EXPERIMENTAL INVESTIGATION OF MODEL REFERENCE ADAPTIVE CONTROL ALGORITHMS

by

GEORGE A. OLUWANDE

A thesis submitted to the
VICTORIA UNIVERSITY OF MANCHESTER
for the degree of
DOCTOR OF PHILOSOPHY

FEBRUARY 1988

Control Systems Centre
The University of Manchester Institute
of Science and Technology

ABSTRACT

Although Model Reference Adaptive Control has been around for sometime, it has never gained wide recognition especially with industries. This is due to doubts about its stability and robustness which was particularly highlighted by Rohrs (M.I.T.) in his Ph.D. work. As a result of this and others, since about four years ago there have been new algorithms proposed.

The aims of this work were partly to investigate the stability of these algorithms and also, by applying them to different laboratory scaled models of industrial processes, encourage or inspire their use in industries.

As a consequence of these an extension of a single input-single output (SISO) algorithm to include MIMO systems control was derived. Also a comparison could be made of the similarities and differences between the algorithms used.

DECLARATION

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institution of learning.

ACKNOWLEDGMENTS

The author is indebted to his supervisor, Dr. P. A. Cook, for his help, guidance and criticism throughout this work; to his family, especially his parents for their financial support and sacrifices, his brothers and sisters for their understanding and patience; to his uncle, Mr. Oyayoyi, for his financial help and support; to his girlfriend for her support; to all the technicians in the Control Systems Centre for their help and assistance, and Mrs. M.V. Butterworth for typing the manuscript.

CONTENTS

Chapter 1

INTRODUCTION

1.1 HISTORICAL BACKGROUND

Two schemes have attracted much interest in adaptive control, namely the Model Reference Adaptive Control (M.R.A.C.) and the Self Tuning Regulator. This thesis is based on the MRAC Scheme, although in [1] it is shown that both can be treated as special cases of a more general philosophy or algorithm, while in [2,3] it is shown that both are quite similar and that they have common characteristics.

Adaptive control started in the late 50's in relation to Aircraft design, especially autopilot design [4] and aircraft pressure dynamics [5]. Model reference adaptive control was originally proposed by Whitaker et al [5] in 1958 to solve the problem of an unknown system parameter relating to dynamic pressure in the control of aircraft dynamics. This led to the establishment of what is known as the M.I.T. rule.

The M.I.T. rule is derived by attempting to minimize a performance index, i.e. the integral squared error. Its adjustment law was derived by approximating a gradient procedure for an integral error squared criterion; but, unfortunately, in application trials with aircraft dynamics the M.I.T. rule based adaptive controller led to unpredicted instability due then to the almost non-existence of supporting theory. Thus, early attempts failed.

From this early effort the theory of adaptive control moved from the criterion minimization approach to a stability based rational. This led to the Lyapunov redesign which was originally proposed in [6] but was further

developed by Parks in [7]. Incidentally, the Lyapunov redesign brought the appearance of Positive Real Conditions in adaptive control. The Lyapunov redesign adaptive controller was based on selecting design equations to satisfy conditions derived from Lyapunov's second method, so that stability of the control system is guaranteed. Again, due to its limitations as proposed in [7],for example use of derivatives of system output which may be too noisy, Monopoli brought out another algorithm based on the introduction of Augmented Error signals for which derivatives of signals were not required in [8]. Also another adaptive controller based on Hyperstability was proposed by Landau [9]. This was based on Popov's hyperstability theorems. Comparisons of the M.I.T. adaptive controller and that based on the Lyapunov redesign can be seen in [10], while in [11] the Lyapunov and Hyperstability approaches are compared.

A key question in MRAC concerned the stability of the resulting system, see [12], which remained an open question for many years but was finally resolved in the late 70's by the composite work of Narendra and Valavani [13], Feuer and Morse [14], Egardt [15] and Goodwin et al [16], amongst others.

Also, it must be noted that most of the early work from the 50's to the late 70's was mainly in continuous time due to non-availability of say microcomputers in those days, and partly because computer technology was still in its infancy; but from the late 70's onwards many discrete time adaptive control algorithms have also been introduced, i.e. [16-19].

1.2 WHAT IS MRACS?

The basic idea in MRAC schemes is to cause the system or plant to behave like a given reference model as shown below, the reference model specifying the desired performance of the plant to be controlled.

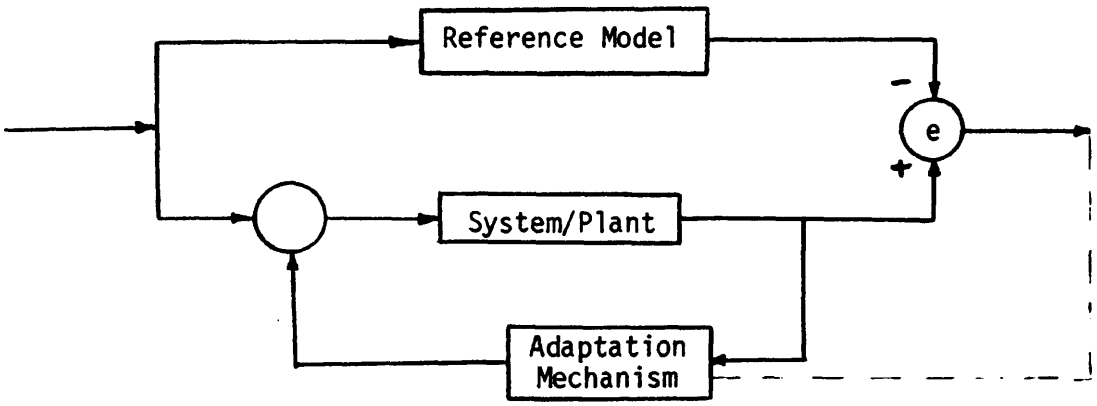


Figure 1 Basic MRAC Scheme.

The Model Reference Adaptive Control System (MRACS) can be used [20] in the solving of basic control problems such as:

(i) on-line and real time parameter identification,

(ii) adaptive state observation,

(iii) adaptive model following control.

In the first two cases above, this involves system identification with the system to be observed acting as the reference model, while in the third case the reference model specifies the desired control performances needed from the system.

There are two broad classes of algorithms which depend on the complexity of the design calculation block. The two different classes/approaches which can be used for the solution of the control problem, i.e. the design of the adaptive controllers required, are:

(a) The Indirect/Explicit Control Design in which the design calculations are carried out based on the estimated system model, i.e. the plant/system parameters are estimated on-line and the adaptive controllers and corresponding feedback laws calculated based on these estimates so that the

overall plant transfer function matches that of the reference model. This class of algorithms is commonly called "Indirect" because the evaluation of the control law is indirectly achieved via the system model. It is also called the "Explicit" scheme since the design is based on an explicit process model.

(b) The Direct or Implicit Control Design. In some cases it is possible to parametrize the system directly, i.e. no effort is made to identify the plant parameters, but the control parameters are adjusted to minimize the error between the plant and reference model outputs. This class of algorithms is called "Direct" because the control law is directly estimated. It is also called the "Implicit" scheme since the design is based on an implicit process model.

1.3 ROBUST ADAPTIVE CONTROL

As mentioned in Section 1.1, the failure of the M.I.T. rule based scheme was due to its instability. Although local stability results were available it was not globally stable as the Lyapunov and Hyperstability based schemes were.

The Lyapunov redesign owes much to the Meyer-Kalman-Yacubovich (M.K.Y.) lemma for the derivation of its global asymptotically stable adaptive systems for positive real transfer functions (see Appendix A1), whilst the Hyperstability concept concerns mainly the stability properties of the feedback systems which can be split into two blocks consisting of a Linear Time Invariant (L.T.I.) part and a Nonlinear Time Varying part [20]. The stability is proven if the feedback blocks satisfy the Popov integral inequality (see Appendix A1).

The importance of globally stable adaptive schemes cannot be overemphasized as they are more likely to perform under real conditions involving

noise, nonlinearities, etc., than localized stable adaptive schemes; but unfortunately, since the work of Rohrs et al [21] which pointed out that under certain conditions and circumstances, i.e. mis-modelling or under-modelling, an unstable control system results, the above has not been enough. Thus the design of adaptive control systems has taken on the added condition that it must be robust and this applies to both continuous and discrete-time design applications.

By robust adaptive control design it is meant that the control scheme must be stable in spite of disturbances, nonlinearities and reduced order modelling or unmodelled plant uncertainties.

There are many papers from different authors in which new robust adaptive control schemes have been suggested, see [22-29].

1.4 AIM OF THIS WORK

Although there are many adaptive control algorithms about, most of these are in continuous time; but in view of limitations of hardware and physical constraints (see [30]), e.g. saturation of analogue computer, this work is done in discrete time, especially as the advent of modern computer technology has made available inexpensive but powerful micro-processors with vast potentials and versatility.

Thus, part of the objectives of this research is to carry out experimental applications of a few of the discrete-time adaptive algorithms available on some laboratory rigs. This is because in most of the literature available the practical implementation of these algorithms seems to have been neglected, though in some papers digital simulations have been carried out, while in a few, actual tests on rigs are done.

Due to the large amount of adaptive algorithms/theories about with different approaches, but all fundamentally the same (see [1,31], it was felt that no new algorithm should be propounded, although modifications will be made where necessary; but a multivariable version of one of the algorithms is derived for use in this work.

The three algorithms chosen, while not being all those available, were felt to represent a broad view of those available. In the light of the controversial work in [21] (see also [32]), two of them are robust adaptive controllers whilst the third is not; though, as will be seen in the course of this report, exhibits some amount of robustness. The algorithm will be used on three different laboratory rigs, which represent different industrial processes. The rigs are in either Single Input Single Output (SISO)/Scalar form, or in multivariable form, though it is possible to run the multivariable rigs as scalar rigs as well.

Another aim of this work is to show in as simple a way as possible the potentials of MRACS, its robustness in industrial applications, since amongst criticisms heard about adaptive control is that it is too complex mathematically, involving a lot of proofs and theorems which make it intimidating to applicants in the industries. Because of this the algorithms have been written in BASIC language which, while slowing down the computation, is easy for most people to follow. It is implemented on the BBC microcomputer.

Thus, by showing that these algorithms work on laboratory rigs in real time, may be more people/industries might become interested in the application of MRAC industrially.

1.5 OUTLINE OF THE REPORT

This report is divided into three parts, namely Part I - the introduction, Part II - the theory, and Part III consisting of applications and tests

carried out on laboratory rigs. In Part II there are two chapters, namely 2and 3. In Chapter 2 the basic general outline of Model Reference Adaptive Control System is discussed, plus analysis and proof of global asymptotic stability; while in Chapter 3, the robust adaptive control system is discussed and how it relates to the two algorithms being used. Modifications of the basic algorithm in Chapter 2 which makes an adaptive controller robust are mentioned and explained. Also, it is shown that the three algorithms used are all related and that the two suggested robust algorithms are quite similar.

Part III consists of the last few chapters with the first three consisting of three different laboratory rigs to which the algorithms are applied. Chapter 4 deals with the Coupled Hydraulic Tank Systems in both SISO and MIMO configurations. Chapter 5 is on the Coupled Electric Drives system which can be run also either in SISO or MIMO configurations, while Chapter 6 is on the Heater Bar, a distributed parameter system assumed to be a lumped one representing a boiler amongst other things. Chapter 7, the last chapter, consists of conclusions and discussions of the work and results obtained in the previous chapters.

Chapter 2

BASIC MRAC THEORY FOR THE ALGORITHMS

2.1 INTRODUCTION

Fundamental to Discrete Adaptive Control is the fact that the system stability and convergence can be analysed based on just its inputs and outputs. The input-output approach to the stability analysis of feedback systems has provided a common framework from which many classes of systems can be studied [33,34]. In [33] it is shown that to behave properly an input-output system must have two properties, (i) bounded inputs must have bounded outputs, i.e. the system must be non-explosive, (ii) outputs must not be critically sensitive to small changes in inputs - changes such as those caused by noise.

Thus the need for global stability which implies boundedness of the sequences $\{y(t)\}$, $\{u(t)\}$ for all time t. In this chapter the Basic MRAC algorithm is explained based on the inputs, outputs of both the reference model and the system/plant. The proofs of the global convergence/stability of the algorithm for both SISO and MIMO cases are also derived.

As mentioned earlier, Model Reference Adaptive Control Schemes can be divided into 2 broad groups, namely: (i) Indirect/Explicit control scheme or (ii) Direct/Implicit control scheme. In the course of this work the Direct/Implicit control scheme is used. The main reason for this is because the tracking of the reference model output by the system output is the primary aim and thus the convergence of the output/tracking error to zero. Also, using Direct adaptive controllers meant that the persistency of excitation of the signals is not of critical importance [31]since it is not compulsory for the parameters to converge to their true values.

There are various Direct adaptive control schemes around, but it has been shown in [31] that all these schemes are variations of a general basic scheme which will be explained in this chapter and whose stability properties are also analysed. Two popular discrete-time algorithms are those of Goodwin et al [16] and Narendra and Lim [35]. The Narendra-Lim algorithm, while being stable, is not particularly attractive as it involves the use of auxiliary inputs which make the scheme appear more complex. It was thus discarded in favour of the Goodwin scheme which, bearing in mind one of the aims of the work, is much simpler. Also, a cursory look at the literature of adaptive control shows how it underlies several different algorithms.

Figure 2.1 shows the block diagram of an adaptive control system and the analysis of the scheme starts with the plant/system characteristics and stability through to the Parameter Estimation algorithm down to the control law.

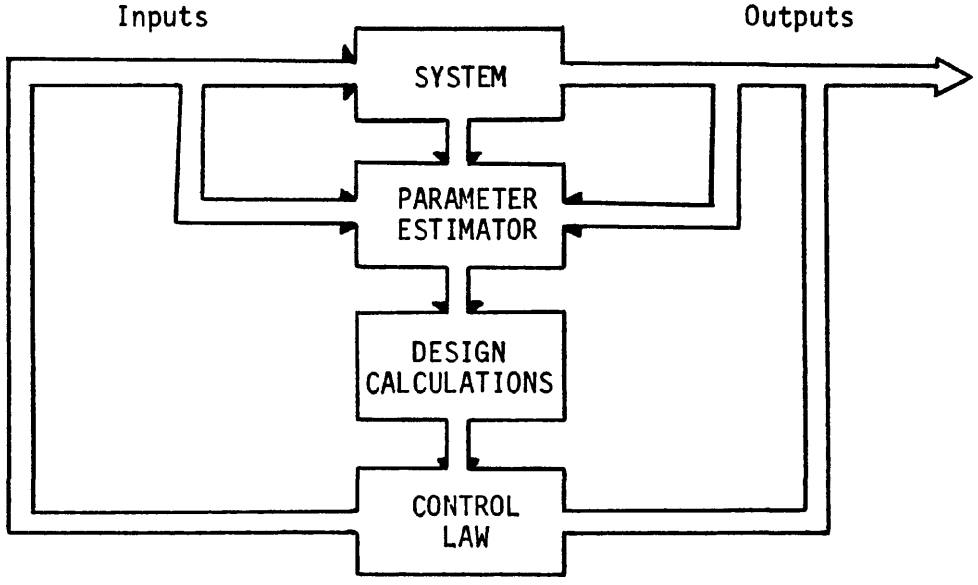


Figure 2.1 Block Diagram of an Adaptive Control System

In the next three sections the SISO case is outlined, while in the following two sections the analysis and proof is extended to the MIMO case and the last section contains some comments on the previous sections.

2.2a THE SYSTEM

The system is assumed to be represented by a Deterministic Auto Regressive Moving Average (DARMA) model of the form:

$$A(q^{-1})y(t) = q^{-d} B(q^{-1})u(t) \qquad (2.2.1)$$

where $\{u(t)\}$, $\{y(t)\}$ denote the plant input and output sequences respectively. $A(q^{-1})$, $B(q^{-1})$ are polynomial functions of the unit delay operator q^{-1},

$$A(q^{-1}) = 1 + a_1 q^{-1} + \ldots + a_n q^{-n}$$
$$B(q^{-1}) = b_0 + b_1 q^{-1} + \ldots + b_m q^{-m} \qquad b_0 \neq 0$$

d represents the system time delay.

The following assumptions will be made about the system [28]:

1. The time delay d is known.
2. An upper bound for the orders of the polynomials, i.e. n,m, is known.
3. All poles of the inverse of the model (2.2.1), (i.e. the zeros of the polynomial $B(z^{-1})$, lie inside or on the closed unit disk.
4. All controllable poles of the inverse of the model (2.2.1), (i.e. the zeros of the transfer function $B(z^{-1})/A(z^{-1})$), lie strictly inside the unit circle.
5. Any poles of the inverse of the model (2.2.1) on the unit circle have a Jordan block size of 1.

Equation (2.2.1) can be replaced by the d-step ahead predictor:

$$y(t+d) = \alpha(q^{-1})y(t) + \beta(q^{-1})u(t) \qquad (2.2.2)$$

where $\alpha(q^{-1}) = \alpha_0 + \alpha_1 q^{-1} + \ldots + \alpha_{n-1} q^{n-1}$

$\beta(q^{-1}) = \beta_0 + \beta_1 q^{-1} + \ldots + \beta_{m+d-1} q^{-(m+d-1)}, \quad \beta_0 \neq 0$

Here it is assumed that the sign and upper bound on the magnitude of β_0 is known.

2.2b THE REFERENCE MODEL

The desired output sequence $\{y_m^*(t)\}$ satisfies the following reference model:

$$E(q^{-1})y_m^*(t) = q^{-d'} H(q^{-1})r(t) \qquad (2.2.3)$$

with associated transfer function $G(z)$:

$$G(z^{-1}) = \frac{z^{-d} g H(z^{-1})}{E(z^{-1})} \qquad (2.2.4)$$

where g is a constant gain and

$$H(z^{-1}) = h_0 + h_1 z^{-1} + \ldots + h_i z^{-i} \qquad h_0 = 1$$
$$E(z^{-1}) = e_0 + e_1 z^{-1} + \ldots + e_k z^{-k} \qquad e_0 = 1$$

subject to: (i) $E(z^{-1})$ is stable

(ii) $d' = d$

(iii) $|y_m^*(t)| \leq m_1 < \infty$ for all t.

The control objective is to achieve:

$$\lim_{t\to\infty} [y(t) - y_m^*(t)] = 0$$

Before going further, it must be pointed out that the assumptions made about the system, i.e. 1-5, cut across almost all of the proposed adaptive control algorithms in the literature from various authors. One of the very few exceptions is [36] in which *a priori* knowledge of the relative degree n^* of the plant and the sign of the high frequency gain are dispensed with, but even them an upper bound on the plant order must be known.

2.3 THE ADAPTIVE CONTROL LAW AND ALGORITHM (See [16,31])

Factoring β_0 from (2.2.2) yields:

$$\frac{1}{\beta_0} y(t+d) = \frac{1}{\beta_0} [\alpha(q^{-1})]y(t) + \frac{1}{\beta_0} [\beta(q^{-1})]u(t) \qquad (2.3.1)$$

where $\beta_0 \neq 0$.

$\Rightarrow$ (2.3.1) can be written as:

$$\frac{1}{\beta_0} y(t+d) = \alpha'(q^{-1})y(t) + \beta'(q^{-1})u(t) \qquad (2.3.2)$$

where $\alpha'(q^{-1}) = \frac{1}{\beta_0}\alpha(q^{-1}) = \frac{\alpha_0}{\beta_0} + \frac{\alpha_1}{\beta_0} q^{-1} + \ldots + \frac{\alpha_{n-1}}{\beta_0} q^{-n+1}$

$$= \alpha_0' + \alpha_1' q^{-1} + \ldots + \alpha_{n-1}' q^{-n+1}$$

$$\beta'(q^{-1}) = \frac{1}{\beta_0}\beta(q^{-1}) = \frac{\beta_0}{\beta_0} + \frac{\beta_1}{\beta_0} q^{-1} + \ldots + \beta_{m+d-1} q^{-(m+d-1)}$$

$$= 1 + \beta_1'(q^{-1}) + \ldots + \beta_{m+d-1}' q^{-(m+d-1)}$$

Thus (2.3.2) becomes:

$$y(t+d) = \beta_0[\alpha_0'y(t) + \ldots + \alpha_{n-1}'y(t-n+1) + u(t) + \beta_1'u(t-1) + \ldots$$

$$\ldots + \beta_{m+d-1}' u(t-m-d+1) \qquad (2.3.3)$$

but $e(t+d) = y(t+d) - y^*(t+d)$

$= $ output/tracking error

$$= \beta_0[u(t) + \alpha_0'y(t) + \ldots + \alpha_{n-1}'y(t-n+1) + \beta_1'u(t-1) + \ldots$$

$$\ldots + \beta_{m+d-1}'u(t-m-d+1) - \frac{1}{\beta_0} y_m^*(t+d)] \qquad (2.3.4)$$

$$= \beta_0[u(t) - \phi(t)^T\theta_0'] \qquad (2.3.5)$$

where $\phi(t)^T = [-y(t),\ldots,-y(t-n+1),-u(t-1),\ldots,-u(t-m-d+1),y_m^*(t+d)]$

$=$ regression vector consisting of past values of system input and output plus projected value of the reference model output.

$\theta_0'^T = [\alpha_0',\ldots,\alpha_{n-1}',\beta_1',\ldots,\beta_{m+d-1}', 1/\beta_0]$ - the parameter vector.

Assuming the values of the parameter vector $\theta_0^T(t)$ are known, it is obvious that for the tracking error to be made zero then:

$$u(t) = \phi(t)^T\theta_0' \qquad (2.3.6)$$

but, since θ_0' is unknown, the control law will be recursively estimated. The Adaptive algorithm will be of the form:

$$\hat{\theta}(t) = f(\hat{\theta}(t-1), D(t), t) \qquad \text{(see [28],}$$

e.g.

$$\hat{\theta}(t) = \hat{\theta}(t-1) + M(t-1)\phi(t-d)\bar{e}(t) \tag{2.3.7}$$

where $\hat{\theta}(t-1)$ = the parameter estimate vector at time t-1

$M(t-1)$ = algorithm gain (possibly a matrix)

$\phi(t-d)$ = regression vector of some kind composed of selected elements of $Y(t-d)$, $U(t-d)$

d = an integer

$e(t)$ = modelling error of some kind.

In general it is the Orthogonalized Projection Algorithm [31] that is used for the Adaptive algorithm, i.e.

Step I:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{P(t-2)\phi(t-1)}{C+\phi(t-1)^T P(t-2)\phi(t-1)}\, e(t) \tag{2.3.8}$$

where $e(t) = y(t) - y_m^*(t+d)$

Step II:

$$P(t-1) = P(t-2) - \frac{P(t-2)\phi(t-1)\phi(t-1)^T P(t-2)}{C+\phi(t-1)^T P(t-2)\phi(t-1)} \tag{2.3.9}$$

with the initial estimate $\hat{\theta}(1)$ given and $P(0) = I$.

If $C = 1$ in the algorithm (2.3.8), (2.3.9) above it becomes the Least Squares Algorithm, while if $P(t)$ is made a constant matrix then the algorithm becomes the Projection algorithm with the elimination of Step II above, i.e.

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{P^*\phi(t-1)\, e(t)}{C+\phi(t-1)^T P^*\phi(t-1)} \tag{2.3.10}$$

where $P^* = aI$ a constant matrix which allows equation (2.3.10) to become:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{a\phi(t-1)\, e(t)}{C+a\phi(t-1)^T\phi(t-1)} \qquad (2.3.11)$$

$$= \hat{\theta}(t-1) + \frac{\phi(t-1)e(t)}{C^*+\phi(t-1)^T\phi(t-1)} \qquad (2.3.12)$$

which is the Projection Algorithm II.

Since part of the aim of this work is to simplify the algorithm used as much as possible, the Projection Algorithm was used; but it must be stated that this is the slowest of the three algorithms mentioned above to convergence, though in terms of computation times it is the fastest due to its relative simplicity as on-line control computations can be a significant fraction of the time between samples. Thus, computation of the Least Squares algorithm controller might take longer than for the Projection algorithm. Also, since the Projection algorithm gain does not go to zero, it can automatically track-time varying parameters whereas for the Least Squares algorithm, modifications such as covariance modification or exponential data weighting or finite data window, need to be made. Therefore the Adaptive algorithm becomes [16] for systems with time delay d,

$$\hat{\theta}(t) = \hat{\theta}(t-d) - \frac{1}{\hat{\beta}_o}\phi(t-d)[1 + \phi(t-d)^T\phi(t-d)]^{-1}e(t) \qquad (2.3.13)$$

and the control law becomes:

$$u(t) = \phi(t)^T\hat{\theta}(t) \qquad (2.3.14)$$

where $\hat{\beta}_o$ is a fixed constant and $\hat{\theta}(t)$ is a p-vector of reals depending on d initial values $\hat{\theta}(o)\ldots\hat{\theta}(d-1)$.

Note: $0 < \beta_o/\hat{\beta}_o \leq 2$ for stability and convergence.

The above forms the basic algorithm used in the course of this work. Alternatively, another projection algorithm that can be used is also suggested in [16] called the Projection Algorithm I. This is given below for completeness for the same system and reference model defined in Section 2.2 .

The Projection Algorithm I:

$$\text{Let } \theta_o^T = (\alpha_o,\ldots,\alpha_{n-1},\beta_o,\ldots,\beta_{m+d-1})$$

$$\text{then } y(t+d) = \phi(t)^T\theta_o \tag{2.3.15}$$

where

$$\phi(t) = (y(t),\ldots,y(t-n+1),u(t),\ldots,u(t-m-d+1))$$

$$\text{Again } e(t+d) = y(t+d) - y_m^*(t+d)$$

$$= \phi(t)^T\theta_o - y_m^*(t+d) \tag{2.3.16}$$

If $u(t)$ is chosen to satisfy $\phi(t)^T\theta_o = y^*(t+d)$ the control objective will be satisfied, but since θ_o is unknown it is estimated at every time step using an adaptive algorithm of the form below:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{a(t)\phi(t-d)[y(t)-\phi(t-d)^T\hat{\theta}(t-1)]}{1 + \phi(t-d)^T\phi(t-d)} \tag{2.3.17}$$

$$\text{and } \phi(t)^T\hat{\theta}(t) = y_m^*(t+d) \tag{2.3.18}$$

The control input $u(t)$ is then computed from (2.3.18) which is the control law, i.e.

$$u(t) = \frac{1}{\hat{\theta}_{n+1}(t)}[-\hat{\theta}_1(t)y(t)-\hat{\theta}_2(t)y(t-1)-,\ldots,-\hat{\theta}_n(t)y(t-n+1),-\hat{\theta}_{n+2}u(t-1)$$

$$-,\ldots,-\hat{\theta}_{n+m+d}u(t-m-d+1)+y_m^*(t+d)] \tag{2.3.19}$$

where $\hat{\theta}_j(t)$ denotes the jth element of $\hat{\theta}(t)$.
Note that $0 < a(t) < 2$. Also, equation (2.3.19) can be compared to equation (2.3.14) for the Projection Algorithm II. It can be seen that there are similarities in the two algorithms.

2.4 PROOF OF THE GLOBAL CONVERGENCE OF THE CONTROL ALGORITHM [16,31]

Conditions for the global stability of the Adaptive control system include global convergence of the tracking error to zero with bounded signals, i.e. the system inputs and outputs remain bounded for all time.

Essential for the proof of global stability is what is known as the Key Technical Lemma [16,31] which is used in the convergence analysis of various adaptive control systems. It will be explained and proved below.

2.4.1 The Key Technical Lemma

Lemma: If the following conditions are satisfied for some given sequences $\{s(t)\}$, $\{\sigma(t)\}$, $\{b_1(t)\}$ and $\{b_2(t)\}$:

1. $$\lim_{t\to\infty} \frac{s(t)^2}{b_1(t)+b_2(t)\sigma(t)^T\sigma(t)} = 0 \qquad (2.4.1)$$

where $\{b_1(t)\}$,$\{b_2(t)\}$ and $\{s(t)\}$ are real scalar sequences and $\{\sigma(t)\}$ is a real $(p\times 1)$ vector sequence.

2. Uniform boundedness condition

$$0 < b_1(t) < K < \infty \quad \text{and} \quad 0 \le b_2(t) < K < \infty \qquad (2.4.2)$$
$$\text{for all} \quad t \ge 1$$

3. Linear boundedness condition

$$||\sigma(t)|| \le C_1 + C_2 \max_{0\le\tau\le t} |s(\tau)| \qquad (2.4.3)$$

where $0 < C_1 < \infty$ and $0 < C_2 < \infty$

it follows that:

(i) $\lim_{t\to\infty} s(t) = 0$

(ii) $\{||\sigma(t)||\}$ is bounded.

Proof: If $\{s(t)\}$ is a bounded sequence then by (2.4.3), $\{||\sigma(t)||\}$ is a bounded sequence. Then by (2.4.1) and (2.4.2) it follows that

$$\lim_{t\to\infty} s(t) = 0 .$$

Now assuming that $\{s(t)\}$ is unbounded, it follows that there exists a subsequence $\{t_n\}$ such that

$$\lim_{t_n\to\infty} |s(t_n)| = \infty$$

and

$$|s(t)| \leq |s(t_n)| \quad \text{for} \quad t \leq t_n$$

Thus, along the subsequence t_n

$$\left|\frac{s(t_n)}{[b_1(t_n)+b_2(t_n)\sigma(t_n)^T\sigma(t_n)]^{\frac{1}{2}}}\right| \geq \frac{|s(t_n)|}{[K+K||\sigma(t_n)||^2]^{\frac{1}{2}}} \qquad \text{using (2.4.2)}$$

$$\geq \frac{|s(t_n)|}{K^{\frac{1}{2}}+K^{\frac{1}{2}}||\sigma(t_n)||}$$

$$\geq \frac{|s(t_n)|}{K^{\frac{1}{2}}+K^{\frac{1}{2}}[C_1+C_2|s(t_n)|]} \qquad \text{using (2.4.3)}$$

Hence,

$$\lim_{t_n \to \infty} \left| \frac{s(t_n)}{[b_1(t_n)+b_2(t_n)\sigma(t_n)^T\sigma(t_n)]^{\frac{1}{2}}} \right| \geq \frac{1}{K^{\frac{1}{2}} C_2} > 0$$

but this contradicts (2.4.1) and therefore the assumption that $\{s(t)\}$ is unbounded is false and the result follows.

2.4.2 Stability Analysis of the Adaptive Projection Algorithm

For the stability analysis of the algorithm given by equations (2.3.13), (2.3.14), it is essential to show that the Euclidean norm of the vector $\tilde{\theta}(t) = \hat{\theta}(t) - \theta_o'$ is a non-increasing function along the trajectories of the algorithm.

Lemma: Along the solutions of (2.3.13) and (2.3.14)

(i) $$||\tilde{\theta}(t)||^2 - ||\tilde{\theta}(t-d)||^2 \leq 0$$

(ii) $$\lim_{t\to\infty} \frac{\phi(t)\tilde{\theta}(t)}{1+\phi(t)^T\phi(t)^{\frac{1}{2}}} = 0$$

where $\tilde{\theta}(t) = \hat{\theta}(t) - \theta_o'$.

Proof of (i) using (2.3.13):

$$||\tilde{\theta}(t)||^2 - ||\tilde{\theta}(t-d)||^2$$

$$= \frac{1}{\hat{\beta}_o} \left[\frac{1/\hat{\beta}_o \phi(t-d)^T\phi(t-d)e(t)^2}{[1+\phi(t-d)^T\phi(t-d)]^2} - \frac{2\phi(t-d)^T\tilde{\theta}(t-d)e(t)}{[1+\phi(t-d)^T\phi(t-d)]} \right] \quad (2.4.4)$$

but on the R.H.S. there is a term $\phi(t-d)\tilde{\theta}(t-d)$ which can be rearranged, i.e.

$$\phi(t-d)^T\tilde{\theta}(t-d) = \phi(t-d)^T[\hat{\theta}(t-d)-\theta_o']$$

$$= \phi(t-d)^T\hat{\theta}(t-d)-\phi(t-d)^T\theta_o'$$

$$= \frac{e(t)}{\beta_o} \quad \text{(from equations (2.3.5) and (2.3.14).}$$

Substituting this back into (2.4.4) and rearranging:

$$||\tilde{\theta}(t)||^2 - ||\tilde{\theta}(t-d)||^2 = \frac{1}{\hat{\beta}_o}\left[-\frac{2}{\beta_o} + \boxed{\frac{1/\hat{\beta}_o.\phi(t-d)^T\phi(t-d)}{1+\phi(t-d)^T\phi(t-d)}}\right]\frac{e(t)^2}{1+\phi(t-d)^T\phi(t-d)} \qquad (2.4.5)$$

from which the dotted block

$$\frac{1/\hat{\beta}_o.\phi(t-d)^T\phi(t-d)}{1+\phi(t-d)^T\phi(t-d)} < \frac{1}{\hat{\beta}_o}$$

and if $\frac{1}{\hat{\beta}_o} \leq -\frac{2}{\beta_o}$

$\longrightarrow$ $\frac{\beta_o}{\hat{\beta}_o} \leq -2$ which is true (see (2.3.14)).

Then the whole R.H.S. of (2.4.5) will be zero or negative, hence

$$||\tilde{\theta}(t)||^2 - ||\tilde{\theta}(t-d)||^2 = \frac{1}{\hat{\beta}_o}\left[-\frac{2}{\beta_o} + \frac{1/\hat{\beta}_o.\phi(t-d)^T\phi(t-d)}{1+\phi(t-d)^T\phi(t-d)}\right]\frac{e(t)^2}{1+\phi(t-d)^T\phi(t-d)}$$
$$\leq 0 \qquad (2.4.6)$$

For part (ii) of the lemma it is noted from (i) that $||\tilde{\theta}(t)||^2$ is a bounded non-increasing function, hence it converges. From equations (2.3.5) and (2.3.14) :

$$\phi(t)\tilde{\theta}(t) = e(t+d)$$

thus (ii) becomes:

$$\lim_{t\to\infty} \frac{e(t+d)}{[1+\phi(t)^T\phi(t)]^{\frac{1}{2}}} = 0$$

Noting that

$$\frac{1}{\hat{\beta}_o}\left[-\frac{2}{\beta_o}+\frac{1/\hat{\beta}_o.\phi(t-d)^T\phi(t-d)}{1+\phi(t-d)^T\phi(t-d)}\right]$$

is bounded away from zero, it is concluded that

$$\lim_{t\to\infty}\frac{e(t+d)^2}{1+\phi(t)^T\phi(t)} = 0 \qquad \text{using (2.4.6)}$$

and hence

$$\lim_{t\to\infty}\frac{e(t+d)}{[1+\phi(t)^T\phi(t)]^{\frac{1}{2}}} = 0 .$$

Alternatively, using the Key Technical Lemma (K.T.L.), condition 1 of K.T.L. is established with $s(t) = e(t)$, $\sigma(t) = \phi(t-d)$, $b_1(t) = 1$, $b_2(t) = 1$. Condition 2 is also clearly satisfied. While to establish condition 3, namely that $||\phi(t-d)||$ is bounded by $e(t)$, it is noted from [31] that there exists constants $m_3 < \infty$ and $m_4 < \infty$ such that

$$u(k-d) \leq m_3 + m_4 \max_{1\leq\tau\leq t}|y(\tau)| \quad \text{for all} \quad 1 \leq k \leq t$$

Therefore, using the definition of $\phi(t-d)$,

$$||\phi(t-d)|| \leq p\{m_3+[\max(1,m_4)]\max_{1\leq\tau\leq t}|y(\tau)|\}$$

where p is the dimension of $\phi(t-d)$; but

$$|e(t)| \geq |y(t)| - |y_m^*(t)| \geq |y(t)| - m_1 \; ; \; m_1 < \infty \quad \text{(see Section 2.2b)} .$$

Hence:

$$||\phi(t-d)|| \leq p\{m_3+[\max(1,m_4)]\max_{1\leq\tau\leq t}(|e(\tau)|+m_1)\}$$

$$= c_1 + c_2 \max_{1 \le \tau \le t} |e(\tau)|$$

$$0 \le c_1 < \infty, \quad 0 < c_2 < \infty$$

and it follows that the linear boundedness condition is also satisfied.

The properties of this adaptive algorithm which is used also include:

(i) $$||\hat{\theta}(t) - \theta_0'|| \le ||\hat{\theta}(t-d) - \theta_0'|| \le ||\hat{\theta}_0 - \theta_0'|| \; ; \; t \ge d$$

(ii) $$\lim_{N\to\infty} \sum_{t=d}^{N} \frac{e(t)}{1+\phi(t-d)^T\phi(t-d)} < \infty$$

(iii) $$\lim_{t\to\infty} ||\hat{\theta}(t) - \hat{\theta}(t-d)|| = 0$$

(iv) $$\lim_{N\to\infty} \sum_{t=d}^{N} ||\hat{\theta}(t) - \hat{\theta}(t-d)||^2 < \infty$$

The proofs of these can be seen in [28] and the appendix.

2.5 MIMO SYSTEMS INTRODUCTION

Although a large amount of literature exists on adaptive control, with few exceptions, e.g. [16,37-42], most have been devoted to SISO problems. While attempts have been made at extending these SISO algorithms to the multivariable case, there have been difficulties, i.e. finding equivalent *a priori* information similar to those in the SISO algorithms (i.e. (a) knowing the upper bounds on the plant order and the relative degree of the plant transfer function, (b) polarity of the high frequency gain, and lastly (c) the time delay d of the system must be known), and what to do about cross couplings between system inputs and outputs.

In [37], Elliot and Wolovich approached the MIMO control problems using

linear multivariable system theory and came up with the idea of an interactor matrix, (this will be defined and explained briefly later), which gives a measure of the *a priori* information on the system. Unfortunately for this work, their algorithm is in continuous time and it is not strictly speaking a model reference scheme. However, their work led to Goodwin and Long [38] extending the results of [16] to a wider class of systems with the introduction of the system interactor matrix.

In [39] Johannsson went a bit further in deriving a direct adaptive controller which can tackle some non-minimum phase systems; but although part of his criticism of [16] includes its being overparametrized, his own algorithm for the minimum phase system reduces to a similar algorithm to that in [16] and [38].

In [40] a robust algorithm is proposed which does not require persistency of excitation and this will be dealt with in the next chapter, while both [41,42], though having interesting algorithms, [41] using an Indirect Adaptive Controller and [42] an algorithm based on the theory of Variable Structured Systems (V.S.S.) but in continuous time, were not considered due to the framework within which this thesis is based.

Since a simple algorithm is required, the MIMO adaptive algorithm applied to the rigs in this work was based on that in [38] which as mentioned above for minimum phase systems is similar to that suggested in [39].

2.5.1 MIMO Systems

For Multi-Input Multi-Output systems described in D.A.R.M.A. form

$$A(q^{-1})y(t) = B(q^{-1})u(t) \qquad (2.5.1)$$

where $\{u(t)\}$, $\{y(t)\}$ denote the $r\times 1$ and $m\times 1$ input and output vectors

respectively, the following assumptions are made:

1. The transfer function $T(z) \triangleq B(z^{-1})/A(z^{-1})$ has rank m.
2. Upper bounds are available for the orders of the polynomial.
3. The system is stably invertible.
4. $\xi(z)$ (the interactor matrix) is known.

From multivariable theory output tracking is possible only when $r \geq m$ and if $r > m$, some inputs are discarded without loss of generality in the control law, but for the course of this work all systems and/or rigs have $r = m$ in their multivariable versions.

2.6.1 The Interactor Matrix $\xi_T(z)$

A reasonable definition of the interactor matrix $\xi_T(z)$ will be to say it is the multivariable equivalent of the time delay in the SISO systems.

For any full rank mxm transfer matrix $T(z)$ there exists a unique non-singular mxm lower left triangular matrix $\xi_T(z)$ known as the interactor matrix of the form

$$\xi_T(z) = H_T(z)\, \mathrm{diag}[z^{f_1},\ldots,z^{f_m}] \qquad (2.6.1)$$

where

$$H_T(z) = \begin{bmatrix} 1 & \cdot & 0 \\ h_{21}(z) & \cdot \quad \cdot & \\ \vdots & & \\ h_{m1}(z) & h_{m2}(z) \quad \ldots & 1 \end{bmatrix} \quad \text{(unimodular} \qquad (2.6.2)$$

and $h_{ij}(z)$ is divisible by z or is zero,

and $f_i \geq d_i \triangleq \min_{1 \leq j \leq m} d_{ij}$, and d_{ij} is the delay between the jth input and the ith output.

In many cases of interest, $\xi_T(z)$ can be taken to have the form of a diagonal matrix [33], i.e.

Form 1: $\xi(z) = z^{d}I$ where $d = \min_{ij} d_{ij}$

where d is a single delay associated with every output.

Form 2: $\xi(z) = \text{diag}[d_1,\ldots,d_m]$ where $d_i = \min_{j} d_{ij}$

here d_i represents the delay to the ith output.

Also the interactor matrix satisfies

(i) $\det \xi(q) = q^{\bar{m}}$ where $\bar{m}$ is an integer

(ii) $\lim_{z\to\infty} \xi(z)T(z) = K$ where K is a nonsingular matrix

(iii) $\xi(z)^{-1}$ is a stable operator

(iv) For a strictly proper transfer matrix $T(z)$ then f_1 to f_m in (2.6.1) are non-zero.

2.6.2 The MIMO Adaptive Algorithm

From the works in [37-39] it becomes obvious that the interactor matrix must be included in the reference model. So in the course of this work the interactor matrix is taken to be a diagonal matrix consisting of time delays between each input and output.

Considering the system given in (2.5.1), i.e.

$$A(q^{-1})y(t) = B(q^{-1})u(t)$$

Let $$\xi(q) = F(q)A(q^{-1}) + G(q^{-1}) \tag{2.6.3}$$

where $$F(q) = F_0 q^{d'} +\ldots+ F_{d'-1}\, q \tag{2.6.4}$$

$$G(q^{-1}) = G_0 + G_1 q^{-1} +\ldots+ G_{n'}\, q^{-n'} \tag{2.6.5}$$

d' = maximum advance in $\xi(q)$.

Multiplying (2.5.1) by F(q) gives:

$$F(q)A(q^{-1})y(t) = F(q)B(q^{-1})u(t) \tag{2.6.6}$$

using (2.6.3)

$$\begin{aligned}\xi(q)y(t) &= G(q^{-1})y(t) + F(q^{-1})B(q^{-1})u(t) \\ &= \alpha(q^{-1})y(t) + \beta(q^{-1})u(t)\end{aligned} \tag{2.6.7}$$

where $\bar{y}(t) = \xi(q)y(t)$ (2.6.8)

and β_0 is non-singular.

However, the objective is to design an adaptive control law such that $\{y(t)\}$ and $\{u(t)\}$ remain bounded for all time and that $\{y(t)\}$ asymptotically tracks a desired reference model output sequence $\{y_m^*(t)\}$. Hence, defining $\bar{y}_m^*(t) = \xi(q)y_m^*(t)$ the closed-loop system will be characterized by:

$$\begin{bmatrix} \xi(q) & 0 \\ A(q^{-1}) & -B(q^{-1}) \end{bmatrix} \begin{bmatrix} y(t) \\ u(t) \end{bmatrix} = \begin{bmatrix} \xi(q)\, y_m^*(t) \\ 0 \end{bmatrix} \tag{2.6.9}$$

Now let $y(t+d) = \bar{y}(t)$ using assumption of $\xi(q)$, and $y_m^*(t+d) = \bar{y}^*(t)$ where

$$y(t+d) = \begin{bmatrix} y(t+d_1) \\ \vdots \\ y_m(t+d_m) \end{bmatrix} \quad \text{and} \quad y_m^*(t+d) = \begin{bmatrix} y_{m1}^*(t+d_1) \\ \vdots \\ y_{m_m}^*(t+d_m) \end{bmatrix}$$

and defining

$$\begin{bmatrix} e_1(t+d_1) \\ \vdots \\ e_m(t+d_m) \end{bmatrix} = \begin{bmatrix} y_1(t+d_1) \\ \vdots \\ y_m(t+d_m) \end{bmatrix} - \begin{bmatrix} y_{m1}^*(t+d_m) \\ \vdots \\ y_{m_m}^*(t+d_m) \end{bmatrix} \tag{2.6.10}$$

† B_s^* = diagonal polynomial matrix

R_o = upper triangular matrix with zeros in the diagonal

y_1 = system output

U(t)= control input

y_1^m = model output

$y(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix}$ where y_1 = measured system output

y_2 = other measured outputs

$T_1^* A_m^*$ is specified by designer

S^* = a diagonal matrix which contains the Λ zeros of the system

T_2^* is a polynomial Λ - unimodular matrix

while from (2.6.7) and (2.6.8)

$$y(t+d) = \alpha(q^{-1})y(t) + \beta(q^{-1})u(t) \tag{2.6.11}$$

factoring out β_0 from the above, it can be rewritten as (see [16,38])

$$y(t+d) = \beta_0\{u(t) + D(q^{-1})u(t-1) + C(q^{-1})y(t)\} \tag{2.6.12}$$

where

$$D(q^{-1}) = \beta_0^{-1}[\beta(q^{-1}) - I] \tag{2.6.13}$$

$$C(q^{-1}) = \beta_0^{-1}[\alpha(q^{-1})] \tag{2.6.14}$$

Subtracting $y_m^*(t+d)$ from both sides of (2.6.12) gives:

$$e(t+d) = \beta_0\{u(t) + D(q^{-1})u(t-1) + C(q^{-1}y(t) - \beta_0^{-1}y_m^*(t+d)\} \tag{2.6.15}$$

$$\Rightarrow \begin{bmatrix} e_1(t+d_1) \\ \vdots \\ e_m(t+d_m) \end{bmatrix} = \beta_0\left\{u(t) + D(q^{-1})u(t-1) + C(q^{-1})y(t) - \beta_0^{-1}\begin{bmatrix} y_{m_1}^*(t+d_1) \\ \vdots \\ y_{m_m}^*(t+d_m) \end{bmatrix}\right\} \tag{2.6.16}$$

Note the similarity of this to what obtains in [39] where

$$T_2 e_f = B_s^*[u+R_0u+R^*u+S^*y-T_3y_1^m]$$

$$e(t) = y_1(t) - y_1^m(t), \quad e_f(t) = T_1^*A_m^*e(t)$$

where

$$T_1^*A_m^* = I \quad \text{and} \quad T_3y_1^m(t) = y_1^m(t)$$

† For explanation of terms used above *see oposite page*

Equation (2.6.15) can be written in the form:

$$e(t+d) = \beta_0\{u(t) - \Theta_0'^T\phi(t)\} \tag{2.6.17}$$

where θ_o' is an $m \times n'$ matrix consisting of the coefficients in $[C(q^{-1}), D(q^{-1}), \beta_o^{-1}]$, while $\phi(t)$ is an $n' \times 1$ vector given by

$$\phi(t) = [-y(t)^T, -y(t-1)^T, \ldots, -u(t-1)^T, -u(t-2)^T, \ldots, y_m^*(t+d)]$$

with all these components being measurable.

Analogously to the SISO case this results in the following algorithm:

$$\hat{\theta}(t+d)^T = \hat{\theta}(t) - P \begin{bmatrix} e_1(t+d_1) \\ \vdots \\ e_m(t+d_m) \end{bmatrix} [1+\phi(t)^T\phi(t)]^{-1}\phi(t)^T \qquad (2.6.18)$$

and the control law:

$$u(t) = \hat{\theta}(t)^T\phi(t) \qquad (2.6.19)$$

where $d = \max(d_1, \ldots, d_m)$

and P = matrix of constants specified *a priori* (it represents β_o^{-1})

For global convergence of the adaptive algorithm, defining

$$\tilde{\theta}(t+d)^T = \hat{\theta}(t+d)^T - \theta_o'^T \quad \text{and} \quad K = P\beta_o .$$

If K^T+K-K^TK is positive definite then along the trajectories of equations (2.6.18) and (2.6.19)

(a) $\text{trace}[\tilde{\theta}(t+d)^T\tilde{\theta}(t+d)] - \text{trace}[\tilde{\theta}(t)^T\tilde{\theta}(t)] \leq 0$

(b) $\lim_{t\to\infty} \dfrac{e_i(t+d_i)}{[1+\phi(t)^T\phi(t)]^{\frac{1}{2}}} = 0 \qquad 1 \leq i \leq m$

and the algorithm ensures that $\{y(t)\}$, $\{u(t)\}$ are bounded and that

$$\lim_{t\to\infty} |y_i(t) - y_m^*(t)| = 0 \quad \text{for} \quad i = 1, \ldots, m$$

The proof of the above is now shown [11].

(a) Rewriting (2.6.17) using (2.6.19) as

$$e(t+d) = \beta_o \tilde{\Theta}(t)^T \phi(t) \tag{2.6.20}$$

then (2.6.18) becomes by subtracting $\Theta_o'^T$ from both sides:

$$\tilde{\Theta}(t+d)^T = \tilde{\Theta}^T(t) - K\tilde{\Theta}(t)^T\phi(t)[1+\phi(t)^T\phi(t)]^{-1}\phi(t)^T \tag{2.6.21}$$

and

$$\text{trace}(\tilde{\Theta}(t+d)^T\tilde{\Theta}(t+d)) - \text{trace}(\tilde{\Theta}(t)^T\tilde{\Theta}(t))$$

$$= -\text{trace}\left[(K^T+K-[K^TK]\ \frac{\phi(t)^T\phi(t)}{[1+\phi(t)^T\phi(t)]}).(\frac{\tilde{\Theta}(t)^T\phi(t).\phi(t)^T\tilde{\Theta}(t)}{[1+\phi(t)^T\phi(t)]}\ )\right]$$

$$\leq 0 \quad \text{if} \quad K^T+K-K^TK \quad \text{is positive definite.}$$

(b) From the above, it is implied that $\tilde{\Theta}(t+d)^T\Theta(t+d)$ is a bounded non-negative, non-increasing function, hence it converges thus:

$$\lim_{t\to\infty} \text{trace}(K^T+K-[K^TK]\ \frac{\phi(t)^T\phi(t)}{[1+\phi(t)^T\phi(t)]}\ ).(\ \frac{\tilde{\Theta}(t)^T\phi(t).\phi(T)^T\tilde{\Theta}(t)}{[1+\phi(t)^T\phi(t)]}\ )$$

$$= 0$$

Since K^T+K-K^TK is positive definite, then

$$\lim_{t\to\infty} \frac{\tilde{\Theta}(t)^T\phi(t).\phi(t)^T\tilde{\Theta}(t)}{[1+\phi(t)^T\phi(t)]} = 0$$

or, using (2.6.20)

$$\lim_{t\to\infty} \beta_o^{-1}\begin{bmatrix} e_1(t+d_1) \\ \vdots \\ e_m(t+d_m) \end{bmatrix}[1+\phi(t)^T\phi(t)]^{-1}[e_1(t+d_1),\ldots,e_m(t+d_m)]\beta_o^{-1} = 0$$

from which (b) is implied and holds.

2.7 COMMENTS

(a) The key conclusions in this chapter for the Model Reference Adaptive Control Scheme (MRAC) are:

(i) Closed-loop stability is achieved for both SISO and MIMO systems/plants, i.e.bounded inputs, bounded outputs.

(ii) The output tracking error e(t) asymptotically goes to zero as the time becomes very large, implying that perfect tracking of the reference model output will be achieved.

(b) Although the algorithms described in this chapter are globally stable adaptive control schemes, this does not necessarily mean that they are robustly stable. This has been shown in [21]. The reasons for this are not far-fetched as it must be realized that the basic algorithm as it is has been designed for the ideal case where it is assumed that the plant is a Linear Time Invariant (LTI) system whose dynamics can be perfectly described by a model and with no consideration for disturbances. Although Goodwin and Chen [43] have results relating to the adaptive control of LTI systems having purely deterministic disturbances, such as sine waves, biases, etc. This shows that the existing algorithms, i.e. [16,38] can be applied without modification to systems having purely deterministic disturbances.

(c) During the course of this work one of the interesting points noted was the way in which the P-matrix in the MIMO algorithm could be used to decouple a system, as will be seen later on in Chapter 5 on the Coupled Electric Drives rig. Experiments carried out on the Coupled Hydraulic Tanks in Chapter 4 show that an effective choice for the P-matrix is a diagonal constant matrix of reals, which is interesting as this choice does not consider the effects of cross coupling and/or interaction. This seems to support the work of Yang and Lee [44] in which it was found

that MIMO systems were less sensitive to accuracy of models than SISO systems, partly because there are a larger number of parameters being used in the MIMO algorithms.

The robustness problem then leads to the next chapter in which two different Robust algorithms are explained and during their analysis it is shown that they both are quite similar.

Chapter 3

ROBUST MRAC SCHEMES

3.1 INTRODUCTION

In adaptive control the automatic adjustment of feedback laws to control unknown fixed or slowly varying plants is the fundamental or core element of the algorithm. The standard means of obtaining this adaptive control is through the use of a recursive algorithm using the input-output data of the plant to calculate on-line a feedback control input based on this information. For the stability and convergence analysis of the schemes, some assumptions are made, i.e. the controlled system is linear time invariant, of finite order with a known upper bound on its order n, its relative degree n^* and the sign plus magnitude of the high frequency gain k_p. Also assumed in most stability analyses of these algorithms is the absence of noise or disturbances. But these finite dimensional, linear differential or difference equations are mathematical tools which, although they have been quite useful and accurate in the description of physical systems, are merely approximations of these physical systems - these physical systems being infinite dimensioned physical phenomena. Thus, in view of all these, difficulties were bound to crop up in schemes such as MRAC in which the physical system is expected to match the Reference model (Designer specified) of finite dimension under even an ideal situation which is noise and disturbance free. This raised the question of how an adaptive algorithm, which assumes a certain model order, will function when used on a system of higher order, affected by noisiness and disturbances. Work done in this area by Rohrs et al [21], Ioannou and Kokotovic [45] and Egardt [15] showed that unmodelled dynamics or even small bounded disturbances could cause most of the algorithms in [13,14,16,11] to go unstable, even though these algorithms have global stability properties.

Also, in the case of bounded disturbances, another problem faced was the possibility of the control parameters drifting and becoming unbounded, even while the state error between the plant and the model remains bounded, see [46].

The problems raised above led to a lot of researchers investigating the robustness properties of adaptive control schemes in [22-29,47-56] amongst others. By robustness, it means stability of the adaptive control system is guaranteed even in the presence of unmodelled dynamics and bounded disturbances or noise. From the work of various researchers it became clear that robust properties are achievable for some systems without modifications (see Cook and Chen [47], Goodwin and Sin [31], Samson [56] amongst others), provided the magnitude of the disturbance or modelling error is small compared to the magnitude of the reference signal and model used; but clearly the problems above indicated a need for modifications in the adaptive algorithms used so as to make them have robust properties. Amongst the different solutions that came up are (i) normalization, (ii) dead zones, (iii) persistency of excitation, (iv) parameter bounds, and (v) the σ-modification. These will now be discussed in the next section.

3.2 MEANS OF ACHIEVING ROBUSTNESS

As mentioned in the previous section, five of the most favoured ways of achieving robustness in adaptive control are:

(i) Persistency of excitation: This is favoured by people such as Narendra [24], Anderson [49], Kreisselmeier [53] and Cook [47,48]. What this means basically is that for robustness the signals, such as the reference and control inputs have to be persistently exciting so as to nullify the effects of disturbances and unmodelled dynamics present in the system. Various

definitions of a persistently exciting signal are available, but they all generally imply that the signal is defined as persistently exciting if it is [49] a combination of p distinct frequencies (usually of sinusoids) for a model of dimension $2p = n+m$. In some papers a signal referred to as being sufficiently rich implies a persistently exciting signal. Some of the definitions are given in mathematical form below.

(a) From [24] a bounded vector $u : R^+ \to R^n$ is said to be persistently exciting if constants t_o, T_o and ε_o exist such that

$$\frac{1}{T_o}\int_{t}^{T+t_o} u^T(\tau)\omega d\tau \geq \varepsilon_o \qquad \forall\ t \geq t_o \tag{3.2.1}$$

for all unit vectors $\omega \in R^n$

OR

a bounded vector $u : R^+ \to R^n$ is said to be persistently exciting if positive constants T_o, δ_o and ε_o exist such that a t_2 exists with $[t_2, t_2+\delta_o] \subset |t, t+T_o|$ and

$$\left|\frac{1}{T_o}\int_{t_2}^{t_2+\delta_o} u^T(\tau)\omega d\tau\right| \geq \varepsilon_o \qquad \forall\ t \geq t_o \tag{3.2.2}$$

for all unit vectors $\omega \in R^n$

(b) From [31,49] another definition is, a scalar input $\{u(t)\}$ is said to be strongly persistently exciting of order n if, for all t, there exists an integer ℓ such that

$$\rho_1 I > \sum_{k=t}^{t+\ell} \begin{bmatrix} u(k+n) \\ \vdots \\ u(k+1) \end{bmatrix} [u(k+n),\ldots,u(k+1)] > \rho_2 I \tag{3.2.3}$$

where $\rho_1, \rho_2 > 0$.

A point of interest is that Astrom [57] stated that part of the reasons for the results in [21] was the fact that the reference signal used was not exciting enough to cope with the disturbances and error used, i.e. for

constant/step reference inputs the persistency is of order 1, which is not sufficiently exciting to cope with, say, sinusoidal disturbances which are of order 2. He also points out that unmodelled dynamics will not cause any difficulties if the frequency of the reference signal is sufficiently low and sufficiently large in magnitude to overcome destabilizing effects from high frequency inputs or noise.

A major criticism of the requirements of persistency of excitation is that the disturbances must be bounded below the exciting signal, otherwise the disturbance can counteract the excitation and this leads to the persistency of excitation being reviewed continuously to make sure this does not happen, as there is no specific or standard persistently exciting signal that can cope with all disturbances.

Also, for direct adaptive control, where exact identification of plant parameters is not essential, the persistency of excitation condition can be avoided.

(ii) Dead Zones: Although there are different approaches to this, it basically means that within certain limits/bounds/thresholds *a priori* defined, the adaptation mechanism is switched off. It is mainly used to tackle bounded disturbances, preventing instability by elimination of the integral action embedded within the adaptive laws. The logic behind the dead zone being that when the tracking error is small, compared to a known upper bound of the amplitude of the disturbance say, the error is no longer useful in bringing out information for the adaptive control laws. Thus, to continue using this tracking error within the dead zone region might be to have the disturbance influencing the adaptation which will have the effect of deteriorating the control parameters and consequently affect the adaptive controller as well. The choice of dead zone is crucial for stability as the larger the dead zone, the shorter the period of time for adaptation to take place and

thus the larger the output and parameter errors. Smaller dead zones lead to smaller output errors and examples of these are shown in Chapter 4. Thus, for a good choice of dead zone, use must be made of *a priori* knowledge of the plant parameters and magnitude of the disturbance. In some papers, effects of unmodelled dynamics have been manoeuvred to represent disturbances for which dead zones can be applied, (see, for example, [46,52,27]).

(iii) Normalization: The use of a normalizing signal allows the unmodelled dynamics to be characterised in the form of noise to signal ratio. Thus, it bounds the modelling error (see Praly [29,54]) and causes the adaptive law to see the effects of unmodelled dynamics as a bounded disturbance. Normalizing signals play an important role during the transient by guaranteeing slow adjustment of the controller parameters where appropriate. The smooth adjustment of the controller parameters reduces the excitation of the unmodelled dynamics by the control input and therefore improves the accuracy of adaptation. Examples of normalizing signals used in the literature are:

From [22] $$\dot{m} = -\delta_0 m + \delta_1(|u|+|y|+1), \quad m(0) \geq \delta_1/\delta_0 \qquad (3.2.4)$$

where δ_0, δ_1 are positive design parameters and
u = input, y = output of system, respectively.

From [27] $$\left.\begin{aligned} N(t) &= \gamma_0 + m(t) \\ m(t) &= \sigma_0 m(t-1) + |u(t-1)|+|y(t-1)| \\ 0 < \sigma_0 &< 1, \quad m(0) \geq 0 \end{aligned}\right\} \qquad (3.2.5)$$
γ_0 is designer chosen.

From [28] $$\rho(t) = \mu\rho(t-1) + \max(||\phi(t-d)||^2, \rho) \qquad (3.2.6)$$
$$\rho > 0, \ \mu \in (0,1), \ \phi(t) = (u(t),\ldots,u(t-n_S), y(t),\ldots,y(t-n_R))^T$$

From [29] $$S(t) = \sigma S(t-1) + \max\{|y(t-1)|+|u(t-d)|, S\} \qquad (3.2.7)$$
$$0 < \sigma < 1, \quad S > 0.$$

All the above can be seen to be of the same form, and later on in Section 3.5 will be shown to be similar.

(iv) Parameter Bounds: Forms of this include [27,29] in which it is proposed to keep the estimated parameters inside a chosen sphere using a projection, while others are of the form in [55,56] where an upper bound M_θ for the norm of the controller parameter vector is defined. In the first case, adaptation takes place only within the chosen set, while in the second adaptation goes on so long as the norm of the parameter vector does not exceed M_θ. Compared to the dead zone modification, this modification guarantees zero residual tracking error and potential convergence of parameters to their true values in the ideal case; but the first of the forms, i.e. "projection sphere" is the best as it eliminates the possibility of burst phenomena associated with parameter drifts as in [27].

(v) The σ-modification: This was first suggested by Ioannou and Kokotovic [45]. It consists of adding an extra term $-\sigma\theta$, $\sigma > 0$ to the adaptive law equation (see also [22]). It is shown that the σ-modification guarantees existence of a large region of attraction from which all signals are bounded and the tracking error converges to a small residual set. A new form of this modification has been proposed recently by Narendra and Annaswamy [58] in which σ is replaced by a term proportional to $|e|$ where e is the output tracking error. One of the advantages of the σ-modification is that it requires no new *a priori* information in its design. It also requires no persistently exciting signals, but it is in continuous-time.

In different papers the various modifications above have been incorporated to the adaptive algorithms to make it more robust. In some cases the modification has been limited to only one, i.e. dead zone in [46] or parameter bounds in [55], or normalization in [28], while in others it has been a combination of two or more of the above, i.e. [27,29,56,59]. In the course of this work, focus was on two particular modified algorithms, namely those

suggested by Kreisselmeier and Anderson [27] and Ortega et al [28], which are both in discrete-time, hence suitable for use on a microcomputer. The description and analysis of the two algorithms now follow in Sections 3.3 and 3.4, respectively.

3.3 THE KREISSELMEIER AND ANDERSON ALGORITHM [27]

This algorithm makes use of a normalizing signal, together with a relative dead zone and a parameter projection in the adaptive control law to achieve robust stability of the scheme with respect to unmodelled plant uncertainties and disturbances. In general, parameter estimates using the scheme do not converge to their true values, even when no modelling errors are present due to the use of a relative dead zone, but will converge to a neighbourhood of the true values. *A priori* information needed includes knowledge of bounds on the plant and controller parameters plus bounds on the modelling error. A description of the algorithm and proof of the robustness now follows.

The plant or system is represented by the equations

$$A(z^{-1})y(t) = B(z^{-1})u(t) + \eta(t) \qquad (3.3.1)$$

where $A(z^{-1}) = 1 + a_1 z^{-1} + \ldots + a_n z^{-n}$

$B(z^{-1}) = b_0 z^{-d}(1 + b_1 z^{-1} + \ldots + b_m z^{-m}$

$= b_0 z^{-d} \tilde{B}(z^{-1})$; $m = n-d$

$u(t), y(t)$ are the measurable input and output respectively, and z^{-1} denotes the unit delay operator; d is the system time delay with $n \geq d > 0$.

$\eta(t) \rightarrow$ modelling error $= A(z^{-1})y(t) - B(z^{-1})u(t)$ (3.3.2)

(also contains the disturbances).

The reference model is chosen as:

$$A_m(z^{-1})y_m(t) = z^{-d}b_m r(t) \tag{3.3.3}$$

where $b_m > 0$

$A_m(z^{-1}) = 1 + a_{m,1}z^{-1} + \ldots + a_{m,d}$; a strictly stable polynomial

and $r(t) \rightarrow$ reference input which is bounded in the modulus by a known constant.

3.3.1 The Model Reference Adaptive Controller and Adaptive Law

Defining

$$\underline{q}^T(z^{-1}) = [z^{-1}, z^{-2}, \ldots, z^{-n}] \tag{3.3.4}$$

$$\left.\begin{aligned} \underline{v}_1(t) &= \underline{q}(z^{-1})u(t) \\ \underline{v}_2(t) &= \underline{q}(z^{-1})y(t) \end{aligned}\right\} \tag{3.3.5}$$

$$\left.\begin{aligned} \underline{v}^T(t) &= [r(t), \underline{v}_1^T(t), \underline{v}_2^T(t)] \\ \underline{\theta}^T(t) &= [\theta_0(t), \underline{\theta}_1^T(t), \underline{\theta}_2^T(t)] \end{aligned}\right\} \tag{3.3.6}$$

then the control law is chosen as:

$$u(t) = \underline{v}^T(t)\,\underline{\theta}(t) \tag{3.3.7}$$

(Notice the similarity of this to the basic algorithm control law in Chapter 2, i.e. (2.3.14)).

Also defining

$$m(t) = \sigma_0 m(t-1) + |u(t-1)| + |y(t-1)| \tag{3.3.8}$$

where $0 < \sigma_o < 1$, $m(0) \geq 0$

then the modelling error $\eta(t)$ is said to be relatively bounded if there is a finite $\mu \geq 0$ and $m(0) \geq 0$ such that

$$\eta(t) \leq \mu m(t) \tag{3.3.9}$$

Proof:

Let the true plant satisfy $y(t) = f(\underline{\xi}(t),t)$ (3.3.10)

where $\underline{\xi}(t) = [y(t-1),\ldots,y(t-\bar{n}), u(t-1),\ldots,u(t-\bar{n})]^T$ (3.3.11)

and $\bar{n} \geq n$

Defining

$$C(z^{-1}) = 1 + c_1 z^{-1} + \ldots\ c_{\bar{n}-n} z^{n-\bar{n}} \tag{3.3.12a}$$

an arbitrary polynomial with

$$\frac{1}{C(z^{-1})} = \sum_{i=0}^{\infty} \gamma_i z^{-i} \tag{3.3.12b}$$

and property:

$$\sum_{i=0}^{\infty} |\gamma_i|\ \sigma_o^{-i} = \bar{\gamma} < \infty \tag{3.3.12c}$$

i.e. $C(z_i^{-1}) = 0$ implies $|z_i| < \sigma_o$,

combining (3.3.2) and (3.3.10), using $C(z^{-1})$, gives:

$$C(z^{-1})\eta(t) = C(z^{-1})[A(z^{-1})y(t)-B(z^{-1})u(t)]$$

$$= y(t) - \underline{p}^T \underline{\xi}(t)$$

where $\underline{p}$ is a constant vector containing the coefficients of the polynomials

$C(z^{-1})A(z^{-1}) - 1$ and $C(z^{-1})B(z^{-1})$

$$= f(\underline{\xi}(t),t) - \underline{p}^T \underline{\xi}(t) \qquad (3.3.13)$$

Now let

$$|f(\underline{\xi},t) - \underline{p}^T \underline{\xi}| \leq \varepsilon_f ||\underline{\xi}||_{\sigma_o} \quad \text{for all } \underline{\xi} \text{ and } t \geq 0$$

where $||\underline{\xi}||_\sigma = \sum_{i=1}^{\bar{n}} \sigma^{i-1}(|\xi_i| + |\xi_{i+\bar{n}}|)$ and $\sigma \in (0,1]$

Also, assuming $m(0) \geq ||\xi(0)||_{\sigma_o}$ means that $m(t) \geq ||\underline{\xi}(t)||_{\sigma_o}$ for all $t \geq 0$, and it follows that

$$|\eta(t)| = |C^{-1}(z^{-1})[f(\underline{\xi},t) - \underline{p}^T \underline{\xi}(t)]|$$

$$= |\sum_{i=0}^{\infty} \gamma_i [f(\underline{\xi}(t-i),t-i) - \underline{p}^T \underline{\xi}(t-i)]|$$

$$\leq \varepsilon_f \sum_{i=0}^{\infty} |\gamma_i| \cdot ||\underline{\xi}(t-i)||_{\sigma_o}$$

$$\leq \varepsilon_f \sum_{i=0}^{\infty} |\gamma_i| \cdot m(t-i)$$

$$\leq \varepsilon_f \sum_{i=0}^{\infty} |\gamma_i| \cdot \sigma_o^{-i} m(t)$$

$$\leq \varepsilon_f \bar{\gamma} m(t) = \mu m(t) \; ; \qquad \text{hence proved.}$$

Define $\underline{\Theta}^*$ a constant parameter vector $= [\Theta_o^*, \underline{\Theta}_{-1}^{*T}, \underline{\Theta}_{-2}^{*T}]^T$ where $\Theta_o^* = b_m/b_o$ and $\underline{\Theta}_{-1}^*, \underline{\Theta}_{-2}^*$ are such that

$$[1 - q^T(z^{-1})\underline{\Theta}_{-1}^*]A(z^{-1}) - B(z^{-1})q^T(z^{-1})\underline{\Theta}_{-2}^* = A_m(z^{-1})\tilde{B}(z^{-1})$$

and let $\underline{\tilde{\Theta}}(t) = \underline{\Theta}(t) - \underline{\Theta}^*$, then the model reference objective is said to be satisfied if $\tilde{\Theta}(t) = \underline{0}$ and $\eta(t) = 0$, i.e. if $\underline{\Theta}(t) = \underline{\Theta}^*$ and $\mu = 0$.

The adaptive law containing the dead zone and projection to achieve this objective using normalization is now defined.

Defining

$$\underline{w}^T(t) = [(1/b_m)z^d A_m(z^{-1})y(t), \underline{v}_1^T(t), \underline{v}_2^T(t)] \tag{3.3.14}$$

the identification error:

$$E(t) = \underline{w}^T(t-d)\underline{\theta}(t) - \underline{v}^T(t-d)\underline{\theta}(t-d) \tag{3.3.15}$$

and normalizing factor $N(t) = \gamma_o + m(t)$; $\gamma_o > 0$ (3.3.16)

then the relative error $E_1(t) = E(t)|N(t)$ (3.3.17)

The adaptive control parameters are then chosen using the following adaptive algorithm in two steps:

Step I $$\hat{\underline{\theta}}(t+1) = \underline{\theta}(t) - \frac{w(t-d)N(t)D(E_1(t))}{\gamma_1 + \underline{w}^T(t-d)\underline{w}(t-d)} \tag{3.3.18a}$$

where $D(E_1(t))$ the relative dead zone (see Figure 3.1) is defined as:

$$D(E_1) = \begin{cases} 0 & \text{if } |E_1| \leq d_o \\ E_1 - d_o & \text{if } E_1 > d_o \\ E_1 + d_o & \text{if } E_1 < d_o \end{cases} \tag{3.3.18b}$$

Step II

$$\theta_i(t+1) = \begin{cases} \hat{\theta}_i(t+1) & \text{if } \theta_{i,min} \leq \hat{\theta}_i(t+1) \leq \theta_{i,max} \\ \theta_{i,min} & \text{if } \hat{\theta}_i(t+1) < \theta_{i,min} \\ \theta_{i,max} & \text{if } \hat{\theta}_i(t+1) > \theta_{i,max} \end{cases} \tag{3.3.18c}$$

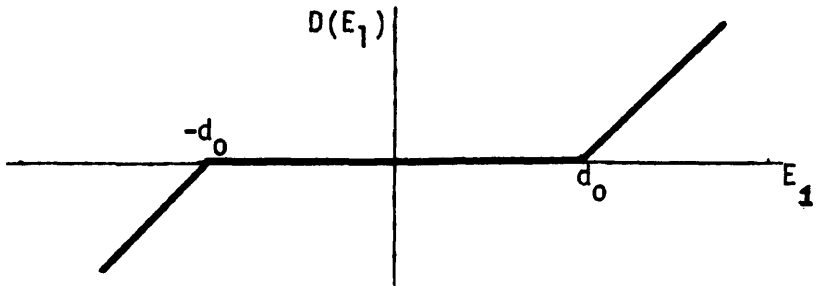


Figure 3.1 The dead zone function $D(E_1)$

Again, as in the Basic MRAC algorithm (see Section 2.4.2), the adaptive parameter algorithm used here has the following properties:

(i) $||\tilde{\theta}(t)|| \leq$ constant where $\tilde{\underline{\theta}}(t) = \underline{\theta}(t) - \underline{\theta}^*$

(ii) $\lim_{t\to\infty} D(E_1(t)) = 0$

(iii) $\lim_{t\to\infty} [\tilde{\theta}(t+1) - \tilde{\theta}(t)] = 0$

Note that the above properties are quite similar to those in Section 2.4.2 with their proofs being derived in a similar manner. Hence the proofs of these will not be given here (see [27,31]).

3.3.2 Proof of Robust Stability [27]

To establish the robust stability of the closed-loop adaptive control system, a bound on $N(t+1)|N(t)$ is derived first.

From (3.3.8), (3.3.16):

$$N(t+1) = \sigma_0 N(t) + \gamma_0(1-\sigma_0) + |y(t)| + |u(t)| \qquad (3.3.19)$$

Rewriting (3.3.2), using (3.3.5), it becomes:

$$y(t) = \underline{b}^T \underline{v}_1(t) - \underline{a}^T v_2(t) + \eta(t) \tag{3.3.20}$$

expanding (3.3.7):

$$u(t) = \theta_0(t)r(t) + \underline{\theta}_1^T(t)\underline{v}_1(t) + \underline{\theta}_2^T(t)\underline{v}_2(t) \tag{3.3.21}$$

(3.3.6), (3.3.8), (3.3.16) imply:

$$||\underline{v}_1(t)|| + ||\underline{v}_2(t)|| \leq n\,\sigma_-^{-n}\,N(t) \tag{3.3.22}$$

for some finite $t_2 \geq t_1$, it follows that for $t \geq t_2$

$$\begin{aligned} N(t+1) &= [\sigma_0 + \sigma_0^{-n} k_1 + \mu]N(t) + \gamma_0(1-\sigma_0) + |\theta_0(t)r(t)| \\ &\leq [1 + \sigma_0^{-n} k_1 + \mu]N(t) + |\theta_0(t)r(t)| \\ &\leq [1 + \sigma_0^{-n} k_1 + d_0 k_0 + k_2]N(t) = \Lambda(d_0)N(t) \end{aligned} \tag{3.3.23}$$

where $$\mu_0 = \frac{d_0}{\sum_{i=0}^{d} |p_i|\sigma_0^{-i}} \tag{3.3.24}$$

and $$k_0 \geq \frac{1}{\sum_{i=0}^{d} |p_i|\sigma_0^{-i}} \tag{3.3.25}$$

$$k_2 \geq \sup_{t \geq t_2} \{\theta_0(t)r(t)|N(t)\} \tag{3.3.26}$$

Rewriting the input and output of the closed-loop system to facilitate analysis of closed-loop stability in terms of $m(t)$ implies:

$$y(t) = [b_m|A_m(z^{-1})]\{r(t-d) + E_2(t)N(t)|\theta_0(t)\} \tag{3.3.27}$$

where $E_2(t) = [\underline{v}^T(t-d)\tilde{\theta}(t-d) + \eta_1(t)]\theta_o(t)|[\theta_o^* N(t)]$

with $\eta_1(t) = P(z^{-1})\eta(t)$; $P(z^{-1})$ as defined in [27],

while

$$u(t) = \frac{A(z^{-1})}{B(z^{-1})} \cdot \frac{b_m z^{-d}}{A_m(z^{-1})} \cdot \left\{ r(t) + E_3(t+d)N(t+d)\left[\frac{1}{\theta_o^*} + \frac{1}{\theta_o(t+d)}\right]\right\} + \frac{\underline{q}^T(z^{-1})\underline{\theta}_2^*}{A_m(z^{-1})\tilde{B}(z^{-1})} n(t) \qquad (3.3.28)$$

where $E_3(t) = [\underline{v}^T(t-d)\tilde{\theta}(t-d)]/\{N(t)[1+\theta_o^*/\theta_o(t)]\}$

Let the impulse responses of b_m/A_m, $b_m z^{-d}A/A_m B$ and $\underline{q}^T\underline{\theta}_2^*/A_m\tilde{B}$ all be bounded in modulus by $K_c\sigma_c^t$ where $0 \leq \sigma_c < 1$. Suitable constants K_0, K_1, K_2, K_c, σ_c can be obtained using available prior information. Then noting that $r(t)$ is bounded by a constant, that for arbitrary $d_1 > d_o$ there exists $T \geq t_2$ such that $E_2(t) \leq d_1$ and $E_3(t+d) \leq d_1$ whenever $t \geq T$ and that $N(t)$, $(t \leq T)$ is bounded, then for $t \geq T$ equation (3.3.27) reduces to:

$$|y(t)| \leq \text{const} + \sum_{i=T}^{t} K_c\sigma_c^{t-i}\, d_1 N(i)/\theta_o(i)$$
$$\leq \text{const} + (1/\theta_{o,\min})K_c d_1 \sum_{i=T}^{t} \sigma_c^{t-i}\, m(i) \qquad (3.3.29)$$

and (3.3.27) becomes:

$$|u(t)| \leq \text{const} + \sum_{i=T}^{t} K_c\sigma_c^{t-i}\{d_1 N(i+d)[1/\theta_o^* + 1/\theta_o(i+d)] + \mu\, N(i)\}$$
$$\leq \text{const} + K_c[2d_1\Lambda^d(d_o)/\theta_{o,\min} + d_o K_o] \sum_{i=T}^{t} \sigma_c^{t-i} m(i) \qquad (3.3.30)$$

where $N(t+d) \leq \Lambda^d(d_o)N(t)$ and $\mu \leq \mu_o \leq d_o K_o$

Equations (3.3.16), (3.3.29), (3.3.30) then yield:

$$m(t+1) \leq \sigma_o m(t) + K_o + K_m(d_1) \sum_{i=T}^{t} \sigma_c^{t-i}\, m(i) \qquad (3.3.31)$$

where K_o is a constant

and $$K_m(d_1) = d_1 K_c\{[1+2\Lambda^d(d_1)]/\theta_{o,min} + K_o\} \qquad (3.3.32)$$

Defining

$$\bar{m}(t+1) = \sigma_o\bar{m}(t) + K_o + K_m(d_1) \sum_{i=T}^{t} \sigma_c^{t-i}\, \bar{m}(i) \; ; \quad (t \geq T) \qquad (3.3.33)$$

it follows by comparison to (3.3.31) that if $\bar{m}(T) \geq m(T)$ then $\bar{m}(t) \geq m(t)$ for all $t \geq T$, i.e. $\bar{m}(t)$ is an upper bound of $m(t)$.

Equation (3.3.33) can also be written as:

$$(z-\sigma_o)\bar{m}(t) = K_o + \{K_m(d_1)/(1 - \sigma_c z^{-1})\}\bar{m}(t) \qquad (3.3.34)$$

which is equivalent to a feedback configuration with a closed-loop characteristic equation:

$$1 + [\sigma_o + \sigma_c + K_m(d_1)]z^{-1} + \sigma_o\sigma_c z^{-2} = 0 \qquad (3.3.35)$$

from which stability is obtained if

$$K_m(d_1) < (1 - \sigma_o)(1 - \sigma_c)$$

where σ_o, σ_c lie in $(0,1)$ and $K_m(d_1)$ is positive. This gives rise to the conclusion that if the dead zone is chosen so that $d_o > 0$ and $K_m(d_o) < (1 - \sigma_o)(1 - \sigma_c)$ then, since $K_m(\cdot)$ is a continuous function, there always exists $d_1 > d_o$ satisfying (3.3.35) which implies stability of the system. Also, since $y(t) - y_m(t) = [b_m/A_m(z^{-1})]\{E_2(t)N(t)/\theta_o(t)\}$ which is the output error, where $E_2(t)$ is known to converge to the interval $[-d_o, d_o]$, then the relative output error $[y(t) - y_m(t)]/N(t)$ will be proportional to the size of the dead zone.

3.4 THE ORTEGA, PRALY AND LANDAU ALGORITHM [28]

This algorithm is based on the use of normalization in parameter estimation for the achievement of robustness. The adaptive control system is assumed to be affected by reduced order modelling and bounded output disturbances. The description of the algorithm now follows.

The plant is described by:

$$A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t) + d(t) \tag{3.4.1}$$

where $A(q^{-1})$ and $B(q^{-1})$ are polynomials in q^{-1} with

$$A(q^{-1}) = 1 + a_1 q^{-1} + \ldots + a_{n_a} q^{-n_a} \tag{3.4.2a}$$

$$B(q^{-1}) = b_0 + b_1 q^{-1} + \ldots + b_{n_b} q^{-n_b} \tag{3.4.2b}$$

and $u(t)$, $y(t)$, $d(t)$ are the input, output and disturbance sequences. Note also that $d(t)$, the disturbance, also includes the effects of unmodelled dynamics.

Defining $S(t,q^{-1})$ and $R(t,q^{-1})$ as two polynomial functions in q^{-1} of degrees n_S and n_R respectively, with time varying coefficients, i.e.

$$\left.\begin{aligned} S(t,q^{-1}) &= s_0(t) + s_1(t)q^{-1} + \ldots + s_{n_S}(t)q^{-n_S} \\ R(t,q^{-1}) &= r_0(t) + r_1(t)q^{-1} + \ldots + r_{n_R}(t)q^{-n_R} \end{aligned}\right\} \tag{3.4.3}$$

Let the reference model be defined as:

$$E(q^{-1})y_m(t) = q^{-d}H(q^{-1})r(t) \tag{3.4.4}$$

where $r(t)$ is the reference input, $y_m(t)$ the reference model output and d is the plant time delay.

For this algorithm the objective is to minimize the filtered tracking error

$$e(t) = C_R(q^{-1})[y(t) - y_m(t)] \tag{3.4.5}$$

where $C_R(q^{-1})$ is a stable polynomial, the roots of which are the desired closed-loop poles.

The approach used here is similar to that in pole-assignment in which one solves

$$R(t,q^{-1})y(t) + S(t,q^{-1})u(t) = C_R(q^{-1})y_m(t+d) \tag{3.4.6}$$

to obtain the control law, but

$$R(t,q^{-1})y(t) + S(t,q^{-1})u(t) = \theta^T(t)\phi(t) \tag{3.4.7}$$

where $$\theta(t) = [s_0(t),\ldots,s_{n_S}(t),r_0(t),\ldots,r_{n_R}(t)]^T \tag{3.4.8}$$

and $$\phi(t) = [u(t),\ldots,u(t-n_S),y(t),\ldots,y(t-n_R)]^T \tag{3.4.9}$$

thus $$\theta^T(t)\phi(t) = C_R(q^{-1})y_m(t+d) \tag{3.4.10}$$

from which the control input is derived as:

$$u(t) = \frac{1}{s_0(t)}\,[-s_1(t)u(t-1),-\ldots,-s_{n_S}u(t-n_S)-r_0(t)y(t),-\ldots \ldots-,r_{n_R}(t)y(t-n_R) + C_R(q^{-1})y_m(t+d)] \tag{3.4.11}$$

Note once again the similarity of this control input to that of the Goodwin et al projection algorithm 1 [16] which was mentioned in Chapter 2, but here the parameter adaptive algorithm is different from that in [16]. This is now described.

3.4.1 The Parameter Adaptive Algorithm Including Normalization

Defining $\rho(t)$ the normalization factor,

$$\rho(t) = \mu\rho(t-1) + \max(|\phi(t-d)|^2, \rho) \; ; \quad \rho > 0, \; \mu \in (0,1) \qquad (3.4.12)$$

where ρ is a small positive constant that defines a lower bound to $\rho(t)$. The choice of time constant μ being a compromise between algorithm alertness and robustness (see Chapter 4). Boundedness of $\phi(t)$ is ensured if there are $\phi(t)$ independent properties which are obtainable when $e(t)$ and $\phi(t)$ are normalized. Normalized signals will be denoted by $\bar{(\cdot)}$ and are defined as:

$$\bar{\phi}(t-d) \triangleq \rho(t)^{-\frac{1}{2}} \phi(t-d) \qquad (3.4.13a)$$

$$\bar{e}(t-d) \triangleq \rho(t)^{-\frac{1}{2}} e(t-d) \qquad (3.4.13b)$$

In [28] two algorithms are suggested for implementation:

1. Constant Gain (C.G.) parameter algorithm given by:

$$\Theta(t) = \Theta(t-d) + F \, \bar{\phi}(t-d)\bar{e}(t) \qquad (3.4.14)$$

where F is a scalar constant > 0.

2. Regularized Least Squares (R.L.S.) parameter algorithm given by:

$$\Theta(t) = \Theta(t-d) + F \, \phi(t-d)\bar{e}(t) \qquad (3.4.15a)$$

where F is a time varying matrix defined as:

$$F = F(t) = (1-\lambda_0/\lambda_1)\left[F(t-d) - \frac{F(t-d)\bar{\phi}(t-d)\bar{\phi}^T(t-d)F(t-d)}{\lambda + \bar{\phi}^T(t-d)F(t-d)\bar{\phi}(t-d)}\right] + \lambda_0 I \tag{3.4.15b}$$

and $\lambda_0 < \lambda_1$; $\lambda_0, \lambda_1, \lambda$ being strictly positive scalars.

The eigenvalues of $F(t)$ are contained in the chosen interval $[\lambda_0, \lambda_1]$.

Bearing in mind the simplicity of the algorithm required for this work, the constant gain parameter adaptive algorithm was chosen.

The proof of the robust stability of the algorithm now follows.

3.4.2 The Proof of Robust Stability [28]

The robust stability of the adaptive control algorithm stated above is based on sector stability theorem by Zames [33], which involves conic sectors. This states that the feedback interconnection of two conic bounded operators is globally stable if one is strictly inside a cone and the inverse of the other one outside it. Thus, conic sector conditions must be established for the adaptive algorithm given above. To achieve this a normalized error model is derived. Direct application of the sector stability theorems to this model then leads to the derivation of stability conditions for the normalized signals. These conditions are then translated to the original signals and operators. Basically stability of the adaptive system reduces to proving that the regressor vector $\phi(t)$ is bounded which ensures that the normalization factor is used as a multiplier. L_2-stability, implying tracking error cancellation for reference inputs and disturbances that are L_2 signals while L_∞ stability is for arbitrary reference inputs and bounded output disturbances, will be considered.

Error Equations:

Defining the polynomial

$$C = S^*A + q^{-d} R^*B \tag{3.4.16}$$

where R^*, S^* are constant coefficient polynomial functions in q^{-1} similarly defined as those in (3.4.3), i.e.

$$\left.\begin{aligned} R^* &= r_o^* + r_1^* q^{-1} + \ldots + r_{n_R}^* q^{-n_R} \\ S^* &= s_o^* + s_1^* q^{-1} + \ldots + s_{n_S}^* q^{-n_S} \end{aligned}\right\} \qquad (3.4.17)$$

and also a polynomial coefficient vector

$$\theta^* \triangleq [s_o^*, s_1^*, \ldots, s_{n_S}^*, r_o^*, r_1^*, \ldots, r_{n_R}^*] \qquad (3.4.18)$$

With the above it is assumed that there exists a non-empty set Θ_{LS} defined as:

$$\Theta_{LS} \triangleq \{\theta^* \in R^n : C(q) \neq 0, \forall\, q \in C, |q| > \mu^{\frac{1}{2}}\} \neq 0 \qquad (3.4.19)$$

where $n = n_S + n_R + 2$ and $\mu \in (0,1)$.
The above assumption will be known as Assumption 1.

Combining (3.4.19) with (3.4.1), using (3.4.9) and manipulating,

$$Cy(t) = B\,\theta^{*T}\phi(t-d) + S^* d(t) \qquad (3.4.20a)$$

$$Cu(t) = A\,\theta^{*T}\phi(t) - R^*\,d(t) \qquad (3.4.20b)$$

Defining

$$\psi(t) = (\theta(t-d) - \theta^*)^T\phi(t-d)$$

$$= \tilde{\theta}^T(t-d)\phi(t-d) \qquad (3.4.21)$$

$\tilde{\theta}(t) = \theta(t) - \theta^*$; the difference between stabilizing and actual parameters.

Using (3.4.5),(3.4.10,(3.4.20) and (3.4.21), the error model can be expressed as:

$$e(t) = -H_2\,\psi(t) + e^*(t) \tag{3.4.22a}$$

where $$e^*(t) = (H_2-I)C_R y_m(t) + C_R C^{-1} S^* d(t) \tag{3.4.22b}$$

$$H_2 = C_R C^{-1} B \tag{3.4.22c}$$

Figure 3.2 shows the complete error model with H_1 denoting a relationship defined by the adaptive algorithm, i.e. $H_1 : e(t) \rightarrow \psi(t)$.

As mentioned earlier, all these signals and operators are normalized for stability analysis, i.e.

$$\bar{\psi}(t) = \rho(t)^{-\frac{1}{2}}\,\psi(t) \tag{3.4.23}$$

$$\bar{H}_i = \rho(t)^{-\frac{1}{2}}\,H_i[\rho(t)^{\frac{1}{2}}] \;; \quad i = 1,2, \tag{3.4.24}$$

plus the error $\bar{e}(t)$ and regressor vector $\bar{\phi}(t)$ as shown earlier. It is assumed that $\rho(t)$ is such that $||\bar{\phi}(t)||_\infty \leq 1$. Stability of $\psi(t)$ assumes that the regressor vector $\phi(t)$ is bounded and consequently $\rho(t)$ is bounded under L_2-stability (see [28]) using multiplier theory.

Equations (3.4.14),(3.4.15) define an operator $\bar{H}_1 : \bar{e}(t) \rightarrow \bar{\psi}(t)$, in addition to which for the RLS/PAA its exponentially weighted counterpart $\bar{H}_1^\alpha : \bar{e}^\alpha(t) \rightarrow \bar{\psi}^\alpha(t)$, where the superscript (α) denotes

$$X^\alpha(t) \triangleq \alpha^t\,X(t) : \quad \alpha > 0$$

will be considered.

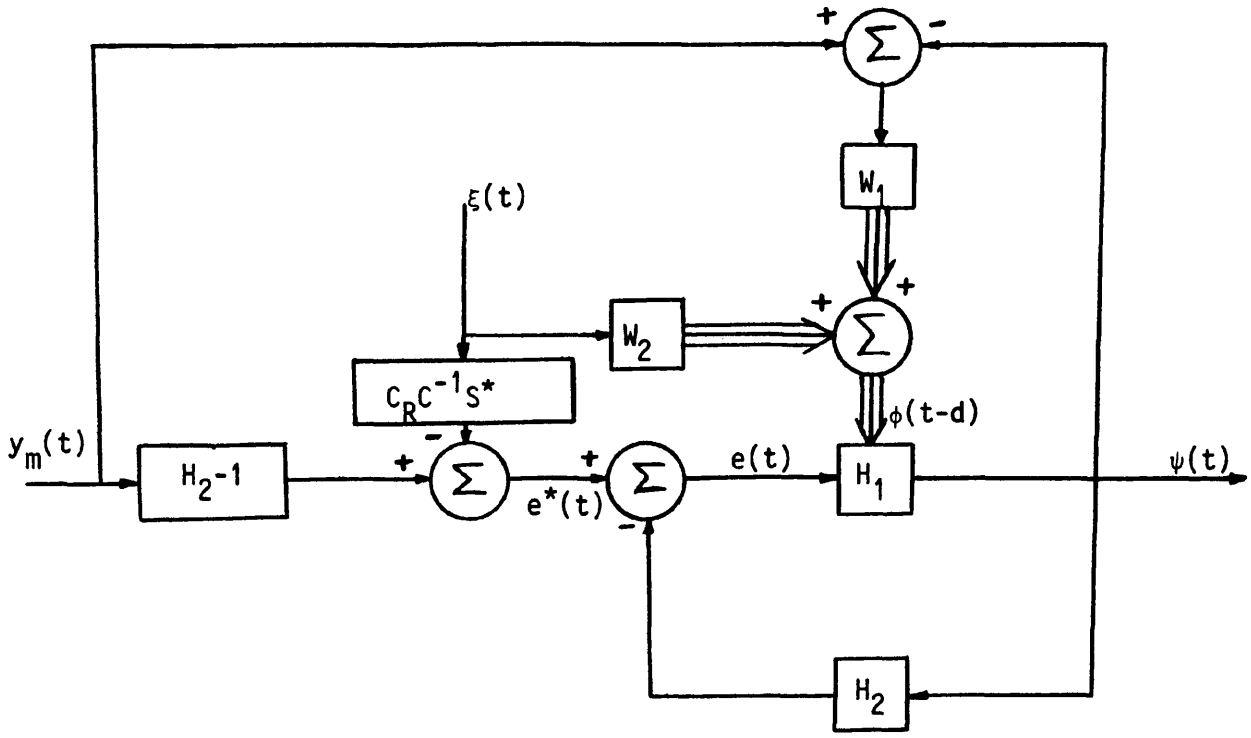


Figure 3.2 The Error Model.

The input-output properties of these two operators can be summarized in the following lemma:

Lemma:

1. For the Constant Gain parameter algorithm, if F is given by (3.4.14), then

$$\bar{H}_1 + \frac{1}{2}\bar{\sigma}_{CG} \quad \text{is passive}$$

for all $\bar{\sigma}_{CG}$ such that

$$\bar{\sigma}_{CG} \geq F||\bar{\phi}^T(t)\bar{\phi}(t)||_\infty$$

2. For the Recursive Least Squares parameter algorithm, if F is given by (3.4.15a), (3.4.15b), then

$$\bar{H}_1^{\alpha}(t) \text{ is outside CONE } (-1, \sqrt{1-\bar{\sigma}_{RLS}})$$

for α verifying

$$\lambda \max\left[F(t)^{-1}\left(F(t-d) - \frac{F(t-d)\bar{\phi}(t-d)\bar{\phi}^T(t-d)F(t-d)}{\lambda + \bar{\phi}^T(t-d)F(t-d)\bar{\phi}(t-d)}\right)\right]\cdot\alpha^{2d} \leq 1$$

and all $\bar{\sigma}_{RLS}$ satisfying

$$\bar{\sigma}_{RLS} \geq \frac{\lambda_1\bar{\phi}^T(t)\bar{\phi}(t)}{\lambda+\lambda_1\bar{\phi}^T(t)\phi(t)}$$

The proof of this is given in [28].

L_2 Stability:

Given $\bar{\psi}(t)$ as defined in (3.4.21), (3.4.23), and $\phi(t)$ as in (3.4.9) which is rewritten as:

$$\phi(t-d) = W_1\,\psi(t) + \phi^*(t-d) \tag{3.4.25a}$$

$$\phi^*(t-d) = W_1 C_R y_m(t) + W_2 d(t) \tag{3.4.25b}$$

where $W_1 \triangleq C^{-1}[A, q^{-1}A, \ldots, q^{-n_R}A, q^{-d}B, q^{-d-1}B, \ldots, q^{-d-n_S}B]^T$

$W_2 \triangleq C^{-1}[-q^{-d}R^*, -q^{-d-1}R^*, \ldots, -q^{-d-n_R}R^*, q^{-d}S^*, q^{-d-1}S^*, \ldots, q^{-d-n_S}S^*]$

then under Assumption 1 if verified

$$\bar{\psi}(t) \in L_2 \Rightarrow \psi(t) \in L_\infty$$

Proof: Define the exponentially weighted signals

$$X_t^\mu = \mu^{-1} X(t) \tag{3.4.26}$$

from (3.4.12)

$$\mu^{-N} \rho(N) \leq \rho(0) + ||\phi_{t-d}^\mu||_N^2 + \mu^{-N} \cdot \frac{\rho}{1-\mu} \tag{3.4.27}$$

Applying the truncated L_2 norm to the exponentially weighted version of (3.4.24) and noting Assumption 1, using $C_R y_m = y_m$,

$$||\phi_{t-d}^\mu||_N \leq \gamma_2'[||y_{mt}^\mu||_N + ||\psi_t^\mu||_N] + \gamma_2''||d_t^\mu||_N \tag{3.4.28a}$$

where γ_2', γ_2'' are L_2-gains defined as:

$$\left.\begin{aligned} \gamma_2' &= \gamma_2\{W_1[(\mu^{\frac{1}{2}}q)^{-1}]\} \\ \gamma_2'' &= \gamma_2\{W_2[(\mu^{\frac{1}{2}}q)^{-1}]\} \end{aligned}\right\} \tag{3.4.28b}$$

using (3.4.13), (3.4.27), (3.4.28)

$$|\psi_N|^2 \leq \frac{\mu^{-N}\psi_N^2}{\rho(0)+\mu^{-N}\rho+2\{[\gamma_2']^2(||y_{mt}^\mu||_N^2+||\psi_t^\mu||_N^2+[\gamma_2'']^2||d_t^\mu||_N^2\}}$$

since $\psi(t) \in L_2$ by assumption $\bar{\psi}(t) \to 0$ so that for all $\delta > 0$, N_o such that for all $N \geq N_o$

$$\bar{\psi}_N^2 \leq \delta$$

therefore

$$\mu^{-N}\psi_N^2 \leq \delta\{\rho(0)+\mu^{-N}\rho+[\gamma_2']^2||y_m\,{}_t^{\mu}||_N^2+2[\gamma_2'']^2||d_t^{\mu}||_N^2 + 2[\gamma_2']^2(\mu^{-N}\psi_N^2+||\psi_t^{\mu}||_{N-1}^2)\}$$

Choosing δ such that $1-2\delta[\gamma_2']^2 > \mu$ and applying the Bellman Gronwall lemma yields:

$$\psi_N^2 \leq \delta^2 K_1\left\{1 + 2\delta(\gamma_2')^2 \sum_{t=0}^{N-1}\left(\frac{\mu}{1-2\delta(\gamma_2')^2}\right)^{N-t}\right\}$$

where the term in brackets is smaller than 1 and the series is convergent and therefore $\psi(t) \in L_\infty$.

The above implies that if $\bar{\psi}(t) \in L_2$, $y_m(t)$, $d(t) \in L_\infty$ and Assumption 1 holds then $\phi(t) \in L_\infty$ and consequently $\rho(t) \in L_\infty$. The extension to the L -stability is shown in [28].

3.5 SIMILARITIES OF ALGORITHMS

While proofs of robust stability for the different algorithms for robust adaptive control by different authors vary, in actual fact the underlying principles are basically the same. Thus, similarities will be shown in this section of the normalization signals, dead zones, control objectives and the projection algorithms used for adaptation. These encourage the author to derive the stability of these algorithms using the fundamental principles of Chapter 2 in a simpler fashion; but first the similarities.

3.5.1 Normalization

The three examples of normalization signals in discrete-time first mentioned in Section 3.2 will be repeated here and compared.

(i) $$N(t) = \gamma_0 + m(t) \; ; \quad \gamma_0 > 0$$

$$m(t) = \sigma_0 m(t-1) + |u(t-1)| + |y(t-1)| \qquad (3.2.5)$$

$$m(0) > 0 \; ; \quad 0 < \sigma_0 < 1$$

(ii) $$s(t) = \sigma s(t-1) + \max\{|y(t-1)| + |u(t-d)|, s\}$$

$$0 < \sigma < 1, \quad s > 0 \qquad (3.2.6)$$

(iii) $$\rho(t) = \mu\rho(t-1) + \max(||\phi(t-d)||^2, \rho) \qquad (3.2.7)$$

$$\rho > 0, \quad 0 < \mu < 1, \quad \phi(t) = (u(t),\ldots,u(t-n_S),y(t),\ldots,y(t-n_R))^T$$

A cursory look shows the similarity of approach in each of the three. Comparing (3.2.5) taken from [27] to (3.2.6) taken from [29], it can be seen that these two are quite similar since

$$N(t) = \gamma_0 + \sigma_0 m(t-1) + |u(t-1)| + |y(t-1)| \qquad (3.5.1)$$

and $$s(t) = \sigma s(t-1) + \max\{|y(t-1)| + |u(t-d)|, s\}$$

$$= \sigma s(t-1) + |y(t-1)| + |u(t-d)| + K \qquad (3.5.2)$$

where K represents the upper bound of the difference between $[|y(t-1)| + |u(t-d)|]$ and s. The only difference between (3.5.1) and (3.5.2) is the particular delayed value of the control input $u(t)$ used.

Likewise, looking at (3.2.7) taken from [28] and comparing it to, say, (3.2.6) one gets:

$$s(t) = \sigma s(t-1) + \max\{|y(t-1)| + |u(t-d)|, s\}$$

$$\rho(t) = \mu\rho(t-1) + \max(||\phi(t-d)||^2, \rho)$$

but from [31], $|u(t-d)| \leq m_3 + m_4 \max |y(\tau)|$ (3.5.3)

where $m_3 < \infty$, $m_4 < \infty$, $1 \leq \tau \leq T$

and assuming $\max_{0\leq\tau\leq T} |y(\tau)| = m_2 < \infty$ (3.5.4)

then (3.5.3) becomes:

$$|u(t-d)| \leq m_3 + m_4 m_2 \quad (3.5.5)$$

also using the definition of $\phi(t-d)$ above, and [16,31],

$$||\phi(t-d)|| \leq n.\{m_3 + [\max(1, m_4)]\max|y(\tau)|\} \quad (3.5.6)$$

where n = dimension of $\phi(t)$.

Substituting (3.5.4) into (3.5.6) it becomes:

$$||\phi(t-d)|| \leq n\{m_3 + [\max(1, m_4)]m_2\} \quad (3.5.7)$$

then

$$s(t) \leq \sigma s(t-1) + \max\{[|y(t-1)| + (m_3 + m_4 m_2)], s\} \quad (3.5.8)$$

while

$$\rho(t) \leq \mu\rho(t-1) + \max\{[n(m_3 + [\max(1, m_4)]m_2)]^2, \rho\} \quad (3.5.9)$$

Although it will appear as if (3.5.9) uses a squared version of the other two as its normalizing factor, this is not true, as was shown in Section 3.4 . Since the actual factor turns out to be the square root of $\rho(t)$, i.e. $\rho(t)^{\frac{1}{2}}$ that is used. Hence,

$$\rho(t)^{\frac{1}{2}} = \mu\rho(t-1))^{\frac{1}{2}} + \max\{[n(m_3+[\max(1,m_4)m_2)]^2,\rho]\}^{\frac{1}{2}} \qquad (3.5.10)$$

$$\leq (\mu\rho(t-1))^{\frac{1}{2}} + \max\{[n(m_3+[\max(1,m_4)m_2)], \rho^{\frac{1}{2}}]\} \qquad (3.5.11)$$

and using $r(t) \triangleq \rho(t)^{\frac{1}{2}}$, $r(t-1) \triangleq \rho(t-1)^{\frac{1}{2}}$, $\mu^{\frac{1}{2}} = \gamma$, $r = \rho^{\frac{1}{2}}$, allows (3.5.11) to be written in the form:

$$r(t) \leq \gamma r(t-1) + \max\{n(m_3+ \max(1,m_4)]m_2),r\} \qquad (3.5.12)$$

which is similar to (3.5.8).

3.5.2 Dead Zones

Dead zones are also basically the same as they are all usually functions of prior information, such as bounds on plant input, output and bounds of the disturbance amplitude (see [27,31,56]), but the one proposed by Kreisselmeir and Anderson is a bit different in that it is a relative dead zone. An example of the more typical dead zone [31,56,46] is given below which may be compared with that in Section 3.3 , (see Figure 3.3, next page)

Projection algorithm with dead zone [31]:

Let the disturbance (noise, unmodelled dynamics) be a bounded sequence $\omega(t)$ such that $\sup|\omega(t)| \leq M_e$, then the algorithm is:

$$\left.\begin{aligned} \hat{\theta}(t) &= \hat{\theta}(t-1) + \frac{a(t-1)\phi(t-1)}{c+\phi^T(t-1)\phi(t-1)} \cdot [y(t)-\phi^T(t-1)\hat{\theta}(t-1)] \\ \text{where} \quad a(t-1) &= \begin{cases} 1 & \text{if } |y(t)-\phi^T(t-1)\hat{\theta}(t-1)| > 2M_e \\ 0 & \text{otherwise} \end{cases} \end{aligned}\right\} \qquad (3.5.13)$$

Let $e(t) = y(t) - \phi^T(t-1)\theta(t-1), c > 0$ and $\hat{\theta}(0)$ is given.

The diagram of the above dead zone is shown below, but the slope of this

is different from that in Figure 3.1 since that is a relative dead zone.

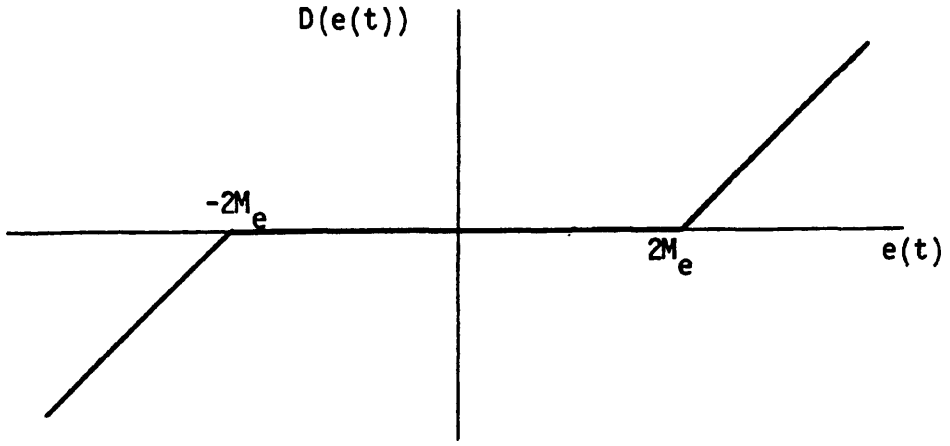


Figure 3.3

3.5.3 Algorithms and Objectives

Basically the three algorithms (the Basic MRAC, 3.3, 3.4) all have the same control objective which is mainly the elimination of the tracking error between the plant and reference model output. Although different error equations are used, once again they are all derived from the fundamental error equation in Chapter 2, namely $e(t) = y(t)-y_m(t)$. For example, in Section 3.3 the error equation used is derived from

$$e(t) = 1/b_o\ A_m(z^{-1})[y(t)-y_m(t)] \qquad \text{(see equation (23) in [27])}$$

and this compares to

$$e(t) = C_R(q^{-1})[y(t)-y_m(t)]$$

used in Section 3.4, from which it can be concluded that these two are merely filtered versions of the basic error equation given above with the basic scheme having a filter equal to one.

The Algorithms

Rewriting equation (3.3.15), using

$$z^{-d}[\underline{w}^T(t)\,\underline{\theta}^* - \underline{v}^T(t)\,\underline{\theta}(t)] - \eta_1(t) = 0\ ,$$

it becomes

$$E(t) = \underline{w}^T(t-d)\,\tilde{\theta}(t) + \eta_1(t) \qquad (3.5.14)$$

using this and the fact that $|\eta_1(t)| < d_0 N(t)$ $(t \geq t_1)$ (see [27]), then $D(E_1)$ which is equation (3.3.18b), can be written as

$$D(E_1) = \alpha(t)w^T(t-d)\phi(t)/N(t), \quad 0 < \alpha(t) \leq 1 \qquad (3.5.15)$$

This then leads to the adaptive algorithm (3.3.18a) becoming

$$\hat{\tilde{\theta}}(t+1) = \left\{\tilde{\theta}(t) - \frac{\alpha(t)\underline{w}(t-d)\underline{w}^T(t-d)\tilde{\theta}(t)}{\gamma_1 + \underline{w}^T(t-d)\underline{w}(t-d)}\right\} \qquad (3.5.16)$$

(notice the absence of both the normalizing factor and the dead zone), which is similar to the Basic projection adaptive algorithm defined in Chapter 2, but which is rewritten and rearranged below for ease of comparison:

$$\hat{\theta}(t) = \hat{\theta}(t-d) - \frac{1/\hat{\beta}_0\ \phi(t-d)\ e(t)}{1 + \phi^T(t-d)\phi(t-d)} \qquad (2.3.13)$$

Substituting $\phi^T(t-d)\tilde{\theta}(t-d) = e(t)/\beta_0$ for $e(t)$, the above becomes, on subtracting $\underline{\theta}^*$ from both sides:

$$\tilde{\theta}(t) = \tilde{\theta}(t-d) - \frac{a(t)\phi(t-d)\phi^T(t-d)\tilde{\theta}(t-d)}{1 + \phi^T(t-d)\phi(t-d)} \qquad (3.5.17)$$

where $a(t) = \beta_0/\hat{\beta}_0$; $0 < a(t) \leq 2$.

This as can be seen is very similar to (3.5.16) above, the differences being the delayed value of the parameter error vector $\tilde{\Theta}(t-d)$ used in (3.5.17) as compared to $\tilde{\Theta}(t)$ used in (3.5.16).

Let $a(t) = 1$ and multiply both sides of (3.5.17) by $\phi^T(t-d)$ to obtain

$$\phi^T(t-d)\tilde{\Theta}(t) = \phi^T(t-d)\tilde{\Theta}(t-d) - \frac{\phi^T(t-d)\phi(t-d)\phi^T(t-d)\tilde{\Theta}(t-d)}{1 + \phi^T(t-d)\phi(t-d)}$$

$$= \frac{\phi^T(t-d)\tilde{\Theta}(t-d)}{1+\phi^T(t-d)\phi(t-d)} \qquad (3.5.18)$$

Substituting this into (3.5.17),

$$\tilde{\Theta}(t) = \tilde{\Theta}(t-d) - \phi(t-d)\phi^T(t-d)\tilde{\Theta}(t) \qquad (3.5.19)$$

This comparable to the Ortega et al algorithm:

$$\tilde{\Theta}(t) = \tilde{\Theta}(t-d) + F\,\bar{\phi}(t-d)\bar{e}(t) \qquad (3.4.19)$$

which, using $\bar{e}(t) = \phi^T(t-d)(\Theta^*-\Theta(t-d))$; $F > 0$ a scalar, becomes:

$$\tilde{\Theta}(t) = \tilde{\Theta}(t-d) - F\,\bar{\phi}(t-d)\bar{\phi}^T(t-d)\tilde{\Theta}(t-d) \qquad (3.5.20)$$

3.5.4 Stability Analysis

Since all the algorithms above can all be expressed in a similar form then their convergence properties can surely be derived similarly and this leads to a simplified proof of robust stability. Most of the modifications mentioned in this chapter are passive modifications [54], except for the persistency of excitation condition, as a result of which the problem of robust stability reduces to proving that the regressor vector, i.e. $\phi(t) = [u(t),\ldots,u(t-n_i),y(t),\ldots,y(t-n_o)]$ is bounded. Also, $u(t)$, $y(t)$ are

required to be uniformly bounded. This allows the proof of stability of all three algorithms used in this work and study to be treated in the same manner. The proof is now a straightforward use of the principles and theorems used for the proof of stability of the basic scheme in Chapter 2. As an example, the robust adaptive algorithm proposed by Kreisselmeier and Anderson is treated here using the Key Technical Lemma (K.T.L.) defined in Section 2.4.1.

Proof:

In Section 3.3 some properties of the algorithm were given and these will now be used in the proof of stability. Already it is known that the parameter vector $\underline{\theta}(t)$ is bounded between $\underline{\theta}_{min}$ and $\underline{\theta}_{max}$, while for the dead zone $D[E_1(t)]$; $\lim_{t\to\infty} D[E_1(t)] = 0$.

Using the Key Technical Lemma on

$$\frac{D(E_1(t)^2)}{\gamma_1 + \underline{w}^T(t-d)\underline{w}(t-d)} = 0$$

it is obvious that the first two conditions are satisfied. Thus, all that is required is to show that the regressor vector $\underline{w}(t-d)$ is also a bounded sequence since $D(E_1(t))$ is a bounded sequence as shown in [27].

From the definition of $\underline{w}(t-d)$ it is possible to show that

$$||\underline{w}(t-d)|| \leq p\{m_3 + m_4 (\max|y_p(\tau)|)\}$$

where p is the dimension of $\underline{w}(t-d)$, $1 \leq \tau \leq t$, but $y_p(t)$ is a function of $E_1(t)$, which implies that

$$||w(t-d)|| \leq c_1 + c_2 \max_{1\leq\tau\leq t} |E_1(\tau)|$$

Using $|E_1(t)| < |D(E_1(t)| + d_0$ (see [60]) in the above equation, it becomes:

$$||\underline{w}(t-d)|| \leq c_1 + c_2 \max[|D(E_1(t)| + d_0]$$

$$\leq K_1 + K_2 \max |D(E_1(t)|$$

$$0 < K_1 < \infty, \quad 0 < K_2 < \infty$$

and this implies the linear boundedness of $\underline{w}(t-d)$.

Similarly for the Ortega et al algorithm, $\tilde{\theta}(t) = \tilde{\theta}(t-d) + F\bar{\phi}^T(t-d)\bar{e}(t)$ with $\bar{e}(t) = \rho(t)^{-\frac{1}{2}}e(t)$; $\bar{\phi}(t-d) = \rho(t)^{-\frac{1}{2}}\phi(t-d)$ where $\rho(t)$ is defined as $\rho(t) = \mu\rho(t-1) + \max(|\phi(t-d)|^2, \rho)$, the algorithm can be rewritten as:

$$\tilde{\theta}(t) = \tilde{\theta}(t-d) + \frac{F\phi^T(t-d)e(t)}{\rho(t)}$$

Noting that $\rho(t)$ is a function of $\phi^T(t-d)\phi(t-d)$ then the equation becomes:

$$\tilde{\theta}(t) = \tilde{\theta}(t-d) + \frac{F\phi^T(t-d)e(t)}{f\{\phi^T(t-d)\phi(t-d)\}}$$

from which the stability analysis can be obtained using the Key Technical Lemma.

3.6 MIMO EXTENSION OF THE KREISSELMEIER AND ANDERSON ALGORITHM

From the S.I.S.O. algorithm and Section 3.5 it can be seen that this algorithm is quite similar to the Basic algorithm in Chapter 2 except for the modifications, i.e. the dead zone, normalization and parameter bounds. This then inspired the derivation of a MIMO version of the robust algorithm based on the basic MIMO algorithm in Section 2.5 . A generalized form of this derived robust MIMO algorithm is shown below and an example based on a 2x2

plant explained. Another work which is in the same direction is in [40], but which is based on adaptive regulation.

3.6.1 The Reference Model

This is given by

$$A_m(z^{-1})\, \underline{y}_m(t) = \xi(z) B_m\, \underline{r}(t) \qquad (3.6.1)$$

where $\xi(z) = z^{-d} I$; $d = \min_{ij} d_{ij}$ and A_m, B_m are diagonal matrices consisting of polynomial functions in z^{-1} of the form:

$$A_m = \begin{bmatrix} A_{m_1} & & & \\ & A_{m_2} & & \\ & & \ddots & \\ & & & A_{m_n} \end{bmatrix} \qquad B_m = \begin{bmatrix} B_{m_1} & & \\ & \ddots & \\ & & B_{m_m} \end{bmatrix}$$

$$A_{m_i} = 1 + a_1 z^{-1} + \ldots a_{n_i} z^{-n_i}$$
$$B_{m_i} = b_o + b_1 z^{-1} + \ldots b_{m_i} z^{-m_i}$$

and $\underline{y}_m$, $\underline{r}(t)$ are the reference model output and input vectors respectively. For this system $n = m$, i.e. the output and input vectors are of the same order.

The Plant

This is assumed to be

$$A(z^{-1})\underline{y}(t) = B(z^{-1})\underline{u}(t) + \underline{n}(t) \qquad (3.6.2)$$

where $\underline{n}(t)$ is the disturbance vector (also represents unmodelled dynamics),

A, B are n×n matrices of polynomial functions, and $\underline{y}(t)$, $\underline{u}(t)$, $\underline{n}(t)$ are all n×1 vectors.

3.6.2 Structure of the Controller

Defining

$$\left.\begin{aligned} v_1^T(t) &= [\underline{u}^T(t-1),\ldots,\underline{u}^T(t-n)] \\ v_2^T(t) &= [\underline{y}^T(t-1),\ldots,\underline{y}^T(t-n)] \end{aligned}\right\} \qquad (3.6.3)$$

then

$$v^T(t) = [\underline{r}^T(t),\ v_1^T(t),\ v_2^T(t)] \qquad (3.6.4)$$

and

$$\underline{u}(t) = \theta^T(t)\ \underline{v}(t) \qquad (3.6.5)$$

where

$$\theta^T(t) = [\theta_0^T(t),\ \theta_1^T(t),\ \theta_2^T(t)] \qquad (3.6.6)$$

θ_0, θ_1, θ_2 are n×n matrices.

The Adaptive Algorithm

As in Section 3.3 define $\underline{w}(t)$ as

$$\underline{w}^T(t) = [(B_m^{-1}\xi^{-1}(z)A_m(z^{-1})\underline{y}(t))^T, v_1^T(t), v_2^T(t)] \qquad (3.6.7)$$

Let $$E(t) = \theta^T(t)\underline{w}(t-d) - \theta^T(t-d)v(t-d) \qquad (3.6.8)$$

Define

$$m_i(t) = \sigma_0 m_i(t-1) + |u_i(t-1)| + |y_i(t-1)| \qquad (3.6.9)$$

$$\sigma_0 \in (0,1)$$

and

$$N_I(t) = \begin{bmatrix} \gamma_o+m_1(t) & & \\ & \ddots & \\ & & \gamma_o+m_n(t) \end{bmatrix} ; \quad \gamma_o > 0 \qquad (3.6.10)$$

then $$E_1(t) = N_I^{-1}(t)E(t) \qquad (3.6.11)$$

and the algorithm is given by:

$$\hat{\theta}^T(t+1) = \theta^T(t) - \frac{N_I(t)D(E_1(t)\underline{w}^T(t-d)}{\gamma_1+w^T(t-d)w(t-d)} \qquad (3.6.12a)$$

where

$$D(E_1(t)) = D\begin{bmatrix} E_{11}(t) \\ \vdots \\ E_{n1}(t) \end{bmatrix} ; \quad D(E_{i1}) = \begin{cases} 0 & \text{if } |E_{i1}| \leq d_{oi} \\ E_{i1}-d_{oi} & \text{if } E_{i1} > d_{oi} \\ E_{i1}+d_{oi} & \text{if } E_{i1} < d_{oi} \end{cases}$$

and (3.6.12b)

$$\theta_{ij}(t+1) = \begin{cases} \hat{\theta}_{ij}(t+1) & \text{if } \theta_{ij}^{min} \leq \hat{\theta}_{ij}(t+1) \leq \theta_{ij}^{max} \\ \theta_{ij}^{min} & \text{if } \hat{\theta}_{ij} < \theta_{ij}^{min} \\ \theta_{ij}^{max} & \text{if } \hat{\theta}_{ij} > \theta_{ij}^{max} \end{cases} \qquad (3.6.12c)$$

Again, as in [27], it can be shown that

$$D[E_1(t)] \simeq \alpha(t)N_I^{-1}(t)\tilde{\theta}^T(t)\underline{w}(t-d) \qquad (3.6.13)$$

where $\tilde{\theta}(t) = \theta(t) - \theta^*$; $0 \leq \alpha(t) \leq 1$

and this leads to (3.6.12a) becoming, on subtracting θ^* from both sides,

$$\tilde{\hat{\theta}}^T(t+1) = \tilde{\theta}(t) - \frac{\alpha(t)\tilde{\theta}^T(t)\underline{w}(t-d)\underline{w}^T(t-d)}{\gamma_1+\underline{w}^T(t-d)w(t-d)} \qquad (3.6.14)$$

This is similar to equation (2.6.21) in Section 2.6, and the proof of stability can be treated in a similar manner. The use of $\underline{w}(t)$ as the regressor vector leads to a reduction in the number of adaptive parameters, as will be shown now for a 2×2 system. This was discovered accidentally during experiments conducted in Chapter 4 (see Figures 4.21 a,b) and it was this which encouraged the derivation of the above algorithm theoretically from which the reduction in parameters was also shown to be correct.

3.6.3 Example, 2×2 Plant

The reference model will be:

$$\begin{bmatrix} A_{m_1} & 0 \\ 0 & A_{m_2} \end{bmatrix} \begin{bmatrix} y_{m_1} \\ y_{m_2} \end{bmatrix} = \begin{bmatrix} B_{m_1} & 0 \\ 0 & B_{m_2} \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$$

with A_{m_i}, B_{m_i} as defined in Section 3.6.1 .

The plant is assumed to be $A(z^{-1})\underline{y}(t) = B(z^{-1})\underline{u}(t) + n(t)$, where A,B are 2×2 matrices and $\underline{y}$, u(t) are 2×1 vectors as defined earlier.

The control input is given by $\underline{u}(t) = \theta^T(t)\underline{v}(t)$ (3.6.5).
Assuming n = 1

$$v^T(t) = [r_1(t), r_2(t), u_1(t-1), u_2(t-1), y_1(t-1), y_2(t-1)]$$
and $$\theta^T(t) = [\theta_0^T(t), \theta_1^T(t), \theta_2^T(t)]$$

where $\theta_0, \theta_1, \theta_2$ are 2×2 matrices implying that $\theta^T(t)$ = 2×6 matrix; but using the definition of $\theta^* = [\theta_0^*, \theta_1^*, \theta_2^*]$, θ_0^* is proportional to B_m, i.e.

$$\theta_0^* = K \begin{bmatrix} B_{m_1} & 0 \\ 0 & B_{m_2} \end{bmatrix}$$

which implies that $\theta_{12}^*, \theta_{21}^* = 0$.

The above leads to the conclusion that setting $\theta_{12} = 0$ and θ_{21} will lead to convergence. More details about this will be given in the next chapter, but as of now this leads to

$$\theta(t) = \begin{bmatrix} \theta_{11}, 0, \theta_{13}, \theta_{14}, \theta_{15}, \theta_{16} \\ 0, \theta_{22}, \theta_{23}, \theta_{24}, \theta_{25}, \theta_{26} \end{bmatrix}$$

$$= \begin{bmatrix} \theta_{11}, \theta_{13}, \theta_{14}, \theta_{15}, \theta_{16} \\ \theta_{22}, \theta_{23}, \theta_{24}, \theta_{25}, \theta_{26} \end{bmatrix}$$

which is a reduction in parameters by two terms.

3.7 ROBUSTNESS OF THE BASIC MIMO MRAC SCHEME

Just as the similarities in the basic MRAC (SISO) scheme and the robust adaptive scheme in [27] led to the derivation of the robust MIMO adaptive scheme proposed in Section 3.6. This also implies that the passive modifications used for robustness can be easily applied to the Basic MIMO MRAC scheme, to give it robust stability properties. Goodwin and Sin in [31] recommended the use of some of these modifications, i.e. dead zone, parameter bounds, but not together as it is proposed here. The differences between the modified basic MRAC scheme suggested below and that in Section 3.6 consists of the use of a different error function and regressor vector (although these have some common elements) compared to that in Section 3.6 .

3.7.1 The Modified Basic MRAC Algorithm (MIMO version)

The modified Basic (MIMO) MRAC algorithm is simply stated below, but not proved since the passive modifications made do not alter the proofs already given in Chapter 2.

$$\hat{\Theta}^T(t+d) = \Theta^T(t) - \frac{P\, D[e(t+d)]\phi^T(t)}{1 + \phi^T(t)\phi(t)} \qquad (3.7.1a)$$

where

$$e(t+d) = \begin{bmatrix} e_1(t+d_1) \\ \vdots \\ e_m(t+d_m) \end{bmatrix}; \quad e_i(t) = y_i(t) - y_{mi}(t)$$

$$\phi(t) = [-\underline{y}^T(t), -\underline{y}^T(t-1), \ldots, -\underline{u}^T(t-1), -\underline{u}^T(t-2), \ldots, y_m^T(t+d)]$$

P = matrix of real constants specified *a priori*

$$D(e_i(t+d_i)) = \begin{cases} 0 & \text{if } |e_i(t+d_i)| \leq 2M_e \\ e_i(t+d_i)+2M_e & \text{if } e_i(t+d_i) < 2M_e \\ e_i(t+d_i)-2M_e & \text{if } e_i(t+d_i) > 2M_e \end{cases} \qquad (3.7.1b)$$

and M_e is an upper bound on the amplitude of the system disturbance,

$$\Theta_{ij}(t+d) = \begin{cases} \hat{\Theta}_{ij}(t+d) & \text{if } \Theta_{ij}^{min} \leq \hat{\Theta}_{ij}(t+d) \leq \Theta_{ij}^{max} \\ \Theta_{ij}^{min} & \text{if } \hat{\Theta}_{ij}(t+d) < \Theta_{ij}^{min} \\ \Theta_{ij}^{max} & \text{if } \hat{\Theta}_{ij}(t+d) > \Theta_{ij}^{max} \end{cases} \qquad (3.7.1c)$$

Also, the control law is given by

$$\underline{u}(t) = \Theta^T(t)\phi(t) \qquad (3.7.2)$$

3.8 COMMENTS

(i) As was shown in this chapter, the three algorithms which will be used in Part II, the applications section of this work, are quite similar in composition and nature. These algorithms are those mentioned in Chapter 2 for both SISO and MIMO systems, plus those of Sections 3.3, 3.4 . Apart from the modifications required for robust stability, it can be seen from 3.5 that the fundamental principles are basically the same, the differences being, say, the particular delayed value of a

parameter to use, i.e. $\theta(t)$, $\theta(t-1)$ or $\theta(t-d)$, but this does not matter much as these vectors tend to become similar and constant as time goes by, e.g.

$$\lim_{t\to\infty} ||\theta(t) - \theta(t-1)|| \leq 0$$

$$\lim_{t\to\infty} ||\theta(t) - \theta(t-d)|| \leq 0$$

$$\Rightarrow \quad \lim_{t\to\infty} ||\theta(t-1) - \theta(t-d)|| \leq 0$$

which implies that as $t \to \infty$, then $\theta(t) = \theta(t-1) = \theta(t-d)$. Thus, choice of time delayed value used has no lasting effect on convergence properties.

(ii) The only algorithm which, although having the same fundamental principles as the others, appears to differ is the Kreisselmeier and Anderson algorithm. This is due to the composition of its regressor vector $\underline{w}(t)$ and error function $E(t)$. But the differences are not that significant since (a) the regressor vector has common terms with those of the others, but has a term $(1/b_m)\, z^d A_m(z^{-1})y(t)$ which is different from the others which usually have the term as either $r(t)$ or $y_m(t)$. The term $(1/b_m)\, z^d A_m(z^{-1})y(t)$ will be a function, or be proportional to, $r(t)$ provided $y(t)$ is tracking $y_m(t)$, since $r(t) = (1/b_m)\, z^d A_m(z^{-1})y_m(t)$, hence $(1/b_m)z^d A_m(z^{-1})y(t) \propto r(t)$; (b) $E(t)$ for this algorithm is known as the identification error and it is defined as:

$$E(t) = \underline{w}^T(t-d)\theta(t) - \underline{v}^T(t-d)\theta(t-d)$$

$$= (1/b_m)\theta_o(t)A_m(z^{-1})[y(t)-y_m(t)]+v^T(t-d)[\theta(t)-\theta(t-d)]$$

A look at the first term on the R.H.S. of the above shows it to be a

Sort of filtered output tracking error $(e(t) = y(t)-y_m(t))$, i.e. $C.[y(t)-y_m(t)]$ where $C = (1/b_m)\theta_o(t)A_m)z^{-1})$, while the second term appears to be a form of input error, since $u(t-d) = v^T(t-d)\theta(t-d)$ the input d steps back and $v^T(t-d)\theta(t)$ can be said to be a form of *a posteriori* input. This as time goes on becomes zero and reduces $E(t)$ to a filtered tracking error; but in general one can say $E(t)$ is a combination of a filtered tracking error and a form of input error. Again this is not so different when compared to the other schemes.

(iii) Although the basic MRAC algorithm has no passive modifications, it should not be forgotten that persistency of excitation also makes an algorithm robust. As mentioned earlier, a step input signal is persistently exciting if just of order one and, as stated earlier, provided parameter identification is not of real importance then (see [57]) unmodelled dynamics would not cause any difficulties if the frequency of the reference signal is sufficiently low and it has a large enough magnitude to overcome destabilizing effects of these. In the following chapters these will be seen to be true from the results obtained using the basic algorithms.

Chapter 4

THE COUPLED HYDRAULIC TANKS

4.1 INTRODUCTION

Fluid flow and fluid level control problems arise in a lot of fluid dependent systems. Fluid level control is a basic and important problem in automatic control, with applications to diverse systems such as:

(i) Power generation plants where fluid level and flow control are essential in the operation of boilers, a vital part of steam generation for electricity production or level water in dams for the production of electricity by hydroelectric generation.

(ii) Water resources systems, water distribution networks which also involve in some cases water levels in dams, reservoirs and tanks for efficient and reliable distribution of water to consumers, or even water treatment plants.

(iii) Sewer control systems using detention reservoirs for treatment of sewage, and lastly

(iv) Chemical process industries which vary from oil to sugar production in which the control of fluid levels in storage tanks, chemical blending and reaction vessels is crucial.

Coupled hydraulic tanks is a common form of the problems involved in the above systems. Typical examples of these fluid levels control are ones in which it is required to supply fluid to a chemical reactor at a constant rate Q_A, to which end a reservoir or hold-up tank may be used with the dual aim of smoothing variations out of the upstream supply flow Q_i and also ensuring a temporary reactant supply should there be failure of supply

from the source (i.e. upstream of the hold-up tank), Figure 4.1a, or one in which the fluid of the second tank (see Figure 4.1b) is required to be held at a constant level H (and thus its total fluid volume), hence the output of this tank will be a product flowing out at a constant flow rate.

Although the dynamical properties of fluid level systems are well known, and relatively straightforward, some difficulties exist in terms of lengthy time constants and nonlinearity of the system as a whole. Also, as in some other physical processes, fluid level control of the coupled tanks needs to function at different numbers of process operating points. The parameters of the linearised process model on which the closed-loop system is based usually assume such a wide range of values during process operations that sometimes a single fixed parameter control system proves inadequate to control the process. As a result of the above, the use of adaptive controllers in solving these problems will be quite advantageous and will be more efficient than the classical methods of proportional, proportional plus integral or even proportional plus integral plus derivative where logic switching between several fixed parameter controls might have occurred.

There are two different versions of the rig that were used for the experiments conducted, namely the MIMO version, consisting of two inputs, two outputs, and the SISO rig which consists of a single input, single output. A brief description of these two versions now follows.

4.2 DESCRIPTION OF RIGS

4.2.1 The SISO Tanks

The Single Input/Single Output system consists [62] of two hold-up tanks (Figure 4.2) which are coupled by an orifice. The coupled tanks apparatus is a transparent perspex container measuring 20 cm in length, 10 cm in width and

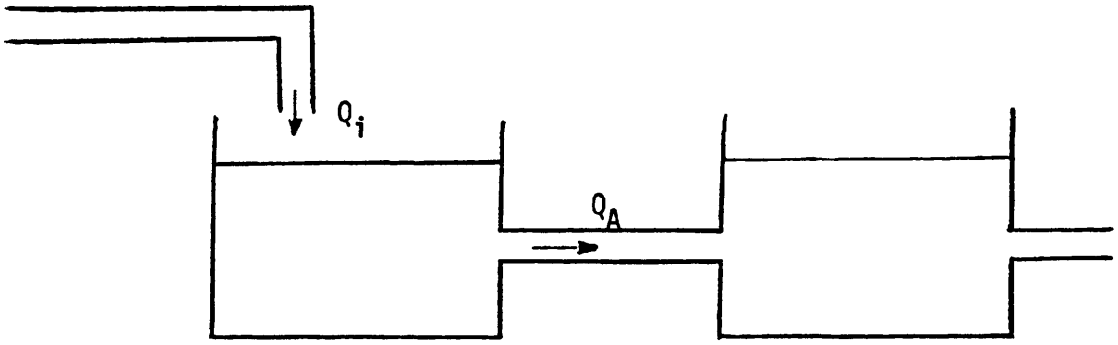


Figure 4.1a

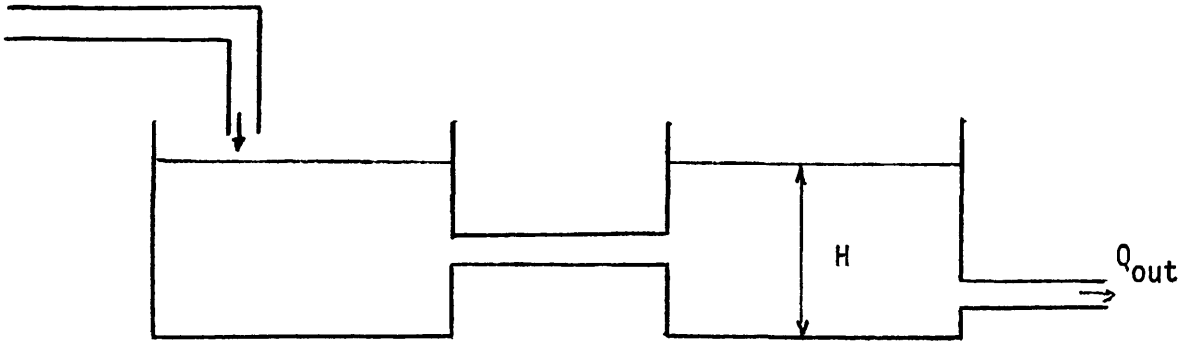


Figure 4.1b

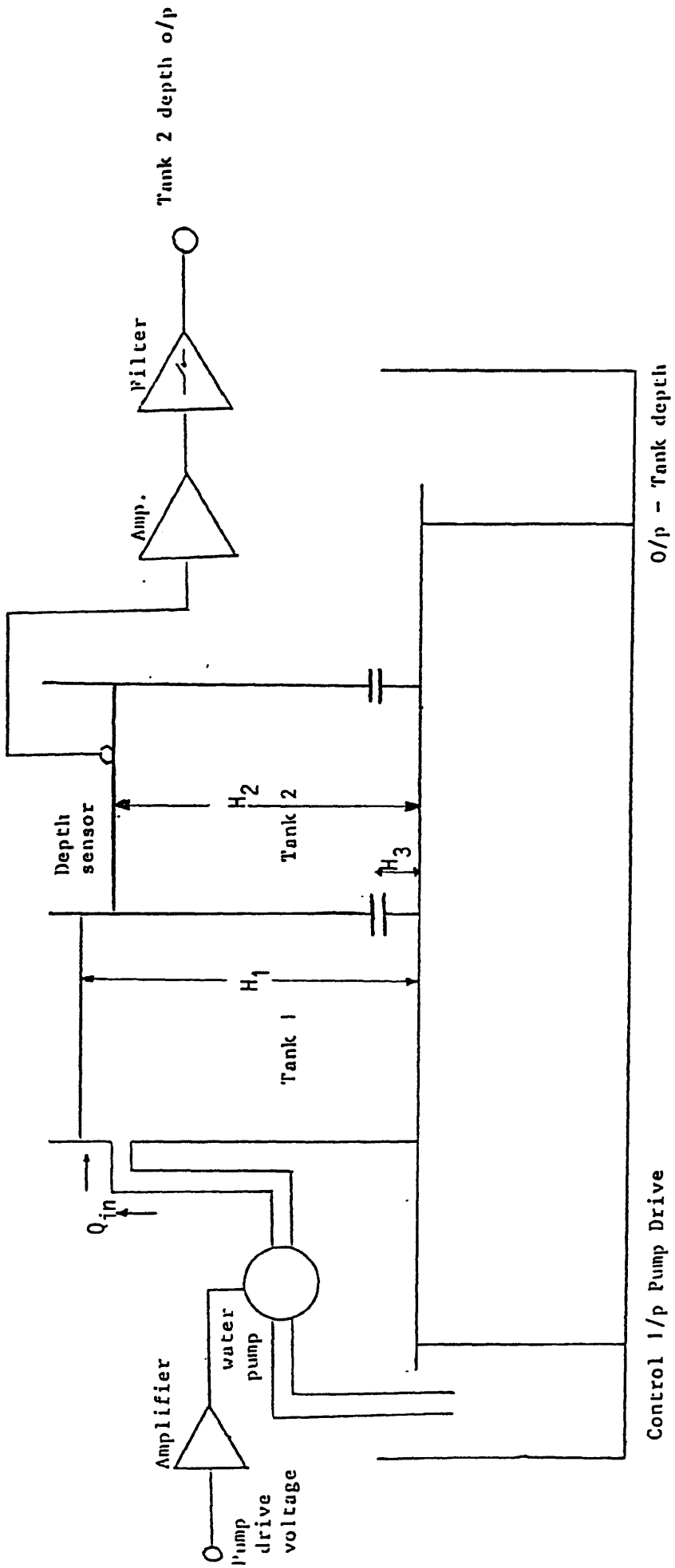


Figure 4.2 The SISO Coupled Hydraulic Tanks Apparatus.

30 cm in height. A centre partition is used to divide the container into two equal tanks. Flow between the tanks is by means of a series of holes drilled at the base of the partition. The holes are situated 2.5 cm above the base of the tanks and have diameters 1.27 cm, 0.95 cm, 0.635 cm, 0.317 cm. These holes constitute the orifice between the tanks. The size of the orifice (and hence the degree of coupling between the tanks) is varied by plugging and unplugging the holes, using the bungs provided. With all the bungs removed the apparatus can be considered to be one big tank and thus a first order system. On the other hand, with the largest three holes plugged, the remaining hole allows only a weak interaction between the fluid levels in the two separate tanks. In this case the apparatus constitutes a second order system.

The input is through a variable speed pump which supplies water to the first tank. Thus water is pumped from a reservoir into the first tank by the variable speed pump which is driven by an electric motor. The orifice allows the water to flow into the second tank and hence out again to the reservoir. The actual flow rate is measured by a flow meter (see Figure 4.3) which is in the flow line between the pump and tank 1. The depth of fluid is measured using parallel rod depth sensors stationed in the second of the tanks. The electrical resistance across this device varies with the water level in the tank. Changes in resistance are detected and produce an electrical signal which is proportional to the height of water in the tank.

The water which flows into tank 2 is allowed to drain out through an adjustable tap, and the entire assembly is mounted on a large tray which also forms the supply reservoir for the pump. Fully open, the drain tap has a diameter of 0.635 cm. There is an instrumentation box inside which the motor drive and depth sensor signals are processed. Calibrations inside the instrumentation box are such that the pump motor may be driven by voltages between 0 and 10 volts applied to the pump drive socket. Likewise

the depth sensor outputs read from the box range in value between 0 and 10 volts.

The depth sensing is prone to noise, and to remove this filters are provided which may be switched in and out of circuit as required.

4.2.2 The MIMO Tanks

The multivariable tank is similar in its description to that of the scalar system except for some additional items, such as (see Figure 4.4) an extra pump, motor and flow meter for tank 2, plus an adjustable drain tap which is fitted to tank 1. Also there is a bigger instrumentation box for the multivariable rig.

For the two rigs described above the pump motor drives were derived from the B.B.C. microcomputer used.

4.3 THE SYSTEM CHARACTERISTICS AND MODEL DERIVATIONS

4.3.1 System Calibration Characteristics

Although the basic control problem is the control of the fluid level in one of the tanks (SISO case) or control of both fluid levels in the two tanks for the MIMO system, there is a need to know the characteristics of the rigs to be used, especially as the pump flow and fluid heights are converted to input and output voltages respectively by means of transducers in the instrumentation box.

Using physical measurements of the input voltages and flow rates, the output voltages and fluid heights, the input and output system calibrations and characteristic curves were obtained. Shown in Figures 4.5a, 4.5b are the SISO plots of (i) flow rate Q_i against input voltage V_i and (ii) output voltage V_2 against water level H_2 which depict the pump characteristic

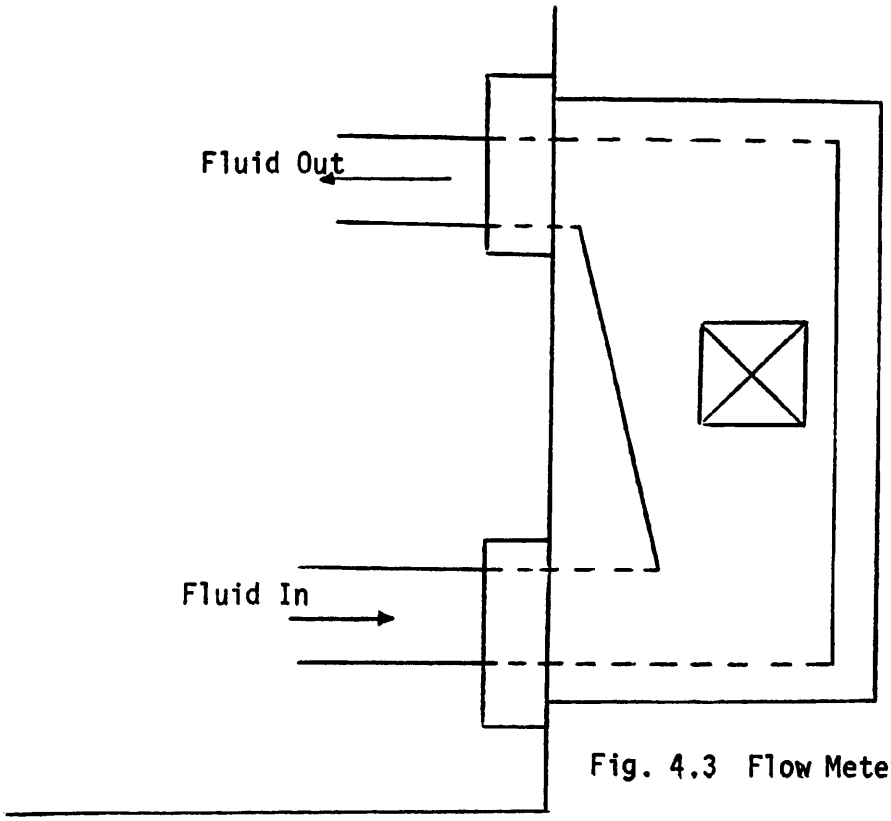


Fig. 4.3 Flow Meter Scheme.

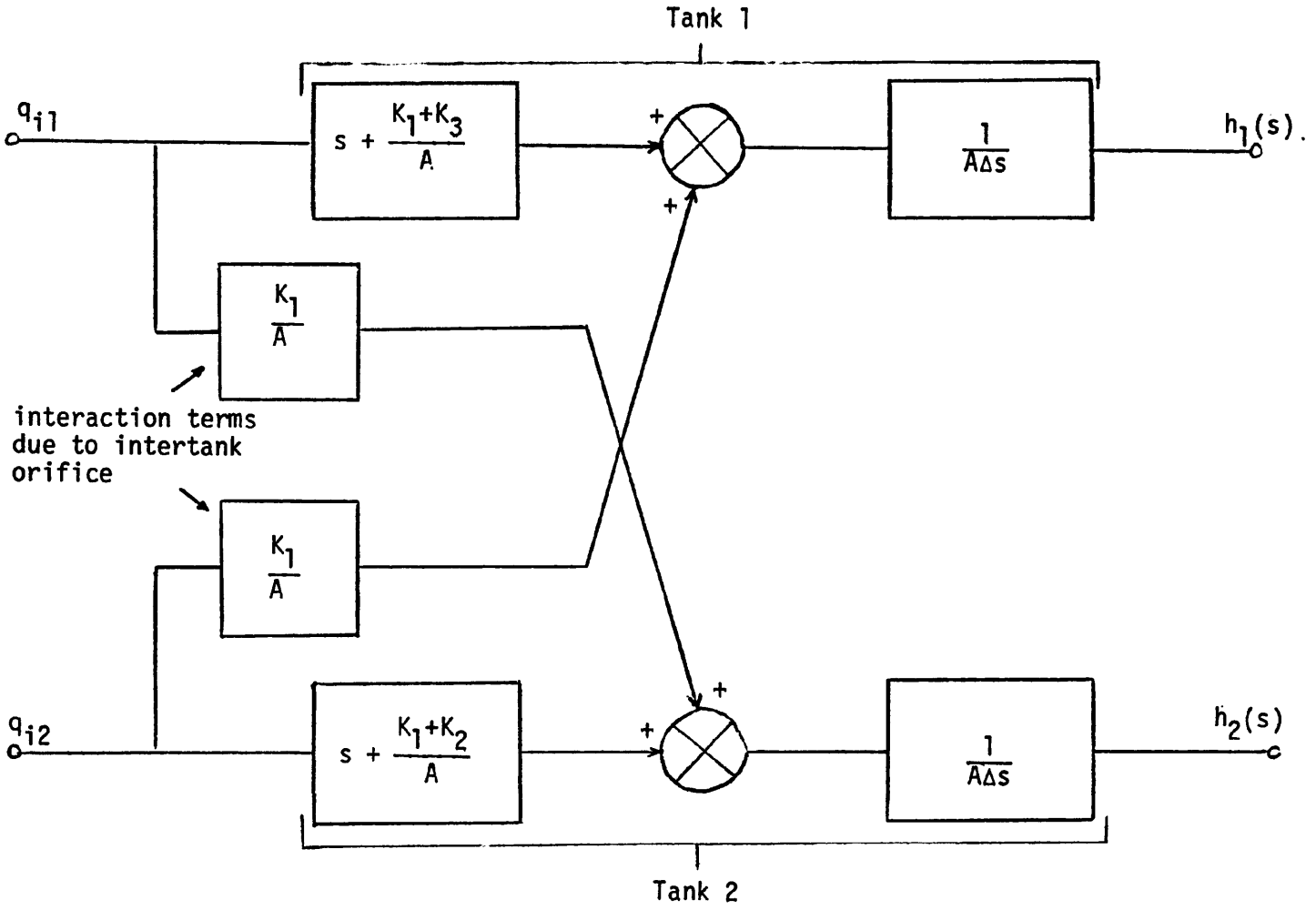


Figure 4.4b Block Diagram of the Multivariable Coupled Tanks.

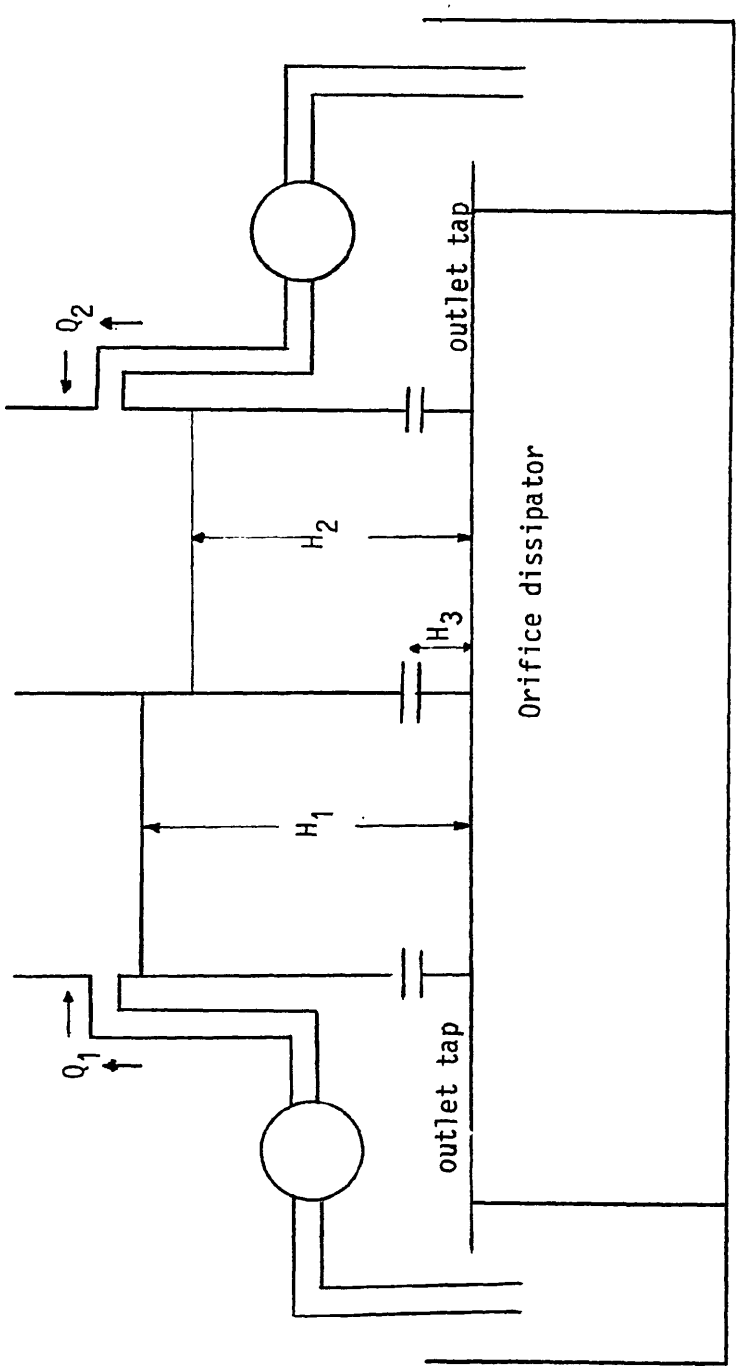


Figure 4.4a The MIMO Coupled Hydraulic Tanks Apparatus.

and depth sensor calibration of the laboratory rig, respectively. Figures 4.6a-d show the pump and depth sensor calibration of the MIMO laboratory rig.

Possibly due to drifts in the transducers it was observed that the characteristics do not stay constant, and that if readings were taken at different times, say days, there would be noticeable differences in the plots. Examples of these differences are as shown in Figure 4.5a for the pump characteristics and Figures 4.6b,d for the depth sensor calibration curves of the MIMO tank. Also apparent is the nonlinearity of the plots with the worst examples being the three depth sensor calibration curves for both rigs. Another reason given for this is the fact that the transducers are affected by the dirt content of the water used and possibly the ambient temperature.

4.3.2 The SISO Model Derivation

The dynamical equations of the system are derived by taking flow balances about each tank. Using Figure 4.2, for the first tank:

Rate of change of fluid volume in tank 1:

$$Q_i - Q_1 = \frac{dV_1}{dt} = A\frac{dH_1}{dt} \qquad (4.3.1)$$

where V_1 = volume of fluid in tank 1

H_1 = height of fluid in tank 1

A = cross-sectional area of tank 1

Q_1 = flow rate of fluid from tank 1 to tank 2

Q_i = pump flow rate.

Similarly for tank 2:

$$Q_1 - Q_0 = \frac{dV_2}{dt} = A\frac{dH_2}{dt} \qquad (4.3.2)$$

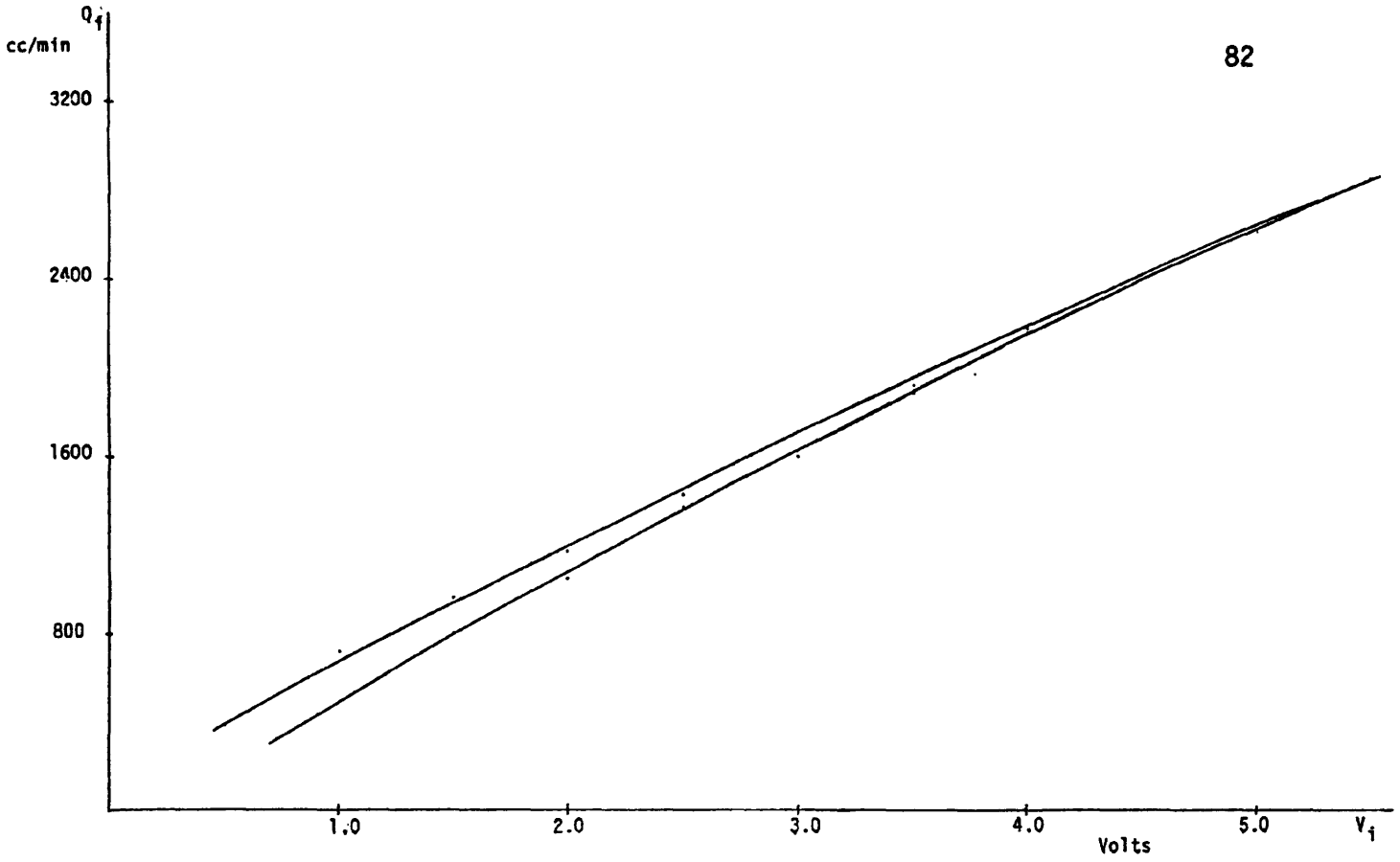


Figure 4.5a Pump Characteristics (SISO rig).

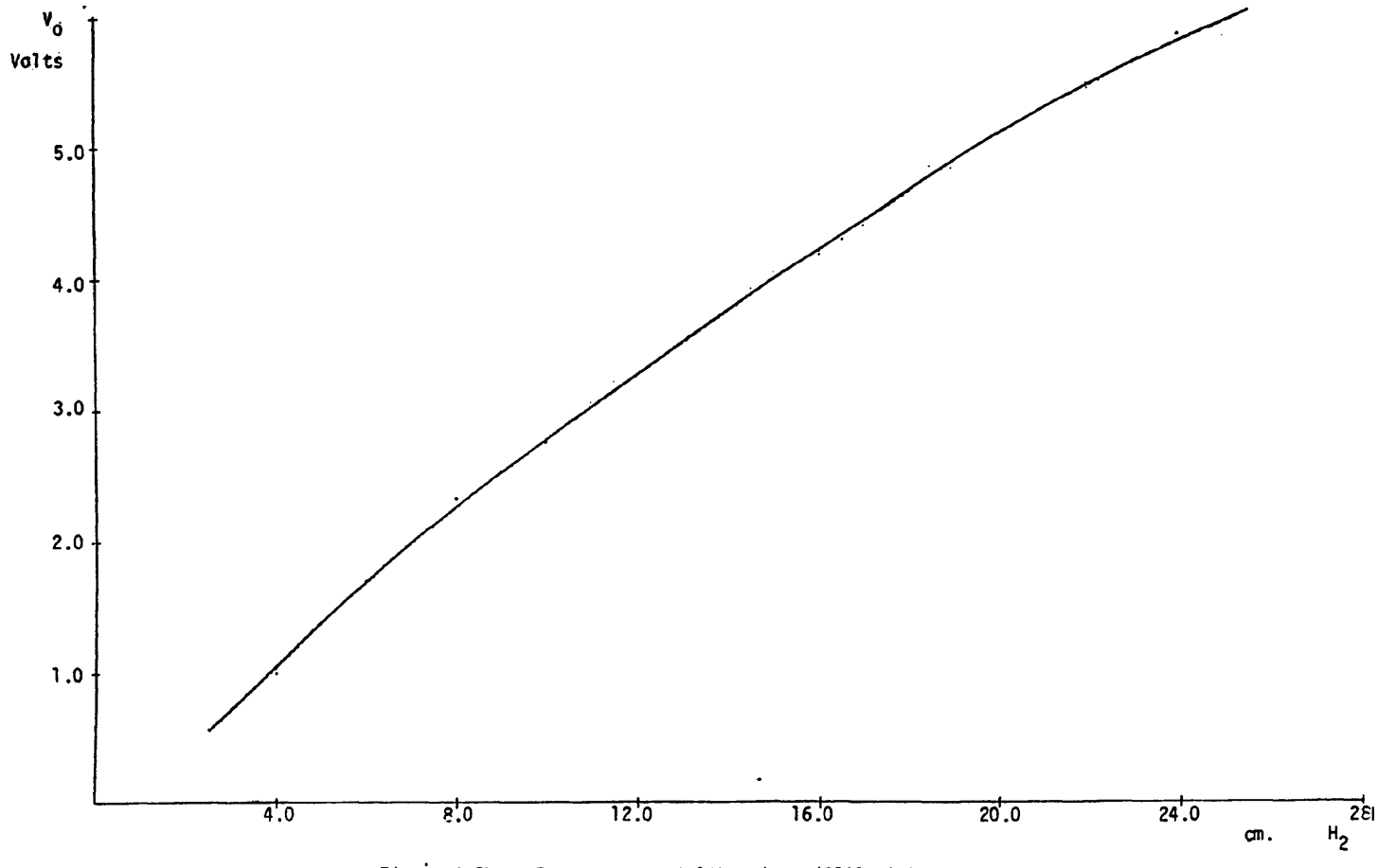


Figure 4.5b Depth Sensor Calibration (SISO rig).

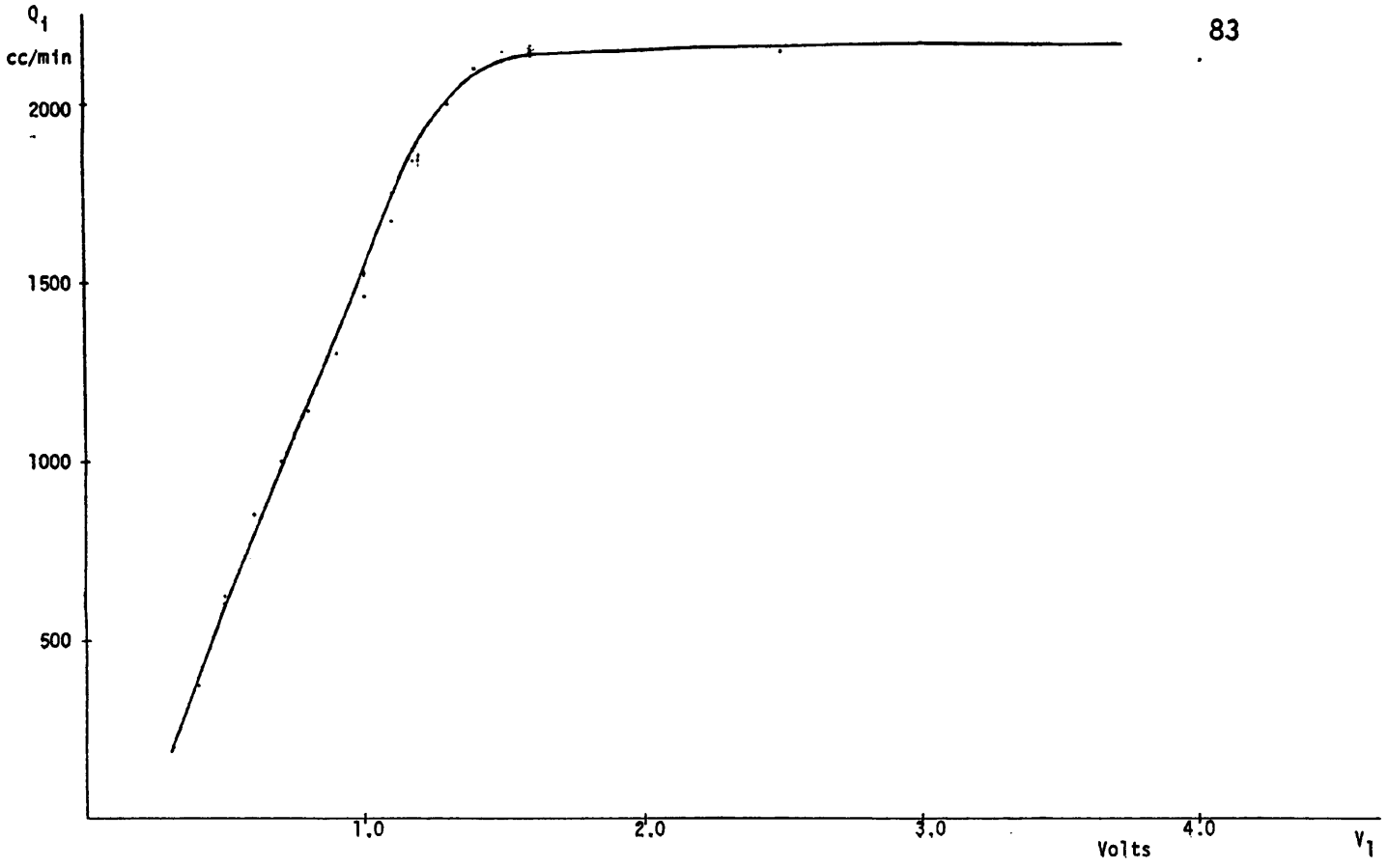


Figure 4.6a Pump Characteristics (MIMO rig) Tank 1.

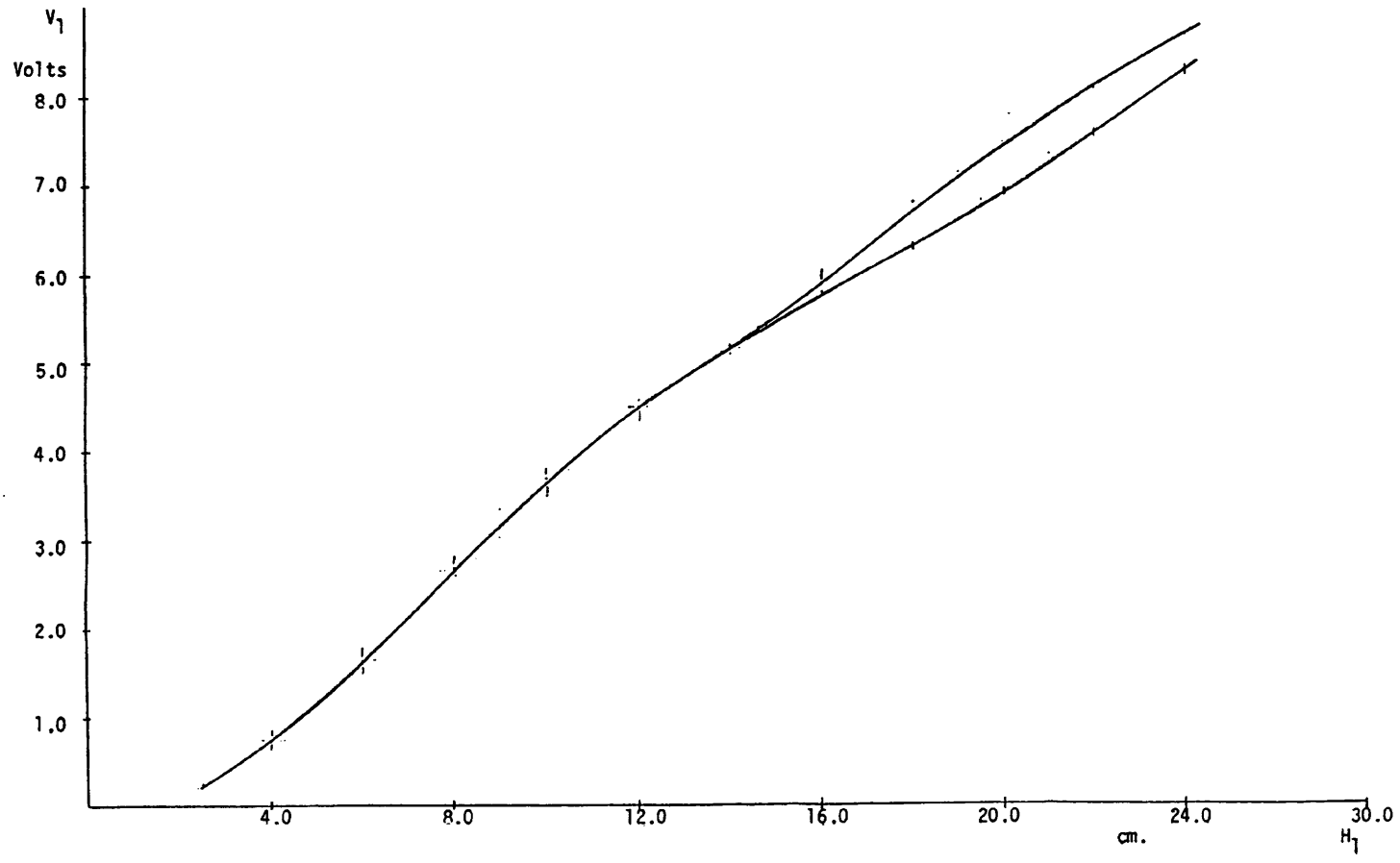


Figure 4.6b Depth Sensor Calibration (MIMO rig) Tank 1.

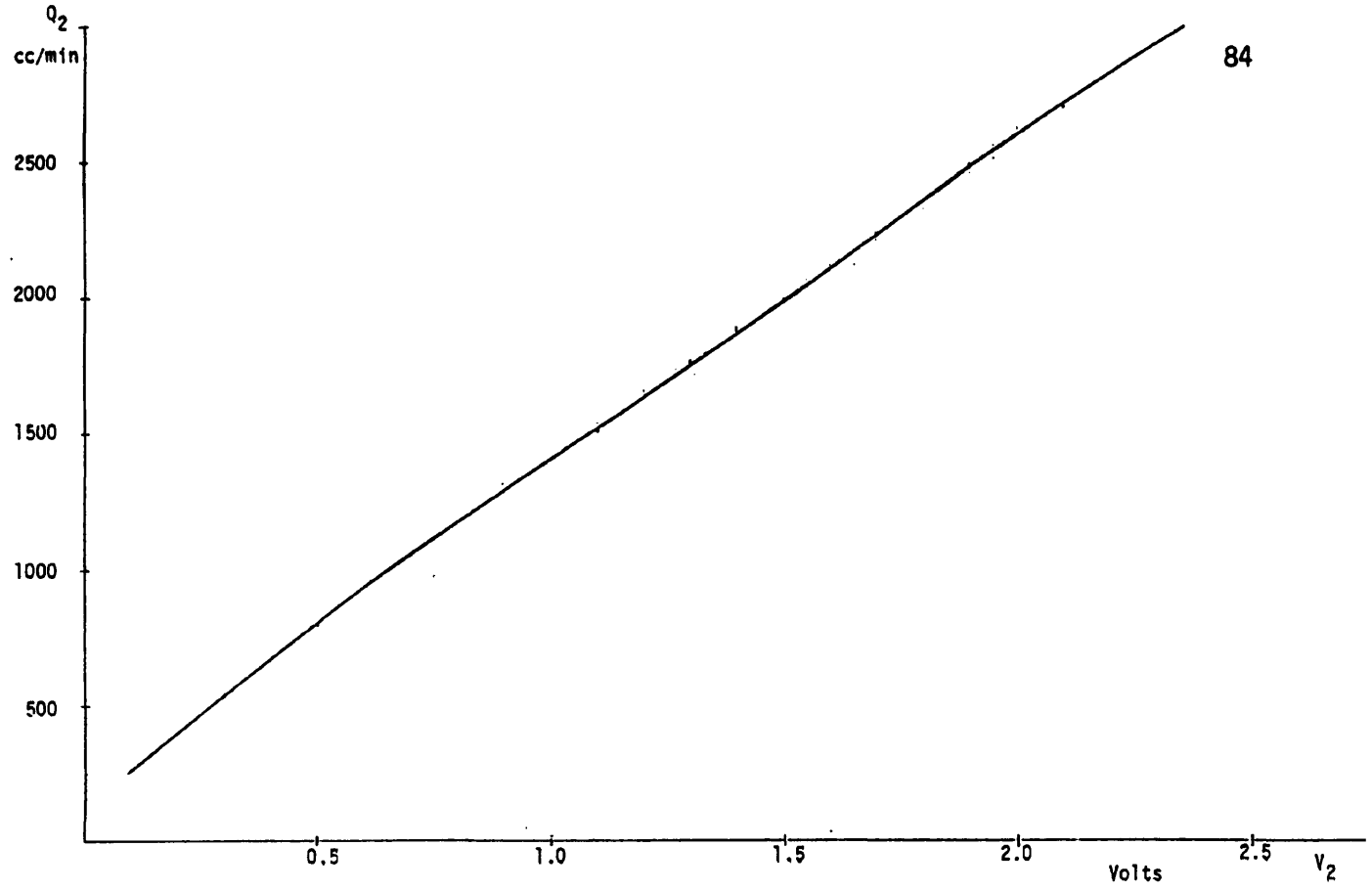


Figure 4.6c Pump Characteristics (MIMO rig) Tank 2.

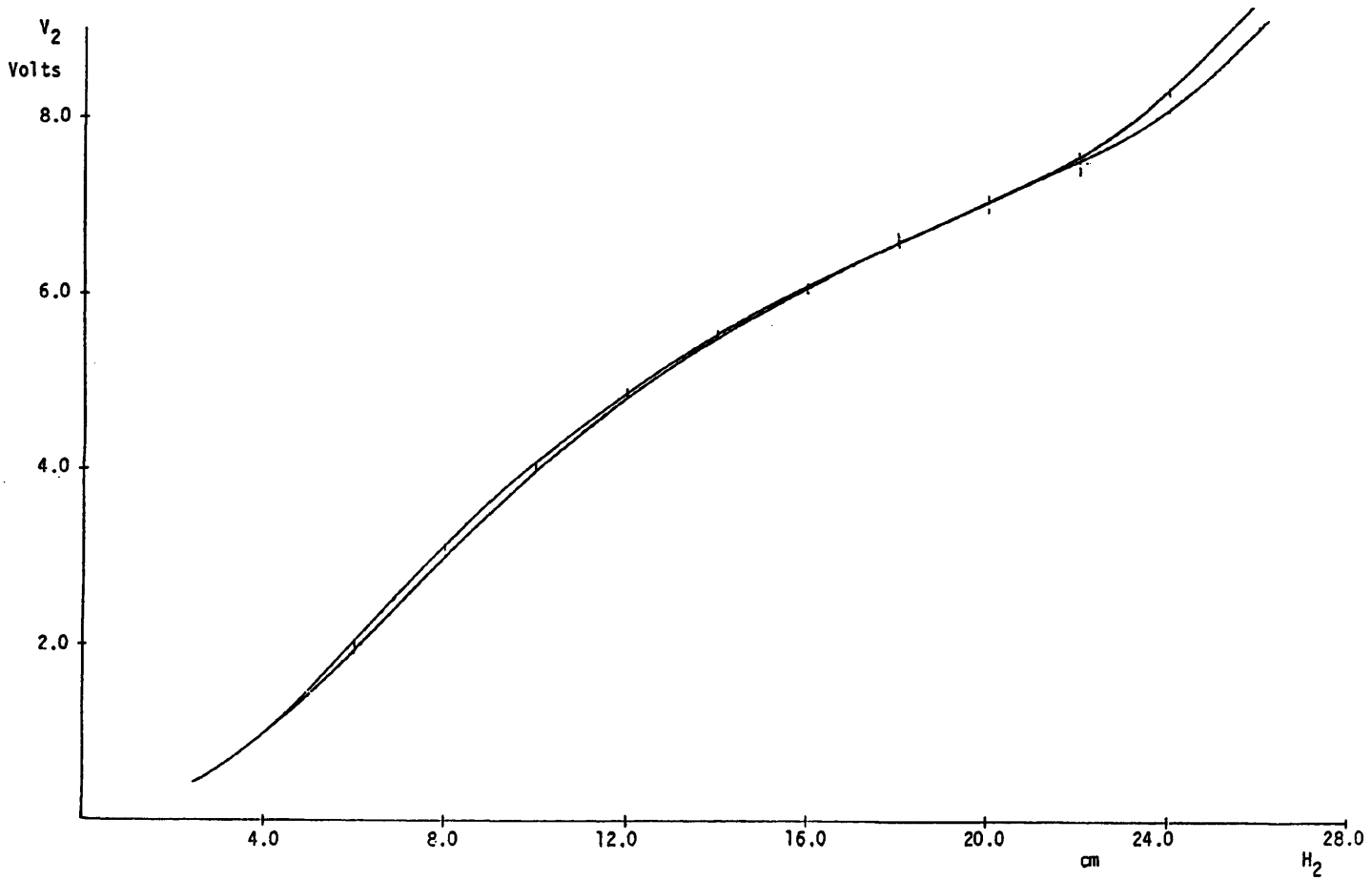


Figure 4.6d Depth Sensor Calibration (MIMO rig) Tank 2.

where V_2 = volume of fluid in tank 2

H_2 = the fluid height in tank 2

Q_0 = flow rate of fluid out of tank 2.

If the inter-tank holes and the drain tap are assumed to behave like orifices then the following equations arise from characteristic relationships for orifices:

$$Q_1 = c_{d_1} a_1 \sqrt{2g(H_1-H_2)} \tag{4.3.3}$$

$$Q_0 = c_{d_2} a_2 \sqrt{2g(H_2-H_3)} \tag{4.3.4}$$

where c_{d_1}, c_{d_2} = discharge coefficients (0.6 for a sharp orifice)

H_3 = height of drain tap

g = gravitational constant

a_1 = cross-sectional area of orifice between tanks 1 and 2

a_2 = cross-sectional area of orifice on tank 2's tap.

Equations (4.3.1) to (4.3.4) describe the system dynamics in their nonlinear form. For control systems studies the equations are linearised by considering small variations q_i in Q_i, q_1 in Q_1, q_0 in Q_0, h_1 in H_1 and h_2 in H_2. Thus the equations become:

$$\left.\begin{aligned} A\frac{dh_1}{dt} &= q_i - K_1(h_1-h_2) \\ A\frac{dh_2}{dt} &= K_1(h_1-h_3) - K_2h_2 \end{aligned}\right\} \tag{4.3.5}$$

where

$$K_1 = \frac{c_{d_1}a_1\sqrt{2g}}{2\sqrt{H_1-H_2}} \quad ; \quad K_2 = \frac{c_{d_2}a_2\sqrt{2g}}{2\sqrt{H_2-H_3}} \tag{4.3.6}$$

Rearranging equation (4.3.5) and using $\dot{h} = dh/dt$, one gets the following state space model:

$$\begin{bmatrix} \dot{h}_1 \\ \dot{h}_2 \end{bmatrix} = \begin{bmatrix} -K_1/A & K_1/A \\ K_1/A & -(K_1+K_2)/A \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} + \begin{bmatrix} 1/A \\ 0 \end{bmatrix} q_i \qquad (4.3.7)$$

Thus, taking the Laplace transform of the above, the transfer function is obtained as:

$$\frac{h_2(s)}{q_i(s)} = \frac{1/K_2}{(A^2/K_1K_2)s^2+(A(2K_1+K_2)/K_1K_2)s+1} \qquad (4.3.8a)$$

$$= \frac{1/K_2}{(sT_1+1)(sT_2+1)} \qquad (4.3.8b)$$

where $$T_1T_2 = \frac{A^2}{K_1K_2} \quad ; \quad T_1+T_2 = \frac{A(2K_1+K_2)}{K_1K_2} \qquad (4.3.8c)$$

The values of K_1 and K_2 can be calculated using static experiments, i.e. steady state operating levels, when $dH_1/dt = dH_2/dt = 0$ which means $Q_i = Q_1 = Q_0$. Thus, if the flow Q_i is measurable (which it is by the flow meter), then:

$$\left.\begin{aligned} c_{d_1}a_1\sqrt{2g} &= \frac{Q_i}{\sqrt{H_1-H_2}} \\ c_{d_2}a_2\sqrt{2g} &= \frac{Q_i}{\sqrt{H_2-H_3}} \end{aligned}\right\} \qquad (4.3.9)$$

from which K_1,K_2 are then derived.

The above equations show that the rig is a second order system; but the removal of all the bungs between the two tanks, however, reduces the model of the apparatus to a first order system. This can easily be proved once K_1 becomes infinity and equation (4.3.8a) reduces to:

$$\frac{h_2(s)}{q_i(s)} = \frac{1/K_2}{sT+1} \qquad \text{where} \quad T = \frac{2A}{K_2}$$

As mentioned earlier, at different operating points the system dynamics alter and thus the system model as well. Examples of the different system equations for different operating points are shown below.

MODEL A:

$$Q_i = 1000cc/min,\ H_3 = 2.5cm,\ H_2 = 5.0cm,\ H_1 = 6.7cm$$

$$\frac{V_2}{V_i} = \frac{4.7035x10^{-4}}{s^2+0.0657s+0.0004085} = \frac{4.7035x10^{-4}}{(s+0.126)(s+0.00536)}$$

MODEL B:

$$Q_i = 2000cc/min,\ H_3 = 2.5cm,\ H_2 = 10.1cm,\ H_1 = 15.4cm$$

$$\frac{V_2}{V_i} = \frac{3.0692x10^{-4}}{s^2+0.0417s+0.000169} = \frac{3.0692x10^{-4}}{(s+0.0372)(s+0.0046)}$$

In [63] the relationship between the operating fluid heights and time constants is shown to be:

$$T_1 \propto (H_1-H_2)^{1/2}$$
$$T_2 \propto (H_2)^{1/2}$$

4.3.3 The MIMO Model Derivation

In a similar manner to the S.I.S.O. rig derivation the multivariable tanks system results in the following state-space model using Figure 4.4:

$$\begin{bmatrix} \dot{h}_1 \\ \dot{h}_2 \end{bmatrix} = \begin{bmatrix} -(K_1+K_3)/A & K_1/A \\ K_1/A & -(K_1+K_2)/A \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} + \begin{bmatrix} 1/A & 0 \\ 0 & 1/A \end{bmatrix} \begin{bmatrix} q_{i1} \\ q_{i2} \end{bmatrix} \quad (4.3.10a)$$

where A, K_1, K_2 are as defined in Section 4.3.2 and

$$K_3 = \frac{c_{d_2} a_2 \sqrt{2g}}{2\sqrt{H_1 - H_3}} \qquad (4.3.10b)$$

The transfer function matrix relating the inputs $q_{i1}(s)$, $q_{i2}(s)$ to the outputs $h_1(s)$, $h_2(s)$ is found directly from the above equation as

$$\begin{bmatrix} h_1(s) \\ h_2(s) \end{bmatrix} = \frac{1}{A\Delta s} \begin{bmatrix} s + \frac{K_1+K_3}{A} & K_1/A \\ K_1/A & s + \frac{K_1+K_2}{A} \end{bmatrix} \begin{bmatrix} q_{i1}(s) \\ q_{i2}(s) \end{bmatrix} \qquad (4.3.11a)$$

where

$$\Delta s = \text{determinant of } \begin{bmatrix} s + \frac{K_1+K_3}{A} & -K_1/A \\ -K_1/A & s + \frac{K_1+K_2}{A} \end{bmatrix} \qquad (4.3.11b)$$

The block diagram of the multivariable coupled tanks is shown in Figure 4.4b. From the above it is seen that the coupling between the two tank outputs is determined by the coefficient K_1, which also determines the influence of both inputs as well. Equation (4.6) shows that K_1 is proportional to a_1 the cross-sectional area of the orifice between the two tanks. When all the holes between the tanks are plugged, the two tanks become two independent first-order systems as K_1 becomes zero.

4.4 MODEL REFERENCE ADAPTIVE CONTROL OF THE TANKS

4.4.1 Introduction

Basically, during most of the control experiments conducted on the tanks, an intentional undermodelled reference model was used, i.e. a first-order reference model. The main reason for this was to allow the study of the robustness properties of the various algorithms to mismodelling. The first to be discussed will be the SISO coupled tanks.

4.4.2 SISO Tanks

Key to the Programs (see Appendix C):

The adaptive parameters:

$$\theta(t) = [\theta_1(t),\theta_2(t),\theta_3(t)] = [Y(11),Y(12),Y(13)]$$

The control input:

$$u(t) = Y(6) \qquad u(t-1) = Y(7)$$

The reference input:

$$R(t) = R$$

The reference output:

$$y_m(t) = Y(4), \qquad y_m(t-1) = Y(19)$$

The system output:

$$y_p(t) = YS , \qquad y_p(t-1) = Y(5)$$

The tracking error:

$$e(t) = Y(9) , \qquad e(t-1) = Y(10)$$

Three different reference models were used for the experiments based on the three algorithms which have been discussed in the previous chapters. The first two were first-order models while the third was a second-order reference model. The first model was intentionally fast as the main concern was the convergence time of both the reference and system outputs, while the second model was much slower to allow observations of how well the three different algorithms control the system output tracking of the reference model output.

The three models are:

(i) $\frac{0.375}{s+0.375}$

(ii) $\frac{0.005}{s+0.005}$

(iii) $\frac{0.000025}{s^2+0.006s+0.000025}$

This also allows the use of different reference time constants, especially as the plant time constants vary from as small as 2.75 seconds to as high as, say, 500 seconds.

Due to the undermodelling and the use of a zero-order hold (Z.O.H.) there are three adaptive parameters used as n = 1, m = 1, d = 1. The sampling period was chosen as Ts = 4.0 seconds, while a time delay of one sampling interval was assumed.

Control Hardware and Software:

The output voltage representing the tank output was the input to the BBC computer used for the schemes and the control input derived from the computer used to drive the pump. As mentioned earlier, the analogue to digital (A/D) converter used had a zero-order-hold embedded in it, while the software for the algorithms was written in BASIC. A listing of these programs can be seen in the Appendix C. The results were first stored on floppy discs before being plotted using a Rikadenki plotter in the laboratory for better result presentation.

Real-Time Control Experiments (the Results):

Using the sampling period of four seconds the reference models were converted to discrete-time transfer functions used in the programs on the

BBC microcomputer. The real time control on-line of the tanks using the different models and algorithms plus their results are now explained and shown graphically.

(i) Model 1: $y_m(t) = 0.7768698r(t) + 0.2231301ym(t-1)$; R = 3.0V

A series of experiments were conducted using the three different algorithms already mentioned in the previous chapters. For the Goodwin algorithm (Basic MRAC) the initial parameter vector was P = [0, 0, 0], while for the Ortega et al algorithm it was P = [0.1, 0.1, 0.1], and for the Kresselmeier and Anderson algorithm it was also P = [0.1, 0.1, 0.1]. The results of the runs are shown in the following pages as follows:

(a) The Goodwin algorithm: Figure 4.7a shows the Reference and System outputs; Figure 4.7b the Control input, while Figure 4.7c shows the adaptive parameters.

(b) The Ortega et al algorithm: the results of this run are depicted in Figure 4.8a showing the Reference and System outputs; Figure 4.8b the Control input, and Figure 4.8c the adaptive parameters.

(c) The Kreisselmeier-Anderson algorithm: Figure 4.9a shows the Reference and System outputs; Figure 4.9b the Control input and Figure 4.9c the adaptive parameters.

By varying the adaptive gains of the Goodwin or Ortega et al algorithms the convergence time of the system output to the reference model output could be shortened, but unfortunately this leads to an increase in the amount of overshoot. Figure 4.10 shows the effect of tripling the adaptive gain value for the Goodwin algorithm as compared to Figure 4.7a.

As mentioned earlier in Chapter 3, by varying the value of $\mu \in (0,1)$ in the normalization factor equation (3.4.12) for the Ortega et al algorithm it can be made to be either more robust or alert, depending on its value. The

lower it is the more alert it becomes. This is shown by comparison of Figures 4.11a and b where $\mu = 0.75$ and $\mu = 0.15$ respectively.

(ii) Model 2: $y_m(t) = 0.0198013r(t) + 0.9801986y_m(t-1)$; R = 4.0 Volts

Using this model the basic idea was to observe the tracking abilities of the different algorithms and also look at the control inputs and adaptive parameters. As before, the experiments start off with (a) the Goodwin algorithm, with Figures 4.12a,b,c showing the Reference and System outputs, the Control input and the adaptive parameters, respectively. An adaptive gain of 0.12 was used; (b) the Ortega et al algorithm; the results are shown in Figure 4.13a for the Reference and System outputs; Figure 4.13b the Control input, and lastly in Figure 4.13c the adaptive parameters. (c) Similar results for the Kreisselmeier and Anderson algorithm are shown in Figures 4.14a,b,c for the Reference and System outputs, the Control input and the adaptive parameters respectively.

(iii) Model 3: $y_m(t) = 0.000413r(t-1) + 1.97589y_m(t-1) - 0.9763y_m(t-2)$

Owing to hardware limitations, i.e. available memory on the BBC micro, the plots of all the adaptive parameters could not be obtained, thus the first adaptive parameter Y(11) was plotted for the two algorithms used. These were the Goodwin and the Ortega et al algorithms. The absence of the Kreisselmeier and Anderson algorithm is based on its structure, where the reference model order is the same as that of the system delay, which in this case is one. Since the reference model used was a second-order system there was need for a longer period of time for the experimental run so as to allow adequate tracking and convergence observations, but the sampling period of four seconds still remained.

(a) The results of the Goodwin algorithm run are shown in Figure 4.15a which depicts the System and Reference outputs and Figure 4.15b which shows the control input and an adaptive parameter.

(b) The Ortega et al algorithm results are shown in Figures 4.15c and d respectively, with the Reference and System outputs shown in Figure 4.15c, while the Control input and the adaptive parameter are shown in Figure 4.15d.

4.4.3 The MIMO Tanks

Key to the Programs:

$y_1(t) = Y(1)$, $y_1(t-1) = Y(2)$

$y_2(t) = Y(3)$, $y_2(t-1) = Y(4)$

$u_1(t) = Y(47)$, $u_1(t-1) = Y(5)$, $u_1(t-2) = Y(6)$

$u_2(t) = Y(48)$, $u_2(t-1) = Y(7)$, $u_2(t-2) = Y(8)$

$y_{m1}(t) = Y(9)$, $y_{m1}(t-1) = Y(10)$, $y_{m1}(t-2) = Y(65)$

$y_{m2}(t) = Y(17)$, $y_{m2}(t-1) = Y(18)$, $y_{m2}(t-2) = Y(66)$

$e_1(t) = Y(19)$

$e_2(t) = Y(20)$

$P_{11} = Y(27)$, $P_{12} = Y(28)$, $P_{21} = Y(37)$, $P_{22} = Y(38)$

$$\theta(t) = \begin{bmatrix} \theta_{11}(t) & \theta_{12}(t) & \theta_{13}(t) & \theta_{14}(t) & \theta_{15}(t) & \theta_{16}(t) \\ \theta_{21}(t) & \theta_{22}(t) & \theta_{23}(t) & \theta_{24}(t) & \theta_{25}(t) & \theta_{26}(t) \end{bmatrix}$$

$$= \begin{bmatrix} Y(11) & Y(12) & Y(13) & Y(14) & Y(15) & Y(16) \\ Y(21) & Y(22) & Y(23) & Y(24) & Y(25) & Y(26) \end{bmatrix}$$

For the MIMO tanks, three algorithms were used for the experiments carried out. The first was the Basic MIMO scheme based on the Goodwin algorithm, the second was the derived MIMO Kreisselmeier and Anderson algorithm of Section 3.6, while the third was a modified Goodwin algorithm mentioned in Section 3.7.1 . The reference models used were first-order systems with the coupling between the tanks acting as disturbances to the two tank outputs. Because the coupling of say input one to output two is second-order, a second order reference model was also used just for completeness.

The Real time control experiments (and results):

Due to the faster time responses of the multivariable tanks, a sampling period Ts of three seconds was used for all the experiments, and a time delay of one period assumed. The control objective was to independently control the water level in each tank of the coupled multivariable system by setting different voltages (and hence heights of water) for the two tanks. The coupling between the tanks is determined by the size of the orifice between them. Most of the work was done using a single hole with diameter 0.317 cm, but a few were done using two holes, with the additional orifice having a diameter of 0.635 cm. Unless otherwise stated, the results given here should be assumed to be for a single hole.

For each tank there were six adaptive parameters for an assumed first order transfer function with a zero order hold D/A converter. Unfortunately the results of only the tank outputs, the reference model outputs and the control inputs can be shown due to memory limitations of the hardware used.

The details of the results now follow, starting with the MIMO Goodwin algorithm.

(a) The MIMO Goodwin algorithm

(i) Model A for tanks 1,2

$$y_{m1}(t) = 0.2231301y_{m3}(t-1) + 0.7768698r_1(t); \quad R = 5.0V$$
$$y_{m2}(t) = 0.2231301y_{m2}(t-1) + 0.7768698r_2(t); \quad RI = 3.0V$$

Using various P-matrix structures, investigations were carried out into the effects of this on the control algorithm. Of interest was whether a diagonal P-matrix would suffice or whether a full 2x2 matrix was needed. The results of the various combinations are shown in Figures 4.16a,b,c.

For Figure 4.16a $P = \begin{pmatrix} 0.12 & 0 \\ 0 & 0.12 \end{pmatrix}$ was used.

For Figure 4.16b $P = \begin{pmatrix} 0.12 & 0.04 \\ 0.04 & 0.12 \end{pmatrix}$ was used.

For Figure 4.16c $P = \begin{pmatrix} 0.12 & -0.04 \\ -0.04 & 0.12 \end{pmatrix}$ was used.

From the results the best P-matrix all round would be that for Figure 4.16c which combines a lower overshoot relatively with the others with a fast convergence time; but for simplicity the diagonal P-matrix was chosen, although it had a higher overshoot it was faster in converging its system output to the desired reference output. Using a diagonal P-matrix and smaller gains, i.e.

$$P = \begin{pmatrix} 0.04 & 0 \\ 0 & 0.04 \end{pmatrix}$$

a reduction in overshoot results but at the expense of having a slower convergence time compared with Figure 4.16a.

The effects of increasing the orifice size, and hence the coupling between the tanks, were also investigated. The results in Figure 4.18 show that using the same diagonal matrix as in Figure 4.16a the convergence time had increased for both tanks, but the overshoot for tank 1 was reduced while that of tank 2 had increased due to the extra coupling between them.

(ii) Model B for tanks 1,2 $y_{m1}(t) = 0.9802y_{m1}(t-1) + 0.0198r_1(t)$; R = 5.0V

$y_{m2}(t) = 0.9802y_{m2}(t-1) + 0.0198r_2(t)$; RI = 3.0V

The results of this, using a diagonal P-matrix, are shown in Figure 4.19.

(iii) Model C for tanks 1, 2

These are second-order reference models of which there are three of them. The results for the three models used are shown in:

Figure 4.20a using $y_{m1}(t) = 0.000303r_1(t-1) + 1.98186y_{m1}(t-1) - 0.98216y_{m1}(t-2)$

$y_{m2}(t) = 0.000303r_2(t-1) + 1.98186y_{m2}(t-1) - 0.98216y_{m2}(t-2)$

$R = 5.0V, \quad RI = 3.0V$

Figure 4.20b using $y_{m1}(t) = 0.000413r_1(t-1) + 1.97589y_{m1}(t-1) - 0.9763y_{m1}(t-2)$

$y_{m2}(t) = 0.000413r_2(t-1) + 1.97589y_{m2}(t-1) - 0.9763u_{m2}(t-2)$

$R = 4.0V, \quad RI = 3.0V$

Figure 4.20c using the same model as in Figure 4.20b with R = 3.0V, RI = 5.0V.

(b) The Extended Kreisselmeier and Anderson algorithm

Initially, when the Kreisselmeier and Anderson scheme was first extended, it was based on the MIMO Goodwin algorithm without any theoretical facts or proofs. Based on this, six adaptive parameters per tank were used as in the MIMO Goodwin, but during the experiments the two tank outputs were found not to be converging to the desired reference outputs. The results in Figure 4.21a show that there was some form of tracking though. Going back and using a theoretical approach (see Section 3.6) it was found that two of the adaptive parameters, namely Θ_{12} and Θ_{21}, were zero in value. Subsequent experiments without these parameters, see Figure 4.21b, later proved this with the reduction if not virtual elimination of the offsets. The reference models used for the tanks were:

(i) Tank 1 $y_{m1}(t) = 0.7768698r_1(t) + 0.2231301y_{m1}(t-1); \quad R = 5.0V$

Tank 2 $y_{m2}(t) = 0.7768698r_2(t) + 0.2231301y_{m2}(t-1); \quad RI = 3.0V$

The importance of smaller dead zones is reflected by comparing the results shown in Figures 4.22a and b, where two different dead zones were used.

Figure 4.22a shows the results obtained using $d_o = 0.5$, the computed value for tank 1, and $d_o = 0.4$ for tank 2. For Figure 4.22b a reduced dead zone was used with $d_o = 0.05$ for both tanks. Comparisons show how greatly reduced the tracking error value was, with better tracking and convergence of the outputs.

(ii) Model B $\quad y_{m1}(t) = 0.9082y_{m1}(t-1) + 0.0198r_1(t)$; $\quad R = 5.0V$

$\qquad\qquad\quad y_{m2}(t) = 0.9082y_{m2}(t-1) + 0.0198r_2(t)$; $\quad RI = 3.0V$

The result of using the above model is shown in Figure 4.23, where the graphs are those of the reference outputs, system outputs and the control inputs.

(c) The Modified MIMO Goodwin algorithm

A few experiments were carried out using this algorithm. The first experiment was made without bounds on the adaptive parameters (see Figure 4.24a) so as to have a rough idea of the minimum and maximum values of these parameters. The same model, Model A, as in the previous algorithms was used. Using the knowledge gained from this, the bounds were placed on the parameters and the experiment repeated. The result, shown in Figure 4.24b, shows the reduction in the amount of overshoot compared to the earlier figure.

4.4.4 More Robust Considerations

Apart from investigating the robustness of the algorithms to under-modelling and mismodelling, some other results were obtained for reference input changes on-line. Also introduced were disturbances to check whether the algorithms might not cope with these and hence throw the system into instability.

The test results shown in this section are for the multivariable algorithms for which the reference models are first-order systems. The same model

$y_m(t) = 0.7768698\ r(t) + 0.2231301\ y_m(t-1)$ is used for both tanks and for all tests carried out. Figure 4.25a shows the result of changing the set points for tank 1 from 5.0 volts to 3.0 volts and tank 2 from 3.0 volts to 4.0 volts, using the MIMO Goodwin algorithm, while Figure 4.25b shows that of the extended Kresselmeier and Anderson algorithm, changing the setpoints of tank 1 from 3.0 volts to 5.0 volts and tank 2 from 5.0 volts to 3.0 volts.

Testing the robustness of the algorithms against external disturbances involved closing the taps on both tanks fully for varying time intervals and opening them again, or keeping the taps half-closed and observing what happens.

In Figure 4.26a, using the Goodwin algorithm, the tap on tank 1 was closed for three sampling periods at 161, while after allowing for the system to settle down at 267 both taps on the tanks were half-closed. In Figure 4.26b, using the same algorithm still, the tap on tank 2 was closed for three sampling periods at 150, while the tap on tank 1 was closed for a sampling period at 267. Using the extended Kresselmeier and Anderson in Figure 4.26c, the tap on tank 2 was closed for two sampling periods at 177, while the tap on tank 1 was closed first at 250 for three sampling periods and then again at 304 for eight sampling periods.

4.5 COMMENTS

(i) A lot of problems were confronted during the implementation of these algorithms on the two rigs. The most recurring problem was that of the characteristics of the depth sensor which were found to be varying from day to day during the research work (see Figures 4.6b,d for examples). This leads to having different results for the same models at different times. By this it is meant that while in all cases the tracking error ultimately reduces to zero, the time taken and the peak overshoots for the same reference outputs were not the same. In a way this shows how suited the

coupled tanks were to model reference adaptive control since the outputs can always be kept the same irrespective of the changes.

Another problem was that of quantization due to the A/D and D/A converters used. For example, in discretization 0 - 255 is equivalent to 0 - 10 volts, which means 1 ≈ 0.04 volts, but 0.04 volts on the tanks' depth sensor could be equivalent to 1 mm and hence makes convergence more difficult.

(ii) Fast reference models were used in most cases because it was the final output tracking that the author felt was of importance here, especially bearing in mind that for processes like those mentioned in Section 4.1 it is usually the aim to have the system output at a particular height irrespective of disturbances, say, but for completeness slow reference models were also investigated.

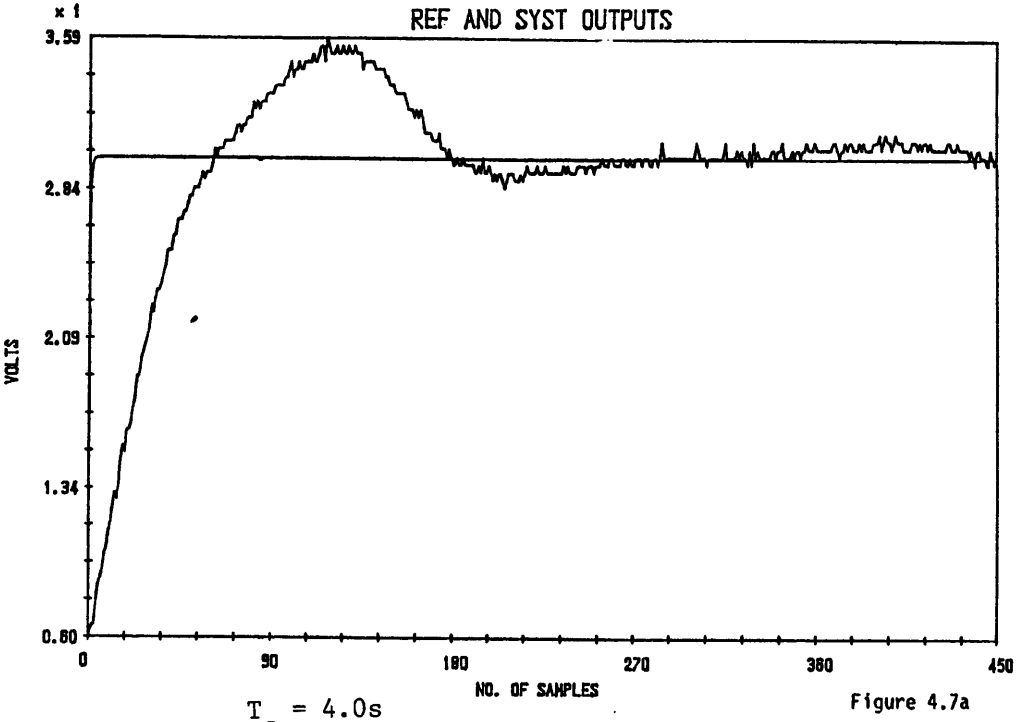
(iii) Although this is just one system and hence statements about comparisons of the different algorithms cannot be generalized from these results, some of the interesting points observed during the experiments cannot but be mentioned. In particular the Goodwin algorithm proved quite robust as it compared favourably with the other two algorithms which were implicitly designed for robustness. Probably the reason for this is the fact that even though some experiments were carried out with reduced order models (i.e. first order systems), the coupled tanks while being a second order system could be approximated by first order models. For example, observing the roots of the transfer functions in Section 4.4.2, it can be seen that the two transfer functions each have a dominant root which can approximate the system model, while in the case of the MIMO tanks it is the interactions that have second order transfer functions.

A particular disadvantage of the Kreisselmeier and Anderson algorithm it was felt was the lack of adaptive gains, unlike the other two which could be used to speed up the adaptation especially nearer to the desired reference

outputs. A way of getting round this though is by setting bigger parameters bounds, but this causes bigger overshoots while not necessarily solving the problem.

Also discovered was that the MIMO Goodwin coped better with the quantization error than the derived MIMO Kreisselmeier and Anderson algorithm.

(iv) In conclusion, three different versions of the model reference adaptive control scheme have been successfully employed to control the height of fluid in the tanks at different levels. The experimental results indicate the benefits gained from using M.R.A.C. for this type of system, as the set points could be changed on-line as well. It is felt that the robustness properties of these algorithms to mismodelling and disturbances have been shown.



T_s = 4.0s

Figure 4.7a

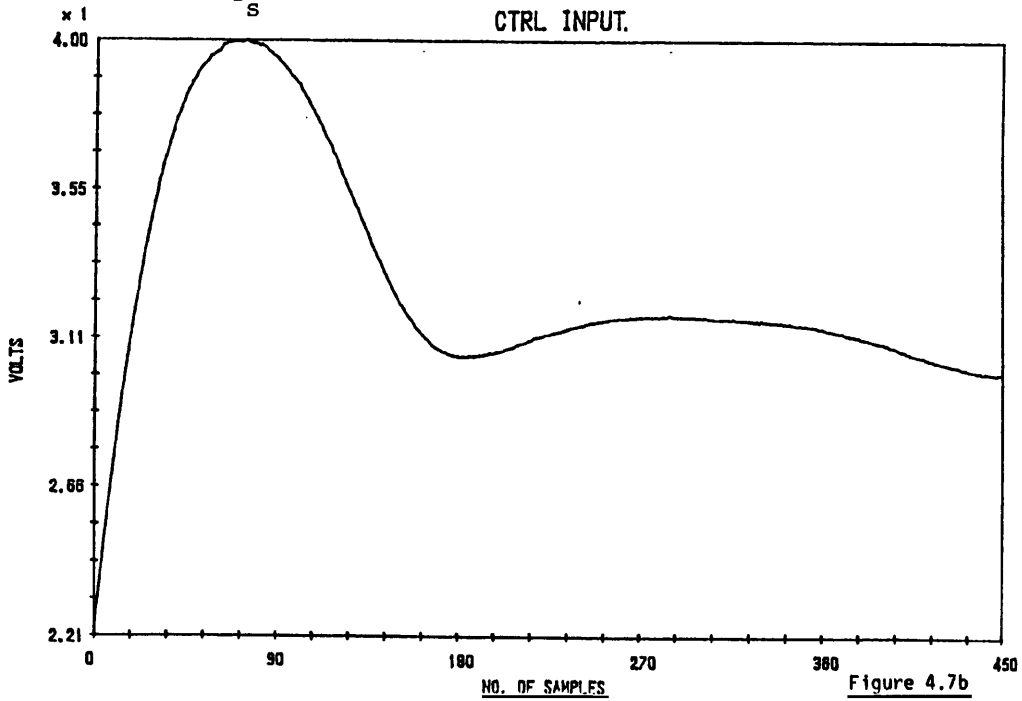


Figure 4.7b

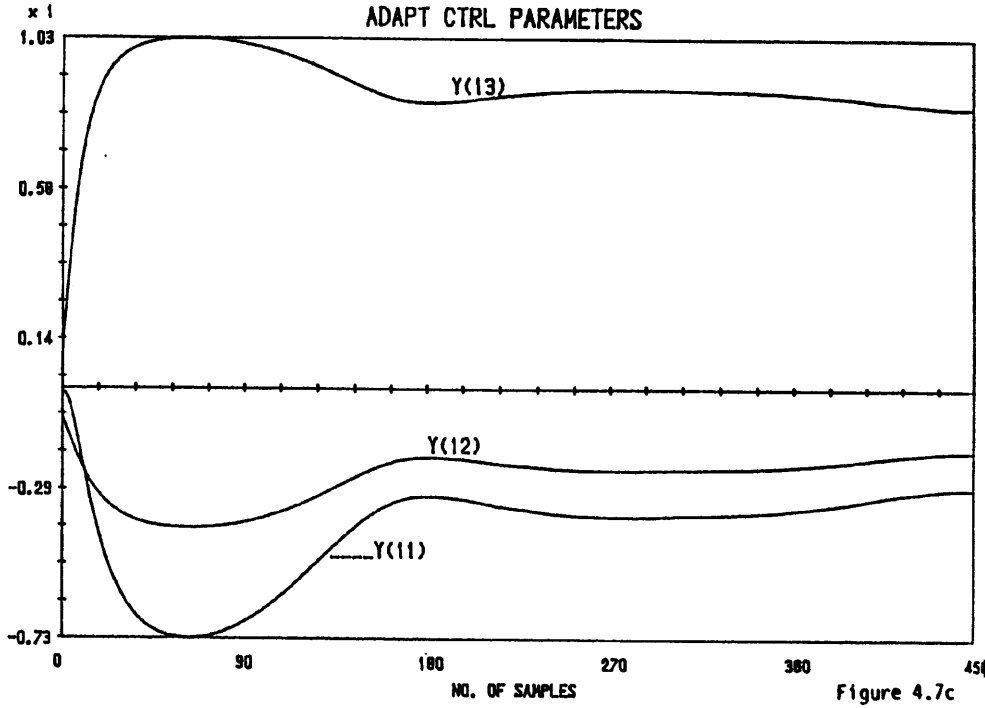


Figure 4.7c

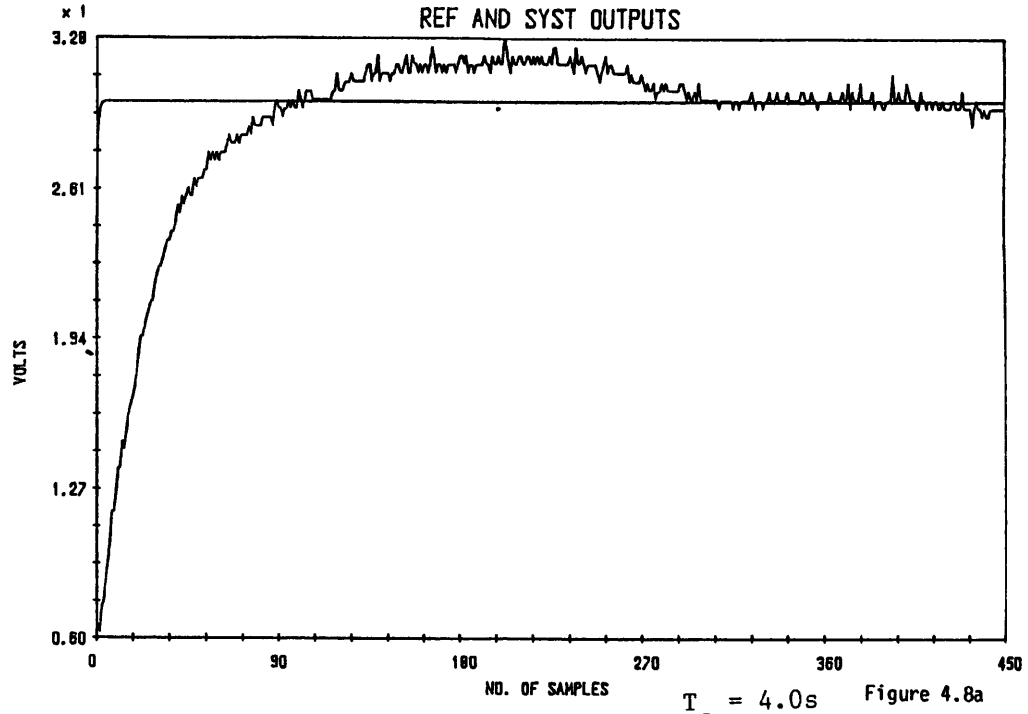
T_s = 4.0s

Figure 4.7 (a) The Reference and System Outputs using the Goodwin et al algorithm for the fast model;

(b) The Control Input u(t);

(c) The adaptive parameters θ(t).

T_s = 4.0s



T_s = 4.0s Figure 4.8a

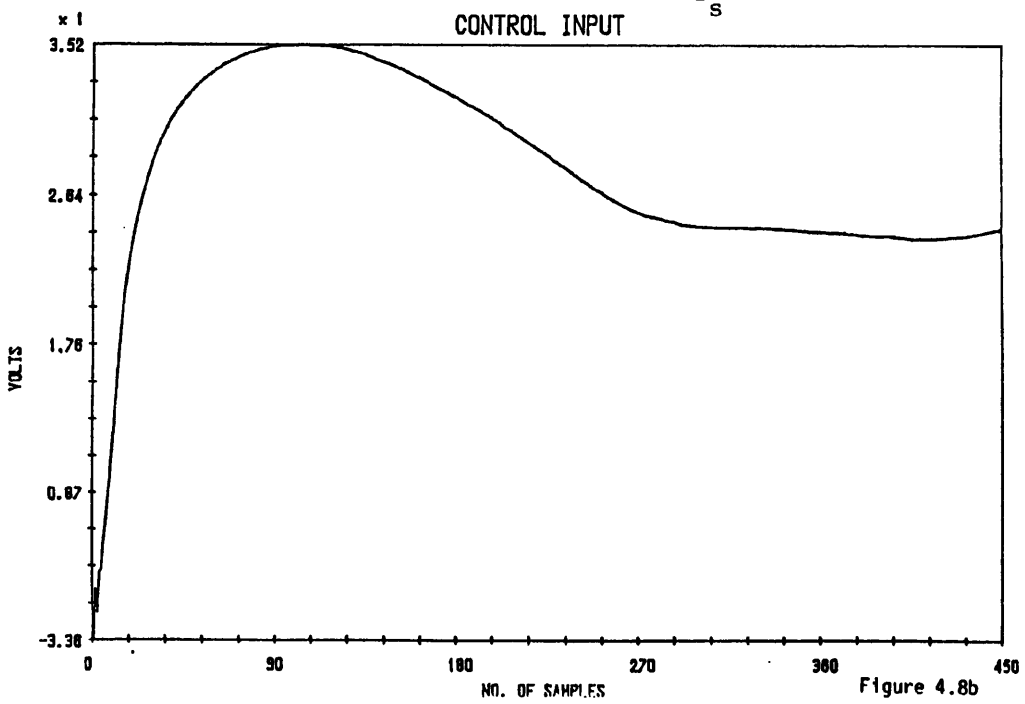


Figure 4.8b

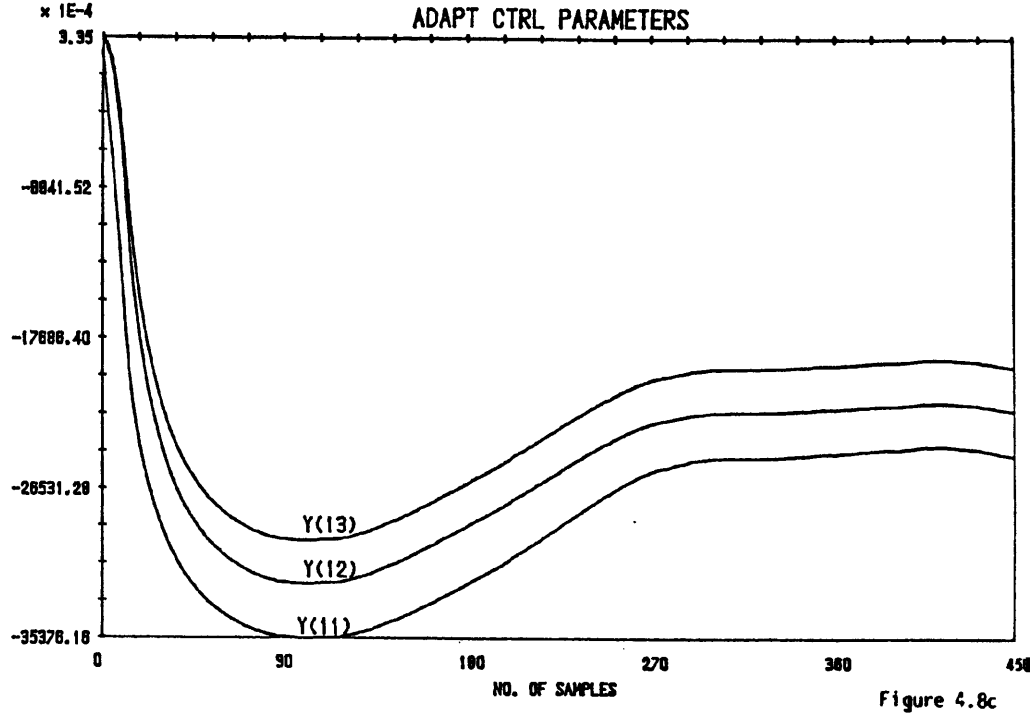


Figure 4.8c

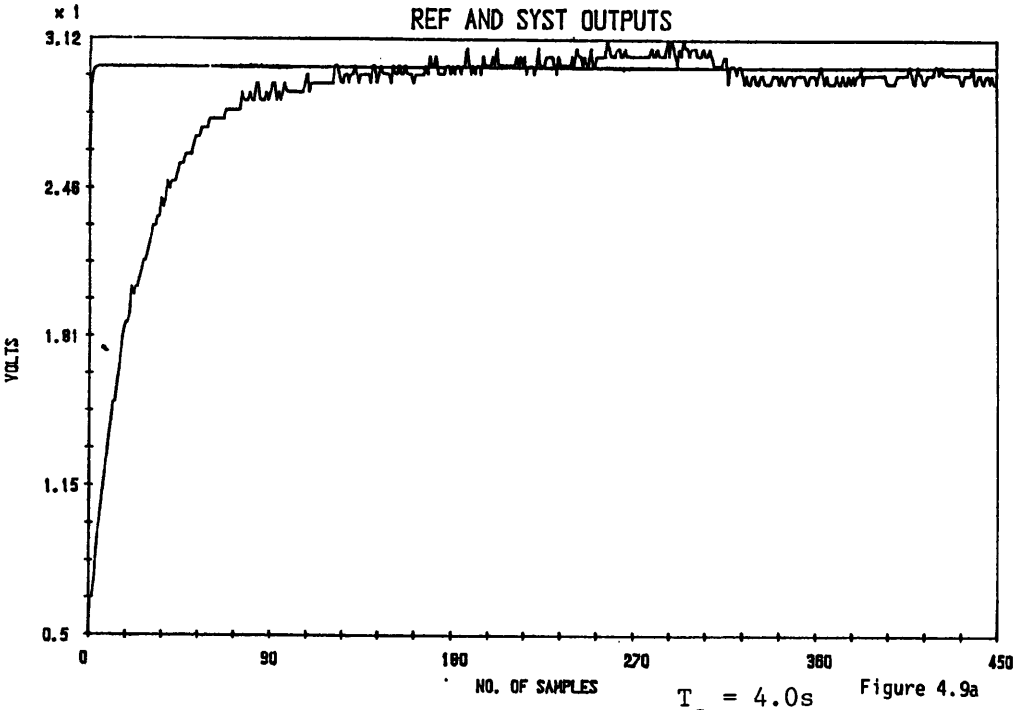
T_s = 4.0s

Figure 4.8 (a) The Reference and System Outputs using the Ortega et al algorithm for the fast Reference model;

(b) The Control Input u(t);

(c) The adaptive parameters θ(t).

T_s = 4.0s



T_s = 4.0s

Figure 4.9a

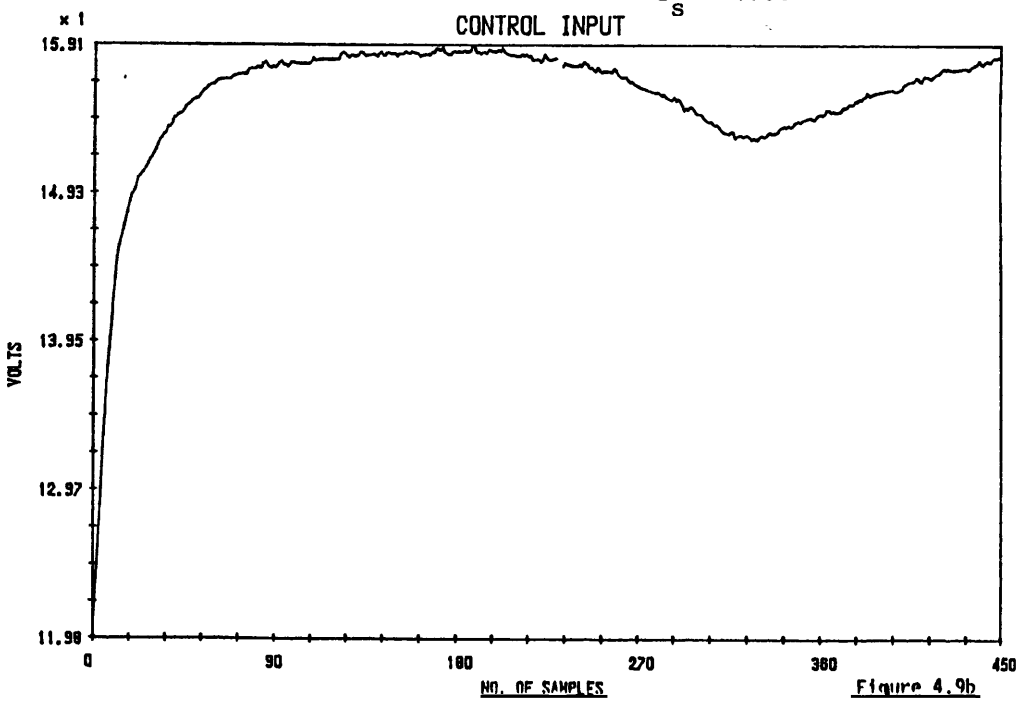


Figure 4.9b

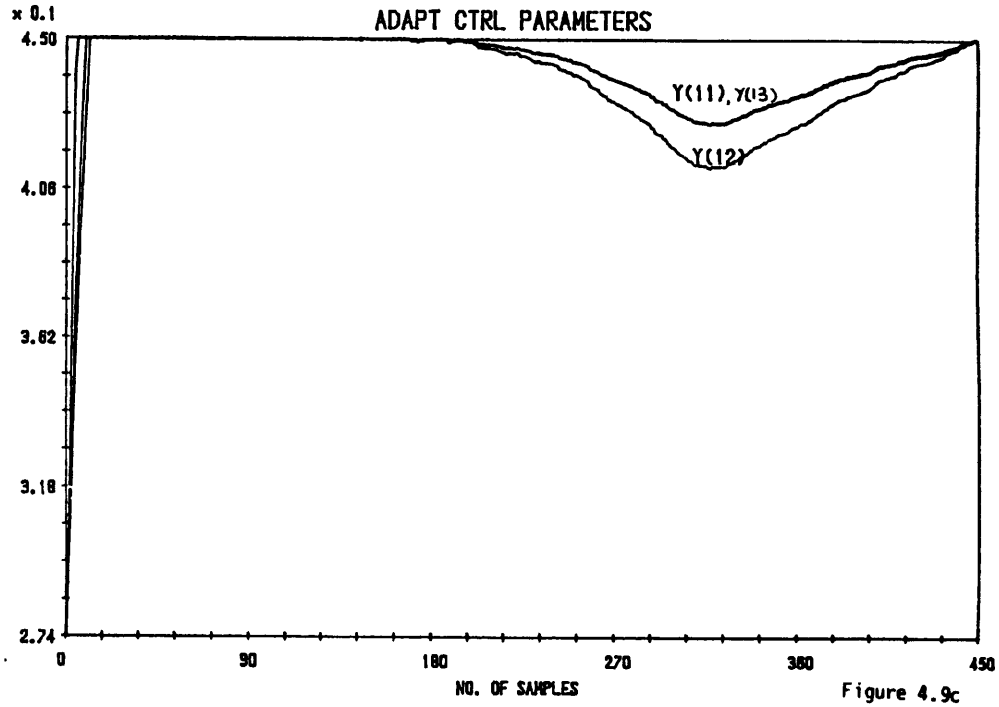


Figure 4.9c

T_s = 4.0s

Figure 4.9 (a) The Reference and System Outputs using the Kreisselmeier and Anderson algorithm for the fast Reference model;

(b) The Control Input u(t);

(c) The adaptive parameters θ(t).

T_s = 4.0s

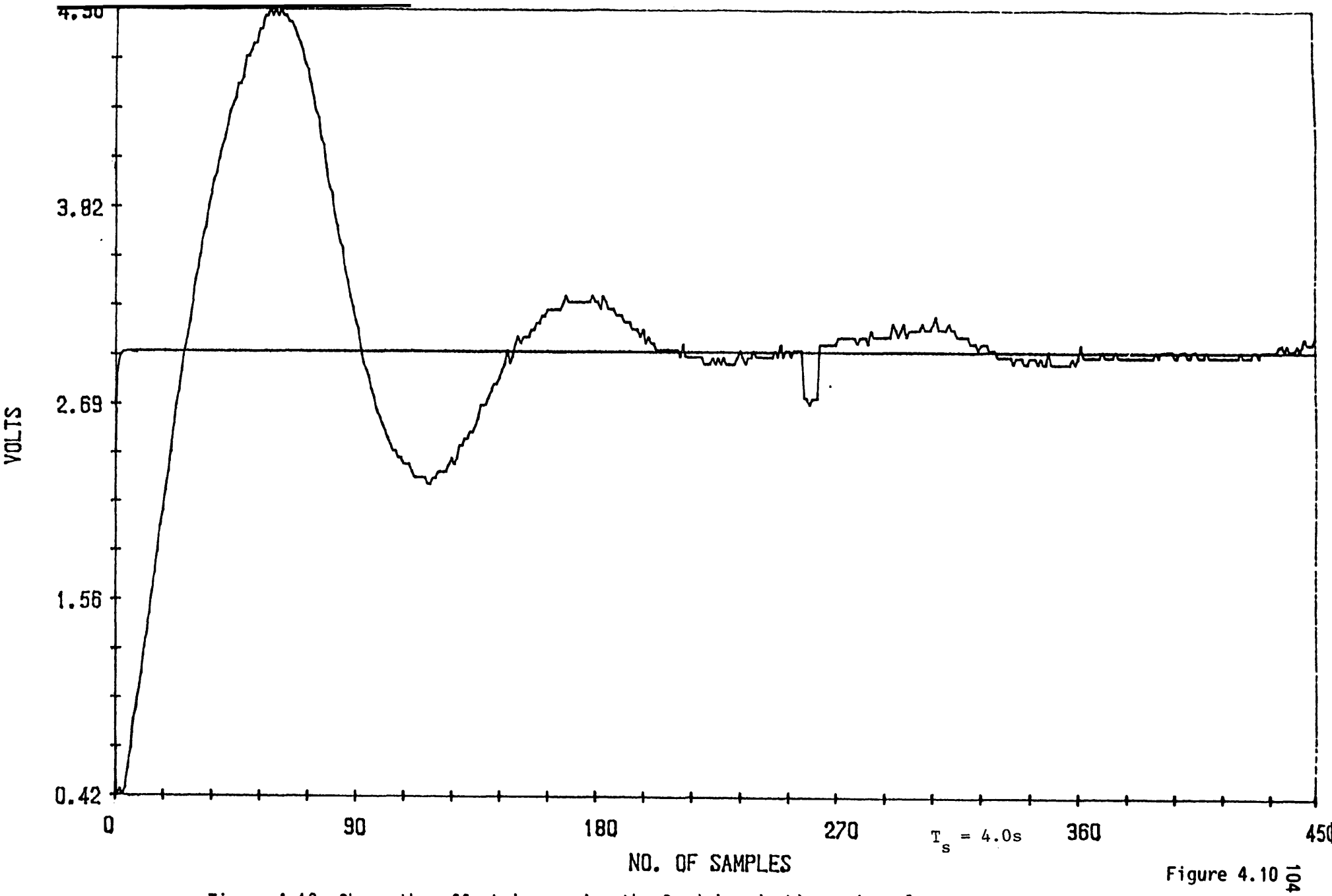
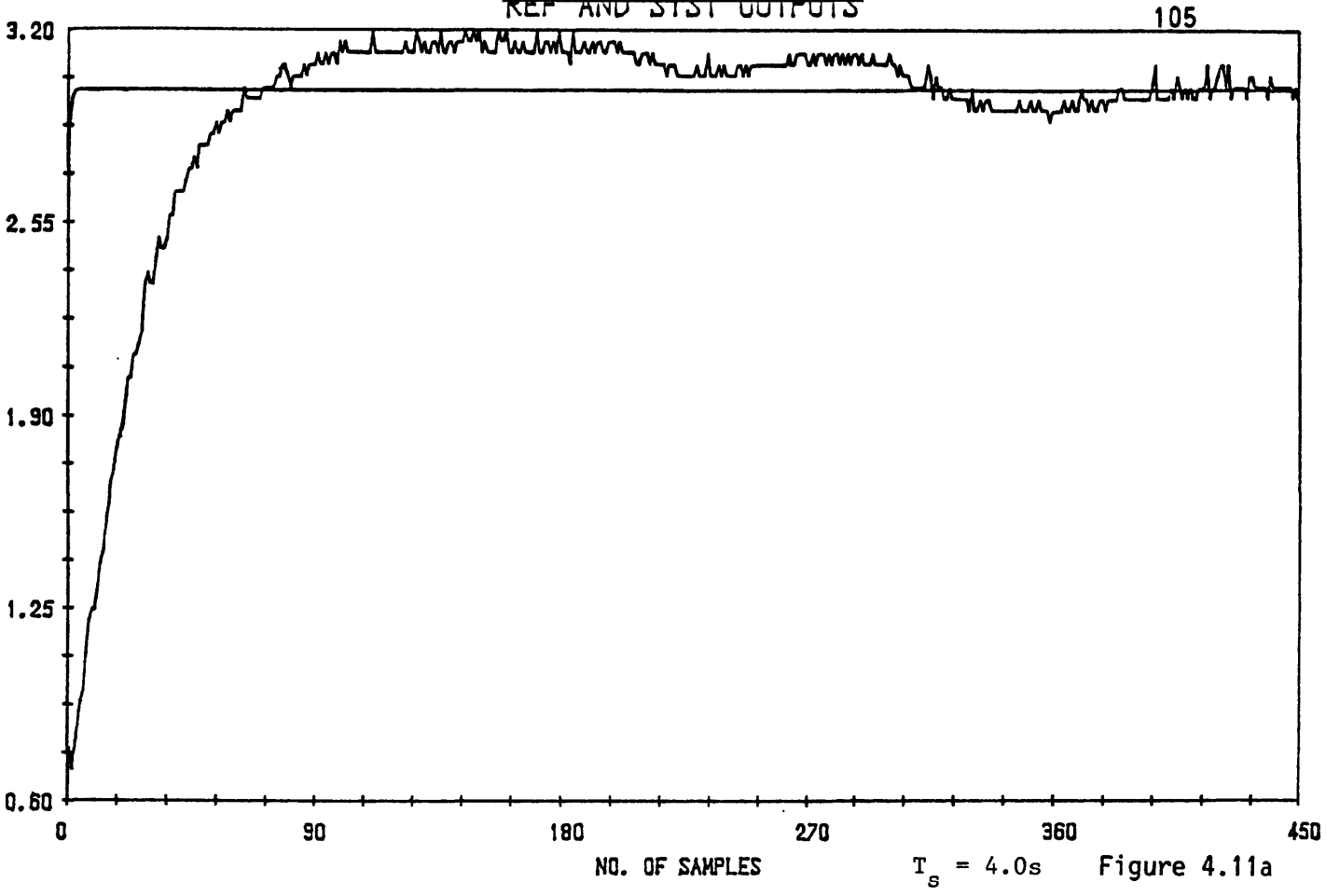
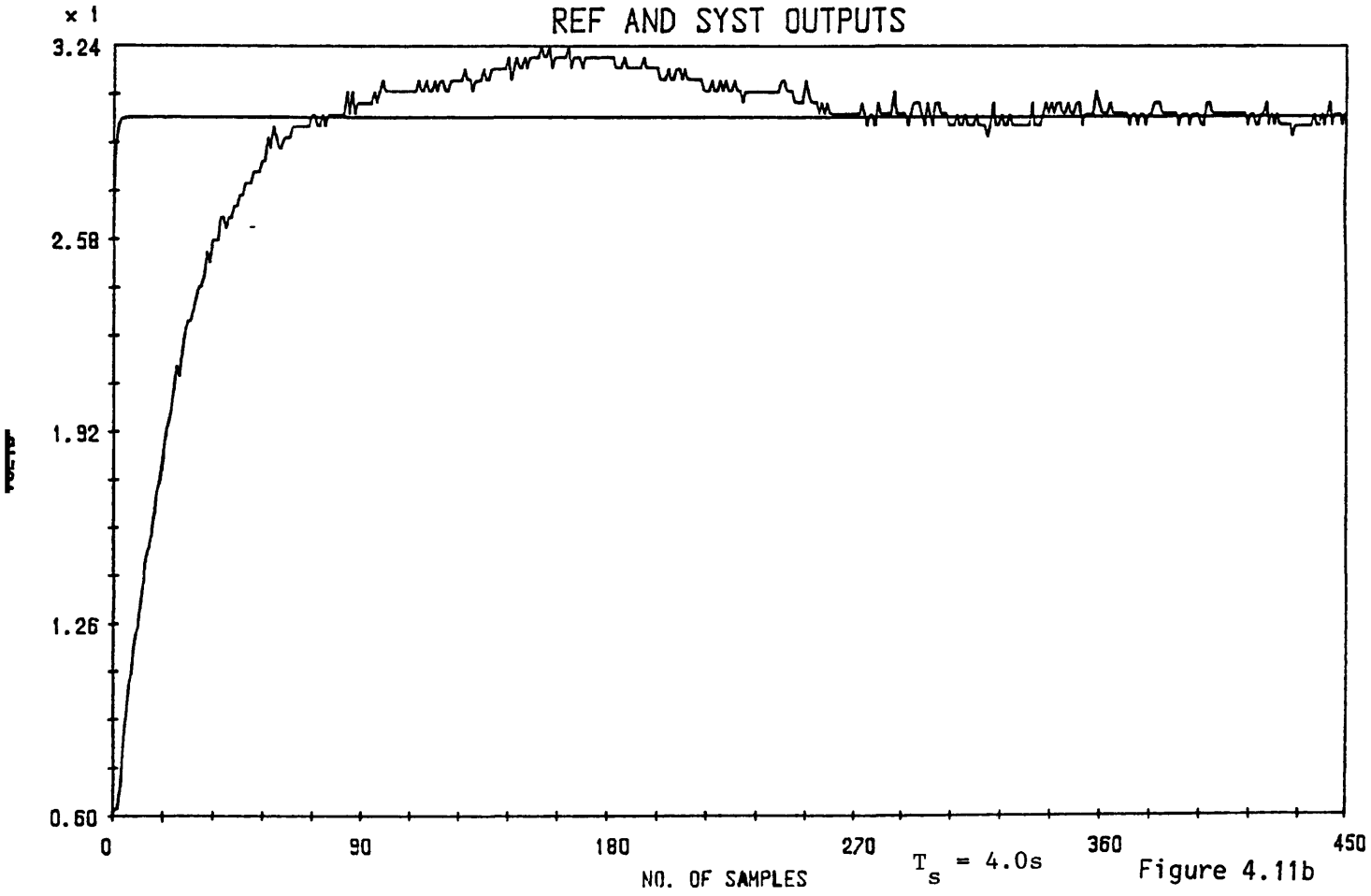


Figure 4.10 Shows the effect increasing the Goodwin adaptive gain value.

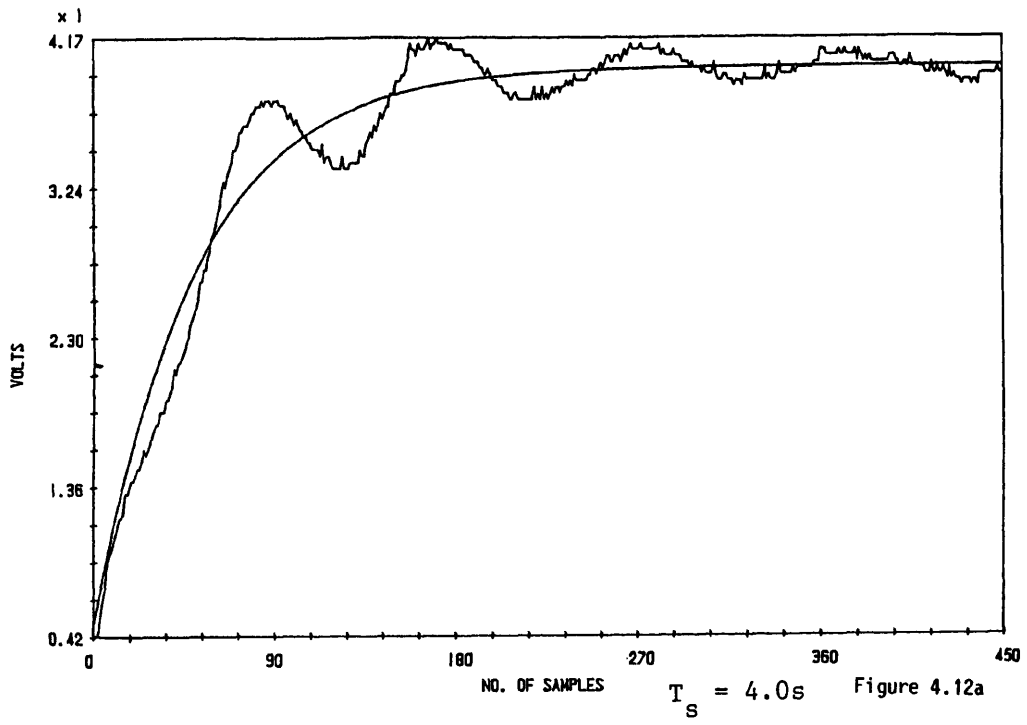


T_s = 4.0s Figure 4.11a



T_s = 4.0s Figure 4.11b

Figure 4.11 (a) Shows the Reference and System Outputs for the Ortega et al algorithm using μ = 0.75.
(b) Shows the Reference and System outputs for the Ortega et al algorithm using μ = 0.15



T_s = 4.0s Figure 4.12a

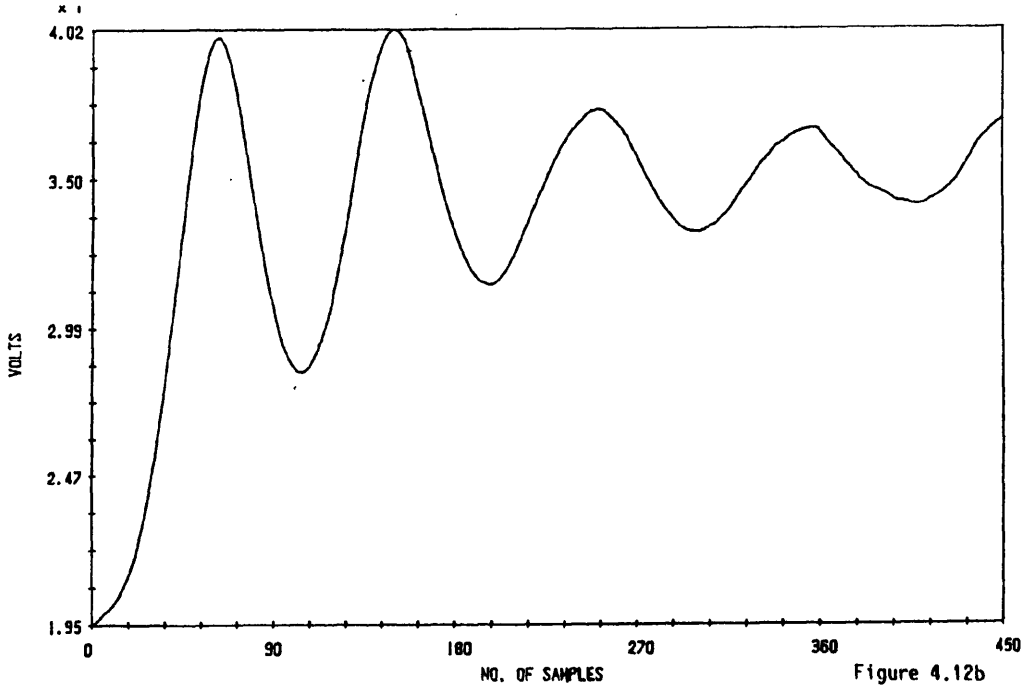
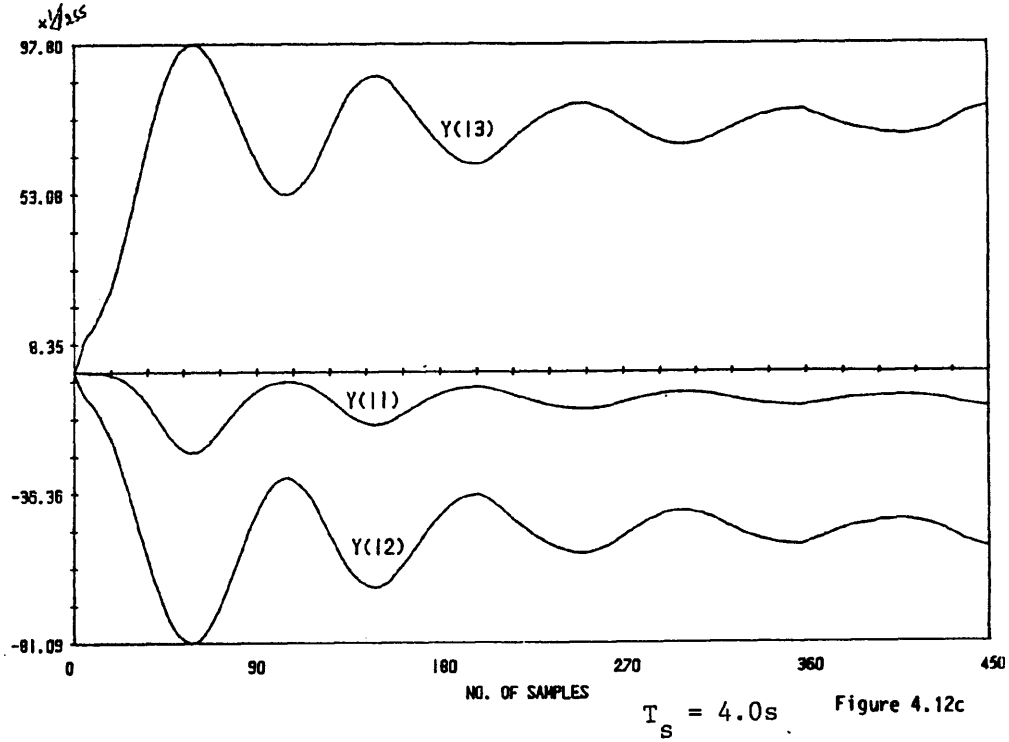


Figure 4.12b



T_s = 4.0s Figure 4.12c

Figure 4.12 (a) The Reference and System Outputs for the Goodwin et al algorithm using the slow Reference model.

(b) The Control Input u(t).

(c) The adaptive parameters θ(t).

T_s = 4.0s

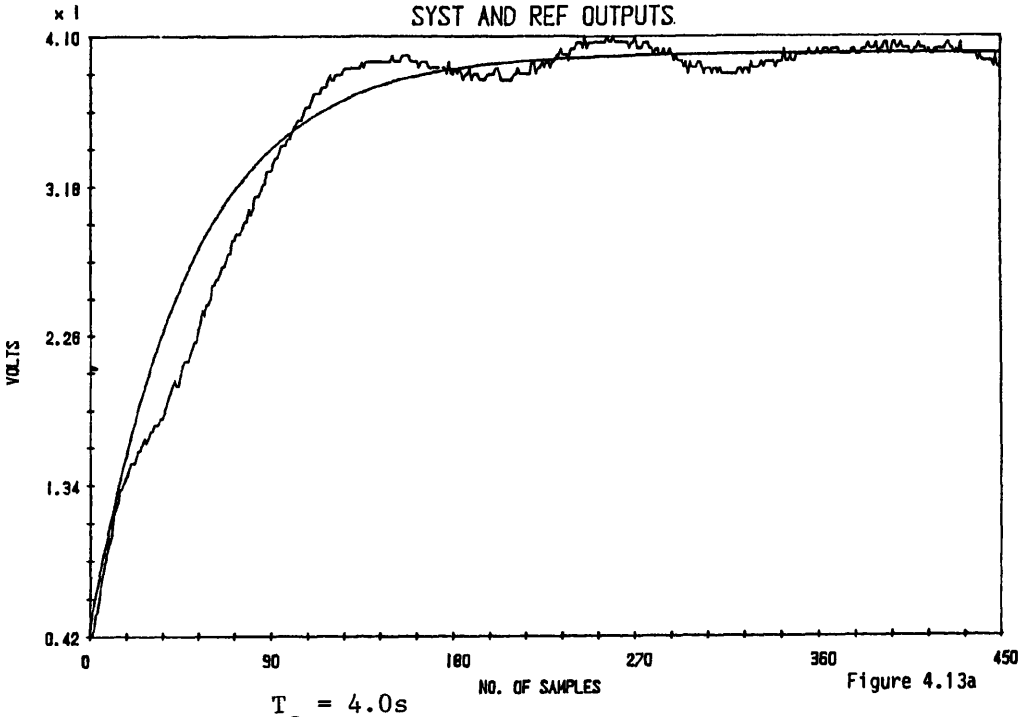


Figure 4.13a

T_s = 4.0s

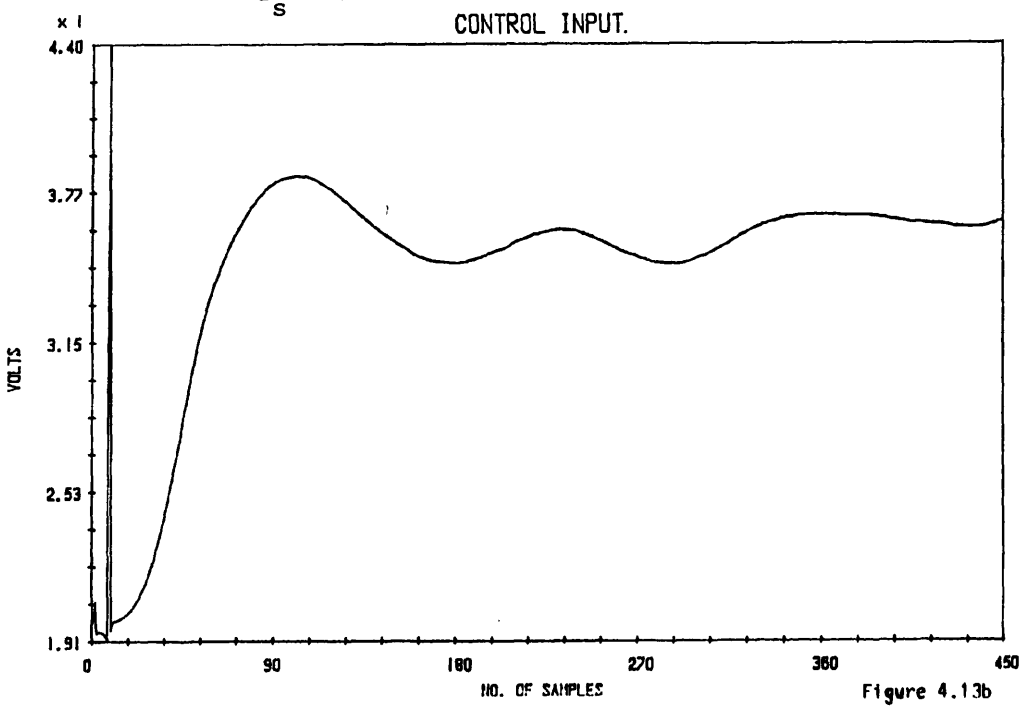


Figure 4.13b

T_s = 4.0s

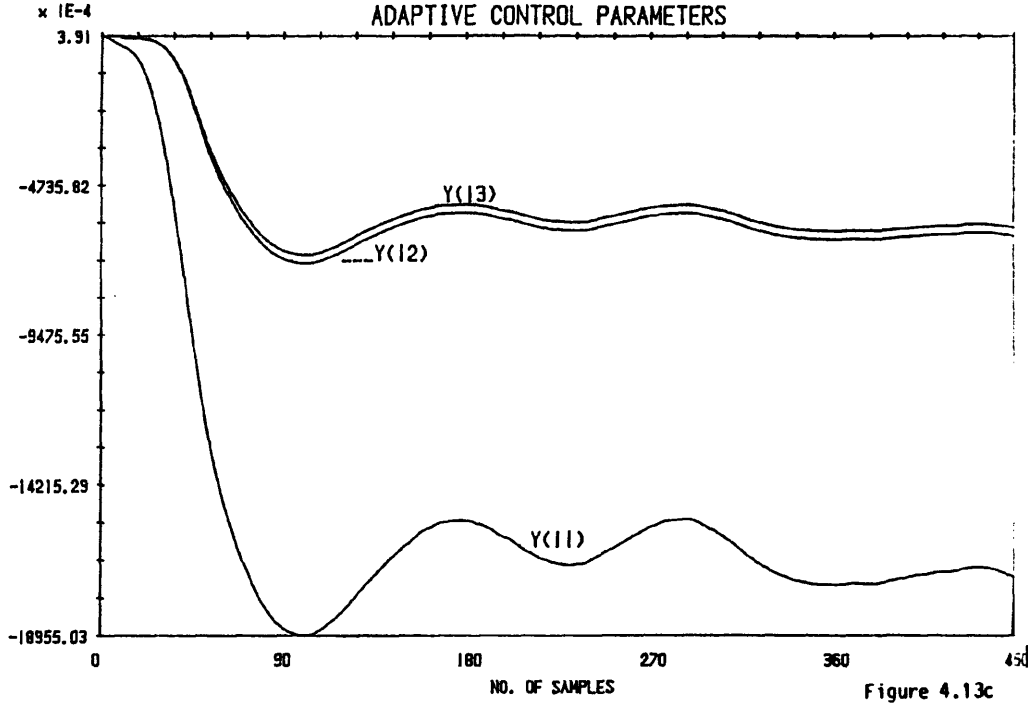


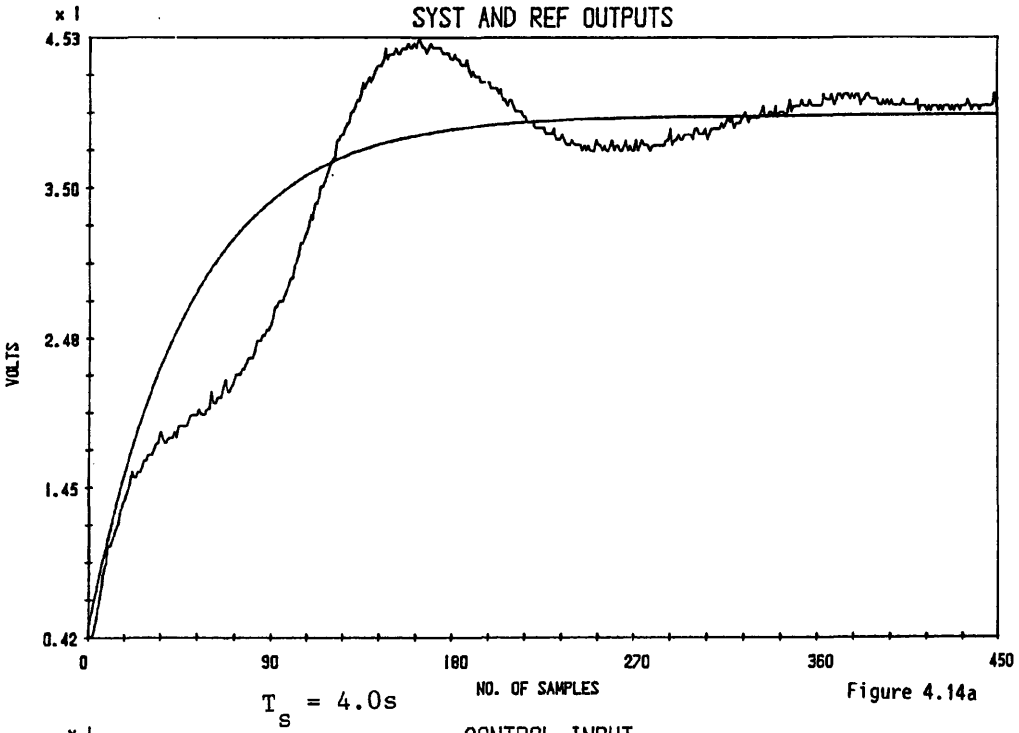
Figure 4.13c

T_s = 4.0s

Figure 4.13 (a) The Reference and System Outputs for the Ortega et al algorithm using the slow ref. model.

(b) The Control Input u(t).

(c) The adaptive parameters.



T_s = 4.0s

Figure 4.14a

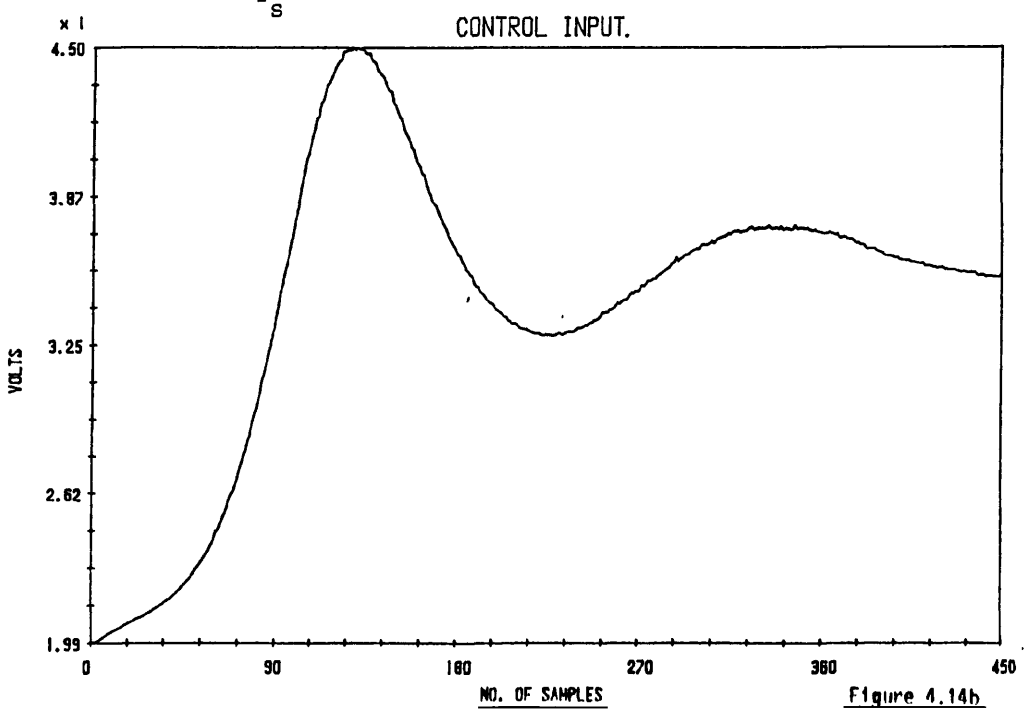
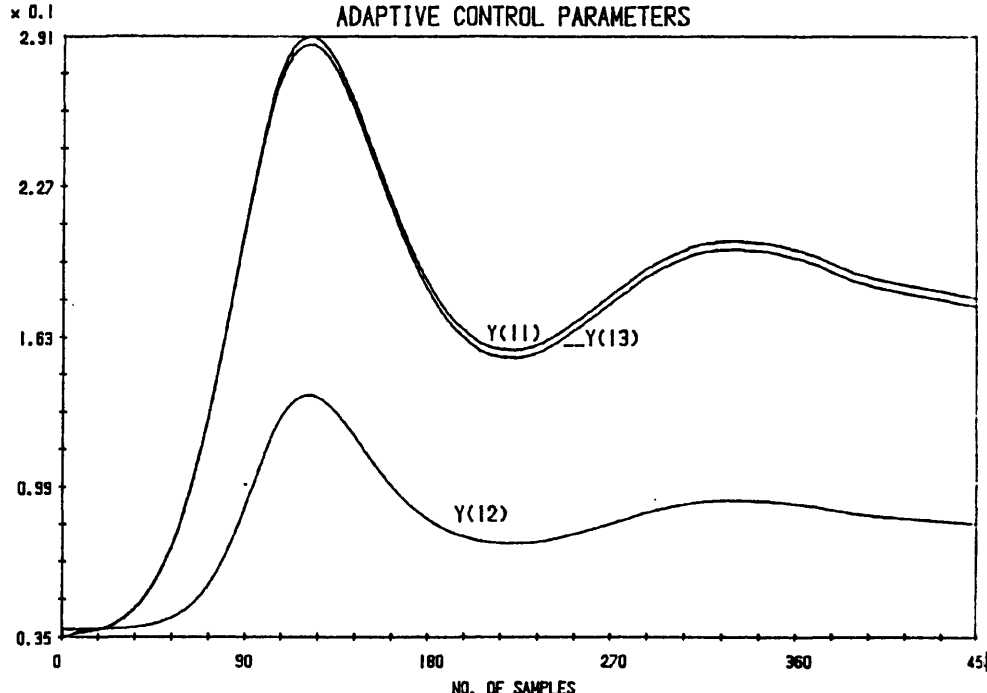


Figure 4.14b



T_s = 4.0s

Figure 4.14c

Figure 4.14 (a) The Reference and System Outputs for the Kreisselmeier and Anderson algorithm using the slow ref. model.

(b) The Control Input u(t).

(c) The adaptive parameters.

T_s = 4.0s

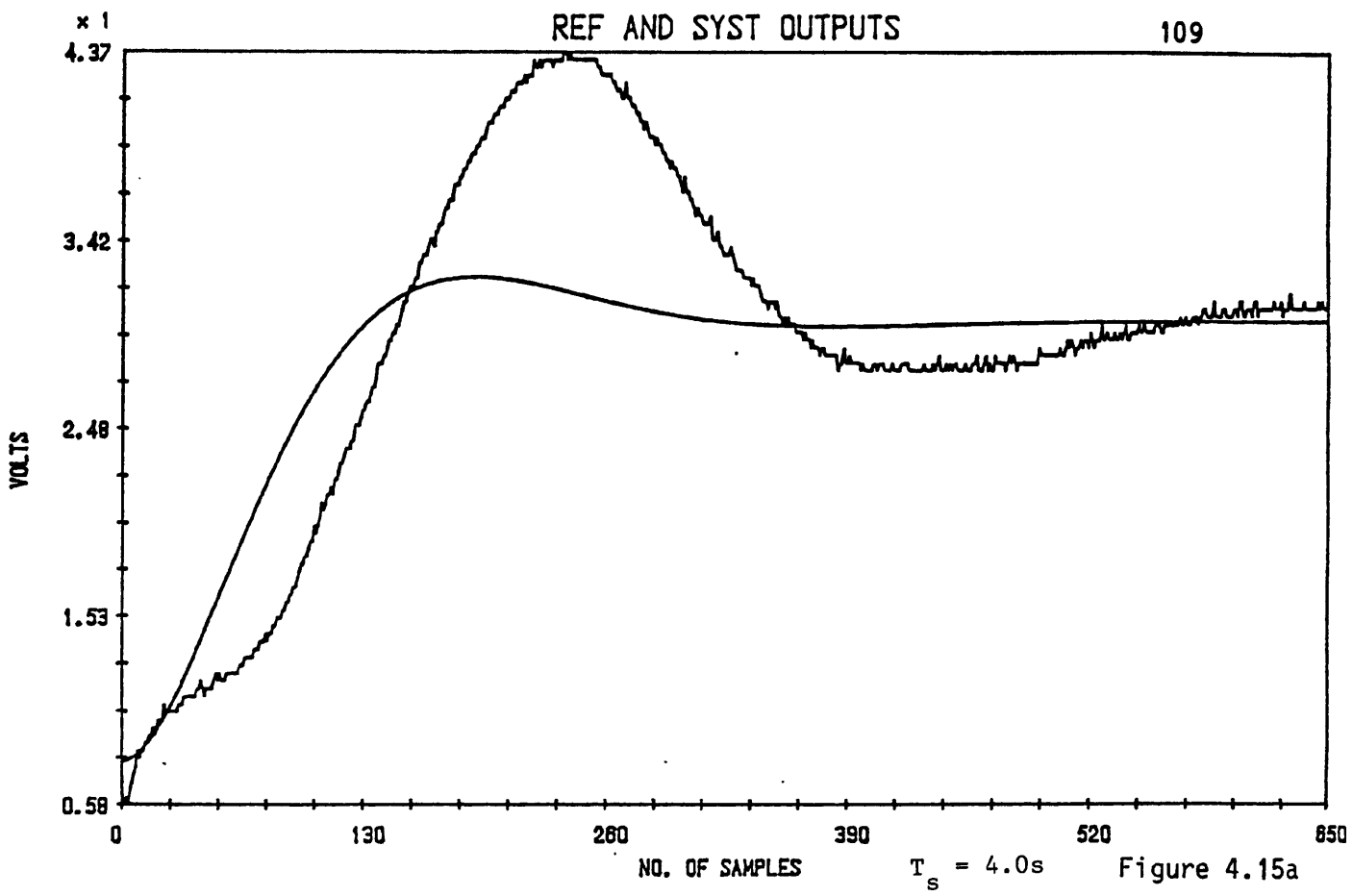


Figure 4.15a

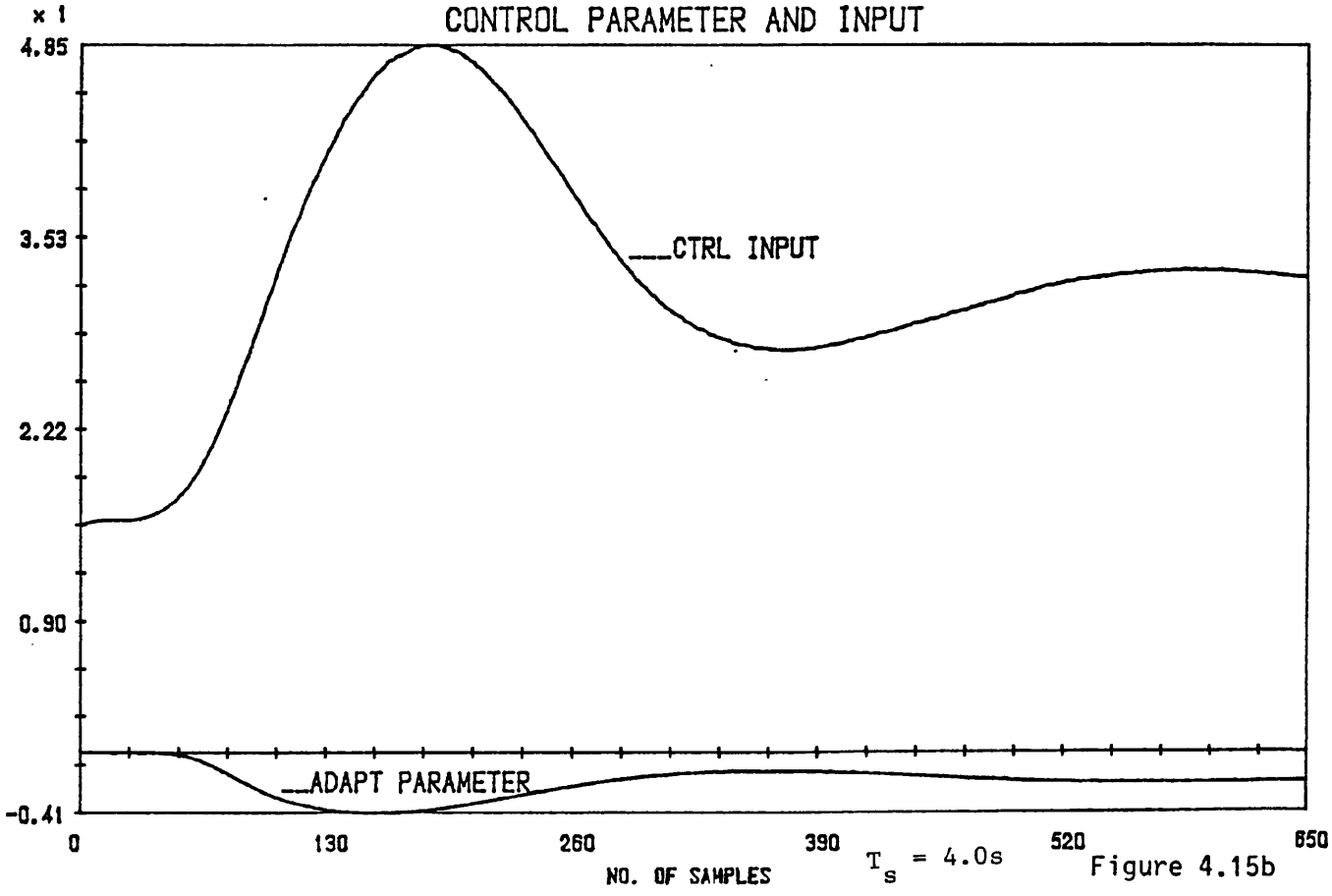


Figure 4.15b

Figure 4.15 (a) The Reference and System Outputs for the Goodwin et al algorithm for a second-order ref. model.
(b) The Control Input u(t) and the adaptive parameters θ(t).

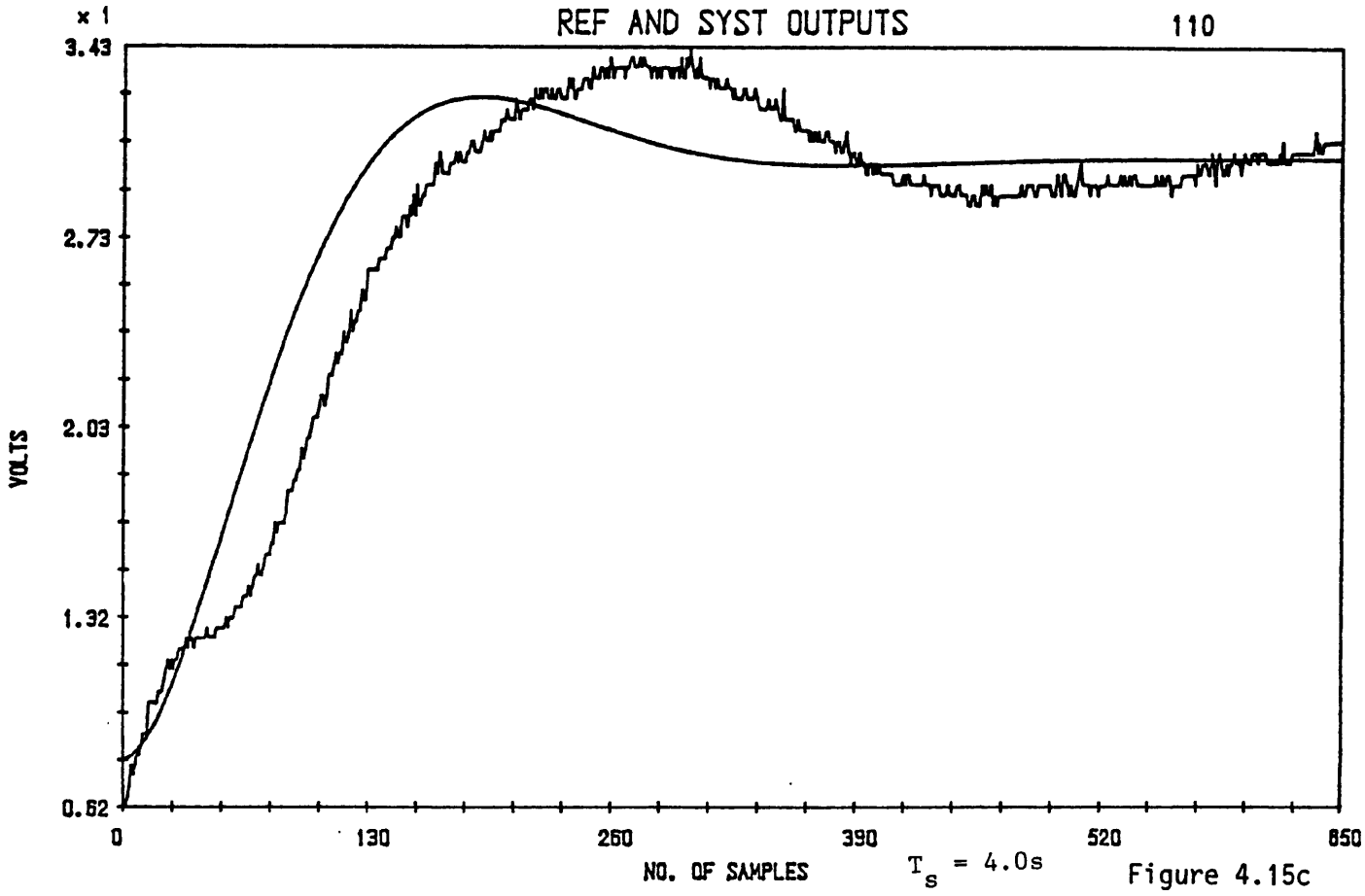


Figure 4.15c

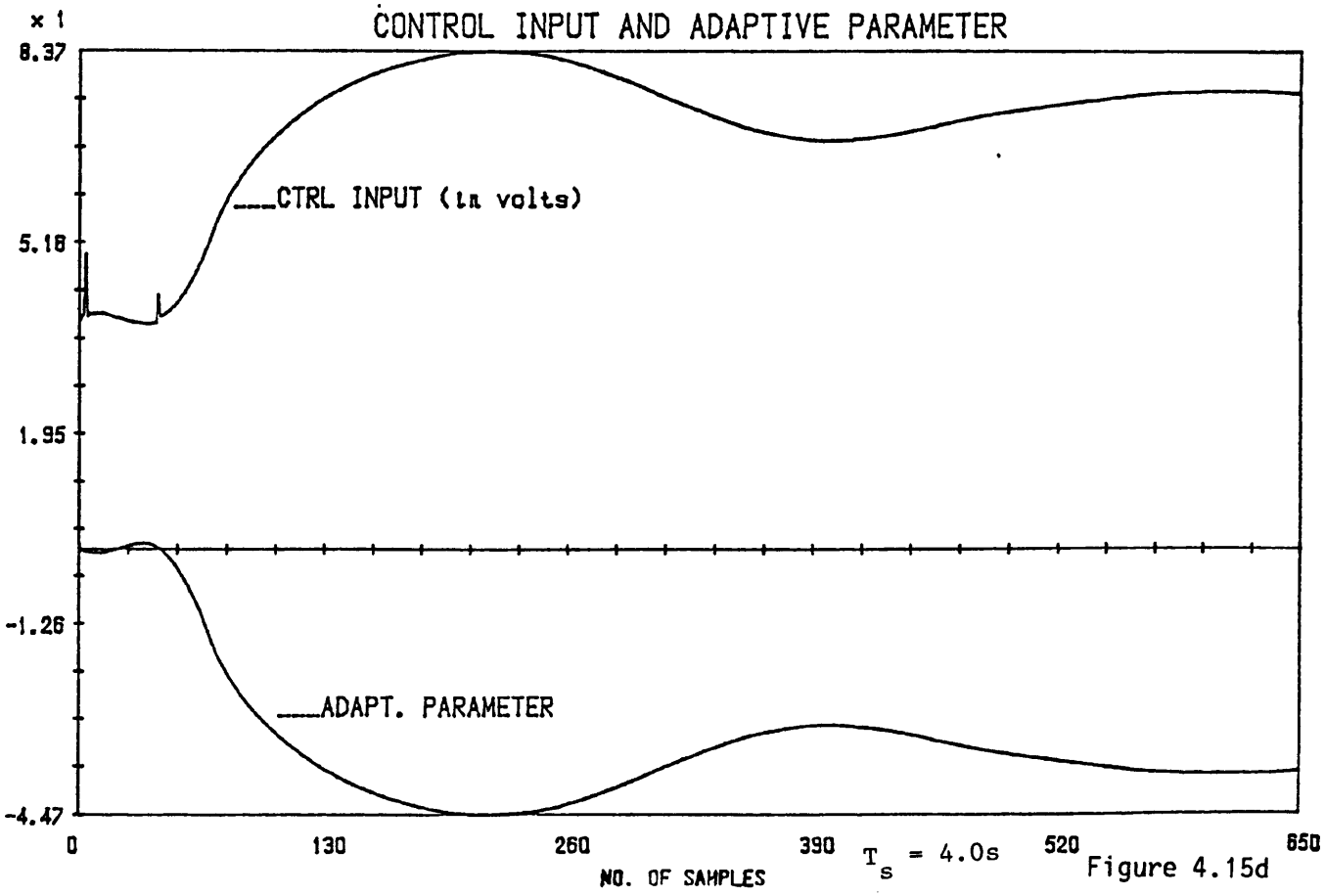
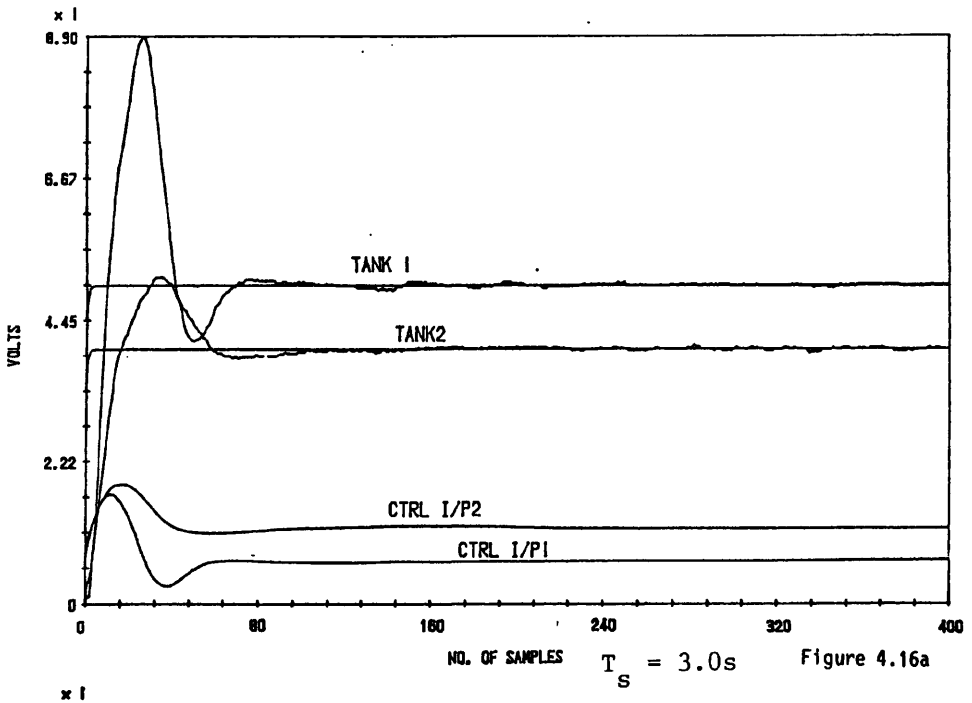


Figure 4.15d

Figure 4.15 (c) The Reference and System Outputs for the Ortega et al algorithm for a second-order ref. model.
(d) The Control Input u(t) and the adaptive parameters θ(t).



T_s = 3.0s

Figure 4.16a

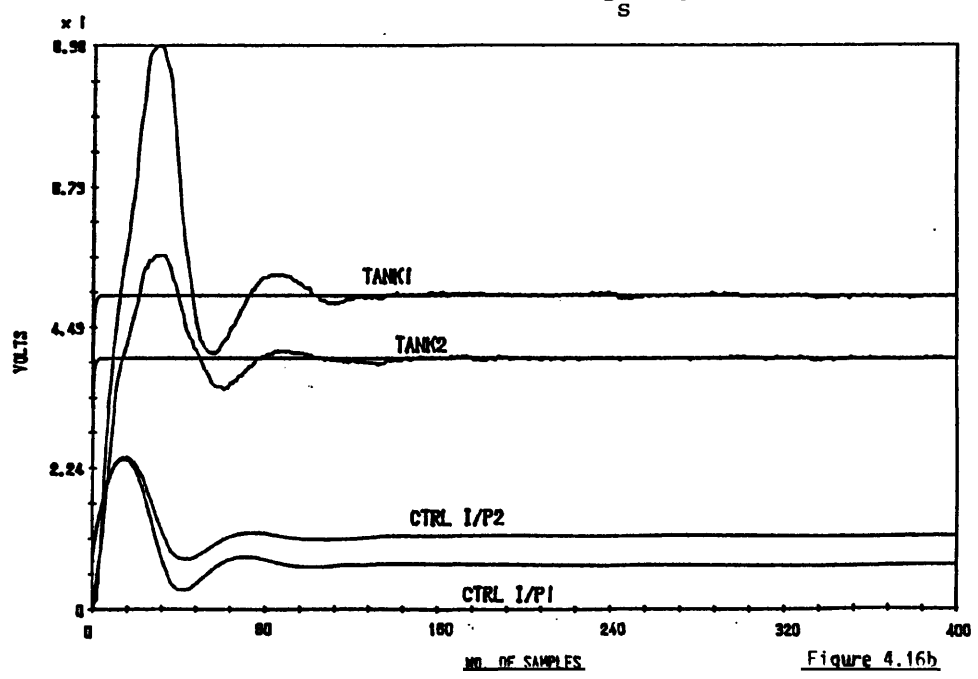


Figure 4.16b

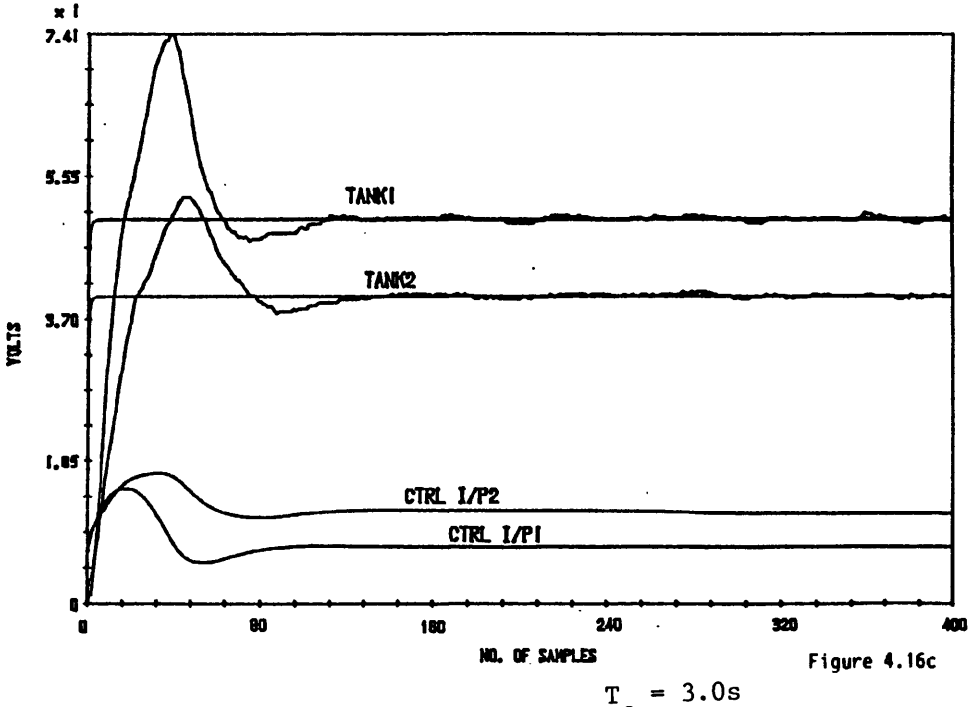
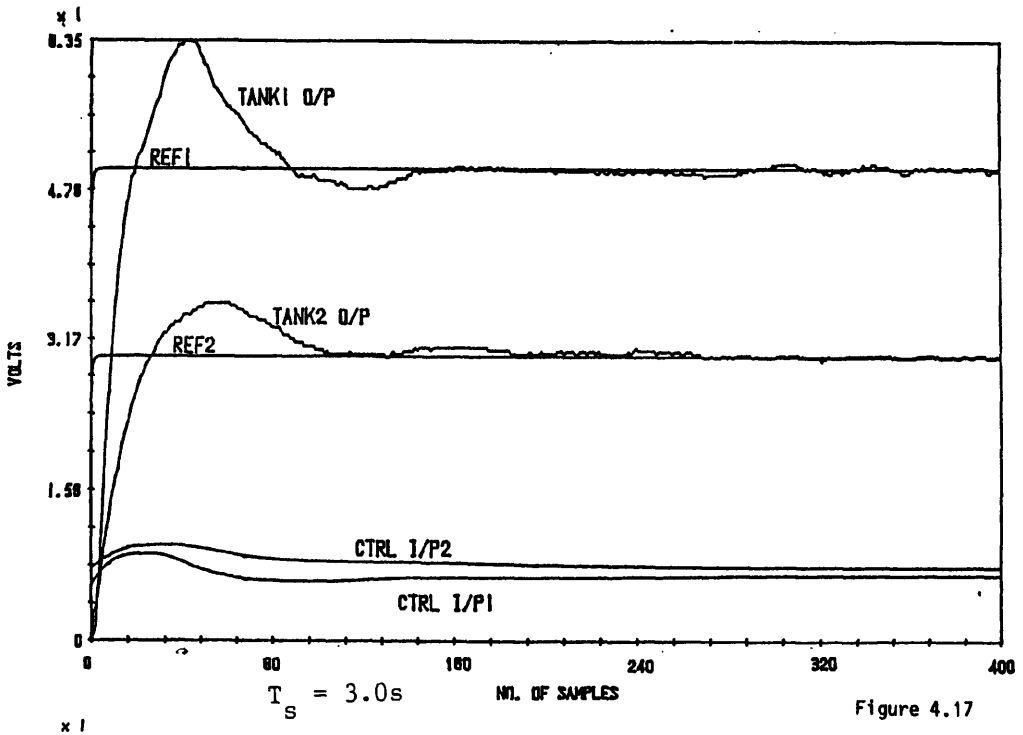


Figure 4.16c

T_s = 3.0s

Figure 4.16 Showing the 2 tank outputs and their ref. outputs plus control inputs for the Goodwin et al multivariable algorithm for different configurations of the P-matrix.

T_s = 3.0s



T_s = 3.0s

Figure 4.17

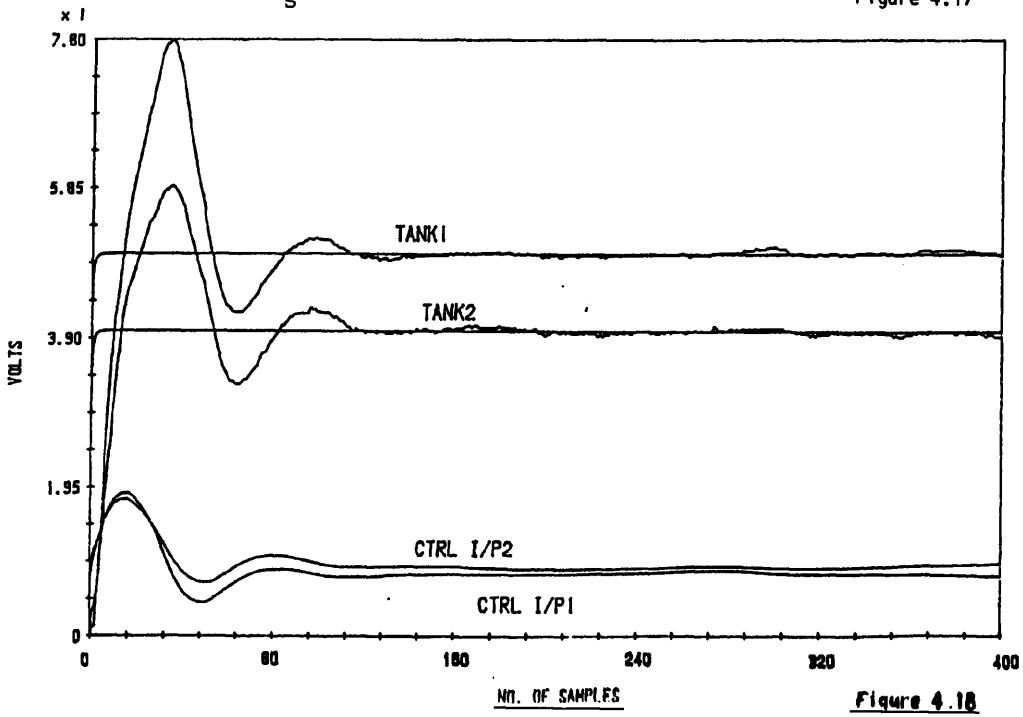


Figure 4.18

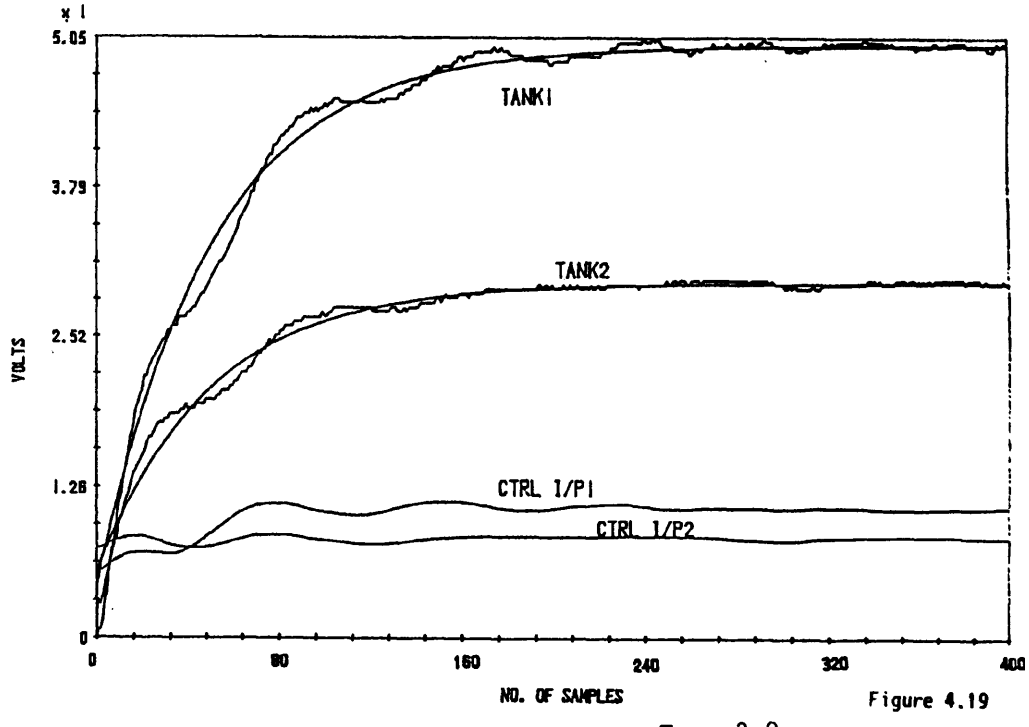


Figure 4.19

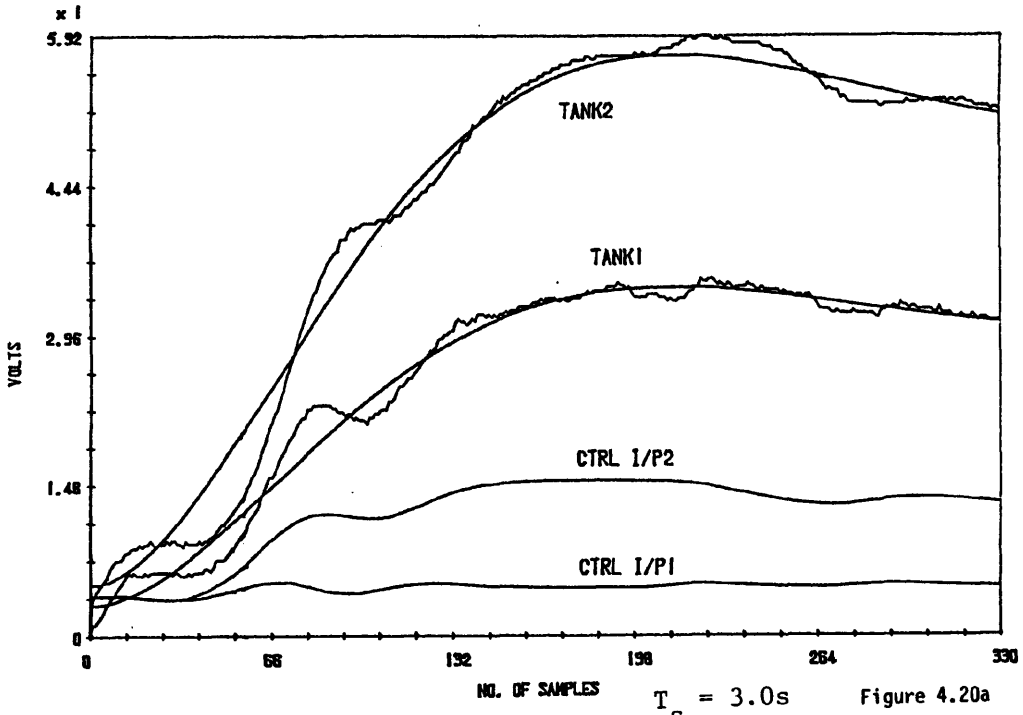
T_s = 3.0s

Figure 4.17 The multivariable Reference and System Outputs plus control inputs for the Goodwin et al algorithm using a diagonal P-matrix, $P = \begin{bmatrix} 0.04 & 0 \\ 0 & 0.04 \end{bmatrix}$

Figure 4.18 Shows the effects of increasing the orifice size (or coupling between the tanks) for the multivariable tanks.

Figure 4.19 The multivariable reference and system outputs plus control inputs for the tanks using slow first-order ref. models for the Goodwin et al MIMO algorithm.

T_s = 3.0s



T_s = 3.0s Figure 4.20a

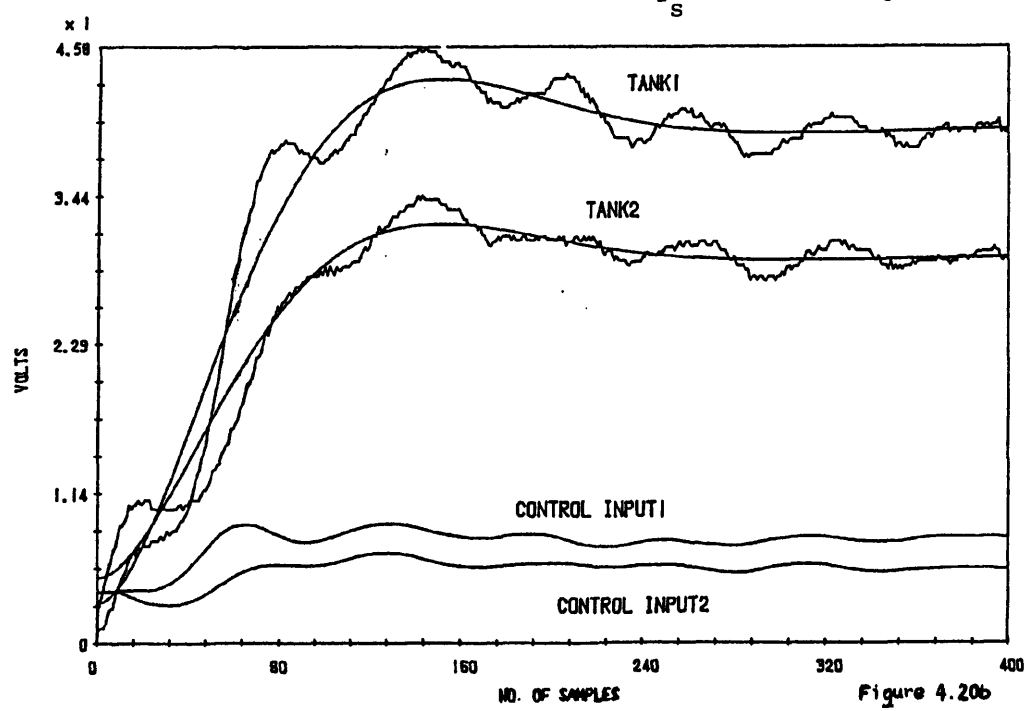


Figure 4.20b

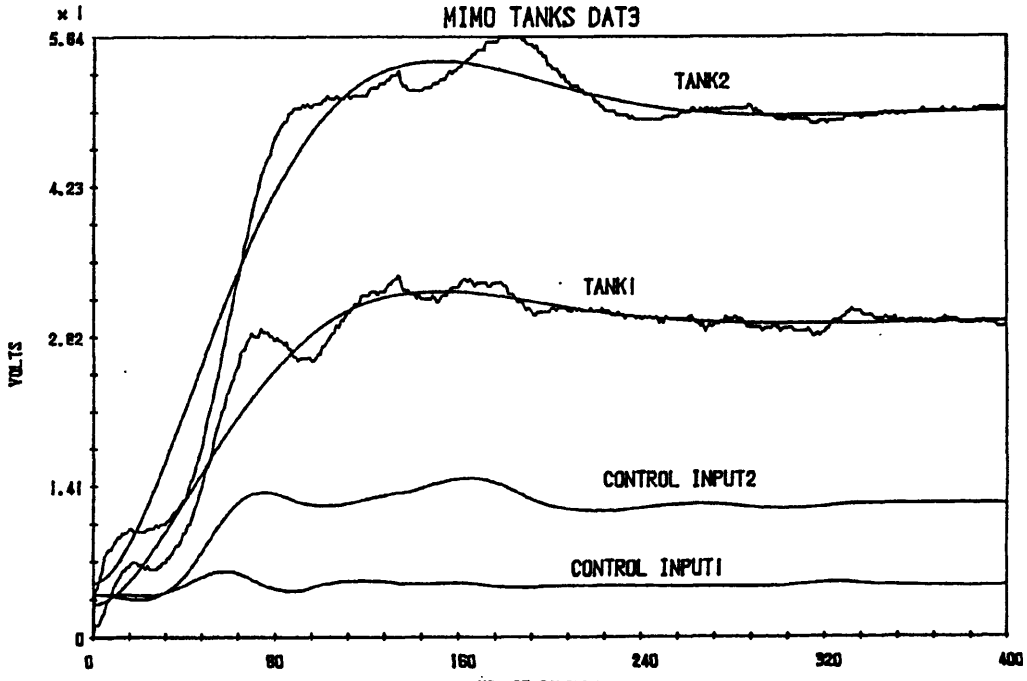
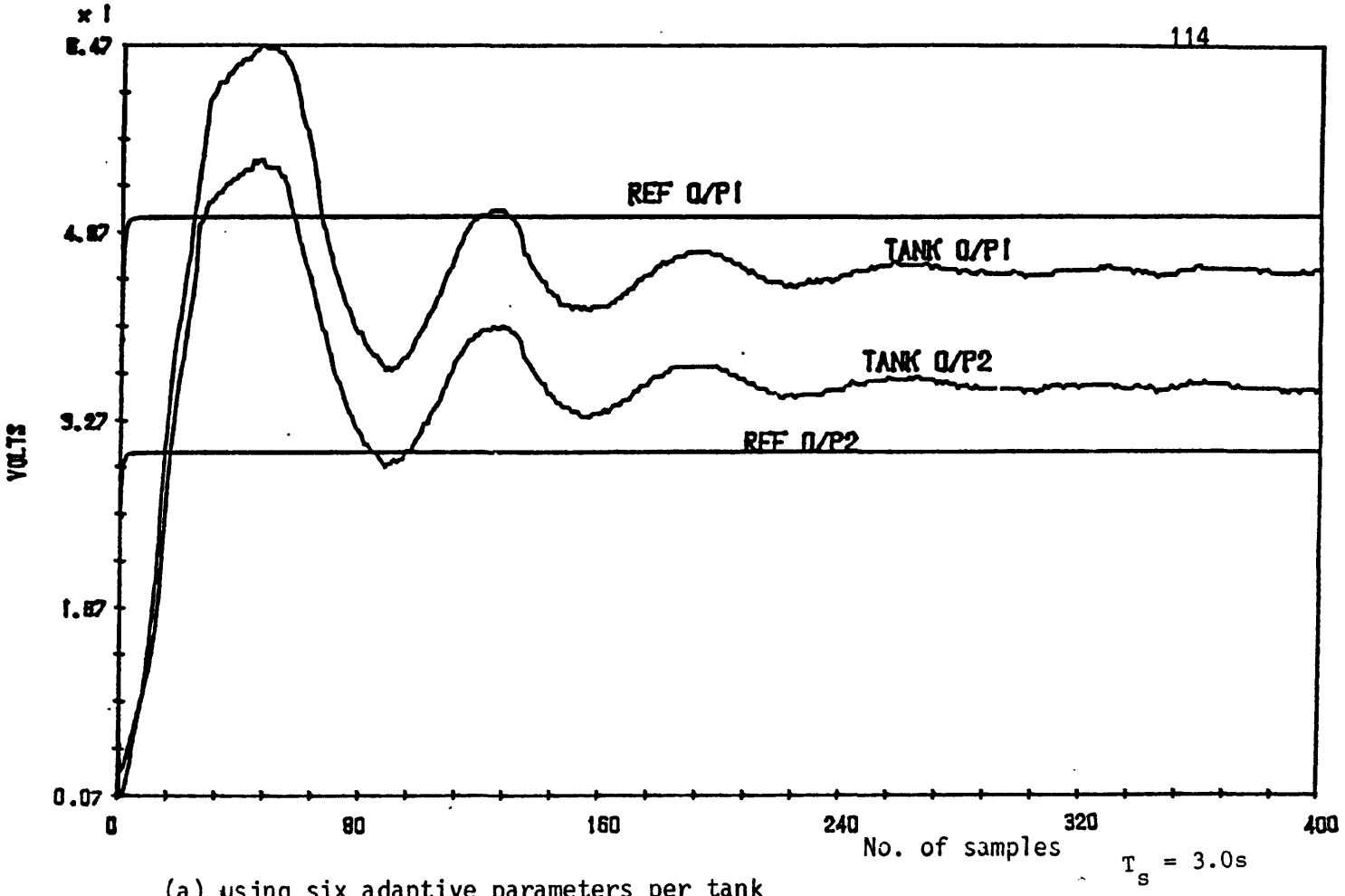


Figure 4.20c

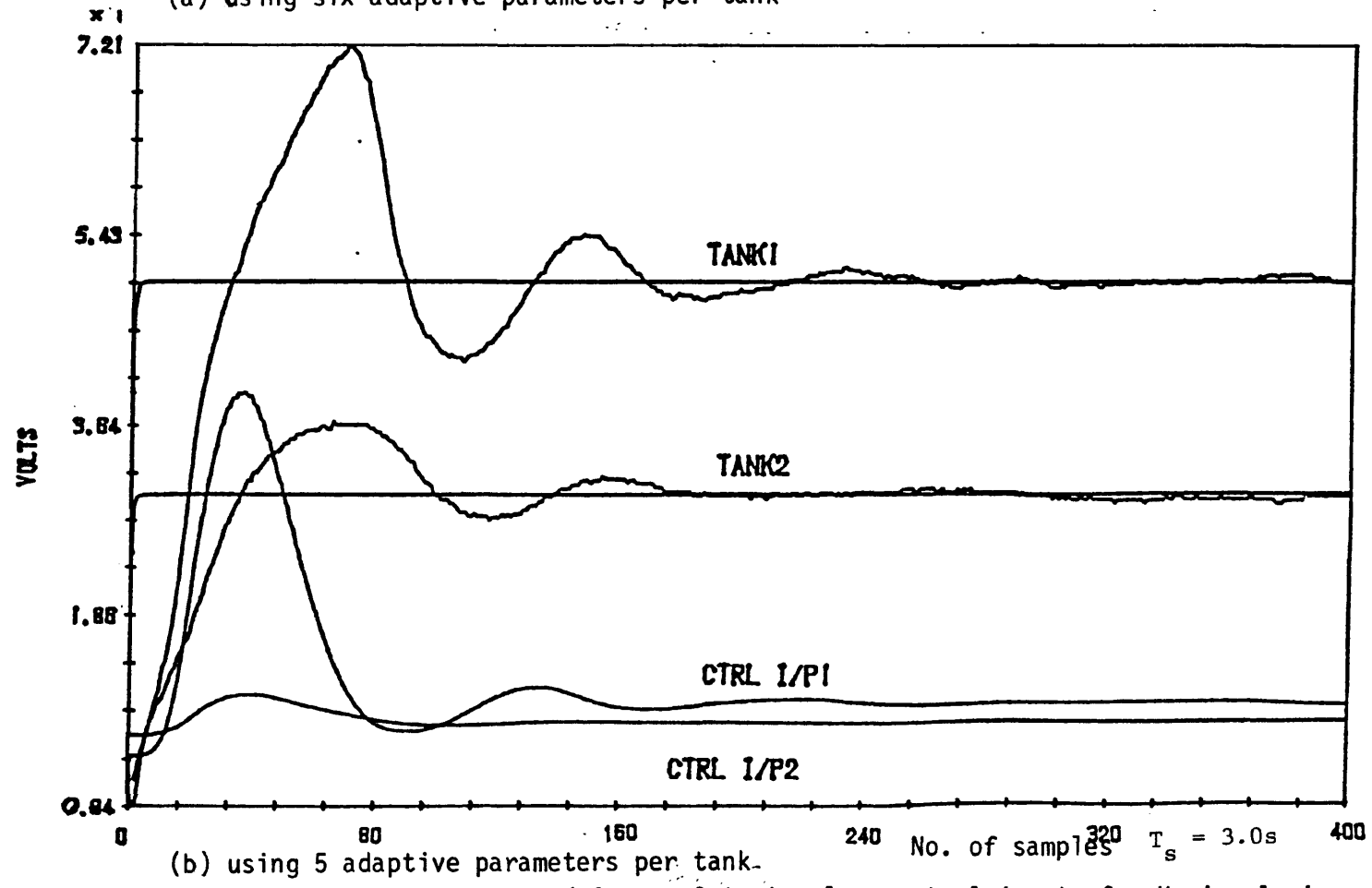
T_s = 3.0s

Figure 4.20 The results of using various second-order reference models on the multivariable tanks for the MIMO Goodwin et al algorithm.

T_s = 3.0s



(a) using six adaptive parameters per tank



(b) using 5 adaptive parameters per tank.

Figure 4.21 The Reference and System Outputs plus control inputs for Kreisselmeier and Anderson's multivariable algorithm.

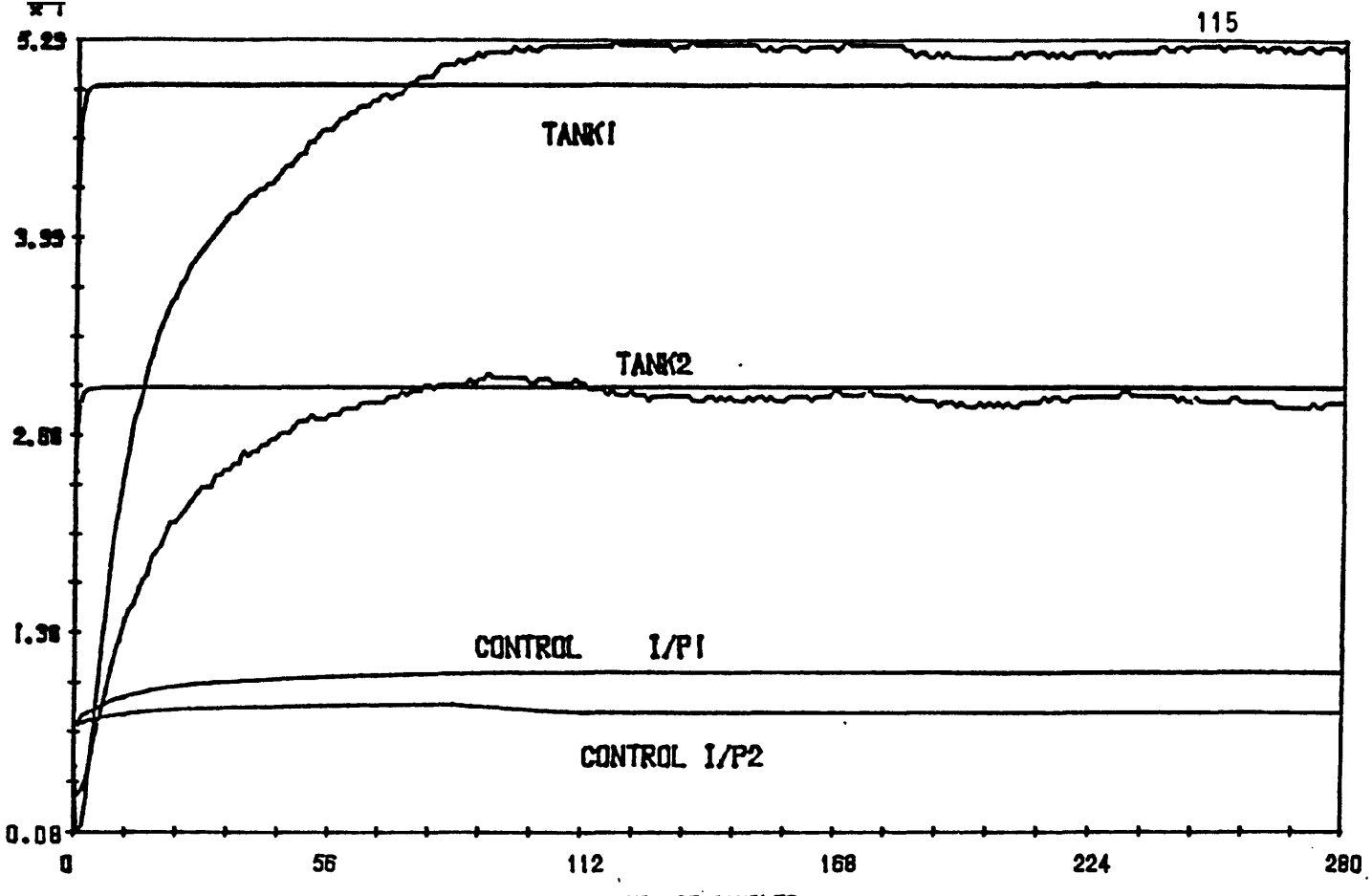


Figure 4.22a

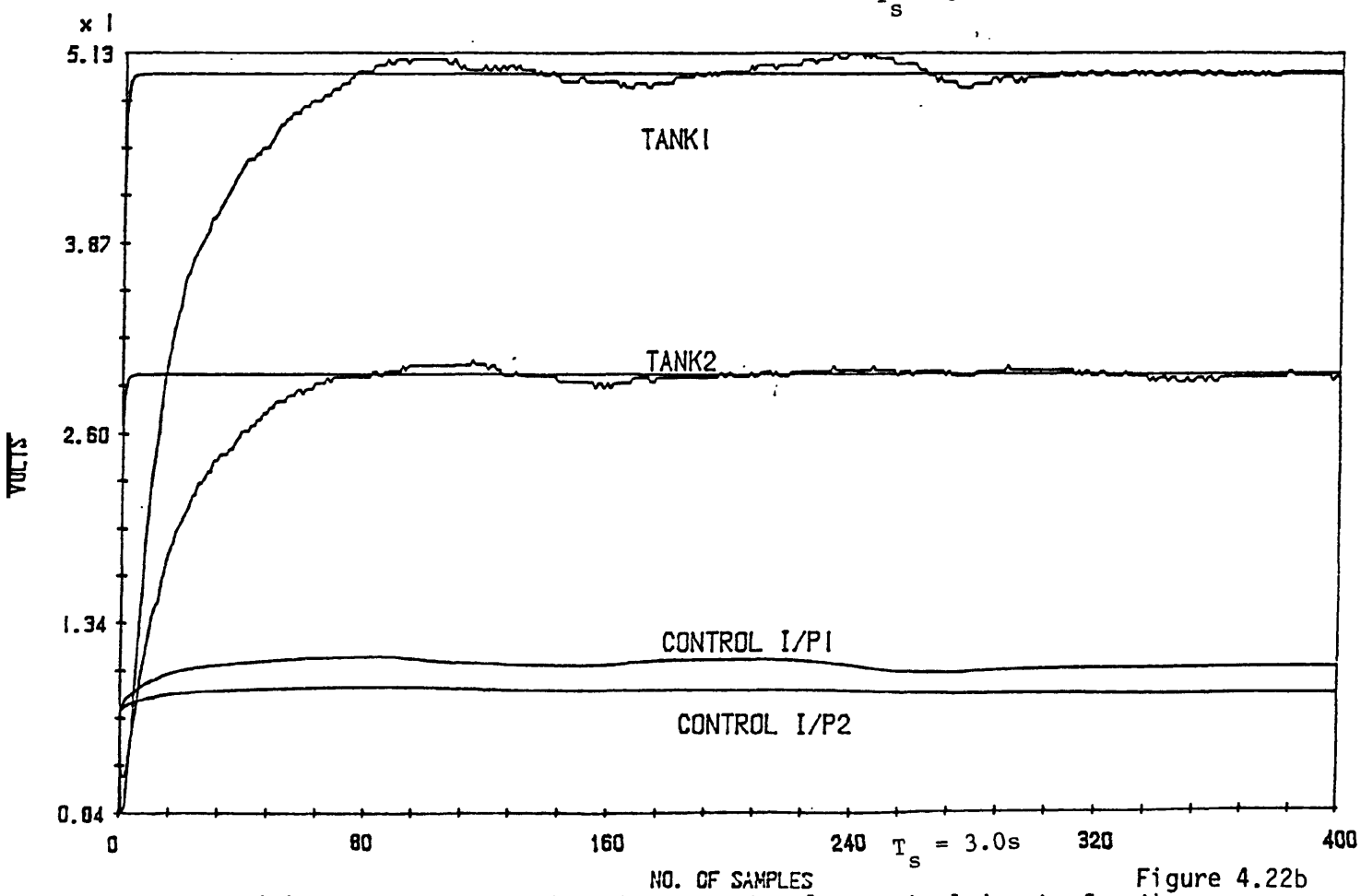


Figure 4.22b

Fig. 4.22 (a) The reference and system outputs plus control inputs for the Kreisselmeier & Anderson MIMO algorithm for (a) large dead zones d_0 = 0.5 for Tank 1, d_0 = 0.4 for Tank 2, and
(b) small dead zones d_0 = 0.05 for Tank 1, d_0 = 0.05 for Tank 2.

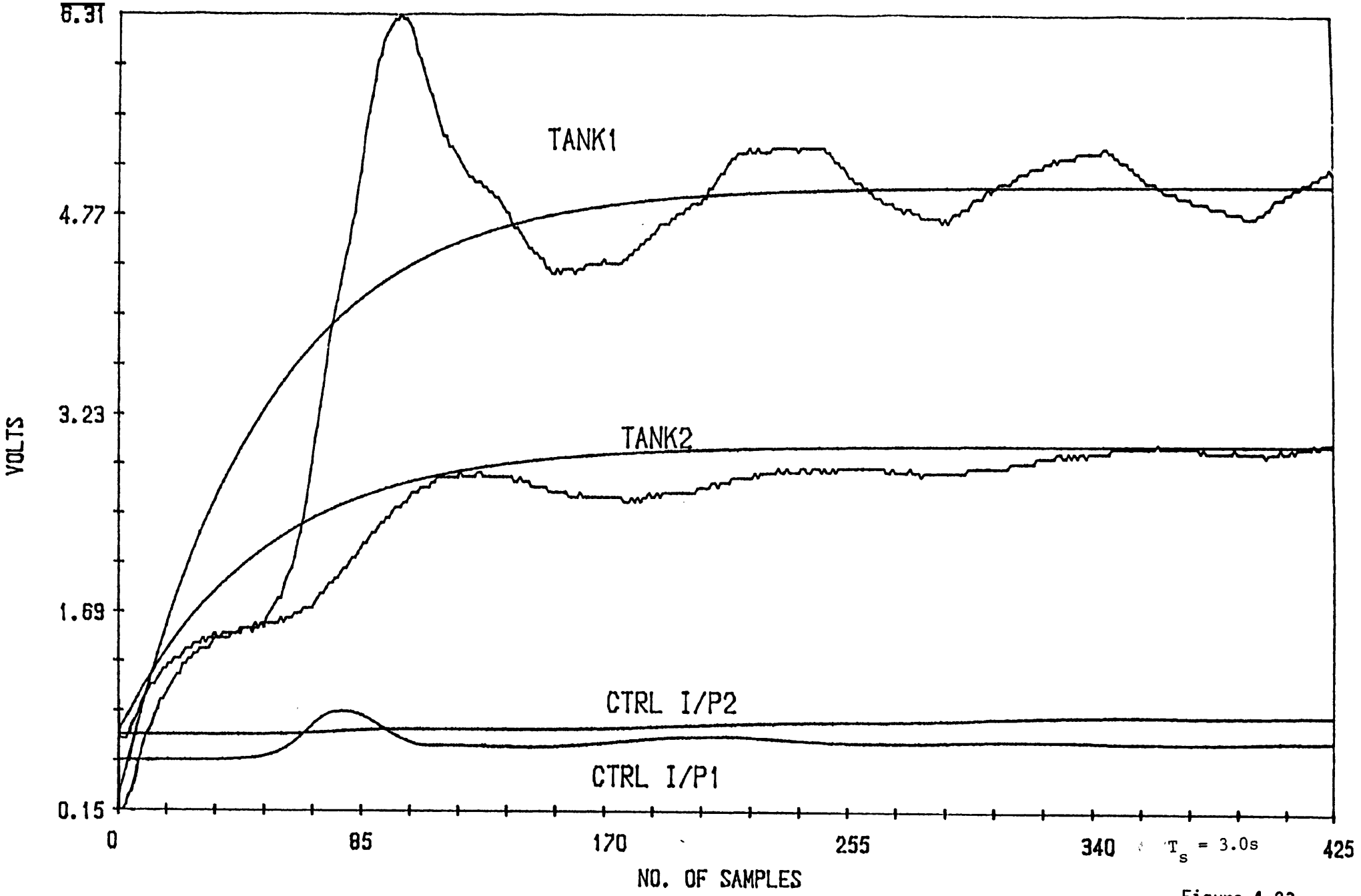


Figure 4.23

Figure 4.23 The Reference and System Outputs plus control inputs for the Kreisselmeier & Anderson MIMO algorithm using slow first-order reference models for the multivariable tanks.

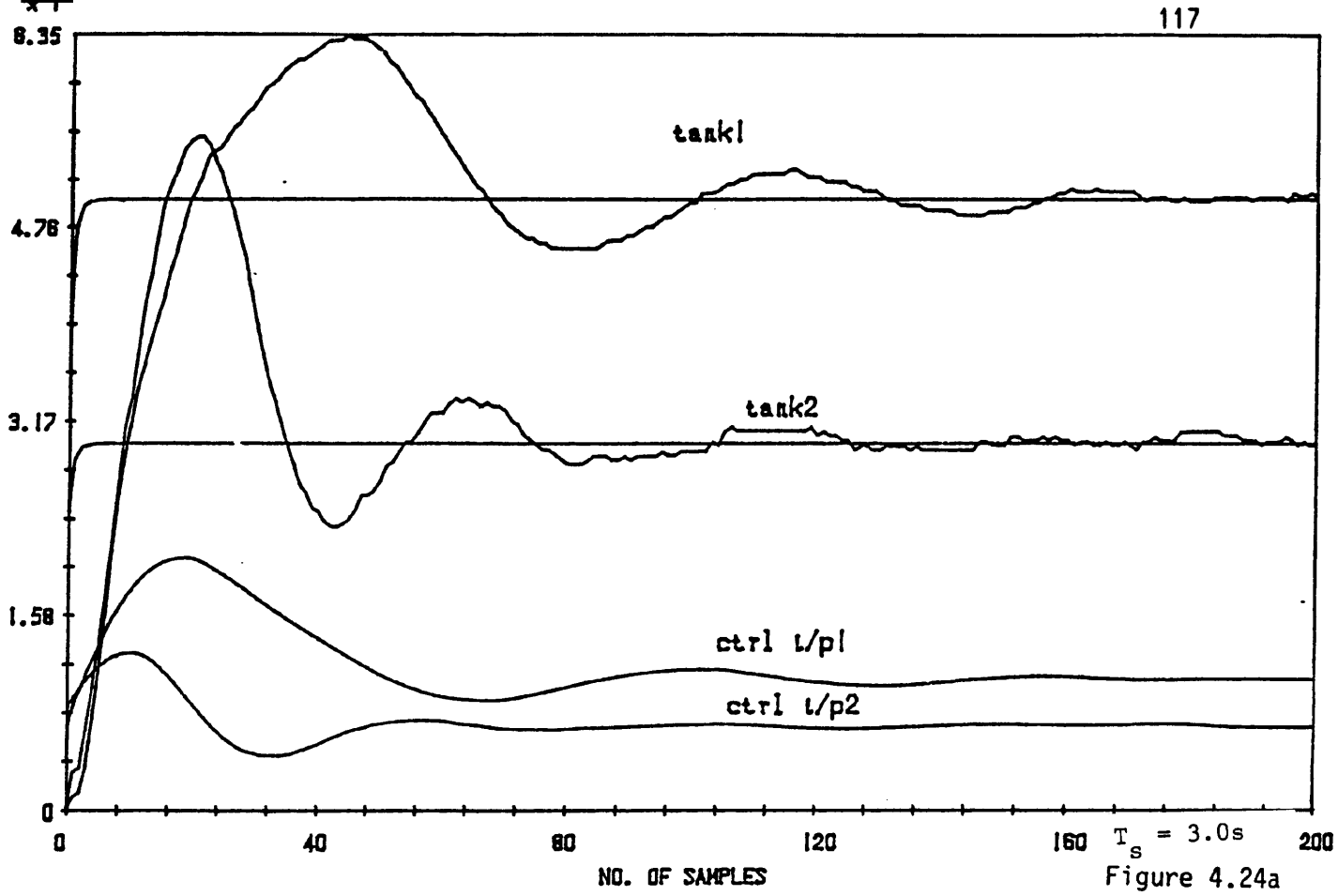


Figure 4.24a

(a) without parameter bounds

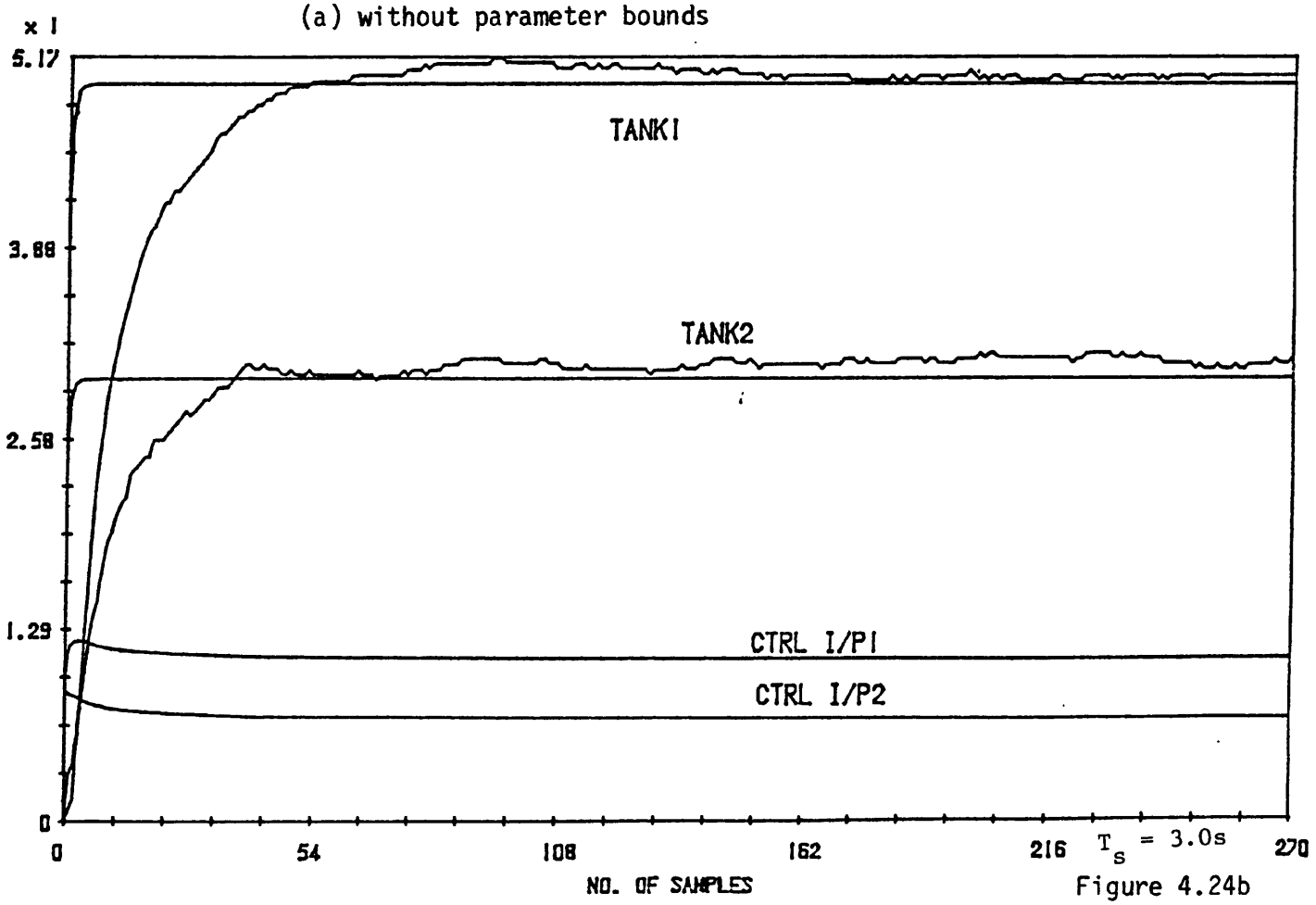


Figure 4.24b

(b) with adaptive parameter bounds

Fig. 4.24 The reference, system outputs plus control inputs for Goodwin's modified MIMO algorithm.

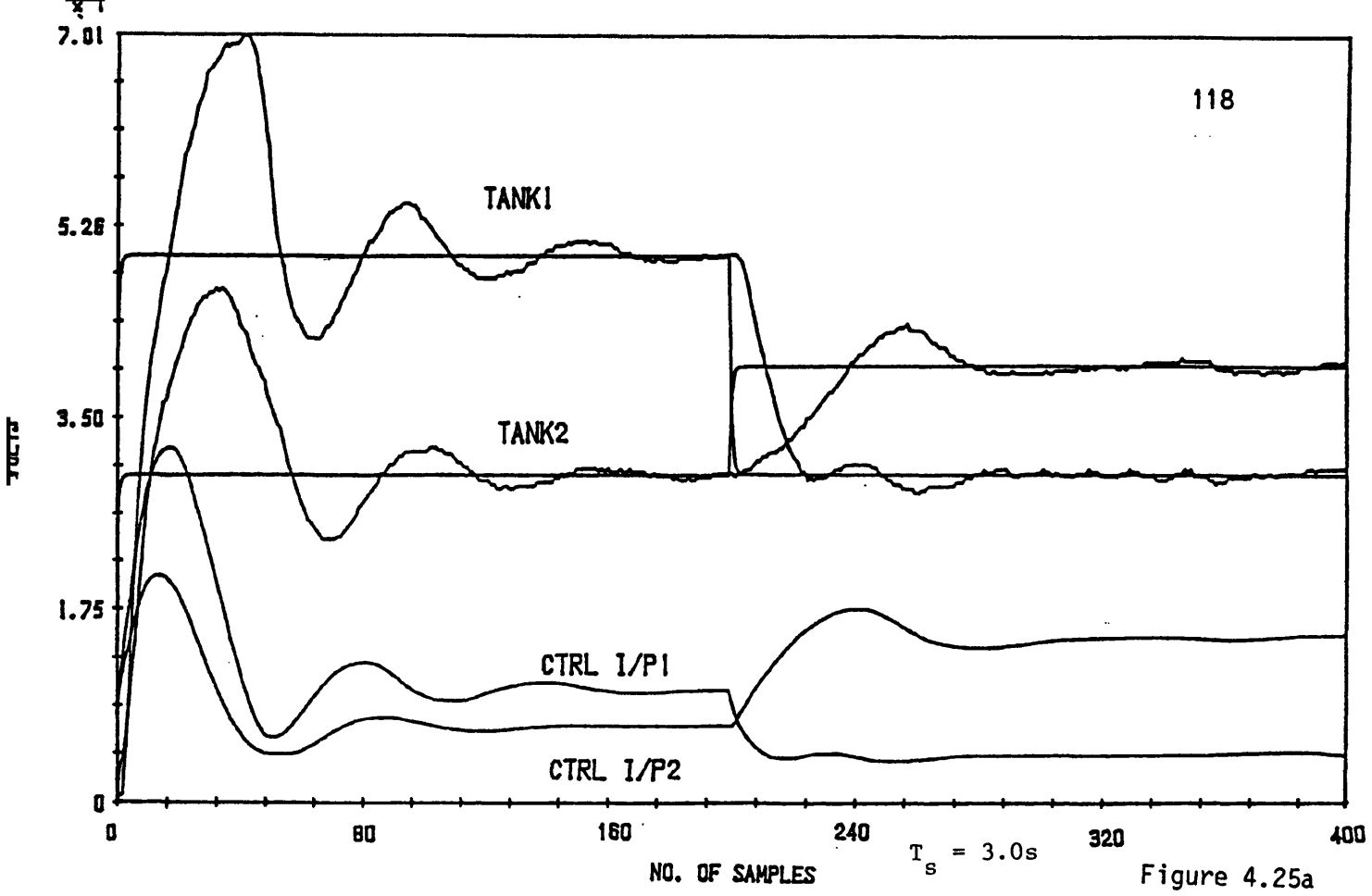


Figure 4.25a

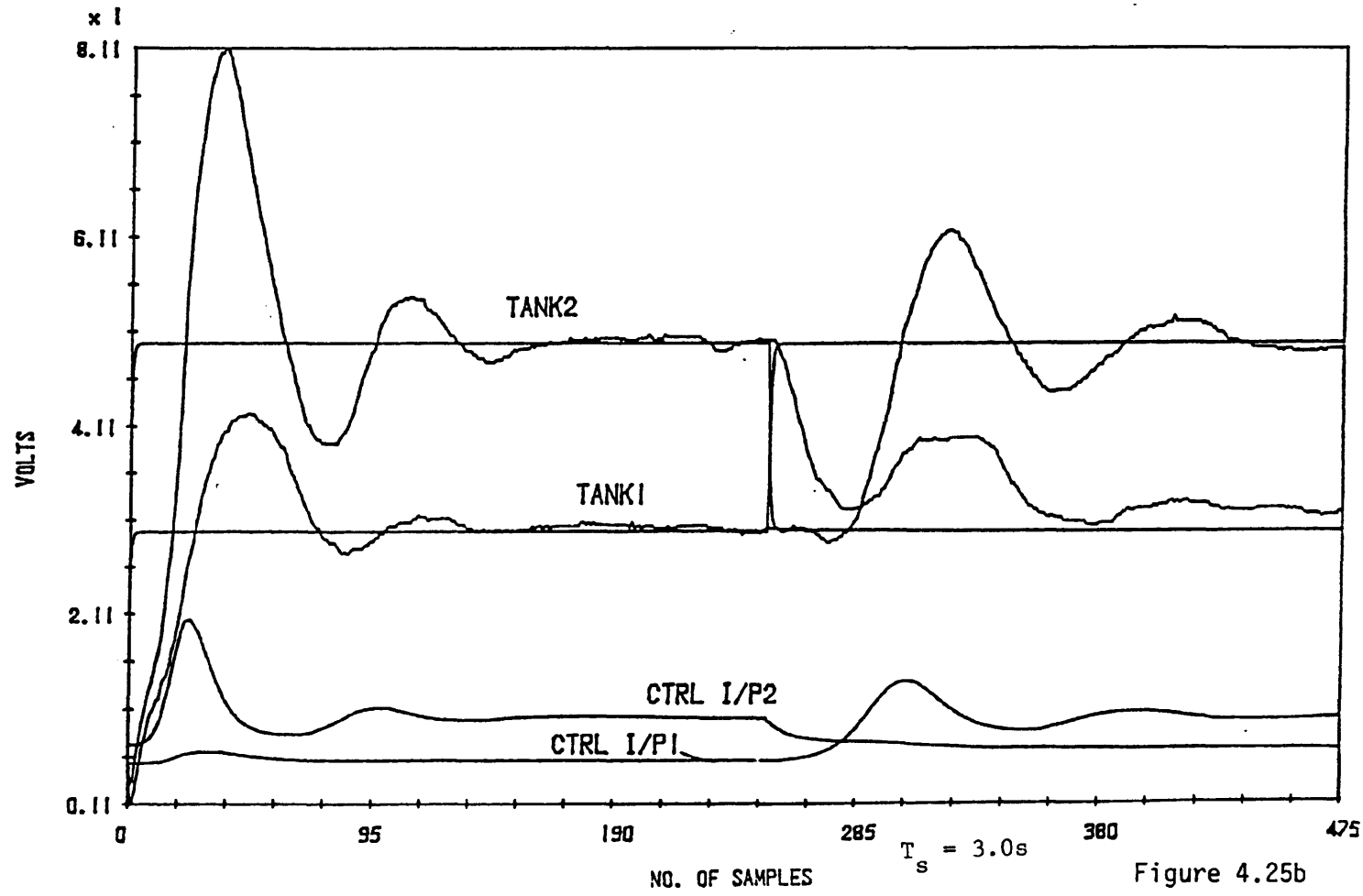
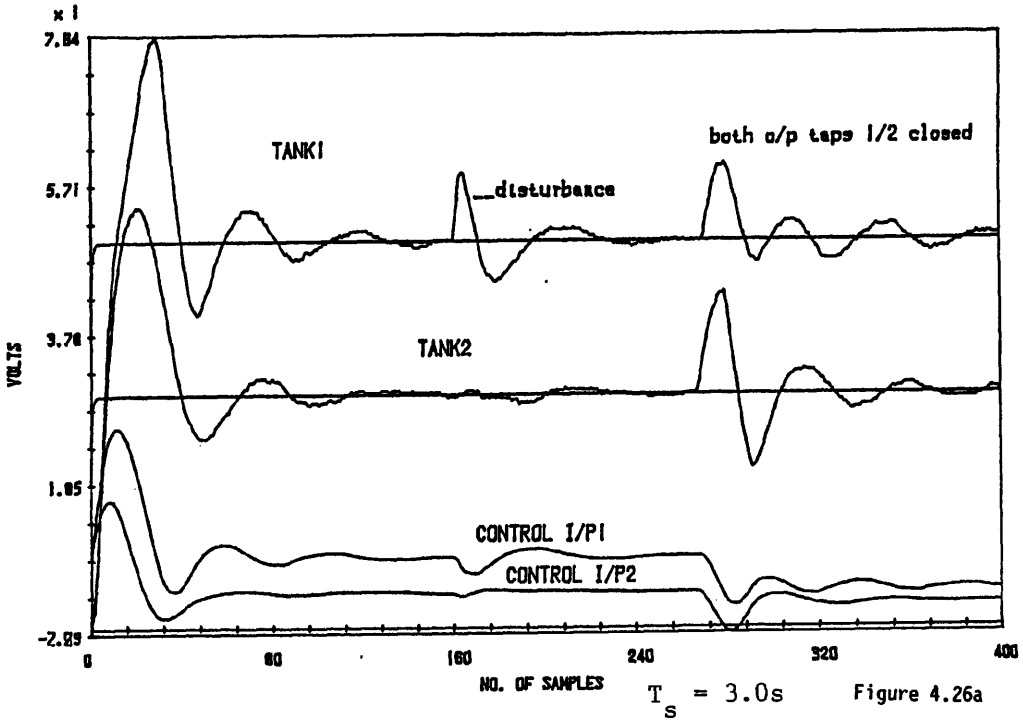


Figure 4.25b

Fig. 4.25 The results of changing on-line the reference set points for
(a) Goodwin et al's multivariable algorithm,
(b) Kreisselmeier-Anderson's multivariable algorithm.



T_s = 3.0s Figure 4.26a

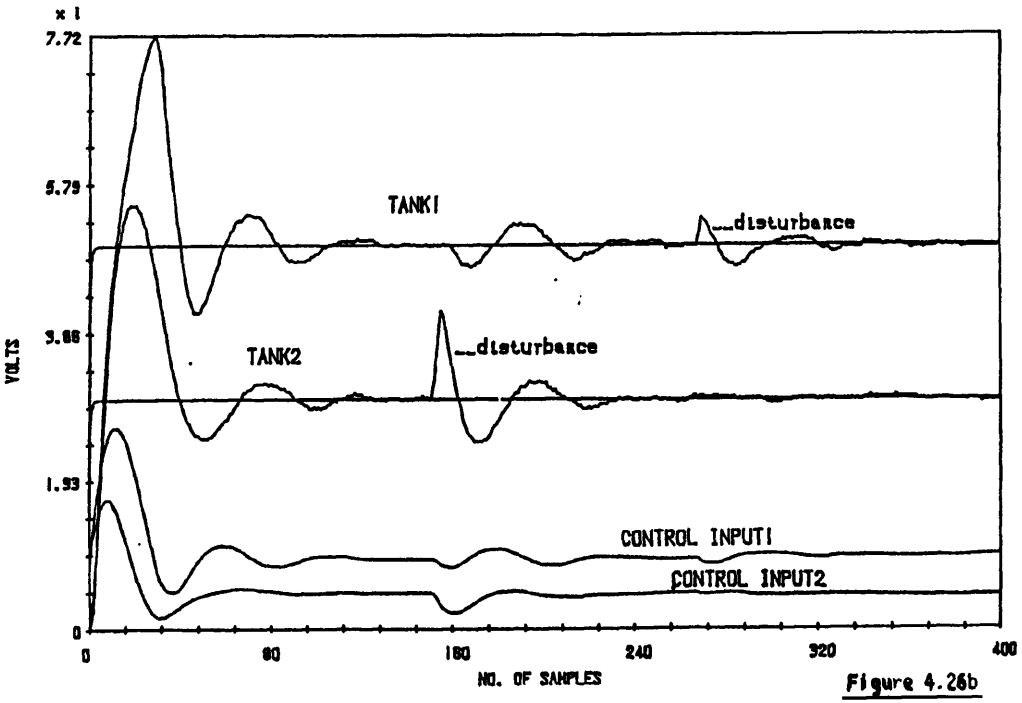
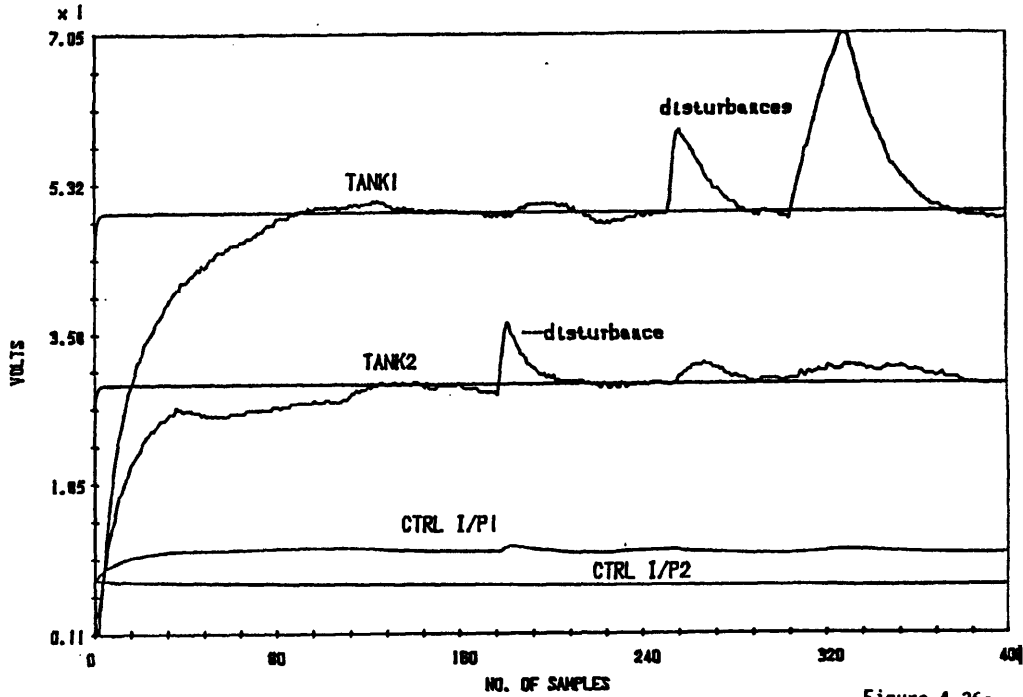


Figure 4.26b



T_s = 3.0s Figure 4.26c

Figure 4.26 Showing the effects of introducing external disturbances to the Multivariable Tanks using (a),(b) Goodwin et al's MIMO algorithm, and (c) the Kreisselmeier and Anderson MIMO algorithm.

T_s = 3.0s

Chapter 5

THE COUPLED ELECTRIC DRIVES SYSTEM

5.1 INTRODUCTION

Another interesting industrial problem involves the control of both speed and tension. This is particularly true of areas involving production of goods requiring material handling and transportation. Examples of manufacturing processes where the control of both speed and tension is of importance include the cable manufacturing industry where, in the production of say electrically insulated conductors or cables, the bare wire is unwound by a controlled tension device, the wire is then fed into a "guider" tube where the insulation coat is then added. The wire speed is controlled by a capstan at the end of the line. Other examples include (i) the textiles industry where yarns are wound from spool to spool, or (ii) the paper mills where reeling of paper sheets takes place. Here the paper sheet must be pulled on to the wind-up roll at nearly constant tension. A reduction in tension produces loose rolls while an increase could result in the paper sheets tearing. If the reel speed is constant the linear velocity of the paper and tension would increase as the wind-up roll diameter increases, hence a need for control of both tension and speed.

Other areas might be in the control of escalators and possibly conveyor belts at ports.

Unfortunately, as can be seen from these examples, these two variables influence each other a lot. Hence, the strong interaction between these variables thus makes it a challenging control problem.

The coupled electric drives apparatus (see Figure 5.1) is a laboratory scale model with which efforts are made in the study of the control of such

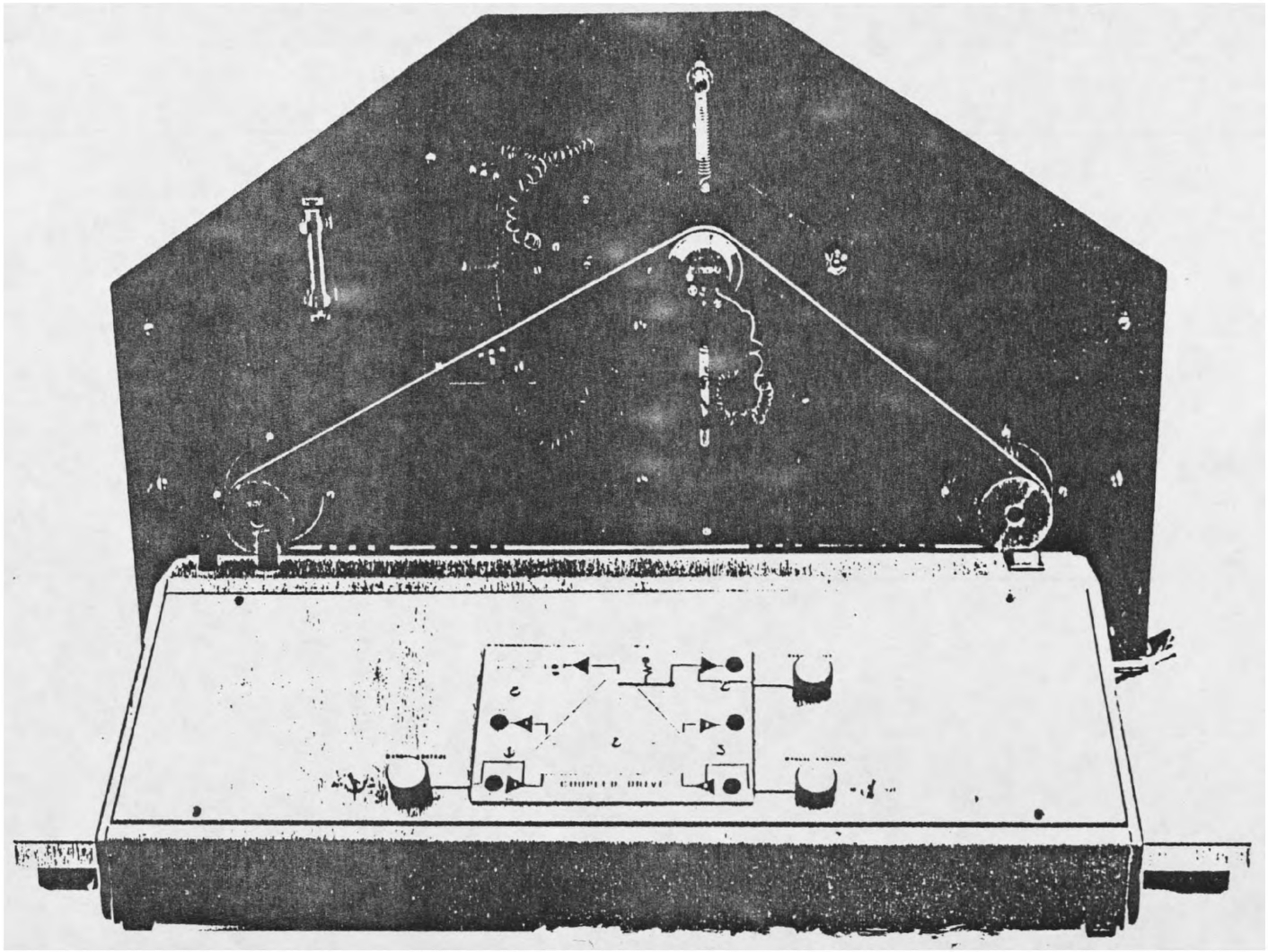


Figure 5.1 The Coupled Electric Drive Apparatus with the Instrumentation Box.

processes and the problems it entails. A brief description of the coupled electric drives system now follows.

5.2 DESCRIPTION OF RIG [see 62]

The coupled electric drives system consists of a continuous flexible belt which loops around two drive motor pulleys and over an intermediate jockey pulley (see Figure 5.2a). The apparatus is controlled by manipulating the drive torques to these servomotors. The principal system outputs are the jockey pulley speed and the belt tension, while other outputs that can be obtained are the drive speeds for motors a and b (see Figure 5.2b). The jockey pulley velocity measurement is derived from a tachogenerator which is driven directly by an extension to the jockey pulley spindle.

The belt tension is measured indirectly by monitoring the angular deflection of the pivoted tension arm to which the jockey pulley is attached. The deflection of this arm is detected by the capacitive transducer mounted at the end of the tension arm. This deflection is related via the tensioning spring stiffness and system geometry to the belt tension as it passes over the jockey arm pulley. This gives a bipolar voltage for the values of the tension.

Using the fixing device provided with the apparatus the tension arm can be locked in a fixed position to facilitate measurement of certain system characteristics and some other experiments.

Inputs to the system are by drive voltages to the servomotors through the instrumentation box. All signal processing and part of the transduction take place in the instrumentation box from which the output voltages are also measured.

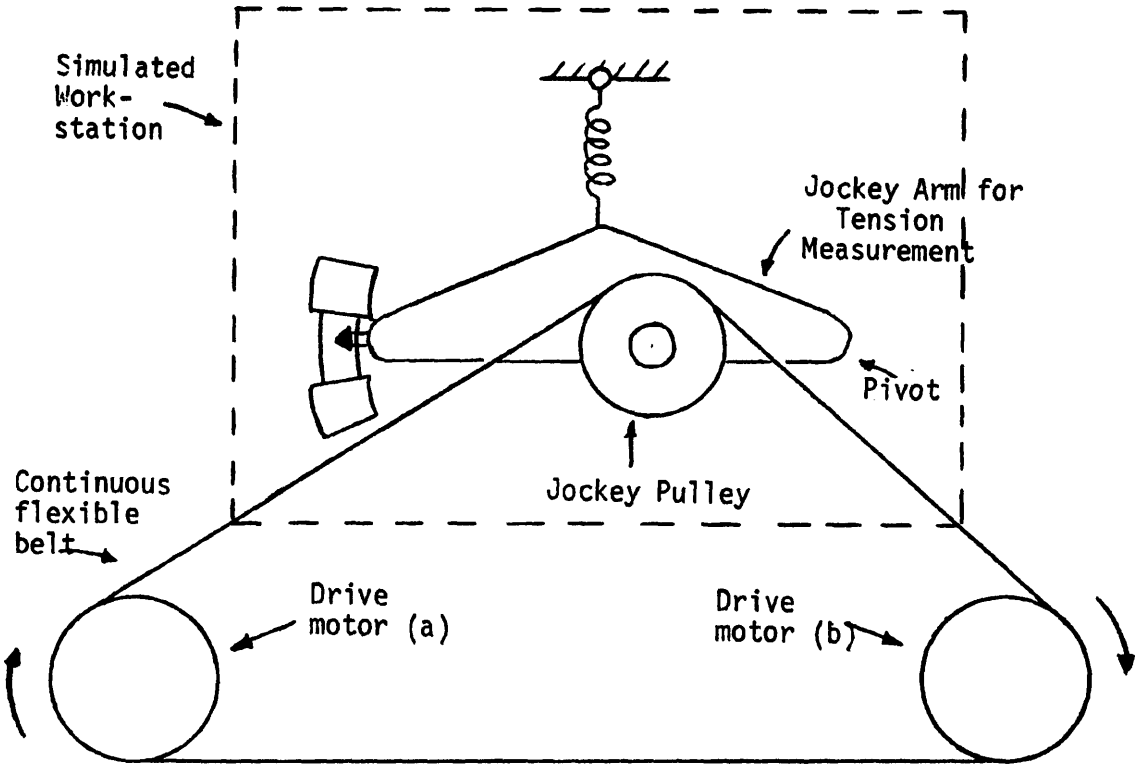


Figure 5.2a The Coupled Electric Drive Apparatus.

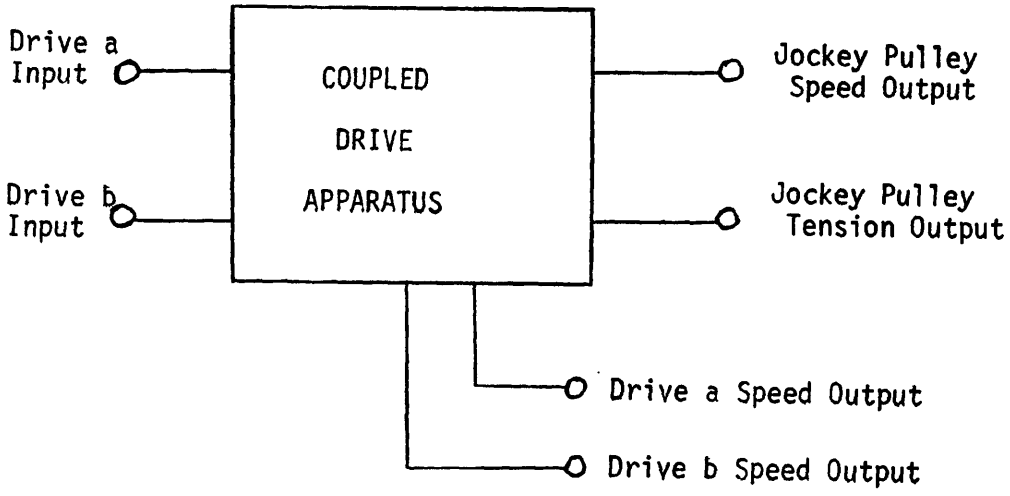


Figure 5.2b Input/Output definitions for the Coupled Electric Drives Apparatus.

5.3 THE SYSTEM CHARACTERISTICS AND MODEL

Using the free body diagram of the coupled drives shown in Figure 5.3, the transfer function of the system can be derived. This is by considering the apparatus as a combination of subsystems, e.g. the jockey pulley assembly, the drive pulleys a, b, and lastly the belt sections from which, using force balance and power conservation principles, the state space equations can be derived (see Appendix B). It is then from these that the system transfer function is found and the resulting block diagram in Figure 5.4 depicts this. The coupling between the inputs and outputs is also shown. The torque inputs for both drive motors are derived from their respective input voltages as:

$$\left.\begin{aligned} \tau_a &= g_a\, v_a \\ \tau_b &= g_b\, v_b \end{aligned}\right\} \qquad (5.3.1)$$

while the output voltages are calculated from the two outputs which are the jockey pulley speed $\omega(s)$ and the jockey arm deflection $x_k(s)$ as follows:

$$\left.\begin{aligned} V_\omega &= g_1\, \omega \\ Vx_k &= g_2\, x_k \end{aligned}\right\} \qquad (5.3.2)$$

The constants g_a, g_b are determined by the motor, amplifier characteristics, while the constants g_1, g_2 are obtained from the transducer characteristics of the system.

Due to the coupling between the two inputs and the outputs, there was a need for the interaction to be reduced for ease of control. Thus the system characteristics were best obtained under conditions where the interaction is reduced if not totally eliminated. To do this, it is fairly obvious that the use of inputs v_a, v_b, as:

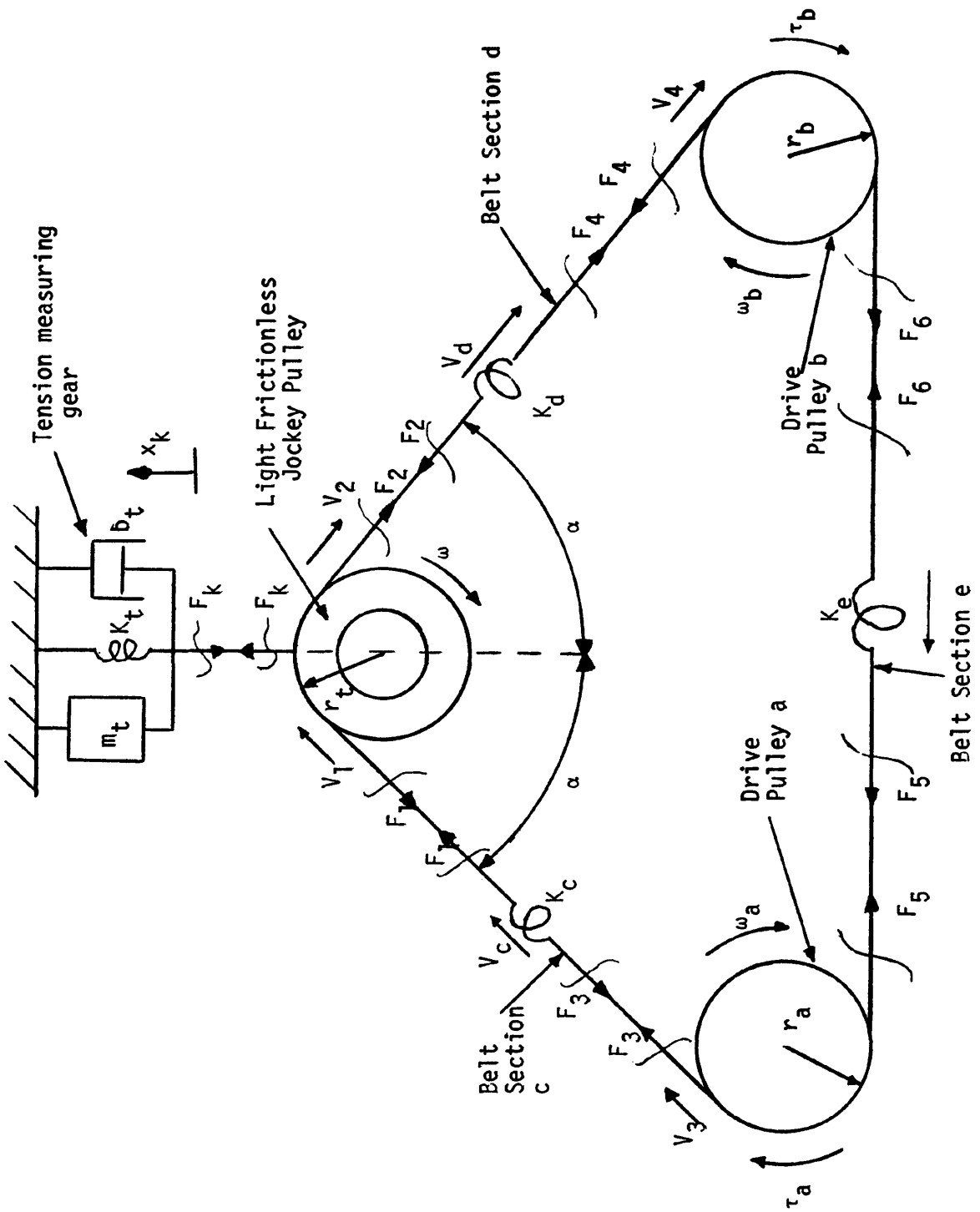


Figure 5.3 Free Body Diagram of the Coupled Electric Drives.

$$\left.\begin{aligned} v_a(s) &= u_1(s) - u_2(s) \\ v_b(s) &= u_1(s) + u_2(s) \end{aligned}\right\} \qquad (5.3.3)$$

in Figure 5.4, would reduce the interactions and thus allow one input to specifically determine one output only. This is illustrated by Figure 5.5 with

$$u_1(s) \rightarrow y_1(s)$$
$$u_2(s) \rightarrow y_2(s)$$

Figure 5.6a depicts the Tachometer calibration with the tension arm locked. Figure 5.6b shows the Tension transducer calibration, while for the steady state relationship between input to motors and jockey arm tachogenerator output see Figure 5.6c, and for that of the jockey arm tension transducer output the result is shown in Figure 5.6d. Three different curves are shown in Figure 5.6d because although the recommended u_1 voltage was six volts, it was found to lead to early saturation of both the analogue computer and the A/D converters, since this leaves only about four volts maximum for the control input. For most of the experiments and tests carried out, u_1 was set at about 3.67 volts.

To obtain the transfer functions of the coupled electric drive system, it was necessary to decouple the system so as to enable one input control one output only. Using the decoupler equation (5.3.3) the transfer functions derived mathematically from the free body diagram in Figure 5.3 were:

(i) for speed control

$$\omega(s) = \frac{g_m\, u_1(s)}{sI+b} \qquad (5.3.4a)$$

where g_m is assumed = g_a = g_b .

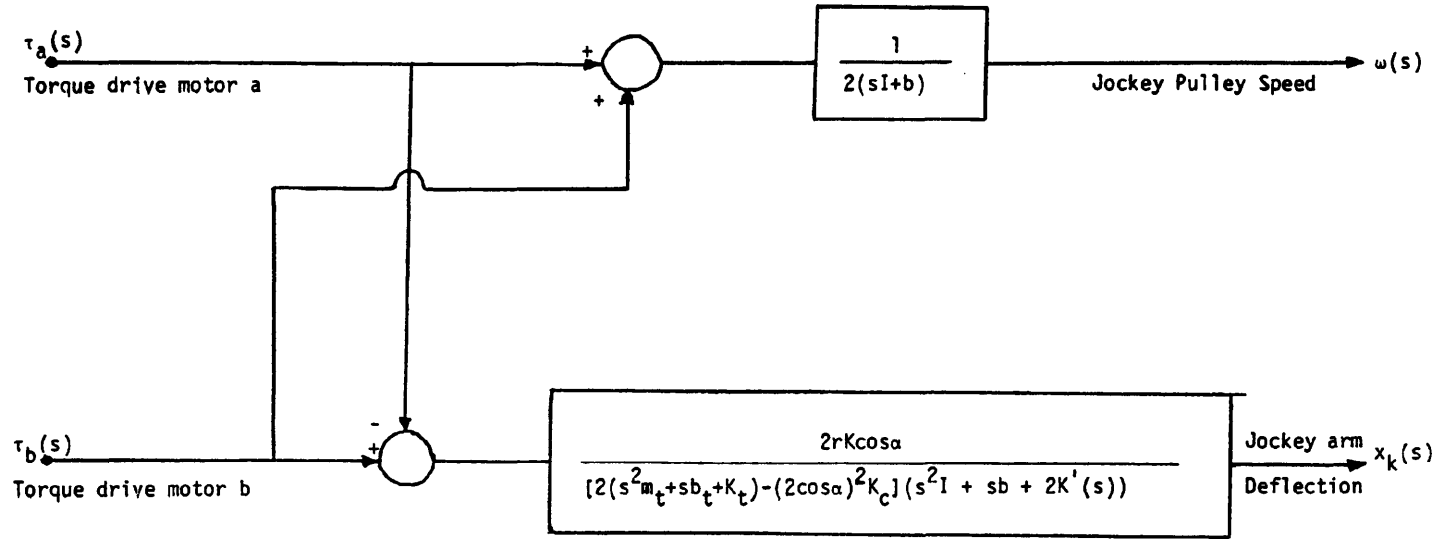


Figure 5.4 Block diagram showing the coupling and transfer functions between the outputs and inputs.

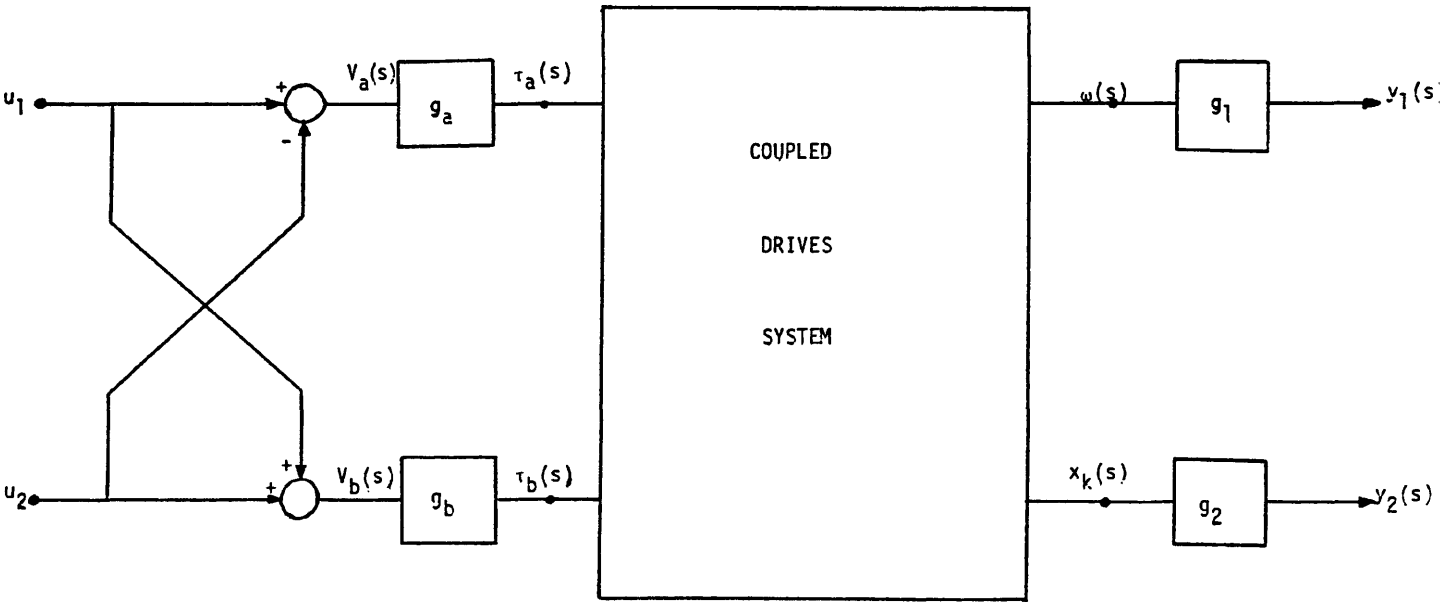


Figure 5.5 An illustration of "the decoupler" for the Coupled Drives.

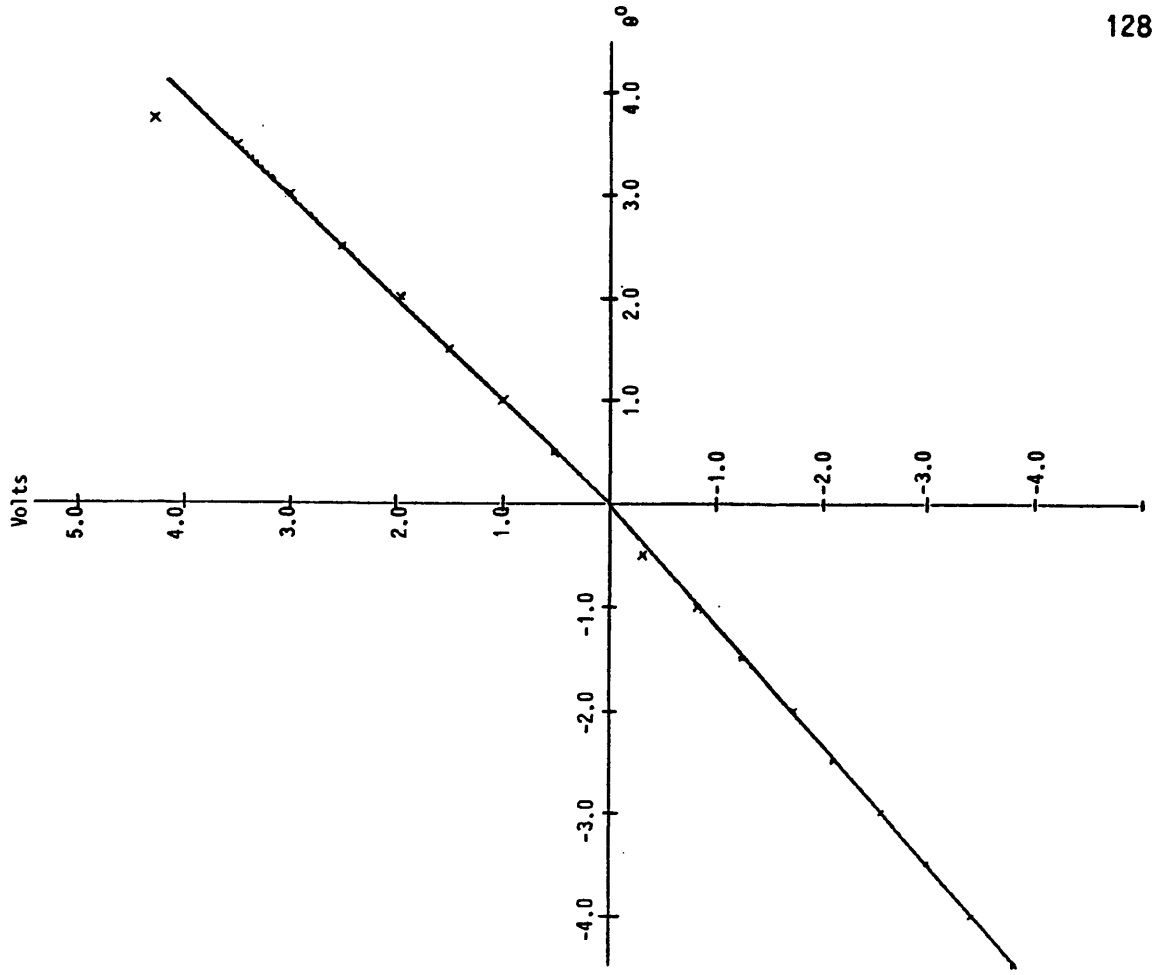


Figure 5.6 (b) The Tension Transducer calibration curve.

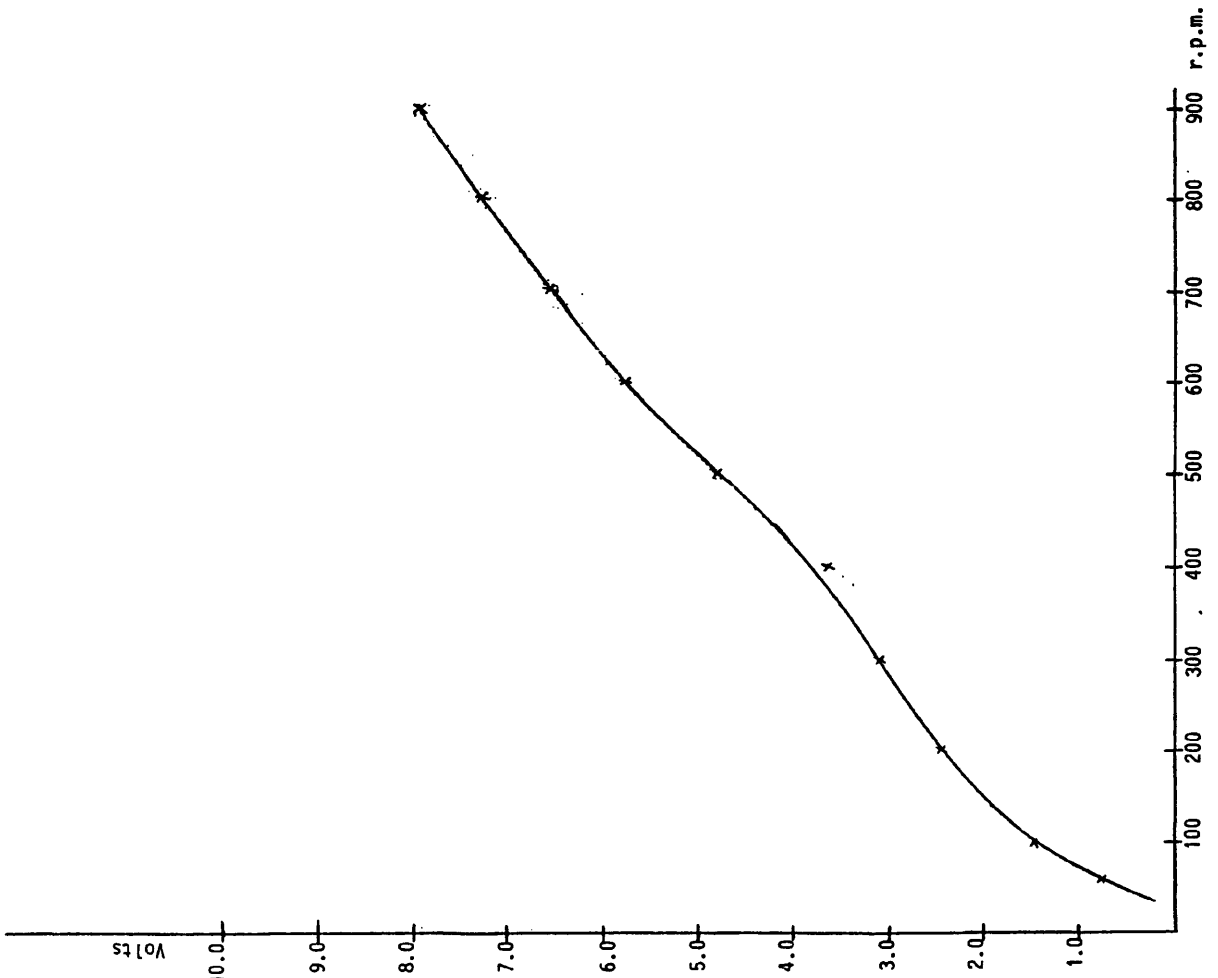


Figure 5.6 (a) The Jockey Arm Tachogenerator calibration curve.

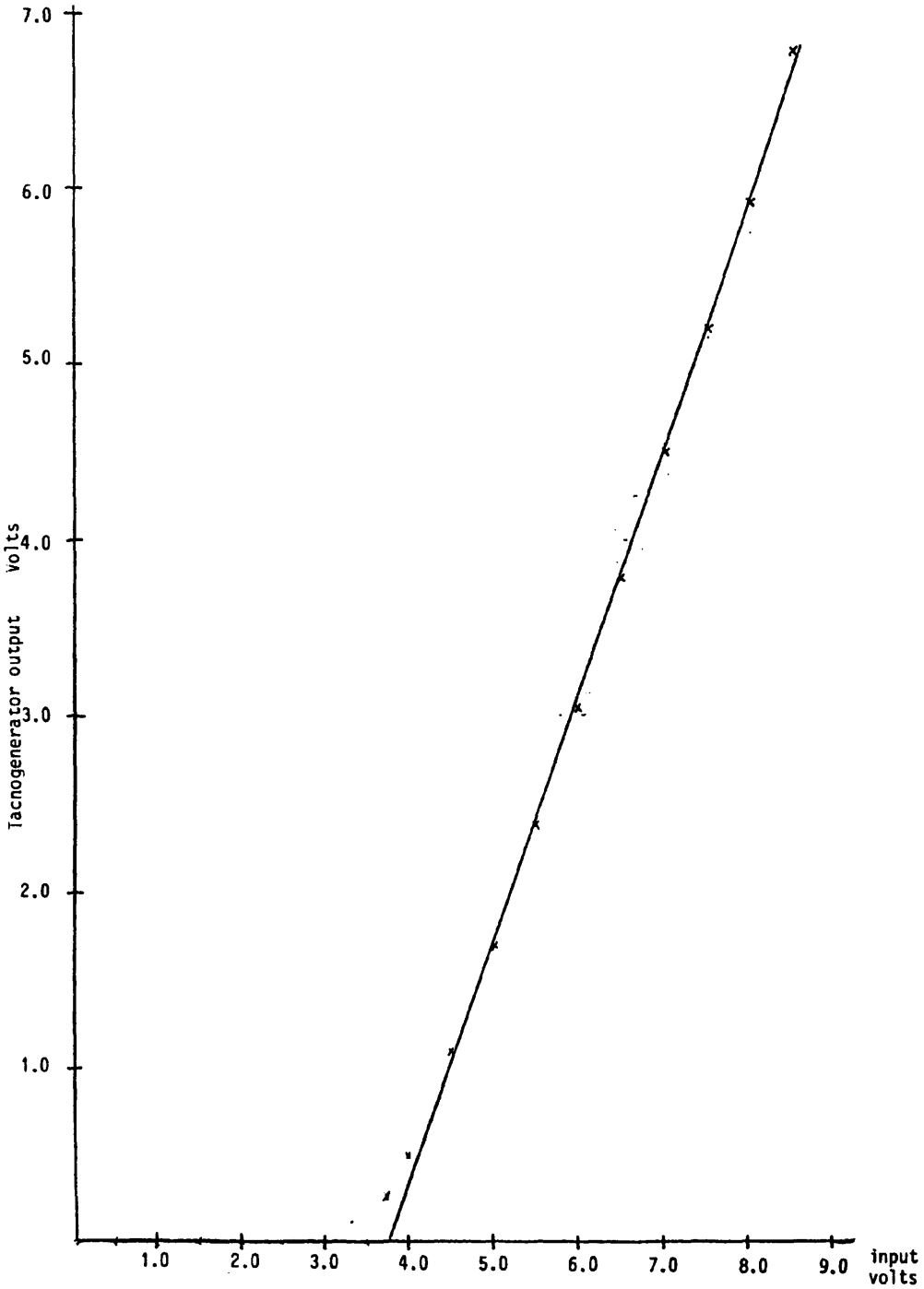


Figure 5.6 (c) The steady state relationship between inputs to motors and the Jockey Arm tachogenerator output.

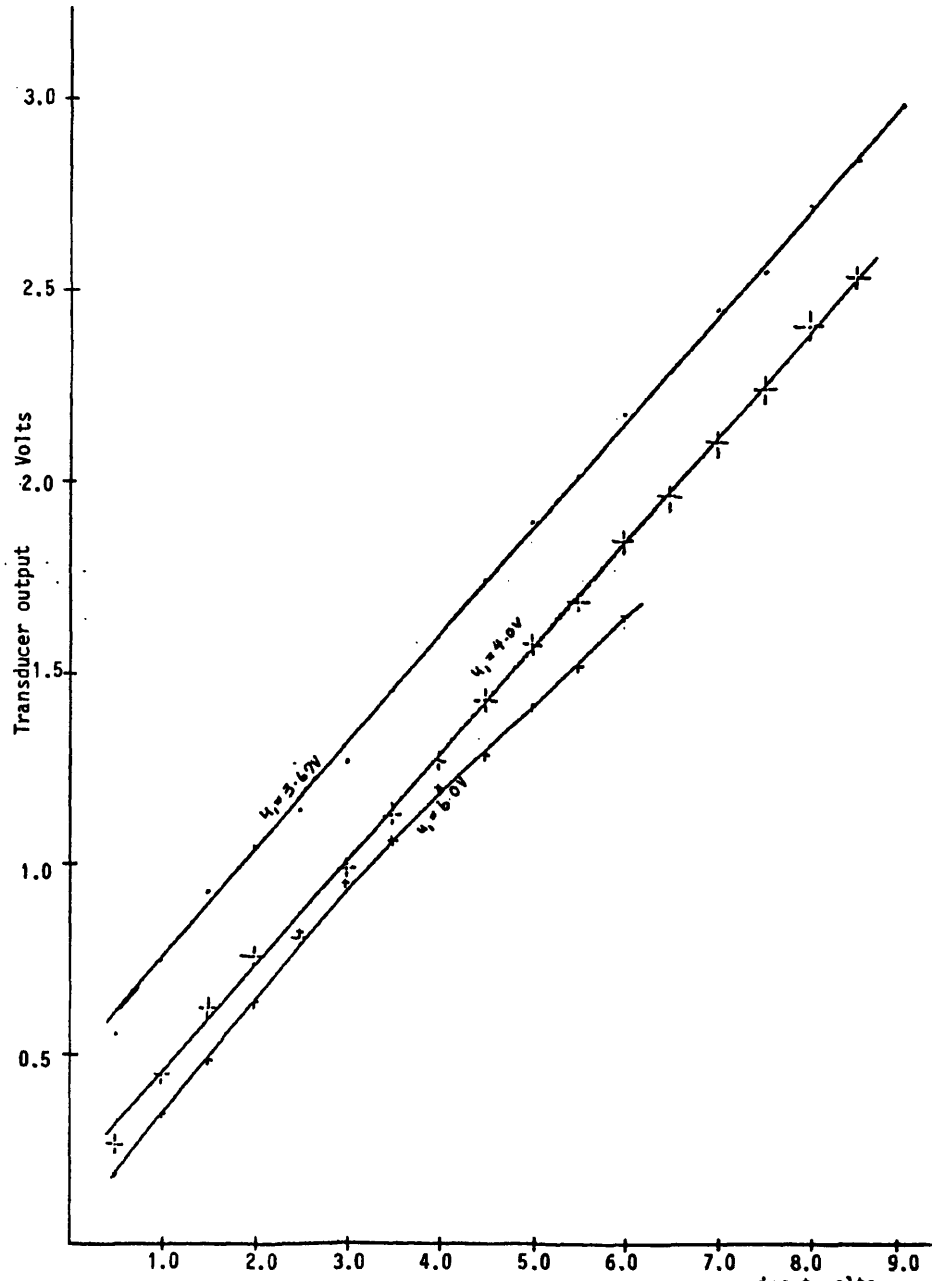


Figure 5.6 (d) The steady state relationship between input to motor and the Jockey Arm tension transducer output.

(ii) for tension control

$$x_k(s) = \frac{4g_m r K \cos\alpha \; u_2(s)}{(s^2 I + bs + 2K'(s))(2[s^2 m_t + sb_t + K_t] - (2\cos\alpha)^2 K)} \qquad (5.3.4b)$$

where

$$K'(s) = Kr^2\left[1 + \frac{s^2 m_t + sb_t + K_t}{2[s^2 m_t + sb_t + K_t] - (2\cos\alpha)^2 K}\right] \qquad (5.3.4c)$$

(N.B. This accounts for the jockey arm dynamics.)

and $b_a = b_b = b$ the rotational coefficient assumed same for both pulleys;

m_t, K_t, b_t are as shown in Figure 5.3;

$r_a = r_b = r$ the radius of the pulleys assumed to be the same;

$K_c = K_d = K_e = K$ the belt stiffness coefficient assumed to be the same for the different belt sections.

It should be noted that if the tension arm is damped using the locking bar, then $K'(s)$ reduces to a constant:

$$K'(s) = K' = r^2\left[K_e + \frac{K_c}{1 + K_c/K_d}\right]$$

$$= \frac{3}{2} K r^2 \qquad (5.3.4d)$$

The versatility of the coupled electric drives apparatus is shown in the way it could be modified and reduced to a scalar system in which only one variable is controlled, e.g.

Speed Control alone by using $v_a(s) = u_1(s)$, $v_b(s) = u_1(s)$ (5.3.5a)

Tension Control alone by using $v_a(s) = u_2(s)$, $v_b(s) = -u_2(s)$ (5.3.5b)

(note, this is for positive tension output $x_k(s)$, if a negative output is

required the signs of the above signals are reversed.)

5.4 MODEL REFERENCE ADAPTIVE CONTROL OF THE ELECTRIC DRIVES

Although the coupled electric drives apparatus is not quite a nonlinear system it is still a good rig to implement MRAC algorithms on. Tests were carried out using the different algorithms on the MIMO system consisting of the inputs to the drive motors and two outputs, namely the jockey pulley speed and the belt tension. Also carried out were tests on the scalar systems of tension control and speed control, respectively. For the scalar systems more emphasis was placed on the tension control since, by virtue of its transfer function, it makes a more interesting study, especially as regards reduced order modelling. The discussion of the results now follows, starting with the scalar systems.

5.4.1 Tension Control

To make the work easier the coupled drive system was decoupled using the inputs $v_a = u_2(s)$, $v_b = -u_2(s)$ (for positive tension voltage output $y_2(s)$), but this was not a recommended approach from the systems manual as it might result in damage to the circuits of the servo-motors due to large current build up. Hence, for decoupling the system was connected up with inputs:

$$v_a = u_1 + u_2 \qquad \text{for positive tension outputs } y_2(s)$$
$$v_b = u_1 - u_2$$

The voltage u_1 is a constant voltage which was generated from an analogue computer to keep the drive-motors rotating at a nominal speed, while voltage u_2 was derived from the BBC microcomputer as the adaptive control input for tension control.

Three different reference models were applied during tests carried out using the three different algorithms, namely two first-order models and a second-order model. The sampling period varied for different models but were either half a second or a second long, with a time delay of one sampling period interval assumed.

Control Hardware and Software:

The output voltage representing the tension output $x_k(s)$ was the input to the BBC computer from which the control input u_2 was also derived. The analogue computer was used to generate u_1 and also for both summing and inverting the voltages. The software used was written in BASIC. The listing of the programs can be seen in Appendix C. The results were first stored on floppy disks before being plotted using the Rikadenki plotter.

Real Time Control Experiments (the results):

As mentioned earlier, three different reference models were used, a fast first-order model, a slow first-order model and a second-order model. The results of the different MRAC algorithms are now explained, starting with the Goodwin algorithm.

A. The GOODWIN et al Algorithm:

Key to the program (see Appendix C):

The control input $u(t) = Y(6)$, $u(t-1)$ Y(7)

The reference input $R(t) = R$

The reference output $y_m(t) = Y(4)$ $y_m(t-1) = Y(19)$

The system output $y_p(t) = YS$ $y_p(t-1) = Y(5)$

The tracking error $e(t) = Y(9)$ $e(t-1) = Y(10)$

The adaptive parameters $\Theta(t) = [\Theta_1(t), \Theta_2(t), \Theta_3(t)] = [Y(11), Y(12), Y(13)]$

$\Theta(t-1) = [Y(14), Y(15), Y(16)]$

(i) Model 1 $y_m(t) = 0.6065y_m(t-1) + 0.3935r(t)$

The initial parameter values were $\Theta(0) = [0,0,0]$ and the results are shown in Figures 5.7a,b using a gain of 0.39, with Figure 5.7a showing the reference and system outputs plus the control input, while the adaptive parameters are shown in Figure 5.7b. As in Chapter 4, the use of a bigger gain, 0.78, leads to quicker convergence time, though an overshoot also occurs as depicted in Figure 5.7c.

In pursuit of the robustness of the algorithm, other experiments were carried out involving changes in the step inputs to see whether the system output could be destabilized through these changes. Also, disturbances were intentionally introduced as well to check on the system response. The step input was changed down as shown in Figure 5.8a, while in others the step input was changed down and then up as in Figures 5.8b,c (which depict the control input plus reference and system outputs and the adaptive parameters respectively), or as in Figures 5.8d,e where the set input change was first up and then down again. From Figures 5.8b,d it can be observed how the control input goes back to the same value for the same output, thus confirming the adaptiveness of the algorithm.

Disturbances to the system output were introduced in another experiment by shaking the tension wire up and down after the convergence of both outputs and then releasing it; later the tension was increased by tightening the belt which thus changed the output voltage to see whether the algorithm could cope with this. The results of all these external disturbances are depicted in Figures 5.9a,b,c.

(ii) Model 2 $y_m(t) = 0.0198r(t) + 0.9802y_m(t-1)$

Using a slower reference model the system was made to track the reference model output, using a gain of 0.78, with the initial parameter vector the same

value as in (i) above. The results are shown in Figures 5.10a,b,c.

(iii) Model 3a $y_m(t) = 0.000413r(t-1) + 1.97589y_m(t-1) - 0.9763y_m(t-2)$

3b $y_m(t) = 0.0015603r(t-1) + 1.9512y_m(t-1) - 0.98334y_m(t-2)$

3c $y_m(t) = 0.000259r(t-1) + 1.98308y_m(t-1) - 0.98334y_m(t-2)$

The above models being second-order functions, the number of adaptive parameters had to increase due to the increase in model order, but the models were designed for the same damping factor, even though the natural frequencies were different. The results of the first model are shown in Figures 5.11a,b with (a) showing the control input and reference plus system outputs, while (b) shows the adaptive parameters and the tracking error. The reference and system outputs for the other two models are shown in Figures 5.11c and d, respectively.

Comments about all the results and figures will be made at the end of the chapter.

B. The ORTEGA et al Algorithm

Key to the program is similar to that of (A).

(i) Model 1 $y_m(t) = 0.3935r(t) + 0.6065y_m(t-1)$

Using different gains but the same initial adaptive parameter vector $\Theta(0) = [0.1, 0.1, 0.1]$, different experiments were carried out on the rig. In Figure 5.12a the control input, reference and system outputs are shown, while Figure 5.12b shows the adaptive parameters for a gain of 0.62, and when the gain was increased to 0.78, this led to an overshoot shown in Figure 5.12c. Amongst other tests done were the changing of the reference input on-line, the result of which is shown in Figure 5.13a,b.

(ii) Model 2 $y_m(t) = 0.0198r(t) + 0.9802y_m(t-1)$

The results are shown in Figures 5.14a,b,c for the reference and system outputs, the control input and the adaptive parameters. For this experiment an adaptive gain of 1.565 was used.

(iii) Model 3 $y_m(t) = 0.000413r(t-1) + 1.97589y_m(t-1) - 0.9763y_m(t-2)$

For this reference model a bigger adaptive gain was used because of the results obtained using the second model above. A gain of 1.96 was used, and the results are shown in Figures 5.15a,b,c, respectively.

(C) The KREISSELMEIER, ANDERSON Algorithm

(i) Model 1 $y_m(t) = 0.3935r(t) + 0.6065y_m(t-1)$

The same initial parameter values as in (B) above were used with the results shown in Figure 5.16a of the reference and system outputs. In Figures 5.16b,c the results shown are for when the reference input was changed to a lower value on-line, while Figures 5.16d,e show the results for when the reference input was changed to a higher value.

(ii) Model 2 $y_m(t) = 0.0198r(t) + 0.9802y_m(t-1)$

The results are shown in Figures 5.17a,b,c of the reference and system outputs, the control input and the adaptive parameters.

5.4.2 Speed Control

Before moving on to the multivariable system, a couple of experiments were carried out on the speed control of the coupled electric drives with the tension output decoupled by the use of the decoupling equation mentioned earlier. The inputs to the drive motors were the same, namely the control input $u(t)$ derived from the BBC micro, i.e. $V_a = u(t)$, $V_b = u(t)$, as the use of this decouples the system. Using the Goodwin algorithm, the results are shown for different gains in Figures 5.18a,b,c. In Figure 5.18a, it is shown that while theoretically, by the use of the same input to the drives to decouple, the tension output should be zero, in practice there is still a bit of interaction probably due to the asymmetricalness of the whole system.

5.4.3 The Multivariable Coupled Electric Drives System

For the multivariable system consisting of the two drive motors as inputs, the jockey pulley speed and belt tension as outputs, making it a 2x2 system, only two algorithms were tested on it. These were the multivariable Goodwin algorithm and the derived MIMO Kreisselmeier-Anderson algorithm.

Although there is interaction between the system inputs and outputs, as mentioned in earlier sections, it is possible to decouple the rig into two almost independent systems by using the inputs as defined in equation (5.3.3), but for the two algorithms used here, there was no explicit attempt at using the decoupler equation mentioned, instead the algorithms were applied with the aim of achieving independent control of the outputs irrespective of the interactions.

The algorithms and their results for different tests are now explained as follows:

A. The MIMO GOODWIN et al Algorithm

Key to the program (see Appendix C):

$y_{m1}(t) = Y(9)$ $\quad y_{m1}(t-1) = Y(10)$

$y_{m2}(t) = Y(17)$ $\quad y_{m2}(t-1) = Y(18)$

$y_{w(s)}(t) = Y(1)$ $\quad y_{w(s)}(t-1) = Y(2)$

$y_{x_k(s)}(t) = Y(3)$ $\quad y_{x_m(s)}(t-1) = Y(4)$

$u_1(t) = Y(47)$ $\quad u_1(t-1) = Y(5)$ $\quad u_1(t-2) = Y(6)$

$u_2(t) = Y(48)$ $\quad u_2(t-1) = Y(7)$ $\quad u_2(t-2) = Y(8)$

$e_1(t) = Y(19)$ $\quad e_2(t) = Y(20)$

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} = \begin{bmatrix} Y(27) & Y(28) \\ Y(37) & Y(38) \end{bmatrix}$$

$$\Theta(t) = \begin{bmatrix} \Theta_1(t) \\ \Theta_2(t) \end{bmatrix} = \begin{bmatrix} \Theta_{11}(t) & \Theta_{12}(t) & \Theta_{13}(t) & \Theta_{14}(t) & \Theta_{15}(t) & \Theta_{16}(t) \\ \Theta_{21}(t) & \Theta_{22}(t) & \Theta_{23}(t) & \Theta_{24}(t) & \Theta_{25}(t) & \Theta_{26}(t) \end{bmatrix}$$

$$= \begin{bmatrix} Y(11) & Y(12) & Y(13) & Y(14) & Y(15) & Y(16) \\ Y(21) & Y(22) & Y(23) & Y(24) & Y(25) & Y(26) \end{bmatrix}$$

A possible way of decoupling the system using the P-matrix was investigated. This involves using the P-matrix as a decoupler. Basically what was done could be explained using the definitions of the control inputs, e.g. $\underline{u}(t) = \Theta(t)\Phi(t)$.

From Chapter 2 it is known that

$$\theta(t) = \theta(t-1) - \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix} [1+\phi^T(t)\phi(t)]^{-1}\phi^T(t) \tag{5.4.1}$$

$$\Rightarrow \quad \left.\begin{aligned} \theta_1(t) &= \theta_1(t-1) - [P_{11}e_1(t) + P_{12}e_2(t)]\cdot K(t)\phi^T(t) \\ \theta_2(t) &= \theta_2(t-1) - [P_{21}e_1(t) + P_{22}e_2(t)]\cdot K(t)\phi^T(t) \end{aligned}\right\} \tag{5.4.2}$$

where $K(t) = [1 + \phi^T(t)\phi(t)]^{-1}$

For speed control alone (remembering equation (5.3.5a)) one obtains:

$$\begin{aligned} \theta_1(t) &= \theta_1(t-1) - [P_{11}e_1(t) + 0.e_2(t)]\cdot K(t)\phi^T(t) \\ \theta_2(t) &= \theta_2(t-1) - [P_{11}e_1(t) + 0.e_2(t)]\cdot K(t)\phi^T(t) \end{aligned}$$

$$\Rightarrow \quad \theta(t) = \theta(t-1) - \begin{bmatrix} P_{11} & 0 \\ P_{11} & 0 \end{bmatrix} \begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix} \cdot K(t)\phi^T(t) \tag{5.4.3}$$

Provided the same initial parameter values were used for $\theta_1(t)$ and $\theta_2(t)$, i.e. $\theta_1(0) = \theta_2(0)$, then $\theta_1(t) = \theta_2(t)$ for all t and $u_1(t) = u_2(t)$ as well.

Similarly for the tension control only (noting equation (5.3.5b)):

$$\theta(t) = \theta(t-1) - \begin{bmatrix} 0 & -P_{22} \\ 0 & P_{22} \end{bmatrix} \begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix} K(t)\phi^T(t) \tag{5.4.4}$$

for negative tension voltage

$$\theta(t) = \theta(t-1) - \begin{bmatrix} 0 & P_{22} \\ 0 & -P_{22} \end{bmatrix} \begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix} K(t)\phi^T(t) \qquad (5.4.5)$$

for positive tension voltage

Thus, for independent control of the two outputs one needs:

$$\theta(t) = \theta(t-1) - \begin{bmatrix} P_{11} & P_{22} \\ P_{11} & -P_{22} \end{bmatrix} \begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix} K(t)\phi^T(t) \qquad (5.4.6)$$

For all the work carried out using this algorithm the above P-matrix form was used. There was only one first-order reference model used for the tests, although with the reference inputs in different variations, i.e. changing them up or down on-line or with higher reference outputs for either the tension or speed control.

A sampling time of half a second was used and the reference model with the results are given below.

The control experiments and results:

Model 1:
$$y_{m1}(t) = 0.3935r_1(t) + 0.6065y_{m1}(t-1)$$
$$y_{m2}(t) = 0.3935r_2(t) + 0.6)65y_{m2}(t-1)$$

Using a reference input of four volts for the speed control and two volts for the tension control, the coefficients of the P-matrix used were $P_{11} = 0.1$, $P_{22} = 0.15$. The results are shown in Figure 5.19a for both system outputs and reference outputs, but because of the long duration of the experiment, due to the large number of samples, it was only possible to store the results of these on disk, hence for all the figures there were no control inputs or adaptive parameters shown.

Also used were a higher reference tension input of three volts and a lower reference speed input (1.75 volts), the result of which is depicted in

Figure 5.19b.

Finally the reference inputs were changed on-line to test more adaptive properties of the algorithm and system reaction. The results of these are shown in Figure 5.19c.

B. The MIMO KREISSELMEIER-ANDERSON Algorithm

Key to the program (see Appendix)

$y_{m1}(t) = Y(9)$ $y_{m1}(t-1) = Y(10)$

$y_{m2}(t) = Y(17)$ $y_{m2}(t-1) = Y(18)$

$y_{w(s)}(t) = Y(1)$ $y_{w(s)}(t-1) = Y(2)$ $y_{w(s)}(t-2) = Y(55)$

$y_{w(s)}(t-3) = Y(57)$

$y_{x_k(s)}(t) = Y(3)$ $y_{x_k(s)}(t-1) = Y(4)$ $y_{x_k(s)}(t-2) = Y(56)$

$y_{x_k(s)}(t-3) = Y(58)$

$u_1(t) = Y(47)$ $u_1(t-1) = Y(5)$ $u_1(t-2) = Y(6)$ $u_1(t-3) = Y(53)$

$u_2(t) = Y(48)$ $u_2(t-1) = Y(7)$ $u_2(t-3) = Y(8)$ $u_2(t-3) = Y(54)$

$m_1(t) = X(2)$ $m_1(t-1) = X(4)$

$m_2(t) = X(5)$ $m_2(t-1) = X(7)$

$$\theta(t) = \begin{bmatrix} \theta_{11}(t) & \theta_{13}(t) & \theta_{14}(t) & \theta_{15}(t) & \theta_{16}(t) \\ \theta_{22}(t) & \theta_{23}(t) & \theta_{24}(t) & \theta_{25}(t) & \theta_{26}(t) \end{bmatrix} = \begin{bmatrix} Y(11) & Y(13) & Y(14) & Y(15) & Y(16) \\ Y(22) & Y(23) & Y(24) & Y(25) & Y(26) \end{bmatrix}$$

The same model as the Goodwin algorithm was used, as well as the same sampling period. Typical of the other experiments using this algorithm was the need to have a rough idea of the adaptive parameter bounds to be used in the algorithm. This is usually done by deleting any parameter bounds initially from the algorithm and conducting experimental runs without them. From these experiments the minimum and maximum parameter bounds can be determined. An example of the results of such an experiment is depicted in Figure 5.20a which leads to the inclusion of parameter bounds in the software. The result in

Figure 5.20b is that of using the algorithm complete with parameter bounds, dead zones and normalization.

5.5 COMMENTS

(i) The system was very noisy due to the vibrations of the tensional spring which is attached to the deflection arm. The vibrations were caused by the movement of the belt at speed. The lower the speed of the belt, the greater the vibrations of the deflection arm which reads the tension output. Due to these vibrations, the tension output tends not to stay constant as the results show, although the quantization error contributed to this as well.

(ii) Another problem which also relates to the tension control was the initial oscillation of the tension output during the beginning of any experiment, although within 2-4 samples this is normally brought under control. This led to all the results having shaky beginnings. Different attempts at stopping this were tried all to no avail, although the smaller the initial input the smaller these oscillations were.

(iii) All the algorithms performed quite well on the rig, although once again the best results came from the Ortega et al algorithm for the SISO algorithms, with the Kreisselmeier-Anderson algorithm being the worst of all three, especially when tracking the slower first-order reference model.

For the multivariable algorithms the Kreisselmeier-Anderson derived MIMO algorithm proved faster in converging the system outputs to the desired reference outputs. Comparisons can be made between Figures 5.19a and 5.20b.

(iv) When the reference output changed on-line there seemed to be difficulties in getting the system output to change fast enough, which

might be due to the mis-modelling of the system order, hence not allowing the adaptive parameters to converge to the true parameter values; but better convergence is achieved when changing up then down.

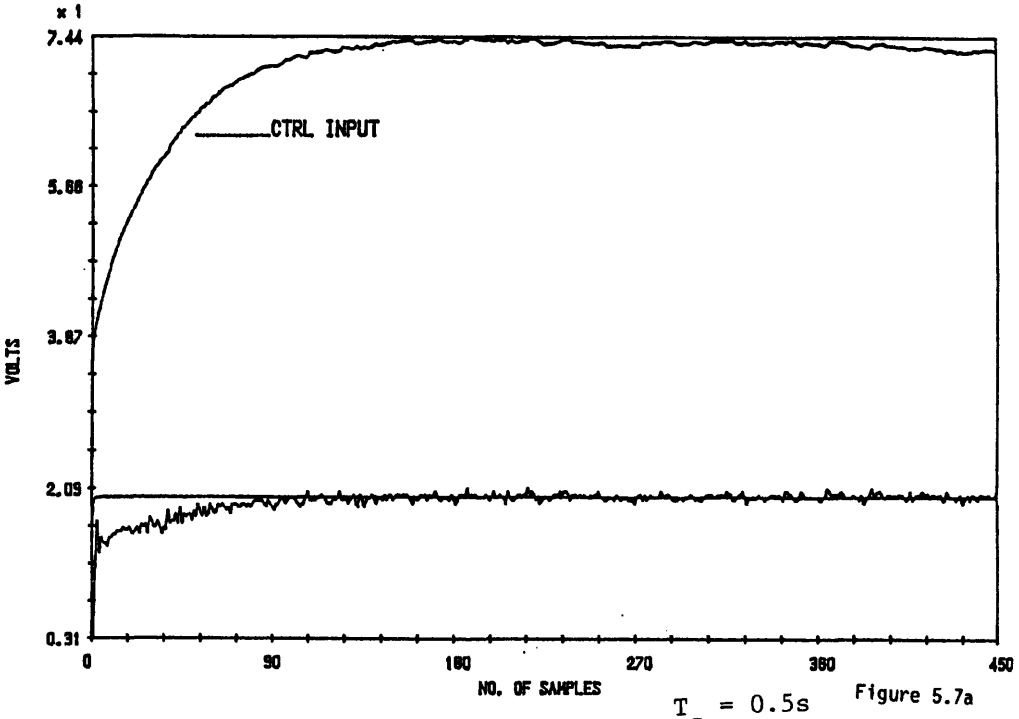


Figure 5.7a

T_s = 0.5s

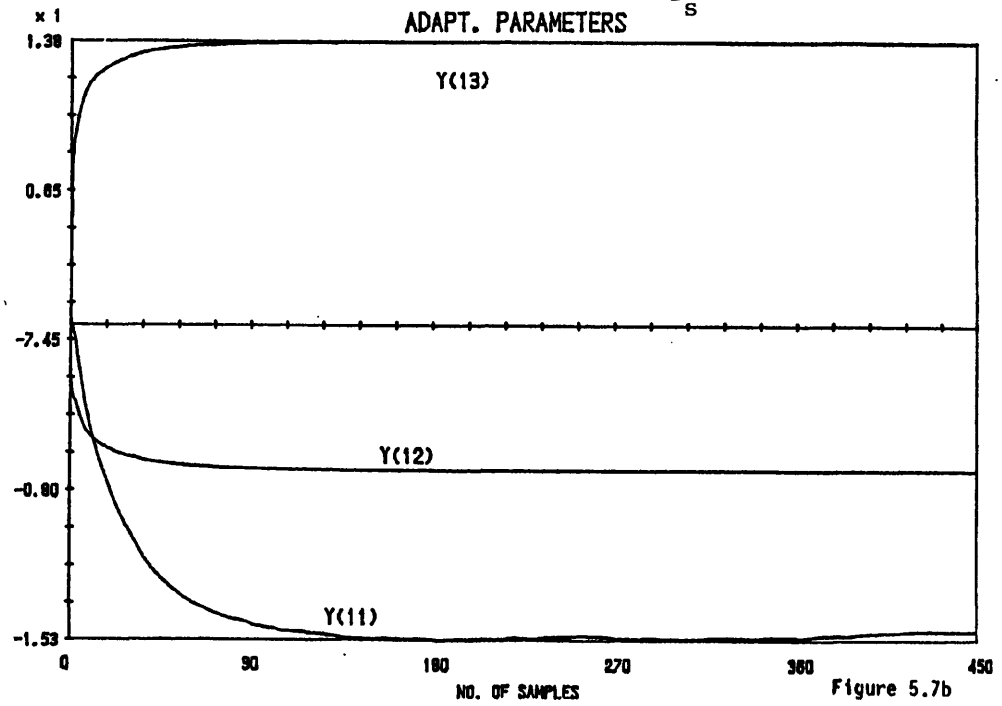


Figure 5.7b

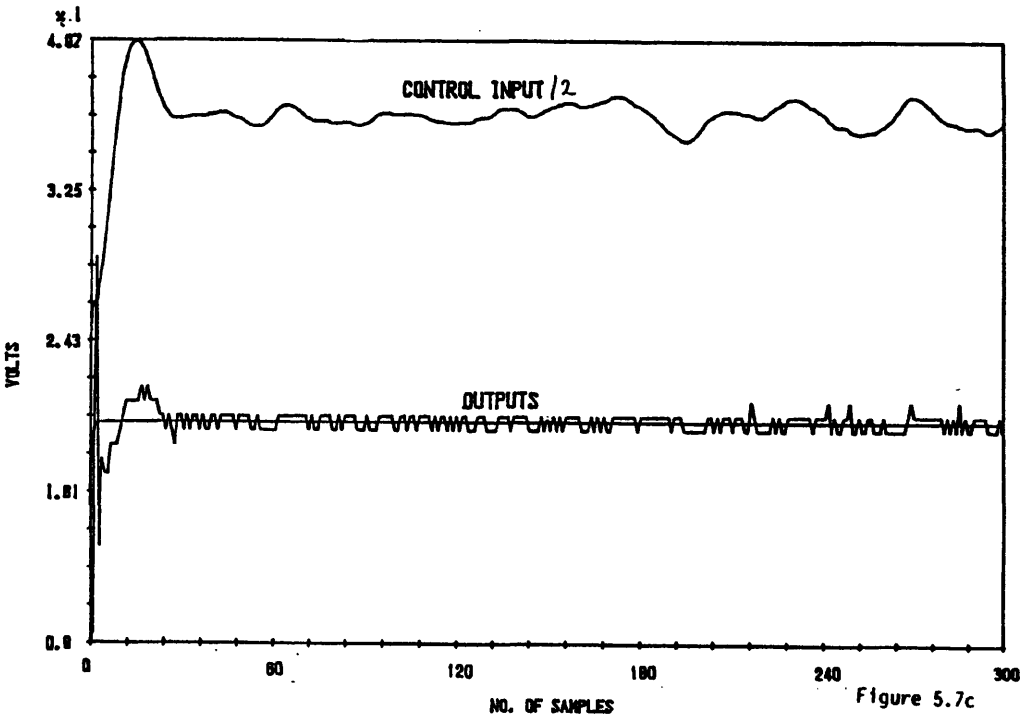


Figure 5.7c

T_s = 0.5s

Figure 5.7 (a) The reference, system outputs plus the control input using the Goodwin et al algorithm.

(b) The adaptive parameters $\theta(t)$.

(c) The reference, system outputs plus the control input for a bigger adaptive gain using the Goodwin et al algorithm.

T_s = 0.5s

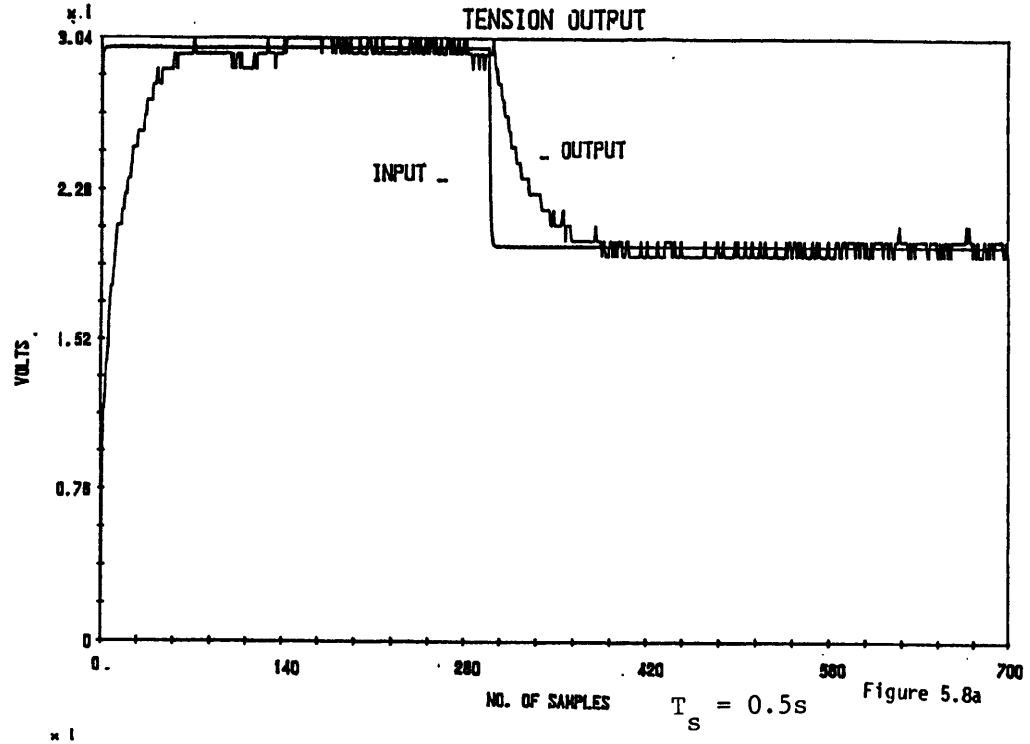


Figure 5.8a

T_s = 0.5s

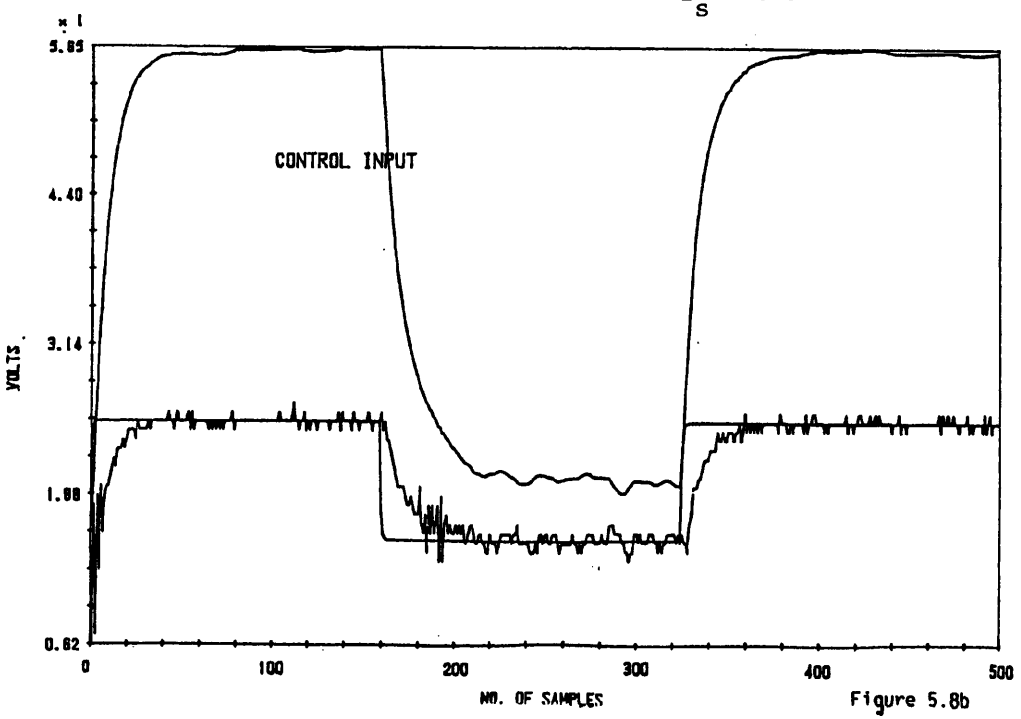


Figure 5.8b

T_s = 0.5s

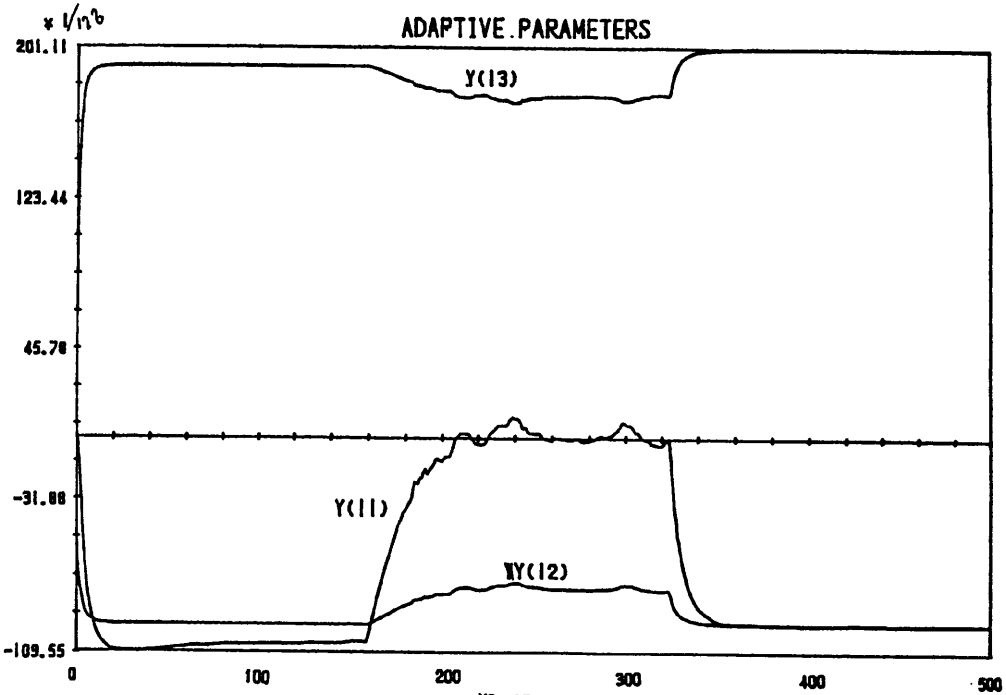
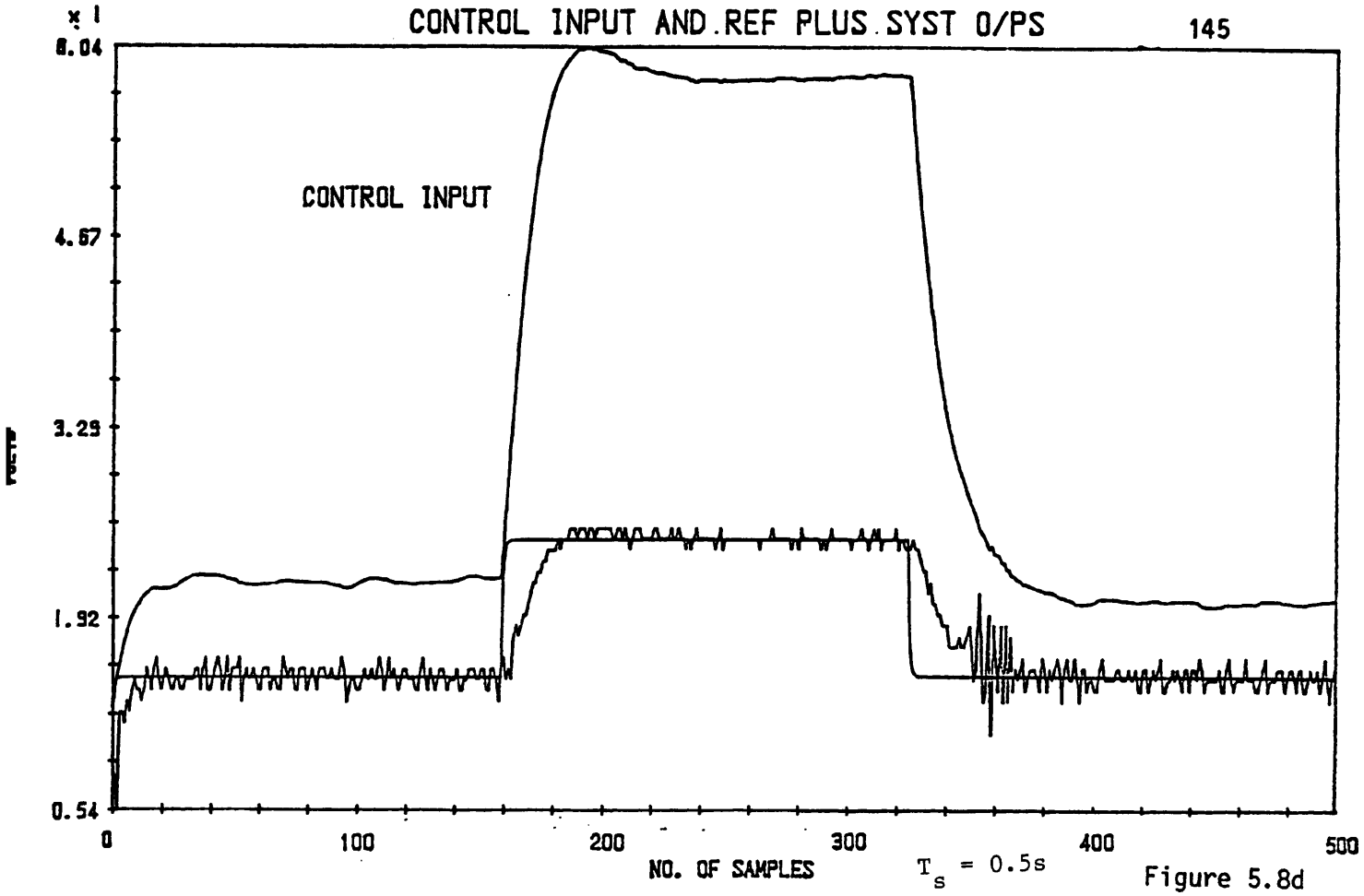


Figure 5.8c

T_s = 0.5s

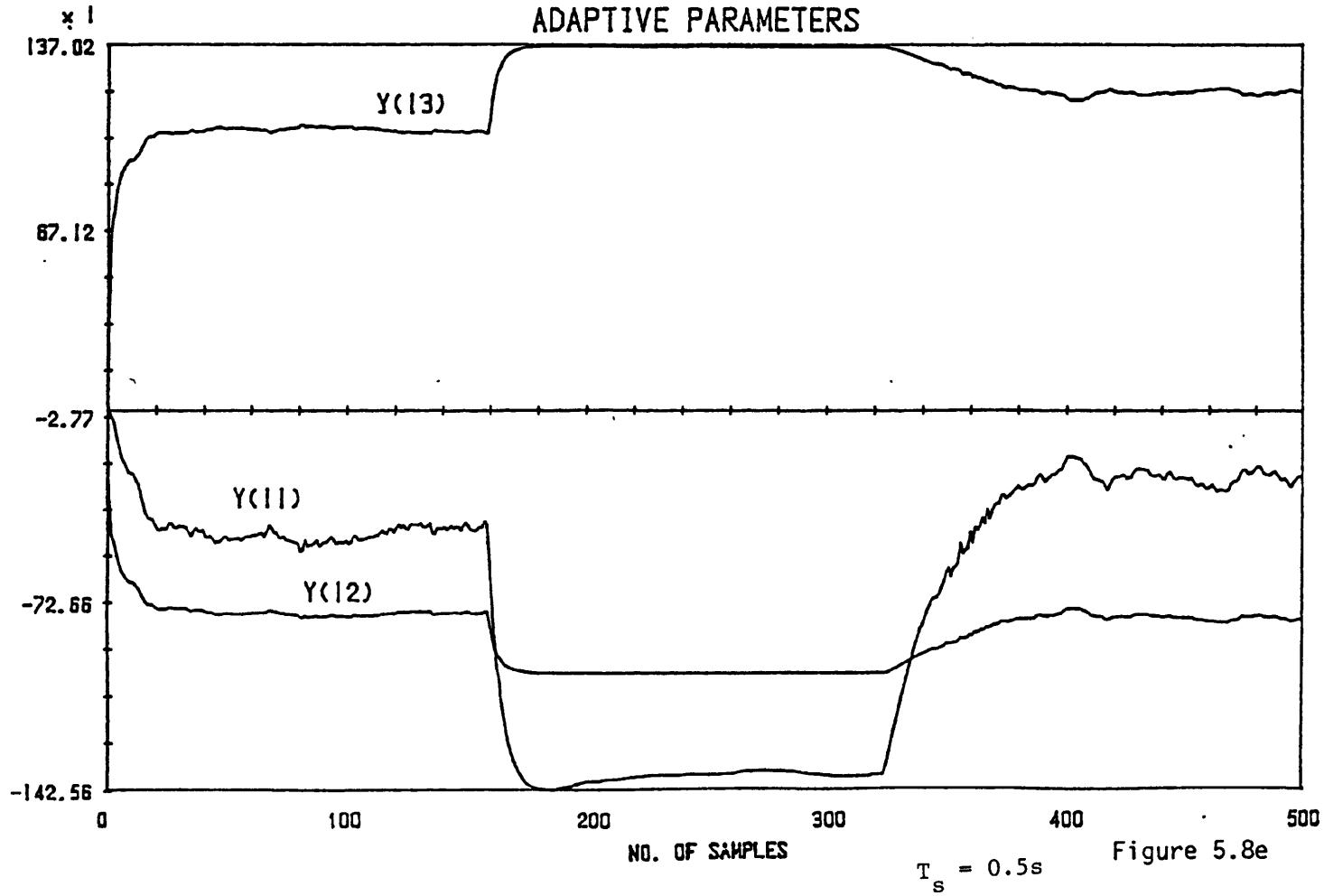
Figure 5.8 (a),(b),(d) show the reference, system outputs and control inputs for different on-line set point changes using the Goodwin et al algorithm.

(c),(e) The adaptive parameters $\theta(t)$ for Figures 5.8b,d respectively.



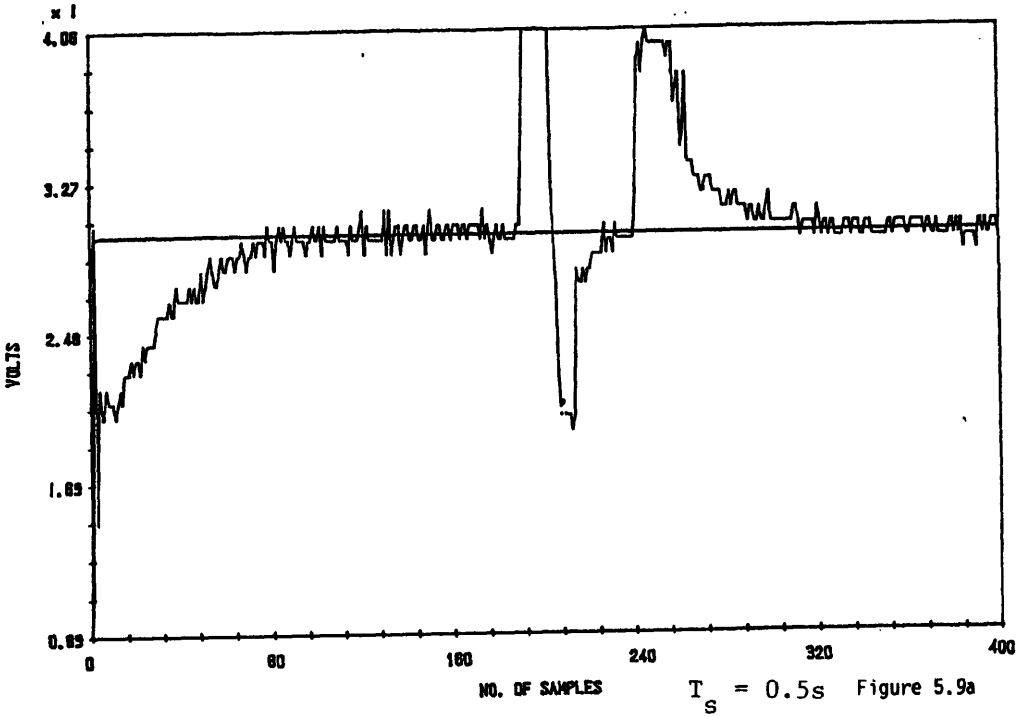
T_s = 0.5s

Figure 5.8d



T_s = 0.5s

Figure 5.8e



T_s = 0.5s Figure 5.9a

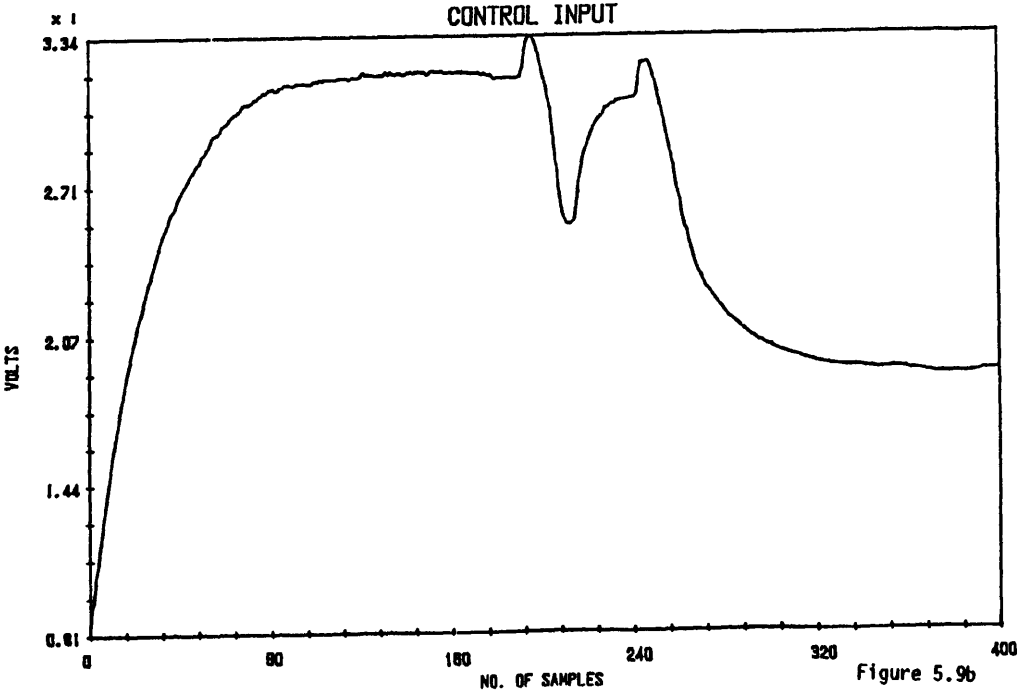


Figure 5.9b

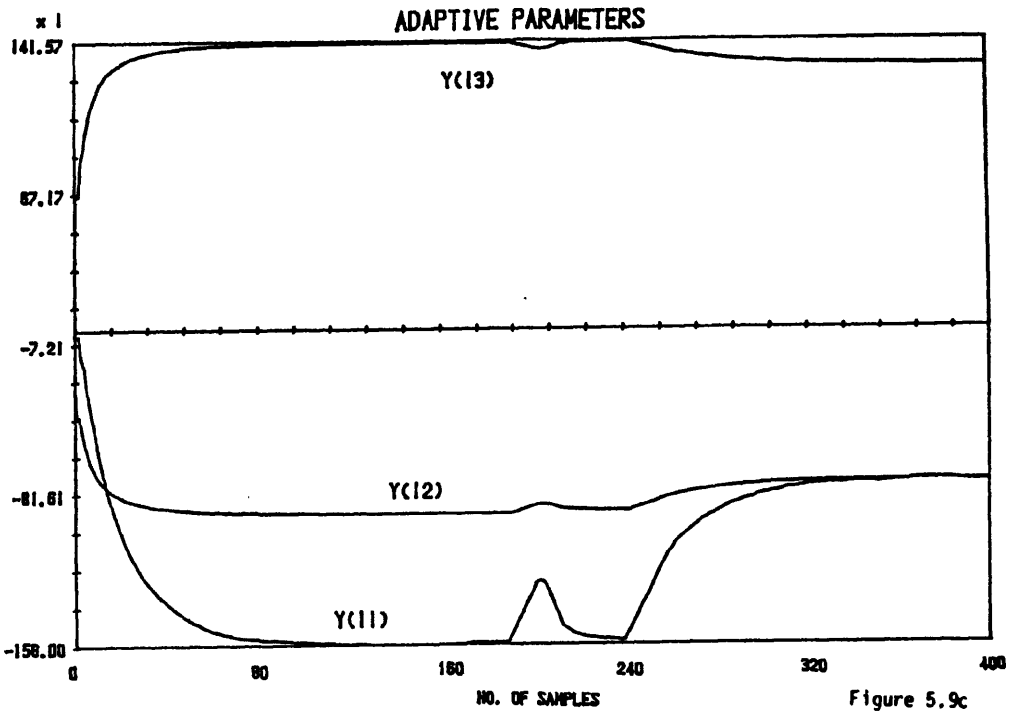


Figure 5.9c

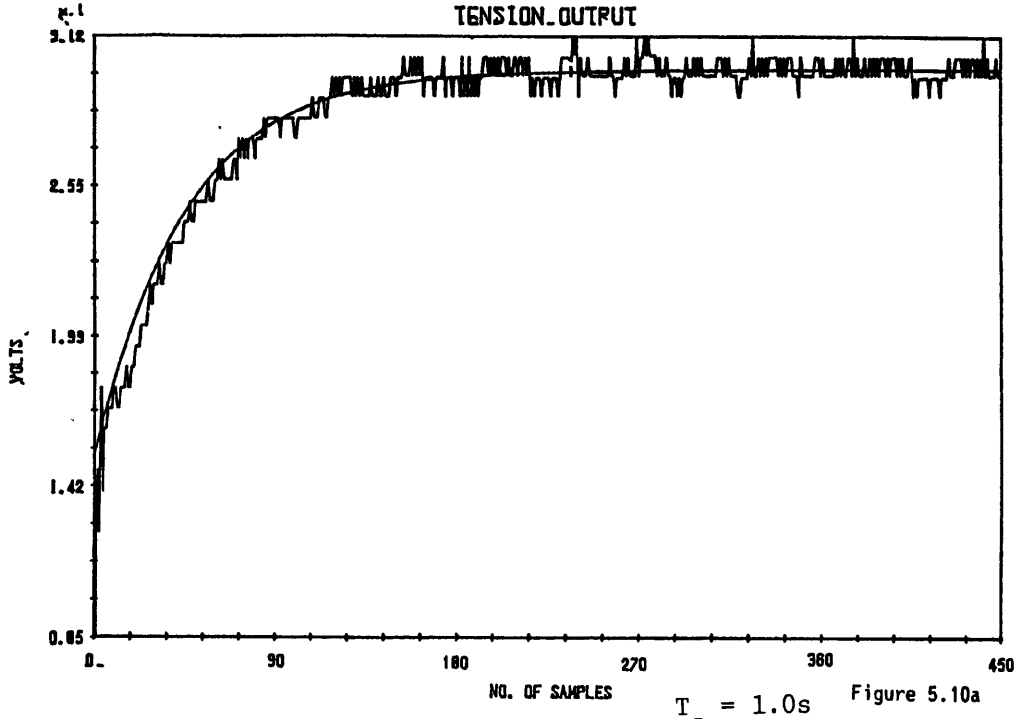
T_s = 0.5s

Figure 5.9 (a) The reference and system outputs for the Goodwin et al algorithm with external disturbances introduced.

(b) The control input u(t).

(c) The adaptive parameters θ(t).

T_s = 0.5s



T_s = 1.0s Figure 5.10a

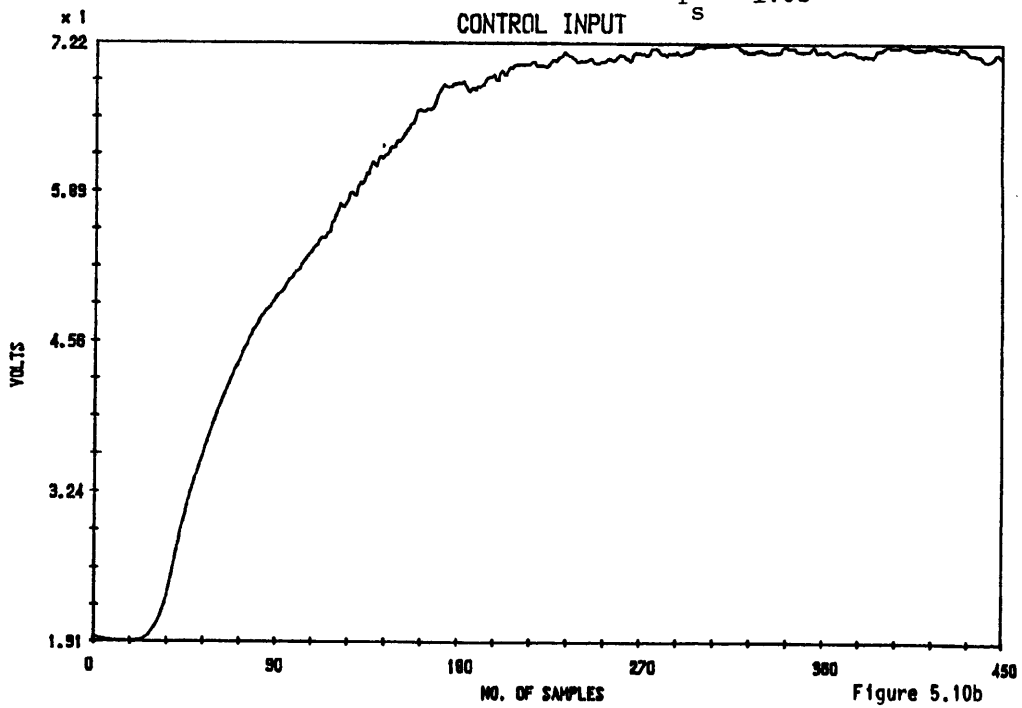


Figure 5.10b

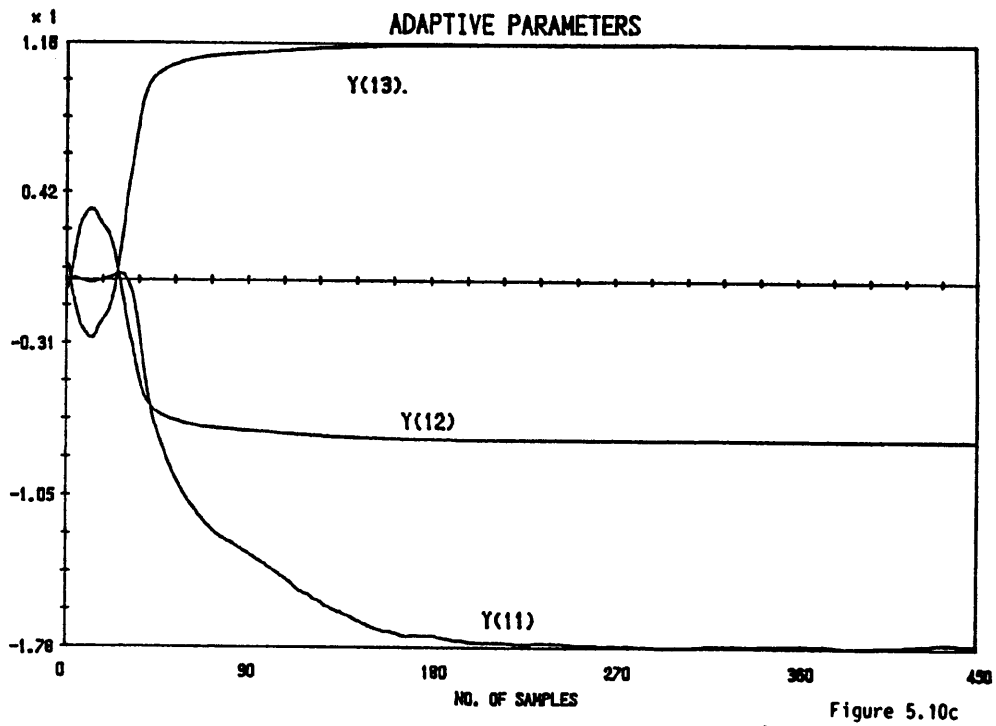


Figure 5.10c

T_s = 1.0s

Figure 5.10 (a) The reference and system outputs for a slower first-order reference model using the Goodwin et al algorithm.

(b) The control input u(t).

(c) The adaptive parameters θ(t).

T_s = 1.0s

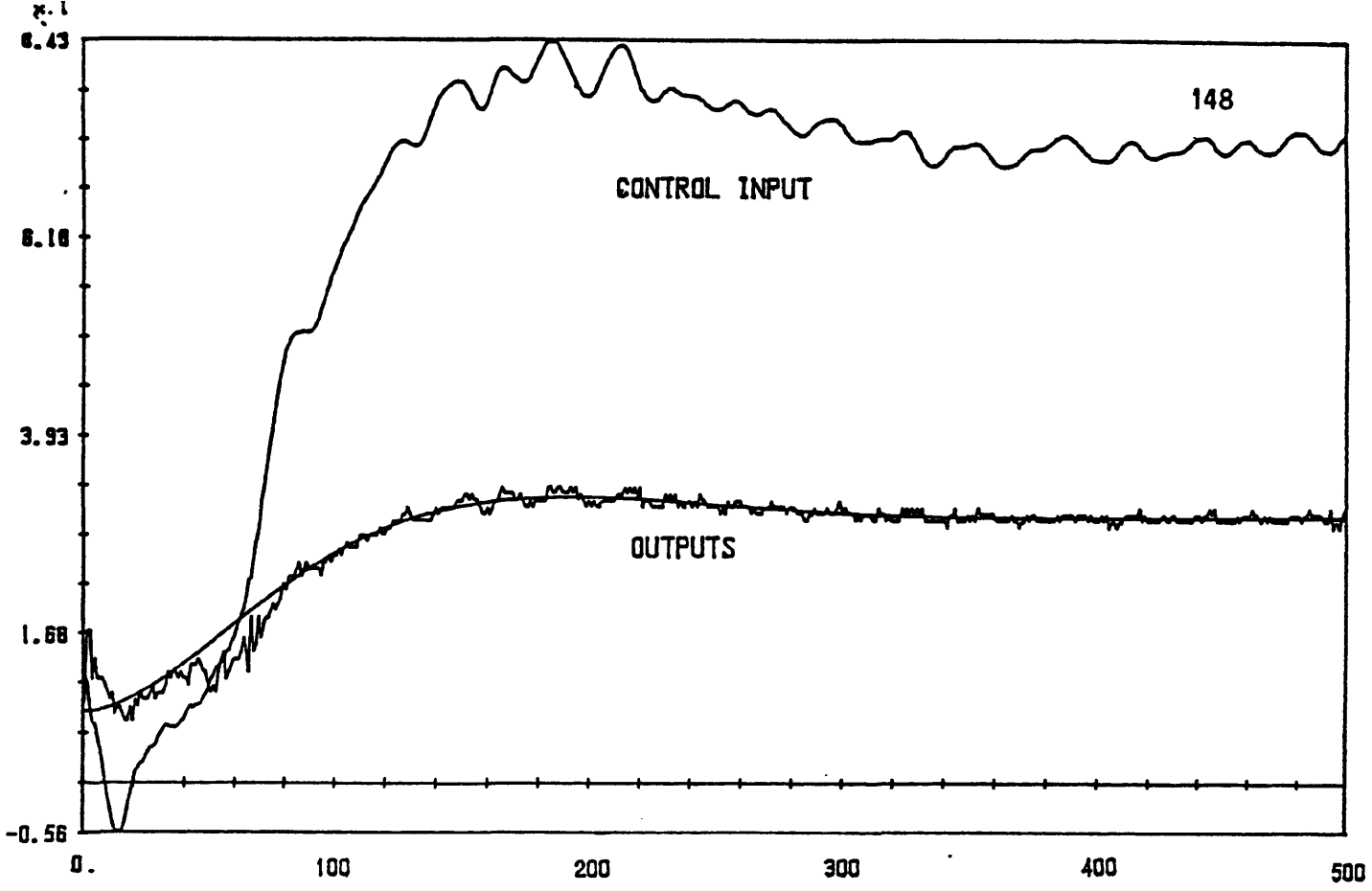


Figure 5.11a

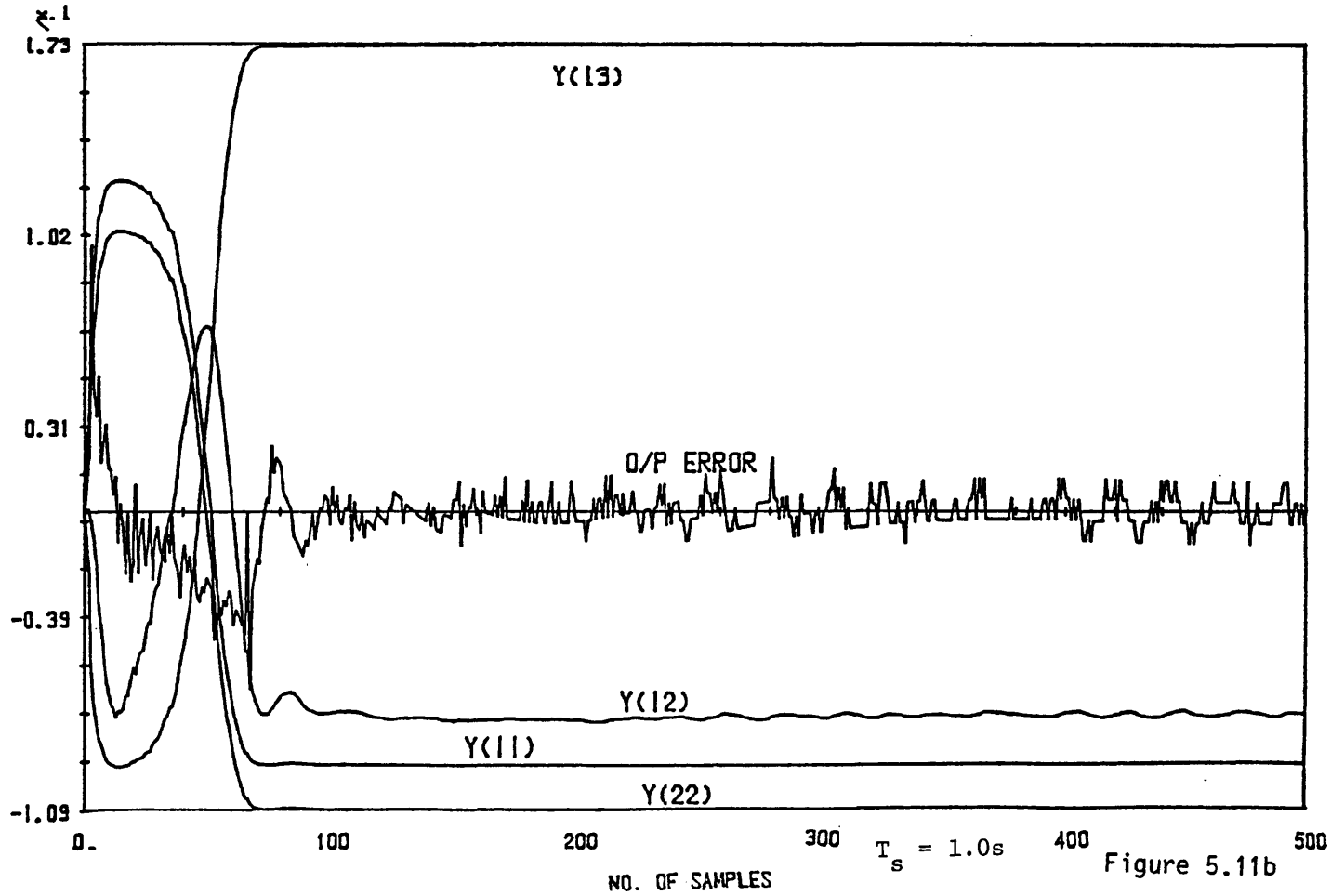


Figure 5.11b

Figure 5.11 (a) The reference and system outputs plus control input for a second-order reference model using the Goodwin et al algorithm.
(b) Showing some of the adaptive parameters $\theta(t)$ and the output error $e(t)$.

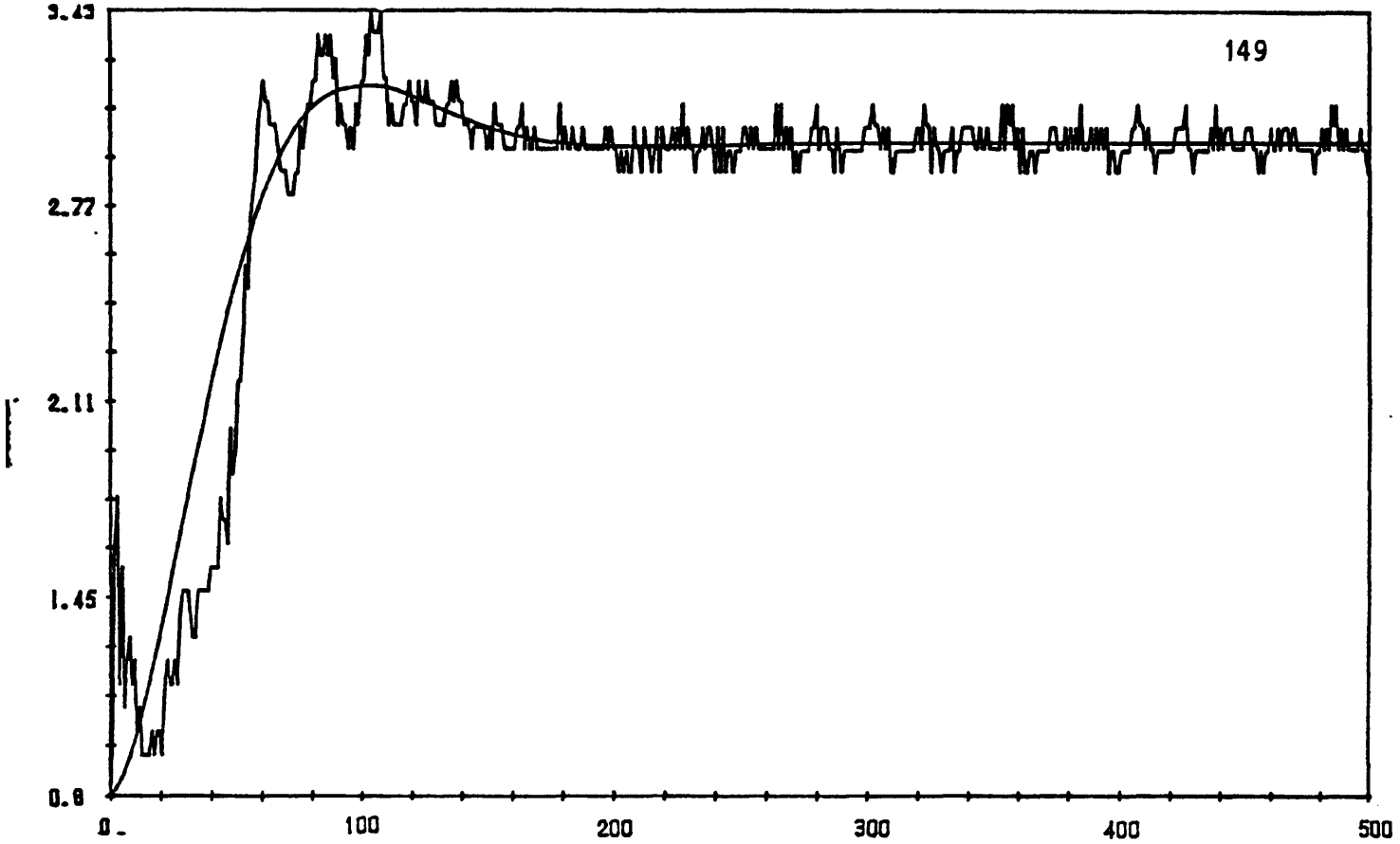


Figure 5.11c

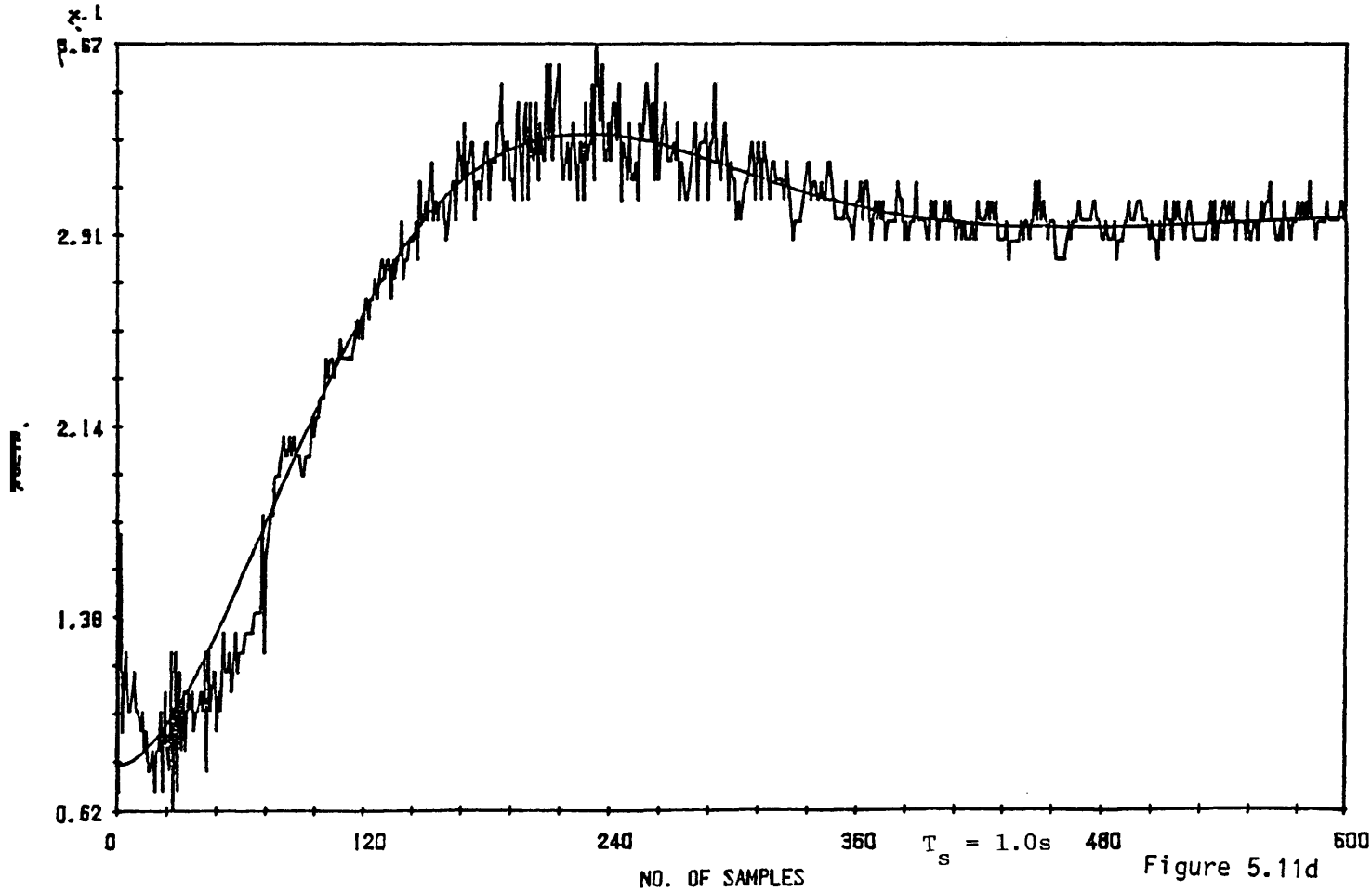
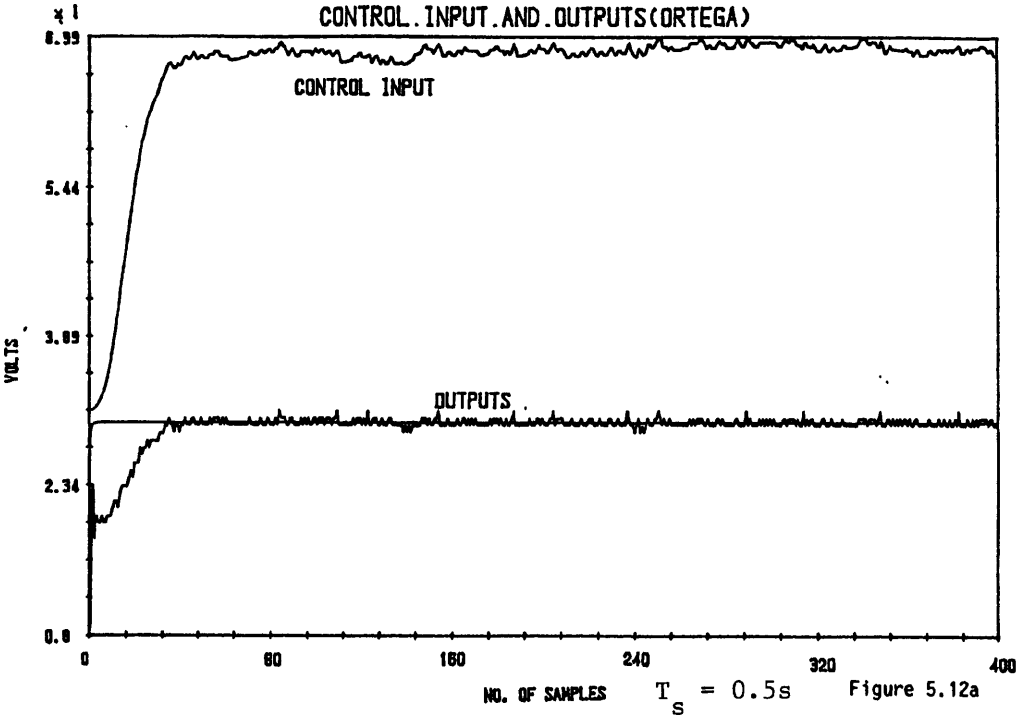


Figure 5.11d

Figure 5.11 (c),(d) The reference and system outputs for two other second-order reference models.



T_s = 0.5s Figure 5.12a

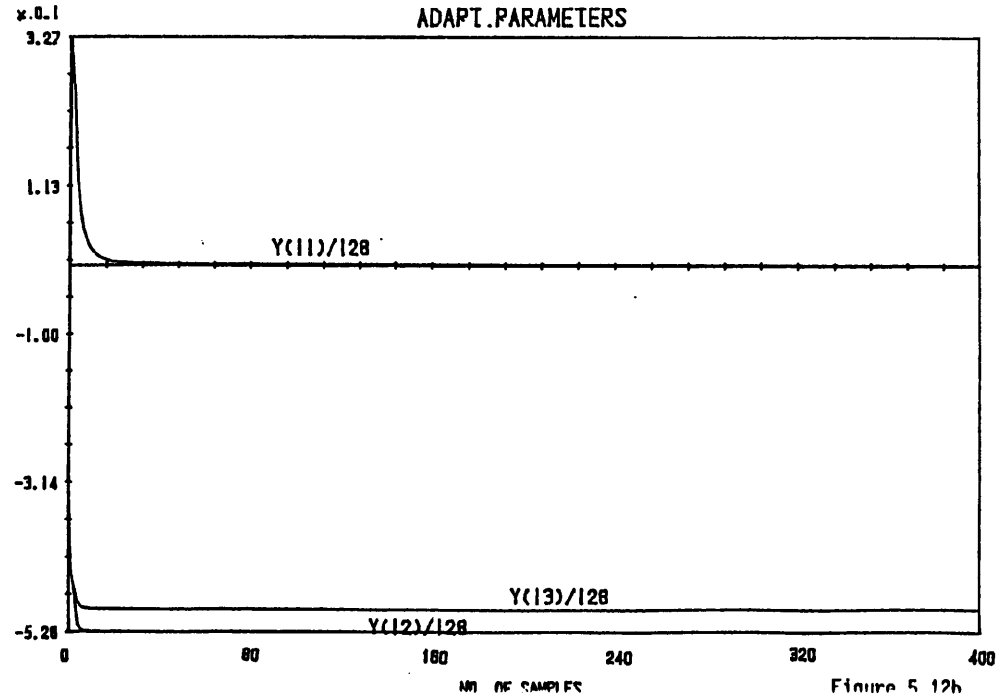


Figure 5.12b

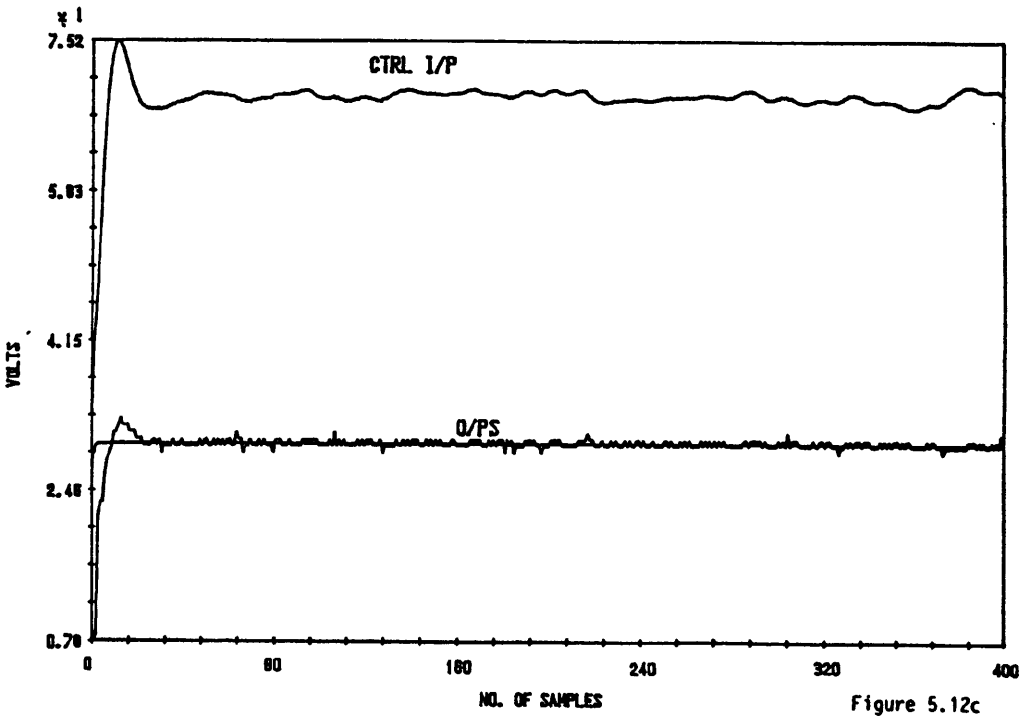


Figure 5.12c

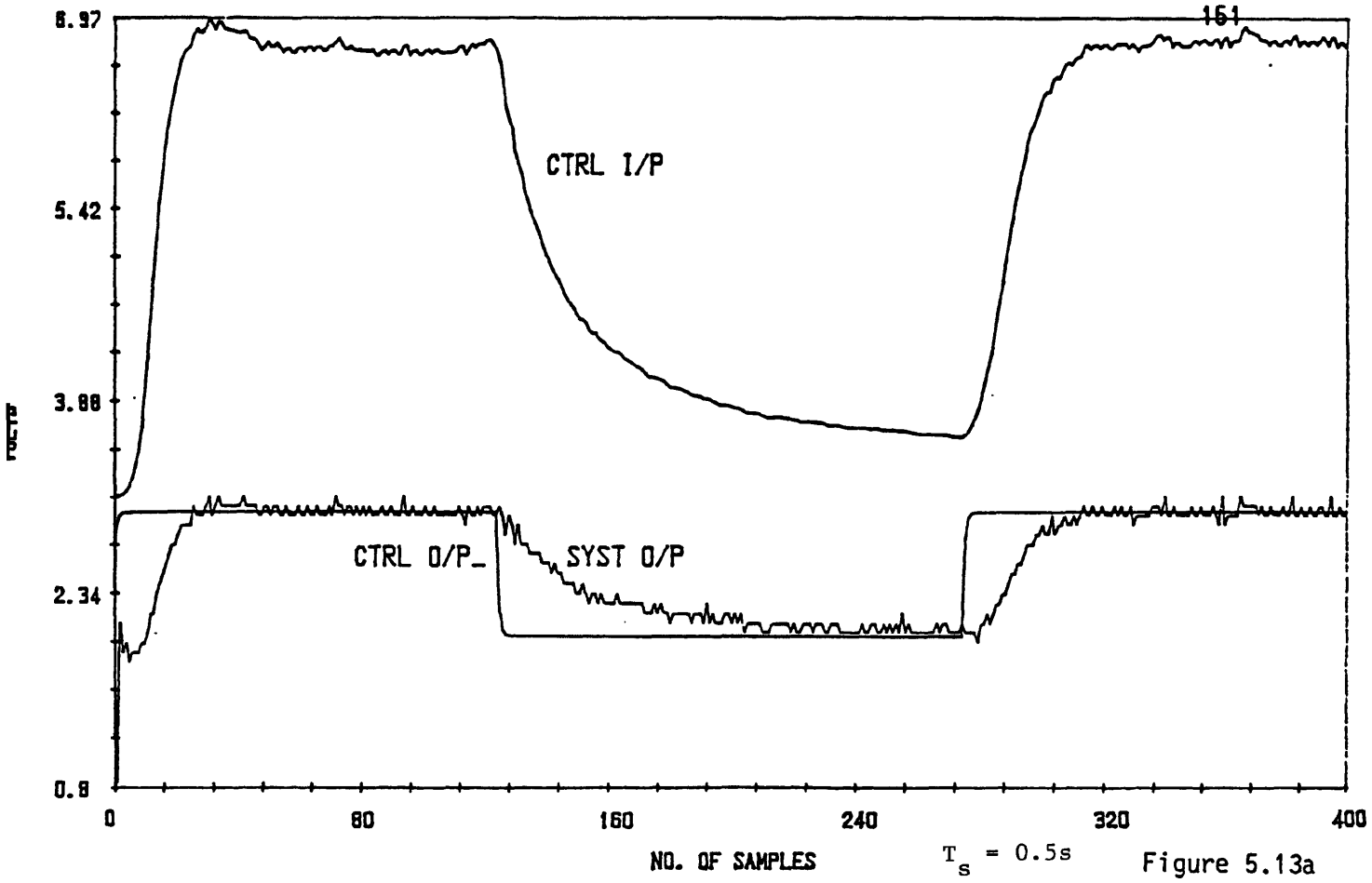
T_s = 0.5s

Figure 5.12 (a) The reference, system outputs and control input for a first-order reference model using the Ortega et al algorithm.

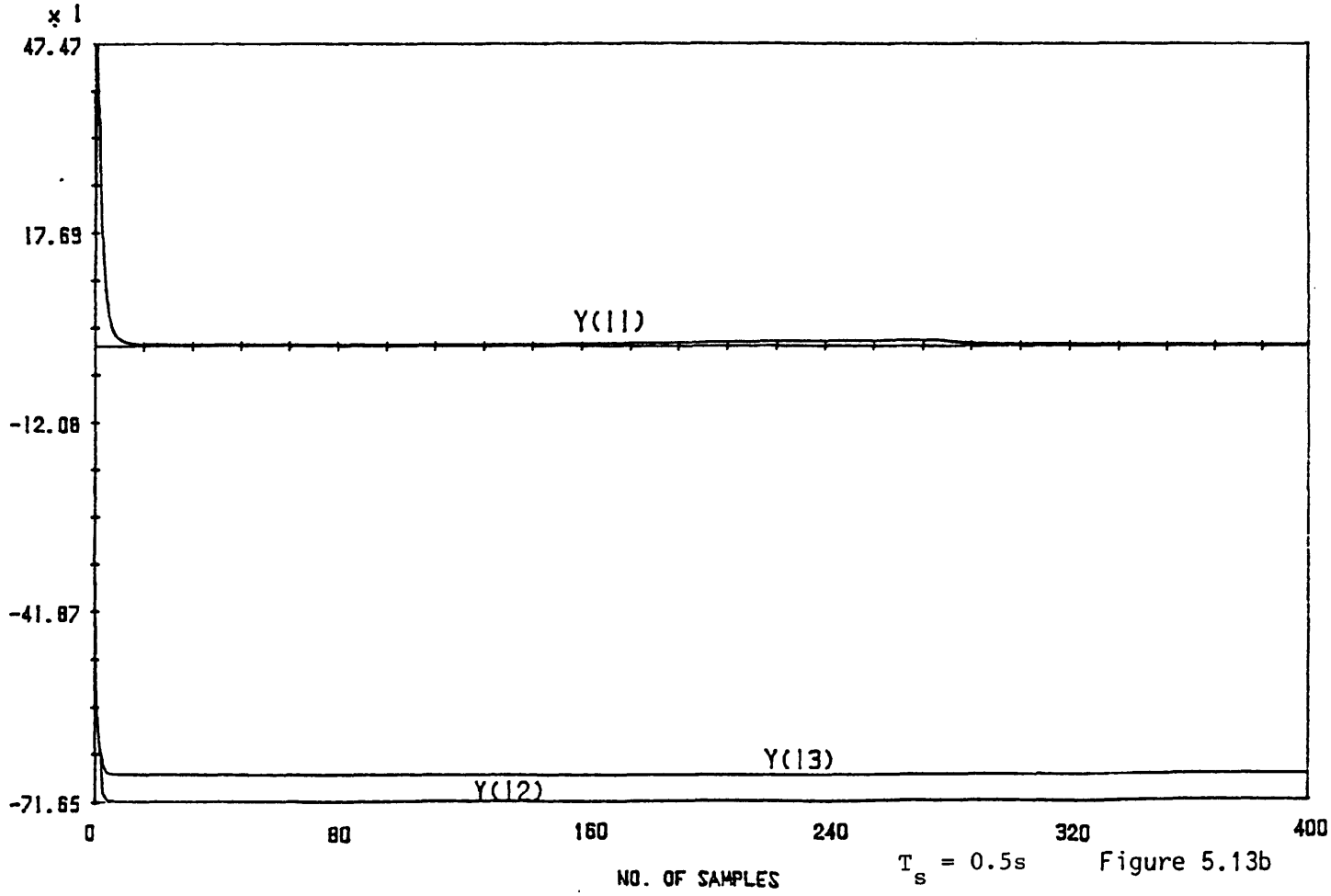
(b) The adaptive parameters $\theta(t)$.

(c) The reference, system outputs plus the control input for a bigger adaptive gain.

T_s = 0.5s



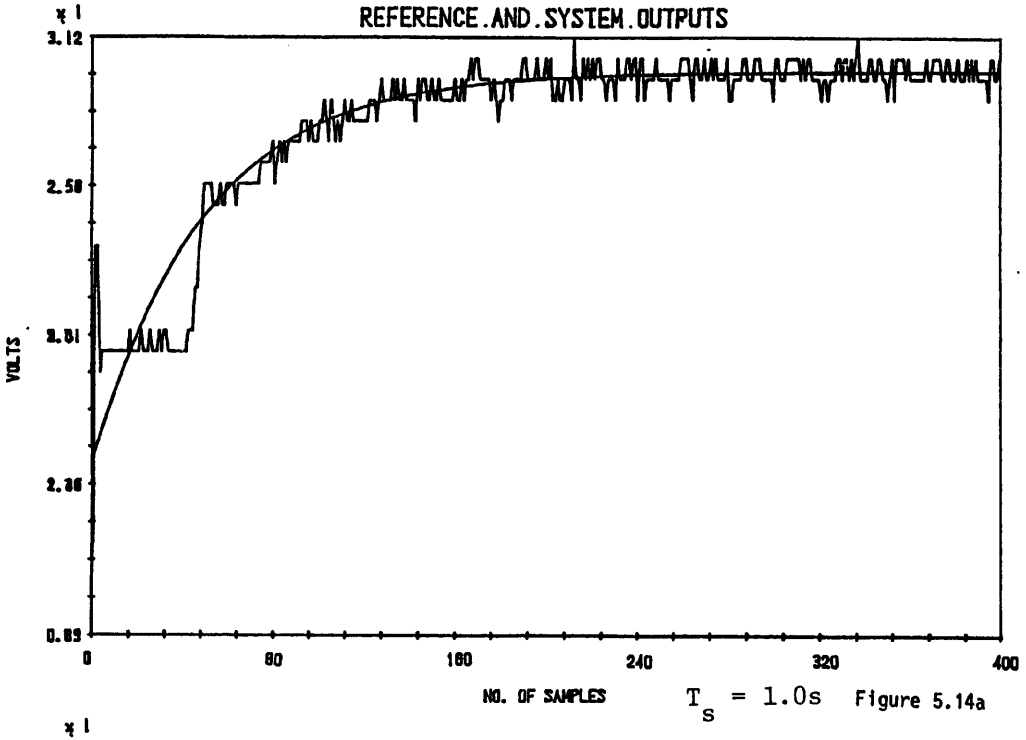
T_s = 0.5s Figure 5.13a



T_s = 0.5s Figure 5.13b

Figure 5.13 (a) The reference, system outputs plus control input for on-line set-point changes using the Ortega et al algorithm.

(b) The adaptive parameters $\theta(t)$.



T_s = 1.0s Figure 5.14a

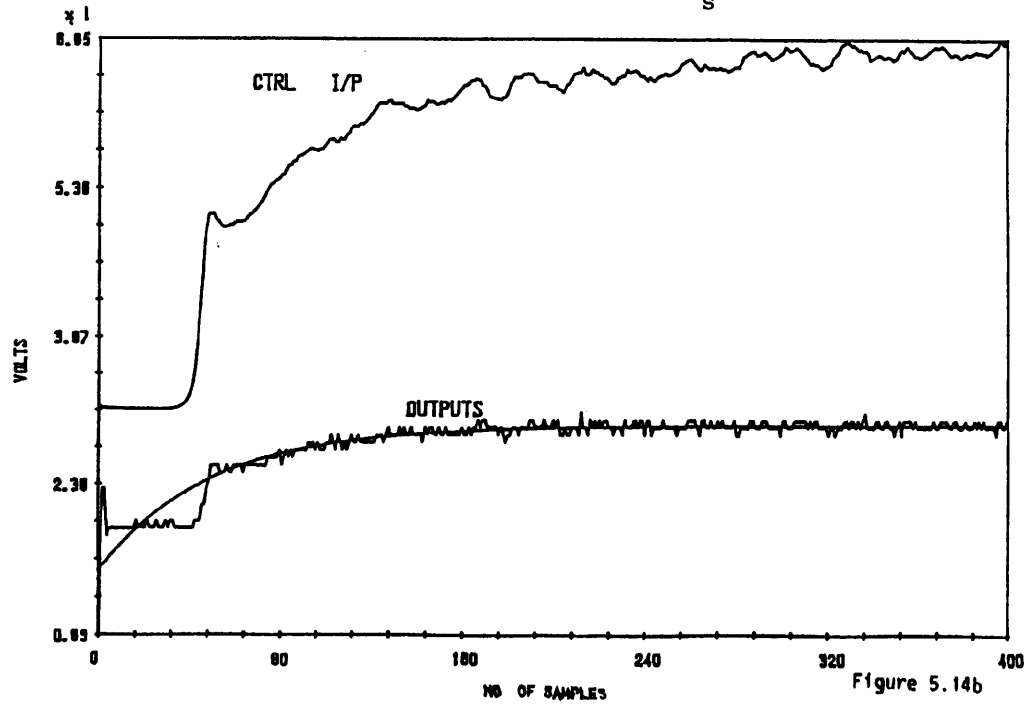


Figure 5.14b

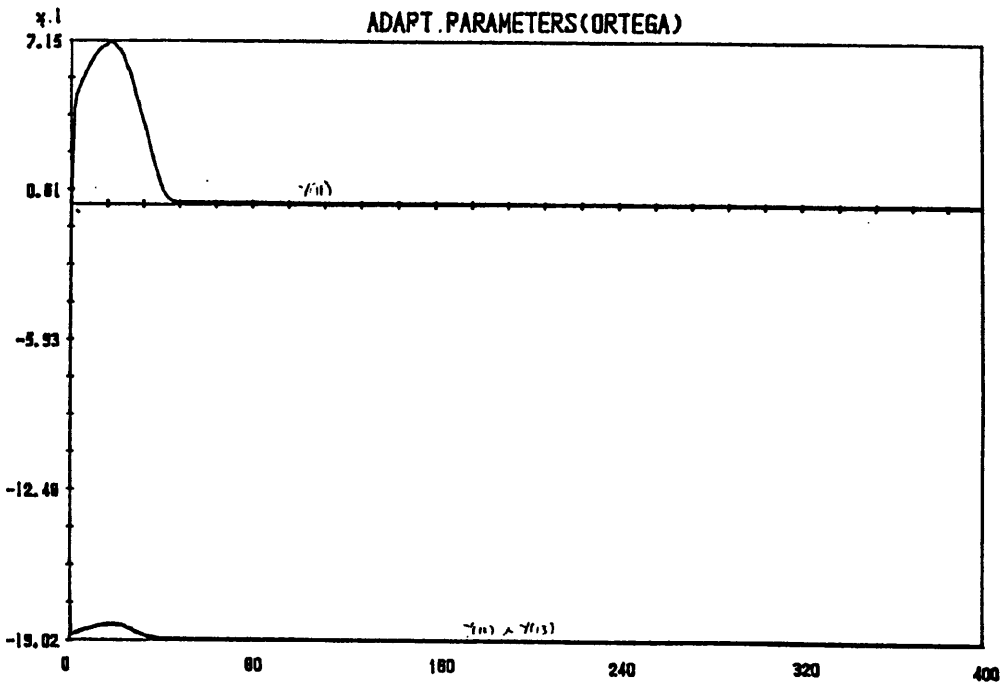


Figure 5.14c

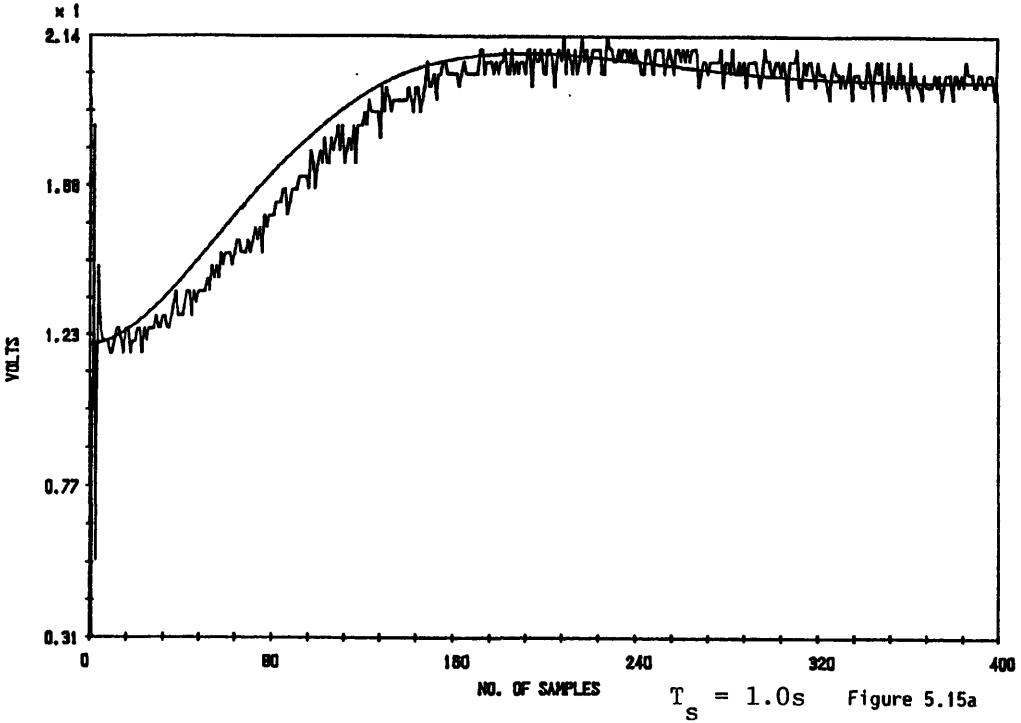
T_s = 1.0s

Figure 5.14 (a) The reference and system outputs for a slower first-order reference model using the Ortega et al algorithm.

(b) The control input u(t) and the outputs.

(c) The adaptive parameters θ(t).

T_s = 1.0s



T_s = 1.0s Figure 5.15a

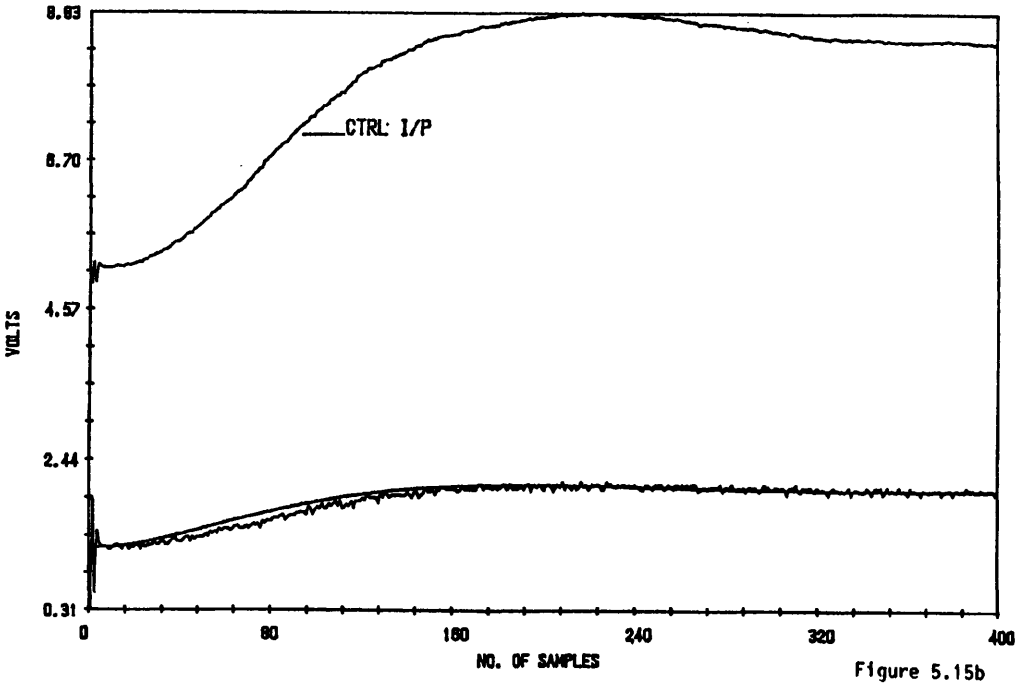
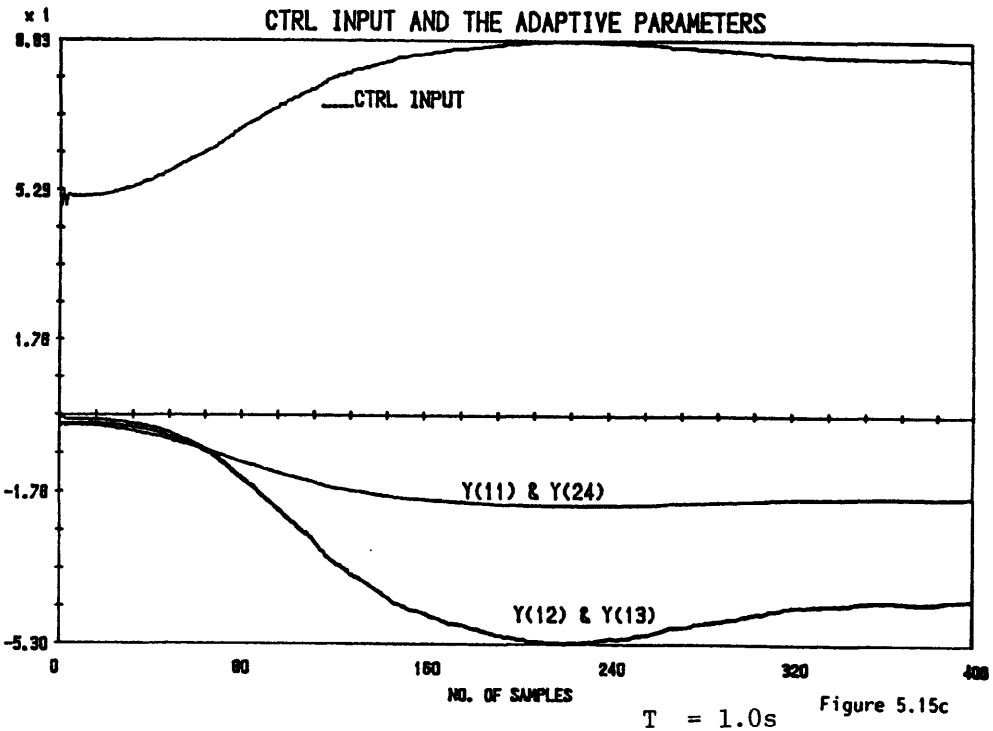


Figure 5.15b



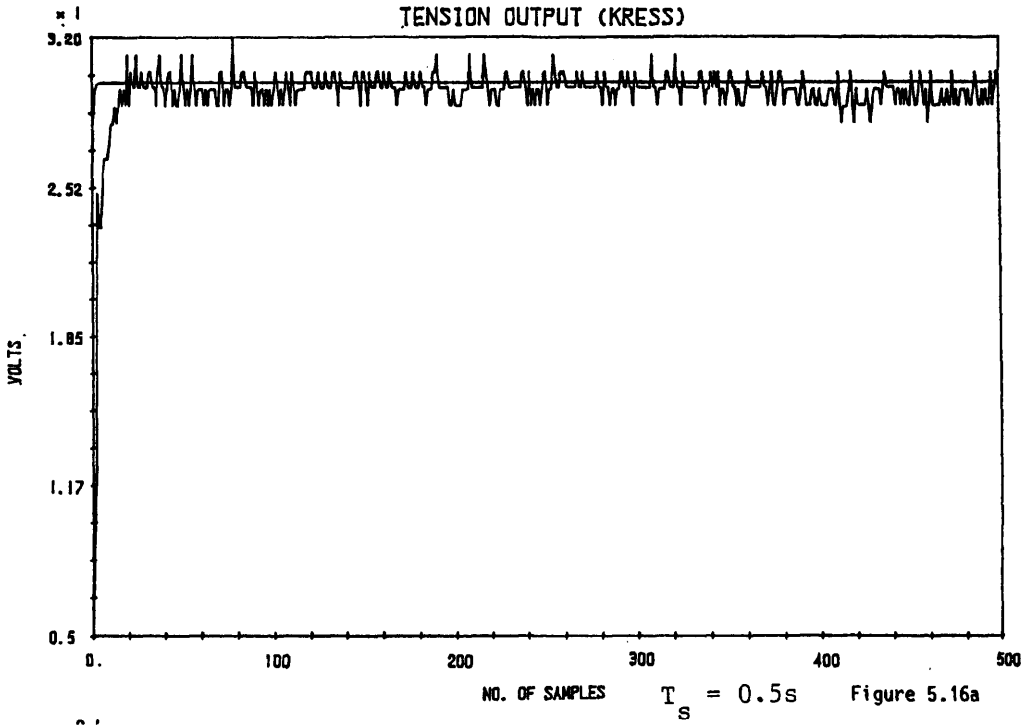
T_s = 1.0s Figure 5.15c

Figure 5.15 (a) The reference and system outputs for a second-order reference model using the Ortega et al algorithm.

(b) The control input u(t) and the reference, system outputs.

(c) Shows some of the adaptive parameters and the control input.

T_s = 1.0s



T_s = 0.5s Figure 5.16a

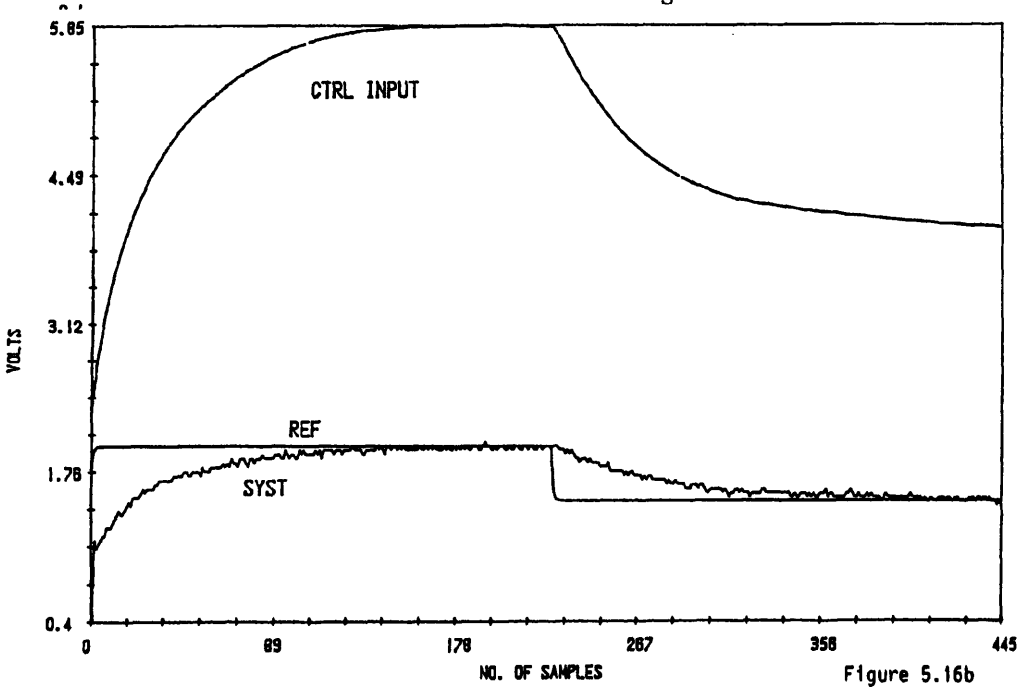


Figure 5.16b

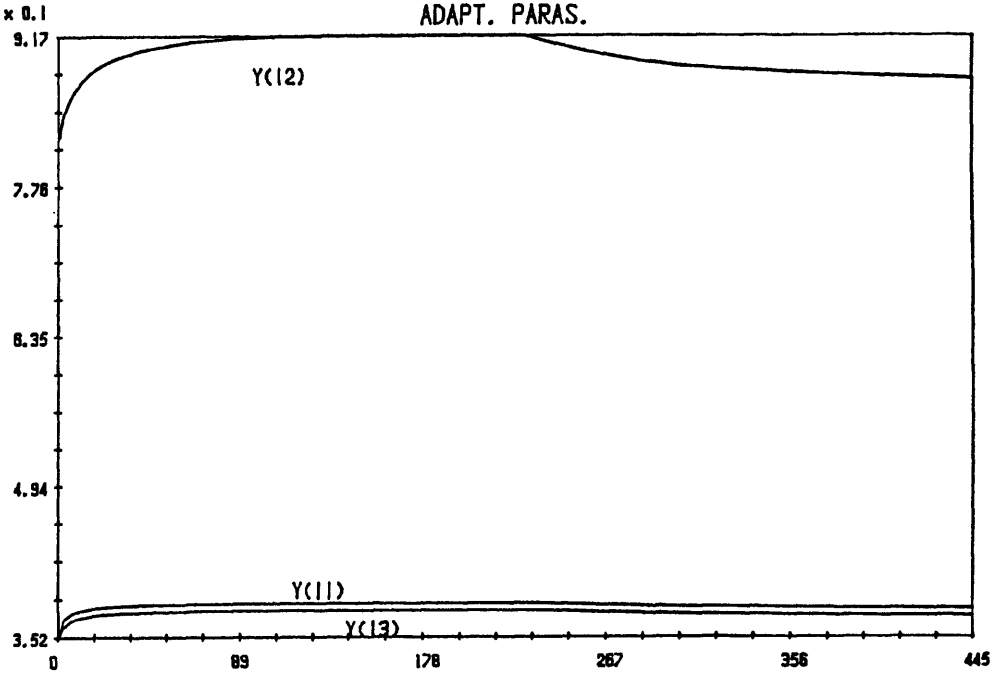


Figure 5.16c

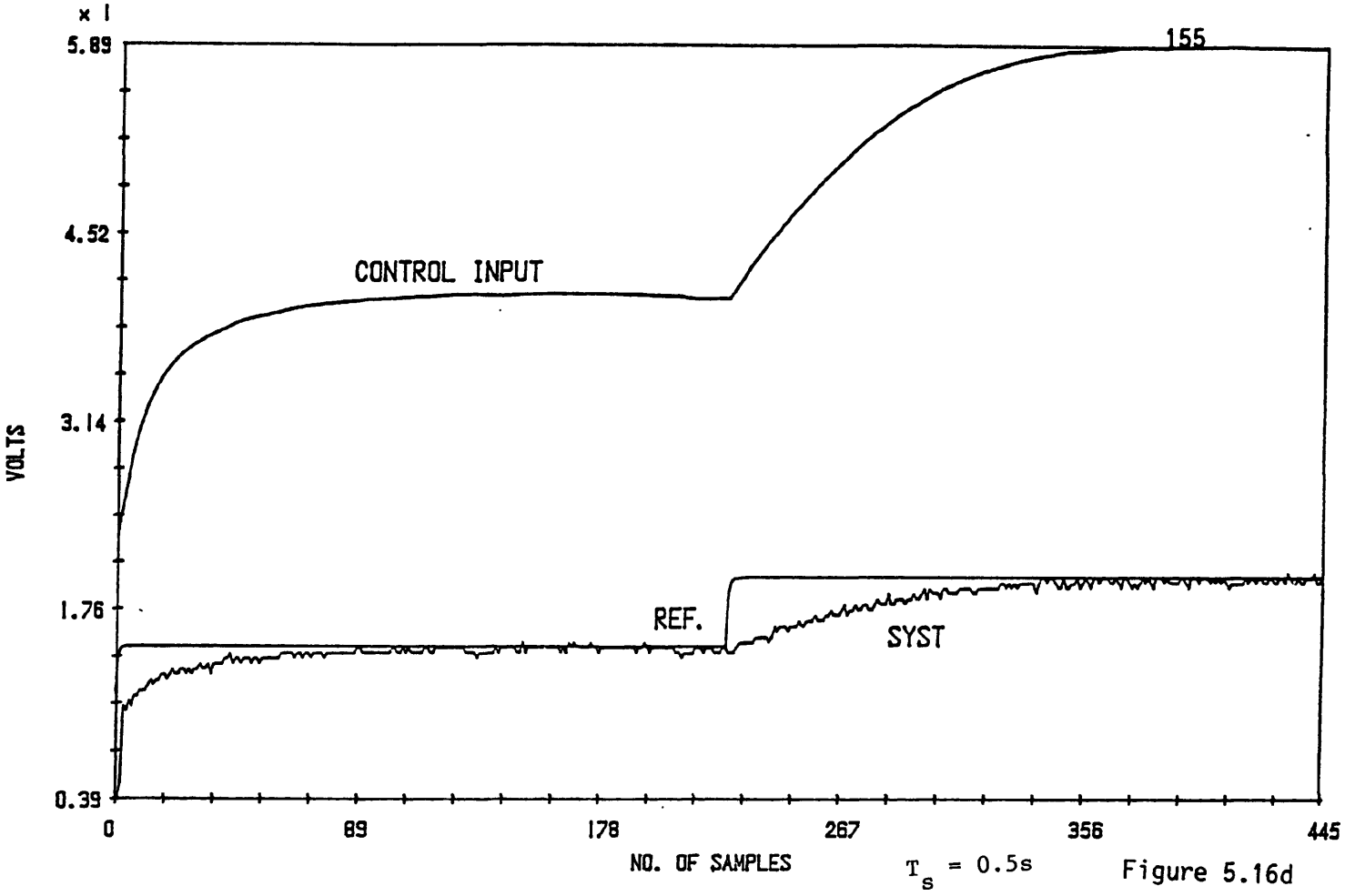
T_s = 0.5s

Figure 5.16 (a) The reference and system outputs for a first-order model using the Kreisselmeier-Anderson algorithm.

(b),(d) Shows the reference, system outputs and control inputs for different on-line set-point changesusing the Kreisselmeier-Anderson algorithm.

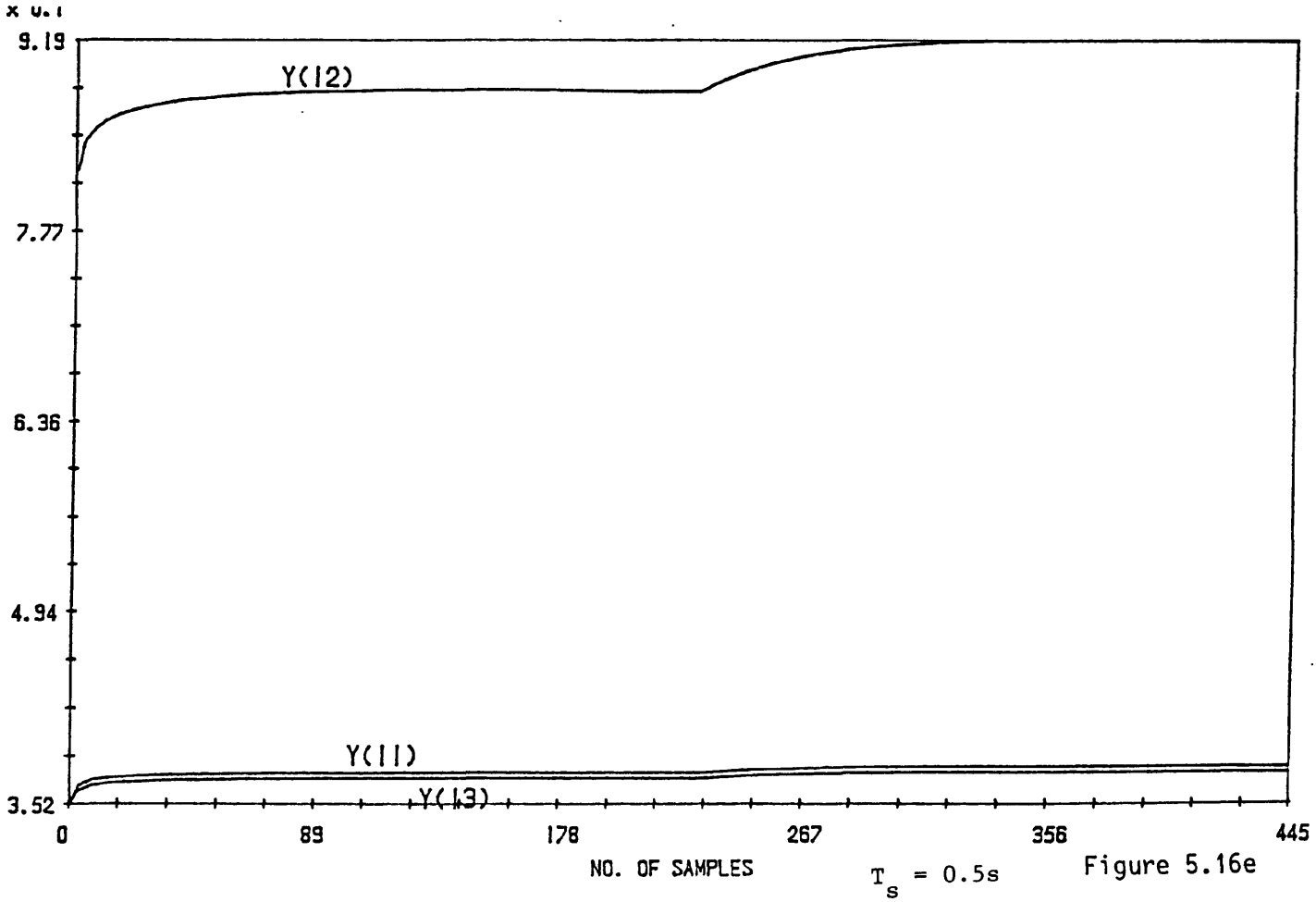
(c),(e) The adaptive parameters $\theta(t)$ for Figures 5.16 (b),(d).

T_s = 0.5s



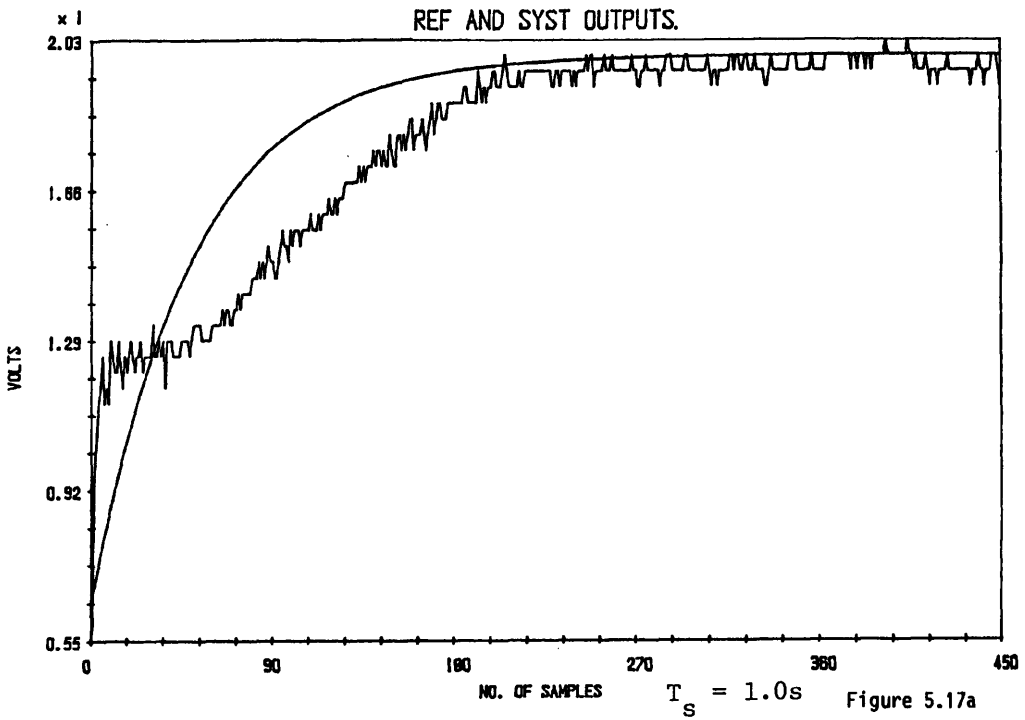
T_s = 0.5s

Figure 5.16d



T_s = 0.5s

Figure 5.16e



T_s = 1.0s

Figure 5.17a

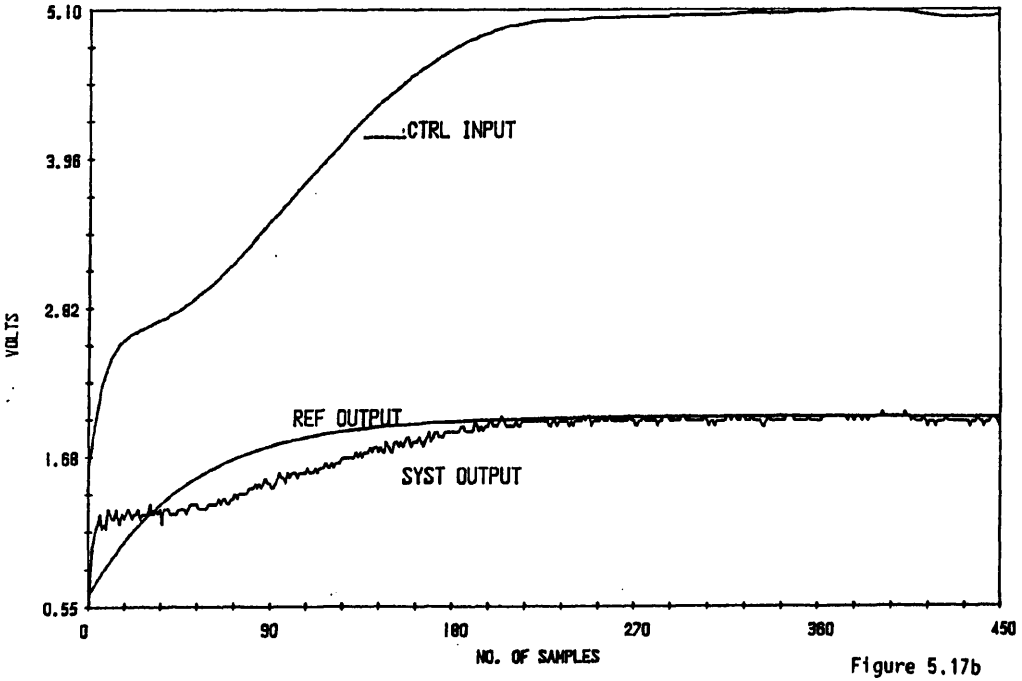
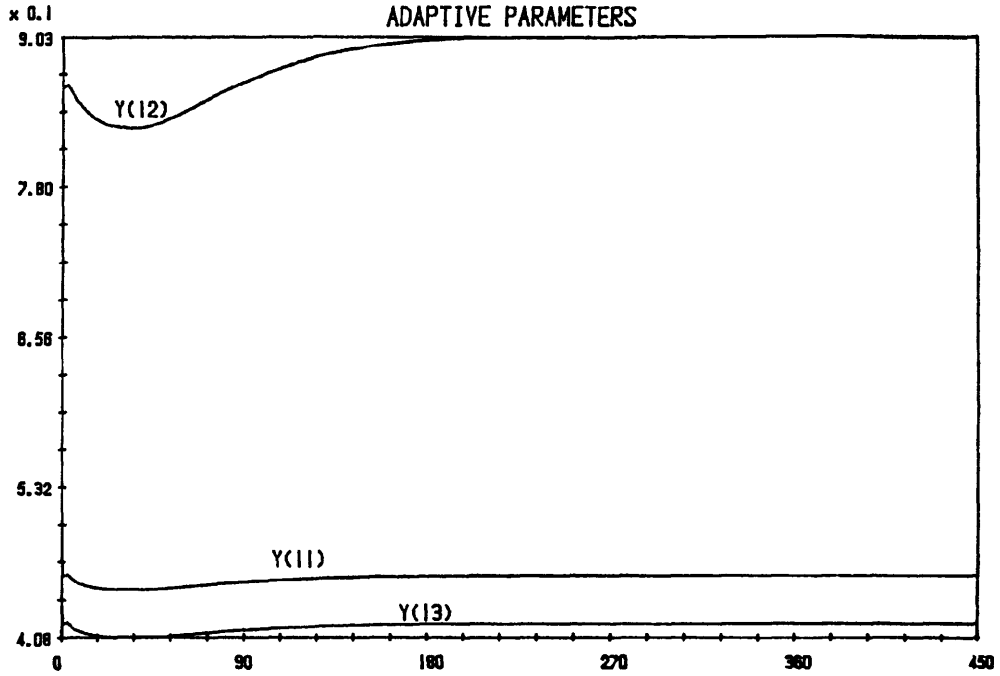


Figure 5.17b



T_s = 1.0s

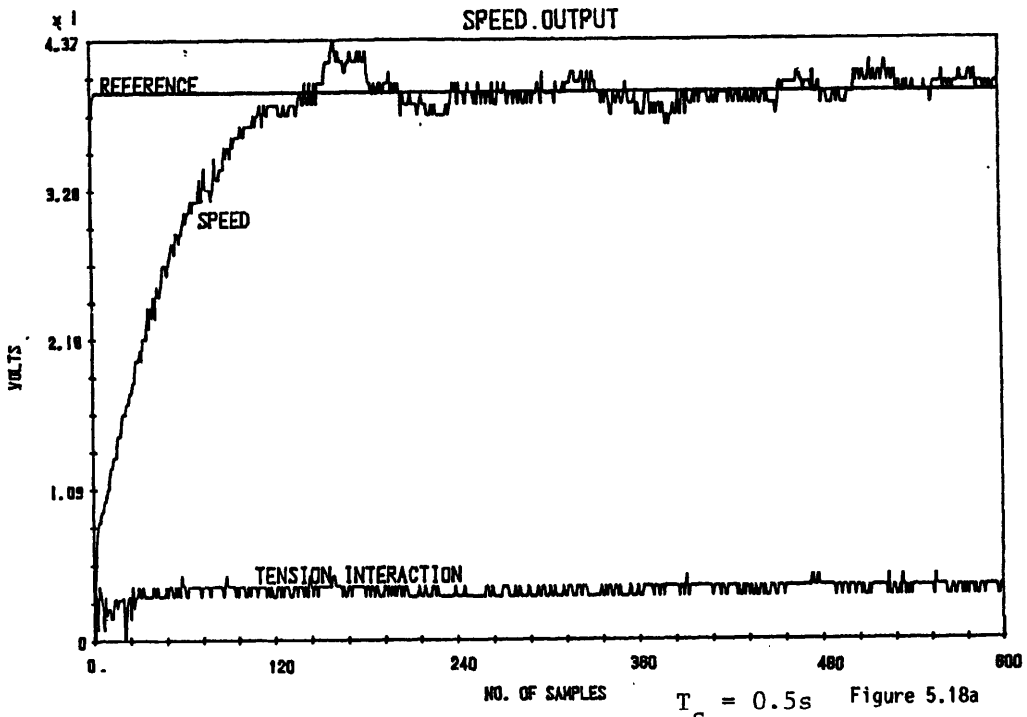
Figure 5.17c

Figure 5.17 (a) The reference and system outputs for a slow first-order model using the Kreisselmeier-Anderson algorithm.

(b) The control input u(t) and the reference, system outputs.

(c) The adaptive parameters $\theta(t)$.

T_s = 1.0s



T_s = 0.5s Figure 5.18a

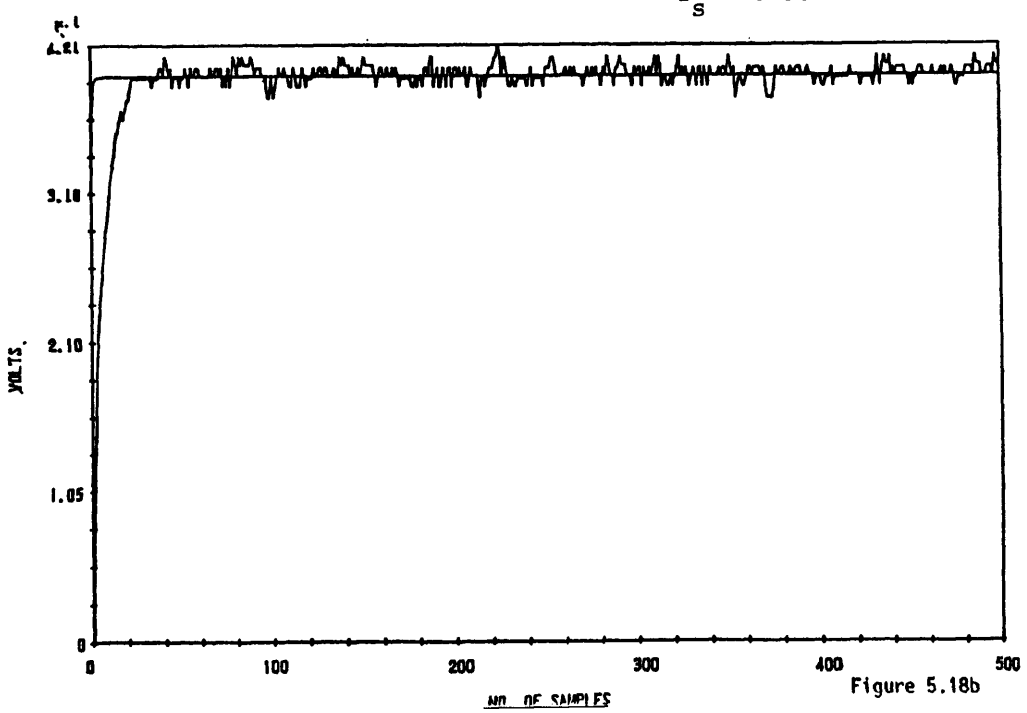


Figure 5.18b

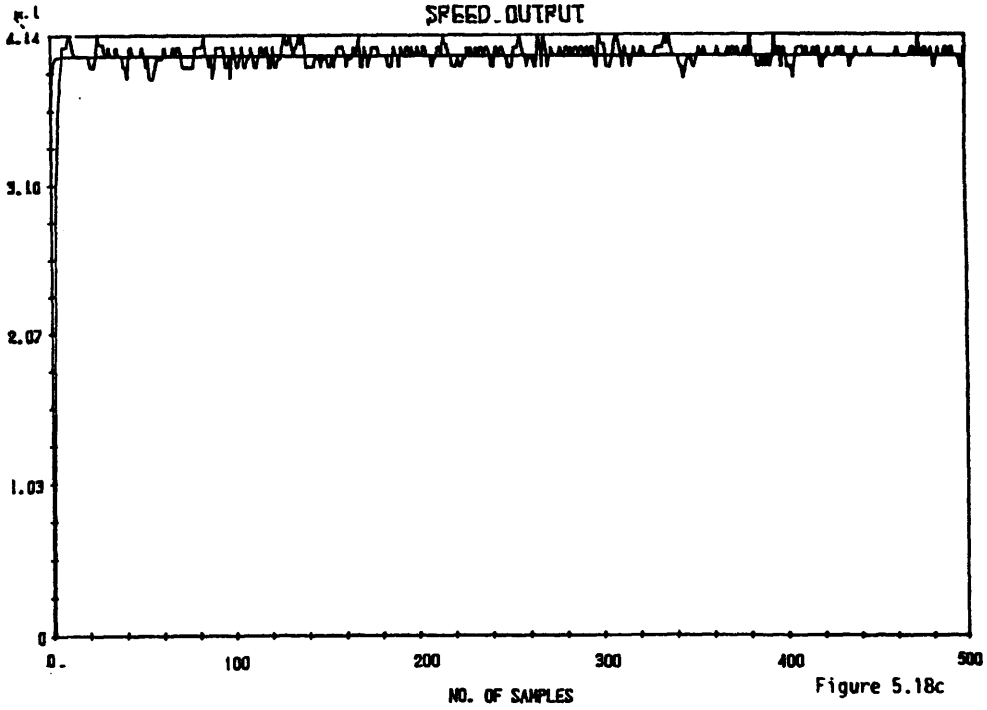
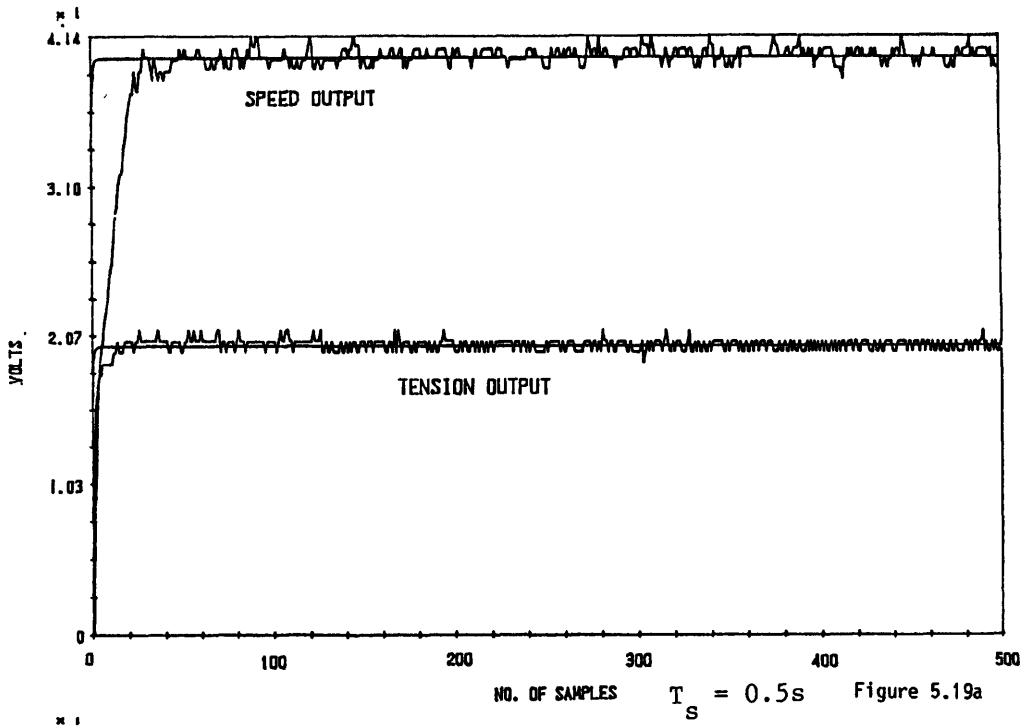


Figure 5.18c

T_s = 0.5s

Figure 5.18 a,b,c Shows the reference and system outputs for speed control of the coupled electric drives for different adaptive gains using the Goodwin et al algorithm.

T_s = 0.5s



T_s = 0.5s Figure 5.19a

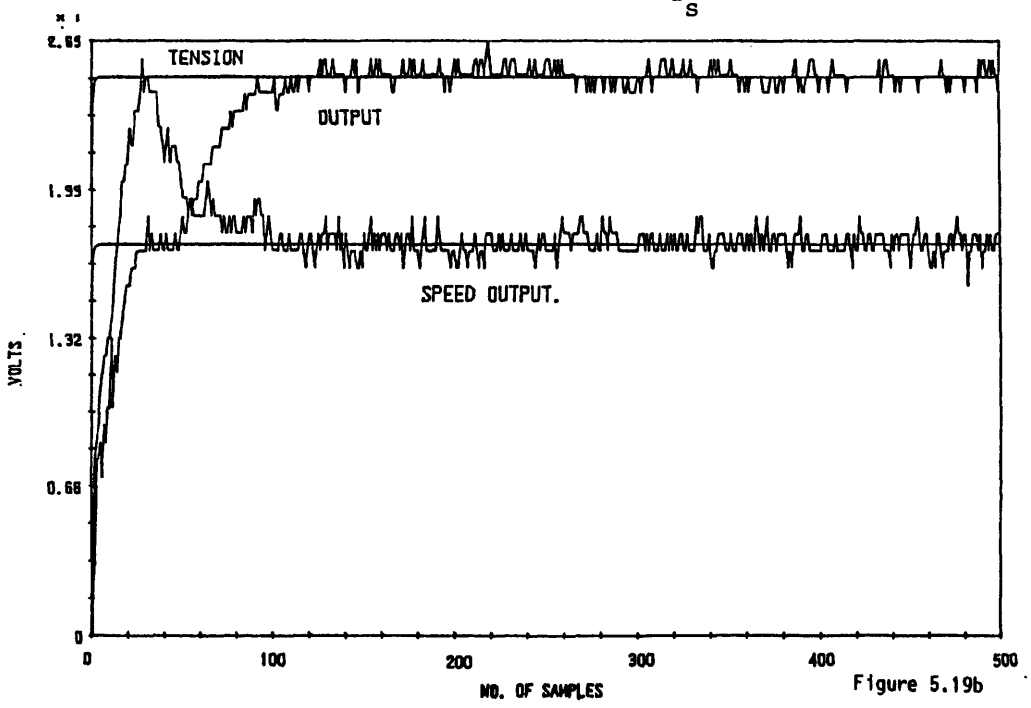


Figure 5.19b

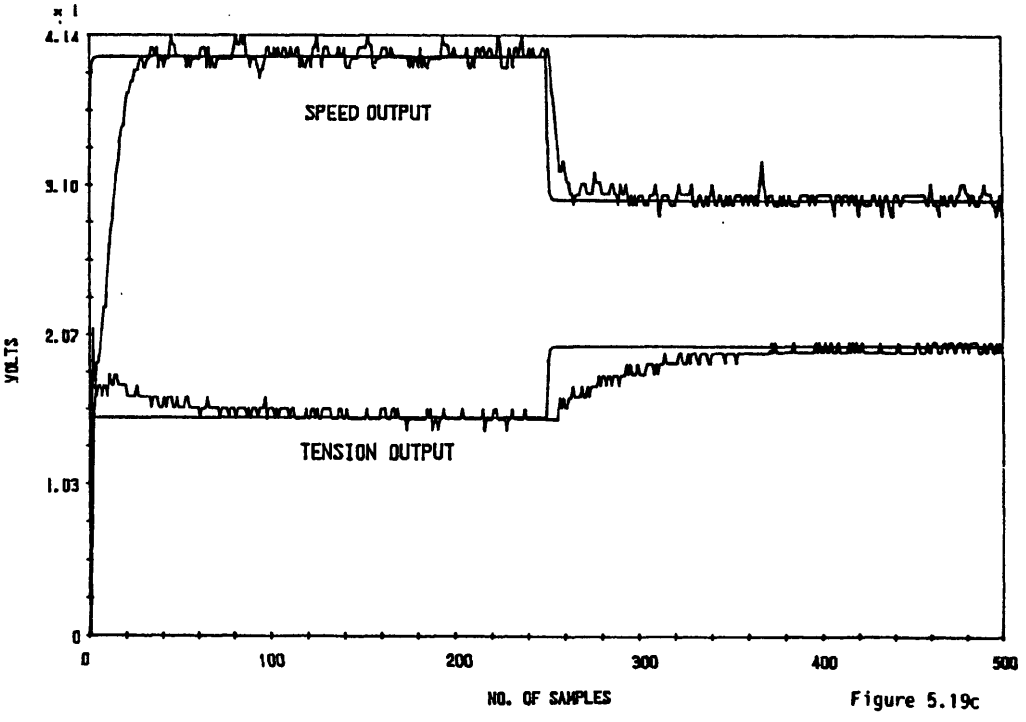


Figure 5.19c

T_s = 0.5s

Figure 5.19 (a),(b) Shows the multivariable control of the drives using different set-points by applying the MIMO Goodwin et al algorithm.

(c) Shows the tension and speed control outputs for changes in the reference set points using the MIMO Goodwin et al algorithm.

T_s = 0.5s

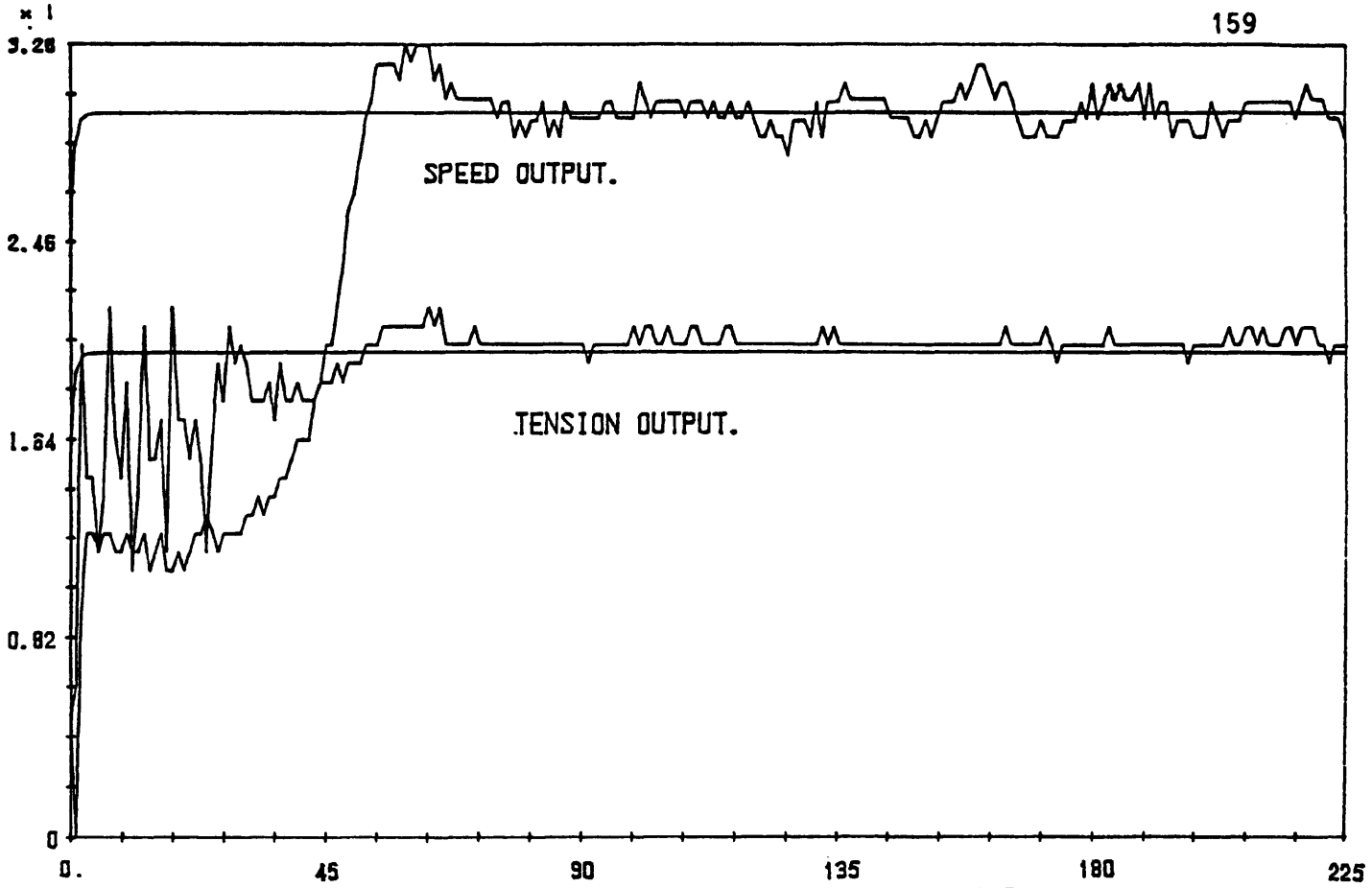


Figure 5.20a

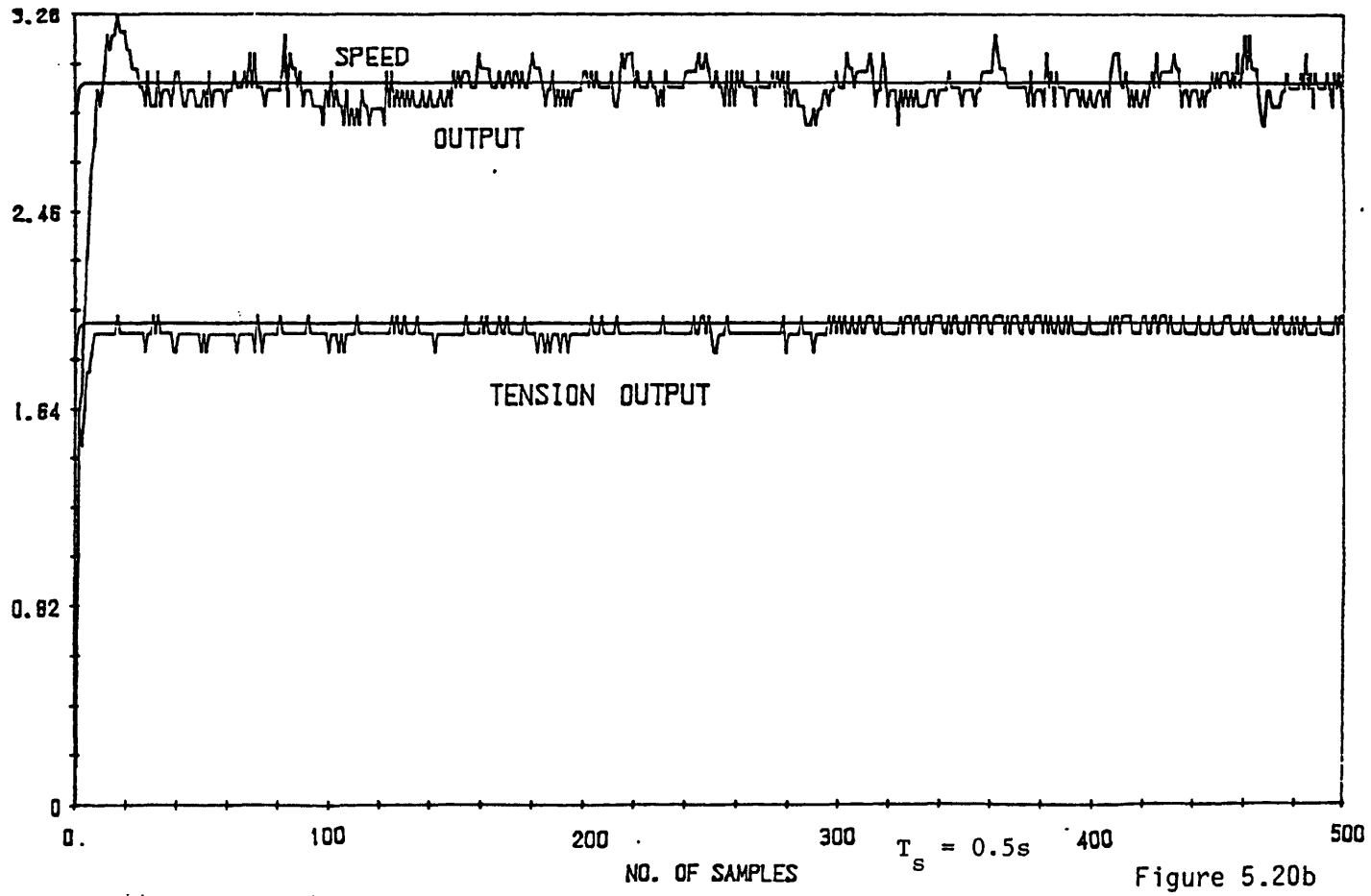


Figure 5.20b

Figure 5.20 (a) Shows the multivariable control of the drives using the derived MIMO Kreisselmeier-Anderson algorithm without parameter bounds.
(b) Shows the multivariable control of the drives using the derived MIMO Kreisselmeier-Anderson algorithm with parameter bounds.

Chapter 6

THE HEATER BAR

6.1 INTRODUCTION

The heater bar is a scaled down model of an industrial problem, a process where molten plastic passes through cylindrical heater tubes which were supposed to be kept at a constant temperature. Due to the slowness of the process it was quite difficult to control the heater temperature at a constant enough value for the molten plastic, as the temperature was found to be either too hot or too cold for the process.

Another use could be made of the laboratory rig by considering it as a boiler in power/steam stations which needs to operate at a certain constant temperature for steam and hence power generation.

It is a distributed parameter system, but which for the course of this work would be assumed to be finite dimensional lumped parameter system, hence with a deterministic transfer function for ease of control.

6.2 SYSTEM DESCRIPTION

The rig (see Figure 6.1) consists of a half mild steel cylinder cone inside a cylindrical sleeve of the same material. The heater, a metal coil, is inserted within the core of the steel bar. Basically, heat is transferred round the system via conduction, radiation and convectional means. The rig is made up of two halves, one of which contains the heater while the second half consists of the part whose temperature needs to be controlled. As shown in Figure 6.1, there are eight, 2.5 cm deep holes along the bar, six of which are on the second half while two are on the first half. Since the second half represents the system output, the temperature of which is to be controlled,

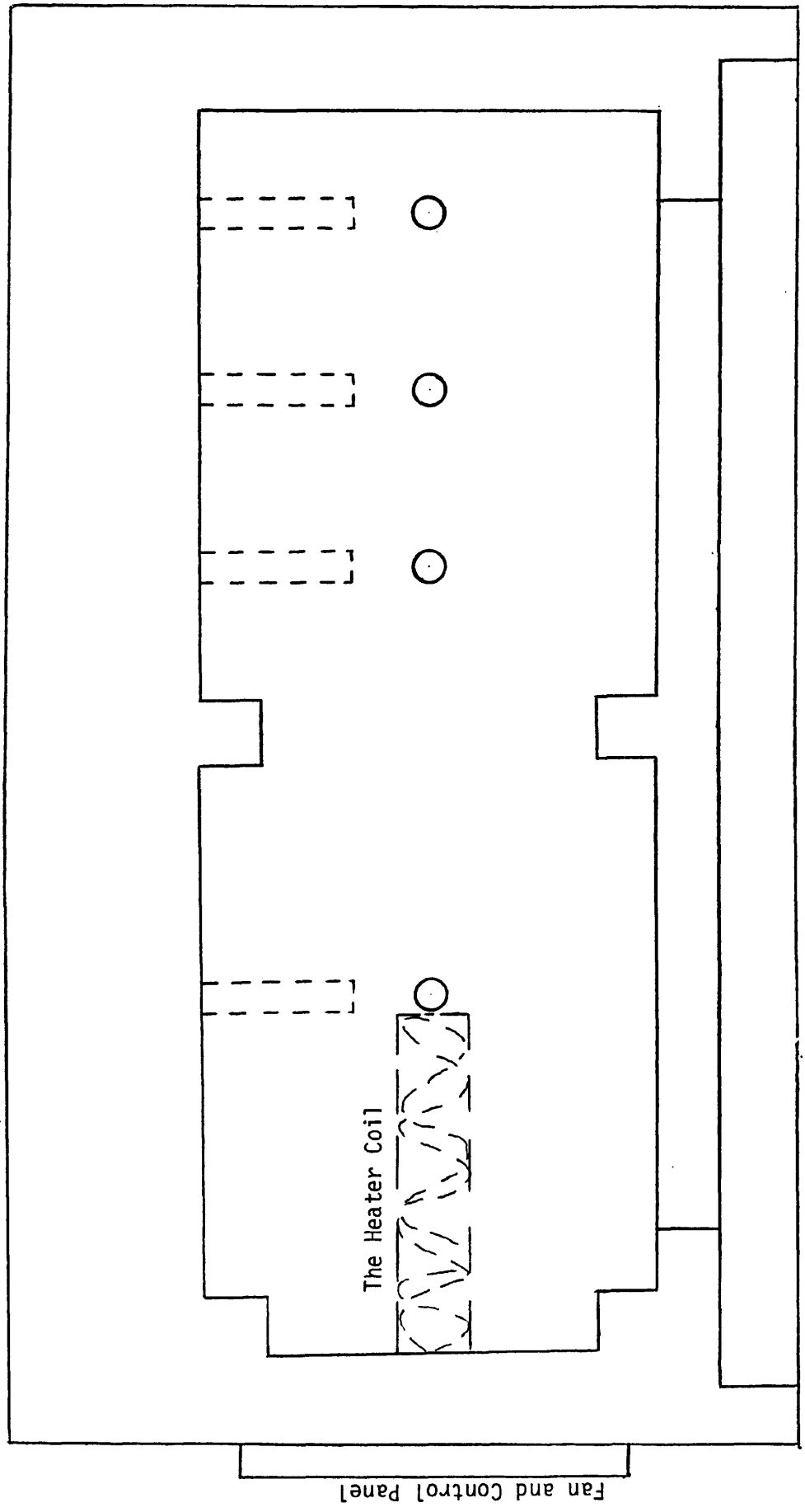


Figure 6.1 Shows the side view of the Heater Bar Rig.

the temperature can be taken from any of its six holes which are in two parallel rows of three each, at the top and side of the bar. Via thermo-couples, the temperature is measured and converted to microvolts, which is then amplified in the control instrumentation box which is apart from the bar itself. Through the transducers and amplifiers, the relationship between the temperature output in centigrade (^{o}C) and volts is approximately $1^{o}C$ to 1mV. Due to the holes it is possible to have the rig used as a Single Input Multi Output (SIMO) system.

The bar is 30 cm long with each half being 15cm. It has an outer radius of 3.44cm and an inner one of 0.5cm for the first half. The mild steel material has a density of 7849.2 Kg/m^3 and a specific heat capacitance of 461 J/KgK. The bar is placed inside a glass case which is perforated and to which is attached two fans which can be used for cooling. Potentially these can also be used as an input, but as of now they are only switched on or off manually.

6.3 SYSTEM CHARACTERISTICS AND MODELLING

From the previous section it becomes apparent that it is a slow responding rig, hence with a very large time constant. Although a distributed parameter system with a temperature profile which gradually decreases in value the further away from the heater coil that one gets along the bar, it is possible to represent the rig as a first or second order system. The step response output of the four holes are shown in Figure 6.2 from which it can be seen that the possibility of approximating the system order as one or two is certainly correct.

On the assumption of a finite order system, analysis of the system model now follows.

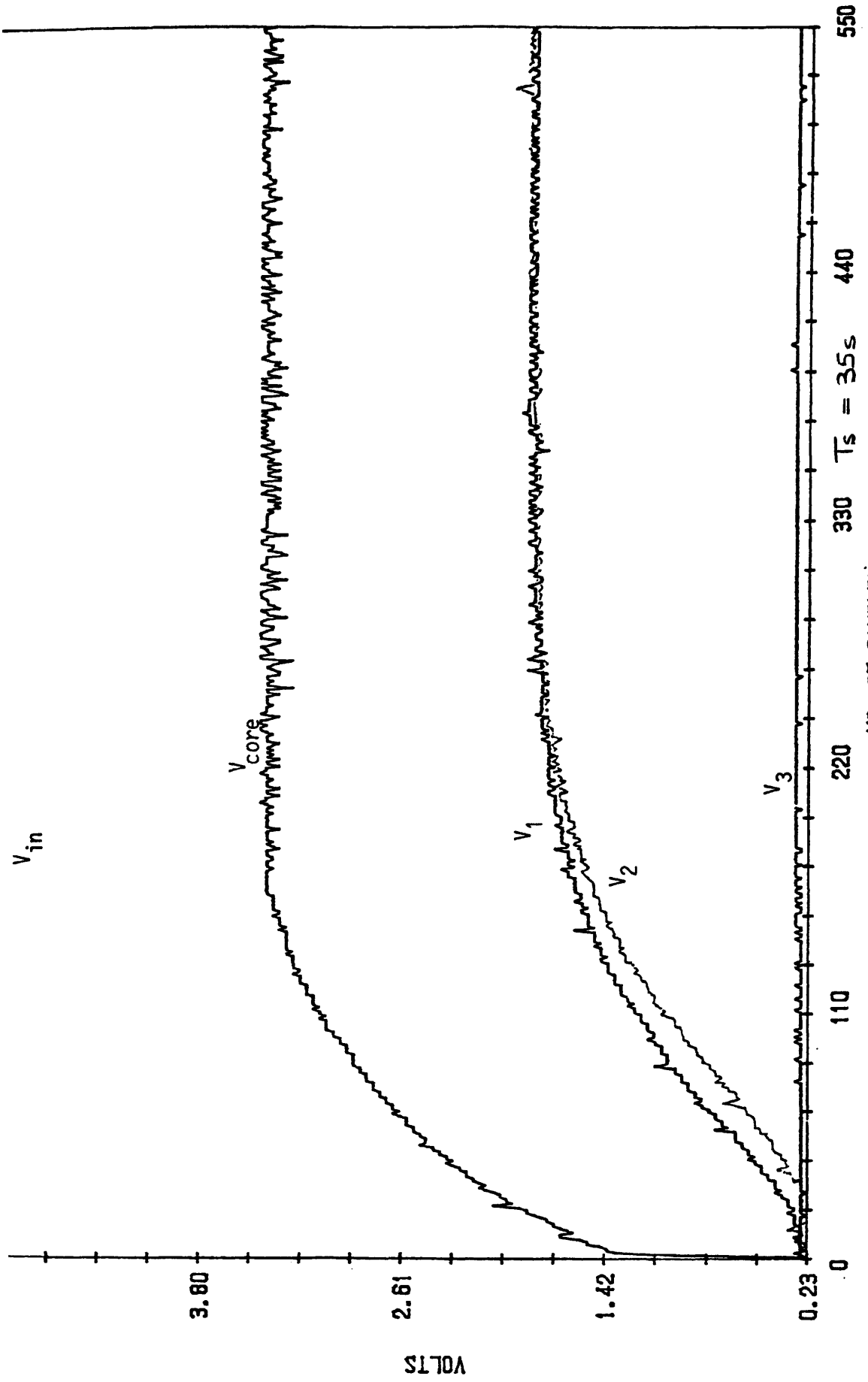


Figure 6.2 Shows the step responses of the four holes on the bar.

MODEL DERIVATION

The Single-Input, Multi-Output (SIMO) System

Since all the holes on the rig could be used as outputs the system can be called a SIMO system. Based on this and the modelling ideas in [64], an analogous electrical system to the rig can be formed as depicted in Figure 6.3a below, from which the system state space model and transfer function could be derived.

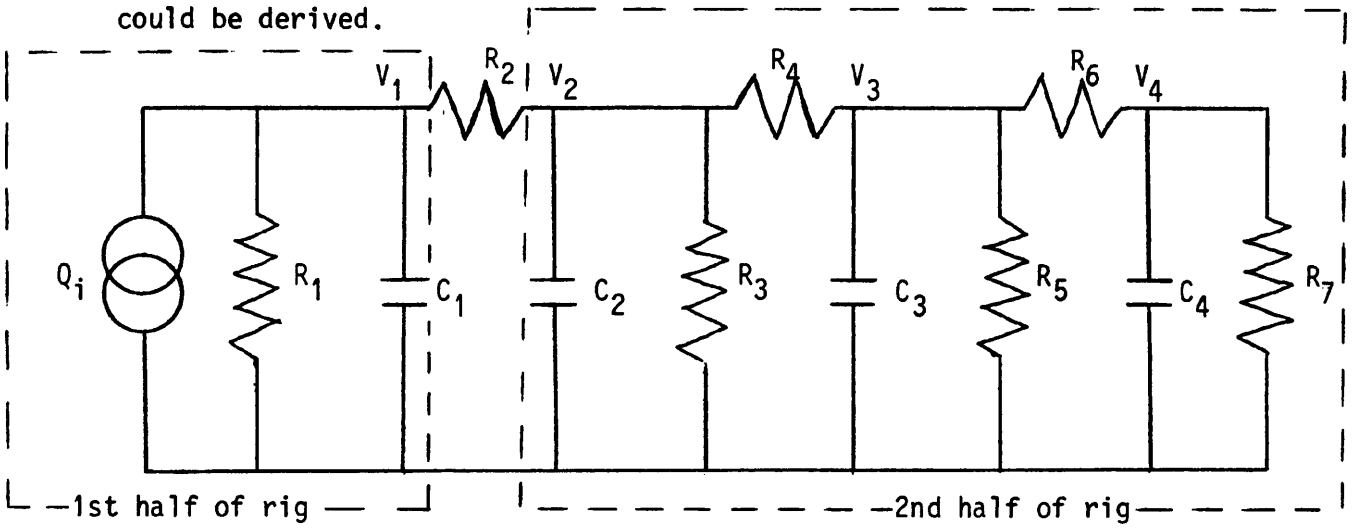


Figure 6.3a The SIMO System

Q_i = Heat source; V_1 ≡ temperature of core (T_{in});
V_2 ≡ temperature of 1st hole (T_1) ; V_2 ≡ temperature of 2nd hole (T_2);
V_3 ≡ temperature of 3rd hole (T_3) ; R_1-R_7 ≡ thermal dissipators;
C_1-C_4 ≡ thermal flow stores.

Although in the above diagram the core temperature is assumed to be T_{in} it could easily be represented as another system output as well. It should also be noted that while the outputs are represented above in terms of voltages, there is a correlation between the hole temperatures and their voltages. The relationship is given by $1^oC \sim 1mV$.

Potentially the temperatures of the three holes on the second half of the rig, T_1, T_2, T_3, can be the different system outputs as indicated above,

but due to the controllability problems the rig is best treated as a Single-Input-Single Output (SISO) system.

The Single-Input, Single-Output (SISO) System

The rig, as a SISO system, implies that only one of the three holes could be used as a system output, which means that only the chosen hole's temperature could be controlled. The new system diagram therefore changes from that shown in Figure 6.3a for a SIMO system to that in Figure 6.3b for a SISO system.

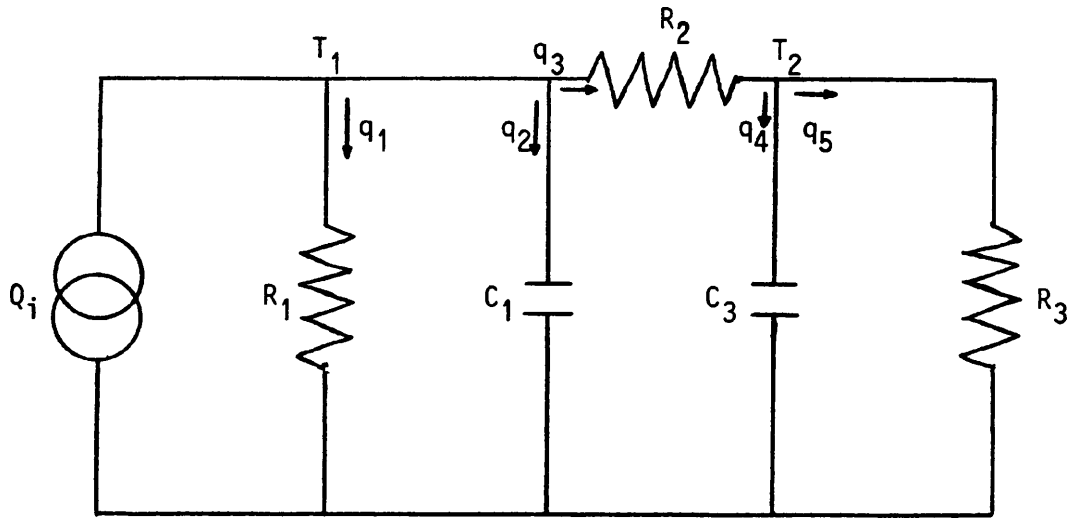


Figure 6.3b The SISO system

Q_i = Heat source; T_1 = temperature of core (1st half);

T_2 = temperature of any of the holes on the 2nd half of rig;

C_1,C_2 thermal flow stores; R_1-R_3 = thermal dissipators.

Using Figure 6.3b the system state space model and transfer function for any given second half hole as output is now easily derived. The derivation now follows.

The heater coil was assumed as the source, with the temperature of the core assumed as T_1, while any one of the holes on the second half of the rig provides the temperature T_2.

Taking T_2 as the system output the dynamic equations are:

$$q_3 = q_4 + q_5 = \frac{T_2}{R_3} + C_2 \frac{dT_2}{dt} = \frac{T_1 - T_2}{R_2} \tag{6.3.1}$$

$$Q_i = q_1 + q_2 + q_3 = \frac{T_1}{R_1} + C_1 \frac{dT_1}{dt} + \frac{T_1 - T_2}{R_2} \tag{6.3.2}$$

Rearranging the two equations above and using $\dot{T} = dT/dt$, then the state space model is obtained as:

$$\left.\begin{aligned} \begin{bmatrix} \dot{T}_1 \\ \dot{T}_2 \end{bmatrix} &= \begin{bmatrix} -\frac{(R_1+R_2)}{C_1 R_1 R_2} & \frac{1}{C_1 R_2} \\ \frac{1}{C_2 R_2} & -\frac{(R_2+R_3)}{C_2 R_2 R_3} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{C_1} \\ 0 \end{bmatrix} Q_i \\ \text{and the output is:} \quad T_2 &= \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} \end{aligned}\right\} \tag{6.3.3}$$

Taking the Laplace transform of the above, the system transfer function is:

$$\frac{T_2}{Q_i} = \frac{R_1 R_3}{s^2 C_1 C_2 R_1 R_2 R_3 + (R_1 R_2 C_1 + R_1 R_3 C_1 + R_1 R_3 C_2 + R_2 R_3 C_2)s + (R_1 + R_2 + R_3)} \tag{6.3.4}$$

Note: If R_2 assumes a value of zero the transfer function reduces to a first order system.

Although the system transfer function was obtained above, because of the various ways that the heat is transferred, i.e. conduction, radiation and

convection, the values of the R's and C's above are never constant but change with the input voltage, hence temperature of the heater coil. So, in a way it is a variable parameter system which would suit adaptive control. It should be noted that for the different holes available, there are different transfer functions obtainable, since the resistance and capacitance values differ for the various hole positions.

From the system responses shown in Figure 6.2, if the transfer function should be approximated as a first order function, perhaps it might be as a time delayed first order system.

6.4 MRAC APPLICATION TO THE RIG

A lot of problems were encountered during the attempts at model reference adaptive control of the rig which essentially stemmed from the slowness of the system response. The time constant of the system was found to be approximately 75 minutes and this coupled with the heat capacitance of the material, made the system very difficult to work with. Because of these reasons, all attempts at getting any form of adaptive control of the rig using the three different SISO algorithms failed. Different ways to solve the problems faced proved ineffective, amongst which was the use of big sampling periods. The basic problem was that due to the large heat capacitance, it was virtually impossible to control the system output, because even when the tracking error is greater than zero (i.e. positive) and the adaptive parameter values and control input were decreasing to counter this, the temperature, hence the system output voltage just kept on increasing. This resulted in a badly controlled output Figures 6.4 and 6.5 show the results of using the Goodwin et al and Kreisselmeier-Anderson algorithms. Due to the large number of samples taken results are shown for the reference and system outputs only.

In the end it was only by switching the adaptation off and on that any semblance of control was imposed on the system. The results and explanations now follow.

6.4.2 Results (see appendix for program listing)

Basically, what was done was that an error dead zone region was created within which the adaptive parameters and control input were switched off, i.e. reduced to zero. In this case the error dead zone was $e(t) > 0$, but as soon as the error goes outside this region, which for this particular problem meant $e(t) < 0$, the adaptive parameters and the control input were switched on again. By this means the system output was made to track the reference model output.

While any of the algorithms could have been used, it was the Goodwin algorithm that was used, although with various reference models and sampling times as given below. The three different hole positions on the second half of the bar were used as outputs on different occasions, though the middle hole was the one used most frequently. The results now follow for the different models.

MODEL 1 $\quad y_m(t) = 0.3935r(t) + 0.6065y_m(t-1)$

For the middle hole there are two results for two different sampling periods, i.e.

(i) for T_s = 10 seconds (see Fig. 6.6a)

(ii) for T_s = 35 seconds. (see Fig. 6.6b)

Since the system is a slow one, it was felt that a slower reference model would be a more reasonable one to use, hence the reference model was changed and the larger sampling time above used, unless otherwise stated.

MODEL 2 $y_m(t) = 0.0198r(t) + 0.9802y_m(t-1)$

(a) For the first hole a faster sampling time of eighteen seconds was used, due to the fact that it had a quicker response than the others, and the reference input was 0.7 volts. The result is shown in Figure 6.7a.

(b) For the middle hole, using a reference input voltage of one volt, the result obtained is shown in Figure 6.7b .

(c) For the extreme hole (the last one), two different step input values of 1.0 volts and 1.3 volts were used with their respective results depicted in Figures 6.7c and d.

From Figure 6.2 it is obvious that the extreme hole, which is furthest away from the heater, would have the slowest time response of all, hence a slower reference model still was used on this hole, i.e.

MODEL 3 $y_m(t) = 0.00995r(t) + 0.99005y_m(t-1)$

Using the larger sampling period and a reference input of one volt, the result obtained is shown in Figure 6.8 for both the reference and system outputs.

6.5 COMMENTS

(i) Apart from the obvious slowness of the heater bar, the most difficult problems encountered were with the analogue to digital (A/D) converter used for the experiments. The first problem was that experienced for all the rigs, i.e. the quantization error for the A/D converter (which was bigger than for the previous rigs) of about ± 0.07 volts, the result of which caused the system output not to be smooth. This caused all the results obtained to have a lot of ups and downs in the value of the system output.

Most of the problems encountered in this chapter can be attributed to errors made in the modelling of the Heater bar rig. The most fundamental was as a result of using a first order model to represent the infinite dimensioned system. This is too small a model order for a good control of the bar temperature and hence the system output.

A confirmation of this can be made from a frequency response analysis of the system. Ideally for research purposes an eighth or tenth order system model will be needed of similar configuration to that shown in fig 6.3b. This will be able to cope with the difficulties in controlling the output temperature since the other heat capacitors in the system will be a contributive factor to the output and also the sampling interval can then be made much bigger than that used in the work here.

The other problem was that of erroneous spikes in the results also caused by the A/D converter, since logic dictates that it is almost impossible to have instantaneous output increments as depicted with circles in Figure 6.9a .

(ii) While the results show the possibility of controlling the system, it must be noted that the system output would track the reference output as much as possible, despite the introduction of the error dead zone and the switching on or off of the adaptive parameters accordingly, although the system output would have a slight oscillatory motion about the reference output as depicted in Figure 6.9b . Depending on the process to be controlled, the system output can be made to be as close to the reference output as possible, provided the quantization error could be reduced to the very minimum.

(iii) The control problem might be a lot easier if the two fans built into the system could be programmed and adaptively controlled, i.e. switched on when the desired temperature level is being reached, or even have their speed, and hence their cooling ability, increased depending on whether the system output was overshooting the desired reference output or not, instead of the manually operated on or off switches used to operate them now.

During all the experiments and results reported here, the fans were not used for any of them.

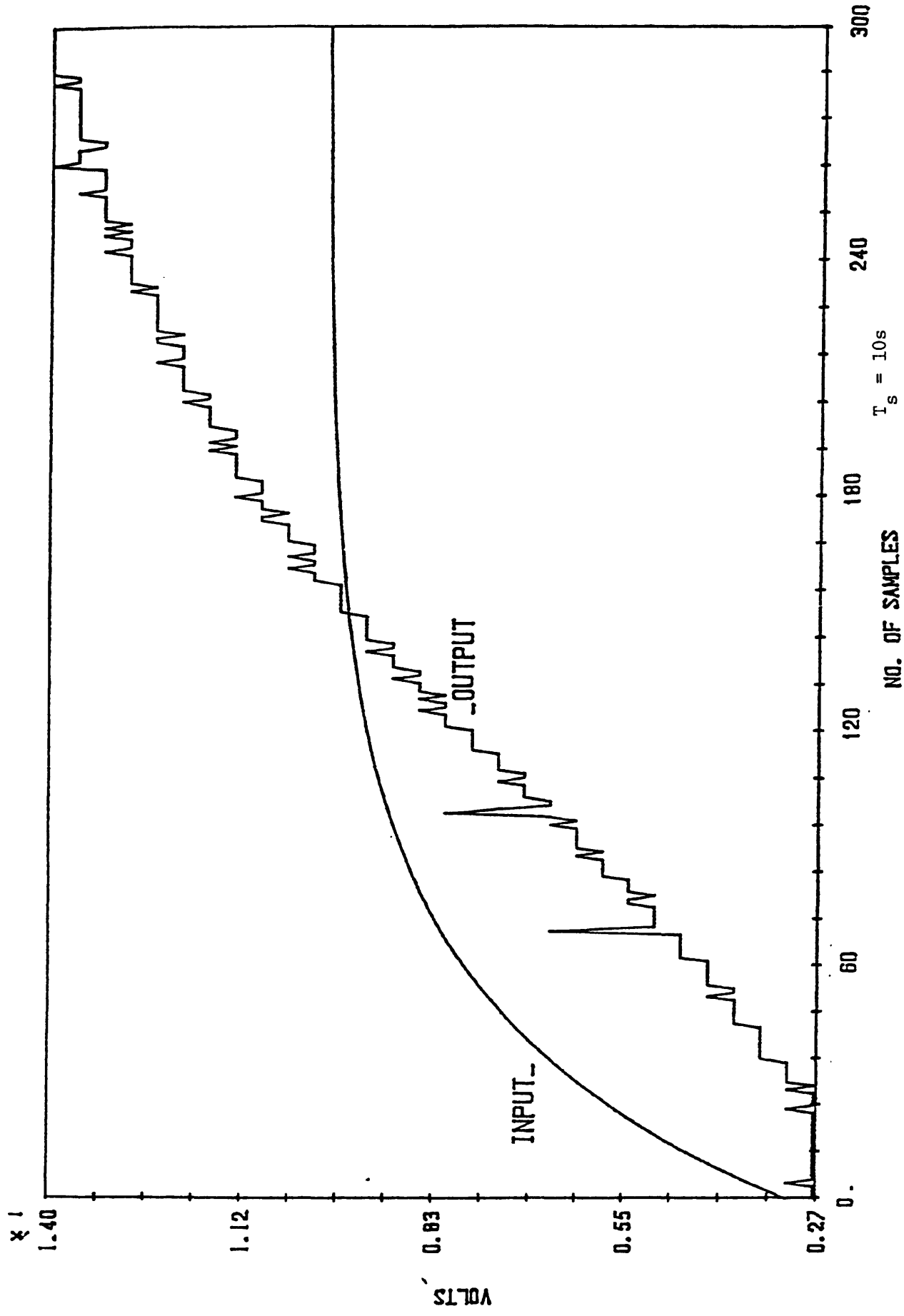


Figure 6.4 Shows the SISO Goodwin et al algorithm results.

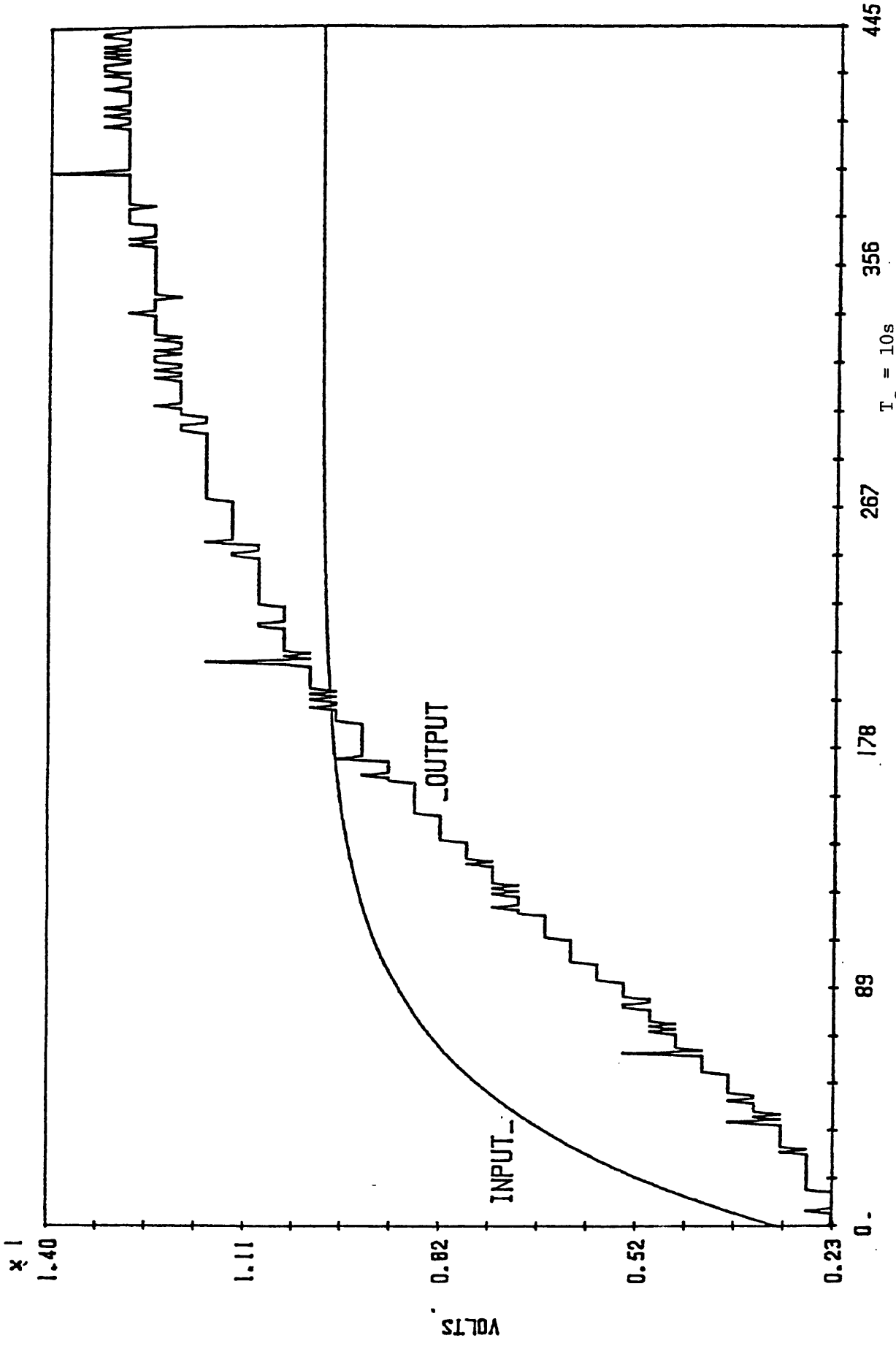


Figure 6.5 Shows the SISO Kreisselmeier and Anderson algorithm results.

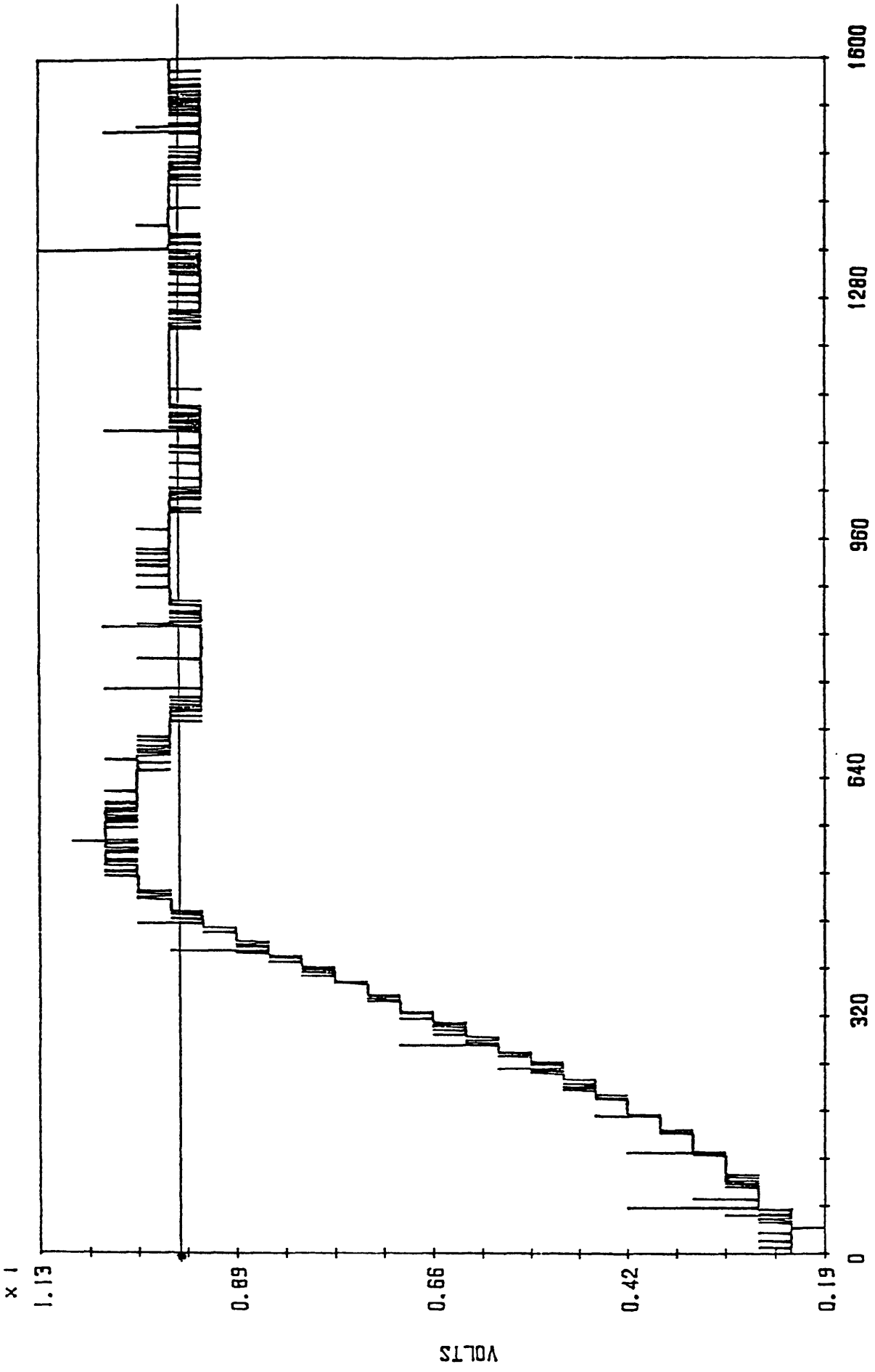


Figure 6.6 (a) The results for a sampling period Ts = 10 secs.

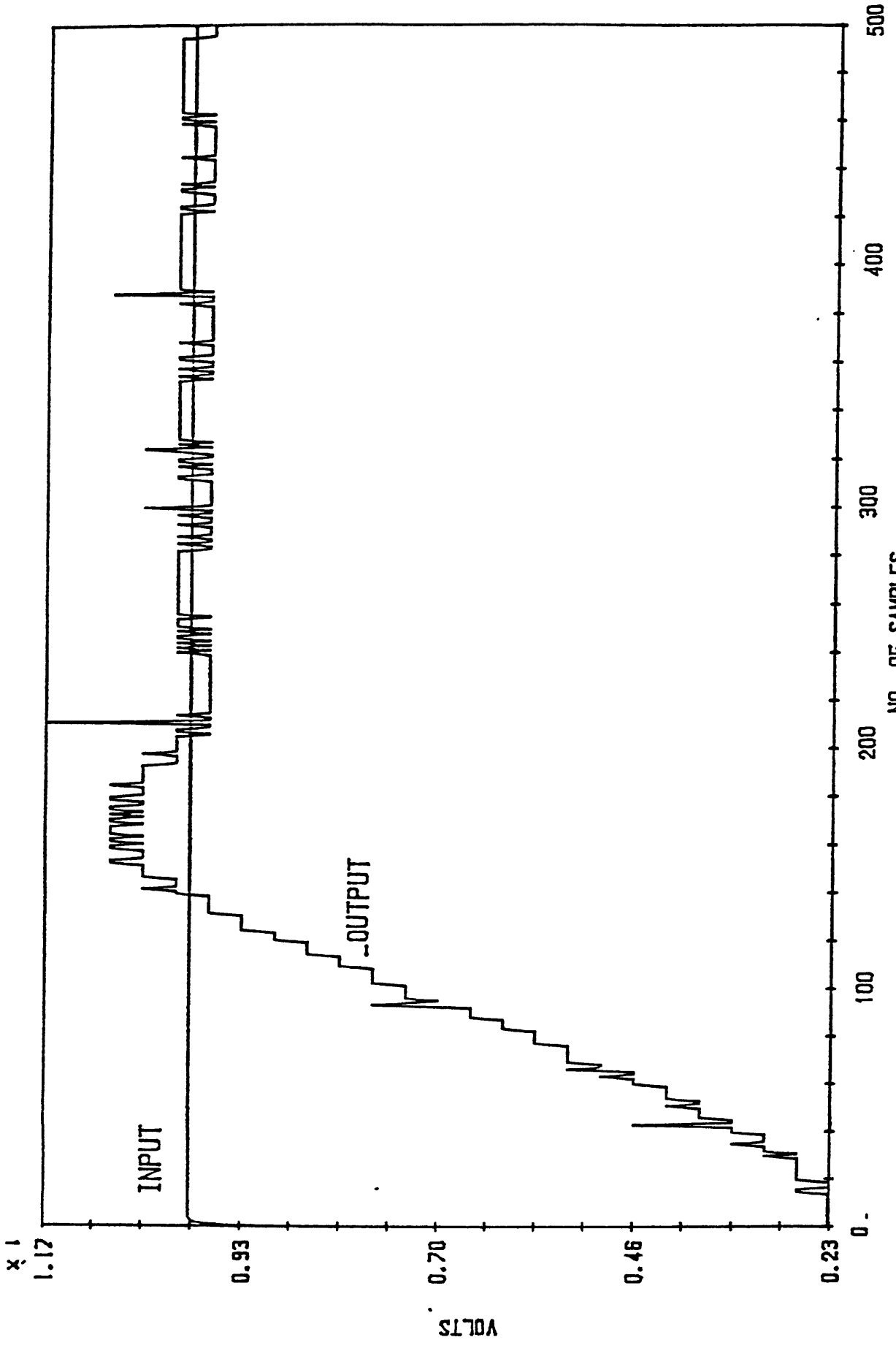
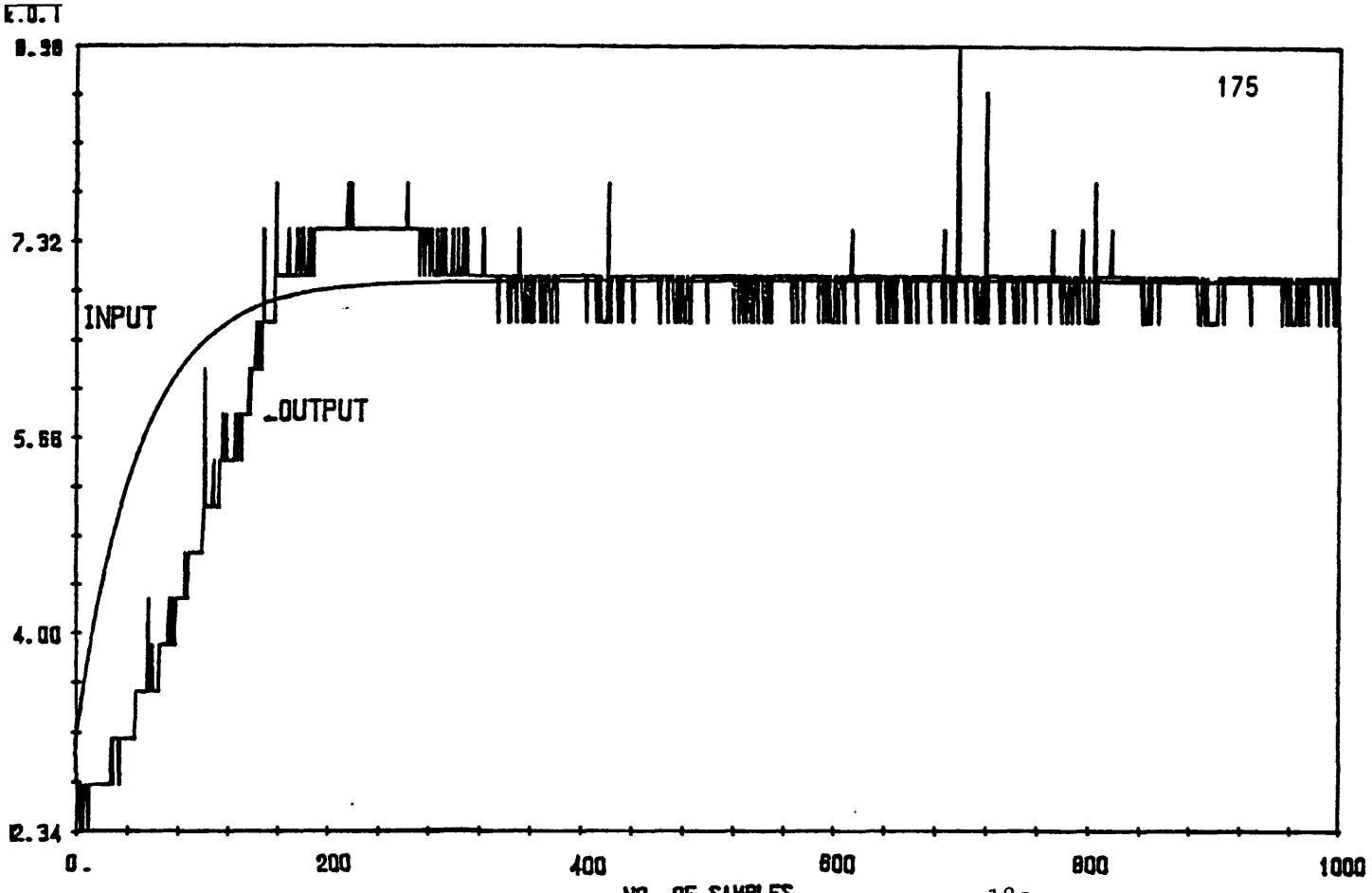
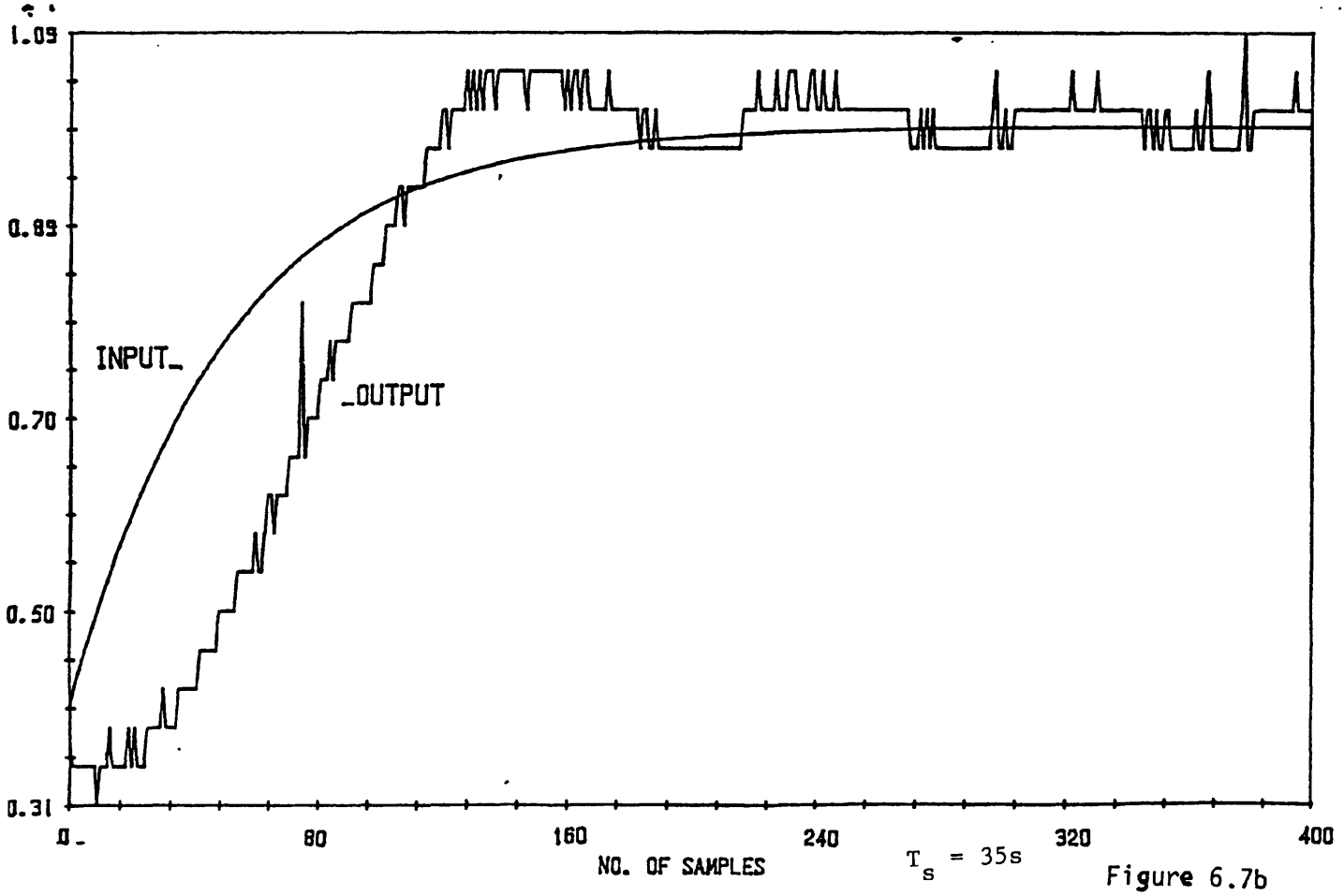


Figure 6.6 (b) The results for a sampling period Ts = 35 secs.



T_s = 18s

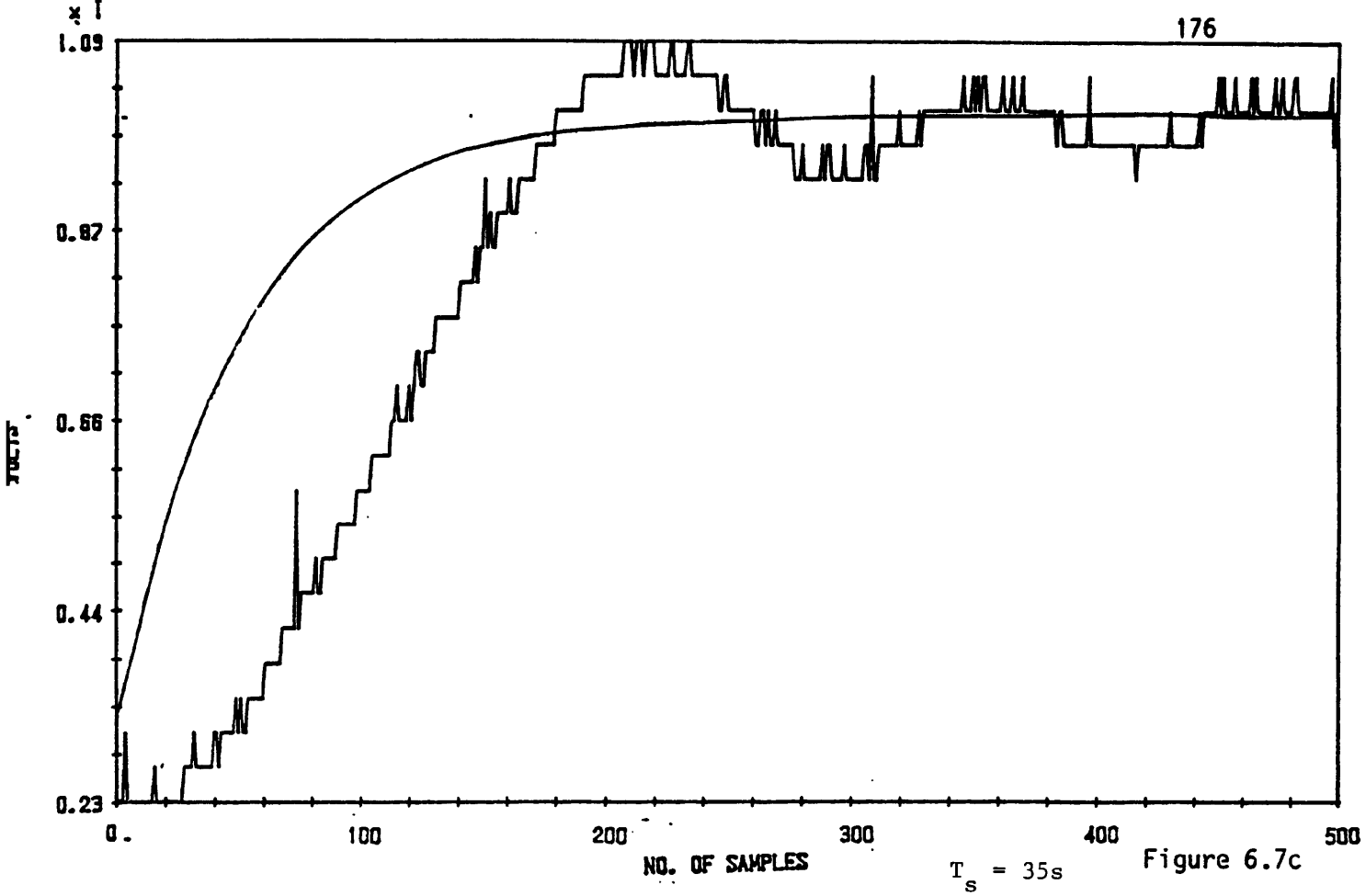
Figure 6.7a



T_s = 35s

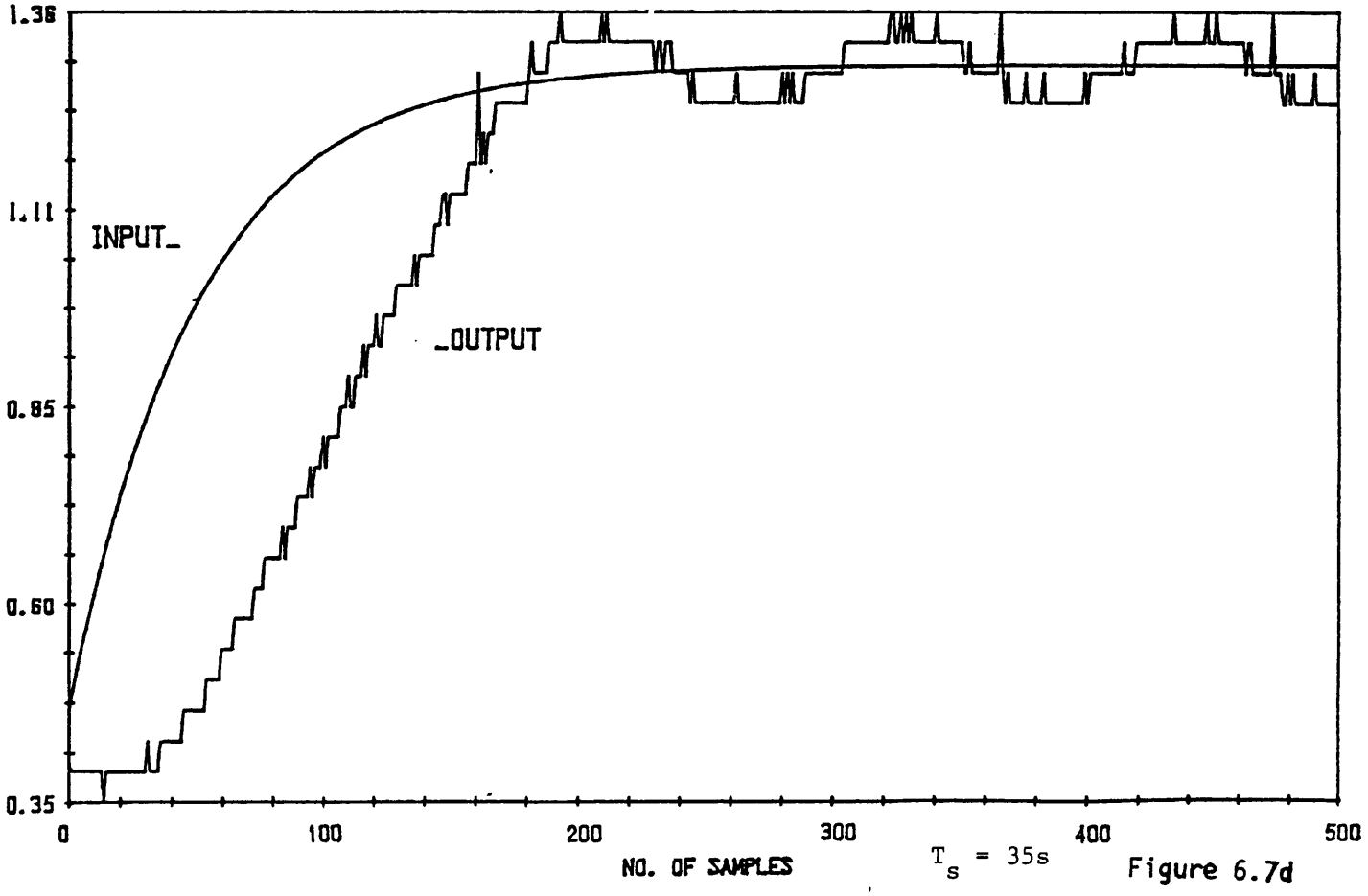
Figure 6.7b

Figure 6.7 (a) The results for the First hole control.
(b) The results for the Second hole control.



T_s = 35s

Figure 6.7c



T_s = 35s

Figure 6.7d

Figure 6.7 (c),(d) The results for the Third hole control.

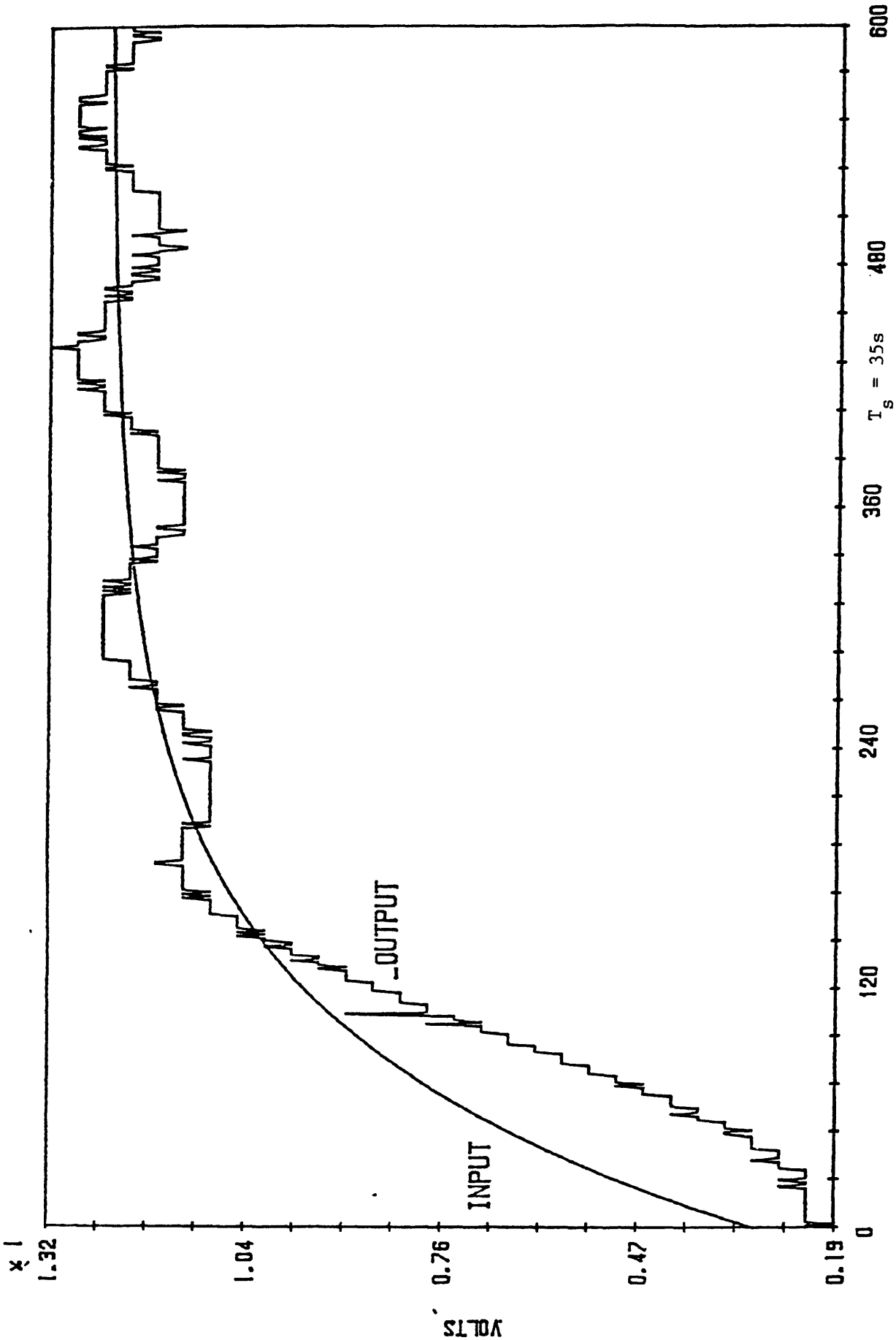


Figure 6.8 The results for the Third hole control using a much slower reference model.

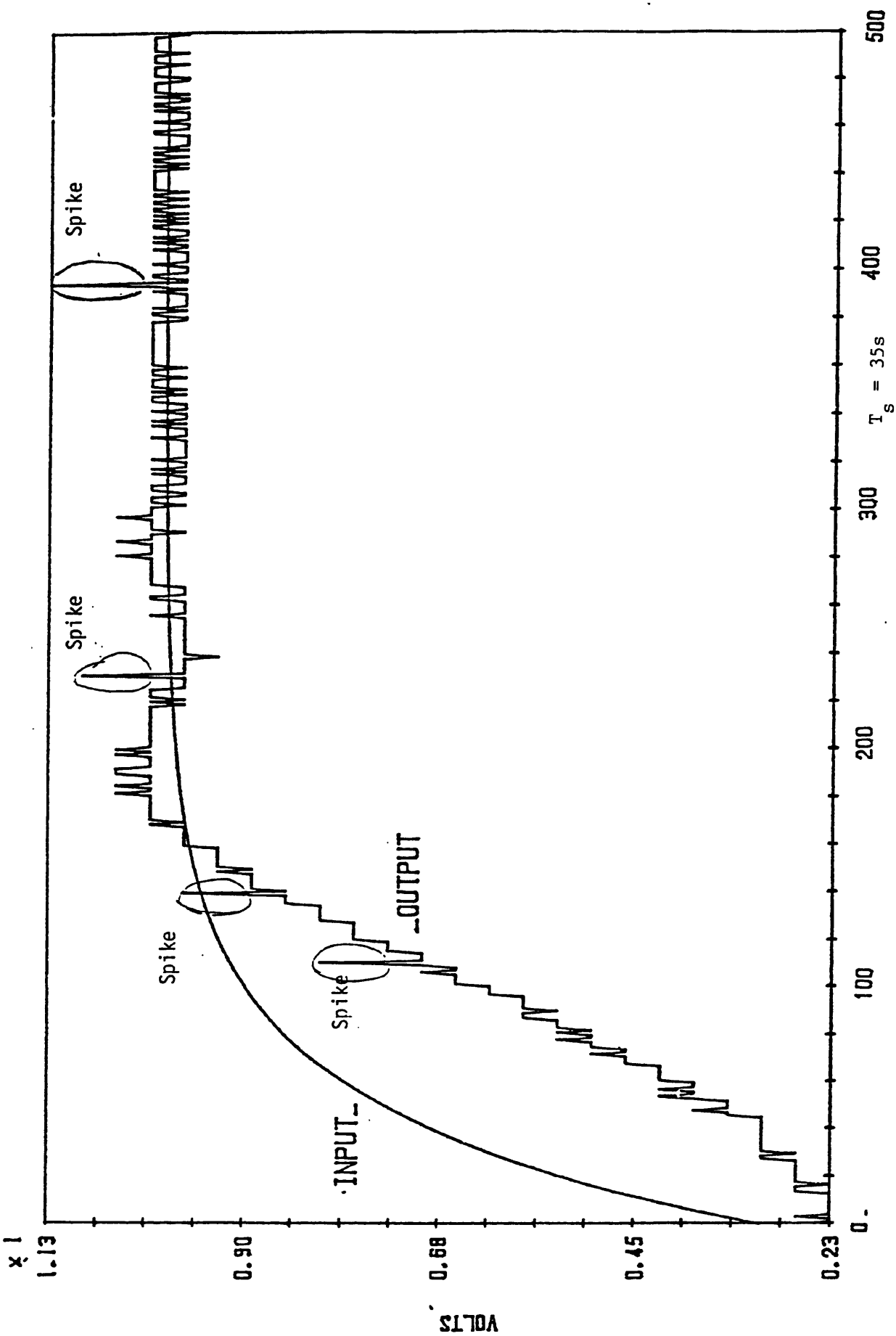


Figure 6.9 (a) Shows the erroneous spikes in the results.

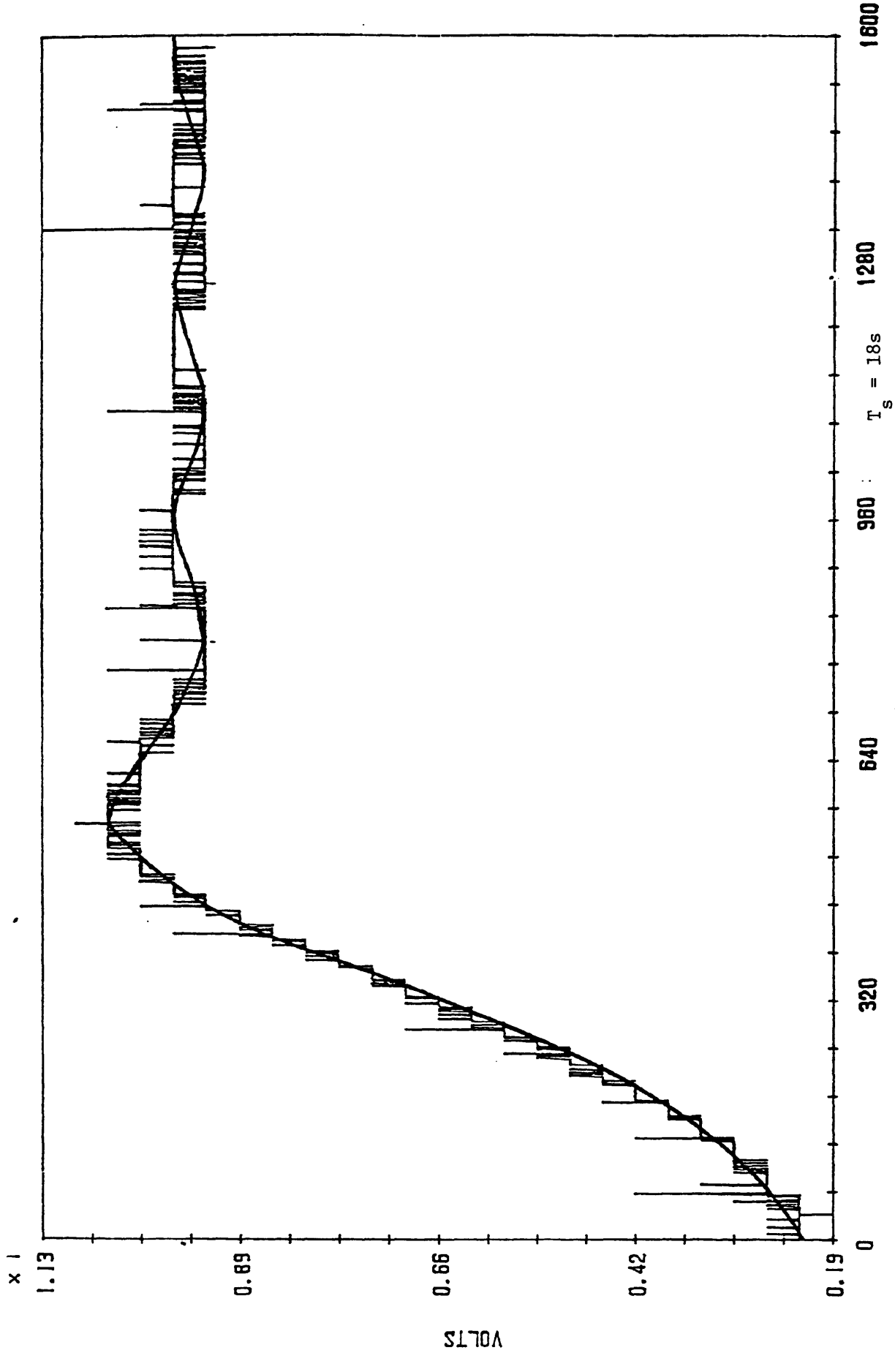


Figure 6.9 (b) Shows the slight oscillatory motion of the system output about the reference output.

Chapter 7

DISCUSSIONS AND CONCLUSIONS

This thesis has been written in a way that permits easy correlation between the 3 parts, which were the historical background, theory in Chapters 1-3, and the experimental section in Chapters 4-6 consisting of practical tests on laboratory rigs.

As mentioned in the first chapter, although Model Reference Adaptive Control (MRAC) has been around for some time, it still has not got wide recognition and application of its algorithms in the industry. It is the author's hope that the work contained in this thesis, which has been made as simple as possible in every way, might persuade more people to have confidence in its theory and applications. Part of the problem has been with the stability of the algorithms as pointed out by Rohrs et al [21] amongst others, but with the emergence of new algorithms, two of which were used in this work, and more investigations into robustness through persistency of excitation, hopefully things will change.

The work has been inspired by all these and the aim was to attract interest through testing algorithms on rigs by making these as simple as possible. The BASIC computer language was used throughout, being the simplest language available, and intentional mismodelling and output disturbances of the systems were introduced. Three different algorithms by different groups of people were used and, although at first appeared to differ, were shown in Chapter 3 to be quite similar with common underlying principles. Though there are numerous other algorithms, basically the same principles as for the three used exist in them as well. It was these fundamental similarities which led to the derivation of a multivariable extension to the scalar algorithm suggested by Kresselmeier and Anderson [27]. The major benefits of this

algorithm were in the reduction of overshoot and faster convergence of the system and reference outputs, even though in terms of tracking there is still a lot of room for improvement.

The laboratory rigs used were chosen because of the variety they offered in terms of industrial processes and problems that go with them. In Chapter 4 the attention was on the Coupled Tanks which symbolise chemical processes amongst others, though the industrial system might be more complex, i.e. in terms of the number of parameters to be controlled, which in most cases would include temperature control. A persistent problem on this rig was the drift of the transducers as a result of which the system parameters at any operating level or point were never exactly the same but drifted with time. The results show the efficacy of using MRAC algorithms.

The Coupled Electric Drives apparatus in Chapter 5 represents a lot of industrial systems, some of which were mentioned in Section 5.1 . Here a new set of problems was encountered which include strong input-output interactions (also in the Coupled Tanks but on a much weaker level), and noisiness of the outputs due to the belt speed which caused the tension output to vibrate.

The last rig was the heater bar based on a temperature control industrial process. This was a very slow process and the problems met had to do with this fact. As a result of the slowness of response, all the three scalar algorithms failed in the attempts at controlling the system temperature. In the end a radical solution was found by switching on or off the adaptive parameters and control input, depending on the tracking error.

Another problem, but one which was encountered with all the rigs, had to do with the quantization error of the A/D converters. As a result of this, depending on how large the outputs were, the system output plots might not be continuously smooth. Also to do with the converters were the spontaneous spikes experienced from time to time in the results.

In using different algorithms, comparisons could not be helped of their performances on the different rigs. Overall, the Ortega et al algorithm proved to be the best of the SISO algorithms for, while it had larger overshoots than the Kreisselmeier-Anderson algorithm, its overshoots were smaller than those of the Goodwin algorithm; but it had the best tracking ability of the three, irrespective of the intentional modelling errors. It also handles external disturbances quite well, thus proving its robustness all round. The Goodwin algorithm also performed quite well considering that it was the only one that was not designed for robustness, since it was pre-Rohrs et al. It tracks well, but unfortunately the better the tracking the bigger its overshoots which might prove an handicap. Finally, the Kreisselmeier-Anderson algorithm as mentioned had the lowest overshoots, but its tracking of the reference outputs was not too good. The author feels that this has to do with its lack of an adaptive gain which could have made a difference if it existed. Probably modifying the algorithm to include this might improve its all round performance and robustness.

Similarly, for the multivariable algorithms, the derived extension to the Kreisselmeier-Anderson algorithm proved better in terms of lower overshoot and faster convergence times of the reference and system outputs, but once again, as with the scalar algorithm, was worse off in terms of tracking the reference outputs compared to the MIMO Goodwin algorithm.

Finally, recommendations for future work. Although it was the robustness and effectiveness of the algorithms that were investigated as regards disturbances and mismodelling, an interesting area to extend the work to would be model reduction whereby reference models would contain only the essential and important information necessary for controlling the system. Another area would be in the modification of the Kreisselmeier-Anderson algorithm to include a variable adaptive speed factor. Though the rigs were scaled down versions of different industrial processes, it would be nice to see the

algorithms being used in the industry itself.

Lastly, with the enormous potentials of the computer and the interest of people in expert systems, there should be ways of incorporating both adaptive control and expert systems into one large but versatile package. Amongst the options that should be available in such a package might be different algorithms to serve various functions such as regulation, identification, control, or all of these. It should be able to assess a given system then choose an optimum sampling time and perhaps decide when to switch on or off the adaptation according to laid down rules, objectives plus design criteria or goals. It must also be able to remember or learn from past problems or systems the similarities of new systems, thus arriving at the controllers and solutions in quicker times.

REFERENCES

1. B.D.O. Egardt, "Unification of some discrete-time adaptive control schemes". IEEE T.A.C., vol. AC-25, No. 4, August 1980.

2. I.D. Landau, "Combining model reference adaptive controllers and stochastic self-tuning regulators". Automatica, vol. 18, No. 1, pp. 77-84, 1982.

3. I.D. Landau, "Model reference adaptive controllers and stochastic self-tuning regulators - A Unified Approach". Journal of Dynamic Systems, Measurement and Control, vol. 103, pp. 404-415, December 1981.

4. P.C. Gregory, Ed. (1959), "Proceedings of the Self Adaptive Flight Control Systems Symposium", W.A.D.C. Technical Report 59-49, Wright Patterson Air Force Base, Ohio.

5. P.V. Osburn, H.P. Whittaker and A. Keezer, "New developments in the design of adaptive control systems". Institute of Aeronautical Sciences, Paper 61-69, 1961.

6. R.L. Butchart and B. Shackcloth, "Synthesis of model reference adaptive control systems by Lyapunov's second method". Proc. 1965 IFAC Symposium on Adaptive Control, (Teddington, England), Instrumen. Soc. America, 1966.

7. P.C. Parks, "Lyapunov redesign of model reference adaptive control systems". IEEE Trans. Automatic Contr., vol. AC-11, pp. 362-367, July 1966.

8. R.V. Monopoli, "Model reference adaptive control with an augmented error signal". IEEE Trans. Automatic Contr., vol. AC-19, pp. 475-484, October 1974.

9. I.D. Landau, "A hyperstability criterion for model reference adaptive control systems". IEEE Trans. Automatic Contr., vol. AC-14, pp. 552-555, October 1969.

10. G. Hang and P.C. Parks, "Comparative studies of model reference adaptive control systems". IEEE Trans. Automatic Contr., vol. AC-18, No. 5, pp. 419-428, October 1973.

11. Narendra, K. and L.S. Valavani, "A comparison of Lyapunov and Hyperstability approaches to adaptive control of continuous systems". IEEE Trans. Automatic Contr., vol. AC-25, No. 2, pp. 243-246, April 1980.

12. A. Feuer, B.R. Bramish and A.S. Morse, "An unstable dynamical system associated with model reference adaptive control". IEEE Trans. Automatic Contr., vol. AC-23, No. 3, June 1978.

13. K. Narendra and L.S. Valavani, "Stable adaptive controller design - Direct Control". IEEE Trans. Automatic Contr., vol. AC-23, No. 4, August 1978.

14. A. Feuer and A.S. Morse, "Adaptive control of single input single output linear systems". IEEE Trans. Automatic Contr., vol. AC-23, No. 4, August 1978.

15. B.D.O. Egardt, "Stability analysis of adaptive controllers". Lecture Notes in Control and Information Sciences, No. 20, Springer-Verlag, 1979.

16. G.C. Goodwin, P.I. Ramedge and P.E. Caines, "Discrete time multivariable adaptive control". IEEE Trans. Automatic Contr., vol. AC-25, No. 3, June 1980.

17. A.N. Payne, "Adaptive one-step-ahead control subject to an input amplitude constraint". Int. J. Contr., vol. 43, No. 4, pp. 1257-1269, 1986.

18. D. Janecki, "Globally stable and exponentially convergent adaptive control". Int. J. Contr., vol. 43, No. 2, pp. 601-613, 1986.

19. M. Kung and B.F. Womack, "Stability analysis of a discrete-time adaptive control algorithm having a Poly Input". IEEE Trans. Automatic Contr., vol. AC-28, No. 12, pp. 1110-1112, Dec., 1983.

20. I.D. Landau, "Adaptive control - The Model Reference Approach". Control and Systems Theory, vol. 8, Marcel Dekker Inc., New York, 1979.

21. C.E. Rohrs, L. Valavani, M. Athens and G. Stein, "Robustness of continuous time adaptive control algorithms in the presence of unmodelled dynamics". IEEE Trans. Automatic Contr., vol. AC-30, No. 9, September 1985.

22. P.A. Ioannon and K.S. Tsakalis, "A robust direct adaptive Controller". IEEE Trans. Automatic Contr., vol. AC-31, No. 11, November 1986.

23. I.M.Y. Marcels and R.R. Bitmead, "Nonlinear dynamics in adaptive control chaotic and periodic stabilization". Automatica, vol. 22, No. 6, pp. 641-655, 1986.

24. K.S. Narendra and A.M. Annaswamy, "Robust adaptive control in the presence of Bounded disturbances". IEEE Trans. Automatic Contr., vol. AC-31, pp. 306-315, No. 4, April 1986.

25. P.V. Kokotovic and P.A. Ioannon, "Robustness redesign of continuous time adaptive schemes". IEEE Proc. Conf. on Decision and Contr., pp. 522-527, 1981.

26. G.C. Goodwin, D.J. Hill, D.Q. Mayne and R.H. Middleton, "Adaptive robust control (convergence, stability and performance)". IEEE Proc. of 25th Conf. on Decision and Contr., Athens, Greece, December, 1986.

27. G.K. Kreisselmeier and B.D.D. Anderson, "Robust model reference adaptive control". IEEE Trans. Automatic Contr., vol. AC-31, No. 2, February 1986.

28. R. Ortega, L. Praly and I.D. Landau, "Robustness of discrete-time direct adaptive controllers". IEEE Trans. Automatic Contr., vol. AC-30, No. 12, December 1985.

29. L. Praly, "Robustness of model reference adaptive control". Proc. 3rd Yale Workshop on Application of Adaptive Systems Theory, June 1983.

30. G.A. Oluwande, M.Sc. Dissertation on Model Reference Adaptive Control, "Application to a Laboratory Rig". UMIST, Oct., 1985.

31. G.C. Goodwin and K.S. Sin, "Adaptive filtering, prediction and control". Information and Systems Science Series, Prentice Hall, New Jersey, 1984.

32. W.R. Cluett, S.L. Shah and D.G. Fisher, "A comment on Rohrs' discrete time adaptive control results".

33. G. Zames, "On the input-output stability of time varying nonlinear feedback systems, Part I - Conditions derived using concepts of loop gain, conicity and positivity". IEEE Trans. Automatic Contr., vol. AC-11, No. 2, 1966.

34. R.R. Leitch, "Input-output relative stability measures". IEE Proc. D, vol. 129, pp. 37-38, January 1982.

35. K. Narendra and Y. Lin, "Stable discrete adaptive control". IEEE Trans. Automatic Contr., vol. AC-25, No. 3, June 1980.

36. R.V. Monopoli and K.N. Shah, "Model reference adaptive control with inexact model matching, Part I". pp. 840-847, Proc. 1982 Americas Control Conference, June 14th-16th.

37. H. Elliot and W.A. Wolovich, "A parameter adaptive control structure for linear multivariable systems". IEEE Trans. Automatic Contr., vol. AC-27, No. 2, April 1982.

38. Goodwin G.C. and Long R.S., "Generalization of results in multivariable adaptive control". IEEE Trans. Automatic Contr., vol. AC-25, No. 6, December 1980.

39. R. Johansson, "Parametric models for linear multivariable systems for adaptive control". IEEE Trans. Automatic Contr., vol. AC-32, No. 4, April 1987.

40. K.A. Osman and E.W. Kamen, "Adaptive regulation of MIMO linear discrete time systems without requiring a persistent excitation". IEEE Trans. Automatic Control, vol. AC-32, No. 5, May 1987.

41. M. Kinnaert, R. Hanus and J.L. Henrotte, "A new decoupling pre-compensator for indirect adaptive control of multivariable linear systems". IEEE Trans. Automatic Contr., vol. AC-32, No. 5, May 1987.

42. M.S.I. Zinober, O.M.E. El-Ghezawi, S.A. Billings, "Multivariable variable-structure adaptive model following control systems". IEE Proc. D, vol. 129, No. 1, pp. 6-12, January 1982.

43. G.C. Goodwin and S.W. Chan, "Model reference adaptive control of systems having purely deterministic disturbances". IEEE Trans. Automatic Contr., vol. AC-28, No. 8, August 1983.

44. D.R. Yang and W.K. Lee, "Effects of model structure, non-zero D.C. values and measurable disturbance on adaptive control". Proc. IFAC Adaptive Systems in Control and Signal Processing, San Francisco, U.S.A., 1983.

45. P.A. Ioannou and P.V. Kokotovic, "Adaptive systems with reduced models". Springer-Verlag (Publishers), New York, 1983.

46. B.B. Peterson and K.S. Narendra, "Bounded error adaptive control". IEEE Trans. Automatic Contr., vol. AC-27, No. 6, pp. 1161-1168, December 1982.

47. P.A. Cook and Z.J. Chen, "Robustness properties of model reference Adaptive control systems". IEE Proc., vol. 129, Pt. D, No. 6, November 1982.

48. P.A. Cook and Z.J. Chen, "Robustness of continuous time model reference adaptive control systems". Proc. Int. Conf. Control '85, (IEE Conference Publication No. 252), pp. 127-132.

49. B.D.O. Anderson and R.M. Johnstone, "Robust Lyapunov stability results and adaptive systems". IEEE Trans. Automatic Contr.

50. B.D.O. Anderson and C.R. Johnson, Jr., "On reduced-order adaptive output error identification and adaptive I.I.R. filtering". IEEE Trans Automatic Contr., vol. AC-27, No. 4, pp. 927-933, 1982.

51. P.J. Gawthrop and K.W. Lim, "Robustness of self-tuning controllers". IEE Proc., vol. 129, Pt. D, No. 1, January 1982.

52. C.R. Johnson and M.J. Balas, "Reduced order adaptive controller studies". Proc. of the Joint Automatic Control Conference, (J.A.C.C.), 13-15 August, 1982, San Francisco, California, WP2-D.

53. G. Kreisselmeier, "On adaptive state regulation". IEEE Trans. Automatic Contr., vol. AC-27, No. 1, pp. 3-16, February 1982.

54. L. Praly, "Robust model reference adaptive controllers, Part I, Stability Analysis". Proc. 23rd IEEE Conf. on Decision Control, Las Vegas, Nevada, December 1984.

55. G. Kreisselmeier and K. Narendra, "Stable model reference adaptive control in the presence of bounded disturbances". IEEE Trans. Automatic Contr., vol. AC-27, No. 6, pp. 1169-1176, Dec., 1982.

56. C. Samson, "Stability analysis of adaptively controlled systems subject to bounded disturbances". Automatica, vol. 19, No. 1, pp. 81-86, 1983.

57. K.J. Astrom, A commentary on the C.E. Rohrs et al paper "Robustness of continuous-time adaptive control algorithms in the presence of unmodelled dynamics". IEEE Trans. Automatic Contr., vol. AC-30, No. 9, p. 889, September 1985.

58. K.S. Narendra and A.M. Annaswamy, "A new adaptive law for robust adaptation without persistent excitation". IEEE Trans. Automatic Contr., vol. AC-32, No. 2, pp. 134-145, February 1987.

59. G.C. Goodwin, D.J. Hill, D.Q. Mayne, R.H. Middleton, "Adaptive robust control (convergence, stability and performance)". Proc. of 25th Conference on Decision and Control, Athens, Greece, Dec., 1986, WP4.

60. R. Lozano-Leal and R. Ortega, "Reformulation of the Parameter identification problem for systems with bounded disturbances". Automatica, vol. 23, No. 2, pp. 247-251, 1987.

61. R. Ortega and R. Lozano-Lead, "A note on direct adaptive control of systems with bounded disturbances". Automatica, vol. 23, No. pp. 253-254, 1987.

62. The Control Systems Centre (UMIST) Laboratory Manuals.

63. P.M. Zanker and P.E. Wellstead, "Practical features of self-tuning". Control Systems Centre Report No. 461, UMIST, September 1979.

64. P.E. Wellstead, "Introduction to physical system modelling". Academic Press Inc., (London), Ltd., 1979.

Appendix A1

Theorems of Lyapunov [20]

(a) 1st Theorem of Lyapunov (Stability Theorem)

For a system of the nth order, if a V function of definite sign can be selected, such that its time derivative W is also definite and opposite in sign, the given system is asymptotically stable in a sufficiently small region, which includes the origin. If the W function is merely semi-definite and opposite in sign, the system is stable but not necessarily asymptotically stable.

(b) 2nd Theorem (Instability Theorem)

For a system with nth order, if there exists a real valued function $V(x_1,x_2,\ldots,x_n)$ with the following properties:

(i) $V(x_1,x_2,\ldots,x_n)$ is continuous

(ii) the time derivative $dV/dt = W$ is negative definite

then (a) the system is unstable in the finite region for which V is not positive semi-definite,

(b) the response of the system is unbounded as $t \to \infty$ if V is not globally positive definite

Note: $$W = \frac{dV}{dt} = \frac{\partial V}{\partial x_1} X_1 + \frac{\partial V}{\partial x_2} X_2 + \ldots + \frac{\partial V}{\partial x_n} X_n$$

$V(x)$ is positive definite in U if for x in U :

(a) V has continuous partial derivatives

(b) $V(0) = 0$

(c) $V(x) > 0$ if $x \neq 0$, if (c) is replaced by (d) the function

is positive semi-definite,

(d) $V(x) \geq 0$.

The Meyer-Kalman-Yacnbovich (MKY) Lemma [7, 8]

Let A be a real n×n matrix, all of whose characteristic roots have negative real parts; let τ be a real nonnegative number, and let $\underline{d},\underline{k}$ be two real n-vectors. If

$$T(z) = \tau + 2\underline{k}'A(z)^{-1}\underline{d}$$

is a positive real function then there exist two n×n real symmetric matrices B, D and a real n-vector $\underline{q}$ such that

(1) $A'B + BA = -\underline{q}\underline{q}' - D$

(2) $B\underline{d} - \underline{k} = \sqrt{\tau}\underline{q}$

(3) D is positive semi-definite and B is positive definite.

(4) $\{\underline{x} \in E^n : \underline{x}' D \underline{x} = 0\} \cap [A', \underline{q}]^0 = \{0\}$

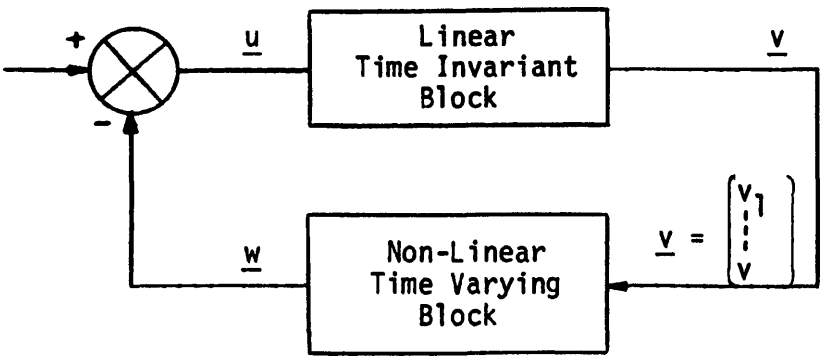
(5) $\underline{q} \notin [A, \underline{d}]^0$

(6) If iw, w real, is a zero of $-\underline{q}'A(z)^{-1}\underline{d} + \sqrt{\tau}$, then it is a zero of $\underline{d}'A(-z)^{-1} DA(z)^{-1} \underline{d}$.

The Popov Integral Inequality [20] for Figure A.1 is given by:

$$\eta(0, t_1) \triangleq \int_0^{t_1} \underline{V}^T \underline{w}\, dt \geq -\gamma_o^2 \quad \text{for all} \quad t_1 \geq 0 .$$

A feedback system of the form given in Figure A.1, when it is globally (asymptotically) stable for all feedback blocks satisfying the inequality equation above, is said to be *hyperstable*.



$\underline{v}$ is the input vector

$\underline{w}$ is the output vector of the feedback block

γ_0^2 is a finite positive constant (which does not depend on t_1).

Figure A.1 The Standard Nonlinear Feedback System.

Appendix A2

(i) $$||\hat{\theta}(t) - \theta_0|| \leq ||\hat{\theta}(t-1) - \theta_0|| \leq ||\hat{\theta}(0) - \theta_0||; \quad t \geq 1 \tag{1}$$

(ii) $$\lim_{N\to\infty} \sum_{t=1}^{N} \frac{e(t)^2}{c+\phi(t-1)^T\phi(t-1)} < \infty \tag{2}$$

and this implies:

(a) $$\lim_{t\to\infty} \frac{e(t)}{[c+\phi(t-1)^T\phi(t-1)]^{\frac{1}{2}}} = 0 \tag{3}$$

(b) $$\lim_{N\to\infty} \sum_{t=1}^{N} \frac{\phi(t-1)^T\phi(t-1)e(t)^2}{[c+\phi(t-1)^T\phi(t-1)]^2} < \infty \tag{4}$$

(c) $$\lim_{N\to\infty} \sum_{t=1}^{N} ||\hat{\theta}(t) - \hat{\theta}(t-1)||^2 < \infty \tag{5}$$

(d) $$\lim_{N\to\infty} \sum_{t=k}^{N} ||\hat{\theta}(t) - \hat{\theta}(t-k)||^2 < \infty \tag{6}$$

(e) $$\lim_{t\to\infty} ||\hat{\theta}(t) - \hat{\theta}(t-k)|| = 0 \tag{7}$$

for any finite k .

Proof:

(i) Subtracting θ_0 from both sides of (2.3.12) and using (2.3.5) and (2.3.14) we obtain:

$$\tilde{\theta}(t) = \tilde{\theta}(t-1) - \frac{a\phi(t-1)}{c+\theta(t-1)^T\theta(t-1)} \phi(t-1)^T \tilde{\theta}(t-1)$$

Hence using (2.4.4)

$$||\tilde{\theta}(t)||^2 - ||\tilde{\theta}(t-1)||^2 = a\left[-2 + \frac{a\phi(t-1)^T\phi(t-1)}{c+\phi(t-1)^T\phi(t-1)}\right] \frac{e(t)^2}{c+\phi(t-1)^T\phi(t-1)} \tag{8}$$

Now since $< a < 2$, $c > 0$, we have:

$$a\left[-2 + \frac{a\phi(t-1)^T\phi(t-1)}{c+\phi(t-1)^T\phi(t-1)}\right] < 0 \tag{9}$$

and then (1) follows from (8).

(ii) We observe that $||\tilde{\theta}(t)||^2$ is a bounded nonincreasing function, and by summing (8) we have:

$$||\tilde{\theta}(t)||^2 = ||\tilde{\theta}(0)||^2 + \sum_{j=1}^{t} a\left[-2 + \frac{a\phi(j-1)^T\phi(j-1)}{c+\phi(j-1)^T\phi(j-1)}\right] \frac{e(j)^2}{c+\phi(j-1)^T\phi(j-1)}$$

Since $||\tilde{\theta}(t)||^2$ is nonnegative, and since (1) holds, we can conclude (2).

(a) Equation (3) follows immediately from (2).

(b) Noting that

$$\frac{e(t)^2}{c+\phi(t-1)^T\phi(t-1)} = \frac{[c+\phi(t-1)^T\phi(t-1)]e(t)^2}{[c+\phi(t-1)^T\phi(t-1)]^2}$$

we establish (4) using (2).

(c) Equation (4) immediately implies (5) by noting the form of the algorithm ( ).

(d) It is clear that

$$||\hat{\theta}(t) - \hat{\theta}(t-k)||^2 = ||\hat{\theta}(t) - \hat{\theta}(t-1) + \hat{\theta}(t-1) - \hat{\theta}(t-2) \ldots \ldots \hat{\theta}(t-k+1) - \hat{\theta}(t-k)||^2$$

Then using the Schwarz inequality:

$$||\hat{\theta}(t) - \hat{\theta}(t-k)||^2 \leq k(||\hat{\theta}(t) - \hat{\theta}(t-1)||^2 + \ldots$$

$$\ldots + ||\hat{\theta}(t-k+1) - \hat{\theta}(t-k)||^2)$$

then the result follows immediately from (5) since k is infinite.

(e) Equation (7) follows immediately from (6).

Appendix B

Mathematical derivation of the Coupled Electric Drives Model [62]

To obtain the model the system is divided into three sub-systems. It is the force, torque and power balancing in these sub-systems that lead to the model derivation.

Using the free body diagram, Figure 5.3, each sub-system is now considered, starting with:

(a) The jockey pulley assembly: The pulley is assumed to be light and rotating frictionless, such that $F_1 = F_2 = F$. Vertical resolution of forces yields:

$$F_k = 2 \cos \alpha F \tag{1}$$

using law of conservation of power

$$\dot{x}_k F_k = F(v_1 - v_2) \tag{2}$$

where v_1, v_2 are velocities. This implies

$$\dot{x}_k(2 \cos \alpha) = v_1 - v_2 \tag{3}$$

A force balance on the tension measuring assembly gives:

$$F_k = m_t \ddot{x}_k + k_t x_k + b_t \dot{x}_k \tag{4}$$

Note: $m_t \dot{x}_k = p$ the momemtum of mass m_t .

(b) The drive pulleys a and b : A torque balance of the drive pulleys gives:

$$\tau_a + F_3 r_a - F_5 r_a = \dot{h}_a + b_a w_a \tag{5}$$

$$\tau_b + F_6 r_b - F_4 r_b = \dot{h}_b + b_b w_b \tag{6}$$

where h_a and h_b are the drive pulley/motor angular moments,

with $h_a = I_a w_a, \quad h_b = I_b w_b$ (7)

$$v_3 = w_a r_a, \quad v_b = r_b w_b \tag{8}$$

(c) The belt sections: A force balance on the different belt section gives:

$$\left.\begin{aligned} F_1 = F_2 = F_3 = F_4 = F = K_c x_c = K_d x_d \\ F' = F_5 = F_6 = K_e x_e \end{aligned}\right\} \tag{9}$$

with x_c, x_d, x_e the respective belt section lengths.

Using equations (1)-(9) the state space equation matrix can be derived with the states taken as h_a, h_b, x_c, x_e, x_k and p, thus one obtains:

$$\left.\begin{aligned} \dot{h}_a &= -\frac{b_a}{I_a} h_a + r_a K_c x_c - r_a K_e x_e + \tau_a \\ \dot{h}_b &= -\frac{b_b}{I_b} h_b - r_b K_c x_c + r_b K_e x_e + \tau_b \\ \dot{x}_c &= \left(1 + \frac{K_c}{K_d}\right)^{-1} \left(-h_a \frac{r_a}{I_a} + h_b \frac{r_b}{I_b} + p \frac{2\cos\alpha}{m_t}\right. \\ \dot{x}_e &= \frac{r_a}{I_a} h_a - \frac{r_b}{I_b} h_b \\ \dot{x}_k &= p/m_t \\ \dot{p} &= 2 \cos \alpha . K_c x_c - K_t x_t - \frac{b_t}{m_t} p \end{aligned}\right\} \tag{10}$$

From the above state space equations set, the transfer functions for the coupled drives system can be derived. To simplify the above state space equations some assumptions need to be made, i.e.

$$I_a = I_b = I$$

$$b_a = b_b = b$$

Also that the pulley radii are the same, and that the belt sections are of the same length, thus:

$$r_a = r_b = r$$

$$K_c = K_d = K_e = K$$

Substituting these into the state equations and adding the angular momentum equations gives:

$$\dot{h}_a + \dot{h}_b = -\frac{b}{I}(h_a+h_b) + \tau_a + \tau_b \qquad (11)$$

$$\omega_a(s) + \omega_b(s) = \frac{1}{sI+b}(\tau_a(s) + \tau_b(s)) \qquad (12)$$

Manipulating the state equations gives:

$$v_c(s) = \left[\frac{s^2m_t + sb_t + K_t}{2[s^2m_t+sb_t+K_t]-(2\cos\alpha)^2K}\right] r(w_b(s) - w_a(s)) \qquad (13)$$

Also the angular velocities of the drive pulleys expressed in terms of the input torques, yield:

$$\omega_a(s) = \left[\frac{s^2I + sb + K'(s)}{(sI+b)(s^2I+sb+2K'(s))}\right]\tau_a(s) + \left[\frac{K'(s)}{(sI+b)(s^2I+sb+2K'(s))}\right]\tau_b(s)$$

(14)

$$\omega_b(s) = \left[\frac{s^2I + sb + K'(s)}{(sI+b)(s^2I+sb+2K'(s))}\right]\tau_b(s) + \left[\frac{K'(s)}{(sI+b)(s^2I+sb+2K'(s))} - \right]\tau_a(s)$$

(15)

where $K'(s) = Kr^2(G_1(s) + 1)$ (16)

with $G_1(s) = \dfrac{s^2m_t + sb_t + K_t}{2[s^2m_t+sb_t+K_t]-(2\cos\alpha)^2K}$ (17)

Subtracting (14) from (15):

$$\omega_b(s) - \omega_a(s) = \frac{s}{s^2I+sb+2K'(s)}\,(\tau_b(s) - \tau_a(s)) \qquad (18)$$

The angular velocity ω of the jockey pulley can be expressed as a function of the drive pulley angular velocities by manipulations resulting in:

$$\omega(s) = \omega_a(s) + G_1(s)[\omega_b(s)-\omega_a(s)] - \frac{sx_k(s)\cos\alpha}{r} \qquad (19)$$

or

$$\omega(s) = \omega_b(s) + G_1(s)[\omega_b(s)-\omega_a(s)] + \frac{sx_k(s)\cos\alpha}{r} \qquad (2\ )$$

Adding (19) and (20) gives:

$$\omega(s) = \frac{1}{2}[\omega_a(s) + \omega_b(s)] \qquad (21)$$

Substituting for $\omega_a(s) + \omega_b(s)$ in the above, using (12), gives:

$$\omega(s) = \frac{\tau_a + \tau_b}{2(sI+b)} \qquad (22)$$

From the state space equations the transfer function relating the displacement of the jockey arm to the input drive torques is obtained as:

$$x_k(s) = \frac{2rK\cos\alpha(\tau_b(s)-\tau_a(s))}{(2(s^2m_t+sb_t+K_t)-(2\cos\alpha)^2K)(s^2I+bs+2K'(s))} \qquad (23)$$

As mentioned in Chapter 5, equations (22) and (23) show the interactions between the two torque inputs and the outputs. Using equation (5.3.3), i.e.

$$v_a(s) = u_1(s) - u_2(s)$$
$$v_b(s) = u_1(s) + u_2(s)$$

to decouple and substituting into (22) and (23), and also using (5.3.1) and (5.3.2), one obtains:

$$\omega(s) = \frac{g_m\, u_1(s)}{sI + b} \qquad (24)$$

$$x_k(s) = \frac{4g_m rK\cos\alpha u_2(s)}{(s^2I+sb+2K'(s)(2[s^2m_t+sb_t+K_t]-(2\cos\alpha)^2K)}$$

where $g_a = g_b = g_m$.

A P P E N D I X C

```
>
    5 ***TYPICAL SISO GOODWIN ET-AL ALGORITHM***
   10 DIM X(6)
   20 DIM Y(21)
   30 INPUT"ENTER I.C'S Y(4),YS,Y(6), Y(11),Y(12),Y(13)";
   32 (30contd)Y(4),YS,Y(6),Y(11),Y(12),Y(13)
   40 R=4
   70 Y(7)=Y(6)
  110 Y(9)=YS-Y(4)
  130 Y(5)=YS
  140 Y(8)=Y(5)
  160 Y(10)=Y(9)
  180 Y(14)=Y(11)
  200 Y(15)=Y(12)
  220 Y(16)=Y(13)
  230 Y(17)=Y(7)
  250 Y(18)=Y(8)
  270 Y(19)=Y(4)
  280 count=1
  285 Z=OPENOUT "DAT1"
  290 FOR K=1 TO 30075
  300 Y(4)=(Y(19)*EXP(-1.5))+(R*0.7768698)
  310 Y(5)=YS
  325 Ts=3.0
  327 K=75=Ts
  330 IF K=(count*75) THEN GOTO 350
  340 NEXT
  350 PRINT:PRINT "time=";K
  360 PRINT
  370 Y(9)=YS-Y(4)
  380 PRINT"Y(4)=";Y(4)
  390 X(1)=Y(6)
  410 Y(7)=X(1)/255
  420 Y(20)=1/(1+(Y(17)*Y(17)+Y(18)*Y(18)+Y(19)*Y(19)))
  430 Y(11)=Y(14)-29.0*(-Y(17))*Y(20)*Y(9)
  435 PRINT"Y(11)=";Y(11)
  437 YO1=Y(11)/255
  440 Y(12)=Y(15)-29.0*(-Y(18))*Y(20)*Y(9)
  445 PRINT"Y(12)=";Y(12)
  447 YO2=Y(12)/255
  450 Y(13)=Y(16)-29.0*Y(19)*Y(20)*Y(9)
  455 PRINT"Y(13)=";Y(13)
  457 YO3=Y(13)/255
  460 Y(21)=((-Y(8))*Y(11)+(-Y(7))*Y(12)+Y(4)*Y(13))
  470 Y(6)=Y(21)
  480 Y(17)=Y(7)
  490 Y(18)=Y(8)
  500 Y(10)=Y(9)
  510 Y(14)=Y(11)
  520 Y(15)=Y(12)
  530 Y(16)=Y(13)
  540 Y(19)=Y(4)
  550 Y(8)=Y(5)
  551 SX=Y(6)/255
  553 PRINT£Z,Y(4),YS,SX,YO1,YO2,YO3
  560 count=count+1
  570 Yin=&FCCO
  580 Adrs=Yin
  590 ?Adrs=0:
```

```
600 X=?Adrs
610 YS=10*X/255
620 PRINT "YS=";YS
630 Es=YS-Y(4)
640 PRINT"Es=";Es
650 PRINT"Y(7)=";Y(7)
660 Si=Y(6)
670 PRINT"Si=";Si
680 PROCDA
690 NEXT
695 CLOSE £Z
700 END
710REM
720DEF PROCDA
730Uout=&FCC4
740Adrs=Uout
750Sid=Si
760?Adrs=Sid
770ENDPROC
```

```
>
    5 ***TYPICAL ORTEGA ET-AL ALGORITHM***
   10 DIM X(6)
   20 DIM Y(25)
   30 INPUT"ENTER I.C'S Y(4),YS,Y(6), Y(11),Y(12),Y(13),X(2)";
   32 (30contd)Y(4),YS,Y(6),Y(11),Y(12),Y(13),X(2)
   40 R=3
   70 Y(7)=Y(6)
  110 Y(22)=YS-Y(4)
  120 X(4)=X(2)
  130 Y(5)=YS
  140 Y(8)=Y(5)
  160 Y(10)=Y(22)
  180 Y(14)=Y(11)
  200 Y(15)=Y(12)
  220 Y(16)=Y(13)
  230 Y(17)=Y(7)
  250 Y(18)=Y(8)
  270 Y(19)=Y(4)
  280 count=1
  285 Z=OPENOUT"DAT2"
  290 FOR K=1 TO 1000000
  300 Y(4)=(Y(19)*EXP(-1.5))+(R*0.7768698)
  310 Y(5)=YS
  325 Ts=3.0
  327 K=75=Ts
  330 IF K=(count*75) THEN GOTO 350
  340 NEXT
  350 PRINT:PRINT "time=";K
  360 PRINT
  370 Y(22)=YS-Y(4)
  380 PRINT"Y(4)=";Y(4)
  390 X(1)=Y(6)
  410 Y(7)=X(1)/255
  420 Y(20)=Y(17)*Y(17)+Y(8)*Y(8)+Y(7)*Y(7)
  421 IF Y(20)>5.0 THEN Y(21)=Y(20)
  422IF  Y(20)<5.0 THEN Y(21)=5.0
  423 PRINT"Y(21)=";Y(21)
  424 X(2)=X(4)*0.75 + Y(21)
  426 Y(9)=Y(22)/X(2)
  430 Y(11)=Y(14)+9.0*Y(8)*Y(9)
  435 PRINT"Y(11)=";Y(11)
  437 YO1=Y(11)/255
  440 Y(12)=Y(15)+9.0*Y(7)*Y(9)
  445 PRINT"Y(12)=";Y(12)
  447 YO2=Y(12)/255
  450 Y(13)=Y(16)+9.0*Y(17)*Y(9)
  455 PRINT"Y(13)=";Y(13)
  457 YO3=Y(13)/255
  460 Y(6)=(Y(4)-(Y(11)*Y(15)+Y(13)*Y(7)))/Y(12)
  480 Y(17)=Y(7)
  490 Y(18)=Y(8)
  500 Y(10)=Y(9)
  510 Y(14)=Y(11)
  520 Y(15)=Y(12)
  530 Y(16)=Y(13)
  540 Y(19)=Y(4)
```

```
550 Y(8)=Y(5)
551 SX=Y(6)/255
553 PRINT £Z,Y(4),YS,SX,YO1,YO2,YO3
560 count=count+1
570 Yin=&FCCO
580 Adrs=Yin
590 ?Adrs=0:
600 X=?Adrs
610 YS=10*X/255
620 PRINT "YS=";YS
630 Es=YS-Y(4)
640 PRINT"Es=";Es
650 PRINT"Y(7)=";Y(7)
660 Si=Y(6)
670 PRINT"Si=";Si
680 PROCDA
690 NEXT
695 CLOSE £Z
700 END
710REM
720DEF PROCDA
730Uout=&FCC4
740Adrs=Uout
750Sid=Si
760?Adrs=Sid
770ENDPROC
```

```
>L.
    5 ***TYPICAL KREISSELMEIER & ANDERSON ALGORITHM***
   10 DIM X(6)
   20 DIM Y(25)
   30 INPUT"ENTER I.C'S Y(4),YS,Y(6), Y(11),Y(12),Y(13),X(2)";
   32 (30contd)Y(4),YS,Y(6),Y(11),Y(12),Y(13),X(2)
   40 R=3
   50 Y(4)=Y(4)
   60 Y(6)=Y(6)
   70 Y(7)=Y(6)
   80 Y(11)=Y(11)
   90 Y(12)=Y(12)
  100 Y(13)=Y(13)
  110 Y(9)=YS-Y(4)
  120 Y(6)=Y(6)
  125 X(4)=X(2)
  128 X(3)=X(2)+1.0
  130 Y(5)=YS
  140 Y(8)=Y(5)
  150 Y(9)=Y(9)
  160 Y(10)=Y(9)
  170 Y(11)=Y(11)
  180 Y(14)=Y(11)
  190 Y(12)=Y(12)
  200 Y(15)=Y(12)
  210 Y(13)=Y(13)
  220 Y(16)=Y(13)
  230 Y(17)=Y(7)
  240 Y(8)=Y(8)
  250 Y(18)=Y(8)
  260 Y(4)=Y(4)
  270 Y(19)=Y(4)
  280 count=1
  285 Z=OPENOUT"DAT3"
  290 FOR K=1 TO 45100
  300 Y(4)=(Y(23)*EXP(-1.5))+(R*0.7768698)
  310 Y(5)=YS
  320 Y(9)=YS-Y(4)
  325 TS=4.0
  327 K=100=TS
  330 IF K=(count*100) THEN GOTO 350
  340 NEXT
  350 PRINT:PRINT "time=";K
  360 PRINT
  362 X(2)=X(4)*0.75 + ((Y(6)+2797)/1275)+Y(8)
  364 X(3)=X(2)+1.0
  370 Y(9)=Y(22)/X(3)
  371 IF ABS(Y(9))<=0.0061 THEN Y(9)=0
  372 IF Y(9)>0.0061 THEN Y(9)=Y(9)-0.0061
  373 IF Y(9)<0.0061 THEN Y(9)=Y(9)+0.0061
  380 PRINT"Y(4)=";Y(4)
  385 Y(19)=1.2873*(Y(8)-0.2231*Y(18))
  387 PRINT"Y(19)=";Y(19)
  390 X(1)=Y(6)
  395 PRINT"X(1)=";X(1)
  410 Y(7)=X(1)/255
  420 Y(20)=1/(1+(Y(17)*Y(17)+Y(18)*Y(18)+Y(19)*Y(19)))
  430 Y(11)=Y(14)-Y(19)*Y(20)*Y(9)*X(3)
```

```
431 IF Y(11)<70.00 THEN Y(11)=70.00
432 IF Y(11)>85.00 THEN Y(11)=85.00
433 IF Y(11)<=85.00 OR Y(11)>=70.00 THEN Y(11)=Y(11)
434 YO1=Y(11)/255
435 IF Y(11)<=85.00 OR Y(11)>=70.00 THEN Y(11)=Y(11)
436 PRINT"Y(11)=";Y(11)
440 Y(12)=Y(15)-Y(17)*Y(20)*Y(9)*X(3)
441 IF Y(12)<70.00 THEN Y(12)=70.00
442 IF Y(12)>85.00 THEN Y(12)=85.00
443 IF Y(12)<=85.00 OR Y(12)>=70.00 THEN Y(12)=Y(12)
444 YO2=Y(12)/255
445 IF Y(12)<=85.00 OR Y(12)>=70.00 THEN Y(12)=Y(12)
446 PRINT"Y(12)=";Y(12)
450 Y(13)=Y(16)-Y(18)*Y(20)*Y(9)*X(3)
451 IF Y(13)<70.00 THEN Y(13)=70.00
452 IF Y(13)>85.00 THEN Y(13)=85.00
453 IF Y(13)<=85.00 OR Y(13)>=70.00 THEN Y(13)=Y(13)
454 YO3=Y(13)/255
455 IF Y(13)<=85.00 OR Y(13)>=70.00 THEN Y(13)=Y(13)
456 PRINT"Y(13)=";Y(13)
460 Y(21)=(Y(8)*Y(11)+Y(7)*Y(12)+Y(4)*Y(13))
466 Y(10)=(Y(19)*Y(11)+Y(17)*Y(12)+Y(18)*Y(13))
468 Y(22)=(Y(10)-X(1))/5
469 PRINT"Y(22)=";Y(22)
470 Y(6)=Y(21)
480 Y(17)=Y(7)
490 Y(18)=Y(8)
500 X(4)=X(2)
510 Y(14)=Y(11)
520 Y(15)=Y(12)
530 Y(16)=Y(13)
540 Y(23)=Y(4)
550 Y(8)=Y(5)
551 SX=Y(6)/255
553 PRINT&Z,Y(4),YS,SX,Y(11),Y(12),Y(13)
560 count=count+1
570 Yin=&FCC0
580 Adrs=Yin
590 ?Adrs=0:
600 X=?Adrs
610 YS=10*X/255
620 PRINT "YS=";YS
630 Es=YS-Y(4)
640 PRINT"Es=";Es
650 PRINT"Y(7)=";Y(7)
660 Si=Y(6)
670 PRINT"Si=";Si
680 PROCDA
690 NEXT
700 END
710REM
720DEF PROCDA
730Uout=&FCC4
740Adrs=Uout
750Sid=Si
760?Adrs=Sid
770ENDPROC
```

```
>
    5 ***TYPICAL MIMO GOODWIN ET-AL ALGORITHM***
   10 DIM X(100)
   20 DIM Y(300)
   30 INPUT"ENTER I.C'S Y(9),Y(47),Y(1),Y(11),Y(12),Y(13),Y(14),Y(15),Y(16
   32 (30 contd)Y(9),Y(47),Y(1),Y(11),Y(12),Y(13),Y(14),Y(15),Y(16)
   35 INPUT"ENTER I.C'S Y(17),Y(48),Y(3),Y(21),Y(22),Y(23),Y(24),Y(25),Y(2
   37 (35contd)Y(17),Y(48),Y(3),Y(21),Y(22),Y(23),Y(24),Y(25),Y(26)
   40 R=3
   45 RI=3
   50 Y(10)=Y(9)
   51 Y(18)=Y(17)
   52 Y(31)=Y(11)
   53 Y(32)=Y(12)
   54 Y(33)=Y(13)
   55 Y(34)=Y(14)
   56 Y(35)=Y(15)
   57 Y(36)=Y(16)
   58 Y(2)=Y(1)
   59 Y(4)=Y(3)
   60 Y(41)=Y(21)
   61 Y(42)=Y(22)
   62 Y(43)=Y(23)
   63 Y(44)=Y(24)
   64 Y(45)=Y(25)
   65 Y(46)=Y(26)
   66 Y(5)=Y(47)
   67 Y(6)=Y(5)
   68 Y(7)=Y(48)
   69 Y(8)=Y(7)
   70 Y(27)=9.0
   71 Y(38)=9.0
   80 count=1
   85 Z=OPENOUT"DAT1"
   90 FOR K=1 TO 1000000
  100 Y(9)=(Y(10)*EXP(-1.5))+R*0.7768698
  110 Y(17)=(Y(18)*EXP(-1.5))+RI*0.7768698
  120 Y(2)=Y(1)
  125 Y(4)=Y(3)
  140 Ts=3.0
  145 K=75=Ts
  150 IF K=(count*75) THEN GOTO 170
  160 NEXT
  170 PRINT:PRINT"time=";K
  180 PRINT
  185 PRINT"Y(9)=";Y(9),TAB(20),"Y(17)=";Y(17)
  190 Y(19)=Y(1)-Y(9)
  200 Y(20)=Y(3)-Y(17)
  210 X(1)=Y(47)
  220 X(11)=Y(48)
  227 Y(5)=X(1)/255
  229 Y(7)=X(11)/255
  230 Y(29)=1/(1+(Y(2)*Y(2)+Y(4)*Y(4)+Y(6)*Y(6)+Y(8)*Y(8)+
  235 (230contd)Y(10)*Y(10)+Y(18)*Y(18)))
  240 Y(11)=Y(31)-Y(27)*Y(29)*Y(19)*(-Y(2))
  245 Y(21)=Y(41)-Y(38)*Y(29)*Y(20)*(-Y(2))
  247 PRINT "Y(11)=";Y(11),TAB(20),"Y(21)=";Y(21)
  250 Y(12)=Y(32)-Y(27)*Y(29)*Y(19)*(-Y(4))
```

```
255 Y(22)=Y(42)-Y(38)*Y(29)*Y(20)*(-Y(4))
257 PRINT "Y(12)=";Y(12),TAB(20),"Y(22)=";Y(22)
260 Y(13)=Y(33)-Y(27)*Y(29)*Y(19)*(-Y(6))
265 Y(23)=Y(43)-Y(38)*Y(29)*Y(20)*(-Y(6))
267 PRINT "Y(13)=";Y(13),TAB(20),"Y(23)=";Y(23)
270 Y(14)=Y(34)-Y(27)*Y(29)*Y(19)*(-Y(8))
275 Y(24)=Y(44)-Y(38)*Y(29)*Y(20)*(-Y(8))
277 PRINT "Y(14)=";Y(14),TAB(20),"Y(24)=";Y(24)
280 Y(15)=Y(35)-Y(27)*Y(29)*Y(19)*Y(10)
285 Y(25)=Y(45)-Y(38)*Y(29)*Y(20)*Y(10)
287 PRINT "Y(15)=";Y(15),TAB(20),"Y(25)=";Y(25)
290 Y(16)=Y(36)-Y(27)*Y(29)*Y(19)*Y(18)
295 Y(26)=Y(46)-Y(38)*Y(29)*Y(20)*Y(18)
297 PRINT "Y(16)=";Y(16),TAB(20),"Y(26)=";Y(26)
300 Y(47)=((-Y(1)*Y(11))+(-Y(3)*Y(12))+(-Y(5)*Y(13))+
305 (300contd) (-Y(7)*Y(14))+Y(15)*Y(9)+Y(16)*Y(17))
320 Y(48)=((-Y(1)*Y(21))+(-Y(3)*Y(22))+(-Y(5)*Y(23))+
325 (320contd) (-Y(7)*Y(24))+Y(25)*Y(9)+Y(26)*Y(17))
353 Y(10)=Y(9)
356 Y(18)=Y(17)
360 Y(31)=Y(11)
362 Y(32)=Y(12)
364 Y(33)=Y(13)
366 Y(34)=Y(14)
368 Y(35)=Y(15)
370 Y(36)=Y(16)
372 Y(41)=Y(21)
374 Y(42)=Y(22)
376 Y(43)=Y(23)
378 Y(44)=Y(24)
380 Y(45)=Y(25)
382 Y(46)=Y(26)
384 Y(2)=Y(1)
386 Y(4)=Y(3)
388 Y(8)=Y(7)
390 Y(6)=Y(5)
393 SX1=Y(47)
395 SX2=Y(48)
397 PRINT £Z,Y(9),Y(17),Y(1),Y(3),SX1,SX2
400 count=count+1
410 Yin=&FCC0
415 Adrs=Yin
420 ?Adrs=0:
425 X=?Adrs
430 Y(1)=10*X/255
450 Yin=&FCC1
455 Adrs=Yin
460 ?Adrs=0:
465 X=?Adrs
470 Y(3)=10*X/255
475 PRINT"Y(1)=";Y(1),TAB(20),"Y(3)=";Y(3)
480 Es1=Y(1)-Y(9)
485 Es2=Y(3)-Y(17)
490 PRINT"Es1=";Es1,TAB(20),"Es2=";Es2
510 Si1=Y(47)
515 Si2=Y(48)
517 PRINT"Si1=";Si1,TAB(20),"Si2=";Si2
```

```
520 PROCDA1
530 PROCDA2
540 NEXT
545 CLOSE £Z
550 END
560REM
570DEF PROCDA1
580Uout=&FCC4
590Adrs=Uout
600 Sid=Si1
610?Adrs=Sid
620ENDPROC
630REM
640DEF PROCDA2
650Uout=&FCC5
660Adrs=Uout
670 Sid=Si2
680?Adrs=Sid
690ENDPROC
```

```
>L.
    5 ***TYPICAL EXTENDED MIMO KREISSELMEIER & ANDERSON ALGORITHM
   10 DIM X(100)
   20 DIM Y(300)
   30 INPUT"ENTER I.C'S Y(9),Y(47),Y(1),Y(11),Y(13),Y(14),Y(15),Y(16),
   31 (30contd)Y(51),Y(52)";Y(9),Y(47),Y(1),Y(11),Y(13),Y(14),Y(15),Y(16),
   32 (31contd)Y(51),Y(52)
   35 INPUT"ENTER I.C'S Y(17),Y(48),Y(3),Y(22),Y(23),Y(24),Y(25),Y(26),
   36 (35contd)Y(61),Y(62)";Y(17),Y(48),Y(3),Y(22),Y(23),Y(24),Y(25),Y(26),
   37 (36contd)Y(61),Y(62)
   40 R=4
   45 RI=3
   50 Y(10)=Y(9)
   51 Y(18)=Y(17)
   52 Y(31)=Y(11)
   53 Y(32)=Y(12)
   54 Y(33)=Y(13)
   55 Y(34)=Y(14)
   56 Y(35)=Y(15)
   57 Y(36)=Y(16)
   58 Y(2)=Y(1)
   59 Y(4)=Y(3)
   60 Y(41)=Y(21)
   61 Y(42)=Y(22)
   62 Y(43)=Y(23)
   63 Y(44)=Y(24)
   64 Y(45)=Y(25)
   65 Y(46)=Y(26)
   66 Y(5)=Y(47)
   67 Y(6)=Y(5)
   68 Y(7)=Y(48)
   69 Y(8)=Y(7)
   72 Y(65)=Y(10)
   73 Y(66)=Y(18)
   74 Y(49)=Y(2)
   75 Y(50)=Y(4)
   76 Y(53)=Y(51)
   77 Y(54)=Y(52)
   78 Y(63)=Y(61)
   79 Y(64)=Y(62)
   80 count=1
   85 Z=OPENOUT"DAT2"
   90 FOR K=1 TO 30075
  100 Y(9)=Y(10)*EXP(-1.5)+R*0.7768698
  110 Y(17)=Y(18)*EXP(-1.5)+R*0.7768698
  120 Y(2)=Y(1)
  125 Y(4)=Y(3)
  140 Ts=3.0
  145 K=75=Ts
  150 IF K=(count*75) THEN GOTO 170
  160 NEXT
  170 PRINT:PRINT"time=";K
  180 PRINT
  185 PRINT"Y(9)=";Y(9),TAB(20),"Y(17)=";Y(17)
  190 Y(19)=Y(1)-Y(9)
  200 Y(20)=Y(3)-Y(17)
  201 X(2)=X(4)*0.75 +(Y(47)/255)+Y(2)
  202 X(5)=X(6)*0.75+(Y(48)/255)+Y(4)
  203 X(3)=1.0 + X(2)
```

```
204 X(7)=1.0 + X(5)
207 Y(63)=Y(61)/X(3)
208 Y(64)=Y(62)/X(7)
209 IF ABS(Y(63))<=0.05 THEN Y(63)=0.0
210 IF ABS(Y(64))<=0.05 THEN Y(64)=0.0
211 IF Y(63)>=0.05 THEN Y(63)=Y(63)-0.05
212 IF Y(64)>=0.05 THEN Y(64)=Y(64)-0.05
213 IF Y(63)<=-0.05 THEN  Y(63)=Y(63)+0.05
214 IF Y(64)<=-0.05 THEN Y(64)=Y(64)+0.05
215 Y(51)=1.2873*(Y(2)-0.2231*Y(55))
216 Y(52)=1.2873*(Y(2)-0.2231*Y(55))
217 PRINT"Y(63)=";Y(63),TAB(20),"Y(64)=";Y(64)
218 X(1)=Y(47)
220 X(11)=Y(48)
227 Y(5)=X(1)/255
229 Y(7)=X(11)/255
230 Y(29)=1/(1+(Y(2)*Y(2)+Y(4)*Y(4)+Y(49)*Y(49)+Y(50)*Y(50)+
235 (230contd) Y(6)*Y(6)+Y(8)*Y(8)+Y(10)*Y(10)+Y(18)*Y(18)))
240 Y(11)=Y(31)-Y(27)*Y(29)*Y(19)*(-Y(2))
242 Y(21)=0.0
245 PRINT "Y(11)=";Y(11),TAB(20),"Y(21)=";Y(21)
246 Y(12)=0.0
248 Y(22)=Y(42)-Y(38)*Y(29)*Y(20)*(-Y(4))
249 PRINT "Y(12)=";Y(12),TAB(20),"Y(22)=";Y(22)
250 Y(51)=Y(53)-Y(27)*Y(29)*Y(19)*(-Y(49))
252 Y(61)=Y(63)-Y(38)*Y(29)*Y(20)*(-Y(49))
255 PRINT"Y(51)=";Y(51),TAB(20),"Y(61)=";Y(61)
256 Y(52)=Y(54)-Y(27)*Y(29)*Y(19)*(-Y(50))
258 Y(62)=Y(64)-Y(38)*Y(29)*Y(20)*(-Y(50))
259 PRINT"Y(52)=";Y(52),TAB(20),"Y(62)=";Y(62)
260 Y(13)=Y(33)-Y(27)*Y(29)*Y(19)*(-Y(6))
265 Y(23)=Y(43)-Y(38)*Y(29)*Y(20)*(-Y(6))
267 PRINT "Y(13)=";Y(13),TAB(20),"Y(23)=";Y(23)
270 Y(14)=Y(34)-Y(27)*Y(29)*Y(19)*(-Y(8))
275 Y(24)=Y(44)-Y(38)*Y(29)*Y(20)*(-Y(8))
277 PRINT "Y(14)=";Y(14),TAB(20),"Y(24)=";Y(24)
280 Y(15)=Y(35)-Y(27)*Y(29)*Y(19)*Y(10)
285 Y(25)=Y(45)-Y(38)*Y(29)*Y(20)*Y(10)
287 PRINT "Y(15)=";Y(15),TAB(20),"Y(25)=";Y(25)
290 Y(16)=Y(36)-Y(27)*Y(29)*Y(19)*Y(18)
295 Y(26)=Y(46)-Y(38)*Y(29)*Y(20)*Y(18)
297 PRINT "Y(16)=";Y(16),TAB(20),"Y(26)=";Y(26)
300 Y(47)=((-Y(1)*Y(11))+(-Y(2)*Y(51))+(-Y(4)*Y(52))+(-Y(5)*Y(13))+
305 (300contd) (-Y(7)*Y(14))+Y(15)*Y(9)+Y(16)*Y(17))
320 Y(48)=(-Y(3)*Y(22))+(-Y(2)*Y(61))+(-Y(4)*Y(62))+(-Y(5)*Y(23))+
325 (320contd) (-Y(7)*Y(24))+Y(25)*Y(9)+Y(26)*Y(17))
342 Y(61)=Y(59)-X(1)
344 Y(62)=Y(60)-X(11)
351 Y(65)=Y(10)
352 Y(66)=Y(18)
353 Y(10)=Y(9)
356 Y(18)=Y(17)
360 Y(31)=Y(11)
362 Y(32)=Y(12)
363 Y(53)=Y(51)
364 Y(54)=Y(52)
365 Y(33)=Y(13)
```

```
366 Y(34)=Y(14)
368 Y(35)=Y(15)
370 Y(36)=Y(16)
372 Y(41)=Y(21)
374 Y(42)=Y(22)
375 Y(63)=Y(61)
376 Y(64)=Y(62)
377 Y(43)=Y(23)
378 Y(44)=Y(24)
380 Y(45)=Y(25)
382 Y(46)=Y(26)
383 Y(49)=Y(2)
384 Y(2)=Y(1)
385 Y(50)=Y(4)
386 Y(4)=Y(3)
388 Y(8)=Y(7)
390 Y(6)=Y(5)
393 SX1=Y(47)/255
394 SX2=Y(48)/255
396 PRINT £Z,Y(9),Y(17),Y(1),Y(3),SX1,SX2
400 count=count+1
410 Yin=&FCC0
415 Adrs=Yin
420 ?Adrs=0:
425 X=?Adrs
430 Y(1)=10*X/255
450 Yin=&FCC1
455 Adrs=Yin
460 ?Adrs=0:
465 X=?Adrs
470 Y(3)=10*X/255
475 PRINT"Y(1)=";Y(1),TAB(20),"Y(3)=";Y(3)
480 Es1=Y(1)-Y(9)
485 Es2=Y(3)-Y(17)
490 PRINT"Es1=";Es1,TAB(20),"Es2=";Es2
510 Si1=Y(47)
515 Si2=Y(48)
517 PRINT"Si1=";Si1,TAB(20),"Si2=";Si2
520 PROCDA1
530 PROCDA2
540 NEXT
545 CLOSE £Z
550 END
560REM
570DEF PROCDA1
580Uout=&FCC4
590Adrs=Uout
600 Sid=Si1
610?Adrs=Sid
620ENDPROC
630REM
640DEF PROCDA2
650Uout=&FCC5
660Adrs=Uout
670 Sid=Si2
680?Adrs=Sid
690ENDPROC
```