Regulators

The design of multivariable self-tuning regulators is another area not fully developed in adaptive control engineering. On the contrary, the bulk of the proposed algorithms are just direct extensions of the SISO self-tuners without explicit considerations for the particular characteristics of the multivariable processes.

Typical problems that appear in a multivariable process are:

a) interaction

b) delay-structure

c) non-minimum phase zeros

d) parameter estimation

a) In a multivariable processes generally all the inputs affect all the outputs, causing serious difficulties in the control.

b) Unlike the SISO case, in multivariable plants the delay is not characterized by a scalar, but by a matrix since each input-output pair has associated a particular delay that can be different for each pair.

Wolovich and Falb (1976) introduced the concept of "interactor matrix", an invariant under dynamic compensation, which is associated with every polynomial matrix representation of a linear dynamic system. The interactor is represented by the multiplication of a diagonal matrix by a lower triangular polynomial matrix. It was later shown [Elliot and Wolovich, 1982] that the interactor matrix is one way of characterizing the system delay structure.

c) The non-minimum phase phenomenon is also peculiar in multivariable systems. A non-minimum phase zero associated with one particular element of the transfer matrix does not necessarily give rise to non-minimum phase effects [MacFarlane and Karcanias, 1976]. On the other hand, non-minimum phase behaviour in multivariable plants need not be associated with all of the system; it can be related with only one output [see Appendix E].

d) The number of parameters required to model a multivariable system tends to be one of the most important obstacles in adaptive multivariable design [Elliot and Wolovich, 1984], especially when polynomial matrix [transfer function] representations are used.

There have been two approaches used to solve the multivariable self-tuning problem: the transfer function [or polynomial matrix representation] and the state-space.

I) The matrix transfer function, which has attracted most of the attention, is a direct consequence of the successful applications of SISO self-tuners.

Borison (1975, 1979) appears to have presented the first MIMO self-tuner, which is an extension of the minimum variance self-tuning regulator of Åström and Wittenmark (1973). It also suffers the same limitations, i.e., it cannot cope with non-minimum phase plants and it produce excessive control action. In the same vein, Koivo (1980) designed a MIMO self-tuner as an extension of the SISO self-tuning controller of Clarke and Gawthrop (1975). The proposed algorithm deals with non-minimum phase processes by weighting the control actions. However, the selection of these weights is an even more difficult problem than in the single variable case because a matrix of weights is required. A similar algorithm is shown by Bayoumi, Wong and EL-Bagouri (1981). In their work, an explicit parameter estimation method was used to avoid the estimation of too many parameters, especially for long time delays.

All of the above algorithms are sensitive to the selection of model structure and to unknown and variable time-delays.

Prager and Wellstead (1980) extended the pole-placement SISO self-tuning controller of Wellstead et al (1979) to give a multivariable regulator which can cope with non-minimum phase systems and unknown and variable delays. However, the algorithm is extremely complicated, requiring excessive computations, and the problem of how to select the pole locations is even harder than in the SISO case.

There have been reported some successful implementations of polynomial matrix self-tuners. For example, Keviczky et al (1978), using a special version of Borison's MIMO self-tuner, controlled the composition of a cement material blending process. They claimed that this self-tuner presented better performance than standard controllers (PI/PID). Also, Buchholt and Kummel (1981) applied a Clarke-type multivariable self-tuning controller to a double effect evaporator pilot plant. The performance obtained was satisfactory, although modern non-adaptive techniques (modal, feedforward, LQG) could do better, but at a considerable extra cost in engineering time.

The algorithms previously mentioned consider a scalar delay instead of considering the more general case of the matrix system delay discussed earlier. Assuming a single delay for all of the input/output pairs is equivalent to representing the interactor as a scaled identity matrix. Undoubtedly, this is an oversimplification that can degrade the performance of some outputs which may not be affected by any delay. Furthermore, many unnecessary parameters are estimated.

Some authors developed regulators which consider more general types of interactor matrices. Goodwin, Ramadge and Caines (1980) were the first to design and analyse such a regulator. The algorithm, a MIMO minimum-variance type self-tuner, considers a different delay for each output, regardless of I/O pair. This is equivalent to assuming a diagonal interactor matrix [Dugard et al, 1984]. The proposed self-tuner minimizes the variance of the predicted output errors. The prediction-horizons are the same as the delays associated with the outputs.

A GMV MIMO self-tuner with a diagonal interactor matrix was proposed by Morris et al (1982). It was successfully implemented in a pilot-scale distillation column [Vagi et al, 1985].

Recently, a more general approach was taken by Dugard, Goodwin, and Xianya (1984), who designed a MIMO minimum-variance (MV) controller for a process with a general interactor matrix. They found that unconditional minimum-variance control can be achieved only for processes with a diagonal interactor matrix. In the general case, unconditional minimum-variance can be obtained only for the first component of the output vector; for the second and subsequent components, the minimization is subject to constraints. In other words, it is impossible for general systems to achieve independently minimum variance for all of the outputs. Furthermore, the total variance depends on the order of the outputs in the model. Considering the knowledge of the interactor matrix unrealistic, they proposed a MIMO self-tuner which is a generalization of Ydstie's extended-horizon SISO controller. The control action is designed to minimize the variance of all of the outputs at the same future time T, the time-horizon, which is larger than or equal to the largest delay affecting the multivariable structure.

Apart from avoiding the necessity of knowing the interactor matrix, the extended-horizon MIMO self-tuner proposed by Dugard can overcome some non-minimum phase plants. However, the idea of using a single-horizon for all of the outputs can affect the performance of the outputs which do not have any associated delay.

An extended-horizon MIMO MV self-tuner was presented by Lee and Lee (1983). The control objective was the minimization of the output error at different horizons for each output. The control horizon for a given output is chosen larger than or equal to the least delay affecting that particular output. Non-minimum phase systems can be controlled by a proper selection of the horizons. Moreover, it is only required to increase the control-horizon of those outputs affected by time-delays. However, Lee and Lee's design assumes knowledge of the interactor matrix.

An important problem associated with the polynomial matrix self-tuning controllers is the large number of parameters needed for estimation, which is drastically increased with model order and time-delays.

II) State-Space: Several authors prefer the state-space approach for designing MIMO self-tuners.

For example, using canonical forms the number of parameters to be estimated can be reduced as compared with the transfer function matrix approach [Guidorzi, 1979; Niederlinski and Hajdasinski, 1979]. It is true that canonical forms can also be used with polynomial matrix equations, but these representations are less understood.

It also seems that for multivariable systems the control formulation is simpler using state-space than that using polynomial matrices and probably the control calculations require less computing time.

Other advantages associated with state-space representation are the possibility to incorporate previous knowledge through mechanistic models, and subsequently the extension to non-linear systems.

Some authors claim that self-tuners based on state space deal better with sudden changes in plant gains and time-delays [Clarke, 1985; Warwick, 1981; Schumann, 1986], often encountered in the chemical industry.

State-space self-tuning controllers can be designed using MV, GMV, LQG, pole placement and GPC's control laws.

Schumann (1979, 1986) and Warwick (1981) compare state-space self-tuners with the matrix polynomial counterparts. Their studies, based on simulations with third and sixth order linear models, suggested that the state space approach is more robust against time-delays, drastic changes and non-minimum phase systems, although the matrix polynomial algorithms gave better regulatory performance and were much less time-consuming. However, they used rather complicated algorithms for the state/parameter estimation.

Schumann (1982) also showed some experimental results on an air conditioning pilot-plant system, and concluded that despite the satisfactory performance, the multivariable self-tuning design needed much more development.

State-space algorithms involve the product of unknown states and parameters in the model. Then, in state-space self-tuners it is necessary to estimate both the states and the parameters, situation which poses a non-linear [or, at least, a bilinear] problem. The estimators tend to diverge easily and to be complicated and time consuming. This is probably the main reason why state space self-tuners are unpopular among practitioners.

The first approach to the state/parameter estimation problem was through an extended-Kalman [E-K] filter [Jazwinski, 1970]. The algorithm in general gives biased estimates and may sometimes diverge. Both of these problems are due to an incorrect specification of the noise covariances and also to the dependence of the Kalman gain on the parameter estimates. Ljung (1979) modified the E-K filter and improved its convergence properties. However, the modified algorithm requires the computation of matrix derivatives which for high-order systems are very difficult to implement. A simplified version for an innovation model is shown in Goodwin and Sin (1984), although the authors warned against this approach since the non-linear nature of the estimator causes a series of difficulties.

Goodwin & Sin (1984) advocate instead a prediction error approach, in which the non-linear estimation is separated in two linear estimators. First, the parameters of a polynomial matrix model are estimated using input/output data. Then, the states are obtained via a linear Kalman filter where the state-space model parameters and the Kalman-gains are derived from the parameters of the polynomial matrix model provided by the first estimator. However, the problem arises of how to relate the parameters of the input/output model to the state space model. Using a very particular state space canonical form, Nelson and Stear (1976) designed an algorithm employing a prediction-error approach. Based on extensive simulation work with this algorithm, the authors claimed much better performance, robustness and convergence than with the E-K filter approach. The algorithm was also much less time consuming. However, R. Padilla, C. Padilla and Bingulac (1978) pointed out that the canonical form used by Nelson and Stear was not general and, in fact, was very restrictive. A similar algorithm was reported by Irwin and Roberts (1976), in which a Luenberger canonical form [always attainable for minimal systems] was used to relate the input/output data to the state space model. Several cases were studied. For the general stochastic case, an innovation state space model was suggested. First, the parameters of a polynomial matrix model were estimated using a Kalman filter and input/output data. The states were then obtained through a linear filter with the state-space model parameters and Kalman-gains derived from the parameter provided by the first estimator. The relationship between the parameters obtained from the input/output data and the state space parameters was not at all simple and involved matrix inversion.

Similar to the prediction error approach, the simpler bootstrap method gives unbiased parameter estimates without the need of estimating noise parameters.

It seems that Rowe (1967, 1969) introduced the term bootstrap to designate an on-line instrumental variable recursive parameter estimator. In the instrumental variable method, a sequence of filtered outputs are used to estimate the model parameters [Wong and Polak, 1967]. The sequence of filtered outputs (instrumental variables) are uncorrelated with the residuals, giving in this way unbiased parameter estimates.

Rowe named his method bootstrap because the parameter estimates were used to generate the instrumental variables, which in turn were used in the parameter estimation; combining parameter estimates and instrumental variables in the same way as a bootstrap.

In this sense, Pandya (1972, 1974) generalized the term bootstrap to include all of the recursive parameter estimators characterized by the presence of two on-line estimators. One estimator is for the process parameters and the other for the filtered process outputs (or states, or instrumental variables); with continuous interchange of information between them. This process of sharing information can be repeated several times per sample interval, if desired, to improve the speed of convergence.

The most important features of the bootstrap scheme method are its simplicity and the generation of unbiased estimates. In general, they are non-minimum variance estimators, although the increment in the variance is moderate [Rowe, 1967]. A more serious problem is that the stability of such algorithms is not guaranteed because of the open-loop structure of the state estimator. More explicitly, the state estimates do not directly depend on the observed outputs, so that if the system is non-stationary the state estimates will tend to diverge. However, some techniques have been proposed [Pandya, 1974; Goodwin and Sin, 1984] to improve the stability of bootstrap algorithms.

The main difference between the bootstrap and the prediction error approaches is the different way in which these methods combine the two linear estimators. In the prediction error approach, the process parameter estimator does not need any state data, only input/output data, although the state estimator needs transformed parameter estimates.

On the other hand, the bootstrap algorithm uses the state estimates in the data vector of the model parameter estimator. The parameter estimates are then used in the state estimator. These two estimators with bidirectional flow of information are combined in the so called "bootstrap manner". The particular way in which the estimators are combined in the bootstrap scheme makes the algorithm simpler since there is no need for transformed parameter estimates. The generation of unbiased estimates is due to the use of filtered outputs [states, instrumental variables] in the parameter estimator.

El-Sherief and Sinha (1979) used an on-line bootstrap scheme to estimate both states and parameters of an innovation state-space model. The states are estimated using a stochastic approximation algorithm assuming known model parameters. The model parameters are provided by a series of RLS estimators applied sequentially, in which the measurement vectors include states and inputs. The two algorithms are combined together in a bootstrap manner. Apart from the fact that this algorithm is much simpler and less time consuming than the error-prediction approach, it gives unbiased parameter estimates even for the stochastic case. Thus there is no need to estimate the noise model.

A similar method was later used [El-sherief and Sinha, 1982] to design an LQG state-space self-tuner. A third order two-output one-input linear system was used to test the algorithm in simulations. The results showed good control and convergence of the parameters and states. A stochastic approximation algorithm is used to estimate the states, which means that the correction-gain for the states will go to zero as the sampled time increases. It is then expected that the proposed self-tuner would not be completely adaptive and would present problems if the plant suffers a change in its dynamics.

The main problem of simultaneous state/parameter estimation can be overcome by the simple Bootstrap Method. However, more robust state estimators are required to cope with time-varying plants.

In conclusion, the state space multivariable self-tuning design appears to be a good alternative to polynomial matrix design. However, much more work is required in this area.

CHAPTER 3 THEORY

In the present chapter all the developed algorithms will be presented. For the sake of completeness, the Robust Self-tuning Regulator of Ydstie (1982) is first introduced in its incremental form. Then, a feedback-feedforward self-tuning controller is derived along the same lines as Ydstie's self-tuner. Finally, a state-space self-tuning controller is considered. The SISO case is studied first and then, the state-space is generalized to multivariable plants.

3.1.- Polynomial Algorithms

3.1.1.- A Robust Feedback Self-tuner

Let us assume that the process to be controlled can be described by the following Delta model:

$$A(z^{-1})\cdot\Delta y_k = z^{-du}\cdot B(z^{-1})\cdot\Delta u_k + \Delta e_k \qquad (3.1)$$

the symbols z^{-1} and Δ denote the delay operator and the backward-difference operator respectively, so that:

$$z^{-1}\cdot y_k = y_{k-1} \quad \text{and} \quad \Delta y_k = (1-z^{-1})\cdot y_k = y_k - y_{k-1} \qquad (3.2)$$

The sequence $\{y_k\}$ represents the measured outputs, $\{u_k\}$ represents the control inputs and $\{e_k\}$ defines a white noise process.

The polynomials A and B are given by:

$$A(z^{-1}) = 1 + a_1\cdot z^{-1} + ... + a_n\cdot z^{-n}$$
$$B(z^{-1}) = b_1 + ... + b_n\cdot z^{-n+1}$$

where n is the order of the plant.

As in the original development of Ydstie (1982), an implicit structure is considered in this section. In Appendix I an explicit version of Ydstie's self-tuner is given. Simulations and experiments, shown in Chapter 5, suggest that the explicit version is more robust than the implicit algorithm against unmeasured deterministic disturbances.

In the implicit derivation it is necessary to transform eq. (3.1) into a predictive model.

The following formula is needed to obtain the predictive model:

$$1 - z^{-T} = F(z^{-1})\cdot A(z^{-1}) + z^{-T}\cdot G(z^{-1}) \qquad (3.3)$$

The parameter T is an integer larger than or equal to the delay du, and defines the prediction-horizon. The polynomial $F(z^{-1})$ is of order T-1 and $G(z^{-1})$ is of order n-1. Hiram (1983) shows that the polynomials F and G can always be found for any T.

If eq. (3.1) is multiplied by F and eq. (3.3) is then applied, yields:

$$[1 - z^{-T} - z^{-T}\cdot G(z^{-1})]\cdot \Delta y_k = z^{-du}\cdot F(z^{-1})\cdot B(z^{-1})\cdot \Delta u_k + F(z^{-1})\cdot \Delta e_k \qquad (3.4)$$

and rearranging:

$$y_k = y_{k-T} + G(z^{-1})\cdot \Delta y_{k-T} + H(z^{-1})\cdot \Delta u_{k-du} + \Delta v_k \qquad (3.5)$$

where $H(z^{-1})=F(z^{-1})\cdot B(z^{-1})$ is a polynomial of order n-2+T, and the residual is given by $v_k=F(z^{-1})\cdot e_k$.

The polynomials G and H are defined by:

$$G(z^{-1}) = \alpha_1 + \alpha_2\cdot z^{-1} + \ldots + \alpha_n\cdot z^{-n+1}$$
$$H(z^{-1}) = \beta_1 + \beta_2\cdot z^{-1} + \ldots + \beta_{n+T-1}\cdot z^{-n-T+2}$$

The structure of eq. (3.5) can be used for both estimation and control purposes.

Estimation:

Using the definitions of the polynomials G and H, eq. (3.5) can also be written in the following way:

$$y_k = y_{k-T} + \sum_{i=1}^{n}\alpha_i\cdot \Delta y_{k+1-i-T} + \sum_{i=1}^{n+T-1}\beta_i\cdot \Delta u_{k+1-i-du} + \Delta v_k \qquad (3.6)$$

The value of the control delay du is in general unknown and time-varying. The safest choice is then to consider du=1, i.e., the minimum possible value for a causal system. With du=1 the most recent values of the control u are used in the estimation. In that way any change in

the true control delay will not affect the stability of the self-tuner.

Equation (3.6) can now be expressed as a measurement equation :

$$y_k = y_{k-T} + \underline{\phi}_{k-T}'\underline{\theta} + \Delta v_k \qquad (3.7)$$

where :

$$\underline{\phi}_{k-T} = [\Delta y_{k-T}, \ldots, \Delta y_{k-n-T+1}; \Delta u_{k-1}, \ldots, \Delta u_{k-n-T+1}]$$
$$\underline{\theta}' = [\alpha_1, \ldots\ldots\ldots, \alpha_n; \beta_1, \ldots\ldots\ldots, \beta_{n+T-1}]$$
$$\Delta v_k = F(z^{-1})\Delta e_k$$

The expected value of v_k given the available information at time k-T is zero. The residual v_k is also uncorrelated with the known data at time k-T. The residual is a linear combination of values of the noise e from time k+1-T to time k, which by definition are zero mean random variables uncorrelated with the information known at time k-T. Then, a recursive least squares (RLS) can be applied to eq. (3.7) to estimate the α and β parameters. A variable forgetting factor is used to automatically adjust the tracking of the parameters according to the characteristics of the process [Fortescue, et al, 1981].

Control:

The problems associated with minimum-variance control, mainly due to excessive control action, were discussed earlier. A simple way to detune the control is using an extended-horizon MV control. The minimization of the variance of the output error is carried out at a future time T, where T is an integer larger than or equal to the process delay. In this way, the control action can be intuitively detuned without selecting a special control weight like in GMV control, or locating closed-loop poles like in pole-placement control.

To compute the control, the predictive output is required. Equation (3.6) can be used to get a predictor for y.

$$\hat{y}_{k+T} = Y_k + \sum_{i=1}^{T} \hat{\beta}_i \Delta u_{k+T-i} \qquad (3.8)$$

where Y_k represents the combination of known input/output data, and is given by:

$$Y_k = y_k + \sum_{i=1}^{n} \hat{\alpha}_i \cdot \Delta y_{k+1-i} + \sum_{i=1}^{n-1} \hat{\beta}_{T+i} \cdot \Delta u_{k-i} \qquad (3.9)$$

It also represents the predicted value of y_{k+T} if $\Delta u_k = \Delta u_{k+1} = \ldots = 0$.

The objective of the control is to minimize the variance of the predicted output error at time k+T, based on known data at time k. It can be shown that in this case, the controller drives the expected value of the predicted output error to zero, [Ydstie, 1982].

Any sequence that satisfies the following equation can achieve the control objective:

$$\sum_{i=1}^{T} \hat{\beta}_i \cdot \Delta u_{k+T-i} = y^*_{k+T} - Y_k \qquad (3.10)$$

where y^*_{k+T} represents the set-point, or reference value, at time k+T.

An infinite combination of inputs can be imagined which obey eq. (3.10). Some simple and reasonable sequences are suggested by Ydstie et al (1985) for the positional algorithm. The incremental version of these strategies are:

a) Chose constant control $u_k = u_{k+1} = \ldots\ldots = u_{k+T-1}$, or expressed in increments, $\Delta u_{k+1} = \Delta u_{k+2} = \ldots\ldots = \Delta u_{k+T-1} = 0$. The control is then:

$$\Delta u_k = (y^*_{k+T} - Y_k) / \hat{\beta}_T \qquad (3.11)$$

b) Concentrate the control effort on the first input of the sequence, i.e., $u_{k+1} = u_{k+2} = \ldots\ldots = u_{k+T-1} = 0$, or in terms of the increments, $\Delta u_{k+1} = -u_k$, $\Delta u_{k+2} = \ldots\ldots = \Delta u_{k+T-1} = 0$.

In this case:

$$\begin{aligned} \sum_{i=1}^{T} \hat{\beta}_i \cdot \Delta u_{k+T-i} &= \hat{\beta}_T \cdot \Delta u_k + \hat{\beta}_{T-1} \cdot (-u_k) \\ &= (\hat{\beta}_T - \hat{\beta}_{T-1}) \cdot u_k - \hat{\beta}_T \cdot u_{k-1} \\ &= (\hat{\beta}_T - \hat{\beta}_{T-1}) \cdot \Delta u_k - \hat{\beta}_{T-1} \cdot u_{k-1} \end{aligned}$$

and the control is:

$$\Delta u_k = (y^*_{k+T} - Y_k + \hat{\beta}_{T-1} \cdot u_{k-1}) / (\hat{\beta}_T - \hat{\beta}_{T-1}) \qquad (3.12)$$

c)Minimize the total control effort of the input sequence:

$$J = \sum_{i=1}^{T} u^2_{k+T-i}$$

The Lagrangian is given by:

$$L = \sum_{i=1}^{T} u^2_{k+T-i} + \lambda \cdot [\, y^*_{k+T} - Y_k - \sum_{i=1}^{T} \hat{\beta}_i \cdot \Delta u_{k+T-i}] \qquad (3.13)$$

the minimum is obtained for:

$$\partial L/\partial u_{k+T-1-i} = 2 \cdot u_{k+T-i} - \lambda \cdot [\hat{\beta}_i - \hat{\beta}_{i-1}] = 0 \quad ; \; i=2,\ldots.T$$

$$0 = 2 \cdot u_{k+T-1} - \lambda \cdot \hat{\beta}_1$$

or

$$\Delta u_{k+T-i} = \lambda \cdot \hat{\beta}_i / 2 \quad ; \; i=1,\ldots.T \qquad (3.14)$$

From eqs. (3.10) and (3.14) follows:

$$(\sum_{j=1}^{T} \hat{\beta}^2_j) \cdot \lambda/2 = y^*_{k+T} - Y_k \qquad (3.15)$$

Using the last two equations yields:

$$(\sum_{j=1}^{T} \hat{\beta}^2_j) \cdot \Delta u_{k+T-i} = \hat{\beta}_i \cdot (y^*_{k+T} - Y_k)$$

then:

$$\Delta u_{k+T-i} = \hat{\beta}_i \cdot (y^*_{k+T} - Y_k) / (\sum_{j=1}^{T} \hat{\beta}^2_j) \qquad (3.16)$$

Strategies (a), (b) and (c) are implemented in a receding-horizon form, i.e., only the first input of the sequence is computed and applied each interval.

It is worth noting at this stage, that the incremental strategies presented by Ydstie et al (1985) can be subjected to a misinterpretation. In order to avoid confusion, in the following analyses these strategies are interpreted in terms of increments.

a)Ydstie's incremental strategy 1

$$\Delta u_k = (y^*_{k+T} - Y_k)/(\sum_{j=1}^{T} \hat{\beta}_j)$$

that implies a control action with equal increments [control action linear with time over the horizon].

b)Ydstie's incremental strategy 2

$$\Delta u_k = (y^*_{k+T} - Y_k)/\hat{\beta}_T$$

as in eq.(3.11), a constant control sequence is applied. In this case, only the first control increment [over the horizon T] is different from zero.

c)Ydstie's incremental strategy 3

$$\Delta u_{k+T-i} = \hat{\beta}_i \cdot (y^*_{k+T} - Y_k)/(\sum_{j=1}^{T} \hat{\beta}^2_j)$$

the control is exactly the same as that defined by eq. (3.16).

In our experiments, eq. (3.11) was used because it is simple and gave good results. Using more sophisticated strategies does not necessarily yield better performance. There is a potential problem associated with the estimated parameter $\hat{\beta}_T$. If the time-horizon is not selected long enough, this estimated parameter can get very small or even change its sign. In this case the control can be seriously affected.

It is necessary to provide some initial design parameters to use the described adaptive controller.

The most important design parameters are:

a)time-horizon

b)order of the assumed plant model

c)information content of the RLS parameter estimator

There are also other parameters which are required, but these are less important for the algorithm performance. For example:

d)initial covariance matrix, usually set to a diagonal with very large elements (10^{12}).

e)initial parameter estimates, generally set to zero [taking care not to divide by zero initially].

The whole self-tuning algorithm can now be summarized:

Algorithm I

i) $v_k = y_k - \hat{y}_k$

$\hat{y}_k = y_{k-\tau} + \underline{\phi}_{k-\tau} \cdot \hat{\underline{\theta}}_{k-1}$

$\underline{\phi}_{k-\tau} = [\Delta y_{k-\tau}, \ldots, \Delta y_{k-n-\tau+1};\ \Delta u_{k-1}, \ldots, \Delta u_{k-n-\tau+1}]$

$\hat{\underline{\theta}}'_{k-1} = [\hat{\alpha}_1, \ldots\ldots\ldots, \hat{\alpha}_n;\ \hat{\beta}_1, \ldots\ldots\ldots, \hat{\beta}_{n+\tau-1}]$

ii) a RLS estimator* with variable forgetting factor is used to update the parameter vector.

$w_k = \underline{\phi}_{k-\tau} \cdot P_{k-1} \cdot \underline{\phi}'_{k-\tau}$

$n_k = 1 - w_k - v^2_k/\Sigma_0$

$\lambda_k = (n_k + \sqrt{n^2_k + 4 \cdot w_k})/2$

$\underline{K}_k = P_{k-1} \cdot \underline{\phi}'_{k-\tau}/(\lambda_k + w_k)$

$\hat{\underline{\theta}}_k = \hat{\underline{\theta}}_{k-1} + \underline{K}_k \cdot v_k$

$P_k = [P_{k-1} - \underline{K}_k \cdot \underline{K}'_k \cdot (\lambda_k + w_k)]/\lambda_k$

P_k = covariance matrix

Σ_0 = information content

λ_k = forgetting factor

iii) control calculation

$\Delta u_k = (y^*_{k+\tau} - Y_k)/\hat{\beta}_\tau$

$$Y_k = y_k + \sum_{i=1}^{n} \hat{\alpha}_i \cdot \Delta y_{k+1-i} + \sum_{i=1}^{n-1} \hat{\beta}_{\tau+i} \cdot \Delta u_{k-i}$$

iv) goto (i)

* The actual computer coding of this routine is based on the square root filter of Peterka (1975).

3.1.2.- A Feedback-Feedforward Self-tuner

The self-tuner presented here is an extension of the Robust Self-tuning Regulator, seen in the previous section, for the case of measurable disturbances.

A Delta model is also considered, but now it includes the measurable disturbance:

$$A(z^{-1})\cdot\Delta y_k = z^{-du}\cdot B(z^{-1})\cdot\Delta u_k + z^{-dw}\cdot C(z^{-1})\cdot\Delta w_k + \Delta e_k \qquad (3.17)$$

w represents the measurable disturbance and dw is the associated delay.

The C polynomial is given by:

$$C(z^{-1}) = c_0 + c_1\cdot z^{-1} + \ldots\ldots\ldots\ldots + c_q\cdot z^{-q}$$

where q is the order of the disturbance process.

The rest of the notation in eq. (3.17) is equivalent to eq. (3.1).

A predictive model can be obtained using eq. (3.3):

$$[1 - z^{-T} - z^{-T}\cdot G(z^{-1})]\cdot\Delta y_k = z^{-du}\cdot F(z^{-1})\cdot B(z^{-1})\cdot\Delta u_k + z^{-dw}\cdot F(z^{-1})\cdot C(z^{-1})\cdot\Delta w_k + F(z^{-1})\cdot\Delta e_k \qquad (3.18)$$

or equivalently:

$$y_k = y_{k-T} + G(z^{-1})\cdot\Delta y_{k-T} + H(z^{-1})\cdot\Delta u_{k-du} + D(z^{-1})\cdot\Delta w_{k-dw} + \Delta v_k \qquad (3.19)$$

The polynomial $D(z^{-1})=F(z^{-1})\cdot C(z^{-1})$ is of order q+T-1, and is defined by:

$$D(z^{-1}) = \delta_0 + \delta_1\cdot z^{-1} + \ldots + \delta_{q+T-1}\cdot z^{-q-T+1}$$

The prediction-horizon T is larger than or equal to the control delay du.

Considering that in real plants both delays, du and dw, are unknown and time-varying, it is safer to use their minimum possible values in eq. (3.19). However, an unnecessarily large number of parameters may be estimated.

The minimum value for du is one, since the control applied at time k cannot affect the output before time k+1. On the other hand, a disturbance measured at time k can already have affected the output at time k. This is the case for example if w_k entered the system at time k-½. Then the minimum value for dw is zero, so direct transmission between the disturbance w_k and the output y_k is considered.

Parameter estimation

Equation (3.19) can also be expressed in the following way:

$$y_k = y_{k-T} + \sum_{i=1}^{n} \alpha_i \cdot \Delta y_{k+1-T-i} + \sum_{i=1}^{n+T-1} \beta_i \cdot \Delta u_{k-i} - \sum_{i=0}^{q+T-1} \delta_i \cdot \Delta w_{k-i} + \Delta v_k \qquad (3.20)$$

and written in a more compact form :

$$y_k = y_{k-T} + \underline{\phi}_{k-T} \cdot \underline{\theta} + \Delta v_k \qquad (3.21)$$

where :

$$\underline{\phi}_{k-T} = [\Delta y_{k-T}, \ldots, \Delta y_{k-n-T+1}; \Delta u_{k-1}, \ldots, \Delta u_{k-n-T+1}; \Delta w_k, \ldots, \Delta w_{k-q-T+1}]$$

$$\underline{\theta}'_k = [\alpha_1, \ldots\ldots\ldots, \alpha_n; \beta_1, \ldots\ldots\ldots, \beta_{n+T-1}; \delta_0, \ldots\ldots\ldots, \delta_{q+T-1}]$$

$$\Delta v_k = F(z^{-1}) \cdot \Delta e_k$$

Like in eq. (3.7), the residual Δv_k is uncorrelated with the available data at time k-T and its expected value is zero. Then, a RLS algorithm with variable forgetting factor can be used to estimate the α, β and δ parameters.

Control Law

The same control criteria used in section (3.1.1) is used in this section. The control is designed to drive the expected value of the predicted output error to zero, in T sample intervals. T is the prediction-control horizon.

Using eq. (3.20) to predict the output at time T, yields:

$$\hat{y}_{k+T} = Y_k + \sum_{i=1}^{T} \hat{\beta}_i \cdot \Delta u_{k-i+T} - \sum_{i=0}^{T-1} \hat{\delta}_i \cdot \Delta w_{k-i+T} \qquad (3.22)$$

where Y_k represents the known data at time k, and is defined by:

$$Y_k = y_k + \sum_{i=1}^{n} \hat{\alpha}_i \cdot \Delta y_{k+1-i} + \sum_{i=1}^{n-1} \hat{\beta}_{T+i} \cdot \Delta u_{k-i} - \sum_{i=0}^{q-1} \hat{\delta}_{T+i} \cdot \Delta w_{k-i} \qquad (3.23)$$

To comply with the control objective, the following equation must be satisfied:

$$\sum_{i=1}^{T} \hat{\beta}_i \cdot \Delta u_{k-i+T} = y^*_k - Y_k - \sum_{i=0}^{T-1} \hat{\delta}_i \cdot \Delta w_{k-i+T} \qquad (3.24)$$

The control sequence $\{\Delta u_i\}_{i=k}^{T-1}$ can be chosen according to the strategies discussed in section (3.1.1).

Future values of the disturbance w, which in general are not known, also appear in eq. (3.24). Several hypothesis can be thought about these future disturbances. Some reasonable assumptions are the following:

a)The measurable disturbance is a white noise process. Then, all the future values are expected to be zero. In this case $\Delta w_{k+1}=-w_k$ and $\Delta w_{k+2}=\Delta w_{k+3}=\ldots=\Delta w_{k+T}=0$, so that:

$$\sum_{i=0}^{T-1} \hat{\delta}_i \cdot \Delta w_{k-i+T} = -w_k \cdot \hat{\delta}_{T-1} \qquad (3.25)$$

b)The disturbance is mainly deterministic, changing by steps and remaining constant for long periods. Then, it can be assumed that $w_k=w_{k+1}=\ldots=w_{k+T}$, or in terms of increments, $\Delta w_{k+1}=\Delta w_{k+2}=\ldots=\Delta w_{k+T}=0$, so that the combination of the future input disturbances is reduced to:

$$\sum_{i=0}^{T-1}\hat{\delta}_i\cdot\Delta w_{k-i+T} = 0 \qquad (3.26)$$

this is an usual type of disturbance in chemical plants.

c)The disturbance is a ramp function:

$$\Delta w_{k+i} = \Delta w_k \quad ; \quad i=1\ldots\ldots T$$

the expression for the future disturbances is in this case given by:

$$\sum_{i=0}^{T-1}\hat{\delta}_i\cdot\Delta w_{k-i+T} = \Delta w_k\cdot\sum_{i=0}^{T-1}\hat{\delta}_i \qquad (3.27)$$

According to the problem, any feedback-control strategy defined by eqs. (3.11), (3.12) and (3.16) can be used together with eqs. (3.25), (3.26) or (3.27) to get a feedback-feedforward control law.

The FB/FF controller requires basically the same design parameters as the purely feedback algorithm. The only extra parameter is the order of the disturbance process [q].

As an example, a combination of control strategy (a) [eq. (3.11)] and assumption (b) for the disturbance [eq. (3.26)] are used in the algorithm that follows.

Algorithm II

i) $\hat{v}_k = y_k - \hat{y}_k$

$\hat{y}_k = y_{k-\tau} + \underline{\phi}_{k-\tau} \hat{\underline{\theta}}_{k-1}$

$\underline{\phi}_{k-\tau} = [\Delta y_{k-\tau}, \ldots, \Delta y_{k-n-\tau+1};\ \Delta u_{k-1}, \ldots, \Delta u_{k-n-\tau+1};\ \Delta w_k, \ldots, \Delta w_{k-q-\tau+1}]$

$\hat{\underline{\theta}}'_{k-1} = [\hat{\alpha}_1, \ldots\ldots\ldots, \hat{\alpha}_n;\ \hat{\beta}_1, \ldots\ldots\ldots, \hat{\beta}_{n+\tau-1};\ \hat{\delta}_0, \ldots\ldots, \hat{\delta}_{q+\tau-1}]$

ii) a RLS estimator* with variable forgetting factor is used to update the parameters.

$w_k = \underline{\phi}_{k-\tau} P_{k-1} \underline{\phi}'_{k-\tau}$

$n_k = 1 - w_k - v^2_k / \Sigma_o$

$\lambda_k = (n_k + \sqrt{n^2_k + 4 w_k})/2$

$\underline{K}_k = P_{k-1} \underline{\phi}'_{k-\tau} / (\lambda_k + w_k)$

$\hat{\underline{\theta}}_k = \hat{\underline{\theta}}_{k-1} + \underline{K}_k v_k$

$P_k = [P_{k-1} - \underline{K}_k \underline{K}'_k (\lambda_k + w_k)] / \lambda_k$

P_k = covariance matrix

Σ_o = information content

λ_k = forgetting factor

iii) $\Delta u_k = (y^*_{k+\tau} - Y_k) / \hat{\beta}_\tau$

$$Y_k = y_k + \sum_{i=1}^{n} \hat{\alpha}_i \Delta y_{k+1-i} + \sum_{i=1}^{n-1} \hat{\beta}_{\tau+i} \Delta u_{k-i} + \sum_{i=0}^{q-1} \hat{\delta}_{\tau+i} \Delta w_{k-i}$$

iv) goto (i)

* The actual computer coding of this routine is based on the square root filter of Peterka (1975).

3.2 State Space Algorithms

In this section, the state-space approach is followed because; it suits better the characteristics of chemical processes, the control laws are easier to derive, and fewer parameters are needed to describe the multivariable system.

The estimator used is a modification of El-Sherief and Sinha (1979) algorithm for the non-stationary case. A RLS with variable forgetting factor estimates the varying parameters. A linear filter with constant gain, instead of a stochastic approximation filter, generates the state estimates. In this way, the stability of the whole scheme can be improved, limiting the possibility of state divergence.

A generalization of Ydstie's extended-horizon regulator which includes a different horizon for each output is developed, taking the idea of Lee and Lee (1983) of multiple prediction-horizons.

The multivariable regulator can cope with multiple delays without the knowledge of the interactor matrix, and can also deal with some non-minimum phase systems. Only the case of equal number of inputs and outputs is considered.

First of all, a single-input single-output case is presented for clarity of exposition, and later extended to the multivariable case. Finally, a feedback-feedforward multivariable adaptive controller is developed along the same lines.

3.2.1 The Single-Input Single-Output Case

A similar algorithm as the one described in section (3.1.1) is developed here. However an explicit structure is used, and the plant is described by a state-space model.

For different kinds of state-space models, refer to Appendix B.

A usual way of representing a linear stochastic process in state-space form is by an innovation model:

$$\bar{x}_{k+1} = \bar{A}\bar{x}_k + \bar{b}u_k + \bar{k}e_k \qquad (3.28.a)$$

$$y_k = \bar{c}\bar{x}_k + e_k \qquad (3.28.b)$$

where $\bar{x} \in R^n$, u, y are the state vector the control input and the measured output respectively.

The white noise process $\{e_k\}$ is the innovation sequence, and it is given by the difference between the measured output y_k and the predicted output $y_{k/k-1}$. In general terms, it can be said that the innovation "corresponds to the new information that is contained in each successive observation" [Goodwin & Sin, 1984].

The model (3.28) retains the structure of the optimal linear steady-state Kalman filter, and is one way of representing the most general kind of linear stochastic process that can be identified [Irwin & Roberts, 1976].

In adaptive control one of the most serious problems is the number of parameters that is necessary to estimate, above all in multivariable systems. Those parameters can be drastically reduced if state-space canonical forms are used.

For example, if the process modelled by eqs. (3.28) is observable, it can be expressed in an equivalent row-companion canonical form, see Appendix B:

$$\underline{x}_{k+1} = A\cdot\underline{x}_k + \underline{b}\cdot u_k + \underline{k}\cdot e_k \qquad (3.29.a)$$

$$y_k = \underline{c}\cdot\underline{x}_k + e_k \qquad (3.29.b)$$

Equations (3.29) contain the minimum number of parameters that completely specify the observable linear stochastic plant (3.29).

The vectors $\underline{b}$ and $\underline{k}$ are full column vectors, but the matrix A and the row vector $\underline{c}$ have the following special form:

$$A = \begin{vmatrix} 0 & 1 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ a_1 & \dots & \dots & a_n \end{vmatrix} \quad ; \quad \underline{c} = [\,1\ 0\ 0\ \dots\dots\ 0\,]$$

The control delay is not explicitly considered in this development, and for simplicity it is assumed that the delay is implicitly included in the model order. In other words, the model order n should be larger than the process delay. However, the time delay can be explicitly incorporated in the model structure [Goodwin & Sin, 1984] through an augmented state vector; but in this case it is necessary to know the delay a priori.

Estimation

For adaptive control purposes it is necessary to estimate both the parameter and the states in eq. (3.29). Then, it is convenient to transform eq. (3.29) into a more suitable form, in order to apply a RLS estimator.

As it is shown in Appendix C for the multivariable case, the output y of eq. (3.29.b) can also be expressed by:

$$y_k = z_{k-1}\,\underline{\theta} + v_k \qquad (3.30)$$

where :

$$z_{k-1} = [x'_{k-n};\ u_{k-1},\ u_{k-2},\ \ldots,\ u_{k-n}]$$

$$\underline{\theta}' = [a_1,\ \ldots,\ a_n;\ b_1,\ \ldots,\ b_n]$$

$$v_k = e_k + \sum_{i=1}^{n} k_i\, e_{k-i} \qquad (3.31)$$

A standard RLS estimator can now be directly applied to eq. (3.30) in order to obtain the parameter estimates vector $\underline{\theta}$.

If $\overline{Y}_{k-n}$ denotes the known information at time k-n, the residual v_k is uncorrelated with $\overline{Y}_{k-n}$. The residual at time k is a linear combination of values of the white noise process $\{e_k\}$ from time k-n to time k, which by definition are uncorrelated with $\overline{Y}_{k-n}$. Furthermore, the expected value of the residual in eq. (3.31) is zero. Then, an unbiased estimate of the vector $\underline{\theta}$ at time k can be obtained given knowledge of the state at time k-n, using ordinary least squares. In particular, a variable forgetting factor scheme is applied, as in section (3.1), to track time-varying parameters.

Note that in the data vector z_{k-1} an estimate of the state vector is required.

Filtering

Using the parameter estimates at time k, a linear filter can update the state estimates:

$$\hat{x}_{k/k-1} = \hat{A}_k\,\hat{x}_{k-1} + \hat{b}_k\,u_{k-1} \qquad (3.32.a)$$

$$\hat{x}_k = \hat{x}_{k/k-1} + \gamma_k\,c'\,(y_k - c\,\hat{x}_{k/k-1}) \qquad (3.32.b)$$

Only the directly observable state is updated by the innovation since the observation vector c is a unitary vector.

The form of the gain sequence $\{\gamma_k\}$ defines different kinds of filters. In the following some of them will be discussed.

a) Stochastic Approximation: A stochastic approximation method is a recursive estimator which converges to the true values in the limit, with the error correction term γ_k arbitrarily chosen [Saridis, 1974]. The first stochastic approximation algorithm was introduced by Robbins and Monro (1951), and later a generalized version was analysed by Dvoretzky (1956). The main feature of this kind of algorithm is its simplicity of formulation and implementation.

According to Dvoretzky, convergence of the method is ensured if the sequence $\{\gamma_k\}$ satisfy the following conditions :

$$\text{i) } \lim_{k\to\infty} \gamma_k = 0 \quad ; \quad \text{ii) } \sum_{k=1}^{\infty} \gamma_k = \infty \quad ; \quad \text{iii) } \sum_{k=1}^{\infty} \gamma_k^2 < \infty \qquad (3.33)$$

Saridis (1974) gives an intuitive interpretation of the above conditions. The first one smooths the error correction term, the second provides unlimited correction effort, and the last one cancels the effect of the individual errors for large number of iterations.

All the sequences of the form $\{a/(b+k)\}$, with $a > 0$ and $b \geq 0$, satisfy conditions (3.33); in particular, $\{1/k\}$ is commonly used [Saridis, 1974].

This is the kind of algorithm used by El-Sherief and Sinha (1979) in their bootstrap estimator. The main drawback of this scheme is the open-loop nature of the state estimator in the limit. For large number of iterations, the gain γ_k tends to zero causing divergence of the state estimates in non-stationary processes; affecting the stability of the whole estimation method. That is the reason why this filter should be changed for adaptive control purposes.

b) Constant Gain Filter : The simplest solution to the problem mentioned above is the use of a small constant γ in the filter eq. (3.32.a) [Goodwin and Sin, 1984]. The states will permanently be updated by the estimation error, limiting in this way any tendency to diverge.

The main drawback of this method is the slow rate of convergence, but because of its simplicity, this is the scheme used in our simulations and experiments; with values of γ ranging between .1 and .3 .

c) Adaptive Gain : Instead of using a constant gain, Pandya (1974) suggests the use of an adaptive gain γ_k in eq. (3.32.b) based on the theory of Jazwinski (1969).

In this case the sequence $\{\gamma_k\}$ is given by :

$$\gamma_k = 0 \qquad \text{if} \quad \hat{\sigma}_k \leq \lambda \qquad (3.34.a)$$

$$\gamma_k = 1 - \lambda/\hat{\sigma}_k \qquad \text{if} \quad \hat{\sigma}_k > \lambda \qquad (3.34.b)$$

λ is an upper bound on the noise variance σ_k, and the latter can be estimated recursively :

$$\hat{\sigma}_k = \{(k-1)/k)^n \cdot \hat{\sigma}_{k-1} + [\ 1 - \{(k-1)/k)^n\ ] \cdot e_k^2 \qquad (3.35)$$

where $e_k = y_k - \underline{c} \cdot \underline{x}_{k/k-1}$ is the estimated residual and n is a number larger than 1.

In eq. (3.35) there is an implicit forgetting method that weights more the recent data. Pandya considered n=2 as being generally adequate.

The adaptive gain method improves the rate of convergence and the stability of the filter by adjusting the error correction term γ_k according to the characteristic of the process noise.

d) Dynamic Kalman Filter: Of course, a non steady-state Kalman filter can be used to get state estimates, but in this case a Riccati equation must be iterated each sample interval, making the algorithm unnecessarily complicated.

In summary, the state/parameter estimation algorithm used in our work consists of two linear estimators combined in a bootstrap manner. The first estimator is a RLS with variable forgetting factor which used the previous inputs and state-estimates to generate unbiased parameter estimates of a canonical row-companion state-space model. The second estimator is a linear filter with constant gain that uses the process parameter estimates to generate the state-estimates. This process is done once at each sample time, but it can be done several times per interval in order to increase the speed of convergence. In our experiments however, it did not appear necessary to iterate.

Control Law

As in section (3.1), the control objective is to drive the expected value of the predicted output error to zero in T time intervals, where T is the control-prediction horizon. Then a predictor for the output at time k+T, given data at time k, is needed.

The optimal T-step ahead state predictor [Goodwin & Sin, 1984] for the model (3.29) is given by:

$$\hat{\underline{x}}_{k+T} = \hat{A}^T_k \hat{\underline{x}}_k + \sum_{j=0}^{T-1} \hat{A}^j_k \hat{\underline{b}}_k u_{k+j} \qquad (3.36)$$

and the optimal output predictor follows from eq. (3.29.b)

$$\hat{y}_{k+T} = \overline{Y}_k + \underline{c} \sum_{j=0}^{T-1} \hat{A}^j_k \hat{\underline{b}}_k u_{k+j} \qquad (3.37)$$

where $Y_k = \underline{c} \hat{A}^T_k \hat{\underline{x}}_k$ represents the knowledge at time k.

Similarly as in eq. (3.10), the input sequence that satisfies the following equation achieves the control objective:

$$y^*_{k+T} - \overline{Y}_k = \underline{c} \sum_{j=0}^{T-1} \hat{A}^j_k \hat{\underline{b}}_k u_{k+j} \qquad (3.38)$$

where y^*_{k+T} is the output reference at time k+T.

As in section (3.1) several strategies can be chosen to select a future control sequence that satisfies eq. (3.38).

For simplicity, only strategy (a) [eq. (3.11)] will be considered. In this case, the control is assumed constant during the horizon time, then:

$$u_k = u_{k+1} = \ldots = u_{k+T-1} \qquad (3.39)$$

Replacing (3.39) into (3.38) yields:

$$d u_k = y^*_{k+T} - \overline{Y}_k \qquad (3.40)$$

Ydstie (1982) has shown that for a controllable, observable and stable system, the scalar d is always different from zero provided a long enough horizon T is selected.

The control is then given by:

$$u_k = (y^*_{k+T} - \overline{Y}_k)/d \qquad (3.41)$$

with

$$d = \underline{c}\,\Sigma_{j=0}^{T-1}\hat{A}^{j}_{k}\,\hat{\underline{b}}_{k}$$

Equation (3.41) is used in a receding-horizon form, namely only the first control u of the sequence $\{u_i\}_{i=k}^{T-1}$ is computed and applied each sample interval.

The state-space self-tuner can be completely specified by the following steps:

Algorithm III

i) $v_k = y_k - \hat{y}_k$

$\hat{y}_k = z_{k-1}\,\hat{\underline{\theta}}_{k-1}$

$z_{k-1} = [\hat{x}'_{k-n};\ u_{k-1}, u_{k-2}, \ldots, u_{k-n}]$

$\hat{\underline{\theta}}'_{k-1} = [\hat{a}_1, \ldots, \hat{a}_n;\ \hat{b}_1, \ldots, \hat{b}_n]$

ii) a RLS estimator with variable forgetting factor is used to update the parameter estimates vector $\underline{\theta}$.

$w_k = z_{k-1}\,P_{k-1}\,z'_{k-1}$

$n_k = 1 - w_k - v^2_k/\Sigma_o$

$\lambda_k = (n_k + \sqrt{n^2_k + 4 w_k}\,)/2$

$\underline{K}_k = P_{k-1}\,z'_{k-1}/(\lambda_k + w_k)$

$\underline{\theta}_k = \underline{\theta}_{k-1} + \underline{K}_k\,v_k$

$P_k = [P_{k-1} - \underline{K}_k\,\underline{K}'_k\,(\lambda_k + w_k)]/\lambda_k$

P_k = covariance matrix

Σ_o = information content

λ_k = forgetting factor

ii) the states are obtained from :

$\hat{\underline{x}}_{k/k-1} = \hat{A}_k\,\hat{\underline{x}}_{k-1} + \hat{\underline{b}}_k\,u_{k-1}$

$\hat{\underline{x}}_k = \hat{\underline{x}}_{k/k-1} + \gamma\,\underline{c}'\,(y_k - \underline{c}\,\hat{\underline{x}}_{k/k-1})$

with γ a small constant

iii) the control law is then computed by:

$u_k = (y^*_{k+T} - \overline{Y}_k)/d$

$\overline{Y}_k = \underline{c}\,\hat{A}^T_k\,\hat{\underline{x}}_k$

$d = \underline{c}\,\Sigma_{j=0}^{T-1}\hat{A}^{j}_{k}\,\hat{\underline{b}}_{k}$

iv) goto (i)

3.2.2 The Multivariable Case

The regulator of section (3.2.1) will be extended to multivariable plants. Only the case of an equal number of inputs and outputs will be studied.

First, the parameter estimation problem will be considered. In this case the multivariable system is decomposed into a set of multi-input single-output (MISO) subsystems, such that the parameter estimation can be done sequentially for each subsystem using a single-variable RLS estimator as described in section (3.2.1).

The decomposition of the MIMO system and the canonical model used in this section are further described in Appendix C.

After the parameter estimates are obtained, the states are computed in a similar way as in eq. (3.32), but a different filter gain is used for each MISO system.

Finally the extended-horizon control strategy is extended to consider a different control-horizon for each output.

For example, the equivalent to the state equation (3.28) for multivariable systems is:

$$\bar{x}_{k+1} = \bar{A}\bullet\bar{x}_k + \bar{B}\bullet u_k + \bar{K}\bullet e_k \qquad (3.42.a)$$

$$y_k = \bar{C}\bullet\bar{x}_k + e_k \qquad (3.42.b)$$

where $\bar{x} \in R^n$ is the state vector and n is the order of the system. $u \in R^m$ and $y \in R^m$ are the input and output vectors respectively. The innovation vector $e \in R^m$ is defined as the difference between the measured output vector y_k and the predicted output vector $y_{k/k-1}$.

The process matrix A, the control matrix B, the Kalman [or innovation] gain matrix K, and the observation matrix C are of adequate dimensions.

If the system described by eqs. (3.42) is observable, the number of parameters which defines the matrices A, B, K, and C, can be reduced by transforming the state model to the row-companion canonical form. The transformed state vector is $x = P\bullet\bar{x}$, where the form of the P matrix is given in Sinha & Rózsa (1976).

The state equation is now:

$$\underline{x}_{k+1} = A \cdot \underline{x}_k + B \cdot \underline{u}_k + K \cdot \underline{e}_k \qquad (3.43.a)$$

$$\underline{y}_k = C \cdot \underline{x}_k + \underline{e}_k \qquad (3.43.b)$$

where $A = P \cdot \bar{A} \cdot P^{-1}$, $B = P \cdot \bar{B}$, and $K = P \cdot \bar{K}$.

In this canonical form the matrices A and C can be expressed in a block manner:

$$A = \begin{vmatrix} A_{11} & \cdots & A_{1m} \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ A_{m1} & \cdots & A_{mm} \end{vmatrix} \qquad C = \begin{vmatrix} \underline{i}(q1) \\ \underline{i}(q2) \\ \cdot \\ \cdot \\ \underline{i}(qm) \end{vmatrix}$$

where the blocks in the A matrix are defined by:

$$A_{ii} = \begin{vmatrix} 0 & 1 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 & 1 \\ a_{ii}(1) & \cdots & a_{ii}(ni) \end{vmatrix} \qquad A_{ij} = \begin{vmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ a_{ij}(1) & \cdots & a_{ij}(nj) \end{vmatrix} \quad \begin{matrix} \uparrow \\ \vdots \\ ni \\ \vdots \\ \downarrow \end{matrix}$$

and the $\underline{i}(qj)$ are unitary row vectors of dimension n with the one on the qj^{th} position, where:

$$n = \sum_{i=1}^{m} ni \quad ; \quad qj = \left(\sum_{i=1}^{j-1} ni \right) + 1$$

The ni's are the invariant parameters, called observability indices, which define the internal structure of the multivariable plant, and should be provided as design parameters.

If the plant is observable and controllable, then the observability indices play a similar role as the model order in SISO systems.

An intuitive interpretation of these parameters can be given; according to Schumann (1979) n_1 is the number of eigenvalues which can be observed by the first output y_1, n_2 is the number of eigenvalues that can be observed by the second output y_2 but cannot be observed by the first output. The next indices can be understood in the same way. That means that if the outputs are ordered in a different way, the form of the matrices A and C will be different.

As in the SISO case, the control delays are not explicitly considered in the model structure.

Estimation

Once the structure of the plant model is defined, it is necessary to estimate the parameters of the plant.

According to the developments of appendix C, the outputs of eqs. (3.45) can also be expressed in the following way:

$$y_k(j) = z_{k-1}(j)\cdot\underline{\theta}(j) + v_k(j) \qquad (3.44)$$

where y(j) and v(j) are the output and the residual of the j^{th} subsystem respectively. The residual v(j) is a linear combination of past and present innovations :

$$v_k(j) = e_k(j) + \sum_{i=1}^{nj} \underline{k}(qj+i-1)\cdot\underline{e}_{k-i} \qquad (3.45)$$

where $\underline{k}(i)$ is the i^{th} row vector of the Kalman gain matrix K, and e(j) is the j^{th} element of the innovations vector.

The j^{th} data vector and the j^{th} parameter vector are:

$$z_{k-1}(j) = [\underline{x}'_{k-nj};\ \underline{u}'_{k-1};.....;\ \underline{u}'_{k-nj}]$$
$$\underline{\theta}'(j) = [\underline{a}(j);\ \underline{b}(qj);.....;\ \underline{b}(pj)]$$
$$\underline{a}(j) = [a_{j1}(1)...a_{j1}(n1)\ a_{j2}(1)...a_{j2}(n2)...a_{jm}(1)...a_{jm}(nm)]$$

the $a_{ij}(l)$ parameters are the elements of the blocks of the A matrix, and the $\underline{b}(j)$'s are the j^{th} row vectors of the matrix B, where pj is given by:

$$pj = \sum_{i=1}^{j} ni$$

From eq. (3.45) it can be seen that the residuals at time k are a linear combination of the innovations from time $k-n_j$ to time k. Then $v_k(j)$ is uncorrelated with the states at time $k-n_j$, consequently a RLS algorithm can be used to get an unbiased estimate of the vector $\underline{\theta}(j)$ at time k based on the states at time $k-n_j$.

As in previous sections, a variable forgetting factor is used for parameter tracking.

The $\underline{\theta}(j)$ vectors are sequentially estimated independently for each MISO system using an ordinary least squares.

Filtering

The state estimates can be updated through the following filter:

$$\hat{x}_{k/k-1} = \hat{A}_k \cdot \hat{x}_{k-1} + \hat{B}_k \cdot u_{k-1} \qquad (3.46.a)$$

$$\hat{x}_k = \hat{x}_{k/k-1} + \Gamma_k \cdot C' \cdot \{\hat{y}_k - C \cdot \hat{x}_{k/k-1}\} \qquad (3.46.b)$$

In the formulation of El-Sherief and Sinha (1979), the gain Γ_k is a scalar sequence. However taking into account that the innovation elements in the vector $\underline{e}$ can have different statistical properties, it seems a good idea to use a different gain sequence for each subsystem.

In eq. (3.46.b) the gain Γ_k is a diagonal matrix. Moreover, because of the special form of the matrix C, the matrix $G = \Gamma_k \cdot C'$ is an nxm matrix with only m non-zero elements. These $\gamma_k(j)$ non-zero elements are the filter gains associated with the outputs y(j). Each of these filter gains can be selected according to the methods discussed in section (3.2.1).

In our simulations and experiments a constant gain filter algorithm was used for simplicity. However it can be easily extended to consider adaptive gains.

Control

An extended-horizon control strategy is developed for multivariable plants. First it is assumed a single control-horizon for all the outputs, then the strategy is generalized to consider an independent control-horizon for each output.

To develop the control it is first necessary to obtain a T-step ahead state predictor:

$$\hat{x}_{k+T} = \hat{A}^T_k \cdot \hat{x}_k + \left[\sum_{j=0}^{T-1} \hat{A}^j_k\right] \cdot \hat{B}_k \cdot u_{k+j} \qquad (3.47)$$

and by eq. (3.42.b), the optimal predictor for the output vector is:

$$\hat{y}_{k+T} = \bar{Y}_k + \left\{C \cdot \left[\sum_{j=0}^{T-1} \hat{A}^j_k\right] \cdot \hat{B}_k\right\} \cdot u_{k+j} \qquad (3.48)$$

where $\bar{Y}_k = C \cdot \hat{A}^T_k \cdot \hat{x}_k$ represents the knowledge at time k.

If the control objective is to drive the expected T-step ahead predicted output error to zero, then the control is given by:

$$y^*_{k+T} - \bar{Y}_k = \{C \cdot [\sum_{j=0}^{T-1} \hat{A}^j{}_k] \cdot \hat{B}_k\} \cdot u_{k+j} \qquad (3.49)$$

where y^*_{k+T} is the vector of set points at time k+T.

Infinite vector input sequences can satisfy eq. (3.49). Some simple and reasonable strategies have been discussed earlier [see section (3.1.1)].

For simplicity, a constant control strategy over the horizon T is used. In this case the control can be computed from:

$$u_k = D^{-1} \cdot \{ y^*_{k+T} - \bar{Y}_k \} \qquad (3.50)$$

where

$$D = C \cdot [\sum_{j=0}^{T-1} \hat{A}^j{}_k] \cdot \hat{B}_k$$

It has been shown by Ydstie (1982) that for a controllable, observable and open-loop stable plant, the matrix D can be made always invertible by selecting a long enough horizon T.

The control eq. (3.50) is similar to the one proposed by Dugard et al (1984), but they use a strategy of minimum control effort over the horizon T.

A single horizon control strategy is not good enough for multivariable processes, because in general each output is affected by a different predominant delay. Then, if a single horizon is used, it must be larger than or equal to the largest delay in the system. Of course, this will unnecessarily retard the response of those outputs with smaller or no delay. It then seems a better strategy to select a specific horizon for each output according to the predominant delay associated with the respective MISO system. This is similar to the idea presented by Lee and Lee (1983), when dealing with long-term predictors. However, in our development the knowledge of the interactor matrix is not required, and only upper bounds of the predominant delays affecting each output are needed.

In the multiple-horizon approach the key idea is to define a predictor for each output y(j) at different time-horizons Tj, where Tj is an integer larger than or equal to the predominant delay associated with y(j).

From eqs. (3.47) and (3.48) it can be seen that a suitable Tj-step ahead predictor for y(j) is:

$$\hat{y}_{k+Tj}(j) = \bar{Y}_k(j) + \underline{i}(qj)\cdot[\sum_{i=0}^{Tj-1}\hat{A}^i{}_k]\cdot\hat{B}_k\cdot u_{k+j} \qquad (3.51)$$

where the $\underline{i}(qj)$ are unitary row-vectors. It was mentioned earlier that these vectors depend on the chosen structural indices n_j.

$\bar{Y}_k(j)$ represents all the known information for the j^{th} subsystem at time k and is given by:

$$\bar{Y}_k = \underline{i}(qj)\cdot\hat{A}^{Tj}{}_k\cdot\hat{X}_k$$

If again a constant control strategy is used, then eq. (3.51) is reduced to:

$$\hat{y}_{k+Tj}(j) = \bar{Y}_k(j) + \underline{d}(j)\cdot u_k \qquad (3.52)$$

with

$$\underline{d}(j) = \underline{i}(qj)\cdot[\sum_{i=0}^{Tj-1}\hat{A}^i{}_k]\cdot\hat{B}_k$$

Repeating the same procedure for each subsystem, a vector of predicted outputs can be obtained:

$$\hat{y}_{k+T} = \bar{Y}_k + D\cdot u_k \qquad (3.53)$$

where:

$$y_{k+T} = \begin{vmatrix} y_{k+T1}(1) \\ y_{k+T2}(2) \\ \vdots \\ y_{k+Tm}(m) \end{vmatrix} \qquad Y_k = \begin{vmatrix} Y_k(1) \\ Y_k(2) \\ \vdots \\ Y_k(m) \end{vmatrix} \qquad D = \begin{vmatrix} \underline{d}(1) \\ \underline{d}(2) \\ \vdots \\ \underline{d}(m) \end{vmatrix}$$

Note that in this case the control is chosen constant over the horizon T, where the horizon T = Max{T1, T2, ..., Tm}.

The predicted output errors vector is then defined by:

$$\underline{err}_{k+T} = \underline{y}^*_{k+T} - \bar{\underline{Y}}_k - D \cdot \underline{u}_k$$

where $\underline{y}^*_{k+T}$ is the vector of the reference values for the outputs y(j) at time k+Tj.

If now the control objective is to drive the expected value of the predicted output errors vector to zero, the control is given by:

$$\underline{u}_k = D^{-1} \cdot \{ \underline{y}^*_{k+T} - \bar{\underline{Y}}_k \} \qquad (3.54)$$

It is assumed that for observable, controllable and stable plants, the matrix D will be non-singular if long enough control horizons are used. In any case, it has been shown that if a single-horizon for all the outputs is chosen long enough, the matrix D can be made invertible [Ydstie, 1982].

As usual, only the first control input of the sequence $\{\underline{u}_i\}_{i=k}^{T-1}$ is computed and applied each sample interval.

The main design parameters in this algorithm are:

a) the time-horizons for each output

b) the order of the MISO subsystems, or equivalently, the structural indices

c) the information content of the RLS estimator for each subsystem

It is also necessary to provide the initial value of the covariance matrix, the initial model parameter estimates and the gains of the states filter, but the performance of the self-tuner is not sensitive to these values.

For example, in this work the values usually employed are:

d) $P_{ii} = 10^{12}$, for all the estimates.

e) $\hat{\underline{\theta}}_0 = \underline{0}$

f) $\Gamma_i = .1$, for all the elements in the innovations vector.

An example of this algorithm is given in Appendix E.

In summary the multivariable self-tuner is given by:

Algorithm IV

i) $v_k(j) = y_k - \hat{y}_k$; $j = 1, \ldots, m$

$\hat{y}_k(j) = z_{k-1}(j) \cdot \hat{\underline{\theta}}_{k-1}(j)$

$z_{k-1}(j) = [\hat{\underline{x}}'_{k-nj};\ \underline{u}'_{k-1};\ \ldots\ldots;\ \underline{u}'_{k-nj}]$

$\hat{\underline{\theta}}'_{k-1}(j) = [\hat{\underline{a}}(j);\ \hat{\underline{b}}(qj);\ \ldots\ldots;\ \hat{\underline{b}}(pj)]$

$\hat{\underline{a}}(j) = [\hat{a}_{j1}(1) \ldots \hat{a}_{j1}(n1)\ \hat{a}_{j2}(1) \ldots \hat{a}_{j2}(n2) \ldots . \hat{a}_{jm}(1) \ldots \hat{a}_{jm}(nm)]$

ii) a RLS estimator with variable forgetting factor is sequentially applied to update the parameter vectors $\hat{\underline{\theta}}(j)$. For j=1 to m:

$w_k = z_{k-1}(j) \cdot P_{k-1}(j) \cdot z'_{k-1}(j)$

$n_k = 1 - w_k - v^2_k(j)/\Sigma_0(j)$

$\lambda_k(j) = (n_k + \sqrt{n^2_k + 4 \cdot w_k})/2$

$\underline{K}_k = P_{k-1}(j) \cdot z'_{k-1}(j)/(\lambda_k(j) + w_k)$

$\hat{\underline{\theta}}_k(j) = \hat{\underline{\theta}}_{k-1}(j) + \underline{K}_k \cdot v_k(j)$

$P_k(j) = [P_{k-1}(j) - \underline{K}_k \cdot \underline{K}'_k \cdot (\lambda_k(j) + w_k)]/\lambda_k(j)$

$P_k(j)$ = covariance matrix

$\Sigma_0(j)$ = information content

$\lambda_k(j)$ = forgetting factor

iii) the states are estimated by the filter:

$\hat{\underline{x}}_{k/k-1} = \hat{A}_k \cdot \hat{\underline{x}}_{k-1} + \hat{B}_k \cdot \underline{u}_{k-1}$

$\hat{\underline{x}}_k = \hat{\underline{x}}_{k/k-1} + \Gamma \cdot C' \cdot (\underline{y}_k - C \cdot \hat{\underline{x}}_{k/k-1})$

the matrix $G = \Gamma \cdot C'$ contains m gains $\gamma(j)$ that are selected as small constants.

iv) the control is computed from:

$\underline{u}_k = D^{-1} \cdot (\underline{y}^*_{k+T} - \overline{\underline{Y}}_k)$

where the rows of the matrix D are:

$$\underline{d}(j) = \underline{i}(qj) \cdot \left[\sum_{i=0}^{Tj-1} \hat{A}^i_k\right] \cdot \hat{B}_k \qquad ; j = 1, \ldots, m$$

and the elements of $\overline{\underline{Y}}_k$ are:

$$\overline{Y}_k(j) = \underline{i}(qj) \cdot \hat{A}^{Tj}_k \cdot \hat{\underline{x}}_k \qquad ; j = 1, \ldots, m$$

v) Go to (i)

3.2.3.- A Feedback-Feedforward Multivariable Self-Tuner

In this section a multivariable regulator is developed which considers feedforward as well as feedback action. The regulator possess the same structure as the previous self-tuner, but a measurable disturbance appears in the model.

The innovations form of the state equations are modified to incorporate the measurable disturbance:

$$\underline{x}_{k+1} = A\cdot\underline{x}_k + B\cdot\underline{u}_k + H\cdot\underline{w}_k + K\cdot\underline{e}_k \qquad (3.55.a)$$

$$\underline{y}_k = C\cdot\underline{x}_k + \underline{e}_k \qquad (3.55.b)$$

where $\underline{w} \in R^m$ is the vector of measured disturbances. $H \in R^n xR^m$, the disturbance matrix, is full like matrices B and K in eq. (3.42).

The rest of the notation corresponds with eq. (3.42), and it is assumed that eqs. (3.55) are in row-companion canonical form.

To get the estimates of the matrices' parameters, the outputs y(j) in eq. (3.55.b) can be expressed [following the same procedure as in Appendix C] by:

$$y_k(j) = z_{k-1}(j)\cdot\underline{\theta}_j + v_k(j) \qquad (3.56)$$

with :

$$z_{k-1}(j) = [\underline{x}'_{k-nj};\ \underline{u}'_{k-1},\ldots;\ \underline{u}'_{k-nj};\ \underline{w}'_{k-1},\ldots,\underline{w}'_{k-nj}]$$

$$\underline{\theta}'(j) = [\underline{a}(j);\ \underline{b}(qj),\ldots,\ \underline{b}(pj);\ \underline{h}(qj),\ldots,\ \underline{h}(pj)]$$

the $\underline{h}(j)$'s are the row-vectors of the H matrix.

The residual is given by:

$$v_k(j) = e_k(j) + \sum_{i=1}^{nj} \underline{k}(qj+i-1)\cdot\underline{e}_{k-i} \qquad (3.57)$$

As in eq. (3.44), the residual v_k is uncorrelated with the state vector $\underline{x}_{k-nj}$, then a RLS estimator gives an unbiased estimate of $\underline{\theta}(j)$ at time k, based on states at time k-nj.

In the same way as in section (3.2.2), an RLS estimator with variable forgetting factor is sequentially used to update the vectors $\hat{\underline{\theta}}(j)$.

Using the parameter estimates $\hat{\underline{\theta}}_k(j)$, the states can be obtained by the following filter:

$$\hat{\underline{x}}_{k/k-1} = \hat{A}_k \cdot \hat{\underline{x}}_{k-1} + \hat{B}_k \cdot \underline{u}_{k-1} + \hat{H}_k \cdot \underline{w}_{k-1} \qquad (3.58.a)$$

$$\hat{\underline{x}}_k = \hat{\underline{x}}_{k/k-1} + \Gamma_k \cdot C' \cdot (\underline{y}_k - C \cdot \hat{\underline{x}}_{k/k-1}) \qquad (3.58.b)$$

where the gain matrix Γ_k is defined in eq. (3.48.b)

The control strategy is to drive the expected value of the predicted output error to zero.

As in eq. (3.50) the predicted output y(j) Tj-step ahead is:

$$\hat{\underline{y}}_{k+Tj}(j) = \bar{Y}_k(j) + \underline{i}(qj) \cdot [\sum_{i=0}^{Tj-1} \hat{A}^i_k] \cdot \hat{B}_k \cdot \underline{u}_{k+j} + \underline{i}(qj) \cdot [\sum_{i=0}^{Tj-1} \hat{A}^i_k] \cdot \hat{H}_k \cdot \underline{w}_{k+j} \qquad (3.59)$$

and

$$Y_k(j) = \underline{i}(qj) \cdot \hat{A}^{Tj}_k \cdot \hat{\underline{x}}_k$$

Equation (3.59) includes future values of the control and the disturbance vectors. Several assumptions can be made about future disturbances [see section (3.1.2)], and also infinite combinations of the input control sequence can comply with the control objective [see section (3.1.1)].

To simplify, it is assumed that the control inputs remain constant during the horizon T, where T=Max{T1, T2, ..., Tm}, and the disturbances change by steps and remain constant for long periods.

If this is the case, then eq. (3.59) can be reduced to:

$$\hat{\underline{y}}_{k+Tj}(j) = \bar{Y}_k(j) + \underline{d}(j) \cdot \underline{u}_k + \underline{s}(j) \cdot \underline{w}_k \qquad (3.60)$$

with:

$$\underline{d}(j) = \underline{i}(qj) \cdot [\sum_{i=0}^{Tj-1} \hat{A}^i_k] \cdot \hat{B}_k$$

$$\underline{s}(j) = \underline{i}(qj) \cdot [\sum_{i=0}^{Tj-1} \hat{A}^i_k] \cdot \hat{H}_k$$

And considering all the outputs:

$$\hat{\underline{y}}_{k+T} = \bar{Y}_k + D \cdot \underline{u}_k + S \cdot \underline{w}_k \qquad (3.61)$$

where S is defined by:

$$S' = [\underline{s}'(1),\ \underline{s}'(2),\ .....,\ \underline{s}'(m)]$$

The control objective is then satisfied by the following input:

$$\underline{u}_k = D^{-1} \{ \underline{y}^*_{k+T} - \overline{\underline{Y}}_k - S \underline{W}_k \} \qquad (3.62)$$

It is again assumed that D can be made non-singular for stable, controllable and observable plants, if the horizons Tj are chosen long enough.

The control (3.62) works in a receding horizon manner; namely only the first control of the sequence over the horizon is computed and applied each time interval.

No other design parameter is required in this algorithm, so that it only needs:

a)time-horizons
b)structural indices
c)information contents
d)initial covariance matrix
e)initial parameter estimates
f)filter gains

An example is developed in Appendix E using this algorithm.

Algorithm V

i) $v_k = y_k - \hat{y}_k$; j = 1, ..., m

$\hat{y}_k(j) = z_{k-1}(j) \cdot \hat{\theta}_k(j)$

$z_{k-1}(j) = [\hat{x}'_{k-nj};\ u'_{k-1}, \ldots;\ u'_{k-nj};\ w'_{k-1}, \ldots,\ w'_{k-nj}]$

$\hat{\theta}'_k(j) = [\hat{a}(j);\ \hat{b}(qj), \ldots,\ \hat{b}(pj);\ \hat{h}(qj), \ldots,\ \hat{h}(pj)]$

ii) a RLS estimator with variable forgetting factor updates the vectors $\hat{\theta}(j)$.

For j=1 to m:

$w_k = z_{k-1}(j) \cdot P_{k-1}(j) \cdot z'_{k-1}(j)$

$n_k = 1 - w_k - v^2_k(j)/\Sigma_0(j)$

$\lambda_k(j) = (n_k + \sqrt{n^2_k + 4 \cdot w_k})/2$

$K_k = P_{k-1}(j) \cdot z'_{k-1}(j)/(\lambda_k(j) + w_k)$

$\hat{\theta}_k(j) = \hat{\theta}_{k-1}(j) + K_k \cdot v_k(j)$

$P_k(j) = [P_{k-1}(j) - K_k \cdot K'_k \cdot (\lambda_k(j) + w_k)]/\lambda_k(j)$

$P_k(j)$ = covariance matrix

$\Sigma_0(j)$ = information content

$\lambda_k(j)$ = forgetting factor

iii) the states are given by:

$\hat{x}_{k/k-1} = \hat{A}_k \cdot \hat{x}_{k-1} + \hat{B}_k \cdot u_{k-1} + \hat{H}_k \cdot w_{k-1}$

$\hat{x}_k = \hat{x}_{k/k-1} + G \cdot (y_k - C \cdot \hat{x}_{k/k-1})$

where G contains m constant filter gains $\gamma(j)$.

iv) control law:

$u_k = D^{-1} \cdot (y^*_{k+T} - \bar{Y}_k - S \cdot w_k)$

the row-vectors of the matrices D and S are given by:

$$d(j) = i(qj) \cdot [\sum_{i=0}^{Tj-1} \hat{A}^i_k] \cdot \hat{B}_k \qquad ; j = 1, \ldots, m$$

$$s(j) = i(qj) \cdot [\sum_{i=0}^{Tj-1} \hat{A}^i_k] \cdot \hat{H}_k \qquad ; j = 1, \ldots, m$$

and the elements of the vector Y by:

$$\bar{Y}_k(j) = i(qj) \cdot \hat{A}^{Tj}_k \cdot \hat{x}_k \qquad ; j = 1, \ldots, m$$

v) goto (i)

CHAPTER 4 EXPERIMENTAL SYSTEM

4.1 The Process

A CO_2 absorption/desorption pilot plant (see Fig. 4.1) was used to test some of the proposed adaptive algorithms.

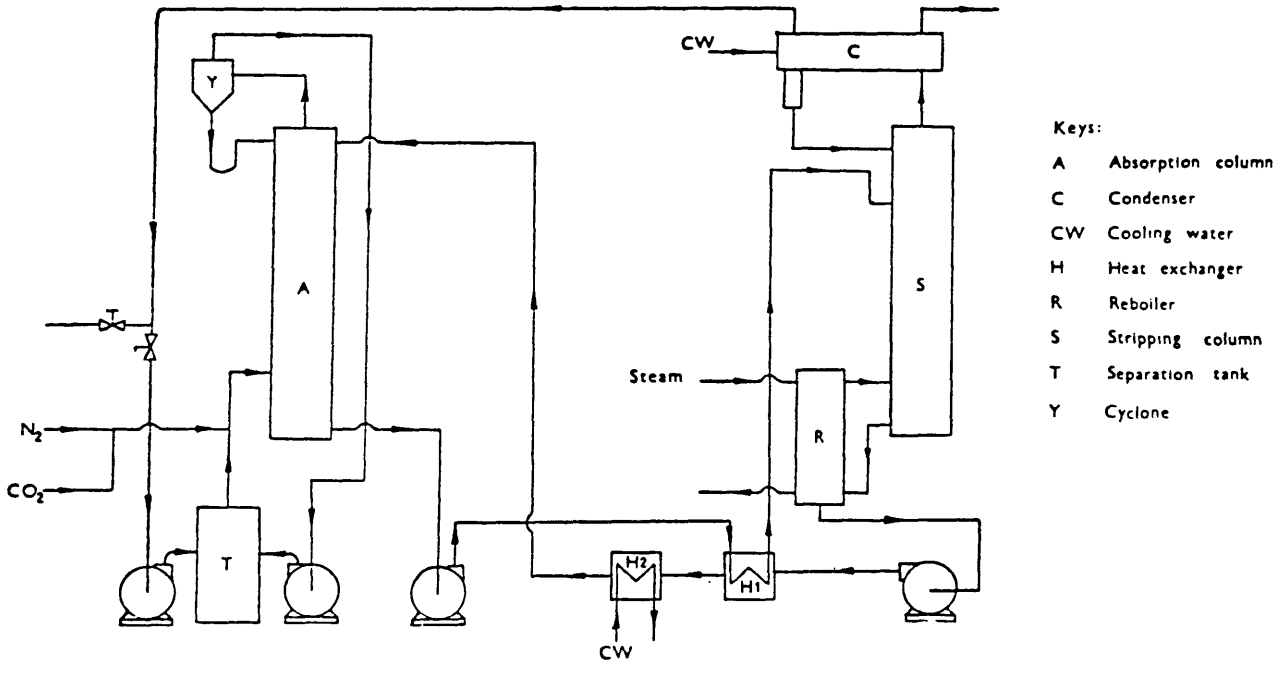


Fig. 4.1 Absorption/Desorption Pilot Plant

Purification of large volumes of sour gas requires a process that is efficient and relatively low in cost. The alkanolamine process has emerged as one of the most commonly used industrial methods. It is based on the reaction of a weak base (alkanolamine) and a weak acid (CO_2 or H_2S) to give a water-soluble, relatively unstable, salt.

The process is carried out in two columns. In one column the sour gas is contacted with the alkanolamine solution, where the CO_2 or H_2S is absorbed. In the other column the solution is boiled and freed of the CO_2 or H_2S, and then fed again to the first column. In this particular pilot plant a gaseous stream of 10% CO_2 in nitrogen is fed into

the base of the absorber and flows upward, countercurrrent to an aqueous monoethanolamine (MEA) solution, which reacts with and absorbs the CO_2. The absorber is a packed column (25 cm diameter, 9 m high) which operates at 2-4 bar pressure and 30-50°C. The exit solution, rich in CO_2, is pumped from the bottom of the absorber through a heat exchanger where it is heated to approximately 70-90°C by the hot solution coming from the desorber. The amine solution is then sent to the upper section of the regenerator and flows downward, contacting the hot vapours from the reboiler, and liberating the CO_2. The desorption column (25 cm diameter) which has 20 sieve trays operates at 0.7-1.3 bar and 90-110°C. These conditions suffice to remove most of the CO_2. The gas liberated from the amine solution contains a large portion of water vapour which is removed by passing the hot vapour mixture through a condenser. The condensate is recycled to the desorber. The liberated CO_2 is normally vented or sent on for further processing. In our plant, in order to economize on material, the CO_2 can be fed again to the system rather than vented. The hot reconcentrated solution flows from the reboiler to a pair of heat exchangers. Here it is first cooled by the incoming cold rich solution and further cooled by water. The cool amine solution is then pumped back to the absorber, thus completing the cycle.

The Control Loops used in the Experiments

1) The cooling water flow rates of the heat exchanger and the condenser were used in several experiments. These are simple, safe and non-interacting loops suitable for testing multivariable controllers and performing other unusual tests such as start-up, large deterministic changes and overnight runs.

2) Similarly the level of the liquid in the absorber was used to test the robustness of the proposed algorithms under unknown and variable time delays. Normally the level of the liquid in the bottom of the absorption column is controlled by the outlet valve (configuration #1 in Fig. 4.2), in this case the time delay between the control and the output is less than 10 seconds. In addition, an artificially difficult configuration was constructed, where the level is controlled by the inlet valve (configuration #2). This abnormal arrangement introduced a large time delay resulting from the trickling flow down the packed column. Furthermore, the time delay is highly dependent on the liquid flow rate, the gas flow rate up the column, and the previous state of the column. This delay can vary between 30 and 90 seconds.

3) Other workers have studied control of the column pressures and exit gas composition.

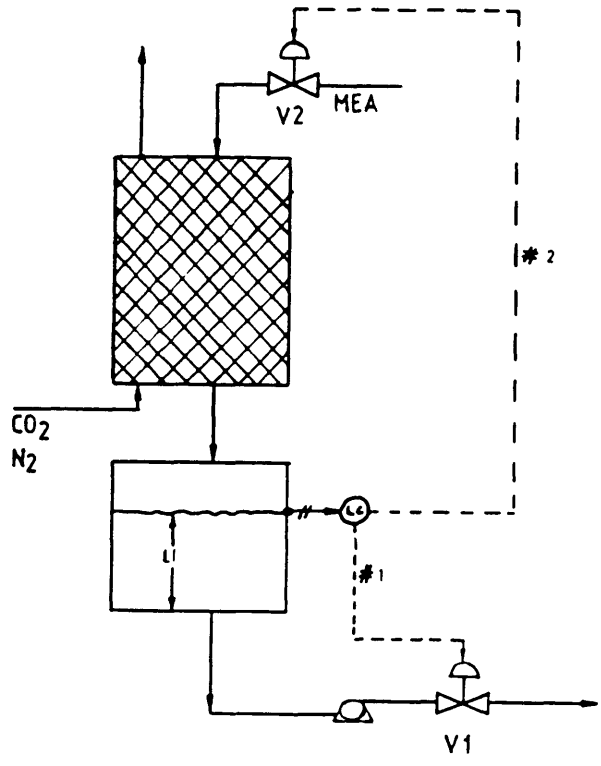


Fig. 4.2 Level Control Configurations

Instrumentation

The flow rates studied are measured by standard orifice plates. The pneumatic signal is converted by the P/I transducer into a 4-20 mA electric signal. The liquid level is measured using a differential pressure cell and transduced to 4-20 mA as above. The control valves are conventional pneumatically driven plug valves. To reduce the travel time of the valve stem, the valves are equipped with pneumatic boosters and, in certain critical cases, valve positioners.

The Control Implementation

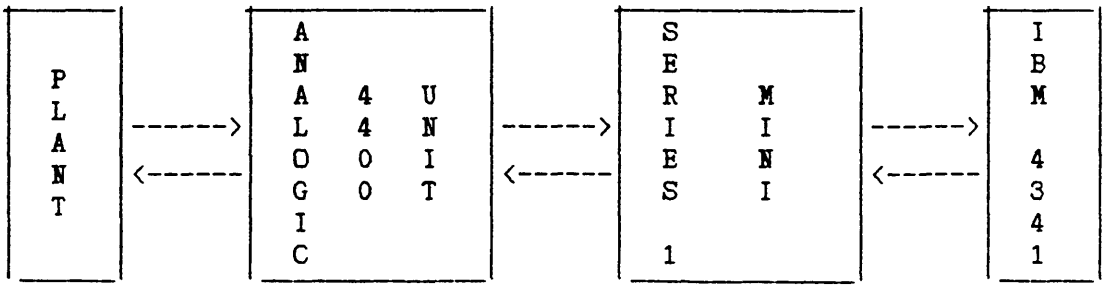


Fig. 4.3 Hardware Configuration

An Analogic 4400 unit receives and sends information to the plant (see fig. 4.3), and it also does the analogue/digital conversion, i.e., it converts all the plant measurements to a suitable digital form, sends the required analogue signals to the control valves and sends digital signals to pumps and motors. A series 1 minicomputer links the Analogic unit with the mainframe computer (IBM 4341), where all of the control calculations are performed.

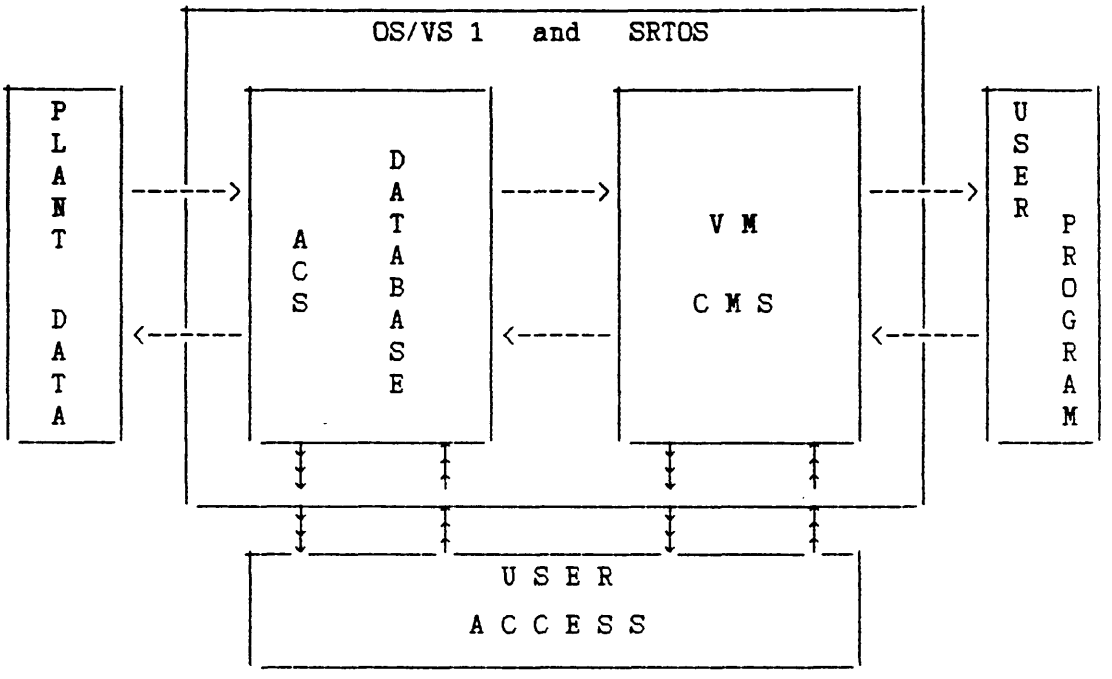


Fig. 4.4 Software Configuration

A special control package called ACS (Advanced Control System) enables the monitoring and control of the plant with built-in standard strategies. The plant can also be controlled from VM/CMS operating system (which is the usual environment for running programs in IBM computers) using an interface with ACS, which allows the engineer to implement any algorithm written in FORTRAN, PL/1 or ASSEMBLER.

The entire process is controlled using the standard IBM operating system (OS/VS1), and a special extension SRTOS (Special Real Time Operating System) which provides the real-time data management.

4.2. The ACS Package

ACS is a package for on-line control of different kinds of processes. Basically ACS consists of a real time database and a processor.

The database is continuously fed with information from the plant, and this information is available to the user at any moment through special displays. This information is called 'variables' in ACS, and there are several types of variables. Each variable has associated attributes called 'items'. The collection of variables and its items form the core of the ACS database.

The processor is formed by a group of programs which operate with the variables of the database, modifying some plant variables and/or the database itself. Among the programs, ACS provides built-in standard control strategies (PID, ratio control, feedforward, cascade, etc.) which the engineer can directly use. If the user needs to do some special calculations, they can be done using a language called GA (General Algorithm), which resembles a pocket calculator programming language, but with limited editing facilities.

For developing more advanced algorithms it is easier to use an interface facility, which enables the engineer to program directly in FORTRAN or PL/1.

The Variables in ACS

Variables are assigned names using a ten-character 'tag'. The tag has three components, the unit identifier, which defines the process associated with the variable; the service identifier, which defines the kind of measurement and a number to completely specify the variable. For example, 'AD V 001' is the valve # 1 of the Absorption Plant, 'XL F 007' is the flow # 7 of the Crystallization Plant, and 'CR T 005' is the temperature # 5 of the Catalytic Reactor. So that AD, XL and CR are unit identifiers; V, F, T are service identifiers; and 001, 007 ,005 are specification numbers.

The most important types of variables in ACS are the full value variables (FVV) and the miscellaneous variables.

The FVV's are directly related with the plant (measurements and manipulated variables) and they have associated approximately 60 different items. The most commonly used items are the current value (item 10), the set point (item 40), and the output (item 80).

The miscellaneous variables have associated 2 items, the current value (item 10) and the initial value (item 11). These variables are mainly used to store and retrieve various results of computations, they can also be used to provide parameters in advanced control algorithms.

Interaction with ACS

The interaction with ACS is based on special tables and displays. The tables are mainly used to define variables and control strategies. The direct access to the database is through displays, where the user can see the value of any variable and change it, when it is possible. Some basic displays will be described as a preliminary guide into ACS for those users interested in FORTRAN implementations.

a) The operator displays : Typing 'UN 0' the user can get a list of all the displays associated with the process unit identified by 'UN'.
b) The engineers menu : There is a higher level menu called by typing 'XX0', which contains a list of the displays that allow the engineer to do more fundamental changes in the system. There are two of these displays which are basic for any FORTRAN implementation in ACS.
c) The XX3 (USER PROGRAM LIBRARY) : It contains a summary of all the programs callable by ACS or the operator. From this display the engineer can change the status of the program from 'TEST' to 'RUN' mode and vice versa (see User Interface, section 4.3).
d) The XX9 (DISPLAY/VARIABLE DEFINITION MENU) : This is probably the most important xx display. Here the engineer can define variables, control structures, displays, etc. .

4.3. The User Interface

ACS provides an interface which allows FORTRAN, PL/1, and ASSEMBLER programs to update the real time database.

This section will describe a method of calling user written FORTRAN programs via the Database Server which is one implementation of the ACS interface.

The FORTRAN programs should be coded according to a certain structure. An example is given in Fig. 4.4. The interface routines that appear in Fig. 4.4 are explained below, and they are grouped in alphabetical order.

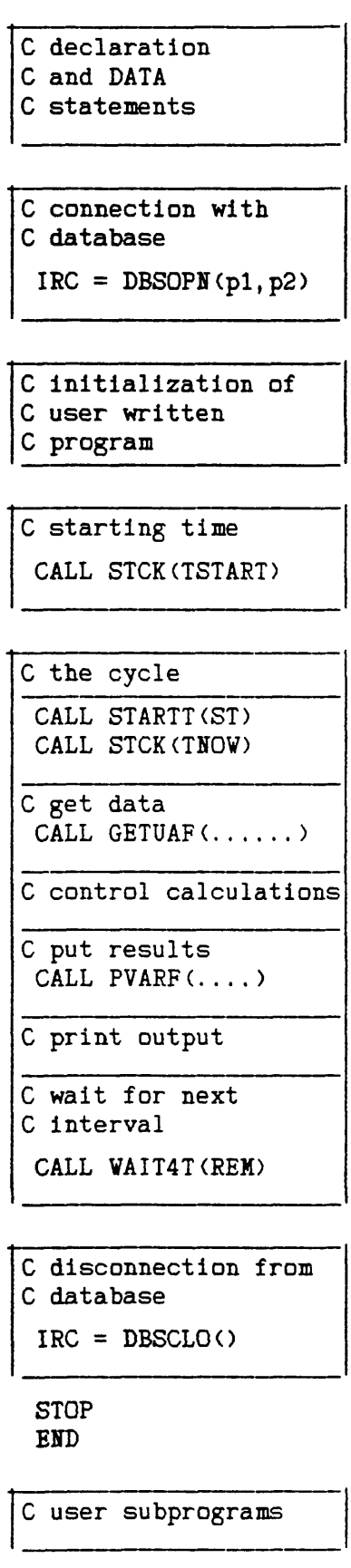


Fig. 4.4 A Possible Structure for a Database Server FORTRAN Program.

Interface Routines

FUNCTION DBSCLO()

This function disconnects the FORTRAN program from the database, and should be called once only at the end of the program before the STOP statement.

FUNCTION DBSOPN(VS1NAM,JOBNAM)

This function connects the program with the database and should be called once only where the program is initialized before any request to the database. The parameters VS1NAM and JOBNAM are CHARACTER*10 variables, and should be provided by the ACS operator. These are user identification codes.

SUBROUTINE GETUAF(IRC,UAT,VALIN,IRETC)

The simplest way to retrieve information from the ACS database, is through special arrays called User Access Tables (UAT). The UAT's are defined via the File Builder (XX9) using a special form. This special form should be filled with a list of the variables (and its item numbers) to be retrieved. The UAT's do not allow more than ten variables, but there is no limitation on the number of UAT's used.

The GETUAF routine gets values from the database by means of the UAT's.

IRC is the return code and it must be defined as INTEGER*4. If IRC is different from zero, it means that the operation was unsuccessful.

UAT is the User Access Table number as defined in the File Builder form in ACS.

VALIN is the array that will receive the values of the UAT variables. It should be an array of dimension at least equal to the number of values in the UAT. If a mixture of real and integer values is expected, the EQUIVALENCE statement must be used.

IRETC defines the array of return codes for each retrieved data. It must be defined as INTEGER*4, and dimension 10. A successful operation return a value of IRETC equal zero for each data.

For example, if the current value and the set point of the flow # 8 of the Absorption Plant are needed, a typical call can be:

```
CALL GETUAF(IRC,19,VALIN,IRETC)
YF8 =VALIN(1)
SPF8=VALIN(2)
```

the UAT # 19 is shown in fig.4.5. The values are received in engineering units.

```
FORMUA01                    UA  19

                          USER ACCESS TABLE
00                      TABLE DEFINITION (T-Define Table. X-Delete Table)
01  AD  F  008    10    VARIABLE NAME; ITEM NUMBER
02  AD  F  008    40    Variable Name; Item Number
03  AD  V  010    40    Variable Name; Item Number
04  AD  V  010    10    Variable Name; Item Number
05  AD  L  996    10    Variable Name; Item Number
06  AD  L  995    10    Variable Name; Item Number
07  AD  L  989    10    Variable Name; Item Number
08  AD  L  937    10    Variable Name; Item Number
09  AD  L  939    10    Variable Name; Item Number
10  AD  L  938    10    Variable Name; Item Number
```

Fig. 4.5 Example of an User Access Table Display

SUBROUTINE PVARF(IRC,TAGOUT,TAGNUM,ITMOUT,ITMNUM,VALOUT,IRETC,IPROG)

This routine put a list of values in the database. The result of the call could be a mixture of successful and unsuccessful operations, so it is convenient to check the return codes for each output.

IRC and IRETC are the same as in GETUAF. IPROG is ignored.

TAGOUT is a CHARACTER*10 array which contains the tag names of the values to be changed.

TAGNUM is the dimension of the TAGOUT array, and should be an INTEGER*4 variable. The maximum value is 50.

ITMOUT is an INTEGER*4 array that contains the item numbers of the values to be changed.

ITMNUM is the dimension of the ITMOUT array, and is also an INTEGER*4 variable.

VALOUT is the array of the values of the variables to be put in the database. It should be defined as REAL*4, and with a suitable dimension.

In case one needs to change the set points of the valves V10 and V6 to control the flows F8 and F6 of the Absorption Plant, a typical call can be:

```
TAGOUT(1)='AD  V  010'
TAGOUT(2)='AD  V  006'
ITMOUT(1)=40
ITMOUT(2)=40
VALOUT(1)=SPV10
VALOUT(2)=SPV6
CALL PVARF(IRC,TAGOUT,2,ITMOUT,2,VALOUT,IRETC,IPROG)
```

SUBROUTINE STARTT(ST)

This routine initiate the cycle time, where ST is a real variable that represents the sample time in seconds. This routine should be called at the beginning of each cycle.

SUBROUTINE STCK(TOD)

STCK will return the CPU Time-of-Day clock in seconds. The value returned can be used to calculate time intervals. TOD is a REAL*8 variable.

SUBROUTINE WAIT4T(REM)

It waits till the time set by STARTT expires. It must be called at the end of each cycle. REM is a REAL*4 variable which contains the amount of time interval left when WAIT4T was called. The computing time per interval can be easily calculated using REM.

There are to ways of getting out of the cycle. The simplest one is by just pressing 'HX' in the CMS terminal. The other one which provides more flexibility, consists on defining a miscellaneous variable in ACS that is read from the FORTRAN program each interval. This variable can be changed by the user from an ACS terminal. The program will end the cycle according to the value of this variable, by diverting the program to the DBSCLO call.

After compilation and loading under CMS the program can run in real time. This means that the engineer can use all the debugging facilities of CMS, store the data directly in CMS, use all the libraries in CMS and even interact while the program is running.

If the Database Server is set to 'TEST' mode (XX3 display in ACS), the programs running under real time CMS can only read from the ACS database but will be unable to alter it; the results are automatically sent to the printer. In that way the user can check the correct operation of the program. Once the Database Server is in 'RUN' mode the database will be updated.

One of the problems associated with the Database Server is related with synchronization. VM/CMS is not reliable enough to guarantee scheduling on less than 2 seconds sampling interval. This uncertainty depends on the mainframe's load, but varies around +/- 0.5 secs. In our case this fact was unimportant, since the smallest cycle time for user-written programs was 6 seconds. A more serious problem arises from the desynchronization between the ACS sample interval and the cycle interval defined in the FORTRAN program. For example, if the variables are sampled each 4 seconds, and the control implementation rate is 10 seconds, it means that the real control interval is variable between 10 and 14 seconds, introducing an undesirable non-linearity in the system. So it is a good practice to use cycle times in the FORTRAN program which are multiples of the sampling rate used by ACS.

It also seems that ACS use fewer bits in its number representation than does VM/CMS; hence it is advisable to scale the inputs and outputs to the same order of magnitude in order to reduce numerical problems.

CHAPTER 5 RESULTS AND DISCUSSION

Results obtained in simulations and pilot-plant experiments, using a range of adaptive algorithms, are reported and discussed in this Chapter. These algorithms are presented in Chapter 3 and Appendix I.

The chapter is divided into two main sections. The first one focuses on SISO transfer function based algorithms and the second deals with MIMO state-space self-tuners. In both cases, feedback and feedback-feedforward controllers are tested.

The merits of the different adaptive algorithms are compared on the basis of their robustness against unmeasured deterministic disturbances, time-delays, interaction [in the multivariable case] and selection of initial parameters like model orders and time-horizons.

Further improvements to the proposed algorithms are discussed.

5.1.- Polynomial Algorithms

This section is divided into two parts. The first part discusses and compares different ways of implementing feedback STR algorithms using simulations and experiments. The comparison of the different strategies is focused on their robustness against unmeasured deterministic disturbances.

The second part studies the performance of an incremental feedback-feedforward [FB/FF] self-tuning regulator [STR] in systems characterized by time-delays. Different model orders and time-horizons are used.

5.1.1.-Feedback self-tuning Control

As mentioned in Chapter 2, there are different ways of implementing a self-tuner. Positional or incremental and direct or indirect self-tuners can be used. In the following sub-section, these approaches are discussed and compared using simulations and experiments. The basis of comparison is the ability of these algorithms to cope with unmeasured deterministic disturbances.

Other important issues like control strategy selection and bias estimation are also treated in this sub-section.

Positional vs Incremental Formulation

Offset is a typical problem that arises when self-tuning regulators are implemented in real plants. Offset is the steady-state deviation of the output of the controlled process from the reference value. It is mainly the result of the intrinsic non-linearity of most real processes and of unmeasured deterministic disturbances. Other sources of offset are discussed by Clarke et al (1983).

Various solutions for the elimination of offset have been proposed [Clarke et al (1983)], but only two are commonly used in real-time applications:

a) positional formulation with a bias term included

b) incremental formulation

In the present study, a range of experiments have been carried out with each of the two methods.

a) The positional formulation method, which is also known as the "one in the data vector" method, consists in a positional model which estimates a bias parameter along with the rest of the model parameters:

$$\bar{A}(z^{-1})\cdot y = \bar{B}(z^{-1})\cdot y + \bar{\delta} \qquad (5.1)$$

In terms of parameter estimation for extended-horizon, the data vector $\underline{\bar{\varphi}}$ and the parameters vector $\underline{\bar{\theta}}$ are given by:

$$\underline{\bar{\varphi}}_k = [y_k,\ y_{k-1},\ \ldots,\ y_{k-n+1};\ u_{k+T-1},\ \ldots,\ u_{k-n+1};\ 1]$$

$$\underline{\bar{\theta}}' = [\bar{\alpha}_1,\ \bar{\alpha}_2,\ \ldots,\ \bar{\alpha}_n;\ \bar{\beta}_1,\ \bar{\beta}_2,\ \ldots,\ \bar{\beta}_{T+n-1};\ \bar{\delta}]$$

In Appendix I, the method is described in detail for the extended-horizon control strategy.

Clarke and collaborators (1983) pointed out a series of disadvantages with this method, which coincide with our own experience.

It may be observed that the nature of $\overline{\delta}$ is different from the rest of the parameters in the $\underline{\overline{\theta}}$ vector. This is because the changes in the bias parameter are caused by load disturbances and non-linearities, while the changes in the rest of the model parameters are mainly caused by non-linearities. Based on this difference, the rate of change of the bias parameter $\overline{\delta}$ would be expected to be much higher than that of the other parameters, as the plant would normally experience deterministic disturbances. Contrary to this expectation, however, when the "one in the data vector" method [with a variable forgetting factor RLS algorithm] is used to estimate the $\underline{\overline{\theta}}$ vector, it can be found that [even in a linear system] the $\hat{\overline{\alpha}}$, $\hat{\overline{\beta}}$, and $\hat{\overline{\delta}}$ parameter estimates change at the same rate; and the rate depends on the information content (Σ_0) used. In other words, if the estimator is made sensitive enough to achieve a good load rejection [by choosing a small Σ_0], not only will the bias $\hat{\overline{\delta}}$ parameter estimate change, but, undesirably, so will the rest of the parameter estimates.

According to Clarke, the "one in the data vector" method is effective when the associated δ_k parameter [see Appendix I] is constant, but cannot easily cope with changes in the load level.

To illustrate this phenomenon, a first order system was simulated and controlled using a SISO self-tuning regulator. The estimation was carried out using the "one in the data vector".

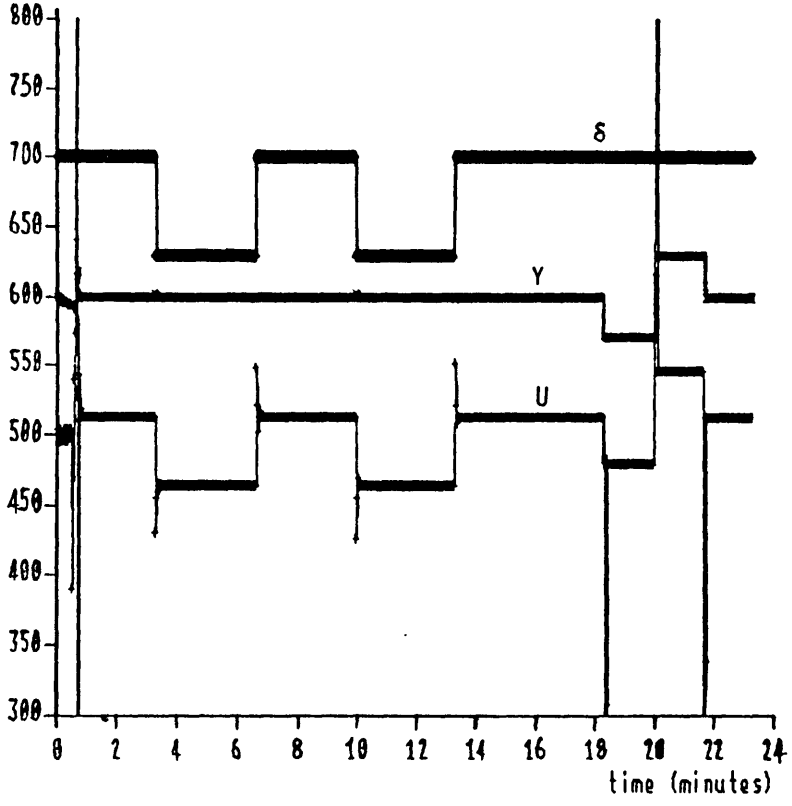
Figure (5.1) shows a typical simulation result when the positional algorithm is used to control systems subjected to frequent load changes. In this particular case, the following first order linear model was used in the simulation:

$$y_k = .951 \cdot y_{k-1} + .439 \cdot u_{k-1} - .212 \cdot \delta_{k-1} + 28.09 \qquad (5.2)$$

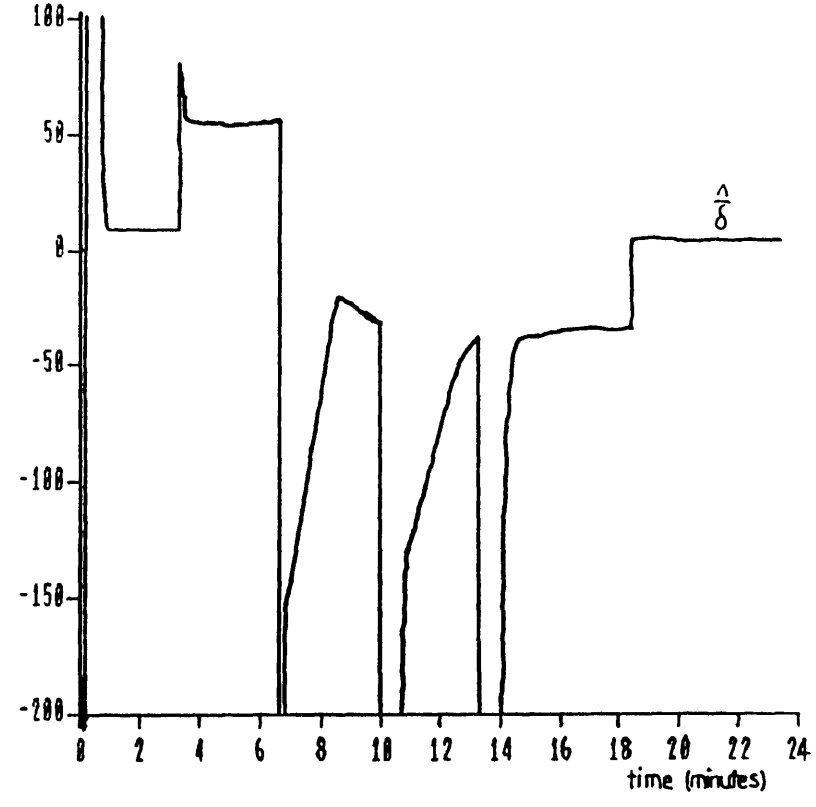
The system was assumed to be second order, the information content Σ_0 was set to 1 and a dead-beat control was applied [T=1].

FIG 5.1 POSITIONAL SELF-TUNER

N=2 T=1



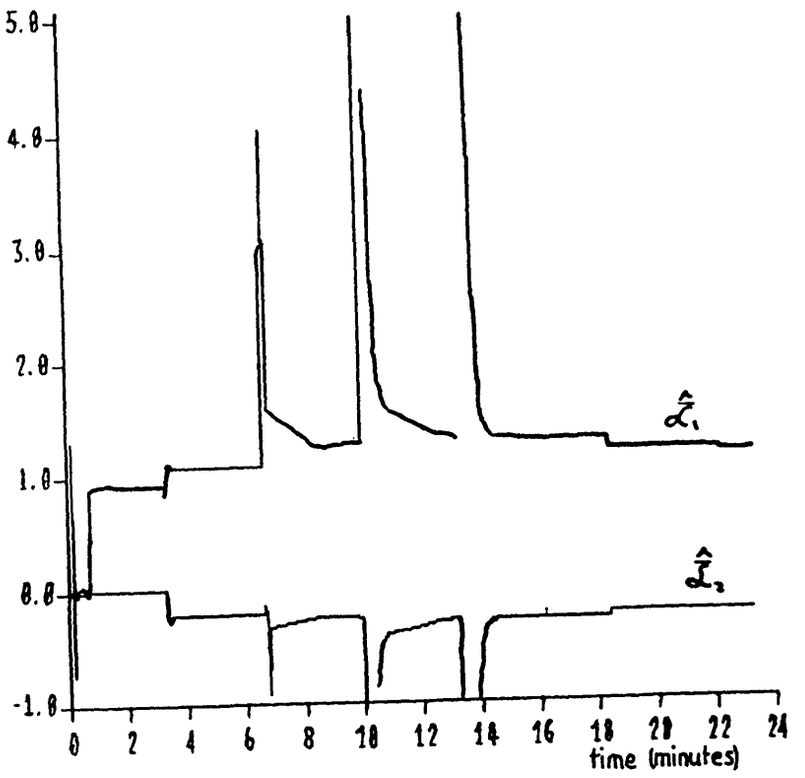
a) Control (U), output (Y) and disturbance (δ)



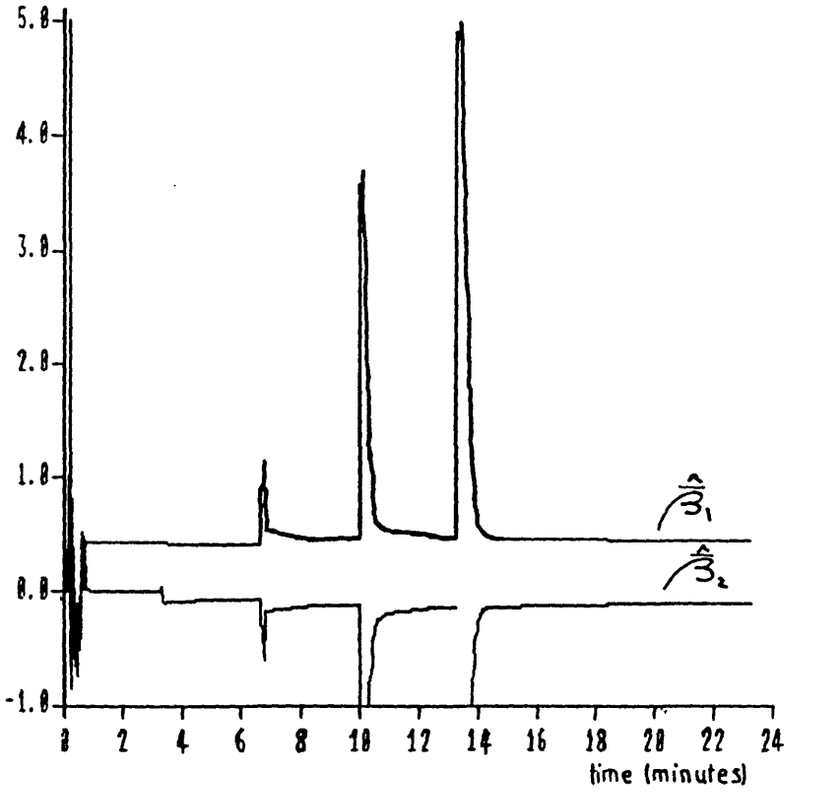
b) Bias parameter estimate

FIG 5.1 POSITIONAL SELF-TUNER

N=2 T=1



c) **Parameter estimates of the output** polynomial



d) Parameter estimates of the input polynomial

It can be seen in Figs. (5.1.b), (5.1.c) and (5.1.d) that all the parameter estimates vary drastically when the load level changes [δ_k varying in steps], even though the rest of the parameters in eq. (5.2) are kept constant. However, as soon as the load stops changing, the parameter estimates stabilize despite the set point changes, denoting that the changes in the parameter estimates are exclusively due to the load disturbances. Simulations performed with other initial parameter settings showed similar results. If a very large value of Σ_0 is used, making the estimator insensitive, the parameter estimates will not change appreciably under load disturbances. However, it will take too long to overcome the offset and to adapt to any changes in model parameters.

At first glance, it seems that the abnormal behaviour shown in Figs. (5.1.c) and (5.1.d) does not affect the performance of the controller [see Fig. (5.1.a)]. However, practical experience has shown that eventually, unnecessary changes in the parameter estimates can induce degradation in the control parameters, affecting the stability and robustness of the whole adaptive control algorithm. This phenomenon has also been observed by Morris et al (1985).

b) It has been argued [Clarke, 1983] that the numerical properties of the RLS estimator can be significantly improved by using a zero-mean data vector; namely the data vector includes only steady-state zero-mean variables. Moreover, it has been shown [Fortescue, 1977] that using difference values [which are steady-state zero-mean variables] in the data vector is equivalent to having an integrator in cascade. These are the reasons why an <u>incremental formulation</u> has been suggested; in addition to increasing the robustness, this formulation overcomes the offset problem in a natural way.

For the case of extended-horizon control, it has been shown in section (3.1) that the data and parameter estimate incremental vectors are:

$$\underline{\phi}_k = [\Delta y_k,\ \Delta y_{k-1},\ \ldots,\ \Delta y_{k-n-1};\ \Delta u_{k+T-1},\ \Delta u_{k+T-2},\ \ldots,\ \Delta u_{k-n+1}]$$

$$\underline{\hat{\theta}}' = [\hat{\alpha}_1,\ \hat{\alpha}_2,\ \ldots\ldots,\ \hat{\alpha}_n;\ \hat{\beta}_1,\ \hat{\beta}_2,\ \ldots\ldots,\ \hat{\beta}_{n+T-1}\ ]$$

The positional and the incremental method can be compared in Figs. (5.1) and (5.2). Figure (5.2) shows the results obtained when the same system used in the simulation shown by Fig. (5.1) is controlled with the incremental algorithm described in section (3.1.1). The initial parameter settings are the same in both simulations.

It can be observed in Fig. (5.2.c) that the parameters converge fairly quickly and remain constant despite the load disturbances. There is no bias estimation, but the output is kept at the desired level [Fig. (5.2.a)]. Many other simulations showed similar results.

Bias Estimation with Incremental Control

In the development of section (3.1.1), it was assumed that the bias parameter δ_k remains constant for long periods. Based on this assumption, the estimation of the bias parameter was eliminated in the incremental algorithm. However, it can be argued that the estimation of a bias parameter would improve the performance of the controller even in the incremental formulation. As is shown below, our own experience contradicts this assertion.

Figure (5.3) compares the incremental algorithm with and without offset estimation in controlling a real-time furnace simulator implemented in the ACS package provided by IBM [see chapter 4].

The simulator is a multivariable interactive system; one of the control objectives is to keep the product at a desired temperature; 600° in this case. The valve which controls the fuel oil feed rate is manipulated to achieve this objective and the unmeasured disturbance is given by step changes in the process stream feed rate. Further details about this simulator are given in IBM (1984).

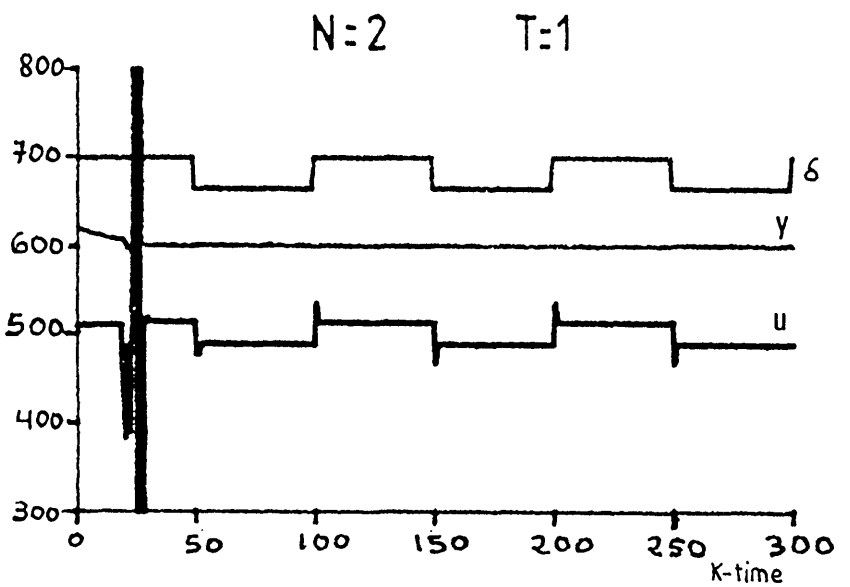
In these simulation experiments, an equal increment control strategy is used [see section (3.1.1)], so that the reciprocal of the gain, with no offset estimation, is given by:

$$d = \sum_{i=1}^{r} \hat{\beta}_i \qquad (5.3)$$

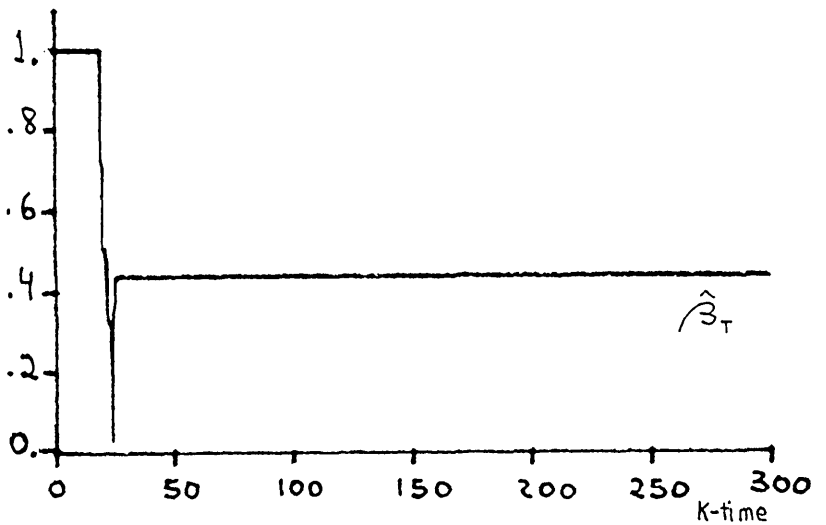
and when the offset is estimated:

$$d' = d + \hat{\delta} \qquad (5.4)$$

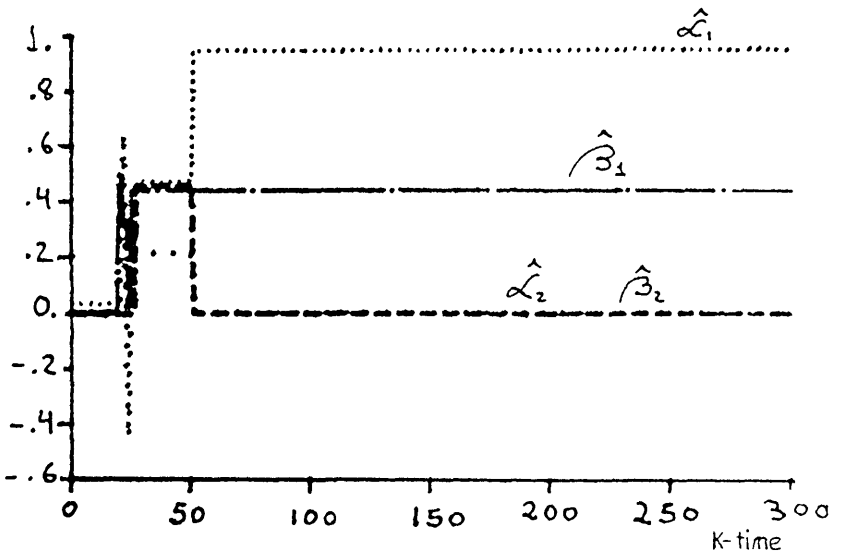
FIG 5.2 INCREMENTAL SELF-TUNER



a) Control (u), output (y) and disturbance (δ)

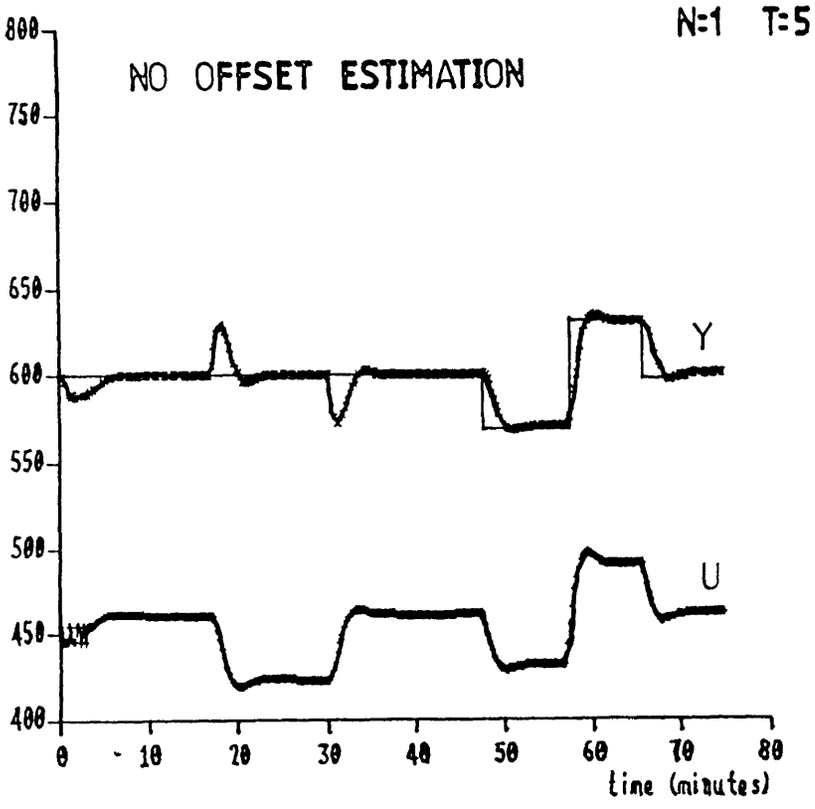


b) Reciprocal of the controller gain

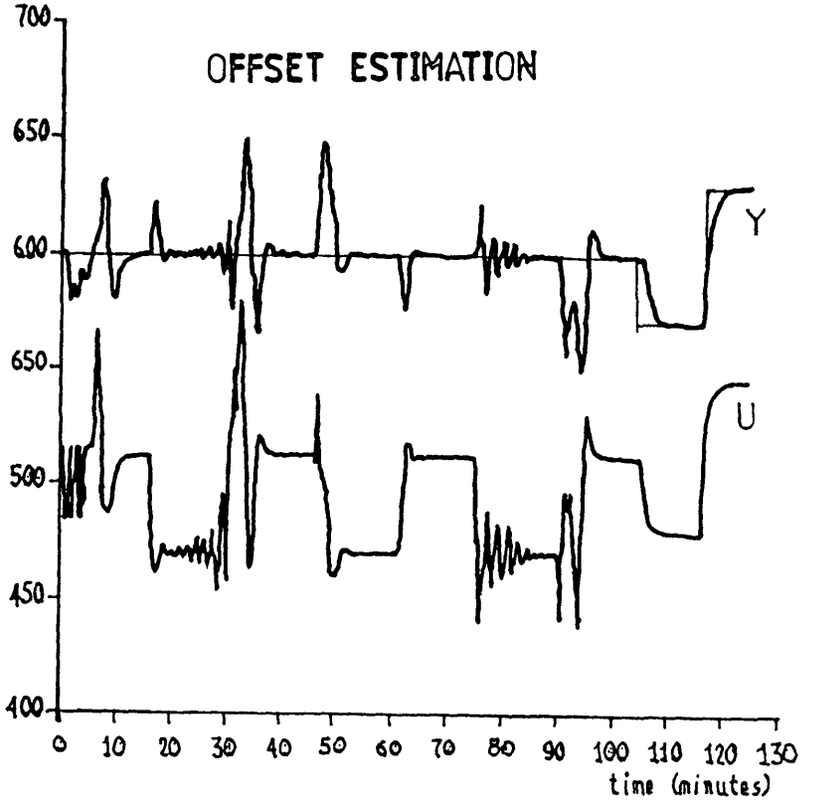


c) Model parameter estimates

FIG 5.3 SELF-TUNING CONTROL WITH DIFFERENT DATA VECTORS



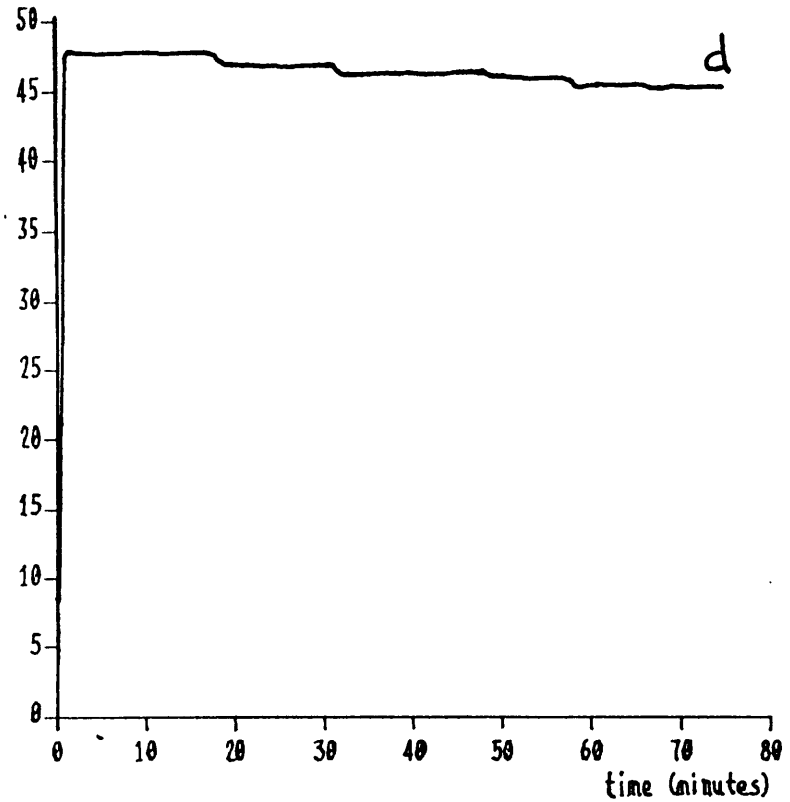
a) Control (U) and output (Y)
of a Furnace Simulator



b) Control (U) and output (Y)
of a Furnace Simulator

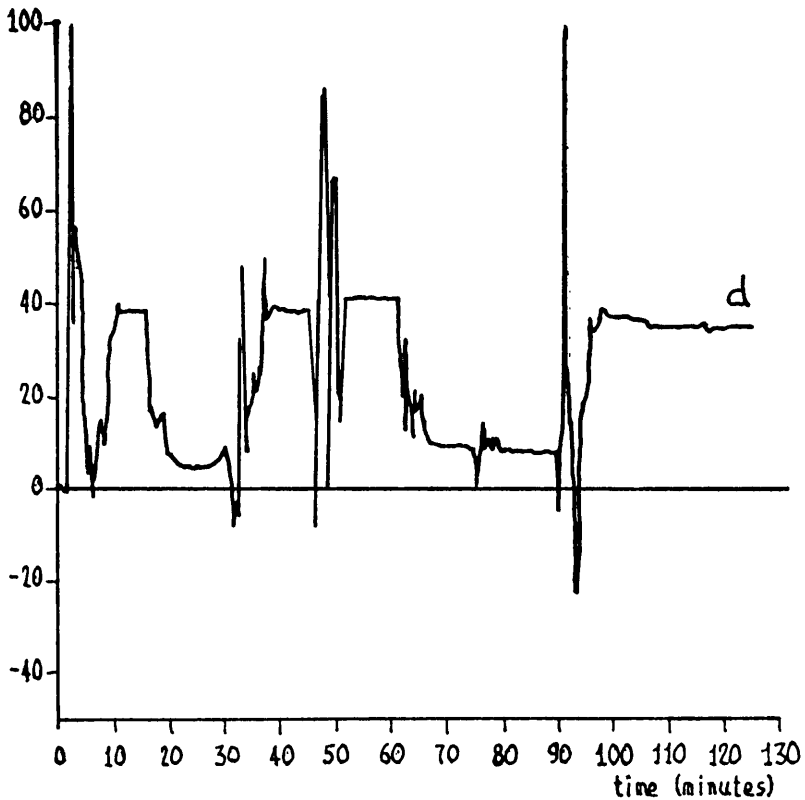
FIG 53 SELF-TUNING CONTROL WITH DIFFERENT DATA VECTORS

NO OFFSET ESTIMATION N:1 T:5



c) Reciprocal of the controller gain

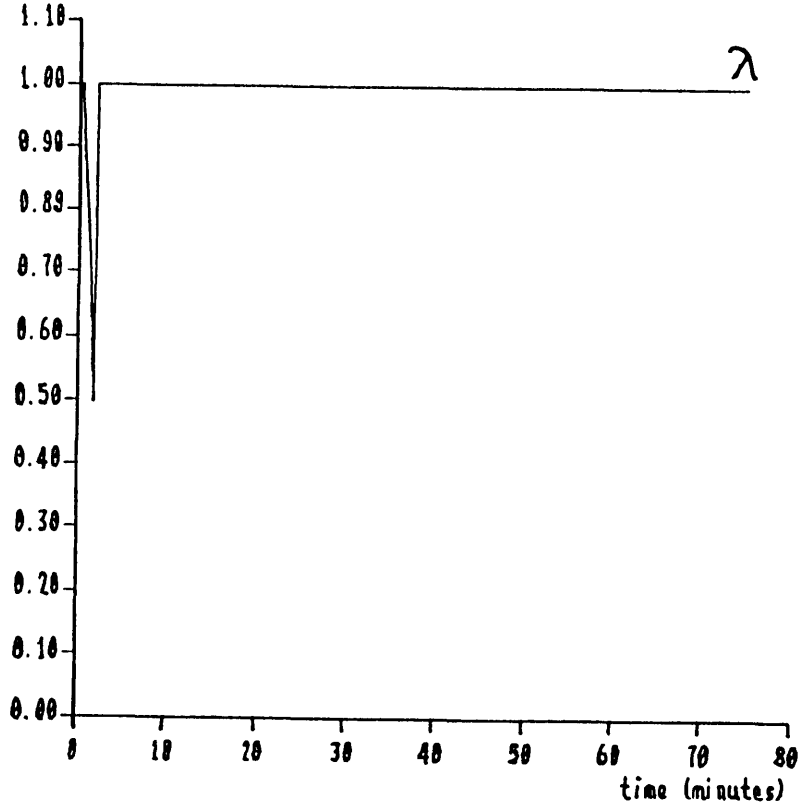
OFFSET ESTIMATION



d) Reciprocal of the controller gain

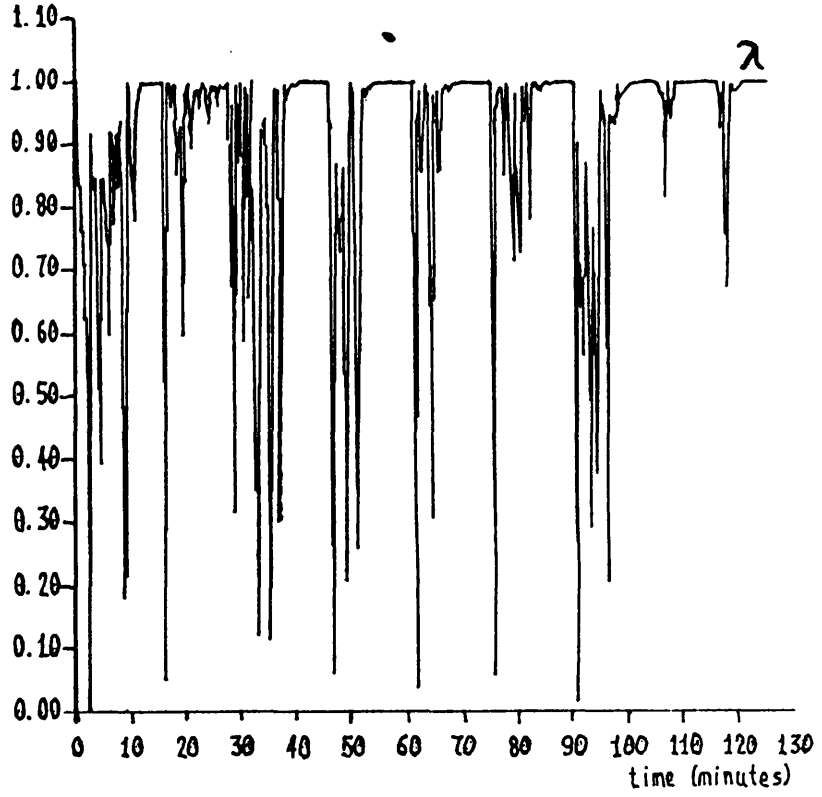
FIG 5.3 SELF-TUNING CONTROL WITH DIFFERENT DATA VECTORS
N:1 T:5

NO OFFSET ESTIMATION



e) Forgetting factor

OFFSET ESTIMATION



f) Forgetting factor

The same values of model order and time-horizon were used in both simulations. The information content Σ_0 was set to one in the simulation without offset. However, it was necessary to use a value of $\Sigma_0 = 100$ in the case with offset estimation, in order to obtain a relatively stable performance. In Figs. (5.3.a) and (5.3.b) the product temperature is measured in °F and represented by y; the control valve position, represented by u, is given in percentagex10.

Figures (5.3.a) and (5.3.b) show that the algorithm without offset estimation performs much better than the one with offset estimation. It can also be seen in Fig. (5.3.c) that the reciprocal of the controller gain (d) is fairly stable in the first case.

On the other hand, when the offset is estimated, the gain 1/d' varies quite drastically [Fig. (5.3.d)] and even changes sign. This is caused by similar changes in the estimate of the offset parameter, δ. It can be observed from the behaviour of λ shown in Fig. (5.3.f) that the RLS estimator permanently forgets information. The effect of this is to decrease the confidence of the estimator in its parameter estimates, so that they jump from one value to another.

Many other simulations with the offset estimation algorithm showed the same poor behaviour which is caused by the abrupt changes in the bias estimate when load changes occur.

Control Strategy

The performance of the extended-horizon self-tuning regulator also depends on the control strategy selected. In section (3.1) several control strategies were mentioned. Two of these are compared below:

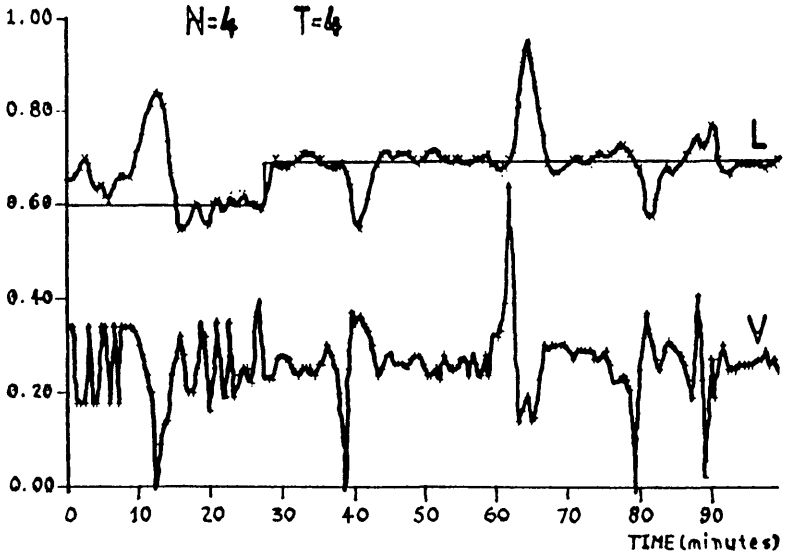
a) equal-increments

b) constant control

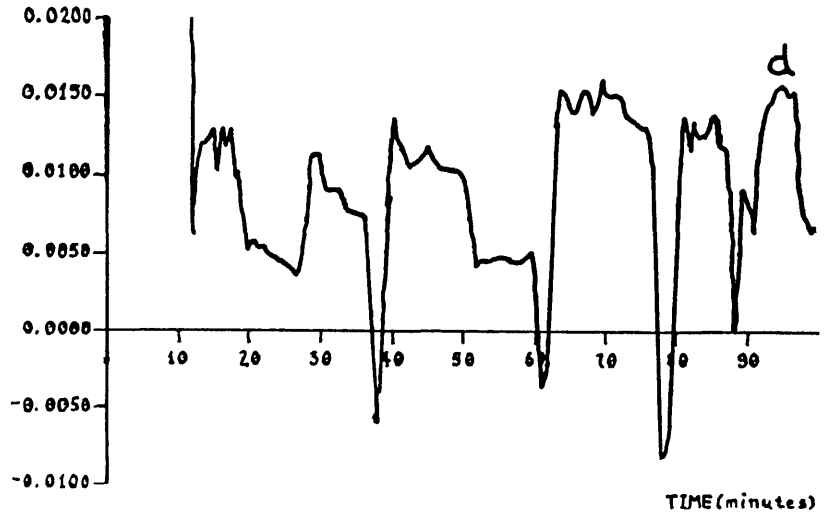
The justification for using these strategies is given in the next section.

a) In Fig. (5.4) two experiments are shown, in which the equal-increments approach is used to control the level of the pilot-plant absorption column. The inlet valve position is the manipulated variable [configuration 2 in Fig. (4.2)] and the disturbance is provided by step changes in the outlet flow. In both experiments the sample time was 35 sec and the information content $\Sigma_0 = .001$.

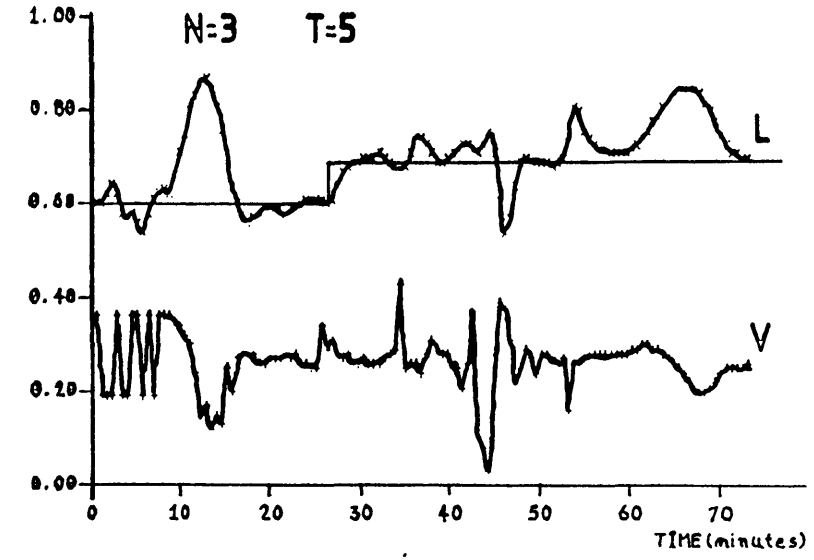
FIG 5.4 EQUAL INCREMENTS CONTROL STRATEGY



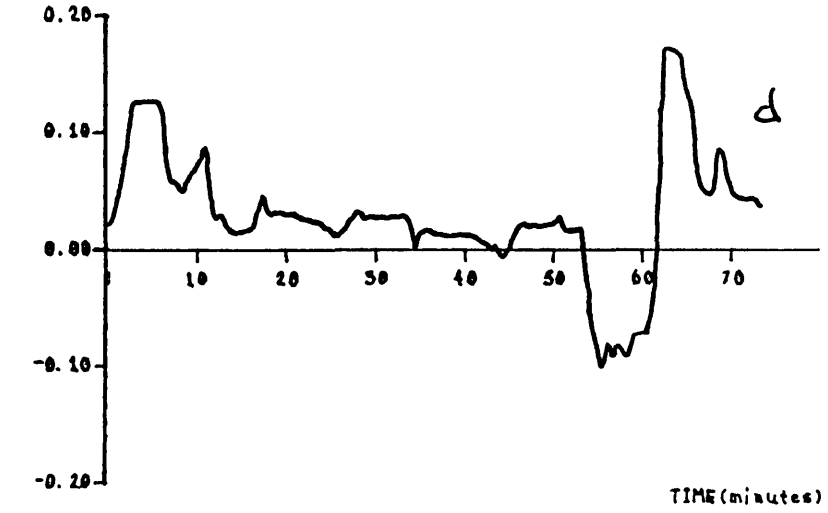
a) Inlet flow valve position (V)
and liquid level (L)



b) Reciprocal of the controller
gain



c) Inlet flow valve position (V)
and liquid level (L)



d) Reciprocal of the controller gain

In one of the experiments [Figs. (5.4.a) and (5.4.b)], the model order n and the time-horizon T were both set to 4. It can be seen in Fig. (5.4.b) that the d parameter changes by jumps each time a disturbance enters the system [t=40, t=60, t=80 and t=90]; sometimes it changes sign causing the valve [Fig. (5.4.a)] to move in the wrong direction.

In the other experiment [Figs. (5.4.c) and (5.4.d)] the initial parameters were set to n=3 and T=5. Figure (5.4.d) shows that the d parameter drifts to zero, instead of changing in jumps; eventually it changes sign. In this experiment the controller gain [1/d] remains for a longer period with an incorrect sign and consequently, the level [Fig. (5.4.c)] stays longer away from the set point.

Performance can apparently be improved if the RLS estimator is made less sensitive by selecting a larger Σ_0, as in the simulation shown in Fig. (5.3). However, if the system is continuously affected by load disturbances, the d parameter will still drift to zero and eventually will change sign. The controller gain will remain for an even longer period with the opposite sign since the RLS estimator has been made relatively insensitive. Clearly, in this condition the closed-loop becomes unstable.

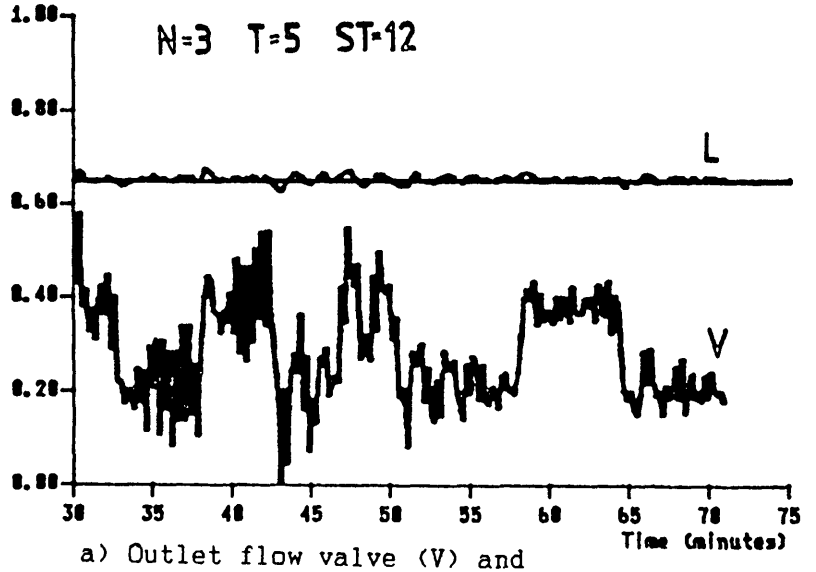
One would have expected better behaviour for equal increments because the called for control action is gradual. However, the reciprocal of the gain d given in eq. (5.3) depends on the first $\hat{\beta}$[from $\hat{\beta}_1$ to $\hat{\beta}_T$] parameter estimates. Practical experience has shown that these estimates change considerably when the system is subjected to load disturbances. These $\hat{\beta}$ estimates drift to zero or change in jumps, causing similar changes in the reciprocal of the controller gain [see next section].

b) A better approach is to select the constant control strategy. In this case, the reciprocal of the controller gain is given by the $\hat{\beta}_T$ estimate. As practical experience has shown, this estimate is not affected by load changes [see next section] so that a more robust controller is obtained.

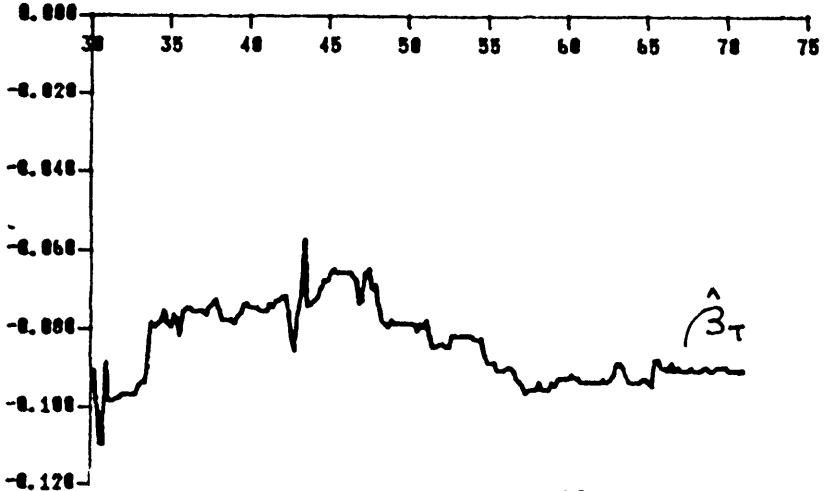
Figure (5.5) compares the constant control and the equal-increments control strategies. The level of the absorption column is controlled by the outlet valve, so that no control delay is present; the disturbance is produced by changing the inlet flow. The initial parameters are set at the same value in both experiments. The only difference is that a sample interval (ST) of 10 sec is used in the

FIG 5.5 SELF-TUNING WITH DIFFERENT CONTROL STRATEGIES

CONSTANT CONTROL

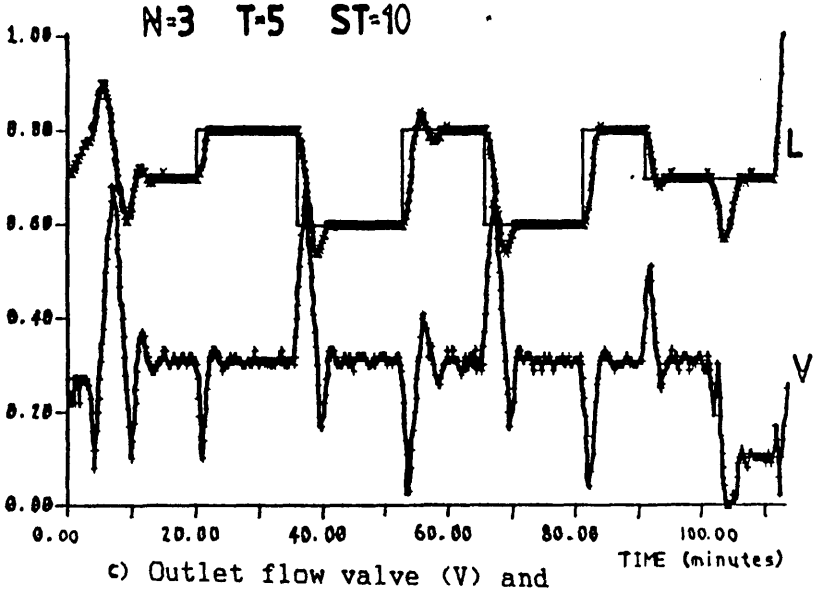


a) Outlet flow valve (V) and liquid level (L)

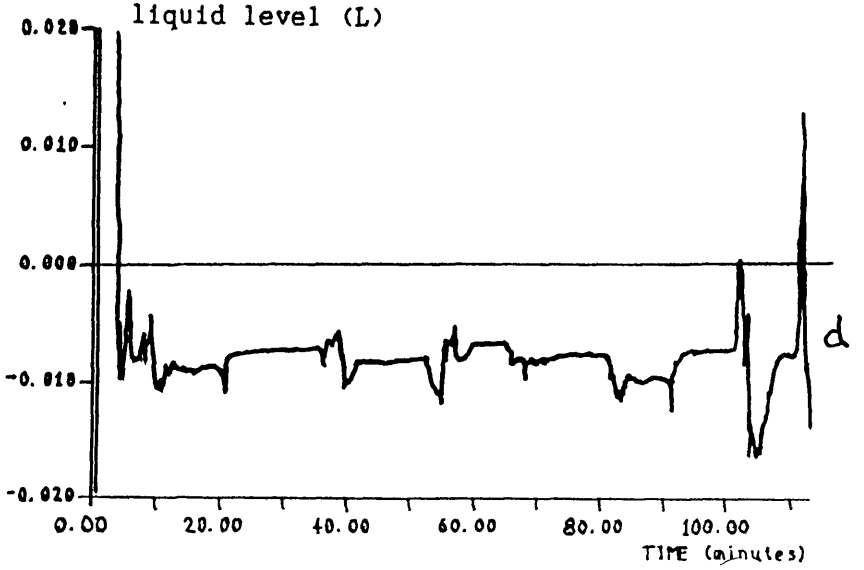


b) Reciprocal of the controller gain

EQUAL INCREMENTS



c) Outlet flow valve (V) and liquid level (L)



d) Reciprocal of the controller gain

first case and one of 12 sec in the second case. However, the difference is irrelevant.

In the experiment shown in Fig. (5.5.c), the performance of the controller with equal increments strategy appears reasonably good when the plant is subjected to set point changes; namely from time t=10 to t=90. However, when load changes occur (t=100 and t=105), the controller gain [Fig. (5.5.d)] changes sign and produces a control action in the wrong direction. In this situation, the system becomes unstable and the column overflows.

On the other hand, the results obtained with the constant control strategy [Figs. (5.5.a) and (5.5.b)] are much better. The reciprocal of the controller gain, after convergence, remains constant despite the load disturbances. In this case, an almost perfect regulation is achieved.

Direct vs Indirect:

a) The control algorithms discussed above were all of the direct type. These algorithms employ a predictor model to estimate the plant parameters. The adaptive controller make use of these estimates without modifying them. It is generally argued that by this method a considerable amount of computing time can be saved since it is not necessary to calculate the control parameters from the model. However, it was found by experience that this approach can lead to problems in cases where the system is affected by a series of deterministic load changes. This is due to the fact that the estimator does not use the most recent values of the output to update the parameter estimates. The convergence of the algorithm is relatively slow, especially when the extended-horizon strategy is used.

Consider for example the measurement equation (3.7) of section (3.1.1):

$$y_k = y_{k-T} + \underline{\phi}_{k-T}'\hat{\underline{\theta}} \qquad (5.5)$$

with:

$$\underline{\phi}_{k-T} = [\Delta y_{k-T}, \Delta y_{k-T-1}, \ldots, \Delta y_{k-T-n+1}; \Delta u_{k-1}, \Delta u_{k-2}, \ldots, \Delta u_{k-n-T+1}]$$

$$\hat{\underline{\theta}}' = [\hat{\alpha}_1, \hat{\alpha}_2, \ldots\ldots, \hat{\alpha}_n; \hat{\beta}_1, \hat{\beta}_2, \ldots\ldots, \hat{\beta}_{n+T-1}]$$

It can be seen that the data vector $\underline{\phi}_{k-T}$ does not contain the values of the known outputs from time t=k-1 to t=k-T+1. This is irrelevant for linear systems with constant parameters since, in principle, once the parameter estimates have converged, no more data is needed. However, if any plant parameter suffers a drastic change at time t=k, the parameter estimator will not realize the change has occurred until T-1 sample times later. The output y_k will reflect the change in the plant earlier than any other item in the data vector. Increasing the horizon, amplifies the problem.

If the prediction-horizon is set to one in a system that does not have delay, the problem described above can be avoided. In this case, the most recent information is used in the data vector $\underline{\phi}$. However, the controlled system becomes minimum-variance, which leads to difficulties of robustness.

The problem described can be appreciated by comparing Figs. (5.2) and (5.6). In both simulations an incremental direct self-tuner was used. The linear system was subjected to drastic changes in its load level δ_k. In Fig. (5.2) T was set to 1 and in Fig. (5.6) T was set to 3. It can be seen that in the system controlled with extended-horizon T=3 the reciprocal of the controller gain $\hat{\beta}_T$ [Fig. (5.6.c)] does not converge as fast as in the system controlled with T=1 [Fig. (5.2.b)].

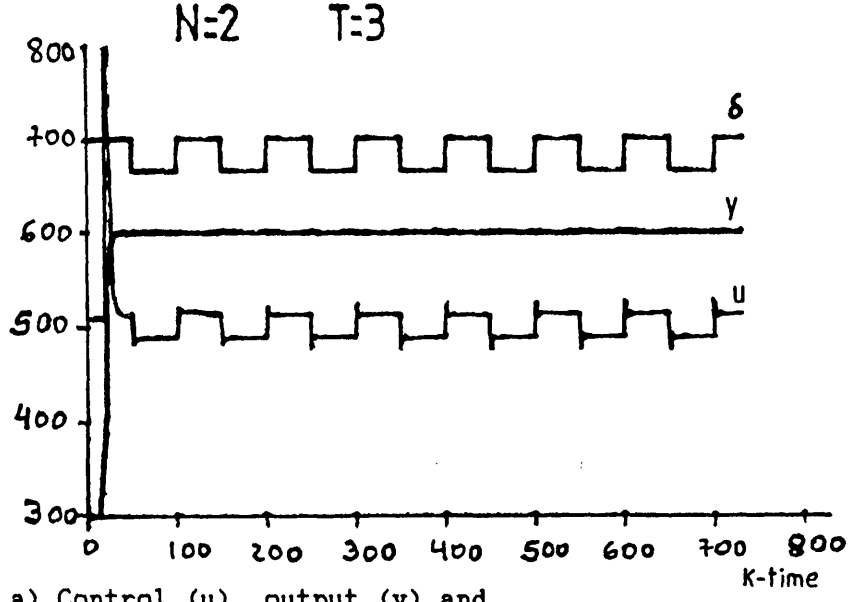
If the control input is affected by any delay [du] and a minimum-variance strategy is employed, problems will also occur with the direct control method. In this case, the first value in the data vector will be y_{k-du} so that the estimation algorithm will only perceive a change in the system du time intervals later.

b) One way to solve the above problem is by using a non-predictive model [and, hence, the most up-to-date data] to estimate the plant parameters. The predicted output at time k+T can be computed from these parameter estimates, so that an extended-horizon control law can be applied.

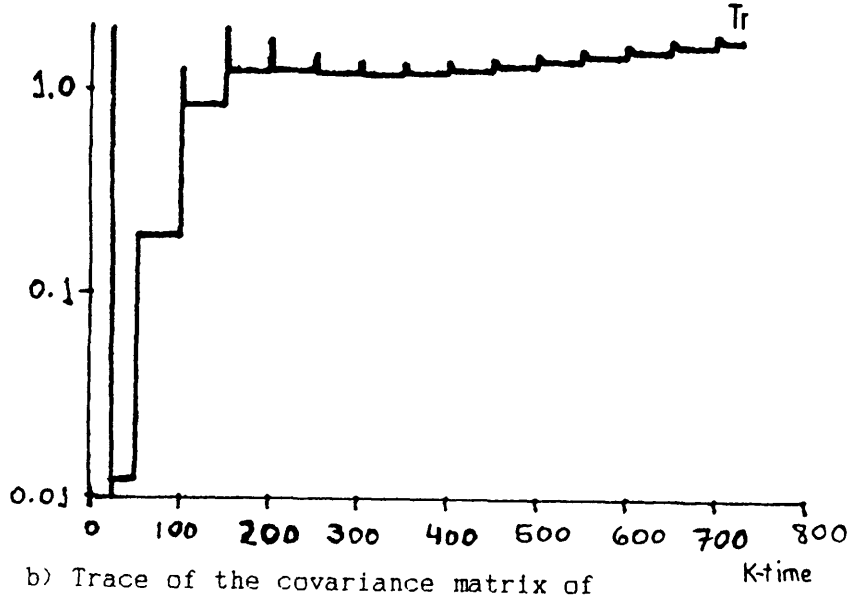
This method is called <u>indirect</u> because the control parameters are obtained <u>indirectly</u> from the model parameters through further computation.

In the indirect approach the most recent values are used in the data vector. Any change in the controlled system is immediately sensed by the estimation algorithm. An extended-horizon control strategy can still be applied as with the direct approach, but at each interval the model parameter estimates are used to compute the control parameters.

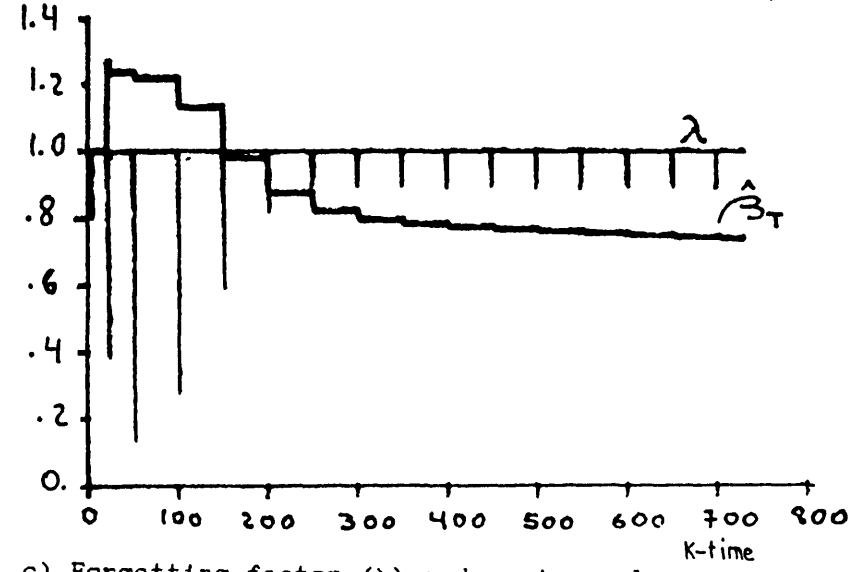
FIG 5.6 DIRECT SELF-TUNER



a) Control (u), output (y) and disturbance (δ)



b) Trace of the covariance matrix of the RLS estimator



c) Forgetting factor (λ) and reciprocal of the controller gain ($\hat{\beta}_T$)

In this way, a more robust self-tuner can be obtained. Of course, the price paid is that at each interval more computing time is needed to obtain the control parameters, via the appropriate transformation.

In Appendix I an indirect extended-horizon controller is derived and the parameter transformation is given in a recursive form.

By way of comparison, the simulation shown in Fig. (5.6) was repeated in Fig. (5.7) with an indirect extended-horizon self-tuner. It can be seen in Figs. (5.7.c) and (5.6.c) that the $\hat{\beta}_T$ parameter converges much faster in the indirect controller than in the direct controller. It can also be observed in the simulation with the indirect approach, that the trace Tr of the covariance matrix [Fig. (5.7.b)] reached lower values [compared with the trace shown in Fig. (5.6.b)], indicating more confidence in the parameter estimates.

The forgetting factor λ [Figs. (5.6.c) and (5.7.c)] also reflects the better performance of the estimator in the indirect case. Figure (5.6.b) shows that, initially, the $\hat{\beta}_T$ estimate had converged to its true value, but then it drifted away because valuable information was forgotten.

Furthermore, it can be appreciated that the control action is smoother with an indirect self-tuner, since the controller gain converges to a lower value [higher $\hat{\beta}_T$]. This is another characteristic of this approach, observed in simulations and experiments, that enhance robustness.

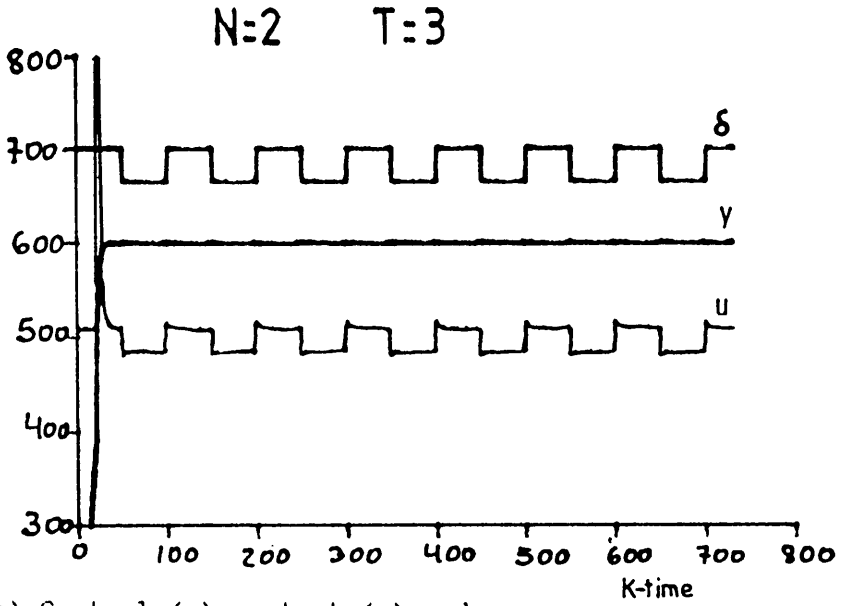
Another advantage of the indirect approach is the ability to change the prediction-horizon T on-line, without the need of having various estimators operating simultaneously.

The main drawback of this method is the need to compute the control parameters through further computations, thus increasing the computing time required.

The indirect [explicit] approach was implemented only towards the end of the project and very few experimental results have been obtained.

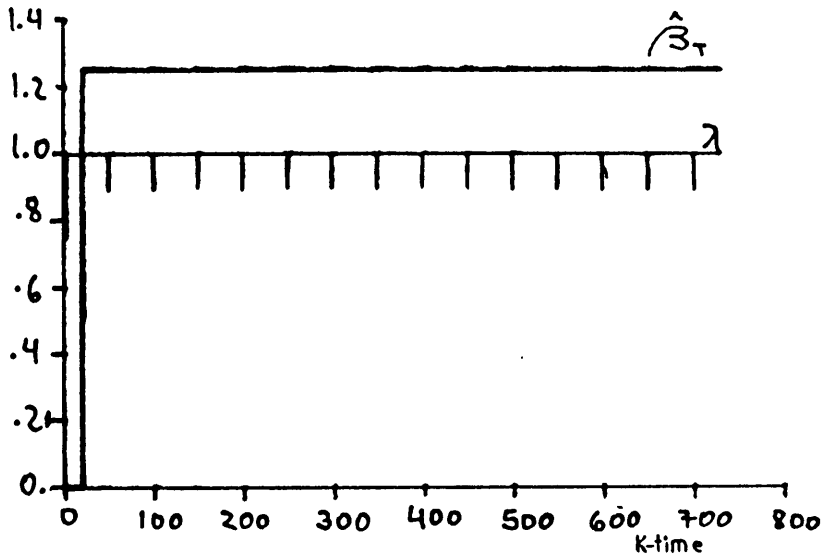
As an example, Fig. (5.8) gives the performance of an explicit algorithm with a constant control strategy controlling the liquid level of the absorption column . The manipulated variable is the outlet valve [configuration 1] and the disturbance is produced by step changes in the inlet flow; these changes are applied approximately every 5 minutes from time t=50 to t=110.

FIG 5.7 INDIRECT SELF-TUNER



a) Control (u), output (y) and disturbance (δ)

b) Trace of the covariance matrix of the RLS estimator



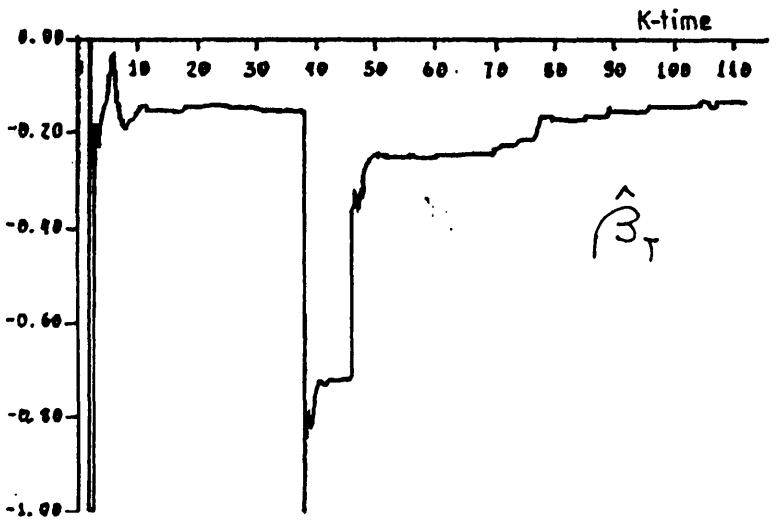
c) Forgetting factor (λ) and reciprocal of the controller gain ($\hat{\beta}_T$)

If Fig. (5.8) is compared with Fig. (5.5), it can be seen that the indirect [explicit] controller tolerates larger and more frequent load changes than the direct [implicit] controller, without much perturbation on the controller gain [Fig. (5.8.a)]. Moreover, the controller gain reached lower values [higher $\hat{\beta}_T$] in the explicit algorithm [compared with $\hat{\beta}_T$ shown in Fig. (5.5.b)], producing a more robust control; even though a smaller order and a smaller control-horizon were used.

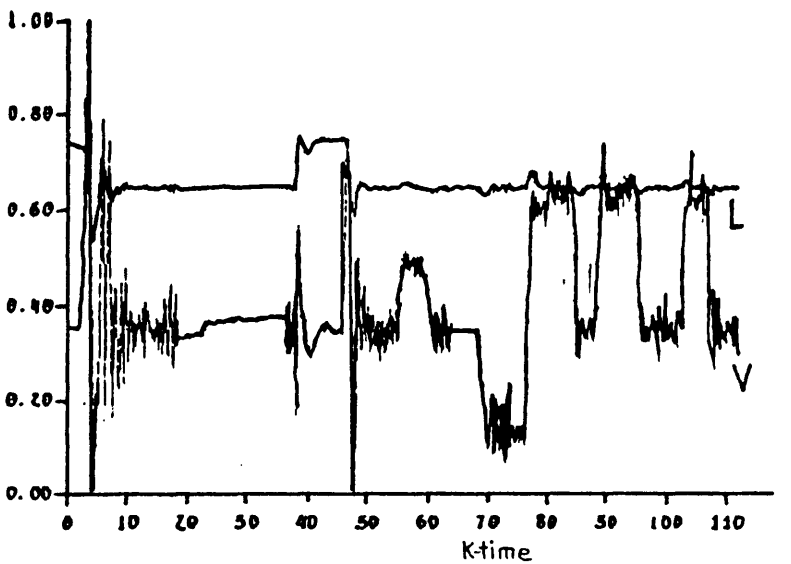
In spite of the limited experimental work, it is believed that the indirect approach is more robust than the direct approach, especially for load changes. However, the evidence is not conclusive and contradicts the experience of Ydstie (1986), so that much more work must be done to elucidate this issue.

FIG 5.8 INDIRECT (EXPLICIT) CONTROLLER

N=2 T=3 Z=.005 DELT=.2



a) Reciprocal of the controller gain



b) Outlet flow valve position (V) and liquid level (L)

5.1.2.- Feedback-Feedforward Self-tuning Control

An incremental FB/FF adaptive controller has been studied with experiments on the CO_2 pilot-plant as well as via simulations. The study has been focused on the following issues:

i) comparison between the FB/FF controller with a purely FB one
ii) sensitivity of the FB/FF controller to initial parameter selection
iii) robustness against time-delays on both the control and the measured disturbance
iv) initialization
v) direct disturbance transmission
vi) auto-tuning and detuning properties of the FB/FF adaptive controller

In order to simulate normal industrial practice, the rate of change of the control in all experiments shown in this sub-section has been limited to 20% of the input range:

$$|u_k - u_{k-1}| \leq .2*(Umax - Umin) \qquad (5.6)$$

where Umax and Umin are the maximum and minimum value of the control variable. This is typical of constraints placed on controllers in chemical industry.

A measure of the feedforward compensation is given in the figures of this sub-section by the term γ:

$$\gamma = \hat{\delta}_T + \hat{\delta}_{T+1} + \ldots + \hat{\delta}_{T+q-1} \qquad (5.7)$$

This term is the summation of all the coefficient estimates of the disturbance polynomial that are included in the FB/FF control law defined in section (3.1.2).

In the following, a series of experimental results obtained with the level control-loop [see chapter 4] are presented, in which both configurations are used:

i) configuration 1 in which the outlet valve is the manipulated variable and the disturbance is produced by changing the inlet flow.
ii) configuration 2 in which the inlet valve is the manipulated variable and the disturbance is produced by changing the outlet flow

The manipulated variables, denoted by u, operate between 0 and 100%, but they appear in Figs. (5.9) to (5.16) scaled between 0 and 1. In these figures, the level is measured in metres and denoted by y.

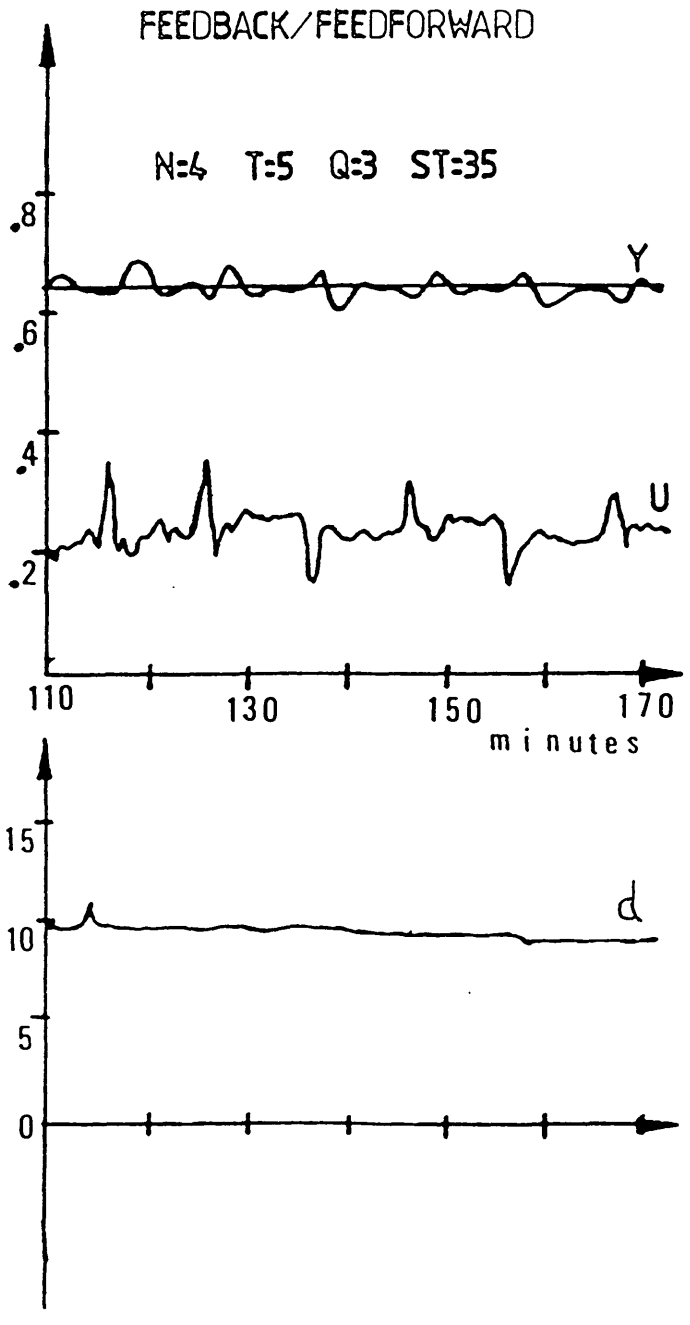
Feedback/Feedforward vs Purely Feedback Control

Figure (5.9) shows two experiments that compare a FB/FF self-tuner with a purely FB one; an equal-increments control strategy is applied in both cases. In these experiments, the level is controlled with the inlet valve [configuration 2 in Fig. (4.2)] and the measured disturbance is the outlet flow. This control configuration, as mentioned in Chapter 4, is characterized by a large an variable control-delay.

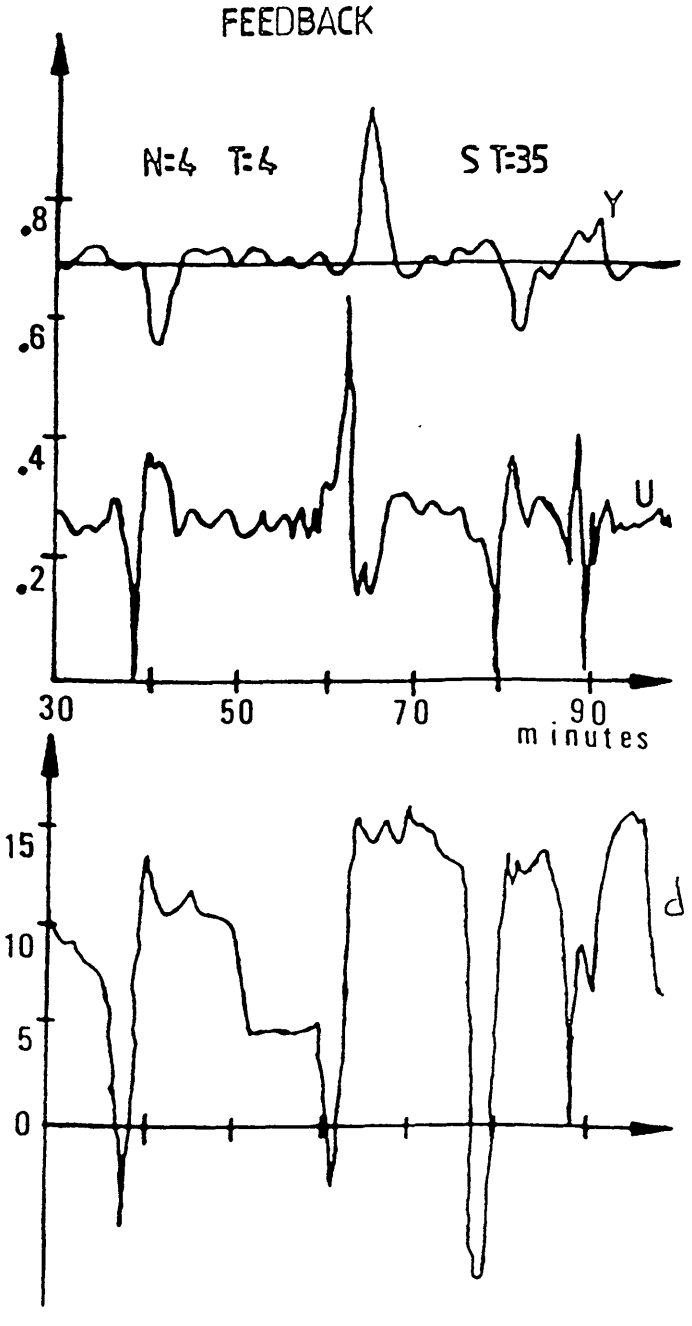
The information length Σ_o was set to .1 and the time-horizon T to 5 in the FB/FF experiment. While in the FB only case, Σ_o was set to .01 and T was set to 4. The model order n and the sample time ST were the same in both experiments.

Figure (5.9.b) shows that the controller gain [d^{-1}] of the FB algorithm is very sensitive to the load changes applied at times t=40, t=60, t=80 and t=90. This gain takes on very large values [near zero values of d] and even changes sign, causing large deviations in the level of the column.

FIG 5.9:
FB/FF AND FB CONTROL UNDER LOAD CHANGES



a) Inlet flow valve position (U), liquid level (Y) and reciprocal of the controller gain (d)



b) Inlet flow valve position (U), liquid level (Y) and reciprocal of the controller gain (d)

On the other hand, the controller gain of the FB/FF algorithm [Fig. (5.9.a)] remains almost constant when the load disturbances enter the system [every five minutes from time t=115 to t=165]. The deviations from the set point shown by the liquid level are mainly due to the delay in the control action. Note that no model had to be provided for the relationship between the measured disturbance and the output; only the model order was required. In this way, the main drawback of feedforward [non-adaptive] control, namely the requirement of the availability of a good process model, is avoided; the necessary I/O relationship has been estimated on-line.

Initial Settings, Time-delays and Control Strategies

It was later detected that, under certain conditions, the performance of the FB/FF self-tuning algorithm becomes deteriorated. Indeed, it has been observed that the reciprocal of the controller gain sometimes drifts to zero and even changes sign. This phenomenon, which affects the performance and stability of the closed-loop system, is due to a combination of the following factors:

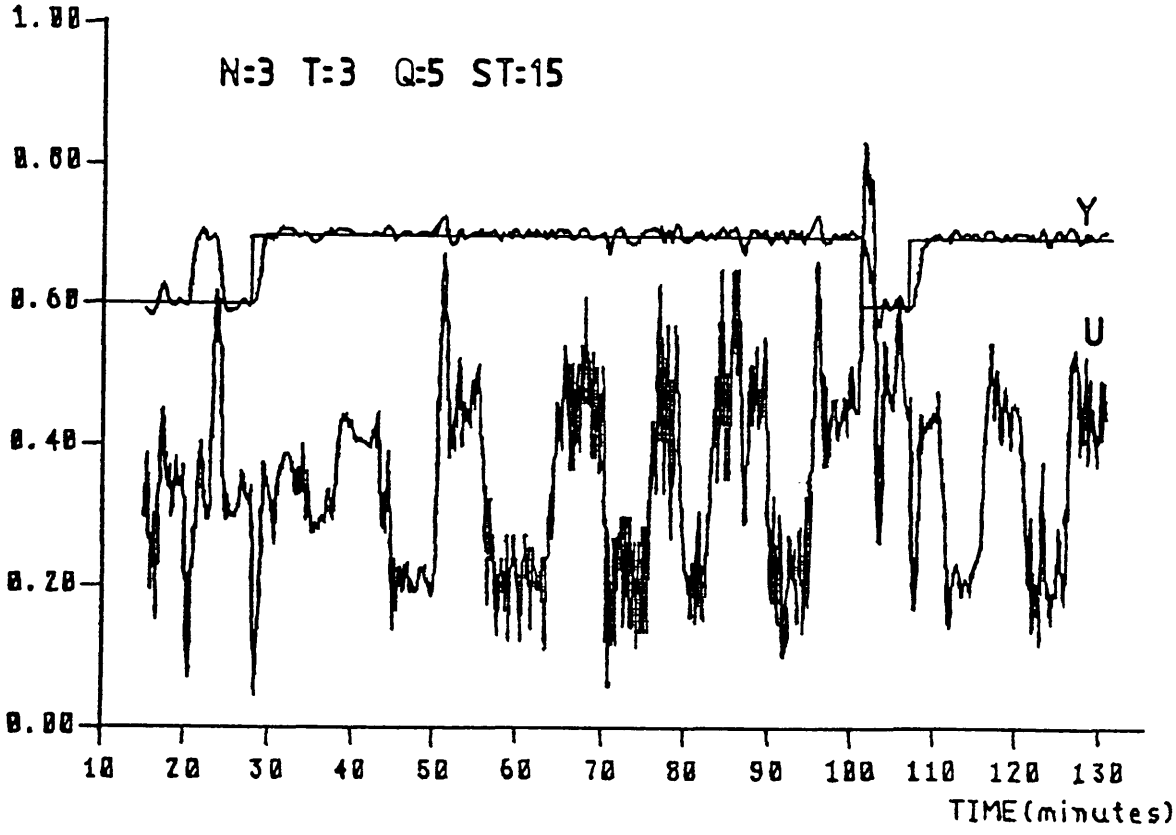
i) setting of initial parameters which are not large enough
ii) presence of time-delays affecting the measured disturbances
iii) use of non-robust extended-horizon control strategies

The experiment of Fig (5.9.b) had not been affected by this phenomenon since there were no significant time delays in the measurement of disturbances.

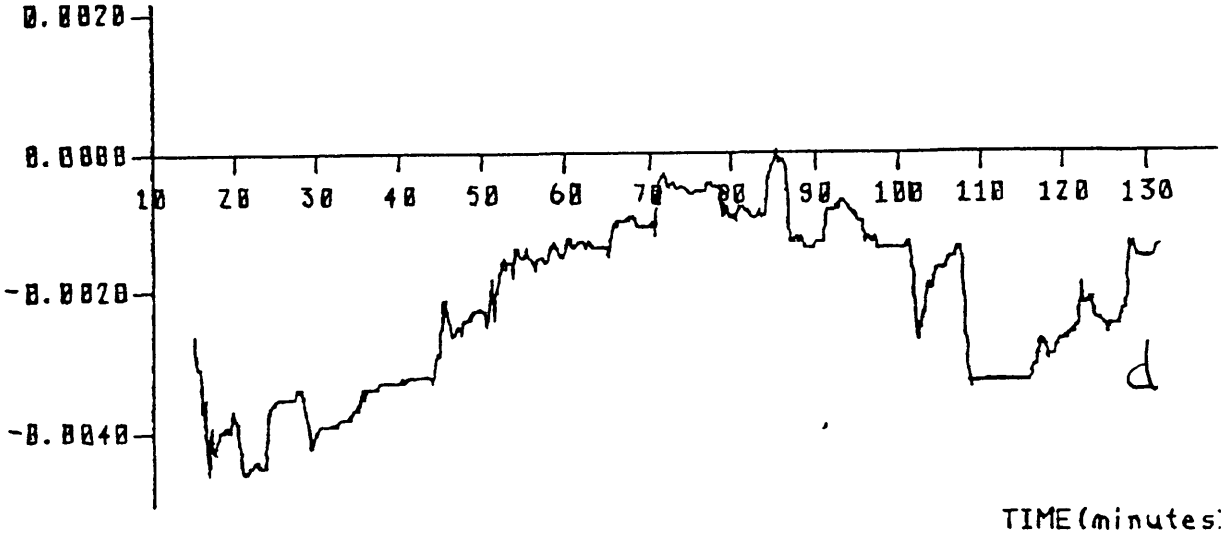
i) Figures (5.10), (5.11) and (5.12) show the experimental results obtained when all the aforementioned conditions are satisfied:

- small values of initial parameters are used
- the liquid level of the absorption column is controlled by the outlet valve position [configuration 1]. The measured disturbance is the inlet flow, which is affected by a large and variable time-delay.
- the system is controlled with an equal-increments control strategy.

FIG 5.10: FB/FF CONTROL WITH DRIFT IN THE CONTROLLER GAIN

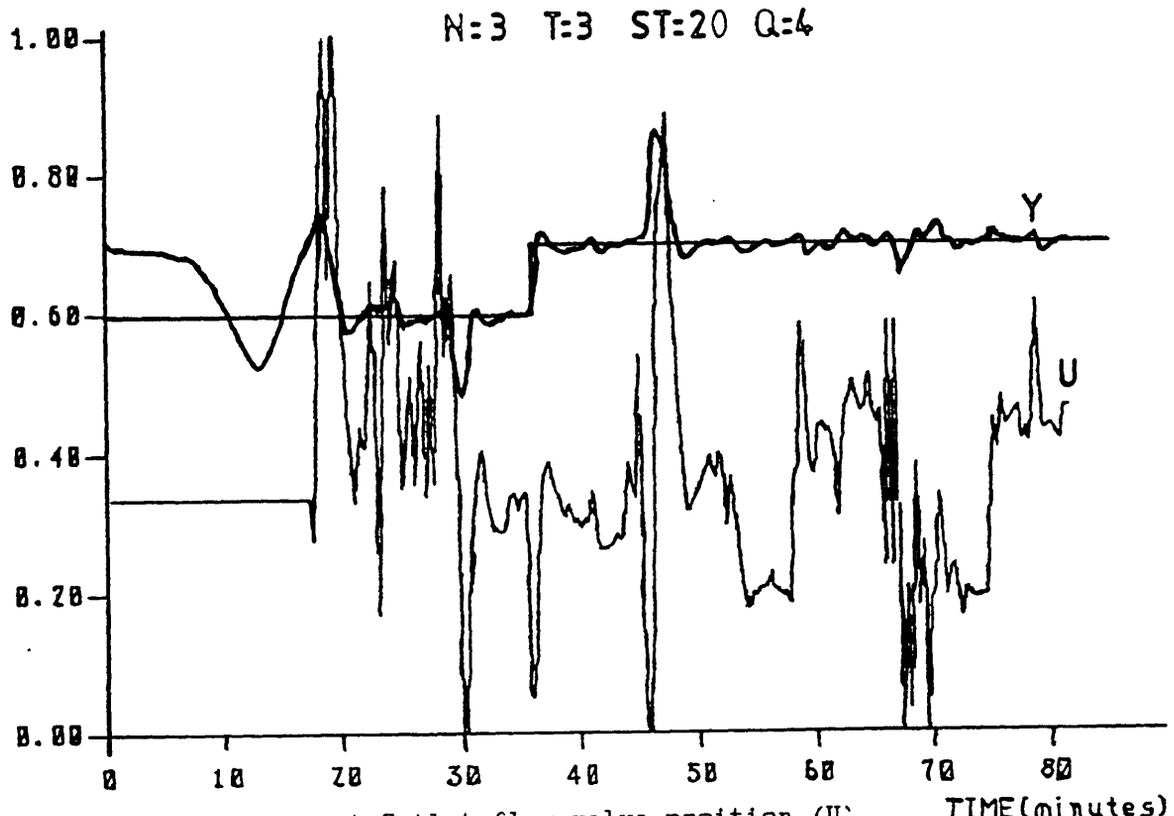


a) Outlet flow valve position (U)
and liquid level (Y)

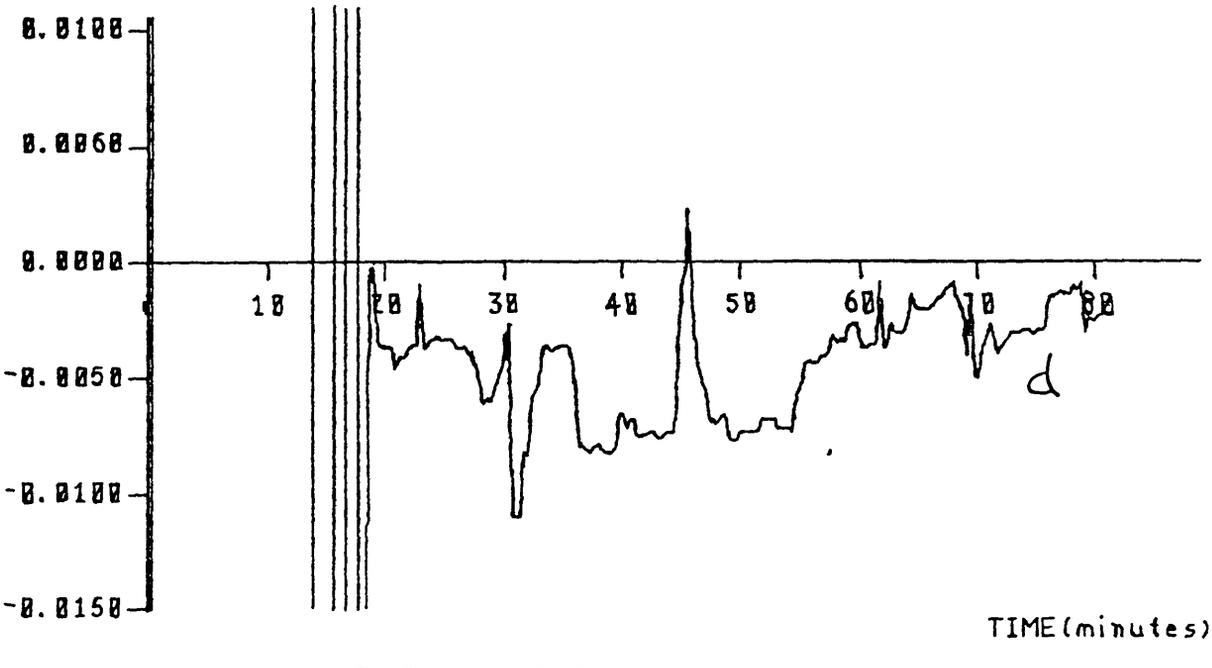


b) Evolution of the reciprocal
of the controller gain (d)

FIG 5.11: FB/FF CONTROL WITH DRIFT IN THE CONTROLLER GAIN

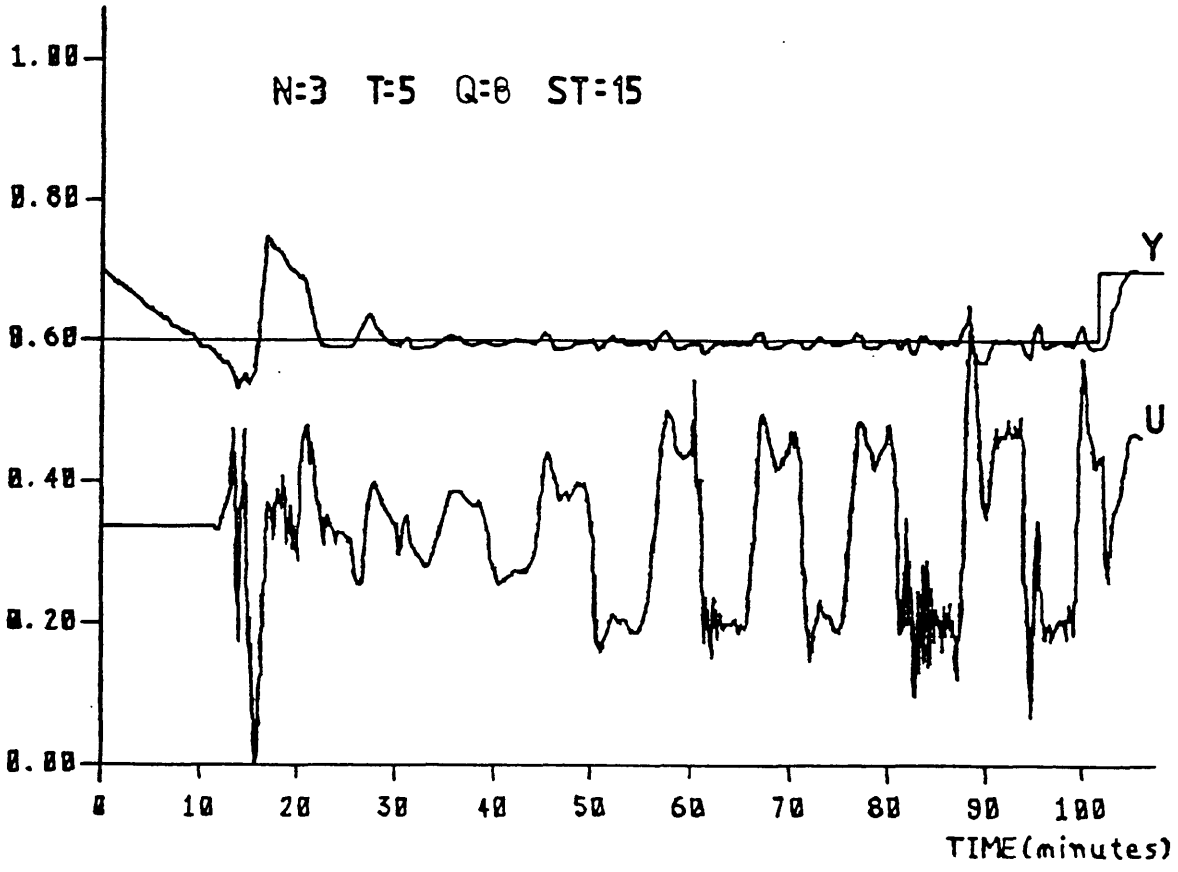


a) Outlet flow valve position (U) and liquid level (Y)

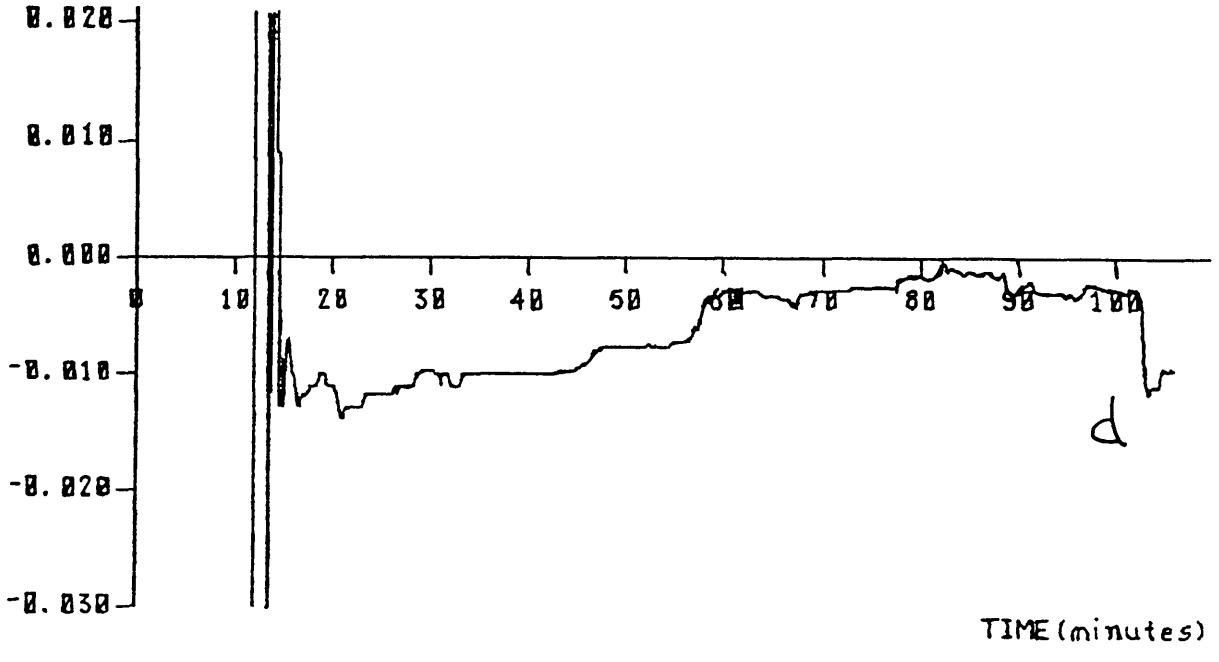


b) Evolution of the reciprocal of the controller gain (d)

FIG 5.12: FB/FF CONTROL WITH DRIFT IN THE CONTROLLER GAIN



a) Outlet flow valve position (U)
and liquid level (Y)



b) Evolution of the reciprocal
of the controller gain (d)

These experiments illustrate the "drift to zero" phenomenon suffered by the reciprocal of the controller gain.

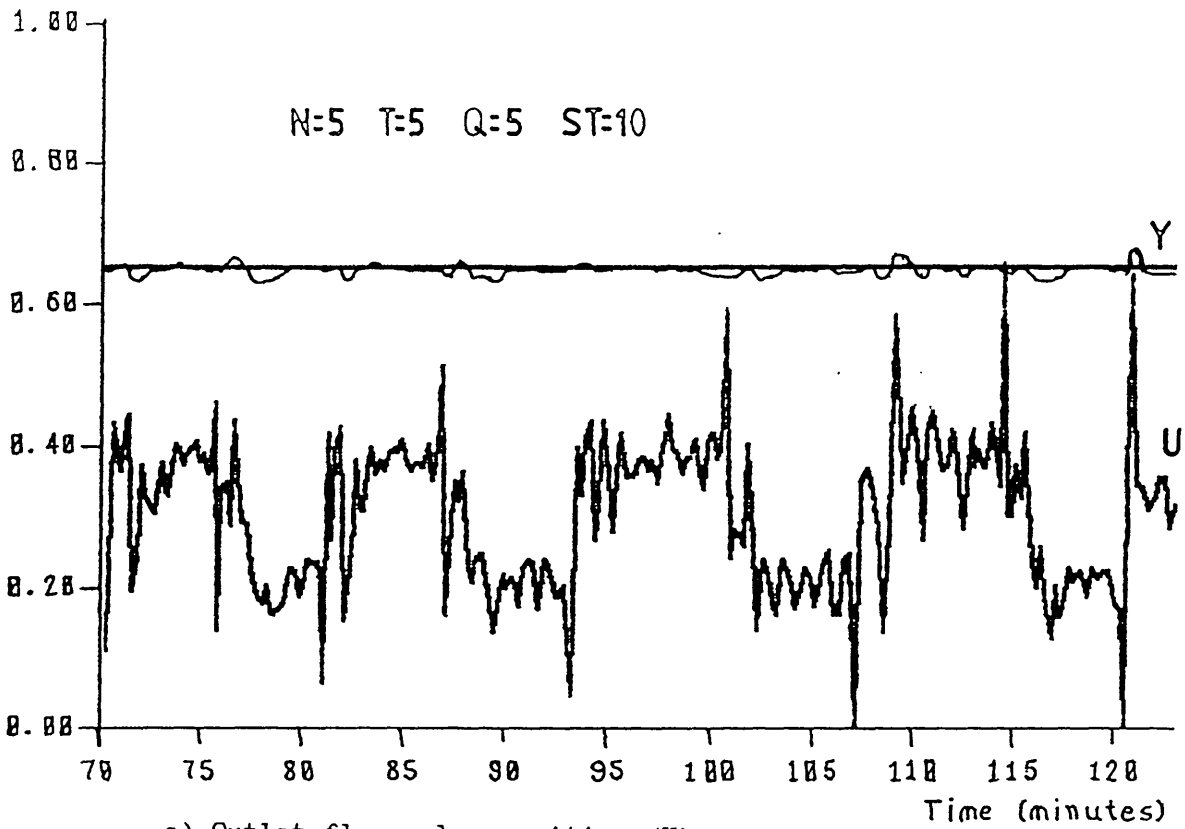
For example, Fig. (5.10) shows the performance of the FB/FF closed-loop system when the inlet flow changes by steps every 5 minutes, from time t=35 to time t=125. In this experiment, the information content Σ_0 was set to .001, the order q was 5 and the time horizon T and the order n were set to 3. It is clearly seen in Fig. (5.10.b) how the the reciprocal of the controller gain d drifts. Large values on the controller gain [small values on d] produce an oscillatory and sensitive control action. Introducing some excitation into the system with set point changes, makes the gain converge to more suitable values. However, if the disturbance continues affecting the system, the controller gain will detune again.

In the experiment represented by Fig. (5.11), a solution to the problem of the controller gain was investigated by increasing the sample time and the sensitivity of the parameter estimator [Σ_0 =.0001]. Here, the disturbance is applied every five minutes from time t=55 to t=80. The increment of the sample time to 20 secs. made no difference; however, the change in the sensitivity of the parameter estimator caused the controller gain to detune even more. The level, in this case, was affected by large deviations from its reference value.

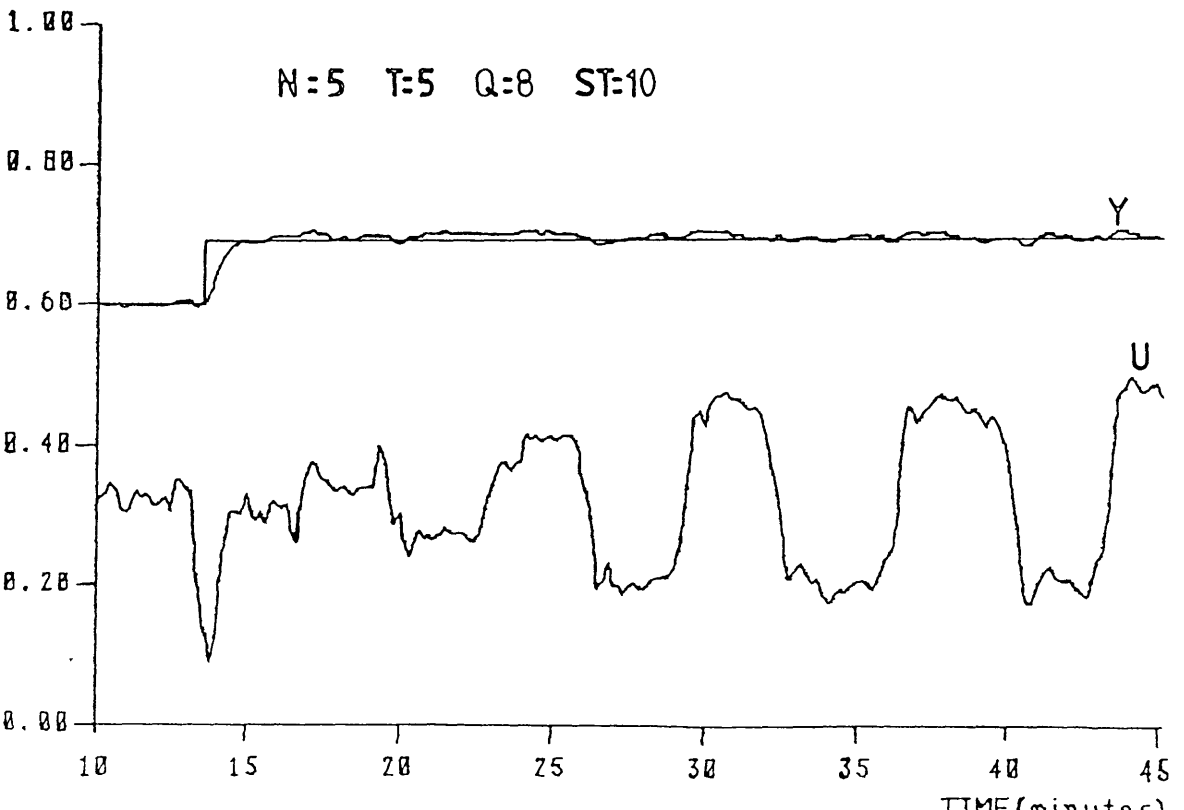
Figure (5.12) shows that by increasing the horizon T and the disturbance model order q, the magnitude of the problem can be reduced. The disturbance changes by steps every five minutes from time t=45 to t=100. However, the reciprocal of the gain still drifts to zero, although more slowly. In this experiment, Σ_0 was set to .001 . It can also be seen in Fig. (5.12) that the adaptive controller easily tunes itself when set point changes are applied.

The situation can be significantly improved if larger information content and model orders are used. Figure (5.13) shows two experiments with similar initial settings. They only differ in the disturbance polynomial order used. The increased Σ_0, n and q produce a more stable response, compared with the previous experiments.

FIG 5.13:FB/FF CONTROL OF A DELAYED DISTURBANCE



a) Outlet flow valve position (U)
and liquid level (Y)



b) Outlet flow valve position (U)
and liquid level (Y)

ii) The fact that the disturbance [inlet flow] is affected by a large and variable time-delay, makes the FB/FF algorithm very sensitive to the selection of the q parameter [the order of the disturbance polynomial].

For example, Fig. (5.13.a) shows the result when the system is subjected to step changes in the inlet flow every six minutes from time t=70 to t=120. In this case, the disturbance model order was chosen to be 5. According to eq. (3.21) in section (3.1), the data vector of the parameter estimator contains 10 values of the disturbance signal, ranging from time t=k to t=k-q-T+1 or more specifically, from t=k to t=k-9. Considering that in this experiment the sample time is st=10 sec, the information available for disturbance parameter estimation range from t=0 to t=-90 sec, at each sample interval. As was mentioned in Chapter 4, the inlet flow has an associated delay that varies between 30 and 90 sec. If in this case, the inlet flow delay is 90 sec, then only one value of the disturbance signal will be significant in the data vector. In this situation, it is expected that the feedforward control action will not have enough information to produce an effective regulation. This is the case shown in Fig. (5.13.a), where the control appears oscillatory and sensitive.

The performance of the controlled system can be improved by increasing either the time horizon T or the disturbance model order q, in order to provide additional I/O data for the estimator. In the former case, an excessive number of coefficients of the control polynomial must be estimated and the control action is unnecessarily retarded. On the other hand, if the q parameter is increased, the time-response of the control is not affected, although a large number of coefficients of the disturbance polynomial need be estimated. In general, it is not advisable to estimate an excessive number of parameters [Ydstie, 1986], because The RLS algorithm may present numerical problems and the adaptation rate becomes slower. The number of parameters to estimate, however, can be reduced by fixing to zero the first d polynomial coefficients of the delayed disturbance if one is certain that there is a delay at least d time intervals.

In the experiment shown in Fig. (5.13.b), where the disturbance enters the system every 4 minutes from time t=20 to t=45, the model order of the disturbance process was increased to q=8. In this case the disturbance signal in the data vector ranges from t=-120 to t=0 sec, providing enough information to the feedforward control action to

produce an effective regulation. The figure shows a stable and smooth control action and a near perfect regulation of the liquid level.

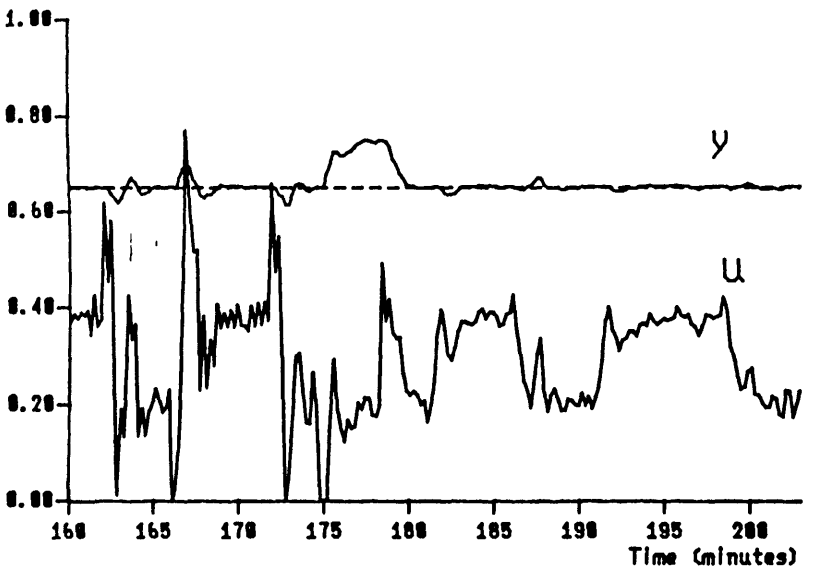
iii) As was earlier mentioned, the selection of the control strategy is very important in the final performance of a self-tuning algorithm. In particular, all the above experiments employed equal-increments extended-horizon control. Moreover, the majority of the results were obtained using this strategy. Initially, it was thought that the equal-increments strategy would be more robust in the incremental algorithm. Indeed, the equal-increments strategy should be robust in the non-adaptive case since the called for control action is more gradual and the controller gain is a summation of various terms:

$$d^{-1} = 1/\sum_{i=1}^{T}\beta_i \qquad (5.8)$$

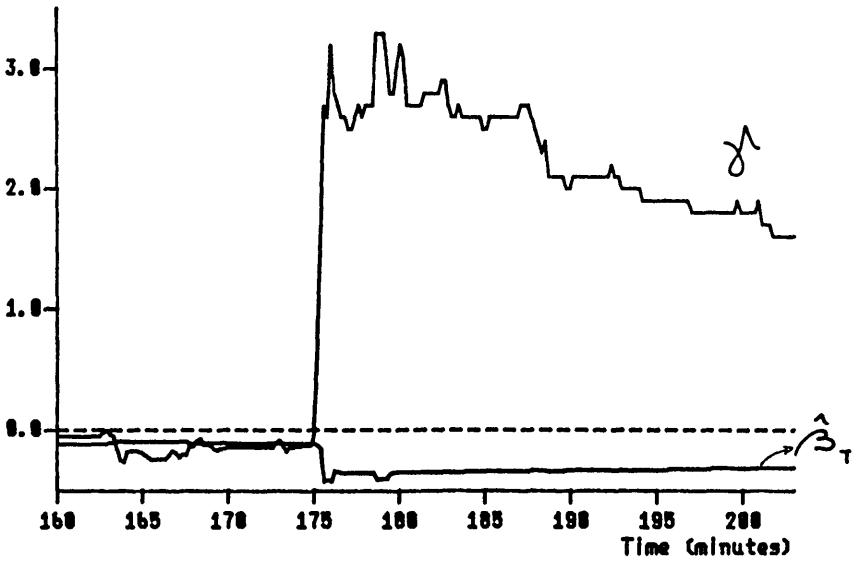
In this way, a smoother control is expected and no abrupt change in the closed-loop is artificially introduced by, for example, concentrating the control effort in the first increment. However, a problem arises in the adaptive case where the β parameters must be estimated. It has been observed, in simulations and pilot plant experiments, that the first $\hat{\beta}$ estimates [from one to T-1] in the vector $\underline{\beta}$ are slow to converge, are easily detuned and sometimes take a sign contrary to the sign of the true process; this is because these parameters, generally, receive poor excitation. Under this condition, the adaptive control-gain reaches very large values [small values of d], or even takes the opposite sign. This is the phenomenon shown by Figs. (5.9), (5.10) and (5.11).

The behaviour of these $\hat{\beta}$ estimates can be appreciated in the experiment shown in Fig. (5.14). In this experiment the level is controlled by the outlet valve and the system is continuously affected [from time t=160 to time t=200] by measured deterministic disturbances provided by inlet flow step changes with a frequency of 5 minutes. A constant control strategy was employed, which is equivalent to concentrating the incremental control effort in the first increment.

FIG 5.14: RECOVERY OF CONTROL AFTER COMPUTER CRASH
N=5 T=5 Q=8 ST=12

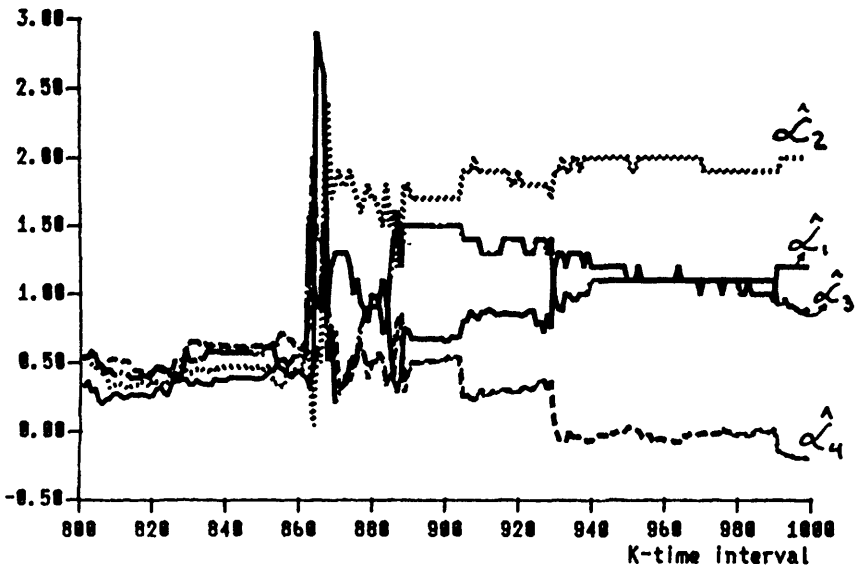


a) Outlet flow valve position (u)
and liquid level (y)

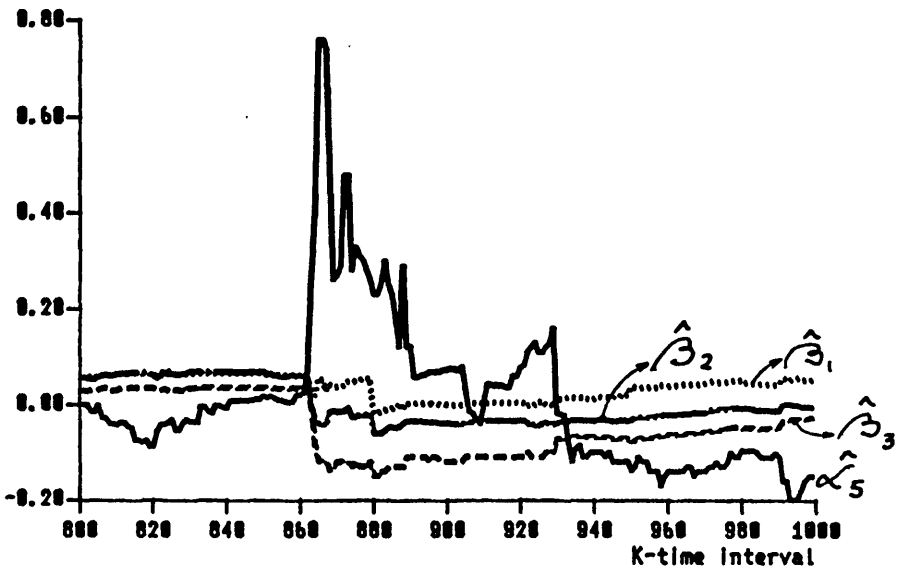


b) Reciprocal of the controller gain(β_T)
and feedforward compensation term (γ)

FIG 5.14: RECOVERY OF CONTROL AFTER COMPUTER CRASH
N=5 T=5 Q=8 ST=12

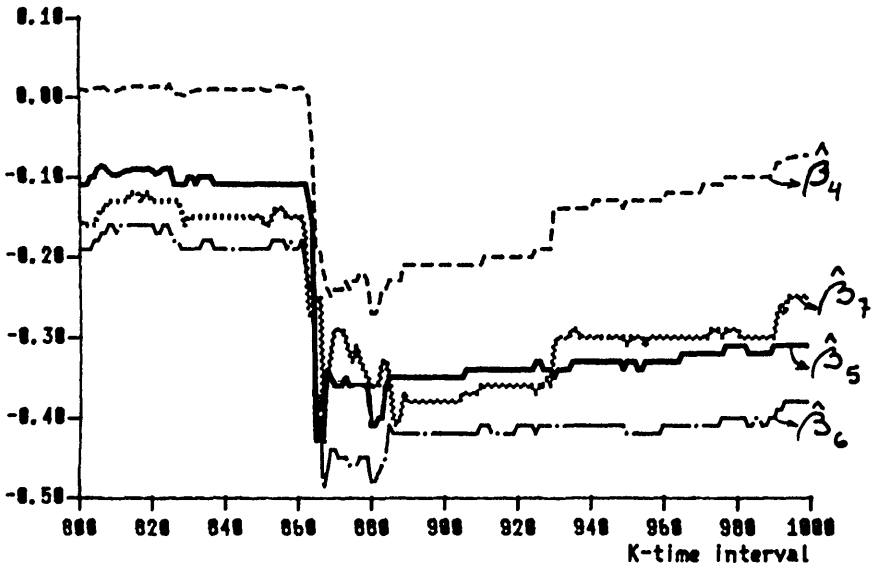


c) Predictive model parameter estimates

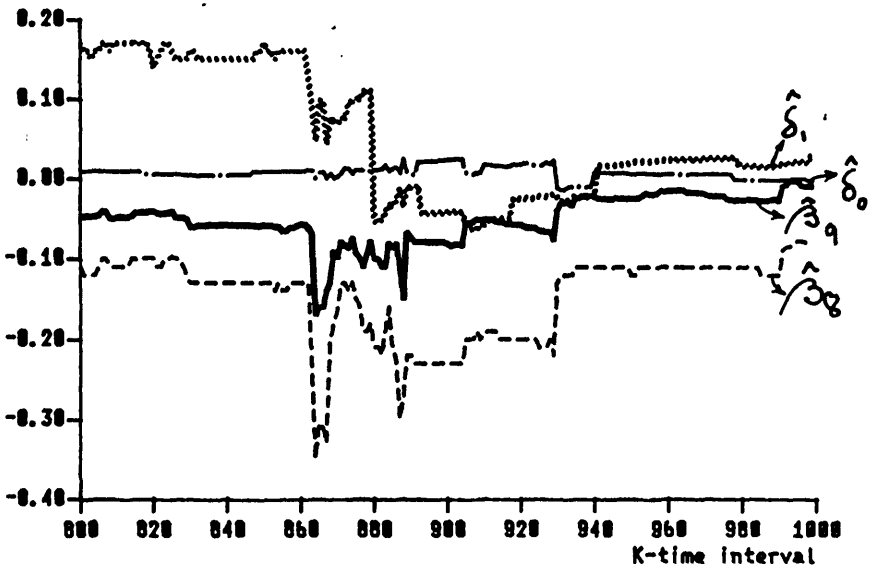


d) Predictive model parameter estimates

FIG 5.14: RECOVERY OF CONTROL AFTER COMPUTER CRASH
N=5 T=5 Q=8 ST=12

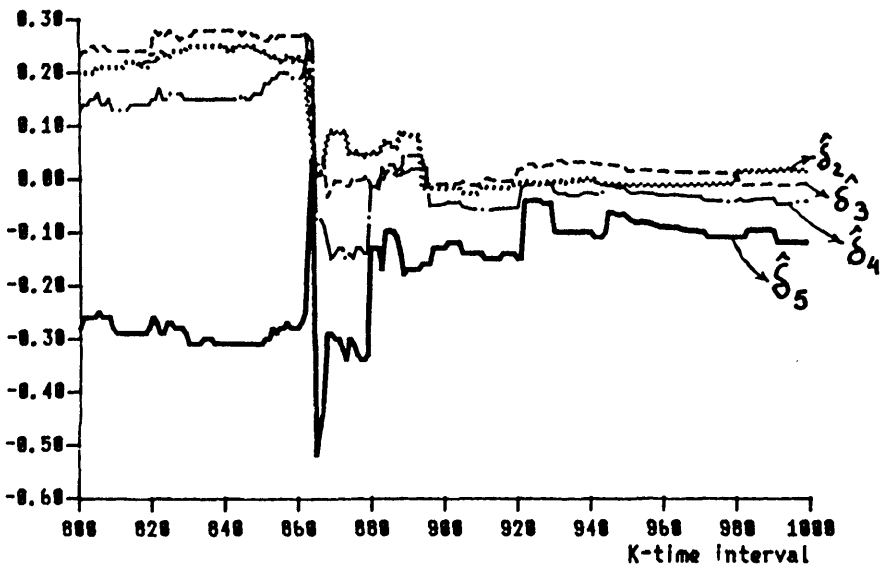


e) Predictive model parameter estimates

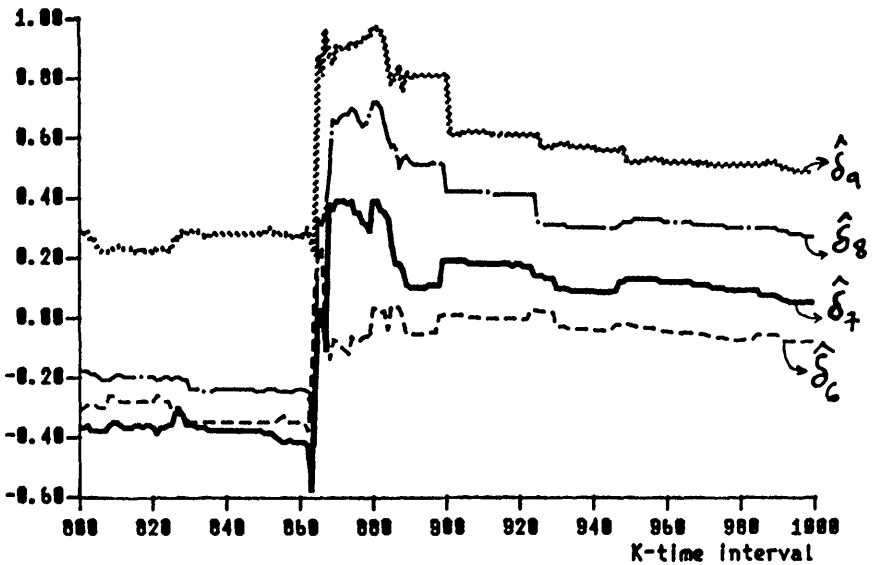


f) Predictive model parameter estimates

FIG 5.14 : RECOVERY OF CONTROL AFTER COMPUTER CRASH
N=5 T=5 Q=8 ST=12

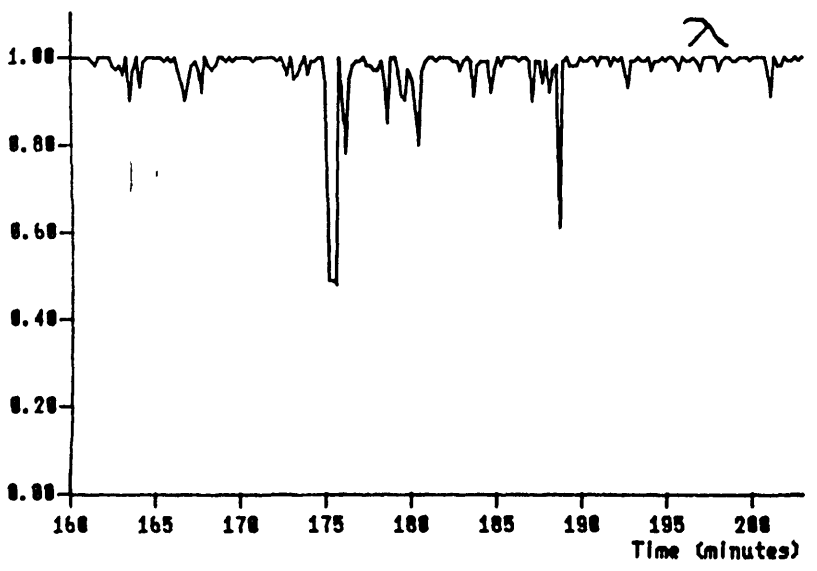


g) Predictive model parameter estimates

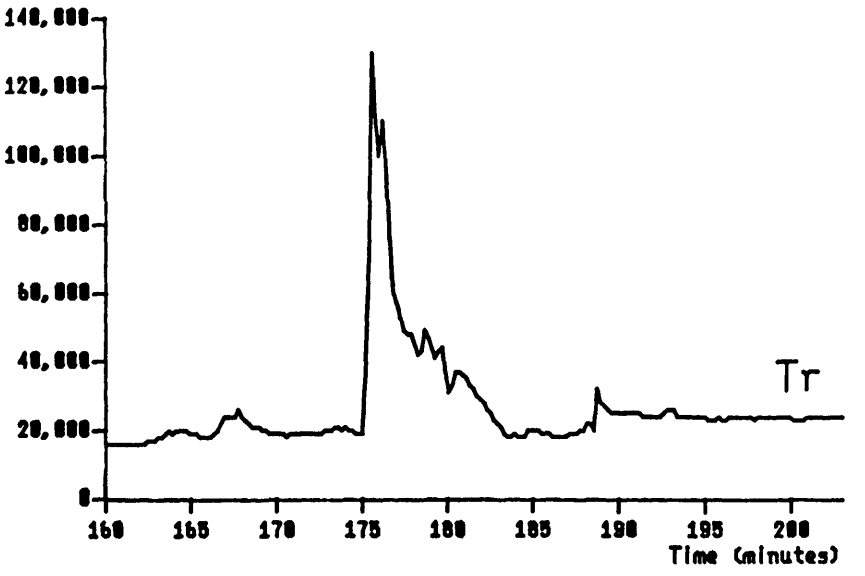


h) Predictive model parameter estimates

FIG 5.14: RECOVERY OF CONTROL AFTER COMPUTER CRASH
N=5 T=5 Q=8 ST=12



i) Forgetting factor



j) Trace of the covariance matrix

Figure (5.14) shows the performance of the controlled plant just after the recovery from a failure in the data acquisition system. From time t=155 to t=160 [780<k<800] the analogic system [see Chapter 4] did not send or receive any information from the plant, so that the estimation algorithm was fed with incorrect data. In this situation, the parameter estimates converged to unsuitable values.

It can be seen in Fig. (5.14.a) that for 160<t<175 [800<K<860] the closed-loop system remains stable despite the unsuitable parameter estimates and the load changes. However, the level presents significant deviations from the set point and the control action looks oscillatory and sensitive.

It is interesting to see in Figs. (5.14.d), (5.14.e) and (5.14.f) the evolution of the $\hat{\beta}$ estimates. After the recovery [t=160, k=88], the first $\hat{\beta}$ estimates [$\hat{\beta}_1$, $\hat{\beta}_2$, $\hat{\beta}_3$ and $\hat{\beta}_4$] have a positive sign, despite the fact that the controller gain should be negative. The rest of the $\hat{\beta}$ estimates have always a negative sign. Clearly there is more uncertainty associated with the first (T-1) parameters [some should actually be zero] and hence they should be given little or no weight in the controller. Had the equal-increments control strategy been employed [giving equal weight to all the T first $\hat{\beta}$ estimates], the results would have been catastrophic because after the recovery the controller gain would have been positive, giving positive feedback. This behaviour of the $\hat{\beta}$ estimates has been observed in many other experiments and simulations. Any adaptive control strategy which uses these estimates is likely to present problems. For this reason, the equal increments strategy was abandoned and replaced by the more robust constant-control strategy.

Autotuning of the Adaptive Controller

To improve the performance of the closed-loop system [Fig. (5.14.a)], at times t=175 and t=180, set points changes were introduced. These changes made the estimates converge to more suitable values so that the controller gain $1/\hat{\beta}_T$ and the feedforward term γ [Fig. (5.14.b)] were autotuned by the adaptive algorithm. The re-tuning caused by the set point changes was so effective, that almost perfect regulation was achieved [Fig. (5.14.a)] despite the load changes. The forgetting factor and the trace of the covariance matrix [Figs. (5.14.i) and (5.14.j)] also reflect the re-tuning of the parameter estimates by a considerable change shown at time t=175. This

change denotes the lack of confidence in the previous estimates. In this way these estimates are "forgotten" by giving more weight to the new information coming in. Wellstead and Zanker (1982) pointed out that "the self-tuning controller optimizes itself according to system input and disturbance signal. The larger these signals are, the faster the tuning-in process is". However, changes in the measured disturbance signal do not appear to posses the same tuning ability of the set point changes [see Fig. (5.14a) from time t=160 to time t=175]. Many simulations and experiments have shown that set point changes are an effective tuning tool, which is not the case with changes in the measured disturbance. The reasons for this do not appear clearly explained in the literature.

It is worth noting the evolution of the $\hat{\delta}$ parameter estimates in Figs. (5.14.f), (5.14.g) and (5.14.h). The first $\hat{\delta}$'s finally converged to near zero values, denoting the large delay affecting the measured disturbance. It is this ability to detect and estimate the delay associated with the disturbance that makes the FB/FF self-tuner compensate for the load changes at the appropriate moment. In this manner, once the algorithm has converged, practically no deviation from the set point is observed in the level in Fig. (5.14.a).

Initialization

Several initialization procedures have been studied in our experiments. Hiram & Kershenbaum (1985) suggested that a constant control could be applied for enough sample times to fill the data vector with non-zero values after which the estimation and the adaptive control algorithms could be simultaneously started. This procedure works well with positional self-tuners, but it fails with incremental algorithms because, here, a constant control procedure essentially fills the data vector with zero increments under near SS operating conditions.

The initialization procedure commonly used in our experiments with incremental algorithms is as follows:

i) a pseudo random binary sequence [PRBS] is applied to the control input [Golumb, 1981] during TI sample intervals with an amplitude of EPS*u_o, where u_o is the initial input.

ii) during all the initialization period, the parameter estimation algorithm is switched-on and the control is switched-off.

iii) when the PRBS sequence is completed, the adaptive control is switched-on.

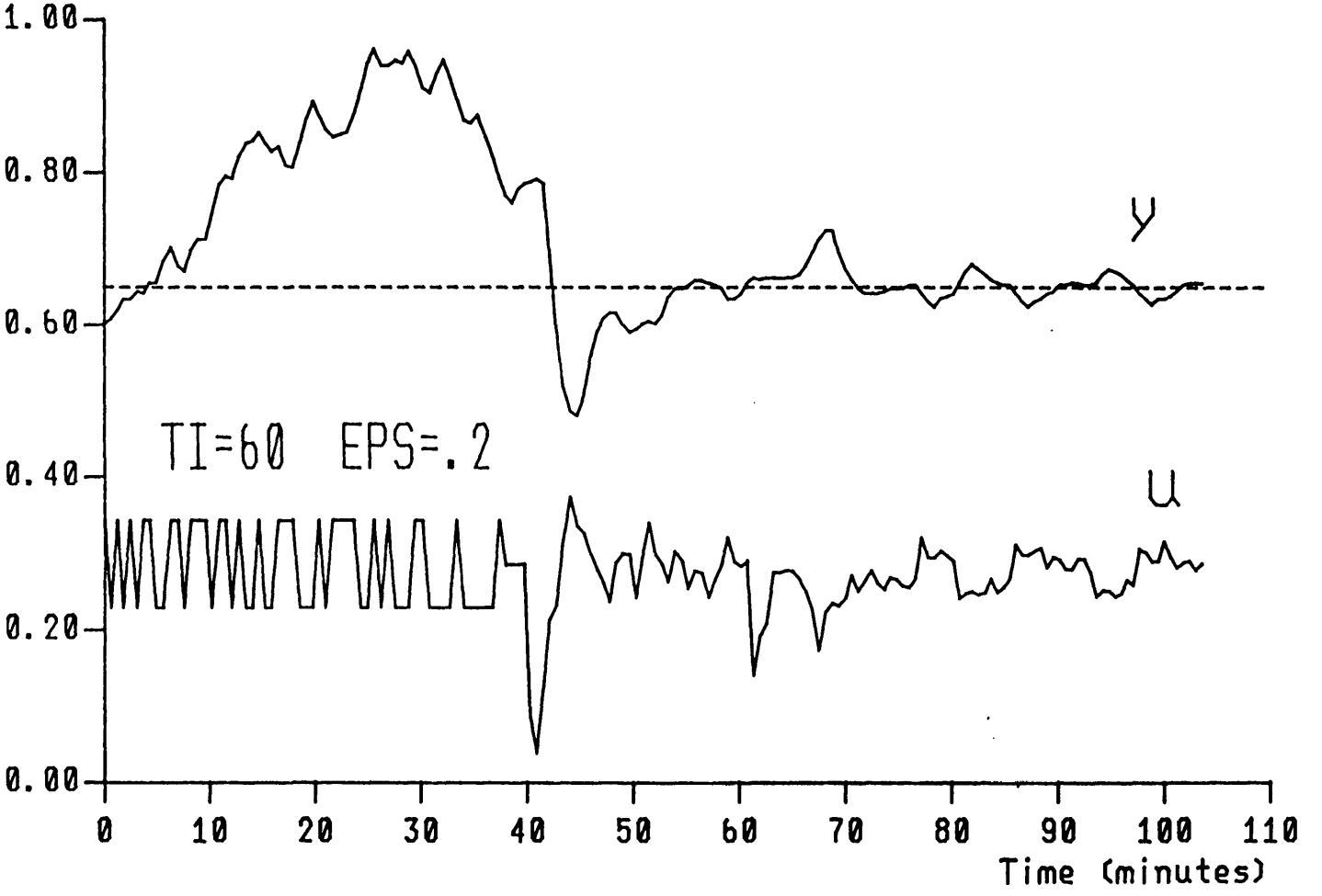
In this way it is ensured that the adaptive control starts with a non-zero data vector.

A typical start-up is illustrated in Fig. (5.15) in which the level is controlled by the inlet valve [see Fig. (4.2)]. Because of the packing in the absorption column, the control action is delayed by 2 or 3 sample intervals [st=35sec]. The system is continuously disturbed every 5 min, from time t=65 to t=95, by changing the position of the outlet valve. A model order n=3, a control-horizon T=5, an order of the disturbance model q=3 and a sample time st=35 seconds were chosen. The information content Σ_o, which defines the speed of adaptation, was set to .1 . The initial model parameter estimates were all set to zero.

In this experiment [Fig. (5.15)], the initialization procedure is given by a PRBS of TI=60 sample intervals with an amplitude of .2*.25 [EPS*u_o]. It can be seen in Fig. (5.15.b) that at the beginning, all the parameter estimates [reflected in $\hat{\beta}_T$ and γ] vary quite drastically. After 15 sample times the variations in $\hat{\beta}_T$ are significantly reduced, this is because at time t=15 the control data vector is filled with non-zero values; before the initialization ends this parameter converged to its final value. On the other hand, the feedforward term γ converges to a suitable value only when the disturbance starts entering into the system [t=65]. This is expected since only at that moment the parameters associated with the disturbance model start to receive information. At time t=70 the output suffers a considerable excursion from its reference, when the system is disturbed for the first time. However, after the convergence of the feedforward term γ, the output remained closer to the set point. Perfect control is not achieved because of the time-delay in the control.

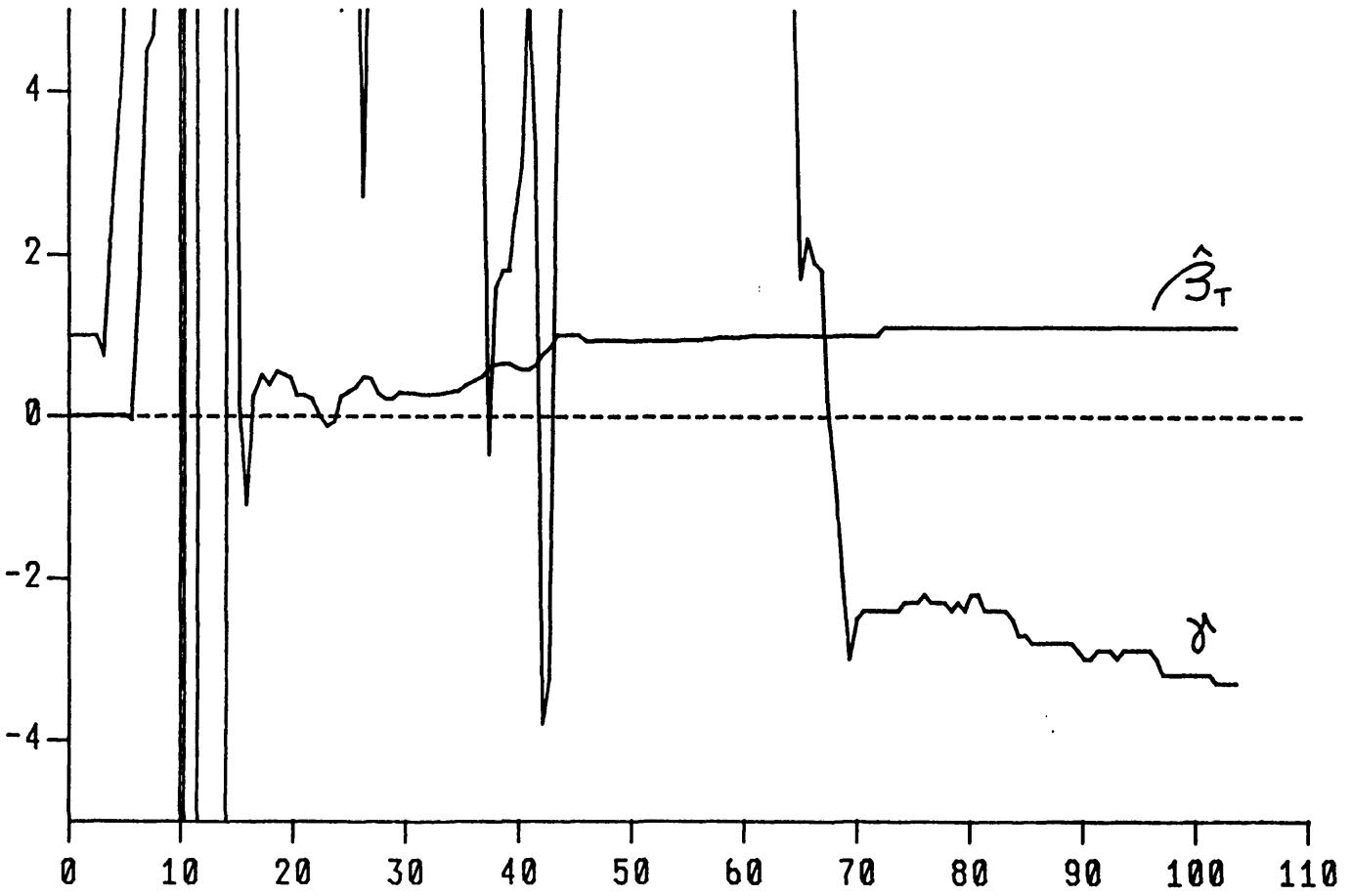
Figures (5.15.c) and (5.15.d) show more clearly the way the parameter estimates converge. The trace Tr of the covariance matrix [of all the parameter estimates], which is proportional to the square of the summation of the estimation errors, suffers a drastic change around time t=15. This change denotes an increment in the estimation confidence. At time 15 the data vector is filled with non-zero values, this causes non-zero parameter estimates which produce smaller estimation errors and consequently, a smaller trace Tr. At the same time [t=15], the parameter λ also varies, "forgetting" the old information based on inadequate parameter estimates, giving more weight to the new data.

FIG 5.15: FB/FF CONTROL OF A DELAYED SYSTEM
N=4 T=5 Q=3 ST=35



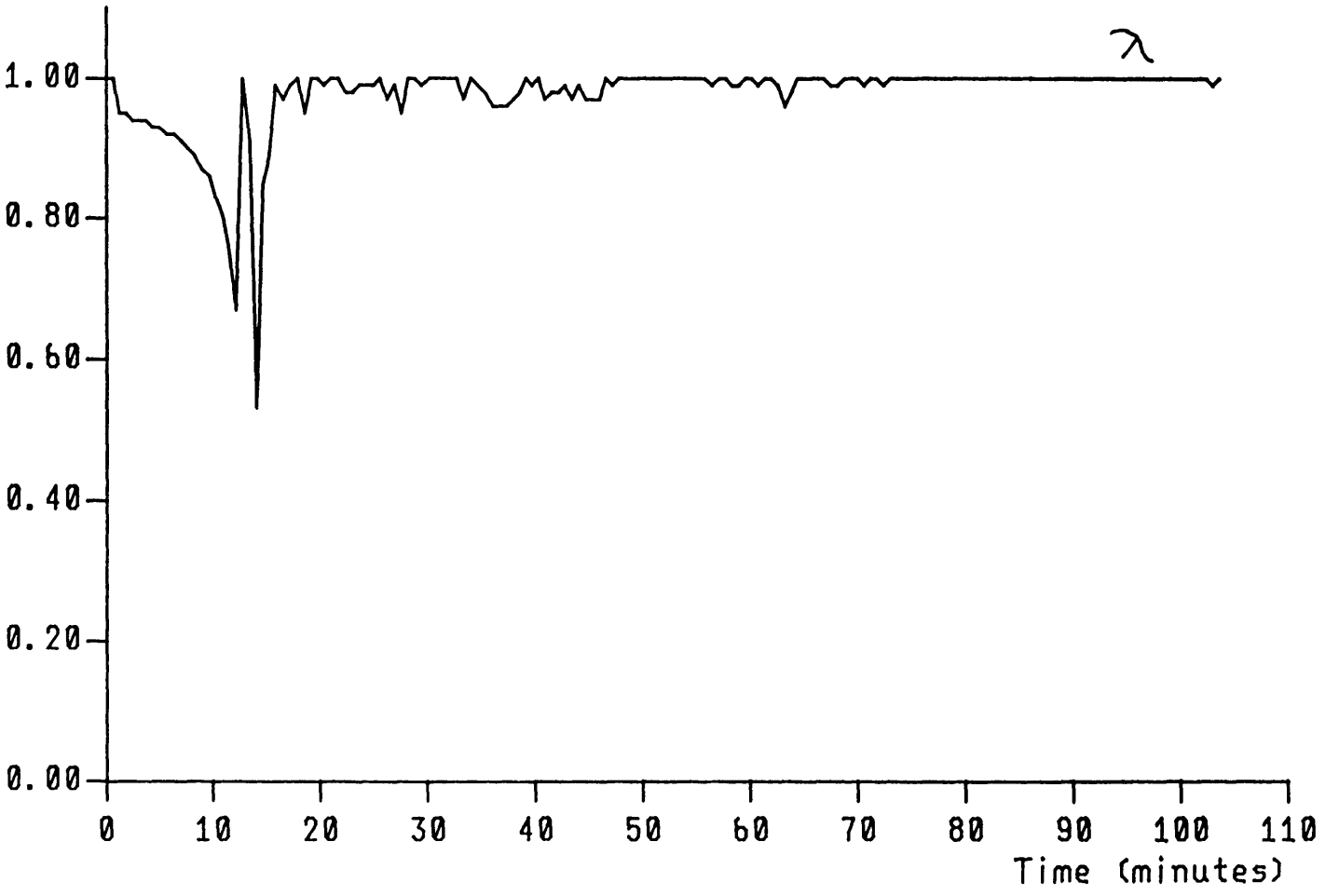
a) Inlet flow valve position (u)
and liquid level (y)

FIG 5.15: FB/FF CONTROL OF A DELAYED SYSTEM
N=4 T=5 Q=3 ST=35



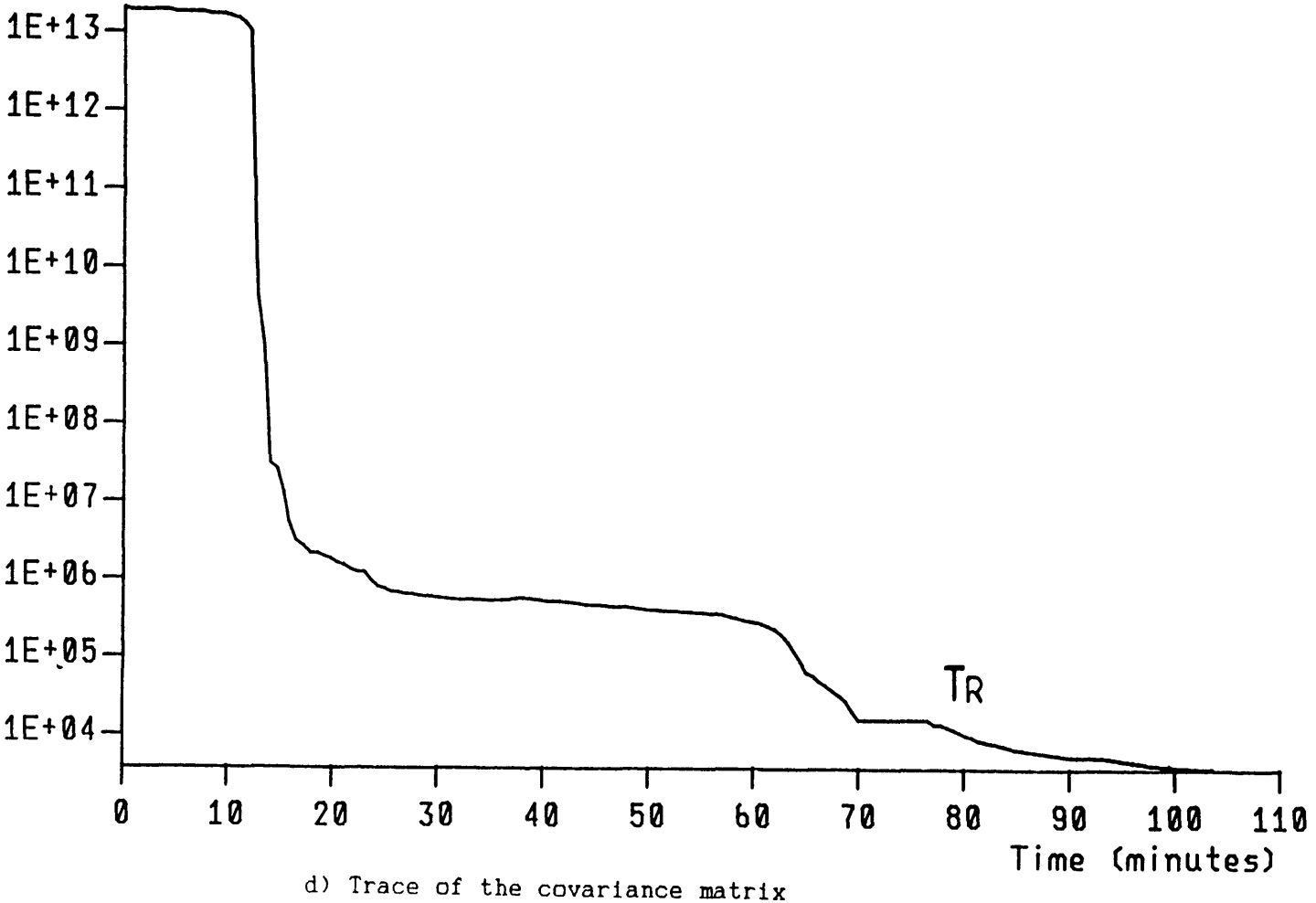
b) Reciprocal of the controller gain ($\hat{\beta}_T$)
and feedforward compensation term (γ)

FIG 5.15: FB/FF CONTROL OF A DELAYED SYSTEM
N=4 T=5 Q=3 ST=35



c) Forgetting factor

FIG 5.15: FB/FF CONTROL OF A DELAYED SYSTEM
N=4 T=5 Q=3 ST=35



d) Trace of the covariance matrix

The trace changes again when the system is perturbed for the first time [t=65]. In this case, the parameter estimates related with the measured disturbance converge to suitable values which allow the adaptive regulator to keep the output closer to its reference value.

In general, the algorithm converges fairly quickly and is able to control the level in spite of the disturbance and the control delay.

To test the robustness of the proposed self-tuner, the same experiment of Fig. (5.15) was repeated with lower values of the initial settings. The results shown in Fig. (5.16) were obtained with n=3, T=4, q=1, and Σ_o=.01. There is not much difference in the control performance.

Based on many simulations and experiments, it can be said that the constant control strategy is very robust to the selection of initial settings. If the system has associated a control-delay, it only needs a careful selection the model order n and the time-horizon T. In this case, the model order should be larger than 2 [depending on the time-delay] and the time-horizon should be larger than the assumed control-delay. On the other hand, if the disturbance is affected by any delay, the disturbance order q must be carefully chosen so that enough relevant disturbance values are included in the data vector.

The selection of the information length Σ_o is, generally, not sensitive. This value can be safely chosen in a range of 2 orders of magnitude around the standard deviation of the process noise. Too small a value could cause unexpected "bursts" in the estimator. A very large value freezes the estimation stopping the adaptivity of the controller.

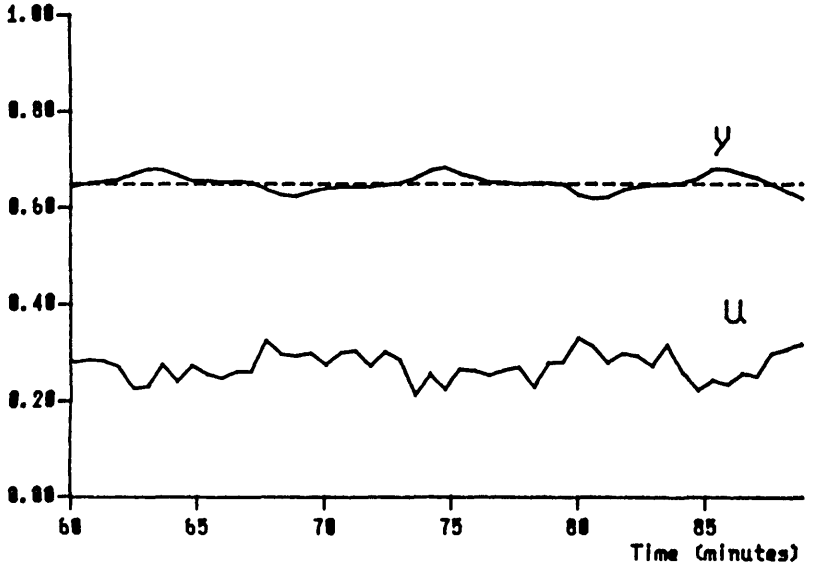
Direct Transmission

In section (3.1.2) it was pointed out that a disturbance measured at time k may have already affected the output at time k. Because of this, eq. (3.21) considers direct transmission between the measured disturbance w_k and the output y_k.

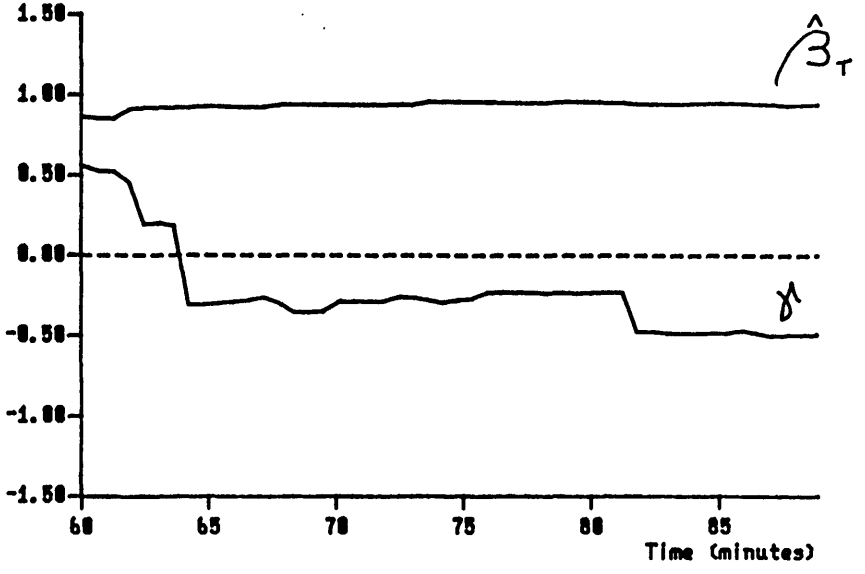
The following first order linear model was used in simulations in order to show the importance of modelling direct transmission:

$$y_k = .64 \cdot y_{k-1} + .5 \cdot u_{k-1} + w_k \qquad (5.9)$$

FIG 5.16: FB/FF CONTROL OF A DELAYED SYSTEM
N=3 T=4 Q=1 ST=35



a) Intlet flow valve position (u)
and liquid level (y)



b) Reciprocal of the controller gain ($\hat{\beta}_T$)
and feedforward compensation term (γ)

The measured disturbance w_k is a square wave that takes the values 1 and 1.2 and immediately affects the output y_k. The importance of this effect is shown in Fig. (5.17) where direct transmission is not modelled, namely dw is assumed 1 in eq. (3.17). In the simulation the initial settings were n=q=T=1.

It can be seen that at the beginning the control is good despite some variations in the control gain $1/\hat{\beta}_T$, but eventually the variations in the parameter estimates, reflected in $\hat{\beta}_T$ and γ, make the algorithm explode.

The same model of eq. (5.9) was controlled with the regulator defined by Algorithm II of section (3.1.2), which considers direct transmission between the measured disturbance and the output [dw=0 in eq. (3.17)]. Figure (5.18) shows that the parameter estimates remain constant and good control is achieved in spite of the fact that the disturbance immediately affects the output. Similar results were obtained with many other simulations.

Setting dw=0 in eq. (3.17) can improve the robustness of the self-tuner when sample times are large and disturbances are likely to be felt between samples. Because of this, Algorithm II is preferred in general.

Detuning the Control Action

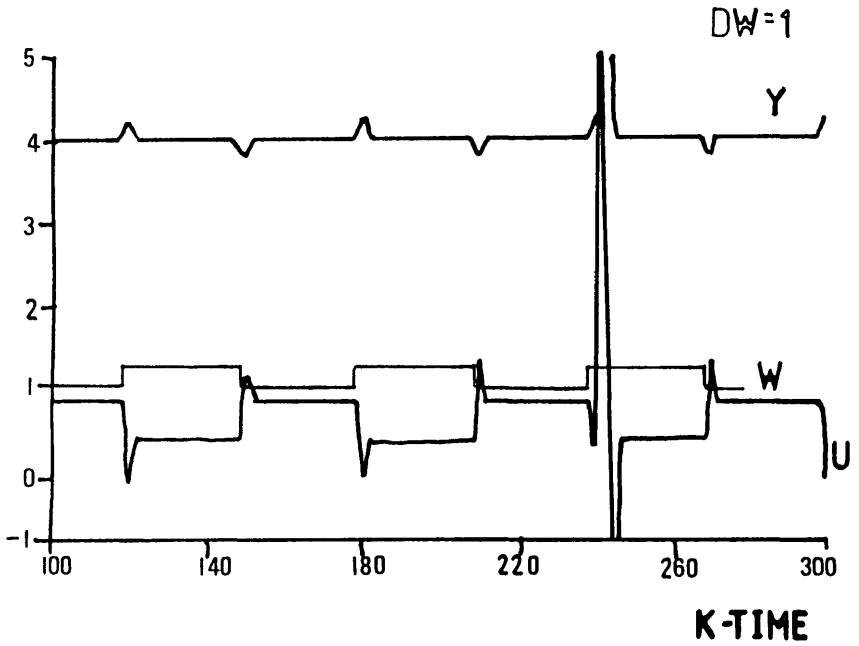
A serious potential problem with feedforward control is the strong control action usually required to obtain complete disturbance rejection. This phenomenon is shown in a simulation with the following third order process:

$$y_k = 1.2 \cdot y_{k-1} - .2 \cdot y_{k-2} - .02 \cdot y_{k-3} - u_{k-1} - .95 \cdot u_{k-2} - .02 \cdot u_{k-3} - w_{k-1} - .5 \cdot w_{k-2} - .08 \cdot w_{k-3} \qquad (5.9)$$

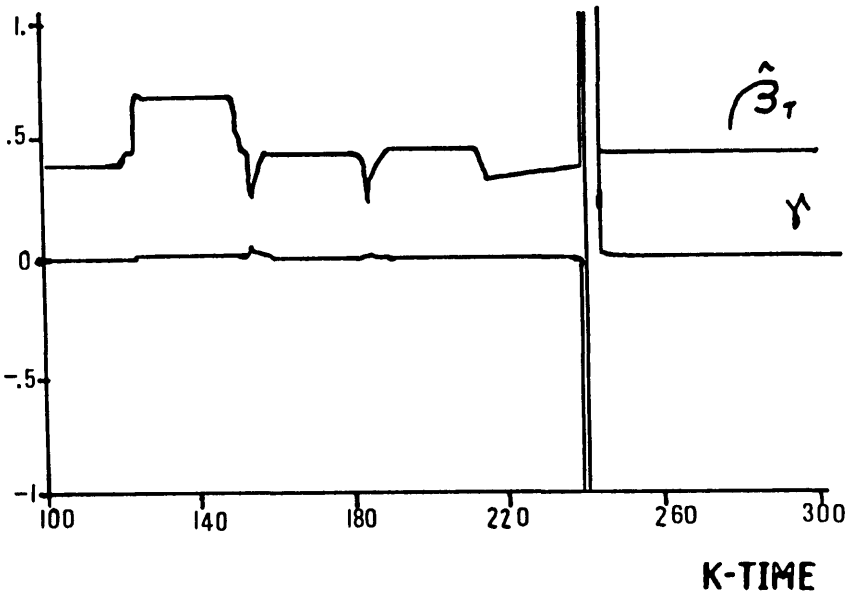
The poles of the model are P={-.0694, .296, .9763}, the zeros of the control polynomial are Zu={-.9285, .0215} and the zeros of the disturbance polynomial are Zw={-.25 +/- .132·i}

The measured disturbance is a square wave which varies between .95 and 1.25.

FIG 5.17 NO DIRECT TRANSMISSION MODELLED

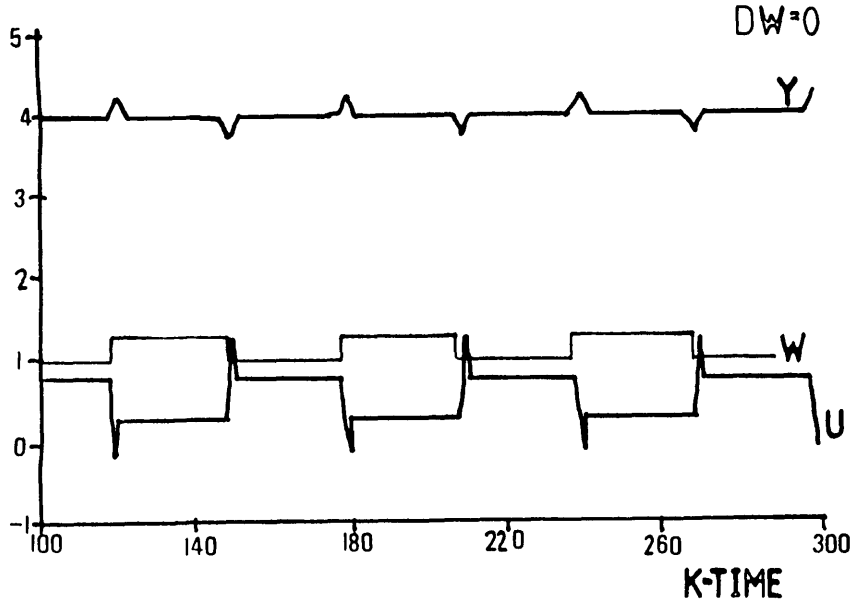


a) Control (U), output (Y) and disturbance (W)

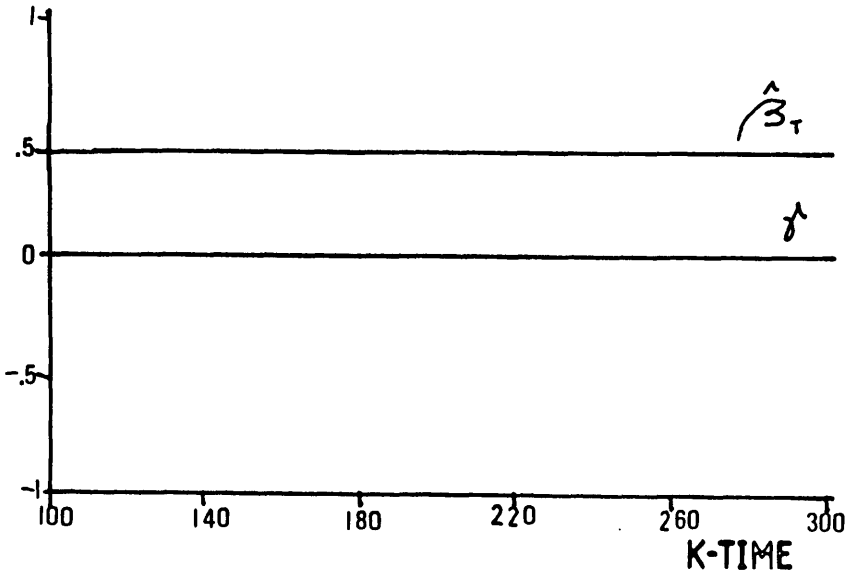


b) Reciprocal of the controller gain ($\hat{\beta}_T$) and feedforward compensation term(γ)

FIG 5.18: DIRECT TRANSMISSION MODELLED



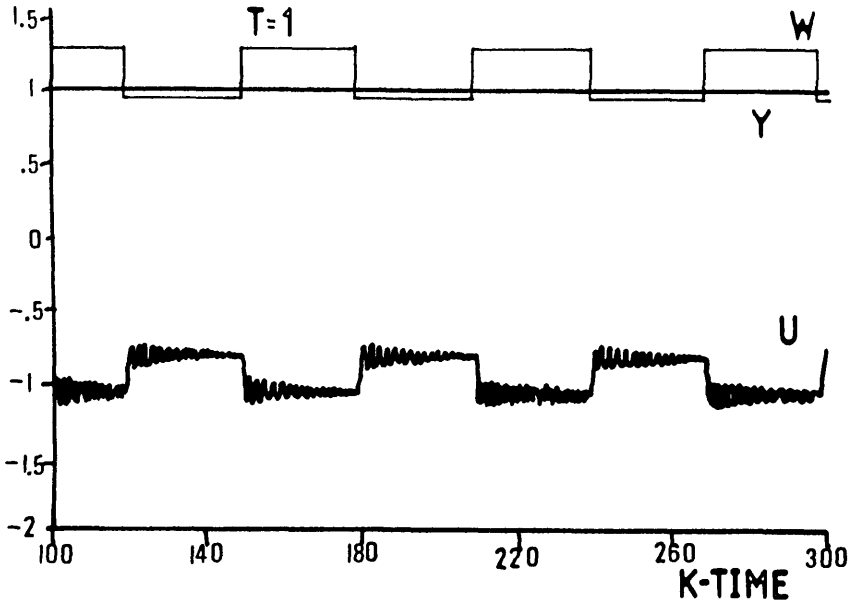
a) Control (U), output (Y) and disturbance (W)



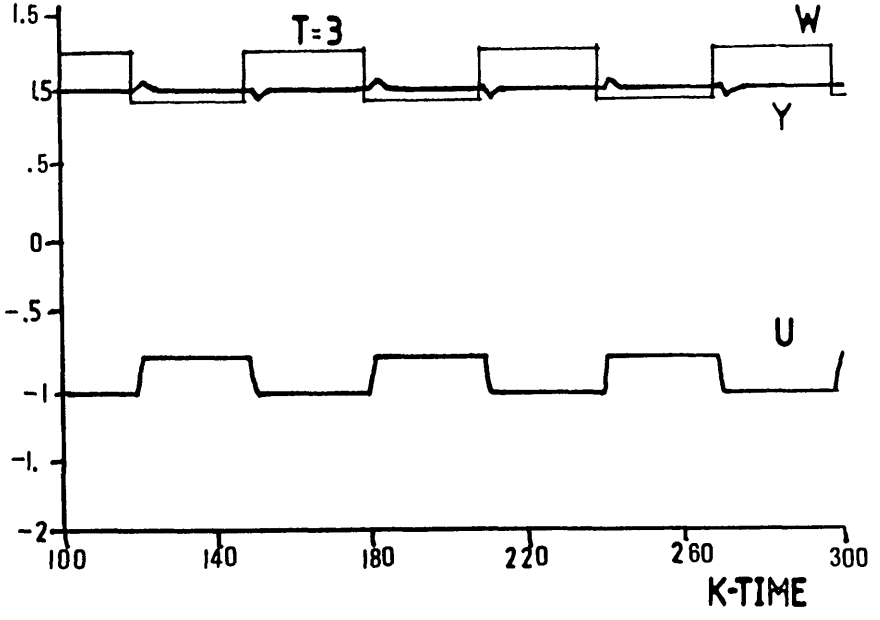
b) Reciprocal of the controller gain ($\hat{\beta}_T$) and feedforward compensation term($\hat{\gamma}$)

Figure (5.19) compares the performance of the controlled system when minimum-variance (T=1) and extended-horizon (T=3) strategies are used. The minimum-variance control achieves perfect regulation, however the control u rings excessively. On the other hand, by increasing the control-horizon to 3, the ringing in the control is avoided with minimum effects in the output. In the same way, saturations can be avoided by increasing the control-horizon. Thus, the same desirable property of the FB/EH controller (detuning) exists for the FB/FF EH controller. As it has been shown [Ydstie et al, 1985; Hiram & Kershenbaum, 1985], the detuning effect of the extended-horizon strategy can also make the controller cope with some non-minimum phase processes.

FIG 5.19: EXTENDED-HORIZON VS MINIMUM-VARIANCE CONTROL



a) Control (U), output (Y) and disturbance (W)



b) Control (U), output (Y) and disturbance (W)

5.2.- Multivariable State-Space Algorithms

This section is divided into three parts. The first part shows simulation results obtained with a positional feedback controller. Several two-input two-output linear systems of different characteristics have been used in the simulations. A study has been carried out on the problems of interaction, initial parameters selection, time-delays and change in model parameters. In the second part, a positional and an incremental FB controller are compared in systems characterized by unmodelled [constant and varying] bias. Simulations and experiments are shown. Finally, the third part presents simulation results obtained with a positional FB/FF state-space self-tuner. Different two-input two-output linear models were simulated. The algorithm was compared with a purely FB positional MIMO controller on systems subjected to load changes.

5.2.1.- MIMO State-space Feedback Control

A multivariable positional feedback self-tuning algorithm has been tested in quasi-nonlinear simulations with two-input two-output linear systems. The MIMO state-space approach was compared with two SISO polynomial self-tuners working in parallel; interactive and non-interactive models were used. The robustness of the state-space self-tuner against the selection of initial settings was tested; different values of filter gains, information lengths, structural indices and time-horizons were used. Results obtained with a system characterized by a general interactor matrix [see section (2.3)] are also shown.

As in the previous section, all the simulations have the rate of change of the control bound to 20% of the input range [as might well be done in an industrial environment].

In all the figures of this section the identification structure used in the simulation is given by N=n1,n2; where n1 and n2 correspond to the observability indices. Similarly the time horizons for both outputs are given by T=t1,t2. The number of the models used in the quasi-nonlinear simulations are also given in the figures; these are listed in Appendix H. In the simulations shown by Figs. 5.27, 5.28, 5.30, 5.31 and 5.36, the self-tuner algorithm was switched on at time k=50.

Two-input Two-output MIMO Controller vs Two SISO Controllers

First of all, a series of results obtained with quasi-non-linear simulations are presented to compare the performance of the state-space algorithm with two independent SISO polynomial self-tuners. In both cases positional algorithms were used. The quasi-non-linear simulations consist of simultaneous set point changes in both outputs accompanied with a change of the simulated model.

Figures (5.20) and (5.21) show the simulation results when model 1 is used and with the change of set point the plant is described by model 2. These models which are given in Appendix H, are stable minimum phase, steady-state non-interactive, linear and of third order. The output signals are contaminated with a zero-mean noise of variance .1.

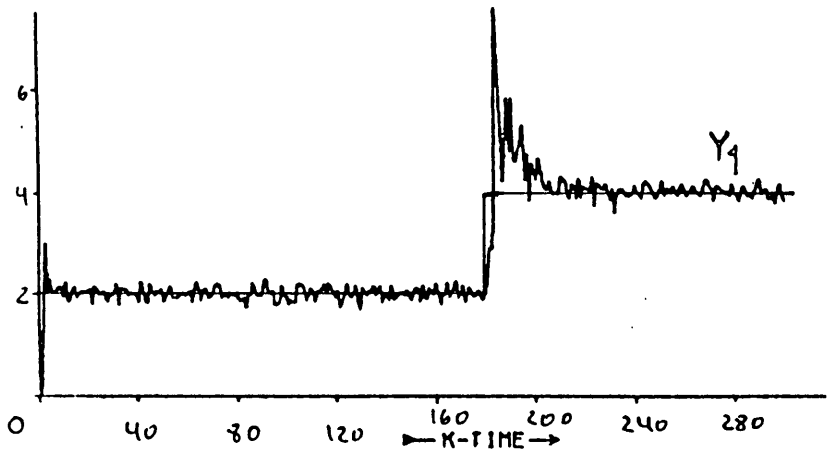
Figure (5.20) shows the response of the system controlled with two SISO STR's . Comparing Fig.(5.20) with Fig.(5.21), where the system is controlled by a MIMO state-space algorithm, it can be seen that in both cases a similar performance is achieved. The initial parameters were not optimally selected but adjusted well enough to produce an acceptable control. It is interesting, however, to see that the system can be controlled with the simplest possible model structure with the state-space algorithm [n_1=1, n_2=1], in fact models 1 and 2 were of third order. This is not possible, in this case, with two SISO controllers.

A more difficult problem is given by the simulations of Fig. (5.22) and (5.23), where interactive models are used. Before the set point changes the plant is described by model 3 and after the set point changes, it is described by model 4 of Appendix H. Both models are stable, minimum-phase and of third order, but steady-state interactive. The difference in performance is quite significant. Comparing Figs. (5.22.a) with (5.23.a) and (5.22.c) with (5.23.c), it is seen that the state-space algorithm presents much smaller overshoot, especially in the second output. It is worth noting that also in this simulation the simplest model structure was used in the state-space algorithm.

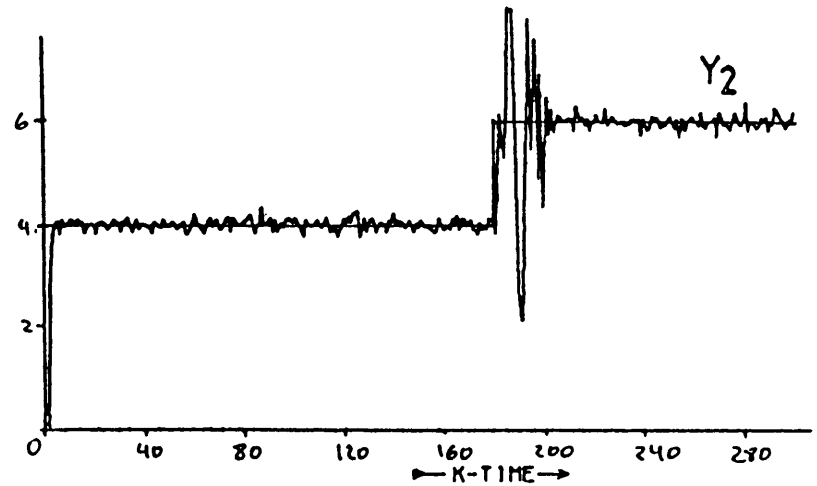
An even more difficult problem appears in the simulations of Figs. (5.24) and (5.25). The models 5 and 6 of Appendix H, used in these simulations, are both of fourth order so that the extra difficulty is given by the increased number of parameters that described the system. Moreover, model 6 is interactive. In this case a similar model structure was used in both control methods, however, the

FIG 5.20: 2 SISO POLYNOMIAL STRS

N=3,3 T=3,3 MODELS=1,2 NO INTERACTION



a) Output of the first polynomial STR

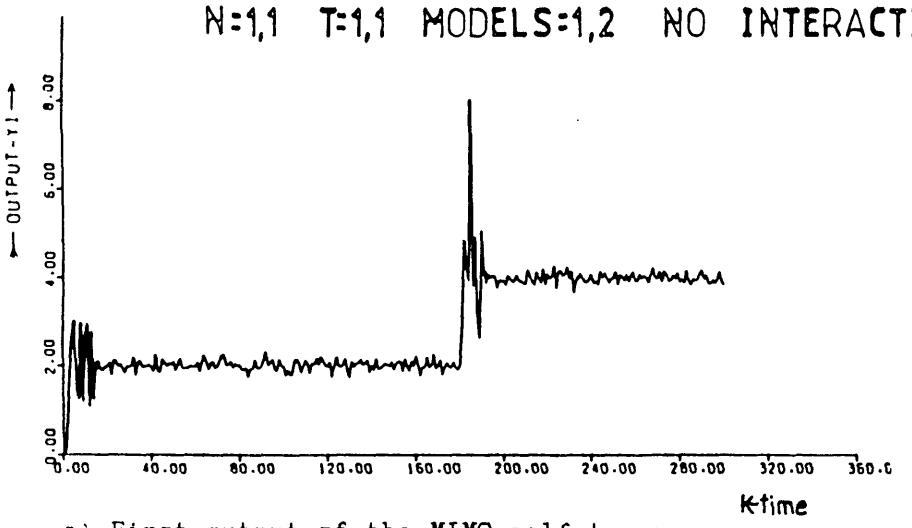


c) Output of the second polynomial STR

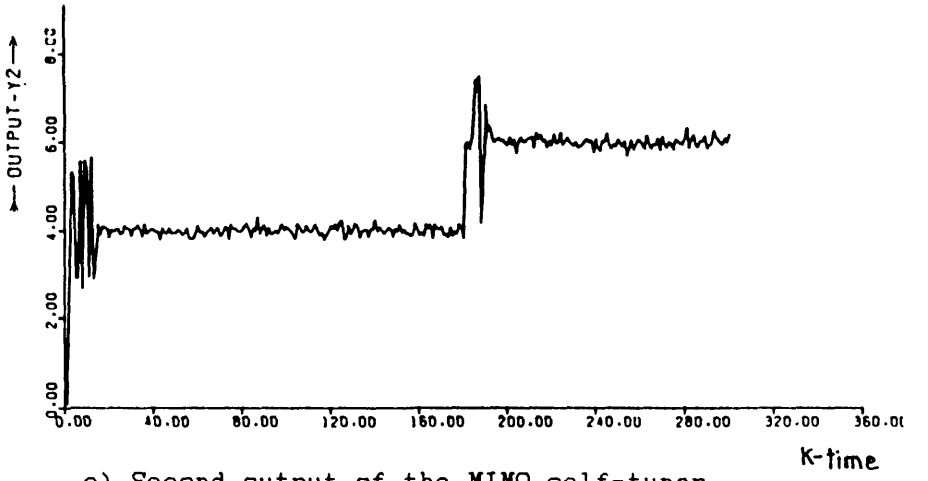
b) Input of the first polynomial STR

d) Input of the second polynomial STR

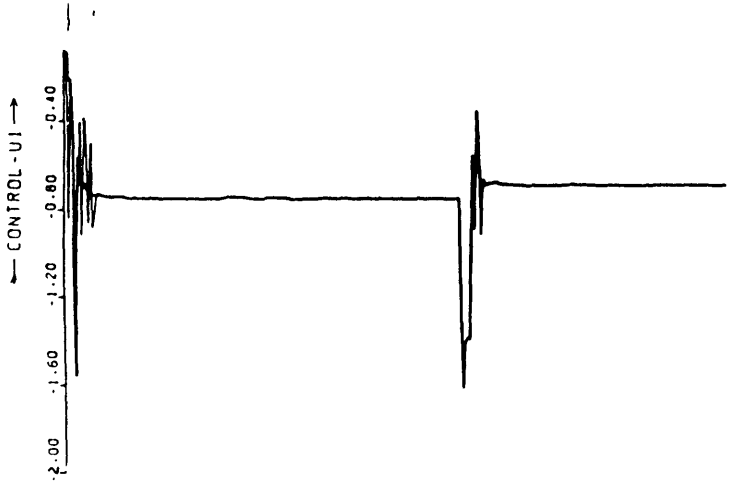
FIG 5.21: MIMO STATE-SPACE STR



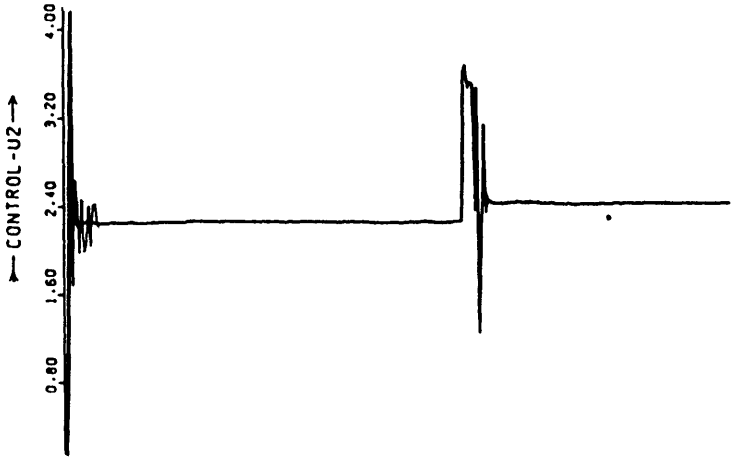
a) First output of the MIMO self-tuner



c) Second output of the MIMO self-tuner



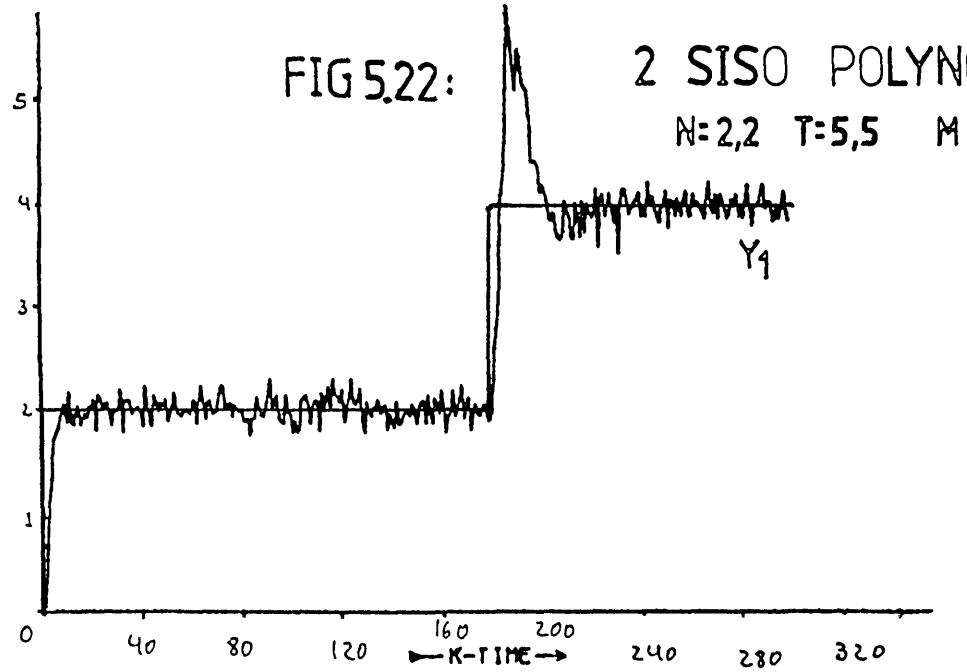
b) First input of the MIMO self-tuner



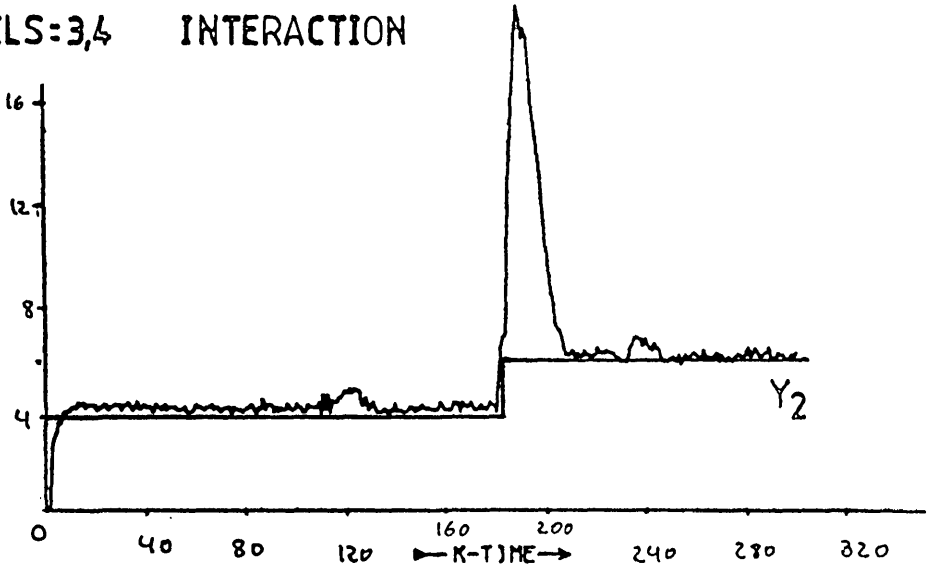
d) Second input of the MIMO self-tuner

FIG 5.22: 2 SISO POLYNOMIAL STRS

N=2,2 T=5,5 MODELS=3,4 INTERACTION



a) Output of the first polynomial STR



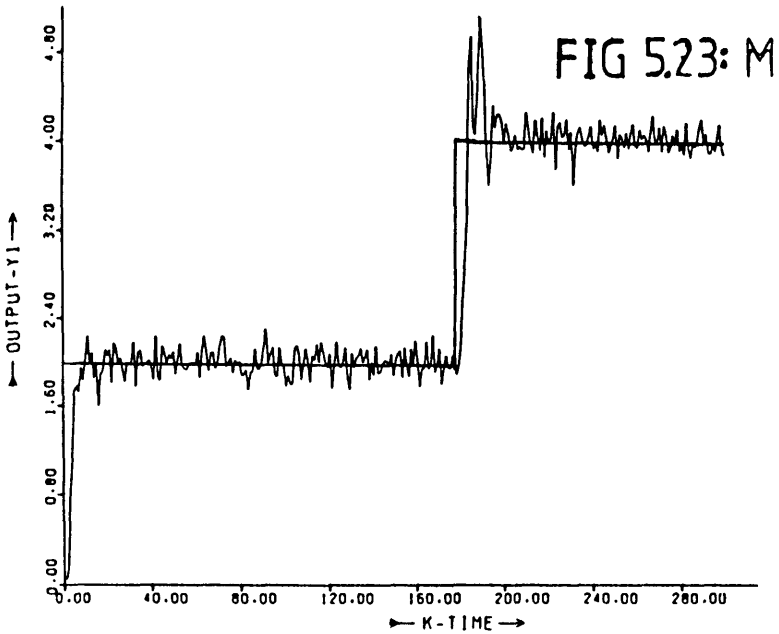
c) Output of the second polynomial STR

b) Input of the first polynomial STR

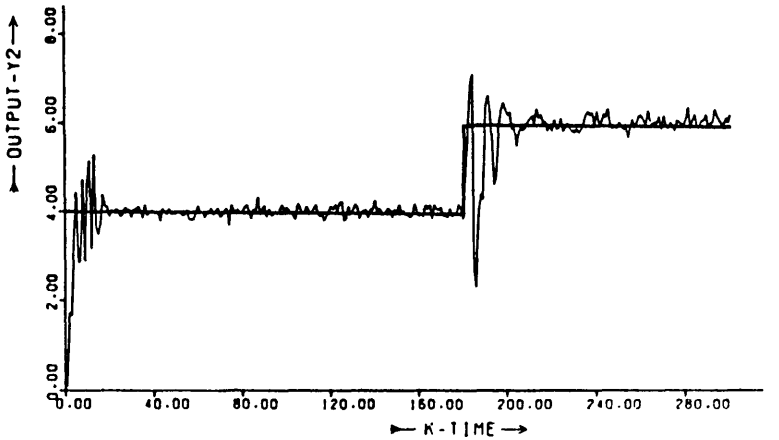
d) Input of the second polynomial STR

FIG 5.23: MIMO STATE-SPACE STR

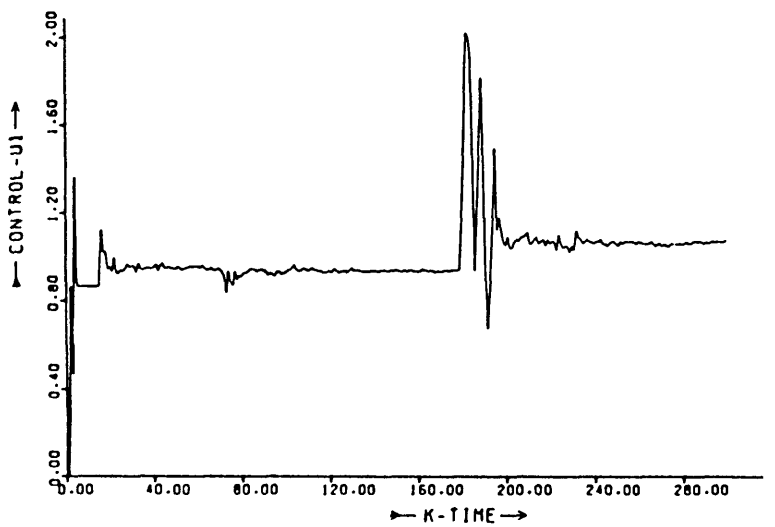
N:1,1 T:1,1 MODELS:3,4 INTERACTION



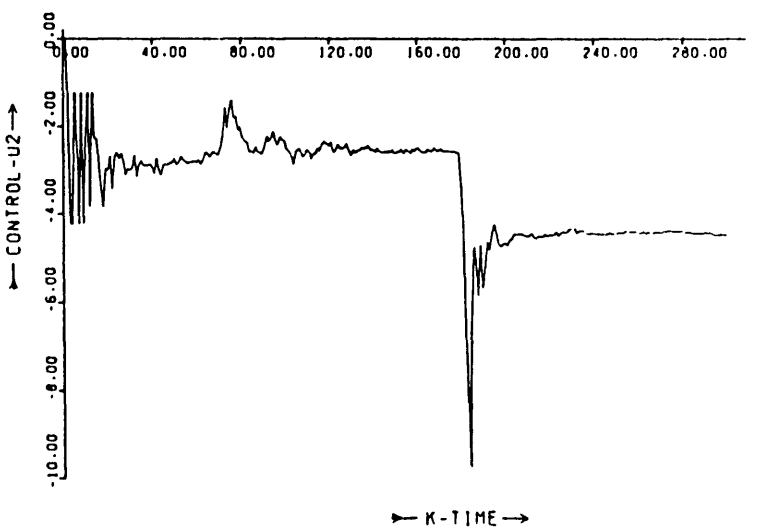
a) First output of the MIMO self-tuner



c) Second output of the MIMO self-tuner

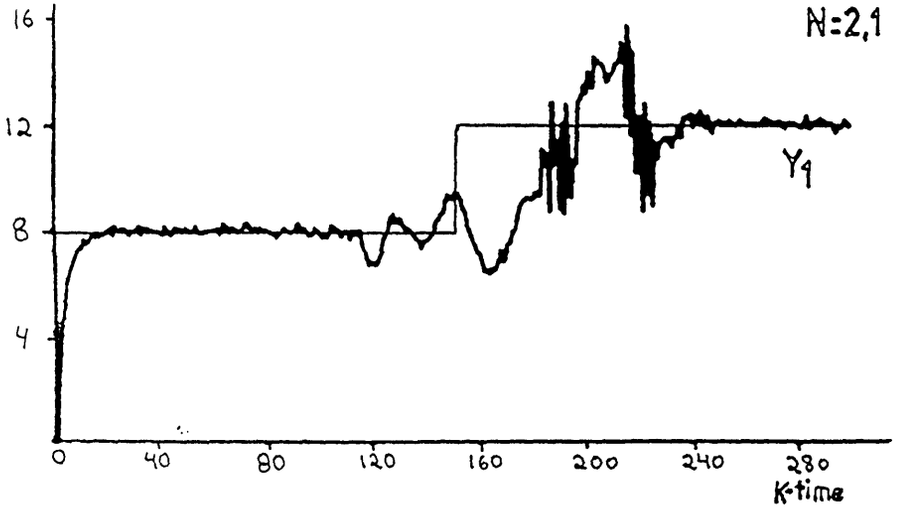


b) First input of the MIMO self-tuner

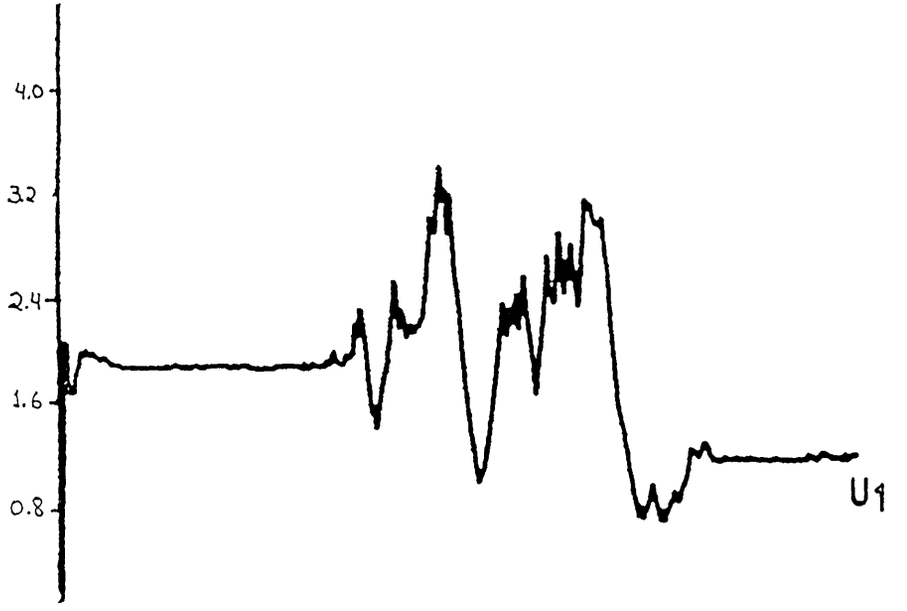


d) Second input of the MIMO self-tuner

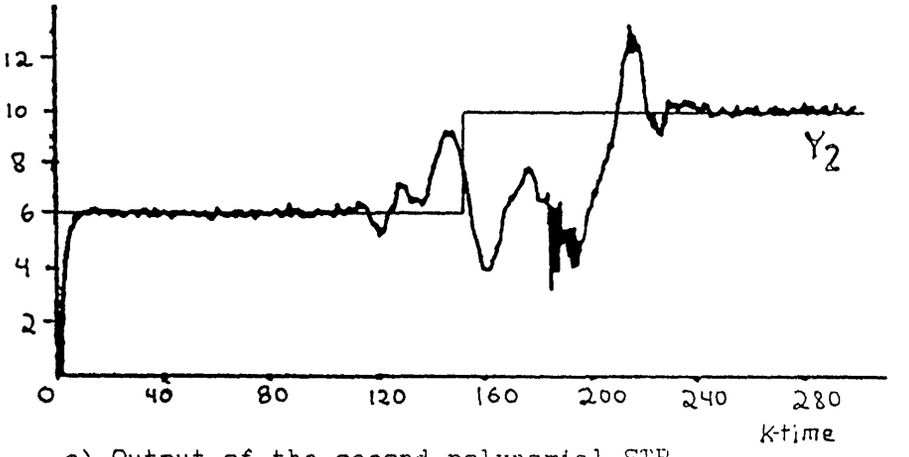
FIG 5.24: 2 SISO POLYNOMIAL STRS



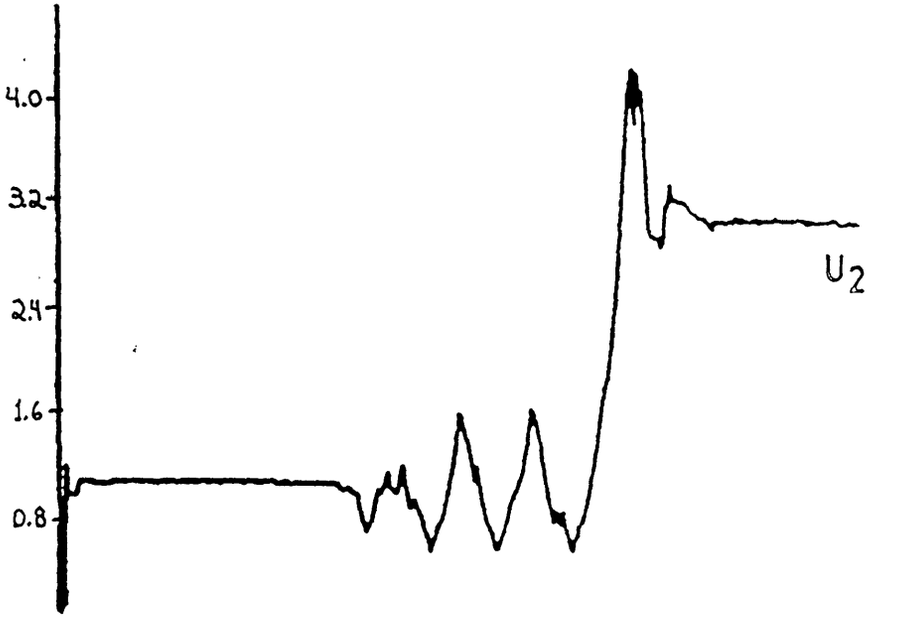
a) Output of the first polynomial STR



b) Input of the first polynomial STR



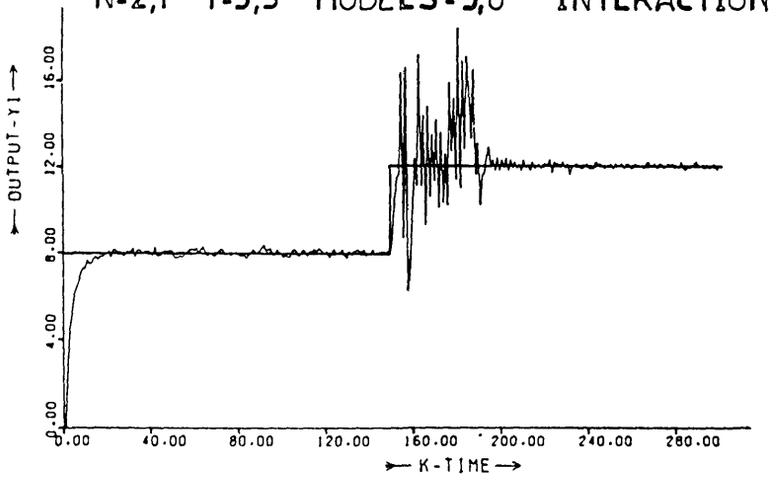
c) Output of the second polynomial STR



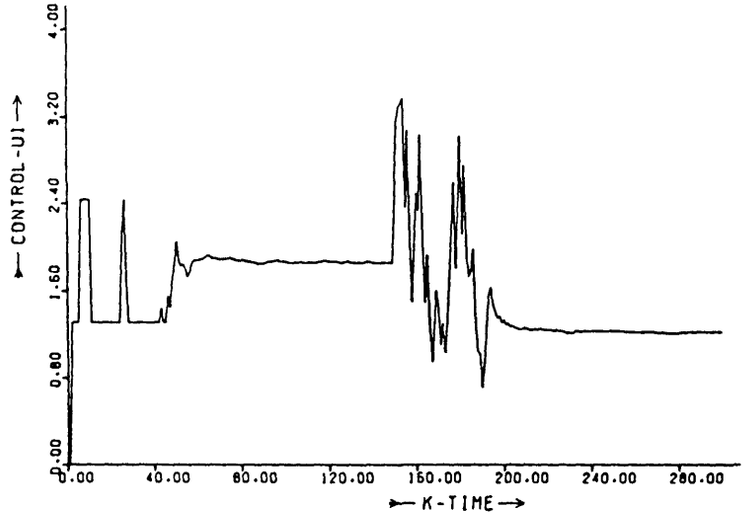
d) Input of the second polynomial STR

FIG 5.25: MIMO STATE-SPACE STR

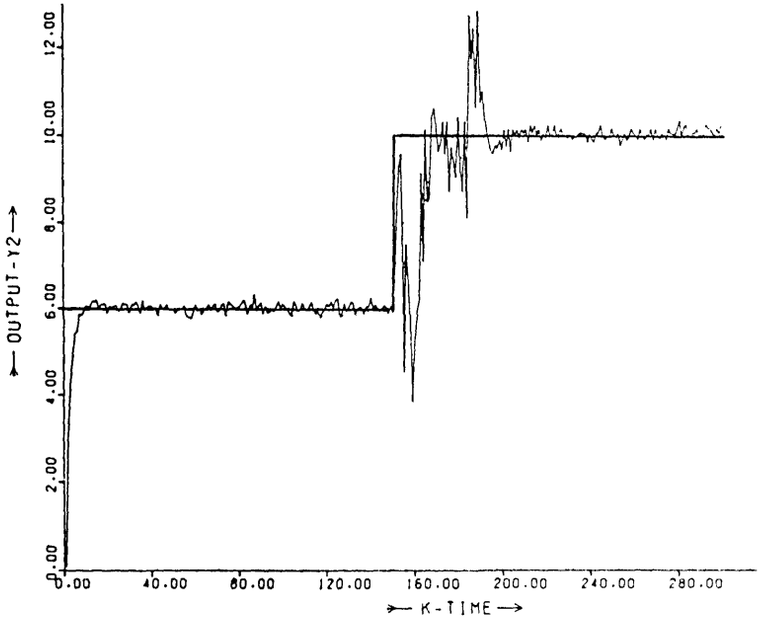
N=2,1 T=3,3 MODELS=5,6 INTERACTION



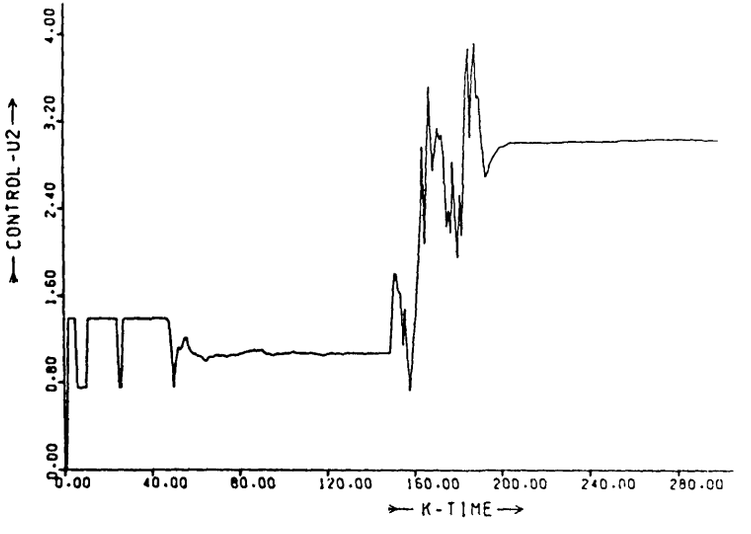
a) First output of the MIMO self-tuner



b) First input of the MIMO self-tuner



c) Second output of the MIMO self-tuner



d) Second input of the MIMO self-tuner

polynomial STR's do not consider the parameters which described the interaction between both input-output pairs. The difference in performance is quite remarkable. The system controlled with the SISO STR's becomes unstable before the set point changes, although, the set point changes help the system converge. The settling times are much longer than those of the state-space controller. The poor behaviour of the SISO self-tuners before the set-point changes may be due to some degree of dynamic interaction not accounted by the Bristol method [Appendix F] and also to the small values used for n_1 and n_2.

Initial Settings

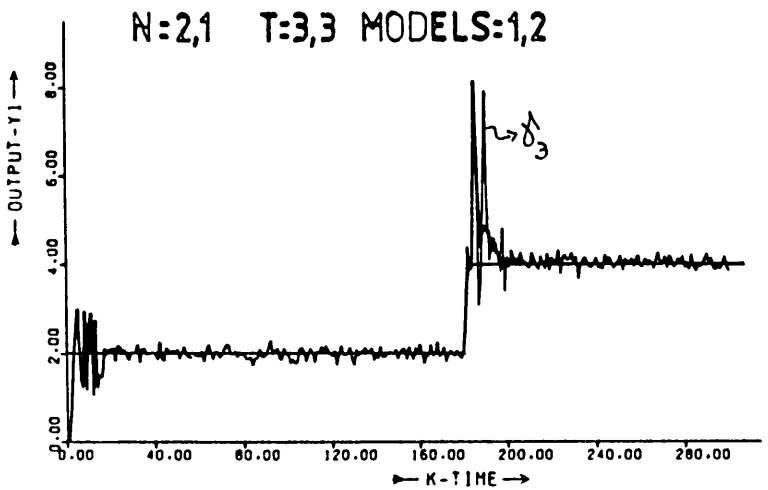
As was mentioned in section (3.2.2), some initial parameters must be given to the state-space self-tuner. In the following, various simulation results are presented to show the sensitivity of the adaptive algorithm performance to the selection of these initial parameters.

The adaptive controller tested in this section requires the estimation of the state vector to compute the control action and to estimate the model parameters. In this particular multivariable self-tuning algorithm, the system is decomposed into m subsystems, where m is the number of I/O pairs. The state estimator is a constant gain filter which can use a different gain for each subsystem, however, for simplicity the same gain is used in all of them. Alternative state estimators are mentioned in section (3.2.1).

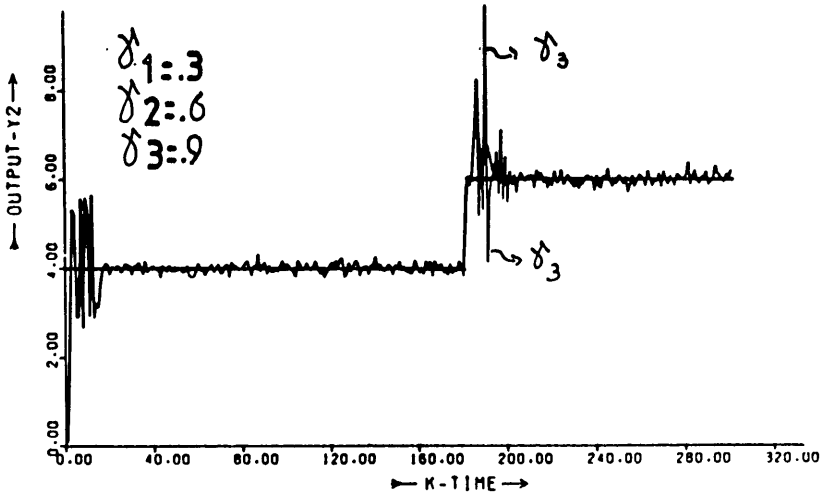
Figure (5.26) shows the responses obtained when 3 different filter gains (γ) were used in the state estimator [see section (3.2)]. Models 1 and 2 were employed in the quasi-nonlinear simulation. According to the figure, the performance of the controlled system appears very robust to the selection of the filter gain γ. The response only became degraded when a very large value [γ=.9] was used. In this extreme case, little weight is given to the state vector predicted by the adaptive model so that the filter becomes very sensitive. Many other simulations gave similar results. In general, values of γ which range between .1 and .3 are used in most of the simulations and experiments shown.

As was earlier mentioned, a standard RLS parameter estimator is unsuitable for adaptive algorithms since the parameter correction gain tends to zero. A RLS with a variable forgetting factor method is used in the adaptive controller to keep the algorithm alert. The rate of

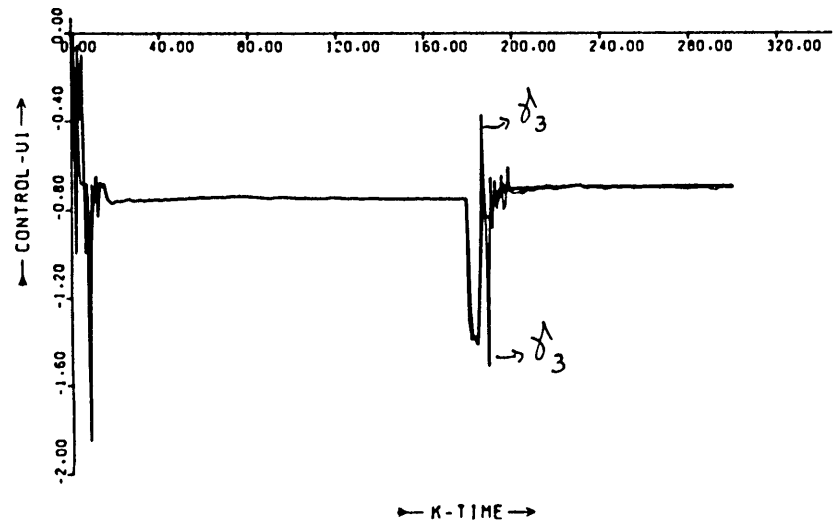
FIG 5.26: MIMO STATE-SPACE STR WITH DIFFERENT FILTER GAINS



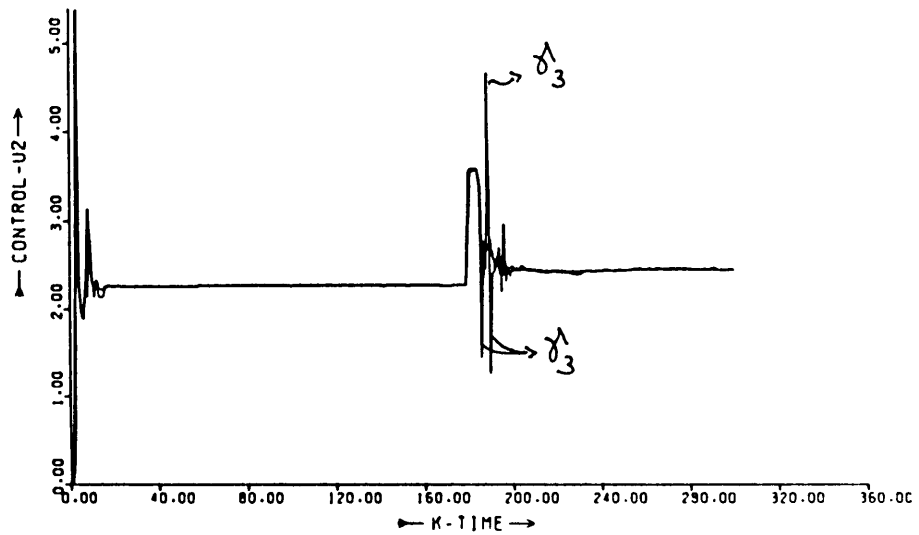
a) First output of the MIMO self-tuner



c) Second output of the MIMO self-tuner



b) First input of the MIMO self-tuner



d) Second input of the MIMO self-tuner

adaptation of the estimator can be controlled by selecting the parameter Σ [information length or content].

The performance of the state-space self-tuner for different information lengths can be seen in Fig. (5.27). If a very large information length is used (Σ_o=2), the estimator is slow to learn and needs large deviations to make the algorithm converge. On the other hand, if a very small information length is used (Σ_o=.02), the estimation algorithm becomes extremely sensitive, causing "bursting" effects. This behaviour has also been reported by Hiram and Kershenbaum (1983) in their study with an EH/SISO robust self-tuning regulator. In the simulations shown in Fig. (5.27), where there is no model change, neither of the responses present any problems at the set point changes.

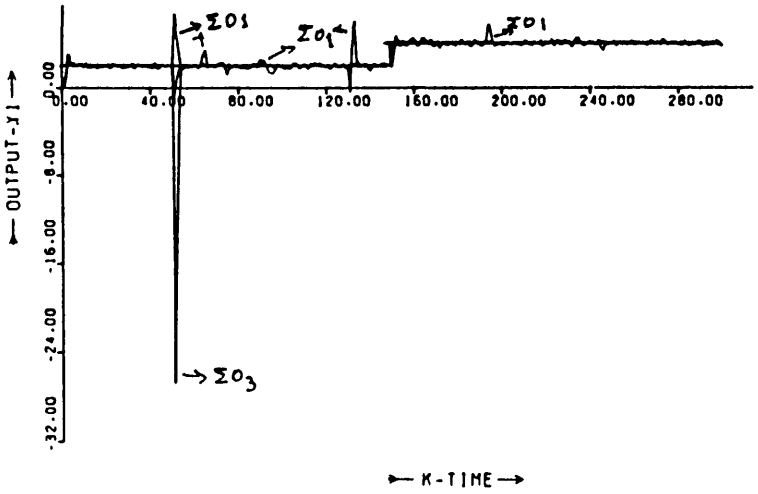
The structural indices [observability indices] are very important initial parameters. In the EH SISO algorithm, the order of the assumed model completely defines the model structure. However, in the MIMO case, the structure of the assumed model cannot be completely specified by just the order of the plant. The whole set of structural indices must be provided, one for each input-output pair in the MIMO system [see eq. (3.43)]. For example, in the case of two-input two-output plants, the MIMO self-tuner needs two structural indices.

An example of the sensitivity to the selection of these parameters is given in Figs. (5.28) and (5.29). In Fig. (5.28) the behaviour of the control system looks practically the same when three different identification model structures were used [N_1=1,1; N_2=1,2; N_3=2,1]. This is similar to controlling a SISO system assuming different model orders. In the simulations shown in Fig. (5.28), the time-horizon was set to 3 in both outputs.

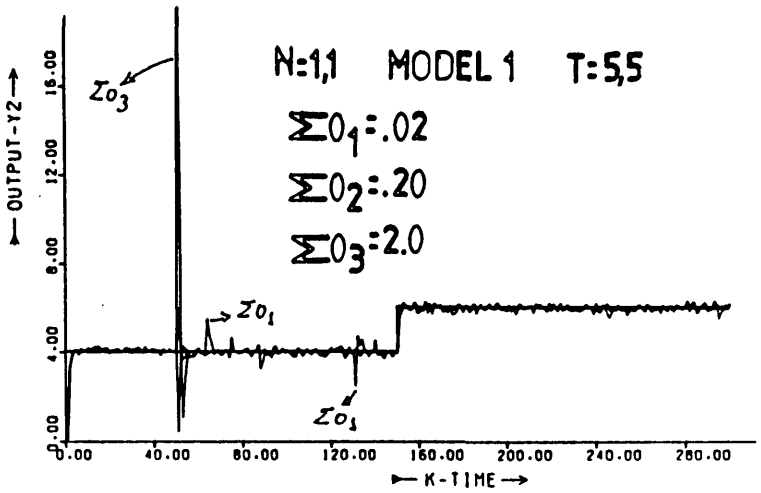
A more important difference in performance can be appreciated in the quasi-nonlinear simulation shown by Fig. (5.29), where the time-horizon was 8 in both outputs. The large horizon produced excessive overshoots, especially in the first output, when wrong model structures were used. In these simulations, the true model structure was N=2,1; namely the first observability index is 2 and the second is 1.

There is no clear guide for the selection of the identification model structure without a priori knowledge about the MIMO plant, however, this is also true for SISO plants. Of course, in the MIMO case this selection is more difficult. However, it appears that no matter which model structure is used for the identification, the state-space algorithm finally converges to the desired set points.

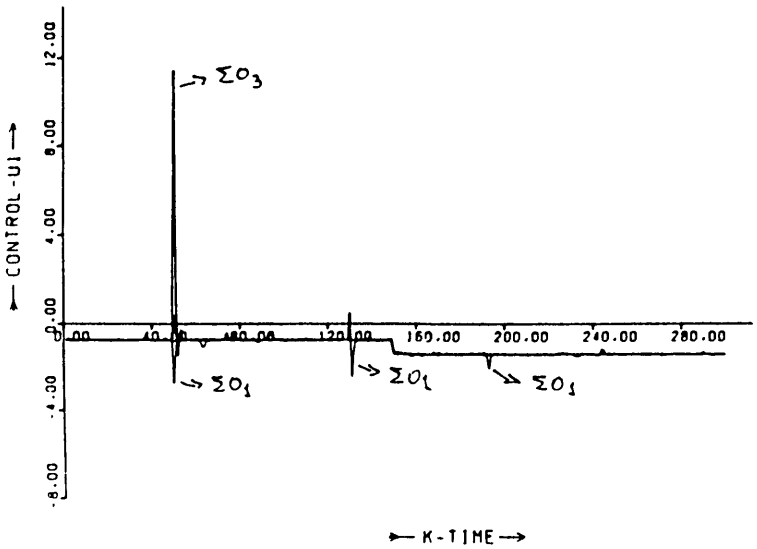
FIG 5.27: MIMO STATE-SPACE STR WITH DIFFERENT INFORMATION LENGTHS



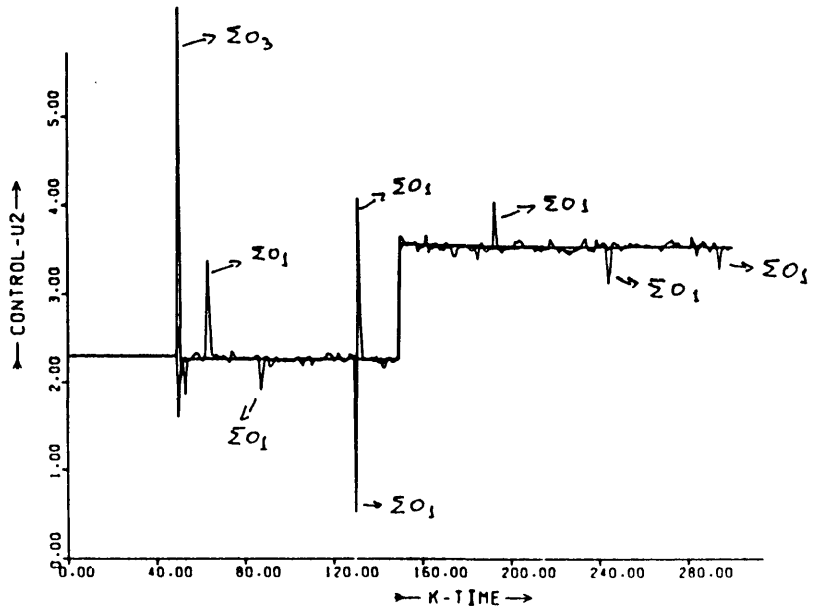
a) First output of the MIMO self-tuner



c) Second output of the MIMO self-tuner

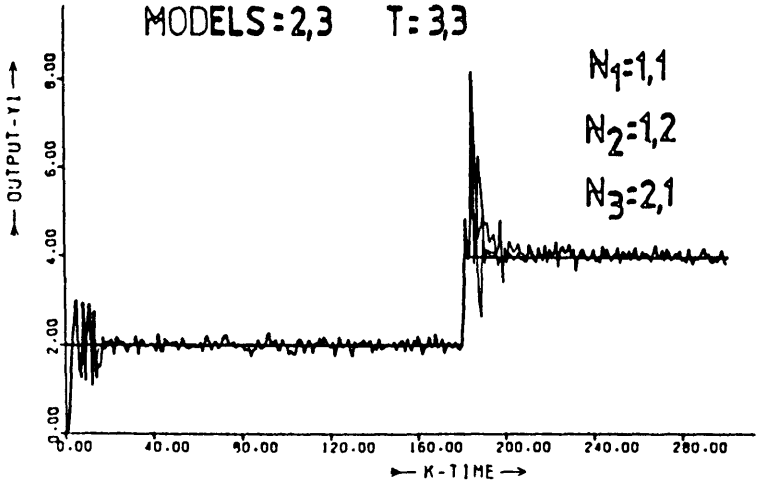


b) First input of the MIMO self-tuner

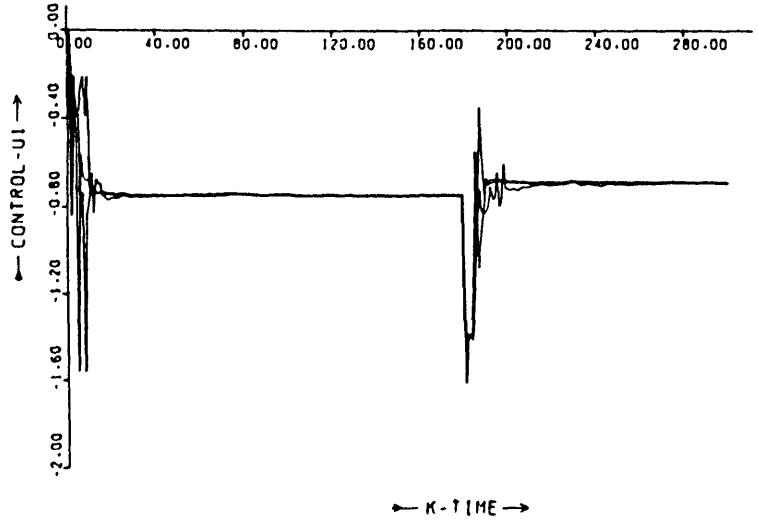


d) Second input of the MIMO self-tuner

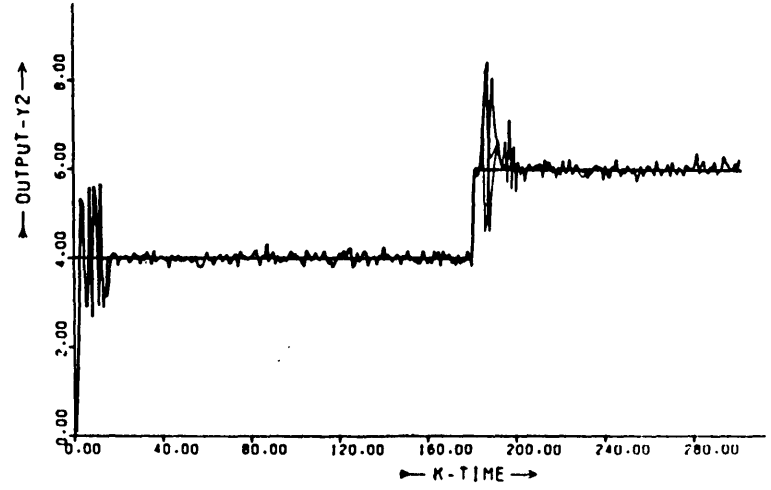
FIG 5.28: MIMO STATE-SPACE STR WITH DIFFERENT STRUCTURES



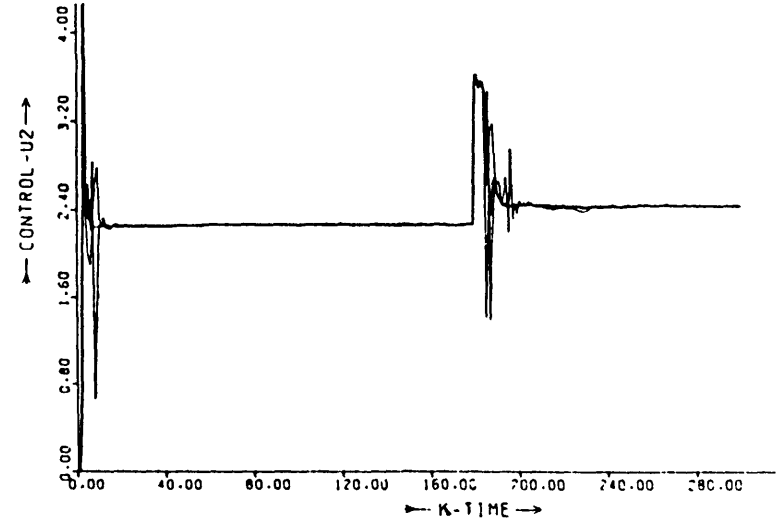
a) First output of the MIMO self-tuner



b) First input of the MIMO self-tuner

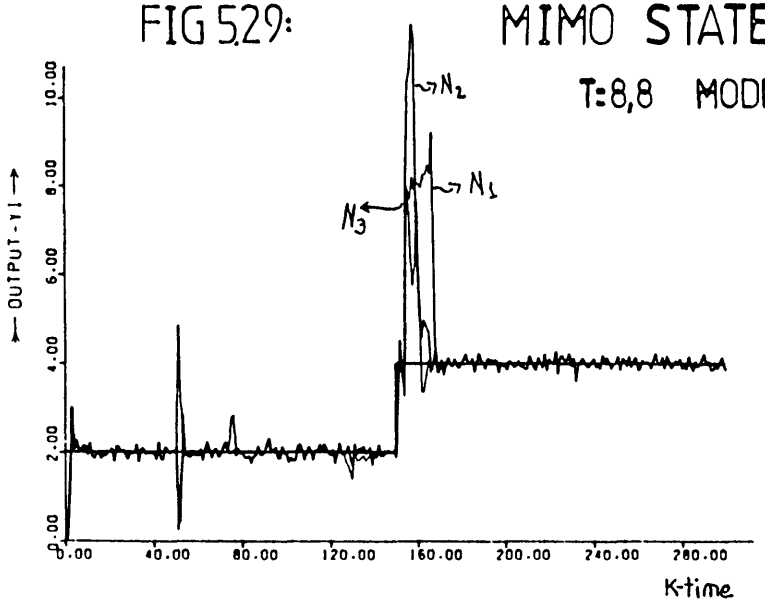


c) Second output of the MIMO self-tuner

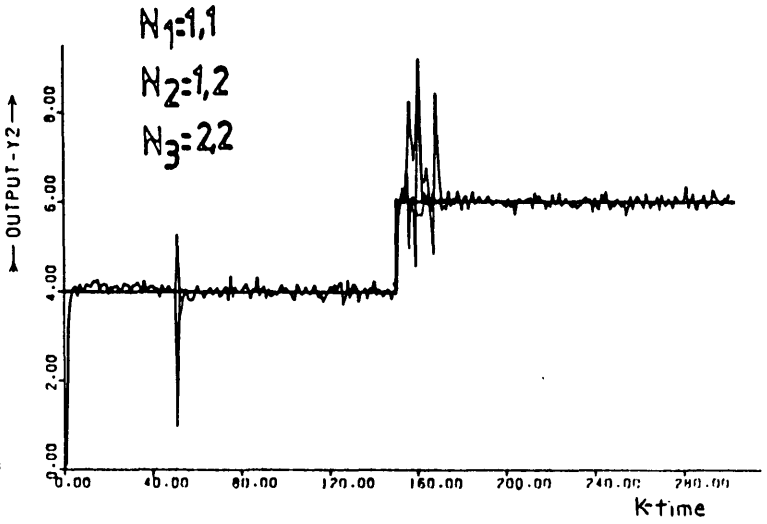


d) Second input of the MIMO self-tuner

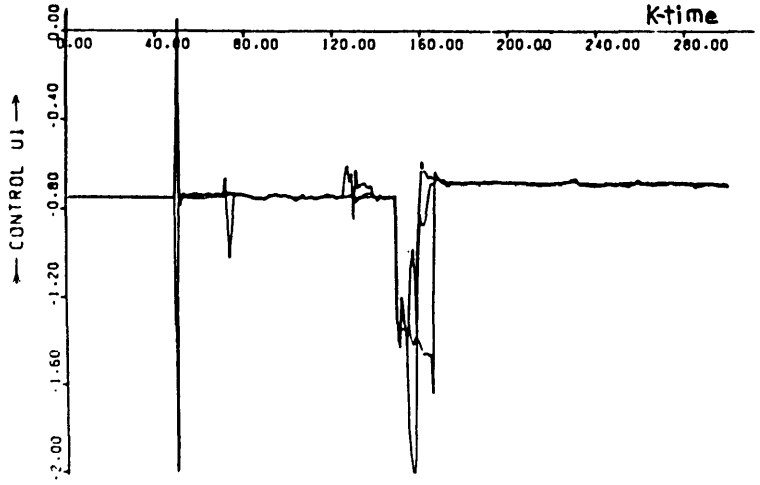
FIG 5.29: MIMO STATE-SPACE STR WITH DIFFERENT STRUCTURES



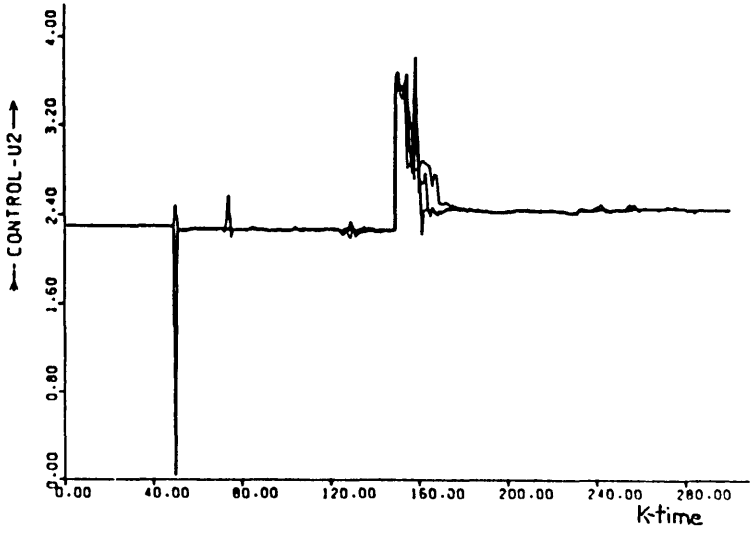
a) First output of the MIMO self-tuner



c) Second output of the MIMO self-tuner



b) First input of the MIMO self-tuner



d) Second input of the MIMO self-tuner

The control-horizons are also very important tuning parameters; they define the response of the controlled system. Figures (5.30) to (5.33) show the sensitivity of the state-space controller to the selection of these parameters. For example, in Fig. (5.30) there is no model change and both outputs use the same horizon. It can be seen that the difference in performance is insignificant despite the different horizons used. The simulation of Fig. (5.31), where a change in the model occurred, shows an appreciable difference in performance when a very large horizon [T=8] is used. However, with a small or moderate horizon, the self-tuner performed in the same way. If a different horizon is used for each output, the self-tuner is more sensitive to the selection of these parameters. In Fig. (5.32) the response of the state-space controller is compared for different combinations of time-horizons. The system controlled with horizon 5 for the first output and horizon 1 for the second output presents a much better response. A similar result is given in Fig. (5.33), where different models and time-horizon combinations were used. In this simulation the system controlled with horizon 8 for the first output and horizon 3 for the second one, presents large overshoots.

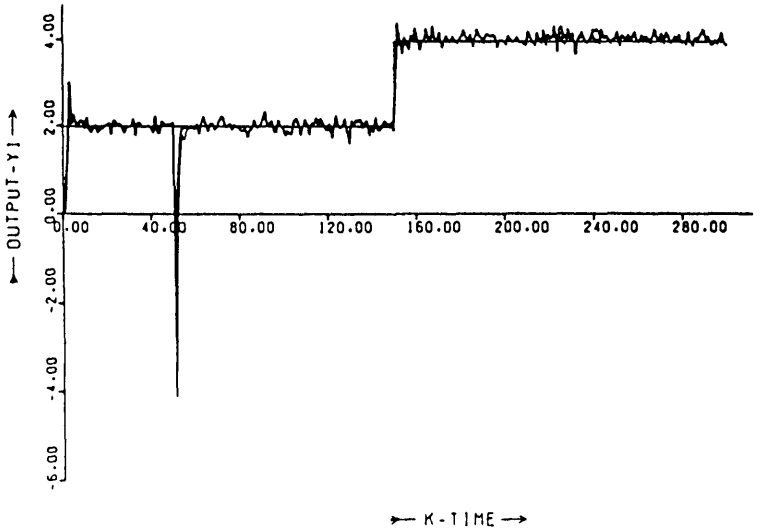
It is not clear why the selection of time-horizons is more difficult when each output uses a different one. Nevertheless, it is important to point out that finally the algorithm converged to the desired set points independently of the combination of time-horizons used. Moreover, the difference in performance only appears when the set point changes are accompanied with model parameter changes [typical of non-linear systems].

In systems without delay it is advisable to use the same horizon for both outputs. Control horizons of T=1 or T=2 produce a good regulation, however, if the control action is very demanding it can be detuned by using larger horizons [T=3, T=4 or T=5]. It is not advisable to use very large horizons [T=8 or more], unless time delays are present.

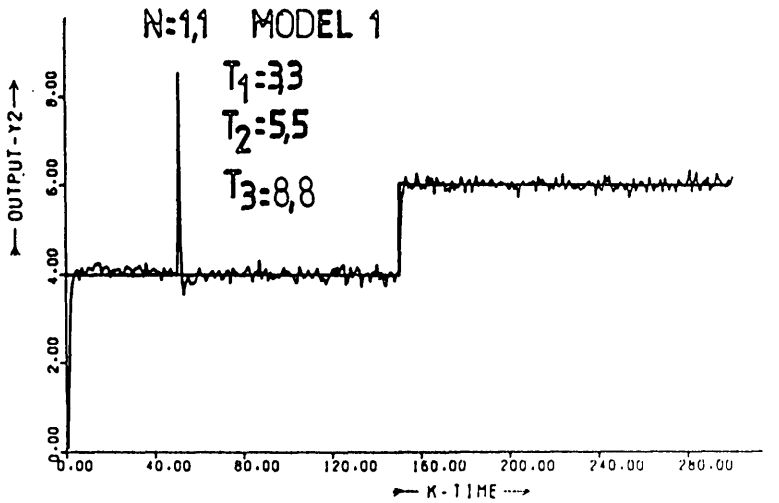
MIMO control with Time-delays

The results above indicate that it would be better to use similar horizons for each output. However, this is not the case with systems characterized by a general delay structure [see section (2.3)].Figures (5.34) and (5.35) give the response of systems with different delays in each input-output pair. In the simulation shown in

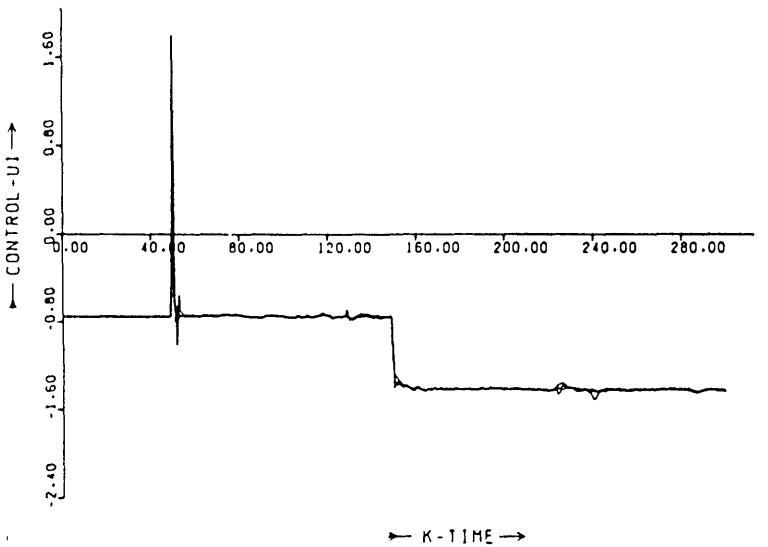
FIG 5.30: MIMO STATE-SPACE STR WITH DIFFERENT HORIZONS



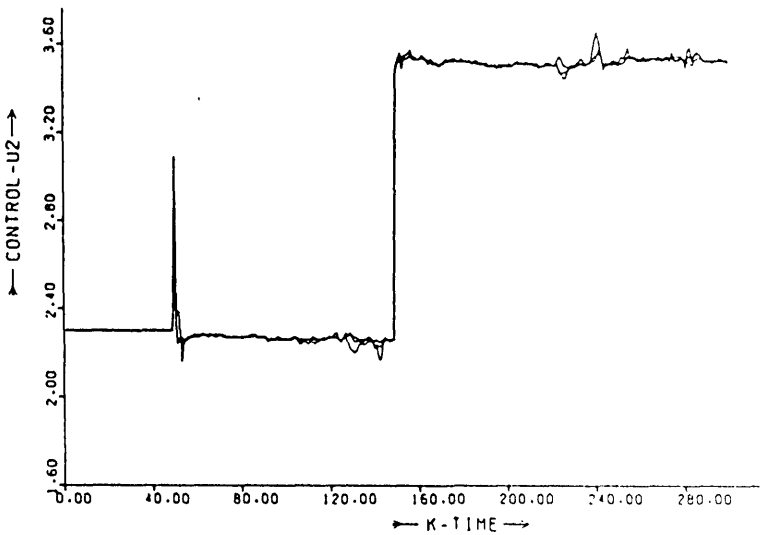
a) First output of the MIMO self-tuner



c) Second output of the MIMO self-tuner.

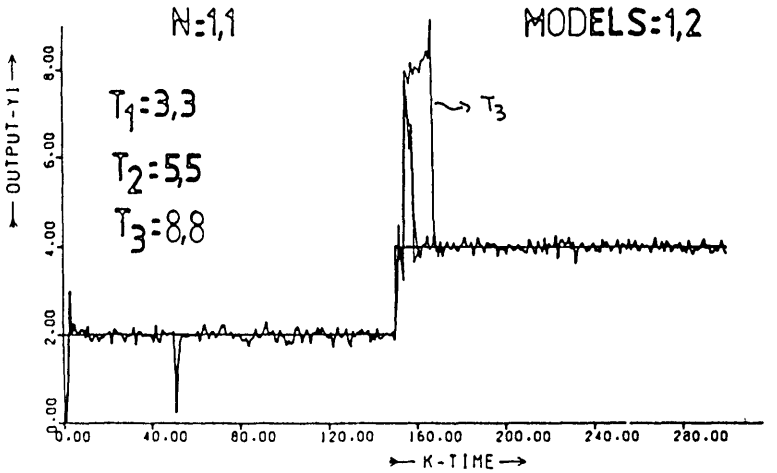


b) First input of the MIMO self-tuner

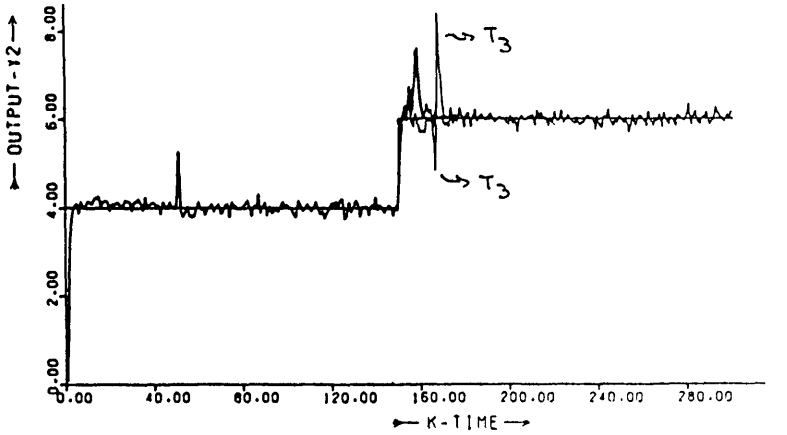


d) Second input of the MIMO self-tuner

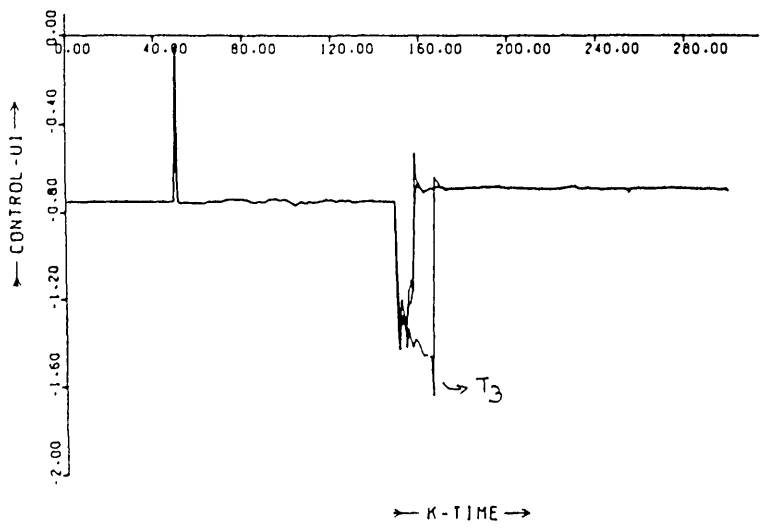
FIG 5.31: MIMO STATE-SPACE STR WITH DIFFERENT HORIZONS



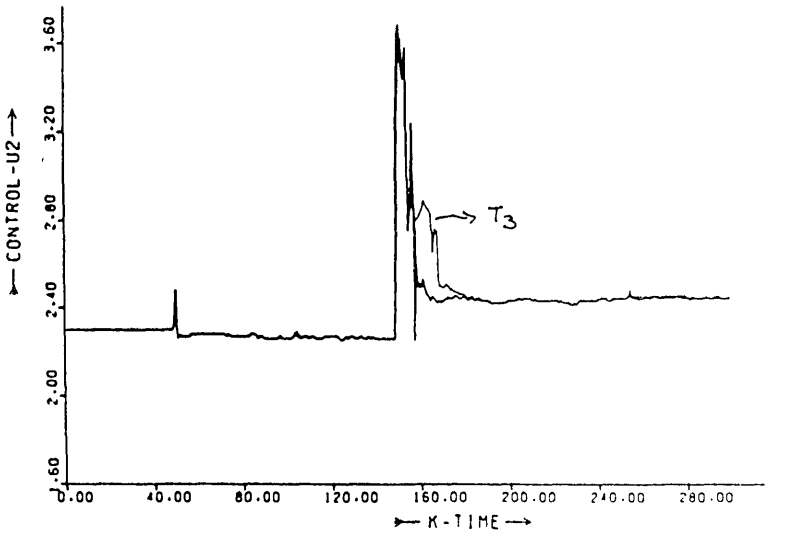
a) First output of the MIMO self-tuner



c) Second output of the MIMO self-tuner

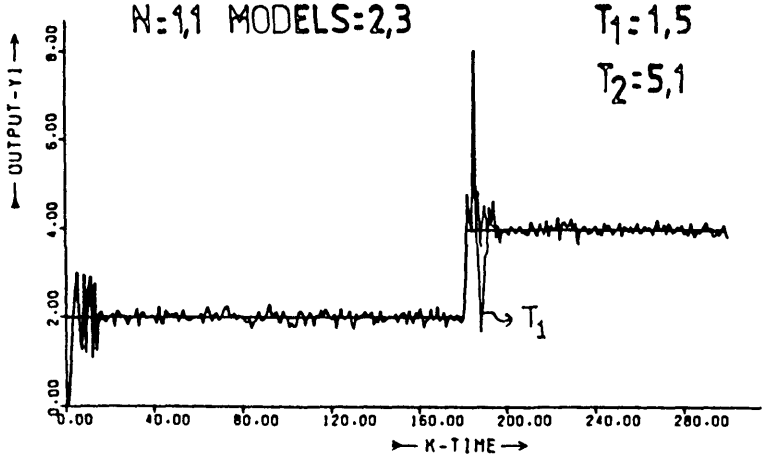


b) First input of the MIMO self-tuner

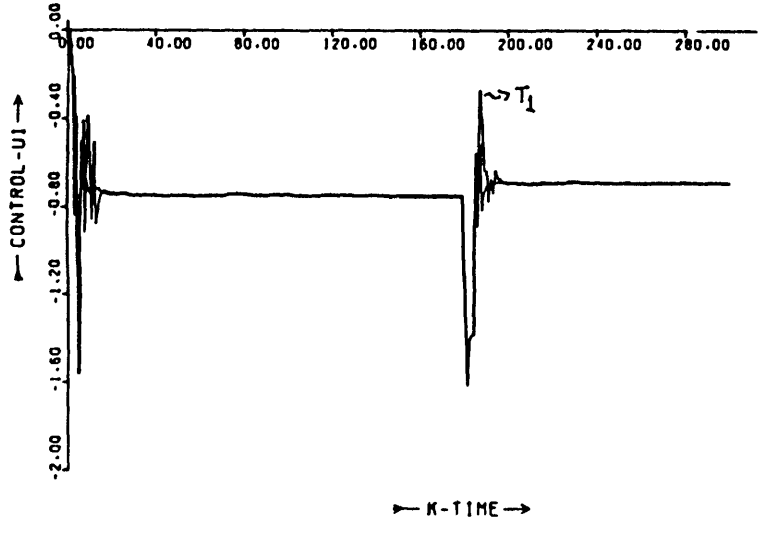


d) Second input of the MIMO self-tuner

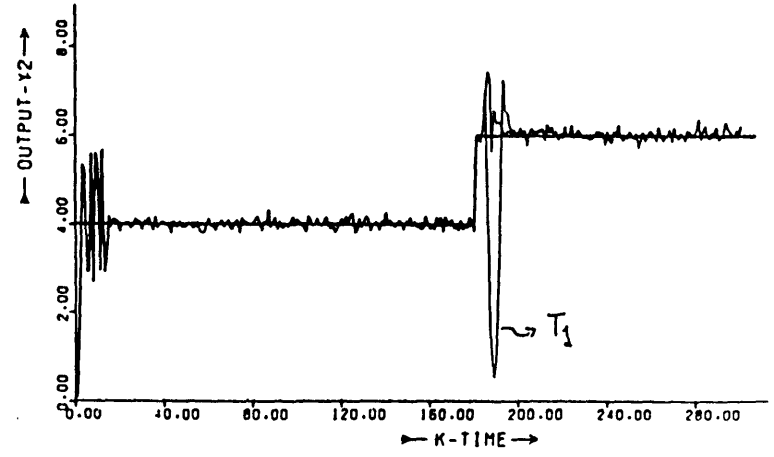
FIG 5.32: MIMO STATE-SPACE STR WITH DIFFERENT HORIZONS



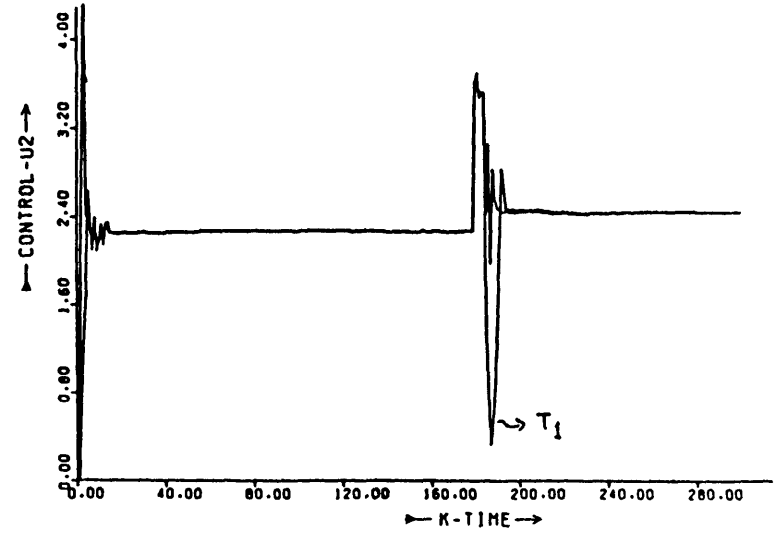
a) First output of the MIMO self-tuner



b) First input of the MIMO self-tuner



c) Second output of the MIMO self-tuner



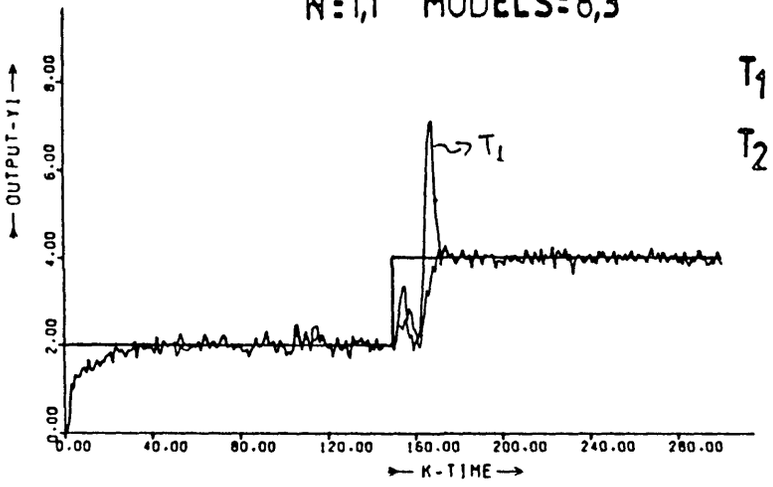
d) Second input of the MIMO self-tuner

FIG 5.33: MIMO STATE-SPACE STR WITH DIFFERENT HORIZONS

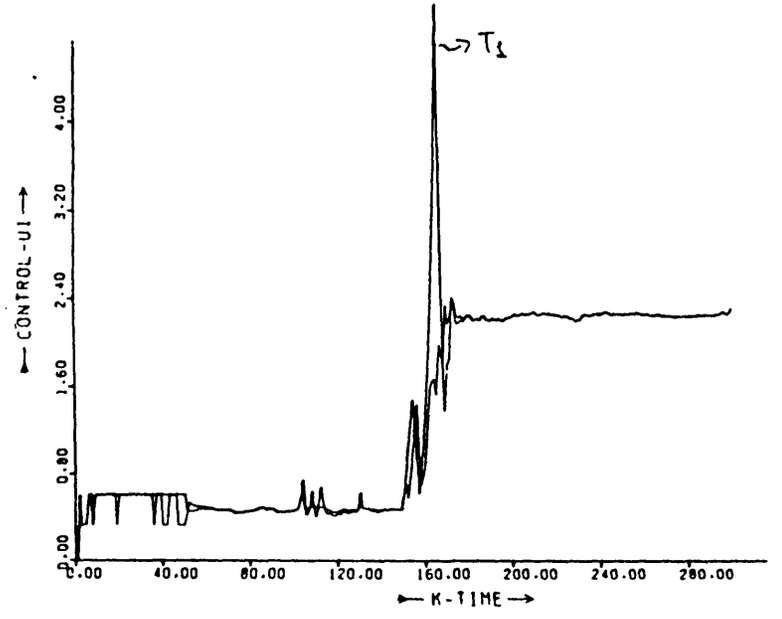
N:1,1 MODELS:8,3

$T_1 = 8,3$

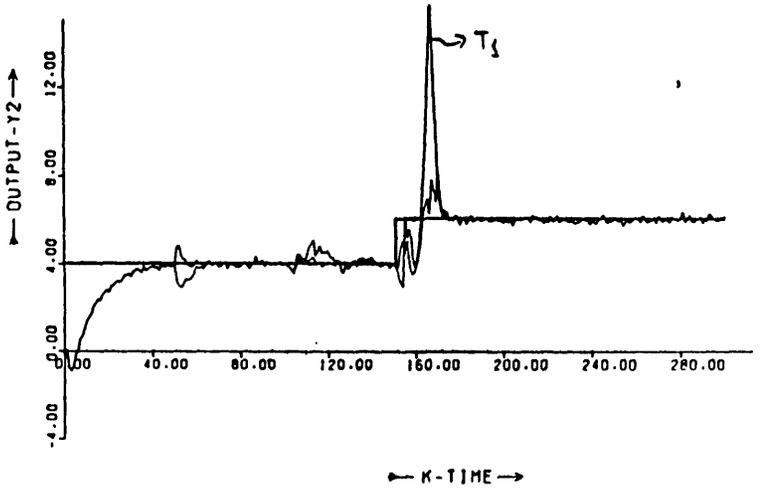
$T_2 = 3,8$



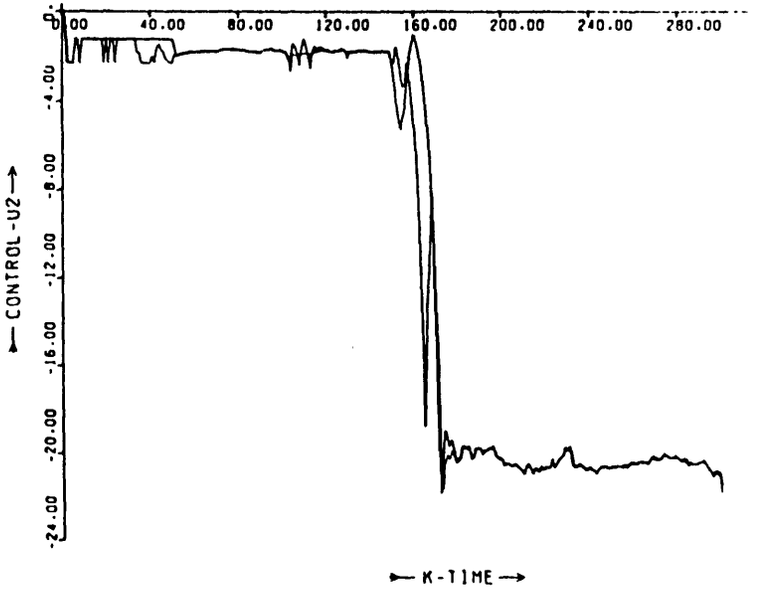
a) First output of the MIMO self-tuner



b) First input of the MIMO self-tuner



c) Second output of the MIMO self-tuner



d) Second input of the MIMO self-tuner

Fig. (5.34) the simulated system [model 11 in Appendix H] is affected by a diagonal delay structure [see Chapter 2]. Here, the second input affects both outputs with the same delay [d=2]. Similarly, the first input affects both outputs with a delay d=1. The delay structure is then defined by D={1,2,1,2). In this simulation the control horizon was chosen to be 5 for both outputs. The response of the second output, shown in Fig. (5.34), appears unnecessarily retarded. Nevertheless, this output finally reached the desired set point.

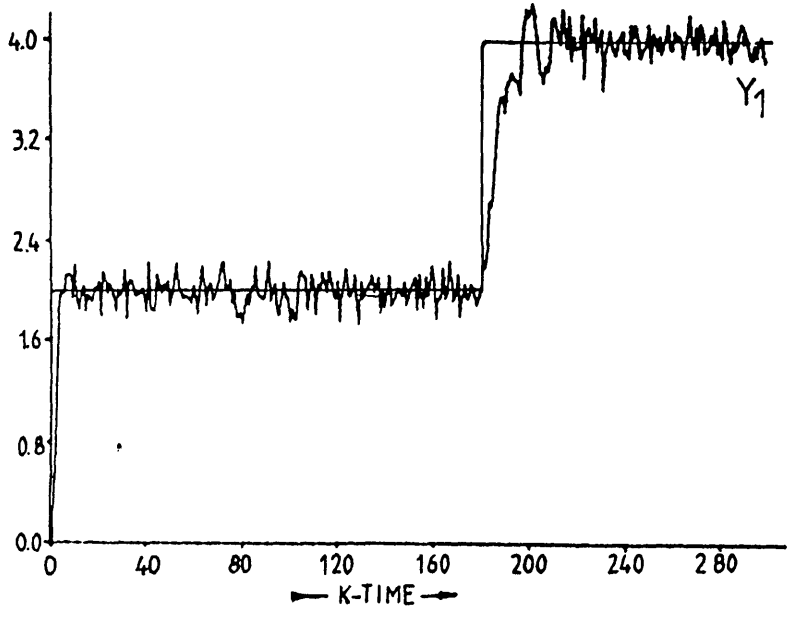
Figure (5.35) gives the response of a system with general delay structure. Here, the first input affects the first output with a delay of d=2 and the second output with a delay d=3. The second input affects the first output with a delay of d=4 and the second output with a delay of d=5. The delay structure is then defined by D={2,3,4,5). A different time horizon was used for each output. It can be seen in the figure that the adaptive algorithm has driven both outputs to the desired set point in spite of the large and difficult delay structure.

State Estimation

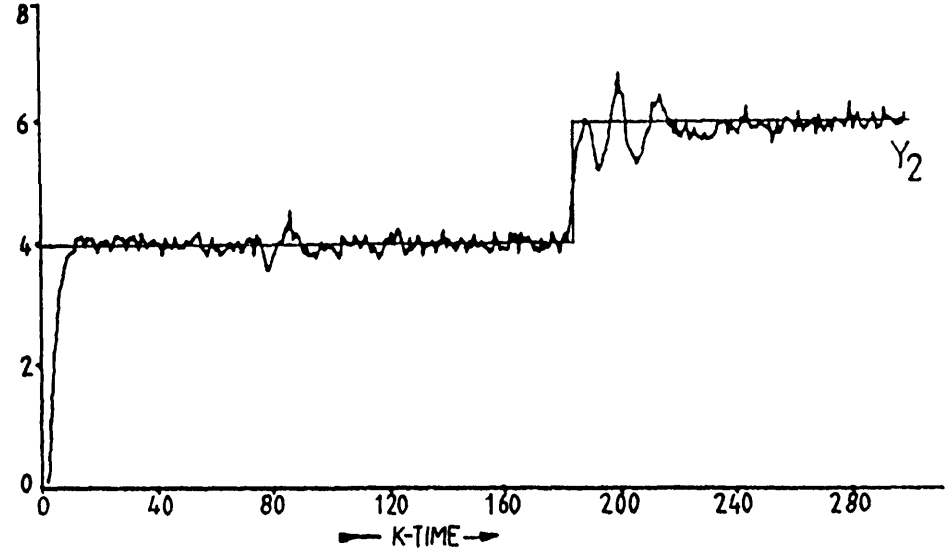
To conclude this series of simulations, the behaviour of the state estimates will be studied. Figure (5.36) shows the response of the controlled system in a particularly difficult problem. It can be seen in Figs. (5.36.a) and (5.36.c) that the controller converged to the set points despite the change of sign of the control inputs [Figs. (5.36.b) and (5.36.d)] denoting a change in the process gains sign. The evolution of the states and states estimates is also shown [Figs. (5.36.e) to (5.36.g)]. The estimate of the third state is considerably better than the other estimates. The estimates z_1 and z_2 oscillate around the true state. Despite the changes in set points and model parameters, the state estimates do not diverge and the outputs are driven to the desired reference. This is because a constant gain filter is used to continuously update these estimates. Had a stochastic approximation filter been used, the state estimates would have diverged. In this case, there is no way to update the states with the plant outputs once the algorithm has converged [see section (3.2)]. Alternative filter methods which also avoid state divergence are mentioned in Chapter 3, but these are more complicated than using a constant gain filter.

FIG 5.34: MIMO STATE-SPACE STR WITH TIME-DELAYS

N=1,2 T=5,5 D=1,2,1,2 MODEL:11

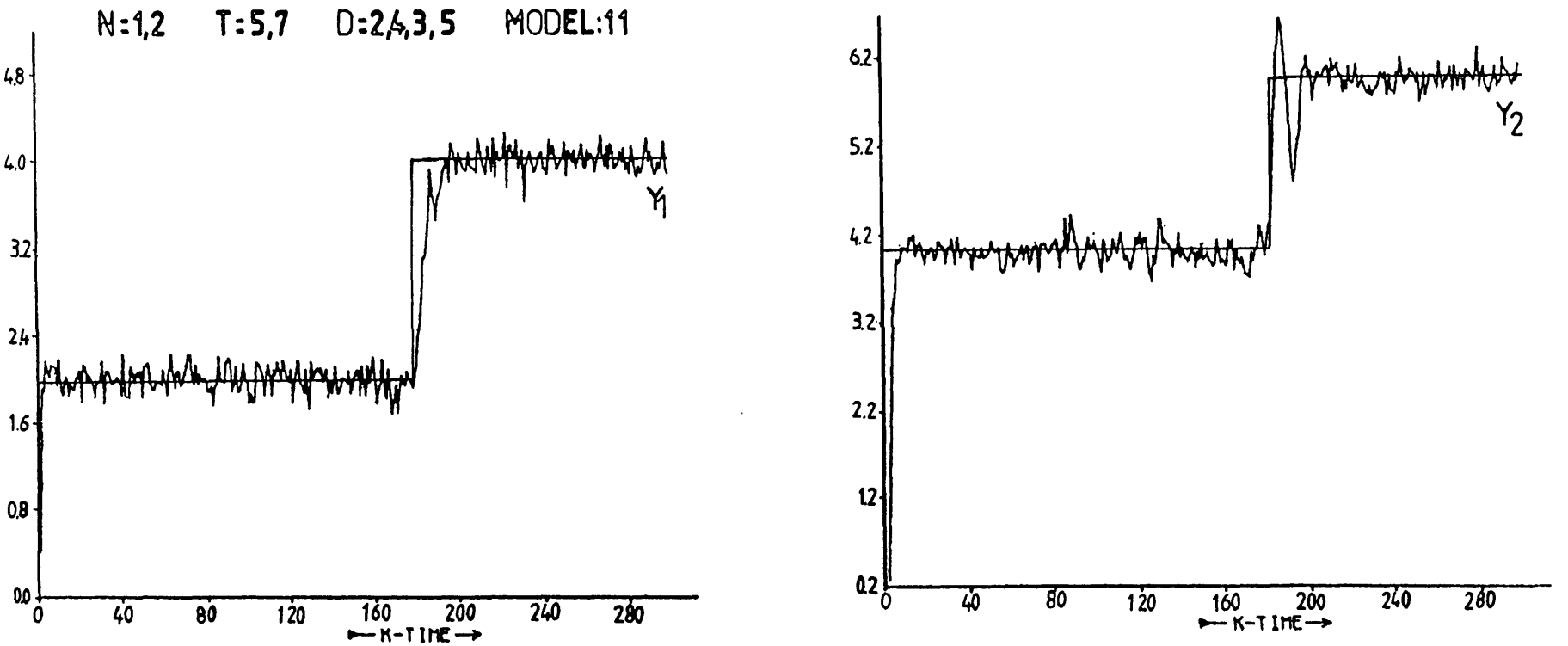


a) First output of the MIMO system



b) Second output of the MIMO system

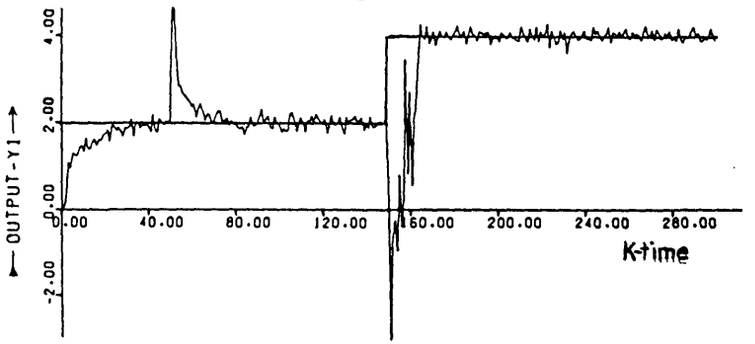
FIG 5.35: MIMO STATE-SPACE STR WITH TIME-DELAYS



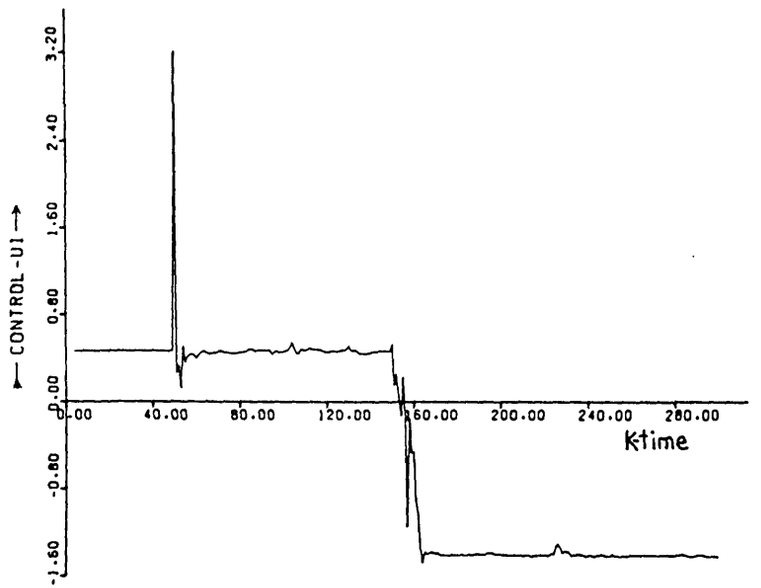
a) First output of the MIMO system

b) Second output of the MIMO system

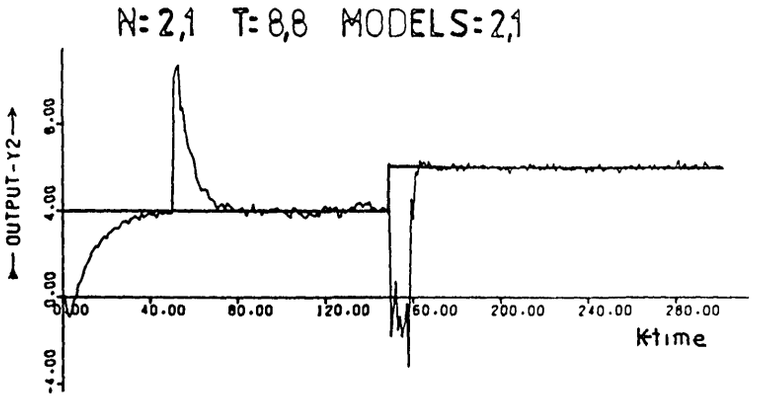
FIG 5.36: MIMO STATE-SPACE STR



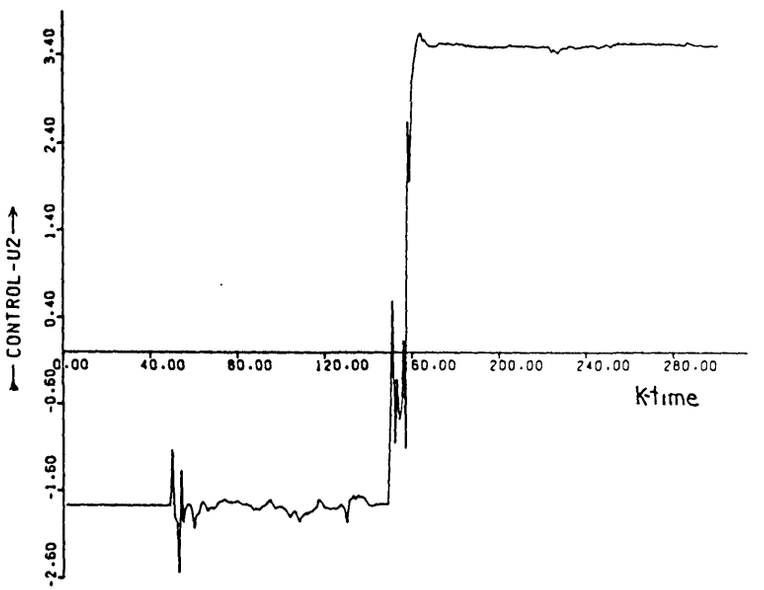
a) First output of the MIMO self-tuner



b) First input of the MIMO self-tuner



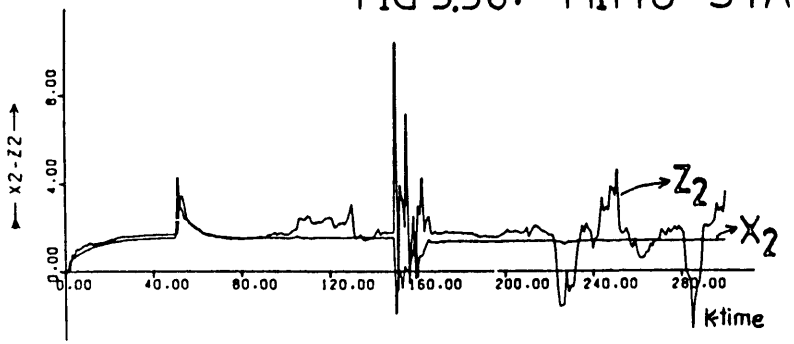
c) Second output of the MIMO self-tuner



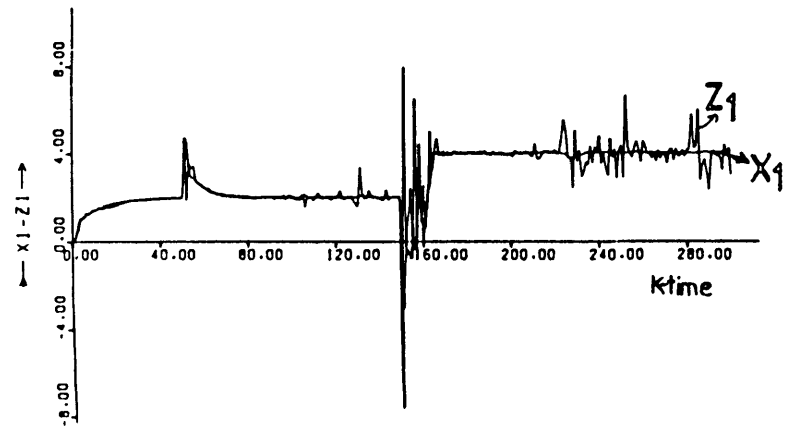
d) Second input of the MIMO self-tuner

FIG 5.36: MIMO STATE SPACE-STR

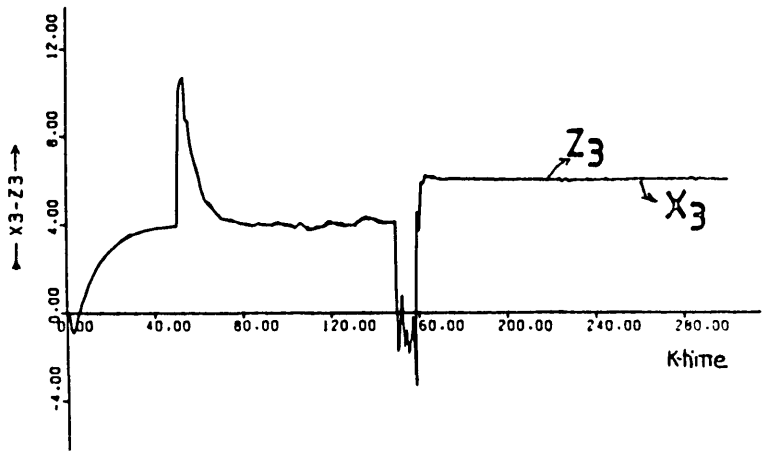
N=2,1 T=8,8 MODELS : 2,1



e) Second model state (x_2) and state estimate (z_2)



f) First model state (x_1) and state estimate (z_1)



9) Third model state (x_3) and state estimate (z_3)

5.2.2.- An Incremental Feedback State-space Self-tuner

In spite of the good results obtained in simulations with the previous state-space self-tuner, the performance of this algorithm in pilot-plant experiments turned out to be very poor. As is shown below by simulations and experiments, this bad performance is related to the positional nature of the self-tuner used above.

An incremental version of the positional self-tuner shown in section (3.2) was implemented, tested and compared in simulations and experiments. The incremental algorithm retains the same structure of the positional version and is described in detail in Appendix I.

The cooling-loops in the absorption/desorption pilot plant of Fig. (4.1) were used in the experiments and first order linear models were used in the simulations. These linear models simulate the cooling-system.

Simulations

It is assumed that the dynamic of the cooling-loops can be described by first order linear models with a bias parameter. This bias depends on the pressure in the pipe lines and also on the position of the manual valves.

An appropriate model can be:

$$y_k = a \cdot y_{k-1} + b \cdot u_{k-1} + \delta_k \qquad (5.10)$$

where y is the flow in kg/s, u is the normalized signal to the valve (from 0 to 1), and δ is a bias parameter.

For a given set of operating conditions, the parameters of the model eq. (5.10) were found [by identification methods] to be:

	a	b	δ
system 1 Heat-exchanger	.607	.787	-.512
system 2 Condenser	.607	.983	-.256

Table 5.1 Model Parameters

Fig. (5.37) shows the results when two linear models of the form (5.10) were simulated with the given values of the a, b and δ parameters. In this manner, a non-interactive two-input two-output system is simulated. This system was controlled with both the positional and the incremental multivariable algorithms.

The simulations are for 600 time intervals. A series of set point changes is first applied to system 1, then, system 2 is subjected to set point changes.

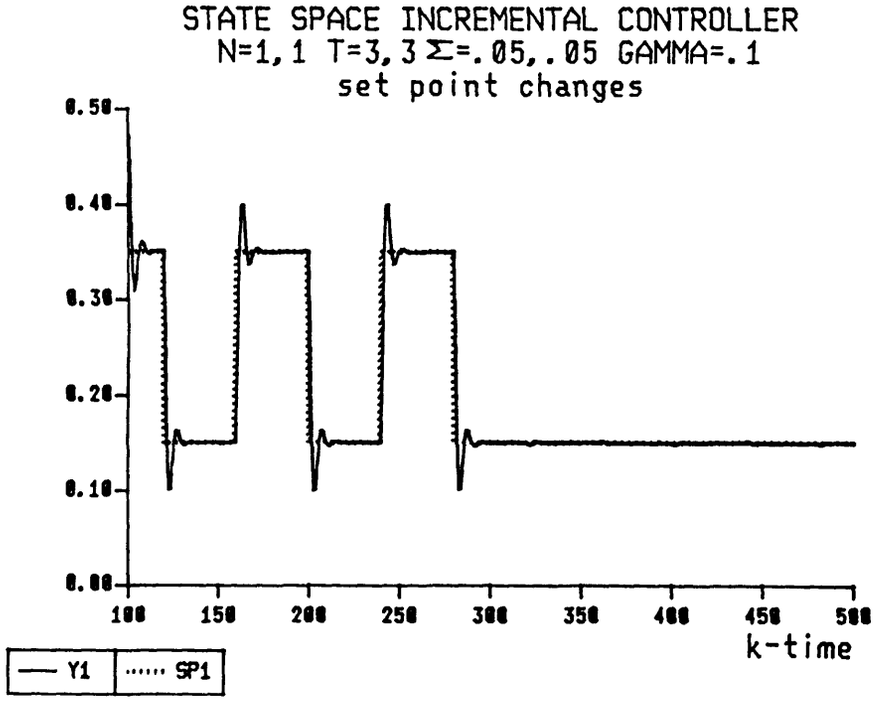
In Figs. (5.37.a), (5.37.c) and (5.37.e) it is shown how the incremental algorithm can cope with both disturbances without any interaction between the control loops, although the set point changes present some overshoots. On the other hand, the positional algorithm [Figs. (5.37.b), (5.37.d) and (5.37.f)] in the first series of set point changes tend to eliminate the overshoot and also the effects of the control-interaction caused by wrong estimates. However, the controlled-system collapses when a series of set point changes is applied to the second loop.

This phenomenon appears only when a bias parameter is used in the simulated model. In fact, because the positional model eq. (3.45.a) does not consider any bias, the parameter estimation gives wrong estimates affecting the control and the stability of the system. The "plant" used in the simulations of the first part of this section does not, in fact, have any bias, so that the described phenomenon was not observed in these simulations. However, as was discussed in section (5.1.2), real systems almost always possess a bias mainly due to non-linearities and unmeasured disturbances. In the particular case of the cooling-system, the bias can be caused by changing the pressure in the pipe lines and by changing the manual valves positions.

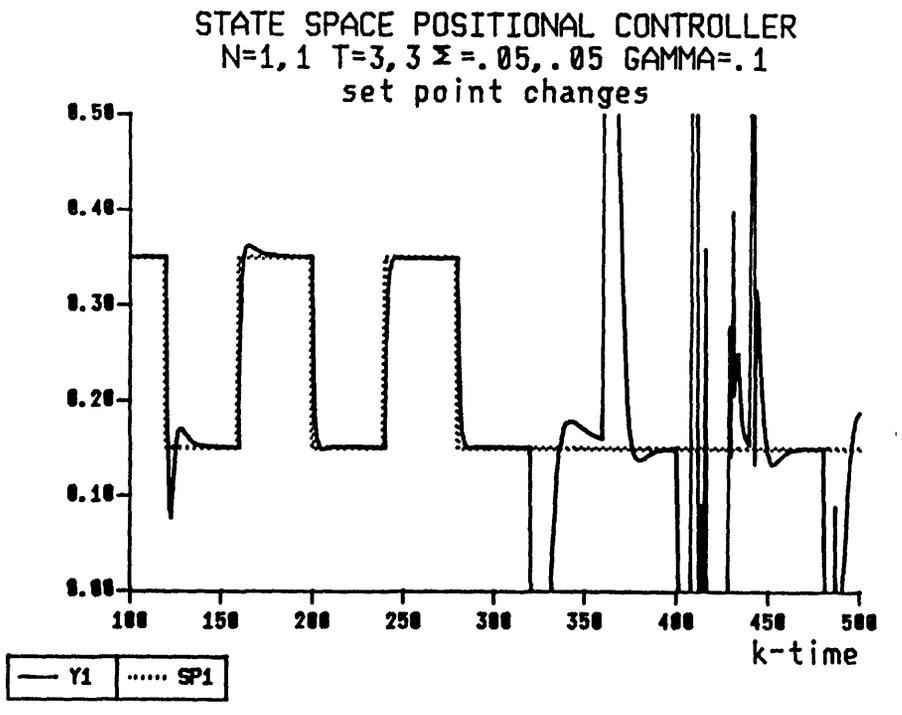
If an incremental model is used [see Appendix I] for parameter estimation, the above phenomenon is avoided and a stable control can be achieved, as is shown in Figs. (5.37.a), (5.37.c) and (5.37.e).

The variations of the control gains are given in Figs. (5.37.g) and (5.37.h). The control gains are defined by the matrix D^{-1} in eq. (3.54) of section (3.2):

$$D^{-1} = \begin{vmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{vmatrix} \qquad (5.11)$$

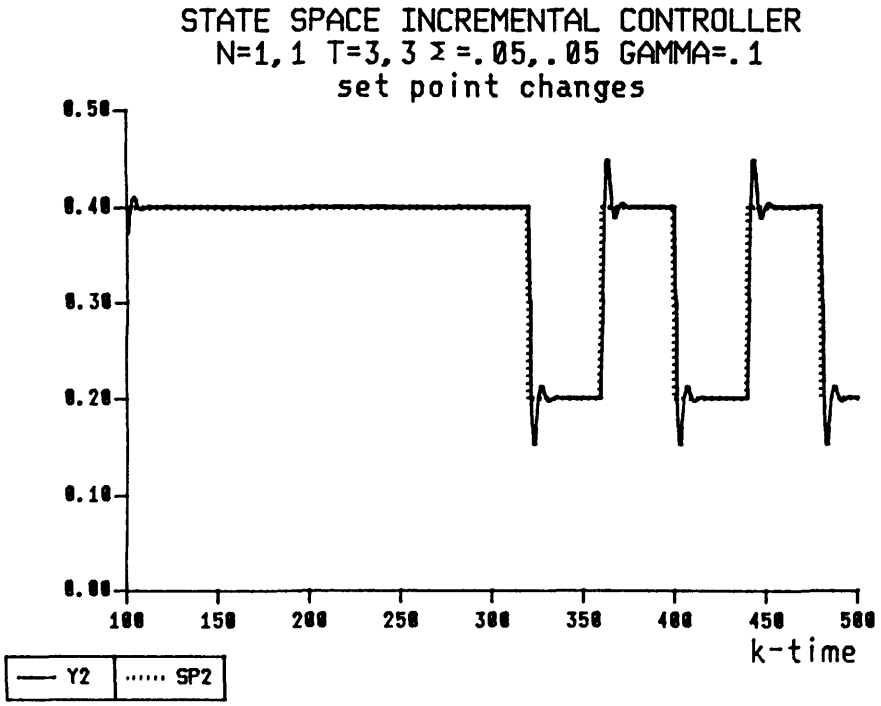


a) Output (Y1) and set point (SP1) of the simulated Heat-exchanger Cooling-loop

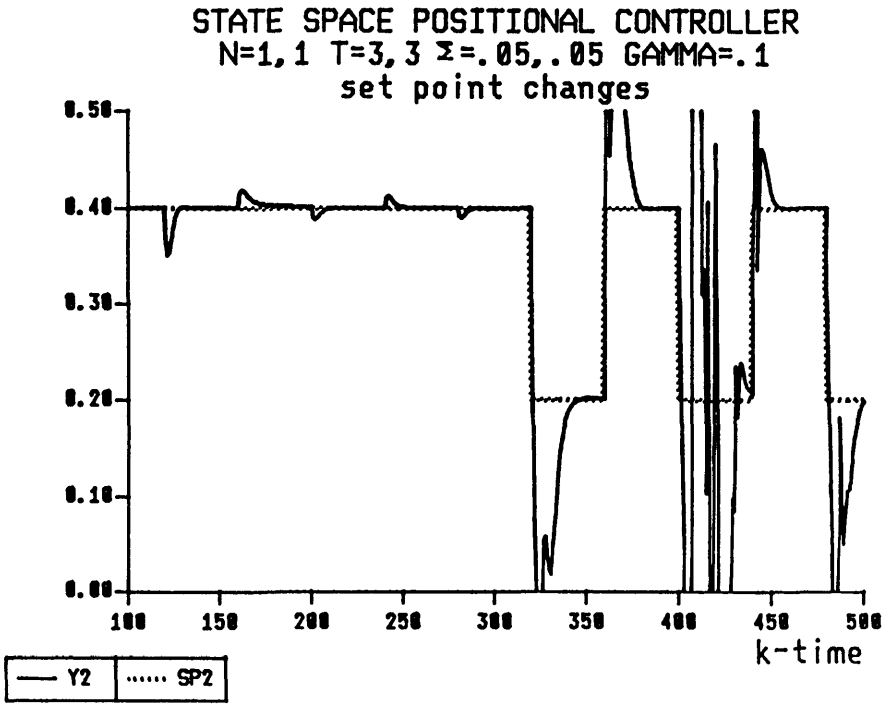


b) Output (Y1) and set point (SP1) of the simulated Heat-exchanger Cooling-loop

FIG 5.37

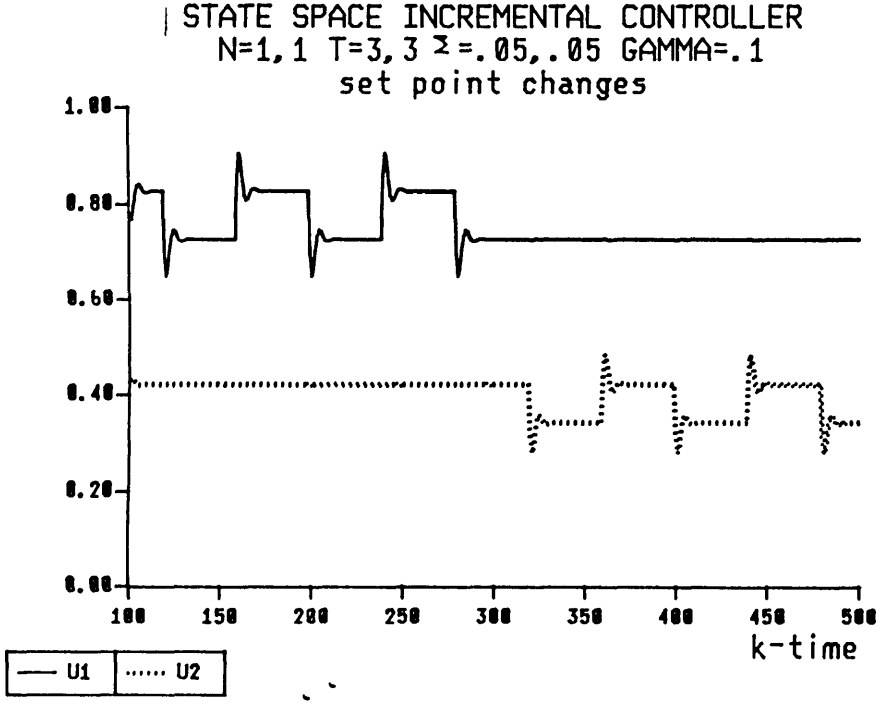


c) Output (Y2) and set point (SP2) of the simulated Condenser Cooling-loop

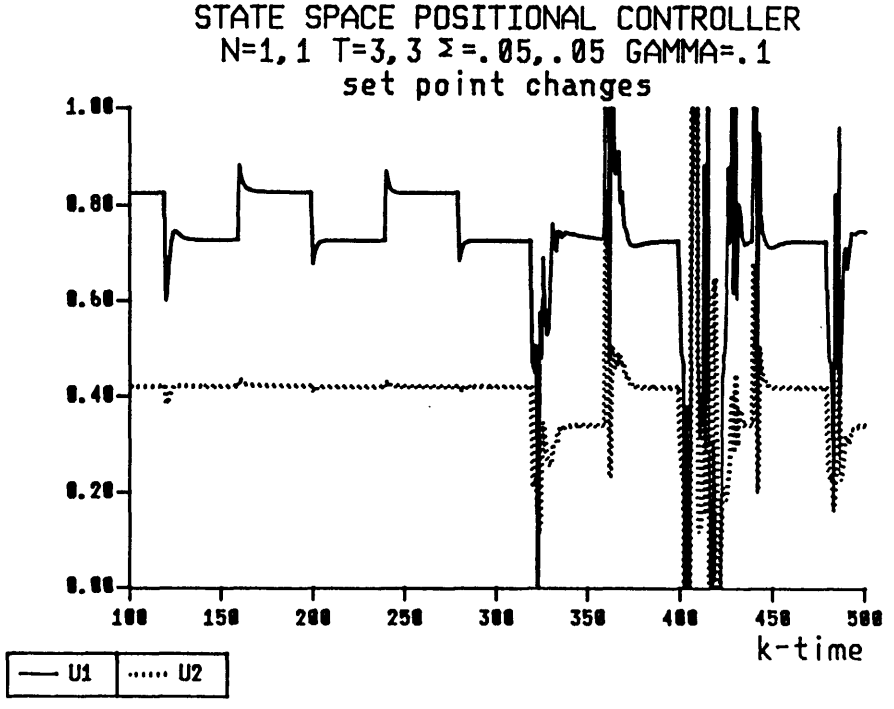


d) Output (Y2) and set point (SP2) of the simulated Condenser Cooling-loop

FIG 5.37

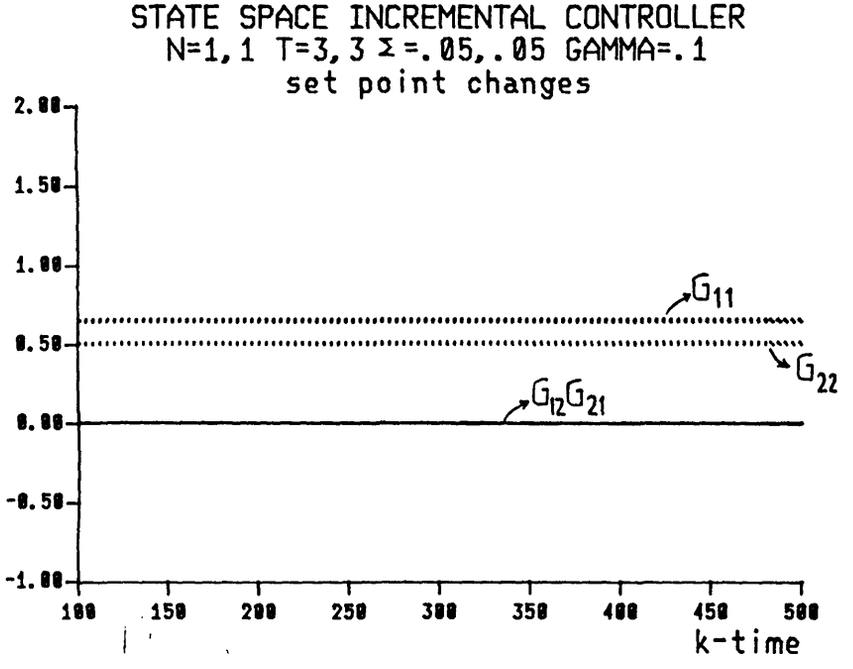


e) Input (U1) of the simulated Heat-exchanger Cooling-loop and input (U2) of the simulated Condenser Cooling-loop

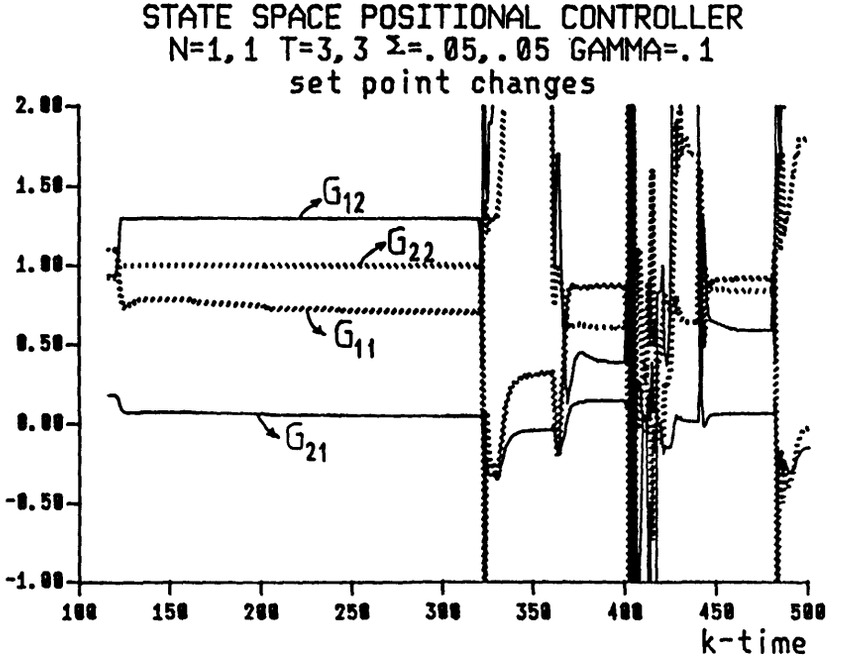


f) Input (U1) of the simulated Heat-exchanger Cooling-loop and input (U2) of the simulated Condenser Cooling-loop

FIG 5.37



g) Controller gains of the simulated Cooling-loop system



h) Controller gains of the simulated Cooling-loop system

FIG 5.37

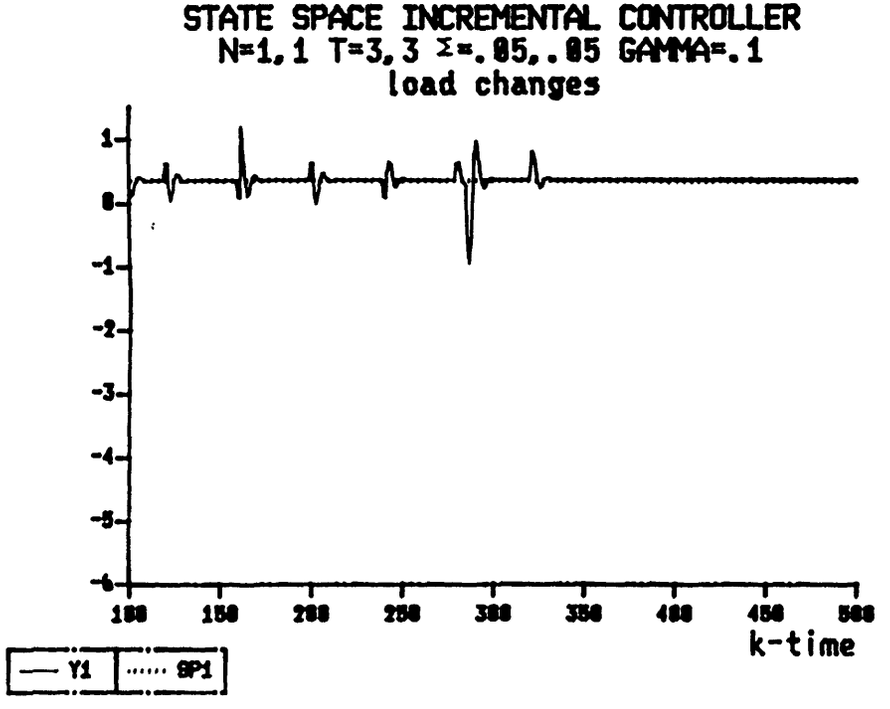
It is seen in Figs. (5.37.g) and (5.37.h) that the gains of the incremental controller remain constant, but the gains of the positional algorithm vary drastically when the second series of set point changes is applied. It is worth noting that the control gains in the positional algorithm converge [from time k=150 to k=300] to different values than the gains of the incremental algorithm. Indeed, the positional off-diagonal gain G_{12} is different from zero in the positional case, causing the deviations in y_2 noted in Fig. (5.37.d) during the first 300 sample intervals.

Considering that the simulated system is non-interactive, the off-diagonal gains should be zero. This means that the positional regulator converged to incorrect parameter estimates in the first part of the simulation. On the other hand, the incremental self-tuner converged to suitable parameters, and produced a good and stable control.

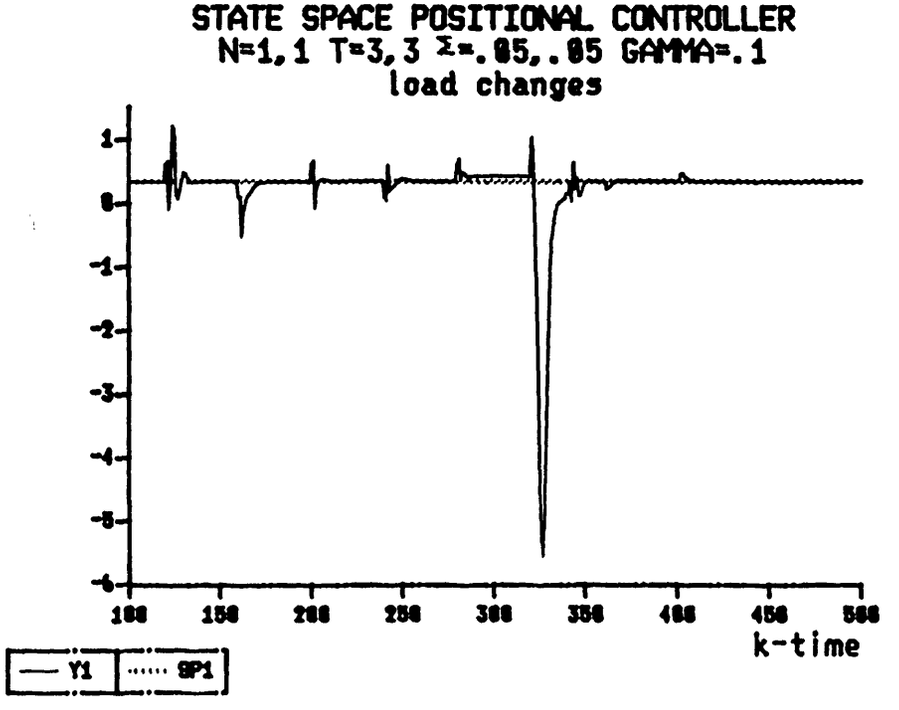
The results above were reminiscent of those obtained with SISO STR's in the level control-loop [see Fig. (5.9)].

In Fig. (5.38), the same simulated system was studied, but, in this case, instead of changing the set points the bias parameters δ_1 and δ_2 were changed. The disturbances were described by a square wave which, every 50 intervals, changes from δ=-.512 to δ=-.256 and vice versa. From time k=100 to k=300, only δ_1 was disturbed and from time k=300 to k=500, δ_1 remained constant and δ_2 was disturbed.

It can be seen in Figs. (5.38.a) to (5.38.d) that both algorithms can control the system and neither of them show control-interaction [Figs. (5.38.e) and (5.38.f)]. However, the outputs of the positional controller present considerable deviations when the second disturbance starts entering the system [k=300]. A small offset also appears in the output y_1 at the end of the first series of disturbances. Both algorithms show drastic changes in their off-diagonal control gains throughout the entire simulation [Figs. (5.38.g) to (5.38.j)], sometimes affecting the control performance; however, the incremental algorithm can easily recover from these deviations, whereas the positional algorithm has substantial difficulties in doing so.

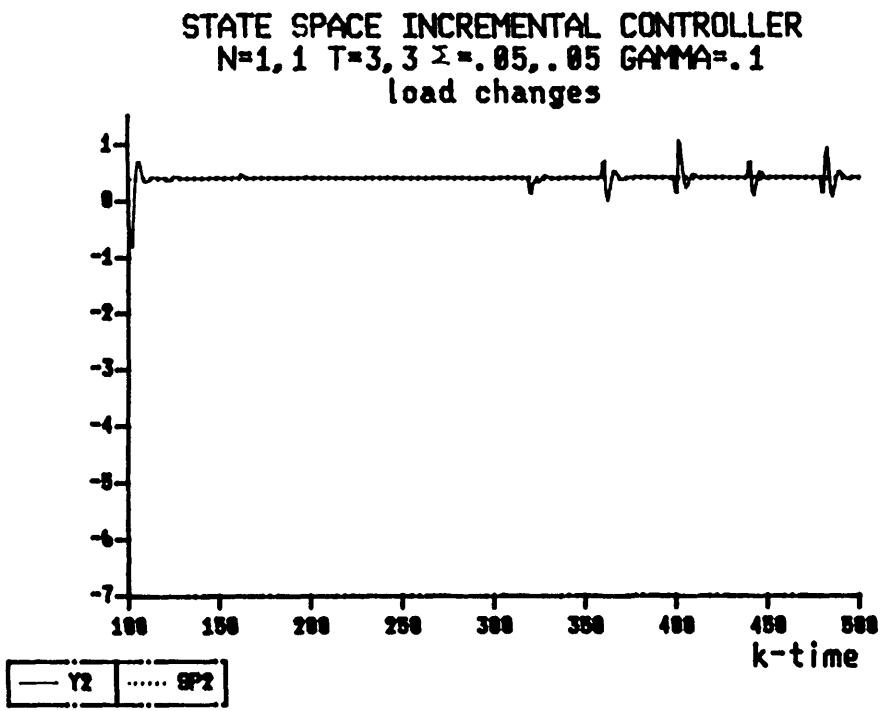


a) Output (Y1) and set point (SP1) of the simulated Heat-exchanger Cooling-loop

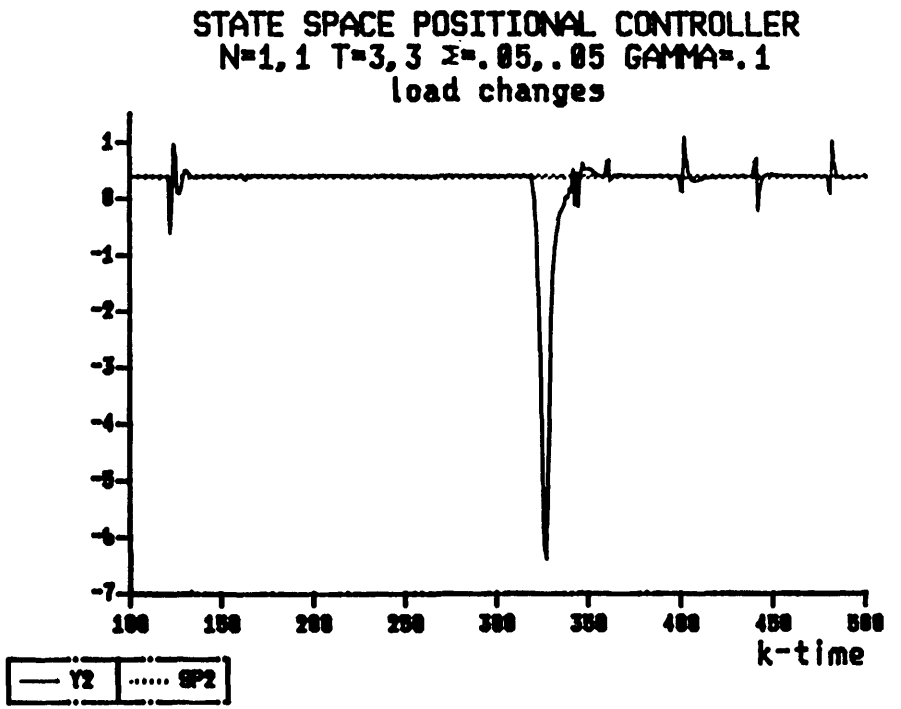


b) Output (Y1) and set point (SP1) of the simulated Heat-exchanger Cooling-loop

FIG 5.38

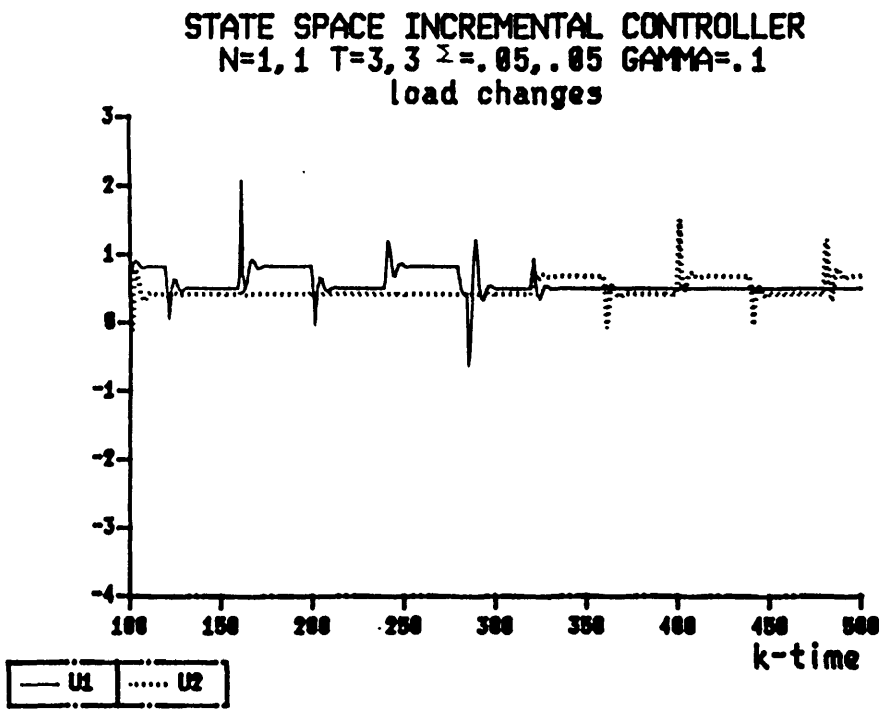


c) Output (Y2) and set point (SP2) of the simulated Condenser Cooling-loop

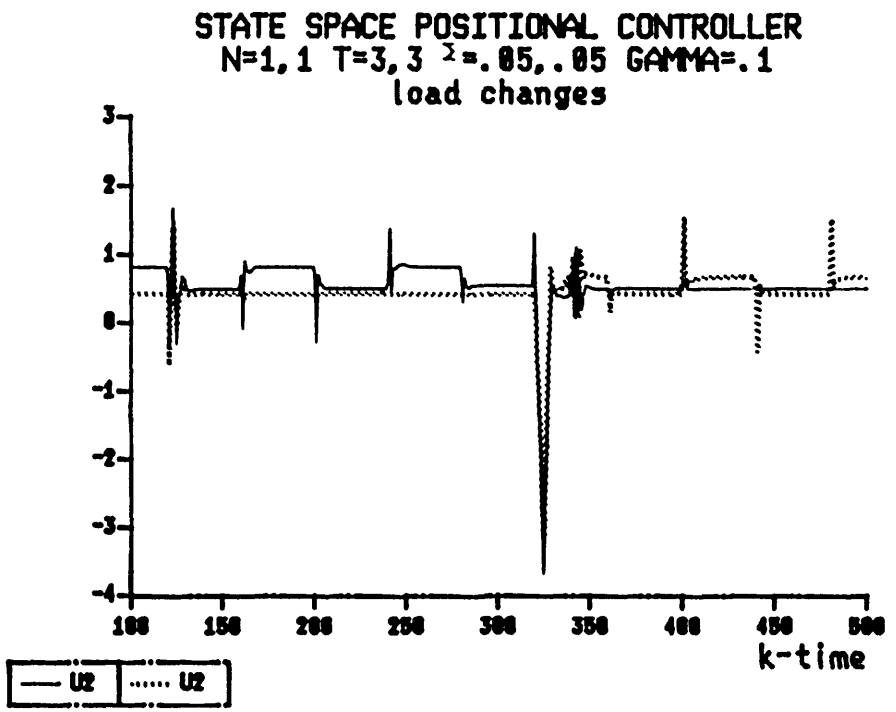


d) Output (Y2) and set point (SP2) of the simulated Condenser Cooling-loop

FIG 5.38

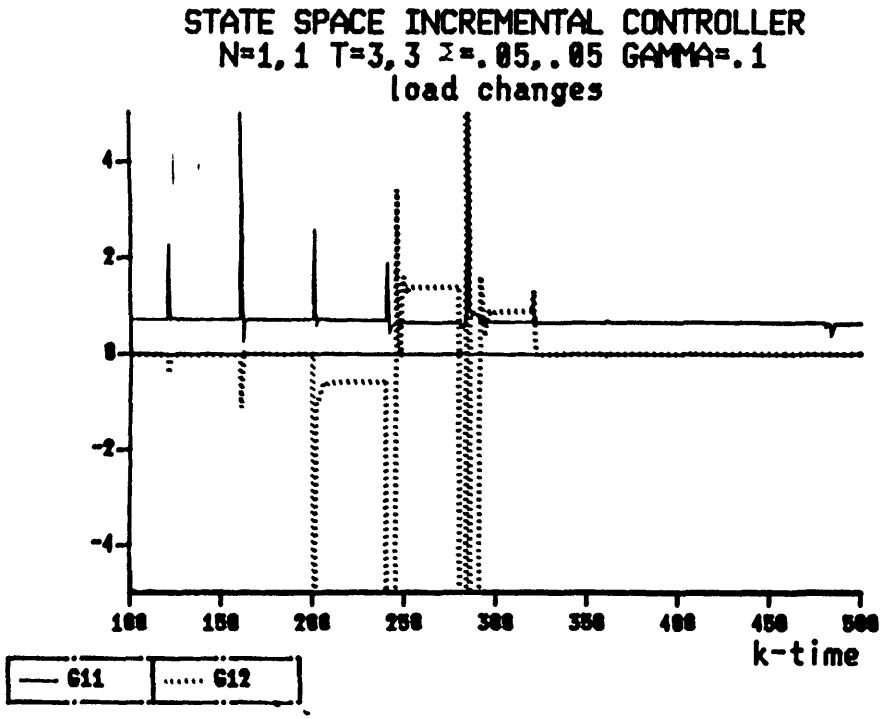


e) Input (U1) of the simulated Heat-exchanger Cooling-loop and input (U2) of the simulated Condenser Cooling-loop

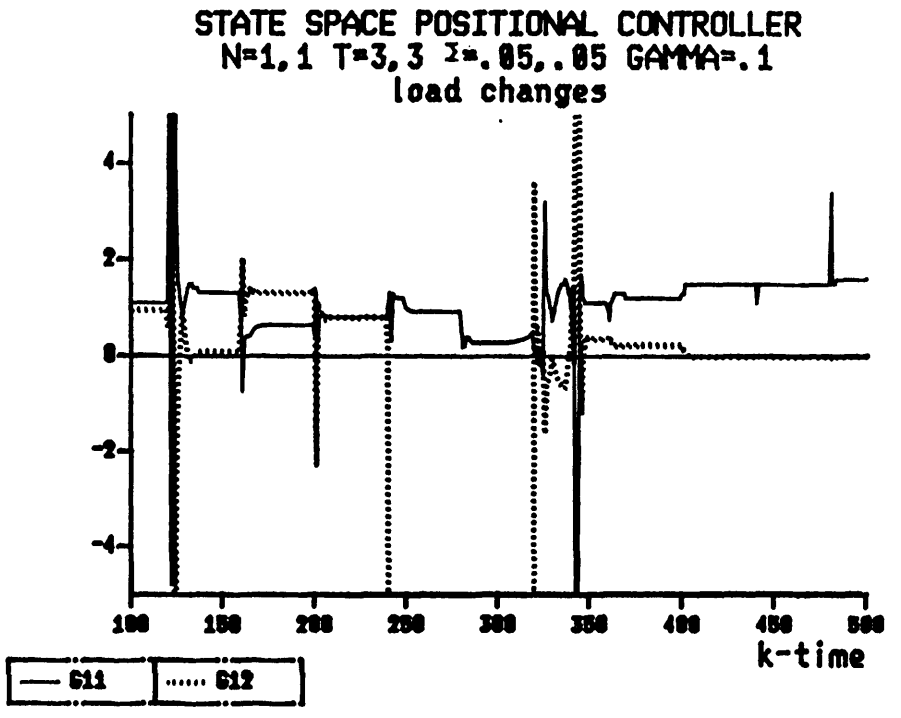


f) Input (U1) of the simulated Heat-exchanger Cooling-loop and input (U2) of the simulated Condenser Cooling-loop

FIG 5.38

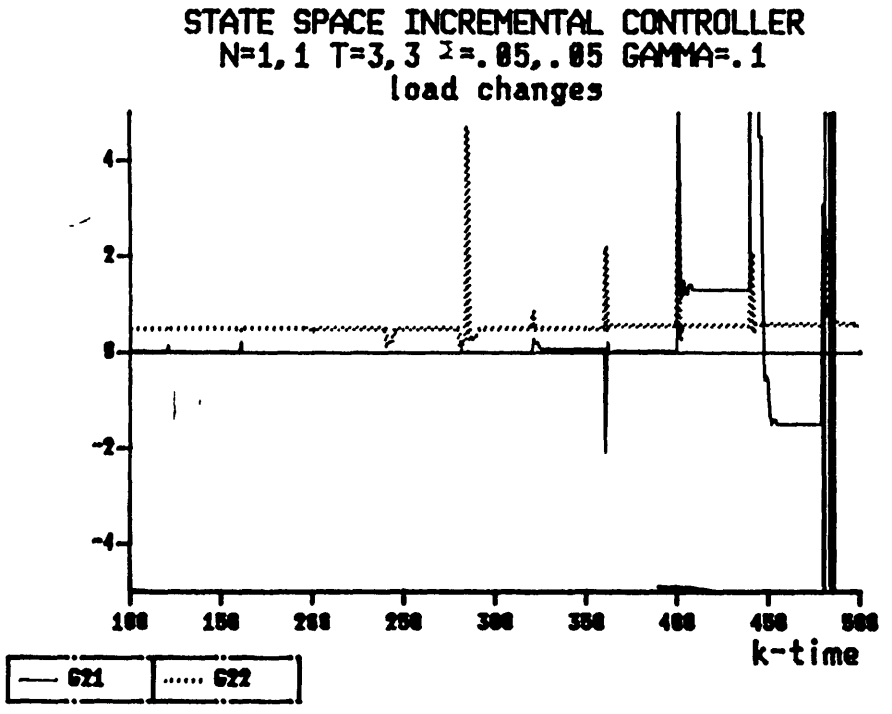


g) Controller gains of the simulated Cooling-loop system

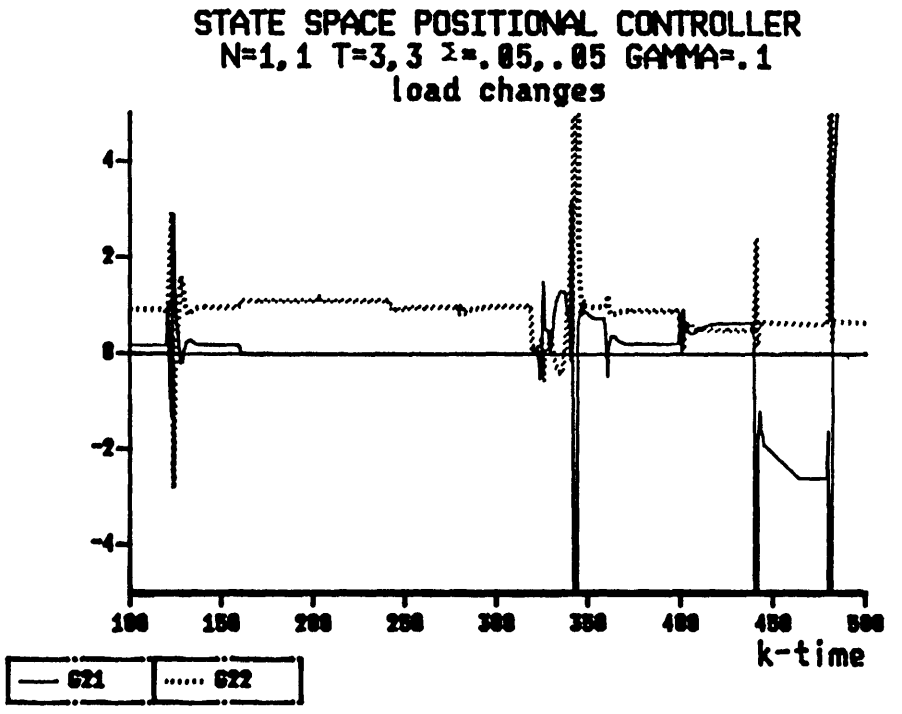


h) Controller gains of the simulated Cooling-loop system

FIG 5.38



i) Controller gains of the simulated Cooling-loop system



j) Controller gains of the simulated Cooling-loop system

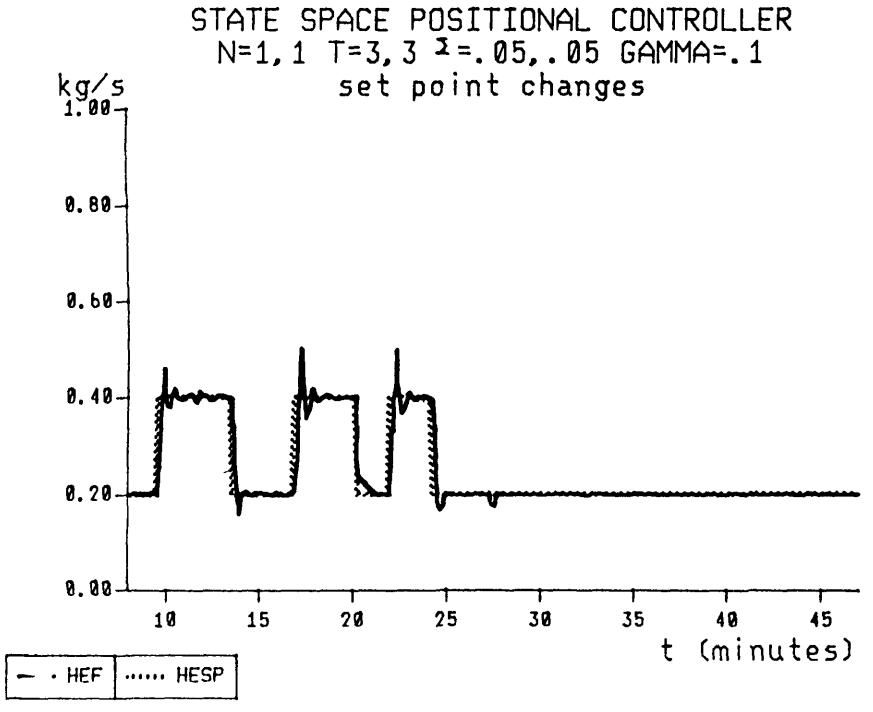
FIG 5.38

Experimental Results

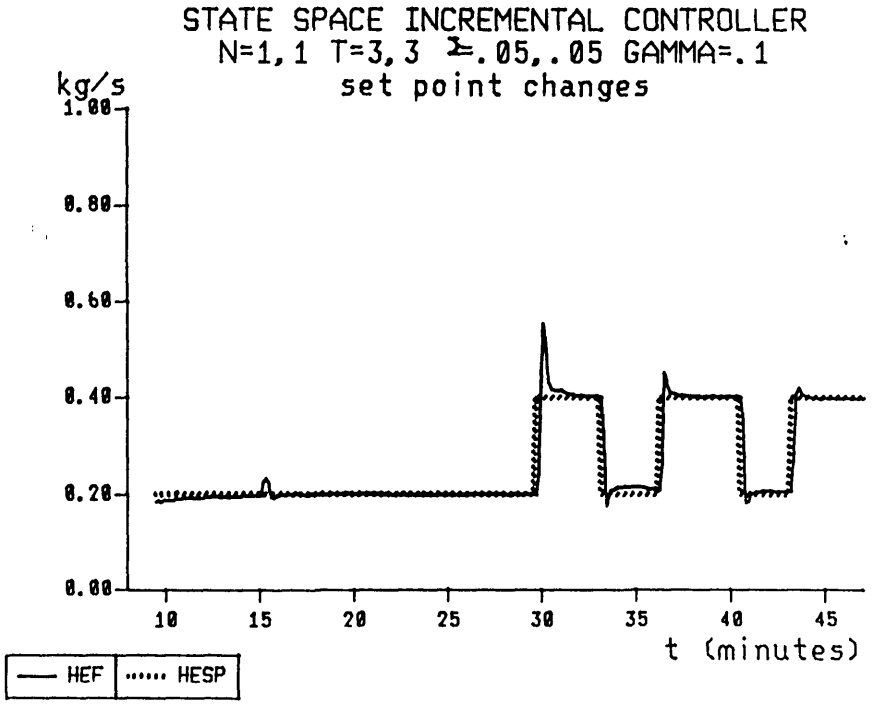
Figures (5.39) and (5.40) show the results obtained when both (positional and incremental) algorithms were used to control the Heat-exchanger and the Condenser cooling loops [see Fig. 4.1]. In these figures HE denotes Heat-exchanger and C denotes Condenser; F is used for flow, V for valve and SP for set point. Then, HEF denotes the cooling water flow in the heat-exchanger, CV the valve position in the condenser and so on. In all the experiments shown below, the initial settings were chosen to be the same. The information lengths Σ_0 were .05, the time-horizons T were 3 and the filter gains [gamma] were set to .1, in both outputs. The sample time ST was 8 sec .

Figure (5.39) compares the results when the positional and incremental algorithms were used to control the cooling-system. In these experiments the cooling-system was subjected to a series of set point changes. These were first applied to the Heat-exchanger cooling-loop and then to the Condenser cooling-loop. It can be seen in Figs. (5.39.a) to (5.39.f) that both algorithms present a similar performance. The set point changes have associated overshoots. The positional self-tuner remains stable when the second series of set point changes is applied, but the output in the condenser starts to oscillate. Apparently, the change in bias in this case is not large enough to produce instability. In both experiments the gains remained bounded, with some bias in the off-diagonal gains, above all in the incremental algorithm [Figs. (5.39.g) to (5.39.j)]. Nevertheless, deviations are small and performance is clearly acceptable.

Figure (5.40) shows the results obtained when the cooling-system was subjected to step load changes by varying the position of the manual valves. This is an unmeasurable disturbance which will clearly affect the bias parameters. In this case the performance of the positional regulator [Figs. (5.40.a), (5.40.c) and (5.40.e)] is considerably worse than the performance of the incremental regulator [Figs. (5.50.b), (5.40.d) and (5.40.f)]. The outputs of the system controlled by the positional self-tuner present large deviations and eventually get unstable. On the other hand, the incremental adaptive regulator easily copes with the unmeasured disturbances, causing no upset in the loop not affected by the disturbance. In Figs. (5.40.g) to (5.40.j) it can also be seen that despite the size of the load changes, the controller gains of the incremental regulator remain bounded; the gains of the positional regulator change almost without limit.

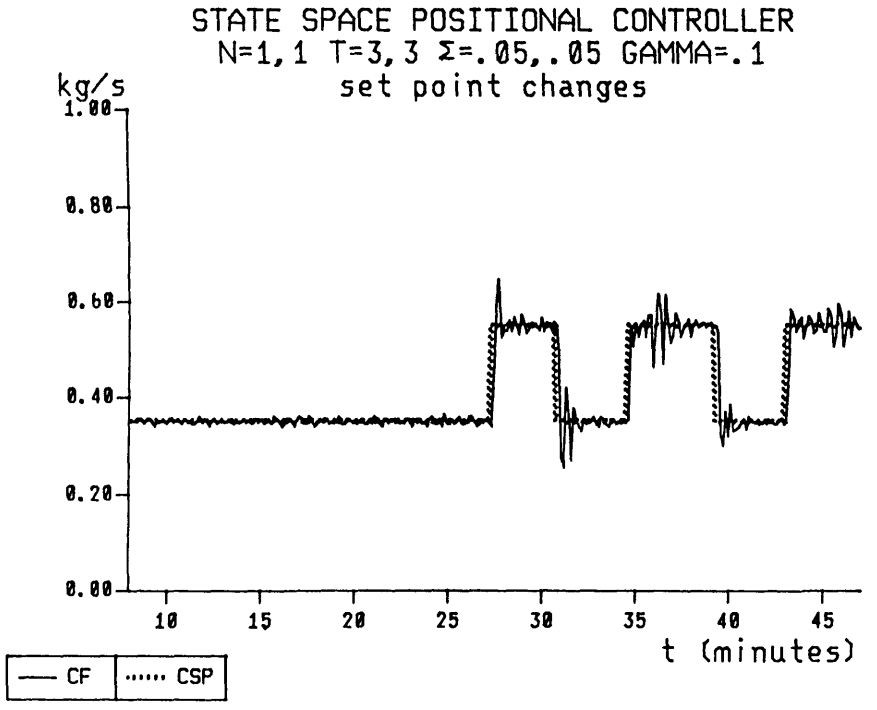


a) Flow (HEF) and set point (HESP) of the Heat-exchanger

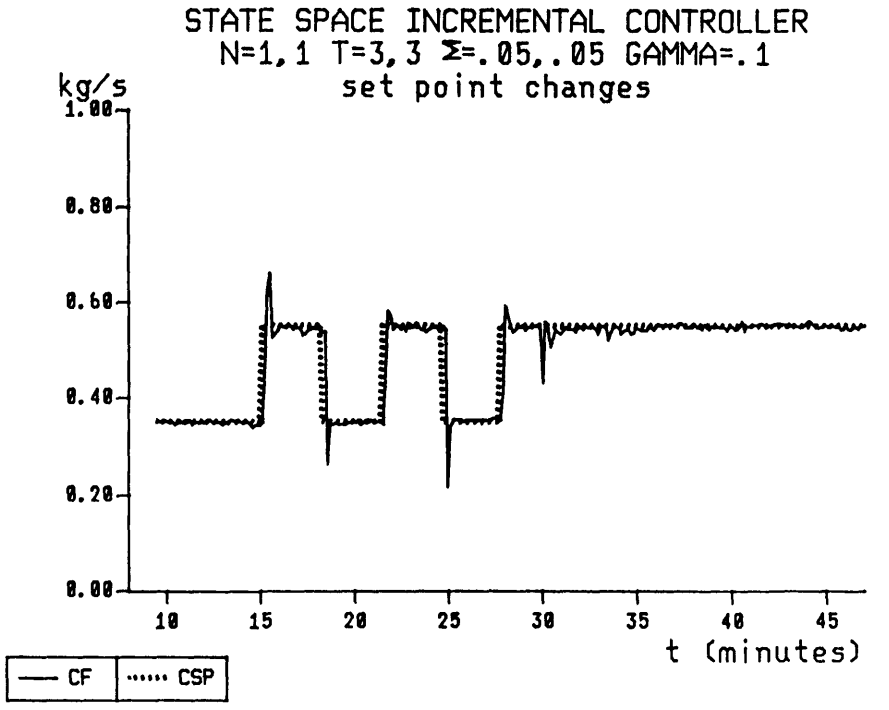


b) Flow (HEF) and set point (HESP) of the Heat-exchanger

FIG 5.39

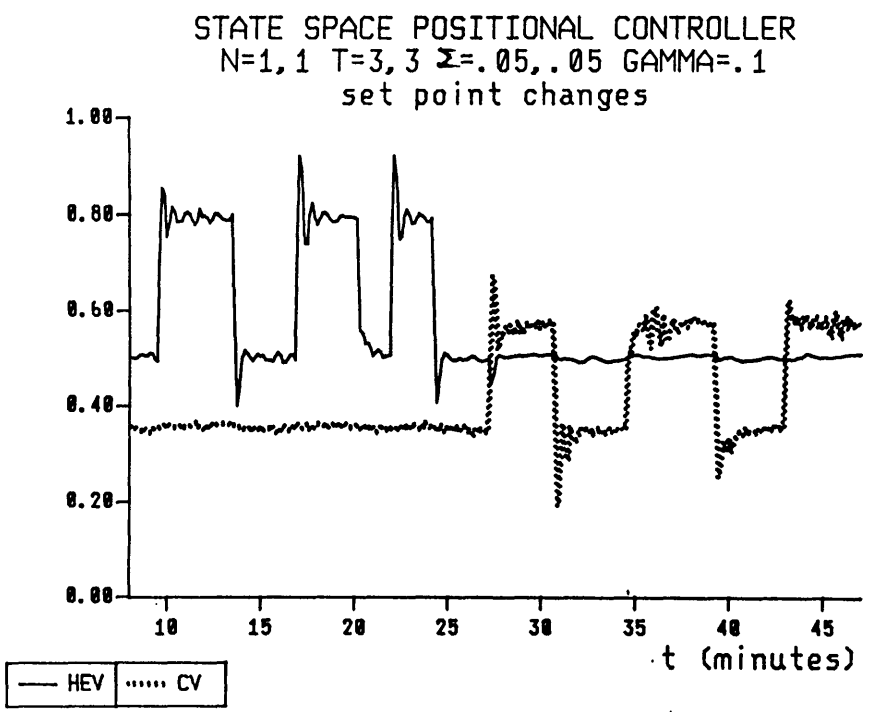


c) Flow (CF) and set point (CSP) of the Condenser

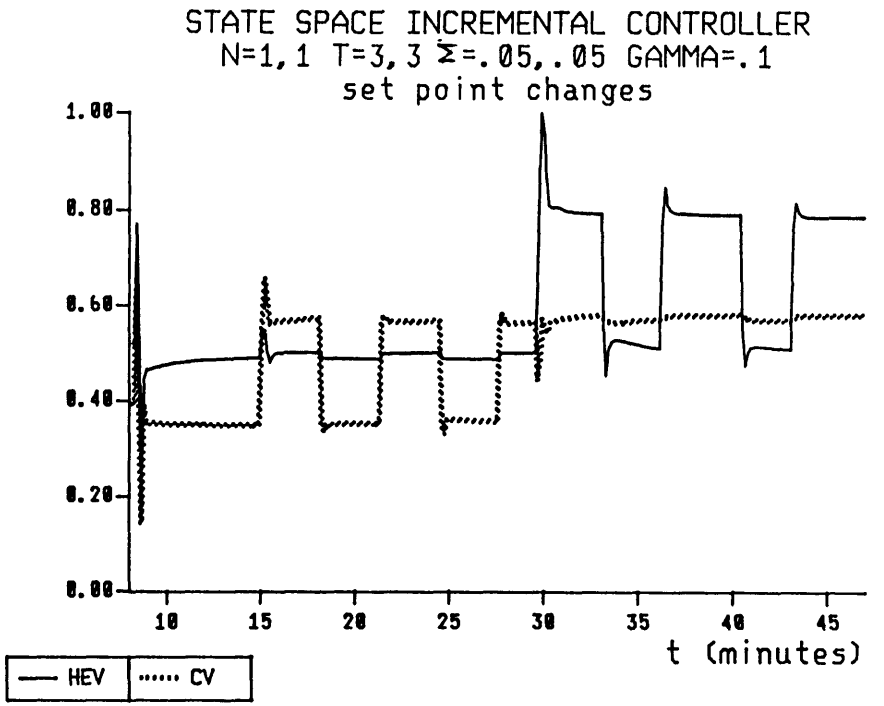


d) Flow (CF) and set point (CSP) of the Condenser

FIG 5.39

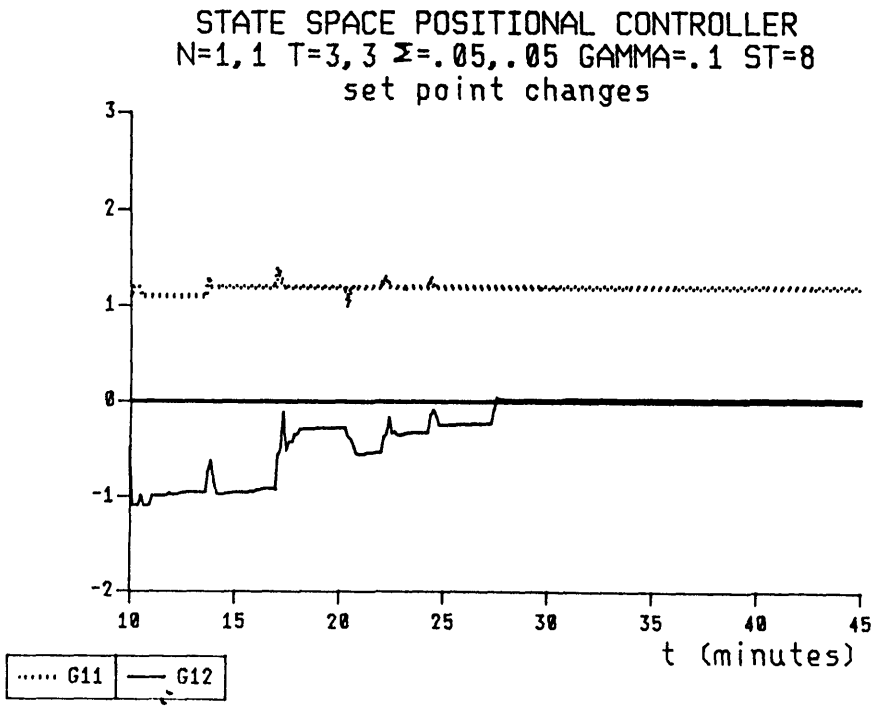


e) Valve positions of the Heat-exchanger (HEV) and Condenser (CV)

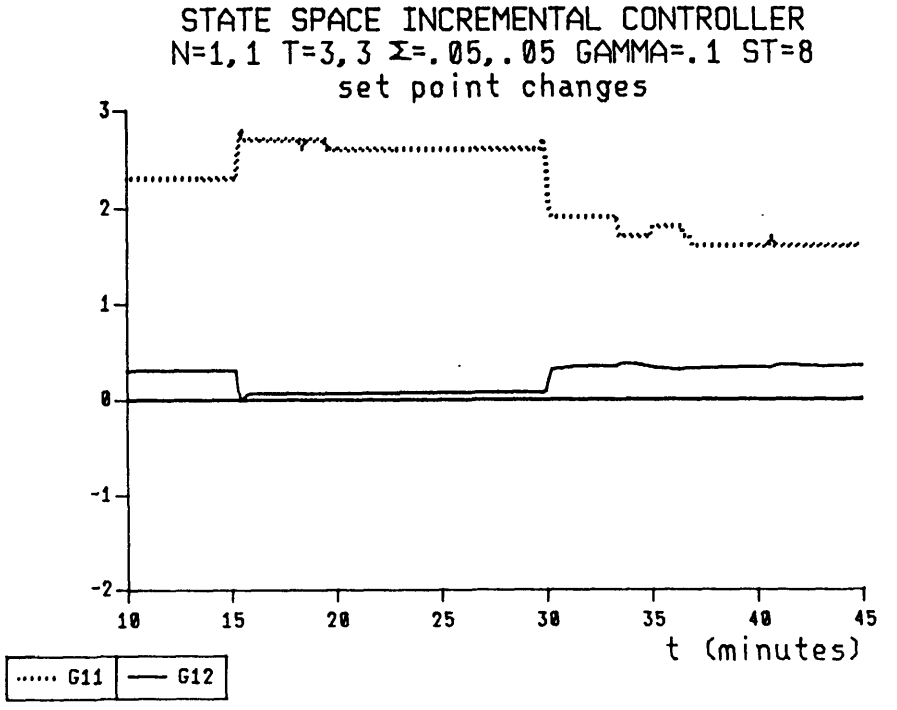


f) Valve positions of the Heat-exchanger (HEV) and Condenser (CV)

FIG 5.39

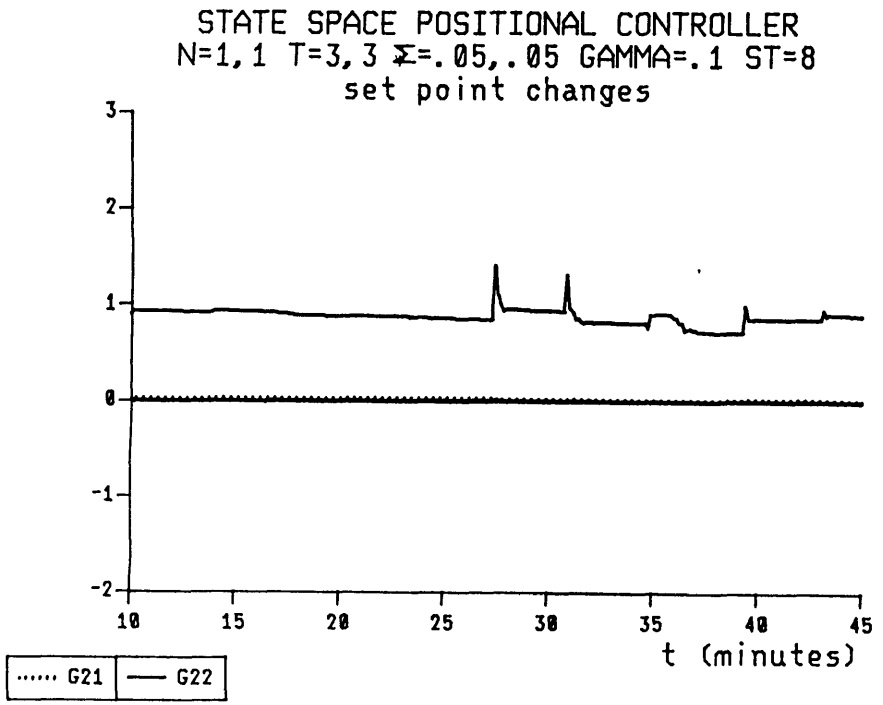


g) Controller gains of the Cooling-system

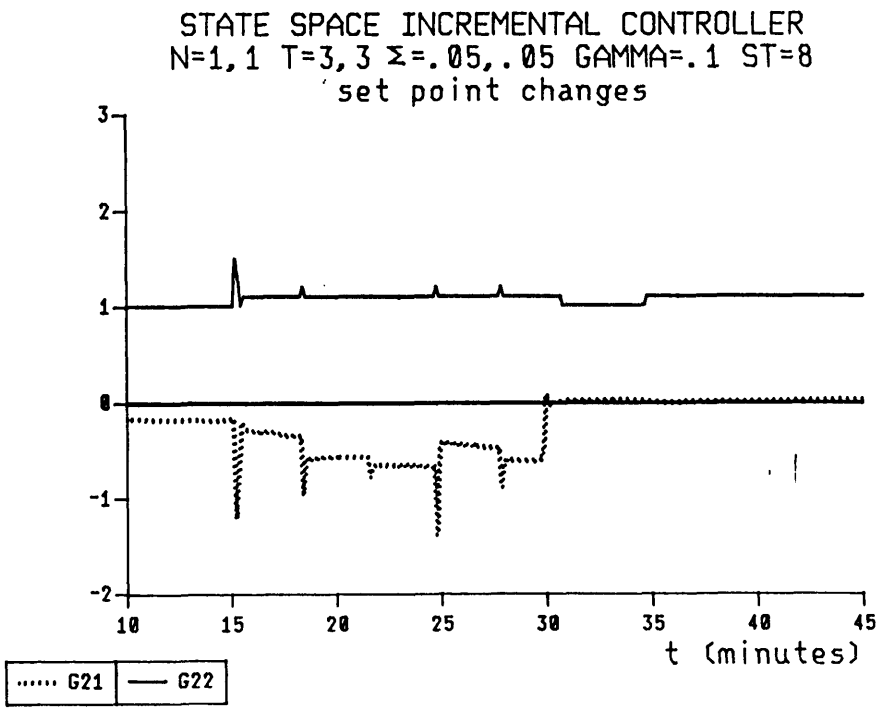


h) Controller gains of the Cooling-system

FIG 5.39

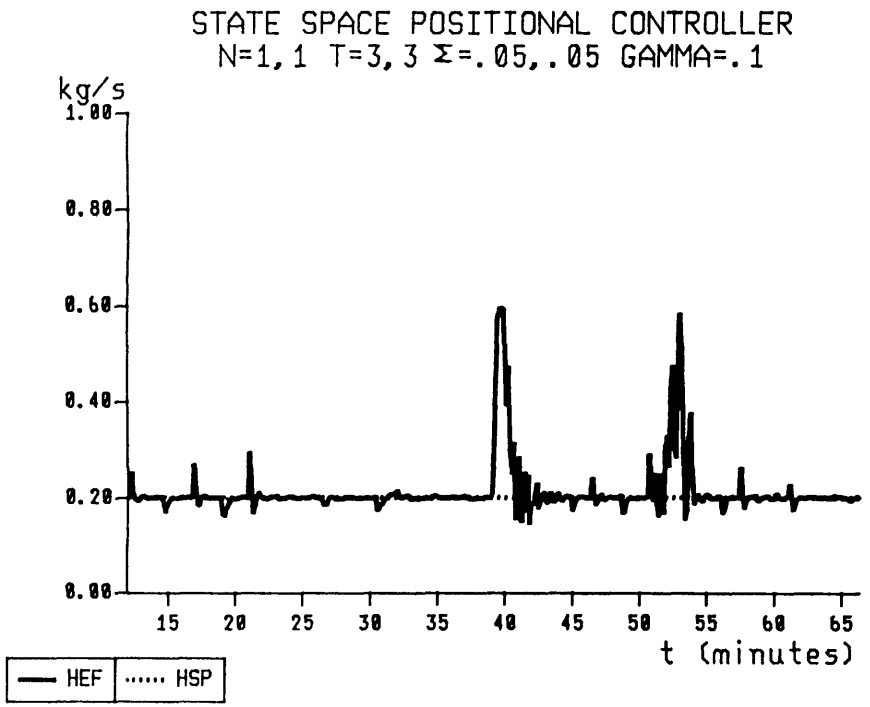


i) Controller gains of the Cooling-system

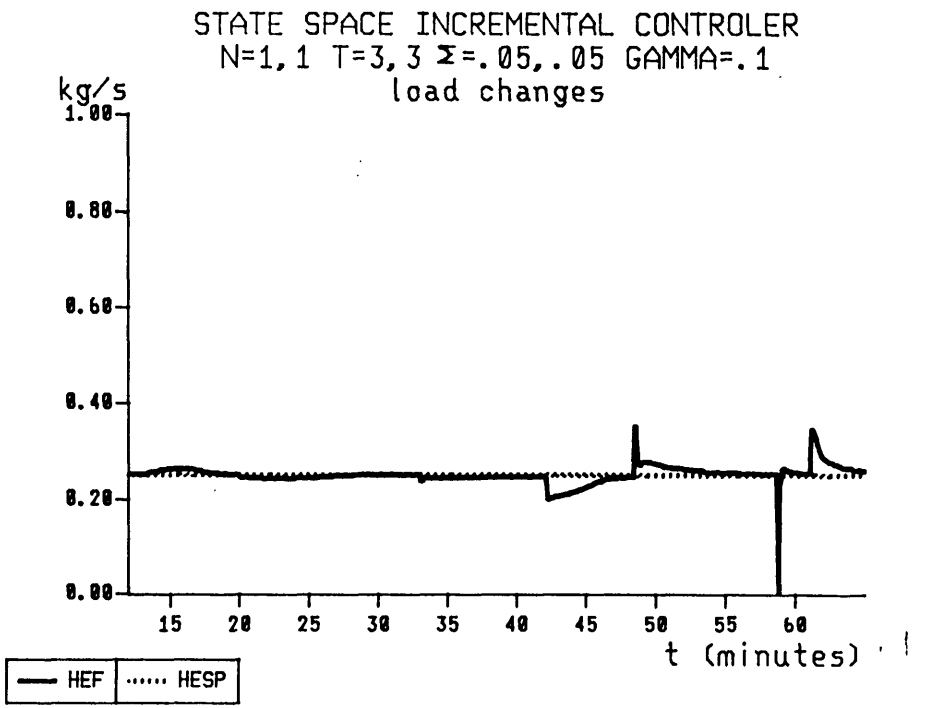


j) Controller gains of the Cooling-system

FIG 5.39

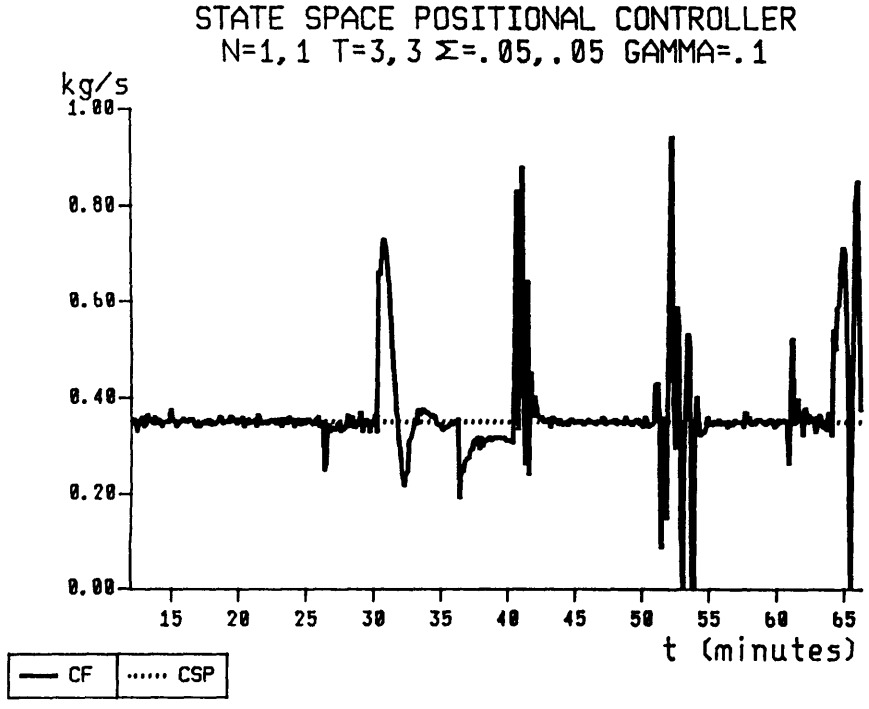


a) Flow (HEF) and set point (HESP) of the Heat-exchanger

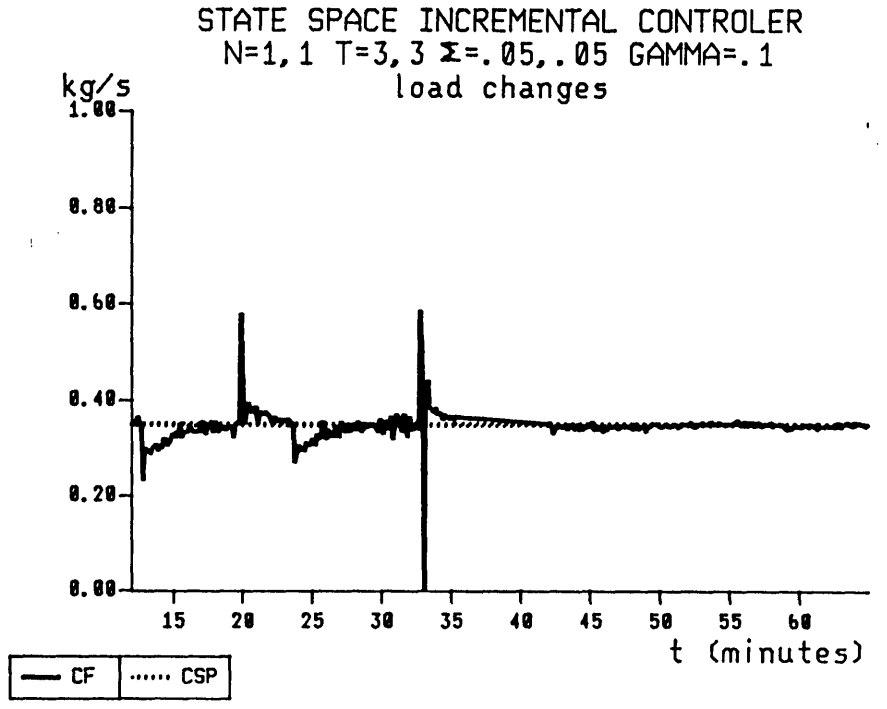


b) Flow (HEF) and set point (HESP) of the Heat-exchanger

FIG 5.40

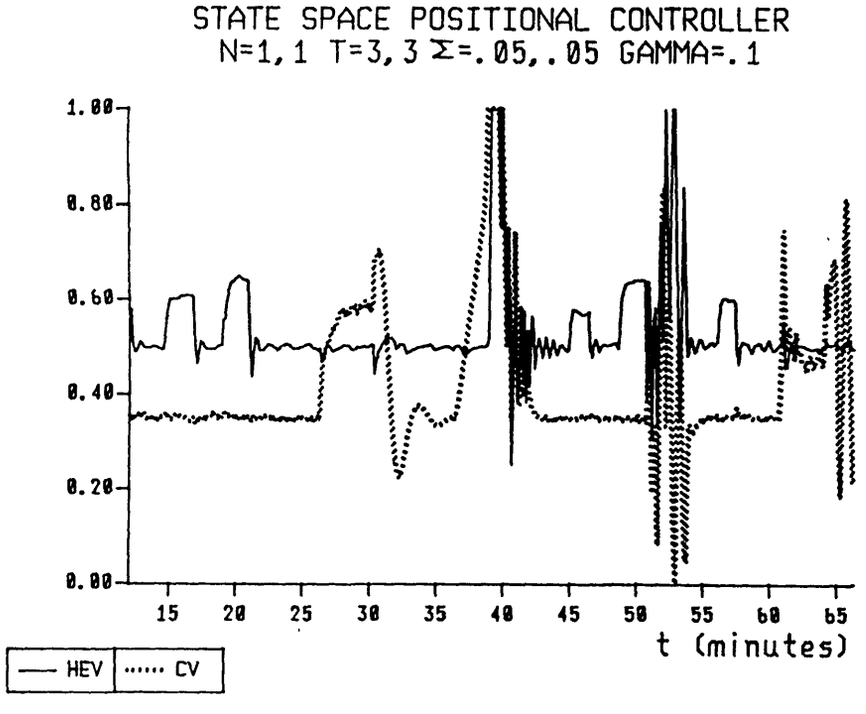


c) Flow (CF) and set point (CSP) of the Condenser

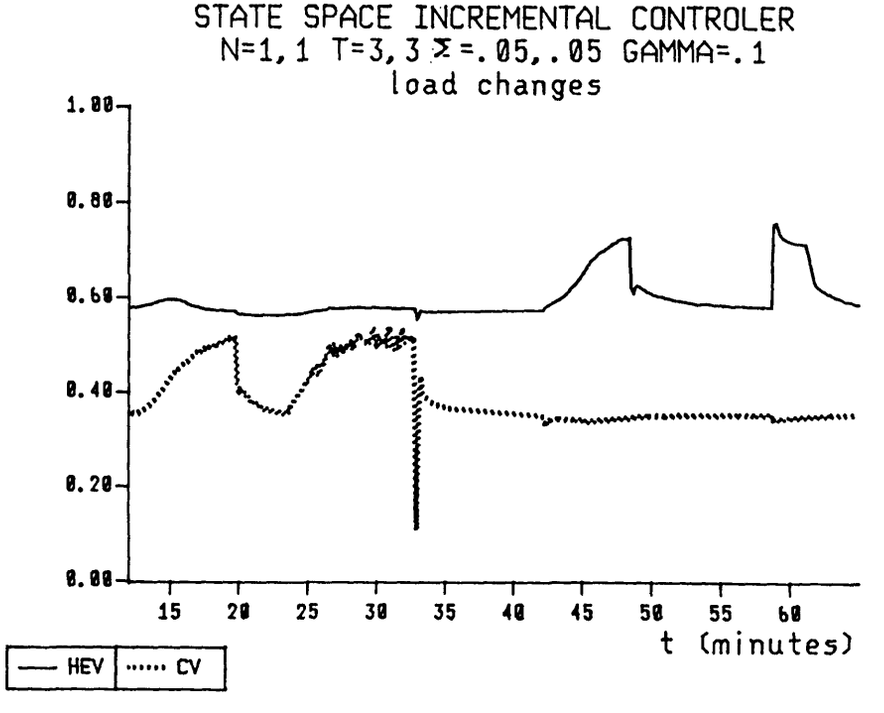


d) Flow (CF) and set point (CSP) of the Condenser

FIG 5.40

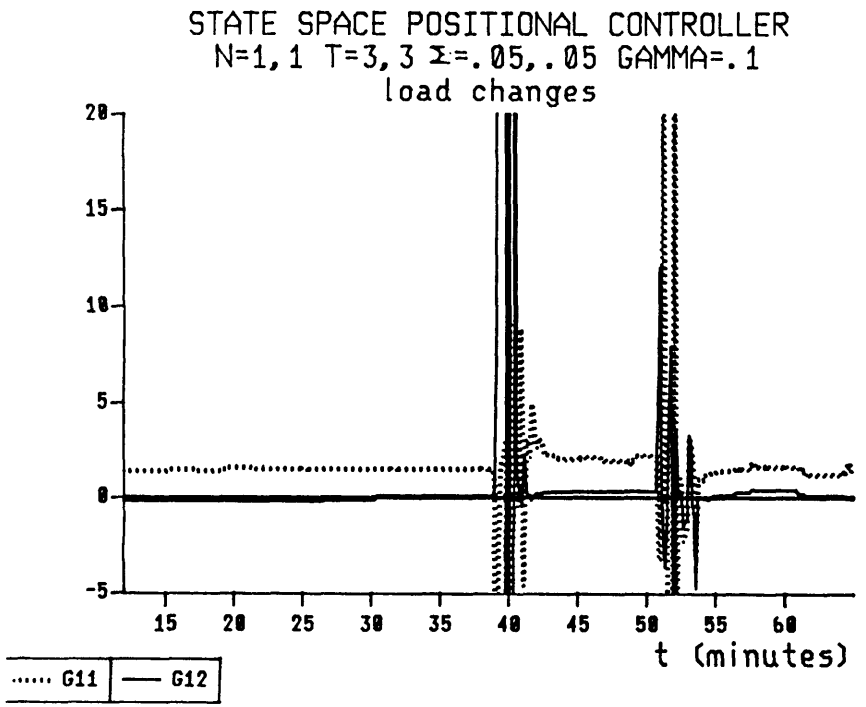


e) Valve positions of the Heat-exchanger (HEV) and Condenser (CV)

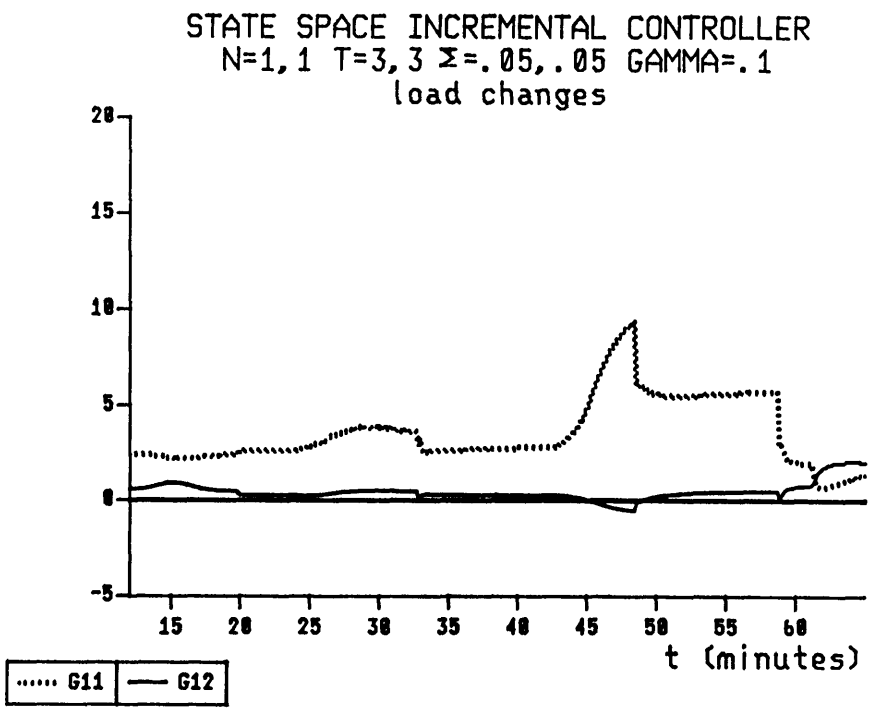


f) Valve positions of the Heat-exchanger (HEV) and Condenser

FIG 5.40

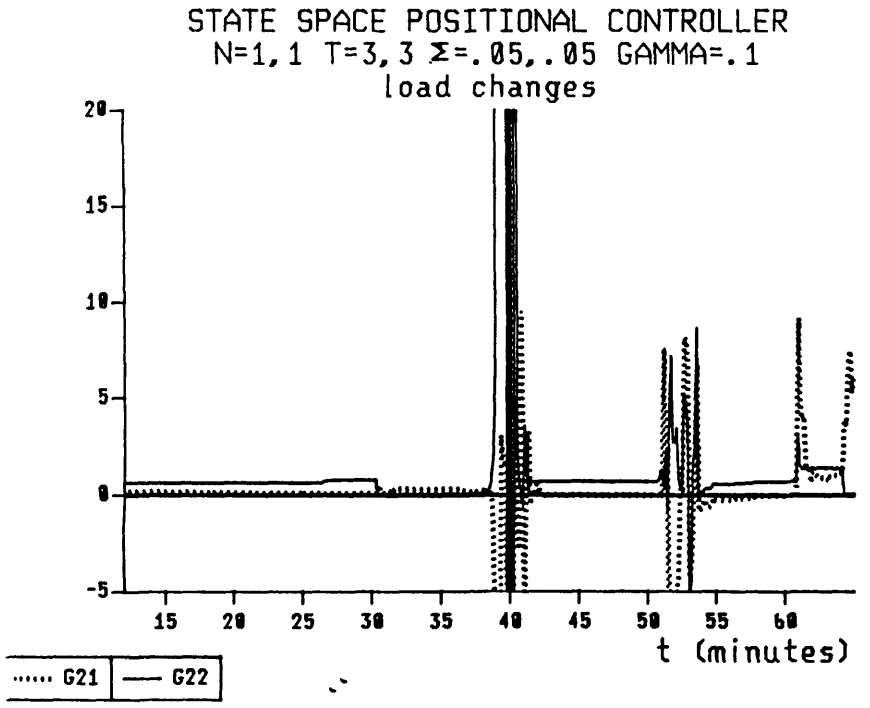


g) Controller gains of the Cooling-system

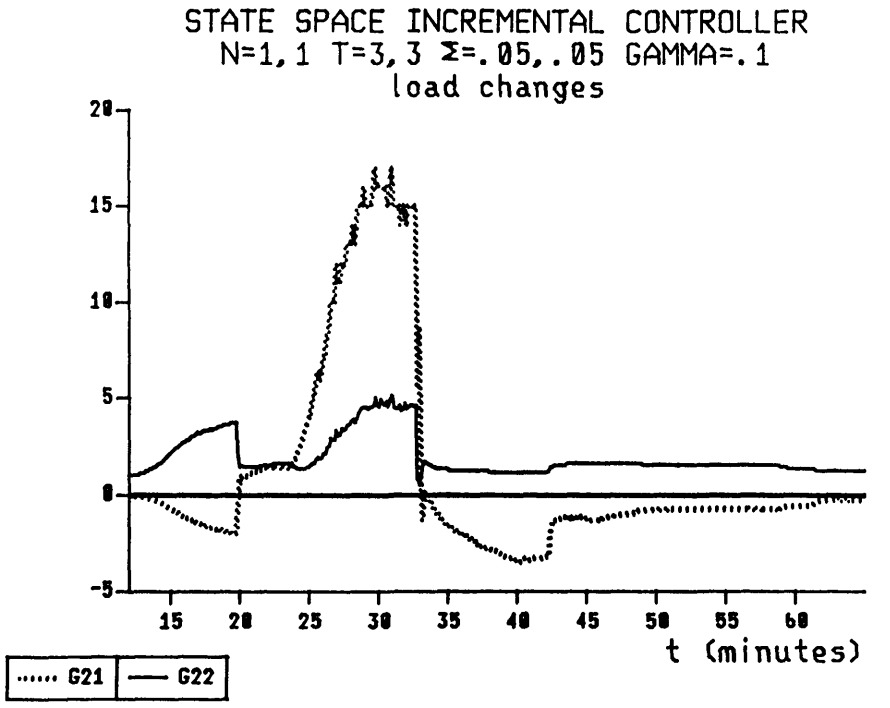


h) Controller gains of the Cooling-system

FIG 5.40



i) Controller gains of the Cooling-system



j) Controller gains of the Cooling-system

FIG 5.40

It has been shown [Hiram and Kershenbaum (1985), Ydstie et al (1985)], that the cooling-system can easily be controlled with SISO STR's. This is expected since the system is non-interactive at all and with SISO controllers the interaction parameters need not be estimated. Moreover, most of the problems shown in non-interactive systems by the MIMO controllers [overshoots and interaction between control-loops] are related with the tuning of the off-diagonal controller gains. Of course, this problem does not arise if SISO STR's are employed. Nevertheless, it is instructive to demonstrate that the multivariable adaptive state-space algorithm in its incremental form can produce an acceptable control in a non-interactive system without placing constraints in the off-diagonal gains.

In short, this section has shown that the positional MIMO self-tuner is sensitive to unmodelled bias caused by non-linearities and load changes. Under these conditions, the controller performance is very poor, making any real-time implementation of this algorithm impractical. The performance may be improved if bias parameter estimation is included in the adaptive controller; however, as it was shown in section (5.1) this is also a bad strategy when large and frequent load changes are expected.

A far better solution is the use of an incremental implementation of the MIMO controller which appears to perform much better in simulations and real-time experiments. Many more experiments are needed to complete the test of this algorithm, mainly with interactive systems; however, the evidence collected so far suggests that with further developments the proposed multivariable adaptive controller may be a good alternative in industrial processes.

5.2.3.- MIMO State-space Feedback-feedforward Control

A positional FB/FF state-space multivariable controller [proposed in section (3.2.2] has been studied with linear and quasi-nonlinear simulations. The sensitivity of the FB/FF algorithm, to the selection of different control-horizons and different identification structures, has been tested. The FB/FF algorithm has also been compared with a purely FB state-space controller.

The two-input two-output models used in the simulations of this section [models 7, 8 and 9 of Appendix H] are steady-state non-interactive, so that the study is focused on the effects of deterministic disturbances.

In all the simulations shown below, the gain filter γ for state estimation was chosen to be .2. As was shown in last section, the state-space algorithm is very robust to the selection of this parameter.

The first set of simulations shown in this section utilize a linear system [model 7] with constant parameters subjected to two measured deterministic disturbances. These disturbances are described by the same square wave which takes the values 0 and 1 and has a period of 120 sample intervals.

The second set show the results obtained with quasi-nonlinear simulations. Here, the set point of both outputs are simultaneously changed. Before the set point changes, the system is described by a given linear model [model 7 or 8 of Appendix H] and after the set point changes, the system is described by a different linear model [model 8 or 9].

5.2.3.a.- Linear Simulations

In the following set of simulations the sensitivity of the FB/FF self-tuner has been tested with a linear system with constant parameters.

Firstly, the effects of the time horizon on the controlled system performance has been studied in simulations without parameter estimation; namely, the system model is assumed known so that the control uses the true model parameters.

Secondly, the simulations are repeated, but with parameter estimation so that the control uses estimated model parameters. However, the true model structure is employed.

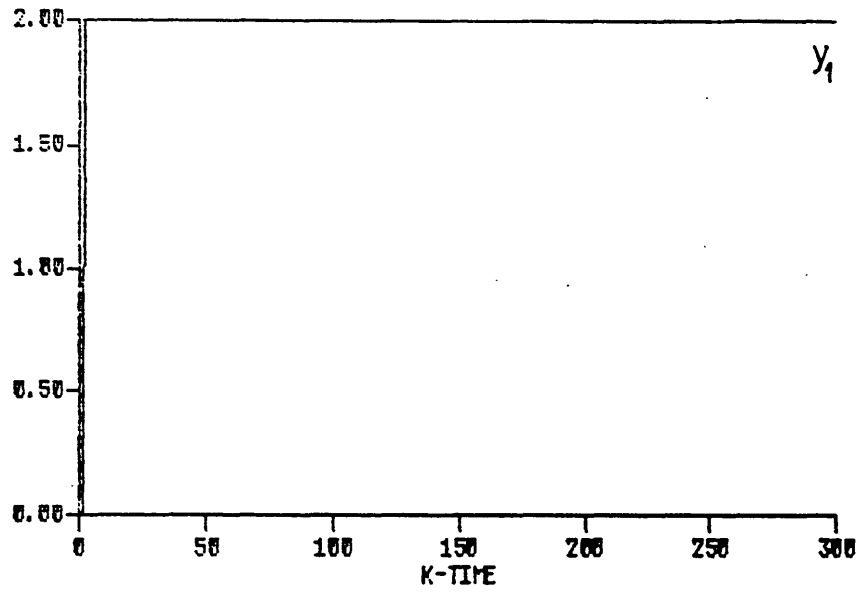
Finally, different identification structures are used to study the robustness of the proposed adaptive algorithm. Of course, in this case the model parameters are estimated.

Non-adaptive Simulations with Different Horizons

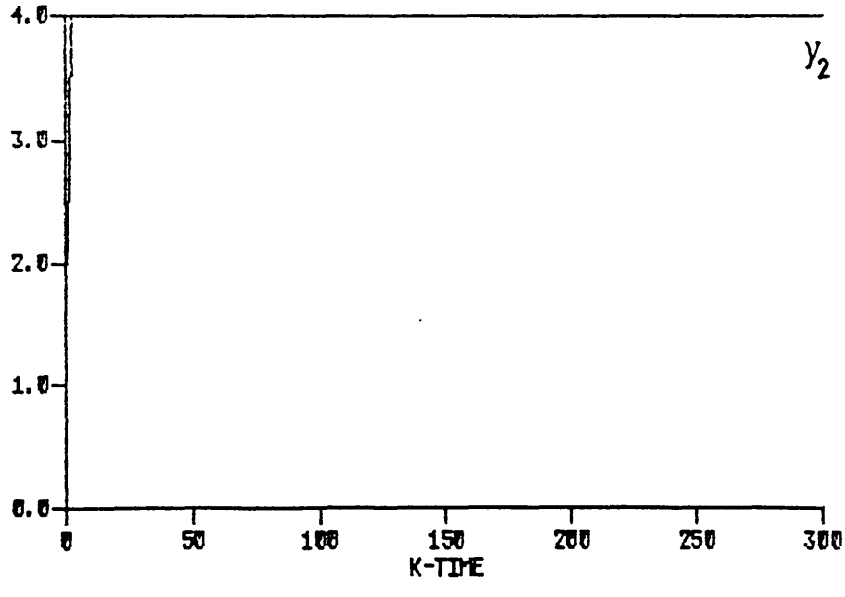
In the simulations shown in Figs. (5.41) to (5.43), the true model parameters were utilized in the control calculation so that no parameter estimation is needed. Moreover, the outputs are noise-free. In this case, the sensitivity of the control-law to the selection of time-horizons can be clearly appreciated.

Figures (5.41.a) and (5.41.b) give the response of the controlled outputs when the system was affected by the previously described disturbances; namely, square waves of period 120 k-times, switching between 0 and 1. In this case, an output dead-beat control-law [T=1] was employed. The figures show perfect regulation; however, strong control-actions are required [Figs. (5.41.c) and (5.41.d)], especially on the first input. This excessive control-action is typical of dead-beat control. It is worth mentioning that the steady-state gains between the disturbances and the output are quite moderate [of the order of one]. However, as it can be deduced from Fig. (5.41.c), the dynamic gains associated with the first output are very high.

FIG 5.41: FB/FF NON-ADAPTIVE SIMULATION
T=1,1 ϒ=.2 model 7

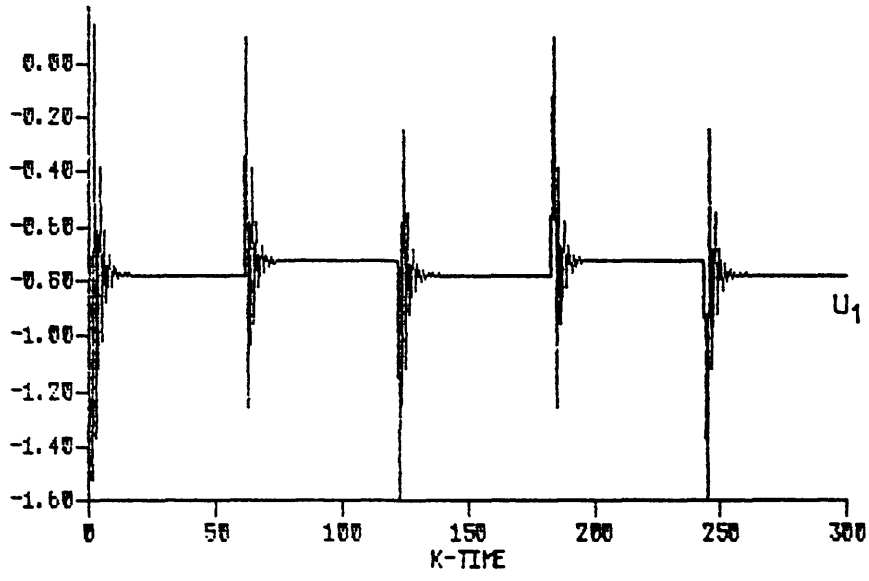


a) First output of the MIMO system

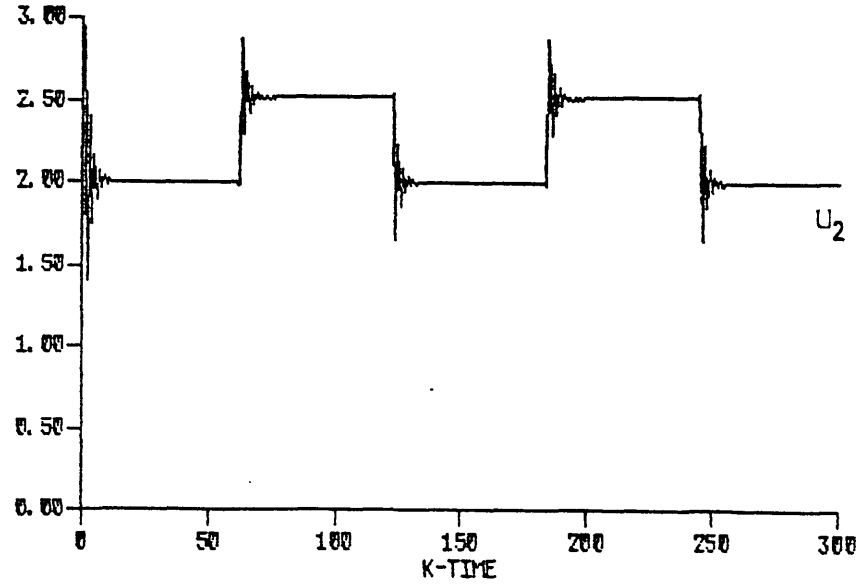


b) Second output of the MIMO system

FIG 5.41: FB/FF NON-ADAPTIVE SIMULATION
T=1,1 ϒ=.2 model 7



c) First input of the MIMO system



d) Second input of the MIMO system

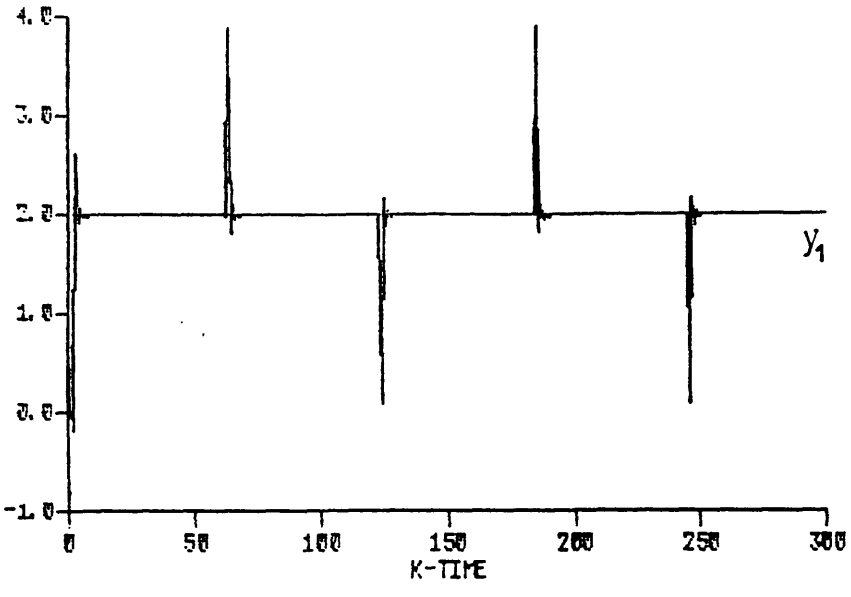
Figure (5.42) shows the results obtained when the control-action was detuned by increasing the time-horizon. It can be seen in Fig. (5.42.a) that the effect of the disturbances is strongly felt by the first output, however, the deviations shown by the second output [Fig. 5.42.b)] are quite moderate. On the other hand, Figs. (5.42.c) and (5.42.d) show very smooth control-actions, denoting the good detuning properties of the extended-horizon controller.

A compromise between the previous responses was attempted in the simulation shown in Fig. (5.43). The control-horizons were chosen to be different for each output: 1 for the first output, 3 for the second one. A perfect regulation is observed in Fig. (5.43.a) for the first output and moderate deviations are shown by the second output [Fig. (5.43.b)]. On the other hand, no detuning is obtained for the first input [Fig. (5.43.c)]. The second input [Fig. (5.43.d) appears detuned compared with Fig. (5.41.d), but less detuned than the control shown in Fig. (5.42.d). This may be due to some degree of dynamic interaction not accounted for by the Bristol method.

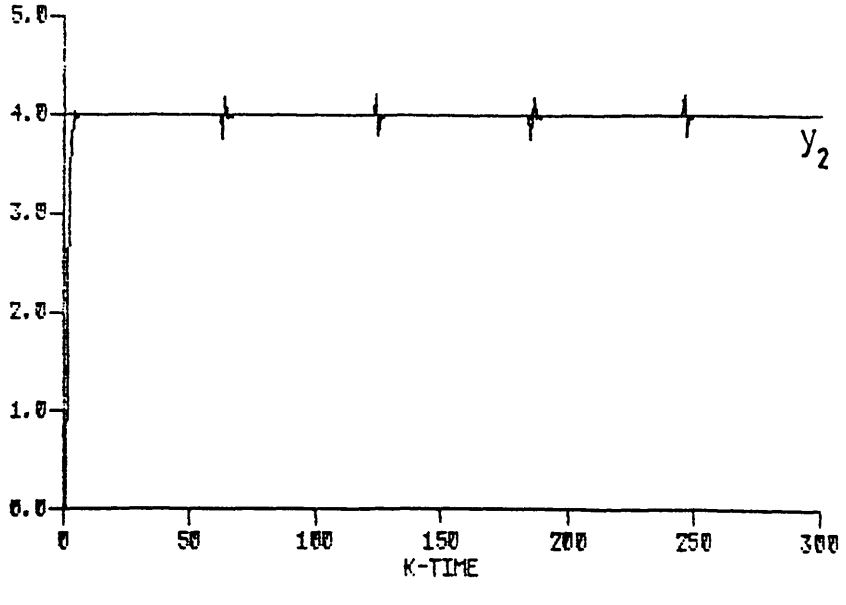
Adaptive Simulations with Different Horizons

Figures (5.44) to (5.46) give the simulation results obtained with a system with known structure and constant but unknown parameters; so that in these simulations, the control utilizes estimated parameters. It should be emphasized that the algorithm estimates both the process and the disturbance parameters. In addition, the outputs are contaminated with white noise of standard deviation σ=.1. The disturbances that perturb the system were described above. The initialization consists of constant input sequences of 50 sample times applied to both controls. There is no disturbance during the initialization period, this would explain the large deviation shown by the outputs at time k=60, when the first disturbance enters the controlled system. The adaptive control is switched on at time k=50 and after 100 time intervals, all the parameter estimates generally converge to suitable values. In the following series of simulations, the information lengths are set at the same value [Σ_{o1}= 2 and Σ_{o1}= 2].

FIG 5.42: FB/FF NON-ADAPTIVE SIMULATION
T=2,2 Y=.2 model 7

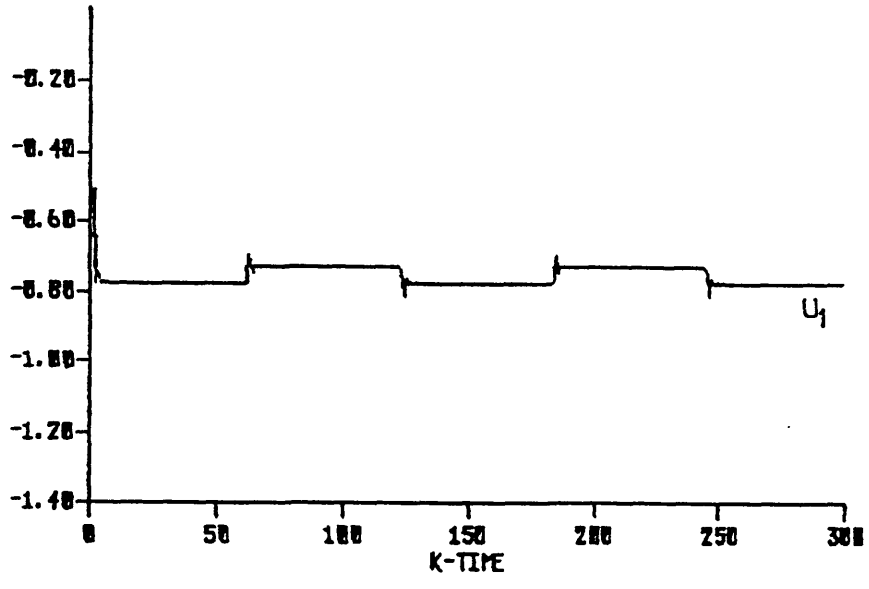


a) First output of the MIMO system

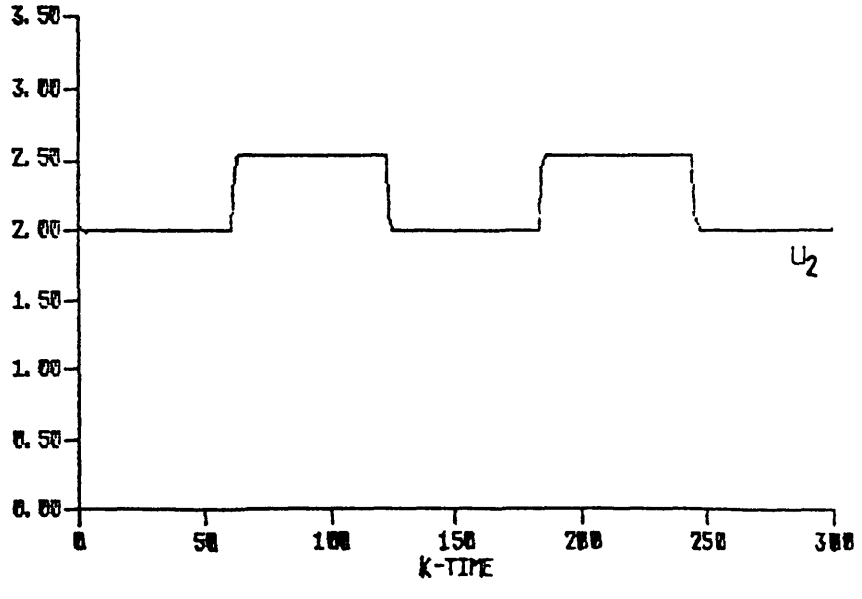


b) Second output of the MIMO system

FIG 5.42: FB/FF NON-ADAPTIVE SIMULATION
T=2,2 Y=.2 model 7

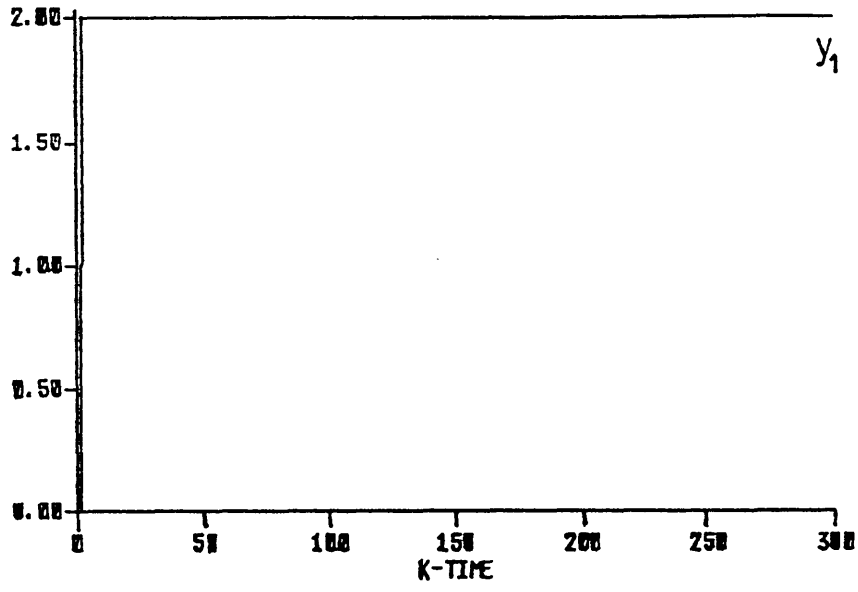


c) First input of the MIMO system

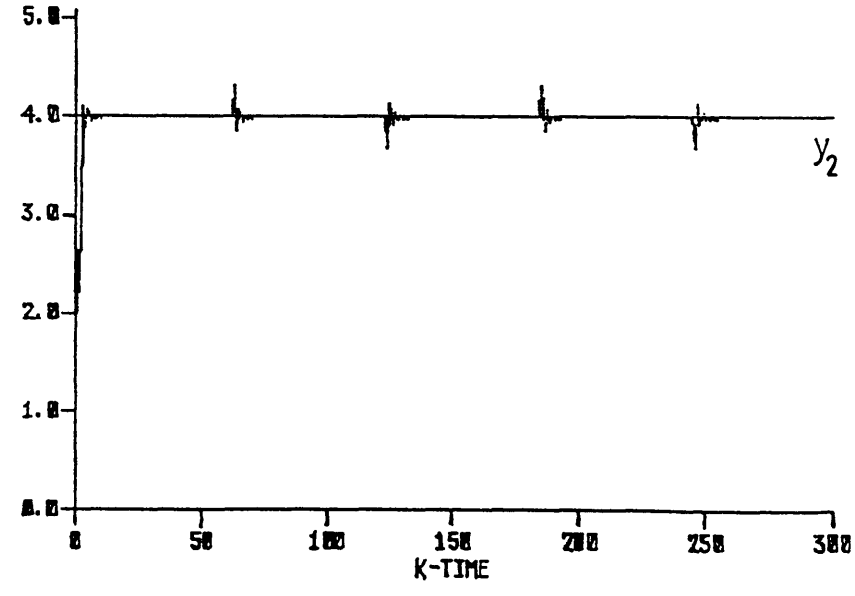


d) Second input of the MIMO system

FIG 5.43: FB/FF NON-ADAPTIVE SIMULATION
T=1,3 Y=.2 model 7

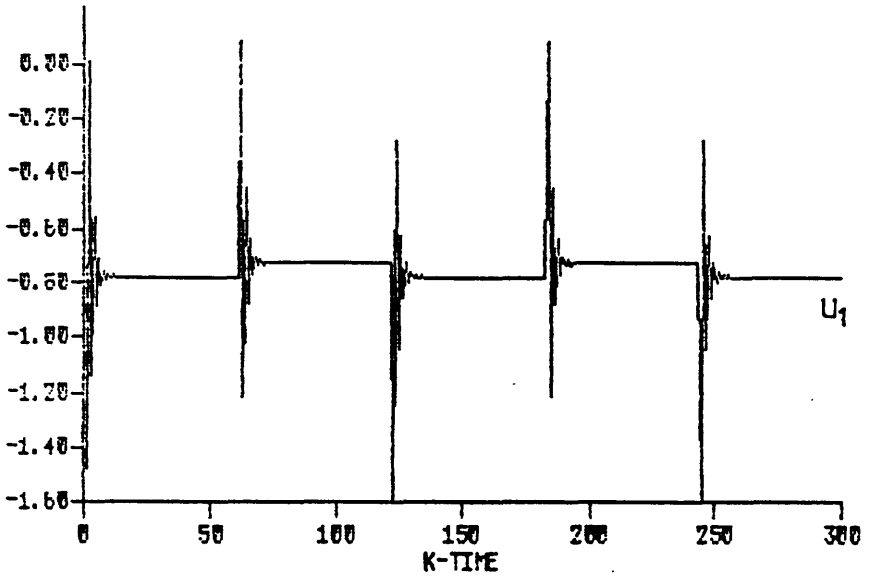


a) First output of the MIMO system

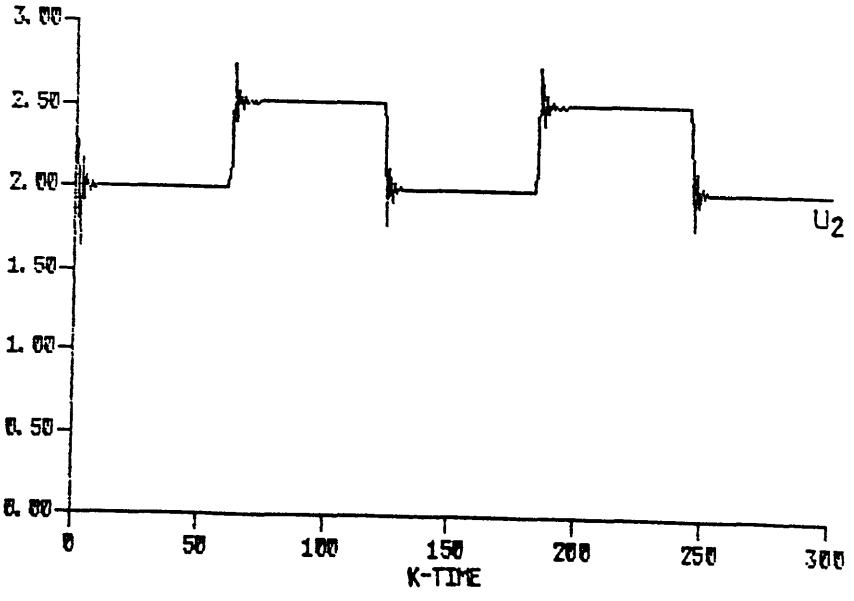


b) Second output of the MIMO system

FIG 5.43: FB/FF NON-ADAPTIVE SIMULATION
T=1,3 Y=.2 model 7



c) First input of the MIMO system



d) Second input of the MIMO system

A minimum variance control criterion was used in the simulation shown in Fig. (5.44). Figures (5.44.a) and (5.44.b), like Figs. (5.41.a) and (5.41.b), show the best achievable regulation. However, in this case, the price paid is also a strong control action [see Figs. (5.44.c) and (5.44.d)].

The control can be severely detuned by increasing the time-horizon. This is shown in Figs. (5.45.c) and (5.45.d). However, the first output [Fig. (5.45.a)] presents large deviations from its set point. In this simulation, the deviations suffered by the second output, caused by load disturbances, are masked by the output noise.

Increasing the horizon even more had no appreciable effect on the I/O responses, as is seen in Fig. (5.46). Here, the control-horizon was chosen to be 3. Once the FB/FF control has been detuned by increasing the horizon, the controlled system appears insensitive to further horizon increments.

Adaptive Simulations with Different Structures

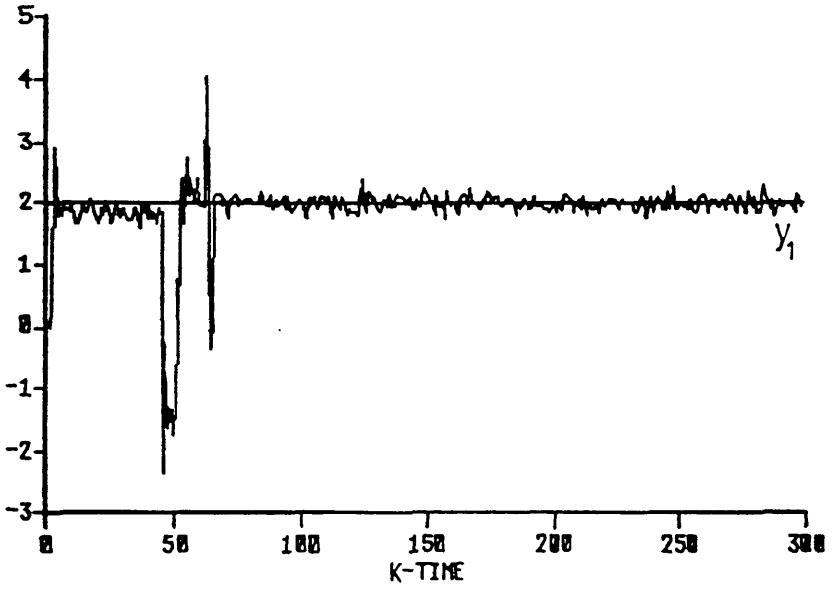
In the following, it is seen how the selection of the identification model structure [given by the observability indices] affects the performance of the controlled system.

The correct model structure [n_1=2 and n_2=1] was employed in the simulation shown in Fig. (5.47). The time-horizons were chosen to be T_1=1 and T_2=3. It can be seen in Figs. (5.47.a) and (5.47.b) that an almost perfect regulation is achieved by a highly demanding control action, especially on the first input [Fig. (5.47.c)]. This behaviour is similar to those of the simulations shown in Figs. (5.41), (5.43) and (5.44).

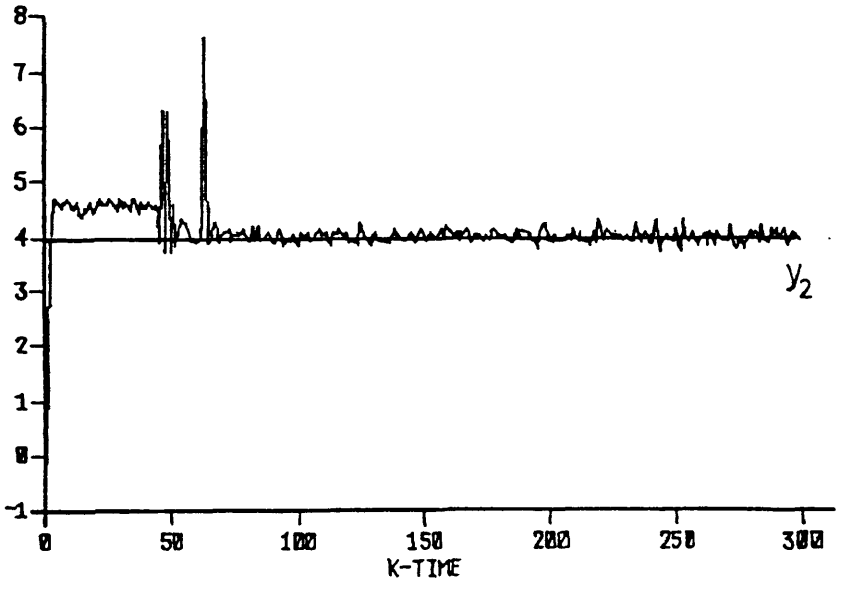
Figure (5.48) shows the results when a wrong model structure is used [n_1=1 and n_2=2]. In this case, the first output [Fig. (5.48.a)] presents large deviations from its set point and the second output response [Fig. (5.48.b)] looks similar to the one shown in Fig. (5.47.b). On the other hand, the control actions [Figs. (5.48.c) and (5.48.d)] appeared detuned, especially the second control input.

The simplest possible structure [n_1=1 and n_2=1] was employed in the simulation shown in Fig. (5.49). The adaptive control converges even in this case; however, as in the simulation of Fig. (5.48), the first output [Fig. (5.49.a)] shows large deviations from its reference value. In this case, the control-actions are highly detuned.

FIG 5.44: DIFFERENT HORIZONS
n=2,1 T=1,1 Σ=2.,2. ϒ=.2 model 7

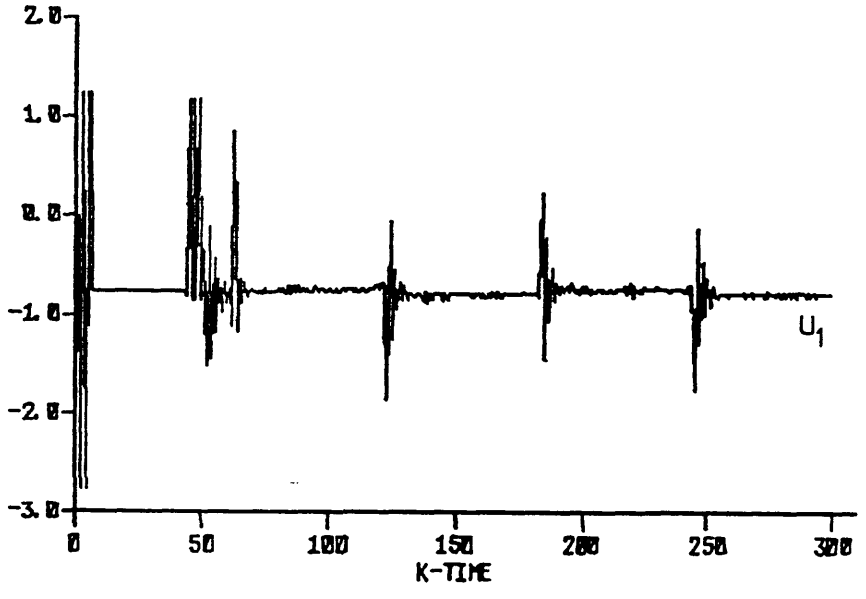


a) First output of the MIMO system

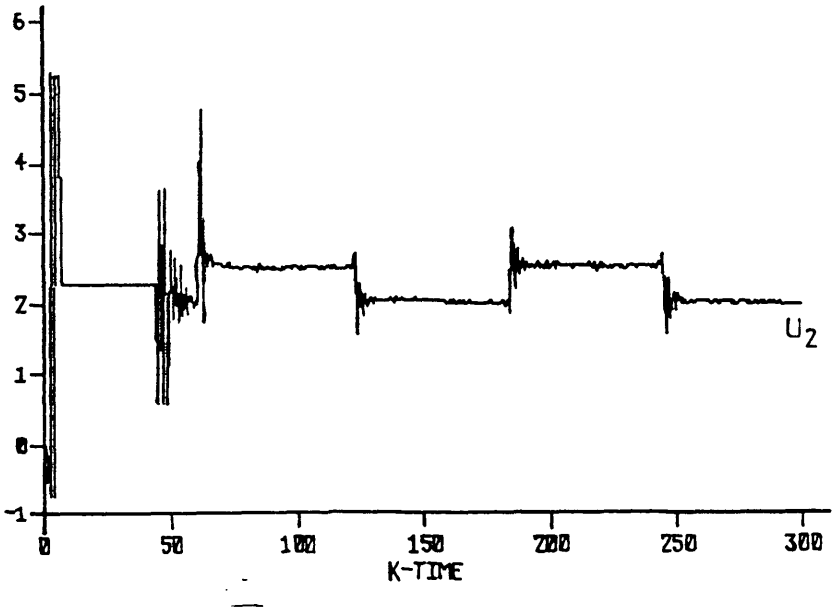


b) Second output of the MIMO system

FIG 5.44: DIFFERENT HORIZONS
n=2,1 T=1,1 Σ=2.,2. γ=.2 model 7

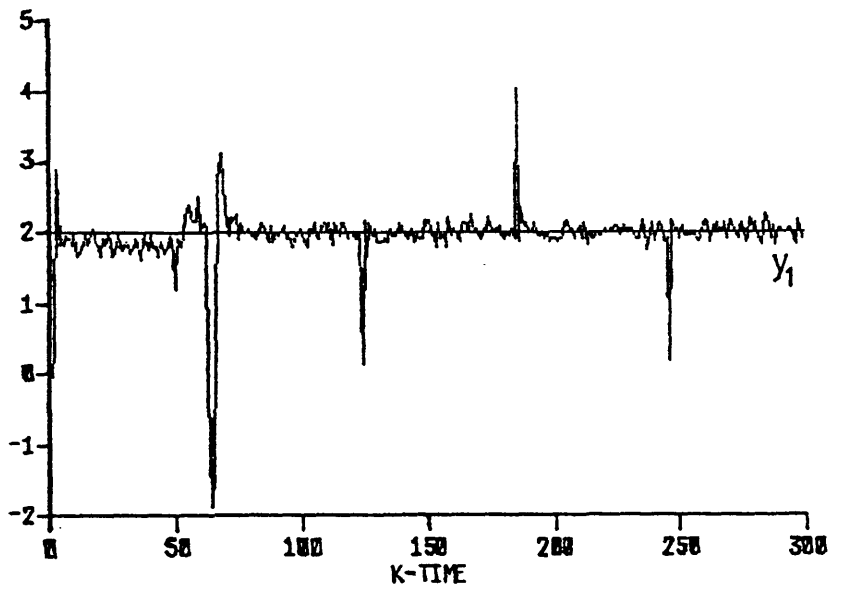


c) First input of the MIMO system

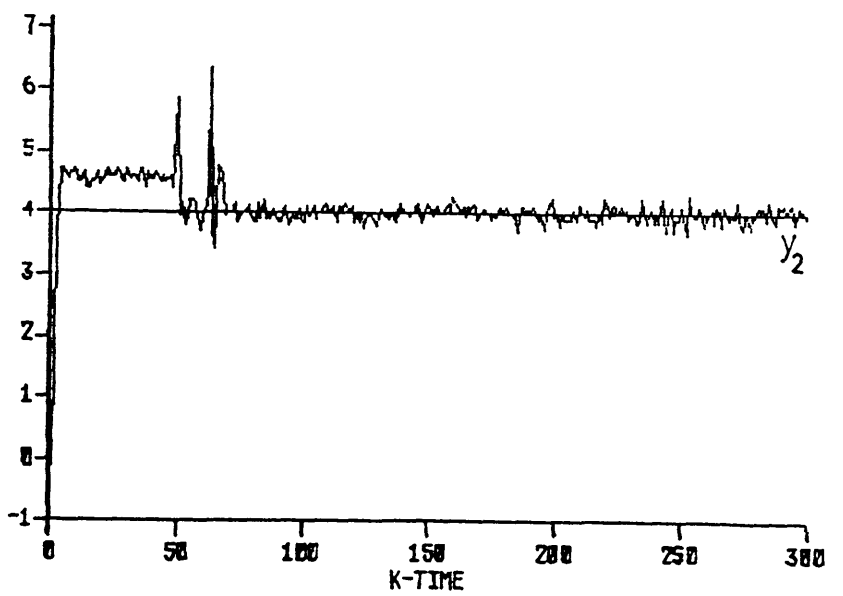


d) Second input of the MIMO system

FIG 5.45: DIFFERENT HORIZONS
n=2,1 T=2,2 Σ=2.,2. ϒ=.2 model 7

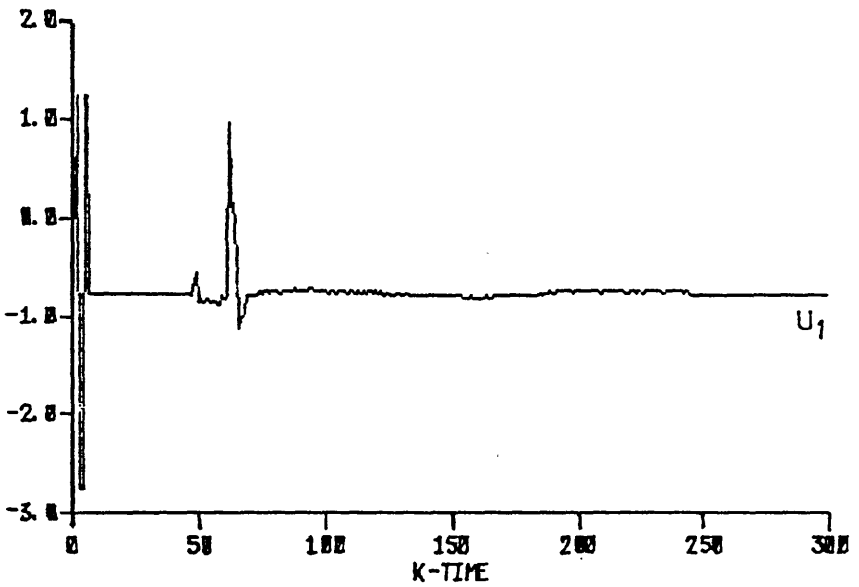


a) First output of the MIMO system



b) Second output of the MIMO system

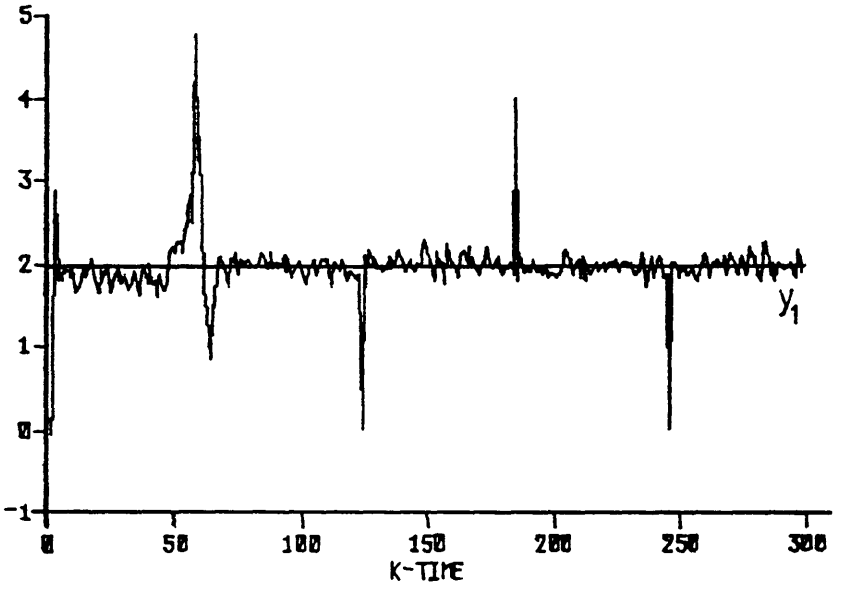
FIG 5.45: DIFFERENT HORIZONS
n=2,1 T=2,2 Σ=2.,2. ϒ=.2 model 7



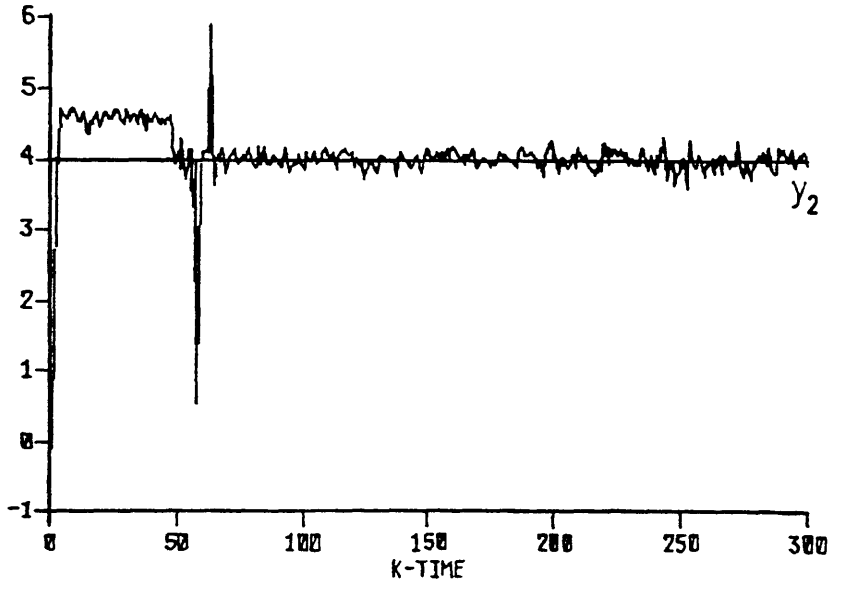
c) First input of the MIMO system

d) Second input of the MIMO system

FIG 5.46: DIFFERENT HORIZONS
n=2,1 T=3,3 Σ=2.,2. Υ=.2 model 7

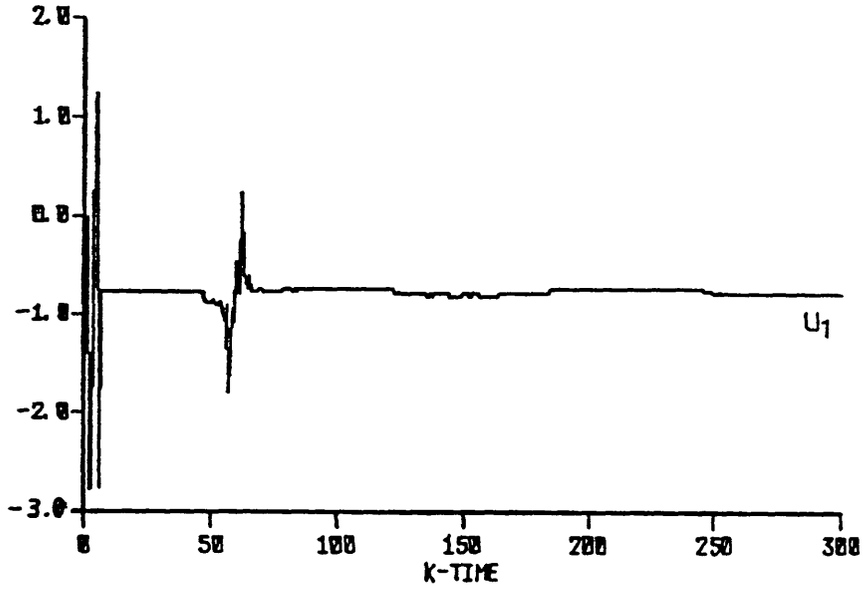


a) First output of the MIMO system

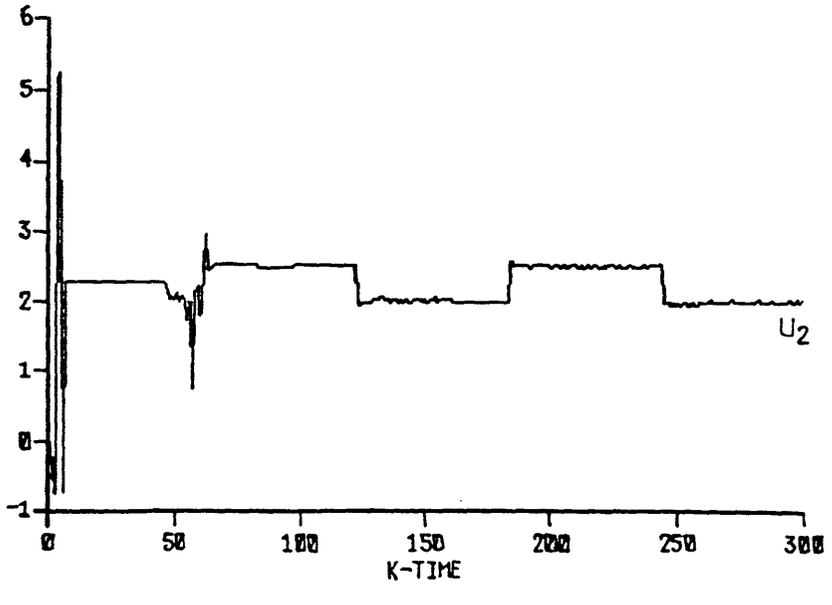


b) Second output of the MIMO system

FIG 5.46: DIFFERENT HORIZONS
n=2,1 T=3,3 Σ=2.,2. ϒ=.2 model 7

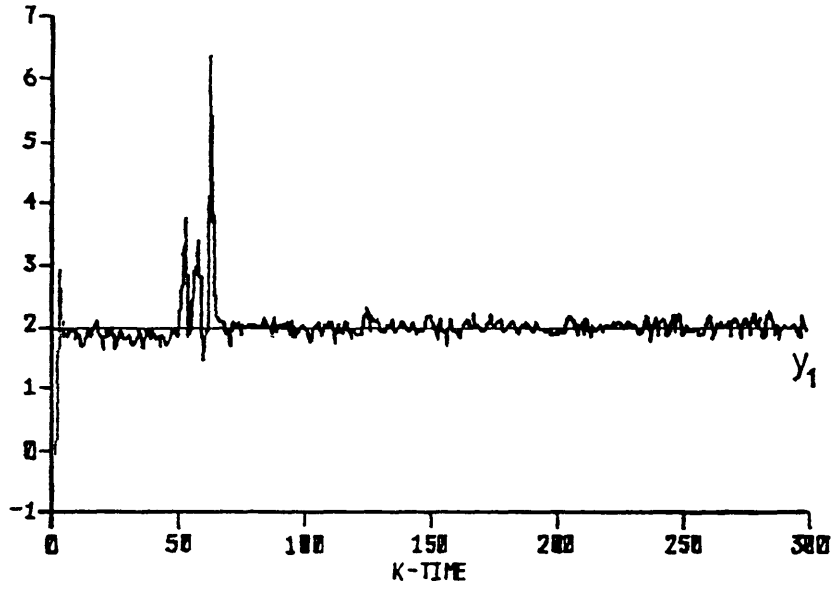


c) First input of the MIMO system

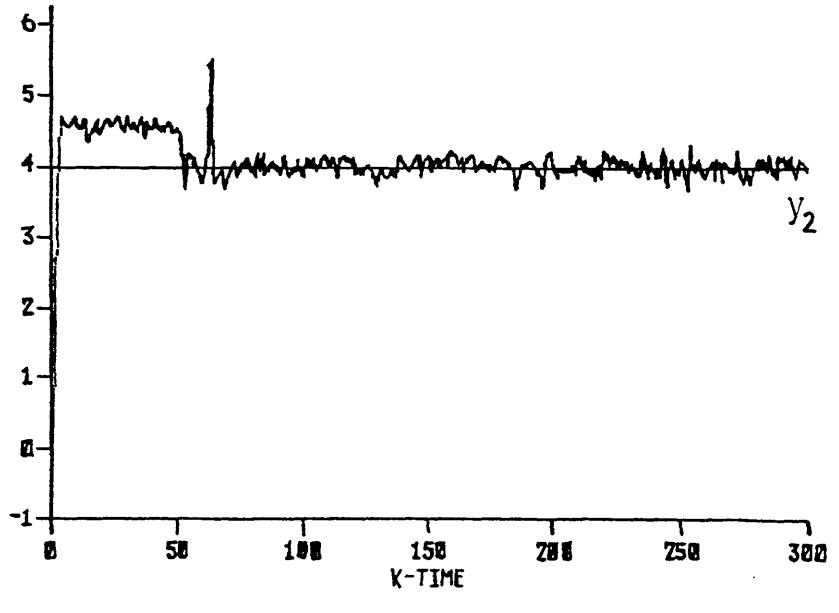


d) Second input of the MIMO system

FIG 5.47: DIFFERENT STRUCTURES
n=2,1 T=1,3 Σ=2.,2. ϒ=.2 model 7

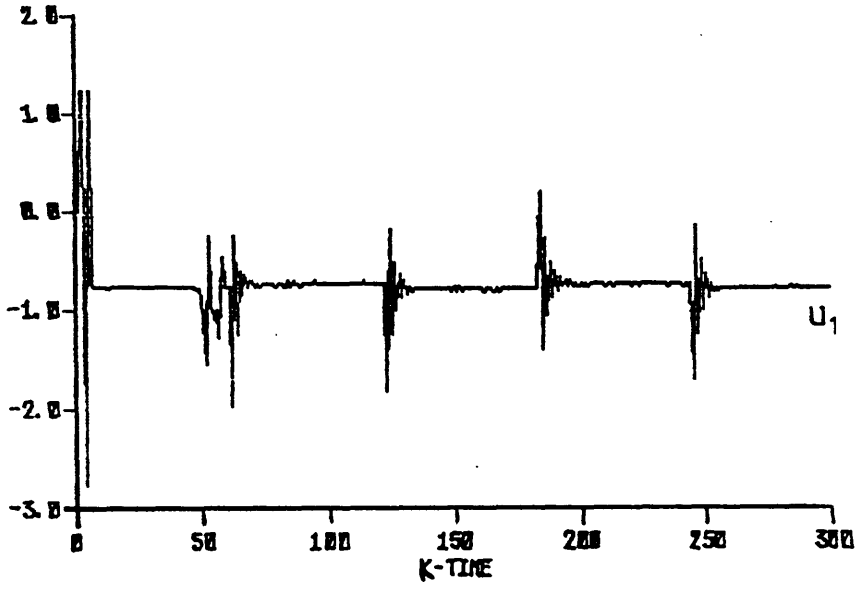


a) First output of the MIMO system

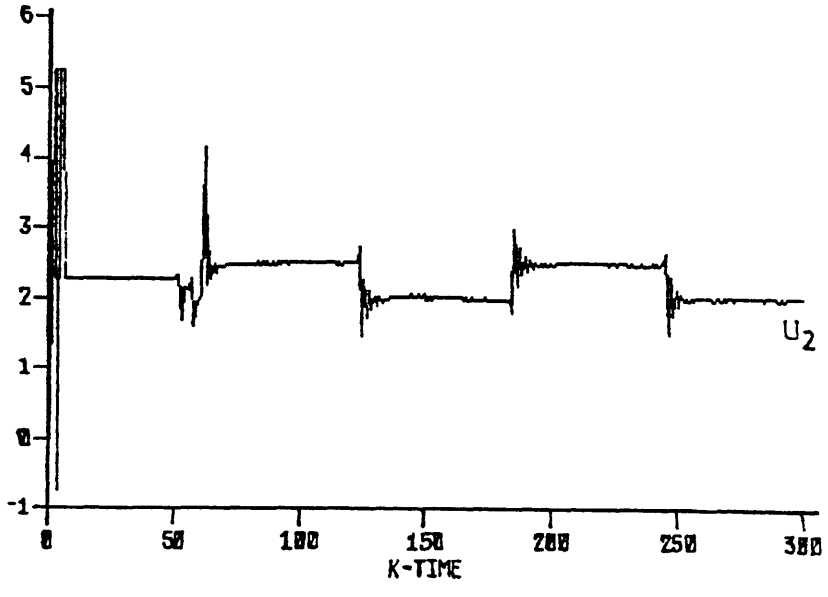


b) Second output of the MIMO system

FIG 5.47: DIFFERENT STRUCTURES
n=2,1 T=1,3 Σ=2.,2. ϒ=.2 model 7

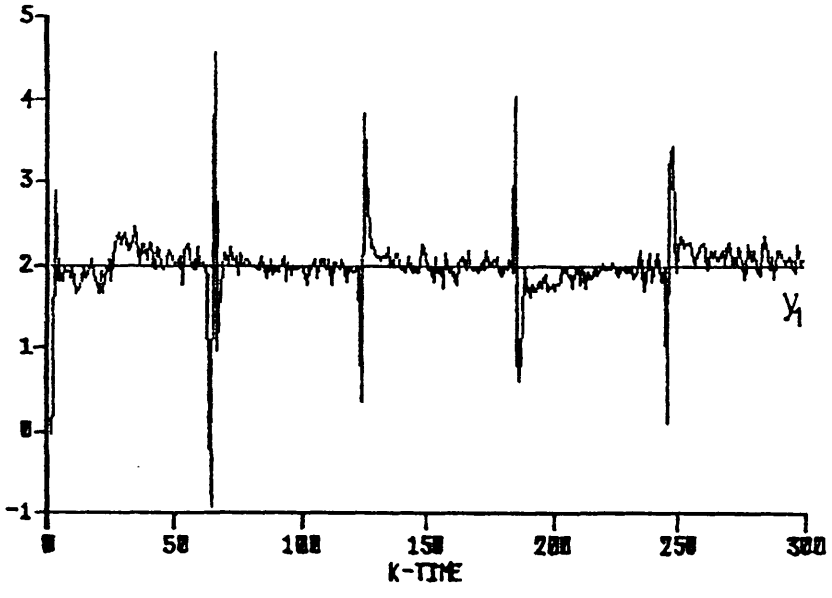


c) First input of the MIMO system

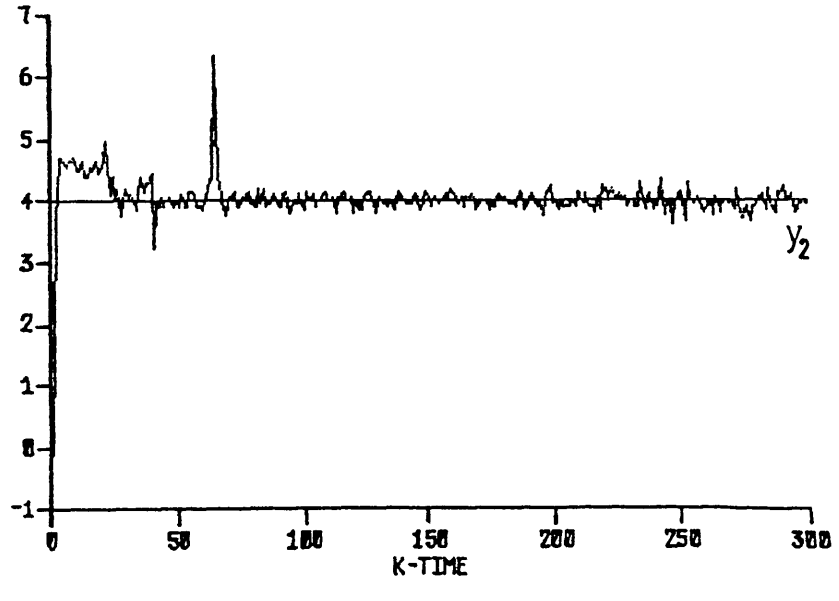


d) Second input of the MIMO system

FIG 5.48: DIFFERENT STRUCTURES
n=1,2 T=1,3 Σ=2.,2. ϒ=.2 model 7

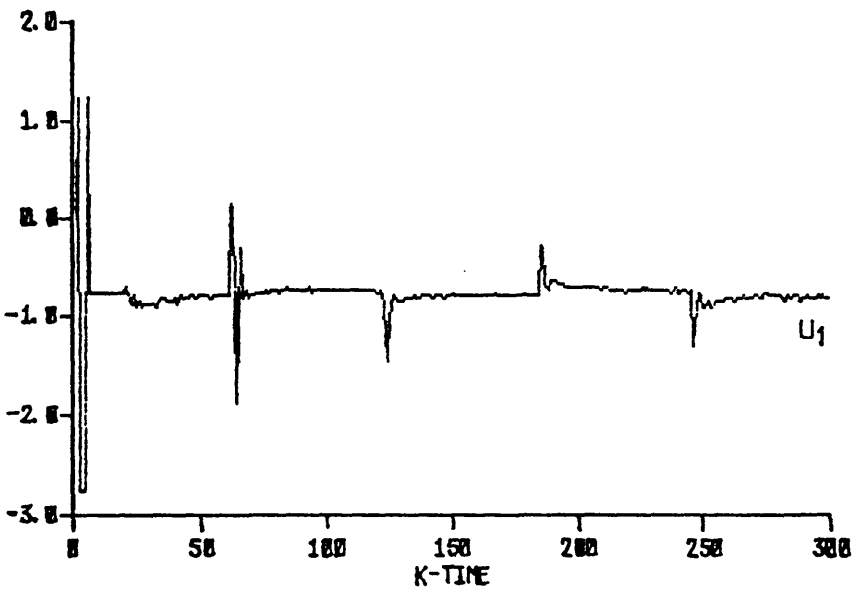


a) First output of the MIMO system

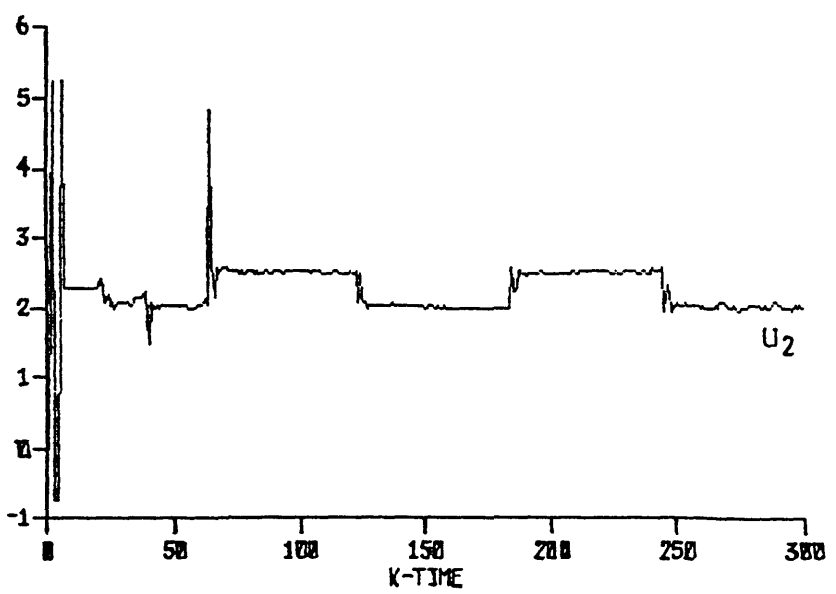


b) Second output of the MIMO system

FIG 5.48: DIFFERENT STRUCTURES
n=1,2 T=1,3 Σ=2.,2. ϒ=.2 model 7

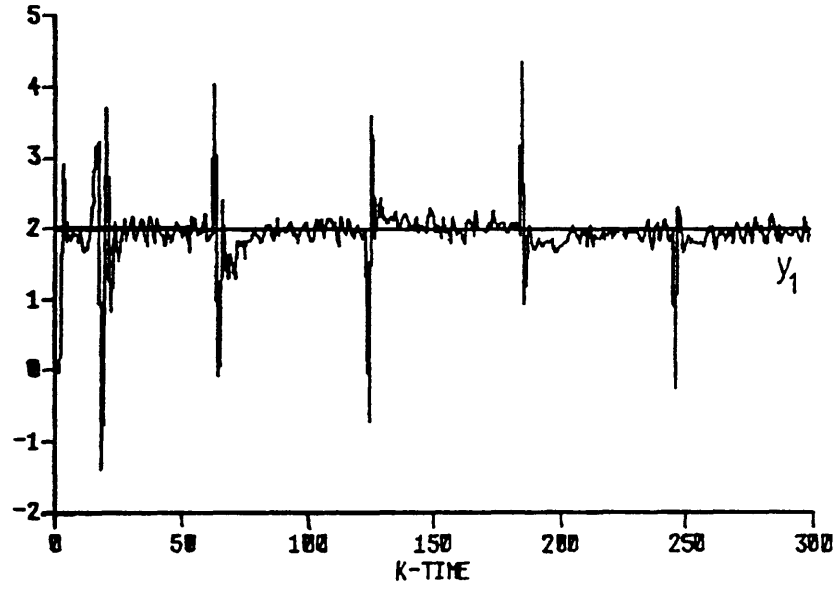


c) First input of the MIMO system

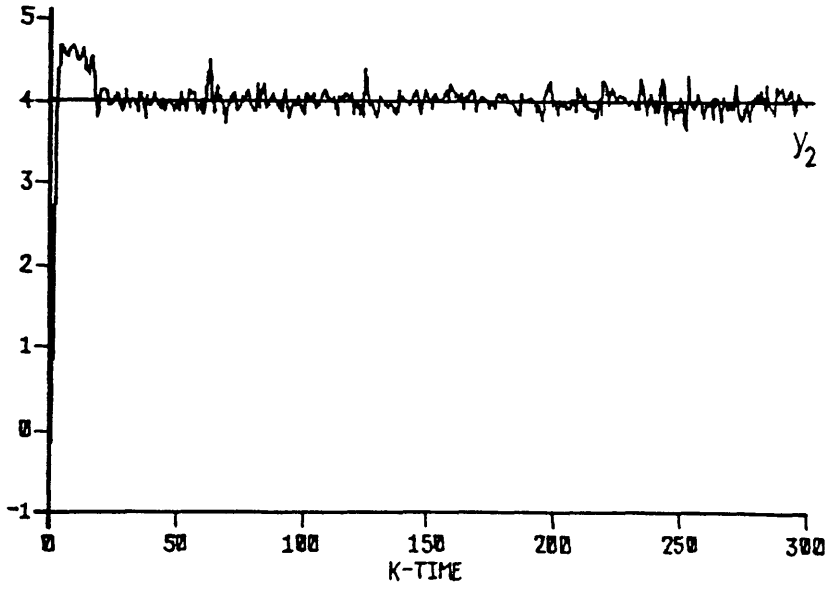


d) Second input of the MIMO system

FIG 5.49: DIFFERENT STRUCTURES
n=1,1 T=1,1 Σ=2.,2. ϒ=.2 model 7

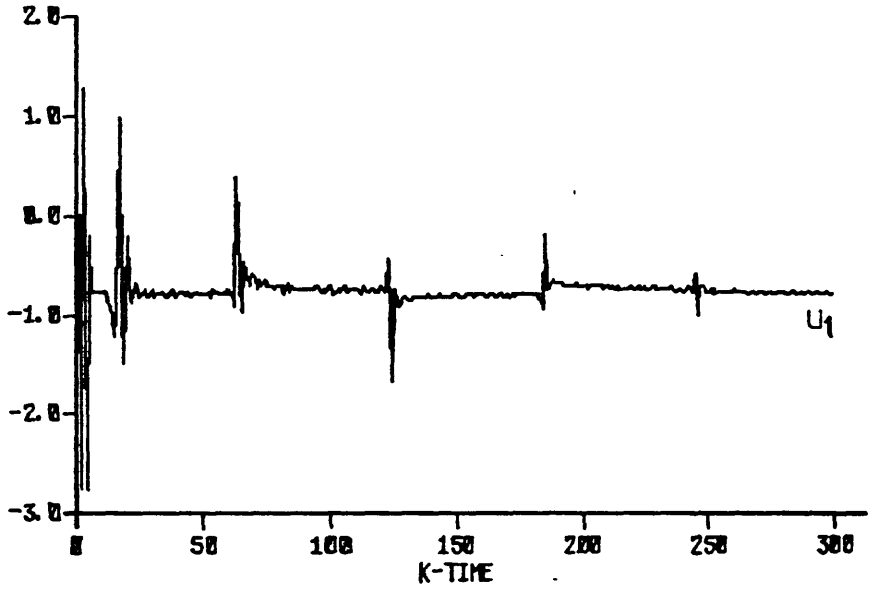


a) First output of the MIMO system

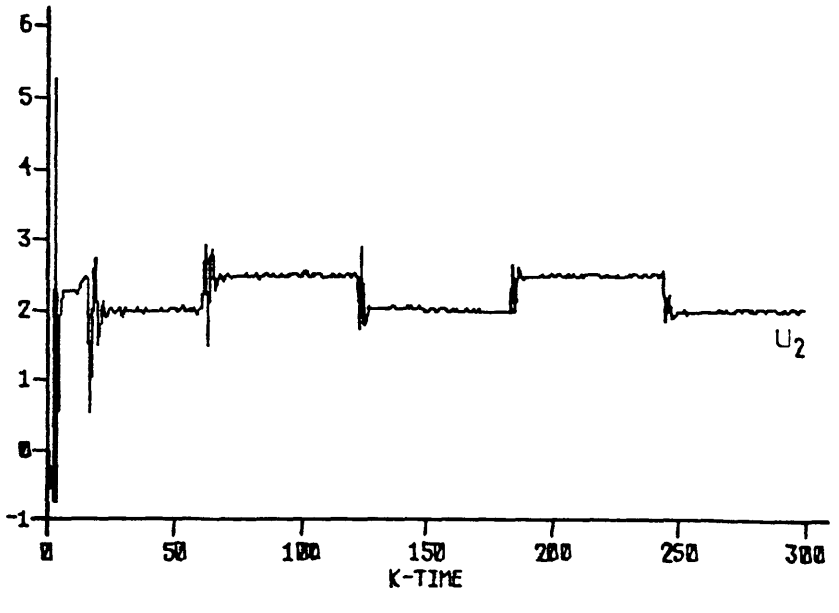


b) Second output of the MIMO system

FIG 5.49: DIFFERENT STRUCTURES
n=1,1 T=1,1 Σ=2.,2. Y=.2 model 7



c) First input of the MIMO system



d) Second input of the MIMO system

In all the figures shown above, the response of the second output y_2 appears quite insensitive to the selection of model structures and time-horizons. This is due to the fact that load changes have little effect on this output [see Figs. (5.42.b) and (5.43.b)]. On the other hand, the response of the first output [which is strongly affected by load changes] looks sensitive to the selection of structural indices [n_1 and n_2] and control-horizons [T_1 and T_2]. In a way, this behaviour was expected since the FB/FF SISO algorithm is also sensitive to model order and control-horizon selection. More work is required in order to fully understand the sensitivity of controller performance to time horizons and structural indices selection. This is especially true in an industrial environment, where the understanding of the processes are, in general, inadequate.

5.2.3.b.- Quasi-nonlinear Simulations

In the following series of quasi-nonlinear simulations, the performance of the FB/FF adaptive control algorithm is compared with a purely FB controller. The simulations basically consists of simultaneous set point changes in both outputs at time k=300; at the same time, the simulated linear model is changed, as it is likely to happen in a non-linear system. The outputs are contaminated with white noise. Both input disturbances are described by the same square wave [amplitude 1, period 120], but with a phase difference of half a period. The RLS algorithm estimates both the process and the disturbance model parameters.

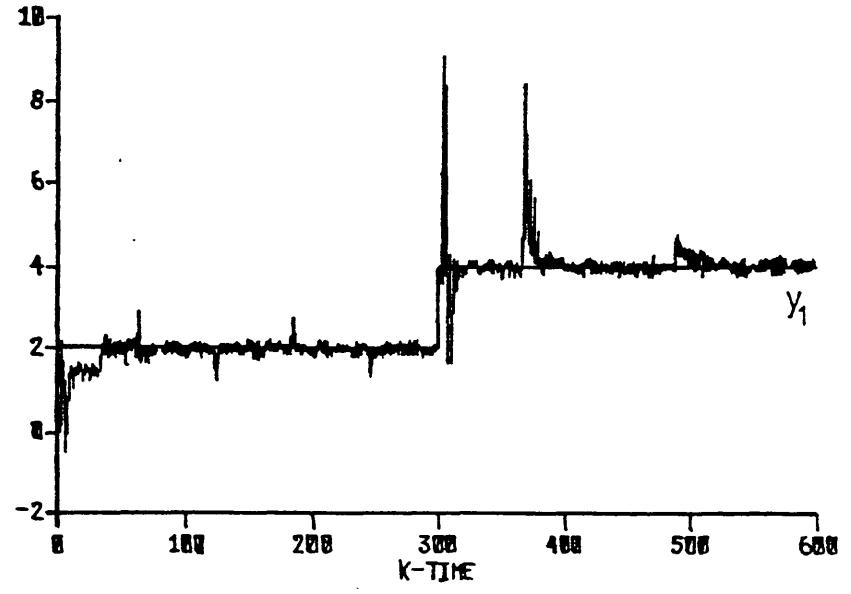
The information lengths were set to Σ_{o1}=1. and Σ_{o2}=2. and the observability indices were chosen to be n_1=2 and n_2=1 [the true model structure].

Figure (5.50) presents the results when the system was controlled with a minimum-variance FB/FF control-law. When the set point changes occur (k=300), the description of the system changes from model 8 to model 7 of Appendix H. A near perfect regulation is achieved. Even after the set points and model have changed, the adaptive control tends to eliminate the effects of the input disturbances. On the other hand, the system controlled with purely feedback action [Fig. (5.51)] presents very large deviations in the steady state.

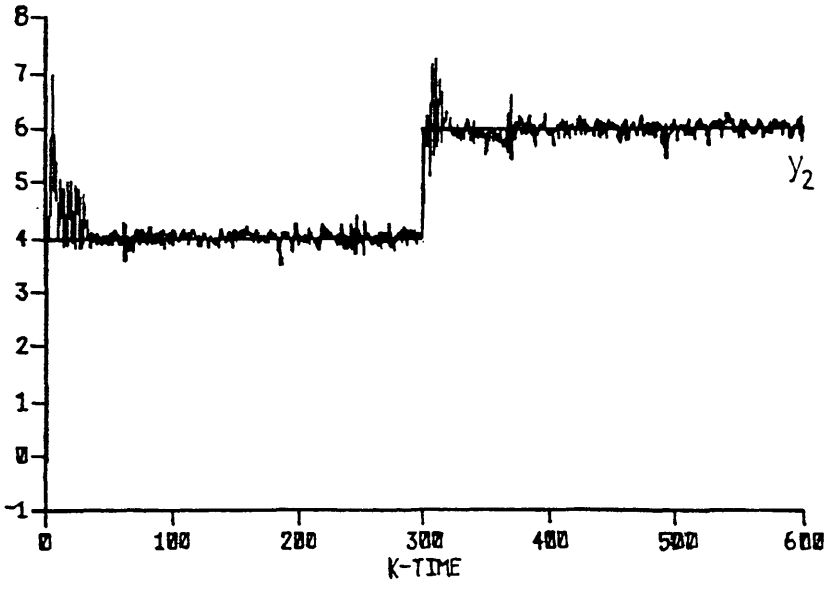
Similar results were observed with different models and different horizons. For example, in the following set of simulations an extended-horizon control strategy is used.

In the simulations shown in Figs. (5.52) and (5.53), the system is firstly described by model 8 and after the set point changes, it is described by model 9. Model 9 differs from model 7 in just one parameter in the disturbance matrix, however, this difference makes the outputs of model 9 much more sensitive to the measured disturbances. In these simulations, the horizons were set to be 3. Again, the performance of the system controlled with the FB/FF algorithm [Fig. (5.52)] is much better compared with the performance of the system controlled with purely feedback [Fig. (5.53)].

FIG 5.50: FB/FF MIMO SELF-TUNER
n=2,1 T=1,1 Σ=1.,2. γ=.2 models=8,7

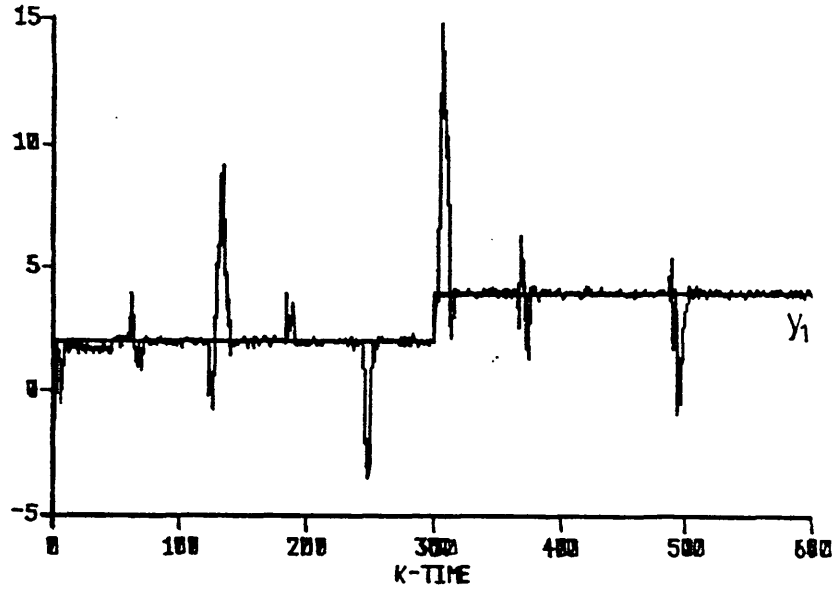


a) First output of the MIMO system

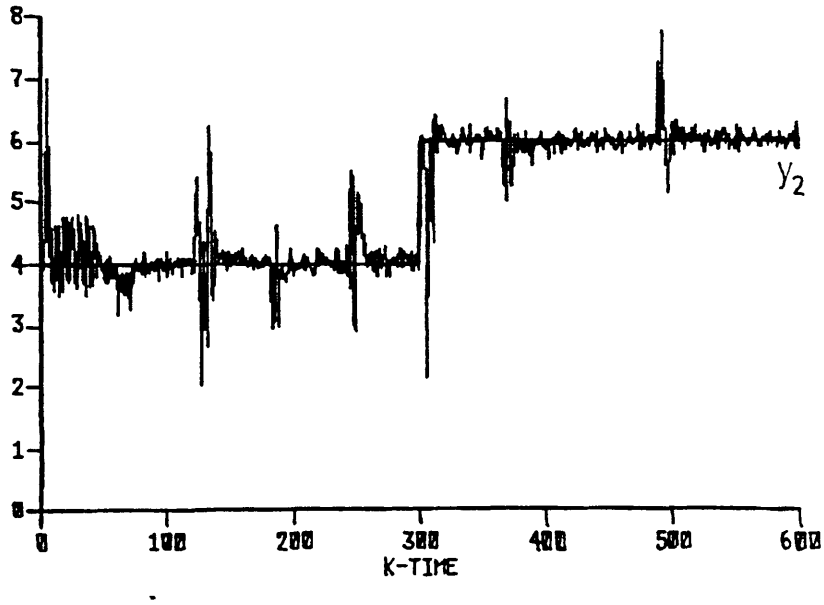


b) Second output of the MIMO system

FIG 5.51: FB MIMO SELF-TUNER
n=2,1 T=1,1 Σ=1.,2. γ=.2 models=8,7

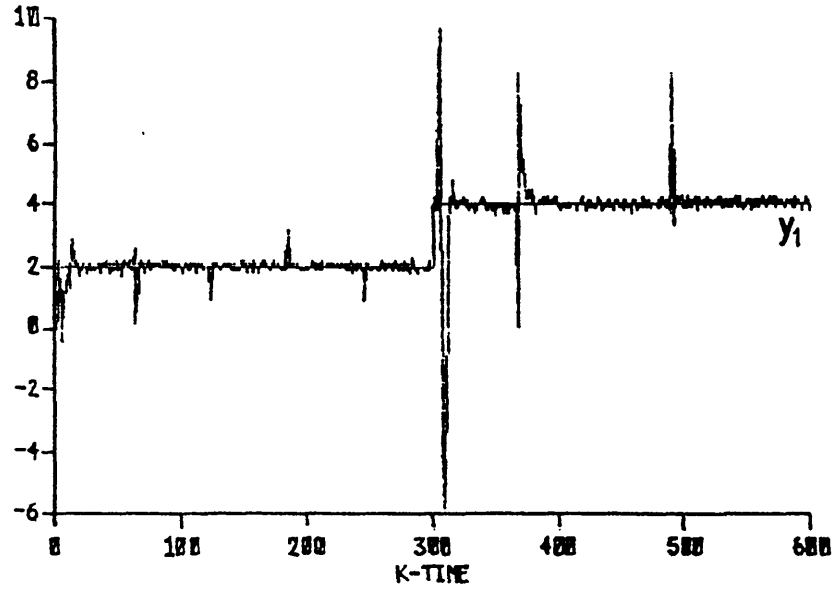


a) First output of the MIMO system

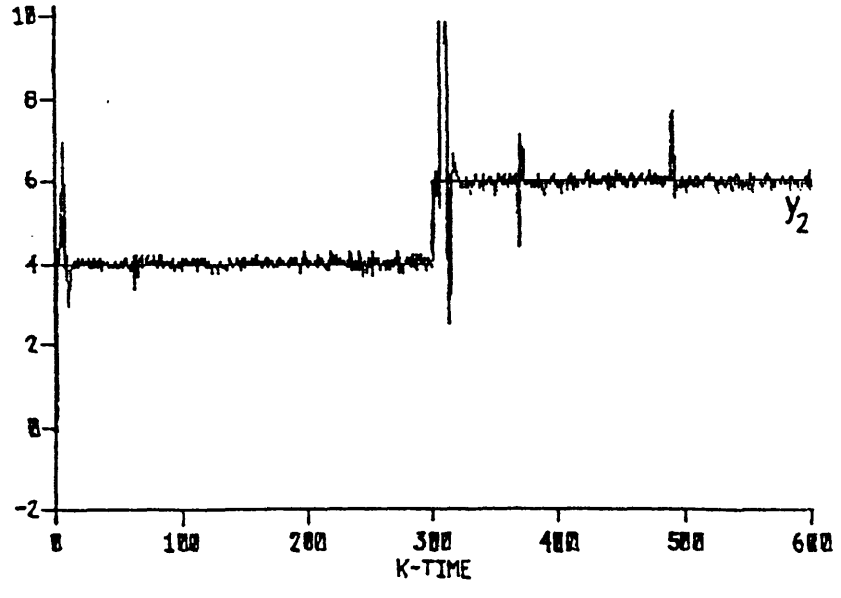


b) Second output of the MIMO system

FIG 5.52: FB/FF MIMO SELF-TUNER
n=2,1 T=3,3 Σ=1.,2. γ=.2 models=8,9

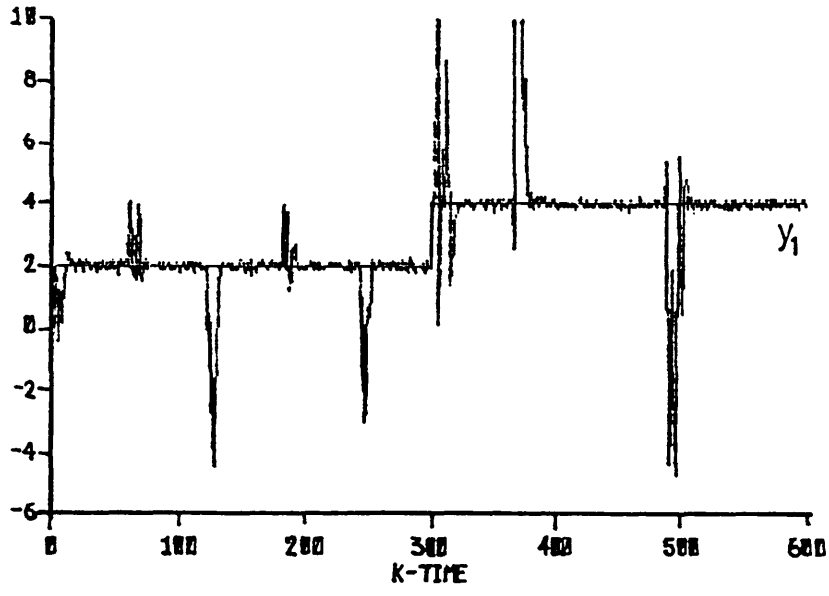


a) First output of the MIMO system

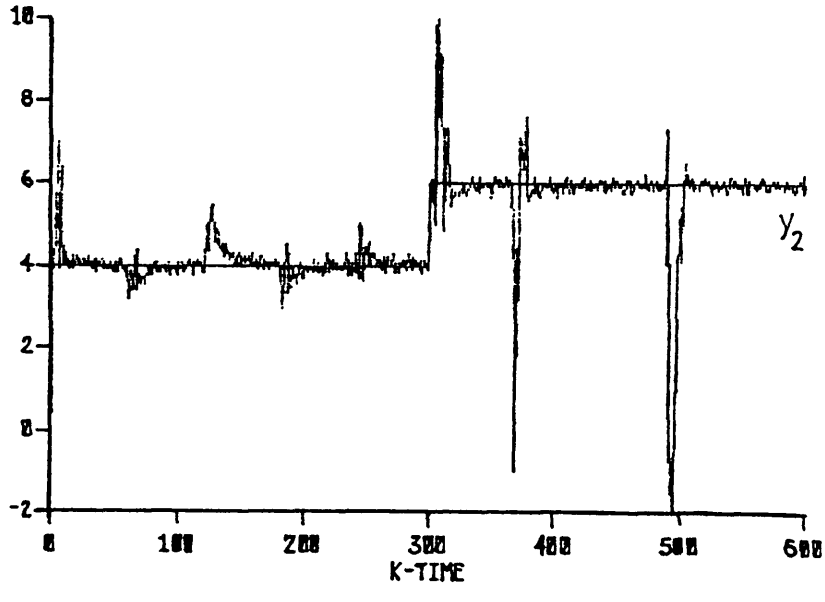


b) Second output of the MIMO system

FIG 5.53: FB MIMO SELF-TUNER
n=2,1 T=3,3 Σ=1.,2. γ=.2 models=8,9



a) First output of the MIMO system



b) Second output of the MIMO system

Figures (5.54) and (5.55) show the results obtained when different horizons were employed. In these simulations, the control-horizons were chosen to be 3 for the first output and 1 for the second. The system controlled with FB/FF presents a much better response than the one controlled with purely FB. It is worth noticing that changing the horizon had almost no effect in the system controlled with FB/FF, as it can be seen by comparing Figs. (5.52) and (5.54). This agrees with the observation mentioned earlier that once the control is detuned [for example Fig. (5.54)] the controlled system becomes insensitive to further horizon increments [as in Fig. (5.52)].

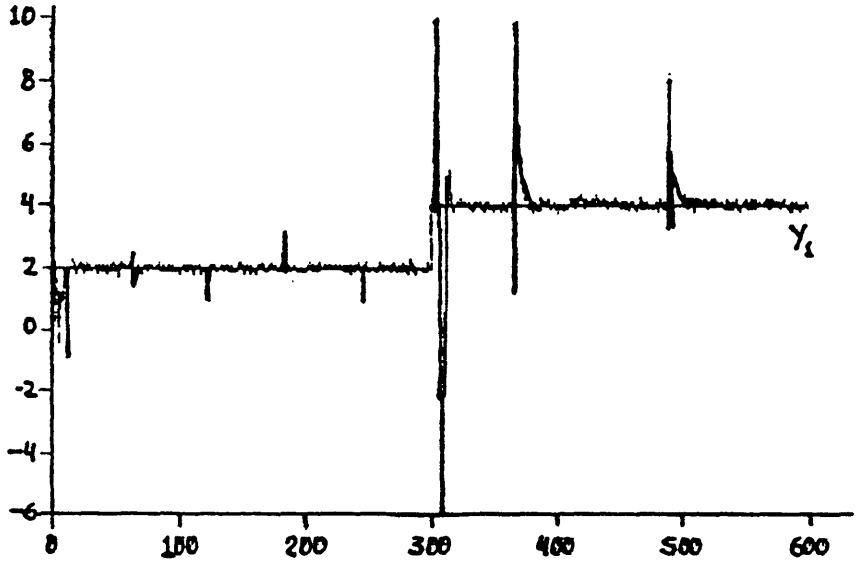
In short, it can be said that a system controlled with FB/FF control action always presents a better performance compared with the one that can be achieved by a system controlled with feedback only.

The FB/FF adaptive algorithm tested in this section requires the same amount of a priori information as does the purely FB adaptive algorithm. The only price which must be paid is that the rate of adaptation may be slower given the extra parameters which need to be estimated; and, of course, the FB/FF algorithm is only usable when the disturbances are measurable.

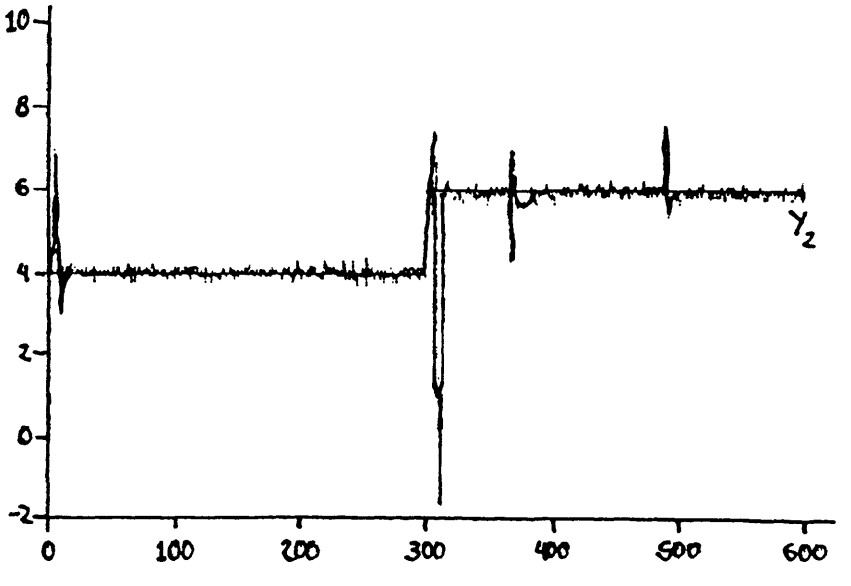
The FB/FF adaptive controller used in this section is a positional algorithm, so that it is expected to be sensitive to unmeasured deterministic disturbances. However, an incremental algorithm can easily be conceived to cope with this problem.

Many more tests (simulations and experiments) are required, nevertheless, the results obtained so far are very promising.

FIG 5.54: FB/FF MIMO SELF-TUNER
n=2,1 T=3,1 Σ=1.,2. γ=.2 models=8,9

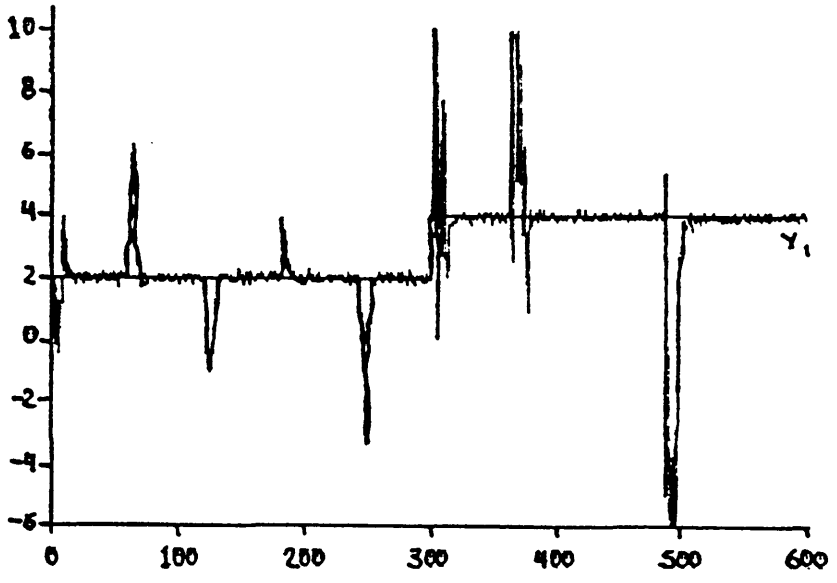


a) First output of the MIMO system

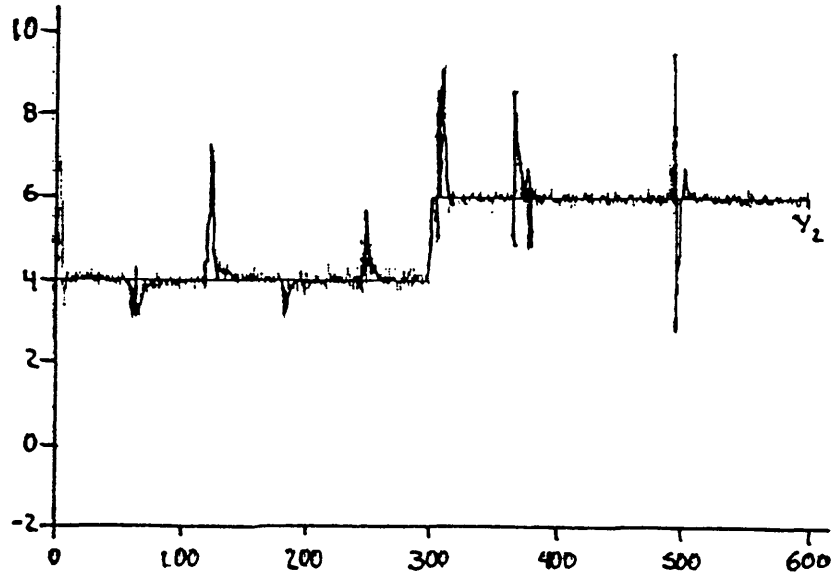


b) Second output of the MIMO system

FIG 5.55: FB MIMO SELF-TUNER
n=2,1 T=3,1 Σ=1.,2. γ=.2 models=8,9



a) First output of the MIMO system



b) Second output of the MIMO system

CHAPTER 6 CONCLUSIONS

Almost thirty years ago, the first self-tuning algorithm was proposed by Kalman. The idea was very advanced for the time, considering the state of development of both control theory and computer technology. Kalman's algorithm generated a great deal of scepticism, this can be illustrated by the following comment to his original work :

"*But I ask last the primary unanswered question: Does it work ?*"*

Nowadays, it is believed that both control theory and computer technology are sufficiently developed so that self-tuners can feasibly implemented in an industrial environment. However, the point of view in industry is still characterized by scepticism.

To make self-tuners widely acceptable in industry, the algorithms must be of simple design and operation, but robust enough to be applied with confidence in a variety of processes.

The robust self-tuning regulator of Ydstie complies with the above requirements. The algorithm, which was designed to control processes usually found in the chemical industry, includes two important features: a variable forgetting factor for parameter estimation, which controls the rate of adaptation; and an extended-horizon control law, which detunes the control action to cope with non-minimum phase systems and time-delays.

It was later detected that Ydstie's self-tuner was sensitive to large and frequent load changes and to interactive control-loops.

The present work shows how to solve these problems without abandoning Ydstie's basic design approach. Feedback SISO, FB/FF SISO, FB MIMO and FB/FF MIMO self-tuning algorithms are proposed and tested with systems characterized by large and frequent load changes, time-delays and control interaction.

* [Kalman, 1958]

The present work shows that in the case of SISO FB control, an incremental algorithm without bias estimation using an equal control-increments control strategy is robust against unmeasured deterministic disturbances. The algorithm was implemented using a direct (implicit) approach; however, preliminary results suggested that an indirect (explicit) approach may lead to an even more robust algorithm.

A SISO FB/FF self-tuner was also implemented in a direct-incremental form. The controller includes direct transmission between the measured disturbance and the measured output to account for inter-sample perturbations. The control law considers disturbances as changing by steps, however other kind of disturbances may also be considered. The FB/FF self-tuner has been shown to be robust to initial parameters selection. It has also shown a much better performance than a FB only algorithm at little extra cost, namely, a slower convergence rate due to the increased number of parameters which need to be estimated.

In the FB multivariable case, a state-space formulation was employed. The main advantage is the reduced number of parameters needed for estimation, compared with a matrix transfer function approach for multivariable systems. A Bootstrap method was used to solve the problem of simultaneous estimation of both states and parameters. The MIMO self-tuning algorithm has shown itself to be more robust than independent SISO regulators when controlling interactive systems, but it can also control non-interactive systems. An incremental implementation of the MIMO controller appears to be robust against unmeasured deterministic disturbances.

A FB/FF state-space multivariable controller can be employed in systems where disturbances are measurable. A much better performance can thus be obtained and the only price that must be paid is a slower rate of adaptation.

Like any other self-tuning algorithm, the controllers proposed in this work require the selection of a set of initial parameters. In the following, a guide to the selection of the most important of these parameters is given:

a) Time-horizon [T]: The selection of this parameter is easier in the SISO FB case. The chosen T must be larger than or equal to the largest expected delay in the plant. If the plant is non-minimum phase or the control action is undesirably strong, the control can be detuned by increasing the horizon T.

In the FB MIMO case, the time-horizon selection is less easy. Here, an horizon must be chosen for each output. If the plant is linear and time invariant, the MIMO self-tuner is very robust to time-horizons selection. On the other hand, in time varying or non-linear plants the algorithm is not very robust to the selection of different horizons for different outputs. Nevertheless, it is advisable to use different horizons for each output when the plant is characterized by a general delay structure.

When FB/FF controllers are employed [both SISO and MIMO], perfect regulation can be achieved, but undesirably strong control action may be required. In this case, the horizon T can be increased to detune the control. It has been observed that once the control has been detuned, increasing the horizon even more has little effect on the control performance.

b) Model orders [n, q] and observability indices [n_j]: In the SISO FB case, the selection of the model order n appears to be very robust. Care must only be taken in systems affected by a time-delay. In this case it is convenient to utilize values of n larger than two.

The SISO FB/FF controller requires the selection of two model parameters. One for the control/output transfer function [n], another for the disturbance polynomial [q]. If the plant has no time-delay, the selection of n and q will appear straightforward an robust. On the other hand, if the disturbance is measured with a delay, the selected parameter q must be large enough to ensure that significant information is included in the data vector of the parameter estimator. The value of q must be larger than dw-T, where dw is the disturbance delay.

In the multivariable case, one observability index [structural index] n_j must be chosen for each output y_j. The observability indices define the structure of the state-space model to be identified. Different model structures can be defined for the same overall system order. It is difficult to know the true model structure of processes in an industrial plant. However, the results obtained so far indicate that the MIMO state-space algorithm is robust against the selection of model structure, except when very large horizons are used. In the case of FB/FF control, choosing a wrong model structure usually leads to a detuned control action and consequently, perfect control is not achievable.

c) Information length or content [Σ_0]: This parameter is

defined as the product of the process noise variance and the number of sample intervals contained in the data window. However, Σ_0 is usually considered as just a tuning parameter which controls the rate of adaptation. It has been observed that too small a value produces undesirable "burst" effects on the estimates and the control. On the other hand, too large a value makes the estimator insensitive so that the algorithm cannot adapt to plant changes. Nevertheless, the selection of the information length is very robust, namely it can be safely selected in a range of two orders of magnitude.

d) State filter gain [γ]: This parameter, which is only used by the state-space algorithms, controls the rate of adaptation of the state estimates. The MIMO self-tuner has shown itself to be very robust to the selection of γ. Problems only arise if large values are employed [$\gamma > .9$], which means that little weight is given to the state predicted by the estimated model and too much weight is given to the measurement. In this case, the algorithm becomes extremely sensitive.

The present work has left the way open for more tests and developments. In the SISO case, for example, more experimental work would be done with indirect [explicit] algorithms. It may also be interesting to implement the algorithm on various kinds of processes.

In the MIMO case, different types of state filters [as mentioned in Chapter 3] can be compared. The MIMO algorithms must also be implemented on real-time interactive processes.

An incremental version of the FB/FF state-space controller can be developed and implemented on real-time processes.

Based on the experience obtained in the present work, it can be said that implementation of self-tuning algorithms in an industrial environment is both feasible and convenient. Nevertheless, the design of robust enough algorithms can only be attained by thorough tests on real-time processes. At this stage, the role of the engineer appears crucial.

In short, we can join in Kalman's answer to the early self-tuning control critics:

"*the author wishes to answer his last and most important point, 'Does it work ?'. The answer is, Yes.*" [Kalman, 1958].

REFERENCES

Allidina, A. Y., Hughes, F. M., Tye, C., Self-tuning Control for Systems Employing Feedforward, IEE proc., Pt. D, vol. 128, No. 6, 283-291, November 1981.

Aström, K. J., B. Wittenmark, ON SELF-TUNING REGULATORS, Automatica, vol. 9, 185-199, 1973.

Bayoumi, M. M., Wong, K. Y., El-Bagoury, M. A., A SELF-TUNING REGULATOR FOR MULTIVARIABLE SYSTEMS, Automatica, vol. 17, no. 4, 575-592, 1981.

Bertin, D., Bittani, S., Bolzen, P., Tracking of Nonstationary Systems by Means of Different Prediction-error Directional Forgetting Techniques, IFAC Workshop on Adaptive Systems in Control and Signal Processing, Lund, July 1986.

Borison, U., Wittenmark, B., An Industrial Application of a Self-tuning Regulator, IFAC Symposium on Digital Computer Applications to Process Control, Zurich, 1974.

Borison U., Self-tuning Regulators - Industrial Application and Multivariable Theory, Report 7513, Dept. Automatic Control, Lund Institute of Technology, Lund, 1975.

Borison, U., Syding, R., Self-tuning Control of an Ore Crusher, Automatica, vol. 12, 1-7, 1976.

Borison, U., SELF-TUNING REGULATORS FOR A CLASS OF MULTIVARIABLE SYSTEMS, Automatica, vol. 15, 209-215, 1979.

Bristol, E. H., ON A NEW MEASURE OF INTERACTION FOR MULTIVARIABLE PROCESS CONTROL, IEEE Transactions on Automatic Control, 133-134, January 1966.

Buchholt, F., Kummel, M., A MULTIVARIABLE SELF-TUNING TO CONTROL A DOUBLE EFFECT EVAPORATOR, Automatica, vol. 17, no. 5, 737-743, 1981.

Cegrell, T., Hedqvist, T., Successful Adaptive Control of Paper Machines, Automatica, vol. 11, 53-59, 1975.

Clarke, D.W., Gawthrop, P.J., SELF-TUNING CONTROLLER, IEE Proc., Control & Science, vol.122, No.9, 929-934, September 1975.

Clarke, D. W., Hodgson, A. J., Tuffs, P. S., Offset Problem and K-incremental Predictors in Self-tuning Control, IEE Proc., Pt. D, vol. 130, No. 5, 217-225, September 1983.

Clarke, D. W., Introduction to Self-tuning Controllers, Chapter 2, Self-tuning and Adaptive Control: Theory and Applications, IEE Control Eng. Series 15, Peter Peregrinus ltd., London, 1985.

Clarke, D. W., Generalized Predictive Control, IEE Colloquium on Advances in Adaptive Control, Digest No. 1986/58, 2/1-2/5, April 1986.

Dugard, L., Goodwin, G. C., Xianya, X., The Role of the Interactor Matrix in Multivariable Stochastic Adaptive Control, Automatica, vol. 20, 701, 1984.

Dumont, G. A., SELF-TUNING CONTROL OF A CHIP REFINER MOTOR LOAD, Automatica, vol.18, No. 3, 307-314, 1982.

Dvoretzky, A., On Stochastic Approximation, Proc. 3rd. Symposium Math. Stat. Prob. 1, University of California Press, 39-55, Berkeley, Calif., 1956.

El-Sherief, H., Sinha, N. K., Bootstrap Estimation of Parameters and States of Linear Multivariable Systems, IEE Transactions on Automatic Control, Vol. AC-24, No. 2, 340-342, April 1979.

El-Sherief, H., Sinha, N. K., Suboptimal Control of Linear Stochastic Multivariable Systems with Unknown Parameters, Automatica, vol. 18, No. 1, 101-105, 1982.

Elliot, H., Wolovich, W. A., Parameterization Issues in Multivariable Adaptive Control, Automatica, vol. 20, No. 5,533-534, 1984.

Elliot, H., Wolovich, W. A., A Parameter Adaptive Control Structure for Linear Systems, IEEE Transactions on Automatic Control, Vol. AC-27, No. 2, 340-352, April 1982.

Fortescue, T. R., Work on Åström's Self-tuning Regulator, Report, Dept. of Chem. Eng., Imperial College, London, 1977.

Fortescue, T. R., Kershenbaum, L. S., Ydstie, B. E., Implementation of Self-tuning Regulators with Variable Forgetting Factors, Automatica, vol. 17, No. 6, 831-835, 1981.

Gawthrop, P. J., A CONTINUOUS-TIME APPROACH TO DISCRETE-TIME SELF-TUNING CONTROL, Optim. Control Applications & Methods, vol.3, 399-414, October-December 1982.

Goodwin, G. C., Ramadge, P. J., Caines, P. C., Discrete-time Multivariable Adaptive Control, IEEE Transactions on Automatic Control, vol. AC-25, No3, 449-456, June 1980.

Goodwin, G. C., Sin, K. S., Adaptive Filtering Prediction and Control, Prentice-Hall Information and System Science Series, Englewood Cliffs, 1984.

Grimble, M. J., Implicit and Explicit LQG Self-tuning Controllers, Automatica, vol. 20, No. 5, 661-669, 1984.

Guidorzi, R. P., Complete Sets of Invariants and Canonical Input-Output Forms for Multivariable Systems Structural and Parametric Identification, IFAC symposium on Identification and Systems Parameter Estimation, 429-436, Darmstadt, 1979.

Hiram, Y., Kershenbaum, L., Simulation Studies of Extended Horizon Self-tuning Regulators, Report, Dept. of Chem. Eng., Imperial College, London, October 1983.

Hiram, Y., Kershenbaum, L., Overcoming Difficulties in the Application of Self-tuning Controllers, American Control Conference, Boston, 1985.

Hiram, Y., Kershenbaum, L., Perez, R., Implementation Problems with Adaptive Controllers, IEE Colloquium on Advances in Adaptive Control, Digest No. 1986/58, 5/1-5/4, London, April 1986.

IBM, ACS Engineers Reference Manual, pub. No. SH20-SS10, May 1984.

Irwin, G. W., Roberts, A. P., The Luenberger Canonical form in the State-Parameter Estimation of Linear Systems, Int. J. Control, vol. 23, No. 6, 851-864, 1976.

Isermann, R., Parameter Adaptive Control Systems - A Review on Methods and Application, 6th. IFAC Symposium on Identification and Systems Parameter Estimation, 373-378, York, 1985.

Jacobs, O. L. R., Introduction to Adaptive Control, Chapter 1, Self-tuning and Adaptive Control: Theory and Applications, IEE Control Eng. Series 15, Peter Peregrinus ltd., London, 1985.

Jazwinski, A. H., Adaptive Filtering, Automatica, Vol. 5, 475-485, 1969.

Jazwinski, A. H., Stochastic Processes and Filtering Theory, Academic Press, New York, 1970.

Kalman, R. E., Design of a Self-Optimizing Control System, Transactions OF THE ASME, 468-478, February 1958.

Keviczky, L., Hetthessy, J., Hilger, M., Kolostori, J., Self-tuning Adaptive Control of Cement Raw Material Blending, Automatica, vol. 14, 525-532, 1978.

Koivo, H. N., A Multivariable Self-tuning Controller, Automatica, vol. 16, 351-366, 1980.

Koppel, L. B., Conditions Imposed by Process Statics on Multivariable Process Dynamics, AIChE J., vol. 31, No. 1, January, 1985.

Kulhavy, R., Restricted Exponential Forgetting in Real-time Identification, Automatica, submitted for publication 1986.

Lam, K. P., Implicit and Explicit Self-tuning Controllers, PhD Thesis, Oxford University, 1980.

Landau, I. D., A Survey of Model Reference Adaptive Techniques -- Theory and Applications, Automatica, Vol. 10, 353-379, 1974.

Lee, K. S., Lee, W-K., Extended Discrete Multivariable Adaptive Control Using Long-term Predictor, Int. J. Control, vol.8, No. 3, 495-514, 1983.

Ljung, L., Asymptotic Behaviour of the Extended Kalman Filter as Parameter Estimator for Linear Systems, IEEE Transactions on Automatic Control, vol ac-24, No. 1, 36-51, 1979.

MacFarlane, A. C. J., Karcanias, N., Poles and Zeros of Linear Multivariable Systems: A Survey of the Algebraic, Geometric and Complex Variable Theory, Int. J. Control, vol. 24, No. 1, 33-74, 1976.

Martín-Sánchez, J. M., Contribution to Model Reference Adaptive Systems from Hyperstability Theory, Doctoral Dissertation, Universidad Politecnica de Barcelona, 1974.

Martin-Sánchez, J. M., Shah, S. L., Multivariable Adaptive Predictive Control of a Binary Distillation Column, Automatica, Vol. 20, No. 5, 607-620, 1984.

Menga, G., Mosca, E., MUSMAR: Multivariable Adaptive Regulators Based on Multistep Cost-functional. Advances in Control, D. G. Lainiotis and N. S. Tzannes (Eds.), D. Reidel Pu. Co., Dordrecht, Holland, 334-341, 1980.

Morris, A.J., Nazer, Y., Wood, R.K., MULTIVARIABLE SELF-TUNING PROCESS CONTROL, Optim. Control Applications & Methods, vol.3, 363-387, October-December 1982.

Nelson, L. W., Stear, E., The Simultaneous On-Line Estimation of Parameters and States in Linear Systems, IEEE Transactions on Automatic Control, 94-98, February 1976.

Niederlinski, A., Hajdasinski, A., Multivariable Systems Identification - A Survey, 5th. IFAC Symposium on Identification and Systems Parameter Estimation, 43-76, vol. 1, 1979.

Padilla, R. A., Padilla, C. S., Bingulac, S. P., Comments on 'The Simultaneous On-line Estimation of Parameters and States in Linear Systems', IEEE Transactions on Automatic Control, vol. AC-23, No. 1, 96-97, February 1978.

Pandya, R. N., A Class of Bootstrap Estimators for Linear System Identification, Int. J. Control, vol. 15, No. 6, 1091-1104, 1972.

Pandya, R. N., A Class of Bootstrap Estimators and their Relationship to the Generalized Two Stage Least Squares Estimators, IEEE Transactions on Automatic Control, Special Issue, vol. AC-19. No. 6, 831-835, December 1974.

Peterka, V., Adaptive Digital Regulation of Noisy Systems, IFAC Symposium on Identification and Parameter Estimation, Prague, 1970.

Peterka, V., Aström, K. J., Control of Multivariable Systems with Unknown but Constant Parameters, IFAC Symposium on Identification and System Parameter Estimation, 535-544, The Hague, 1973.

Peterka, V., A Square Root Filter for Real Time Multivariate Regression, Kibernetika, vol. 11, No. 1, 53-67, 1975.

Peterka, V., Predictor-based Self-tuning control, IFAC Symposium on Identification and Systems Parameter Estimation, 985-990, Washington D. C., June 1982.

Prager, D. L., Wellstead, P. E., Multivariable Pole-assignment Self-tuning Regulators, IEE Proc., Pt. D, vol. 18, No. 1, 9-18, January 1980.

Rowe, I., Statistical Methods for the Identification and Control of Multivariable Stochastic Systems, PhD. Thesis, Imperial College, University of London, 1967.

Rowe, I., A Bootstrap Method for the Statistical Estimation of Model Parameters, Proc. Joint Automatic Control Conf., Boulder, Colorado, 966-967, August 1969.

Robbins, H., Monro, S., A Stochastic Approximation Method, Ann. Math. Stat., vol. 22, 400-407, 1951.

Saelid, S., Foss, B., A solution to the Blow-up Problem in Adaptive Controllers, Modelling, Identification and Control, vol. 6, No. 1, 39-56, 1985.

Saridis, G. N., Stochastic Approximation Methods for Identification and Control - a Survey, IEEE Transactions on Automatic Control, Special Issue, vol. AC-19. No. 6, 798-809, December 1974.

Schumann, R., Identification and Adaptive Control of Multivariable Stochastic Linear Systems, IFAC Symposium on Identification and Systems Parameter Estimation, 1203-1212, Vol. 2, Darmstadt, 1979(a).

Schumann, R., Christ, H., Adaptive Feedforward Controllers for Measurable Disturbances, Joint Automatic Control Conference, Denver, 500-506, 1979(b).

Schumann, R., Digital Parameter-adaptive Control of an Air-conditioning Plant, Automatica, vol. 18, No. 5, 569-575, 1982.

Schumann, R., Design and Application of Multivariable Self-tuning controllers, Automatica, to be published.

Seborg, D. E., Edgar, T. F., Shah, S. L., Adaptive Control Strategies for Process Control: A Survey, AIChEJ, vol. 32, No. 6, 881-913, June 1986.

Sinha, N. K., Rozsa, P., Some Canonical Forms for Linear Multivariable Systems, Int. J. Control, vol. 23, No. 6, 865-883, 1976.

Sternard, M., Disturbance Decoupling Adaptive Control, IFAC Workshop on Adaptive Systems in Control and Signal Processing, Lund, July 1986.

Trulsson, E., Ljung, L., Adaptive Control Based On Explicit Criterion Minimization, Automatica, Vol. 21, No. 4, 385-399, 1985.

Tung, L. S., Edgar, T. F., Analysis of Control-Output Interactions in Dynamic Systems, AIChE J., vol. 27, No. 4, July, 1981.

Vagi, F., Wood, R. K., Morris, A. J., Tham, M., Self-tuning Control of Distillation Columns: Theory and Practice, American Control Conference, Boston, 1268-1269, 1985.

Warwick, K., Self-tuning Regulators - A State Space Approach, Int. J. Control, vol. 33, No. 5, 839-858, 1981.

Wellstead, P. E., Edmunds, J. M., Prager, D., Zanker, P., Self-tuning Pole-zero Assignment Regulators, Int. J. Control, vol. 30, No. 1, 1-26, 1979.

Wellstead, P. E., Zanker, P., Techniques of Self-tuning Control, Optimal Control Applications & Methods, vol. 3, 305-322, oct.-dec. 1982.

Wolowich, W. A., Falb, P. L., Invariants and Canonical Forms Under Dynamic Compensation, SIAM J. Control and Optimization, Vol. 14, No. 6, 996-1008, November 1976.

Wong, K. Y., Polak, E., Identification of Linear Discrete Time Systems Using the Instrumental Variable Method, IEEE Transactions on Automatic Control, Vol. AC-12, No, 6, 707-718, December 1967.

Ydstie, B. E., Robust Adaptive Control of Chemical Processes, PhD. Thesis, Imperial College, University of London, 1982.

Ydstie, B. E., Kershenbaum, L. S., Sargent, R. W. H.,Theory and Application of an Extended Horizon Self-tuning Controller, AIChEJ, vol. 31, No. 1, 771, 1985.

Ydstie, B. E., Auto Tuning of the Time Horizon, IFAC Workshop on Adaptive Systems in Control and Signal Processing, Lund, July 1986.

ADDITIONAL REFERENCES

Morris, A. J., Nazer, Y., Wood, R. K., Single and Multivariable Application of Self-tuning Controllers, Self-tuning and Adaptive Control: Theory and Applications, Harris and Billings, IEE Control Eng. Series 15, Peter Peregrinus ltd., London, 1985.

Hunt, K. J., Grimble, M. J., Jones, R. W., Developments in LQG Self-tuning Control, IEE Colloquium on Advances in Adaptive Control, Digest No. 1986/58, 4/1-1/7, London, April 1986.

APPENDIX A UNEXPECTED BEHAVIOUR OF THE EXTENDED HORIZON CONTROL

Our experience in simulations and experimental work illustrated that by detuning the control objective from the rigid requirements of minimum-variance the extended-horizon approach gives a fairly robust control; however, there is not much theoretical work to support that idea.

In general the available theoretical results are concerned with limiting situations. For example, Ydstie(1982) showed that if a large enough horizon is used, the extended-horizon control strategy can overcome non-minimum phase behaviour in an open-loop stable system. But the theorem does not say anything about how large the horizon should be, nor does it say if it is always a good idea to increase the horizon in order to increase robustness.

In the following a very simple system is examined in order to determine the effect of increasing the horizon on the stability of the controlled system. This study is not guided by any idea of generalization, but only by the prospect of understanding a bit more the way the extended-horizon control works.

Let us consider a single-input single-output discrete second order deterministic system, expressed in a state-space row companion canonical form :

$$x_{k+1} = \begin{vmatrix} 0 & 1 \\ a_1 & a_2 \end{vmatrix} \cdot x_k + \begin{vmatrix} b_1 \\ b_2 \end{vmatrix} \cdot u_k \qquad \text{(A.1.a)}$$

$$y_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot x_k \qquad \text{(A.1.b)}$$

The most reliable kind of system that one can think of will be considered in order to illustrate what happens to this sort of system when an extended-horizon control is used.

The canonical system (A.1) is always observable (see appendix B). In order to be controllable, the parameters in (A.1) must satisfy the relationship :

$$b_1 \neq b_2^2/(a_1 \cdot b_1 + a_2 \cdot b_2) \qquad \text{(A.2)}$$

(A.2) follows from condition (1.a) of appendix G.

The system should have also a stable inverse; hence, the zero of the equivalent input/output model of (A.1) must be inside the unit circle, i.e. :

$$|b_2/b_1 - a_2| < 1 \qquad (A.3)$$

The conditions for open-loop stability can be obtained through Jury's stability test , and they are given by :

$$a_1 > -1 \qquad (A.4.a)$$

$$a_1 < 1 - a_2 \qquad (A.4.b)$$

$$a_1 < 1 + a_2 \qquad (A.4.c)$$

These conditions are represented by the large triangle in Fig. (A1).

Finally, for controlling the process (A.1) in a single step (i.e., achievable perfect control), then :

$$b_1 \neq 0 \qquad (A.5)$$

It is clear that if the system (A.1) satisfies all the above restrictions, a stable closed-loop behaviour will be obtained using time-horizon 1 (i.e., output deadbeat control). But what happens if we use time-horizon 2 ?

According to (A.1), the predicted output at time k+2 is given by:

$$y_{k+2} = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \mathbf{x}_{k+2}$$

$$= \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \left\{ \begin{vmatrix} 0 & 1 \\ a_1 & a_2 \end{vmatrix}^2 \cdot \mathbf{x}_k + \begin{vmatrix} b_1 \\ b_2 \end{vmatrix} \cdot u_{k+1} + \begin{vmatrix} 0 & 1 \\ a_1 & a_2 \end{vmatrix} \cdot \begin{vmatrix} b_1 \\ b_2 \end{vmatrix} \cdot u_k \right\}$$

$$\Theta \qquad y_{k+2} = \begin{bmatrix} a_1 & a_2 \end{bmatrix} \cdot \mathbf{x}_k + b_1 \cdot u_{k+1} + b_2 \cdot u_k$$

considering $u_k = u_{k+1}$, the extended-horizon control is given by :

$$u_k = (y_k^* - [a_1 \; a_2] \; \mathbf{x}_k) / (b_1 + b_2) \qquad (A.6)$$

To simplify the analysis, values of $b_1=1$ and $b_2=0$ are given that do not affect the above described desirable properties of the system (A.1), but of course, generality is lost.

Replacing (A.6) into (A.1.a), yields the closed-loop equation :

$$\mathbf{x}_{k+1} = \begin{vmatrix} -a_1 & 1-a_2 \\ a_1 & a_2 \end{vmatrix} \cdot \mathbf{x}_k + \begin{vmatrix} 1 \\ 0 \end{vmatrix} y^*_k \qquad (A.7)$$

With the given values of b_1 and b_2, the system (A.1) is controllable, observable and with achievable perfect control. If the parameters a_1 and a_2 are inside the area defined by (A.4) the system (A.1) will be open-loop stable.

Closed-loop stability for time-horizon 2, depends on the following equation :

$$\lambda^2 + (a_1-a_2)\cdot\lambda - a_1 = 0 \qquad (A.8)$$

According to Jury's test, this equation presents unstable poles if the values of a_1 and a_2 are out of the triangle defined by :

$$a_1 = -1 \qquad (A.9.a)$$

$$a_1 = a_2/2 - 1/2 \qquad (A.9.b)$$

$$a_2 = 1 \qquad (A.9.c)$$

In Fig. (A1) the shaded region shows the combinations of a_1 and a_2 that produce an unstable closed-loop for time-horizon 2, even if the original system is observable, controllable, stable, minimum-phase and with achievable perfect control.

For example, if $a_1=.6$ and $a_2=-.3$, then :

$$\mathbf{x}_{k+1} = \begin{vmatrix} 0 & 1 \\ .6 & -.3 \end{vmatrix} \cdot \mathbf{x}_k + \begin{vmatrix} 1 \\ 0 \end{vmatrix} \cdot u_k \qquad (A.10)$$

The open-loop poles are P=(-.939,.639), and the closed-loop poles for time-horizon 2 are λ=(-1.346,.446), i.e., <u>the closed-loop system is unstable</u>.

This behaviour also appears sometimes when the sampling time is increased to deal with non-minimum phase systems, and seems to be related with the aliasing effect in sampling.

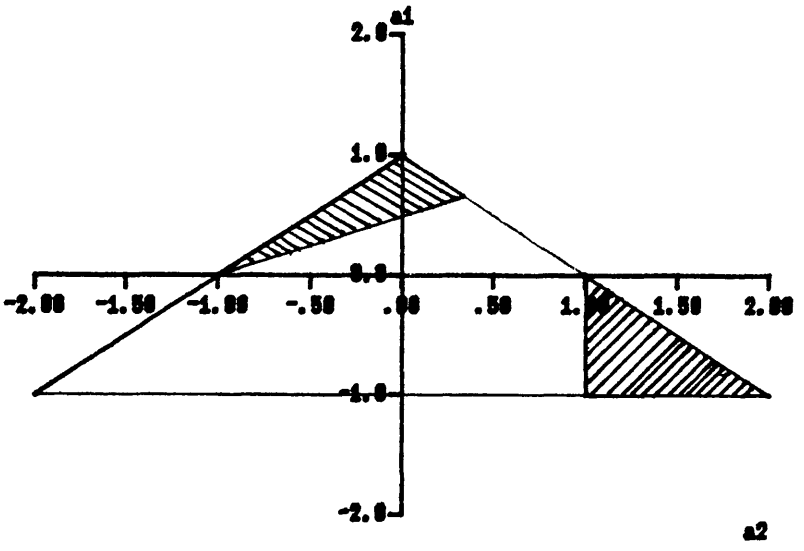


Fig. A1: Instability Region

On the other hand fig. A2 shows the values of the closed loop pole of maximum magnitude with varying time-horizon, for system (A.10).

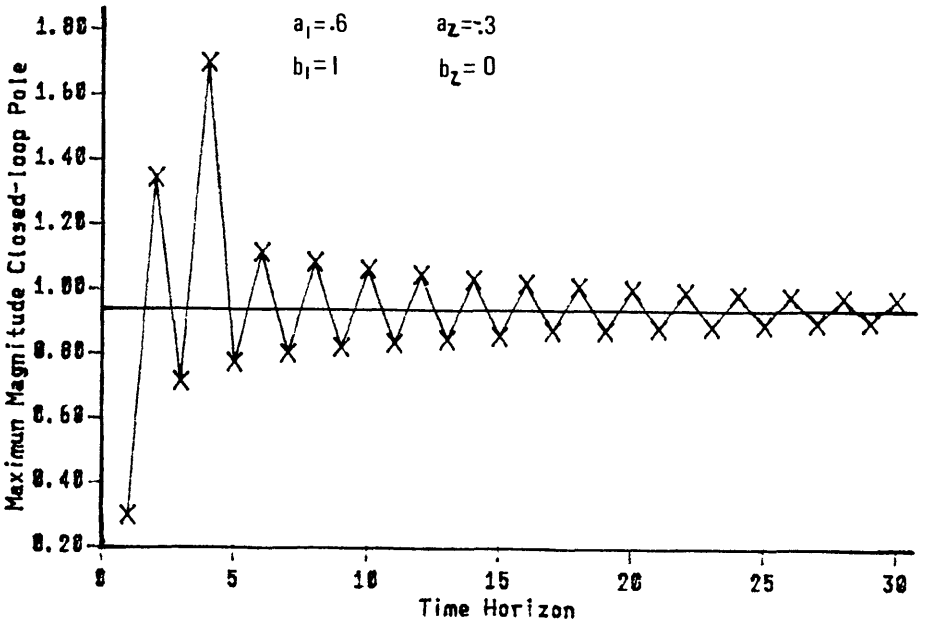


Fig. A2: Closed-loop Pole Variations

In general, the closed-loop poles approach the open-loop poles when the time-horizon is large enough.

In the particular case examined, the closed-loop poles oscillate around the open-loop poles for intermediate time-horizons, causing closed-loop instability in certain cases. Probably these oscillations are due to the fact that the control at time k cannot affect the output at time k+1 ($b_2=0$).

The important thing to note is that even for systems with desirable open-loop characteristics it is not always a good idea to increase the control horizon when an extended-horizon control strategy is used.

Despite the fact that we did not have any problem with this controller in our experiments, it is worth bearing in mind the limitations of the extended-horizon strategy in future applications.

Recently, Ydstie (1986) proposed a controller which automatically selects the control-horizon on-line, checking the closed-loop poles.

APPENDIX B STATE-SPACE AND INPUT-OUTPUT REPRESENTATION FOR A SINGLE-INPUT SINGLE-OUTPUT DETERMINISTIC SYSTEM

Consider the following input-output model:

$$(1+a_1 z^{-1}+a_2 z^{-2}+\ldots+a_n z^{-n})\, y_k = (b_1 z^{-1}+b_2 z^{-2}+\ldots+b_n z^{-n})\, u_k \qquad (B.1)$$

where y_k and u_k are the output and input respectively, z is the shift operator, and n is the order of the process.

This model can also be represented in a state-space form. Some of the most important canonical forms equivalent to (B.1) are presented below. Here we follow Sinha and Rozsa(1976), but their work is concerned with continuous-time multivariable representations.

The state-space model is given by :

$$\underline{x}_{k+1} = A\,\underline{x}_k + \underline{b}\,u_k \qquad (B.2.a)$$

$$y_k = \underline{c}\,\underline{x}_k \qquad (B.2.b)$$

where $\underline{x} \in R^n$ is the state vector, y and u are the same as in (B.1). $A \in R^n x R^n$ is the process matrix, $\underline{b} \in R^n$ is the control vector and $\underline{c} \in R^n$ is the observation vector.

i) Observable Canonical Form :

In this case the matrix A and the vectors $\underline{b}$ and $\underline{c}$ have a special form given by :

$$\underline{x}_{k+1} = \begin{vmatrix} 0 & 0 & \ldots & 0 & -a_n \\ 1 & 0 & \ldots & 0 & -a_{n-1} \\ \vdots & & & & \vdots \\ 0 & 0 & \ldots & 1 & -a_1 \end{vmatrix} \underline{x}_k + \begin{vmatrix} b_n \\ b_{n-1} \\ \vdots \\ b_1 \end{vmatrix} u_k \qquad (B.3.a)$$

$$y_k = [\,0\ 0 \ldots\ldots\ldots 1\,]\,\underline{x}_k \qquad (B.3.b)$$

Independently of the values of the a's and b's, the form (B.3) is always observable; thus a non-observable process does not have an observable canonical representation.

ii) Column Companion Form :

If the system (B.2) is controllable, doing some algebraic manipulations, the form (B.2) can be transformed into the equivalent model:

$$\underline{x}^*_{k+1} = \begin{vmatrix} 0 & 0 & .. & 0 & -a_n \\ 1 & 0 & .. & 0 & -a_{n-1} \\ . & . & . & . & . \\ . & . & . & . & . \\ 0 & 0 & .. & 1 & -a_1 \end{vmatrix} \cdot \underline{x}^*_k + \begin{vmatrix} 1 \\ 0 \\ . \\ . \\ 0 \end{vmatrix} \cdot u_k \qquad \text{(B.4.a)}$$

$$y_k = [c_1 . . . . . c_n] \cdot \underline{x}^*_k \qquad \text{(B.4.b)}$$

where the states x*'s are a linear combination of the previous states x's. The observation vector is given by :

$$\underline{c} = [0\ 0\ \ldots 1] \cdot P, \qquad \text{where } P = [\underline{b} : A \cdot \underline{b} : \ldots : A^{n-1} \cdot \underline{b}]$$

and A, $\underline{b}$ are defined by (B.3).

This form only exists for controllable processes, i.e., when the matrix P is non singular.

It can be shown that in general, the vectors $\underline{c}$ and $\underline{b}$ are related according to the following formula:

$$c_1 = b_1$$
$$c_i = b_i - b_{i-1} \cdot a_{i-1} \ ; \ i=2 . . . . . n \qquad \text{(B.4.c)}$$

EXAMPLE :

Consider a controllable system expressed by an observable canonical form model:

$$\underline{x}_{k+1} = \begin{vmatrix} 0 & -a_2 \\ 1 & -a_1 \end{vmatrix} \cdot \underline{x}_k + \begin{vmatrix} b_2 \\ b_1 \end{vmatrix} \cdot u_k$$

$$y_k = [0 \quad 1] \cdot \underline{x}_k \qquad ,$$

Then, computing the P matrix and the $\underline{c}$ vector :

$$P = \begin{vmatrix} b_2 & -b_1 \cdot a_2 \\ b_1 & b_2 - a_1 \cdot b_1 \end{vmatrix}$$

and $\quad [0\ 1] \cdot P = [b_1 \quad b_2 - a_1 \cdot b_1]$

the column companion form can now be defined :

$$\underline{x}^*_{k+1} = \begin{vmatrix} 0 & -a_2 \\ 1 & -a_1 \end{vmatrix} \cdot \underline{x}^*_k + \begin{vmatrix} 1 \\ 0 \end{vmatrix} \cdot u_k$$

$$y_k = [b_1 \quad b_2 - b_1 \cdot a_1] \cdot \underline{x}^*_k$$

iii) Row Companion Form :

The form (B.2) can be transformed, if the system is observable, to the following equivalent representation with state vector $\underline{z}^*$:

$$\underline{z}^*_{k+1} = \begin{vmatrix} 0\ 1 \ldots . 0 \\ 0\ 0 \ldots . 0 \\ \ldots\ldots\ldots \\ \ldots\ldots\ldots \\ -a_n \ldots . -a_1 \end{vmatrix} \cdot \underline{z}^*_k + \begin{vmatrix} c_1 \\ c_2 \\ . \\ . \\ c_n \end{vmatrix} \cdot u_k \qquad \text{(B.5.a)}$$

$$y_k = [\ 1\ 0 \ldots\ldots 0\ ] \cdot \underline{z}^*_k \qquad \text{(B.5.b)}$$

Like the observable canonical form (B.3) the above representation is always observable.

iv) Controllable Canonical Form :

Given a controllable (B.2) model another canonical form can be obtained with state vector $\underline{z}$:

$$\underline{z}_{k+1} = \begin{vmatrix} 0\ 1 \ldots . 0 \\ 0\ 0 \ldots . 0 \\ \ldots\ldots\ldots \\ \ldots\ldots\ldots \\ -a_n \ldots . -a_1 \end{vmatrix} \cdot \underline{z}_k + \begin{vmatrix} 0 \\ 0 \\ . \\ . \\ 1 \end{vmatrix} \cdot u_k \qquad \text{(B.6.a)}$$

$$y_k = [\ b_n \ldots\ldots b_1\ ] \cdot \underline{z}_k \qquad \text{(B.6.b)}$$

This representation is always controllable; thus a non controllable process can never be expressed in a controllable canonical form.

If a system is controllable and observable, all the presented canonical forms are equivalent, they are just a different way of representing the model.

They can always be transformed one into another; this is easy to see if one considers that all the canonical forms presented are based on just 3 vectors: the vector of the a elements in the A matrix and the vectors $\underline{b}$ and $\underline{c}$ in eq. (B.2), which are related by eq. (B.4.c).

APPENDIX C DERIVATION OF THE MEASUREMENT EQUATION FROM THE ROW COMPANION MULTIVARIABLE STATE-SPACE CANONICAL FORM

I.- Basic model :

Consider an innovations model expressed in a row companion canonical form. This model is an extension of the (B.5) model shown in Appendix B, to include multiple inputs, multiple outputs, and noise. It can be expressed by:

$$\underline{x}_{k+1} = \begin{vmatrix} A_{11} & A_{12} \ldots A_{1m} \\ A_{21} & A_{22} \ldots A_{2m} \\ \ldots & \ldots \\ \ldots & \ldots \\ A_{n1} & A_{n2} \ldots A_{nm} \end{vmatrix} \cdot \begin{vmatrix} \underline{x}_k(1) \\ \underline{x}_k(2) \\ . \\ . \\ \underline{x}_k(m) \end{vmatrix} + \begin{vmatrix} \underline{b}(1) \\ \underline{b}(2) \\ . \\ . \\ \underline{b}(n) \end{vmatrix} \cdot \underline{u}_k + \begin{vmatrix} \underline{k}(1) \\ \underline{k}(2) \\ . \\ . \\ \underline{k}(n) \end{vmatrix} \cdot \underline{e}_k \qquad \text{(C.1.a)}$$

$$\underline{y}_{k+1} = \begin{vmatrix} \underline{i}(q1) \\ . \\ . \\ . \\ \underline{i}(qm) \end{vmatrix} \cdot \begin{vmatrix} \underline{x}_{k+1}(1) \\ . \\ . \\ . \\ \underline{x}_{k+1}(m) \end{vmatrix} + \underline{e}_{k+1} \qquad \text{(C.1.b)}$$

The vectors and block matrices in eq. (C.1) are defined as follows:

The state-vector of the system $\underline{x} \in R^n$ is partitioned into m state-vectors $\underline{x}(j)$ of dimension nj, where the nj's are the structural parameters called observability indices and m is the number of input/output pairs.

The state-vectors of the m sub-systems of orders nj are defined by:

$$\underline{x}'(j) = [\, x(qj),\ x(qj+1),\ \ldots.,\ x(pj)\,] \quad ;\quad j = 1,m$$

where the nj's, qj's and pj's satisfy the following relationships:

$$n = \sum_{i=1}^{m} ni \qquad \text{(C.2.a)}$$

$$qj = \left(\sum_{i=1}^{j-1} ni \right) + 1 \qquad \text{(C.2.b)}$$

$$pj = \sum_{i=1}^{j} ni \qquad \text{(C.2.c)}$$

The vectors $\underline{u}$, $\underline{e}$, $\underline{y}$, all of dimension m, are the inputs, the innovations, and the outputs respectively.

The innovation $\underline{e}_k$ is the new information contained in $\underline{y}_k$ which is not contained in $(\underline{y}_{k-1}, \ldots, \underline{y}_0)$. The innovation $\underline{e}_k$ is defined as the difference between the measured output at time k ($\underline{y}_k$) and the output predicted by model (C.1) at time k-1 ($\hat{\underline{y}}_{k/k-1}$); $\underline{e}_k$, generally, accounts for the noise and parameter uncertainties in the model.

In eq. (C.1.a), the process matrix A is a block matrix, where the blocks A_{ij} have the following special form:

$$A_{ii} = \begin{vmatrix} 0 & 1 & 0 \ldots\ldots & 0 \\ 0 & 0 & 1 \ldots\ldots & 0 \\ \ldots\ldots\ldots\ldots & & & \\ 0 & 0 & \ldots\ldots & 1 \\ a_{ii}(1) & \ldots & a_{ii}(ni) & \end{vmatrix} \qquad A_{ij} = \begin{vmatrix} 0 & 0 & \ldots\ldots\ldots & 0 \\ 0 & 0 & \ldots\ldots\ldots & 0 \\ \ldots\ldots\ldots\ldots & & & \\ \ldots\ldots\ldots\ldots & & & \\ a_{ij}(1) & \ldots\ldots & a_{ij}(nj) & \end{vmatrix} \quad \begin{matrix} \uparrow \\ \vdots \\ ni \\ \vdots \\ \downarrow \end{matrix}$$

The observation matrix C has also a special form, where its row vectors $\underline{i}(qj)$ of dimension m are unitary vectors, with the one in the qj^{th} position.

The control matrix B and the steady state Kalman gain matrix K have no special form, where the $\underline{b}(j)$'s and the $\underline{k}(j)$'s represent row-vectors.

The following example will clarify the previous definitions:

$$\underline{x}_{k+1} = \left| \begin{array}{cc:c} 0 & 1 & 0 \\ a_{11} & a_{12} & a_{13} \\ \hdashline a_{21} & a_{22} & a_{23} \end{array} \right| \cdot \underline{x}_k + \begin{vmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{vmatrix} \cdot \underline{u}_k \qquad \text{(C.3.a)}$$

$$\underline{y}_k = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{vmatrix} \cdot \underline{x}_k \qquad \text{(C.3.b)}$$

Equation (C.3) posses 3 states and 2 input/output pairs, then n=3 and m=2.

The matrix A can be partitioned in the following form:

$$A_{11} = \begin{vmatrix} 0 & 1 \\ a_{11} & a_{12} \end{vmatrix} \; ; \qquad A_{12} = \begin{vmatrix} 0 \\ a_{13} \end{vmatrix}$$

$$A_{21} = \begin{bmatrix} a_{21} & a_{22} \end{bmatrix} \; ; \qquad A_{22} = \begin{bmatrix} a_{23} \end{bmatrix}$$

According to this partition, the the structural indices are: n1=2 and n2=1.

The qj and pj indices can be obtained from eqs. (C.2), so that: q1=1, q2=3, p1=2, and p2=3.

The state-partition vectors are:

$$\underline{x}'(1) = [\ x(1)\quad x(2)\ ]$$
$$\underline{x}'(2) = [\ x(3)\ ]$$

The row-vectors of the B matrix are:

$$\underline{b}(1) = [\ b_{11}\quad b_{12}\ ]$$
$$\underline{b}(2) = [\ b_{21}\quad b_{22}\ ]$$
$$\underline{b}(3) = [\ b_{31}\quad b_{32}\ ]$$

Finally, the unitary vectors of the observation matrix C are defined by the indices qj, so that:

$$\underline{i}(q1) = [\ 1\quad 0\quad 0\ ]$$
$$\underline{i}(q2) = [\ 0\quad 0\quad 1\ ]$$

The above example does not consider innovations, but with innovationsthe definitions are the same.

II.-Derivation of the measurement equation :

It will be shown that the state-space eq. (C.1) can be decomposed into m subsystems. These subsystems can be expressed in a measurement equation form, so that a standard RLS algorithm can be used to obtain estimates of the state-space model. To do this, the first state-partition in eq. (C.1.a) will be developed next:

$$\underline{x}_{k+1}(1) = [\ A_{11} \ldots\ldots A_{1m}\ ]\cdot\underline{x}_k + \begin{vmatrix} \underline{b}(1) \\ \vdots \\ \underline{b}(p1) \end{vmatrix}\cdot u_k + \begin{vmatrix} \underline{k}(1) \\ \vdots \\ \underline{k}(p1) \end{vmatrix}\cdot e_k \qquad (C.4)$$

where the indices pj are defined in eq. (C.2.c).

The left-hand side of eq. (C.4) considers only the elements of the first state-partition. According to this, eq. (C.4) contains the upper blocks of the A matrix and the first p1 row-vectors of the B and K matrices.

Equation (C.4) can also be written in the following form:

$$\begin{vmatrix} x_{k+1}(1) \\ \cdot \\ \cdot \\ x_{k+1}(p1) \end{vmatrix} = \begin{vmatrix} 0\ 1\ \ldots\ 0\ \ldots\ \ldots\ \ldots\ 0\ 0\ \ldots\ 0 \\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots \\ 0\ 0\ \ldots\ 1\ \ldots\ \ldots\ \ldots\ 0\ 0\ \ldots\ 0 \\ a_{11}(1)\ldots a_{11}(n1)\ldots\ldots a_{1m}(1)\ldots a_{1m}(nm) \end{vmatrix} \cdot \underline{x}_k +$$

$$+ \begin{vmatrix} \underline{b}(1) \\ \cdot \\ \cdot \\ \underline{b}(p1) \end{vmatrix} \cdot \underline{u}_k + \begin{vmatrix} \underline{k}(1) \\ \cdot \\ \cdot \\ \underline{k}(p1) \end{vmatrix} \cdot \underline{e}_k \qquad (C.5)$$

and developing the above equation yields:

$$x_{k+1}(1) = x_k(2) + \underline{b}(1)\cdot\underline{u}_k + \underline{k}(1)\cdot\underline{e}_k \qquad (C.6.1)$$

$$x_{k+1}(2) = x_k(3) + \underline{b}(2)\cdot\underline{u}_k + \underline{k}(2)\cdot\underline{e}_k \qquad (C.6.2)$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$x_{k+1}(p1) = \underline{a}(1)\cdot\underline{x}_k + \underline{b}(p1)\cdot\underline{u}(k) + \underline{k}(p1)\cdot\underline{e}(k) \qquad (C.6.p1)$$

The vectors $\underline{a}(j)$ are, in general, defined by:

$$\underline{a}(j) = [a_{j1}(1)\ldots a_{j1}(n1)\ldots\ldots\ldots a_{jm}(1)\ldots a_{jm}(nm)] \qquad (C.7)$$

The first output can be obtained from eq. (C.1.b):

$$y_{k+n1}(1) = x_{k+n1}(1) + e_{k+n1}(1) \qquad (C.8)$$

substituting (C.6.1) into (C.8):

$$y_{k+n1}(1) = x_{k+n1-1}(2) + \underline{b}(1)\cdot\underline{u}_{k+n1-1} + \underline{k}(1)\cdot\underline{e}_{k+n1-1} + e_{k+n1}(1) \qquad (C.9)$$

If the substitution is continued for the first (p1-1) states in eq. (C.6), the following result can be obtained:

$$y_{k+n1}(1) = x_{k+1}(p1) + \sum_{i=1}^{n1-1} \{ \underline{b}(i)\cdot\underline{u}_{k+n1-i} + \underline{k}(i)\cdot\underline{e}_{k+n1-i} \} + e_{k+n1}(1) \qquad (C.10)$$

Finally, replacing (C.6.p1) into (C.10):

$$y_{k+n1}(1) = \underline{a}(1)\bullet\underline{x}_k + \sum_{i=1}^{n1}\{\ \underline{b}(i)\bullet\underline{u}_{k+n1-i} + \underline{k}(i)\bullet\underline{e}_{k+n1-i}\ \} + e_{k+n1}(1) \qquad (C.11)$$

and writing (C.11) in a more compact form:

$$y_{k+n1}(1) = \underline{z}_{k+n1-1}(1)\bullet\underline{\theta}(1) + v_{k+n1}(1) \qquad (C.12)$$

where the row vector $\underline{z}$ and the column vector $\underline{\theta}$ are defined by:

$$\underline{z}_{k+n1-1}(1) = [\ \underline{x}'_k;\underline{u}'_k;.....;\underline{u}'_{k+n1-1}\ ]$$
$$\underline{\theta}'(1) = [\ \underline{a}(1);\ \underline{b}(p1);....;\underline{b}(q1)\ ]$$

The value of v is given by a linear combination of previous innovations:

$$v_{k+n1}(1) = e_{k+n1}(1) + \sum_{i=1}^{n1}\underline{k}(i)\bullet\underline{e}_{k+n1-i} \qquad (C.13)$$

q1 and p1 are given by eqs. (C.2.b) and (C.2.c) respectively.

Equation (C.13) can be generalized to an arbitrary output following the same procedure as above :

$$\begin{aligned}
y_{k+nj}(j) &= x_{k+nj}(qj) + e_{k+nj}(j)\\
&= x_{k+nj-1}(qj+1) + \underline{b}(j)\bullet\underline{u}_{k+nj-1} + \underline{k}(qj)\bullet\underline{e}_{k+nj-1} + e_{k+nj}(j)\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&= x_{k+1}(pj) + \sum_{i=1}^{nj-1}\{\ \underline{b}(qj+i-1)\bullet\underline{u}_{k+nj-i} + \underline{k}(qj+i-1)\bullet\underline{e}_{k+nj-i}\ \}\\
&\qquad + e_{k+nj}(j)
\end{aligned} \qquad (C.14)$$

The general form of eq. (C.6.pj) is:

$$\underline{x}_{k+1}(pj) = \underline{a}(j)\bullet\underline{x}_k + \underline{b}(pj)\bullet\underline{u}_k + \underline{k}(pj)\bullet\underline{e}_k \qquad (C.15)$$

Introducing (C.15) into (C.14) yields :

$$y_{k+nj}(j) = \underline{a}(j)\cdot\underline{x}_k + \sum_{i=1}^{nj} \{ \underline{b}(qj+i-1)\cdot\underline{u}_{k+nj-i} + \underline{k}(qj+i-1)\cdot\underline{e}_{k+nj-i} \} + e_{k+nj}(j) \qquad (C.16)$$

and finally, expressing (C.16) as a measurement equation for least-squares estimation :

$$y_{k+nj}(j) = \underline{z}_{k+nj-1}(j)\cdot\underline{\theta}(j) + v_{k+nj}(j) \qquad (C.17)$$

where :

$$\underline{z}_{k+nj-1}(j) = [\ \underline{x}'_k ; \underline{u}'_k ; \ldots\ldots ; \underline{u}'_{k+nj-1}\ ]$$

$$\underline{\theta}'(j) = [\ \underline{a}(j);\ \underline{b}(pj);\ldots\ldots;\ \underline{b}(qj)\ ]$$

$$v_{k+nj}(j) = e_{k+nj}(j) + \sum_{i=qj}^{pj} \underline{k}(i)\cdot\underline{e}_{k+pj-i}$$

Equation (C.17) is the one presented by Sinha (1982), but it can be rearranged to get a more suitable measurement equation for adaptive control :

$$y_{k+1}(j) = \underline{z}_k(j)\cdot\underline{\theta}(j) + v_{k+1}(j) \qquad (C.18)$$

with :

$$\underline{z}_k(j) = [\ \underline{x}'_{k+1-nj} ; \underline{u}'_k ; \ldots\ldots ; \underline{u}'k+1-nj\ ]$$

$$\underline{\theta}'(j) = [\ \underline{a}(j)\ ;\ \underline{b}(qj)\ ;\ldots\ldots;\ \underline{b}(pj)\ ]$$

$$v_{k+1}(j) = e_{k+1}(j) + \sum_{i=qj}^{pj} \underline{k}(i)\cdot\underline{e}_{k+qj-i}$$

where $v_{k+1}(j)$ is the residual uncorrelated with the measurement vector $\underline{z}_k(j)$.

The measurement eq. (C.18) permits the estimation of the model parameters in eq. (C.1). The estimation can be independently accomplished for each subsystem j, so that a standard RLS algorithm can be employed. If eq. (C.18) is combined with a state-estimator in a bootstrap manner [see section (3.2)], a state-parameter estimation method can be obtained. This method only uses I/O data and the parameter estimates are the parameters of a state-space model of the form (C.1); so that no parameter transformation is needed.

To clarify the above development, the measurement equations of the model (C.3) are given:

$$y_{k+1}(1) = \underline{z}_k(1) \cdot \underline{\theta}(1) + v_{k+1}(1)$$

where:

$\underline{z}_k(1) = [\ x_{k-1}(1),\ x_{k-1}(2),\ x_{k-1}(3);\ u_k(1),\ u_k(2);\ u_{k-1}(1),\ u_{k-1}(2)\ ]$

$\underline{\theta}'(1) = [\ a_{11}\ ,\ a_{12}\ ,\ a_{13}\ ;\ b_{11}\ ,\ b_{12}\ ;\ b_{21}\ ,\ b_{22}\ ]$

$$v_{k+1}(1) = e_{k+1}(1) + \sum_{i=1}^{2} \underline{k}(i) \cdot \underline{e}_{k+1-i}$$

and for the second output:

$$y_{k+1}(2) = \underline{z}_k(2) \cdot \underline{\theta}(2) + v_{k+1}(2)$$

where:

$\underline{z}_k(2) = [\ x_k(1),\ x_k(2),\ x_k(3);\ u_k(1),\ u_k(2)\ ]$

$\underline{\theta}'(2) = [\ a_{21}\ ,\ a_{22}\ ,\ a_{23}\ ;\ b_{31}\ ,\ b_{32}\ ]$

$$v_{k+1}(j) = e_{k+1}(j) + \sum_{i=3}^{3} \underline{k}(i) \cdot \underline{e}_{k+3-i}$$

APPENDIX D NOISE GENERATION

The need to simulate more than one independent series of random numbers, drove us to write a subroutine for this purpose, since the available random generator of the NAG library could generate only one series of random numbers at a time.

I.- Uniform Distribution Generator

A prime modulus multiplicative congruential generator is used [see Fishman (1978)].

Let $\{Z_i\}$ denote a sequence of integers defined by the recursion:

$$Z_i = A \cdot Z_{i-1} - M \cdot K_i \qquad (D.1)$$

$$K_i = (A \cdot Z_{i-1})/M \qquad (D.2)$$

where the "multiplier" A and the "modulus" M are integers. K_i is the largest positive integer in the evaluation of (D.2). Z_0 is called the seed or initial value.

Properly chosen A and M can produce a sequence whose behaviour closely resembles that of a sequence of purely random numbers.

Fishman recommended the following values :

$A = 397204094$

$M = 2^{31}-1$

$Z_0 = \{$ 1973272912, 890944011, 392021456, 1861780802,
1059483461, 1218424646, 806543588, 1855115416,
38090947, 926128229 $\}$

Each seed produces a different sequence.

II.-Normal Distribution Generator

It is known by the central limit theorem that :

$$X_k = \{ \sum_{i=1}^{k} Z_i - k \cdot m_z \} / (\sigma_z \cdot \sqrt{k}) \qquad (D.3)$$

where the Z_i's are random variables (r.v.) with variance σ_z^2 and average m_z, and X_k is approximately a normally distributed r.v. with zero mean and variance 1. If $\{Z_i\}$ is uniform, then :

$$m_z = (a + b)/2$$
$$\sigma_z^2 = (b - a)^2/12$$

considering $a \leq Z_i \leq b$ and $a = 0$, $b = 1$, yields :

$$m_z = .5 \quad \text{and} \quad \sigma_z = 1/\sqrt{12} \qquad (D.4)$$

On the other hand, it is well known that :

$$X = (V - m_v)/\sigma_v \qquad (D.5)$$

where V is normal with mean m_v and variance σ_v^2. X is normal with 0 mean and variance 1. Using k = 12 in eq. (D.3) and replacing (D.4) and (D.5), yields:

$$\{ \sum_{i=1}^{12} Z_i - 12 \cdot 0.5 \} / (\sqrt{12}/\sqrt{12}) = (V - m_v)/\sigma_v$$

finally :

$$V = \sigma_v (\sum_{i=1}^{12} Z_i - 6) + m_v \qquad (D.6)$$

then, a normally distributed sequence $\{V_i\}$, with arbitrary mean and variance can be generated from a uniformly distributed sequence $\{Z_i\}$.

APPENDIX E EXAMPLES OF MULTIVARIABLE EXTENDED-HORIZON CONTROL

a) Feedback Control

Consider the following deterministic state-space discrete-time model:

$$\underline{x}_{k+1} = A \cdot \underline{x}_k + B \cdot \underline{u}_k \qquad \text{(E.1.a)}$$

$$\underline{y}_k = C \cdot \underline{x}_k \qquad \text{(E.1.b)}$$

The predicted output vector at an arbitrary horizon can be expressed as$^{\Delta}$:

$$\hat{\underline{y}}_{k+T} = \overline{\underline{Y}}_k + D \cdot \underline{u}_k \qquad \text{(E.2)}$$

where $\hat{\underline{y}}'_{k+T} = [\hat{y}_{k+t1}, \hat{y}_{k+t2}, \ldots\ldots, \hat{y}_{k+tm}]$ and the ti's are the extended horizons for each output.

The vector $\overline{\underline{Y}}$,which represents the past information, and the controller gain matrix D are defined in eq. (3.53) in section (3.2.2).

If $\underline{y}^*$ is the set point vector, then the multivariable extended-horizon control can be obtained from eq. (E.2) :

$$\underline{u}_k = D^{-1} \cdot (\underline{y}^* - \overline{\underline{Y}}_k) \qquad \text{(E.3)}$$

The vector $\overline{\underline{Y}}_k$ can be expressed by:

$$\overline{\underline{Y}}_k = E \cdot \underline{x}_k \qquad \text{(E.4)}$$

where the row vectors of the E matrix are defined as:

$$\underline{e}(j) = \underline{i}(qj) \cdot A^{Tj}$$

and $\underline{i}(qj)$ is the unitary j^{th} row-vector of the observation matrix C.

Replacing (E.3) into (E.1.a) and using (E.4), yields :

$$\underline{x}_{k+1} = (A - B \cdot D^{-1} \cdot E) \cdot \underline{x}_k + B \cdot D^{-1} \cdot \underline{y}^* \qquad \text{(E.5)}$$

Δ See section 3.2.2

Defining the state controller gain:

$$K = D^{-1} \cdot E \qquad (E.6)$$

yields the following closed-loop equation :

$$\underline{x}_{k+1} = (A - B \cdot K) \cdot \underline{x}_k + B \cdot D^{-1} \cdot \underline{y}^* \qquad (E.7)$$

The closed-loop poles are given by the eigenvalues of the matrix $(A-B \cdot K)$.

EXAMPLE 1 :

Consider model 10 of Appendix H:

$$\underline{x}_{k+1} = \begin{vmatrix} 0.0 & 1.0 & 0.0 \\ 0.1 & 0.3 & 0.1 \\ 0.95 & 0.1 & 0.7 \end{vmatrix} \cdot \underline{x}_k + \begin{vmatrix} 0.12 & 0.10 \\ 0.36 & 0.20 \\ 0.20 & 0.30 \end{vmatrix} \cdot \underline{u}_k$$

$$\underline{y}_k = \begin{vmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{vmatrix} \cdot \underline{x}_k$$

i) If perfect control is applied, i.e., using extended-horizon 1 for both outputs, then $E_1 = C \cdot A$ and $D_1 = C \cdot B$ in (E.2) :

$$E_1 = \begin{vmatrix} 0.0 & 1.0 & 0.0 \\ 0.95 & 0.1 & 0.7 \end{vmatrix} \quad ; \quad D_1 = \begin{vmatrix} 0.12 & 0.10 \\ 0.20 & 0.30 \end{vmatrix}$$

Computing the control gain matrix K from eq. (E.6) and premultiplying by the control matrix B:

$$B \cdot K = \begin{vmatrix} 0.0 & 1.0 & 0.0 \\ 1.24 & 4.18 & 0.53 \\ 0.95 & 0.1 & 0.7 \end{vmatrix}$$

the closed-loop process matrix can be obtained :

$$(A - B \cdot K) = \begin{vmatrix} 0.0 & 0.0 & 0.0 \\ -1.14 & -3.88 & -0.43 \\ 0.0 & 0.0 & 0.0 \end{vmatrix}$$

The closed loop-poles are given by $\underline{\lambda_1 = -3.88}$, $\lambda_2 = \lambda_3 = 0$, and the controlled system is unstable.

Note that these closed-loop poles correspond to the open-loop zeros since the control-gain matrix computed with control-horizon 1 is the inverse of the process-gain matrix. Then, the system is non-minimum phase because it contains an unstable zero.

ii) The plant described by the model above cannot be controlled with a minimum-variance strategy [T = 1]. However, a stable control can be achieved with an extended-horizon law.

For example using horizon 2 for both outputs, $E_2=C\bullet A^2$ and $D_2=C\bullet(A\bullet B + B)$

$$E_2 = \begin{vmatrix} 0.1 & 0.3 & 0.1 \\ 0.68 & 1.05 & 0.5 \end{vmatrix} ; \quad D_2 = \begin{vmatrix} 0.48 & 0.30 \\ 0.49 & 0.63 \end{vmatrix}$$

$\Rightarrow$

$$K = \begin{vmatrix} -0.92 & -0.83 & -0.57 \\ 1.80 & 2.33 & 1.25 \end{vmatrix}$$

$\Rightarrow$

$$(A-B\bullet K) = \begin{vmatrix} -0.07 & 0.87 & -0.06 \\ 0.07 & 0.13 & -.06 \\ 0.59 & -0.43 & 0.44 \end{vmatrix}$$

The closed loop-poles are the eigenvalues of the matrix above :

$$\lambda_1 = 0.02; \ \lambda_2 = -0.30; \ \lambda_3 = -0.36.$$

the unstable zero has been overcome and the control produces a stable closed-loop behaviour.

iii) Different horizons can also be used for the two outputs, for example, t1=1 and t2=2 :

Consider $E_{ij} = [\ \underline{e}'_{i1} \quad \underline{e}'_{j2}\ ]'$ and $D_{ij} = [\ \underline{d}'_{i1} \quad \underline{d}'_{j2}\ ]'$, then; $E_{12} = [\ \underline{e}'_{11} \quad \underline{e}'_{22}\ ]'$ and $D_{12} = [\ \underline{d}'_{11} \quad \underline{d}'_{22}\ ]'$.

$$E_{12} = \begin{vmatrix} 0.0 & 1.0 & 0.0 \\ 0.68 & 1.05 & 0.5 \end{vmatrix} ; \quad D_{12} = \begin{vmatrix} 0.12 & 0.10 \\ 0.49 & 0.63 \end{vmatrix}$$

$\Rightarrow$

$$K = \begin{vmatrix} -2.60 & 20.0 & -1.92 \\ 3.12 & -23.70 & 2.31 \end{vmatrix}$$

$\Rightarrow$

$$(A-B\bullet K) = \begin{vmatrix} 0.0 & 1.0 & 0.0 \\ 0.41 & -2.16 & 0.33 \\ 1.37 & 3.21 & 0.39 \end{vmatrix}$$

$\Rightarrow \quad \lambda_1 = 0.96; \ \lambda_2 = -0.11; \ \underline{\lambda_3 = -2.61}$

In this case the controlled system is <u>unstable</u>

iv) Using a control with t1=2 and t2=1, $E_{21} = [\ \underline{e}'_{21} \quad \underline{e}'_{12}\ ]'$ and $D_{21} = [\ \underline{d}'_{21} \quad \underline{d}'_{12}\ ]'$:

$$E_{21} = \begin{vmatrix} 0.1 & 0.3 & 0.1 \\ 0.95 & 0.1 & 0.7 \end{vmatrix} \quad ; \quad D_{21} = \begin{vmatrix} 0.48 & 0.30 \\ 0.20 & 0.30 \end{vmatrix}$$

$$\Rightarrow$$

$$K = \begin{vmatrix} -3.04 & 0.71 & -2.14 \\ 5.19 & -0.14 & 3.76 \end{vmatrix}$$

$$\Rightarrow$$

$$(A-B*K) = \begin{vmatrix} -0.16 & 0.93 & -0.12 \\ 0.16 & 0.07 & 0.12 \\ 0.0 & 0.0 & 0.0 \end{vmatrix}$$

$$\Rightarrow \quad \lambda_1 = 0.0;\ \lambda_2 = 0.35;\ \lambda_3 = -0.43$$

and the control yields a stable closed-loop.

From this example it can be seen, as it was mentioned in chapter 2, that sometimes the non-minimum phase behaviour is associated with only one of the outputs and not with the entire system.

b) Feedback-Feedforward Control

The plant model includes now measured disturbances:

$$\underline{x}_{k+1} = A \cdot \underline{x}_k + B \cdot \underline{u}_k + H \cdot \underline{w}_k \qquad (E.8.a)$$

$$\underline{y}_k = C \cdot \underline{x}_k \qquad (E.8.b)$$

where $\underline{w}$ is the measured disturbance vector and H is a matrix of proper dimension.

The predicted output vector is[Δ]:

$$\hat{\underline{y}}_{k+T} = \overline{\underline{Y}}_k + D \cdot \underline{u}_k + S \cdot \underline{w}_k \qquad (E.9)$$

where the feedforward matrix S is defined in section (3.2.3).

If $\underline{y}^*$ is the set point vector, then the multivariable extended-horizon control law can be obtained from eq. (E.9):

$$\underline{u}_k = D^{-1} \cdot (\underline{y}^* - \overline{\underline{Y}}_k - S \cdot \underline{w}_k) \qquad (E.10)$$

Replacing eq. (E.10) into (E.8.a) and using (E.4), yields the closed-loop equation:

$$\underline{x}_{k+1} = (A - B \cdot K) \cdot \underline{x}_k + B \cdot D^{-1} \cdot \underline{y}^* + B \cdot D^{-1} \cdot S \cdot \underline{w}_k \qquad (E.11)$$

EXAMPLE 2 :

Equation (E.11) will be used to compute the control law of model 9 in Appendix H, with different control-horizons.

The model is:

$$\underline{x}_{k+1} = \begin{vmatrix} 0.000 & 1.000 & 0.00 \\ 0.038 & -.054 & -.22 \\ 0.130 & -.004 & 0.14 \end{vmatrix} \cdot \underline{x}_k + \begin{vmatrix} -2.3 & .17 \\ -1.1 & 0.30 \\ 1.2 & 1.80 \end{vmatrix} \cdot \underline{u}_k + \begin{vmatrix} 1.5 & -0.5 \\ -2.0 & 1.0 \\ 0.5 & -1.0 \end{vmatrix} \cdot \underline{w}_k$$

$$\underline{y}_k = \begin{vmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{vmatrix} \cdot \underline{x}_k$$

Δ See section 3.2.2

i) The minimum-variance control law in this case is calculated by setting the control-horizons of both outputs to 1. The E, D and S matrices are:

$$E_1 = C \cdot A = \begin{vmatrix} 0.00 & 1.000 & 0.00 \\ 0.13 & -0.004 & 0.14 \end{vmatrix}$$

$$D_1 = C \cdot B = \begin{vmatrix} -2.3 & -0.17 \\ 1.2 & 1.80 \end{vmatrix}$$

$$S_1 = C \cdot H = \begin{vmatrix} 1.5 & -0.5 \\ 0.5 & -1.0 \end{vmatrix}$$

The control matrix according to (E.6) is:

$$K_1 = D_1^{-1} \cdot E_1 = \begin{vmatrix} -.006 & -.457 & -.006 \\ .076 & .303 & .082 \end{vmatrix}$$

The closed-loop process matrix is then given by:

$$(A - B \cdot K_1) = \begin{vmatrix} 0.00 & 0.00 & 0.00 \\ 0.01 & -0.65 & -0.25 \\ 0.00 & 0.00 & 0.00 \end{vmatrix}$$

The matrices $B \cdot D_1^{-1}$ and $B \cdot D_1^{-1} \cdot S_1$ are:

$$B \cdot D_1^{-1} = \begin{vmatrix} 1.10 & 0.00 \\ 0.59 & 0.22 \\ 0.00 & 1.00 \end{vmatrix}$$

$$B \cdot D_1^{-1} \cdot S_1 = \begin{vmatrix} -1.66 & 0.55 \\ 0.90 & -0.52 \\ 0.05 & -1.00 \end{vmatrix}$$

The closed-loop equation is finally described by:

$$\underline{x}_{k+1} = \begin{vmatrix} 0.00 & 0.00 & 0.00 \\ 0.01 & -.65 & -.25 \\ 0.00 & 0.00 & 0.00 \end{vmatrix} \cdot \underline{x}_k + \begin{vmatrix} -1.1 & 0.0 \\ 0.59 & .22 \\ 0.00 & 1.0 \end{vmatrix} \cdot \underline{u}_k + \begin{vmatrix} -1.66 & .55 \\ 0.90 & -.52 \\ -0.05 & -1.0 \end{vmatrix} \cdot \underline{w}_k$$

ii) If feedback-feedforward minimum-variance control is employed, complete disturbance rejection can be achieved. However, this could mean excessive control action, ringing or even saturations.

An extended-horizon strategy can be used to detune the control. For example, using time-horizon 2 in both outputs yields:

$$E2 = C \cdot A^2 = \begin{vmatrix} 0.038 & -.054 & -.22 \\ 0.018 & 0.130 & 0.02 \end{vmatrix}$$

$$D_2 = C \cdot (I+A) \cdot B = \begin{vmatrix} -3.400 & 0.130 \\ 1.073 & 2.029 \end{vmatrix}$$

$$S_2 = C \cdot (I+A) \cdot H = \begin{vmatrix} -0.500 & -0.500 \\ 0.773 & 1.209 \end{vmatrix}$$

according to (E.6):

$$K_2 = D_2^{-1} \cdot E_2 = \begin{vmatrix} -.011 & 0.018 & 0.064 \\ -.014 & -0.055 & 0.024 \end{vmatrix}$$

and the closed-loop process matrix is:

$$(A - B \cdot K_2) = \begin{vmatrix} -.03 & 0.95 & -.15 \\ 0.02 & -.09 & 0.04 \\ 0.17 & -.13 & 0.02 \end{vmatrix}$$

The rest of the matrices in the closed-loop equation are:

$$B \cdot D_2^{-1} = \begin{vmatrix} .688 & .041 \\ .271 & .165 \\ .619 & -.848 \end{vmatrix}$$

$$B \cdot D_2^{-1} \cdot S_2 = \begin{vmatrix} -.003 & 0.015 \\ .114 & 0.186 \\ -.686 & 1.056 \end{vmatrix}$$

The closed-loop equation with extended-horizon 2 is:

$$\underline{x}_{k+1} = \begin{vmatrix} -.03 & 0.95 & -.15 \\ 0.02 & -.09 & 0.04 \\ 0.17 & -.13 & 0.02 \end{vmatrix} \cdot \underline{x}_k + \begin{vmatrix} -.69 & .04 \\ .27 & .17 \\ .62 & -.85 \end{vmatrix} \cdot \underline{u}_k + \begin{vmatrix} 0.00 & 0.02 \\ 0.11 & 0.19 \\ -.69 & 1.06 \end{vmatrix} \cdot \underline{w}_k$$

iii) Different control-horizons can also be used in the feedback-feedforward algorithm. If horizon 1 is selected for the first output and horizon 2 for the second one, the closed-loop matrices are obtained by:

$$E_{12} = \begin{vmatrix} 0.000 & 1.000 & 0.00 \\ 0.018 & 0.130 & 0.02 \end{vmatrix}$$

$$D_{12} = \begin{vmatrix} -2.300 & -.170 \\ 1.073 & 2.029 \end{vmatrix}$$

$$S_{12} = \begin{vmatrix} -1.500 & -0.500 \\ 0.773 & 1.209 \end{vmatrix}$$

Note that the first rows of the matrices above correspond with the first rows of the matrices E_1, D_1 and S_1; the second rows correspond with the second row of E_2, D_2 and S_2.

The control matrix is calculated according to (E.6):

$$K_{12}=D_{12}^{-1}\cdot E_{12}= \begin{vmatrix} -.001 & -.457 & .001 \\ 0.009 & 0.306 & .010 \end{vmatrix}$$

and the closed-loop matrices are:

$$(A-B\cdot K_{12}) = \begin{vmatrix} 0.00 & 0.00 & 0.00 \\ 0.03 & -0.65 & -0.22 \\ 0.12 & -0.01 & 0.16 \end{vmatrix}$$

$$B\cdot D_{12}^{-1} = \begin{vmatrix} 1.00 & 0.00 \\ 0.57 & 0.20 \\ -0.11 & 0.88 \end{vmatrix}$$

$$B\cdot D_{12}^{-1}\cdot S_{12} = \begin{vmatrix} 1.50 & -0.50 \\ 1.00 & -0.04 \\ 0.52 & 1.12 \end{vmatrix}$$

The closed-loop equation is:

$$\underline{x}_{k+1}= \begin{vmatrix} 0.00 & 0.00 & 0.00 \\ 0.03 & -.65 & -.22 \\ 0.12 & 0.01 & 0.16 \end{vmatrix} \cdot \underline{x}_k + \begin{vmatrix} 1.00 & .00 \\ 0.57 & .20 \\ -.11 & -.88 \end{vmatrix} \cdot \underline{u}_k + \begin{vmatrix} 1.50 & -0.50 \\ 1.00 & -0.04 \\ 0.52 & 1.12 \end{vmatrix} \cdot \underline{w}_k$$

APPENDIX F INTERACTION MEASURE WITH BRISTOL ARRAY

One of the main characteristic of a multivariable system is the interaction affecting the different inputs and outputs, which is the cause of a series of difficulties in process control. Then, it is very important to have a reliable criteria for measuring interaction. The relative-gain array [Bristol, 1966] is the most popular criteria for interaction measure, but represents only steady state interaction. There have been more recent developments in this subject [Tung and Edgar,1981; Koppel, 1985], but these are beyond the scope of this thesis.

Bristol's method will be applied to a couple of models, described in Appendix H, which were used in our simulations [see Chapter5].

I.- Consider the following two-input two-output discrete-time third order state-space model, which corresponds to model 4 in Appendix H:

$$\underline{x}_{k+1} = \begin{vmatrix} 0.0 & 1.0 & 0.0 \\ 0.1 & 0.3 & 0.1 \\ 0.95 & 0.1 & 0.7 \end{vmatrix} \cdot \underline{x}_k + \begin{vmatrix} 1.0 & 0.0 \\ 1.0 & 0.0 \\ 2.0 & 1.0 \end{vmatrix} \cdot \underline{u}_k \qquad \text{(F.1.a)}$$

$$\underline{y}_k = \begin{vmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{vmatrix} \cdot \underline{x}_k \qquad \text{(F.1.b)}$$

the transfer function $G(z) = C\cdot(z\cdot I - A)^{-1}\cdot B$, can be computed :

$$G(z) = \frac{1}{z^3-z^2+.1\cdot z-.025} \cdot \begin{vmatrix} z^2-.3 & .1 \\ 2\cdot z^2+.45\cdot z+.48 & z^2-.3\cdot z-.1 \end{vmatrix} \qquad \text{(F.2)}$$

The steady-state (s.s.) gain matrix is obtained by replacing z by 1 in eq. (F.2) :

$$\underline{y}_k = \begin{vmatrix} 9.31 & 1.33 \\ 38.9 & 7.98 \end{vmatrix} \cdot \underline{x}_k = G(1)\cdot\underline{u}_k = \Omega\cdot\underline{u}_k$$

According to Bristol, the Ω matrix itself is not a good measure of interaction because it is highly dependent on scaling and choice of units.

A relative-gain array is proposed as a better criterion. The elements m_{ij} of the matrix of relative gains are given by "the ratio of two gains representing first the process gain in an isolated loop and, second, the apparent process gain in that same loop when all other control loops are closed", [Bristol, 1966].

The matrix **M** can be easily computed as follows [Bristol, 1966]: the inverse of the s.s. gain matrix is designated by Ω^{-1} which, in the current example is:

$$\Omega^{-1} = \begin{vmatrix} 0.35 & -0.06 \\ -1.72 & 0.41 \end{vmatrix} = G^{-1}(1)$$

then, the relative-gain array is given by :

$$m_{ij} = \omega_{ij} \cdot \omega_{ji}^{-1} \qquad (F.3)$$

where the ω_{ij}'s and the ω_{ij}^{-1}'s are elements of the Ω and Ω^{-1} matrices respectively. Then, for the model we have in this case:

$$M = \begin{vmatrix} 3.29 & -2.29 \\ -2.29 & 3.29 \end{vmatrix}$$

If the largest element of **M** is close to one, it means that the system is weakly interactive.

To check the computations, the summation of each row and each column should be one.

According to this criteria the model (F.1) is highly interactive.

II.- Another system used in section (5.2.1) is model 1 in Appendix H:

$$\underline{x}_{k+1} = \begin{vmatrix} .0 & 1.0 & .0 \\ .038 & -.054 & -.22 \\ .13 & .004 & .14 \end{vmatrix} \cdot \underline{x}_k + \begin{vmatrix} -2.3 & -17.0 \\ -1.1 & 0.3 \\ 1.2 & 1.8 \end{vmatrix} \cdot \underline{u}_k \qquad (F.4.a)$$

$$\underline{y}_k = \begin{vmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{vmatrix} \cdot \underline{x}_k \qquad (F.4.b)$$

The transfer function is given by :

$$G(z) = \frac{1}{z^3-.09z^2-.045z+.034} \cdot \begin{vmatrix} -2.3z^2-.9z-.1 & -.17z^2+.32z-.44 \\ 1.2z^2-.24z-.21 & 1.8z^2-.08z-.07 \end{vmatrix}$$

and the s.s. gain matrix :

$$\Omega = \begin{vmatrix} -0.365 & -.32 \\ 0.84 & 1.83 \end{vmatrix}$$

taking the inverse :

$$\Omega^{-1} = \begin{vmatrix} -0.29 & -0.05 \\ 0.13 & 0.57 \end{vmatrix}$$

and according to (F.3) the interaction measure matrix is :

$$M = \begin{vmatrix} 1.04 & -0.04 \\ -0.04 & 1.04 \end{vmatrix}$$

then the model 2 is essentially non-interactive.

It can be seen that only the first element of the matrix M is needed to apply the criteria of interaction for two-input two-output systems. In Appendix H, a table is given which summarizes the main characteristics of the models used in our simulations. In that table the steady state interaction is given by the largest element of the M matrix.

APPENDIX G CONTROLLABILITY AND OBSERVABILITY OF A STATE-SPACE DISCRETE-TIME MODEL

There are many definitions of controllability and observability in the literature, and they are not always compatible.

To avoid confusions the definitions used in this work are given below.

A time-invariant state-space discrete-time model can be expressed in the following form.

$$\underline{x}_{k+1} = A \cdot \underline{x}_k + B \cdot \underline{u}_k \qquad \text{(G.1.a)}$$

$$\underline{y}_k = C \cdot \underline{x}_k \qquad \text{(G.1.b)}$$

where $\underline{x} \in R^n$ is the state vector, $\underline{y} \in R^q$ is the outputs vector and $\underline{u} \in R^m$ is the inputs vector.

$A \in R^n x R^n$ is the process matrix, $B \in R^n x R^m$ is the control matrix and $C \in R^q x R^n$ is the observation matrix.

1.a) Definition of controllability : A process described by eqs. (G.1) is said to be controllable, if for any initial state vector $\underline{x}_{k0}$ and any $\underline{x}_{k1}$ in the state space, there exists an input sequence $\{\underline{u}_k\}$ of finite length that transfers $\underline{x}_{k0}$ to $\underline{x}_{k1}$.

1.b) Condition for controllability : The system (G.1) is controllable iff the rank of the P matrix is equal to the number of states in the state vector, i.e., Rank[P] = n, where P is defined by :

$$P = [\ B : A \cdot B : \ldots\ldots\ldots : A^{n-1} \cdot B\ ]$$

2.a) Definition of observability : A process defined by (G.1) is said to be observable if for any state vector $\underline{x}_{k0}$, the knowledge of the input sequence $\{\underline{u}_k\}$ and the output sequence $\{\underline{y}_k\}$ over the finite sampling interval [ko,k1] suffices to determine the state vector $\underline{x}_{k0}$.

2.b)Condition for observability : The system (G.1) is observable iff the rank of the Q matrix is equal to the number of states in the state vector, i.e., Rank[Q] = n, wher Q is defined by :

$$Q = \begin{vmatrix} C \\ C\bullet A \\ \cdot \\ \cdot \\ \cdot \\ C\bullet A^{n-1} \end{vmatrix}$$

APPENDIX H MIMO MODELS USED IN SIMULATIONS

In the following, all the multivariable models used in the simulations of Chapter 5 are given. The main properties of these models are summarized in a table at the end of the appendix.

Model 1.-

$$x_{k+1} = \begin{vmatrix} 0.000 & 1.000 & 0.00 \\ 0.038 & -.054 & -.22 \\ 0.130 & -.004 & 0.14 \end{vmatrix} \cdot x_k + \begin{vmatrix} -2.3 & -.17 \\ -1.1 & 0.30 \\ 1.2 & 1.80 \end{vmatrix} \cdot u_k$$

$$y_k = \begin{vmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{vmatrix} \cdot x_k$$

Model 2.-

$$x_{k+1} = \begin{vmatrix} 0.00 & 1.000 & 0.00 \\ 0.02 & -.070 & -.10 \\ 0.27 & -.016 & 0.05 \end{vmatrix} \cdot x_k + \begin{vmatrix} -5.0 & -.10 \\ -2.5 & 0.50 \\ 3.2 & 2.80 \end{vmatrix} \cdot u_k$$

$$y_k = \begin{vmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{vmatrix} \cdot x_k$$

Model 3.-

$$x_{k+1} = \begin{vmatrix} 0.0 & 1.0 & 0.00 \\ 0.2 & 0.1 & 0.15 \\ 0.6 & 0.3 & 0.01 \end{vmatrix} \cdot x_k + \begin{vmatrix} 0.5 & 0.0 \\ 0.4 & 0.0 \\ 3.0 & 0.2 \end{vmatrix} \cdot u_k$$

$$y_k = \begin{vmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{vmatrix} \cdot x_k$$

Model 4.-

$$x_{k+1} = \begin{vmatrix} 0.00 & 1.0 & 0.0 \\ 0.10 & 0.3 & 0.1 \\ 0.95 & 0.1 & 0.7 \end{vmatrix} \cdot x_k + \begin{vmatrix} 1.0 & 0.0 \\ 1.0 & 0.0 \\ 2.0 & 1.0 \end{vmatrix} \cdot u_k$$

$$y_k = \begin{vmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{vmatrix} \cdot x_k$$

Model 5.-

$$x_{k+1} = \begin{vmatrix} 0.3 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.8 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.6 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.2 \end{vmatrix} x_k + \begin{vmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \\ 1.0 & 0.0 \\ 0.0 & 1.0 \end{vmatrix} u_k$$

$$y_k = \begin{vmatrix} 1.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 1.0 \end{vmatrix} x_k$$

Model 6.-

$$x_{k+1} = \begin{vmatrix} -.5 & 0.0 & 0.0 & 0.0 \\ 0.0 & -.9 & 0.0 & 0.0 \\ 0.0 & 0.0 & -.8 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.3 \end{vmatrix} x_k + \begin{vmatrix} 5.0 & 0.0 \\ 0.0 & 5.0 \\ 2.0 & 0.0 \\ 0.0 & 2.0 \end{vmatrix} u_k$$

$$y_k = \begin{vmatrix} 1.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 1.0 \end{vmatrix} x_k$$

Model 7.-

$$x_{k+1} = \begin{vmatrix} 0.000 & 1.000 & 0.00 \\ 0.038 & -.054 & -.22 \\ 0.130 & -.004 & 0.14 \end{vmatrix} x_k + \begin{vmatrix} -2.3 & -.17 \\ -1.1 & 0.30 \\ 1.2 & 1.80 \end{vmatrix} u_k + \begin{vmatrix} 1.5 & -0.5 \\ -1.3 & 1.0 \\ 0.5 & -1.0 \end{vmatrix} w_k$$

$$y_k = \begin{vmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{vmatrix} x_k$$

Model 8.-

$$x_{k+1} = \begin{vmatrix} 0.00 & 1.000 & 0.00 \\ 0.02 & -.070 & -.10 \\ 0.27 & -.016 & 0.05 \end{vmatrix} x_k + \begin{vmatrix} -5.0 & -.10 \\ -.25 & 0.50 \\ 3.2 & 2.80 \end{vmatrix} u_k + \begin{vmatrix} 1.0 & -.2 \\ -3.0 & 1.5 \\ 0.8 & -.6 \end{vmatrix} w_k$$

$$y_k = \begin{vmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{vmatrix} x_k$$

Model 9.-

$$x_{k+1} = \begin{vmatrix} 0.000 & 1.000 & 0.00 \\ 0.038 & -.054 & -.22 \\ 0.130 & -.004 & 0.14 \end{vmatrix} x_k + \begin{vmatrix} -2.3 & -.17 \\ -1.1 & 0.30 \\ 1.2 & 1.80 \end{vmatrix} u_k + \begin{vmatrix} 1.5 & -0.5 \\ -2.0 & 1.0 \\ 0.5 & -1.0 \end{vmatrix} w_k$$

$$y_k = \begin{vmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{vmatrix} x_k$$

Model 10.-

$$\underline{x}_{k+1} = \begin{vmatrix} 0.00 & 1.0 & 0.0 \\ 0.10 & 0.3 & 0.1 \\ 0.95 & 0.1 & 0.7 \end{vmatrix} \cdot \underline{x}_k + \begin{vmatrix} 0.12 & 0.10 \\ 0.36 & 0.20 \\ 0.20 & 0.30 \end{vmatrix} \cdot \underline{u}_k$$

$$\underline{y}_k = \begin{vmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{vmatrix} \cdot \underline{x}_k$$

Model 11.-

$$(1 + .2 \cdot z^{-1} - .24 \cdot z^{-2}) \cdot y1_k = (4 + 2.0 \cdot z^{-1}) \cdot u1_{k-2} + (1.0 + .5 \cdot z^{-1}) \cdot u2_{k-4}$$

$$(1 - .2 \cdot z^{-1} - .35 \cdot z^{-2}) \cdot y2_k = (2 - 1.4 \cdot z^{-1}) \cdot u1_{k-3} + (1.5 + .6 \cdot z^{-1}) \cdot u2_{k-5}$$

model	poles	zeros	Mmax$^{\Delta}$
1	-.3396 .2128 + .234•i .2128 - .234•i	-.649	1.038
2	-.370 .175 + .213•i .175 - .213•i	-.241	1.001
3	.533 .021 -.444	-.7	1.88
4	.92 .04 + .16•i .04 - .16•i	-.7	3.29
5	.3 .8 .6 .2	.6	1.015
6	-.6 -.9 -.8 .3	.33	1.44
7	-.3396 .2128 + .234•i .2128 - .234•i	-.649 (Gu) .797 (Gw)	1.038
8	-.370 .175 + .213•i .175 - .213•i	-.2410 (Gu) .5156 (Gw)	1.001
9	-.3396 .2128 + .234•i .2128 - .234•i	-.649 (Gu) 1.140 (Gw)	1.038
10	.92 .04 + .16•i .04 - .16•i	-3.88	11.47
11	.754 -.464 -.400 -.600	-.95	1.076

Table H.1: Model Properties

$^{\Delta}$ Mmax ≡ maximum element in the Bristol array matrix (see Appendix F).

APPENDIX I ALTERNATIVE IMPLEMENTATIONS OF SELF-TUNING REGULATORS

A.- A positional/implicit feedback SISO algorithm

The Robust Self-tuning Regulator of Ydstie (1982) will be presented in its positional form. The derivation is very similar to that of the incremental regulator of section (3.1.1).

As it was mentioned in chapter 2, the plant model in the positional formulation includes the full values of the input/output data.

For example, a suitable linearized positional model for a SISO controlled plant is:

$$\bar{A}(z^{-1})\cdot y_k = z^{-du}\cdot\bar{B}(z^{-1})\cdot u_k + e_k + \delta_k \qquad (I.1)$$

where y is the measured output, u the control input and e_k a white noise. The bias parameter δ_k is related to the offset and represents the net effect of load disturbances on the output. The control delay is represented by du and the delay operator by z^{-1}. The polynomials A and B are given by:

$$\bar{A}(z^{-1}) = 1 + \bar{a}_1\cdot z^{-1} + \bar{a}_2\cdot z^{-2} + \ldots\ldots + \bar{a}_n\cdot z^{-n}$$
$$\bar{B}(z^{-1}) = \bar{b}_1 + \bar{b}_2\cdot z^{-1} + \ldots\ldots\ldots + \bar{b}_n\cdot z^{-n+1}$$

where n is the order of the process.

If an extended-horizon control strategy with an implicit self-tuning structure is used, then a predictor for the output y must be constructed. For prediction, one can use the polynomial equation:

$$1 = \bar{F}(z^{-1})\cdot\bar{A}(z^{-1}) + z^{-T}\cdot\bar{G}(z^{-1}) \qquad (I.2)$$

where T, the prediction-horizon, is an integer larger than or equal to the upper bound of the control-delay du. $\bar{F}$ and $\bar{G}$ are polynomials of dimension T-1 and n-1 respectively.

Multiplying eq. (I.1) by $\bar{F}$, yields:

$$[1 - z^{-T}\bar{G}(z^{-1})]y_k = \bar{F}(z^{-1})\bar{B}(z^{-1})u_{k+T-du} + \bar{F}(z^{-1})e_k + \bar{F}(z^{-1})\delta_k \quad (I.3)$$

If one assumes that the bias parameter δ_k remains constant for long periods, the last term in eq. (I.3) can be replaced by a constant:

$$\bar{F}(z^{-1})\delta_k = \bar{\delta} \quad (I.4)$$

Introducing eq. (I.4) in eq. (I.3) yields:

$$\hat{y}_{k+T} = \bar{G}(z^{-1})y_k + \bar{H}(z^{-1})u_{k+T-du} + v_k + \bar{\delta} \quad (I.5)$$

where the residual is given by:

$$v_k = \bar{F}(z^{-1})e_k \quad (I.6)$$

and the polynomials $G(z^{-1})$ and $H(z^{-1})=F(z^{-1})B(z^{-1})$ are:

$$\bar{G}(z^{-1}) = \bar{\alpha}_1 + \bar{\alpha}_2 z^{-1} + \ldots\ldots + \bar{\alpha}_n z^{-n+1}$$
$$\bar{H}(z^{-1}) = \bar{\beta}_1 + \bar{\beta}_2 z^{-1} + \ldots\ldots + \bar{\beta}_{n+T-1} z^{-n-T+2}$$

As in section (3.1.1). the minimum value of du is considered, namely du=1. Then eq. (I.5) can be expressed by:

$$\hat{y}_{k+T} = \underline{\phi}_k \underline{\hat{\theta}} + v_k \quad (I.7)$$

with:

$$\underline{\phi}_k = [y_k,\ y_{k-1},\ \ldots,\ y_{k-n+1};\ u_{k+T-1},\ \ldots,\ u_{k-n+1};\ 1]$$
$$\underline{\hat{\theta}}' = [\hat{\alpha}_1,\ \hat{\alpha}_2,\ \ldots\ldots,\ \hat{\alpha}_n;\ \hat{\beta}_1,\ \hat{\beta}_2,\ \ldots\ldots,\ \hat{\beta}_{T+n-1};\ \hat{\delta}]$$

A RLS estimator can be used to estimate the parameter vector $\underline{\hat{\theta}}$. The algorithm is named positional because the full value of the variables is used in the data vector. However, it is also called the "one in the data vector" method because of the 1 used in $\underline{\phi}$ to estimate the additional parameter $\bar{\delta}$.

If the bias parameter δ_k is continuously changing, assumption (I.4) is incorrect, then (T-1) δ_i parameters must be estimated.

Given that an implicit formulation is used [see section 2.1], the control can be computed using directly the model parameter estimates.

As in section (3.1.1), the control objective is to drive the expected value of the predicted output to zero in T time intervals, where T is the control-prediction horizon.

Following Ydstie (1982), the optimal output predictor is:

$$\hat{y}_{k+T} = \overline{Y}_k + \sum_{i=1}^{T} \overline{\beta}_i u_{k+T-i} \qquad (I.8)$$

where T is larger than or equal to the control-delay du and Y_k represents the knowledge at time k and is given by:

$$\overline{Y}_k = \sum_{i=1}^{n} \hat{\alpha}_i y_{k+1-i} + \sum_{i=1}^{n-1} \hat{\beta}_{T+i} u_{k-i} + \hat{\delta} \qquad (I.9)$$

In the same way as in section (3.1.1), it can be shown that the constant control strategy yields:

$$u_k = (y^*_{k+T} - \overline{Y}_k)/d \qquad (I.10)$$

where y^*_{k+T} is the set point at time k+T and d is the reciprocal of the control gain; d is calculated by the following equation:

$$d = \sum_{i=1}^{T} \hat{\beta}_i \qquad (I.11)$$

Different extended-horizon control strategies are discussed in section (3.1.1)

The performance of the algorithm described above is studied and compared with other formulations in section (5.1.1).

B.- An explicit/incremental self-tuning regulator

Using an incremental formulation, an explicit self-tuner is derived.

In this case, as in section (3.1.1), an incremental model is used to represent the controlled plant:

$$A(z^{-1})\cdot\Delta y_k = B(z^{-1})\cdot\Delta u_k + \Delta\delta_k + \Delta e_k \qquad (I.12)$$

where Δ represents the difference operator and A, B are polynomials as in eq. (3.1).

If the bias parameter δ_k is introduced to take into account the steady-state offset, then it is natural to consider that this bias parameter remains constant for long periods, so that $\Delta\delta_k = 0$.

In section (5.1.1) an algorithm which does not consider $\Delta\delta_k = 0$ is compared with a one that assumes $\Delta\delta_k = 0$.

It can be shown [see section (3.1.1)] that for a constant control strategy, the incremental version of the control eq. (I.10) is:

$$\Delta u_k = (y^*_{k+T} - Y_k)/\hat{\beta}_T \qquad (I.13)$$

where y^*_{k+T} is the set point at time k+T and Y_k is a linear combination of known data:

$$Y_k = y_k + \sum_{i=1}^{n}\hat{\alpha}_i\cdot\Delta y_{k+1-i} + \sum_{i=1}^{n-1}\hat{\beta}_{T+i}\cdot\Delta u_{k-i} \qquad (I.14)$$

and $\hat{\beta}_T$ can be seen as the reciprocal of the control gain.

The $\hat{\alpha}$ and $\hat{\beta}$ parameters of the control eq. (I.13) can be calculated from the coefficients of the A and B polynomials in eq. (I.12).

In the following, a method is presented to compute the control parameters recursively from the model parameter estimates.

Using the polynomial identity:

$$1 - z^{-T} = F(z^{-1})\cdot A(z^{-1}) + z^{-T}\cdot G(z^{-1}) \qquad (I.15)$$

the following expression can be obtained by substituting in eq. (I.12):

$$[1 - z^{-T} - z^{-T}\cdot G(z_{-1})]\cdot \Delta y_k = F(z^{-1})\cdot B(z^{-1})\cdot \Delta u_{k-1} + F(z^{-1})\cdot \Delta e_k \qquad (I.16)$$

As usual, in eq. (I.16) the minimum possible value for du is considered and a constant bias parameter is assumed.

This eq. can be expressed in a different way:

$$y_k = y_{k-T} + G(z^{-1})\cdot \Delta y_{k-T} + H(z^{-1})\cdot \Delta u_{k-1} + v_k \qquad (I.17)$$

with $v_k = F(z^{-1})\cdot \Delta e_k$, $H(z^{-1}) = F(z^{-1})\cdot B(z^{-1})$ and :

$$F(z^{-1}) = f_0 + f_1\cdot z^{-1} + \ldots\ldots + f_{T-1}\cdot z^{-T+1}$$
$$G(z^{-1}) = \alpha_1 + \alpha_2\cdot z^{-1} + \ldots\ldots + \alpha_n\cdot z^{-n+1}$$
$$H(z^{-1}) = \beta_1 + \beta_2\cdot z^{-1} + \ldots\ldots + \beta_{n+T-1}\cdot z^{-n-T+2}$$

Following Hiram (1983) and Fortescue (1977), the parameters of the polynomials F, G and H can be derived from the model parameter estimates.

For example, the f_i's parameters can be obtained by identifying the coefficients of the same order in eq. (I.15):

$$\hat{f}_0 = 1$$
$$\hat{f}_1 + \hat{a}_1\cdot\hat{f}_0 = 0 \qquad \Rightarrow \hat{f}_1 = -\hat{a}_1\cdot\hat{f}_0$$
$$\hat{f}_2 + \hat{a}_1\cdot\hat{f}_1 + \hat{a}_2\cdot\hat{f}_0 = 0 \qquad \Rightarrow \hat{f}_2 = -\hat{a}_1\cdot\hat{f}_1 - \hat{a}_2\cdot\hat{f}_0$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$\hat{f}_{T-1} + \hat{a}_1\cdot\hat{f}_{T-2} + \ldots + \hat{a}_n\cdot\hat{f}_{T-n-1} = 0$$
$$\Rightarrow \hat{f}_{T-1} = -\hat{a}_1\cdot\hat{f}_{T-2} - \ldots - \hat{a}_n\cdot\hat{f}_{T-n-1}$$

Then, it can be seen that the parameters of the $F(z^{-1})$ polynomial can be computed by the following recursive formula:

$$\hat{f}_0 = 1$$

$$\hat{f}_i = -\sum_{j=0}^{i-1}\hat{a}_{i-j}\cdot\hat{f}_j \qquad ; \; i = 1, \ldots, T-1 \qquad (I.18)$$

The $\hat{\alpha}$ parameters can be similarly obtained from eq. (I.15):

$$\hat{a}_n \cdot \hat{f}_{T-n} + \hat{a}_{n-1} \cdot \hat{f}_{T-n+1} + \ldots + \hat{a}_1 \cdot \hat{f}_{T-1} = - \hat{\alpha}_1$$
$$\hat{a}_n \cdot \hat{f}_{T-n+1} + \hat{a}_{n-1} \cdot \hat{f}_{T-n+2} + \ldots + \hat{a}_2 \cdot \hat{f}_{T-1} = - \hat{\alpha}_2$$
. . . . . .
. . . . . .
$$\hat{a}_n \cdot \hat{f}_{T-1} = - \hat{\alpha}_n$$

and expressing in a recursive manner:

$$\hat{\alpha}_i = \sum_{j=i}^{r} \hat{a}_j \cdot \hat{f}_{T+i-j-1} \quad ; \; i = 1, \ldots, n \qquad (I.19)$$

where r = Min(n, T-1+i).

The $\hat{\beta}$ parameters are obtained by identifying coefficients in the equation:

$$H(z^{-1}) = F(z^{-1}) \cdot B(z^{-1})$$

then:

$$\hat{\beta}_1 = \hat{b}_1$$
$$\hat{\beta}_2 = \hat{b}_2 + \hat{f}_1 \cdot \hat{b}_1$$
$$\hat{\beta}_3 = \hat{b}_3 + \hat{f}_1 \cdot \hat{b}_2 + \hat{f}_2 \cdot \hat{b}_1$$
. . . . . .
. . . . . .
$$\hat{\beta}_i = \hat{b}_n + \hat{f}_{i-n} \cdot \hat{b}_{i-1} + \ldots + \hat{f}_{T-1} \cdot \hat{b}_{i+1-T}$$
. . . . . .
. . . . . .
$$\hat{\beta}_{n+T-1} = \hat{f}_{T-1} \cdot \hat{b}_n$$

and using a recursive formula:

$$\hat{\beta}_i = \sum_{j=p}^{q} \hat{f}_j \cdot \hat{b}_{i-j} \quad ; \; i = 1, \ldots\ldots, n+T-1 \qquad (I.20)$$

where p = Max(0, i-n) and q = Min(i-1, T-1)

In summary, if the following model is used for identification:

$$A(z^{-1})\cdot\Delta y_k = B(z^{-1})\cdot\Delta u_{k-1} \qquad (I.21)$$

the predicted output at time k+T is given by:

$$\hat{y}_{k+T} = Y_k + \sum_{i=1}^{T}\hat{\beta}_i\cdot\Delta u_{k+T-i} \qquad (I.22)$$

where:

$$Y_k = y_k + \sum_{i=1}^{n}\hat{\alpha}_i\cdot\Delta y_{k+1-i} + \sum_{i=1}^{n-1}\hat{\beta}_{T+i}\cdot\Delta u_{k-i} \qquad (I.23)$$

is the known information at time k.

If a minimum-variance extended-horizon control law with constant control over the horizon is used, the control is given by:

$$\Delta u_k = (y^*_{k+T} - Y_k)/\hat{\beta}_T \qquad (I.24)$$

The α and β parameters in eqs. (I.22), (I.23), and (I.24) can be obtained from the parameter estimates of the model (I.21) by the following recursive formulas:

$$\hat{\alpha}_i = \sum_{j=i}^{r}\hat{a}_j\cdot\hat{f}_{T+i-j-1} \quad ; \; i = 1, \ldots, n$$

$r = \min(n, T-1+i)$.

$$\hat{\beta}_i = \sum_{j=p}^{q}\hat{f}_j\cdot\hat{b}_{i-j} \quad ; \; i = 1, \ldots, n+T-1$$

$p = \mathrm{Max}(0, i-n)$; $q = \mathrm{Min}(i-1, T-1)$

where the f parameters are given by:

$$\hat{f}_0 = 1$$

$$\hat{f}_i = -\sum_{j=0}^{i-1}\hat{a}_{i-j}\cdot\hat{f}_j \quad ; \; i = 1, \ldots, T-1$$

The explicit algorithm described above was compared with an implicit controller in section (5.1.1).

C.- An Incremental MIMO State-Space Self-tuning Regulator

An incremental version of the MIMO state-space self-tuner presented in section (3.2.2) is developed next. An innovation state-space equation in a row-companion canonical form is used. In Chapter 5 it is shown that an incremental formulation is more robust against unmeasured deterministic disturbances.

The incremental model of the controlled process is:

$$\underline{x}_{k+1} = A \cdot \underline{x}_k + B \cdot \Delta \underline{u}_k + K \cdot \underline{e}_k \qquad \text{(I.25.a)}$$

$$\underline{e}_k = \Delta \underline{y}_k - C \cdot \underline{x}_k \qquad \text{(I.25.b)}$$

The above equation can easily be obtained from eq. (3.42) in section (3.2.2).

In this case, the vector $\underline{x} \in R^n$ represents the state-increments [which are actually the difference between two successive states of a positional equation], $\Delta \underline{u} \in R^m$ represents the vector of input-increments, $\Delta \underline{y} \in R^m$ represents the vector of output-increments and $\Delta \underline{e} \in R^m$ represents the vector of innovation-increments. The matrices A, B, K and C are of proper dimensions and are expressed in the row-companion canonical form, as it is described in Appendix C.

Parameter Estimation

It has been shown in Appendix C that the model eq. (I.25) can be decomposed in m subsystems and the following measurement equations for parameter estimation can be derived:

$$y_k(j) = y_{k-1}(j) + z_{k-1} \cdot \hat{\underline{\theta}}(j) + v_k(j) \quad ; \; j=1,\ldots m \qquad \text{(I.26)}$$

with :

$$z_{k-1} = [\underline{x}'_{k-nj};\ \Delta \underline{u}'_{k-1};\ \ldots;\ \Delta \underline{u}'_{k-nj}]$$

$$\hat{\underline{\theta}}'(j) = [\ \hat{\underline{a}}(j)\ ;\ \hat{\underline{b}}(qj)\ ;\ldots\ldots;\ \hat{\underline{b}}(pj)\ ]$$

The vectors $\underline{a}$ and $\underline{b}$ are rows of the A and B matrices as it is explained in Appendix C.

The residual is given by:

$$v_{k+1}(j) = e_{k+1}(j) + \sum_{i=qj}^{pj} \underline{k}(i) \cdot \underline{e}_{k+qj-i}$$

Equation (I.26) allows the estimation of a sub-set of the model parameters [those associated with the j output] through a standard SISO RLS algorithm. Using eq. (I.26) successively for each subsystem all the elements of the matrices A and B in eq. (I.25) can be estimated.

State Estimation

The state estimates in the incremental case can be obtained by the following filter:

$$\hat{\underline{x}}_{k+1/k} = \hat{A}_k \cdot \hat{\underline{x}}_k + \hat{B}_k \cdot \Delta \underline{u}_k \qquad \text{(I.27.a)}$$

$$\hat{\underline{x}}_{k+1} = \hat{\underline{x}}_{k+1/k} + \Gamma_k \cdot C' \cdot (\Delta \underline{y}_{k+1} - C \cdot \hat{\underline{x}}_{k+1/k}) \qquad \text{(I.27.b)}$$

where Γ is a diagonal matrix which includes a constant filter gain for each subsystem [see section (3.2.2)]. As was mentioned in Chapter 3, more sophisticated filters can be employed, however, eq. (I.27) have shown to be sufficient [see Chapter 5].

Control

To obtain the control-law, the T-step ahead predictor for the state vector must be computed. The following i-step ahead state estimator can be easily obtained using eq. (I.25):

$$\hat{\underline{x}}_{k+i} = \hat{A}^i{}_k \cdot \hat{\underline{x}}_k + \left[\sum_{j=0}^{i-1} \hat{A}^j{}_k\right] \cdot \hat{B}_k \cdot \Delta \underline{u}_{k+j} \qquad \text{(I.28)}$$

If a constant-control strategy is employed, eq. (I.28) can be reduced to:

$$\hat{\underline{x}}_{k+i} = \hat{A}^i{}_k \cdot \hat{\underline{x}}_k + \hat{A}^{i-1}{}_k \cdot \hat{B}_k \cdot \Delta \underline{u}_k \qquad \text{(I.29)}$$

Then, the i-step ahead output-increments are given by:

$$\Delta \hat{\underline{y}}_{k+i} = C \cdot (\hat{A}^i{}_k \cdot \hat{\underline{x}}_k + \hat{A}^{i-1}{}_k \cdot \hat{B}_k \cdot \Delta \underline{u}_k) \qquad \text{(I.30)}$$

The T-step ahead output predictor can be obtained by adding all the predicted increments from time k+1 to time k+T:

$$\hat{\underline{y}}_{k+T} = \underline{y}_k + C \cdot (\sum_{i=1}^{T} \hat{A}^i{}_k \cdot \hat{\underline{x}}_k + \sum_{i=1}^{T} \hat{A}^{i-1}{}_k \cdot \hat{B}_k \cdot \Delta \underline{u}_k) \qquad \text{(I.31)}$$

The extended-horizon control-law can now be computed from:

$$\Delta \underline{u}_k = D^{-1} \cdot (\underline{y}^*_{k+T} - \underline{Y}_k) \qquad (I.32)$$

in the same way as in eq. (3.52), save that in this case $\underline{Y}_k$ is calculated according to:

$$\underline{Y}_k = \underline{y}_k + C \cdot \sum_{i=1}^{T} \hat{A}^i{}_k \cdot \hat{\underline{x}}_k \qquad (I.33)$$

Equation (I.33) represents a MIMO EH control-law when all the time-horizons T_j associated with each output are set at the same value. However, different horizons for each output con also be chosen using the development of section (3.2.2).

The incremental algorithm previously described was tested and compared with a positional controller in simulations and experiments. The results are shown in section (5.2.2).