NASA Technical Memorandum 85737

AN ON-LINE EQUIVALENT SYSTEM IDENTIFICATION SCHEME FOR ADAPTIVE CONTROL

Steven M. Sliwa

January 1984

NASA
National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

AN ON-LINE EQUIVALENT SYSTEM IDENTIFICATION SCHEME FOR ADAPTIVE CONTROL

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF AERONAUTICS AND ASTRONAUTICS

AND THE COMMITTEE ON GRADUATE STUDIES

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

By

Steven Mark Sliwa

January 1984

ABSTRACT

A prime obstacle to the widespread use of adaptive control is the degradation of performance and possible instability resulting from the presence of unmodeled dynamics. The approach taken is to explicitly include the unstructured model uncertainty in the output error identification algorithm. The order of the compensator is successively increased by including newly identified modes. During this model building stage, heuristic rules are used to test for convergence prior to designing new compensators. Additionally, the recursive identification algorithm has been extended to multi-input, multi-output systems. Enhancements were also made to reduce the computational burden of an algorithm for obtaining minimal state space realizations from the inexact, multivariable transfer functions which result from the identification process. A number of potential adaptive control applications for this approach are illustrated using computer simulations. Results indicated that when speed of adaptation and plant stability are not critical, the proposed schemes converge to enhance system performance.

ACKNOWLEDGEMENTS

I was extremely fortunate to have had Professors Dan DeBra and Arthur E. Bryson, Jr., as my co-advisors. This research was accomplished primarily through their technical guidance, their alacrity to discuss philosophical issues, and their encouragement when the work slowed and motivation waned. More importantly, I am grateful to have had the opportunity to interact so closely with both of them as I am sure they are examples which will guide my life.

I was sponsored by the Flight Dynamics and Control Division of NASA Langley Research Center largely through the efforts of my division chief, Dr. Willard W. Anderson, who was a major advocate throughout my program. My branch head, Jarrell R. Elliott, provided me with the computer resources, unhampered working environment and moral support to complete my research in a timely manner. Additionally, his interest and questions helped provoke a great deal of insight. Lawrence W. Taylor, Jr., played a significant role as mentor and often had suggestions which lead to advances in my work on this and other projects. Fortunately, Virginia B. Marks was the skilled typist who supervised the construction of this document.

Since this effort represents the culmination of my formal training, it seems appropriate to thank some of the outstanding teachers I have had along the way. In particular, I would like to express my gratitude to: Margaret Ridell Davis (6th grade); Larry Josbeno (high school); David Hazen and Edward Seckel (Princeton); and, of course, Professors Bryson and DeBra (Stanford). I could have easily been dissuaded anywhere along my path had it not been for these individuals.

The friendships of numerous individuals were important during the course of this research as distractions which helped make life enjoyable. In particular, I would like to thank Kimberly Anderson, Maureen Doherty, Ken Flaig, Gail Jonkouski, Louis Lerman, Tracy Looney, Daniel Neal, Jane Neal, and Scott Pelton.

ACKNOWLEDGEMENTS (Cont.)

My officemate and colleague, P. Douglas Arbuckle, was instrumental in shielding me from distractions at work and supportive during the mood swings that often accompany such research. I am also extremely grateful for my association with Bradley L. Schrick, my friend and partner. In the very beginning he drove with me out to California; took many of the control system courses with me; willingly helped me with all aspects of my work; made several east coast visits (progress checks); and, in general, provided an anchor to insure sanity for the past three years.

The entire effort was made possible through the support and love of my parents, Kenneth and Shirley Sliwa. I am what I am today largely through their efforts in my behalf; and, it is in their honor that this paper is dedicated.

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Cont.)

LIST OF FIGURES

LIST OF FIGURES (Cont.)

LIST OF FIGURES (Cont.)

LIST OF FIGURES (Cont.)

LIST OF TABLES

LIST OF SYMBOLS

a	coefficient of characteristic equation
A	state space dynamics matrix
b	element of control influence matrix
B	state space control influence matrix
B_i	numerator polynomial matrix for s^i
C	ch. IV, numerator term for $(sI-A)^{-1}$
C	ch. V, regulator gain matrix
d(s)	denominator polynomial matrix
e	input signal
e_p	summation of the percentage parameter errors
E	expected value
F	feedforward gain matrix
F_1, F_2	feedforward gain matrices
h_{ij}	element of output distribution matrix
H	state space output distribution matrix
H(z)	discrete transfer function
I	identity matrix
J	cost function
k	time step for difference equation
k_1, k_2	spring constants, lbs/in
K	filter gain matrix
L	control-to-output distribution matrix
L_i	lower limit of allowable range for ith parameter
m	number of inputs
m_1, m_2, m_3	masses of spring-mass-damper example
M	gain magnitude in dB
M(s)	modified numerator polynomial matrix
n	system order
n'	order of numerator polynomial
N	number of samples
N(s)	numerator polynomial matrix

LIST OF SYMBOLS (Cont.)

p	section II-f, order of incremental mode
p	parameter
P	covariance matrix defined by equation (2.16)
P	ch. III, in-phase component of filter response
P_i	ith probability
$P(s)$	polynomial matrix representation of transfer function
Q	ch. III, quadrature component of filter response
Q	ch. V, estimated covariance of process noise used in filter design process
Q_u	input weighting factor in regulator cost function
Q_y	output weighting factor in regulator cost function
r	number of measurements
R	covariance matrix defined by equation (2.16)
$R(s)$	residual polynomial matrix
s	Laplace transform variable
s_a	modal amplitude of sensor noise
s_b	sensor bias
s_ω	modal frequency of sensor noise
S_1, S_2	intermediate matrices during feedforward gain computation
tr	trace
T	defined by equations (4.46) and (4.50)
u	control input
U_i	upper limit of allowable range for ith parameter
$U(N)$	defined by equation (2.55)
v	measurement noise input
V	defined by equations (4.47) and (4.51)
w	process noise input
W	matrix of weighting parameters
X	state variable
y	output
y'_ℓ	output data at ℓth sensor attributable to the control input only
$Y'_\ell(N)$	defined by equation (2.54)
z	discrete transform variable
z_i	ith optimization variable

LIST OF SYMBOLS (Cont.)

Greek

α	denominator polynomial coefficient of discrete transfer function
α^*	optimum (converged) solution for ith denominator coefficient
α'	computed initial guess for ith denominator coefficient
α°	initial guess for ith denominator coefficient for an incremental mode
β	numerator polynomial coefficient of discrete transfer function
Γ	control input distribution matrix
Γ_w	process noise input distribution matrix
ε	error
ζ	damping ratio
$\eta(a,b)$	normal distribution of mean a and variance b
θ	vector of parameters to be identified
μ	step size factor
ρ	scaler multiplicative factor utilized in suboptimal state space design
σ_c	rms error of commanded output
σ_I	rms error of prediction for suboptimal filter
σ_{LMS}	rms error of prediction for LMS algorithm
Φ	state space transition matrix
ω	natural frequency
ω_{samp}	sample rate
Ω	frequency of square pulse wave input
∇	gradient of the error squared with respect to the parameter weighting vector

Superscripts

$\hat{}$	estimate
$\tilde{}$	error
$\cdot$	time derivative

LIST OF SYMBOLS (Cont.)

Superscripts (Cont.)

-	normalized
*	optimum
-1	inverse
L	lower partition of a matrix
T	transposed
U	upper partition of a matrix

Subscripts

amp	amplitude
c	command
con	constant
d	desired
f	feedback
I	identification
samp	sample
ss	steady state
_	vector

Abbreviations

ACSL	advanced continuous simulation language
ARMA	autoregressive, moving average
HARF	hyperstable adaptive recursive filter
IMLMS	incremental mode least mean square
LMS	least mean square
MIMO	multi-input, multi-output
MRAC	model reference adaptive control
ORACLS	optimum regulator algorithms for the control of linear systems
RMS	root mean square
SHARF	simple hyperstable adaptive recursive filter
SISO	single-input, single-output
STURE	self-tuning regulator

Chapter I

INTRODUCTION

A. ADAPTIVE CONTROL BACKGROUND

Automatic control systems are often required to operate physical plants which have a wide range of dynamic properties. These changes in dynamic properties could be caused by one of the following: 1) alterations in operating environment; 2) structural modification of the plant; or 3) failure of one of its components. If the variation of the dynamic properties is sufficiently large, a single point design of the control system (no matter how robust) may not be able to satisfy the performance specifications. Hence, a control system is required that can reconfigure itself to provide enhanced performance in the face of these variations. A control system is said to "adapt" if its internal structure or method of obtaining feedback control is altered in response to these changes.

Examples of problems which would benefit from efficient adaptive control systems exist in most areas of engineering. Flexible flight vehicles may be required to fly over a wide range of velocity, height and Mach number which significantly modifies their flight characteristics. In the case of jettisonable flight stores, rapid changes in dynamic properties occur. Space structures are difficult to analyze and test on Earth, have a wide range of changing external disturbances and may actually grow significantly during deployment and/or construction. Additionally, if for any application some element of the plant or control system fails, a reconfiguration of the control system may be desirable for either safety or performance.

Adaptive control strategies have two subdivisions of effort which can be used to distinguish them from nonadaptive systems (See fig. (I-1)). The first subdivision is a learning system which improves the information about the unknown system variables. Another task of this estimation subdivision may be the construction of state variables for use by the other subdivision, the controller. The controller subdivision determines the control inputs to the plant based upon the estimated state variables and plant parameters. Adaption occurs when either one or both of the subdivisions alter the computation scheme based upon the values of the state variables, primary plant parameters or secondary (dependent) plant parameters.

Especially in the case of continuously adapting systems, the two subdivisions interrelate. This interrelationship makes convergence and stability proofs difficult, even in cases where perfect modeling exists. The estimation subdivision uses past control inputs and past plant output measurements to generate current state estimates and parameter estimates. A key feature of this subdivision, however, is that the estimation accuracy is heavily dependent upon the control inputs. A clever sequence of controls can be used to excite specific modes, isolate effects of certain gain parameters and regulate signal-to-noise ratios at the sensors. However, control inputs that are good for estimation purposes may not be good for mission or performance specifications.

The controller subdivision computes gains based upon the current estimate of the state variables and plant parameters. While the estimation subdivision utilizes previous control inputs and output measurements, the controller uses the current values and estimates of future values to compute current control inputs. If the estimation subdivision fails, the controller subdivision will also fail. For this reason, most adaptive control research emphasizes the importance of accurate and efficient algorithms in the estimation subdivision.

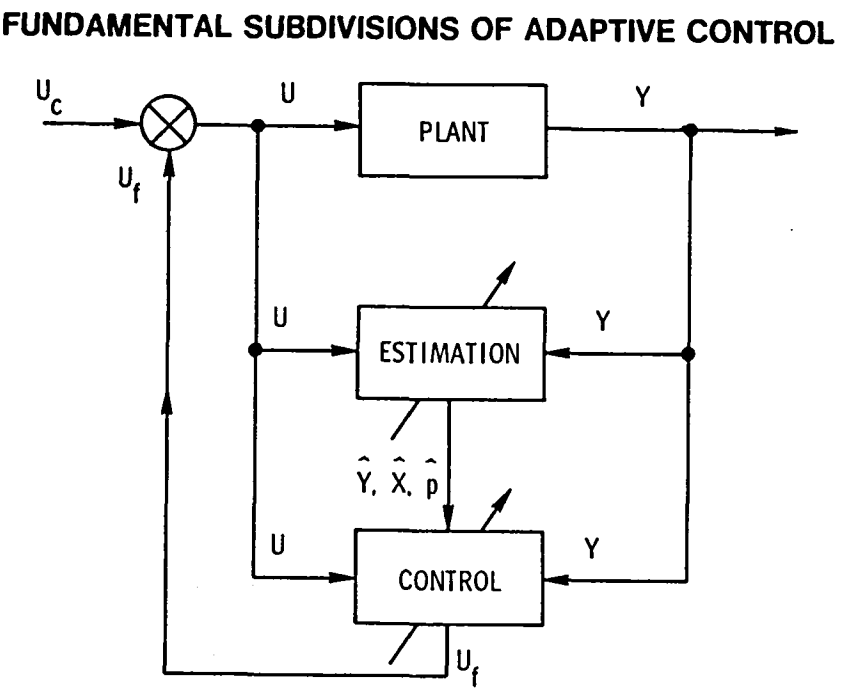

Figure I-1 — Block diagram showing fundamental subdivisions of adaptive control.

An adaptive system may be defined as one which measures its performance relative to some index and modifies its internal parameters to approach a set of optimum values [1]. It has also been suggested that an adaptive system is one that is designed from an adaptive viewpoint [2]. Clearly it becomes a logical impossibility to determine by observation of performance characteristics whether or not a control system is adaptive [3]. The very use of feedback in control systems reduces the system sensitivities to external disturbances and changes in the plant operating characteristics. The behavior of a system with feedback tends to be invariant to internal changes in itself or in the environment and may legitimately be called adaptive in the customary usage of that word.

In the context of control theory, an adaptive system usually excludes control system designs in which the state variables are measured or estimated and the parameters are assumed to be known. Although the distinction between state variables and parameters is usually based upon historical perspective, it really is a convenience of the control system design engineer and not necessarily an objective observation [3]. As an example, consider the following process

$$\dot{x} = -px + u \tag{1.1}$$

It would be customary to call x the state variable and p the parameter. Suppose that p is not a constant and can really be modeled as

$$\dot{p} = xv \tag{1.2}$$

with v as another input variable (control or external disturbance). Now the designer can approach the problem in two different ways.

If an adaptive system is desired, a controller is designed using equation (1.1) only. The parameter p is tracked by an identification scheme or estimated through explicit measurements to maintain acceptable performance as it varies throughout its range.

On the other hand, the designer could implement a nonadaptive system considering both (1.1) and (1.2) and explicitly using x and p as state variables. It would be easy to evaluate the performance of either system, but it would be impossible to determine which viewpoint was used by the designer.

A universal adaptive controller is a fictitious device which is the ultimate goal of control designers. It consists of a box with inputs and outputs which could be connected to any process. It has no prior knowledge of the system it tries to control, but after a period of time, it generates an internal model and begins to control the plant. It continues to be sensitive to any plant changes or variations in the external disturbance field while maintaining good performance characteristics. Universal adaptive controllers for general plants are assuredly a long way in the future. However, the research reported herein is a step in this direction. It represents an attempt to identify and build an internal model structure with limited prior knowledge for use in adaptive control of a limited class of problems.

In this report, a system will be considered adaptive if it is implemented with the following design principles: 1) continuous monitoring of system performance; 2) use of a figure-of-merit for decision making; and, 3) adjustment of internal parameters or structure to improve the performance. In addition, a distinction will be made between parameter adaptive control and adaptive control. Parameter adaptive control is where the control structure or internal model order is held constant. In contrast, adaptive control is the wider class of problems which requires a change in the order (number of state variables) of the internal representation. Most of the research in this area has been with parameter adaptive systems. The next section outlines some approaches to parameter adaptive control and discusses some of the inherent problems when unmodeled dynamics are neglected.

B. CURRENT STATUS OF PARAMETER ADAPTIVE CONTROL

B-1 CURRENT ALGORITHMS

Adaptive control has received attention from theoreticians and practitioners for the past 25 years. Nearly a dozen books and hundreds of papers have been devoted to the subject. Most of this research has been on parameter adaptive control which maintains the order of the modeled system as constant, thereby neglecting unmodeled dynamics. Examples of early approaches which rely principally upon analog circuit technology can be found in references [4,5]. With the advent of computer technology, new methods of parameter adaptive control were developed, including the following: model reference adaptive control, self-tuning regulators, dual-control methods, multiple-model adaptive control, and adaptive observation. These methods are reviewed below as a framework for understanding the problems that are being addressed by this research.

ADAPTIVE OBSERVERS--A logical approach toward adaptive control is to use an adaptive observer in conjunction with a constant gain matrix multiplying the estimated states as depicted in figure (I-2). Advances have been made which guarantee stability of this approach under certain restrictions [6] and [7]. This method has been further extended in [8] to include adaptive gain selection for the control law and is illustrated in figure (I-3).

MODEL REFERENCE ADAPTIVE CONTROL--Explicit and implicit model reference adaptive control (MRAC) algorithms have proven to be extremely useful for controlling plants with wide variations in plant parameters [9]. Explicit MRAC is where a reference model is specified and an adaptive control algorithm is used to make the plant output asymptotically approach the reference model output. Implicit MRAC includes the case where the reference model is adjusted during the adaptation process as an intermediate step. Again the purpose is to drive the output of the plant asymptotically toward the model response. Both approaches are illustrated in figures (I-4) and (I-5).

ADAPTIVE OBSERVER METHOD OF ADAPTIVE CONTROL

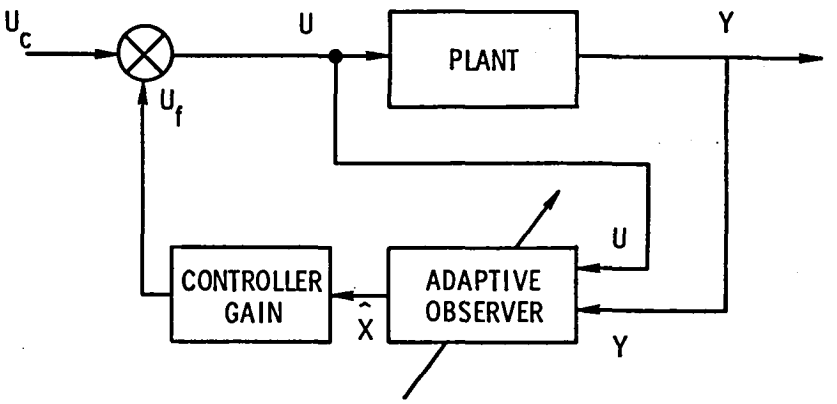

Figure I-2 — Block diagram of adaptive observer method of parameter adaptive control.

ADAPTIVE OBSERVER WITH ASYMPTOTIC GAIN SELECTION FOR ADAPTIVE CONTROL

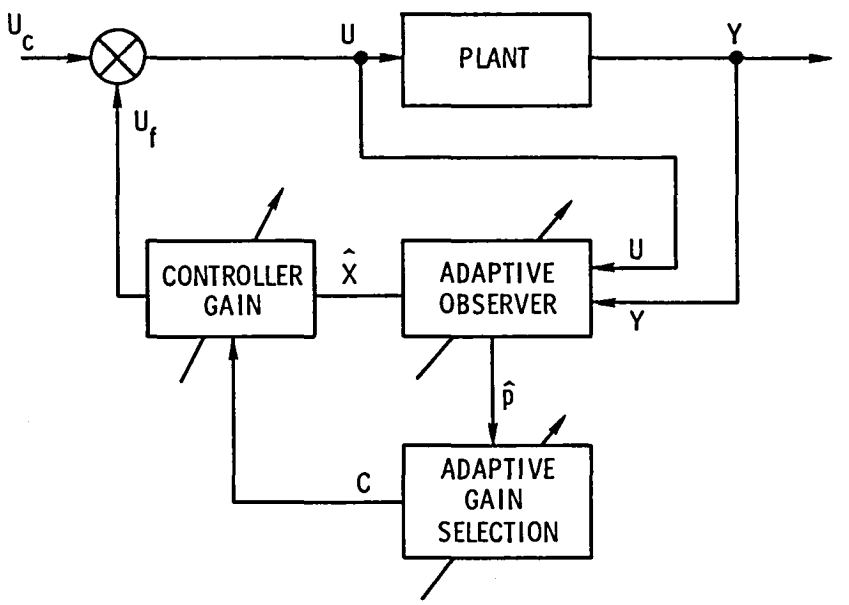

Figure I-3 — Block diagram showing adaptive control via adaptive observation with asymptotic controller gain selection.

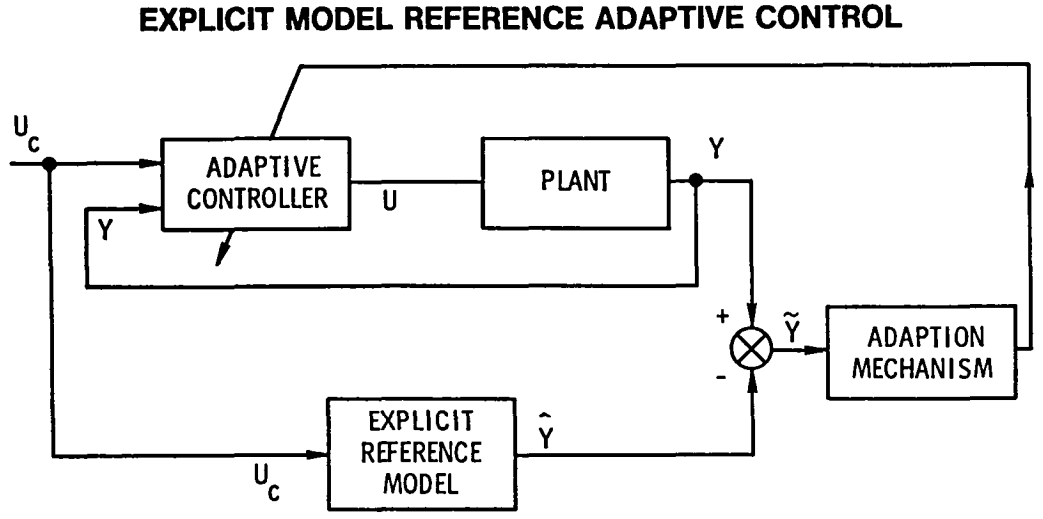

Figure I-4 — Fundamental block diagram for explicit model reference adaptive control (MRAC) algorithms.

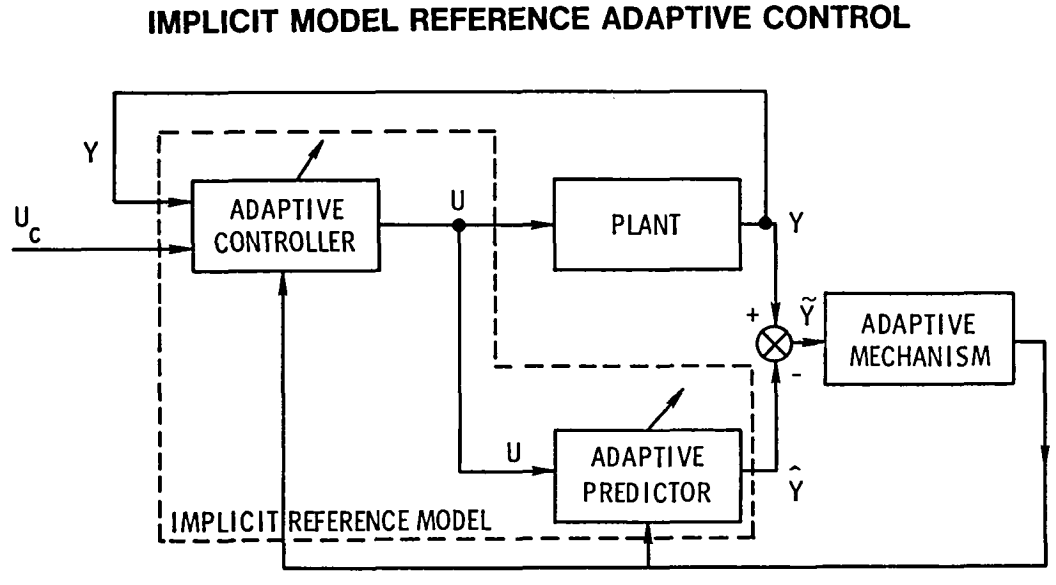

Figure I-5 — Fundamental block diagram for implicit model reference adaptive control (MRAC) algorithms.

SELF-TUNING REGULATOR--A structure representing a self-tuning regulator (STURE) is shown as figure (I-6). Although there are a number of ways to implement this approach [10,11], it usually consists of the use of output error identification techniques to update model parameters. A linear dependence of the control gains upon these parameters is typically developed so that the control strategy is adaptively updated on-line with the parameters.

MULTIPLE MODEL ADAPTIVE CONTROL--Multiple model adaptive control (MMAC) was recently investigated in references [12,13]. Figure (I-7) depicts the basic approach. A number of models are assumed and used in the state estimation subdivision. The models are evaluated based upon their performance and either 1) the best model is used in developing the control law, or 2) a weighted sum of the models is used in the control law computation. The first method affords more flexibility than pure gain scheduling. The second method allows one to hypothetically span the range of model parameters with good performance.

DUAL CONTROL--Dual control methods [14-16] use an iterative algorithm which explicitly includes a cost for identification in the total cost function. As previously mentioned, identification accuracy is highly dependent upon the control inputs, the state variables and the plant outputs. By explicitly including a cost for identification, control energy is expended in order to improve the estimates of the states and the plant parameters. Unfortunately, dual control schemes are currently computationally burdensome, which has so far prevented their use in aerospace applications.

There are approaches to parameter adaptive control other than those illustrated above. However, the information flow is similar in all approaches and can be represented by the fundamental scheme of adaptive control (fig. (I-1)). In the next subsection the impact of unmodeled dynamics upon these adaptive algorithms will be considered.

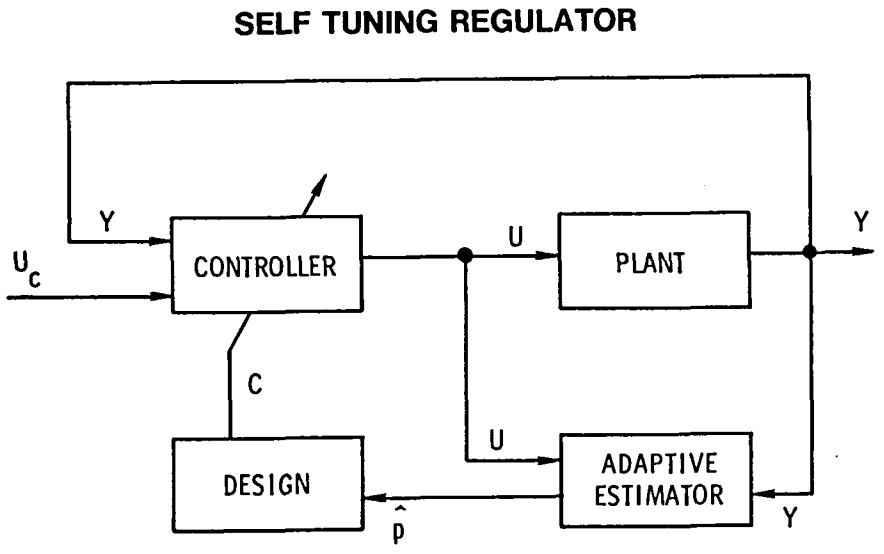

Figure I-6 — Block diagram for self-tuning regulator (STURE).

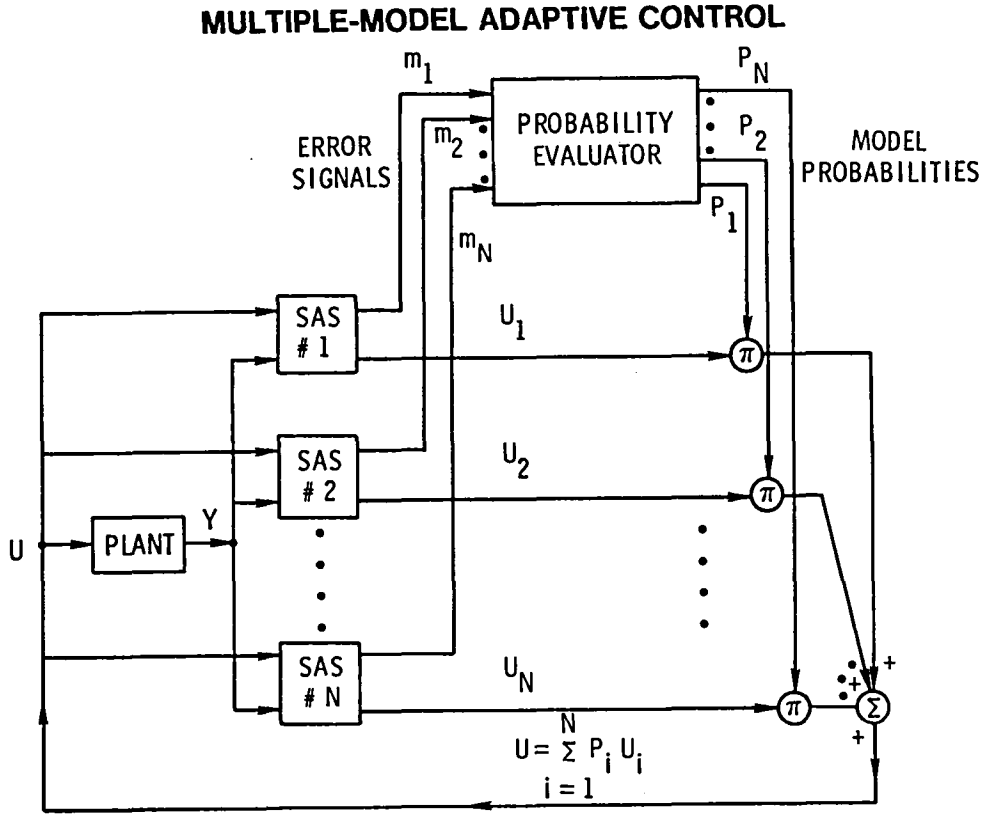

Figure I-7 -- Block diagram showing overall structure for multiple-model adaptive control (MMAC) algorithm.

B-2 CONVERGENCE PROBLEMS OF ADAPTIVE ALGORITHMS

Although adaptive control has been actively researched for the past 25 years, there have been very few cases where it has been used in actual control systems. Direct utilization of any particular algorithm without fine tuning to the problem is not currently possible. Recent research has been directed at trying to understand the fundamental problems of parameter adaptive control algorithms [17]. For example, in the case of MRAC algorithms problems tend to occur in cases where prior knowledge of the plant or operating environment is poor [18]. Specifically, MRAC algorithms suffer from the following problems: a) generation of high frequency control inputs; b) high susceptibility to instability in the presence of unmodeled dynamics; and c) poor performance in the presence of observation noise.

Adaptive control algorithms have a number of universal problems which need to be resolved. If large reference inputs are used, there is a tendency for the adaptation process to react too fast, sometimes leading to instability. An implicit adaptive algorithm which relies upon internal system identification as part of the adaptation process can be proven to converge with global stability only when unmodeled dynamics do not exist. In fact, parameter adaptive algorithms characteristically result in high gain/high bandwidth systems where the gain and bandwidth tend to grow in the presence of disturbances. Clearly this can be disastrous in terms of control spillover for truncated models.

These unfavorable aspects are a direct consequence of the information flow between the estimation subdivision and the control subdivision of the adaptation process. Errors in the predicted output and the commanded output are utilized by both subdivisions to simultaneously improve the performance, oftentimes resulting in growing bandwidth and gain. Classical solutions such as low pass filtering confuse the issue since stability and convergence proofs for the currently available adaptive algorithms require known model order and a fixed relative order.

The analytical research performed in references [17-20] relied upon a linearization procedure called "final approach analysis" to study adaptive algorithms near the solution. The closed-loop, nonlinear, time varying equations were linearized for single-input, single-output (SISO) first order systems when the system outputs and reference model outputs were close. This allowed a study of many of the adaptive algorithms for making relative performance comparisons. In addition, an exhaustive set of simulation studies were performed on many of the known parameter adaptive algorithms [19]. Results of these studies showed that most of the algorithms converged and were stable under ideal conditions. Some were even robust with adverse white noise in the process. However, all of the adaptive algorithms studied diverged under the following conditions:

1. Small bias in controls or sensors
2. Unmodeled, noncontrollable mode as sensor noise
3. Unmodeled dynamics in plant

These conditions are almost always encountered in real system implementations. Therefore, current parameter adaptive control techniques cannot be used without careful tuning.

Clearly, the direction of adaptive control research must change. If we assume that a controller exists which can handle unmodeled dynamics, sensor noise and biases, what should its form be? Methods for the estimating parameters of the structured model uncertainty are fairly mature; however, emphasis is needed to develop parameter identification algorithms for working on what is called the unstructured model uncertainty. This implies model building (or modification) on-line. A successful adaptive controller will need to modify or add on to its internal model representation whenever the unstructured model uncertainty significantly degrades the control system performance.

It has been argued that a requisite feature of a truly adaptive control algorithm which can minimize the risk of failure due to unstructured uncertainty is that it must have some level of machine intelligence. It should take advantage of computer technology to monitor its performance and adjust its characteristics in response to unmodeled (but predictable) dynamics or system errors. So a set

of heuristic rules may be necessary to insure widespread application to a variety of problems with a minimum of specific problem fine tuning. The decisions required of an intelligent controller may be similar to those an experienced engineer would need to make to optimize an algorithm to a particular system.

C. PROPOSED APPROACH FOR ADAPTIVE CONTROL

The research reported herein is an engineering approach to try and solve some of the characteristic problems of adaptive control that were mentioned in the previous sections. A fundamental difference of this research when compared to previous efforts in adaptive control is that the internal model order of the compensator will not necessarily be considered fixed. A systematic approach for building a model during real-time processing will be developed. At the crux of this development is the extension of an output error identification algorithm to explicitly include the unstructured model uncertainty. It will be shown that under a number of restrictions, it is possible to implement an algorithm with the capability of performing on-line equivalent system identification for a number of applications of adaptive control.

The algorithms are developed assuming that digital implementation [21] for current microprocessors is desired. The implicit parameter estimation algorithm that was chosen comes from digital signal processing and is referred to as the LMS (Least Mean Square) algorithm. The LMS algorithm has a very low computational burden at each time step, making real-time, recursive processing in a microprocessor possible. The LMS algorithm estimates parameters of a z-domain SISO transfer function. In chapter II this is extended to multi-input, multi-output (MIMO) systems with an autoregressive moving average (ARMA) model format. Since linear quadratic gaussian (LQG) multivariable design techniques [22] are used for the estimator and controller design, it is necessary to transform the MIMO z-domain transfer function to a minimal state space form. A computationally efficient algorithm for performing the required state space realization is developed in chapter IV.

Figure (I-8) shows the structure of the model identification algorithm. It is similar to the structure proposed in references [23,24]. A suboptimal Kalman-Bucy filter, which is suboptimal in the sense that prior knowledge of the noise statistics and the actual plant are not available, is designed and used to estimate the states for possible use by the controller. Simultaneously and in parallel, the adaptive algorithm is processing the input and output data to identify new parameters or update the ones currently used in the model for designing the Kalman-Bucy filter. Periodically the asymptotic Kalman-Bucy filter is redesigned either with the updated parameters or by adding another mode to the Kalman-Bucy filter model (increasing its order). This process is continued until some evaluation of system performance is satisfied.

The closed-loop flow diagram is depicted in figure (I-9). The parallel structure between the adaptation process and the controller is critical. The separation of the two functions is important as it prevents many of the failures mentioned in the previous sections. Large inputs, colored noise (possibly due to unmodeled dynamics) and biases, which may cause parameter estimation inaccuracies during adaptation do not feed directly through to the controller design. The properties of LQG optimal compensators in the presence of unmodeled disturbances have been extensively studied [25,27]; and while system performance may be compromised, the divergence characteristic of parameter adaptive control algorithms will only occur in extreme cases.

Another advantage of the separation of the implicit model identification task from the system control task is that internal monitoring is possible. The identification accuracy is principally a function of the control inputs used to excite the system. Since a computer program is used to monitor system performance and to periodically update system parameters and/or model order, it can also be used to make decisions about how to improve the equivalent system identification accuracy, either through adjusting some free parameters in the adaptive identification scheme or through specifying a

ADAPTIVE FILTER IN PARALLEL WITH KALMAN FILTER

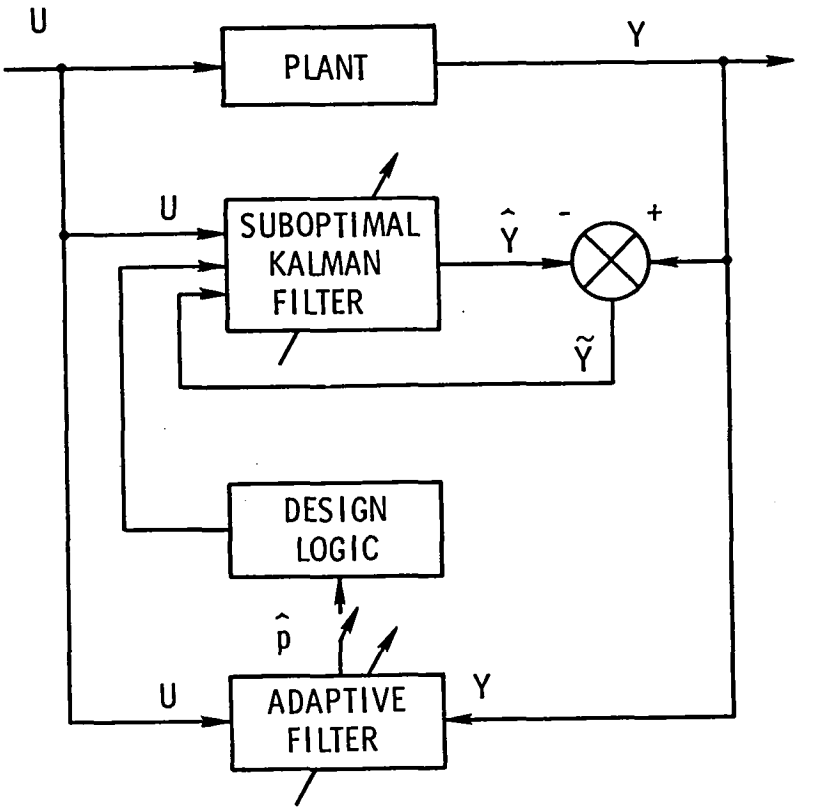

Figure I-8 — Block diagram showing use of adaptive recursive identifier for model building in parallel to Kalman-Bucy filter.

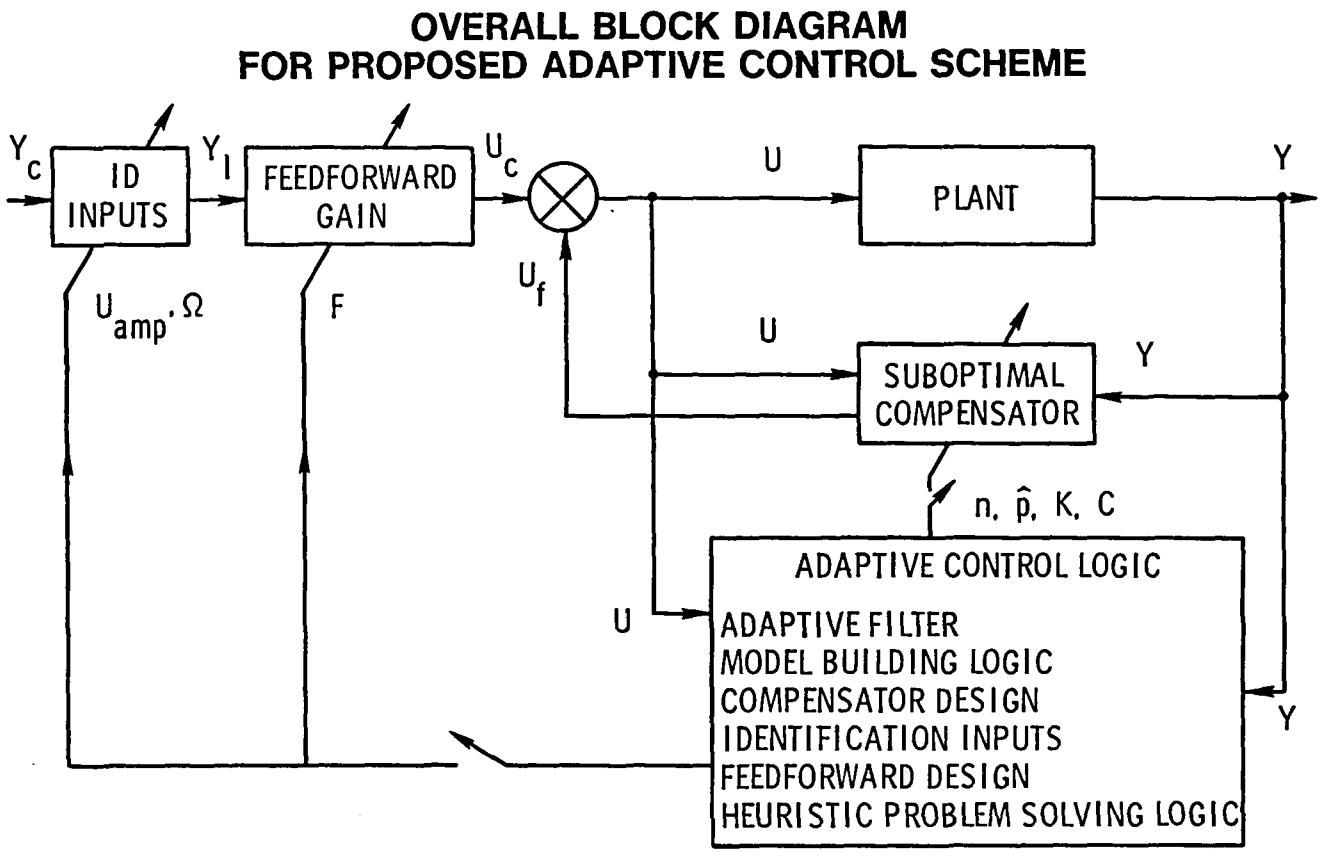

Figure I-9 — Block diagram for proposed on-line equivalent system identification scheme for adaptive control.

different set of control inputs. This decision making process is equivalent to the first stages of programming heuristic problem solving logic.

A significant contribution of this research is the modification to the output error identification algorithm, LMS, to include additional states and is called the Incremental Mode LMS (IMLMS) algorithm. The IMLMS algorithm is developed in chapter II. A number of simulations illustrating the parameter identification capabilities of the LMS and IMLMS algorithm are presented in chapter III. Complete examples of equivalent system identification with model building and adaptive control are presented in chapter VI.

Chapter II

OUTPUT ERROR IDENTIFICATION - ANALYSIS

A. INTRODUCTION

The performance of adaptive control algorithms depends primarily on the manner in which the estimation subdivision reduces the model uncertainty. Implicit adaptive control techniques use parameter identification to formally find an internal model representation of the plant prior to applying a control law. Hence, asymptotic stability of the overall algorithm requires that the model parameters approach those of the best equivalent system representation of the plant. This is easily understood in the case of plants which can be accurately represented by low order, linear, time-invariant models but becomes complicated when the plant has significant nonlinearities--higher than modeled order or time variations. It is unfortunate that these cases of high model uncertainty are the very ones which drive the engineer toward adopting adaptive control strategies as a means of obtaining or maintaining system performance. Adaptive control cannot be an attractive alternative to robust control [28] until it can treat both modeled and unmodeled plant uncertainties.

In this research, an implicit adaptive control scheme is utilized which requires the on-line identification of plant parameters. Two anticipated applications are aircraft flight control (particularly in regions of significant nonlinearities or changing parameters) and spacecraft modal damping with pointing control. In the case of spacecraft problems, a sample rate in excess of 200 Hz has been envisioned [29]. The high sample rate required, coupled with the relatively long periods of the rigid body modes, make purely batch processing techniques (e.g., least squares or maximum likelihood) impractical both from computational burden and data storage aspects. This is true especially for microprocessor

implementations. For these reasons, a recursive algorithm was choosen for the discrete output error identification operation of the adaptive control scheme.

Recursive identification has the advantage of always having an updated parameter estimate for use in the controller subdivision. However, recursive algorithms are less precise due to the simplifications needed to run at high sample rates. Hence, batch processing tends to yield more accurate results with fewer samples of data. The engineering trade-off is whether or not by the time a batch identification scheme has processed the data, a recursive scheme could have already converged to an acceptable answer. As previously mentioned, the storage and speed requirements of certain applications make microprocessor implementation of batch identification methods impractical.

Specifically, the algorithm utilized for output error identification in this research is the LMS (Least Mean Square) Adaptive Predictor Filter. It was first introduced by Widrow and Hoff [30] and has been further analyzed in references [31-33]. The algorithm is very similar to a class of adaptive filters known as SHARF (simple hyperstable adaptive recursive filter) [34-36] or sometimes HARF (hyperstable adaptive recursive filter) [37-39]. The LMS, SHARF and HARF filters are similar in formulation, but differ only in the choice of scale factors. References [31-33] analyze the LMS algorithm showing a derivation, a study of selecting the magnitude of the step size factor, an estimate of the speed adaption, and convergence proofs in stationary stochastic environments. References [37-39] are particularly useful as they also discuss parameter convergence in the presence of unmodeled dynamics.

In this chapter, the output error identification algorithms for the adaptive control scheme will be developed. In section II-B the single-input, single-output (SISO) form of the LMS algorithm will be developed. Section II-C extends the LMS algorithm to multivariable (multi-input, multi-output or MIMO) plants. Section II-D studies the analytical convergence properties of the output error identifi-

cation algorithm. A hybrid batch and recursive version of the algorithm is developed in section II-E. A prime contribution of this research, an explicit inclusion of the unstructured model uncertainty in the output error identification, is included as section II-F.

B. DERIVATION OF SISO OUTPUT ERROR IDENTIFICATION

In this section the LMS adaptive filter algorithm is derived following the development of references [31,32]. It should be noted that the notation used in this derivation prevents the extension to a multivariable representation, a necessity for the algorithm's use in practical stochastic control applications.

Using the autoregressive moving average (ARMA) representation [21], a dynamic system is represented in terms of past measurements, y(k), and past control inputs, u(k), as:

$$\hat{y}(k) = \underline{x}^T(k)\ \underline{\alpha}(k) \tag{2.1}$$

where

$$x^T(k) = [y(k-1),y(k-2)\ \ldots\ y(k-n),u(k-1),u(k-2)\ \ldots\ u(k-n)] \tag{2.2}$$

$\alpha(k)$ is the vector of weights multiplying past state measurements and controls for obtaining the predicted output, $y(k)$, of output $y(k)$, and n is the order of the model. The error is defined by comparing the measured output and the predicted output

$$\varepsilon(k) = y(k) - \hat{y}(k) \tag{2.3}$$

Substituting (2.1) into (2.3), results in

$$\varepsilon(k) = y(k) - x^T(k)\underline{\alpha}(k) \tag{2.4}$$

It is desired to minimize the square of the error for all time ($-\infty<k<\infty$) by finding the optimum set of weights, $\underline{\alpha}^*$. A logical approach would be to update a trial set of weights using a simple steepest descent algorithm.

$$\underline{\alpha}_{i+1} = \underline{\alpha}_i - \mu\nabla \ , \tag{2.5}$$

where the gradient is given by

$$\nabla \equiv \frac{\partial}{\partial \underline{\alpha}_i} \sum_{j=-\infty}^{\infty} \varepsilon^2(j) \tag{2.6}$$

Since a recursive algorithm is desired, an update for each discrete time step is needed. The gradient of the square of the error is formulated as a time dependent variable

$$\nabla_\varepsilon(k) = \frac{\partial}{\partial \underline{\alpha}_i} \sum_{j=-\infty}^{k} \varepsilon^2(j) \tag{2.7}$$

This would still require a great deal of storage and computation for each α_i at each step k. The convention is to update the $\underline{\alpha}_i$ at each time step, so adopting the notation

$$\underline{\alpha}_i = \underline{\alpha}(k) \ , \tag{2.8}$$

and making the following crucial approximation

$$\nabla(k) \approx \hat{\nabla}(k) = \frac{\partial}{\partial \underline{\alpha}(k)} \varepsilon^2(k) \ , \tag{2.9}$$

an estimate for the gradient is obtained. That is, instead of summing all the past errors using the current value of $\underline{\alpha}(k)$, just

use the current error. The definition of the gradient from (2.6) yields

$$\hat{\nabla}(k) = 2\varepsilon(k) \frac{\partial \varepsilon(k)}{\partial \underline{\alpha}(k)} \tag{2.10}$$

Using (2.4)

$$\frac{\partial \varepsilon(k)}{\partial \underline{\alpha}(k)} = -\underline{x}(k) \tag{2.11}$$

so (2.10) becomes

$$\hat{\nabla}(k) = -2\varepsilon(k)\underline{x}(k) \tag{2.12}$$

Substituting (2.12) into (2.5) using (2.6), the LMS adaptive filter for updating the weights is obtained

$$\underline{\alpha}(k+1) = \underline{\alpha}(k) + 2\mu\varepsilon(k)\underline{x}(k) \ , \tag{2.13}$$

or it can be rewritten as

$$\underline{\alpha}(k+1) = \underline{\alpha}(k) + \mu\underline{x}(k)[y(k)-\underline{x}^T(k)\alpha(k)]. \tag{2.14}$$

This differs from recursive least squares [21] only in the term μ, which replaces the estimate-error covariance matrix P(k+1). There P is updated at each step by

$$P^{-1}(k+1) = P^{-1}(k) + x(k)x^T(k), \tag{2.15}$$

or

$$P(k+1) = P(k) - P(k)x(k)[I+x^T(k)x(k)]x^T(k)P(k) \tag{2.16}$$

In effect, the LMS algorithm replaces the step-varying P matrix by a constant diagonal matrix μI, where I is the unity matrix. Hence, a reasonable estimate for μ is

$$\mu \approx \frac{1}{n} \mathrm{tr}(P), \tag{2.17}$$

i.e., the average error variance of the parameters α_i. The difference between the LMS algorithms and the SHARF algorithms also lies in the selection of weighting factors.

The parameter update from (2.13) can be made very quickly as it requires only about $5n$ operations. Despite its simplicity, it is data adaptive and has convergence properties that approach those of more cumbersome conventional methods. Some simple convergence arguments follow. Take the expected value of (2.12)

$$E[\hat{\nabla}(k)] = E[-2\varepsilon(k)\underline{x}^{-}(k)] \tag{2.18}$$

Noting that $\varepsilon(k)$ is a scalar and using (2.4)

$$E[\hat{\nabla}(k)] = -2E[y(k)\underline{x}^{T}(k) - \underline{x}(k)\underline{x}^{T}(k)\underline{\alpha}(k)] \tag{2.19}$$

Defining the stochastic covariances as

$$\begin{aligned} P(k) &\equiv y(k)\underline{x}^{T}(k) \\ R(k) &\equiv \underline{x}(k)\underline{x}^{T}(k) \end{aligned} \tag{2.20}$$

It can be shown that the gradient is equal to (e.g., [41])

$$\nabla = R\underline{\alpha} - P \tag{2.21}$$

Hence

$$E[\hat{\nabla}(k)] = \nabla_{\varepsilon} \tag{2.22}$$

Solving (2.21) for the optimum weight vector gives the well known Wiener solution

$$\underline{\alpha}^* = R^{-1}P \tag{2.23}$$

Since the mean value of the gradient estimate, $\hat{\nabla}(k)$ is equal to the gradient, ∇, the estimate must be unbiased.

The above analysis assumes that the weight vector was held constant. References [31,33,41] extend the analysis to include variation of weights and show that the Weiner solution is again obtained for the optimum weights if the inputs are uncorrelated over time and are stochastically stationary. References [31,33] remove the stationarity requirement in a formal manner. A discussion of the weight vector convergence is given in section II-D.

This development of the LMS algorithm can be extended to find bounds for the estimate of step size factor, μ, to be chosen by the engineer. The analyses of references [31,32,41] shows that

$$0 < \mu < \frac{1}{tr[R]} \tag{2.24}$$

as a requirement for convergence. Simulation studies in reference [32] indicate that the normalized error of misadjustment of weight parameters is linearly proportional to the number of weights. In addition, a discussion of the speed of adaptation shows it to be an exponential function of μ, the eigenvalues of R and the number of weight parameters. References [31,33] indicate that based upon experience, a "good" value for μ for design purposes is around .001 plus average input signal power.

Reference [33] is significant in that it discusses the LMS algorithm in terms of a wider class of problems and allows that adequate performance is possible under dependent random environments, slowly changing parameters and in the presence of unmodeled dynamics. A prime contribution of the literature for SHARF and HARF

algorithms is the explicit inclusion of the recursive identification scheme in an adaptive control function. Analyses in these references [34-39] are concerned with proving global asymptotic convergence of the parameters, studying the impact of unmodeled dynamics and quantifying the sufficient excitation requirements for parameter convergence. Some of these issues will be discussed in section D. The next section develops a MIMO formulation of the LMS adaptive filter.

C. MIMO FORMULATION OF LMS ALGORITHM

Most practical applications of adaptive control involve MIMO systems, that is, systems with more than one control input and more than one output. In this section the LMS algorithm is extended to MIMO systems. The vector notation utilized in the previous section will not accomodate multi-output formulations, so a tensor notation is used to maintain the same AR (autoregressive) coefficients for each output. In addition, a distinction shall be made between the terms multiplying the measurements and the controls.

Let

$$y_\ell(k) = \ell\text{th measurement at time } k$$

$$e_j(k) = j\text{th input at time } k$$

Assume that the ℓth measurement can be predicted as

$$\hat{y}_\ell(k) = \alpha_1 y_1(k-1) + \alpha_2 y_2(k-2) + \ldots + \alpha_n y_\ell(k-n) + \beta_{1j\ell} e_1(k-1) + \beta_{2j\ell} e_2(k-2) + \ldots + \beta_{nj\ell} e_j(k-n) \qquad (2.25)$$

the notation can be condensed by making the following definitions consistent with an ARMA model:

$$x_{i\ell}(k) \equiv y_\ell(k-i) \tag{2.26}$$

$$u_{ij}(k) \equiv e_j(k-i) \tag{2.27}$$

Tensor notation with the summation convention implying the following summation limits will be used:

$$\sum_{i=1}^{n} \qquad \sum_{j=1}^{m} \qquad \sum_{\ell=1}^{r}$$

where n is the model order, m is the number of control inputs and r is the number of outputs. Now (2.25) can be rewritten

$$\hat{y}_\ell(k) = \alpha_i(k)x_{i\ell}(k) + \beta_{ij\ell}(k)u_{ij}(k) \ . \tag{2.28}$$

The error of prediction becomes

$$\varepsilon_\ell(k) \equiv y_\ell(k) - \hat{y}_\ell(k) \tag{2.29}$$

or

$$\varepsilon_\ell(k) = y_\ell(k) - x_i(k)\alpha_{i\ell}(k) - \beta_{ij\ell}(k)u_{ij}(k) \ . \tag{2.30}$$

As in the previous section it is desired to find an approximate gradient of the square of the error with respect to the weights, $\alpha_i(k)$ and $\beta_{ij\ell}(k)$.

So assuming that the gradient can be approximated by using the current weights and errors,

$$\frac{\partial}{\partial\alpha_i}(\varepsilon_\ell\varepsilon_\ell) = 2\ \varepsilon_\ell\left(\frac{\partial}{\partial\alpha_i}\varepsilon_\ell\right) \tag{2.31}$$

$$= 2\ \varepsilon_\ell \frac{\partial}{\partial\alpha_i}(y - \alpha_i x_{i\ell} - \beta_{ij\ell}u_{ij})\varepsilon_\ell \tag{2.32}$$

including time dependence

$$\frac{\partial}{\partial \alpha_i(k)} \left(\varepsilon_\ell(k)\varepsilon_\ell(k)\right) = -2\varepsilon_\ell(k)x_{i\ell}(k) \ , \tag{2.33}$$

and for the control influence terms

$$\frac{\partial}{\partial \beta_{ij\ell}} \left(\varepsilon_\ell \varepsilon_\ell\right) = 2\varepsilon_\ell \left(\frac{\partial}{\partial \beta_{ij\ell}}\right)\varepsilon_\ell \tag{2.34}$$

$$\frac{\partial}{\partial \beta_{ij\ell}} \left(\varepsilon_\ell{}^T \varepsilon_\ell\right) = 2\varepsilon_\ell(y - \alpha_i x_{i\ell} - \beta_{ij\ell} u_{ij}) \tag{2.35}$$

$$\frac{\partial}{\partial \beta_{ij\ell}} \left(\varepsilon_\ell{}^T(k)\varepsilon_\ell(k)\right) = -2\varepsilon_\ell(k)u_{ij}(k) \ . \tag{2.36}$$

So again using a steepest descent approach, equation (2.5) can be used to give

$$\alpha_i(k+1) = \alpha_i(k) - \mu \frac{\partial\left(\varepsilon_\ell{}^T(k)\varepsilon_\ell(k)\right)}{\partial \alpha_i(k)} \tag{2.37}$$

$$\alpha_i(k+1) = \alpha_i(k) + 2\mu\varepsilon_\ell(k)x_{i\ell}(k) \tag{2.38}$$

and

$$\beta_{ij\ell}(k+1) = \beta_{ij\ell}(k) - \mu \frac{\partial\left(\varepsilon_\ell{}^T(k)\varepsilon_\ell(k)\right)}{\partial \beta_{ij\ell}(k)} \tag{2.39}$$

$$\beta_{ij\ell}(k+1) = \beta_{ij\ell}(k) + 2\mu\varepsilon_\ell(k)u_{ij}(k) \ . \tag{2.40}$$

The parameter update equations which comprise the MIMO LMS algorithm are (2.38) and (2.40). Previous formulations implied a separate set of α_i's for each output, where as now there is a single set of parameters constrained to remain valid for all outputs

while maintaining the desirable convergence properties of the SISO LMS algorithm. This permits a simpler, multivariable ARMA formulation for the plant. The α and β parameters are merely the denominator and numerator terms of a discrete, multivariable transfer function. Chapter IV develops an efficient way to compute an approximate minimal state space realization from the experimentally determined transfer functions. The next section discusses convergence of the LMS algorithm, applicable to both SISO and MIMO systems.

D. NOTES ON OUTPUT ERROR IDENTIFICATION CONVERGENCE

In this section, the notation of the previous section will be used to develop a first order example to illustrate many of the convergence properties of the LMS algorithm. The plant is assumed to be known and the convergence of the identifier algorithm is to be investigated.

Plant: $$x(k+1) = ax(k) + bu(k) \tag{2.41}$$

Estimator: $$\hat{x}(k+1) = \alpha(k)x(k) + \beta(k)u(k) \tag{2.42}$$

Error: $$\varepsilon(k) \equiv x(k) - \hat{x}(k) \tag{2.43}$$

$$\varepsilon(k) = \big(a-\alpha(i-1)\big)x(i-1)+\big(b-\beta(i-1)\big)u(k-1) \tag{2.44}$$

LMS Identifier: $$\alpha(k+1) = \alpha(k) + 2\mu\{[a-\alpha(k)]x(k) + [b-\beta(k)]u(k)\}x(k+1)\} \tag{2.45}$$

$$\beta(k+1) = \beta(k) + 2\mu\{[a-\alpha(k)]x(k) + [b-\beta(k)]u(k)\}x(k+1)\} \tag{2.46}$$

Equations (2.42) and (2.43) become

$$\begin{bmatrix}\alpha(k+1)\\ \beta(k+1)\end{bmatrix} = \begin{bmatrix}\alpha(k)\\ \beta(k)\end{bmatrix} + 2\mu\begin{bmatrix}x(k)x(k+1) & u(k)x(k+1)\\ x(k)u(k+1) & u(k)u(k+1)\end{bmatrix}\begin{bmatrix}a-\alpha(k)\\ b-\beta(k)\end{bmatrix} \tag{2.47}$$

From (2.47) it is easy to see the requirement for sufficient excitation. [c.f.,36,39,40,42]. If steady state conditions are reached such that $x(k) \cong x(k+1)$ or $u(k) \cong u(k+1)$ no improvement is made in the parameters no matter how large the parameter errors, $a-\alpha(k)$ and $b-\beta(k)$, are. Control inputs are required to improve β.

If a step input of magnitude u_o is applied at $k=0$ with $x(0)=0$, it follows that

$$x(k) = (1-a^k)\frac{bu_o}{1-a} \tag{2.48}$$

indicating that any stable parameterization of α and β with a zero frequency gain of $b/(1-a)$ will show $\hat{x} \to x$ [42]. Clearly, α and β may not have converged to a and b.

Combining (2.47) and (2.48) near $k=0$, we get

$$\begin{bmatrix}\alpha(k+1)\\ \beta(k+1)\end{bmatrix} = \begin{bmatrix}\alpha(k)\\ \beta(k)\end{bmatrix} + 2\mu\begin{bmatrix}0 & 0\\ 0 & u_o^2\end{bmatrix}\begin{bmatrix}\alpha - \alpha(k)\\ b - \beta(k)\end{bmatrix} \tag{2.49}$$

whereas for any k,

$$\begin{bmatrix}\alpha(k+1)\\ \beta(k+1)\end{bmatrix} = \begin{bmatrix}\alpha(k)\\ \beta(k)\end{bmatrix} + 2\mu\begin{bmatrix}a(1-a^k)^2\left(\frac{bu_o}{1-a}\right)^2 & (1-a^k)\frac{abu_o^2}{1-a}\\ (1-a^k)\frac{bu_o^2}{1-a} & u_o^2\end{bmatrix}\begin{bmatrix}a-\alpha(k)\\ b-\beta(k)\end{bmatrix} \tag{2.50}$$

and for $k \to \infty$

$$\begin{bmatrix}\alpha(k+1)\\ \beta(k+1)\end{bmatrix} = \begin{bmatrix}\alpha(k)\\ \beta(k)\end{bmatrix} + 2\mu\begin{bmatrix}a\left(\frac{bu_o}{1-a}\right)^2 & \frac{abu_o^2}{1-a}\\ \frac{bu_o^2}{1-a} & u_o^2\end{bmatrix}\begin{bmatrix}a-\alpha(k)\\ b-\beta(k)\end{bmatrix} \tag{2.51}$$

For small k the only improvement is in β and it can have a relatively large magnitude depending upon the sizes of μ and u_o.

As $x(k)$ moves away from 0, corrections in α begin. It is possible, however, for α to initially move in the wrong direction depending upon u_o, $b-\beta(k)$ and μ.

Considering equations (2.49) and (2.50), an input strategy suggests itself for obtaining sufficient excitation for parameter convergence. A step input provides immediate improvement in β. Experience, such as the results in chapter III, indicates that the first few samples after the step impulse provide the majority of the improvement. As the system begins to respond to a step input, the improvement in α grows in magnitude; but the improvement decreases at an exponential rate, albeit slower than the improvement in β. If only a single step pulse is applied through the control, the parameters will most likely not converge to an accurate answer.

Multiple step pulses are required for sufficient excitation. This is consistent with the general identification research for optimal inputs of references [43-45] and the specific LMS results of [33,36,37,39]. The pulses are needed to improve the control influence terms, β, but improvement in the denominator terms, α, comes only after waiting for the system to respond. So the designer is confronted with a conflict when choosing input signals to enhance the parameter estimate. Frequent pulses would be advantageous for β, but may not give α enough time to begin its improvement. Conversely, allowing the system to respond to infrequent pulses would yield an ideal convergence environment for α but would be insufficient for β.

In chapter IV it will be shown that equation (2.42) has a simple representation as a discrete (z-domain) transfer function and that α corresponds to the denominator term and β to the numerator term. The need for several rapidly changing control inputs to estimate the numerator terms is typical of identification problems. In contrast, a low frequency externally applied signal (e.g., a dither signal) is advantageous for identifying the denominator terms. Although white process noise is capable of providing a parameter convergence for α, studies [31,32] indicate that a control-induced, signal-to-noise ratio of 2 or more will start to yield the exponential convergence rates indicated in equation (2.50).

It is apparent that the rate of convergence is dependent upon the relative sizes of μ and the mean values of x and u. The implication is that μ should be sized to handle the expected values of x and u. An alternative approach, however, may be to scale the outputs of the state variable representation so as to maintain expected values for x and u near unity. This later case is the approach adopted in this research.

E. BATCH LEAST SQUARES NUMERATOR DERIVATION

In the previous section it was observed that the improvement in the numerator term of the discrete transfer function representation of the LMS adaptive algorithm comes primarily within the first few samples following a pulse. Therefore, many pulses are required to obtain convergence of the numerator terms. However, it is advantageous to wait on the order of a time constant between pulses so the denominator terms can approach their true value. The result is that the denominator terms converge more rapidly than the numerator terms.

Since for most adaptive control applications the speed of adaptation is critical, a hybrid batch and recursive identification scheme is proposed. Once the denominator terms have been identified it becomes a relatively simple matter to use a batch least squares algorithm to identify the numerator terms. This operation is tantamount to finding the zero frequency gain of the system; hence, data from the last two pulses are probably all that are necessary for sufficient accuracy. It has already been argued that collecting data for several system time constants and applying a computationally burdensome batch identification scheme (e.g., maximum likelihood techniques [46]) for finding a best model order and/or parameter estimates is inappropriate for microprocessor applications. This is true from both a data storage and a time for computation standpoint.

Using a batch scheme for the numerator terms has some potential advantages. A prime problem with least squares batch algorithms are the inconsistent results obtained with model truncation. This is

not as significant a problem for the LMS algorithm, as will be illustrated in section III-C. Also, batch data that use only the last two pulses are all that is required to be stored for the accurate estimation of the control influence terms. Since the parameters for the denominator terms from the LMS algorithm are available at each sample, the innovation sequence can be recursively modified to contain only the numerator content, making the solution for the numerator terms a simple computation prior to obtaining a state space realization.

Using an ARMA model representation [21],

$$\hat{\underline{y}}(k) = -\sum_{i=1}^{n} \alpha_i \underline{y}(k-i) + \sum_{i=1}^{n'} B_i \, \underline{u}(k-i) + \underline{\varepsilon}(k) \tag{2.52}$$

Note that B_i is summed up to n', the assumed order of the numerator terms, where $n' \leq n$. B_i has the dimensions of r x m, where r is the number of measurements and m is the number of control inputs. So rewriting (2.52) in tensor notation from section II-B

$$\hat{y}_\ell(k) = -\alpha_i y_\ell(k-i) + \beta_{ij\ell} u_j(k-i) + \varepsilon_\ell(k) \tag{2.53}$$

Assume that the α_i are known and that only the part of error associated with not knowing the $\beta_{ij\ell}$ is needed. The part of the output attributable to the control input u can be written

$$y'_\ell(k) = \hat{y}_\ell(k) - \alpha_i y_\ell(k-i) \tag{2.54}$$

Hence, we define a parameter vector, with dimension (nm x 1),

$$\theta_\ell = [\beta_{11\ell}\beta_{21\ell}\cdots\beta_{n1\ell}\beta_{12\ell}\cdots\beta_{nm\ell}]^T, \tag{2.55}$$

a control vector of dimension (n'm x 1)

$$u(k) = [u_1(k-1)u_1(k-2)\ldots u_1(k-n')u_2(k-1)\ldots u_m(k-n')]^T, \qquad (2.56)$$

and a data vector with N samples with dimension (N-n+1x1) is defined as,

$$Y'_\ell(N) = [y'_\ell(n)y'_\ell(n+1)\ldots y'_\ell(N)]^T. \qquad (2.57)$$

By forming the following matrices,

$$U(N) = [\underline{u}(n), \underline{u}(n+1), \ldots \underline{u}(N)]^T \qquad (2.58)$$

$$\varepsilon_\ell(N,\theta) = [\varepsilon_\ell(n),\varepsilon_\ell(n+1)\ldots\varepsilon_\ell(N)]^T \qquad (2.59)$$

The N-n+1 equation errors can be written in matrix notation as

$$Y'_\ell(N) = U(N)\theta_\ell + \varepsilon_\ell(N,\theta) \qquad (2.60)$$

The cost function is the square of the errors and is defined by

$$J_\ell(\theta_\ell) = \underline{\varepsilon}_\ell{}^T(N,\theta_\ell)\underline{\varepsilon}_\ell(N,\theta_\ell) \ . \qquad (2.61)$$

The normal equations for this case are

$$U^T W U\theta_\ell = U^T W\overline{Y} \qquad (2.62)$$

yielding the conventional weighted least squares solution

$$\hat{\theta}_\ell = (U^T WU)^{-1}U^T W\overline{Y}_\ell \ . \qquad (2.63)$$

The weighting matrix, W, is assumed to be the identity matrix for the remaining analysis. All that is necessary for this computation is to save Nn'm control inputs in the vector U(N) and ℓN samples of the control contribution to output, y'_ℓ, in vectors $y'_\ell(N)$. It should be noted that y'_ℓ is a byproduct of the LMS algorithm computation. The α_i used in the LMS algorithm should be satisfactory, provided that enough time constants have elapsed and little adaptation is taking place. Finally, it should be observed that the θ_ℓ need to be identified for each output, ℓ. This comes from the formulation of (2.61) which was adopted for saving computer storage space.

F. INCREMENTAL MODE LMS ALGORITHM DERIVATION

LMS adaptive filters perform well in the presence of higher frequency, unmodeled modes since they act as low-pass filters. This capability of the LMS algorithm is studied analytically in references [33,39,40,47] and by simulation in section III-C. It is clear, however, that there are times when it would be advantageous to add modes to the assumed model structure. Furthermore, if some prior knowledge of the plant is available, it would be beneficial to include it in the identification process. Chapter I pointed out the intrinsic problems of unstructured model uncertainty in terms of stability for adaptive control algorithms. In this section an approach for adding modes to the model structure will be developed by distinguishing between the known (or already identified) part of the dynamics and the unknown (or incremental) part.

One reason that most parameter convergence studies for output error identification algorithms are limited to low order is because of the inherent numerical inaccuracies that predominate as the order is increased. Equation (2.64) is the discrete (z-domain) transfer function of equation (2.25)

$$H_{\ell j}(z) = \frac{\beta_{1j\ell} z^{-1} + \beta_{2j\ell} z^{-2} + \ldots \beta_{nj\ell} z^{-n}}{1 - \alpha_1 z^{-1} - \alpha_2 z^{-2} + \ldots \alpha_n z^{-n}} \qquad (2.64)$$

The effect of a small error in α_i on the location of the poles increases with n.

A natural approach would be to find the optimum set of α^* using the LMS algorithm for n, then assume that the order will be increased to n+p, and compute a starting set of n+p values of α' using the n values of α^* obtained from the LMS algorithm and initial guess for p values of α^0 for the new modes to be identified. The indicated product is given by

$$1-\alpha_1' z^{-1}+\ldots\alpha_{n+p}' z^{-(n+p)} = \left(1-\alpha_1^* z^{-1}+\ldots+\alpha_n^* z^{-n}\right) \cdot \left(1-\alpha_1^0 z^{-1}+\ldots+\alpha_p^0 z^{-p}\right) \qquad (2.65)$$

The LMS algorithm can then be used to identify n+p values of α', the new coefficients of the larger model. However, this does not solve the numerical accuracy problem of higher order systems posed above.

Instead, a method whereby only the new modes are adapted and the previously identified modes are held constant is desired. This would tend to minimize the impact of numerical inaccuracies of the adaptation process upon the estimated system dynamics. It also provides a way to include the known dynamics into the identification. The algorithm that is derived below is termed the Incremental Mode LMS (IMLMS) algorithm since it provides a way to add incremental modes to the assumed model.

We cast the form of the discrete transfer function of (2.64) into one which explicitly distributes dynamics into two parts: (1) the nonvarying or known part and (2) the incremental or unknown part.

$$H(z) = \frac{\sum_{i=1}^{n+p} B_i z^{-i}}{\left(1 - \sum_{i=1}^{n} a_i z^{-i}\right)\left(1 - \sum_{f=1}^{p} \alpha_f z^{-f}\right)} \qquad (2.66)$$

where B_i is a rxm matrix for a multivariable system and is identified each time,

$$B_i = \begin{bmatrix} \beta_{i11} & \beta_{i12} & \cdots & \beta_{i1m} \\ \beta_{21} & \beta_{22} & \cdots & \beta_{i2m} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{ir1} & \beta_{ir2} & \cdots & \beta_{irm} \end{bmatrix} \tag{2.67}$$

The a_i are assumed constant and only the α_f, the coefficients of the incremental modes are identified. The new system order is taken as n+p. Multiplying the denominator out in (2.66), the following expression is obtained

$$H(z) = \frac{\sum_{i=1}^{n+p} B_i z^{-i}}{1 - \sum_{i=1}^{n} a_i z^{-i} - \sum_{f=1}^{p} \alpha_f z^{-f} + \sum_{i=1}^{n} \sum_{f=1}^{p} a_i \alpha_f z^{-i-f}} \tag{2.68}$$

Writing (2.68) in finite difference form [21]:

$$\hat{y}_\ell(k) = a_i y_\ell(k-i) + \alpha_f y_\ell(k-f) - a_i \alpha_f y_\ell(k-i-f) + \beta_{i\ell j} u_j(k-i) \tag{2.69}$$

The implied summation limits of i are 1 to n, of f are 1 to p, of ℓ are 1 to r, and of j are 1 to m. Using equations (2.28), (2.37) and (2.29) with the above notation a form for the parameter update is

$$\alpha_f(k+1) = \alpha_f(k) - 2\mu\varepsilon_\ell(k) \frac{\partial \varepsilon_\ell}{\partial \alpha_f}(k) \tag{2.70}$$

Finding the first partial of the error, $\varepsilon_\ell(k)$

$$\frac{\partial \varepsilon_\ell}{\partial \alpha_f}(k) = -\frac{\partial}{\partial \alpha_f}\left(y_\ell(k)\right) \tag{2.71}$$

$$\frac{\partial \varepsilon_\ell}{\partial \alpha_f}(k) = y_\ell(k-f) - a_i y_\ell(k-i-f) \tag{2.72}$$

So the unknown parameter update equation becomes

$$\alpha_f(k+1) = \alpha_f(k) + 2\mu\varepsilon_\ell(k)[y_\ell(k-f) - a_i y_\ell(k-i-f)] \tag{2.73}$$

The parameter update equations for the numerator terms, $\beta_{i\ell j}$, remained unchanged as

$$\beta_{i\ell j}(k+1) = \beta_{i\ell j}(k) + 2\mu\varepsilon_\ell u_{ij}(k) \ . \tag{2.40}$$

The LMS algorithm has been extended to include only the adaptation of the unknown modes of arbitrary order. No assumptions used in proving asymptotic convergence of the basic LMS algorithm have been violated; hence, we expect the IMLMS algorithm to exhibit the same asymptotic robustness as the unmodified LMS algorithm. The derivation of the IMLMS algorithm was accomplished in the MIMO formulation of section II-C. Since the numerator update equations are unchanged, the same batch least squares identification of the numerator terms is still possible using the development in section II-E.

While it is anticipated that the IMLMS algorithm will exhibit robust performance under nonideal conditions (sensor noise, processing noise or unmodeled dynamics), it will be similar to other identification algorithms in that global convergence proofs will not be possible in such circumstances. The inclusion of the IMLMS algorithm in an adaptive control application will need heuristic tests to check for divergence prior to utilization of the newly identified model. Furthermore, there is a danger of incremental fitting of multiple modes which have eigenvalues with nearly equal magnitudes. Section III-C will illustrate that spectral separation is necessary for good performance in the presence of unmodeled dynamics, enabling the IMLMS algorithm to distinguish between modes.

The IMLMS algorithm should be especially suitable for aerospace applications. Flight vehicles often have a single mode which is relatively easy to estimate a priori, but have additional modes which are susceptible to wide variations (e.g., phugoid and short period modes or roll and Dutch roll modes). Spacecraft typically have rigid body modes which are easy to predict, but the interaction with the flexible body modes is difficult to estimate or even measure with experiments prior to deployment. In both types of vehicles, the use of an algorithm which holds part of the model constant while identifying the incremental part would be advantageous compared to approaches which identify the entire model for a given order.

In this section, the IMLMS algorithm was presented as an effective way to identify explicitly part of a model while holding the dynamics of another part fixed. There are many applications where this may be a useful approach. In the next chapter, the LMS algorithm parameter convergence properties will be examined by simulation. First, the standard LMS algorithm performance will be studied from the viewpoint of a designer. Then, the impact of using the multivariable formulation will be ascertained; the influence of unmodeled modes will be evaluated; and, the robustness of the IMLMS algorithm will be demonstrated.

Chapter III

OUTPUT ERROR IDENTIFICATION - SIMULATION STUDIES

In this chapter the results of simulation studies investigating the use of the LMS algorithm as a parameter estimator are presented. These studies are useful in that they can be used to help guide the designer in choosing the free parameters of the LMS algorithm for particular applications. A description of the techniques utilized for the computer simulations in this report is given in appendix I. The Fortran code for the LMS and IMLMS algorithms is given in appendix II.

A. SISO CHARACTERISTICS OF LMS ALGORITHM

A-1 First Order Demonstration of LMS Algorithm

In this subsection a first order model will be studied to illustrate the parameter estimation convergence rate of the LMS algorithm. Several control input strategies are used to excite the system.

The roll mode of an airplane is an example of a first order system which is spectrally separated from the remaining lateral-directional dynamics (much faster) and may require on-line identification since the damping is a function of angle-of-attack and dynamic pressure. The linear, time-invariant plant for this example is modeled by

$$\dot{x} = ax + bu + w \tag{3.1}$$

where the baseline value of a is $.5\ \text{sec}^{-1}$ and b is $2\ \text{sec}^{-1}$. Note that the time scale has been transformed for convenience as values for a of $10\ \text{sec}^{-1}$ are more typical for aircraft. The LMS filter equations become

$$\hat{x}(k+1) = \alpha(k)x(k) + \beta(k)u(k) \tag{3.2}$$

$$\varepsilon(k) = x(k) - \hat{x}(k) \tag{3.3}$$

$$\alpha(k+1) = \alpha(k) + 2\mu\varepsilon(k)x(k) \tag{3.4}$$

$$\beta(k+1) = \beta(k) + 2\mu\varepsilon(k)u(k) \tag{3.5}$$

A simulation was performed for 50 seconds at a sample rate of 5 Hz with the inputs scheduled by table 3.1. The step size factor, μ, was chosen to be .005. A step input was applied at t = 0; from 10 to 20 seconds a sine wave was superimposed; from 20 to 30 seconds a square wave was superimposed; from 30 to 40 seconds the control consists of discrete random inputs; and, from 40 to 50 seconds discrete process noise is added. Time histories of x and $\dot{x}$ are presented in figure (III-1) and the control input time history is shown in figure (III-2).

The time histories of the estimated parameters are depicted in figures (III-3) and (III-4). It is possible to verify some of the analytical results of the previous chapter. The denominator coefficient, α, moves in the wrong direction initially while the numerator coefficient, β, makes some improvement after a step input. Then α begins to move in the correct direction while β reaches a steady state. Little real benefit comes from the sine wave input while the square wave input yields the most improvement in the parameters. Random inputs have little impact for this case as signal-to-noise ratios of approximately 10 were used. A good rule of thumb is to expect parameter convergences with the LMS filter for signal-to-process-noise ratios of 2 or more and signal-to-measurement-noise ratios of 5 or more [33].

At the end of this simulation, the adapted parameters had the values shown in table 3.2. Fairly good agreement between the theoretical and experimental coefficients were obtained indicating that the identification inputs and other problem parameters were well

Table 3.1 — Input schedule for first order demonstration of LMS algorithm.

TIME	CONTROL INPUT	NOISE INPUT
t, sec	u	w
0 - 10	u_{con}	0
10 - 20	$u_{con} + u_{amp} \sin(\Omega-10)$	0
20 - 30	$u_{con} + u_{amp} \text{ sgn}(\sin(\Omega-10))$	0
30 - 40	$\eta(0,.1)$	0
40 - 50	0	$\eta(0,.1)$

Table 3.2 — Comparison of theoretical and experimentally determined parameters for first order demonstration of LMS algorithm.

PARAMETERS	THEORETICAL	EXPERIMENTAL
α	.9048	.9040
β	.3807	.3857

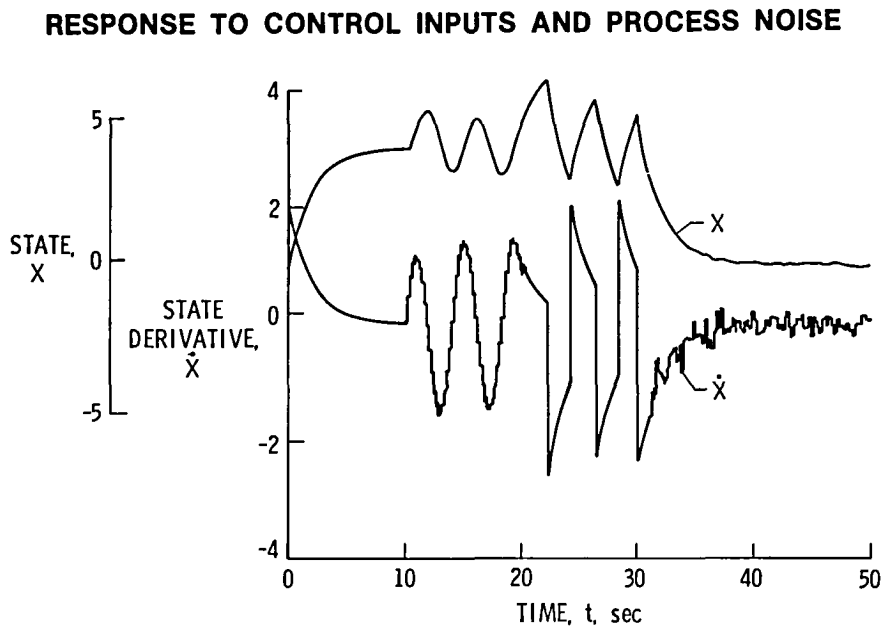

Figure III-1 — Time histories of state and state derivative for various control and noise inputs.

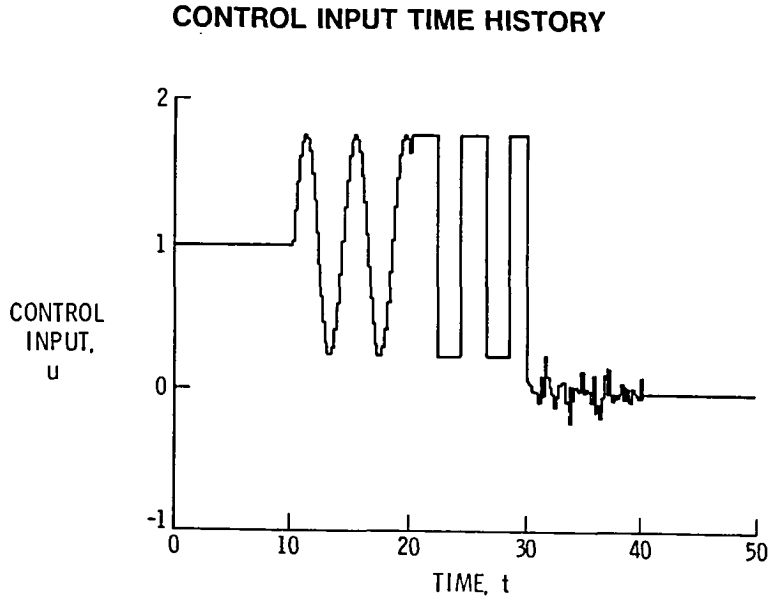

Figure III-2 — Time history of control input for first order demonstration of LMS filter.

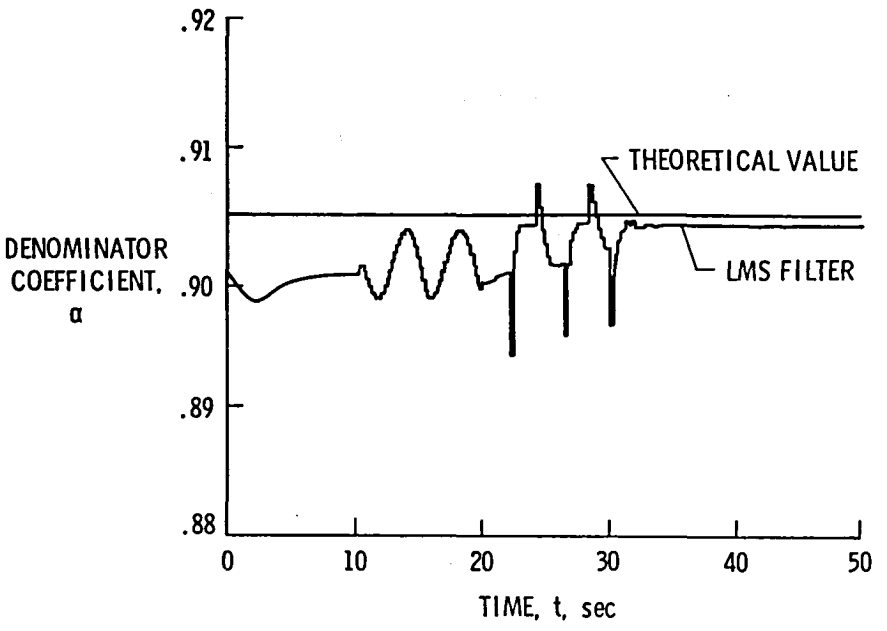

Figure III-3 — Time history of LMS filter denominator coefficient for first order demonstration.

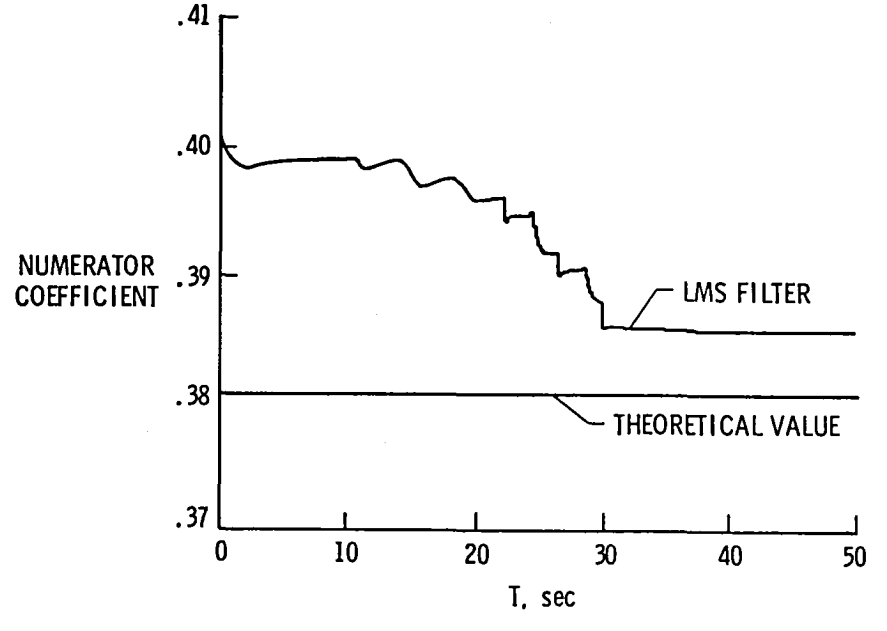

Figure III-4 — Time history of LMS filter numerator coefficient first order demonstration.

matched. In the next subsection, the impact of choosing these free parameters will be illustrated.

A-2 Study of Important Identification Parameters

The first order system of the previous subsection is studied further in this subsection. The importance of the problem parameters to the identification accuracy is determined by systematic variation of each parameter. The parameters considered and their nominal values are listed in table 3.3.

A normalized total parameter error, e_p, is defined by comparing the estimated parameter value, p_i, with the actual parameter, p_i, and summing for all parameters as follows

$$\bar{p}_i = \frac{p_i - \hat{p}_i}{p_i} \times 100\% \tag{3.6}$$

$$e_p = \sqrt{\sum_{i=1}^{n+m} \bar{p}_i^{\,2}} \tag{3.7}$$

In this case the parameter vector, $\underline{p}$, includes only two items

$$p_i = [\alpha, \beta]^T \tag{3.8}$$

The system is excited by a square wave for 40 seconds and then allowed to settle for 10 seconds. At the end of 50 seconds the estimated parameters, α and β, are compared with their theoretical values and the total percent error, e_p, is computed. This was done for several cases by fixing the values of the problem parameters to their nominal values and varying each one at a time.

The influence of varying μ, the step size factor of the LMS algorithm, is depicted in figure (III-5). It shows the expected trends. Above a certain value ($\mu \approx .8$ for this case) the algorithm diverges. Two minimum values occur, one for best fit of α and the other for β. This indicates that possibly a different μ may be desired for each coefficient. It is common to have a different μ

Table 3.3 — Nominal values of problem parameters first order studies of LMS algorithm.

PARAMETER	NOMINAL VALUE
a	-.5
b	2
μ	.005
u_{amp}	1.5
Ω	0.5
ω_{samp}	5 Hz
u_{con}	0.

IMPACT OF STEP SIZE FACTOR UPON CONVERGENCE OF LMS FILTER

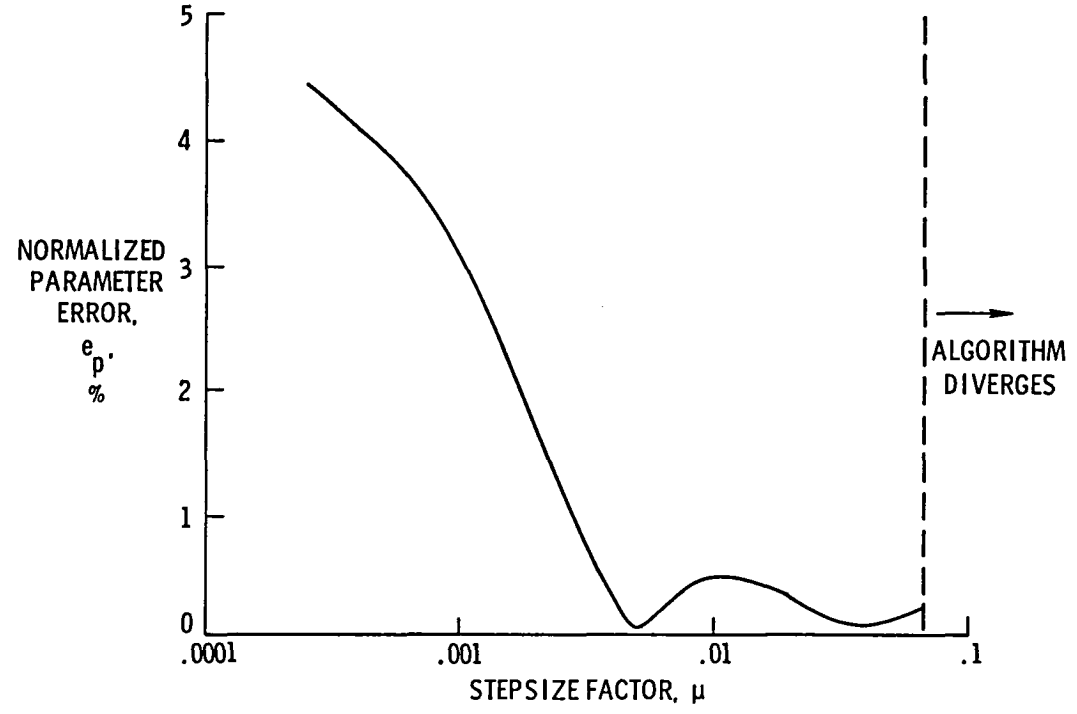

Figure III-5 — Normalized parameter error versus step size factor for first order demonstration of LMS filter.

IMPACT OF AMPLITUDE OF EXCITING INPUT UPON IDENTIFICATION ACCURACY

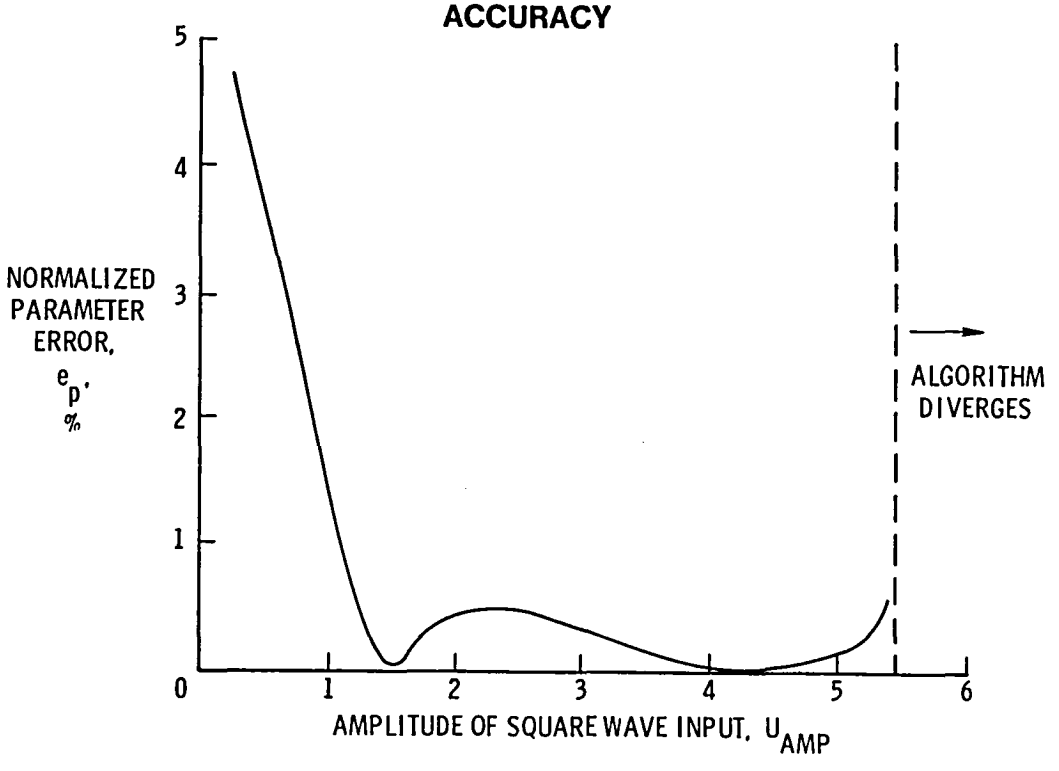

Figure III-6 — Normalized parameter errors versus amplitude of square wave input for first order demonstration of LMS filter.

for denominator terms and numerator terms for the SHARF and HARF algorithms [36]. If the value of μ is too small, the speed of adaption is slowed, preventing satisfactory parameter convergence. The choice of μ is highly problem dependent and is difficult to select optimally a priori.

The amplitude of square wave input has a similar double minimum in e_p as shown in figure (III-6). If the amplitude of input is too large, the algorithm diverges. This is equivalent to too large of a stepsize factor, as β over-corrects in equation (3.5). If the amplitude of the input is too small, sufficient information for β is not available and the system response, x, is too small for the nominal value of μ.

The parameter estimation error is very sensitive to the frequency of the square wave input as depicted in figure (III-7). The optimum frequency is at .5 rad/sec, which is the system time constant. Although a square wave has a wide spectrum of sine wave frequencies, it may be advantageous to select square wave inputs with varying frequencies. Reference [43] indicates that pulsing the controls hard against the stops may be optimum for many linear systems, with the switch time for optimum identification a function of system parameters. The data in figure (III-7) seem to be consistent with this result.

Although the data have inconsistent variations between 5 and 30 Hz, figure (III-8) verifies what is expected about system performance with respect to sample rate. Sampling too slowly results in estimation problems due to aliasing effects with respect to the input square wave frequency. Sampling too fast runs into numerical problems for the nominal values of problem parameters and the accuracy is somewhat reduced. However, it is wiser to sample too fast than too slow as the penalties are less and divergence is not a problem.

The plant parameters, a and b, were also varied and the estimated parameter errors are shown in figures (III-9) and (III-10). Changing a had a much bigger impact than changing b. If a is increased in magnitude beyond -1.5, the error becomes

IMPACT OF FREQUENCY OF SQUARE WAVE INPUT UPON IDENTIFICATION ACCURACY

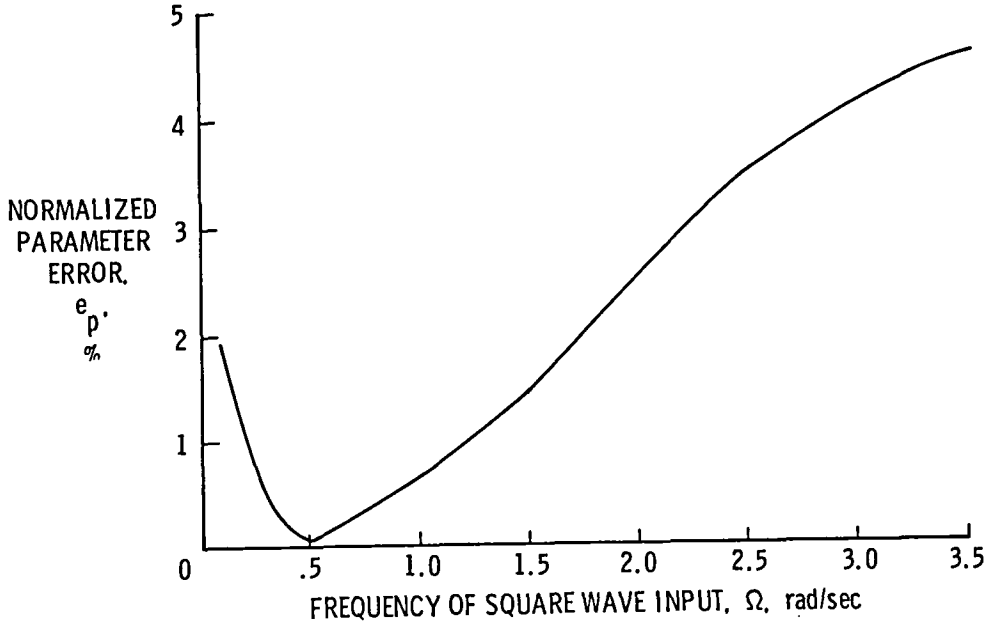

Figure III-7 — Normalized parameter error versus frequency of square wave input for first order demonstration of LMS filter.

IMPACT OF SAMPLE RATE UPON IDENTIFICATION ACCURACY

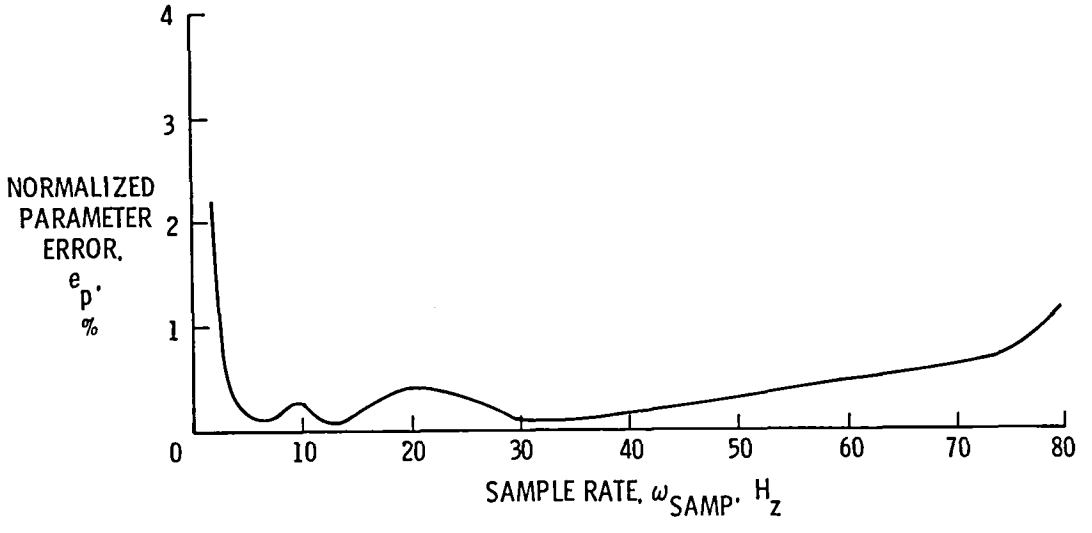

Figure III-8 — Normalized parameter error versus digital sample rate for first order demonstration of LMS filter.

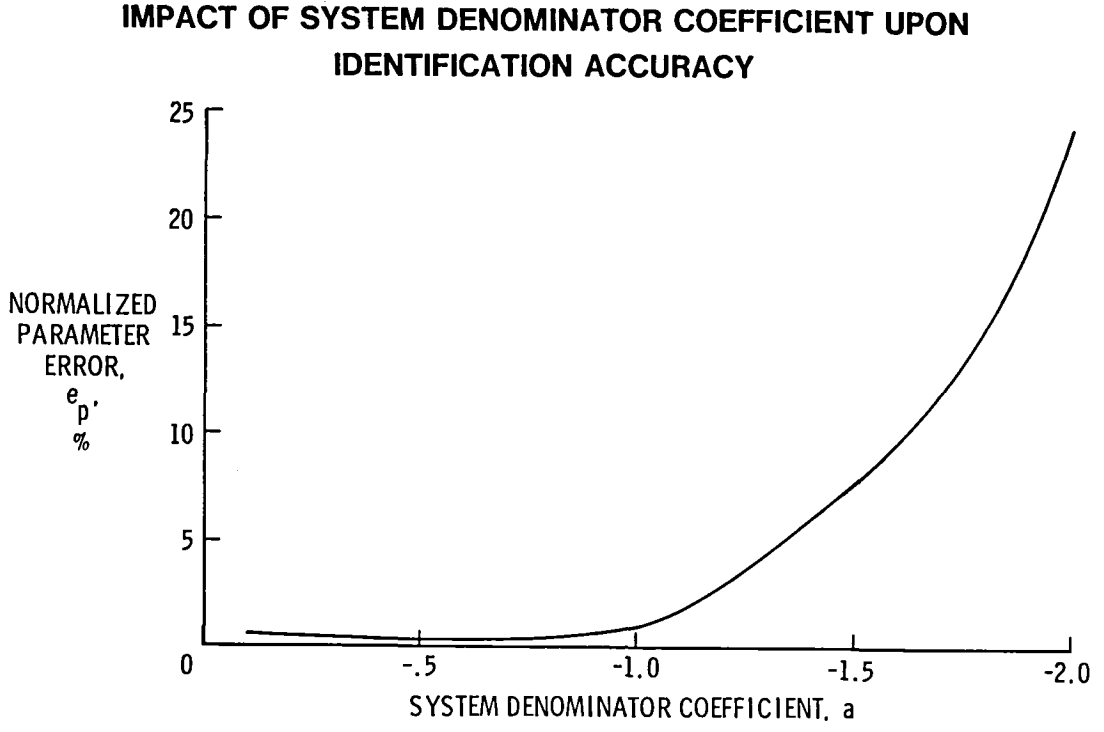

Figure III-9 — Normalized parameter errors versus plant denominator coefficient for first order demonstration of LMS filter.

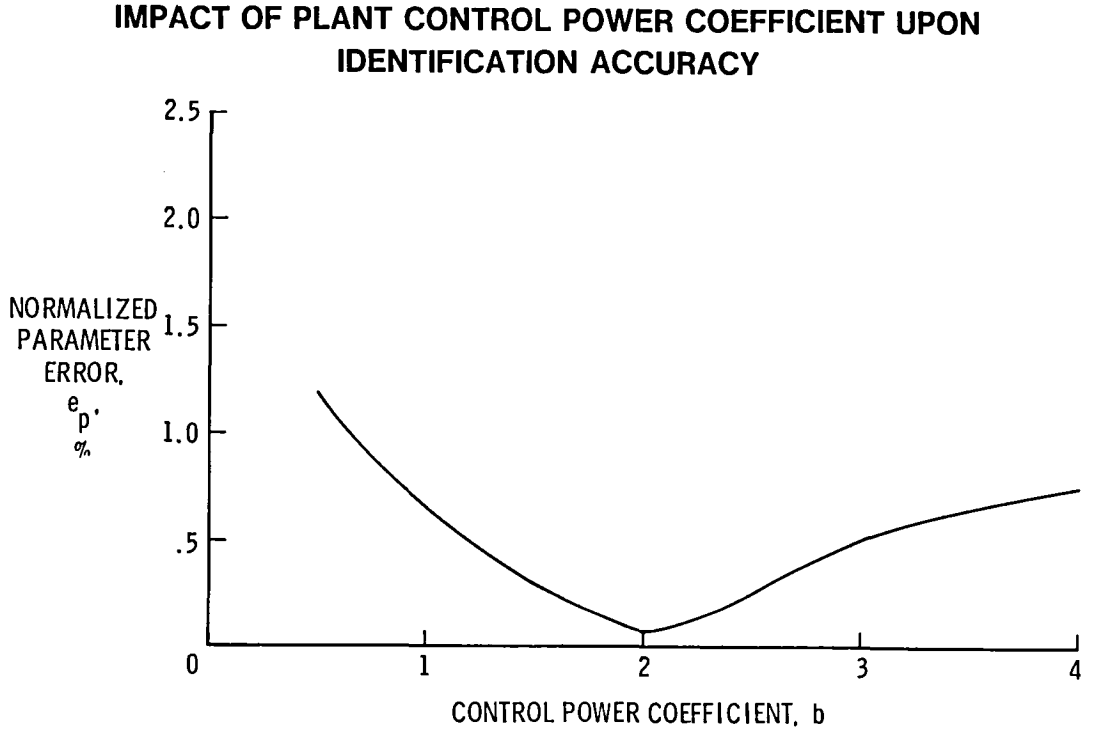

Figure III-10 — Normalized parameter errors versus plant control influence coefficient for first order demonstration of LMS filter.

significant, while fairly large changes in b have a much smaller impact upon the identification accuracy.

All of the trends mentioned in this subsection illustrate the importance of the input (noise or control) signal on identifiability. The designer must select input signals to sufficiently excite the system. In the absence of prior knowledge of system characteristics, it may be desirable to use digital implementation of learning concepts to vary the input signals. Such heuristic approaches are currently under consideration for adaptive control applications.

The studies of this subsection also show that the step size factor, μ, is of paramount importance for obtaining good performance of the LMS algorithm. It was observed that there are optimum values and that if the values are too large, divergence may occur. The step size factor may also be a prime candidate for systematic variations by computer learning logic. Separate μ's for each term being adapted may be appropriate.

A-3 Study of a Second Order System

In this subsection a second order system will be considered. The LMS algorithm implementation will have four terms to be identified: two denominator system terms; and, two numerator control influence terms. Reference [32] points out that the parameter error is a linear function of the number of parameters, so it is expected that the accuracy of parameter estimation may be reduced relative to the first order studies considered previously.

A typical second order system that is of concern to the airplane control system designer is the short period mode. It is spectrally separated from the phugoid mode and its damping ratio varies significantly with aircraft geometric characteristics and flight condition. Such a system could be generically modeled in modal coordinates as

$$\dot{\underline{x}} = A\underline{x} + Bu \tag{3.9}$$

$$y = H\underline{x} \tag{3.10}$$

where

$$u = \begin{cases} u_{amp}\ \text{sgn}\left(\sin(\Omega t)\right) & \text{If } \left|\sin(\Omega t)\right| \geq .707 \\ 0 & \text{If } \left|\sin(\Omega t)\right| < .707 \end{cases} \tag{3.11}$$

$$A = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -2\zeta\omega \end{bmatrix} \tag{3.12}$$

$$B = [0 \quad 1]^T \tag{3.13}$$

$$H = [1 \quad 0] \tag{3.14}$$

The control input (3.11) was a pulsed wave alternating between equal segments of u_{amp}, 0, and $-u_{amp}$. The nominal values for the design parameters were: μ=.005, u_{amp}=.5 sec^{-2}, Ω=1 rad/sec, ω_s=20 Hz and the length of the simulation was for 200 seconds. The plant natural frequency, ω, was 1 rad/sec. Only the damping factor, ζ, was varied. Its impact upon average parameter estimation accuracy is computed based upon the theoretical estimates for the parameters. The results are plotted in figure (III-11).

The lowest error for the denominator terms of the discrete transfer function occurred for a damping ratio of .5. The accuracy of the numerator terms increased with the damping ratio and was less than the denominator terms when the system was overdamped with ζ greater than 1.3. These results are consistent with the previous observations that the input signal is of prime importance for obtaining sufficient excitation for identification. The signal and the plant need to be matched for optimum estimation performance.

The denominator terms converge to accurate results more readily than the numerator terms when ζ is less than 1. This is a characteristic of using the LMS algorithm for ARMA type model implementations. As previously discussed, the improvements in the estimates of the numerator terms, the β's, comes within the first few samples of the step input. In contrast, the denominator terms, the α's,

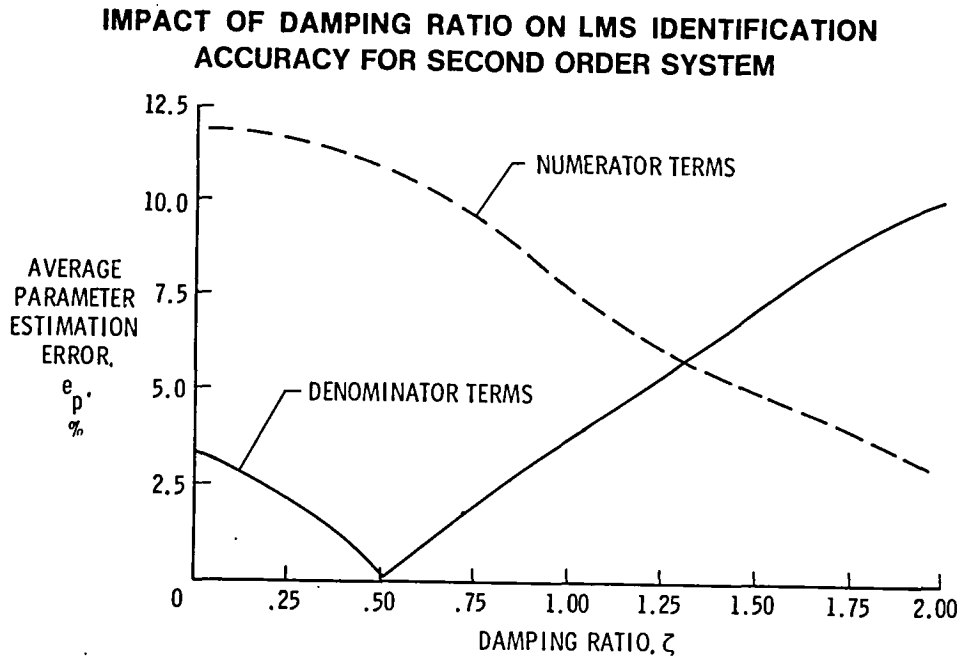

Figure III-11 — Average normalized parameter errors versus damping ratio for second order system.

make the majority of the improvement after the system has had time to react. This suggested a hybrid technique using both the recursive and the batch least squares identification schemes. This approach to improve the estimation of the control influence parameters was developed in section II-E.

A-4 Phase/Gain Evaluation of LMS Filter

Phase and gain plots of the LMS filter were obtained as a function of frequency. Such Bode plots are potentially useful for evaluating the effect of sampling rate on filter performance. These approaches may also be useful for determining the potential application of the LMS adaptive algorithm to nonlinear problems. In this study, the frequencies of the input signal and the sample rate were varied. If describing functions for nonlinear applications are desired, the magnitude of the input signal should be systematically varied as well.

The phase and gain analysis was performed by driving a unit magnitude sinusoidal signal at frequency Ω as the system output,

$$y(t) = \sin(\Omega t) \tag{3.15}$$

To estimate the output, a second order LMS filter was implemented at sample rate ω_s. The filter equations and parameter updates became

$$\hat{y}(k) = \alpha_1(k)y(k-1) + \alpha_2(k)y(k-2) \tag{3.16}$$

$$\varepsilon(k) = y(k) - \hat{y}(k) \tag{3.17}$$

$$\alpha_1(k+1) = \alpha_1(k) + 2\mu\varepsilon(k)y(k-1) \tag{3.18}$$

$$\alpha_2(k+1) = \alpha_2(k) + 2\mu\varepsilon(k)y(k-2) \tag{3.19}$$

The system signal was initiated and the LMS filter transients were allowed to die for 20 seconds or 3 periods of system signal,

whichever was greater. The in-phase and quadrature components were computed, respectively, for one cycle as:

$$P = \frac{\Omega}{\pi} \int^{T} y(t)\sin(\Omega t)dt \tag{3.20}$$

$$Q = \frac{\Omega}{\pi} \int^{T} y(t)\cos(\Omega t)dt \tag{3.21}$$

The phase and gain were computed by the following relations

$$\phi = \tan^{-1}\left(\frac{Q}{P}\right) \tag{3.22}$$

$$M = 10 \log_{10}[Q^2 + P^2] \tag{3.23}$$

The RMS error of prediction was also computed during the period of phase and gain evaluation.

Figure (III-12) is a plot of phase and gain as a function of frequency at a sample rate of 25 Hz. The phase maintains good agreement ($|\phi| < 15$ degree) up to an input signal frequency of about 30 rad/sec. The gain remains relatively flat up through about 20 rad/sec. Figure (III-13) shows a plot of the RMS error of prediction, confirming the region of difficulty at frequencies greater than 20 rad/sec.

As expected, it was observed that the sample rate is a prime factor in determining the frequency where the performance of the filter begins to degrade. Figure (III-14) is a plot of the ratio of frequency for a 15 degree phase margin to sample rate as a function of sample rate. A ratio of 1/4 to 1/10 bounds most of the answers, so it seems that good design practice would call for using a sample rate of 10 times the highest frequency being modeled by the LMS adaptive algorithm.

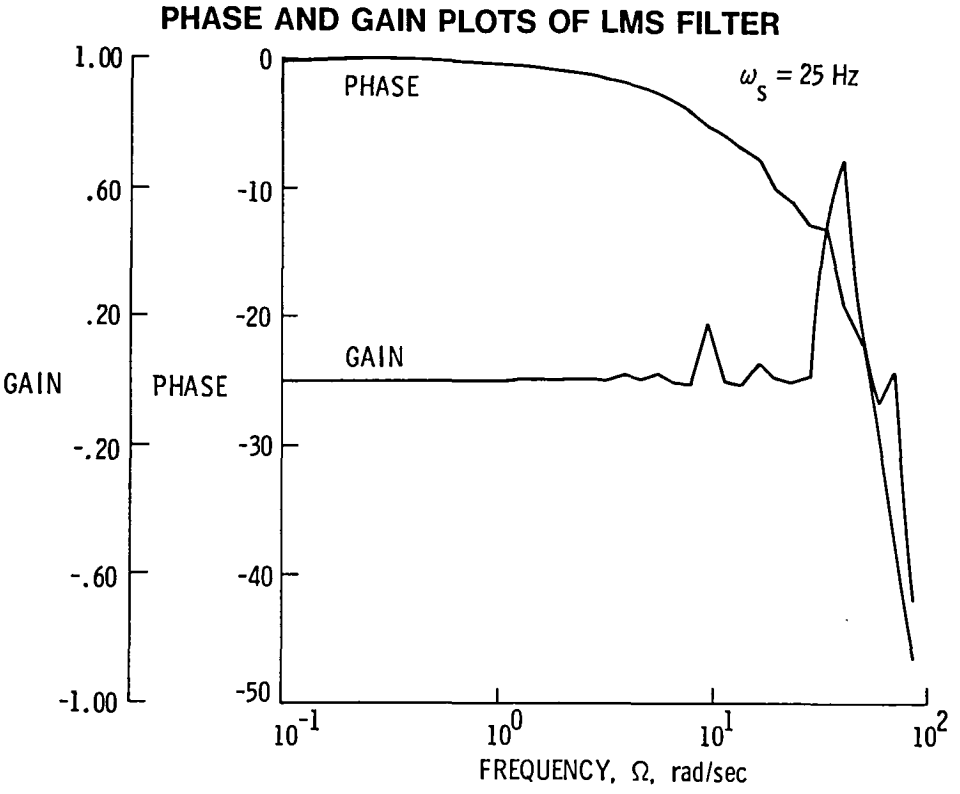

Figure III-12 — Phase and gain plots of second order LMS filter as a function of input signal frequency at a sample rate of 25 Hz.

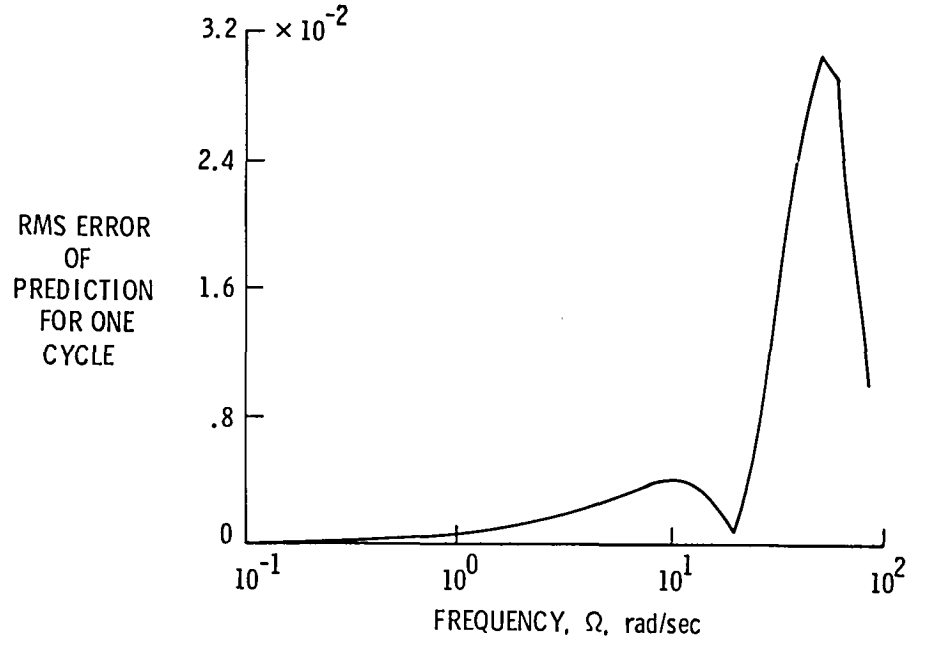

Figure III-13 — RMS error of prediction for second order LMS filter as a function of input signal frequency at a sample rate of 25 Hz.

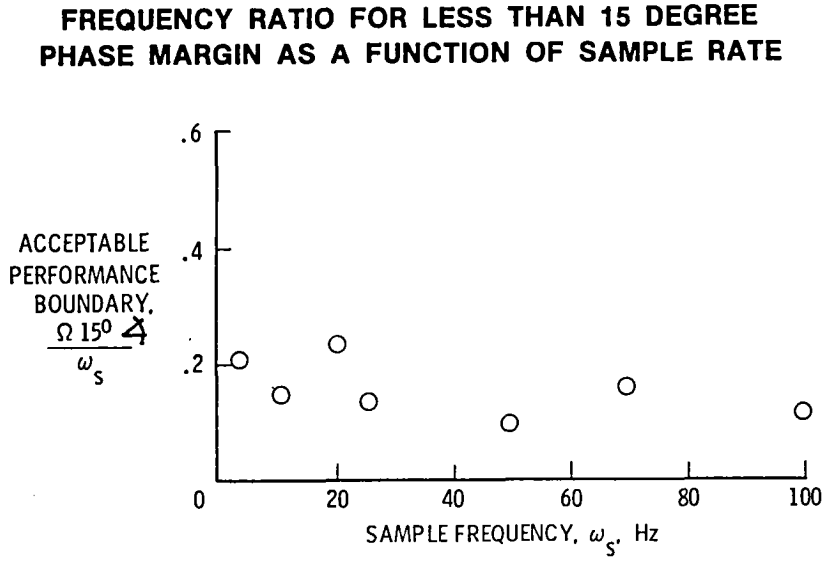

Figure III-14 — Frequency ratio for satisfying 15 degree phase margin for second order LMS filter as a function of sample rate.

B. MIMO CHARACTERISTICS OF LMS ALGORITHM

A number of simulations were performed studying multi-input, multi-output identification to verify the multivariable formulation of the LMS algorithm developed in section II-C. No unusual characteristics were observed; as the number of parameters being identified increased, the accuracy decreased as predicted.

As previously mentioned, the mean parameter error increases linearly with the number of parameters being identified. There are 2 parameters for the first order system of subsection III-A.1, 4 parameters for the second order SISO system of subsection III-A.3 and there are 10 parameters for a second order system with 2 inputs and 2 outputs. There are $n+nrm$ parameters in a MIMO system of order n with m inputs and r outputs. Clearly, the number of parameters grows significantly for MIMO systems.

During this research, systems were studied ranging from first order to tenth order; some were SISO and some had 2 inputs and 2 outputs. Although the results are somewhat inconsistent (no effort was made to systematically match sampling rates, duration of adaption, plant dynamics or step size factors) a plot of average parameter error is shown as figure (III-15). It is a plot of what is supposed as the best achievable for each problem. There is significant scatter in the data, but the anticipated trends are observable. One can expect a degradation in parameter estimation accuracy as the number of parameters increases. This is another good argument for using the IMLMS formulation as it allows for fewer parameters to be estimated at each stage of the model building.

Figure (III-15) also illustrates the fact that the numerator terms converge slower in time and have much larger parameter errors. Inaccuracy is substantially reduced by using the recursive LMS algorithm in conjunction with the batch least squares identification for the numerator terms as was developed in section II-E.

PARAMETER ERROR PERFORMANCE AS A FUNCTION OF THE NUMBER OF PARAMETERS

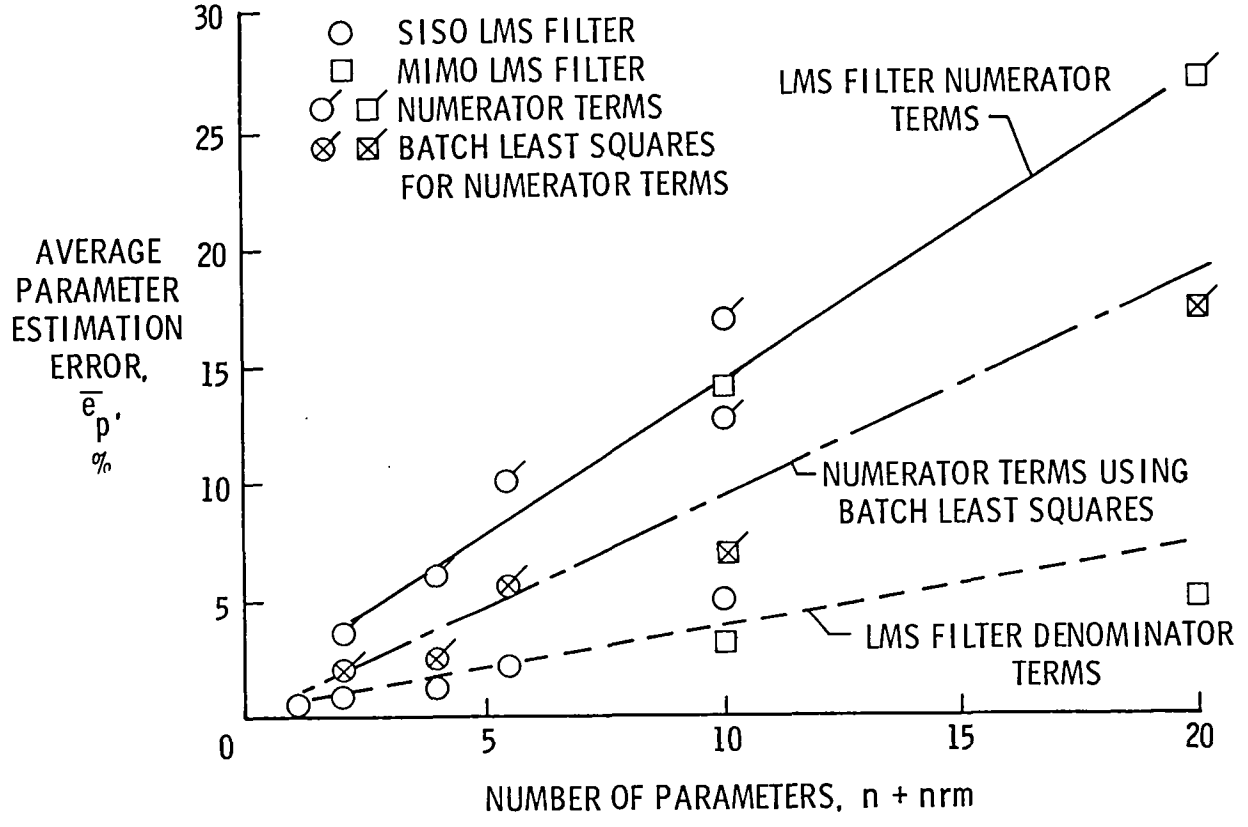

Figure III-15 — Average normalized parameter errors as a function of the number of model parameters.

C. CONVERGENCE IN PRESENCE OF UNMODELED MODES

References [31-33,39-40,48-50] have indicated that LMS adaptive filters converge well even when used with mismatched model order. A simulation study to verify these assertions will be presented in this section. A fourth order SISO plant is modeled as equations (3.9) and (3.10) with

$$A = \begin{bmatrix} A_1(\omega_1,\zeta_1) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & A_2(\omega_2,\zeta_2) \end{bmatrix} \tag{3.23}$$

$$B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \tag{3.24}$$

$$H = [H_1, H_2] \tag{3.25}$$

The submatrices are defined as

$$A_i(\omega_i,\zeta_i) = \begin{bmatrix} 0 & 1 \\ -\omega_i^2 & -2\zeta_i\omega_i \end{bmatrix} \tag{3.26}$$

$$B_i = \begin{bmatrix} 0 \\ b_i \end{bmatrix} \tag{3.27}$$

$$H_i = [h_i \quad 0] \tag{3.28}$$

Measurement and control influence terms are normalized by the first mode to define

$$\bar{h} = \frac{h_2}{h_1} \tag{3.29}$$

$$\overline{b} = \frac{b_2}{b_1} \tag{3.30}$$

The input signal is the pulsed wave of equation (3.11) where $u_{amp} = .5$, $\Omega = 1$ rad/sec and the sample rate, ω_s, is 20 Hz. The simulations were run for 200 seconds with $\mu = .008$ and for all runs $\omega_1 = 1$ rad/sec and $h_1 = b_1 = 1$.

The LMS adaptive filter was implemented assuming only one mode and the parameters of the second mode were varied. Identification accuracy was found by comparing the experimental LMS coefficients with the theoretical discrete coefficients of mode 1. A summary of the results with $\zeta_1 = \zeta_2 = .5$ is plotted as figure (III-16).

The total percentage error for the denominator terms is shown in figure (III-16) as a function of the frequency of the unmodelled mode, ω_2. The LMS filter does remarkably well for ω_2 greater than 2. Looking at the curves with $\overline{h} = 2$ and greater, it is apparent that the LMS filter tries to identify the lowest frequency mode until the signal power of the new mode becomes prohibitively strong. As the frequency of the unmodeled mode approaches the frequency of the modeled mode, 1 rad/sec, the parameter error grows. When $\overline{h} > 2$, the parameter estimate near 1 rad/sec is nearly divergent as the LMS algorithm begins to follow the second mode.

A survey was done varying ζ_1 and ζ_2 but making them equal, and the same shape curves as figure (III-16) were obtained. However, the actual parameter estimation accuracy was dependent upon the input sequence used to excite the system. With the same input as was used for generating figure (III-16) and with varying the damping ratios between .005 and 1.0, the asymptotic total parameter estimation error varied between .12 percent and 5.9 percent.

If the damping ratios of the two systems are not held constant, the impact of the input signal is felt more strongly. However, the same trends are apparent. If the spectral separation between two modes is wide and they have nearly equal signal power, the LMS filter can be expected to perform well in estimating the values of the parameters of the lowest frequency mode.

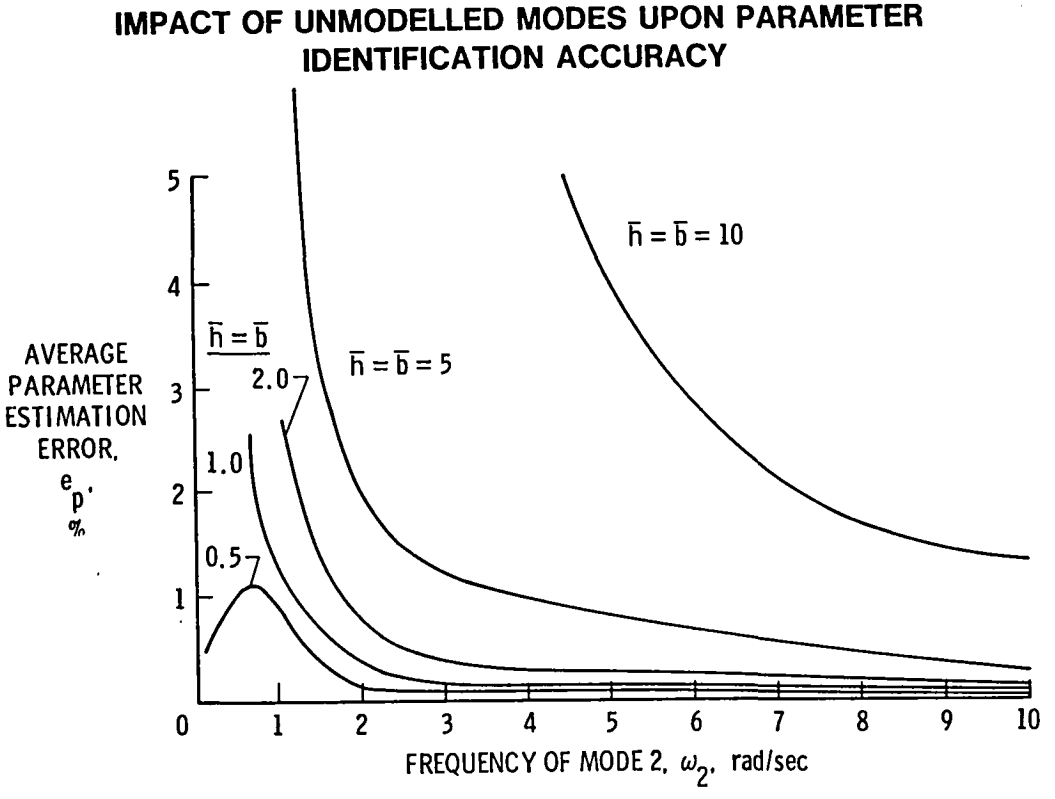

Figure III-16 — Total parameter estimation error versus the frequency of the unmodeled mode.

D. IMLMS CONVERGENCE CHARACTERISTICS

The Incremental Mode LMS (IMLMS) algorithm was derived in section II-F and is useful because only part of the model is adapted. It allows the known part to be held constant while the unknown part is adjusted. This provides a natural way to perform model building--that is, by adding one mode at a time. Although no additional assumptions were needed to prove convergence of the IMLMS filter compared to the LMS algorithm, it is still of interest to verify the performance of the parameter estimates.

The generic sixth order plant is modeled by assuming the linear, time-invariant, state space representation of equations (3.9) and (3.10), with A in modal form:

$$A = \begin{bmatrix} A_1 & & 0 \\ & A_2 & \\ 0 & & A_3 \end{bmatrix} \tag{3.31}$$

$$B = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix} \tag{3.32}$$

$$H = [H_1, H_2, H_3] \tag{3.33}$$

The submatrices are defined by equations (3.26-3.28). For this simulation the plant parameters were selected as follows:

$$\omega_{1,2,3} = 1.,4.,8. \text{ ras/sec}$$

$$b_{1,2,3} = h_{1,2,3} = 1.,2.5,1.$$

$$\zeta_{1,2,3} = .1,.1,.1$$

The sample rate, ω_s, is 10 Hz so that $\omega_s \gg \omega_3$ and the input signal is a pulsed wave given by equation (3.11) with u_{amp}=.5 and Ω=1 rad/sec.

Time histories for the simulation are shown in figures (III-17) through (III-20). It is assumed that the first mode is known exactly and the IMLMS algorithm is used to identify the second mode. The third mode is not modelled and is treated as process noise similar to the previous section. Since the spectral separation between modes 2 and 3 is 4 rad/sec or a ratio of 1 to 2, it is expected that good parameter estimation is possible. The output, y, the control input, u, and the modal positions, x_1, x_3, and x_5 are plotted in figures (III-17), (III-18) and (III-19), respectively.

The recursive parameter estimates for the two denominator parameters of the second mode are shown in figures (III-20) and (III-21). The parameters obtained 90 percent of their asymptotic limits within about 3 or 4 periods of oscillation of the second mode. The total parameter error was small (.23 percent). There was no tendency toward a bias and it converged quickly, even in the presence of another mode. So as predicted, this illustrates parameter convergence properties similar to the LMS algorithm for the same number of parameters being adapted.

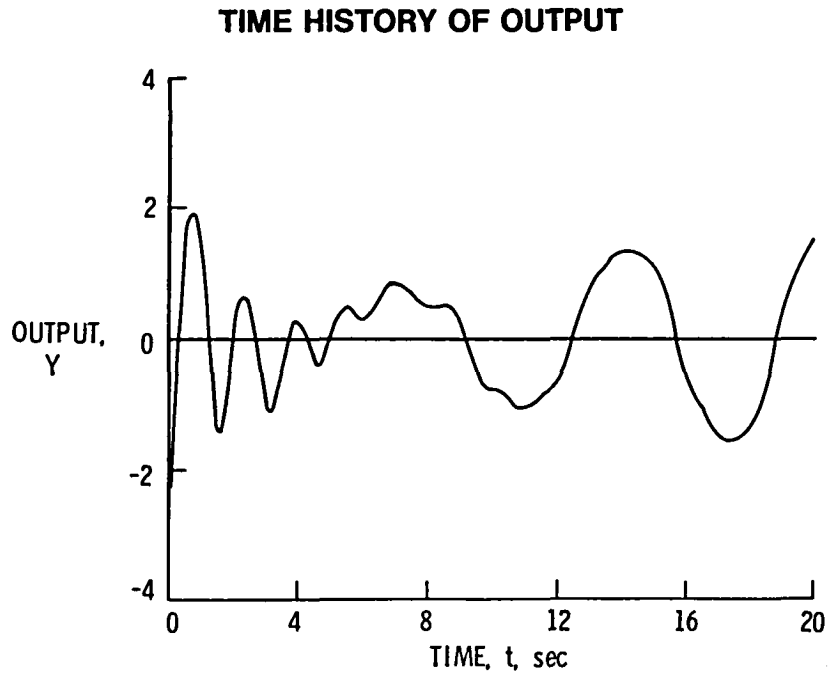

Figure III-17 — Time history output of third order system used for studying IMLMS algorithm.

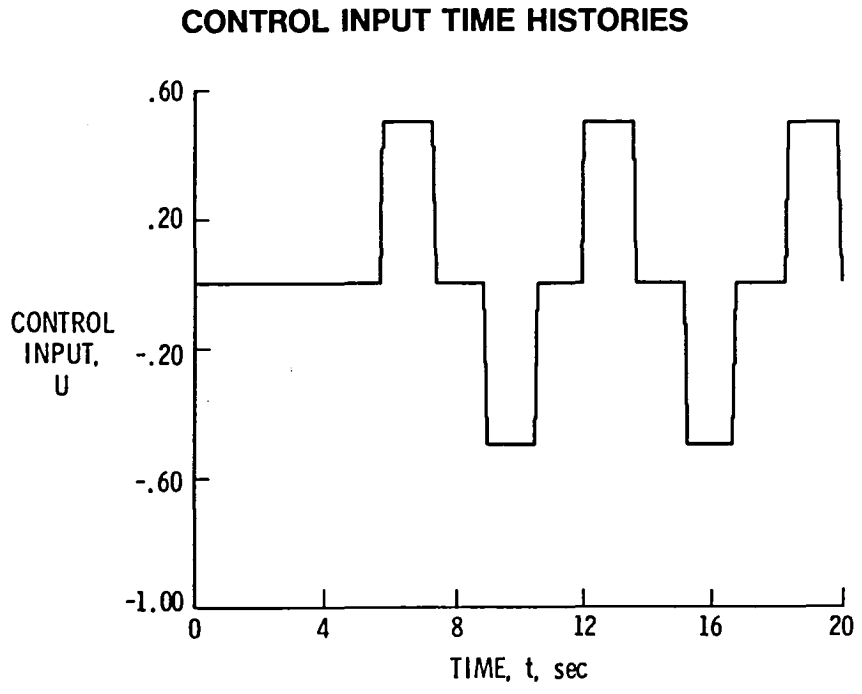

Figure III-18 — Time history of control input used to excite third order system for studying IMLMS algorithm.

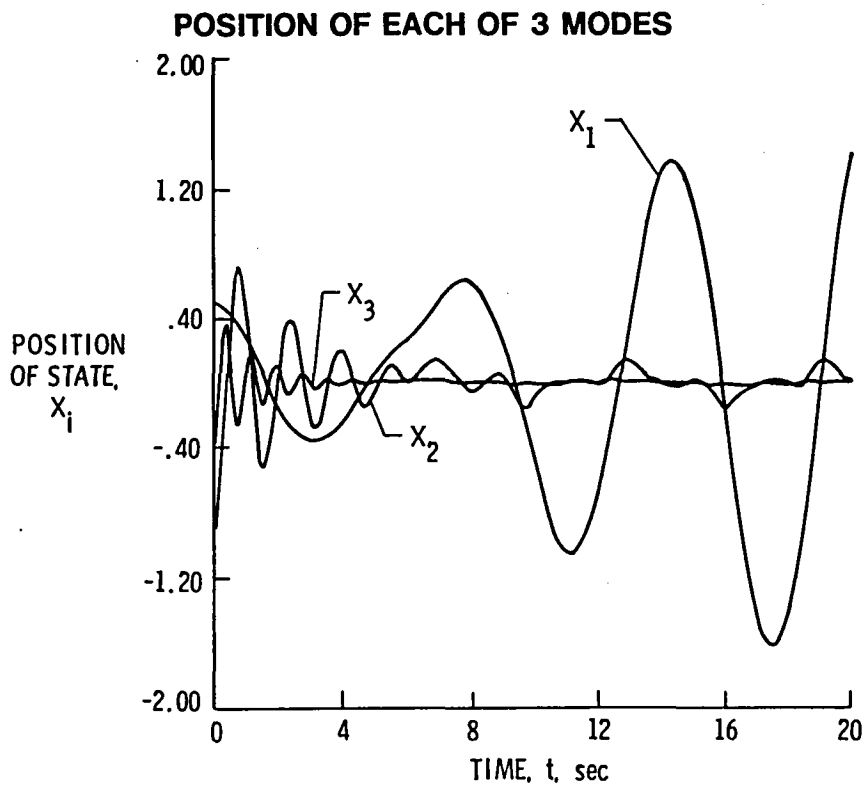

Figure III-19 — Time history of the positions of each of three modes during simulation used to study IMLMS algorithm.

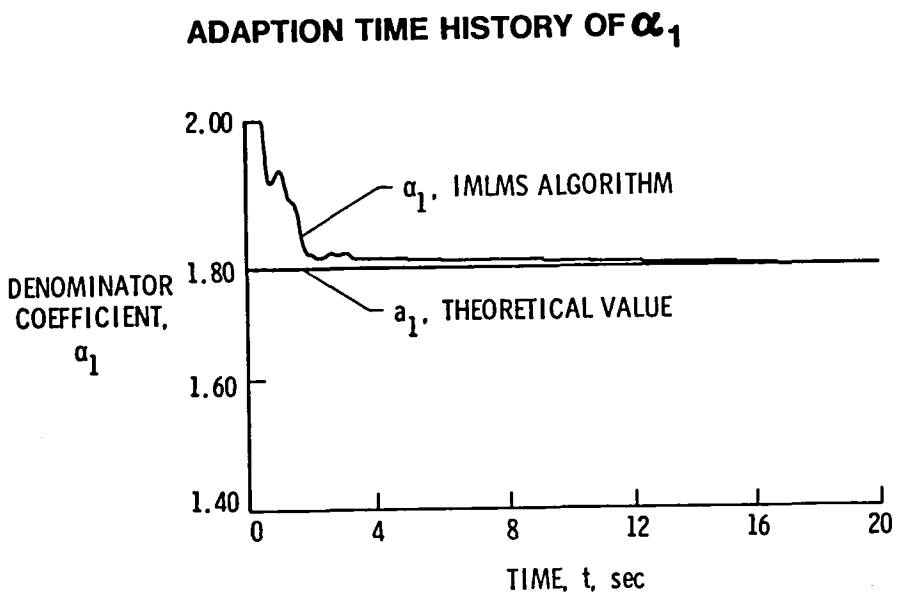

Figure III-20 — Time history of denominator coefficient, α_1, for IMLMS algorithm demonstration.

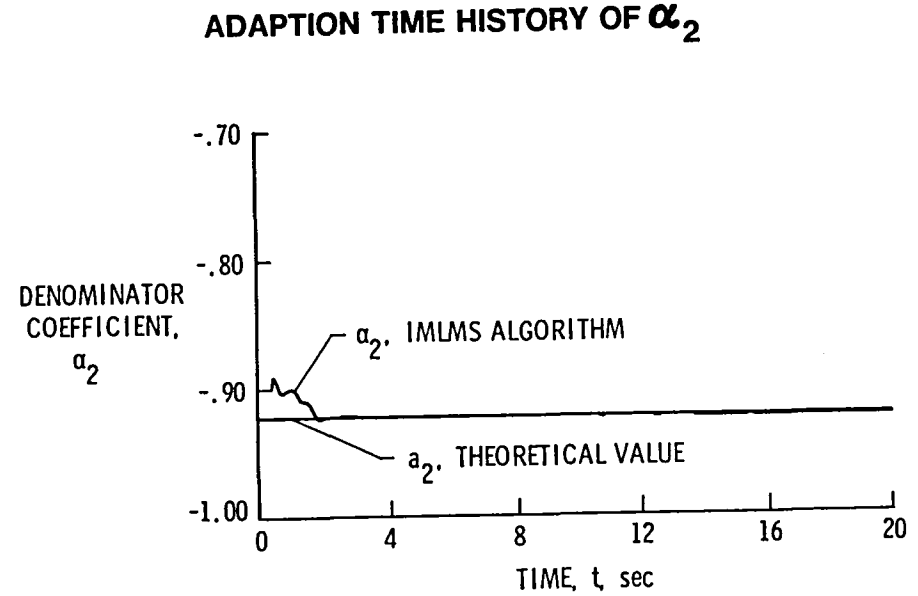

Figure III-21 — Time history of denominator coefficient, α_2, for IMLMS algorithm demonstration.

Chapter IV

STATE SPACE REALIZATIONS FROM MIMO TRANSFER FUNCTIONS

A. PROBLEM DESCRIPTION

The multi-input, multi-output (MIMO) version of the LMS and IMLMS adaptive filters yield the coefficients for the following discrete ARMA model in indicial notation

$$\hat{y}_{\ell}(k) = \alpha_i y_{\ell}(k-i) + \beta_{i\ell j} u_j(k-i) \tag{4.1}$$

This represents a set of discrete transfer functions [21]

$$H(z) = \frac{\sum_{i=1}^{n} B_i z^{-i}}{1 - \sum_{i=1}^{n} \alpha_i^{-i}} \tag{4.2}$$

The compensator portion of the adaptive control scheme proposed here will be implemented on-line using optimal control techniques. This requires a state space realization of the plant model to take advantage of the computer codes that are available for optimal design of linear quadratic gaussian (LQG) systems [51].

Finding a state space realization given a single-input, single-output (SISO) transfer function is straight forward [52]. However, finding a state-space realization given a MIMO transfer function as a state space representation is considerably more complicated. Several methods have been proposed [53,54], but they tend to be effective only when the coefficients of (4.2) are exact. The coefficients of (4.2) as determined by the LMS or IMLMS algorithms will not be exact. Additionally, a truncated, equivalent system of a higher order system may be desired.

A robust algorithm for finding minimal state space realizations was proposed in reference [55]. However, it is costly in terms of computational burden as it requires nonlinear programming techniques. In section C an improved version of the algorithm is developed which greatly reduces the required computations. A further refinement is proposed in section D for the special case of a single mode with 2 inputs and 2 outputs (a 2x2x2 system). A partitioned linear algorithm is derived in section E which requires no iterations. In the final section of this chapter, some examples are given.

B. ALGORITHM FOR STATE SPACE REALIZATIONS

The algorithm of reference [55] is developed below in the s-domain, but it is easily extended to the z-domain. It converges only for stable systems (poles inside the unit circle).

The transfer function matrix is represented as

$$P(s) = \frac{N(s)}{d(s)} \quad , \tag{4.3}$$

and a state space realization is desired

$$P(s) = H(sI-A)^{-1}B + L \quad , \tag{4.4}$$

where A is a nxn matrix, B is a nxm matrix, H is a rxn matrix and L is a rxm matrix. P(s) is proper and rational and d(s) is the least common denominator of order n.

L is found in the usual way

$$L = \lim_{s\to\infty} P(s) \quad . \tag{4.5}$$

The ARMA representations in this research are usually assumed to be of unit delay, so that L = 0. Define the following

$$M(s) \equiv N(s) - Ld(s) \tag{4.6}$$

therefore,

$$M(s) = H(sI-A)^{-1}Bd(s) \tag{4.7}$$

The denominator can be represented as

$$d(s) = a_0 s^n + a_1 s^{n-1} + a_2 s^{n-2} + \ldots + a_n s^0 \tag{4.8}$$

or by the summation

$$d(s) = \sum_{i=0}^{n} a_{n-i} s^i \tag{4.9}$$

The term $(sI-A)^{-1}$ can be represented by the exponential series given by [52]

$$(sI-A)^{-1} = \frac{1}{s} \sum_{k=0}^{\infty} s^{-k} A^k \tag{4.10}$$

Substituting (4.9) and (4.10) into (4.7) yields

$$M(s) = \frac{1}{s} H \sum_{k=0}^{\infty} s^{-k} A^k B \sum_{i=0}^{n} a_{n-1} s^i \tag{4.11}$$

which also equals

$$M(s) = H \sum_{i=0}^{n} \sum_{k=0}^{\infty} s^{i-k-1} A^k a_{n-i} B \tag{4.12}$$

If the following definition is made

$$k' \equiv -k+i-1 \ , \tag{4.13}$$

then

$$k = -k'+i-1 \ , \tag{4.14}$$

and

$$M(s) = H \sum_{i=0}^{n} \sum_{-k'+i-1=0}^{\infty} s^{k'} A^{-k+i-1} a_{n-i} \ . \tag{4.15}$$

Adjusting the k' summation gives

$$M(s) = H \sum_{i=0}^{n} \sum_{k'=i-1}^{-\infty} s^{k'} A^{-k'+i-1} a_{n-i} \ B \ . \tag{4.16}$$

Breaking (4.16) into positive and negative summations of k' gives

$$M(s) = H \sum_{i=0}^{n} \sum_{k'=i-1}^{0} s^{k'} A^{-k'+i-1} a_{n-i}$$

$$+ \sum_{i=0}^{n} \sum_{k'=-1}^{-\infty} s^{k'} A^{-k'+i-1} a_{n-i} \ B \tag{4.17}$$

Note that a zero i gives a negative k'. So in the positive k' summation, i goes from 1 to n; and in the negative k' summation, i goes from 0 to n. The negative summation of equation (4.17) equals

$$\sum_{k'=-1}^{-\infty} s^{k'} A^{-k'-1} \sum_{i=0}^{n} A^{i} a_{n-i} \tag{4.18}$$

But

$$\sum_{i=0}^{n} A^i a_{n-1} = Ia_n + Aa_{n-1} + \ldots A^n a_0 = 0 \tag{4.19}$$

Equation (4.19) is equal to zero by the Cayley-Hamilton Theorem [52] since $d(s)$ is the minimal polynomial of $P(s)$. Therefore, equation (4.17) becomes

$$M(s) = H \sum_{i=1}^{n} \sum_{k'=0}^{i-1} s^{k'} A^{-k'+i-1} a_{n-i} B \tag{4.20}$$

Equation (4.20) is solved iteratively by specifying a set of residuals, $R(s)$, for minimization using nonlinear programming techniques.

$$R(s) = M(s) - \sum_{k=0}^{n-1} s^k \sum_{i=k}^{n-1} a_{i-k} HA^{n-1-i} B \tag{4.21}$$

Ideally the elements of $R(s)$ approach zero at a solution determined by the numerical procedures outlined below.

Developing the following indicial notation

$$R(s) = R_0 s^{n-1} + R_1 s^{n-2} + \ldots + R_{n-1} s^0 \tag{4.22}$$

$$R_i = \begin{bmatrix} r_{i11} & r_{i12} \cdots r_{i1j} \cdots r_{i1j} \\ r_{i21} & \ddots \\ \vdots & \vdots \\ r_{ik1} & \\ r_{ir1} & r_{ir2} \cdots r_{ikj} \cdots r_{irm} \end{bmatrix} \tag{4.23}$$

and if M utilizes the same notation, we can equate indices in (4.21) and write

$$s^{n-i} r_{ikj} = s^{n-i} m_{ikj} - s^{n-i} \sum_{\ell=1}^{n-1} a_{\ell-i} HA^{n-1-\ell} B \qquad (4.24)$$

It is desired to make the sum of the squares of the residuals, r^2_{ikj}, as small as possible. Since values of m_{ikj} may vary significantly in magnitude, causing numerical problems, (4.24) is normalized by m_{ikj} (in the computer code, m_{ikj} is set to 1 if less that 10^{-6}). Normalizing and equating terms of like powers of s^{n-i} gives

$$\bar{r}_{ikj} = \frac{m_{ikj} - \sum_{\ell=i}^{n-1} a_{\ell-i} HA^{n-1-\ell} B}{m_{ikj}} \; . \qquad (4.25)$$

The implied summation limits for i, j and k are 0 to n-1, 1 to m, and 1 to r. Hence, (4.25) implies that nrm equations are required for the minimization of r^2_{ikj}.

A unique solution of (4.20) is obtained through the minimization of r^2_{ikj} from (4.25) by choosing an A matrix with the proper eigenvalues from the coefficients (4.9) and by specifying one nonzero element in each row of B or each column of H. The remaining elements of B and H are found iteratively using nonlinear programming techniques.

For convenience, the A matrix is chosen in observer canonical form [21,52] and the elements of the first row of H are chosen as unity. Equation (4.22) is solved computationally by using Stewart's adaption of the Davidon-Fletcher-Powell (SDFP) algorithm [56,57], where the gradient is computed by numerical differentiation. The minimization is performed over (m+r-1)n optimization variables using a computer code for the SFDP algorithm available at Langley Research Center.

Initial implementations of this multivariable state space algorithm were found to be extremely sensitive to initial conditions. Two solutions were adopted to improve operating

characteristics under varying initial conditions: 1) If a satisfactory convergence is not obtained in a certain number of function calls, the routine restarts itself with a new set of parameters chosen randomly between some preselected boundaries; and, 2) an optimization variable transformation was used to scale the variables and apply constraints to the region that the search is allowed to take place over.

The optimization variable transformation [58] from the parameters, p_i, of H and B to the optimization variables, z_i, is given by

$$p_i = \frac{(U_i - L_i)}{2} \sin\left(\frac{\pi}{2} z_i\right) + \left(\frac{U_i + L_i}{2}\right) \tag{4.26}$$

where z_i is the ith variable the SDFP code optimizes, U_i and L_i are the upper and lower values, respectively, that are constraining the ith parameter, p_i. The inverse transformation is clearly given by

$$z_i = \frac{2}{\pi} \sin^{-1}\left(\frac{p_i - \left(\frac{U_i + L_i}{2}\right)}{\frac{(U_i - L_i)}{2}} \right) \tag{4.27}$$

The key feature of this transformation is that the optimization codes search parameters over a range of -1 to +1. Furthermore, any value of z_i will always return a value for p_i in the range L_i to U_i. Hence, the code optimizes variables of approximately the same size and the p_i's have been effectively constrained to lie between L_i and U_i.

In the next three sections, further improvements in the algorithm are developed. Section F of this chapter gives examples illustrating the use of these algorithms.

C. MODIFIED ALGORITHM BY PARTITIONING

An algorithm for obtaining state space realizations from multivariable transfer functions [55] was developed in the previous section. It is noted, however, that the solution of equation (4.20) through the normalized error residuals of (4.25) involves the use of nonlinear programming with the optimization code searching over (m+r-1)n variables. In the simplest useful case of a 2x2x2 system (nxrxm), there are 6 degrees-of-freedom the program must search over. A typical representation for the longitudinal motions of an airplane would require 12 parameters (4x2x2). The overall adaptive control algorithm should be able to translate the parameters estimated by the LMS or IMLMS filters into state space realizations in real-time on microprocessors prior to the suboptimal compensator design. Performing a search over this many parameters would be quite slow.

In this section an improvement to the algorithm of [54] is proposed that reduces the execution time by reducing the number of degrees of freedom. The approach used is to partition the problem into a linear part and a nonlinear part. The linear part is solved with simple matrix operations using the current values of the iterated nonlinear part. In this fashion the size of the nonlinear portion of the problem is reduced.

Take the matrix equation of (4.25) for the s^0 term of the numerator (i=0)

$$\bar{R}_0 = \frac{M_0 - \sum_{\ell=0}^{n} a_\ell HA^{n-1-\ell}B}{M_0} \tag{4.28}$$

Since it is desired that R_0 approach zero, we can rewrite (4.28) as

$$M_0 \approx \sum_{\ell=0}^{n} a_\ell HA^{n-1-\ell}B \tag{4.29}$$

which is just a different form for one power of s from (4.20). Assume H is known, then (4.29) can be solved for B

$$B \simeq \sum_{\ell=0}^{n} \left(a_\ell H A^{n-1-\ell}\right)^+ M_0 \tag{4.30}$$

The $()^+$ indicates a matrix pseudo inverse, which is needed if r and m are not equal. In the likely event that r and m are equal, then a simple matrix inverse is needed.

Equation (4.30) implies that once the elements of H have been chosen, B can be computed using linear algebra. Therefore, the nonlinear optimization code now only has to search over the elements of H; and, B is computed by (4.30). Hence, the solution of (4.20) has been effectively partitioned into a nonlinear part (finding the remaining elements of H) and a linear part (finding the elements of B).

The degrees of freedom for the algorithm of [54] have been reduced from $(r+m-1)n$ to $(r-1)n$. In the case of a 2x2x2 example, this corresponds to a reduction from 6 degrees-of-freedom to 2, which would translate into more than a 66 percent savings in execution time for the optimization code. The 4x2x2 example is reduced from 12 to 4 parameters. It is anticipated that this would correspond to a reduction of nearly 85 percent in execution time. These savings are significant, especially when on-line implementations are considered. Additionally, the accuracy of the algorithm can be expected to improve with a reduction in the number of free parameters.

It may be advantageous to alternatively specify n nonzero elements in B and find the remaining elements in B by nonlinear optimization. Equation (4.29) can then be rearranged for a linear solution of H

$$H \simeq M_0 \left(\sum_{\ell=0}^{n} a_\ell A^{n-1-\ell} B\right)^+ \tag{4.31}$$

Using (4.31) would result in $(m-1)n$ degrees-of-freedom. In this research (4.30) was used, but the choice should be based upon the relative magnitudes of m and r.

D. PARTITIONED ALGORITHM WITH ANALYTICAL GRADIENTS FOR SINGLE MODE

In the previous section, the algorithm of [54] was partitioned into a linear and a nonlinear part to improve computational efficiency. The implementation of the algorithm with Stewart's adaption of the Davidon-Fletcher-Powell (SDFP) computer code utilizes numerical differencing techniques for the computation of the gradients. This is time consuming and is required only when analytical relations for the gradients are not available. General relations for the gradients were not possible because of the inverse/pseudo inverse aspect of equation (4.30) that is used in the quadratic form of (4.25). However, it is possible to derive analytical gradients for special cases.

In the multivariable examples used in this research, the problem of adding a single mode with two inputs and two outputs (2x2x2) to a model was encountered frequently. In order to speed up the partitioned algorithm further, analytical relations for the first partials and second partials of the cost function with respect to the parameters of H were found. This allowed direct implementation into Newton's method of optimization [22]. The iteration in terms of optimization variables, z_i, would be processed for the kth iteration as

$$z_i(k+1) = z_i(k) + J_{zz}^{-1} J_z \tag{4.32}$$

where J is the cost function defined from (4.25) as

$$J = \sum_{i=0}^{n-1} \sum_{j=1}^{m} \sum_{k=1}^{r} \bar{r}_{ikj}^{2} \tag{4.33}$$

J_z and J_{zz} are the first and second partials of J with respect to the optimization variables, z_i.

For the 2x2x2 case considered here, B is found from (4.30) as

$$B = (a_1 H+HA)^{-1} M_0 \tag{4.34}$$

and the residuals come from (4.25) as only the s term since (4.34) exactly satisfies the i=0 term

$$\bar{r}^2_{ijk} = \frac{[m_{ijk}-a_0 HB]^2}{m^2_{ijk}} \tag{4.35}$$

For convenience H is taken as

$$H = \begin{bmatrix} 1 & 1 \\ h_{21} & h_{22} \end{bmatrix} \tag{4.36}$$

so that the parameters that are sought in the nonlinear part become

$$\underline{p} = \begin{bmatrix} h_{21} \\ h_{22} \end{bmatrix} \tag{4.37}$$

The first and second partials of J with respect to $\underline{p}$ were computed on MACSYMA [59,60]. The 4 second partials and the 2 first partials took up over 50 lines of Fortran code. Although the computational burden is lower, the complexity is much greater.

The optimization variable transformation of [58] was used in this case, as well, to scale the parameters and to apply limits to the range the search is taken over. Hence, the partials of J with respect to z_i are computed as follows

$$\frac{\partial J}{\partial z_i} = \frac{\partial J}{\partial p_i} \cdot \frac{\partial p_i}{\partial z_i} \tag{4.38}$$

$$\frac{\partial^2 J}{\partial z_i \partial z_j} = \frac{\partial^2 J}{\partial p_i \partial p_j} \cdot \frac{\partial p_i}{\partial z_i} \cdot \frac{\partial p_j}{\partial z_j} \tag{4.39}$$

The correction terms in (4.38) and (4.39) for transforming the gradients between the z-domain and the p-domain are developed from (4.26) as

$$\frac{\partial p_i}{\partial z_i} = \frac{(U_i - L_i)}{2} \frac{\pi}{2} \cos\left(\frac{\pi}{2} z_i\right) \tag{4.40}$$

This 2x2x2 version of the multivariable state space realization had significant savings as the number of required function calls was reduced 50 percent over the partitioned method of the previous section.

In this section the analytical gradients, J_p and J_{pp}, were found on MACSYMA [59,60] only for the 2x2x2 problem. MACSYMA was used to directly obtain Fortran code, indicating the possibility of finding the gradients for any specific problem if a particular application warranted it. For example, a 4x2x2 system would be useful for airplane applications. So the general procedures developed in this section would be useful for developing adaptive control systems for other cases. However, the lack of generality and the length of the sensitivity equations limit the widespread applicability of this approach.

E. PARTITIONED LINEAR ALGORITHM

After performing the simulations for this report, another method was found which partitions the problem into two linear

problems.[1] One is solved with a simple matrix inverse and the other is solved with a pseudo inverse (least squares). No nonlinear programming techniques are required. This approach is derived here and results are presented in the next section which illustrate the increased efficienty and accuracy.

The partitioned linear algorithm is derived by rewriting the infinite matrix polynomial equation (4.10) as a finite partial fraction expansion

$$(sI-A)^{-1} = \frac{\sum_{\ell=0}^{n-1} s^{\ell} C_{n-\ell}}{d(s)} , \tag{4.41}$$

where $C_{n-\ell}$ are the numerator matrices. Then by equating the transfer function numerator matrices in powers of s, equation (4.41) can be partitioned into two linear problems.

The numerator terms from (4.6), (4.7) and (4.41) are equated as

$$H \sum_{\ell=0}^{n-\ell} s^{\ell} C_{n-\ell} B = \sum_{\ell=0}^{n-1} s^{\ell} M_{n-\ell} . \tag{4.42}$$

As for the previous approaches, a nonzero element is specified in each column of H. For convenience and compact notation, 1's are chosen for all the elements of the first row. If the notation $(.)^U$ is used to indicate taking the first (uppermost) row of (.) and $(.)^L$ is used to indicate taking the remaining (lower) rows of (.), H can be partitioned as

$$H = \begin{bmatrix} H^U \\ H^L \end{bmatrix} \tag{4.43}$$

[1] This method was proposed to the author by Professor Arthur E. Bryson, Jr., of Stanford University in a personal letter dated April 20, 1983. This was later refined in a letter from Prof. Bryson on May 14, 1983. The author has subsequently generalized the approach for high order systems for any number of inputs and outputs.

where

$$H^U = [1 \quad 1 \quad \ldots \quad 1] \tag{4.44}$$

with dimension 1xn. H^L has dimension (r-1)xn. The desired numerator matrices, M(s), are partitioned in the same fashion

$$M = \begin{bmatrix} M^U \\ M^L \end{bmatrix} \tag{4.45}$$

The dimension of M^U is 1xm and the dimension of M^L is (r-1)xm.

Define a new matrix, T_H, which stacks the first rows of the matrix products $H^U C_k$

$$T_H = \begin{bmatrix} H^U C_1 \\ H^U C_2 \\ \vdots \\ H^U C_3 \end{bmatrix} . \tag{4.46}$$

Also, the first rows of the M(s) matrices are stacked vertically in V_B

$$V_B = \begin{bmatrix} (M_1)^U \\ (M_2)^U \\ \vdots \\ (M_3)^U \end{bmatrix} . \tag{4.47}$$

Equation (4.42) is partitioned using (4.46) and (4.47) into a linear equation in terms of B.

$$T_H B = V_B \tag{4.48}$$

B is found by premultiplying (4.48) by T_H^{-1}

$$B = T_H^{-1} V_H . \tag{4.49}$$

The remaining nr-n values in H are found by solving the remaining equations of (4.42). Define two new matrices as

$$T_B = [C_1 B \quad C_2 B \quad \cdots \quad C_n B] \tag{4.50}$$

$$V_H = [(M_1)^L \quad (M_2)^L \quad \cdots \quad (M_n)^L] . \tag{4.51}$$

The dimensions of T_B are nxnm and the dimensions of V_H are (r-1)xnm. The lower partition of (4.42) is written as

$$H^L T_B = V_H , \tag{4.52}$$

which is a set of nmr-nm equations for nr-n unknowns. The unknown values for H are found in a least squares sense as

$$H^L = V_H (T_B)^+ \tag{4.53}$$

where $()^+$ indicates a pseudo inverse.

The numerical solution is found in a direct way with no requirement for nonlinear programming. Nonzero values for the first row of the output distribution matrix, H, are assumed. The partial fraction expansion for the numerator terms of $(sI-A)^{-1}$, C, is computed using the Leverrier-Souriau-Faddeeva-Frame formula [51,52]. The control influence matrix, B, is computed using (4.49). Then a least squares solution for H_L, the remaining terms of the output distribution matrix are computed using (4.53). This approach requires no iteration and is computationally fast. The most stringent computational burden is the need for two matrix inverses of size nxn.

F. NUMERICAL EXAMPLES

In this section a 2x2x2 transfer function from [54] is considered and the algorithms of the previous four sections are used to find a state space realization. The answers are compared by evaluating the residual at the solution. First a problem with exact coefficients is given and secondly the same problem with inexact coefficients is utilized. The exact transfer function is

$$P(s) = \frac{\begin{bmatrix} s+6 & s+3 \\ 4 & s+3 \end{bmatrix}}{s^2+5s+6} \tag{4.54}$$

The chosen A matrix becomes

$$A = \begin{bmatrix} -5 & 1 \\ -6 & 0 \end{bmatrix} \tag{4.55}$$

and the M matrices are

$$M_1 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \tag{4.56}$$

$$M_2 = \begin{bmatrix} 6 & 3 \\ 4 & 3 \end{bmatrix}. \tag{4.57}$$

The H matrix is assumed to be

$$H = \begin{bmatrix} 1 & 1 \\ h_{21} & h_{22} \end{bmatrix}. \tag{4.58}$$

Initial guesses for the parameters of the partitioned method were $h_{21}=1$ and $h_{22}=.25$.

The exact solution for the state space realization is

$$B = \begin{bmatrix} 0 & .25 \\ 1.0 & .75 \end{bmatrix} \tag{4.59}$$

$$H = \begin{bmatrix} 1.0 & 1.0 \\ 4.0 & 0 \end{bmatrix} \tag{4.60}$$

A comparison of the straight SDFP, the partitioned SDFP, the Newton's iterative, and the partitioned linear methods are shown in table (4.1). The partitioned SDFP algorithm reduced the number of function calls from 25 to 7 and improved the accuracy of the fit. The partitioned Newton's method with analytical gradients reduced the number of function calls further to 3. The partitioned linear method required no iterations and was much more accurate than the other three.

Table 4.1 — Convergence characteristics of realization algorithms for a problem with exact coefficients and of size 2 X 2 X 2.

METHOD	NO. OF FUNCTION CALLS	RESIDUAL
SDFP	25	4.01 E -07
PARTITIONED SDFP	7	1.21 E -14
PARTITIONED NEWTON WITH ANALYTICAL GRADIENTS	3	1.70 E -11
PARTITIONED LINEAR LEAST SQUARES	1	2.31 E -18

The solution that the partitioned linear method achieved was

$$B = \begin{bmatrix} -2.39\text{x}10^{-10} & .25 \\ 1.0 & .75 \end{bmatrix} \tag{4.61}$$

$$H = \begin{bmatrix} 1.0 & 1.0 \\ 4.0 & 9.57\text{x}10^{-10} \end{bmatrix} \tag{4.62}$$

Now consider the case where the transfer function is an inexact representation of the same system. This might occur with experimental data, for example. The inexact transfer function is given by

$$P(s) = \frac{\begin{bmatrix} 1.11s+6.19 & 1.02s+3.25 \\ 4.11 & .9s+3.12 \end{bmatrix}}{s^2+5.03s+6.62} \tag{4.63}$$

The matrices A, M_1, M_2 and H are defined in the same fashion as equations (4.55-4.58). The results of applying the three algorithms is shown in table 4.2. Partitioning the SDFP algorithm resulted in a reduction of function calls from 39 to 10. Computing analytical partials of the cost function allowed a further reduction in function calls to 5. The partitioned linear method gave acceptable performance with no iterations.

The solution that the partitioned linear method obtained for the inexact case is given by

$$B = \begin{bmatrix} .04 & .23 \\ 1.07 & .79 \end{bmatrix} \tag{4.64}$$

$$H = \begin{bmatrix} 1.0 & 1.0 \\ 4.32 & -.107 \end{bmatrix} \tag{4.65}$$

Table 4.2 — Convergence characteristics of realization algorithms for a problem with inexact coefficients and of size 2 X 2 X 2.

METHOD	NO. OF FUNCTION CALLS	RESIDUAL
SDFP	39	3.40 E -07
PARTITIONED SDFP	10	4.44 E -06
PARTITIONED NEWTON WITH ANALYTICAL GRADIENTS	5	7.36 E -07
PARTITIONED LINEAR LEAST SQUARES	1	3.51 E -4

Using (4.64) and (4.65), P(s) is

$$P(s) = \frac{\begin{bmatrix} 1.11s+6.19 & 1.02s+3.25 \\ .057+4.086 & .907s+3.156 \end{bmatrix}}{s^2+5.03s+6.62} \qquad (4.66)$$

Most approaches to state space realizations of this second example would have resulted in a fourth order system due to the inexactness of the experimentally-determined coefficients. Since the LMS or IMLMS algorithm is implemented by stating the desired size of the block to be added, it is advantageous for the algorithm to return a minimal state space realization in a least squares, best fit sense. If the value of the residual at the solution is too large, further estimation of the parameters by the output error identification scheme would be warranted.

Since real-time applications for the algorithm using microprocessors are anticipated, the linear partitioned method appears to be the most promising. It requires no iteration and is totally general for any size system with any number of inputs and outputs. Additionally, it is computationlly simple as it requires only two matrix inverses of matrices with rank equal to the system order. It sacrifices some accuracy for inexact problems, but these differences were negligible for the cases considered.

Chapter V

SUBOPTIMAL COMPENSATOR DESIGN AND IMPLEMENTATION

A. INTRODUCTION

Section I-C describes the approach to adaptive control in this research. A fast, robust identification scheme estimates a set of discrete transfer functions for the process being controlled (chapters II and III). These transfer functions are then converted into a state space realization (chapter IV). Using the compact matrix notation of the state space realization, digital compensator logic is designed using linear-quadratic-gaussian (LQG) techniques (this chapter).

Figure (V-1) is a block diagram that shows the form of the suboptimal compensator. Albeit LQG or so-called optimal control techniques are used, the compensator is referred to as suboptimal because generally no prior knowledge of noise covariances is available and because the model utilized for the compensator design is understood to be an inexact representation of the plant. Constant gain matrices K, C, and F are the output of the design process. To keep the computational burden low for the digital compensator, both the filter and the controller are implemented with their steady state values. The LQG digital designs were performed using ORACLS [51]. A more complete description of the computing techniques is given in appendix I.

Some simplifications were required to obtain the capability to perform on-line designs. The form of the weighting matrices was assumed; and, it was assumed that the separation theorem [52] holds in spite of plant-model mismatch. This allows the filter and the controller to be designed sequentially. However, the overall system is checked for stability of the compensator.

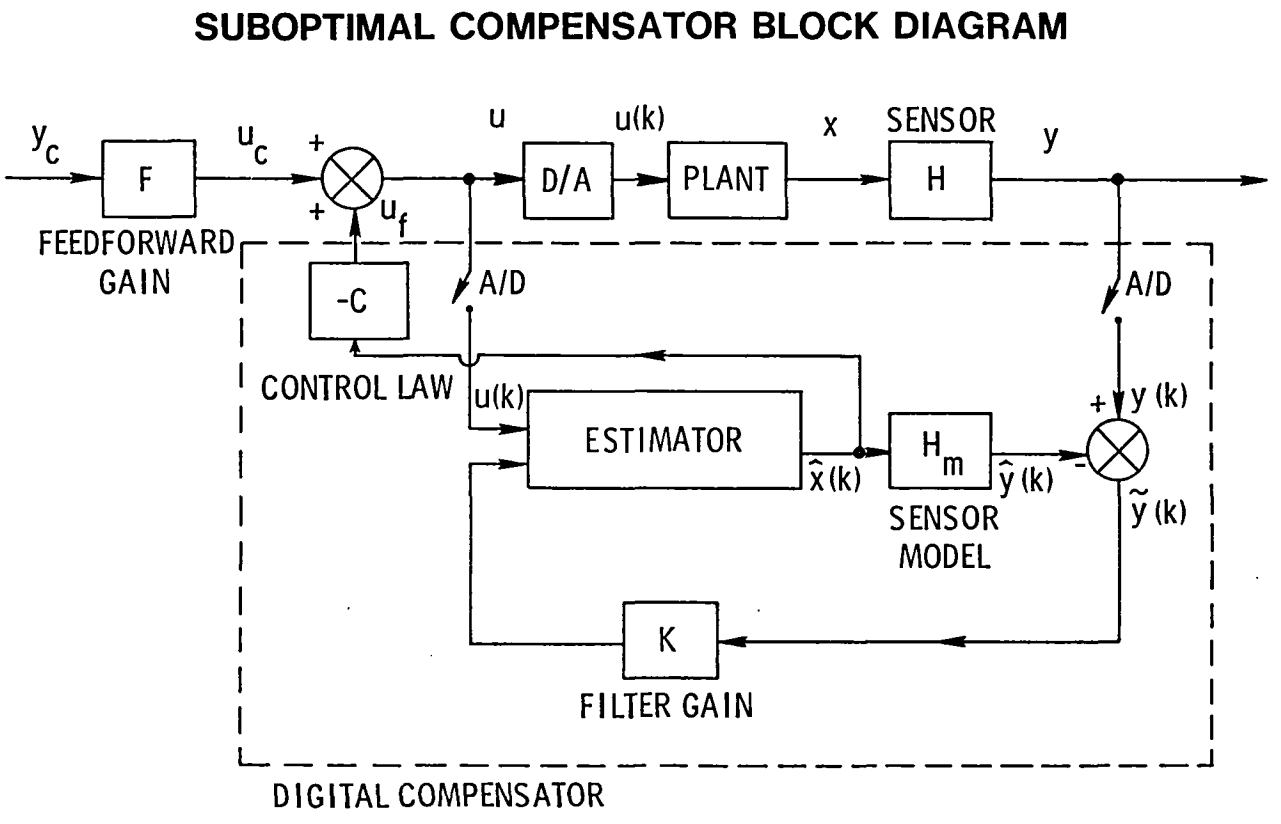

Figure V-1 — Block diagram showing fundamental structure of suboptimal discrete compensator.

B. STEADY STATE KALMAN FILTER DESIGN

Once the order and parameters of the model have been found, an asymptotic Kalman-Bucy filter [63] is found using ORACLS [51]. The system is modeled as a discrete, LQG, time-invariant process. The state equation is given by

$$\underline{x}(k+1) = \Phi\underline{x}(k) + \Gamma_w\underline{w}(k) \tag{5.1}$$

with output

$$\underline{y}(k+1) = H\underline{x}(k) + \underline{v}(k) \tag{5.2}$$

where Φ is the state transition matrix, Γ_w is the discrete influence matrix of process noise input, $\underline{w}$. It is assumed for simplicity in most filter designs that Γ_w, is an identity matrix of order n. The elements of Φ and H are determined by the state space realization methods of the previous chapter from the identified parameters of the LMS or IMLMS algorithms.

The process noise $\underline{w}(k)$ and the measurement noise $\underline{v}(k)$ are modeled as purely random gaussian vectors with zero means and covariance matricies Q and R. The solution requires that w(k) and v(k) are mutually uncorrelated. Since prior knowledge of the process is limited due to the model building approach being proposed, the on-line design is implemented by assumming that Q and R are identity matrices of rank n and r, respectively. The designer choices ρ, a scaler multiplicative factor of Q, as a means of varying the relative weights in the cost function. This simplification is possible since the output variables are scaled during the identification process.

The cost function is defined as

$$J = \lim_{k_0 \to -\infty} E[\underline{\varepsilon}^T(k)W\underline{\varepsilon}(k)] \tag{5.3}$$

where $\underline{\varepsilon}$ is defined as

$$\underline{\varepsilon}(k) = \underline{x}(k) - \hat{\underline{x}}(k) \tag{5.4}$$

and W is a weighting matrix which does not appear in the computation. If the pair (Φ,Γ_w) is stabilizable and the pair (Φ,H) is detectable, then a solution to the optimal observer problem exists and the predicted state, $\hat{\underline{x}}$, is given by

$$\hat{\underline{x}}(k+1) = \Phi\hat{\underline{x}}(k) + K[\underline{y}(k)-H\hat{\underline{x}}(k)] \ , \tag{5.5}$$

where

$$\hat{\underline{x}}(0) = E[\underline{x}(0)] \ . \tag{5.6}$$

The steady state filter gain is found from

$$K = \Phi PH^T(R+HPH^T)^{-1} \ , \tag{4.7}$$

where P satisfies the discrete algebraic Riccatti equation

$$P = (\Phi-KH)P(\Phi-KH)^T + KRK^T + \rho Q \ . \tag{4.8}$$

The matrix P represents the constant, steady-state variance matrix of the reconstruction error, $\varepsilon(i)$, given by

$$\lim_{i=0\to-\infty} E\left[\tilde{x}(k)\tilde{x}^T(k)\right] = P \tag{5.9}$$

The product of this computation is K, the Kalman filter gain matrix in figure (V-1). The designer chooses the weighting factor, ρ, and the on-line code uses the values of Φ and H to find a K matrix. Some skill is needed in choosing ρ and it could

conceivably be delegated to the heuristic, problem solving logic of the adaptive processor. Clearly, if this approach was being implemented on a process where prior knowledge can be used to estimate Q and R, an optimal Kalman-Bucy filter may be possible.

C. STEADY STATE REGULATOR DESIGN

After the order and parameters of the model have been determined, a state space representation found, and an asymptotic filter gain computed, an asymptotic quadratic regulator gain, C, is calculated (see fig. (V-1)). The discrete, time-invariant, linear, optimal output regulator problem is solved using ORACLS [51]. The system is given by

$$\underline{x}(k+1) = \Phi\underline{x}(k) + \Gamma\underline{u}(k) \ , \tag{5.10}$$

and the output vector is

$$\underline{y}(k) = H\underline{x}(k) \ . \tag{5.11}$$

The cost function that is optimized is

$$J = \lim_{N\to\infty} E \sum_{i=0}^{N-1} [\underline{y}^T(k+1)\rho Q_y \underline{y}(k+1) + \underline{u}^T(k) Q_u \underline{u}(k)] \tag{5.12}$$

subject to the constraints of (5.10) and (5.11). The matrices Φ, Γ and H are obtained from the estimation and state space realization sections of the code. Q_y and Q_u are weighting matrices that are assumed to be identity matrices of order r and m, respectively. The design parameter, ρ, is a scalar multiplicative factor of Q_y.

If the pair (Φ,Γ) is stabilizable and the pair (Φ,H) is detectable, a solution to the optimal control exists and is given by

$$\underline{u}(k) = -C\underline{x}(k) \ , \tag{5.12}$$

where

$$C = (R+\Gamma^T P \Gamma)^{-1} \Gamma^T P \Phi \tag{5.13}$$

and

$$P = (\Phi-\Gamma C)^T P(\Phi-\Gamma C) + C^T Q_u C^T + H^T \rho Q_y H \tag{5.14}$$

The steady-state controller is designed by computing the constant regulator gain C. The code uses as input the matrices Φ, Γ and H and the designer chooses ρ. The weighting factor, ρ, is chosen by the designer as a relative trade in the cost function between output power and allowable control input power that is minimized. The convenience of utilizing ρ as the design parameter is possible since the variables are scaled during the identification process. It may be desirable to allow the adaptive processor the opportunity to heuristically select the magnitude of ρ.

The separation theorem was used to design the controller and observer separately, but it should be noted that when actually implemented they are linked by

$$\underline{u}(k) = -C\hat{\underline{x}}(k) \tag{5.14}$$

That is, the estimated states are fedback. Stability is not guaranteed since the transition matrices of the estimator and regulator may not be sufficiently accurate representations of the process being controlled. This assumption that the separation theorem holds is not necessarily true, but the suboptimal compensators designed in this fashion do exhibit good control characteristics for a wide range of problems.

D. NON-ZERO SET POINT FOR DISCRETE REGULATOR

During the model building process proposed here, it is necessary to sufficiently excite all the modes of the system. It was observed in chapter III that square wave pulses of the control inputs are a good way to do this. Although it may not be desirable during a control task to indiscriminately pulse the controls, some sort of dither signal is necessary to insure the convergence of the identification schemes.

An alternative approach, which has proven to be successful, is to command the output to follow a pulsed square wave pattern. If the plant model is good, the output will closely follow the commanded output. However, if there are unmodeled modes present, they will most likely be excited and discerned in the output. The reason for this is that feedforward gain matrix, F, is based on the plant model.

The feedforward gain matrix, F, for a discrete regulator is based upon the non-zero set point calculations from reference [64]. Since the calculation is fundamentally requiring an asymptotic response to step commands, the zeros have been neglected [21]. The state equations are

$$\underline{x}(k+1) = \Phi\underline{x}(k) + \Gamma\underline{u}(k) \tag{5.15}$$

$$\underline{y}(k) = H\underline{x}(k) + J\underline{u}(k) \tag{5.16}$$

The steady state equations occur when

$$\underline{x}_{ss} \simeq \Phi\underline{x}_{ss} + \Gamma\underline{u}_{ss} \ , \tag{5.17}$$

$$\underline{y}_{ss} = H\underline{x}_{ss} + J\underline{u}_{ss} \ . \tag{5.18}$$

Combining terms in (5.17) yields

$$\underline{0} \simeq (\Phi - I)\underline{x}_{ss} + \Gamma\underline{u}_{ss} \tag{5.19}$$

writing (5.19) and (5.18) in matrix form

$$\begin{bmatrix} \underline{0} \\ \underline{y}_{ss} \end{bmatrix} = \begin{bmatrix} \Phi-I & \Gamma \\ H & L \end{bmatrix} \begin{bmatrix} \underline{x}_{ss} \\ \underline{u}_{ss} \end{bmatrix} \tag{5.20}$$

Solving for $\underline{x}_{ss}$ and $\underline{u}_{ss}$ yields

$$\begin{bmatrix} \underline{x}_{ss} \\ \underline{u}_{ss} \end{bmatrix} = \begin{bmatrix} \Phi-I & \Gamma \\ H & L \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ y_{ss} \end{bmatrix} \tag{5.21}$$

The matrix inverse is partitioned as follows:

$$\begin{bmatrix} \Phi-I & \Gamma \\ H & L \end{bmatrix}^{-1} = \begin{bmatrix} S_1 & F_1 \\ S_2 & F_2 \end{bmatrix} \tag{5.22}$$

where the dimensions of S_1 are nxn, S_2 are mxn, F_1 are nxr and F_2 are mxr. The steady state values of the states and control inputs are

$$\underline{x}_{ss} = F_1 \underline{y}_{ss} \tag{5.23}$$

$$\underline{u}_{ss} = F_2 \underline{y}_{ss} \tag{5.24}$$

Since it is desired to have the steady state output approach the steady state commanded output, let

$$\underline{y}_{ss} = \underline{y}_c \tag{5.25}$$

Also, in the steady state $\hat{\underline{x}}_{ss} \rightarrow \underline{x}_{ss}$, so from (4.14)

$$\underline{u} \simeq -C\underline{x} \ . \tag{5.26}$$

Adding a steady state component to both sides of (5.26) implies that corrective inputs would occur only for deviations from the non-zero set point

$$\underline{u} - \underline{u}_{ss} \simeq -C(\underline{x}-\underline{x}_{ss}) \tag{5.27}$$

Solving for the total control $\underline{u}$

$$\underline{u} = \underline{u}_{ss} - C\underline{x} + C\underline{x}_{ss} \tag{5.28}$$

Substituting (5.23) and (5.24) into (5.28) gives

$$\underline{u} = F\,\underline{y}_c - C\underline{x} \tag{5.29}$$

where

$$F = (CF_1+F_2) \tag{5.30}$$

The constant feedforward gain matrix, F, is implemented in the suboptimal compensator of figure (V-1). The commanded output, y_{com}, is used to pulse the output so as to excite the unmodeled dynamics. The amplitude and frequency of the square wave pulse can be selected by the designer. Alternatively, these design parameters could be systematically varied and optimized by the heuristic learning logic of the adaptive processor.

E. COMPENSATOR IMPLEMENTATION STRATEGIES

The overall adaptive control scheme described in section I-C requires that the compensator design be updated periodically. The output error identification scheme is in parallel to the plant and the compensator (see fig. (V-2)). Information from the identification block passes to the compensator block only when the compensator is to be redesigned, which may include a change in model order as

OVERALL SCHEME FOR ON-LINE EQUIVALENT SYSTEM IDENTIFICATION FOR ADAPTIVE CONTROL

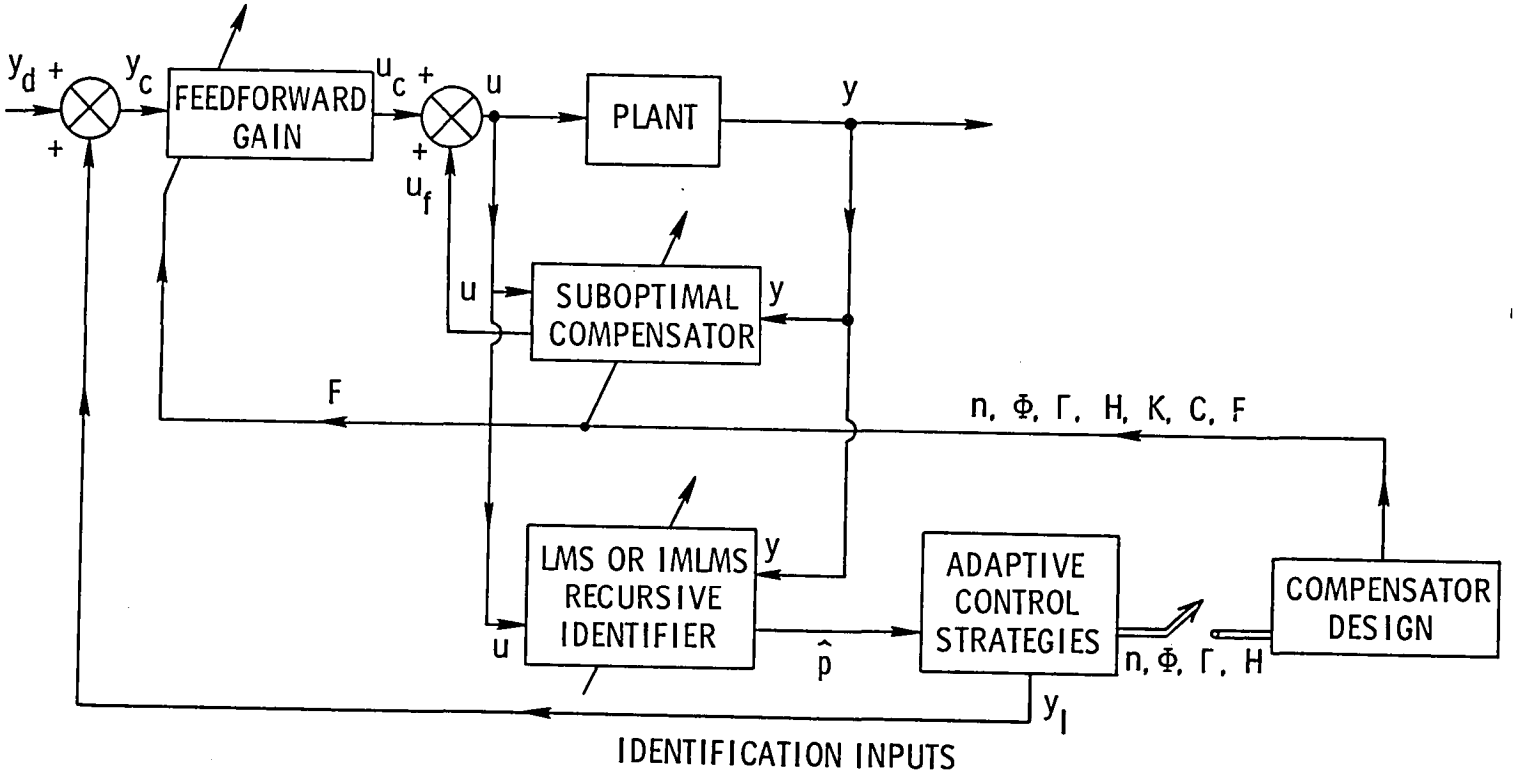

Figure V-2 — Block diagram showing on-line equivalent system identification scheme for adaptive control.

well as model parameters. Information is passed when divergence of the parameter estimation scheme is detected, so divergence problems like the ones described in subsection I-B.2 do not occur for this approach. However, advantage is gained at the expense of adaption speed; it takes time to collect data, build a model, check for consistency and compatibility, and then to design a compensator.

The questions addressed in this section are related to the issue of how often to update the compensator design. The strategies for choosing and evaluating figures-of-merit are also discussed. The fundamental issue is how long should the recursive LMS or IMLMS algorithms be run before a satisfactory convergence is obtained; and, once obtained, when is it warranted to replace the compensator? Therefore, the scheme must have a means of self-evaluation so overall system performance can be enhanced.

E-1 Restructurable Control

The first case considered is when the adaptive control scheme is used in a conventional way. That is, the order of the model is constant, but parameters have a significant uncertainty, time dependence or possible failure modes that might require the compensator to be redesigned (restructured) to augment the control system performance.

Three primary figures-of-merit are used for evaluating subsystem performance. The parameter convergence of the LMS (or IMLMS) algorithm is initially determined by considering an RMS error of prediction from the recursive identification scheme,

$$\sigma_{LMS} = \left[\frac{\sum_{k=1}^{N} \left(\underline{y}(k) - \underline{y}_{LMS}(k) \right)^2}{N} \right]^{1/2} \tag{5.31}$$

where $\underline{y}_{LMS}(k)$ is the predicted output from the LMS or IMLMS algorithms, N is the number of samples.

If σ_{LMS} is above some minimum threshold, say 10^{-1} in normalized units, it is assumed that convergence has not been achieved. If σ_{LMS} is computed over a sufficiently long time and

is above some maximum threshold value, it is assumed that the LMS algorithm is diverging and should be reinitialized.

The estimation subdivision of the compensator is evaluated using an expression similar to (5.31)

$$\sigma_I = \left[\frac{\sum_{k=1}^{N} \left(\underline{y}(k)-\hat{\underline{y}}(k)\right)^2}{N} \right]^{1/2} \tag{5.32}$$

where $\hat{\underline{y}}(k) = H\hat{x}(k)$ is the predicted output of the Kalman filter portion of the compensator.

The RMS estimation error, σ_I, is a measure of how well the model utilized in the compensator design is doing in predicting the response of the plant. If σ_I, after a certain minimum period of time, is above some preselected value, a new model with the parameters obtained from the LMS or IMLMS algorithms is implemented. σ_I includes modeling errors, process noise and measurement noise. The threshold value for σ_I that is selected needs to be considered by the designer in the context of these other disturbances.

Another performance index which is critical for the restructurable control problem is the RMS error of commanded output. It is computed by comparing the actual output with the commanded (desired) output. Typically, a set of square wave pulses is commanded. This figure-of-merit is computed by

$$\sigma_c = \left[\frac{\sum_{k=1}^{N} \left(\underline{y}_{com}(k)-\underline{y}(k)\right)^2}{N} \right]^{1/2} \tag{5.34}$$

If σ_c becomes too large, either the model should be updated or the regulator portion of the compensator should be redesigned with a higher gain factor, ρ. The RMS error of commanded output, σ_c, includes the effort of the process noise and measurement noise, as well as the performance of the compensator.

Determining if a failure has occurred is a complicated and unresolved problem [65,66] that will not be considered here. However, if a failure has been detected, the adaptive control logic proposed here for a restructurable control problem is applicable as shown in the flow chart of figure (V-3). This chart shows an approach that was successful for the examples considered; it consists of heuristic rules selected by the designer to implement an adaptive controller. The delay in time at the second decision level and the other checks prevent the recursively identified parameters from being fed directly through to the compensator, avoiding some of the divergences mentioned in chapter I.

The logic of figure (V-3) is meant to systematically isolate what needs to be improved and is meant to operate simultaneously with the control, estimation and identification processes. If a failure is detected, a predetermined amount of time is allowed to pass so the identification schemes can have time to converge. If the control index is not satisfied, the model prediction is checked. If the model prediction is satisfactory, the controller is redesigned; otherwise, the convergence of the LMS algorithm is checked. Once σ_{LMS} is low enough, the model parameters are updated, a state space realization is found and a compensator is designed using the techniques of this chapter. Once a compensator has been brought on-line, it may be necessary to further update the model or redesign the compensator as the LMS algorithm convergence improves.

The prime limitation of this approach is that an unstable system could fail or break before the adaptive process could work. In general, the scheme has inherent delays to prevent adaptation divergence, but these may result in unsatisfactory performance. The next chapter shows an example of where the approach in this subsection was used successfully to restructure a control system after a component failure.

E-2 Model Building

A prime contribution of this research is the IMLMS algorithm which includes the unstructured model uncertainty in the adaptation

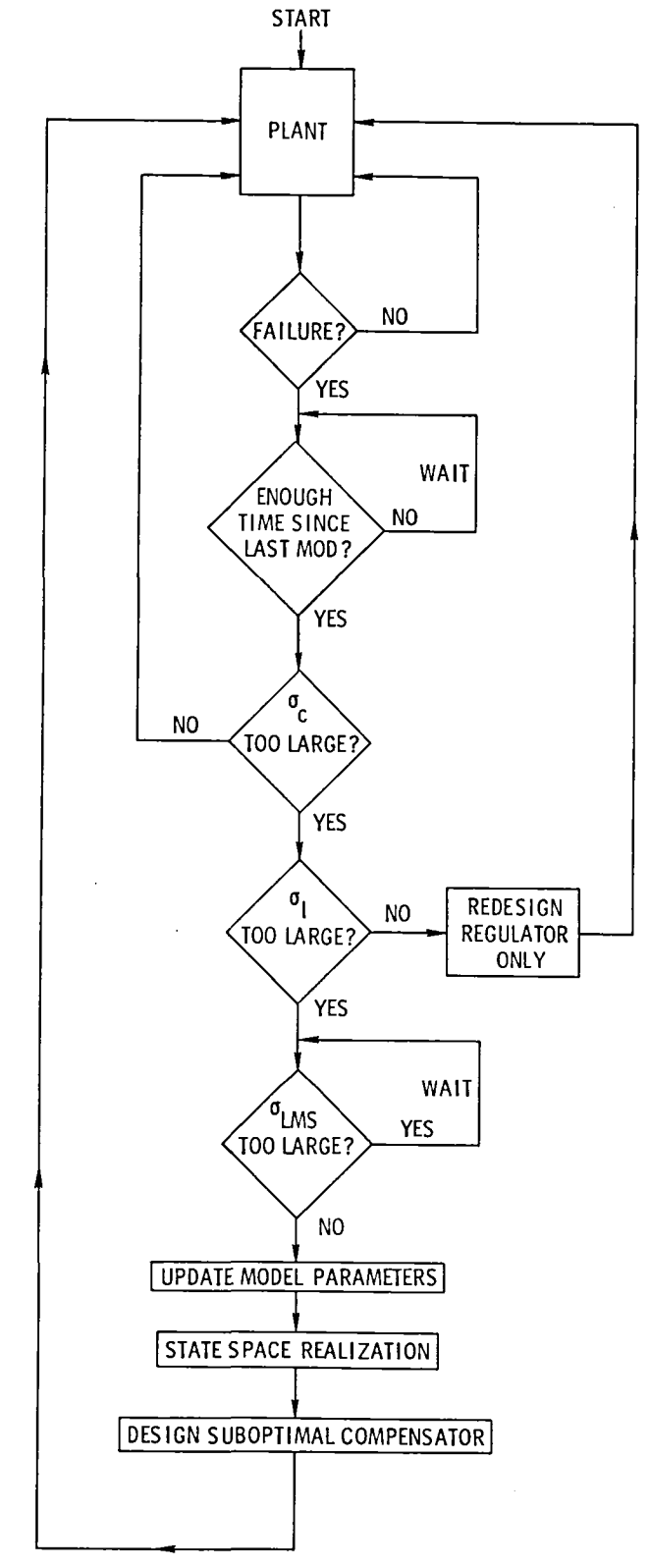

Figure V-3 — Flow chart showing decisions required for restructurable control applications of adaptive control algorithm.

process. Part of the model is assumed to be known and the other part is adapted by IMLMS algorithm. When satisfactory convergence of the IMLMS filter has been achieved, the order of the compensator model is increased as an additional mode is added. This process can be repeated over and over, thereby building a model. The key question is when to add a mode and when to stop.

The model building logic is shown in figure (V-4); it uses the same figures-of-merit as the restructurable control application, σ_{LMS}, σ_I, and σ_L. However, in addition, the designer selects a minimum model order to begin control and maximum model order for the model building process. These parameters are chosen based on prior knowledge of the plant and its environment. Stability problems may occur if a plant is controlled with too few modes represented in the compensator design model. The maximum limit is a practical one which may be dictated by hardware constraints.

If the maximum model order limit has not been exceeded, the adaptive model building logic tests to see if enough time has elapsed since the last control system modification, this allows the IMLMS algorithm enough time to converge. The designer chooses this based upon prior knowledge of system dynamics and the chosen sample rate. If σ_c is satisfactory, no adaptation is required. Otherwise, σ_I is tested.

If σ_I is significantly bigger than the last time it was computed, experience has shown that this is because the model order was too large the last time a mode was added. Most likely the last mode was added based upon trying to identify nearly white noise, and when implemented in the Kalman-Bucy filter, degrades its ability to predict the output. A successful solution to this predicament is to reduce the internal structure order by removing the last mode.

The controller portion of the compensator is redesigned if σ_c is too large and σ_I is satisfactorily small. If σ_I is too large, another mode is desired for the compensator model. It is added once the IMLMS algorithm satisfies its convergence check. The controller portion of the model is implemented only if the number of modes in the compensator model is above the minimum value required by the designer.

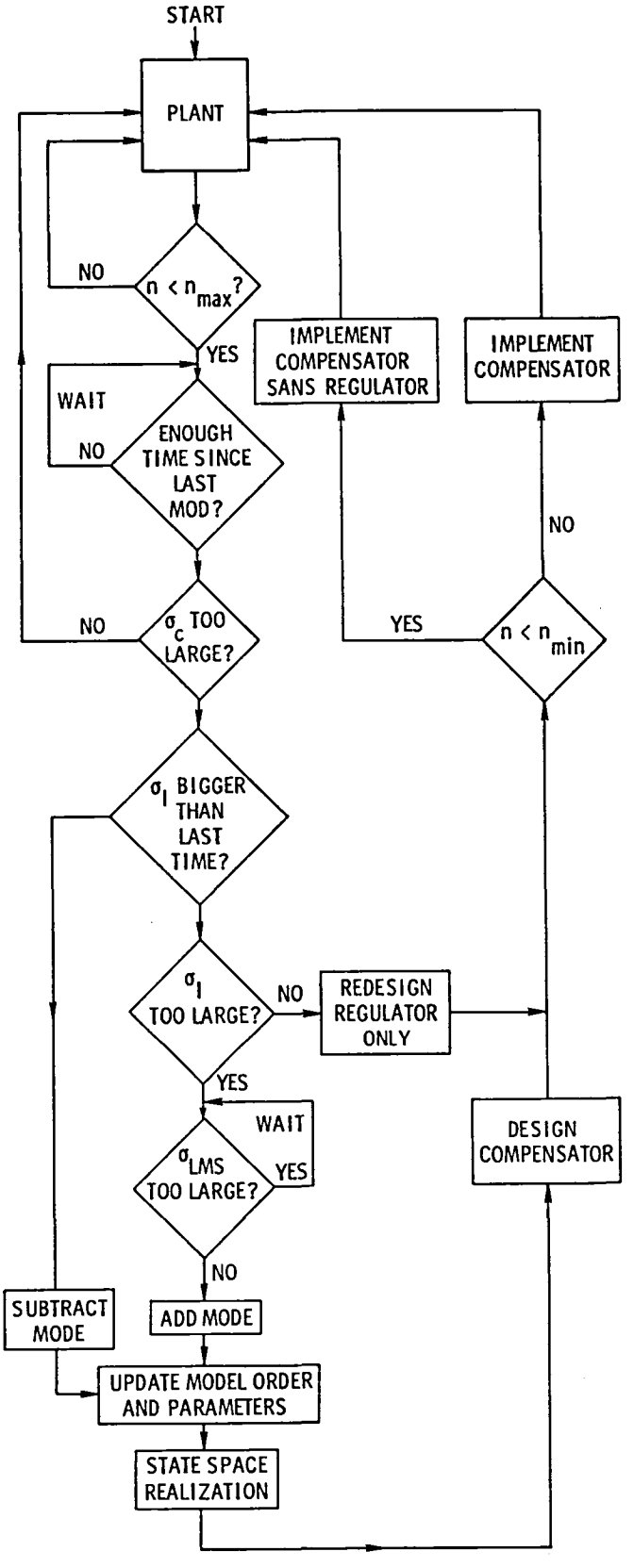

Figure V-4 — Flow chart showing decisions required for model building during adaptive control of plants with unstructured model uncertainty.

The logic shown in figure (V-4) is the logic that was implemented for the model building example in the next chapter. It could be expanded to apply to wider classes of problems. The time needed to perform the model building precludes its use for unstable plants or where the speed of adaptation is critical. In the next subsection, the ability to avoid problems resulting from on-line adaptation is discussed.

Although not studied here, it may be possible to use the LMS algorithm in parallel with the IMLMS algorithm; the LMS filter could be used to refine the estimated parameters of the modes that are being added by the IMLMS filter. Depending upon the application, it may be possible to restart the identification process after one complete model has been built using the refined estimates of the previous model as initial estimates. This is possible because of the parallel structure of the plant, compensator, and IMLMS algorithm.

E-3 Discussion of Additional Adaptive Problem Solving Logic

It is desirable, especially for the case of model building, to have an autonomous adaptive system that is capable of handling most adverse contingencies that may be encountered. A prime example is that of a space structure under construction. An adaptive controller would be required to satisfy certain pointing or damping requirements. Since it is envisioned that the entire control mission would be satisfied by on-board computers, it will be necessary to expand the capabilities mentioned in the previous two subsections to include recovery from predictable (or even unpredictable) errors or divergences that may occur.

In this subsection some potentially important factors in the adaptive design are considered. Although none of the problems described in this subsection were encountered during the simulations of the next chapter, some were encountered during other applications of this equivalent system identification scheme for adaptive control. Other points are natural extensions that could conceivably be added. When a control system is fine tuned for use, it requires many iterations by a knowledgeable controls expert. Hence, it is envisioned that an "expert system" with knowledge to apply heuristic rules would be required to solve the problem.

This becomes especially feasible if multiple processors are used so that controller speed is not impacted while decisions are made. Looking at figures (I-8) and (I-9), a logical separation of functions would be to put the controller and recursive identification in one processor and the state space realization, compensator design, and adaptive logic rules in the other processor.

An obvious protective mechanism would be the capability to turn off the compensator. Sometimes the overall scheme breaks down, either due to an inherent limitation or implementation error. This probably can initially be detected by a control input saturation or too large of an amplitude of output, the first indication of a divergence. Since this scheme, at this point, is intended for stable plants, the wisest thing would most likely be to stop controlling and restart the adaptive identification logic. A reasonable expection would be that the identification has failed (inaccurate) or the compensator model order is too small.

A problem that immediately comes out of the analysis in section I-D is the scaling of the state variables and the control inputs with respect to each other and the step size factor, μ. Otherwise, parameter divergence is likely. The codes for LMS and IMLMS algorithms of appendix II include normalization factors for the measurements and inputs. It may be necessary for the computer to adjust these factors during on-line, autonomous operations. This can normally be detected by σ_{LMS} growing too large or by the parameters of the LMS or IMLMS filter becoming unrealistically large.

As it turns out, σ_{LMS} is not a very accurate measure of convergence. As indicated in section II-D, the error of prediction of the LMS or IMLMS algorithm is easily satisfied even prior to parameter convergence. So actually σ_{LMS} is an indication of divergence of the LMS or IMLMS algorithm. A better measure of convergence of the algorithm would be to check and see the rate-of-change of the parameters over a sufficiently long time. This is especially true for the control influence terms of the multivariable LMS or IMLMS algorithms, if a batch least squares method is not used for the numerator terms.

Another check for convergence of the LMS or IMLMS algorithms would be to evaluate the pole and zero locations of the discrete transfer function prior to realizing in state space form. It should be possible to isolate certain numerical difficulties prior to implementing a compensator. For example, sometimes a pole on the negative real axis in the z-domain may be identified during parameter convergence problems. This corresponds to an imaginary frequency and usually results in an inaccurate model for the compensator design. When the numerator terms are not converging, experience has shown that the discrete transfer function zeros frequently lie significantly outside the unit circle in the z-domain, corresponding to zeros deep in the right half plane of the s-plane.

In the case of large space structures, it is expected that a significant period of time will be required to perform the model identification needed to construct a model for the control system design. In addition to using the control system actuators to excite the structure and making measurements from the control system sensors, it may be beneficial to utilize special purpose actuators and sensors to aid in the model identification process during this period of time. Application of disturbances other than through the actuators may be necessary to insure that the system has had all its modes excited. Auxillary sensors could be used to distinguish between uncontrollable modes and sensor noise. The use of on-board heuristic logic would make the application of such practices feasible.

Another task that could be turned over to an intelligent computer system that is required to function autonomously, is the selection of the designer's step size factor, μ and the actual control (dither) inputs. These are the prime variables available to the controls engineer in this scheme for influencing overall system performance. Ideally, it would be advantageous to develop a learning system that could vary μ, u_{amp} and Ω to enhance performance. Most likely a trial and error approach with systematic

variations would be required. So again, heuristic logic would need to be programmed based upon the designer's experience.

Other factors could be considered. The fundamental issue is that a truly adaptive system will need some implementation of heuristic rules to help prevent instabilities and reduce the risk of divergence. However, the proper balance between the use of artificial intelligence and control theory for solving real time control problems will need considerable research.

Initially, fairly simple decisions need to be evaluated under the formalism of artificial intelligence as a means of verifying the approach. When the required decision process becomes highly complex, the approach can then be extended with confidence to enable an efficient search for the proper solution.

In this research an attempt was made to include some heuristic rules to aid in the solving of off-nominal problems. What was actually implemented was described in the previous two subsections. In the next chapter some examples are presented where the overall scheme for adaptive control are applied to some generic problems.

Chapter VI

ADAPTIVE CONTROL EXAMPLES

A. RESTRUCTURABLE CONTROL EXAMPLE

In this section a generic example of an unstable plant will be considered. The actuator will be assumed to break and an implementation of the overall scheme proposed in this research will be used to restructure the control to achieve acceptable performance as described in section V-E.1. Notice that the terms that will be tracked by the adaptive filter are the numerator terms. It was shown in sections II-D and III-A that the adaptation of the control influence terms are the slowest and most inaccurate of this approach. As a means of improving the adaptive controller performance, the batch least squares scheme of section II-E will also be used in conjunction with the recursive LMS filter.

The linear, time-invariant plant which is representative of a highly unstable attitude hold mode for a helicopter in hover is given by

$$\dot{\underline{x}} = A\underline{x} + Bu \tag{6.1}$$

with output

$$y = Hx \tag{6.2}$$

where the system matrices are given in modal coordinates as

$$A = \begin{bmatrix} 2 & -2 \\ 1 & 0 \end{bmatrix} \tag{6.3}$$

$$B = \begin{bmatrix} 2 & 0 \end{bmatrix}^T \tag{6.4}$$

$$H = \begin{bmatrix} 0 & 1 \end{bmatrix} \tag{6.5}$$

The open loop eigenvalues are $s=1\pm j$.
The plant is clearly unstable and stabilization is required to maintain accurate regulation. A digital compensator is used, as described in chapter V. The estimator is

$$\underline{\hat{x}}(k+1) = \Phi\underline{x}(k) + \Gamma u(k) - K[y(k)-H\underline{x}(k)] \ , \qquad (6.6)$$

and the control law is

$$u(k) = -C\underline{\hat{x}}(k) + Fy_{com}(k) \ , \qquad (6.7)$$

where, for a sample rate of 20 Hz,

$$\Phi = \begin{bmatrix} 1.10 & -.105 \\ .053 & .997 \end{bmatrix} \ , \qquad (6.8)$$

$$\Gamma = \begin{bmatrix} .105 & .00258 \end{bmatrix}^T . \qquad (6.9)$$

The open loop poles in the z-domain are $z=1.05\pm.05j$ (unstable). The constant compensator gain matrices, that were selected using the methods of chapter V, are

$$K = \begin{bmatrix} 4.22 & 1.20 \end{bmatrix}^T \qquad (6.10)$$

$$C = \begin{bmatrix} 5.26 & 15.4 \end{bmatrix} \qquad (6.11)$$

$$F = \begin{bmatrix} 16.4 \end{bmatrix} \qquad (6.12)$$

The closed loop plant performance has eigenvalues $z=.754\pm.16j$. The apparent closed loop natural frequency of the plant is 6.68 rad/sec with a damping ratio of .48.

A simulation was performed using the approaches outlined in appendix I where the output is commanded by

$$Y_{com} = \begin{Bmatrix} Y_{amp} \\ 0 \\ -Y_{amp} \end{Bmatrix} \text{ for } \begin{cases} \sin(\Omega t) > .767 \\ .707 > \sin(\Omega t) > -.707 \\ \sin(\Omega t) \leq -.707 \end{cases} \qquad (6.13)$$

with $Y_{amp}=.25$ and $\Omega=.25$ rad/sec.

At time t=25 seconds, half of a double "actuator" is assumed to break, resulting in only half the original control authority. That is, B changes to

$$B = \begin{bmatrix} 1 & 0 \end{bmatrix}^T \qquad (6.14)$$

The failure is sensed and the adaptive algorithms are given 50 seconds to find a new representation of the plant using the recursive LMS algorithm. At time t=75 seconds a new controller is brought on-line in an effort to improve the control system performance.

Time histories of the output and the commanded output are shown in figure (VI-1). The closed loop damping ratio goes from .78 to approximately .05 at 25 seconds. Stability is barely maintained and there is a large error in the commanded output. Although the system has not diverged, its performance is poor.

After a preselected interval of 50 seconds has expired, the new compensator is brought on-line using the coefficients identified by the LMS algorithm. The system is stabilized with an apparent damping ratio of approximately .45. However, there is still an error--a bias in the nonzero set point of the output command due to poor knowledge of B, a necessity in the computation of the feedforward gain matrix, F.

If the LMS algorithm is given more time to find the numerator terms, it does a better job. In fact, a good strategy would be to update the compensator again after a certain interval in time.

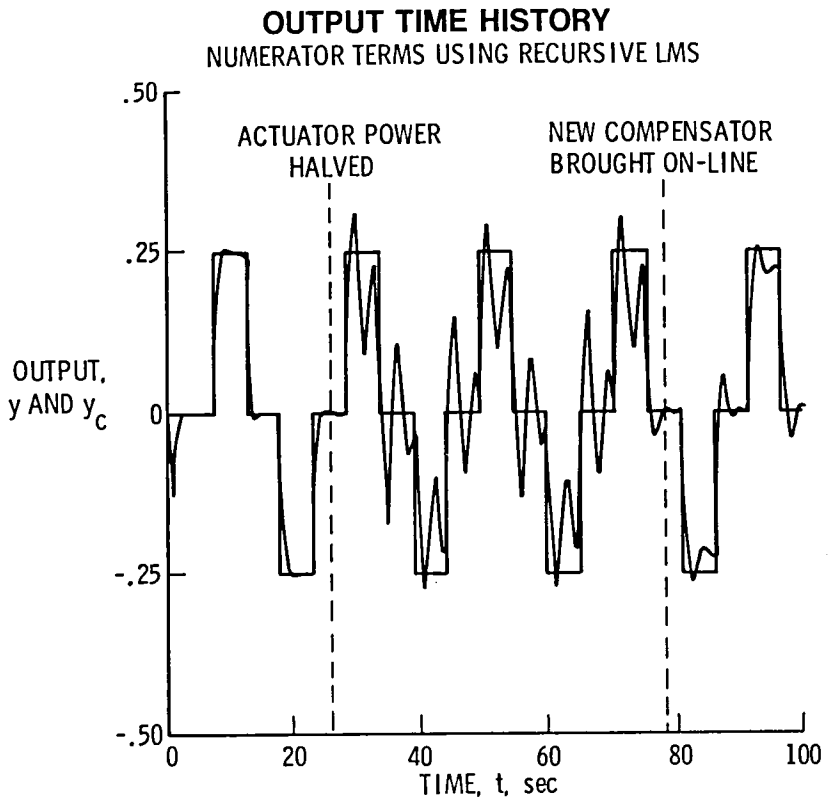

Figure VI-1 — Time history of output and commanded output for restructurable control example using recursive LMS algorithm.

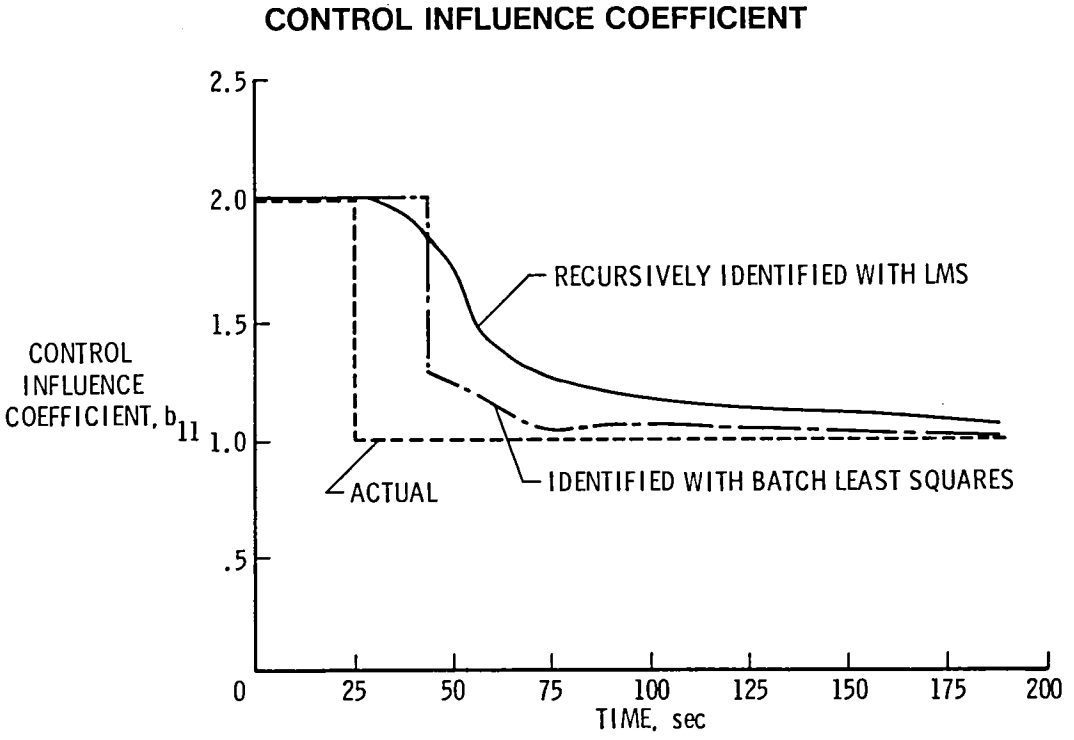

Figure VI-2 — Time history of actual control and estimated control influence coefficients using the recursive LMS and batch least square algorithms.

Figure (IV-2) shows time histories of the actual and estimated control influence term, b_{11}. The estimate lags (as observed in section II-D); improvements are made only within a few samples after each nonzero pulse.

Figure (VI-2) also shows a locus of estimated b_{11} from a single batch least squares estimate of the numerator terms as described in section II-E. It uses a minimum of the last 400 data points (20 seconds) or a maximum of the last 1000 data points (50 seconds). As can be seen, the batch least squares estimate for the numerator terms converges faster. At t=75 seconds the standard LMS has identified b_{11} with a 25 percent error, while the batch least squares approach has only a 5 percent error.

A time history of the simulation using the batch least squares approach for identifying the numerator terms is shown as figure (VI-3). When the new compensator is brought on-line, it stabilizes the system to nearly the same damping ratio as before the change in actuator characteristics and has a small commanded output error.

The performance index for the controller is the RMS commanded output error (c.f. from eq. (5.35)). A plot of this parameter, normalized by y_{com}, is shown as a time history in figure (VI-4). When the new compensator is implemented at t=75 seconds, the error is significantly reduced. However, if a goal of having σ_c under 10 percent was set into the computer logic (see fig. (V-2)), then the adaptive scheme attempts to improve by implementing another compensator. It continues to do so until the goal is reached. The algorithm with just the LMS filter achieves this goal after t=120 seconds.

There are some weaknesses in the scheme. If b_{11} had become .25, say, then the closed loop system would have gone unstable. The LMS algorithm tends to diverge for unstable systems and the system may break before a new controller is brought on-line. However, with limits on the control input, u, the system may not be stabilizable anyway. Also, it take 50 seconds to bring an acceptable compensator on-line, which may not be fast enough for many applications.

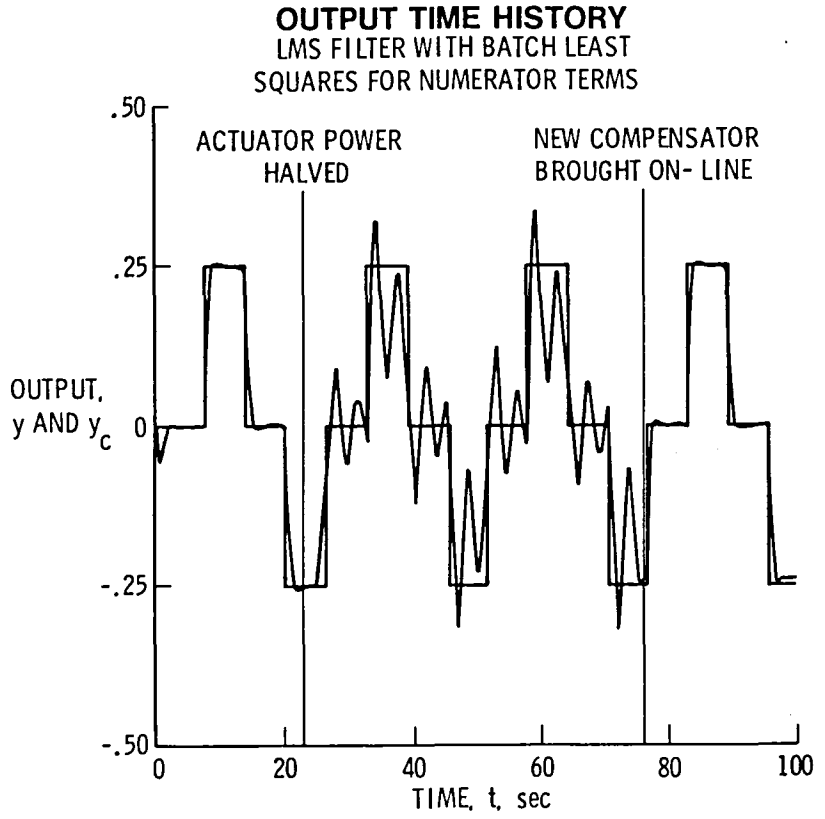

Figure VI-3 — Time history of output and commanded output for restructurable control example using recursive LMS algorithm in conjunction with batch least squares identification of the control influence terms.

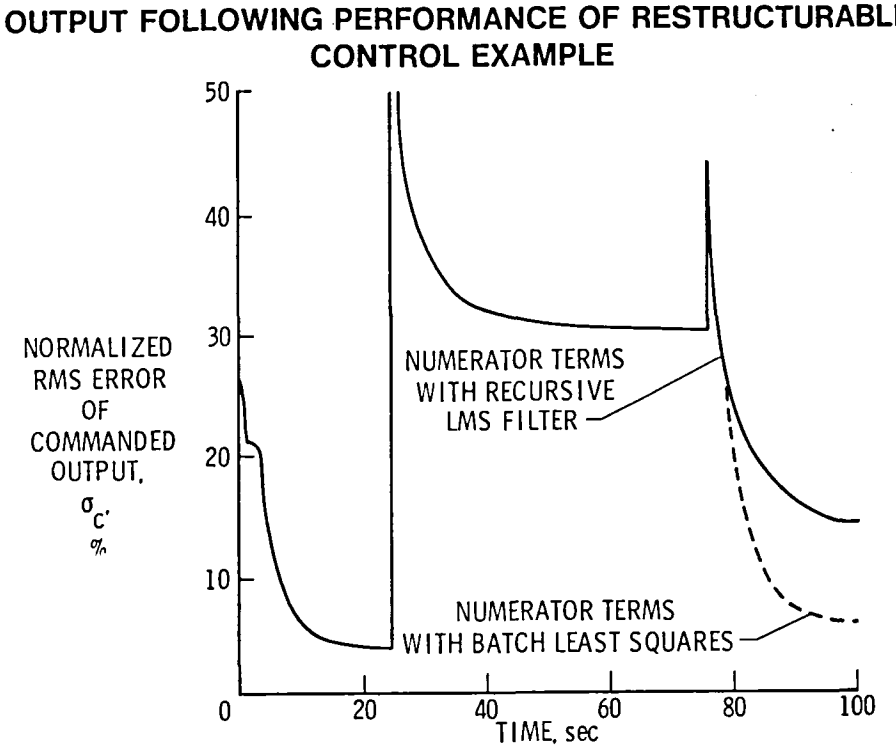

Figure VI-4 — Time history of normalized error of commanded output for restructurable control example.

B. MODEL BUILDING EXAMPLE

In this example a low order model of a flexible spacecraft is considered. The rigid body mode is, of course, known and the flexible body modes are identified one at a time by the IMLMS algorithm and added to the control model. Although the plant is stable, acceptable control performance is possible only when an accurate model is used for the compensator design.

Reference [67] shows that a sixth order model of the OSO-8 spacecraft [68,69] is possible when near pole zero cancellations are taken into account. In fact, figure (VI-5) is a convenient representation of the model, which is similar to the study of flexibility upon control in reference [21].

Using the state space form of equations (6.1) and (6.2), the matrices for the system in figure (VI-5) are

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{-k_1}{m_1} & 0 & \frac{k_1}{m_1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{k_1}{m_2} & 0 & \frac{-(k_1+k_2)}{m_2} & 0 & \frac{k_2}{m_2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{k_2}{m_3} & 0 & \frac{-k_2}{m_3} & 0 \end{bmatrix} \tag{6.15}$$

$$B = [0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0]^T \tag{6.16}$$

$$H = [0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0] \tag{6.17}$$

The OSO-8 spacecraft had the following equivalent characteristics for the system of figure (VI-5)

$$\frac{k_1}{m_1} = 13.96 \text{ sec}^{-1} \qquad \omega_1 = 16.5 \text{ sec}^{-1} \qquad \omega_2 = 40.64 \text{ sec}^{-1} \tag{6.18}$$

SIXTH ORDER SYSTEM EQUIVALENT TO REDUCED ORDER SPACECRAFT CONTROL PROBLEM

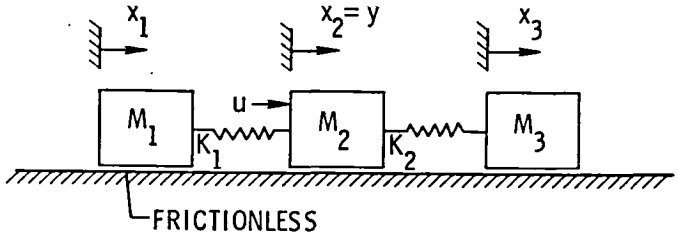

Figure VI-5 — Diagram showing equivalent system description of of reduced order spacecraft control model.

For the convenience of the simulation, let time be measured in units of ω_1 and mass in units of m_1. The model parameters for the simulation are chosen to be

$$\bar{m}_1 = \bar{m}_2 = \bar{m}_3 = 1 \tag{6.19}$$

$$\frac{k_1}{m_1} = 13.96 \text{ sec}^{-1} \tag{6.20}$$

$$\bar{k}_1 = .716 \qquad \bar{k}_2 = 2.817 \tag{6.21}$$

Viscous damping corresponding to a damping ratio of .005 was added to the two modes. The simulation is performed using the techniques of appendix I at a sample rate of 20 Hz. The output is commanded through a pulsed wave given by (6.13) with y_{amp}=.5 and Ω=.25 rad/sec.

A time history showing the output when a compensator was designed neglecting any knowledge of the flexible body modes is shown in figure (VI-6). The normalized RMS error of commanded output, σ_c, from equation (5.35) approached a steady state value of 18 percent. Additionally, the regulation about zero is apparently poor as a bias exists.

The time history in figure (VI-7) shows the output if a compensator is designed including the first flexible mode. The RMS of commanded output error, σ_c, approaches a steady state value of .126 and the bias about zero is no longer a problem. While the closed loop system is effectively regulated, stable and damped, the remaining flexible mode is lightly damped and prominent in the output. Clearly, the response is unsatisfactory for any precision control requirement.

Figure (VI-8) is a time history of the output when all three modes are included in the model to design the compensator. In this

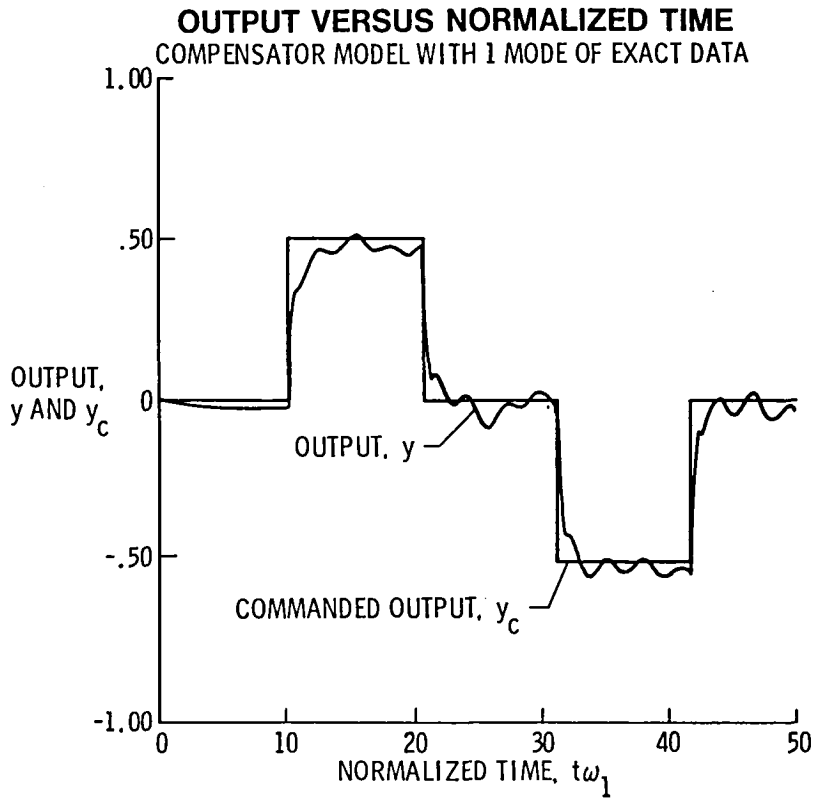

Figure VI-6 — Time history of output with rigid body mode modeled exactly for the suboptimal compensator design.

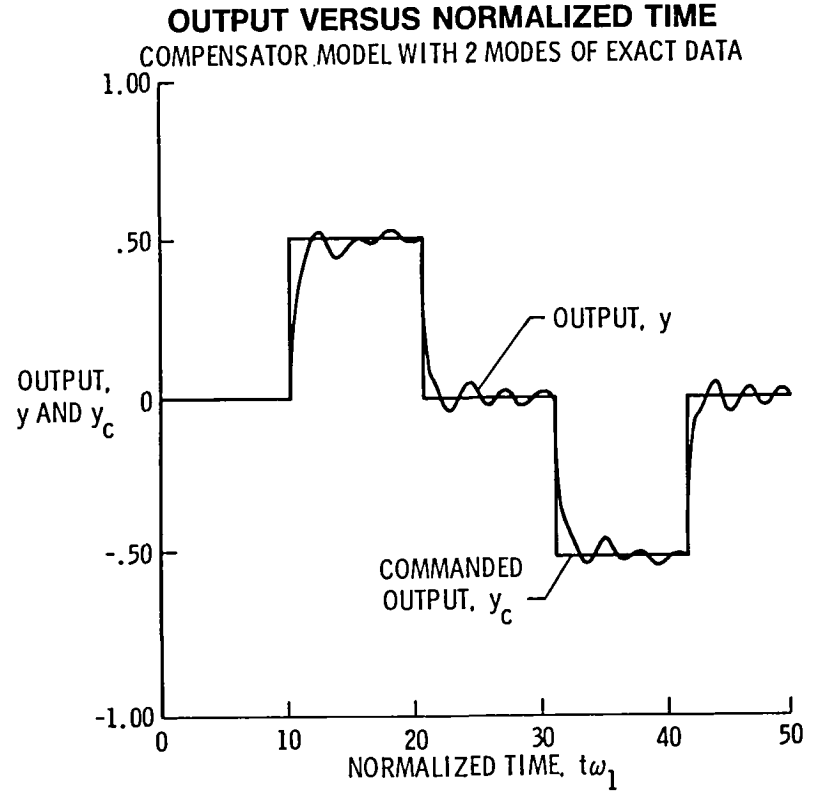

Figure VI-7 — Time history of output with rigid body and first flexible body mode modeled exactly for the suboptimal compensator design.

case, σ_c approaches a steady state value of .083 which is representative of the good performance. There is no steady-state error and the modes are fairly well damped. However, use of the optimal control design strategies described in chapter V leads to notch filter designs for the compensators [70,71]. Other methods of designing the control system may in fact lead to superior performance, especially given inaccuracies in knowledge about the dynamics.

Designing control systems for spacecraft can be difficult [73] because ground-based testing generally leads to inaccurate representations of spacecraft performance. Furthermore, large space structures may require months to construct. Although certain pointing and damping tasks may be required during this deployment stage, it would be difficult to model the configuration at each possible, intermediate step. For these reasons, it is desirable to have learning or adaptive control schemes similar to the one suggested by this research.

The IMLMS algorithm was used to build a model for this example. The rigid body dynamics are known and the output was commanded as in figure (VI-6) for 50 units of normalized time. At that time, the coefficients from the IMLMS algorithm are used to add the first flexible mode to the model used to design the compensator. The next 50 time units are shown in figure (VI-9). The normalized σ_c has been reduced from 18.2 percent to 13.1 percent, which is similar to the 12.6 percent of the compensator with exact knowledge for two modes only. Figures (VI-7) and (VI-9) look fairly similar. The steady-state error has been removed, but the undamped oscillation of the third mode still exists.

After 100 time units, the IMLMS algorithm is used to add the second flexible mode to the model. A time history of the ensuing motion is shown as figure (VI-10). The performance is nearly as good as the exact case shown in figure (VI-8). The RMS error of commanded output, σ_c, is equal to 10.1 percent versus 8.3 percent for the exact case. The output is well regulated and the modes are fairly well damped.

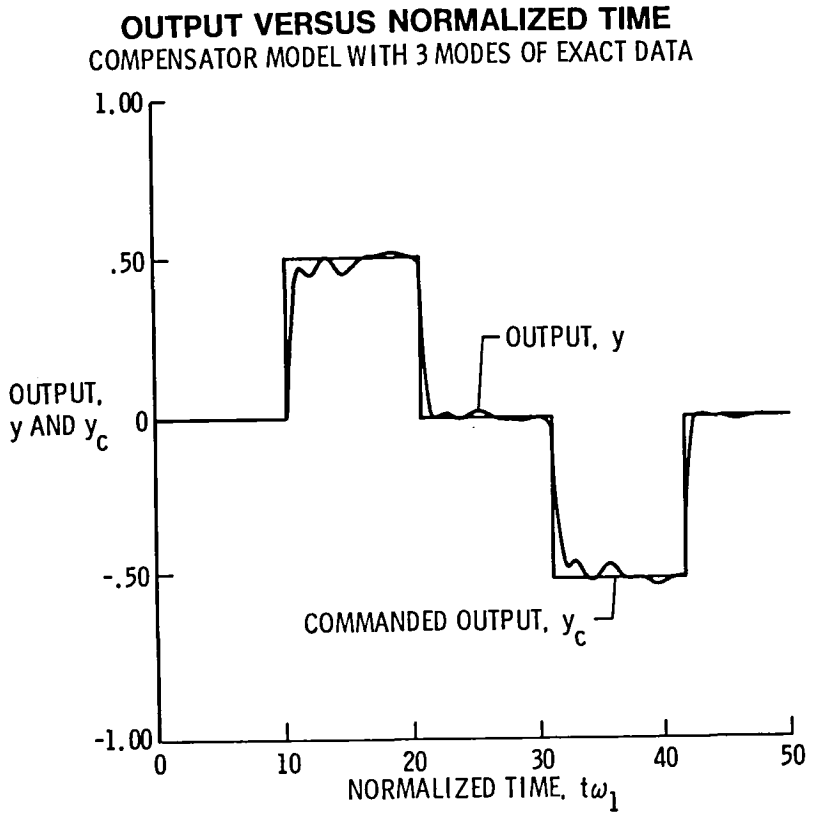

Figure VI-8 — Time history of output with plant modeled exactly for suboptimal compensator design.

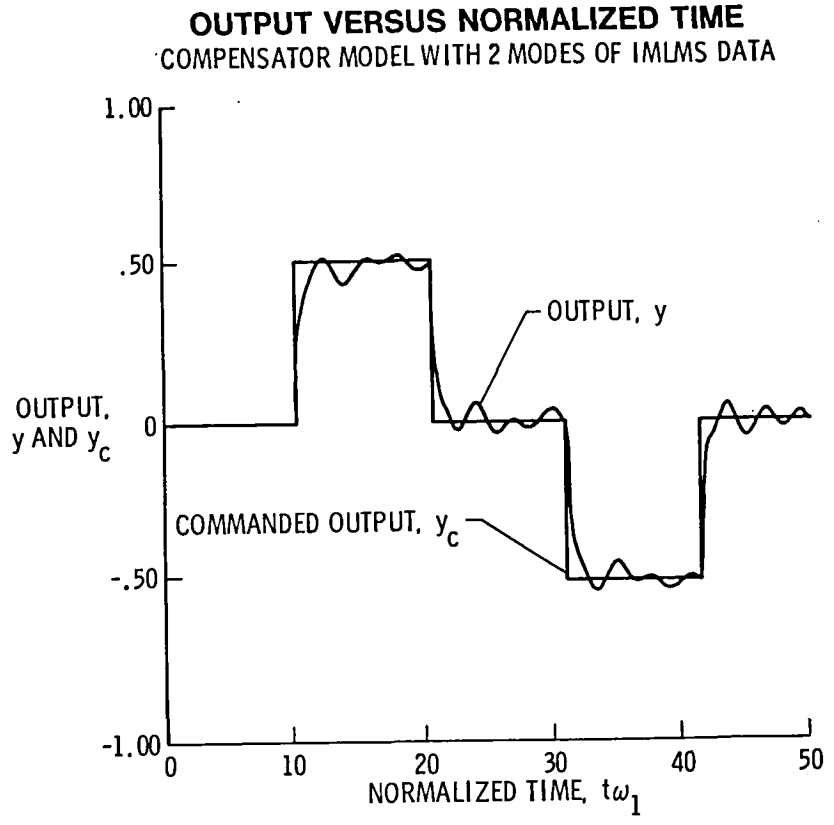

Figure VI-9 — Time history of output with rigid body mode modeled exactly and first flexible mode obtained from IMLMS filter for suboptimal compensator design.

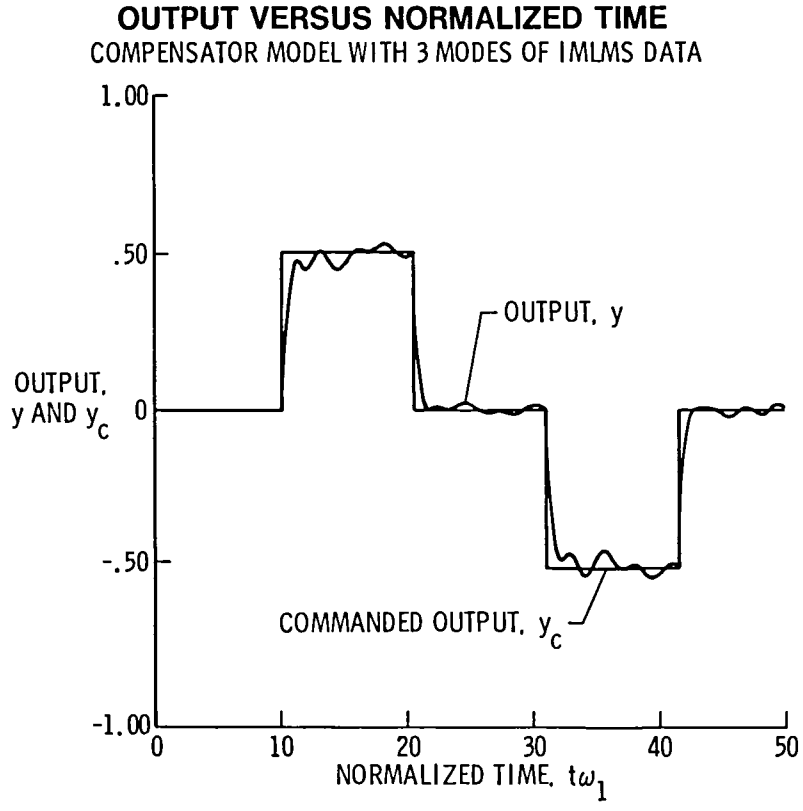

Figure VI-10 -- Time history of output with rigid body mode modeled exactly and both flexible modes obtained from IMLMS filter for suboptimal compensator design.

The compensator that results from the optimal control design is of the same order as its system model and contains a notch filter. This filter is highly sensitive to identification errors. Other design methods (e.g. [70,71]) may not be as sensitive to the parameter estimation accuracy of the flexible body modes, but may give poorer performance.

Simulations were also made using the LMS algorithm where all modes are identified at the same time. An improvement over no knowledge of the flexible body modes was possible using the LMS algorithm to identify and add these modes to the compensator model. However, small parameter errors translate into large dynamics errors as the order of the system to be identified increases. In fact, when the order of the system was assumed to be four, the LMS filter identified a negative real z-plane pole which is quite inaccurate.

The results of these simulations are summarized in table VI-1. The IMLMS algorithm used in conjunction with the model building scheme improves the accuracy and damping of the commanded response. One reason such good estimation accuracy is possible is because some prior knowledge about the flexible body modes is used in selecting the step-size scale factors of the IMLMS algorithm. The denominator of the discrete transfer function for a single damped mode can be written as [21]

$$d(z) = z^2 - 2e^{-aT}(\cos bT)z + e^{-2aT} \qquad (6.22)$$

T is the time between samples, a is the real part of the s-domain roots and b is the imaginary part of the s-domain roots. Since space structures are lightly damped, the real part of any pair of unaugmented roots will be approximately zero. So, if the coefficients of the LMS or IMLMS filters correspond to the following denominator equation,

$$d(z) = z^2 - \alpha_1 z - \alpha_2, \qquad (6.23)$$

the parameters can be equated as

Table 6.1 — Performance of adapted and nonadapted compensators in controlling the output of a reduced order model of the OSO-8 spacecraft.

SOURCE OF MODEL	EXACT			IMLMS		LMS	
NO. MODES IN MODEL	1	2	3	2	3	2	3
EIGENVALUES OF MODE 1	0,0	0,0	0,0	0,0	0,0	NA[2]	0,0
EIGENVALUES OF MODE 2	NA[1]	-.005 ± j	-.005 ± j	-.006 ± 1.04j	-.006 ± 1.04j	-.071 ± 1.19j	.006 ± 1.23j
EIGENVALUES OF MODE 3	NA[1]	NA[1]	-.012 ± 2.52j	NA[1]	-.013 ± 2.52j	NA[1]	-.19 ± 2.3j
NORMALIZED ERROR OF COMMAND FOLLOWING	.182	.126	.083	.131	.101	.169	.157

NA[1] Not available, mode is not included in truncated compensator model

NA[2] Not available, root found by LMS has imaginary frequency (negative real root on z-plane)

$$\alpha_1 \simeq 2(\cos bT) \tag{6.24}$$

$$\alpha_2 \simeq -1 \tag{6.25}$$

Since α_2 is approximately -1 with good accuracy (especially for high sample rates), it can be preset to -1 and the appropriate step size factor nulled. There is no need to identify a term that is known a priori. This, in effect, reduces the number of degrees of freedom, thereby improving the speed of adaption and the accuracy of convergence.

Even better accuracy for the model building adaptive controller using the IMLMS could be achieved by allowing more time for convergence and varying the input signals (especially pulse frequency) in a systematic way. A logical expectation is that minutes or possibly hours could be taken by a learning system to identify a model of high order to perform a control mission in space on a structure under modification or construction.

A verification of the MIMO capabilities of the algorithm was performed by adding a colocated actuator-sensor pair at mass m_3. The number of free parameters in the denominator polynomial coefficients remained unchanged so the identification accuracy remained approximately the same. Adequate pulsing of both controls for both outputs required more time to identify the coefficients of the numerator polynomial. Equivalent accuracy for the numerator terms was obtained in twice the time as used for the SISO system. As previously illustrated, the full model built by using the IMLMS algorithm resulted in fairly similar output as using the exact model in the compensator design. The inclusion of the additional sensor and actuator improved the damping and yielded σ_c's of .038 and .051 for the exact model and IMLMS sequentially identified models, respectively. Hence, no problems were encountered extending the scheme to MIMO systems.

C. ADAPTIVE CONTROL IN THE PRESENCE OF SENSOR NOISE

Many of the adaptive control algorithms that have been studied diverge in the presence of observation noise (c.f. chapter I). A small bias (constant or oscillatory) in the sensor output produces divergence. This instability is avoided by separating the system identification and the controls tasks. Only occasionally is the information allowed to flow to the control block. Hence, the adaptive control scheme of this research will not exhibit any worse performance than that from using a time-invariant, suboptimal controller. In this example, the adaptive control scheme proposed in this research is used to augment sensor performance.

Assume for this example a linear, time-invariant plant which is known exactly and has the following familiar form

$$\dot{\underline{x}} = A\underline{x} + Bu \tag{6.26}$$

with output

$$y = H\underline{x} + s_b + s_a \sin(s_\omega t) \tag{6.27}$$

where s_b is the sensor bias, s_a is the sensor noise amplitude and s_ω is the sensor noise frequency. The bias and noise are, of course, uncontrollable by u. The system matrices were chosen from an unstable Dutch roll mode of an airplane and are given by:

$$A = \begin{bmatrix} 0 & 1 \\ -1 & -.2 \end{bmatrix} \tag{6.28}$$

$$B = [0 \quad 2]^T \tag{6.29}$$

$$H = [2 \quad 0] \tag{6.30}$$

The open loop eigenvalues are $s=-.1\pm j$, damping ratio of .1 and a natural frequency of 1 rad/sec.

A digital compensator of the form of equations (6.6) and (6.7) was designed to control the plant output. The equivalent discrete system, for a sample rate of 20 Hz, has system matrices

$$\Phi = \begin{bmatrix} .999 & .049 \\ -.049 & .989 \end{bmatrix} \tag{6.31}$$

$$\Gamma = [.002 \quad .099]^T \tag{6.32}$$

The gain matrices for the digital compensator, designed using the methods of chapter V, are

$$K = [.519 \quad .361]^T \tag{6.33}$$

$$C = [-55.8 \quad -7.40] \tag{6.34}$$

$$F = [28.2] \tag{6.35}$$

The closed loop eigenvalues are z=.586 + .289j. The output is commanded to follow equation (6.13) with y_{amp}=.25 and Ω=.2 rad/sec.

In this example, an extreme case is chosen with s_b equal to .05, s_a equal to .05, and s_ω equal to 3. Figure (VI-11) shows the actual and commanded output for s_a=01. The corresponding sensor output is plotted in figure (VI-12). The control effort expended to reduce the oscillation in the sensor output results in exciting an oscillation in the plant output, which is clearly not desirable.

After 100 seconds of recursive identification using the IMLMS algorithm, an extra mode is added to the compensator model. The sensor noise frequency was identified within 2 percent and the control influence terms were very small, less than 10^{-4}, making the mode uncontrollable for the tolerance levels selected in the software design package [51]. The result of applying the new model to the system is shown in figures (VI-13) and (VI-14).

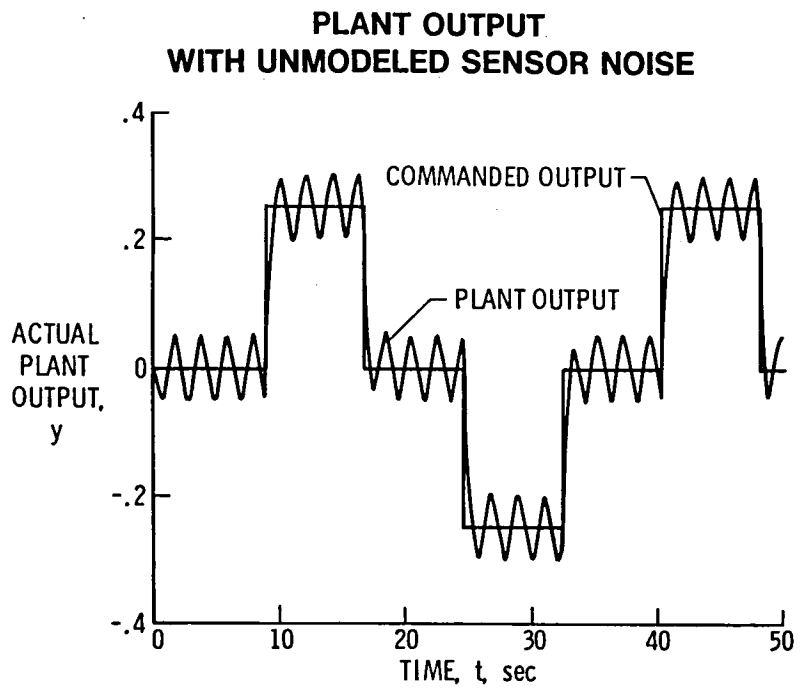

Figure VI-11 — Time history of plant output with oscillatory sensor noise.

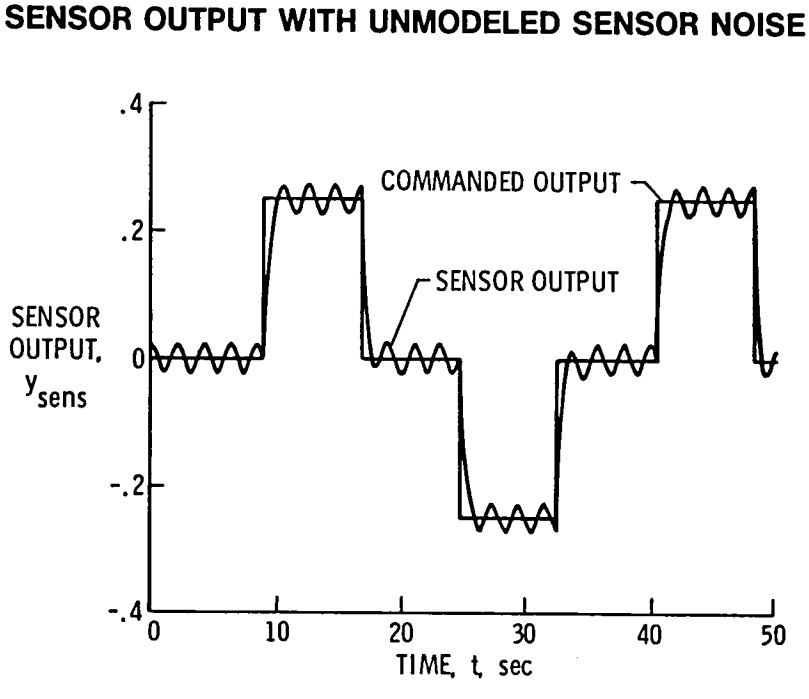

Figure VI-12 — Time history of sensor output with oscillatory sensor noise.

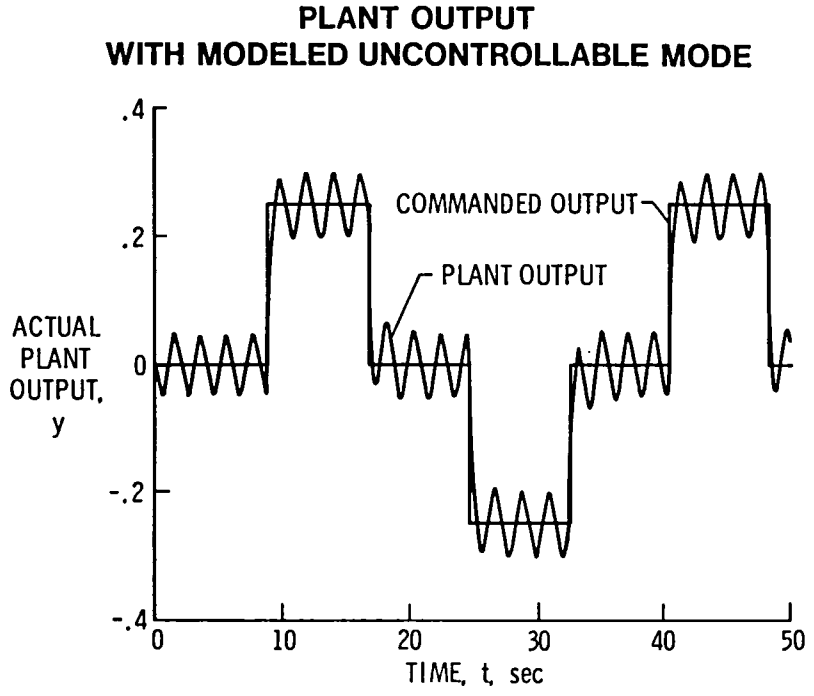

Figure VI-13 — Time history of plant output with oscillatory sensor noise. A mode has been added to the compenstator model from the IMLMS filter.

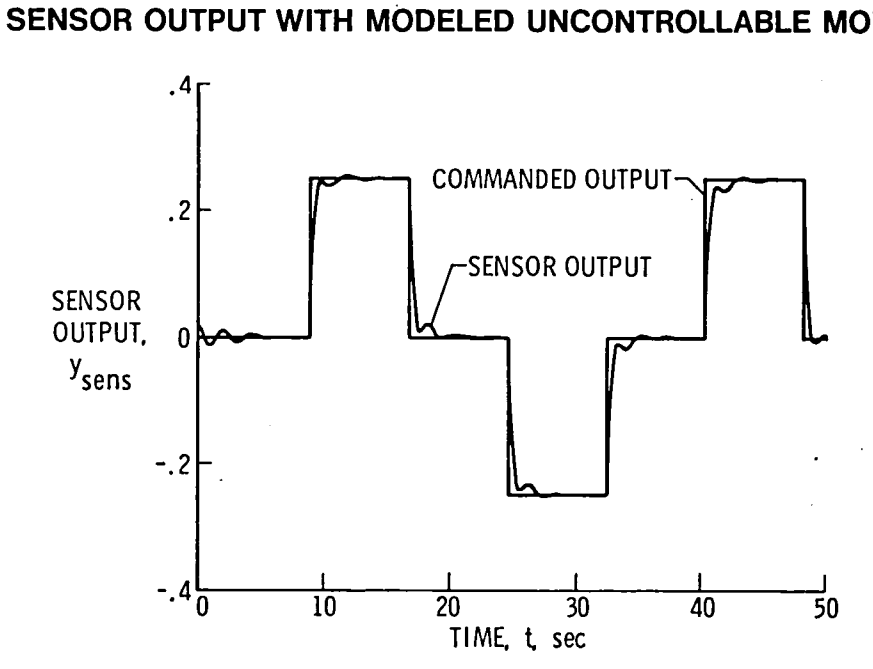

Figure VI-14 — Time history of sensor output with oscillatory sensor noise. A mode has been added to the compensator model from the IMLMS filter.

The plant output in figure (VI-13) remains virtually unchanged, so the performance is not impacted by knowledge of the sensor noise. However, the sensor output is significantly better as shown in figure (VI-14). This is a result of using output in the cost function for finding the control law; this causes the extra mode to be included as if it were a physical mode of the plant. While the perceived performance is enhanced, the actual performance is poor, indicating the need to identify and minimize sensor errors.

The problem can be solved if the oscillatory mode can be classified as sensor noise. This is accomplished by testing the identified mode to see if it satisfies some threshhold of controllability. Equation (5.11) is then modified such that no ouput from the uncontrollable mode appears in the y used in the cost function for computing the controller feedback gains. This makes y equivalent to the plant output instead of the sensor output. The results from this experiment are shown in figures (VI-15) and (VI-16). The plant output has relatively good performance and the sensor output has reverted to oscillations. In effect, the uncontrollable but detectable mode is subtracted from the total sensor output to estimate the plant mode. This made it possible to observe and control the actual plant output in spite of the observation noise. However, the problem would still exist for cases where prior knowledge does not allow one to distinguish between sensor noise and an uncontrollable mode of the plant.

The influence of sensor bias is shown in figures (VI-17) and (VI-18). The plant output is well damped, stable and follows the commanded output well, except that it is offset from the desired output by s_b. The sensor output is also offset slightly, but to a smaller extent. The bias was not included in the control model and the threshholds that were chosen for the adaptive logic did not result in the addition of another mode when used in parallel with the IMLMS algorithm. If good following performance is desired, the sensor bias needs to be small.

A main observation from this example is that the present adaptive control scheme does not diverge in the presence of uncontrollable modes. Thus, if some heuristic decision logic can be

SENSOR OUTPUT WITH SENSOR NOISE MODELED

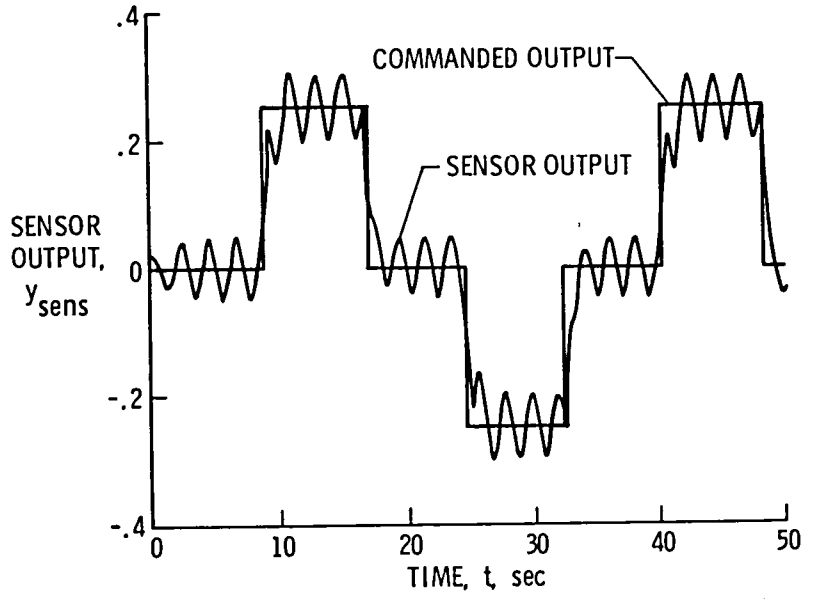

Figure VI-15 — Time history of plant output when uncontrollable mode is modeled as sensor noise.

PLANT OUTPUT WITH MODELED SENSOR NOISE

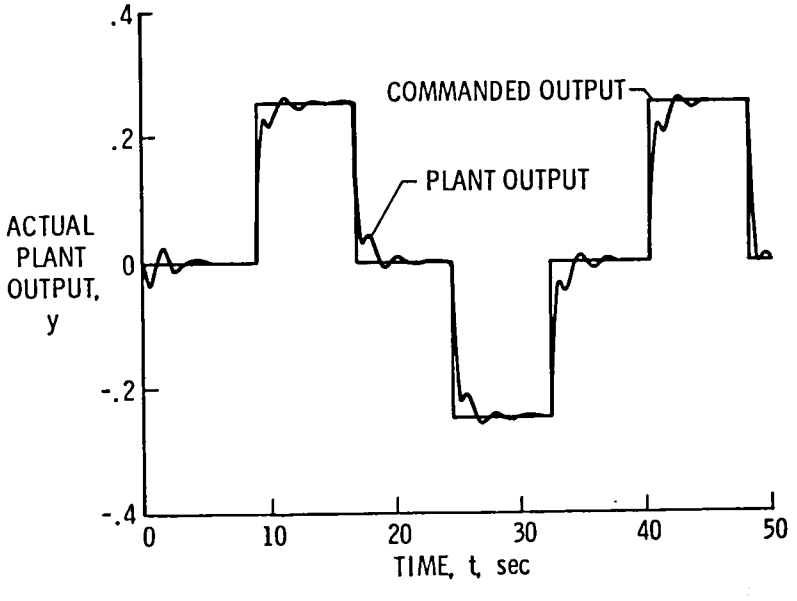

Figure VI-16 — Time history of sensor output when uncontrollable mode is modeled as sensor noise.

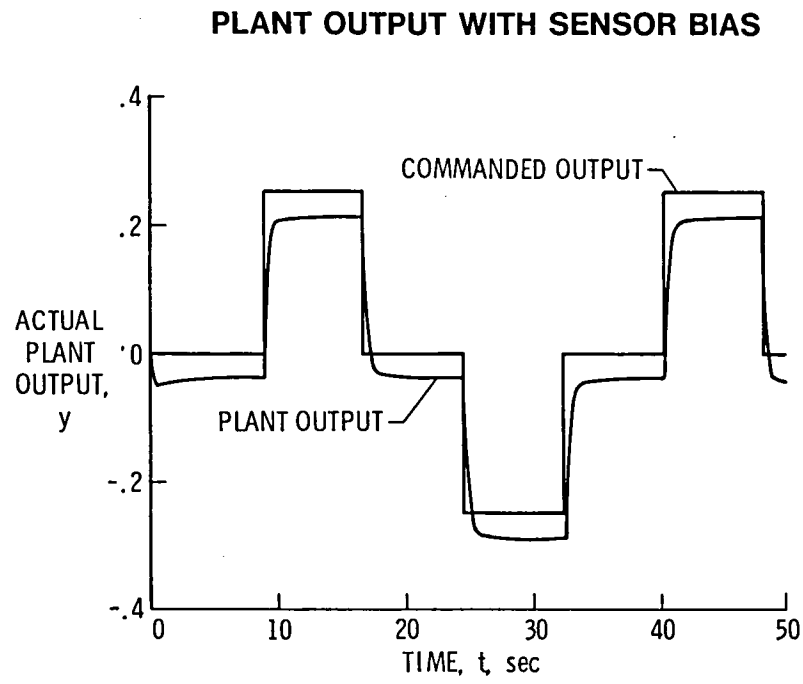

Figure VI-17 — Time history of plant output with an unmodeled sensor bias.

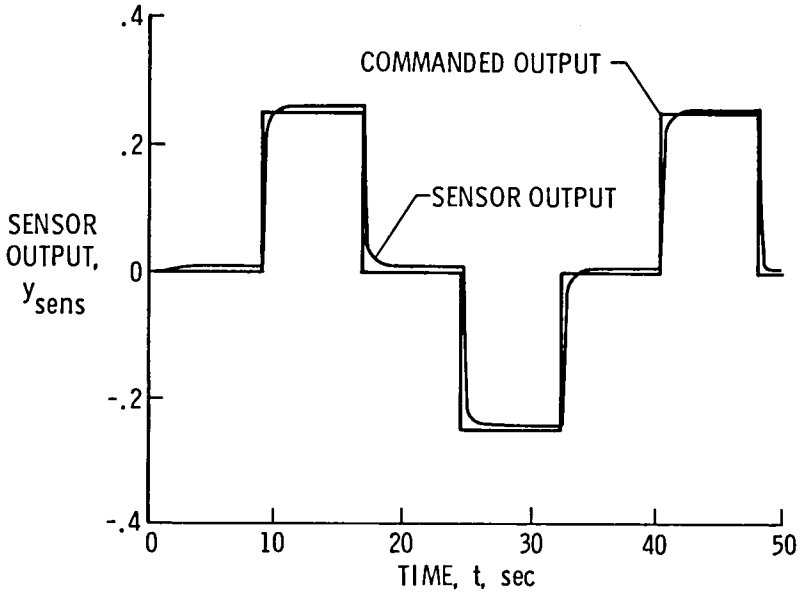

Figure VI-18 — Time history of sensor output with an unmodeled sensor bias.

implemented to distinguish between an uncontrollable mode of the plant and observation noise, it is possible to use this adaptation scheme to improve plant performance in the presence of such sensor noise.

CHAPTER VII

CONCLUDING REMARKS AND RECOMMENDATIONS

A. SUMMARY

A scheme for the adaptive control of multi-input/multi-output (MIMO) plants is proposed that utilizes heuristic logic. Control law restructuring takes place intermittently after satisfying a number of convergence tests instead of continuously. The plant is constructed by starting with a simple model and identifying one additional mode at a time until the added mode makes no substantial improvement in closed-loop performance. The recursive identification scheme which allows the adaptation of part of the assumed model while holding the remaining part fixed is a modification of the LMS (Least Mean Square) algorithm. It is termed the incremental mode LMS (IMLMS) algorithm.

On-line identification is performed using an extension of the LMS algorithm of Widrow and Hoff to MIMO systems. This is a recursive identifier having the same form as the extended Kalman filter (EKF), but it uses constant instead of time-varying gains. Hence, it does not require the propagation of a large covariance matrix, and it avoids the "oblivious filter" problem that occurs with the EKF when the variance of the estimated parameters becomes small. Excitation for identification is performed using rectangular pulse output-commands.

Only the coefficients of the denominator polynomial of the MIMO z-transfer function are identified using the IMLMS algorithm. The coefficients of the numerator polynomials are identified using a batch least squares algorithm over the last several output commands. Control law synthesis is done by LQG methods which require the formation of a MIMO state space realization from the z-transfer function identified by the IMLMS algorithm. An efficient regression algorithm is presented to do this and is illustrated by a second order example with two inputs and two outputs.

Three adaptive control examples were presented: (1) a control restructuring example where a partial failure of the actuator resulted in a change in actuator gain; (2) a model building example of a spacecraft with a rigid body mode and two flexible body vibration modes; and, (3) a sensor noise example where uncontrollable oscillatory noise appears in the sensor signal, but not in the plant output being controlled.

As in previous adaptive control schemes, there is no way of discerning whether detectable but uncontrollable modes appear in the actual plant output or only in the sensor output. If this determination can be made by some other means, then an appropriate model can be formed.

The proposed scheme for adaptive control is not suitable for situations where rapid adaptation is required; in particular, it will not handle plants with significant instabilities.

B. CONCLUDING REMARKS

Several continuously adapting control algorithms have been proven to be globally stable. It has been shown, however, that the assumptions required for proving stability are overly restrictive. In fact, divergence is likely when operated with sensor noise, with biases in controls or measurements, or with unmodelled dynamics [17-20] present. The goal of this research was to extend a simple recursive estimation algorithm to a form which will minimize the impact of these real world considerations. This is accomplished with the proposed adaptive control scheme, but the added complexity prevents proving global stability.

The LMS algorithm was selected as the baseline parameter estimator because of its low computational burden and good convergence characteristics in the presence of process noise, sensor noise and unmodelled dynamics. The algorithm was enhanced by the systematic inclusion of MIMO systems. A prime contribution of this research is the development of the IMLMS algorithm which provides a systematic means for including unmodelled dynamics. The IMLMS algorithm is a recursive identification scheme that estimates the parameters of

part of the assumed model while holding the remaining part of the model constant. This provided the basis for an on-line structure identification scheme using heuristic rules to build models.

Although linear equivalent systems are sought as models, plants with small or locally insignificant nonlinearities will be readily controlled by the proposed adaptive control scheme. Plants with large nonlinearities which prevent the ready representation as a linear system will not be easily controlled by this or any adaptive controller now being considered in the research literature. Exceptions would be cases where complex plants are modelled with good physical understanding and have only a few time varying or uncertain parameters. In this research, it is assumed that very little prior knowledge about the plant is available, necessitating model learning and building in real time. The approach in this research is useful for plants which can be effectively linearized, but have a wide range of uncertain modal characteristics or are slowly changing.

The model used in the compensator for controlling the outputs was updated only after satisfying a number of convergence tests and passing performance threshholds. By preventing the controller gains from being updated at each sample, the divergence problems that plague continuously adaptive control systems are avoided. Unfortunately, this approach of waiting to evaluate performance and convergence properties on-line prevents direct analysis. Without the ability to directly analyze the overall scheme, proofs for global stability are impossible to obtain. However, this research does provide practical solutions to the generic types of problems that adaptive control algorithms suffer from. Although mathematical rigor is replaced with engineering judgement, this scheme of using heuristic logic points to a class of learning controllers which may have high reliability for a wide range of applications.

A final concern of the adaptive control scheme proposed in this research is the added complexity of utilizing heuristic logic to avoid divergence. Software reliability is already a difficult

problem and an area of intense research. The use of simple hypothesis testing may make the issue of program quality assurance a significiant hindrance in its use for other than highly experimental or back-up applications of adaptive control. Additionally, the proper balance between control theory and machine intelligence concepts needs to be considered in terms of problem solving capability, computer requirements, problem application and software reliability.

C. RECOMMENDATIONS FOR FURTHER STUDY

The prime recommendations for developing enhanced adaptive control methodologies are those involved with making adaptive control a practical alternative to robust control. As a first step, the results reported in this document should be verified through hardware implementation. A dual microprocessor control experiment for a lightly-damped, flexible beam using the algorithms of this report is being considered for the experimental apparatus described in reference [73]. It will provide a benchmark for comparing with other real-time adaptive control experiments. Trade-offs between recursive identification and batch processing will be possible, plus the computational efficiency of performing the compensator design can be evaluated. It could also be an effective check of the approach of using heuristic adaptive logic for dealing with practical problems as they arise. It may be more cost effective to perform hybrid (digital and analog) simulations than trying to perform accurate, high-order, digital simulations.

One advantage of hardware experimentation is that the LMS adaptive filter was originally developed for analog implementation. It may be possible to find a similar way to implement the IMLMS filter in a hybrid control system. Analog implementation of the recursive identification techniques could provide faster adaptation, more accurate parameter estimation and less sensitivity to the numerical convergence, especially at high sample rates. However, the use of analog adaption will introduce sampling errors into the parameter estimates during digitizing.

Lattice filters [74] are a potentially attractive alternative to the LMS class of adaptive algorithms considered in this research. It has been shown that an estimate for the system order can be recursively included in the identification scheme [75], making rapid model identification possible. If n is the model order, the lattice filter algorithm has approximately 10n more computations per cycle than the LMS algorithm [74]. It does, however, provide the capability to iterate the model order based upon hypothesis testing at frequent, recursive intervals. Research shows that it can be useful for the identification of space structures [76], but its use for cases with substantial damping has not been verified. Furthermore, direct use of the lattice filter output for compensator design is still of concern and needs further research. Another shortcoming is that the lattice filter cannot explicitly include any prior knowledge of the plant. In spite of these limitations, the use of lattice filters with their model order iteration capability and low computational burden does look to be promising for future adaptive control research.

The use of quadratic cost techniques may not be ideal for on-line compensator design. An alternative design technique should be considered for the observer and controller implementation. It may be possible to utilize computationally efficient and robust algorithms that take advantage of the modal forms preserved from the recursive identification schemes. In addition, the feedforward controller design algorithm should be extended to handle a different number of control inputs from output measurements [64] and to be able to follow other than step inputs with arbitrary placement of the zeros [21].

An area of active research that all adaptive control approaches would benefit from is the continued development of a method for determining optimal control inputs for system identification [43-45]. Implementation of a full dual control scheme--where control power expenditures for improving identification, as well as for

performance requirements, are explicitly included in the cost function that is optimized--appears to be too computationally burdensome at this time. However, an inexact technique may be possible whereby suboptimal inputs are computed or suggested by the adaptive strategies implemented in the computer by designers.

In this vein of "smart" controllers, whole new avenues need to be explored. It may be possible to couple an "expert" system from the science of artificial intelligence utilizing rules and knowledge from experienced controls engineers to help build an on-line adaptive controller. An adaptive control scheme is envisioned which could be successfully applied to a wide variety of problems while requiring a minimum of problem dependent fine-tuning and still maintain a high probability of success. However, a logical next step would be merging the science and rules under development for artificial intelligence with the formal analysis frequently applied to control system design. This may eventually result in a flexible and fast "universal controller" which could be used to improve control system performance for many types of problems.

APPENDIX I

DESCRIPTION OF SIMULATION EXPERIMENTS

A. ADVANCED CONTINUOUS SIMULATION LANGUAGE (ACSL)

The computer simulations were performed using the CDC Cyber network at Langley Research Center. ACSL (Advanced Continuous Simulation Language) [77] was the prime tool used in these studies. It provides a facility for performing integration of nonlinear equations of motion and for obtaining a wide variety of outputs from either interactive or batch use. ACSL interprets the set of commands given it, which look like FORTRAN, and writes a FORTRAN program for the user. Hence, it is relatively easy to couple FORTRAN subroutines to the ACSL program.

The simulations shown in this thesis were performed using a second order Runge-Kutta integration scheme with a step size at least as small as the sample time of the discrete compensator and identification schemes. The program was segmented so the required field length at execution would be small enough to allow interactive execution. Graphic output was obtained from Tektronix copiers during interactive simulation and Varian plotters for batch processing.

Subroutines were added to the ACSL programs to provide discrete sampling with zero order hold, the compensator and control law implementation, performance evaluation and the heuristic adaptive decison making. A brief description of these routines is given in table I-1. Program flexibility was maintained making it relatively easy to interchange models and adjust parameters during the debugging stages of the adaptive control law development. Input decks minimized the need to recompile the entire simulation during problem formulation and evaluation.

Table I-1 — Subroutines used to augment ACSL for simulating adaptive control schemes.

SUBROUTINE	DESCRIPTION
ADMODE	Adds modes to compensator model during model building process.
COMMAG	Selects input pulses for identification.
COMND	Distributes pulses to appropriate control or output command.
CONTROL	Computes control input from feedback and feedforward law.
ERREVAL	Accumulates figure-of-merit computation for adaptive control.
FORM	Assembles dynamics matrix from modal information.
IMLMS	Performs IMLMS algorithm computation. (see app. II)
KFILT	Computes state estimates using asymptotic KBF gains.
LMS	Performs LMS algorithm computation with MIMO extensions (see app. II)
SAMPCK	Code for implementing discrete sampling logic.
SETUP	Routine for initializing plant and compensator model selection.

B. OPTIMUM REGULATOR ALGORITHMS FOR THE CONTROL OF LINEAR SYSTEMS (ORACLS)

The code used for performing the matrix operations used by ACSL and the compensator designs as described by chapter V is ORACLS [51]. ORACLS is a library of subroutines for synthesizing control logic for multivariable state space systems. ORACLS also provides a set of routines for matrix operations and manipulations.

ORACLS was coupled to ACSL making on-line compensator design possible. However, a number of routines had to be added to customize the library to the present problem. In addition, several analysis routines were added to aid in debugging the adaptive controller. Descriptions of the subroutines that were used to augment the ORACLS library are given in table I-2.

Arrays for ORACLS are stored as a single subscripted array stacked by columns in combination with an integer pointer array. For example, if a 3 X 2 matrix A is to be used by a routine from ORACLS, it is saved as array A(I) of length 6, where the first three elements are the first column of A and the last three elements are the second column. In addition, an integer array NA(J) of length 2 is also used. The first value is the number of rows and the second value is the number of columns.

Table I-2 — Subroutines used to augment ORACLS to facilitate on-line implementation for adaptive control schemes.

SUBROUTINE	DESCRIPTION
COMPEN	Computes gain matrices for asymptotic, suboptimal compensators.
CONVRT	Transforms roots between s-plane and z-plane.
CTRLAW	Computes control law for asymptotic, discrete, suboptimal regulator.
ELEMENT, GRAB, ETC.	Several matrix manipulation routines to faciliate the use of ORACLS.
INIT	Initializes common blocks and local variables for ORACLS and ACSL.
KFILTD	Computes gain matrices for asymptotic, discrete Kalman-Bucy filter.
MATINV	Finds general matrix inverse including pseudo inverses for non-square matrix.
PHIGEN	Computes discrete equivalents for linear system matrices.
POLZER	Finds poles and zeros of linear system on both s-plane and z-plane.
SETPT	Computes the feedforward gain matrix for non-zero set point of regulator.

APPENDIX II

COMPUTER CODES FOR OUTPUT ERROR IDENTIFICATION

This appendix contains FORTRAN listings of the LMS and IMLMS recursive, adaptive filters. The subroutines are called at each discrete time step to update the predicted output and the parameter estimates. YP is the predicted output. The storage arrays for computing the ARMA predictions are XS and US and are updated each time step. XNORM and UNORM are used to normalize the inputs to the filter as a means of scaling the data to near unity as suggested in chapters II and III. The parameters being updated are ALPHA and BETA, which are the coefficients of denominator and numerator polynomials, respectively.

Subroutine listing of FORTRAN code for LMS Adaptive Filter.

```
      SUBROUTINE LMS(IORD,XMU,ALPHA,BETA,Y,NY,YNORM,U,NU
     2               ,UNORM,XS,US,YP,ERR,IOUT)
************************************************************************
*           SUBROUTINE LMS IS THE LMS ADAPTIVE FILTER USED AT EACH     *
*               TIME STEP.  IT IS USED FOR ADAPTIVE ESTIMATION         *
*               AND PARAMETER IDENTIFICATION.                          *
************************************************************************
*                                                                      *
*           INPUTS:                                                    *
*                                                                      *
*               IORD IS THE ORDER OF THE DENOMINATOR                   *
*               XMU IS THE STEP SIZE FACTOR                            *
*               ALPHA ARE THE DENOMINATOR COEFFICIENTS                 *
*               BETA ARE THE NUMERATOR COEFFICIENTS                    *
*               Y IS THE MEASUREMENT OF THE OUTPUT                     *
*               U IS THE INPUT (CONTROL)                               *
*               XS IS A STORAGE ARRAY OF LAST Y'S                      *
*               US IS A STORAGE ARRAY OF LAST U'S                      *
*               XNORM AND UNORM ARE NORMALIZING FACTORS                *
*               NY AND NU ARE THE DIMENSIONS OF Y AND U                *
*               IOUT=1 FOR OUTPUT                                      *
*                                                                      *
*           OUTPUTS:                                                   *
*                                                                      *
*               ALPHA & BETA ARE UPDATED EACH TIME STEP                *
*               XS AND US ARE UPDATED EACH TIME STEP                   *
*               YP IS THE PREDICTED OUTPUT FOR THE CURRENT TIME        *
*               ERR IS THE ERROR OF PREDICTION FOR THE CURRENT TIME    *
************************************************************************
      REAL ALPHA(1),BETA(2,2,2),Y(1),U(1),XS(2,2),US(2,2)
      REAL YP(1),ERR(1)
      INTEGER NY(1),NU(1)
      IMEAS=NY(1)
      ICTRL=NU(1)
C
C     COMPUTE PREDICTION
C
      DO 200 L=1,IMEAS
      XSTO=0.
      USTO=0.
      DO 190 I=1,IORD
      XSTO=XSTO+ALPHA(I)*XS(I,L)
  190 CONTINUE
      DO 195 I=1,IORD
      DO 195 J=1,ICTRL
      USTO=USTO+BETA(I,J,L)*US(I,J)
  195 CONTINUE
      YP(L)=USTO+XSTO
```

```
  200 CONTINUE
C
C     COMPUTE ERR & NEW WTS
C
      DO 250 L=1,IMEAS
      ERR(L)=Y(L)-YP(L)
  250 CONTINUE
      DO 300 I=1,IORD
      DO 300 L=1,IMEAS
      ALPHA(I)=ALPHA(I)+2.*XMU*XS(I,L)*ERR(L)
  300 CONTINUE
      DO 305 J=1,ICTRL
      DO 305 I=1,IORD
      DO 305 L=1,IMEAS
      BETA(I,J,L)=BETA(I,J,L)+2.*XMU*US(I,J)*ERR(L)
  305 CONTINUE
C
C     UPDATE MEAS & CTRL STATES
C
      DO 320 L=1,IMEAS
      DO 310 II=2,IORD,1
      I=IORD+2-II
      XS(I,L)=XS(I-1,L)
  310 CONTINUE
      XS(1,L)=Y(L)/YNORM
  320 CONTINUE
      DO 330 J=1,ICTRL
      DO 325 II=2,IORD,1
      I=IORD+2-II
      US(I,J)=US(I-1,J)
  325 CONTINUE
      US(1,J)=U(J)/UNORM
  330 CONTINUE
C
C     DEBUGGING OUTPUT
C
      IF(IOUT.EQ.0) GO TO 500
      WRITE(6,410) IORD,IMEAS,ICTRL
  410 FORMAT(/5X"     LMS ALGORITHM OUTPUT     "/10X,"SYSTEM ORDER= "
     1 ,I5/10X,"NO. OF MEAS.= ",I5/10X,"NO. OF CTRLS.= ",I5)
      WRITE(6,415)(ALPHA(I),I=1,IORD)
  415 FORMAT(T20,"ALPHA:"5(T20,5F12.5/))
      WRITE(6,420)
  420 FORMAT(10X,"BETA:")
      DO 430 J=1,ICTRL
      DO 430 L=1,IMEAS
      WRITE(6,425)J,L,(BETA(I,J,L),I=1,IORD)
  425 FORMAT(10X,"CTRL= ",I5,5X,"IMEAS= ",I5,5(T50,5F12.5,))
  430 CONTINUE
      WRITE(6,440)
  440 FORMAT(/10X,"STATES:")
      DO 450 L=1,IMEAS
      WRITE(6,445) L,(XS(I,L),I=1,IORD)
```

```
445 FORMAT(10X,"MEAS= ",I5,5(T30,5F12.5,))
450 CONTINUE
    WRITE(6,455)
455 FORMAT(/10X,"CONTROL STATES:")
    DO 460 J=1,ICTRL
    WRITE(6,458) J,(US(I,J),I=1,IORD)
458 FORMAT(10X,"CONTROL= ",I5,5(T30,5F12.5,))
460 CONTINUE
500 RETURN
    END
```

Subroutine listing of FORTRAN code for IMLMS Adaptive Filter.

```
      SUBROUTINE IMLMS(IDEN,INUM,XMUD,XMUN,ALPHAF,BETA,IDAD,ALPHAI
     2                ,Y,NY,YNORM,U,NU,UNORM,XS,US,YP,ERR,IOUT)
*****************************************************************
*              SUBROUTINE IMLMS IS THE INCREMENTAL MODE LMS      *
*                   ALGORITHM APPLIED AS A RECURSIVE IDENTIFIER  *
*                   ON PART OF THE MODEL.                        *
*****************************************************************
*                                                               *
*              INPUTS:                                          *
*                                                               *
*                   IDEN IS ORDER OF FIXED PART OF DENOMINATOR  *
*                   INUM IS ORDER OF NUMERATOR                  *
*                   XMUD IS ARRAY OF DENOMINATOR STEP SIZE FACTORS *
*                   XMUN IS ARRAY OF NUMERATOR STEP SIZE FACTORS *
*                   ALPHAF ARE COEFFICENTS OF FIXED PART OF DENOM. *
*                   BETA ARE NUMERATOR COEFFICIENTS             *
*                   IDAD IS ORDER OF UNKNOWN PART OF DENOM.     *
*                   ALPHAI ARE THE COEFFICENTS OF INCREMENTAL PART *
*                   Y IS OUTPUT MEASUREMENT                     *
*                   US IS CONTROL INPUT                         *
*                   NU AND NY ARE DIMENSIONS OF U & N           *
*                   UNORM AND YNORM ARE NORMALIZING FACTORS     *
*                   XS AND US ARE STORAGE ARRAYS                *
*                   IOUT=1 FOR OUTPUT                           *
*                                                               *
*             OUTPUTS:                                          *
*                                                               *
*                   ALPHAI & BETA ARE UPDATED EACH TIME STEP    *
*                   XS & US ARE UPDATED EACH TIME STEP          *
*                   YP IS THE PREDICTED OUTPUT AT CURRENT TIME  *
*                   ERR IS THE ERROR OF PREDICTION              *
*****************************************************************
      REAL ALPHAF(1),BETA(2,2,2),Y(1),U(1),XS(2,2),US(2,2)
      REAL YP(1),ERR(1),ALPHAI(1),XMUD(1),XMUN(1)
      INTEGER NY(1),NU(1)
      IMEAS=NY(1)
      ICTRL=NU(1)
      NEWD=IDEN+IDAD
      IF(IOUT.EQ.1) WRITE(6,410) IDEN,IMEAS,ICTRL
C     IDELAY = 1
C
C     COMPUTE PREDICTION
C
      DO 200 L=1,IMEAS
      XSTO1=0.
      XSTO2=0.
      XSTO3=0.
      USTO1=0.
```

```
      DO 190 I=1,IDEN
      XSTO1=XSTO1+ALPHAF(I)*XS(I,L)
      DO 190 K=1,IDAD
      XSTO3=XSTO3-ALPHAF(I)*ALPHAI(K)*XS(I+K,L)
  190 CONTINUE
      DO 192 K=1,IDAD
      XSTO2=XSTO2+ALPHAI(K)*XS(K,L)
  192 CONTINUE
      DO 195 I=1,INUM
      DO 195 J=1,ICTRL
      USTO1=USTO1+BETA(I,J,L)*US(I,J)
  195 CONTINUE
      YP(L)=USTO1+XSTO1+XSTO2+XSTO3
  200 CONTINUE
C
C     COMPUTE ERR & NEW WTS
C
      DO 250 L=1,IMEAS
      ERR(L)=Y(L)-YP(L)
  250 CONTINUE
C
C      COMPUTE NEW PARAMETERS
C
      IF(XS(NEWD,1).EQ.0) GOTO 309
      DO 300 K=1,IDAD
      DO 300 L=1,IMEAS
      T1=0.
      F=2.*XMUD(K)*ERR(L)
      DO 299 I=1,IDEN
      T1=T1-ALPHAF(I)*XS(I+K,L)
  299 CONTINUE
      UPDATE=F*(XS(K,L)+T1)
      ALPHAI(K)=ALPHAI(K)+UPDATE
      IF(IOUT.EQ.0) GOTO 300
      WRITE(6,40) K,ALPHAI(K),XS(K,L),T1,UPDATE
   40 FORMAT(10X"K,ALPHAI,XS,T1,UPDATE== ",I5,4F15.6)
  300 CONTINUE
      IF(IOUT.EQ.0) GOTO 301
      WRITE(6,42) YP(1),Y(1),ERR(1),XSTO1,XSTO2,XSTO3,USTO1
   42 FORMAT(10X"YP,Y,ERR="3F12.6,5X,"XSTO(1,2,3)=",3F12.6,
     1       5X,"USTO=",F12.6)
C
C
  301 DO 305 J=1,ICTRL
      DO 305 I=1,INUM
      DO 305 L=1,IMEAS
      F=2.*XMUN(I)*ERR(L)
      BETA(I,J,L)=BETA(I,J,L)+F*US(I,J)
  305 CONTINUE
C
C     UPDATE MEAS & CTRL STATES
C
  309 DO 320 L=1,IMEAS
```

```
      DO 310 II=2,NEWD,1
      I=NEWD+2-II
      XS(I,L)=XS(I-1,L)
  310 CONTINUE
      XS(1,L)=YP(L)/YNORM
  320 CONTINUE
      DO 330 J=1,ICTRL
      IF(INUM.LT.2) GOTO 326
      DO 325 II=2,INUM,1
      I=INUM+2-II
      US(I,J)=US(I-1,J)
  325 CONTINUE
  326 US(1,J)=U(J)/UNORM
  330 CONTINUE
C
C       DEBUGGING OUTPUT
C
      IF(IOUT.EQ.0) GO TO 500
  410 FORMAT(/5X"     LMS ALGORITHM OUTPUT     "/10X,"SYSTEM ORDER= "
     1 ,I5/10X,"NO. OF MEAS.= ",I5/10X,"NO. OF CTRLS.= ",I5)
      WRITE(6,415)(ALPHAI(I),I=1,IDAD)
  415 FORMAT(10X,"ALPHAI:"5(T20,5F12.5/))
      WRITE(6,420)
  420 FORMAT(10X,"BETA:")
      DO 430 J=1,ICTRL
      DO 430 L=1,IMEAS
      WRITE(6,425)J,L,(BETA(I,J,L),I=1,INUM)
  425 FORMAT(10X,"CTRL= ",I5,5X,"IMEAS= ",I5,5(T50,5F12.5,))
  430 CONTINUE
      WRITE(6,440)
  440 FORMAT(10X,"STATES:")
      DO 450 L=1,IMEAS
      WRITE(6,445) L,(XS(I,L),I=1,NEWD)
  445 FORMAT(10X,"MEAS= ",I5,5(T30,5F12.5,))
  450 CONTINUE
      WRITE(6,455)
  455 FORMAT(10X,"CONTROL STATES:")
      DO 460 J=1,ICTRL
      WRITE(6,458) J,(US(I,J),I=1,INUM)
  458 FORMAT(10X,"CONTROL= ",I5,5(T30,5F12.5,))
  460 CONTINUE
  500 RETURN
      END
```

REFERENCES

1. Eveleigh, V. W.: Adaptive Control and Optimization Techniques, McGraw-Hill Book Co., New York, 1967.

2. Mishkin, E.; and Braun, L. (eds.): Adaptive Control Systems, McGraw Hill Book Co., New York, 1961.

3. Friedland, Bernard: "On Adaptive Control," Proceedings of Workshop on Adaptive Control, Champaign, IL, May 1979, pp. 154-156.

4. Aseltine, J. A.; Mancini, A. R.; and Sarture, C. W.: "A Survey of Adaptive Control," I.R.E. Transactions on Automatic Control, December 1958, pp. 102-108.

5. Ehlers, H. L.; and Smyth, R. K.: "Adaptive Control Applications to Aerospace Vehicles," Journal of Aircraft, Vol. 7, No. 2, March-April 1970, pp. 97-116.

6. Kreisselmeir, Gerhard: "Some Aspects of Insensitive and Adaptive Control," Proceedings of Workshop on Adaptive Control, Champaign, IL, May 1979, pp. 154-156.

7. Kreisselmeir, Gerhard: "Algebraic Separation in Realizing a Linear State Feedback Control Law by Means of an Adaptive Observer," IEEE Transactions on Automatic Control, Vol. AC-25, August 1980.

8. Kreisselmeir, Gerhard: "Adaptive Control via Adaptive Observation and Assymptotic Feedback Matrix Synthesis," IEEE Transactions on Automatic Control, Vol. AC-25, August 1980, pp. 717-722.

9. Landau, I.D.: Adaptive Control, The Model Reference Approach, Dekker Publishing Co., New York, 1979.

10. Astrom, K. J.; and Wittenmark, B.: "A Self-Tuning Regulator," Automatica, No. 8, 1973, pp. 185-199.

11. Astrom, K. J.; Borisson, V.; Ljung, L.; and Wittenmark, B.: "Theory and Applications of Self-Tuning Regulators," Automatica, Vol 13, 1977.

12. Athans, M., et al: Investigation of the Multiple Model Adaptive Control (MMAC) Method for Flight Control Systems. NASA Contractor Report 3089, May 1979.

13. Athans, M., et al: "The Stochastic Control of the F8-C Aircraft Using the Multiple Model Adaptive Control (MMAC) Method," IEEE Transactions on Automatic Control, Vol. AC-22, Oct 1977.

14. Tse, E.; Bar-Shalom, Y.; and Meir, L.: "Wide Sense Adaptive Dual Control of Stochastic Non-Linear Systems," IEEE Transactions on Automatic Control, Vol. AC-19, 1974, pp. 494-501.

15. Tse, E.; and Bar-Shalom, Y.: "Actively Adaptive Control for Discrete Time Systems with Random Parameters Via the Dual Control Approach," IEEE Transactions on Automatic Control, Vol. AC-18, 1973, pp. 109-117.

16. Sevald, A. V.: "A Computationally Efficient Optimal Solution to the LQG Discrete Time Dual Control Algorithm," Proceedings of the 17th IEEE Conference on Decison and Control, San Diego, CA, January 1979.

17. Rohrs, C.; Valavani, L.; and Athans, M.: "Convergence Studies of Adaptive Control Algorithms, Part I: Analysis," Proceedings of the 18th IEEE Conference on Decison and Control, Albuquerque, NM, 1980, pp. 1138-1141.

18. Rohrs, C.; Valavani, L.; and Athans, M.: "Analytical Verification of Undesirable Properties of Direct Model Reference Adaptive Control Algorithms," LIDS-P-1122, MIT, Cambridge, MA, August 1981, also Proceedings of the 20th IEEE Conference on Decision and Control, San Diego, CA, December 1981.

19. Rohrs, C.: Adaptive Control in the Presence of Unmodeled Dynamics, PhD Dissertation, MIT, LIDS-TH-1254, November 1982.

20. Rohrs, C.; Valavani, L.; Athans, M.; and Stein, G.: "Robustness of Adaptive Control Algorithms in the Presence of Unmodeled Dynamics," Proceedings of the 21st IEEE Conference on Decision and Control, Orlando, FL, December 1982.

21. Franklin, G. F.; and Powell, J. D.: Digital Control of Dynamic Systems, Addison-Wesley Publishing Co., Menlo Park, CA, 1980.

22. Bryson, A. E.; and Ho, Y. C.: Applied Optimal Control, John Wiley and Sons, New York, 1975.

23. Mischne, D.: "On Line Parameter Estimation Using a High Sensitivity Estimator," PhD Dissertation, Department of Aeronautics and Astronautics, Stanford University, Stanford, CA, 1978.

24. Mischne, D.; and Bryson, A. E.: "On Line Parameter Estimation Using a High Sensitivity Estimator," AIAA Paper No. 78-1303, 1978.

25. Likens, P.: "The Application of Multivariable Control Theory to Spacecraft Attitude Control," Proceedings of IFAC Multivariable Technological Symposium, Frederiction, N.B., Canada, 1977, pp. 11-20.

26. Gupta, N. K, et al: "Modeling, Control and System Identification Methods for Flexible Structures," Agardograph on Spacecraft Pointing and Position Control, AGARD-AG-260, 1982.

27. Aubrun, J. N.: "Theory of the Control of Structures by Low-Authority Controllers," AIAA Journal of Guidance and Control, Vol. 3, No. 5, 1980, p. 494.

28. Gupta, N. K.: "Linear Parameter Insensitive Controllers," Systems Control Technical Memo 5203-01, Palo Alto, CA, August 1977.

29. Fogel, E., et al: "Preliminary Flight Control Avionic Requirements for Orbiter-Attached Large Space Structures," Charles Stark Draper Laboratory, CSDL-R-1531, Cambridge, MA, January 1982.

30. Widrow, B.; and Hoff, M. E.: "Adaptive Switching Circuits," 1960 WESCON Convention Record, Part 4, 1960, pp. 96-140.

31. Widrow, B., et al: "Stationary and Nonstationary Learning Characteritics of the LMS Adaptive Filter," Proceedings of the IEEE, Vol. 64, No. 8, August 1976.

32. Widrow, B.; and McCool, J. M.: "A Comparison of Adaptive Algorithms Based on the Methods of Steepest Descent and Random Search," IEEE Transactions on Antennas and Propagation, Vol. AP-24, No. 5, September 1976.

33. Bitmead, R. R.; and Anderson, B. D. O.: "Performance of Adaptive Estimation Algorithms in Dependent Random Environments," IEEE Transactions on Automatic Control, Vol. AC-25, No. 4, August 1980.

34. Johnson, C. R., et al: "Simple Adaptive Output Error Identification," Proceedings of the 10th Southeastern Symposium on Systems Theory, Part IIIA-1, 1978.

35. Johnson, C. R., et al: "Remarks On Use of SHARF as an Output Error Identifier," Proceedings of the IEEE Conference on Decision and Control, 1979, p. 1094.

36. Johnson, C. R.: "An Output Error Identification Interpretation of Model Reference Adaptive Control," Automatica, Vol. 16, 1980, pp. 419-421.

37. Johnson, C. R.: "A Convergence Proof for a Hyperstable Adaptive Recursive Filter," IEEE Transactions on Information Theory, Vol. IT-25, November 1979, pp. 745-749.

38. Lin, Y.-H.; and Narendra, K. S.: "A New Error Model for Adaptive Systems," IEEE Transactions on Automatic Control, Vol.AC-25, June 1980, pp. 585-587.

39. Johnson, C. R.; and Anderson, B. D. O.: "Sufficient Excitation and Stable Reduced-Order Adaptive IIR Filtering," IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-29, Vol. 6, December 1981.

40. Anderson, B. D. O.; and Johnson, C. R.: "On Reduced-Order Adaptive Output Error Identification and Adaptive IIR Filtering," IEEE Transactions on Automatic Control, Vol. AC-27, No. 4, August 1982.

41. Widrow, B.; and Stearns, S.: "An Introduction to Adaptive Signal Processing," Class Notes to be published, Department of Electrical Engineering, Stanford University, Stanford, CA, 1981.

42. Johnson, C. R.: "The Common Parameter Estimation Basis of Adaptive Filtering, Identification and Control," Proceedings of the 19th IEEE Conference on Decision and Control, Albuquerque, NM, December 1980.

43. Reid, D. B.: "Optimal Inputs for System Identification," PhD Dissertation, Department of Aeronautics and Astronautics, SUDAAR No. 440, Stanford University, Stanford, CA, May 1972.

44. Chen, R. J. N.: "Input Design for Parameter Identification--Part I," Proceedings of the JACC, Austin, TX, 1974.

45. Mehra, R. K.: "Optimal Input Signals for Parameter Estimation in Dynamic Systems--Survey and New Results," IEEE Transactions on Automatic Control, Vol. AC-19, December 1974, pp. 753-768.

46. Taylor, L. W.; and Iliff, K. W.: "Systems Identification Using a Modified Newton-Raphson Method--A Fortran Program," NASA TN D-6734, May 1972.

47. Ioannonv, P. A.; and Kokotovic, P. V.: "An Asymptotic Error Analysis of Identifiers and Adaptive Observers in the Presence of Parasitics," IEEE Transactions on Automatic Control, Vol. AC-27, No. 4, August 1982, pp. 921-927.

48. Bitmead, R. R.: "Convergence in Distribution of LMS-Type Adaptive Parameter Estimators," IEEE Transactions on Automatic Control, Vol. AC-28, No. 1, January 1983.

49. Bitmead, R. T.; and Anderson, B. D. O.: "Exponentially Convergent Behavior of Simple Stochastic Estimation Algorithms," Proceedings of the 17th IEEE Conference on Decision and Control, San Diego, CA, 1979.

50. Bitmead, R. R.: "Convergence Properties of LMS Adaptive Estimators with Unbounded Dependent Inputs," Proceedings of the 20th IEEE Conference on Decision and Control, San Diego, CA, 1981, pp. 607-612.

51. Armstrong, E. S.: ORACLS--A Design System for Linear Multivariable Control, Marcel Dekker, Inc., New York, 1980.

52. Kailath, T.,: Linear Systems, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1980.

53. Sain, M. K.: "A Free-Modular Algorithm for Minimal Design of Linear Multivariable Systems," Proceedings of the 6th Triennial IFAC World Congress, Section 9.1, August 24-30, 1975.

54. Chen, C. T.: Introduction to Linear System Theory, Holt, Rinehart, and Winston, Inc., New York, 1970, pp. 216-258.

55. Raven, E. A.: "A Minimal Realization Method," IEEE Control Systems Magazine, September 1981, pp. 14-20.

56. Davidon, William C.: "Variable Metric Method for Minimization," AEC Research and Development Report ANL-5990 Rev. (Contract W-31-109-eng-38), Argonne National Laboratory, November 1959.

57. Fletcher, R.; and Powell, M. J. D.: "A Rapidly Convergent Descent Method for Minimization," Computer Journal, Vol. 6, No. 2, July 1963, pp. 163-168.

58. Park, S. K.: "A Transformation Method for Constrained-Function Minimization," NASA TN D-7983, November 1975.

59. Martin, W. A.; and Fateman, R. J.: "The MACSYMA System," ACM Proceedings of the Second Symposium on Symbolic and Algebraic Manipulation, Los Angeles, CA, March 1972.

60. Bogen, Richard, et al: MACSYMA Reference Manual, Version 9, The Mathlab Group, MIT, Cambridge, MA, December 1977.

61. Johnson, C. D.: "State Variable Design Methods May Produce Unstable Feedback Controllers," International Journal of Control, Vol. 29, No. 4, 1979, pp. 607-619.

62. Bryson, A. E.: "Some Connections Between Modern and Classical Control Concepts," Journal of Dynamic Systems, Measurements and Control, Vol. 101, June 1979, pp. 91-98.

63. Kwakernaak, H.; and Sivan, R.: Linear Optimal Control Systems, John Wiley and Sons, Inc., New York, 1972.

64. Bryson, A. E.: Stabilization and Control of Flight Vehicles, Class notes to be published, Department of Aeronautics and Astronautics, Stanford University, 1982.

65. Wilsky, A. S.: "A Survey of Design Methods for Failure Detection in Dynamic Systems, Automatica, Vol. 12, 1976, pp. 601-611.

66. Desai, M.; and Ray, A.: "A Fault Detection and Isolation Methodology," Proceedings of the 20th Conference on Decision and Control, December 1981, pp. 1363-1369.

67. Wie, Bong: "On the Modeling and Control of Flexible Space Structures," SUDAAR 525, Department of Aeronautics and Astronautics, Stanford University, Stanford, CA, June 1981.

68. Yocum, J. F.; and Slafer, L. J.: "Control System Design in the Presence of Severe Structural Dynamics Interactions," Journal of Guidance and Control, Vol. 1, No. 2, March-April 1978, pp. 109-116.

69. Slafer, L. I.: "On-Orbit Evaluation of the Control System and Structural Mode Interactions on OSO-8," Journal of Guidance and Control, Vol. 3, No. 3, May-June 1980, pp. 203-209.

70. Martin, G. D.: "On the Control of Flexible Mechanical Systems," Stanford University, SUDAAR 511, Department of Aeronautics and Astronautics, Stanford University, May 1978.

71. Martin, G. D.; and Bryson, A. E.: "Attitude Control of a Flexible Spacecraft," Journal of Guidance and Control, Vol. 3, No. 1, January-February 1980, pp. 37-41.

72. Balas, M. J.: "Trends in Large Space Structures Control Theory: Fondest Hopes, Wildest Dreams," IEEE Transactions on Automatic Control, Vol. AC-27, No. 3, June 1982.

73. Williams, J. P.; and Montgomery, R. C.: "Simulation and Testing of Digital Control on a Flexible Beam," Proceedings of the AIAA Guidance and Control Conference, August 1982, pp.403-409.

74. Friedlander, B.: "Lattice Filters for Adaptive Processing," Proceedings of the IEEE, Vol. 70, No. 8, August 1982, pp.829-267.

75. Sundararajan, N.; and Montgomery, R. C.: "Decoupling the Structural Modes Estimated Using Recursive Lattice Filters," Proceedings of the 21st IEEE Conference on Decision and Control, December 1982, pp. 998-999.

76. Sundararajan, N.; and Montgomery, R. C.: "Adaptive Identification for the Dynamics of Large Space Structures," Proceedings of the 1982 AIAA Guidance and Control Conference, AIAA Paper No. 82-1565, August 1982, pp. 379-387.

77. ACSL -- Advance Continuous Simulation Language - User Guide and Reference Manual, Mitchell and Gauthier Assoc., Inc., P. O. Box 685, Concord, MA, 1975.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-85737		

4. Title and Subtitle	5. Report Date
An On-Line Equivalent System Identification Scheme for Adaptive Control	January 1984
	6. Performing Organization Code: 505-34-03-05
7. Author(s): Steven M. Sliwa	8. Performing Organization Report No.
	10. Work Unit No.
9. Performing Organization Name and Address: NASA Langley Research Center, Hampton, VA 23665	11. Contract or Grant No.
	13. Type of Report and Period Covered: Technical Memorandum
12. Sponsoring Agency Name and Address: National Aeronautics and Space Administration, Washington, DC 20546	14. Sponsoring Agency Code

15. Supplementary Notes

This report was submitted to the Department of Aeronautics and Astronautics and the Committee on Graduate Studies of Stanford University in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

16. Abstract

A prime obstacle to the widespread use of adaptive control is the degradation of performance and possible instability resulting from the presence of unmodeled dynamics. The approach taken is to explicitly include the unstructured model uncertainty in the output error identification algorithm. The order of the compensator is successively increased by including newly identified modes. During this model building stage, heuristic rules are used to test for convergence prior to designing new compensators. Additionally, the recursive identification algorithm has been extended to multi-input, multi-output systems. Enhancements were also made to reduce the computational burden of an algorithm for obtaining minimal state space realizations from the inexact, multivariable transfer functions which result from the identification process. A number of potential adaptive control applications for this approach are illustrated using computer simulations. Results indicated that when speed of adaptation and plant stability are not critical, the proposed schemes converge to enhance system performance.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
adaptive control, identification, estimation, model identification, learning systems, heuristics	Unclassified - Unlimited Subject Category 08

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	177	A09

For sale by the National Technical Information Service, Springfield, Virginia 22161