

STOCHASTIC ADAPTIVE ESTIMATION WITH APPLICATIONS

TO NONLINEAR CONTROL

by

Philip Edward Zwicke

Dissertation submitted to the Graduate Faculty of the  
Virginia Polytechnic Institute and State University  
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Electrical Engineering

APPROVED:

*R. L. Moose*

R. L. Moose

*H. F. VanLandingham*

H. F. VanLandingham

*A. W. Bennett*

A. W. Bennett

*C. Richard Johnson*

C. R. Johnson

*F. H. Tutze*

F. H. Tutze

May 1978

Blacksburg, Virginia

LD  
5655  
Y856  
1978  
295  
c.2

## ACKNOWLEDGMENTS

The author wishes to express a sincere thank you to those persons who have provided the intellectual and emotional support that has made this dissertation possible. The most appreciation is directed to my wife Onolee whose unselfish encouragement, love and friendship have been crucial.

A special thanks to the fine individuals who served as members of my committee: Dr. Richard L. Moose, my major professor, who provided the motivation and able direction of my program; Dr. H. F. VanLandingham, whose many valuable and enthusiastic suggestions have added greatly to my research; Dr. C. R. Johnson, whose thorough readings of this dissertation are greatly appreciated; Dr. F. H. Lutze, whose expertise in aerodynamics and aircraft control have been very helpful; Dr. A. W. Bennett, who provided key assistance and understanding throughout my graduate program.

It is a pleasure to acknowledge the people at NASA LaRC, who have donated a considerable amount of their time in supplying information and answering questions concerning the B737 project that has supported my research. Dr. Jerry Creedon, Dr. Thomas M. Walsh, Mr. R. M. Hueschen and the rest of the people in the Flight Instrumentation division have been indispensable. In addition, appreciation is directed to Dr. E. S. Armstrong, whose excellent computational routines were used extensively in the execution of the algorithms in this dissertation.

The author is most proud and grateful to have been associated with the Electrical Engineering Department of Virginia Polytechnic Institute and State University. Dr. W. A. Blackwell, Professor G. R. Powley and

the rest of the faculty have contributed to a meaningful and enjoyable educational experience. The author wishes to express a sincere thanks to Dr. L. L. Grigsby, whose friendship and character have been greatly valued both personally and professionally throughout my academic years.

Finally, a big thanks to Cheryl West who endured the trying circumstances of typing this dissertation, always offering encouragement and understanding.

## TABLE OF CONTENTS

	<u>Page</u>
ACKNOWLEDGMENTS. . . . .	ii
TABLE OF CONTENTS. . . . .	iv
LIST OF FIGURES. . . . .	vii
LIST OF TABLES . . . . .	xi
1.0 INTRODUCTION. . . . .	1
1.1 State of the Art . . . . .	1
1.2 New Results. . . . .	2
1.3 Related Investigations and Applications. . . . .	3
2.0 FUNDAMENTAL BACKGROUND. . . . .	13
2.1 Introduction . . . . .	13
2.2 The Basic Control Problem. . . . .	13
2.3 State Variable Feedback. . . . .	14
2.4 Optimal Control Formulation. . . . .	17
2.5 The Structure of the Kalman Filter . . . . .	18
3.0 THE MODIFIED PARTITIONED ADAPTIVE ESTIMATOR . . . . .	25
3.1 Introduction . . . . .	25
3.2 The Partitioned Adaptive Estimator with Semi-Markov Plant Modeling . . . . .	28 41
3.3 The Modified PAE Algorithm . . . . .	49
3.4 Nonlinear Estimation Using the Modified PAE Algorithm. . . . .	59
3.5 Summary. . . . .	64
4.0 STOCHASTIC NONLINEAR CONTROL. . . . .	64
4.1 Introduction . . . . .	67

TABLE OF CONTENTS, cont.	<u>Page</u>
4.2 The Modified PAC Controller. . . . .	67
4.3 Set-Point Control. . . . .	70
4.4 Simulations. . . . .	75
4.5 Summary. . . . .	88
5.0 THE DETECTION OF PLANT CONFIGURATION CHANGES. . . . .	90
5.1 Introduction . . . . .	90
5.2 The Optimal Estimator for the Switches Linear Plant. . .	91
5.3 Sliding Window Detector/Estimator, SWDE. . . . .	98
5.4 Implementation . . . . .	103
5.5 Weighted Sum Estimation. . . . .	111
5.6 Initialization Algorithm . . . . .	112
5.7 Optimal Delayed Estimation . . . . .	114
5.8 Simulations. . . . .	114
5.9 Summary. . . . .	137
6.0 ADAPTIVE CONTROL OF THE B737 AIRCRAFT . . . . .	138
6.1 Introduction . . . . .	138
6.2 The Equations of Motion. . . . .	139
6.3 The Need for Adaptive Control. . . . .	150
6.4 Closed-Loop Design Via Model Following . . . . .	152
6.5 Nonlinear Measurement Model. . . . .	162
6.6 B737 Simulation and Results. . . . .	170
6.7 Summary. . . . .	172
7.0 CONCLUSION. . . . .	177
BIBLIOGRAPHY . . . . .	181
APPENDIX A - SWDE ALGORITHM. . . . .	186

TABLE OF CONTENTS, cont.	<u>Page</u>
A-1 Flowchart of the Basic Algorithm. . . . .	186
A-2 Initialization and Re-Start Algorithm . . . . .	189
A-3 Weighted-Sum Estimation Modification. . . . .	189
APPENDIX B - THE "TCVA-737 TRIM DATA VALIDATION" PROGRAM. . . . .	193
B-1 Description . . . . .	193
B-2 Use . . . . .	193
B-3 Output. . . . .	194
B-4 Example . . . . .	194
APPENDIX C - SUMMARY OF THE MODEL-FOLLOWING PROBLEM . . . . .	197
C-1 Continuous Model-Following Problem. . . . .	197
C-2 Discrete Model-Following Problem. . . . .	200
VITA. . . . .	202
ABSTRACT	

## LIST OF FIGURES

<u>Figure</u>	<u>Title</u>	<u>Page</u>
1.3.1	Approaches and Investigators of the Stochastic Adaptive Estimation Problem. . . . .	7
2.5.1	Kalman Filter . . . . .	21
3.2.1	The Partitioned Adaptive Estimator Structure. . . . .	29
3.2.2	Modified Partitioned Adaptive Estimator, Showing Semi-Markov Addition . . . . .	34
3.3.1	The Tracking Performance of the PAE Algorithm with Semi-Markov Addition but Without Re-Initialization . . . . .	46
3.3.2	Tracking Performance of the Modified PAE Algorithm, i.e. with Re-Initialization . . . . .	50
3.4.1	Variation in System Eigenvalues for the Design Example	55
3.4.2	The Three Linear Approximations for the Cubic Nonlinearity . . . . .	57
3.4.3	Actual Nonlinear Plant Position and the Modified PAE Position Estimate . . . . .	60
4.2.1	Modified Partitioned Adaptive Controller. . . . .	69
4.4.1	Open-Loop Response to A Set-Point of 4 Units. . . . .	77
4.4.2	Closed-Loop Response of Nonlinear Oscillator to A Set-Point of 4 Units. Also shown are the three Kalman filter estimates . . . . .	79
4.4.3	Closed-Loop Step Response with Varying Amounts of Measurement Noise . . . . .	80
4.4.4	Position Response to Set-Point Changes. . . . .	81

LIST OF FIGURES, cont.

<u>Figure</u>	<u>Title</u>	<u>Page</u>
4.4.5	Set-Point Control for $u = 2.0$ Using Different Approximations in Region $s_2$ . . . . .	83
4.4.6	Set-Point Outside the Design Range . . . . .	84
4.4.7	Set-Point Control for Extended Design Range. . . . .	86
4.4.8	Set-Point Control for $\theta = \{\text{diag. } .8, \text{ off-diag. } .1\}$ . . . . .	87
5.2.1	A Tree Having 31 Branches. . . . .	94
5.3.1	The Decision Regions, $H_0$ , $H_2$ and $H_3$ . . . . .	102
5.3.2	Estimator Operation Over Several Configuration Changes . . . . .	104
5.4.1	Branch Divergence. . . . .	109
5.8.1	Plot of Actual Position and Position Estimate of the SWDE Algorithm . . . . .	117
5.8.2	Plot of Actual Velocity Versus Estimate of the SWDE Algorithm in Example 5-1 . . . . .	118
5.8.3	Delayed Estimate Compared to the Actual Plant Position in Example 5-1 . . . . .	120
5.8.4	Comparison of SWDE Estimates for Example 5-1 . . . . .	121
5.8.5	Estimation Performance of the PAE Algorithm of Chapter 3 . . . . .	124
5.8.6	Relative Probabilities and Measurement Residual Density Values of a Tree Having Window Width, $N = 5$ . . . . .	125
5.8.7	Comparison of Estimates After A Sequence of Configuration Changes. . . . .	128
5.8.8	Performance Improvement by Using the Initialization Algorithm to Detect Divergence . . . . .	129

LIST OF FIGURES, cont.

<u>Figure</u>	<u>Title</u>	<u>Page</u>
5.8.9	Comparison of the Weighted Sum Estimate of the SWDE Algorithm and the Estimate of the PAE Algorithm for the Position of the Plant in Example 5-2. . . . .	132
5.8.10	Comparison of SWDE Estimates for Example 5-2. . . . .	133
5.8.11	Position Estimates for the Nonlinear Plant of Example 5-3 . . . . .	135
5.8.12	Nonlinear Estimation Using the SWDE Algorithm . . . . .	136
6.2.1	Example of State Equation Matrices. . . . .	144
6.2.2	Ten Flight Conditions Used in Simulation. . . . .	145
6.2.3	Augmented System Matrices . . . . .	147
6.2.4	Definition of Coordinate Axes, Angles and Forces. . . . .	148
6.3.1	Open-Loop Response to A 10-Degree Elevator Step Input . . . . .	151
6.3.2	Angle of Attack, $\alpha$ , Response for Different Flight Conditions. . . . .	153
6.3.3	Closed-Loop Pitch Rate Response in Various Regions Using Feedback Gains Derived for Region 1 . . . . .	154
6.3.4	Closed-Loop Response Using Feedback Gains Derived for Region 10 . . . . .	155
6.3.5	Closed-Loop Response Using Feedback Gains Derived for Region 5. . . . .	156
6.3.6	Closed-Loop Elevator Response Required to Capture a 20 ft/sec Sink Rate Glideslope in the Different Regions . . . . .	157
6.4.1	Block Diagram of Model-Following System . . . . .	159
6.4.2	Gain Matrix for Configuration $s_1$ . . . . .	163

LIST OF FIGURES, cont.

<u>Figure</u>	<u>Title</u>	<u>Page</u>
6.5.1	Prefiltered Measurement Coefficient Matrix, H. . . . .	168
6.5.2	Prefiltered Noise Covariance Matrix, R . . . . .	169
6.6.1	Glideslope Simulation Parameters . . . . .	171
6.6.2	Actual Versus Desired Glideslope, x-position . . . . .	173
6.6.3	Actual Versus Desired Glideslope, z-position . . . . .	174
6.6.4	Weighting Coefficients for Glideslope Following. . . . .	175
A.1.1	SWDE Algorithm Flowchart . . . . .	187
A.1.2	DETECT Flowchart . . . . .	188
A.1.3	PROPAGATE Flowchart. . . . .	191
A.2.1	TRACK Algorithm Flowchart. . . . .	192

LIST OF TABLES

<u>Table</u>	<u>Title</u>	<u>Page</u>
3.2.1	Time-Varying Weighting Coefficients for Different Semi-Markov Matrices. . . . .	38
3.2.2	Time-Varying Weighting Coefficients for Different Semi- Markov Matrices and Different $p(\tilde{z})$ . . . . .	40
3.3.1	Weighting Coefficients for $\theta = \{\text{diag. } 996, \text{ off-diag. } .002\}$	47
3.3.2	Weighting Coefficients for $\theta = \{\text{diag. } 333, \text{ off-diag. } .333\}$	48
3.3.3	Weighting Coefficients for Figure 3.3.2 . . . . .	51
3.4.1	Weighting Coefficients for Figure 3.4.3 . . . . .	61
4.3.1	Design Parameters for the Set-Point Control Example . . .	76
5.8.1	Comparison of Actual Position with the Optimum Estimate and the SWDE Delayed Estimate . . . . .	122
6.3.1	Longitudinal Eigenvalue Variation for the Ten Flight Conditions. . . . .	149
6.5.1	Actual Measurement Noise Characteristics. . . . .	165
B.4.1	Flight Conditions Used in the Trim Data Validation Program . . . . .	195
B.4.2	The Ten Configurations Used in Chapter 6. . . . .	196

## 1.0 INTRODUCTION

### 1.1 State of the Art

Although the vast majority of physical systems are nonlinear, there exists no general method of analysis for such problems. Moreover, it is "the opinion of the great majority of workers in the field, that a general method of synthesis for nonlinear problems is impossible" [G-1]. Many techniques have been devised to handle particular nonlinear problems or classes of problems; however, they usually suffer from being too specific and too approximate.

Indeed, the control of a nonlinear plant in a noisy environment is an extremely complex problem. The powerful techniques of modern linear control theory do not generally apply to such problems. Traditionally, a unique controller is designed for each specific problem; hence a considerable amount of engineering is repeated each time a new problem arises.

The existing techniques of nonlinear control can be lumped loosely into two methodologies. First is the linearization approach. The nonlinear equations are expanded about some operating point via a Taylor series expansion. Linear control theory is then applied as though the approximate linear equations were exact. Linearization techniques are especially well-suited in stability analysis. For example, Oaks and Cook use a linearization method in the design of a controller for a nonlinear oscillator [O-1]. Unfortunately, many nonlinear systems are not amenable to linearizing, i.e. switching functions and other abrupt nonlinearities, tabular functions, and partially unknown or

underspecified functions, to mention a few.

The second general method is the adaptive controller approach. The motivation for this method is that a nonlinear plant can be thought of as a linear plant having unknown, time-varying parameters. The adaptive controller first identifies these parameters and then formulates the control law. A wide variety of adaptive controllers have been developed over the years. Several of the more recent designs will be discussed later in this chapter.

## 1.2 New Results

This dissertation is concerned with the development of an adaptive state estimator that is capable of tracking rapidly changing plant configurations. As will be shown, nearly all nonlinear systems can be approximated by a linear plant having a finite number of possible configurations. These plants are referred to as *Switched Linear Plants*. For such nonlinear systems, the adaptive estimator to be developed will give very good performance and will offer significant implementation advantages over other existing estimators. In addition to its direct application as an estimator, the algorithm will be used in closed loop control applications. In this capacity the actual estimation is of secondary import to the adaptive identification capabilities of the method.

Two adaptive estimators will be examined and applied in this dissertation. The first, a modification of the estimator introduced by Magill in 1965, is extended to nonlinear estimation in Chapter Three, and to nonlinear control in Chapter Four. The second method, developed

in Chapter Five, is based upon joint detection/estimation and approaches the accuracy of the unrealizable optimum adaptive estimator. The advantages and weaknesses of each method are analyzed, and it is found that a very accurate and reliable estimator can be formulated by combining the two methods.

Chapter Two of this dissertation provides a review of modern and classical control theory, and estimation theory as they relate to the discussions and derivations in the later chapters. The "*Partitioned Adaptive Estimator*" is analyzed in Chapter Three and modified for nonlinear estimation. Chapter Four develops an adaptive controller for nonlinear systems. The *Sliding Window Detector/Estimator* is developed in Chapter Five to track a rapidly switching linear plant. Chapter Six applies the previously developed algorithms to the closed loop control of a B737 aircraft in adverse weather conditions. The results and contributions of this dissertation are summarized in Chapter Seven. The remainder of the present chapter is devoted to a review of the various adaptive estimation methods that have been reported in the literature.

### 1.3 Related Investigations and Applications

The problem of estimating the state (position, velocity, etc.) of a system in a noisy environment was formulated and solved independently in the 1940s by Wiener [W-1] and Kolmogorov [K-1]. The estimator, referred to as the Wiener filter, is applicable to linear, continuous time, completely defined systems, and minimizes the mean square, steady state estimation error. Historically, the Wiener filter is widely acknowledged as being the first successful attempt to optimize the pro-

cess of extracting information from noisy measurements. It has been used extensively in a variety of applications.

A second major advance in estimation theory came in 1961 with the work of Kalman and Bucy [K-2]. The well-known Kalman filter is the optimal estimator for linear systems in gaussian noise, in the sense that it minimizes the mean square estimation error at each point in time. The advent of high speed digital computers has led to wide scale usage of the Kalman filter.

In the last decade, optimal filtering theory and its dual, optimal control theory, have been rigorously developed and extended to much broader classes of problems. One such class is that referred to as adaptive systems, which encompasses systems that are capable of altering themselves in one way or another. A maneuvering target may be an example of an adaptive system. An adaptive estimator or adaptive controller has the ability to modify its behavior, depending upon the performance of the actual system. An excellent survey of different stochastic adaptive control methods was done in 1975 by Wittenmark [W-2]. Much of the terminology to follow is taken from that work.

Stochastic adaptive systems are those systems where the variations of the process parameters have been described by stochastic models and have been taken into account in the derivation of the estimation or control algorithm. A stochastic adaptive controller can artificially be divided into two components, an estimator and a controller, the design of which may or may not be carried out independently. If the controller can be designed using variables for the unknown plant parameters, and if the controller remains optimal when the estimated plant parameters

are used in place of the design variables, then the controller is referred to as a "Certainty Equivalence" controller. This condition is satisfied for nonadaptive linear plants in gaussian noise for quadratic cost functionals [W-2]. For adaptive plants, the "Certainty Equivalence" principle is not valid in general but has been successfully applied as an ad hoc suboptimal design method. A somewhat weaker requirement than the "Certainty Equivalence" principle was introduced in 1971 by Witsenhausen and is known as the "Separation Principle" [W-3]. The principle is valid if it is possible to make a separation between the identification of the parameters in the process and the determination of the parameters in the controller. Note that the requirement for optimality has been removed. Controllers designed on this principle are sometimes referred to as being "cautious" because the parameters of the controller are often functions of the uncertainties of the identified plant parameters, as well as the parameters themselves [W-2].

A concept that has received recent attention is that of Dual control [T-1]. The basis of Dual control is that system identification and system control are, in general, conflicting processes. Identification is usually aided by using large variations in plant inputs to cause sizable transients in plant outputs, whereas good control usually requires small input and output fluctuations. By formulating a performance index that takes into account the probability distributions of future observations, a better overall controller can be designed. The drawback to this approach is the extreme complexity of the resulting equations.

For purposes of classification, the methods of stochastic adaptive estimation developed by various investigators over the last decade will

be grouped into four categories. The first category will be referred to as "*Partitioned Adaptive Estimators*" [L-1]. The estimator discussed in Chapter Three is a member of this group. Techniques based upon approximating the unrealizable optimal estimator will comprise the second group. The detector/estimator developed in Chapter Five fits this classification. The third category will be referred to as bandwidth modulation filters. All other methods will be grouped together in the fourth category. The stochastic estimation techniques discussed in the following paragraphs are diagrammed in Figure 1.3.1.

The adaptive estimation problem becomes more tractable when the unknown parameters are constrained to belong to a finite set. A plant identification problem of this type was investigated by Magill in 1965 [M-1]. Magill found that the optimal, minimum mean square error, state estimate is a weighted sum of elemental estimates derived from Kalman filters matched to the possible plant configurations. The weighting coefficients are nonlinear functions of the measurements, and the coefficient corresponding to the actual plant configuration approaches unity while the other coefficients approach zero.

The method of computing the weighting coefficients was improved by Sims and Lainiotis [S-1] in 1969. A Bayesian derivation approach and usage of the measurement residual (rather than the measurement itself) resulted in a simplified recursive algorithm for the coefficients. Hilborn and Lainiotis [H-1] further generalized the algorithm to include vector observations.

Moose and Wang extended Magill's work to the identification of a randomly changing plant [M-2]. The plant configuration changes are

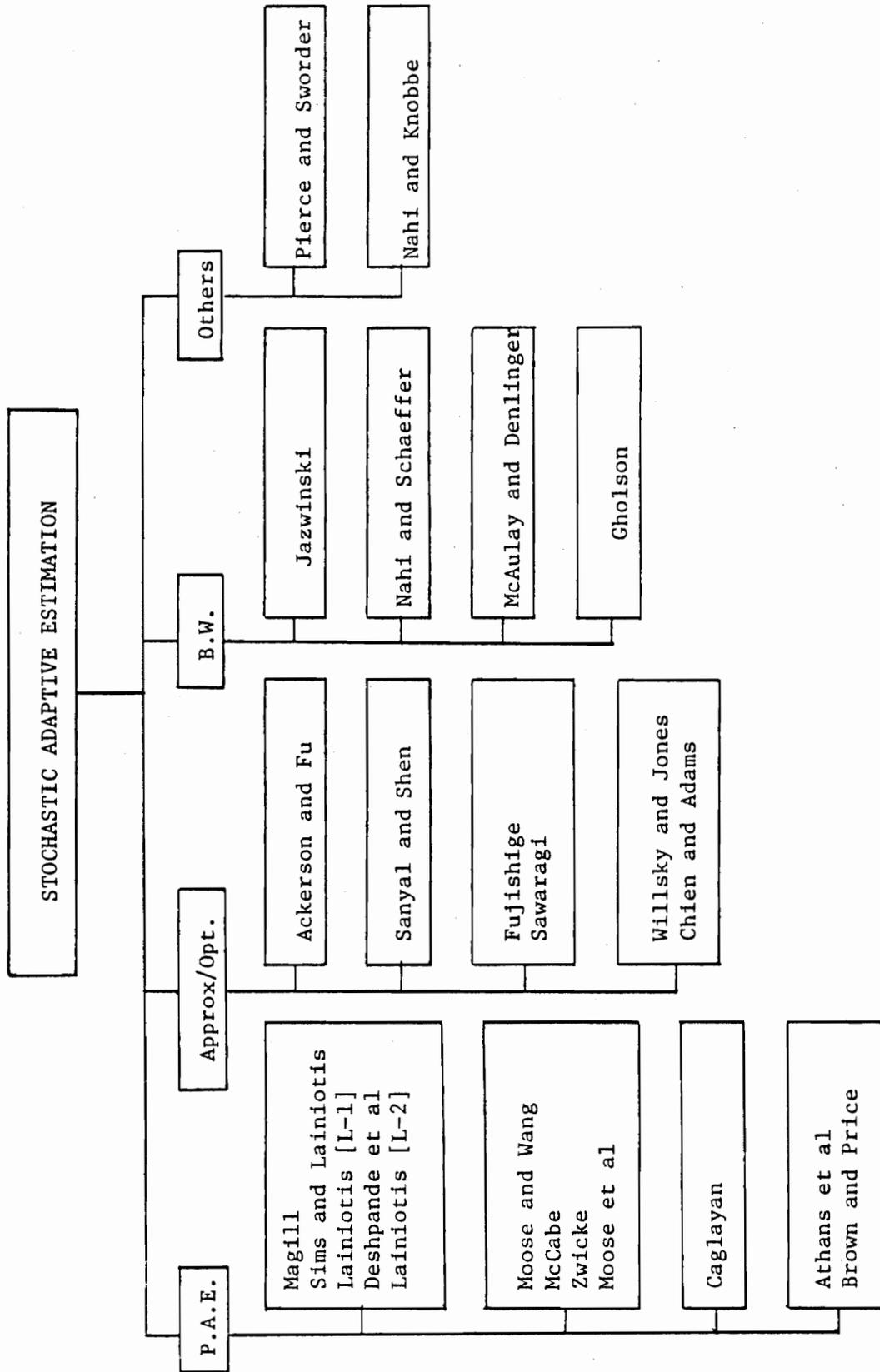


Figure 1.3.1 Approaches and Investigators of the Stochastic Adaptive Estimation Problem

modeled as a semi-Markov process [H-2], the statistics of which are incorporated into a recursive algorithm for the calculation of the weighting coefficients. A modified version of this adaptive estimator is analyzed in detail in Chapter Three. See also[Z-1].

Lainiotis formalized the finite parameter adaptive estimation problem in 1971 by stating and proving the "Partition Theorem" [L-1]. In addition to giving a recursive algorithm for calculating the weighting coefficients, the theorem also gives a formula for calculating the time-varying estimation error covariance matrix.

Deshpande et al, in 1973, motivated the use of the partitioned adaptive estimator in closed loop applications [D-1]. It was shown that for unknown model parameters the separation principle does not generally hold, and the optimal stochastic controller is necessarily a nonlinear controller. The "Partitioned Adaptive Controller" was then derived by minimizing a cost functional. Although suboptimal in general, the controller has given very good performance in a number of different applications [A-1]. Caglayan generalized the partitioning concept to adaptive systems influenced by non-gaussian statistics [C-1]. A summary of the partitioning method for stochastic adaptive estimation and control can be found in Lainiotis [L-2].

Athans et al applied the partitioned adaptive control method to the F-8c aircraft in 1975 as part of a National Aeronautics and Space Administration program to examine modern control techniques [A-1]. The method, referred to as "Multiple Model Adaptive Control" by the investigators, was quite successful in controlling the aircraft in its various flight configurations. Several observations about the method were

pointed out. First, the identification portion of the method is very dependent on the regularity of the measurement residual behavior. Second, the weighting coefficients are not truly a posteriori probabilities. Rather, they should be interpreted as time sequences that have a reasonable physical interpretation. These observations are expanded upon in the following chapters and in reference [M-8].

A comparison of adaptive tracking filters for targets of variable maneuverability was carried out by Brown and Price in 1976 [B-1]. The target was characterized by a "Singer model" [S-2] having a low mode and a high mode of evasive accelerations. They found that the Partitioned adaptive estimator performed poorly because the weighting coefficients could not track the true hypothesis when it changed. This shortcoming was attributed to the weighting coefficients going to zero for those configurations not currently matched to the plant. An ad hoc method of adjusting the weights between measurements according to a Markov switching relationship solved this problem and reliable tracking was achieved.

A finite parameter adaptive plant that undergoes frequent changes in configuration is referred to as a "*Switched Linear Plant*". The inability of the Partitioned Adaptive Estimator to track rapid configuration changes is a major weakness of the method. Ad hoc methods, such as the previously described technique of Brown and Price, must be used to obtain satisfactory performance. These additions are usually specific to one type of problem and often require trial and error tuning of various parameters. An example of this is shown in Chapter Three.

A second approach used in tracking a switched linear plant involves direct simplification of the unrealizable optimal estimator for that

plant. One of the earlier investigations of this type was that of Ackerson and Fu in 1970 [A-2]. Their problem was to estimate the state of a plant in a randomly changing noise environment. The optimal estimator was found to be a weighted sum of an exponentially growing number of elemental estimators. They next developed a suboptimal algorithm by characterizing the state estimate as a gaussian distribution at each iteration rather than the actual weighted sum of gaussian distributions. The mean of the assumed distribution is the weighted sum of the elemental estimates after one iteration. Similarly, the covariance of the distribution is approximated by a weighted sum of the elemental covariances after one iteration. After each iteration the elemental estimators are re-initialized with the above mean and covariance. The net effect is that the overall estimate will be taken predominantly from the elemental estimator currently matched to the plant.

Sanyal and Shen investigated the problem of detecting the time of occurrence of impulses of unknown magnitudes [S-3]. Restricting the problem to only one impulse reduced the estimator complexity from an exponentially growing memory requirement to a linearly growing memory requirement. A Bayesian decision rule was applied to the weighting coefficients at each iteration to detect the impulse. One shortcoming of this technique is that the starting probabilities and Bayesian costs are often unknown.

The optimal estimator for the general switched linear plant problem was rigorously formulated and solved by Fujishige and Sawaragi in 1974 [F-1]. As before, the memory requirement of the estimator was found to increase exponentially with time. No attempt was made to simplify the

estimator. Sawargi et al applied the optimal estimator to the problem of interrupted observations characterized by a Jump-Markov process [S-4]. Their solution, although mathematically rigorous, proved to be quite unwieldy.

Willisky and Jones used a generalized likelihood ratio to detect failures modeled as jumps in state variables [W-4]. A finite "Data Window" was used to keep the memory requirements of the estimator at a manageable level. Upon detection of a failure, the state estimate and the covariance of the estimation error were readjusted. Chien and Adams used a sequential probability ratio test to detect system failures [C-1]. The method was successfully applied to the inertial measurement unit in the space shuttle vehicle.

A less structured approach to adaptive estimation involves increasing the Kalman filter gains, the estimation error covariance or the system disturbance covariance to prevent divergence. These techniques are usually ad hoc in formulation and are often referred to as bandwidth modulation techniques because changing the Kalman gain changes the amount of high frequency noise that will be present in the estimate. For example, a larger Kalman gain makes a filter more responsive to plant changes at the expense of overall noisier performance.

Jazwinski [J-1] prevented divergence due to modeling errors by incorporating an additional covariance term in the Kalman filter equations. The additional term is adjusted sequentially by feeding back measurement residual information. Nahi and Schaeffer used a Neyman-Pearson decision rule to detect when the calculated estimation error covariances became inconsistent with the actual covariances [N-1]. A

rational procedure was used to increase the estimation error covariance at the time of detection.

McAulay and Denlinger used a generalized likelihood ratio to detect maneuvers in a target described by Singer's acceleration model [M-3]. The discriminant in the detection process is the measurement residual. They formulated a suboptimal recursive algorithm using a single Kalman filter that switches between a high bandwidth mode and a low bandwidth mode, depending on whether a maneuver has been detected. Gholson developed an estimator for a maneuvering target that could be described by a semi-Markov process [G-1]. His analysis produced a filter, similar to a Kalman filter, but with an additional covariance term that maintained the filter gains at a level sufficient to track the maneuvers.

Several other formulations of the stochastic adaptive problem have appeared in the literature. Pierce and Sworder used an optimal control formulation to derive a closed loop controller for the switched linear plant problem [P-1]. Their results appear to be of more theoretical than practical interest. Nahi and Knobbe derived the optimal linear estimator for the switched linear plant problem by augmenting the system to include all the possible configurations and then using the Orthogonal-Projection theorem to minimize the mean square estimation error [N-2]. Although interesting, their method is inferior to the previous nonlinear estimators.

Additional comments on the above adaptive techniques will be made throughout the remainder of the dissertation.

## 2.0 FUNDAMENTAL BACKGROUND

### 2.1 Introduction

The purpose of this chapter is to review certain results of linear system theory and estimation theory that will be used in this dissertation. In addition, much of the notation used in the following chapters will be introduced here. Section 2.2 defines the basic linear control problem. State variable feedback is discussed in Section 2.3, and the optimal control formulation is reviewed in Section 2.4. The Kalman filter is analyzed in Section 2.5, with special emphasis given to the measurement residual characteristics.

The above topics can be reviewed in greater detail in numerous excellent texts. Chen provides a very readable introduction to linear system theory [C-3]. Another good source is the text by Brogan [B-2]. Estimation theory is well covered by Meditch [M-4], Jazwinski [J-2] and Gelb [G-3]. The last text is especially recommended for its practical applications.

### 2.2 The Basic Control Problem

The state variable description of linear systems is at a high degree of development. An important advantage of this description is the reduction of the system differential equations to a single first order matrix equation, called the state equation.

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{z} &= \mathbf{Hx}\end{aligned}\tag{2.2.1}$$

where,

$\mathbf{x} = n \times 1$       system state vector

$$\begin{aligned}x(k+1) &= \Phi x(k) + \Psi u(k) \\z(k+1) &= Hx(k+1)\end{aligned}\tag{2.2.7}$$

$\Phi = n \times n$  state transition matrix  
 $\Psi = n \times m$  input coefficient matrix  
 $H = r \times n$  measurement coefficient matrix

### 2.3 State Variable Feedback

The transient response of a linear system can be adjusted by the use of state variable feedback, i.e.

$$u(k+1) = r(k+1) + Fx(k+1)\tag{2.3.1}$$

where,

$r(k+1) = m \times 1$  reference input  
 $u(k+1) = m \times 1$  total system input  
 $F = m \times n$  feedback gain matrix

Substituting equation (2.3.1) into (2.2.5) and collecting terms gives

$$x(k+1) = [\Phi + \Psi F]x(k+1) + \Psi r(k)\tag{2.3.2}$$

If the original system is controllable [C-3], then it is possible to find an F matrix such that  $[\Phi + \Psi F]$  has any desired set of eigenvalues. For multiple input systems F is non-unique.

A method for determining F for a single input system is now given.

Step 1. Find the characteristic equation of the original system,

$$|zI - A| = z^N + a_{N-1}z^{N-1} + \dots + a_1z + a_0.\tag{2.3.3}$$

This can be done directly by using Leverrier's algorithm [M-6] or indirectly by calculating the eigenvalues,  $\lambda_i$ , and multiplying out the expression

$u = m \times 1$  deterministic input vector

$z = r \times 1$  measurement vector

$A = n \times n$  system dynamics matrix

$B = n \times m$  input coefficient matrix

$H = r \times n$  measurement coefficient matrix

The solution of the state equation is readily found by integrating (2.2.1).

$$x(t) = \exp(A(t-t_0)) x(t_0) + \int_{t_0}^t \exp(A(t-\tau)) Bu(\tau) d\tau \quad (2.2.2)$$

where

$$\exp(At) = I + At + \frac{(At)^2}{2!} + \frac{(At)^3}{3!} + \dots$$

and  $x(t_0)$  is the initial value of the state vector.

In systems using a digital controller the sampled data version of 2.2.1 is required. Holding the input constant between sample intervals reduces (2.2.3) to,

$$x(t_{k+1}) = \exp(A(t_{k+1}-t_k))x(t_k) + \left[ \int_{t_k}^{t_{k+1}} \exp(A(t_{k+1}-\tau)) d\tau \right] Bu(k) \quad (2.2.4)$$

Rewriting 2.2.4,

$$x(k+1) = \Phi x(k) + \Psi u(k) \quad (2.2.5)$$

where,

$$\begin{aligned} \Phi &= \exp(A(t_{k+1}-t_k)) \\ \Psi &= \left[ \int_{t_k}^{t_{k+1}} \exp(A(t_{k+1}-\tau)) d\tau \right] B \end{aligned} \quad (2.2.6)$$

The complete discrete time system description is,

$$\prod_{i=1}^N (z - \lambda_i) \quad (2.3.4)$$

Step 2. Construct the controllable canonical form for the original system using the characteristic equation of Step 1.

$$\Phi_c = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-1} \end{bmatrix}; \Psi_c = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad (2.3.5)$$

Step 3. Compute the controllability matrices for the original system structure and the controllable structure.

$$C_1 = [\Psi \quad \Phi\Psi \quad \cdots \quad \Phi^{N-1}\Psi]$$

$$C_2 = [\Psi_c \quad \Phi_c\Psi_c \quad \cdots \quad \Phi_c^{N-1}\Psi_c] \quad (2.3.6)$$

Step 4. Compute the transformation matrix P.

$$P = C_1 C_2^{-1} \quad (2.3.7)$$

Step 5. For the desired closed loop poles, calculate the corresponding characteristic equation and controllable system structure.

$$\Phi_2 = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -b_0 & -b_1 & -b_2 & \cdots & -b_{N-1} \end{bmatrix}; \Psi_2 = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad (2.3.8)$$

Step 6. Equate the desired system's controllable structure to the original system's controllable structure with state variable feedback, and solve for  $F_c$ .

$$\phi_2 = [\phi_c + \psi_c F_c]$$

$$\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -b_0 & -b_1 & -b_2 & \cdots & -b_{N-1} \end{bmatrix} = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \cdots & 1 \\ f_1^{-a_0} & f_2^{-a_1} & \cdots & f_N^{-a_{N-1}} \end{bmatrix}$$

$$f_i = a_{i-1} - b_{i-1} \quad (2.3.9)$$

Step 7. Transform the feedback matrix from the controllability coordinate system to the original coordinate system.

$$F = F_c P^{-1}$$

#### 2.4 Optimal Control Formulation

As shown in Section 2.3, the feedback gain matrix,  $F$ , can be found by specifying the eigenvalues for the desired closed-loop system. An alternative means of determining  $F$  is via an optimal control formulation. This requires the specification of a quadratic cost functional to effect a trade-off between control effort and state variable performance. For example, the cost functional for the infinite time optimal regulator problem is given by,

$$J = \frac{1}{2} \sum_{k=0}^{\infty} \{x^T(k) Q x(k) + u^T(k) R u(k)\} \quad (2.4.1)$$

where

$Q = n \times n$  is a positive semi-definite matrix that penalizes non-zero state values.

$R = m \times n$  is a positive definite matrix that penalizes large control efforts.

Minimizing  $J$  subject to the system constraints, equation (2.2.5), leads to the following result for  $u(k)$ .

$$u(k) = Fx(k) \quad (2.4.2)$$

$$F = (R + \Psi^T P \Psi)^{-1} \Psi^T P \Phi$$

where  $P$  is the solution of the steady state discrete Ricatti equation.

$$P = (\Phi - \Psi F)^T P (\Phi - \Psi F) + F^T R F + Q \quad (2.4.3)$$

In using the optimal control formulation, the designer loses the freedom to specify the closed-loop eigenvalue locations, and hence, loses the ability to directly set the transient response. On the other hand, the designer gains the ability to adjust the control performance relative to the control effort, indirectly controlling the transient response. For example, penalizing the state vector more and the control vector less will cause the system to respond faster.

## 2.5 The Structure of the Kalman Filter

The discrete time description of a linear system corrupted by noise is,

$$\begin{aligned} x(k+1) &= \Phi x(k) + \Psi u(k) + \Gamma w(k) \\ z(k+1) &= Hx(k+1) + v(k+1) \end{aligned} \quad (2.5.1)$$

The definitions for the above quantities are given in (2.2.1) and (2.2.7), and

$w = p \times 1$       random disturbance input  
 $v = r \times 1$       measurement noise  
 $\Gamma = n \times p$       disturbance coefficient matrix

The general filtering problem is to obtain an estimate  $\hat{x}(k+1)$  for  $x(k+1)$  given the measurements up to and including  $z(k+1)$ . For linear systems having Gaussian distributed noise processes the optimal estimator is the well known Kalman filter [K-2][G-3][M-4][J-2]. The Kalman filter is optimal in the sense that it minimizes the mean square state estimation error, and for this reason it is alternately referred to as the mmse filter or the optimal linear filter.

It is assumed that  $w$  and  $v$  are zero mean, white Gaussian random variables, and are therefore completely described by their covariance matrices  $Q$  and  $R$ , respectively. The structure of the Kalman filter is

$$\hat{x}(k+1) = \hat{x}(k+1/k) + K(k+1)[z(k+1) - H\hat{x}(k+1/k)] \quad (2.5.2)$$

where

$$\hat{x}(k+1/k) = \Phi\hat{x}(k) + \Psi u(k)$$

The time varying Kalman gain,  $K(k+1)$ , is calculated as follows.

$$M(k+1) = \Phi P(k)\Phi^T + \Gamma Q \Gamma^T \quad (2.5.3)$$

$$K(k+1) = M(k+1)H^T [HM(k+1)H^T + R]^{-1} \quad (2.5.4)$$

$$P(k+1) = [I - K(k+1)H]M(k+1) \quad (2.5.5)$$

where,

$$Q = E[w(k)w^T(k)] \text{ disturbance covariance}$$

$$R = E[v(k)v^T(k)] \text{ measurement noise covariance} \quad (2.5.6)$$

The physical interpretations and dimensions of the above matrices are,

$$\begin{aligned}
 \hat{x}(k+1) &= n \times 1 && \text{filtered state estimate at time } k+1 \\
 \hat{x}(k+1/k) &= n \times 1 && \text{predicted state estimate at time } k+1 \text{ given measurements up to } t = k \\
 K(k+1) &= n \times r && \text{Kalman filter gain} \\
 P(k) &= n \times n && \text{covariance of the filtered state estimation error} \\
 M(k+1) &= n \times n && \text{covariance of the predicted state estimation error}
 \end{aligned} \tag{2.5.7}$$

A block diagram of the Kalman filter is given in Figure 2.5.1.

Since the original work of Kalman and Bucy, the optimal linear filter has been derived through a variety of formulations. Notably, it has been shown that the optimal linear filter is isomorphic to the optimal linear regulator reviewed in Section 2.4 (for example see [G-3]). Hence, by a simple renaming of terms the optimal regulator derivation, via the Ricatti equation, is equally valid for the Kalman filter. A major implication of this 'duality' is that the Kalman filter, under fairly loose conditions, is stable. Rewriting equation (2.5.2) for  $u(k)=0$ ,

$$\hat{x}(k+1) = [I-K(k+1)H]\phi\hat{x}(k) + K(k+1)z(k+1) \tag{2.5.8}$$

This is a first order difference equation with the measurement sequence driving the filter dynamics. Stability implies that the eigenvalues of  $[I-K(k+1)H]$  are inside the unit circle. The inherent stability of the Kalman filter is an important ingredient in the operation of the adaptive estimator analyzed in Chapter Three.

Increasing the Kalman gain is equivalent to increasing the filter bandwidth. Referring to equation (2.5.8), if  $k = 0$ , then  $\hat{x}(k+1)=\phi\hat{x}(k)$ .

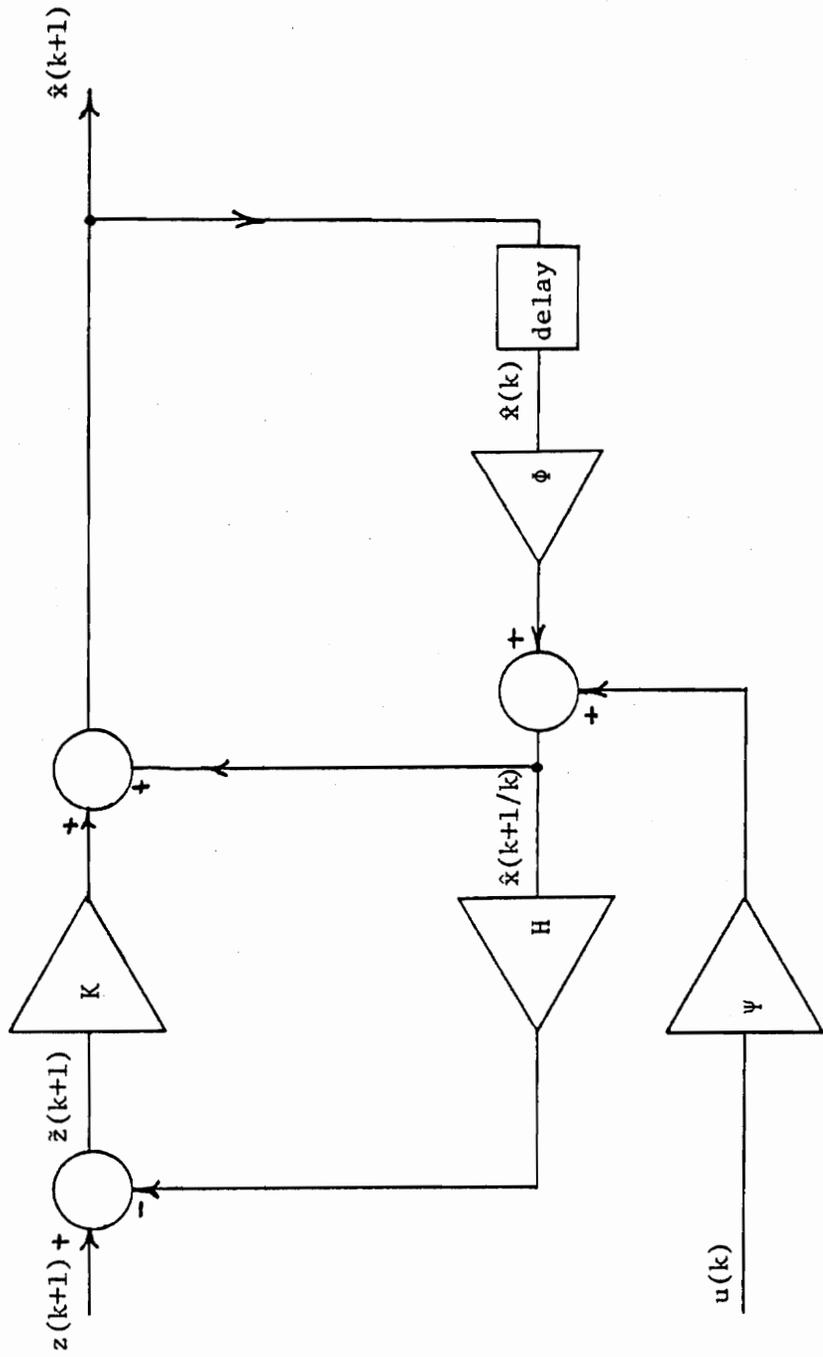


Figure 2.5.1 Kalman Filter Structure

That is, the estimate is not affected by the measurement, and hence, the bandwidth is zero. As  $K$  increases from zero, a larger portion of the noisy measurement is incorporated in the estimate, and the higher frequency fluctuations become visible. This corresponds to a higher frequency bandwidth.

The Kalman filter performs two different functions. First, as already mentioned, is noise filtering. The second function is state reconstruction from a measurement vector having fewer elements than the number of states. For example, let  $R = 0$  indicating no measurement noise, then

$$K(k+1) = M(k+1)H^T [HM(k+1)H^T]^{-1}. \quad (2.5.9)$$

Substituting (2.5.9) into (2.5.5) gives

$$P(k+1) = [I - M(k+1)H^T [HM(k+1)H^T]^{-1}H]M(k+1) \quad (2.5.10)$$

Post-multiplying by  $H^T$ ,

$$\begin{aligned} P(k+1)H^T &= M(k+1)H^T - M(k+1)H^T [HM(k+1)H^T]^{-1}HM(k+1)H^T \\ P(k+1)H^T &= 0 \end{aligned} \quad (2.5.11)$$

Now if  $H$  has full rank,  $P(k+1) = 0$  indicating that once the  $z(k+1)$  measurement is processed the state is known exactly. This is intuitively satisfying. On the other hand, if  $H$  does not have full rank, then  $P(k+1)$  is simply the least-square-estimation-error associated with observing a reduced number of measurements.

The Kalman filter has a predictor/corrector structure. First, the current filtered estimate is extrapolated to the next iteration, and then a correction term is added when the subsequent measurement is processed. The amount of correction is determined by subtracting the actual measurement from the predicted measurement, and multiplying by

the Kalman gain, as shown in equation (2.5.2). The measurement residual (also called the 'innovation' [K-3] is defined as,

$$\tilde{z}(k+1) = z(k+1) - H\hat{x}(k+1/k) \quad (2.5.12)$$

It is now shown that the measurement residual is a zero mean white gaussian random process. Substituting for  $z(k+1)$  using (2.5.1) and  $x(k+1/k)$  using (2.5.2) gives,

$$\begin{aligned} \tilde{z}(k+1) &= H[\phi x(k) + \Psi u(k) + \Gamma w(k)] + v(k) - H[\phi \hat{x}(k) + \Psi u(k)] \\ &= H\phi(x(k) - \hat{x}(k)) + H\Gamma w(k) + v(k) \end{aligned} \quad (2.5.13)$$

Taking the expected value of each side gives,

$$\begin{aligned} E\{\tilde{z}(k+1)\} &= H\phi(E\{x(k)\} - \hat{x}(k)) + H\Gamma E\{w(k)\} + E\{v(k)\} \\ &= 0 \end{aligned} \quad (2.5.14)$$

Also, using (2.5.13), and defining  $\tilde{x}(k) = x(k) - \hat{x}(k)$ ,

$$\begin{aligned} E\{\tilde{z}\tilde{z}^T\} &= H\phi E\{\tilde{x}(k)\tilde{x}^T(k)\}\phi^T H^T + H\Gamma E\{w(k)w^T(k)\}\Gamma^T H^T + E\{vv^T\} \\ &= H\phi P(k)\phi^T H^T + H\Gamma Q\Gamma^T H^T + R \\ &= H[\phi P(k)\phi^T + \Gamma Q\Gamma^T]H^T + R \\ E\{\tilde{z}(k)\tilde{z}^T(k)\} &= HM(k+1)H^T + R \end{aligned} \quad (2.5.15)$$

Comparing equations (2.5.15) and (2.5.4), it is seen that the measurement residual covariance is available from the Kalman filter equations.

The whiteness of the measurement residual process is shown by substituting equation (2.5.13) into,

$$\begin{aligned} &E\{\tilde{z}(k+N)\tilde{z}^T(k)\} \\ E\{\tilde{z}(k+N)\tilde{z}^T(k)\} &= [H\phi\tilde{x}(k+N-1) + H\Gamma w(k+N-1) + v(k+N-1)] \\ &[H\phi\tilde{x}(k-1) + H\Gamma w(k-1) + v(k-1)]^T \end{aligned} \quad (2.5.16)$$

Because  $w$  and  $v$  are both white gaussian processes and are uncorrelated,

$$E\{w(k+N-1)w^T(k-1)\} = 0$$

$$E\{w(k+N-1)v^T(k-1)\} = 0$$

Also, due to the optimality of the Kalman filter, the estimation error is uncorrelated with the noise sequences  $w$  and  $v$ . Moreover, the estimation error has an auto-correlation function (see [M-4]).

$$\Xi(k+N, k) = P(k)\delta(N) \quad (2.5.17)$$

Hence,

$$\begin{aligned} E\{\tilde{z}(k+N)\tilde{z}^T(k)\} &= H\Phi E\{\tilde{x}(k+N-1)\tilde{x}^T(k-1)\}\Phi^T H^T \\ &= 0 \end{aligned} \quad (2.5.18)$$

In the actual computations a better formula is used in place of equation (2.5.5). This equivalent formula is

$$P(k+1) = [I-K(k+1)H]M(k+1)[I-K(k+1)H]^T + KRK^T \quad (2.5.19)$$

Equation 2.5.19 is superior to equation (2.5.5) because it guarantees that  $P(k+1)$  remains symmetric despite accumulated computer round-off errors.

## 3.0 THE MODIFIED PARTITIONED ADAPTIVE ESTIMATOR

### 3.1 Introduction

The main result of this chapter is the development of an estimator for highly nonlinear systems based upon the structure of a stochastic adaptive estimator. In addition, a design method will be presented that will be applicable to a large class of nonlinear systems. The estimator has several key advantages over the other existing methods. First, the structure of the adaptive estimator is an arrangement of linear filters. This allows the off-line computation of a large percentage of the filtering equations, thus reducing the on-line cycle time of the estimation algorithm. Furthermore, this structure takes advantage of the powerful methods of linear systems and linear estimation theory which are at a much higher level of development as compared to nonlinear estimation theory. Second, the estimator is readily implemented in a parallel processing type of organization. This has advantages in computational speed and estimator redundancy.

The existing estimators for nonlinear systems can be grouped, loosely, into linearization methods and adaptive methods. The best known of the linearization methods is the Extended Kalman Filter (EKF) [G-3]. This algorithm requires that the nonlinear equations be expressed in a Taylor series expansion with the expansion point being the last state estimate. The linear terms of the Taylor series form the bases for filtering equations that are similar to the Kalman filtering equations. If the nonlinear system is well specified, then the EKF can be very accurate. However, if the nonlinear system model is inexact,

then the EKF will diverge. The requirement for a precise model is a major disadvantage of this method. Another disadvantage of the EKF is the formidable amount of data processing that is required between measurements. The Taylor series terms must be re-evaluated at each iteration and, therefore, the filter equations must be solved on-line.

A motivation for applying adaptive techniques to the nonlinear estimation problem is that a nonlinear system can be thought of as a linear system having unknown time varying parameters. If these parameters could be identified at each iteration, a linear estimator could be constructed that would give nearly optimal estimates. Several methods have been developed and applied successfully to plants having relatively small nonlinearities. That is, the usual trajectory of the state vector is such that the nature of the plant changes slowly with time. Many of these methods are based on the adaptive estimator structure introduced by Magill [M-1] and refined by Lainiotis et al, and will be referred to as "Partitioned Adaptive Estimators" or simply PAE's [L-2]. Moose and Wang developed an estimation algorithm for a linear plant undergoing random configuration changes that could be modeled as a semi-Markov process [M-2]. A major advantage of this method is that the designer can utilize more of the available a priori information regarding the plant behavior to develop a better estimator. This estimator will form the basis of the nonlinear estimation algorithm developed in this chapter.

Athans et al applied an adaptive estimation technique to the nonlinear flight dynamics of the F-8c aircraft. The estimator was part of an overall adaptive controller referred to as "Multiple Model Adaptive

Control" [A-1]. It consisted of a bank of Kalman filters, each matched to a specific flight condition. In addition, a learning section utilized the measurement residual of each filter to identify the actual current plant configuration. The simulations that were conducted indicate that the method would track the aircraft configuration changes *provided* that sufficient time was spent in each configuration. Brown and Price applied the PAE method to a maneuverable target described by Singer's acceleration model [B-1]. They found that the estimator could not track the plant when it underwent relatively rapid maneuvers.

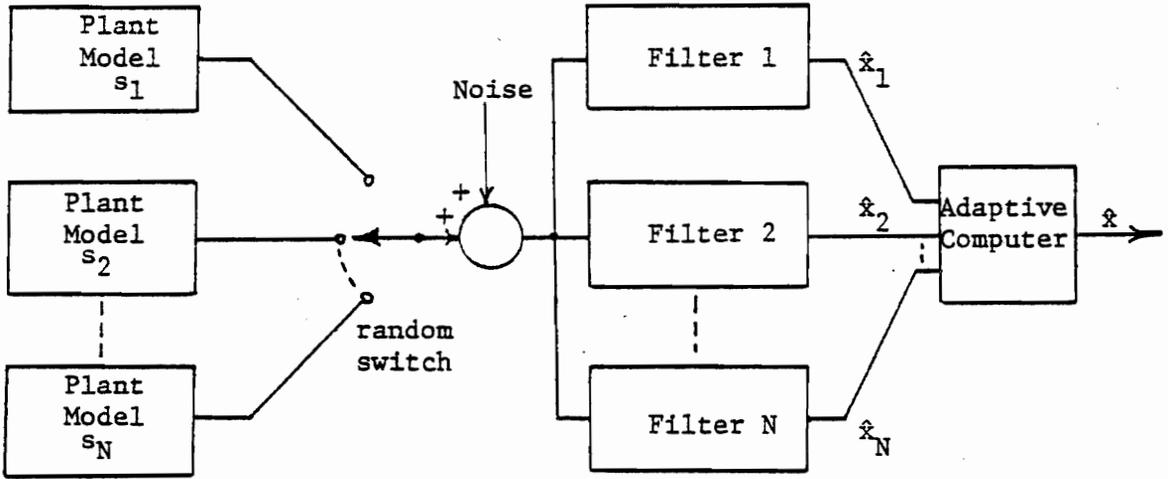
The fact that the PAE algorithm cannot track a plant undergoing rapid configuration changes is not surprising because, as discussed in the sections to follow, the algorithm was not designed to do so. It is inherently an identification algorithm and works excellently in that type of problem, but poorly when required to track a rapidly changing plant.

Section 3.2 analyzes the operation of a specific PAE algorithm based upon semi-Markovian plant modeling. This particular algorithm has features that enable it to be readily modified to track rapidly changing plant configurations. This modification is described in Section 3.3, and an example is given. The extension of this modified algorithm to the nonlinear estimation problem is straightforward and is done in Section 3.4, with a design example discussed in detail. The advantages and disadvantages of the method are discussed in Section 3.5.

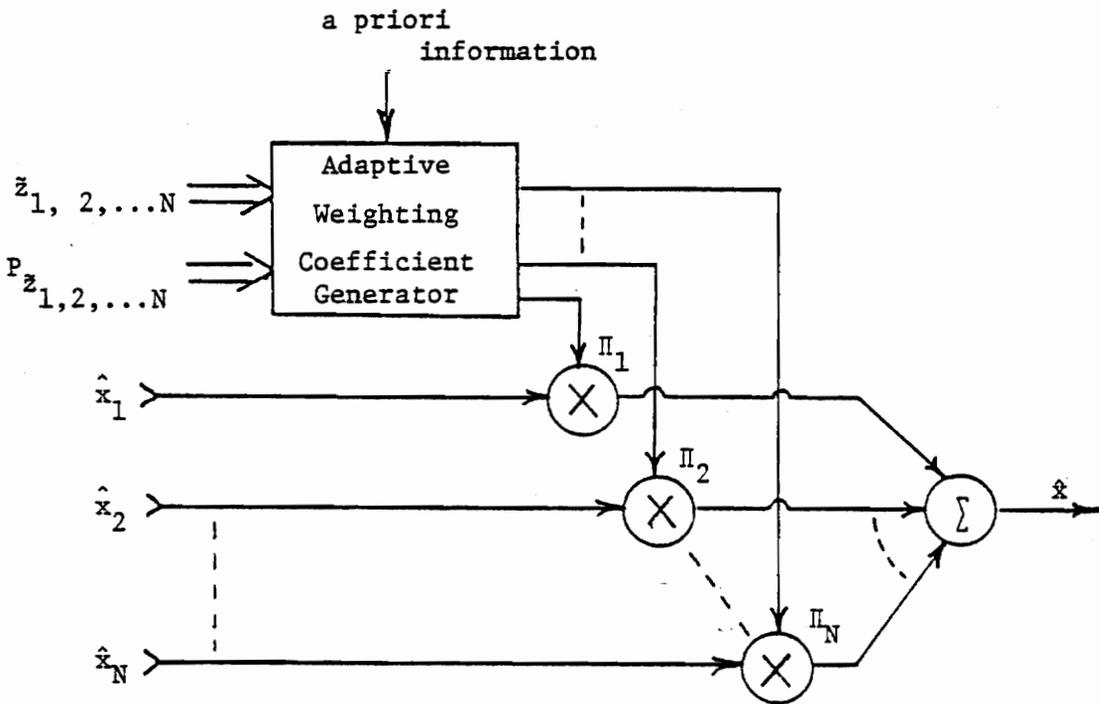
### 3.2 The Partitioned Adaptive Estimator with Semi-Markov Plant Modeling

The structure of the Partitioned Adaptive Estimator is shown in Figure 3.2.1. First introduced by Magill [M-1], the estimator consists of a bank of Kalman filters, called elemental estimators, each matched to a possible plant configuration. In addition, a learning or adaptive section transforms the set of elemental estimates,  $\hat{x}_i(k)$ , to an overall estimate,  $\hat{x}(k)$ . The identification problem is defined to be the joint problem of estimating the state and identifying the configuration of a plant that is in an unknown, but unchanging, configuration. Magill proved that the optimal estimate for the identification problem is a weighted sum of the elemental estimates. The time varying weighting coefficients,  $\pi_i$ , perform the identification function of the algorithm. Several methods exist for calculating  $\pi_i$ . The differences are primarily due to the extent of a priori information that is assumed about the plant, and how this information is assimilated into the calculations [L-2][A-1][M-2].

In operation the elemental estimators are all initialized with the same initial guess and uncertainty covariance matrix, and the weighting coefficients are initialized to reflect the a priori knowledge, if any, of the starting configuration. As the plant measurements are processed, the weighting coefficients become time varying a posteriori probabilities of the plant being in the respective configurations. Hawkes and Moore derived an upper bound for the estimation mean square error and proved that the PAE algorithm will converge exponentially to the mean square error of a perfectly matched Kalman filter [H-3]. A major implication of this result is that one of the weighting coefficients will



(a) The bank of Kalman filters.



(b) The structure of the adaptive computer.

Figure 3.2.1 The Partitioned Adaptive Estimator Structure

approach unity, thus identifying the plant configuration.

Moose extended the PAE algorithm to the switched linear plant problem by incorporating a statistical description of the plant configuration changes into the computation of the weighting coefficients [M-2]. The net result is an identification algorithm that is sensitized to the possibility that a configuration change can occur. For completeness, the derivation of this method is synopsized here.

Given a switched linear plant description,

$$\begin{aligned}
 x(k+1) &= \Phi_i x(k) + \Psi_i u(k) + \Gamma_i w(k) \\
 z(k+1) &= Hx(k+1) + v(k+1) \\
 \Phi_i &\in \{\Phi_1 \ \Phi_2 \ \dots \ \Phi_m\} \\
 \Psi_i &\in \{\Psi_1 \ \Psi_2 \ \dots \ \Psi_m\} \\
 \Gamma_i &\in \{\Gamma_1 \ \Gamma_2 \ \dots \ \Gamma_m\}
 \end{aligned} \tag{3.2.1}$$

The system matrices switch randomly amongst the M possible plant configurations.

The optimal estimate of the state at time k+1 is

$$\hat{x}(k+1) = E[x(k+1)/\hat{x}(k), z(k+1)], \tag{3.2.2}$$

$$\hat{x}(k+1) = \int_{-\infty}^{\infty} x p[x(k+1) = x/\hat{x}(k), z(k+1)] dx. \tag{3.2.3}$$

And using  $s_i(k)$  to represent the  $i^{\text{th}}$  plant configuration at time k,

$$p[x(k+1)=x/\hat{x}(k), z(k+1)] = \sum_{i=1}^N p[x(k+1)=x/\hat{x}(k), z(k+1), s_i(k+1)] \Pi_i(k+1), \tag{3.2.4}$$

where

$$\Pi_i(k+1) = \text{Prob. } \{s_i(k+1)/\hat{x}(k), z(k+1)\}. \tag{3.2.5}$$

The well known result that  $\hat{x}(k+1)$  is a weighted sum of the individual

parameter-conditioned estimates is now forthcoming. From (3.2.2) with expression (3.2.4) used for the density function

$$\hat{x}(k+1) = \sum_{i=1}^N \hat{x}_i(k+1) \Pi_i(k+1), \quad (3.2.5)$$

where

$$\hat{x}_i(k+1) = E[x(k+1)/\hat{x}(k), z(k+1), s_i(k+1)]. \quad (3.2.7)$$

The key to (3.2.5) lies in the computation of  $\Pi_i(k+1)$  since  $\hat{x}_i(k+1)$  can be obtained by the standard Kalman filter algorithm for  $i = 1, 2, \dots, N$ . By Bayes' rule

$$\Pi_i(k+1) = \frac{p[z(k+1)/s_i(k+1), \hat{x}(k)] P\{s_i(k+1)/\hat{x}(k)\}}{p[z(k+1)/\hat{x}(k)]}, \quad (3.2.8)$$

for each  $i = 1, 2, \dots, N$ . Since the denominator is not dependent on  $i$ , the normalizing can be done without calculating  $p[z(k+1)/\hat{x}(k)]$  through the relation

$$\sum_{i=1}^N \Pi_i(k+1) \text{ for any } k. \quad (3.2.9)$$

The probability density  $p[z(k+1)/s_i(k+1), \hat{x}(k)]$  is approximately normally distributed and will be represented by a Gaussian density function whose mean and covariance are available from the parameter conditioned Kalman filter algorithm. Thus, separate filters are constructed, one for each linearized region  $s_i$ ;  $i = 1, 2, \dots, N$ . The  $i^{\text{th}}$  filter algorithm is given by equations (3.2.10) to (3.2.13) below [M-4].

$$\hat{x}_i(k+1) = \phi_i \hat{x}_i(k) + \psi_i u(k) + K_i(k+1)[z(k+1) - H_i \phi_i \hat{x}_i(k) - H_i \psi_i u(k)], \quad (3.2.10)$$

$$M_i(k+1) = \phi_i P_i(k) \phi_i^T + \Gamma_i Q \Gamma_i^T, \quad (3.2.11)$$

$$K_i(k+1) = M_i(k+1)H_i^T [H_i M_i(k+1)H_i^T + R]^{-1}, \quad (3.2.12)$$

$$P_i(k+1) = [I - K_i(k+1)H_i]M_i(k+1), \quad (3.2.13)$$

where  $Q = E[w(k)w^T(k)]$ ,  $R = E[v(k)v^T(k)]$ , and  $\hat{x}_i(0)$  and  $P_i(0)$  are initial conditions given in terms of the  $i^{\text{th}}$  model statistics. Using the abbreviated notation that for each  $i = 1, 2, \dots, N$

$$P[z(k+1)/s_i(k+1), \hat{x}(k)] = P_i(\tilde{z}(k+1)); \quad (3.2.14)$$

then, from (3.2.1) and (3.2.11) the Gaussian density mentioned above is

$$P_i(\tilde{z}) = D_i \exp\{-\frac{1}{2}\tilde{z}^T V_i^{-1} \tilde{z}\}, \quad (3.2.15)$$

where  $V_i = H_i M_i(k+1)H_i^T + R$  and  $D_i$  is the appropriate normalizing constant for the density function. The time dependence of  $\tilde{z}$  in (3.2.15) was dropped to simplify the expression.

Finally, the numerator factor  $P\{s_i(k+1)/\hat{x}(k)\}$  in (3.2.8) can be expanded as follows:

$$P\{s_i(k+1)/\hat{x}(k)\} = \sum_{m=1}^N P\{s_i(k+1)/s_m(k)\}P\{s_m(k)/\hat{x}(k-1), z(k)\}, \quad (3.2.16)$$

where the conditioning information  $\hat{x}(k)$  in the second factor has been equated to  $(\hat{x}(k-1), z(k))$  and the conditioning on  $\hat{x}(k)$  in the first factor has been dropped because of its redundancy with  $s_m(k)$ . Thus, the first factors of (3.2.16) represent the semi-Markov probabilities of the plant's transitions. The effect of these transition probabilities will be discussed subsequently in detail. The second factors of (3.2.16) are the probabilities  $\Pi_m(k)$  calculated at the previous stage for  $m = 1,$



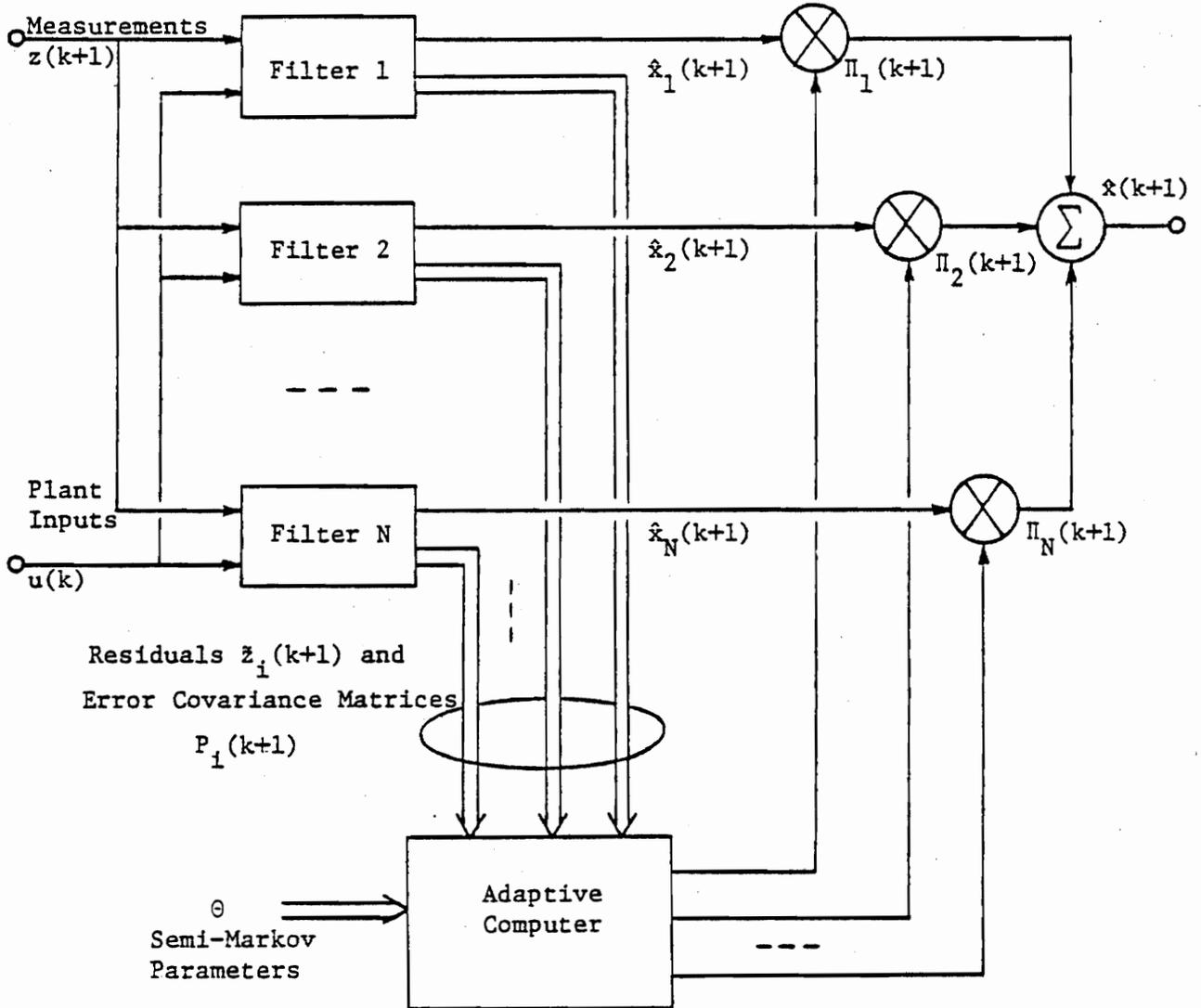


Figure 3.2.2 Modified Partitioned Adaptive Estimator, Showing Semi-Markov Addition.

where  $D = (2\pi)^{-\frac{n}{2}} |V|^{-\frac{1}{2}}$ . Note that  $V^{-1}$  is available from each separate filter algorithm, see (3.2.12).

The parameters  $\theta_{ji}$  represent the probability of the plant's changing from configuration  $s_i$  to configuration  $s_j$  during a sample interval. These transition probabilities can be pre-computed exactly [H-2] if statistical knowledge is known concerning the changes of configuration. Otherwise, the  $\theta_{ij}$  may be adjusted to achieve the desired degree of filtering and speed of learning in an empirical manner as suggested by the following analysis.

Equation (3.2.17) can be viewed as a discrete-time filter where the coefficients  $\theta_{ij}$  provide the coupling from one time step to another.

From (3.2.17)

$$\Pi_i(k+1) = [C_{k+1} \sum_{j=1}^N \theta_{ji} \Pi_j(k)] p_i(\bar{z}(k+1)), \quad (3.2.19)$$

for  $i = 1, 2, \dots, N$ . In general, the time-varying probabilities  $\Pi_i(k)$  will lag the density values  $p_i(\bar{z})$ . The extent to which this is true is demonstrated by considering two limiting cases, corresponding to no a priori knowledge and to total a priori knowledge, respectively. Case 1:  $\theta_{ij} = \frac{1}{N}$ ;  $i, j = 1, 2, \dots, N$ . From (3.2.19)

$$\Pi_i(k+1) = [C_{k+1} \frac{1}{N}] p_i(\bar{z}(k+1)), \quad (3.2.20)$$

for  $i = 1, 2, \dots, N$ . In other words, the raw density values are normalized to be the updated probabilities. Thus, (3.2.20) provides no additional "filtering" and the variations of the  $\Pi_i(k)$  will appear to be very noisy. At the same time, however, the response time for detecting changes in configuration will be minimal. Now consider a second extreme

where the off-diagonal terms of the  $\theta$ -matrix are zero.

$$\text{Case 2: } \theta_{ij} = \delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j; i, j = 1, 2, \dots, N. \end{cases}$$

Again, from (3.2.19)

$$\Pi_i(k+1) = [C_{k+1} \Pi_i(k)] p_i(\bar{z}(k+1)), \quad (3.2.21)$$

for  $i = 1, 2, \dots, N$ . In this case the  $\Pi_i$  will tend to lag the indications provided by the  $p_i(\bar{z})$ . As an extreme condition, with  $\Pi_1 = 1$  and  $\Pi_i = 0$  for  $i \neq 1$  at some time, there is no response to the residual densities. This total filtering condition is intuitively satisfying based upon the assumption of total a priori knowledge. In practice, there is never total certainty about the plant configuration and if e.g.  $\Pi_1 \approx 1$  with  $\Pi_j \approx 10^{-30}$  when a configuration change is made from  $s_i$  to  $s_j$ , then from (3.2.21)  $\Pi_j$  begins to increase (since  $p_j > p_i$ ). However, because  $\Pi_j$  was initially so small, a number of iterations is required before  $\Pi_j > \Pi_1$ , thus indicating that a transition has, in fact, occurred. This delay is referred to as the learning time of the adaptive estimator.

The two extreme cases just considered provide some insight into the learning mechanism; but, in general, the  $\theta_{ij}$  values will be chosen between these limiting cases. It has been found empirically that a useful form for the  $\theta$  matrix is to make the diagonal terms equal and close to unity and the off-diagonal terms equal and small. The off-diagonal terms, although small, are critical in determining the learning time. They act as an influence on the lower limits to which the probabilities can drop. This can be seen from (3.2.19) by assuming that the plant is holding in configuration  $s_j$  (thus,  $p_j > p_i$  and  $\Pi_j \approx 1$ , and noting that  $C_{k+1} \approx 1/\theta_{jj} p_j$ ),

$$\Pi_i(k+1) \approx \frac{1}{\theta_{jj} p_j} \theta_{ji} p_i \approx \theta_{ji} \frac{p_i}{p_j}, \quad (3.2.22)$$

where  $p_i$  is consistently small. Henceforth,  $\Pi_i$  will remain at this small value until the plant changes to a new configuration.

When choosing the  $\theta_{ij}$  parameters, a compromise must be made between estimator responsiveness and noise immunity. The smaller the off-diagonal terms, the longer the learning time, but the less likelihood of false identification.

A program was written to quantitatively examine the effect of the semi-Markov matrix on learning time. The results are summarized below.

#### Example 3-1

Verification that off-diagonal terms being zero will cause  $\Pi_i$  to approach zero for all but one probability. Given,

$$\theta = I; p(\tilde{z}_k) = \begin{bmatrix} 10^{-2} \\ 10^{-3} \\ 10^{-5} \end{bmatrix} \text{ for all } k; \Pi(0) = \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{bmatrix}$$

Table 3.2.1a gives  $\Pi_i(k)$  for the three configurations. As expected,  $\Pi_1$  rapidly approaches unity whereas  $\Pi_2$  and  $\Pi_3$  become asymptotically small, with  $\Pi_3$  approaching zero faster.

#### Example 3-2

Verification that non-zero off-diagonal terms will lead to steady state values for  $\Pi$ . Given,

$$\theta = \begin{bmatrix} .98 & .01 & .01 \\ .01 & .98 & .01 \\ .01 & .01 & .98 \end{bmatrix}; p(\tilde{z}_k) = \begin{bmatrix} 10^{-2} \\ 10^{-3} \\ 10^{-5} \end{bmatrix} \text{ for all } k; \Pi(0) = \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{bmatrix}$$

Table 3.2.1 Time-Varying Weighting Coefficients  
For Different Semi-Markov Matrices

(a)  $\theta = I$ ; weights approach zero.

Time	$\Pi_1$	$\Pi_2$	$\Pi_3$
1	3.333E-01	3.333E-01	3.333E-01
2	9.083E-01	9.083E-02	9.083E-04
3	9.901E-01	9.901E-03	9.901E-17
4	9.990E-01	9.990E-04	9.990E-10
5	9.999E-01	9.999E-05	9.999E-13
6	1.000E 00	1.000E-05	1.000E-15
7	1.000E 00	1.000E-06	1.000E-18
8	1.000E 00	1.000E-07	1.000E-21
9	1.000E 00	1.000E-08	1.000E-24
10	1.000E 00	1.000E-09	1.000E-27
11	1.000E 00	1.000E-10	1.000E-30
12	1.000E 00	1.000E-11	1.000E-33
13	1.000E 00	1.000E-12	1.000E-36
14	1.000E 00	1.000E-13	1.000E-39
15	1.000E 00	1.000E-14	1.000E-42
16	1.000E 00	1.000E-15	1.000E-45
17	1.000E 00	1.000E-16	1.000E-48
18	1.000E 00	1.000E-17	1.000E-51
19	1.000E 00	1.000E-18	1.000E-54
20	1.000E 00	1.000E-19	1.000E-57
21	1.000E 00	1.000E-20	1.000E-60
22	1.000E 00	1.000E-21	1.000E-63
23	1.000E 00	1.000E-22	1.000E-66
24	1.000E 00	1.000E-23	1.000E-69
25	1.000E 00	1.000E-24	1.000E-72

(b) Off-diagonal  $\theta_{ij} = .01$ ; weights have lower bounds.

Time	$\Pi_1$	$\Pi_2$	$\Pi_3$
1	3.333E-01	3.333E-01	3.333E-01
2	9.083E-01	9.083E-02	9.083E-04
3	9.891E-01	1.089E-02	1.208E-05
4	9.979E-01	2.117E-03	1.031E-05
5	9.988E-01	1.231E-03	1.022E-05
6	9.988E-01	1.142E-03	1.021E-05
7	9.989E-01	1.133E-03	1.021E-05
8	9.989E-01	1.133E-03	1.021E-05
9	9.989E-01	1.132E-03	1.021E-05
10	9.989E-01	1.132E-03	1.021E-05

Table 3.2.1b verifies equation (3.2.22), i.e.  $\Pi_2$  reaches a steady state value of

$$\Pi_2 \approx \theta_{12} p_2(z) / p_1(z) = (.01)(10^{-3}) / 10^{-2} = 10^{-3}$$

Example 3-3

Verification that learning time depends on off-diagonal entries.

Given,

$$\theta = \begin{bmatrix} .998 & .001 & .001 \\ .001 & .998 & .001 \\ .001 & .001 & .998 \end{bmatrix}; p(z) = \begin{bmatrix} 10^{-2} \\ 10^{-3} \\ 10^{-3} \end{bmatrix}; \Pi(0) = \begin{bmatrix} 10^{-20} \\ 1 \\ 10^{-20} \end{bmatrix}$$

Table 3.2.2a shows that 9 iterations are required for  $\Pi_1$  to change from  $10^{-20}$  to approximately unity. For

$$\theta = \begin{bmatrix} 1. & 10^{-10} & 10^{-10} \\ 10^{-10} & 1. & 10^{-10} \\ 10^{-10} & 10^{-10} & 1. \end{bmatrix}$$

16 iterations are required as shown in Table 3.2.2b. For a greater difference in probability density values, the learning time decreases, as shown in Table 3.2.2c.

The PAE algorithm with the additional semi-Markov ingredient has been shown to give good tracking performance for plants having configuration changes that occur at a rate that is low when compared to the actual plant dynamics [M-2][M-5]. A large number of physical situations fall in this category. For example, an aircraft undergoing a change in dynamics due to, say, a change in altitude. However, for plants having rapid configuration changes, the method will give poor results. This is the subject of the next section.

Table 3.2.2 Time-Varying Weighting Coefficients for Different Semi-Markov Matrices and Different  $p(\bar{z})$ .(a) Off-diagonal  $\theta_{ij} = .001$ ;  $p^T(\bar{z}) = [10^{-2} \quad 10^{-3} \quad 10^{-3}]$ .

Time	$\Pi_1$	$\Pi_2$	$\Pi_3$
1	1.000E-20	1.000E 00	1.000E-20
2	9.911E-03	9.891E-01	9.911E-04
3	9.911E-02	8.991E-01	1.811E-03
4	5.258E-01	4.727E-01	1.478E-03
5	9.171E-01	8.247E-02	4.319E-04
6	9.908E-01	9.009E-03	1.549E-04
7	9.989E-01	1.008E-03	1.166E-04
8	9.997E-01	2.011E-04	1.119E-04
9	9.998E-01	1.203E-04	1.114E-04
10	9.998E-01	1.122E-04	1.113E-04
11	9.998E-01	1.114E-04	1.113E-04
12	9.998E-01	1.113E-04	1.113E-04

(b) Off-diagonal  $\theta_{ij} = 10^{-10}$ ;  $p^T(\bar{z})$  of Part (a).

Time	$\Pi_1$	$\Pi_2$	$\Pi_3$
1	1.000E-20	1.000E 00	1.000E-20
2	1.000E-09	1.000E 00	1.000E-10
3	1.100E-08	1.000E 00	2.000E-10
4	1.110E-07	1.000E 00	3.000E-10
5	1.111E-06	1.000E 00	4.000E-10
6	1.111E-05	1.000E 00	5.000E-10
7	1.111E-04	9.999E-01	5.999E-10
8	1.110E-03	9.989E-01	6.992E-10
9	1.099E-02	9.890E-01	7.913E-10
10	1.000E-01	9.000E-01	8.111E-10
11	5.263E-01	4.737E-01	4.795E-10
12	9.174E-01	8.257E-02	1.010E-10
13	9.911E-01	8.920E-03	2.172E-11
14	9.991E-01	8.992E-04	1.227E-11
15	9.999E-01	8.999E-05	1.124E-11
16	1.000E 00	9.000E-06	1.112E-11
17	1.000E 00	9.000E-07	1.111E-11

(c)  $\theta$  of Part (b);  $p^T(\bar{z}) = [10^{-2} \quad 10^{-5} \quad 10^{-5}]$ .

Time	$\Pi_1$	$\Pi_2$	$\Pi_3$
1	1.000E-20	1.000E 00	1.000E-20
2	1.000E-07	1.000E 00	1.000E-10
3	1.001E-04	9.999E-01	2.000E-10
4	9.099E-02	9.090E-01	2.727E-10
5	9.901E-01	9.891E-03	4.056E-12
6	1.00E 00	9.990E-06	1.051E-13
7	1.000E 00	9.990E-09	1.001E-13
8	1.000E 00	1.009E-11	1.001E-13

### 3.3 Modified PAE Algorithm

Incorporating the semi-Markov statistics into the weighting coefficient calculations prevents the weights from going to zero. This is necessary if the estimator is to track a plant that will undergo configuration changes. However, the following analysis of the filter operation reveals that more is required if the estimator is to track a plant that has configuration changes occurring at a rate comparable to its own dynamics.

As previously discussed, the PAE algorithm consists of a bank of Kalman filters operating in parallel on the measurement data. The Kalman filters have a predictor/corrector structure. First, the future state is predicted from the current estimate and a model of the plant dynamics. Second, a correction is added to this predicted value when the noisy measurement is received. The amount of correction is determined by multiplying the measurement residual by the Kalman gain, i.e.

$$\hat{x}(k+1) = \phi_1 \hat{x}(k) + K_1(k+1) [z(k+1) - H_1 \phi_1 \hat{x}(k)] \quad (3.3.1)$$

where  $K$  is given by equation (3.2.12). Regrouping gives

$$\hat{x}(k+1) = (I - K_1 H_1) \phi_1 \hat{x}(k) + K_1 z(k+1). \quad (3.3.2)$$

When the Kalman filter is initialized,  $K$  is large and more measurement data is incorporated into the estimates. Consequently, the estimator converges fairly quickly to the actual plant state. The eigenvalues of  $(I-KH)$  can be shown to lie inside the unit circle, thus the Kalman filtering algorithm is inherently stable. See Chapter Two.

As the Kalman gains decrease, all the filters except the one matched to the plant will diverge. Eventually, the gains will reach steady state, and the diverging filters will stabilize around respective levels. The differences between these levels and the actual plant state are referred to as bias errors. Rewriting equation (3.2.2),

$$\hat{x}_i(k+1) = \phi_i \hat{x}_i(k) + K_i \tilde{z}_i(k+1)$$

where,

$$\tilde{z}_i(k+1) = z(k+1) - H_i \phi_i \hat{x}_i(k) \quad (3.3.3)$$

At first  $\tilde{z}_i$  is relatively small because all the filters have converged to the measurement sequence. The estimate,  $\hat{x}_i$ , propagates mainly according to  $\phi_i$ . If  $\phi_i$  is not the same as the plant,  $\hat{x}_i$  diverges and  $\tilde{z}_i$  increases. However,  $K_i \tilde{z}_i$  acts as a correcting term on  $\hat{x}$ , i.e. if  $\hat{x}$  is diverging positively with respect to the plant, then the measurements on the plant will be less than those predicted by the filter,  $z_{k+1} < H_i \phi_i \hat{x}_i(k+1)$ , and thus  $\tilde{z}$  will be negative. Eventually,  $\tilde{z}$  will be sufficiently large so that the correction at each iteration just offsets the additional error introduced by the mismatched dynamics, and an equilibrium bias error is attained. (Leondes and Pearson derived an upper bound for this bias error due to mismatch [L-3].)

At this point one filter is tracking the plant and the others are biased off the plant. The weighting coefficients indicate this by their values; nearly unity for the tracking filter, and approximately zero for the other filters. Now suppose a configuration change occurs. One of the other filters is now matched to the plant, but is also giving

bad estimates due to its initial bias error. Moreover, because the Kalman gain is at a small steady state value, a large number of iterations will be required before the bias error is subtracted away. Eventually, however, the newly matched Kalman filter will converge to the plant and start tracking it with small error. This is due to the fact that the Kalman filter is stable, as mentioned earlier.

The convergence time depends on the steady state value of  $K_1$ . A larger value will lead to quicker convergence. From equations (3.2.11), (3.2.12) and (3.2.13) it is seen that the gain can be artificially increased by increasing the value of  $Q$  used in the system description. The drawback to doing this is that the state estimate will always contain more of the noisy measurement.

A better approach is to detect when a configuration change occurs and then re-initialize the error covariance matrices to appropriately large values, and the weighting coefficients to the equal uncertainty condition. All the filters will then quickly converge to the measurements and the proper filter will begin tracking the plant. Shortly, the Kalman gains will decrease to their steady state values, so that only during the re-initialization period will large noise levels be present in the estimate. The adaptive estimator as constructed contains all that is needed to detect plant configuration changes. The key observation is that, when a transition occurs, the measurement residual values for the previously tracking filter will steadily increase. Thus, monitoring a filtered version of the measurement residual probabilities, i.e. the weighting coefficient for the tracking filter, will be sufficient to identify a transition. To summarize,  $\Pi_1$  of the tracking

filter will be monitored. When  $\Pi_1$  falls below a predetermined threshold, all the filters will be re-initialized. After a brief convergence time,  $\Pi_j$  of the newly tracking filter will be approximately unity. At this time,  $\Pi_j$  will be monitored to detect a second configuration change.

An example will illustrate the operation of the filter both with and without the gain re-initialization mechanism.

#### Example 3-4

The plant will switch randomly among the three configurations:

$$s_1: \begin{bmatrix} \dot{x}_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -30 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & -63.2 \end{bmatrix} \begin{bmatrix} u \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w \quad (3.3.4a)$$

$$s_2: \begin{bmatrix} \dot{x}_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w \quad (3.3.4b)$$

$$s_3: \begin{bmatrix} \dot{x}_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -30 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 63.2 \end{bmatrix} \begin{bmatrix} u \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w \quad (3.3.4c)$$

Where 'u' is a deterministic input, and 'w' is a random disturbance input. Notice that configurations  $s_1$  and  $s_3$  differ only in their input coefficient matrix.

An adaptive estimator using three Kalman filters was constructed according to the previous discussions. A computer simulation was performed under various conditions with the following results.

Referring to Figure 3.3.1, the plant is initially in configuration 1, with  $x(0) = -4.8$ . Noisy measurements are taken on both position and velocity; the noise covariance is 1.0 for both position and velocity; and the input is a step of 12.0. At time  $t = 1.05$  seconds (15 itera-

tions), the plant switches to configuration 2.

Figure 3.3.1a shows the actual plant position as a function of time, and the three Kalman filter position estimates. Filter 1 tracks the plant very well during the time the plant is in configuration 1. When the plant switches, filter 1 starts to diverge and filter 2 starts to converge. However, filter 2 is not providing good estimates until  $t \approx 3$  secs.

Table 3.3.1 is a table of the filter weighting coefficients,  $\Pi_1$ , corresponding to a semi-Markov matrix

$$\theta = \begin{bmatrix} .996 & .002 & .002 \\ .002 & .996 & .002 \\ .002 & .996 & .996 \end{bmatrix}.$$

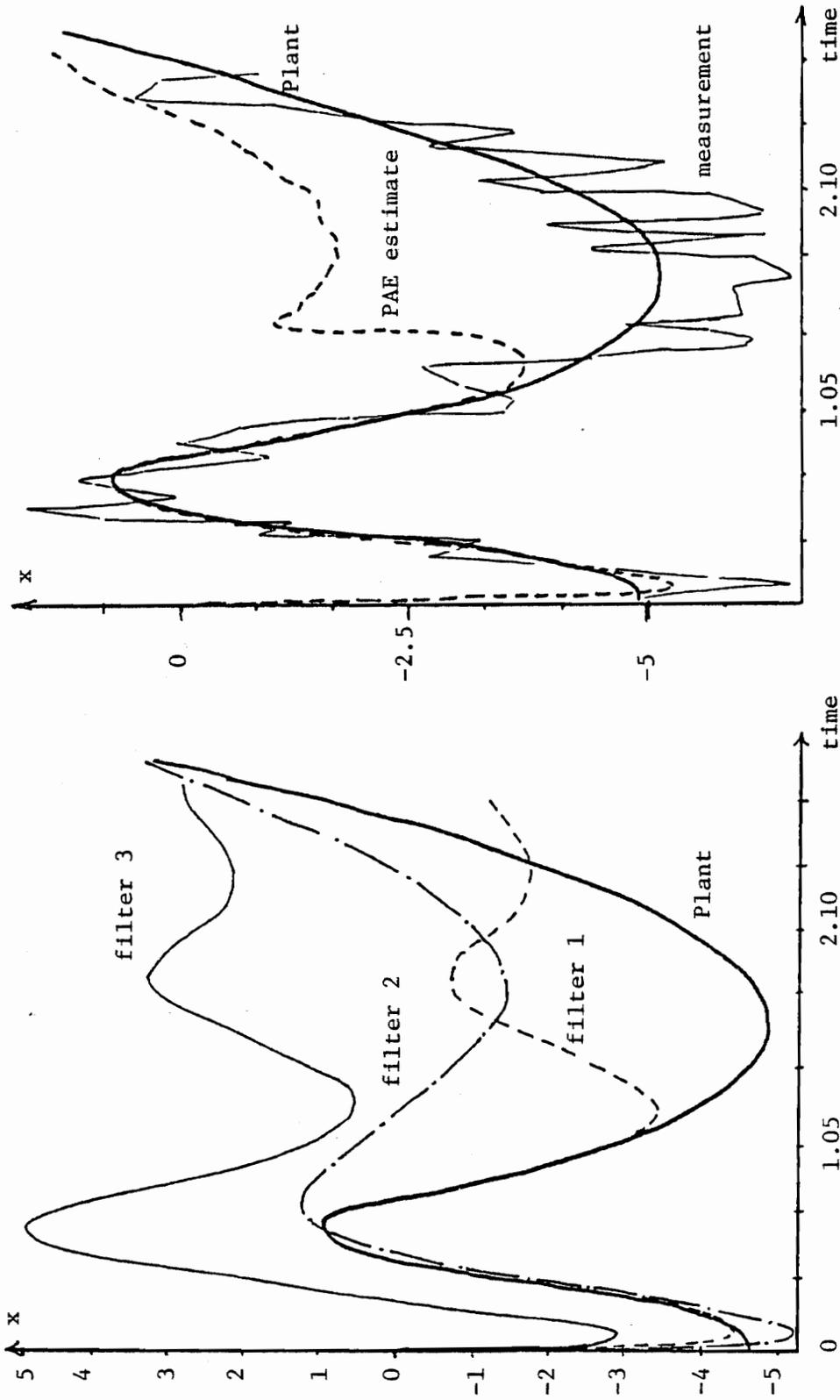
Initially,  $\Pi_1$  quickly approaches unity, whereas  $\Pi_2$  becomes very small ( $\Pi_2$  is set to zero whenever the exponent would otherwise cause underflow). After the switch,  $\Pi_2$  increases steadily until at time  $t \approx 1.4$  secs.,  $\Pi_2$  is unity. Hence the learning time is about .35 seconds.

Table 3.3.2 corresponds to

$$\theta = 1/3 \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

It was shown earlier that for this choice of semi-Markov matrix, the weights will follow the measurement residual probabilities, noise and all. Indeed, at iterations 7 and 8, because of the large noise present,  $\Pi_1$  is erroneously low.

Figure 3.3.1b shows the overall estimate corresponding to the weights in Table 3.3.1.



(a) The three filter estimates

(b) The weighted sum estimate

Figure 3.3.1 The Tracking Performance of the PAE Algorithm With Semi-Markov Addition But Without Re-Initialization

Table 3.3.1 Weighting Coefficients for  $\theta = \{\text{diagonal .996, off-diagonal .002}\}$ 

k	Time	$\Pi_1$	$\Pi_2$	$\Pi_3$	$P_1(z)$	$P_2(z)$	$P_3(z)$
1	0.00E-01	3.333E-01	3.333E-01	3.333E-01	1.824E-04	3.365E-04	7.427E-08
2	7.00E-02	3.386E-01	6.612E-01	1.378E-04	2.864E-02	2.118E-05	1.631E-07
3	1.40E-01	9.986E-01	1.437E-03	3.584E-08	4.954E-02	4.401E-12	6.238E-04
4	2.10E-01	1.000E 00	3.061E-13	2.531E-05	4.284E-02	2.573E-15	3.583E-05
5	2.80E-01	1.000E 00	1.204E-16	1.701E-06	5.461E-02	8.937E-06	5.726E-08
6	3.50E-01	1.000E 00	3.284E-07	2.107E-09	3.164E-02	2.945E-02	1.798E-05
7	4.20E-01	9.981E-01	1.866E-03	1.139E-06	1.574E-02	1.001E-02	6.456E-03
8	4.90E-01	9.967E-01	2.457E-03	8.225E-04	9.417E-02	3.189E-10	6.860E-06
9	5.60E-01	1.000E 00	1.515E-11	2.064E-07	5.754E-02	7.398E-38	2.581E-05
10	6.30E-01	1.000E 00	2.582E-39	9.007E-07	8.344E-02	0.000E-01	3.948E-05
11	7.00E-01	1.000E 00	0.000E-01	9.500E-07	7.802E-02	0.000E-01	1.041E-06
12	7.70E-01	1.000E 00	0.000E-01	2.681E-08	9.464E-02	0.000E-01	8.725E-04
13	8.40E-01	1.000E 00	0.000E-01	1.851E-05	5.941E-02	0.000E-01	1.862E-03
14	9.10E-01	9.999E-01	0.000E-01	6.349E-05	8.773E-02	0.000E-01	8.538E-07
15	9.80E-01	1.000E 00	0.000E-01	2.016E-08	3.894E-03	0.000E-01	2.305E-07
16	1.05E 00	1.000E 00	0.000E-01	1.188E-07	2.069E-02	3.020E-25	4.272E-05
17	1.12E 00	1.000E 00	2.932E-26	4.144E-06	3.879E-09	2.137E-24	8.630E-11
18	1.19E 00	1.000E 00	1.106E-18	4.477E-05	1.654E-14	1.532E-21	9.952E-21
19	1.26E 00	1.000E 00	1.854E-18	1.234E-09	6.662E-19	1.238E-17	1.430E-27
20	1.33E 00	9.640E-01	3.594E-02	4.155E-12	1.062E-23	3.912E-14	2.473E-31
21	1.40E 00	6.905E-09	1.000E 00	3.344E-19	5.324E-22	1.000E-12	1.249E-32
22	1.47E 00	1.069E-12	1.000E 00	2.509E-23	1.037E-20	2.766E-12	4.429E-32
23	1.54E 00	7.533E-12	1.000E 00	3.214E-23	1.734E-16	2.904E-12	7.985E-30
24	1.61E 00	1.202E-07	1.000E 00	5.521E-21	1.484E-09	7.624E-09	2.683E-22
25	1.68E 00	3.904E-04	9.996E-01	7.063E-17	2.987E-04	4.553E-05	7.805E-14
26	1.75E 00	1.550E-02	9.845E-01	3.390E-12	3.225E-07	3.225E-07	5.727E-22
27	1.82E 00	1.403E-04	9.999E-01	3.621E-18	3.149E-13	4.598E-03	1.516E-21
28	1.89E 00	1.472E-13	1.000E 00	6.623E-22	6.003E-17	3.152E-07	3.894E-29
29	1.96E 00	3.825E-13	1.000E 00	2.481E-25	6.063E-11	1.956E-08	7.414E-22
30	2.03E 00	6.225E-06	1.000E 00	7.613E-17	2.107E-12	2.963E-03	1.207E-18
31	2.10E 00	1.432E-12	1.000E 00	8.180E-19	2.324E-17	2.959E-04	3.064E-25
32	2.17E 00	1.577E-16	1.000E 00	2.079E-24	8.112E-17	1.072E-05	1.345E-25
33	2.24E 00	1.520E-14	1.000E 00	2.521E-23	9.921E-08	1.657E-03	3.448E-12
34	2.31E 00	1.073E-07	1.000E 00	3.729E-12	2.563E-10	4.388E-04	2.570E-16
35	2.38E 00	1.173E-09	1.000E 00	1.176E-15	2.747E-08	1.037E-01	5.642E-08
36	2.45E 00	5.316E-10	1.000E 00	1.092E-09	1.799E-09	3.937E-02	3.854E-11
37	2.52E 00	9.177E-11	1.000E 00	1.966E-12	2.430E-08	1.159E-01	1.216E-07
38	2.59E 00	4.209E-10	1.000E 00	2.104E-09	8.994E-11	7.180E-02	2.289E-10
39	2.66E 00	2.516E-12	1.000E 00	6.401E-12			

Table 3.3.2 Weighting Coefficients for  $\theta = \{\text{diagonal .333, off-diagonal .333}\}$ 

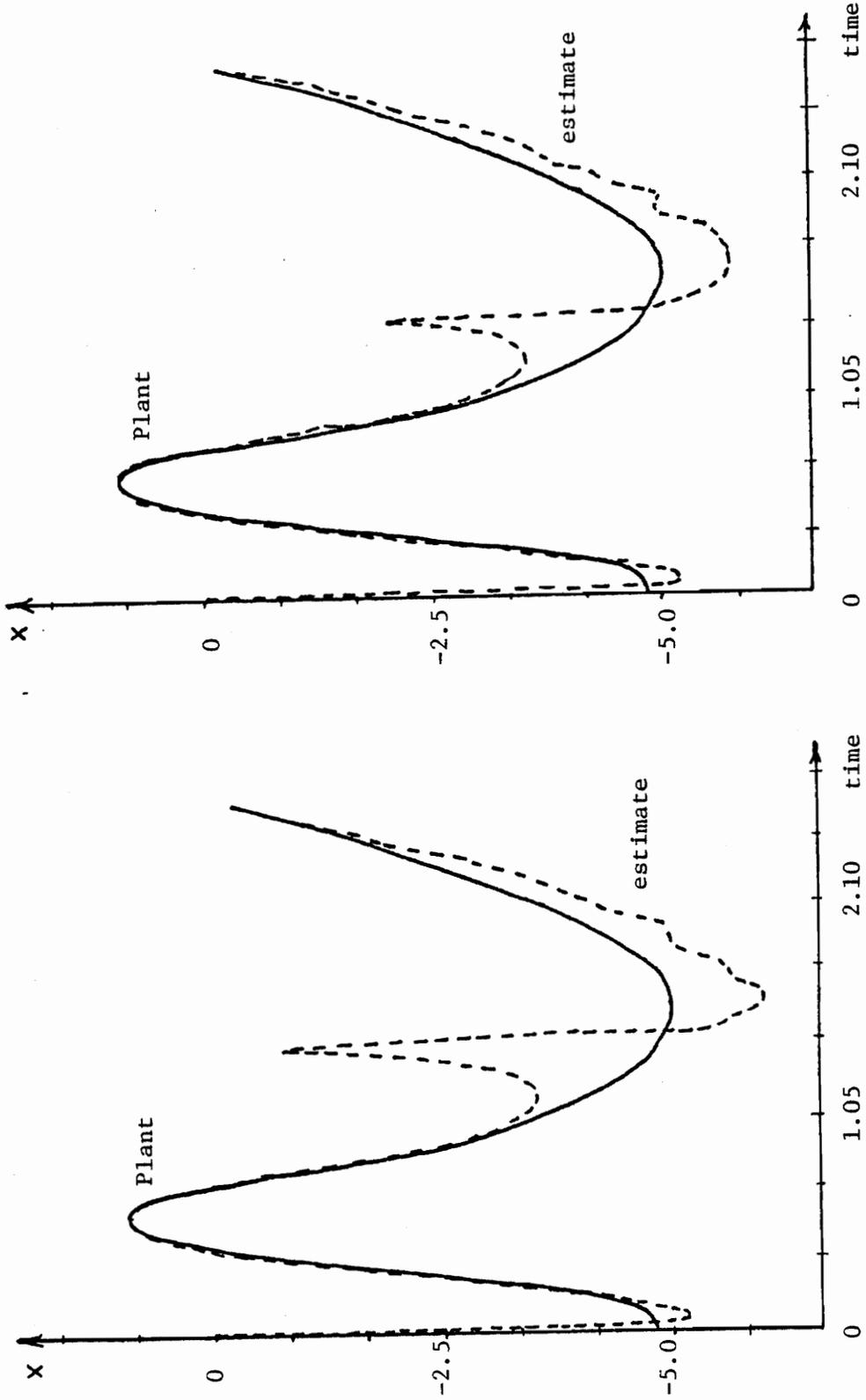
k	Time	$\Pi_1$	$\Pi_2$	$\Pi_3$	$P_1(z)$	$P_2(z)$	$P_3(z)$
1	0.00E-01	3.333E-01	3.333E-01	3.333E-01	1.624E-04	3.265E-04	7.427E-08
2	7.00E-02	3.386E-01	4.612E-01	1.374E-04	2.864E-02	2.118E-05	1.631E-07
3	1.40E-01	9.992E-01	7.456E-04	5.624E-06	4.954E-02	4.401E-12	6.238E-04
4	2.10E-01	9.879E-01	8.516E-11	1.207E-02	4.280E-02	2.573E-15	3.583E-05
5	2.80E-01	9.992E-01	5.827E-14	8.117E-04	5.461E-02	8.937E-06	5.726E-08
6	3.50E-01	9.998E-01	1.588E-04	1.014E-06	3.164E-02	2.945E-02	1.798E-05
7	4.20E-01	5.252E-01	4.745E-01	2.894E-04	1.574E-02	1.001E-02	6.456E-03
8	4.90E-01	4.906E-01	3.114E-01	1.980E-01	9.417E-02	3.189E-10	6.860E-06
9	5.60E-01	9.999E-01	3.368E-09	7.221E-05	5.754E-02	7.398E-38	2.581E-05
10	6.30E-01	9.996E-01	1.247E-36	4.351E-04	8.348E-02	0.000E-01	3.948E-05
11	7.00E-01	9.995E-01	0.000E-01	4.584E-04	7.802E-02	0.000E-01	1.041E-06
12	7.70E-01	1.000E-00	0.000E-01	1.295E-05	9.444E-02	0.000E-01	8.725E-04
13	8.40E-01	9.911E-01	0.000E-01	8.669E-03	5.941E-02	0.000E-01	1.862E-03
14	9.10E-01	9.705E-01	0.000E-01	2.953E-02	8.773E-02	0.000E-01	8.538E-07
15	9.80E-01	1.000E-00	0.000E-01	9.461E-06	3.894E-03	0.000E-01	2.305E-07
16	1.05E-00	9.999E-01	0.000E-01	5.744E-05	2.069E-02	3.020E-25	4.272E-05
17	1.12E-00	9.980E-01	1.414E-23	2.001E-03	3.879E-09	2.137E-24	8.630E-11
18	1.19E-00	9.789E-01	5.235E-16	2.114E-02	1.654E-14	1.532E-21	9.952E-21
19	1.26E-00	1.000E-00	8.986E-08	5.840E-07	6.662E-19	1.238E-17	1.430E-27
20	1.33E-00	5.252E-02	9.475E-01	1.094E-10	1.062E-23	3.912E-14	2.473E-31
21	1.40E-00	2.644E-10	1.000E-00	6.144E-18	5.324E-22	1.000E-12	1.249E-32
22	1.47E-00	5.169E-10	1.000E-00	1.213E-20	1.037E-20	2.766E-12	4.429E-32
23	1.54E-00	3.641E-09	1.000E-00	1.554E-20	1.734E-16	2.904E-12	7.985E-30
24	1.61E-00	5.604E-05	9.999E-01	2.664E-18	1.884E-09	7.624E-09	2.683E-22
25	1.68E-00	1.589E-01	8.411E-01	2.873E-14	2.987E-04	4.553E-05	7.805E-14
26	1.75E-00	8.654E-01	1.346E-01	2.250E-10	2.550E-09	3.225E-07	5.727E-22
27	1.82E-00	8.016E-03	9.920E-01	1.754E-15	3.149E-13	4.598E-03	1.516E-21
28	1.89E-00	6.651E-11	1.000E-00	3.201E-19	6.003E-17	3.152E-07	3.094E-29
29	1.96E-00	1.849E-10	1.000E-00	1.199E-22	6.063E-11	1.956E-08	7.414E-22
30	2.03E-00	3.000E-03	9.970E-01	3.669E-14	2.107E-12	2.963E-03	1.207E-16
31	2.10E-00	6.903E-10	1.000E-00	3.954E-16	2.324E-17	2.959E-04	3.064E-25
32	2.17E-00	7.624E-14	1.000E-00	1.005E-21	8.112E-17	1.072E-05	1.345E-25
33	2.24E-00	7.344E-12	1.000E-00	1.219E-20	9.921E-08	1.857E-03	3.448E-12
34	2.31E-00	5.185E-05	9.999E-01	1.802E-09	2.563E-10	4.388E-04	2.570E-16
35	2.38E-00	5.669E-07	1.000E-00	5.685E-13	2.747E-08	1.037E-01	5.642E-08
36	2.45E-00	2.569E-07	1.000E-00	5.274E-07	1.799E-09	3.937E-02	3.854E-11
37	2.52E-00	4.436E-04	1.000E-00	9.501E-10	2.430E-08	1.159E-01	1.216E-07
38	2.59E-00	2.034E-07	1.000E-00	1.018E-06	8.994E-11	7.180E-02	2.289E-10
39	2.66E-00	1.216E-09	1.000E-00	3.094E-09			

From the previous results it is seen that after a plant transition there is a relatively long convergence time (not to be confused with learning time) before the filter matched to the new configuration begins tracking the plant. Performance will be improved if the convergence time is decreased by detecting the configuration change and reinitializing the Kalman filter gains.

The previous example was run again with the addition of the above switch detection logic. The following improved results were obtained. See Figure 3.3.2 and Table 3.3.3. Figure 3.3.2a is identical to Figure 3.3.1b up until  $t \approx 1.4$  seconds, at which time the switch detection logic detects a transition. The filters are then reinitialized and Kalman filter 2 begins tracking the plant. It was found that increasing the off diagonal entries in the semi-Markov matrix decreased the learning time, thus reducing the large spike at  $t \approx 1.4$  seconds. See Figure 3.3.2b. Table 3.3.3 gives the weighting coefficients corresponding to Figure 3.3.2a. The arrow at  $t = 1.4$  seconds indicates the reinitializing of the filters; i.e. all the weights are made equal.

#### 3.4 Nonlinear Estimation Using the Modified PAE Algorithm

A nonlinear system can be approximated in different regions by corresponding linear systems. Moreover, a set of linear systems can be assembled to satisfactorily approximate the nonlinear system over its entire operational range. Now, as the state of the nonlinear plant moves through the state space, it will continuously move from a region closely approximated by one linear model to another region closely approximated by a second linear model. If Kalman filters were matched to the various linear approximations, the filter matched to the region that



(a)  $\theta = \{\text{diagonal } .996, \text{ off-diagonal } .002\}$  (b)  $\theta = \{\text{diagonal } .99, \text{ off-diagonal } .005\}$

Figure 3.3.2 Tracking Performance of the Modified PAE Algorithm, i.e. With Re-Initialization

Table 3.3.3 Weighting Coefficients for Figure 3.3.2

k	Time	$\Pi_1(z)$	$\Pi_2(z)$	$\Pi_3(z)$	$P_1(z)$	$P_2(z)$	$P_3(z)$
1	0.00E-01	3.333E-01	3.333E-01	3.333E-01	1.824E-04	3.565E-04	7.427E-04
2	7.00E-02	3.364E-01	6.612E-01	1.374E-04	2.864E-02	2.118E-05	1.631E-07
3	1.40E-01	9.984E-01	1.437E-03	3.584E-08	4.955E-02	4.401E-12	6.234E-04
4	2.10E-01	1.000E 00	3.061E-13	2.531E-05	4.284E-02	2.573E-15	3.583E-05
5	2.80E-01	1.000E 00	1.204E-16	1.701E-06	5.461E-07	8.937E-06	5.726E-08
6	3.50E-01	1.000E 00	3.286E-07	2.107E-09	3.164E-02	2.945E-02	1.794E-05
7	4.20E-01	9.981E-01	1.864E-03	1.139E-06	1.575E-02	1.001E-02	6.456E-03
8	4.90E-01	9.967E-01	2.457E-03	8.25E-04	9.417E-02	3.189E-10	6.860E-08
9	5.60E-01	1.000E 00	1.515E-11	2.068E-07	5.754E-02	7.398E-38	2.581E-05
10	6.30E-01	1.000E 00	2.582E-39	9.007E-07	8.344E-02	0.000E-01	3.948E-05
11	7.00E-01	1.000E 00	0.000E-01	9.500E-07	7.802E-02	0.000E-01	1.041E-06
12	7.70E-01	1.000E 00	0.000E-01	2.681E-08	9.464E-02	0.000E-01	8.725E-04
13	8.40E-01	1.000E 00	0.000E-01	1.851E-05	5.941E-02	0.000E-01	1.862E-03
14	9.10E-01	9.999E-01	0.000E-01	6.349E-05	8.773E-02	0.000E-01	8.538E-07
15	9.80E-01	1.000E 00	0.000E-01	2.014E-08	3.895E-03	0.000E-01	2.305E-07
16	1.05E 00	1.000E 00	0.000E-01	1.188E-07	2.069E-02	3.020E-25	4.272E-05
17	1.12E 00	1.000E 00	2.932E-26	4.148E-06	3.879E-09	2.137E-24	8.630E-11
18	1.19E 00	1.000E 00	1.106E-18	4.477E-05	1.654E-14	1.532E-21	9.952E-21
19	1.26E 00	1.000E 00	1.858E-10	1.234E-09	6.662E-19	1.238E-17	1.430E-27
20	1.33E 00	9.640E-01	3.598E-02	4.155E-12	1.062E-23	3.912E-14	2.473E-31
21	1.40E 00	3.333E-01	3.333E-01	3.333E-01	1.967E-07	1.647E-04	1.367E-14
22	1.47E 00	1.193E-03	9.988E-01	8.290E-11	1.464E-05	2.436E-02	1.123E-15
23	1.54E 00	1.925E-06	1.000E 00	9.267E-17	1.624E-06	4.689E-02	1.638E-18
24	1.61E 00	6.980E-04	1.000E 00	7.016E-20	2.780E-05	1.176E-01	3.055E-16
25	1.68E 00	4.747E-07	1.000E 00	5.215E-18	6.849E-03	4.303E-02	5.079E-11
26	1.75E 00	3.194E-04	9.997E-01	2.369E-12	1.629E-05	1.059E-01	2.064E-16
27	1.82E 00	3.583E-07	1.000E 00	3.914E-18	1.684E-03	5.174E-03	9.206E-10
28	1.89E 00	7.323E-04	9.993E-01	3.570E-10	1.900E-06	5.950E-02	7.094E-18
29	1.96E 00	4.752E-08	1.000E 00	2.396E-19	1.204E-05	6.485E-03	6.153E-17
30	2.03E 00	3.740E-06	1.000E 00	1.905E-17	4.313E-03	3.684E-02	1.050E-09
31	2.10E 00	2.355E-04	9.998E-01	5.723E-11	1.394E-07	1.368E-01	3.806E-16
32	2.17E 00	2.285E-09	1.000E 00	5.587E-18	9.514E-11	7.053E-02	1.175E-20
33	2.24E 00	2.709E-12	1.000E 00	3.347E-22	6.492E-06	2.643E-02	8.822E-12
34	2.31E 00	4.933E-07	1.000E 00	6.703E-13	8.456E-12	1.276E-01	2.496E-19
35	2.38E 00	1.331E-13	1.000E 00	3.924E-21	2.770E-11	3.572E-03	2.161E-12
36	2.45E 00	1.557E-11	1.000E 00	1.215E-12	7.897E-16	8.853E-02	8.584E-19
37	2.52E 00	1.791E-17	1.000E 00	1.947E-20	4.917E-15	2.655E-02	2.901E-15
38	2.59E 00	3.714E-16	1.000E 00	2.194E-16	4.584E-19	7.516E-02	3.188E-19
39	2.66E 00	1.224E-20	1.000E 00	8.517E-21			

the plant is currently in will provide better estimates than the other Kalman filters. This motivates the use of the adaptive estimator structure discussed in Section 3.3 for nonlinear systems.

A general nonlinear plant model is given by

$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}, \mathbf{u}) \quad (3.4.1)$$

where  $\mathbf{x}$  is a vector of plant states and  $\mathbf{u}$  is the vector of inputs. The actual nonlinearities implied by the function  $f$  in (3.4.1) are assumed to be known in some form, typically as curve-fitted information from tabulated data or a combination of tabulated data and analytical expressions. From the nonlinear data, linear models are developed by expanding (3.4.1) in the neighborhoods of a preselected set of points  $(\mathbf{x}_i, \mathbf{u}_i)$ ,  $i = 1, 2, \dots, N$  which represent the "centers" of a set of regions which forms a partition set for the complete configuration space, i.e. the space of points  $(\mathbf{x}, \mathbf{u})$ . Using a Taylor series expansion, the set of linear approximations becomes

$$\dot{\mathbf{x}} = \mathbf{A}_i \mathbf{x} + \mathbf{B}_i \mathbf{u} + \mathbf{C}_i \quad \text{for } i = 1, 2, \dots, N, \quad (3.4.2)$$

where  $\mathbf{A}_i = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}_i, \mathbf{u}_i)$ ,  $\mathbf{B}_i = \frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\mathbf{x}_i, \mathbf{u}_i)$  and  $\mathbf{C}_i = \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i)$ . The expressions given in (3.4.2) are used to approximate the plant dynamics for  $(\mathbf{x}, \mathbf{u})$  in  $S_i$  defined by

$$S_i = \{(\mathbf{x}, \mathbf{u}) : \|\mathbf{x} - \mathbf{x}_i\| < a_i, \|\mathbf{u} - \mathbf{u}_i\| < b_i\}, \quad (3.4.3)$$

where  $a_i$  and  $b_i$  are chosen from an eigenvalue sensitivity analysis to insure that each approximating linearization (3.4.2) is within an acceptable tolerance. The actual partition regions  $\{R_i\}$ ,  $i = 1, 2, \dots, N$  are related to the spherical regions  $\{S_i\}$  in (3.4.3) in the sense that  $\bigcup_{i=1}^N R_i = \bigcup_{i=1}^N S_i$ , but that the regions  $R_i$  are taken to be mutually dis-

joint. In practice, it is not necessary to consider the distinction between  $R_i$  and  $S_i$ .

From the set of linear approximations (3.4.2) equivalent discrete-time models are formulated. Thus, the set of approximations becomes

$$x(k+1) = \phi_i(T)x(k) + \psi_i(T)u(k) + \theta_i(T), \quad (3.4.4)$$

for  $i = 1, 2, \dots, N$  and where  $\phi_i(T) = \exp\{A_i T\}$ ,  $\psi_i(T) = \int_0^T \phi_i(t)B_i dt$  and  $\theta_i(T) = \int_0^T \phi_i(t)C_i dt$ . These equations (3.4.4) are equivalent to those of (3.4.2) at the sample times provided that the inputs  $u(t)$  are held constant during the (uniform) sample intervals, i.e. when  $u(t) = u(kT)$  for  $kT \leq t < kT + T$ . This is a reasonable assumption when digital control is to be used.

Since the design must consider the system's random disturbances, discrete noise processes will be incorporated into the approximating linear models. To do this the equations (3.4.4) are used in addition to an assumed measurement model which itself has been linearized about the points  $(x_i, u_i)$  as was done to obtain equations (3.4.2). The complete set of linearized models is given by

$$x(k+1) = \phi_i x(k) + \psi_i u(k) + \theta_i + w(k) \quad (3.4.5)$$

$$z(k) = H_i x(k) + v(k) \quad (3.4.6)$$

for  $(x(k), u(k))$  in region  $S_i$  and for  $i = 1, 2, \dots, N$ . The vectors  $w(k)$  and  $v(k)$  represent the effective plant disturbance and measurement noise, respectively.

The basic scheme for partitioning the nonlinear plant assumes a reasonably accurate nonlinear model. From this nonlinear model, variations of the open-loop eigenvalues are precomputed off line by calculat-

ing the eigenvalues for a multitude of linearized conditions. Once the key parameters are isolated, i.e. those to which the plant eigenvalues are most sensitive, eigenvalue variations are then plotted versus these parameters. Linear regions are then specified according to allowable tolerances in eigenvalue location.

The present approach, while certainly not limited to low-order plants, is most easily comprehended by developing a simple prototype example. To this end, consider the simple nonlinear oscillator [0-1]

$$\ddot{y} + \dot{y} + y^3 = u, \quad (3.4.7)$$

where  $u$  represents the system input containing both deterministic and stochastic terms, and  $y$  represents a scalar position variable.

A state model for (3.4.7) is given by

$$\dot{x}_1 = x_2 \quad (3.4.8)$$

$$\dot{x}_2 = x_2 - x_1^3 + u(t) \quad (3.4.9)$$

$$z_1 = x_1 + v_1 \quad (3.4.10)$$

$$z_2 = x_2 + v_2 \quad (3.4.11)$$

Equations (3.4.10) and (3.4.11) are the measured variables with  $v_1$  and  $v_2$  representing additive noise. If  $x_{10}$  is used as an expansion point, the linearized system becomes

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ -3x_{10}^2 & -1 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 & 0 \\ 1 & 2x_{10}^3 \end{bmatrix} \begin{bmatrix} u \\ 1 \end{bmatrix}. \quad (3.4.12)$$

Clearly, the linearized system dynamics depend on the value of the parameter  $x_{10}$ . By establishing an operating range for the system, say  $|x_1| \leq 6$ , the open-loop system eigenvalues vary as shown in Fig. 3.4.1.

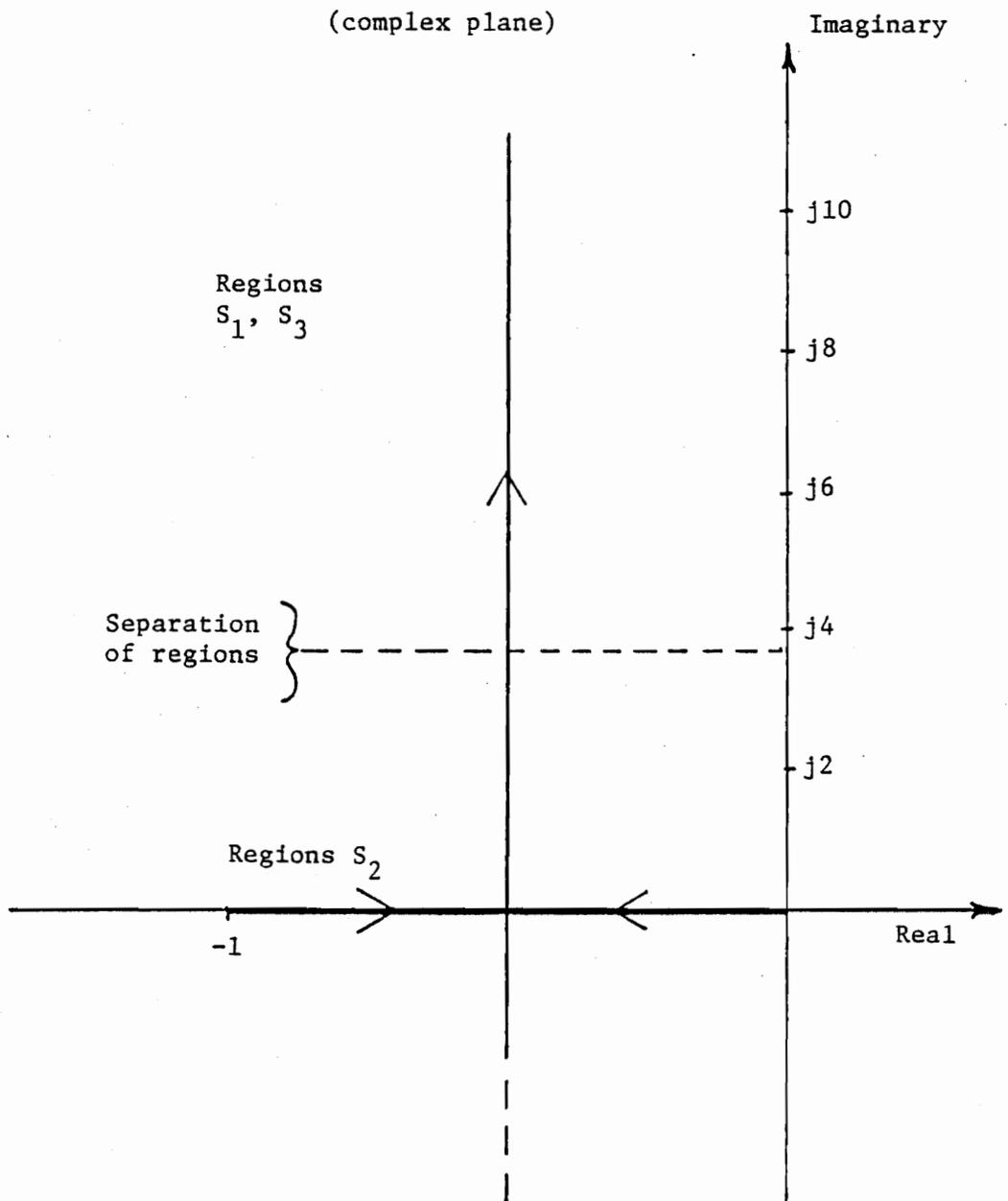


Figure 3.4.1 Variation in System Eigenvalues for the Design Example

Note that  $x_1 = 6$  represents a significant restoring force, corresponding to a linear spring constant of 108. Motivated by the open-loop eigenvalue variation, i.e. roughly equal eigenvalue variation, two intervals were taken; viz.

$$S_2 = \{\alpha: 0 \leq \alpha \leq 15\} \quad (3.4.13)$$

and

$$S_1 = \{\alpha: 15 \leq \alpha \leq 108\}, \quad (3.4.14)$$

where  $\alpha$  is the effective spring constant. With this partition three filters are effectively used, two differing only in the input matrix as a result of different bias offsets. The points of expansion for  $S_2$  and  $S_1$  and  $S_3$  were  $\alpha = 0$  and  $\alpha = 30$ , respectively.

Figure 3.4.2 illustrates the three linear approximations to the cubic nonlinear term. It is seen that the lines intersect at approximately  $x = 2.2$ , which verifies the boundary determined via eigenvalue variation (i.e. for region  $S_2$ ,  $|x_1| < \sqrt{5}$ ).

Inherent in the Kalman filter algorithm is the tendency of the filter gains to become small as time progresses. This occurs under conditions of large measurement noise versus small system noise. For known linear plants and their associated Kalman filters, this is usually not a problem. But for nonlinear plants, or for linear plants with unknown parameters, it is impossible to identically match the filter dynamics to those of the plant and a tendency toward divergence can be expected. The gains become small, little new measurement information is incorporated into the estimate and the filter values diverge from the actual plant values. A common practice is to artificially increase the intensity of the plant disturbance,  $Q$ , thus insuring that the filter gains

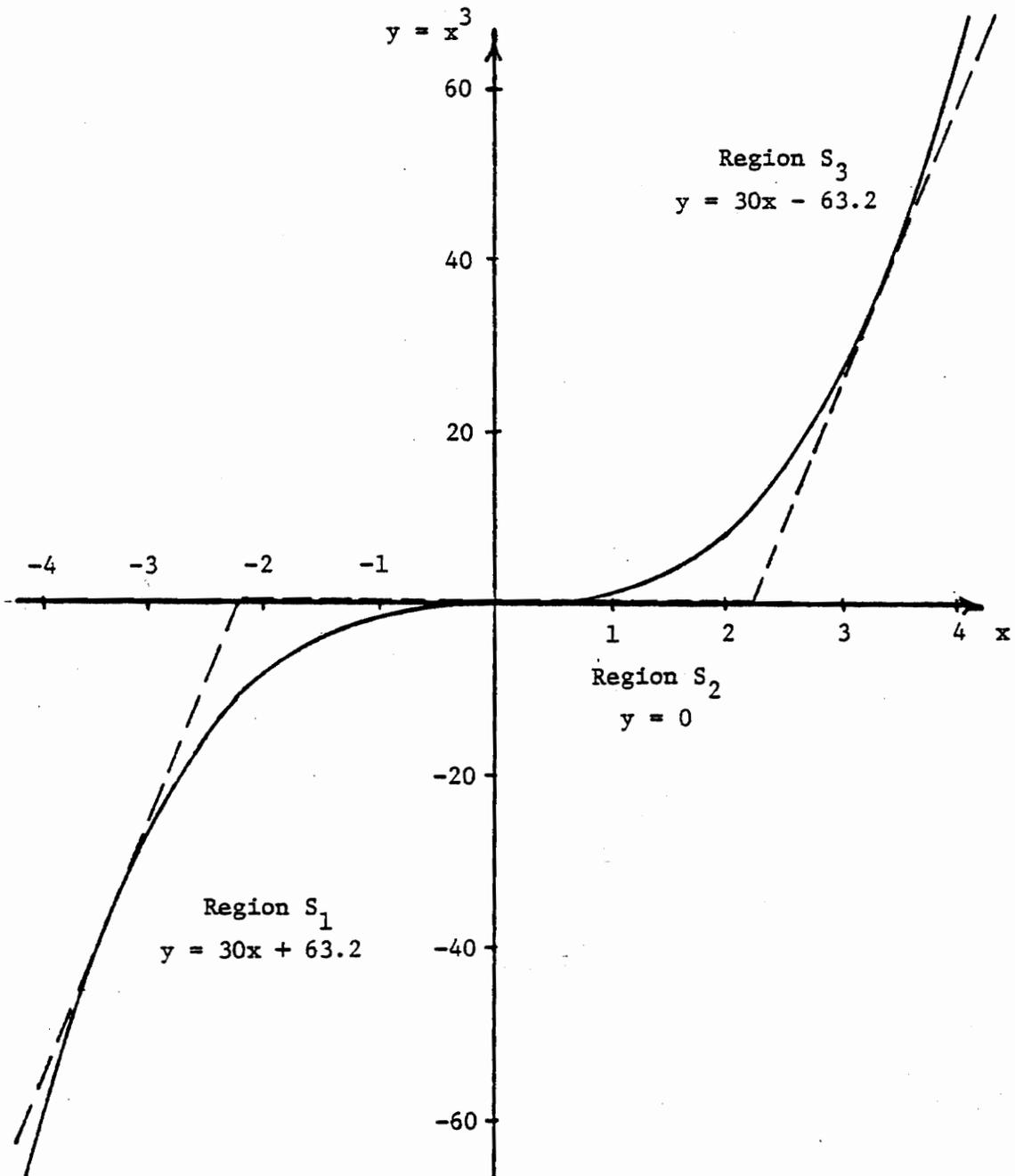


Figure 3.4.2 The Three Linear Approximations for the Cubic Nonlinearity

stay reasonably large. Usually, this is a trial and error process. A noteworthy exception is the bandwidth modulation technique developed by Jazwinski [J-1]. His method uses an additional covariance term that is adjusted sequentially by feeding back measurement residual information.

A more direct method has been devised specifically for the PAE algorithm. From Figure 3.4.2 one can quickly tabulate the error in  $x$  between the exact equation and the approximation for an arbitrary number of points in a given region. The average error for each region can then be calculated, i.e. the average error in regions  $s_1$  and  $s_3$  is .44, and in region  $s_2$  is 1.0. These errors are upper limits on the estimation accuracy in the various regions. In addition, the modeling error values aid in deciding how much additional  $Q$  should be added to the filter equations to maintain desired tracking performance. For example, the predicted error covariance equation, (3.2.11), can be augmented as,

$$M_i(k+1) = \phi_i P_i(k) \phi_i^T + \Gamma_i Q \Gamma_i^T + Q_m \quad (3.4.15)$$

where  $P$  is the error covariance,  $Q$  the system disturbance covariance and  $Q_m$  the additional modeling error covariance. The magnitude of  $Q_m$  is given approximately by the mean square value of error in  $x$ , i.e. for regions  $s_1$  and  $s_3$ ,  $Q \approx .3$ , and for region  $s_2$ ,  $Q \approx 1.5$ . The relatively large value in region  $s_2$  reflects the gross approximation in that region.

The modified PAE algorithm of this chapter was carried out on the previous nonlinear oscillator example for various levels of noise. The plant was simulated using the Jump-Matrix technique [V-2]. The state variable description of the oscillator is given in 3.4.8 and 3.4.9.

Defining  $x_3 = x_1^3$ , the Jump-Matrix description of the oscillator becomes

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{u} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -1 & -1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ u \end{bmatrix}_{t^+} ; \quad \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{u} \end{bmatrix}_{t^+} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ (\cdot)^3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ u \end{bmatrix}_{t^-} \quad (3.4.16)$$

These equations were executed at a sufficiently fast rate to keep the cubic term nearly constant from iteration to iteration, thus insuring an accurate nonlinear plant simulation.

The plant input, input disturbance covariance, measurement noise covariance and semi-Markov matrix are

$$u = 12; \quad Q = 4; \quad R = \begin{bmatrix} 3 & 0 \\ 0 & 10 \end{bmatrix}; \quad \theta = \begin{bmatrix} .996 & .002 & .002 \\ .002 & .996 & .002 \\ .002 & .002 & .996 \end{bmatrix} \quad (3.4.17)$$

Figure 3.4.3 shows the PAE algorithm tracking the plant very well over all three regions. The noisy regions near times  $t = .75$  and  $t = 1.5$  are due to the re-initialization of the Kalman filters when the nonlinear plant moves into a different region. Table 3.4.1 gives the weighting coefficient values at each iteration. The times that the estimator is re-initialized are shown by arrows. Notice that the weighting coefficients are then assigned equal probability values.

### 3.5 Summary

The nonlinear oscillator example studied in this chapter is a formidable estimation problem because of the rapid oscillations from one region to another. It has been shown that the PAE algorithm with switch detection can track the oscillator quite well. For less extreme

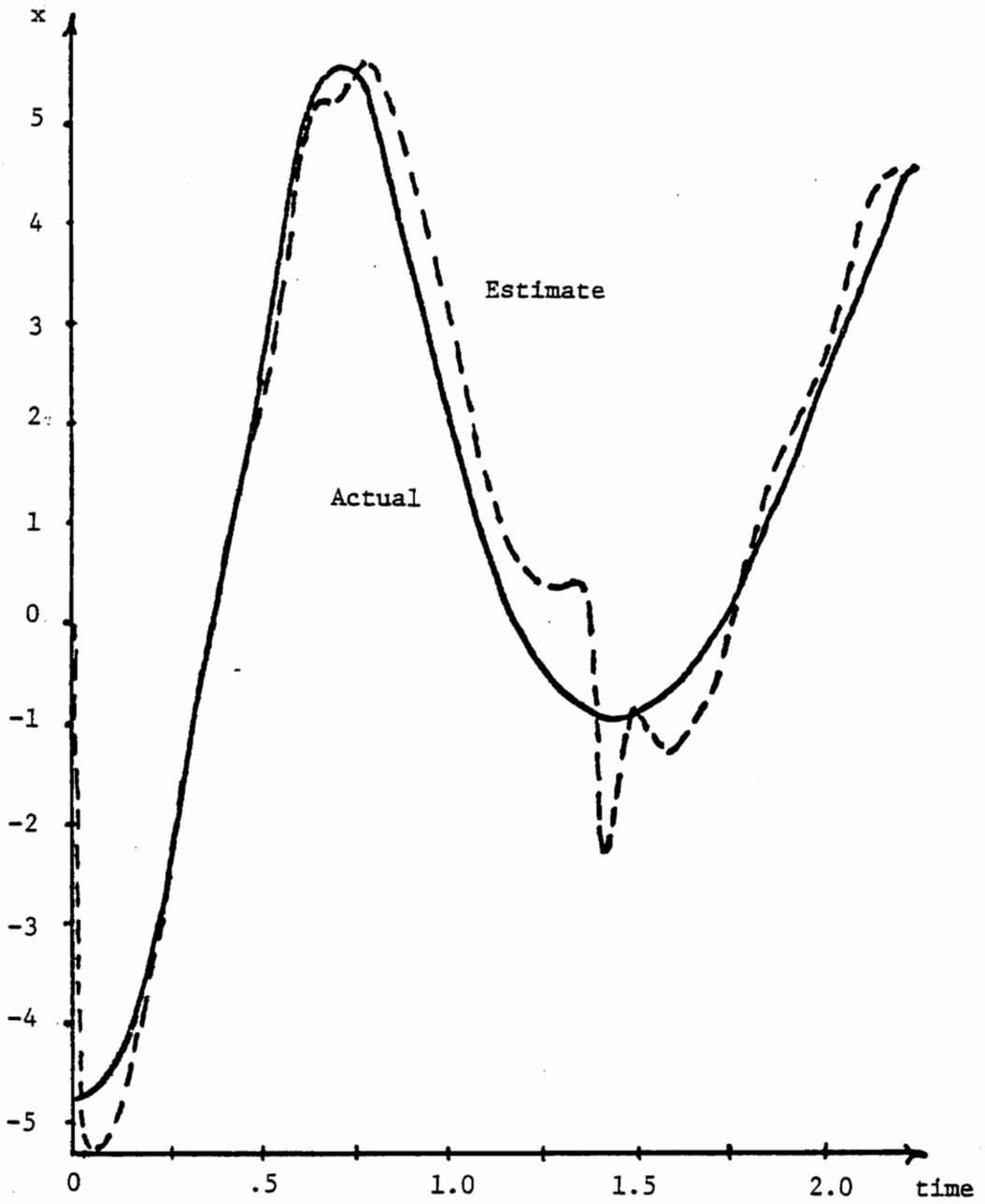


Figure 3.4.3 Actual Nonlinear Plant Position and the Modified PAE Position Estimate

Table 3.4.1 Weighting Coefficients for Figure 3.4.3

k	time	$\Pi_1$	$\Pi_2$	$\Pi_3$	$P_1(z)$	$P_2(z)$	$P_3(z)$
1	0.00E-01	3.33E-01	3.33E-01	3.33E-01	2.024E-04	1.865E-04	2.829E-05
2	5.00E-02	4.85E-01	4.87E-01	5.79E-01	1.324E-02	5.223E-03	1.004E-04
3	1.00E-01	7.32E-01	2.66E-01	7.65E-01	7.594E-03	4.151E-05	8.629E-04
4	1.50E-01	9.07E-01	1.99E-01	4.33E-01	1.749E-04	1.591E-06	5.831E-04
5	2.00E-01	9.99E-01	7.82E-01	1.749E-04	1.194E-02	3.929E-03	1.952E-08
6	2.50E-01	9.99E-01	6.59E-01	3.56E-01	1.207E-03	3.375E-03	1.444E-08
7	3.00E-01	9.97E-01	2.80E-01	9.02E-01	2.751E-03	3.729E-05	1.087E-03
8	3.50E-01	9.99E-01	6.52E-01	7.95E-01	2.751E-03	3.729E-05	1.087E-03
9	4.00E-01	9.94E-01	1.53E-01	8.30E-01	6.123E-03	4.532E-03	1.616E-06
10	4.50E-01	9.97E-01	2.91E-01	2.83E-01	1.340E-02	1.138E-02	1.950E-05
11	5.00E-01	3.33E-01	3.33E-01	3.33E-01	1.922E-06	4.68E-03	7.477E-06
12	5.50E-01	8.52E-01	5.41E-01	4.50E-01	2.404E-05	1.512E-03	1.258E-03
13	6.00E-01	7.35E-01	6.94E-01	3.05E-01	3.110E-05	5.764E-03	3.043E-03
14	6.50E-01	3.29E-01	7.20E-01	2.78E-01	8.994E-07	5.937E-03	5.214E-03
15	7.00E-01	1.13E-01	5.21E-01	9.97E-01	2.735E-04	3.512E-05	1.722E-02
16	7.50E-01	6.13E-01	1.02E-01	9.38E-01	2.314E-03	1.140E-14	7.562E-05
17	8.00E-01	3.72E-01	1.11E-01	1.00E-01	9.672E-06	8.958E-11	1.717E-02
18	8.50E-01	5.42E-01	1.06E-01	1.00E-01	3.254E-06	7.143E-19	1.320E-03
19	9.00E-01	8.35E-01	1.00E-01	9.99E-01	9.521E-04	1.143E-20	2.289E-03
20	9.50E-01	9.02E-01	6.04E-01	9.99E-01	3.460E-04	3.281E-19	1.090E-03
21	1.00E-00	2.94E-01	1.76E-01	1.00E-01	1.449E-05	1.255E-17	1.433E-02
22	1.05E-00	4.67E-01	1.14E-01	1.00E-01	8.345E-05	2.097E-17	3.666E-03
23	1.10E-00	5.78E-01	2.54E-01	9.99E-01	1.381E-04	6.205E-18	4.904E-03
24	1.15E-00	2.11E-01	1.08E-01	9.97E-01	9.951E-04	5.246E-18	9.694E-04
25	1.20E-00	3.21E-01	2.35E-01	9.99E-01	2.444E-04	3.012E-14	2.573E-03
26	1.25E-00	8.18E-01	3.08E-01	1.00E-01	3.152E-07	1.419E-09	9.247E-03
27	1.30E-00	5.11E-01	2.27E-01	9.94E-01	8.784E-04	3.873E-12	3.403E-04
28	1.35E-00	1.11E-01	1.40E-01	1.00E-01	3.014E-06	1.367E-05	1.966E-02
29	1.40E-00	3.33E-01	3.33E-01	3.33E-01	2.640E-03	6.209E-12	2.176E-06
30	1.45E-00	6.30E-01	3.67E-01	2.11E-01	3.677E-05	2.144E-05	1.235E-07
31	1.50E-00	8.22E-01	1.77E-01	2.34E-01	1.453E-02	5.358E-03	6.343E-04
32	1.55E-00	6.76E-01	3.23E-01	1.17E-01	6.402E-03	1.848E-02	5.355E-04
33	1.60E-00	5.19E-01	4.86E-01	6.46E-01	4.453E-03	8.781E-03	1.784E-05
34	1.65E-00	7.31E-01	2.68E-01	1.09E-01	1.284E-02	4.987E-03	4.904E-06
35	1.70E-00	6.14E-01	3.85E-01	1.78E-01	1.232E-02	2.090E-02	1.303E-05
36	1.75E-00	1.49E-01	6.37E-01	1.31E-01	1.134E-04	1.016E-03	3.059E-03
37	1.80E-00	4.34E-01	0.937E-01	1.29E-01	5.095E-04	2.096E-02	2.258E-03
38	1.85E-00	4.18E-01	9.95E-01	4.64E-01	5.668E-05	8.604E-03	1.015E-02
39	1.90E-00	5.16E-01	9.98E-01	1.29E-01	5.337E-06	2.084E-02	4.054E-03
40	1.95E-00	1.56E-01	9.99E-01	6.95E-01	2.257E-05	2.316E-02	4.872E-04

nonlinear systems the tracking should be even better. In fact, if the plant time constants are larger than the convergence times of the filters, then the switch detection part of the estimation algorithm can be deleted with little loss in overall tracking accuracy. It should be pointed out, however, that this nonlinear estimation technique is not intended to compete with the current methods in terms of accuracy. Rather, it is a fairly straightforward approach with significant advantages in implementation that can be applied to a large class of nonlinear equations.

This method of nonlinear estimation is readily implemented in a parallel processing arrangement. Each Kalman filter can act simultaneously on the measurement data (rather than consecutively, as was done in the simulations). A supervisory computer would then weight the individual processor outputs (state estimates) to derive the best nonlinear estimate. Inherent in this implementation is the ease of adding system redundancy in the form of additional parallel processors. Greater accuracy is attainable by partitioning the plant into a larger density of linear approximations. And if the parallel processing implementation is used, this increase in accuracy will not significantly add to computation time. Each Kalman filter, being linear, also allows off-line precalculation of Kalman gains, thus eliminating much of the on-line computational burden.

Several disadvantages, however, subtract from the general utility of this method. First, the switch detection performance is quite sensitive to the threshold value. If the filters are re-initialized too often, then the resulting estimate will be unacceptably noisy. On the

other hand, a threshold value that is too low will allow too much divergence. Unfortunately, the only sure way to achieve a satisfactory threshold value is by trial and error. A value that works well for one problem may be unsatisfactory for another.

A second disadvantage is the noise that is present in the estimate during re-initialization. This degradation is due to the inherent inability of the method to track rapid configuration changes. In Chapter Five an estimator will be developed to remedy this problem.

## 4.0 STOCHASTIC NONLINEAR CONTROL

### 4.1 Introduction

This chapter applies the modified PAE algorithm of Chapter Three to the stochastic nonlinear control problem. A major portion is devoted to the development of a design procedure for a specific type of problem referred to as set-point control. This procedure will be applied to the nonlinear oscillator introduced in Chapter Three. Referring to the terminology of Chapter One, the nonlinear controller to be developed is an adaptive controller. Use is made of the Separation-Principle to subdivide the identification and the control aspects of the design. In addition, the controller is "cautious" in that the feedback gains depend on the degree of uncertainty in the state estimates. Before discussing the advantages of the controller, a brief description of the hierarchy of control problems and a summary of existing stochastic nonlinear controllers is presented.

The methods of deterministic linear control are by far the most completely developed of all the control problems. Given the plant dynamics, the problem is to determine the feedback gains that will give the plant the desired closed loop performance. The feedback gains can be found using classical, modern or optimal techniques. The separation principle allows the state variable feedback to be determined, assuming that all the states are available, even though a state reconstructor (observer) might be required.

Stochastic linear control theory is also very well developed, in the context of the linear-quadratic-Gaussian, LQG, problem. That is,

a linear plant having Gaussian distributed random processes with the performance index expressed as a quadratic cost functional. The separation principle, which holds under these restrictions, allows a Kalman filter observer to be designed independently of the feedback gains.

There are no general methods for designing a controller for a deterministic nonlinear problem. Describing functions have been applied successfully to a class of problems, primarily relay type switching functions. The method of input matching [J-3] is effective for controlling plants that can be described using a generalized input/output model. For most problems, however, the usual approach is to linearize the function around some operating point and then apply linear control theory. This method works well if the state of the plant remains in a neighborhood of the operating point. But, poor performance and possible instability may result if the state is not so contained. Oaks and Cook [O-1] developed a design method for driving a nonlinear plant from one operating point to another. Their method required the determination of regions of stability around the desired and intermediate operating points by solving Liapunov stability equations. By overlapping these regions, optimal trajectories were determined to connect the initial, intermediate and final points. Although elegant, the computational burden of their method, even for low order problems, is excessive.

The stochastic nonlinear control problem has all the difficulties of the deterministic problem plus the complication of random system disturbances and measurement noise. The separation principle is not, in general, valid for this problem. There are basically two approaches of stochastic nonlinear control. The first method imposes the separation

principle, valid or not, and then employs a nonlinear observer such as the EKF, to reconstruct the state vector for feedback purposes. If a single feedback matrix is designed to give suitable performance for all plant conditions, the controller is said to be "Robust" [D-2]. Alternatively, a set of feedback gain matrices can be calculated based upon different linearizations of the nonlinear plant. These gains can then be scheduled according to the state estimates to give more precise control in the different regions. Gain scheduling is probably the most widely used method.

A second approach is to use adaptive techniques to learn key parameters in the plant and formulate the control law accordingly. These nonlinear adaptive control methods are motivated by the observation that a nonlinear plant can be thought of as a linear plant having unknown, time-varying parameters. The adaptive approach has several advantages over the linearization approach. First, the plant description is often difficult to ascertain. The linearization of a nonlinear plant as required by the EKF and by the feedback calculations cannot be carried out effectively unless the plant equations are accurately known. On the other hand, the adaptive method is quite tolerant of plant model inaccuracies. Second, gain scheduling per se does not take into account uncertainties in the state estimates. Using an inappropriate gain because of noise in the estimate could very possibly cause an unstable situation. Adaptive controllers can be designed to reduce this possibility.

The motivation for using the PAE algorithm in closed loop control is attributed to Deshpande et al [D-1]. It was shown that for unknown

model parameters the separation principle does not generally hold, and that the optimal stochastic controller necessarily has a nonlinear structure. The "Partitioned Adaptive Controller", PAC, was then motivated by arguments centered around minimizing a cost functional. Similar approaches to the adaptive control problem were investigated by Saridis and Dao [S-5], and Stein and Saridis [S-6]. Athans et al successfully applied an equivalent technique, called "Multiple Model Adaptive Control" to the F-8C aircraft [A-1].

The shortcomings of the PAC formulation are the same as those of the PAE algorithm discussed in Chapter Three. The main problem being the inability to track, hence control, a plant undergoing rapid configuration changes. However, the modified PAE algorithm with semi-Markov plant modeling and gain re-initialization eliminates this problem.

In this chapter a modified PAC controller based upon the results of Chapter Three will be applied to the stochastic nonlinear control problem. Section 4.2 describes the basic structure of the modified PAC controller. A specific control problem, set-point control, is solved in Section 4.3. The nonlinear oscillator considered by Oaks and Cook is used as a design example. Section 4.4 gives the results of several simulations of the modified PAC controller. The advantages and disadvantages of this method of stochastic nonlinear control are discussed in Section 4.5.

## 4.2 The Modified PAC Controller

The structure of the modified PAC algorithm is shown in Figure 4.2.1. A bank of Kalman filters operates in parallel and independently

on the noisy plant measurements,  $z$ . Each filter produces an estimator,  $\hat{x}_i$ , for the state conditioned on the plant being matched to that particular linearized model,  $s_i$ . Unlike the PAE algorithm, however, a weighted sum state estimate is not calculated. Instead, each estimate is multiplied by feedback gains,  $F_i$ , which are particular to each linearized model. The overall state variable feedback is then calculated as a weighted sum of the elemental state variable feedbacks,  $u_i$ . The weighting coefficients,  $\Pi_i$ , are calculated in exactly the same manner as they were for the PAE algorithm. Thus, they are nonlinear functions of the measurements. In addition to the individual feedback gains, each model will in general have a different linearization constant as seen in equation (3.4.2). Depending on the control function to be implemented, these biases may have to be eliminated by subtracting the off-set values,  $H_i$ . For uniform steady state performance, the actual input,  $r$ , may require different scale factors,  $G_i$ , for the different linearized models. The net plant input is a weighted sum of all three inputs; actual scaled input, input off-sets and feedback inputs.

This structure is intuitively satisfying. As long as the nonlinear plant is in one particular region, the filter matched to that region and the feedback gains for the corresponding linear approximation are being used in a straightforward state variable feedback control scheme. When the plant moves to another region, then the weighting coefficients will automatically update to identify the best matched Kalman filter.

Because of the digital computer in the feedback loop, the controller has a sampled data control structure. Moreover, the measurements themselves may be available only at discrete times, such as a radar

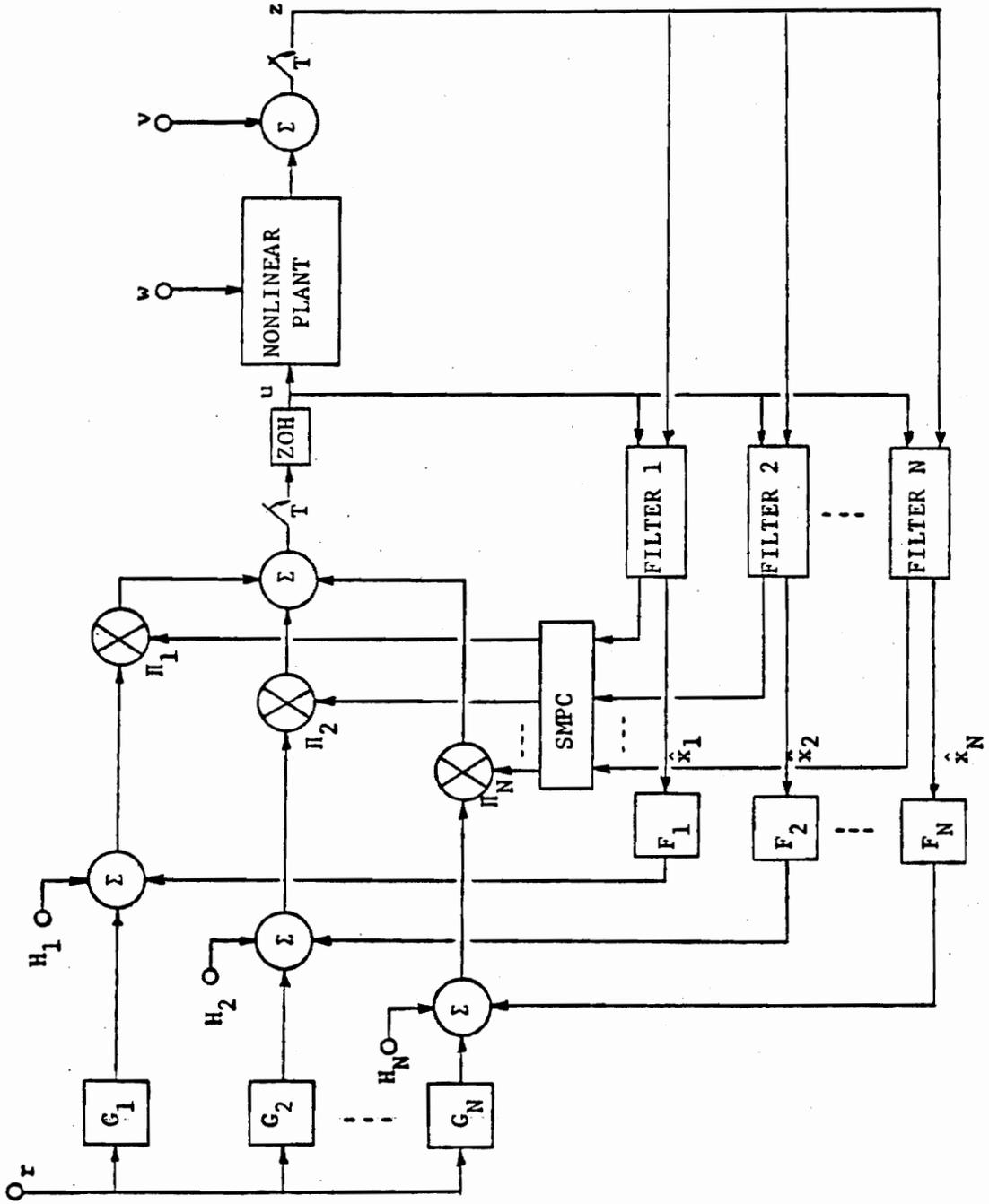


Figure 4.2.1 Modified Partitioned Adaptive Controller

signal. The continuous time linearization models must therefore be converted to discrete time models. From the discrete time models, the required sampled data feedback gains can be determined. A variety of techniques can be used to obtain the feedback gains. Classical criteria for the transient response, such as rise-time, overshoot, etc. can be specified, thus locating the closed-loop eigenvalues. In general, each linearized region could have a different desired performance, and therefore different eigenvalues. For larger order systems an optimal control formulation is much more versatile. Both techniques are reviewed in Chapter Two.

### 4.3 Set Point Control

An important class of tracking problems consists of those problems in which the reference variables remain constant over periods of time which are long relative to desired response times. For such systems it is customary to call the constant reference variable the *set-point* for the system. The specific application of this section is that of set-point control of nonlinear stochastic systems by which is meant the design of a suitable controller for the nonlinear plant so that the controlled output responds to step reference inputs in a desired manner. The two classical considerations for such a response are the transient and steady-state behavior. To the extent that these concepts carry over to nonlinear stochastic control systems, the purpose of the present design will be to achieve 'zero' steady-state error and a prespecified rise-time for the controlled response. Of course, the actual error will never be identically zero because of the disturbance process which is

driving the system in a random fashion.

Once the configuration regions have been established, following the linearization procedure described in Chapter Three, the next step in the controller design is to construct a set of feedback gains. It is desirable to maintain a uniform closed-loop system transient response. This is accomplished by computing the individual feedback gains  $F_i$ ;  $i = 1, 2, \dots, N$  to give the same closed-loop eigenvalues.

In order to complete the controller design, it is necessary to compensate for the offset terms given by the matrices  $\theta_i$  in (3.4.5). In addition, designing for a classical type-1 response generally requires the inclusion of input gains  $G_i$  as shown in Figure 4.2.1, i.e. the reference input must be weighted differently for each region to achieve a zero average steady-state error to step commands. For example, let the continuous time plant description in region  $s_i$  be

$$\begin{aligned}\dot{x} &= A_i x + B_i u + C_i \\ &= A_i x + [B_i C_i] \begin{bmatrix} u \\ 1 \end{bmatrix} \\ &= A_i x + \tilde{B}_i \begin{bmatrix} u \\ 1 \end{bmatrix}\end{aligned}\quad (4.3.1)$$

The discrete time equivalent is

$$x(k+1) = \phi_i x(k) + [\psi_{i1} \psi_{i2}] \begin{bmatrix} u \\ 1 \end{bmatrix}\quad (4.3.2)$$

If  $u = F_i x(k+1) + G_i r + H_i$ , then the closed loop system is

$$x(k+1) = [\phi_i + \psi_{i1} F_i] x(k) + [\psi_{i1} \psi_{i2}] \begin{bmatrix} G_i r + H_i \\ 1 \end{bmatrix}$$

If the reference input,  $r$ , is held constant at a value  $r_0$ , and if the closed loop system is stable, then  $x$  will reach a steady state value given by

$$x_{ss} = [(I - \phi_i - \psi_{i1} F_i) \psi_{i1}] G_i r_o + [(I - \phi_i - \psi_{i1} F_i) \psi_{i1}] H_i + [(I - \phi_i - \psi_{i1} F_i) \psi_{i2}] \quad (4.3.4)$$

The unknown scale factor,  $G_i$ , and offset correction,  $H_i$ , can be solved for the zero steady state error case,  $x_{ss} = r_o$ . The result is

$$G_i = [(I - \phi_i - \psi_{i1} F_i) \psi_{i1}]^{-1}; H_i = G_i [(I - \phi_i - \psi_{i1} F_i) \psi_{i2}] \quad (4.3.5)$$

The modified PAC algorithm will be applied to the nonlinear oscillator of Chapter Three. It is desired to achieve set-point control of

$$\ddot{y} + \dot{y} + y^3 = u \quad (4.3.6)$$

so that the closed loop performance matches that of the linear, critically damped system,

$$\ddot{y} + 20\dot{y} + 100y = u \quad (4.3.7)$$

This corresponds to a double pole at  $s = -10$ , which in turn yields a rise time of approximately 0.4 seconds. In addition, it is desired to have classical type-1 steady state performance, i.e. zero tracking error for step inputs. The following steps outline the design procedure.

- Step 1. Decide on the operational limits of the state variables associated with the nonlinearities. As in Chapter Three, it is assumed that  $-6 < x < 6$ . Later this range will be extended to  $-8 < x < 8$ .
- Step 2. Decide on the number of linear approximations required to represent the nonlinear differential equation. The number and selection of linear approximations depends primarily on two requirements. From an estimation point of view the Taylor series expansion points should be chosen close enough to insure good accuracy. From a closed loop viewpoint it is necessary that the feedback gains calculated from the linear approx-

imation give suitable closed loop behavior over the range of the approximation. Moreover, in set-point control it is necessary that each region sufficiently overlap the adjacent regions so that the actual steady state of the nonlinear plant, using the feedbacks of an intermediate region, will also lie in the desired region. This requirement insures the controller's ability to drive the state from region to region, and is similar to the overlapping regions of stability constraint discussed in [0-1]. Fortunately, the control requirements are usually satisfied if the linearization regions are chosen solely for estimation accuracy. The three linearized systems of Chapter Three will be used in this design example and are repeated below.

$$S_1 \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -30 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & -63.2 \end{bmatrix} \begin{bmatrix} u \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w$$

$$S_2 \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w$$

$$S_3 \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -30 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 63.2 \end{bmatrix} \begin{bmatrix} u \\ 1 \end{bmatrix} + \begin{bmatrix} u \\ 1 \end{bmatrix} w$$

Step 3. Discretize the continuous time linear models, for a sample time of .07 seconds.

$$S_1 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{k+1} = \begin{bmatrix} .929 & .066 \\ -1.979 & .863 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_k + \begin{bmatrix} .0024 & -.142 \\ .066 & -3.96 \end{bmatrix} \begin{bmatrix} u \\ 1 \end{bmatrix} + \begin{bmatrix} .0024 \\ .066 \end{bmatrix} w$$

$$S_2 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{k+1} = \begin{bmatrix} 1. & .068 \\ 0. & .932 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_k + \begin{bmatrix} .0024 & 0. \\ .068 & 0. \end{bmatrix} \begin{bmatrix} u \\ 1 \end{bmatrix} + \begin{bmatrix} .0024 \\ .068 \end{bmatrix} w$$

$$S_3 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{k+1} = \begin{bmatrix} .979 & .066 \\ 1.979 & .863 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_k + \begin{bmatrix} .0024 & +.142 \\ .066 & +3.96 \end{bmatrix} \begin{bmatrix} u \\ 1 \end{bmatrix} + \begin{bmatrix} .0024 \\ .066 \end{bmatrix} w$$

Step 4. Using the SVF method outlined in Section 3.2, calculate the sampled data feedback gains for each region. First, the desired continuous time poles  $s = -10, -10$ , transform into the discrete time poles  $z = .5, .5$ . The feedback gains are then

$$\begin{aligned} F_1 &= [-24.2 & -11.2] \\ F_2 &= [-53.6 & -12.0] \\ F_3 &= [-24.2 & -11.2] \end{aligned} \quad (4.3.10)$$

Step 5. Calculate the required input scaling factors and offsets for each region to achieve zero steady state error, using equation (4.3.5).

$$\begin{aligned} G_1 &= 54.5 & H_1 &= 63.2 \\ G_2 &= 53.6 & H_2 &= 0.0 \\ G_3 &= 54.5 & H_3 &= -63.2 \end{aligned} \quad (4.3.11)$$

Step 6. Decide on the semi-Markov matrix entries. The approach taken will be to obtain a balance between noise immunity and estimator responsiveness. This will be done empirically. An analytical derivation of the entries, as done in reference [M-5], is not possible because the Markov transitions between regions and the holding times are very dependent on both the reference input and on the disturbance inputs.

After trying several different matrices, the following semi-Markov matrix was chosen.

$$\theta = \begin{bmatrix} .996 & .002 & .002 \\ .002 & .996 & .002 \\ .002 & .002 & .996 \end{bmatrix}$$

The results of Steps Four and Five are summarized in Table 4.3.1.

#### 4.4 Simulations

In this section several simulation results are presented on the set-point control of the nonlinear oscillator. A nonlinear integration routine was used for the plant [V-2]. Otherwise, the simulation follows the adaptive estimation and control algorithms discussed in the previous sections and in Chapter Three. Figure 4.4.1 provides a plot of the oscillator response to a reference step of 4 units, the amplitude 4 insuring a strongly nonlinear response. Note the nonlinear effect of the oscillation period's dependence on the amplitude. Several closed-loop simulations were made with the three design regions and associated parameters given in Table 4.3.1. The three regions in Figure 4.4.1 are dependent only on position; regions  $S_1$ ,  $S_2$  and  $S_3$  are -6 to -2.2, -2.2 to 2.2 and 2.2 to 6, respectively. Figure 4.4.2 illustrates both the closed-loop system response to a reference input step of 4 units amplitude, and the relative tracking behavior of the three filters. Note that the proper filter for region 3 maintains a close track on the plant trajectory. The initial plant state was taken to be  $[-4.8 \ 0]^T$ , the input covariance  $Q = 1$ , the measurement noise covariance  $R$  and semi-

Table 4.3.1 Design Parameters for the Set-Point Control Example

discrete time feedback gains	$F_i$	offset correction $H_i$	feedforward gains $G_i$
Region - 1	-24.2, -11.2	63.2	54.5
Region - 2	-53.6, -12.0	0	53.6
Region - 3	-24.2, -11.2	-63.2	54.5

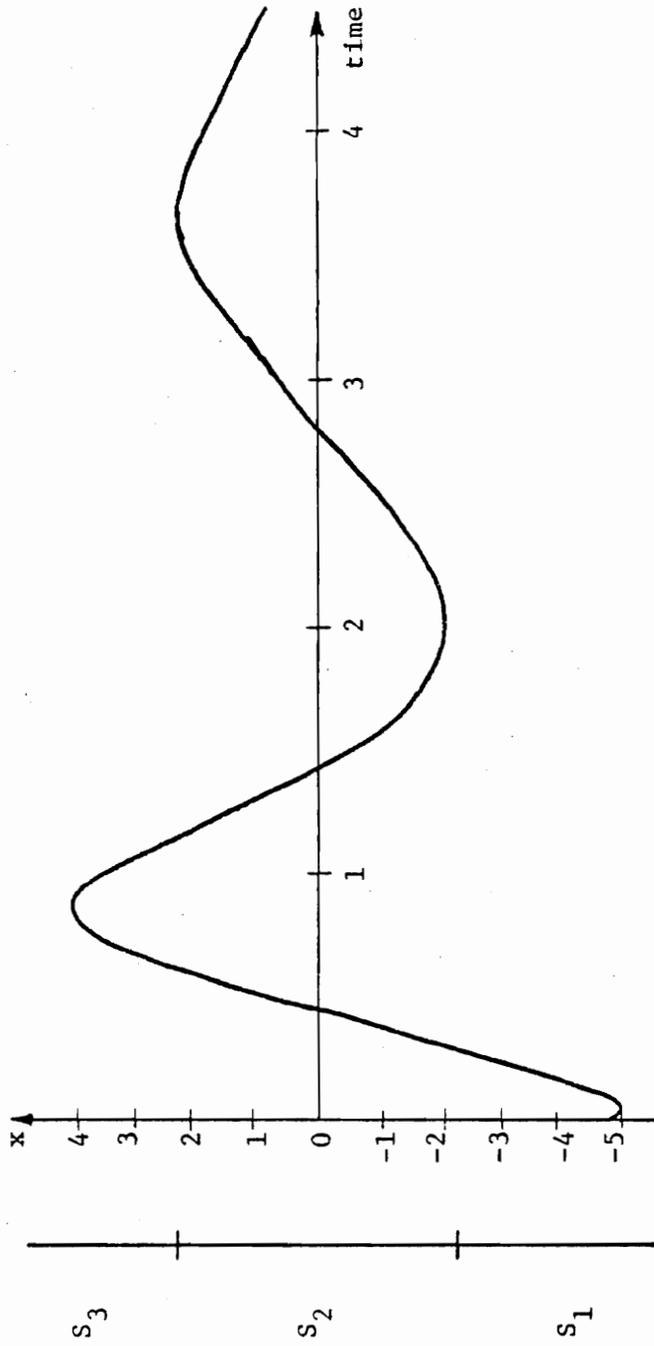


Figure 4.4.1.1 Open-Loop Response To A Set-Point of 4 Units

Markov matrix are

$$R = \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix} \quad \theta = \begin{bmatrix} .996 & .002 & .002 \\ .002 & .996 & .002 \\ .002 & .002 & .996 \end{bmatrix} \quad (4.4.1)$$

For convenience, this form of  $\theta$  will be denoted as

$$\theta = \{\text{diagonal } .996, \text{ off-diagonal } .002\}$$

The actual position response closely follows the design goal of a rise time of 0.4 seconds.

The effect of increasing the measurement noise is presented in Figure 4.4.3. Only  $R$  has been changed over the conditions of Figure 4.4.2. A high degree of settling is seen even with large measurement errors present.

In Figure 4.4.4 the initial set-point is again  $R = 4$ , but after 2.8 seconds the set-point is changed to  $-3$ . For this simulation  $Q = 4$ ;  $R = \text{diagonal } \{3 \quad 10\}$ ;  $\theta = \{\text{diagonal } .99, \text{ off-diagonal } .005\}$ . Here the semi-Markov matrix has been modified to be more sensitive to configuration changes. It is interesting to follow the plant weighting coefficients shown in Figure 4.4.4b. Initially, filter 1 provides good estimates because the plant position begins at  $-4.8$ . In response to the set-point of 4, the position increases through region  $S_2$  and into region  $S_3$ . At 2.8 seconds the set-point switches to  $-3$  and after a learning time of approximately 0.8 seconds region  $S_1$  is recognized. The overshoot at this time (Figure 4.4.4a) is due to the learning time of the controller. It is seen that the weighting coefficient for filter 3 is dominant until about  $t = 3.7$  seconds, indicating that the controller is using the feedback and input gains based on regions  $S_3$ , when in

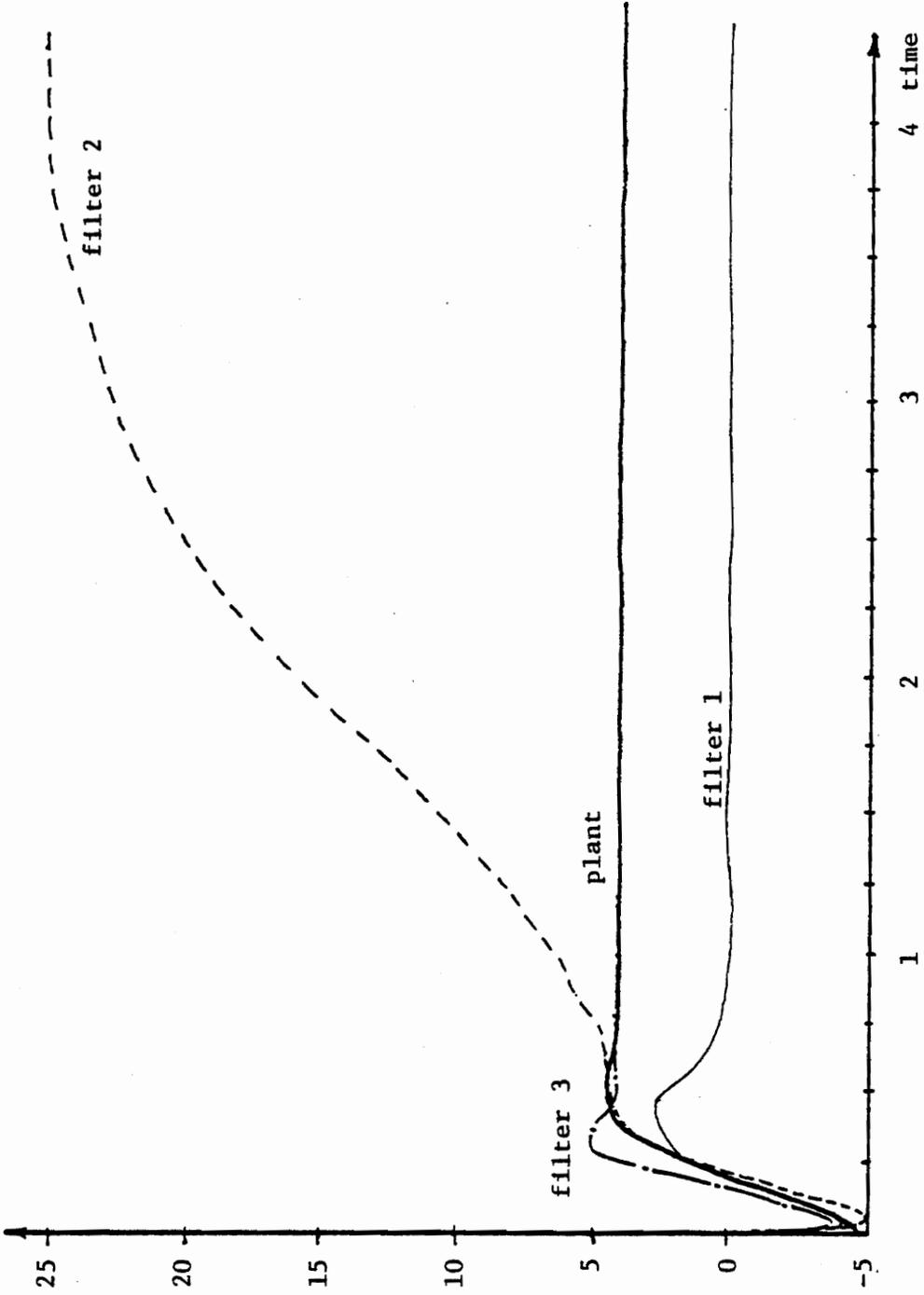


Figure 4.4.2 Closed-Loop Response of Nonlinear Oscillator To A Set-Point of 4 Units.

Also Shown Are The Three Kalman Filter Estimates.

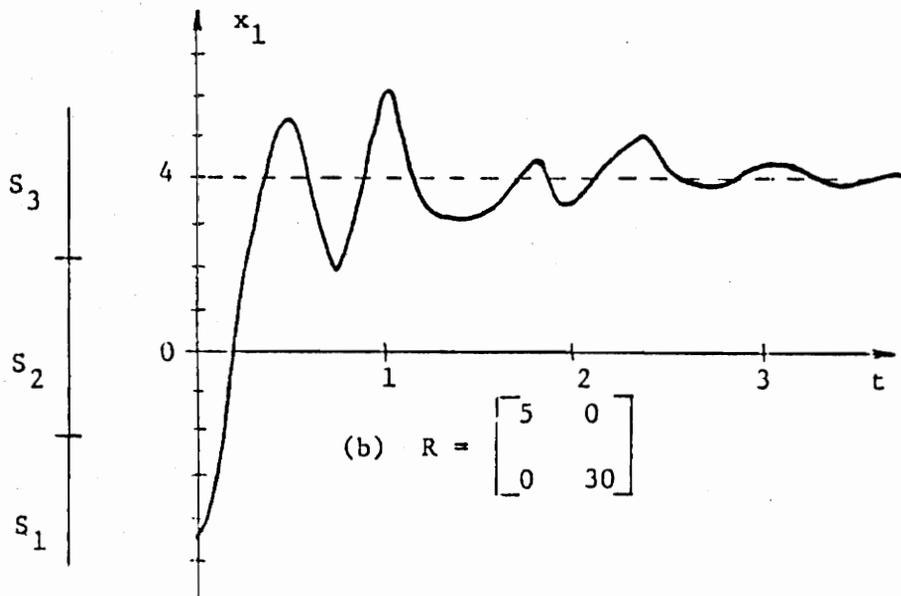
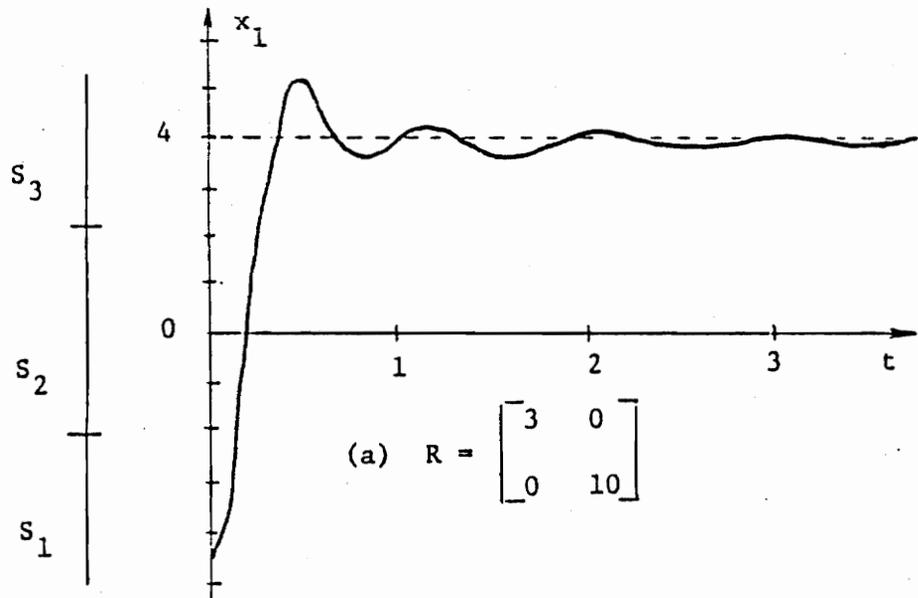


Figure 4.4.3 Closed-Loop Step Response with Varying Amounts of Measurement Noise

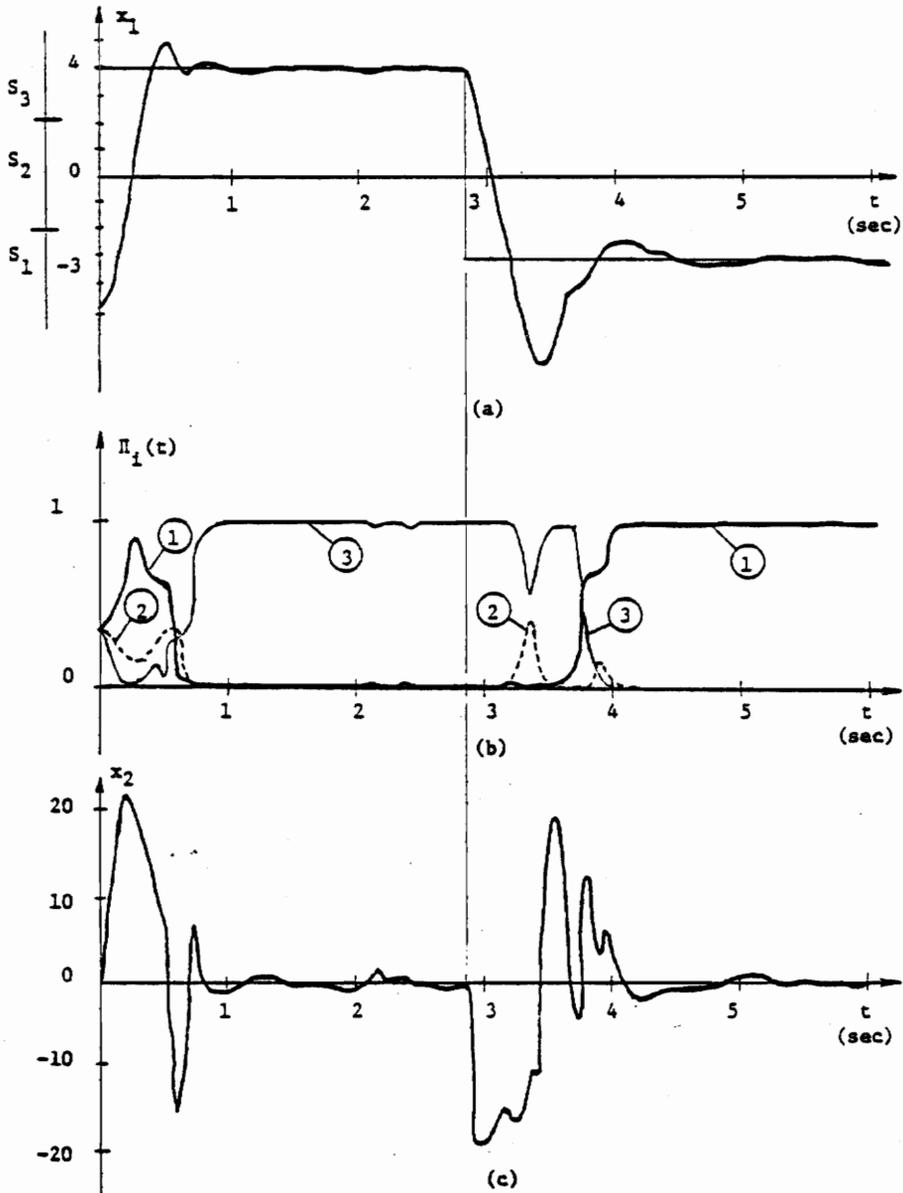


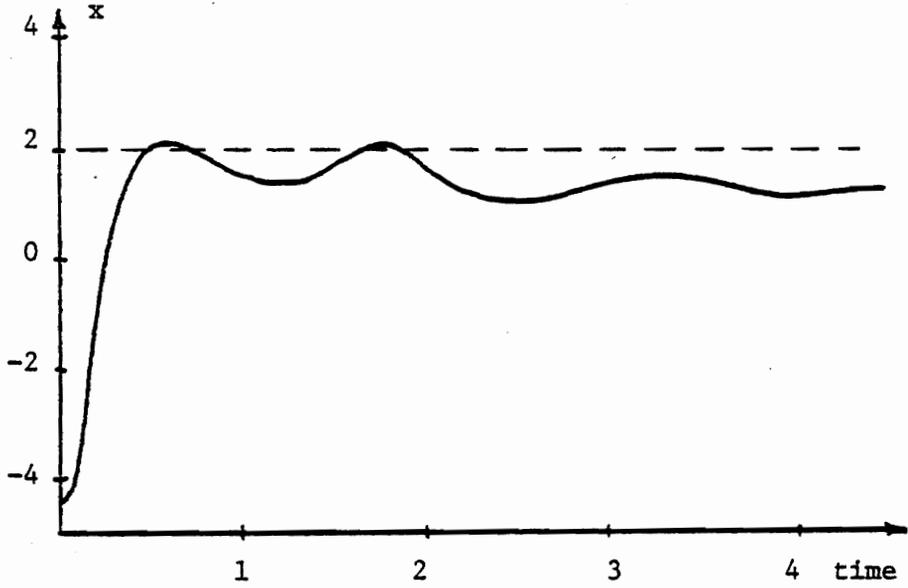
Figure 4.4.4 (a) Position Response to Set-Point Changes;  
 (b) Corresponding Time-Varying Probabilities  $\Pi_i$   $i = 1, 2, 3$  and  
 (c) Velocity Response.

actuality the plant is moving through region  $S_2$  and into region  $S_1$ . This excessive compensation causes the overshoot. The learning time can be reduced by increasing the off-diagonal entries of the semi-Markov matrix. However, this makes the controller more susceptible to noise, as discussed earlier. Figure 4.4.4b is a plot of plant velocity versus time. It is seen that the velocity does in fact approach zero when the position settles out to the input value.

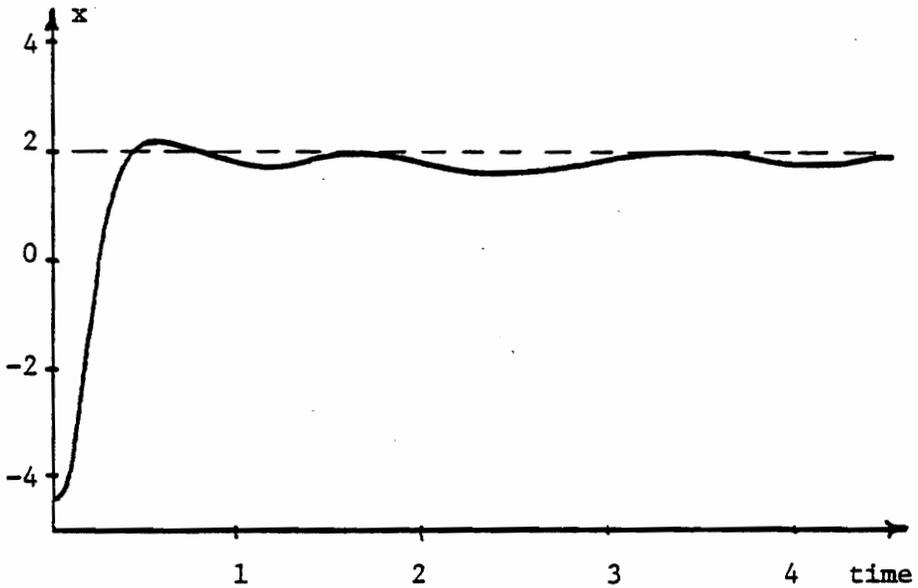
The question arises as to how effective will the control be when the set-point is between regions. For example, an input of  $u = 2.0$  attempts to drive the plant between regions  $S_2$  and  $S_3$ , so that neither filter 2 nor 3 will be giving good estimates. The result, Figure 4.4.5a, shows both poor transient response and steady state response, i.e., the position settles out to  $x = 1.3$ .

Better performance between regions can be achieved by altering the linear approximation method in one or more regions. The original method calls for a Taylor series expansion at the center of a region (a slope approximation). A different approach is to use a chord approximation. For example, the cubic nonlinearity in region  $S_2$  was originally modeled as a line having zero slope. It was determined graphically that a chord having a slope of 2.5 gives less overall modeling error throughout the region. Rematching filter 2 to this new linear approximation gives the improved results shown in Figure 4.4.5b. The plant position settles out faster to a reasonably close value of  $x = 1.8$ .

Attempting to drive the plant outside the design range gives understandably poorer results. Figure 4.4.6 shows the position response to a



(a) Original Taylor Series Approximation.



(b) Chord Approximation.

Figure 4.4.5 Set-Point Control for  $u = 2.0$  Using Different Approximations in Region  $S_2$ .

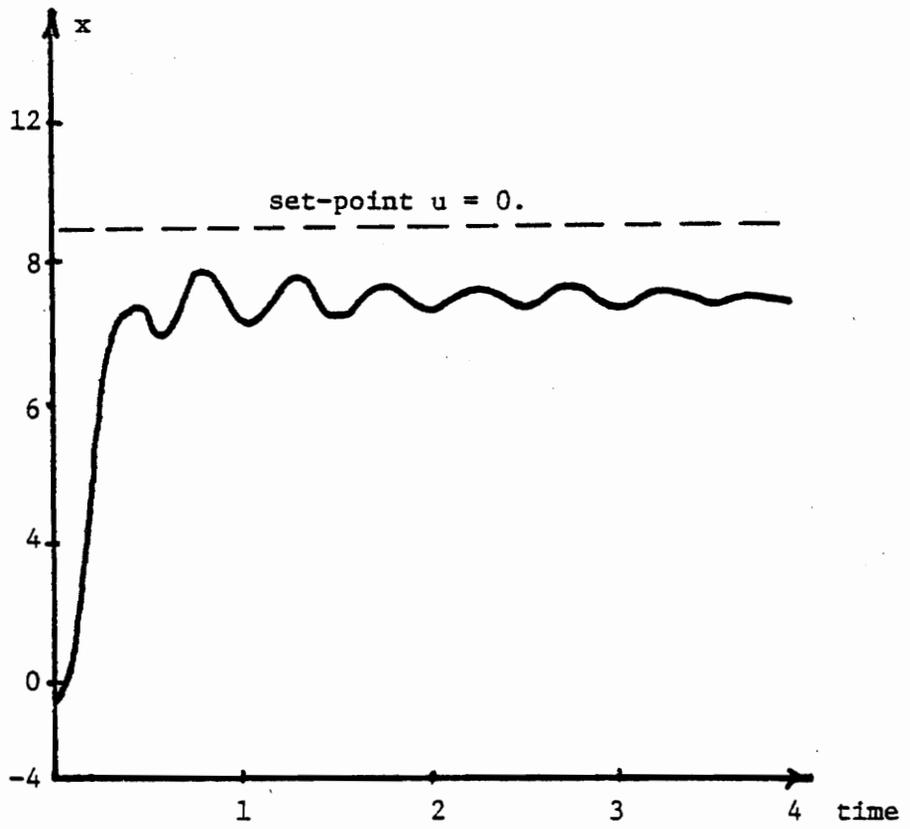


Figure 4.4.6 Set-Point Outside the Design Range.

control input of  $u = 9$ . It is seen that the plant position oscillates around  $x = 7.0$ . The operational range of the controller can be extended by adding more Kalman filters matched to larger values of  $x$ . A five-region controller was constructed and simulated for the position range of  $-8$  to  $8$ . The two added linear approximations are given below.

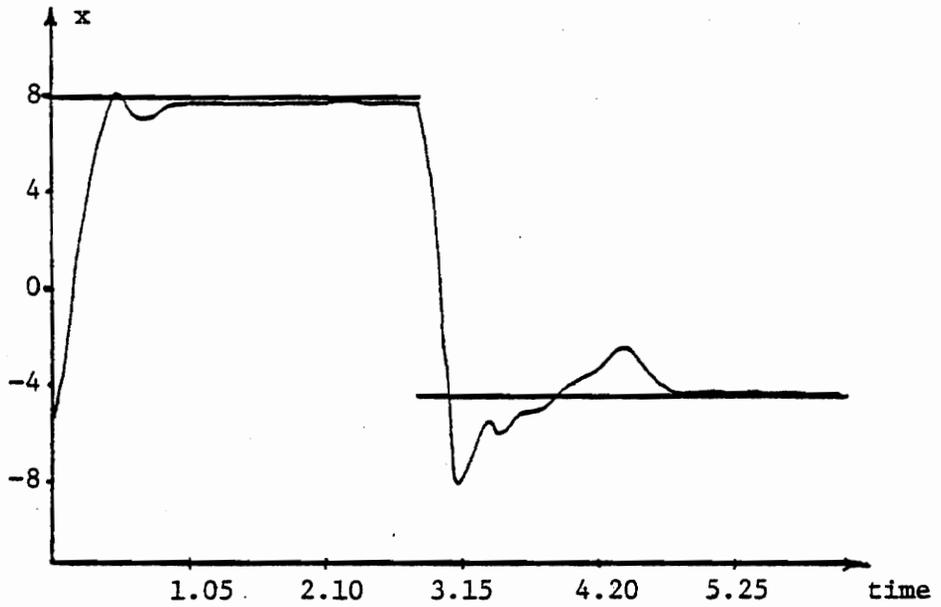
$$\begin{aligned}
 S_1 \begin{bmatrix} \dot{x}_1 \\ x_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -108 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & -432 \end{bmatrix} \begin{bmatrix} u \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w \\
 S_5 \begin{bmatrix} \dot{x}_1 \\ x_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -108 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & +432 \end{bmatrix} \begin{bmatrix} u \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w \quad (4.4.2)
 \end{aligned}$$

The three original regions are now  $S_2$ ,  $S_3$  and  $S_4$ . (This increased position range represents a doubling of the forces involved.) Figure 4.4.7a shows how the plant position follows an input of  $u = 8$  changing to  $u = -4$ . The corresponding five weighting coefficients are plotted in Figure 4.4.7b. (To avoid confusion, only the coefficients of filters 2 and 5 are shown connected.) Again, the learning time lag is responsible for the overshoot at  $t = 3.15$  seconds.

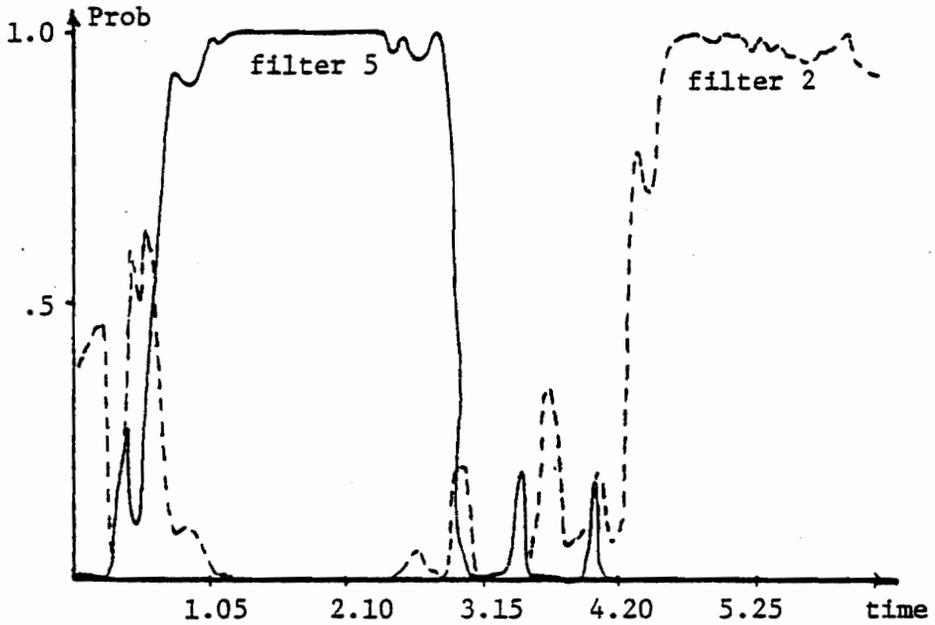
The semi-Markov matrix entries balance the controllers ability to follow configuration changes, with its noise filtering ability. The above examples used off-diagonal elements of  $.002$  and  $.005$ . It was experimentally determined that these values gave good overall performance. To illustrate the effect of larger off-diagonal entries, Figure 4.4.8 is a repeat of Figure 4.4.5b, but with a semi-Markov matrix of

$$\theta = \{\text{diagonal } .8, \text{ off-diagonal } .1\}$$

The plant is clearly out of control. The algorithm is responding too



(a) Set-Point Changing from  $u = 8$  to  $u = -4$ .



(b) Weighting Coefficients

Figure 4.4.7 Set-Point Control for Extended Design Range.

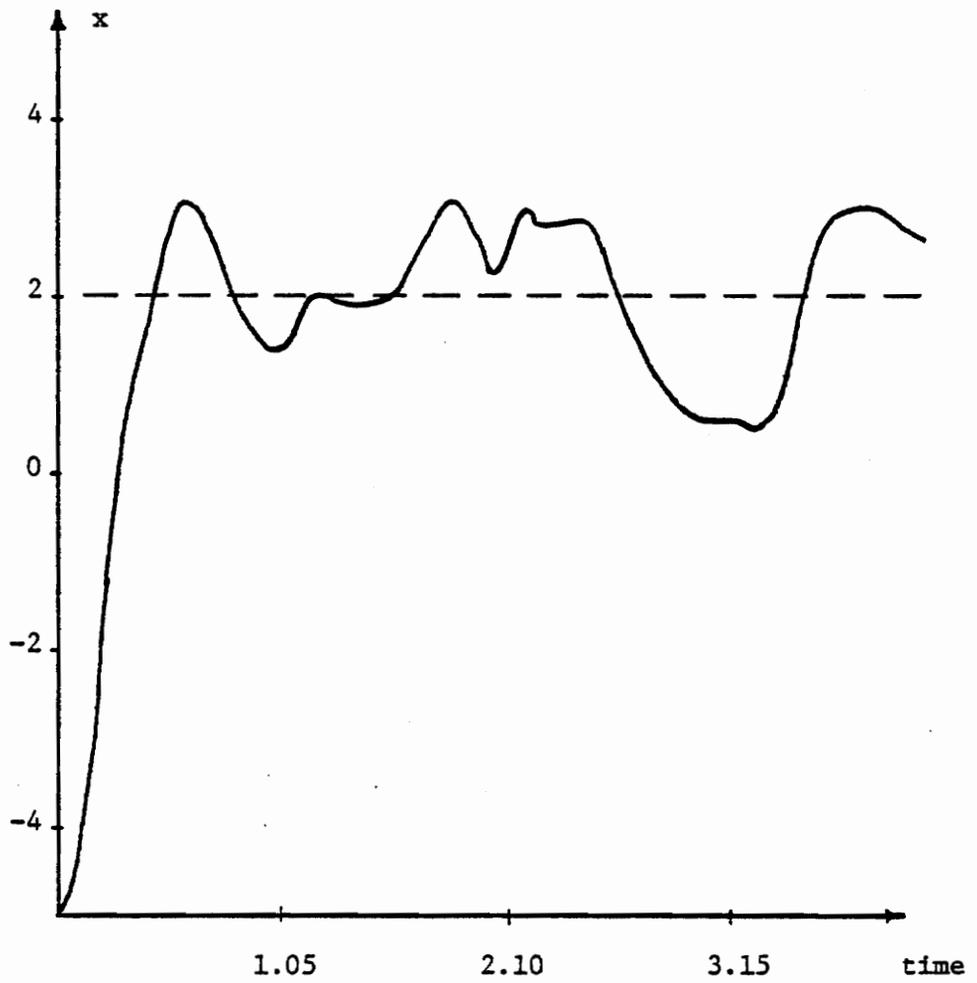


Figure 4.4.8 Set-Point Control for  $\theta = \{\text{diagonal } .8, \text{ off-diagonal } .1\}$ .

much to the noise in the system with the result that the feedback and input gains are constantly being taken from different regions.

#### 4.5 Summary

A method of nonlinear stochastic control, applicable to a large class of problems, and having significant advantages in implementation has been presented. The method is based on partitioning the nonlinear problem into regions where basic linear control techniques and standard Kalman filtering theory can be applied. The manner in which the nonlinear plant's state vector moves through the set of regions (called the configuration space) and the probable holding time spent in each region are incorporated into the controller. This information is used in conjunction with the plant measurements to learn which region the plant is in so that the proper control can be applied.

The structure of the controller is a bank of Kalman filters each matched to a region of the configuration space. The filter estimates are multiplied by corresponding feedback gains (different regions require different gains) and the overall state variable feedback to the plant is computed as a weighted sum of the individual feedback values.

The number of regions used in the formulation of the controller depends on the design criteria and on the nature of the nonlinear system itself. For example, a larger operational range or better accuracy will require more regions. The type and magnitude of the nonlinearity will also influence the number of regions. It is necessary that the adjacent regions overlap in order to drive the plant from one region to another; however, this last requirement is usually satisfied if the number of

regions is chosen solely to give reasonably good accuracy.

Each Kalman filter operates independently on the current measurement thus making this controller amenable to parallel processing. In addition, the filter gains can be precomputed and stored in tabular format (unlike most nonlinear filters, such as the EKF [G-3]). These two advantages yield a very low execution time for the implemented controller.

The modified PAC algorithm has several disadvantages. First, the PAE cannot, nor was it developed to, compete with the EKF in terms of accuracy. The estimation portion of the control algorithm is, therefore, similarly limited. Second, when the set-point is between the linearized regions, i.e. both approximations are equally bad, the error between the set-point and the actual plant steady state can be unacceptable. If real time computational constraints prevent adding more linearized filters to reduce this error, then a different method of control may have to be used.

## 5.0 THE DETECTION OF PLANT CONFIGURATION CHANGES

### 5.1 Introduction

The PAE algorithm described in Chapter Three performs satisfactorily for plants having a low frequency of configuration changes. However, when the changes occur at a rate comparable to the actual plant dynamics, the estimator can lose track. For cases of this type, the method was applied successfully only after an ad hoc switch detection mechanism was implemented to re-initialize the bank of Kalman filters after each configuration change.

In this chapter an algorithm will be developed to track a rapidly changing switched linear plant. First, the optimal estimator will be derived. This estimator will consist of a set of time-invariant linear Kalman filters. Unfortunately, the number of filters in the set will increase linearly with time, which makes the algorithm unrealizable for all but trivial problems. Next, a practical approximation to the optimal estimator will be developed based on the idea of joint estimation and detection. A constant number of filters will use the  $N$  most recent measurements to detect the plant configuration and to estimate the plant state. Various implementation considerations will be examined to improve estimation accuracy and to prevent divergence. Three examples will be given to clarify the operation of the algorithm. In addition, the tracking performance will be compared to the PAE algorithm of Chapter Three, and to a perfectly matched, unrealizable Kalman filter.

A variety of other interesting problems similar to the switched plant problem have been studied in the literature. Ackerson and Fu [A-1]

formulated an estimator for a system influenced by randomly changing noise statistics. The optimal estimate was found to be a weighted sum of estimates taken from an exponentially growing number of filters. McAulay and Denlinger [M-1] employed statistical decision theory to detect maneuvers in a target tracking problem. Their formulation successfully merged estimation theory with a generalized likelihood ratio test to obtain a practical adaptive tracker. The problem of detecting system component failures, modeled as jumps in certain state variables, was solved by Willsky and Jones [W-1]. Again, the optimum estimator proved to be a growing bank of matched filters. A "finite data window" in conjunction with a generalized likelihood ratio test was used to overcome the infinite memory problem. Additional discussion on these and other techniques is found in Chapter One.

## 5.2 The Optimal Estimator for the Switched Linear Plant

Consider a linear system described by the discrete time state equations,

$$\begin{aligned}x_{k+1} &= \Phi(k+1)x_k + \Psi(k+1)u_k + \Gamma(k+1)w_k \\z_{k+1} &= H(k+1)x_{k+1} + v_{k+1}\end{aligned}\tag{5.2.1}$$

The optimal estimator for this system is simply a matched filter (Kalman filter), generalized to include the time-varying transition and coefficient matrices. The above system is deterministic in the sense that the system matrices are known functions of time.

A second class of systems is that set whose system matrices are probabilistic functions of time. A frequently encountered example in

this class is the switched linear plant problem with the state variable description,

$$x_{k+1} = D_i x_k + C_i u_k + Q_i w_k$$

$$z_{k+1} = H x_{k+1} + v_{k+1}$$

where,

$$\begin{aligned} \phi_i &\in \{\phi_1, \phi_2, \dots, \phi_m\} \\ \psi_i &\in \{\psi_1, \psi_2, \dots, \psi_m\} \\ \Gamma_i &\in \{\Gamma_1, \Gamma_2, \dots, \Gamma_m\} \end{aligned} \quad (5.2.2)$$

The system matrices switch randomly among the elements of the above three finite sets. A probabilistic law governing the switching may or may not be given. For convenience, the elements are regrouped as,

$$s_i = (\phi_i, \psi_i, \Gamma_i) \quad (5.2.3)$$

where the 3-tuples will be referred to as configurations, and the set,  $S$ , of 3-tuples, the configuration space.

$$S = \{s_i, i=1, m\}$$

It is desired to construct a state estimator for the switched linear plant problem. First, it is noted that if the configuration changes and the times they occur are known exactly, then the problem reduces to the time varying linear system of equation (5.2.1), and the estimator structure is known immediately. Now consider a switched linear plant where the times of the configuration changes are not known in advance. For now, assume that the initial configuration of the plant is given. At some later time the plant configuration may change to another of the

M possible configurations. In addition, it is assumed that the configuration change can occur only at discrete points in time, and that only one change will occur. This last assumption will be removed later in the chapter.

At each iteration the plant can change in any of M-1 directions. Accounting for the possibility that no change is made during the N time intervals, the total number of possible paths is  $(N)(M-1) + 1$ . This is illustrated in Fig. 5.2.1, for a plant having M=3 configurations over an interval of N=15 iterations. The plant is initially in  $s_1$ , and at each succeeding iteration a configuration change to  $s_2$  can occur, as illustrated by a positive sloped line branching off at that iteration. Likewise a configuration change to  $s_3$  is shown as a negative sloped line. Once a change occurs, however, the plant remains in the new configuration. This is illustrated by moving continually along the positive or negative sloped line for the remaining time after the change. No configuration change at a time interval is signified by moving horizontally along the time axis.

A branch is defined to be a time history of possible plant configurations over N time intervals.

$$B_j = \{s(k) : k = 1, n; \text{ where } s(k) \in S\} \quad (5.2.4)$$

For example, the double line on Fig. 5.2.1 is the graphical representation of a branch defined by,

$$B_7 = \{s_1 s_1 s_1 s_1 s_1 s_1 \quad s_2 s_2 s_2 s_2 s_2 s_2 s_2 s_2\}$$

In this case, the plant is in  $s_1$  for six time intervals. At the seventh

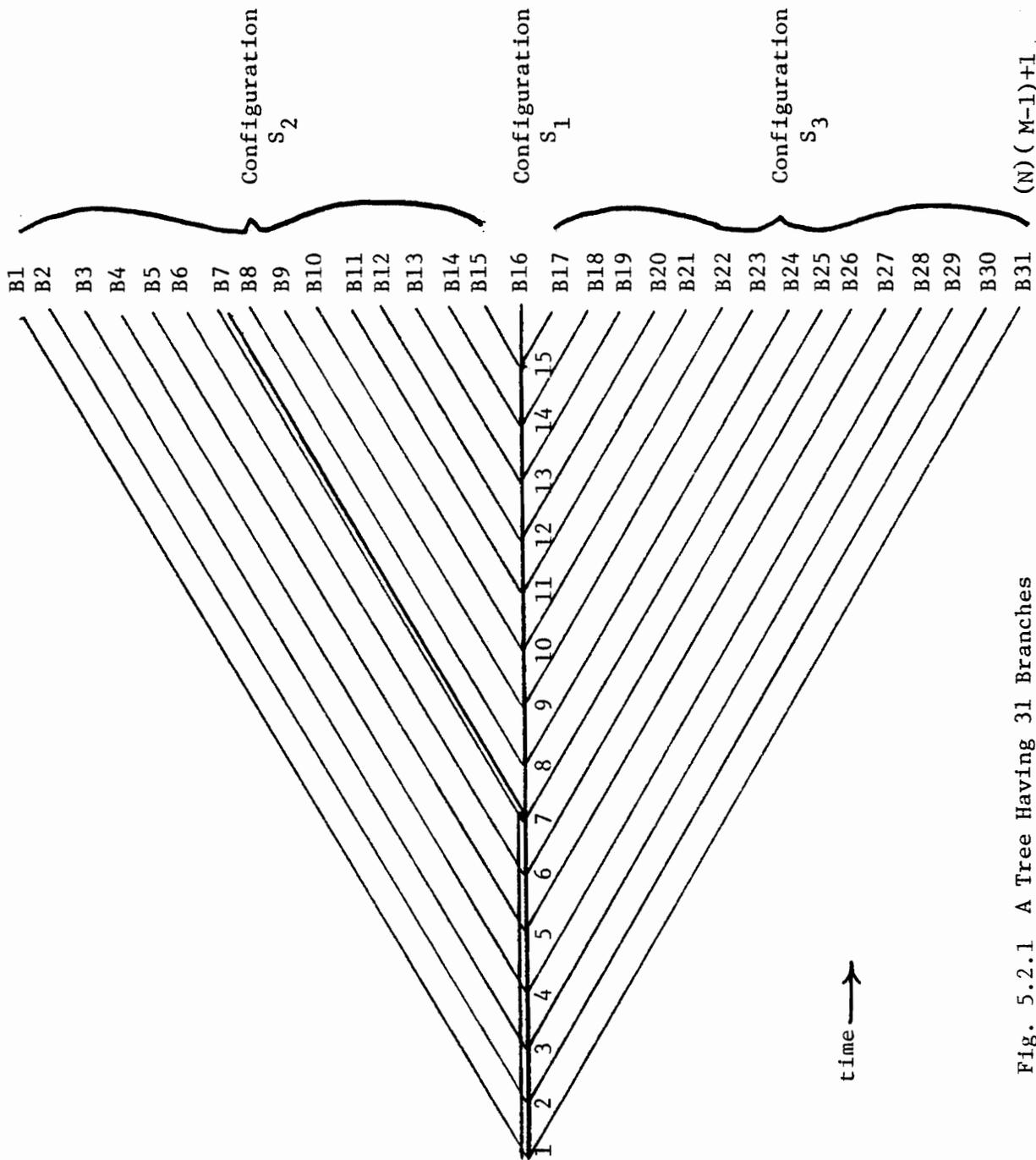


Fig. 5.2.1.1 A Tree Having 31 Branches

iteration a configuration change is made to  $s_2$ , and the remaining nine intervals are spent in  $s_2$ . If the plant did indeed follow this configuration history, then after 15 iterations the plant state would be given by  $B_7$ .

A TREE, initialized at time  $k$ , is defined to be the set of all branches,

$$T(k) = \{B_j; J=1, (N)(M-1) + 1\}. \quad (5.2.5)$$

The TREE in Fig. 5.2.1 has 31 branches, numbered consecutively from the top branch down.

Given a probabilistic description governing the configuration changes, a minimum mean square error, mmse, state estimator can be derived using Bayes' rule. Let  $Z_k$  be a sequence of measurements on the state vector taken at the first  $k$  iterations. The conditional mean of the state vector at time  $K$  can be expressed as

$$\hat{x}_k = \int_{\mathbf{x}} \mathbf{x} p(\mathbf{x}_k = \mathbf{x} / Z_k) d\mathbf{x}. \quad (5.2.6)$$

The probability distribution can be expanded using Bayes' rule as follows.

$$\begin{aligned} p(\mathbf{x}_k / Z_k) &= \frac{p(\mathbf{x}_k, Z_k)}{p(Z_k)} = \frac{\sum_{B_J=B_1}^{B(N)(M-1)+1} p(\mathbf{x}_k, Z_k, B_J)}{p(Z_k)} \\ &= \sum_{B_J} p(\mathbf{x}_k / Z_k, B_J) p(Z_k, B_J) / p(Z_k) \\ &= \sum_{B_J} p(\mathbf{x}_k / Z_k, B_J) p(B_J / Z_k) \end{aligned} \quad (5.2.7)$$

Substituting (5.2.7) into (5.2.6) gives,

$$\hat{x}_k = \int_{\mathbf{x}} \mathbf{x} d\mathbf{x} \sum_{B_J} p(x_k = \mathbf{x}/Z_k, B_J) p(B_J/Z_k)$$

Taking  $x_k$  inside the summation and interchanging the operations of integration and summation, the conditional mean becomes

$$\begin{aligned} \hat{x}_k &= \sum_{B_J} \int_{\mathbf{x}} \mathbf{x} p(x_k = \mathbf{x}/Z_k, B_J) p(B_J/Z_k) d\mathbf{x} \\ &= \sum_{B_J} p(B_J/Z_k) \int_{\mathbf{x}} \mathbf{x} p(x_k = \mathbf{x}/A_k, B_J) d\mathbf{x} \end{aligned}$$

The integral on the right is the mean of the state vector, conditioned on the observed measurements,  $Z_k$ , and on the branch  $B_J$ . It is the optimal estimate (minimum mean square error) of the state given that  $B_J$  is the configuration history of the plant. Denoting this integral by  $\hat{x}_k(B_J)$ , the unconditional optimal estimate is,

$$\hat{x}_k = \sum_{B_J} \hat{x}_k(B_J) p(B_J/Z_k) \quad (5.2.8)$$

Using Bayes' rule, the a posteriori probability can be expressed as,

$$p(B_J/Z_k) = p(Z_k/B_J) p(B_J) / p(Z_k) \quad (5.2.9)$$

The probability distribution function governing the configuration changes,  $p(B_J)$ , is assumed to be known. The conditional measurement density function,  $p(Z_k/B_J)$ , can be expanded by representing the sequence  $Z_k$  by  $z_k, Z_{k-1}$ , i.e.

$$p(Z_k/B_J) = p(z_k, Z_{k-1}/B_J) = p(z_k/Z_{k-1}, B_J) p(Z_{k-1}/B_J) \quad (5.2.10)$$

where  $z_k$  is the measurement at time  $k$ . It is readily shown in the derivation of the Kalman filter (see Chapter 2) that,

$$p(z_k/Z_{k-1}, B_J) = N(\mu_k^J, P_z^J(k)) \quad (5.2.11)$$

where  $\mu_k^J$  is the predicted measurement, and  $P_z^J(k)$  is the measurement residual covariance, given by,

$$\mu_k^J = H[\Phi^J(k)\hat{x}_{k-1}(B_J) + \Psi^J(k)u_{k-1}]$$

$$P_z^J(k) = HP_x^J(k/k-1)H^T + R$$

The notation  $\Phi^J(k)$ ,  $\Gamma^J(k)$  or  $\Psi^J(k)$  specifies the matrix  $\Phi$ ,  $\Gamma$  or  $\Psi$ , according to the configuration  $B_J$  at time  $k$ . Superscript  $J$  is chosen to differentiate between  $\Phi_i(k)$  which was previously defined to be configuration  $i$  at time  $k$ .  $P_x^J(k/k-1)$  is the predicted state estimation error covariance conditioned on  $B$ , which was shown in Chapter Two to be

$$P_x^J(k/k-1) = \Phi^J(k)P_x^J(k-1)\Phi^J(k)^T + \Gamma^J(k)Q\Gamma^J(k)^T$$

The measurement residual at time  $k$  is given by,  $\tilde{z}_k = z_k - \mu_k^J$ . Hence, the value obtained from evaluating the measurement *residual* density function at  $\tilde{z}_k$  equals the value obtained from evaluating the measurement density function at  $z_k$ , i.e.

$$p_{\tilde{z}/B_J}(\tilde{z}_k/B_J) = p_{z/Z_{k-1}, B_J}(z_k/Z_{k-1}, B_J). \quad (5.2.12)$$

The conditioning on  $Z_{k-1}$  is not required on the measurement residual density function since, given Branch  $B_J$ , it does not depend explicitly on the past history, Now,

$$P_{\tilde{z}/B_J}(\tilde{z}/B_J) = [(2\pi)^N |P_Z^J(k)|]^{-1/2} \exp[-1/2 \tilde{z}^T (P_Z^J(k))^{-1} \tilde{z}] = N(0, P_Z^J(k)) \quad (5.2.13)$$

Substituting (5.2.12) into (5.2.10) gives,

$$P(Z_k/B_J) = P_{\tilde{z}/B_J}(\tilde{z}_k/B_J) P(Z_{k-1}/B_J) \quad (5.2.14)$$

The form of the optimal state estimate is found by substituting (5.2.12) into (5.2.10), (5.2.10) into (5.2.9), and (5.2.9) into (5.2.8).

$$\hat{x}_k = \frac{1}{P(Z_k)} \sum_{B_J} \hat{x}_k(B_J) P(\tilde{z}/B_J) P(B_J) P(z_{k-1}/B_J) \quad (5.2.15)$$

where the probability density function's subscripts have been removed for simplicity. Thus, for the switched linear plant problem, the optimal estimate is a weighted sum of the conditional estimates that are obtained from filters matched to all possible branches of the configuration tree. The weighting coefficients are found recursively. First,  $p(Z_k/B_J)$  is found in equation (5.2.14) by multiplying  $p(Z_{k-1}/B_J)$ , from the last iteration, by  $p(\tilde{z}_k/B_J)$  of equation (5.2.13). The coefficients are then computed in equation (5.2.0) by multiplying  $p(Z_k/B_J)$  by the a priori branch probabilities,  $p(B_J)$ , and normalizing. This last step makes the actual calculation of  $p(Z_k)$  unnecessary.

### 5.3 Sliding Window Detector/Estimator, SWDE

The estimator derived in the previous section is optimal in the sense that it minimizes the mean square estimation error in the presence of uncertainty regarding the plant configuration. This uncertainty is manifested by the incorporation of a governing probability distribution function into the plant description. There are, however, several

reasons that make this method impractical. First, a probabilistic description governing the configuration changes might not be known. Second, only single configuration changes are allowed. And, third, the order of the estimator grows linearly with time. This limits the real time applicability of the algorithm because of constraints on available computer storage and processing speed.

A different approach to tracking a switched linear plant is based upon the detection of configuration changes. The distinction between pure estimation theory and signal detection theory is rather arbitrary. Both strive to extract information from noisy environments. One tangible difference is that the detection process is a choice between a finite number of hypotheses, whereas the estimation process can have a continuum of possible outcomes. In addition, detection algorithms often require less information about the system than do estimation algorithms. For example, a Neyman-Pearson [V-1] algorithm does not require a priori probabilities for the various hypotheses. In this section, a detector will be constructed that will determine when a configuration change occurs and what the new configuration is. This information will then be used to formulate a state estimate for the plant. The algorithm will not require a priori knowledge of configuration changes. In addition, after a change has been detected, the algorithm can be re-initialized to detect a second change. Thus multiple changes can be tracked.

The joint detection/estimation problem can be solved by growing trees of predetermined size at each iteration similar to the tree in Section 5.2. Now, however, as time progresses, the size of the trees will be kept constant by pruning the branches at each iteration according

to a decision rule. A second difference between this section and Section 5.2 is the method by which the estimate is determined.

Assume that the configuration prior to time  $k$  is known. At time  $k$  the plant can change to another of the  $M$  possible configurations. Starting at  $t=k$ , measurements are taken at each iteration for  $N$  iterations. At  $t=k+N-1$  a decision is made regarding what changes, if any, occurred at  $t=k$ . The number of measurements used in making a decision will, henceforth, be referred to as the window width,  $N$ .

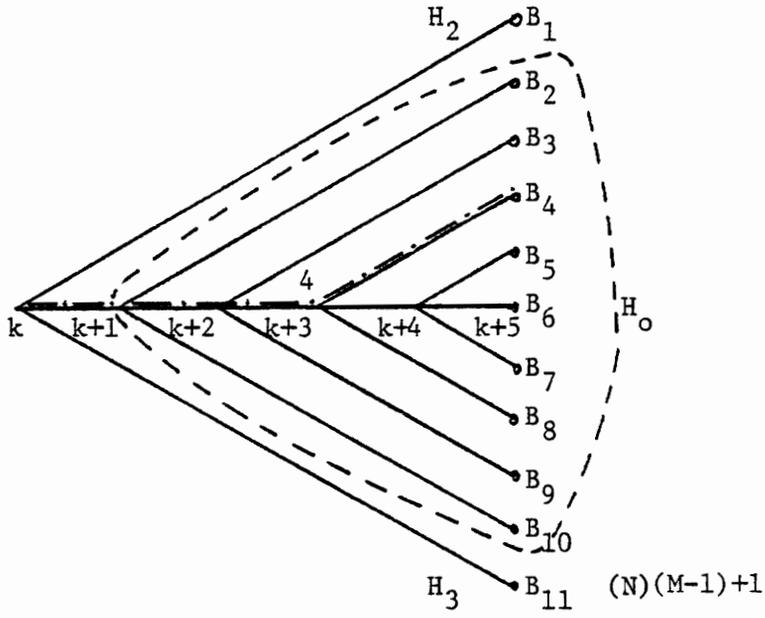
All of the information required in the detection/estimation process is available in a tree  $T(k)$  initiated at  $t=k$ . The branches of  $T(k)$  correspond to all the possible configuration histories of the plant from  $t=k$  to  $t=k+N-1$ . Associated with each branch are two types of information. First, a state estimate is propagated along each branch. Every branch shares the same initial state estimate called the *seed* estimate. The estimate along any given branch is the optimum estimate provided that the plant followed the configuration history defined by that branch. Since all configuration histories are accounted for, the optimum estimate is always available somewhere in the tree. Second, probability information is propagated along each branch. In Section 5.2, these probabilities were used in conjunction with the a priori branch probabilities to compute the weighting coefficients for the optimal estimator. Now they will be used in the detection of configuration changes.

At time  $t=k+N-1$  a decision is made as to what configuration change, if any, occurred at  $t=k$ . The null hypothesis,  $H_0$ , is that no change was made at  $t=k$ . The  $H_i$  hypothesis corresponds to a change to configuration  $s_i$  at  $t=k$ . Thus, the null hypothesis encompasses all changes that could

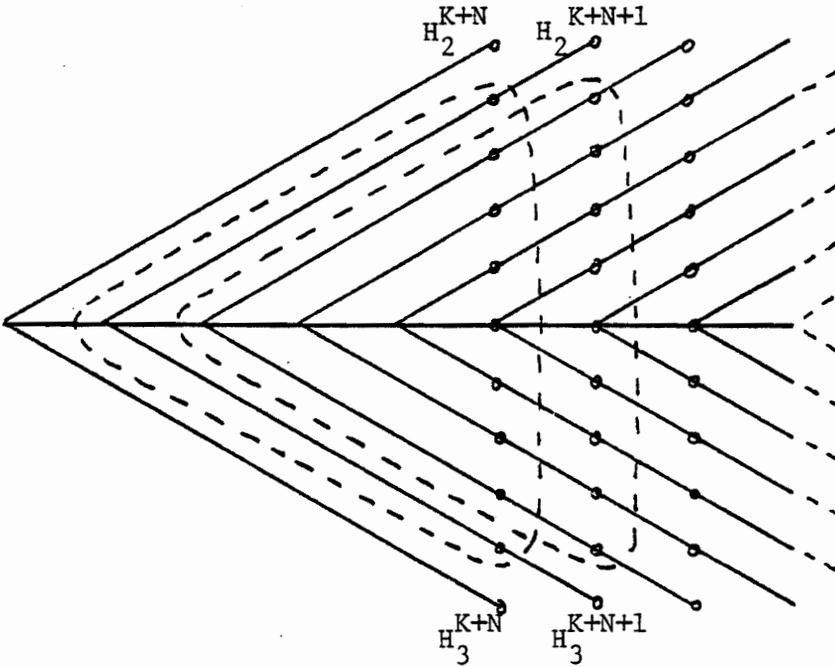
occur between  $t=k+1$  and  $t=k+N-1$ , including the possible outcome that no change is made during the  $N$  iterations. For example, Figure 5.3.1a shows a tree having a window width of  $n=6$ . At the sixth iteration a decision is made and one of the eleven branches is chosen. If  $B_1$  ( $B_{11}$ ) is chosen, then the hypothesis  $H_2$  ( $H_3$ ) is accepted. On the other hand, if  $B_2$  through  $B_{10}$  is chosen, then  $H_0$  is accepted.

Depending upon the decision outcome at  $t=k+N-1$ , one of two courses of action will be taken. First, if  $H_0$  is accepted, then the state estimate at  $t=k+N-1$  is taken from the branch corresponding to no configuration changes, i.e. branch  $B_6$  in Figure 5.3.1a, and the configuration tree is propagated forward one iteration. The size of the tree is kept constant, however, by disregarding the branches corresponding to configuration changes at  $t=k$ . This can be done because the decision at  $t=k+N-1$  has eliminated these possibilities. A test is then carried out at  $t=k+N$  for a configuration change that may have occurred at  $t=k+1$ . This forward propagation of the configuration tree continues with each iteration. Because the decision process always uses the  $N$  most recent measurements, the algorithm is referred to as a sliding window detector/estimator (SWDE). Figure 5.3.1b shows the propagation of the tree in Figure 5.3.1a over several iterations.

The second course of action is taken when  $H_1$  is accepted. The state estimate at  $t=k+N-1$  is now taken from the branch corresponding to a change to  $s_1$  at  $t=k$ . For example, in Figure 5.3.1a, if  $H_2$  is accepted, then the state estimate is taken from branch  $B_1$ . In addition, a new configuration tree is grown with  $s_2$  as the initial configuration, and with the estimate at  $t=k+N-1$  being used as the new seed estimate. The search for a second



(a) Branch  $B_4$  Lies in Decision Region  $H_0$ .



(b) Time Propagation of the Decision Regions.

Fig. 5.3.1 The Decision Regions,  $H_0$ ,  $H_2$  and  $H_3$ .

configuration change can now commence.

This method of joint detection/estimation insures that the state estimate for the plant is taken from the optimum filter, except during the lag time between a configuration change and detection. Figure 5.3.2 illustrates the operation of SWDE over several configuration changes. For clarity, a small window width was chosen.

#### 5.4 Implementation

It is assumed that the estimator knows the initial plant configuration and that steady state filtering has been achieved. This allows the use of the steady state values of Kalman gain and measurement residual covariance. If these assumptions are not justifiable, then an initialization algorithm, discussed later in the chapter, can be used to bring the system to these conditions.

A Kalman filter will be matched to each branch of the configuration tree. The information available from these filters is used not only in obtaining the state estimate, but also in calculating the probabilities required in the detection algorithm. The required filter matrices can be found by solving the steady state discrete time Riccati equation for each configuration. The solution will depend on the noise matrices  $Q$  and  $R$ , as well as the system matrices  $\Phi$ ,  $\Gamma$  and  $H$ .

Ordinarily, when a configuration change occurs in the plant, the corresponding matched filter undergoes a transient period in which the filter gains and error covariance matrices change from the steady state values associated with the initial configuration to the steady state values associated with the new configuration. The settling time for this

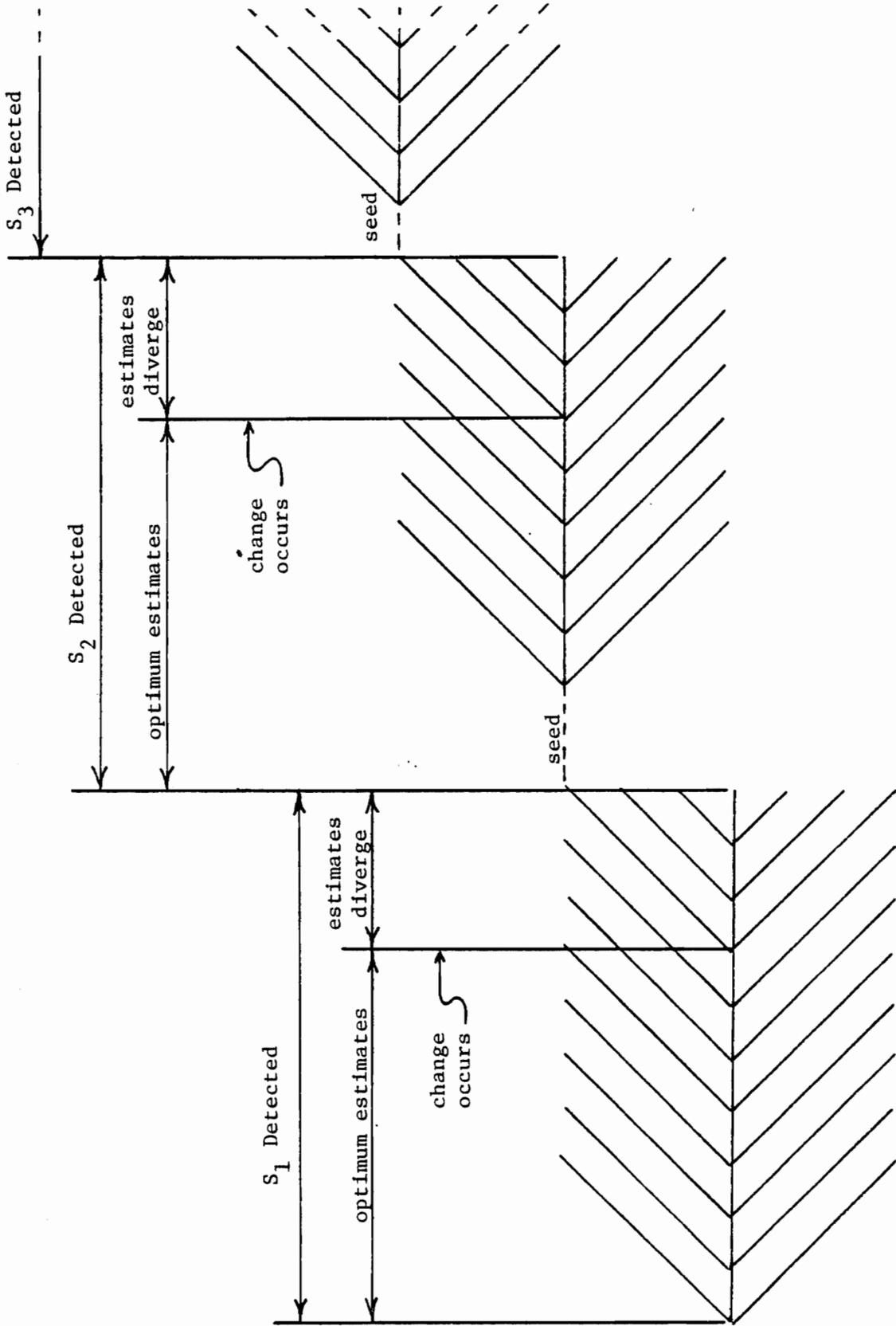


Fig. 5.3.2 Estimator Operation Over Several Configuration Changes.

transient is usually fairly short; therefore, in the implementation of the detector the transient will be ignored. That is, when a configuration change occurs along a branch, the associated matched filter matrices will switch immediately to their new steady state values. This precludes the calculation and storing of the filter matrices for the various transient situations.

The detection algorithm requires a relative measure of likelihood for the various branches. In Section 5.2, the set of weighting coefficients was such a measure. However, in the current problem a probability distribution,  $p(B_J)$ , governing the configuration changes is not available, and hence the weighting coefficients cannot be computed. See equation (5.2.9). A more appropriate measure, that does not require a priori branch information, is the conditional measurement sequence probability.

Recall that  $Z_N$  was defined as a sequence of  $N$  measurements. This definition is now extended by defining

$Z_N(k)$  to be a sequence of  $N$  measurements  $z_k, z_{k+1}, z_{k+2}, \dots, z_{k+N-1}$ . The probability of the measurement sequence conditioned on branch  $B_J$  is  $P(Z_N/B_J)$ . Expanding  $Z_N(k) = \{z_{n+k-1}, Z_{N-1}(k)\}$ , the probability becomes,

$$p(Z_N(k)/B_J) = p(z_{n+k-1}, Z_{N-1}(k)/B_J) \quad (5.4.1)$$

$$= p(z_{N+k-1}/Z_{N-1}(k), B_J)p(Z_{N-1}(k)/B_J) \quad (5.4.2)$$

Repeating the expansion for  $Z_{N-1}(k), Z_{N-2}(k)$ , etc., the probability is reduced to factored form,

$$\begin{aligned}
 p(Z_N(k)/B_J) &= p(z_{N+k-1}/Z_{N-1}(k), B_J) p(z_{N+k-2}/Z_{N-2}(k), B_J) \dots \\
 &\dots p(z_{k+1}/Z_1(k), B_J) p(z_k/B_J). \quad (5.4.3)
 \end{aligned}$$

Now consider a configuration tree initiated at time  $k$  and having a window length of  $N$ . Iteration  $k$  is common to all branches and the state estimate at time  $k$  is referred to as the seed estimate. As the measurements are processed, a state estimate is propagated along each branch of the tree. In addition, equation (5.4.3) is used to calculate the conditional probability of obtaining the measurement sequence,  $Z_N(k)$ , for each branch.

Using the same reasoning that resulted in equation (5.2.12), each factor of equation (5.4.3) can be replaced with an equivalent measurement residual density function, such that the values obtained by equating are identical,

$$\begin{aligned}
 p(Z_N(k)/B_J) &= p(\tilde{z}_{N+k-1}/B_J) p(\tilde{z}_{N+k-2}/B_J) \dots \\
 &\dots p(\tilde{z}_{k+1}/B_J) p(\tilde{z}_k/B_J) \quad (5.4.4)
 \end{aligned}$$

where

$$p(\tilde{z}_i/B_J) = N(0, P_2^J(i)) \quad (5.4.5)$$

and  $P_2^J(k)$  is the measurement residual covariance at time  $i$ , for the filter matched to  $B_J$ . Equation (5.4.4) is evaluated for all  $(N)(M-1)+1$  branches and the resulting probabilities are referred to as the relative branch probabilities at time  $N+k-1$ . Similarly, the relative branch probabilities at time  $N+k$  are, with respect to time  $t_{k+1}$ ,

$$p(Z_N(k+1)/B_J) = p(\tilde{z}_{N+k}/B_J) \dots p(\tilde{z}_{k+2}/B_J)p(\tilde{z}_{k+1}/B_J) \quad (5.4.6)$$

Comparing (5.4.6) to (5.4.3), it is seen that,

$$p(Z_N(k+1)/B_J) = p(\tilde{z}_{N+k}/B_J)p(Z_N(k)/B_J)/p(\tilde{z}_k/B_J) \quad (5.4.7)$$

Hence the relative branch probability at time  $k+N$  is computed from the relative branch probability at time  $k$  by multiplying by the probability of the  $k+N$  measurement residual and dividing by the probability of the measurement residual at time  $k$ .

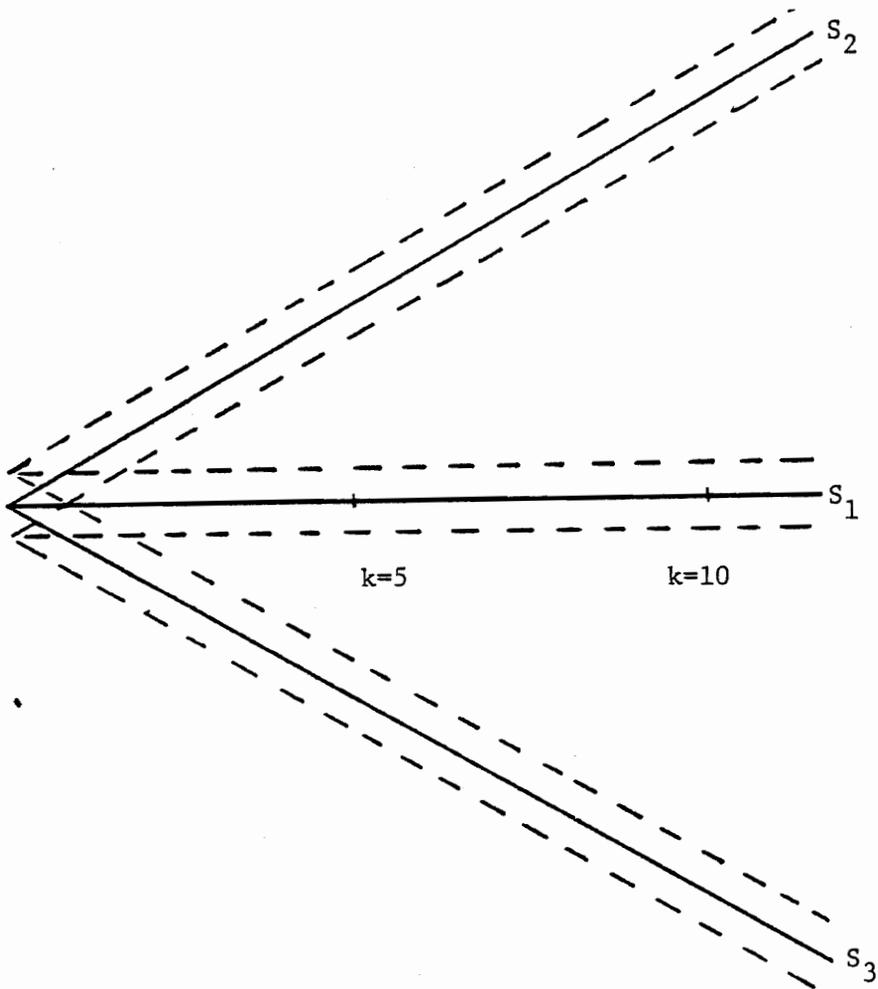
The relative branch probabilities are used directly in the decision making process. Referring to Figure 5.3.1b, the number of branches at each iteration is a constant given by  $(N)(M-1)+1$ . A vertical line drawn at each iteration identifies the branches that are tested at that iteration. The discrete distances above or below the centerline, defined by the intersections of branches and vertical lines, are referred to as levels. The number of levels equals the window width,  $N$ . A configuration change is thus detected when any level  $N$  branch has a relative probability greater than any other relative branch probability terminating on the same vertical line.

Many factors influence the selection of the window width,  $N$ . A wider window gives a smaller value of probability of error for detection, but can give poorer estimation results. In addition, hardware constraints will impose an upper limit on window width. First, consider estimator performance during a configuration change. Prior to the change, the estimate is taken from a filter perfectly matched to the plant, and is therefore optimum. When the change occurs, the estimate

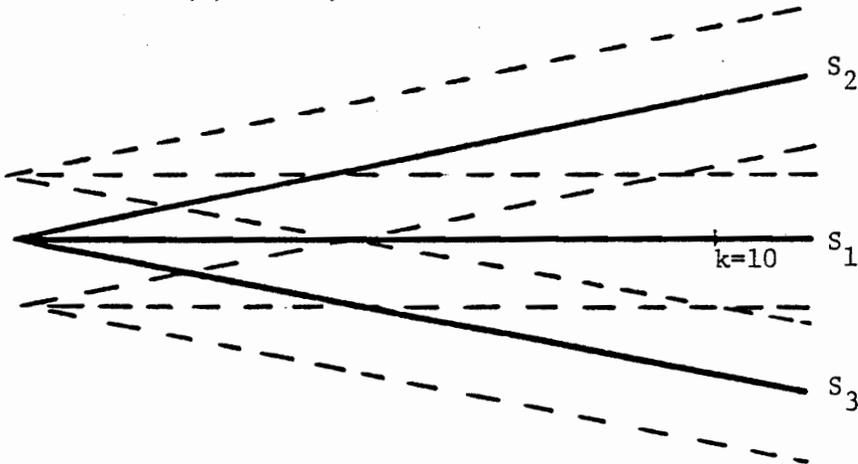
is still taken from the above same filter until  $N$  iterations have passed. At this time the configuration change is detected, and a new matched filter is used for the estimates. Before detection occurs, the estimate is diverging from the true state trajectory. Hence, a wider window will lead to more divergence in the period immediately following a configuration change.

On the other hand,  $N$  must be chosen wide enough to insure the detection of configuration changes, as well as the rejection of false alarms. A false alarm occurs when a configuration change is detected when, in fact, none occurred. This is undesirable because, during the  $N$  iterations following a false alarm, the estimator cannot detect a true configuration change. Moreover, the estimates will be taken from the wrong filter for at least  $N$  iterations. After this interval the estimator may or may not be able to recover. A miss occurs when a configuration change is not detected after  $N$  iterations have passed. This type of error is more serious for the estimator because the optimum state estimate is irretrievably lost. When a miss occurs, the estimator will operate erratically and may have to be reinitialized, as discussed later in the chapter.

The false alarm and the miss probabilities are dependent upon the degree of overlap of the measurement distributions along the various tree branches. This, in turn, is determined by the values of system disturbance and measurement noise, and by how rapidly the branches diverge from one another. This is illustrated in Figure 5.4.1. A larger degree of divergence with moderate noise is shown in 5.4.1a. The dashed lines represent the standard deviations of the measurements. If



(a) Fast, with moderate noise.



(b) Slow, with high noise.

Fig. 5.4.1 Branch divergence.

the plant is actually following branch  $B_2$ , then at the fifth iteration the measurement will be, on the average, very distant from branches  $B_1$  and  $B_3$ ; the respective probability density values will be very small, allowing rapid detection. In contrast, Figure 5.4.1b depicts slowly diverging branches in a highly noisy environment. It is seen that a larger number of measurements are needed to discriminate between the branches.

The rate at which the various branches diverge is governed by several factors. First, if the sample time is long with respect to the dynamics of the system, the amount of divergence between iterations will be greater. Second, strong inputs usually cause rapid divergence. Third, the dynamics of the different configurations may cause the state trajectories to cross one another. When this occurs, the measurement residuals can become misleadingly small, thus raising the prospect of a false alarm.

The dependence of detection on input makes an analytical derivation of probability of false alarm very difficult. An alternative method of determining window width is to run a simulation for several hundred iterations without any configuration changes. Start with a window width large enough to give no false alarms, and decrease the width until false alarms just start to occur, then increase the width a couple of iterations. Repeat this procedure for several monte carlo simulations, and use the worst case (largest value) window width.

### 5.5 Weighted Sum Estimation

The method by which the SWDE algorithm determines the state estimate will be optimum at all times except for the intervals following configuration changes. During these intervals, equal to the window width  $N$ , the estimate will diverge until SWDE detects the change. If the configuration changes occur infrequently, then the overall tracking will nearly always be optimum. However, if the configuration changes occur at a high frequency, the overall performance will be seriously degraded by numerous divergence intervals.

The amount of divergence can be substantially reduced by computing the estimate as a weighted sum of the individual branch estimates at each time. The weighting coefficients are simply the relative probabilities for the various branches normalized so that the sum of all the relative probabilities is unity. Thus,

$$\hat{x}_k = \frac{\sum_{B_J} \hat{x}_k(B_J) p(Z_N(k)/B_J)}{\sum_{B_J} p(Z_N(k)/B_J)} \quad (5.5.1)$$

where  $p(Z_N(k)/B_J)$  is given in equation (5.4.4).

In this manner the estimate is always composed of percentages of all the branch estimates, with the most likely branch being dominant. The drawback to this method is that the estimate is no longer optimum at those times when no configuration change is occurring. Furthermore, the overall estimate will be noisier because the weighting coefficients are directly affected by the measurement noise.

## 5.6 Initialization Algorithm

In many cases involving a switched plant problem, the initial configuration is not known. Moreover, an accurate initial value of the state vector is seldom available. Thus, the SWDE algorithm in its present form cannot be utilized. Even for well-defined problems where the algorithm can be used, there is always the possibility that a miss will cause the estimator to lose track of the plant's state. When this happens, the algorithm must be re-initialized.

The PAE algorithm of Chapter Three can be used to solve both the identification and the re-initialization problems. In this algorithm, time-varying weighting coefficients are calculated for each of the plant configurations. As the plant measurements are processed, the weight associated with the actual plant configuration will approach unity and the other weights will go to zero. In addition, an optimal state estimate is available at each iteration. After a number of iterations the PAE algorithm will reach steady state. At this time, the Kalman filter matrices will be at their constant steady state values and the state estimate will be taken principally from the filter matched to the actual plant configuration.

Configuration identification and steady state determination in the PAE algorithm is accomplished by monitoring the weighting coefficients and the error covariance matrices. In the implementation, the weighting coefficients are compared to a threshold probability value. If a weight exceeds the threshold, then the configuration corresponding to that weight is identified as being the actual plant configuration. Steady state can be determined by comparing the current error covariance matrix

with the covariance at the preceding iteration. If the norm of the difference is less than a given tolerance, then steady state has been achieved. Alternately, if the filter matrices have been precomputed offline, then steady state is determined by simply counting the number of iterations. The latter method is used in the examples.

Once steady state filtering and plant identification have been achieved, the SWDE algorithm developed in this chapter can be used to track a randomly switching plant. If, for whatever reason, the SWDE algorithm diverges from the actual plant state, then the PAE algorithm should again be executed.

Divergence can be readily detected by monitoring the measurement residual. The measurement residual of a perfectly matched filter is a zero mean, white gaussian random variable [G-1]. When divergence occurs, the residual will lose these attributes. A variety of statistical tests can be employed to determine divergence, the simplest being a test for a non-zero mean. However, the problem is complicated by the normal operation of the SWDE algorithm, which produces a diverging estimate for a number of iterations, equal to the window width  $N$ , after each configuration change. This problem is resolved by calculating the mean over several adjacent intervals, each containing  $N$  iterations. If all the mean values are greater than a given tolerance, then divergence is assumed and the PAE algorithm is executed. If a weighted sum estimate is used, then the number of intervals over which the mean is calculated can be reduced. In the following examples, two intervals were sufficient to detect estimator divergence. Appendix B contains a flowchart diagramming the computer program implemented to accomplish the above tasks.

### 5.7 Optimal Delayed Estimation

A method exists to eliminate entirely the divergence that occurs in the interval immediately following a configuration change. During this interval, which is equal to the window width  $N$ , the configuration tree is grown but the configuration change is not detected until  $N$  iterations after it occurred. In the real time estimation problem considered in the previous sections, the estimate is available shortly after the current measurement has been processed. However, if an estimation delay equal to the window width can be tolerated, then a delayed estimate can be constructed that will be optimal at all iterations.

The implementation of this feature requires only slightly more computer storage and virtually no additional processing time. Recall, that when the configuration tree is grown, the state estimates are calculated at each iteration for every branch. If these estimates are stored, then at the time the configuration change is detected, the algorithm can retrace the optimal state estimate trajectory.

The advantage of delayed estimation over weighted sum estimation is that the former method eliminates the divergence intervals without sacrificing estimator performance elsewhere.

### 5.8 Simulations

The algorithms developed in this chapter were verified by extensive off-line simulation on a digital computer. Three different plant models are considered in this section. Example One is the switched linear plant that was examined in Chapter Three. Example Two is a plant that has a parameter that can take on one of several possible values. The third

example applies the SWDE algorithm to the nonlinear oscillator of Chapter Three.

Example 5-1:

$$\begin{aligned} \text{Given} \quad \dot{\mathbf{x}} &= \mathbf{A}_1 \mathbf{x} + \mathbf{B}_1 u + \mathbf{C}_1 w \\ z &= \mathbf{H} \mathbf{x} + v \end{aligned}$$

where,

$$\mathbf{A}_1 = \begin{bmatrix} 0 & 1 \\ -30 & -1 \end{bmatrix} \quad \mathbf{B}_1 = \begin{bmatrix} 0 & 0 \\ 1 & -63.2 \end{bmatrix} \quad \mathbf{C}_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\mathbf{A}_2 = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \quad \mathbf{B}_2 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \quad \mathbf{C}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\mathbf{A}_3 = \begin{bmatrix} 0 & 1 \\ -30 & -1 \end{bmatrix} \quad \mathbf{B}_3 = \begin{bmatrix} 0 & 0 \\ 1 & 63.2 \end{bmatrix} \quad \mathbf{C}_3 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\mathbf{H} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} .3 & 0 \\ 0 & 3. \end{bmatrix} \quad Q = 4$$

The plant is simulated for an input of  $U = 4.0$ , an initial position of  $-4.5$  and an initial velocity of  $0$ . Unknown to the estimator, the plant changes configuration approximately every two seconds. A window width of  $N=7$  was found to be sufficient to suppress false alarms. The digital simulation requires the discretization of the continuous time system, and the solution of the steady state Kalman filter equations. The results of these two preliminary steps, for a sample time of  $T=.07$  seconds, are given below.

$$\phi_1 = \begin{bmatrix} .929 & .066 \\ -1.98 & .863 \end{bmatrix} \quad \psi_1 = \begin{bmatrix} .0024 & -.142 \\ .066 & 3.96 \end{bmatrix} \quad \Gamma_1 = \begin{bmatrix} .0024 \\ .066 \end{bmatrix}$$

$$\phi_2 = \begin{bmatrix} 1.0 & .068 \\ 0.0 & .932 \end{bmatrix} \quad \psi_2 = \begin{bmatrix} .0024 & 0. \\ .068 & 0. \end{bmatrix} \quad \Gamma_2 = \begin{bmatrix} .0024 \\ .068 \end{bmatrix}$$

$$\phi_3 = \begin{bmatrix} .929 & .066 \\ -1.98 & .863 \end{bmatrix} \quad \psi_3 = \begin{bmatrix} .0024 & -.142 \\ .066 & 3.96 \end{bmatrix} \quad \Gamma_3 = \begin{bmatrix} .0024 \\ .066 \end{bmatrix}$$

$$F_1, F_3 = \begin{bmatrix} 2.8 \times 10^{-3} & 1.5 \times 10^{-4} \\ 1.5 \times 10^{-4} & 8.2 \times 10^{-2} \end{bmatrix}, \quad F_2 = \begin{bmatrix} 5.4 \times 10^{-2} & 2.6 \times 10^{-2} \\ 2.6 \times 10^{-2} & 8.0 \times 10^{-2} \end{bmatrix}$$

$$[P_z^1]^{-1} = [P_z^3]^{-1} = \begin{bmatrix} 1 & -1.5 \times 10^{-4} \\ -1.5 \times 10^{-4} & .92 \end{bmatrix}, \quad [P_z^2]^{-1} = \begin{bmatrix} .95 & -2.6 \times 10^{-2} \\ -2.6 \times 10^{-2} & .92 \end{bmatrix}$$

The determinants of the covariance matrices are 1.09 and 1.15, respectively.

Figure 5.8.1 shows the ability of the estimator to track the rapidly changing plant. The first configuration change, from  $s_1$  to  $s_2$ , occurs at  $t = 1.05$  seconds. During the next 5 iterations the estimator is giving poor estimates because it has not yet detected a configuration change. At  $t = 1.40$  seconds, detection occurs and the estimator resumes accurate tracking. Similar results occur at times  $t = 3.15, 5.25$  and  $6.65$  seconds, when the other changes occur. The time history of the plant configuration is  $s_1 s_2 s_3 s_2 s_1$ . Figure 5.8.2 compares the velocity estimate with the actual velocity.

The estimation accuracy at times between the configuration changes equals that of a matched Kalman filter and is optimum in that sense. On

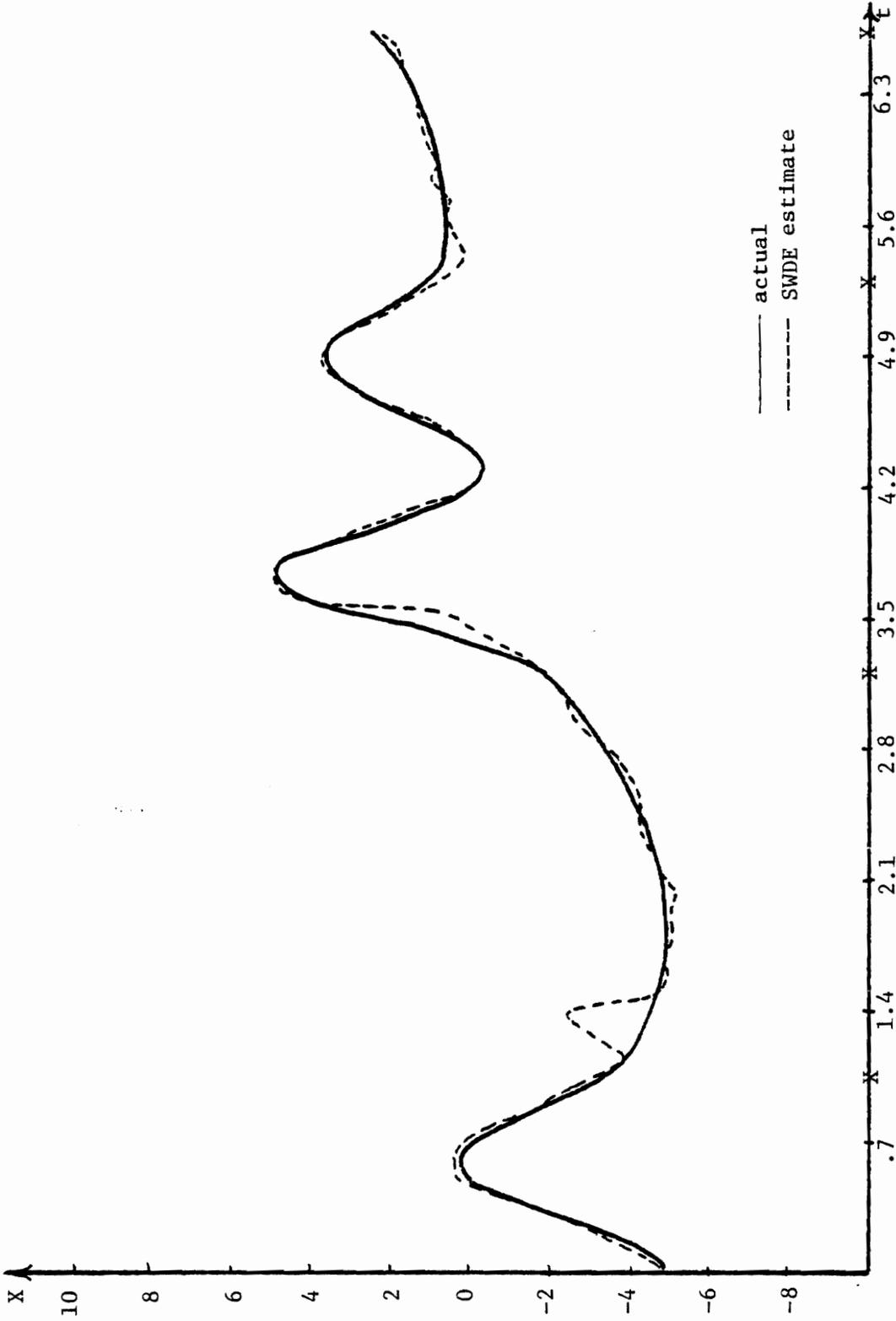


Fig. 5.8.1 Plot of Actual Position and Position Estimate of the SWDE Algorithm. 'X' signifies the times that configuration changes occur.

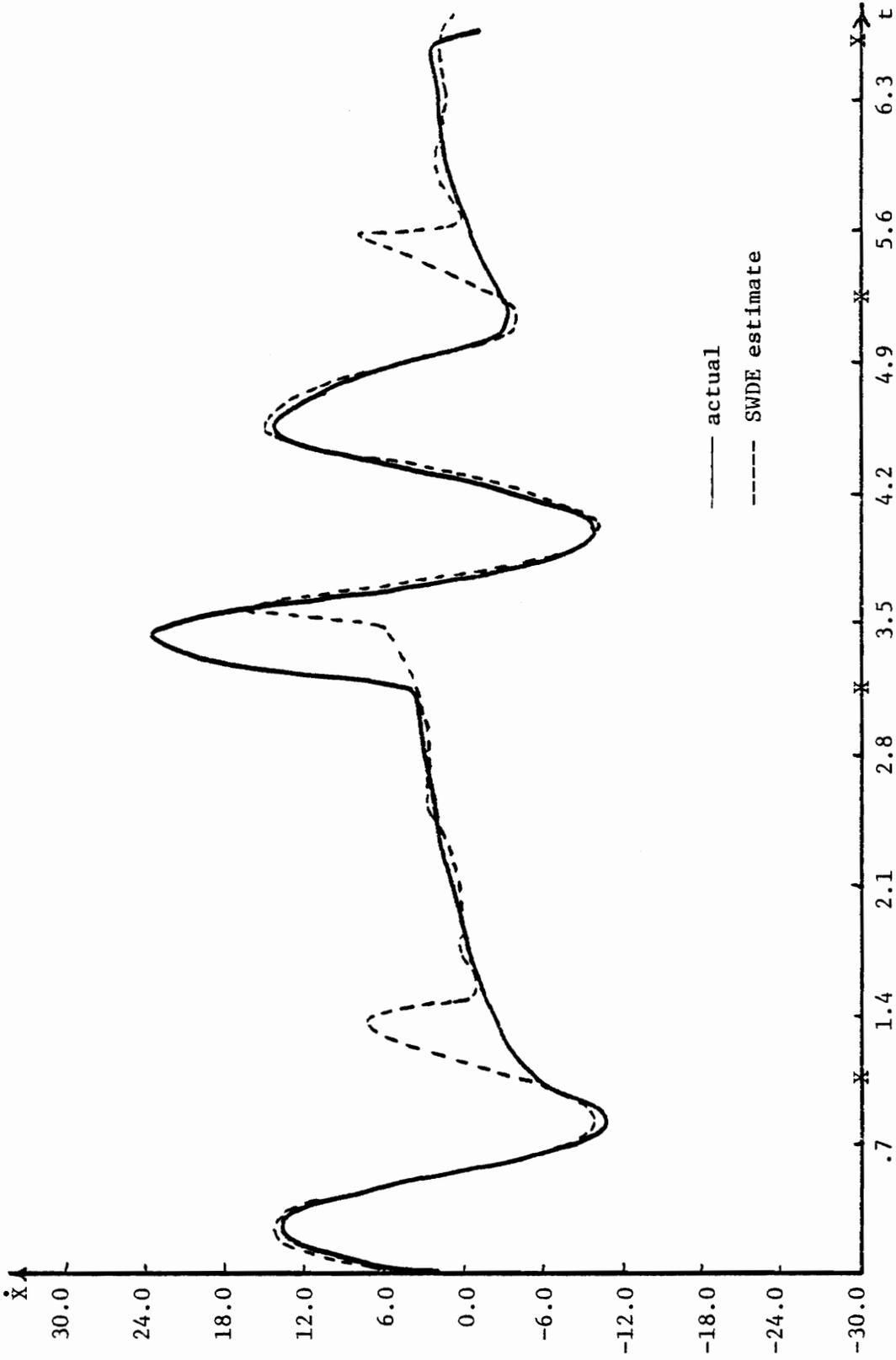


Fig. 5.8.2 Plot of Actual Velocity Versus Velocity Estimate of the SWDE Algorithm in Example 4-1.

the other hand, for the  $N$  iterations after a configuration change the divergence can be substantial, as can be seen in Figure 5.8.1. If the configuration changes have a low frequency of occurrence, then the occasional divergence spikes are of minor importance. However, for frequently occurring configuration changes, the divergence intervals substantially degrade the overall performance.

Two methods exist to reduce and even to eliminate the divergence intervals. The delayed estimate of Section 5.7 can be used if a time delay equal to the window width can be tolerated. This estimate is shown in Figure 5.8.3. Assuming that the estimator stays on track, the only difference between the delayed estimate and the unrealizable optimum estimate is the small error introduced by ignoring the transient interval when the Kalman filter matrices change between steady state values. Table 5.8.1 compares the actual position, optimum position estimate and optimal delayed position estimate during the interval following the first configuration change. It is seen that the maximum percentage difference is approximately two percent.

The weighted sum estimate of Section 5.5 will significantly reduce the divergence interval at the expense of making the overall estimator performance somewhat noisier. For example, the previous plant is shown in Figure 5.8.4a,b for an input, input disturbance covariance, and a measurement noise covariance of,

$$u = 2 \quad Q = 4 \quad R = \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix}.$$

The configuration history of the plant is  $s_3 s_2 s_1 s_2 s_3$ , with the changes occurring at times 1.05, 2.45, 3.85, 5.25 and 6.65 seconds.

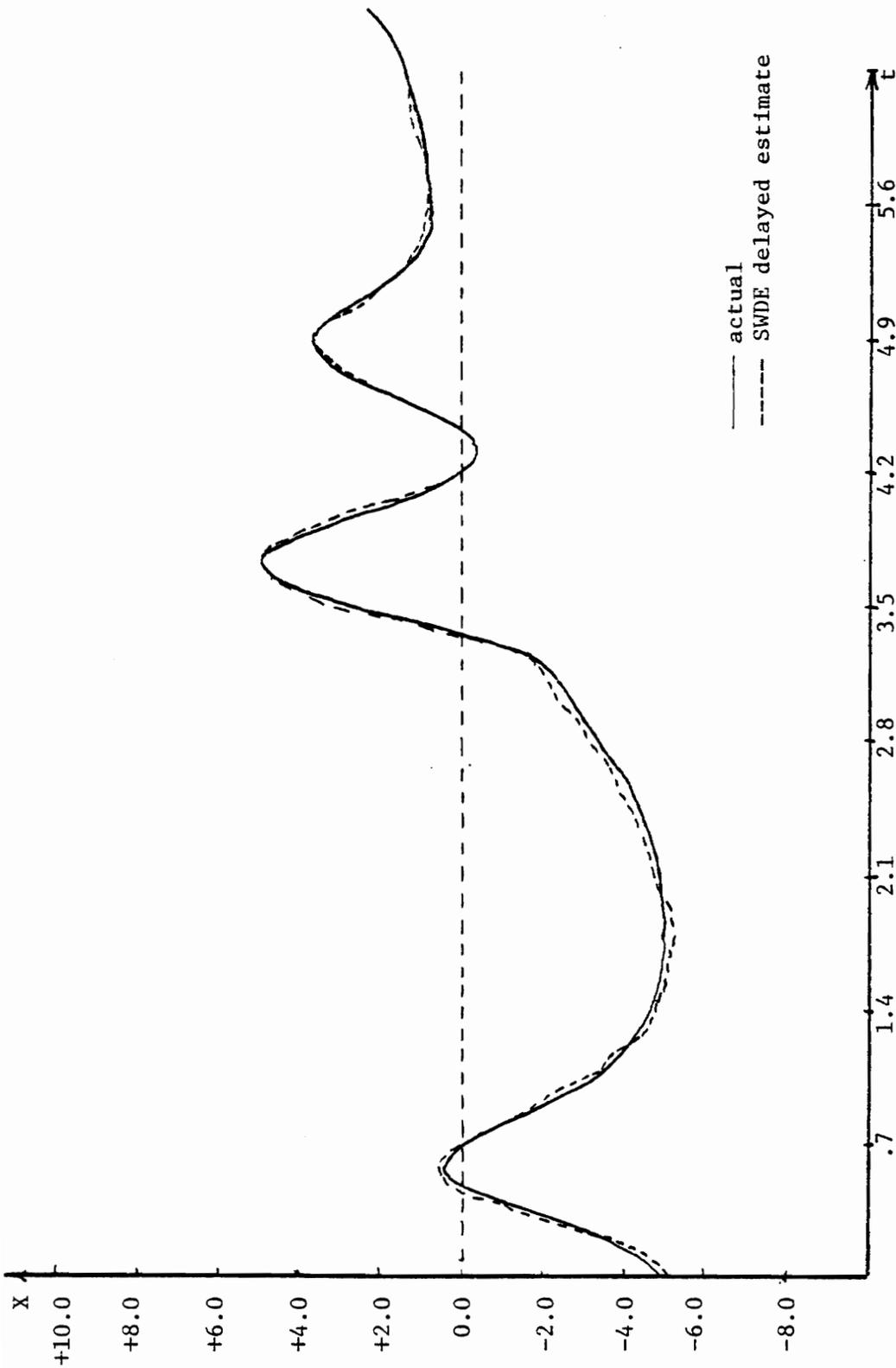
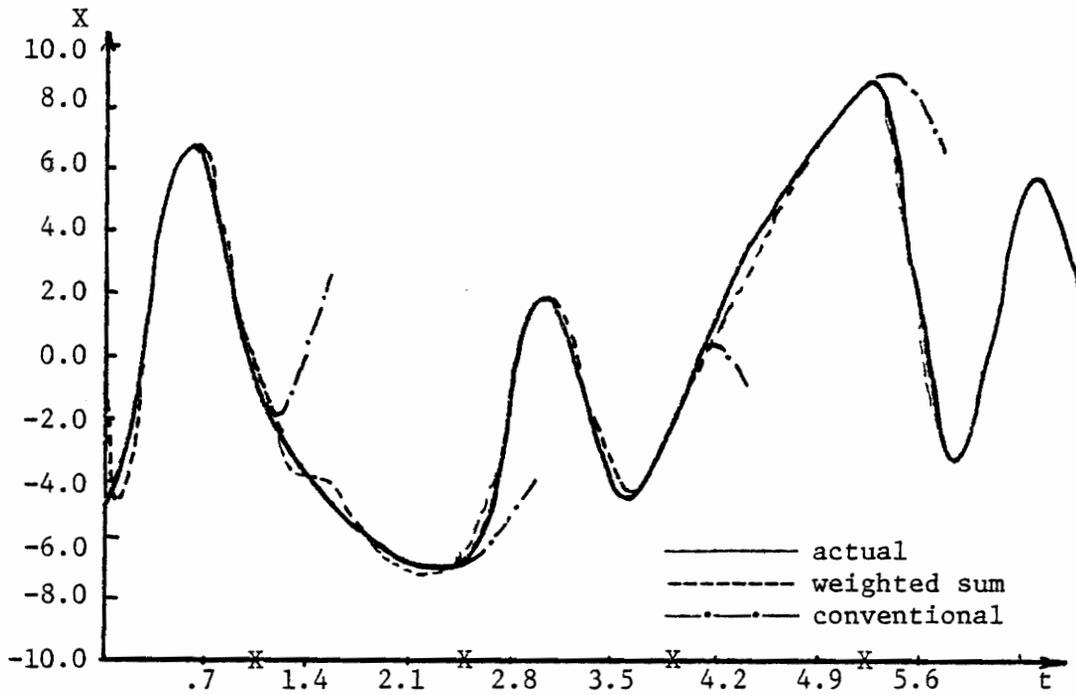
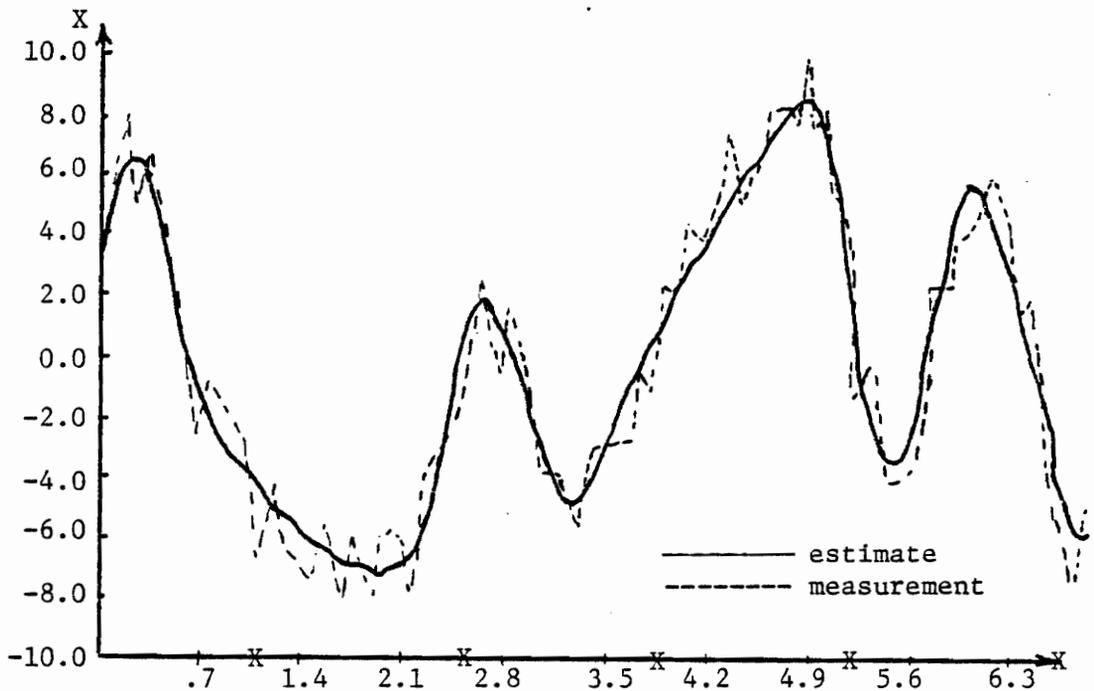


Fig. 5.8.3 Delayed Estimate Compared to the Actual Plant Position in Example 4-1.



(a) Comparison of Weighted Sum Estimate and Conventional SWDE Estimate.



(b) Weighted Sum Estimate Compared to Noisy Position Measurement.

Fig. 5.8.4 Comparison of SWDE Estimates for Example 5-1.

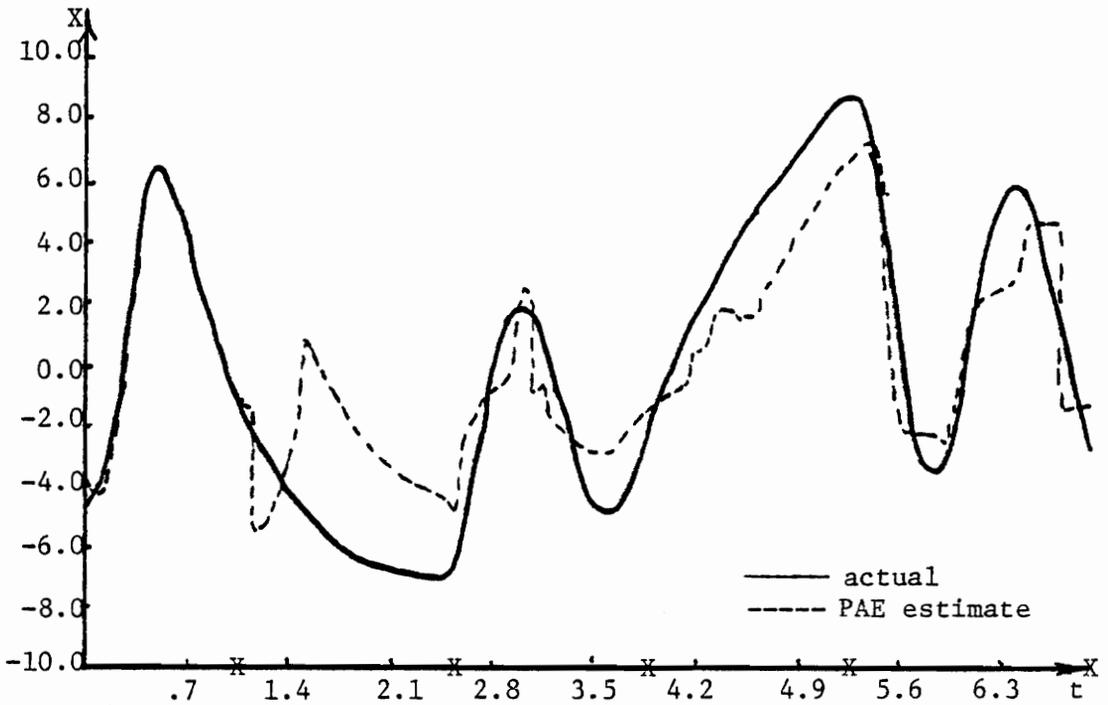
Table 5.8.1 Comparison of Actual Position  
With The Optimum Estimate and The SWDE Delayed Estimate

Time	X-Actual	X-Optimum	X-Delayed
1.05	-3.345	-3.283	-3.283
1.12	-3.653	-3.600	-3.600
1.19	-3.916	-3.894	-3.992
1.26	-4.138	-4.121	-4.172
1.33	-4.335	-4.335	-4.403
1.40	-4.454	-4.488	-4.519

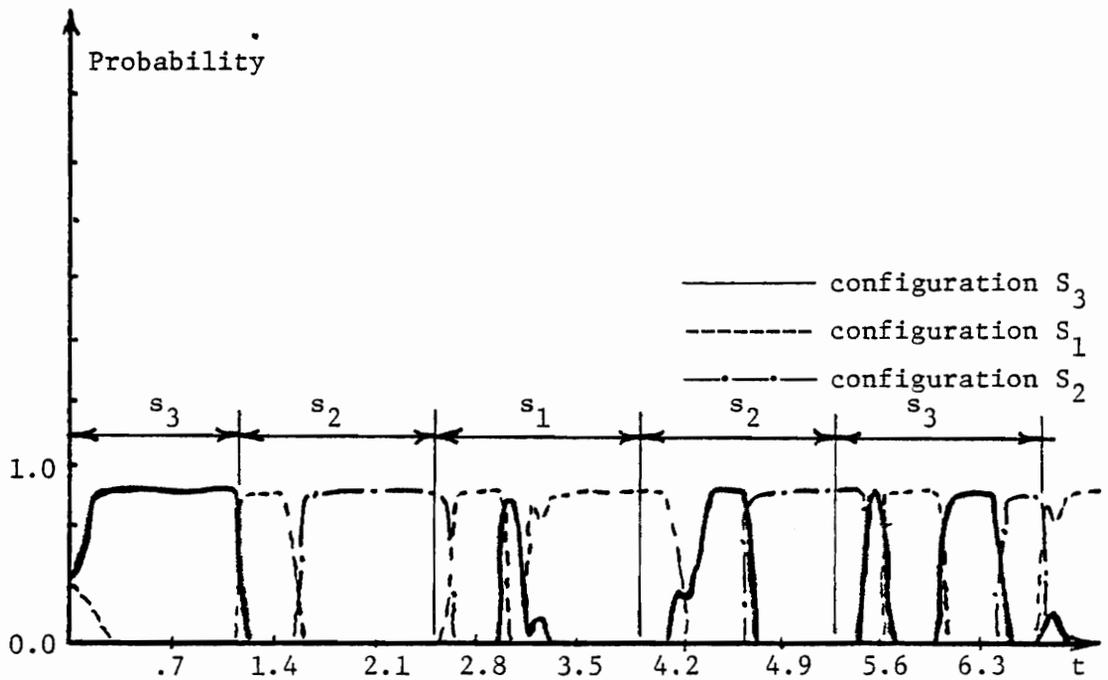
For this noisier condition the required window width was determined to be  $N=9$ . Figure 5.8.4a compares the weighted sum estimate with the actual plant position. The dashed lines indicate the divergence that was eliminated. The weighted sum estimate is compared to the position measurement in Figure 5.8.4b.

It is instructive to contrast the performance of the PAE estimator of Chapter Three with the above results. Recall that the PAE estimator operates satisfactorily for a slowly switching plant but loses track for a higher frequency of configuration changes. This is verified in Figures 5.8.5a,b. In 5.8.5a the PAE estimate of the plant position is highly unreliable after the first configuration change occurs. Figure 5.8.5b shows the erratic behavior of the Kalman filter weighting coefficients.

The operation of the estimator is better understood by looking at the probabilities associated with the various branches at each iteration. Figure 5.8.6 gives probability data before, during and after a configuration change. In order to use a window width of five iterations (for graphical reasons) the measurement noise is reduced to about a third of its previous level. The circles represent the termination points of the branches at each iteration. The enclosed number is the log of the relative probability for the branch. (Zero indicates a probability less than  $10^{-40}$ .) Since every branch is 5 iterations long, the probability is the product of four measurement residual density values, which are printed alongside the branch. The actual branch followed by the plant is shown by a dashed line. It is seen that a configuration change from  $s_3$  to  $s_2$  occurs at the tenth iteration. The measurement residual probabilities along the dashed branch are indeed higher than those along the other



(a) Actual Position Compared to Position Estimate of the PAE Algorithm.



(b) Weighting Coefficients Versus Configuration Sequence.

Fig. 5.8.5 Estimation Performance of the PAE Algorithm of Chap. 3.

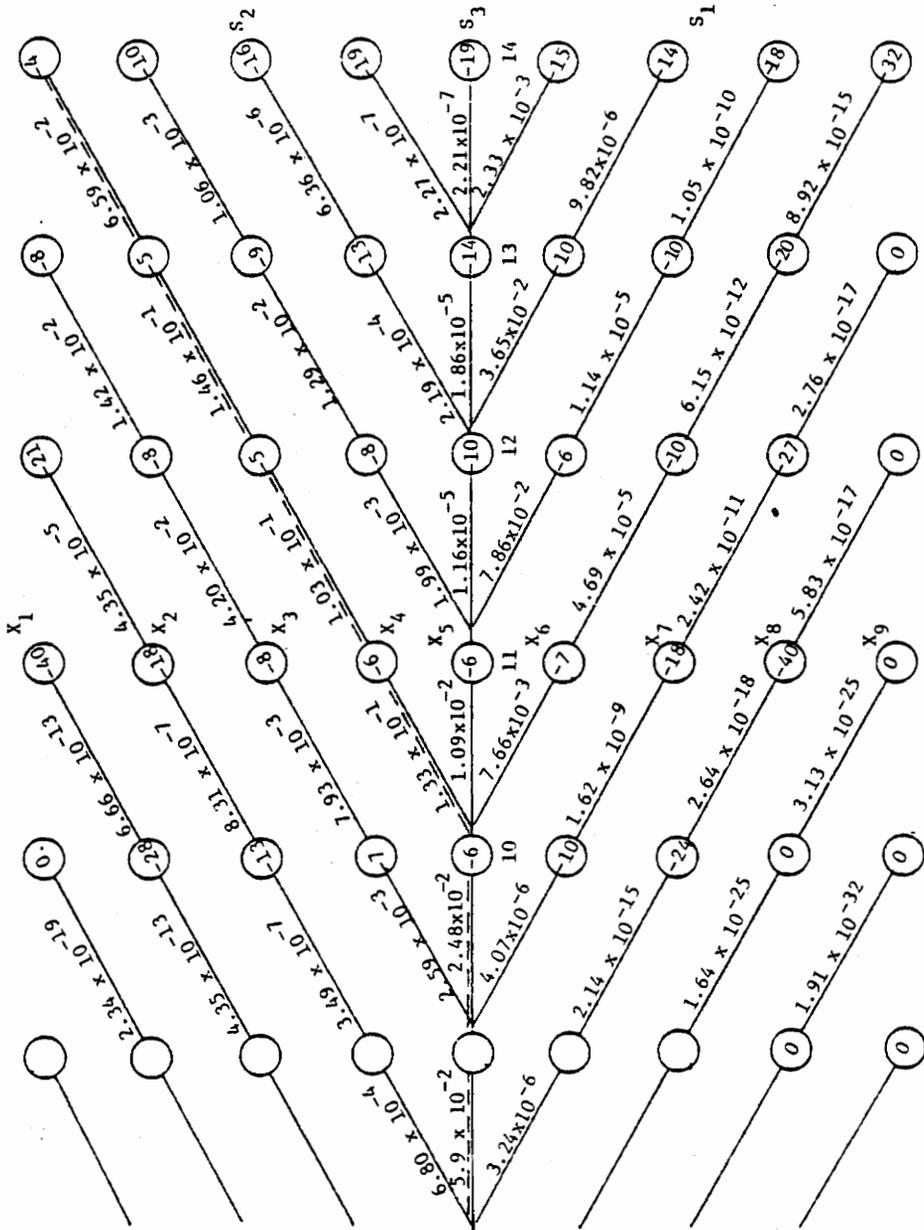


Fig. 5.8.6 Actual Probabilities and Measurement Residual Density Values Of A Tree Having Window Width,  $N = 5$ .

branches.

A decision is made at each iteration regarding possible configuration changes. At iterations 11, 12 and 13 the dashed branch is identified as the correct branch because the relative probability is largest for this branch. However, no configuration change is detected because the relative probabilities at the fifth level are not maximum. For example, at iteration 12, the fifth level probabilities are  $10^{-2}$  for  $s_2$  and 0.0 for  $s_1$ , whereas the current configuration,  $s_3$ , has a relative probability of  $10^{-10}$ . At iteration 14 the estimator decides that the plant changed to configuration  $s_2$  at iteration 10. The state estimate is then taken from the dashed branch, a new tree is grown from this estimate, and the algorithm is ready to detect another configuration change.

The weighted sum estimation method can also be examined using Figure 5.8.6. At iteration 11 the 9 branch estimates are labeled  $x_1$  to  $x_9$ . The weighted sum estimate at this time is calculated as

$$\hat{x}(11) = \frac{10^{-40}\hat{x}_1 + 10^{-18}\hat{x}_2 + 10^{-8}\hat{x}_3 + 10^{-6}\hat{x}_4 + 10^{-6}\hat{x}_5 + 10^{-7}\hat{x}_6 + 10^{-18}\hat{x}_7 + 10^{-40}\hat{x}_8}{10^{-40} + 10^{-18} + 10^{-8} + 10^{-6} + 10^{-7} + 10^{-18} + 10^{-40}}$$

$$\hat{x}(11) \approx (10^{-6}\hat{x}_4 + 10^{-6}\hat{x}_5 + 10^{-7}\hat{x}_6) / 2.1 \times 10^{-6} \quad (5.8.1)$$

In the normal mode of operation, SWDE would use  $x_5$  exclusively for the estimate at iteration 11. The weighted sum estimator produces a better estimate at iteration 11 by incorporating a percentage of the optimal estimate,  $x_4$ ; even though the configuration change has not yet been detected. The weighted sum estimate becomes increasingly more accurate as detection time nears because the natural divergence of the branches

results in a larger relative probability for the correct branch. Unfortunately, the weighted sum estimate is noisier than the SWDE estimate at all times other than the intervals following configuration changes. For example, at iteration 10 the SWDE estimate is optimum whereas the weighted sum estimate is corrupted by percentages of the other branch estimates.

The next experiment illustrates the divergence detection capability of the initialization algorithm described in Section 5.6. The plant of Example 5-1 is intentionally put through rapid configuration changes to cause the SWDE algorithm to lose track. The noise covariance, input, input disturbance covariance and configuration history are

$$u = 2 \quad Q = 4 \quad R = \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix} \quad (s_3 \ s_2 \ s_1 \ s_3)$$

The window width is 9 iterations. The plant remains in configuration  $s_1$  for only 5 iterations. Figure 5.8.7 compares the actual plant position with the PAE estimate (of Chapter 3) and with the weighted sum estimate without the initialization algorithm. The PAE estimate is virtually useless after the first configuration change at time  $t = 1.75$  seconds. The weighted sum estimate is accurate until the changes at  $t = 2.52$  and  $t = 2.87$  occur. These changes cause the weighted sum estimate to diverge.

Figure 5.8.8a shows the improvement in the weighted sum estimate when the initialization algorithm is implemented. Divergence is detected at  $t = 3.71$  seconds, an interval of 1.2 seconds (approximately two window widths) after the onset of the quick configuration changes. At this time, the tracking algorithm is turned off and the PAE algorithm is initialized to re-identify the plant configuration. During the next 15

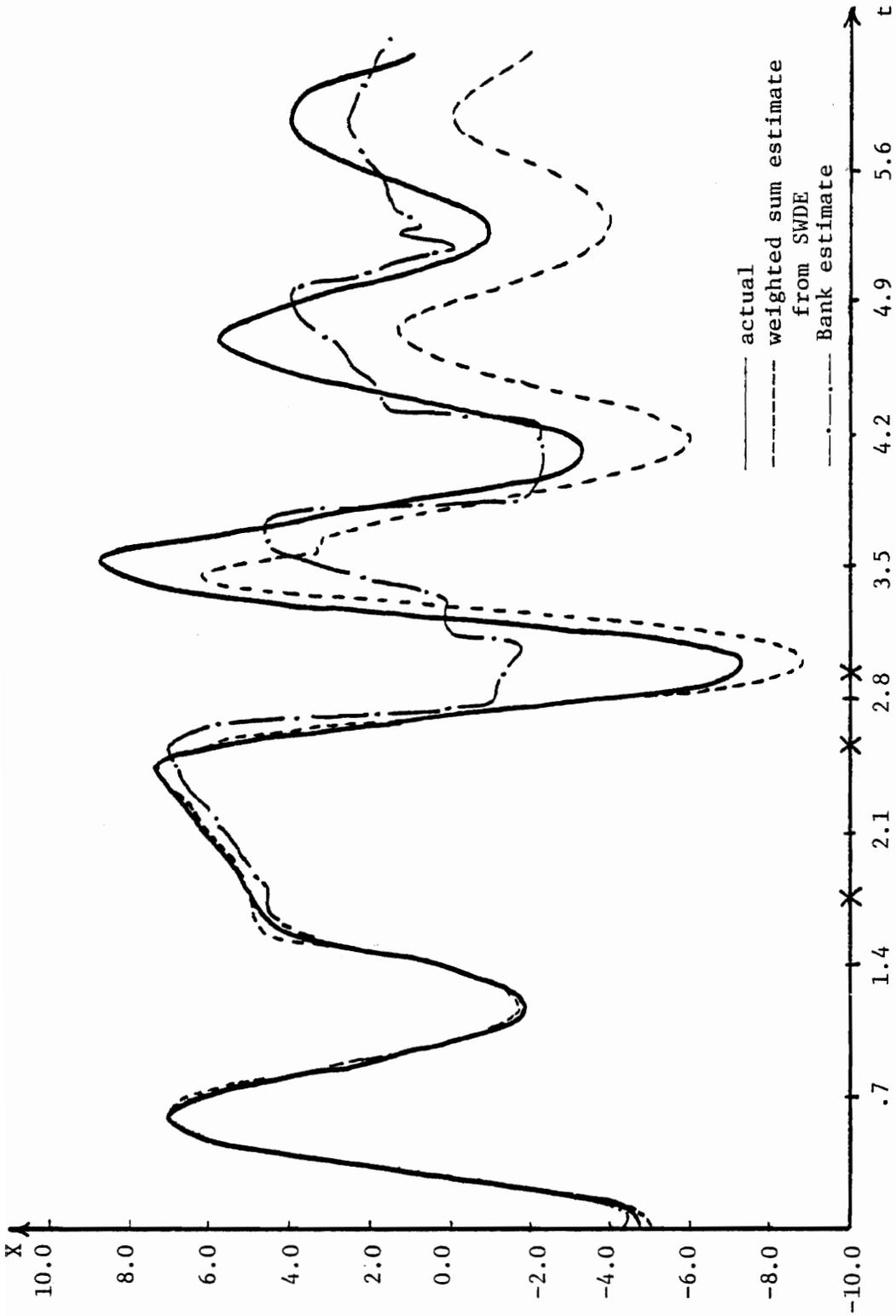
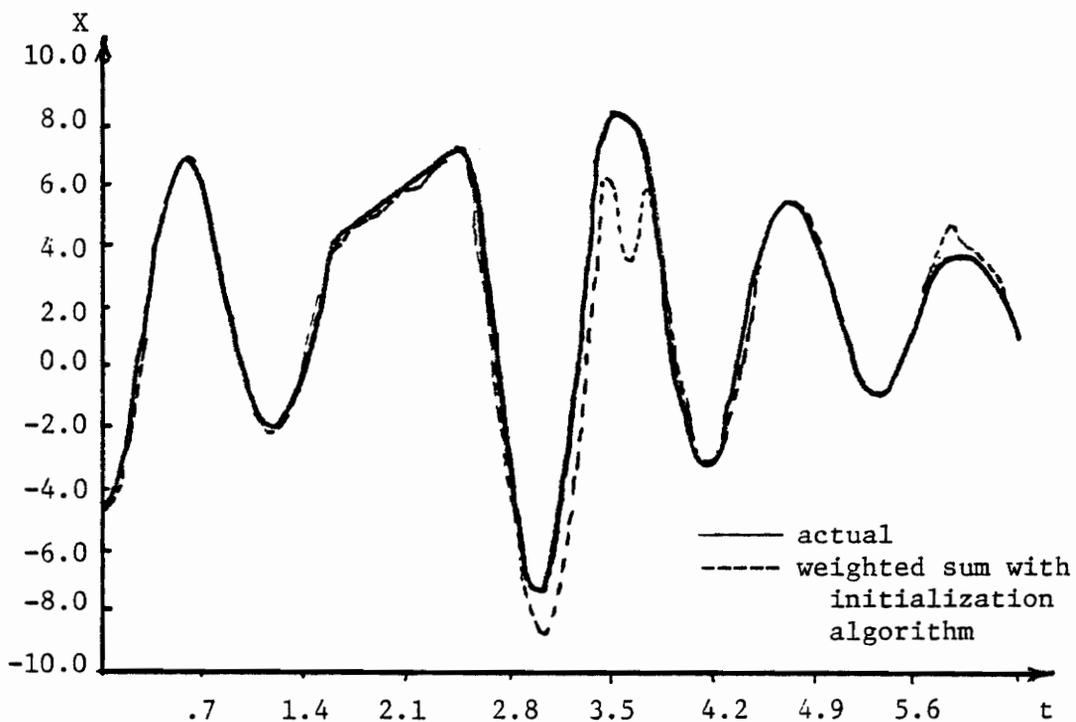
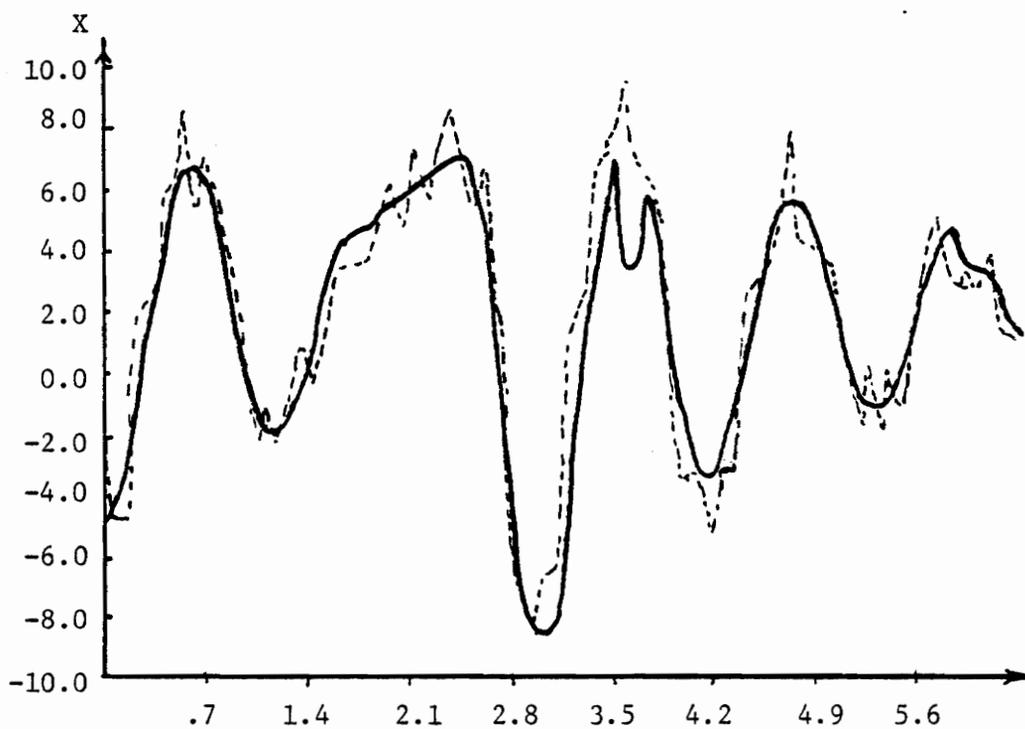


Fig. 5.8.7 Comparison of Estimates After A Sequence of Rapid Configuration Changes. Divergence occurs after  $t = 2.87$  seconds. 'X' denotes the times of the configuration changes.



(a) Actual plant position compared to SWDE estimate.



(b) SWDE estimate compared to noisy measurement.

Fig. 5.8.8 Performance Improvement by Using the Initialization Algorithm to Detect Divergence.

iterations the plant estimate is given by the PAE algorithm. At  $t=4.76$  seconds, the tracking algorithm is turned back on and the weighted sum estimate is again used. The overall tracking ability of this arrangement is very good. Incidentally, the estimation error around  $t=5.8$  seconds is attributable to the inherently noisy weighted sum algorithm. Figure 5.8.8b compares the overall position estimate to the actual position measurement.

Example 5-2:

Consider the second order linear system

$$\begin{aligned} \ddot{x} &= -\beta_1 x - 2\alpha_1 \dot{x} + u + w \\ z_1 &= x + v_1 \quad z_2 = \dot{x} + v_2 \end{aligned} \quad (5.8.2)$$

In state variable notation,

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -\beta_1 & -2\alpha_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w \\ \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \end{aligned}$$

The restoring force factor,  $\beta_1$ , and the damping factor,  $\alpha_1$ , are random variables that can switch between the following possible values.

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1.5 \\ 2 \end{bmatrix}, \begin{bmatrix} .67 \\ .5 \end{bmatrix}$$

Discretizing the three possible state variable equations for a sample

time of .5 seconds gives,

$$\Phi_1 = \begin{bmatrix} .91 & .303 \\ -.303 & .303 \end{bmatrix} \quad \Phi_2 = \begin{bmatrix} .845 & .239 \\ -.477 & .129 \end{bmatrix} \quad \Phi_3 = \begin{bmatrix} .95 & .358 \\ -.179 & .474 \end{bmatrix}$$

$$\Gamma_{1,\Psi_1} = \begin{bmatrix} .0902 \\ .303 \end{bmatrix} \quad \Gamma_{2,\Psi_2} = \begin{bmatrix} .0774 \\ .239 \end{bmatrix} \quad \Gamma_{3,\Psi_3} = \begin{bmatrix} .100 \\ .358 \end{bmatrix}$$

The corresponding steady state Kalman filter matrices are,

$$F_1 = \begin{bmatrix} 2.67 \times 10^{-2} & 7.38 \times 10^{-3} \\ 1.84 \times 10^{-3} & 9.85 \times 10^{-2} \end{bmatrix}, F_2 = \begin{bmatrix} 9.61 \times 10^{-3} & 4.56 \times 10^{-3} \\ 1.14 \times 10^{-3} & 6.22 \times 10^{-2} \end{bmatrix}, F_3 = \begin{bmatrix} 6.26 \times 10^{-2} & 2.00 \times 10^{-2} \\ 5.01 \times 10^{-3} & 1.41 \times 10^{-1} \end{bmatrix}$$

$$P_1^{-1} = \begin{bmatrix} 2.43 \times 10^{-1} & -1.84 \times 10^{-3} \\ -1.84 \times 10^{-3} & 9.02 \times 10^{-1} \end{bmatrix}; P_2^{-1} = \begin{bmatrix} 2.48 \times 10^{-1} & -1.14 \times 10^{-3} \\ -1.14 \times 10^{-3} & 9.38 \times 10^{-1} \end{bmatrix}; P_3^{-1} = \begin{bmatrix} 2.34 \times 10^{-1} & -5.01 \times 10^{-3} \\ -5.01 \times 10^{-3} & 8.59 \times 10^{-1} \end{bmatrix}$$

with determinants 4.56, 4.31 and 4.97, respectively.

The plant simulation uses the following values,

$$u = 12 \quad Q = 1 \quad R = \begin{bmatrix} 22 & 0 \\ 0 & 1 \end{bmatrix} \quad x = \begin{bmatrix} 6 \\ 1 \end{bmatrix}$$

A window width of  $N=9$  was found to give adequate false alarm suppression.

The configuration history, unknown to the SWDE algorithm is  $s_3 s_1 s_3 s_2 s_1 s_2$ . Figure 5.8.9 compares the actual position to the position estimates of the weighted sum algorithm and of the PAE algorithm. As in

Example 5-1, the weighted sum algorithm gives better performance. The

weighted sum estimate is plotted against the actual measurement in

Figure 5.8.10a, and against the conventional SWDE estimate in 5.8.10b.

In the latter plot the conventional estimate is seen to be less noisy

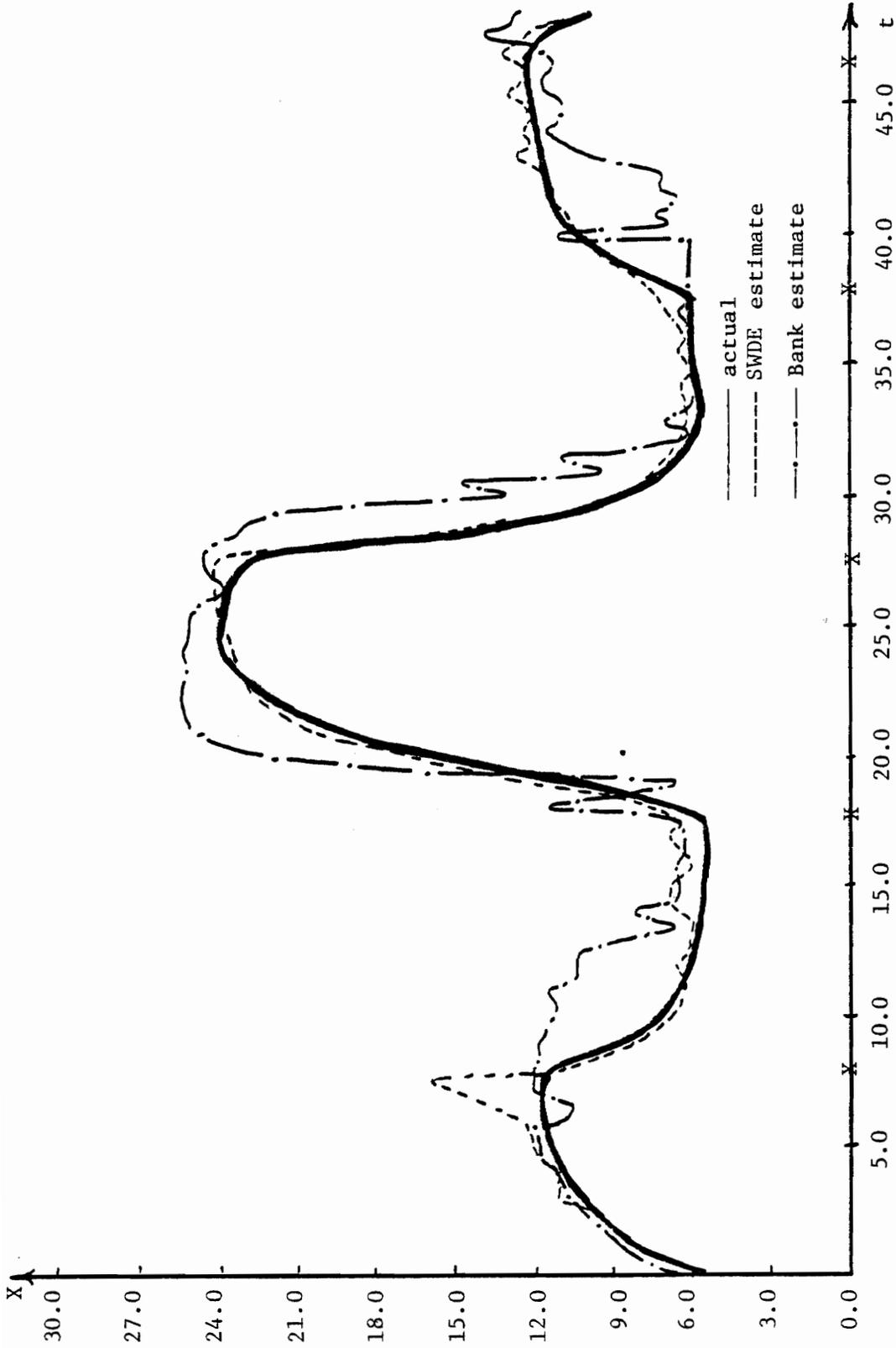
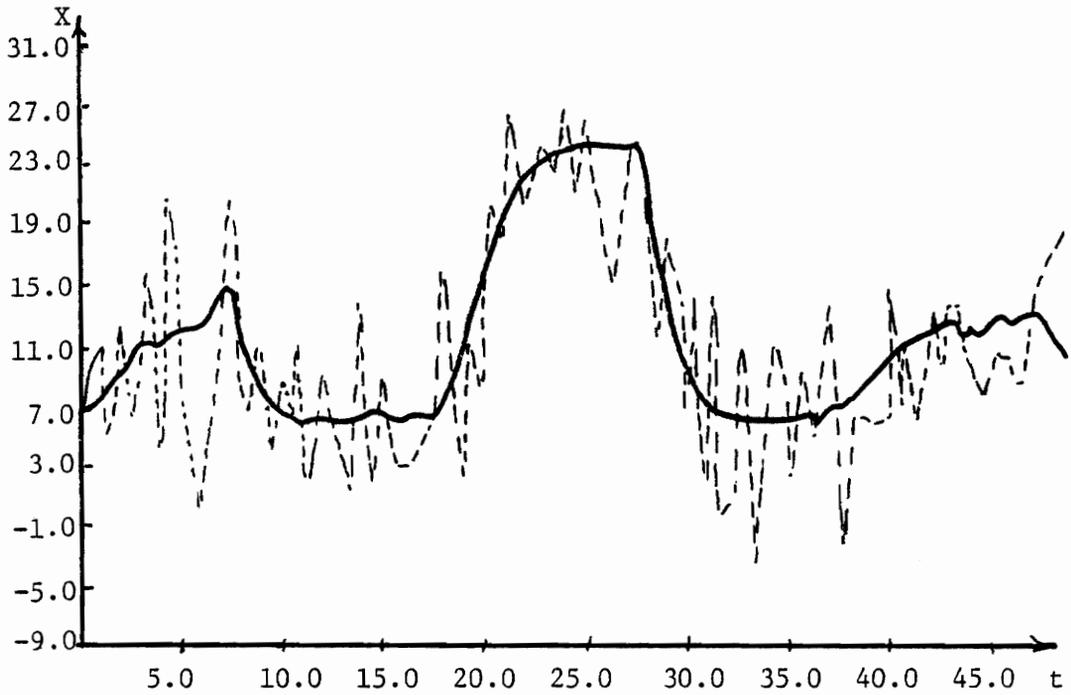
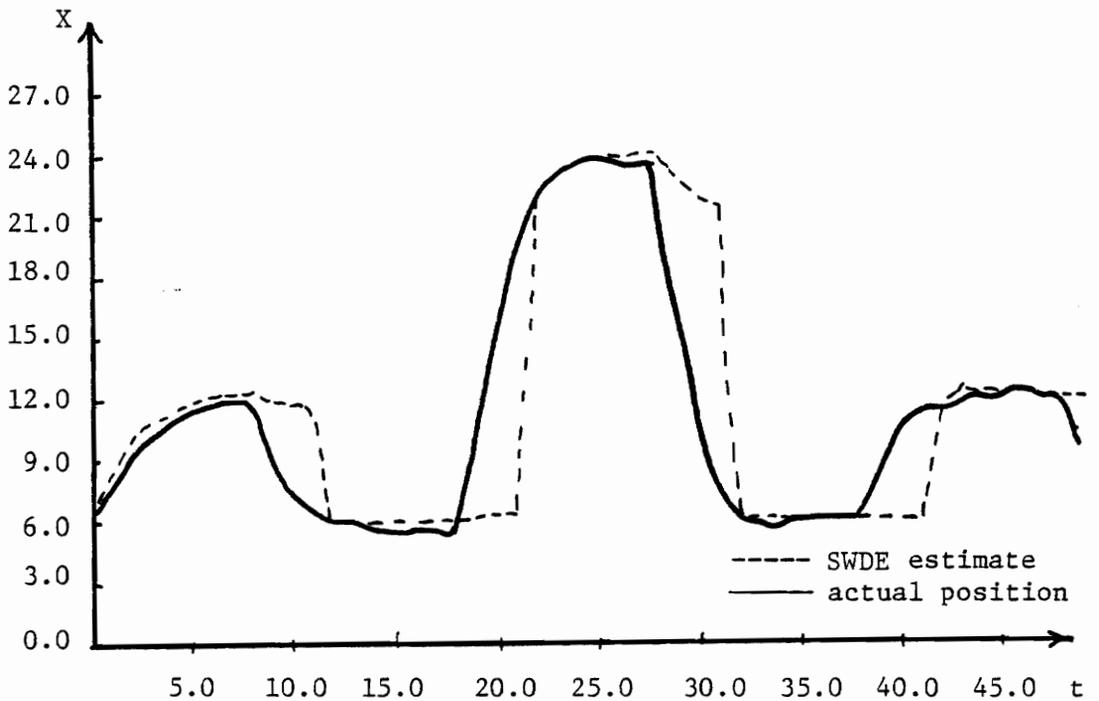


Fig. 5.8.9 Comparison of the Weighted Sum Estimate of the SWDE Algorithm and the Estimate of the Bank Algorithm for the Position of the Plant in Example 4-2. 'X' denotes the times that the configuration changes occur.



(a) Weighted Sum Estimate Plotted Against the Noisy Position Measurement.



(b) SWDE Estimate Without Weighted Sum Algorithm, Plotted Against the Actual Plant Position.

Fig. 5.8.10 Comparison of SWDE Estimates for Example 4-2.

than the weighted sum estimate. However, the divergence intervals make the conventional estimate undesirable.

The previous examples have shown that the SWDE algorithm in conjunction with the initialization algorithm gives excellent tracking when the dynamics of the various plant configurations are known exactly. The next experiment will show the ability of the SWDE algorithm to track a plant whose configurations are not known exactly. Specifically, the plant will be the nonlinear oscillator of Chapter Three, and the configurations will be the linearized plant approximations derived in that chapter. Thus, the configuration changes are a result of the state trajectory moving from one configuration's region of applicability to that of another configuration's.

Example 5-3:

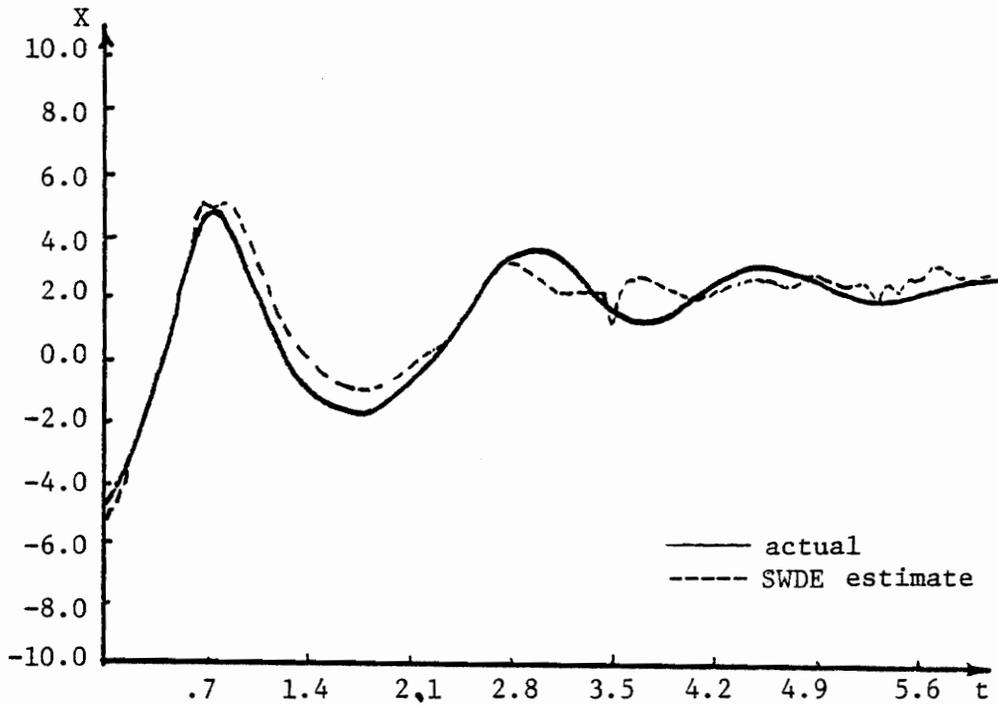
The nonlinear plant is given by,

$$\begin{aligned}\ddot{x} + \dot{x} + x^3 &= u + w \\ z &= Hx + v\end{aligned}\tag{5.8.3}$$

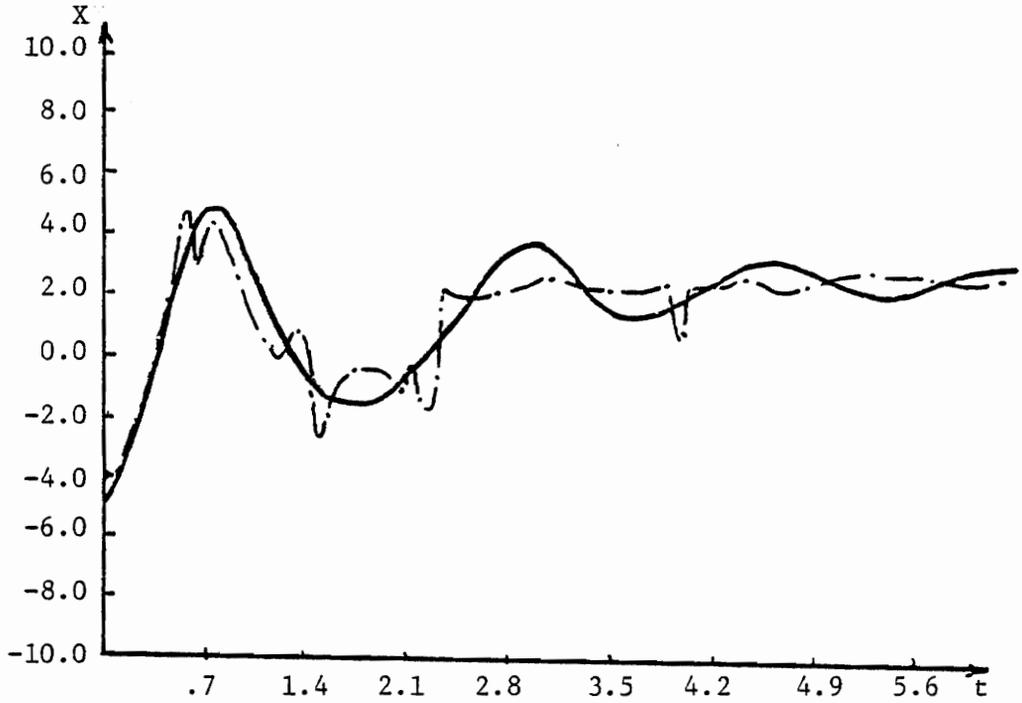
The three configurations are the same as those in Example 5-1. The noise covariance, input and input disturbance covariance are:

$$R = \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix} \quad u = 12 \quad Q = 4$$

Figure 5.8.11a compares the weighted sum estimate to the actual plant position. For contrast, the position tracking of the PAE estimator of Chapter Three is shown in 5.8.11b. The weighted estimate tracks the position better than the PAE estimate, except perhaps when the plant

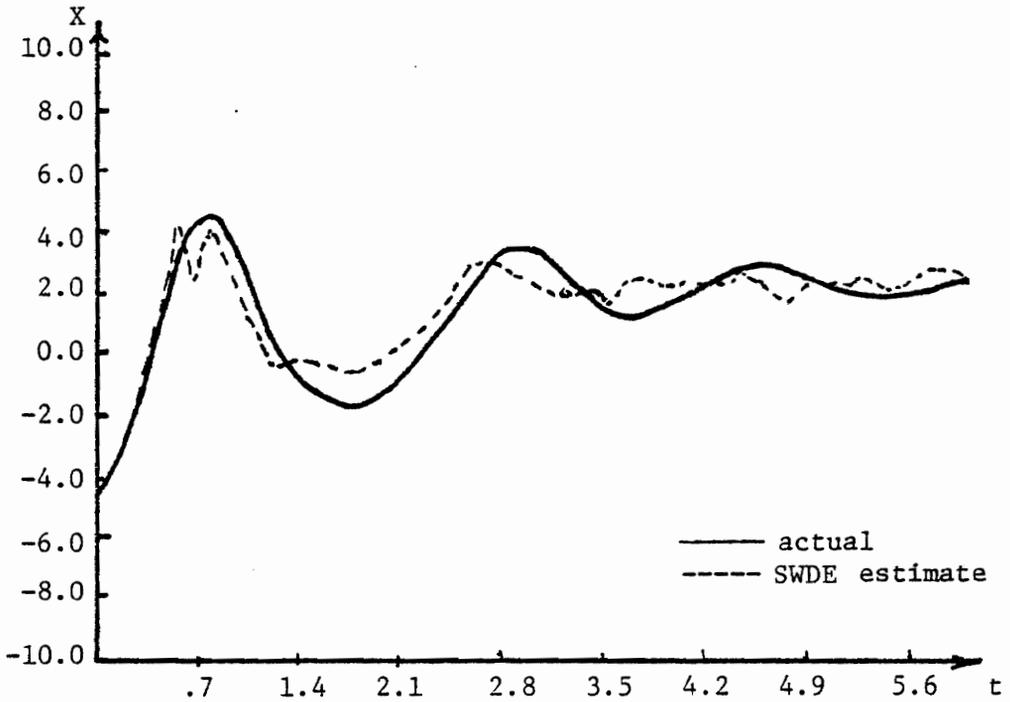


(a) SWDE Position Estimate (Weighted Sum)

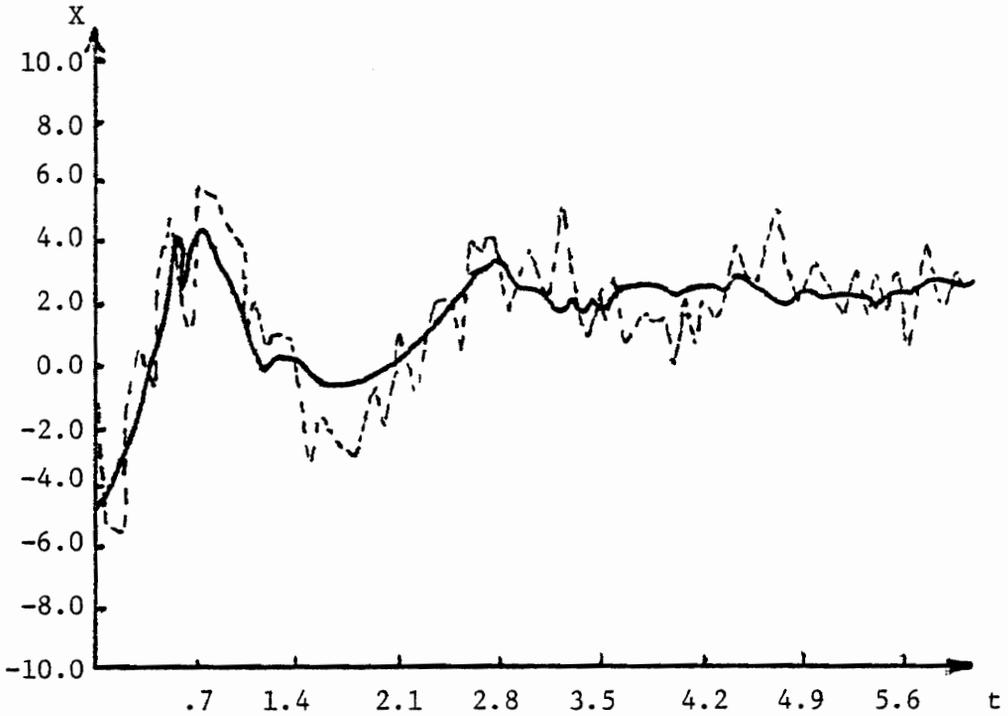


(b) PAE Algorithm Estimate

Fig. 5.8.11 Position Estimates for Nonlinear Plant of Example 5-3.



(a) SWDE Estimate (Without Weighted Sum) and With the Initialization Algorithm



(b) SWDE Estimate (Weighted Sum) with Initialization Algorithm Plotted Against the Noisy Measurement.

Fig. 5.8.12 Nonlinear Estimation Using the SWDE Algorithm

position has settled out. Both estimates are about equal at this time. Using the SWDE algorithm without the weighted sum estimation gives the result shown in Figure 5.8.12aa. The divergence intervals around times  $t = .35, .9, 1.4, 2.8$  and  $3.85$  significantly degrade the tracking performance. The only improvement is in the settled out region where the estimates are less noisy than the weighted sum estimates. Figure 5.8.12b compares the weighted sum estimate with the actual position measurement.

### 5.9 Summary

The SWDE algorithm in conjunction with the initialization algorithm provides excellent tracking performance for a switched linear plant. In cases where the configuration changes occur frequently with respect to the window width, tracking can be improved by altering the SWDE algorithm to produce a weighted sum estimate. The disadvantage of the weighted sum estimate is that it is inherently noisier during those intervals when configuration changes are not occurring.

The SWDE algorithm also performs quite well in tracking a nonlinear plant. In this case the weighted sum estimate usually gives better tracking performance. The threshold level of the initialization algorithm should be set higher when tracking nonlinear plants. This is to avoid re-initializing the algorithm due to the natural biases that will be present because of modeling mismatch.

## 6.0 ADAPTIVE CONTROL OF THE B737 AIRCRAFT

### 6.1 Introduction

In recent years there has been a steady increase in the complexity and sophistication of aircraft designs. With this complexity follows the need for added sophistication in controller design. Typically, the equations of motion of an aircraft are linearized about some nominal flight condition so that feedback and cross-control gains may be determined to satisfy flying quality criteria [M-10]. For some conventional aircraft these gains are often adequate for the entire flight envelope and are thus held constant throughout the flight. The control technique reported in this chapter is directed toward aircraft with nonlinear aerodynamics that are not satisfactorily controlled by constant feedback gains. An additional benefit of this method will be the reduction of control difficulties encountered during the landing and take-off phases of flight which oftentimes are critical to the safety of the passengers.

This chapter applies the modified PAC algorithm of Chapter Four to the longitudinal dynamics of the B737 aircraft. The motivation for applying adaptive control to the B737 aircraft is due to the NASA "Terminal Configured Vehicle", TCV, program. One goal of this program is to study control methods that will enable quicker and safer landing approaches in adverse weather conditions. The results given in this chapter indicate that the modified PAC algorithm will help accomplish this goal.

Section 6.2 describes the longitudinal dynamics of the B737 aircraft. The need for adaptive control is shown in the flight simulation

results of Section 6.3. The model-following design method is used in Section 6.4 to obtain the feedback and input gains required to meet the control objectives. Section 6.5 analyzes the nonlinear measurements that are available to the controller. The simulation of the B737 aircraft is described and the closed-loop response results are given in Section 6.6. The method and results are summarized in Section 6.7.

## 6.2 The Equations of Motion

This section describes the longitudinal perturbation equations of the B737 aircraft. The three input controls that are available to the controller are the elevator, stabilizer rate and throttle rate. The differential equations relating these inputs to the delivered thrust and stabilizer position and the earth reference variables required to monitor the aircraft's position will be used with the equations of motion to obtain an overall state variable description for the system.

The longitudinal equations of motion for an aircraft are, after simplification, [H-4][R-1][E-1]

$$\begin{aligned} m(\dot{u} + W_0 q) &= -mg\theta \cos\theta_0 + f_{ax} + f_{Tx} \\ m(\dot{w} - U_0 q) &= -mg\theta \sin\theta_0 + f_{az} + f_{Tz} \\ I_{yy} \dot{q} &= m_a + m_T \end{aligned} \tag{6.2.1}$$

where

$U_0$  = steady state inertial speed, x-direction

$W_0$  = steady state inertial speed, z-direction

$\theta_0$  = steady state pitch angle

$u$  = speed perturbation, x-direction

$w$  = speed perturbation, z-direction

$\theta$  = pitch angle perturbation

$q$  = pitch rate perturbation

$f_{ax}, f_{az}$  = Aerodynamic Force perturbations

$f_{Tx}, f_{Tz}$  = thrust perturbations

$m_a, m_T$  = perturbation in pitching moment due to Aerodynamic forces and thrust.

$m, g, I_{yy}$  = mass, gravity and moment of inertia around the y-axis.

#### Assumptions:

- (1) All lateral variables have very small values (i.e. yaw, roll, sideslip and their rates).
- (2) Pitch angle is small.
- (3) The above equations are linearizations of the nonlinear equations of motion around  $U_0, W_0, \theta_0$ .
- (4) The perturbations in the angles are small so that  $\cos \theta \approx 1$ ,  $\sin \theta \approx \theta$ .

The aerodynamic forces and moments involved are the lift and drag forces and the pitching moment; the effects of thrust are described separately by the terms with subscript T. These forces and moments can be expressed as follows:

$$L = C_L(u, \alpha, \dot{\alpha}, q, \delta e, \delta s) \bar{q} S$$

$$D = C_D(u, \alpha, \dot{\alpha}, q, \delta e, \delta s) \bar{q} S$$

$$T = C_T(u, \alpha, \dot{\alpha}, q) \bar{q} S$$

$$M = C_M(u, \alpha, q, e, s, T) \bar{q} \bar{S} \bar{c} \quad (6.2.2)$$

$$\bar{q} = \frac{1}{2} \rho V_a^2$$

where

$\alpha$  = angle of attack perturbation

$S$  = effective aerodynamic surface area

$\rho$  = atmospheric density

$\bar{q}$  = dynamic pressure

$V_a$  = airspeed

$\bar{c}$  = mean aerodynamic wing chord

$\delta e, \delta s, \delta T$  = perturbations in elevator, stabilizer and thrust

$C_L, C_D, C_T, C_M$  = nonlinear lift coefficients

Linearizing the aerodynamic and thrust forces, and regrouping the small perturbation variables into state variable format, the following equation is established with respect to the stability axis coordinate system [H-4][R-1][E-1]:

$$\dot{\mathbf{x}}_p = \mathbf{A} \mathbf{x}_p + \mathbf{B} \mathbf{r}_p + \mathbf{C} \mathbf{w}_p \quad (6.2.3)$$

where

$$\mathbf{x}_p^T = [u \quad \alpha \quad q \quad \theta]^T$$

$$\mathbf{r}_p^T = [\delta e \quad \delta T \quad \delta s]^T$$

$$\mathbf{w}_p^T = [u_g \quad \alpha_g \quad q_g]$$

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (6.2.4)$$

$$B = \begin{bmatrix} 0 & b_{12} & 0 \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \\ 0 & 0 & 0 \end{bmatrix} \quad (6.2.5)$$

$$C = \begin{bmatrix} c_{11} & c_{12} & 0 \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \\ 0 & 1 & 0 \end{bmatrix} \quad (6.2.6)$$

and,  $u_g$  = gust perturbation in x-inertial speed

$\alpha_g$  = gust perturbation in angle of attack

$q_g$  = gust perturbation in pitch rate

$x_p$  = state vector for equations of motion

$r_p$  = input vector for equations of motion

$w_p$  = gust input vector for equations of motion

A = system coefficient matrix

B = input coefficient matrix

C = gust input coefficient matrix

The coefficients in the state equation are nonlinear functions of several different parameters, such as the aircraft weight, altitude, flap

setting, center of gravity, etc., and their calculation, even for just one flight condition is quite involved. Fortunately, this problem was circumvented by taking advantage of the extensive literature and analysis routines available for the B737 aircraft at NASA/LRC.

The "TCVA Trim Data Validation" program and the accompanying state variable matrix generation program were used extensively to obtain the A, B and C matrices for several hundred different flight conditions. For example, see Figure 6.2.1. (Appendix B contains information outlining the use of these programs.) From an analysis of these linearized models, ten were selected that represented a variety of flight conditions from 10,000 feet to sea level. See Figure 6.2.2.

The thrust will be modeled as having a first order time-lag with respect to the throttle, and the throttle rate will be the actual control input. The stabilizer rate will also be available as a control input. The elevator response time is assumed to be instantaneous. Thus,

$$\begin{aligned}\delta\dot{T} &= -.5\delta T + .5\delta th & (6.2.7) \\ \delta\dot{th} &= r_{th} \\ \delta\ddot{s} &= r_s \\ \delta e &= r_e\end{aligned}$$

where,

$\delta th$  = throttle perturbation

$r_{th}$  = throttle rate input

$r_s$  = stabilizer rate input

$r_e$  = elevator perturbation input.

In order to locate the aircraft with respect to the earth reference

$$\begin{bmatrix} \dot{u} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -0.047 & .082 & 0. & -.152 & 0. & 0. & 0. & 0. \\ -0.310 & -.756 & 1.0 & 0. & 0. & 0. & 0. & 0. \\ -0.142 & -1.07 & -.529 & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 1. & 0. & 0. & 0. & 0. & 0. \end{bmatrix} \begin{bmatrix} u \\ \alpha \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} 0. \\ -.047 \\ -1.20 \\ 0. \end{bmatrix} \delta e + \begin{bmatrix} \delta e \\ \delta T \\ \delta S \end{bmatrix} + \begin{bmatrix} -.047 & .082 & 0. \\ -.310 & -.756 & -.042 \\ -.142 & -1.07 & -.434 \\ 0. & 1. & 0. \end{bmatrix} \begin{bmatrix} u_g \\ \alpha_g \\ q_g \end{bmatrix}$$

Parameters Specified:

- Altitude = 200 feet
- Velocity = 125 knots
- Flaps = 40 degrees
- Flight path angle = 0 degrees
- Gear = down
- Weight = 80,000 lbs.
- C.G. = .25
- Banking angle = 0 degrees

Figure 6.2.1 Example of State Equation Matrices

Weight = 80,000 lbs.  
C.G. = .25  
Flight Path Angle = 0°  
Banking Angle = 0°

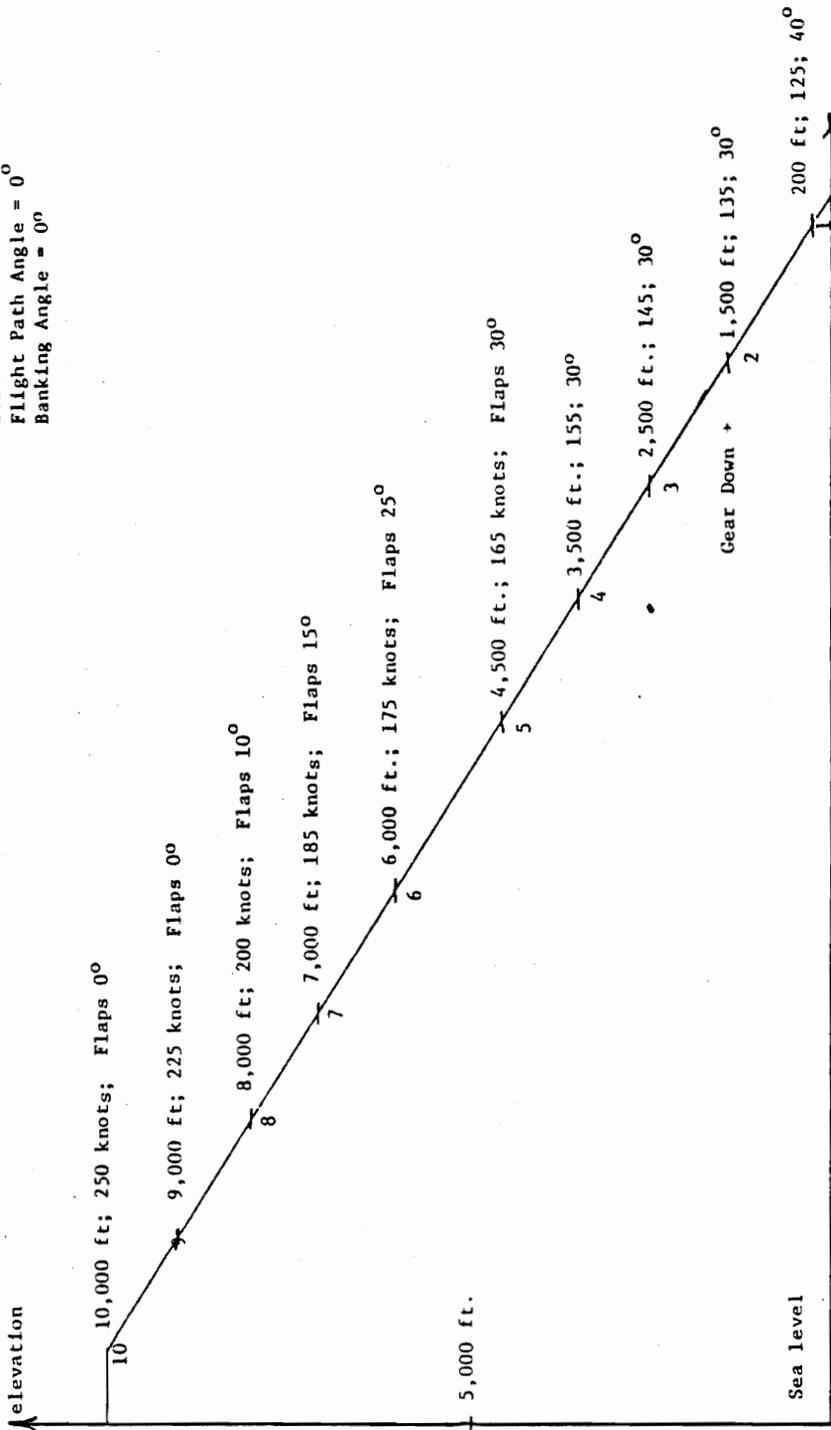


Figure 6.2.2 Ten Flight Conditions Used in Simulation

frame, two additional state variables are required. These variables are the altitude loss (sink) perturbation and the closing position perturbation given by

$$\begin{aligned}\dot{z} &= -\left(\frac{Z-Z_0}{U_0}\right) \approx u(-\sin\gamma_0) + \alpha(\cos\gamma_0) + \theta(-\cos\gamma_0) \\ \dot{x} &= \left(\frac{X-X_0}{U_0}\right) \approx u(\cos\gamma_0) + \alpha(\sin\gamma_0) + \theta(-\sin\gamma_0)\end{aligned}\tag{6.2.8}$$

where,

$z$  = sink perturbation

$x$  = closing position perturbation

$Z$  = altitude

$Z_0$  = nominal altitude

$X$  = x-position of aircraft in earth coordinate system

$X_0$  = nominal x-position

$\gamma_0$  = steady state flight path angle,  $\theta_0 - \alpha_0$ .

Incorporating equations (6.2.7) and (6.2.8) into equation (6.2.3) gives,

$$\dot{x}_a = A_a x_a + B_a r_a + C w_p\tag{6.2.9}$$

where

$$x_a^T = [X_p^T \quad \delta T \quad \delta th \quad \delta S]^T$$

$$r_a^T = [r_e \quad r_{th} \quad r_s]^T$$

$$A_a = \begin{bmatrix} A & | & B_T & 0 & B_S \\ \hline 0 & | & -.5 & .5 & 0 \\ 0 & | & 0 & 0 & 0 \\ 0 & | & 0 & 0 & 0 \end{bmatrix}\tag{6.2.10}$$

$$A_a = \begin{bmatrix} -.047 & .082 & 0. & -.152 & 0. & 0. & 1.9 & 0. & 0. \\ -.310 & -.756 & 1. & 0. & 0. & 0. & -.042 & 0. & .097 \\ -.142 & -1.07 & -.529 & 0. & 0. & 0. & -.19 & 0. & 2.51 \\ 0. & 0. & 1. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 1. & 0. & -1. & 0. & 0. & 0. & 0. & 0. \\ 1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & -.5 & .5 & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \end{bmatrix}$$

$$B_a = \begin{bmatrix} 0. & 0. & 0. \\ -.047 & 0. & 0. \\ -1.2 & 0. & 0. \\ 0. & 0. & 0. \\ 0. & 0. & 0. \\ 0. & 0. & 0. \\ 0. & 0. & 0. \\ 0. & 1. & 0. \\ 0. & 0. & 1. \end{bmatrix}$$

$$C = \begin{bmatrix} -.047 & .082 & 0. \\ -.31 & -.756 & -.042 \\ -.142 & -1.07 & -.434 \\ 0. & 0. & 0. \\ 0. & 0. & 0. \\ 0. & 0. & 0. \\ 0. & 0. & 0. \\ 0. & 0. & 0. \\ 0. & 0. & 0. \end{bmatrix}$$

Figure 6.2.3 Augmented System Matrices

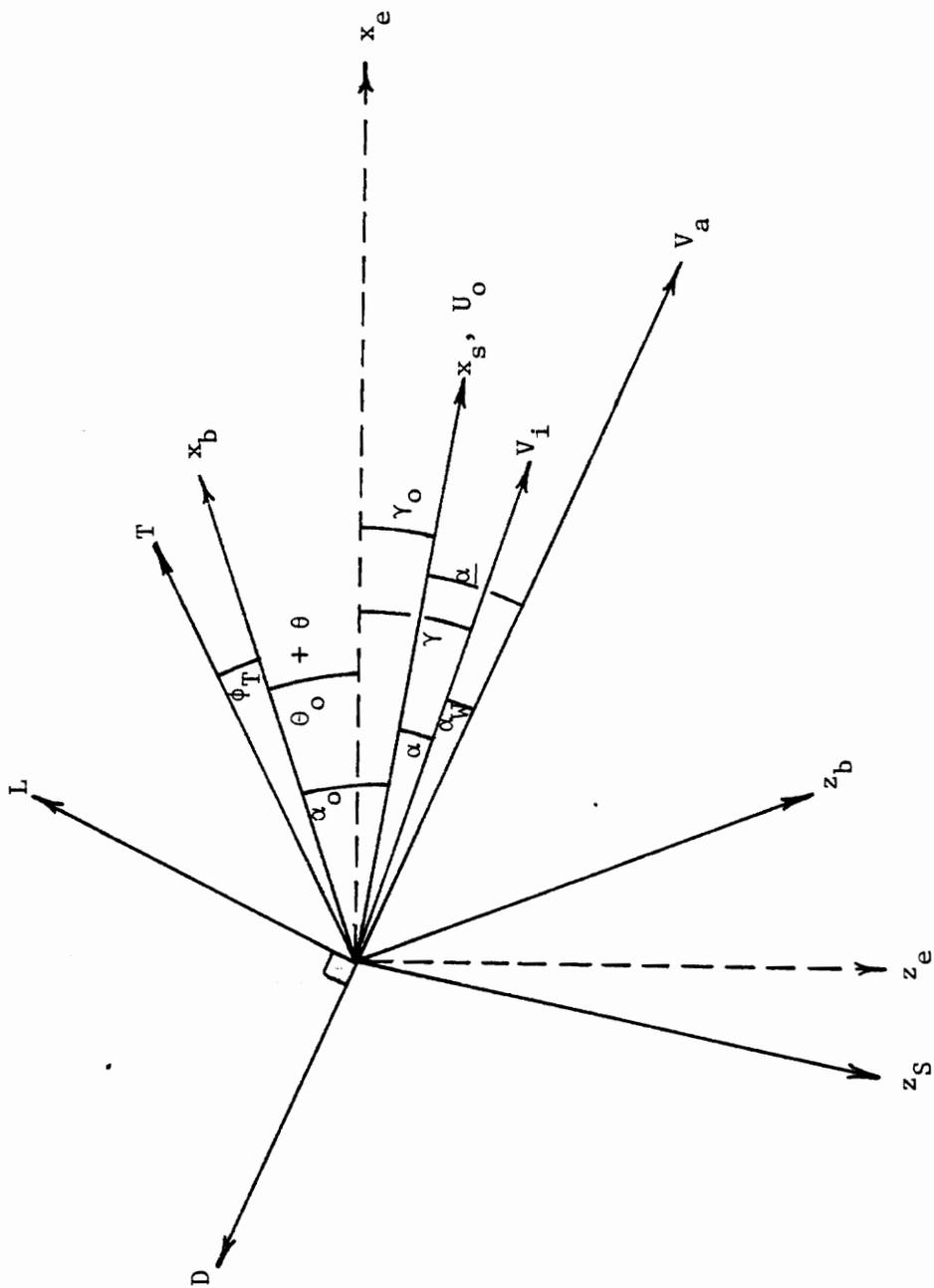


Figure 6.2.4 Definition of Coordinate Axes, Angles and Forces

Table 6.3.1 Longitudinal Eigenvalue Variation  
For The Ten Flight Conditions

REGION	ALTITUDE	SHORT MODE	PHUGOID
1	200	$-.65 \pm J1.03$	$-.014 \pm J.150$
2	1500	$-.91 \pm J1.04$	$-.014 \pm J.088$
3	2500	$-.75 \pm J1.30$	$-.009 \pm J.140$
4	3500	$-.96 \pm J1.40$	$-.010 \pm J.070$
5	4500	$-1.01 \pm J1.50$	$-.009 \pm J.056$
6	6000	$-.87 \pm J1.60$	$-.008 \pm J.110$
7	7000	$-.85 \pm J1.33$	$-.006 \pm J.100$
8	8000	$-.85 \pm J1.44$	$-.006 \pm J.095$
9	9000	$-.86 \pm J1.70$	$-.003 \pm J.095$
10	10000	$-.94 \pm J1.90$	$-.004 \pm J.085$

$$B_a = \begin{bmatrix} B_e & 0 & 0 \\ \hline 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where,

$x_a$  = augmented state vector

$r_a$  = augmented input vector

$A_a$  = augmented system coefficient matrix

$B_a$  = augmented input coefficient matrix

$B_e$  = first column of B, corresponds to elevator

$B_T$  = second column of B, corresponds to thrust

$B_s$  = third column of B, corresponds to stabilizer.

For example, the overall system's state equation corresponding to Figure 6.2.1 is shown in Figure 6.2.3. For clarification, the relationship between various angles and velocity vectors is shown in Figure 6.2.4.

### 6.3 The Need for Adaptive Control

The ten flight conditions listed in the preceding section are now analyzed and simulated. Table 6.3.1 shows the longitudinal eigenvalue variation as the aircraft goes from region to region. The short period mode and the phogoid mode eigenvalues vary approximately 100% and 200%, respectively.

The typical open-loop response of the B737 given a step elevator command is shown in Figure 6.3.1. For this particular example it is seen that the short mode has a period of 3 seconds, and the phugoid has

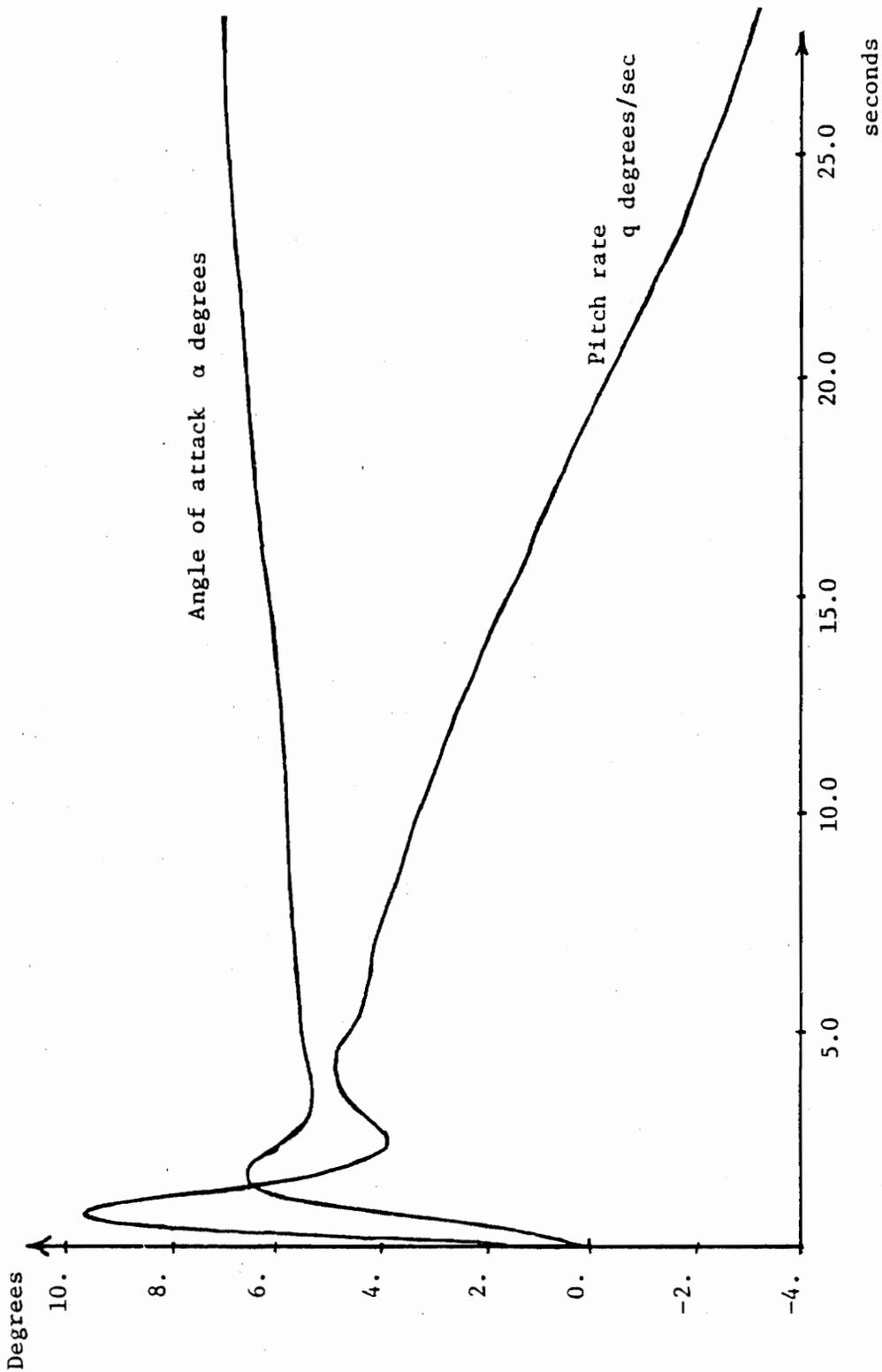


Figure 6.3.1.1 Open Loop Response to A 10-Degree Elevator Step Input

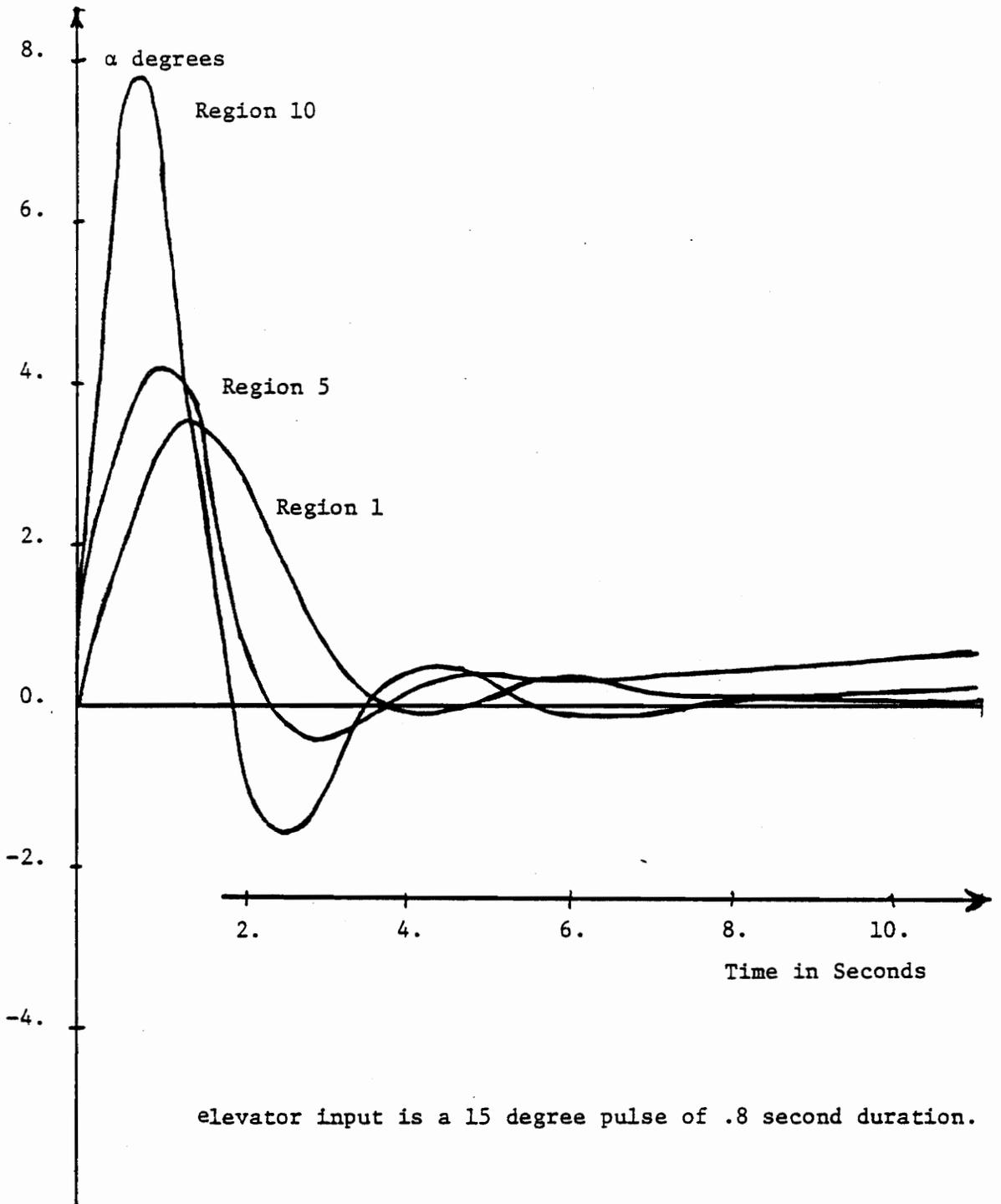
a period of 2 minutes. For different flight conditions the response of the aircraft varies. Figure 6.3.2 shows that the angle of attack response varies considerably in magnitude, frequency and damping in the different regions.

The need for an adaptive controller can be seen in Figures 6.3.3, 6.3.4 and 6.3.5. In Figure 6.3.3 the feedback gains were computed to give the linear model in Region 1 short-period poles at  $S = -3$ . Applying these gains when the aircraft is in the other regions results in either a prolonged oscillation or instability. Similar unacceptable performance results when the gains calculated for Region 5 or Region 10 are used for all regions. See Figures 6.3.4 and 6.3.5, respectively. In addition, even if each region was correctly compensated, there is still a need for region dependent input scaling, as can be seen in Figure 6.3.6a,b,c, which gives the elevator response required to capture and follow a constant sink rate glideslope in different regions and shows the increased efficiency of the elevator at high velocities. Figure 6.3.6c is obviously unrealizable.

#### 6.4 Closed-Loop Design Via Model Following

The control objectives are as follows. First, the aircraft should follow a predetermined glideslope. Second, the response of the aircraft should be as uniform as possible in the various flight configurations. It was decided that these objectives could best be achieved through the methods of model following [T-2][W-5][M-9][A-3].

Model following is an application of optimal regulator theory which is summarized in Chapter Two. The desired goal is to determine feedback



elevator input is a 15 degree pulse of .8 second duration.

Figure 6.3.2 Angle of attack,  $\alpha$ , response for different flight conditions.

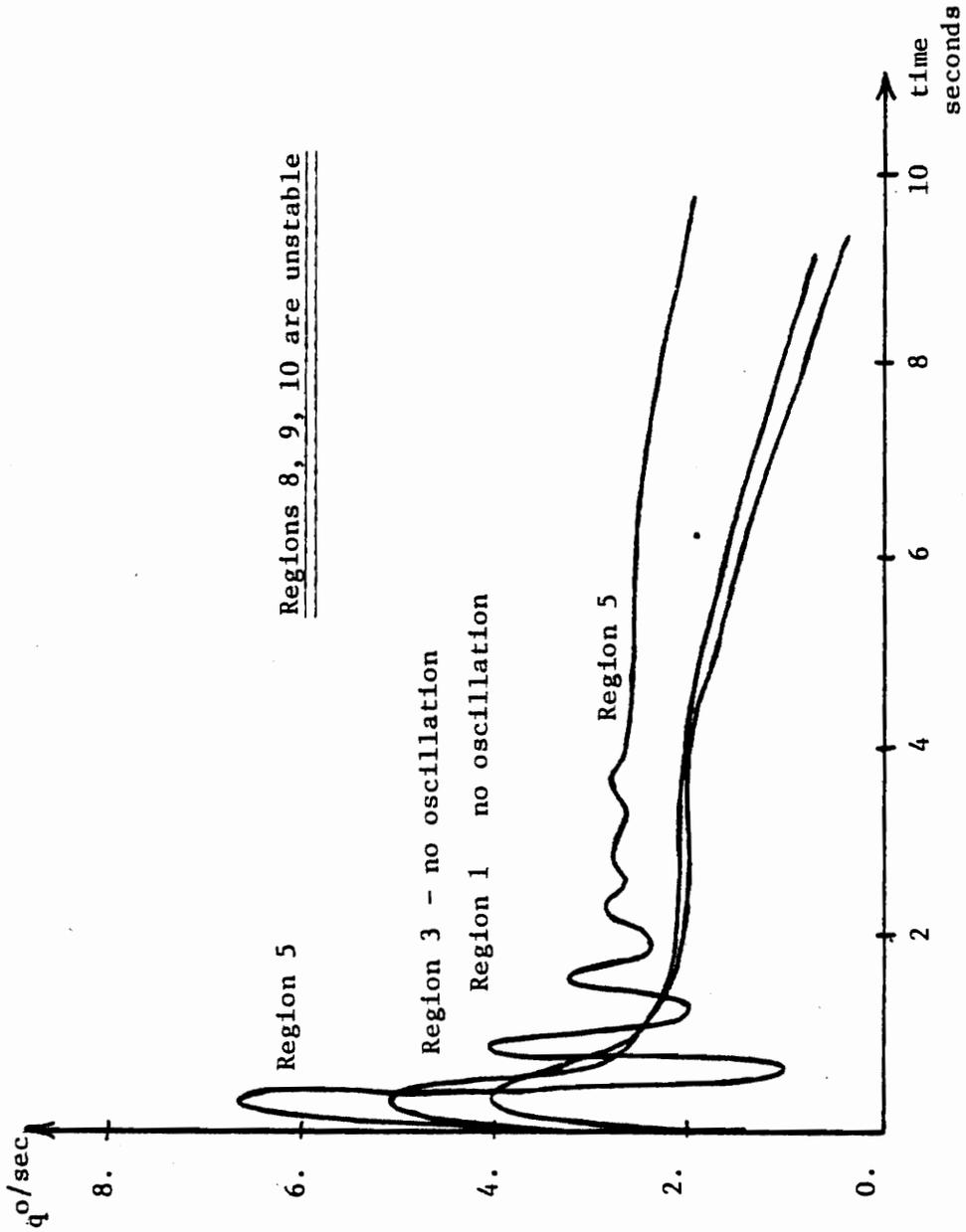


Figure 6.3.3 Closed-Loop Pitch Rate Response in Various Regions Using Feedback Gains Derived for Region 1

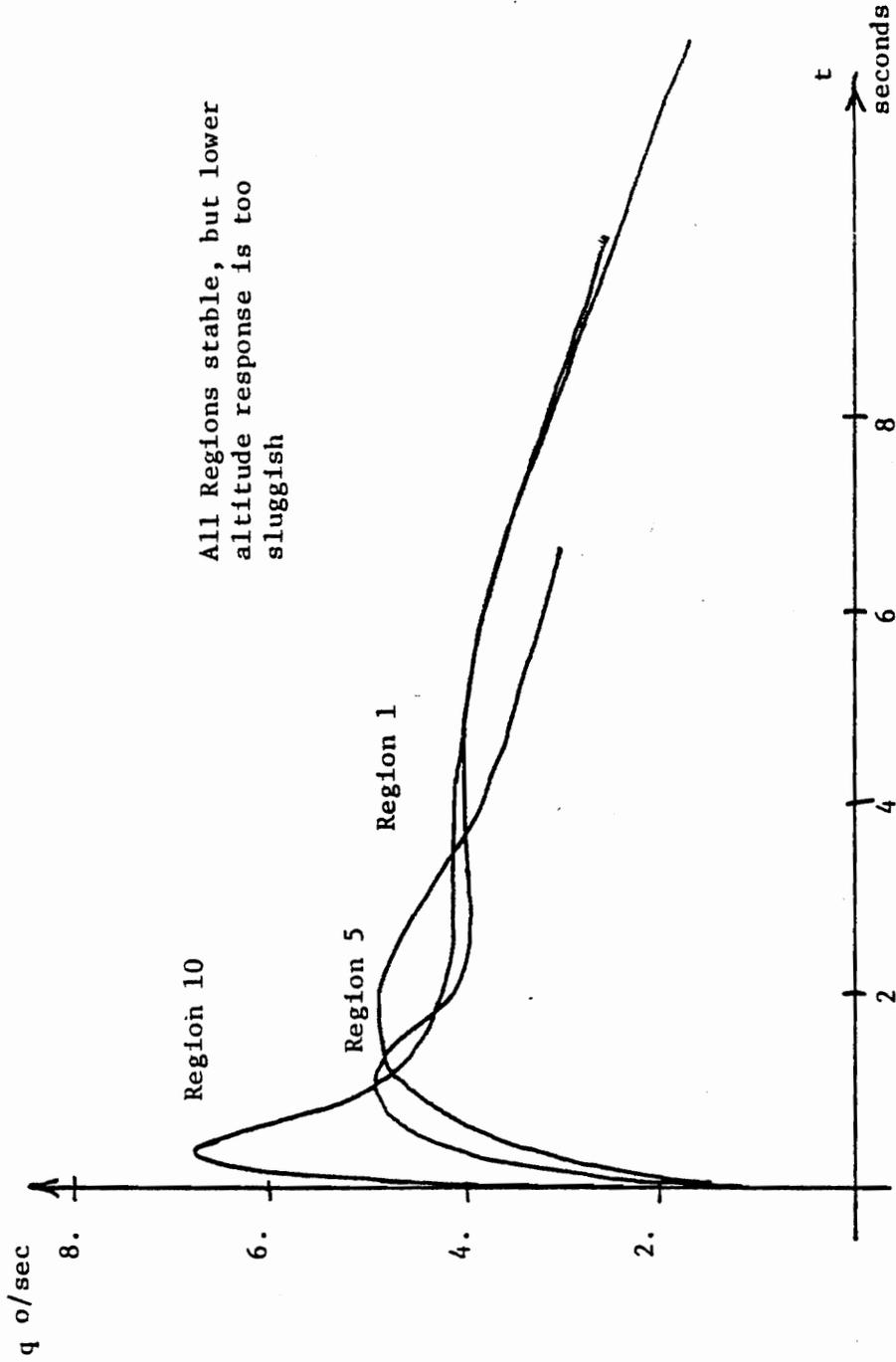


Figure 6.3.4 Closed-Loop Response Using Feedback Gains Derived for Region 10

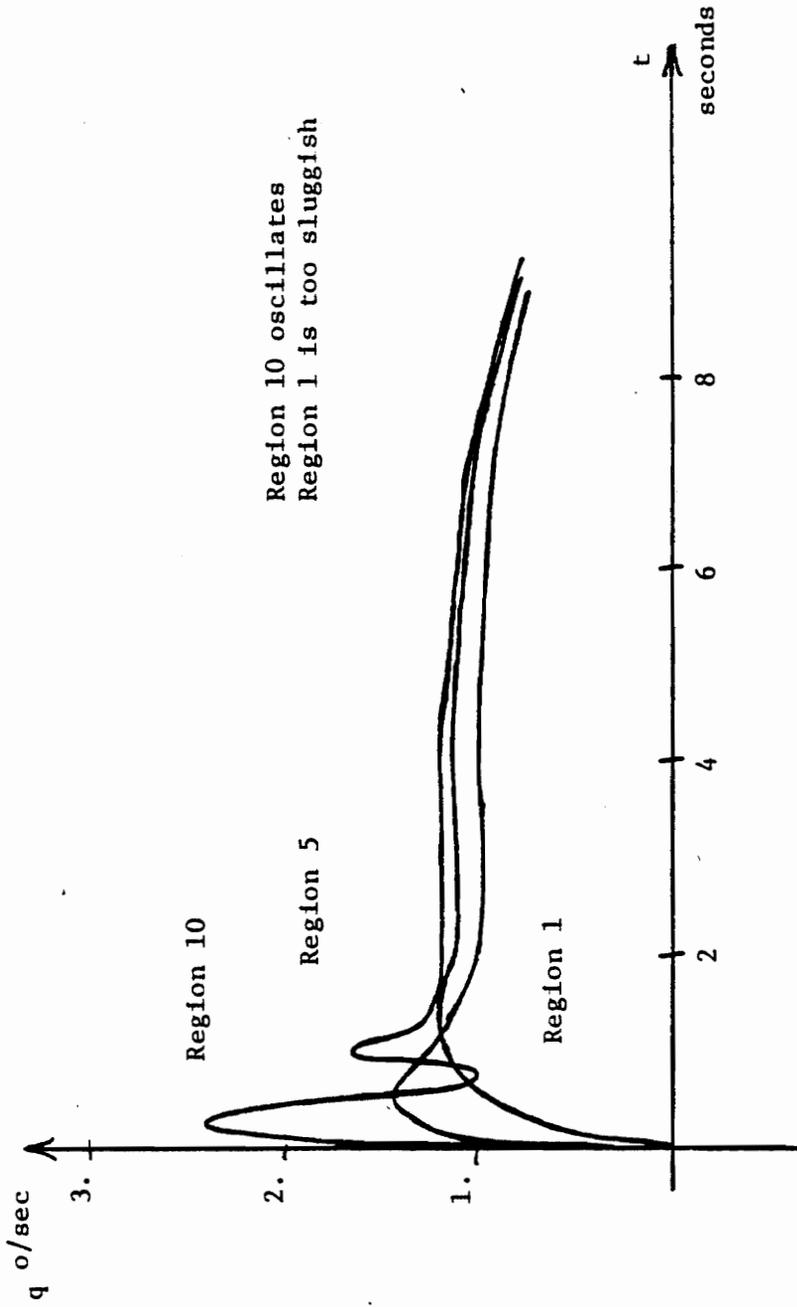


Figure 6.3.5 Closed-Loop Response Using Feedback Gains Derived for Region 5

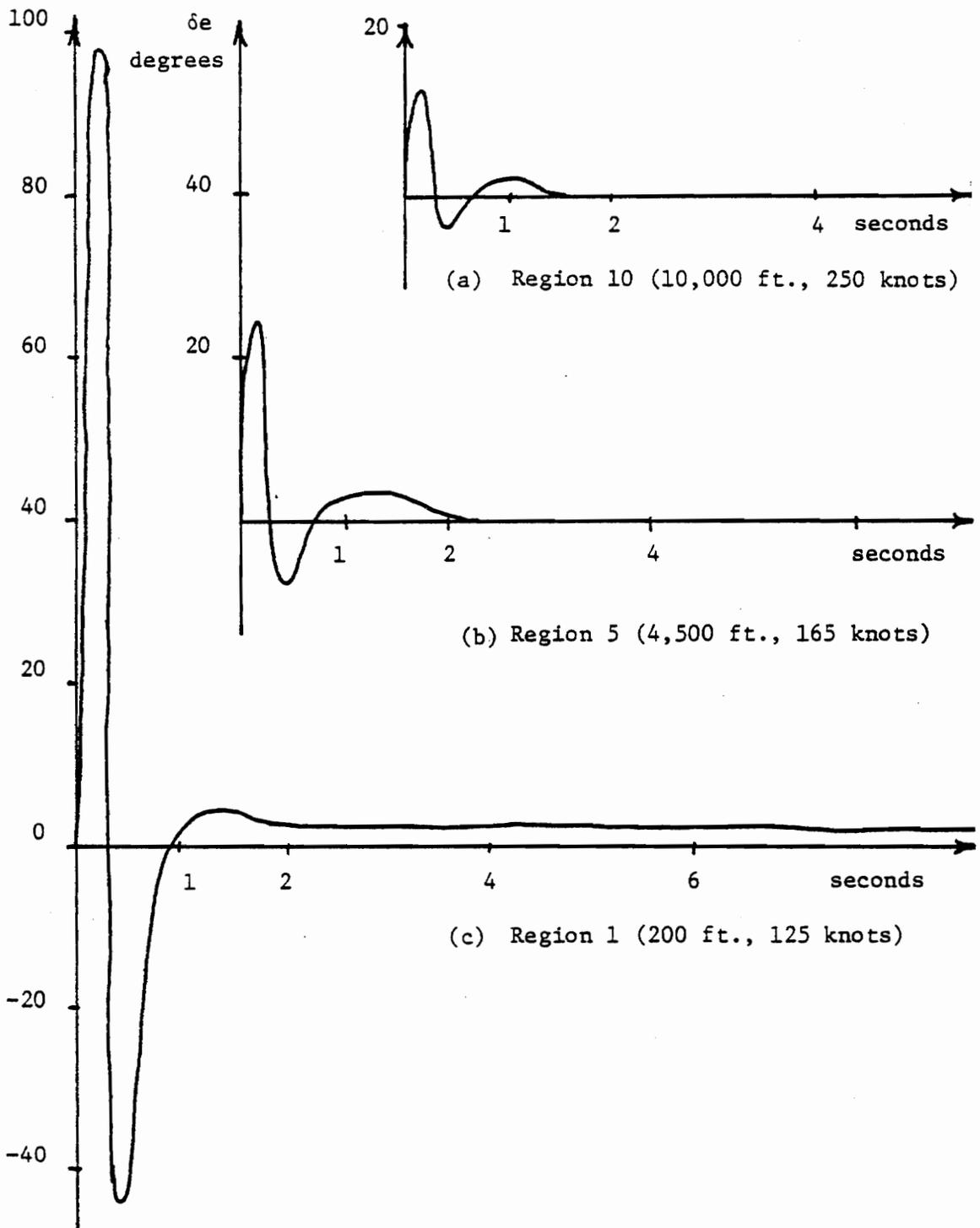


Figure 6.3.6 Closed loop elevator response required to capture a 20 ft/sec sink rate glide-slope in the different regions.

gains and input gains that will cause a given system to have a response identical to a given model. The requirements that a system and model must satisfy in order that the problem be solvable are rigorously formulated and solved in reference [M-9]. Model following has found wide usage in the area of aircraft simulation. Specifically, it has been useful as a method of altering the response of an aircraft to emulate the response of a second aircraft [T-2][A-3]. The application of model following to the problem of trajectory following as done in this chapter further illustrates the utility of this method of design.

An excellent summary of the model-following method is given in reference [A-3] and is repeated in Appendix C. A block diagram of a closed-loop model-following system is shown in Figure 6.4.1.

The purpose of the model in this application is to produce the glideslope that the B737 aircraft will be required to follow. The glideslope is defined by the initial altitude, initial distance, initial velocity and landing velocity of the aircraft. It is desired to have a linear decrease in altitude with time and a linear decrease in velocity from the initial velocity to the final landing velocity. From the initial conditions then,

$$U_{\text{avg}} = (U_o + U_f)/2 \quad (6.4.1)$$

$$T = X_o/U_{\text{avg}}$$

$$A_x = (U_f - U_o)/T$$

$$U_z = Z_o/T$$

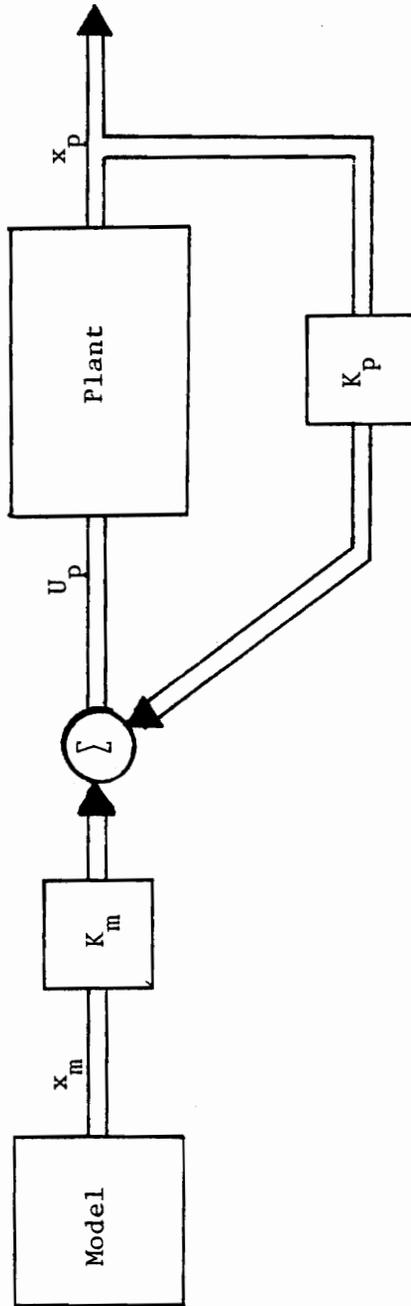


Figure 6.4.1 Block Diagram of Model-Following System

where,

$U_0$  = initial velocity, x-direction

$U_f$  = final velocity, x-direction

$U_{avg}$  = average velocity

$U_z$  = sink velocity, z-direction

$X_0$  = initial distance from landing point

$A_x$  = x-acceleration

$T$  = expected time interval till landing.

A state model producing the desired glideslope is,

$$\dot{x}_m = A_m x_m \quad x_m(0) = x_{m0} \quad (6.4.2)$$

where,

$$x_m^T = [x \quad \dot{x} \quad \ddot{x} \quad z \quad \dot{z} \quad s]$$

$$A_m = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$x_{m0} = \begin{bmatrix} 0 \\ 0 \\ A_x/U_0 \\ 0 \\ 0 \\ V_z/U_0 \end{bmatrix}$$

The variables have been normalized to conform to the dimensionless state variables, in the state equations for the B737 aircraft, i.e. equation (6.2.8). The stabilizer position, S, has been included as a state having a constant value of zero. This allows nonzero stabilizer perturbations to be penalized in the subsequent solution of the model-following equations (see Appendix C). The state variables of the aircraft to be matched to the model are the x-position, the sink position and the stabilizer perturbation. Referring to Appendix C, this requires

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

$$H_m = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (6.4.3)$$

Errors in these three state variables are weighted relative to control effort by the adjustment of the Q and R matrices in the cost functional. By trial and error the following matrices were found to give good glide-slope following with acceptable control effort:

$$Q = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix} \quad R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6.4.4)$$

The combined feedback/input gain matrices for the various flight configurations were found using a model-following computer program given in

reference [A-3]. For example, the gain matrix corresponding to configuration 1 is given in Figure 6.4.2.

### 6.5 Nonlinear Measurement Model

The modified PAC algorithm requires a linear measurement matrix for each of the Kalman filters. However, several of the measurements that are available to the controller are nonlinear functions of the state variables. In this section a prefilter is developed to eliminate these nonlinearities. (See also references [H-4], [J-4].)

The measurements are available from two sources, on-board sensors and MLS (Microwave Landing System) data. The measurements considered and the associated sources of the measurements are listed below.

Z(1) = pitch angle (gyro)

Z(2) = pitch rate (gyro)

Z(3) = slant range (MLS)

Z(4) = elevation angle (MLS)

Z(5) = altitude (barometric)

Z(6) = altitude rise rate (barometric)

Z(7) = acceleration along  $z_5$  (body-mounted accelerometer)

Z(8) = true airspeed (air data computer)

Z(9) = acceleration along  $x_5$  (body-mounted accelerometer) (6.5.1)

The relationships between the measurements and the state variables are:

$$Z(1) = \theta_o + K_d x(4) + v(1)$$

$$Z(2) = K_d x(3) + v(2)$$

$$Z(3) = [(Z_o - U_o x(5))^2 + (X_o - U_o x(6) - U_o t)^2]^{\frac{1}{2}} + v(3)$$

$$F^T = \begin{bmatrix} -.28 & 4.69 & .108 \\ 3.24 & -.144 & -.931 \\ -1.36 & -.015 & .453 \\ -5.18 & .0366 & 1.49 \\ 2.43 & -.369 & -.595 \\ .369 & 2.31 & -.0938 \\ -.0958 & 7.00 & .0604 \\ .011 & 2.55 & .0043 \\ -.755 & -.002 & 2.59 \\ \hline -.377 & -2.35 & .0951 \\ -.776 & -4.92 & .232 \\ -.741 & -5.11 & .273 \\ -2.43 & .368 & .597 \\ -5.18 & .797 & 1.50 \\ -.932 & .00075 & 1.80 \end{bmatrix} = \begin{bmatrix} F_x \\ F_{x_m} \end{bmatrix}$$

Figure 6.4.2 Gain Matrix for Configuration  $S_1$

$$Z(4) = (K_d) \tan^{-1} [(Z_o - U_o x(5)) / (X_o - U_o x(6) - U_o t)] + v(4)$$

$$Z(5) = Z_o - U_o x(5) + v(5)$$

$$Z(6) = -[U_o \dot{x}(5) + \dot{Z}_o] v(6)$$

$$Z(7) = U_o [\dot{x}(1) \tan x(2) + \frac{(1+x(1)) \cdot}{\cos^2 x(2)} \dot{x}(2) - x(3) (1+x(1) + \cos x(4) - \cos x(2))] + v(7)$$

$$Z(8) = \frac{U_o (1+x(1))}{\cos x(2)} v(8)$$

$$Z(9) = U_o [\dot{x}(1) + (1+x(1)) x(3) \tan x(2) + \dot{x}(2) \sin x(2) - x(3) \sin x(4)] + v(9)$$

where,

$$x_a^T = [u \quad \alpha \quad q \quad \theta \quad z \quad x \quad T \quad th \quad S]^T$$

$$K_d = 57.29578 \text{ deg/rad}$$

$$\dot{Z}_o = -U_o \sin \gamma_o$$

and  $v^T = [v(1) \quad v(2) \quad \dots \quad v(9)]^T$  are noise quantities. (6.5.2)

The standard deviations of the noise elements are given in Table 6.5.1. It is seen that  $v(6)$  and  $v(8)$  are multiplicative noise factors. These noise factors can be approximated by additive noise terms as shown below. This step is required in the implementation of the linear Kalman filter equations.

$$v(8) = (1 + \epsilon_8) \quad \epsilon_8 = .02$$

$$v'(8) = \epsilon_8 U_{avg}$$

$$v(6) = (1 + \epsilon_6) \quad \epsilon_6 = .05$$

$$v'(6) = \epsilon_6 U_Z \quad (6.5.3)$$

Table 6.5.1 Actual Measurement Noise Characteristics

Noise Variable	Standard Deviation	Covariance	Type
V(1)	.15°	.0225	additive
V(2)	.10°/sec	.01	additive
V(3)	1 ft	1.0	additive
V(4)	.031°	.001	additive
V(5)	25 ft	625	additive
V(6)	5%	.0025	multiplicative
V(7)	.16 ft/sec <sup>2</sup>	.0256	additive
V(8)	2%	.0004	multiplicative
V(9)	.16 ft/sec <sup>2</sup>	.0256	additive

where  $U_{avg}$  and  $U_z$  are given by equation 6.4.1.

After making several small angle approximations and substituting equation 6.5.3 into 6.5.2, the measurement equations become:

$$Z(1) \text{ through } Z(5) = \text{same as equations (6.5.2)}$$

$$Z(6) \approx -[U_o \dot{x}(5) + \dot{Z}_o] + v'(6)$$

$$Z(7) \approx U_o [\dot{x}(2) - x(3)] + v(7)$$

$$Z(8) \approx U_o (1 + x(1)) + v'(8)$$

$$Z(9) \approx U_o \dot{x}(1) + v(9) \quad (6.5.4)$$

A prefilter is now constructed to give a second set of measurements that are linear combinations of the states. Although approximate, these equations give very good results for small values of noise [J-4][H-4].

$$z(1) = [Z(1) - \theta_o] / kd = x(4) + v(1)$$

$$z(2) = Z(2) / kd = x(3) + v(2)$$

$$z(3) = [X_o - U_o t - Z(3) \cos(Z(4) / kd)] / U_o = x(6) + v'(3)$$

$$z(4) = [Z_o - Z(3) \sin(Z(4) / kd)] / U_o = x(5) + v'(4) \quad (6.5.5)$$

$$z(5) = [Z_o - Z(5)] / U_o = x(5) + \frac{v(5)}{U_o}$$

$$z(6) = [U_o \sin(\theta_o - \alpha_o) - Z(6)] / U_o = \dot{x}(5) + \frac{v'(6)}{U_o}$$

$$z(7) = Z(7) / U_o - \epsilon_2^T BU = \dot{x}(2) - x(3) + \frac{v(7)}{U_o}$$

$$z(8) = Z(8) / U_o - 1 = x(1) + \frac{v'(8)}{U_o}$$

$$z(9) = Z(9)/U_o = x(1) + \frac{v(9)}{U_o} \quad (6.5.5)$$

The prefiltered measurements are thus linearly related to the state variables by the measurement coefficient matrix, H, as shown in Figure 6.5.1.

The noise terms  $v'(3)$  and  $v'(4)$  represent equivalent errors in the x-distance and altitude variables, respectively, and are derivable from the slant range and elevation angle errors. For example, let the slant range plus error be  $R + dR$ , and the elevation angle plus error be  $\xi + d\xi$ . Then,

$$\begin{aligned} Z + dz &= (R + dR)\sin(\xi + d\xi) \\ &= (R + dR)[\sin\xi\cos d\xi + \cos\xi\sin d\xi] \\ &\approx (R + dR)[\sin\xi + d\xi\cos\xi] \\ &\approx R\sin\xi + dR\sin\xi + (R\cos\xi)d\xi \end{aligned}$$

Thus,

$$v'(4) = dz \approx (\sin\xi)dR + (R\cos\xi)d\xi.$$

Similarly,

$$v'(3) = dx \approx (\cos\xi)dR - (R\sin\xi)d\xi \quad (6.5.6)$$

Using nominal values for  $R$  and  $\xi$ , and the standard deviations of  $d\xi$  and  $dR$  given in Table 6.5.1, the standard deviations  $v'(3)$  and  $v'(4)$  can be found.

The actual measurement noise was simulated using the covariances given in Table 6.5.1. The covariance matrix of the prefiltered noise terms given in equations 6.5.5 and 6.5.6 is shown in Figure 6.5.2. This matrix represents the prefiltered noise covariances and was used in the

$$H = \begin{bmatrix} 0 & 0 & 0 & 1.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & -1.0 & 0 & 0 & 0 & 0 & 0 \\ - .31 & - .756 & 0 & 0 & 0 & 0 & - .042 & 0 & .097 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ - .047 & .082 & 0 & - .152 & 0 & 0 & 1.9 & 0 & 0 \end{bmatrix}$$

Figure 6.5.1 Prefiltered Measurement Coefficient Matrix, H



solution of the Kalman filter equations.

## 6.6 B737 Simulation and Results

The simulation of the B737 is accomplished by initializing the aircraft at some altitude, x-position and x-velocity, and then controlling it to follow a glideslope corresponding to a linearly decreasing altitude and a linearly decreasing velocity. A model system that produces this glideslope is given in equation (6.4.2). As the aircraft altitude decreases the appropriate linearized model is switched in according to the schedule of configurations given in Figure 6.2.2. In this manner a very good approximation to the actual nonlinear aircraft response can be achieved.

The times of the configuration changes, as well as the configurations themselves, are, of course, unknown to the controller. The controller will consist of three filters matched to configurations  $s_1$ ,  $s_5$  and  $s_{10}$ , respectively. The feedback and input gains required for each of these configurations to follow the glideslope are precalculated using the model-following method discussed in Section 6.4. The modified PAC of Chapter Four is then employed to adaptively learn the most appropriate aircraft configurations (of  $s_1$ ,  $s_5$  and  $s_{10}$ ) and to apply the corresponding control gains.

For example, Figure 6.6.1 gives the initial conditions, wind disturbance covariance, initial state uncertainty and semi-Markov matrix for a 6-degree glideslope. The wind disturbance is assumed to be a zero-mean, white gaussian process. Correlated wind can be accommodated by increasing the dimensions of the system to include the necessary extra

## Glideslope Parameters

$$U_o = 400 \text{ ft/sec}$$

$$U_f = 200 \text{ ft/sec}$$

$$X_o = 50,000 \text{ ft}$$

$$Z_o = 10,200 \text{ ft}$$

$$P(0) = \text{diagonal} \begin{Bmatrix} 4.0 \times 10^{-4} & 2.5 \times 10^{-3} & 3 \times 10^{-5} & 5 \times 10^{-5} \\ 5 \times 10^{-3} & 6 \times 10^{-2} & 3 \times 10^{-5} & 3 \times 10^{-5} & 1 \times 10^{-3} \end{Bmatrix}$$

$$Q = \text{diagonal} \{1 \times 10^{-5} \quad 1 \times 10^{-5} \quad 1 \times 10^{-5}\}$$

$$SM = \begin{bmatrix} .99 & .005 & .005 \\ .005 & .99 & .005 \\ .005 & .005 & .99 \end{bmatrix}$$

Figure 6.6.1 Glideslope Simulation Parameters

state variables. However, this was not done in this example.

The aircraft simulation and modified PAC algorithm were carried out on a high speed digital computer. Figures 6.6.2 and 6.6.3 show the accuracy of the resulting glideslope. The weighting coefficients are shown in Figure 6.6.4. It is seen that they accurately follow the changing aircraft dynamics.

### 6.7 Summary

A major goal of the NASA Terminal configured vehicle (TCV) project is the automatic control of aircraft in the vicinity of the terminal during adverse weather conditions. During the landing approach, wind shear and gusts can excite the various dynamic modes of the aircraft, such as the Dutch roll and the phogoid oscillation, which, if not effectively compensated for, could lead to a less than optimum, if not disastrous, flight termination. Several factors combine to make effective compensation a difficult problem. The dynamics of the aircraft are nonlinear, time-varying and are only approximately known. In addition, many aerodynamic coefficients are given in tabular form, and some are only estimated. Measurement errors on the output variables, state variable estimation errors and random input modeling errors further complicate the problem.

The modified PAC algorithm has been shown to be successful in controlling the longitudinal dynamics of the B737 aircraft. It was demonstrated that the changes in aircraft dynamics due to altitude and velocity variations, flap extension, landing gear position, etc., could be identified, thus allowing the proper feedback to be applied to give

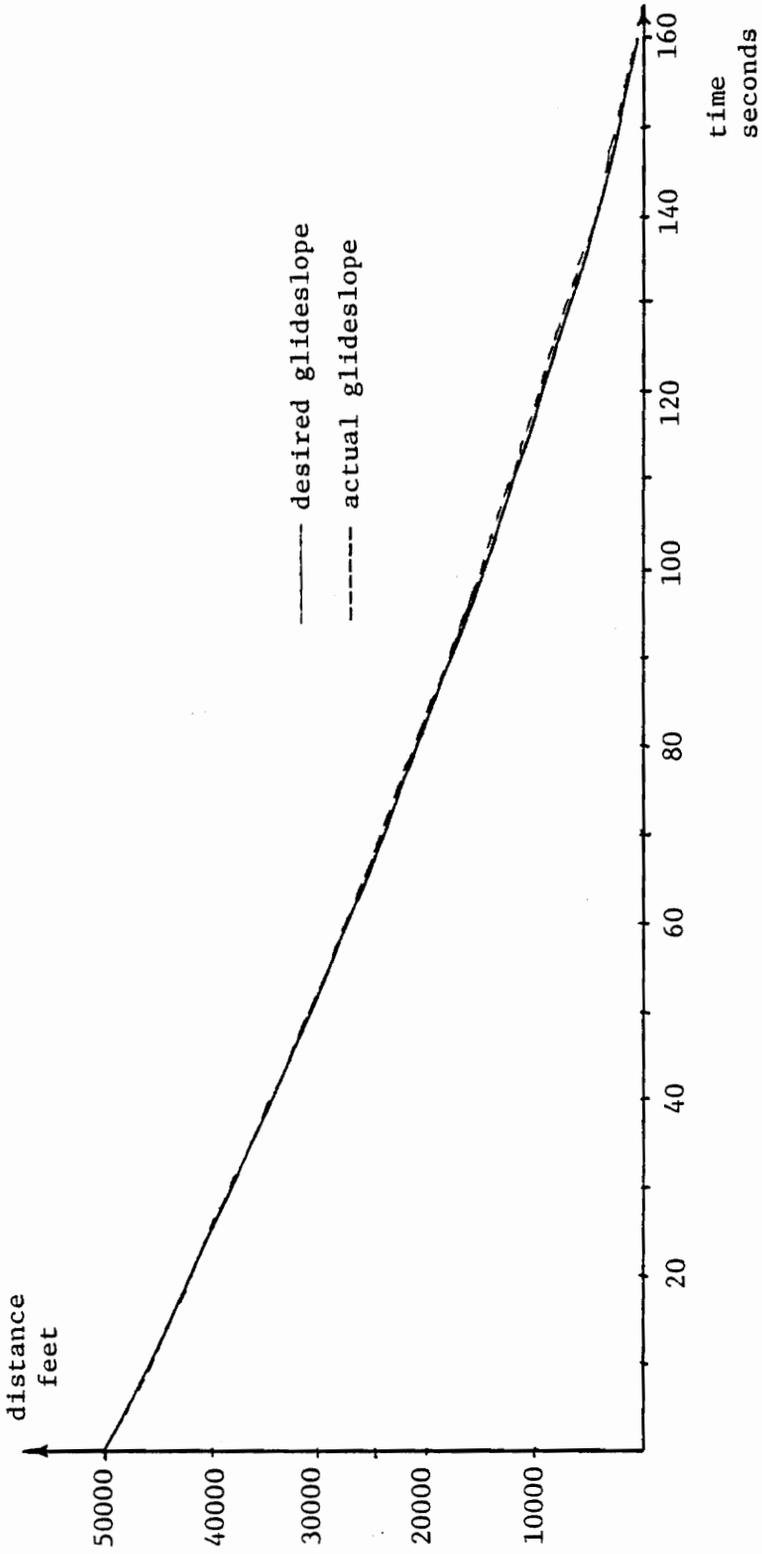


Figure 6.6.2 Actual Versus Desired Glideslope, x-position

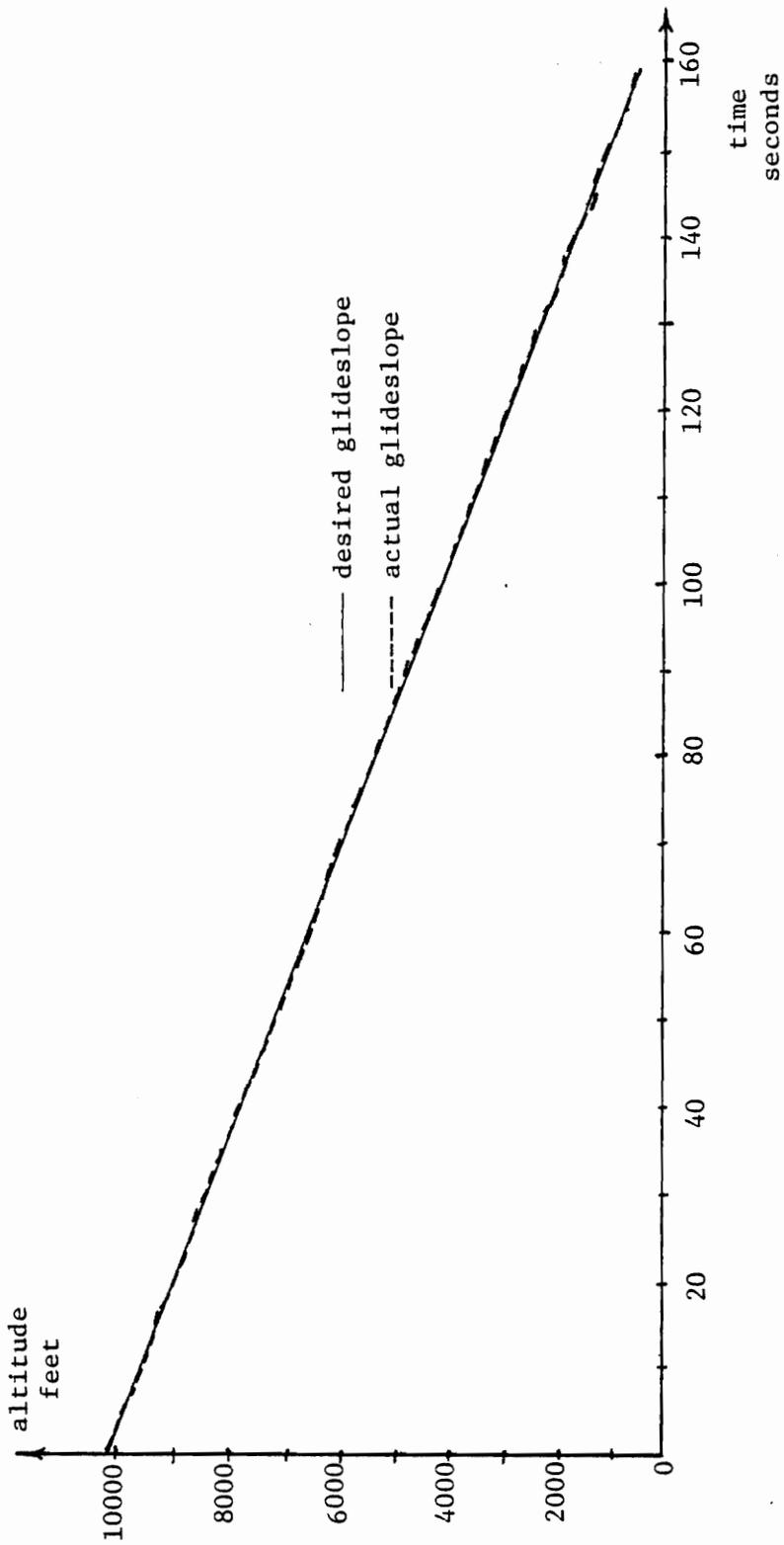


Figure 6.6.3 Actual Versus Desired Glideslope, z-position

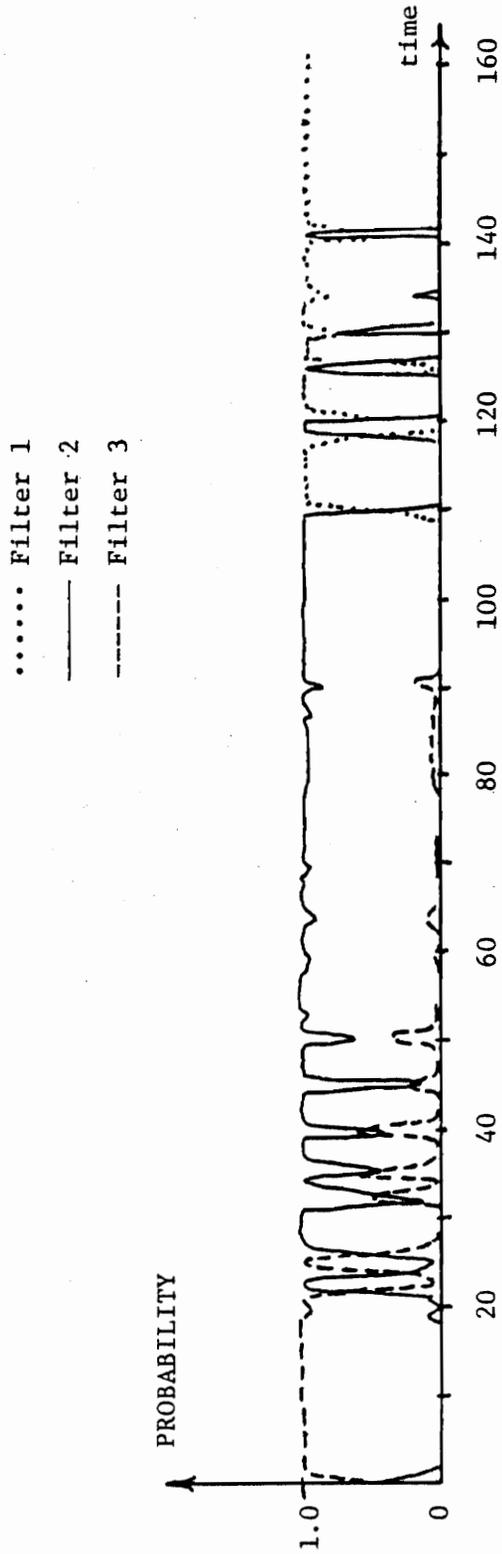


Figure 6.6.4 Weighting Coefficients for Glideslope Following

overall uniform performance. This was shown by controlling the aircraft down a pre-determined glideslope. This required compensating for the phugoid oscillation, as well as the short period mode oscillation, by calculating feedback and input gains using a model-following approach. Again, the ability to determine the changing dynamics allowed for uniform control all along the glideslope.

The set of possible configurations can be arrived at in several different ways. If a reasonably accurate mathematical model exists for the nonlinear system (as in the case for the B737 aircraft), the equations can be linearized at a sufficient number of points in the state space  $[Z-1][M-8]$ , and these linear approximations will then comprise the configuration set. On the other hand, if no mathematical model exists, one needs (only) to experimentally determine a set of reduced order linear systems that satisfactorily describe the nonlinear plant. The order of these approximations will be determined by the accuracy required, but, even though no approximation will be exactly matched to the plant, the adaptive controller will find the best match.

Several implementation advantages exist because the controller uses standard Kalman filters. First, the Kalman filter is reliable and widely used, as contrasted with the existing nonlinear filtering techniques. Second, the filter gains and covariances can be precomputed and stored off-line, for considerable savings in on-line computational time.

Finally, the structure of the controller makes it amendable to a parallel processing implementation. This has advantages in on-line execution time, as well as a potential for redundancy.

## 7.0 CONCLUSIONS

The primary contributions of this dissertation are, first, the extension of an adaptive identification algorithm to nonlinear estimation and control problems, and, second, the development of a nearly optimum estimator for the switched linear plant problem. The stochastic nonlinear controller developed in Chapters Three and Four eliminates a large portion of the computational burden and performance difficulties associated with current methods. In addition, it has several implementation advantages which are summarized in the following paragraphs.

The adaptive identification algorithm introduced by Magill [M-1], refined by Lainiotis et al [L-2], and referred to as the Partitioned Adaptive Estimator (PAE), has found wide usage over the last decade. However, the application of the algorithm to plants having frequent configuration changes has achieved only limited success. Brown and Price attributed the inability of the PAE algorithm to track a maneuvering target, to the weighting coefficients becoming zero [B-1]. As shown in Chapter Three, this is only partly correct. Moose eliminated the problem of the weights going to zero by incorporating a semi-Markovian plant model into the estimator structure [M-2]. The net result is an identification algorithm that is sensitized to the possibility that configuration changes can occur. At this point, the PAE algorithm with the semi-Markov addition is capable of tracking slowly changing plants. However, as shown in Chapter Three, another problem with the PAE algorithm is that the Kalman gains become asymptotically small, which results in a rather long lag time between when a configuration change

occurs and when the newly matched filter begins tracking the plant. The PAE algorithm was modified in Chapter Three to significantly reduce this lag time by re-initializing the Kalman gains when a configuration change occurs. The addition of the re-initialization mechanism enables the PAE to track rapidly changing plant configurations.

The nonlinear estimation problem was "solved" using the above modified PAE algorithm. First, the nonlinear equations are partitioned into a finite set of linear approximations, such that the operation of the nonlinear plant is given, approximately, by a time sequence of the linear models. The modified PAE algorithm then regards the nonlinear plant to be simply a switched linear plant.

The PAE algorithm has several advantages over existing nonlinear estimation methods. First, the bank of filters can be implemented using a parallel processing structure. This structure readily accepts system redundancy and system failure models in the form of additional elemental estimators. Second, the elemental estimators, being linear, allow a large portion of the filter computations to be done off-line. Third, the modified PAE algorithm is quite flexible in regards to the nature of the nonlinear plant. The linear approximations can be made just as readily from tabular data, discontinuous nonlinearities and even underspecified functions. Other methods, such as the EKF, normally require an accurate nonlinear description.

The main disadvantage of the modified PAE algorithm is its inaccuracy. It is not meant to compete in this respect with, say, the EKF. Accuracy can be improved by increasing the density of linear approximations. However, this will also increase the learning time and

thus decrease the responsiveness of the estimator. A second disadvantage is the required tuning of the re-initialization mechanism. As pointed out in Chapter Three, this must be done empirically for each problem.

The modified PAE algorithm was extended to the stochastic nonlinear control problem in Chapter Four. A major advantage of this method, referred to as the modified PAC algorithm, is that the well developed methods of linear systems theory can be applied to calculate the feedback gains. Either a classical eigenvalue specification or an optimal control formulation can be used for this purpose. The set-point control example in Chapter Four and the B737 slideslope tracking application of Chapter Six show the utility of the modified PAC algorithm.

The problem of detecting configuration changes led to the development of the Sliding Window Detector/Estimator, SWDE, algorithm. The performance of this technique approaches the accuracy of the unrealizable optimum estimator for the switched linear plant problem. Previous attempts in this direction have all led to sub-optimal filters that are unnecessarily approximate. (For example see [A-2].) The SWDE algorithm utilizes steady state gains and a finite measurement window to minimize the computational burden while maximizing tracking performance. The three examples in Chapter Five illustrate the capabilities of this algorithm.

A disadvantage of the SWDE algorithm is the possibility of its losing track of the plant state. When this occurs, the algorithm must be re-initialized. It was shown in Chapter Five that the modified PAE algorithm together with the SWDE algorithm results in an estimator that is both accurate and reliable. It is felt that this combined structure

significantly extends the state of the art in adaptive estimation and merits further application in the tracking of maneuverable targets.

## BIBLIOGRAPHY

- [A-1] Athans, M., et al., "The stochastic control of the F-8c aircraft using the multiple model adaptive control (MMAC) method", pp. 217-228, 1975 JACC Proceedings.
- [A-2] Ackerson, G. A. and Fu, K. S., "On state estimation in switching environments", IEEE T.AC., Vol. AC-15, February 1970.
- [A-3] Armstrong, E. S., "ORACLS - A system for linear-quadratic-gaussian control law design", NASA TP-1106, L-11769, April 1978.
- [B-1] Brown, C. M., Jr., and Price, C. F., "A comparison of adaptive tracking filters for targets of variable maneuverability", pp. 554-563, 1976 JACC proceedings.
- [B-2] Brogan, W. L., Modern Control Theory, Quantum Publishers, Inc., N.Y., 1974.
- [C-1] Caglayan, A. K., "Optimal Parameter Adaptive Estimation of Stochastic Processes," Ph.D. Dissertation, Dept. of Electrical Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA., 1974.
- [C-2] Chien, T., Adams, M. B., "A sequential failure detection technique and its application", IEEE T.AC., pp. 750-757, Oct. 1976.
- [C-3] Chen, C. T., Introduction to Linear System Theory. Holt, Rinehart and Winston, Inc., N.Y., 1970.
- [D-1] Deshpande, J. G., et al., "Adaptive control of linear stochastic systems", Automatica, Vol. 9, 1973, p. 107.
- [D-2] Davison, E. J., "The robust decentralized control of a general servomechanism problem", IEEE T.AC., Vol. AC-21, No. 1, Feb. 1976.
- [E-1] Etkin, B., Dynamics of Atmospheric Flight. John Wiley and Sons, Inc., N.Y., 1972.
- [F-1] Fujishige, S. and Sawaragi, Y., "Optimal estimation for continuous systems with jump-processes", IEEE T.AC., pp. 225-228, June 1974.
- [G-1] Gibson, J. E., Nonlinear Automatic Control, McGraw-Hill Book Co., Inc., 1963.
- [G-2] Gholson, N. H., "Tracking maneuvering targets via semi-Markov maneuver modeling", Ph.D. Dissertation, Dept. of Electrical Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA., 1977.

- [G-3] Gelb, A., et al., Applied Optimal Estimation, MIT Press, Cambridge, MA., 1974, Chap. 6.
- [H-1] Hilborn, C. G. and Lainiotis, D. G., "Optimal adaptive filter realization for sampled stochastic processes with an unknown parameter", IEEE T.AC., Vol. AC-14, Dec. 1969.
- [H-2] Howard, R. A., "System analysis of semi-Markov processes", IEEE Trans. ME, Vol. 8, April 1964.
- [H-3] Hawkes, R. M. and Moore, J. R., "Performance of Bayesian parameter estimators for linear signal models", IEEE T.AC., Vol. AC-21, June 1978.
- [H-4] Halyo, Nesim, "Development of an optimal automatic control law and filter algorithm for steep glideslope capture and glideslope tracking", NASA CR-2720, August 1976.
- [J-1] Jazwinski, A. H., "Adaptive filtering", Automatica, Vol. 5, 1969, pp. 475-485.
- [J-2] Jazwinski, A. H., Stochastic Processes and Filtering Theory. New York: Academic Press, 1970.
- [J-3] Johnson, C. R., "Adaptive single-stage control of a broad class of nonlinear multivariable plants", Proc. 10th Southeastern Symposium on System Theory, Mississippi State, pp. VI A-1, 9, March 1978.
- [J-4] Joglekar, Anil N., "A technique to improve the performance of a nonlinear filter with an application to satellite-aided aircraft navigation", Proc. IEEE CDC, 1974, Phoenix, Arizona, pp. 159-164.
- [K-1] Kolmogorov, A., "Interpolation and Extrapolation ven Stationären Zufälligen Folgen", Bulletin de L'Academie des sciences de URSS, ser. Math 5, 3-14, 1941.
- [K-2] Kalman, R. E. and Bucy, R. S., "New results in linear filtering and prediction theory", Trans. ASME, J. Basic Eng., ser. D., Vol. 83, pp. 95-108, March 1961.
- [K-3] Kailath, T., "An innovations approach to least-squares estimation, Part I", IEEE T.AC., AC-13, pp. 646-655, 1968.
- [L-1] Lainiotis, D. G., "Optimal adaptive estimation: structure and parameter adaptation", IEEE T.AC., Vol. AC-16, No. 2, April 1971.
- [L-2] Lainiotis, D. G., "Partitioning: A unifying framework for adaptive systems, I and II", Proc. of the IEEE, Vol. 64, No. 8, August 1976.

- [L-3] Leones, C. T. and Pearson, J. O., "Bias estimation errors induced by parameter uncertainties", IEEE T.AC., October 1973, pp. 505-509.
- [M-1] Magill, D. T., "Optimal adaptive estimation of sampled stochastic processes", IEEE T.AC., Vol. AC-10, pp. 434-439, October 1965.
- [M-2] Moose, R. L. and Wang, P. O., "An adaptive estimation with learning for a plant containing semi-Markov switching parameters", IEEE Trans. SMC, Vol. 3, May 1973.
- [M-3] McAulay, R. J. and Denlinger, E., "A decision directed adaptive tracker", IEEE T.AES, Vol. AES-9, No. 2, March 1973.
- [M-4] Meditch, J. S., Stochastic Optimal Linear Estimation and Control, McGraw-Hill Book Co., New York, 1969.
- [M-5] Moose, R. L., "An adaptive estimator with learning for a plant containing semi-Markovian switching parameters", Ph.D. Dissertation, Dept. of Electrical Engineering, Duke University, 1971.
- [M-6] Melsa, J. L., Computer Programs for Computational Assistance in the Study of Linear Control Theory. McGraw-Hill Co., N.Y., 1970.
- [M-7] McCabe, D. H., "A nonlinear digital stochastic controller for a plant containing semi-Markov switching parameters", Master's Thesis, Dept. of Electrical Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA., 1975.
- [M-8] Moose, R. L., VanLandingham, H. F. and Zwicke, P. E., "Digital set point control of nonlinear stochastic systems", IEEE T.IECI, Vol. IECI-25, No. 1, February 1978.
- [M-9] Morse, Stephen A., "Structure and design of linear model-following systems", IEEE T.AC., Vol. AC-18, No. 4, August 1973, pp. 346-354.
- [M-10] Montgomery, R. C., "Analytic design of digital flight controllers to realize aircraft flying quality specifications", AIAA Journal of Aircraft, Vol. 9, July 1972, pp. 456-460.
- [N-1] Nahi, N. E. and Schaeffer, B. M., "Decision-directed adaptive recursive estimators: Divergence prevention", IEEE T.AC., Vol. AC-17, February 1972.
- [N-2] Nahi, N. E. and Knobbe, E. J., "Optimal linear recursive estimation with uncertain system parameters", IEEE T.AC., pp. 263-266, April 1976.
- [O-1] Oaks, O. J. and Cook, G., "Piecewise linear control of nonlinear systems", IEEE Trans. IECI, Vol. 23, No. 1, pp. 56-63, February 1976.

- [P-1] Pierce, B. D. and Sworder, D. D., "Bayes and minimax controllers for a linear system with stochastic jump parameters", IEEE T.AC., Vol. AC-16, No. 4, August 1971.
- [R-1] Roskam, J., Flight Dynamics of Rigid and Elastic Airplanes, Roskam Aviation and Engineering Corp., 519 Boulder, Lawrence, Kansas, 1972.
- [S-1] Sims, F. L., Lainiotis, D. G., "Recursive algorithm for the calculation of the adaptive filter weighting coefficients", IEEE T.AC., pp. 215-217, April 1969.
- [S-2] Singer, R. A., "Estimating optimal tracking filter performance for manned maneuvering targets", IEEE T.AES, AES-6, pp. 473-483, 1970.
- [S-3] Sanyal, P. and Shen, C. N., "Bayes' decision rule for rapid detection and adaptive estimation scheme with space applications", IEEE T.AC., June 1974.
- [S-4] Sawaragi, Y., et al, "State estimation for continuous time systems with interrupted observations", IEEE T.AC., Vol. AC-19, No. 4, August 1974.
- [S-5] Saridis, G. N. and Dao, T. K., "A learning approach to the parameter adaptive self-organizing control problem", Automatica, Vol. 8, 1972, p. 589.
- [S-6] Stein, G. and Saridis, G. N., "A parameter-adaptive control technique", Automatica, Vol. 5, 1969, p. 731.
- [T-1] Tse, E. and Bar-shalom, Y., "An actively adaptive control for linear systems with random parameters via the dual control approach", IEEE T.AC., Vol. AC-18, No. 2, April 1973.
- [T-2] Tyler, J. S., Jr., "The characteristics of model-following systems as synthesized by optical control", IEEE T.AC., Vol. AC-9, No. 4, October 1964, pp. 485-498.
- [V-1] Van Trees, H. L., Detection, Estimation and Modulation Theory, Part I, John Wiley and Sons, Inc., N.Y., 1968.
- [V-2] VanLandingham, H. F. and Moose, R. L., "Digital control of high performance aircraft using estimation techniques", IEEE Trans. AES, Vol. 12, No. 2, March 1977.
- [V-3] VanLandingham, H. F., Moose, R. L. and Zwicke, P. E., "Control of nonlinear stochastic systems using adaptive estimation", Southeastcon Proceedings, Williamsburg, VA., April 1977.

- [W-1] Weiner, N., The Extrapolation, Interpolation and Smoothing of Stationary Time Series, John Wiley and Sons, Inc., N.Y., 1949.
- [W-2] Wittenmark, B., "Stochastic adaptive control methods: A survey", International Journal of Control, Vol. 21, No. 3, 1975, p. 699.
- [W-3] Witsenhausen, H. S., "Separation of estimation and control for discrete time systems", Proc. IEEE, Vol. 59, p. 1557, 1971.
- [W-4] Willsky, A. S. and Jones, H. L., "A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems", IEEE T.AC., February 1976.
- [W-5] Winsor, C. A. and Roy, R. J., "The application of specific optimal control to the design of desensitized model-following control systems", IEEE T.AC., Vol. AC-15, No. 3, June 1970, pp. 326-333.
- [Z-1] Zwicke, P. E., Moose, R. L. and VanLandingham, H. F., "Estimation for nonlinear systems", Proc. 9th Southeastern Symposium on System Theory, UNCC, Charlotte, N.C., March 1977.

## APPENDIX A - SWDE ALGORITHM

### A-1 Flowchart of The Basic Algorithm

The basic SWDE algorithm is illustrated by the flowchart in Figure A.1.1. The four steps of the algorithm are: grow the configuration TREE, propagate the tree one iteration, test for a configuration change and compute the state estimate. SWDE is invoked when each measurement is received, and requires the following input data:

$\phi$	State transition matrices for all $s_i$ .
$\Gamma$	Input coefficient matrices for all $s_i$ .
H	Measurement coefficient matrix.
F	Kalman gain steady state matrices.
ZCOV	Measurement residual covariance inverse.
DET	Determinants of ZCOV.
$U_k$	Input vector at time k.
$z_k$	Measurement vector at time K.
$XF_{k-1}$	State estimate at time k-1.
$XW_{k-1}$	Set of branch estimates at time k-1.
$PRBW_{k-1}$	Set of relative probabilities at k-1.
PROLD	The seed probability value at k-N.
LKF	The last detected configuration.
IGROW	Iteration count since last detection.
NKF	The number of configurations; same as M.
NW	The window width; same as N.

The algorithm for detecting a configuration change is shown by the DETECT flowchart in Figure A.1.2. Basically, the relative probabilities

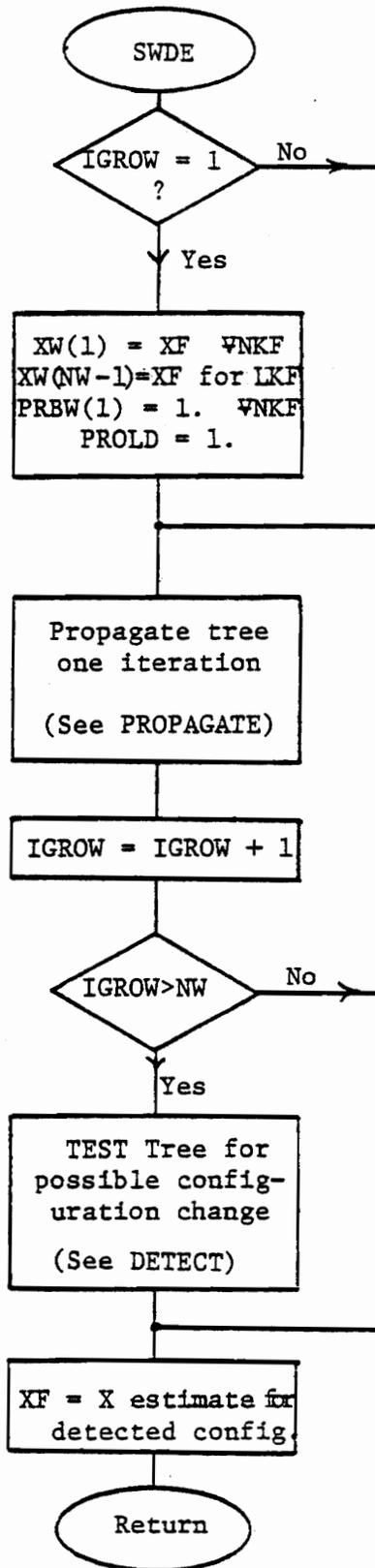


Figure A.1.1 SWDE Algorithm Flowchart

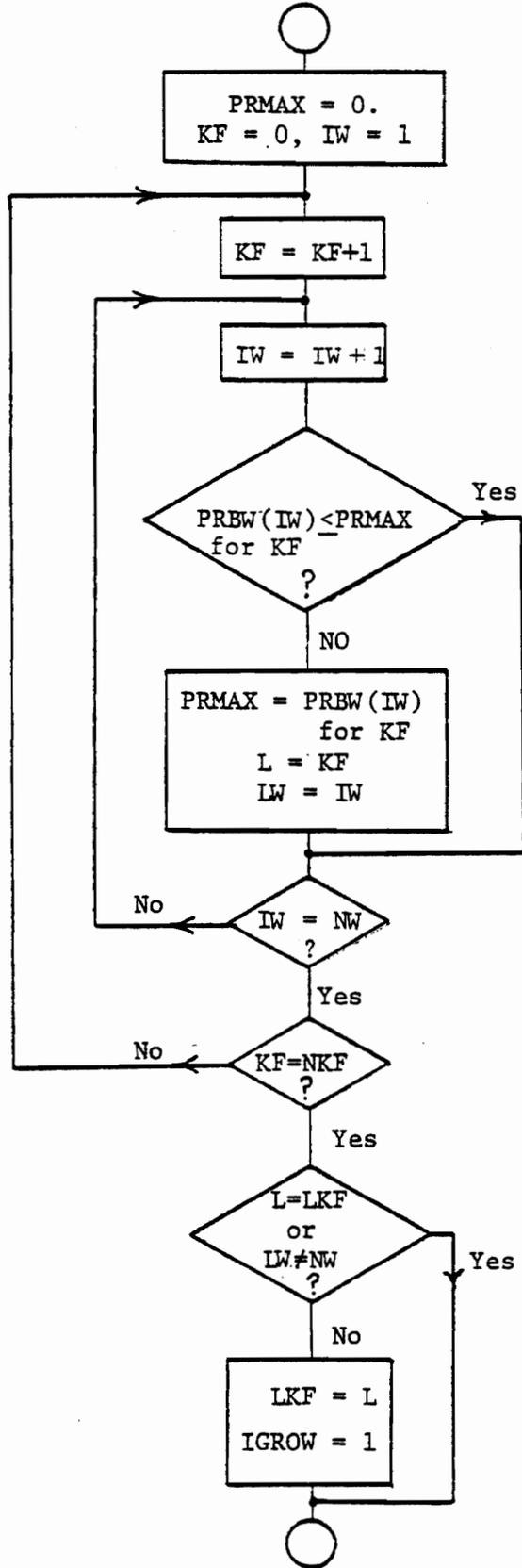


Figure A.1.2 DETECT Flowchart

are examined and, if the maximum probability belongs to a branch corresponding to a configuration change  $NW$  iterations past, then the algorithm detects the change and initiates the growing of a new TREE.

Updating the TREE one iteration is shown in the PROPAGATE flowchart, Figure A.1.3. The state estimates of each branch are updated using the Kalman filter equations, and the relative probabilities are updated by multiplying by the new measurement residual probability value and dividing by the old seed probability.

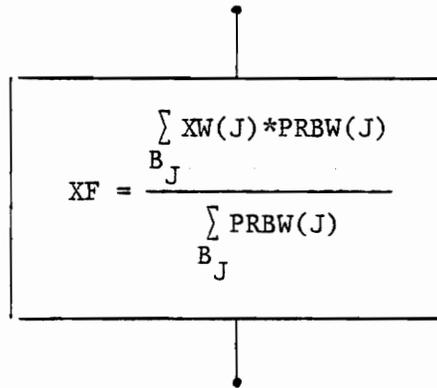
### A-2 Initialization and Re-Start Algorithm

The SWDE algorithm is initialized by executing the modified PAE algorithm, as discussed in Section 5.6. This is illustrated by the TRACK flowchart in Figure A.2.1. With  $IPAE = 0$  initially, the PAE algorithm is executed until the filter gains reach steady state and one of the weighting coefficients,  $\Pi_i$ , exceeds a given threshold. The SWDE algorithm is then continually executed. If divergence ever occurs, the measurement residual will acquire a non-zero value. This is detected by comparing a residual average,  $\bar{z}_{avg}$ , to a threshold value. If  $\bar{z}_{avg}$  exceeds the threshold for three window lengths, the TRACK algorithm is re-started.

### A-3 Weighted-Sum Estimation Modification

As discussed in Section 5.5, estimation performance can often be improved by using a weighted-sum of the branch estimates at every iteration. The modification required to implement this estimate is to replace the final block of the SWDE flowchart in Figure A.1.1 with the

following block.


$$XF = \frac{\sum_{B_J} XW(J) * PRBW(J)}{\sum_{B_J} PRBW(J)}$$

where the summation is over all  $NW(NKR-1) + 1$  branches.

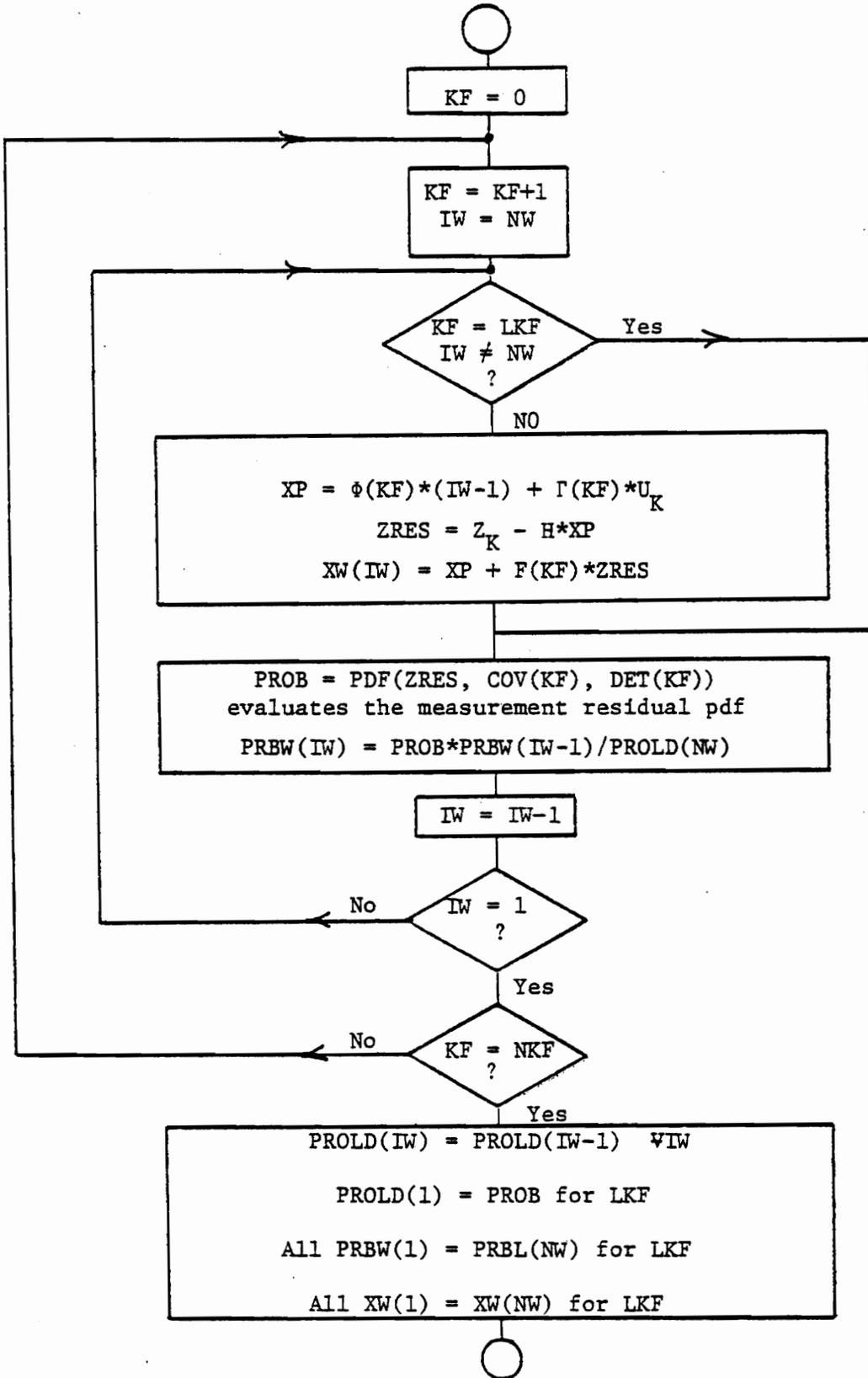


Figure A.1.3 PROPAGATE Flowchart

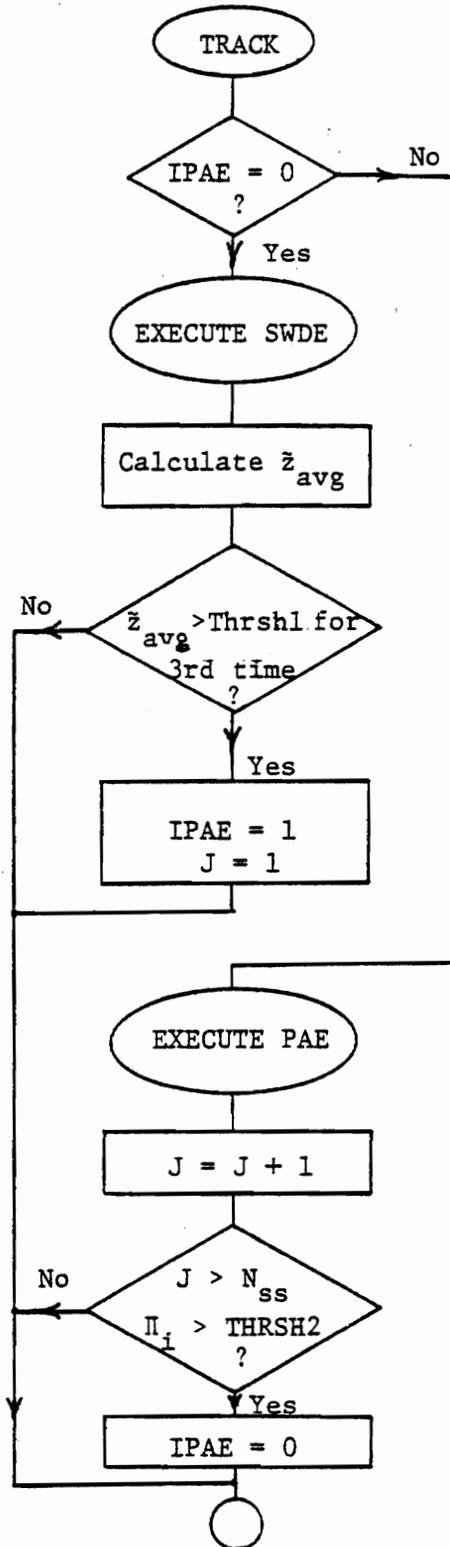


Figure A.2.1 TRACK Algorithm Flowchart

APPENDIX B - THE "TCVA - 737 TRIM DATA VALIDATION" PROGRAM

B-1 Description

By specifying the desired steady state flight conditions, the program will determine whether the aircraft can be trimmed. If successful, the required trim settings will be printed out. In addition, the state variable (perturbation) matrices will be printed in the body, stability and wind axes systems; and the aircraft eigenvalues and eigenvectors for the specified flight condition will be listed.

B-2 Use

<u>Specify</u>	<u>Units</u>
(1) VEAIC (Equivalent air speed)	knots
(2) WEIGHT	lbs.
(3) ALT (Altitude)	feet
(4) GEAR (0 = up; 1 = down)	
(5) FLAPS (0 to 40 degrees)	degrees
(6) C.G. (Center of gravity, .1 to .3)	
(7) GAMMA (Flight path angle)	degrees
(8) PHIDEG (Roll, banking angle)	degrees

If the B737 can be trimmed for the above conditions, the program will print out:

"Aircraft trimmed T"

If the B737 cannot be trimmed, the following output is not valid, and the program will print out:

"Aircraft trimmed F"

B-3 Output

Page 1 and 3: The trim settings:

(1)	ALPDEG	$\alpha$ , angle of attack	degrees
(2)	BETADEG	$\beta$ , angle of sideslip	degrees
(3)	DELS	stabilizer	degrees
(4)	DELE	elevator	degrees
(5)	DELA	aileron	degrees
(6)	DELR	rudder	degrees
(7)	T1, T2	engine thrust	lbs.
(8)	VTB	true air speed	feet/sec.

As a verification, the state derivatives are printed out; they should be very small values.

Page 2 and 3: All the lift and drag coefficients.

Page 4: A, B matrices (Body axis).

Page 5: Eigenvalues and eigenvectors.

Page 6: A, B matrices (stability axis).

Page 7: A, B matrices (wind axis).

B-4 Example

The following parameters were used:

(a)	weight = 70,000	C.G. = .3
(b)	weight = 80,000	C.G. = .25
(c)	weight = 90,000	C.G. = .15

For each of these cases one hundred runs were made for various flight conditions, as shown in Table B.4.1. The ten flight conditions' state equation matrices used in the B737 simulation in Chapter 6 are given in Table B.4.2.

Table B.4.1 Flight Conditions Used in the Trim Data Validation Program

Altitude ALT	Velocity VEAIC	FLAPS	Flight Page Angle GAMMA	Banking Angle PHIDEG	Landing Gear GEAR	Total Number of Runs at this Altitude
200	125 135	40	0 -3 -6	0	1	6
1,500	125 135	40	0 -3 -6	0 15 30	1	18
2,500	145	30 40	0 -3 -6	0 15 30	0	18
3,500	155	30 40	0 -3 -6	0 15 30	0	18
4,500	165	30 40	0 -3 -6	0 15 30	0	18
6,000	175	25 30	0 -3 -6	0	0	6
7,000	185	15 25	0 -3 -6	0	0	6
8,000	200	0 10	0 -3 -6	0	0	6
9,000	225	0	0 -3 -6	0	0	3
10,000	250	0	0 -3 -6	0	0	3

Total runs for given weight and center of gravity - 102

Table B.4.2 The Ten Configurations Used in Chapter 6

A				B			C			REGION
-0.047	.082	0.	-.152	0	0	0	-0.047	.082	0	1
-.310	-.756	1.0	0.	-.042	-.042	.097	-.310	-.756	-.042	
-.142	-1.07	-.529	0.	-1.20	-.19	2.51	-1.42	-1.07	-.434	
0.	0.	1.	0.	0	0	0	0.	0.	0.	
-.037	.083	0.	-.134	0	1.68	0	-.037	.083	0	2
-.274	-1.13	1.0	0.	-.05	.011	.106	-.274	-1.13	-.041	
-.230	-1.30	-.576	0.	-1.47	.053	3.09	-.230	-1.30	-.475	
0.	0.	1.	0.	0	0	0	0.	0.	0.	
-.030	.068	0.	-.127	0	1.58	0	-.030	.068	0.	3
-.259	-.897	1.0	0.	.051	-.021	.109	-.259	-.897	-.04	
-.102	-1.64	-.593	0.	-1.59	-.107	3.38	-.102	-1.64	-.485	
0.	0.	1.	0.	0	0	0	0.	0.	0.	
-.022	.074	0.	-.114	0	1.42	0	-.022	.074	0.	4
-.231	-1.28	1.0	0.	-.056	.006	.118	-.231	-1.28	-.038	
-.27	-2.05	-.643	0.	-1.9	.035	4.09	-.27	-2.05	-.528	
0.	0.	1.	0.	0	0	0	0.	0.	0.	
-.022	.063	0.	-.105	0	1.31	0	-.022	.063	0.	5
-.214	-1.34	1.0	0.	-.058	.022	.125	-.214	-1.34	-.037	
-.303	-2.37	-.678	0.	-2.16	.126	4.68	-.303	-2.37	-.555	
0.	0.	1.	0.	0	0	0	0.	0.	0.	
-.021	.060	0.	-.10	0	1.24	0	-.021	.060	0.	6
-.203	-1.06	1.0	0.	-.058	-.02	.125	-.203	-1.06	-.036	
-.087	-2.48	-.681	0.	-2.26	-.119	4.93	-.087	-2.48	-.551	
0.	0.	1.	0.	0	0	0	0.	0.	0.	
-.018	.055	0.	-.093	0	1.16	0	-.018	.055	0.	7
-.187	-.98	1.	0.	-.06	-.027	.13	-.187	-.98	-.035	
-.068	-1.80	-.710	0.	-2.57	-.167	5.31	-.068	-1.80	-.573	
0.	0.	1.	0.	0	0	0	0.	0.	0.	
-.015	.066	0.	-.085	0	1.06	0	-.015	.066	0.	8
-.169	-.936	1.	0.	-.063	-.035	.139	-.169	-.936	-.033	
-.055	-2.08	-.758	0.	-2.9	-.225	6.45	-.055	-2.08	-.609	
0.	0.	1.	0.	0	0	0	0.	0.	0.	
-.009	.047	0.	-.074	0	0	0	-.009	.047	0.	9
-.147	-.871	1.0	0.	0	.923	0	-.147	-.871	-.032	
.022	-3.0	-.842	0.	-.068	-.059	.154	.022	-3.0	-.671	
0.	0.	1.	0.	-3.6	-.418	8.16	0.	0.	0.	
-.009	.036	0.	-.066	0	0	0	-.009	.036	0.	10
-.130	-.956	1.0	0.	0	.818	0	-.130	-.956	-.031	
.029	-3.66	-.928	0.	-.072	-.038	.169	.029	-3.66	-.727	
0.	0.	1.	0.	-4.3	-.295	10.0	0.	0.	0.	

## APPENDIX C - SUMMARY OF THE MODEL-FOLLOWING PROBLEM

The following description and solution outline of the time-invariant, asymptotic, explicit (model-in-the-system) model-following problem is taken from the documentation of ORACLS [A-3].

### C-1 Continuous Model-Following Problem

For the continuous case, the state and output equations are given as

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t)$$

$$y(t) = \mathbf{H}\mathbf{x}(t)$$

where  $\mathbf{x}(0) = \mathbf{x}_0$  is given, and the constant matrices  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{H}$  are of dimension  $n \times n$ ,  $n \times r$  ( $r \leq n$ ), and  $m \times n$  ( $m \leq n$ ), respectively. The control function  $u(t)$  is required to minimize

$$J = \lim_{t_1 \rightarrow \infty} \int_0^{t_1} [e'(t)Qe(t) + u(t')Ru(t)]dt$$

where

$$e(t) = y(t) - y_m(t)$$

$$y_m(t) = \mathbf{H}_m \mathbf{x}_m(t)$$

and

$$\dot{\mathbf{x}}_m(t) = \mathbf{A}_m \mathbf{x}_m(t)$$

where  $\mathbf{x}_m(0) = \mathbf{x}_m^0$  is given. The constant matrices  $\mathbf{H}_m$  and  $\mathbf{A}_m$  have dimension  $m \times \ell$  ( $m \leq \ell$ ) and  $\ell \times \ell$ , respectively. Also,  $Q = Q' \geq 0$  and  $R = R' > 0$ . The optimization of the performance index causes the output  $y(t)$  of the state to track the output  $y_m(t)$  of a prescribed model. After substituting  $e(t)$  into the performance index, the model-following problem

can be transformed into choosing  $u(t)$  to minimize

$$J = \lim_{t_1 \rightarrow \infty} \int_0^{t_1} [\tilde{x}'(t) \tilde{Q} \tilde{x}(t) + u'(t) R u(t)] dt$$

with

$$\dot{\tilde{x}}(t) = \tilde{A} \tilde{x}(t) + \tilde{B} u(t)$$

where

$$\tilde{A} = \begin{bmatrix} A & 0 \\ 0 & A_m \end{bmatrix}$$

$$\tilde{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}$$

$$\tilde{Q} = \begin{bmatrix} H'QH & -H'QH_m \\ -H_m'QH & H_m'QH_m \end{bmatrix}$$

and

$$\tilde{x} = \begin{bmatrix} x \\ x_m \end{bmatrix}$$

This transformed problem can be solved directly using optimal linear regulator theory. If the  $(\tilde{A}, \tilde{B})$  pair is stabilizable and the  $(\tilde{A}, D)$  pair (with  $D'D = \tilde{Q}$ ) is detectable, the solution exists and is given by

$$u(t) = -F\tilde{x}(t) = -F_{11}x(t) - F_{12}x_m(t)$$

Computationally, it is inefficient to work with the composite  $(\tilde{A}, \tilde{B})$  system directly. If the steady-state Riccati equation is formed and  $\tilde{A}$ ,  $\tilde{B}$  and  $\tilde{Q}$  are substituted, it readily follows (Ref. 29) that

$$F_{11} = R^{-1}B'P_{11}$$

with  $P_{11} = P_{11}' \geq 0$  satisfying

$$P_{11}A + A'P_{11} - P_{11}BR^{-1}B'P_{11} + H'QH = 0$$

and

$$F_{12} = R^{-1}B'P_{12}$$

with  $P_{12}$  satisfying

$$P_{12}A_m + (A - BF_{11})'P_{12} = H'QH_m$$

The computation of  $(F_{11}, F_{12})$  thus separates into two parts:

- (1) Evaluate the feedback gain  $F_{11}$  on the state  $x$  by solving a reduced-order optimal regulator problem of the form

$$\dot{x}(t) = Ax(t) + Bv(t)$$

$$y(t) = Hx(t)$$

$$\min_{v(t)} \left\{ \lim_{t_1 \rightarrow \infty} \int_0^{t_1} [y'(t)Qy(t) + v'(t)Rv(t)] dt \right\}$$

leading to

$$v(t) = -F_{11}x(t)$$

- (2) Using  $F_{11}$  from step (1), compute the feedforward gain  $F_{12}$  on the model  $x_m$  from the linear equations

$$P_{12}A_m + (A - BF_{11})'P_{12} = H'QH_m$$

$$RF_{12} = B'P_{12}$$

C-2 Discrete Model-Following Problem

For the discrete case, the state and output equations are given as

$$x(i+1) = Ax(i) + Bu(i)$$

$$y(i) = Hx(i)$$

with A, B and H as previously defined. The control sequence  $u(i)$  ( $i = 0, 1, \dots, N-1$ ) is required to minimize

$$J = \lim_{N \rightarrow \infty} \sum_{i=0}^{N-1} [e'(i+1)Qe(i+1) + u'(i)Ru(i)]$$

where

$$e(i) = y(i) - y_m(i)$$

$$y_m(i) = H_m x_m(i)$$

$$x_m(i+1) = A_m x_m(i)$$

with Q, R,  $H_m$  and  $A_m$  as previously defined. As in the continuous case, the discrete model-following problem can be solved in terms of an  $(\tilde{A}, \tilde{B}, \tilde{Q}, \tilde{R})$  optimal regulator formulation, but a simplified computational algorithm also exists (Ref. 150:

- (1) Compute a feedback gain  $F_{11}$  on the state  $x$  by solving the reduced-order optimal regulator problem

$$x(i+1) = Ax(i) + Bv(i)$$

$$y(i) = Hx(i)$$

$$\min_{v(i)} \left\{ \lim_{N \rightarrow \infty} \sum_{i=0}^{N-1} [y'(i+1)Qy(i+1) + v'(i)Rv(i)] \right\}$$

leading to

$$v(i) = -F_{11}x(i)$$

- (2) Using  $(P_{11}, F_{11})$  from step (1), compute a feedforward gain  $F_{12}$  on the model state  $x_m$  from the linear equations,

$$P_{12} = (A - BF_{11})'P_{12}A_m - H'QH_m$$

$$(B'P_{11}B + R)F_{12} = B'P_{12}A_m$$

The complete optimal model-following control law is then given by

$$u(i) = -F_{11}x(i) - F_{12}x_m(i).$$

## VITA

Philip Edward Zwicke was born on July 24, 1949, in Sacramento, California. He received the B.S. and M.S. degrees in Electrical Engineering from Virginia Polytechnic Institute and State University, Blacksburg, in 1971 and 1973, respectively.

From June 1974 to September 1975 he was with NORD Instruments, Roanoke, Virginia, where he developed and implemented the control algorithms for computer-controlled forge presses. From October 1975 to December 1976 he was an Instructor in the Department of Electrical Engineering at VPI & SU. For the past two years he has been supported as a Research Associate for a NASA/LRC project applying stochastic adaptive control algorithms on the B737 aircraft.

Mr. Zwicke is a member of the Institute of Electrical and Electronics Engineers, Eta Kappa Nu and Phi Kappa Phi. His major interests are in the theory and applications of modern control and estimation theory.

*Philip Edward Zwicke*

---

Philip Edward Zwicke

STOCHASTIC ADAPTIVE ESTIMATION WITH APPLICATIONS  
TO NONLINEAR CONTROL

by

Philip Edward Zwicke

(ABSTRACT)

This dissertation is concerned with the development of two adaptive state estimators that are capable of tracking linear plants that undergo rapid configuration changes. The first is a modification of the Partitioned Adaptive Estimator, PAE, first introduced by Magill in 1965, improved and named by Lainiotis, and used in a number of applications, primarily aerospace. The PAE algorithm was derived for the problem of identifying which, of  $N$ , configurations that a linear plant is in; the key assumption being that the configuration is unknown but unchanging. There are two main difficulties in extending the PAE algorithm to the problem of estimating the state of a linear plant that can undergo configuration changes (the switched-linear plant problem). These two difficulties are addressed and solved in this dissertation. The result is called the modified PAE algorithm.

The second adaptive estimator developed in this dissertation is the "Sliding Window Detector/Estimator" or SWDE algorithm. Unlike the modified PAE algorithm whose basic structure is designed to solve a different problem, the SWDE algorithm is designed specifically for the switched-linear plant problem. It uses a joint detection/estimation approach to give a very close approximation to the unrealizable optimum

switched-linear estimator.

The advantages and disadvantages of the two adaptive estimators are discussed, and it is found that a very reliable and accurate estimator can be constructed by combining both algorithms. Several different examples are given to clarify the operation of the estimator.

A second contribution of this dissertation is in the application of the above estimators to the nonlinear estimation problem. The motivation for this approach is that a nonlinear plant can be approximated by a sequence of linear approximations, or configurations. Thus, an estimator that works for a switched-linear plant can perform as a sub-optimum nonlinear estimator. In addition, a stochastic nonlinear controller can be constructed using the nonlinear estimator as the observer. This approach has several significant implementation and design advantages which are discussed in the dissertation and illustrated by two examples, a set-point control example and a trajectory-following aircraft example.

The above examples and algorithms were fully verified by extensive computer simulation. The implementation advantages afforded by these methods make them practical in a wide variety of applications.