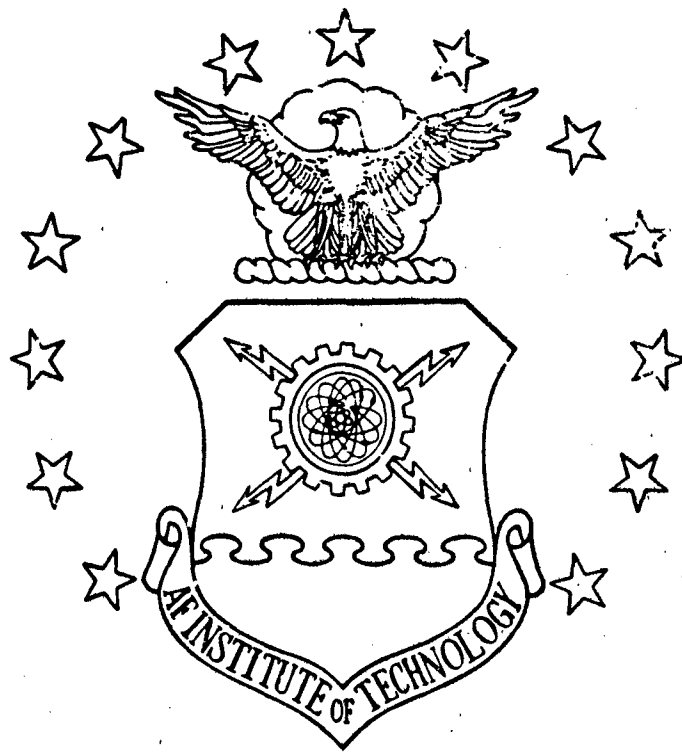


REPRODUCED AT GOVERNMENT EXPENSE

DTIC

1

AD-A152 015



20000814168

TRANSFER OF TECHNOLOGY
RESEARCH AND DEVELOPMENT
COLLECTION ALGORITHM

THESIS

Karl P. Davis
First Lieutenant, USAF

AFTR/34/10/040-12

DISTRIBUTION STATEMENT A

Approved for public release
Distribution Unlimited

DTIC FILE COPY

DTIC
ELECTE
APR 3 1985
S

B

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

85 03 13 053

FEASIBILITY STUDIES OF MOVING FROM
MULTIPLE POINT ADMINISTRATIVE RESEARCH AND
CONTROL TO CORRELATION

THESE

Karl P. Hertz
First Lieutenant, USAF

AFIT/OL/11/84-32

DTIC
ELECTE
APR 3 1985
S B D

DISTRIBUTION STATEMENT A

Approved for public release
Distribution Unlimited

AFIT/GE/ENG/84D-32

FEASIBILITY ANALYSIS OF MOVING BANK MULTIPLE
MODEL ADAPTIVE ESTIMATION AND CONTROL ALGORITHMS

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

Karl P. Hentz, B.S.E.E.

First Lieutenant, USAF

December 1984

Approved for public release; distribution unlimited

Prefac.

The purpose of this effort was to investigate the feasibility of a moving bank multiple model adaptive estimation algorithm. Multiple model adaptive estimation/control appeals to me through its straightforward method of parameter adaptation. However, practical limitations to this method of adaptive estimation/control arise when there are a large number of uncertain parameters, or a fine parameter space discretization. The modified multiple model adaptive estimation/control method explored in this thesis circumvents the practical limitations of the conventional approach, without significantly increasing complexity.

I wish to express my gratitude to my thesis advisor, Dr. P. S. Maybeck, for his guidance, encouragement, and enthusiasm throughout this effort. Thanks are also due Maj. W. H. Worsley for his assistance. Finally, I wish to thank my wife Sandi for her encouragement, support, and understanding.

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	



Karl P. Hentz

Contents

	Page
Preface	ii
List of Figures	v
List of Tables	vii
Abstract	viii
I. Introduction	1
I.1. Background	1
I.2. Problem	11
I.3. Scope	12
I.4. Approach	13
I.5. Organization	14
II. Estimator	15
II.1. Introduction	15
II.2. True System	16
II.3. Sliding Bank	19
II.4. Moving the Bank	22
II.5. Summary	31
III. Controller	33
III.1. Introduction	33
III.2. Basic LQ Controller Structure	33
III.3. Single Changeable-Gain Controller	36
III.4. Single Fixed-Gain Controller	39
III.5. Sliding Bank Multiple Model Adaptive Controller	40
III.6. Summary	42
IV. Simulation	44
IV.1. Introduction	44
IV.2. Monte Carlo Analysis	44
IV.3. Software Structure	48
IV.4. Simulation Plan	49
IV.5. Summary	53
V. Results	55
V.1. Introduction	55
V.2. Estimator Performance	57
V.3. Controller Performance	87

	Page
V.4. Computational Loading	93
V.5. Summary	99
VI. Conclusions	102
VI.1. Introduction	102
VI.2. Sliding Bank Multiple Model Adaptive Estimator	103
VI.3. Controller	104
VI.4. Areas of Further Study	105
VI.5. Summary	107
Appendix A: Simulation Plots	109
Appendix B: Parameter Estimation Error Plots	286
Appendix C: Simulation Software	325
Appendix D: Ambiguity Function Analysis	332
Bibliography	344
Vita	346

List of Figures

Figure	Page
I-1. Multiple Model Filtering Algorithm	6
II-1. Overall System	15
II-2. Bank Discretizations	25
II-3. Bank Changes	27
II-4. Bank Move	28
II-5. Filter Warm-up	30
III-1. Changeable-Gain Controller	37
III-2. Interpolation	39
III-3. Fixed-Gain Controller	41
III-4. Sliding Bank Multiple Model Adaptive Controller	43
IV-1. Estimator Simulation	47
IV-2. Estimator and Controller Simulation	48
V-1. Bank Discretizations	65
V-2. Bank Contraction	66
V-3. Power Spectral Density of 2 nd Order System	73
V-4. Single-Gain Controller Computational Burden	95
V-5. Sliding Bank Multiple Model Adaptive Controller Computational Burden	97
C-1. Simulation Program Hierarchy	326
C-2. Program BANK Flow Chart	327
D-1. Ambiguity Function, $\underline{a}_t = (1, 3)$	334
D-2. Ambiguity Function, $\underline{a}_t = (2, 9)$	336
D-3. Ambiguity Function, $\underline{a}_t = (5, 4)$	337
D-4. Ambiguity Function, $\underline{a}_t = (9, 2)$	339

Figure

Page

D-5. Ambiguity Function, $\underline{a}_t = (10, 10)$	340
D-6. Ambiguity Function, $\underline{a}_t = (0.07, 9.0)$	341
D-7. Ambiguity Function, $\underline{a}_t = (0.93, 41.0)$	343

List of Tables

Table	Page
II-1. Parameter Discretizations	20
IV-1. Estimator Analysis Plan	52
IV-2. Controller Analysis Plan	54
V-1. True Parameters	56
V-2. Thresholds	70
V-3. Acquisition Times	86
V-4. Computation Times	98

Abstract

This investigation examines the feasibility of a moving bank multiple model adaptive estimation/control algorithm. Sliding bank multiple model adaptive estimation differs from conventional multiple model adaptive estimation in that a substantially reduced number of elemental filters is required for the sliding bank estimator (9 elemental filters 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 sliding bank of filters is moved about the parameter space in search of the true parameter point.

Critical to the performance of the sliding bank estimator is the decision method that governs movement of the bank of elemental filters. Because of this, a number of different decision algorithms are discussed and their respective performance compared. Three controller designs are also examined: a single changeable-gain, a single fixed-gain, and a sliding bank multiple model adaptive controller.

States of a damped second order system, with uncertain parameters (damping ratio and undamped natural frequency) are estimated by the sliding bank estimator and then regulated to the quiescent state by the controller. Performance of the sliding bank estimator/controller is compared to a bench-

mark of a single Kalman filter/LQ controller that has (artificial) knowledge of the true parameter values. Comparisons are based upon Monte Carlo analysis results.

The results demonstrate that the sliding bank multiple model adaptive estimator/controller is a possible alternative to the conventional full bank multiple model adaptive estimator/controller. For the system modeled in this study, performance of the sliding bank estimator/controller is essentially identical to that of the benchmark after a short parameter acquisition period.

FEASIBILITY ANALYSIS OF MOVING BANK MULTIPLE MODEL ADAPTIVE ESTIMATION AND CONTROL ALGORITHMS

I. Introduction

I.1. Background

In many estimation and control problems, uncertainties exist in the system model and in parameters defining the properties of driving noises and measurement noises. The performance of the Kalman filter is suboptimal when the information describing the system and required to construct the filter is only approximately known. This gives rise to the need for adaptive estimation/control. There are many approaches to adaptive estimation/control, and a comprehensive classification of the various techniques and an extensive bibliography is presented by Asher et al (1).

In a large class of problems, the system can be modeled as a linear system with uncertain parameters affecting the system, driving noises, and/or measurement noises. Using this model, one approach* to adaptive estimation is to form a bank of Kalman filters, one filter for each possible value of the uncertain parameter vector. The output of each filter is then weighted by the a posteriori probability of that particular parameter being correct, conditioned on the observed history of measurements. The weighted

* Referred to as Bayesian, Partitioned, or Multiple Model Adaptive Estimation.

outputs are then summed to form an optimal estimate of the system state.

This approach was first presented by Magill (2) for the case of perfect scalar observations of a stochastic process modeled as the output of a linear dynamic system, with a parameter vector modeled as unknown but selected from a finite set of known vectors, driven by white Gaussian noise. The results of Magill have been extended by Hilborn and Lainiotis (3) to vector processes. Lainiotis and his associates (4,5) have also discussed both continuous and discrete time adaptive estimation schemes based on the Bayesian approach with either continuous or discrete parameters.

I.1.1. Bayesian Estimation Algorithm Development.

(6:129-136) 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 underscored ~ 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
$B_d(t_i)$:	input matrix
$\underline{\tilde{w}}_d(t_i)$:	white Gaussian input noise vector
$G_d(t_i)$:	noise input matrix
$\underline{\tilde{z}}(t_i)$:	measurement vector
$H(t_i)$:	measurement matrix
$\underline{\tilde{v}}(t_i)$:	white Gaussian measurement noise vector

and the following statistics apply:

$$E\{\underline{w}_d(t_i)\} = \underline{0}$$

$$E\{\underline{w}_d(t_i)\underline{w}_d^T(t_j)\} = Q_d(t_i)\delta_{ij}$$

$$E\{\underline{v}(t_i)\} = \underline{0}$$

$$E\{\underline{v}(t_i)\underline{v}^T(t_j)\} = R(t_i)\delta_{ij}$$

where δ_{ij} is the Kroneker 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 constant parameter vector which is an element of A , where A is a subset of R^p . The parameter vector $\underline{a}$ can affect any or all of the following: Φ , B_d , Q_d , H , and R . The Bayesian estimator computes the following conditional probability density function:

$$f_{\underline{x}(t_i), \underline{a} | \underline{z}(t_i)}(\underline{x}, \underline{a} | \underline{z}_i) = f_{\underline{x}(t_i) | \underline{a}, \underline{z}(t_i)}(\underline{x} | \underline{a}, \underline{z}_i) \cdot f_{\underline{a} | \underline{z}(t_i)}(\underline{a} | \underline{z}_i) \quad (I-1)$$

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 (I-1) can be further evaluated:

$$f_{\underline{a} | \underline{z}(t_i)}(\underline{a} | \underline{z}_i) = f_{\underline{a} | \underline{z}(t_i), \underline{z}(t_{i-1})}(\underline{a} | \underline{z}_i, \underline{z}_{i-1})$$

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

Conceptually, equation (I-2) can now be solved recursively, starting from an a priori probability density of $f_{\underline{a}}(\underline{a})$, as $f_{\underline{z}(t_i)|\underline{a},\underline{z}(t_{i-1})}(\underline{z}_i|\underline{a},\underline{z}_{i-1})$ is Gaussian with a mean of $H(t_i)\hat{\underline{x}}(t_i^-)$ and covariance $[H(t_i)P(t_i^-)H^T(t_i) + R(t_i)]$, where $\hat{\underline{x}}(t_i^-)$ 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 $\underline{x}(t_i)$ becomes:

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

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

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 para-

meter 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 (I-3) and (I-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:

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

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

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

where $\underline{a} \in [\underline{a}_1, \underline{a}_2, \dots, \underline{a}_K]$ 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. Pictorially the algorithm appears as in Figure I-1.

The probability weighting factors for each Kalman filter are calculated from equation (I-5), where

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

and

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

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

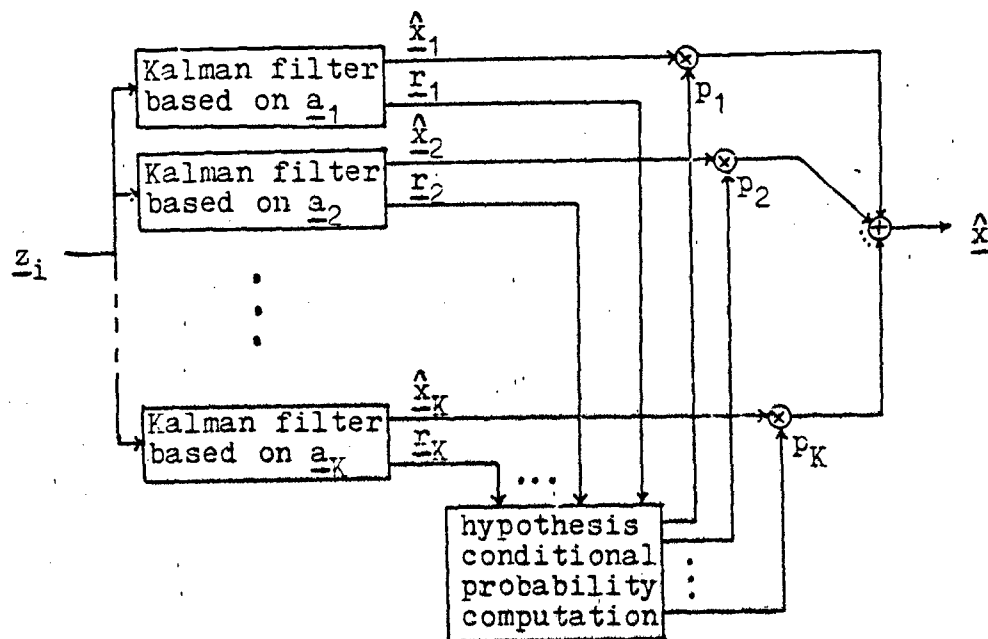


Figure I-1. Multiple Model Filtering Algorithm

Both the residual covariance $A_k(t_i)$ and the residual $r_k(t_i)$ itself are readily available from the k^{th} elemental filter. The estimate of the parameter and the covariance of the parameter are given by:

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

and

$$E\left\{\left[\underline{a} - \hat{\underline{a}}(t_i)\right]\left[\underline{a} - \hat{\underline{a}}(t_i)\right]^T \middle| \underline{z}(t_i) = \underline{z}_i\right\} = \sum_{k=1}^K \left[\underline{a}_k - \hat{\underline{a}}(t_i)\right]\left[\underline{a}_k - \hat{\underline{a}}(t_i)\right]^T p_k(t_i) \quad (I-9)$$

The covariance of the state estimate is given by:

$$\begin{aligned} P(t_i^+) &= E\left\{\left[\underline{x}(t_i) - \hat{\underline{x}}(t_i^+)\right]\left[\underline{x}(t_i) - \hat{\underline{x}}(t_i^+)\right]^T \middle| \underline{z}(t_i) = \underline{z}_i\right\} \\ &= \int_{-\infty}^{\infty} \left[\underline{x} - \hat{\underline{x}}(t_i^+)\right]\left[\underline{x} - \hat{\underline{x}}(t_i^+)\right]^T f_{\underline{x}(t_i) | \underline{z}(t_i)}(\underline{x} | \underline{z}_i) d\underline{x} \\ &= \sum_{k=1}^K p_k(t_i) \left\{ \int \left[\underline{x} - \hat{\underline{x}}(t_i^+)\right]\left[\underline{x} - \hat{\underline{x}}(t_i^+)\right]^T \right. \\ &\quad \left. f_{\underline{x}(t_i) | \underline{a}, \underline{z}(t_i)}(\underline{x} | \underline{a}, \underline{z}_i) d\underline{x} \right\} \\ &= \sum_{k=1}^K p_k(t_i) \left\{ p_k(t_i^+) + \left[\hat{\underline{x}}_k(t_i^+) - \hat{\underline{x}}(t_i^+)\right] \right. \\ &\quad \left. \left[\hat{\underline{x}}_k(t_i^+) - \hat{\underline{x}}(t_i^+)\right]^T \right\} \end{aligned}$$

I.1.2. Convergence. The above estimation algorithm is optimal under the conditions that the parameter space is in fact finite, discrete, and the parameter vector remains at a constant unknown value. The algorithm will converge to the true parameter value, i.e.

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

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

as has been shown by Hawkes and Moore (7). When the underlying parameter space is in fact continuous and the true parameter lies somewhere between the discretized parameters, the algorithm does not, as one might expect, converge to the

condition of nonzero probability associated with the several discretized values that are close to the true value, but instead it converges to the single discrete parameter that is "nearest", as defined in (7), to the true parameter.

The above results have been extended to 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) by Dasgupta and Westphal (8). Under these conditions, the filter may converge to a parameter value that is not close to the true value, and erroneous estimates may result.

One should note that the performance of the filter is dependent on the difference between the residuals of the "correct" elemental filter and the residuals of the mismatched filters. Should the residuals be consistently of the same size, equation (I-7) indicates that the filter with the smallest value of $|A_k|$ would eventually be identified as the correct filter. Since $|A_k|$ is independent of both the residuals and the elemental filter's correctness, erroneous system identification could result. It is therefore important that pseudonoise added during filter tuning to mask model inadequacies not be so strong as to mask the correctness/incorrectness of the elemental filters, thus allowing the adaptive filter to converge to an erroneous parameter value.

To date, satisfactory theoretical convergence results are not available for more general conditions, as, for example, varying parameters(8). When the parameters are slowly varying,

the filter may converge to a particular parameter value before the true parameter varies significantly and remain at that value from then on, while the true parameter may eventually become significantly different from the estimated value.

I.1.3. Varying Parameters. Under the conditions of varying parameters, several modifications have been made to the multiple model adaptive estimation algorithm. If the parameters are slowly varying, an ad hoc approach has been to lower bound the $p_k(t_i)$'s to prevent them from converging to zero (6:135).

Another approach is to model the possible parameter variations; however, the optimal state estimate would then be the weighted sum of estimates produced by filters matched to all possible parameter histories. This would require K^i elemental estimators at time t_i , clearly an impractical approach. If, however, the parameter variation is a Markov process, it has been shown that the number of elemental estimators would remain at K^2 (6:136). Even this may be impractical, however; for the case of two uncertain parameters, each discretized to 10 values, $K = 100$ while $K^2 = 10,000$.

I.1.4. Control. There are several "assumed certainty equivalence design" (9:241) approaches to controlling systems with uncertain parameters. One approach is to use a multiple model adaptive estimator to generate an accurate state estimate and then input the estimate to a fixed gain controller robustified about some nominal value of the parameter vector,

$\underline{a}_{nom}$. The form of the algorithm is

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

or the steady-state constant-gain version

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

Another approach is to use the multiple model adaptive estimator to generate both a state estimate and an estimate of the parameter (Equation I-8). The parameter estimate can then be used to change the gain of the controller as,

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

or the steady-state constant-gain version

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

where $\hat{\underline{a}}(t_i^-)$ is used instead of $\hat{\underline{a}}(t_i^+)$ to reduce computational delay.

A third approach, referred to as multiple model adaptive control, (9:253), is to form K elemental controllers, each fed by the output of one of the K elemental estimators. The controllers are matched to a particular parameter value in the same fashion as the elemental estimators. The outputs of the elemental controllers are then put through the

same weighted average as were the elemental estimators:

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

$$\underline{u}(t_i) = \sum_{k=1}^K p_k(t_i) \underline{u}_k(t_i)$$

I.1.5. Applications. The multiple model approach to adaptive estimation/control has been successfully applied to a number of practical problems. It has shown promising results in the area of maneuvering target tracking (10, 11, 12, 13, 14). It also has been used in flight control (15), adaptive deconvolution of seismic signals (16), multiple hypothesis testing (17), and detection of incidents on free-ways (18). In addition, multiple model adaptive estimation has demonstrated promising results in situations where large initial uncertainties exist, as in terrain aided navigation (19, 20) where non-adaptive extended Kalman filtering techniques can lead to filter divergence.

I.2. Problem

The major drawback to the multiple model adaptive estimation approach is the number of filters needed for implementation. Consider a simple case of a two element parameter vector, with each element taking on 10 possible values, as mentioned previously. The number of elemental Kalman filters needed for the algorithm is then $10^2 = 100$. This clearly presents a computational burden.

Several approaches have been suggested to alleviate the computational demands of the algorithm. Lamb and Westphal (21) presented a method in which the adaptive filter is initialized with a coarse discretization of the parameter space. After the filter converges to the "nearest" parameter, the parameter space is rediscretized using a simplex-directed method. The discretization becomes finer and finer in a contracted parameter space about the true parameter value. Another approach is to structure the algorithms hierarchically to reduce the number of elemental filters (22). "Pruning" or "merging" branches of "decision trees" are also approaches aimed at reducing the computational burden (23, 24).

The approach presented in this thesis is to form a "sliding bank" of elemental estimators. By only implementing, for example, the nine filters corresponding to the parameter values that most closely surround the current best estimate of the parameter, a significant reduction in the number of elemental filters is achieved. The method dynamically re-declares what parameter values are assumed by each filter in the bank.

The purpose of this effort is to evaluate the performance potential of a sliding bank multiple model adaptive estimator/controller.

I.3. Scope

A sliding bank adaptive estimator is evaluated for a simple, but physically motivated, application. A damped second-order, single input, single output system model is

used, with two uncertain parameters (damping ratio ζ and undamped natural frequency ω). The input and measurement noise characteristics are assumed known with no uncertainty and modeled as independent white Gaussian processes with strengths Q and R respectively. The parameters are discretized into 10 values each.

I.4. Approach

The performance of the sliding bank multiple model adaptive estimator/controller, its ability to estimate both 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
- b. the true parameter value is constant and lies in a continuous range between the discrete values
- c. the true parameter is constant and lies outside the initial conditions chosen for the bank (but within the overall parameter space)
- d. the true parameter value is slowly varying within the overall parameter space
- e. the true parameter value undergoes jumps in value

Additionally, the relative performance and computational burden are evaluated for the following controller design strategies:

- a. the sliding bank estimator is used to generate a good state estimate which is passed to a single fixed-gain controller

- b. the parameter estimate is used to change a single controller gain that is used to premultiply the "good" state estimate provided by the sliding bank estimator
- c. a separate controller is tied to each estimator and the controller outputs are then put through the weighted average

The sliding bank algorithm structures investigated are:

- a. acquisition vs. track discretizations
- b. means of determining parameter values associated with the filters in the sliding bank (different algorithms)
- c. constant gain filters and controllers

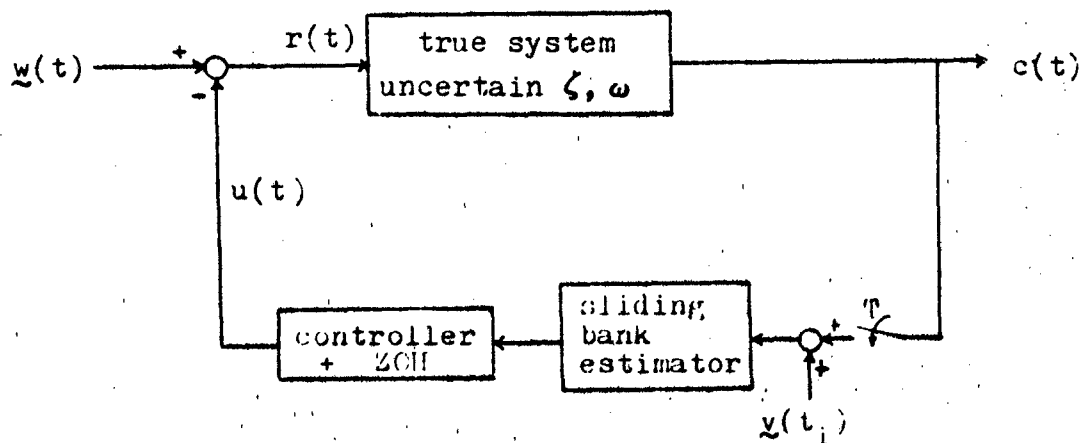
I.5. Organization

The remaining sections of this thesis are organized as follows. Chapter II develops the sliding bank estimator. Chapter III discusses the controller structures. Chapter IV presents the simulation used to evaluate the sliding bank estimator/controller performance. Chapter V is an analysis of the results of the simulation. Chapter VI contains the conclusions.

II. Estimator

II.1. Introduction

A second order system with uncertain damping ratio, ζ , and undamped natural frequency, ω , is to be regulated. Although a simple system, this is physically motivated; for example, in the design of controllers for flexible vehicles, the parameters describing bending modes are often subject to considerable uncertainty. Figure II-1 is a block diagram of the overall system. This chapter details the true system and the sliding bank estimator.



$w(t)$: driving noise

$v(t_i)$: measurement noise

Figure II-1. Overall System

II.2. True System

The true system is modeled as a second order continuous-time system with a transfer function

$$\frac{C(s)}{R(s)} = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2} \quad (\text{II-1})$$

where ζ is the damping ratio and ω is the undamped natural frequency. The standard controllable form state space representation of the true system is

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -2\zeta\omega \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \omega^2 \end{bmatrix} u(t)$$

$$c(t) = [1 \quad 0] \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

or

$$\begin{aligned} \dot{\underline{x}} &= \underline{F}\underline{x} + \underline{B}u \\ c &= \underline{H}\underline{x} \end{aligned} \quad (\text{II-2})$$

where

$$u(t) = r(t)$$

Physically, $x_1(t)$ is a position type variable and $x_2(t)$ is a velocity type variable.

It is assumed that the system is driven by a zero-

mean, white Gaussian noise $\underline{w}(t)$ with a constant strength Q , in addition to the control input $u(t)$. Equation (II-2) now becomes

$$\dot{\underline{x}} = \underline{F}\underline{x} + \underline{B}u + \underline{G}w \quad (\text{II-3})$$

or

$$\begin{bmatrix} \dot{\underline{x}}_1(t) \\ \dot{\underline{x}}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -2\zeta\omega \end{bmatrix} \begin{bmatrix} \underline{x}_1(t) \\ \underline{x}_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \omega^2 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ \omega^2 \end{bmatrix} w(t)$$

with

$$E\{\underline{w}(t)\underline{w}(t+r)\} = Q\delta(r)$$

The filter will be using sampled-data measurements and the controller will be applying sampled-data control to enable implementation of the filter/controller on a digital computer. We are therefore motivated to form an equivalent discrete-time model of the true system for propagating estimates from one sample time to the next. The equivalent discrete-time model of equation (II-3) is

$$\underline{x}(t_{i+1}) = \Phi \underline{x}(t_i) + B_d u(t_i) + G_d w_d(t_i) \quad (\text{II-4})$$

where

$\Phi = \Phi(t_{i+1}, t_i)$ the state transition matrix from t_i to t_{i+1}

$$B_d = \int_{t_i}^{t_{i+1}} \Phi(t_{i+1}, r) B dr$$

$$E\{G_d \underline{w}_d(t_i) \underline{w}_d^T(t_i) G_d^T\} = \int_{t_i}^{t_{i+1}} \Phi(t_{i+1}, r) G Q G^T \Phi^T(t_{i+1}, r) dr$$

The uncertain and/or varying parameters are ζ and ω . The parameters will affect the Φ , B_d , and G_d matrices, assuming that

$$E\{\underline{w}_d(t_i) \underline{w}_d^T(t_j)\} = I \delta_{ij}$$

and

$$G_d G_d^T = \int_{t_i}^{t_{i+1}} \Phi(t_{i+1}, r) G Q G^T \Phi^T(t_{i+1}, r) dr$$

It is assumed that the damping ratio, ζ , can assume values between 0 and 1 and the undamped natural frequency, ω , can assume values between 2π radians/sec and 20π radians/sec ($1\text{Hz} \leq f \leq 10\text{Hz}$).

Measurement of the state vector is available in the form

$$\underline{z}(t_i) = H \underline{x}(t_i) + \underline{v}(t_i) \quad (\text{II-5})$$

or

$$\underline{z}(t_i) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t_i) \\ x_2(t_i) \end{bmatrix} + \underline{v}(t_i)$$

where $y(t_i)$ is a zero-mean discrete white Gaussian noise with constant variance R . In other words, noise corrupted measurements of the position-type state variable are available for the estimator.

II.3. Sliding Bank

The parameters ζ and ω are each discretized into 10 values, forming a 10 by 10 grid within the two dimensional parameter space. The full-bank multiple model adaptive estimator is composed of 100 filters, one for each point in the parameter space. The sliding bank estimator on the other hand, is composed of a subset of the full bank. Only 9 filters are implemented, corresponding to 9 points in the parameter space, and the specification of which particular 9 points is dynamically changeable.

II.3.1. Parameter Space. The damping ratio, ζ , is discretized linearly between 0 and 1. Since the frequency response characteristics of the true system vary logarithmically, the undamped natural frequency, ω , is discretized logarithmically between 2π and 20π rad/sec. Table II-1 summarizes the discretizations.

II.3.2. Elemental Estimator. The elemental estimators are each constant gain Kalman filters, the gains dependent upon which point in the parameter space is associated with that particular filter.

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

TABLE II-1
Parameter Discretizations

ζ	ω (rad/sec)
0	6.28
.1111	8.12
.2222	10.48
.3333	13.54
.4444	17.48
.5556	22.58
.6667	29.16
.7778	37.67
.8889	48.65
1	62.83

to just before the next measurement by

$$\hat{\underline{x}}_k(t_i^-) = \Phi_k \hat{\underline{x}}_k(t_{i-1}^+) + B_d u(t_{i-1}) \quad (\text{II-6})$$

and is updated by

$$\hat{\underline{x}}_k(t_i^+) = \hat{\underline{x}}_k(t_i^-) + K_k [z(t_i) - H \hat{\underline{x}}_k(t_i^-)] \quad (\text{II-7})$$

where

Φ_k is the state transition matrix from t_{i-1} to t_i , based on the k^{th} point in the parameter space

B_{d_k} is the input matrix based on the k-th point in the parameter space

K_k is the steady state Kalman filter gain associated with the k-th point in the parameter space

$z(t_i)$ is the measurement at time t_i

H is the constant measurement matrix

II.3.3. Weighted Average. The outputs of each elemental filter, $\hat{x}_k(t_i^+)$ are put through a weighted average, in the same manner as in Section I.1.1, to produce an overall adaptive estimate of $\underline{x}(t)$. The only difference is that 9 filters, representing a subset of the parameter space, are averaged instead of a full set of filters, representing the entire parameter space:

$$\hat{x}(t_i^+) = \sum_{k=1}^9 \hat{x}_k(t_i) p_k(t_i) \quad (II-8)$$

The probability weightings, $p_k(t_i)$'s, are calculated recursively from equation (I-5), or equivalently

$$p_k(t_i) = \frac{f_k[z(t_i)] p_k(t_{i-1})}{\sum_{j=1}^9 f_j[z(t_i)] p_j(t_{i-1})} \quad (II-9)$$

where

$$f_k[z(t_i)] = \frac{1}{\sqrt{2\pi(\sigma_k^2 + R)}} \exp \left\{ \frac{-[z(t_i) - \hat{x}_{k1}(t_i^-)]^2}{2(\sigma_k^2 + R)} \right\}$$

and

σ_k^2 is the steady state filter computed variance of $\hat{x}_{k_1}(t_i^-)$ associated with the k^{th} parameter point, i. e. $P_{k_1}(t_i^-)$ in steady state

R is the measurement noise strength

II.4. Moving the Bank

When the true parameter point lies within the space spanned by the sliding bank, the sliding bank estimator behaves much as a standard full-bank multiple model adaptive estimator. It is the condition of the true parameter point residing outside the space spanned by the sliding bank that must be accommodated. This condition first must be detected, and then appropriate action must be taken, i. e. move the bank or expand the bank in some fashion. Additionally, should the true parameter point lie near the perimeter of the space spanned by the bank, a move to center the bank over the true parameter point may be desired.

II.4.1. Detection. A number of methods to determine whether or not the true parameter point lies within the space spanned by the sliding bank are proposed.

II.4.1.1. Residual Monitoring. When the true parameter point is within the space spanned by the sliding bank, all of the likelihood quotients, defined as

$$L_k = \frac{r_k^2}{\sigma_k^2 + R} \quad (\text{II-10})$$

where

$$r_k = z - H\hat{x}_k = z - \hat{z}_{k_1}$$

should be below some threshold level, where the threshold level is selected in an ad hoc manner. When the true parameter point is outside the space spanned by the sliding bank, the L_k 's should all be above this threshold. Therefore, a possible detection scheme would be to move the bank when

$$\min(L_1, L_2, \dots, L_9) \geq T$$

where T is the chosen threshold value. Additionally, the elemental filter nearest the true parameter point should have the smallest likelihood quotient, thereby giving an indication of the direction to move the bank.

II.4.1.2. Parameter Position Estimate Monitoring. Another means of keeping the true parameter point inside the space spanned by the bank is to keep the bank centered on the current estimate of the parameter, obtained from Equation (I-8). (Note that the bank cannot always be centered over the estimated parameter position, as when the estimated parameter position is near the outer edges of the overall parameter space, centering the bank would require that some elemental filters lie outside the overall parameter space.) If the distance from the parameter point associated with the

center of the bank to the estimated parameter position becomes larger than some chosen threshold, a move of the bank is indicated.

II.4.1.3. Parameter Position and Velocity

Estimate Monitoring. When the true parameters are slowly changing, several past values of the estimated parameter position can be used to compute a parameter velocity estimate.

The parameter velocity estimate, along with the current estimate of parameter position, can then be used to compute a projected parameter position, one sample period into the future. When the magnitude of the distance between the bank center and the projected parameter position exceeds some selected threshold level, the bank can be moved in anticipation of the true parameter point movement.

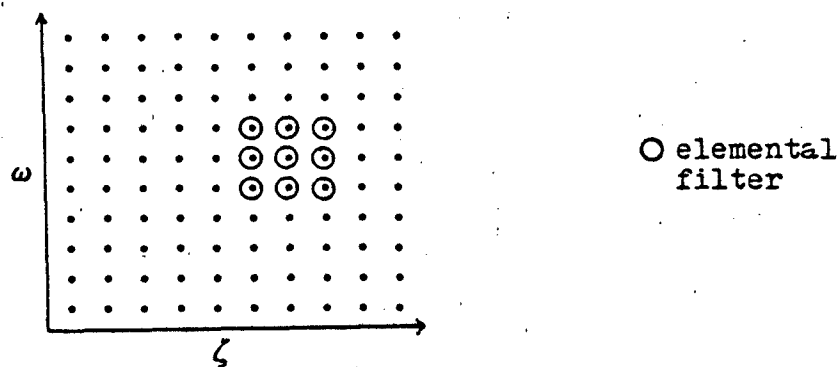
II.4.1.4. Probability Monitoring. The

probability weightings, $p_k(t_i)$'s, are another indication of elemental filter correctness. If any of the probability weightings rise above a chosen threshold level, the bank can then be moved in the direction of the filter with the highest weighting. Should the highest weighting be on the elemental filter corresponding to the center of the bank, the bank is not moved. In this scheme, the bank seeks to center itself on the elemental filter with the highest probability weighting.

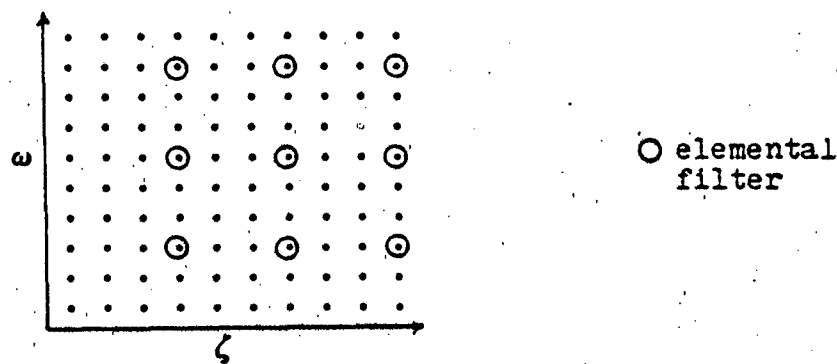
II.4.1.5. Parameter Covariance Monitoring.

Another potential variable in design is the size of the bank.

Instead of the elemental filters occupying adjacent points in the parameter space, the elemental filters could be spaced farther apart, thus covering a larger area of the parameter space with a coarser discretization. See Figure II-2. This coarse discretization would be useful in an ac-



a.



b.

Figure II-2. Bank Discretizations a. fine discretization
b. coarse discretization

quisition cycle. When the sliding bank estimator is first turned on, the discretization could be set so that the sliding bank spans the entire parameter space. The parameter covariance is then monitored (Equation I-9) to indicate when the bank should be contracted. When the covariance drops below some selected threshold, the bank is contracted about the estimated parameter position.

Expanding the bank can also be performed during normal operation. Should the true parameter undergo a jump in position or vary in such a manner that the sliding bank "looses track" of the true parameter position, the bank can be expanded. An indication to expand the bank can be given by residual monitoring. Again, the bank is contracted when the parameter covariance drops below some selected threshold level.

II.4.1.6. Combinations. Several of the above detection schemes can be used in conjunction with one another. For example, parameter covariance monitoring and residual monitoring can be used together, as discussed in Section II.4.1.4. Another example is residual monitoring and parameter position estimate monitoring either "anded" or "or-ed" together.

II.4.2. Appropriate Action. Two courses of action are available when it is detected that the true parameter position is outside the region of the sliding bank. The bank can either be moved in the direction of the true parameter point

so as to encompass it, or it can be expanded to encompass it. Figure II-3a depicts a move and Figure II-3b depicts expansion. When the bank is moved or expanded, the matrices Φ_k , B_{d_k} , and K_k of each elemental filter are changed to correspond to the

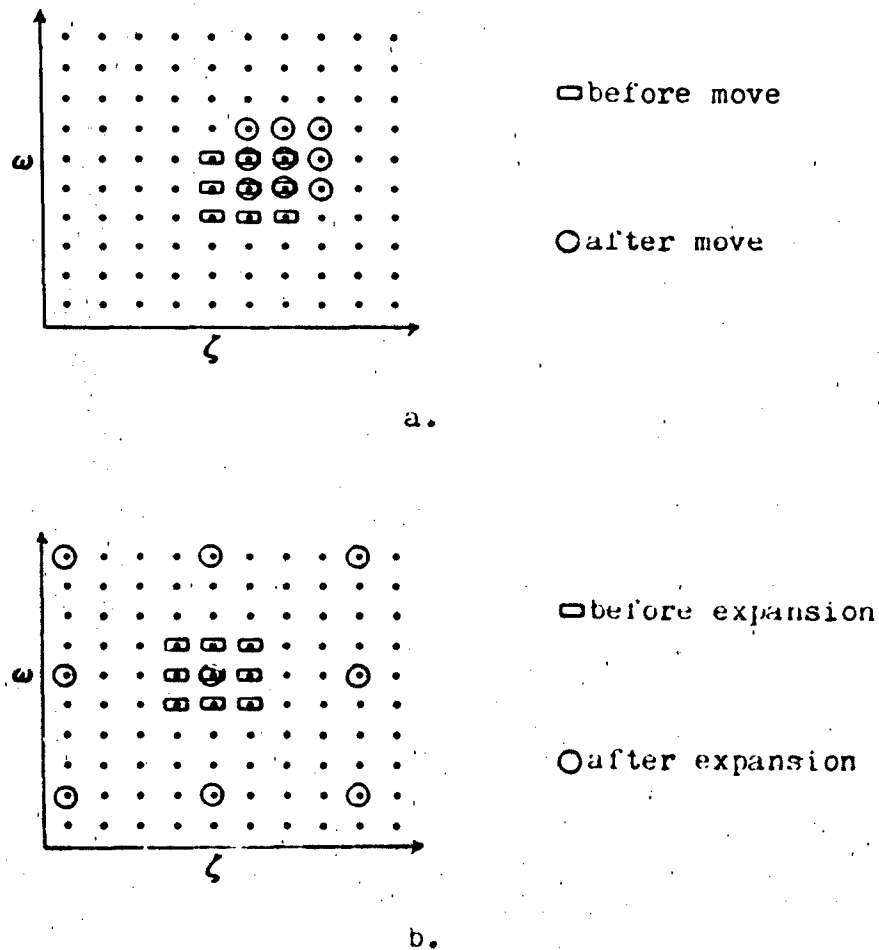


Figure II-3. Bank changes a. bank move b. bank expansion

new parameter points. If there is clear evidence of which direction points toward the true parameter, it is more appropriate to move the bank. If, on the other hand, all of the L_k 's are large and close to one another in magnitude, for example, no clear indication of the true parameter's position is given and it is more appropriate to expand the bank.

When the filter is either moved or expanded, any changed elemental filters must be initialized with values for $\hat{x}_k(t_i)$ and $p_k(t_i)$. Consider a move depicted in Figure II-4. Elemental filters 1, 2, 4, and 5 of the bank after the move, are the same as elemental filters 5, 6, 8, and 9, respectively, before the move and therefore do not need to be changed to new values of Φ_k , B_{d_k} , K_k , $\hat{x}_k(t_i)$, and $p_k(t_i)$. Elemental

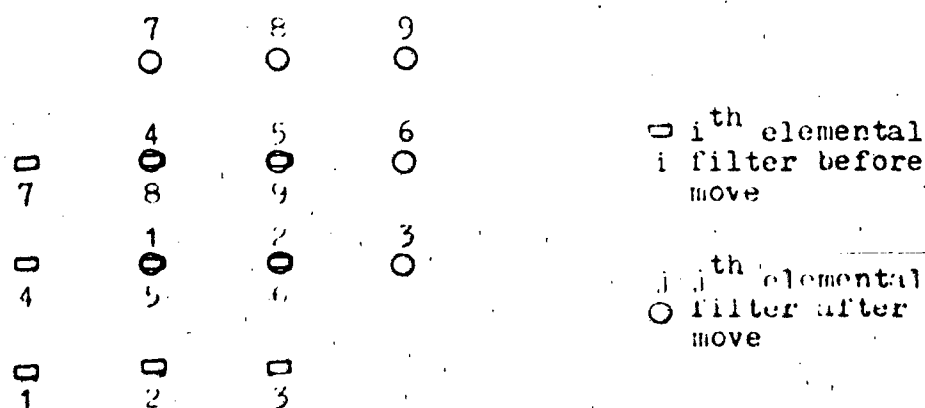


Figure II-4. Bank Move

filters 3, 6, 7, 8, and 9 after the move, on the other hand, are changed filters, i. e. they occupy new positions in the parameter space, not covered by the bank before the move. The unchanged elemental filters (1, 2, 4, and 5 after the move) keep their current values of $\hat{x}_k(t_i)$ and $p_k(t_i)$. The changed elemental filters (3, 6, 7, 8, and 9 after the move) are the ones that must be re-initialized.

One appropriate value for the $\hat{x}_k(t_i)$'s is the current sliding bank estimate of the system state, $\hat{x}(t_i^+)$. Several options exist for the $p_k(t_i)$'s. One option is to set all of the $p_k(t_i)$'s (both changed and unchanged elemental filters) to 1/9, which may result in sluggish convergence to a parameter estimate. Another choice is to reset only the changed elemental filters' $p_k(t_i)$'s to some value indicating the changed filters' correctness, such as

$$p_{i_{ch}}(t_i) = \frac{1}{\sqrt{2\pi\sigma_{i_{ch}}^2}} \exp\left\{-\frac{(z - \hat{x}_1)^2}{2(\sigma_{i_{ch}}^2 + R)}\right\} s \quad (\text{II-11})$$

(note: ch = changed, unch = unchanged)

where s is a scaling factor defined by

$$s = \frac{1 - \sum_{unch} p_j(t_i)}{\sum_{ch} \frac{1}{\sqrt{2\pi\sigma_{i_{ch}}^2}} \exp\left\{\frac{-(z - \hat{x}_1)^2}{2(\sigma_{i_{ch}}^2 + R)}\right\}} \quad (\text{II-12})$$

$$\sum_{k=1}^9 p_k(t_i) = 1$$

	○	○	○
□	□	□	○
□	□	□	○
□	□	□	

○ warm-up filters

	○	○	○
△	▢	▢	○
△	▢	▢	○
△	△	△	

old filters taken
off line

b.

30

when the "warm up" threshold is exceeded. The original 9 filters of the sliding bank continue to provide the overall estimate while the 5 additional filters are warming up. Once the move threshold is exceeded, the 5 warm-up filters are brought on line and the 5 filters of the bank before the move are taken off line, thus accomplishing a move.

II.5. Summary

The true system and the sliding bank estimator are described in this chapter. The true system is a second order, continuous-time system with uncertain parameters, ζ and ω , from which noisy sampled-data measurements of the position type variable are available. The sliding bank estimator is developed to provide an accurate adaptive estimate of the true system states.

Several issues are discussed pertinent to "sliding" the bank. Five decision methods are given for changing the bank:

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

Once the decision is made to change the bank, the bank can either be moved or expanded/contracted. When moving the bank, the probability weightings and state estimates of some of the filters must be reset. The current overall estimate,

$\hat{\underline{x}}(t_i)$ is used to reset the elemental filter's state estimate, while the probability weightings can be either set to an equal weighting or only the changed elemental filters reset. Additionally, a set of elemental filters can be "warmed up" prior to a move or expansion to allow transients to die out.

This research is intended to investigate the effectiveness of these various decision-making and reset methods.

III. Controller

III.1. Introduction

In this chapter, three "forced" certainty equivalence design (9:241) controller schemes are outlined. The purpose of the controller is to regulate the system to the quiescent state. The first scheme is a single linear, quadratic cost, (LQ) full-state feedback controller, fed state estimates by the sliding bank estimator. The controller gains are dependent upon the value of the parameter estimate; thus, the controller is fed both state and parameter estimates by the sliding bank estimator. The second scheme is an approximation to the first. Instead of changing the controller gains according to the parameter estimate, fixed gains are used, based on some nominal value of the parameter vector. The third scheme is a sliding bank multiple model adaptive controller: an elemental LQ controller is tied to each elemental filter of the sliding bank and the outputs of the elemental controllers are put through the weighted average.

III.2. Basic LQ Controller Structure

The basic structure of each of the controller schemes is similar. Each employs a constant-gain LQ full-state feedback controller, where the gains are dependent upon a particular parameter value.

Given the stochastic system

$$\dot{\underline{x}}(t) = \underline{F}\underline{x}(t) + \underline{B}\underline{u}(t) + \underline{G}\underline{w}(t) \quad (\text{III-1})$$

$$E\{\underline{w}(t)\} = 0$$

$$E\{\underline{w}(t)\underline{w}(t+r)\} = Q\delta(r)$$

and the quadratic cost function to be minimized

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

the solution for the optimal control, assuming full-state access, is the LQG regulator

$$u(t_i) = -G_c \underline{x}(t_i) \quad (\text{III-3})$$

where the constant gains, G_c , that minimize J are given by

(9:68-122)

$$G_c = [U + B_d^T K_c B_d]^{-1} [B_d^T K_c \Phi + S^T] \quad (\text{III-4})$$

where K_c satisfies the algebraic Riccati equation

$$K_c = X + \Phi^T K_c \Phi - [B_d^T K_c \Phi + S^T]^T G_c \quad (\text{III-5})$$

and

$$X = \int_{t_i}^{t_{i+1}} \Phi^T(r, t_{i+1}) \underline{W}_x \Phi(r, t_i) dr$$

$$U = \int_{t_i}^{t_{i+1}} [\underline{W}^T(r, t_i) \underline{W}_x \underline{W}(r, t_i) + w_u] dr$$

$$S = \int_{t_i}^{t_{i+1}} \Phi^T(r, t_i) W_X \bar{B}(r, t_i) dr$$

$$\bar{B}(t, t_i) \triangleq \int_{t_i}^t \Phi(t, r) B dr$$

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

$\Phi(t_2, t_1)$ is the state transition matrix from t_1 to t_2

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

Note that Equation (III-3) is also the solution to the deterministic LQ optimal control problem with no driving noise $w(t)$, and that if the assumption of full state access is replaced by noise-corrupted measurements being available, then $\underline{x}(t_i)$ in Equation (III-3) is replaced by $\hat{\underline{x}}(t_i^+)$ generated by a Kalman filter. This certainty equivalence is valid if all system parameters are assumed to be known perfectly.

For the true system where

$$F = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -2\zeta\omega \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 0 \\ \omega^2 \end{bmatrix}$$

G_c will be a function of ζ and ω :

$$G_c[\zeta, \omega] = G_c[\underline{a}]$$

where $\underline{a}$ is the parameter vector

$$\underline{a} = \begin{bmatrix} \zeta \\ \omega \end{bmatrix}$$

III.3. Single Changeable-Gain Controller

In this scheme, the controller gains are calculated based on the value of $\hat{\underline{a}}(t_i)$ (from Equation I-8). The control becomes

$$u(t_i) = -G_c[\hat{\underline{a}}(t_i)]\hat{\underline{x}}(t_i^+) \quad (\text{III-6})$$

or in an approximate form (to reduce computational delay)

$$u(t_i) = -G_c[\hat{\underline{a}}(t_{i-1})]\hat{\underline{x}}(t_i^+)$$

or

$$u(t_i) = -G_c[\hat{\underline{a}}(t_{i-1})]\hat{\underline{x}}(t_i^-)$$

Figure III-1 depicts this scheme.

Rather than solve Equation (III-4) in real time at each sample time for a new value of $\hat{\underline{a}}(t_i)$, what is done is to set up a table of $G_c[\underline{a}_i]$'s, one for each point in the discretized

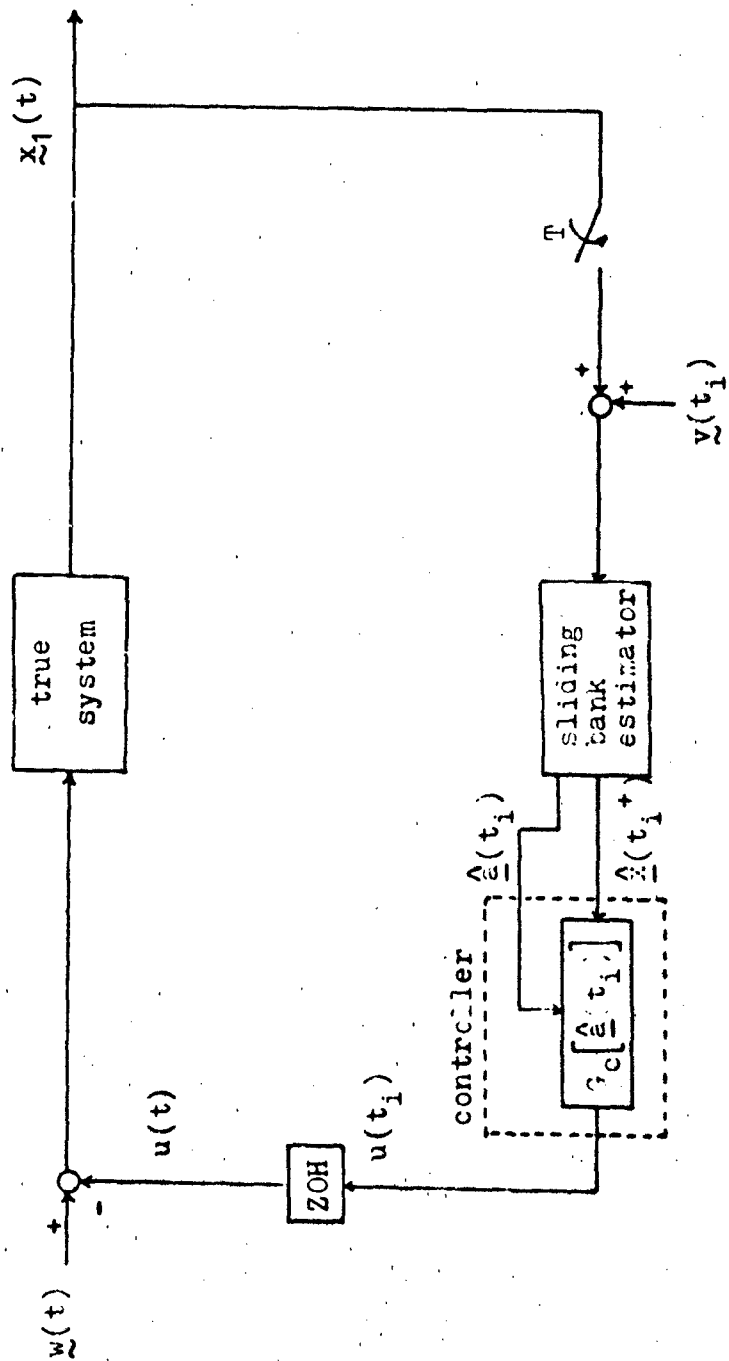


Figure III-1. Changeable-Gain Controller

parameter space, and interpolate between them. If, for example,

$$a_{1j} \leq \hat{a}_1(t_i) < a_{1j+1}$$

$$a_{2k} \leq \hat{a}_2(t_i) < a_{2k+1}$$

then one simple form of interpolation computes

$$m_1 = \frac{\hat{a}_1(t_i) - a_{1j}}{a_{1j+1} - a_{1j}}$$

$$m_2 = \frac{\hat{a}_2(t_i) - a_{2k}}{a_{2k+1} - a_{2k}}$$

$$G_1 = G_c[a_{1j}, a_{2k}] + m_1 \{G_c[a_{1j+1}, a_{2k}] - G_c[a_{1j}, a_{2k}]\}$$

$$G_2 = G_c[a_{1j}, a_{2k+1}] + m_2 \{G_c[a_{1j+1}, a_{2k+1}] - G_c[a_{1j}, a_{2k+1}]\}$$

and then finally

$$G_c[\hat{a}_1(t_i), \hat{a}_2(t_i)] \approx G_1 + m_2(G_2 - G_1)$$

The interpolation is presented pictorially in Figure III-2.

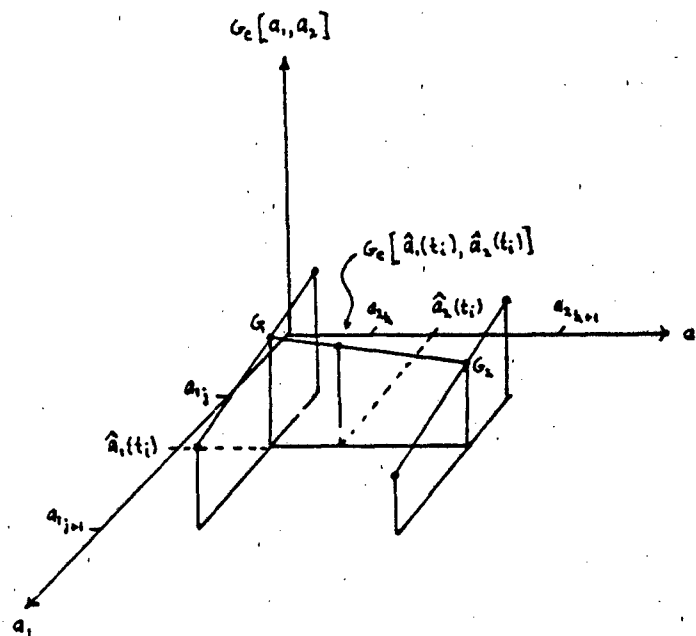


Figure III-2. Interpolation

III.4. Single Fixed-Gain Controller

The second controller structure is an approximation to the first. The gains are not changed according to $\hat{\underline{a}}(t_1)$ but are based on a nominal value of the parameter vector, $\underline{a}_{nom}$. The weighting matrices, W_x and W_u , and nominal values of ζ and ω are selected so that the controller provides adequate regulation for any true system parameter value. (Note that

selecting $\underline{a}_{nom}$ may be a non-trivial task.) This single fixed-gain controller receives estimates of the system states by the sliding bank estimator and generates the control from

$$u(t_i) = -G_c[\underline{a}_{nom}]\hat{\underline{x}}(t_i^+) \quad (III-7)$$

or (for reduced computational delay)

$$u(t_i) = -G_c[\underline{a}_{nom}]\hat{\underline{x}}(t_i^-)$$

where $G_c[\underline{a}_{nom}]$ is computed from Equation (III-4) with

$$\underline{a}_{nom} = \begin{bmatrix} \zeta_{nom} \\ \omega_{nom} \end{bmatrix}$$

This is a reasonable strategy because full-state feedback controllers are inherently robust, and the adaptiveness in the estimator is used only to produce more accurate state estimates to this one set of robust controller gains. Figure III-3 depicts the single fixed-gain controller.

III.5. Sliding Bank Multiple Model Adaptive Controller

This controller design (9:23) differs from both of the single controller gain designs. In this design, a separate elemental controller is cascaded with each of the elemental filters of the sliding bank. The controller outputs (instead of the $\hat{\underline{x}}_k(t_i)$'s are put through the weighted average (Section II.3.3). The elemental controller outputs are

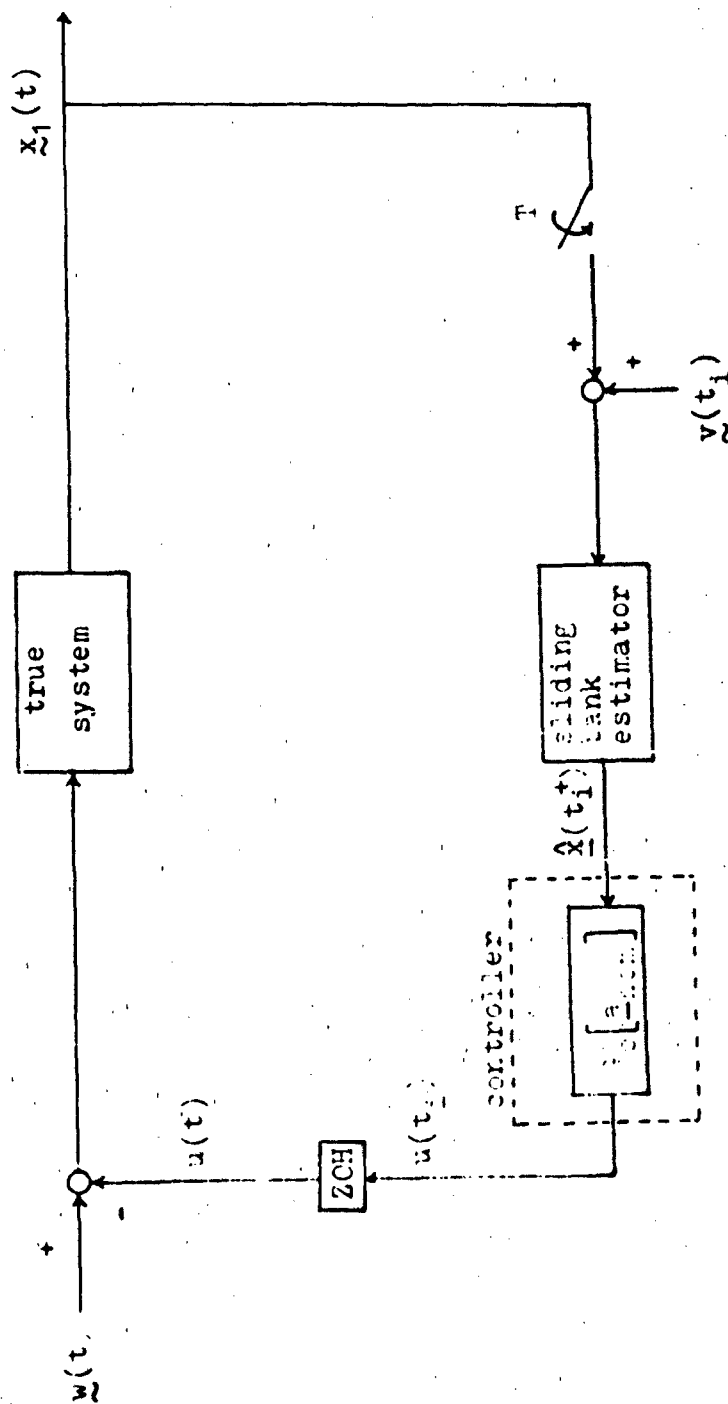


Figure III-3. Fixed-Gain Controller

$$u_k(t_i) = -G_c[\underline{a}_k]\hat{\underline{x}}_k(t_i^+) \quad (\text{III-8})$$

where $\underline{a}_k$ is the parameter value associated with the k^{th} filter/controller in the bank. The overall control is

$$u(t_i) = \sum_{k=1}^9 u_k(t_i)p_k(t_i) \quad (\text{III-9})$$

where $p_k(t_i)$ is the probability weighting from Equation (II-9). Figure III-4 depicts the sliding bank multiple model adaptive controller design.

III.6. Summary

The three controller schemes have been presented in this chapter. The relative performance and computational burden of each is investigated in this thesis.

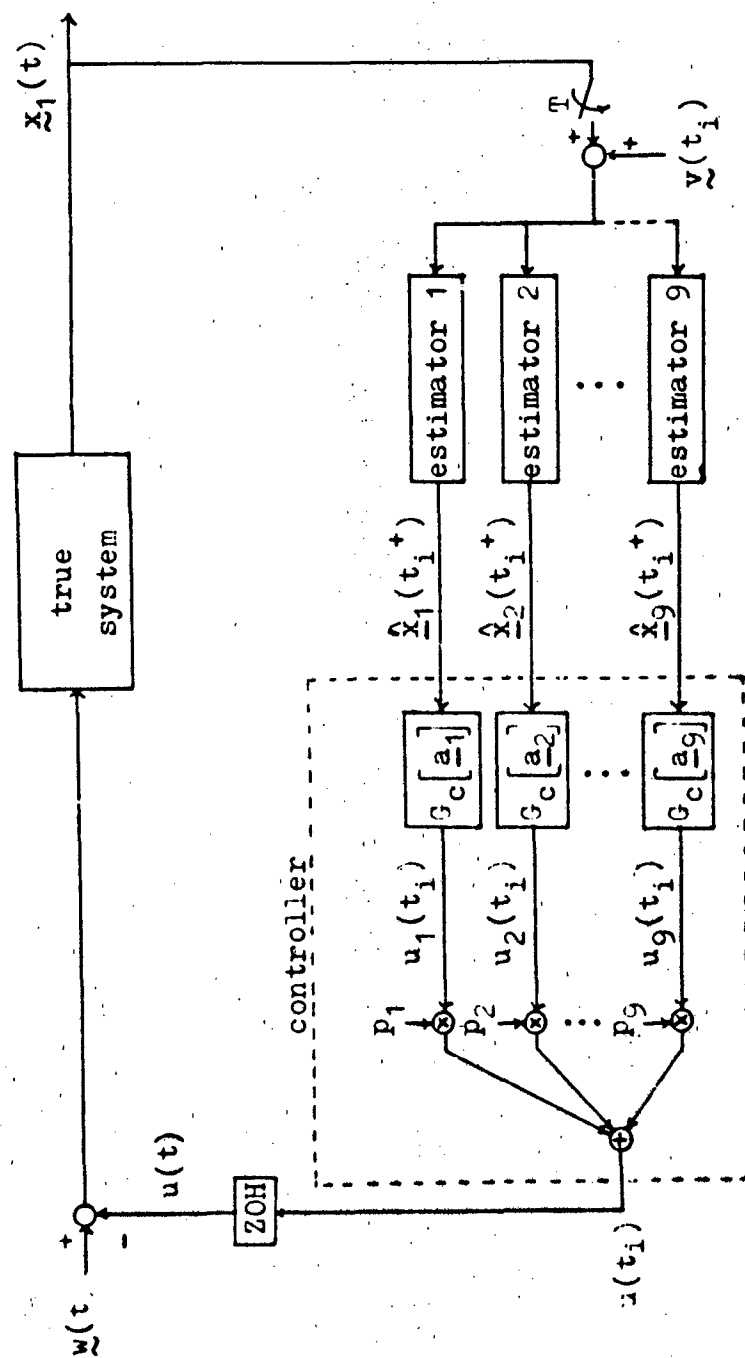


Figure III-4. Sliding Bank Multiple Model Adaptive Controller

IV. Simulation

IV.1. Introduction

A means of evaluating the performance of the sliding bank estimator/controller is needed. The evaluation is provided by a computer simulation of the true system and the sliding bank estimator/controller. In this chapter, the simulation is outlined. Included is a discussion of Monte Carlo analysis, a brief discussion of the software used to accomplish the simulation (see Appendix C for more details on the software), and finally the plan for analyzing the performance of the various move/expand/contract decision schemes and controllers.

IV.2. Monte Carlo Analysis

The purpose of a Monte Carlo analysis is to generate statistical information about the performance of the estimator/controller under study. The Monte Carlo analysis essentially generates many samples of the random variables of interest and uses these samples to compute means and covariances.

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

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

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

In this model, Φ , B_d , and G_d are all functions of the true parameter vector $\underline{a}_t$ where

$$\underline{a}_t = \begin{bmatrix} \zeta_t \\ \omega_t \end{bmatrix}$$

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

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

where

$H = [1 \ 0]$ is the measurement matrix

$\underline{v}(t_i)$ is a discrete zero-mean white Gaussian measurement noise of variance R .

The driving noise $\underline{w}(t_{i-1})$ and the measurement noise $\underline{v}(t_i)$ are provided by the simulation. A zero-mean Gaussian random variable with a variance of 1 can be approximated by

$$\underline{z} = \sum_{i=1}^{12} y_i - 6 \quad (\text{IV-1})$$

where

y_i is a random variable that is distributed uniformly between 0 and 1 (available through a random number generator)

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

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

where the elements of $\underline{r}$ are computed by Equation (IV-1) and

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

The true system and estimator/controller are operated from initial time t_0 to final time t_f for many runs. These runs are then used to compute the sample means and covariances of any random variables of interest: the number of runs is chosen large enough that these averages over N samples are good approximations to ensemble averages (expectations).

Figure IV-1 depicts a simulation of the estimator alone. Here, the variables of interest are the errors $\underline{e}(t_i)$ between the true system states $\underline{x}(t_i)$ and the estimates of the states $\hat{\underline{x}}(t_i)$. The error $\underline{e}(t_i)$ gives a measure of the performance of the estimator. The mean of $\underline{e}(t_i)$ is computed by

$$E\{\underline{e}(t_i)\} \approx \hat{\underline{m}}_e(t_i) \triangleq \frac{1}{N} \sum_{k=1}^N \underline{e}_k(t_i)$$

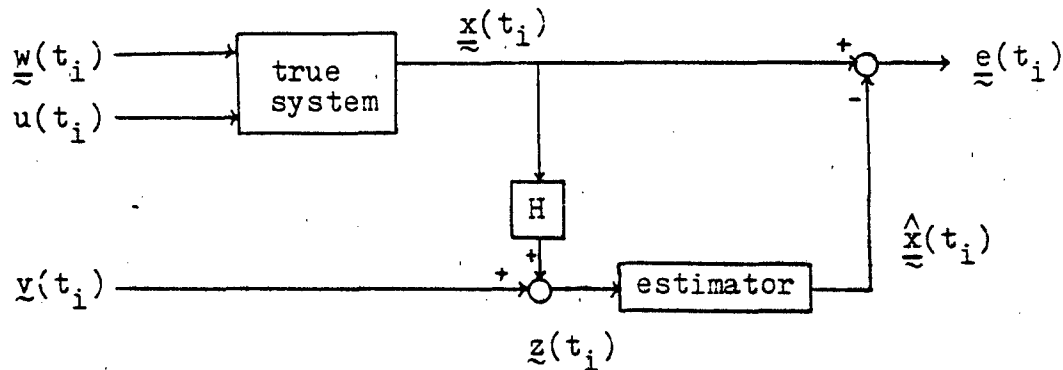


Figure IV-1. Estimator Simulation

where N is the total number of runs performed in the Monte Carlo analysis and $\underline{e}_k(t_i)$ is the value of $\underline{e}(t_i)$ during the k^{th} run. The covariance of $\underline{e}(t_i)$ is computed by

$$E\{[\underline{e}(t_i) - E\{\underline{e}(t_i)\}][\underline{e}(t_i) - E\{\underline{e}(t_i)\}]^T\} \approx$$

$$\hat{P}_e(t_i) \triangleq \frac{1}{N-1} \sum_{k=1}^N \underline{e}_k(t_i) \underline{e}_k^T(t_i) - \frac{N}{N-1} \hat{\underline{M}}_e(t_i) \hat{\underline{M}}_e^T(t_i)$$

In order to study the estimator/controller combination, an appropriate variable of interest is the true system state $\underline{x}(t_i)$, since in this study it is desired to regulate the true system to the quiescent state. Figure IV-2 depicts this. It may also be desirable to examine the control input $\underline{u}(t_i)$ in order to detect unreasonable control levels. The means and covariances of $\underline{x}(t_i)$ are computed in the same manner as above:

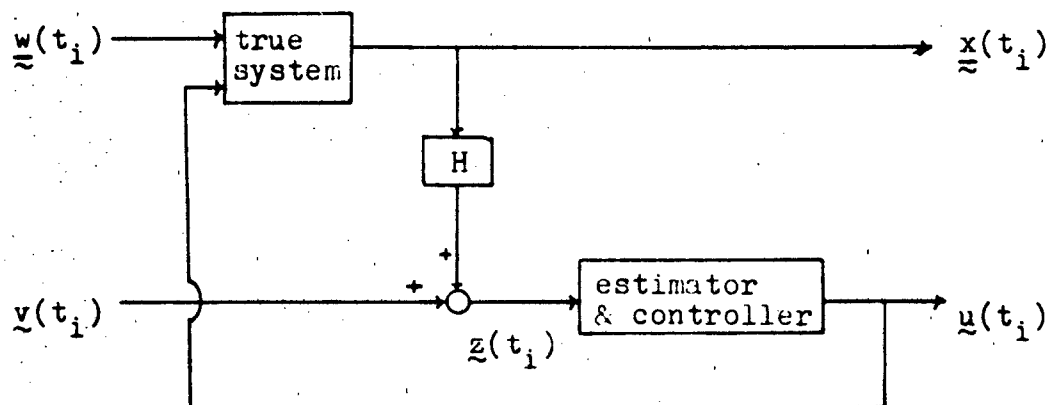


Figure IV-2. Estimator and Controller Simulation

$$\hat{\underline{M}}_x(t_i) = \frac{1}{N} \sum_{k=1}^N \underline{x}_k(t_i)$$

$$\hat{\underline{P}}_x(t_i) = \frac{1}{N-1} \sum_{k=1}^N \underline{x}_k(t_i) \underline{x}_k^T(t_i) - \frac{1}{N-1} \hat{\underline{M}}_x(t_i) \hat{\underline{M}}_x^T(t_i)$$

IV.3. Software Structure

In order to accomplish the Monte Carlo analysis, two computer programs were developed. (For a detailed discussion of the software, refer to Appendix C.) The first program performs the simulation runs and generates the data from each run. The second program reads the data from the first program, computes pertinent means and covariances, and then creates plots.

The first program consists of an executive routine which calls a number of subroutines. The executive contains two

main loops. The inner loop accomplishes a single run of the system from t_0 to t_f . Within the inner loop are calls to subroutines that propagate the true system from one sample time to the next, propagate the filters in the bank from one sample time to the next, take a measurement of the true system states, update the filters, calculate the control, make the decision as to whether or not to move/expand/contract the bank, and finally output the variables of interest to a data file. The outer loop performs the number of runs necessary for the Monte Carlo analysis, setting all variables to their proper initial value prior to the start of each run. Inputs to the first program are true system parameters, initial probability weightings, and initial filter states. Additional inputs are the particular move/expand/contract decision method (Sections II.4.1.1 through II.4.1.6) and the associated thresholds.

The second program performs the analysis of the data created by the first program. The means and covariances of the variables of interest are computed for each sample time from t_0 to t_f . Plots of the means $\pm 1\sigma$ are then generated, where σ is the standard deviation of the variables of interest.

IV.4. Simulation Plan

In order to study the performance of the sliding bank estimator/controller in a systematic manner, the performance analysis is divided into two parts. First, the performance of the estimator alone (i. e., without feedback control) is evaluated. Upon completion of the estimator evaluation, the "best"

estimator is used in conjunction with each of the controllers to determine the best adaptive estimator/controller.

The performance analysis of the estimator alone is accomplished by driving the true system with a zero-mean white noise alone and then with white noise plus a dither signal for each of the various move/expand/contract decision schemes. The dither signal is used to enhance system identifiability, over the noise-only case, by sufficiently and persistently exciting the true system modes with a known input. The analysis is performed with the true system parameters taking on a number of different values within the parameter space. The parameters will:

- a. be constant over t_0 to t_f and equal to one of the discretized points in the parameter space; or
- b. be constant over t_0 to t_f and between discretized points in the parameter space; or
- c. undergo a jump in value at t_j where $t_0 < t_j < t_f$; or
- d. vary continuously over t_0 to t_f .

The performance analysis for these various true parameter cases is intended to give an indication of how well the estimator performs when the true parameter is at one of the discretized parameter points vs. when the true parameter is between discretized parameter points (i. e., no one filter in the bank is "perfectly matched" to the real world environment). Additionally, the determination of how well the estimator adapts to jumps in parameter values (as due to catastrophic changes in a system) and how well the estimator can track slowly varying parameters (as due to an aerospace vehicle changing

its operating point within its flight envelope) can be accomplished. Table IV-1 displays the estimator-only performance analysis plan.

Upon completion of the estimator-only study, the results will be qualitatively analyzed to determine the "best" move/expand/contract decision scheme (as developed in Sections II.4.1.1 through II.4.1.6). The determination of "best" will be based on fastest state and parameter acquisition time, lowest state estimate error biases, and lowest state estimate error variances. Conclusions will be drawn about the performance of the various move-the-bank decision methods, whether or not filter warm-up (Section II.4.2) offers any performance improvement, and whether it is more appropriate to have an acquisition cycle (Section II.4.1.5) or simply to let the bank move itself to the appropriate region of the parameter space, corresponding to the true parameter point position. The best move decision method should consistently have the fastest acquisition time, lowest variances, and lowest bias errors. If filter warm-up is to be effective, it should substantially reduce state estimate error biases and variances in addition to improving acquisition times. The acquisition cycle should significantly reduce the time required to establish precise state and parameter estimates to prove its effectiveness.

When the best move/expand/contract decision method has been determined, the estimator (using that method) will then be used in conjunction with each of the following controllers:

- a. single changeable-gain controller (Section III.3)

TABLE IV-1

Estimator Analysis Plan

Decision Mode	Filter Warm-up	True Parameters			
residual monitoring	no	D	B	J	V
	yes	D	B	J	V
parameter position estimate monitoring	no	D	B	J	V
	yes	D	B	J	V
parameter position and velocity estimate monitoring	no	D	B	J	V
	yes	D	B	J	V
probability monitoring	no	D	B	J	V
	yes	D	B	J	V
covariance monitoring *	no	D	B	J	V
	yes	D	B	J	V

D: constant and equal to discretized parameter points

B: constant and between discretized parameter points

J: jump

V: varying

* The covariance monitoring acquisition cycle is to be used with the best of the four move decision modes.

- b. single fixed-gain controller (Section III.4)
- c. sliding bank multiple model adaptive controller (Section III.5)

As with the estimator-only analysis, the estimator/controller combinations will be evaluated under various true parameter characteristics. Table IV-2 is the estimator/controller analysis plan.

IV.5. Summary

The simulation of the sliding bank estimator/controller was discussed in this chapter. Monte Carlo analysis, simulation software, and the simulation plan were presented. The results of the simulation are presented in the next chapter.

TABLE IV-2
Controller Analysis Plan

Controller	True parameter			
	D	B	J	V
single changeable-gain controller				
single fixed-gain controller				
sliding bank multiple model adaptive controller				

D: constant and equal to discretized parameter point

B: constant and between discretized parameter points

J: jump

V: varying

V. Results

V.1. Introduction

This chapter presents the results of the Monte Carlo simulation. The initial goal of the simulation was to determine the "best" of the various move/expand/contract decision methods for the sliding bank multiple model adaptive estimator. Once the best estimator move scheme was determined, that scheme was used for the estimator in conjunction with each of the three controller designs.

The true system driving noise strength used throughout the simulation was $Q = 1.0$ (refer to Equation II-3). The measurements, taken every 0.01 second, were corrupted by a discrete-time zero-mean white Gaussian noise with a variance of $R = 0.01$. Each decision method was exercised with the true system parameters being defined as: equalling one of five different discretized parameter points, equalling one of two different points that were between discretized parameter points, undergoing a jump change, and varying slowly (see Table V-1). The particular parameter points chosen in Table V-1 were selected to cover the parameter space: one point near the center, others on the boundary (where the bank cannot center itself on the true parameter point), some points near the boundary (where the bank can center itself on the true parameter point), and points between discretized parameter values.

The center of the bank at the beginning of each run was always placed at the discrete parameter space coordinates

TABLE V-1
True Parameters

	ζ	ω (rad/sec)	discretized parameter space coordinates
1	0.0000	10.48	(1, 3)
2	0.1111	48.65	(2, 9)
3	0.4444	13.54	(5, 4)
4	0.8889	8.12	(9, 2)
5	1.0000	62.83	(10, 10)
6	0.0700	9.00	
7	0.9300	41.00	
8	0.0700 jumps to 0.9300	9.00 41.00	
9	0.0700 varies to 0.9300	9.00 41.00	

(5, 5), corresponding to $\zeta = 0.4444$ and $\omega = 17.48$ rad/sec. The true system was quiescent at the beginning of each run (however, the option of an excitation input to enhance identifiability was available). The initial state estimate for each elemental filter was $\hat{x}_k(0) = \underline{0}$ and the initial probability weightings were $p_k(0) = 1/9$ for $k = 1, \dots, 9$.

The Monte Carlo analysis is based on 100 runs of the system, for $0 \leq t \leq 8$ seconds. The same seed number for the random number generator was used for each 100-run simulation.

Section V.2 describes estimator performance and Section

V.3 discusses the controller performance. Finally, Section V.4 presents controller computational loading comparisons.

V.2. Estimator Performance

The performance of the sliding bank estimator, operating under the various move/expand/contract decision methods, is discussed in this section. The ability of the sliding bank estimator to provide adequate state estimates is the criterion used to determine the best performing decision method. The state estimate errors, for each of the decision methods, are compared to a performance bound of a single Kalman filter that has artificial knowledge of the true parameters. Figures A-1a through A-1g of Appendix A are plots of the state estimate errors of the benchmark, a single Kalman filter. An attempt was also made to compare the performance of the sliding bank estimator to that of the full bank multiple model adaptive estimator. However, the full bank approach required such an excessive amount of computation time that only ten Monte Carlo runs could be performed. The resulting plots were so noisy (because of the low number of runs) that comparisons were impossible.

For each of the decision methods, the true system was first driven by the white noise signal alone. However, due to the relatively low driving noise strength and the fact that only noise-corrupted measurements of the position type variable were available, the estimator experienced considerable difficulty identifying the true system parameters. Additionally, the difficulty in system identification is not apparent when

comparing the state estimate error plots of the sliding bank estimator to those of the benchmark. Because the true system modes are not sufficiently excited by the noise signal and because the measurement noise strength is quite low relative to the driving noise strength, the difficulty in system identification appears to have little effect on the estimator's ability to provide good state estimates, when the true system is driven by the noise signal alone. As a result, with the noise signal alone driving the true system, there is little or no basis of comparison between the various decision methods.

In order to aid system identification, the true system was next driven by the noise signal plus a 20 Hz square wave dither signal with an amplitude of 25. The dither signal was not chosen through a rigorous analysis of optimal inputs for system identification, as that topic is beyond the scope of this effort, but rather by several trial and error iterations. The 20 Hz square wave signal was found to enhance the estimator's ability to identify the parameters for all true parameter points exercised in the simulation. The dither signal had the additional benefit, from the standpoint of decision method comparisons, of indicating in the state estimate error plots whether or not the estimator had identified the system parameters. When the sliding bank estimator has properly identified the true system parameters, the bias errors of the estimator disappear and the effects of the dither signal are no longer present. While it may appear that performance is degraded with the dither signal when comparing the dither vs.

non-dither state estimate error plots, system identification is in fact improved. Without the dither signal, system identification is degraded and the state estimate error plots do not clearly depict what is happening. With the dither signal, system identification is enhanced, and it becomes quite evident from the state estimate error plots how well the sliding bank estimator is performing. A further explanation of the difference between the state estimate error plots of the dither vs. non-dither cases can be found through examination of the relative magnitudes of the system drivers. The dither magnitude is 25.0 while the rms white noise value is only $Q_d \approx Q\Delta t = 0.01$.

Plots of parameter estimate errors for two of the true parameter points exercised that demonstrate the effect of the dither signal on system identification can be found in Appendix B. In Figures B-1a through B-1d, residual monitoring was used to move the bank. Figures B-1a and B-1b are for the case of the true parameters equalling $(\zeta, \omega) = (1.0, 62.83)$, with the true system driven by the noise signal alone and by the noise plus dither signal, respectively. In Figures B-1c and B-1d, the true parameters are $(\zeta, \omega) = (0.07, 9.0)$. In Figures B-2 through B-5, parameter position estimate, parameter position and velocity estimate, probability, and probability plus covariance monitoring were used to move the bank. These figures readily demonstrate the enhancement of parameter identification when the dither signal is applied.

The primary objective of this effort is not to design a

good parameter estimator, but instead to design a good state estimator. Hence, the state estimate errors were chosen as the basis of comparison for the various decision methods. In the discussions to follow, the state estimate error plots, with the true system driven by the noise plus dither signal, are used to compare the performance of the various decision methods. The primary figure of merit used is acquisition time. This is defined as the time at which the state estimate errors of the sliding bank estimator match the state estimate errors of the benchmark to four significant figures.

Section V.2.1 presents a discussion of threshold setting and Sections V.2.2 through V.2.6 discuss the individual performance of the various decision methods. Section V.2.7 presents the comparisons between the performance of the various decision methods.

V.2.1. Threshold Setting. Each of the move/expand/contract decision methods required the determination of an appropriate threshold level within the decision logic. The threshold chosen was required to provide adequate performance for each of the true parameter points exercised. It should be noted that the thresholds have a more pronounced effect on where the bank is centered than on where $\hat{a}(t_1)$ is located. This is due to the manner in which the probability weightings are calculated when the bank is moved (Section II.4.2 and Equations II-11 and II-12). Since, as a minimum, four of the filters will retain the probability weightings and a_k values that they had before the move, some continuity of the $\hat{a}(t_1)$ cal-

ulation (Equation I-8) is maintained during a move.

As examples of threshold level characteristics, parameter estimate error plots for a single run simulation are included in Appendix B. The appendix contains three plots for each decision method: one with the threshold set too high, one with the threshold set at the selected value, and one with the threshold set too low. All plots are for the true parameters equalling $(\zeta, \omega) = (0.0, 10.48)$ with the true system driven by the noise signal plus the dither signal.

The threshold level selected for the residual monitoring move method (Section II.4.1.1) was 7.0. When the smallest of the nine likelihood quotients (Equation II-10) exceeds 7.0, the bank is moved in the direction of the elemental filter with the smallest likelihood quotient. When the threshold is set too high, the sliding bank takes longer to identify the system parameters. (Compare Figure B-6a where the threshold is set at 14.0 to Figure B-6b where the threshold is set at 7.0.) When the threshold is set too low, the sliding bank fails to maintain "lock" on the true parameters. (Compare Figure B-6c where the threshold is set at 1.0 to Figure B-6b.)

Parameter position estimate monitoring (Section II.4.1.2) required a threshold level of 0.04. When the estimated parameter position (Equation I-8) is farther than 0.04 from the center of the bank, the bank is moved in the direction of the estimated parameter position. Again, when the threshold is too high, as in Figure B-7a, where it is set at 0.111, the bank takes longer to move to the appropriate region of the

parameter space. When the threshold is set too low, it also takes longer to acquire the true parameters. Additionally, the sliding bank tends to oscillate between discretized parameter points (see Figure B-7c, where the threshold is set at 0.008).

A level of 0.04 was also chosen for the threshold of the parameter position and velocity estimate monitoring decision method (Section II.4.1.3). When the predicted position (one sample period ahead) of the parameter position estimate is farther than 0.04 from the center of the bank, the bank is moved in the direction of the predicted parameter position estimate. When the threshold is set too high (0.111), as in Figure B-8a, the sliding bank takes longer to acquire the true parameters than when it is set at the proper level as in Figure B-8b. Here, a tradeoff was made between acquisition time and the tendency of the bank to loose lock under this decision method; the excursion from the true parameter position at around 6.5 seconds is larger in Figure B-8b than in Figure B-8a. When the threshold is set too low (0.01 for example), as in Figure B-8c, the bank has an even greater tendency to loose lock.

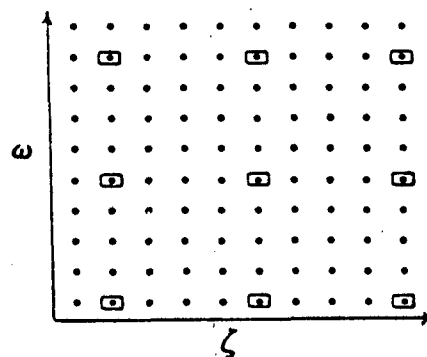
Probability monitoring (Section II.4.1.4) required a threshold of 0.05. When the largest probability weighting on any of the elemental filters exceeds 0.05, the bank is moved in the direction of that filter, unless that filter is the center filter. With this decision method, as the threshold is decreased from high values to low values performance improves. Compare Figure B-9a, where the threshold is set at

0.8, with Figure B-9b. No improvement in performance is gained over the selected value of 0.05, however, when the threshold is set at its lowest value of 0.0 (compare Figure B-9c). Essentially, with these low threshold values, the bank moves anytime the probability weightings on the perimeter elemental filters exceed the weighting on the center filter.

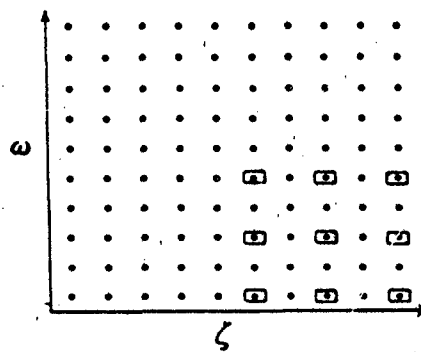
A different acquisition strategy than that of the other decision methods is used with covariance monitoring. With the other methods, the spacing between the discretizations in parameter space of the elemental filters of the bank is constant and at its smallest value for the entire run. The sliding bank starts the run with its center near the center of the parameter space and then uses the various decision modes to move the bank to the proper position in the parameter space, i. e., corresponding to the true parameter point. The covariance monitoring acquisition strategy on the other hand does not keep the spacing the same for all time. The spacing is initially set such that the sliding bank covers nearly the entire parameter space (a coarse discretization for acquisition). The covariance of the parameter estimate is then used to contract the bank as appropriate. (Refer to Section II.4.1.5.) In this way, the true parameter would always be within the region spanned by the large initial sliding bank, perhaps offering improved initial performance over the other methods, when the true parameter initially is outside the space spanned by the small initial sliding bank (except when the true parameter point is located at $(0, 4)$ in parameter space).

The scheme used in the simulation, was to contract the spacing between discretizations of the elemental filters in two steps. The bank initially has a spacing of four between discretizations of elemental filters, as in Figure V-1a. When the covariance of the parameter estimate drops below the threshold level (comparison to a scalar threshold is discussed in the next paragraph), the bank then contracts to an intermediate spacing of two, as in Figure V-1b. Finally, when the covariance of the parameter estimate drops below a second threshold, the bank contracts to the finest discretization, a spacing of one, as in Figure V-1c. When the bank contracts, it contracts to the quadrant of the bank before contraction that contains the estimated parameter point position. This ensures that four filters of the bank before contraction remain in the bank after contraction, in order to minimize the impact of initial transients in the state estimators (see Figure V-2). Another option would be to contract the bank such that its center is the discretized parameter point nearest to $\hat{a}$, but this would have the disadvantage of all of the nine filters generally going through an initialization transient.

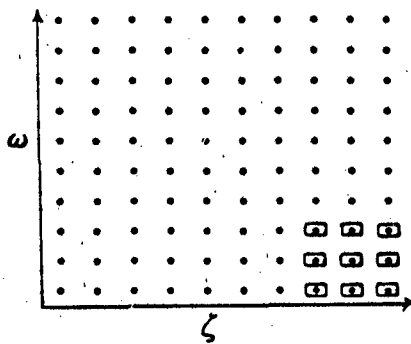
One should note that the covariance of the parameter estimate is a two-by-two matrix (Equation I-9). In order to compare this matrix to the threshold level, a scalar value associated with the matrix is needed. A function of the two diagonal elements (the variance of the ζ estimate and the variance of the ω estimate) is a strong candidate. For the simulation, what was used was the larger of the two diagonal



a.



b.



c.

elemental filter: 

Figure V-1. Bank Discretizations. a. Initial-Spacing = 4. b. Intermediate-Spacing = 2. c. Final-Spacing = 1.

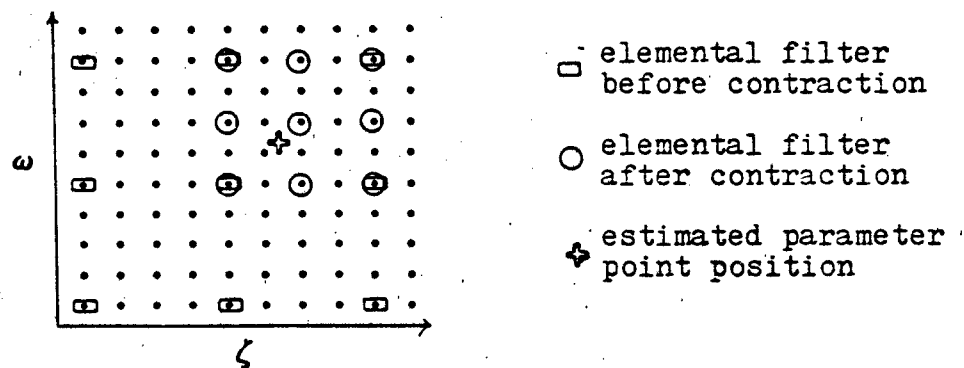


Figure V-2. Bank Contraction

elements, after performing an inverse mapping on the ω estimate variance to give both elements the same relative magnitude.

The spacing between adjacent ζ parameters is 0.1111, while the spacing between adjacent ω parameters is dependent upon where they are in the parameter space (recall that the ω parameters are discretized logarithmically). However, the difference between the logarithms of adjacent frequencies (in Hz) is 0.1111:

$$|\log(f_1) - \log(f_2)| = 0.1111$$

where f_1 and f_2 are adjacent frequencies, or since $\omega = 2\pi f$

$$\left| \log \frac{\omega_1}{2\pi} - \log \frac{\omega_2}{2\pi} \right| = 0.1111$$

In order to force the relative magnitudes of the $\hat{P}_{a_{11}}$ and $\hat{P}_{a_{22}}$

terms to be the same, an inverse mapping of the ω parameter must be performed. In other words, the problem is rescaled so that equal $P_{\hat{a}_{11}}$ and $P_{\hat{a}_{22}}$ correspond to equal distances as depicted on the parameter space plots of Figures V-1 and V-2.

From Equation (I-9) we have

$$\begin{aligned} P_{\hat{a}_{22}} &= \sum_{k=1}^9 [a_{2k} - \hat{a}_2(t_i)]^2 p_k(t_i) \\ &= \sum_{k=1}^9 [\omega_k - \hat{\omega}(t_i)]^2 p_k(t_i) \end{aligned}$$

where the order of the $\omega_k - \hat{\omega}(t_i)$ terms vary according to their relative location in the parameter space. A modified $P_{\hat{a}_{22}}$ that will have the same relative magnitude of $P_{\hat{a}_{11}}$ is

$$\begin{aligned} P'_{\hat{a}_{22}} &= \sum_{k=1}^9 \left[\log \frac{\omega_k}{2\pi} - \log \frac{\hat{\omega}(t_i)}{2\pi} \right]^2 p_k(t_i) \\ &= \sum_{k=1}^9 \left[\log \frac{\omega_k}{\hat{\omega}(t_i)} \right]^2 p_k(t_i) \\ &= \sum_{k=1}^9 \left[\log \frac{a_{2k}}{\hat{a}_2(t_i)} \right]^2 p_k(t_i) \end{aligned}$$

The bank is contracted when

$$\max(P_{\hat{a}_{11}}, P'_{\hat{a}_{22}}) \leq T$$

where T is the contract threshold.

The threshold levels used for the covariance monitoring acquisition cycle were 0.08 for the contraction from the coarse discretization to the intermediate discretization, and 0.0375 for contraction from the intermediate discretization to the final discretization. If the thresholds were set too high, as in Figure B-10a, where they are set at 0.15 and 0.06, the bank contracts too soon, before a good estimated parameter position is available, and frequently contracts to the wrong quadrant, leaving the bank in no better position than it would have been at the beginning of a run without the acquisition cycle. When the thresholds are set too low as in Figure B-10c, where they are set at 0.003 and 0.0007, the bank does not contract until well into the run, degrading performance while it is at coarser discretizations.

An additional threshold was needed for the covariance monitoring acquisition cycle. The bank was given the option of expanding in order to handle the case of jump parameters. When the bank expands, it is set at the initial coarse discretization with its center at (5, 5). The probability weightings for each elemental filter are reset to 1/9 and the $\hat{x}_k$'s set to $\hat{x}$. Essentially, the sliding bank starts the acquisition cycle over when an expand decision is made. The expand decision is governed by residual monitoring (Section II.4.1.1). Residual monitoring is used because the needed information (that state estimates have gone awry) is not available from either the covariance of $\hat{a}$ or $\hat{a}$ itself. When

the smallest of the likelihood quotients exceeds the expand threshold, the bank expands. The expand threshold value used in the simulation was 25.0 (Figure B-11b). When the expand threshold was set too high (1000.), it waits too long after the parameters have undergone a jump in value before expanding, thereby degrading performance (see Figure B-11a). When the expand threshold was set too low (at 7.0 for example), the bank expands inappropriately during the run (see Figure B-11c).

When filter warm-up is used (Section II.4.2), an additional set of thresholds are needed. In determining the thresholds for filter warm-up, care was taken not to set the warm-up thresholds too low, as this would have meant that the wrong set of filters would consistently be warmed up (i. e., the "move" decision would later go to a different set of filters) or that the estimator would frequently change the filters being warmed up, both having the effect of not warming up any filters at all despite the additional computational loading and complexity. The warm-up threshold also could not be set too close to the move threshold, as this would mean that the decision to warm up filters would be made concurrently with the move decision, again having the effect of not warming up any filters. Table V-2 summarizes the move and warm-up thresholds used in the simulation. Note that 0.0 was used as the warm-up threshold for probability monitoring. Because the move threshold is so low and because performance improves when the threshold is lowered, the lowest possible value of 0.0 is an appropriate choice for the warm-up threshold.

TABLE V-2

Thresholds

decision method	move	warm-up
residual monitoring	7.0	3.0
parameter position estimate monitoring	0.04	0.02
parameter position and velocity estimate monitoring	0.04	0.02
probability monitoring	0.05	0.0
covariance monitoring coarse - intermediate	0.08	0.1
covariance monitoring intermediate - fine	0.0375	0.05
covariance monitoring expand	25.0	not used

V.2.2. Residual Monitoring. Figures A-1a through A-1g of Appendix A are plots of the state estimate errors of the benchmark, a single Kalman filter that has artificial knowledge of the true parameters. Figures A-2a through A-2i are plots of the state estimate errors of the sliding bank estimator using residual monitoring to move the bank, with the true system driven by the noise signal alone. When these are compared to Figures A-1a through A-1g, it is seen that there are two fewer benchmark plots than sliding bank plots; there are no benchmark plots for the cases of jump parameters and varying parameters due to computational resource limits. However,

the jump parameter case and the varying parameter case start and end with the same parameters as in benchmark Figures A-1f and A-1g respectively, so there is some basis of comparison for this and all other proposed filter designs.

As was mentioned previously, Figures A-2a through A-2i do not shed significant light on the sliding bank's performance. With the exception of Figure A-2a, where the true parameter point is at (1, 3), there is little difference between the sliding bank plots and the benchmark plots. In Figure A-2a, the sliding bank failed even to come close to identifying the parameters, so that its poor performance is apparent even on the state estimate error plot with the true system driven by the noise signal alone.

Figures A-3a through A-3i are the state estimate error plots with the true system driven by the noise plus the dither signal. When the sliding bank estimator has properly identified the true system parameters, the bias errors of the estimator disappear and the effects of the dither signal are no longer present. In Figure A-3c, the parameters are identified very rapidly and the graphs match the benchmark of Figure A-3a for nearly the entire 8 second interval. (Recall that the sliding bank is initially centered at (5, 5) and in Figures A-1c and A-3c, the true parameter is at (5, 4).) In Figure A-3d, the parameter is almost identified within two seconds, but a slight bias error still remains over the remainder of the interval. In this run, the true parameter is never fully identified; the sliding bank never centers itself

on (9, 2), the true parameter position. The true frequency is identified after two seconds but the true damping ratio is never identified. In Figure A-3e, the performance is poor over the entire eight second interval. Again, the true frequency is identified rapidly, but again the damping ratio is never identified. Figure A-3f and A-3g display the errors for the case of the true parameters lying between discretized parameter points. In Figure A-3f, the sliding bank moves to the vicinity of the true parameter point in just over one second and in Figure A-3g, in just over two seconds. For both A-3f and A-3g, a small bias error remains for the rest of the interval. This is due to the parameters lying between discretized parameter points; there is no elemental filter in the sliding bank which matches exactly the true parameter point. In Figure A-3h, the parameter undergoes a jump change at four seconds. In Figure A-3i, the parameter varies linearly from $(\zeta, \omega) = (0.07, 9.0)$ to $(0.93, 41.0)$. In this case, the sliding bank always lags behind the true parameter resulting in bias errors over the entire interval.

In all cases the sliding bank estimator had little difficulty in identifying the natural frequency parameter, but had significant difficulty in identifying the damping ratio parameter. A heuristic explanation for this can be found through consideration of a power spectral density plot of the output of a second order system driven by white noise (see Figure V-3). As the natural frequency of the system varies, the break frequency ω_b will shift, which is easy to identify. When the

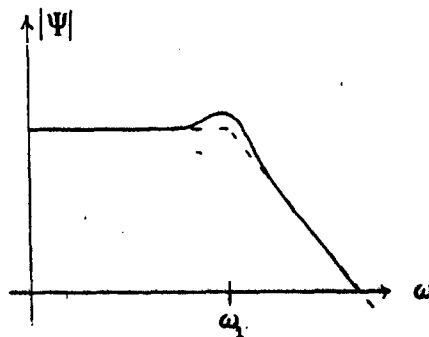


Figure V-3. Power Spectral Density of 2nd Order System

damping ratio shifts, however, only the peak of the spectral curve will shift, and, when only noise corrupted measurements of the position type state are available, this shift is much harder to identify.

A more rigorous approach was taken to explain this further. An ambiguity function analysis was performed for each of the true parameter points exercised in the simulation. (For a detailed discussion of the ambiguity function analysis, see Appendix D.) Evaluation of the ambiguity function comes from an indication of the sensitivity of a single Kalman filter based on a particular parameter value to changes in the true system parameters. This function provides insights to system identifiability with adaptive estimators. Sharp curvature of the ambiguity function means that system identification will be relatively easy and precise. Flat curvature, or multiple peaks of the ambiguity function mean that system

identification will be more difficult.

Ambiguity function plots for each of the true system parameters exercised in the simulation can be found in Appendix D (Figures D-1 through D-7). In all of these plots, there is a sharper curvature in the ω direction than in the ζ direction. From these plots, one would expect an adaptive estimator to have less difficulty identifying the natural frequency parameter but greater difficulty identifying the damping ratio parameter. This is borne out by the results seen in the simulation.

In Figures A-4a through A-4i, the true system was driven by the noise signal plus the dither signal as used in Figures A-3a through A-3i; however, a warm-up of elemental filters before moving was performed. In all cases, the warm-up only slightly improved performance. The peaks in the variances of the errors were only slightly reduced during acquisition (compare Figure A-4a with A-3a), not enough to warrant the additional computation time required to warm up elemental filters before moving.

V.2.3. Parameter Position Estimate Monitoring. Figures A-5a through A-5i are graphs of the state estimate errors of the sliding bank using parameter position estimate monitoring, with the true system driven by the noise signal alone. Figures A-6a through A-6i are plots of the state estimate errors with the true system driven by the noise plus dither signal. In Figures A-6a through A-6c, the true parameters are identified in under 1.0 second and from 1.0 to 8.0 seconds, the graphs

compare identically with the benchmarks, Figures A-1a through A-1c. In Figure A-6d, the acquisition time is slower; the parameters are identified by 2.5 seconds. In Figure A-6e, the parameters are not identified until 4.0 seconds. In Figures A-6f and A-6g, the parameters are identified as nearly as is possible in 1.0 and 2.0 seconds respectively. Figure A-6h represents a jump in parameters and A-6i represents varying parameters.

In all but the varying parameter case, the true parameter was eventually identified. In some instances, the acquisition took much longer, but this again is explained by the ambiguity function analysis (recall Appendix D).

Figures A-7a through A-7i depict estimation errors of the sliding bank using parameter position estimate monitoring with filter warm-up, with the real world model driven by the noise plus dither. There is no detectable improvement in performance over the no warm-up case (Figures A-6a through A-6i). Upon examination of the results from a single run of the simulation, it was found that the estimator would sometimes decide to move; at other times, it would warm up the wrong filters, and at other times it would switch back and forth between several warm-up directions. When any of these events occurred, the effect was equivalent to not warming up any filters at all. On the rare instances that the proper filters did warm up, the state estimates differed only in the third decimal place. As was true with residual monitoring, filter warm-up does not offer a performance advantage to warrant the

additional computation time.

V.2.4. Parameter Position and Velocity Estimate Monitoring. Figures A-8a through A-8i are the plots of the state estimate errors of the sliding bank using parameter position and velocity estimate monitoring, with the true system driven by the noise signal alone. As with the previous sections, there is virtually no difference between these graphs and the benchmark graphs, with the exception of Figure A-8a. In Figure A-8a, there is a slight increase in the variance of the error in the velocity-type variable near 4.0 seconds.

In Figures A-9a through A-9i, the true system is driven by the noise signal plus the dither signal. The performance is roughly equivalent to that of parameter position estimate monitoring; however, there is an increased tendency for the bank to wander away from the true parameter position, as in Figures A-9a, A-9c, A-9d, and especially A-9f.

Figures A-10a through A-10i are graphs of the state estimate errors for position and velocity estimate monitoring with the true system driven by the noise signal plus the dither signal and with filter warm-up being used before moving the bank. Again there is no appreciable difference between the warm-up and no warm-up cases. With a slight reduction of the threshold to allow for warm-up, the bank exhibits oscillatory characteristics and consistently warms up the wrong filters or continually changes the filters being warmed up, having the effect of not warming up any filters at all. If the warm-up threshold is increased to a level where the oscillations are

no longer present, the warm-up threshold will be so near the move threshold that a decision to warm up filters will be made concurrently with a move decision.

V.2.5. Probability Monitoring. Figures A-11a through A-11i depict the state estimate errors of the estimator using probability monitoring with the true system driven by the noise signal alone. Figures A-12a through A-12i are the error plots for the true system driven by the noise signal plus the dither signal. In Figure A-12a, the system parameters are identified in approximately 0.5 second. There is a slight increase in the variance just before 2.0 seconds due to an excursion of the bank from the true parameter point, but recovery is quite rapid and the bank remains locked on the true parameter value for the remainder of the 8.0 second run. In Figure A-12b, the true parameters are identified in approximately 0.75 second. In Figures A-12c and A-12d, the parameters are again identified in under 1.0 second. In Figure A-12e, the parameters are not identified until sometime between 3.0 and 4.0 seconds and again the flat curvature of the ambiguity function explains the difficulty in identification for this particular true parameter point. When the true parameter point lies between discretized values, as in Figures A-12f and A-12g, the true parameters are identified as nearly as is possible in less than 3.0 seconds. Figures A-12h and A-12i are graphs of the estimation errors for a jump in true parameter value and varying parameters, respectively.

As was seen previously, there is no performance gain

when filter warm-up is used. Figures A-13a through A-13i are the plots of estimation errors with the true system driven by the dither signal plus the noise signal, using filter warm-up. Here, because the move threshold is so low, the decision to warm up any filters is made at the same time a move decision is made and no time is available to warm up any filters.

V.2.6. Covariance Monitoring. The covariance monitoring acquisition cycle was used in conjunction with the probability monitoring move decision method. Probability monitoring was chosen as it offered the best overall performance of any of the other decision methods (this conclusion will be explained in Section V.2.7). Throughout the acquisition period (when the bank is expanded) the bank still is given the option of moving; however, priority is given to bank contraction in the decision logic, as performance is degraded while the bank is expanded (recall expansion decision via residual monitoring discussion).

Figures A-14a through A-14i depict the case of the truth model driven by the noise signal alone. Figures A-15a through A-15i are plots of the estimation errors when the true system was driven by the noise signal plus the dither signal. Performance was significantly degraded in Figure A-15a over that achieved in Figure A-12a for probability monitoring alone. What happened in this case was that the bank contracts to the wrong quadrant when going from the initial to intermediate discretization. Even though the bank could still move through the parameter space while at the intermediate discretization,

it would typically contract again to the finest discretization before moving. The bank was then left in no better position than it would have been at the beginning of a run where the covariance monitoring acquisition cycle was not used. This same degradation in performance, although not quite as severe, can also be found in Figures A-15c, A-15d, A-15f, and A-15i.

There is a performance improvement with the covariance monitoring acquisition cycle, however, with Figures A-15b, A-15e, and A-15g. In these cases, the bank quickly contracted to the proper quadrants, leaving the contracted bank in the proper region of the parameter space relative to the true parameter point. This occurred more rapidly than simply allowing a contracted bank to move itself from the center of the parameter space to the proper position in the parameter space, as in Figures A-12b, A-12e, and A-12g. In Figure A-15h, the performance was degraded during the first 4.0 seconds of the run, but was improved over the non-acquisition cycle case (Figure A-12h) during the last 4.0 seconds of the run. When the parameters underwent a jump in value at 4.0 seconds, the residuals grew quite large, causing the bank to expand. This second acquisition cycle for the new parameter values resulted in a performance improvement.

Figures A-16a through A-16i represent the case of the truth model driven by the noise signal plus the dither signal with filter warm-up. There is no difference in performance between filter warm-up and no warm-up. Again, raising the threshold to allow for warm-up meant that the wrong filters

would be warmed up or, when the correct filters were warmed up, no significant improvement in the performance was achieved.

Finding the exact covariance monitoring threshold settings that would improve performance rather than degrade it, for all of the possible true parameter positions (if possible), was not accomplished in the limited time available for this effort. It is noted that the covariance monitoring acquisition cycle results in improved performance when the true natural frequency is high (Figures A-15b, A-15e, and A-15g, where the natural frequency is 48.65, 62.83, and 41.0 rad/sec respectively). When the true natural frequency is low (Figures A-15a, A-15c, A-15d, and A-15f, where the natural frequency is 10.48, 13.54, 8.12, and 9.0 rad/sec respectively), the bank fails to contract to the proper quadrant. In view of this, alternate expand/contract decision methods warrant investigation. Furthermore, in light of the ambiguity function analysis that revealed different precisions in estimating ζ and ω as a fundamental aspect of this problem, perhaps a weighted norm of the two-by-two covariance matrix would offer improved characteristics for threshold evaluations.

V.2.7. Comparisons. The comparisons in this section are based on a qualitative analysis of the figures in Appendix A. Acquisition times are evaluated by comparing each of the sliding bank estimation error figures with the benchmark figures (A-1a through A-1g). When the sliding bank estimation error figure matches the single Kalman filter estimation error figure to four significant figures, that time is picked as the

acquisition time. In some cases, several of the move decision methods had the same acquisition times, but variances and bias errors were different before acquisition. The decision methods that exhibited the lowest variance and lowest bias errors were deemed "best" in these instances. In this section, the only figures compared are those in which the true system is driven by the noise plus dither signal with no filter warm-up.

V.2.7.1. True Parameter Point = $(\zeta, \omega) = (0.0, 10.48)$. Here, the residual monitoring decision mode had an acquisition time of 1.0 second. Rather large biases and variances relative to the other decision modes were exhibited prior to 1.0 second. (Compare Figure A-3a with A-6a; bias errors are four times as large and the variance is fifteen times as large.) The parameter position estimate monitoring method acquired the parameter in 0.56 second with the smallest variance and bias errors. Parameter position and velocity monitoring had an acquisition time of 0.84 second, but had larger variances: five times as large as parameter position estimate monitoring (compare Figure A-9a with A-6a). Additionally, with parameter position and velocity estimate monitoring, the bank made three excursions from the true parameter point after acquisition (total time away: 0.47 second), causing the variance to increase to four times its steady state value. Probability monitoring had, by a slim margin, the quickest acquisition time of 0.53 second, but made one 0.13 second excursion from the true parameter point. Probability monitoring with the covariance monitoring acquisition cycle had the slow-

est acquisition time of 1.06 seconds and had variance and bias errors that were 25 and 12 times larger, respectively, than parameter position estimate monitoring (compare Figure A-15a with A-6a). The best performer for this particular parameter point was parameter position estimate monitoring.

V.2.7.2. True Parameter Point = (0.1111, 48.65).

For this true parameter point, all of the decision methods performed equally, acquiring the parameters in 1.06 seconds, with the exception of probability monitoring with the covariance monitoring acquisition cycle. Probability and covariance monitoring had a slight improvement in performance over the other methods; acquisition time was less (0.63 second) and the initial bias errors were only 1/2 as large (compare Figure A-15b with A-12b, for example).

V.2.7.3. True Parameter Point = (0.4444, 13.54).

In this case, residual monitoring and parameter position estimate monitoring performed quite well. They each had an acquisition time of 0.09 second with very slight bias errors (see Figures A-3c and A-6c). Parameter position and velocity estimate monitoring performed similarly; acquisition time was 0.13 second, but the peak variance was three times larger than its steady state value (compare Figure A-9c with A-3c and A-6c). Probability monitoring had a longer acquisition time of 0.47 second but the bias error was the smallest of any of the methods: 1/2 as large as residual monitoring, for example (compare Figure A-12c with A-3c). The covariance monitoring acquisition cycle had the worst performance. The acquisition

time for covariance monitoring was 1.25 seconds and the bias errors were twice as large as with the other methods (compare Figures A-15c and A-3c).

V.2.7.4. True Parameter Point = (0.3889, 8.12).

At this true parameter point, residual monitoring did quite poorly. Acquisition time was 5.21 seconds with a peak variance six times as large as for probability monitoring and with persistent bias errors as large as five times the maximum bias error committed by probability monitoring (compare Figure A-3d with A-12d). Parameter position estimate monitoring showed better performance with an acquisition time of 2.63 seconds and bias errors only twice as great as probability monitoring (compare Figures A-6d and A-12d). Parameter position and velocity estimate monitoring had the fastest acquisition time of 0.88 second but had two instances where the bank drifted away from the true parameter point (total time away: 0.31 second). Probability monitoring had the next fastest acquisition time of 1.25 seconds. Covariance monitoring had an acquisition time of 3.25 seconds. The best performer in this case was probability monitoring.

V.2.7.5. True Parameter Point = (1.0, 62.83).

Residual monitoring failed altogether in its attempt to identify the true parameter point. Parameter position estimate, parameter position and velocity estimate, and probability monitoring all performed equally, with an acquisition time of 3.63 seconds. Covariance monitoring showed the most promise, with an acquisition time of 2.25 seconds and initial biases only 1/2

as large as the other methods. (Compare Figures A-3e, A-6e, A-9e, A-12e, and A-15e.)

V.2.7.6. True Parameter Point = (0.07, 9.0).

None of the various decision modes completely identified the true parameters. All but parameter position and velocity estimate monitoring identified the parameters as nearly as was possible in 1.38 seconds. Parameter position and velocity estimate monitoring failed to identify the parameters; the variance remained consistently on the order of five times greater than for probability monitoring (compare Figure A-9f with A-12f). Residual monitoring and the covariance monitoring acquisition cycle had larger variances and bias errors (eight and three times larger, respectively) than parameter position and probability monitoring (compare Figures A-3f and A-15f with A-6f or A-12f). Parameter position estimate monitoring and probability monitoring tied for the best performance when the true parameter point was (0.07, 9.0).

V.2.7.7. True Parameter Point = (0.93, 41.0).

For this true parameter point position, residual monitoring failed to identify the parameters. Parameter position estimate monitoring identified the parameters in 3.63 seconds. Parameter position and velocity estimate monitoring took slightly longer: 3.33 seconds. Probability monitoring with the covariance monitoring acquisition cycle did far better than any of the other methods with an acquisition time of 0.25 second.

V.2.7.8. Jump and Varying Parameters. For the case of jump parameters, each of the methods performed as they

did in Section V.2.7.6 for the first 4.0 seconds and for the last 4.0 seconds they performed as they did in Section V.2.7.7. When the parameters varied, performance for the first few seconds was the same as that of Section V.2.7.6, as the true parameters had not had enough time to shift significantly away from the starting position of (0.87, 9.0) during acquisition. All of the methods lagged behind the true parameter position as it varied from (0.87, 9.0) to (0.93, 41.0) over the 8.0 second run. It was anticipated that parameter position and velocity estimate monitoring would out-perform the other methods when the true parameters varied, by anticipating the parameter movement; however, this method actually fared no better than any of the others.

V.2.7.9. Comparison Summary. Table 7-3 summarizes the various method's performances. The best overall performance of any of the methods was achieved by both parameter position estimate monitoring and by probability monitoring. In one case, parameter estimate monitoring attained a faster acquisition time (Section V.2.7.2) but in another case, probability monitoring was faster (Section V.2.7.4). Residual monitoring consistently took longer to acquire the true parameters, and in two cases it failed completely to identify the parameters. Parameter position and velocity estimate monitoring had no faster acquisition time than did parameter position estimate monitoring and exhibited the additional undesirable characteristic of "losing lock" once the parameters were identified. The covariance monitoring acquisition

TABLE V-3
Acquisition Times (sec)

Mode \ Parameter							
	1	2	3	4	5	6	7
1	1.00	1.06	0.09	5.25	--	1.38	--
2	0.56	1.06	0.09	2.65	3.63	1.38	3.63
3	0.85*	1.06	0.13*	0.88*	3.63	--	3.88
4	0.53*	1.06	0.47	1.25	3.63	1.38	3.63
5	1.06*	0.63	1.25	3.25	2.25	1.38	0.25
* = lost lock -- = failed to identify							
parameter				mode			
1 - (0.0, 10.48)				1 - residual monitoring			
2 - (0.1111, 48.65)				2 - parameter position estimate monitoring			
3 - (0.4444, 13.54)				3 - parameter position and velocity estimate monitoring			
4 - (0.8889, 2.12)				4 - probability monitoring			
5 - (1.0, 62.33)				5 - probability monitoring with covariance monitoring acquisition cycle			
6 - (0.07, 9.0)							
7 - (0.93, 41.0)							

cycle performed quite well in some instances (high true natural frequency) but failed in others (low true natural frequency).

For all of the various decision methods, filter warm-up was found to have little, if any, effect on performance. In view of this and the additional computation time and elemental filters needed, filter warm-up is not warranted.

Some of the similarities and differences in performance of the various decision methods can be explained through examination of the values that are compared against the thresholds. With residual monitoring, the likelihood quotients are used. These are a function of the single most recent measurement. Both parameter position estimate and probability monitoring, on the other hand, use functions of the recursive probability calculations, which are based on all past measurements. Parameter position and velocity estimate monitoring performs in a similar but degraded manner, degraded by the introduction of a pseudorate of the parameter movement into the calculations (in an attempt to predict future characteristics based on trends observed in the past).

V.3. Controller Performance

Three types of controllers are discussed in this section. Probability monitoring is the decision method used to move the sliding bank estimator (estimator/controller in the case of the sliding bank multiple model adaptive controller) because, as was discussed in the previous section, it has the best overall performance (along with parameter position estimate monitoring, but with the additional benefit of a slightly

less costly decision algorithm from a computational loading standpoint). In all cases, the control was disabled for the first second of the run. This was necessary to maintain stable operation of the system while the estimator attempted to identify the true system parameters. If the control is applied before the parameters have been identified (especially the ω parameter), the wrong control is applied, driving the system unstable. The choice of 1.0 second was made through an examination of parameter estimation errors; the ω parameter is always identified by 1.0 second.

The control cost weightings (necessary for Equation III-2) used were:

$$W_x = \begin{bmatrix} 10^5 & 0 \\ 0 & 10 \end{bmatrix} \quad W_u = 0.01$$

These values were picked to show a dramatic impact of the control on the position type variable, assuming that tight control of the position type variable is desired. The relative magnitudes of the $W_{x_{11}}$ and $W_{x_{22}}$ terms place a heavy cost on deviations of the position type variable from zero, while little cost is placed on deviations of the velocity type variable from zero. In other words, very tight control is desired on the position type variable, with less emphasis on the velocity type variable's deviations from zero. The very low value of the W_u cost weighting means that the amount of control energy expended

to regulate the states is of little concern.

Figures A-17a through A-17g are plots of the two states of the system using a single Kalman filter and LQ controller that have artificial knowledge of the parameters. These figures are the benchmark figures to which the various controllers are compared. (Again, due to computational resource limits, there are no benchmark figures for jump and varying parameters.)

V.3.1. Single Changeable-Gain Controller. This controller structure is described in Section III.3. Figures A-18a through A-18i are graphs of the system states, using the sliding bank estimator to provide estimates of the system states to a single changeable-gain controller, the gain changed by the sliding bank estimator's estimate of the true system parameters. In comparing figures A-17a through A-17g with Figures A-18a through A-18g, there is no discernible difference between the benchmark performance and that of the adaptive estimator/controller of this section, for any of the true parameter positions used.

Figure A-18h is a plot of the system states for the case of the true parameters undergoing a jump in value at 4.0 seconds. The control was disabled for 1.0 second after the jump occurred to enable the estimator to identify the new parameter values. If the control had been left on, the controller would have continued to apply the control based on the previous values of the parameters, resulting in unstable behavior. The decision to turn off the control was made by residual monitoring (Section II.4.1.1), with a threshold value of 25.0. (The value of

15.0 turned off the control at 2.5 seconds, before the parameters had even jumped while a value of 35.0 allowed the improper control to be applied for 5-6 sample periods after the parameters had jumped before turning off the control.) When the smallest of the likelihood quotients (Equation II-10) exceeded 25.0, the control was disabled for 1.0 second (the same time as used for the initial acquisition). An attempt was made to disable the control only while the threshold level was exceeded; however, erratic behavior resulted, suggesting that a different threshold or a different approach might be needed for deciding when to resume control. These questions are topics of further study.

Figure A-18i is a plot of the system states for the case of the true parameters varying. Here, the parameters varied slowly enough that the proper control could be applied throughout. Note that the two ends of the plots compare closely with those of the benchmark figures, Figures A-17f and A-17g.

V.3.2. Single Fixed-Gain Controller. This controller is a fixed-gain controller that is designed based upon a nominal value of the parameters. The controller is then supplied with the estimated system states from the sliding bank estimator. (Refer to Section III.4.) The nominal parameter value used for the controller was $(\zeta, \omega) = (0.4444, 62.83)$ corresponding to parameter space coordinates (5, 10). In determining the nominal parameter value, it was found that the controller had little sensitivity to variations in the true ζ parameter. There was a marked dependency of the controller

performance on the true ω parameter, however. When the true parameter point was in a higher frequency region in the parameter space than that of the nominal value used for the controller design, the true-system/estimator/controller combination became unstable. For example, when the nominal parameter point chosen was (5, 5), corresponding to (ζ , ω) = (0.4444, 17.48), a true frequency of 18.61 rad/sec and above resulted in unstable performance. When the true parameter point was at a lower frequency point than the nominal, the system was stable but performance was degraded over that of the benchmark. In order to maintain stability for all true parameter values, the highest possible natural frequency value was picked for the nominal ω parameter. Because of the insensitivity to changes in the damping ratio parameter, a mid range value was picked for the ζ parameter.

Figures A-19a through A-19i are plots of the state variables for the single fixed-structure controller. In Figure A-19a, the true frequency is much lower than the nominal frequency and the performance is severely degraded. Compare Figure A-17a with A-19a; the variance of the position type variable is roughly 15 times greater in Figure A-19a than in Figure A-17a. In Figure A-19b the true frequency is much closer to the nominal frequency and as a result the variance of the position type variable in Figure A-19b is only $1 \frac{1}{3}$ times as great as in Figure A-17b. In all of the plots where the true frequency is high, performance is only slightly degraded (variances on the order of 1.5 times greater than the

benchmark). Compare Figures A-19e and A-19f with Figures A-17e and A-17f. In the other figures where the true parameter point is at the low frequency end (Figures A-19c and A-19d), the degradation in performance is greater (variances on the order of 6 to 10 times greater than the benchmark). Compare Figures A-19c and A-19d with A-17c and A-17d.

Figure A-19h is a plot of the state variables for the jump parameter case. No instability problems were encountered when the parameters jumped, as expected, since the nominal parameter value was chosen such that stable operation would be achieved for any true parameter position. Performance during the first 4.0 seconds is comparable to that of Figure A-19f and performance during the last 4.0 seconds is comparable to that of Figure A-19g.

In Figure A-19i, the parameters are slowly varying. As the frequency is increased, the performance improves, as expected.

V.3.3. Sliding Bank Multiple Model Adaptive Controller.

The controller of this section is made up of nine elemental LQG controllers in which each elemental LQ controller is fed state estimates by its corresponding elemental filter. The elemental controller outputs are then put through the weighted average. (See Section III.5.) The performance of the sliding bank multiple model adaptive controller is identical to that of the single changeable-gain controller and the benchmark controllers, for all values of the true parameters examined. See Figures A-20a through A-20i. The same insta-

bility problems encountered by the single changeable-gain controller for the jump parameter case were also encountered here, and the remedy of a 1.0 second reacquisition cycle, initiated via residual monitoring, allowed stable performance to be achieved.

V.3.4. Comparisons. Of the three types of controllers examined, two were identical in performance. The single changeable-gain controller and the sliding bank multiple model adaptive controller both performed exceptionally well, comparing essentially identically with the benchmark controllers (a single Kalman filter feeding a single LQ controller, both having artificial knowledge of the true parameters). The single fixed-gain controller performed adequately when the true frequency was high (with variances on the order of 1.5 times greater than those of the benchmark) but did not perform as well when the true frequency was low (with variances as high as 15 times greater than those of the benchmark).

V.4. Computational Loading

Multiple model adaptive estimation/control is ideally suited for a parallel processing implementation. As a result, discussion of computational loading is somewhat difficult. Nevertheless, an attempt is made in this section to evaluate computational loading.

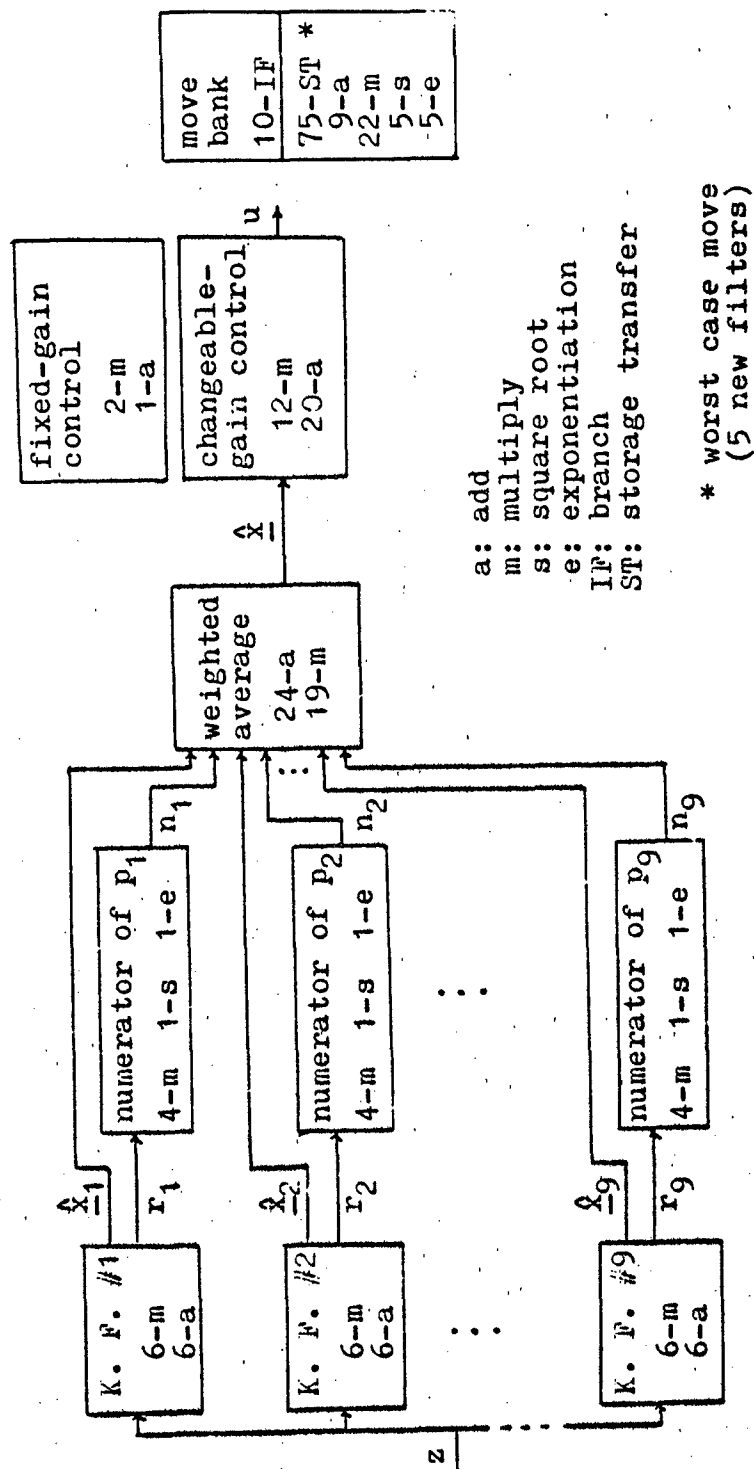
It is assumed that the elemental filters (filter/controllers for the sliding bank multiple model adaptive controller) would be processed in a parallel fashion (nine Kalman-

filter-on-a-chip's, for example). The propagations and updates of the state estimates of the elemental estimators and the numerators of the recursive probability weightings (Equation II-9) could all be accomplished in a parallel fashion. For the simple two-state system of this study, assuming steady-state-gain filters, a total of six multiplies and six adds would be needed to propagate an elemental state estimate from one sample time to the next and to update it with the latest measurement. To calculate the numerators of the probability weightings, four multiplies, one square root, and one exponentiation would be required.

The computations that need to be carried out in a serial manner are the denominator of the recursive probability weightings and the weighted averaging of the elemental state estimates. These two calculations combined require a total of 19 multiplies and 24 adds.

If the feedback control calculations are also to be conducted in serial (as for the single changeable-gain controller of Section III.3), a total of 12 multiplies and 20 adds would be needed (assuming the interpolation scheme of Section III.3 is used to compute the steady-state controller gains). If the fixed-gain controller is used, only two multiplies and one add are needed (again, steady-state gains are assumed). See Figure 7-4.

If the feedback control is to be calculated via the sliding bank multiple model adaptive control method (Section III.5), it is assumed that the elemental control calculations would



a: add
m: multiply
s: square root
e: exponentiation
IF: branch
ST: storage transfer

* worst case move
(5 new filters)

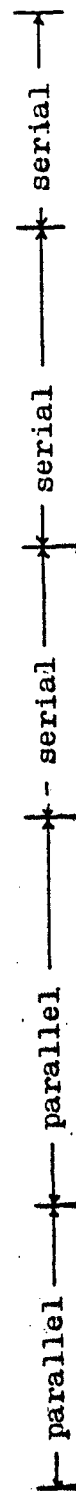


Figure V-4. Single-Gain Controller Computational Burden

also be calculated in parallel. This would mean that an additional two multiplies and one add would be calculated in parallel. The weighted averaging computation requires fewer operations with the sliding bank multiple model adaptive control method, since only a single scalar control variable needs to be computed instead of the two state variables necessary for the single-gain (changeable or fixed) controller methods. The weighted averaging, with this approach, requires 16 adds and 10 multiplies. See Figure V-5.

The only serial processing item left is the move-the-bank decision logic. Assuming probability monitoring is used to move the bank, 10 "if's" would need to be processed in order to check the probability weightings against the threshold and to decide whether or not to move. If the decision is made to move, the worst case would be if five filters were changed to new locations in the parameter space. The reinitialization of five different filters would require 75 storage transfers (to change the filters to their new values of Φ_k , B_{d_k} , $\bar{K}_k$, $P_{11_k} + R$, $\hat{x}_k$, and p_k). In addition to the storage transfers, 22 multiplies, 9 adds, 5 square roots, and 5 exponentiations must be accomplished to calculate the p_k 's for the new filters (Equation II-11). If the multiple model adaptive control is used, an additional 10 storage transfers would be needed for the G_k 's.

Table V-4 presents a summary of the computation times (keeping in mind that these times are for comparisons between methods only) of the various control schemes, based on representative operation times of 1μ second for a branch, 1μ sec-

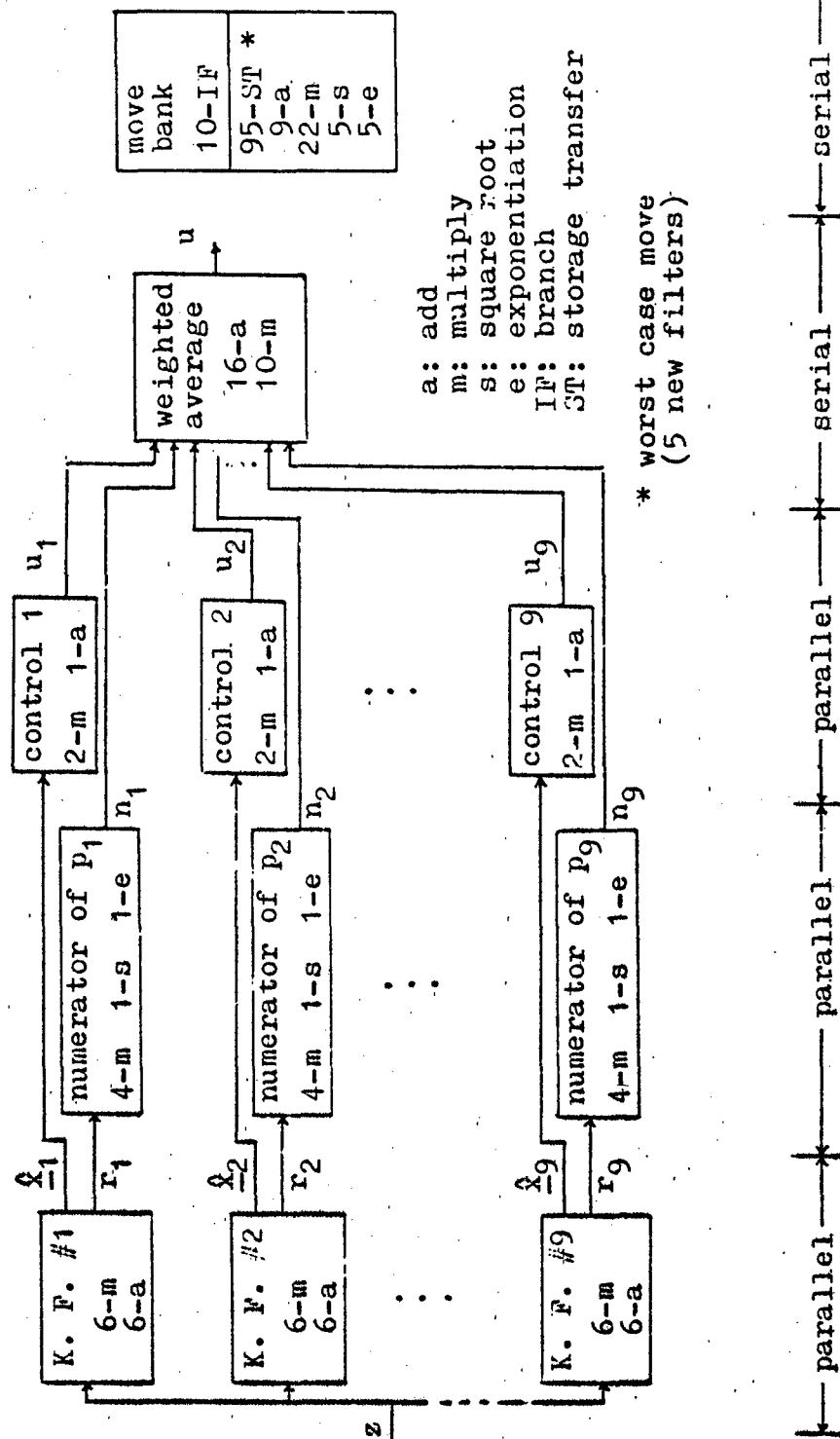


Figure V-5. Sliding Bank Multiple Model Adaptive Controller Computational Burden

TABLE V-4

Computation times (milliseconds)

	K. F.	Num.	Control	W. Av.	Move	Total
single changeable-gain controller *	0.054	0.144	0.132	0.186	0.844	1.360
	0.486	1.296	0.132	0.186	0.844	2.944
single fixed-gain controller *	0.054	0.144	0.018	0.186	0.844	1.246
	0.486	1.296	0.018	0.186	0.844	2.830
sliding bank multiple model adaptive controller *	0.054	0.144	0.018	0.108	0.864	1.188
	0.486	1.296	0.018	0.108	0.864	2.772
full bank multiple model adaptive controller *	0.054	0.144	0.018	1.200	0	1.416
	5.400	14.400	0.018	1.200	0	21.018

* all computations conducted serially

ond for a storage transfer, 3μ seconds for an add, 6μ seconds for a multiply, and 60μ seconds for either square roots or exponentiations. (Recall that the numbers of individual operations are specified in Figures V-4 and V-5, if it is desired to tailor this comparison to other machine capabilities.) For comparison, the full bank approach (100 filter/controllers) is presented. The full bank requires 198 adds and 101 multiplies to perform the weighted averaging but does not require any time for a move-the-bank decision.

The least amount of processing time is required by the sliding bank multiple model adaptive control scheme (1.188 milliseconds). Next is the single fixed-gain controller with 1.246 milliseconds required processing time. Finally, the single gangeable-gain controller needs 1.36 milliseconds for execution. All three methods require less time than the full bank multiple model adaptive controller, which requires 1.416 milliseconds. If all computations are to be conducted serially, the three sliding bank multiple model adaptive estimator/controller schemes require very nearly the same amount of computation time (2.9, 2.8, and 2.7 milliseconds). The full bank multiple model adaptive controller, on the other hand, requires an order of magnitude increase in computation time (21 milliseconds) over the sliding bank schemes.

V.5. Summary

The performances of the various move decision schemes for a sliding bank multiple model adaptive estimator and the

characteristics of three controller schemes based on sliding bank multiple model adaptive estimation were presented in this chapter.

The best overall move decision schemes were parameter position estimate monitoring and probability monitoring. A separate acquisition cycle, using parameter covariance monitoring, offered a performance improvement in some cases, but degraded performance in others. Letting a fixed size bank "walk" to the true parameter point worked well for all true parameter conditions. Filter warm-up before moving did not prove fruitful; in many cases there was no detectable improvement in performance when filter warm-up was used. In all of the move decision schemes, parameter acquisition was enhanced when either a dither signal was applied in addition to a true system driving noise signal or when feedback control was applied.

Probability monitoring was used in conjunction with each of the three controller schemes. Both the single changeable-gain controller and the sliding bank multiple model adaptive controller performed exceptionally well, comparing essentially identically with the benchmark of a single Kalman Filter/ χ^2 controller that had artificial knowledge of the true system parameters. The single fixed-gain controller, based on nominal parameter values of a mid-value ζ and the closest discrete value of ω under consideration (in order to provide stability for all true parameter values), performed well when the true parameters were near the nominal values chosen for the con-

troller, but exhibited degraded performance when the true parameter point was far from the nominal point.

The three controller schemes require roughly the same amount of computation time. The single changeable-gain controller requires 1.36 milliseconds, the single fixed-gain controller requires 1.246 milliseconds, and the sliding bank multiple model adaptive controller requires 1.188 milliseconds. All three require less than the full bank multiple model adaptive controller, which requires 1.416 milliseconds. If all computations are performed serially, the full bank multiple model adaptive controller requires 21 milliseconds while the three sliding bank controllers only require 2.9, 2.8, or 2.7 milliseconds.

VI. Conclusions

VI.1. Introduction

The feasibility of a moving bank multiple model adaptive estimator/controller has been examined in this thesis. Sliding 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 sliding bank estimator (9 elemental filters vs. 100 for the system modeled in this thesis). Reduction of the number of elemental filters is accomplished through a dynamic re-declaration of the positions in parameter space that the sliding bank's elemental filters occupy; i. e., the smaller number of filters in the sliding bank are moved about the parameter space in search of the true parameter point. Critical to the performance of the sliding bank estimator is the decision method that governs movement of the bank of elemental filters. Because of this, a number of different decision algorithms have been discussed and their respective performance compared.

Three controller designs have also been examined in this thesis. The first design that has been explored uses a single controller fed by state estimates, with control gains that are calculated based on the current parameter position estimate, which is provided by the sliding bank estimator. A second design that has been examined consists of state estimates being premultiplied by a single fixed-gain con-

troller that is based upon a single nominal parameter value, chosen so as to provide robust, adequate control for all true parameter values. Sliding bank multiple model adaptive control has been examined as well; an elemental controller is cascaded with each of the elemental filters of the sliding bank, computing feedback control from a weighted average of the elemental controller outputs.

States of a damped second order system, with uncertain parameters (damping ratio ζ and undamped natural frequency ω) have been estimated by the sliding bank estimator and then regulated to the quiescent state by the controller. Performance of the sliding bank estimator/controller has been compared to a benchmark of a single Kalman filter/LQ controller that has (artificial) knowledge of the true parameter values. Comparisons have been based upon Monte Carlo analysis results.

VI.2. Sliding Bank Multiple Model Adaptive Estimator

Four move-the-bank decision methods have been examined:

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

Also examined have been a separate acquisition cycle in which the spacing of parameters to define elemental filters is allowed to vary, using parameter covariance monitoring, and the use of "filter warm-up" before moving the bank.

The best performing move-the-bank decision methods are

parameter position estimate monitoring and probability monitoring. The state estimate error plots of the sliding bank estimator using these decision methods compare essentially identically with those of the benchmark, once parameter acquisition has occurred (acquisition times range from 0.47 to 3.63 seconds depending upon the values of the true parameters). Parameter position estimate monitoring and probability monitoring are quite similar algorithms; both are based on the recursive probability weightings inherent in a multiple model adaptive algorithm. Probability monitoring is less demanding computationally, however.

The covariance monitoring acquisition cycle improves performance when the true system natural frequency is high, but degrades performance when the true natural frequency is low. "Warming filters up" before actual use within the sliding bank does not significantly affect performance. The additional computation time and complexity required to warm up filters before a move are not warranted.

VI.3. Controller

The best-performing move decision method, probability monitoring, has been used for the sliding bank estimator in conjunction with each of the controller schemes. Of the three controller schemes, two perform in an essentially identical manner. There is no detectable difference between state variable plots of the single changeable-gain controller and the sliding bank multiple model adaptive controller, and

both are indistinguishable from state variable plots of the benchmark. The single fixed-gain controller performs well when the true parameters are near the nominal parameter point chosen for the controller, but it performs poorly when the true parameters lie far from the nominal parameter point. Additionally, when the true natural frequency is greater than the nominal natural frequency, the sliding bank estimator/fixed-gain controller drives the system unstable; therefore a nominal natural frequency corresponding to the highest frequency in the parameter space is used for the fixed-gain controller. The performance degradation is not warranted by the reduced computational loading.

VI.4. Areas of Further Study

There are a number of issues concerning the sliding bank multiple model adaptive estimator/controller that warrant further exploration. One area that deserves further attention is performance sensitivity to poor initial conditions and to driving and measurement noise strengths. Both sensitivity to Q and R , with the filter knowledgeable of correct Q and R , and robustness to incorrect Q and R values should be explored.

Another area worth investigation is the appropriate method of enabling control. For the controller designs examined in this thesis, the feedback control is disabled for the first second of operation to allow the sliding bank estimator to identify the system parameters (at least the ω parameter); otherwise unstable operation results. In light of the fact

that the system becomes unstable when the true frequency is greater than the estimated frequency used to compute the controller gains, nominal control gains based on the highest possible frequency might be used during parameter acquisition so that at least some control is applied. Additionally, an appropriate method of deciding when to enable control can be investigated. Rather than just disabling control or using gains based on the highest ω for one second (or some other fixed interval of time) to allow for parameter acquisition, a method of deciding when parameter acquisition has occurred (perhaps based on residual monitoring) might be developed so that a more rapid application of appropriate control can be obtained.

The covariance monitoring acquisition cycle offers a substantial performance improvement when the true natural frequency is high but degrades performance when the true natural frequency is low. Analysis of other acquisition schemes (perhaps as simple as comparing a weighted norm of the parameter covariance matrix to an appropriate threshold level) may yield a method that improves performance for all true parameter values.

In light of the fact that filter warm-up does not improve performance, other methods of contraction than "best quadrant" should also be explored. This method was selected on the basis that it would retain four previously defined elemental filters through a contraction cycle, rather than requiring nine totally new elemental filters to be initiated.

Since the higher number of newly initiated filters does not seem to degrade performance significantly, it may be better to center the newly-declared, contracted bank on the location closest to the parameter estimate $\hat{a}(t_1)$. This and perhaps other contraction logics warrant further study.

Different parameter space discretizations is another area deserving of consideration. A finer parameter space discretization may improve performance when the true parameters are between discretized values. Sliding bank multiple model adaptive estimator/controller performance might also be investigated for different true systems (higher order) with different uncertain parameters.

Performance of MAP (maximum a posteriori; i. e., one that uses the "best" elemental filter's state or parameter estimate as the overall state or parameter estimate) vs. the Bayesian weighted average of filters should also be explored. This is tied closely to appropriate discretization of the parameter space; MAP methods can be expected to respond more quickly to true changes in the true parameter values, but suffer more from coarse discretizations.

VI.5. Summary

This thesis demonstrates that the sliding bank multiple model adaptive estimator/controller is a possible alternative to the conventional full bank multiple model adaptive estimator/controller. For the two state system of this study, the sliding bank estimator/controller offers performance that is

essentially identical to that of a single Kalman filter/LQ controller that has (artificial) knowledge of the true parameters, once parameter acquisition has occurred. In addition, only 9 elemental filters are needed for the sliding bank estimator vs. 100 for the full bank multiple model adaptive estimator based on 10 discrete values for each of two uncertain parameters. The advantages of the sliding bank concept become even more striking as the discretization becomes finer or the number of uncertain parameters increases. Research should continue in order to develop this practical adaptive algorithm more fully.

Appendix A

Simulation Plots

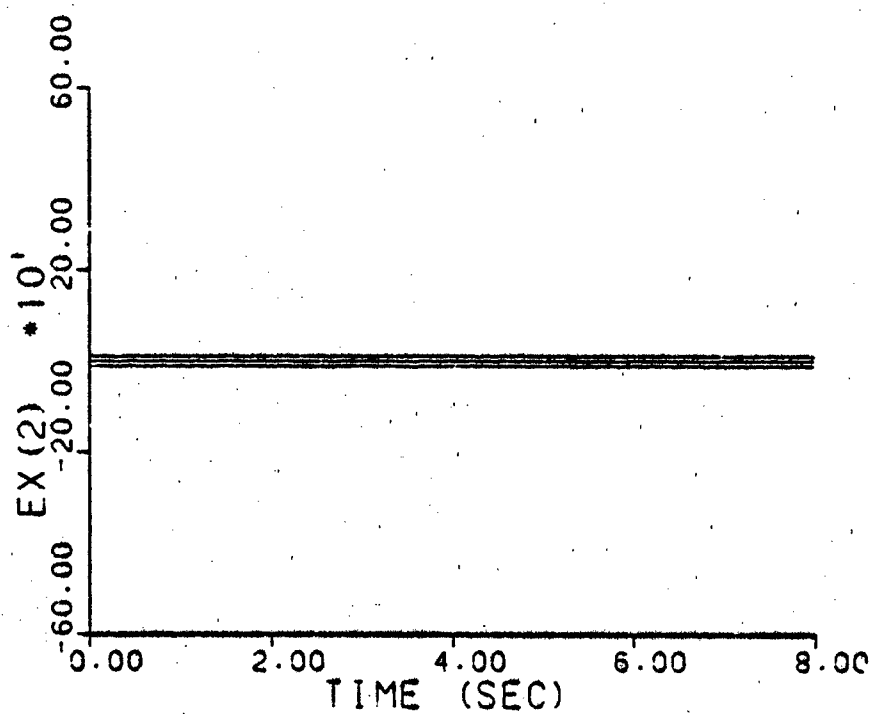
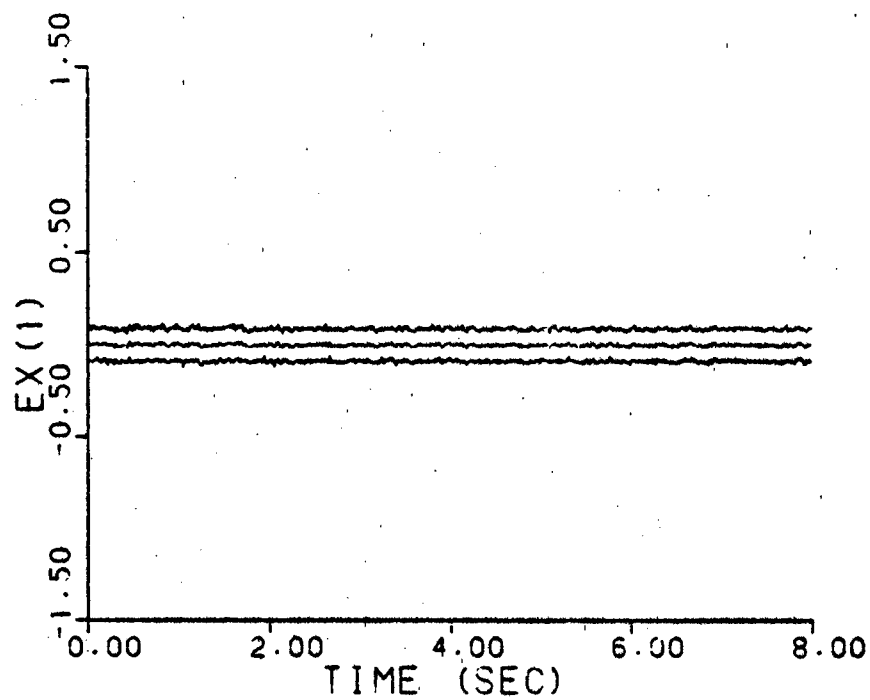


Figure A-1a. Benchmark: $\underline{a} = (1, 3)$

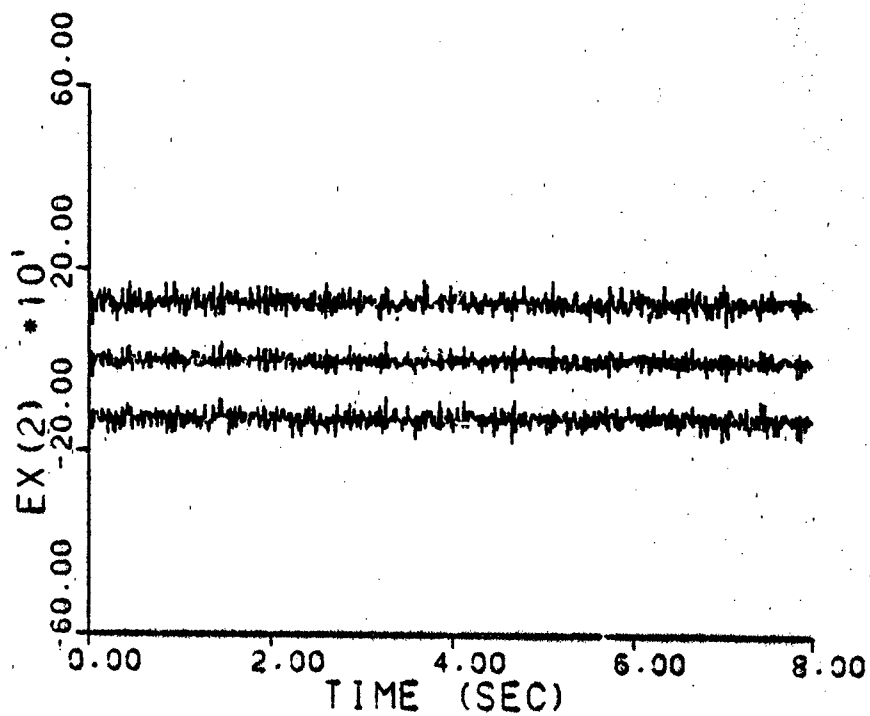
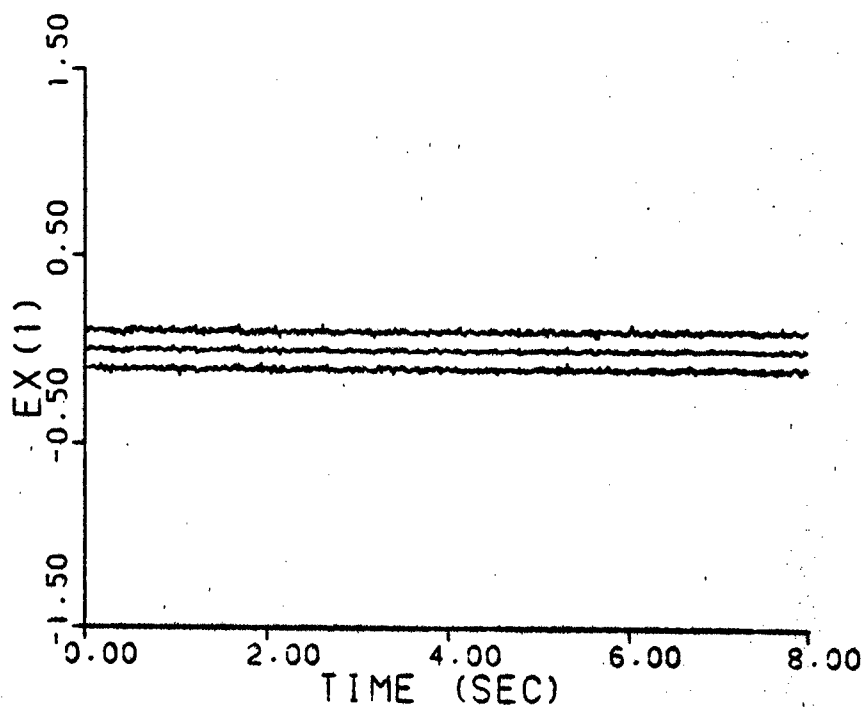


Figure A-1b. Benchmark: $\underline{a} = (2, 9)$

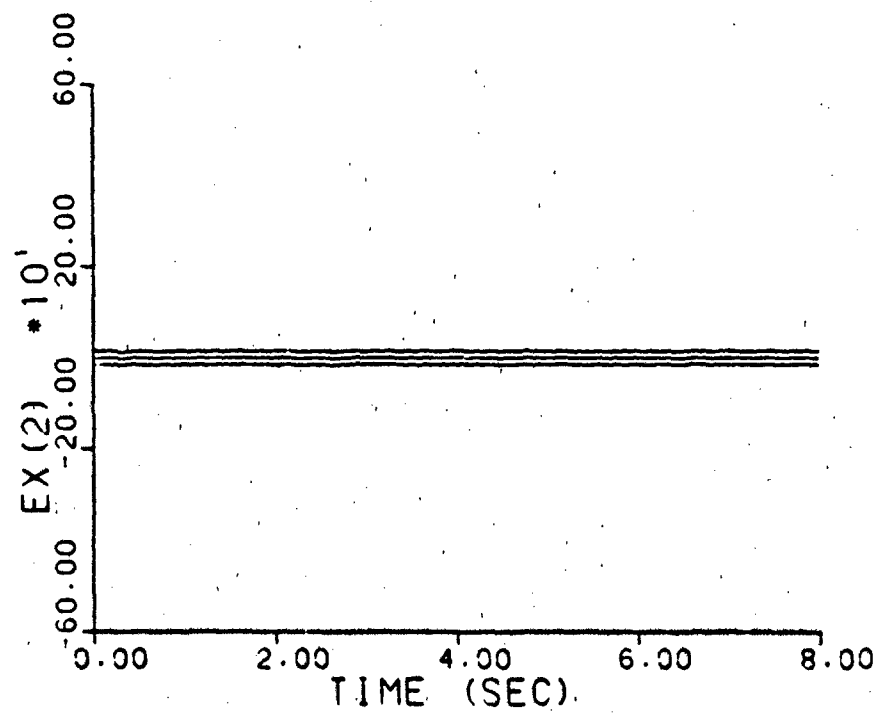
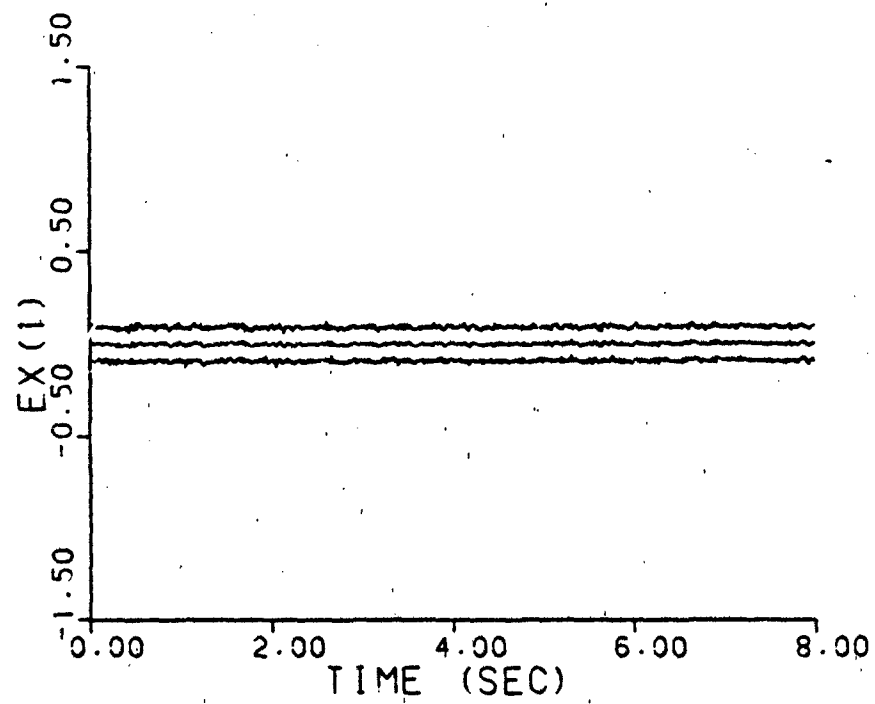


Figure A-1c. Benchmark: $\underline{a} = (5, 4)$

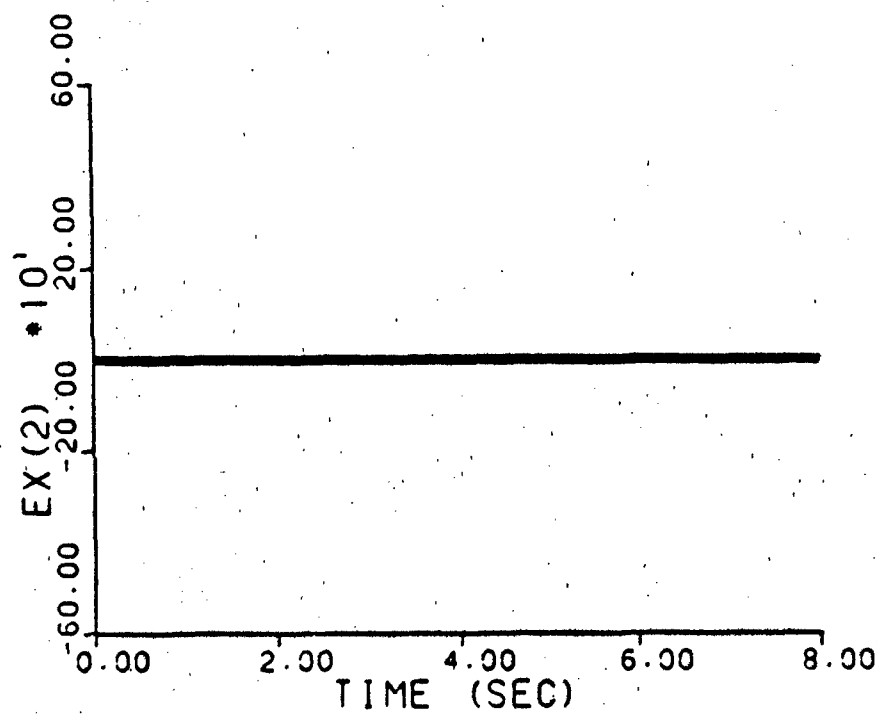
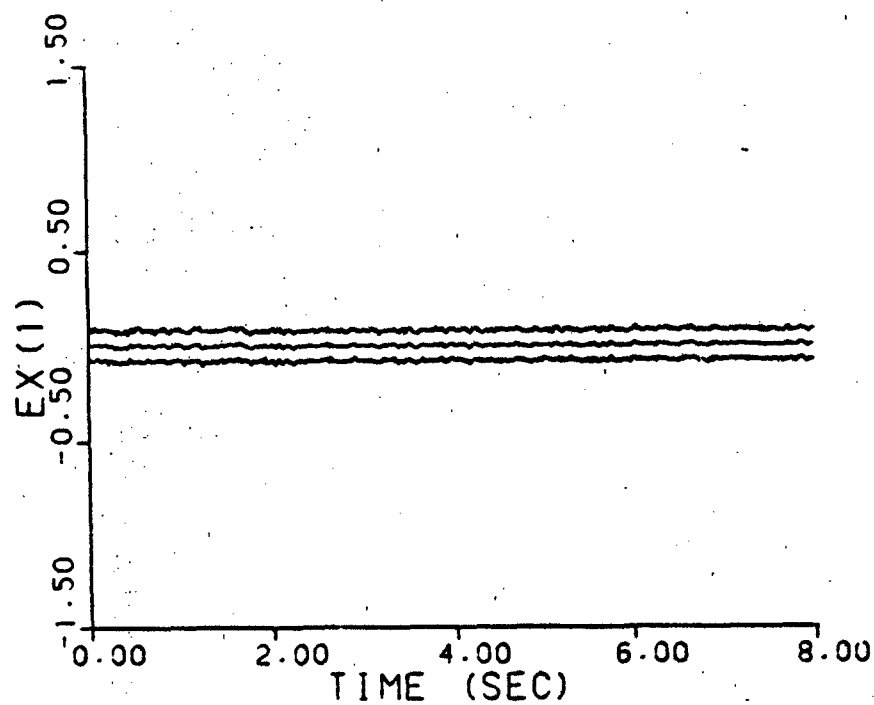


Figure A-1d. Benchmark: $\underline{a} = (9, 2)$

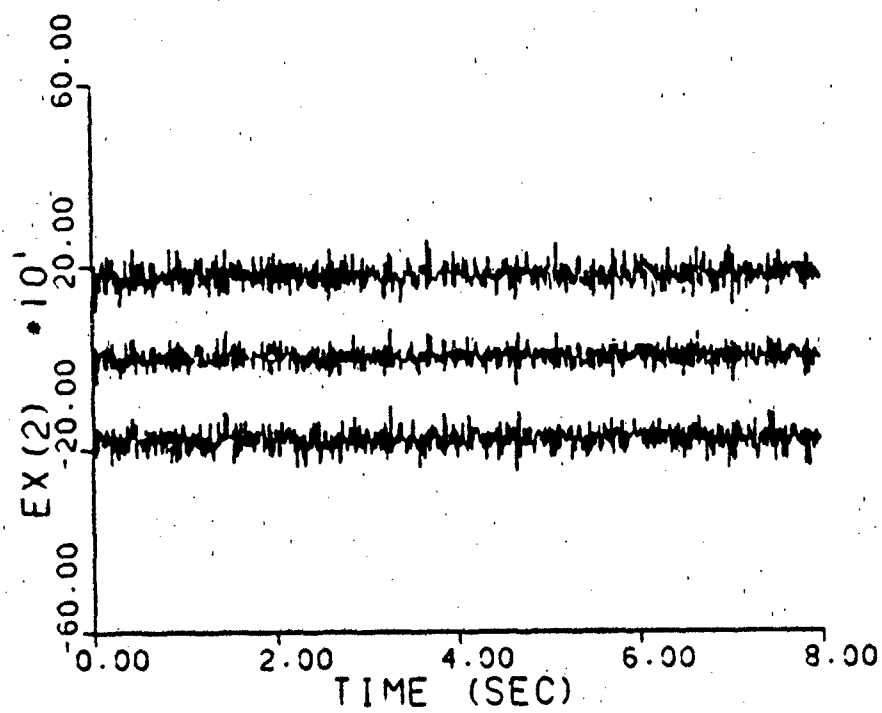
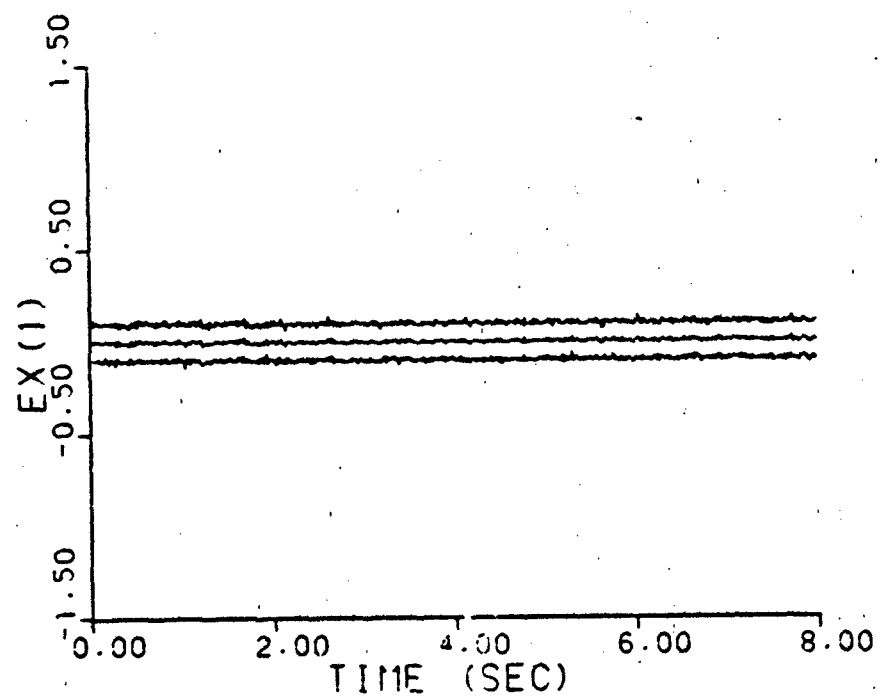


Figure A-1e. Benchmark: $\underline{a} = (10, 10)$

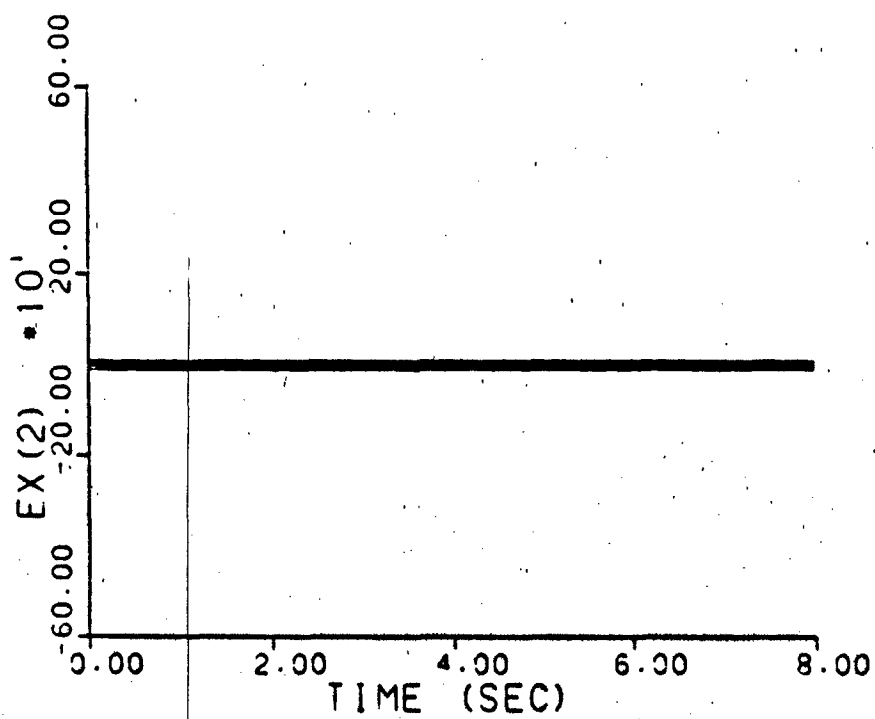
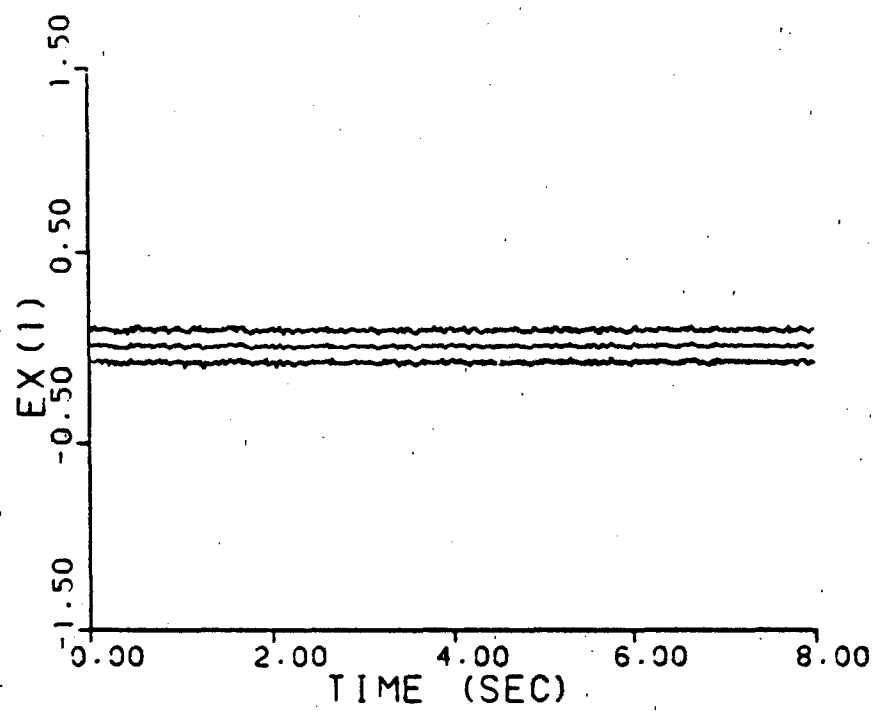


Figure A-1f. Benchmark: $\underline{a} = (0.07, 9.0)$

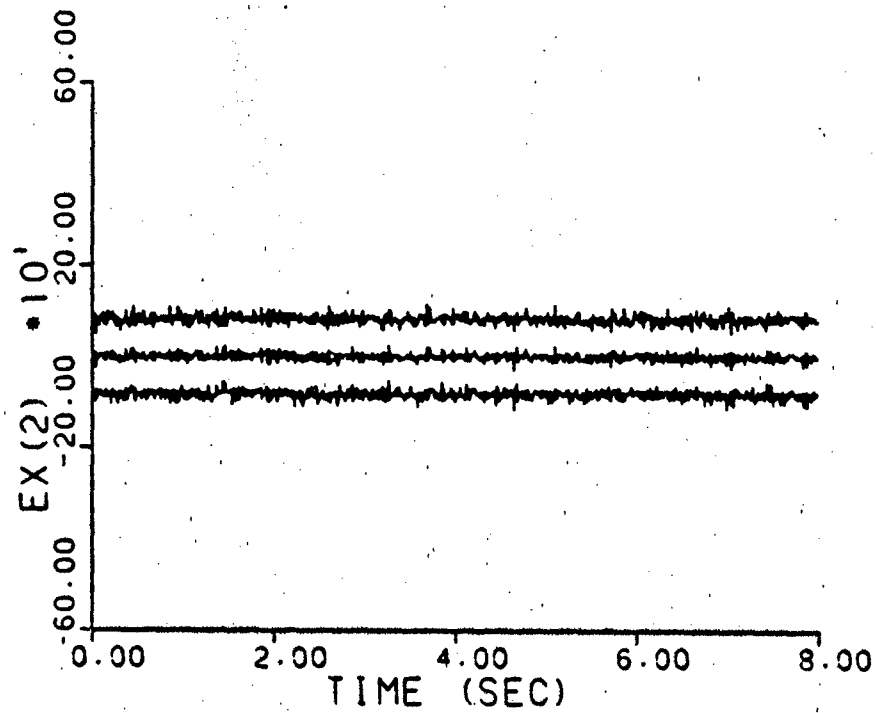
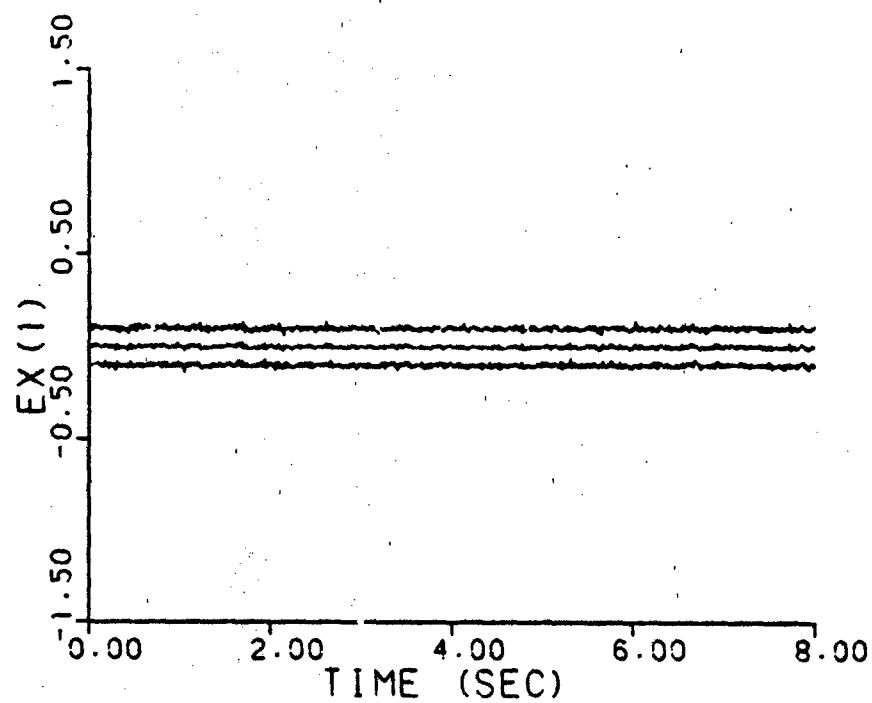


Figure A-1g. Benchmark: $\underline{a} = (0.93, 41.0)$

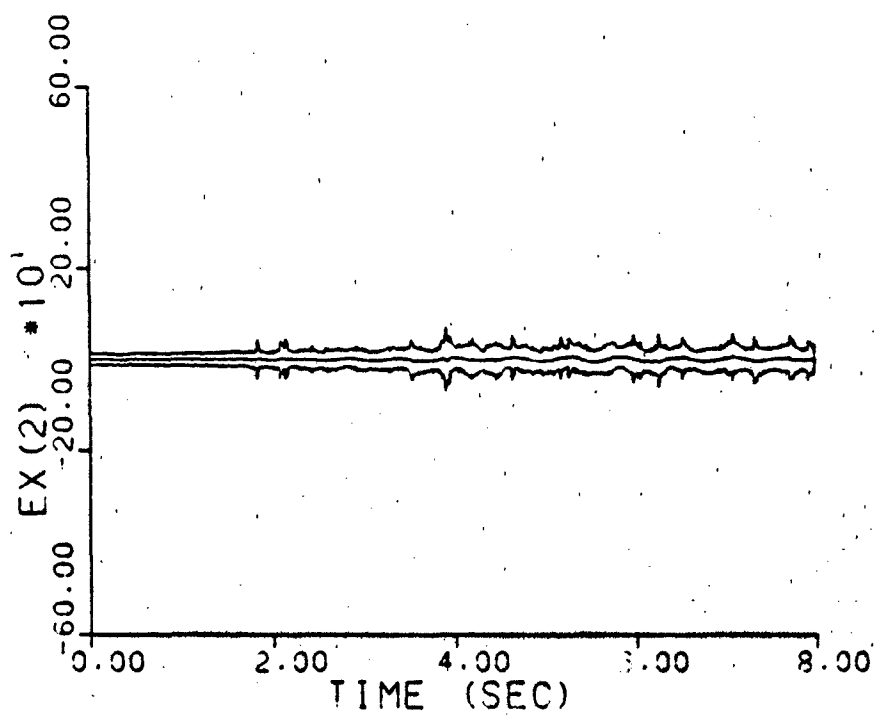
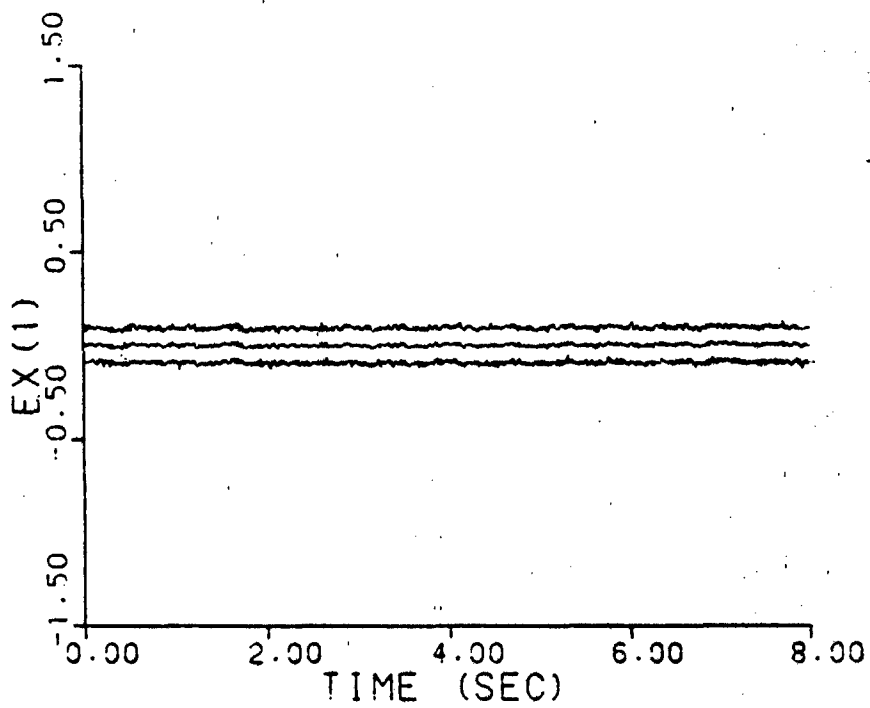


Figure A-2a. Residual Monitoring
 $\underline{a} = (1, 3)$, no dither

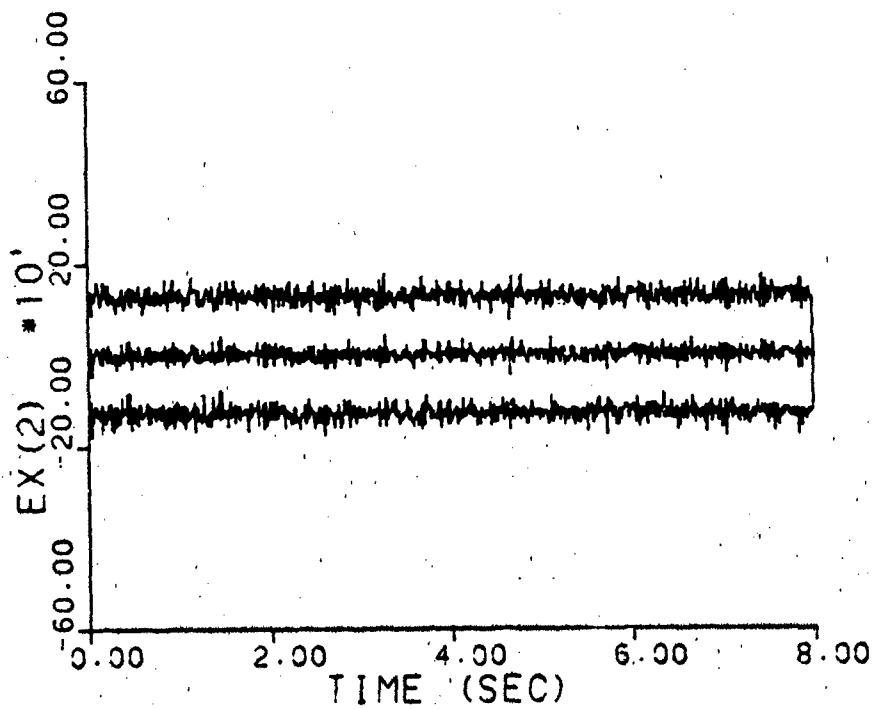
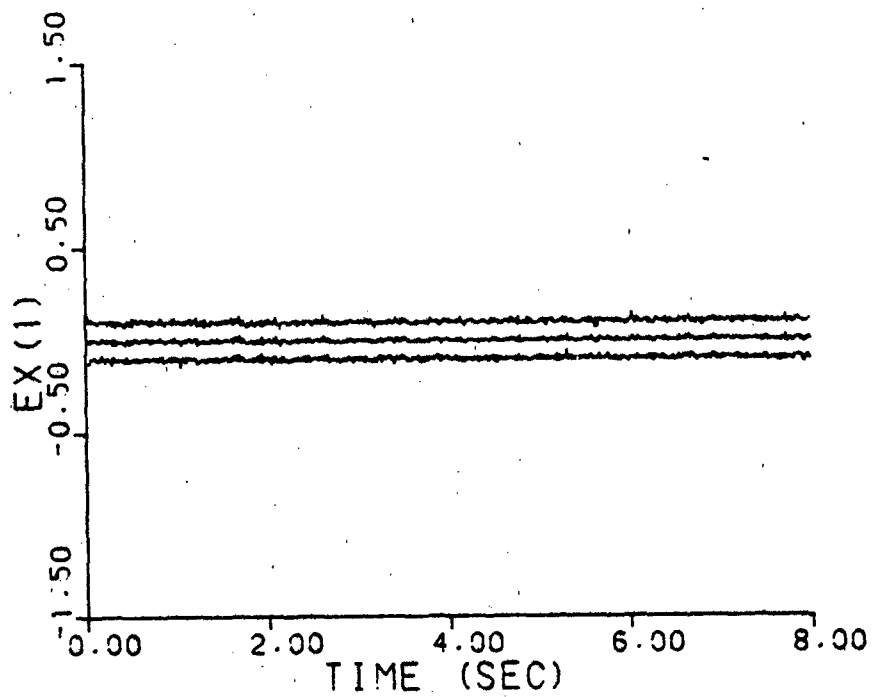


Figure A-2b. Residual Monitoring
 $\underline{a} = (2, 9)$, no dither

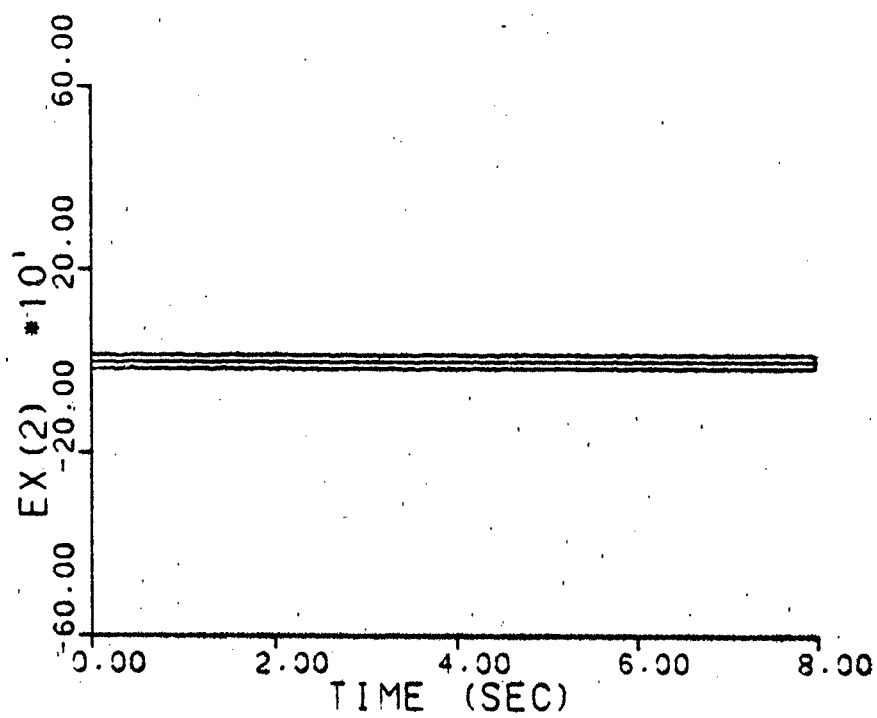
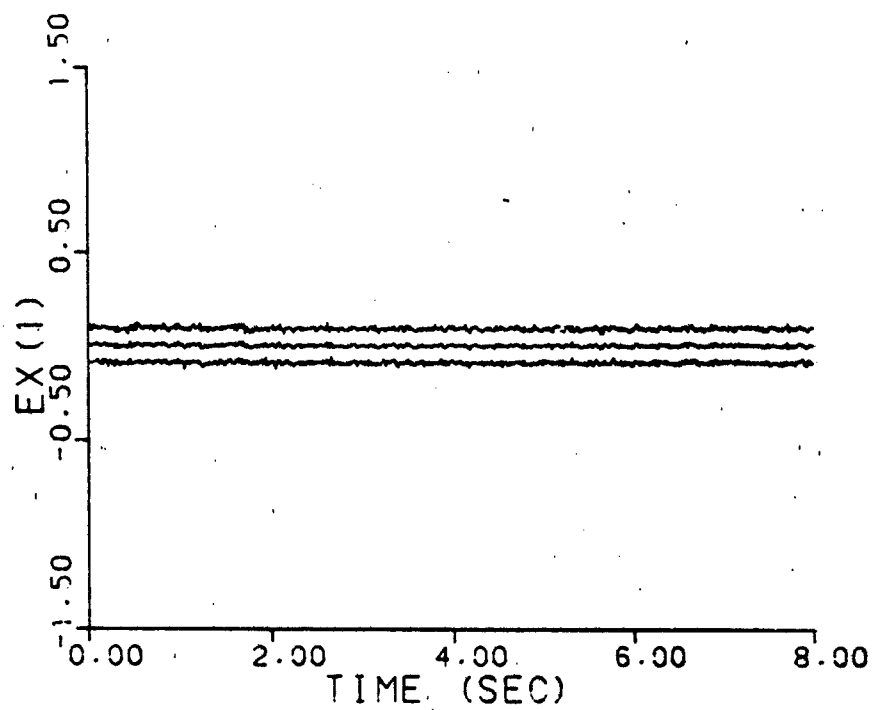


Figure A-2c. Residual Monitoring
 $\underline{a} = (5, 4)$, no dither

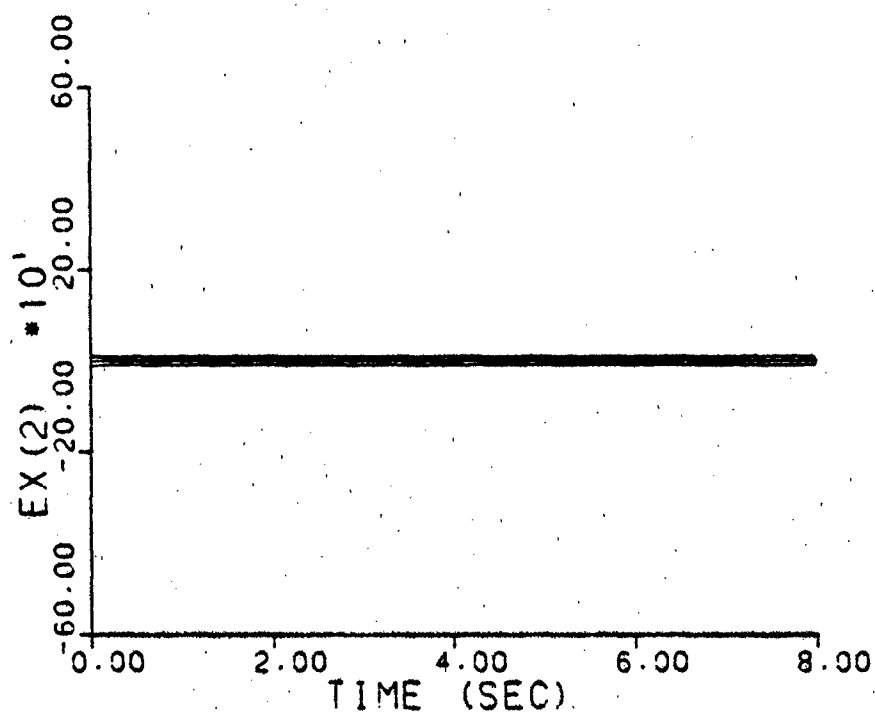
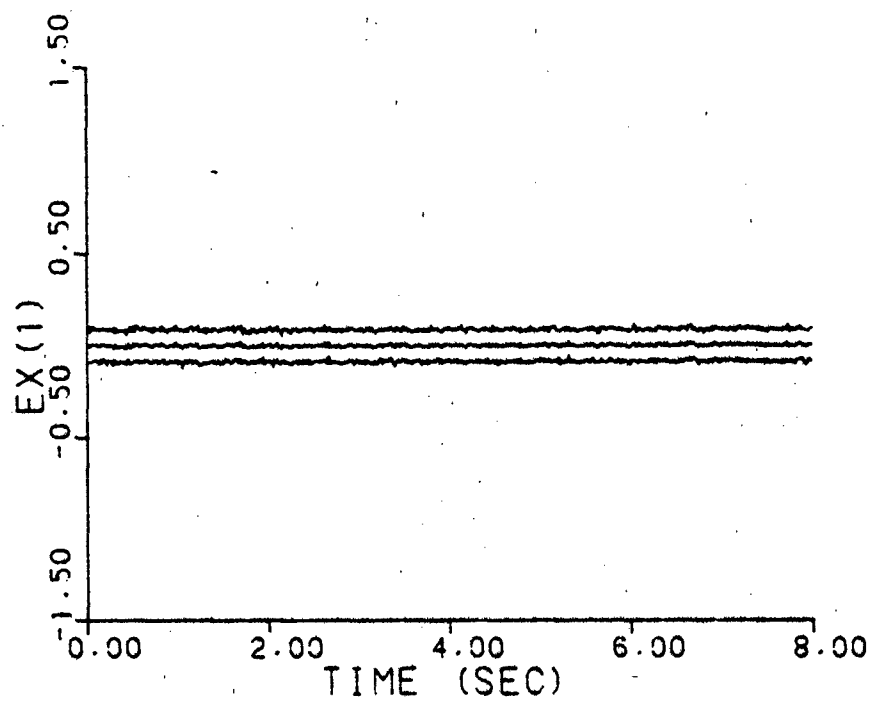


Figure A-2d. Residual Monitoring
a = (9, 2), no dither

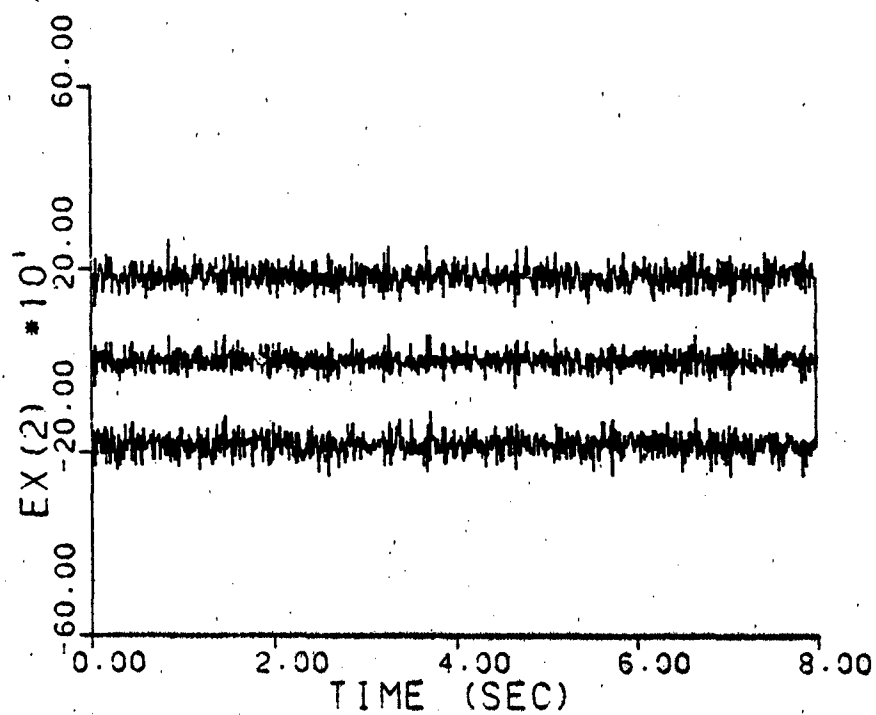
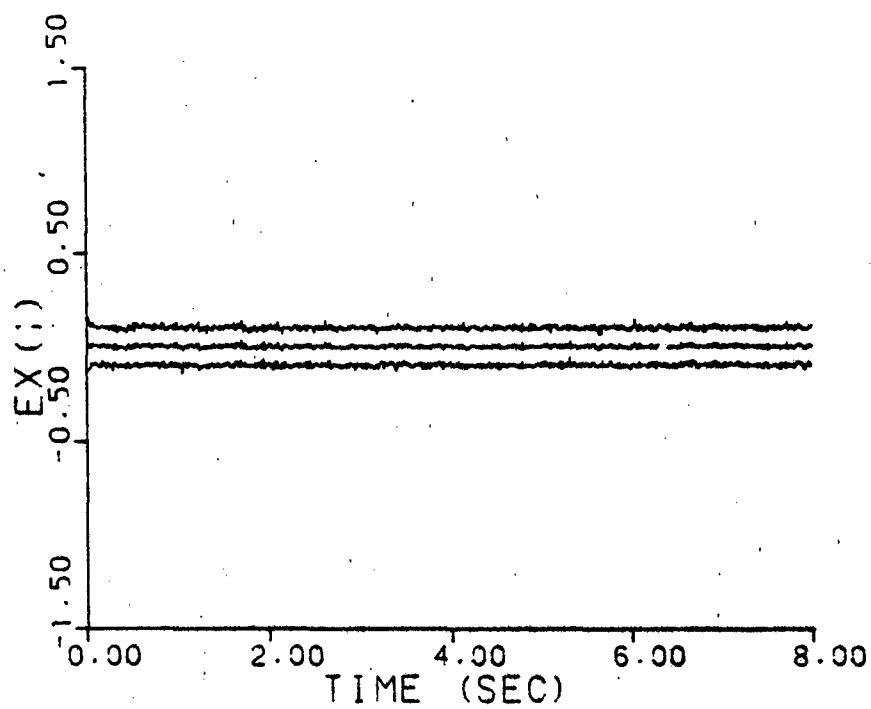


Figure A-2e. Residual Monitoring
 $\underline{a} = (10, 10)$, no dither

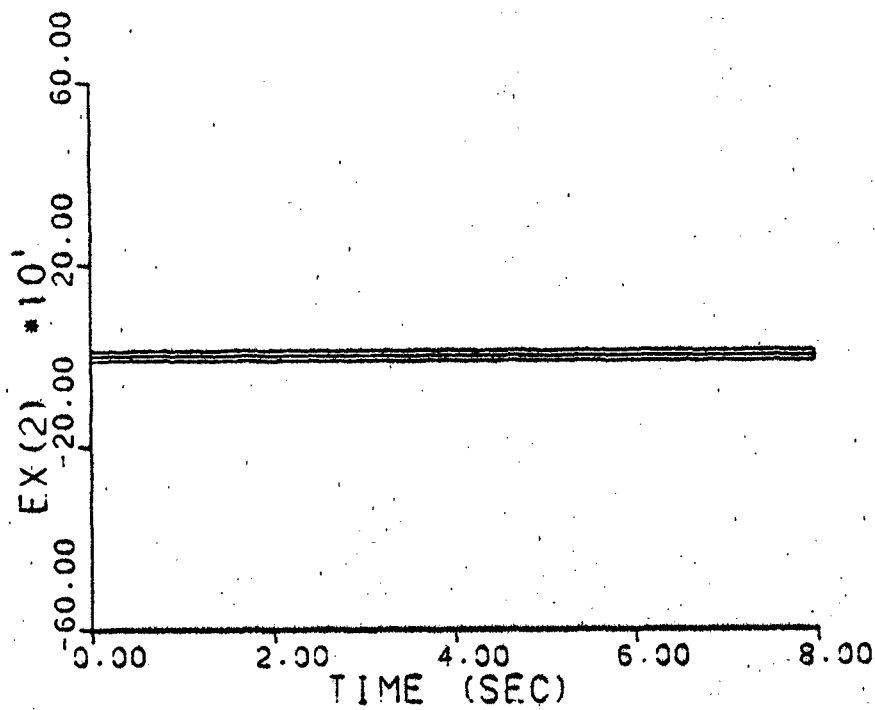
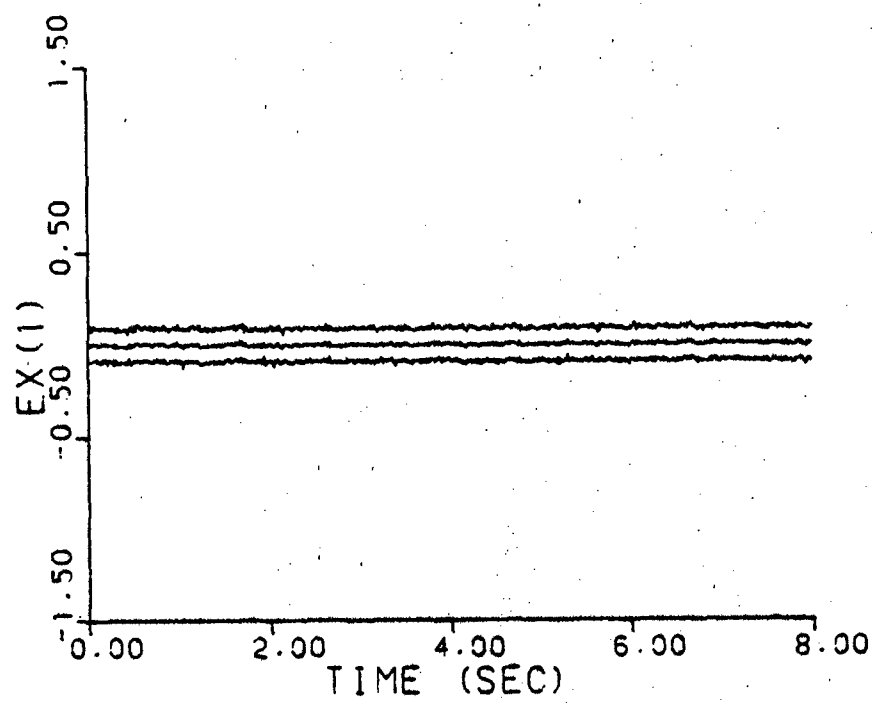


Figure A-2f. Residual Monitoring
 $\underline{a} = (0.07, 9.0)$, no dither

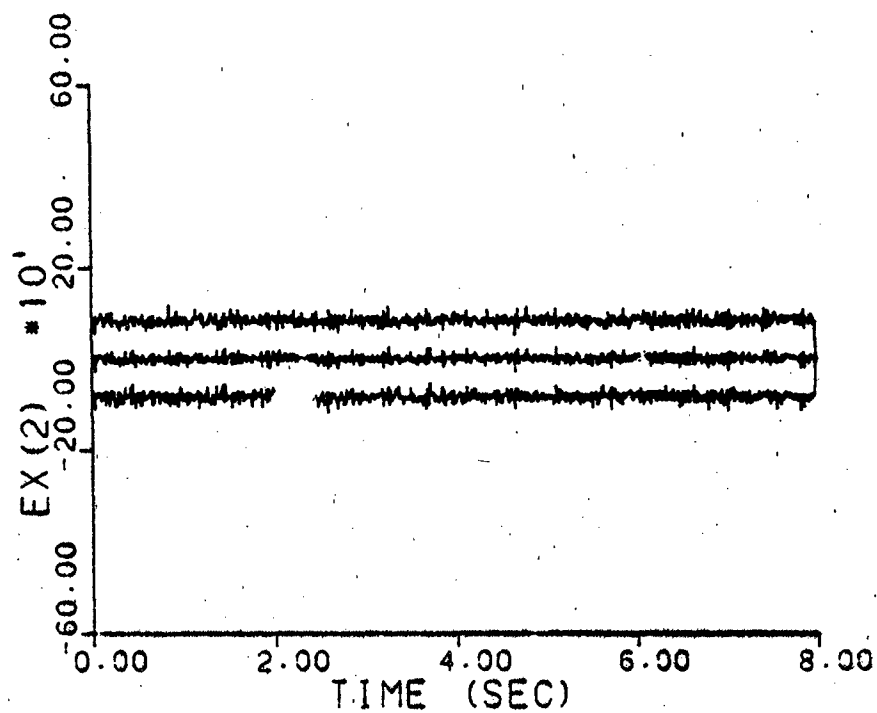
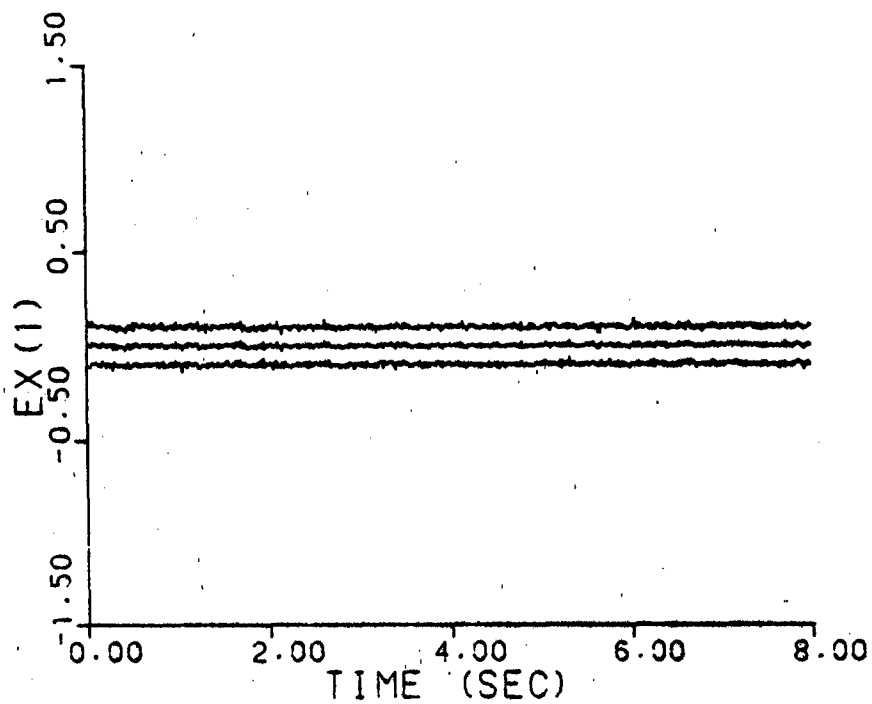


Figure A-2g. Residual Monitoring
 $\underline{a} = (0.93, 41.0)$, no dither

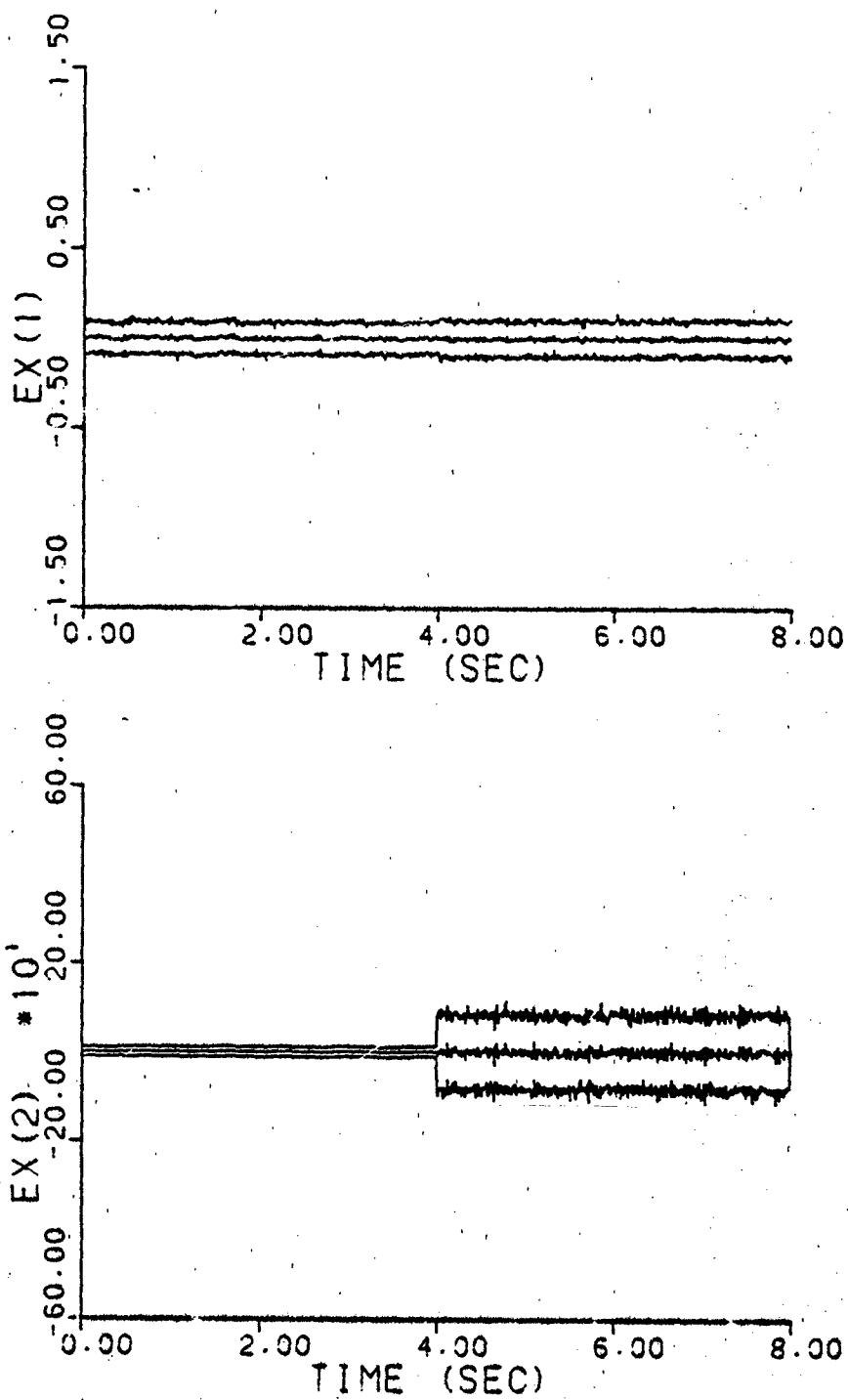


Figure A-2h. Residual Monitoring
a jumps: (0.07, 9.0) - (0.93, 41.0)
no dither

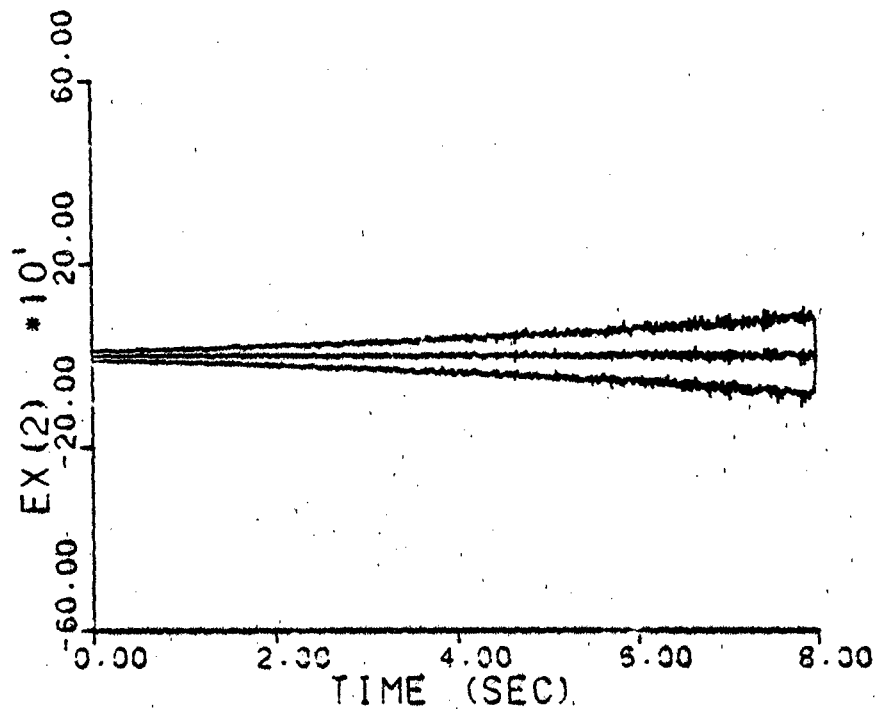
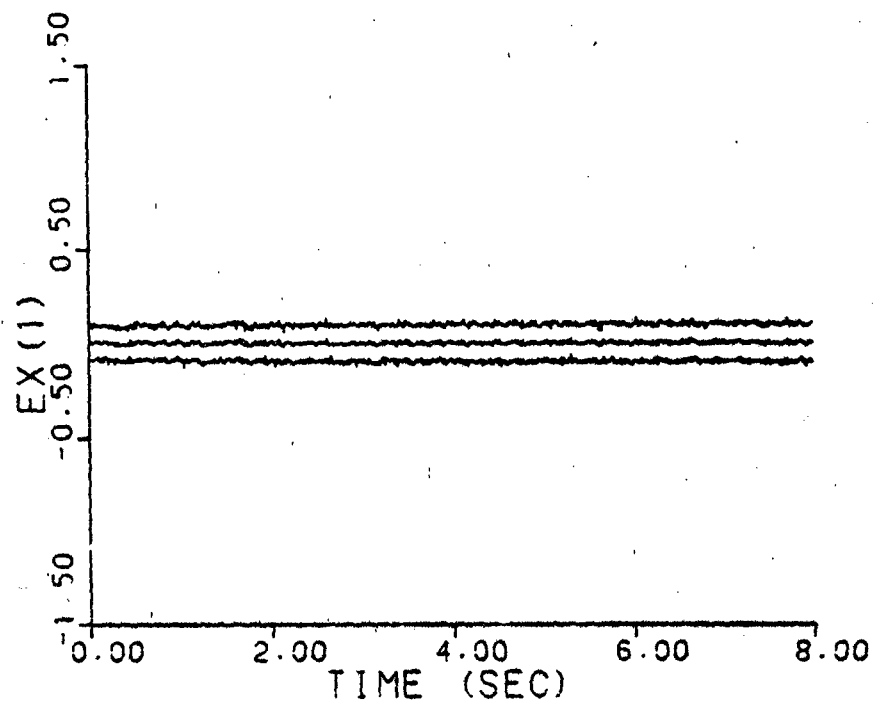


Figure A-2i. Residual Monitoring
a varies (0.07, 9.0) - (0.93, 41.0)
no dither

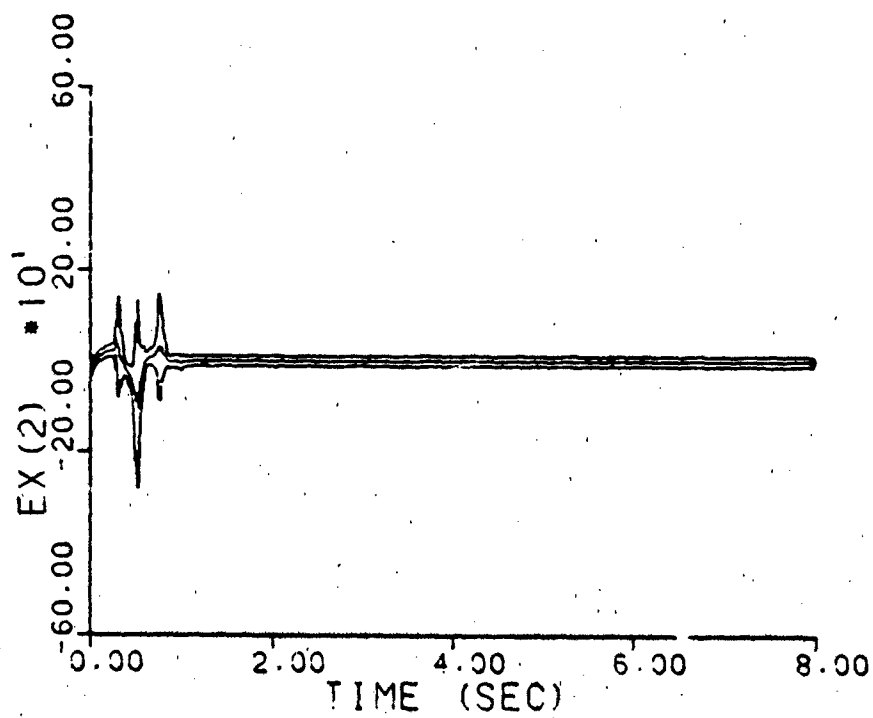
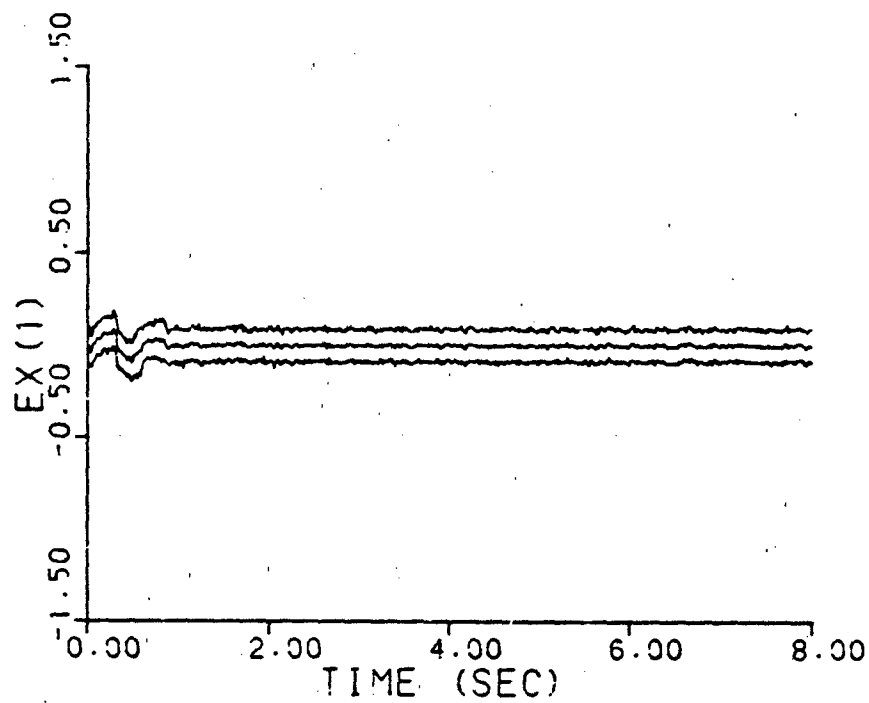


Figure A-3a. Residual Monitoring
 $\underline{a} = (1, 3)$

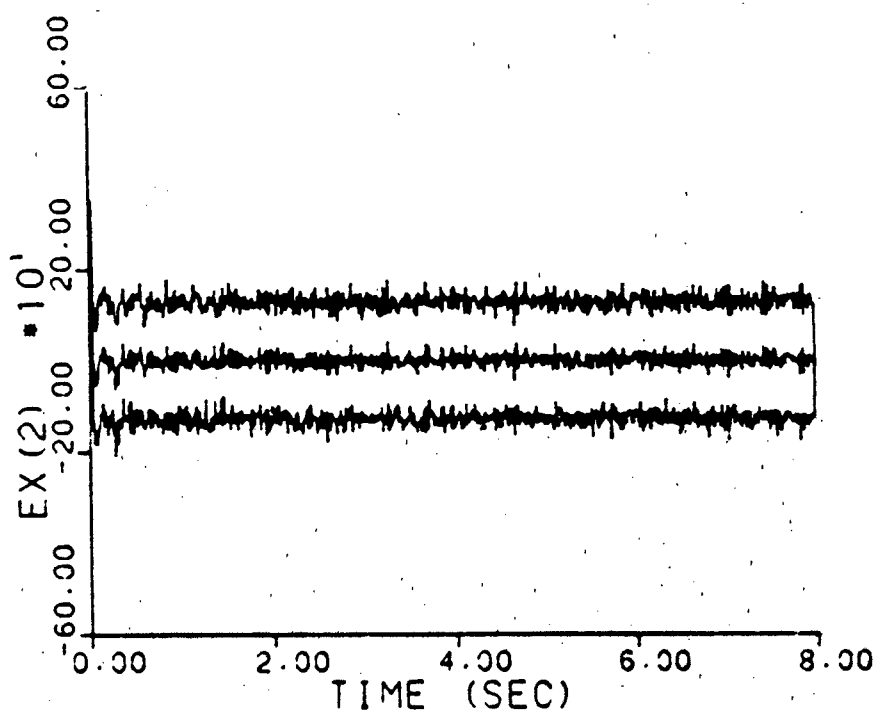
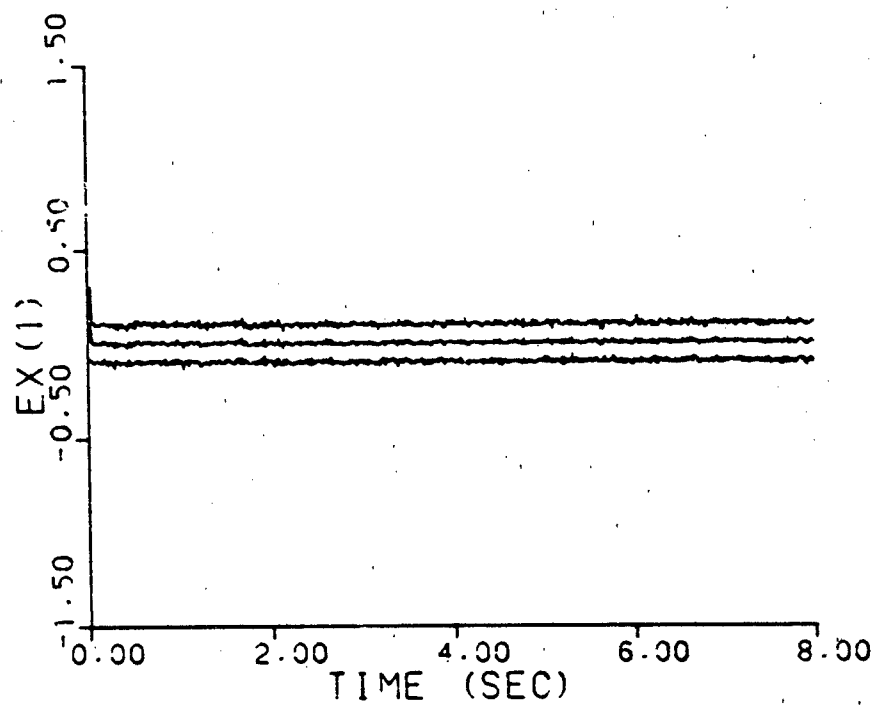


Figure A-3b. Residual Monitoring
 $\underline{a} = (2, 9)$

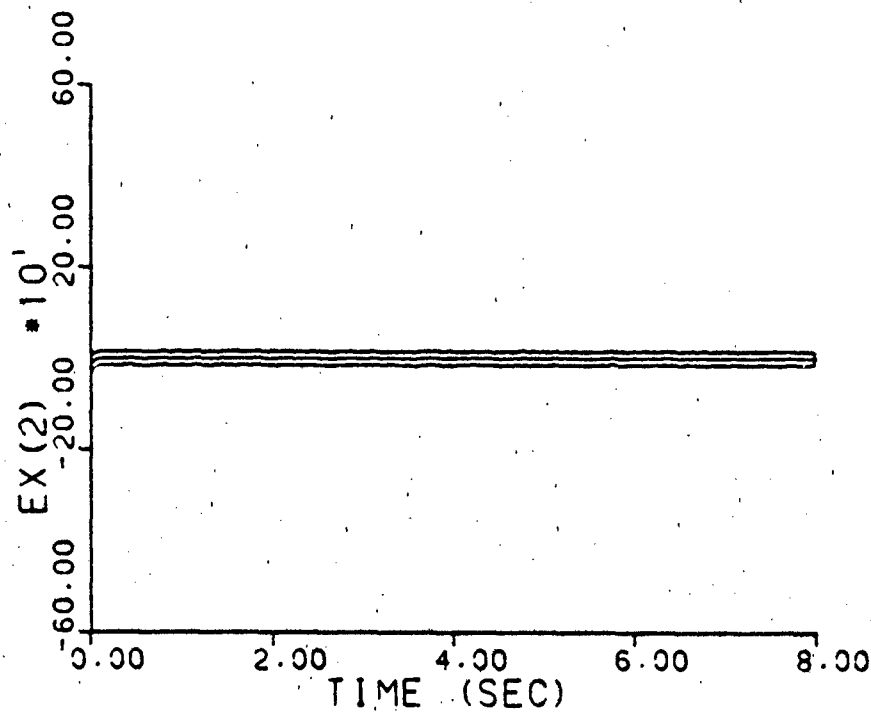
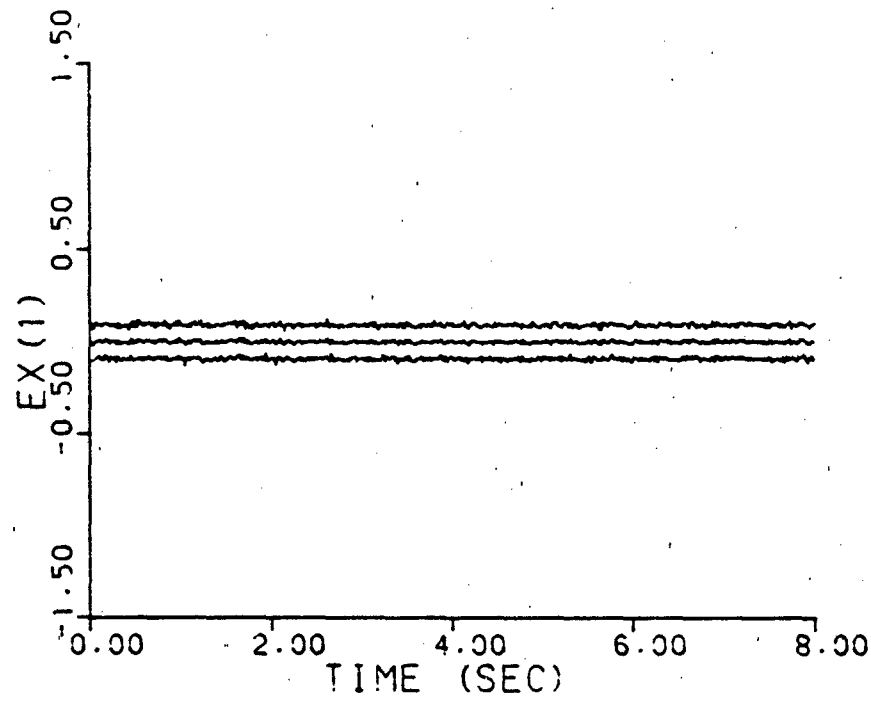


Figure A-3c. Residual Monitoring
 $\underline{a} = (5, 4)$

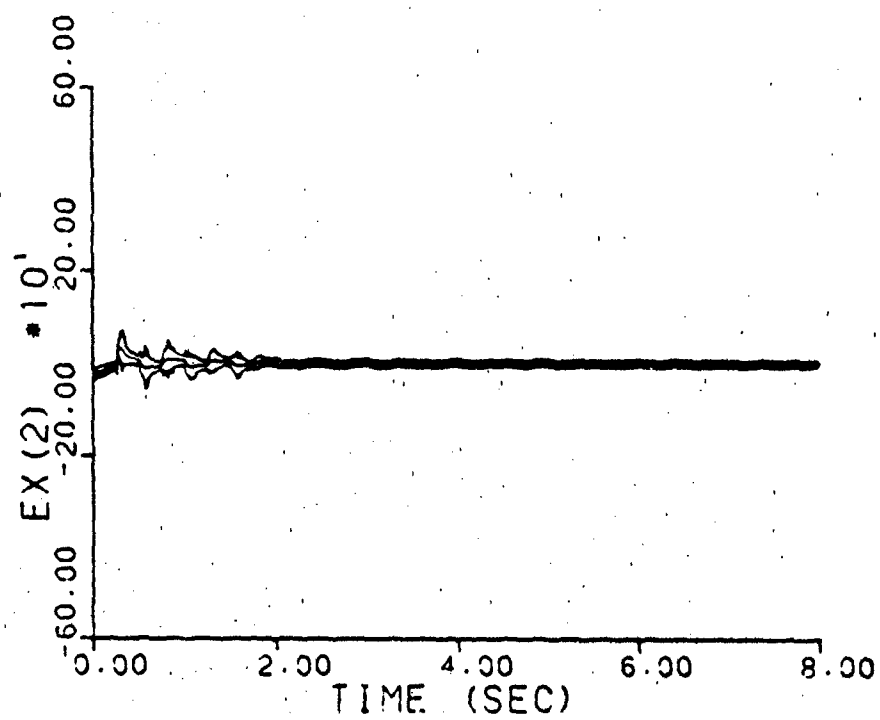
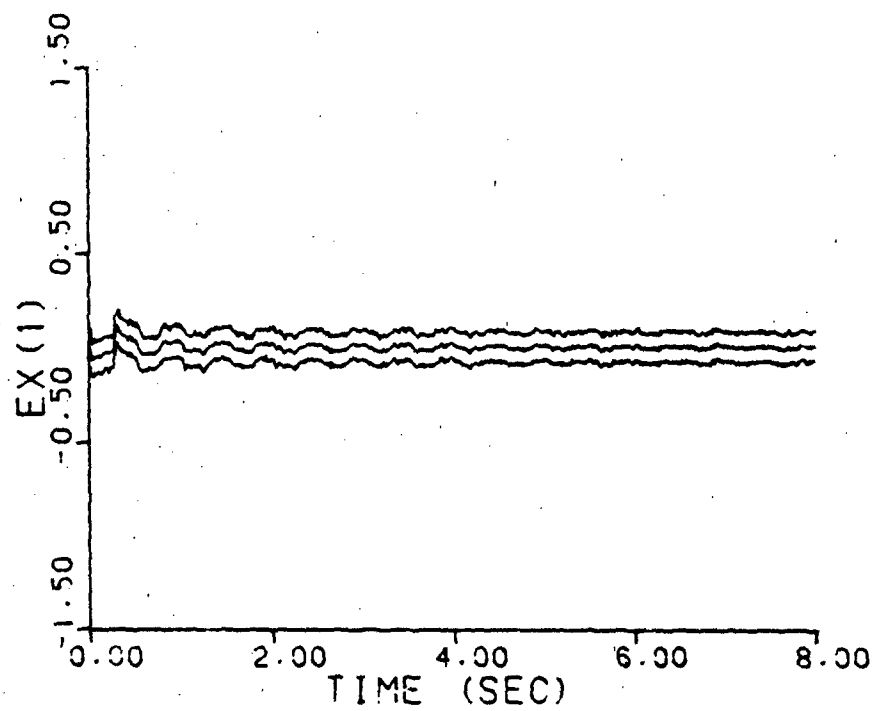


Figure A-3d. Residual Monitoring
 $\underline{a} = (9, 2)$

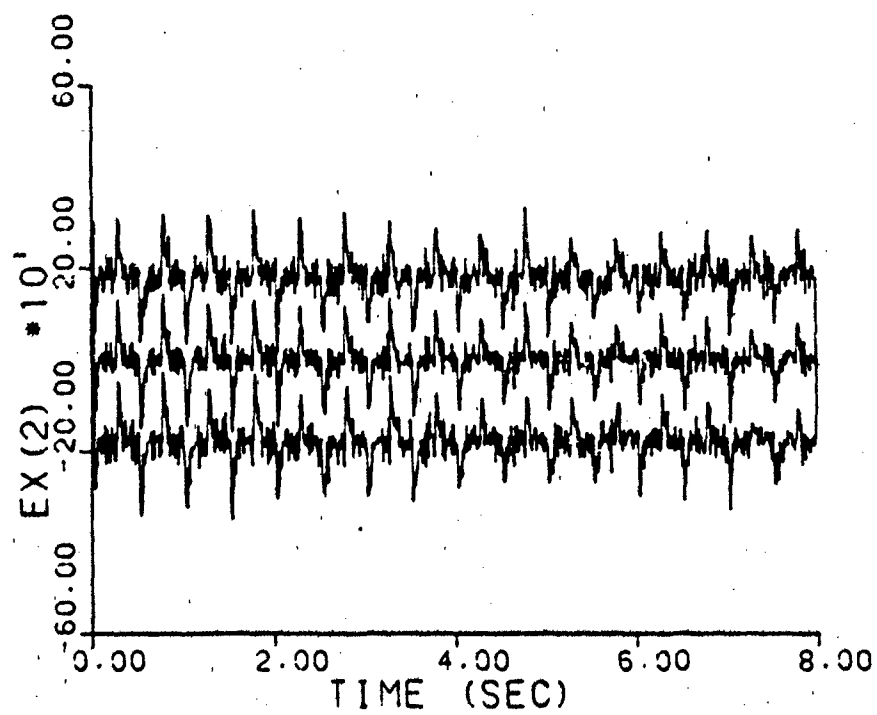
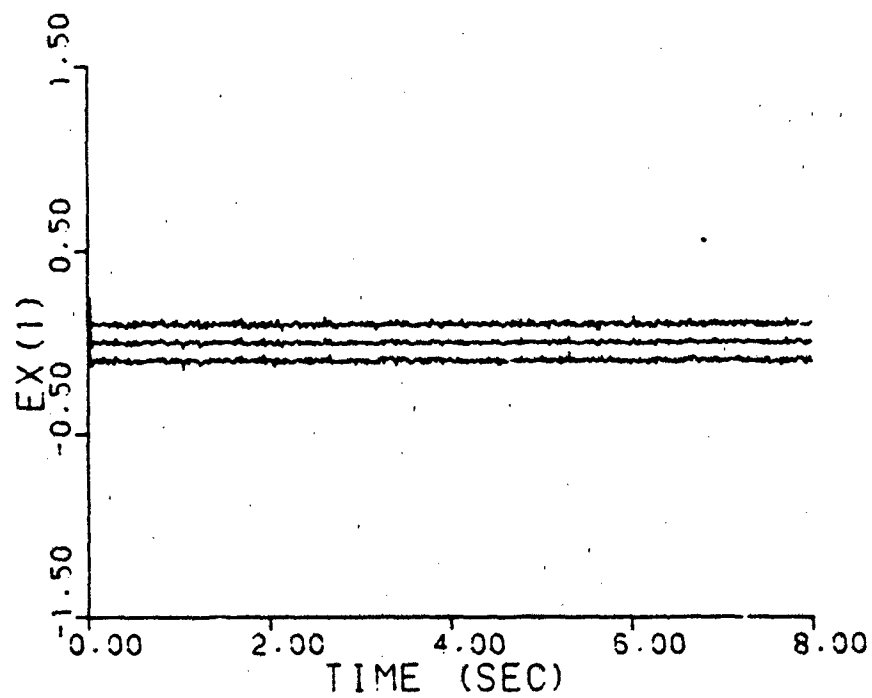


Figure A-3e. Residual Monitoring
 $\underline{a} = (10, 10)$

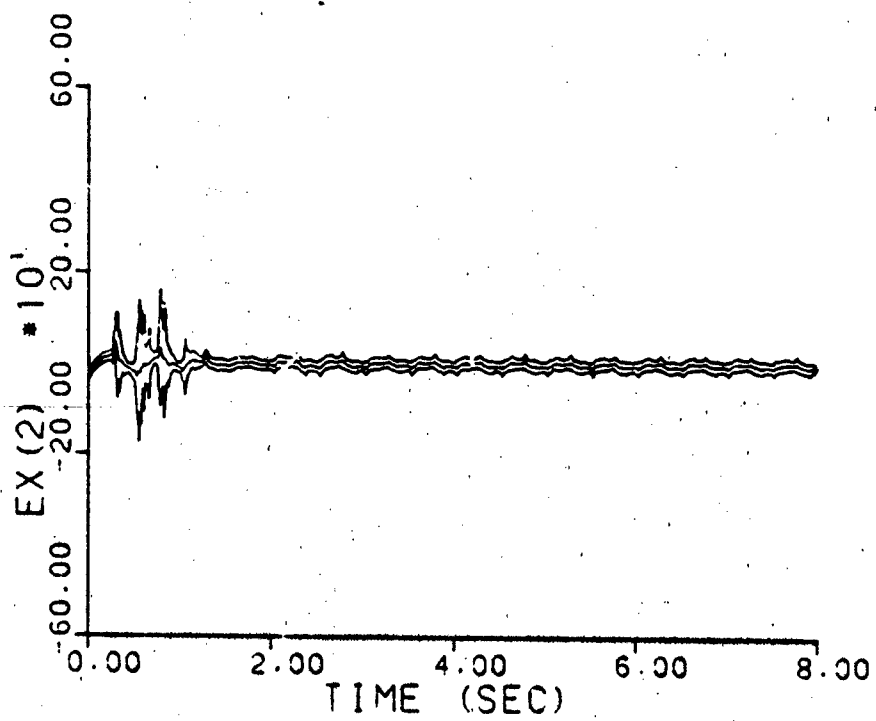
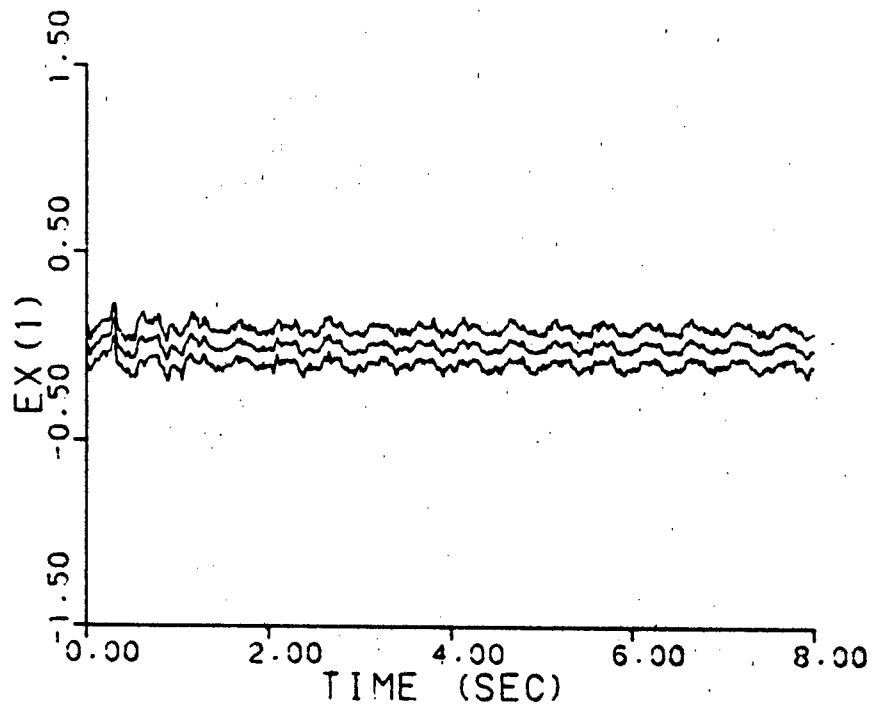


Figure A-3f. Residual Monitoring
 $\underline{a} = (0.07, 9.0)$

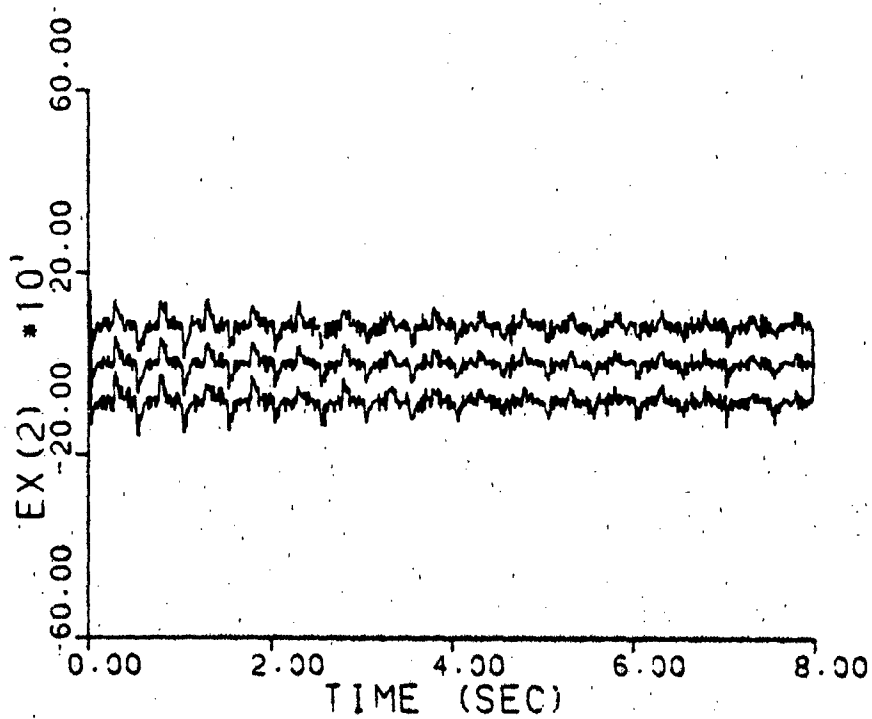
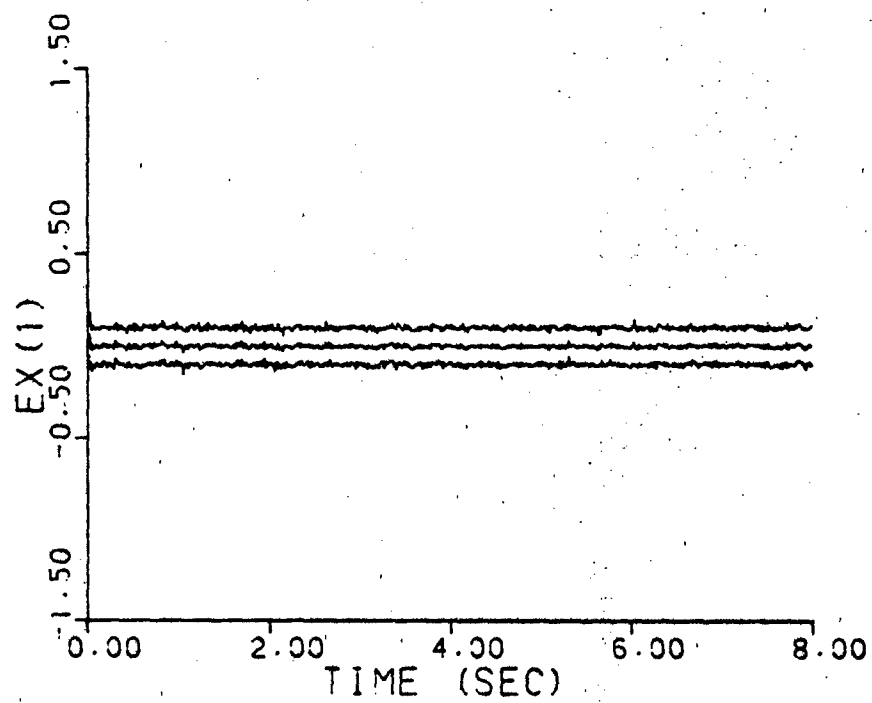


Figure A-3g. Residual Monitoring
 $\underline{a} = (0.93, 41.0)$

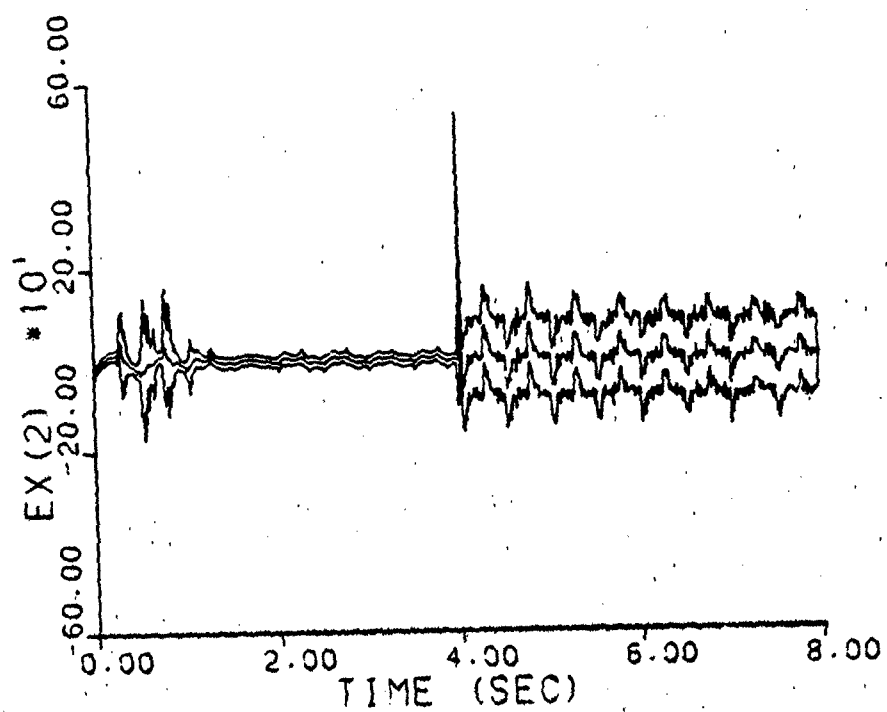
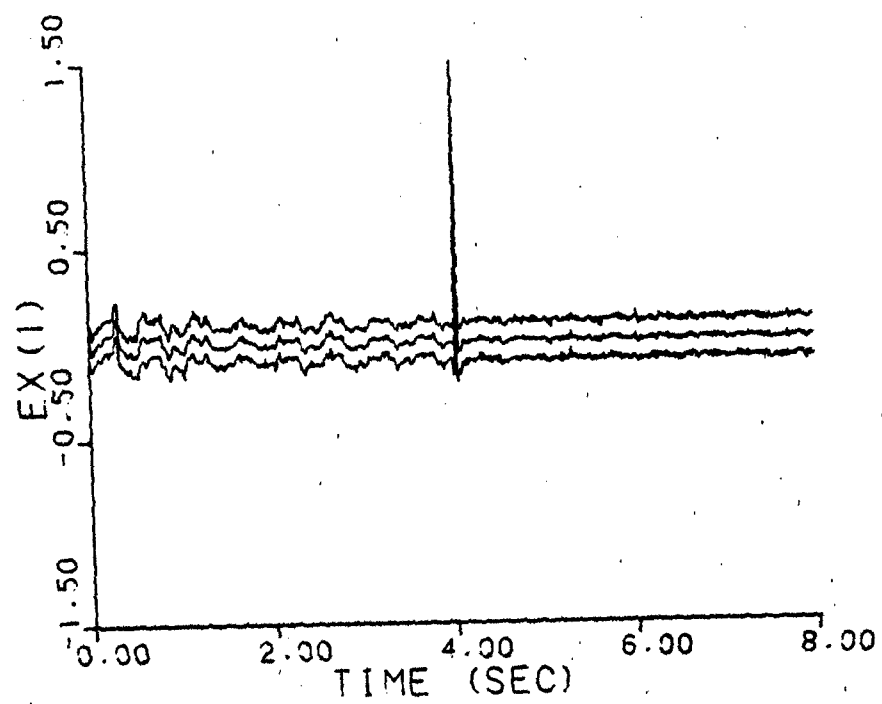


Figure A-3h. Residual Monitoring
a jumps: (0.07, 9.0) - (0.93, 41.0)

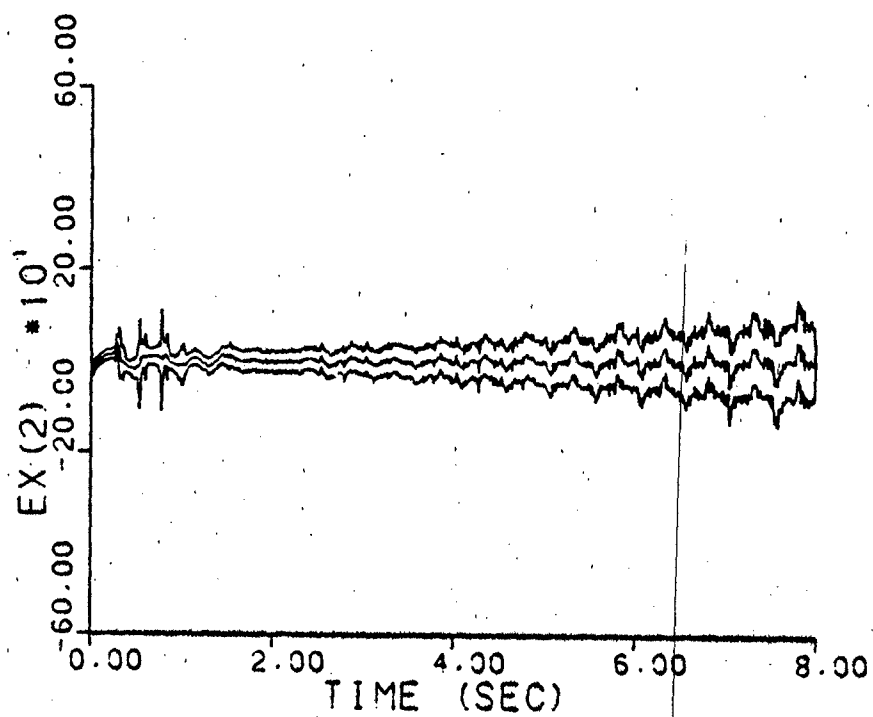
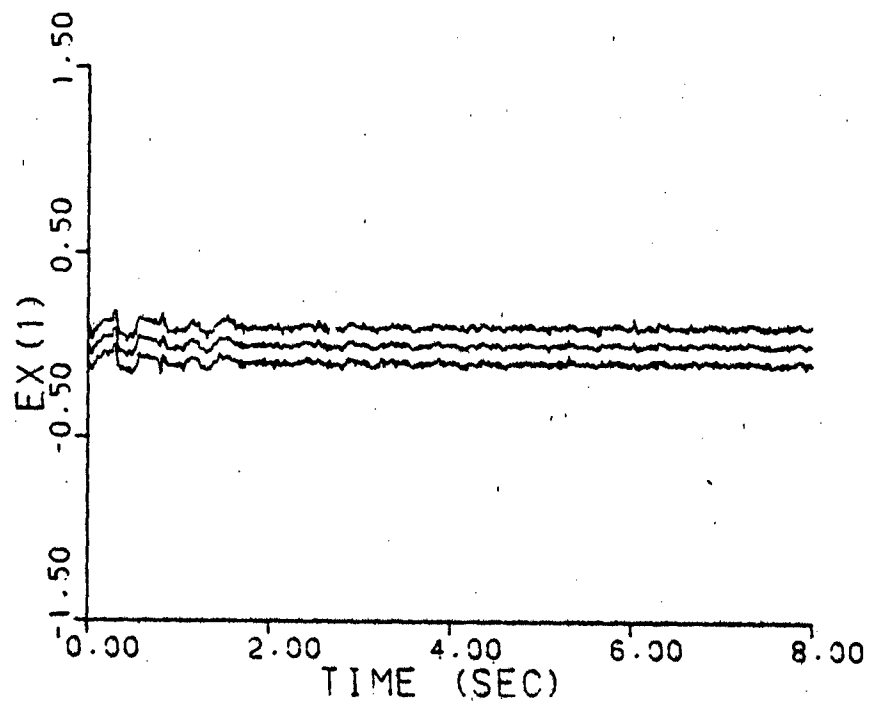


Figure A-31. Residual Monitoring
a varies: (0.07, 9.0) - (0.93, 41.0)

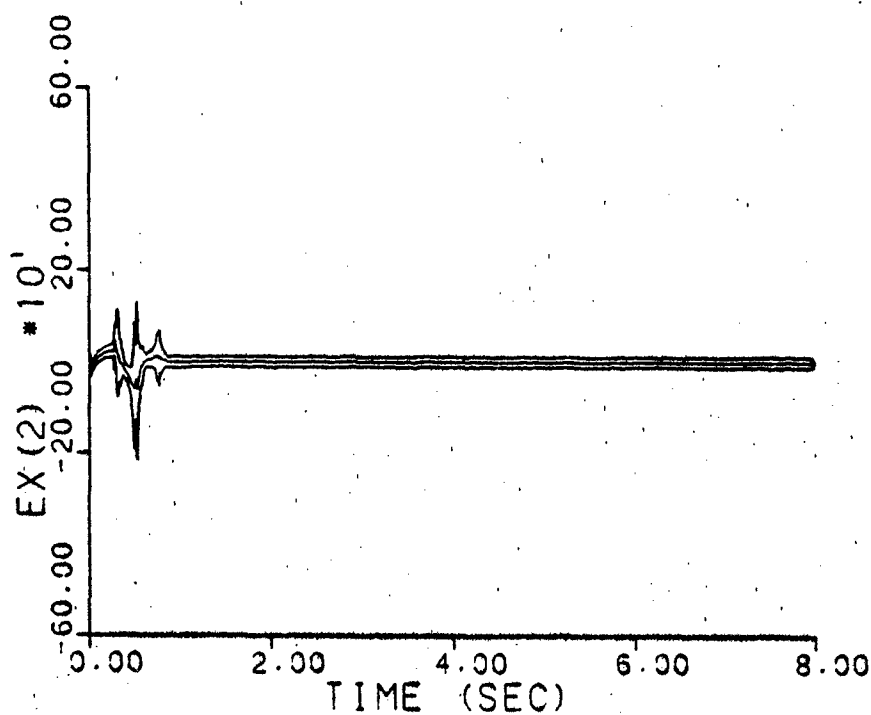
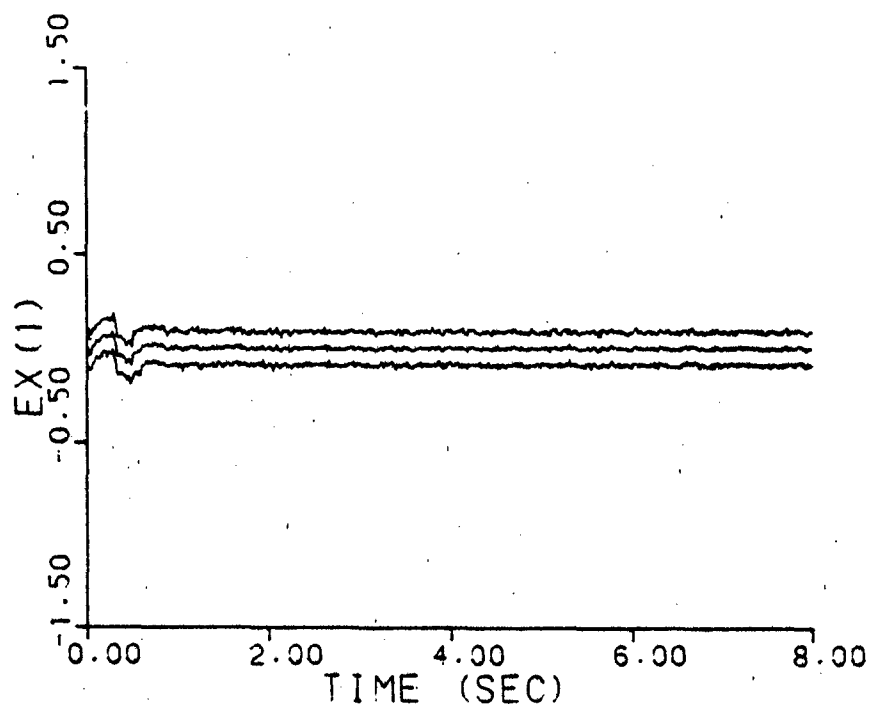


Figure A-4a. Residual Monitoring
 $\underline{a} = (1, 3)$, warm-up

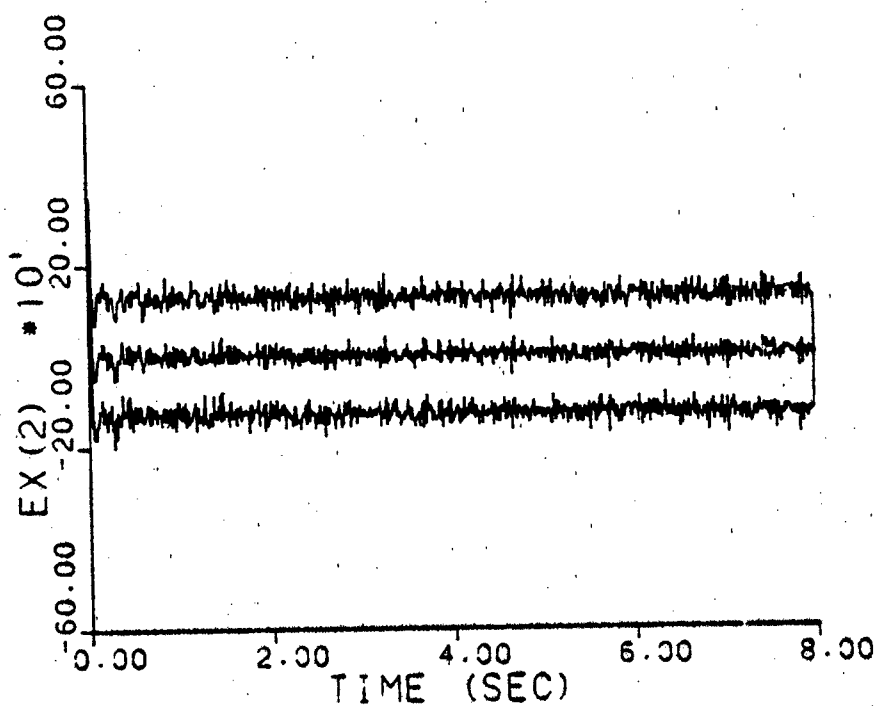
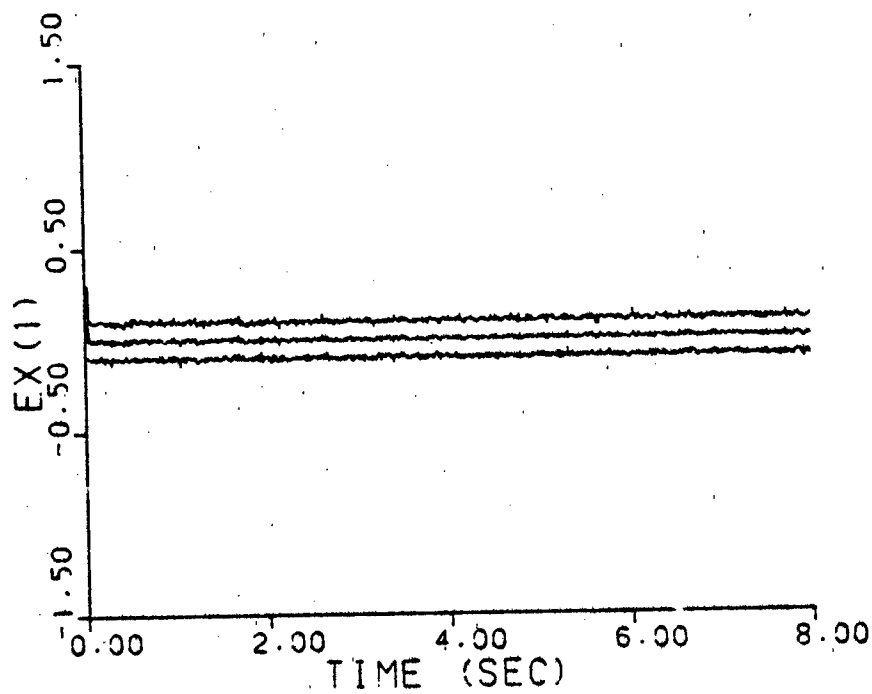


Figure A-4b. Residual Monitoring
 $\underline{a} = (2, 9)$, warm-up

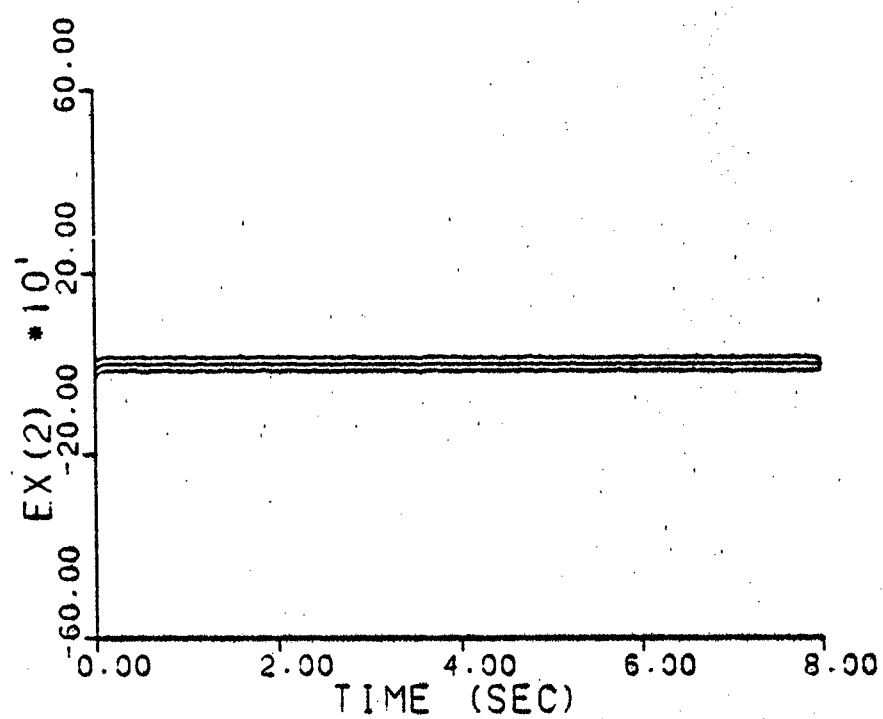
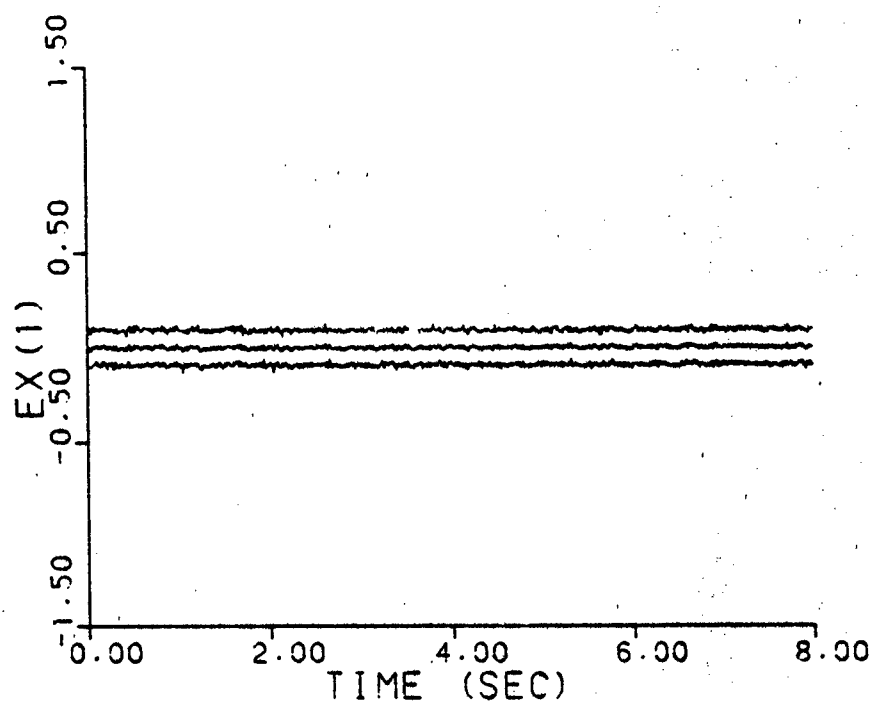


Figure A-4c. Residual Monitoring
 $\underline{a} = (5, 4)$, warm-up

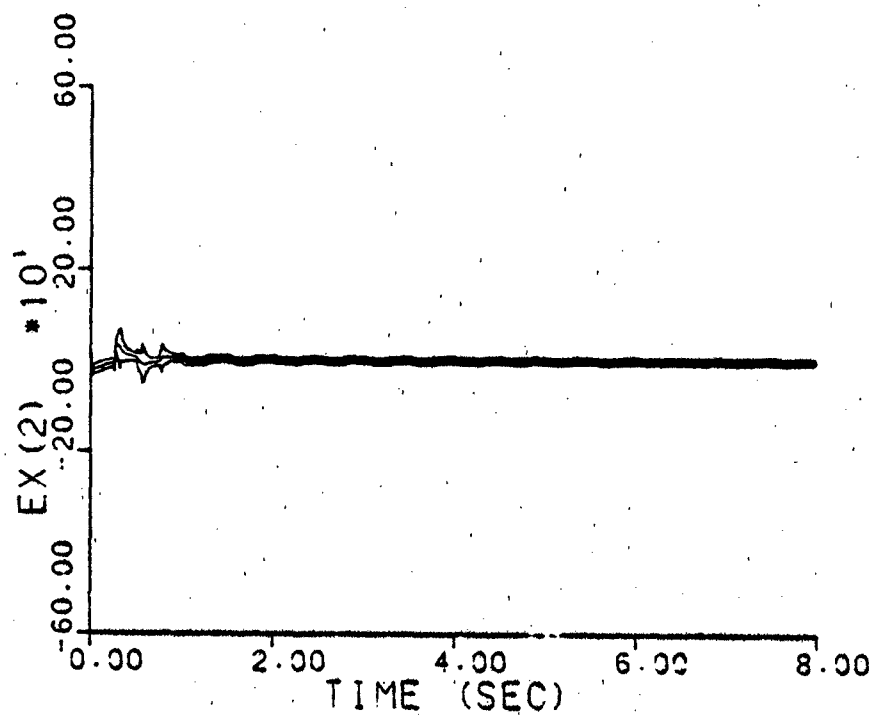
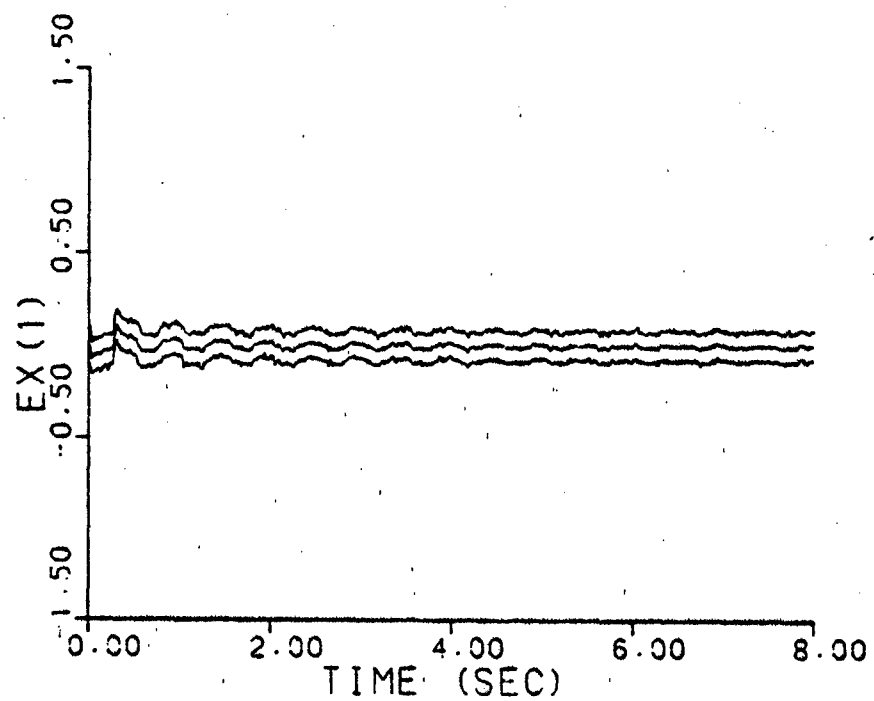


Figure A-4d. Residual Monitoring
 $a = (9, 2)$, warm-up

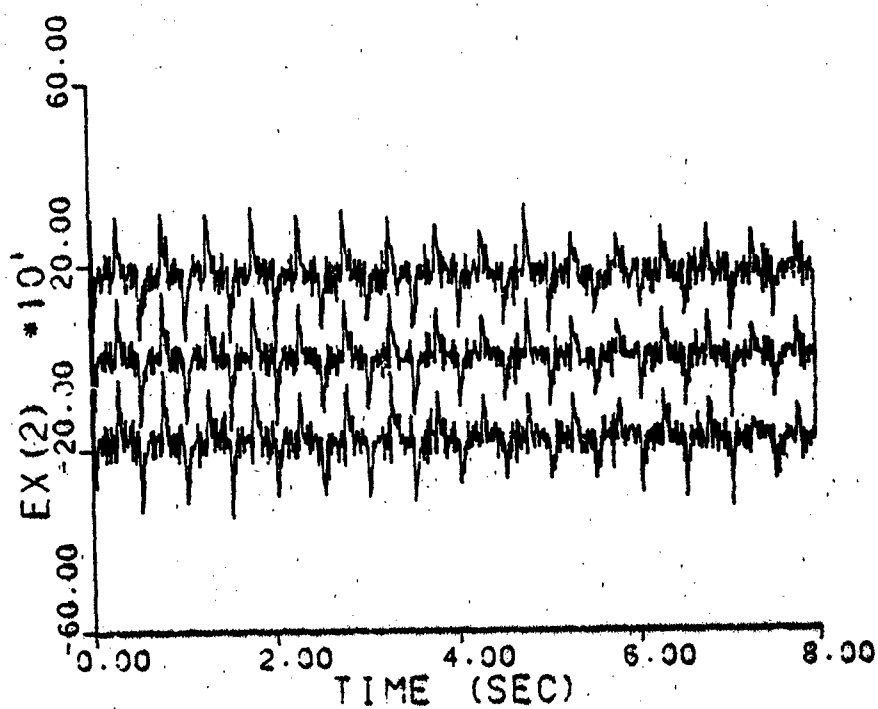
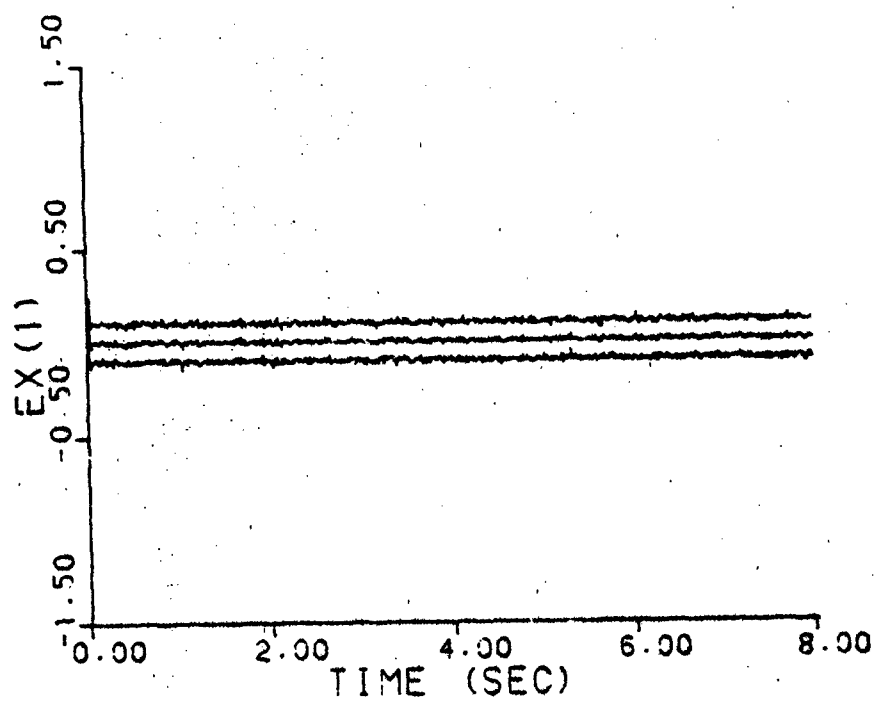


Figure A-4e. Residual Monitoring
 $\underline{a} = (10, 10)$, warm-up

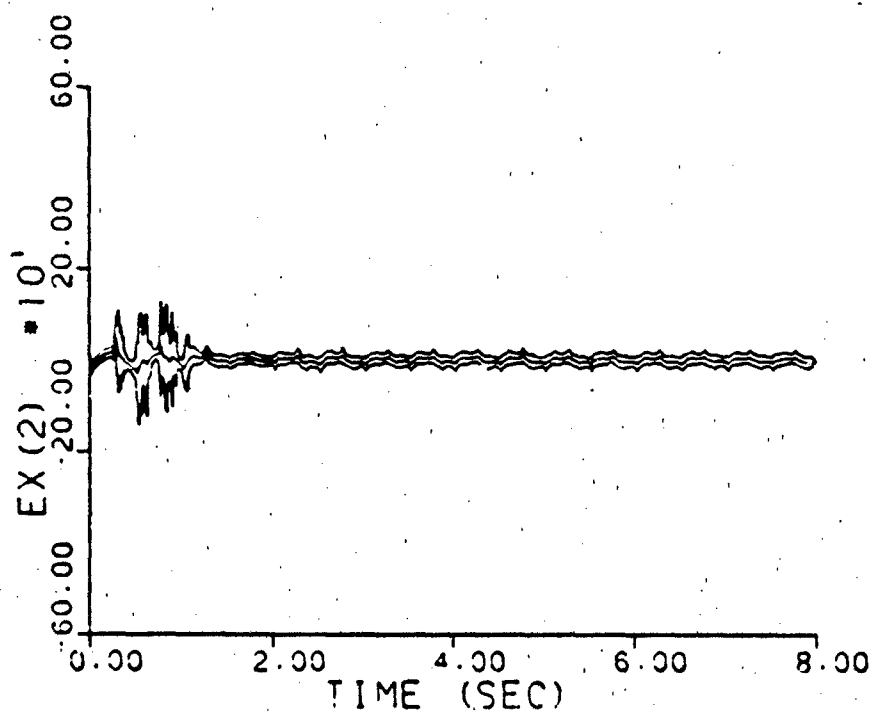
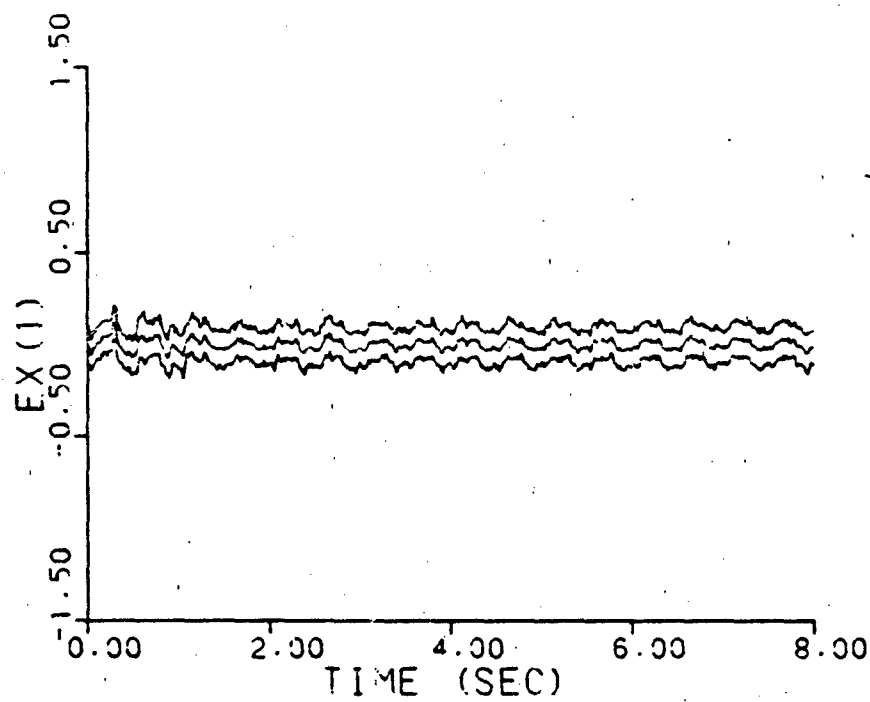


Figure A-4f. Residual Monitoring
 $\underline{a} = (0.07, 9.0)$, warm-up

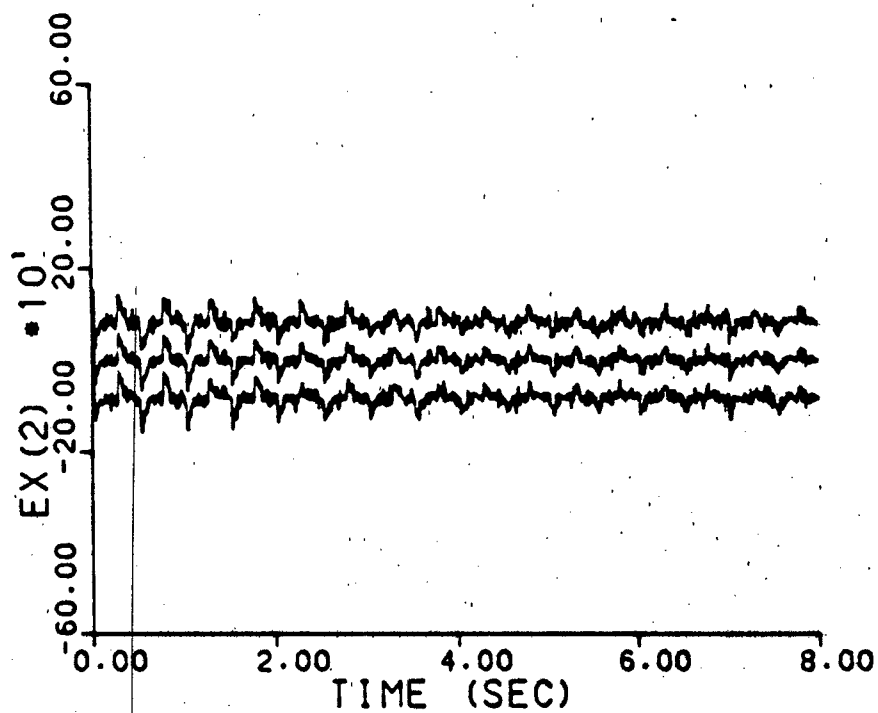
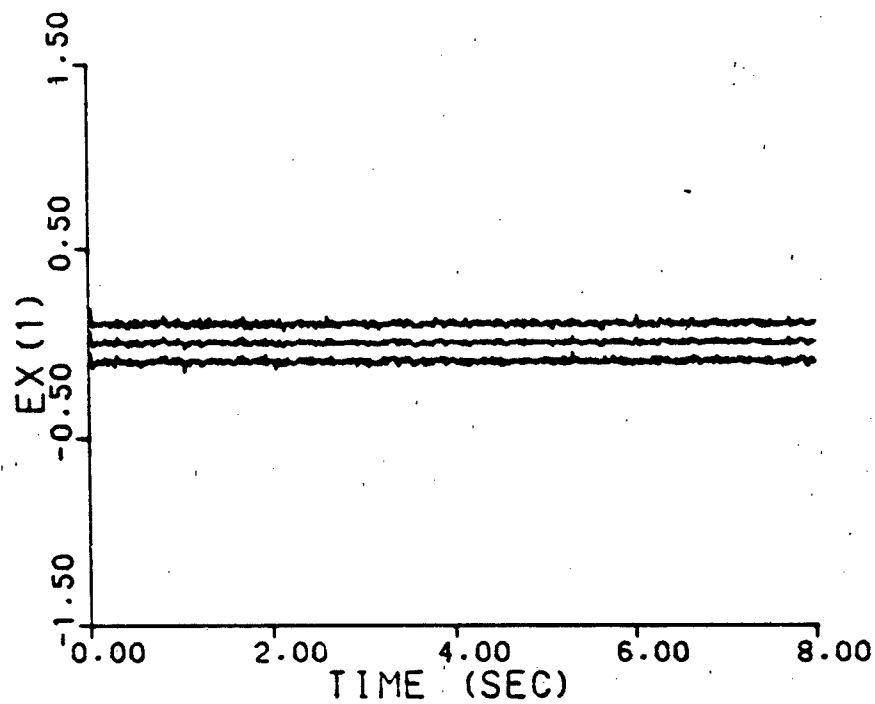


Figure A-4g. Residual Monitoring
 $\underline{a} = (0.93, 41.0)$, warm-up

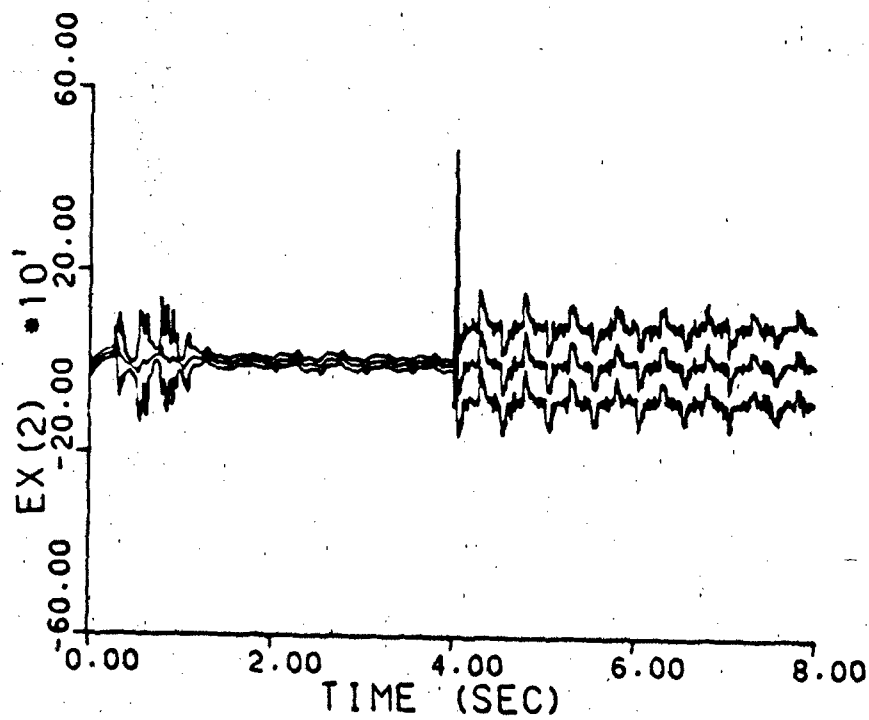
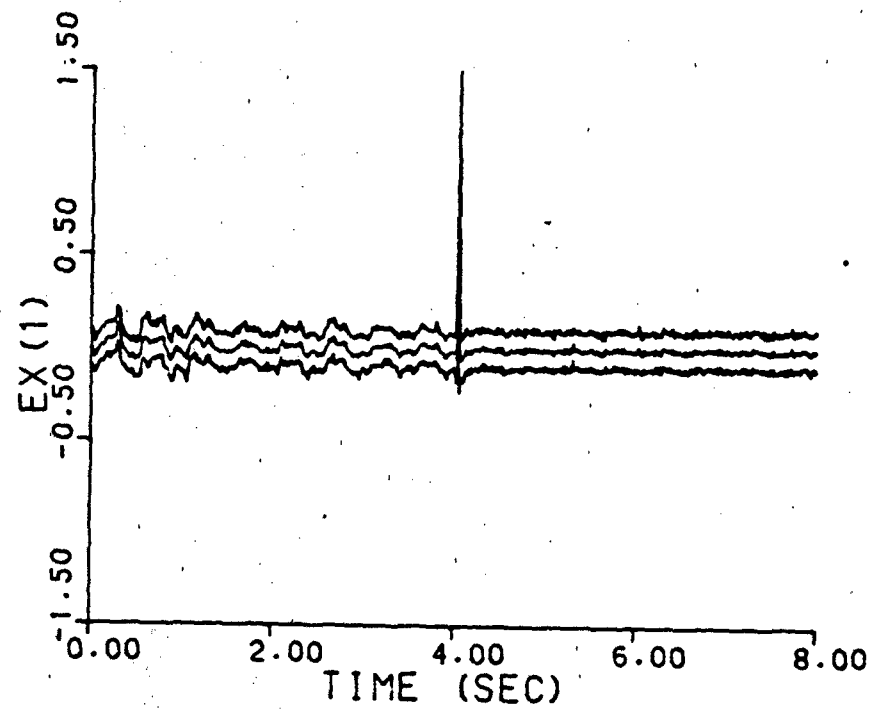


Figure A-4h. Residual Monitoring
a jumps: (0.07, 9.0) - (0.93, 41.0)
warm-up

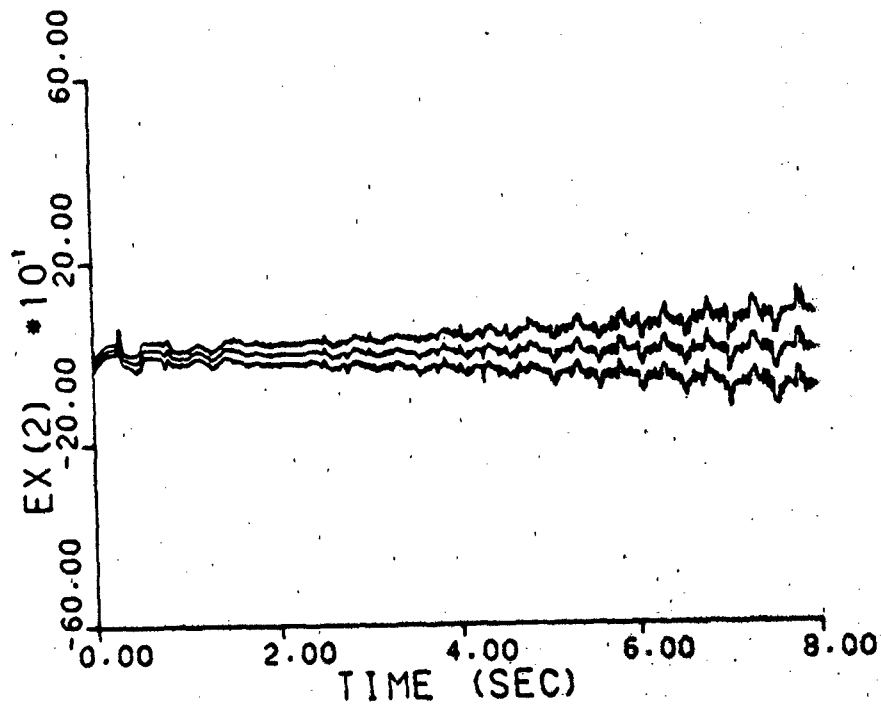
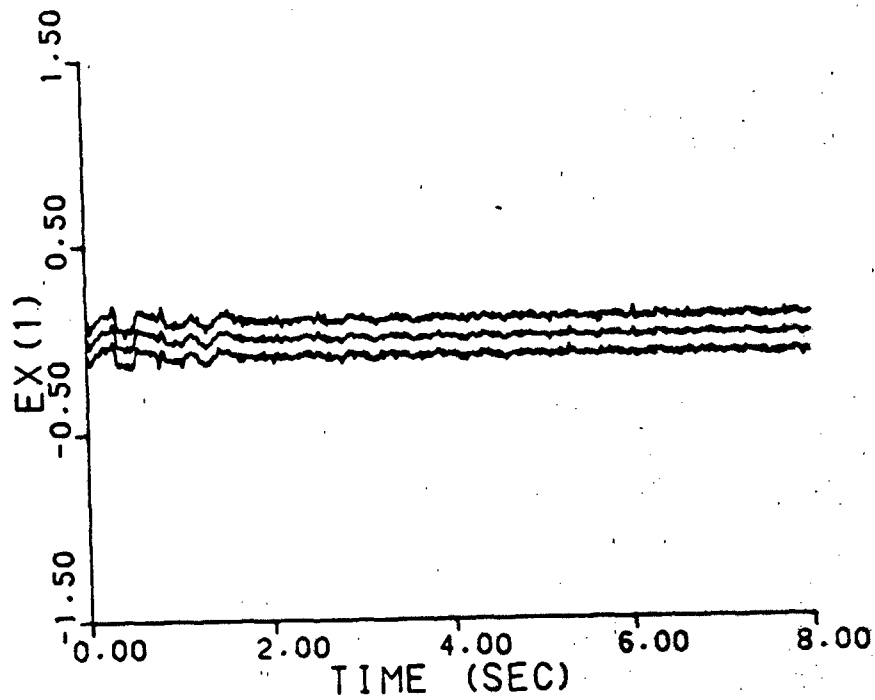


Figure A-41. Residual Monitoring
a varies: (0.07, 9.0) - (0.93, 41.0)
warm-up

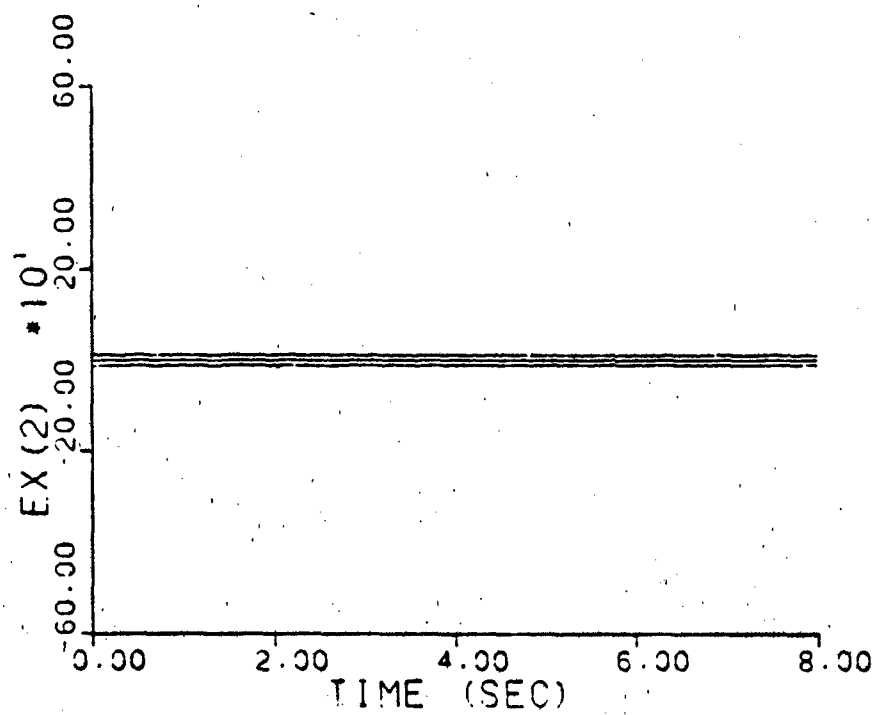
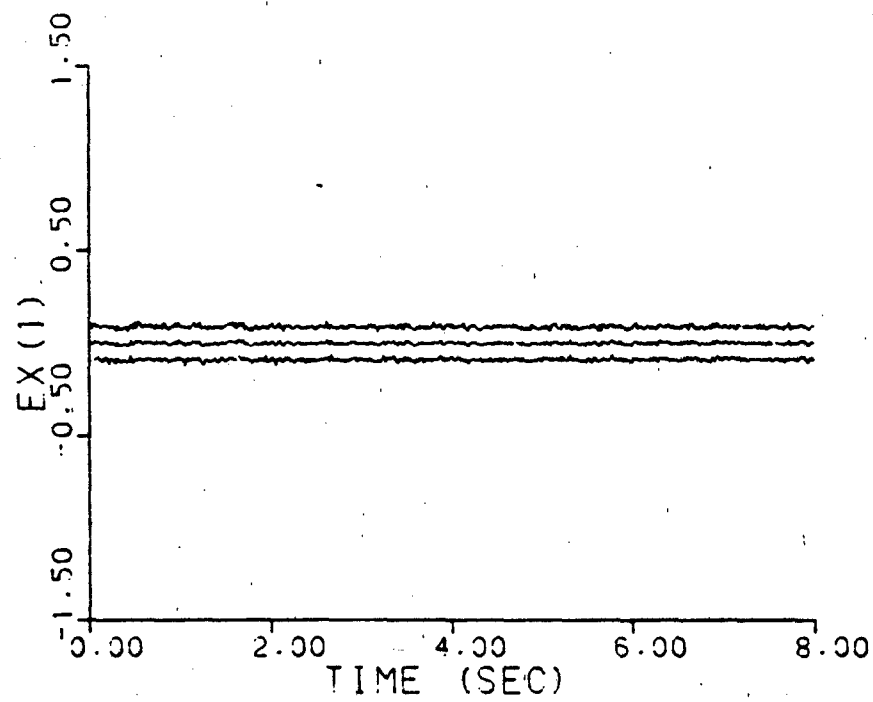


Figure A-5a. Parameter Position Estimate Monitoring
 $\underline{a} = (1, 3)$, no dither

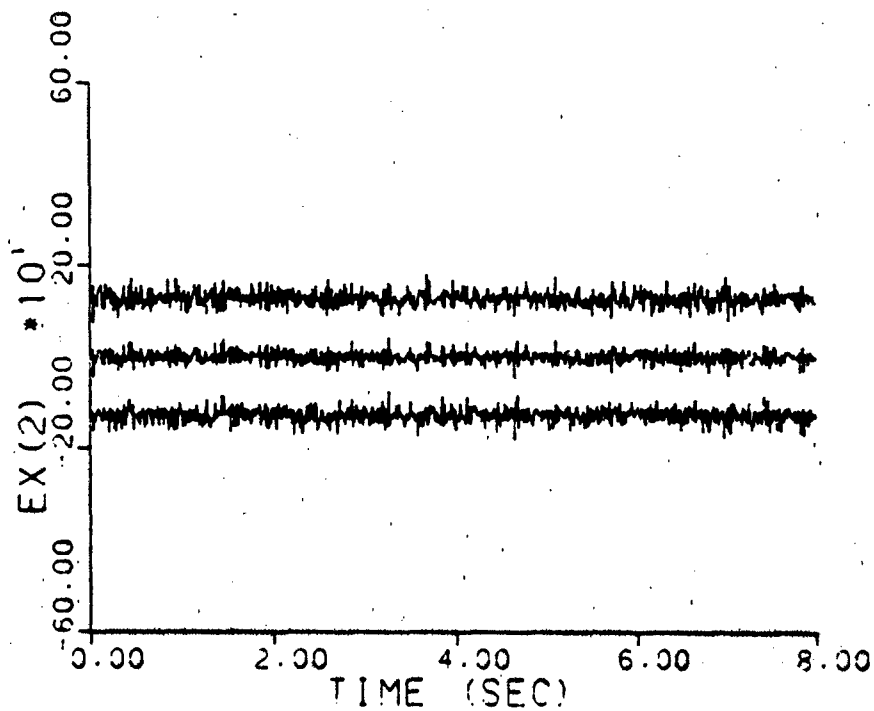
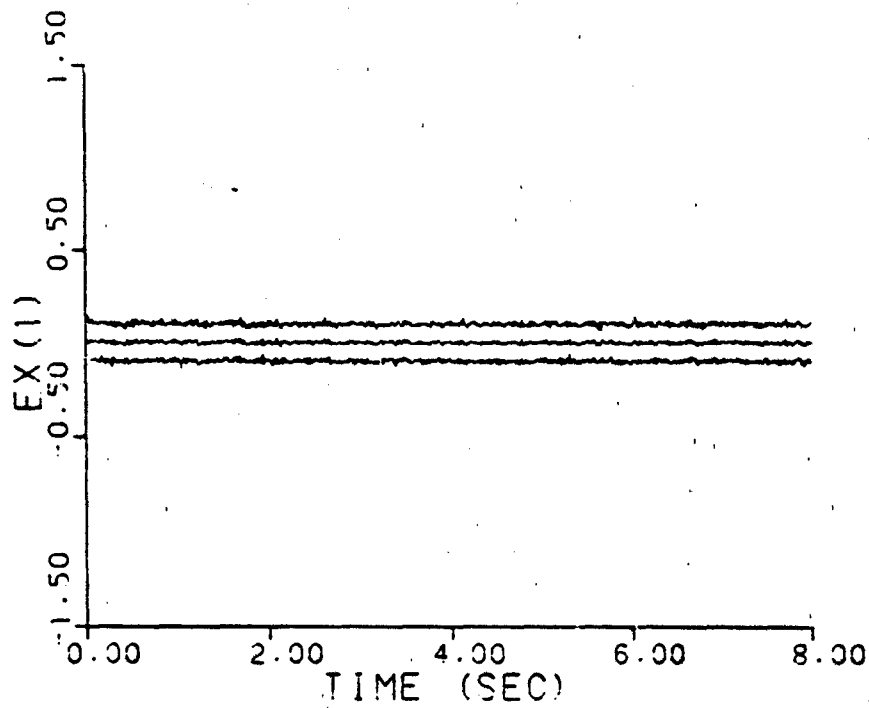


Figure A-5b. Parameter Position Estimate Monitoring
 $\underline{a} = (2, 9)$, no dither

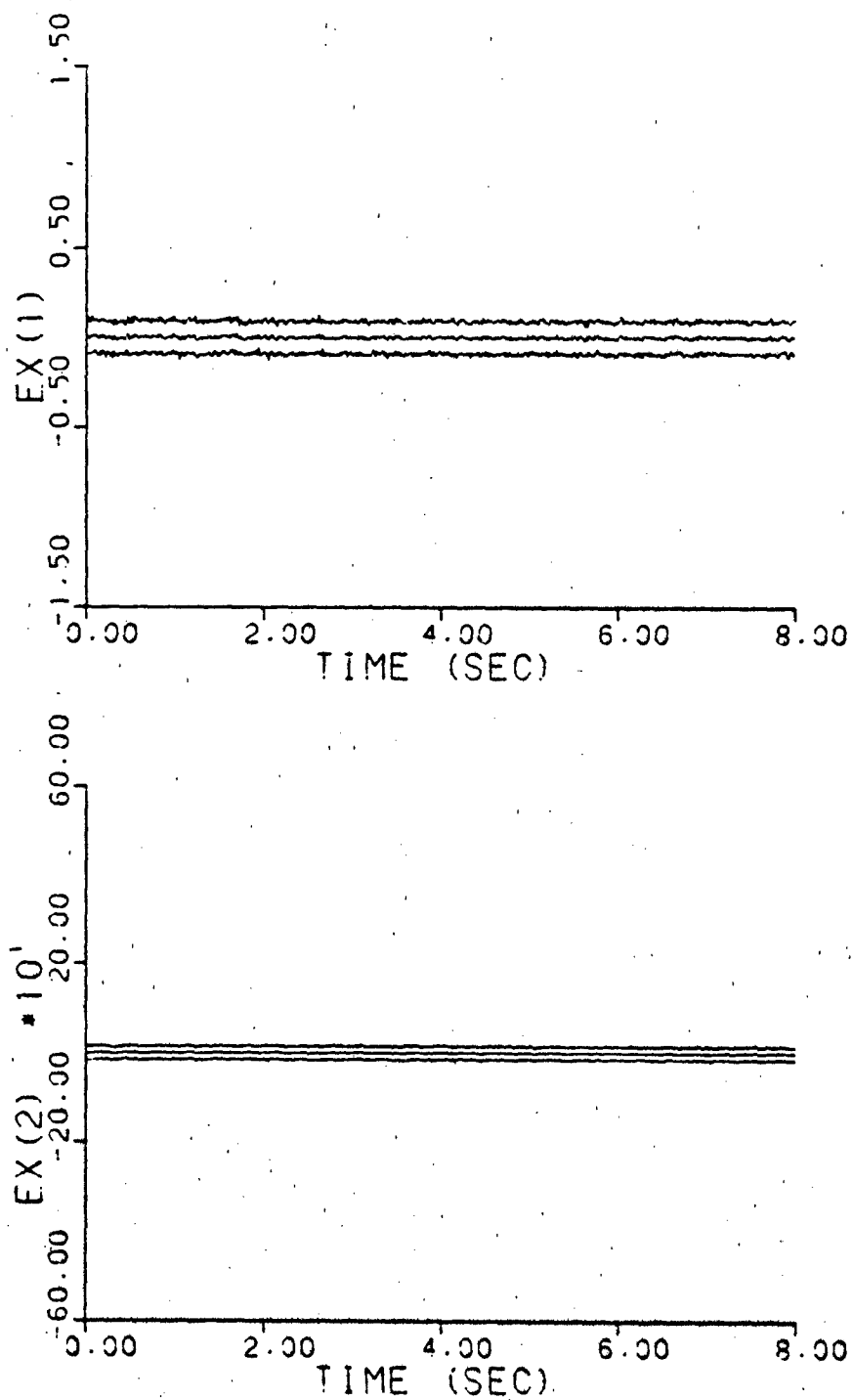


Figure A-5c. Parameter Position Estimate Monitoring
 $\underline{a} = (5, 4)$, no dither

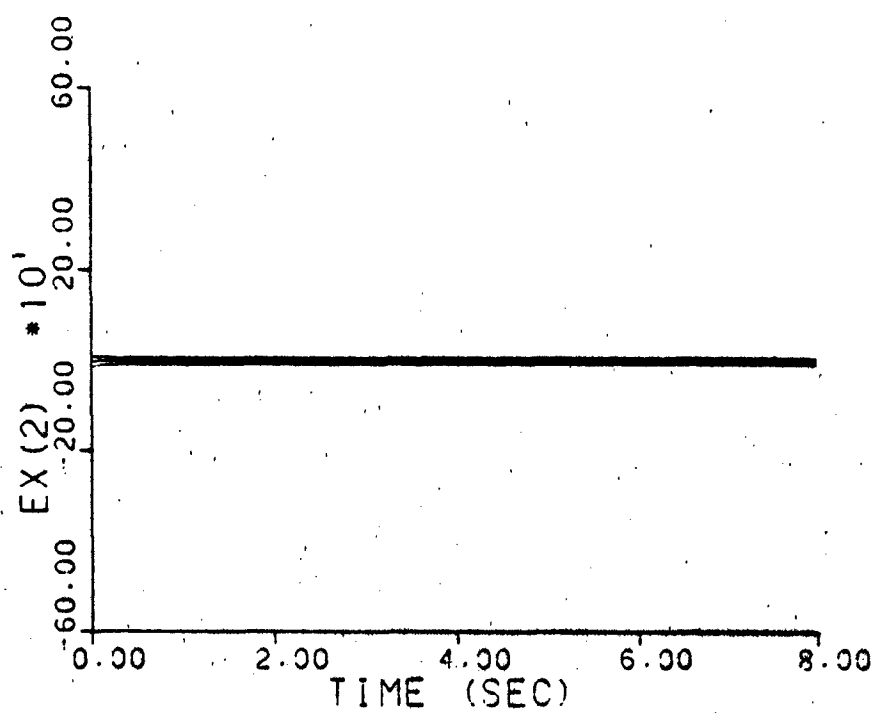
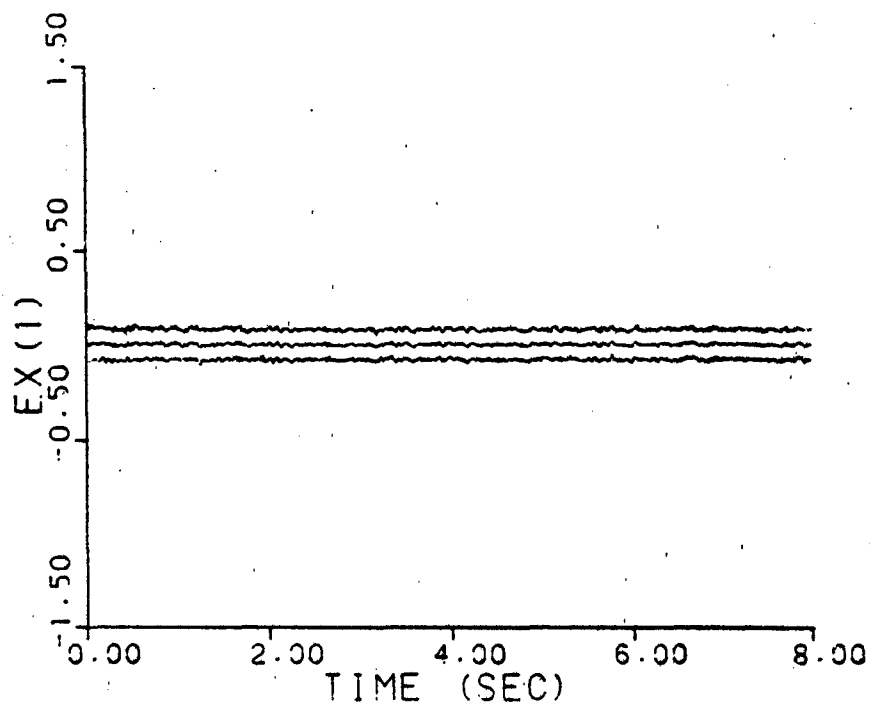


Figure A-5d. Parameter Position Estimate Monitoring
 $\underline{a} = (9, 2)$, no dither

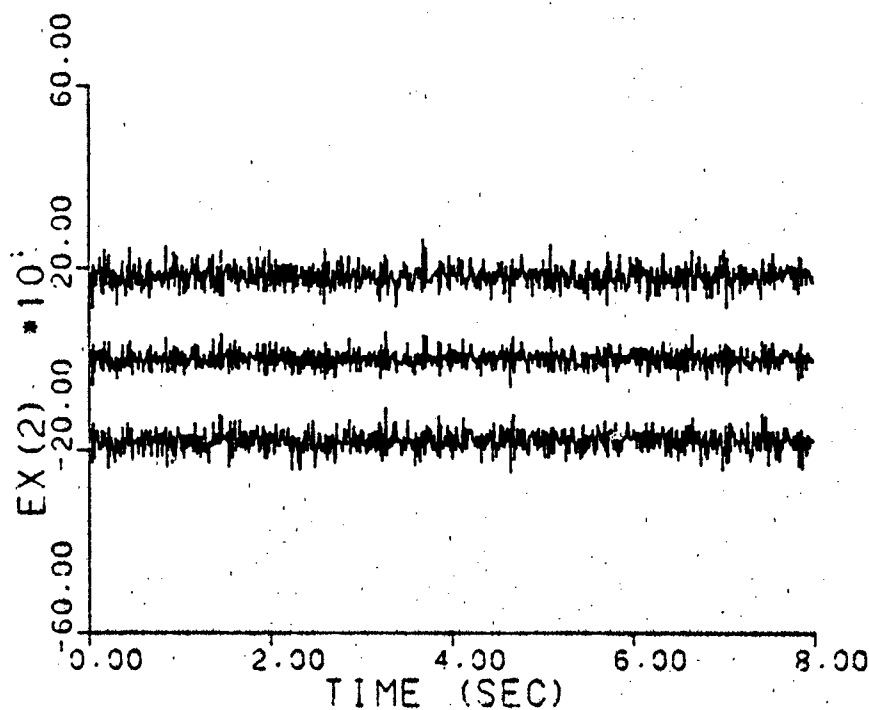
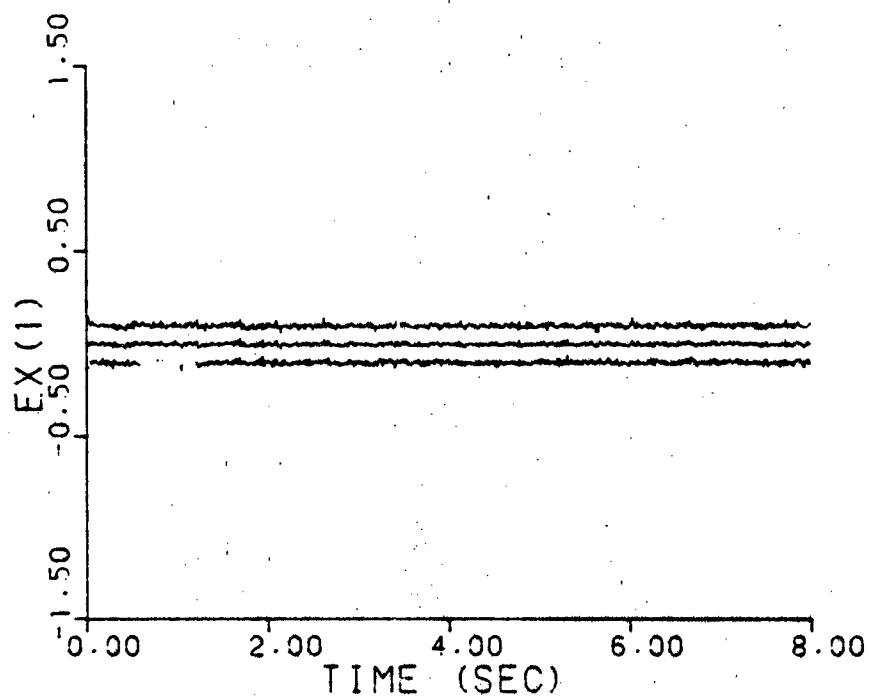


Figure A-5e. Parameter Position Estimate Monitoring
 $\underline{a} = (10, 10)$, no dither

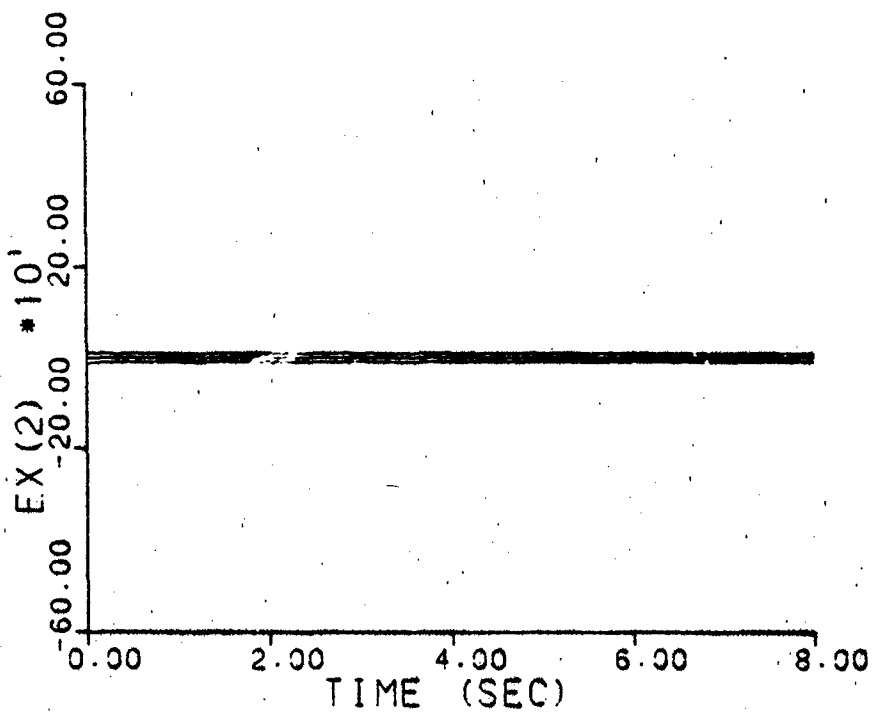
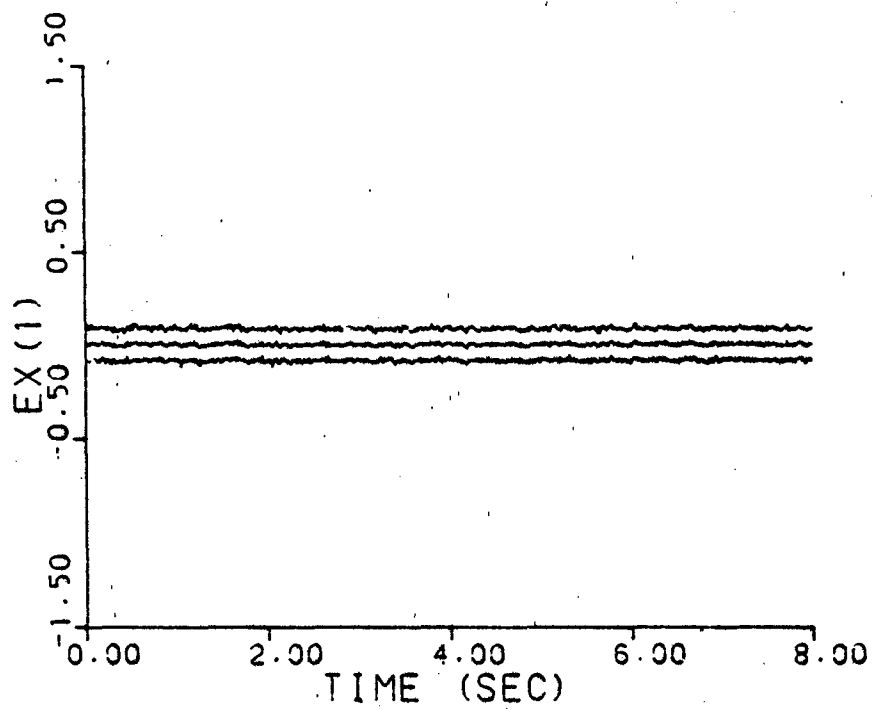


Figure A-5f. Parameter Position Estimate Monitoring
 $\underline{a} = (0.07, 9.0)$, no dither

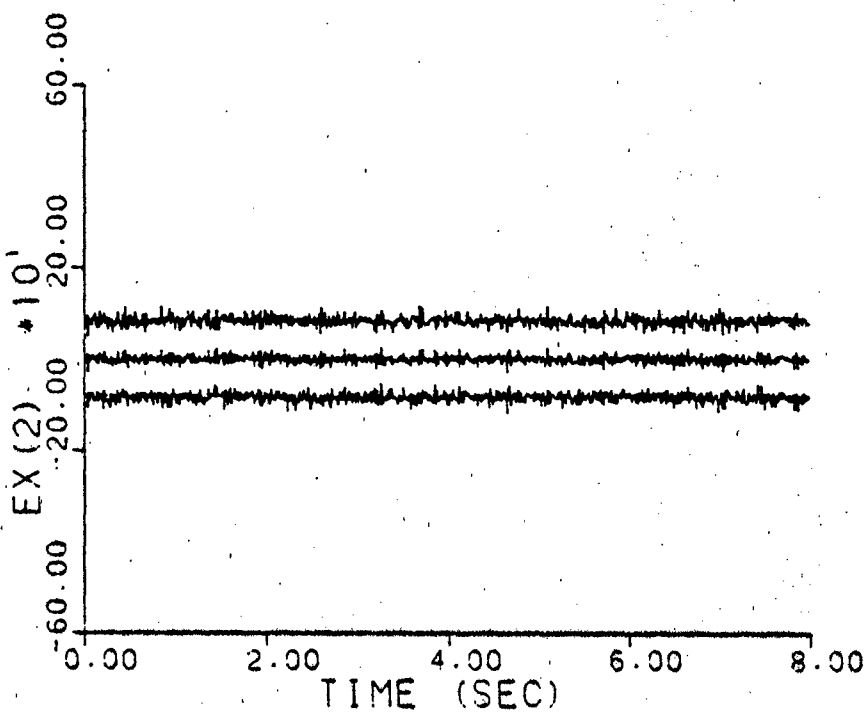
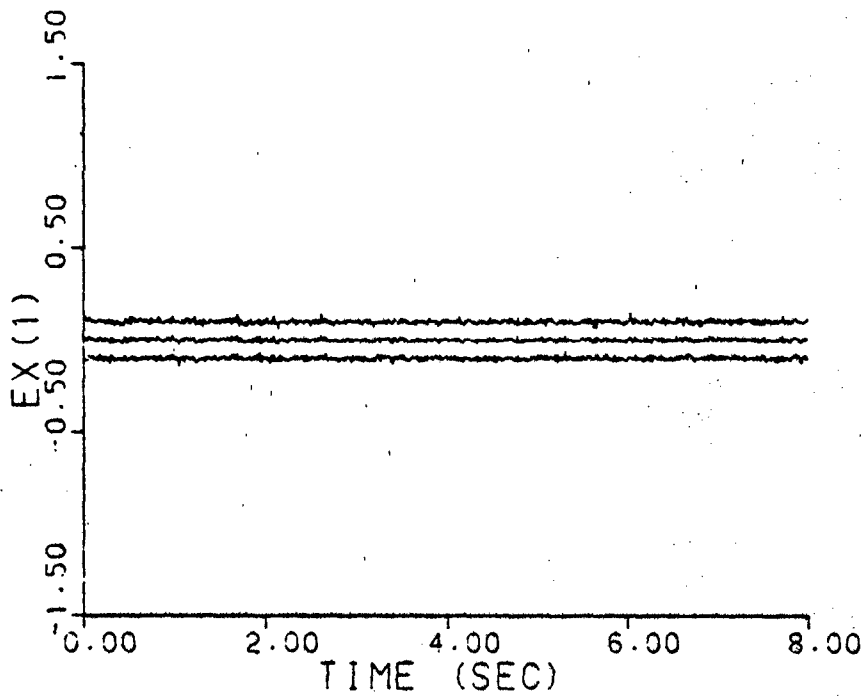


Figure A-5g. Parameter Position Estimate Monitoring
 $\underline{a} = (0.93, 41.0)$, no dither

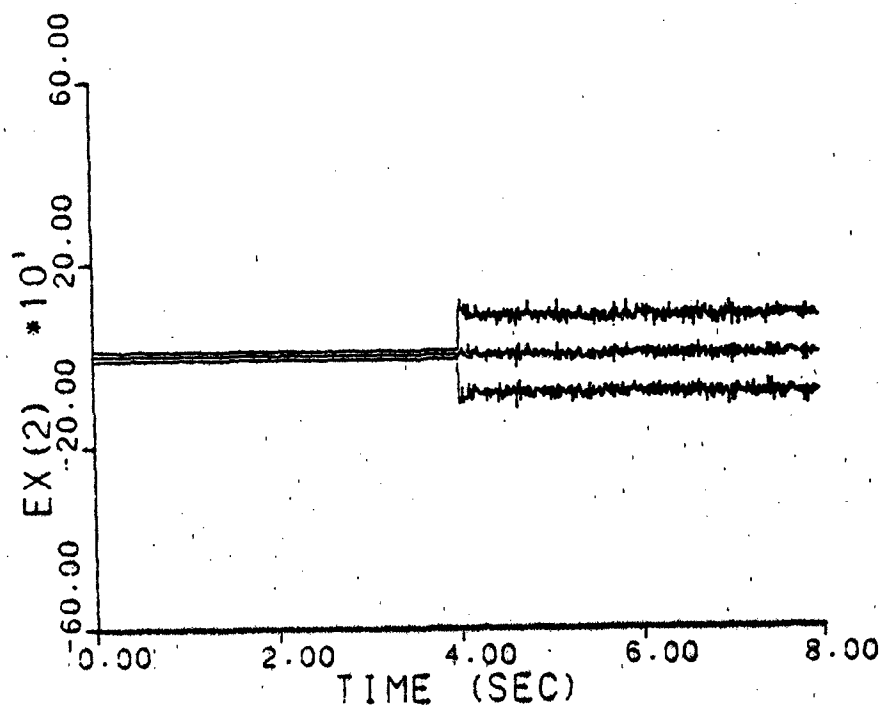
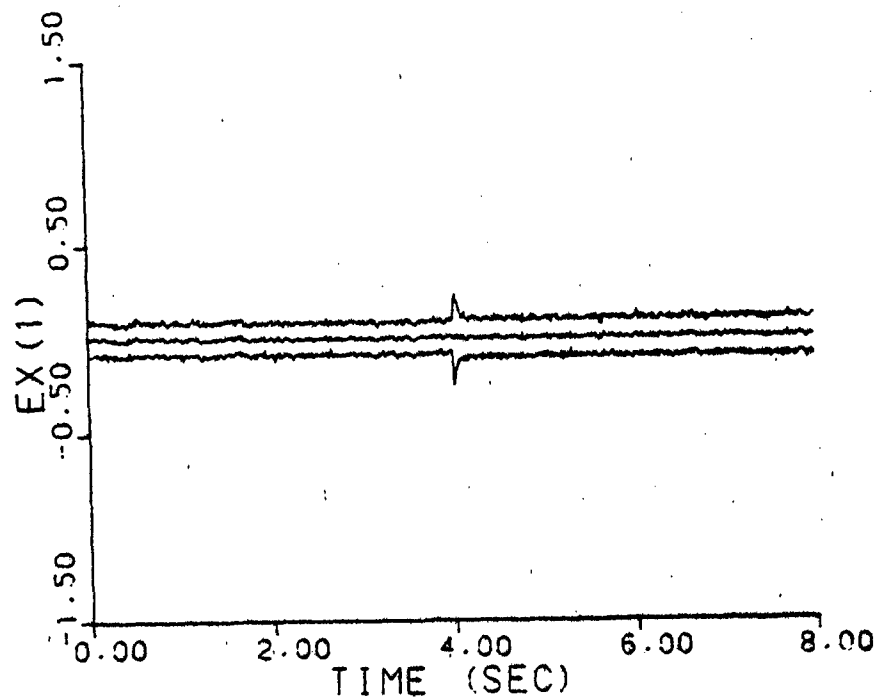


Figure A-5h. Parameter Position Estimate Monitoring
a jumps: (0.07, 9.0) - (0.93, 41.0)
No dither

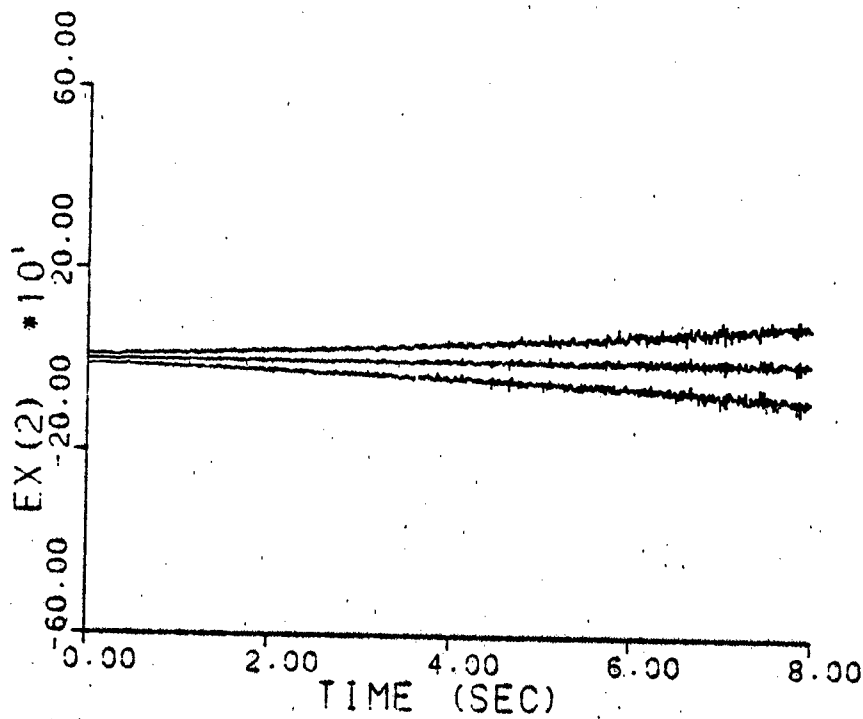
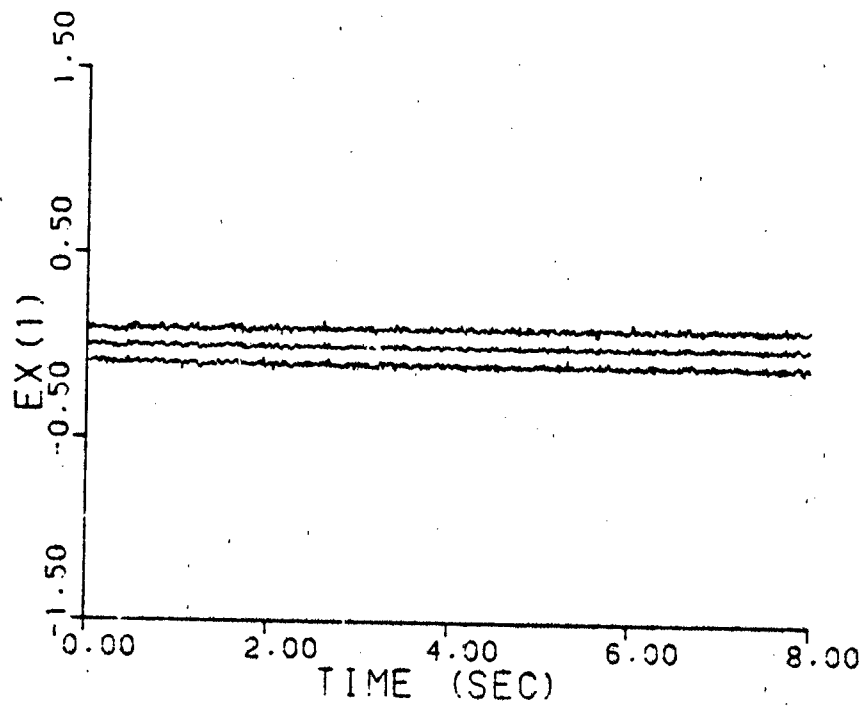


Figure A-51. Parameter Position Estimate Monitoring
 a varies: (0.07, 9.0) - (0.93, 41.0)
 no dither

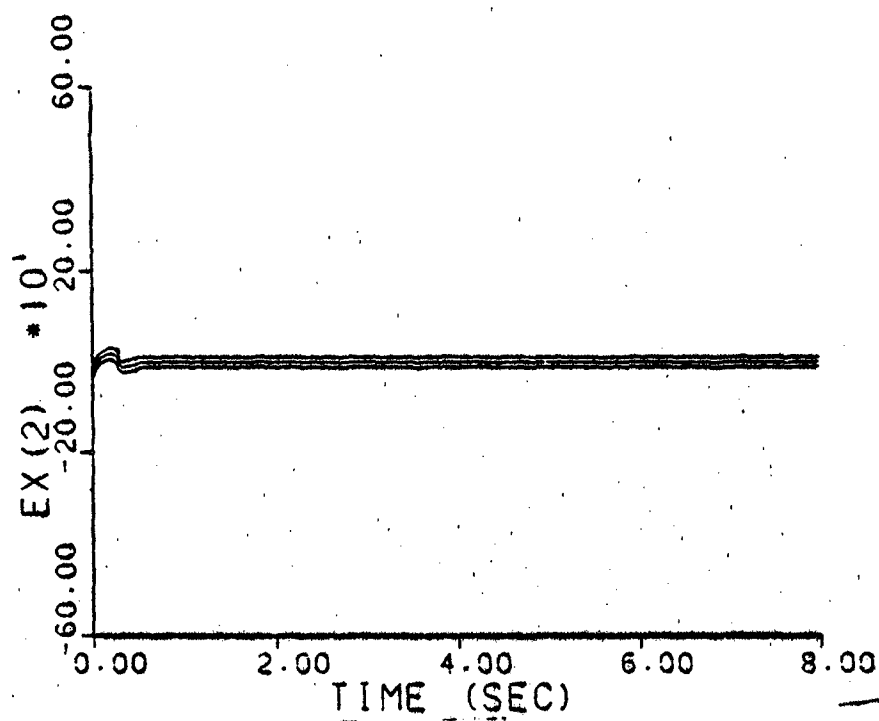
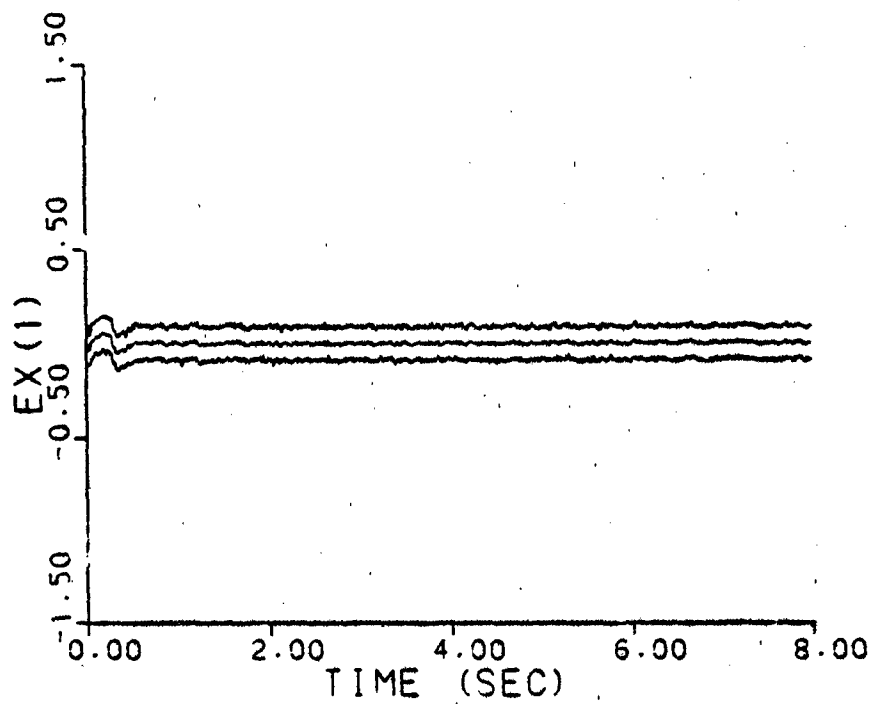


Figure A-6a. Parameter Position Estimate Monitoring
 $\underline{a} = (1, 3)$

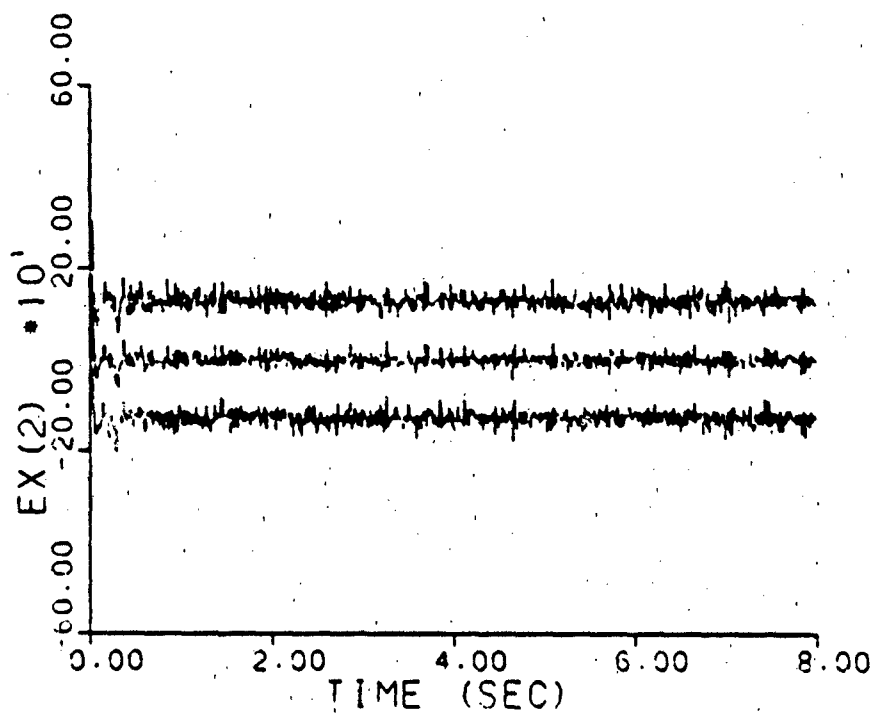
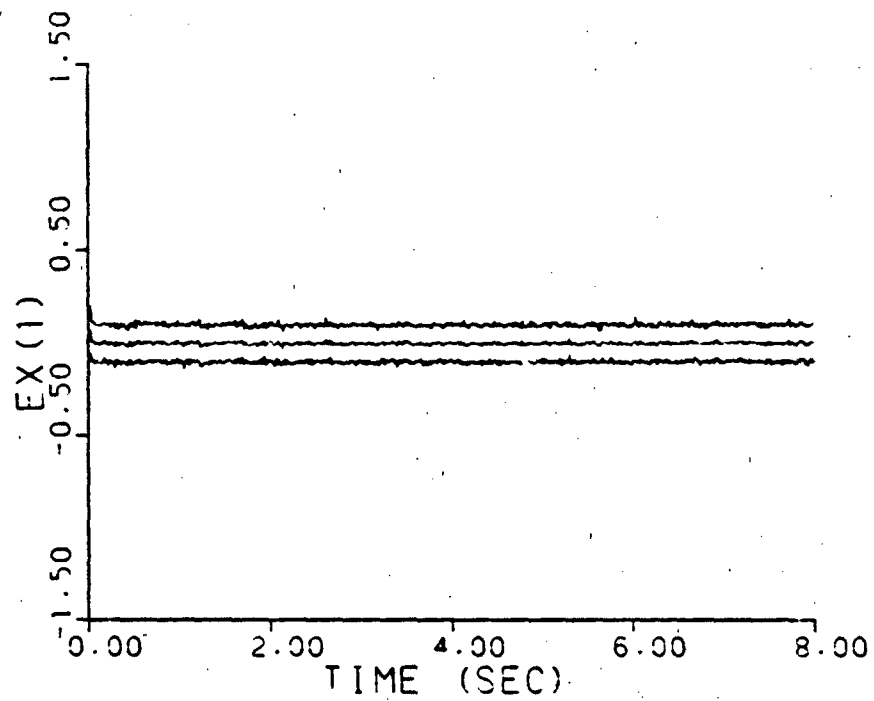


Figure A-6b. Parameter Position Estimate Monitoring
 $\underline{a} = (2, 9)$

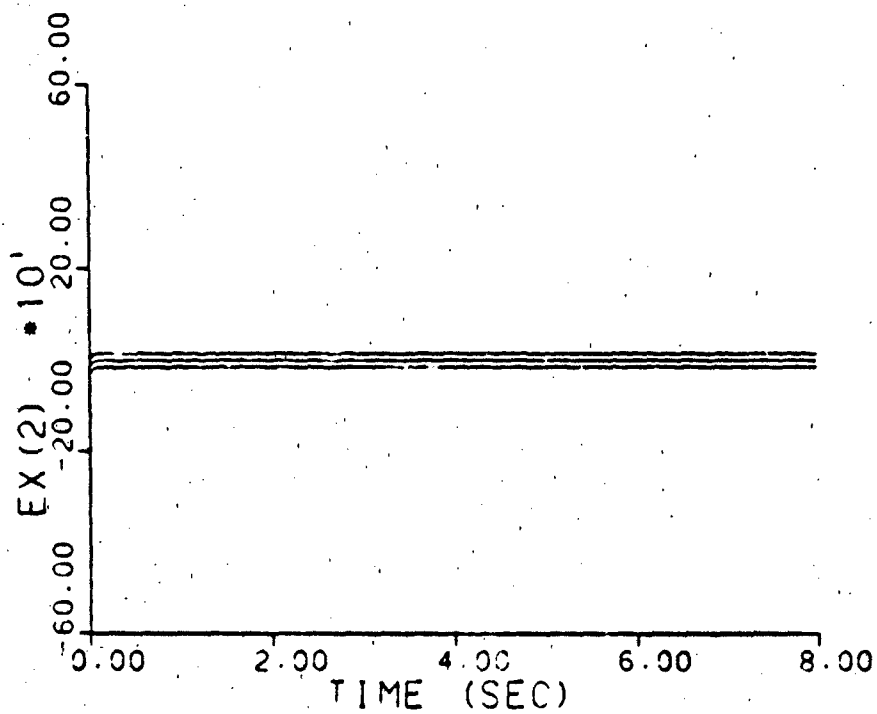
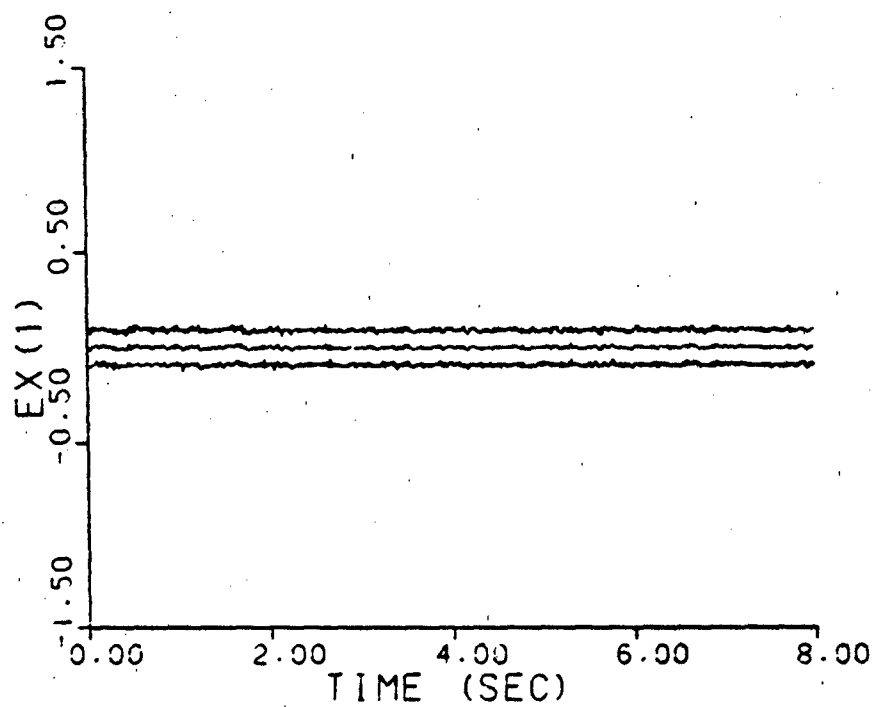


Figure A-6c. Parameter Position Estimate Monitoring
 $\underline{a} = (5, 4)$

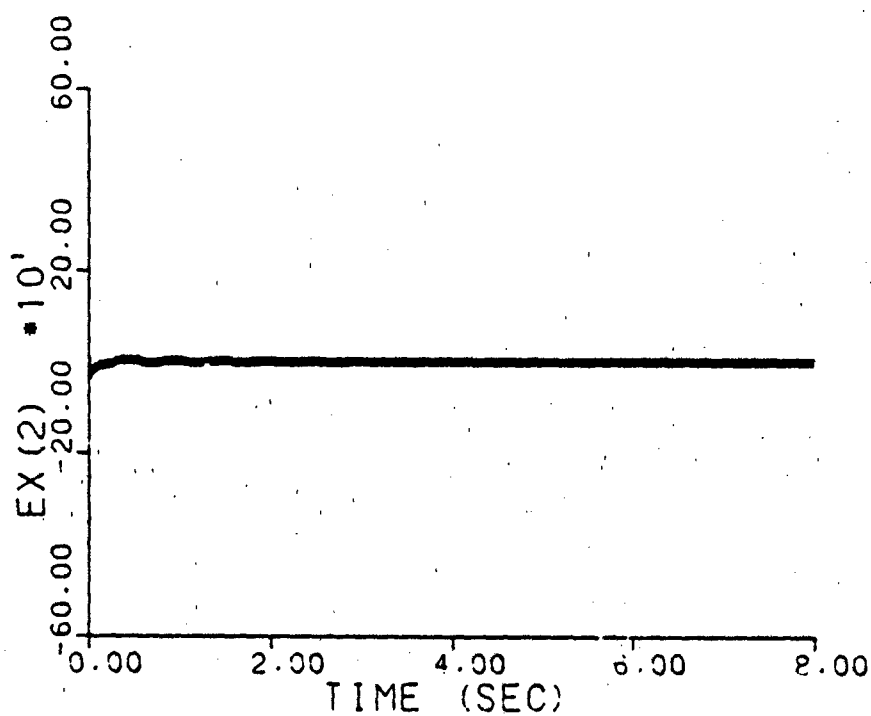
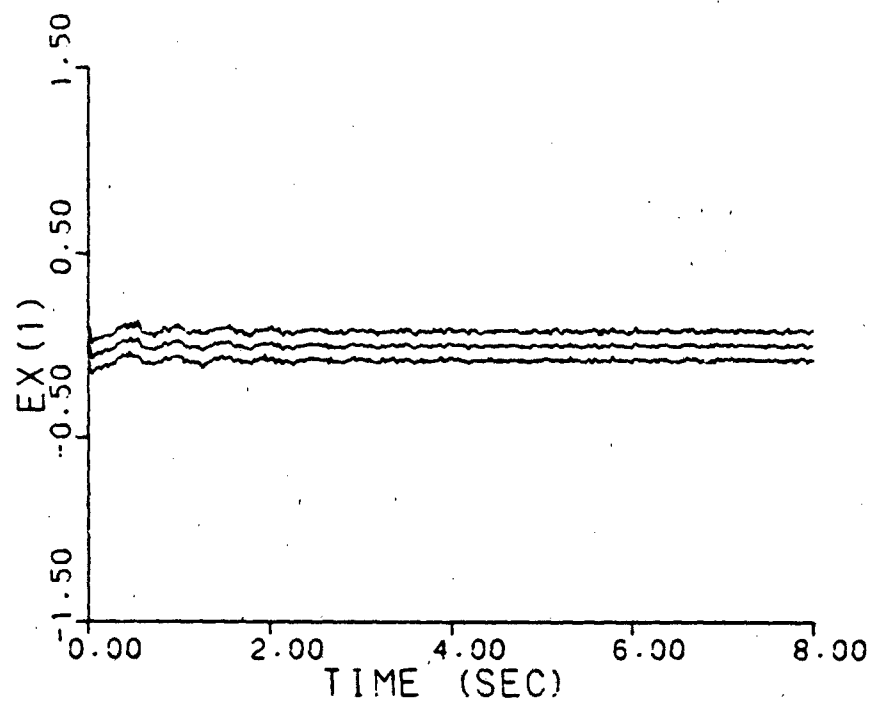


Figure A-6d. Parameter Position Estimate Monitoring
 $\underline{a} = (9, 2)$

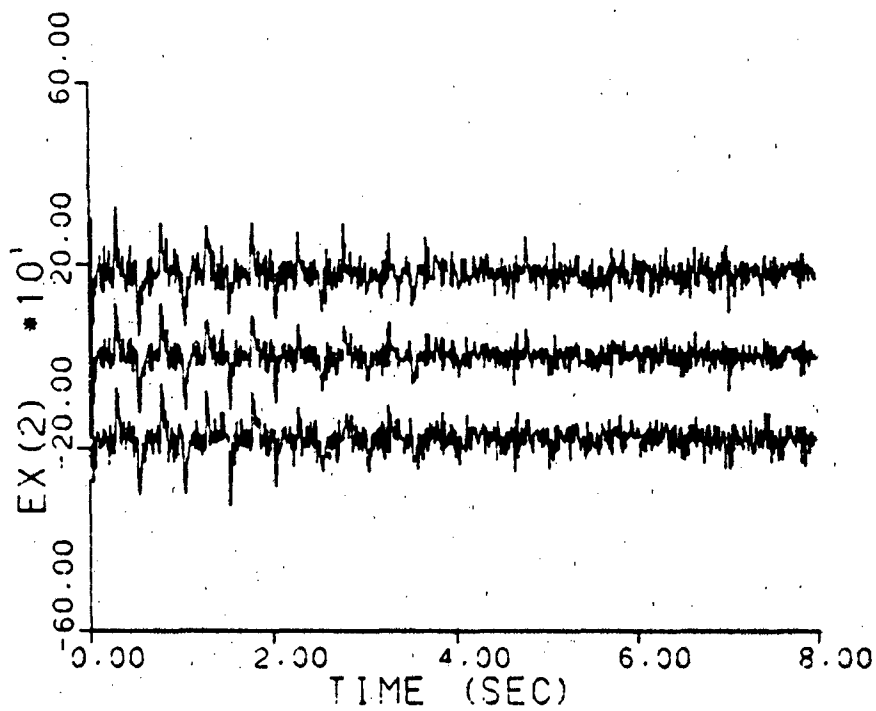
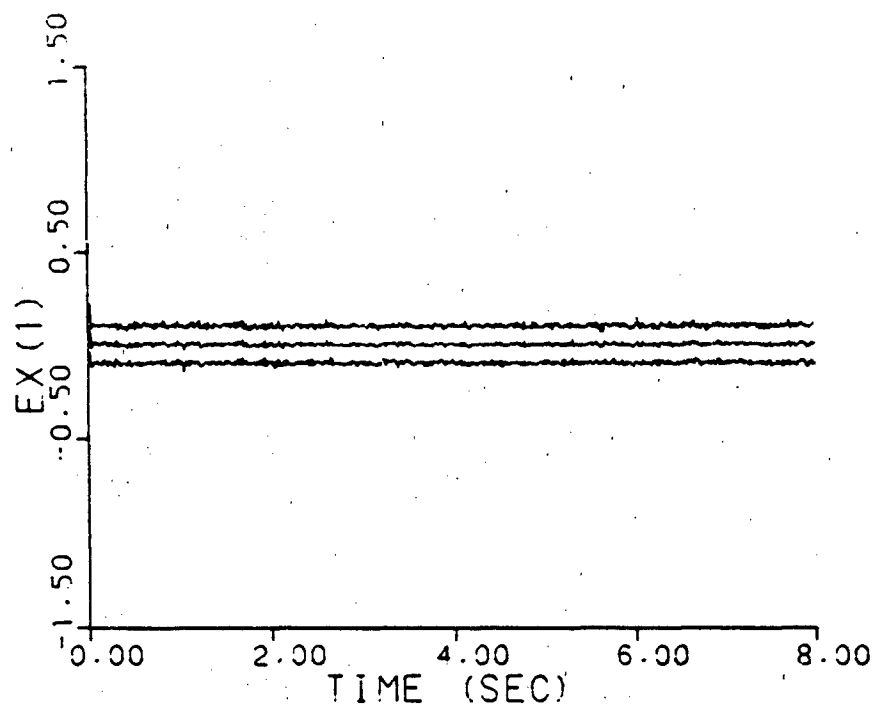


Figure A-6e. Parameter Position Estimate Monitoring
 $\underline{a} = (10, 10)$

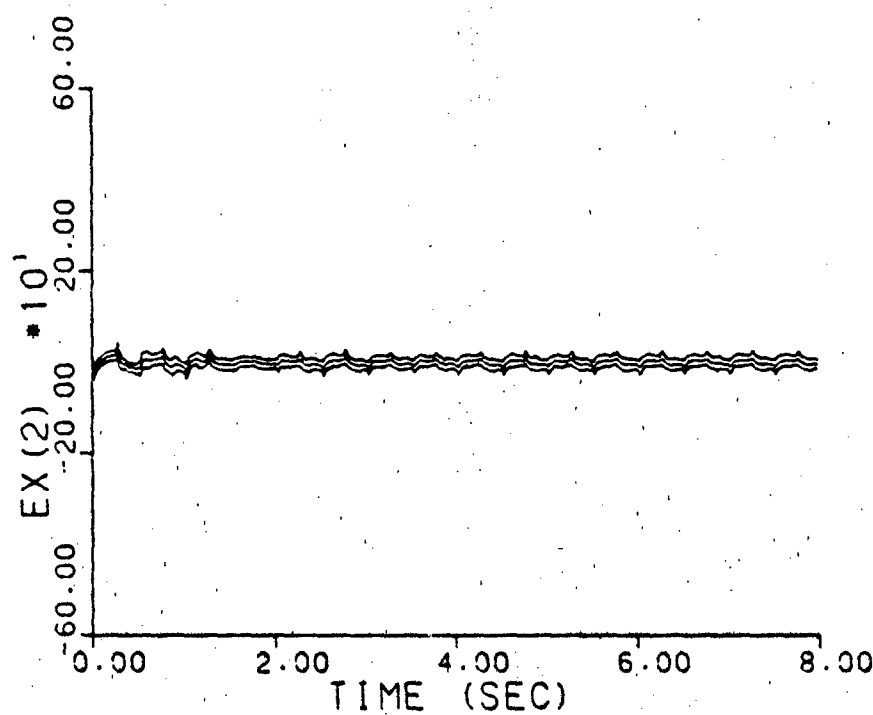
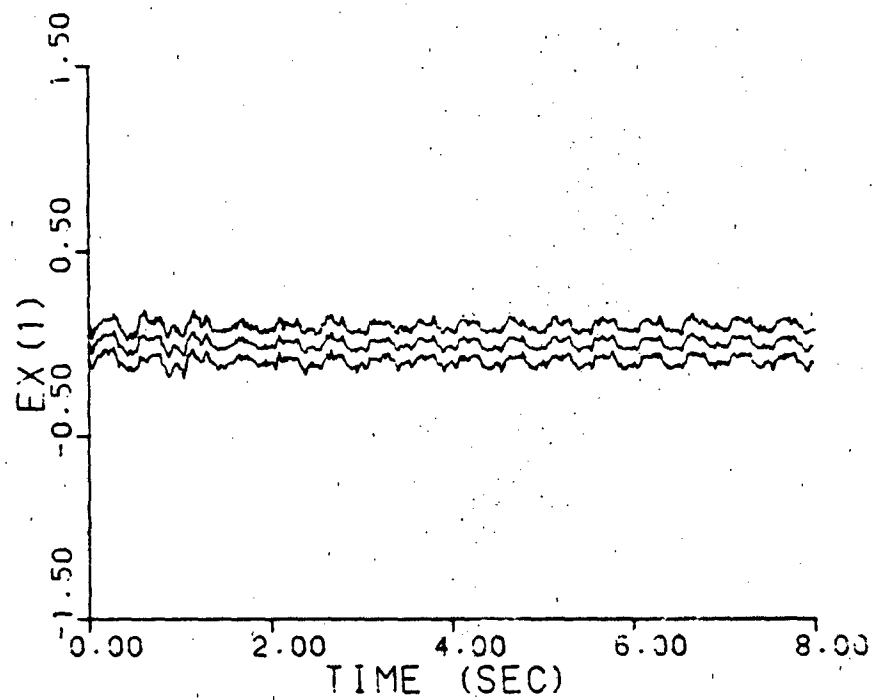


Figure A-6f. Parameter Position Estimate Monitoring
 $\underline{a} = (0.07, 9.0)$

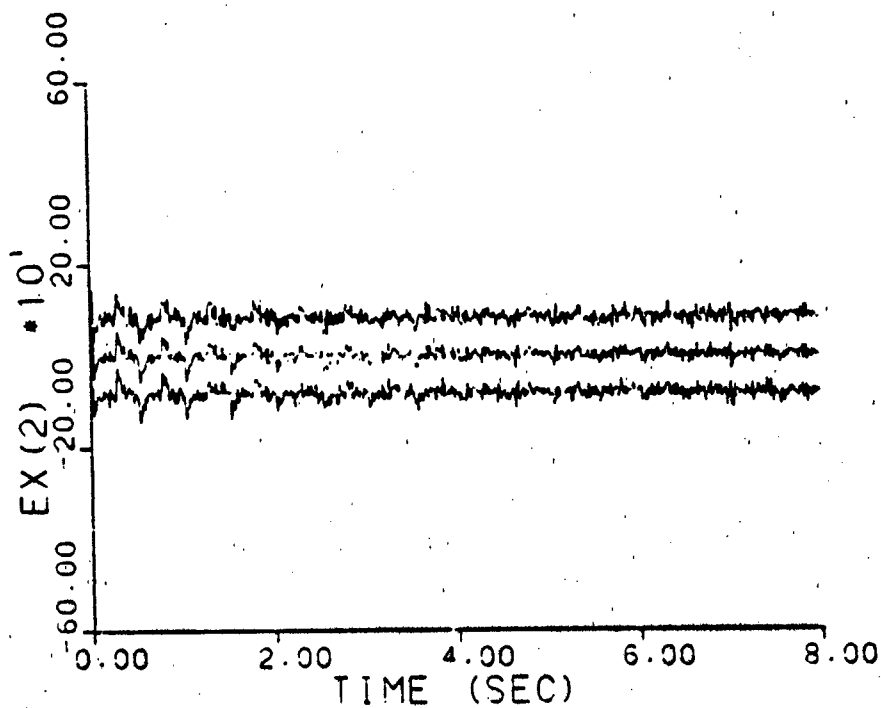
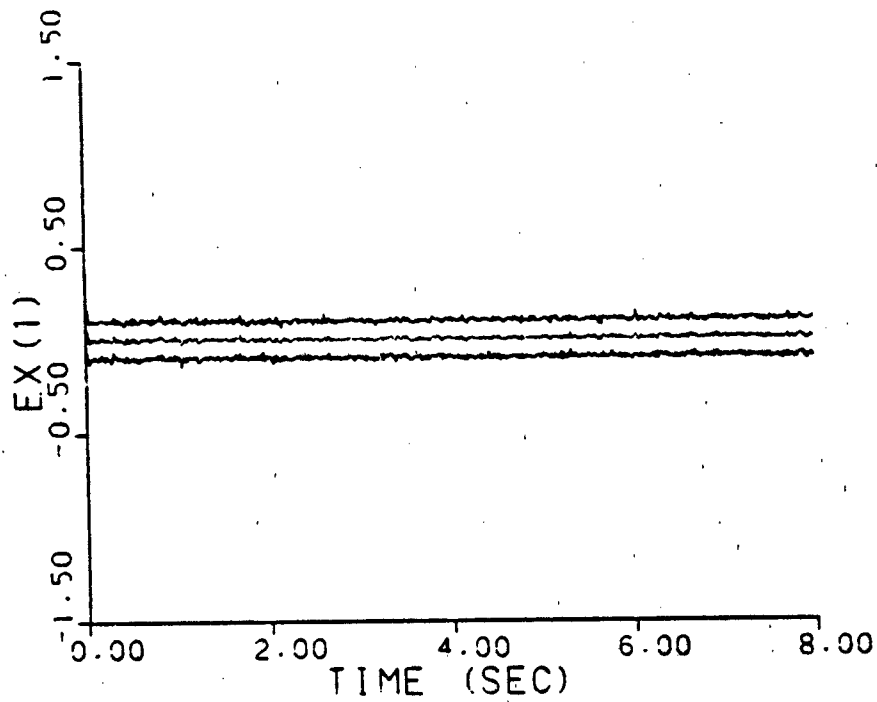


Figure A-6g. Parameter Position Estimate Monitoring
 $\underline{a} = (0.93, 41.0)$

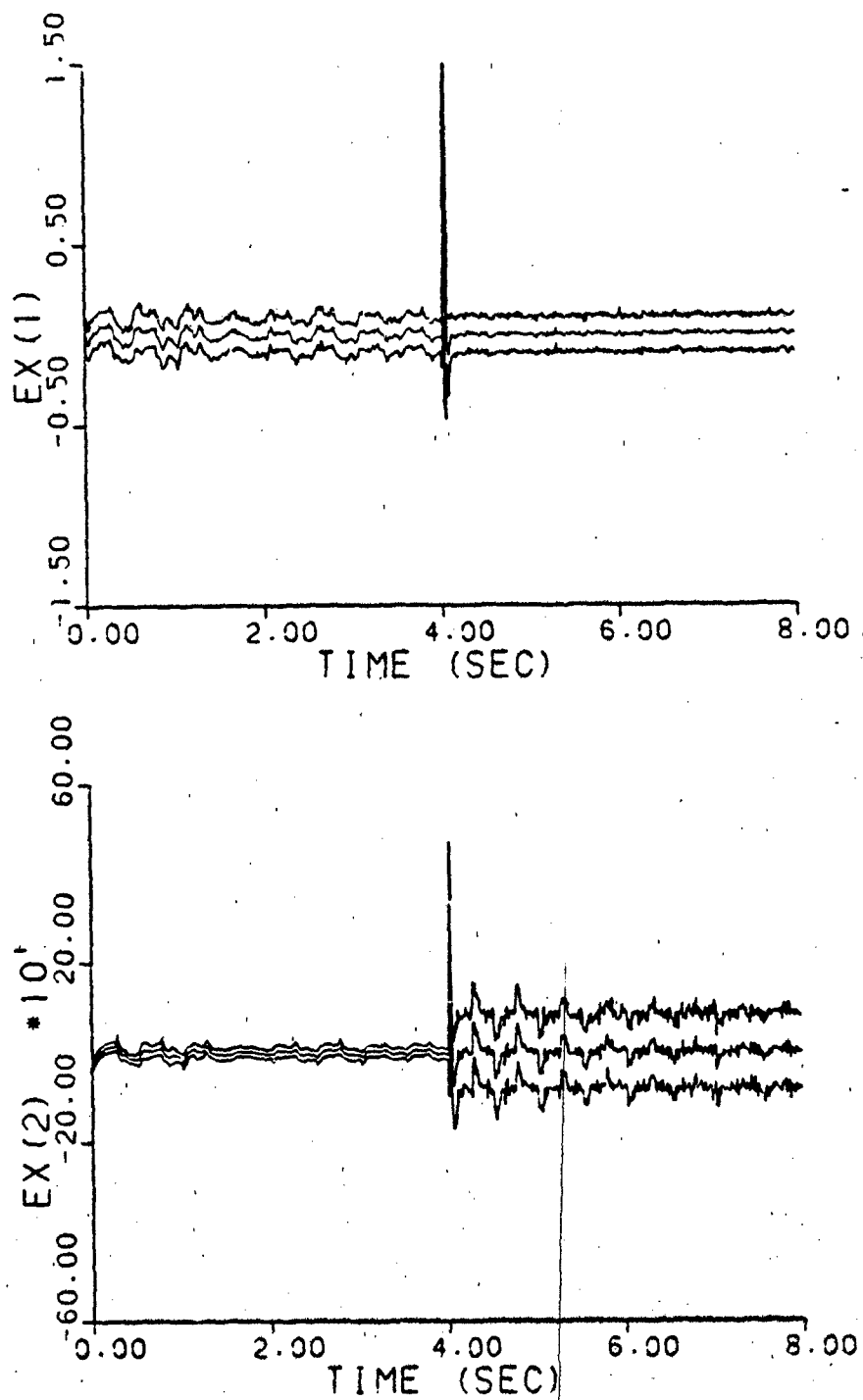


Figure A-6h. Parameter Position Estimate Monitoring
a jumps: (0.07, 9.0) - (0.93, 41.0)

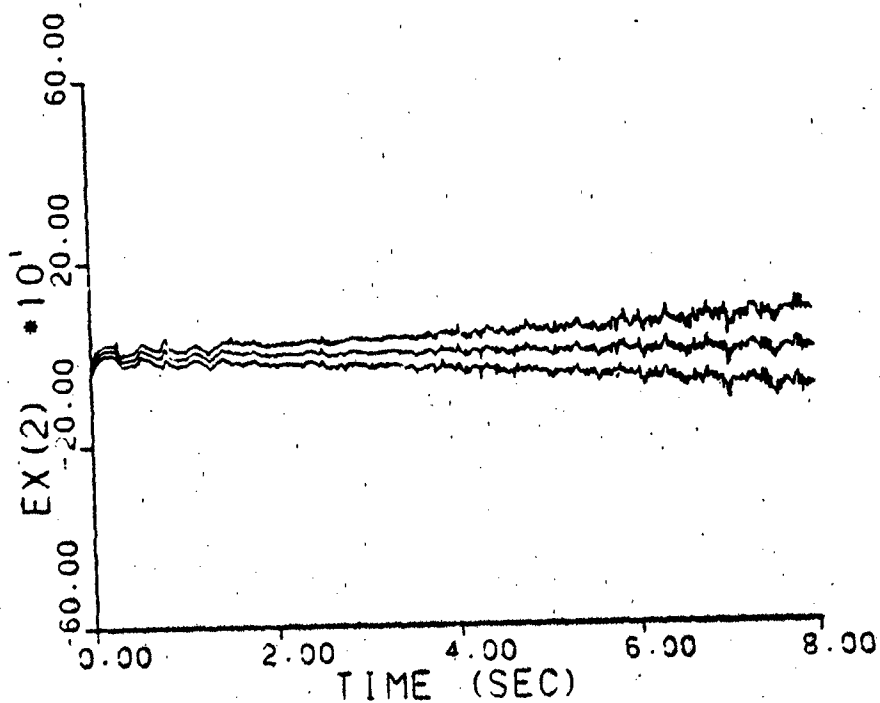
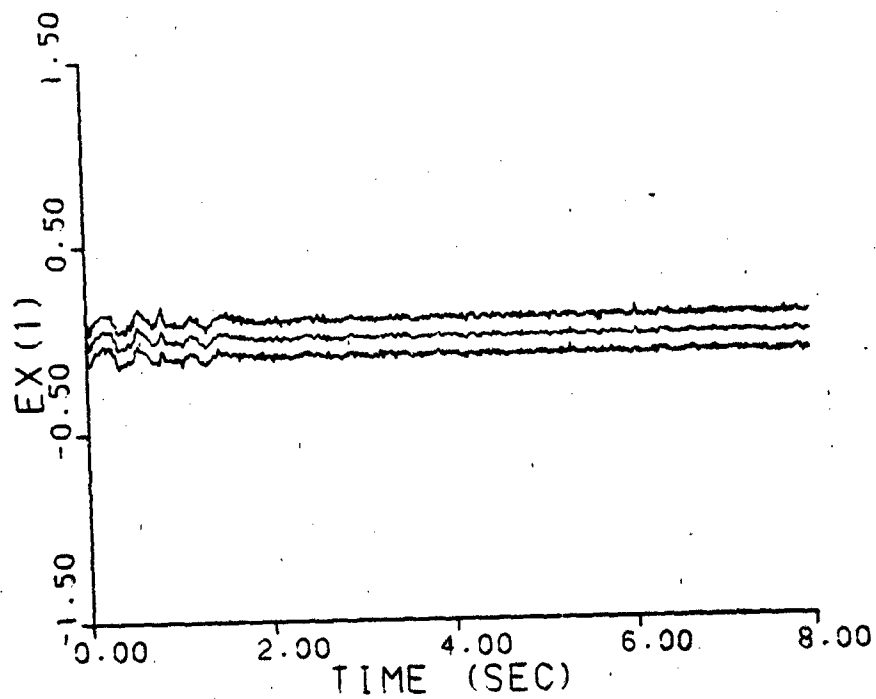


Figure A-61. Parameter Position Estimate Monitoring
a varies: (0.07, 9.0) - (.)93, 41.0)

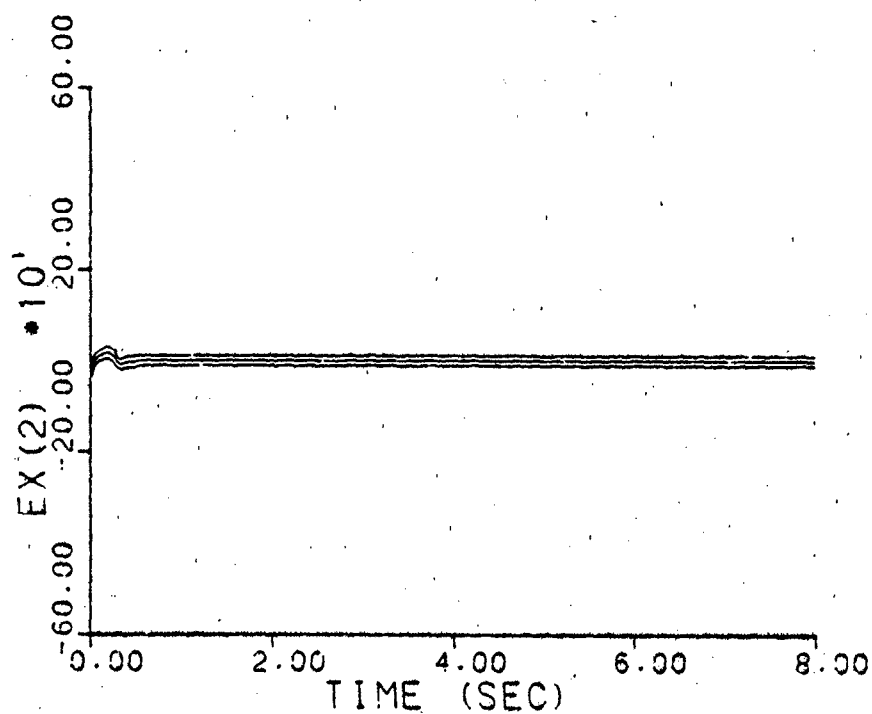
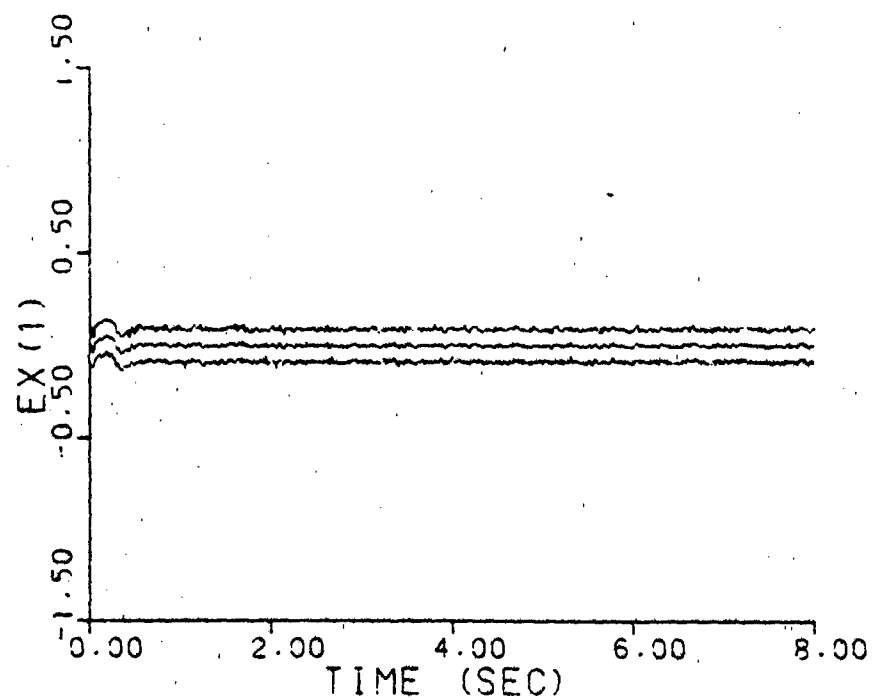


Figure A-7a. Parameter Position Estimate Monitoring
 $\underline{a} = (1, 3)$, warm-up

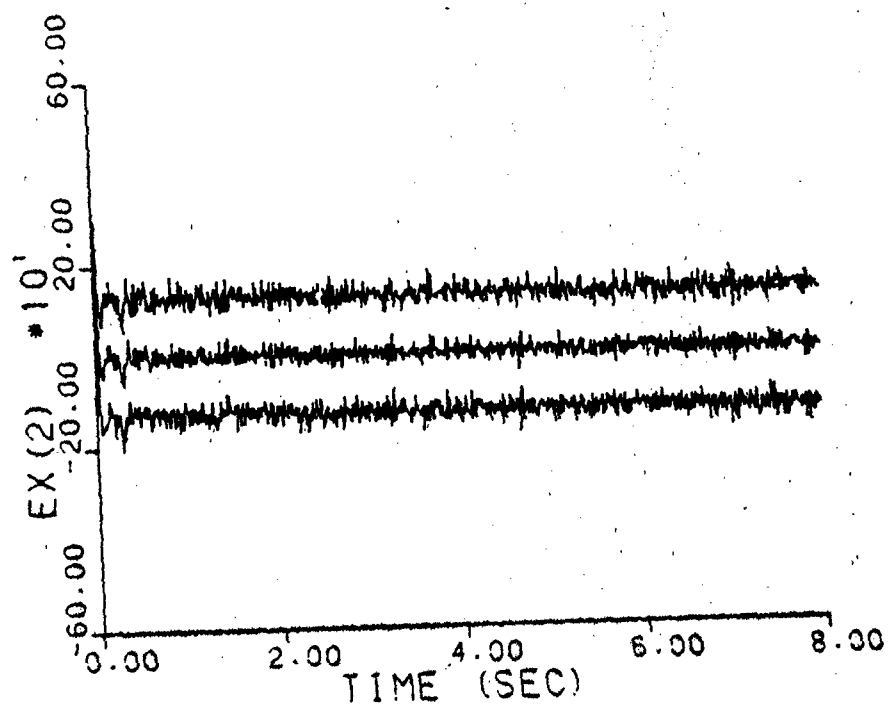
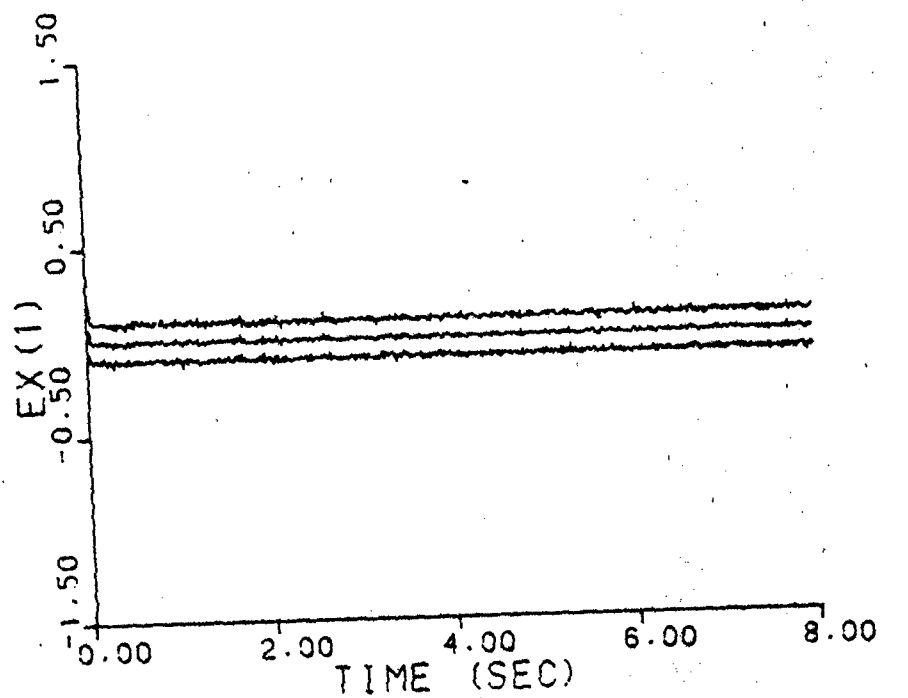


Figure A-7b. Parameter Position Estimate Monitoring
 $\underline{a} = (2, 9)$, warm-up

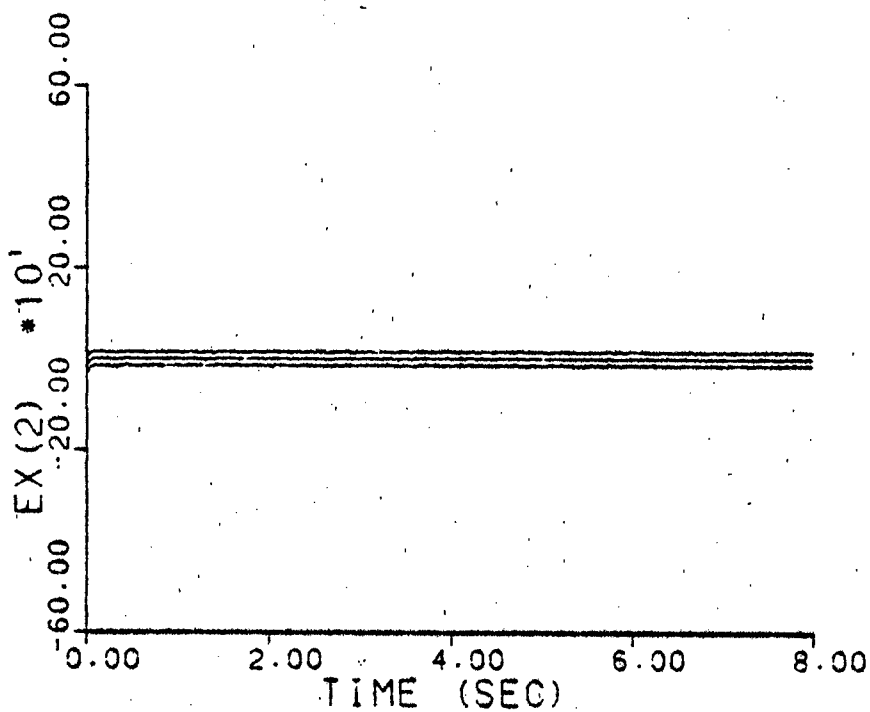
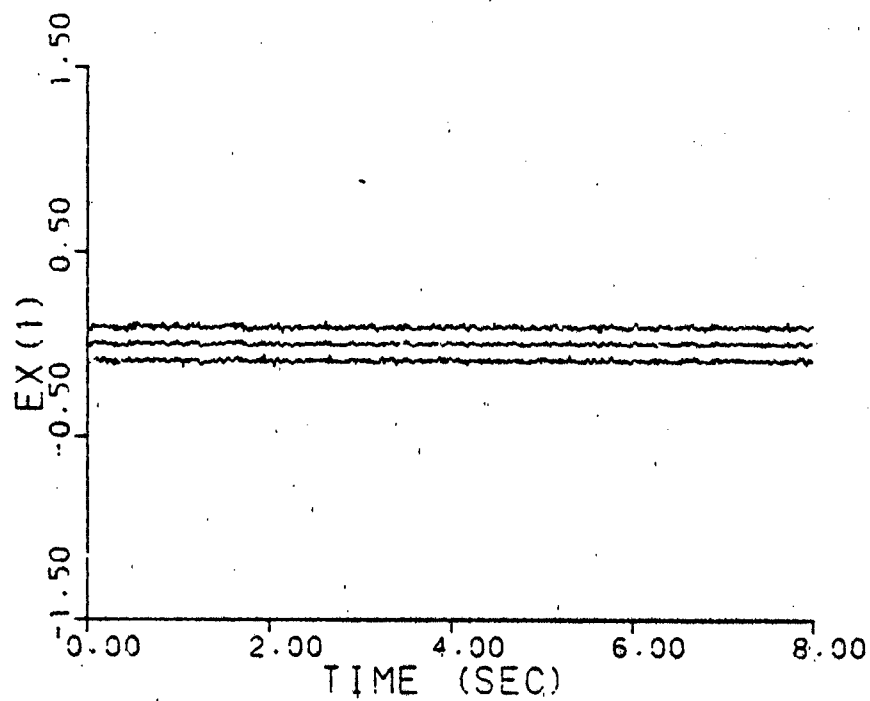


Figure A-7c. Parameter Position Estimate Monitoring
 $\underline{a} = (5, 4)$, warm-up

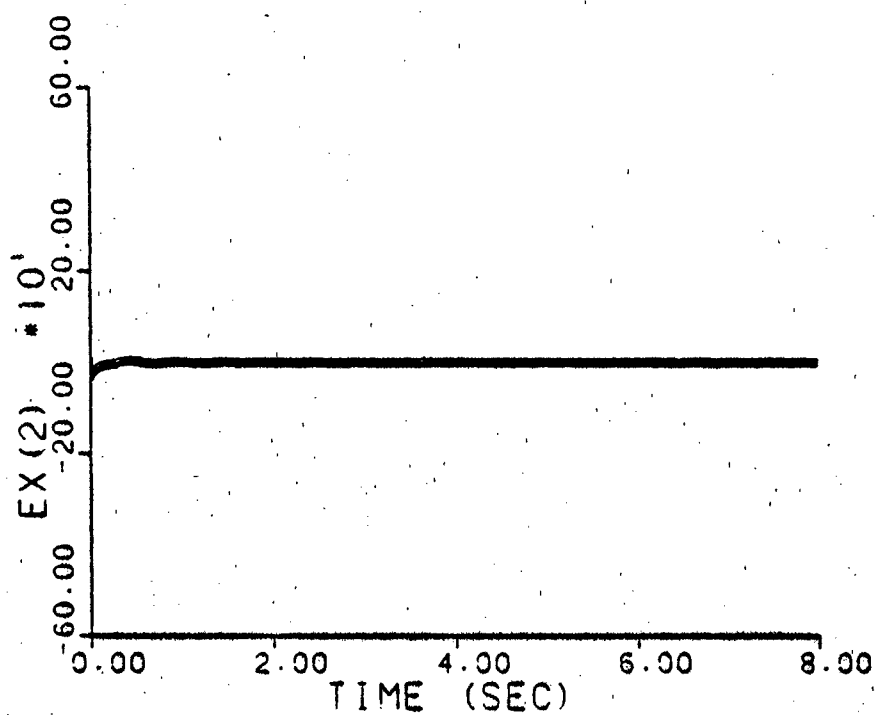
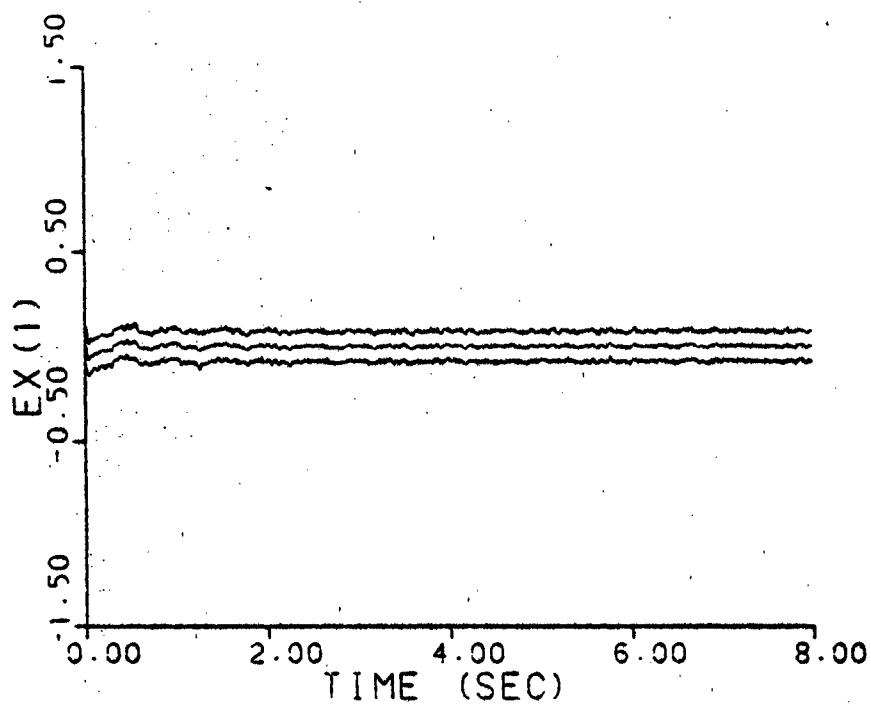


Figure A-7d. Parameter Position Estimate Monitoring
 $\underline{a} = (9, 2)$, warm-up

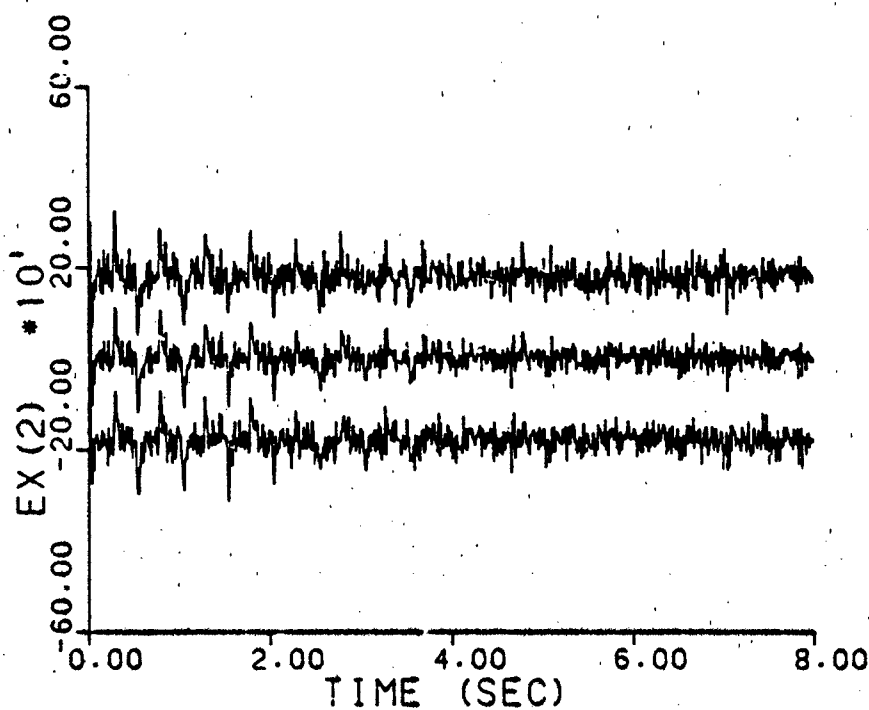
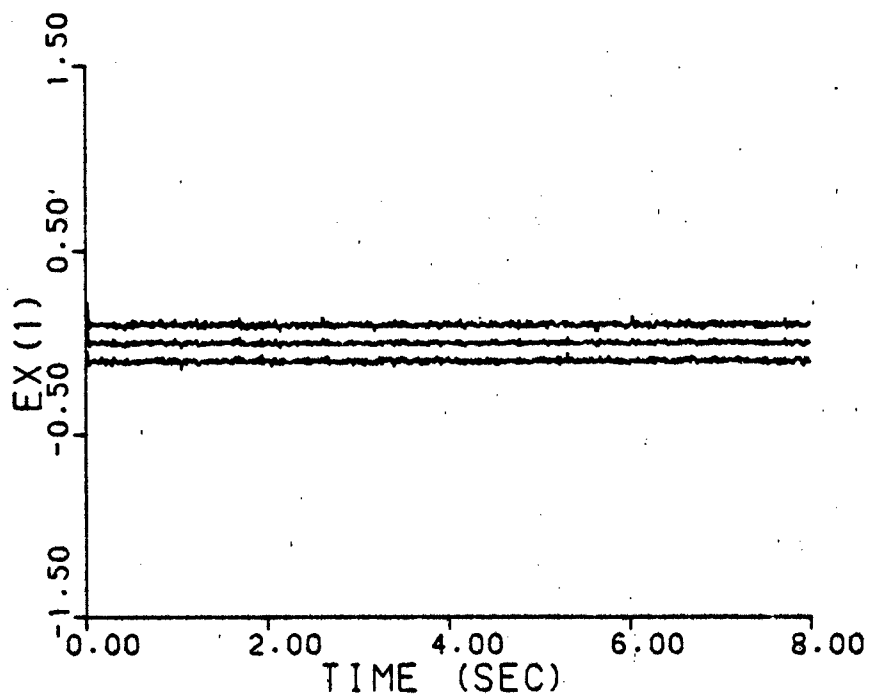


Figure A-7e. Parameter Position Estimate Monitoring
 $\underline{a} = (10, 10)$, warm-up

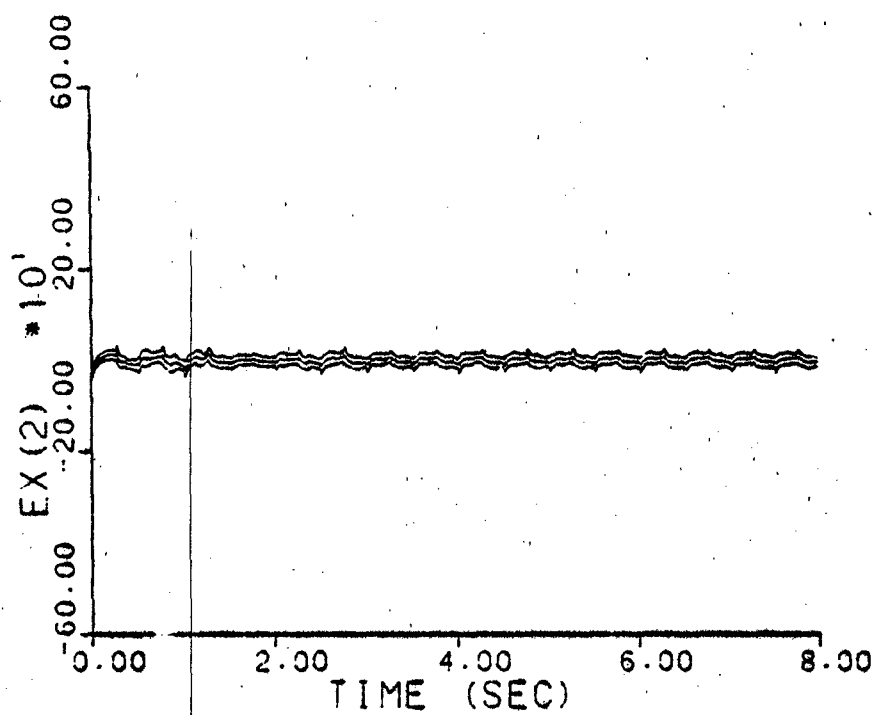
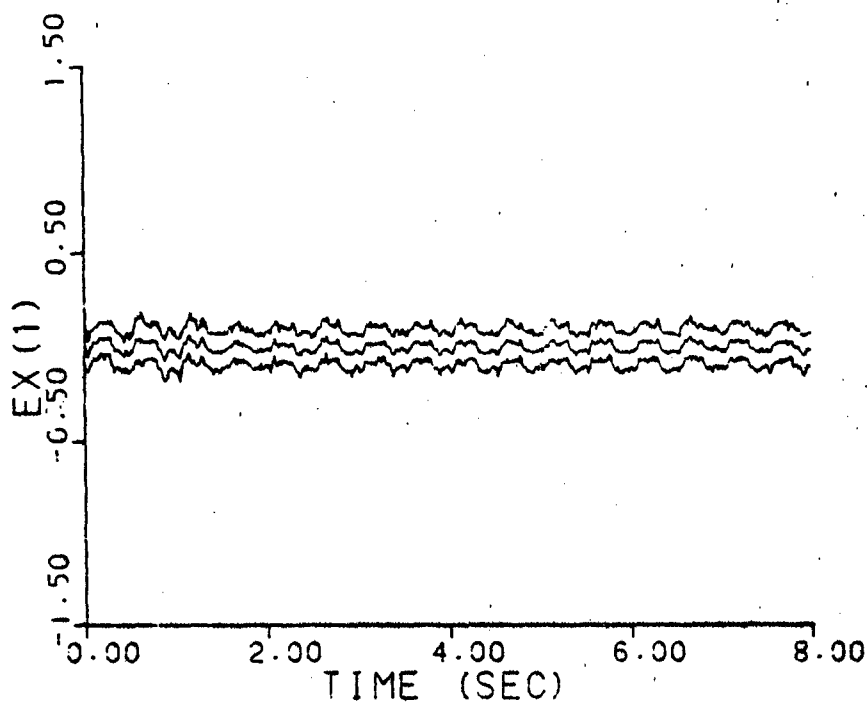


Figure A-7f. Parameter Position Estimate Monitoring
 $\underline{a} = (0.07, 9.0)$, warm-up

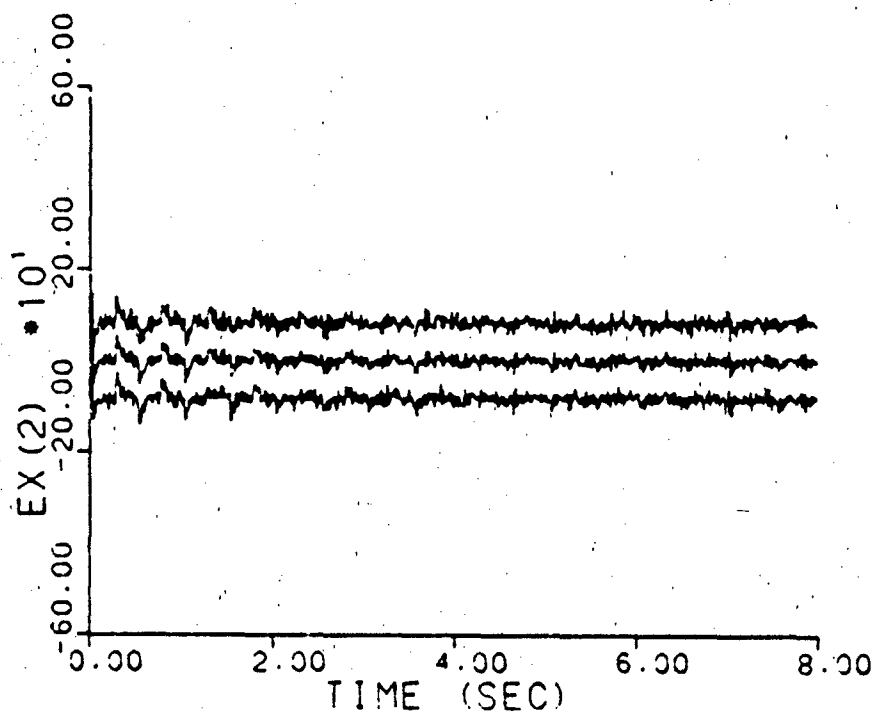
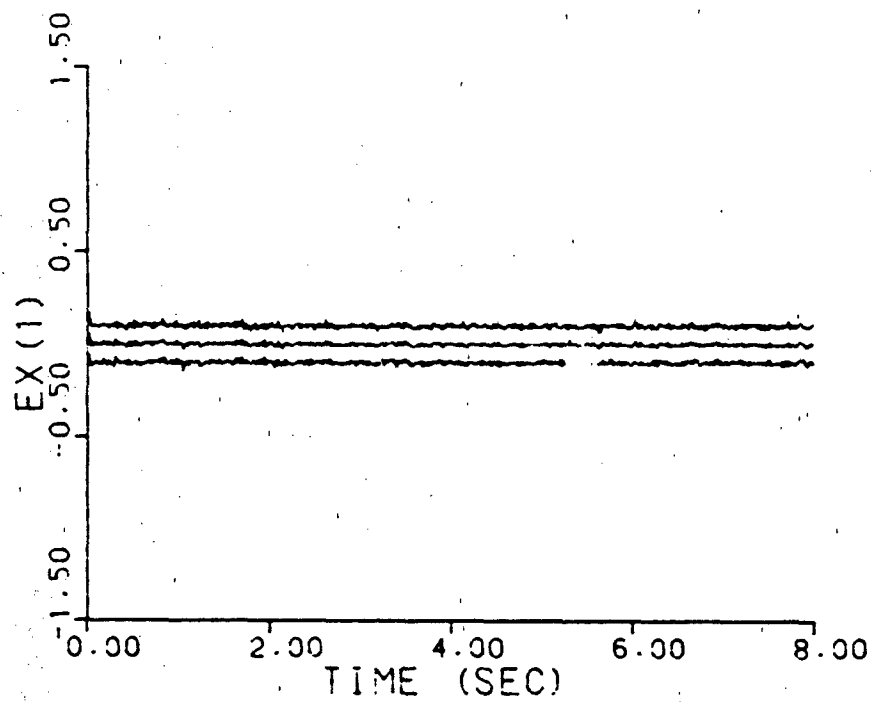


Figure A-7g. Parameter Position Estimate Monitoring
 $\underline{a} = (0.93, 41.0)$, warm-up

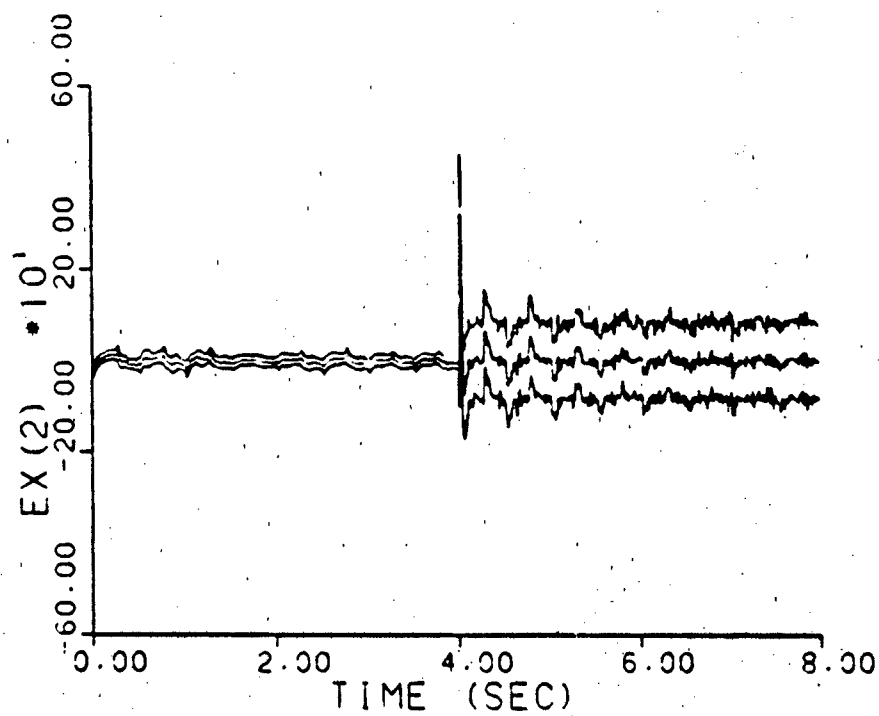
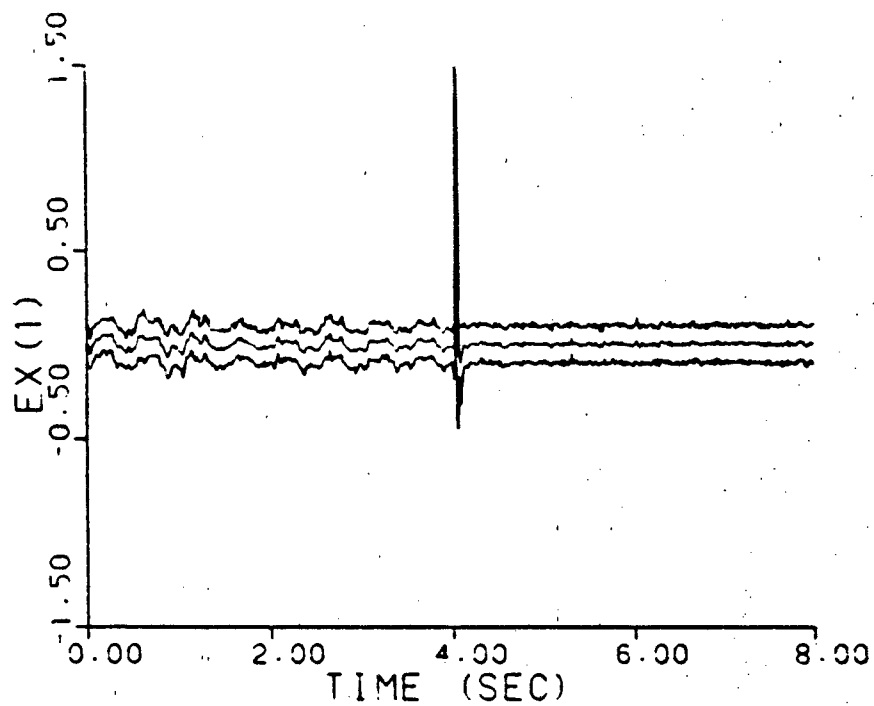


Figure A-7h. Parameter Position Estimate Monitoring
a jumps: (0.07, 9.0) - (0.93, 41.0)
warm-up

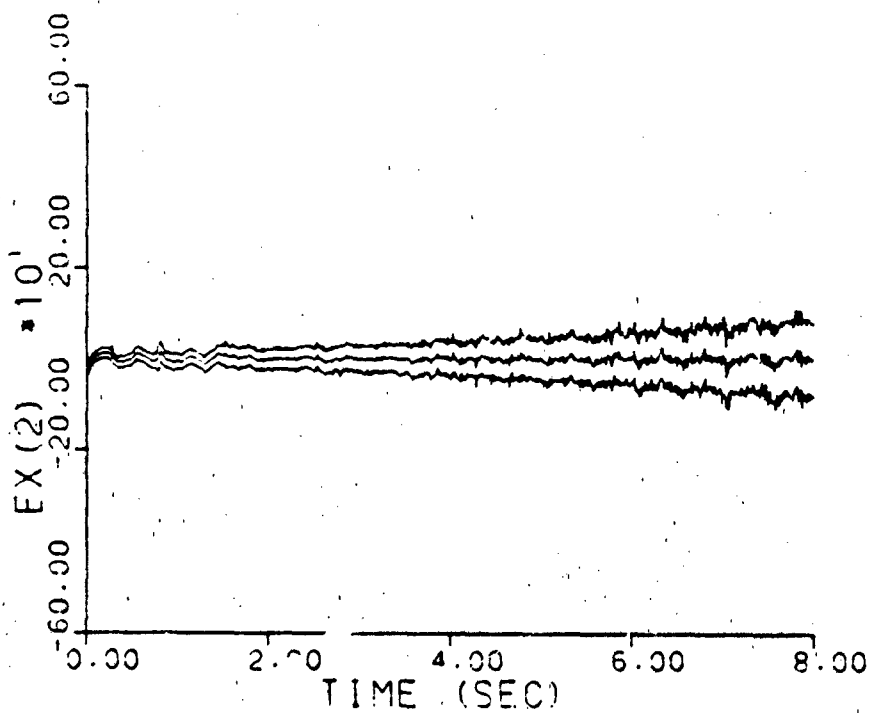
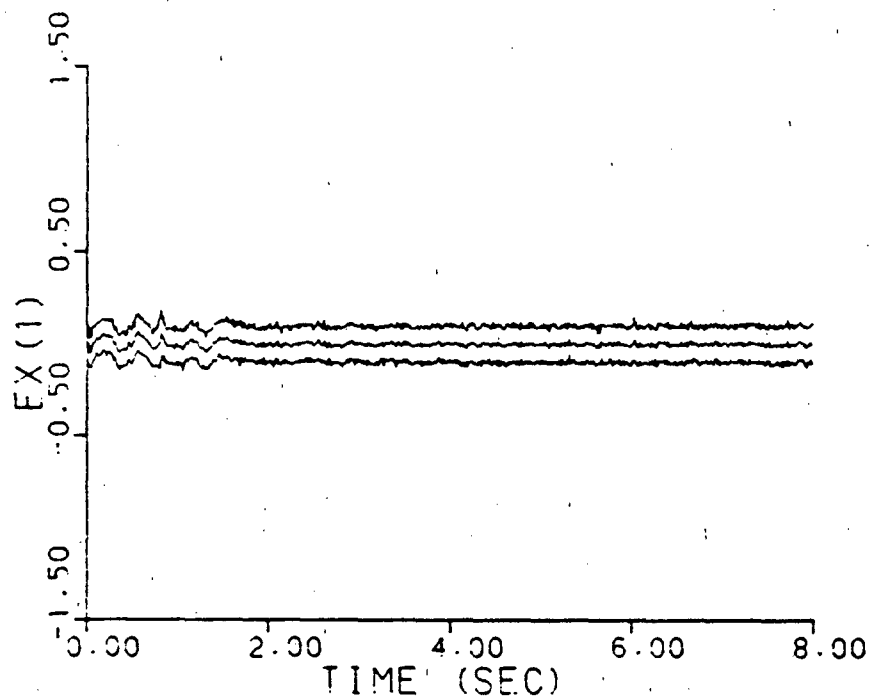


Figure A-7i. Parameter Position Estimate Monitoring
a varies: (0.07, 9.0) - (0.93, 41.0)
warm-up

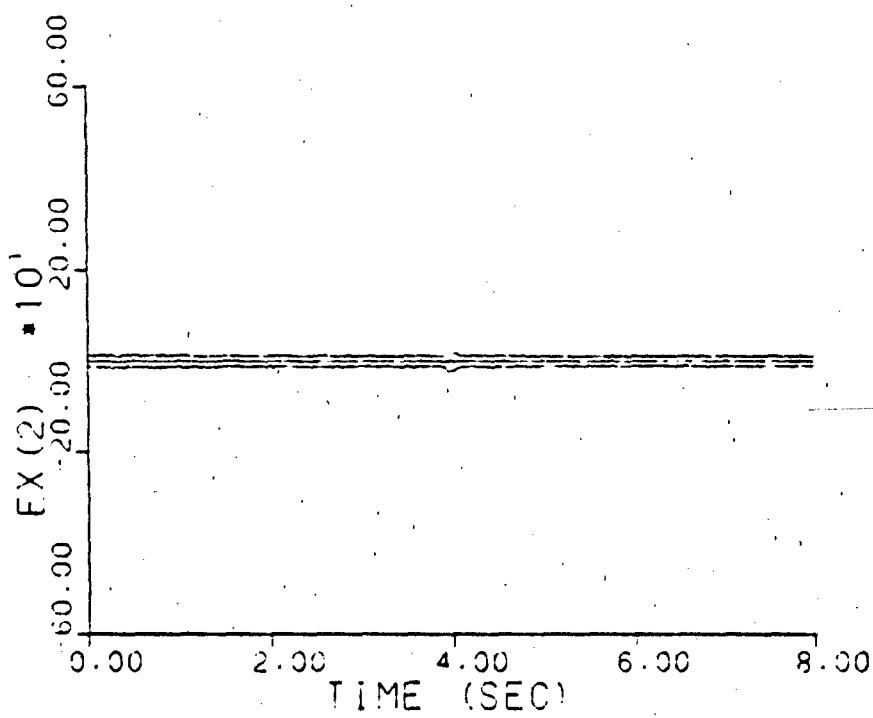
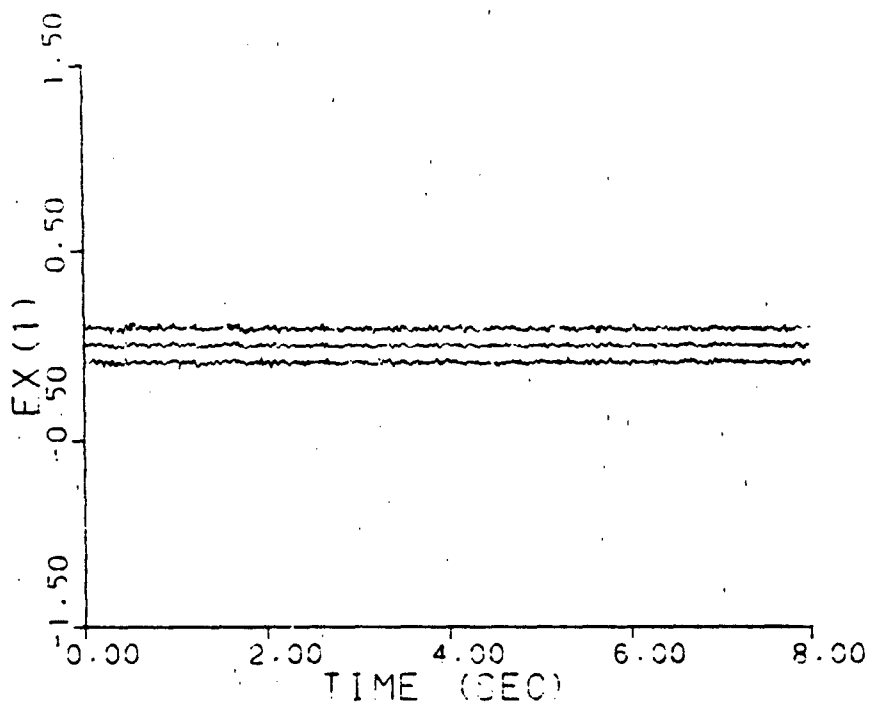


Figure A-8a. Parameter Position and Velocity
Estimate Monitoring
 $\underline{a} = (1,3)$, no dither

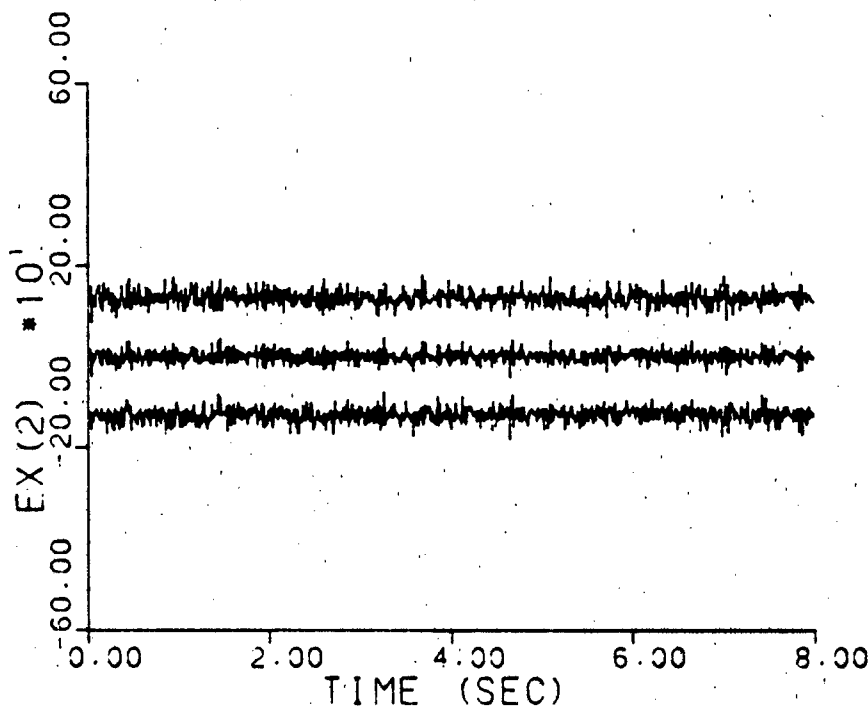
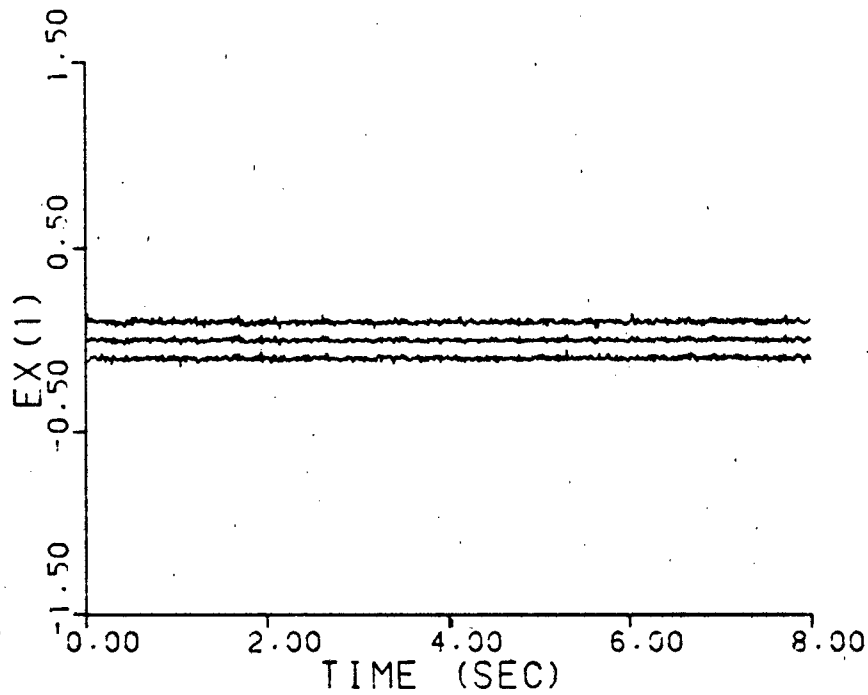


Figure A-8b. Parameter Position and Velocity
Estimate Monitoring
 $\underline{a} = (2, 9)$, no dither

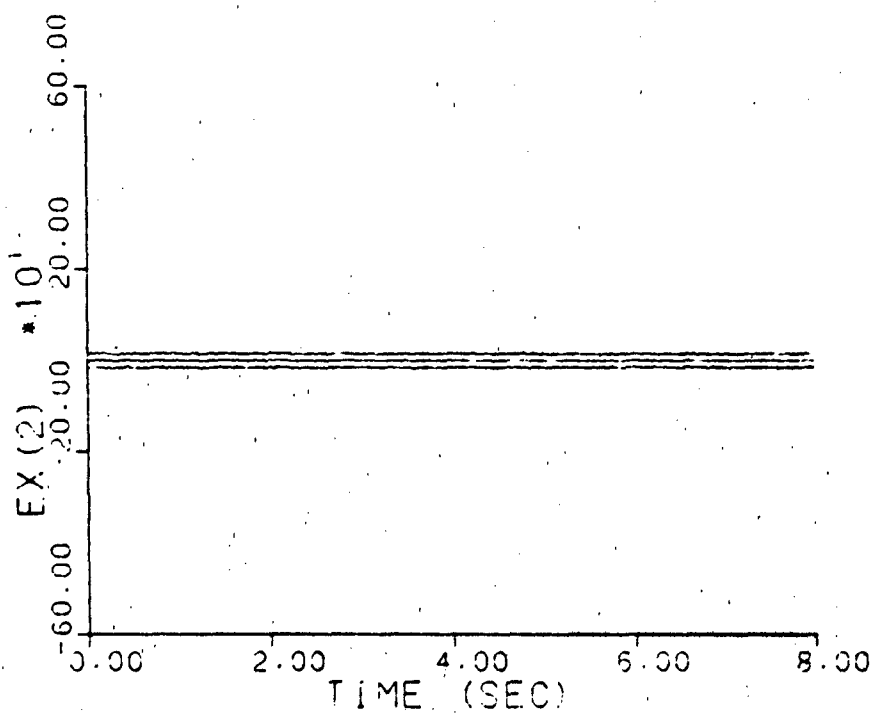
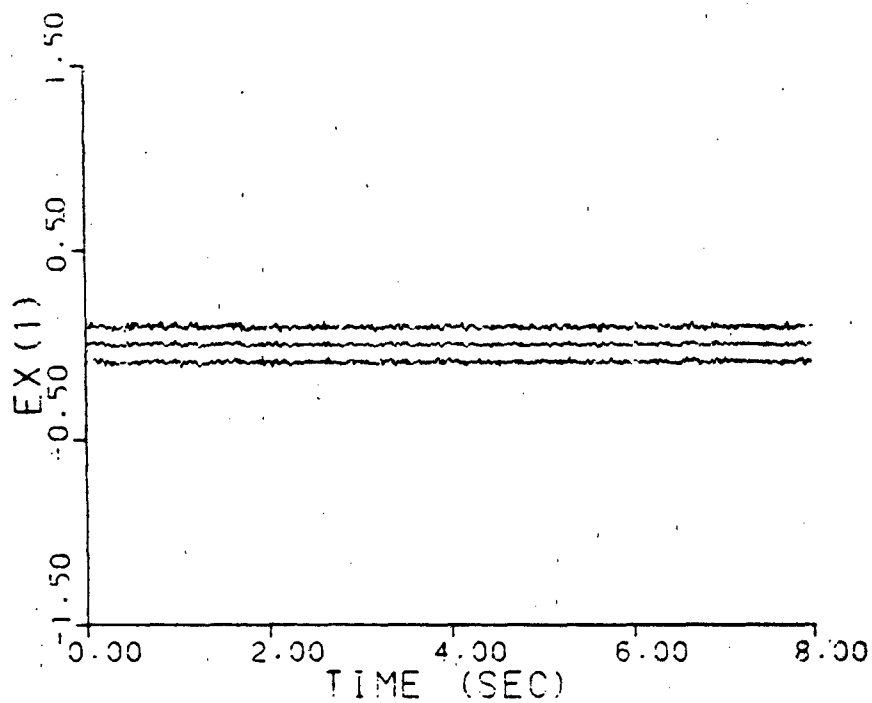


Figure A-8c. Parameter Position and Velocity
Estimate Monitoring
 $\underline{a} = (5, 4)$, no dither

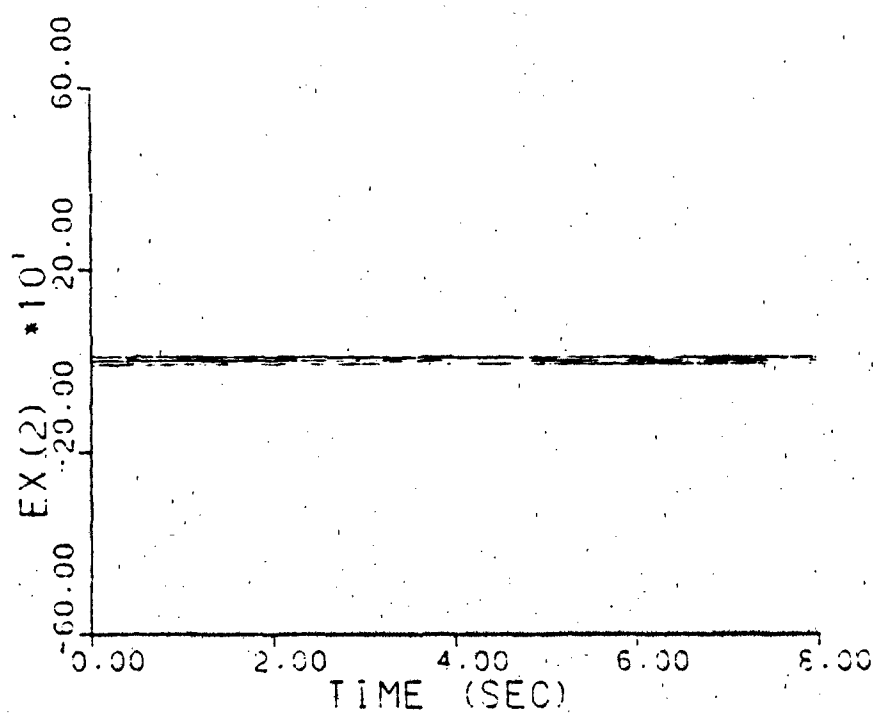
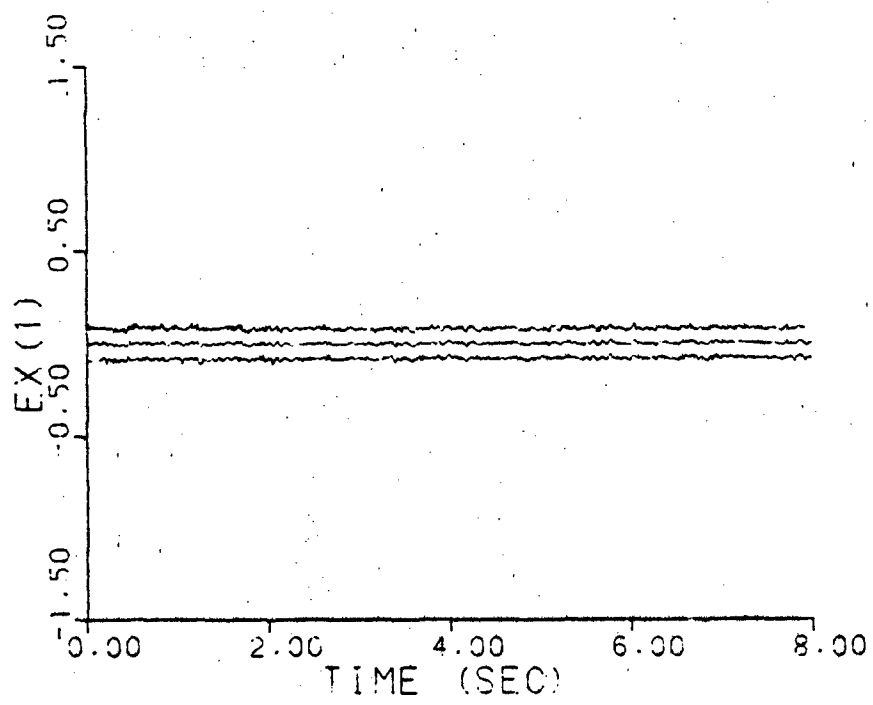


Figure A-8d. Parameter Position and Velocity
Estimate Monitoring
 $\underline{a} = (9, 2)$, no dither

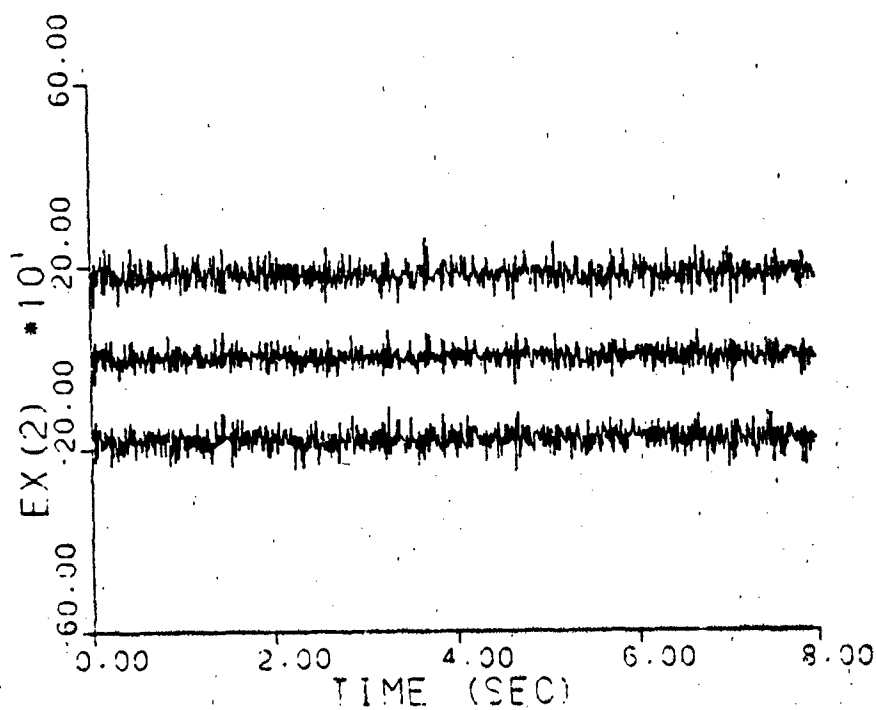
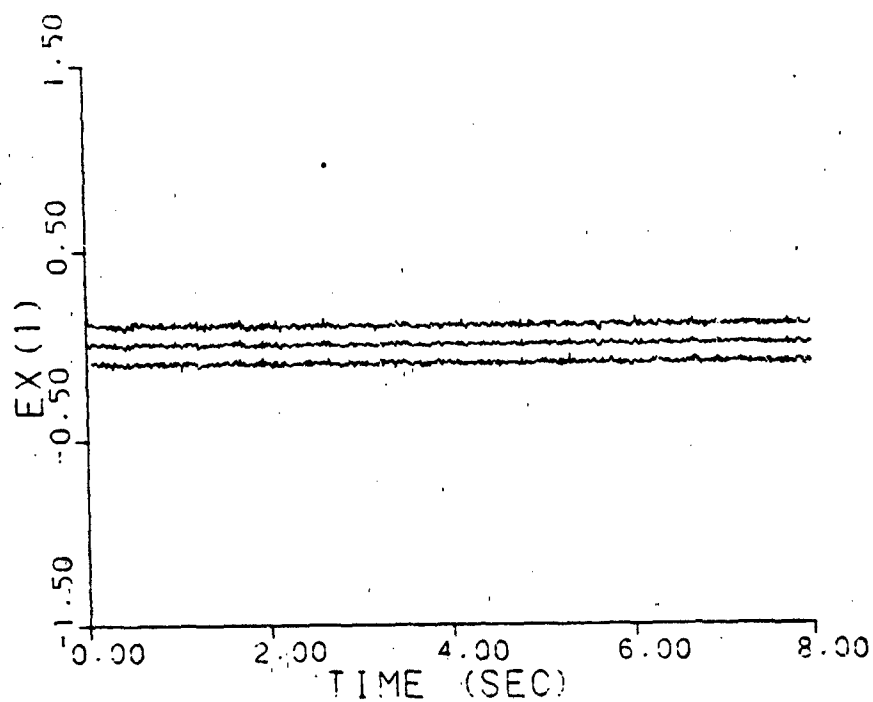


Figure A-8e. Parameter Position and Velocity
Estimate Monitoring
 $\underline{a} = (10, 10)$, no dither.

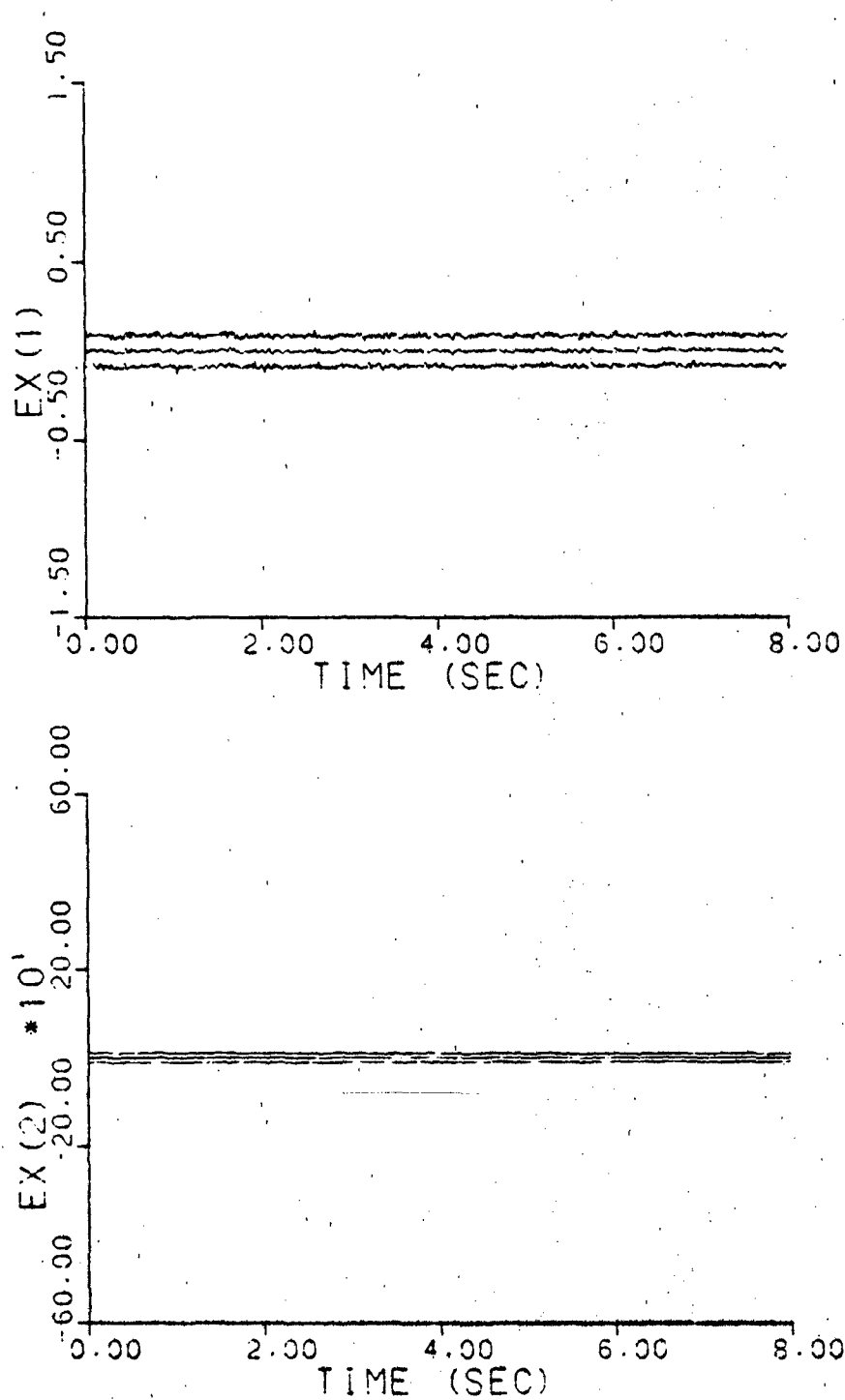


Figure A-8f. Parameter Position and Velocity
Estimate Monitoring
 $\underline{a} = (0.07, 9.0)$, no dither

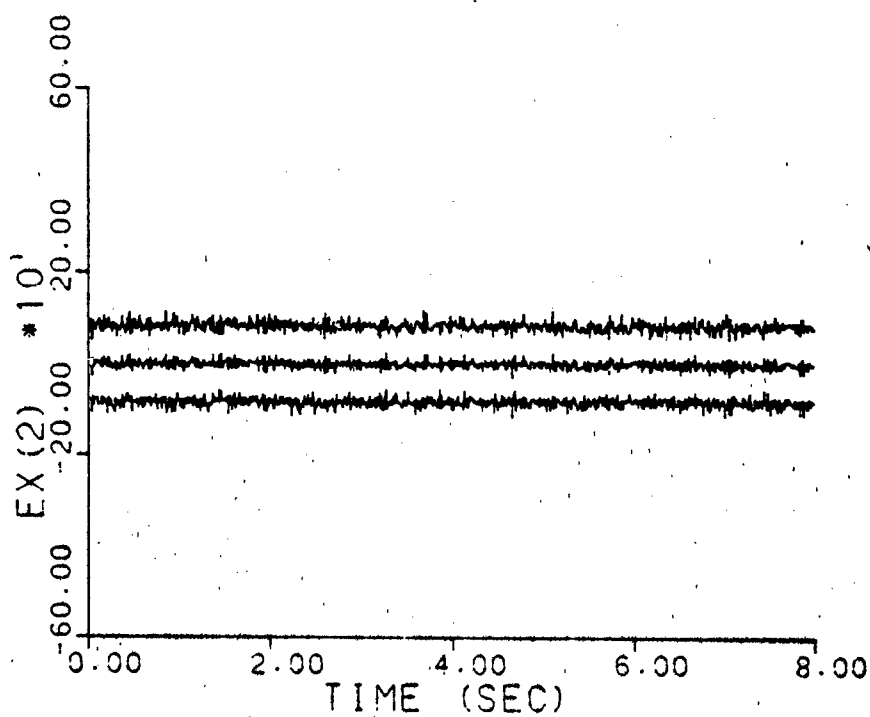
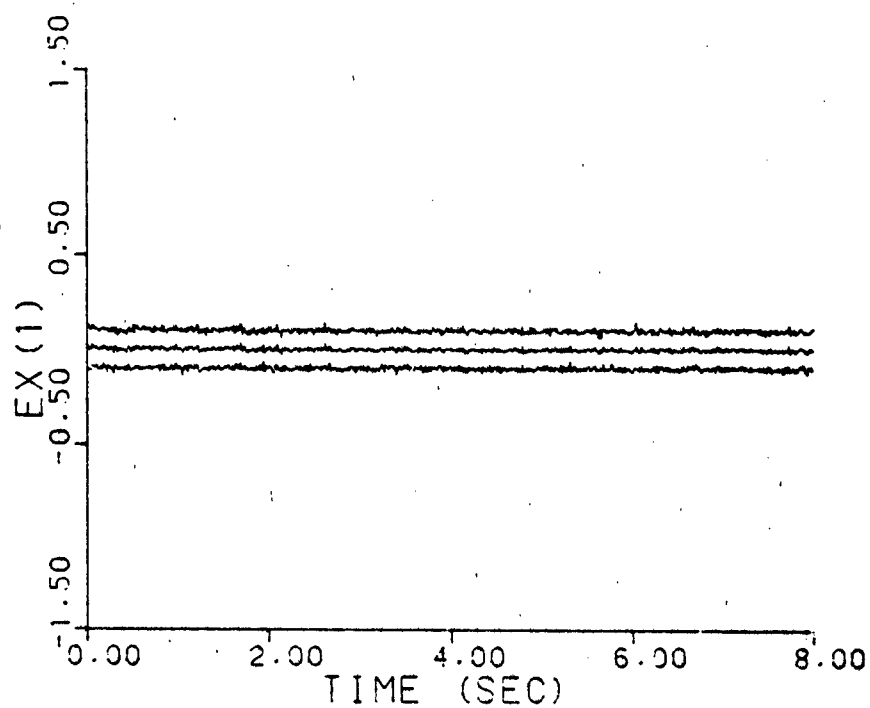


Figure A-8g. Parameter Position and Velocity
Estimate Monitoring
 $\underline{a} = (0.93, 41.0)$, no dither

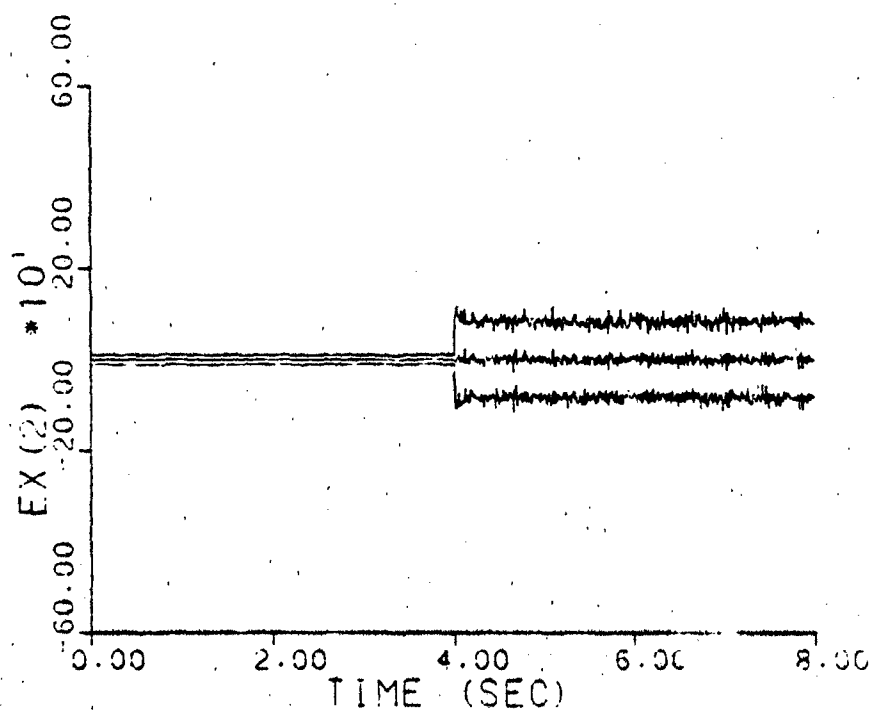
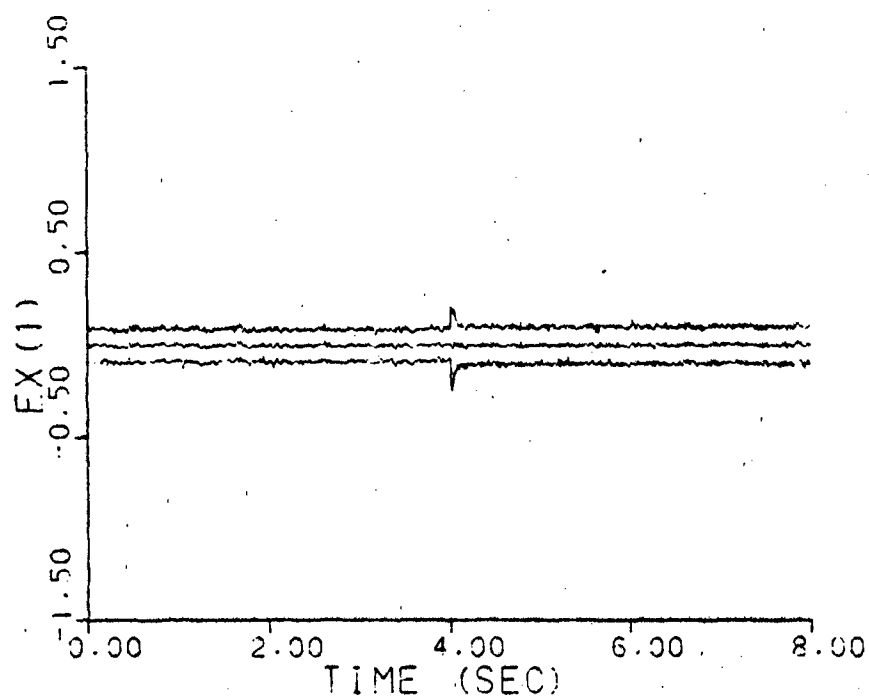


Figure A-8h. Parameter Position and Velocity
 Estimate Monitoring
 a jumps: (0.07, 9.0) - (0.93, 41.0)
 No dither

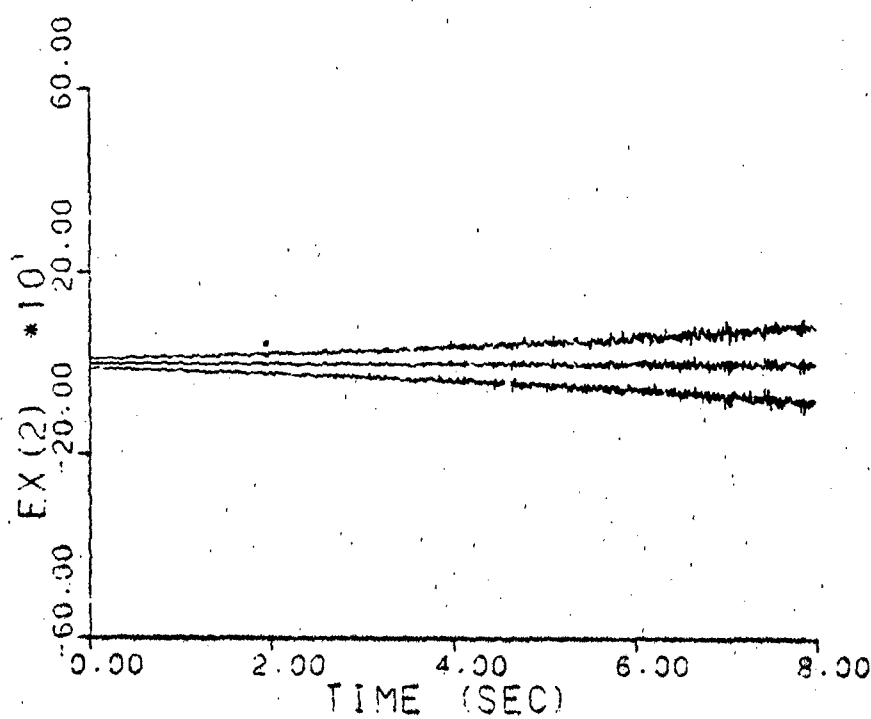
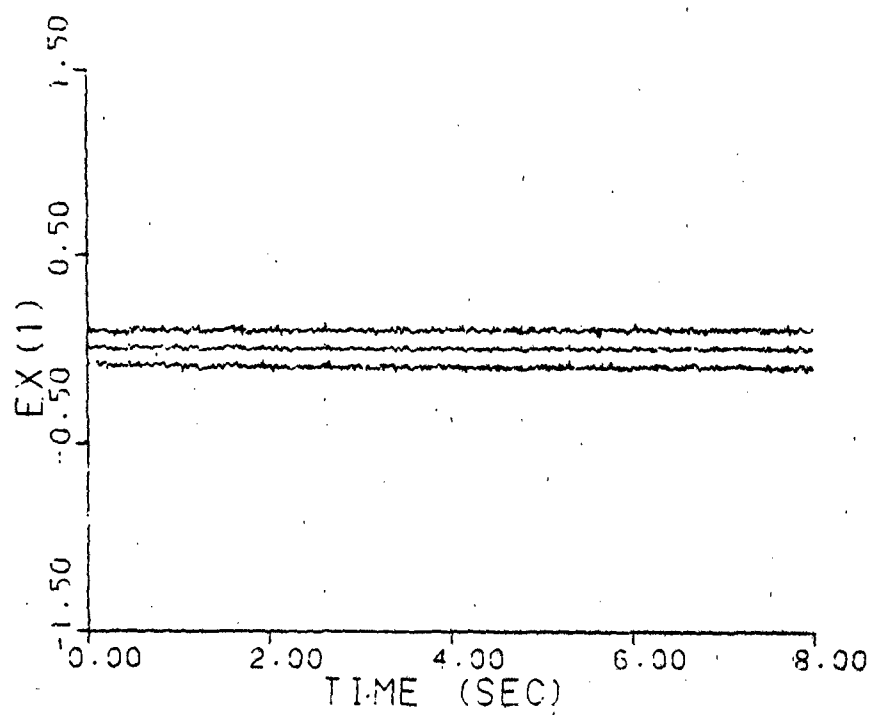


Figure A-81. Parameter Position and Velocity
 Estimate Monitoring
 a varies: (0.07, 9.0) - (0.93, 41.0)
 No dither

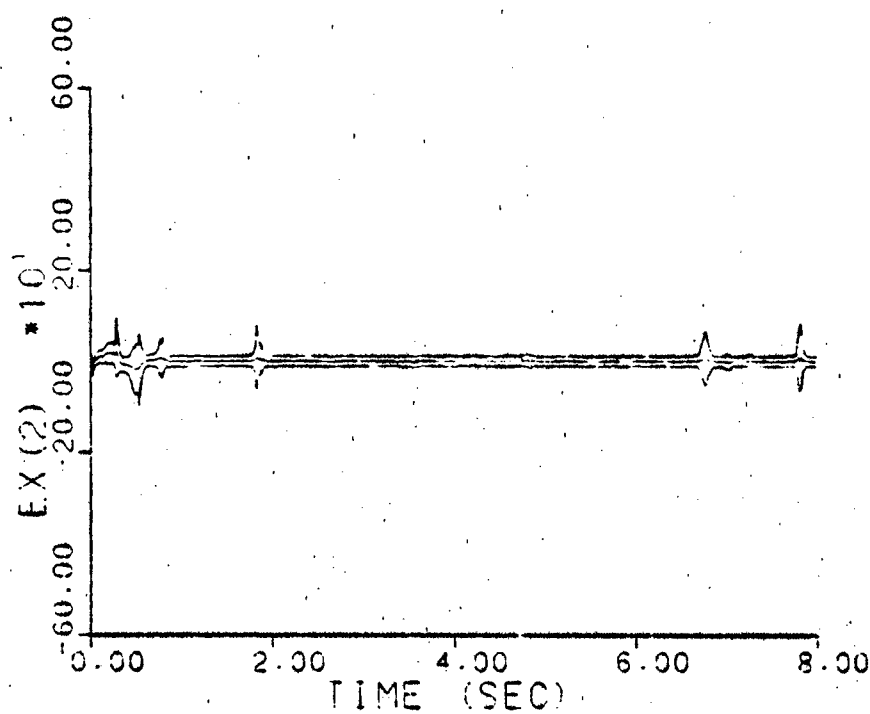
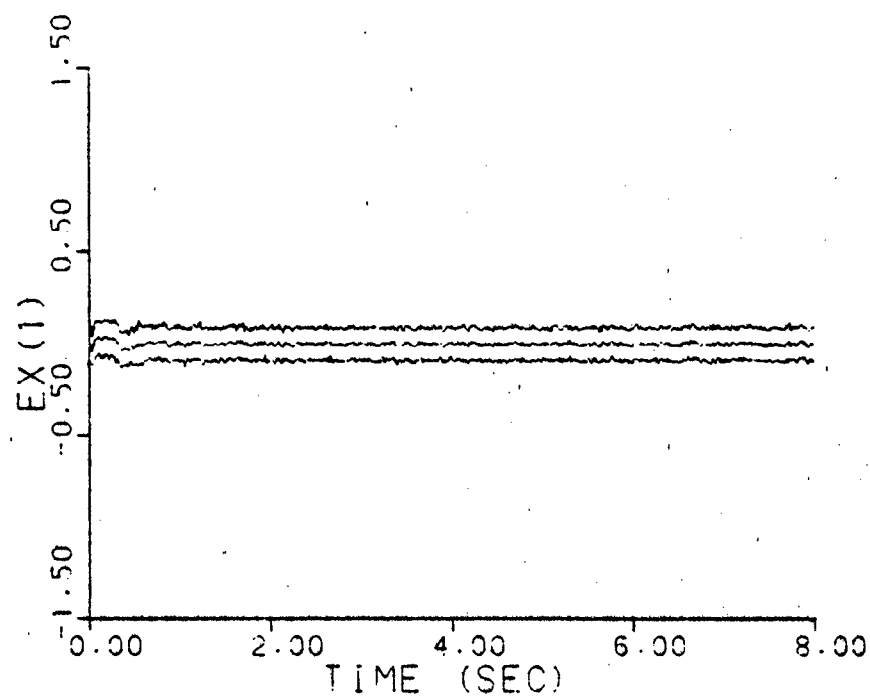


Figure A-9a. Parameter Position and Velocity
Estimate Monitoring
 $\underline{a} = (1, 3)$

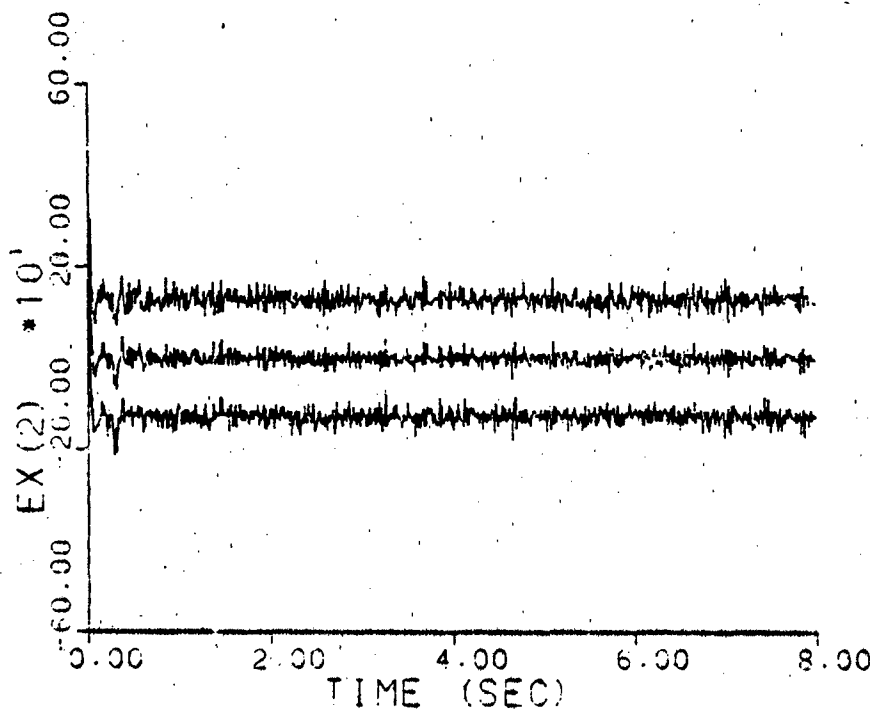
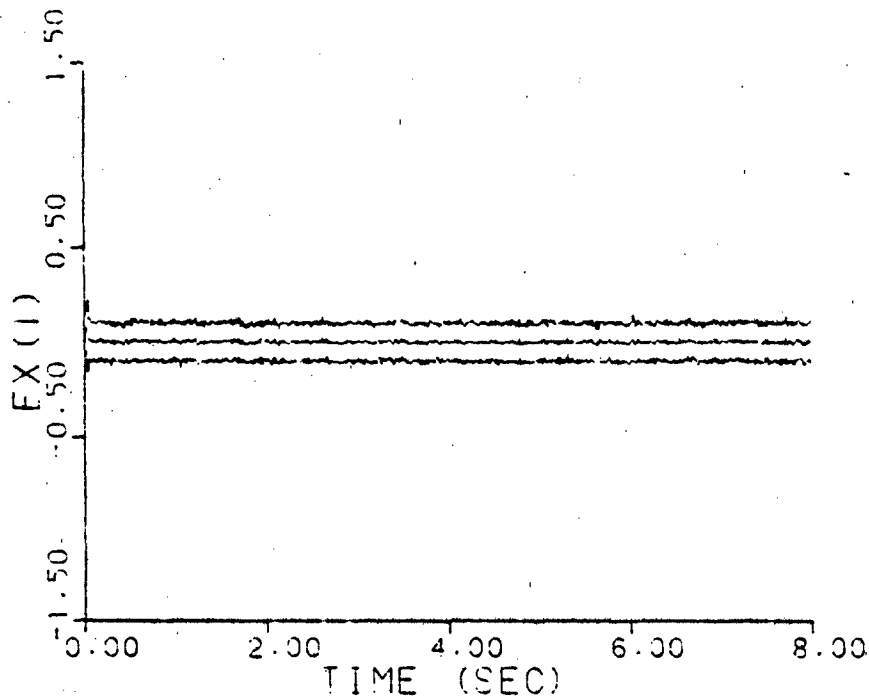


Figure A-9b. Parameter Position and Velocity
Estimate Monitoring
 $\underline{a} = (2, 9)$

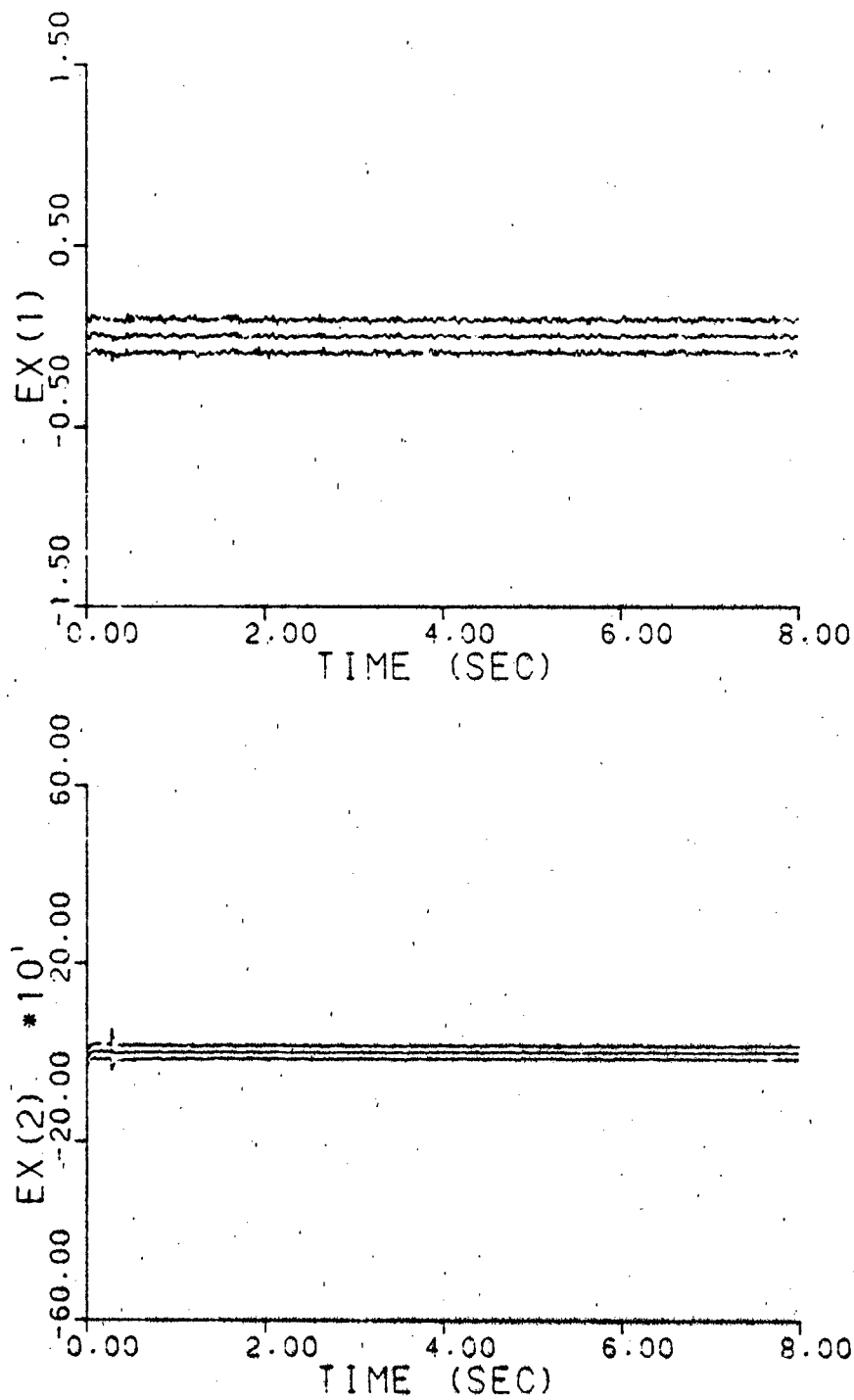


Figure A-9c. Parameter Position and Velocity
Estimate Monitoring
 $\underline{a} = (5, 4)$

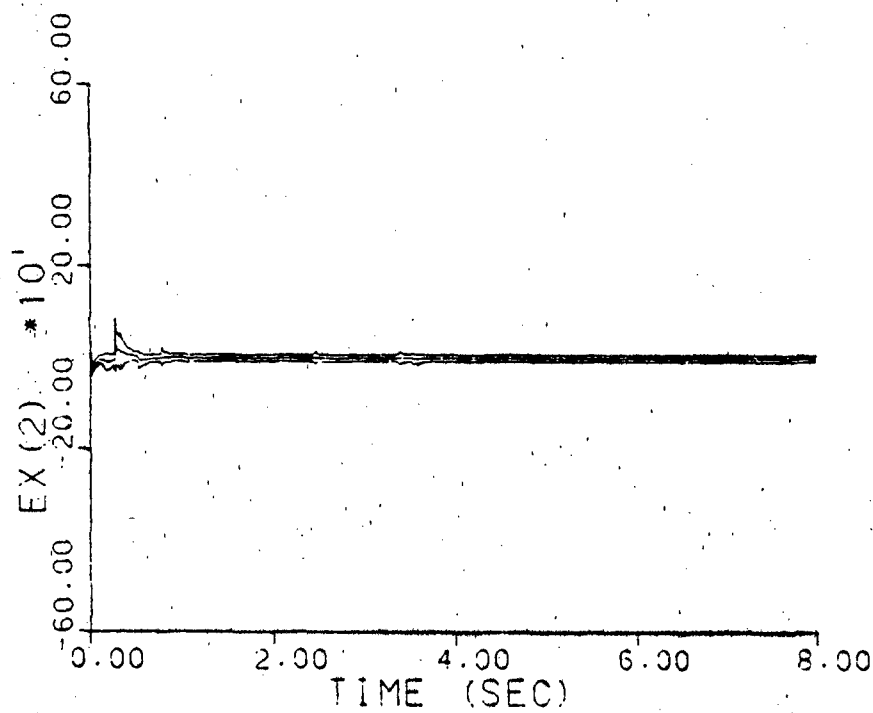
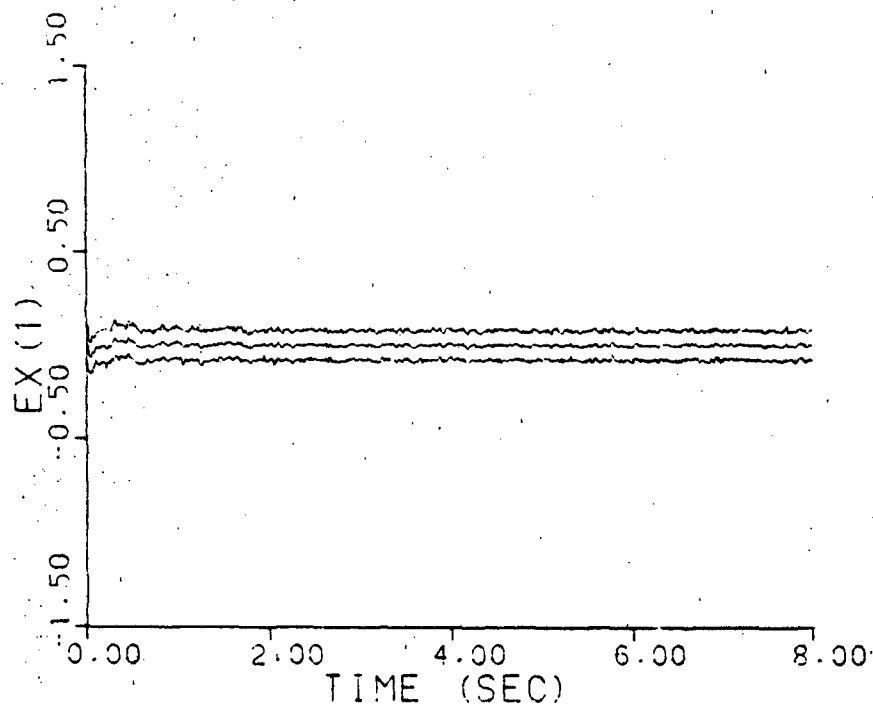


Figure A-9d. Parameter Position and Velocity
Estimate Monitoring
 $\underline{a} = (9, 2)$

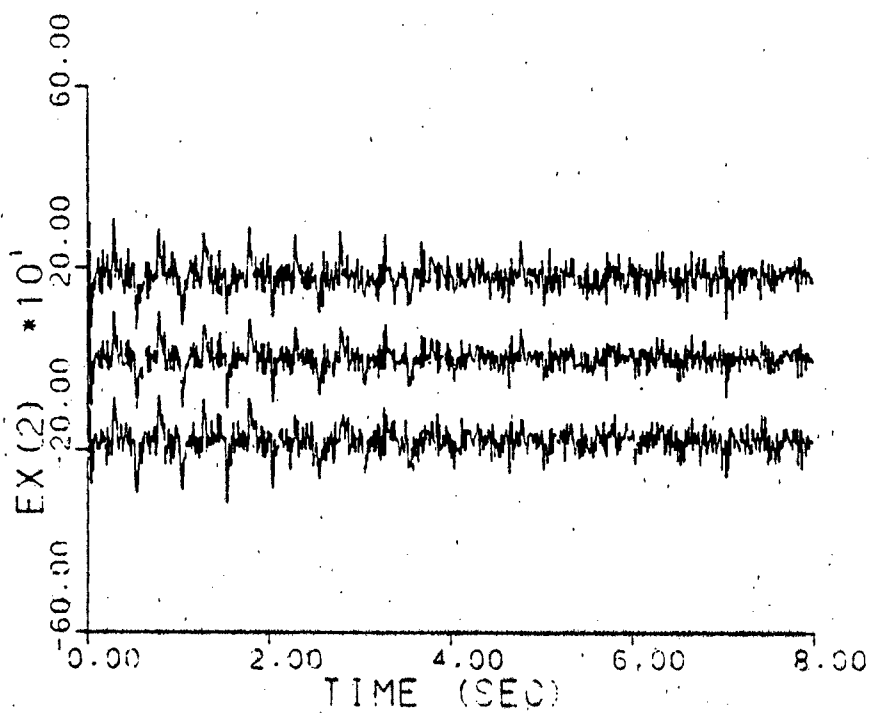
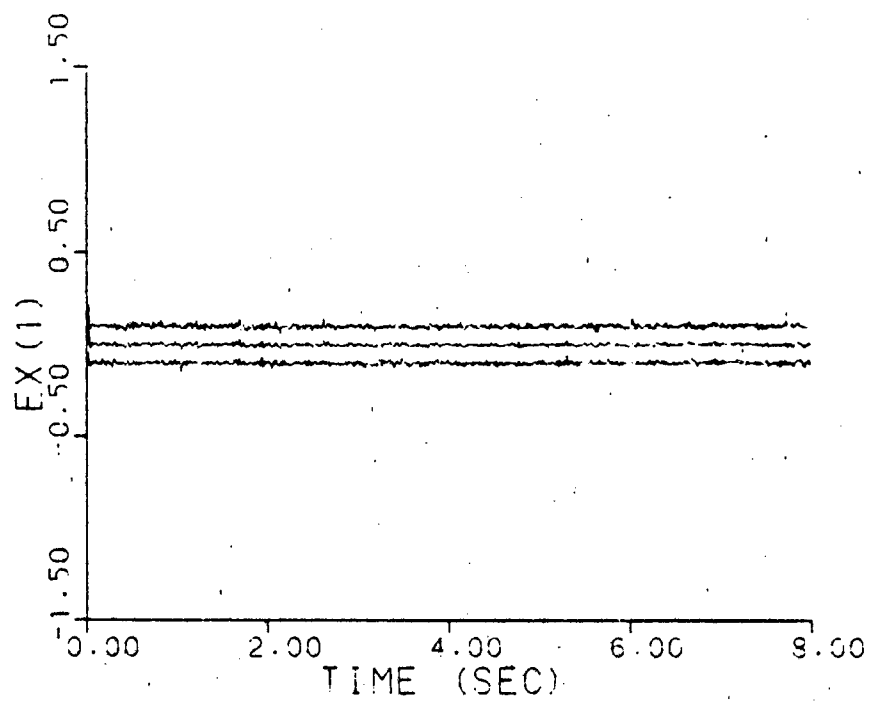


Figure A-9e. Parameter Position and Velocity
Estimate Monitoring
 $\underline{a} = (10, 10)$

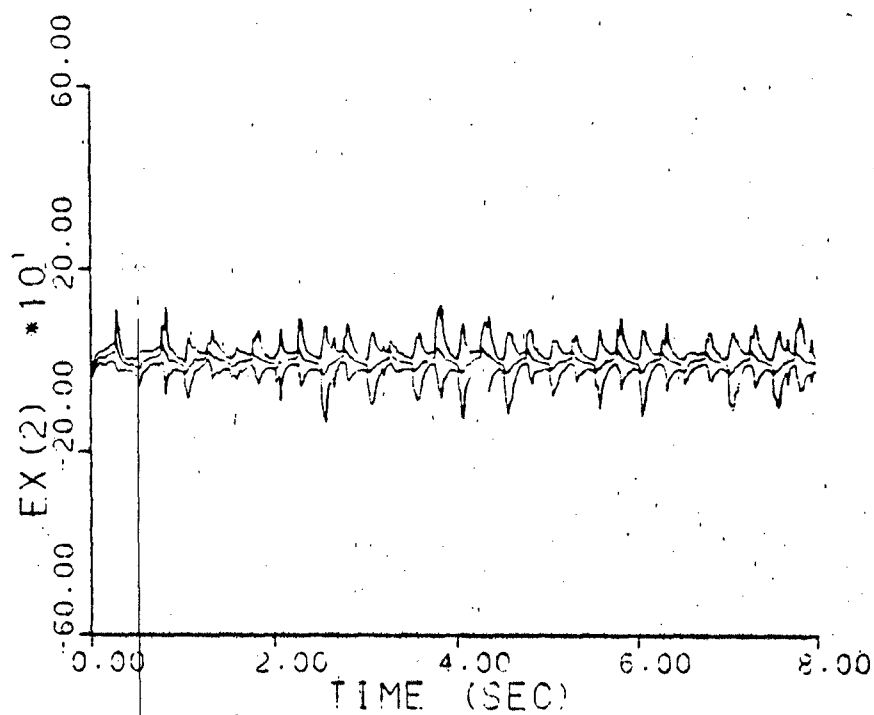
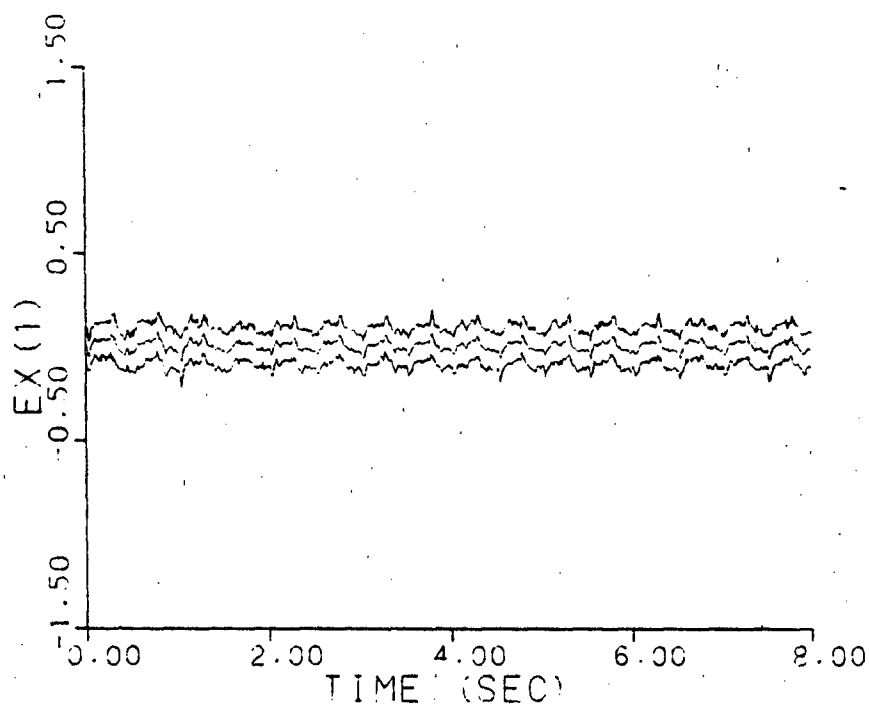


Figure A-9f. Parameter Position and Velocity
Estimate Monitoring
 $\underline{a} = (0.07, 9.0)$

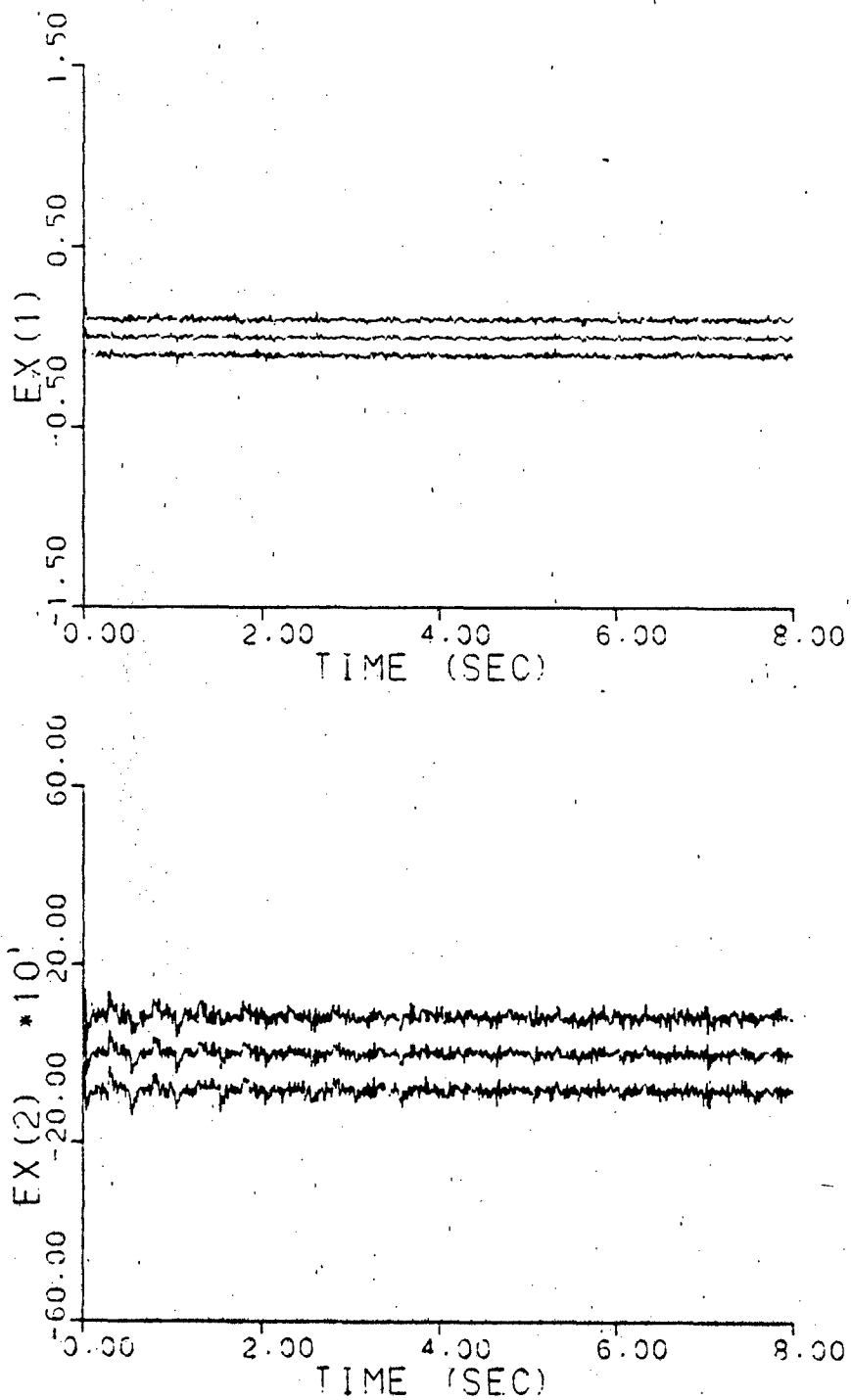


Figure A-9g. Parameter Position and Velocity
Estimate Monitoring
 $\underline{a} = (0.93, 41.0)$

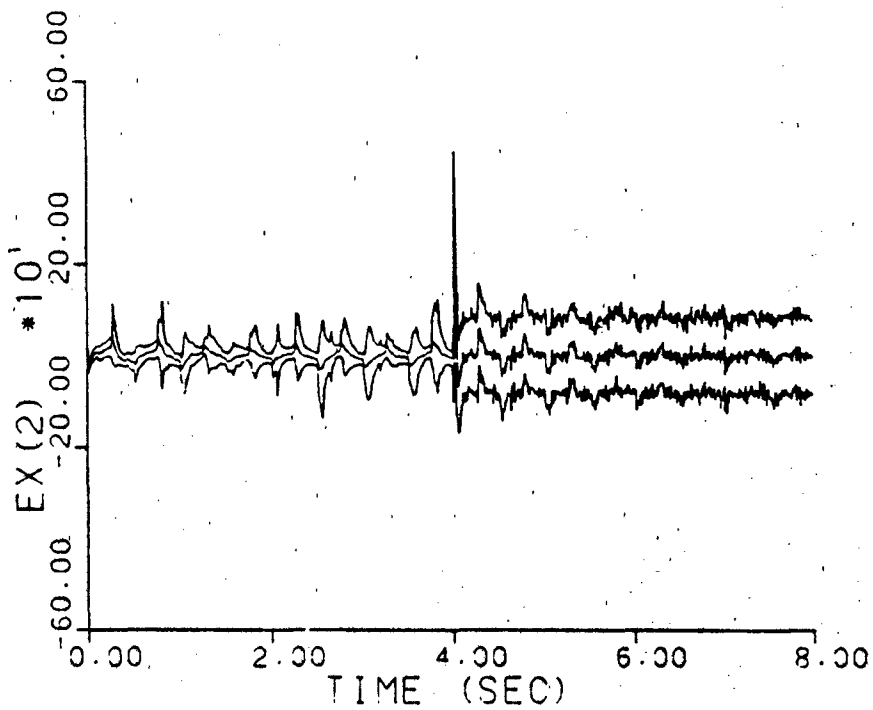
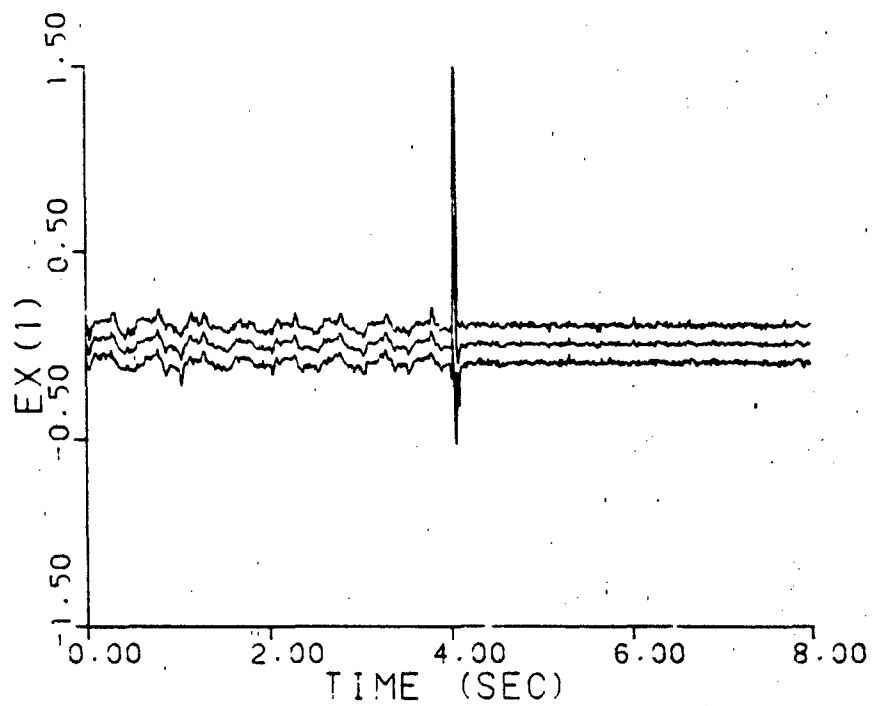


Figure A-9h. Parameter Position and Velocity
Estimate Monitoring
a jumps: (0.07, 9.0) - (0.93, 41.0)

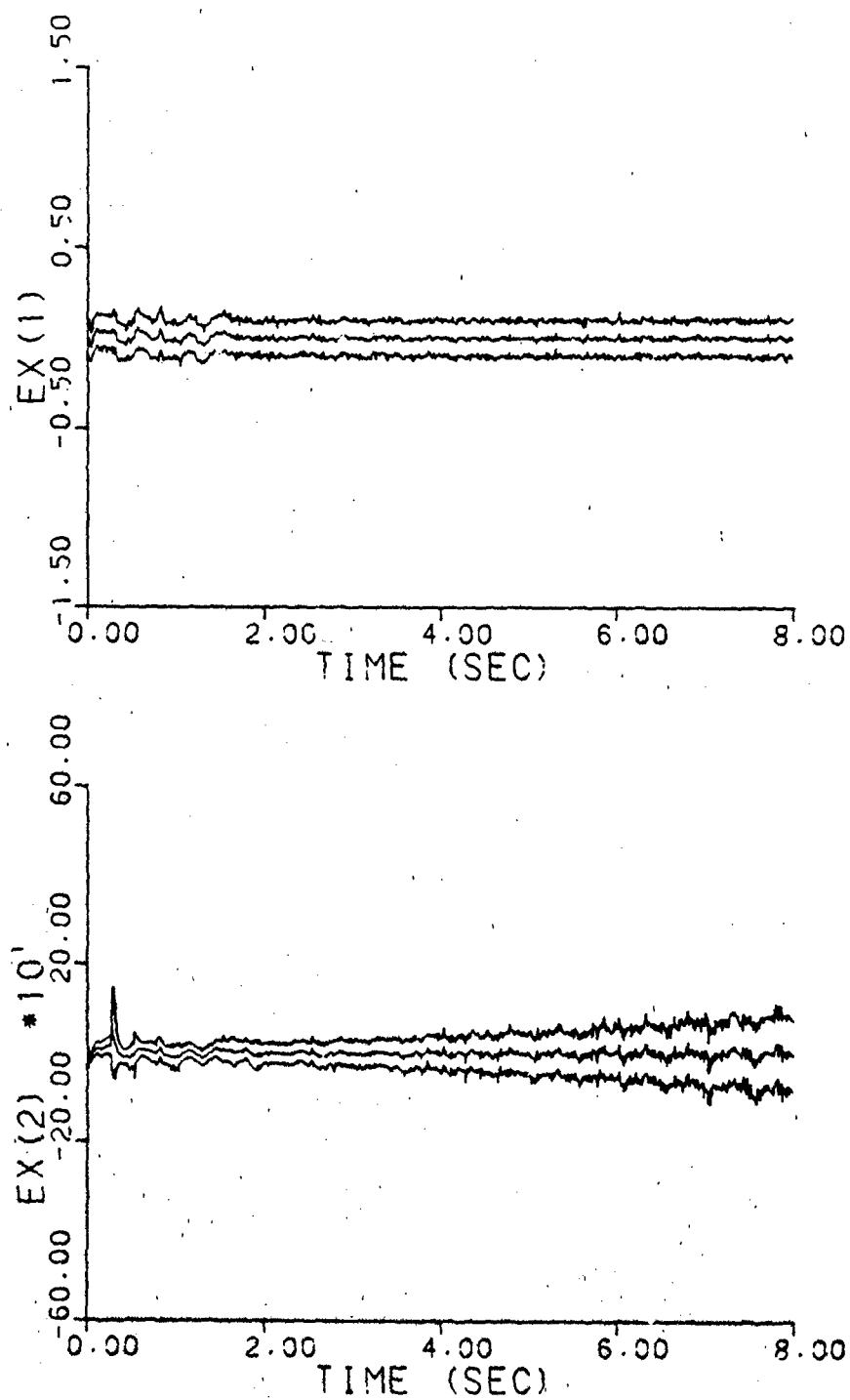


Figure A-91. Parameter Position and Velocity
Estimate Monitoring
a varies: (0.07, 9.0) - (0.93, 41.0)

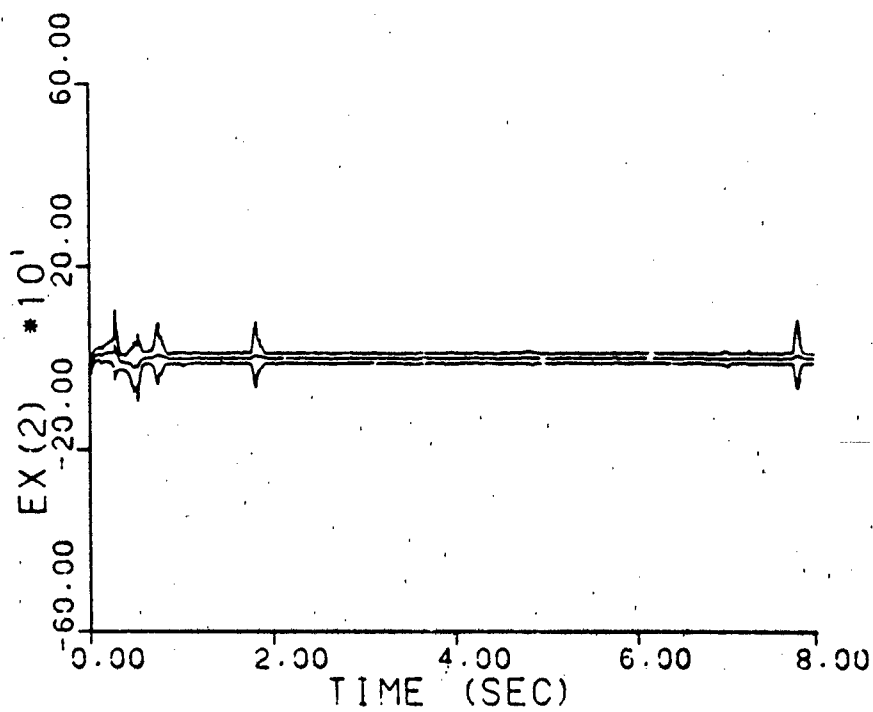
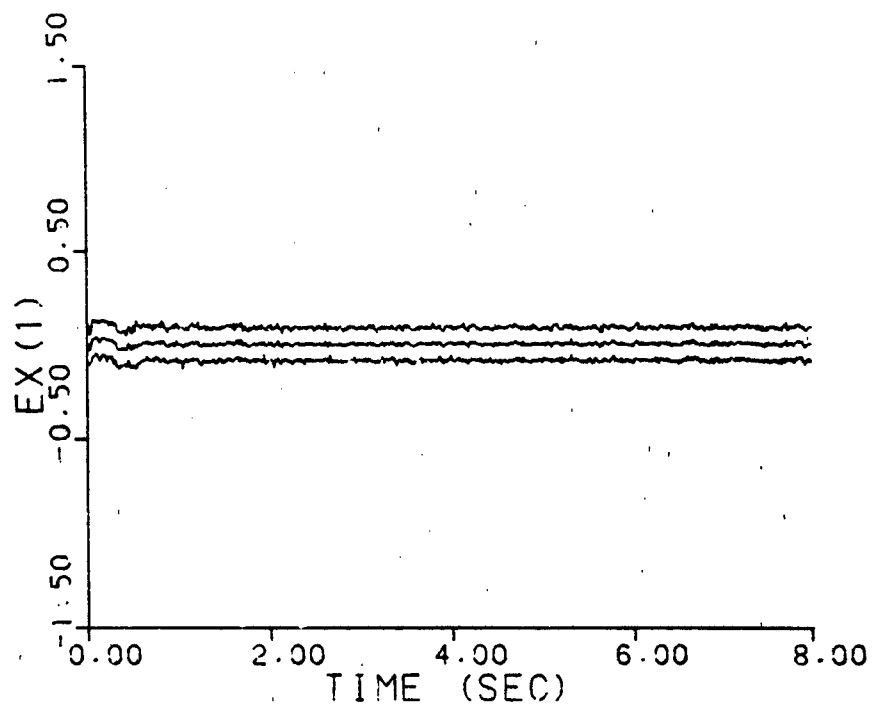


Figure A-10a. Parameter Position and Velocity
Estimate Monitoring
 $\underline{a} = (1, 3)$, warm-up.

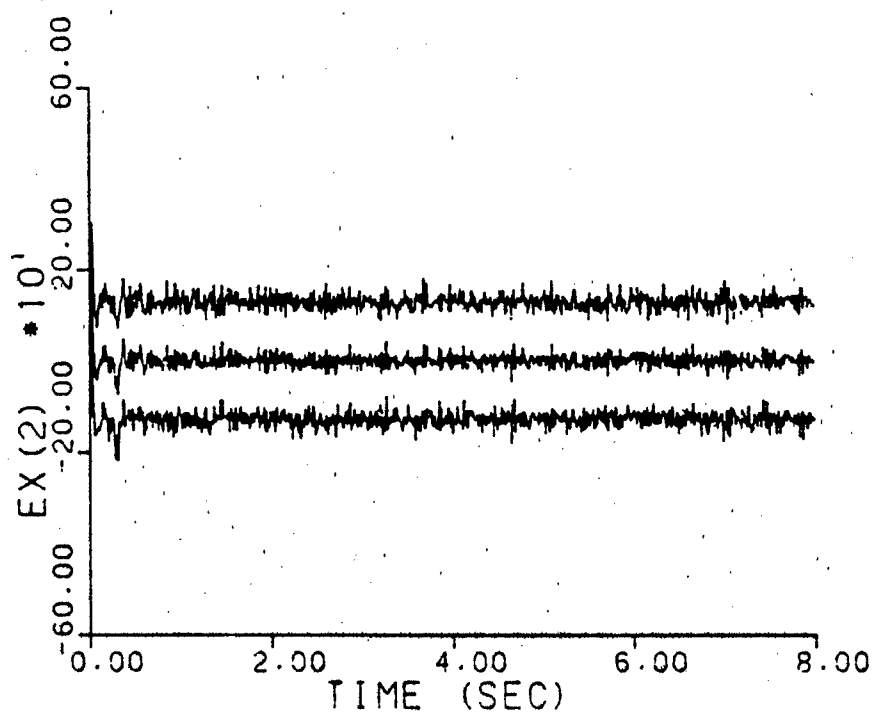
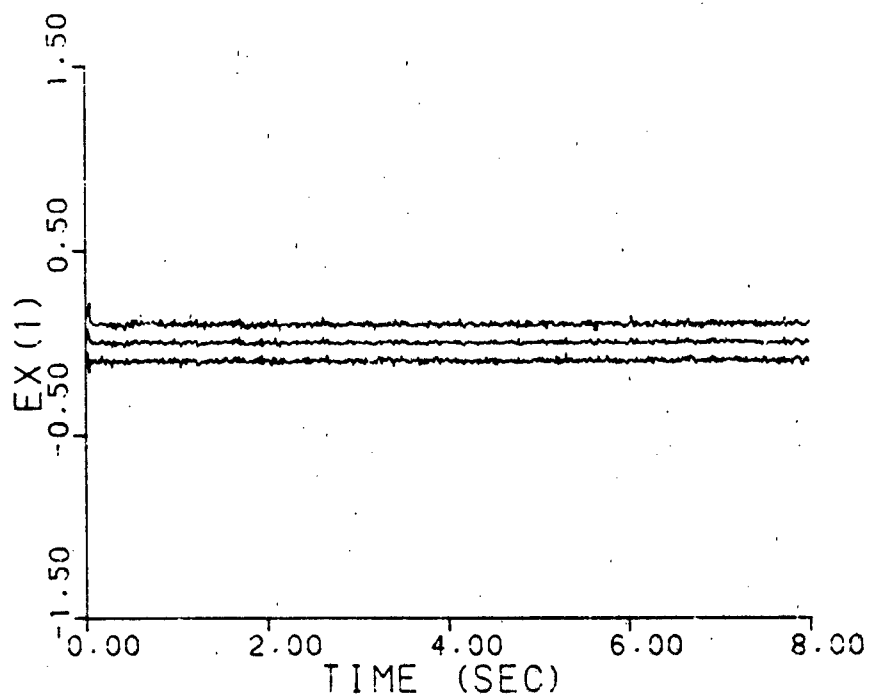


Figure A-10b. Paramater Position and Velocity
Estimate Monitoring
 $\underline{a} = (2, 9)$, warm-up

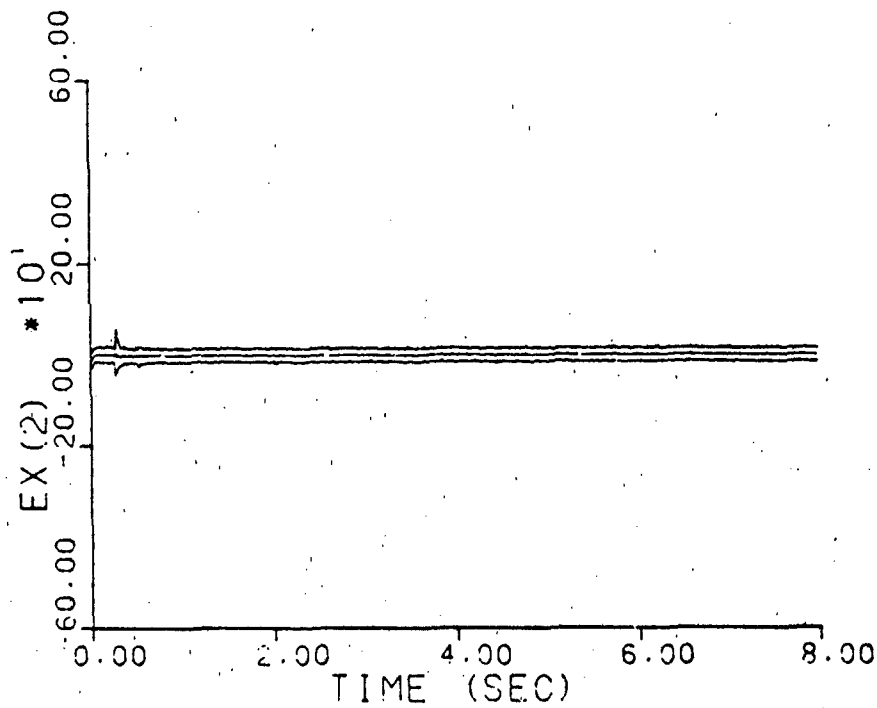
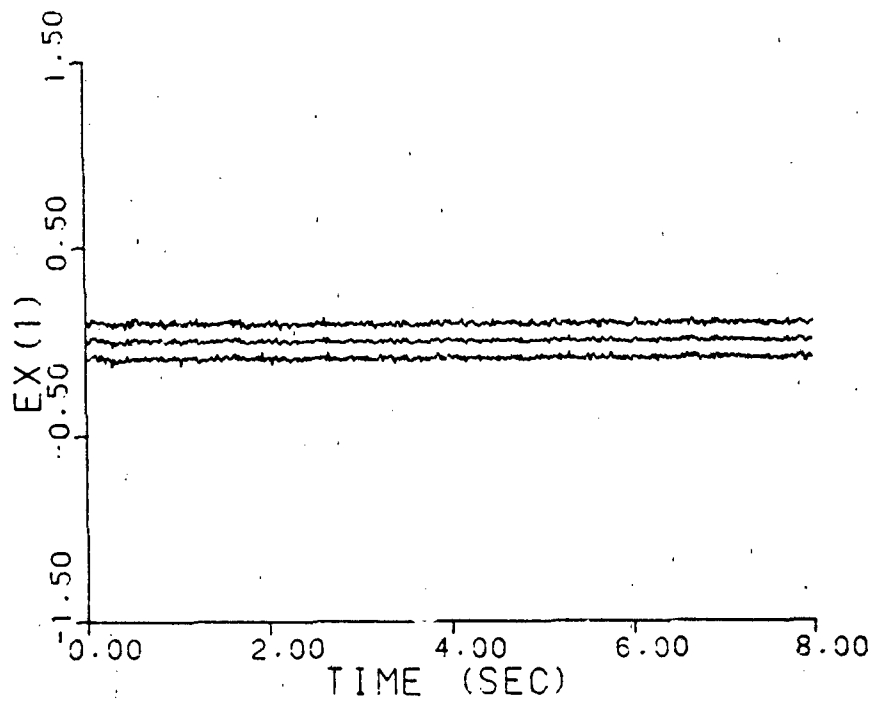


Figure A-10c. Parameter Position and Velocity
Estimate Monitoring
 $\underline{a} = (5, 4)$, warm-up

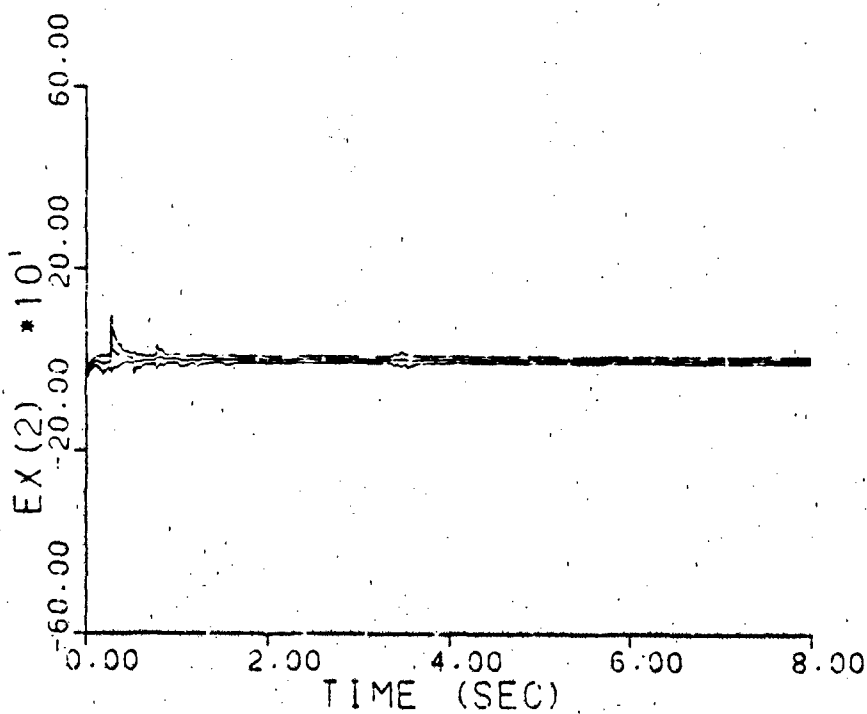
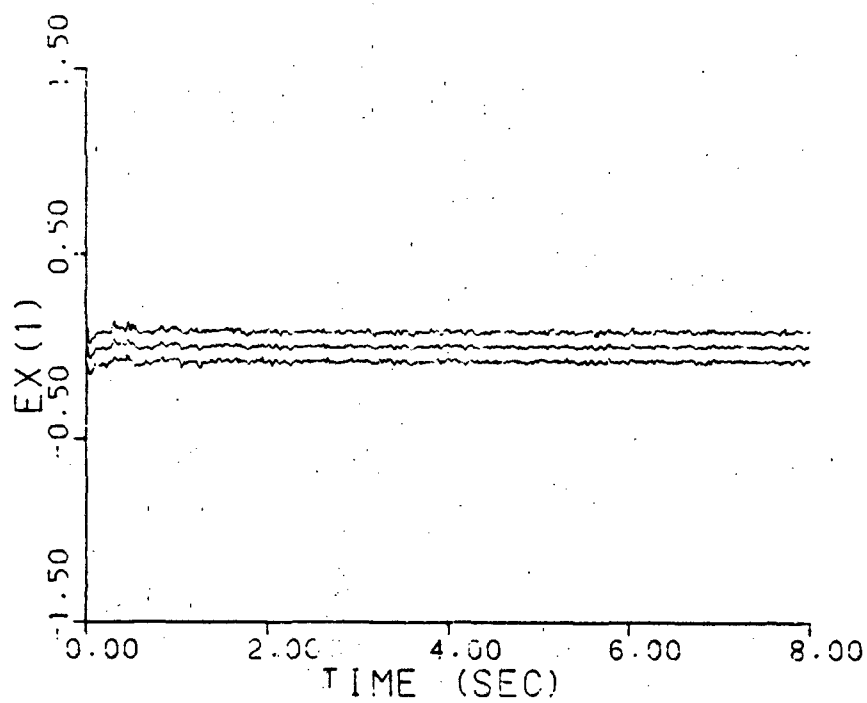


Figure A-10d. Parameter Position and Velocity
Estimate Monitoring
 $\underline{a} = (9, 2)$, warm-up

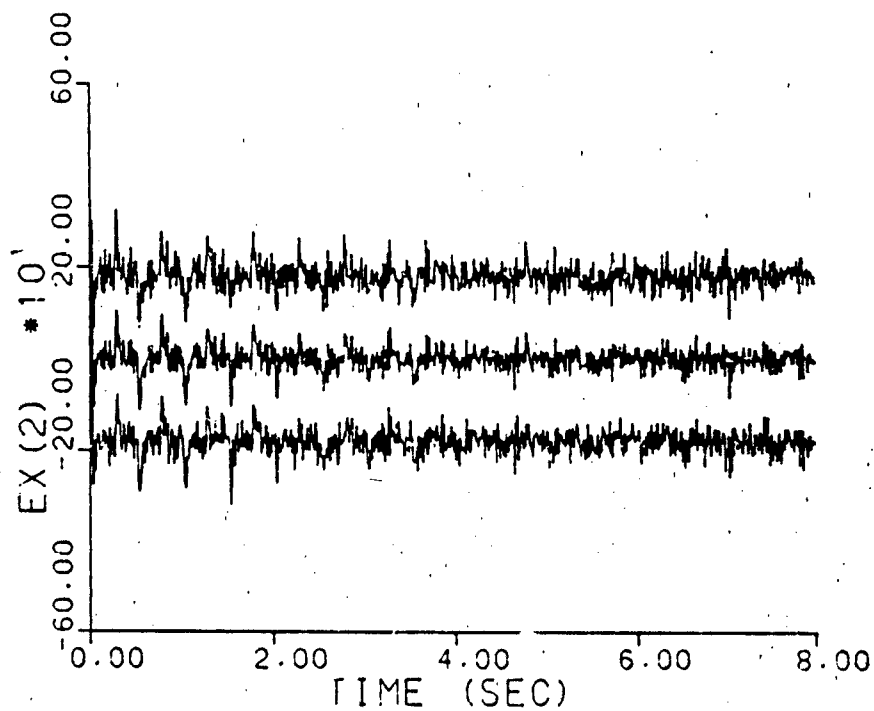
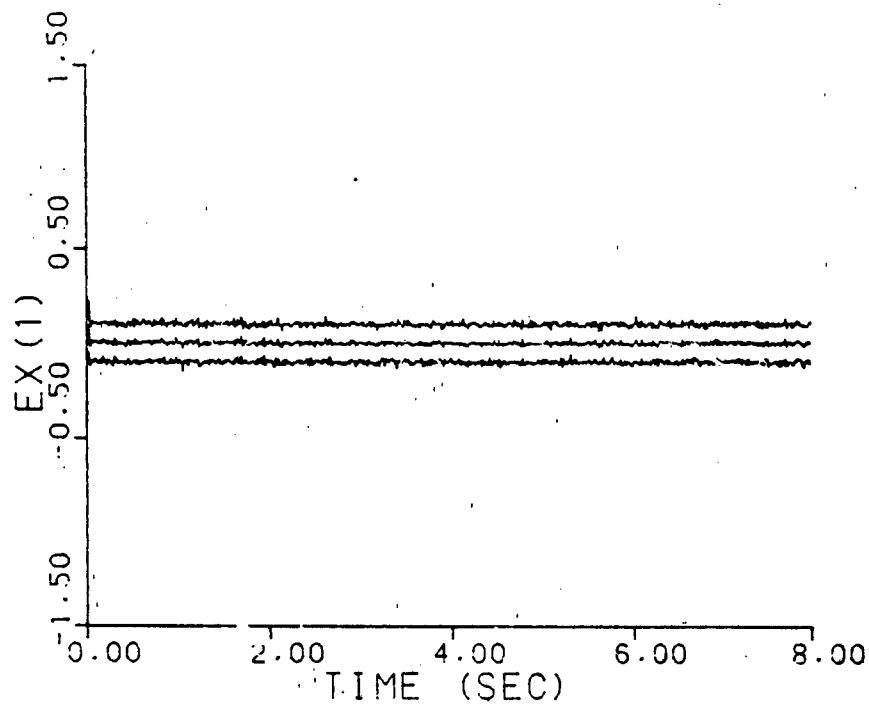


Figure A-10e. Parameter Position and Velocity
Estimate Monitoring
 $\underline{a} = (10, 10)$, warm-up

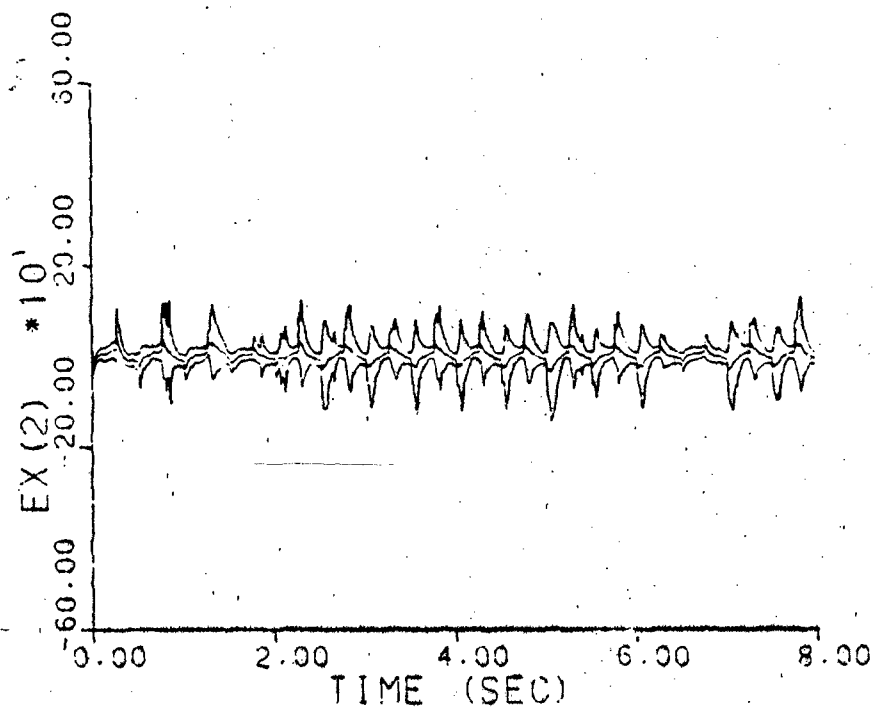
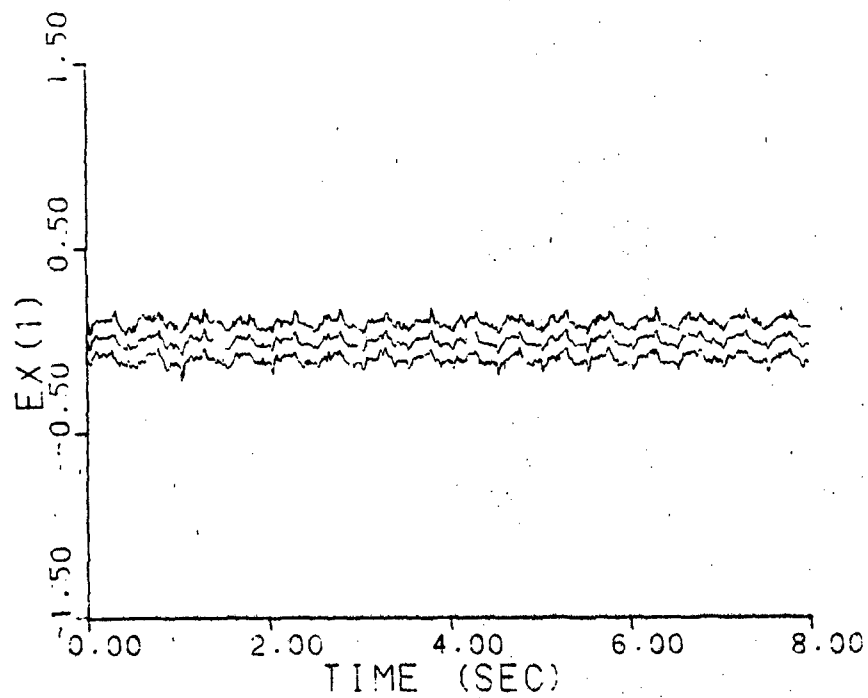


Figure A-10f. Parameter Position and Velocity
Estimate Monitoring
 $\underline{a} = (0.07, 9.0)$, warm-up

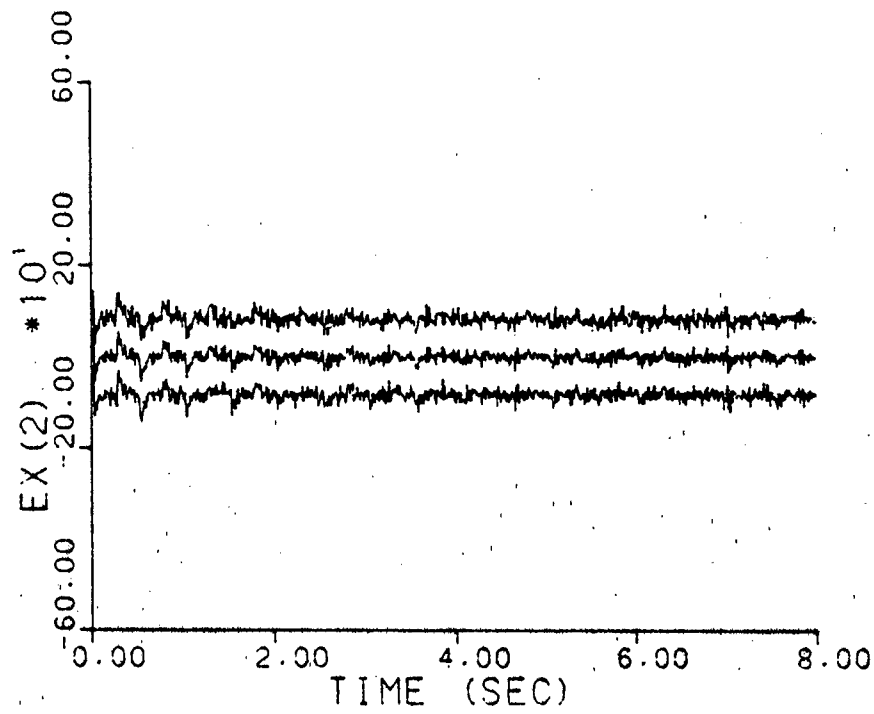
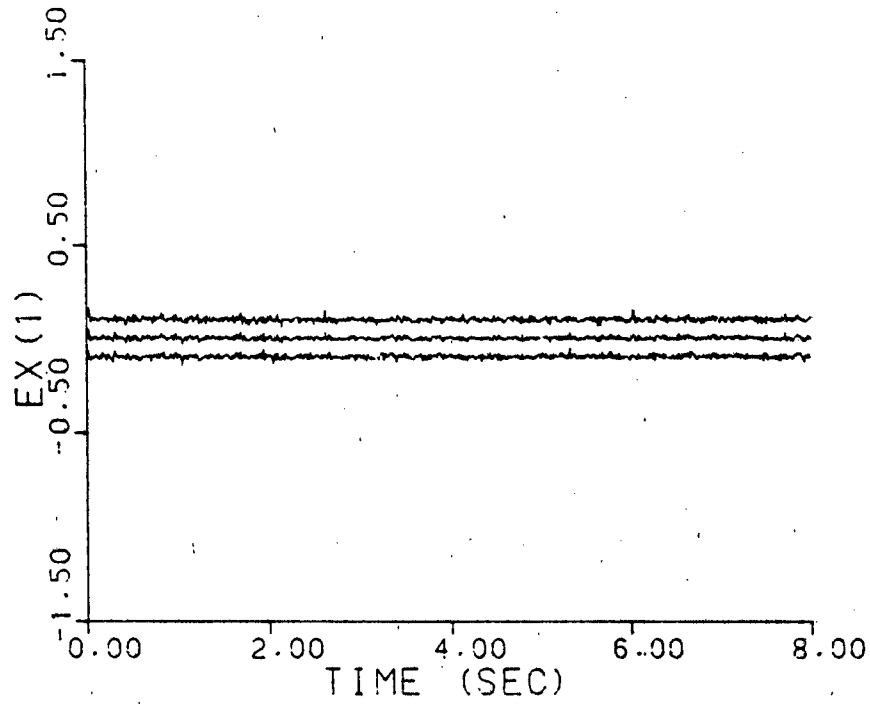


Figure A-10g. Parameter Position and Velocity
Estimate Monitoring
 $\underline{a} = (0.93, 41.0)$, warm-up

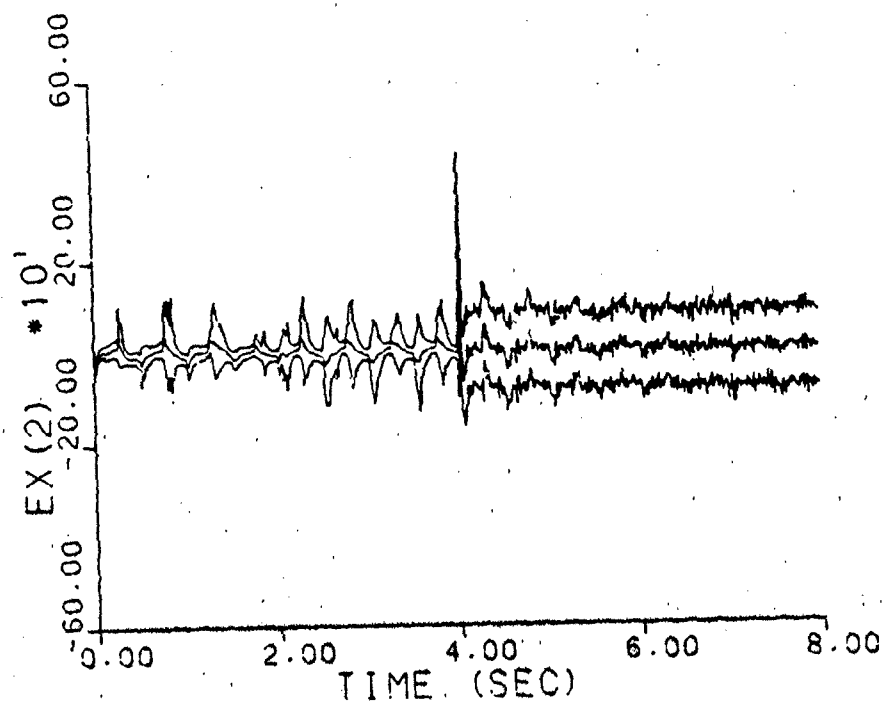
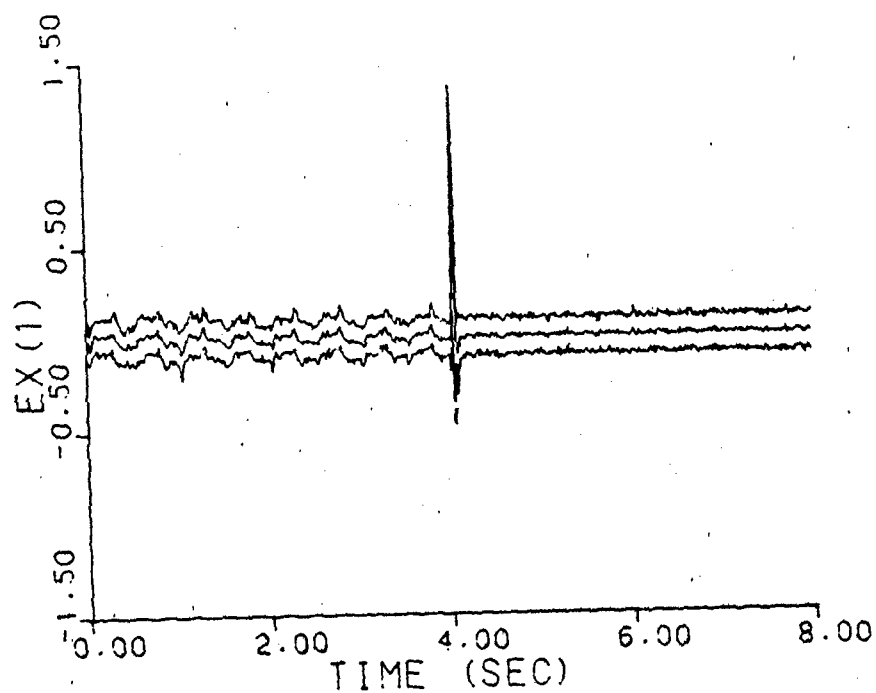


Figure A-10h. Parameter Position and Velocity
Estimate Monitoring
a jumps: (0.07, 9.0) - (0.93, 41.0)
warm-up

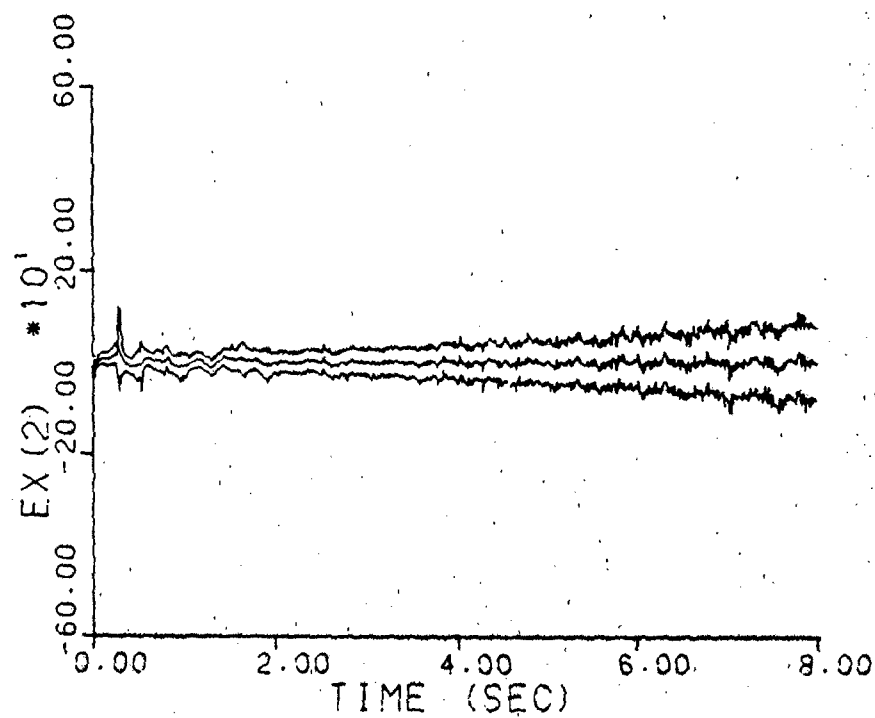
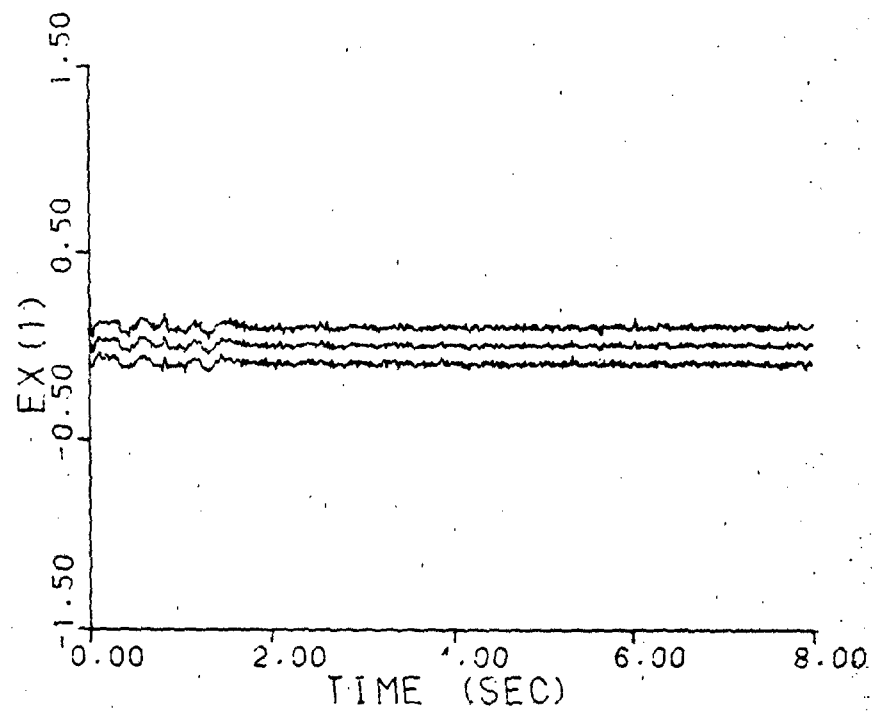


Figure A-10i. Parameter Position and Velocity
Estimate Monitoring
a varies: (0.07, 9.0) - (0.93, 41.0)
Warm-up

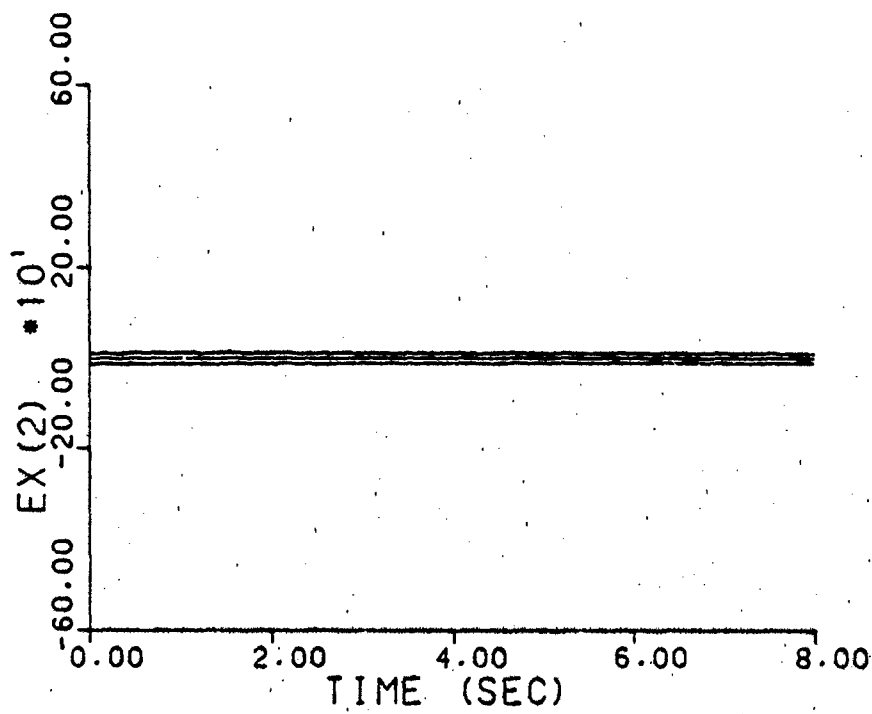
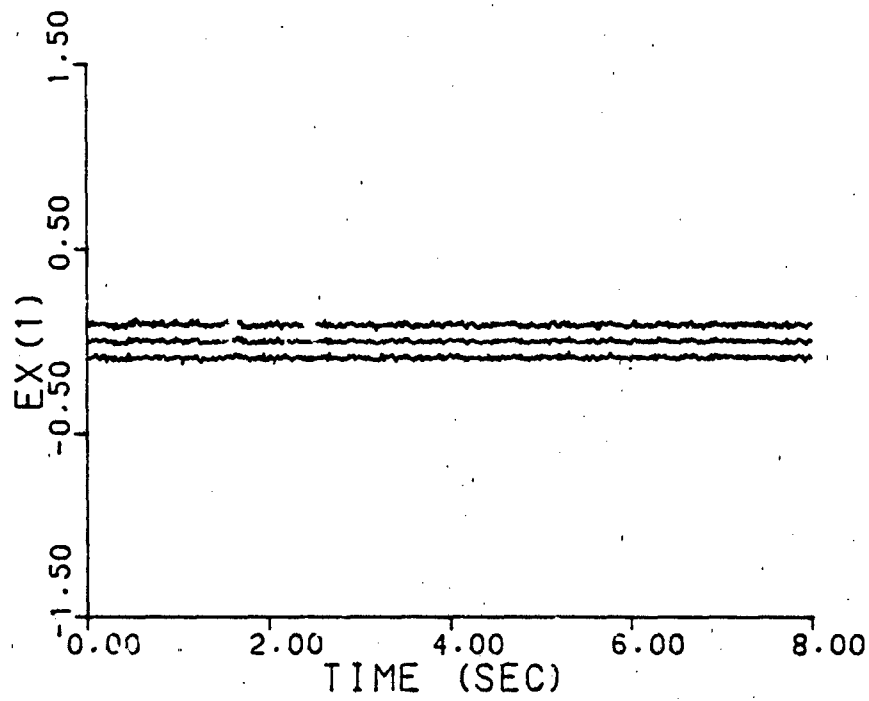


Figure A-11a. Probability Monitoring
 $\underline{a} = (1, 3)$, no dither

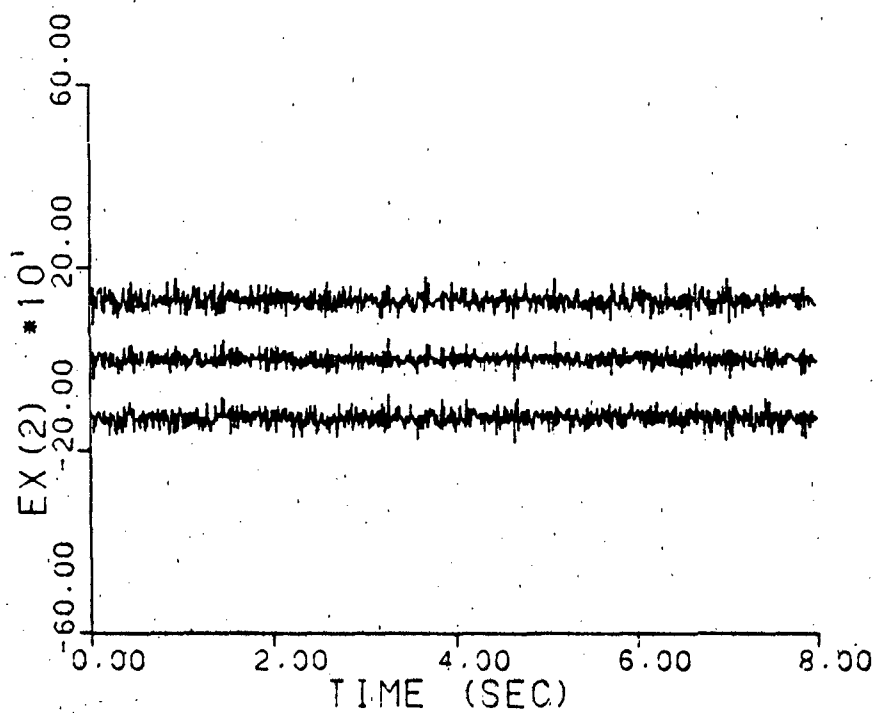
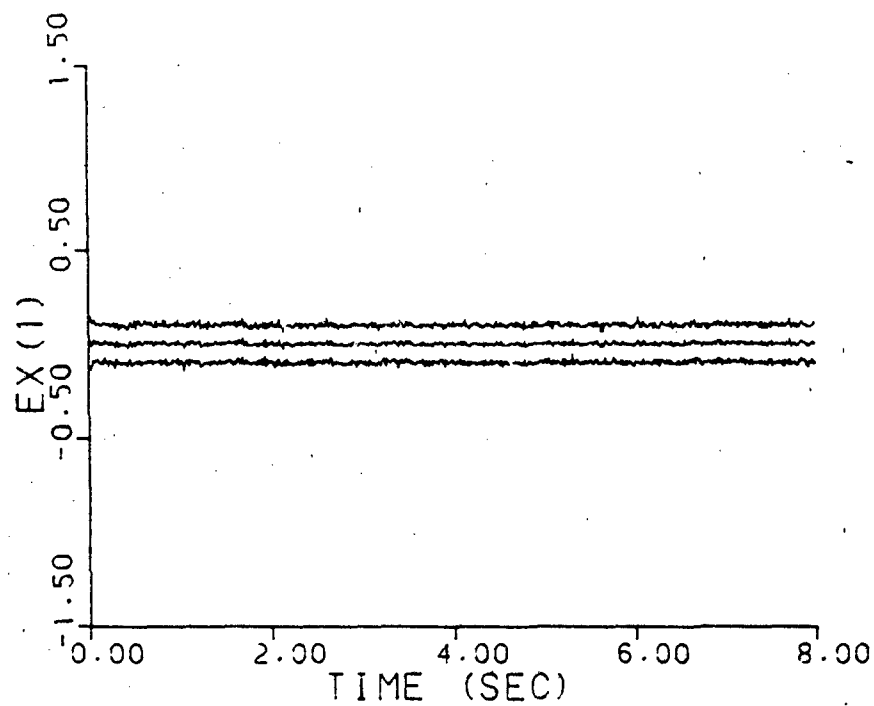


Figure A-11b. Probability Monitoring
 $\underline{a} = (2, 9)$, no dither

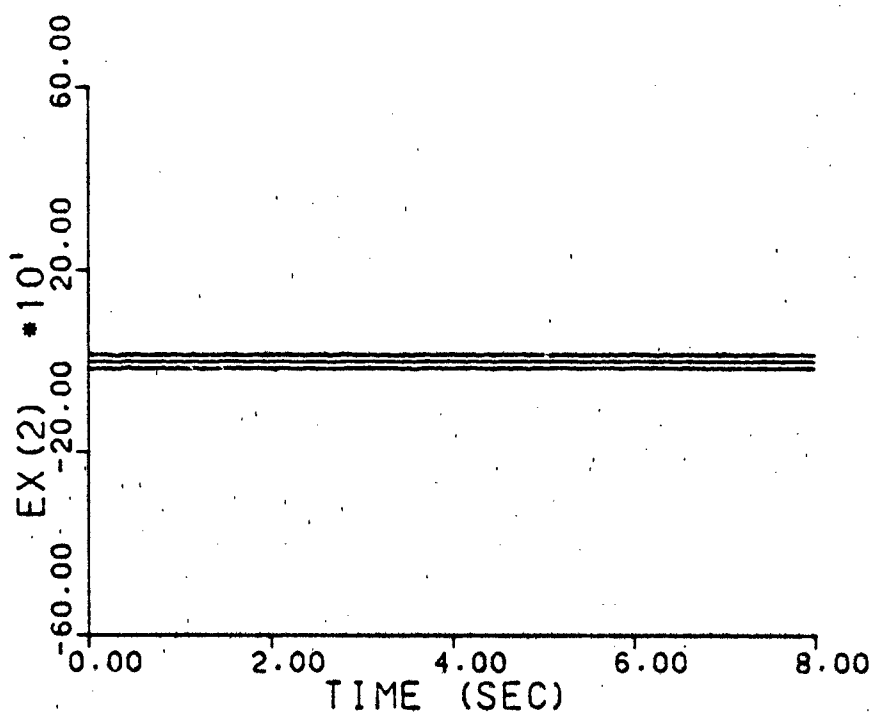
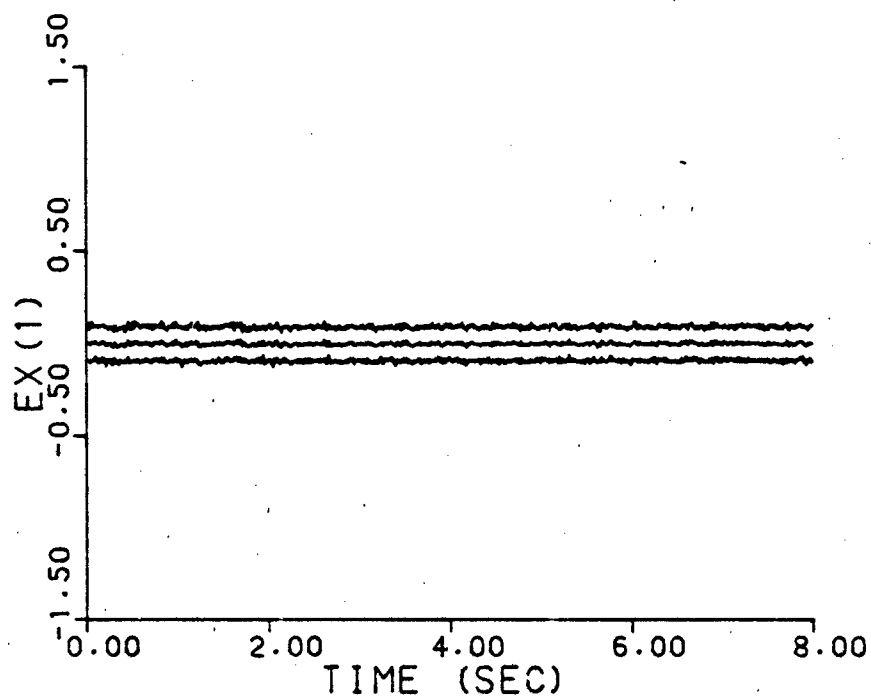


Figure A-11c. Probability Monitoring
 $\underline{a} = (5, 4)$, no dither

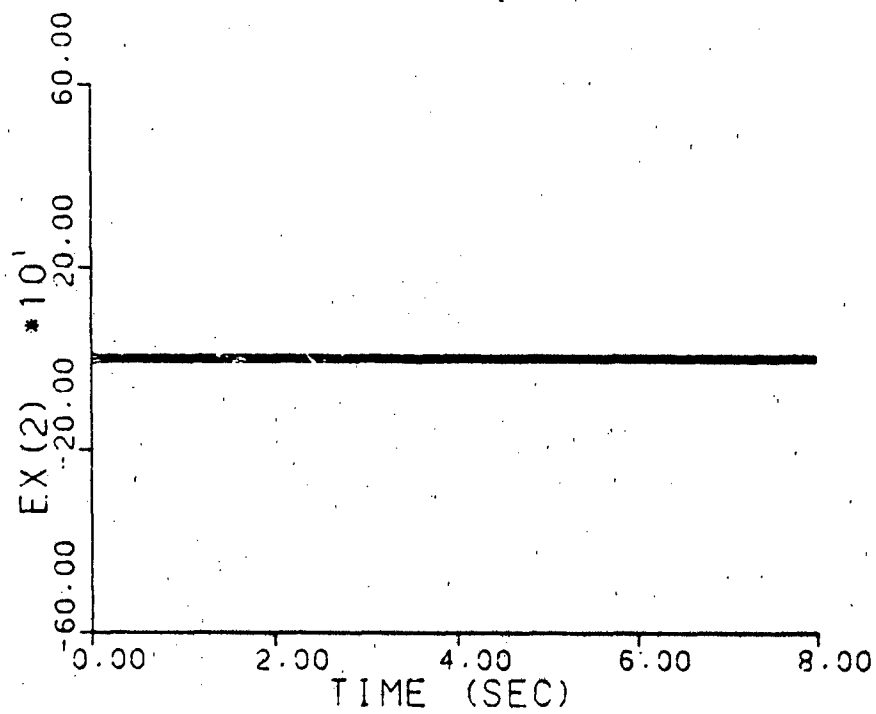
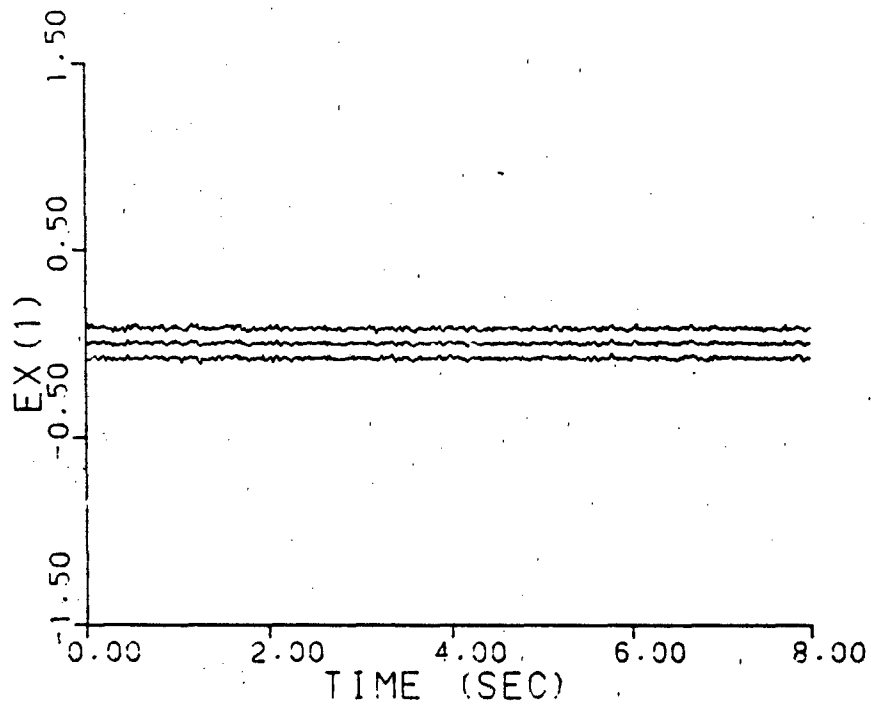


Figure A-11d. Probability Monitoring
 $\underline{a} = (9, 2)$, no dither

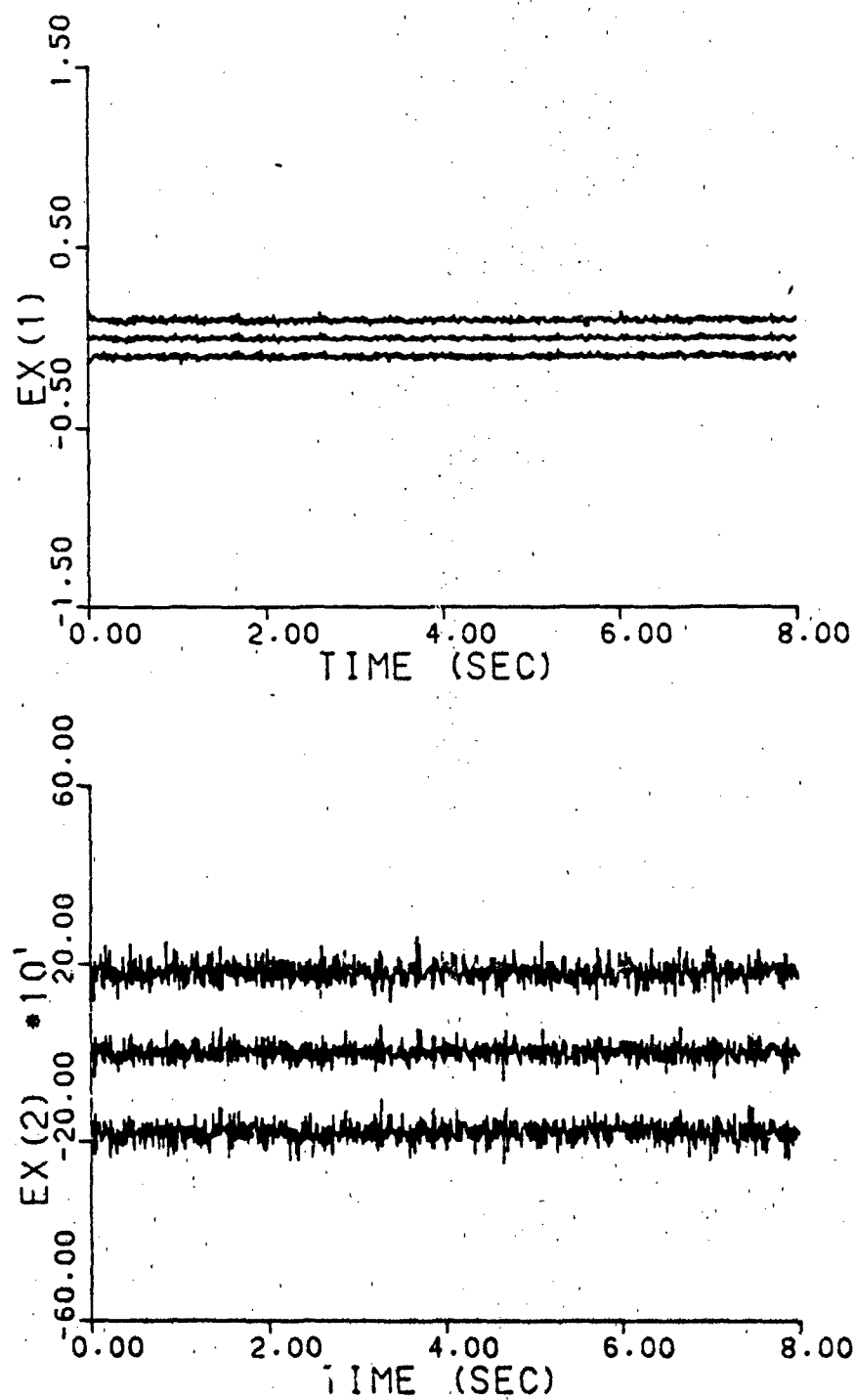


Figure A-11e. Probability Monitoring
 $\underline{a} = (10, 10)$, no dither

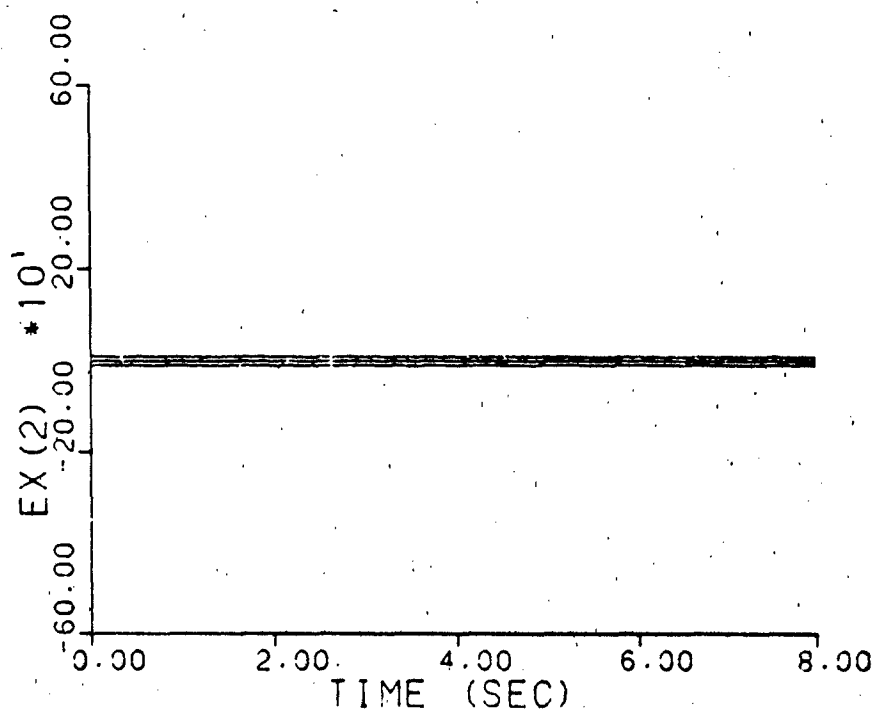
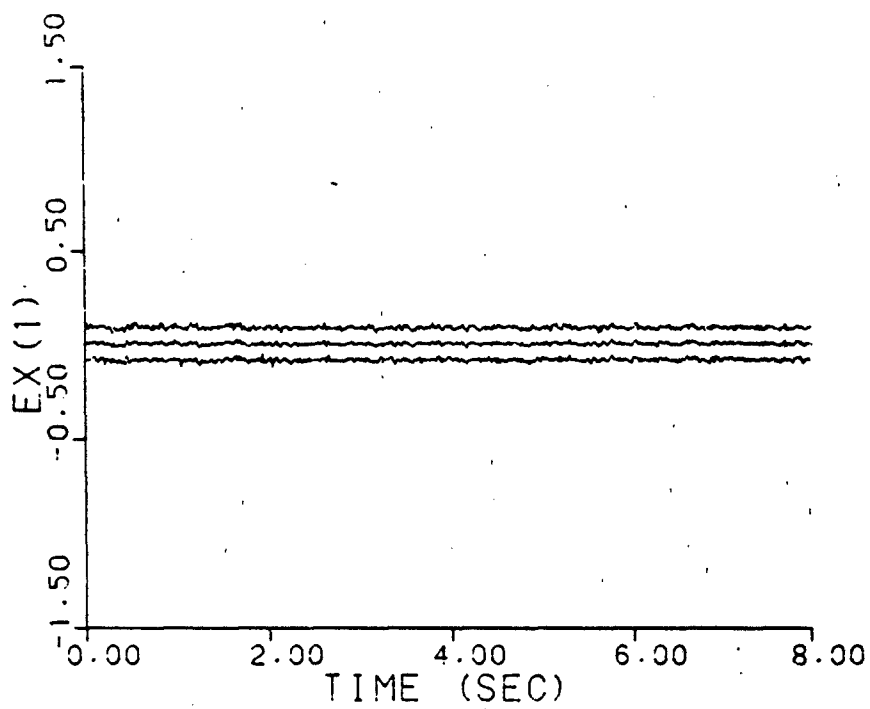


Figure A-11f. Probability Monitoring
 $\underline{a} = (0.07, 9.0)$, no dither

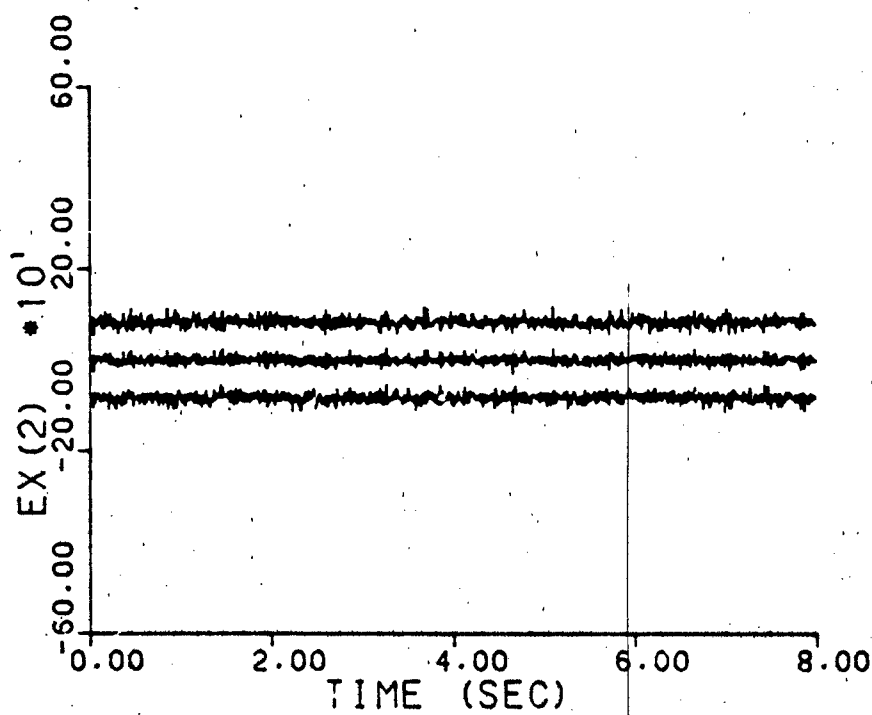
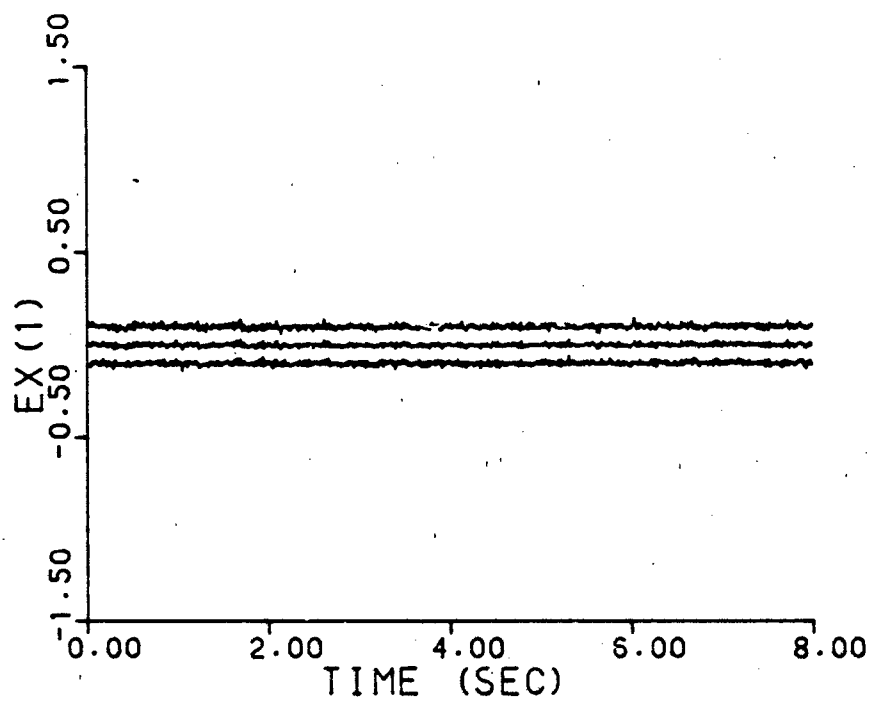


Figure A-11g. Probability Monitoring
 $\underline{a} = (0.93, 41.0)$, no dither

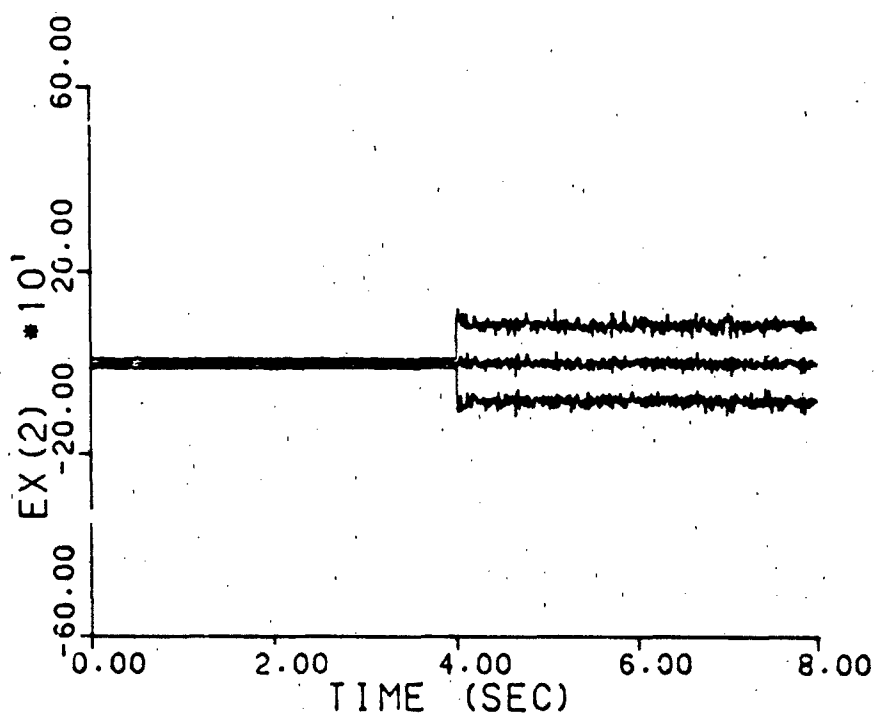
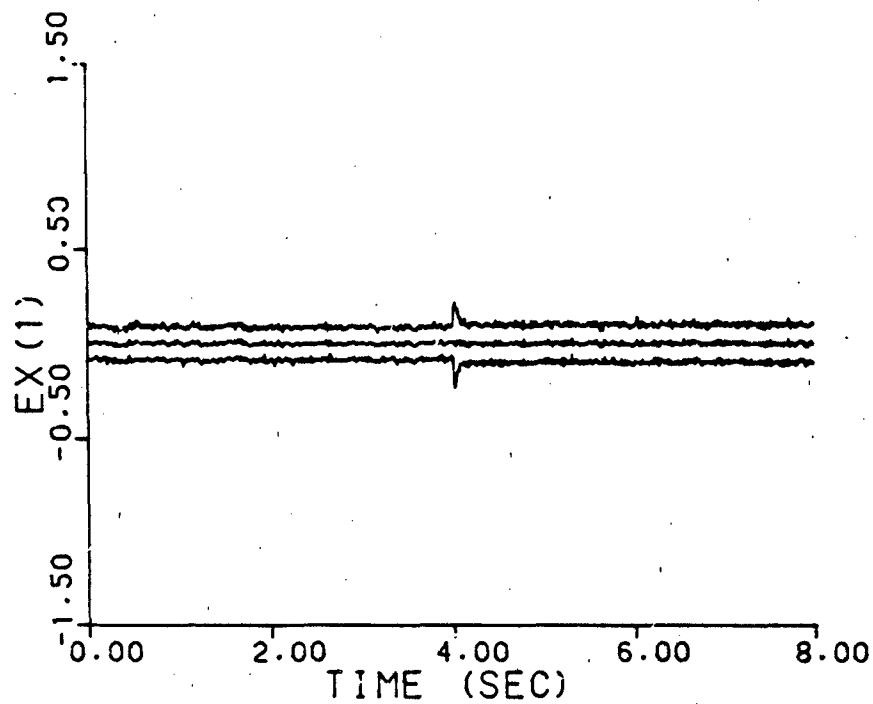


Figure A-11h. Probability Monitoring
 a jumps: (0.07, 9.0) - (0.93, 41.0)
 no dither

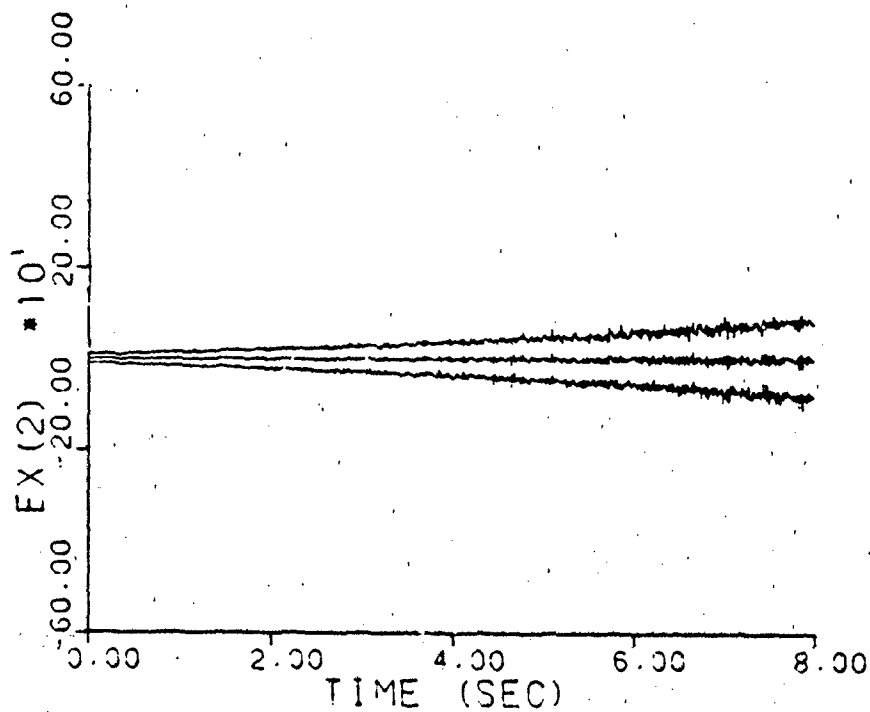
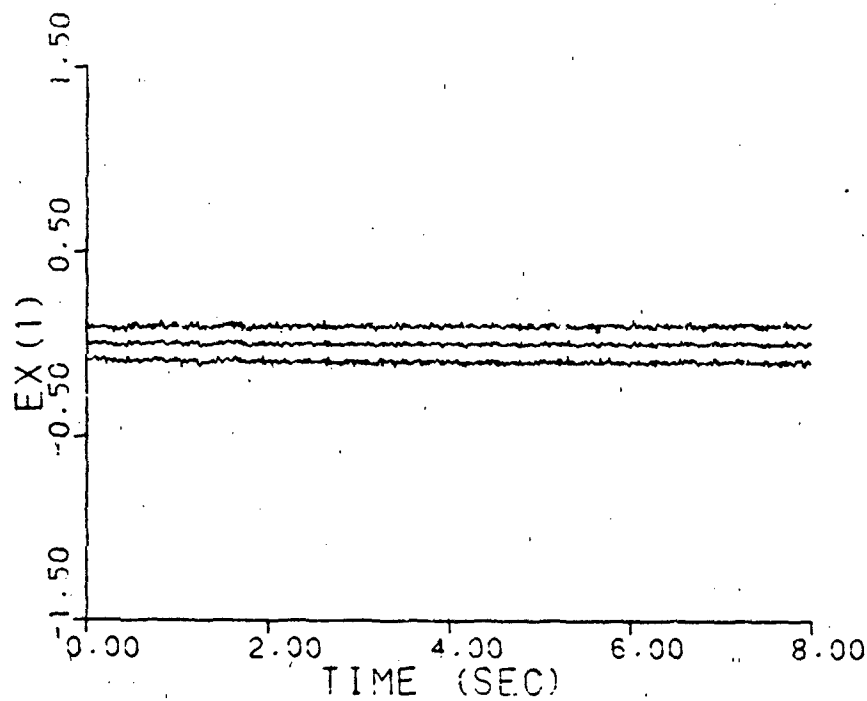


Figure A-11i. Probability Monitoring
 a varies: (0.07, 9.0) - (0.93, 41.0)
 no dither

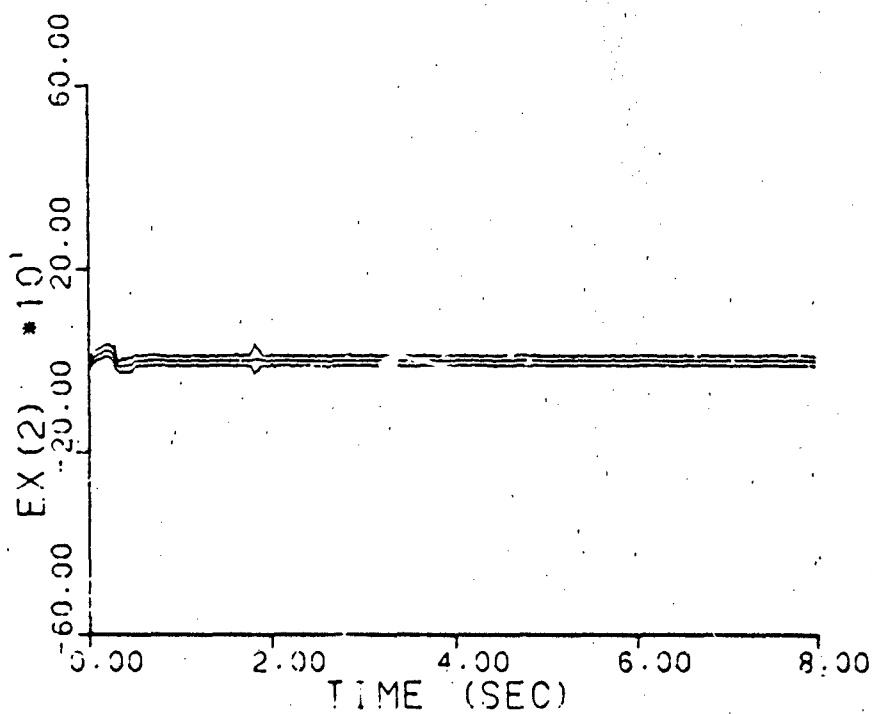
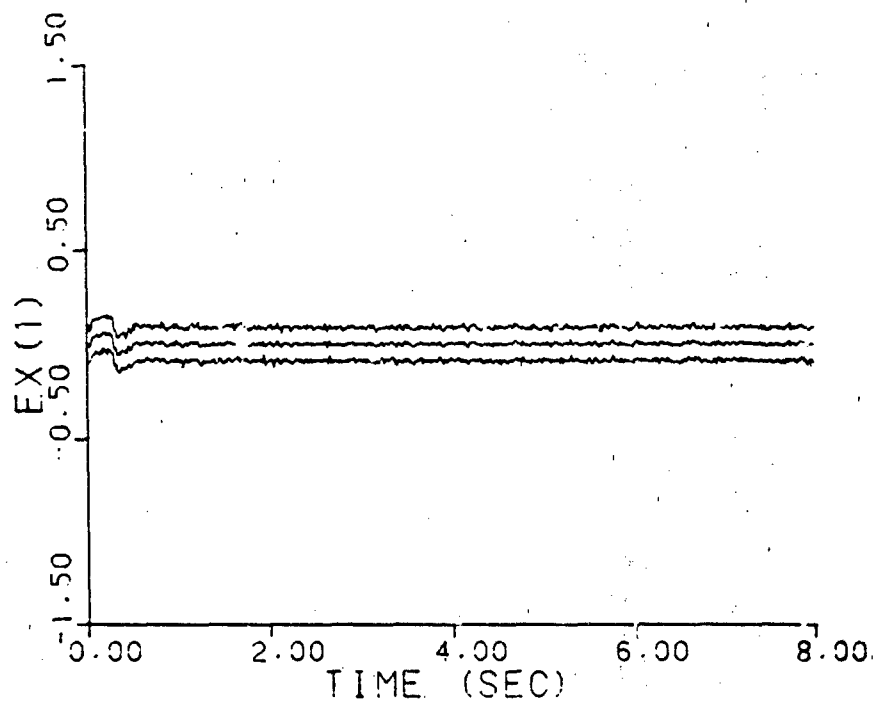


Figure A-12a. Probability Monitoring
 $\underline{a} = (1, 3)$

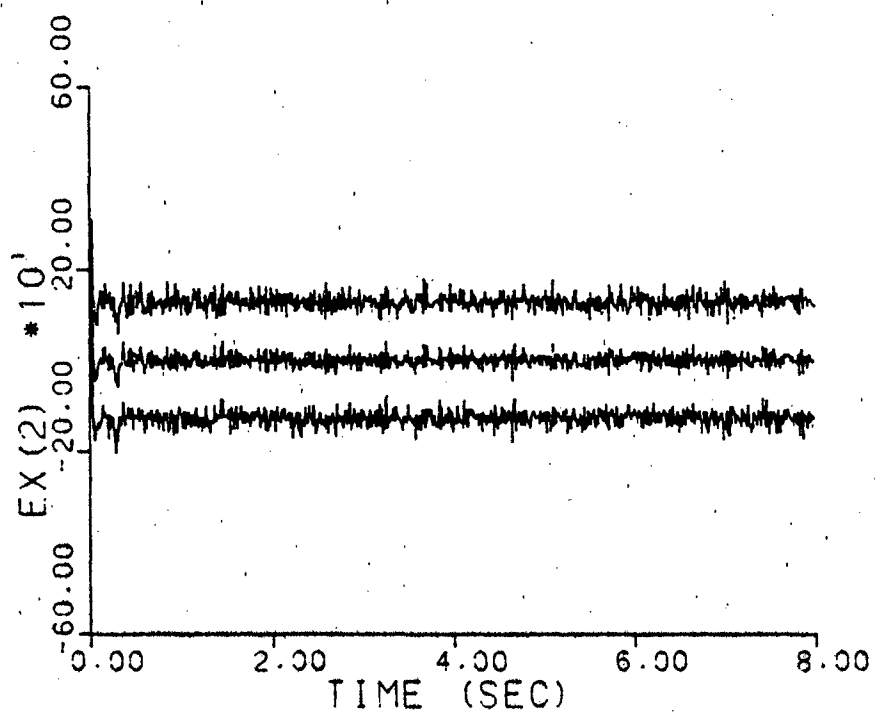
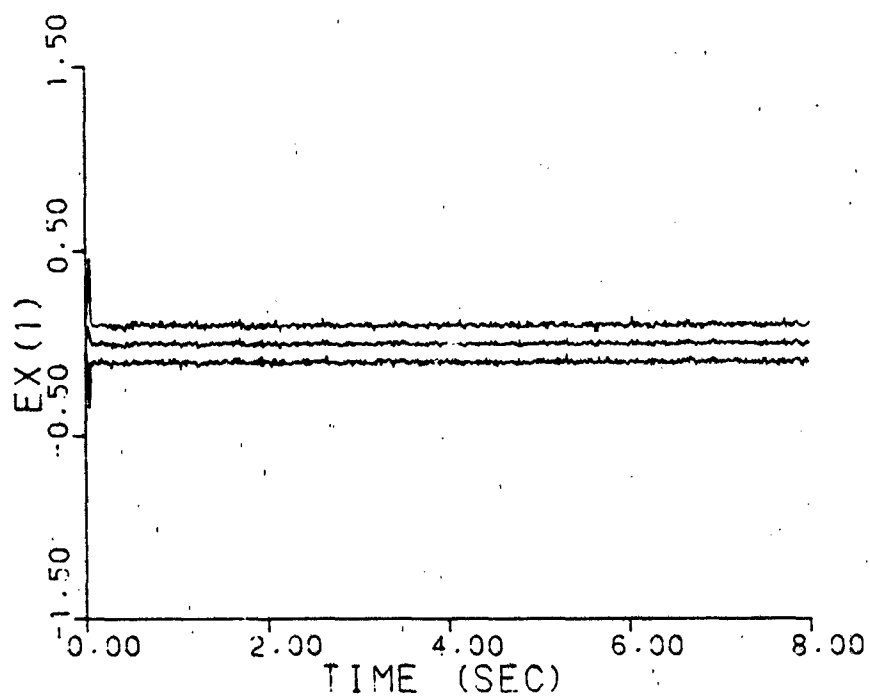


Figure A-12b. Probability Monitoring
 $\underline{a} = (2, 9)$

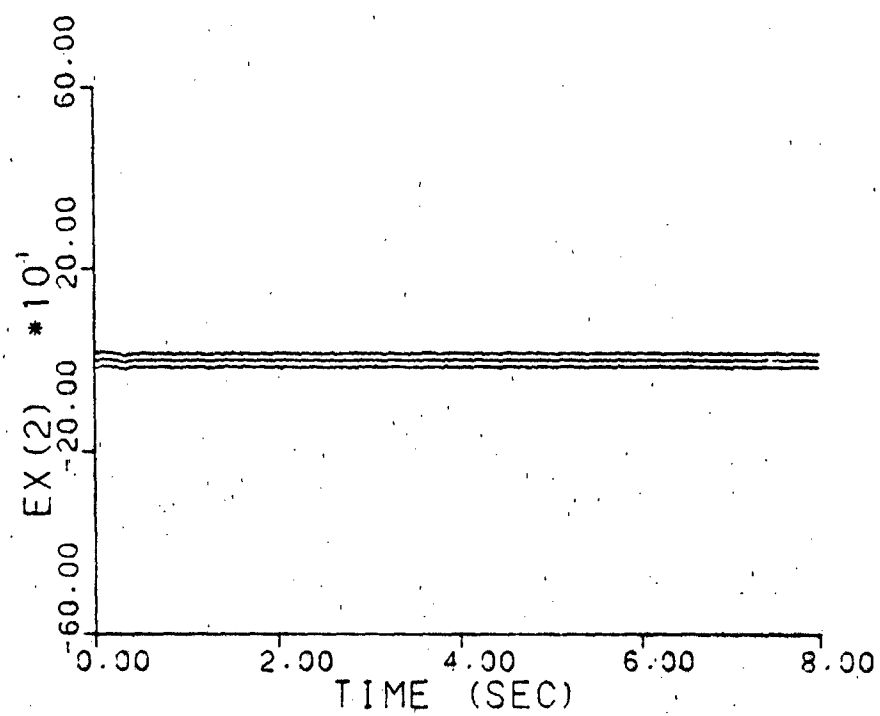
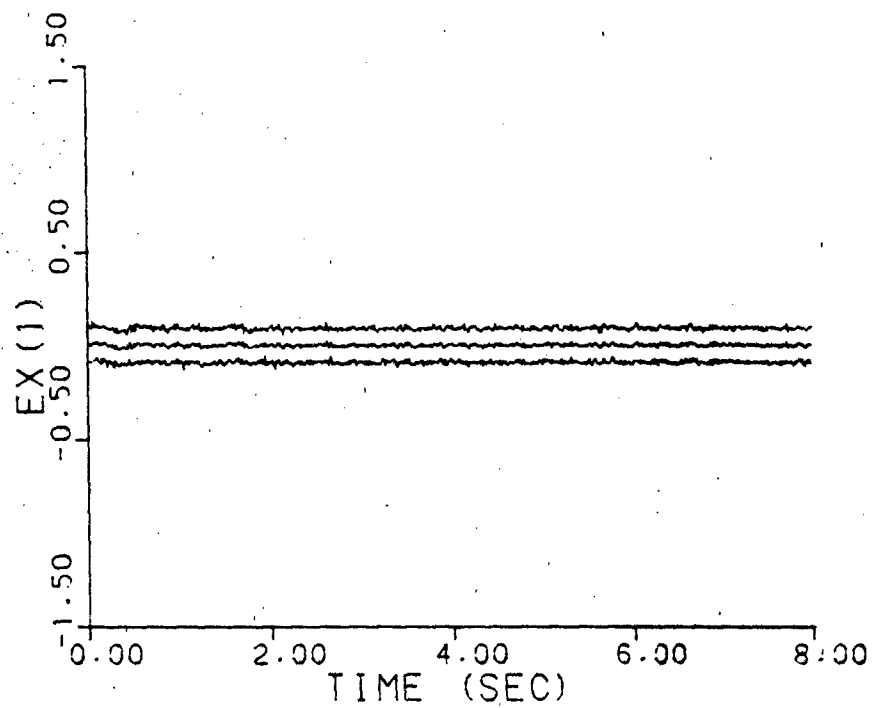


Figure A-12c. Probability Monitoring
 $\underline{a} = (5, 4)$

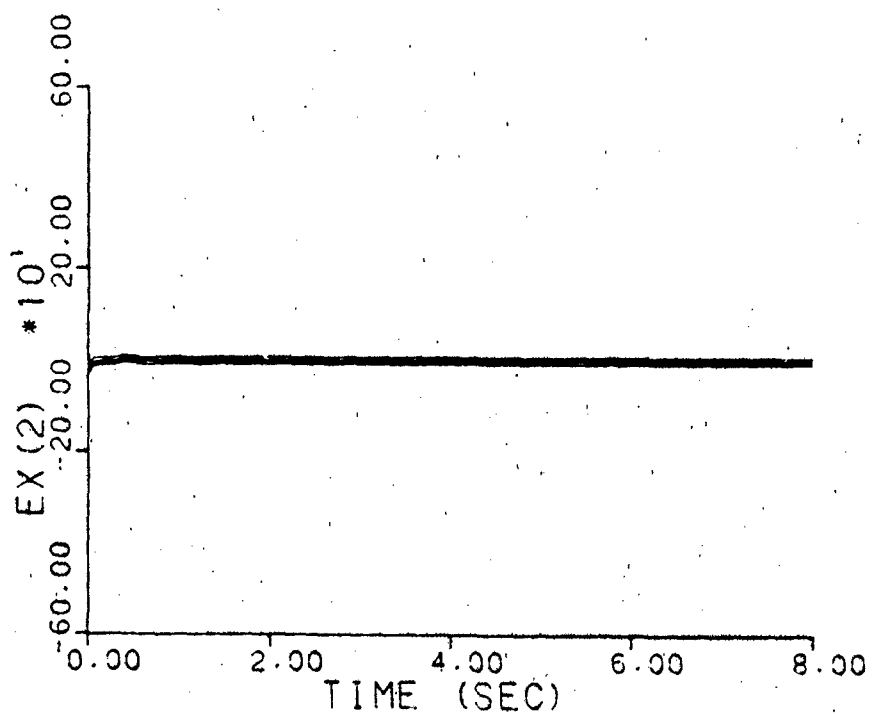
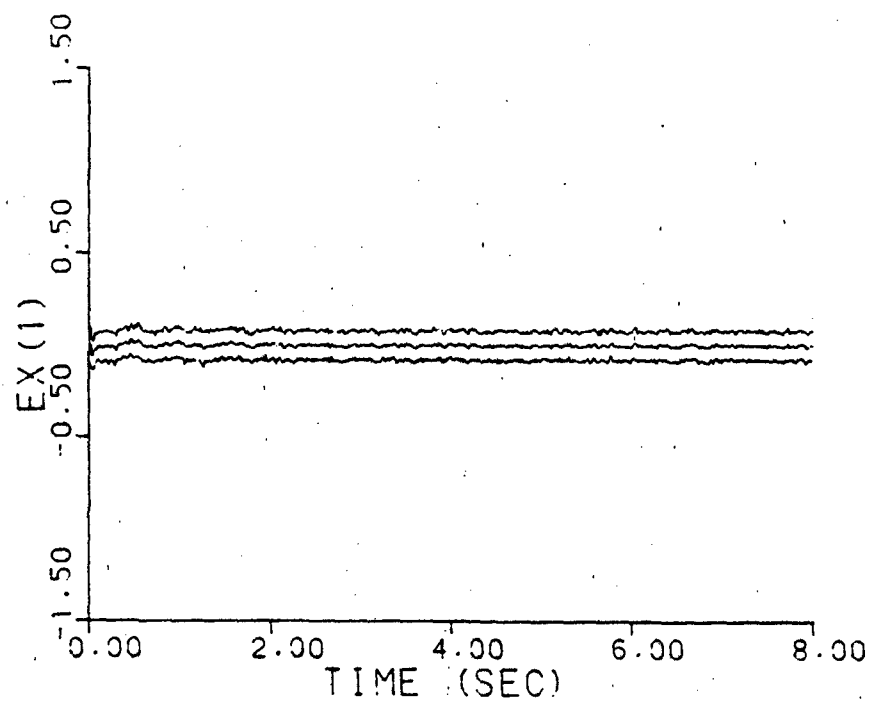


Figure A-12d. Probability Monitoring
 $\underline{a} = (9, 2)$

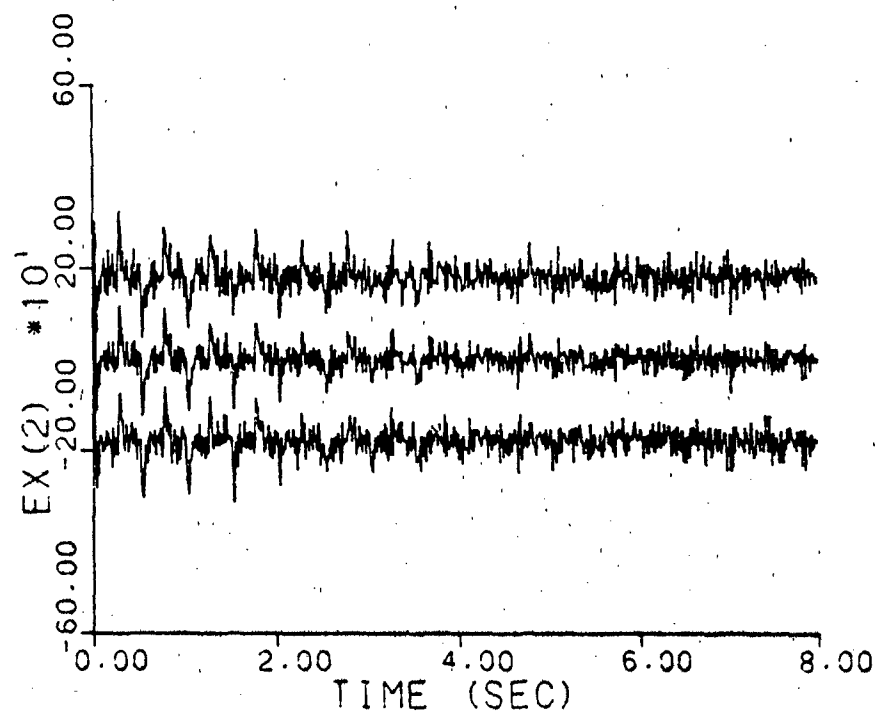
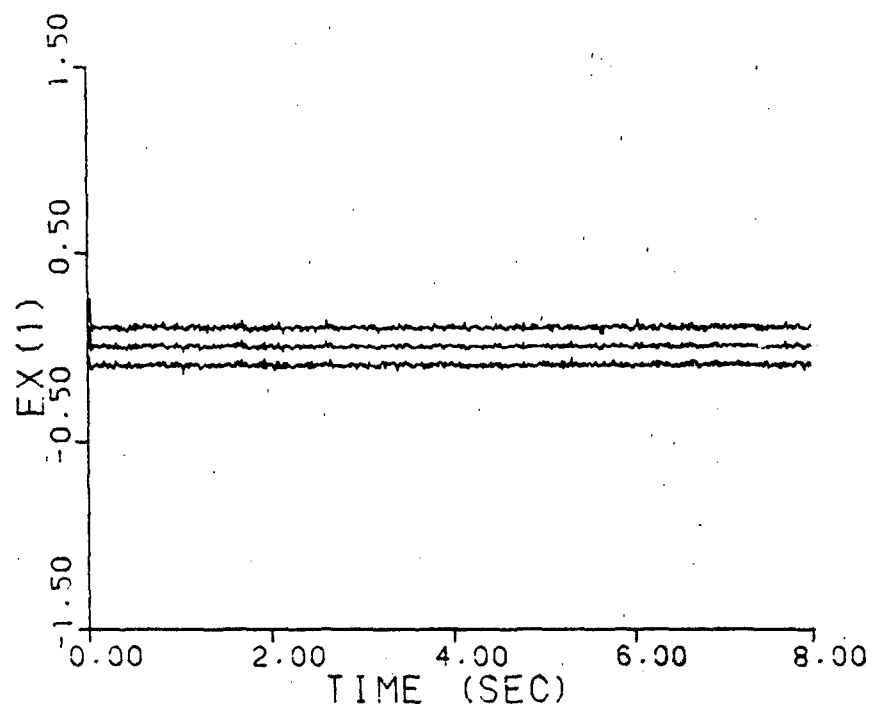


Figure A-12e. Probability Monitoring
 $\underline{a} = (10, 10)$

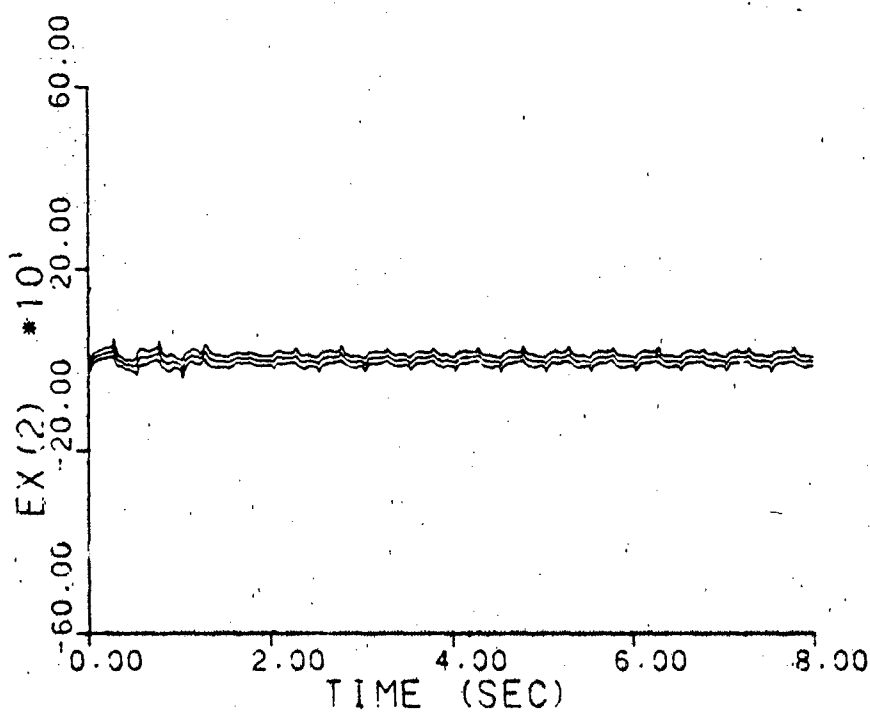
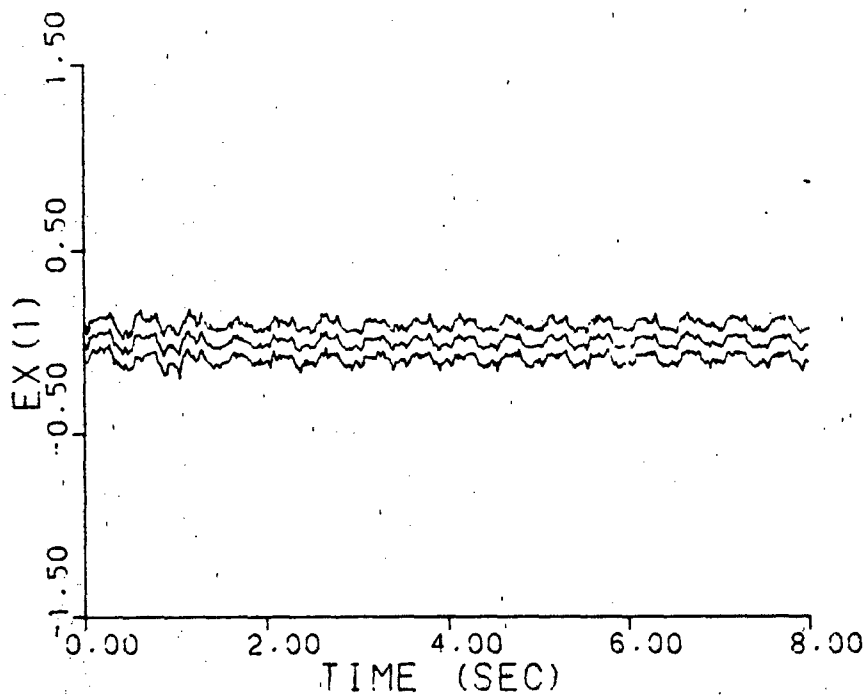


Figure A-12f. Probability Monitoring
 $\underline{a} = (0.07, 9.0)$

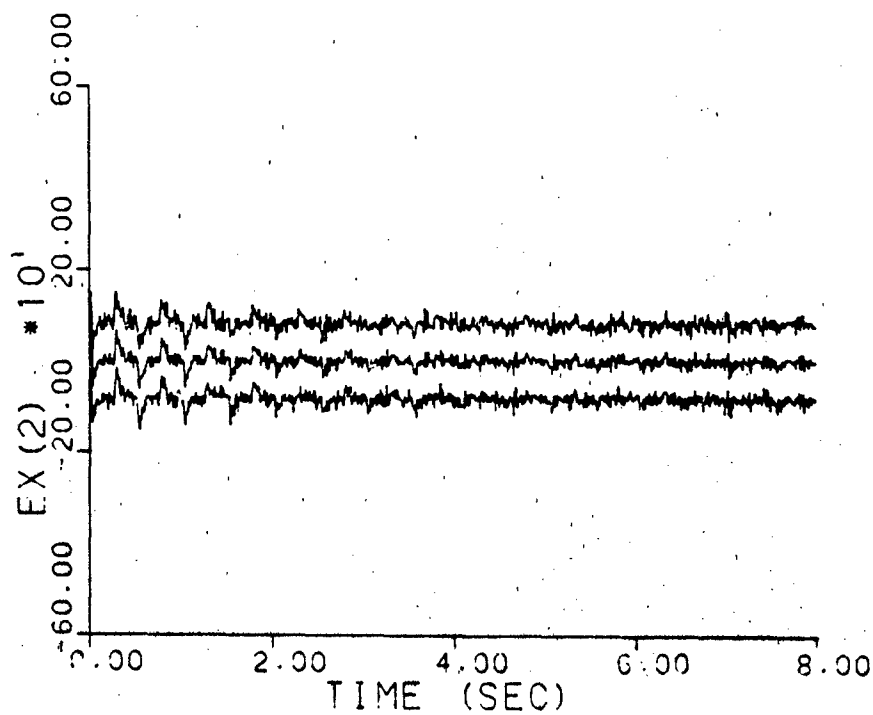
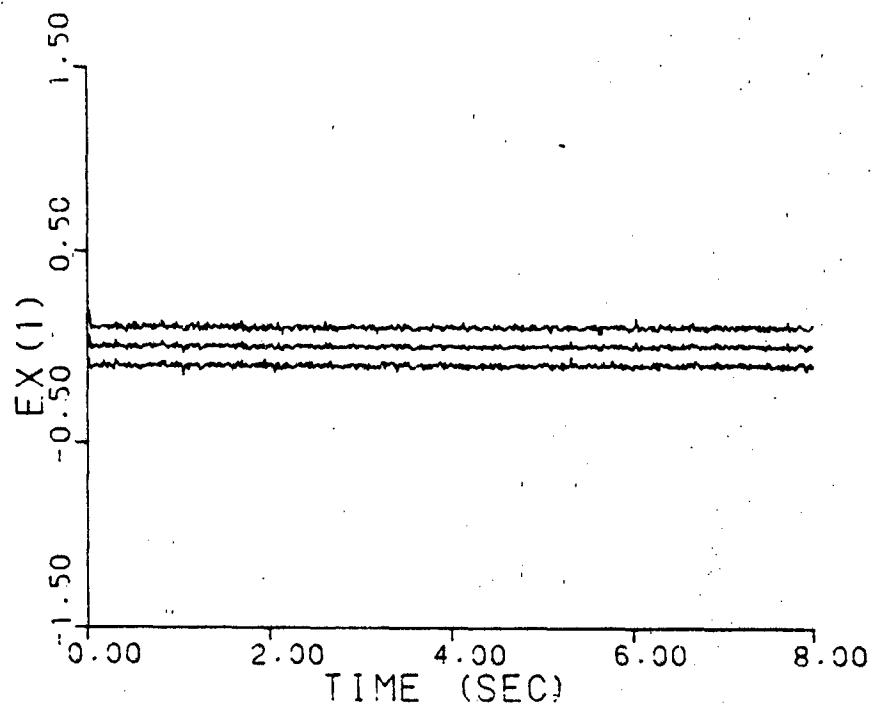


Figure A-12g. Probability Monitoring
 $\underline{a} = (0.93, 41.0)$

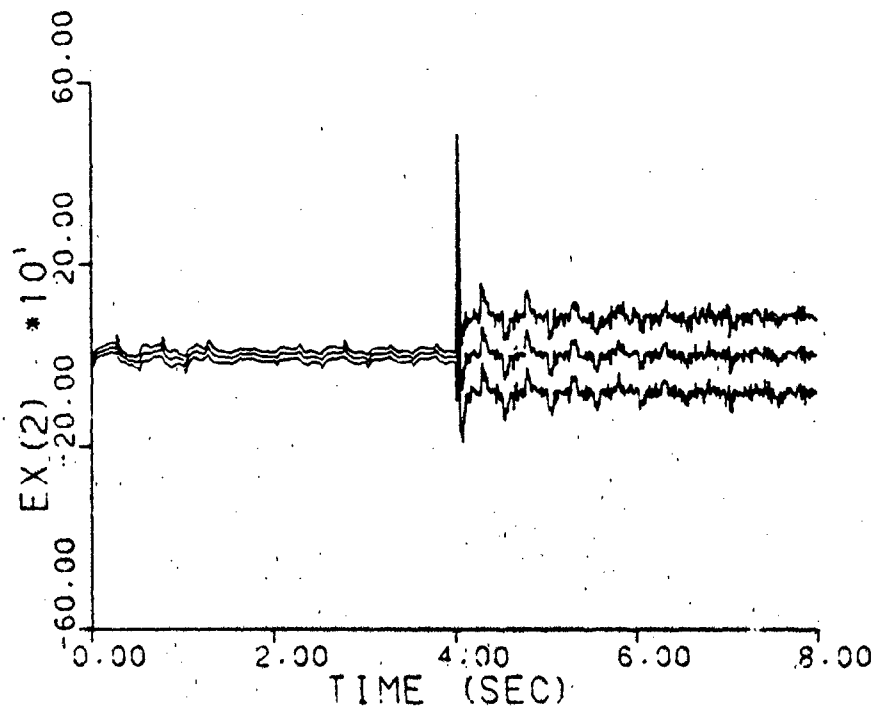
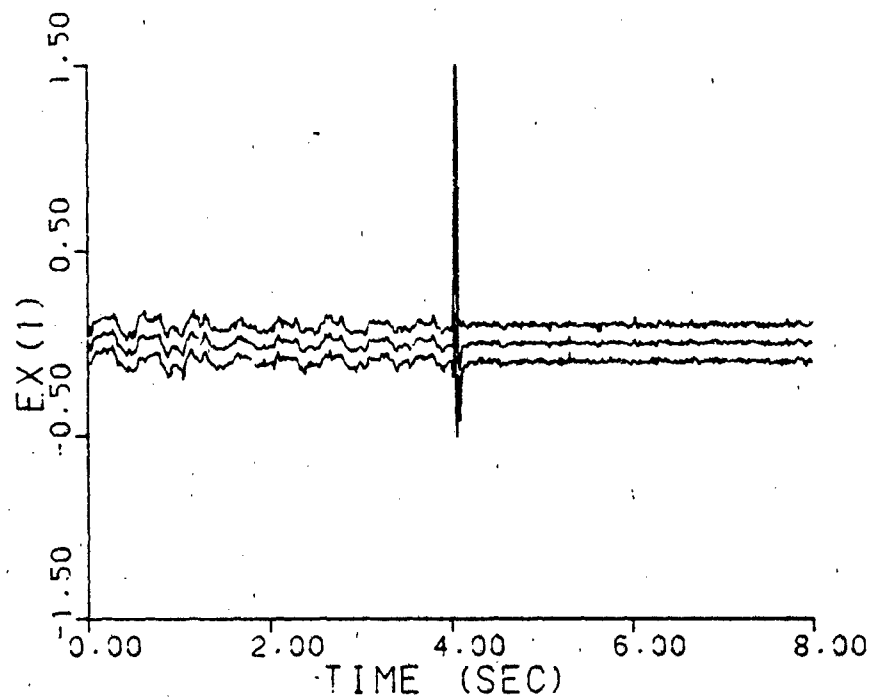


Figure A-12h. Probability Monitoring
a jumps: (0.07, 9.0) - (0.93, 41.0)

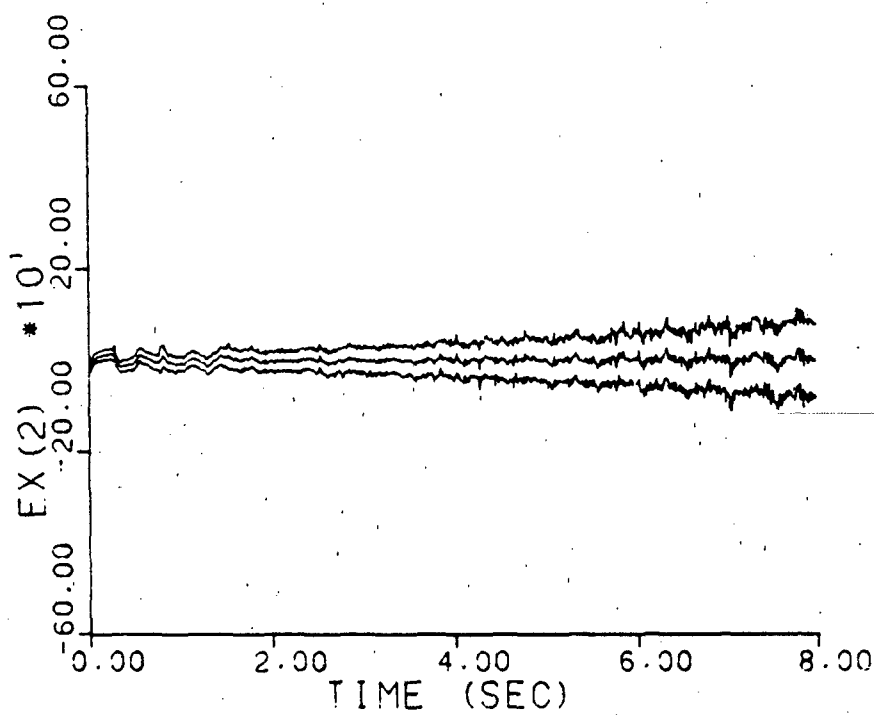
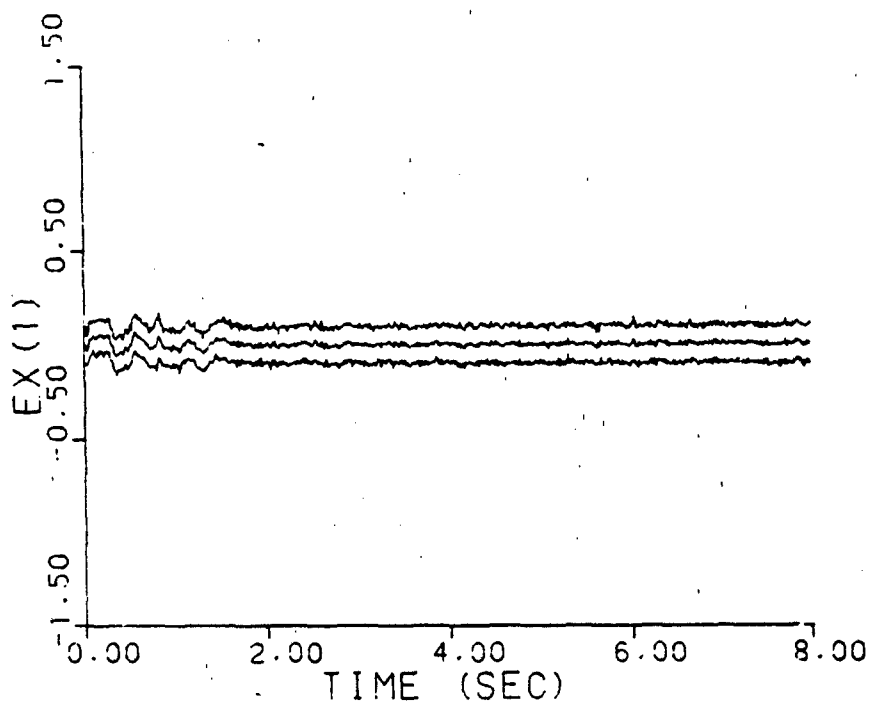


Figure A-12i. Probability Monitoring
a varies: (0.07, 9.0) - (0.93, 41.0)

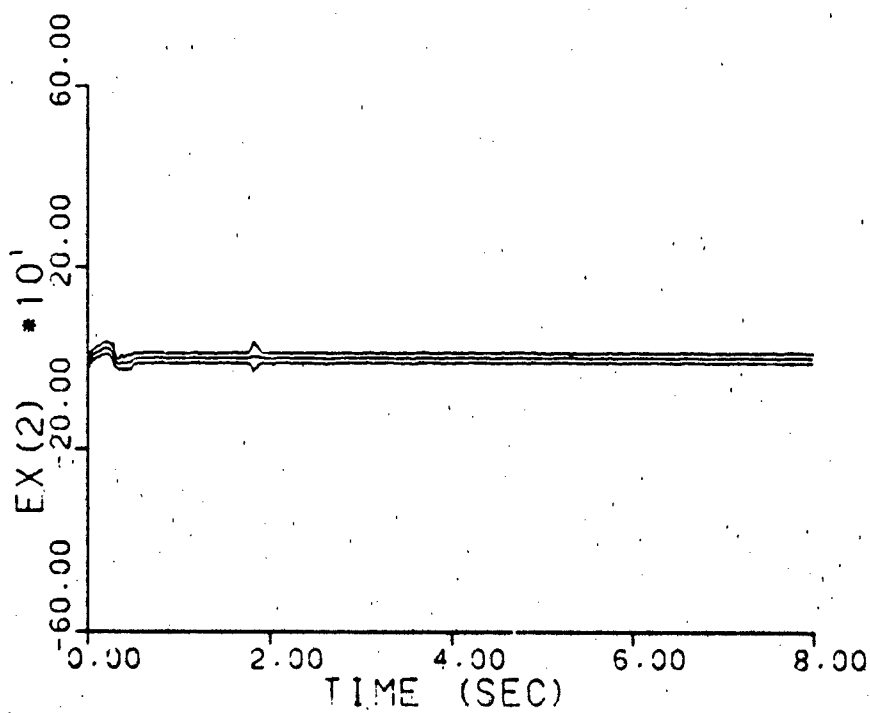
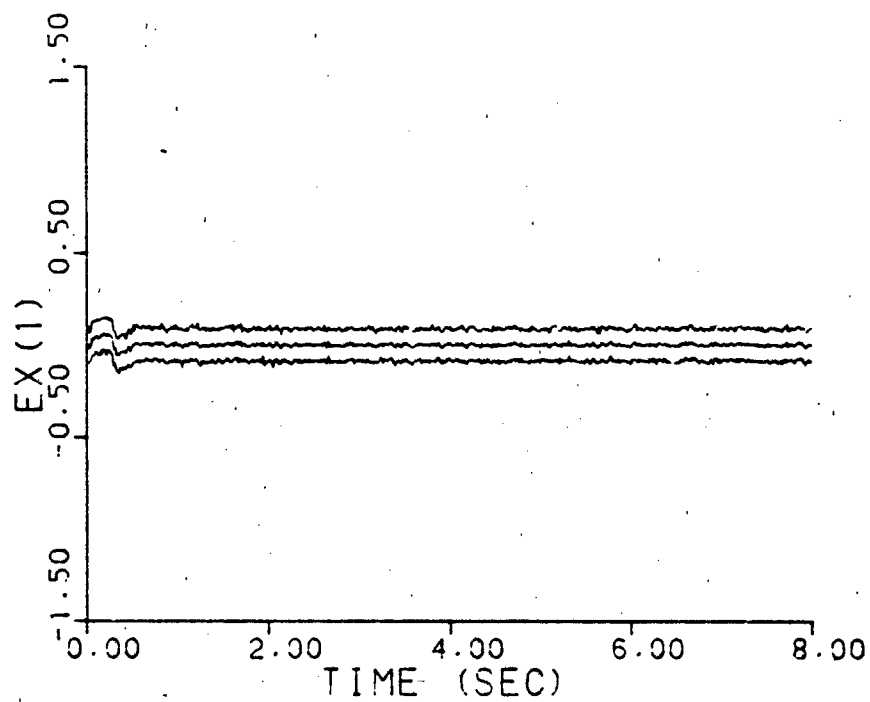


Figure A-13a. Probability Monitoring
 $\underline{a} = (1, 3)$, warm-up

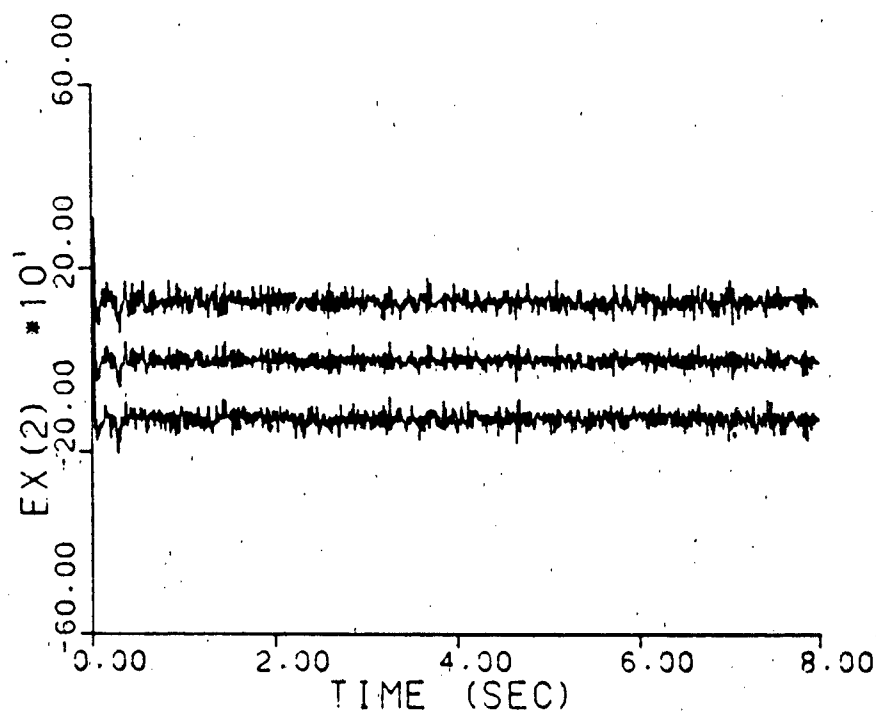
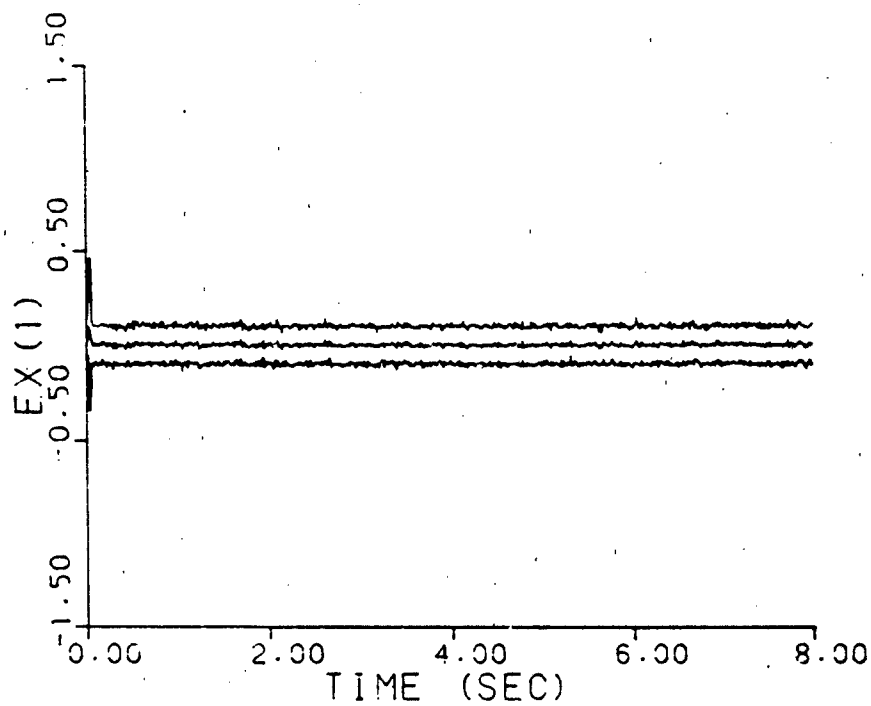


Figure A-13b. Probability Monitoring
 $\underline{a} = (2, 9)$, warm-up

THIS
PAGE
IS
MISSING
IN
ORIGINAL
DOCUMENT

A-13C.

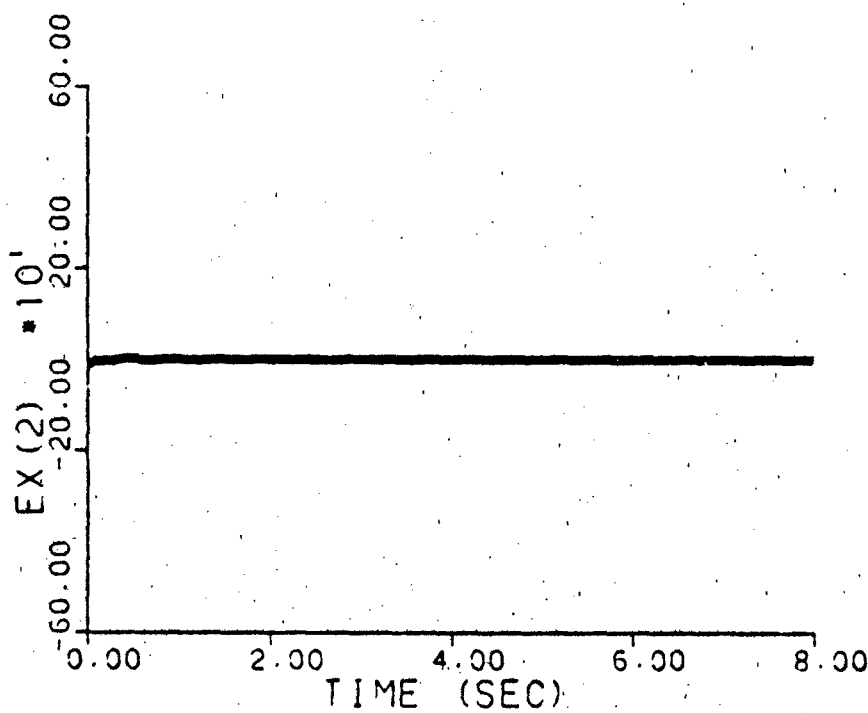
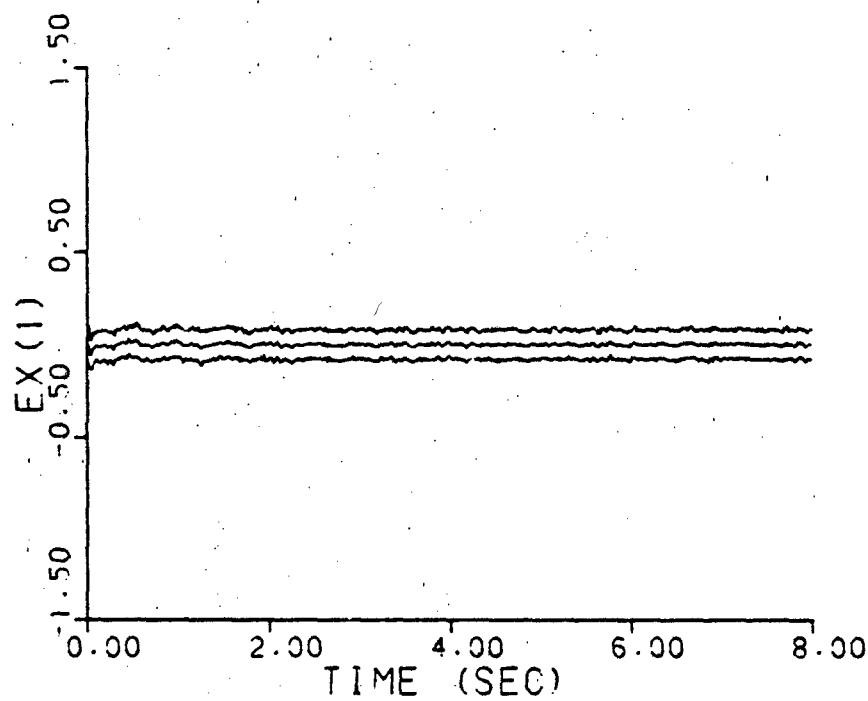


Figure A-13d. Probability Monitoring
 $\underline{a} = (9, 2)$, warm-up

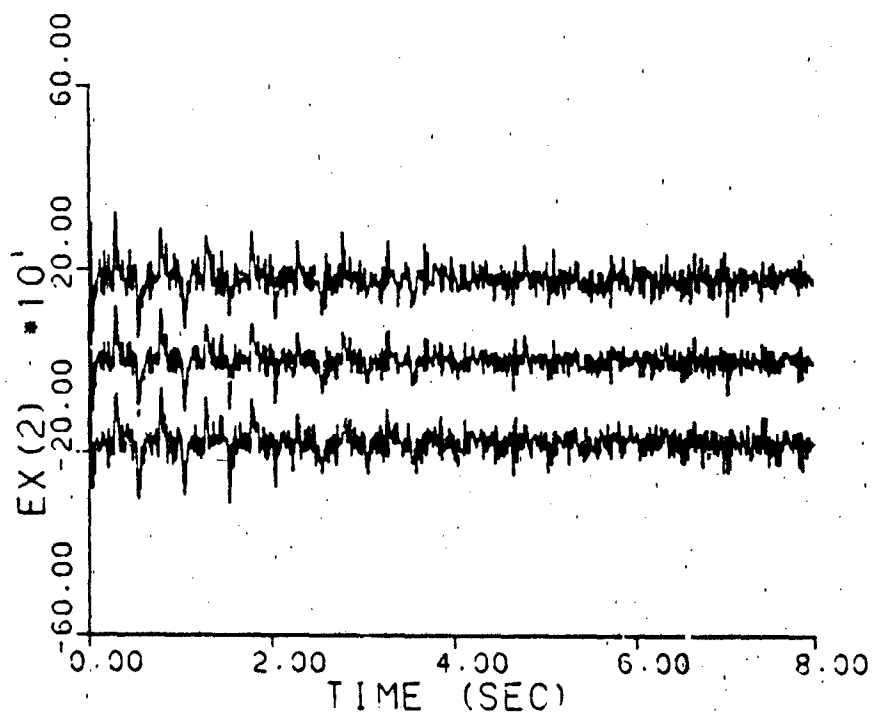
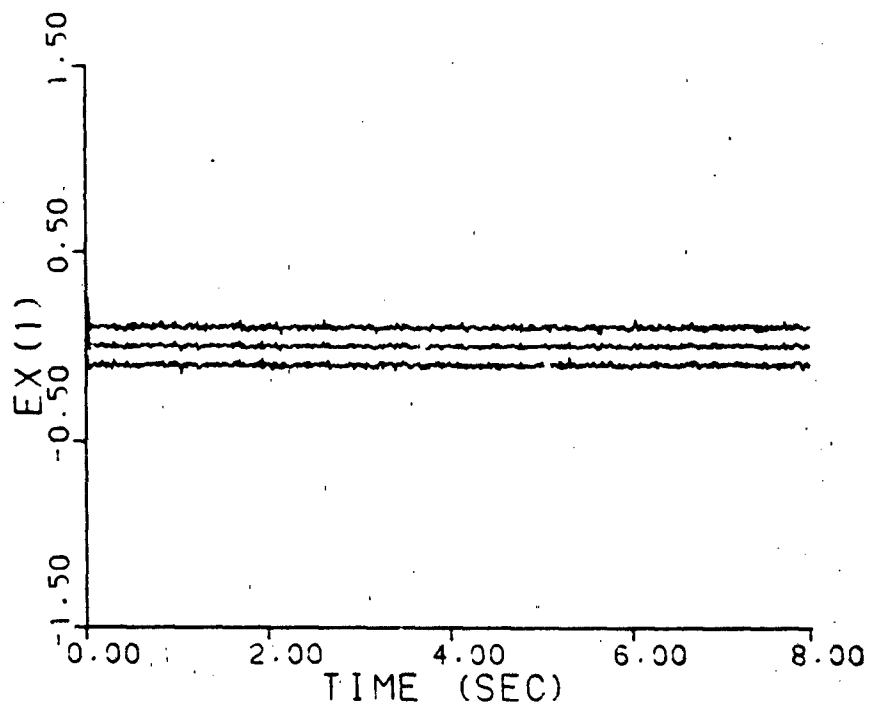


Figure A-13e. Probability Monitoring
 $\underline{a} = (10, 10)$, warm-up

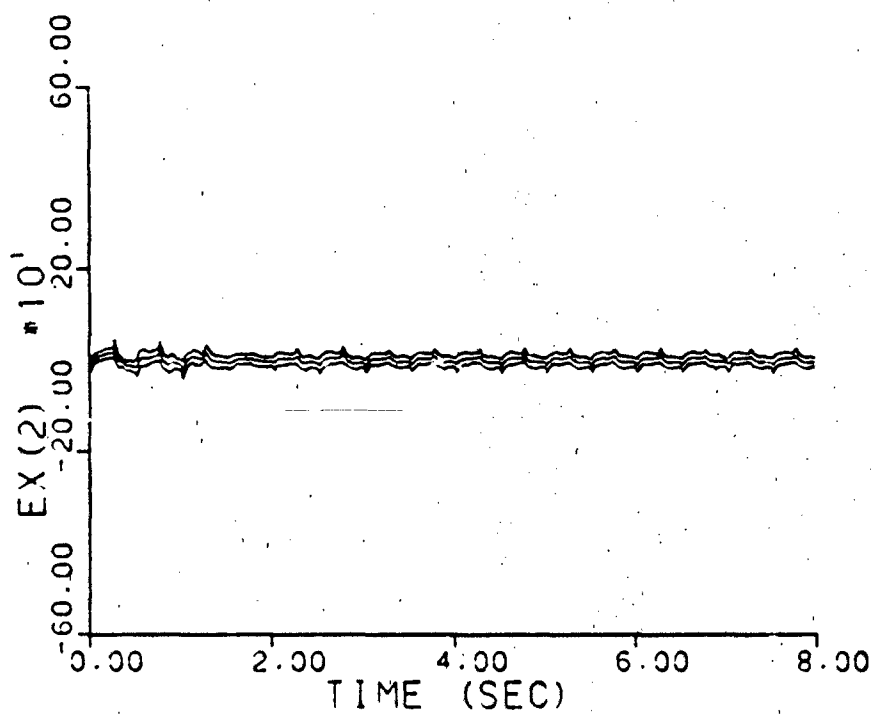
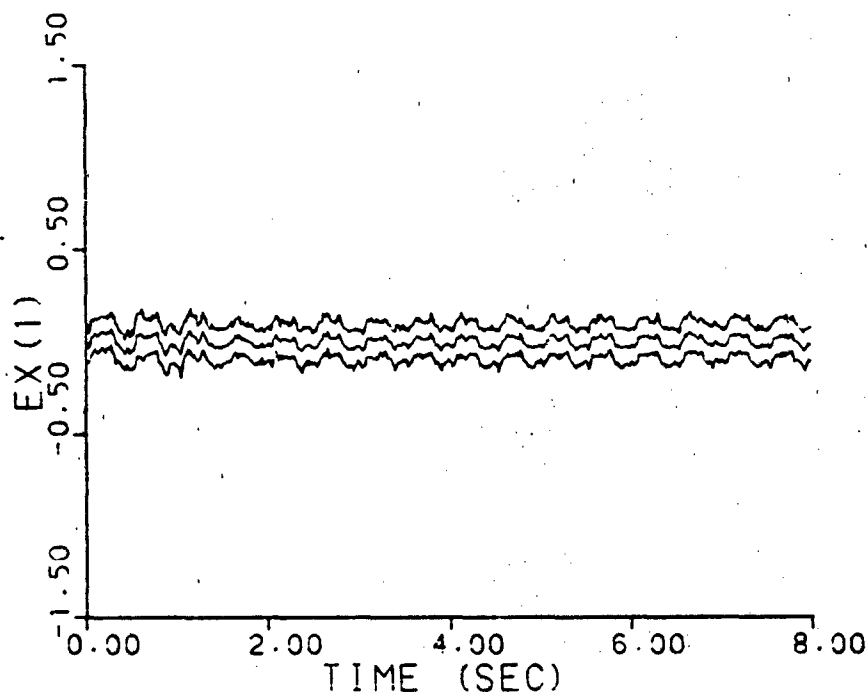


Figure A-13f. Probability Monitoring
 $\underline{a} = (0.07, 9.0)$, warm-up

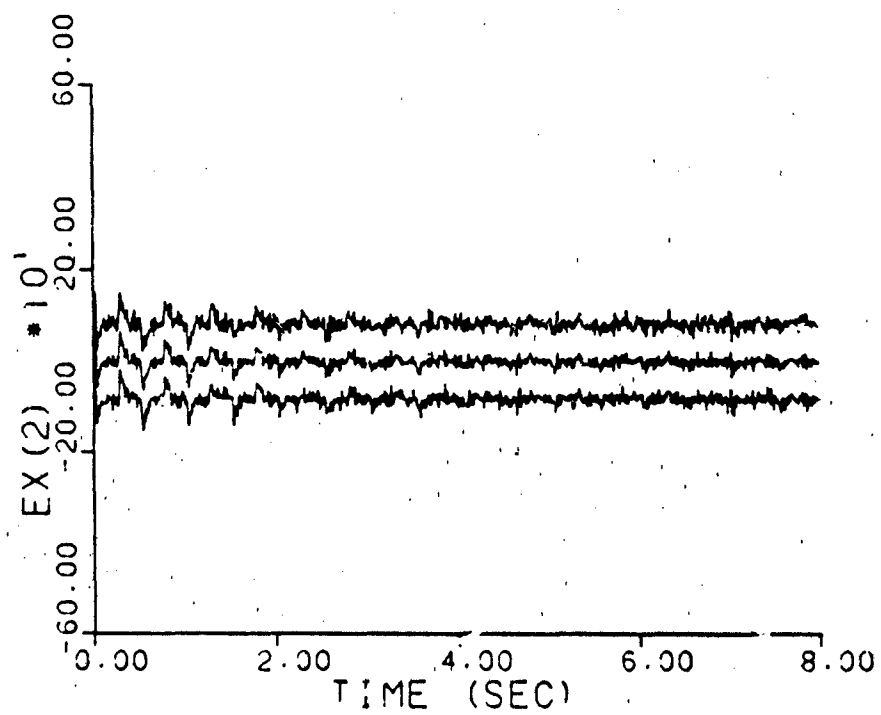
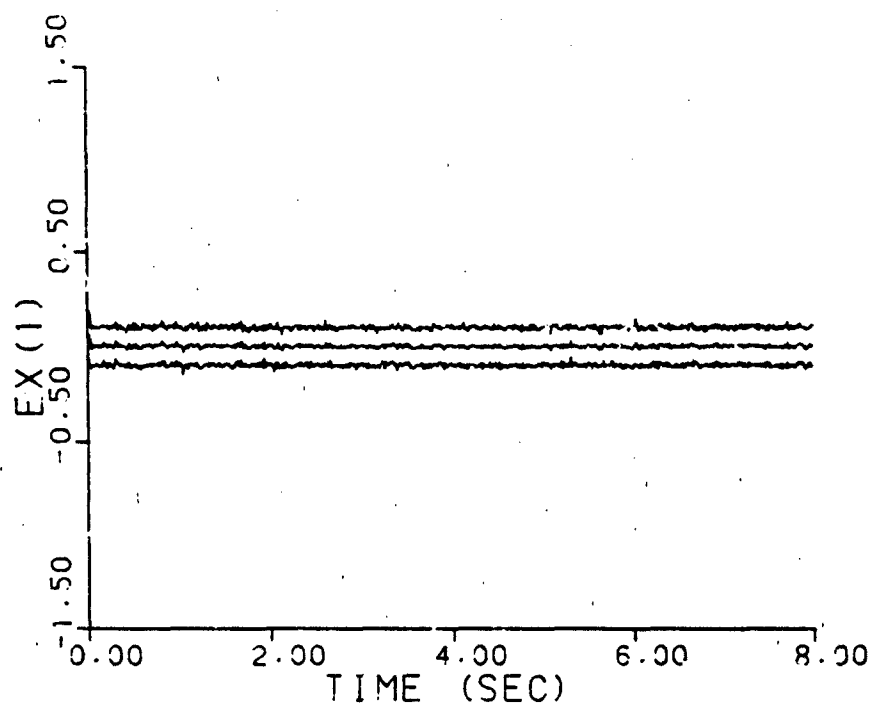


Figure A-13g. Probability Monitoring
 $\underline{a} = (0.93, 41.0)$, warm-up

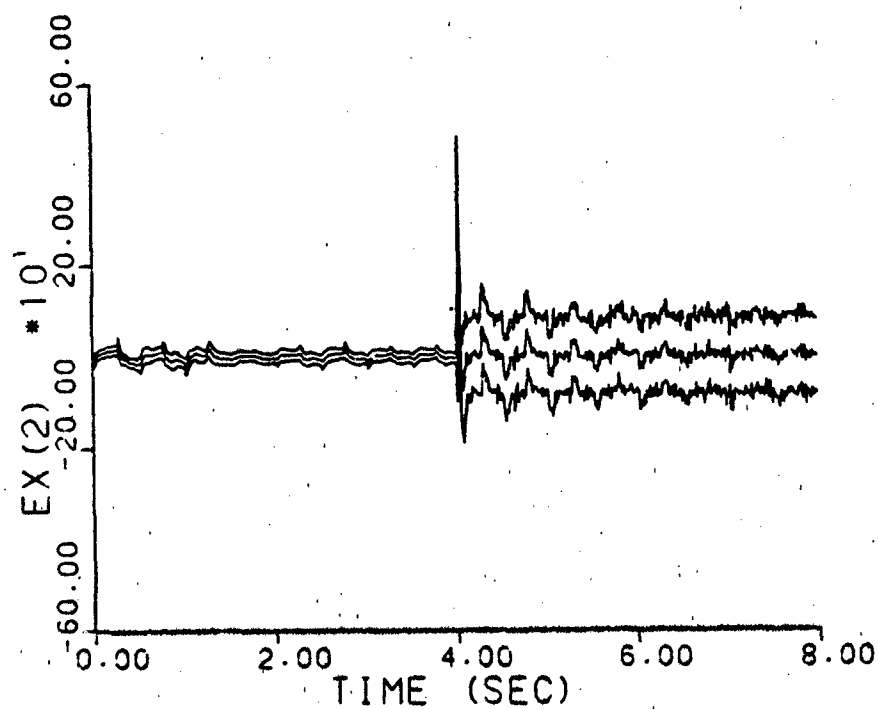
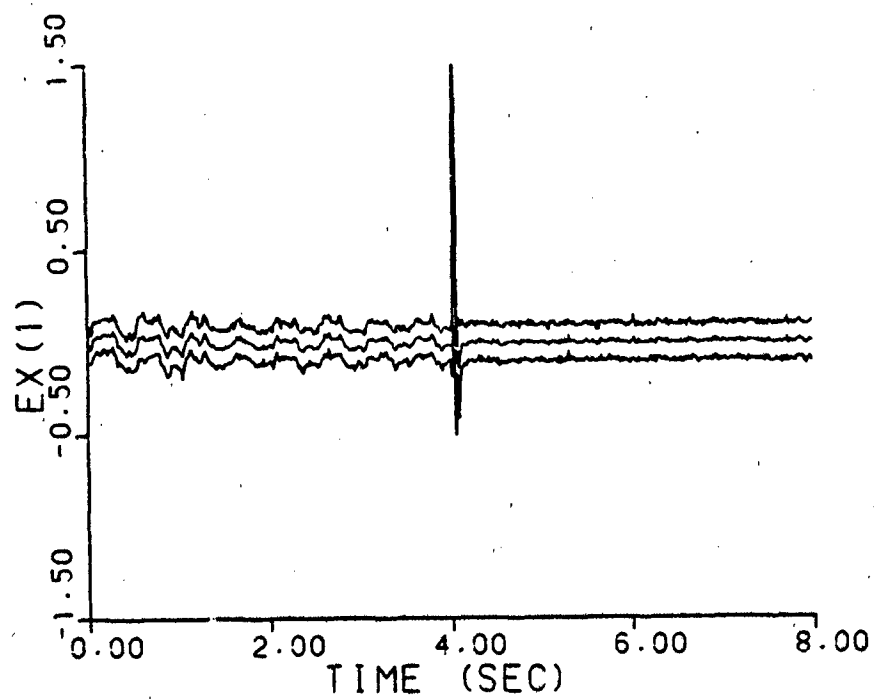


Figure A-13h. Probability Monitoring
a jumps: (0.07, 9.0) - (0.93, 41.0)
warm-up

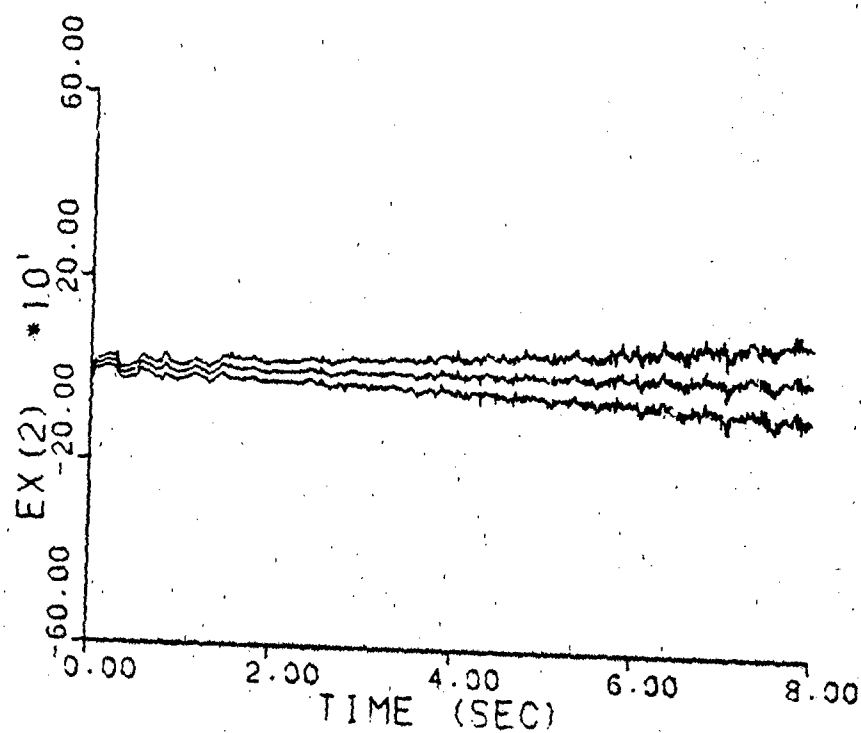
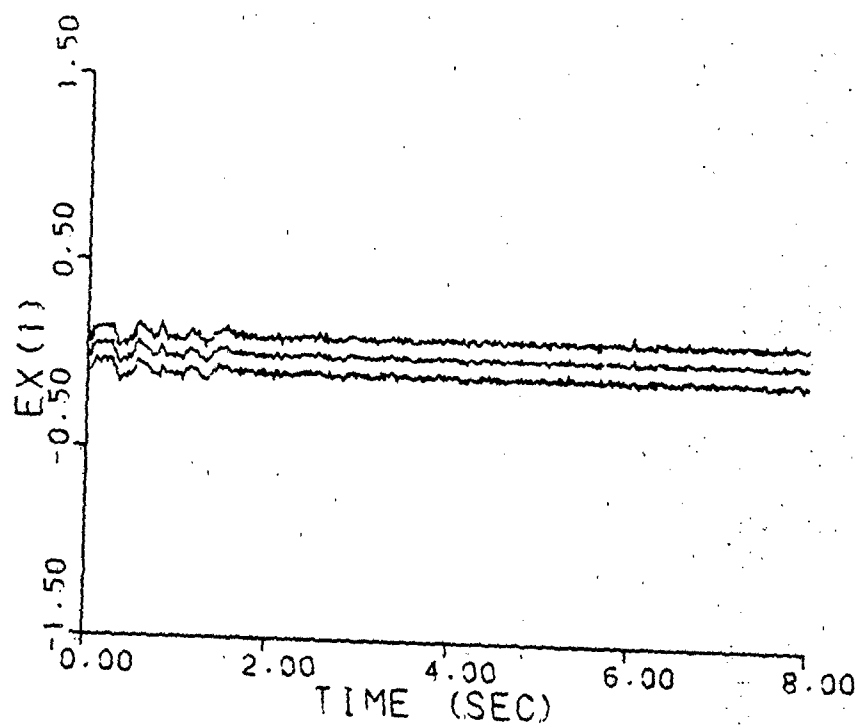


Figure A-13i. Probability Monitoring
a varies: (0.07, 9.0) - (0.93, 41.0)
 warm-up

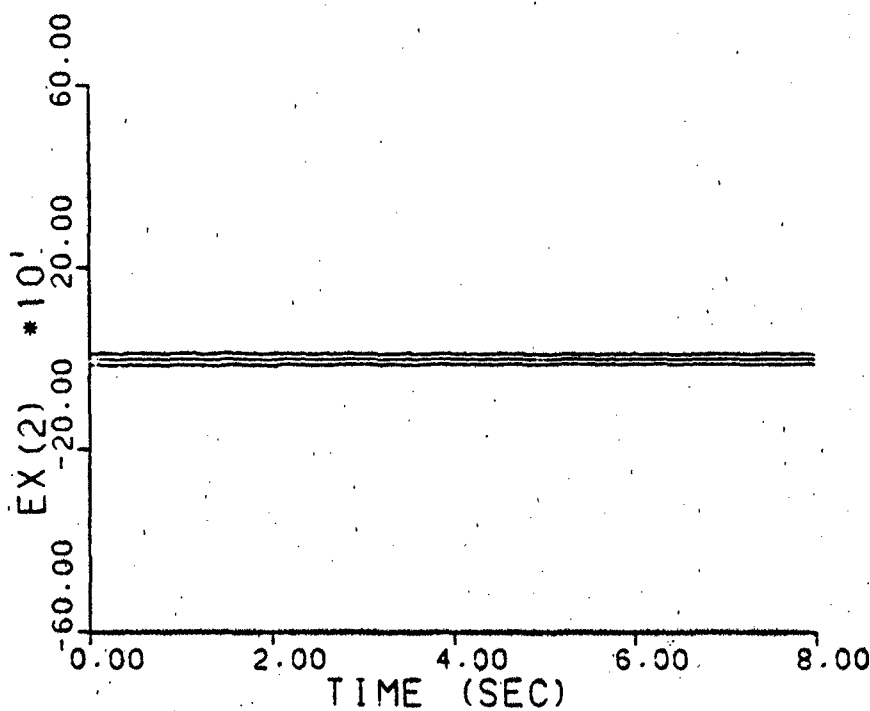
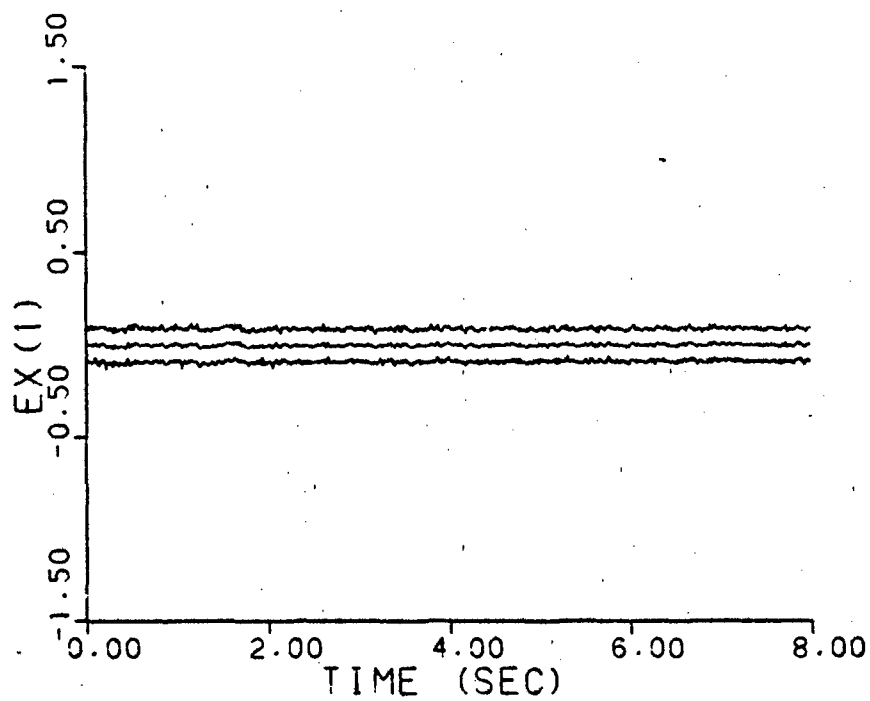


Figure A-14a. Parameter Covariance Monitoring
 $\underline{a} = (1, 3)$, no dither

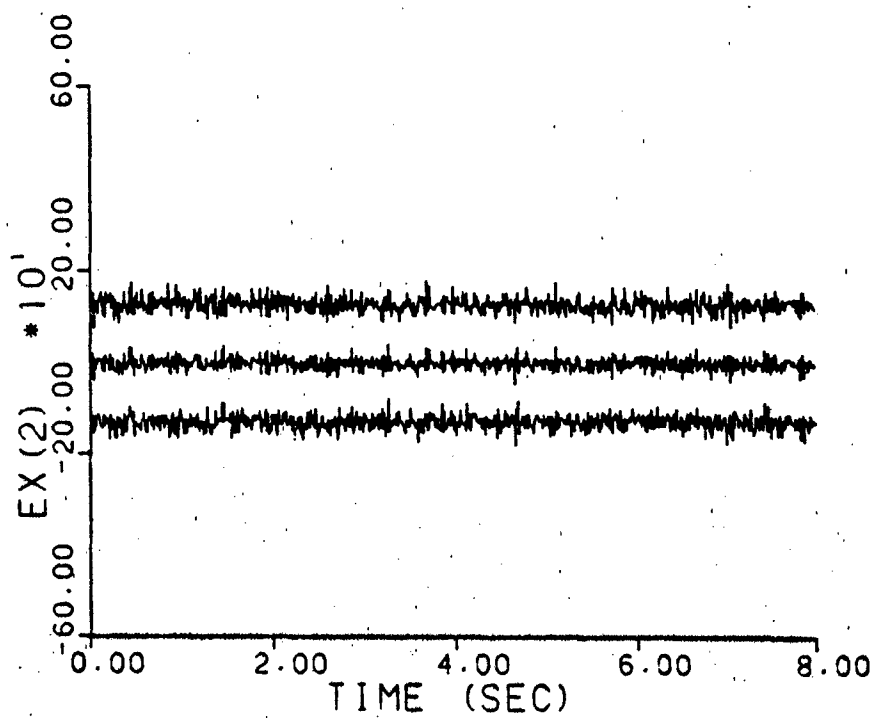
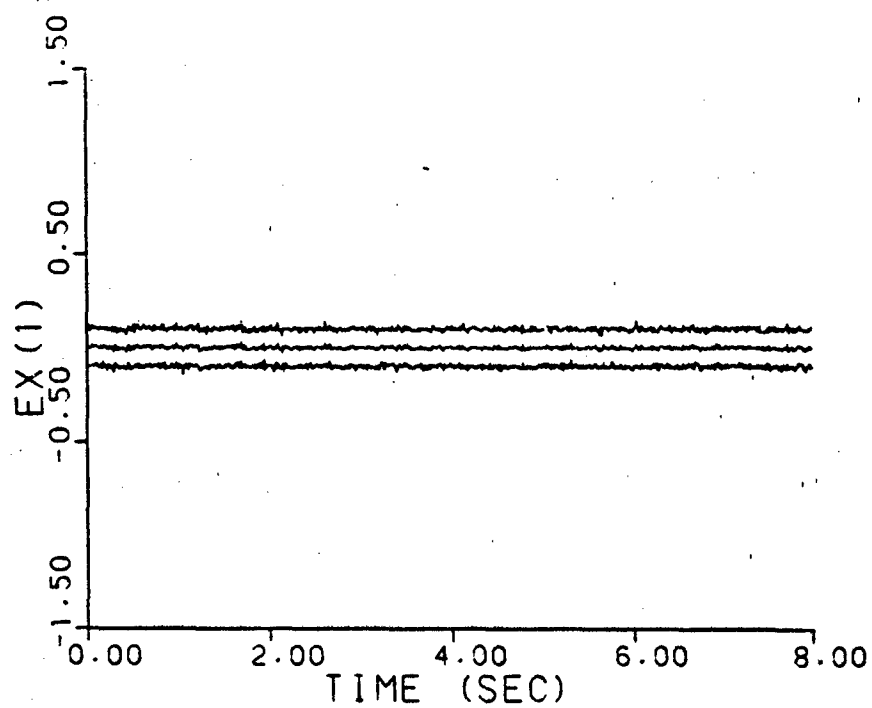


Figure A-14b. Parameter Covariance Monitoring
 $\underline{a} = (2, 9)$, no dither

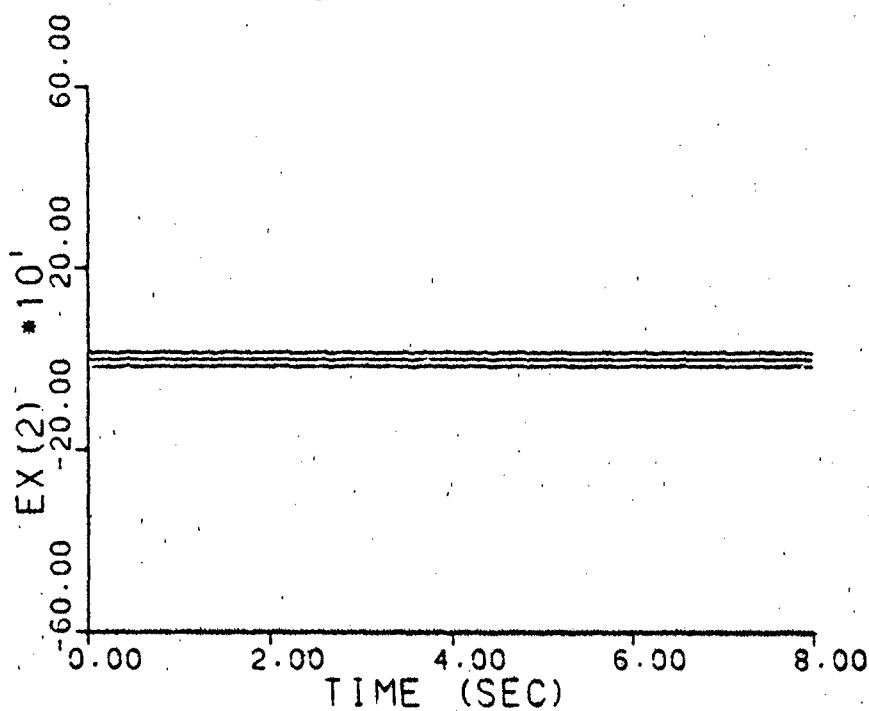
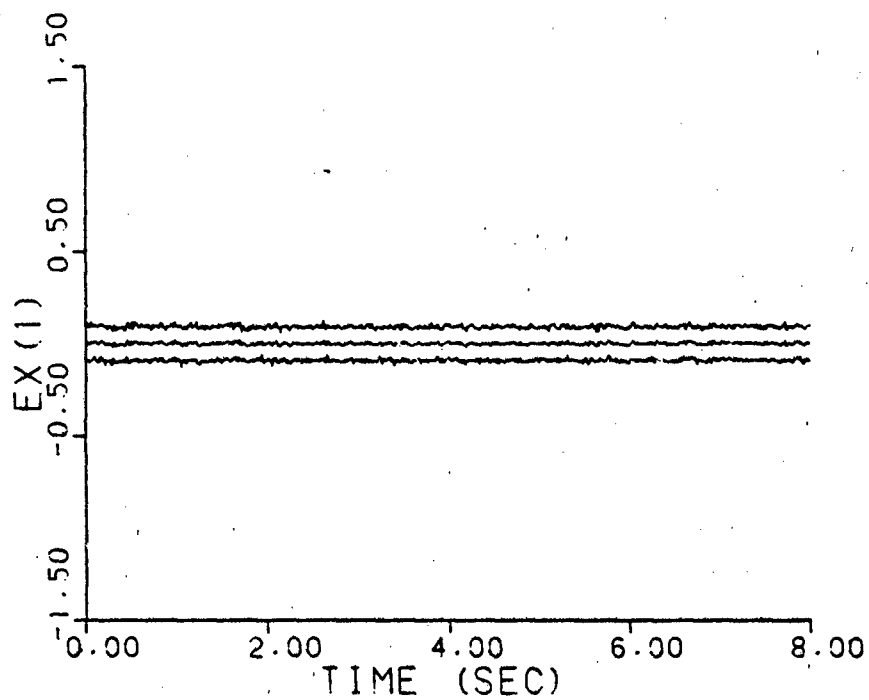


Figure A-14c. Parameter Covariance Monitoring
 $\underline{a} = (5, 4)$, no dither

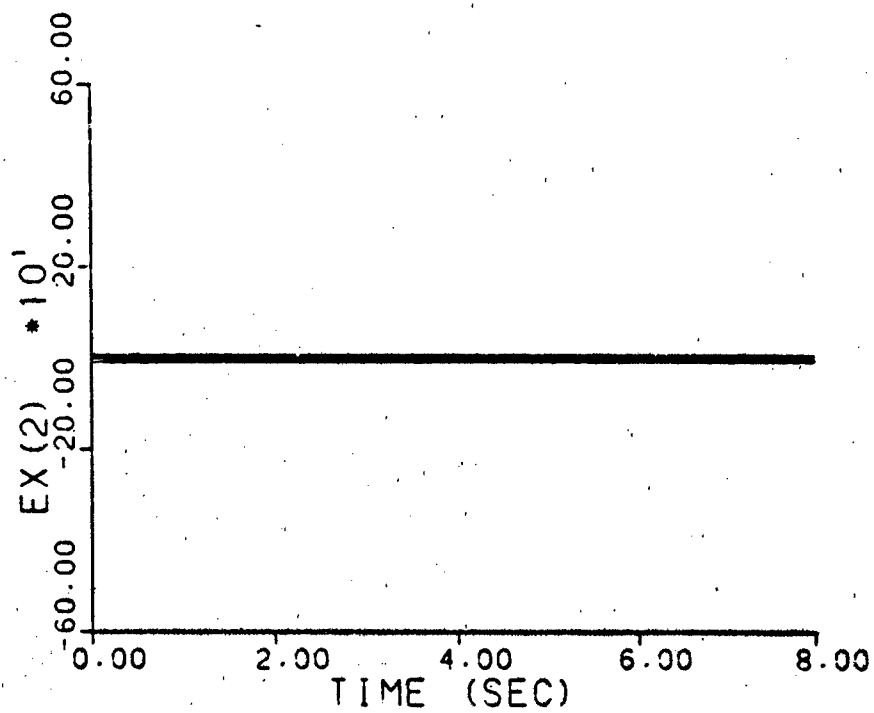
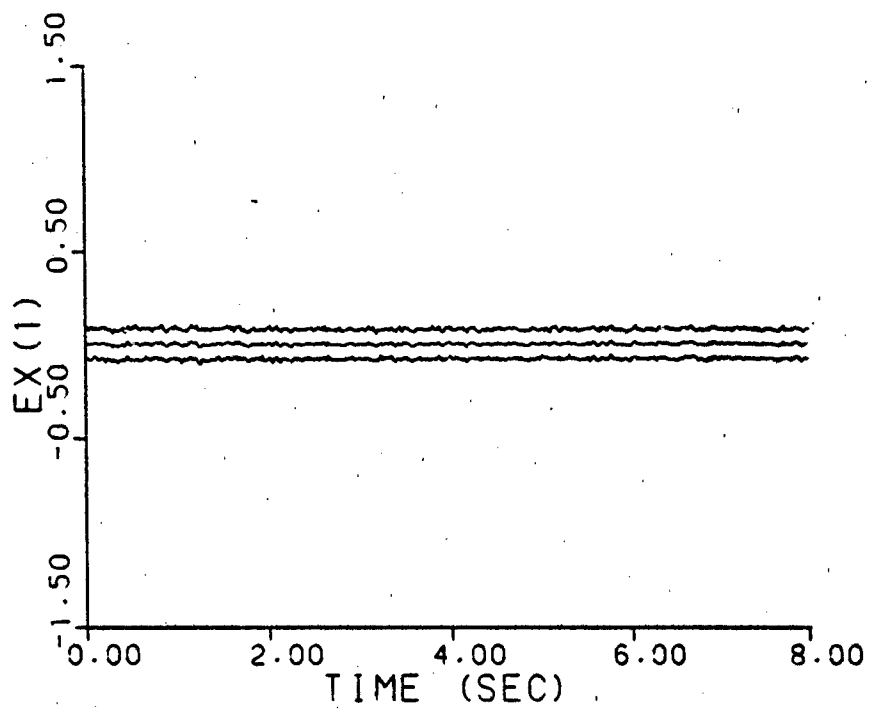


Figure A-14d. Parameter Covariance Monitoring
 $\underline{a} = (9, 2)$, no dither

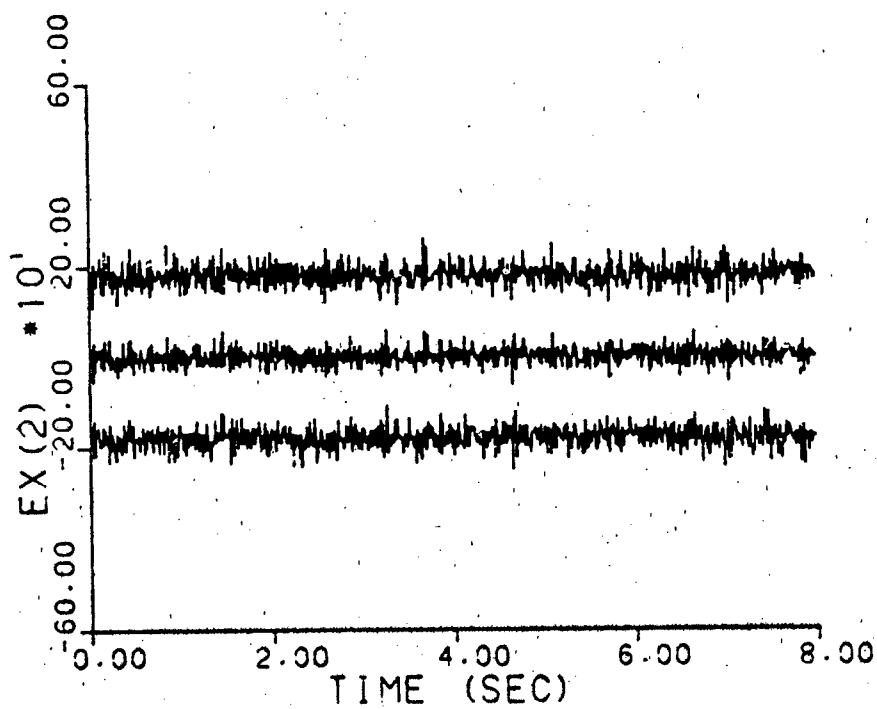
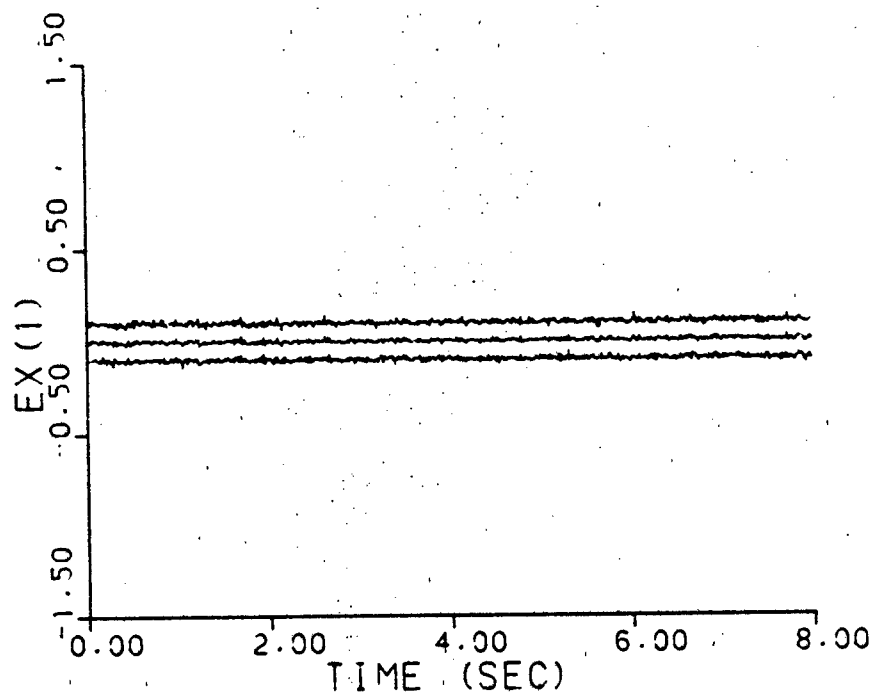


Figure A-14e. Parameter Covariance Monitoring
 $\underline{a} = (10, 10)$, no dither

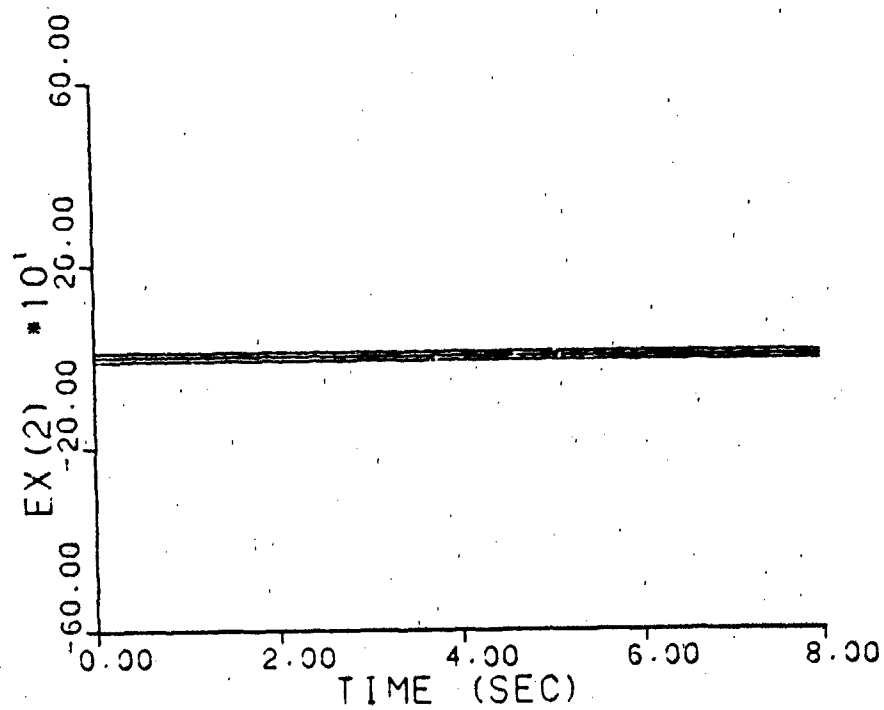
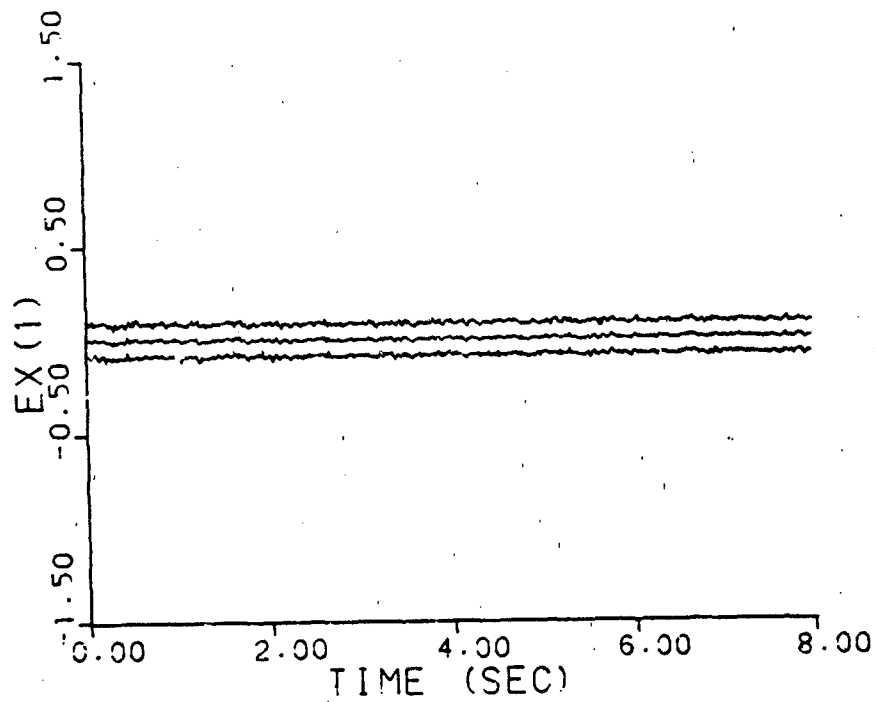


Figure A-14f. Parameter Covariance Monitoring
 $\underline{a} = (0.07, 9.0)$, no dither

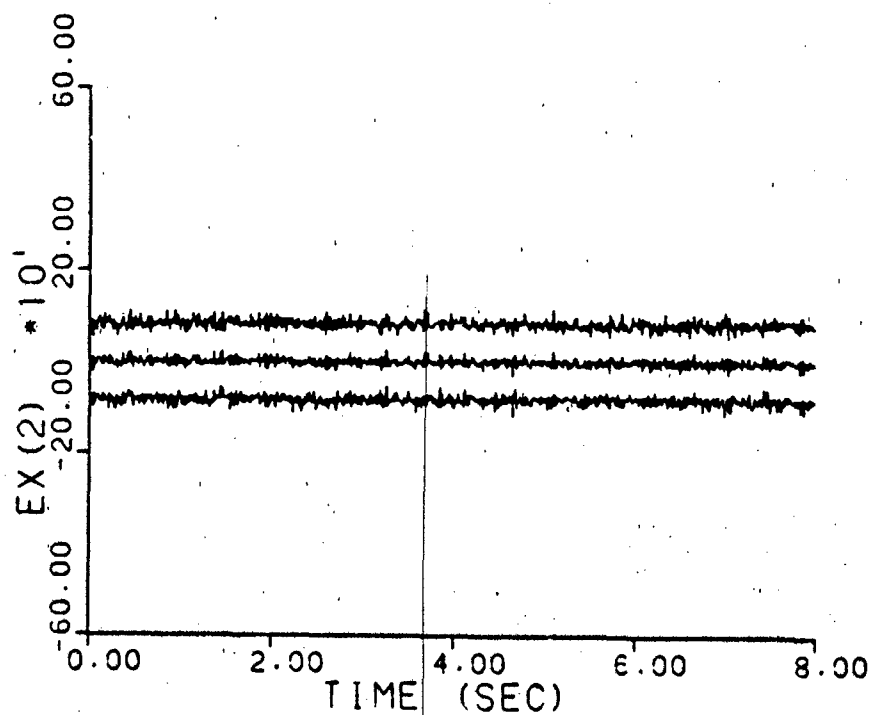
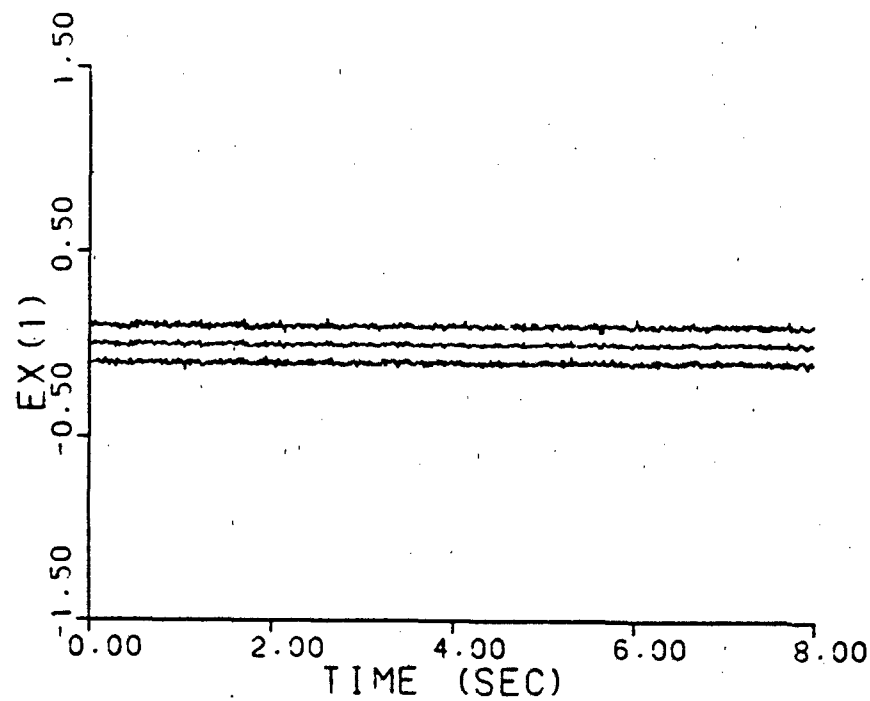


Figure A-14g. Parameter Covariance Monitoring
 $\underline{a} = (0.93, 41.0)$, no dither

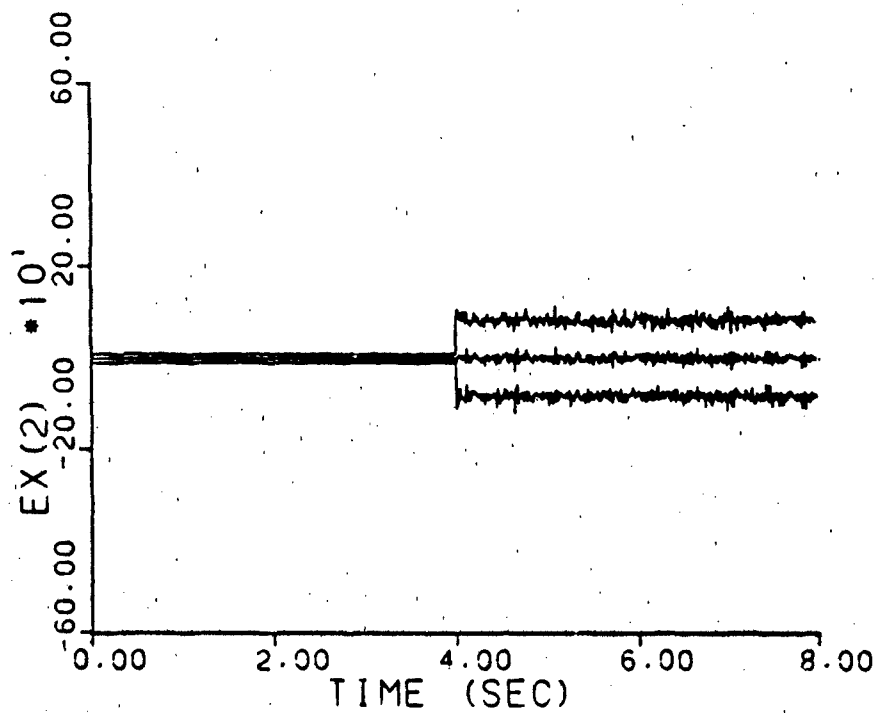
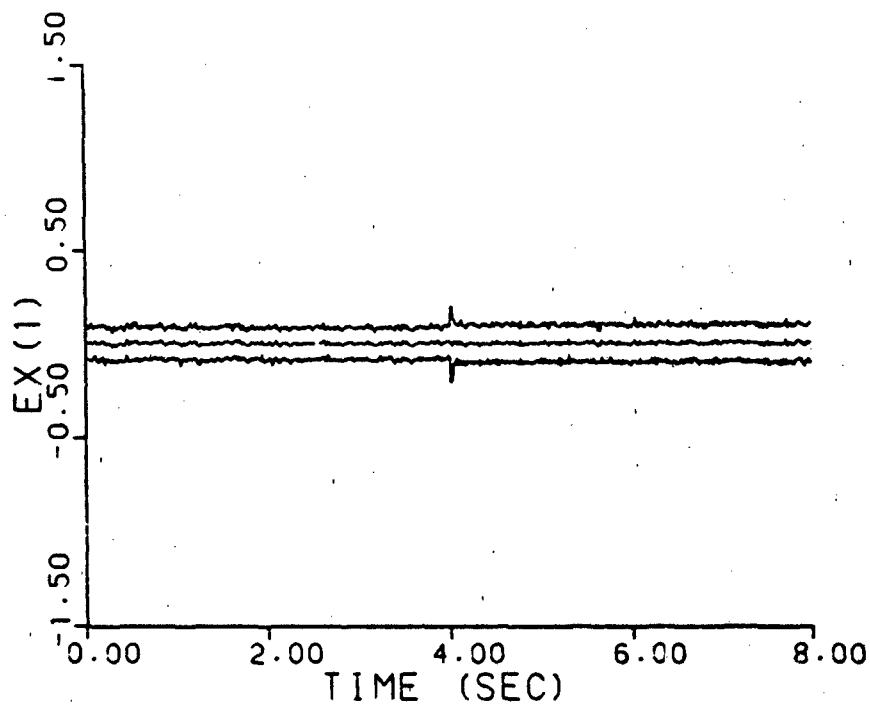


Figure A-14h. Parameter Covariance Monitoring
 a jumps: (0.07, 9.0) - (0.93, 41.0)
 no dither

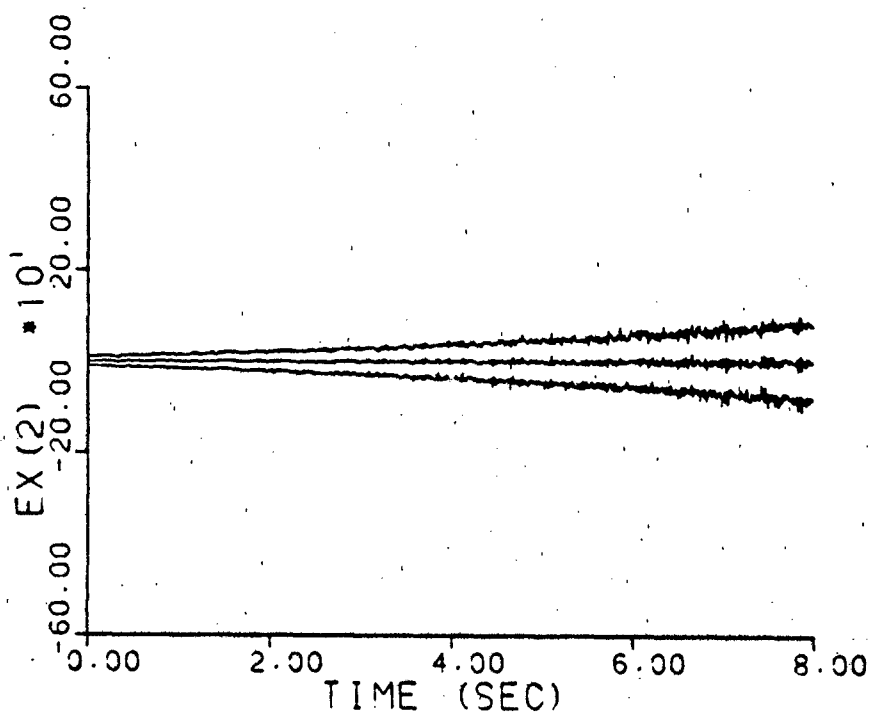
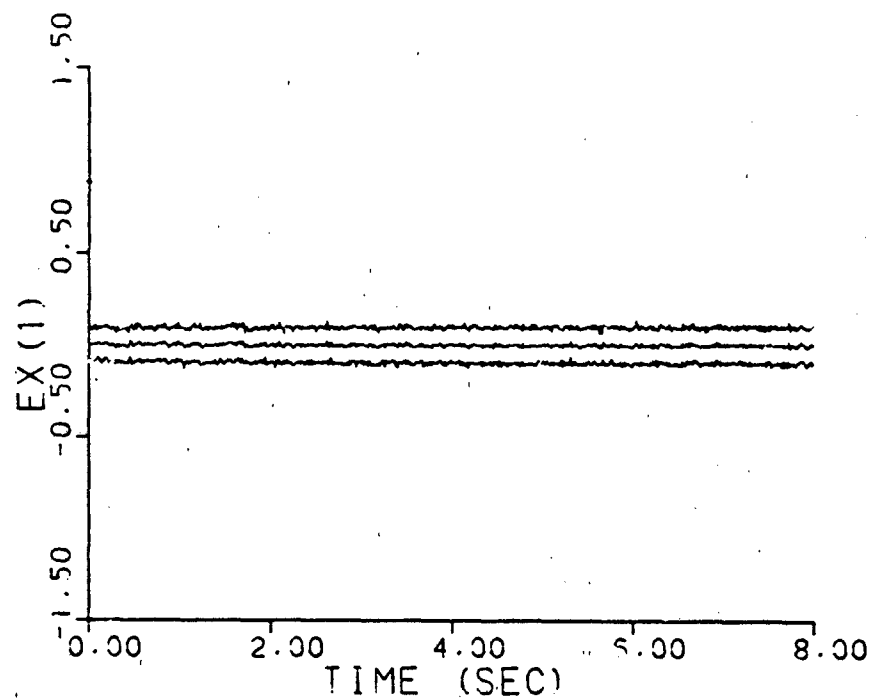


Figure A-14i. Parameter Covariance Monitoring
 a varies: (0.07, 9.0) - (0.93, 41.0)
 no dither

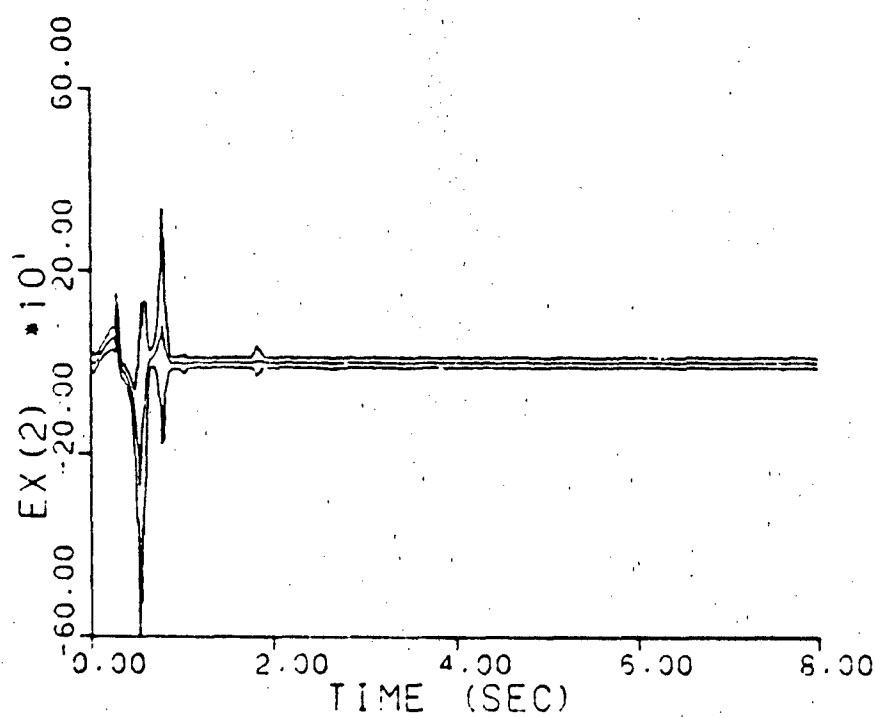
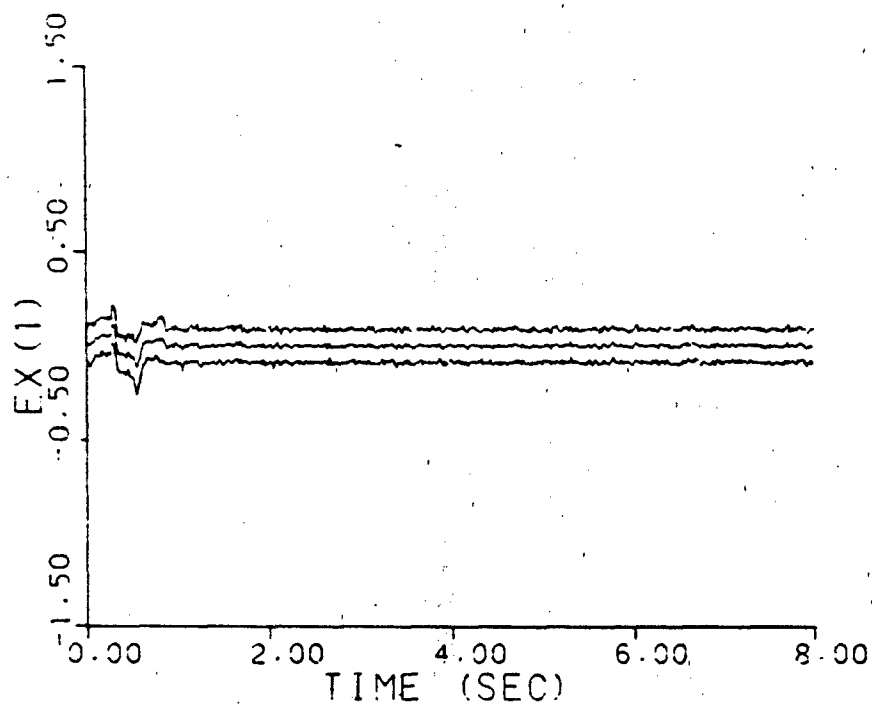


Figure A-15a. Parameter Covariance Monitoring
 $\underline{a} = (1, 3)$

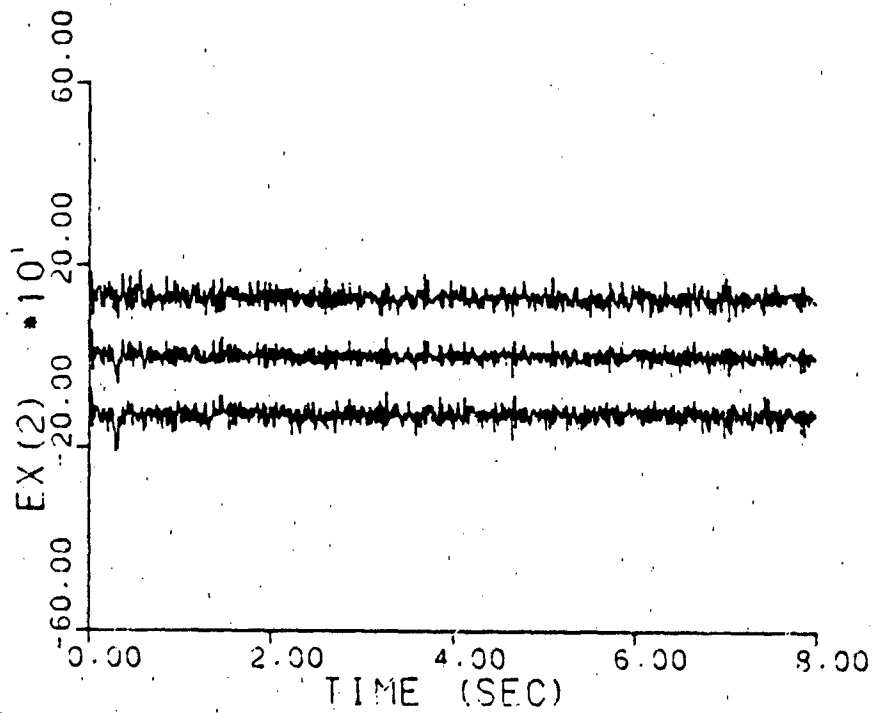
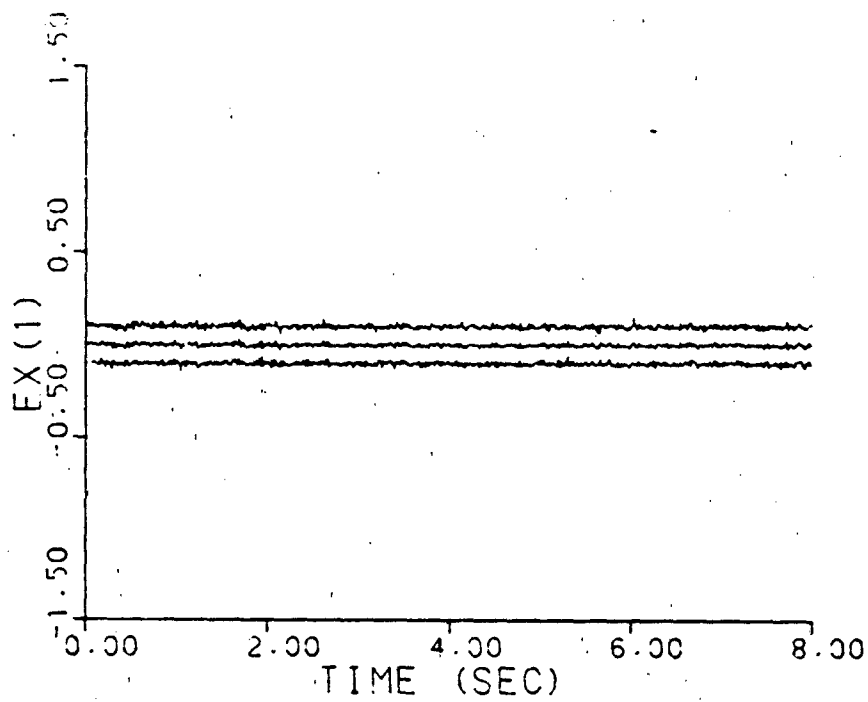


Figure A-15b. Parameter Covariance Monitoring
 $\underline{a} = (2, 9)$

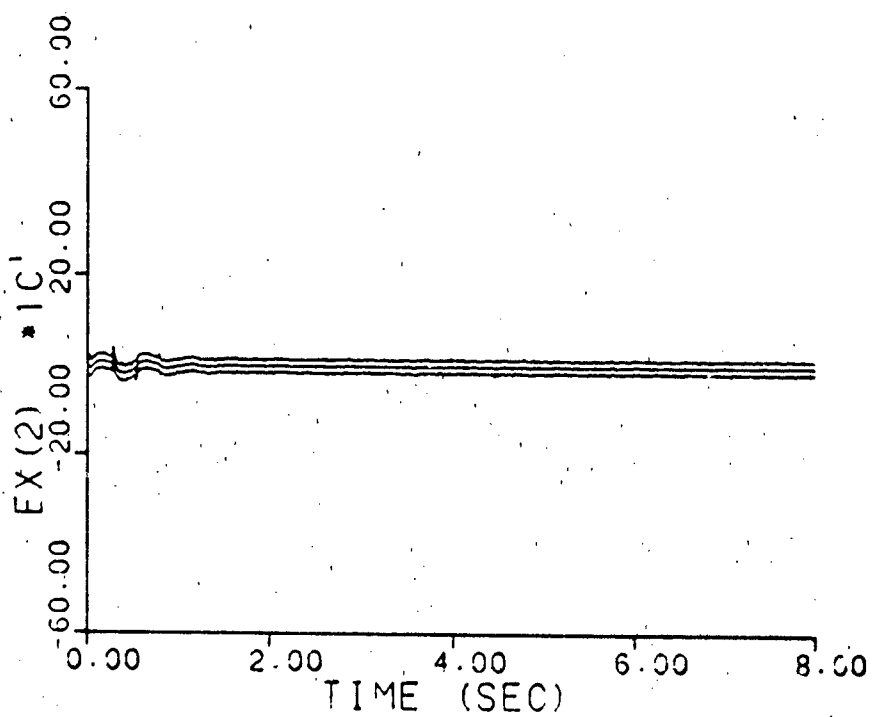
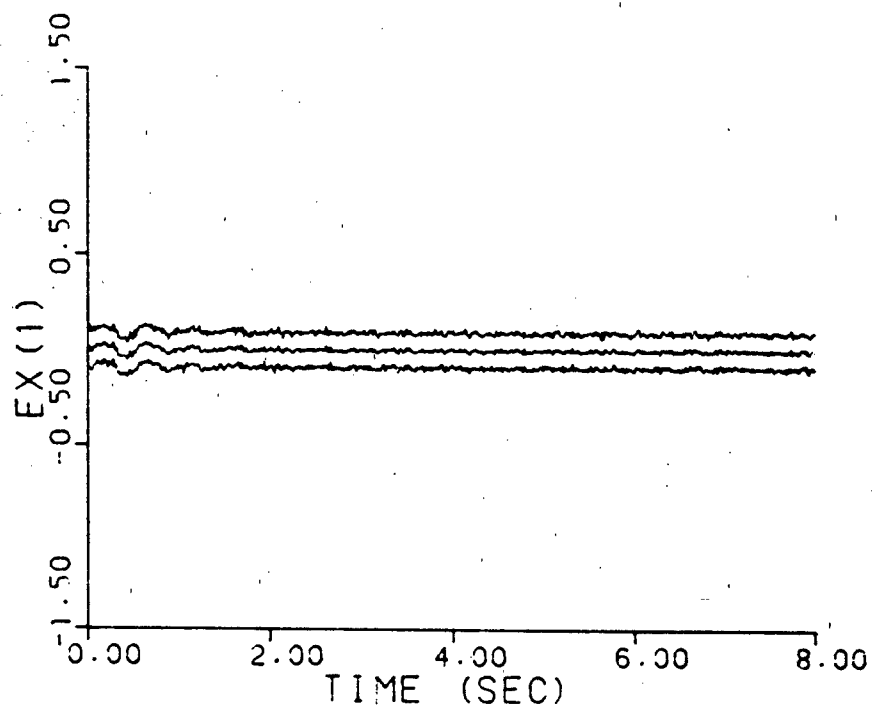


Figure A-15c. Parameter Covariance Monitoring
 $\underline{a} = (5, 4)$

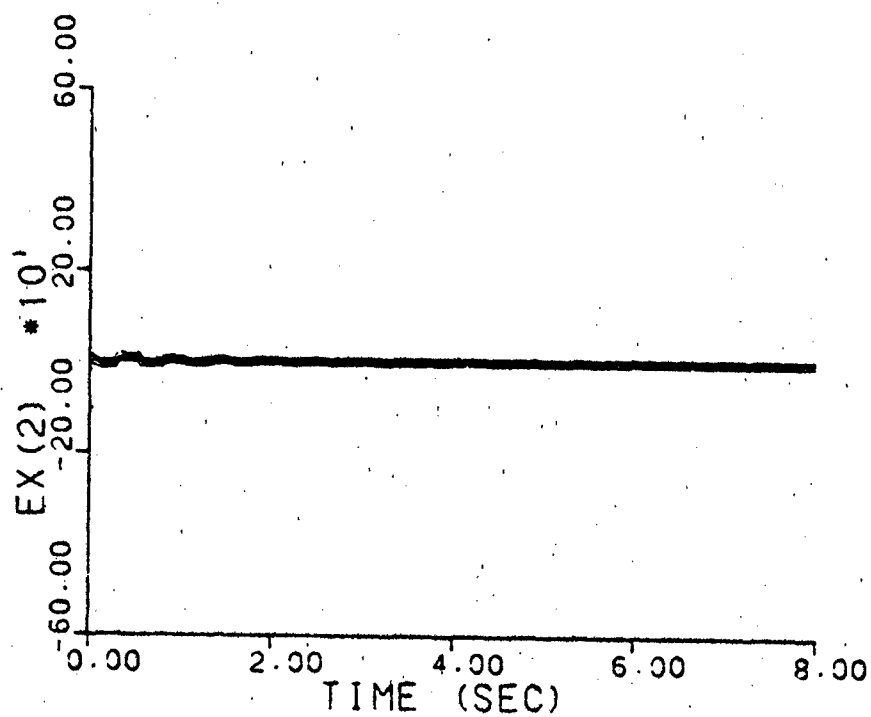
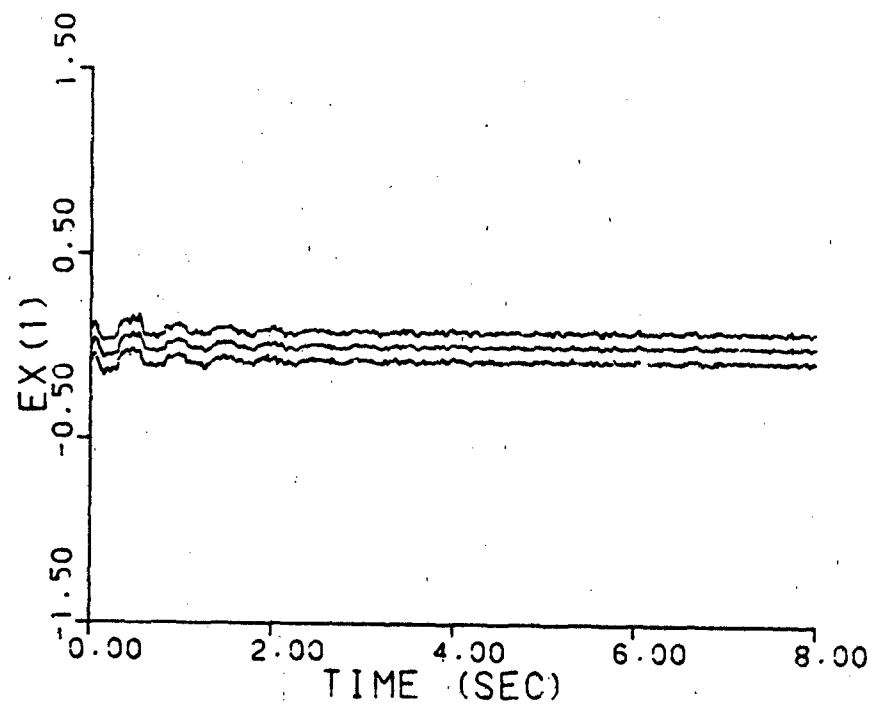


Figure A-15d. Parameter Covariance Monitoring
 $\underline{a} = (9, 2)$

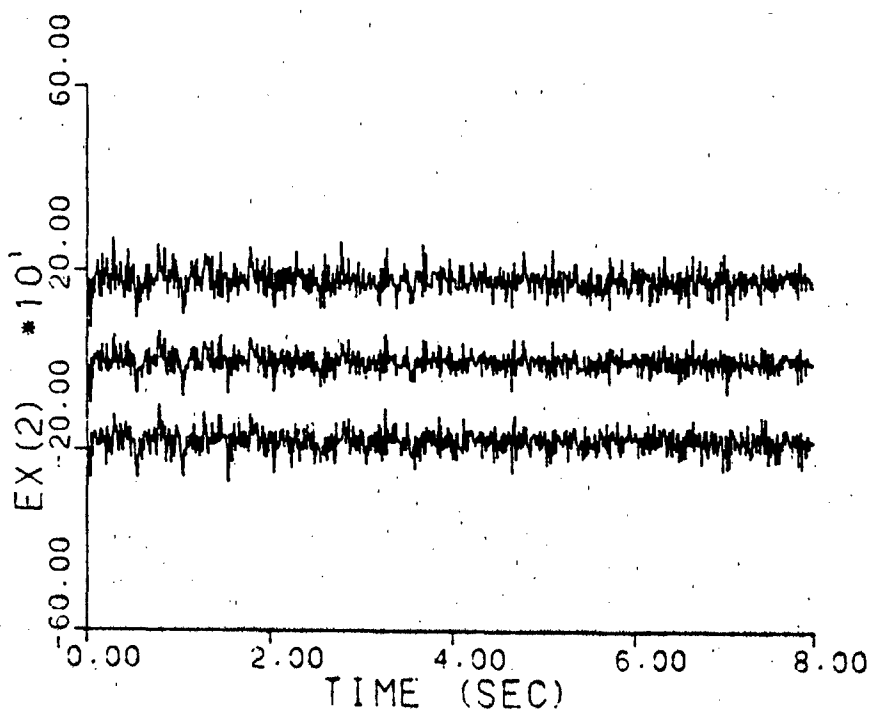
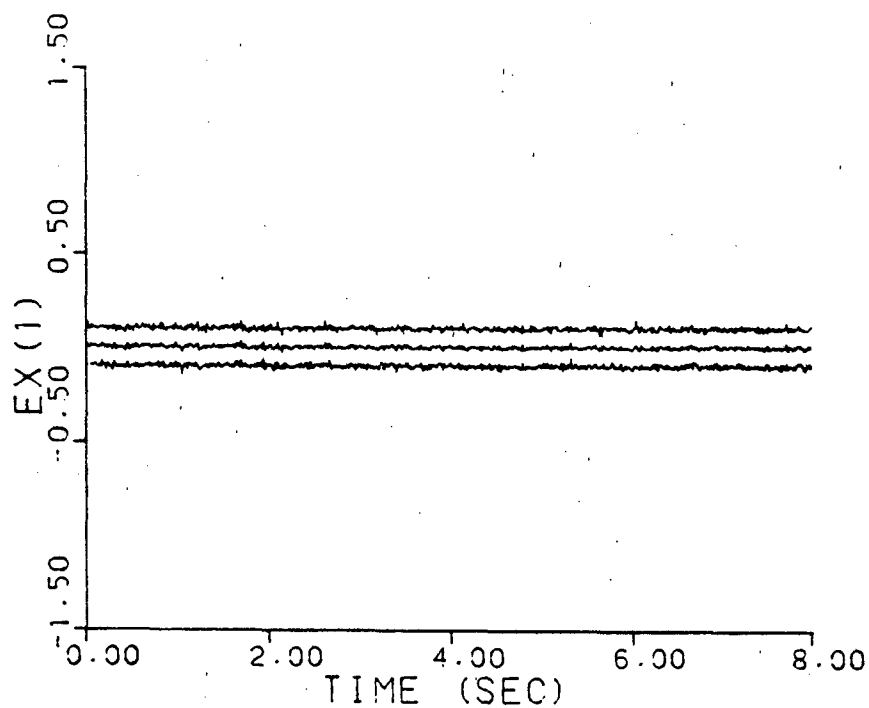


Figure A-15e. Parameter Covariance Monitoring
 $\underline{a} = (10, 10)$

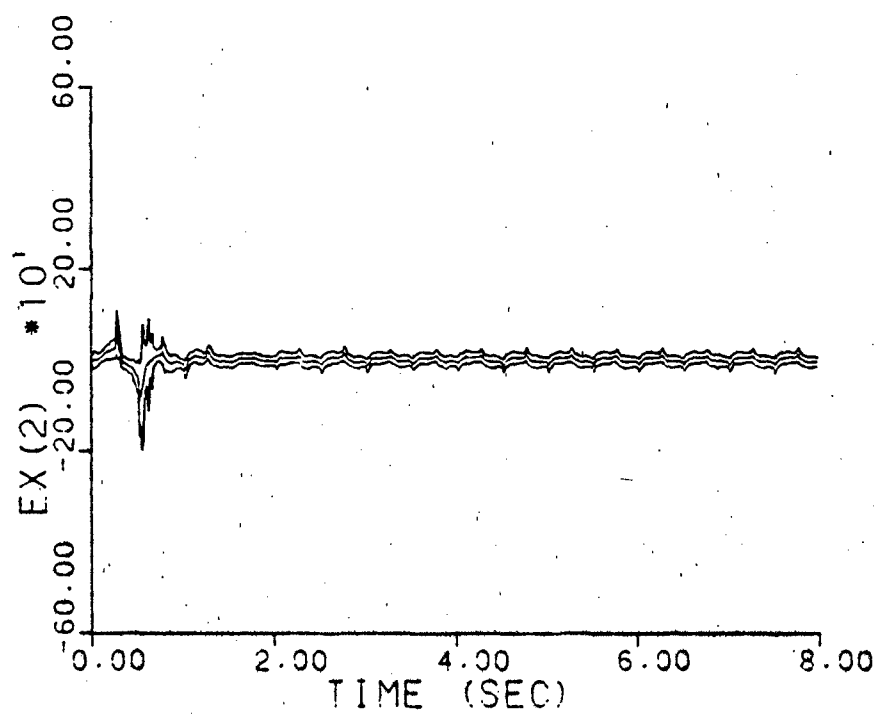
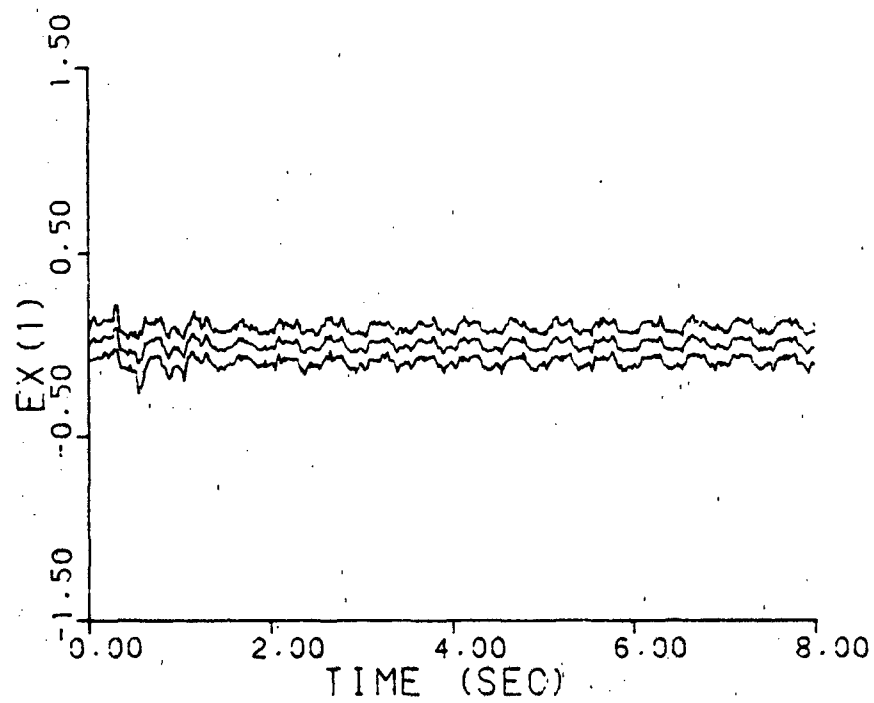


Figure A-15f. Parameter Covariance Monitoring
 $\underline{a} = (0.07, 9.0)$

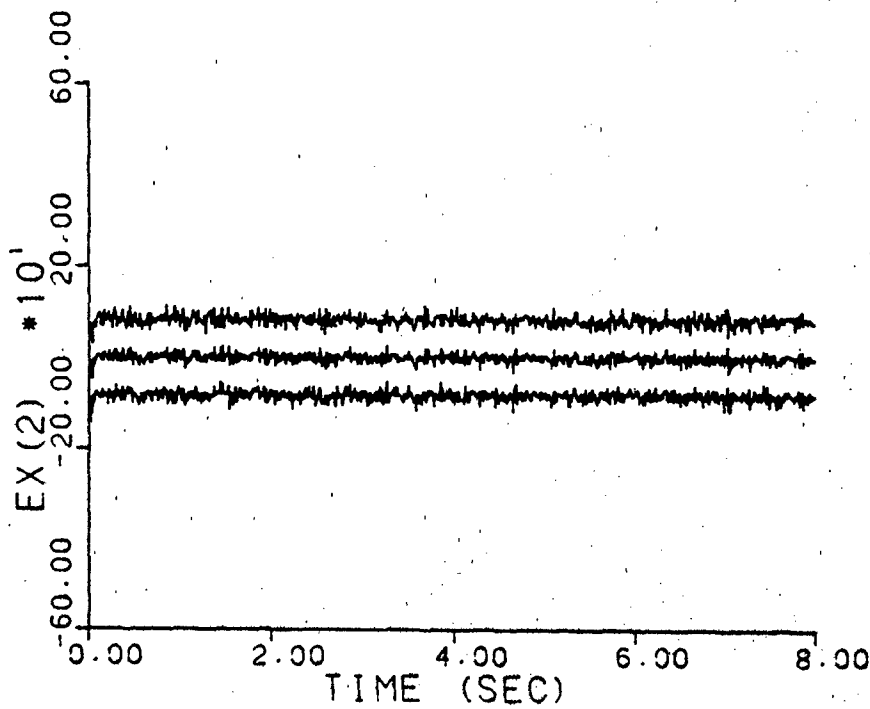
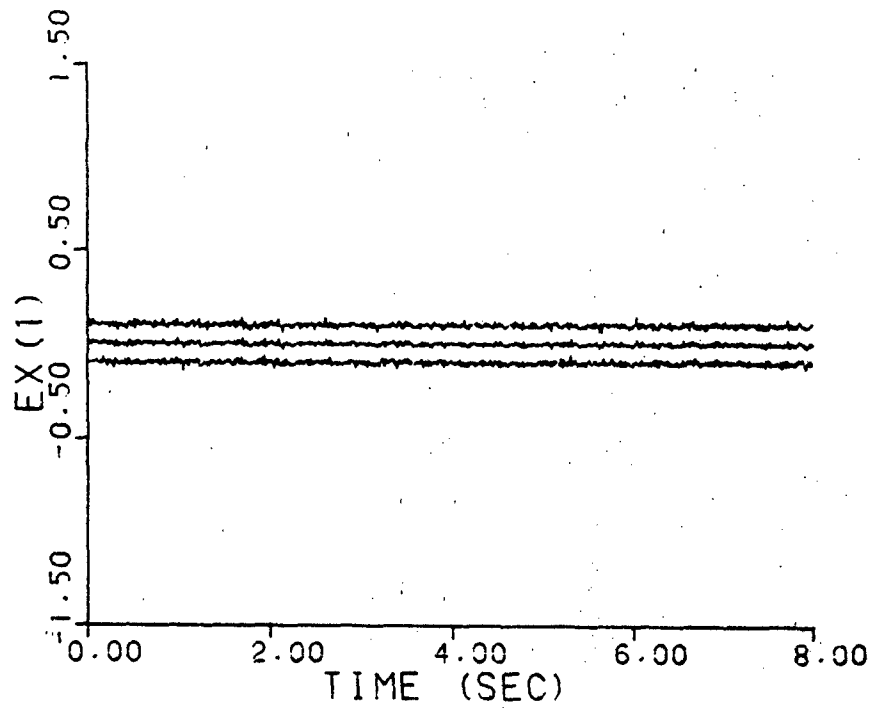


Figure A-15g. Parameter Covariance Monitoring
 $\underline{a} = (0.93, 41.0)$

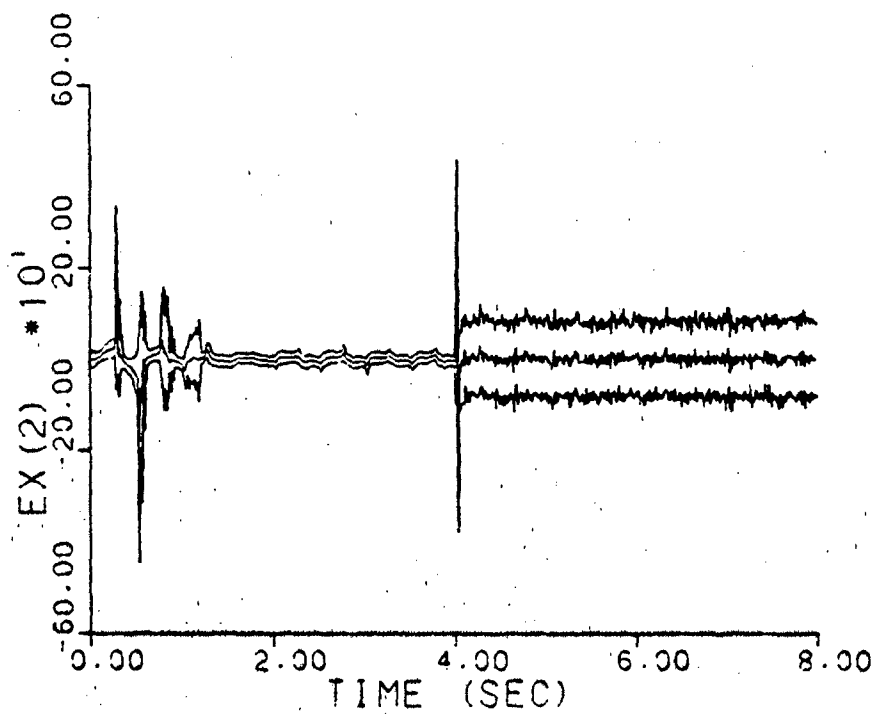
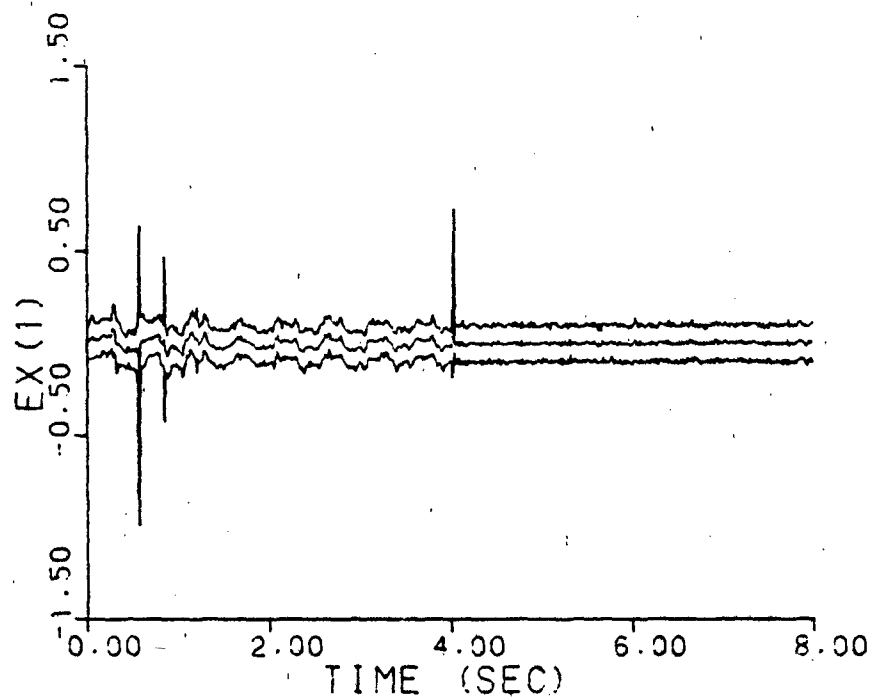


Figure A-15h. Parameter Covariance Monitoring
a jumps: (0.07, 9.0) - (0.93, 41.0).

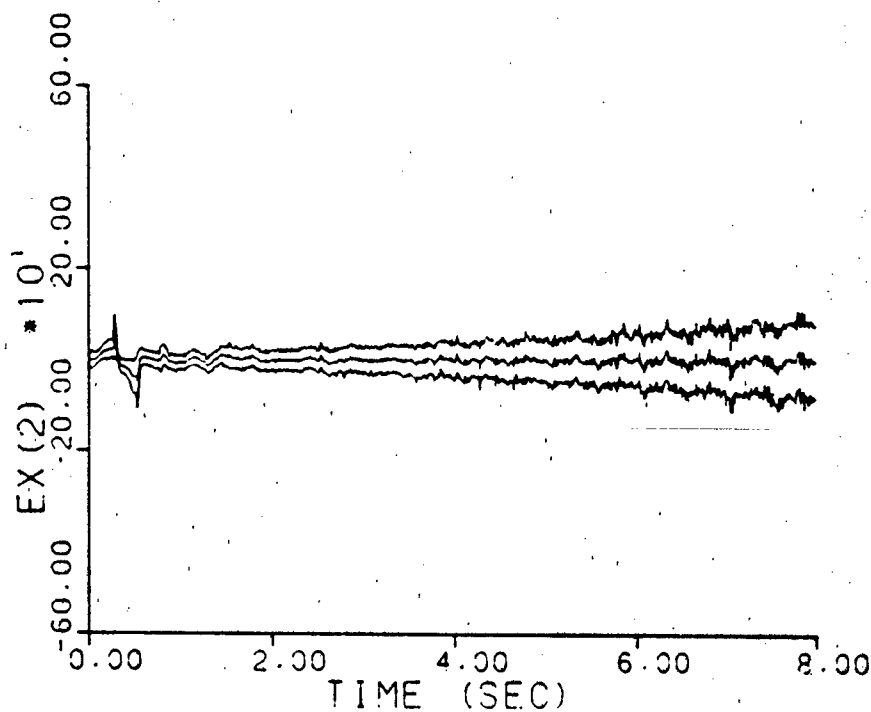
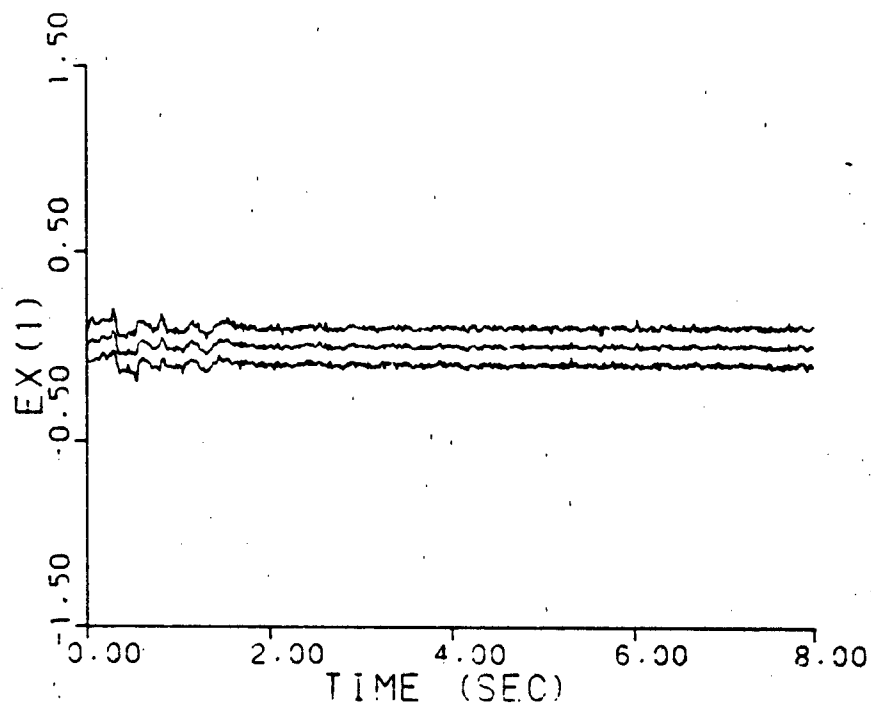


Figure A-15i. Parameter Covariance Monitoring
a varies: (0.07, 9.0) - (0.93, 41.0)

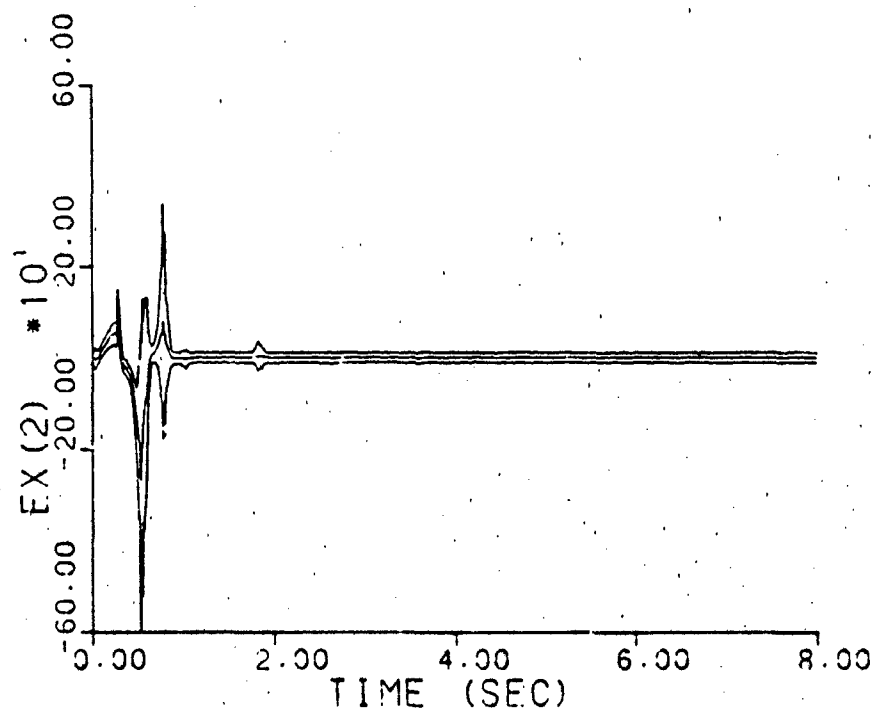
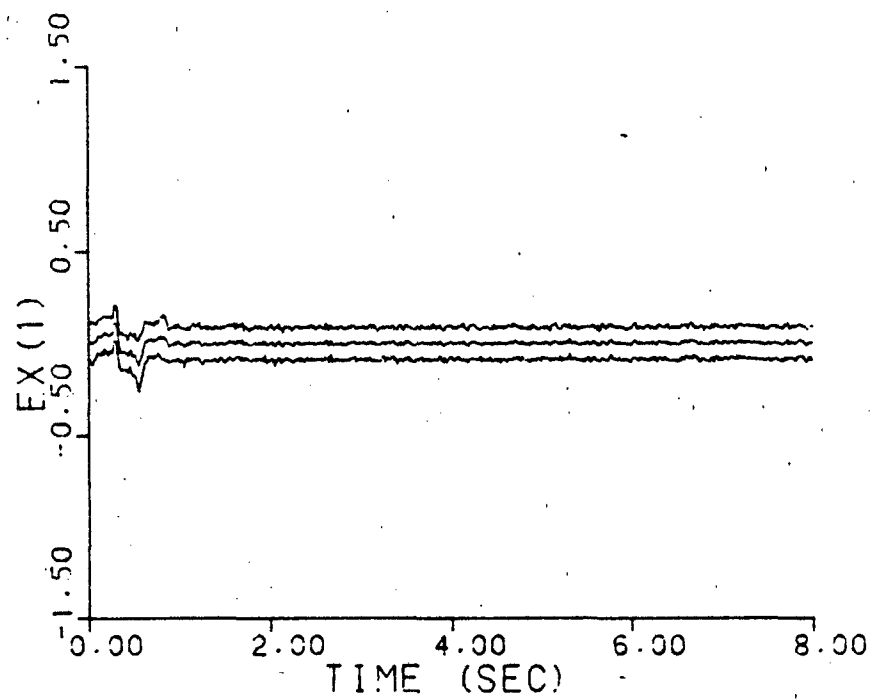


Figure A-16a. Parameter Covariance Monitoring
 $\underline{a} = (1, 3)$, warm-up

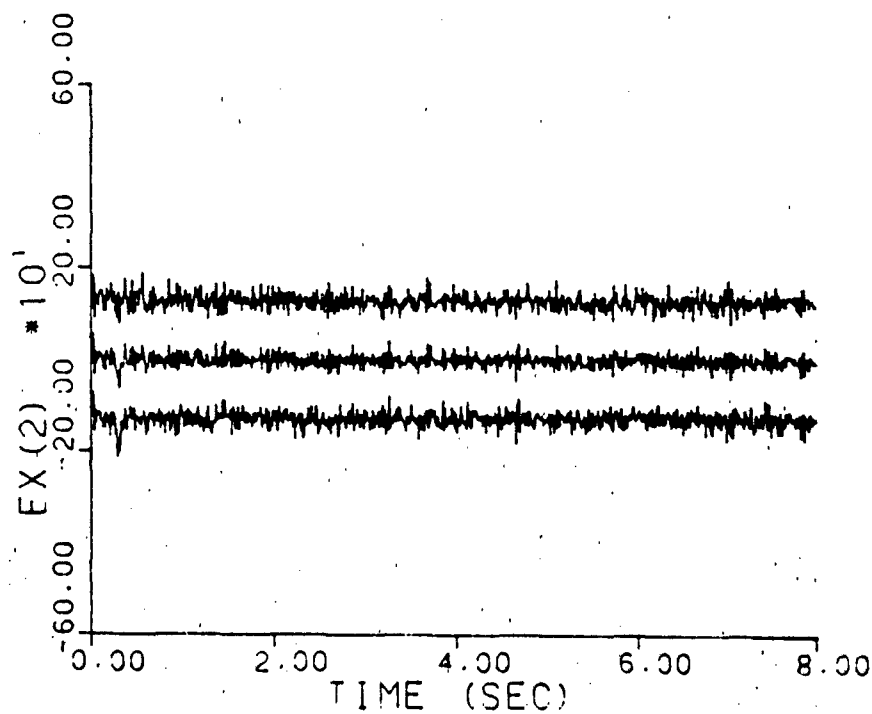
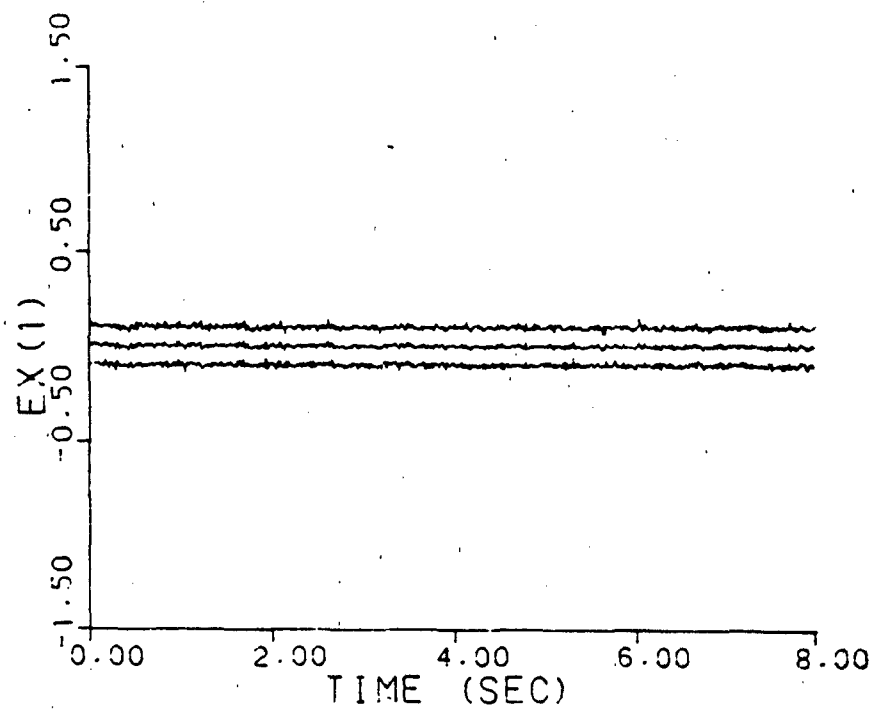


Figure A-16b. Parameter Covariance Monitoring
 $\underline{a} = (2, 9)$, warm-up

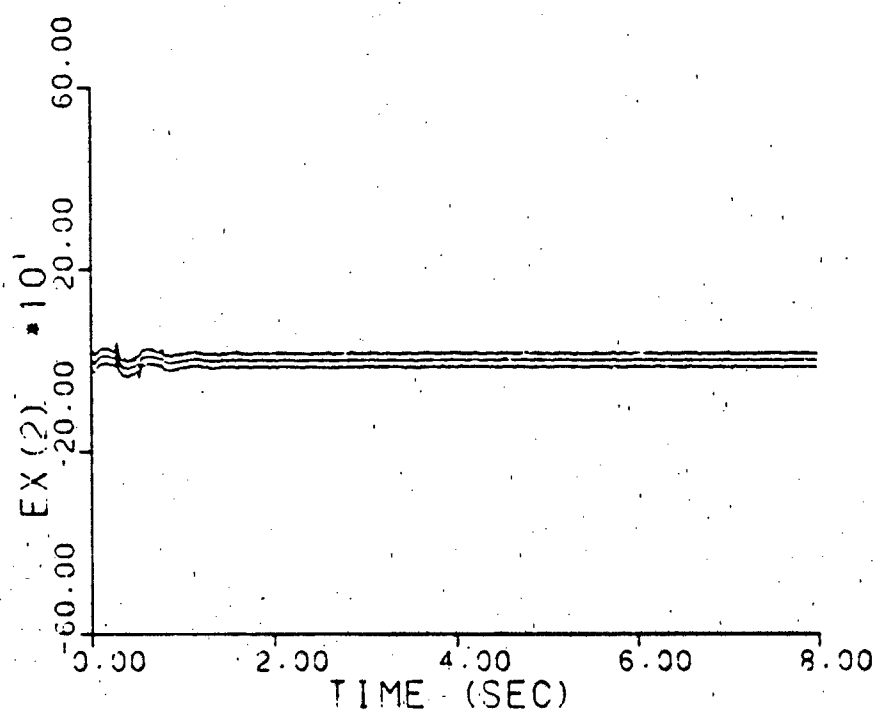
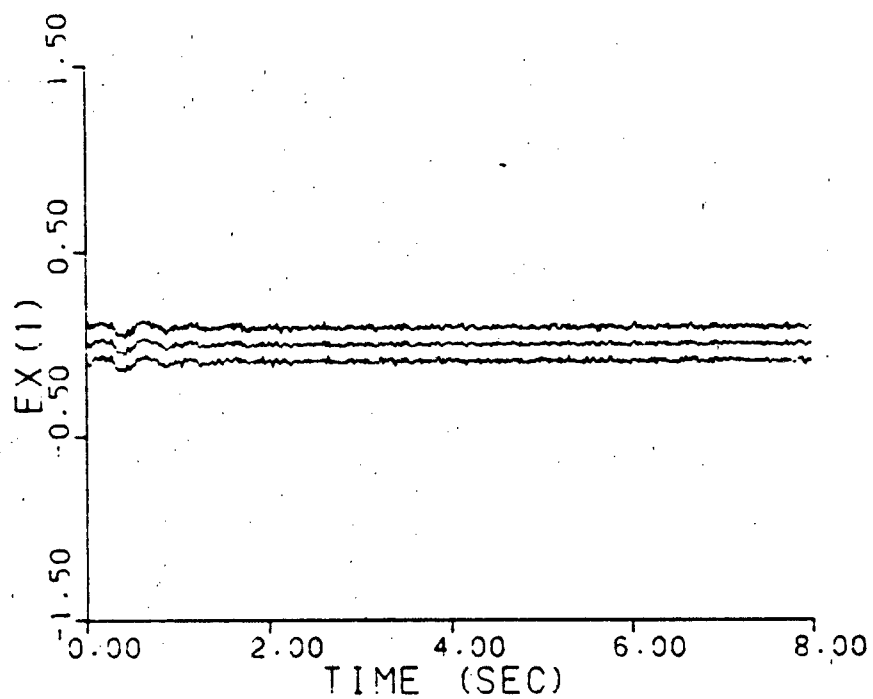


Figure A-16c. Parameter Covariance Monitoring
 $\underline{a} = (5, 4)$, warm-up

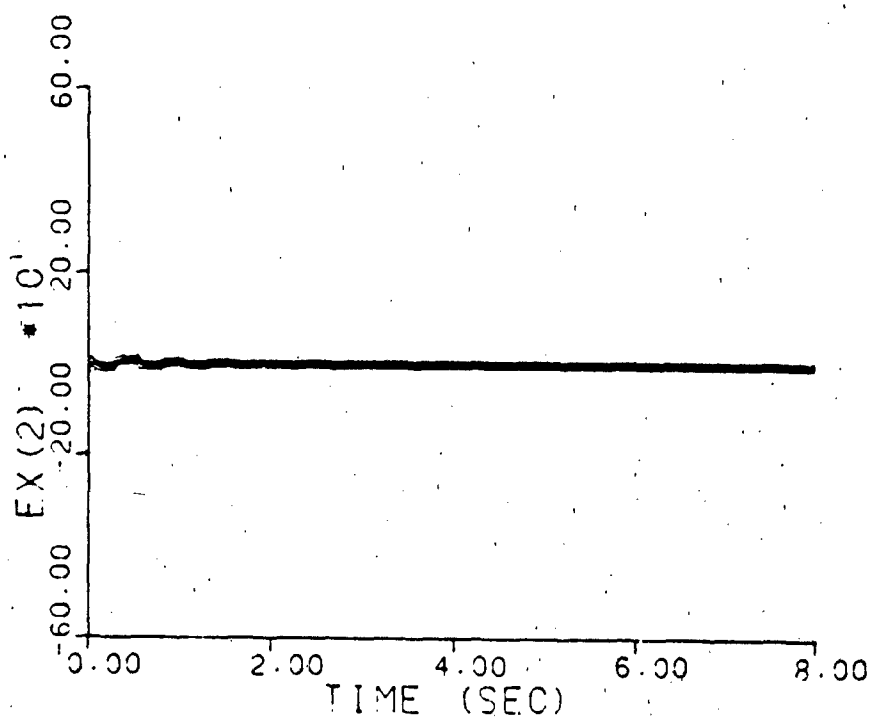
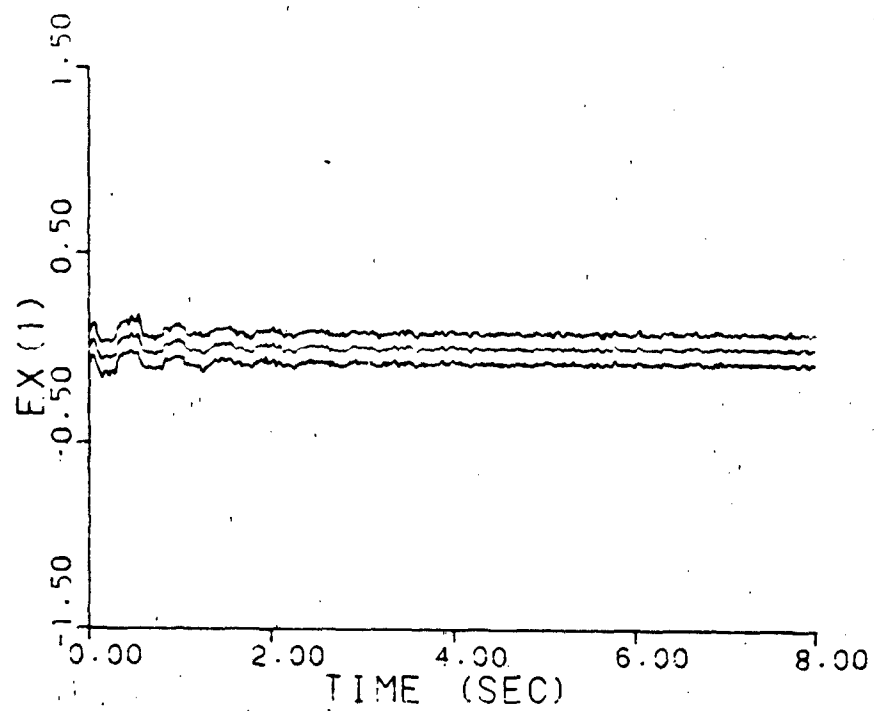


Figure A-16d. Parameter Covariance Monitoring
 $\underline{a} = (9, 2)$, warm-up

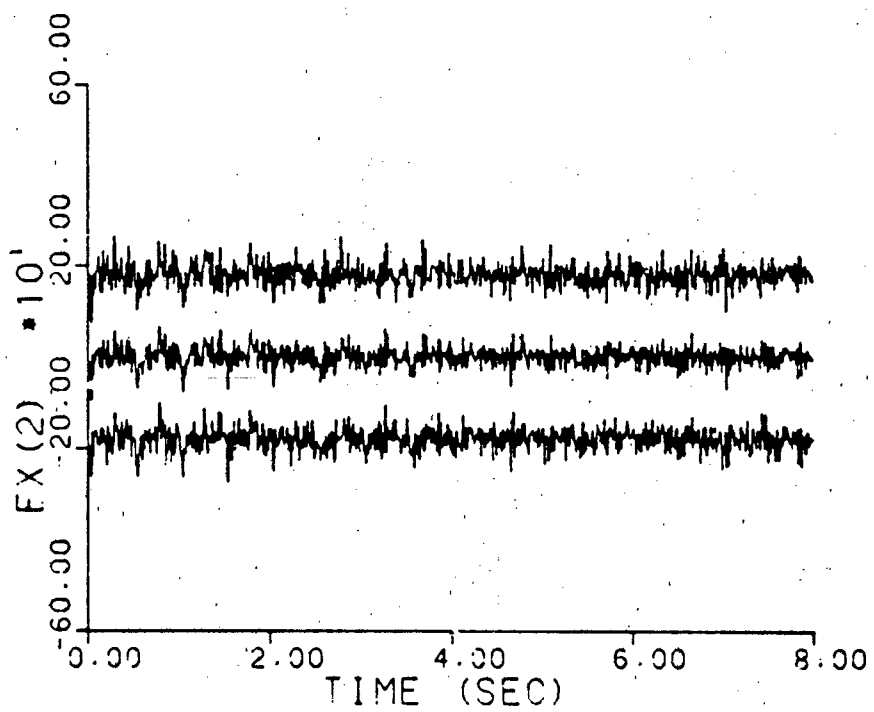
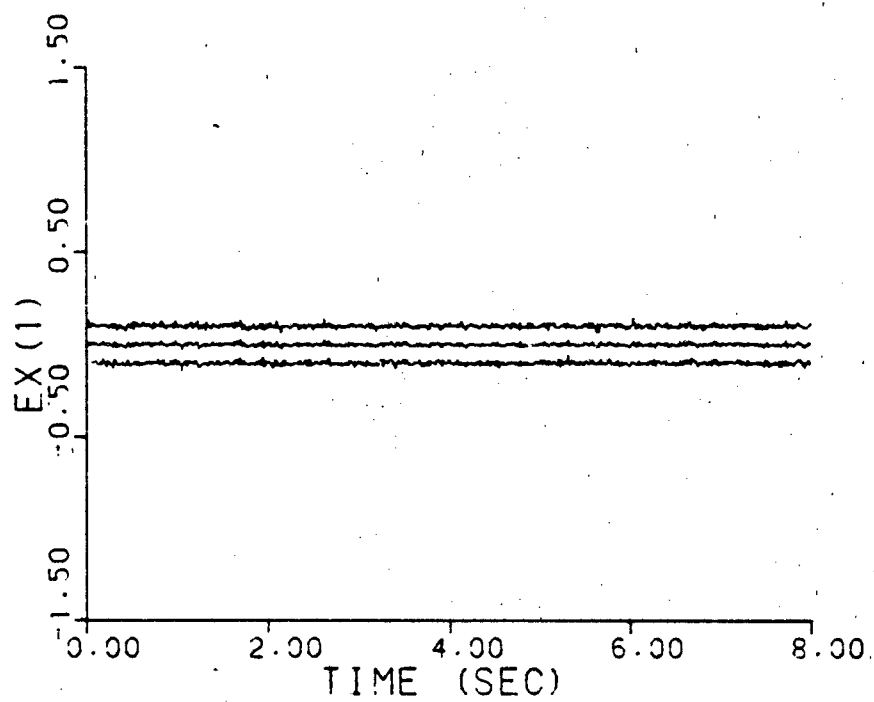


Figure A-16e. Parameter Covariance Monitoring
 $\underline{a} = (10, 10)$, warm-up

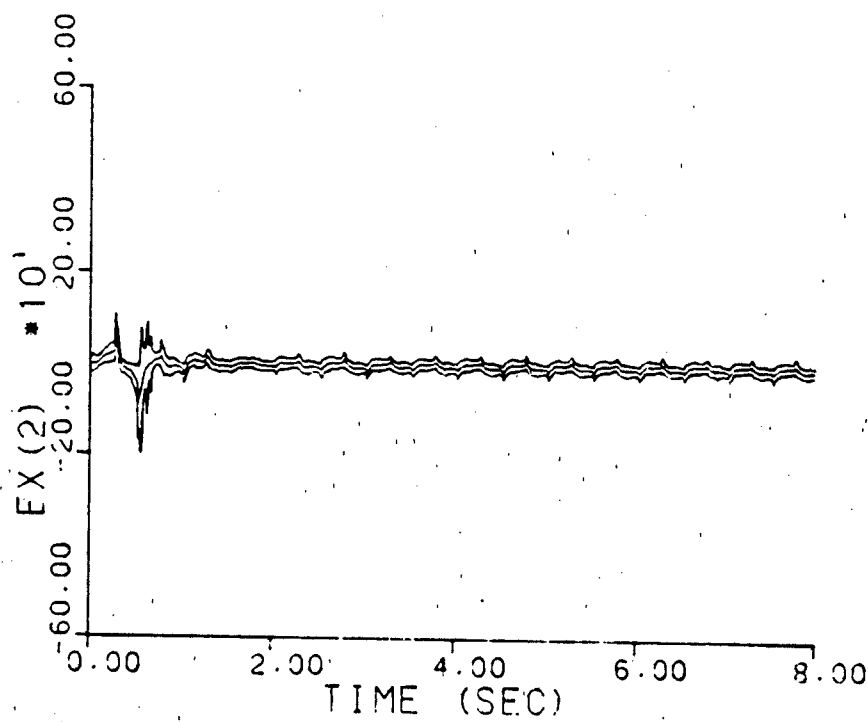
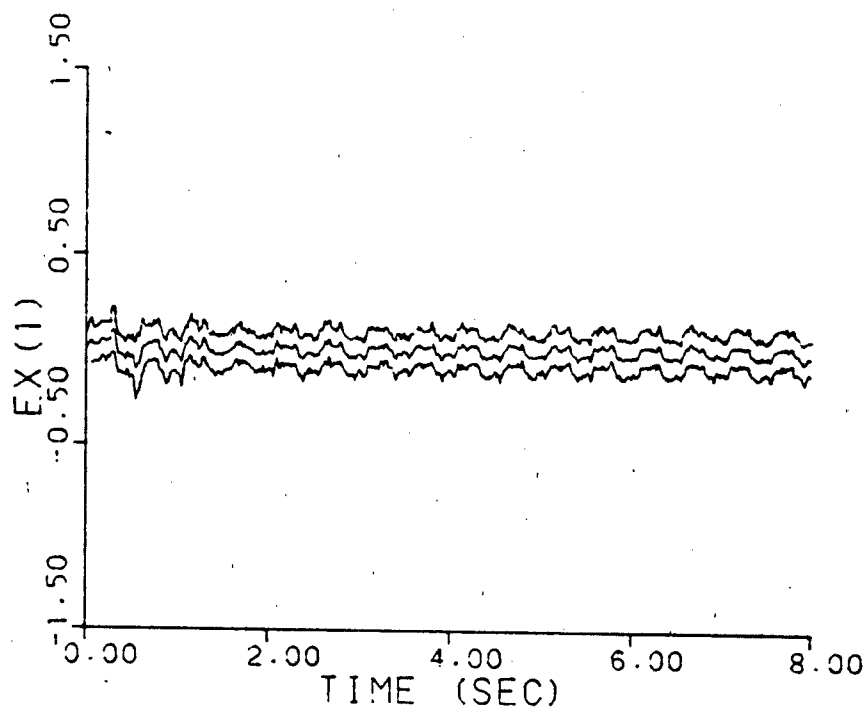


Figure A-16f. Parameter Covariance Monitoring
 $\underline{a} = (0.07, 9.0)$, warm-up

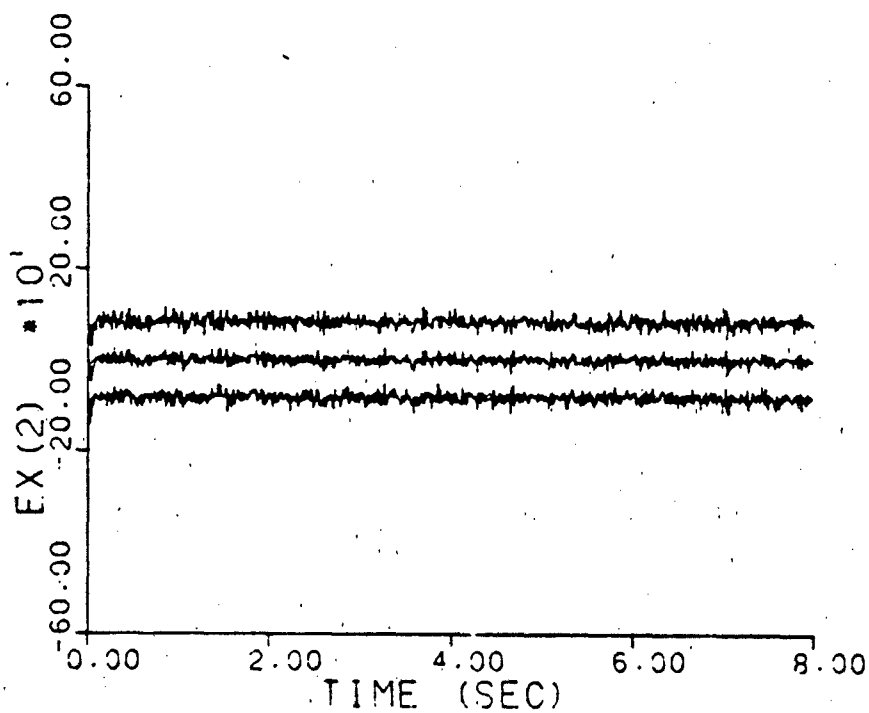
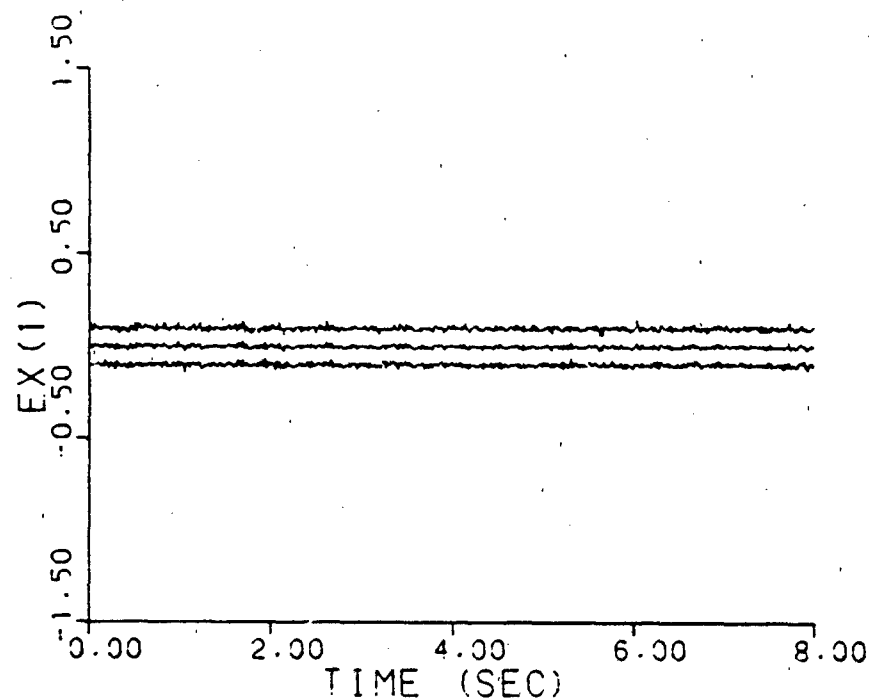


Figure A-16g. Parameter Covariance Monitoring
 $\underline{a} = (0.93, 41.0)$, warm-up

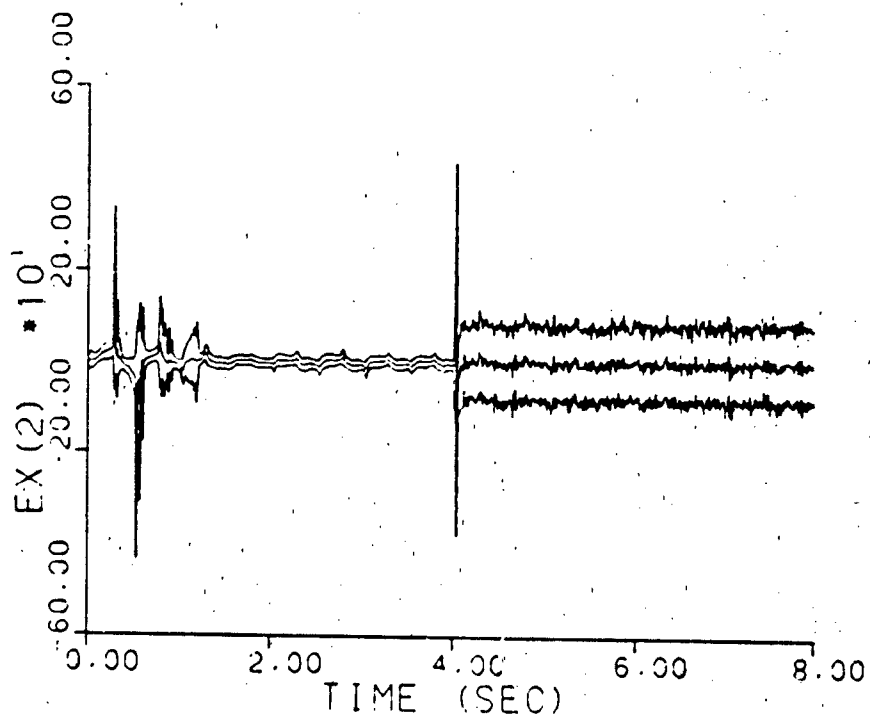
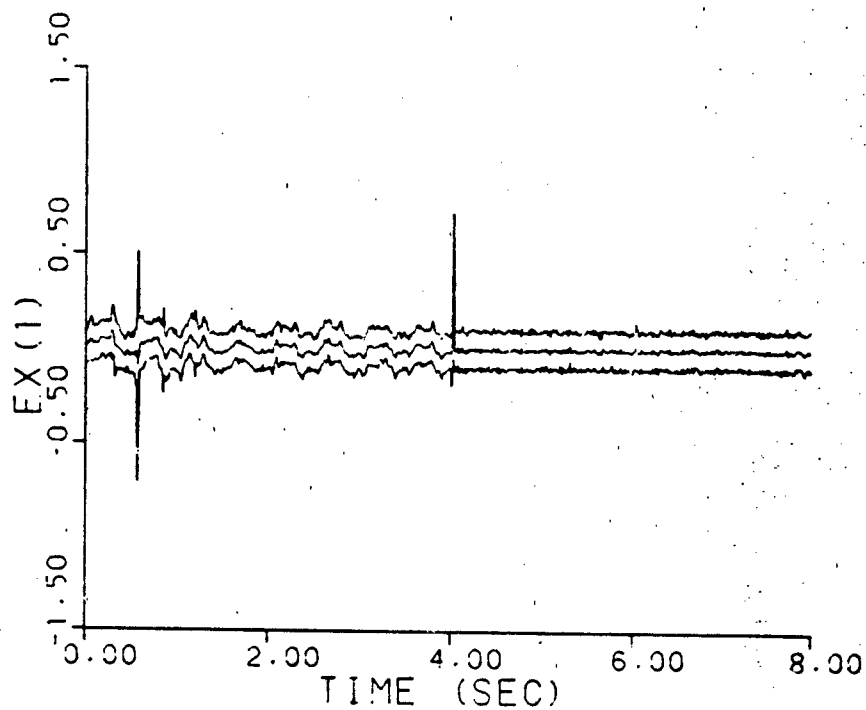


Figure A-16h. Parameter Covariance Monitoring
a jumps: (0.07, 9.0) - (0.93, 41.0)
warm-up

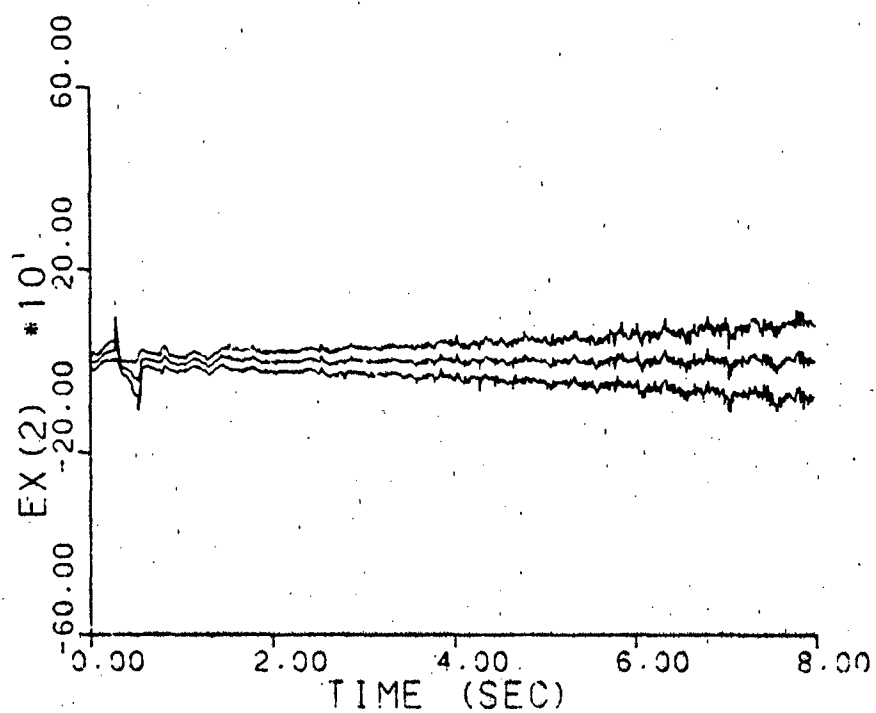
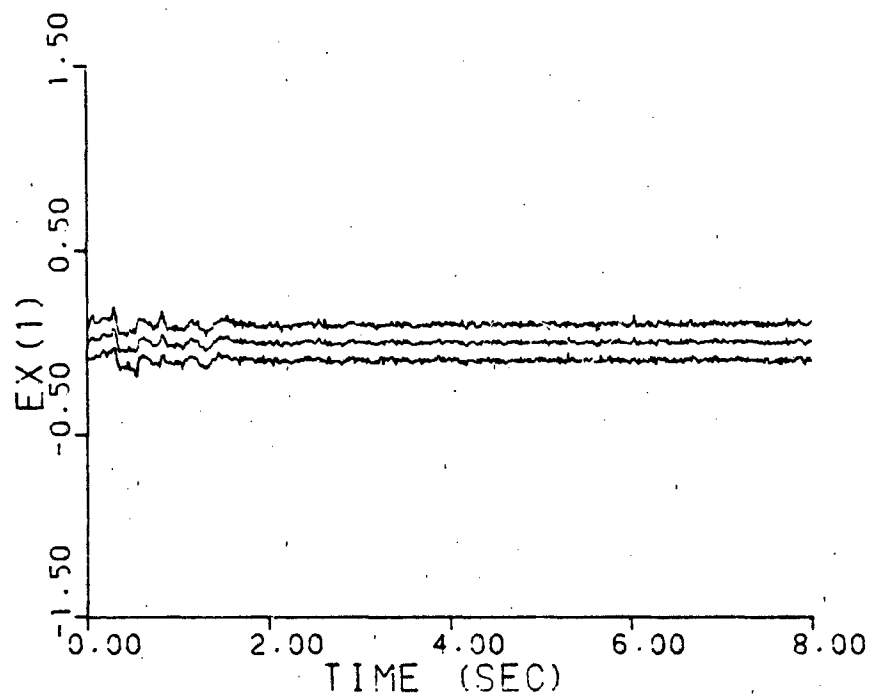


Figure A-16i. Parameter Covariance Monitoring
a varies: (0.07, 9.0) - (0.93, 41.0)
warm-up

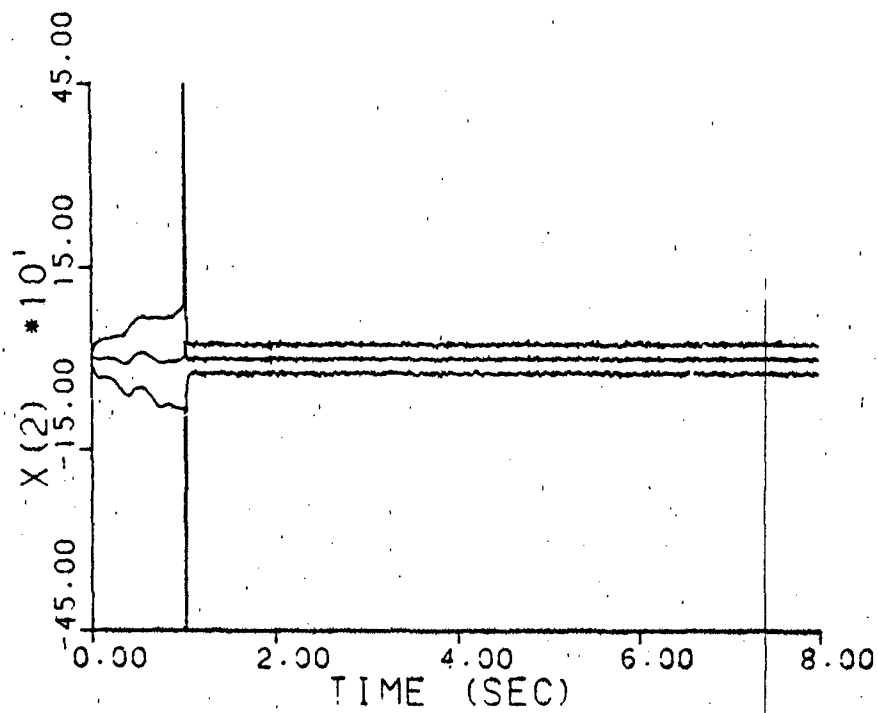
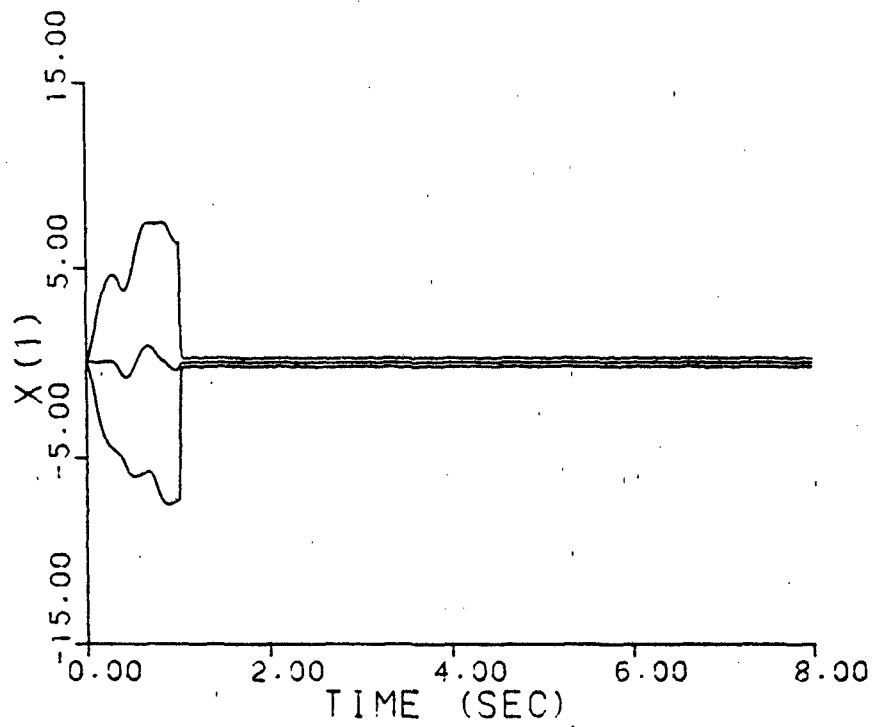


Figure A-17a. Benchmark Controller, $\underline{a} = (1, 3)$

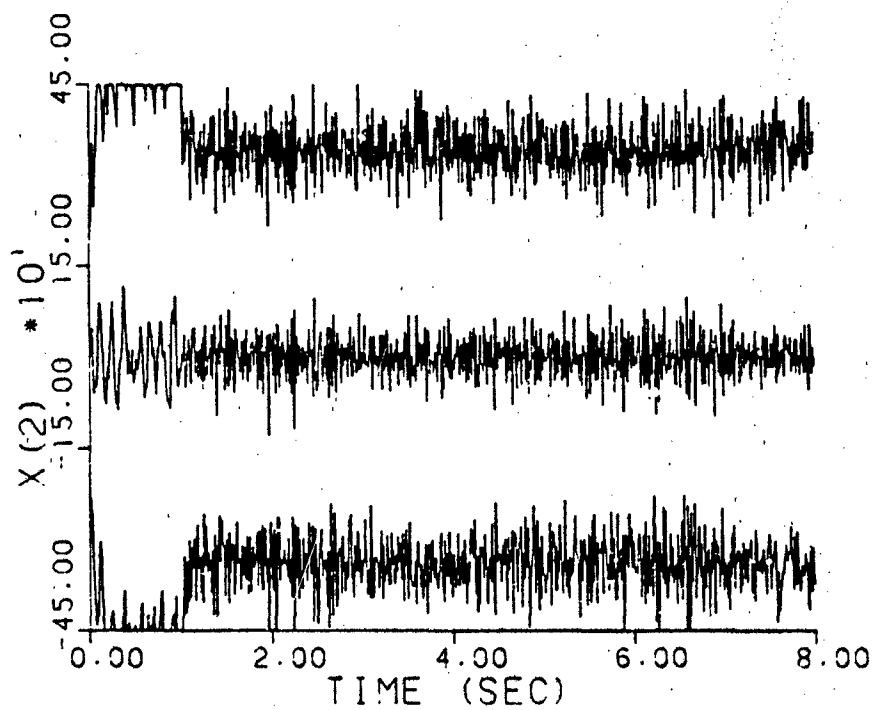
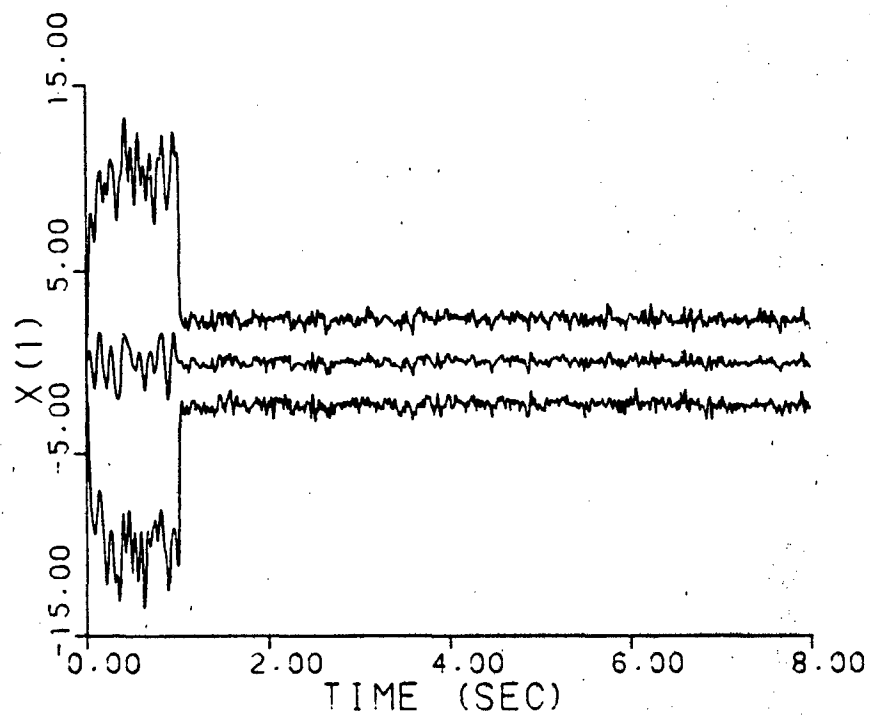


Figure A-17b. Benchmark Controller, $\underline{a} = (2, 9)$

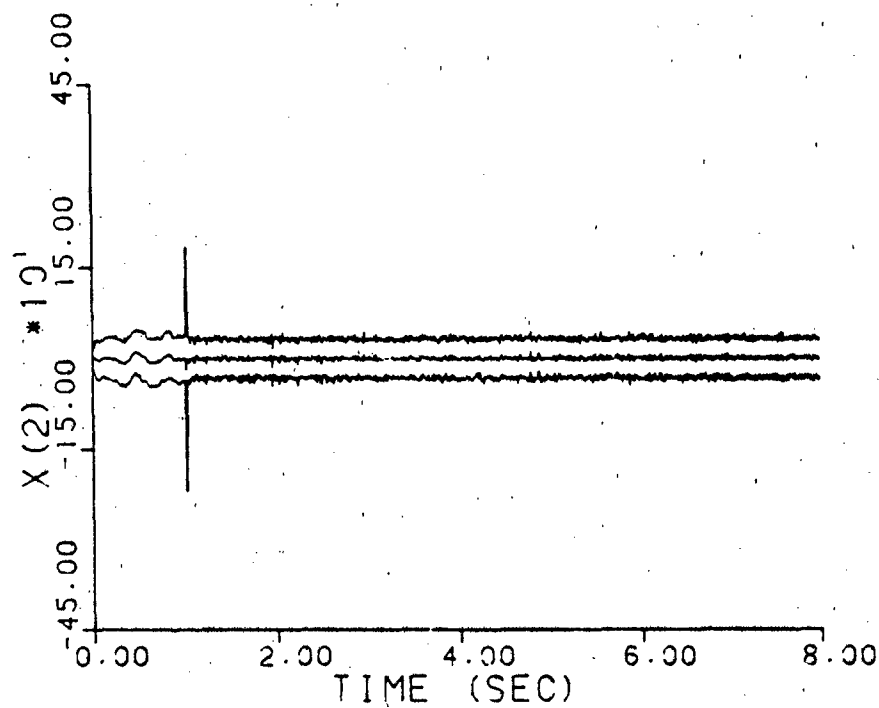
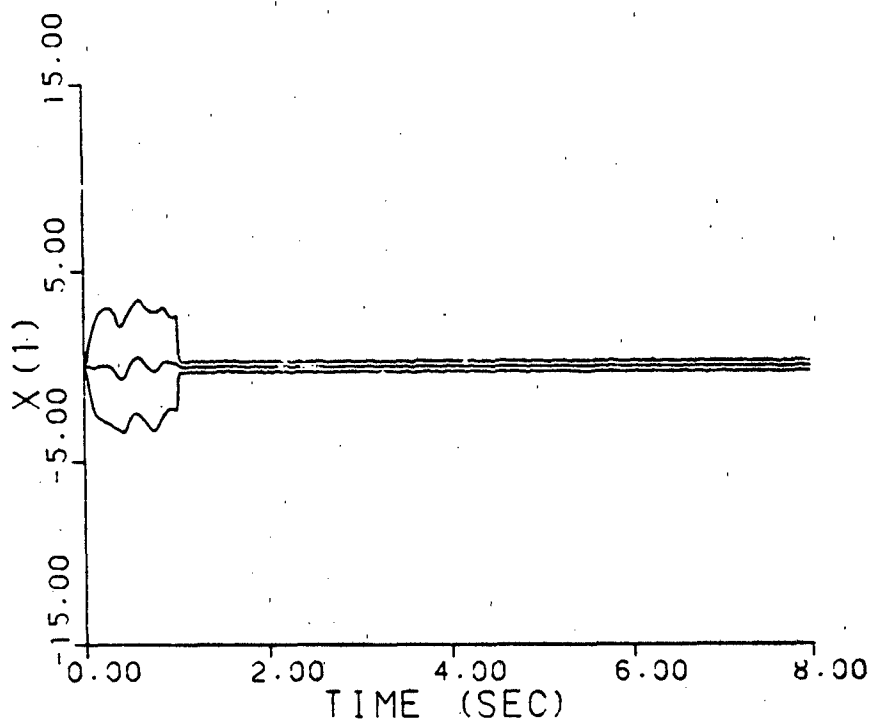


Figure A-17c. Benchmark Controller, $\underline{a} = (5, 4)$

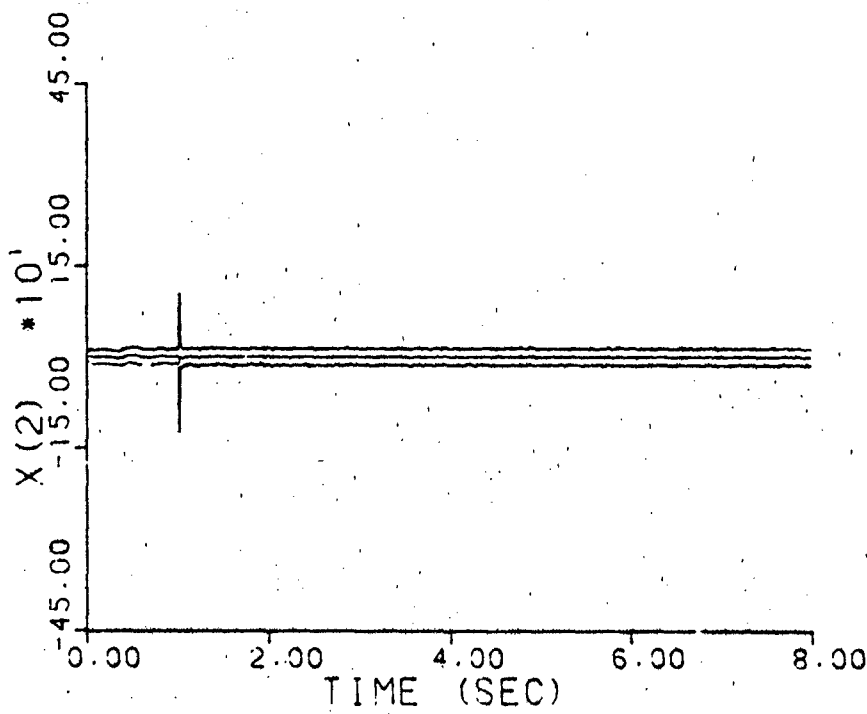
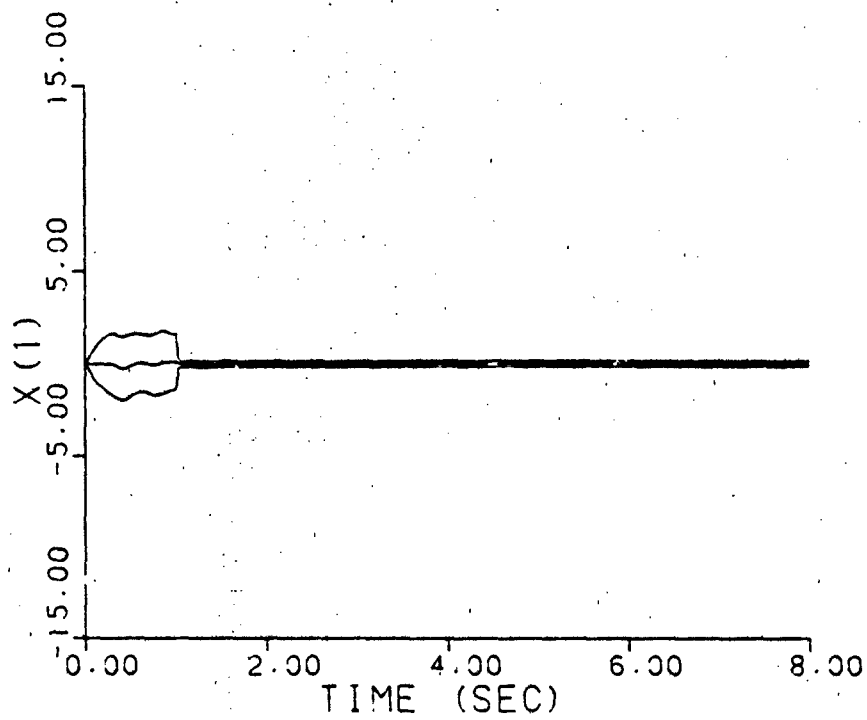


Figure A-17d. Benchmark Controller, $\underline{a} = (9, 2)$

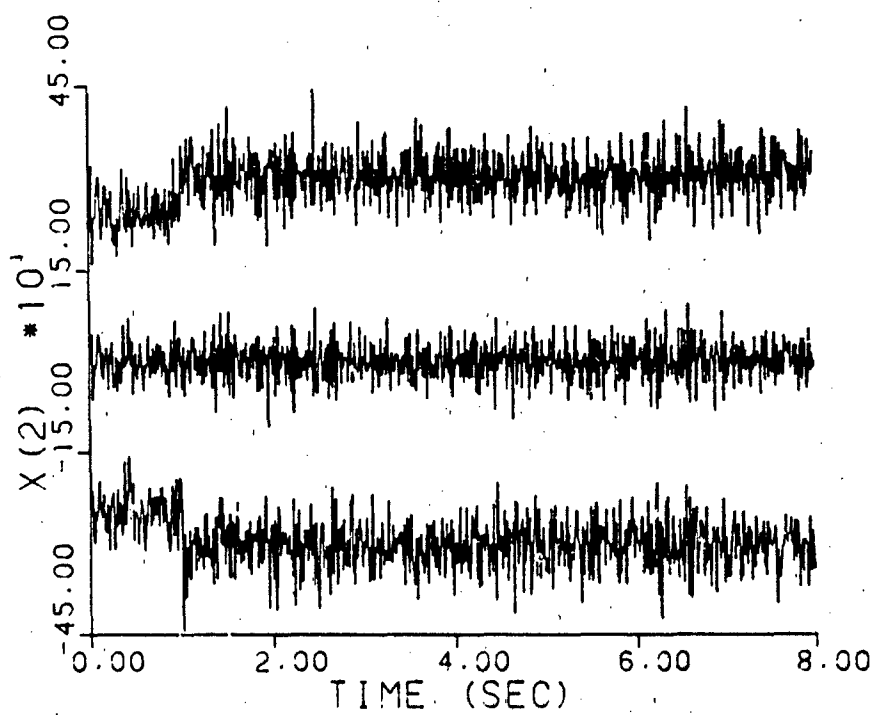
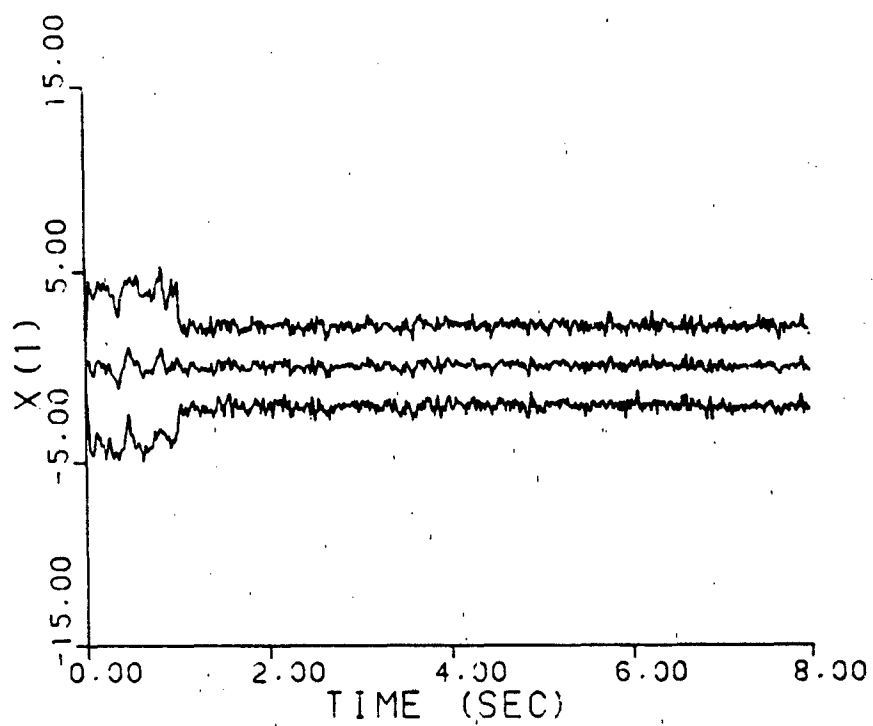


Figure A-17e. Benchmark Controller, $\underline{a} = (10, 10)$

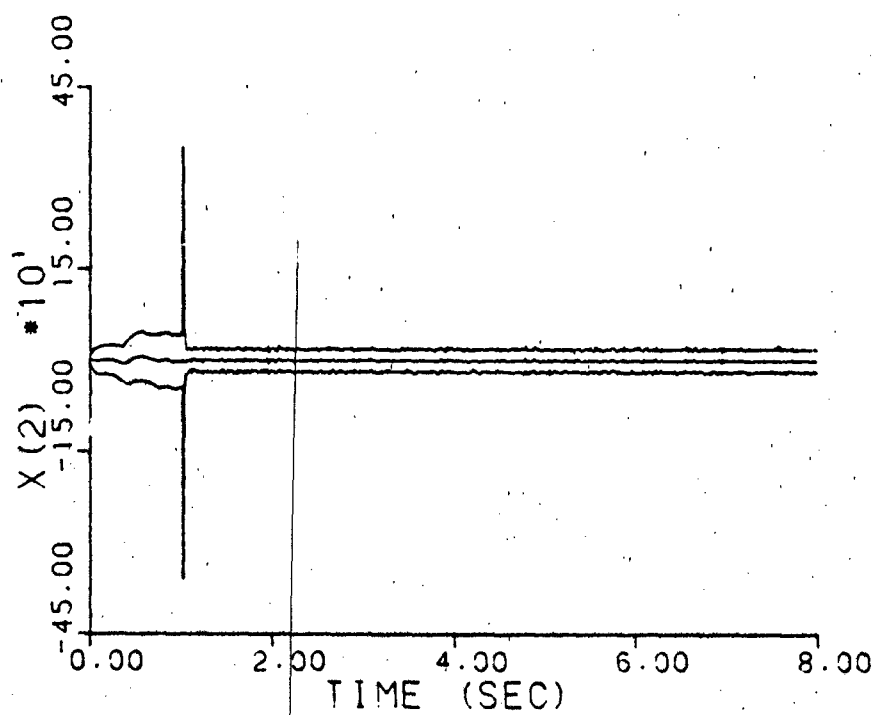
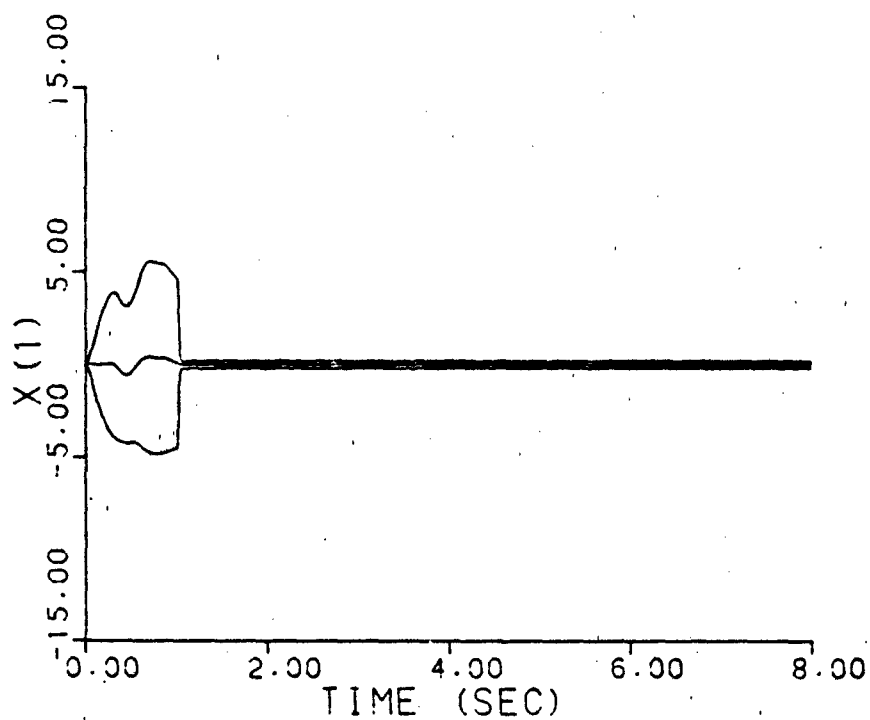


Figure A-17f. Benchmark Controller, $\underline{a} = (0.07, 9.0)$

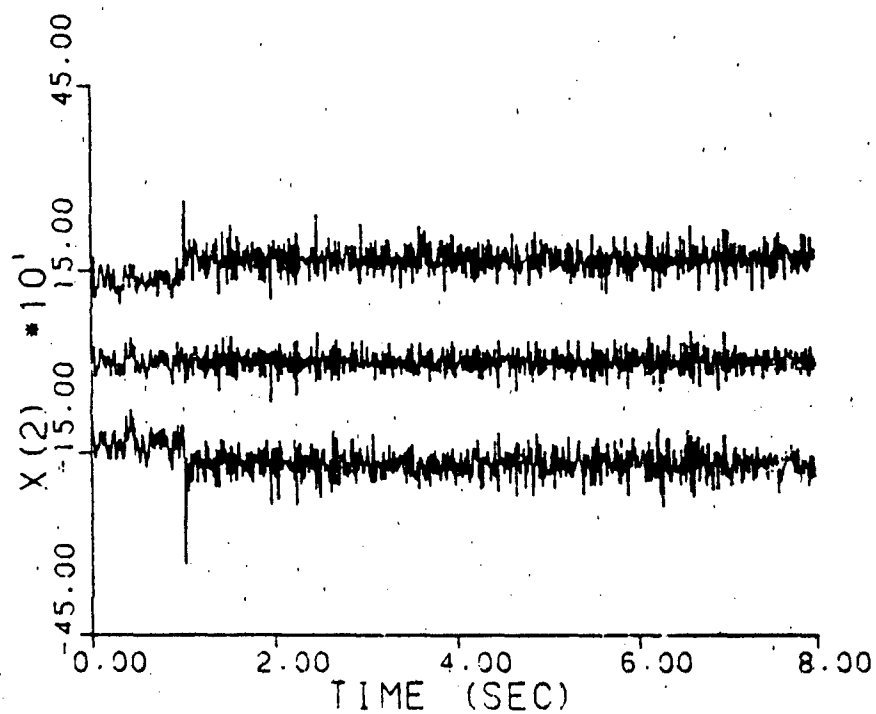
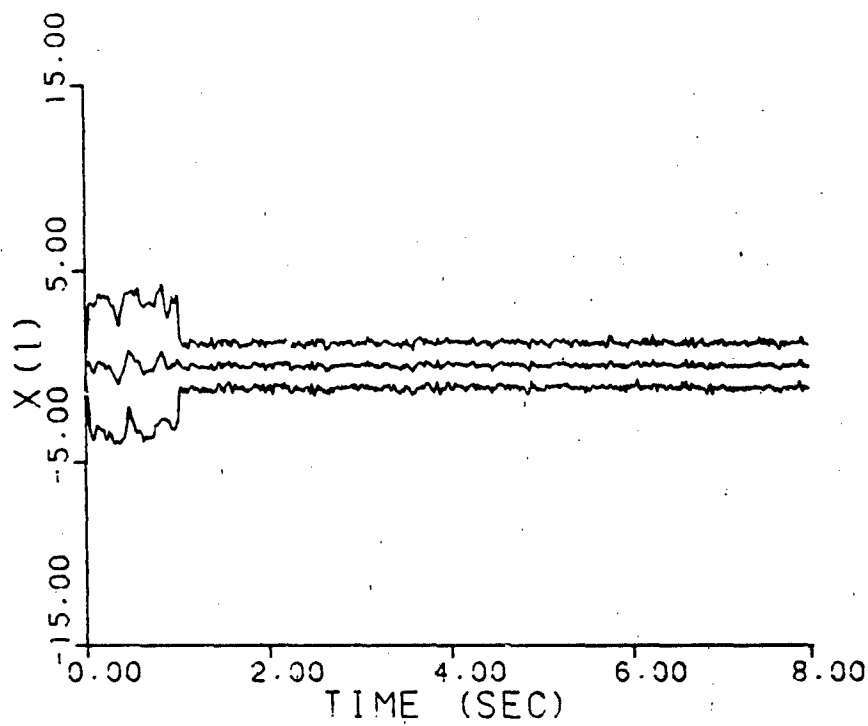


Figure A-17g. Benchmark Controller, $\underline{a} = (0.93, 41.0)$

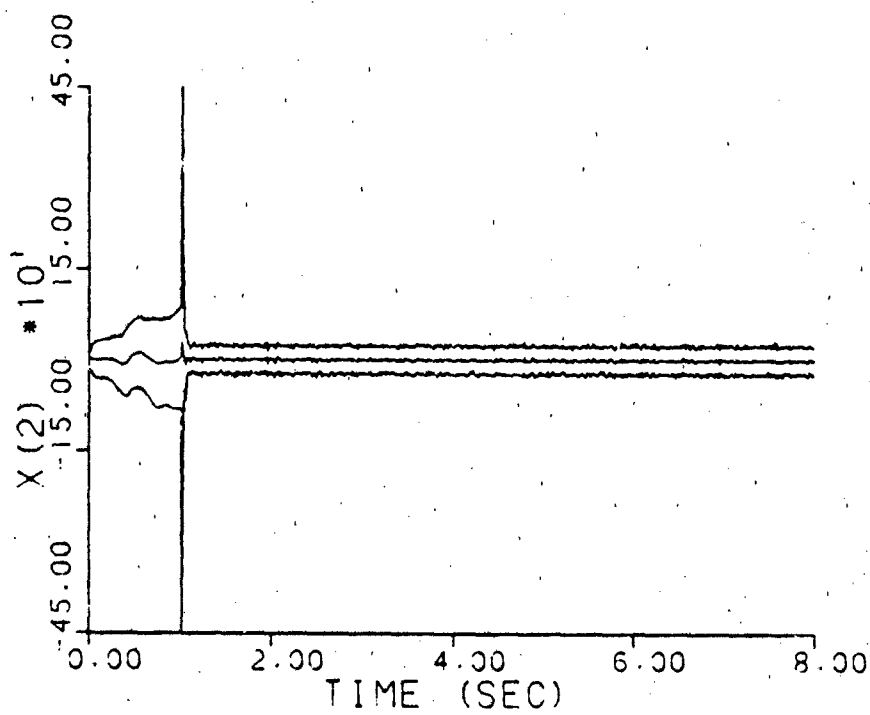
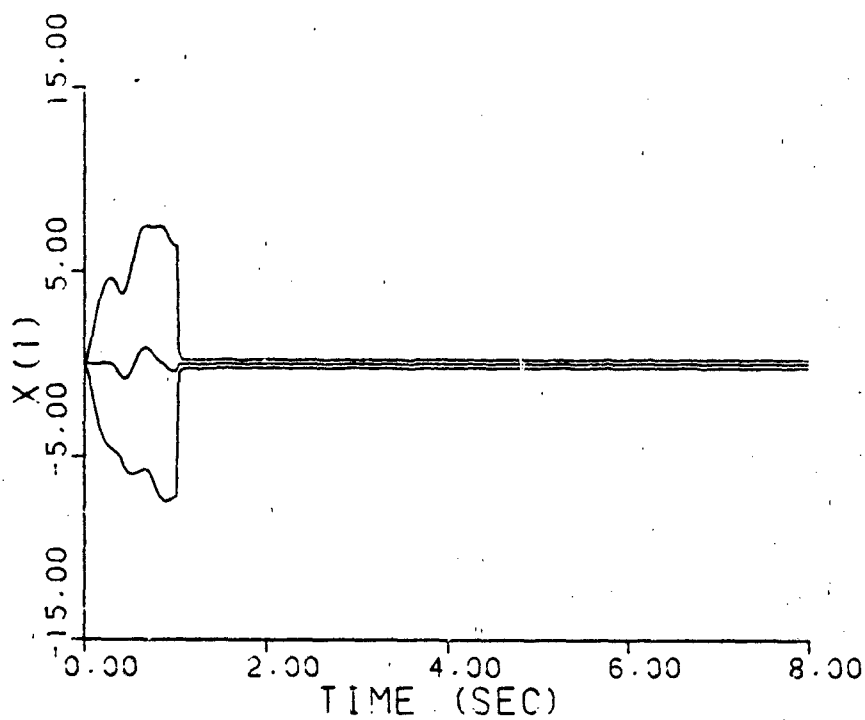


Figure A-18a. Single Changeable-Gain Controller
 $\underline{a} = (1, 3)$

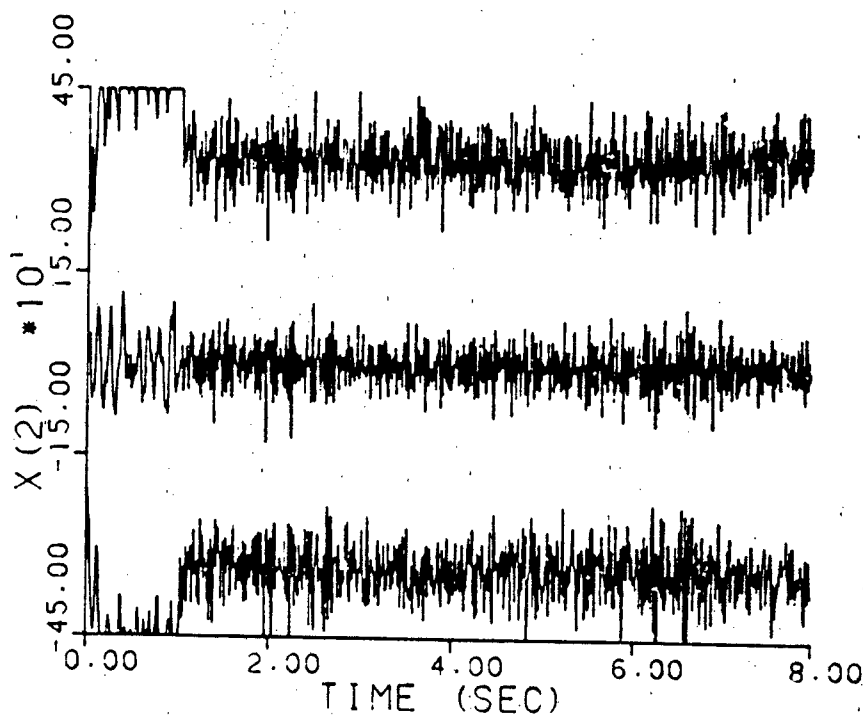
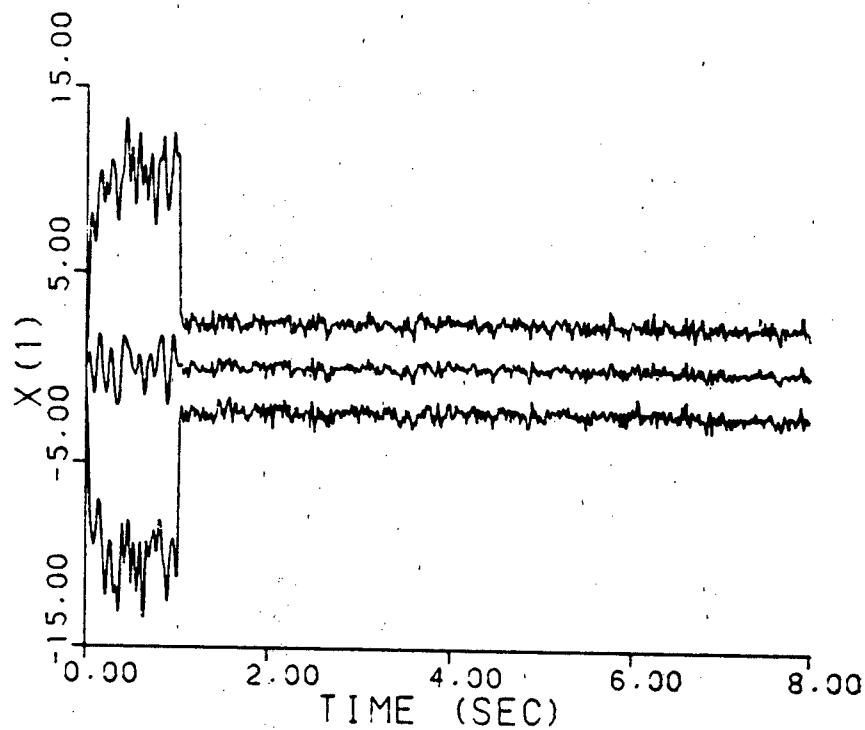


Figure A-18b. Single Changeable-Gain Controller
 $\underline{a} = (2, 9)$

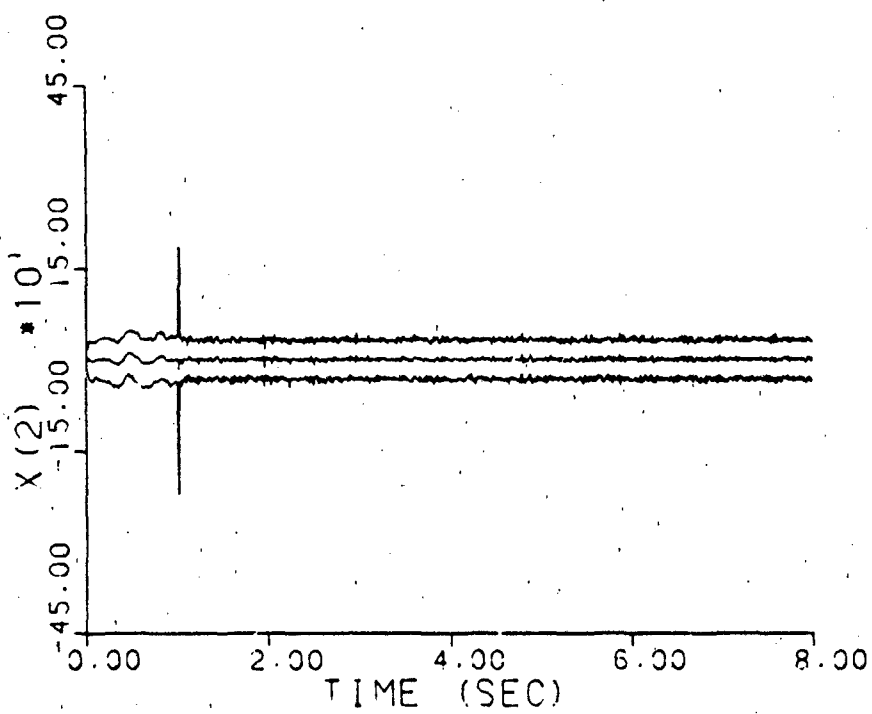
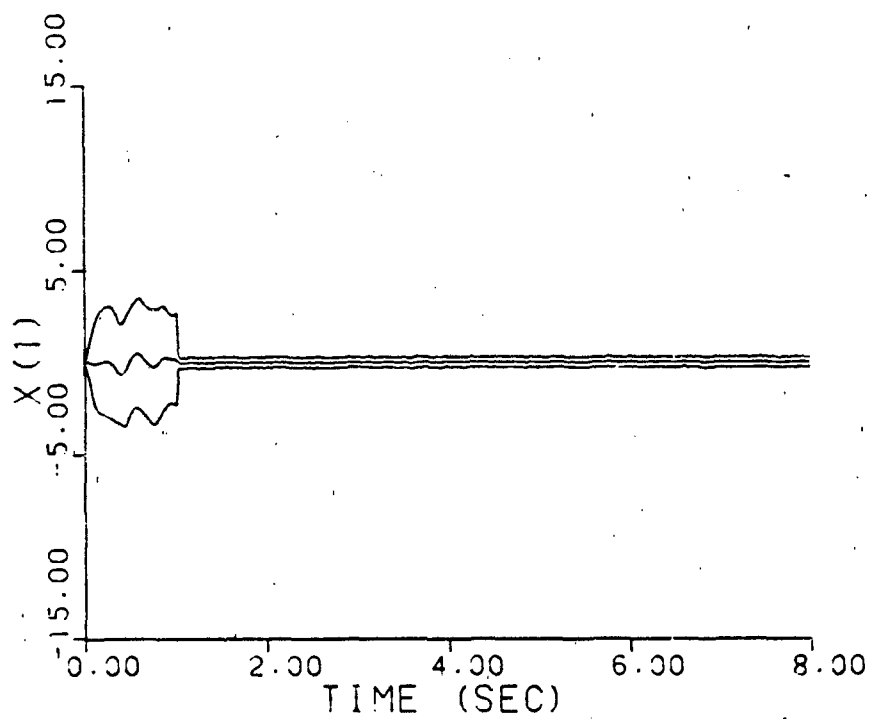


Figure A-13c. Single Changeable-Gain Controller
 $\underline{a} = (5, 4)$

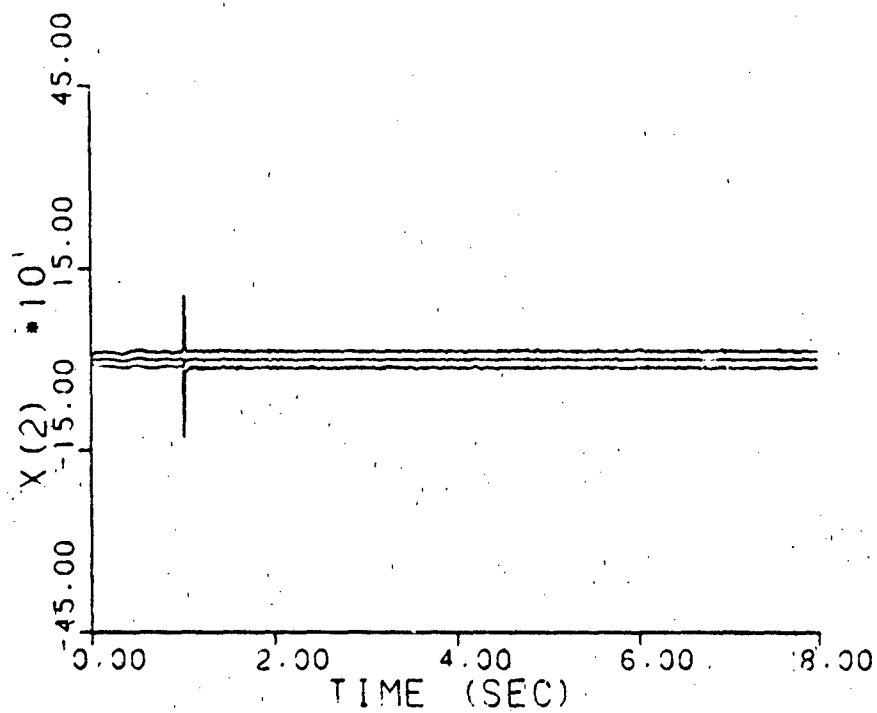
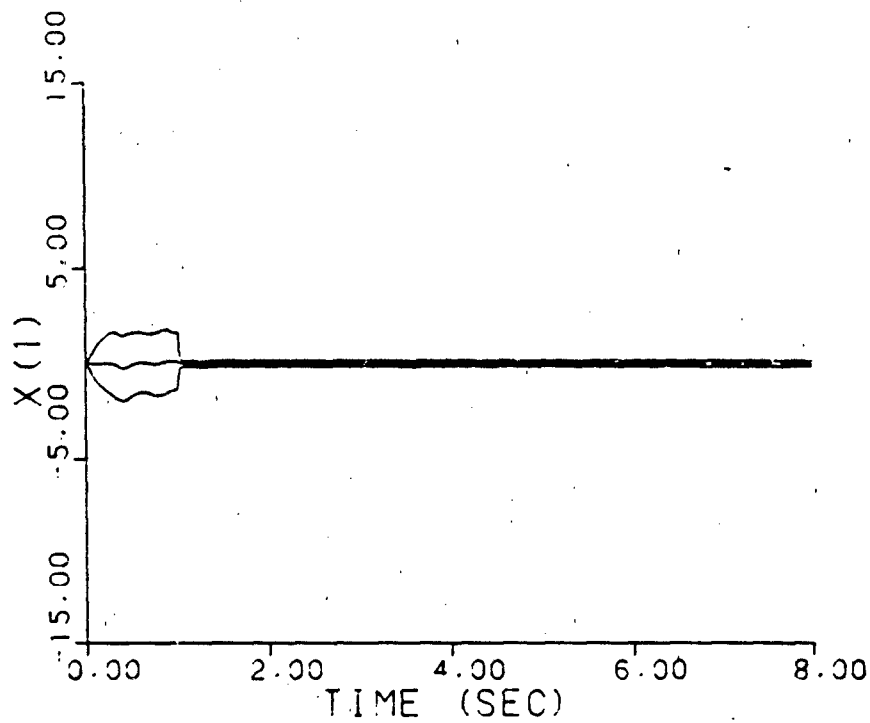


Figure A-18d. Single Changeable-Gain Controller
 $\underline{a} = (9, 2)$

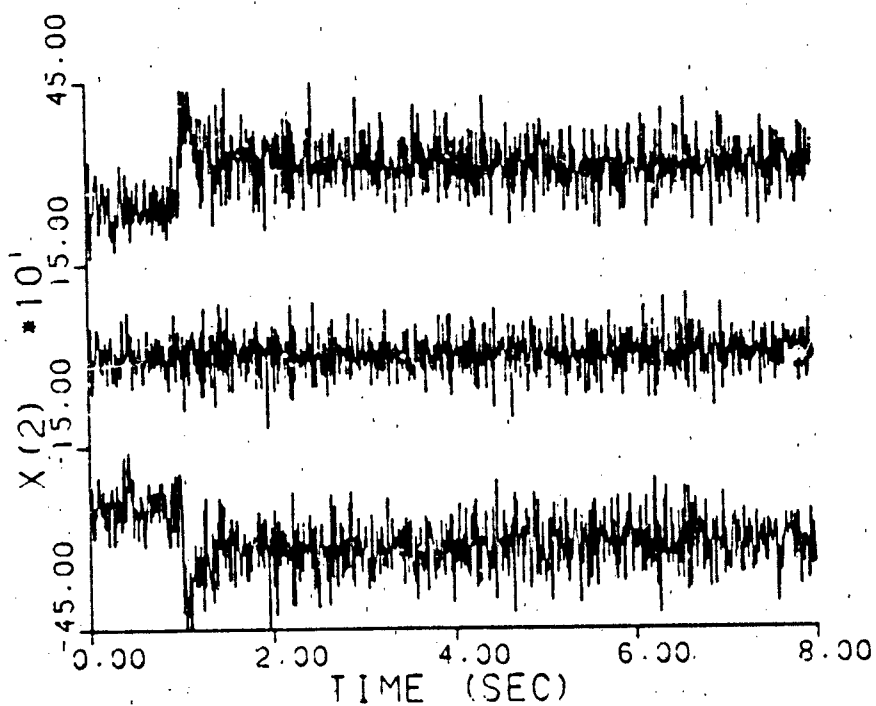
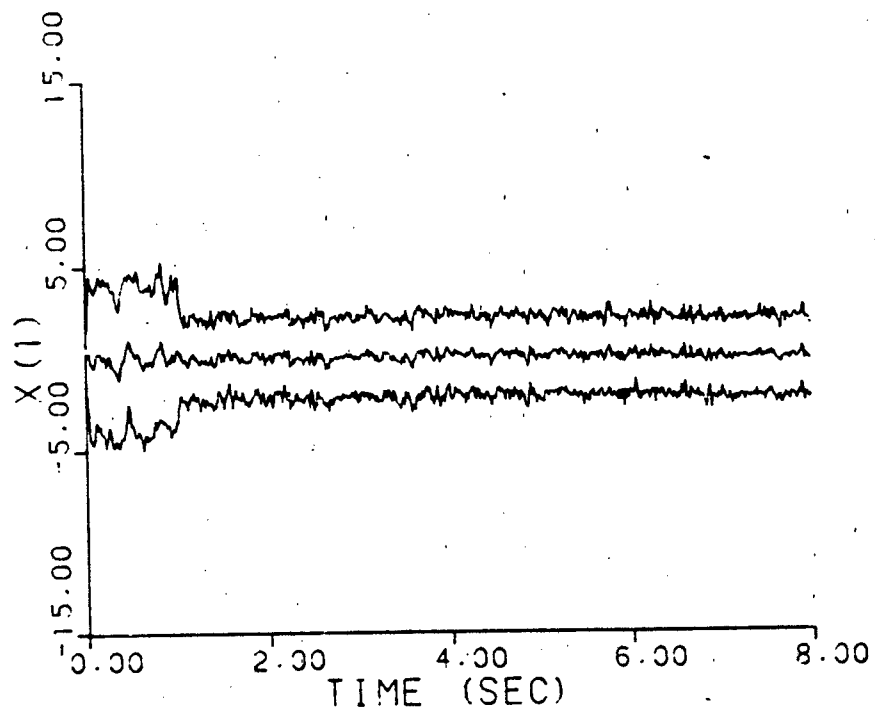


Figure A-18e. Single Changeable-Gain Controller
 $a = (10, 10)$

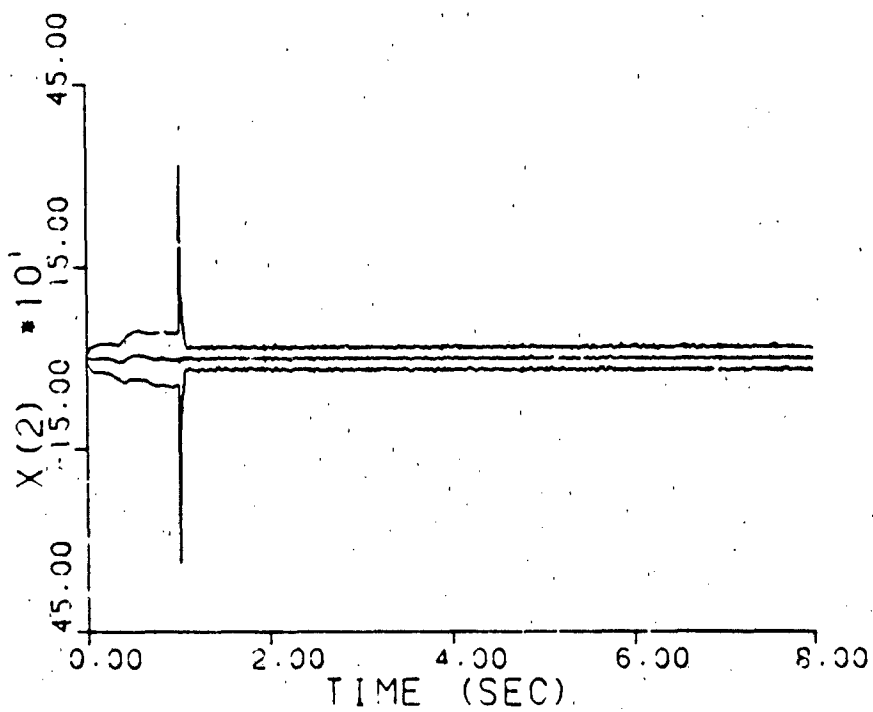
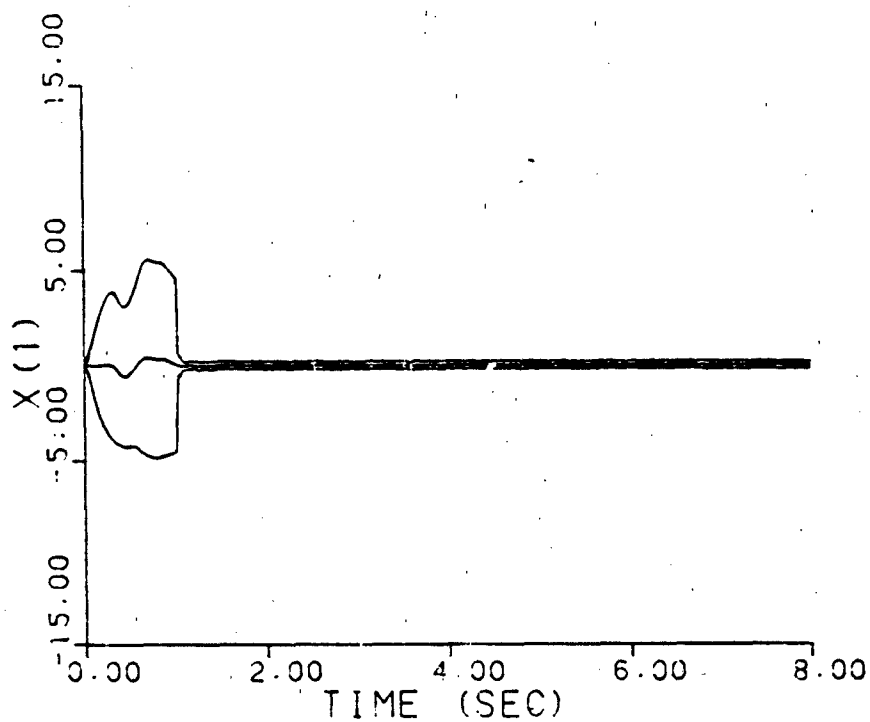


Figure A-18f. Single Changeable-Gain Controller
 $\underline{a} = (0.07, 9.0)$

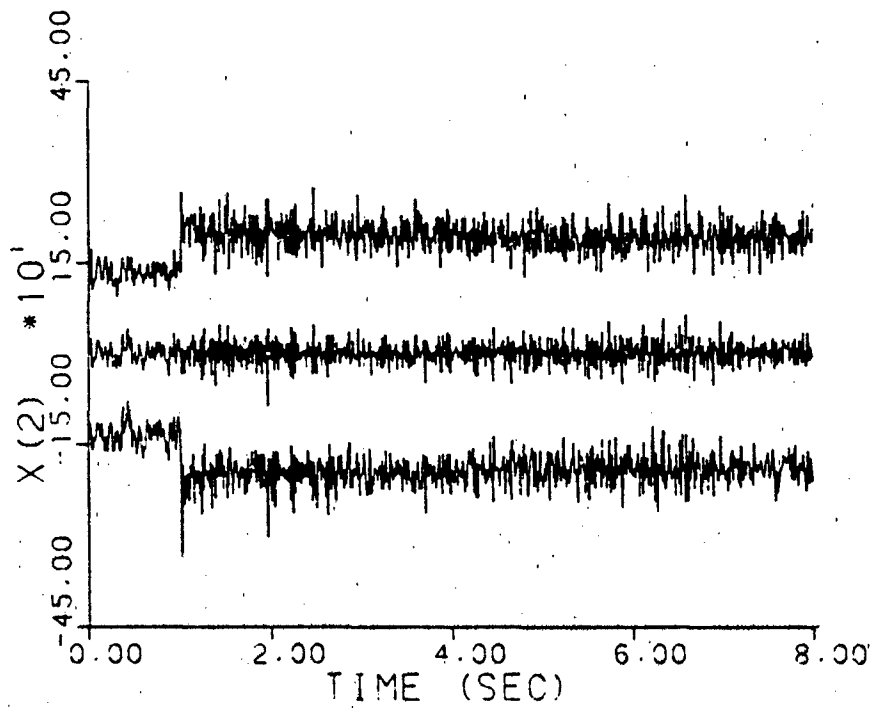
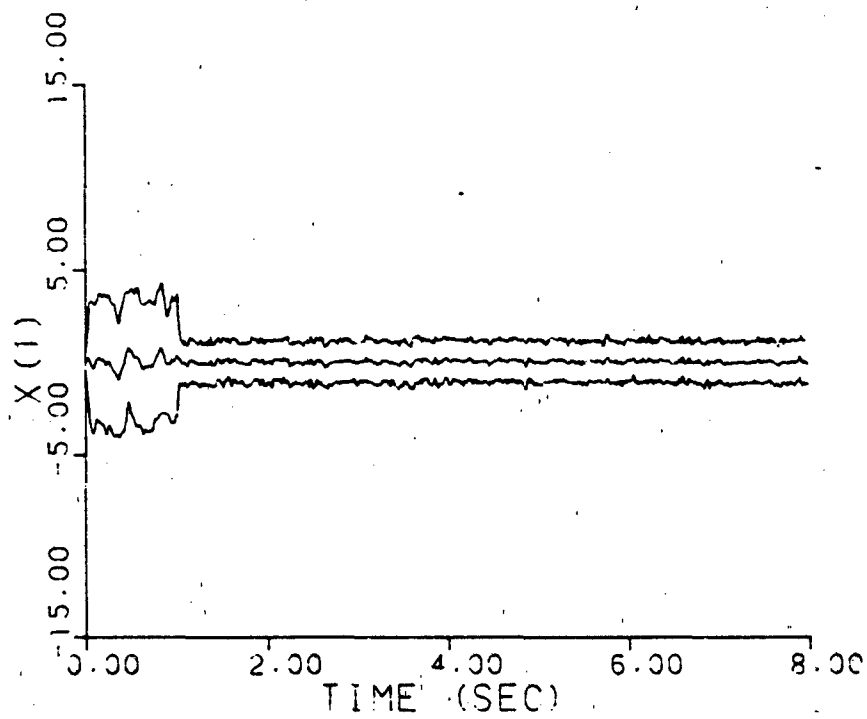


Figure A-18g. Single Changeable-Gain Controller
 $\underline{a} = (0.93, 41.0)$

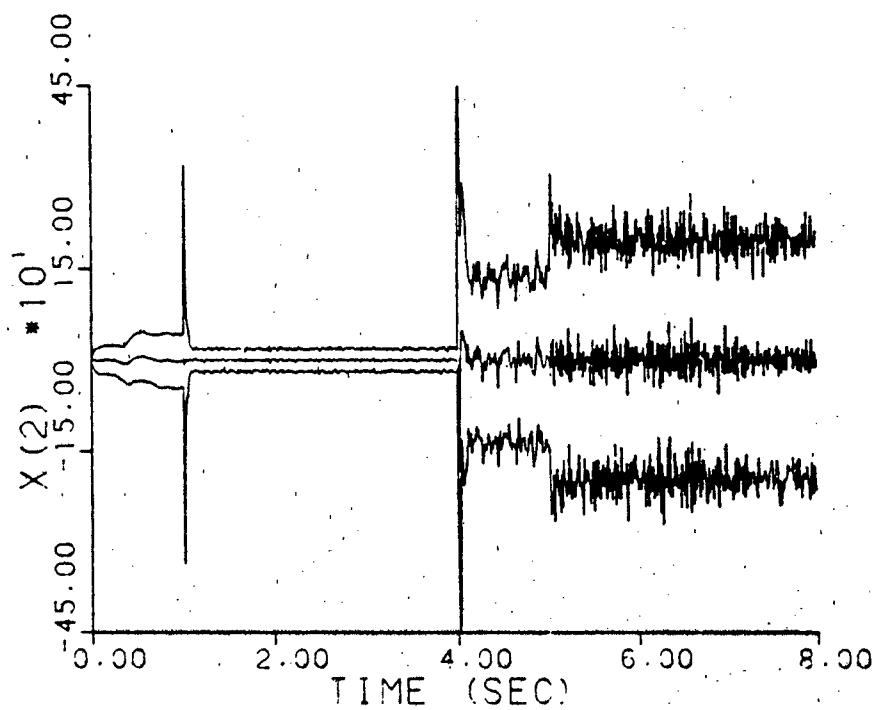
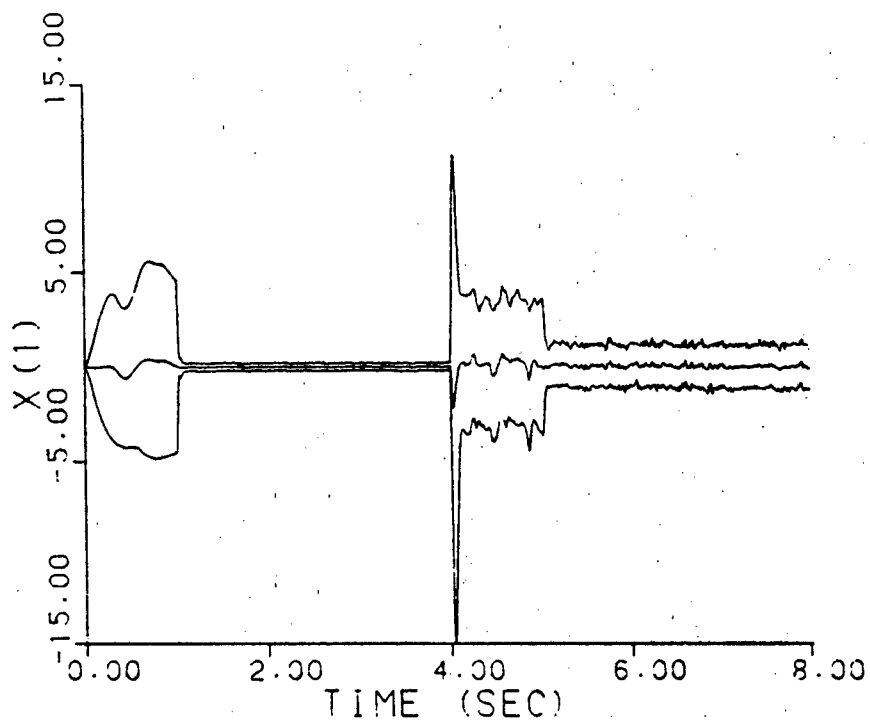


Figure A-18h. Single Changeable-Gain Controller
a jumps: (0.07, 9.0) - (0.93, 41.0)

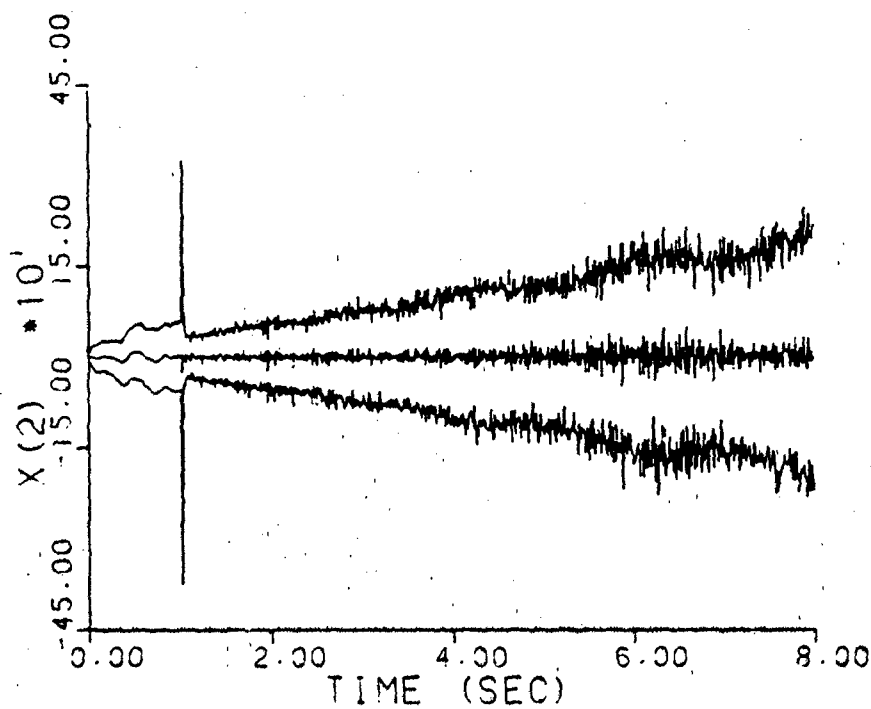
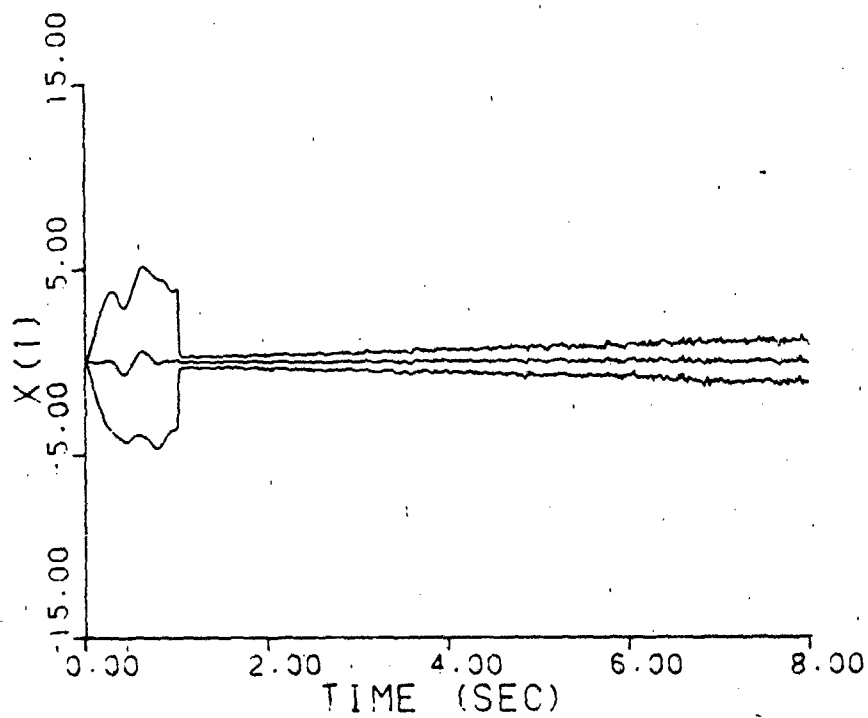


Figure A-18i. Single Changeable-Gain Controller
a varies: (0.07, 9.0) - (0.93, 41.0)

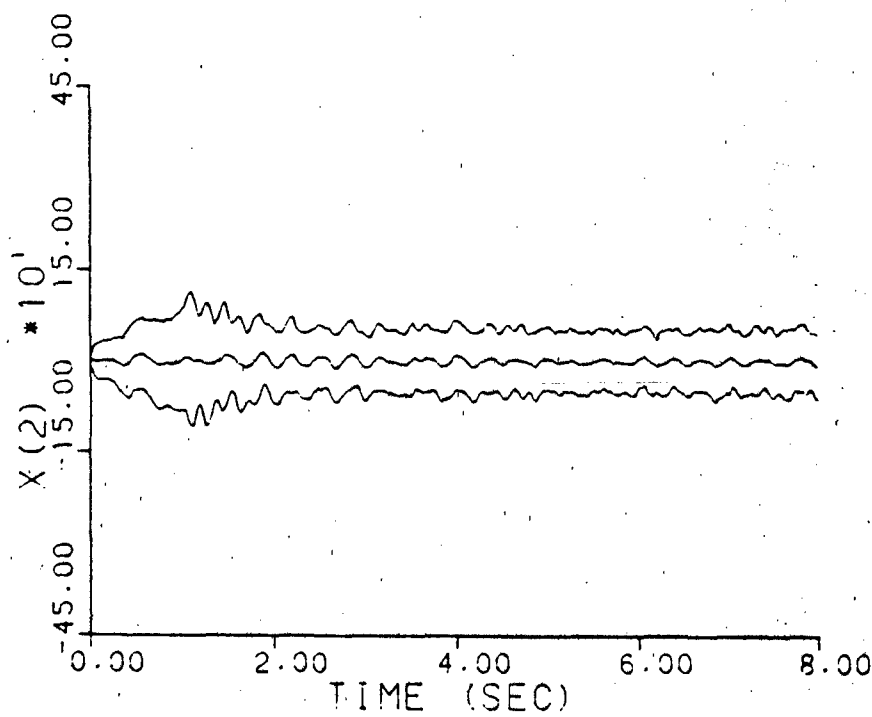
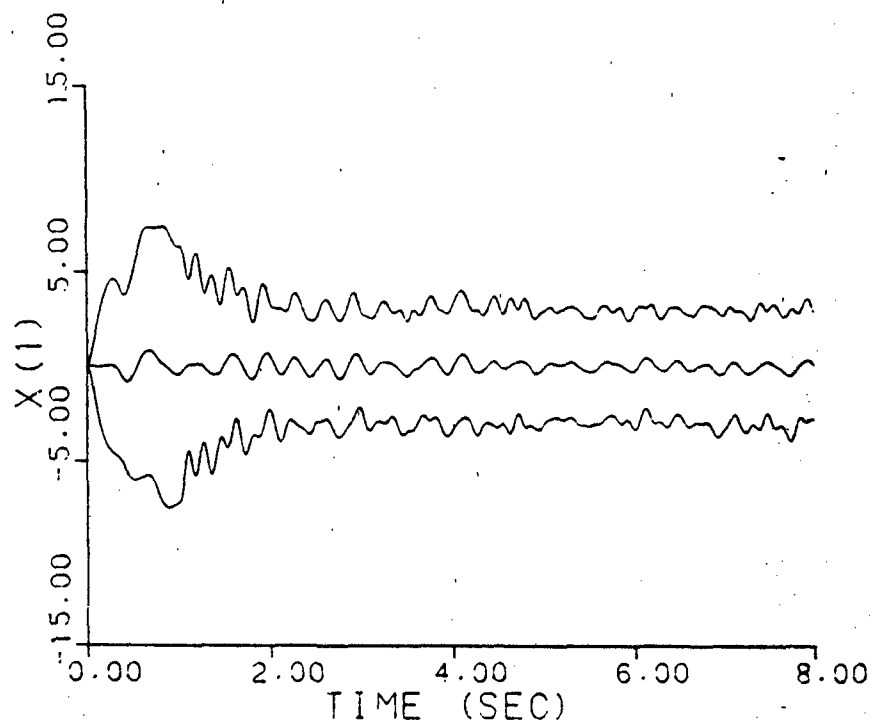


Figure A-19a. Single Fixed-Gain Controller
 $\underline{a} = (1, 3)$

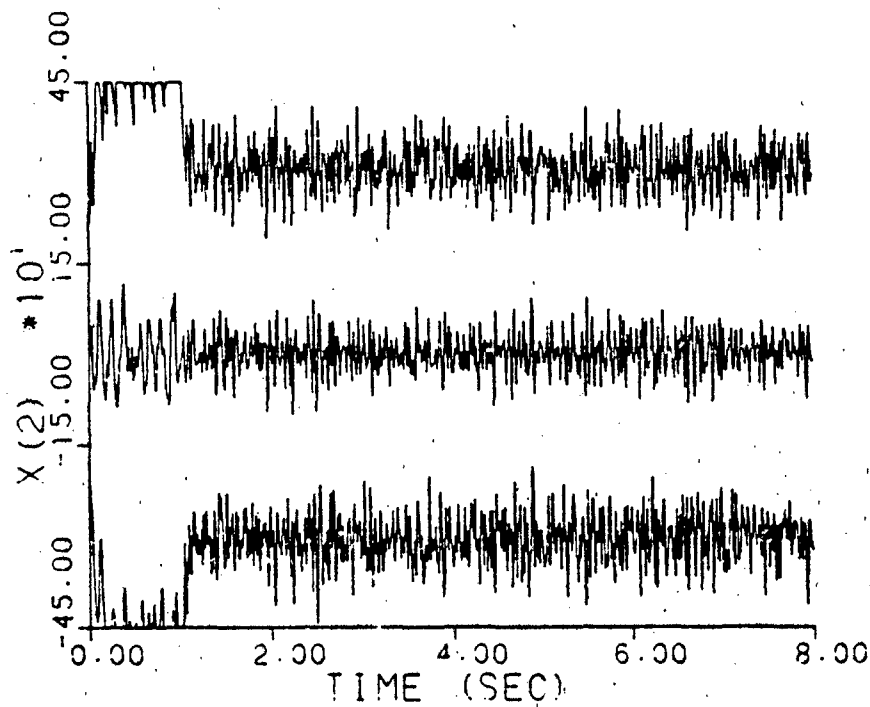
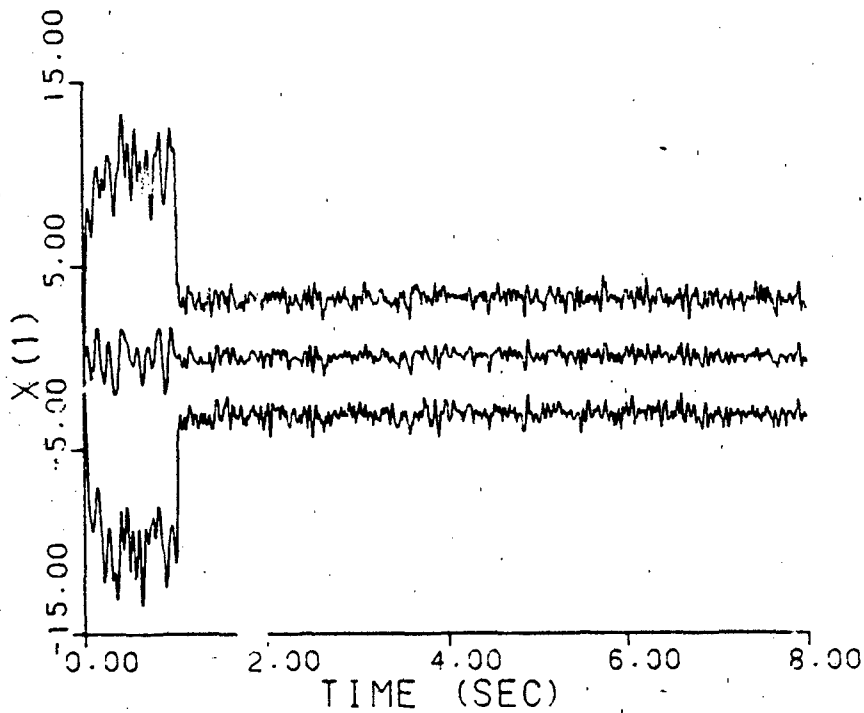


Figure A-19b. Single Fixed-Gain Controller
 $\underline{a} = (2, 9)$

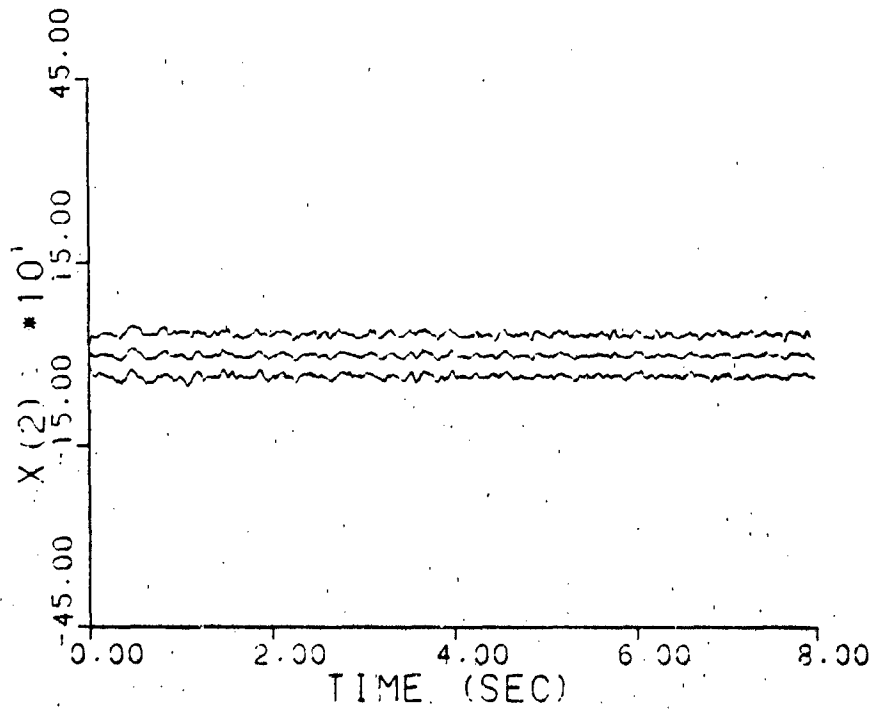
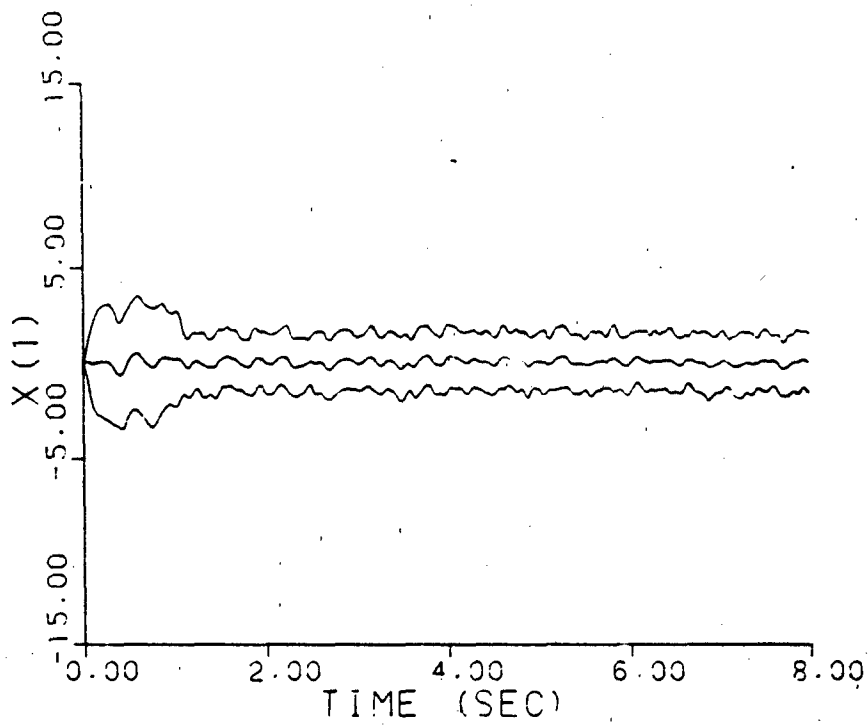


Figure A-19c. Single Fixed-Gain Controller
 $\underline{a} = (5, 4)$

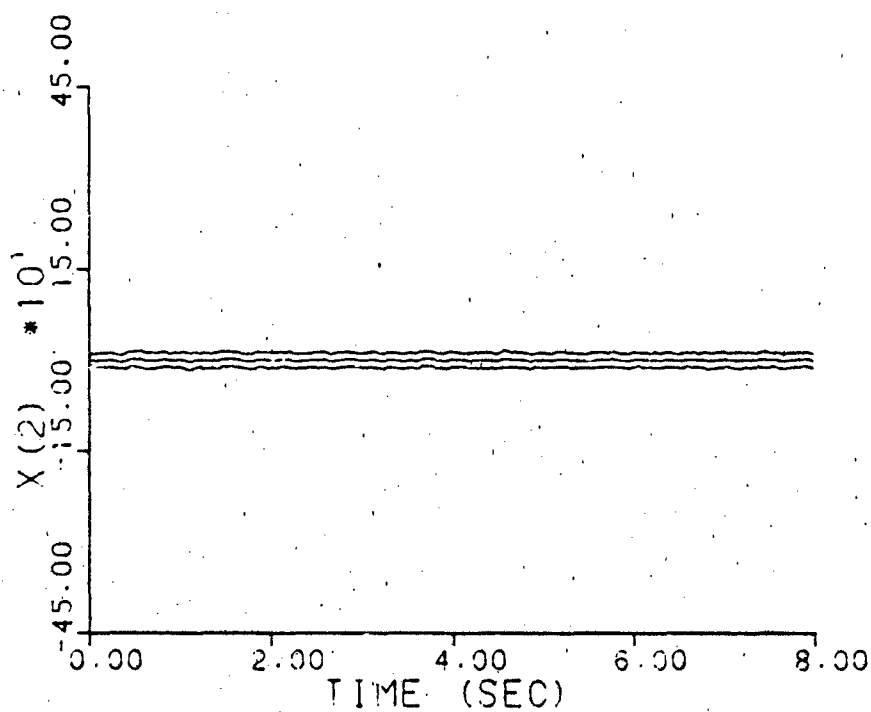
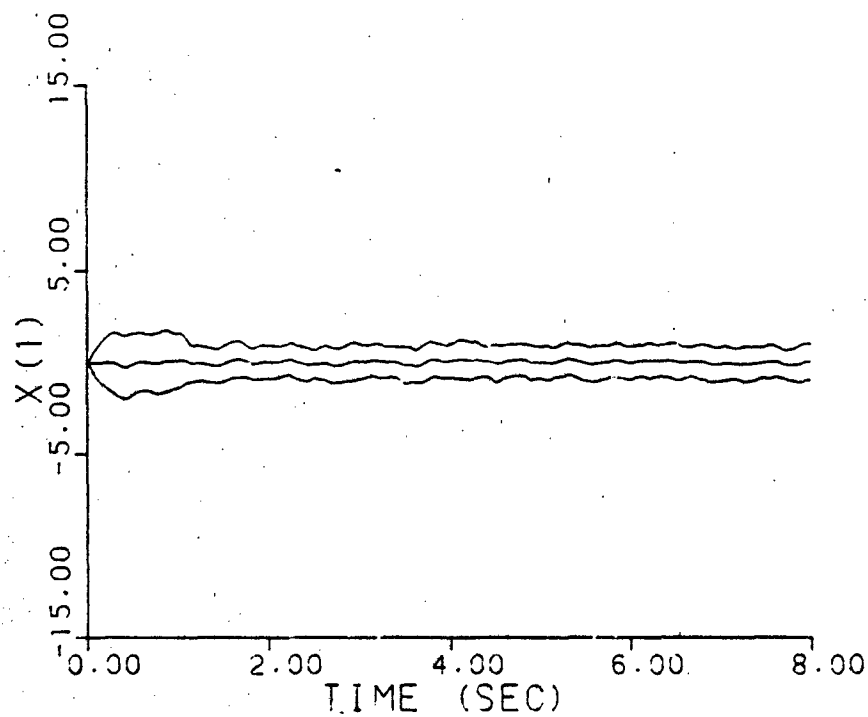


Figure A-19d. Single Fixed-Gain Controller
 $\underline{a} = (9, 2)$

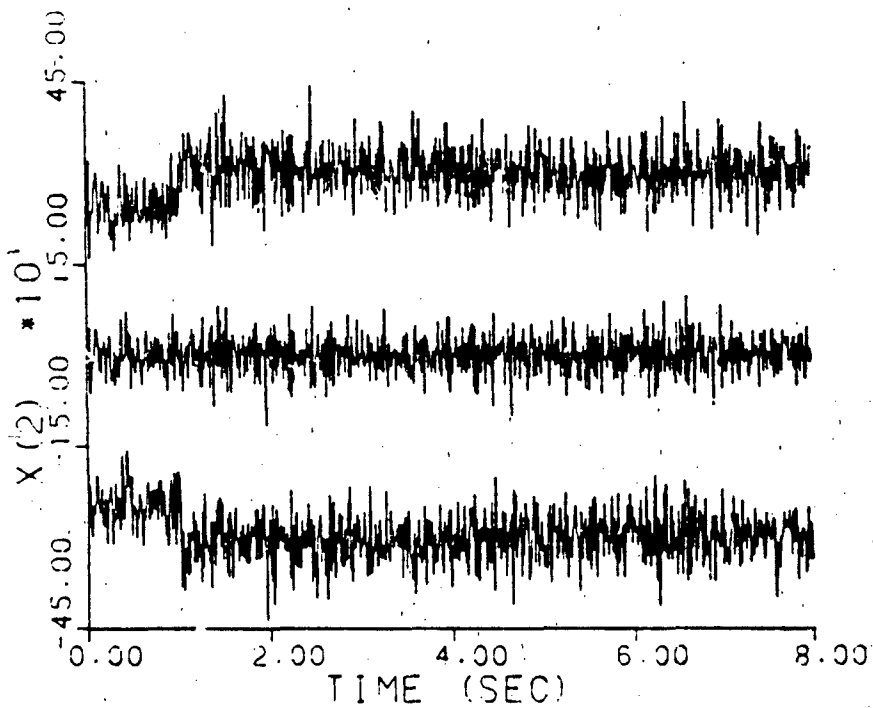
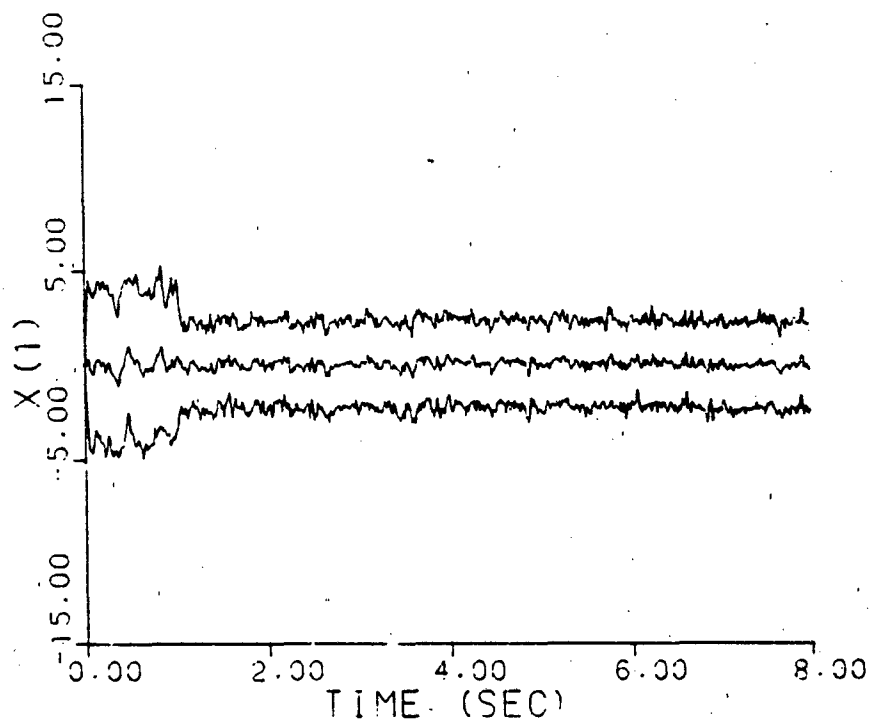


Figure A-19e. Single Fixed-Gain Controller
 $\underline{a} = (10, 10)$

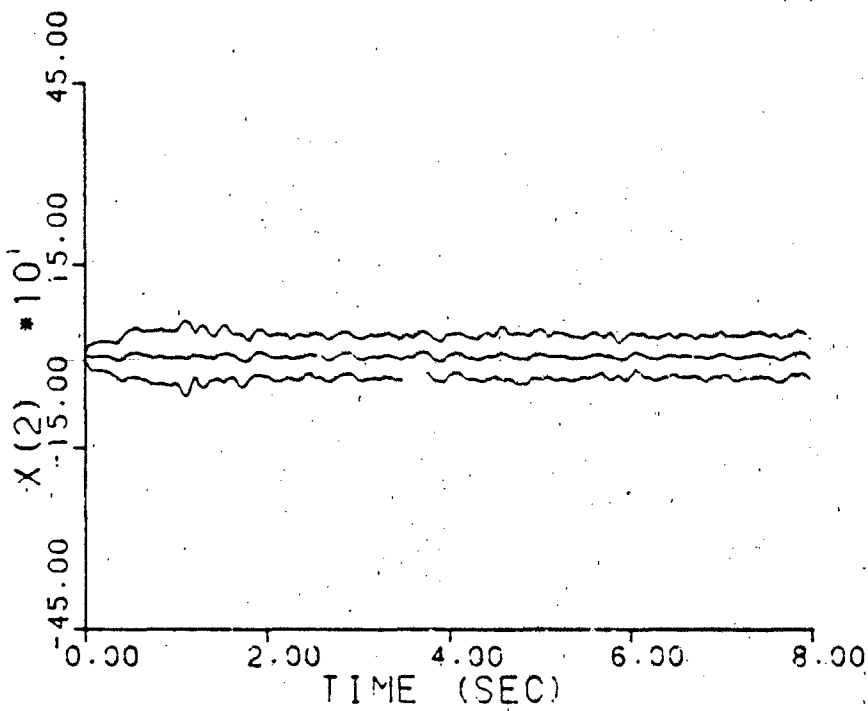
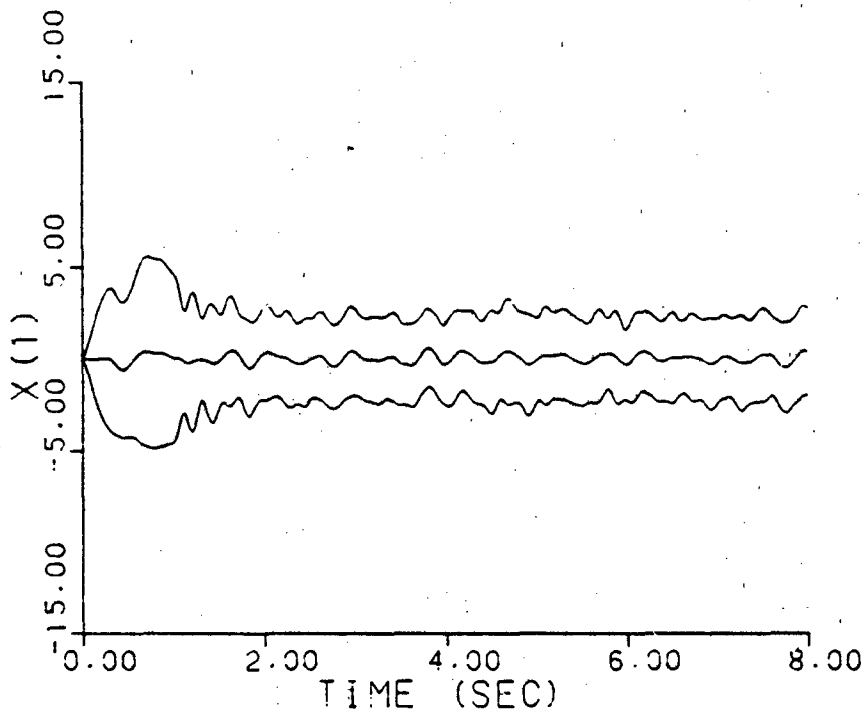


Figure A-19f. Single Fixed-Gain Controller
 $\underline{a} = (0.07, 9.0)$

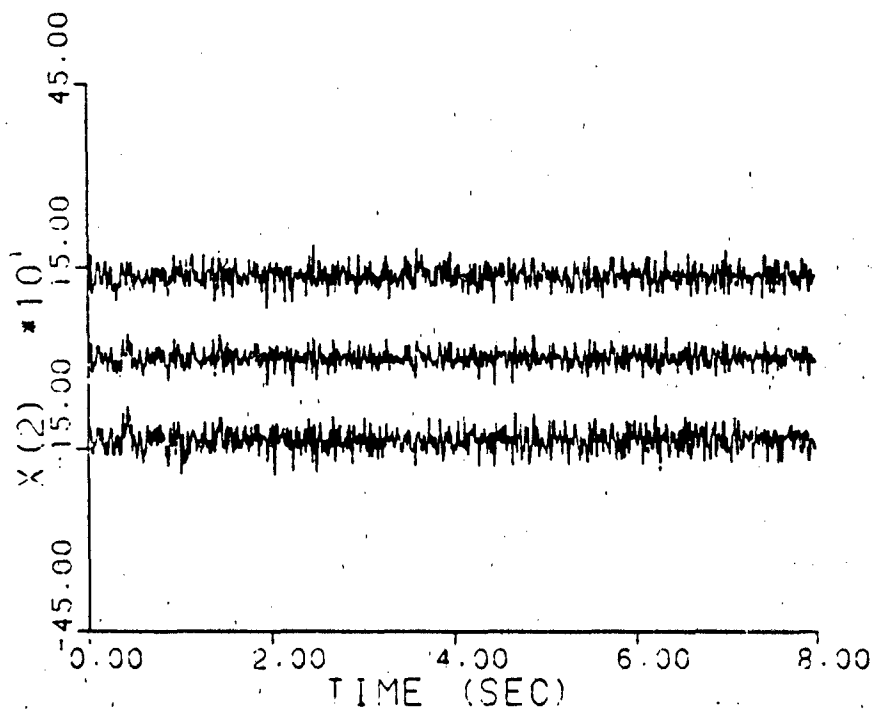
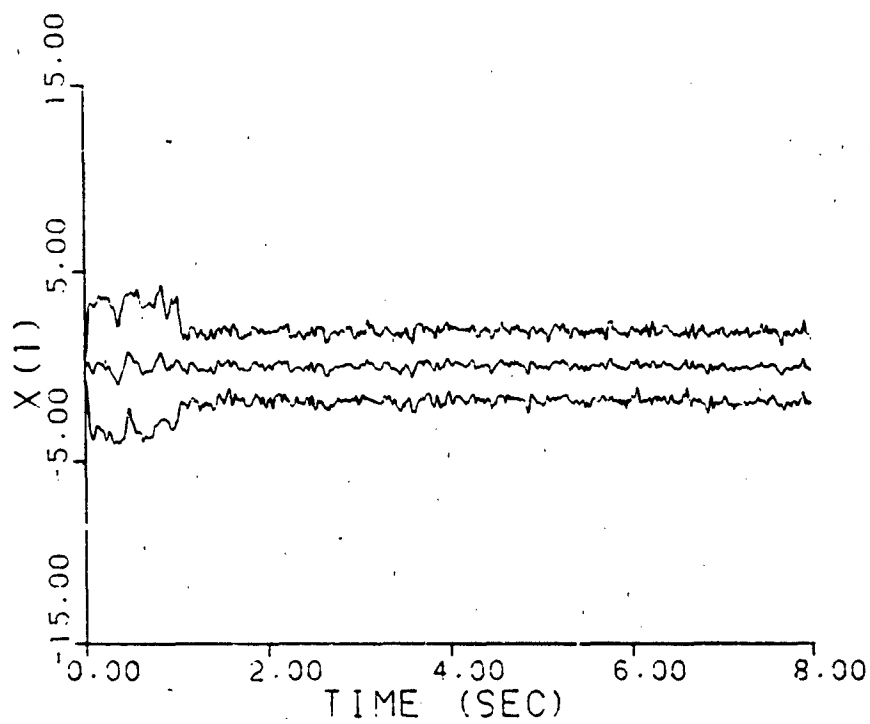


Figure A-19g. Single Fixed-Gain Controller
 $\underline{a} = (0.93, 41.0)$

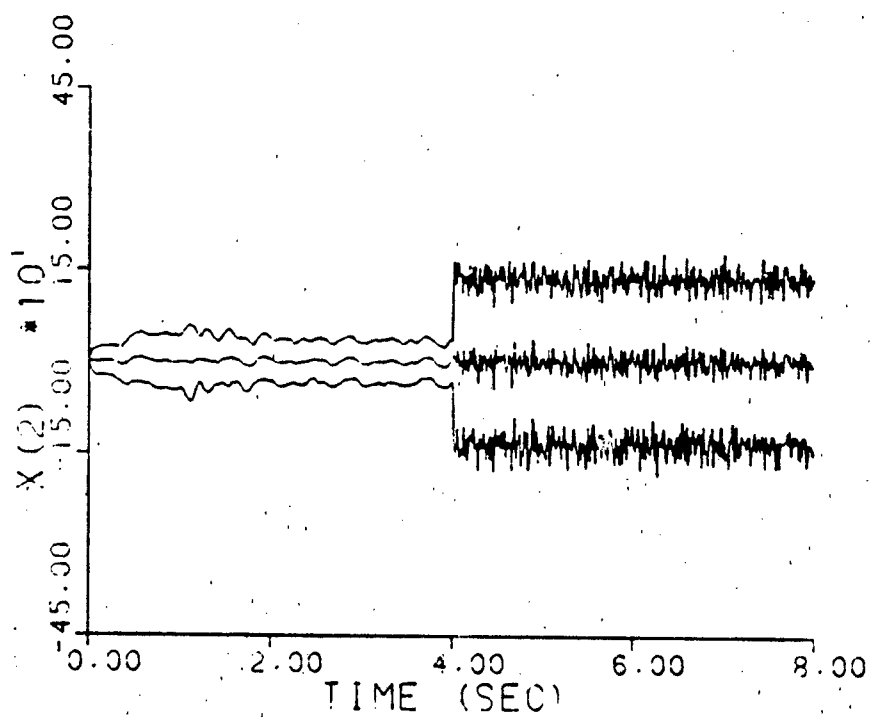
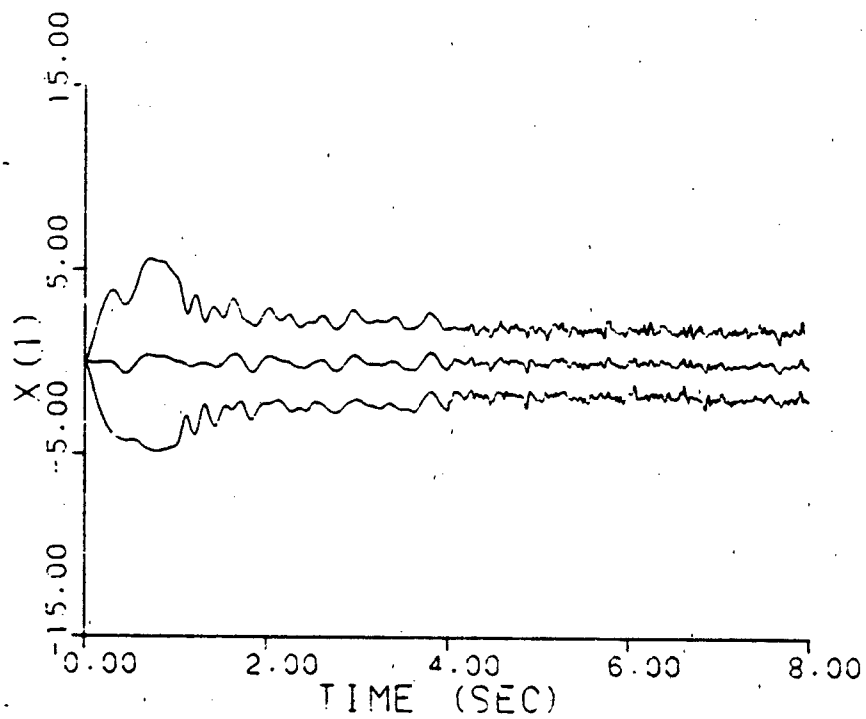


Figure A-19h. Single Fixed-Gain Controller
a jumps: (0.07, 9.0) - (0.93, 41.0)

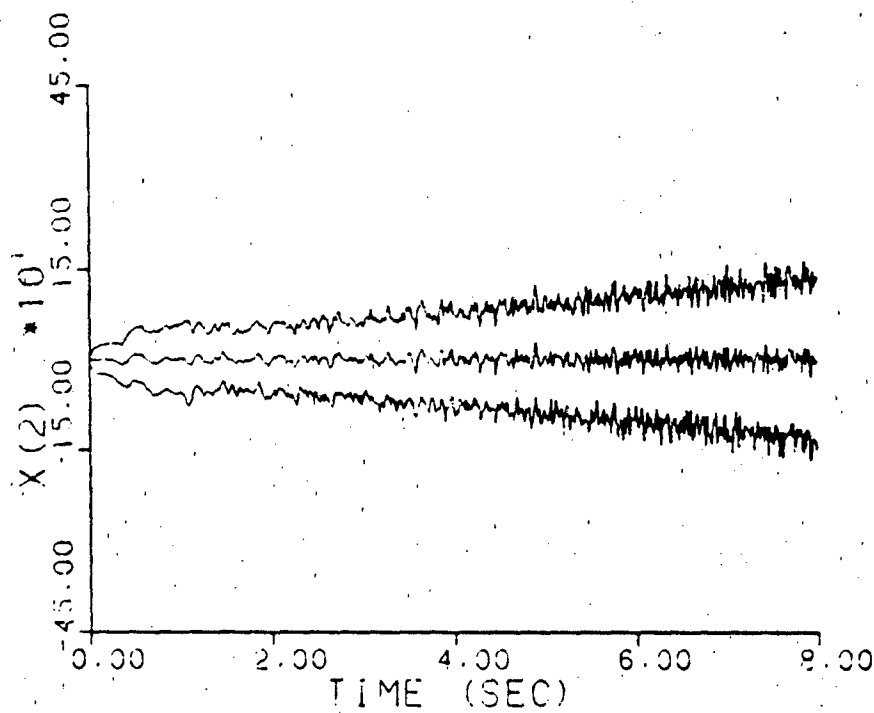
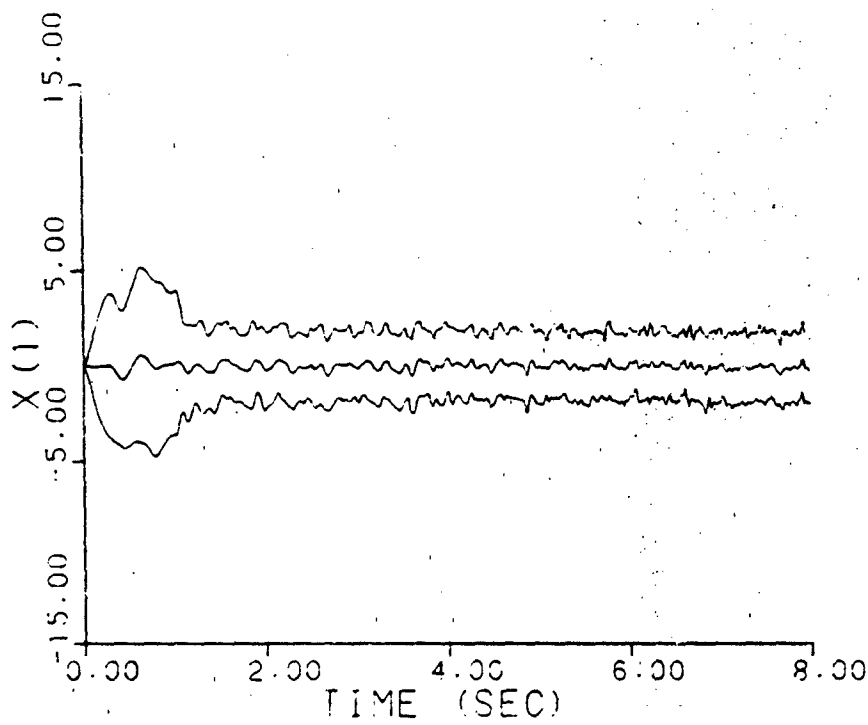


Figure A-19i. Single Fixed-Gain Controller
a varies: (0.07, 9.0) - (0.93, 41.0)

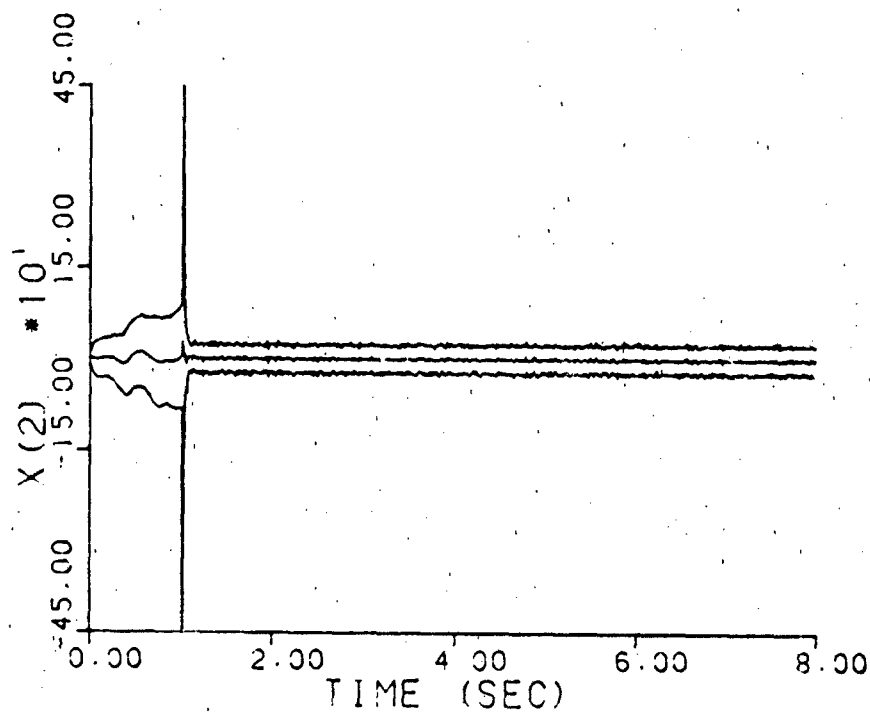
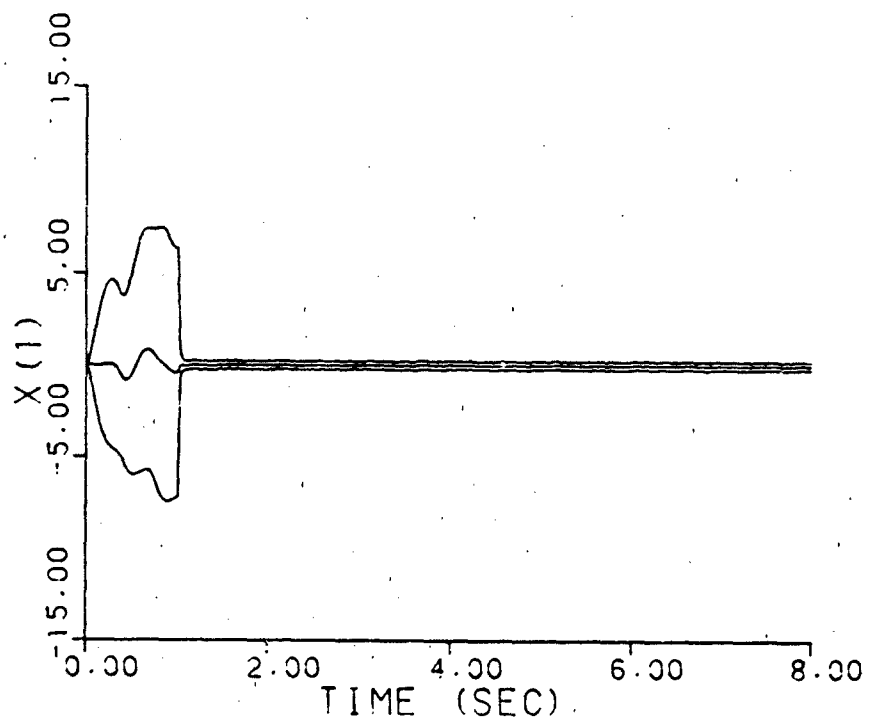


Figure A-2Ca. Sliding Bank Multiple Model Adaptive Controller
 $\underline{a} = (1, 3)$

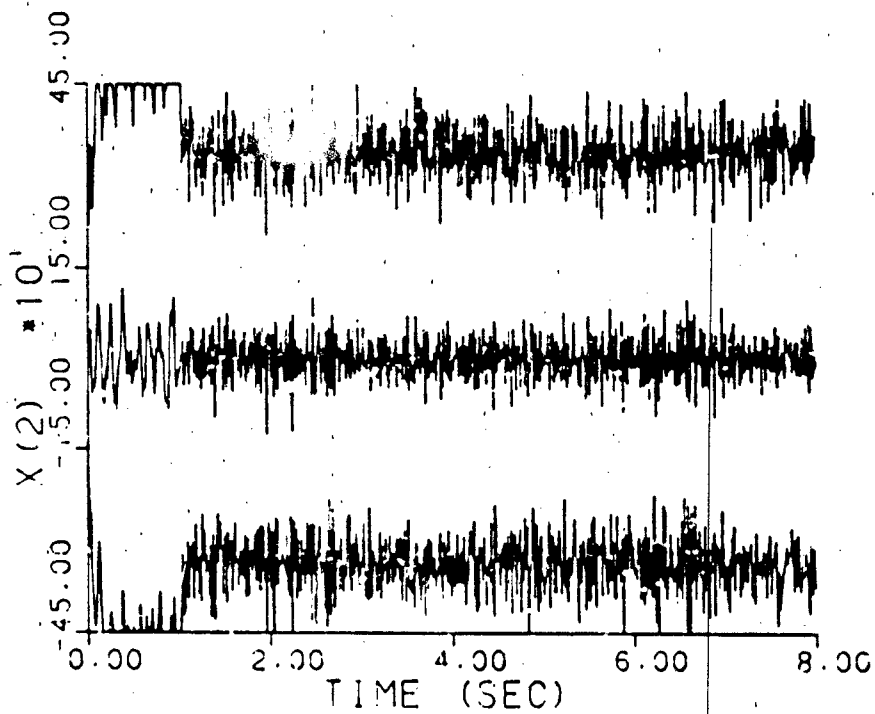
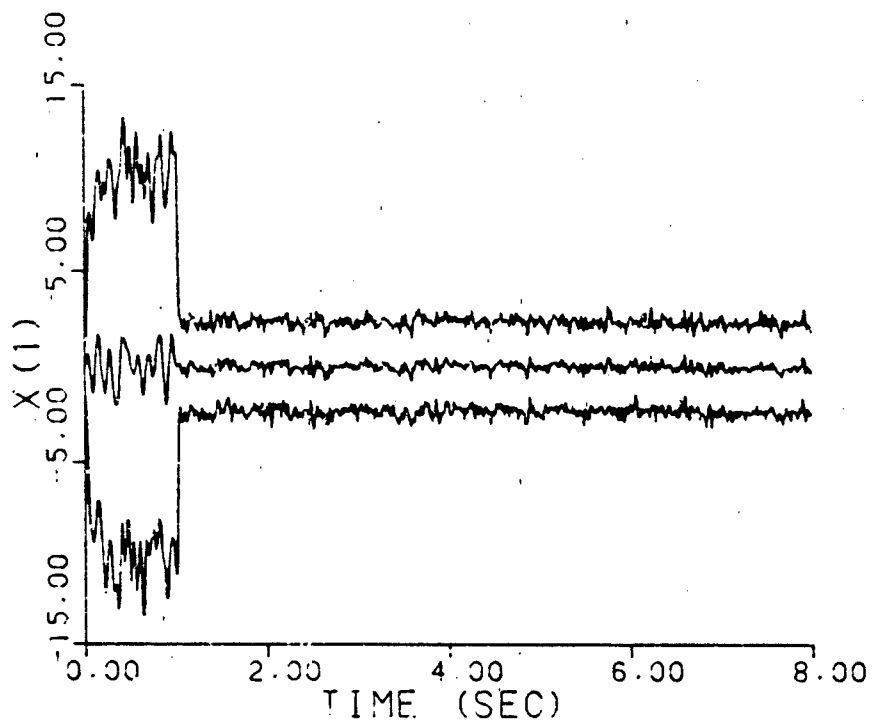


Figure A-20b. Sliding Bank Multiple Model Adaptive Controller
 $\underline{a} = (2, 9)$

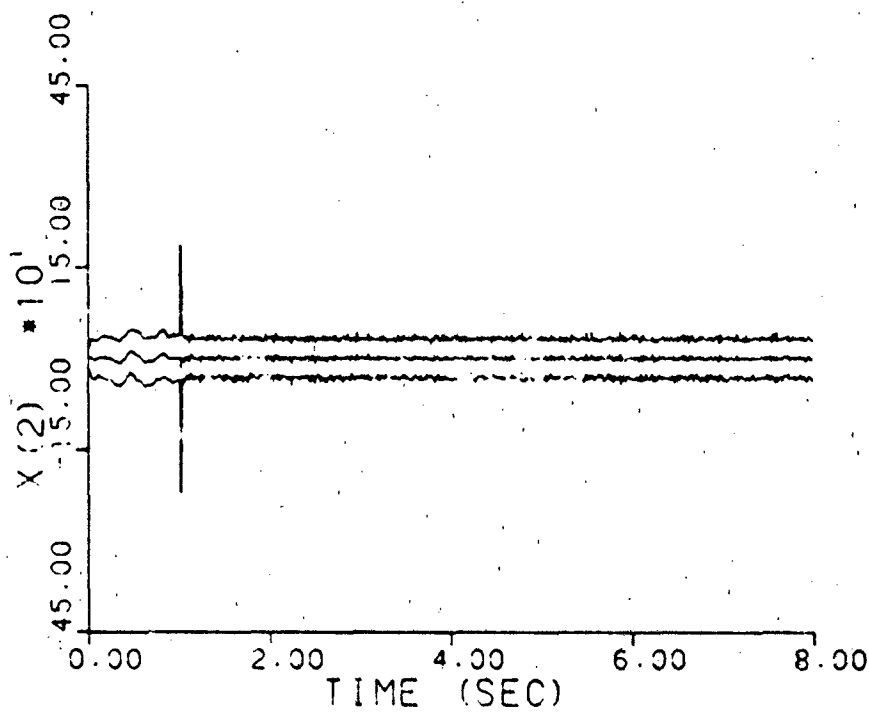
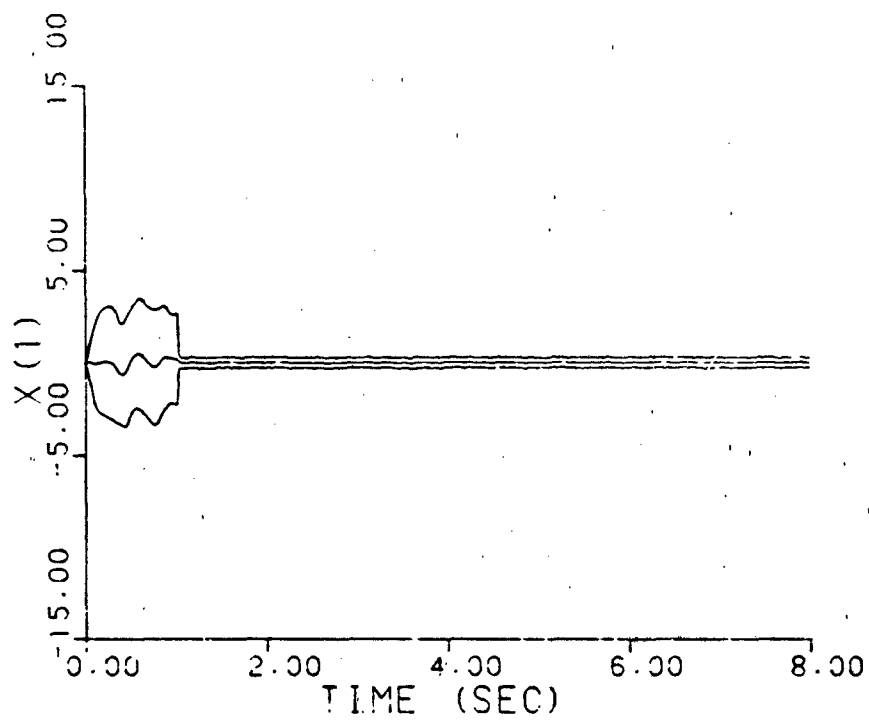


Figure A-20c. Sliding Bank Multiple Model Adaptive Controller
 $\underline{a} = (5, 4)$

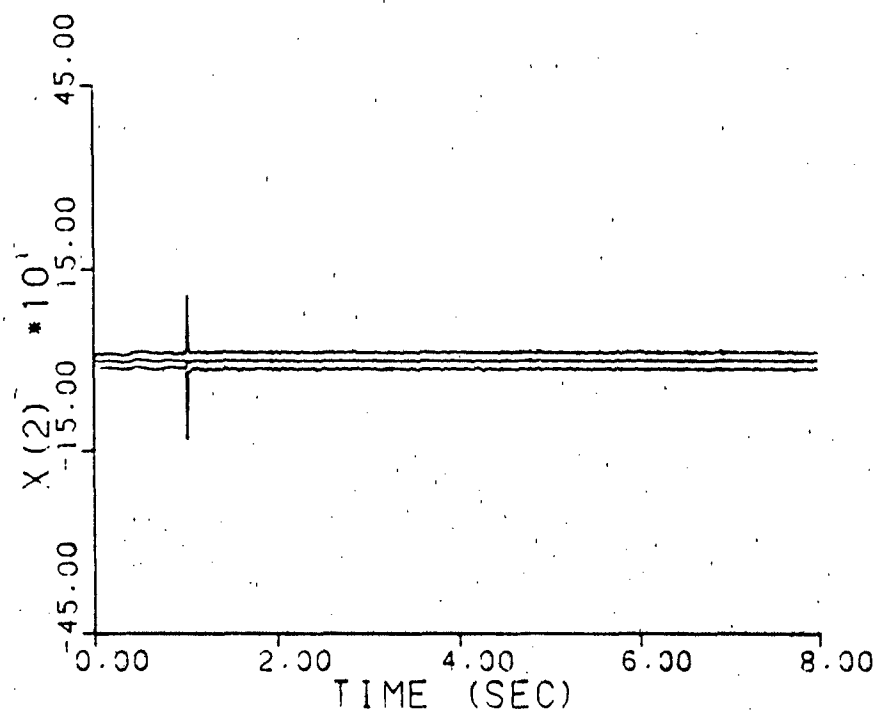
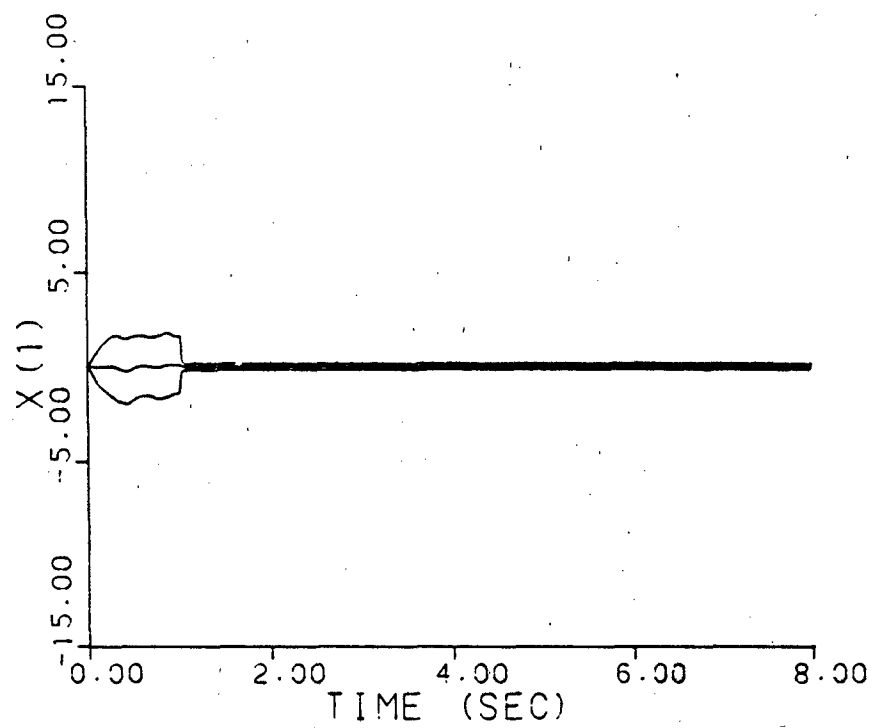


Figure A-20d. Sliding Bank Multiple Model Adaptive Controller
 $\underline{a} = (9, 2)$

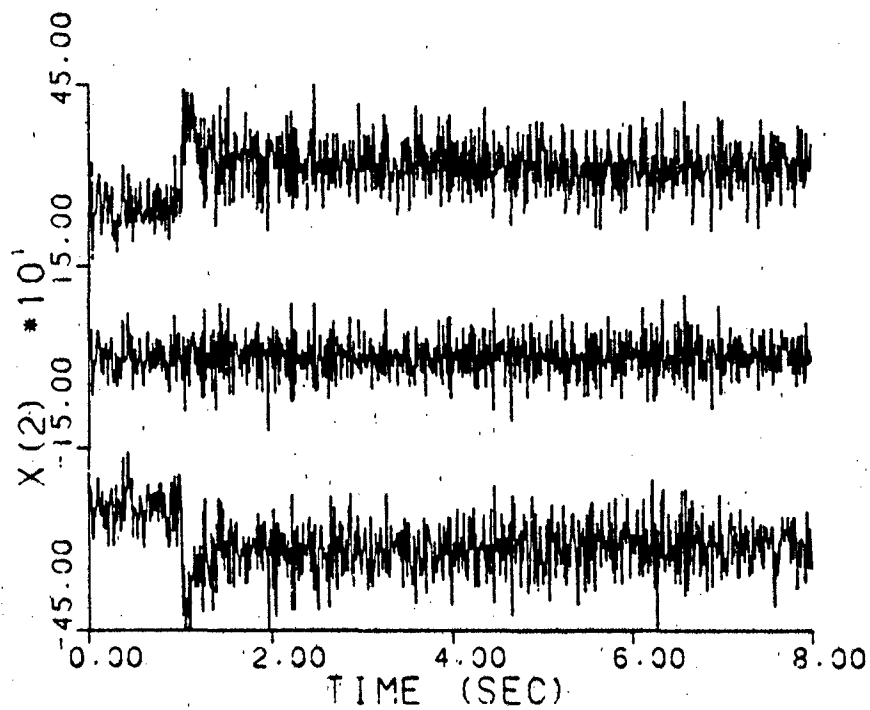
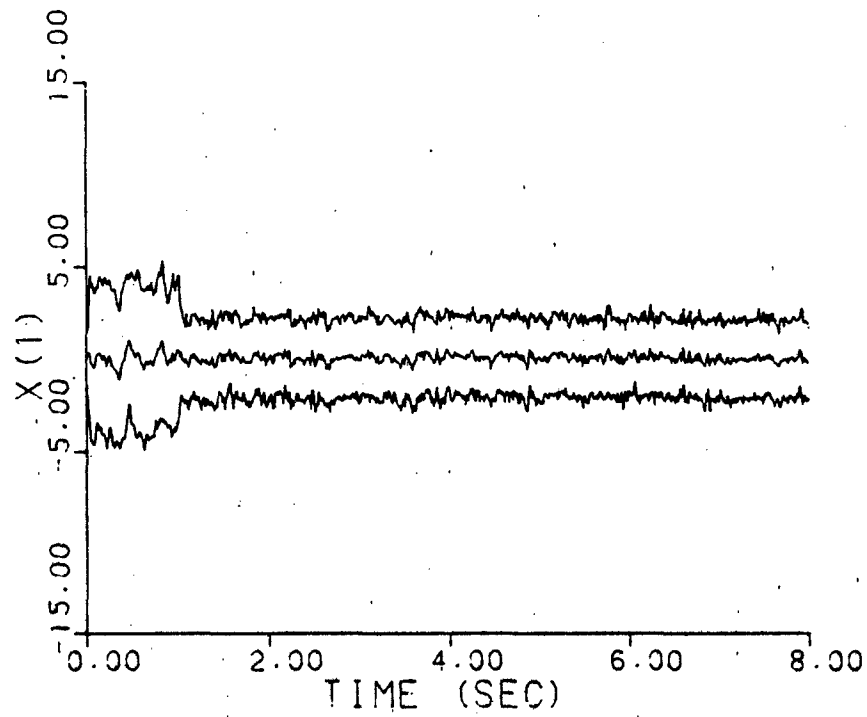


Figure A-20e. Sliding Bank Multiple Model Adaptive Controller
 $\underline{a} = (10, 10)$

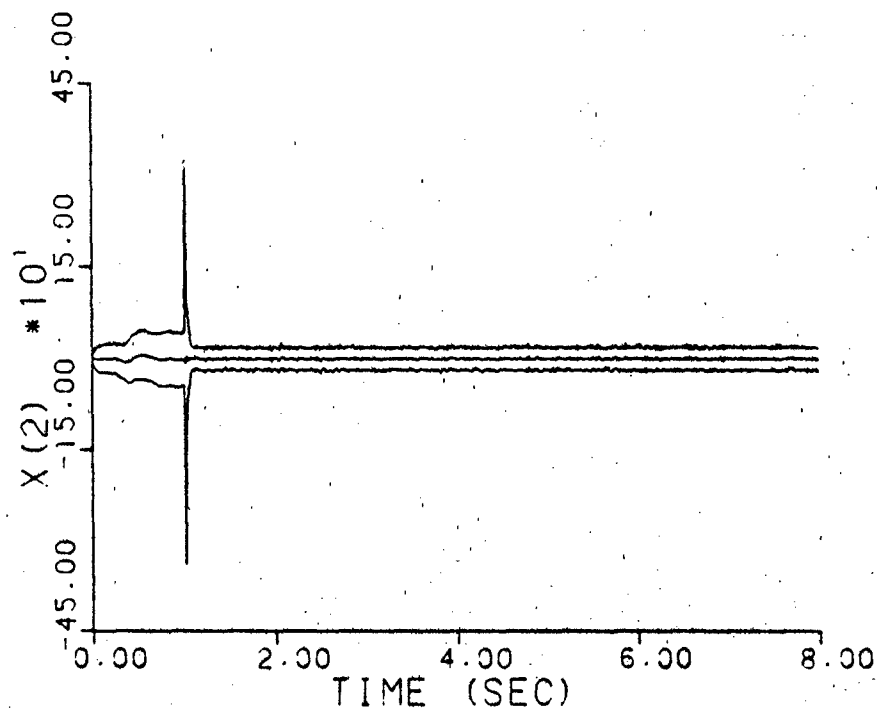
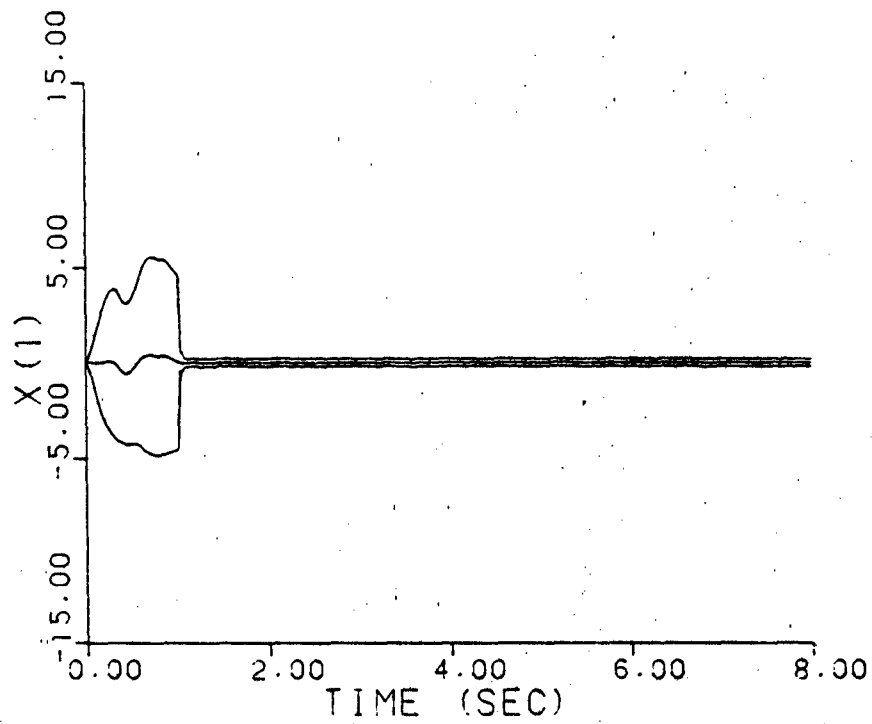


Figure A-20f. Sliding Bank Multiple Model Adaptive Controller
 $\underline{a} = (0.07, 9.0)$

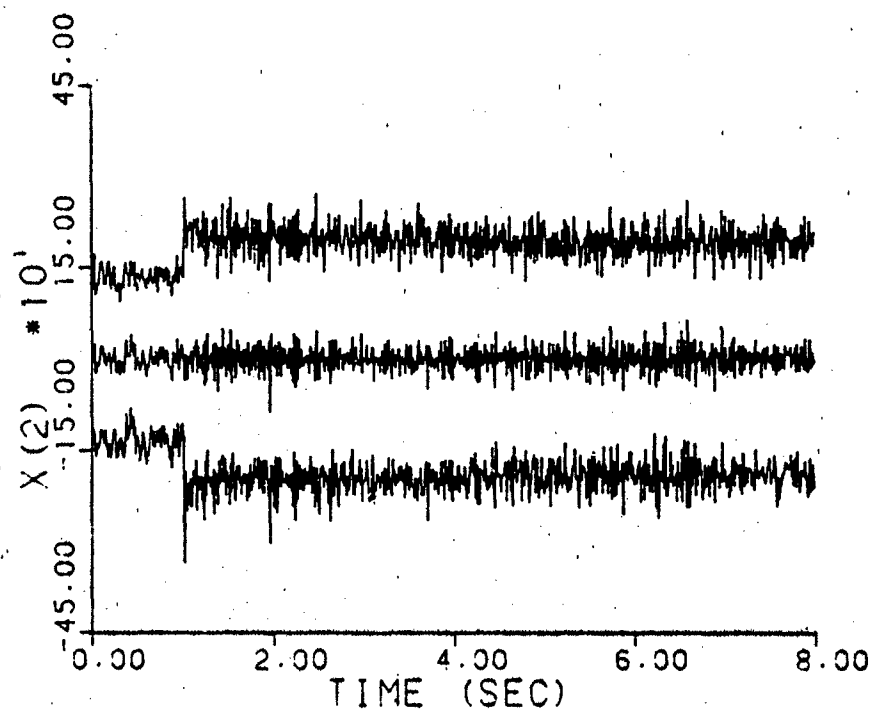
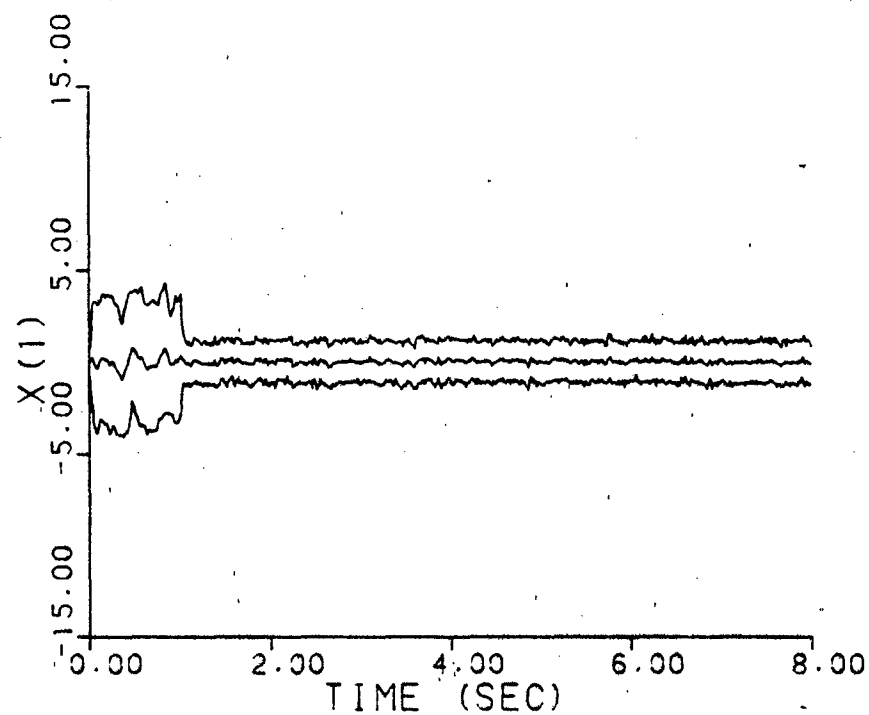


Figure A-20g. Sliding Bank Multiple Model Adaptive Controller
 $\underline{a} = (0.93, 41.0)$

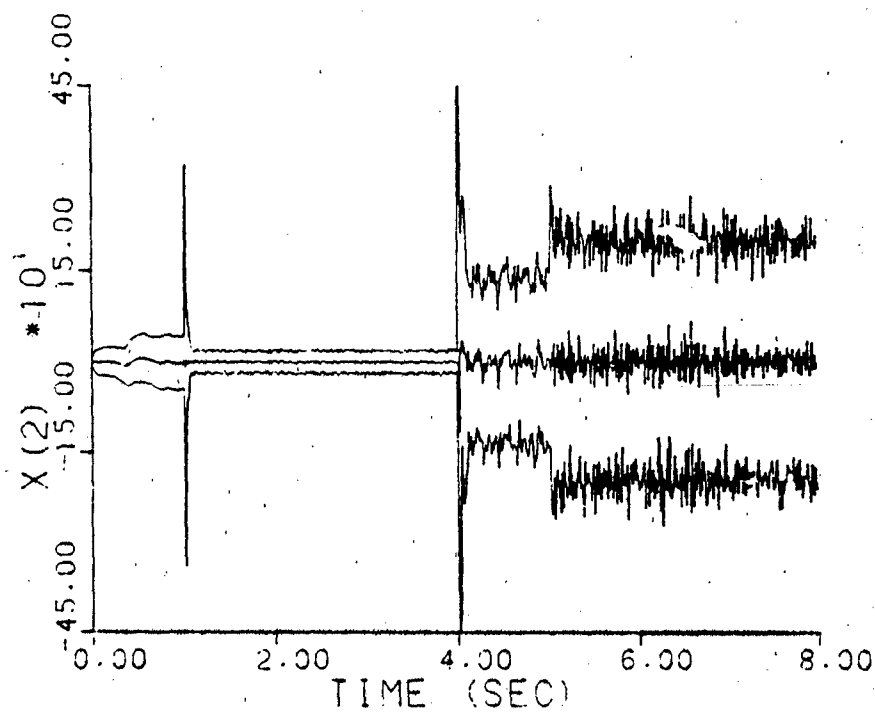
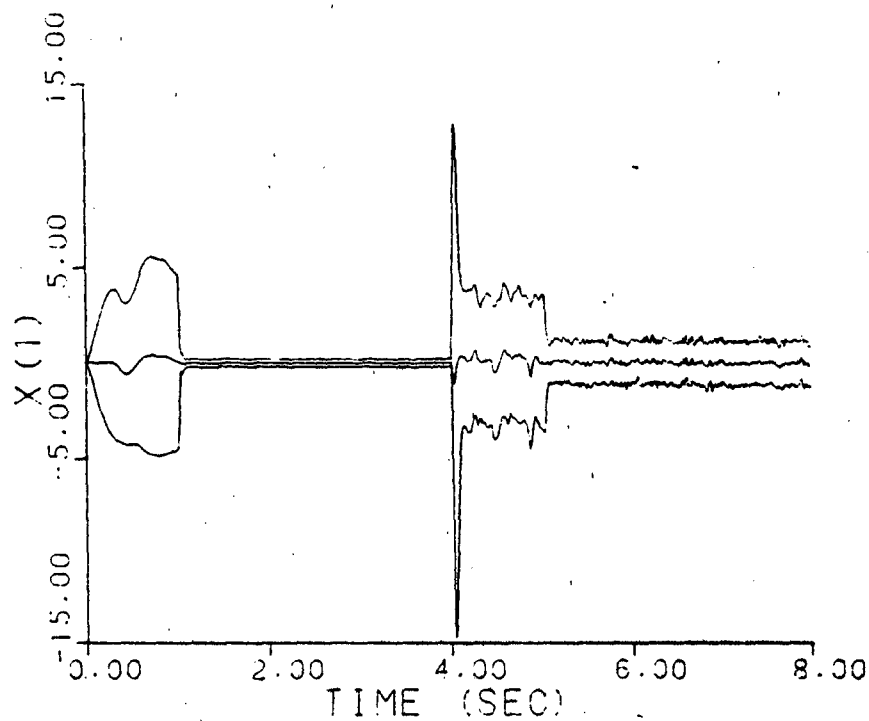


Figure A-20h. Sliding Bank Multiple Model Adaptive Controller
a jumps: (0.07, 9.0) - (0.93, 41.0)

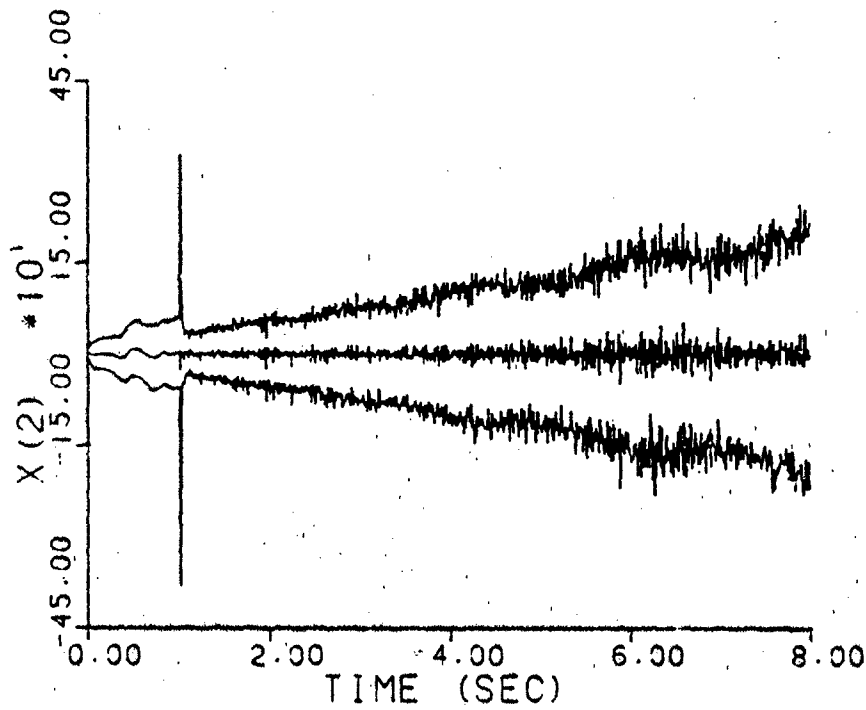
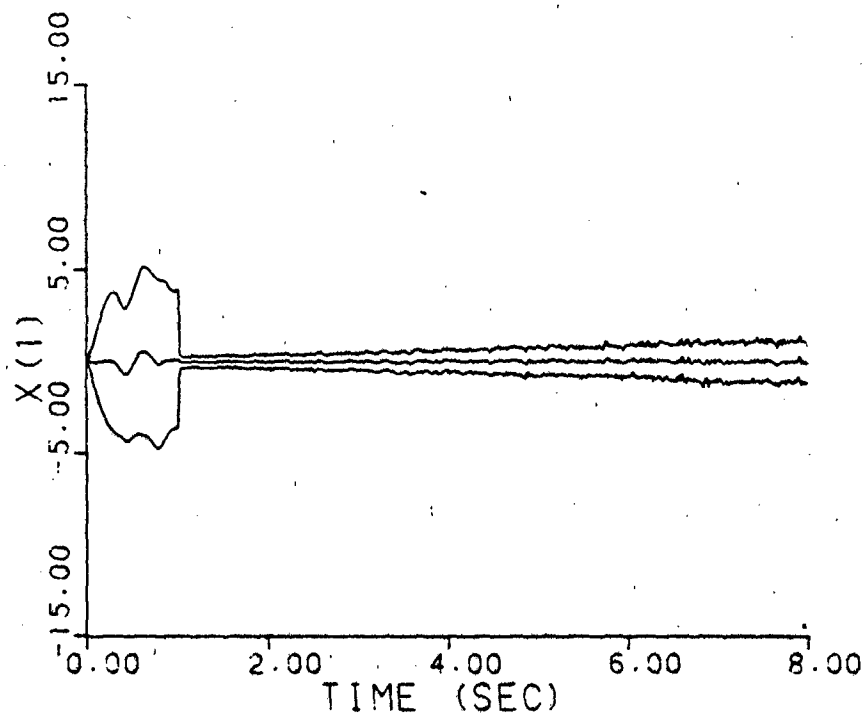


Figure A-20i. Sliding Bank Multiple Model Adaptive Controller
a varies: (0.07, 9.0) - (0.93, 41.0).

Appendix B

Parameter Estimation Error Plots

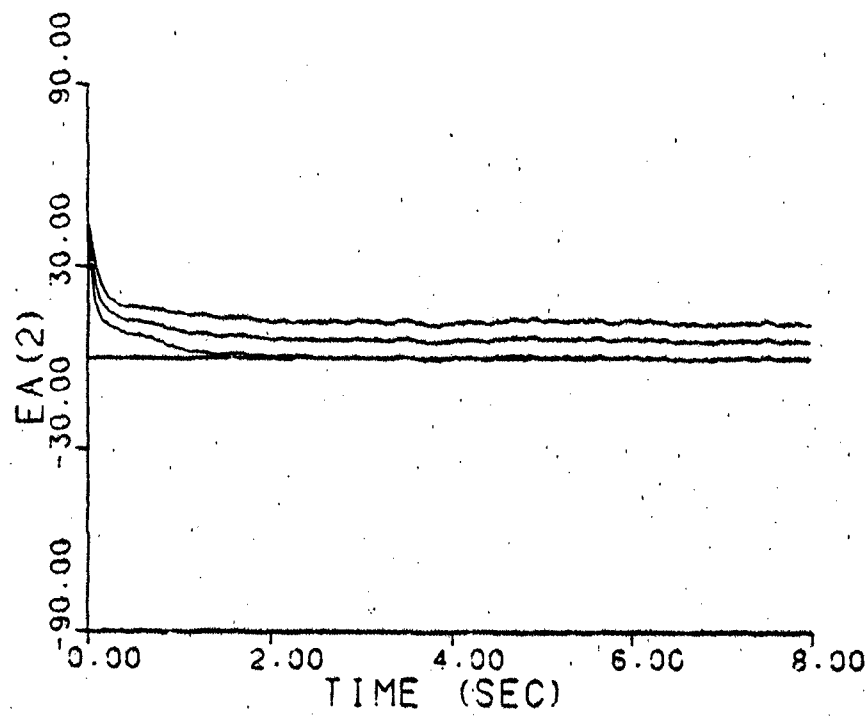
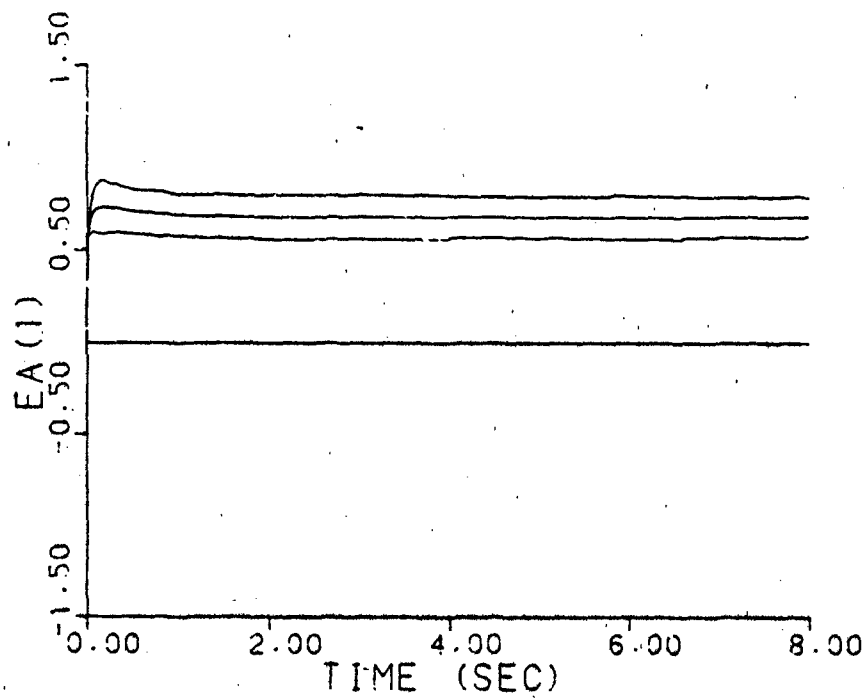


Figure B-1a. Residual Monitoring
 $\underline{a} = (10, 10)$, no dither

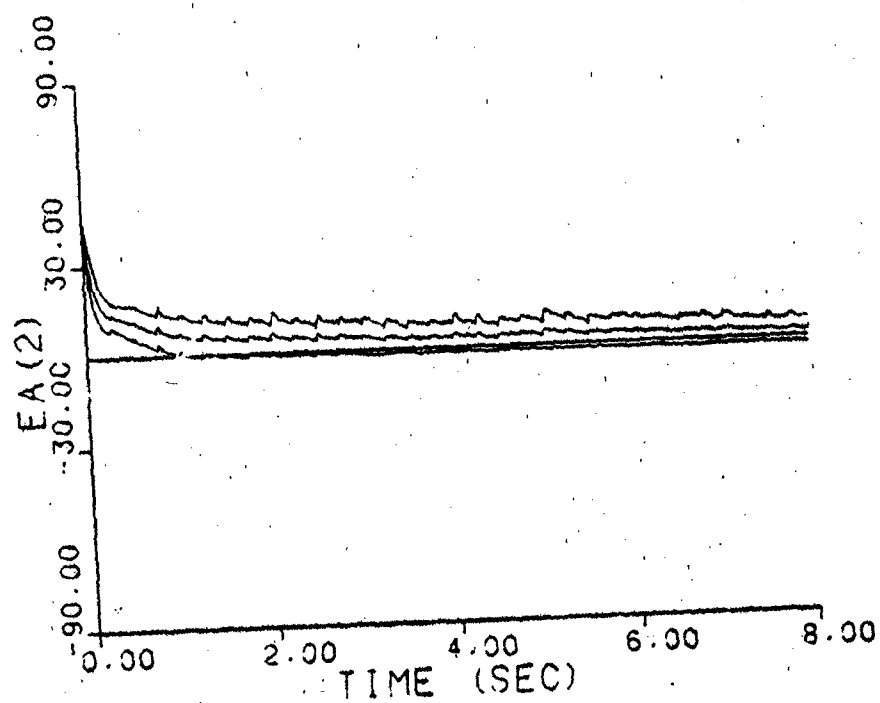
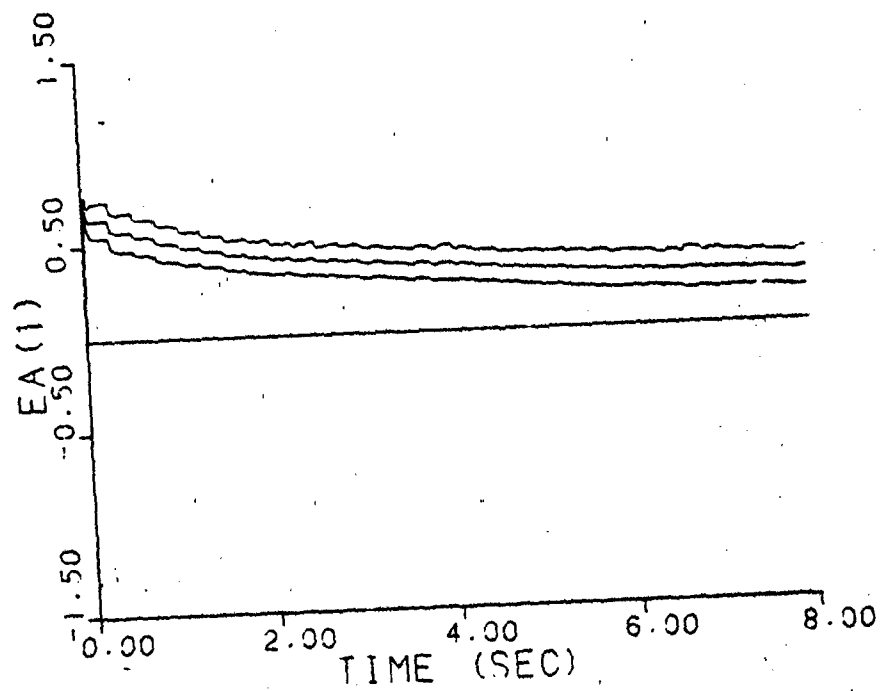


Figure B-1b. Residual Monitoring
 $\underline{a} = (10, 10)$, dither

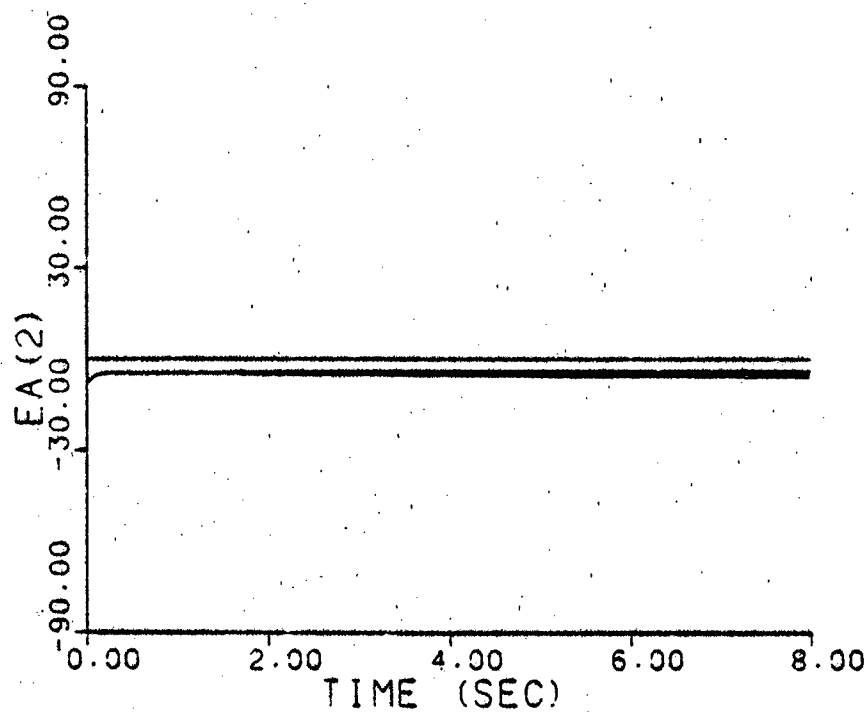
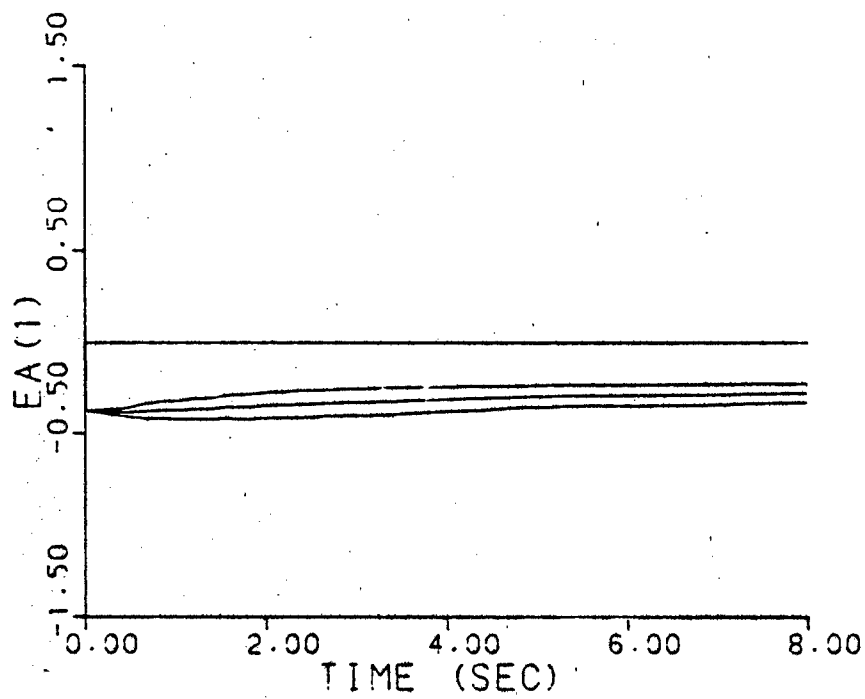


Figure B-1c. Residual Monitoring
 $\underline{a} = (0.07, 9.0)$, no dither

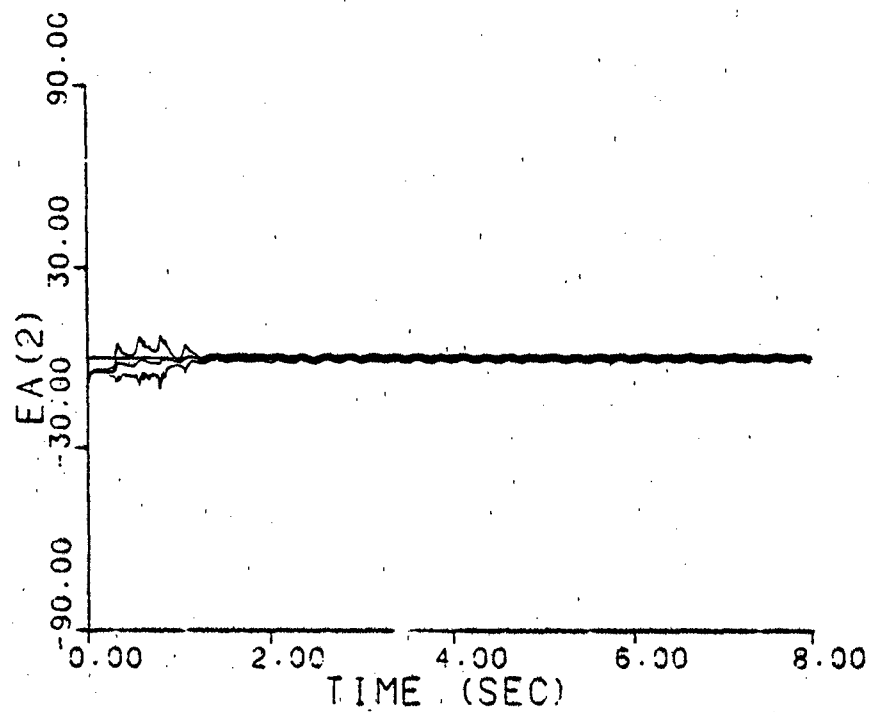
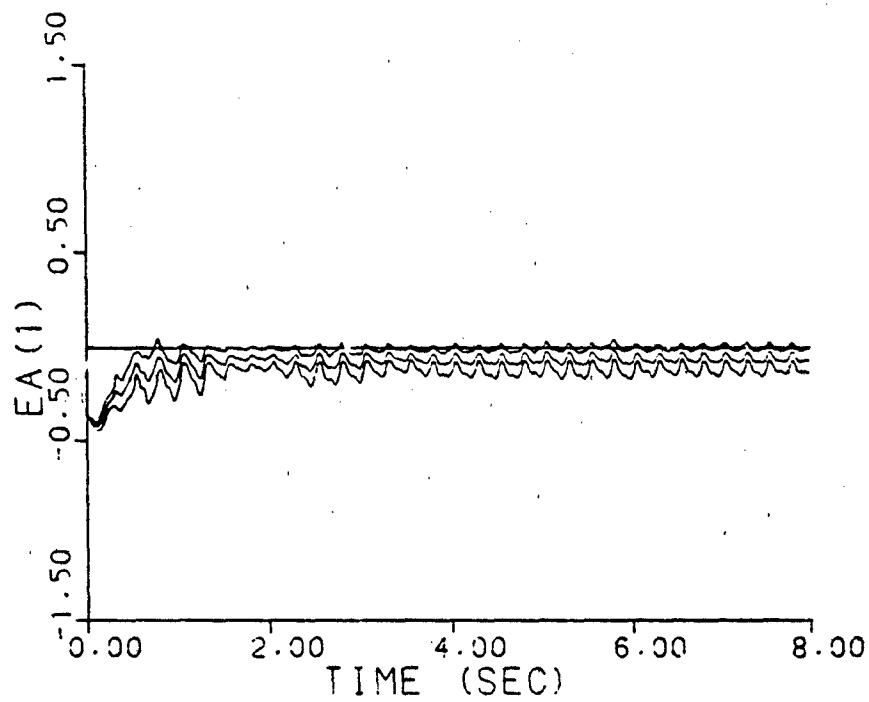


Figure B-1d. Residual Monitoring
 $\underline{a} = (0.07, 9.0)$, dither

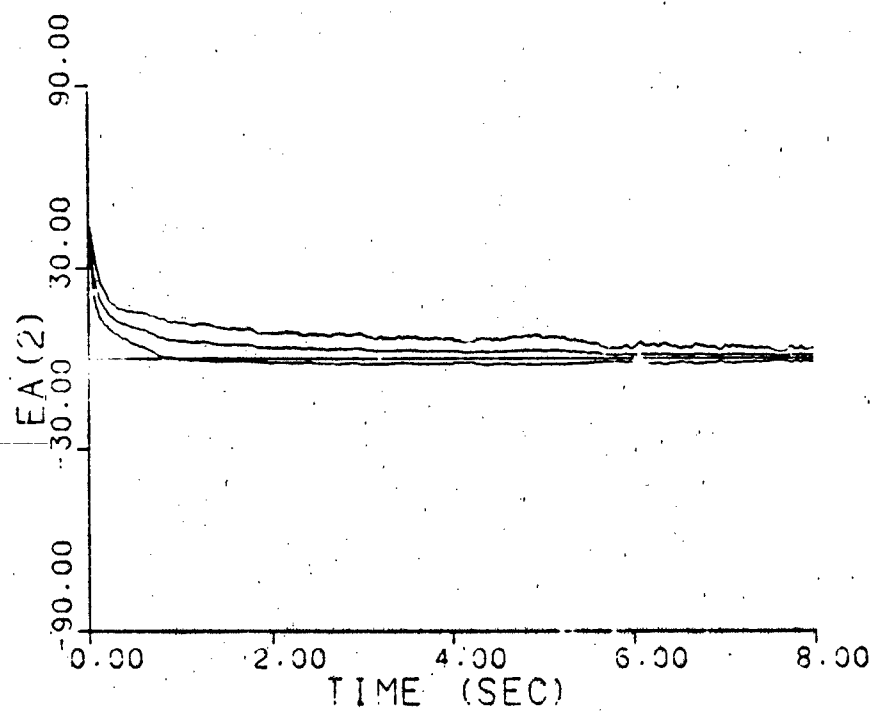
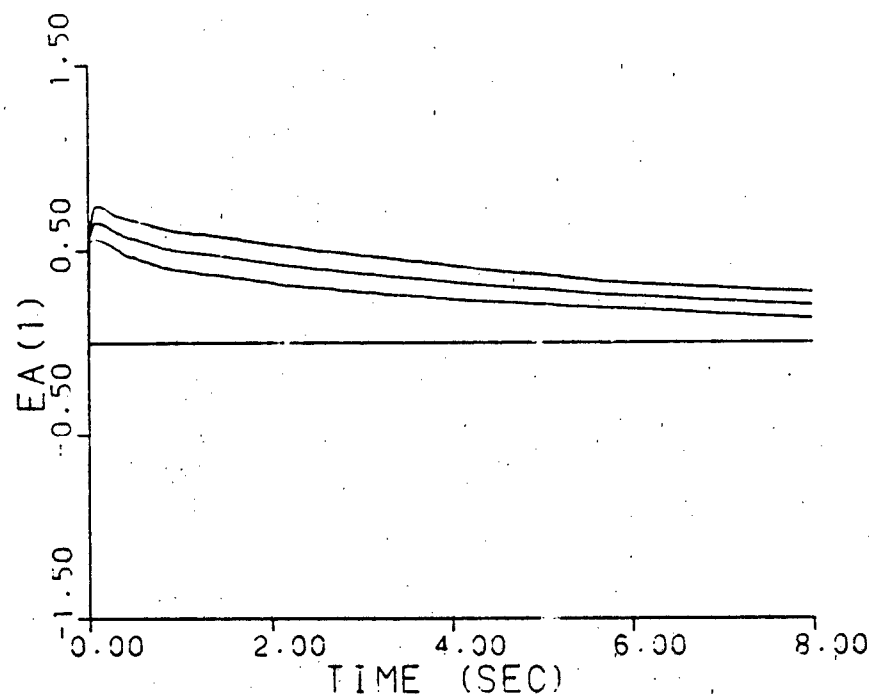


Figure B-2a. Parameter Position Estimate Monitoring
 $\underline{a} = (10, 10)$, no dither

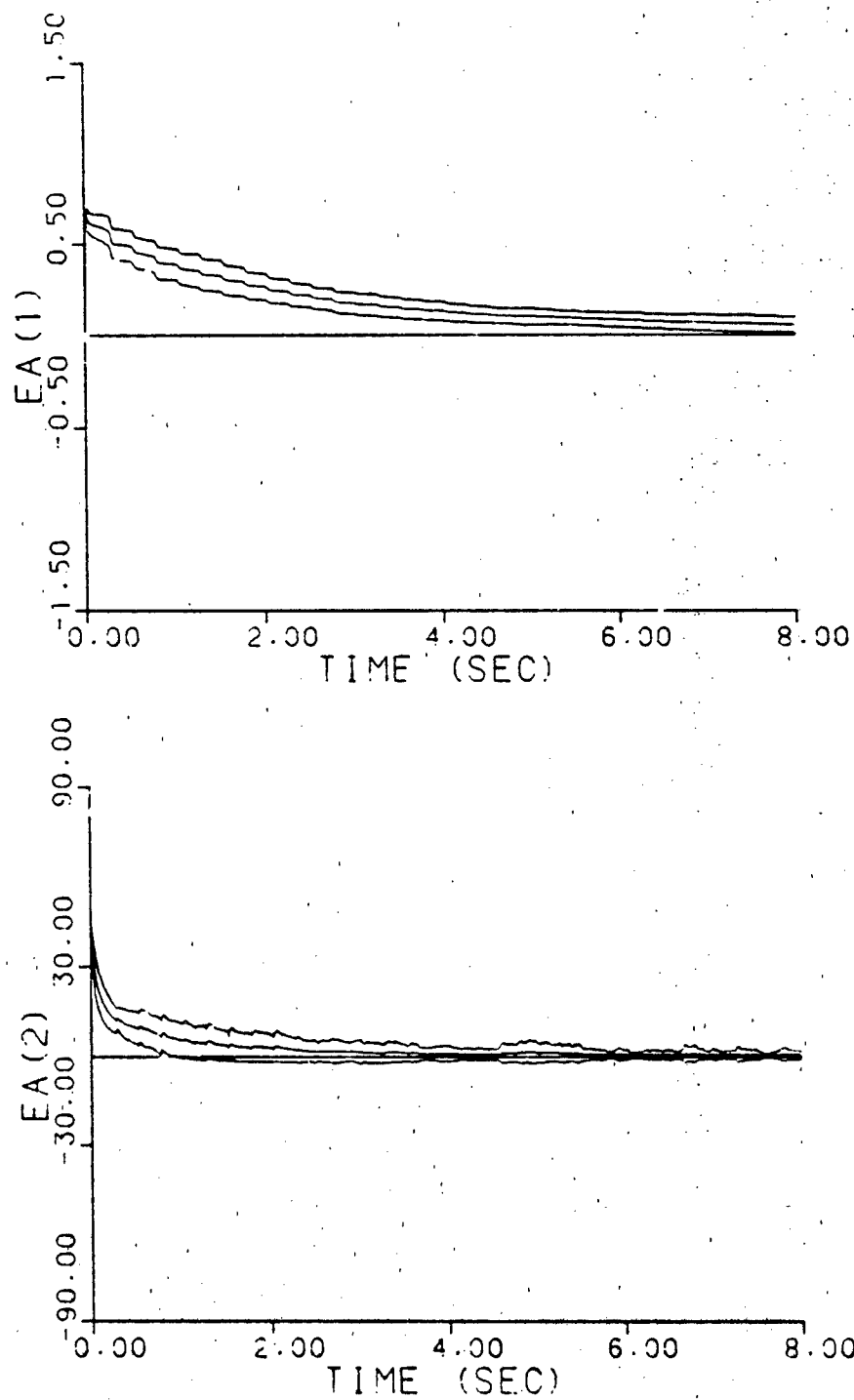


Figure B-2b. Parameter Position Estimate Monitoring
 $\underline{a} = (10, 10)$, dither

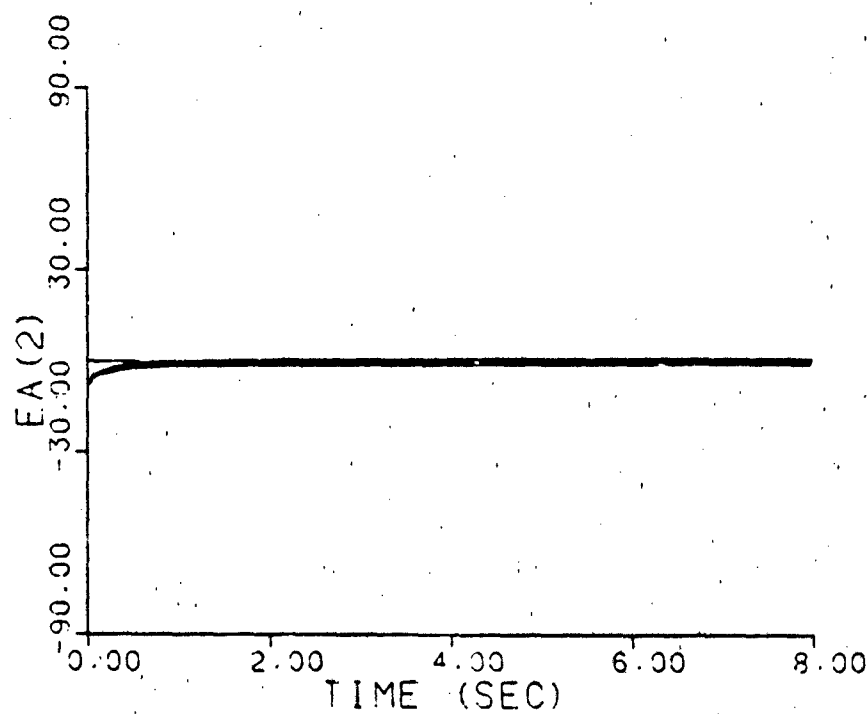
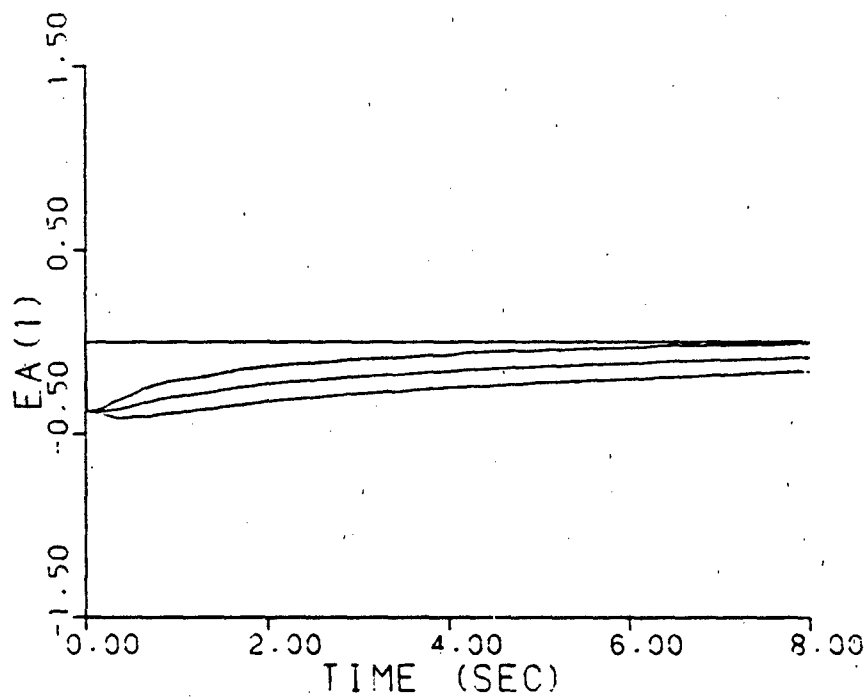


Figure B-2c. Parameter Position Estimate Monitoring
 $\underline{a} = (0.07, 9.0)$, no dither

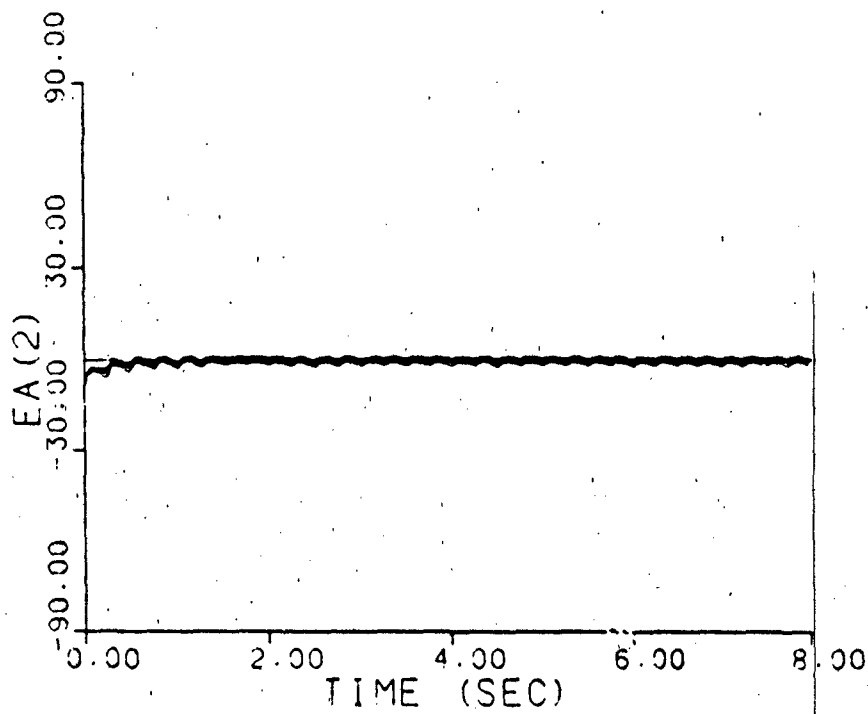
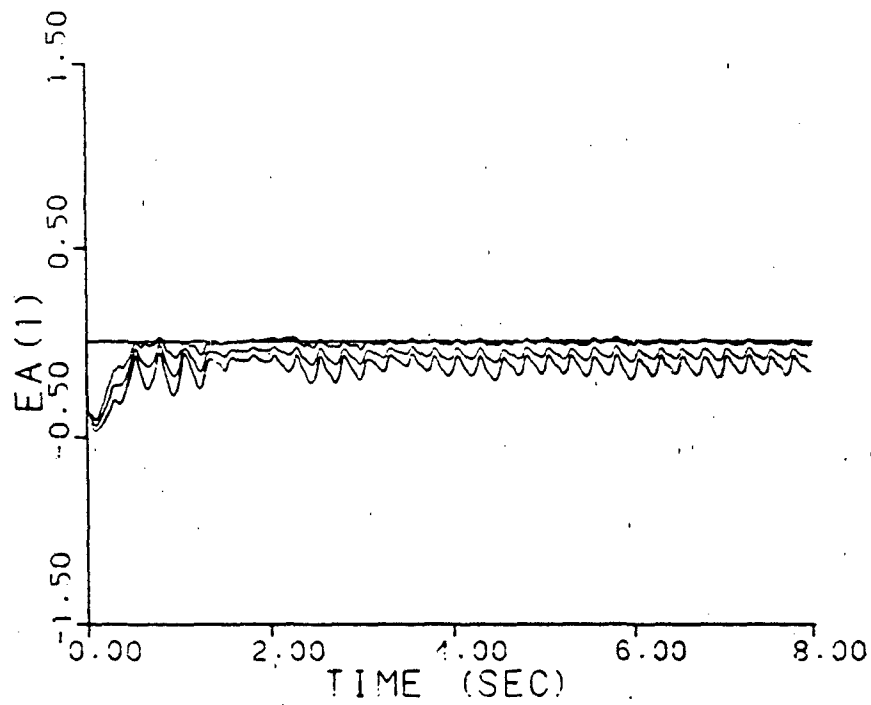


Figure B-2d. Parameter Position Estimate Monitoring
 $\underline{a} = (0.07, 9.0)$, dither

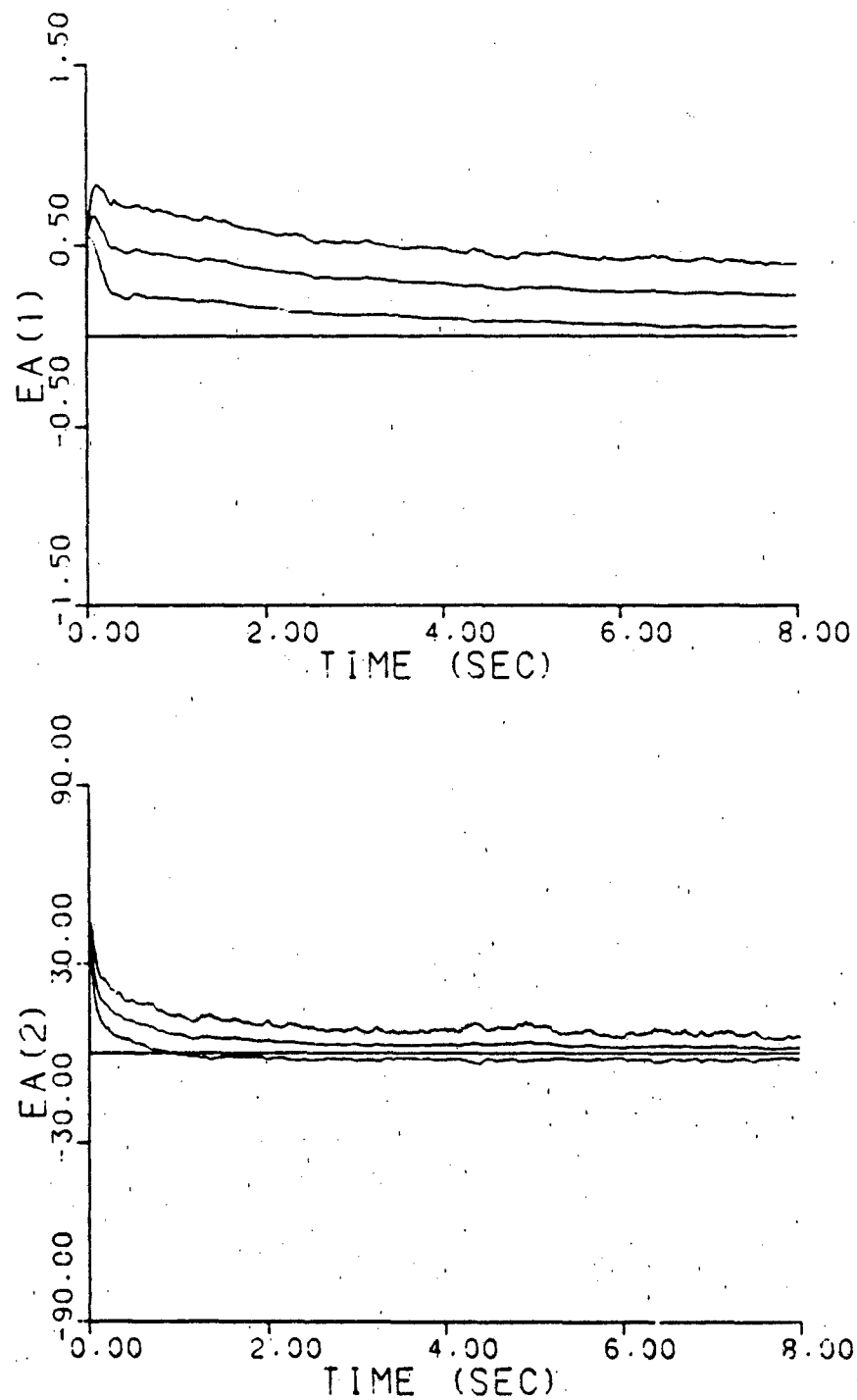


Figure B-3a. Parameter Position and Velocity
Estimate Monitoring
 $\underline{a} = (10, 10)$, no dither

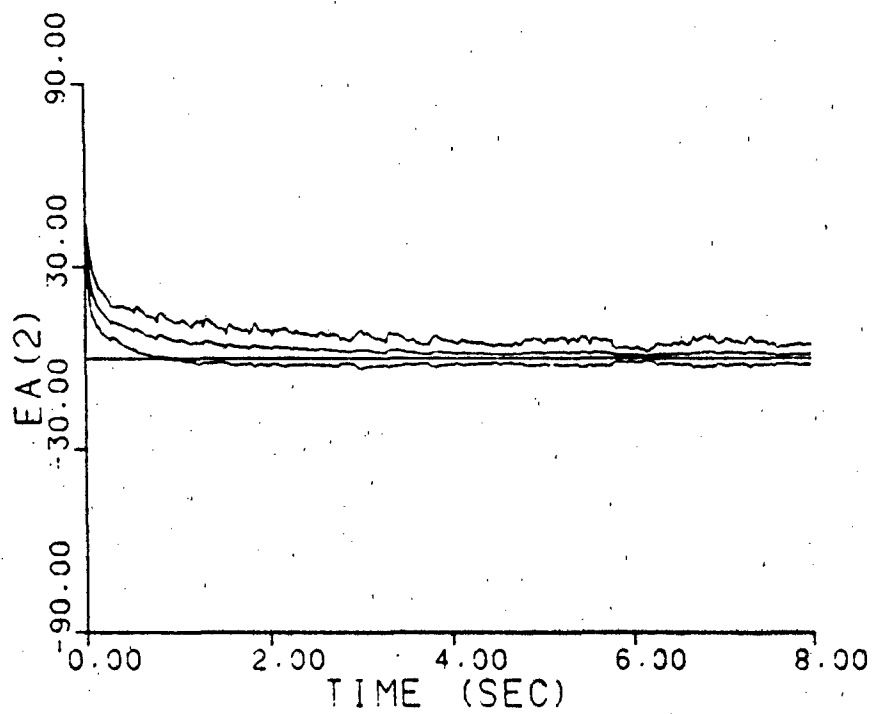
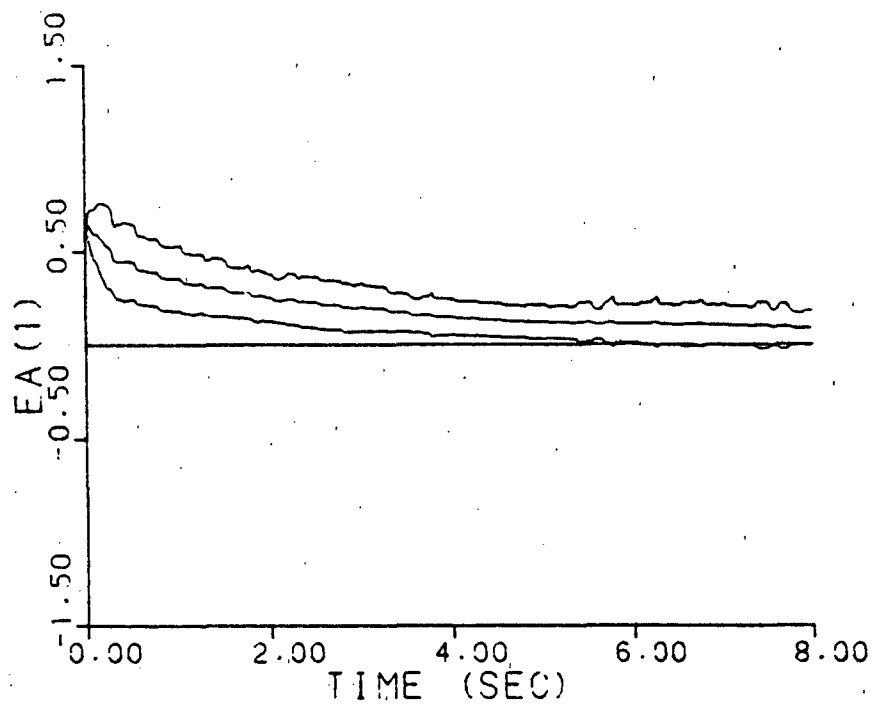


Figure B-3b. Parameter Position and Velocity
Estimate Monitoring
 $\underline{a} = (10, 10)$, dither

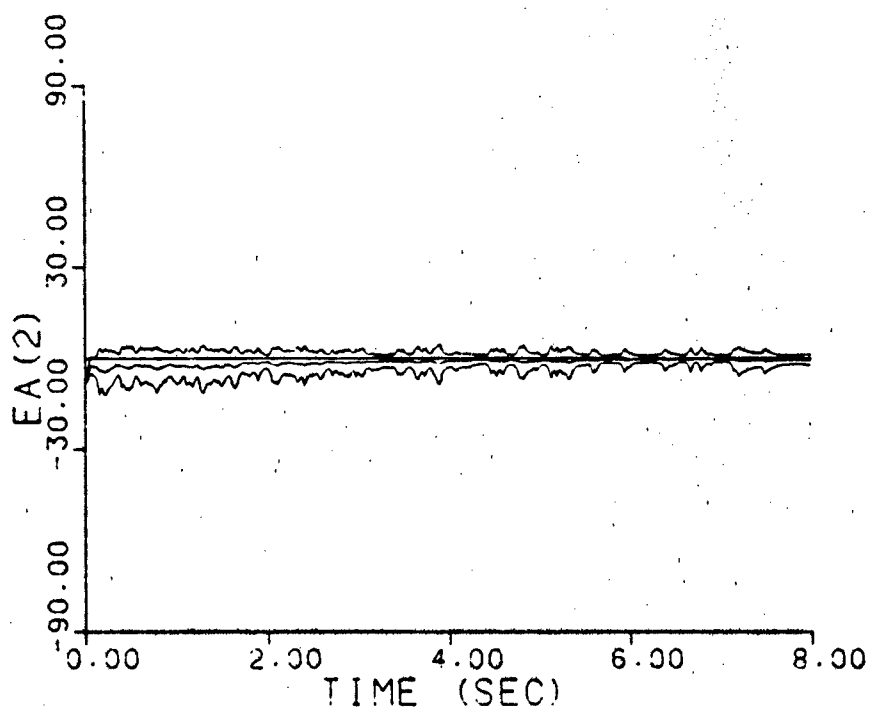
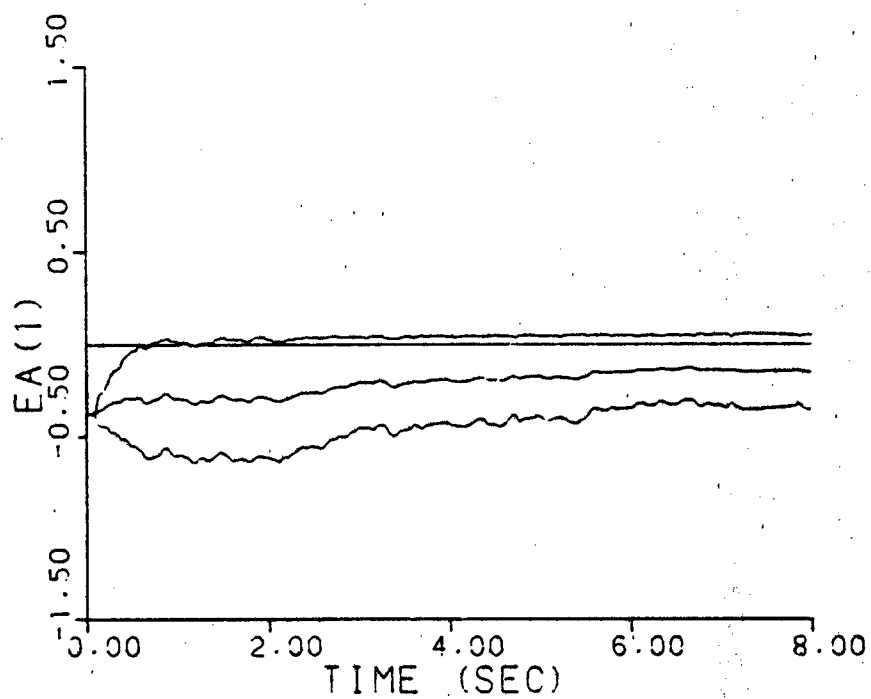


Figure B-3c. Parameter Position and Velocity
Estimate Monitoring
 $\underline{a} = (0.07, 9.0)$, no dither

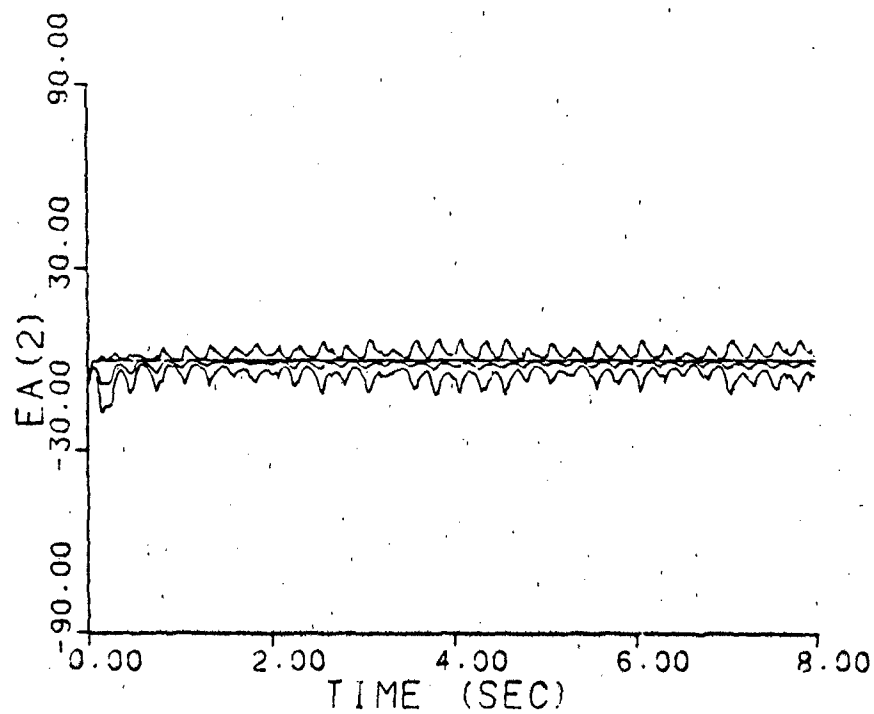
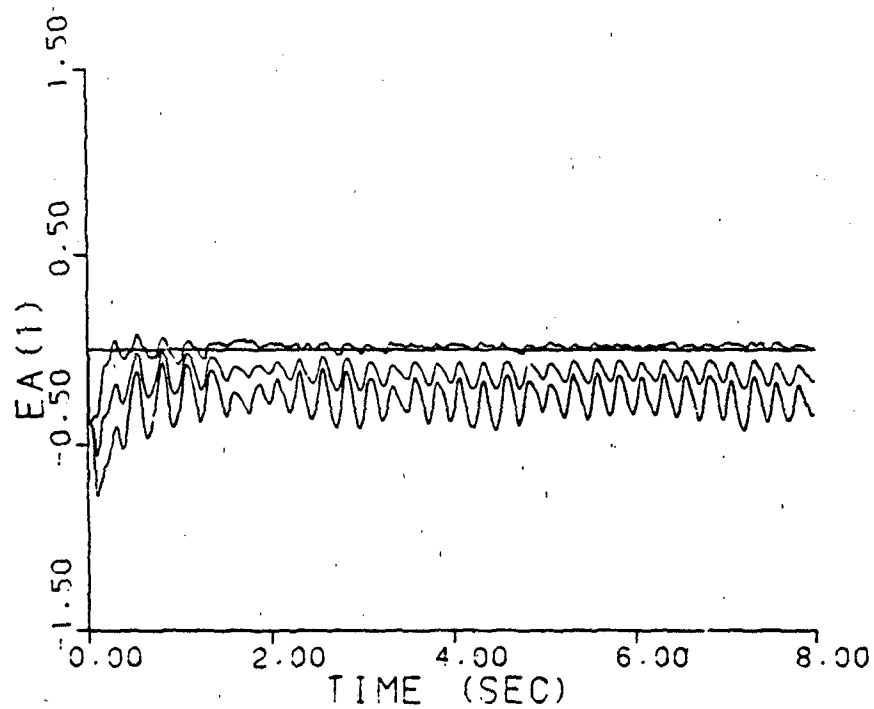


Figure B-3d. Parameter Position and Velocity
Estimate Monitoring
 $\underline{a} = (0.07, 9.0)$, dither

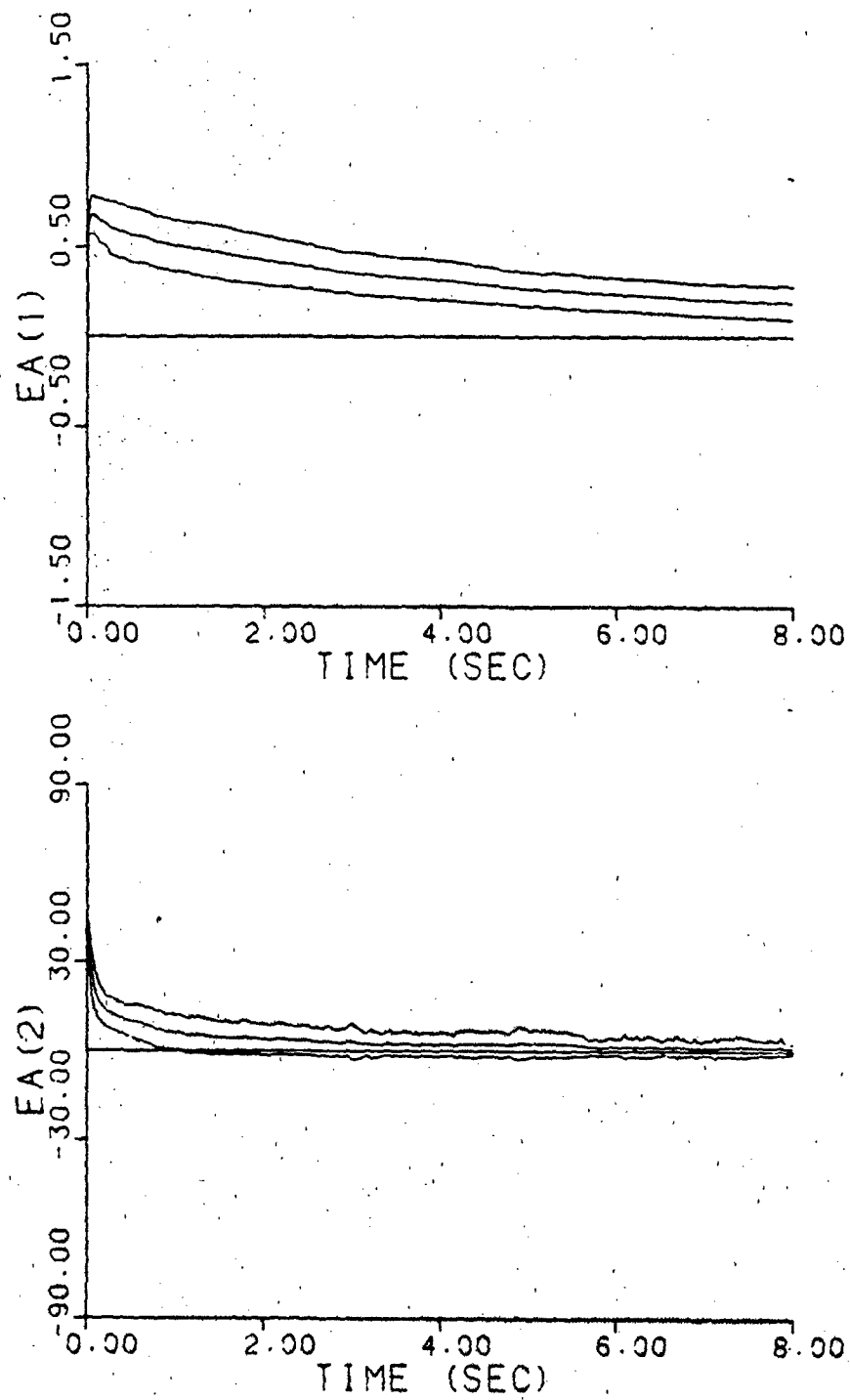


Figure B-4a. Probability Monitoring
 $\underline{a} = (10, 10)$, no dither

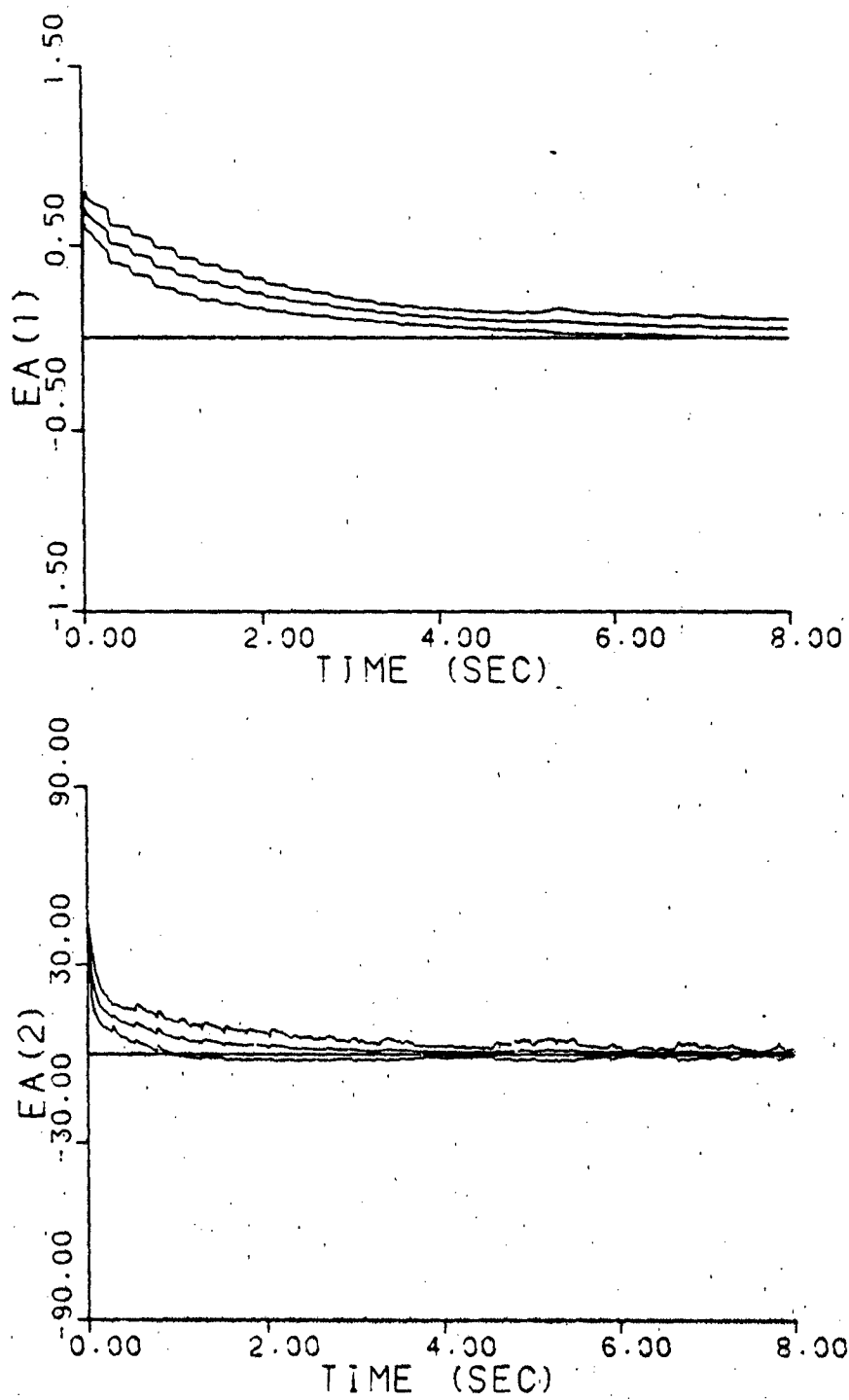


Figure B-4b. Probability Monitoring
 $\underline{a} = (10, 10)$, dither

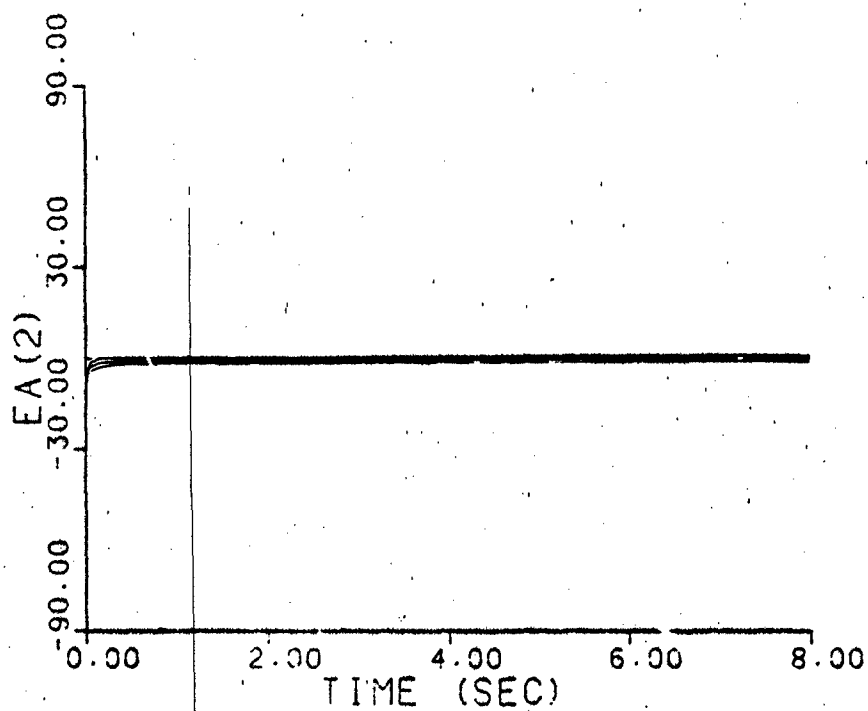
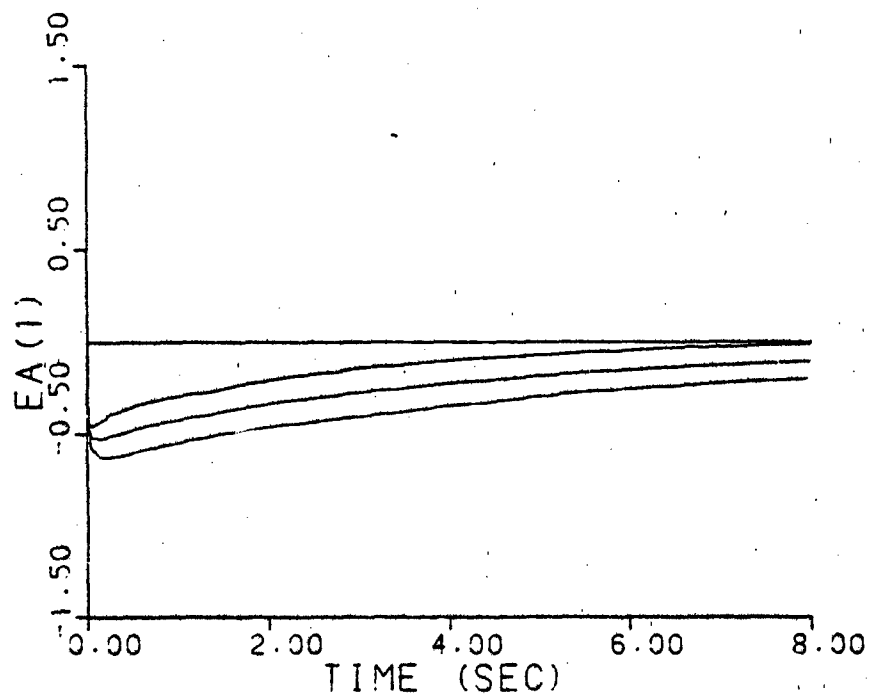


Figure B-4c. Probability Monitoring
 $\underline{a} = (0.07, 9.0)$, no dither

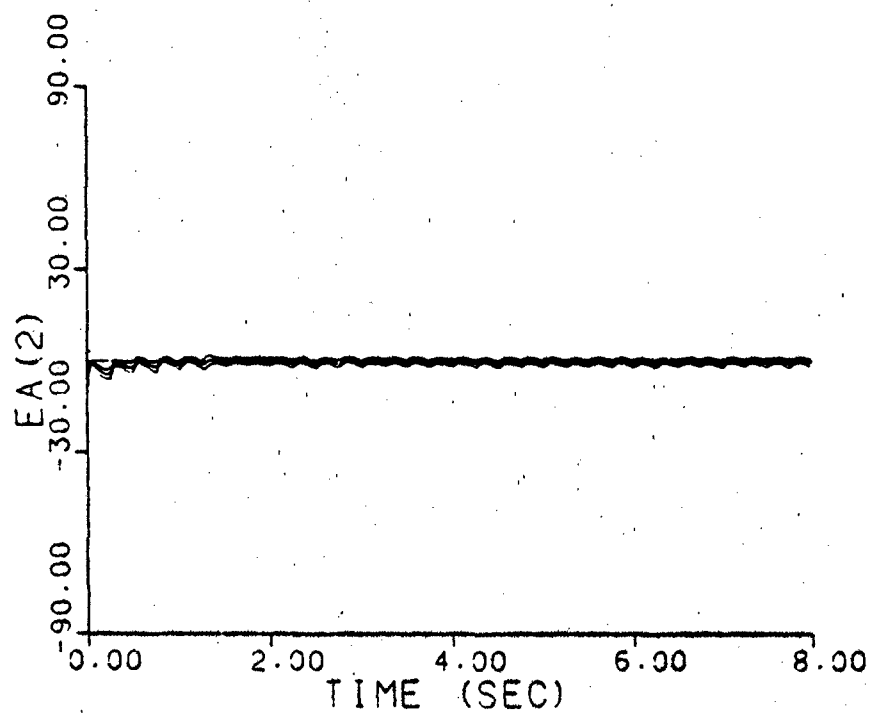
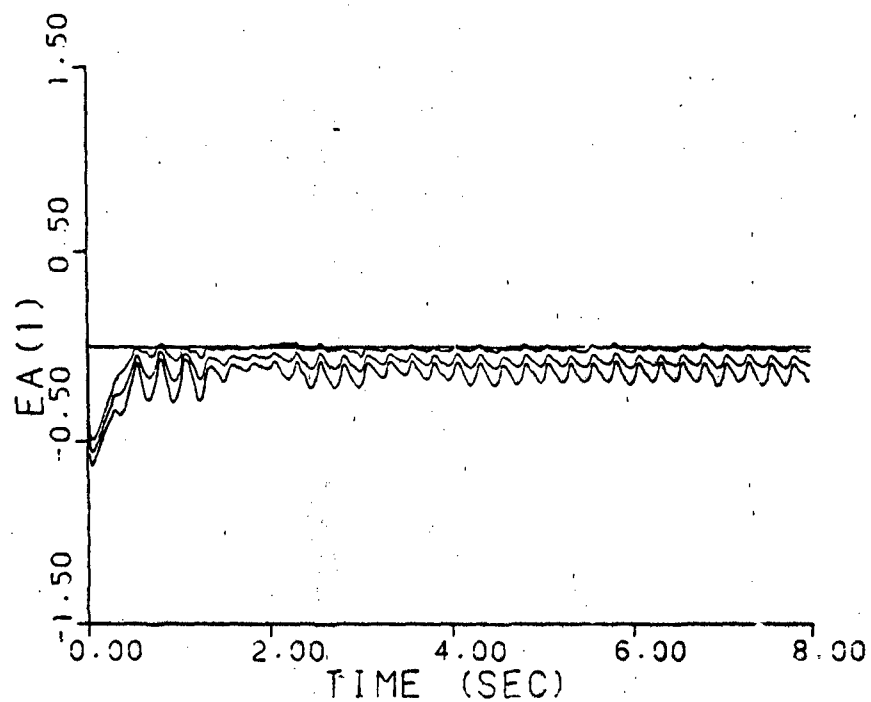


Figure E-4d. Probability Monitoring
 $\underline{a} = (0.07, 9.0)$, dither

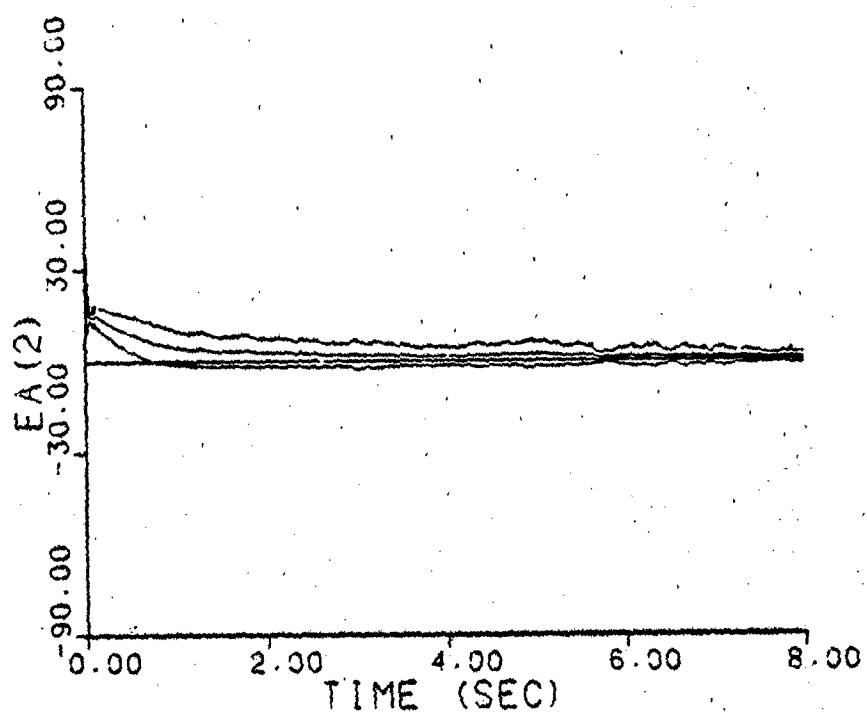
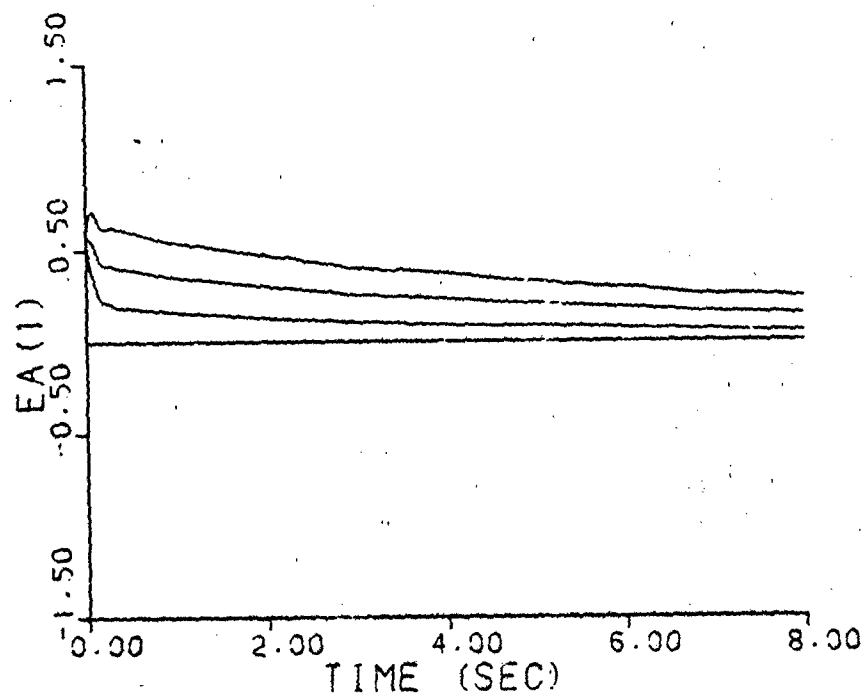


Figure B-5a. Parameter Covariance Monitoring
 $\underline{a} = (10, 10)$, no dither

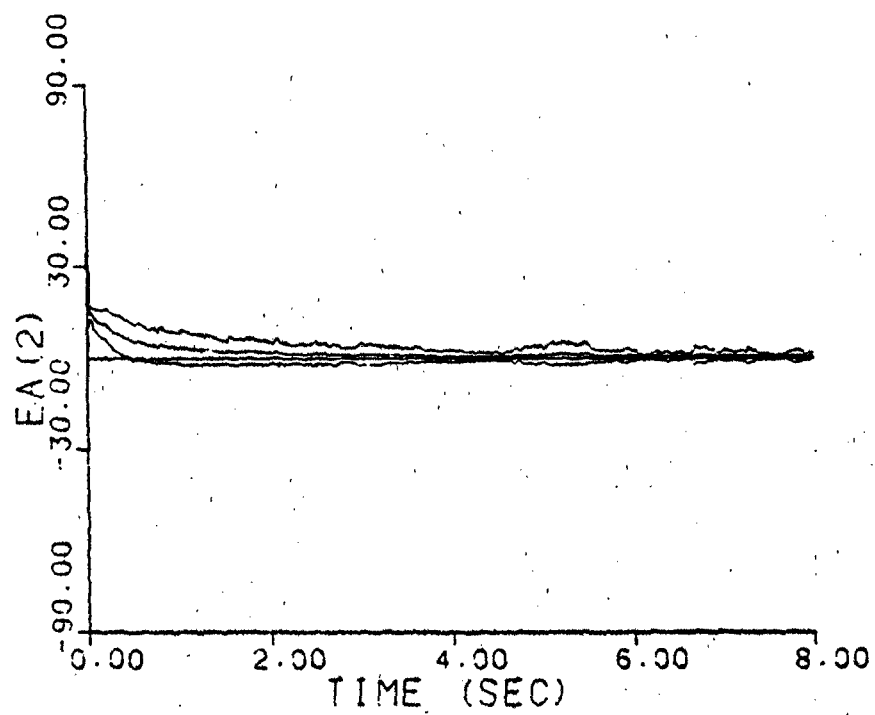
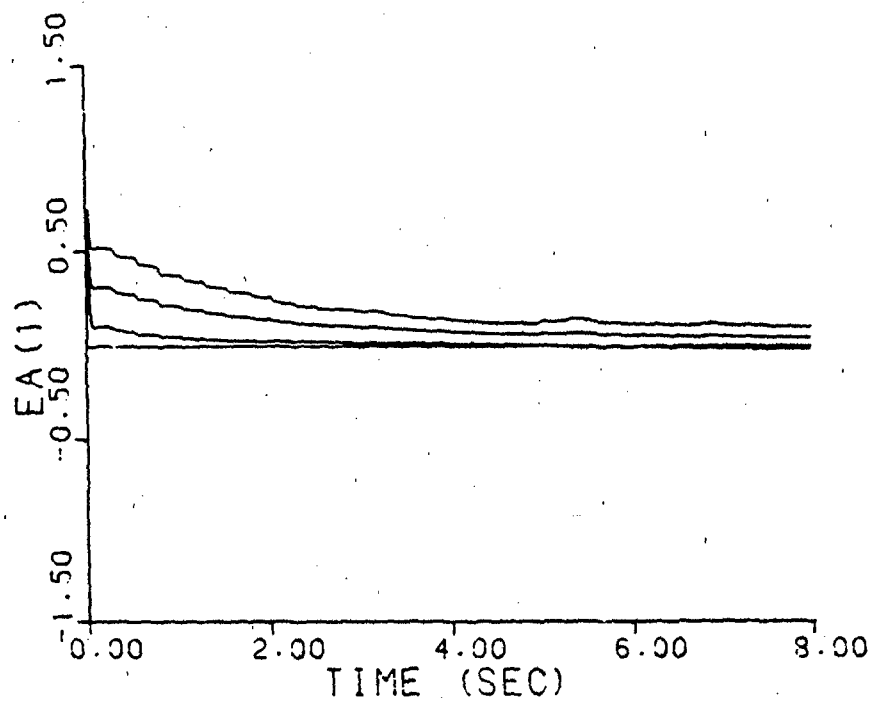


Figure B-5b. Parameter Covariance Monitoring
 $\underline{a} = (10, 10)$, dither

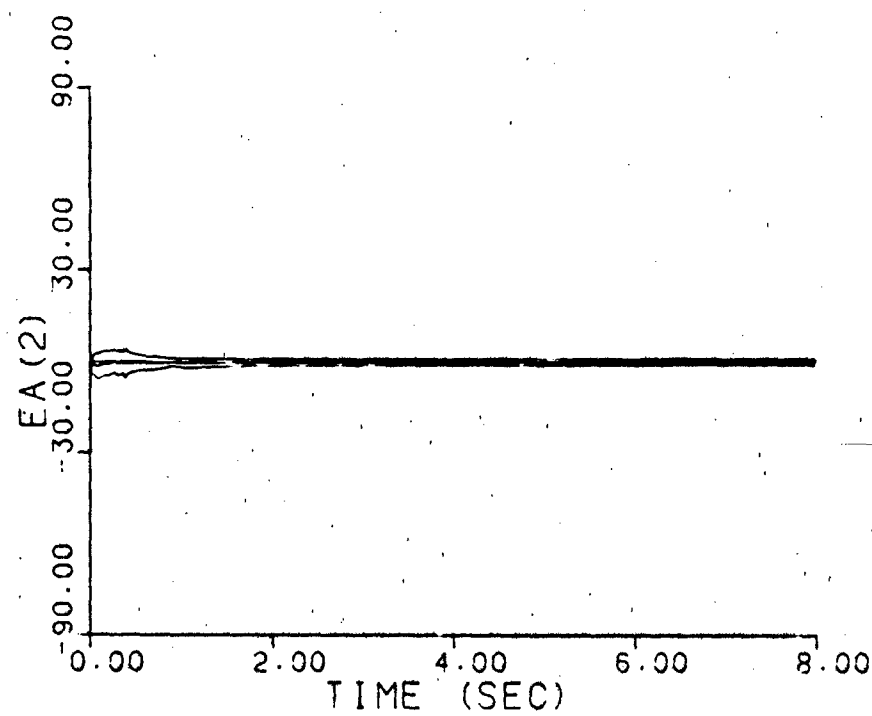
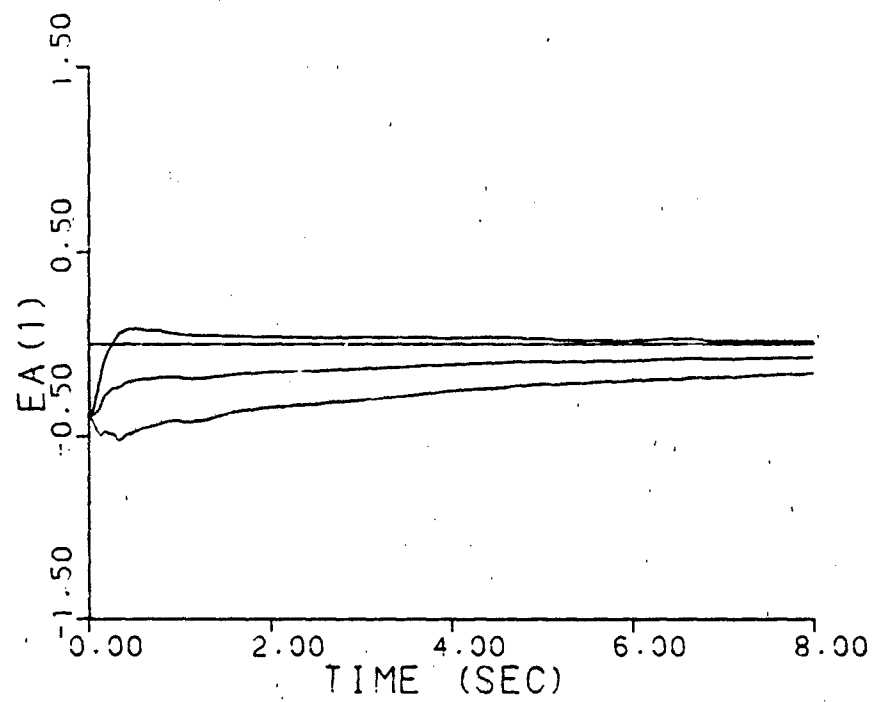


Figure B-5c. Parameter Covariance Monitoring
 $\underline{a} = (0.07, 9.0)$, no dither

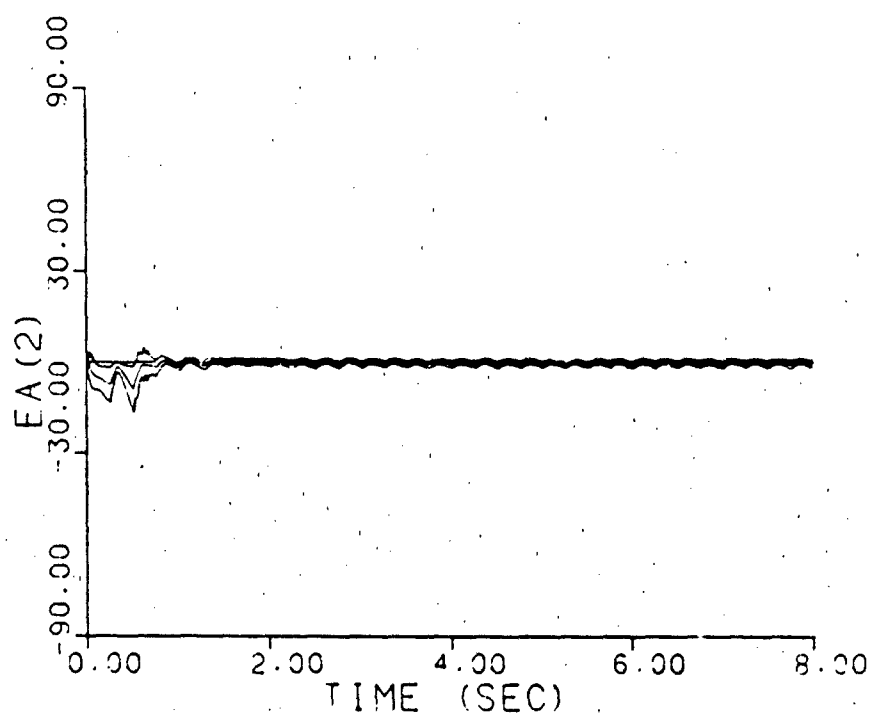
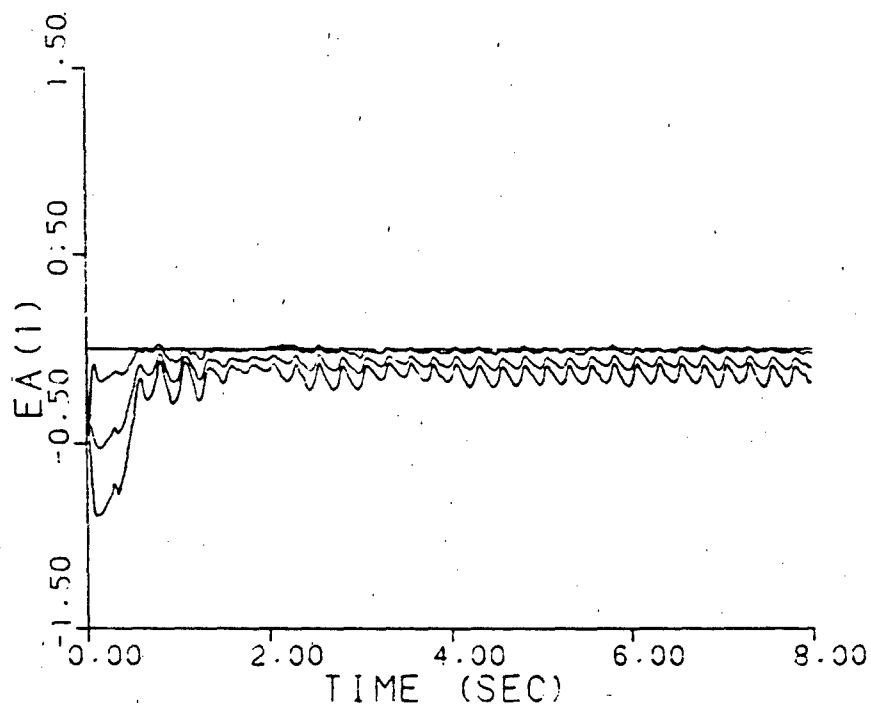


Figure B-5d. Parameter Covariance Monitoring
 $\underline{a} = (0.07, 9.0)$, dither

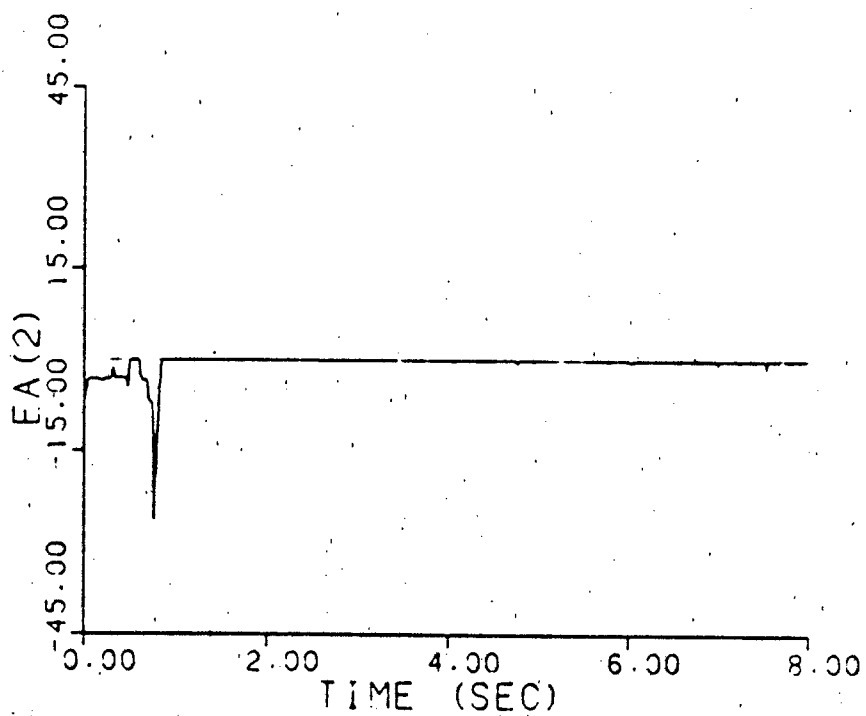
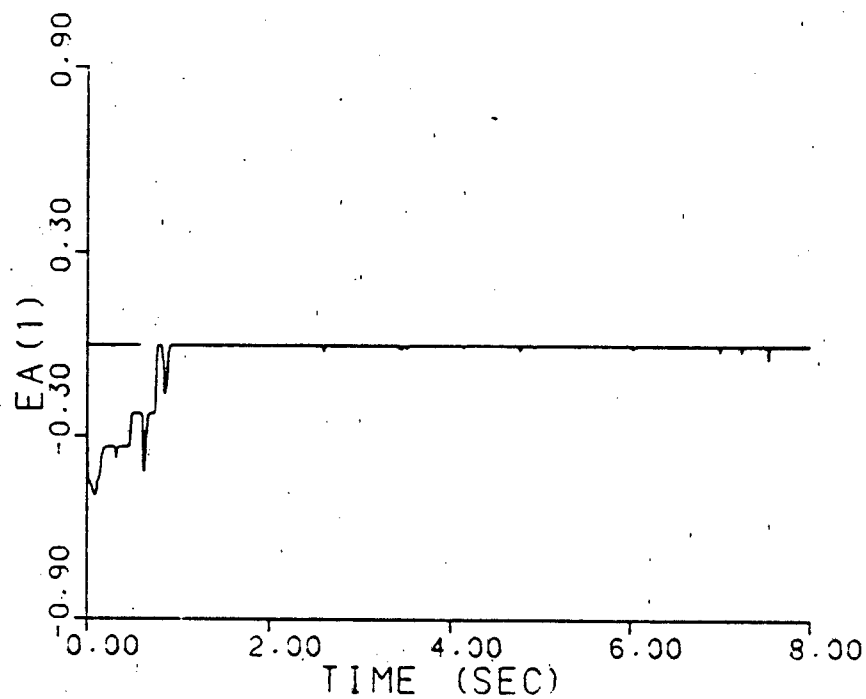


Figure B-6a. Residual Monitoring
Threshold: 14.0

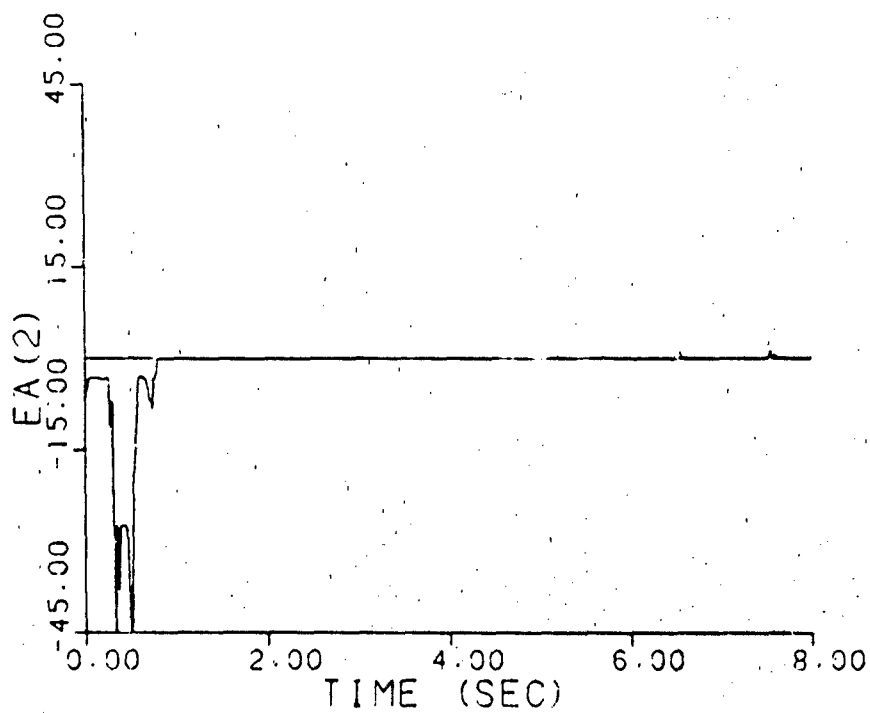
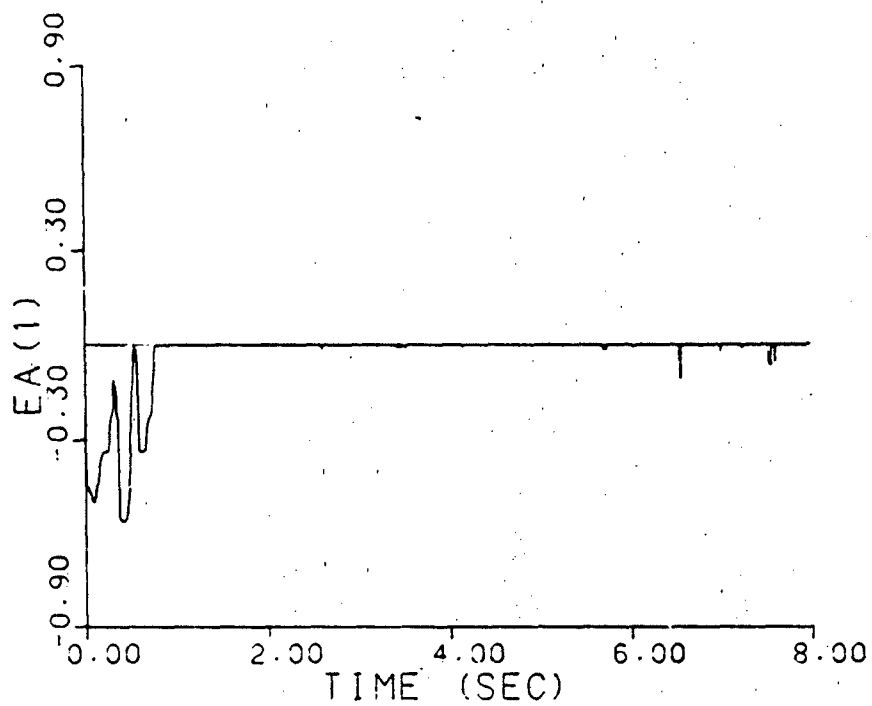


Figure B-6b. Residual Monitoring
Threshold: 7.0

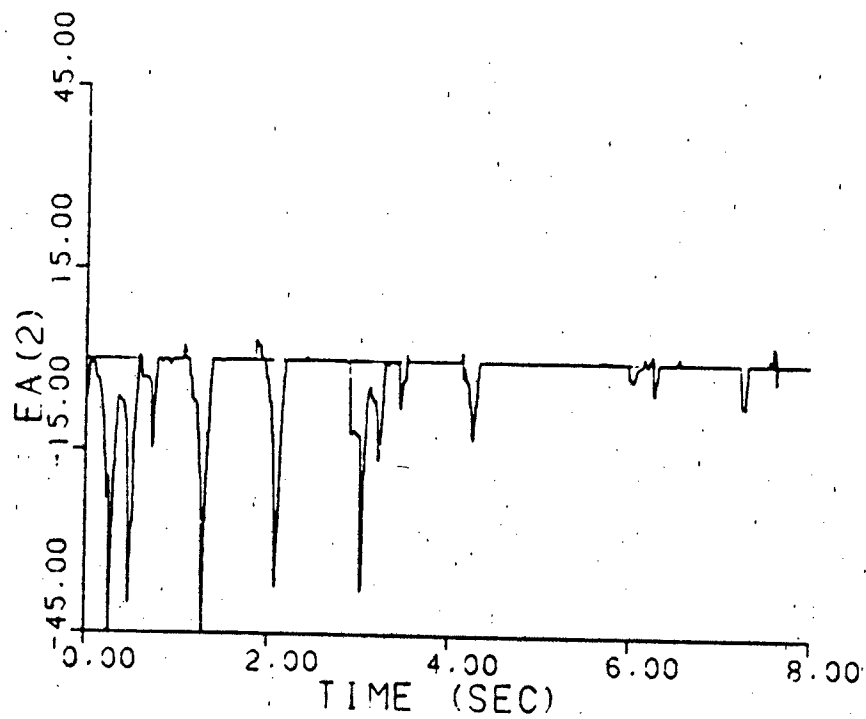
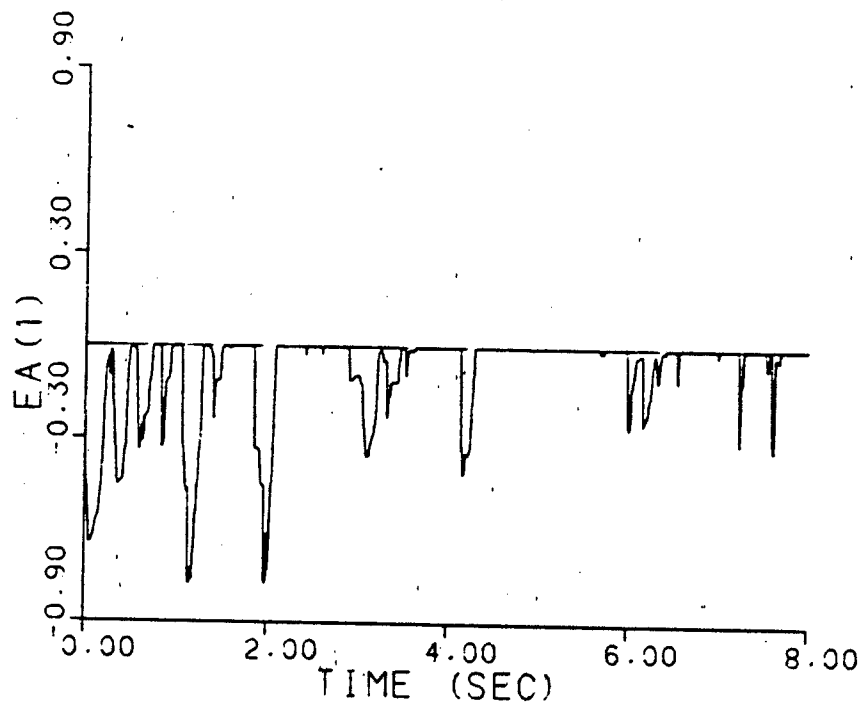


Figure B-6c. Residual Monitoring
Threshold: 1.0

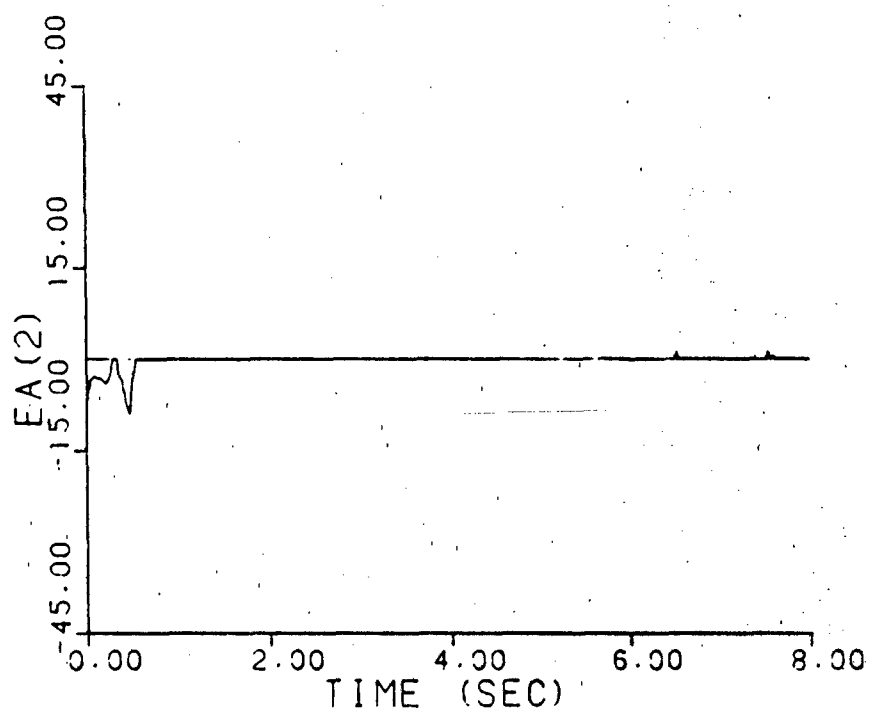
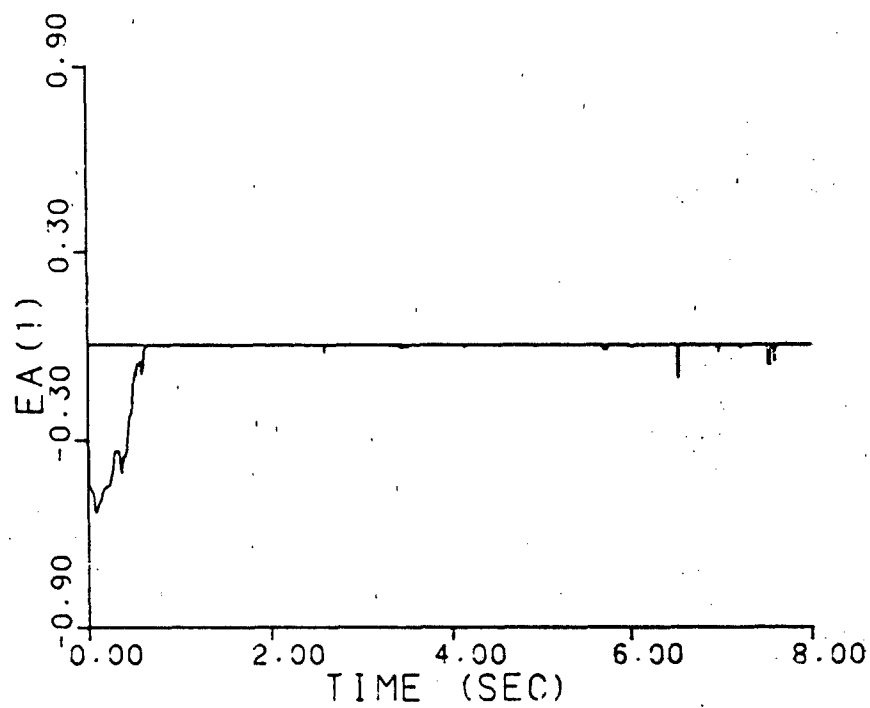


Figure B-7a. Parameter Position Estimate Monitoring
Threshold: 0.111

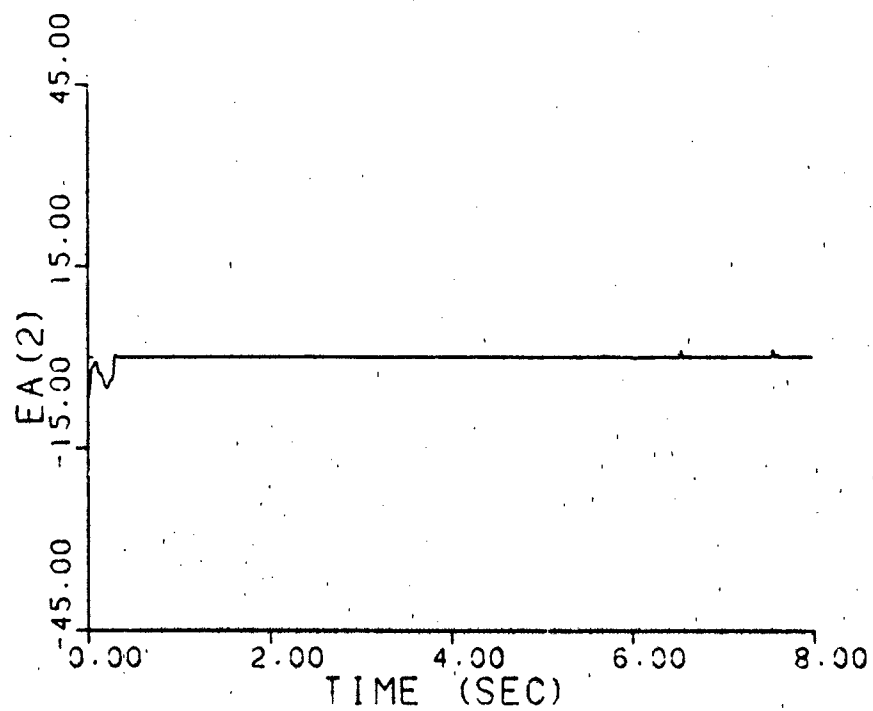
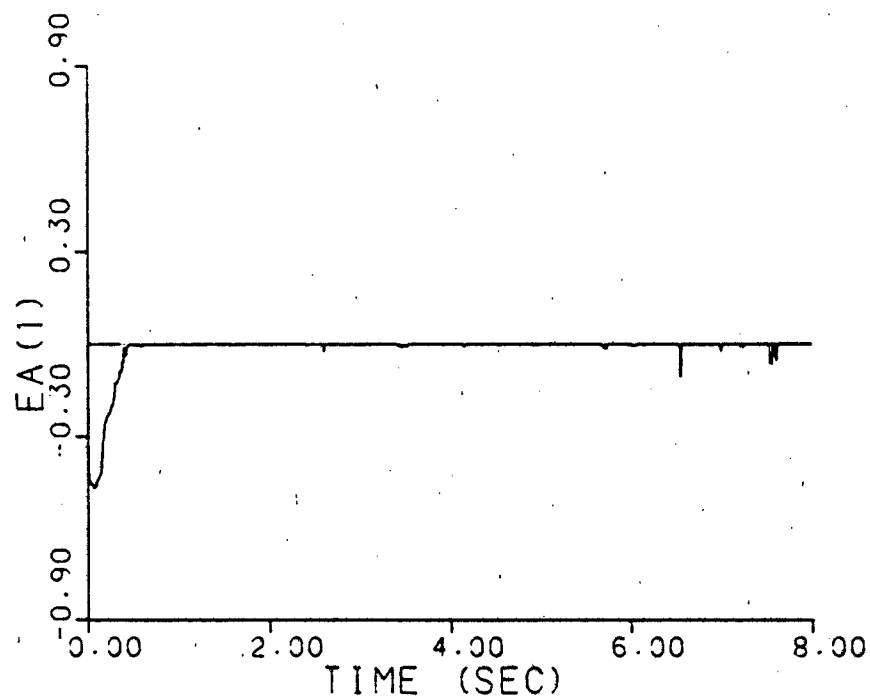


Figure B-7b. Parameter Position Estimate Monitoring
Threshold: 0.04

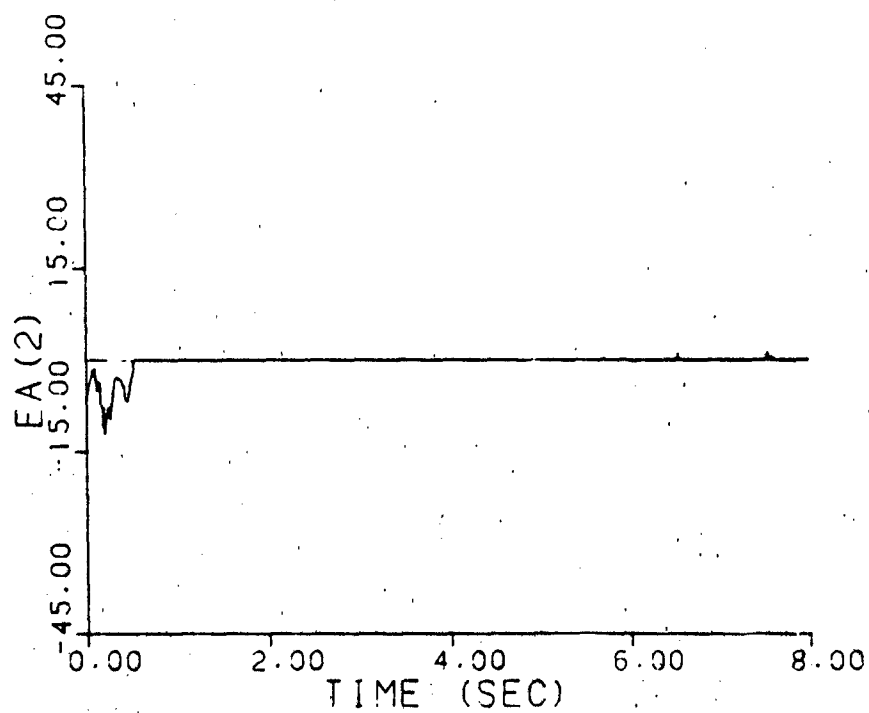
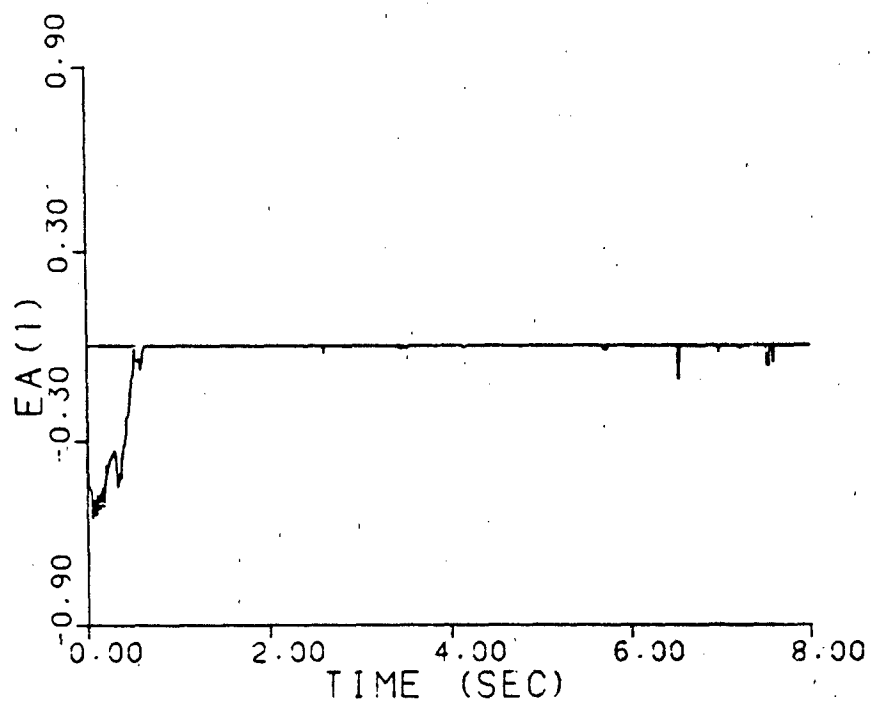


Figure B-7c. Parameter Position Estimate Monitoring
Threshold: 0.008

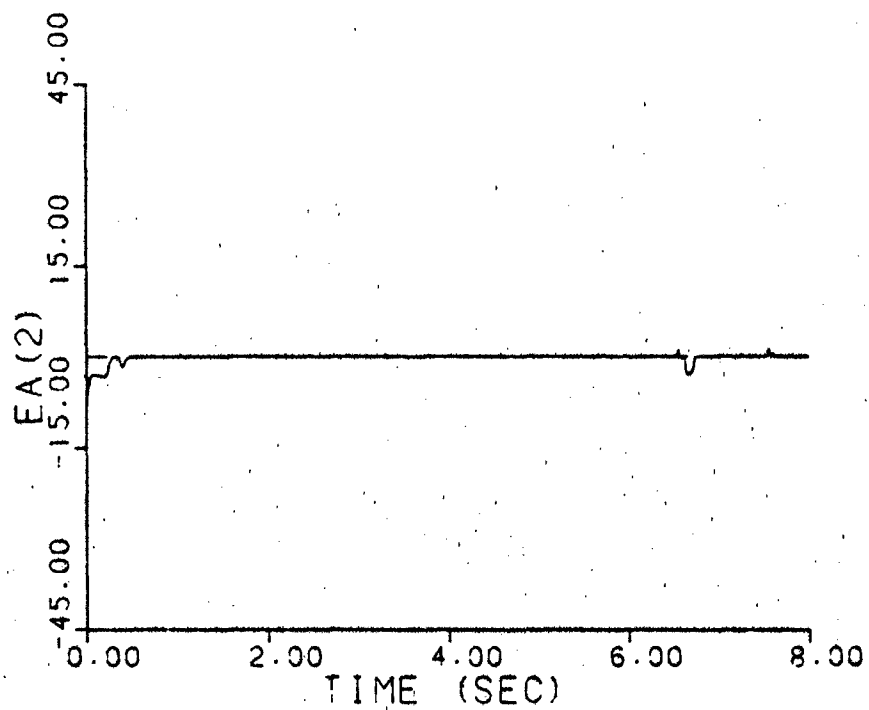
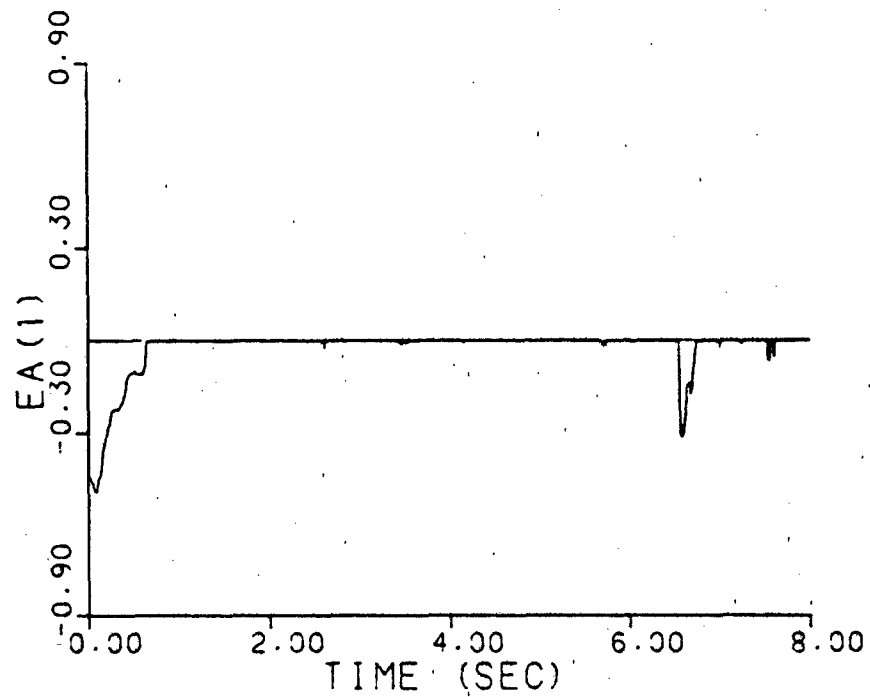


Figure B-8a. Parameter Position and Velocity
Estimate Monitoring
Threshold: 0.111

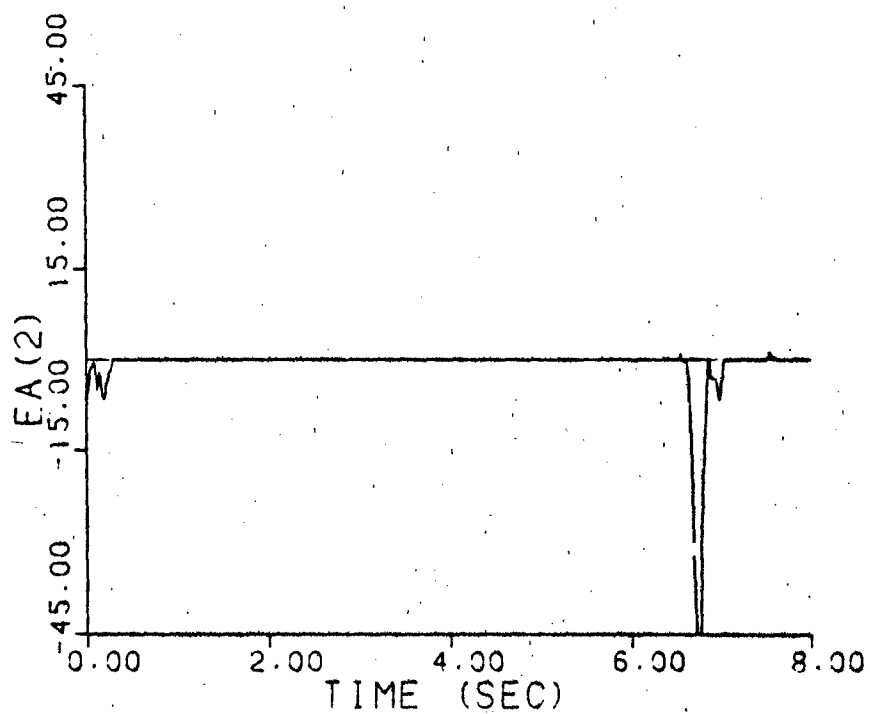
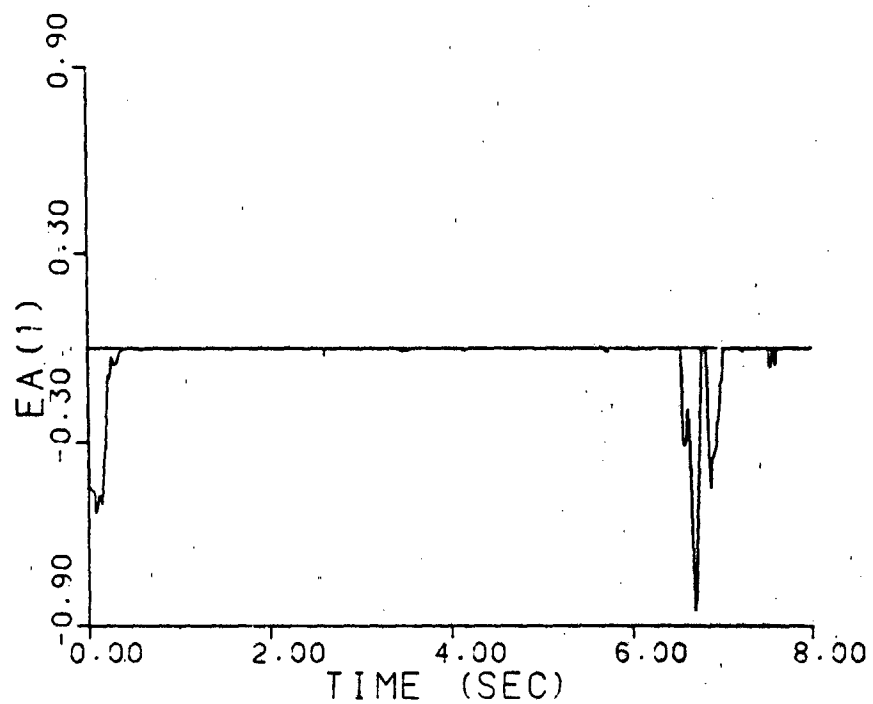


Figure B-8b. Parameter Position and Velocity
Estimate Monitoring
Threshold: 0.04

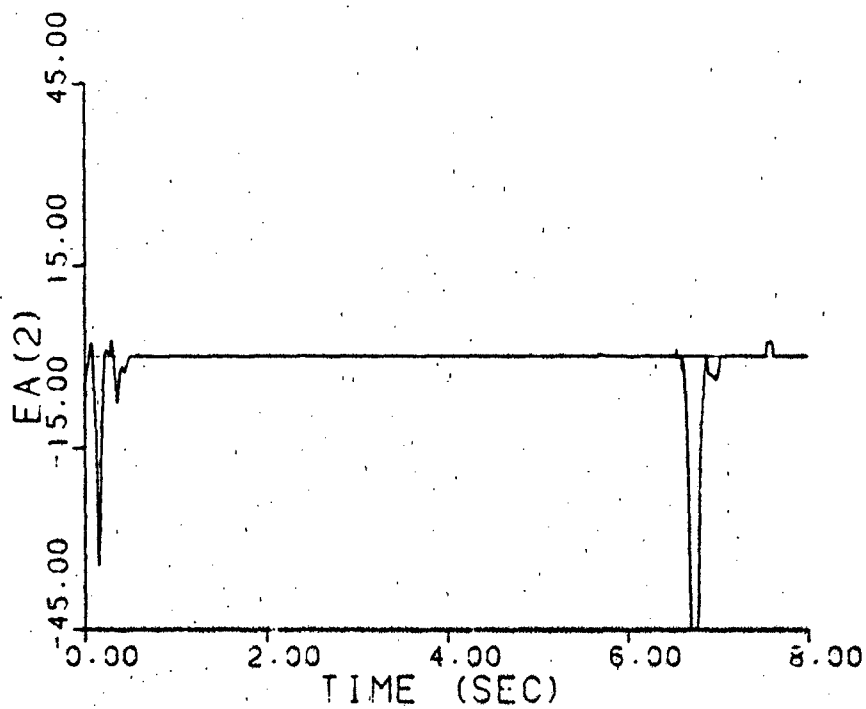
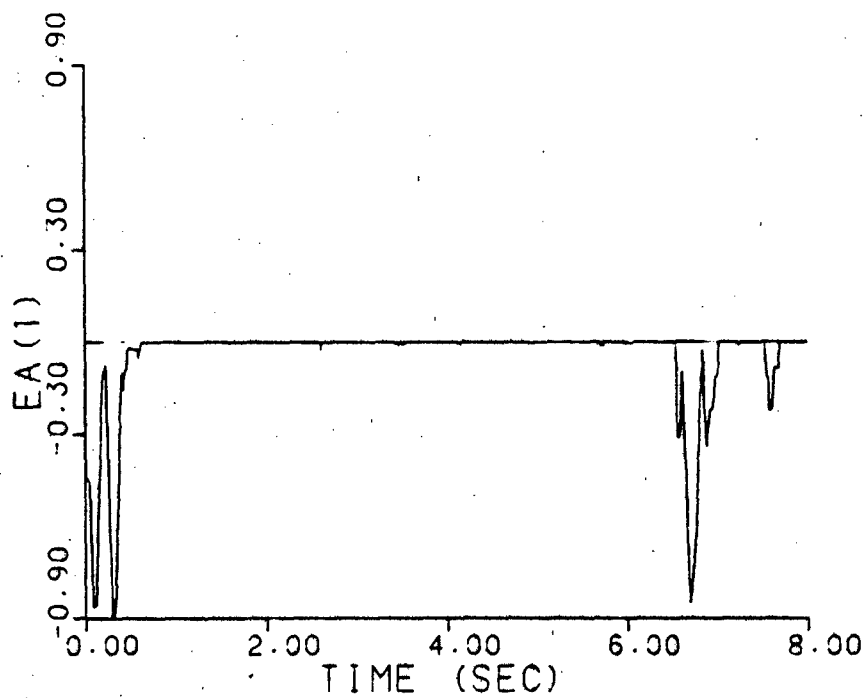


Figure B-8c. Parameter Position and Velocity
Estimate Monitoring
Threshold: 0.01

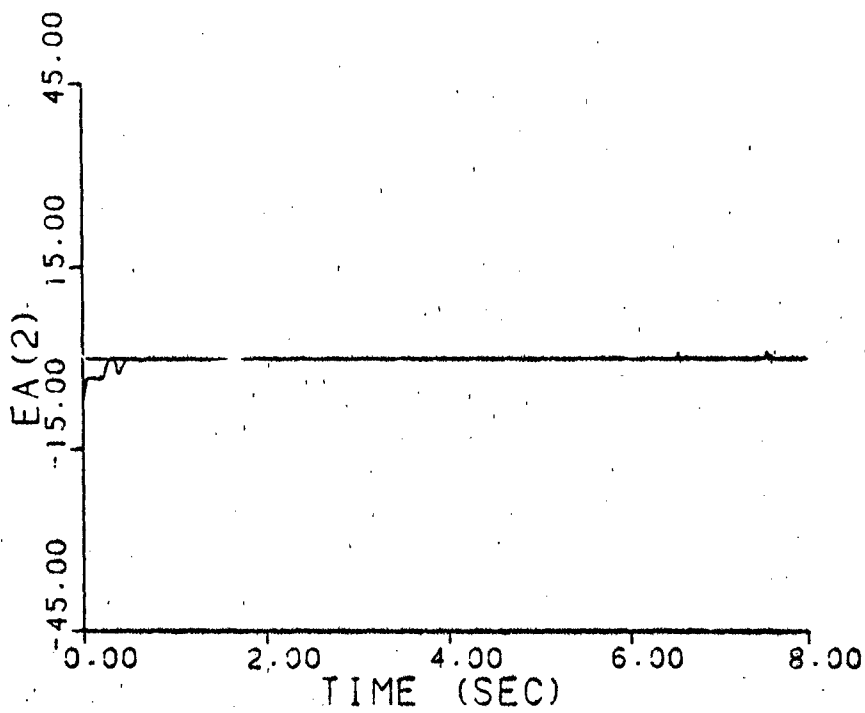
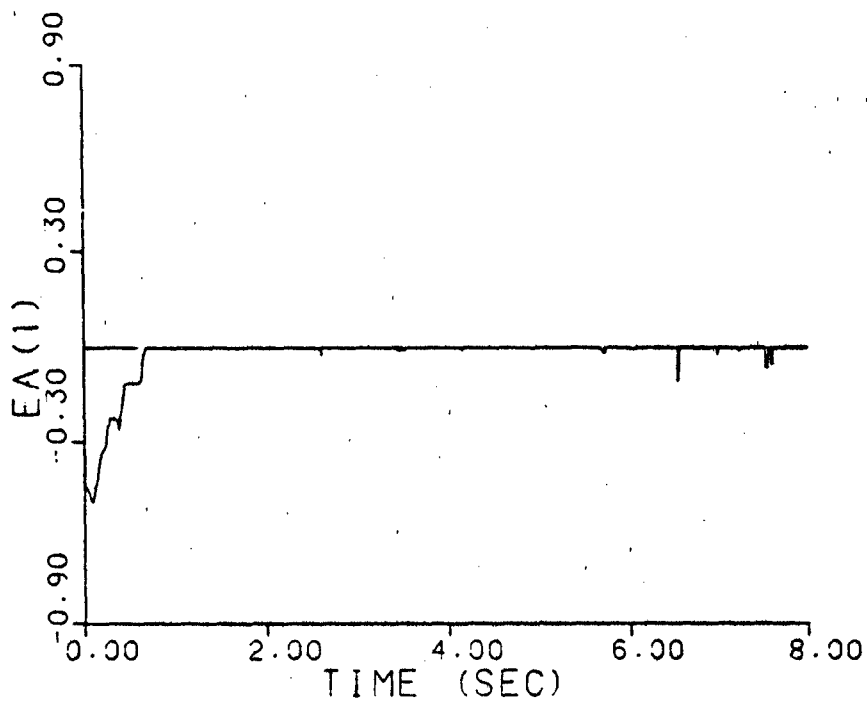


Figure B-9a. Probability Monitoring
Threshold: 0.8

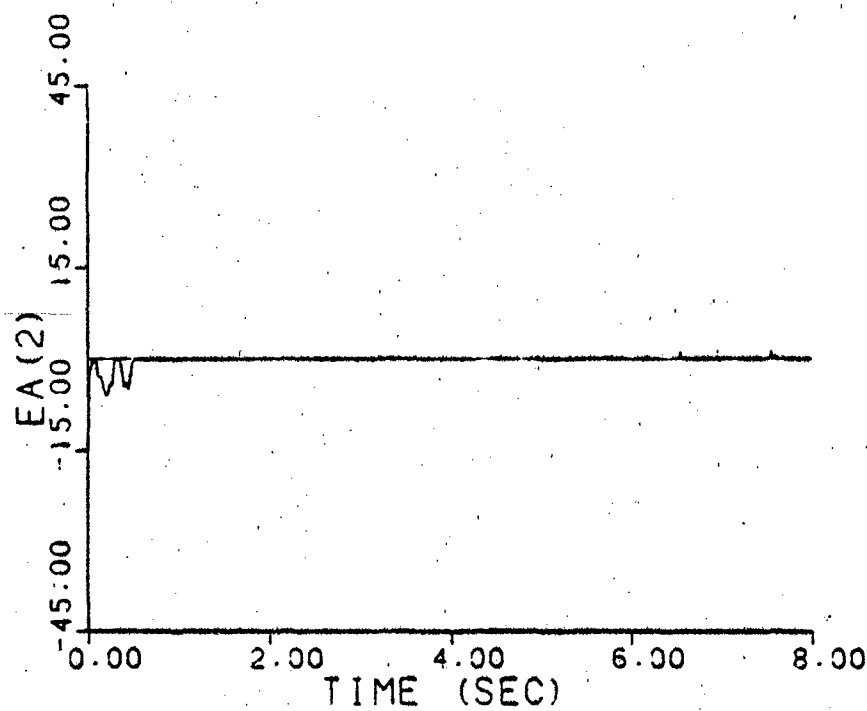
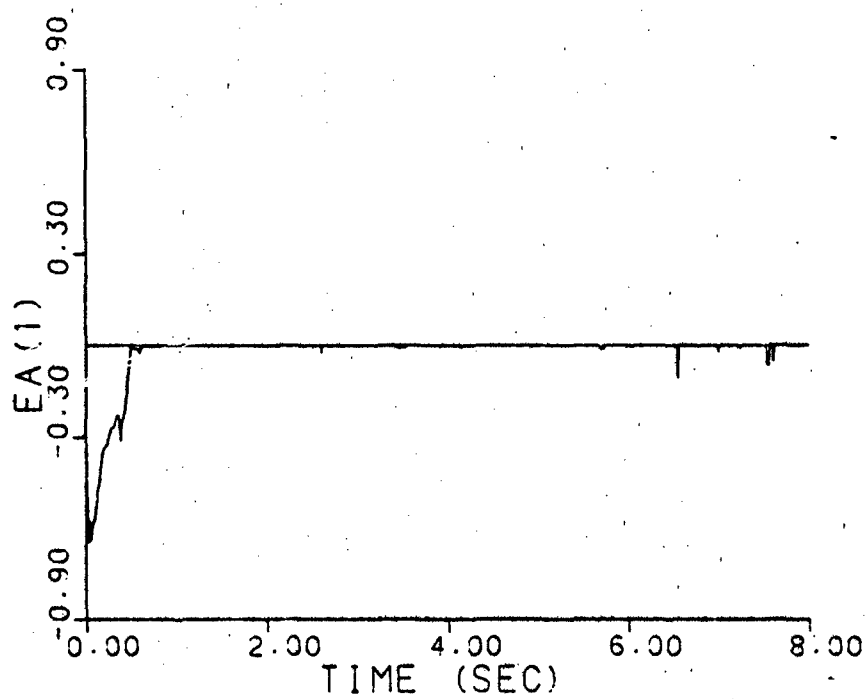


Figure B-9b. Probability Monitoring
Threshold: 0.05

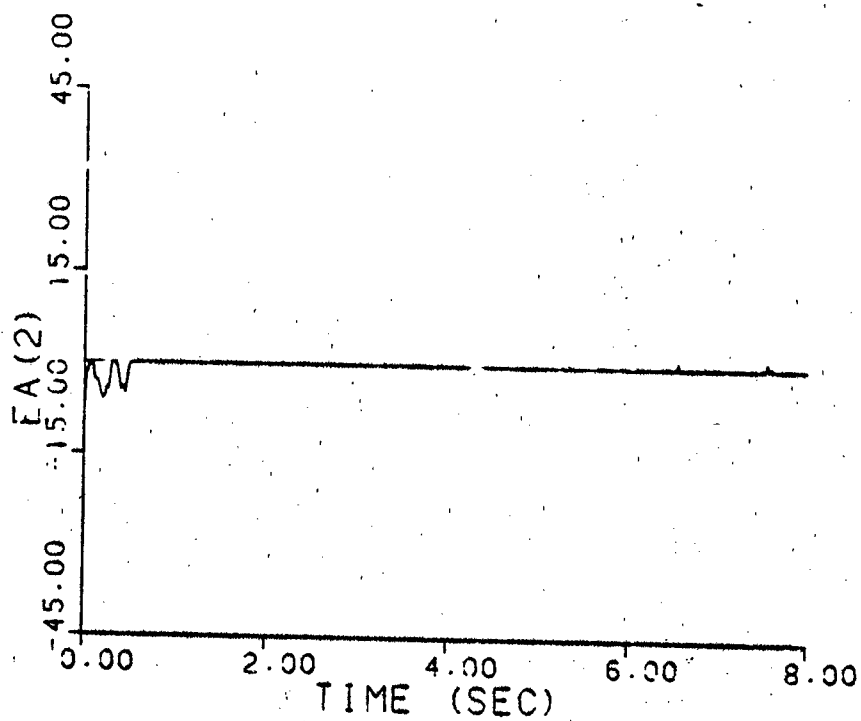
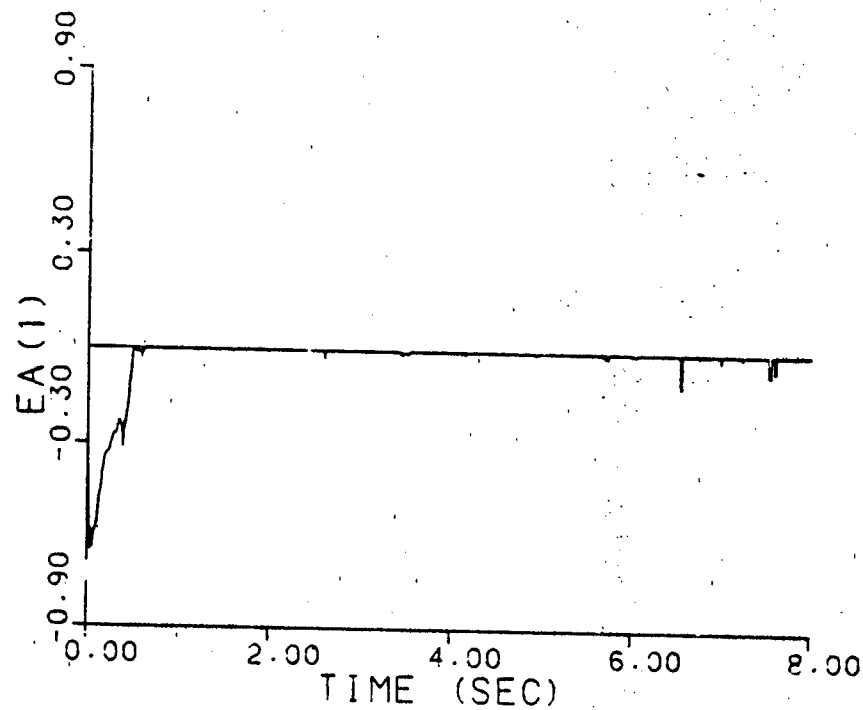


Figure B-9c. Probability Monitoring
Threshold: 0.0

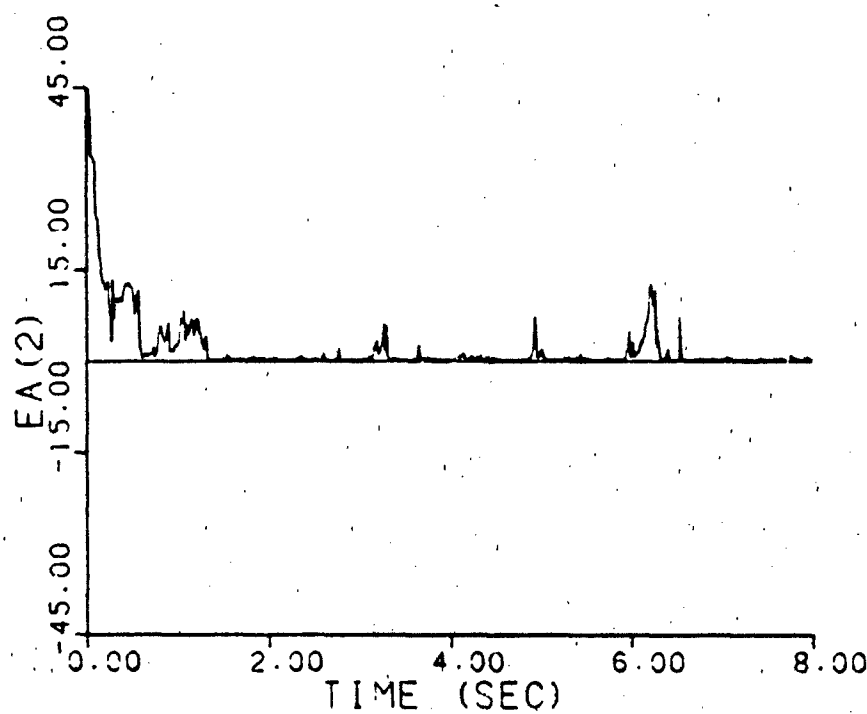
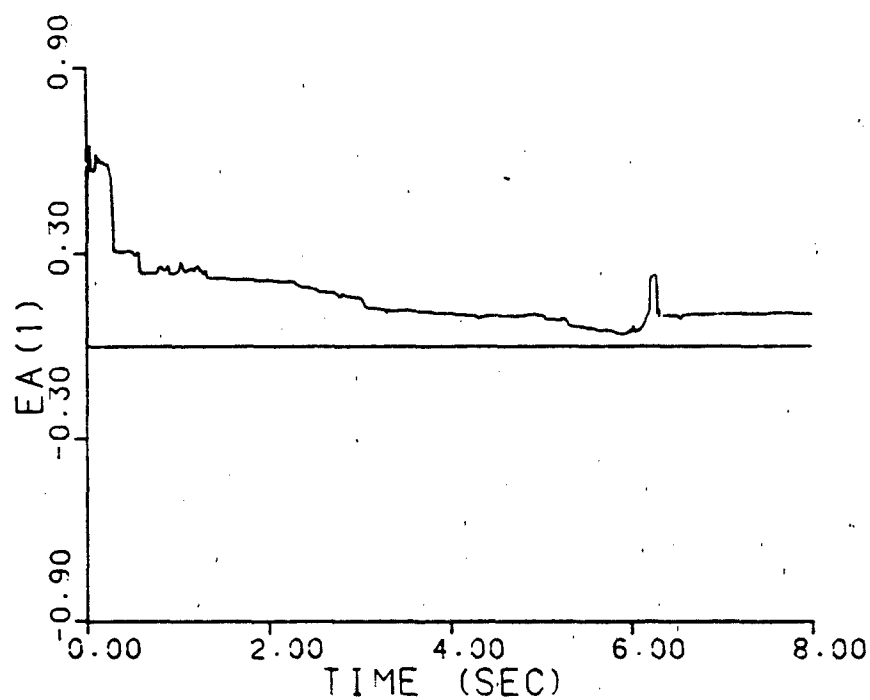


Figure E-10a. Parameter Covariance Monitoring (Contraction)
 Threshold 1: 0.15
 Threshold 2: 0.06

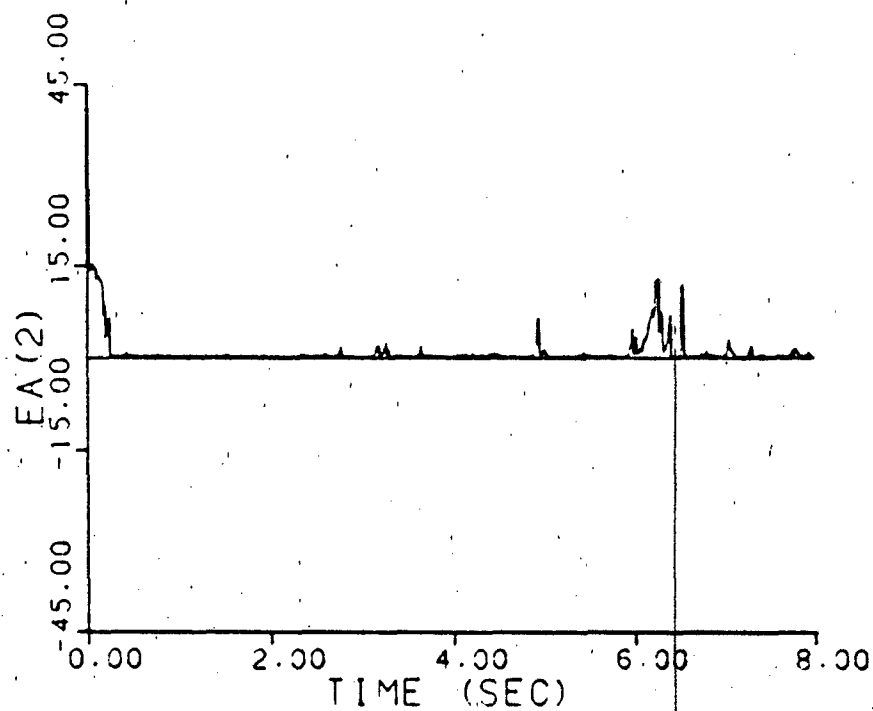
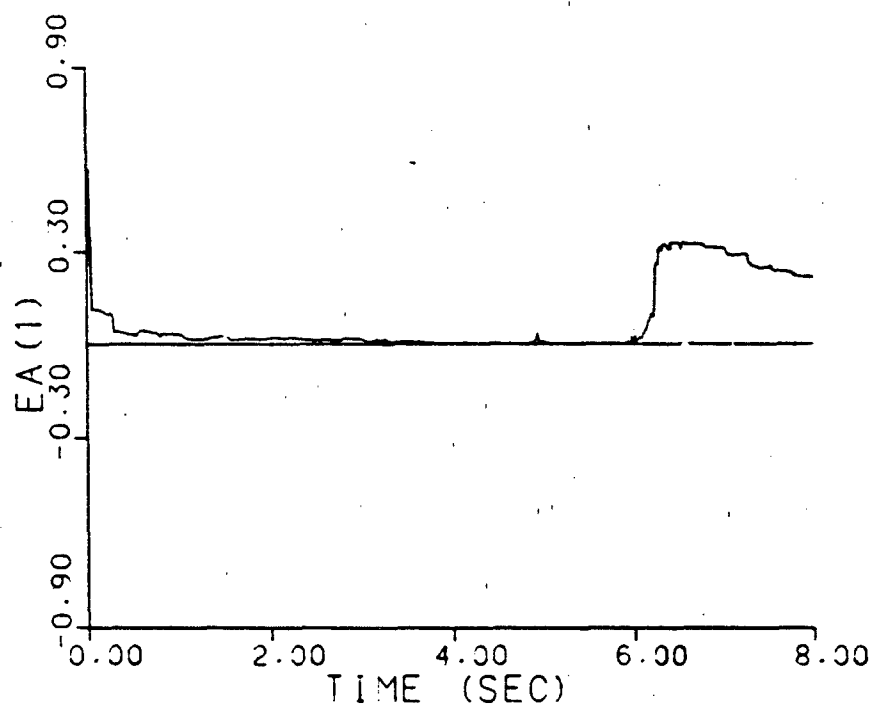


Figure B-10b. Parameter Covariance Monitoring (Contraction)
Threshold 1: 0.03
Threshold 2: 0.0375

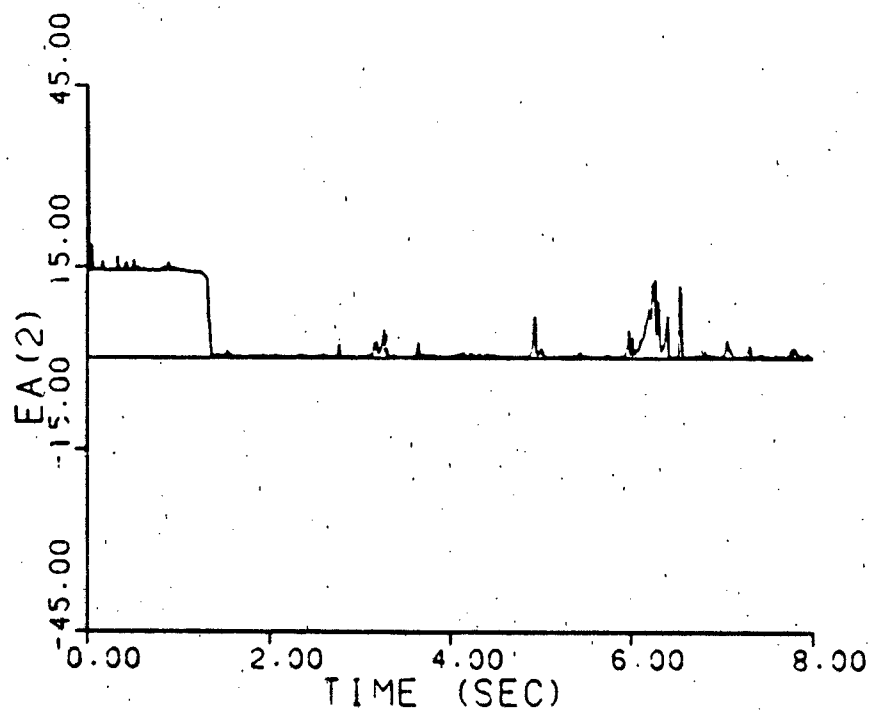
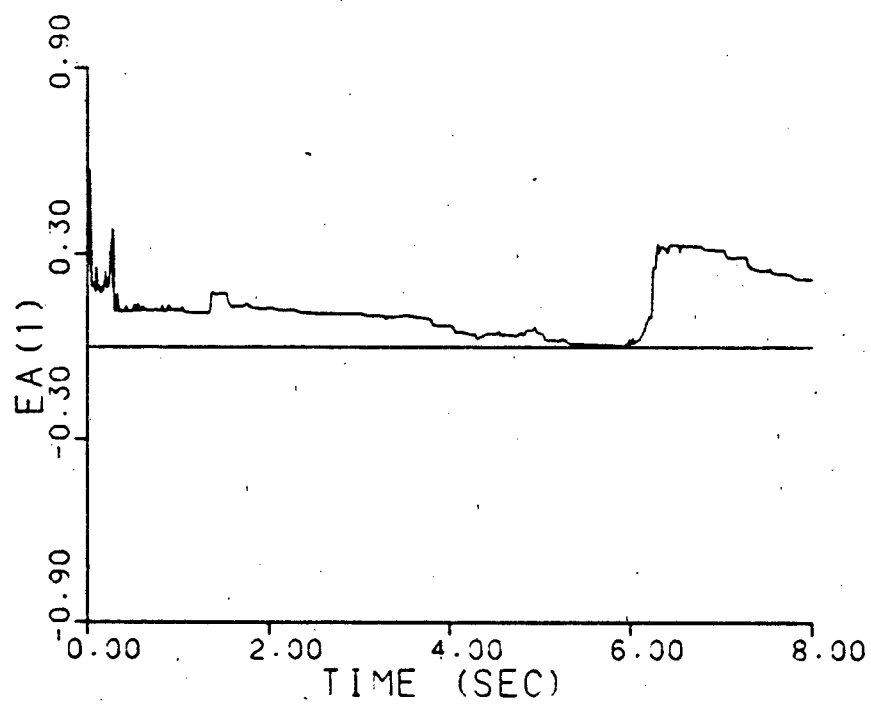


Figure B-10c. Parameter Covariance Monitoring (Contraction)
Threshold 1: 0.003
Threshold 2: 0.0007

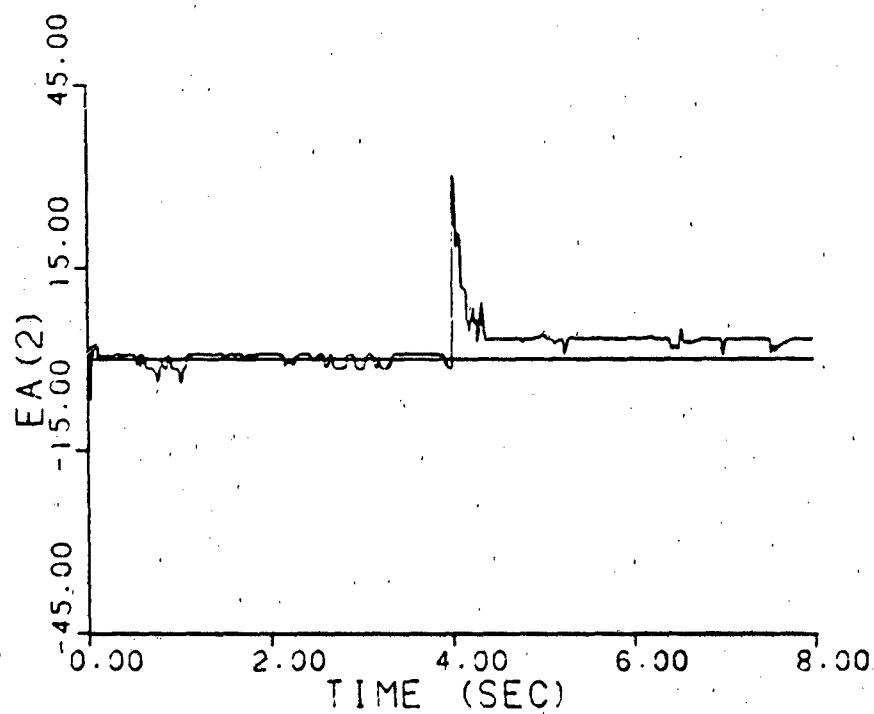
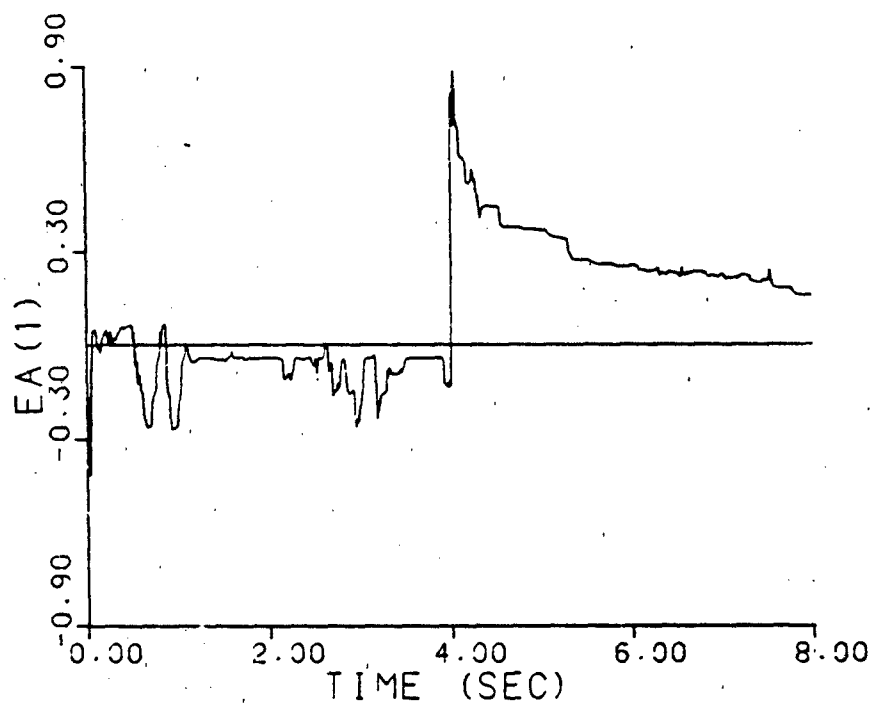


Figure B-11a. Parameter Covariance Monitoring (Expansion)
Threshold: 1000.0

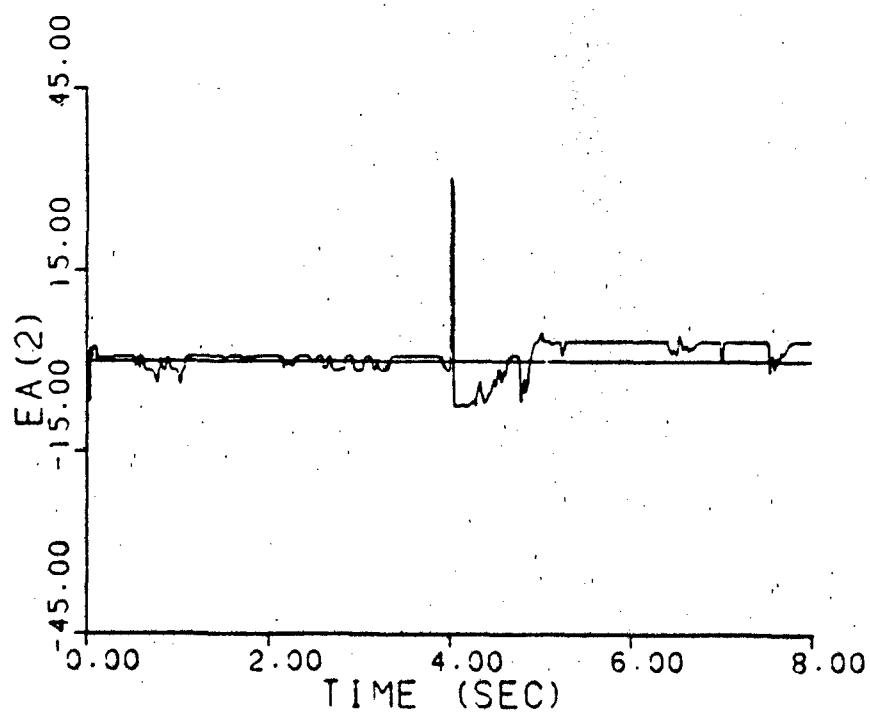
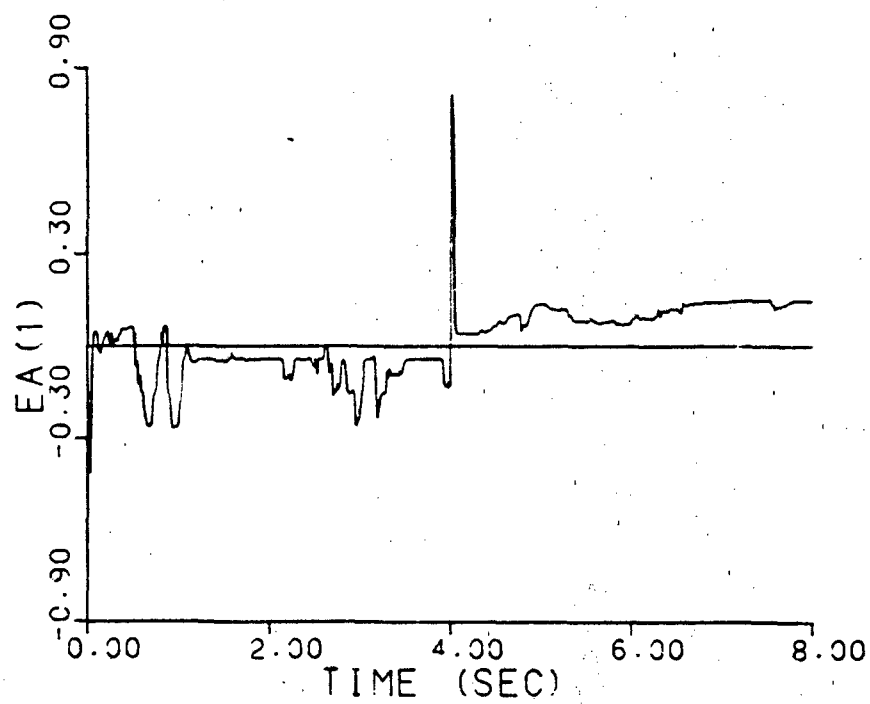


Figure B-11b. Parameter Covariance Monitoring (Expansion)
Threshold: 25.0

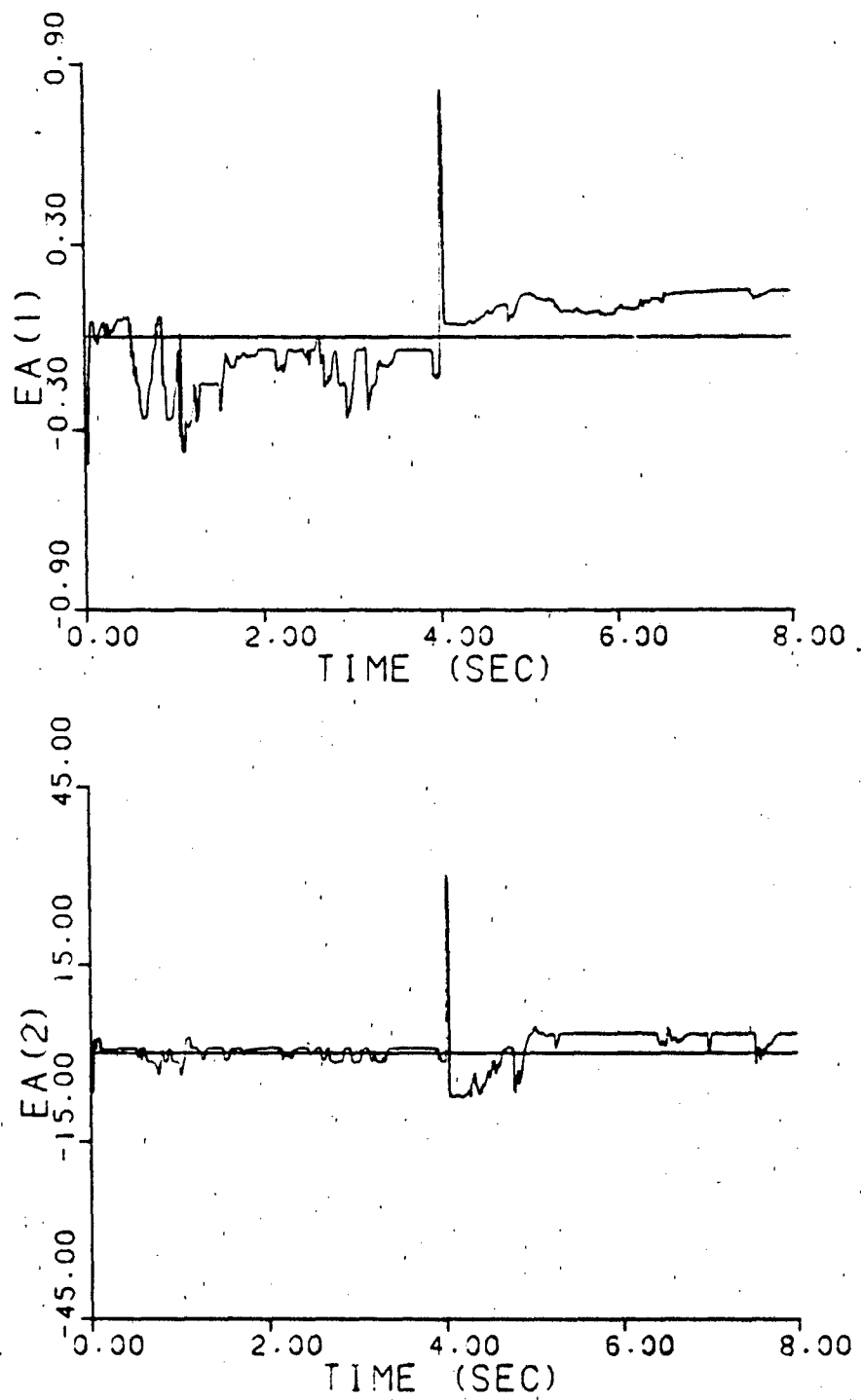


Figure B-11c. Parameter Covariance Monitoring (Expansion)
Threshold: 7.0

Appendix C: Simulation Software

C.1. Introduction

The simulation software created for this thesis consists of two main FORTRAN V programs. The first program (simulation) generates the data necessary for the Monte Carlo analysis and the second program (post processor) reads that data, and then creates plots of the desired variables of interest.

C.2. Simulation

The simulation program consists of an executive routine, BANK, which calls a number of subroutines. The hierarchy of the routines is depicted in Figure C-1.

C.2.1. BANK. BANK is the executive routine. Its functions are data input, initialization of variables before each Monte Carlo run, and data output during each run. Figure C-2 is a flow chart of BANK. There are two main loops in the program. The outer loop controls the number of Monte Carlo runs while the inner loop controls a single run.

C.2.2. TRUE. This routine is called by BANK and its function is to propagate the true system states from one sample time to the next via Equation (II-4). TRUE calls two other subroutines, RW and MTX.

C.2.2.1. RW. Subroutine RW is the random vector generator. It generates a discrete time white Gaussian random vector $\underline{w}_d$ with a mean of 0 and a covariance of I.

C.2.2.2. MTX. This subroutine calculates the state transition, control input, and noise input matrices for

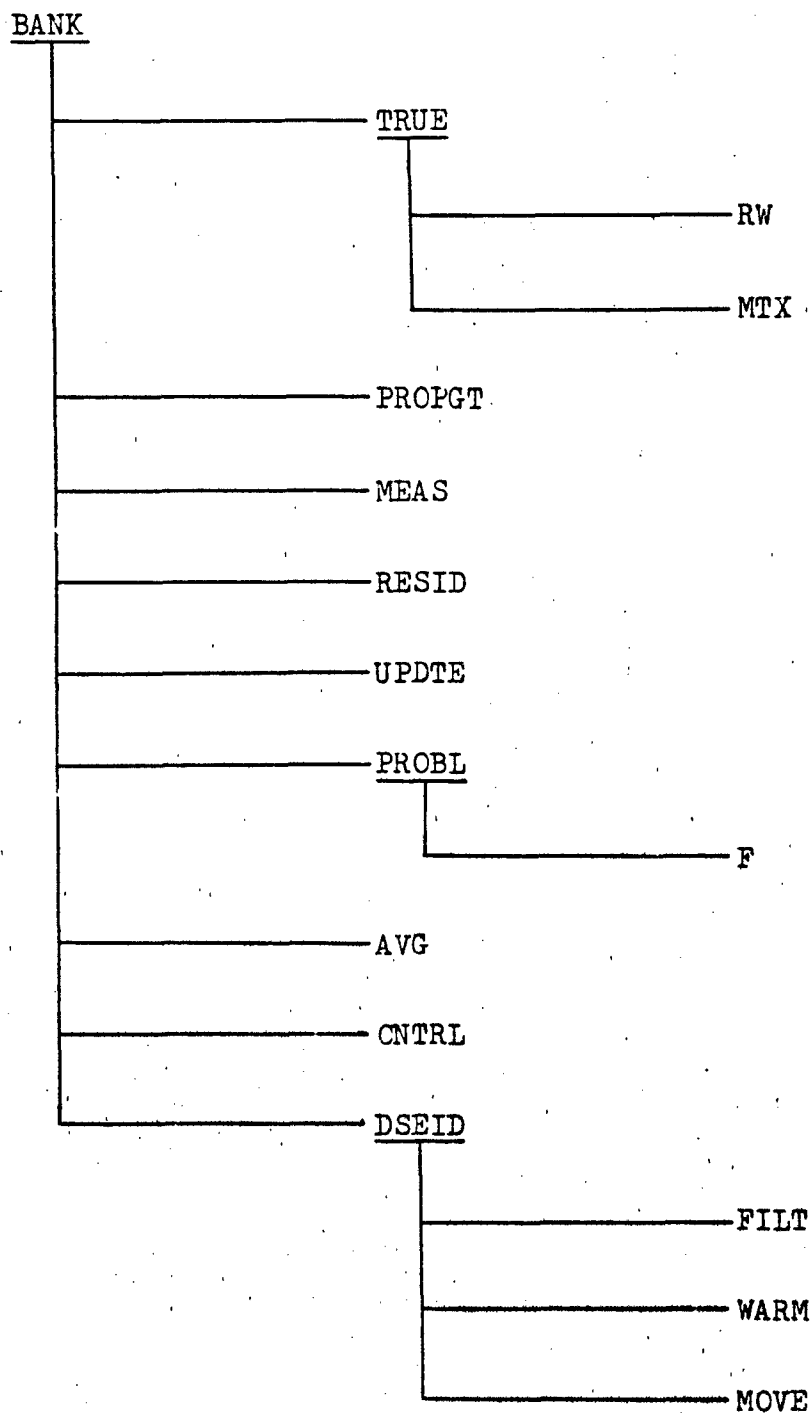


Figure C-1. Simulation Program Hierarchy

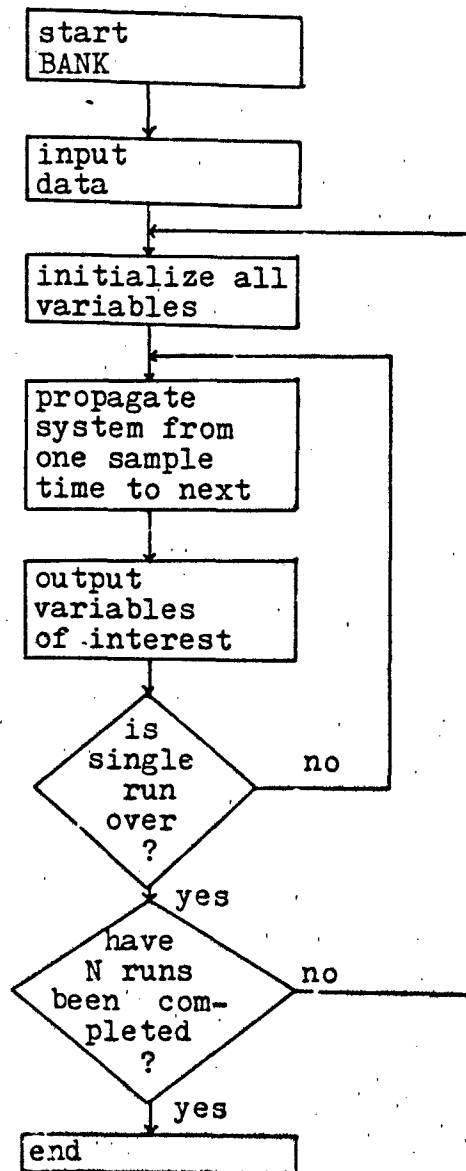


Figure C-2. Program BANK Flow Chart

the true system (Φ_t , B_{d_t} , and G_{d_t} , respectively) from the current value of the parameter vector.

C.2.3. PROPGT. Subroutine PROPGT propagates the state estimates of each elemental filter in the bank from one sample time to the next. It implements the equations

$$\hat{x}_{k-}(t_{i+1}) = \Phi_k \hat{x}_{k-}(t_i^+)$$

C.2.4. MEAS. This subroutine takes a measurement of the true system. MEAS also makes a call to RW (Section C.2.2.1) via the equation

$$z(t_i) = x_{t_i}(t_i) + v(t_i)$$

The measurement is created by adding the first element of the random vector obtained from RW (after multiplying by the square root of the measurement noise variance) to the true system position type state variable.

C.2.5. RESID. This routine computes the residuals for each of the elemental filters of the sliding bank, i. e.,

$$r_k(t_i) = z(t_i) - \hat{x}_{k-}(t_i^-)$$

C.2.6. UPDTE. UPDTE updates the state estimates of each elemental filter with the most recent measurement according to

$$\hat{\underline{x}}_k(t_i^+) = \hat{\underline{x}}_k(t_i^-) + \mathbb{E}_k r_k(t_i)$$

C.2.7. PROBL. This routine calculates the recursive probability weightings $p_k(t_i)$ for each elemental filter. PROBL calls an additional subroutine, F.

C.2.7.1. F. This subroutine computes the numerator terms of the recursive probability calculations (in Equation II-9).

C.2.8. AVG. Subroutine AVG computes the weighted average of the elemental state estimates (Equation II-8). AVG also computes the parameter estimate and the parameter estimate covariance (Equations I-8 and I-9).

C.2.9. CTRL. This subroutine calculates the control to be applied to the true system. There are five versions of this subroutine to allow for no control, the dither signal of Section V.2, the single changeable-gain controller of Section III.3, the single fixed-gain controller of Section III.4, and sliding bank multiple model adaptive control (Section III.5).

C.2.10. DBRID. DBRID controls movement of the bank. It calls FILT, WARM, and MOVE. After DBRID calls FILT, the decision is made either to expand, contract, move, or warm up filters, depending upon the threshold levels. If it is decided simply to warm up filters, only WARM will be called and DBRID will return to AVG. If no filters are currently warming up and it is decided to move (or expand/contract) the

bank, WARM and MOVE will be called consecutively. If filters are warming up and it is decided to move, MOVE alone will be called. If it has been decided not to use filter warm-up, setting the warm-up and the move thresholds to the same value eliminates the possibility of filter warm-up occurring.

3.2.10.1. PIBT. This subprogram returns both a value that is to be compared to the threshold and the move direction associated with that value. The value is calculated according to the decision method in operation (supplied by the simulation program input data file).

3.2.10.2. WARM. This routine transfers all of the appropriate matrices to an array of nine additional filters to be "warmed up". Each filter is initialized with its appropriate state estimate and probability weighting.

3.2.10.3. MOVE. This routine transfers all of the matrices, state estimates, and probability weightings from the nine additional filters established in WARM to the nine on line filters.

3.3. Post Processor

The post processor program consists of a main routine and one subroutine. The main program computes means and variances and the subroutine creates plots.

3.3.1. MAIN. This is the main program. MAIN reads a data file generated by PIPT and computes the means and variances of the variation of interest. The output is the means and variances of the variation of interest to a prin-

ter file. RESLT makes one call to subroutine SLOT8.

C.3.2. SLOT8. This subroutine generates a plot file of the means and variances computed in RESLT.

C.4. Summary

This appendix has discussed each of the routines contained in the simulation software. Two primary programs are used, one for the simulation which generates the necessary data and one for post processing which computes statistics of the data and generates plots.

Appendix D: Ambiguity Function Analysis (6:97-99)

Much information about the expected performance of an adaptive estimator can be obtained from an ambiguity function analysis. The generalized ambiguity function is given by

$$A_i(\underline{a}, \underline{a}_t) \triangleq \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} L[\underline{a}, \underline{Z}_i] f_{\underline{Z}(t_i)}(\underline{Z}_i | \underline{a}_t) d\underline{Z}_i \quad (D-1)$$

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. For a given value of $\underline{a}_t$, this function of $\underline{a}$ yields information about the expected ability of the estimator to estimate parameters. The curvature of the ambiguity function at $\underline{a}_t$ predicts the precision with which the adaptive estimator can estimate the parameters: the sharper the curvature, the greater the precision. This curvature is inversely related to the Cramer Rao lower bound on the estimate error covariance matrix by

$$E\{[\underline{a} - \underline{a}_t][\underline{a} - \underline{a}_t]^T\} \geq \left[- \frac{\partial^2}{\partial \underline{a}^2} A_i(\underline{a}, \underline{a}_t) \Big|_{\underline{a}=\underline{a}_t} \right]^{-1}$$

Global parameter space performance is also obtained from the ambiguity function. If there are multiple peaks of the ambiguity function, the adaptive estimator may converge to incorrect parameter values corresponding to one of the local maxima or it may not converge at all.

The ambiguity function (Equation D-1) 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 is given by

$$\begin{aligned} A_i(\underline{a}, \underline{a}_t) &= \frac{1}{2\pi} \ln(2\pi) - \frac{1}{2} \ln \left[|A(t_i, \underline{a})| \right] \\ &\quad - \frac{1}{2} \text{tr} \left\{ A^{-1}(t_i, \underline{a}) \left[Y(t_i) P_e(t_i^-; \underline{a}_t, \underline{a}) Y^T(t_i) + R(t_i) \right] \right\} \\ &\quad - \frac{1}{2\pi} \ln(2\pi) - \frac{1}{2} \ln \left[|r(t_i^+, \underline{a})| \right] \\ &\quad - \frac{1}{2} \text{tr} \left\{ r^{-1}(t_i^+, \underline{a}) r_e(t_i^+; \underline{a}_t, \underline{a}) \right\} \quad (D-2) \end{aligned}$$

where

$$A(t_i, \underline{a}) = \left[Y(t_i) P(t_i^-, \underline{a}) Y^T(t_i) + R(t_i) \right]^{-1} \text{ of the Kalman filter based on } \underline{a}$$

$P_e(t_i^+; \underline{a}_t, \underline{a})$ = 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.

For each of the true parameter points examined in this thesis, Equation (D-2) was evaluated. The true parameter vector $\underline{a}_t$ was set equal to a parameter point under study and $A_i(\underline{a}, \underline{a}_t)$ was calculated for $\underline{a}$ equaling each of the discretized points in the parameter space. This process was then repeated for other values of $\underline{a}_t$. The resulting surfaces are presented in Figures 4-1 through 4-7. (Note: arrow located $\underline{a}_t$.)

ZETA: 0.000 WN: 10.48

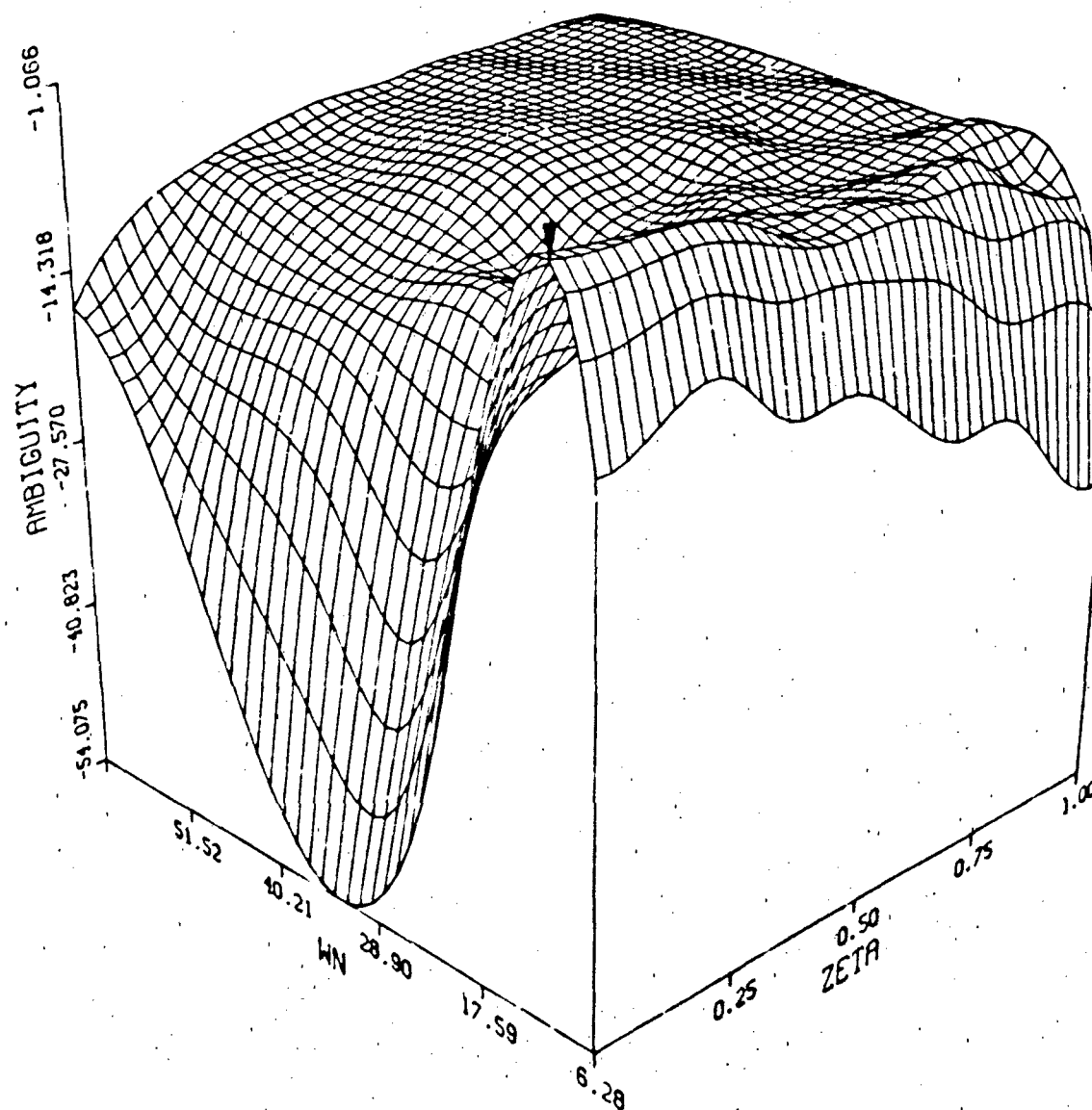


Figure D-1a. Ambiguity Function, $\underline{a}_t = (1, 3)$

ZETA: 0.000 WN: 10.48

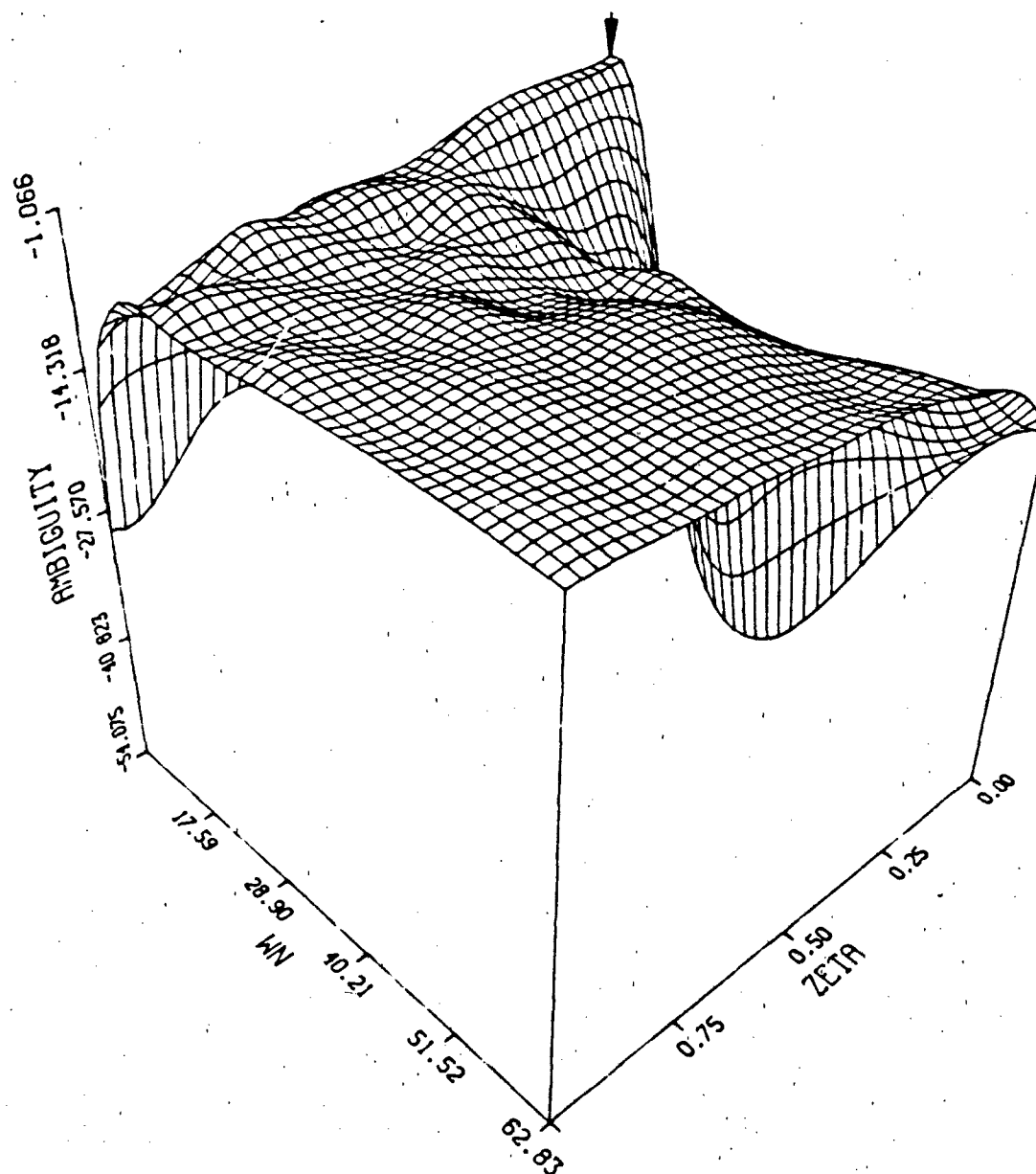


Figure D-1b. Ambiguity Function, $\underline{a}_t = (1, 3)$

ZETA: .1111 WN: 48.65

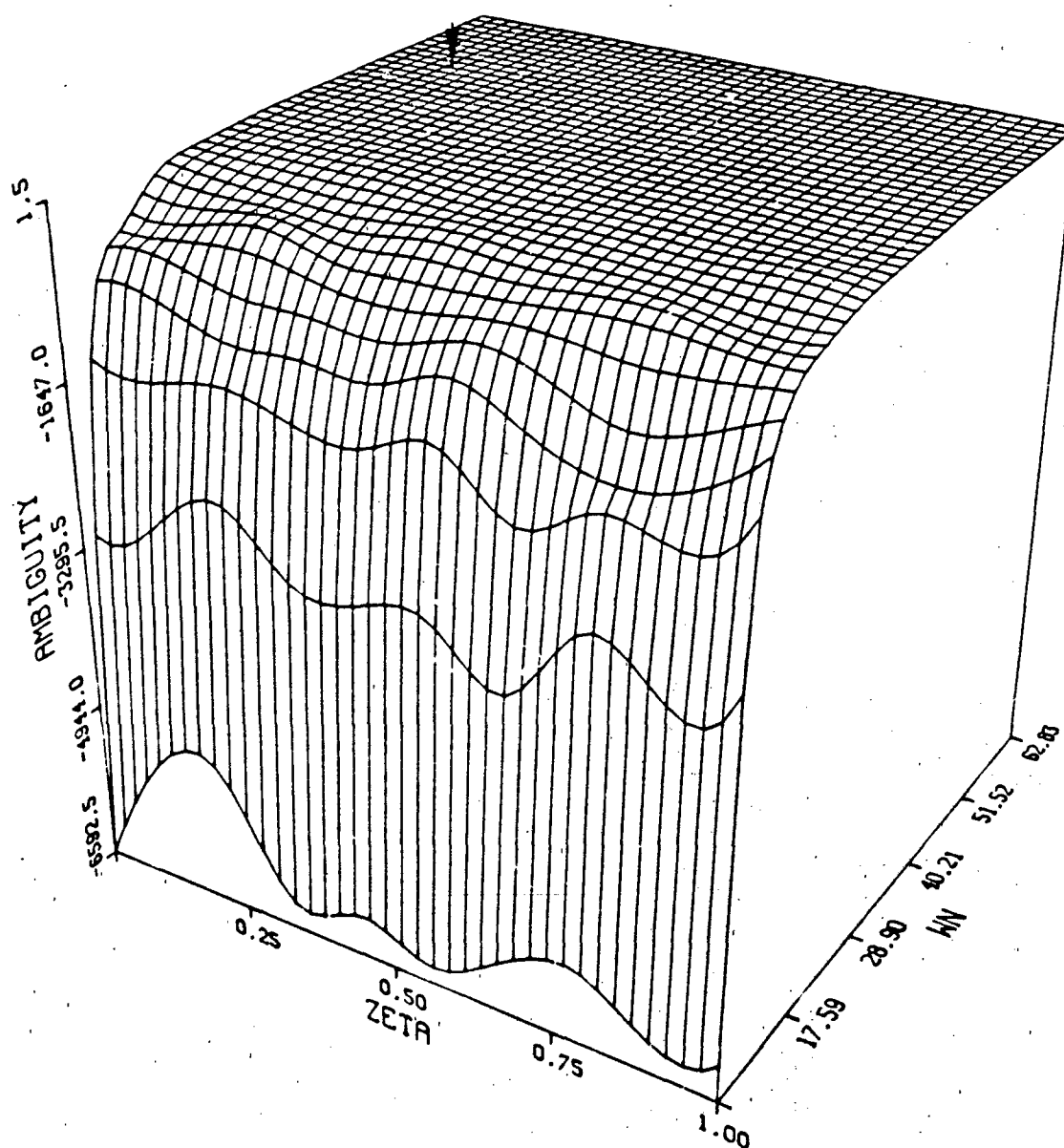


Figure D-2. Ambiguity Function, $\underline{a}_t = (2, 9)$

ZETA: .4444 WN: 13.54

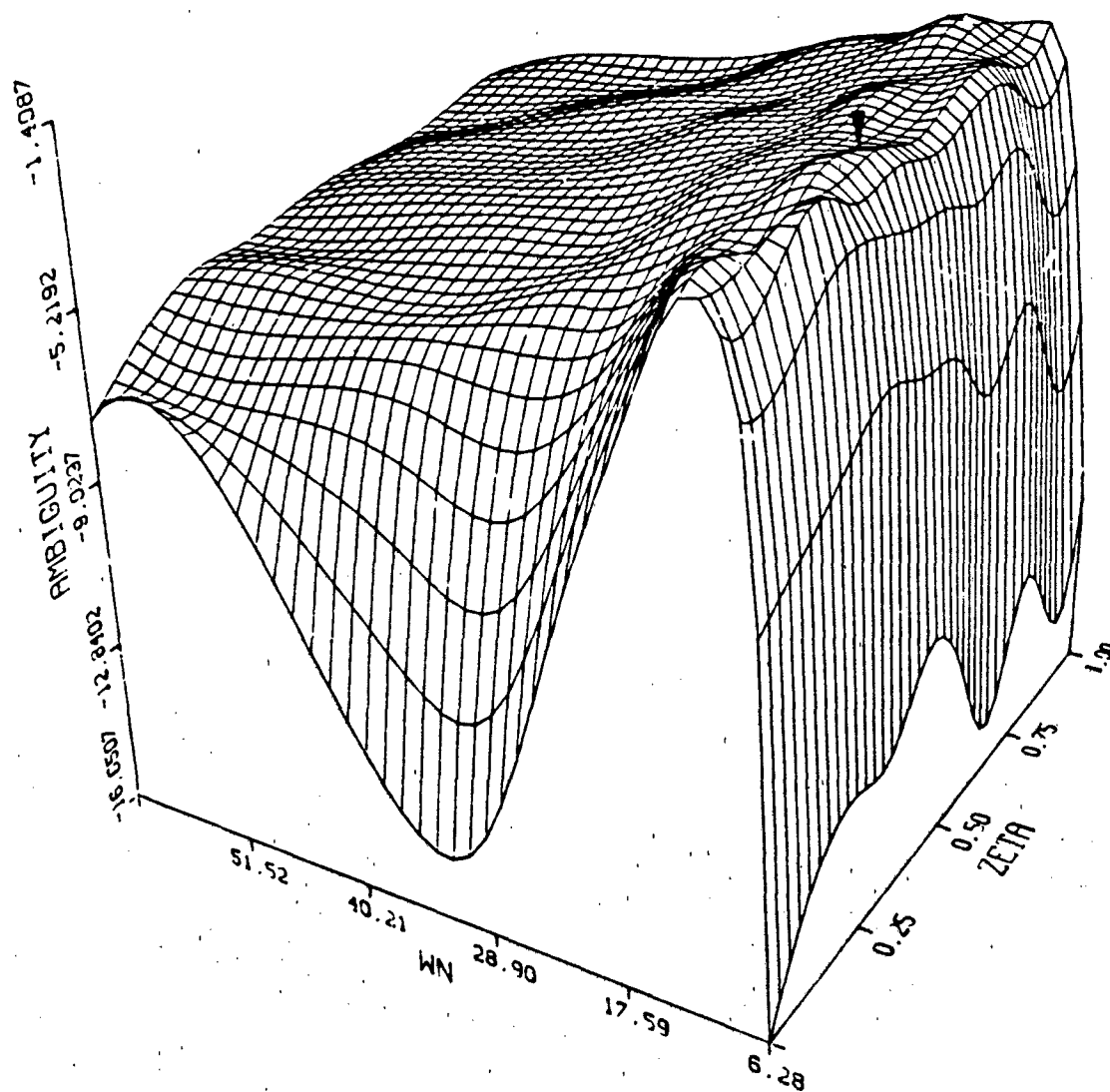


Figure D-3a. Ambiguity Function, $\underline{a}_t = (5, 4)$

ZETA: .4444 WN: 13.54

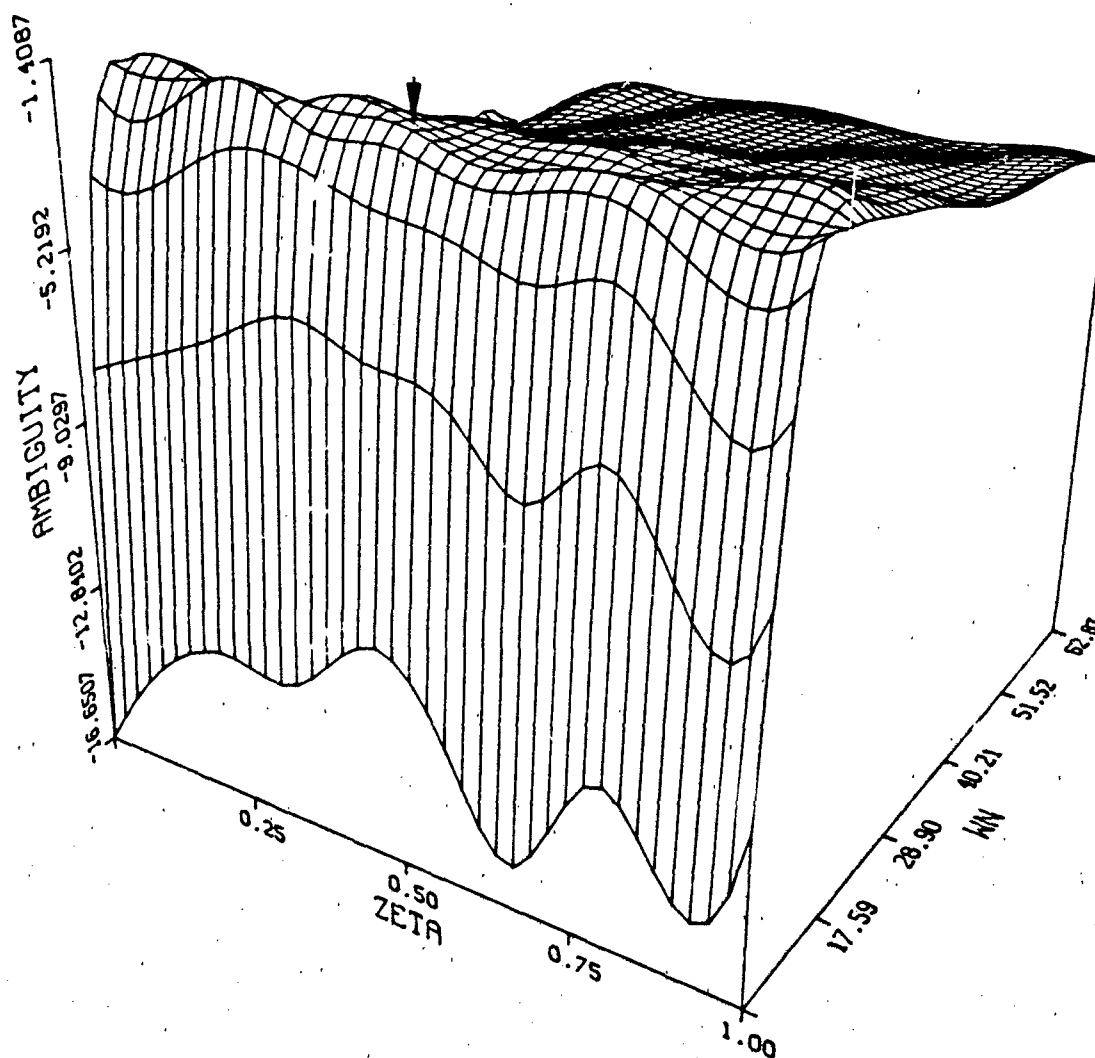


Figure D-3b. Ambiguity Function, $\underline{a}_t = (5, 4)$

ZETA: .8889 WN: 8.12

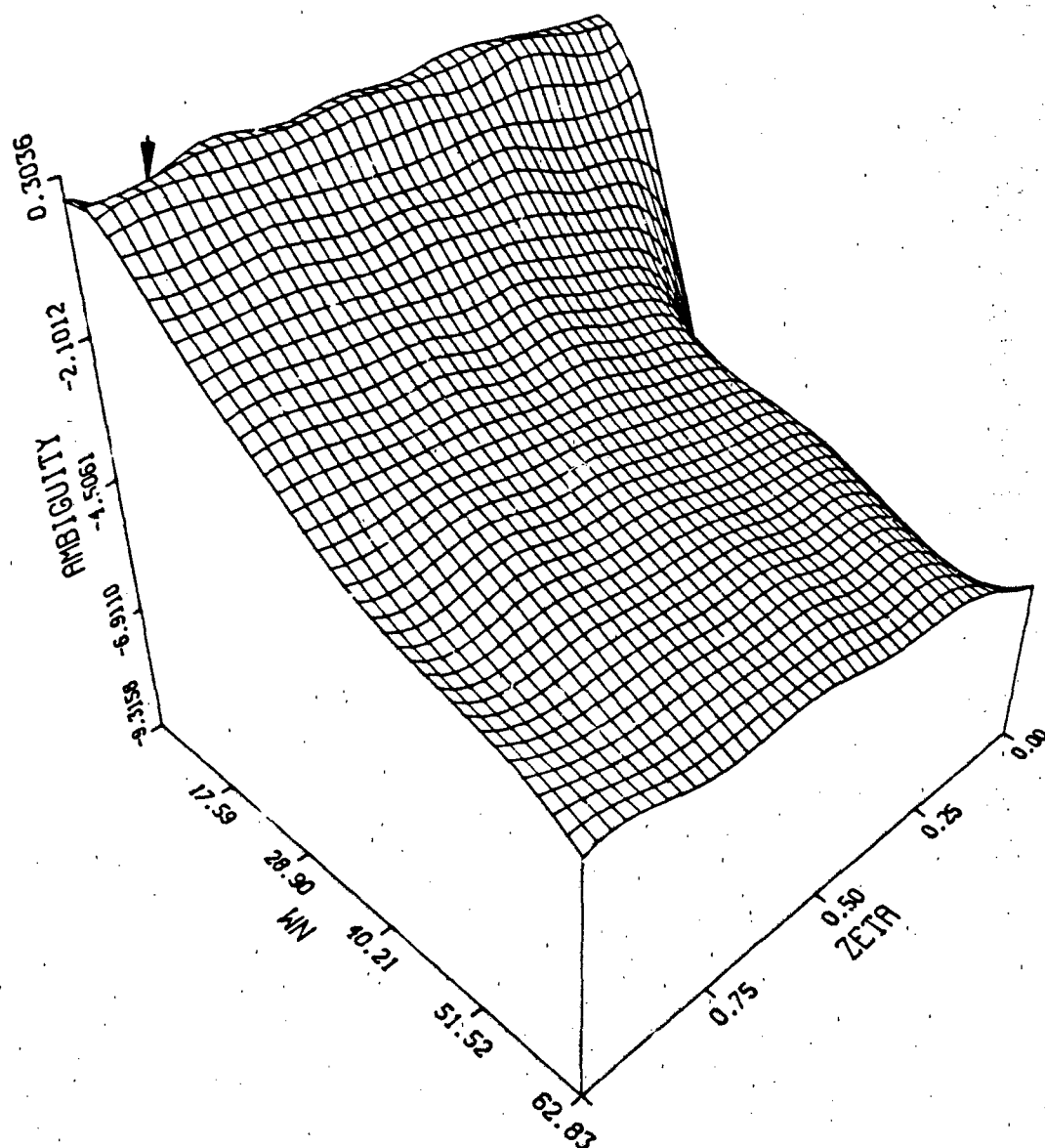


Figure D-4. Ambiguity Function, $\underline{a}_t = (9, 2)$

ZETA: 1.000 WN: 62.83

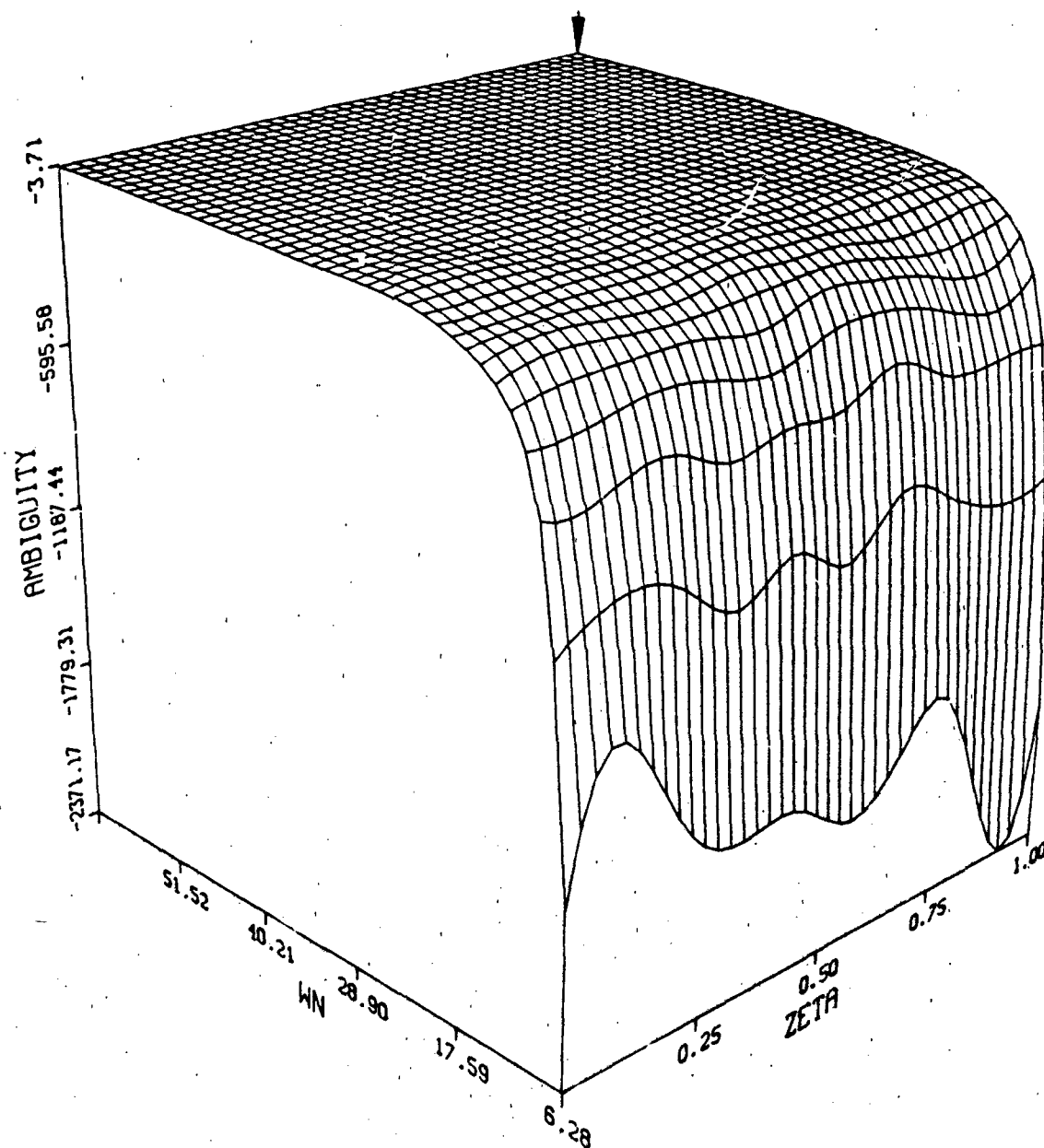


Figure D-5. Ambiguity Function, $\underline{a}_t = (10, 10)$

ZETA: .0700 WN: 9.00

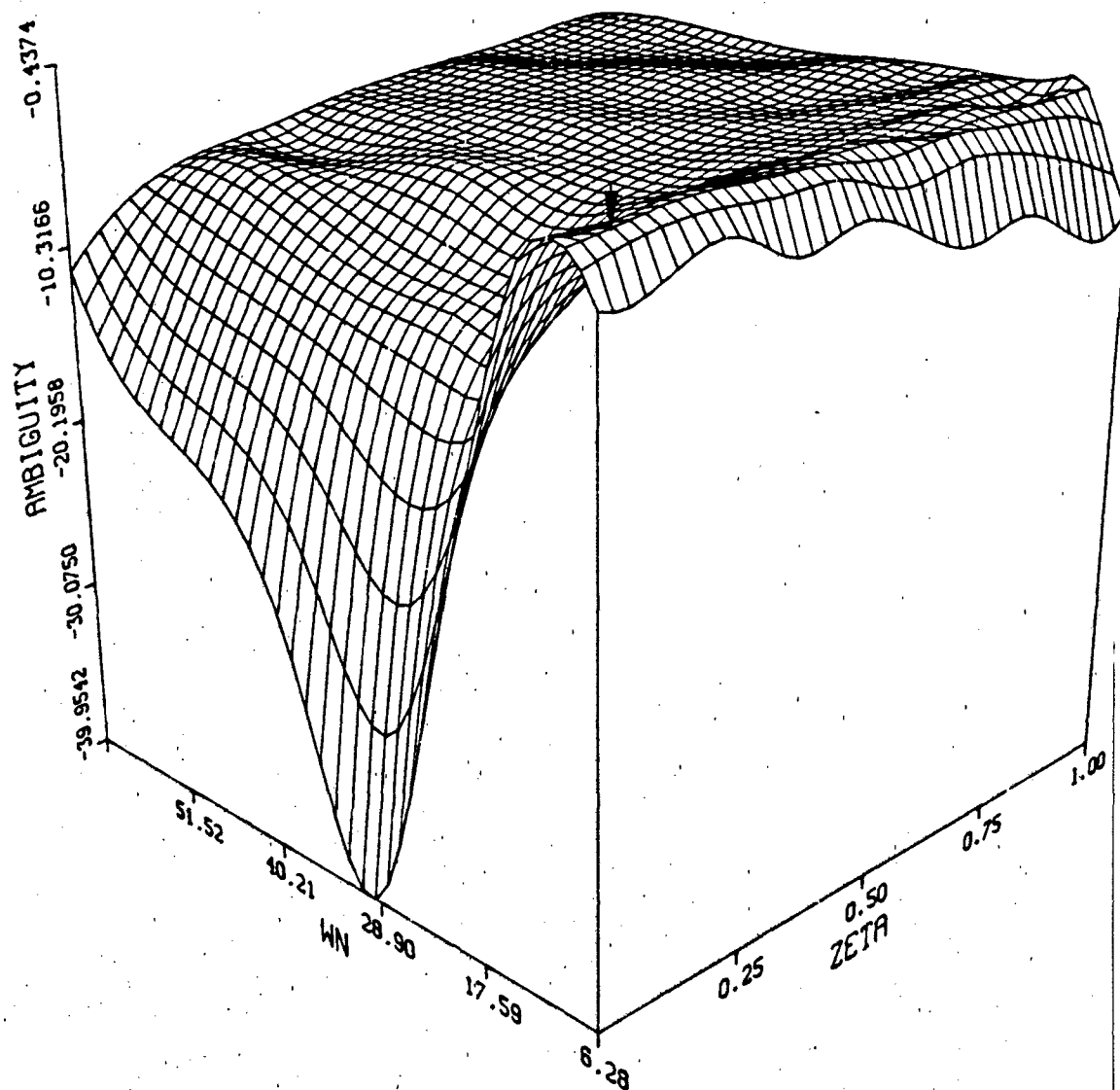


Figure D-6a. Ambiguity Function, $\underline{a}_t = (0.07, 9.0)$

ZETA: .0700 WN: 9.00

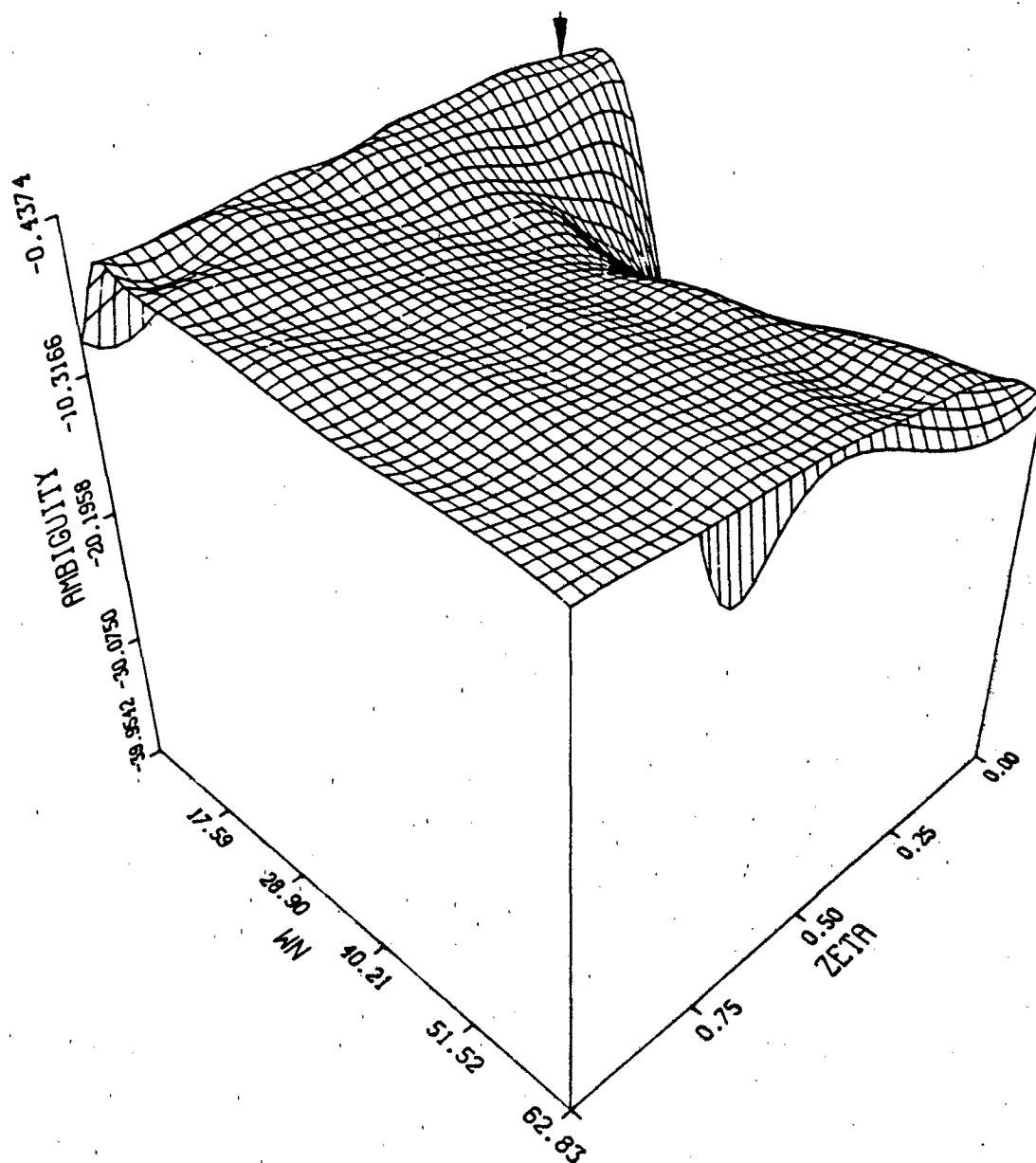


Figure D-6b. Ambiguity Function, $\underline{a}_t = (0.07, 9.0)$

ZETA: .9300 WN: 41.00

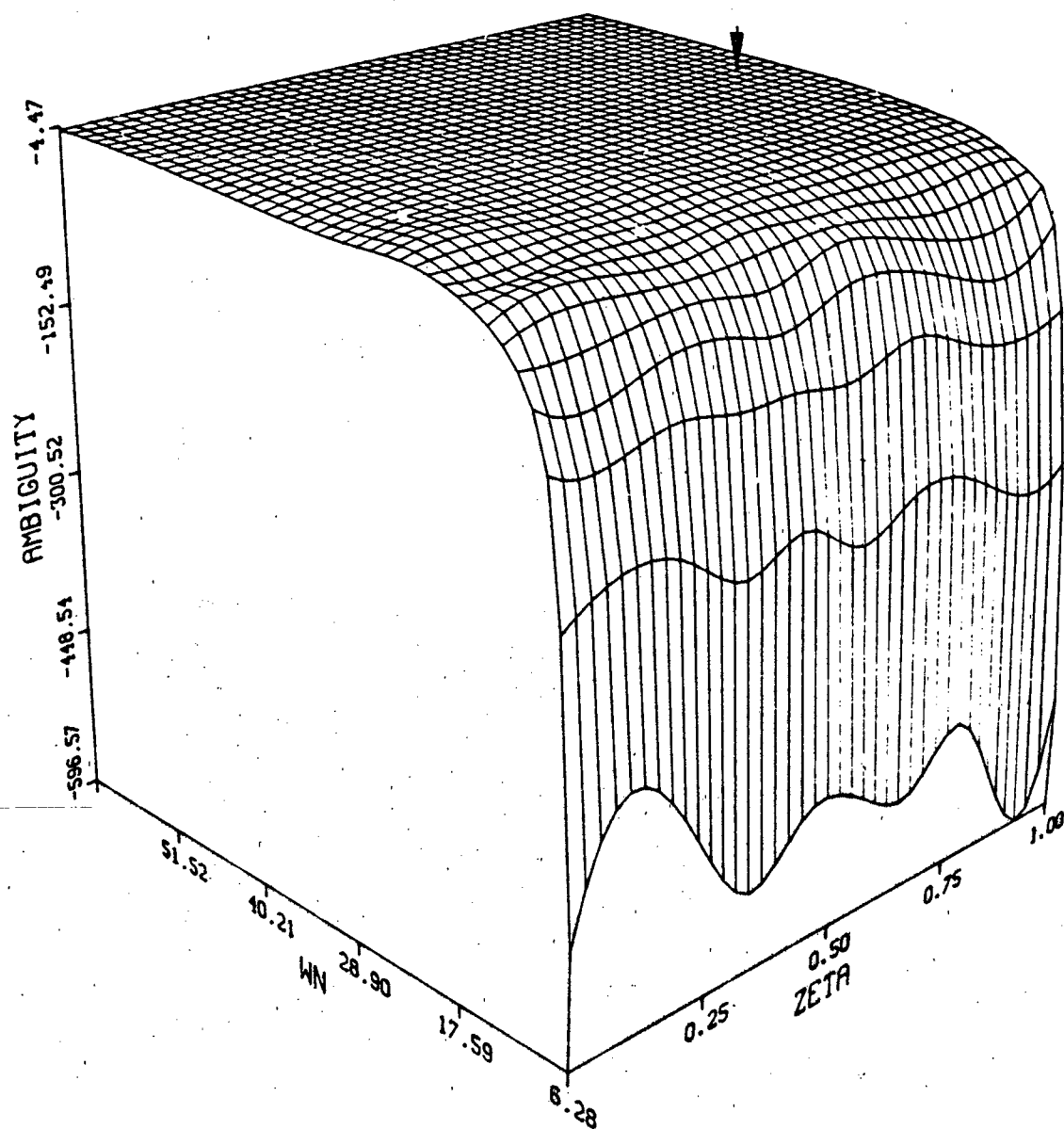


Figure D-7. Ambiguity Function, $\underline{a}_t = (0.93, 41.0)$

Bibliography

1. Asher, Robert E. et al. "Bibliography on Adaptive Control Systems," Proceedings of the IEEE, 64: 1226-1240 (August 1976).
2. Magill, D. T. "Optimal Adaptive Estimation of Sampled Stochastic Processes," IEEE Transactions on Automatic Control, 10: 434-439 (October 1965).
3. Hilborn, C. G., Jr. and Demetrios G. Lainiotis. "Optimal Estimation in the Presence of Unknown Parameters," IEEE Transactions on Systems Science and Cybernetics, 5: 38-43 (January 1969).
4. Lainiotis, Demetrios G. "Optimal Adaptive Estimation: Structure and Parameter Adaptation," IEEE Transactions on Automatic Control, 16: 160-170 (April 1971).
5. ----- "Partitioning: A Unifying Framework for Adaptive Systems, I: Estimation," Proceedings of the IEEE, 64: 1126-1143 (August 1976).
6. Maybeck, Peter S. Stochastic Models, Estimation, and Control, Volume 2. New York: Academic Press, 1982.
7. Hawkes, Richard M. and John B. Moore. "Performance Bounds for Adaptive Estimation," Proceedings of the IEEE, 64: 1143-1150 (August 1976).
8. Dasgupta, S. and L. C. Westphal. "Convergence of Partitioned Adaptive Filters for Systems with Unknown Biases," IEEE Transactions on Automatic Control, 28: 614-615 (May 1983).
9. Maybeck, Peter S. Stochastic Models, Estimation, and Control, Volume 3. New York: Academic Press, 1982.
10. Thcrp, James S. "Optimal Tracking of Maneuvering Targets," IEEE Transactions on Aerospace and Electronic Systems, 9: 512-519 (July 1973).
11. Moose, Richard L. "An Adaptive State Estimation Solution to the Maneuvering Target Problem," IEEE Transactions on Automatic Control, 20: 359-362 (June 1975).
12. Gholson, Norman H. and Richard L. Moose. "Maneuvering Target Tracking Using Adaptive State Estimation," IEEE Transactions on Aerospace and Electronic Systems, 13: 310-317 (May 1977).

13. Moose, Richard L. et al. "Modeling and Estimation for Tracking Maneuvering Targets," IEEE Transactions on Aerospace and Electronic Systems, 15: 448-456 (May 1979).
14. Korn, J. and L. Beem. Application of Multiple Model Adaptive Estimation Algorithms to Maneuver Detection and Estimation. Contract DAAK10-82-C-0020. ALPHATECH, Inc., Burlington, MA, June 1983 (AD-B075921).
15. Athans, Michael et al. "The Stochastic Control of the F-8C Aircraft Using a Multiple Model Adaptive Control (MMAC) Method - Part 1: Equilibrium Flight," IEEE Transactions on Automatic Control, 22: 768-780 (October 1977).
16. Sims, Craig S. and M. R. D'Mello. "Adaptive Deconvolution of Seismic Signals," IEEE Transactions on Geoscience Electronics, 16: 99-103 (April 1978).
17. 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, 30: 765-768 (October 1983).
18. Willsky, Alan S. et al. "Dynamic Model-Based Techniques for the Detection of Incidents on Freeways," IEEE Transactions on Automatic Control, 25: 347-359 (June 1980).
19. Hostetler, Larry D. and Ronald D. Andreas. "Nonlinear Kalman Filtering Techniques for Terrain-Aided Navigation," IEEE Transactions on Automatic Control, 28: 315-323 (March 1983).
20. Mealy, Gregory L. and Wang Tang. "Application of Multiple Model Estimation to a Recursive Terrain Height Correlation System," IEEE Transactions on Automatic Control, 28: 323-331 (March 1983).
21. Lamb, P. R. and L. C. Westphal. "Simplex-directed Partitioned Adaptive Filters," International Journal of Control, 30: 617-627 (1979).
22. Fry, C. M. and A. P. Sage. "On Hierarchical Structure Adaptation and Systems Identification," International Journal of Control, 20: 433-452 (1974).
23. 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, September 1982.
24. Meer, D. E. and P. S. Maybeck. "Multiple Model Adaptive Estimation for Space-Time Point Process Observations," Proceedings of the IEEE Conference on Decision and Control, Las Vegas, Nevada, Dec 1984.

VITA

Lieutenant Karl P. Hentz was born on September 7, 1956 in Lowell, Massachusetts. He graduated from high school in Northampton, Massachusetts in 1974. After serving two years in the USAF as an Airman, he was discharged to attend the University of Massachusetts at Amherst. He received the degree of Bachelor of Science in Electrical Engineering and a commission in the USAF through ROTC in May 1981. He was assigned to Electronic Systems Division, Deputy for Development Plans at Hanscom AFB, Massachusetts until entering the Air Force Institute of Technology, in June 1983.

Permanent address: 118 Spruce Hill Avenue
Florence, Massachusetts 01060

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b. RESTRICTIVE MARKINGS		
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE					
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GE/ENG/84D-32			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 of 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) Sec Box 19			PROGRAM ELEMENT NO.		PROJECT NO.
			TASK NO.		WORK UNIT NO.
12. PERSONAL AUTHOR(S) Karl P. Hentz, B. S. E. E., 1st Lt, USAF					
13a. TYPE OF REPORT MS Thesis		13b. TIME COVERED FROM _____ TO _____		14. DATE OF REPORT (Yr., Mo., Day) 1984 December	
15. PAGE COUNT 358					
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB GR	Adaptive Control Systems, Adaptive Filters, Multiple Model Adaptive Estimation, Multiple Model Adaptive Control.		
02	05				
19. ABSTRACT (Continue on reverse if necessary and identify by block number)					
Title: FEASIBILITY ANALYSIS OF MOVING BANK MULTIPLE MODEL ADAPTIVE ESTIMATION AND CONTROL ALGORITHMS Thesis Chairman: Peter S. Maybeck, Prof. of Electrical Engineering Approved for public release: IAW AFIT POLYGRAPHIC ANALYSIS DATE 12-1-84 BY AFIT/GE/ENG/84D-32 AFIT/GE/ENG/84D-32 (AIC) Wright-Patterson AFB OH 45433					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☒ SAME AS APT. ☐ DTIC USERS ☐			21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a. NAME OF RESPONSIBLE INDIVIDUAL Peter S. Maybeck, Prof. of Elec. Engineering			22b. TELEPHONE NUMBER (Include Area Code) (513) 255-2057		22c. OFFICE SYMBOL AFIT/ENG

This investigation examines the feasibility of a moving bank multiple model adaptive estimation/control algorithm. Sliding bank multiple model adaptive estimation differs from conventional multiple model adaptive estimation in that a substantially reduced number of elemental filters is required for the sliding bank estimator (9 elemental filters 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 sliding bank of filters is moved about the parameter space in search of the true parameter point.

Critical to the performance of the sliding bank estimator is the decision method that governs movement of the bank of elemental filters. Because of this, a number of different decision algorithms are discussed and their respective performance compared. Three controller designs are also examined: a single changeable-gain, a single fixed-gain, and a sliding bank multiple model adaptive controller.

States of a damped second order system, with uncertain parameters (damping ratio and undamped natural frequency) are estimated by the sliding bank estimator and then regulated to the quiescent state by the controller. Performance of the sliding bank estimator/controller is compared to a benchmark of a single Kalman filter/LQ controller that has (artificial) knowledge of the true parameter values. Comparisons are based upon Monte Carlo analysis results.*

The results demonstrate that the sliding bank multiple model adaptive estimator/controller is a possible alternative to the conventional full bank multiple model adaptive estimator/controller. For the system modeled in this study, performance of the sliding bank estimator/controller is essentially identical to that of the benchmark after a short parameter acquisition period.

END

FILMED

5-85

DTIC