

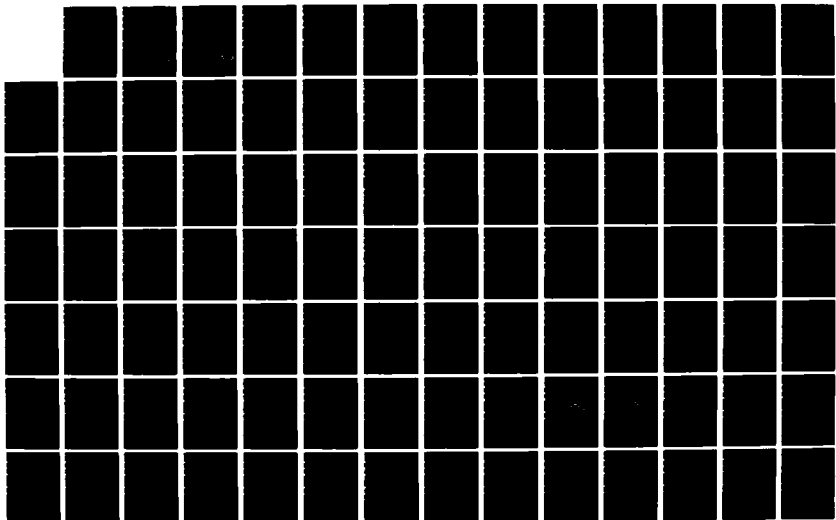
AD-A164 016

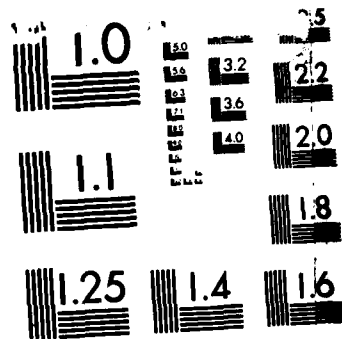
MOVING-BANK MULTIPLE MODEL ADAPTIVE ALGORITHMS APPLIED
TO FLEXIBLE SPACEC (U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI P G FILIOS
DEC 85 AFIT/GE/ENG/85D-14 F/G 22/2

1/2

UNCLASSIFIED

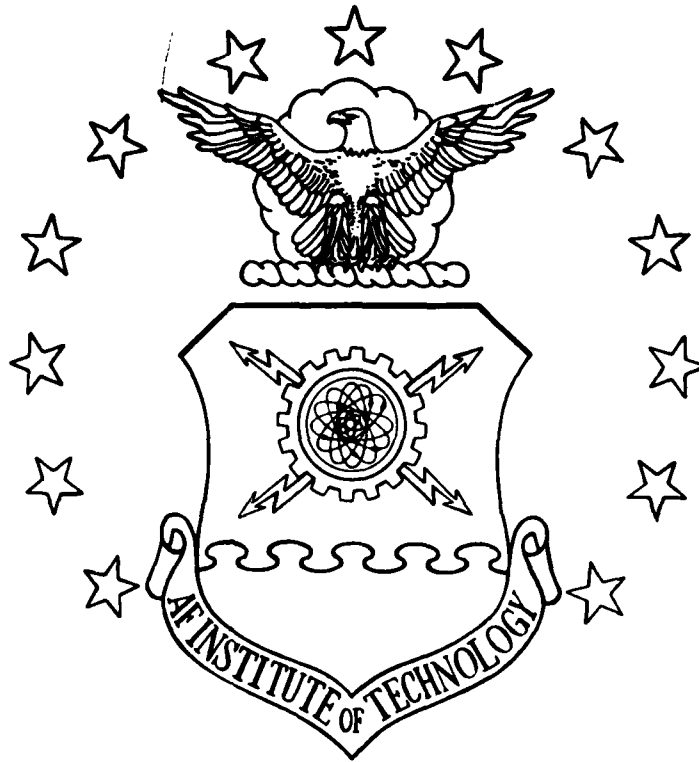
NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A164 016



MOVING-BANK MULTIPLE MODEL ADAPTIVE
ALGORITHMS APPLIED TO FLEXIBLE
SPACECRAFT CONTROL

THESIS

Paul G. Filios
Captain, USAF

AFIT/GE/ENG/85D-14

DISTRIBUTION STATEMENT A

Approved for public release
Distribution Unlimited

DTIC
ELECTE
FEB 13 1986

S

B

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

86 2 12 0

DTIC FILE COPY

**MOVING-BANK MULTIPLE MODEL ADAPTIVE
ALGORITHMS APPLIED TO FLEXIBLE
SPACECRAFT CONTROL**

THESIS

**Paul G. Filios
Captain, USAF**

AFIT/GE/ENG/85D-14

**DTIC
ELECTE
S FEB 13 1986 D
B**

DISTRIBUTION STATEMENT A

**Approved for public release
Distribution Unlimited**

AFIT/GE/ENG/85D-14

**MOVING-BANK MULTIPLE MODEL ADAPTIVE ALGORITHMS
APPLIED TO FLEXIBLE SPACECRAFT CONTROL**

THESIS

**Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology**

Air University

**In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering**

Paul G. Filios, B.S.E.E.

Captain, USAF

December 1985

Approved for public release; distribution unlimited

Preface

The purpose of this effort was to demonstrate that moving-bank multiple model adaptive estimation algorithms could be applied to a realistic and practical control problem. Moving-bank multiple model adaptive estimation/control is an attempt to create from full-bank multiple model adaptive estimation/control an adaptive estimation/control technique which maintains its desirable on-line parameter adaptation, but which reduces the intensive computational loading which makes the full-bank method impractical. The results of this thesis do indicate that the moving-bank multiple model adaptive algorithm can be successful in this goal.

I wish to thank my thesis advisor, Dr. P. S. Maybeck for the help he gave me at every step of this effort, and for the insights he gave me into the algorithms explored. I would also like to thank my classmates for thier encouragement, especially S. Eckert for helping me procure valuable computer resources.

Paul G. Filios

Accession No.	
Author	
Editor	
Printer	
AVAIL	
1	
A-1	



Contents

	Page
Preface	ii
List of Figures	v
List of Tables	vi
Abstract	vii
I. Introduction	1
I.1. Background	1
I.2. Problem	6
I.3. Scope	7
I.4. Approach	8
I.5. Organization	9
II. Algorithm Development	10
II.1. Introduction	10
II.2. Bayesian Estimation Algorithm Development	11
II.3. Moving-Bank Algorithm Development	22
II.4. Ambiguity Functions Analysis.	33
II.5. Summary	35
III. Space Structure Model	38
III.1. Introduction	38
III.2. Second Order Form	40
III.3. State Space Form	45
III.4. Stochastic Form.	47
III.5. Physical System Constants.	47
III.6. State Reduction	48
III.7. Summary	49
IV. Simulation	50
IV.1. Introduction	50
IV.2. Monte Carlo Analysis	50
IV.3. Software Description.	55
IV.4. Simulation Plan	59
IV.5. Summary	50

	Page
V. Results	64
V.1. Introduction	64
V.2. Numerical Problems	64
V.3. Ambiguity Function Analysis.	66
V.4. Bank Movement	73
V.5. Bank Contraction	78
V.6. Jump Changes	83
V.7. Controller Analysis.	84
V.8. Summary	84
VI. Conclusions and Recommendations	86
VI.1. Introduction	86
VI.2. Conclusions	87
VI.3. Areas of Further Study	91
VI.4. Summary	93
Appendix A: Detailed Model Development	94
Appendix B: Simulation Plots	102
Appendix C: Simulation Software	133
Bibliography	167
Vita	170

List Of Figures

Figure		Page
I-1.	Moving-Bank Multiple Model Adaptive Estimator .	6
II-1.	Multiple Model Filtering Algorithm	14
II-2.	Bank Discretizations	28
II-3.	Bank Move	30
III-1.	Draper/RPL Configuration Model	39
III-2.	Anti-Symmetric Deflections	41
III-3.	Opposition and Unison Modes	45
IV-1.	System, Estimator, and Controller Simulation . .	53
V-1.	Ambiguity Function at point (5,5)	67
V-2.	Ambiguity Function at point (7,3)	68
V-3.	Ambiguity Function at point (3,7)	69

List of Tables

Table	Page
III-1. Configuration Constants for Draper/RPL Model . .	48
A-1. Eigenvalues for Second Bending Mode	100
A-2. Eigenvalues for Fourth Bending Mode	101

Abstract

This investigation applies moving-bank multiple model adaptive estimation/control algorithms to the control of a realistic model of a large flexible spacecraft. Moving-bank multiple model adaptive estimation differs from conventional (full-bank) multiple model adaptive estimation in that a substantially reduced number of elemental filters is required for the moving-bank estimator (9 vs. 100 for the system modeled in this thesis). The positions in parameter space that the reduced number of elemental filters occupy are dynamically re-declared; i.e., the moving-bank slides about the parameter space in search of the true parameter point.

Critical to the performance of the moving-bank multiple model adaptive estimator is the decision logic used to determine which elemental filters are implemented in the bank, and when to change this decision. The decision logics discussed focus on three situations: initial acquisition of the unknown parameter values, through reducing bank discretization; tracking the unknown parameter values, through bank movement; and reacquisition of the unknown parameters following a large jump change in their values, through expanding bank discretization. Ambiguity function analysis is used to predict performance in these situations.

The system to be controlled is a simplified model of a large scale space structure. Its equations of motion are developed and placed in state space form, the states being the positions and velocities of the rigid body mode and the second and fourth bending modes. The state space matrices describing the system are computed based on nominal values for all physical parameters with the exception of the mass density of the structure arms and their modulus of elasticity. These two parameters are allowed to vary in discrete steps, establishing the parameter space. It is then attempted to control the states to the quiescent state, using moving-bank multiple model adaptive algorithms.

The results indicate that, although the system under study did not have a great need for adaptive estimation and control, the multiple model adaptive estimator performs essentially identically to a single filter artificially knowledgeable of the uncertain parameter values. In addition changing bank discretization for the initial parameter acquisition phase speeded acquisition. However, the bank was unable to expand following a jump change in the uncertain parameter values, in order to restart the acquisition phase the bank tracked the jump change through movement alone. Ambiguity function analysis proved to be an excellent predictor of bank performance, and should be used as a design tool.

MOVING-BANK MULTIPLE MODEL ADAPTIVE ALGORITHMS APPLIED TO FLEXIBLE SPACECRAFT CONTROL

I. Introduction

A common problem in estimation and control problems is the uncertainty of parameters used in the design of the system model and embedded in the estimator or controller. These parameters can be initially unknown but unchanging, or they can vary slowly, or they can undergo abrupt changes as in the case of partial system failure. If these parameters vary enough, it may be necessary to estimate their values along with the system state, and adapt the estimator, controller, or both, to incorporate the current value of the uncertain parameters. This requirement is not based solely upon the magnitude of the parameter variation but also upon the sensitivity of the system to the variation. Some parameters may vary widely with no degradation in system performance, while a small variation in another parameter may cause the system to become unstable. This thesis expands the exploration of one method for accomplishing this, known as moving-bank multiple model adaptive estimation.

I.1 Background

In a large class of problems, which can be modeled as linear stochastic systems with uncertain parameters

affecting the matrices defining the state model, characteristics of driving noises, or measurement devices, Kalman filters can be used for the estimation/control algorithms if a means for adapting them to the uncertainties can be found.

Multiple model adaptive estimation (MMAE) [1,2,3,4,5:129-135] is a means of adapting the Kalman filters for the case where the uncertain parameters can be modeled as assuming only discrete values, as opposed to a value in a continuous range; this may either be physically reasonable (as in the case of failure states) or representative values can be chosen from a continuous range of possible values. This approach creates a bank of elemental Kalman filters, one for each possible value of the uncertain parameter vector. The output of each elemental filter is then weighted by the a posteriori probability of that particular parameter value vector being correct, conditioned on the observed history of measurements. The conditional probabilities are computed iteratively based upon the observed characteristics of the filter residuals. The adaptive state estimate is then the sum of the weighted elemental filter outputs. The equations for this algorithm are fully developed in the next chapter. As an alternative, using maximum a posteriori (MAP) criteria for optimality, the output of the elemental filter with the largest associated conditional probability can be used as the state

The convergence of the MMAE algorithm estimate to the true state has been shown for the case of uncertain but constant parameters [6, 7]. Convergence has not been shown for varying parameters; however, promising results have been obtained for an ad hoc approach where constant parameters are assumed for algorithm design, but the computed conditional probabilities are lower bounded to prevent 'locking onto' a single parameter value [5,8,9].

If the possible parameter time variations are modeled, the optimal state estimate can be obtained as the weighted sum of the estimates produced by filters matched to all possible parameter time histories [5]. This approach is, however, impractical as K^i elemental filters at sample time t_i would be required. Even when the parameter temporal variation can be well modeled as a Markov process, it has been shown that the number of elemental estimators required would be K^2 [5,10]. Thus if only 2 uncertain parameters were each discretized to 10 values, K would be equal to 100, and $K^2 = 10,000$: still impractical for implementation [5].

For control applications, the state estimate obtained from a multiple model adaptive estimator can be premultiplied by a controller gain established via forced equivalence design [12]. The gain may be based on a single nominal parameter value, or can be evaluated using the estimated values of the uncertain parameters provided by the estimator. If a separate controller gain is associated with each elemental filter in the bank, the

control can be produced as the probabilistically weighted average of the elemental filter/controller outputs, in the same way the state estimate is obtained in a multiple model adaptive estimator. This is known as multiple model adaptive control (MMAC) [12:253,11]. An alternative, using MAP criteria, is to select that filter/controller output associated with the highest weighting probability [11].

MMAE has successfully been applied to several practical problems. The tracking of maneuvering targets has been shown to lend itself to this approach [13,14,15,16,17]. Other applications have been demonstrated in flight control [11], multiple hypotheses testing [18], detection of incidents on freeways [19], adaptive deconvolution of seismic signals [20], and problems in which initial uncertainties are so large that nonadaptive extended Kalman filters diverge [21,22].

In implementations where there are a large number of uncertain parameters, or where the uncertain parameters can take on a large number of discrete values, the storage required for the elemental filters, and the computational loading required become unwieldy. Consider, if there are only 2 uncertain parameters, each of which can assume 10 discrete values, the number of elemental filters is $10^2 = 100$; this may be implementable but it can be seen that the number of filters required will grow quickly with additional uncertain parameters or finer levels of discretization. Approaches to reducing the computational

burden include: use of Markov models for parameter variation [5,10,23]; 'pruning' and 'merging' of branches in a 'tree' of possible parameter time histories [24,25], hierarchical structuring [26], and dynamic coarse-to-finer discretization [27].

A method proposed by Hentz and Maybeck [28,29] involves implementing only a smaller number of elemental filters selected from the total bank. This 'moving-bank' is adjusted by replacing filters currently implemented with others from the larger bank in response to the changes in the weighting probabilities and filter residuals. In the above example, only the filters with the three closest discrete values to each of the estimated parameter values might be implemented. This would result in only nine filters being implemented, and as the estimate of the parameters changed, the implemented bank of filters would 'move'; see Figure I-1. This approach was shown to achieve performance essentially equivalent to that of a single Kalman filter artificially knowledgeable of true parameter values, for a simple but physically motivated example of a single input/single output second order system with uncertain damping ratio, and undamped natural frequency. This corresponds, for example, to a bending mode in an aerospace vehicle [28:15,57-101; 29:17-27]. The algorithms used for implementing the moving-bank are developed in detail in Chapter II.

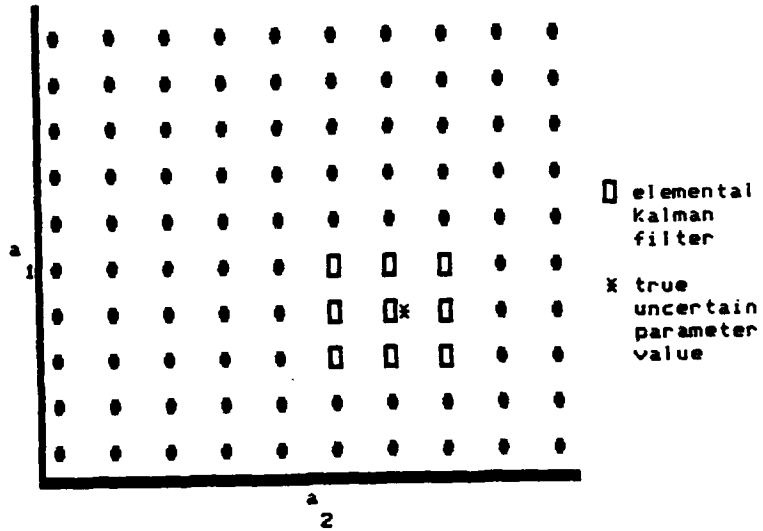


Figure I-1. Moving-Bank Multiple Model Adaptive Estimator.

I.2 Problem

Full-scale MMAE requires too large a computational load to be practical for most applications requiring adaptive estimation/control. Moving-bank MMAE has been shown to be feasible for a simple but physically motivated application. However, it still needs to be demonstrated that moving-bank MMAE can be useful in a more complicated and realistic application requiring adaptive estimation/control.

The purpose of this effort is to apply moving-bank MMAE to a practical application with a higher level of complexity and further evaluate its performance potential. In addition alternative decision logics for choosing which of the elemental filters in the parameter space to implement in the moving-bank will be evaluated.

I.3 Scope

A moving-bank multiple model adaptive estimator/controller is evaluated for a realistic application. A simplified model of a large flexible spacecraft, consisting of a central rigid hub with four radiating flexible arms, is used. The flexible arms are representative of the flexible appendages such as solar panels or antennas attached to actual spacecraft. The spacecraft is described in terms of its physical parameters (mass, height, length of the arms, etc.), with two uncertain parameters (mass density of the arms, and the modulus of elasticity of the arms). The uncertain parameters are discretized into 10 values each, to provide a ten by ten (100 point) parameter space. Three point actuators (representative of the pulse rocket motors used on actual spacecraft) provide the control input, and five position sensors along with five velocity sensors provide the measurements for updating the state estimates. The dynamics and measurement noise characteristics are assumed known with no uncertainty, and are modeled as white Gaussian processes with strengths Q and R respectively.

I.4 Approach

The performance of the moving-bank multiple model adaptive estimator/controller, its ability to estimate both the states and parameters as well as to apply adequate control to the true system is compared to a benchmark of a single Kalman filter/controller based on (artificial) perfect knowledge of the true parameter value. Estimator performance is evaluated under the following conditions:

a. The true parameter value is constant and exactly equal to one of the discretized parameter values within the initial conditions chosen for the bank.

b. The true parameter value is constant and lies outside the initial conditions chosen for the bank (but within the overall parameter space).

c. The true parameter value undergoes jumps in value from one of the discretized parameter values to another, the values chosen so that the new true parameter point is outside the area covered by the moving bank.

True parameter values which are not exactly equal to one of the discrete parameter values must be investigated before a full evaluation of moving-bank multiple model adaptive estimation/control can be done [28]; however, due to time constraints, they are not treated in this thesis.

The effect on performance and computational burden of various algorithms that can be used for initial acquisition of the true parameter values, and for identification of when

a jump change has taken place, are of interest. Specific algorithms developed in Chapter II will be tested to determine their effect on estimator performance.

Comparison of controller design strategies, contrasting a single fixed-gain controller, a single changeable-gain controller, and a moving-bank multiple model adaptive controller, will also be accomplished.

I.5 Organization

The remaining sections of this thesis are organized as follows. Chapter II develops the algorithms used for both full-scale MMAE and moving-bank MMAE. Chapter III presents the flexible spacecraft structure model. Chapter IV details the simulation used for the moving-bank estimator/controller performance evaluation. Chapter V contains an analysis of the results of the evaluations. And Chapter VI presents conclusions and recommendations.

II. ALGORITHM DEVELOPMENT

II.1. Introduction

This chapter details the algorithms for two of the adaptive estimation techniques introduced in the first chapter. First full-scale Bayesian Multiple Model Adaptive Estimation is developed, then the modifications necessary for the moving-bank case are discussed.

II.2. Bayesian Estimation Algorithm Development

Full-scale Bayesian Multiple Model Adaptive Estimation algorithms are presented in this section. For a more complete development of concepts, see reference [5:129-136].

Let the system under consideration be described by the following:

$$\underline{\tilde{x}}(t_{i+1}) = \Phi(t_{i+1}, t_i) \underline{\tilde{x}}(t_i) + B_d(t_i) \underline{u}(t_i) + G_d(t_i) \underline{\tilde{w}}_d(t_i)$$

$$\underline{\tilde{z}}(t_i) = H(t_i) \underline{\tilde{x}}(t_i) + \underline{\tilde{v}}(t_i)$$

where, letting the '~' denote random process:

$\underline{\tilde{x}}(t_i)$:	state vector
$\Phi(t_{i+1}, t_i)$:	state transition matrix
$\underline{u}(t_i)$:	known input vector
$B_d(t_i)$:	control input matrix
$\underline{\tilde{w}}_d(t_i)$:	white Gaussian dynamics noise vector
$G_d(t_i)$:	noise input matrix

$\underline{z}(t_i)$: measurement vector
 $\underline{H}(t_i)$: measurement matrix
 $\underline{v}(t_i)$: white Gaussian measurement noise vector

and the following statistics apply:

$$\begin{aligned}
 E\{\underline{w}_d(t_i)\} &= \underline{0} \\
 E\{\underline{w}_d(t_i)\underline{w}_d^T(t_j)\} &= Q_d(t_i)\delta_{ij} \\
 E\{\underline{v}(t_i)\} &= \underline{0} \\
 E\{\underline{v}(t_i)\underline{v}^T(t_j)\} &= R(t_i)\delta_{ij}
 \end{aligned}$$

where δ_{ij} is the Kronecker delta function. It is also assumed that $\underline{x}(t_0)$, $\underline{w}_d(t_i)$, and $\underline{v}(t_i)$ are independent for all t_i .

Now, let \underline{a} be the uncertain p -dimensional parameter vector which is an element of A , where A is a subset of R^p . \underline{a} may be uncertain but constant, it may be slowly varying, or it may undergo jump changes. The parameter vector \underline{a} can affect any or all of the following: ϕ , B_d , G_d , Q_d , H , and R . The Bayesian estimator computes the following conditional probability density function:

$$\begin{aligned}
 f_{\underline{x}(t_i), \underline{a} | \underline{Z}(t_i)}(\underline{x}, \underline{a} | \underline{Z}_i) &= f_{\underline{x}(t_i) | \underline{a}, \underline{Z}(t_i)}(\underline{x} | \underline{a}, \underline{Z}_i) \\
 &\cdot f_{\underline{a} | \underline{Z}(t_i)}(\underline{a} | \underline{Z}_i)
 \end{aligned} \tag{II-1}$$

where $\underline{Z}(t_i)$ is a vector of measurements from t_0 to t_i ,

$$\underline{Z}(t_i) = [\underline{z}^T(t_i), \underline{z}^T(t_{i-1}), \dots, \underline{z}^T(t_0)]^T$$

The second term on the right side of Equation (II-1) can be further evaluated:

$$\begin{aligned}
 f_{\underline{a}}|\underline{Z}(t_i)(\underline{a}|\underline{Z}_i) &= f_{\underline{a}}|\underline{Z}(t_i), \underline{Z}(t_{i-1})(\underline{a}|\underline{z}_i, \underline{Z}_{i-1}) \\
 &= f_{\underline{a}, \underline{z}_i}(t_i) | \underline{Z}(t_{i-1})(\underline{a}, \underline{z}_i | \underline{Z}_{i-1}) \\
 &= f_{\underline{z}_i}(t_i) | \underline{Z}(t_{i-1})(\underline{z}_i | \underline{Z}_{i-1}) \\
 &= f_{\underline{z}_i}(t_i) | \underline{a}, \underline{Z}(t_{i-1})(\underline{z}_i | \underline{a}, \underline{Z}_{i-1}) f_{\underline{a}}|\underline{Z}(t_{i-1})(\underline{a} | \underline{Z}_{i-1}) \\
 &= \int_A f_{\underline{z}_i}(t_i) | \underline{a}, \underline{Z}(t_{i-1})(\underline{z}_i | \underline{a}, \underline{Z}_{i-1}) f_{\underline{a}}|\underline{Z}(t_{i-1})(\underline{a} | \underline{Z}_{i-1}) d\underline{a} \quad (II-2)
 \end{aligned}$$

Conceptually, Equation (II-2) can now be solved recursively, starting from an a priori probability density of $f_{\underline{a}}(\underline{a})$, since $f_{\underline{z}_i}(t_i) | \underline{a}, \underline{Z}(t_{i-1})(\underline{z}_i | \underline{a}, \underline{Z}_{i-1})$ is Gaussian with a mean of $H(t_i)\underline{x}(t_i^-)$ and covariance $[H(t_i)P(t_i^-)H^T(t_i)+R(t_i)]$, where $\underline{x}(t_i^-)$ and $P(t_i^-)$ are the conditional mean and covariance respectively of $\underline{x}(t_i)$ just prior to the measurement at t_i , assuming a particular realization \underline{a} of \underline{a} .

Using the conditional mean, the estimate of $\underline{x}(t_i)$ can be generated as:

$$\begin{aligned}
 E\{\underline{x}(t_i) | \underline{Z}(t_i) = \underline{Z}_i\} &= \int_{-\infty}^{\infty} \underline{x} f_{\underline{x}(t_i) | \underline{Z}(t_i)}(\underline{x} | \underline{Z}_i) d\underline{x} \\
 &= \int_{-\infty}^{\infty} \underline{x} \left[\int_A f_{\underline{x}(t_i), \underline{a}} | \underline{Z}(t_i)(\underline{x}, \underline{a} | \underline{Z}_i) d\underline{a} \right] d\underline{x} \quad (II-3)
 \end{aligned}$$

$$\begin{aligned}
 &= \int_{-\infty}^{\infty} \underline{x} \left[\int_A f_{\underline{x}(t_i) | \underline{a}, \underline{Z}(t_i)}(\underline{x} | \underline{a}, \underline{Z}_i) f_{\underline{a}}|\underline{Z}(t_i)(\underline{a} | \underline{Z}_i) d\underline{a} \right] d\underline{x} \\
 &= \int_A \int_{-\infty}^{\infty} \underline{x} f_{\underline{x}(t_i) | \underline{a}, \underline{Z}(t_i)}(\underline{x} | \underline{a}, \underline{Z}_i) \underline{x} f_{\underline{a}}|\underline{Z}(t_i)(\underline{a} | \underline{Z}_i) d\underline{x} d\underline{a} \quad (II-4)
 \end{aligned}$$

where the term in brackets is the estimate of $\underline{x}(t_i)$ based on a particular value of the parameter vector. This would be the output of the Kalman filter based on a particular parameter value. When \underline{a} is continuous over A, this would require an infinite number of filters in the bank. In the continuously distributed parameter case, what is typically done is to discretize the parameter space, yielding a finite number of filters. The integrals over A in Equations (II-2) through (II-4) then become summations: letting $p_k(t_i)$ be defined as the probability of the k^{th} elemental filter being correct, conditioned on the measurement history, (II-2) and (II-4) are replaced by:

$$p_k(t_i) = \frac{f_{\underline{z}}(t_i) | \underline{a}_k, \underline{Z}(t_{i-1}) (z_i | \underline{a}_k, \underline{Z}_{i-1}) p_k(t_{i-1})}{\sum_{j=1}^K f_{\underline{z}}(t_i) | \underline{a}_j, \underline{Z}(t_{i-1}) (z_i | \underline{a}_j, \underline{Z}_{i-1}) p_j(t_{i-1})} \quad (\text{II-5})$$

$$\hat{\underline{x}}(t_i^+) = E\{\underline{x}(t_i) | \underline{Z}(t_i) = \underline{Z}_i\}$$

$$= \sum_{k=1}^K \hat{\underline{x}}_k(t_i^+) p_k(t_i) \quad (\text{II-6})$$

where $\underline{a} = [\underline{a}_1, \underline{a}_2, \dots, \underline{a}_K]$ and $\hat{\underline{x}}_k(t_i^+)$ is the mean of $\underline{x}(t_i)$ conditioned on $\underline{a} = \underline{a}_k$ and $\underline{Z}(t_i) = \underline{Z}_i$, i.e. the output of the k^{th} Kalman filter in the bank, based on the assumption that $\underline{a} = \underline{a}_k$. Pictorially the algorithm appears as in Figure II-1.

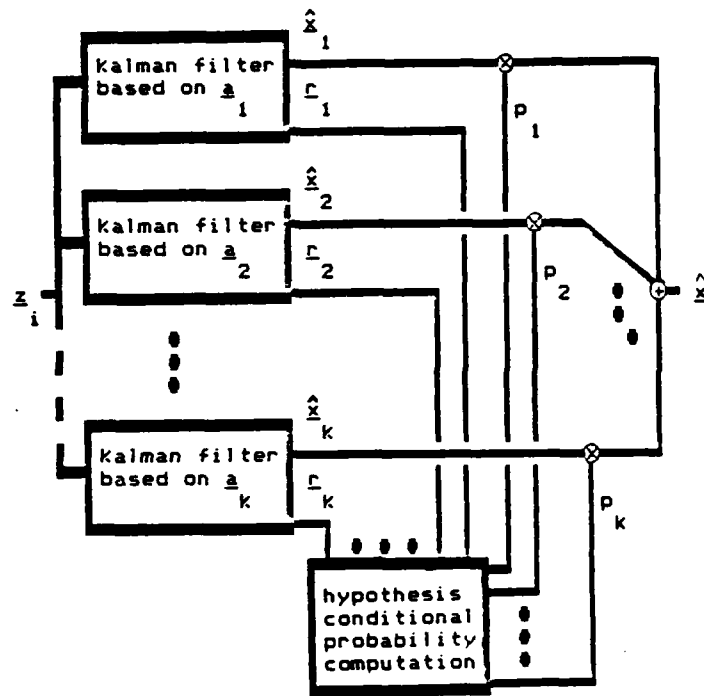


Figure II-1. Multiple Model Filtering Algorithm

The probability weighting factors for each Kalman filter are calculated from Equation (II-5), where:

$$f_{\underline{z}(t_i) | \underline{a}, \underline{Z}(t_{i-1})}(\underline{z}_i | \underline{a}_k, \underline{Z}_{i-1}) = \frac{1}{(2\pi)^{m/2} |A_k(t_i)| (1/Z)} \exp[-(1/2) \underline{r}_k^T(t_i) A_k^{-1}(t_i) \underline{r}_k(t_i)] \quad (\text{II-7})$$

and

$$A_k(t_i) = H_k(t_i) P_k(t_i^-) H_k^T(t_i) + R_k(t_i)$$

$$\underline{r}_k(t_i) = \underline{z}_i - H_k(t_i) \hat{\underline{x}}_k(t_i^-)$$

m = dimension of \underline{z} (the number of measurements)

Both the residual covariance $A_k(t_i)$ and the residual $\underline{r}_k(t_i)$ itself are readily available from the k^{th} elemental filter.

The estimate of the parameter and the covariance of the parameter are given by:

$$\begin{aligned}
 \hat{\underline{a}}(t_i) &\stackrel{\Delta}{=} E\{\underline{a} | \underline{Z}(t_i) = \underline{Z}_i\} = \int_{-\infty}^{\infty} \underline{a} f_{\underline{a} | \underline{Z}(t_i)}(\underline{a} | \underline{Z}_i) d\underline{a} \\
 &= \int_{-\infty}^{\infty} \underline{a} \left[\sum_{k=1}^K p_k(t_i) \delta(\underline{a} - \underline{a}_k) \right] d\underline{a} \\
 &= \sum_{k=1}^K \underline{a}_k p_k(t_i) \quad (II-8)
 \end{aligned}$$

and

$$\begin{aligned}
 E\{[\underline{a} - \hat{\underline{a}}(t_i)][\underline{a} - \hat{\underline{a}}(t_i)]^T | \underline{Z}(t_i) = \underline{Z}_i\} = \\
 \sum_{k=1}^K [\underline{a}_k - \hat{\underline{a}}(t_i)][\underline{a}_k - \hat{\underline{a}}(t_i)]^T p_k(t_i) \quad (II-9)
 \end{aligned}$$

The covariance of the state estimate is given by:

$$\begin{aligned}
 P(t_i^+) &= E\{[\underline{x}(t_i) - \hat{\underline{x}}(t_i^+)] [\underline{x}(t_i) - \hat{\underline{x}}(t_i^+)]^T | \underline{Z}(t_i) = \underline{Z}_i\} \\
 &= \int_{-\infty}^{\infty} [\underline{x} - \hat{\underline{x}}(t_i^+)] [\underline{x} - \hat{\underline{x}}(t_i^+)]^T f_{\underline{x}(t_i) | \underline{Z}(t_i)}(\underline{x} | \underline{Z}_i) d\underline{x} \\
 &= \sum_{k=1}^K p_k(t_i) \{ [\underline{x} - \hat{\underline{x}}(t_i^+)] [\underline{x} - \hat{\underline{x}}(t_i^+)]^T \\
 &\quad \cdot f_{\underline{x}(t_i) | \underline{a}, \underline{Z}(t_i)}(\underline{x} | \underline{a}, \underline{Z}_i) d\underline{x} \} \\
 &= \sum_{k=1}^K p_k(t_i) \{ P_k(t_i^+) + [\hat{\underline{x}}(t_i^+) - \hat{\underline{x}}(t_i^+)] \\
 &\quad [\hat{\underline{x}}(t_i^+) - \hat{\underline{x}}(t_i^+)]^T \} \quad (II-10)
 \end{aligned}$$

where $P_k(t_i^+)$ is the covariance of the state estimate of the k^{th} elemental filter.

It is important to note that it is not necessary to compute the covariance of the parameter estimate and the covariance of the state estimate online. They are used for the evaluation of estimator performance during estimator design and tuning.

II.2.1. Elemental Estimator. The elemental estimators are each constant gain-Kalman filters, the gains of which are dependent upon the associated point in the parameter space. Thus, each is designed on the basis of a time invariant system model, and a stationary noise model, and the initial transient in the resulting filter is ignored.

The k^{th} elemental filter estimate of $\underline{x}(t)$, denoted by $\hat{\underline{x}}_k(t)$, is propagated from just after the last measurement to just before the next measurement by:

$$\hat{\underline{x}}_k(t_i^-) = \Phi_k \hat{\underline{x}}_k(t_{i-1}^+) + B_{dk} \underline{u}(t_{i-1}) \quad (\text{II-11})$$

and is updated by:

$$\hat{\underline{x}}_k(t_i^+) = \hat{\underline{x}}_k(t_i^-) + K_k [z(t_i) - H_k \hat{\underline{x}}_k(t_i^-)] \quad (\text{II-12})$$

In these equations, the subscript 'k' identifies each vector and matrix as being specifically based on the k^{th} point in the parameter space.

II.2.2. Convergence. The adaptive estimation algorithm just developed is optimal under the conditions that the discretized parameter space is a true model of the physical parameters, and the true parameter vector remains at a constant, but unknown, value. Under these conditions the algorithm will converge to the true parameter value [6], i.e.,

$$\lim_{i \rightarrow \infty} p_k(t_i) = 0 \quad \text{for } \underline{a} \neq \underline{a}_k$$

$$\lim_{i \rightarrow \infty} p_k(t_i) = 1 \quad \text{for } \underline{a} = \underline{a}_k$$

When the true parameter space is in fact continuous and the true parameters lie somewhere between the discretized points, the algorithm converges to the single discrete parameter point that is 'nearest', as defined in [6], to the true parameters.

These results were extended by Dasgupta and Westphal [30] for the case of unknown biases in the measurement process, ($E\{\underline{v}(t_i)\} = \underline{m}_v(t_i)$, where $\underline{m}_v(t_i)$ can be affected by the parameter vector). Under these conditions the algorithm may converge to a parameter point that is not close to the true value, and erroneous estimates may result.

It is important to remember that the algorithm is able to identify the closest parameter point only by observing which elemental filter consistently has the smallest value of $\underline{r}_k^T \underline{A}_k^{-1} \underline{r}_k$: the smallest residual relative to the k^{th} filter-predicted residual squared value. If the residuals are of the same size, Equation (II-7) shows that the filter

with the smallest value of $|A_k|$ will be identified as the correct filter. Since $|A_k|$ is independent of both the residuals and the elemental filter's correctness, if pseudonoise is added during the filter tuning process, to account for model inadequacies, it must not be so strong as to mask the correctness/incorrectness of the elemental filters. Such strong pseudonoise may well allow the adaptive filter to converge to an erroneous parameter value [30].

As noted in the introduction, no satisfactory theoretical convergence results are available for more general conditions, such as slowly varying parameters, although empirical information suggests convergence. The most successful approach used to prevent the algorithm from locking onto one elemental filter before the parameters have varied significantly has been to lower bound the p_k 's to prevent them from converging to zero [5,10].

II.2.3. Control. There are several 'assumed certainty equivalence design' [12:241] approaches to controlling systems with uncertain parameters that can be used with a multiple model adaptive estimator. In the first, the estimator provides only a state vector estimate to a fixed-gain controller robustified around a nominal value of the uncertain parameter vector, \underline{a}_{nom} . The controller design method and controller gains are independent of the adaptive nature of the estimator.

The controller algorithm is of the form:

$$\underline{u}(t_i) = -G_c^* [t_i, \underline{a}_{nom}] \hat{\underline{x}}(t_i^+) \quad (\text{II-13})$$

or the steady-state constant-gain version:

$$\underline{u}(t_i) = -G_c^* [\underline{a}_{nom}] \hat{\underline{x}}(t_i^+) \quad (\text{II-14})$$

A second approach is to have the estimator provide both an estimate of the state vector and an estimate of the uncertain parameter vector (Equation II-8) to the controller. The controller gain then becomes dependent on the parameter vector estimate:

$$\underline{u}(t_i) = -G_c^* [t_i, \hat{\underline{a}}(t_i^-)] \hat{\underline{x}}(t_i^+) \quad (\text{II-15})$$

or the steady-state constant-gain version:

$$\underline{u}(t_i) = -G_c^* [\hat{\underline{a}}(t_i^-)] \hat{\underline{x}}(t_i^+) \quad (\text{II-16})$$

where $\hat{\underline{a}}(t_i^-)$ (as generated at the previous sample time) is used instead of $\hat{\underline{a}}(t_i^+)$ to reduce computational delay. It is important that the control input be applied as close to the start of the sample period as possible, as the filter propagation equation is written on the assumption that the control input is present for the entire sample period.

A third approach, referred to as multiple model adaptive control, [12:253], is to form K elemental controllers, each associated with one of the elemental estimators in the parameter space. Then the final control input becomes the probabilistically weighted average of the

individual controller results in the same manner as the state estimate was achieved:

$$\underline{u}_k(t_i) = -G_c^* [t_i, \hat{\underline{x}}_k] \hat{\underline{x}}_k(t_i^+) \quad (\text{II-17})$$

$$\underline{u}(t_i) = \sum_{k=1}^K p(t_i) \underline{u}_k(t_i) \quad (\text{II-18})$$

For this thesis, the controllers to be used will be linear, quadratic cost, (LQ) full-state feedback optimal deterministic controllers, designed to regulate the system to the quiescent state, incascade with the state estimator (invoking assumed certainty equivalence). The basic structure of all the controllers described above will be similar, where the gain matrices will be dependent upon a particular parameter vector value.

If we are given the stochastic system [28:33-35]:

$$\dot{\underline{x}}(t) = \underline{F}\underline{x}(t) + \underline{B}\underline{u} + \underline{G}\underline{w}(t) \quad (\text{II-19})$$

where

$$E\{\underline{w}(t)\} = 0 \quad \text{and} \quad E\{\underline{w}(t)\underline{w}^T(t+\tau)\} = Q\delta(\tau)$$

and the quadratic cost function to be minimized is:

$$J = E\left\{ \int_0^{\infty} (1/2) [\underline{x}^T(t) \underline{W}_x \underline{x}(t) + \underline{u}^T(t) \underline{W}_u \underline{u}(t)] dt \right\} \quad (\text{II-20})$$

where W_x and W_u are weighting matrices describing the relative costs associated with deviations of the system states from the quiescent state, and the cost of control inputs. The solution for the optimal control, assuming full-state access, is then the LQG optimal regulator:

$$\underline{u}(t_i) = -G_c^* \underline{x}(t_i) \quad (\text{II-21})$$

where the constant gains, G_c^* , that minimize J are given by [12:68-2]:

$$G_c^* = [U + B_d^T K_c B_d]^{-1} [B_d^T K_c \phi + S^T] \quad (\text{II-22})$$

where K_c satisfies the algebraic Riccati equation

$$K_c = X + \phi^T K_c \phi - [B_d^T K_c \phi + S^T]^T G_c^* \quad (\text{II-23})$$

and

$$X = \int_{t_i}^{t_{i+1}} \phi^T(\tau, t_{i+1}) W_x \phi(\tau, t_i) d\tau$$

$$U = \int_{t_i}^{t_{i+1}} [\bar{B}^T(\tau, t_i) W_x \bar{B}(\tau, t_i) + W_u] d\tau$$

$$S = \int_{t_i}^{t_{i+1}} \phi^T(\tau, t_i) W_x \bar{B}(\tau, t_i) d\tau$$

$$\bar{B}(t, t_i) \stackrel{\Delta}{=} \int_{t_i}^t \phi(t, \tau) B \, d\tau$$

$$B_d = \bar{B}(t_{i+1}, t_i)$$

$\phi(t_2, t_1)$ is the state transition matrix from t_1 to t_2 and:

$$\phi = \phi(t_{i+1}, t_i)$$

Note that Equation (II-21) is also the solution to the deterministic LQ optimal control problem with no driving noise $\underline{w}(t)$, and that if the assumption of full state access is replaced by noise-corrupted measurements being available, then $\underline{x}(t_i)$ in Equation (II-21) is replaced by the state estimates $\hat{\underline{x}}(t_i^+)$ generated by a Kalman Filter. This assumed equivalence is valid if all system parameters are known perfectly. Therefore for the uncertain parameter case G_c^* should be a function of the uncertain parameter vector \underline{a} , and 'forced' or 'assumed' certainty equivalence design is used for synthesis [12:241].

II.3. Moving-Bank Algorithm Development

As the number of uncertain parameters to be estimated grows larger and the discretization of the parameter space becomes finer, the MMAE algorithm becomes computationally impractical for real time applications. Maybeck and Hentz [28,29] have demonstrated that it is feasible to begin

with the entire bank of Kalman filters used for MMAE but only compute the state estimate and parameter estimate for those filters 'closest' to the current estimate of the parameter vector. The p_k 's of the unimplemented filter are inherently set to 0, and all of the probability weighting is distributed amongst the implemented filters. As the parameter vector estimate changes, the implemented bank 'slides' within the larger bank: those implemented filters 'farthest' from the current estimate of the parameter vector are dropped from the bank (no longer computed), and new filters 'closer' to the current parameter vector estimate are implemented instead. It is also possible for the moving bank to change discretization level, i.e., the filters implemented need not be adjacent in the full bank. Maybeck and Hentz examined changing discretization level, starting with a coarse discretization during an acquisition period, then changing to a fine discretization once parameter acquisition was achieved [28,29].

II.3.1. The Weighted Average. The outputs of each elemental filter, $\hat{x}_j(t_i^+)$, are put through a weighted average in the same manner as in Equation (II-6), except that instead of summing over the full set of K filters, only those implemented in the moving bank are summed. Thus if J

filters are implemented, Equation (II-6) becomes:

$$\hat{\underline{x}}(t_i^+) = \sum_{j=1}^J \hat{\underline{x}}_j(t_i^+) p_j(t_i) \quad (\text{II-24})$$

Similarly, Equation (II-5) describing the $p_j(t_i)$'s becomes:

$$p_j(t_i) = \frac{f_j(\underline{z}(t_i)) p_j(t_{i-1})}{\sum_{k=1}^J f_k(\underline{z}(t_i)) p_k(t_{i-1})} \quad (\text{II-25})$$

where, as before:

$$f_j(\underline{z}(t_i)) = \frac{1}{(2\pi)^{m/2} |A_j(t_i)|^{(1/2)}} \exp[-(1/2) \underline{z}_j^T(t_i) A_j^{-1}(t_i) \underline{z}_j(t_i)]$$

and

$$A_j(t_i) = H_j P_j(t_i^-) H_j^T + R_j$$

$$\underline{z}_j(t_i) = \underline{z}_i^- - H_j \hat{\underline{x}}_j(t_i^-)$$

m is the dimension of \underline{z} (number of measurements)

R_j is the measurement noise strength

II.3.2 Sliding the Moving Bank. When the true parameter vector is associated with a point within the moving bank, the moving bank operates essentially like a smaller version of the full-bank estimator. However, when the true parameter point lies outside the moving bank, this condition must be detected and some action must be taken to bring it within the moving bank or the estimate will be erroneous [28:22,29:10]. Since the true parameter point is both unknown and uncertain, some means detecting when it is not within the moving bank must be determined. Additionally, the estimator can be expected to operate best when the moving bank is centered on the true parameter point, therefore even if the true parameter point is within the bank but is close to the perimeter, the bank should be moved. Maybeck and Hentz [28:22-24,29:10:12] investigated four means of detecting when the true parameter point was not within the moving bank.

II.3.2.1 Residual Monitoring. Let a likelihood quotient $L_j(t_i)$ be defined of the quadratic form appearing in Equation (II-5):

$$L_j(t_i) = \underline{x}_j^T(t_i) A_j^{-1}(t_i) \underline{x}_j(t_i) \quad (\text{II-26})$$

When the true parameter point is outside the moving bank, all of the likelihood quotients for the elemental filters within the moving bank can be expected to exceed some threshold level T , the numerical value of which is set in an

ad hoc manner during performance evaluations. This detection method would indicate that the moving bank should be moved at time t_i if:

$$\min\{L_1(t_i), L_2(t_i), \dots, L_J(t_i)\} \geq T \quad (\text{II-27})$$

The bank should be moved in the direction of the filter with the smallest L_j , as that filter can be expected to be nearest to the true parameter point. This method of detection responds effectively and quickly to a real need to move the bank but is also apt to respond erroneously to a single instance of large measurement corruption noise.

II.3.2.2. Parameter Position Estimate Monitoring.

The current estimate of the parameter vector is adapted from Equation (II-8):

$$\hat{\underline{a}}(t_i) = \sum_{j=1}^J \underline{a}_j p_j(t_i) \quad (\text{II-28})$$

Using this logic would require a move anytime the bank was not centered on the point closest to the estimated parameter point. Since $\hat{\underline{a}}(t_i)$ depends on a history of measurements, this method of detection is not as sensitive to a single instance of large measurement noise as is residual monitoring.

II.3.2.3. Parameter Position and Velocity Estimate Monitoring. The history of parameter position estimates can be used to generate an estimate of the velocity of the true parameter point in the case where the parameters are varying slowly and steadily in time. In this case, the bank is moved after the current update but before the next propagation cycle, to center it upon the point closest to the projected parameter point. In Maybeck and Hentz's investigations, this method of detection was found to perform worse than did parameter position estimate monitoring or probability monitoring described below [28:85,29:23].

II.3.2.4 Probability Monitoring. The conditional hypothesis probabilities $p_j(t_i)$ computed via Equation (II-25) are monitored, and if the largest $p_j(t_i)$ is larger than a chosen threshold, the bank is centered on that filter. Maybeck and Hentz [28:85,29:23] found that this method of detection, when used by itself, provided performance as good as parameter position monitoring and required less computation.

II.3.3. Changing the Discretization The filters in the moving bank need not be those associated with adjacent points in the parameter space. As seen in Figure II-2, it may be more appropriate to space the filters implemented in the moving bank widely over the parameter space. This can be expected to decrease the accuracy of the estimate, but the probability that the true parameter point will lie

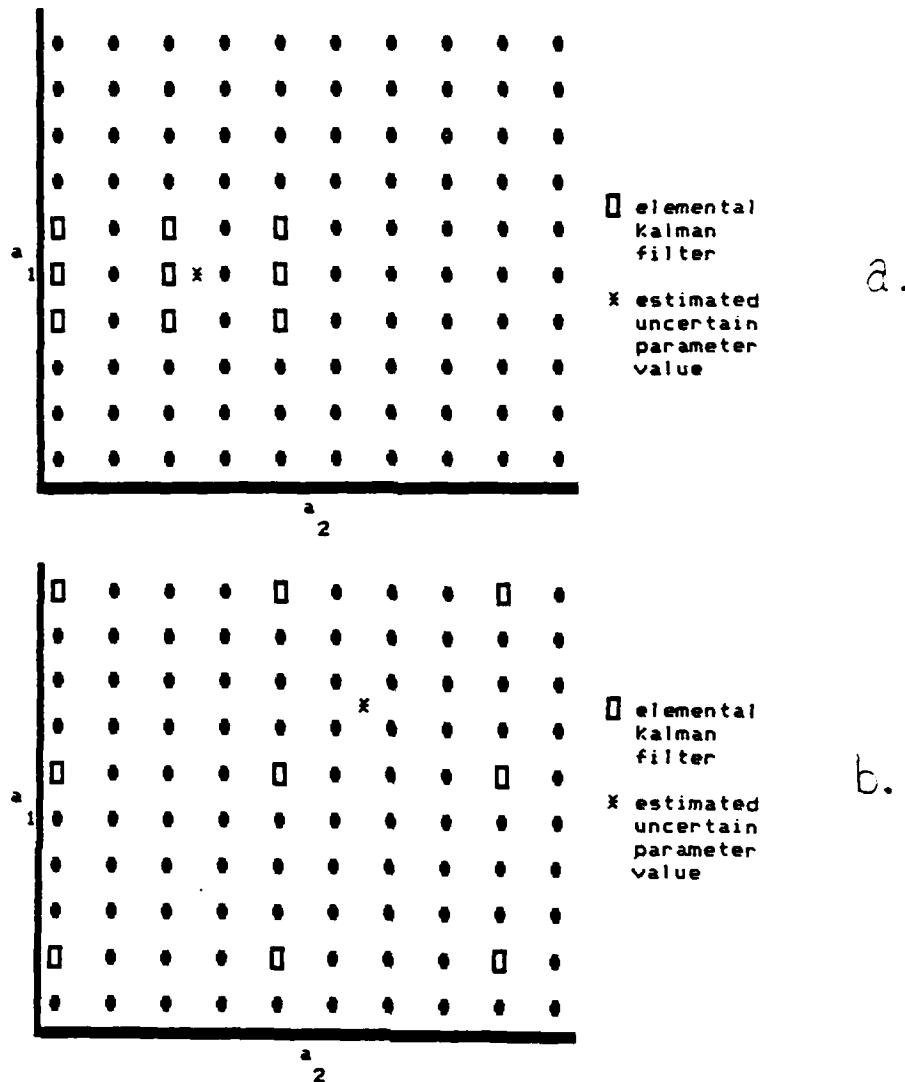


Figure II-2. Bank Discretizations: a. fine, b. coarse.

within moving bank can be expected to increase. Maybeck and Hentz [28:87,29:25] found that parameter acquisition was enhanced in some cases by starting the moving bank with a discretization coarse enough to cover the entire parameter space, then contracting it to a finer discretization when a scalar distance measure associated with the covariance of

the parameter estimate:

$$E\{[\underline{a} - \hat{\underline{a}}(t_i)][\underline{a} - \hat{\underline{a}}(t_i)]^T | \underline{Z}(t_i) = \underline{Z}_i\} = \sum_{j=1}^J [\underline{a}_j - \hat{\underline{a}}(t_i)][\underline{a}_j - \hat{\underline{a}}(t_i)]^T p_j(t_i) \quad (\text{II-29})$$

drops below some selected threshold. The contraction from the coarsest discretization to the finest discretization can be done either in one step or several.

The bank can also be expanded during normal operation. This would be appropriate if the true parameter point underwent a large jump change in position (due perhaps to some large and sudden physical change in the system). Such a condition can be detected by the residual monitoring discussed earlier [28:84,29:20]; in this case all of the L_j 's would be expected to be large and close to each other in value. Probability monitoring may also give some indication of this condition by the p_j 's all becoming close to each other in value. After the bank is expanded, it is then allowed to contract subsequently about the new parameter point, in the manner discussed earlier.

II.3.4 Initialization. When the moving bank is moved, expanded or contracted, it is necessary to initialize any filters that were not implemented in the moving bank before the action took place. Each new filter must be assigned values for Φ_j , B_{dj} , K_j , H_j , G_{dj} , $\hat{\underline{x}}_j(t_i)$, and $p_j(t_i)$ [28:28,29:13]. All of these except $\hat{\underline{x}}_j$ and p_j

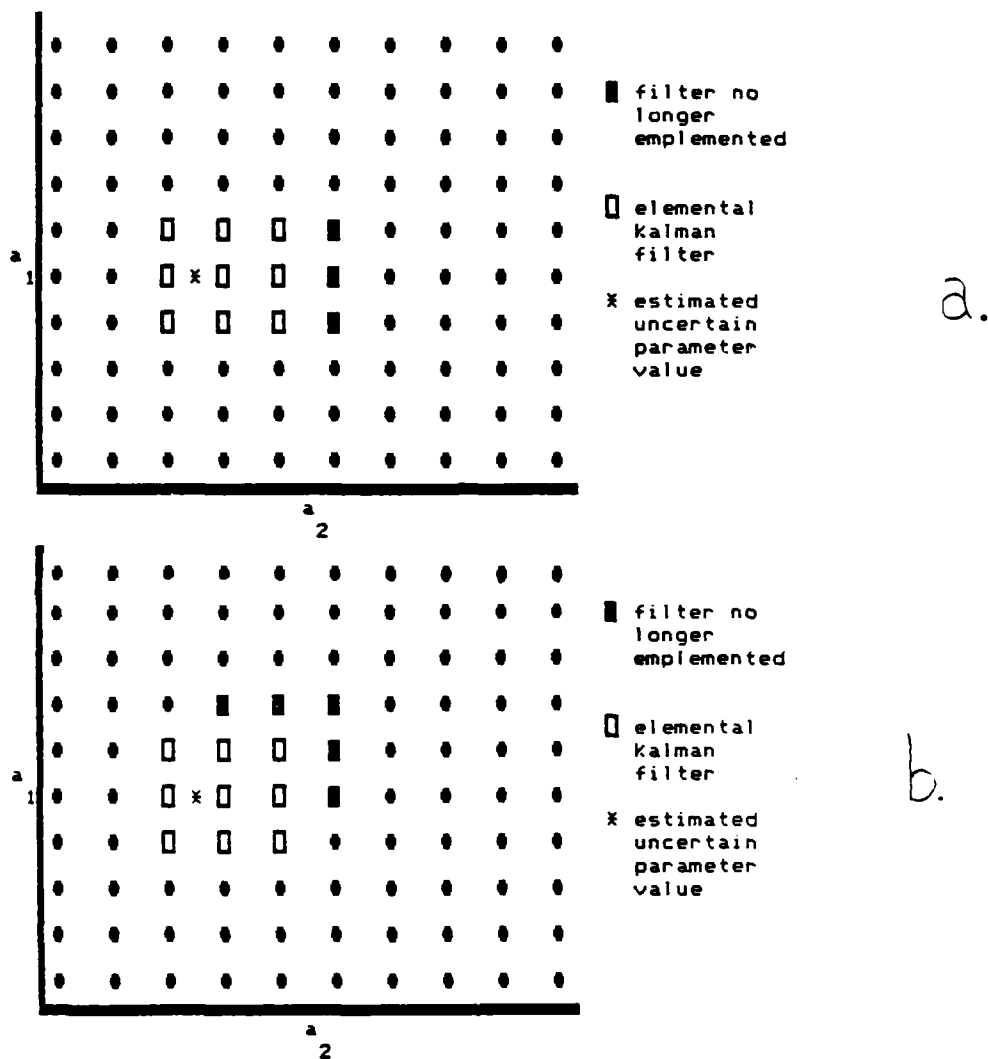


Figure II-3. Bank Move: a. horizontal, b. diagonal.

are determined by the point in the parameter space the filter occupies.

An appropriate choice for the $\hat{x}_j(t_i)$'s is the current moving-bank estimate of the system states $\hat{x}(t_i^+)$. The new values for the $p_j(t_i)$'s will be dependent on the number of new filters. For a move this will be some fraction of the total number of filters in the moving bank; as seen in Figure II-3, for the case of nine filters in the moving

bank, it would either be three or five filters. Good performance may be achieved by dividing the total probability weight of one minus the sum of the unreset p_j 's among the new filters. This can either be divided equally, or it can be apportioned in a manner indicating the new filters' expected correctness as described in Equation (II-30) [28:29,29:13]:

$$p_{jch}(t_i) = \frac{f_j(\underline{z}(t_i))(1 - \sum_{unch} p_k(t_i))}{\sum_{ch} f_k(\underline{z}(t_i))_j} \quad (II-30)$$

where ch = changed, unch = unchanged, and:

$$f_j(\underline{z}(t_i)) = \frac{1}{(2\pi)^{m/2} |A_j(t_i)|^{(1/2)}} \exp[-(1/2) \hat{\underline{x}}_j^T(t_i) A_j^{-1}(t_i) \hat{\underline{x}}_j(t_i)]$$

$A_j(t_i)$, m , and R are as defined before, and:

$$\hat{\underline{x}}_j(t_i) = \underline{z}_i - R_j \hat{\underline{x}}_j(t_i^+)$$

This may require enough additional computation time that there is no net performance improvement over dividing the net probability weight evenly among the changed filters. For an expansion or contraction, it is likely that all of the filters are changed or that the old $p_j(t_i)$'s are no longer valid, in which case setting all the $p_j(t_i)$'s to $1/J$ is appropriate.

II.3.5. Moving-Bank Adaptive Control. The controller for use with the moving-bank multiple model adaptive estimator can be designed following any of the three methods discussed for the full-bank estimator in Section II.2.3. The only difference is that for the moving bank case, the multiple model adaptive controller only uses the controllers associated with the filters currently implemented in the moving bank. Maybeck and Hentz received good and essentially identical results with a single changeable-gain controller and a moving-bank multiple model adaptive controller. The single fixed-gain controller performed poorly when the true parameter vector differed significantly from the nominal parameter vector for which the controller was designed. In addition, performance was sensitive not only to the magnitude of the error made in estimating the parameter vector, but also was more sensitive to errors in one parameter than the other, and whether the parameter was overestimated or underestimated [28:104-105, 29:25-27].

Maybeck and Hentz also found it necessary to shut off control for the initial period during which the moving bank was acquiring the parameter estimate. If control was applied before the parameters were identified, the wrong control was often applied, driving the system unstable [28:104, 29:25]. Turning off the control was also necessary for the same reason after a jump change in the true parameter point. This led to the recommendation that an appropriate method of deciding when to enable control be

investigated, based on a determination that parameter acquisition had taken place. The possibility of using the fixed gain controller based on a nominal parameter vector during the acquisition phase, rather than simply disabling control entirely was also suggested [28:106].

II.4 Ambiguity Functions Analysis

A tool that can be used to predict the ability of the moving-bank multiple model adaptive estimator to center itself correctly on the filter in the bank which is closest to the true parameter point, is ambiguity function analysis [5:97]. The generalized ambiguity function is given by:

$$A_i(\underline{a}, \underline{a}_t) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} L[\underline{a}, \underline{Z}_i] f_{\underline{Z}(t_j) | \underline{a}(t_j)}(\underline{Z}_i | \underline{a}_t) d\underline{Z}_i \quad (\text{II-31})$$

where \underline{a} is the parameter vector, \underline{a}_t is the true parameter vector, and $L[\underline{a}, \underline{Z}_i]$ is a likelihood function [5:97-99]. For a given value of \underline{a}_t , this function of \underline{a} yields information about the expected ability of the adaptive filter to estimate parameters. When plotted on a three dimensional surface, over the plane of the two-dimensional parameter space, the curvature of the ambiguity function at \underline{a}_t predicts the precision with which the adaptive estimator can estimate the parameters. The moving bank will center itself on the highest local peak within its area; if there is no local peak, the moving bank will move uphill until it encounters either a local peak or the edge of the parameter

space. If there are multiple peaks in the parameter space, the moving bank may converge to incorrect parameter values, depending on the bank starting position. In addition, the greater the curvature at the peak, the greater is the precision [28:332]. Examination of the ambiguity functions may be useful in determining starting points for the moving bank, appropriate discretization levels, and contraction and expansion strategies. It is important to note that the ambiguity function varies with \underline{a}_t ; thus a different plot is required for each true parameter point of interest.

The curvature of the ambiguity function is inversely related to the Cramér Rao lower bound on the estimate error covariance matrix by [5:97]:

$$E\{[\underline{a} - \underline{a}_t][\underline{a} - \underline{a}_t]^T\} \geq [-(\partial^2/\partial \underline{a}^2) A_i(\underline{a}, \underline{a}_t)|_{\underline{a}=\underline{a}_t}]^{-1} \quad (\text{II-32})$$

The ambiguity function (Equation (II-31)) can be calculated through the evaluation of covariance analysis results in which the true system is based on \underline{a}_t and the estimator is a single Kalman filter of the same structure as the true system, but based on \underline{a} instead of \underline{a}_t . The ambiguity function can thus be written [5:98, 28:333]:

$$\begin{aligned} A_i(\underline{a}, \underline{a}_t) &= m/2 \ln(2\pi) - 1/2 \ln[|A(t_i; \underline{a})|] \\ &\quad - 1/2 \text{tr}\{A^{-1}(t_i; \underline{a})[H(t_i)P_e(t_i^-; \underline{a}_t, \underline{a})H^T(t_i) + R(t_i)]\} \\ &\quad - n/2 \ln(2\pi) - 1/2 \ln[|P(t_i^+; \underline{a})|] \\ &\quad - 1/2 \text{tr}\{P^{-1}(t_i^+; \underline{a})P_e(t_i^+; \underline{a}_t, \underline{a})\} \end{aligned} \quad (\text{II-33})$$

where

$$A(t_i; \underline{a}) = [H(t_i)P(t_i^-; \underline{a})H^T(t_i) + R(t_i)]^{-1}$$

for the Kalman filter based on \underline{a}

$P_e(t_i^+; \underline{a}_t, \underline{a})$ is the covariance matrix of the error between the state estimates of the Kalman filter based on \underline{a} and the states of the true system based on \underline{a}_t , where '-' or '+' denotes before or after incorporation of the i^{th} measurement.

'm' is the number of measurements.

and 'n' is the number of states.

The first three terms are in actuality summed over the last N sample times; here N is set equal to one, which reduces the size of the fluctuations in the value of $A_i(\underline{a}, \underline{a}_t)$ (flattens the plotted surface). This does not aid the analysis of the plots, but does make it computationally simpler.

II.5 Summary

The algorithms for full-scale MMAE and the moving-bank multiple model adaptive estimator have been developed in this chapter. Both estimators are expected to give accurate adaptive estimates of true system states in many applications; however, the moving-bank estimator is expected to be a more practical estimator for implementation because of the reduced computational loading.

Several areas specific to the moving-bank estimator were explored. Four means of detecting when movement of the moving bank is required were discussed:

- a. residual monitoring
- b. parameter position estimate monitoring
- c. parameter position and velocity estimate monitoring
- d. probability monitoring

Changing the discretization of the moving bank was discussed both as an approach to initial acquisition of the unknown parameter vector and as means for reacquisition after a jump change in the unknown parameter vector. Parameter estimate covariance monitoring and probability monitoring were discussed as means of detecting when the bank should be contracted. Residual monitoring and probability monitoring were discussed as means of detecting when the bank should be expanded.

When moving, expanding, or contracting the moving bank, it is necessary to reset the state estimates and probability weightings of the new filters in the bank. The current overall state estimate is appropriate for resetting the individual state estimates. After the bank has been expanded or contracted, it is appropriate to reset the probability weightings of all of the filters to $1/J$. After the bank has been moved, only the probability weightings of the new filters should be reset by dividing among them the total probability weight left after subtracting the

probability weightings of the unchanged filters from the total weight of one. The remaining weight can be divided either evenly or in a manner reflecting the estimated correctness of the new filter.

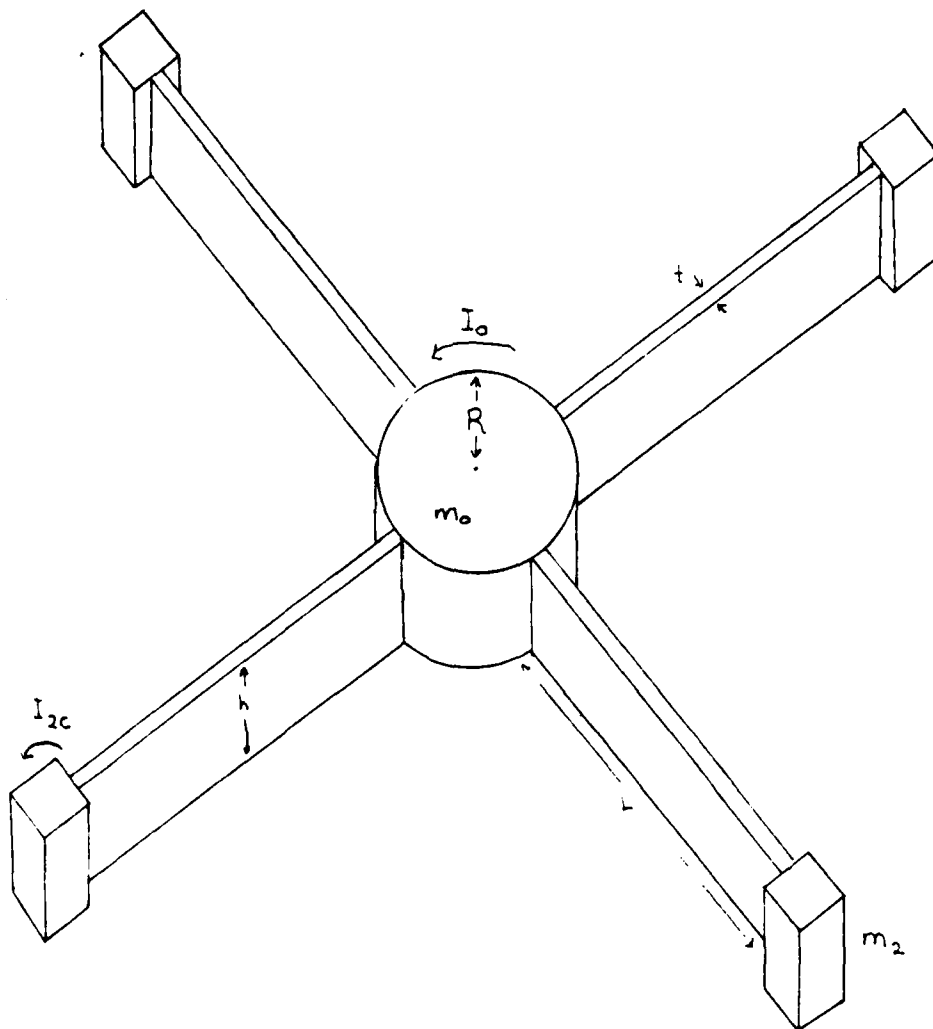
This research is intended to explore and expand these movement, expansion, and contraction decision making and reset methods, as applied to the control of the bending modes in a realistic model of a large flexible space structure. The model of the flexible space structure is developed in the following chapter.

III. SPACE STRUCTURE MODEL

III.1. Introduction

This chapter describes the system equations for the 'Draper Laboratory / Rocket Propulsion Laboratory (RPL) Configuration' model of a large flexible space structure, illustrated in Figure III-1. The structure consists of a rigid central hub with four structurally identical flexible appendages cantilevered radially from the hub. Experimental work concerning the planar rotational/ vibrational dynamics of this U.S.A.F. Rocket Propulsion Laboratory demonstration model is being conducted at the Charles Stark Draper Laboratory, where a physical model of the structure is supported on an air bearing table [31].

The differential equations describing the model are developed from the equations of motion for the unforced system by Muckenthaler [32]; the expressions for the kinetic and potential energy are developed and discretized, using the assumed modes method with terms of higher order than two ignored. This allows the mass and stiffness matrices to be identified. The integral equations for the mass and stiffness matrices are solved symbolically and a closed form expression for them is achieved. The second order differential equations are then placed in the standard state space form for a stochastic system model. This allows the construction of the elemental Kalman filters used in this thesis for the investigation of moving-bank MMAE.



R : hub radius

L : arm length

t : arm thickness

h : arm height

m_0 : mass of the hub

I_{2c} : rotational moment
of the tip mass

I_0 : rotational moment
of the hub

m_2 : mass of the tip mass

Figure III-1. Draper/RPL Configuration Model.

III.2. Second Order Form

As described by Muckenthaler [32] the motion of the uncontrolled Draper/RPL configuration model can be described by the second order vector differential equation [32:25]

$$M \ddot{\underline{q}} + C \dot{\underline{q}} + K \underline{q} = \underline{0} \quad (\text{III-1})$$

where:

M: is the $N \times N$ symmetric and positive definite mass matrix

C: is the $N \times N$ structural damping matrix

K: is the $N \times N$ symmetric positive semi-definite stiffness matrix

q: is the N dimension vector of generalized coordinates

$$\underline{q} = [\theta, U_1 \dots U_p, V_1 \dots V_p]^T$$

where θ is the rotational displacement of the hub, and U_i and V_i are the generalized displacement coordinates of the arms, and $N = 2p + 1$, where p is the number of modes of interest.

III.2.1. Assumptions. The equations developed to describe the Draper/RPL configuration model are based on the following assumptions [32:11]:

a) The longitudinal and out-of-plane vibrations of the arms are of much higher frequency than the transverse vibrations and are negligible.

b) Anti-symmetric deformations are such that the deflections of the first arm are equal in magnitude but opposite in direction to the second arm, and the deflections

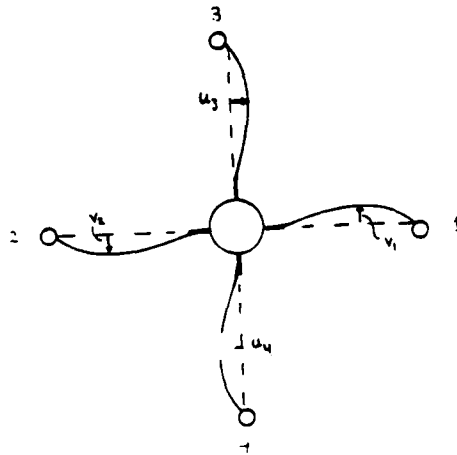


Figure III-2. Anti-Symmetric Deflections [32:11].

of the third arm are equal and in magnitude but opposite in direction to the fourth arm; see Figure III-2.

c) Since the hub of the physical model is supported by an air bearing table, the force of gravity can be neglected in the derivation of potential energy.

d) The structural damping coefficients are very small; therefore the structural damping matrix C is approximated as the zero matrix.

e) All four arms are structurally identical with:

$$L_1 = L_2 = L_3 = L_4 = L$$

$$t_1 = t_2 = t_3 = t_4 = t$$

$$h_1 = h_2 = h_3 = h_4 = h$$

and the four tip masses are identical and equal to m_2 .

III.2.2. Mass and Stiffness Equations. The mass and stiffness matrices can be found by solving for the kinetic and potential energy of the free vibration system. Neglecting the terms of order three or higher, the kinetic and potential energies can be written in the form [32:23]:

$$T = 1/2 \dot{q}^T M \dot{q}$$

$$V = 1/2 q^T K q$$

where the elements of the mass matrix are given for

$i = 1, \dots, p$; and $j = 1, \dots, p$; by:

$$M(1,1) = 2 (I_c + I_{2c})$$

$$M(1, i+1) = 2\rho th \int_R^{R+L} x \theta_i dx + 2m_2(R+L)\theta_i(z=x-R=L) + 2I_{2c}\theta'_i(z=x-R=L)$$

$$M(i, i+p+1) = 2\rho th \int_R^{R+L} y \theta_i dy + 2m_2(R+L)\theta_i(z=y-R=L) + 2I_{2c}\theta'_i(z=y-R=L)$$

$$M(i+1, 1) = M(1, i+1)$$

$$M(i+p+1, 1) = M(1, i+p+1)$$

$$M(i+1, j+1) = 2\rho th \int_R^{R+L} \theta_i \theta_j dx + 2m_2\theta_i(z=x-R=L)\theta_j(z=x-R=L) + 2I_{2c}\theta'_i(z=x-R=L)\theta'_j(z=x-R=L)$$

$$M(i+p+1, j+p+1) = 2\rho th \int_R^{R+L} \theta_i \theta_j dy$$

$$+ 2m_2 \theta_i(z=y-R=L) \theta_j(z=y-R=L)$$

$$+ 2I_{2c} \theta'_i(z=y-R=L) \theta'_j(z=y-R=L)$$

$$M(i+1, j+p+1) = 0$$

$$M(i+1+p, i+1) = 0$$

where θ_i is a function describing the bending modes of the arms, defined as:

$$\theta_i(z) = 1 - \cos(i\pi z/L) + (1/2)(-1)^{i+1}(i\pi z/L)^2$$

$$z = x - R \text{ (or } y - R)$$

and θ'_i is the first derivative of θ_i with respect to z .

The physical dimensions of the model t , h , m_2 , I_{2c} , I_{00} , R , and L , are as shown in Figure III-1. In addition ρ is the mass density of the arms, and:

$$I_c = (1/2)I_0 + 2I_1 + 2m_2(R+L)^2$$

$$I_1 = \rho th \int_R^{R+L} x^2 dx$$

The elements of the stiffness matrix are given for

$i = 1, \dots, p$; and $j = 1, \dots, p$; by:

$$K(1,1) = 0$$

$$K(i+1,1) = 0$$

$$K(i+1+p,1) = 0$$

$$K(1, j+1) = 0$$

$$K(1, j+p+1) = 0$$

$$K(i+1, j+1) = 2EI \int_0^L \theta_i''(z) \theta_j''(z) dz$$

$$K(i+1+p, j+1+p) = K(i+1, j+1)$$

$$K(i+1+p, j+1) = 0$$

$$K(i+1, j+1+p) = 0$$

where E is the Modulus of Elasticity of the arms, and I is the area moment of inertia based only on t and h as:

$$I = (1/12)ht^3$$

and θ''_i is the second derivative of θ_i with respect to z.

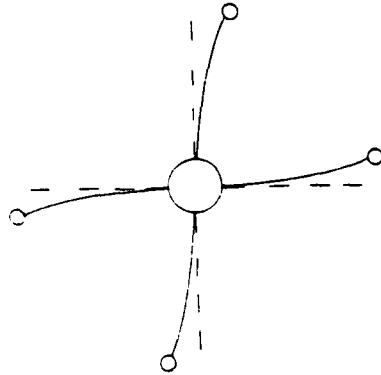
The equations for the elements of the M and K matrices are simplified and evaluated in Appendix A.

III.2.3. Eigenvalues of the free system. The eigenvalues of the free system can be found by solving the generalized eigenvalue equation:

$$K\underline{x} = \lambda M\underline{x} \quad (\text{III-2})$$

The square roots of the eigenvalues represent the vibrational frequencies of the modes depicting the relative motion of the free arms. The modes appear in pairs, very close in frequency, depicting what Junkins [31] calls 'unison' and 'opposition' modes. The opposition modes are simple cantilever beam modes characterized by the adjacent beams moving in opposition. He states the unison modes are perturbed cantilever modes, with just slightly higher frequencies, with all four beams moving in unison and the

Opposition Mode



Unison Mode

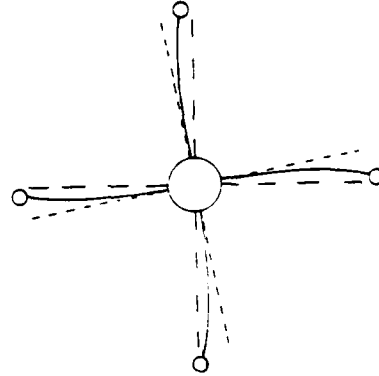


Figure III-3. Opposition and Unison Modes [32:26].

hub having non-zero rotation. The hub rotation is necessary to counter the unison movement of the arms and tip masses to conserve angular momentum in the system [32:26]; see Figure III-3.

III.3 State Space Form

The second order differential equation describing the free system, Equation (III-1), can be modified to describe the controlled system as:

$$M \ddot{\underline{q}} + K \underline{q} = B \underline{u} \quad (\text{III-3})$$

where B is an $N \times M$ control matrix and \underline{u} is an M -dimensional control vector added to the original Equation (III-1).

Recall that structural damping was assumed to be very small so the damping matrix C has been neglected.

If the generalized coordinate vector \underline{q} is augmented with its derivative:

$$\underline{\Omega} = [\underline{q}^T, \dot{\underline{q}}^T]^T$$

then Equation (III-3) can be rewritten as:

$$\underline{\Omega} = \begin{bmatrix} 0 & I \\ -M^{-1}K & 0 \end{bmatrix} \underline{\Omega} + \begin{bmatrix} 0 \\ -M^{-1}B \end{bmatrix} \underline{u} \quad (\text{III-4})$$

in first order form.

On the Draper/RPL configuration model, actuators are located one on the hub and one halfway along each arm. These apply a torque u_1 to the hub, a torque u_2 at $z=L/2$ to arms 1 and 2, and a torque u_3 at $z=L/2$ to arms 3 and 4. Recall Figure III-2, in which the arms are numbered. Matrix B then becomes [32:30]:

$$B = \begin{bmatrix} 1 & 2 & 2 \\ 0 & 2\underline{q}'_i(z=L/2) & 0 \\ 0 & 0 & 2\underline{q}'_i(z=L/2) \end{bmatrix}$$

where $\underline{q}'_i(z)$ is of dimension p (the number of modes of interest). B is therefore an $N \times 3$ matrix.

Position and velocity measurements of the model are available from collocated position and velocity sensors on the hub and at positions along each arm. For the state space model, the measurement equation (assuming for the moment noise-free measurements) is thus:

$$\underline{z} = \begin{bmatrix} H & 0 \\ 0 & H \end{bmatrix} \underline{\Omega} \quad (\text{III-5})$$

where the measurement matrix H is [32:30]:

$$H = \begin{bmatrix} 1 & \underline{0}^T & \underline{0}^T \\ 0 & \underline{g}_i^T(z=L/2) & \underline{0}^T \\ 0 & \underline{g}_i^T(z=L) & \underline{0}^T \\ 0 & \underline{0}^T & \underline{g}_i^T(z=L/2) \\ 0 & \underline{0}^T & \underline{g}_i^T(z=L) \end{bmatrix}$$

1 column p columns p columns

H is thus a 5 x N matrix.

III.4. Stochastic Form

The mathematical model can be placed in stochastic form by adding a noise matrix G_w , multiplying the control input noise \underline{w} , to Equation (III-4), and a measurement noise vector to Equation (III-5). If it is assumed that input noise will enter through the actuators and measurement noise through the sensors, then G_w becomes identical to the augmented B matrix. The system is thus described by:

$$\dot{\underline{\Omega}} = \begin{bmatrix} 0 & I \\ -M^{-1}K & 0 \end{bmatrix} \underline{\Omega} + \begin{bmatrix} 0 \\ -M^{-1}B \end{bmatrix} \underline{u} + \begin{bmatrix} 0 \\ -M^{-1}B \end{bmatrix} \underline{w} \quad (\text{III-6})$$

and

$$\underline{z} = \begin{bmatrix} H & 0 \\ 0 & H \end{bmatrix} \underline{\Omega} + \underline{v} \quad (\text{III-7})$$

III.5 Physical System Constants

The nominal constants describing the physical system are listed in Table III-1. For the purpose of setting up a

parameter space in which to test moving-bank MMAE algorithms, ρ , the mass density of the arms, and E , the modulus of elasticity for the arms, were allowed to vary - 20% to +16% from their nominal values in discrete steps of 4%, yeilding a 10 by 10 point parameter space. This is an unrealistically large amount of variation in these parameters, but it proved to be necessary for the purpose of investigating moving-bank MMAE algorithms as applied to this system.

Table III-1. Configuration Constants for Draper/RPL Model

R: hub radius	=	1 ft
I_0 : hub rotary inertia	=	8 slug-ft ²
ρ : mass density of the arms	=	5.22 slug-ft ³
E: arm modulus of elasticity	=	1.584E09 lb/ft ²
t: arm thickness	=	0.0104166 ft
h: arm height	=	0.5 ft
L: arm length	=	4.0 ft
m_2 : tip mass	=	0.156941 slug
I_{2c} : tip mass rotary inetria	=	0.0018 slug-ft ²

III.6. State Reduction

If p is taken to be two (two opposition and two unison modes) then N is five, and this results in a ten-state model for the augmented system. A ten-state model is needlessly more complex than is required for this thesis. If the hypothetical purpose of this estimator/controller is to eliminate uncommanded torque to the hub of the spacecraft (as might be required if it is desired to point an apparatus on the hub in a specified direction), then it is only

necessary to control those bending modes which produce a torque on the hub.

As discussed earlier, only the unison modes produce a torque on the hub [32:26]. Junkins [31] determined that the unison modes were the even numbered modes (i.e. 2,4,6,...). If the opposition modes are assumed to be stable the matrices describing the system can be simplified by eliminating those state variables depicting the positions and velocities of the opposition modes. The state vector is then reduced to six states and the matrices describing the system need to be simplified by eliminating the rows and columns pertaining to the opposition modes. Thus for the F matrix, every other row and column is eliminated, and the result is a six by six matrix. For the augmented B and G_w matrices, every other row is eliminated making them six by three matrices. For the augmented H matrix, every other column is eliminated, leaving a ten by six matrix. Sample matrices for a nominal case are shown in Appendix A.

III.7. Summary

This chapter developed equations describing a realistic model of a large flexible space structure. The stochastic state space mathematical model derived is dependent on the physical parameters describing the real-world system. Variation in the physical parameters will create variation in the stochastic model, and this will allow investigation into the use of moving-bank MMAE to estimate both the system states and the varying parameters of the physical system.

IV. Simulation

IV.1. Introduction

In order to evaluate the performance of the moving-bank multiple model adaptive estimator/controller, it is necessary to simulate the space structure and the estimator/controller operation. The computer simulation used provides a Monte Carlo analysis of both the space structure model and the estimator/controller. This chapter contains a brief discussion of Monte Carlo analysis, an outline of the computer software used to accomplish the analysis, and the plan for analyzing the performance of the moving-bank algorithm and the specific logics used for the move, contract, and expand decisions.

IV.2. Monte Carlo Analysis

The random nature of the input and measurement noise processes makes it impossible to select a single typical example of them. Thus in order to characterize the performance of the moving-bank multiple model adaptive estimator/controller statistically, it is necessary to examine the ensemble average of the estimator/controller performance using many samples of the error process. Monte Carlo analysis does precisely this: a number of individual time simulations of the estimator/controller are made and sample statistics (means and covariances) are computed directly for each sample time [33:29].

For the analysis performed in this thesis, the true system model is described by the linear time-invariant stochastic difference equation:

$$\underline{\tilde{x}}(t_i) = \Phi \underline{\tilde{x}}(t_{i-1}) + B_d \underline{\tilde{u}}(t_{i-1}) + G_d \underline{\tilde{w}}(t_{i-1}) \quad (\text{IV-1})$$

where

Φ is the state transition matrix from t_{i-1} to t_i

B_d is the control input matrix

G_d is the noise input matrix

Recall that, for the Draper/RPL model as shown in Equation (III-6), the noise input matrix is identical to the control input matrix, therefore:

$$B_d = G_d$$

where B_d is the discrete-time equivalent of the augmented B matrix of Equation (III-6), given by [33:171]:

$$B_d = \int_{t_{i-1}}^{t_i} \Phi(t_i, \tau) B \, d\tau$$

and $\underline{\tilde{x}}(t_i)$ is the discrete time equivalent of the augmented state vector $\underline{\tilde{Q}}$ of Equation (III-6). Also note that Φ , and B_d (and therefore G_d) are all functions of the true parameter vector \underline{a}_t , where

$$\underline{a}_t = \begin{bmatrix} P_t \\ E_t \end{bmatrix}$$

The model is driven by both the known control $\underline{\tilde{u}}(t_{i-1})$ and a discrete zero-mean white Gaussian noise $\underline{\tilde{w}}(t_{i-1})$ of covariance Q_d . Noise-corrupted measurements of the system

states are provided to the estimator in the form of:

$$\underline{z}(t_i) = H\underline{x}(t_i) + \underline{v}(t_i) \quad (\text{IV-2})$$

where H is now the augmented version of the measurement matrix shown in Equation (III-7) and the measurements are corrupted by a discrete-time zero-mean white Gaussian measurement noise $\underline{v}(t_i)$ of covariance R.

The true system and the estimator/controller are operated from time t_0 to time t_f for a sufficient number of runs that the computed sample means and covariances of the random variables of interest are good approximations to ensemble averages (expectations). It is possible to determine the number of runs that is sufficient by observing how the computed sample statistics change as the number of runs is increased; after a sufficient number of runs, the sample statistics will converge to a constant [33:29]. For this problem that entails on the order of 10 or more runs.

Figure IV-1 depicts the simulation of the true system, the estimator, and the controller. The variables of interest are:

the system ('truth model') states - $\underline{x}_t(t_i)$

the error in the estimate of the system states -

$$\underline{e}_x(t_i) = \underline{x}_t(t_i) - \hat{\underline{x}}(t_i)$$

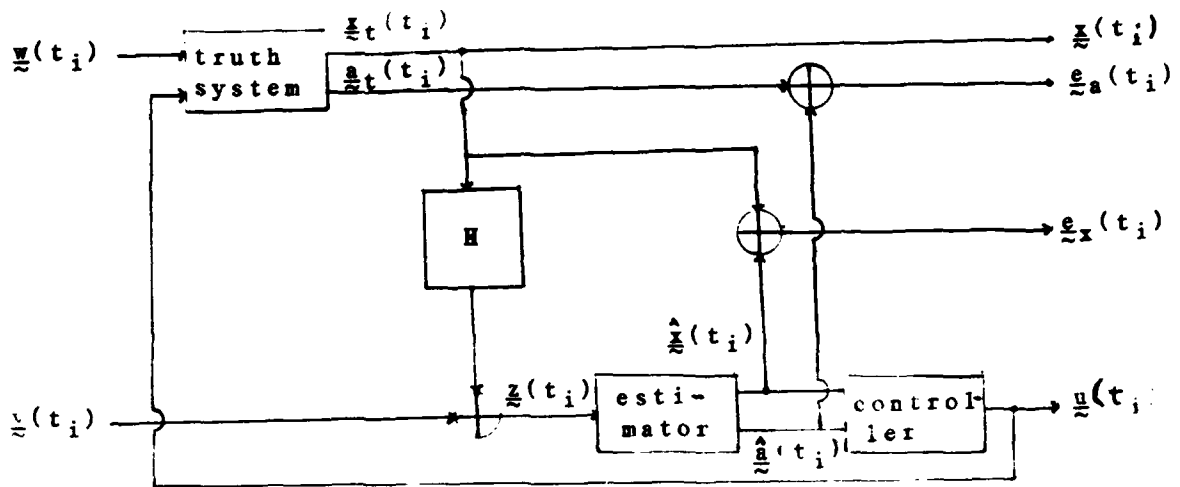


Figure IV-1. System, Estimator, and Controller Simulation.

the error in the parameter estimate -

$$\underline{e}_a(t_i) = \underline{a}_t(t_i) - \hat{\underline{a}}(t_i)$$

and the control input - $\underline{u}(t_i)$

The means and covariances of the variables of interest are all computed similarly; for example, the mean of the error in the state estimate is computed as:

$$E\{\underline{e}_x(t_i)\} \approx \hat{\underline{M}}_{e_x}(t_i) \triangleq (1/N) \sum_{k=1}^N \underline{e}_{xk}(t_i) \quad (\text{IV-1})$$

where N is the total number of runs performed in the Monte Carlo analysis, and $\underline{e}_{xk}(t_i)$ is the value of $\underline{e}_x(t_i)$ during the

k^{th} run. The covariance of $\underline{e}_x(t_i)$ is computed as:

$$E\{[\underline{e}_x(t_i) - E\{\underline{e}_x(t_i)\}][\underline{e}_x(t_i) - E\{\underline{e}_x(t_i)\}]^T\} \simeq P_{e_x}(t_i) \stackrel{\Delta}{=} \\ (1/(1-N)) \sum_{k=1}^N \underline{e}_{xk}(t_i) \underline{e}_{xk}^T(t_i) - (1/(1-N)) \hat{M}_{e_x}(t_i) \hat{M}_{e_x}^T(t_i) \quad (\text{IV-4})$$

When evaluating the estimator alone the controller block in Figure IV-1 is replaced by a dither signal which excites the system states, but is independent of the performance of the estimator. The dither signal is used based on the experience of Hentz [28:58], who found that system identifiability was enhanced by sufficiently and persistently exciting the true system modes with a known periodic input. Without the dither, the estimator had considerable difficulty in identifying the uncertain parameters, and little basis for evaluating various decision algorithms was found because the system would essentially reach a quiescent state regardless of the algorithm used [28:57-58]. The specific dither signal to be used (magnitude, frequency, etc.) is determined experimentally as a rigorous determination of the optimal input to enhance system identifiability is beyond the scope of this effort.

It is most appropriate to look at the statistics of the error in the state estimate and the error in the parameter estimate in evaluating the performance of the estimator. The error in the state estimate gives the best means of

comparing the estimator to other types of estimators, and the primary reason for adaptation is to enhance state estimation precision, rather than to provide accurate parameter estimates for their own sake. The characteristics of the error in the parameter estimate reveals the accuracy of the parameter estimates that may later be fed to the controller. The error in the parameter estimate can also give insight into how good a job of identifying the closest elemental filter the moving-bank is doing, thereby giving a means of evaluating the various move, contract, and expand algorithms; statistics on the location of the center of the bank are also valuable in this evaluation.

When evaluating the estimator/controller combination it is more appropriate to look at the statistics of the true state values and commanded controls. In this thesis, the object of the control input will be to drive the states to the quiescent state; therefore, deviations from zero are undesirable characteristics to be analyzed in evaluating controller performance. It may also be useful to examine the control input to detect unreasonable commanded control levels.

IV.3. Software Description

The Monte Carlo analysis of the moving-bank estimator/controller required the development of three computer programs (for a detailed discussion, see Appendix C.). The first program is a preprocessor which creates the parameter

space. The second program is a primary processor which performs the Monte Carlo simulation runs and generates the data describing each run. The third program is a postprocessor which computes the means and covariances of the variables of interest and generates plots based on their values. In addition, a program which computes the ambiguity functions and generates their plots was developed as a separate analytical tool; it requires a variation of the preprocessor to set up the parameter space values it uses.

The preprocessor sets up the parameter space. That is, for each realization of the uncertain parameter vector, it computes the matrices necessary to describe that parameter point uniquely in the equations used to describe the truth model and filters. This requires that, for the k^{th} parameter point, the ϕ_k , B_{dk} , H_k , K_k , $P_k(t_i^-)$, A_k^{-1} , and the G_k^* matrices and the determinant of A_k be computed. The preprocessor takes as input the ten discrete values each parameter can assume, and the weighting matrices W_x , and W_u which describe the quadratic cost function used in the design of the LQG controller gain (G^*) matrices. The preprocessor outputs a file containing all the matrices calculated, the weighting matrices, the strength of the input noise, and the covariance of the measurement noise. Unless the inputs change, the file containing the output can be used for all of the simulation runs, making it unnecessary to rerun the preprocessor.

The primary processor performs the Monte Carlo simulations. It consists of an executive routine which calls several subroutines. After input and initialization, the executive routine consists of an outer loop that counts the desired number of Monte Carlo runs, and an inner loop that performs the operations necessary for each sample period from the starting time of the simulation (t_0) to the ending time (t_f). For each sample period, separate routines are called to propagate the true system from the last sample time, propagate the filters currently implemented in the moving bank from the last sample time, take a noise-corrupted measurement of the true system, update the filters in the moving bank, calculate the control input, and finally make the decision whether to move, expand, or contract the bank. If it is decided to alter the bank, separate routines are called to perform the move, expansion, or contraction. After the computation for each sample period is complete, the variables of interest are written to a data file. Following each run, the outer loop reinitializes all necessary variables before beginning another run. Inputs to the primary processor are the file containing the description of the parameter space, true system parameters, a mode input which specifies the move/contract/expand algorithms to be used and associated thresholds, initial probability weightings for the filters in the moving bank, and initial filter states. The output of the primary processor is a data file for each variable of interest

covering all of the runs, and a more detailed print file covering just the first run which includes information describing the exact filters implemented in the moving bank and the variable affecting the decision algorithms. The print file allows detailed analysis of unexpected or unusual results, but avoids excessive output.

The post-processor takes a data file of a variable of interest and calculates the sample means and covariances for each sample time from t_0 to t_f . The post-processor then generates plots of time histories of the means of each variable of interest $\pm 1\sigma$, where σ is its standard deviation. Thus each data file generated by the primary processor requires a separate run of the post-processor; this provides for simplicity and flexibility in determining which variables to plot.

The simulation requires that the driving noise $\tilde{w}(t_{i-1})$ and the measurement noise $\tilde{v}(t_i)$ be zero-mean white Gaussian processes. Fortran provides a random number generator which can be used to approximate the required random variable's realizations at each time. If \tilde{y}_i is the random variable available directly from the random number generator, uniformly distributed between 0 and 1, then a zero-mean Gaussian random variable with a variance of 1 can be approximated by:

$$\tilde{r} = \sum_{i=1}^{12} \tilde{y}_i - 6 \quad (\text{IV-5})$$

In order to simulate a zero-mean Gaussian random vector with covariance Q_d , the following operation is performed:

$$\underline{w} = D \underline{x}$$

where the elements of \underline{x} are computed by independent calls to Equation (IV-1) for each scalar component, and where

$$D = \sqrt{Q_d}; \text{ i.e. } Q_d = DD^T$$

For the sake of simulation, the Cholesky decomposition is used to generate the square root [33:408].

IV.4. Simulation Plan

A systematic approach will be used to study the performance of the moving-bank estimator/controller. The performance analysis will be divided into two main parts. First the performance of the estimator alone, without feedback control (i.e., the controller block in Figure IV-1 is replaced with a dither signal independent of the state estimates), is evaluated. Analysis in this portion of the study will concentrate on qualitatively identifying the 'best' estimator configuration which will then be used for the analysis of the controllers to identify the 'best' adaptive estimator/controller combination.

The performance of the estimator alone is accomplished by driving the true system with a zero-mean white noise in combination with a dither signal for each of the move/contract/expand decision algorithms to be tested. For each decision algorithm being evaluated, the true

parameter vectors $\underline{a}_t(t_i)$ will be chosen which exercise that algorithm. The estimator will be evaluated first using only movement at the finest discretization level, and the issue of initializing the probability weightings of the new filters added to the bank will be explored. Then the effects of contraction from a coarse discretization to a fine discretization will be explored and contrasted to straight movement. If contraction proves beneficial, expansion to a coarse discretization following the detection of a jump change in the true parameter point will be investigated.

To evaluate the effect of computing probability weightings for new filters added to the bank during a move based on their expected 'correctness' (e.g., based on how far the new filters' \underline{a} values are from the current $\hat{\underline{a}}(t_i)$) vs. setting the probability weightings of the new filter to an equal share of the total probability of the filters removed, the system uncertain parameters will be constant over t_0 to t_f , and set equal to one of the discretized points in the parameter space different from the initial bank center. The effect of the probability weightings can then be seen in the speed with which the parameter estimate converges to the true system parameter point, and in the error in the state estimates. Movement primarily takes place when the true parameter point lies outside the area encompassed by the moving bank; it does not matter whether the true parameter point is exactly equal to

one of the discrete parameter points or not as long as it is initially outside the moving bank area; therefore, the true parameter point is set equal to one of the discrete parameter points for simplicity. The issue of how the moving bank tracks a slowly varying true parameter point is not investigated by this thesis due to time constraints. For the evaluation of simple bank movement, the bank size will begin at the finest discretization and no expansion or contraction will take place.

In order to evaluate the effect of contraction of the bank about the current parameter estimate, the true system parameters will be chosen the same as for the probability computation evaluation, and the performance of the estimator with bank contraction and movement can then be compared directly to the performance of the estimator using only movement to identify the uncertain parameter. For this portion of the evaluation, no expansion will be allowed to take place.

The evaluation of the expansion algorithm using residual monitoring will be conducted by using a jump change in the uncertain parameters from one discrete parameter point to another discrete parameter point at a time t_j where $t_0 < t_j < t_f$; and t_j is greater than t_0 by an amount sufficient to allow the estimator to converge to the first uncertain parameter point. The expansion algorithm's purpose is to allow the estimator to react more quickly to a jump change in the uncertain parameter vector's value (as if there had

been a catastrophic failure in one of the arms of the space structure model) than would be possible by allowing the estimator to converge to the new point by movement alone. Therefore, the results of the simulations using expansion will be compared to simulations using the same jump changes in parameter value where expansion is not allowed.

After the evaluation of the individual estimator algorithms, a composite 'best' estimator will be determined, made up of the move/contract/expand algorithms that performed the 'best'. The determination of 'best' will be based upon a tradeoff between added computational loading and faster state and parameter acquisition times, lower state and parameter estimate biases, and lower state and parameter estimate error variances. The goal in moving from full-scale MMAE to moving-bank MMAE is to obtain an estimator with similar performance but which has enough of a reduction in the computational loading required to make it practical for more applications. Therefore, any decision algorithm which increases computational loading is a step in the wrong direction unless this additional loading is outweighed by significant gains in performance.

When the best composite estimator has been determined, that composite estimator will be used to evaluate the controller configurations discussed in Chapter II:

- a. single fixed-gain controller
- b. single changeable-gain controller
- c. moving-bank multiple model adaptive controller

The estimator/controller combination will be evaluated using both constant true system uncertain parameter values and jump changes in the true system uncertain parameter values. In addition, an evaluation will be made of the effect of turning off control during the parameter acquisition phase vs. using the the adaptive controller from time t_0 , or a nominal fixed gain controller until parameter acquisition takes place then transferring to the adaptive controller. As in the estimator-only case, any increase in computational complexity must be offset by significant gains in performance (lower state biases and variances from the quiescent state) in order to justify the decrease in practicality associated with the increase in computational loading.

IV.5 Summary

This chapter has discussed the overall method of Monte Carlo analysis, the specific organization of the software used for the simulation, the plan and criteria for evaluation of the various estimation and control algorithms, and associated move, contract, and expand logics. It was noted that increases in the estimator/controller performance at the expense of increased computational loading must be significant to justify the decrease in practicality. The results of the simulations are presented in the next chapter.

RESULTS

V.1 Introduction

This chapter presents the results of the Monte Carlo simulations. The goals of the simulations were to evaluate the effectiveness of moving-bank multiple model adaptive estimation/control when applied to the realistic situation described in Chapter III, and specifically to evaluate the various move, contract, expand, and algorithms described in Chapter II. These goals were only partially met due to numerical difficulties which forced the use of approximations in computing probability weightings, and the ambiguity function. In addition the system being controlled proved not to require adaptive control, despite the uncertainties in the parameters describing it. This made the evaluation of the control algorithms impossible.

V.2. Numerical Problems

The covariance matrix P at time t_i^+ for all of the elemental filters proved to be numerically ill-conditioned (a function of there being a accurate data on some of the states from the measurements). Therefore it was impossible to compute the ambiguity function as described by Equation (II-33), as that equation requires the determinant of $P(t_i^+)$. Since the $P(t_i^+)$ matrix was used to compute $P(t_i^-)$, which is used in the computation of the A matrix used in Equations (II-7), (II-25), (II-30) and (II-33), more problems resulted, as the computed determinants of all of

the A matrices proved to be negative. Tuning of the filters could result in improving the numerical condition of $P(t_i^-)$, and could prevent the A matrices determinants from being negative. However; this would involve the use of less realistic noise strengths for Q and R, this could be worthwhile as an academic exercise but time constraints prevented that from being done for this thesis.

In order to overcome these numerical difficulties, the expressions for the probability weighting factors (Equation (II-25)), and the ambiguity function (Equation (II-33)) were approximated to remove the necessity to use the determinants of the $P(t_i^+)$ and the A matrices. In Equation (II-25) the density function is approximated as:

$$f_j(\underline{z}(t_i)) = \exp[-(1/2)\underline{z}_j^T(t_i)A_j^{-1}(t_i)\underline{z}_j(t_i)]$$

Note this is no longer a true density function because the scale factor is now incorrect; but because of the denominator in Equation (II-25) the probability weightings are still scaled correctly. This change is also reflected in Equation (II-30) describing how the probability weighting for a new filter in the bank can be computed based on its expected correctness. This approximation is reasonable as long as the A matrices of the the elemental filters are close enough to each other so that their determinants, in the absence of numerical problems, would be expected to be of approximately equal magnitudes. The ambiguity function described in Equation (II-33) was approximated by removing

the terms containing the determinants of $P(t_i^+)$ and A ; it then became:

$$\begin{aligned}
 A_i(\underline{a}, \underline{a}_t) \approx & m/2 \ln(2\pi) - n/2 \ln(2\pi) \\
 & - 1/2 \operatorname{tr}\{A^{-1}(t_i; \underline{a})[H(t_i)P_e(t_i^-; \underline{a}_t, \underline{a})H^T(t_i) + R(t_i)]\} \\
 & - 1/2 \operatorname{tr}\{P^{-1}(t_i^+; \underline{a})P_e(t_i^+; \underline{a}_t, \underline{a})\}
 \end{aligned}$$

This approximation is reasonable as the determinants of $P(t_i^+)$, and A (in the absence of the numerical difficulties) can be expected to have minimal impact on the ambiguity function as the primary sensitivity of the ambiguity function is in the quadratic terms which are being preserved.

II.3 Ambiguity Function Analysis

Analysis of the ambiguity function was based on three-dimensional view plots which show the magnitude of the ambiguity function as a continuous surface over the two dimensional parameter space, for specific true parameter values. Recall that the two uncertain parameters are the mass density of the spacecraft arms and the modulus of elasticity of the arms. Three typical plots are included as Figures V-1, V-2, and V-3, based on true parameter points (5,5), (3,7), and (7,3) respectively, the arrow on the plot indicates the true parameter point. The numbers for the uncertain parameters are not the true values of the parameters at those points, but indices indicating which of the ten discrete values for each uncertain parameter is used. As can be seen the ambiguity functions are fairly

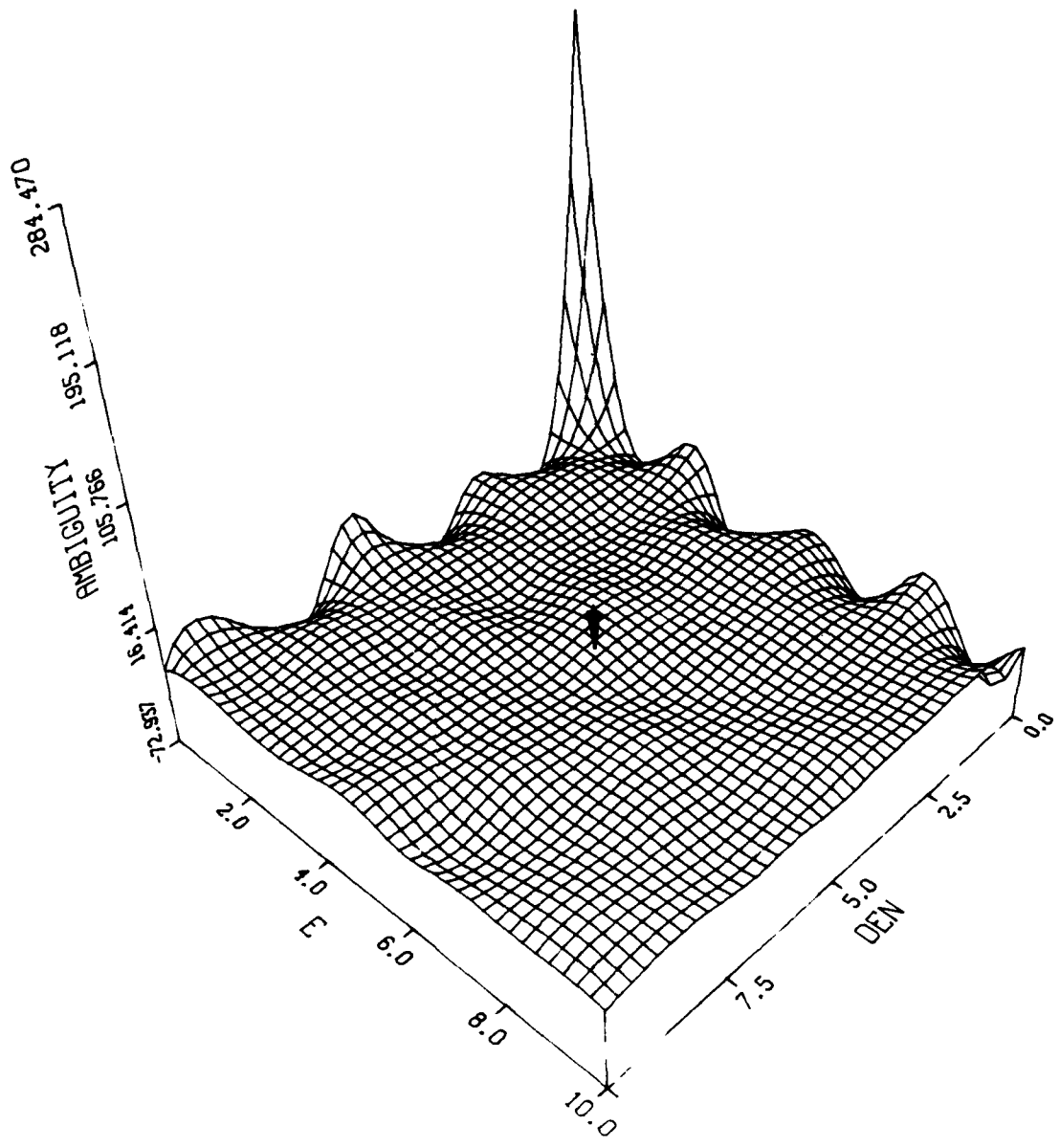


Figure V-1. Ambiguity Function at point (5,5).

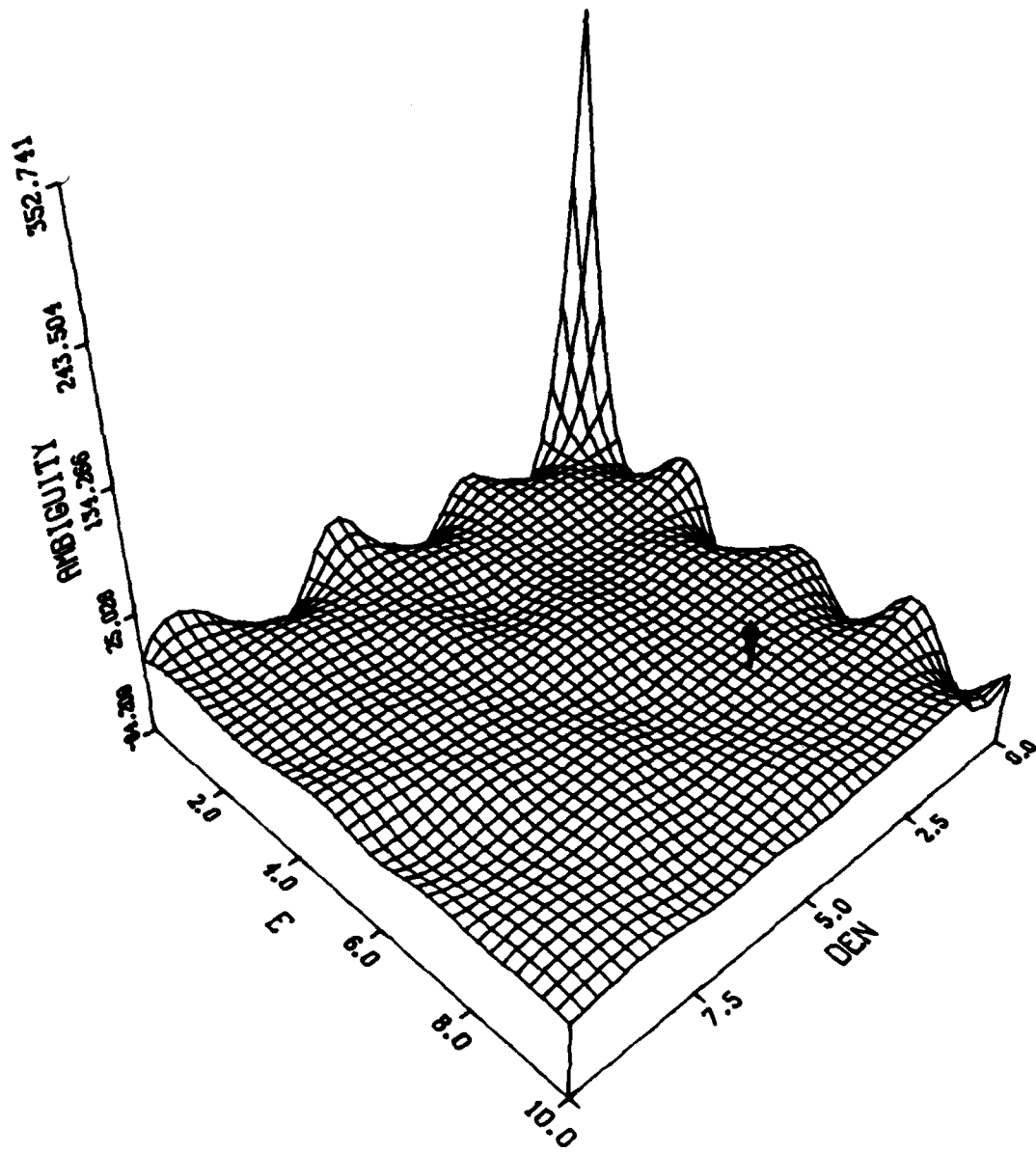


Figure V-2. Ambiguity Function at point (7,3).

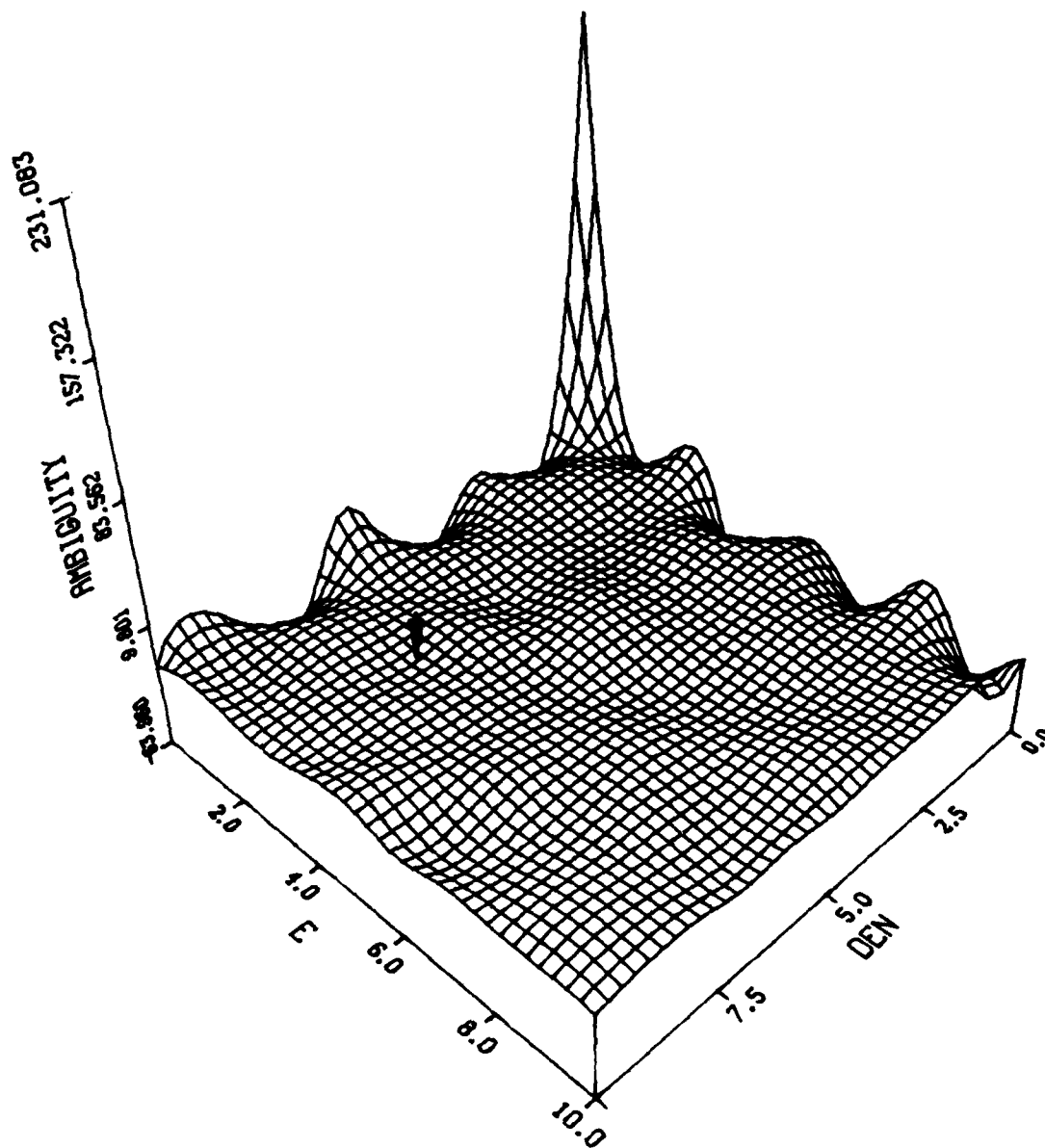


Figure V-3. Ambiguity Function at point (3,7).

flat, which indicates possible sensitivity problems, and there are several ripples, which indicates possible problems identifying the correct parameter values. The large peak at the (0,0) point should be ignored as it is an artificial result of the curve fitting routine used to generate the contours and of the fact that there is no data point at (0,0). In addition, the plots for different true parameter points are very similar. This indicates that the moving bank multiple model adaptive estimator will have difficulties identifying the correct parameter point, this is especially true when the true parameter point is not located on at least a local peak, as is the case in Figure V-3 for true parameter point (7,3). However, even if the true parameter point cannot be identified, the state estimates may still be accurate, if the different models are all doing a good job of estimating the states even though based on the incorrect system model. The true parameter point not being located on a local peak in the ambiguity surface as occurs in Figure V-3 was not anticipated, nor believed possible, and is likely due to the approximations used in the calculation of the ambiguity function. However, a rigorous proof that the true parameter point must lie on at least a local peak is not easily done and is beyond the scope of this thesis.

The ripples and multiple local peaking argue for the use of a contraction algorithm to aid initial acquisition of the true parameter point. Without a contraction algorithm,

the moving bank must start at its smallest discretization and move to center itself on the local peak with the greatest height. If a local peak not due to the true parameter point is within the area of the bank, or if while the bank moves it reaches an erroneous peak first, the bank may remain there (noise may make it move off). The goal of the contraction algorithm is not only to speed the initial movement of the bank, but by covering more area, the bank may be able to contract to the neighborhood of the correct parameter point and ignore local peaks not due to the true parameter point.

Conversely, the flatness of the surfaces argues against residual monitoring using the likelihood quotient being able to signal an expansion of the bank by identifying when a jump change in the true parameter point has taken place. On a flat ambiguity function surface it is likely that changes in the residuals due to the jump change will be masked by the normal changes in residuals seen as the effects of the dynamics and measurement noises. For residual monitoring using the likelihood quotient to work, the jump change in the true parameter point has to cause a change in the residuals greater than the fluctuations due to noise. If not a jump event detection threshold level can not be set, because the noise changes would signal a jump change. The flatness of the ambiguity surfaces also indicates that adaptive estimation and control, of any kind, is probably not worthwhile for this system. In fact, a filter based on

a nominal parameter point may provide adequate state estimates and an acceptable controller gain matrix. Of course the definition of adequate state estimates π , and acceptable controller gain matrix π are based on the level of accuracy required by the individual implementation. The flatness of the surfaces was not anticipated, considering the large changes in the parameters used to create the parameter space, and the correspondingly large changes in the eigenvalues of the bending modes that were observed (see Appendix A for a listing of the eigenvalues). The approximations used in computing the ambiguity function may be one reason that the surfaces are so flat. The wealth of measurement data (five position measurements and five acceleration measurement) available for this model also worked to minimize the impact of the changes in the model used to set up the parameter space upon the precision of state estimation or control. This was seen by examining the Kalman filter gain matrices in conjunction with the measurement matrices and observing that a heavy weight was put on the measurements as opposed to the estimated states before the measurements. The inaccurate estimated states are thus essentially ignored.

The academic solution to the difficulties encountered at this point might be to reduce the number or precision of the measurements to obtain a problem for which adaptive estimation and control would be more useful. For this

problem it would be more useful to reduce the amount of adaptivity by only using one uncertain parameter, or by taking the adaption off line entirely and only using a robustified filter/controller. However; despite the potential difficulties forecast by the ambiguity function analysis, the addaptive estimator was applied to this problem. This was done to confirm the usefulness of the ambiguity function as a predictor of estimator performance, and to evaluate how large an effect the difficulties had on estimator perfomance. The following sections describe the algorithm characteristics for this application.

V.4. Bank Movement

The first simulations started the moving bank at it finest discretization level, and located the bank center at (5,5), chosen as the approximate center of the parameter space, and allowed it to move using the probability monitoring movement algorithm developed by Maybeck and Hentz [26,27] described in Chapter II, where the bank is centered over the filter having the largest probability weighting. The points (2,2), (3,7), (5,5), (8,8), and (7,3) where chosen as the true parameter points to test the bank's ability to move in different directions. Before making multiple Monte Carlo runs to test the probability initialization algorithms, single runs were made to set the probability move threshold, and to determine a strength for the dither signal be used to excite the system states and

allow parameter point acquisition in the absence of any feedback control. A dither signal of 200 amplitude, divided equally between the three inputs, at 20 Hz was applied to all the control inputs. The dither signal was determined by trial and error as a determination of an optimal signal (see [5:68-151,12:223-260] for a discussion of optimal inputs) would be beyond the scope of this thesis.

Maybeck and Hentz found that a probability move threshold of 0.05 enabled the bank to move anytime the bank was not centered on the filter having the highest probability weighting [26:62,27:19]. This will be true for any threshold below $1/9 = 0.111$, which is the lowest probability weighting the center filter can have, and still be weighted equal to, or heavier than the other filters. In this application however, it was found that a threshold slightly larger than 0.111 speeded acquisition of the true parameter point by eliminating movement (possibly in the wrong direction) due to noise when the filter probability weightings were all close in value. A threshold of 0.15 allowed the bank to center itself on the (3,7) true parameter point after 0.98 seconds vs. 1.54 seconds with a threshold of 0.05. Increasing the threshold further to 0.2 however increased the acquisition time to 1.28 seconds. A threshold level of 0.15 was therefore used for all subsequent runs. For the single runs, the pseudo-random number generator used to generate the noise signals was always seeded with the same starting number to ensure that

the runs were directly comparable.

Monte Carlo studies of the bank's ability to move to each of the five chosen true parameter points, using ten runs apiece, were made for both the simple method of initializing the probability weightings of the new filters brought into the bank by weighting each new filter equally, and for the more complicated method of estimating a probability weighting for each new filter based on Equation (II-30). The results can be seen in Appendix B in Figures B-1 through B-10.

These figures are plots of the statistics (mean \pm one standard deviation) of the error in the parameter estimates. EA(1) is the error in the estimate of the parameter reflecting the mass density of the arms, and EA(2) is the error in the estimate of the parameter reflecting the modulus of elasticity of the arms. Once the matrices describing the system at a particular parameter point have been calculated, the true values of the two uncertain parameter were no longer used. It is far more convenient to index the parameter values to the whole numbers one through ten reflecting their position in the parameter space. Thus, the (1,1) parameter point is where the uncertain parameters take on the discrete value 20% less than their nominal values, and the point (6,6) is where they are equal to their nominal values. An increase of one in the parameter index corresponds therefore, to an increase of 4% of the nominal value. The errors in the parameter estimates are calculated

in terms of the parameter indices; if the true parameter point is at (4,5) and the location of the estimated parameter vector is (3.5,6.0), the error in the parameter estimate plots will show a value of 0.5 for the mass density parameter, and a value of -1.0 for modulus of elasticity parameter. It is important to note that because the estimated parameter vector is a weighted average of the indices of the discrete parameter points whose associated filters are currently implemented in the moving bank, the location of the estimated parameter vector is not likely to be exactly equal to one of the discrete parameter points.

The first set of plots shows the error in the parameter estimate (\underline{e}_a) when the probability weightings of the new filters in the bank were calculated using Equation (II-30) (Figures B-1 through B-5). The second set of plots shows \underline{e}_a when the new filters are initialized by equally distributing the unused probability (Figures B-6 through B-10). A comparison shows that there is no appreciable difference in the results of the two methods. However, it is a considerably larger computational burden (by several orders of magnitude, depending on the architecture of the processor) to compute the probability weightings using Equation (II-30).

The performance predicted by the ambiguity function analysis can also be seen in the plots of \underline{e}_a error statistics for the case of only moving the bank without changing its size, using either of the filter initialization

algorithms. In all cases the estimator does a better job of estimating the second parameter than the first: it is always identified within one standard deviation by the time one second has elapsed. However, only the true parameter points at (5,5) and (3,7) were correctly identified, for both parameters, within one standard deviation. These statistics are matched by center of the bank statistics: in all cases the bank center had moved to the correct second parameter by 0.8 seconds.

Examination of the ambiguity function plots shows ridges running diagonally across the surfaces. The bank will not move down into the valleys between the ridges, seeking instead the peaking regions of the ambiguity function surface. Therefore, when the true parameter point is not at a local peak accessible from the ridge on which the starting point (5,5) is located, the estimator is forced to make a choice as to which parameter it will do a good job estimating. The reason it is the second parameter that is favored can be seen by examining Tables A-1 and A-2 in Appendix A. These tables show the eigenvalues of the second and forth bending modes at the various parameter points. It can be seen that a change in the second parameter (E: the modulus of elasticity) causes a larger change in the eigenvalues than does a change in the first parameter (ρ : the mass density). Thus the filters on the ridge with the more correct value of E receive a higher probability weighting than the filters with the more correct value of

density, and the bank moves toward the correct value for the second parameter.

The overall flatness of the ambiguity surfaces was reflected in the plots of the statistics of the errors in the state estimates (\underline{e}_x). These plots showed \underline{e}_x to be essentially independent of e_a . Figure B-11 shows \underline{e}_x statistics for the case where the true parameter point was at (3,7), and Figure B-12 depicts \underline{e}_x for the case where the true parameter point was at (8,8). The states 1-6 reflect the position of the rigid body, the second bending mode, and the fourth bending mode respectively, and then the velocities in the same order. Both sets of plots are identical, as were the plots no matter where the true parameter point was set. The point is especially illustrated by Figures B-13, and B-14, which show \underline{e}_x characteristics for just a single filter based on assumed parameter values at (5,5). In the first case (Figure B-13) the true parameter point was at (3,7), and in the second case the true parameter point was actually at (5,5): the filter with parameters at (5,5) does an equally good job of estimating the states in both cases.

V.5. Bank Contraction

Beginning the run with an acquisition phase using a coarser bank discretization, then contracting the bank to the finest discretization in two steps was tested to see if it speeded parameter acquisition. Parameter error

covariance monitoring was used to detect the acquisition of the parameters, and signal the change in discretization. The scalar measures associated with the parameter estimate covariance matrix, selected to signal contraction, were the parameter variances (the diagonal elements of the covariance matrix). The variances of both parameters were compared to a contraction threshold and the result was used to signal contraction.

Single runs were used to set the contraction thresholds. Two were required, one for each level of contraction. The first coarsest discretization level places the filters four parameter points apart. The second intermediate discretization level places the filters two parameter points apart. And the final finest discretization level places the filters one parameter point apart. An additional consideration was whether to require both or only one of the variances to be below the threshold in order to signal the contraction. It was found that requiring both variances to be below the threshold gave a more accurate determination of when contraction was warranted; however, this required tradeoffs in the determination of the threshold, as the same threshold was not necessarily most appropriate for both variances. The threshold values were then chosen equal to the larger of the two variances at the time when the true parameter point would lie within the area of the bank after it contracted about the estimated parameter point. This required

artificial knowledge of the true parameter point, and the printout of the time period by time period statistics on the bank location and estimates. A threshold was thus selected for each of the four test points (the same test points as for the move analysis with the exception of (5,5)) and smallest of the five then selected as the best overall. This threshold level setting mechanism worked well for signaling contraction from coarsest discretization level to the intermediate discretization level. But it could not be used in all cases to select a threshold for signaling the second contraction to the finest discretization level. When the true parameter point was at (8,8), the parameter estimate never moved closer to the true parameter point after the first contraction. Therefore a contraction threshold based only on the other three points was used. The contraction thresholds chosen were 6.0 for the four-to-two contraction, and 2.6 for the two-to-one contraction.

The contraction thresholds proved to be sensitive to the probability weight lower bound used to keep the bank from locking on to a single filter. When the probability weight lower bound is set too high, it prevents the the variances from decreasing below the thresholds in all cases. Therefore, the contraction thresholds need to be reset if the probability lower bounds change. Lower bounding the probability weights also causes a larger error in the parameter estimate, and possibly the state estimate (although that was not true in this implementation), as it

requires an artificially high probability weight be assigned to a poor estimate in steady state. The purpose in lower bounding the probability weights is to allow the bank to react to a change in the true parameter point after the initial true parameter point has been identified. There is thus a tradeoff to be made between keeping the probability lower bound high enough to enable quick reaction to changes in the true parameter point, and low enough not to interfere with the contraction algorithm nor cause too great an error in the parameter and state estimates. A lower bound of 0.01 worked well in this implementation; when it was raised to 0.05, the bank would not contract from the coarsest discretization.

The results of using the contraction algorithm can be seen in Figures B-15 through B-18, which show the plots of $\underline{\epsilon}_a$ statistics vs. time for the true parameter points (2,2), (3,7), (7,3), and (8,8) respectively. The results can be compared to the plots of $\underline{\epsilon}_a$ characteristics vs. time for the movement-only options depicted in Figures B-1 through B-10. It can be seen that the use of the contraction acquisition phase is successful in correctly identifying parameter two (E) more quickly; however only small improvements in reducing the error in parameter one (ρ) are achieved and only temporarily. The expense is the larger standard deviations which result from the coarser discretizations. In terms of the shape of the ambiguity function; the acquisition cycle is successful because the flatness of the

surface, and the ripples and local peaks slow the movement of the bank.

There is very little added (and perhaps less) computation due to the contraction algorithm. Either one or three additional comparisons are required in the logic (one to determine if the bank is at a coarse discretization; if so, each of the two parameter error variances is compared to the appropriate threshold), but the operations required for the actual contraction are very similar to those required for a move and, as all of the probability weights are reset to 0.111 upon contraction, less initialization computation is required than is required following a bank move. In addition some computation is saved, since with contraction fewer moves are required to get the bank to its final location. Thus computation is saved in the short run, but as the run continues, in the absence of a re-expansion to the coarse discretization, the total computation will eventually be slightly greater. If re-expansion occurs following the signaling of a jump change in the true parameter values there is again a savings during the reacquisition of the true parameter values. The real issue here is the amount of computation per sample period; and the contraction algorithm is essentially equal to the non-contraction algorithm in that regard.

V.6. Jump Changes

It was hoped that residual monitoring would be able to detect jump changes in the true parameter point. Unfortunately, when an attempt to set a detection threshold was made, the likelihood function proved much more sensitive to the noise inputs than to the difference in the residuals due to the change in the true parameter point. When the attempt to set a threshold was made, on one run the likelihood function actually decreased at the time of the jump change due to a smaller noise input that sample period. The failure of the jump change detection by residual monitoring can be explained in terms of the flatness of the ambiguity function as noted earlier. Therefore, no evaluation of the effectiveness of a reacquisition period, using the bank expansion and contraction algorithms, following a jump change in the true parameter point could be made.

A plot of \hat{e}_a statistics vs. time for a case where the bank was started at its largest discretization, allowed to contract and move to a true parameter point at (3,7), and react to a jump change in the parameter point to (5,5) at 1.0 second, reveals that the move algorithm does continue to track the true parameter point via bank movement without a change in the bank discretization level (Figure B-19). As before parameter two is consistently estimated better, both before and after the jump change. For the same situation, Figure B-20 reveals that, as before, the state estimates for this implementation do not suffer.

V.7 Controller Analysis

The evaluation of the effectiveness of the various adaptive control algorithms was also abandoned due to the accuracy of the state estimates, independent of the accuracy of the parameter estimate. In addition, inspection of sample B_d matrices reveals little change due to ρ , and almost none due to E ; so no large errors in the control input can be expected as a result of the use of the wrong B_d matrix.

V.8 Summary

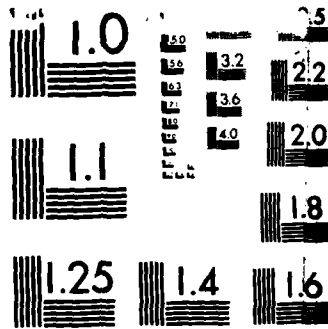
The results of the simulations were presented in this Chapter. Initial problems due to numerical difficulties were overcome through the use of judicious approximations. Analysis of the ambiguity function revealed several insights into the expected performance of the multiple model adaptive estimator, and the probable lack of a need for an adaptive controller. These insights were borne out by the actual simulation results. The two movement algorithms worked equally well but were unable to move across the valleys depicted in the ambiguity function plot. The estimator proved more sensitive to the second uncertain parameter ($E =$ the modulus of elasticity of the arms) than to the first ($\rho =$ the mass density of the arms). The contraction algorithm gave some increase in performance, but essentially only for the second uncertain parameter. Importantly, none of the movement or contraction logics affected the overall accuracy

of the state estimates in any discernible manner. Residual monitoring proved ineffectual in detecting jump changes in true parameter points, making an evaluation of a reacquisition phase, using the bank contraction and expansion algorithms, impossible. The evaluation of the adaptive controllers was also abandoned due to the accuracy of the state estimates. Conclusions regarding these results and recommendations for areas of further study are presented in Chapter VI.

VI. Conclusions and Recommendations

VI.1 Introduction

The initial, obvious, conclusion to be drawn from this research is that, for this system model, adaptive estimation and control is not required. Despite this disappointing conclusion, this thesis does provide valuable insight on the performance of multiple model adaptive estimation. For the class of problems where very accurate estimates of the system states are required, adaptive estimation may prove worthwhile, if the adaptation provides only marginally better state estimates. This research provides information on the problems that may arise when the measurement accuracy is great, resulting in insensitivity to incorrect modeling of uncertain parameters, similar problems may also arise if the uncertain parameter variation is small. Additionally, even for problems which require adaptation over a wide range of uncertain parameters, where parameter variation sensitivity is good, some of the problems discussed here may have applicability in what may be local spots of parameter variation insensitivity. The importance that early ambiguity function analysis was discovered to have, should have impact on estimator/controller design regardless of the amount of system adaptation required, or sensitivity of the system to uncertain parameter variation.



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

VI.2 Conclusions

Ambiguity function analysis was able to predict/explain most of the problems encountered in the simulations of the multiple model adaptive estimator/controller. It proved to be a valuable tool that should be used early in the design of the adaptive estimator in order to aid in the selection of appropriate parameters for estimation, and of workable move/contract/expansion strategies. This is especially true, as the results of the simulations showed that each of the strategies investigated have shortcomings which need to be overcome. The bank may not be able to reach all points by movement alone, or at all. This occurs when the points are located in a valley in the ambiguity function, or if the bank encounters a local peak while moving towards the correct parameter values. Square contraction may require compromise in the setting of the contraction thresholds, which will hinder its performance. This occurs when the bank does a better job of estimating one parameter than another. Separate contraction thresholds which enable the bank to contract separately in each direction, when it is most appropriate to contract in that direction, should improve performance. And residual monitoring via the likelihood quotient may not be able to detect jump changes, depending on the relative effect on the residuals of noise vs. system changes.

VI.2.1 Ambiguity function Analysis. The first question that early ambiguity function analysis can answer is whether adaptive estimation/control is needed at all. This is not always known in the initial stages of system design, as was the case for the model used for this investigation. Once it is determined that adaptive estimation/control is required, the ambiguity function analysis can determine which of the uncertain parameters need to be estimated and which may be ignored. It may be possible to combine multiple uncertain parameters into a single artificial parameter, or into a reduced number of artificial parameters, to be used by the adaptive estimator/controller. This enables the estimator to concentrate on estimating the uncertain system parameters (either physical quantities, or deriv.d quantities) that have the greatest impact on system performance.

Once the parameter space is chosen, further analysis of the ambiguity function may be useful in choosing the discretizations used as a basis for elemental filter design. The discretization could be finer in regions of greater parameter variation sensitivity, and coarser in regions of lesser parameter variation sensitivity, instead of being spread evenly over the entire parameter space. Ambiguity function analysis may then be helpful in choosing the initial conditions for the moving bank: bank center and discretization level. As seen, from certain starting positions, the bank may not be able to move to the true

parameter point. Judicious choice of the starting position may alleviate this problem for a large class of true parameter points. Then thought can be given to the bank contraction and expansion strategies to be implemented. If multiple local peaks are present, contraction strategies may be especially valuable, as contraction about the estimated parameter values could prevent the bank from moving to, and remaining on, a local peak positioned between the bank starting position and the true parameter point.

VI.2.2. Bank Movement. Despite the overall lack of a requirement for adaptive estimation in this application, because all of the elemental filters did an acceptable job of state estimation, the probability monitoring method of moving the bank was able to do a surprisingly good job. Even when the bank was not able to center itself over the true parameter point, it was able to move in the right general direction. Two refinements to the algorithm were discovered. Calculation of the probability weightings for initialization of new filters in the bank based on the distance from the estimated true parameter point is not worthwhile. Evenly deviding the total probability weighting taken from the filters removed from the bank, amongst those added to the bank gave equally good performance with less computation. And, it may be worthwhile to raise the bank movement threshold above 0.111, in order to prevent noise jitter movement of the bank. If the threshold is set below 0.111, it is unnecessary and can be eliminated, simply check

to insure the bank is centered on the filter with the highest probability weighting.

VI.2.3. Contraction. Contraction of the bank from an initial coarse discretization to a fine discretization, proved to be useful in improving the speed of parameter acquisition. Little additional computation is involved in the use of this strategy; however, there is a price to pay in larger standard deviations of the parameter estimate error, as noise causes a greater change in the parameter estimate when the bank is in the coarser discretization.

The best contraction thresholds for the two uncertain parameters were not identical and a compromise threshold had to be chosen. This meant that contraction did not always move the bank center in the right direction relative to the true value of the first uncertain parameter, whereas the bank center always move towards the true value of the second uncertain parameter. A two-step contraction was used, but an advantage of the two step contraction vs. a one-step contraction is unproven.

VI.2.4. Residual Monitoring. Monitoring the residuals through the use of a likelihood quotient in order to detect a jump change in the true parameter vector value did not prove useful because of the lack of sensitivity in either state estimate precision, or residual characteristics, to assumed parameter value. The magnitude of the likelihood quotient varied more as a result of the system noises than as a result of the changes in the uncertain parameters,

making it impossible to select a threshold value at which to trigger an expansion. It must be remembered however, that the residuals are the only real source of information on the true system and must be the basis for all move/contract/expand strategies.

VI.3. Areas of Further Study

Further study in all areas of moving bank multiple model adaptive control is still needed. A specific area where further study is needed was revealed by the importance ambiguity function analysis was found to have. If the ambiguity function could be shaped, by changes in the uncertain parameters chosen to be estimated, or by changes in the uncertain parameter discretization, to reflect desirable characteristics for the move/contract/expand logics, estimator/controller performance may be enhanced. The ambiguity function could be shaped through selection of artificial uncertain parameters and choice of discretized parameter points.

An evaluation of the advantages/disadvantages of rectangular contraction vs. square contraction to eliminate the undesirable compromises necessary in the selection of a contraction threshold would be worthwhile. Rectangular contraction could allow the contraction in each direction when it is most appropriate, making bank size in one direction independent of bank size in the other. Rectangular contraction could prevent the movement of the

bank center away from the true value of one of the uncertain parameters because of the current inaccurate estimate of that parameter. These possible advantages would have to be weighed against the required increase in computational loading. Evaluation of two-step vs. one-step contraction under different conditions might reveal when one is more valuable than the other. Other methods of triggering the contractions could also be workable. It may be possible to contract away from filters with very small probability weightings.

Methods of detecting jump changes in the uncertain parameter values still need further investigation. Residual monitoring using the likelihood function may be adequate in applications, unlike the application investigated here, where wrongly assumed parameter values do affect state estimation. It may be possible to make modifications to the likelihood quotient method to prevent noise from masking the results of uncertain parameter jump changes by adding additional conditions. An evaluation of the changes in the probability weightings relative to each other may also be capable of indicating jump changes. If all of the probabilities converge on a value of about 0.111, this may furnish the necessary indication.

The use of artificial intelligence techniques to evaluate the residuals and trigger all of the manipulations of the moving bank has been suggested. This has the advantage of looking directly at the real source of

true system information. However, artificial intelligence techniques tend to be computationally intensive, which is the opposite of the goal of moving-bank MMAE as opposed to full-bank MMAE, so there is a tradeoff inherent in this proposal.

Finally, issues which were identified prior to this thesis still need to be addressed. These include the evaluation of the controllers for a realistic system (which had to be abandoned here). Also the study of slowly varying uncertain parameter values, and uncertain parameter values not equal to the discrete parameter values was not addressed here.

VI.4. Summary

This thesis has addressed the evaluation of moving-bank multiple model adaptive estimation and control algorithms, as applied to a model of a large flexible spacecraft. It was seen that, although the model turned out not to require adaptive control, that moving-bank MMAE could still provide estimates of the uncertain parameters. Based on the performance of the moving bank estimator, several refinements in the move and contraction decision logics were made. In addition, it was revealed that ambiguity function analysis can be an invaluable tool in designing a moving bank estimator. The moving bank technique continues to show great promise as a practical adaptive estimation/control design strategy, and research in this area should continue.

Appendix A.

Detailed Model Development

The equations for the elements of the mass and stiffness matrices were given in Chapter III in integral form. They are repeated here for clarity, then the simplified closed form is presented. The equations for the control and Measurement matrices are also developed in a more complete form. Then samples of the M (mass), K (stiffness), B (control), and H (measurement) matrices are given, followed by two tables showing how the free system eigenvalues of the second and fourth bending modes vary throughout the parameter space.

The equations for the elements of the mass matrix are repeated from Chapter III for $i = 1, \dots, p$; and $j = 1, \dots, p$:

$$M(1,1) = 2 (I_c + I_{2c})$$

$$M(1, i+1) = 2\rho th \int_R^{R+L} x\theta_i dx + 2m_2(R+L)\theta_i(z=x-R=L) \\ + 2I_{2c}\theta'_i(z=x-R=L)$$

$$M(I, i+p+1) = 2\rho th \int_R^{R+L} y\theta_i dy + 2m_2(R+L)\theta_i(z=y-R=L) \\ + 2I_{2c}\theta'_i(z=y-R=L)$$

$$M(i+1,1) = M(1, i+1)$$

$$M(i+p+1,1) = M(1, i+p+1)$$

$$M(i+1, j+1) = 2\rho th \int_R^{R+L} \theta_i \theta_j dx$$

$$+ 2m_2 \theta_i(z=x-R=L) \theta_j(z=x-R=L)$$

$$+ 2I_{2c} \theta'_i(z=x-R=L) \theta'_j(z=x-R=L)$$

$$M(i+p+1, j+p+1) = 2\rho th \int_R^{R+L} \theta_i \theta_j dy$$

$$+ 2m_2 \theta_i(z=y-R=L) \theta_j(z=y-R=L)$$

$$+ 2I_{2c} \theta'_i(z=y-R=L) \theta'_j(z=y-R=L)$$

$$M(i+1, j+p+1) = 0$$

$$M(i+1+p, i+1) = 0$$

where θ_i is a function describing the bending modes of the arms, defined as:

$$\theta_i(z) = 1 - \cos(i\pi z/L) + (1/2)(-1)^{i+1}(i\pi z/L)^2$$

$$z = x - R \text{ (or } y - R)$$

and θ'_i is the first derivative of θ_i with respect to z .

The physical dimensions of the model t , h , m_2 , I_{2c} , I_{00} , R , and L , are as shown in Figure III-1. In addition ρ is the mass density of the arms, and:

$$I_c = (1/2)I_0 + 2I_1 + 2m_2(R+L)^2$$

$$I_1 = \rho th \int_R^{R+L} x^2 dx$$

The elements of the stiffness matrix are given for

$i = 1, \dots, p$; and $j = 1, \dots, p$; by:

$$K(1,1) = 0$$

$$K(i+1,1) = 0$$

$$K(i+1+p,1) = 0$$

$$K(1,j+1) = 0$$

$$K(1,j+p+1) = 0$$

$$K(i+1,j+1) = 2EI \int_0^L \theta_i''(z) \theta_j''(z) dz$$

$$K(i+1+p,j+1+p) = K(i+1,j+1)$$

$$K(i+1+p,j+1) = 0$$

$$K(i+1,j+1+p) = 0$$

where E is the Modulus of Elasticity of the arms, and I is the area moment of inertia based only on t and h as:

$$I = (1/12)ht^3$$

and θ_i'' is the second derivative of θ_i with respect to z .

When the necessary substitutions are made and the integrals are evaluated, the simplified equations become:

$$M(1,1) = I_0 + 4th(L^3/3 + RL^2 + R^2L) + m_2(R^2 + 2RL + L^2) + 2I_{2c}$$

$$M(1,i+1) = 2\rho th \{L^2/2 + [(-1)^{(i+1)} + 1]L/(i\pi)^2 + (-1)^{(i+1)}(i\pi)^2 L(L/8 + R/6) + RL\} + 2m_2(R + L)[1 + (-1)^{(i+1)}(1 + (i\pi)^2/2)] + 2I_{2c}(-1)^{(i+1)}I^2\pi^4/L$$

$$M(1,i+p+1) = M(i+p+1,1) = M(i+1,1) = M(1,i+1)$$

$$\begin{aligned}
M(i+1, j+1) = & 2\rho_{th} \{L[(-1)^{(i+1)}i^2 + (-1)^{(j+1)}j^2\pi^2L/6 \\
& + (-1)^{(i+j)}L[Ri^2/Ri^2 + Ri^2/Rj^2 \\
& + (ij\pi)^2/20]\} \\
& + 2m_2 [1 + (-1)^{(i+1)}(1 + (i\pi)^2/2)] \\
& [1 + (-1)^{(j+1)}(1 + (j\pi)^2/2)] \\
& + 2I_{2c}(-1)^{(i+j)}(ij\pi/L)^2 \\
& + \{0: i \neq j; \rho_{th}L: i=j\}
\end{aligned}$$

$$M(i+p+1, j+p+1) = M(i+1, j+1)$$

$$M(i+1, j+p+1) = M(i+p+1, j+1) = 0$$

and

$$\begin{aligned}
K(i+1, j+1) = & Et^3 h(-1)^{(i+j)}L(ij\pi)^2/(6L^4) \\
& + \{0: i \neq j; Et^3 h(ij\pi)^2/(12L^3): i=j\}
\end{aligned}$$

$$\begin{aligned}
K(1, 1) = & K(i+1, 1) = K(1, i+1) = K(i+p+1, 1) = K(1, i+p+1) \\
& = K(i+1, j+p+1) = K(i+p+1, j+1) = 0
\end{aligned}$$

The B matrix is given by:

$$B = \begin{bmatrix} 1 & 2 & 2 \\ 0 & 2\vartheta'_i(z=L/2) & 0 \\ 0 & 0 & 2\vartheta'_i(z=L/2) \end{bmatrix}$$

where $\vartheta_i(z)$ is of dimension p (the number of modes of interest), and

$$\vartheta_i(z=L/2) = (i\pi/L)\sin(i\pi/2) + (-1)^{(i+1)}(i\pi)^2/8$$

The H matrix is given by:

$$H = \begin{bmatrix} 1 & \underline{0}^T & \underline{0}^T \\ 0 & \underline{\phi}_i^T(z=L/2) & \underline{0}^T \\ 0 & \underline{\phi}_i^T(z=L) & \underline{0}^T \\ 0 & \underline{0}^T & \underline{\phi}_i^T(z=L/2) \\ 0 & \underline{0}^T & \underline{\phi}_i^T(z=L) \end{bmatrix}$$

1 column p columns p columns

H is thus a 5 x N matrix. and

$$\phi_i(z=L/2) = 1 - \cos(i\pi/2) + 0.5(-1)^{(i+1)}(i\pi)^2$$

and

$$\phi_i(z=L) = 1 - (-1)^{(i+1)}(i\pi)^2$$

When these expressions are evaluated for the constant values given in Table III-1 (this equates to the (6,6) point in the ρ/E parameter space), and for $p = 2$, the following matrices result:

$$M = \begin{bmatrix} 28.19 & 13.02 & -36.09 & 13.02 & -36.09 \\ 13.02 & 17.65 & -49.07 & 0.0 & 0.0 \\ -36.09 & -49.07 & 137.50 & 0.0 & 0.0 \\ 13.02 & 0.0 & 0.0 & 17.65 & -49.07 \\ -36.09 & 0.0 & 0.0 & -49.07 & 137.50 \end{bmatrix}$$

$$K = \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 340.61 & -908.30 & 0.0 & 0.0 \\ 0.0 & -908.30 & 5449.83 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 340.61 & -908.30 \\ 0.0 & 0.0 & 0.0 & -908.30 & 5449.83 \end{bmatrix}$$

$$B = \begin{bmatrix} 1.0 & 2.0 & 2.0 \\ 0.0 & 4.04 & 0.0 \\ 0.0 & -9.87 & 0.0 \\ 0.0 & 0.0 & 4.04 \\ 0.0 & 0.0 & -9.87 \end{bmatrix}$$

$$H = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 2.23 & -2.93 & 0.0 & 0.0 \\ 0.0 & 4.93 & -17.73 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 2.23 & -2.93 \\ 0.0 & 0.0 & 0.0 & 4.93 & -17.73 \end{bmatrix}$$

As seen in Chapter III the free system eigenvalues of the system can be found via Equation (III-2):

$$K\underline{x} = \lambda\underline{x}$$

The eigenvalues of the second and fourth bending modes (the unison bending modes to be controlled in this application) are tabulated in tables A-1 and A-2 for all the discrete values of ρ and E in the parameter space.

Table A-1. Eigenvalues for Second Bending Mode.

	E									
	1	2	3	4	5	6	7	8	9	10
1	49.23	51.69	54.15	56.62	59.08	61.54	64.00	66.46	68.92	71.35
2	49.05	51.50	53.95	56.41	58.86	61.31	63.76	66.22	68.67	71.12
3	48.87	51.31	53.75	56.20	58.64	61.08	63.53	65.97	68.41	70.86
4	48.69	51.12	53.56	55.99	58.42	60.86	63.29	65.73	68.16	70.50
5	48.51	50.93	53.36	55.78	58.21	60.63	63.05	65.49	67.91	70.34
6	48.33	50.75	53.16	55.58	58.00	60.41	62.83	65.25	67.66	70.08
7	48.15	50.56	52.97	55.38	57.78	60.19	62.50	65.01	67.42	69.82
8	47.98	50.38	52.78	55.18	57.57	59.97	62.37	64.77	67.17	69.57
9	47.80	50.19	52.59	54.98	57.37	59.76	62.15	64.54	66.93	69.32
10	47.63	50.01	52.40	54.78	57.16	59.54	61.92	64.30	66.68	69.07

Table A-2. Eigenvalues for Fourth Bending Mode.

	E									
	1	2	3	4	5	6	7	8	9	10
1	2734.25	2870.96	3007.67	3144.39	3281.10	3417.81	3554.52	3691.23	3827.95	3964.66
2	2615.07	2745.83	2876.58	3007.33	3138.09	3268.84	3399.59	3530.35	3661.10	3791.86
3	2506.41	2631.73	2757.05	2882.37	3007.69	3133.01	3258.33	3383.65	3508.97	3634.29
4	2406.91	2527.26	2647.60	2767.95	2888.29	3008.64	3128.98	3249.33	3369.68	3490.02
5	2315.50	2431.17	2546.94	2662.71	2778.48	2894.25	3010.02	3125.79	3241.56	3357.33
6	2231.07	2342.62	2454.18	2565.73	2677.29	2788.94	2900.39	3011.95	3123.50	3235.05
7	2152.96	2260.61	2368.26	2475.91	2583.55	2691.20	2798.85	2906.50	3014.15	3121.79
8	2086.54	2184.56	2288.59	2392.62	2496.65	2600.67	2704.70	2808.73	2912.75	3016.78
9	2012.05	2113.70	2214.35	2315.01	2415.66	2516.31	2616.96	2717.62	2818.27	2918.92
10	1950.17	2047.69	2145.12	2242.65	2340.20	2437.71	2535.22	2632.73	2730.24	2827.74

Appendix B.
Simulation Plots

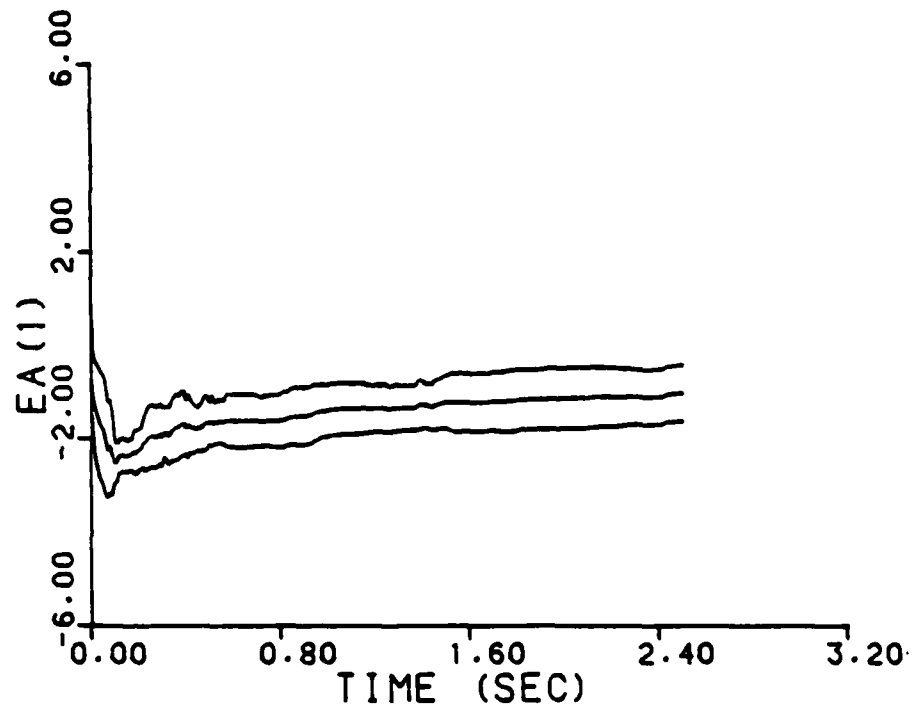
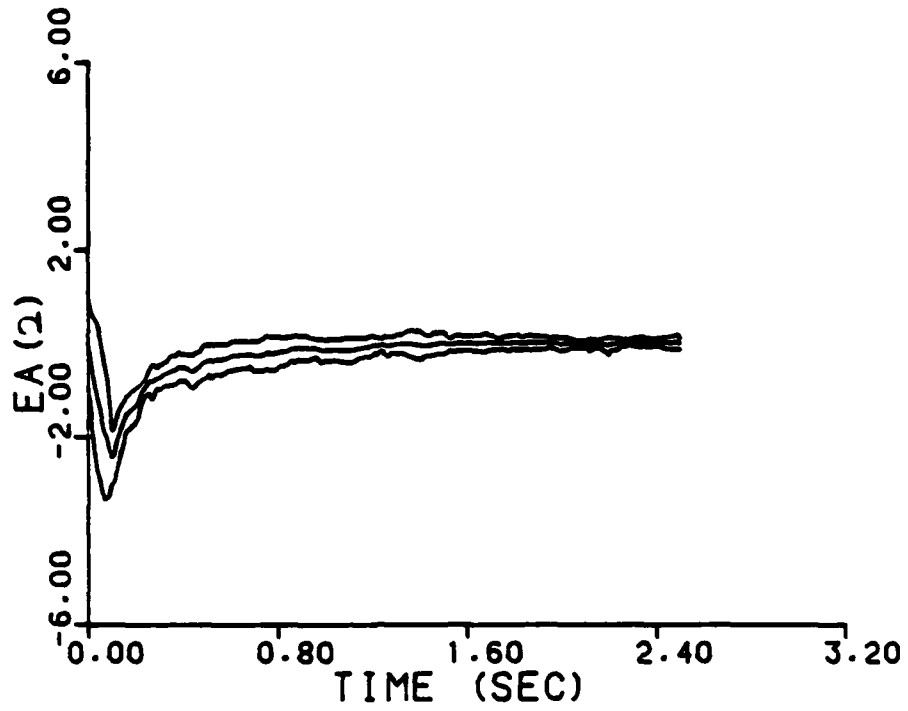


Figure B-1. Calculated probability weightings, no contraction, parameter estimate error, $\mathbf{a}_t = (2, 2)$.

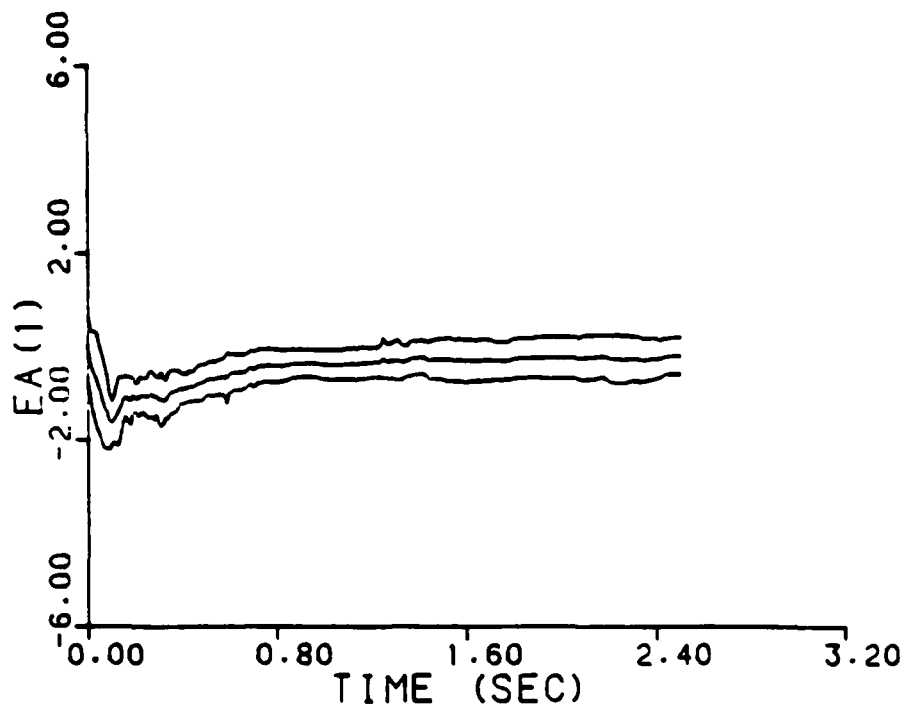
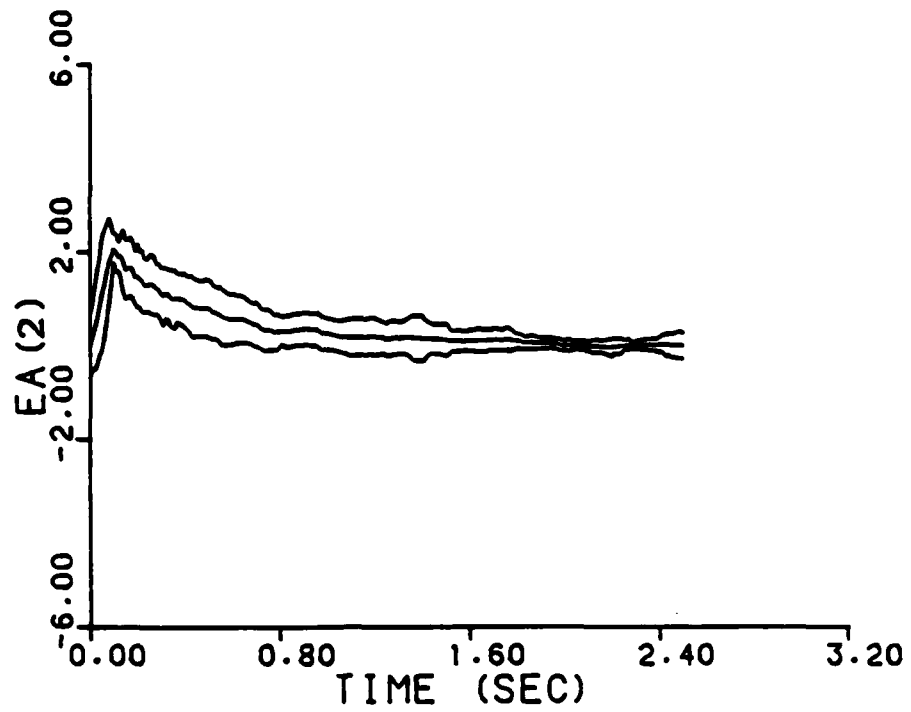


Figure B-2. Calculated probability weightings, no contraction, parameter estimate error, $\underline{a}_t = (3,7)$.

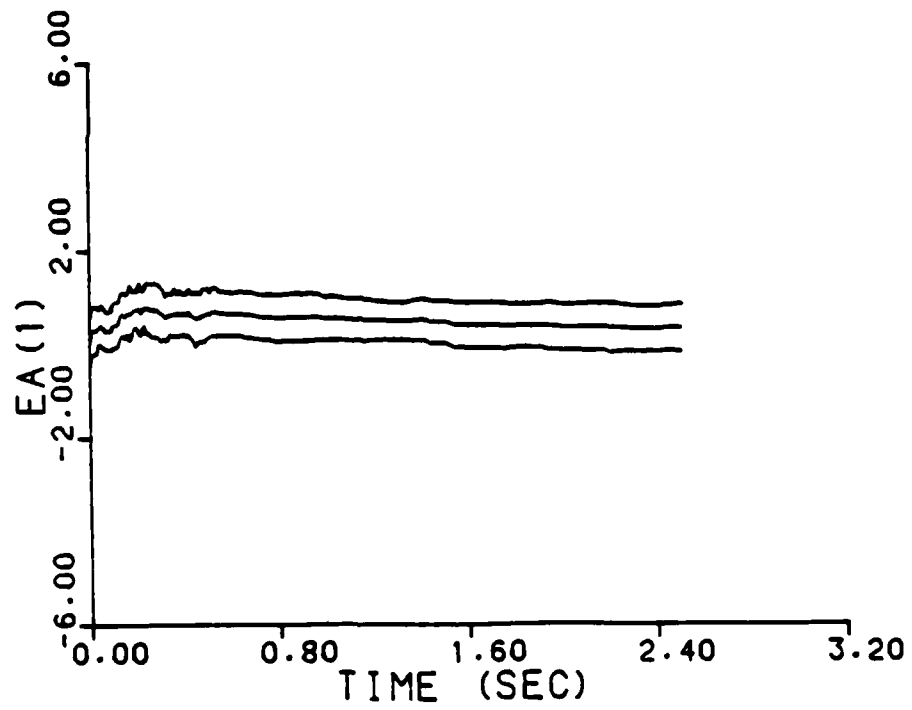
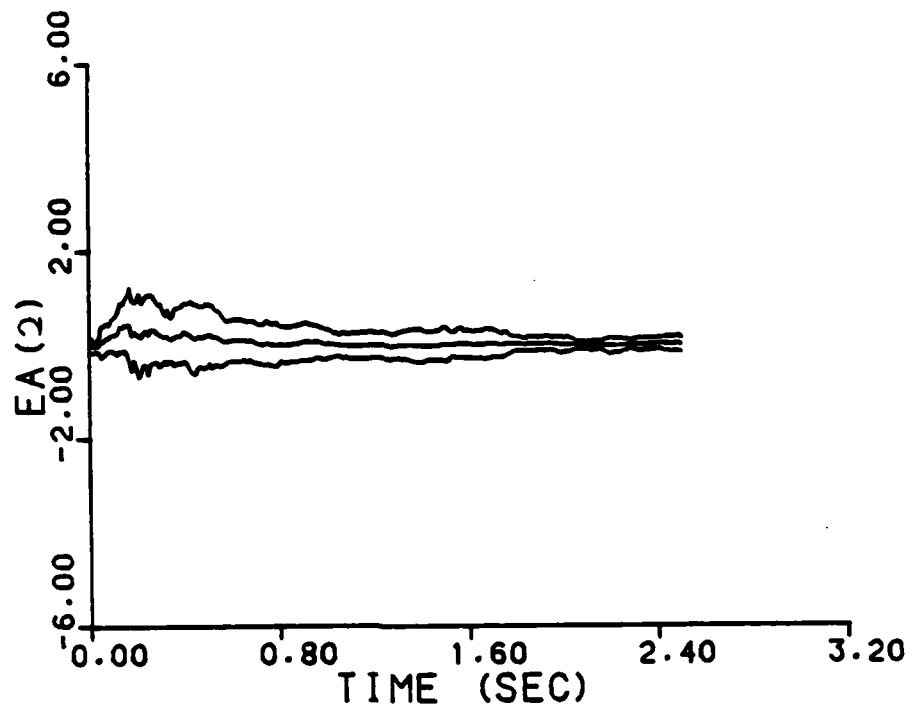


Figure B-3. Calculated probability weightings,
 no contraction, parameter estimate error,
 $\underline{a}_t = (5,5)$.

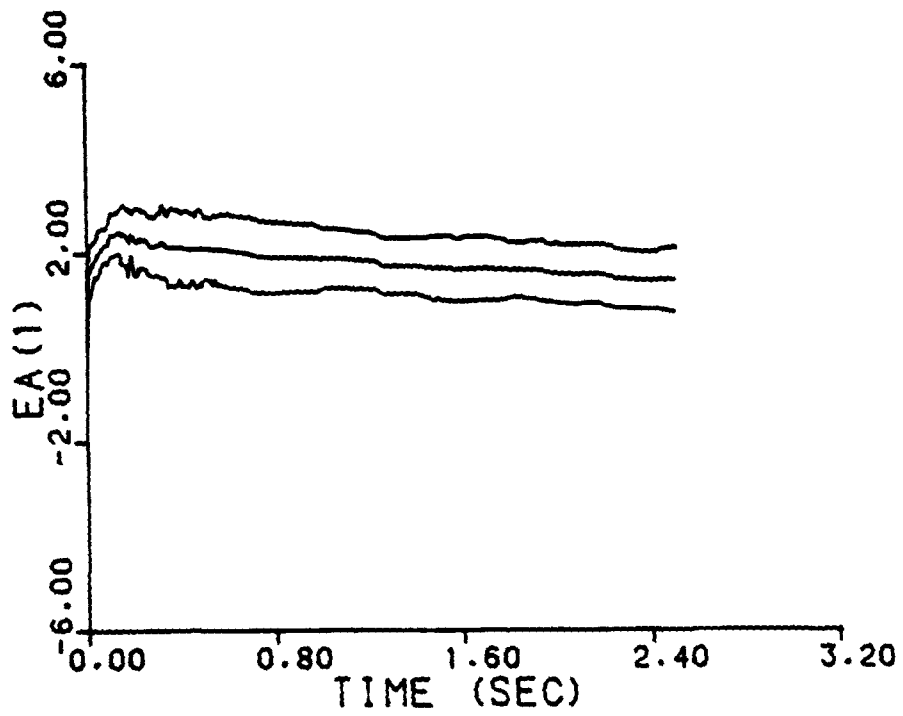
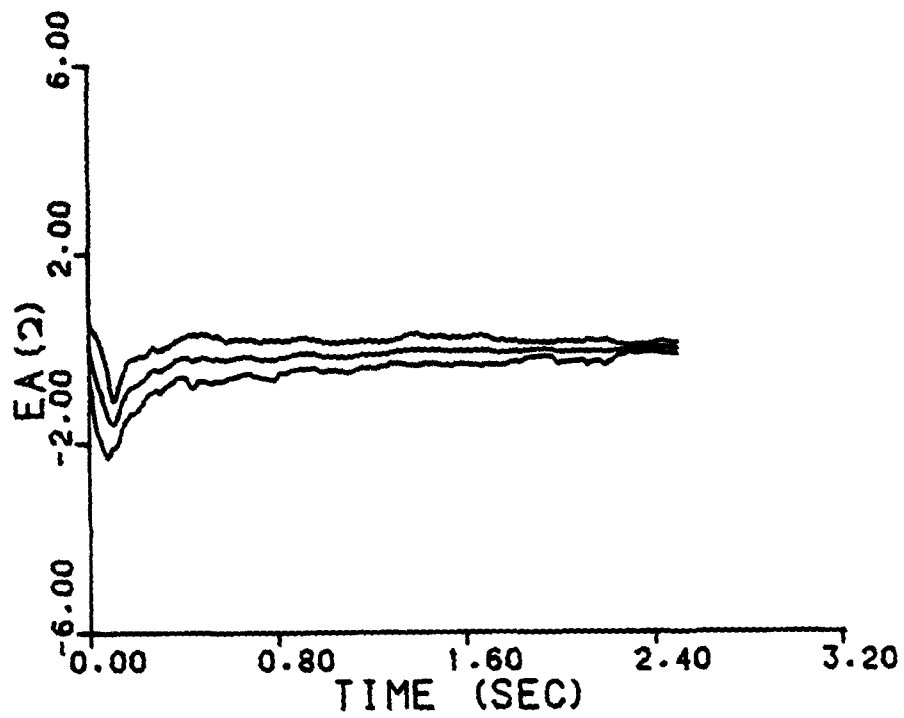


Figure B-4. Calculated probability weightings,
no contraction, parameter estimate error,
 $\mathbf{a}_t = (7,3)$.

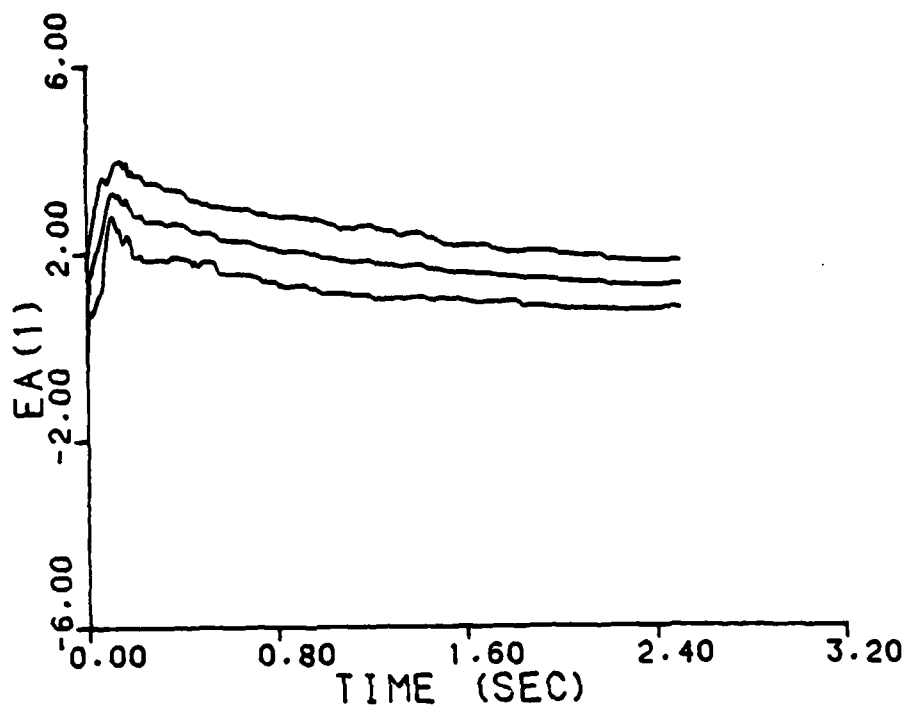
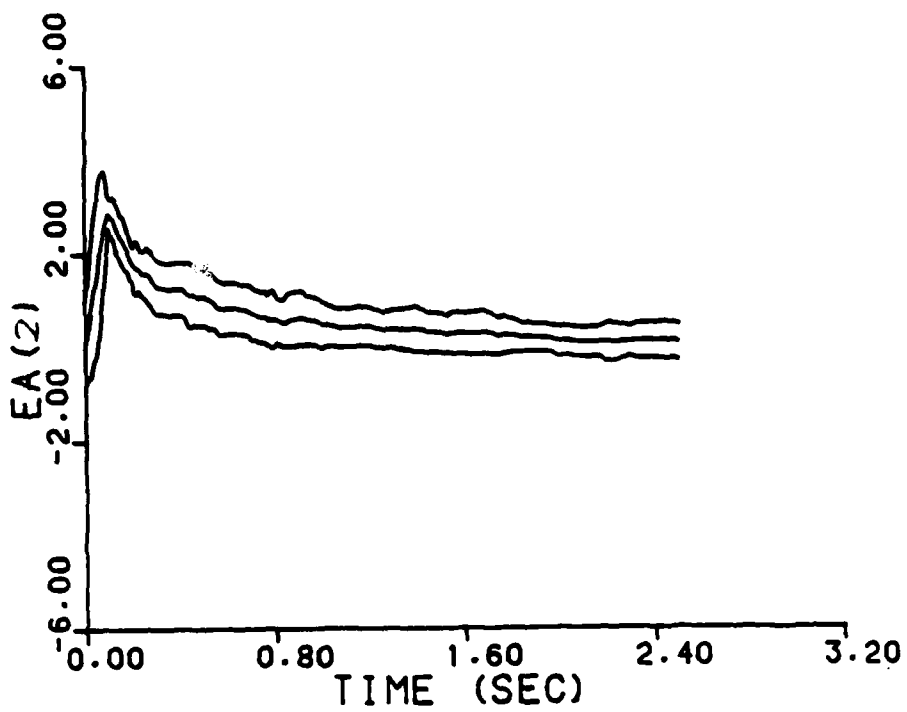


Figure B-5. Calculated probability weightings,
 no contraction, parameter estimate error,
 $\underline{a}_t = (8,8)$.

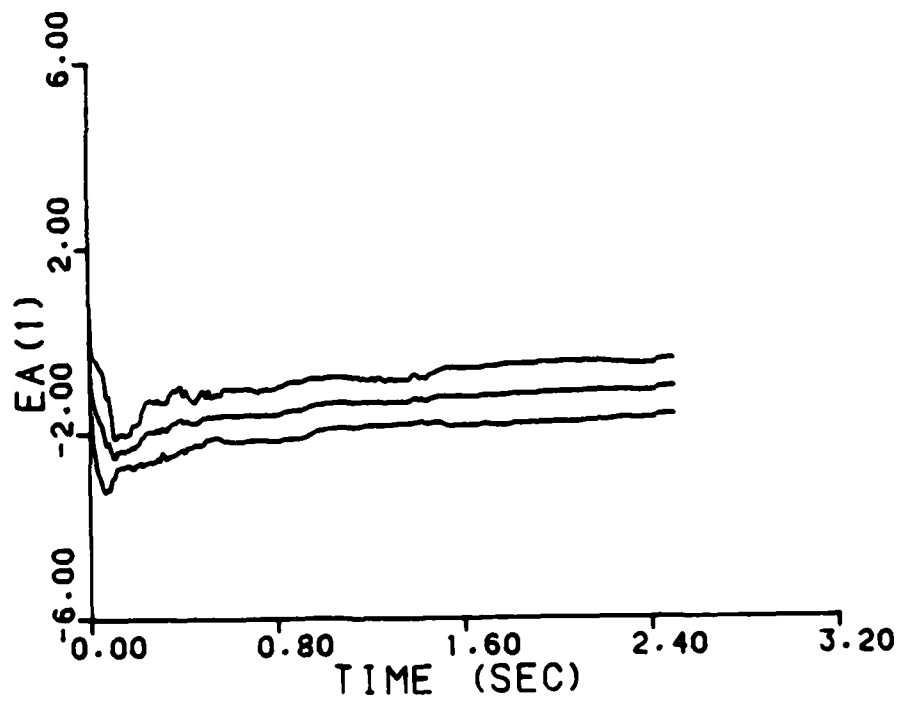
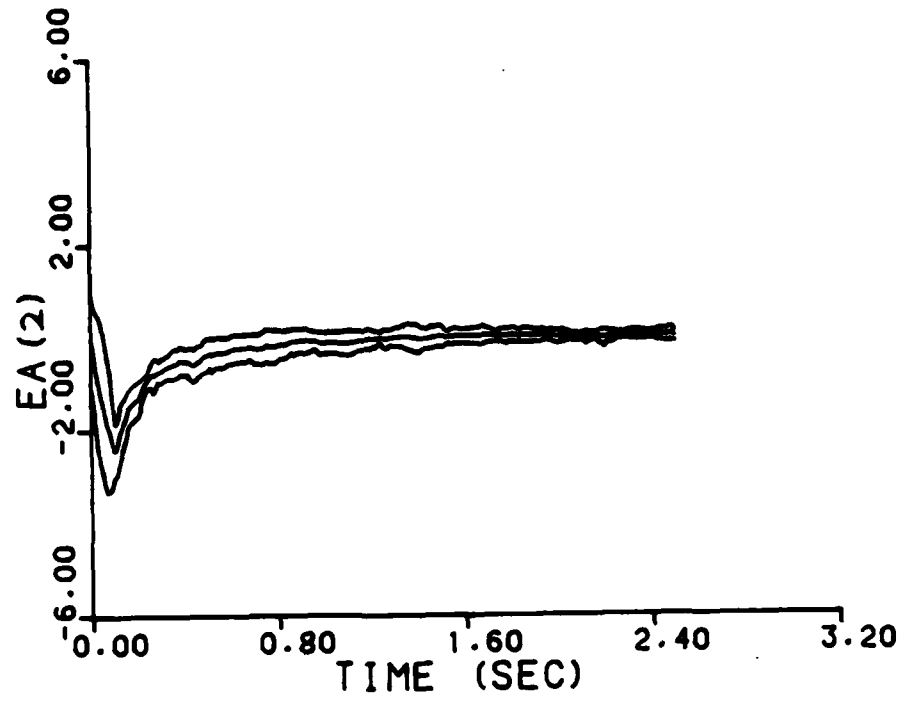


Figure B-6. Simplified probability weightings,
 no contraction, parameter estimate error,
 at = (2,2)

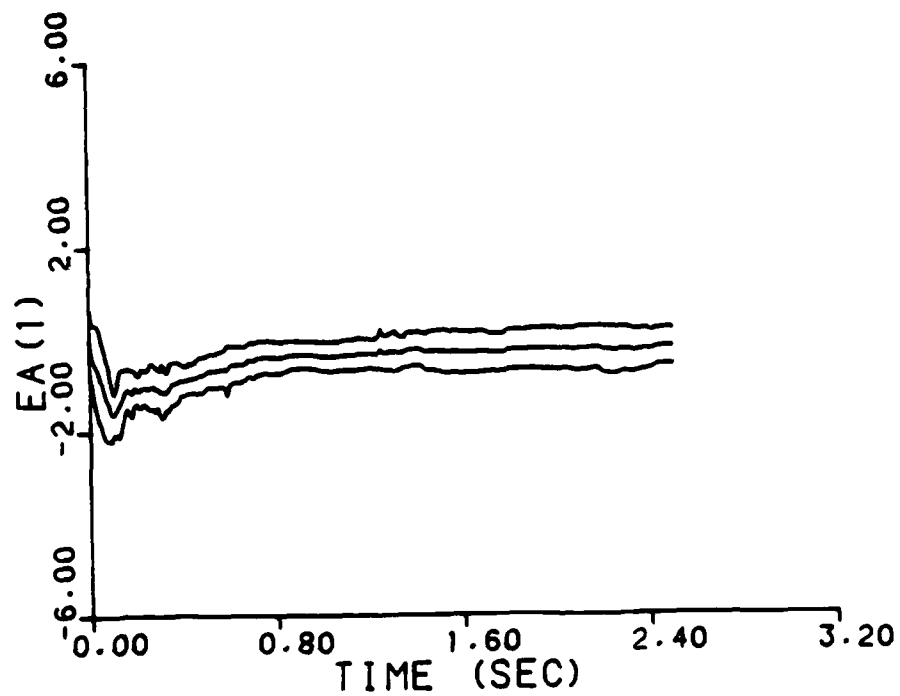
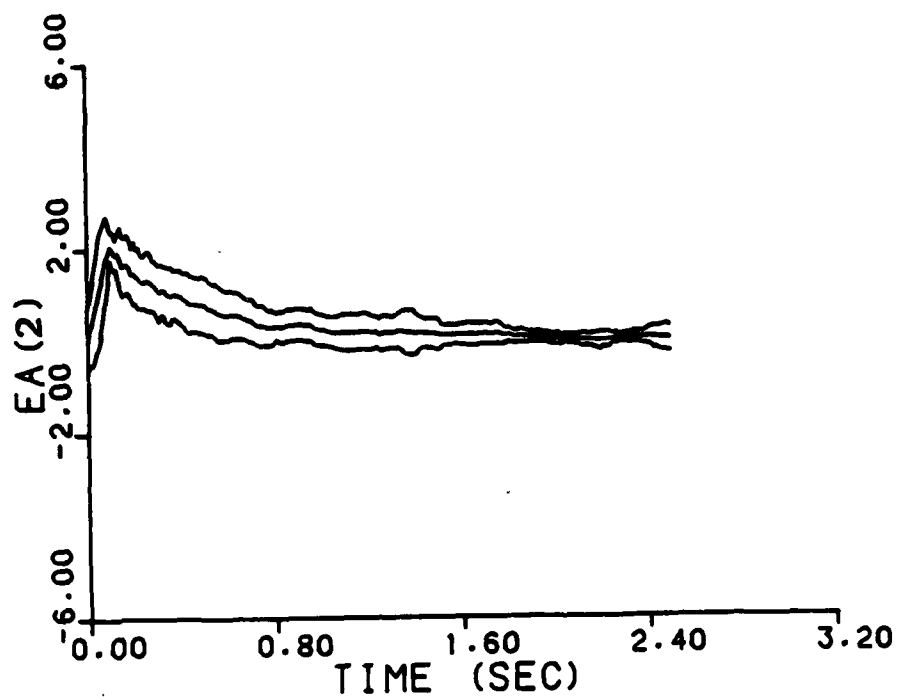


Figure B-7. Simplified probability weightings,
 no contraction, parameter estimate error,
 $\mathbf{a}_t = (3,7)$

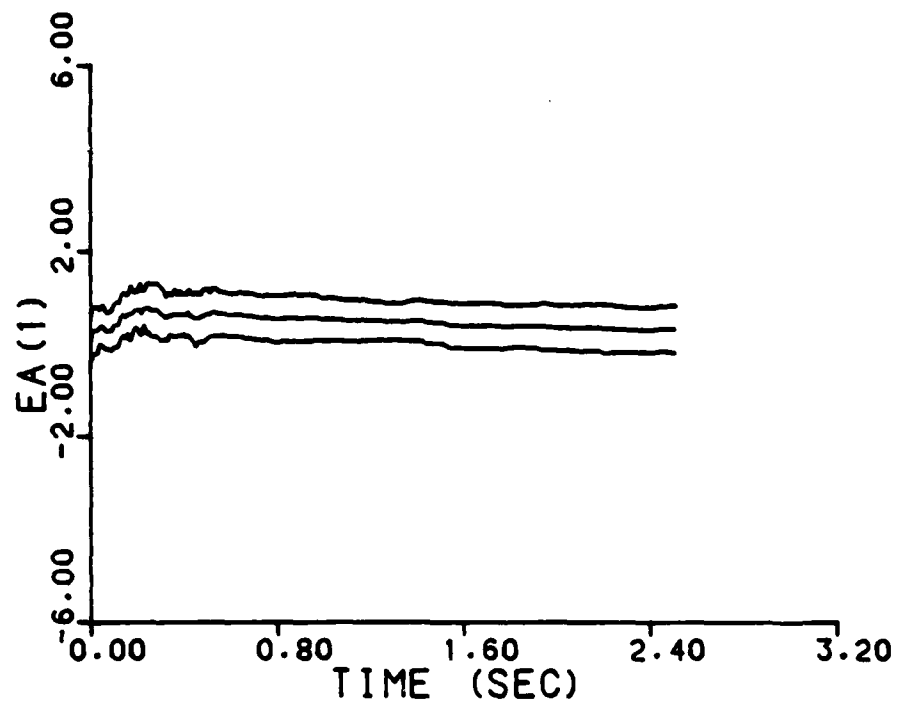
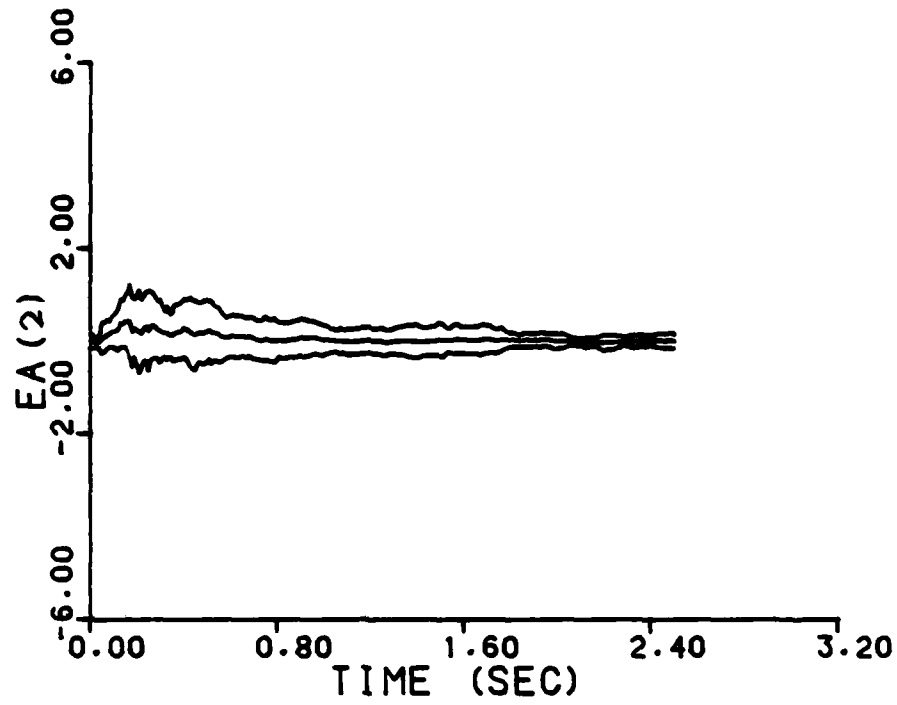


Figure B-8. Simplified probability weightings,
 no contraction, parameter estimate error,
 $\hat{a}_t = (5,5)$

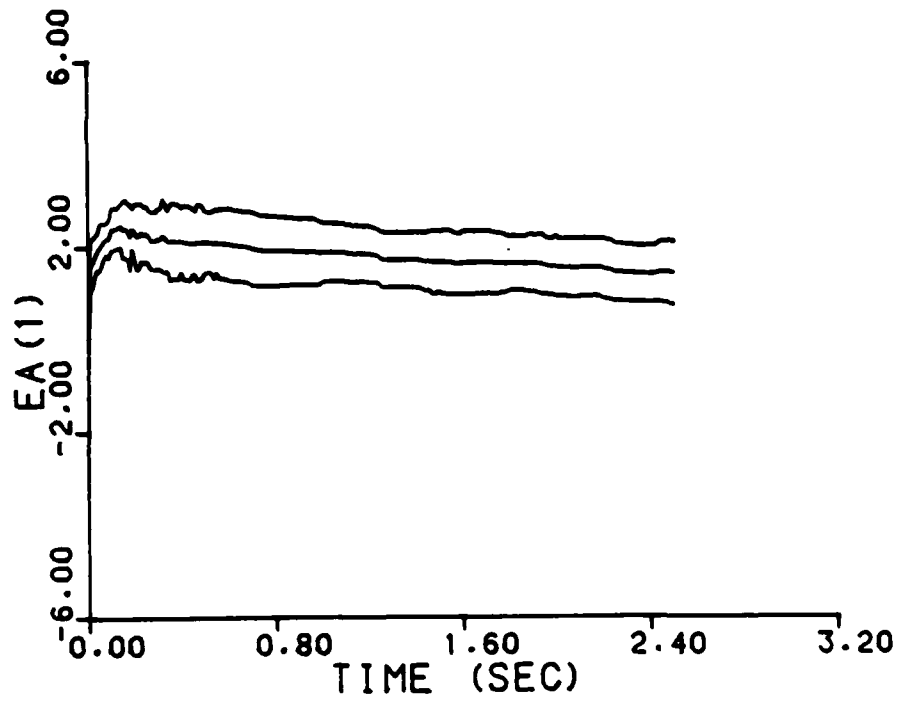
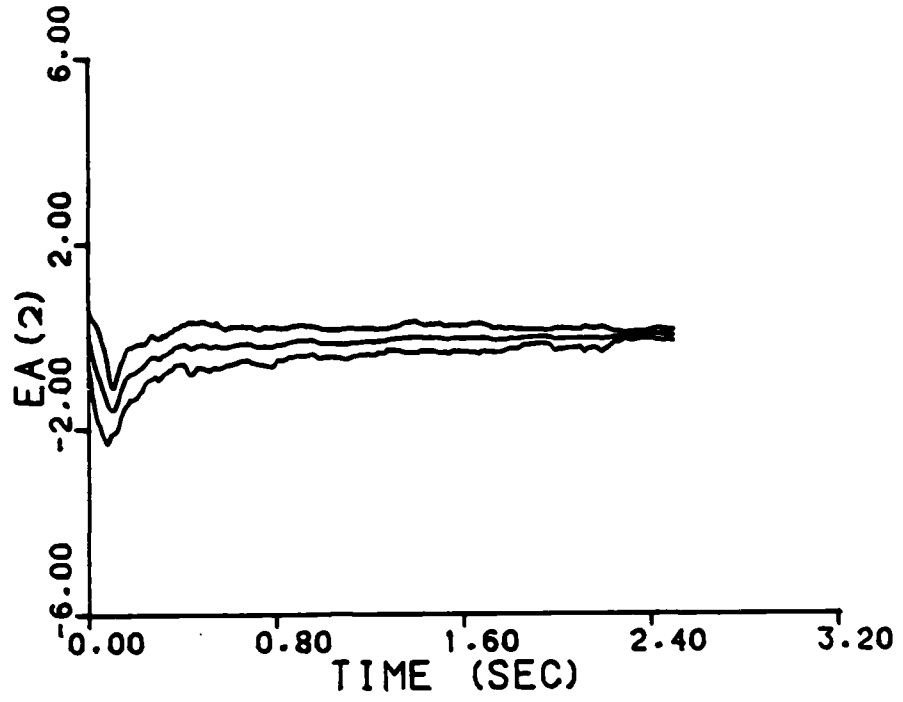


Figure B-9. Simplified probability weightings,
 no contraction, parameter estimate error,
 $\underline{a}_t = (7,3)$

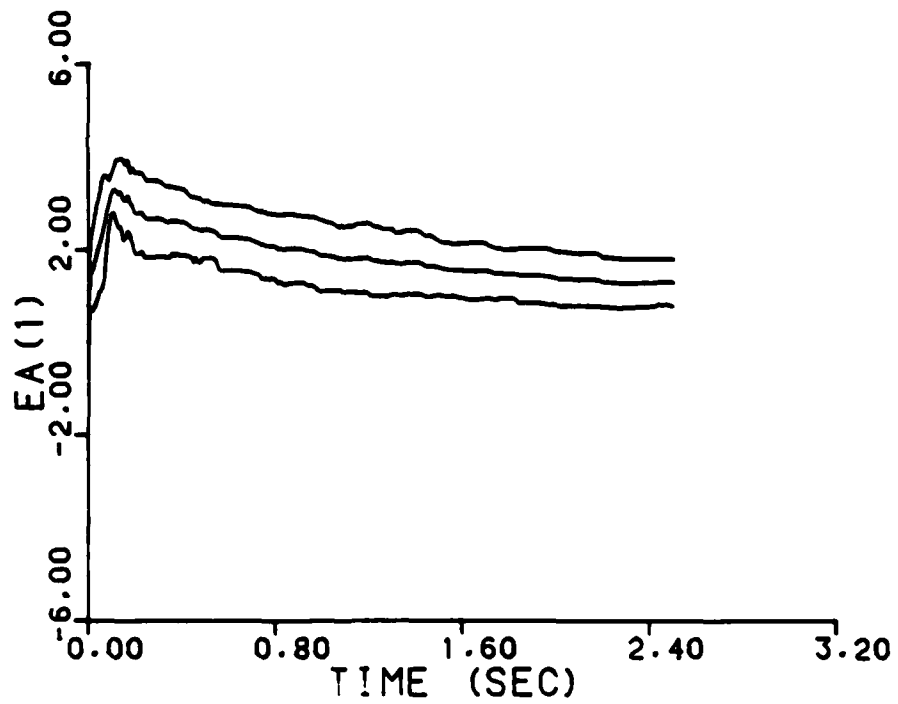
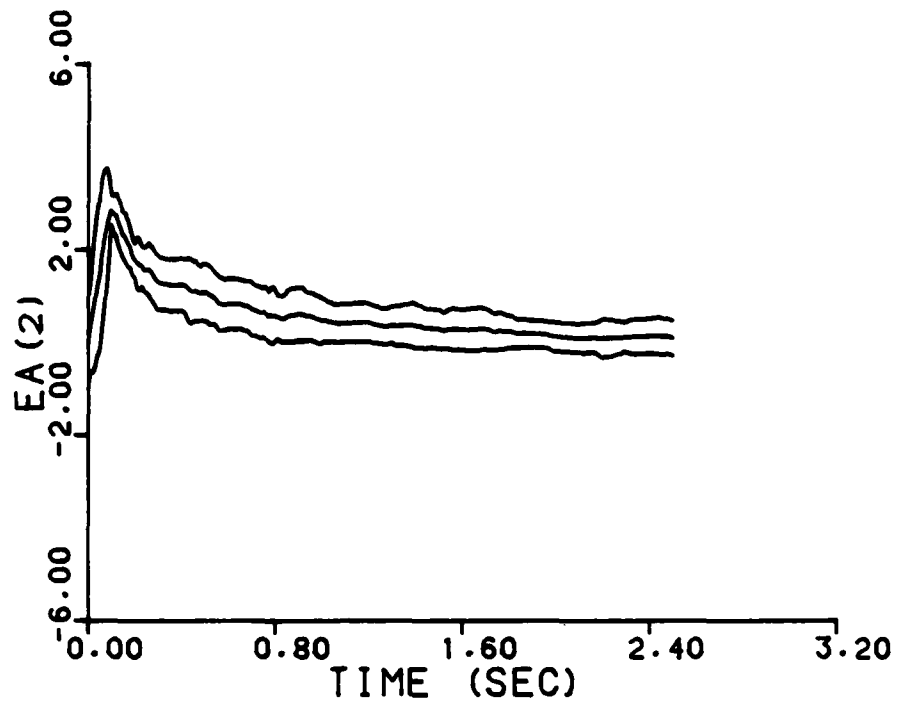
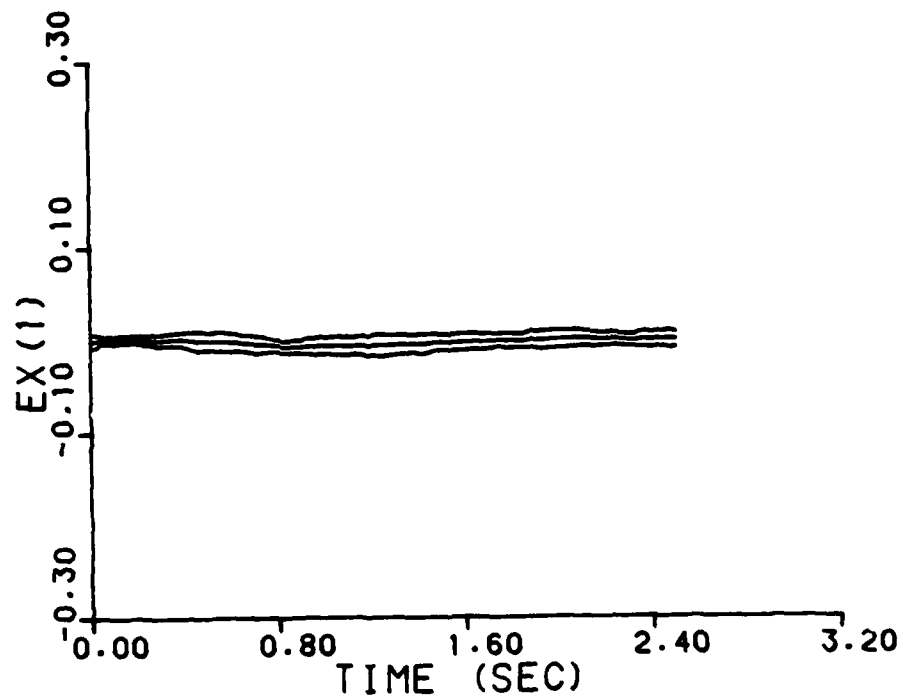
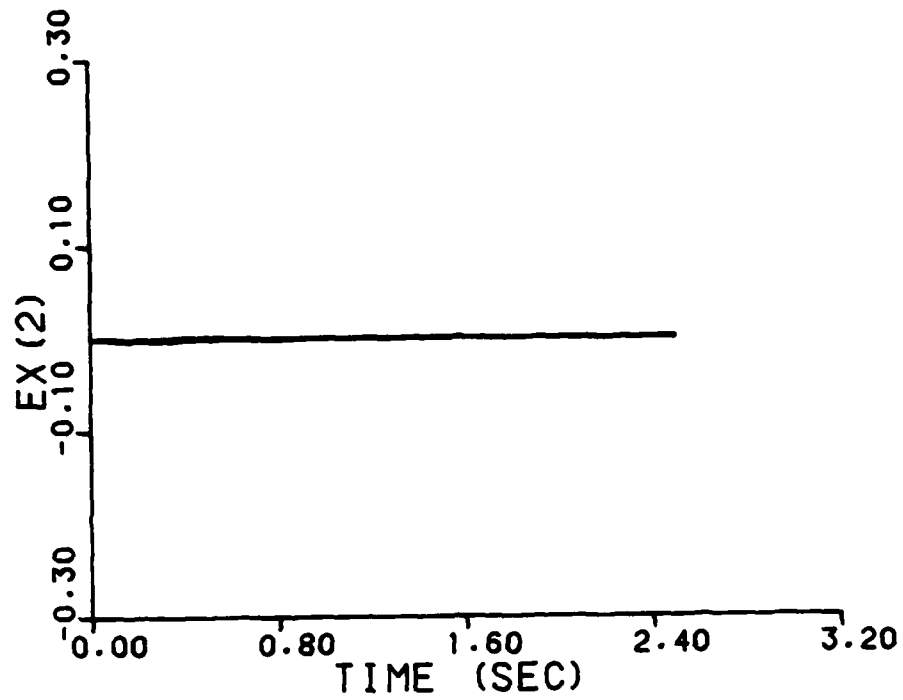
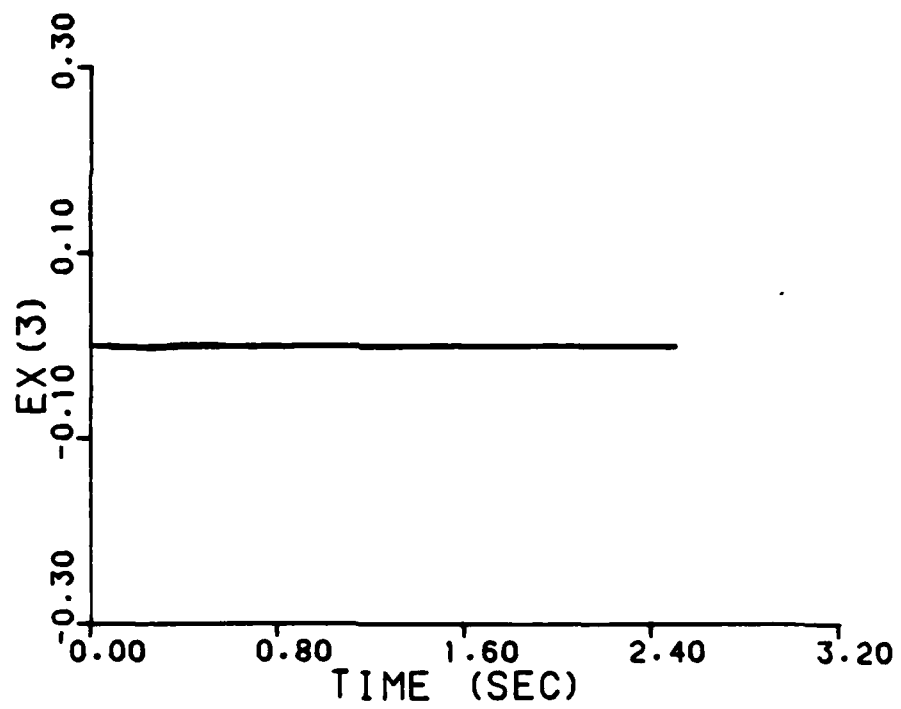
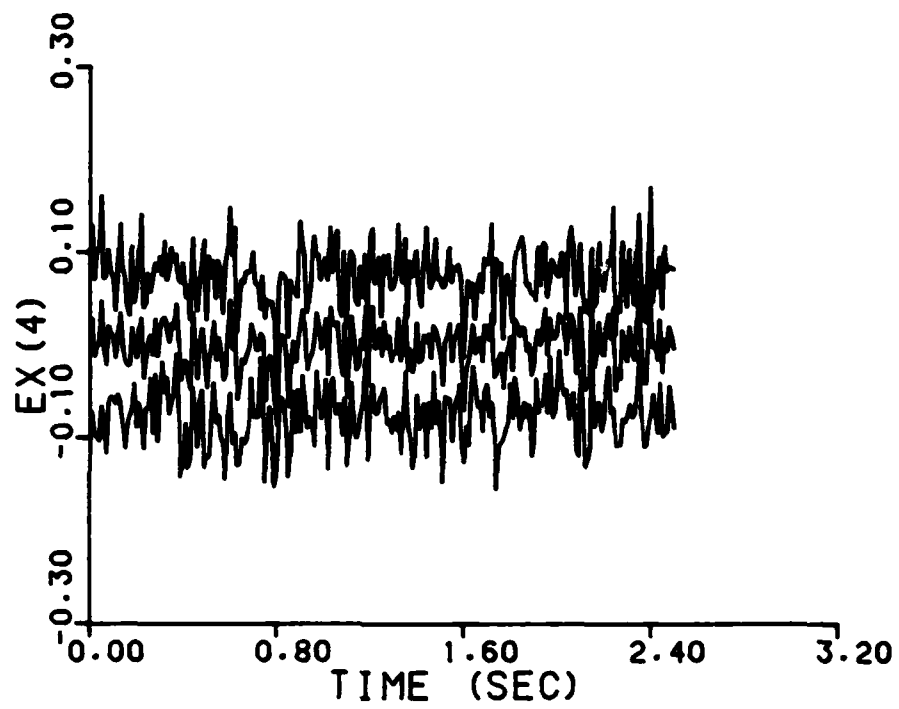


Figure B-10. Simplified probability weightings,
 no contraction, parameter estimate error,
 at = (8,8)



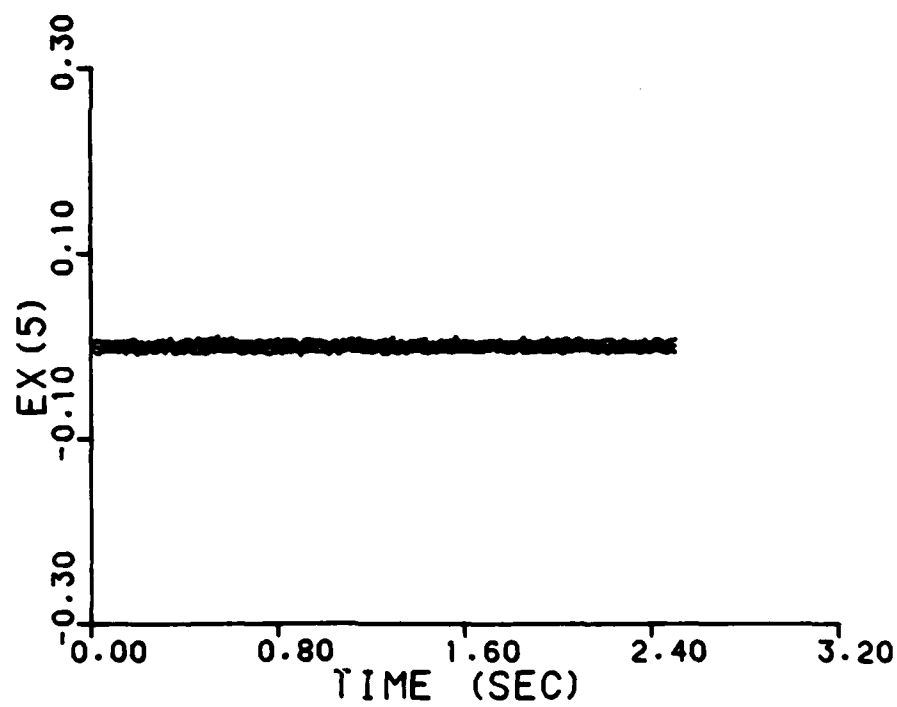
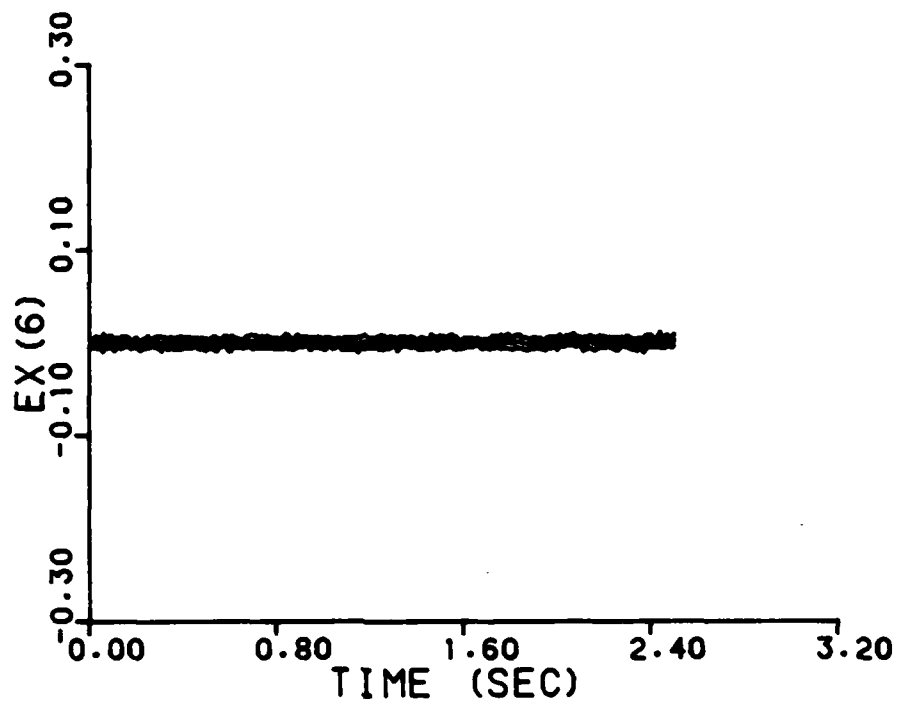
(a)

Figure B-11. Simplified probability weightings, no contraction, state estimate error, at = (3,7).



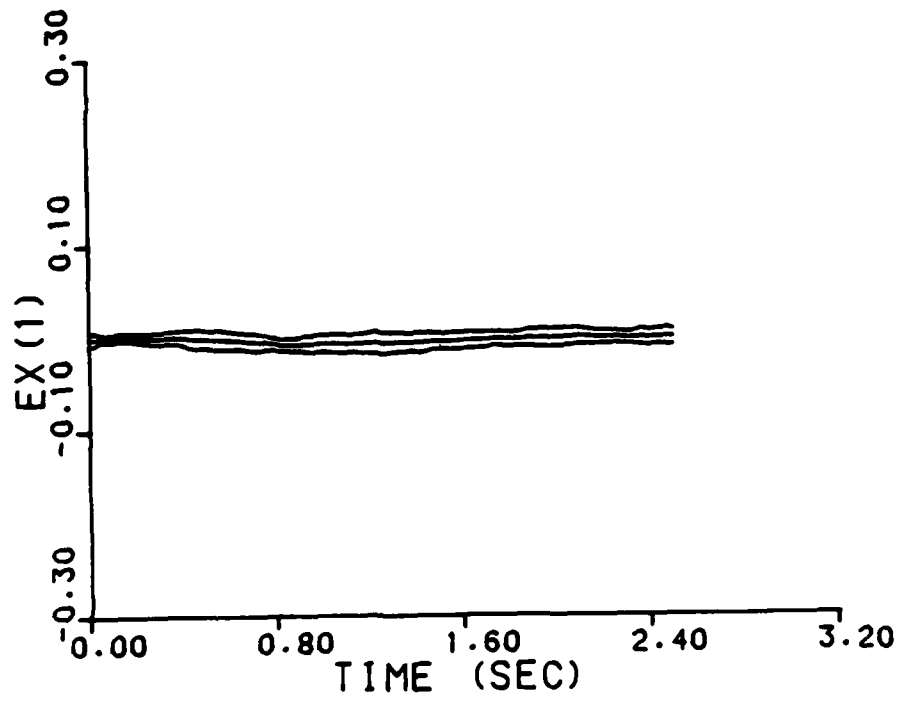
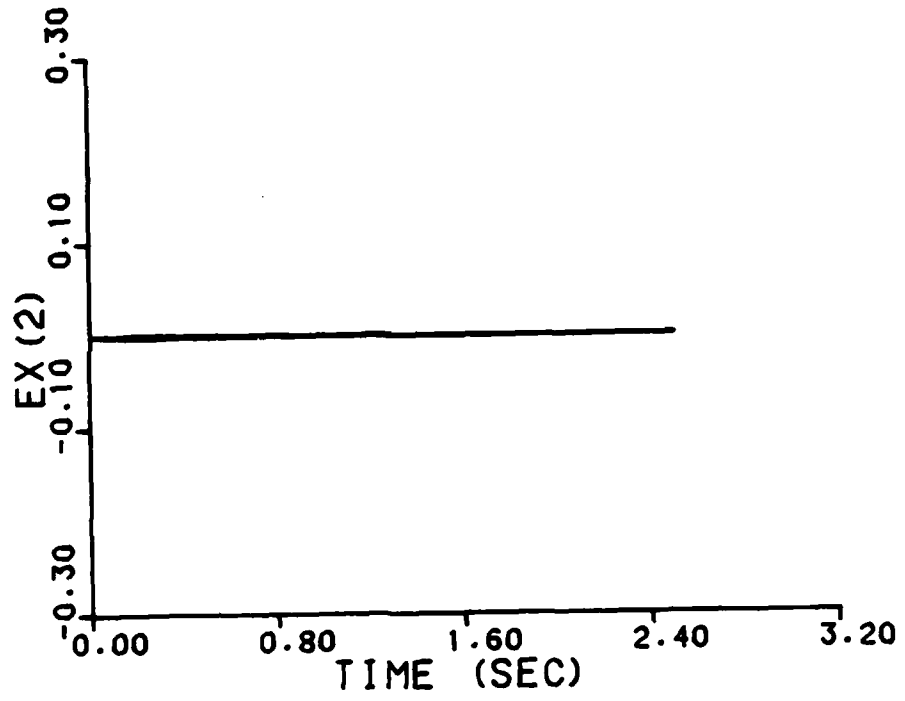
(b)

Figure B-11. (cont.)



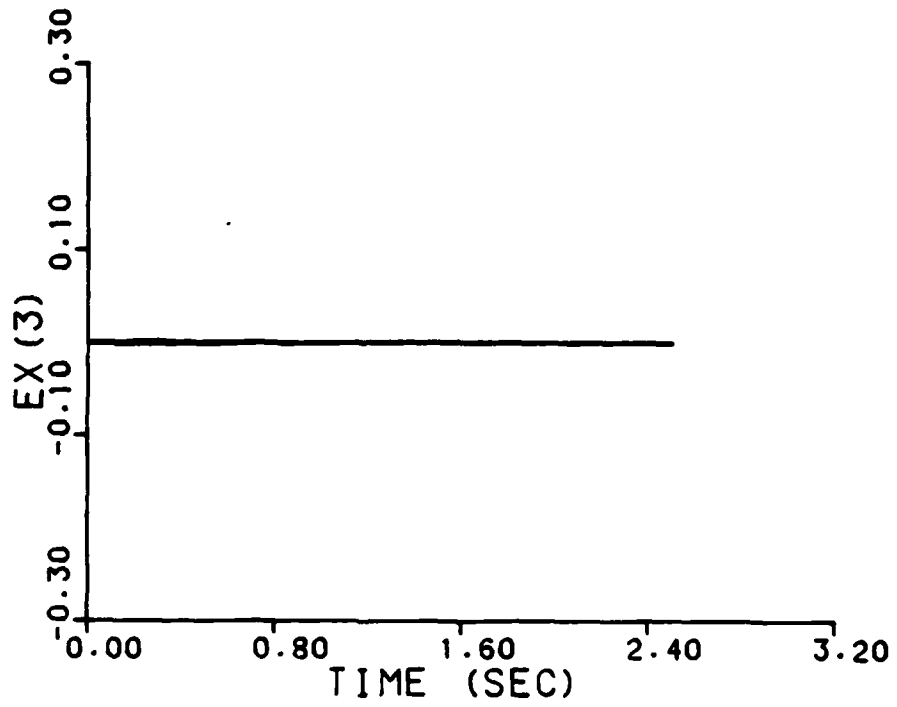
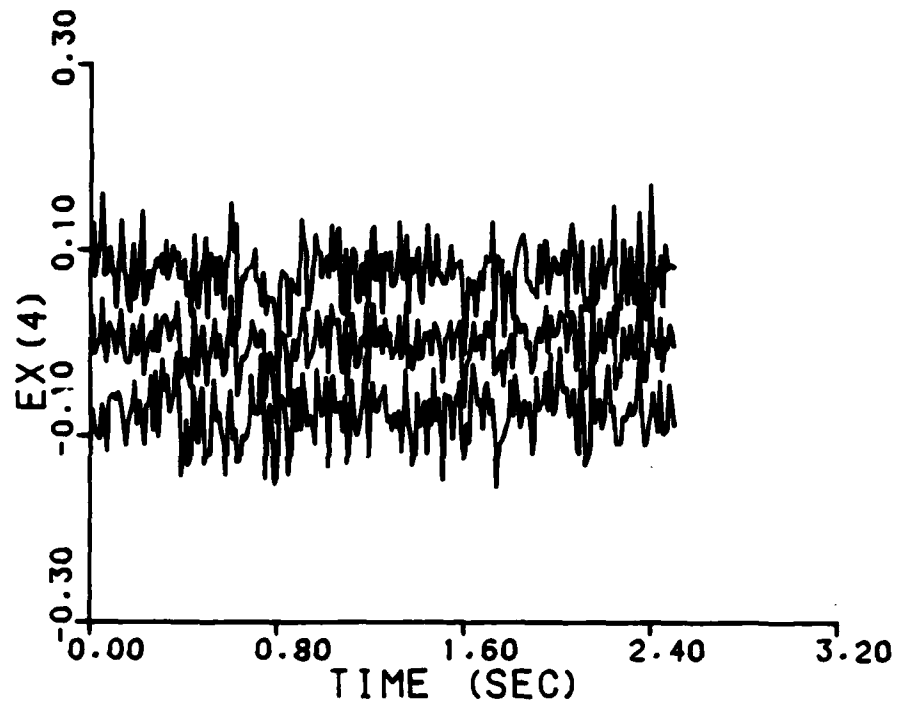
(c)

Figure B-11. (cont.)



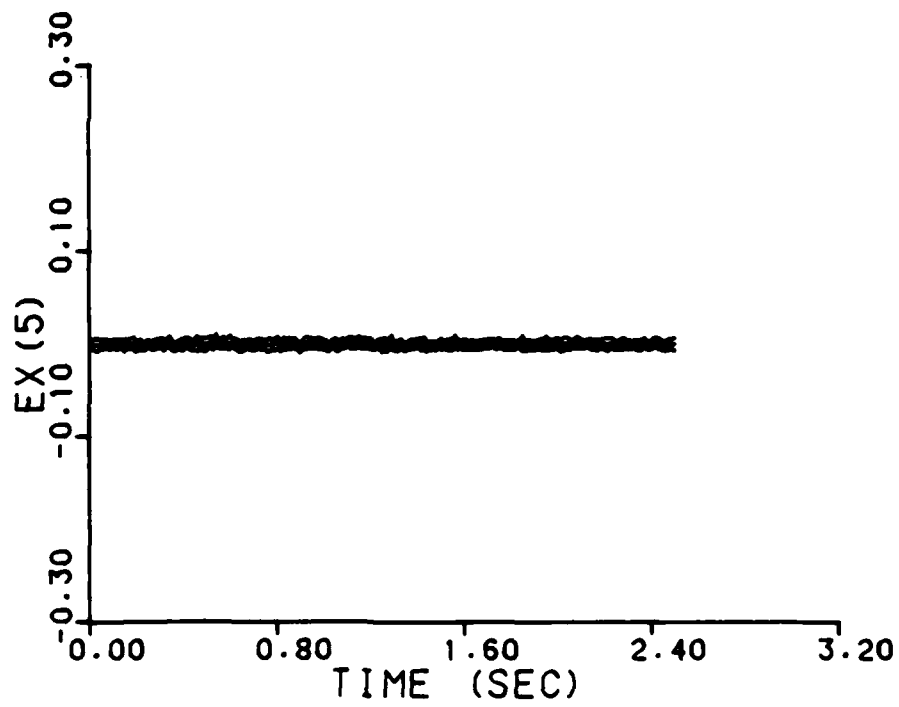
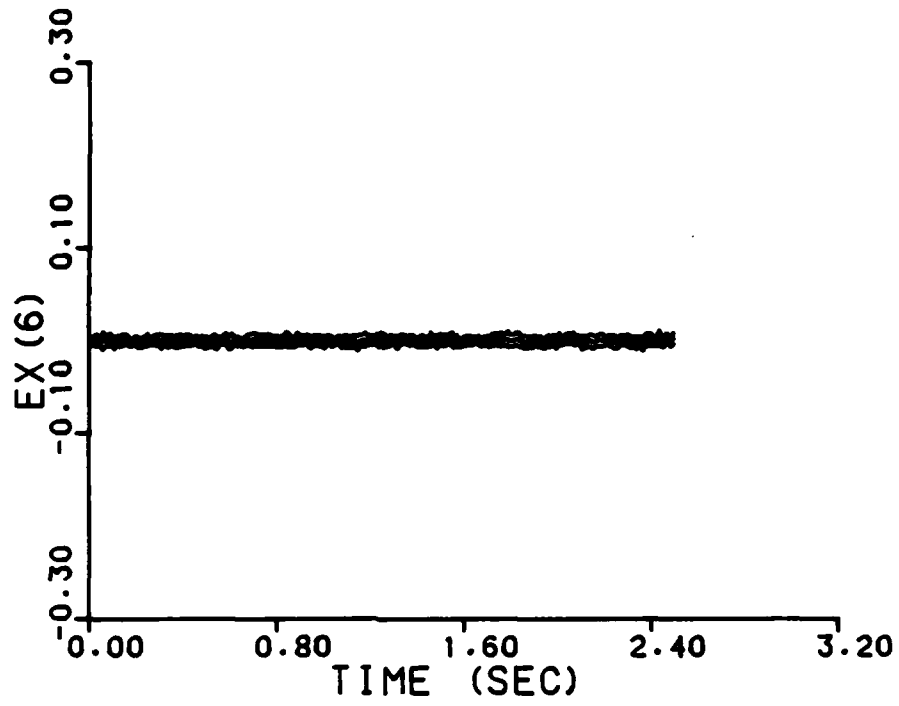
(a)

Figure B-12. Simplified probability weightings, no contraction, state estimate error, at $\mathbf{x} = (8,8)$.



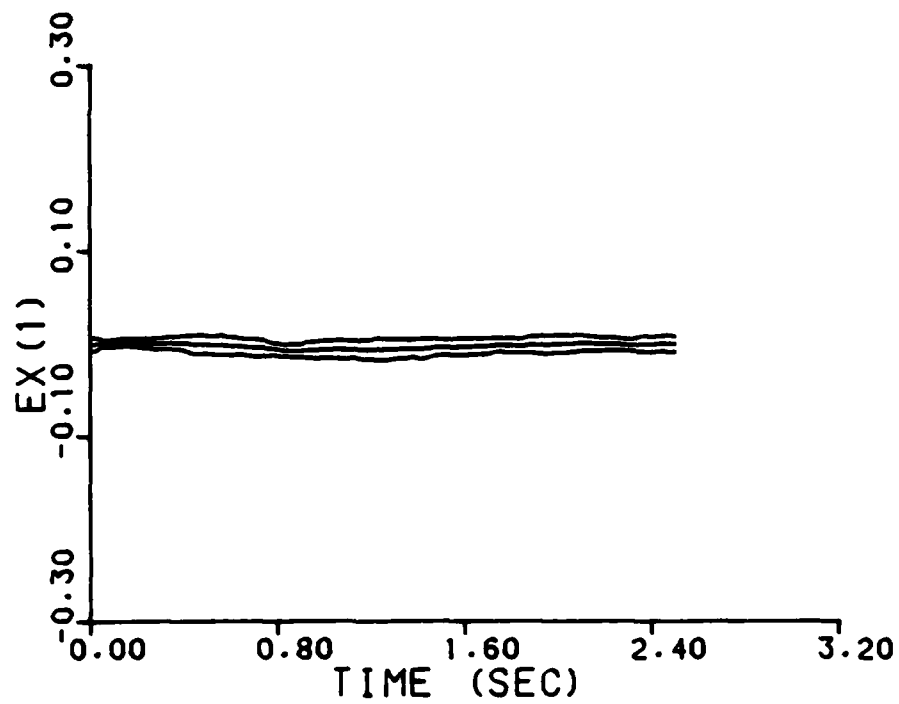
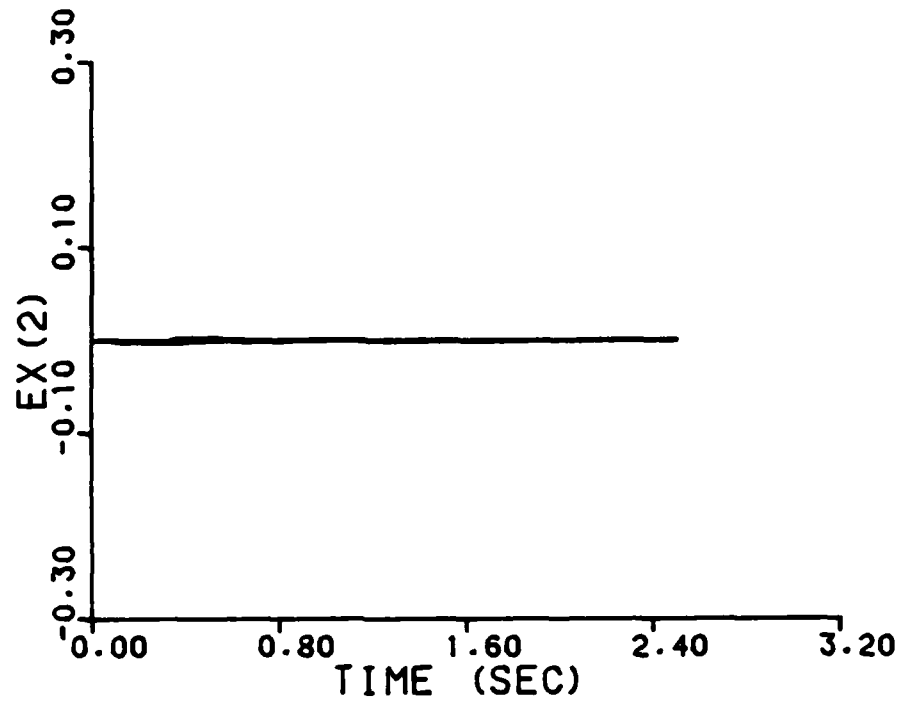
(b)

Figure B-12. (cont.)



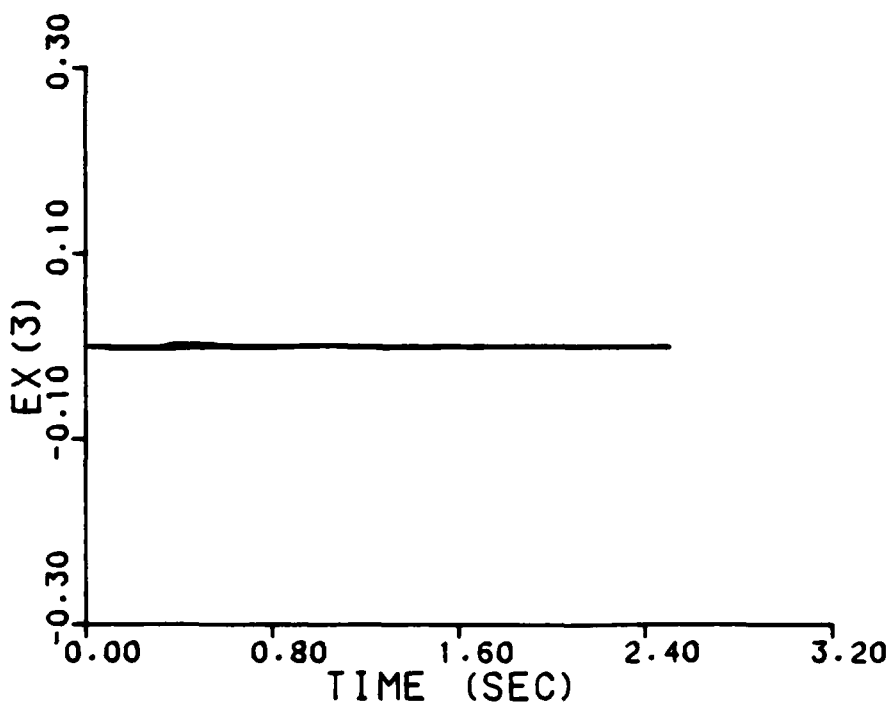
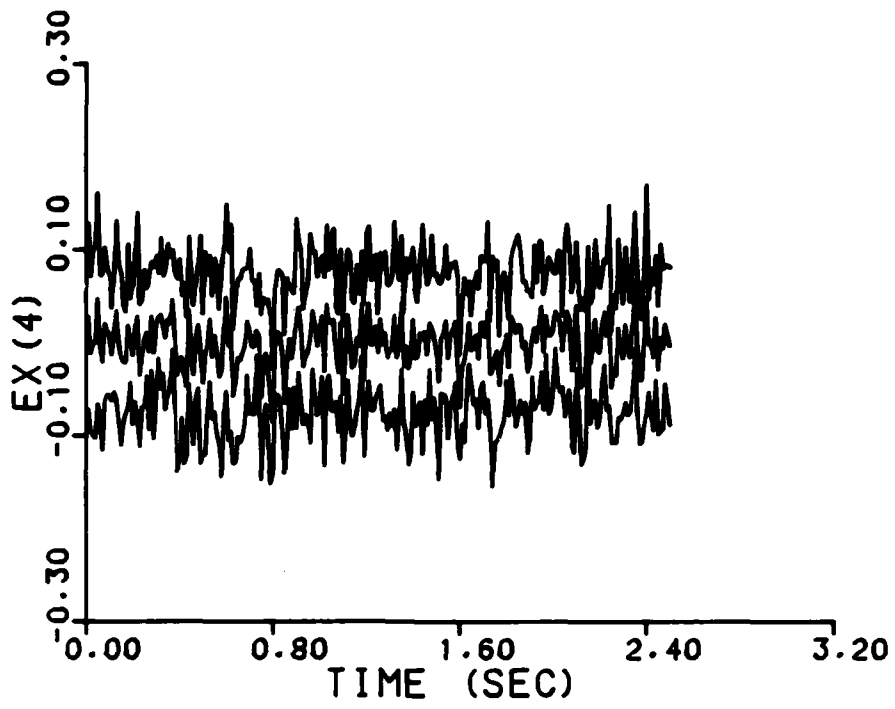
(c)

Figure B-12. (cont.)



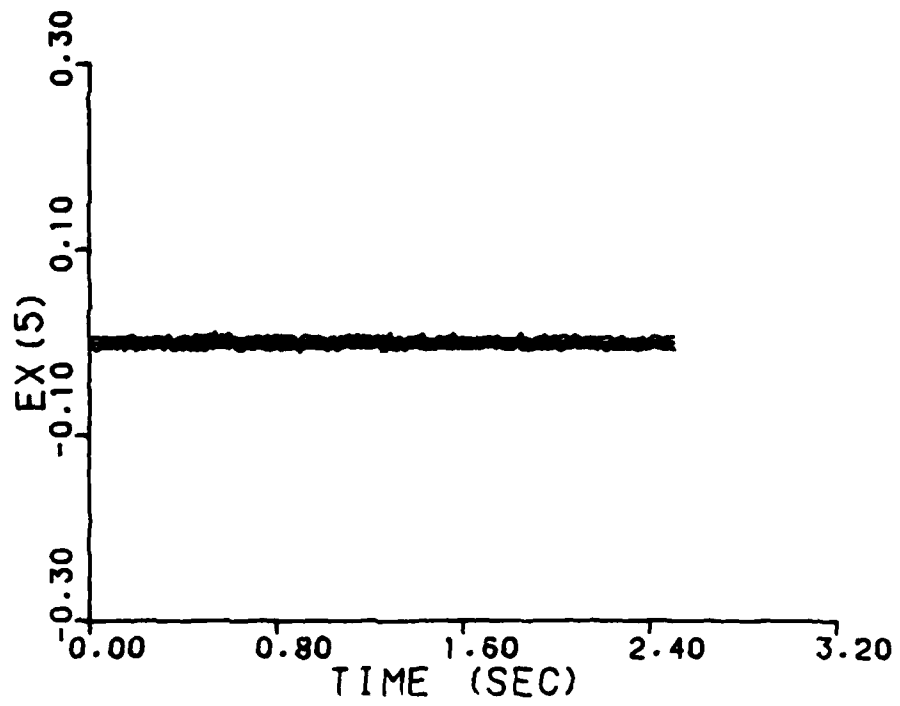
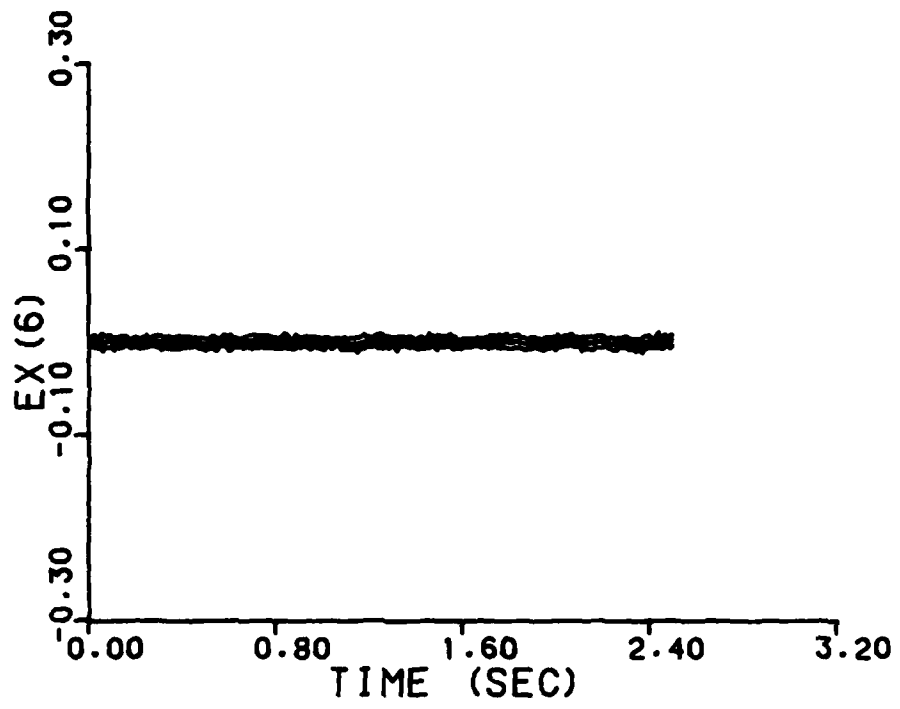
(a)

Figure B-13. Single filter at (5,5)
state estimate error,
at = (3,7).



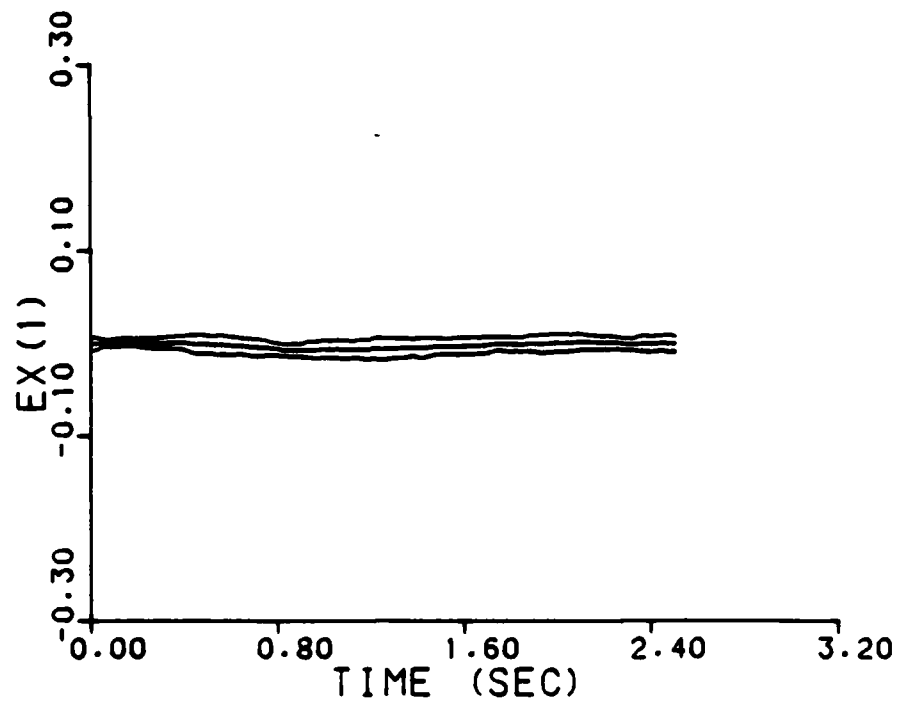
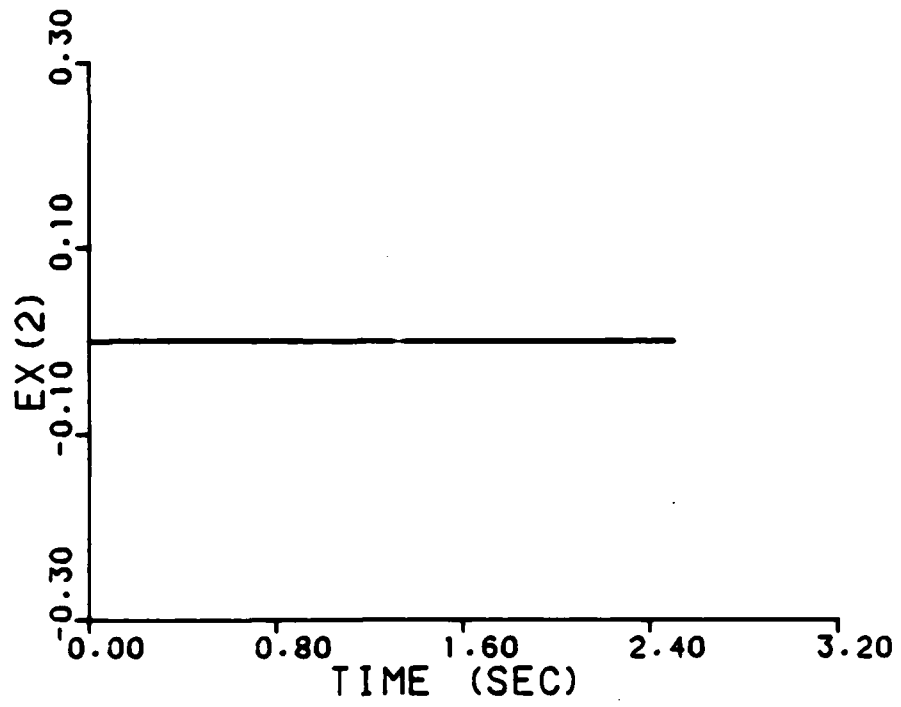
(b)

Figure B-13. (cont.)



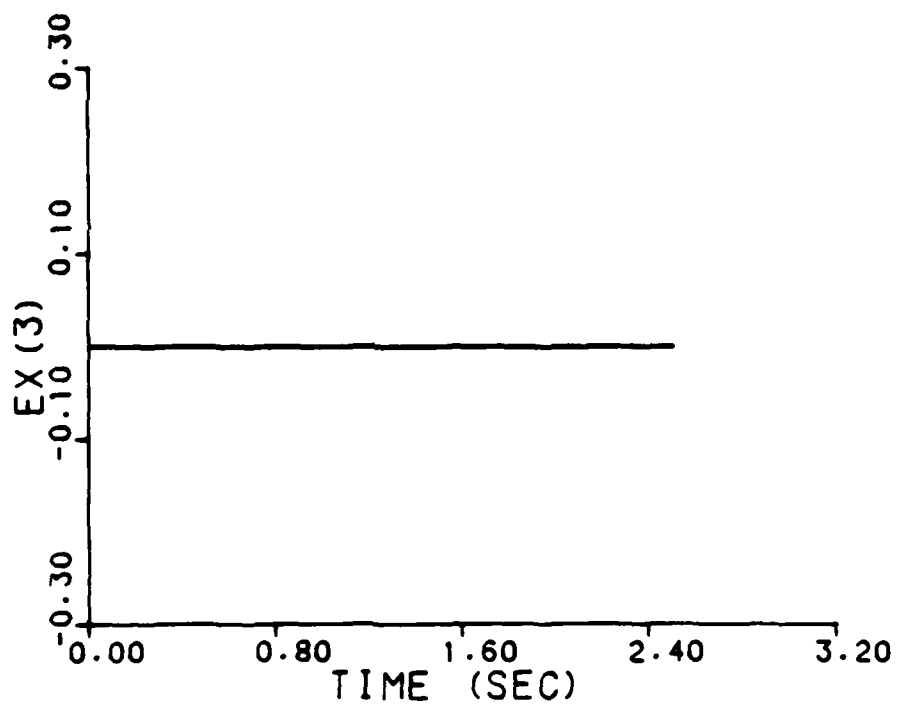
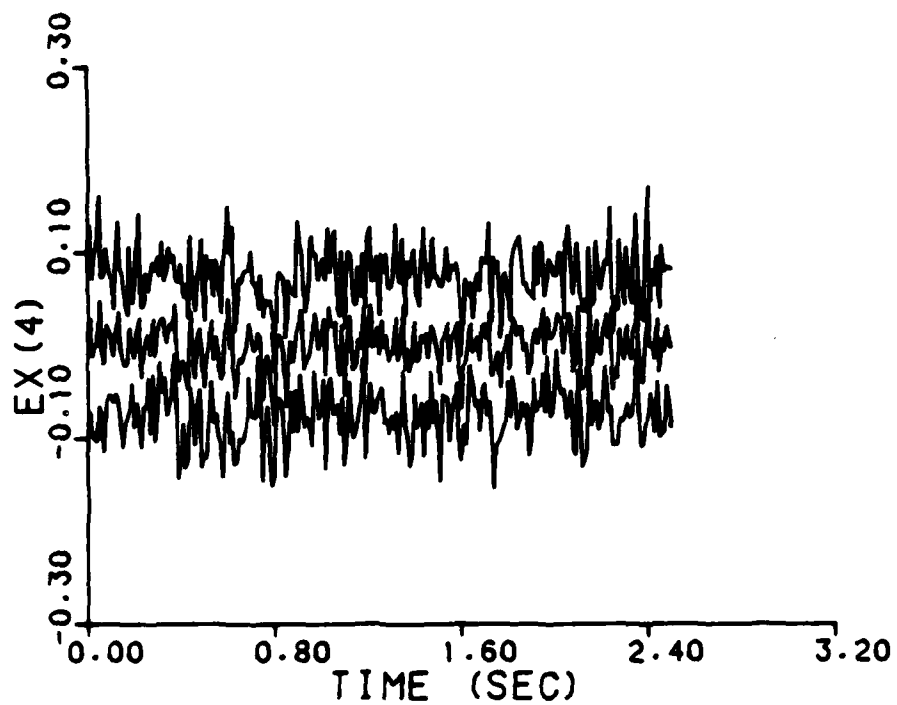
(c)

Figure B-13. (cont.)



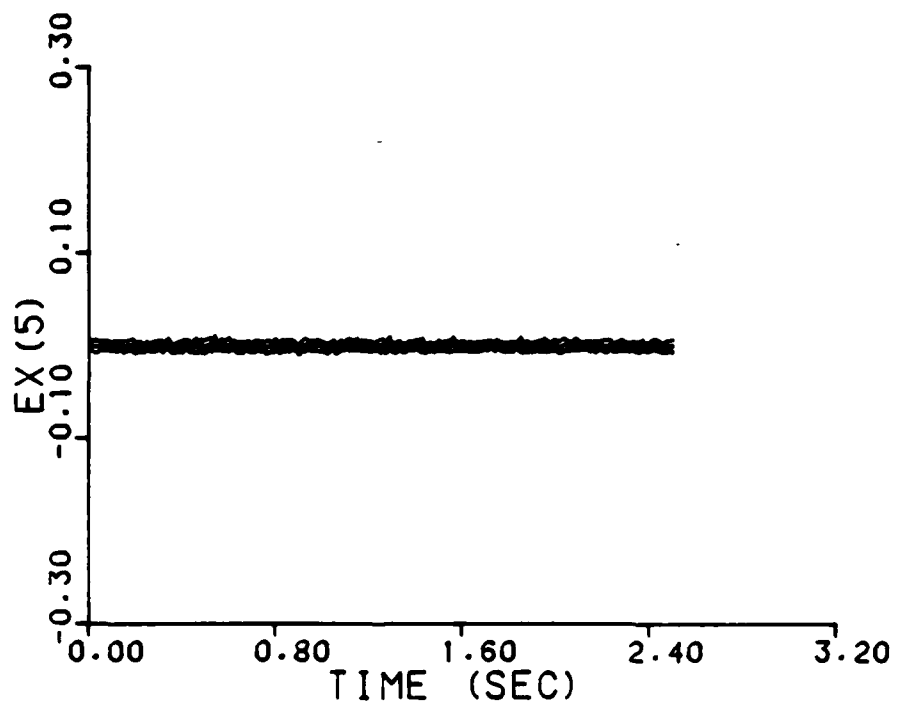
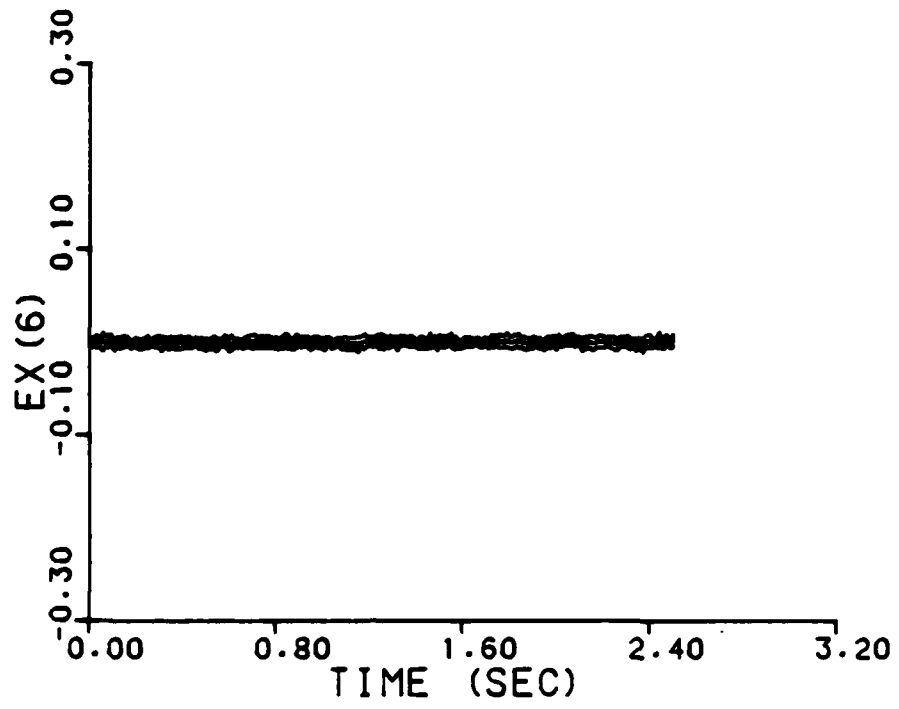
(a)

Figure B-14. Single filter at (5,5)
state estimate error,
at = (5,5).



(b)

Figure B-14. (cont.)



(c)

Figure B-14. (cont.)

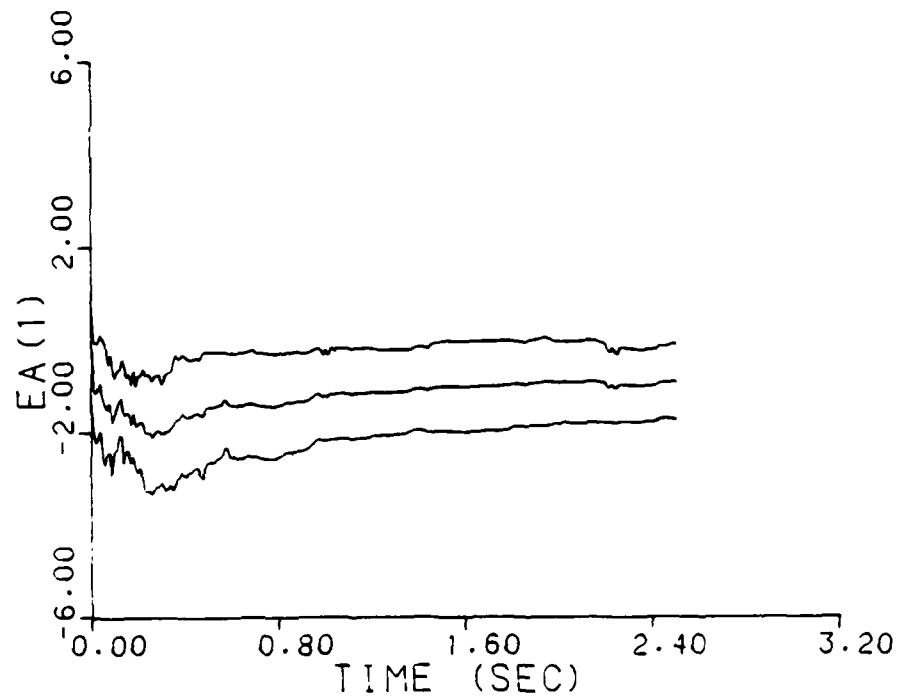
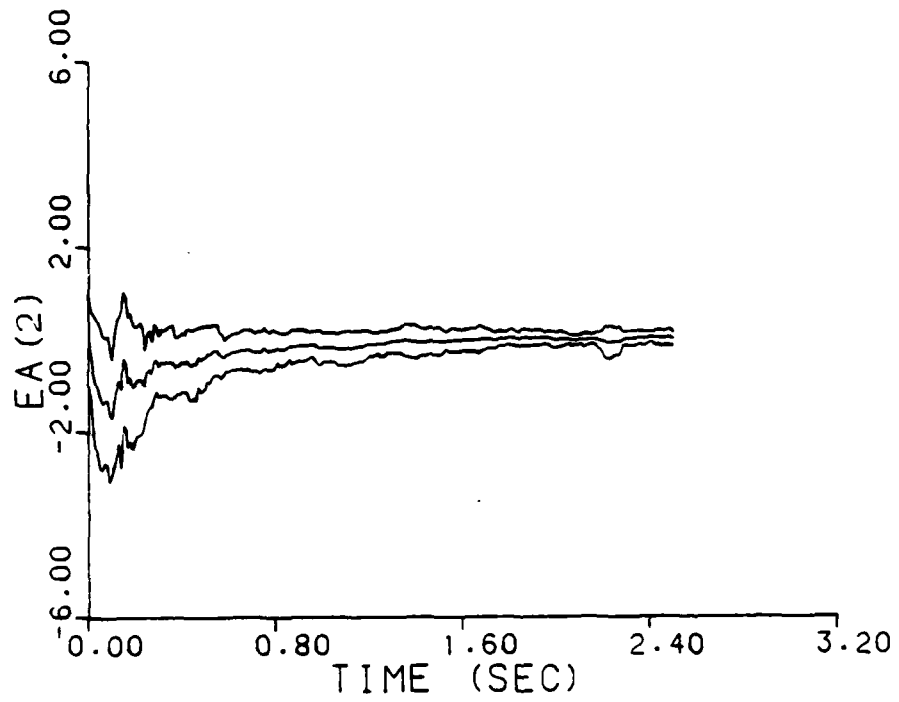


Figure B-15. Simplified probability weightings, with contraction, parameter estimate error, at = (2,2).

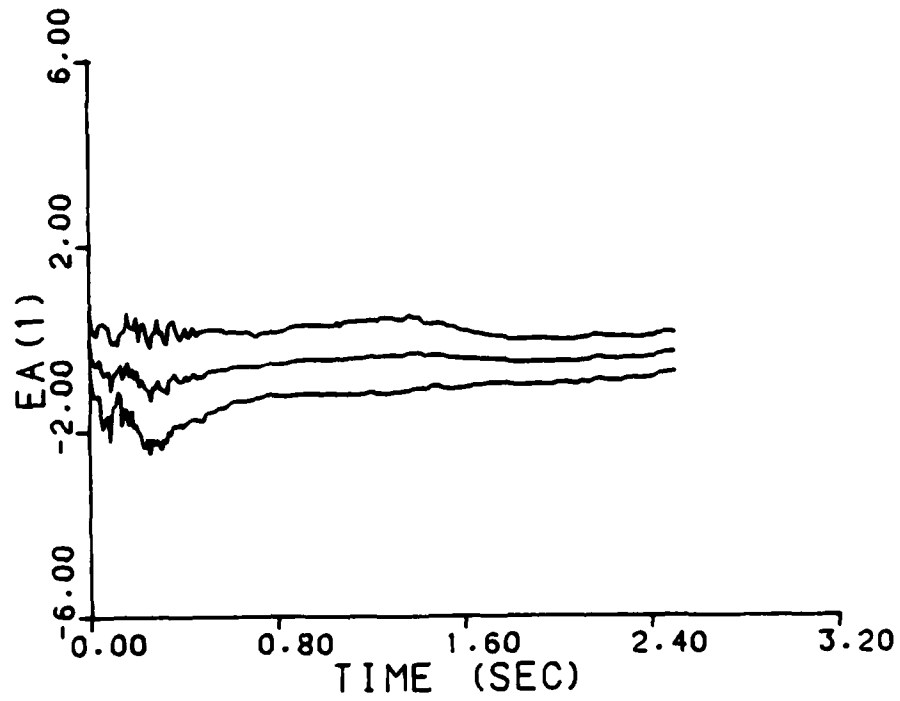
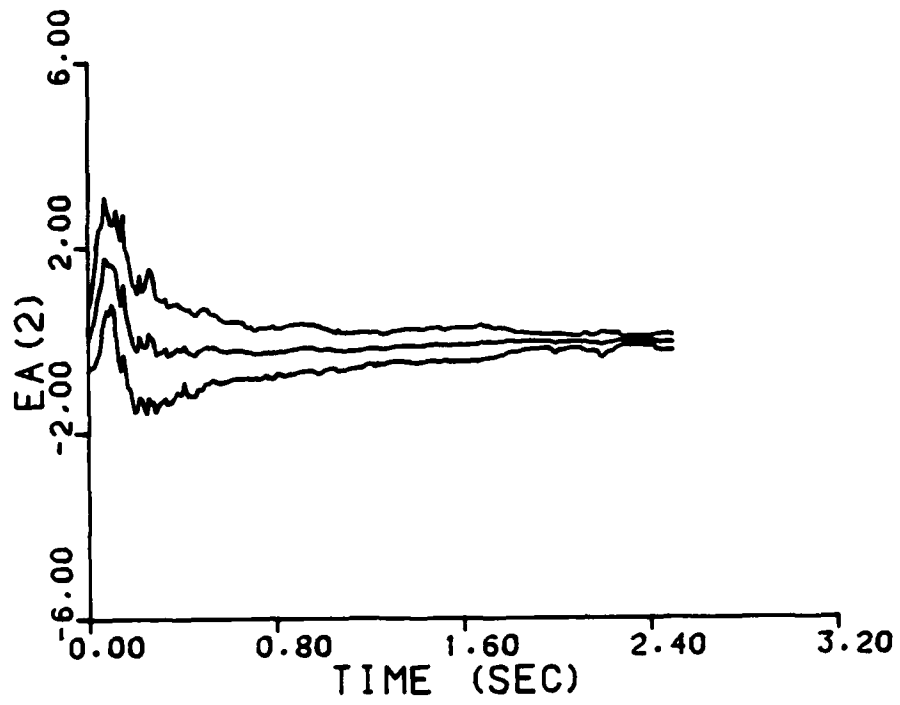


Figure B-16. Simplified probability weightings, with contraction, parameter estimate error, at = (3,7).

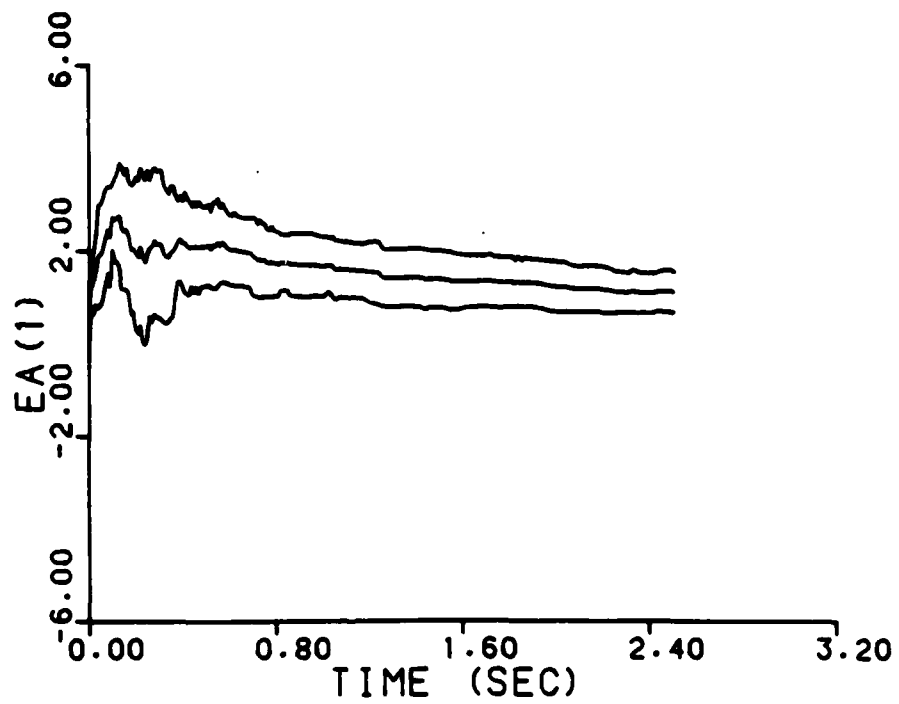
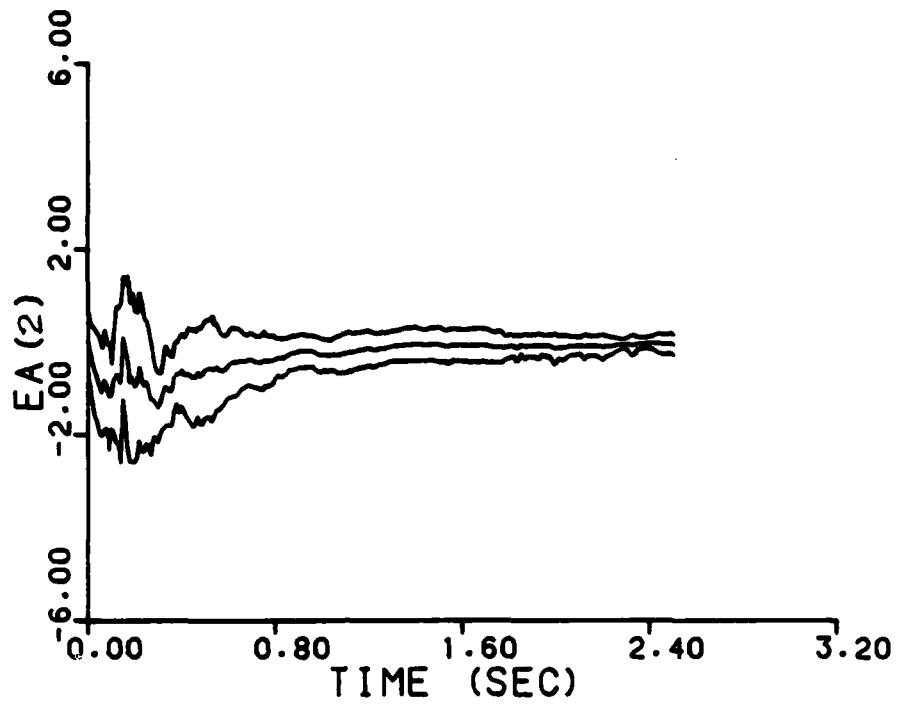


Figure B-17. Simplified probability weightings, with contraction, parameter estimate error, $a_t = (7,3)$.

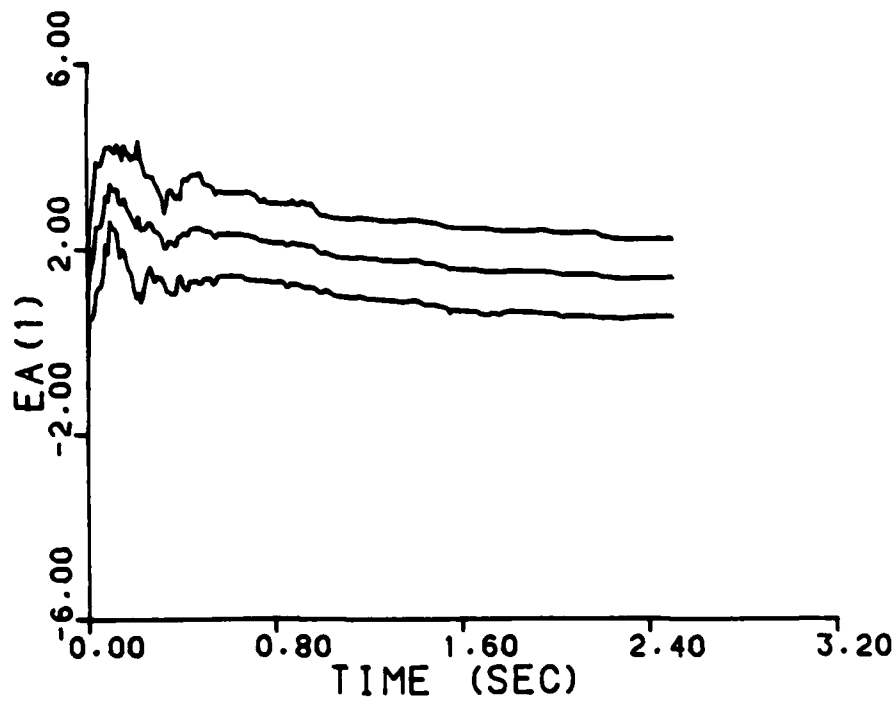
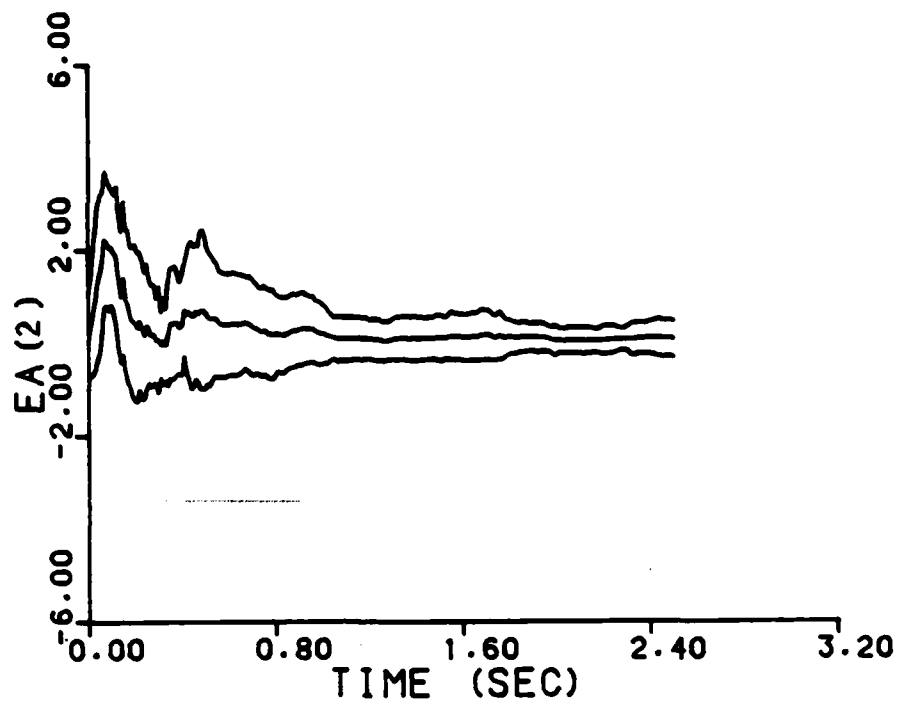


Figure B-18. Simplified probability weightings, with contraction, parameter estimate error, at $\theta = (8,8)$.

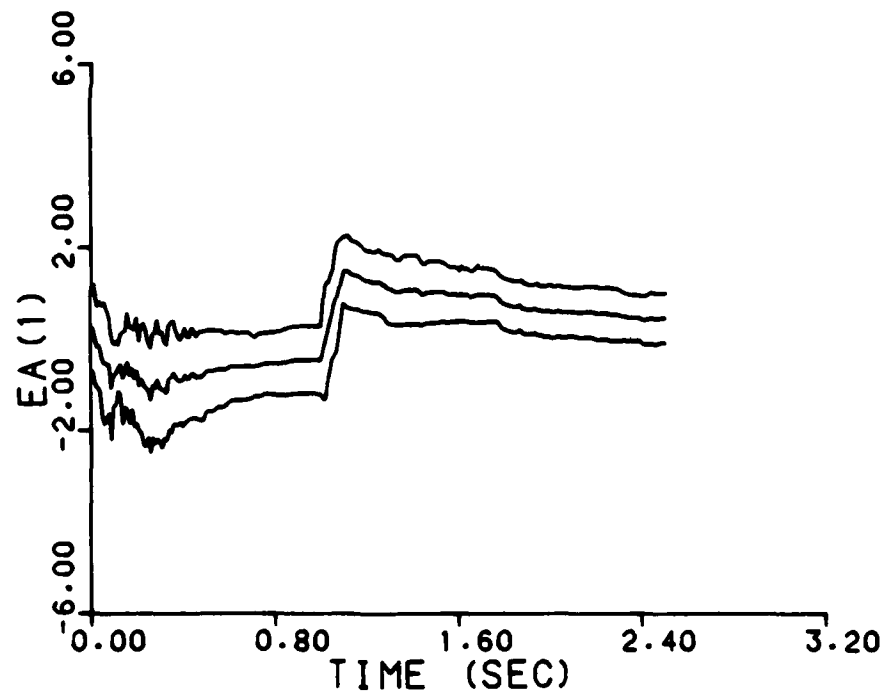
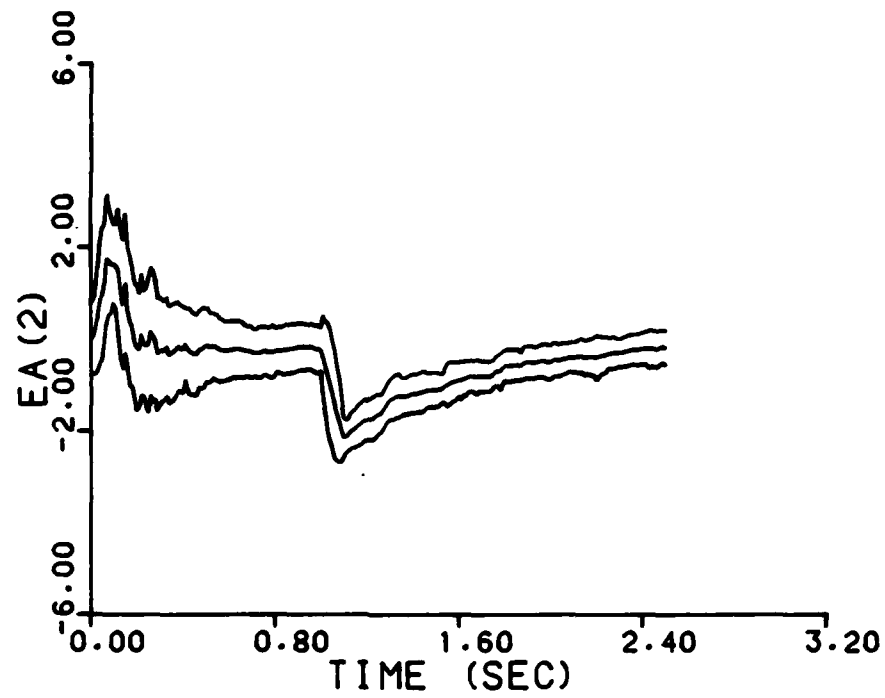
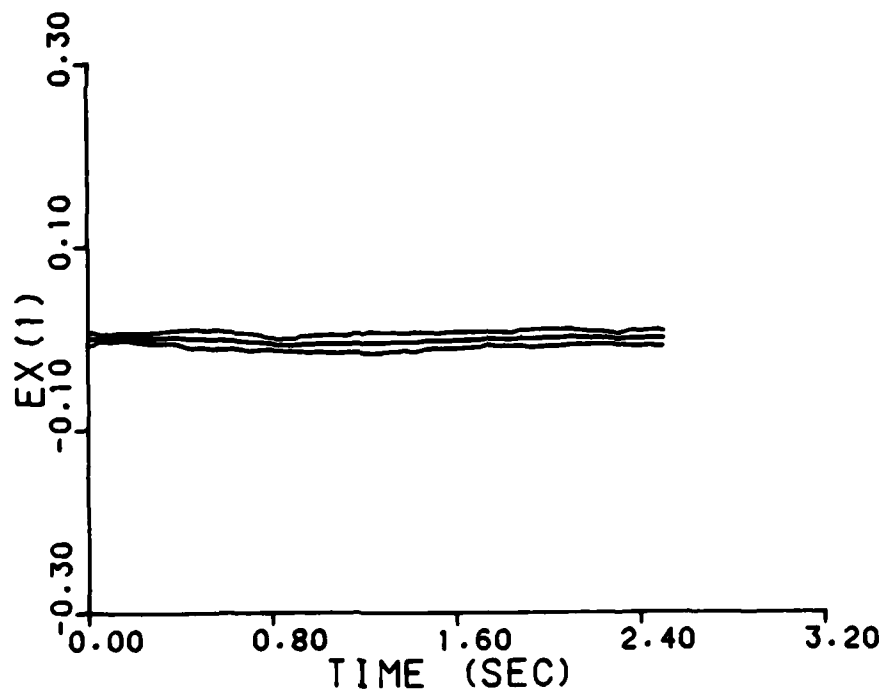
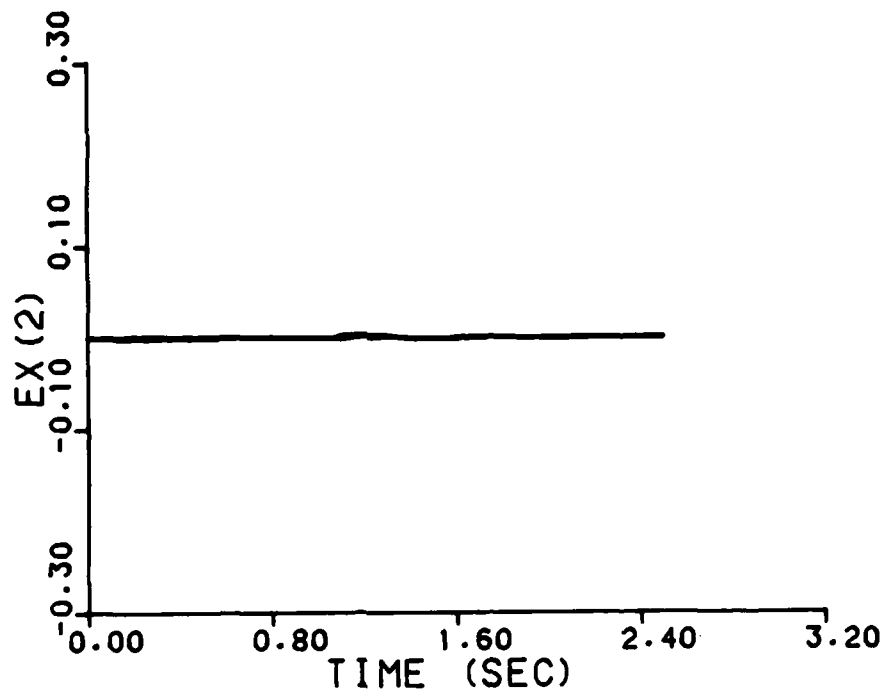
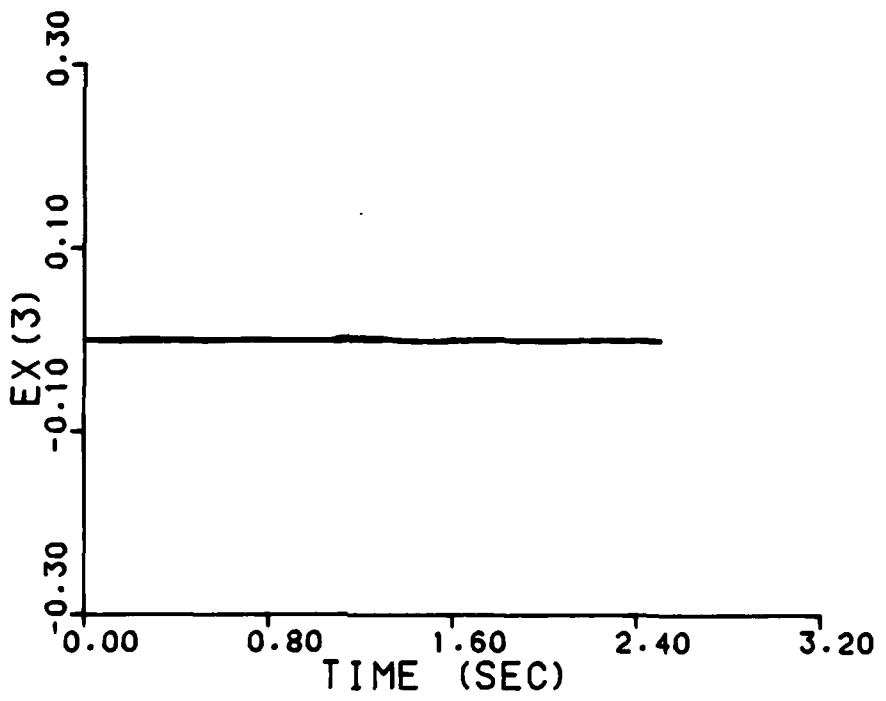
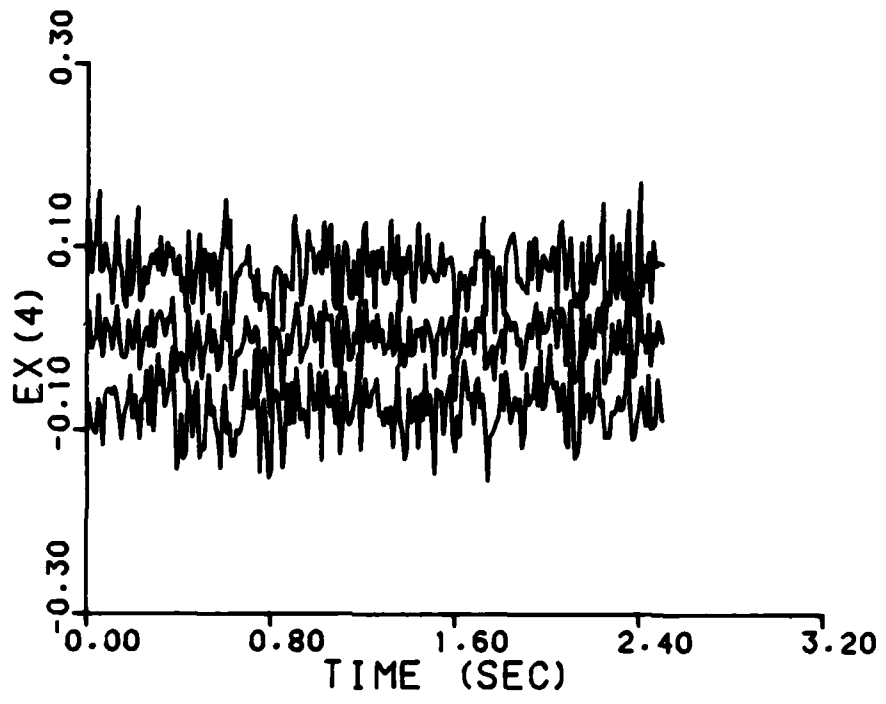


Figure B-19. Jump change in \underline{a}_t at 1.0 sec, with contraction, parameter estimate error, $\underline{a}_t(0) = (3,7)$, $\underline{a}=\underline{a}_t(1.0) = (5,5)$.



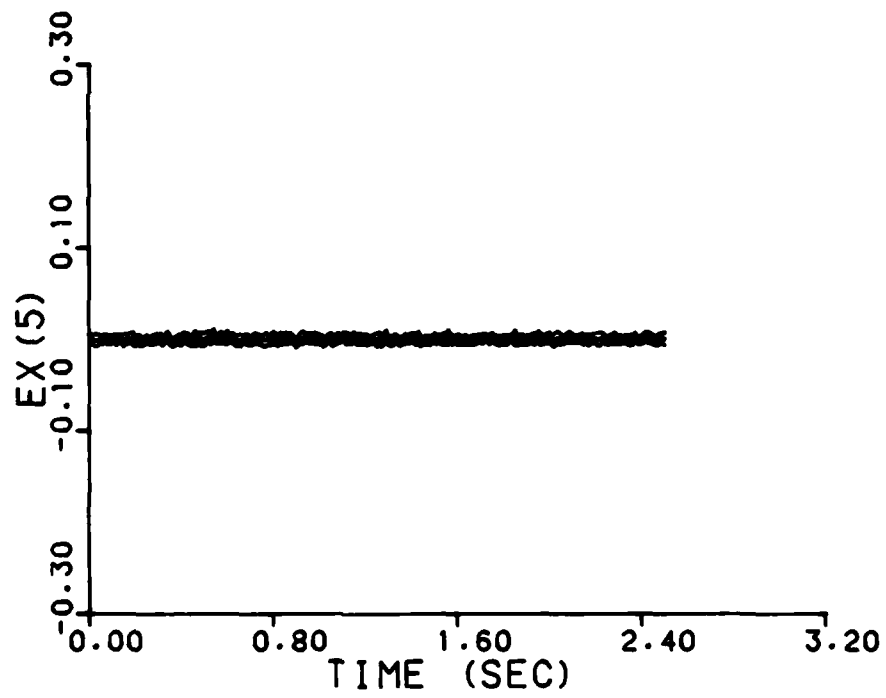
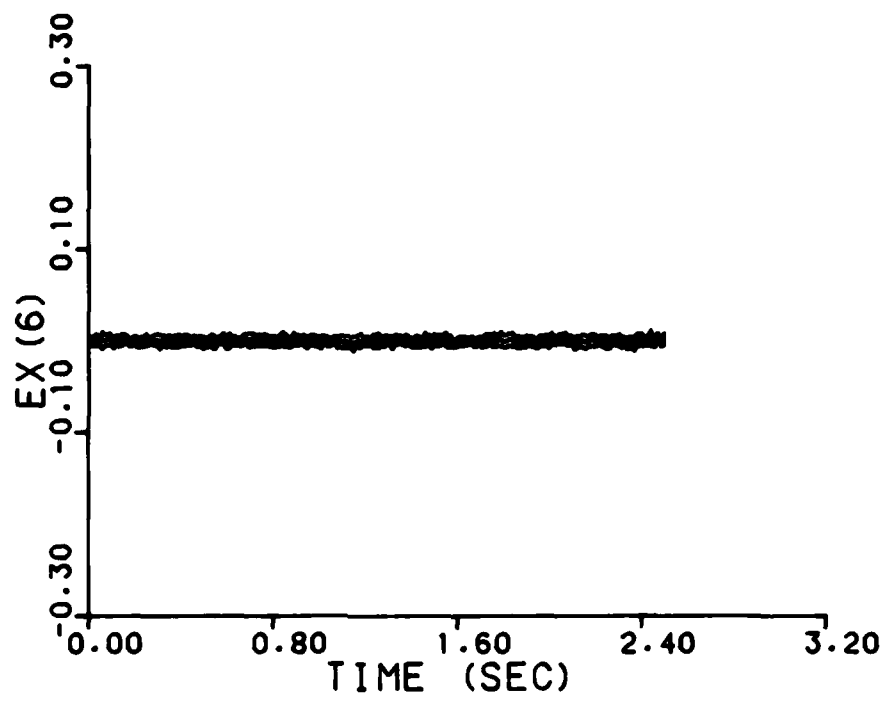
(a)

Figure B-20. Jump change in a_t at 1.0 sec, with contraction, state estimate error, $a_t(0) = (3,7)$, $a_{=t}(1.0) = (5,5)$.



(b)

Figure B-20. (cont.)



(c)

Figure B-20. (cont.)

Appendix C.

Simulation Software

Four main problems were used for this investigation. A preprocessor (PROGRAM SETUPS), used to create the parameter space; a primary processor (PROGRAM BANK), used to generate the simulation data; a post processor (PROGRAM RESULT), used to plot the results; and a program to perform ambiguity function analysis (PROGRAM AMBIG). For each program this Appendix contains a structure chart, and the program and subroutine headers (which describe each module and present pseudocode for the algorithms performed). The actual FORTRAN code is not included but is available through Dr. P. S. Maybeck at the Air Force Institute of Technology, Department of Electrical Engineering. The programs make use of four Libraries available on the Aeronautical Systems Division CDC Cyber computer: IMSL5, LQGLIB, DISSPLA, and CCLIB.

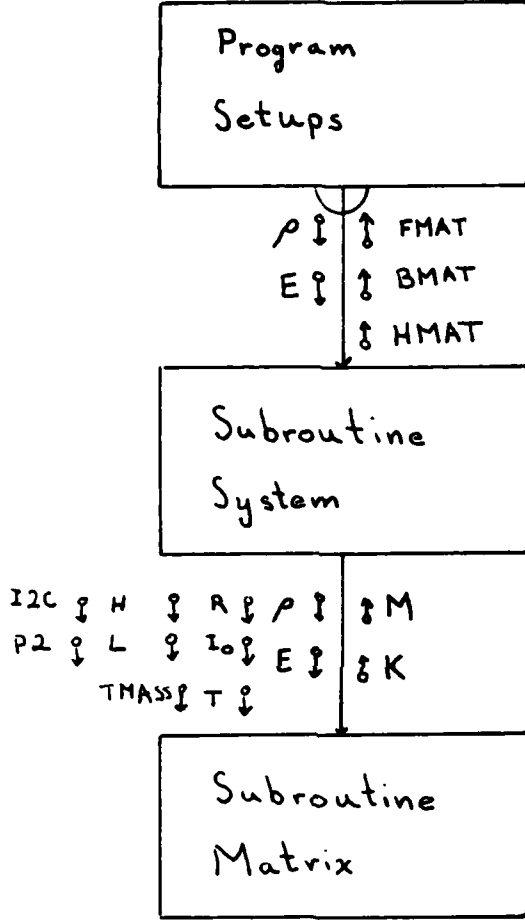


Figure C-1. Program Setups Structure Chart

PROGRAM SETUPS

PROGRAM SETUPS

DESCRIPTION: THIS PROGRAM CALCULATES ALL OF THE CONSTANT MATRICES
REQUIRED BY PROGRAM BANK AND AMBIG.

AUTHOR : KARL HENTZ AND PAUL G. FILIOS
VERSION: 3.03
DATE : 15 AUG 85

INPUT: FILE #PARAMTR# - CONTAINS THE DISCRETE PARAMETER VALUES
FOR THE TWO UNCERTAIN PARAMETERS

OUTPUT: FILE #SPACE# - CONTAINS ALL OF THE CONSTANT MATRICES FOR
EACH DISCRETE PARAMETER POINT

THIS IS THE MAIN PROGRAM

PSEUDOCODE:

READ IN THE PARAMETER VALUES
FOR EACH PARAMETER POINT
COMPUTE THE CONTINUOUS TIME SYSTEM MATRICES
DISCRETIZE THE NOISE INPUT MATRIX
DISCRETIZE THE PLANT AND INPUT MATRICES
COMPUTE THE COVARIANCE MATRIX AND THE FILTER GAIN MATRIX
COMPUTE THE A INVERSE MATRIX
COMPUTE THE DETERMINANT OF A
SAVE THE MATRICES TO LARGER STORAGE ARRAYS
CONTINUE
WRITE THE MATRICES TO THE OUTPUT FILE
END

SUBROUTINE SYSTEM (DEN, E, FMAT, BMAT, HMAT)

SUBROUTINE SYSTEM

DESCRIPTION:

THIS PROGRAM COMPUTES M (MASS), K (STIFFNESS), SUBB (CONTROL) AND SUBE (MEASUREMENT) MATRICES FOR THE DRAPER/RPL SIMPLIFIED LARGE SPACE STRUCTURE, THEN USES THEM TO COMPUTE THE STANDARD STATE SPACE F, B, AND E MATRICES.

AUTHOR: PAUL G. FILIOS

VERSION: 1.0

DATE: 29 JUL 1985

INPUT: DEN, E - THE VALUES FOR THE TWO UNCERTAIN PARAMETERS

OUTPUT: FMAT, BMAT, HMAT - CONTINUOUS TIME MATRICES THAT DESCRIBE THE SYSTEM

PSEUDOCODE:

BEGIN

CALCULATE THE DIMENSIONS OF THE MATRICES

CALL SUBROUTINE THAT COMPUTES THE M AND K MATRICES

CALL EIGENVALUE/EIGENVECTOR ROUTINE

FORM SMAT MATRIX

FORM SUBB MATRIX

FORM SUBE MATRIX

CALCULATE SMAT TRANSPOSE * SUBB = NEWB

FORM BMAT MATRIX

CALCULATE SUBE * S = NEWH

FORM HMAT MATRIX

END

SUBROUTINE MATRIX (MODES, M, K,
+ R, IO, DEN, E, T, H, L, TMASS, I2C, P2)

```
C.....  
C  
C   S U B R O U T I N E   M A T R I X  
C.....  
C  
C   DESCRIPTION: THIS ROUTINE COMPUTES THE M (MASS) AND K (STIFFNESS)  
C   MATRICES FOR THE DRAPER/RPL LARGE SPACE STRUCTURE MODEL.  
C  
C   AUTHOR : PAUL G. FILIOS  
C   VERSION: 2.0  
C   DATE   : 10 JUN 85  
C  
C   PARAMETERS PASSED: MODES, R, IO, DEN, E, T, H, L, TMASS, I2C, P2  
C  
C   PARAMETERS RETURNED: M, K  
C.....
```

```
C PSEUDOCODE:  
C   BEGIN  
C   DETERMINE M AND K ELEMENTS NOT DEPENDENT ON AN INDEX  
C   FOR I = 1 TO MODES  
C   DETERMINE M AND K ELEMENTS DEPENDENT ON A SINGLE INDEX  
C   FOR J = 1 TO MODES  
C   DETERMINE M AND K ELEMENTS DEPENDENT ON TWO INDICIES  
C   NEXT J  
C   NEXT I  
C   END
```

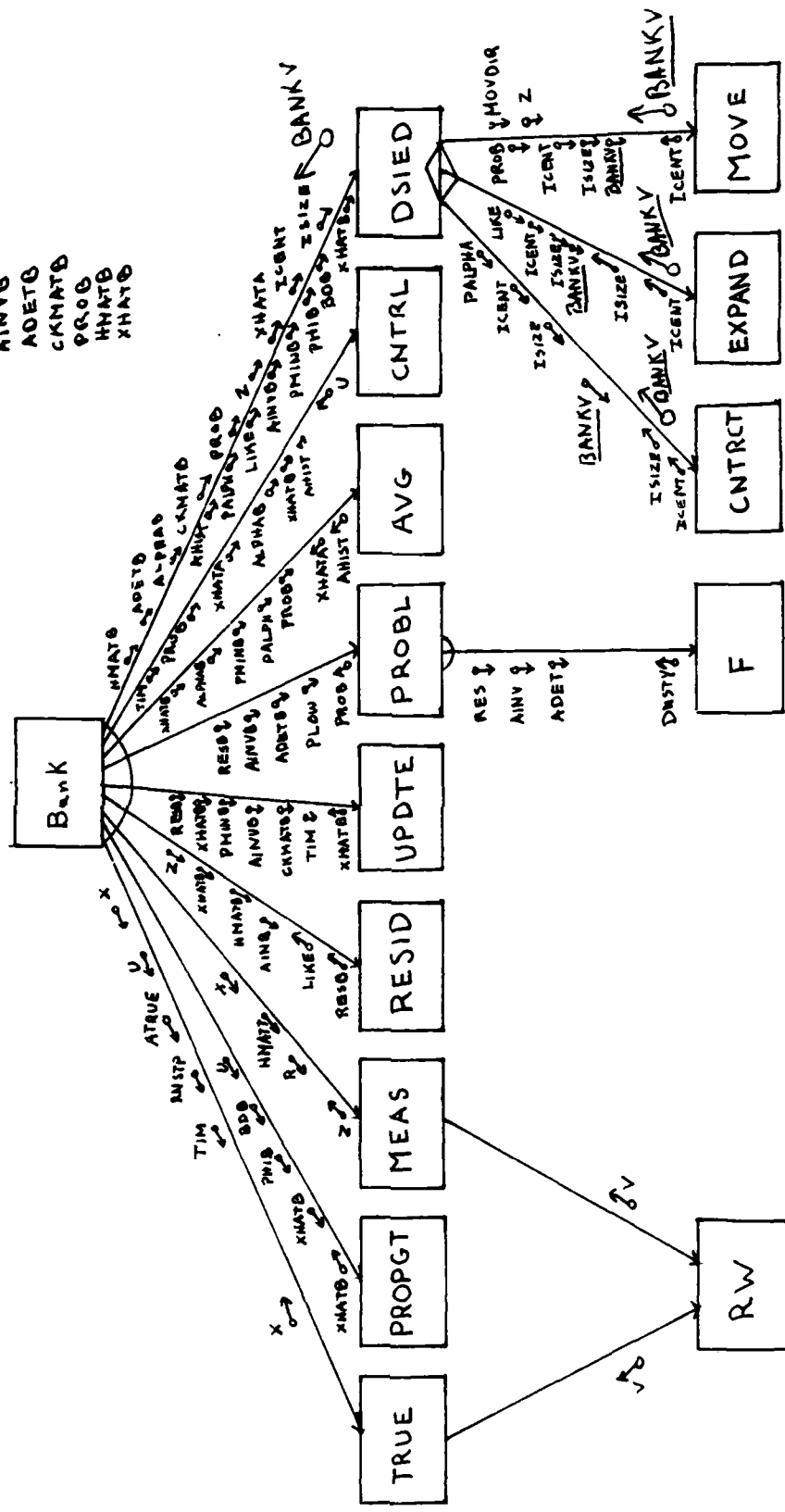


Figure C-2. Program Bank Structure Chart.

PROGRAM BANK

PROGRAM BANK

DESCRIPTION: THIS PROGRAM SIMULATES A MOVING BANK MULTIPLE MODEL
ADAPTIVE KALMAN FILTER, BASED ON THE TRUTH MODEL DESCRIBED BY
SUBROUTINE TRUE AND CONTROLS THE SYSTEM BASED ON THE
CONTROL SYSTEM DESCRIBED BY SUBROUTINE CNTRL.

AUTHOR : KARL HENTZ AND PAUL G. FIIIOS
VERSION: 2.4
DATE : 02 OCT 85

INPUT: FILE 'SPACE'µ WRITTEN BY THE SETUPS PROGRAM, CONTAINS THE
MATRICES FOR EACH FILTER IN THE PARAMETER
SPACE.

FILE 'PARAMTR'µ CONTAINS THE DISCRETE PARAMETER VALUES FOR
EACH VARYING PARAMETER.

FILE 'CONDIT'µ CONTAINS THE INPUT CONDITIONS FOR THE
SITUATION UNDER STUDY.

OUTPUT: FILE 'OUT1'µ CONTAINS THE RAW DATA FOR THE RESLTS
PROGRAM TO PUT INTO GRAPH FORM.

FILE 'OUT2'µ CONTAINS THE DATA GENERATED BY THE FIRST
MONTE CARLO RUN IN A FORM SUITABLE FOR
PRINTING.

FILE 'STATES'µ CONTAINS THE VALUES OF THE SYSTEM STATE
VARIABLES AT EACH SAMPLE TIME OF THE FIRST
MONTE CARLO RUN.

NOTE: THIS PROGRAM HAS BEEN MODIFIED TO USE ONLY AN APPROXIMATION
FOR THE BANK WEIGHTING PROBABILITIES DUE TO NUMERICAL DIFFICULTIES
IN COMPUTING ADET IN THE SETUPS PROGRAM. ONLY SUBROUTINE µFRµ
IS AFFECTED.

.....
.
.
T H I S I S T H E M A I N P R O G R A M
.
.....

. PSEUDOCODE:
. READ THE INPUT VALUES
. INITIALIZE OUTPUT FILES
. FOR EACH MONTE CARLO RUN
. INITIALIZE THE TRUE PARAMETERS
. INITIALIZE THE BANK VARIABLES AND MATRICES
. FOR EACH SAMPLE PERIOD
. PROPAGATE THE TRUE SYSTEM
. PROPAGATE THE ESTIMATE
. TAKE A MEASUREMENT
. UPDATE THE BANK ESTIMATES
. DETERMINE THE BANK PROBABILITIES
. DETERMINE THE NEW ESTIMATE
. COMPUTE THE CONTROL INPUTS
. FOR THE FIRST RUN
. OUTPUT THE PROBS, STATES, AND THRESHOLD VARIABLES
. MOVE, CONTRACT, OR EXPAND THE BANK IF NECESSARY
. COMPUTE THE PERFORMANCE STATISTICS
. OUTPUT THE STATISTICS OF INTEREST
. LOOP
. LOOP
. END

SUBROUTINE CNTRL (AHIST,XHATA,U,TIM,ALPHAB,XHATB,PROB)

SUBROUTINE CNTRL

DESCRIPTION: THIS ROUTINE COMPUTES THE CONTROL INPUT

AUTHOR : KARL HENTZ AND PAUL G. FILIOS
VERSION : 3.0
DATE : 17 SEP 85

PARAMETERS PASSED:

AHIST: THE ESTIMATES OF THE PARAMETER VALUES FOR THE LAST 10
TIME PERIODS, USED FOR ADAPTIVE CONTROL
XHATA: THE WEIGHTED AVERAGE OF STATE ESTIMATES, USED FOR ADAPTIVE
CONTROL
U: THE CONTROL INPUT VECTOR, COMPUTED HERE
TIM: THE CURRENT SIMULATION TIME PERIOD
ALPHAB: THE INDIVIDUAL PARAMETER ESTIMATES OF THE BANK FILTERS
XHATB: THE INDIVIDUAL STATE ESTIMATES OF THE BANK FILTERS
PROB: THE PROBABILITY WEIGHTINGS OF THE BANK FILTERS

PSEUDOCODE:

CASE OF
DITHER SIGNAL
FIXED GAIN CONTROLLER
COMPUTE CONTROL
VARIABLE GAIN CONTROLLER
SELECT GAIN
COMPUTE CONTROL
NNAC
FOR EACH FILTER
COMPUTE CONTROL
ACCUMULATE WEIGHTED AVERAGE
LOOP

END

SUBROUTINE TRUE (X,U,ATRUE,RNSTP,TIM)

SUBROUTINE TRUE

DESCRIPTION: THIS ROUTINE PROPAGATES THE TRUE SYSTEM
FROM ONE SAMPLE TIME TO THE NEXT, ADJUSTING THE TRUE
PARAMETER POINT WHEN DESIRED

AUTHOR : KARL HENTZ AND PAUL G. FILIOS
VERSION : 4.0
DATE : 23 SEP 85

PARAMETERS PASSED:

X: THE TRUE SYSTEM STATES
U: THE CONTROL INPUT VECTOR
ATRUE: THE TRUE PARAMETER POINT
RNSTP: TIME OF THE END OF THE SIMULATION
TIM: THE CURRENT SIMULATION TIME PERIOD

PSEUDOCODE:

IF THE TRUE PARAMETER VARIES
GET THE NEW MATRICES FROM THE PARAMETER SPACE
MULTIPLY THE STATES BY THE STATE TRANSITION MATRIX
MULTIPLY THE CONTROL BY THE TRUE BD MATRIX
GET A NOISE VECTOR
MULTIPLY THE NOISE VECTOR BY BD
ADD ALL THE PRODUCTS FOR THE NEW TRUE STATES
END

SUBROUTINE RW(V)

SUBROUTINE RW

DESCRIPTION: GENERATES A WHITE GAUSSIAN RANDOM VECTOR.

AUTHOR : KARL HENTZ AND PAUL G. FILIOS

VERSION: 2.0

DATE : 26 JUN 85

PARAMETERS PASSED:

V: A RANDOM NOISE VECTOR

PSEUDOCODE:

FOR EACH ELEMENT OF THE NOISE VECTOR

DO 12 TIMES

ACCUMULATE A UNIFORMLY DISTRIBUTED RANDOM NUMBER

LOOP

SUBTRACT SIX

LOOP

END

SUBROUTINE PROPGT(XHATB,PHIB,BDB,U)

SUBROUTINE PROPGT

DESCRIPTION: THIS ROUTINE PROPAGATES THE STATE ESTIMATES OF EACH FILTER IN THE MOVING BANK FROM THE END OF ONE SAMPLE PERIOD TO JUST BEFORE THE MEASUREMENTS ARE TAKEN OF THE NEXT.

AUTHOR : KARL HENTZ AND PAUL G. FILIOS
VERSION: 2.2
DATE : 3 AUG 85

PARAMETERS PASSED:

XHATB: CURRENT STATE ESTIMATE FOR EACH FILTER IN THE BANK, THESE VALUES ARE ALTERED
PHIB: PHI MATRIX FOR EACH FILTER IN THE BANK
BDB: DISCRETE INPUT MATRIX FOR EACH FILTER IN THE BANK
U: CONTROL INPUT VECTOR

PSEUDOCODE:
FOR EACH FILTER
MULTIPLY PHI BY XHAT
MULTIPLY BDB BY U
ADD THE PRODUCTS TOGETHER
LOOP
END

SUBROUTINE MEAS(X, HMATT, Z, R)

SUBROUTINE MEAS

DESCRIPTION: THIS ROUTINE TAKES A MEASUREMENT OF THE TRUE SYSTEM.

AUTHOR : KARL HENTZ AND PAUL G. FILIOS
VERSION: 2.02
DATE : 9 AUG 85

PARAMETERS PASSED:
X: TRUE STATES
HMATT: TRUE MEASUREMENT MATRIX
Z: THE MEASUREMENT RETURNED
R: THE COVARIANCE OF THE MEASUREMENT NOISE

PSEUDOCODE:
CALL THE PSEUDO-RANDOM NUMBER GENERATOR FOR THE V VECTOR
MULTIPLY THE V VECTOR BY THE SQUARE ROOT OF R
MULTIPLY X BY HMATT
ADD THE NOISE TO THE MEASUREMENT
END

SUBROUTINE RESID(Z, XHATB, HMATB, RESS, LIKE, AINVB)

SUBROUTINE RESID

DESCRIPTION: THIS ROUTINE COMPUTES THE RESIDUALS FOR EACH OF THE
ELEMENTAL FILTERS IN THE SLIDING BANK, AND THE LIKELIHOOD
QUOTIENT FOR EACH FILTER

AUTHOR : KARL HENTZ AND PAUL G. FILIOS
VERSION: 2.03

DATE : 3 AUG 85

PARAMETERS PASSED:

Z: THE MEASUREMENTS TAKEN
XHATB: THE CURRENT STATE ESTIMATES FOR EACH FILTER IN THE BANK
HMATB: THE MEASUREMENT MATRICES FOR EACH FILTER IN THE BANK
RESS: THE RESIDUALS RETURNED
LIKE: THE MINIMUM LIKELIHOOD QUOTIENT RETURNED
AINVB: THE A INVERSE MATRIX FOR EACH FILTER IN THE BANK

PSEUDOCODE:

FOR EACH FILTER

MULTIPLY HMAT BY XHAT TO DETERMINE Z EXPECTED

COMPUTE THE RESIDUAL = Z - Z EXPECTED

FORM THE QUADRATIC $R \cdot AINV \cdot R(TRANSPOSE)$

IF THAT IS LESS THAN THE MINIMUM LIKELIHOOD QUOTIENT THEN

MAKE IT THE NEW MINIMUM LIKELIHOOD QUOTIENT

LOOP

END

SUBROUTINE UPDTE(RESB,INATB,PMINB,AINV,CKMATB,TIM)

SUBROUTINE UPDTE

DESCRIPTION: THIS ROUTINE UPDATES EACH OF THE FILTERS IN THE
SLIDING BANK WITH THE RESULTS OF THE MEASUREMENTS.

AUTHOR : KARL HENTZ AND PAUL G. FILIOS
VERSION: 2.03
DATE : 3 AUG 85

PARAMETERS PASSED:

RESB: THE RESIDUALS FOR EACH FILTER IN THE BANK
INATB: THE CURRENT STATE ESTIMATES FOR EACH FILTER IN THE BANK
PMINB: THE COVARIANCE MATRIX BEFORE UPDATE FOR EACH FILTER
AINV: THE A INVERSE MATRIX FOR EACH FILTER IN THE BANK
CKMATB: THE KALMAN FILTER GAIN MATRIX FOR EACH FILTER IN THE BANK
TIM: THE CURRENT TIME IN THE SIMULATION

PSEUDOCODE:

FOR EACH FILTER
MULTIPLY THE GAIN MATRIX BY THE RESIDUALS TO GET THE CHANGE IN
THE STATE ESTIMATES
ADD THE CHANGE TO THE CURRENT STATE ESTIMATES
LOOP
END

SUBROUTINE PROBL(RESB,AINV,ADTB,PROB,PLOW)

SUBROUTINE PROBL

DESCRIPTION: THIS ROUTINE COMPUTES THE PROBABILITY WEIGHTING
FACTORS FOR EACH FILTER IN THE BANK

AUTHOR : KARL HENTZ AND PAUL G. FILIOS
VERSION: 2.00
DATE : 20 JUL 85

PARAMETERS PASSED:

RESB: THE RESIDUALS FOR EACH FILTER IN THE BANK
AINV: THE A INVERSE MATRIX FOR EACH FILTER IN THE BANK
ADTB: THE DETERMINATE OF THE A MATRIX FOR EACH FILTER
PROB: THE CURRENT PROBABILITY WEIGHTINGS FOR EACH FILTER, RETURNED
PLOW: THE PROBABILITY WEIGHTING LOWER BOUND

PSEUDOCODE:

FOR EACH FILTER
GET THE PROBABILITY DENSITY FUNCTION FOR EACH FILTER
MULTIPLY IT BY THE LAST PROBABILITY WEIGHTING
ACCUMULATE THE TOTAL VALUES FOR USE IN THE DNOMINATOR
LOOP
DIVIDE ALL PROBABILITY VALUES BY THE ACCUMULATED TOTAL
END

SUBROUTINE F(RES,AINV,ADET,DNSTY)

SUBROUTINE F

DESCRIPTION: THIS ROUTINE COMPUTES THE PROBABILITY DENSITY
USED IN COMPUTING THE BANK FILTER PROBABILITIES.

AUTHOR : KARL HENTZ AND PAUL G. FILIOS

VERSION: 2.00

DATE : 20 JUL 85

PARAMETERS PASSED:

RES : INPUT RESIDUALS FOR ONE FILTER

AINV: INPUT, THE INVERSE A MATRIX FOR ONE FILTER

ADET: INPUT, THE DETERMINANT OF THE A MATRIX FOR ONE FILTER

DNSTY: OUTPUT, THE PROBABILITY DENSITY COMPUTED

NOTE: BECAUSE OF NUMERICAL DIFFICULTIES IN CALCULATING ADET
DNSTY IS INCOMPLETELY CALCULATED. THE TRUE VALUE OF DNSTY
CAN BE IMPLEMENTED BY REMOVING THE * COMMENT SYMBOL FROM
THE LINE BEFORE THE RETURN STATEMENT.

PSEUDOCODE:

MULTIPLY THE RESIDUALS TRANSPOSE BY A INVERSE

MULTIPLY THE PRODUCT BY THE RESIDUALS

DIVIDE THE PRODUCT BY -2

RAISE E TO THE RESULTING QUOTIENT

SCALE BY $2 \cdot \pi \cdot 5 \cdot \text{SQUARE ROOT OF THE DETERMINANT OF A}$

END

SUBROUTINE AVG(XHATB,ALPHAB,PMINB,PALPH,PROB,XHATA,AHIST)

SUBROUTINE AVG

DESCRIPTION: THIS ROUTINE TAKES THE WEIGHTED AVERAGE OF THE
INDIVIDUAL FILTER'S STATE AND PARAMETER ESTIMATES

AUTHOR : KARL HENTZ AND PAUL G. FILIOS
VERSION: 2.00
DATE : 20 JUL 85

PARAMETERS PASSED:

XHATB: THE STATE ESTIMATES FROM THE FILTERS IN THE BANK
ALPHAB: THE PARAMETER ESTIMATES FROM THE FILTERS IN THE BANK
PMINB: THE COVARIANCE OF THE PARAMETER ESTIMATES BEFORE UPDATE
PALPH: THE COMPUTED COVARIANCE OF THE OVERALL PARAMETER ESTIMATE
RPROB: THE PROBABILITY WEIGHTINGS FOR EACH FILTER
XHATA: THE AVERAGED STATE ESTIMATE COMPUTED AND RETURNED
AHIST: THE HISTORY OF THE LAST 10 PARAMETER ESTIMATES, UPDATED
AND RETURNED

PSEUDOCODE:

SHIFT THE OLD VALUES OF ALPHA IN THE HISTORY ARRAY
FOR EACH FILTER
ACCUMULATE XHAT * THE PROBABILITY WEIGHTING
ACCUMULATE ALPHA * THE PROBABILITY WEIGHTING
LOOP
COMPUTE PALPH
END

SUBROUTINE DSEID(AHIST,PALPH,LIKE,AINV,PMINB,PHIB,BDB,INATB,
+ RMATB,ADETB,ALPHAB,CKMATB,PROB,Z,XHATA,ICENT,ISIZE)

SUBROUTINE DSEID

DESCRIPTION: THIS ROUTINE EXAMINES THE SLIDING BANK AND DECIDES
IF IT NEEDS TO BE MOVED, EXPANDED, OR CONTRACTED, THEN CALLS
THE APPROPRIATE ROUTINE TO ACCOMPLISH ANY NECESSARY ACTION.

AUTHOR : PAUL G. FILIOS
VERSION: 3.00
DATE : 16 AUG 85

PARAMETERS PASSED:

AINV, PMINB, PHIB, BDB, INATB, RMATB, ADETB, ALPHAB, CKMATB,
PROB: THE ARRAYS THAT DESCRIBE THE FILTERS IN THE BANK, IF THE
BANK MOVES, CONTRACTS, OR EXPANDS THESE ARRAYS WILL BE UPDATED
TO REFLECT THE NEW FILTERS IN THE BANK
PALPH, LIKE, PROB: THE VALUES OF THESE PARAMETERS ARE USED IN THE
DECISION LOGIC
ICENT, ISIZE: DESCRIBE THE CURRENT CENTER AND SIZE OF THE BANK
AHIST: CONTAINS THE LAST ESTIMATE OF THE UNCERTAIN PARAMETERS,
USED TO DETERMINE THE NEW BANK CENTER AFTER A CONTRACTION
XHATA: THE CURRENT STATE ESTIMATE, USED TO INITIALIZE NEW FILTERS

PSEUDOCODE:
IF THE BANK NEEDS TO CONTRACT
CONTRACT
IF THE BANK NEEDS TO MOVE
MOVE
IF THE BANK NEEDS TO EXPAND
EXPAND
END

SUBROUTINE CNTRCT (ICENT, ISIZE, PHIB, BDB, PMINB, AINVB, ADET, CKMATB,
PROB, EMATB, XHATB, IHATA, ALPHAB)

SUBROUTINE CNTRCT

DESCRIPTION: THIS ROUTINE CONTRACTS THE SLIDING BANK TO A FINER
DISCRETIZATION.

AUTHOR : PAUL G. FILIOS
VERSION: 1.00
DATE : 20 AUG 85

PARAMETERS PASSED:

ICENT, ISIZE: THE NEW CENTER AND SIZE OF THE BANK
PHIB, BDB, PMINB, AINVB, ADET, CKMATB, PROB, EMATB, XHATB, ALPHAB: THE
ARRAYS DESCRIBING THE FILTERS IN THE BANK
IHATA, THE CURRENT STATE ESTIMATE, USED TO INITIALIZE THE NEW
FILTERS IN THE BANK

PSEUDOCODE:

ENSURE ICENT WILL KEEP SLIDING BANK INSIDE PARAMETER SPACE
IF NOT ADJUST ICENT
REASSIGN THE PHI MATRICES
REASSIGN THE K MATRICES
REASSIGN THE BD MATRICES
REASSIGN THE B MATRICES
REASSIGN THE P MINUS MATRICES
REASSIGN THE AINV MATRICES
REASSIGN THE A DETERMINATES
REASSIGN THE ALPHAS
REASSIGN THE PROBABILITIES
ASIGN XHAT TO NEW FILTERS
END

SUBROUTINE EXPAND(ICENT, ISIZE, PHIB, BDB, PMINB, AINVB, ADET, CKMAT, PROB, HMAT, XHAT, XHATA, ALPHAB)

SUBROUTINE EXPAND

DESCRIPTION: THIS ROUTINE EXPANDS THE SLIDING BANK TO ITS LARGEST DISCRETIZATION.

AUTHOR : PAUL G. FILIOS
VERSION: 1.00
DATE : 20 AUG 85

PARAMETERS PASSED:
ICENT, ISIZE: THE NEW CENTER AND SIZE OF THE BANK
PHIB, BDB, PMINB, AINVB, ADET, CKMAT, PROB, HMAT, XHAT, ALPHAB: THE ARRAYS DESCRIBING THE FILTERS IN THE BANK
XHATA, THE CURRENT STATE ESTIMATE, USED TO INITIALIZE THE NEW FILTERS IN THE BANK

PSEUDOCODE:
SET ISIZE TO 4
SET THE BANK CENTER TO THE PARAMETER SPACE CENTER
RESET PHIB, BDB, HMAT, CKMAT, PMINB, AINVB, ADET,
ALPHAB, PROB, AND XHAT
END

SUBROUTINE MOVE (ICENT, ISIZE, PHIB, BDB, PMINB, AINVB, ADETB, CKMATB,
+ PROB, HMATB, XHATB, XHATA, ALPHAB, MOVDIR, Z)

.....
*
* S U B R O U T I N E M O V E
*
*.....

* DESCRIPTION: THIS ROUTINE MOVES THE SLIDING BANK ONE STEP IN ANY
* DIRECTION

* AUTHOR : PAUL G. FILIOS
* VFRSION: 3.00
* DATE : 20 AUG 85

* PARAMETERS PASSED:
* ICENT, ISIZE: THE NEW CENTER AND SIZE OF THE BANK
* PHIB, BDB, PMINB, AINVB, ADETB, CKMATB, PROB, HMATB, XHATB, ALPHAB: THE
* ARRAYS DESCRIBING THE FILTERS IN THE BANK
* XHATA, THE CURRENT STATE ESTIMATE, USED TO INITIALIZE THE NEW
* FILTERS IN THE BANK

.....
* PSEUDOCODE:
* COMPUTE NEW BANK CENTER
* IF MOVE PUTS PART OF BANK OUTSIDE PARAMETER SPACE
* THEN RETURN
* RESET PHIB, BDB, HMAT, CKMAT, PMINB, AINVB, ADETB,
* ALPHAB, AND XHATB
* ASSIGN NEW PROBABILITIES
* END

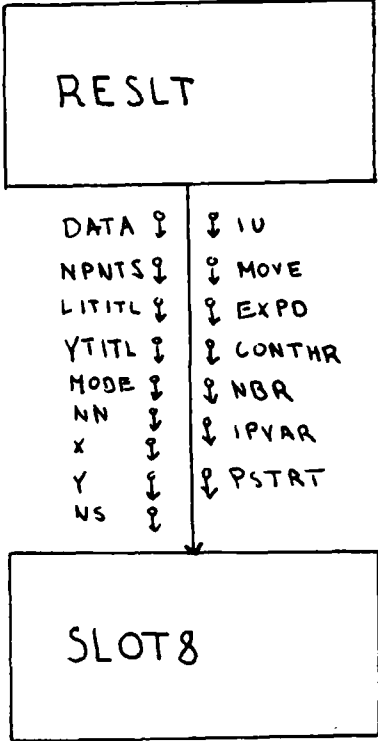


Figure C-3. PROGRAM RESULT structure Chart.

PROGRAM RESULT

PROGRAM RESULT

DESCRIPTION: THIS PROGRAM IS THE POST PROCESSOR FOR PROGRAM BANK,
ORGANIZING THE RAW DATA INTO A FORM THAT CAN BE PLOTTED BY THE
CALCOMP PLOTTER.

AUTHOR : KARL HENTZ AND PAUL G. FILIOS
VERSION: 2.3
DATE : 20 OCT 85

INPUT: FILE 'OUT1' CREATED BY PROGRAM BANK.

OUTPUT: FILE 'TAPE99' READY FOR ROUTING TO THE CALCOMP PLOTTER.

THIS IS THE MAIN PROGRAM

PSEUDOCODE:

INITIALIZE INPUT ARRAYS TO 0
READ IN RAW DATA, SUMMING THE MONTE CARLO RUNS
NORMALIZE DATA BY THE NUMBER OF MONTE CARLO RUNS
SET UP THE PLOT TITLES
PUT THE DATA INTO THE PLOT ARRAYS
PLOT THE DATA
END

SUBROUTINE SLOTS (DATA, NPNTS, LTITL, YTITL, MODE, NN, X, Y, NS, IU,
MOVE, EXPD, CONTR, NBR, IPVAR, PSTRT)

SUBROUTINE SLOTS

DESCRIPTION: THIS ROUTINE PLOTS THE DATA PREPARED IN THE MAIN
PROGRAM USING CALCOMP PLOTTING ROUTINES.

AUTHOR : KARL HENTZ AND PAUL G. FILIOS

VERSION: 2.2

DATE : 21 OCT 85

PARAMETERS PASSED: (ALL ARE INPUT)

DATA: THE DATA POINTS TO BE PLOTTED

NPNTS: THE NUMBER OF DATA POINTS

LTITL: THE TITLE OF THE PLOT

YTITL: AN ARRAY OF TITLES, ONE FOR THE Y AXIS OF EACH SUB-PLOT

MODE: THE MODE THE SLIDING BANK WAS OPERATING IN

MOVE,

EXPD,

CONTR: THE THRESHOLDS USED FOR MOVE, EXPAND, AND CONTRACT DECISIONS.

NBR: THE NUMBER OF MONTE CARLO RUNS MADE.

IPVAR: CODE SIGNIFYING THE VARIABLE OF INTEREST BEING PLOTTED.

PSTART: THE TIME IN THE MONTE CARLO RUNS WHEN THE PLOT STARTS.

PSEUDOCODE:

WRITE TITLE AND NOTES

FOR EACH SUB-PLOT

DRAW AND LABEL THE AXIS

DRAW THE MEAN LINE

DRAW THE PLUS ONE SIGMA LINE

DRAW THE MINUS ONE SIGMA LINE

ADJUST STARTING POINT FOR NEXT SUB-PLOT

END

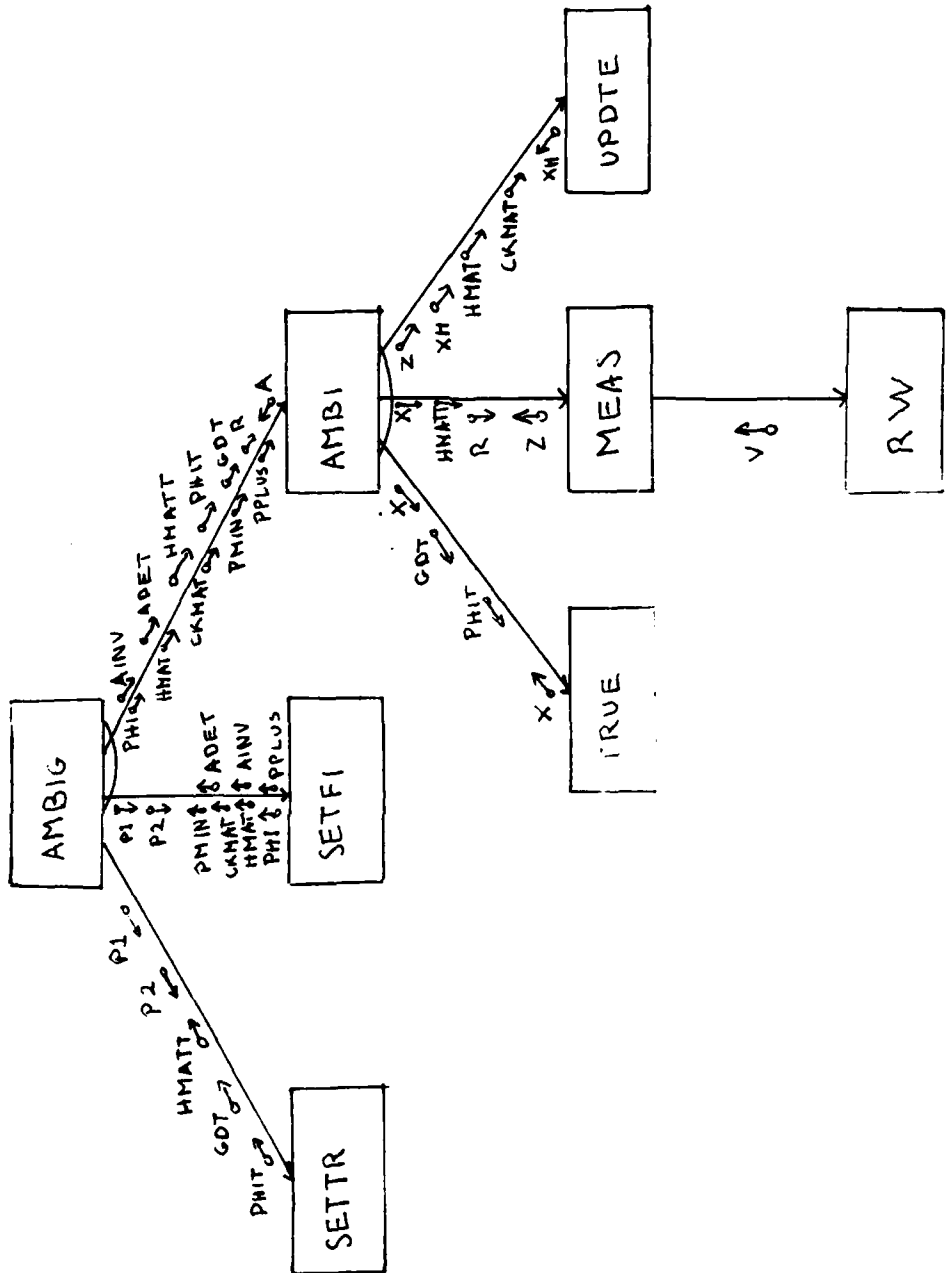


Figure C-4. PROGRAM AMBIG. Structure Chart.

PROGRAM ANBIG

P R O G R A M A N B I G

DESCRIPTION: PERFORMS AMBIGUITY FUNCTION ANALYSIS OF PARAMETER POINTS
IN THE PARAMETER SPACE OF A SLIDING BANK MULTIPLE MODEL ADAPTIVE
ESTIMATOR.

AUTHOR : KARL HENTZ AND PAUL G. FILIOS
VERSION: 2.0
DATE : 20 JUN 85

INPUT: FILE «ANSFAC» CONTAINING THE MATRICES DESCRIBING THE PARAMETER
SPACE

OUTPUT: «TAPE99» PLOT FILE FOR THE CALCOMP PLOTTER

T H I S I S T H E M A I N P R O G R A M

PSEUDOCODE:

READ IN THE PARAMETER SPACE
SET UP THE TRUTH MODEL
FOR EACH FILTER
 SET UP THE FILTER
 DETERMINE THE AMBIGUITY FUNCTION
LOOP
 PREPARE DATA FOR PLOTTING
 PLOT DATA
END

SUBROUTINE SETTR(P1,P2,PHIT,GDT,EMATT)

SUBROUTINE SETTR

DESCRIPTION: SETS UP THE TRUTH MODEL FOR THE PARAMETER POINT DESIRED

AUTHOR : KARL HENTZ AND PAUL G. FILIOS
VERSION: 2.0
DATE : 25 JUN 85

PARAMETERS PASSED:

P1,P2: INDICES TO THE TRUE PARAMETER POINT IN THE PARAMETER SPACE
PHIT: TRUE PHI MATRIX, RETURNED
GDT: TRUE NOISE INPUT MATRIX, RETURNED
EMATT: TRUE MEASUREMENT MATRIX, RETURNED

PSEUDOCODE:

GET PHIT FROM PARAMETER SPACE
GET GDT FROM PARAMETER SPACE
GET EMATT FROM PARAMETER SPACE
END

SUBROUTINE SETFI(P1,P2,PHI,HMAT,CKMAT,PMIN,PPLUS,AINV,ADET)

SUBROUTINE SETFI

DESCRIPTION: SETS UP THE FILTER MODEL FOR THE PARAMETER POINT
DESIRED.

AUTHOR : KARL HENTZ AND PAUL G. FILIOS
VERSION: 2.0
DATE : 25 JUN 85

PARAMETERS PASSED:

PI,P2: INDICES TO THE PARAMETER POINT DESIRED
PHI: STATE TRANSITION MATRIX FOR DESIRED POINT
HMAT: MEASUREMENT MATRIX FOR DESIRED POINT
CKMAT: FILTER GAIN MATRIX FOR DESIRED POINT
PMIN: COVARIANCE OF STATE ESTIMATES BEFORE UPDATES
PPLUS: COVARIANCE OF STATE ESTIMATES AFTER UPDATES
AINV: A MATRIX FOR DESIRED POINT
ADET: DETERMINANT OF A MATRIX

PSEUDOCODE:

GET PHI FROM PARAMETER SPACE
GET PMIN FROM PARAMETER SPACE
GET PPLUS FROM PARAMETER SPACE
GET CKMAT FROM PARAMETER SPACE
GET HMAT FROM PARAMETER SPACE
GET AINV FROM PARAMETER SPACE
GET ADET FROM PARAMETER SPACE
END

SUBROUTINE AMBIG(PHI, HMAT, CKMAT, PMIN, PPLUS, AINV, ADET,
+ HMATT, PHIT, GDT, R, A)

.....
*
*
* S U B R O U T I N E A M B I G
*
*.....

DESCRIPTION: PERFORMS THE AMBIGUITY FUNCTION EVALUATION FOR A POINT
IN THE PARAMETER SPACE. USES A MONTE CARLO SIMULATION TO COMPUTE
THE ENSEMBLE AVERAGE.

AUTHOR : KARL HENTZ AND PAUL G. FILIOS
VERSION: 2.0
DATE : 25 JUN 85

PARAMETERS PASSED:
PHI, HMAT, CKMAT, PMIN, PPLUS, AINV, ADET: MATRICES DESCRIBING
THE FILTER AT THE PARAMETER POINT OF INTEREST
HMATT, PHIT, GDT, R: MATRICES DESCRIBING THE TRUE SYSTEM
A: THE VALUE OF THE AMBIGUITY FUNCTION AT THE POINT OF INTEREST
*
*.....

* PSEUDOCODE:
* FOR THE NUMBER OF MONTE CARLO RUNS DESIRED
* FOR EACH TIME PERIOD
* PROPAGATE THE STATE ESTIMATE
* TAKE A MEASUREMENT OF THE TRUE SYSTEM
* UPDATE THE FILTER WITH THE MEASUREMENT
* ACCUMULATE ERROR STATISTICS
* LOOP
* LOOP
* COMPUTE AMBIGUITY FUNCTION FROM ERROR STATISTICS
* END

SUBROUTINE TRUE(X,PHIT,GDT)

SUBROUTINE TRUE

DESCRIPTION: PROPAGATES THE TRUE SYSTEM FROM ONE SAMPLE TIME TO THE NEXT.

AUTHOR : KARL HENTZ AND PAUL G. FILIOS
VERSION: 2.0
DATE : 25 JUN 85

PARAMETERS PASSED:
X: TRUE SYSTEM STATES
PHIT: TRUE STATE TRANSMISSION MATRIX
GDT: TRUE NOISE INPUT MATRIX

PSEUDOCODE:
GET A RANDOM NOISE VECTOR
MULTIPLY PHIT BY X
MULTIPLY GDT BY THE NOISE VECTOR
ADD THE PRODUCTS
END

SUBROUTINE RW(V)

.....
S U B R O U T I N E . R W
.....

DESCRIPTION: GENERATES A WHITE GAUSSIAN RANDOM VECTOR.

AUTHOR : KARL HENTZ AND PAUL G. FILIOS

VERSION: 2.0

DATE : 26 JUN 85

PARAMETERS PASSED:

V: A RANDOM NOISE VECTOR
.....

• PSEUDOCODE:

• FOR EACH ELEMENT OF THE NOISE VECTOR

• DO 12 TIMES

• ACCUMULATE A UNIFORMLY DISTRIBUTED RANDOM NUMBER

• LOOP

• SUBTRACT SIX

• LOOP

• END

SUBROUTINE MEAS(X, HMATT, Z, R)

SUBROUTINE MEAS

DESCRIPTION: THIS ROUTINE TAKES A MEASUREMENT OF THE TRUE SYSTEM.

AUTHOR : KARL HENTZ AND PAUL G. FILIOS
VERSION: 2.02
DATE : 3 AUG 85

PARAMETERS PASSED:

X: TRUE SYSTEM STATES
HMATT: TRUE SYSTEM MEASUREMENT MATRIX
Z: MEASUREMENTS TAKEN
R: COVARIANCE OF THE MEASUREMENT NOISE

PSEUDOCODE:

GET A RANDOM NOISE VECTOR
MULTIPLY EACH ELEMENT BY THE SQUAREROOT OF R
MULTIPLY HMATT TIMES X
ADD THE NOISE TO THE PRODUCT
END

SUBROUTINE UPDTE(Z,IB,HMAT,CKMAT)

SUBROUTINE UPDTE

DESCRIPTION: THIS ROUTINE UPDATES THE FILTER WITH THE MEASUREMENT.

AUTHOR : KARL HENTZ AND PAUL G. FILIOS
VERSION: 2.04
DATE : 8 SEP 85

PARAMETERS PASSED:

Z: THE CURRENT MEASUREMENT
IB: THE CURRENT FILTER STATE ESTIMATE
HMAT: THE FILTER MEASUREMENT MATRIX
CKMAT: THE KALMAN FILTER GAIN MATRIX

PSEUDOCODE:

COMPUTE RESIDUAL
COMPUTE CHANGE IN STATE ESTIMATE
ADD CHANGE TO CURRENT STATE ESTIMATE
END

Bibliography

1. Magill, D. T. "Optimal Adaptive Estimation of Sample Stochastic Processes," IEEE Transactions on Automatic Control, Vol. 10, No. 5: 434-439, Oct. 1965.
2. Hilborn, C. G., Jr. and Demetrios G. Lainiotis, "Optimal Estimation in the Presence of Unknown Parameters," IEEE Transactions on Systems Science and Cybernetics, Vol. 5: 38-43, Jan. 1969.
3. Lainiotis, Demetrios G. "Optimal Adaptive Estimation: Structure and Parameter Adaptation," IEEE Transactions on Automatic Control, Vol 16: 160-170, Apr. 1971.
4. _____. "Partitioning: A unifying Framework for Adaptive Systems, I: Estimation," Proceedings of the IEEE, Vol. 64: 1126-1143, Aug. 1976.
5. Maybeck, Peter S. Stochastic Models, Estimation, and Control, Volume 2. New York: Academic Press, 1982.
6. Hawkes, Recharad M. and John B. Moore. "Performance Bounds for Adaptive Estimation," Proceedings of the IEEE, Vol. 64: 1143-1150, Aug. 1976.
7. Dasgupta, S. and L. C. Westphal. "Convergence of Partitioned Adaptive Filters for Systems with Unknown Biases," IEEE Transactions on Automatic Control, Vol. 28: 614-615, May 1983.
8. Athans, M., and C. B. Chang, "Adaptive Estimation and Parameter Identificaion Using Multiple Model Estimation Algorithm," Technical Note 1976-28, ESD-TR-76-184, Lincoln Laboratory, Lexington, Mass., June 1976.
9. Chang, C. B., and J. A. Tabaczynski, "Application of State Estimation to Target Tracking," IEEE Transactions Automatic Control, Vol. 29, No. 2: 98-109, Feb. 1984.
10. Chang, C. B., and M. Athans, "State Estimation for Discrete Systems with Switching Parameters," IEEE Transactions on Aerospace and Electronic Systems, Vol. 14, No. 3: 418-424, May 1978.
11. Athans, Michael et al. "The Stochastic Control of the F-8C Aircraft Using a Mutiple Model Adaptive Control (MMAC) Method - Part 1: Equilibrium Flight," IEEE Transactions on Automatic Control, Vol. 22, No. 5: 768-780. Oct. 1977.
12. Maybeck, Pete S. Stochastic Models, Estimation, and Control, Volume 3. New York: Academic Press, 1982.

13. Thorp, James S. "Optimal Tracking of Maneuvering Targets," IEEE Transactions on Aerospace and Electronic Systems, Vol. 9: 512-519, July 1973.
14. Moose, Richard L. "An Adaptive State Estimation Solution to the Maneuvering Target Problem," IEEE Transactions on Automatic Control, Vol. 20: 359-362, June 1975.
15. Gholson, N H. and R L. Moose. "Maneuvering Target Tracking Using Adaptive State Estimation," IEEE Transactions on Aerospace and Electronic Systems, Vol. 13: 310-317, May 1977.
16. Moose, Richard L. et al. "Modeling and Estimation for Tracking Maneuvering Targets," IEEE Transactions on Aerospace and Electronic Systems, Vol. 15: 448-456, May 1979.
17. Korn, J. and L. Beean. Application of Multiple Model Adaptive Estimation Algorithms to Maneuver Detection and Estimation. Contract DAAK10-82-C-0020. ALPHATECH, Inc., Burlington, Mass., June 1983 (AD-B075921).
18. Brown, Grover R. "A New Look at the Magill Adaptive Filter as a Practical Means of Multiple Hypothesis Testing," IEEE Transactions on Circuits and Systems, Vol. 30: 765-768, Oct. 1983.
19. Willsky, Alan S. et al. "Dynamic Model-Based Techiques for the Detection of Incidents on Freeways," IEEE Transactions on Automatic Control, Vol. 24: 347-359, June 1980.
20. Sims, Craig S. and M. R. D'Mello. "Adaptive Deconvolution of Seismic Signals," IEEE Transactions on Geoscience Electronics, Vol. 16: 99-103, Apr. 1978.
21. Hostetler, Larry D. and Ronald D. Andreas. "Nonlinear Kalman Filtering Techiques for Terrain-Aided Navigation," IEEE Transactions on Automatic Control, Vol. 28: 315-323, March 1983.
22. Mealy, Gregory L. and Wang Tang. "Application of Multiple Model Estimation to a Recursive Terrain Height Correlation System," IEEE Transactions on Automatic Control, Vol. 28: 315-323.
23. Moose, R. L., and P. P. Wang. "An Adaptive Estimator with Learning for a Plant Containing Semi-Markov Switching Parameters," IEEE Transactions on Systems Science and Cybernetics, 277-281, May 1973.

24. Meer, D. E. Multiple Model Adaptive Estimation for Space-Time Point Process Observations. PhD dissertation. School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, Sept. 1982.
25. Meer, D. E. and P. S. Maybeck. "Multiple Model Adaptive Estimation for Space-Time Point Process Observation," Proceedings of the IEEE Conference on Decision and Control, Las Vegas, Nevada, Dec. 1984.
26. Fry, C. M. and A. P. Sage. "On Hierarchical Structure Adaptation and Systems Identification," International Journal of Control, Vol. 20: 433-452, 1974.
27. Lamb, P. R. and L. C. Westphal. "Simplex-directed Partitioned Adaptive Filters," International Journal of Control, Vol. 30: 617-627, 1979.
28. Hentz, K. P., "Feasibility Analysis of Moving Bank Multiple Model Adaptive Estimation and Control Algorithms." M.S. thesis, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, Dec. 1984.
29. Maybeck, P. S. and K. P. Hentz. "Investigation of Moving-Bank Multiple Model adaptive Algorithms," Proceedings of the 24th IEEE Conference on Decision and Control, Ft. Lauderdale, Florida, Dec. 1985.
30. Maybeck, P. S. Stochastic Models, Estimation, and Control, Volume 3. New York: Academic Press, 1982.
31. Junkins, J. L., Bodden, D. S., and Kamat, M. P. "An Eigenvalue Optimization Approach For Feedback Control of flexible Structures." Proceedings of the Southeastern Conference on Theoretical and Applied Mechanics. Vol. II: 303-308, May 1984.
32. Muckenthaler, T. V. "Incorporating Control Into the Optimal Structural Design of Large Flexible Space Structures" M.S. thesis, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, Dec. 1984.
33. Jensen, R. L. and D. A. Harnly. "An Adaptive Distributed-Measurement Extended Kalman Filter For A Short Range Tracker," M.S. thesis, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, Dec. 1979.

VITA

Captain Paul G. Filios was born on June 4, 1957, in Northampton Massachusetts. He graduated from high school in Amherst, Massachusetts in 1975. He then matriculated at the University of Massachusetts, Amherst campus. He graduated 1979, with the degree of Bachelor of Science in Electrical Engineering and a commission in the USAF through ROTC in September 1979. He was assigned to the Air Logistics Center in San Antonio Texas, Directorate of Material Management; and to Headquarters, US Air Forces Europe, Ramstein AB West Germany, Deputate for Logistics; before being assigned to the Air Force Institute of Technology, in June 1984.

Permanent address: 69 Pelham Rd
Amherst, MA 01002

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

AD-1164016

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED		1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE		Approved for public release; distribution unlimited	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GE/ENG/85D-14		5. MONITORING ORGANIZATION REPORT NUMBER(S)	
6a. NAME OF PERFORMING ORGANIZATION School of Engineering	6b. OFFICE SYMBOL (If applicable) AFIT/ENG	7a. NAME OF MONITORING ORGANIZATION	
6c. ADDRESS (City, State and ZIP Code) Air Force Institute Technology Wright-Patterson AFB, Ohio 45433		7b. ADDRESS (City, State and ZIP Code)	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c. ADDRESS (City, State and ZIP Code)		10. SOURCE OF FUNDING NOS.	
11. TITLE (Include Security Classification)		PROGRAM ELEMENT NO.	PROJECT NO.
		TASK NO.	WORK UNIT NO.
12. PERSONAL AUTHOR(S) Paul G. Filios, B.S.E.E., Capt., USAF			
13a. TYPE OF REPORT MS Thesis	13b. TIME COVERED FROM _____ TO _____	14. DATE OF REPORT (Yr., Mo., Day) 1985 Dec	15. PAGE COUNT 179
16. SUPPLEMENTARY NOTATION			
17. COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB. GR.	
09	05		Adaptive Control Systems, Adaptive Filters, Multiple Model Adaptive Estimation, Multiple Model Adaptive Control
19. ABSTRACT (Continue on reverse if necessary and identify by block number)			
Title: MOVING-BANK MULTIPLE MODEL ADAPTIVE ALGORITHMS APPLIED TO FLEXIBLE SPACECRAFT CONTROL			
Thesis Chairman: Peter S. Maybeck, Professor of Electrical Engineering			
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED <input checked="" type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS <input type="checkbox"/>		21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a. NAME OF RESPONSIBLE INDIVIDUAL Peter S. Maybeck, Prof. of Elec. Eng.		22b. TELEPHONE NUMBER (Include Area Code) (513) 255-2057	22c. OFFICE SYMBOL AFIT/ENG

Approved for public release: LAW APR 1984
 EYON E. WOLVER 16 JAN 86
 Dean for Research and Professional Development
 Air Force Institute of Technology (AFIT)
 Wright-Patterson AFB OH 45433

This investigation applies moving-bank multiple model adaptive estimation/control algorithms to the control of a realistic model of a large flexible spacecraft. Moving-bank multiple model adaptive estimation differs from conventional (full-bank) multiple model adaptive estimation in that a substantially reduced number of elemental filters is required for the moving-bank estimator (9 vs. 100 for the system modeled in this thesis). The positions in parameter space that the reduced number of elemental filters occupy are dynamically re-declared; i.e., the moving-bank slides about the parameter space in search of the true parameter point.

Critical to the performance of the moving-bank multiple model adaptive estimator is the decision logic used to determine which elemental filters are implemented in the bank, and when to change this decision. The decision logics discussed focus on three situations: initial acquisition of the unknown parameter values, through reducing bank discretization; tracking the unknown parameter values, through bank movement; and reacquisition of the unknown parameters following a large jump change in their values, through expanding bank discretization. Ambiguity function analysis is used to predict performance in these situations.

The system to be controlled is a simplified model of a large scale space structure. Its equations of motion are developed and placed in state space form, the states being the positions and velocities of the rigid body mode and the second and fourth bending modes. The state space matrices describing the system are computed based on nominal values for all physical parameters with the exception of the mass density of the structure arms and their modulus of elasticity. These two parameters are allowed to vary in discrete steps, establishing the parameter space. It is then attempted to control the states to the quiescent state, using moving-bank multiple model adaptive algorithms.

The results indicate that, although the system under study did not have a great need for adaptive estimation and control, the multiple model adaptive estimator performs essentially identically to a single filter artificially knowledgeable of the uncertain parameter values. In addition changing bank discretization for the initial parameter acquisition phase speeded acquisition. However, the bank was unable to expand following a jump change in the uncertain parameter values, in order to restart the acquisition phase; the bank tracked the jump change through movement alone. Ambiguity function analysis proved to be an excellent predictor of bank performance, and should be used as a design tool.

END

FILMED

3 - 86

DTIC