

AD-A150 770

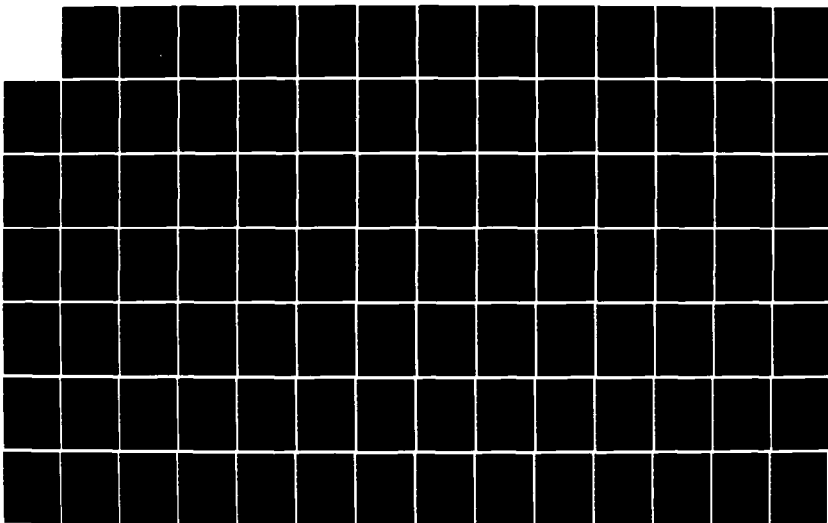
DIGITAL COMPUTER APPLICATIONS OF KALMAN FILTER IN
TARGET TRACKING(U) NAVAL POSTGRADUATE SCHOOL MONTEREY
CA V I HARTZOUROS JUN 84

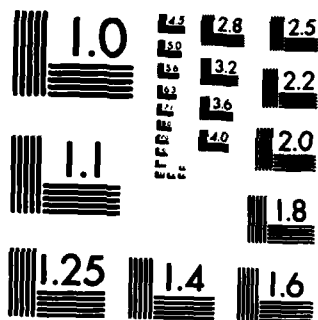
1/3

UNCLASSIFIED

F/G 9/2

NL





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

AD-A150 770

2

NAVAL POSTGRADUATE SCHOOL
Monterey, California



THESIS

DIGITAL COMPUTER APPLICATIONS OF
KALMAN FILTER IN TARGET TRACKING

by

Vasilios I. Martzoucos

June 1984

Thesis Advisor:

H. H. Loomis

Approved for public release; distribution unlimited.

DTIC FILE COPY



85 02 19 074

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
	AD-A150 770	
4. TITLE (and Subtitle)		5. TYPE OF REPORT & PERIOD COVERED
Digital Computer Applications of Kalman Filter in Target Tracking		Master's Thesis; June 1984
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s)		8. CONTRACT OR GRANT NUMBER(s)
Vasilios I. Martzoucos		
9. PERFORMING ORGANIZATION NAME AND ADDRESS		10. PROGRAM ELEMENT PROJECT, TASK AREA & WORK UNIT NUMBERS
Naval Postgraduate School Monterey, California 93943		
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE
Naval Postgraduate School Monterey, California 93943		June 1984
		13. NUMBER OF PAGES
		192
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS (of this report)
		Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)		
Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
Target Tracking Kalman Filter Monte Carlo Simulation		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)		
This paper analyzes the problem of tracking targets in noisy conditions. First a basic background is provided. That includes general concepts from estimation theory and a specific description of the Kalman filter and its use for treating the various aspects of the target tracking problem.		

DD FORM 1473

JAN 73

EDITION OF 1 NOV 65 IS OBSOLETE

S N 0102-LF-014-6601

1

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

#20 - ABSTRACT - (CONTINUED)

Then progressively more difficult situations of target tracking examples are simulated and the results are analyzed and compared with the literature.

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-2	



S/N 0102- LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Approved for public release; distribution unlimited.

Digital Computer Applications of
Kalman Filter in Target Tracking

by

Vasilios I. Martzoucos
Lieutenant, Hellenic Navy
B.S., Naval Academy of Greece, 1975

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ENGINEERING SCIENCE

from the

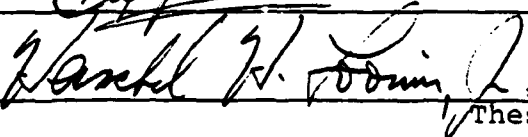
NAVAL POSTGRADUATE SCHOOL

June 1984

Author:



Approved by:



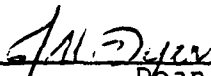
Thesis Advisor



Second Reader



Chairman, Department of Electrical Engineering



Dean of Science and Engineering

ABSTRACT

This paper analyzes the problem of tracking targets in noisy conditions.

First a basic background is provided. That includes general concepts from estimation theory and a specific description of the Kalman filter and its use for treating the various aspects of the target tracking problem.

Then progressively more difficult situations of target tracking examples are simulated and the results are analyzed and compared with the literature.

TABLE OF CONTENTS

I.	INTRODUCTION -----	14
II.	SOME DEFINITIONS AND BASIC CONCEPTS FROM ESTIMATION THEORY -----	19
	A. DEFINITIONS -----	19
	B. BASIC CONCEPTS IN ESTIMATION -----	22
III.	KALMAN FILTER AND APPLICATION TO TARGET TRACKING -----	29
	A. BASICS ON KALMAN FILTER -----	29
	B. DISCRETE KALMAN FILTER -----	30
	C. CONTINUOUS KALMAN FILTER -----	35
	D. EXTENDED KALMAN FILTER -----	39
IV.	ERROR COVARIANCE MATRIX AND TARGET TRACKING QUALITY -----	48
	A. DEFINITION OF ERROR COVARIANCE MATRIX -----	48
	B. INFORMATION CONTAINED IN THE ERROR COVARIANCE MATRIX -----	48
	C. ERROR ELLIPSOIDS ABOUT A TARGET POSITION -----	49
V.	MANEUVERING TARGETS -----	55
	A. GENERAL DESCRIPTION OF THE PROBLEM -----	55
	B. THE WHITE NOISE MODEL WITH ADJUSTABLE LEVEL ---	56
VI.	CHARACTERISTIC COMPUTER EXAMPLES -----	58
	A. A LINEAR KALMAN FILTER PROBLEM -----	58
	B. A PROBLEM USING E.K.F. (NON-LINEAR CASE) -----	94
	C. MANEUVERING TARGET PROBLEM -----	100
VII.	CONCLUSIONS -----	175
	APPENDIX A: COMPUTER ALGORITHMS -----	178

LIST OF REFERENCES -----	189
BIBLIOGRAPHY -----	190
INITIAL DISTRIBUTION LIST -----	191

LIST OF TABLES

I.	Probabilities for Error Ellipsoids -----	53
II.	Observed and Estimated Trajectories (50 Monte Carlo Runs) -----	109
III.	Observed and Estimated Trajectories (Single Runs) -----	110
IV.	Root Mean Square Position Error (50 Monte Carlo Runs) -----	112
V.	Conclusions from Tables II, III, IV -----	113
VI.	Multiple Process Noise Levels Results -----	151

LIST OF FIGURES

1.1	Basic Components of a Tracking System -----	16
3.1	Schematic of the K.F. Algorithm -----	36
3.2	E.K.F. Block Diagram of the System and Estimator --	44
4.1	Error Ellipse -----	51
6.1	True Trajectory in x_1, x_2 Plane. Case 1(a) -----	66
6.2	Estimated Trajectory in x_1, x_2 Plane. Case 1(a) ---	67
6.3	Position Error Variance. Case 1(b) -----	68
6.4	Normalized Position Error. Case 1(c) -----	69
6.5	Velocity Error Variance. Case 1(d) -----	70
6.6	Normalized Velocity Error. Case 1(e) -----	71
6.7	Normalized State Error Squared. Case 1(f) -----	72
6.8	Normalized Innovation Error. Case 1(g) -----	73
6.9	True Trajectory in x_1, x_2 Plane. Case 2(a) -----	74
6.10	Estimated Trajectory in x_1, x_2 Plane. Case 2(a) ---	75
6.11	Position Error Variance. Case 2(b) -----	76
6.12	Normalized Position Error. Case 2(c) -----	77
6.13	Velocity Error Variance. Case 2(d) -----	78
6.14	Normalized Velocity Error. Case 2(e) -----	79
6.15	Normalized State Error Squared. Case 2(f) -----	80
6.16	Normalized Innovation Error. Case 2(g) -----	81
6.17	True Trajectory in x_1, x_2 Plane. Case 3(a) -----	82
6.18	Estimated Trajectory in x_1, x_2 Plane. Case 3(a) ---	83
6.19	Position Error Variance. Case 3(b) -----	84

6.20	Normalized Position Error. Case 3(c) -----	85
6.21	Velocity Error Variance. Case 3(d) -----	86
6.22	Normalized Velocity Error. Case 3(e) -----	87
6.23	Normalized State Error Squared. Case 3(f) -----	88
6.24	Normalized Innovation Error. Case 3(g) -----	89
6.25	Descriptive Diagram for the Non-linear Problem ----	95
6.26	Movement of the Maneuvering Target -----	105
6.27	Obs. and Estim. Traj. Non-adjust. Filter 50 Monte Carlo Runs -----	114
6.28	Obs. and Estim. Traj. Non-adjust. Filter Single Run of the Program -----	115
6.29	Observed and Estim. Traject. $Q' = 100$ TR = .1 50 Monte Carlo Runs -----	116
6.30	Observed and Estim. Traject. $Q' = 100$ TR = 3 50 Monte Carlo Runs -----	117
6.31	Observed and Estim. Traject. $Q' = 100$ TR = 20 50 Monte Carlo Runs -----	118
6.32	Observed and Estim. Traject. $Q' = .1$ TR = .1 Single Run of the Program -----	119
6.33	Observed and Estim. Traject. $Q' = 3$ TR = .1 Single Run of the Program -----	120
6.34	Observed and Estim. Traject. $Q' = 10$ TR = .1 Single Run of the Program -----	121
6.35	Observed and Estim. Traject. $Q' = 100$ TR = .1 Single Run of the Program -----	122
6.36	Observed and Estim. Traject. $Q' = 1000$ TR = .1 Single Run of the Program -----	123
6.37	Observed and Estim. Traject. $Q' = .1$ TR = 3 Single Run of the Program -----	124
6.38	Observed and Estim. Traject. $Q' = 3$ TR = 3 Single Run of the Program -----	125
6.39	Observed and Estim. Traject. $Q' = 10$ TR = 3 Single Run of the Program -----	126

6.40	Observed and Estim. Traject. $Q' = 100$ TR = 3 Single Run of the Program -----	127
6.41	Observed and Estim. Traject. $Q' = 1000$ TR = 3 Single Run of the Program -----	128
6.42	Observed and Estim. Traject. $Q' = .1$ TR = 20 Single Run of the Program -----	129
6.43	Observed and Estim. Traject. $Q' = 3$ TR = 20 Single Run of the Program -----	130
6.44	Observed and Estim. Traject. $Q' = 10$ TR = 20 Single Run of the Program -----	131
6.45	Observed and Estim. Traject. $Q' = 100$ TR = 20 Single Run of the Program -----	132
6.46	Observed and Estim. Traject. $Q' = 1000$ TR = 20 Single Run of the Program -----	133
6.47	Mean Square Error of Pos. $Q' = .1$ TR = .1 -----	134
6.48	Mean Square Error of Pos. $Q' = 4$ TR = .1 -----	135
6.49	Mean Square Error of Pos. $Q' = 10$ TR = .1 -----	136
6.50	Mean Square Error of Pos. $Q' = 100$ TR = .1 -----	137
6.51	Mean Square Error of Pos. $Q' = 1000$ TR = .1 -----	138
6.52	Mean Square Error of Pos. $Q' = .1$ TR = 3 -----	139
6.53	Mean Square Error of Pos. $Q' = 4$ TR = 3 -----	140
6.54	Mean Square Error of Pos. $Q' = 10$ TR = 3 -----	141
6.55	Mean Square Error of Pos. $Q' = 100$ TR = 3 -----	142
6.56	Mean Square Error of Pos. $Q' = 1000$ TR = 3 -----	143
6.57	Mean Square Error of Pos. $Q' = 3000$ TR = 3 -----	144
6.58	Mean Square Error of Pos. $Q' = .1$ TR = 20 -----	145
6.59	Mean Square Error of Pos. $Q' = 4$ TR = 20 -----	146
6.60	Mean Square Error of Pos. $Q' = 10$ TR = 20 -----	147
6.61	Mean Square Error of Pos. $Q' = 100$ TR = 20 -----	148

6.62	Mean Square Error of Pos. Q' = 1000 TR = 20 -----	149
6.63	Mean Square Position Error; Many Q' Levels; Q' Levels: 0,.1,100 -----	152
6.64	Mean Square Position Error; Many Q' Levels; Q' Levels: 0, 3,100 -----	153
6.65	Mean Square Position Error; Many Q' Levels; Q' Levels: 0,10,100 -----	154
6.66	Mean Square Position Error; Many Q' Levels; Q' Levels: 0,.1,10 -----	155
6.67	Mean Square Position Error; Many Q' Levels; Q' Levels: 0,1,10 -----	156
6.68	Mean Square Position Error; Many Q' Levels; Q' Levels: 0,3,10 -----	157
6.69	Mean Square Position Error; Many Q' Levels; Q' Levels: 0,.1,1000 -----	158
6.70	Mean Square Position Error; Many Q' Levels; Q' Levels: 0,1,1000 -----	159
6.71	Mean Square Position Error; Many Q' Levels; Q' Levels: 0,3,1000 -----	160
6.72	Mean Square Position Error; Many Q' Levels; Q' Levels: 0,10,1000 -----	161
6.73	Mean Square Velocity Error; Many Q' Levels; Q' Levels: 0,.1,100 -----	162
6.74	Mean Square Velocity Error; Many Q' Levels; Q' Levels: 0,3,100 -----	163
6.75	Mean Square Velocity Error; Many Q' Levels; Q' Levels: 0,10,100 -----	164
6.76	Mean Square Velocity Error; Many Q' Levels; Q' Levels: 0,.1,10 -----	165
6.77	Mean Square Velocity Error; Many Q' Levels; Q' Levels: 0,1,10 -----	166
6.78	Mean Square Velocity Error; Many Q' Levels; Q' Levels: 0,3,10 -----	167
6.79	Mean Square Velocity Error; Many Q' Levels; Q' Levels: 0,1,1000 -----	168

6.80	Mean Square Velocity Error; Many Q' Levels; Q' Levels: 0,1,1000 -----	169
6.81	Mean Square Velocity Error; Many Q' Levels; Q' Levels: 0,3,1000 -----	170
6.82	Mean Square Velocity Error; Many Q' Levels; Q' Levels: 0,10,1000 -----	171
6.83	References 9 and 10 Results of Maneuvering Target Example -----	173

ACKNOWLEDGMENTS

I would like to thank Professor H. Loomis for his guidance, direction and help in preparing this thesis.

I must also thank my wife, Maria, for her patience and encouragement. Her support greatly facilitated the completion of this work.

I. INTRODUCTION

The tracking problem today is of great interest particularly in the military area. The term "tracking" is considered to be the processing of measurements obtained from a target in order to maintain an estimate of its current state. The latter usually contains:

- a. Kinematic components (position, velocity, acceleration, etc.).
- b. Other components (radiated strength of signal, spectral characteristics, etc.).
- c. Constant or slowly-varying parameters (propagation velocity, coupling coefficients, etc.).

A track is a state trajectory estimated from a set of measurements which are associated with the same target.

Measurements are noise-corrupted observations related to the state of the target. For example a measurement can be direct estimate of position, range, bearing, time difference of arrival, frequency of narrow band signal, or frequency difference of arrival.

Many approaches have been developed for the solution of the tracking problem. Many of them are very reliable and work with satisfactory results. The problem however is complicated, so there is always need for further improvement of the tracking procedures.

The complexity of the tracking problem is due to the following main reasons:

- a. Presence of countermeasures (evasive maneuvers, decoys, etc.).
- b. Need of advanced information processing techniques for manipulation of the numerous and complex data to be possible.
- c. Uncertainty associated with the measurements related to the origin of the measurements, i.e., a measurement may not have originated from the target of interest.
- d. Inaccurate measurements because of random noise.
- e. Random false alarms in the detection process.
- f. Clutter because of reflectors or radiators near the target of interest.
- g. Interfering targets.

The basic components of a tracking system are shown in Figure 1.1.

The term "filter" is often applied to devices or systems which process incoming signals or other data in such a way as to eliminate noise, to smooth signals, or to predict the input signal from instant to instant. There is much literature covering the theories of estimation, identification, modeling, prediction, etc. The design of such filters falls in the domain of optimal filtering, which originated with the work of Wiener [Ref. 1] and was extended by the work of Kalman-Busy [Ref. 2] and others. It was only around 1970

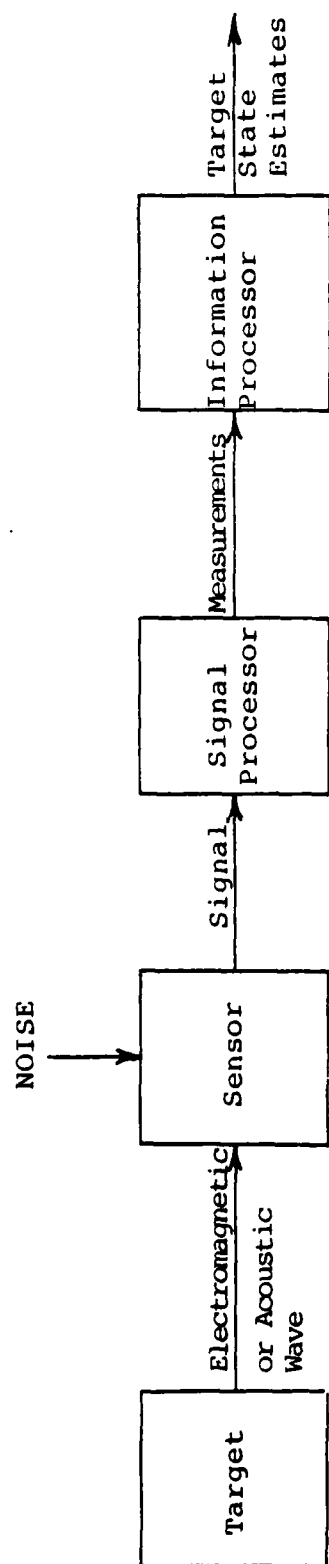


Figure 1.1 Basic Components of a Tracking System

that the systematic treatment of tracking multiple targets in the presence of false alarms using Kalman filtering techniques has started with the work of [Ref. 3] and [Ref. 4].

It has been generally accepted that the Kalman filtering technique is one of the most desirable of existing tracking algorithms. The reason is that it is able to accept many inputs (bearing, range, course, speed, doppler, etc.). The inputs can be available from a variety of locations and sensors. The Kalman filter then weights them properly with respect to their errors and generates an estimate of target position, course, and speed. It contains also information within its structure sufficient to give the user or decision maker a useful indicator of the estimate's quality.

The purpose of this paper is to describe the target tracking problem under noisy conditions. The Kalman filter is used as the basic tool to treat this problem. Some characteristic problems are presented and solved.

In Chapter II some basic concepts and definitions are provided from estimation theory. The contents of this chapter form the theoretical background needed for the rest of the chapters. It is assumed that the reader is familiar with basic probability theory.

Chapter III contains a general description of the Kalman filter (discrete and continuous) and the nonlinear case where a linearization of target tracking problem takes place for the application of Kalman filter to be possible (Extended Kalman Filter).

In Chapter IV, a brief analysis of the error covariance matrix and the information contained in it has been made. Also its use in setting error ellipsoids (confidence regions) about an estimated target position has been described.

In Chapter V the maneuvering target problem is described and one of the various existing approaches is analyzed, named "The White Noise Model with Adjustable Level." The same method is used later in Chapter VI, in a relative example. Chapter VI contains three characteristic examples covering the application of Kalman filter in some individual cases of target tracking problems. Specifically the presented examples are the following:

- a. Linear target problem (measurements of target's position available).
- b. Non-linear target problem (measurements of target's bearing available).
- c. Linear maneuvering target's problem (measurements of position available). The target maneuvers two times with different acceleration each time.

The relative calculations are provided analytically together with the corresponding computer programs. The resulting plots are also provided with proper comments.

II. SOME DEFINITIONS AND BASIC CONCEPTS FROM ESTIMATION THEORY

The following definitions and concepts from probability and estimation theory respectively are described in detail in Reference 5 and Reference 6.

A. DEFINITIONS

1. Bayes' Rule for Random Variables

For random variables Bayes' rule is given by:

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)} = \frac{P(y|x)P(x)}{\int P(y|x)P(x)dx} \quad (2.1)$$

The conditional probability of an event is sometimes referred to as "posterior" while the unconditional one as "prior." In this case $p(x)$ is called the "prior" probability density function (p.d.f.) and $p(x|y)$ is the posterior one.

2. Gaussian Random Variables

The p.d.f. of a Gaussian random variable is

$$p(x) = N(x; \bar{x}, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(x - \bar{x})^2}{2\sigma^2}\right\} \quad (2.2)$$

where $N(\)$ represents the normal density with argument x , mean $\bar{x}$ and variance σ .

An equivalent expression is

$$x \sim N(\bar{x}, \sigma^2) \quad (2.3)$$

where x is normally distributed with respective mean and variance.

3. White Noise

A discrete Gaussian random process v is called "white noise" if for any k time points $t_1, t_2, \dots, t_k$, the random vectors $v(t_1), \dots, v(t_k)$ (which are Gaussian) are independent. The autocorrelation of this random process is zero for any two different times.

A discrete Gaussian random process is often used as a model for measurement noise and disturbance noise.

4. Markov Processes

The common property of Markov processes is the following:

$$p[x(t) | x(\tau), \tau \leq t_1] = p[x(t) | x(t_1)] \quad \forall t > t_1 \quad (2.4)$$

i.e., the past up to t , is completely described by the value of the process at t .

The state of a dynamic system with input a white noise

$$\dot{x}(t) = f[x(t), n(t)] \quad (2.5)$$

is a Markov process.

5. Random Sequences

A random sequence, or a discrete time stochastic process, is a time-indexed sequence of random variables.

$$x^k = \{x(j), j = 1, \dots, k\} \quad k = 1, \dots \quad (2.6)$$

A random sequence is Markov if

$$p[x(k)|x^j] = p[x(k)|x(j)] \quad \forall k > j \quad (2.7)$$

The sequence $v(j)$, $j = 1, \dots$, is a (discrete time) white noise if

$$E[v(k)v(j)] = q\delta_{kj} \quad (2.8)$$

where δ_{kj} is the Kronecker delta function:

$$\delta_{kj} = \begin{cases} 1 & \text{if } k = j \\ 0 & \text{if } k \neq j \end{cases} \quad (2.9)$$

The state of a dynamic system excited by white noise

$$\underline{x}(k+1) = f[k, x(k), v(k)] \quad (2.10)$$

is a discrete-time Markov process or Markov sequence.

A stochastic process $x(t)$, $t \in I$ is a Gaussian white (or independent) process if for any m time points $t_1, \dots, t_m$ in I ($m = \text{integer}$), the m random n vectors $x(t_1), \dots, x(t_m)$ are independent Gaussian random vectors.

The state of a linear dynamic system excited by white Gaussian noise

$$\underline{x}(k+1) = \underline{F} \underline{x}(k) + \underline{v}(k) \quad (2.11)$$

is a Gauss-Markov process.

B. BASIC CONCEPTS IN ESTIMATION

1. Estimating Problem--Random and Non-random Parameters

The problem of estimating a (time-invariant) parameter x is:

Given the measurements

$$z(j) = h[j, x, w(j)] , \quad j = 1, \dots \quad (2.12)$$

made in the presence of random noises $w(j)$, find a function

$$\hat{x}(k) = \hat{x}(k, z^k) \quad (2.13)$$

where

$$z^k = \{z(j), j = 1, \dots, k\} \quad (2.14)$$

that estimates the value of x in some sense.

The function (2.13) is called an estimator and its value is called the estimate. We can use two models for a time-invariant parameter:

a. Non-random: We have an unknown true value x_0 . Here it is desired that the estimates converge in some sense to this true value as $k \rightarrow \infty$ (non-Bayesian approach).

b. Random: The parameter is a random variable with a prior p.d.f. $p(x)$. In this case a realization of x according to $p(x)$ is assumed to have taken place; It is desired for each measurement sequence to yield an estimate that converges to the corresponding realization of x and this should hold for all x (Bayesian approach).

In the second approach given the prior p.d.f. of the parameter, its posterior can be obtained using the Bay's rule.

In the first case the likelihood function

$$\Lambda_k(x) = p(z^k|x) \quad (2.15)$$

is used as a measure of how "likely" a parameter value is for the observations that are made.

A usual method of estimation of non-random parameters is the maximum likelihood method. Its maximum likelihood estimate (MLE) is

$$\hat{x}^{ML}(k) = \arg \max_x p(z^k|x) \quad (2.16)$$

The corresponding estimate for a random parameter is the maximum a posteriori (MAP) estimate

$$\hat{x}^{MAP}(k) = \arg \max_x p(x|z^k) = \arg \max_x [p(z^k|x)p(x)] \quad (2.17)$$

The $\hat{x}^{\text{MAP}}$ will coincide with $\hat{x}^{\text{ML}}$ for a certain prior p.d.f., called "diffuse."

2. Linear Estimation--Static Case

The minimum mean square error (m.m.s.e.) estimate of a random variable x in terms of another random variable y is the conditional mean $E[x|y]$. In practice the evaluation of the conditional mean is complicated so usually the "linear m.m.s.e. estimation" method is applied. According to this method the best estimate is such that it is unbiased and the estimation error is uncorrelated from the observable(s), i.e., they are orthogonal.

The estimation of two random vectors x and z jointly normally (Gaussian) distributed is now examined:

Let

$$Y = \begin{bmatrix} \underline{x} \\ \underline{z} \end{bmatrix} \quad (2.18)$$

The random variable y is normally distributed with mean

$$\bar{Y} = \begin{bmatrix} \underline{\bar{x}} \\ \underline{\bar{z}} \end{bmatrix} \quad (2.19)$$

and covariance matrix (assumed nonsingular)

$$\underline{P}_{\underline{y}\underline{y}} = \begin{bmatrix} \underline{P}_{\underline{x}\underline{x}} & \underline{P}_{\underline{x}\underline{z}} \\ \underline{P}_{\underline{z}\underline{x}} & \underline{P}_{\underline{z}\underline{z}} \end{bmatrix} \quad (2.20)$$

where

$$\underline{P}_{\underline{x}\underline{x}} = E[(\underline{x} - \bar{\underline{x}})(\underline{x} - \bar{\underline{x}})'] \quad (2.21)$$

and

$$\underline{P}_{\underline{x}\underline{z}} = E[(\underline{x} - \bar{\underline{x}})(\underline{z} - \bar{\underline{z}})'] \quad (2.22)$$

Then the m.m.s.e. estimate of x in terms of z is

$$\hat{\underline{x}} \triangleq E[\underline{x}|\underline{z}] = \bar{\underline{x}} + \underline{P}_{\underline{x}\underline{z}} \underline{P}_{\underline{z}\underline{z}}^{-1} (\underline{z} - \bar{\underline{z}}) \quad (2.23)$$

and the corresponding covariance matrix is

$$\underline{P}_{\underline{x}\underline{x}|\underline{z}} \triangleq E[(\underline{x} - \hat{\underline{x}})(\underline{x} - \hat{\underline{x}})'|\underline{z}] = \underline{P}_{\underline{x}\underline{x}} - \underline{P}_{\underline{x}\underline{z}} \underline{P}_{\underline{z}\underline{z}}^{-1} \underline{P}_{\underline{z}\underline{x}} \quad (2.24)$$

We can see that the optimal estimate is a linear function of z (this is because of the Gaussian assumption), while the covariance which measures the "quality" of the estimate is independent of the observation $\underline{z}$.

If the random variables are not Gaussian, then the "best linear" estimate of x in terms of z (lin. m.m.s.e.) is:

$$\hat{\underline{x}} = \underline{\bar{x}} + \underline{P}_{xz} \underline{P}_{zz}^{-1} (\underline{z} - \underline{\bar{z}}) \quad (2.25)$$

The m.s.e. matrix corresponding to Equation (2.25) is given by

$$E[\hat{\underline{x}}\hat{\underline{x}}'] = \underline{P}_{xx} - \underline{P}_{xz} \underline{P}_{zz}^{-1} \underline{P}_{zx} = \underline{P}_{xx,z} \quad (2.26)$$

From the above results it follows that the best estimator (in the m.m.s.e. sense) for Gaussian random variables is the same as the best linear estimator for arbitrarily distributed random variables with the same first and second moments.

3. Estimating with the "Least Squares" Method

The least squares method is another common estimation procedure for non-random parameters. For measurements

$$z(j) = h(j,x) + w(j) \quad (2.27)$$

The least square estimate of x is

$$\hat{x}^{LS}(k) = \arg \min_x \sum_{j=1}^k [z(j) - h(j,x)]^2 \quad (2.28)$$

Equation (2.28) does not make assumptions about the noises $w(j)$. If these noises are independent identical distributed (i.i.d.) zero-mean Gaussian random variables, then Equation (2.28) coincides with the MLE for these assumptions.

The corresponding case for random parameters is the minimum mean square error (m.m.s.e.) estimate

$$\hat{x}^{\text{MMSE}}_{(k)} = \arg \min_{\hat{x}} E[(\hat{x} - x)^2 | z^k] \quad (2.29)$$

The solution to the above is

$$\hat{x}^{\text{MMSE}}_{(k)} = E[x | z^k] = \int x p(x | z^k) dx \quad (2.30)$$

where the expectation is with respect to the conditional p.d.f.:

$$P(x | z^k) = \frac{P(z^k | x) P(x)}{P(z^k)} \quad (2.31)$$

4. Consistent Estimators

For a non-random parameter case, the estimator is called consistent if the estimate converges to the true value in some stochastic sense. Using the convergence in mean square,

$$\lim_{k \rightarrow \infty} E\{[\hat{x}(k) - x_0]^2\} = 0 \quad (2.32)$$

The expectation is

$$E[\hat{x}(z^k)] = x_0 \quad (2.33)$$

Requirement for the convergence of the estimator
in the random parameter case (m.s. sense) is

$$\lim_{k \rightarrow \infty} E\{[\hat{x}(k) - x]^2\} = 0 \quad (2.34)$$

The expectation here is over z^k and x .

III. KALMAN FILTER AND APPLICATION TO TARGET TRACKING

A. BASICS ON KALMAN FILTER

Kalman filtering technique is very popular in target tracking applications. It is basically a method of solving the problems of optimal filtering and prediction, which according to Reference 1 is defined as:

- a. Optimal filtering is the estimation of state vector $\underline{X}(t)$ from data $\underline{Z}(\tau)$ where $\tau \leq t$.
- b. Prediction is the estimation of a state vector $\underline{X}(t)$ at time t from data $\underline{Z}(\tau)$, where $\tau < t$.

In target tracking both the above problems are encountered. The Kalman filter is desirable because as an estimation model it has the following features:

- a. At time t , the filter generates an unbiased estimate $\hat{\underline{X}}(t)$ of the state vector $\underline{X}(t)$. That means that the expected value of the estimate is the value of the state vector at time t .
- b. The estimate is a minimum variance estimate (i.e., it has the property that its error covariance is less than or equal to that of any other linear unbiased estimate).
- c. The filter is recursive and it does not store past data.

- d. The filter is linear which simplifies calculations and lends itself to machine computations. Gelb [Ref. 3] discusses the above features.

B. DISCRETE KALMAN FILTER

Assuming that we are dealing with a discrete time system of the form:

$$\underline{X}(k+1) = \underline{\phi}(k+1,k) \underline{X}(k) + \underline{\Gamma}(k+1,k) \underline{u}(k) + \underline{W}(k) \quad (3.1)$$

$$\underline{Z}(k) = \underline{C}(k) \underline{X}(k) + \underline{V}(k) \quad (3.2)$$

the following declaration must be done:

" $\hat{\underline{X}}(j|j)$ " is read as: "X hat of j given j," which means that "The estimate of $\underline{X}$ at time j given measurements at times up to and including time j."

Then there are eight main components that make up the Kalman filter:

- a. $\underline{X}(k)$ is the state vector at time k, and $\hat{\underline{X}}(k)$ is the state vector estimate which is an unbiased minimum variance estimate of the true state vector $\underline{X}(k)$ at time k.
- b. The error covariance matrix $\underline{P}(k)$ is a matrix representing the covariance of the difference between the true state vector $\underline{X}(k)$ and the estimate $\hat{\underline{X}}(k)$, and can be expressed as:

$$\underline{P}(k) = E\{(\underline{X}(k) - \hat{\underline{X}}(k))(\underline{X}(k) - \hat{\underline{X}}(k))'\} \quad (3.3)$$

- c. The transition matrix $\underline{p}(k)$ is used in the calculation of the state vector estimate at the present point, in time $(k+1|k)$, from the state vector estimate of a past point in time $(k|k)$. It is also used in the calculation of the error covariance matrix at time $(k+1|k)$, using the error covariance matrix at time $(k|k)$.
- d. The measurement vector $\underline{Z}(k)$, is the data sample or observation taken at time k and its components are linear combinations of the components of the state vector $\underline{X}(k)$ which have been corrupted by uncorrelated Gaussian noise $V(k)$ whose mean is zero and has a covariance matrix $\underline{R}(k)$.
- e. The covariance matrix $\underline{R}(k)$ associated with the Gaussian noise that corrupts the observation or measurement at time k , must be provided by the user. It is the covariance of a sensor measurement vector $\underline{Z}(k)$. For example a range sensor may have a measurement variance of 60 yards. A measurement made with that sensor would have a scalar value of 60 yards for $\underline{R}(k)$. As will be seen later, it is important that the value of $\underline{R}(k)$ be accurate as it affects the performance of the filter.
- f. The conversion matrix $\underline{C}(k)$ describes the linear combinations of the components of the state vector $\underline{X}(k)$

which makes the components of the measurement vector $\underline{Z}(k)$. The relationship between the measurement vector $\underline{Z}(k)$, the conversion matrix $\underline{C}(k)$, and the Gaussian noise $\underline{V}(k)$ is:

$$\underline{Z}(k) = \underline{C}(k) \underline{X}(k) + \underline{V}(k) \quad (3.4)$$

- g. The Kalman gain matrix $\underline{G}(k)$ is instrumental in minimizing the difference between the estimate $\hat{\underline{X}}(k)$ and the state vector $\underline{X}(k)$. $\underline{G}(k)$ is chosen to minimize the trace of the estimate error covariance matrix $\underline{P}(k)$ and is used to revise the estimate $\hat{\underline{X}}(k)$ of the state vector $\underline{X}(k)$ and the error covariance matrix $\underline{P}(k)$ after the observation $\underline{Z}(k)$ has been made.
- h. The last component to be discussed is the system dynamics covariance matrix $\underline{Q}(k)$. The assumption of the Kalman model is that the state vector $\underline{X}(t)$ exists in a random environment that is Gaussian with zero mean and covariance $\underline{Q}(k)$. The system dynamics covariance matrix must be input by the user, and its determination is important as it also affects the performance of the filter.

The remainder of this subsection shows the relationships between the algorithm. Two important points follow:

- a. The transition matrix $\underline{\phi}(k)$, measurement noise covariance matrix $\underline{R}(k)$, systems dynamics covariance matrix

$\underline{Q}(k)$ and the conversion matrix $\underline{C}(k)$ may be different for each point in time k .

- b. The filter must be initialized by the user providing the initial estimate $\hat{\underline{X}}(k)$ and its associated estimate error covariance matrix $\underline{P}(k)$. A poor initialization will require more observations for the algorithm estimate to converge near the value of the state vector.

A Kalman filter iteration can be divided into phases: Prediction and Filtering.

During the prediction phase, the state vector estimate $\hat{\underline{X}}(k+1|k)$ and its error covariance matrix $\underline{P}(k+1|k)$ are updated from the previous value at time $(k|k)$ to the time $(k+1|k)$ when the current measurement $\underline{Z}(k)$ is observed.

During the same phase the system dynamics is introduced into the algorithm. The covariance matrix $\underline{Q}(k)$ accounts for the system dynamics (environment) from time (k) to $(k+1|k)$. The update is performed by multiplying the estimate $\hat{\underline{X}}(k|k)$ and the error covariance matrix $\underline{P}(k|k)$ by the transition matrix $\underline{\phi}(k+1,k)$ as follows:

$$\hat{\underline{X}}(k+1|k) = \underline{\phi}(k+1,k) \hat{\underline{X}}(k|k) \quad (3.5)$$

$$\underline{P}(k+1|k) = \underline{\phi}(k+1,k) \underline{P}(k|k) \underline{\phi}'(k+1,k) + \underline{Q}(k) \quad (3.6)$$

If the system has an input u , the first of the above equations is extended as follows:

$$\hat{\underline{X}}(k+1|k) = \underline{\phi}(k+1,k) \underline{X}(k,k) + \underline{\omega}(k+1,k) \underline{u}(k) \quad (3.7)$$

During the filtering phase, the estimate $\hat{\underline{X}}(k,k)$ of the state vector $\underline{X}(k|k)$ and the error covariance matrix $\underline{P}(k|k)$ are revised, based on the latest measurement $\underline{Z}(k)$ observed at time k . To do this, the Kalman gain matrix $\underline{G}(k)$ is first computed using the error covariance matrix $\underline{P}(k|k-1)$ which was updated in the prediction phase. The covariance matrix $\underline{R}(k)$ which is the covariance of the Gaussian noise associated with the latest measurement $\underline{Z}(k)$ and the conversion matrix $\underline{C}(k)$ are also associated with the latest measurement. The sequence of computations during the filtering phase is:

$$\underline{G}(k) = \underline{P}(k|k-1) \underline{C}'(k) [\underline{C}(k) \underline{P}(k|k-1) \underline{C}'(k) + \underline{R}(k)]^{-1} \quad (3.8)$$

$$\hat{\underline{X}}(k|k) = \hat{\underline{X}}(k|k-1) + \underline{G}(k) [\underline{Z}(k) - \underline{C}(k) \hat{\underline{X}}(k|k-1)] \quad (3.9)$$

$$\hat{\underline{P}}(k|k) = [\underline{I} - \underline{G}(k) \underline{C}(k)] \underline{P}(k|k-1) \quad (3.10)$$

The measurement prediction covariance is

$$\underline{S}(k+1) = \underline{C}(k+1) \underline{P}(k+1|k) \underline{C}'(k+1) + \underline{R}(k+1) \quad (3.11)$$

The innovation or measurement residual is

$$\underline{v}(k+1) \triangleq \underline{z}(k+1) - \hat{\underline{z}}(k+1|k) \quad (3.12)$$

An important property of the innovations is that they are an orthogonal sequence, i.e.,

$$E[\underline{v}_k \underline{v}_j^T] = \underline{S}_k \delta_{kj} \quad (3.13)$$

where δ_{kj} is the Kronecker delta function.

It must be noticed that, at every stage k , the entire past is summarized by the "sufficient statistic" $\hat{\underline{x}}(k|k)$ and the associated covariance. The covariance equations are independent of the measurements. The covariance equations can thus be iterated forward offline.

Figure 3.1 shows a schematic of the Kalman filter algorithm and relationships among the components described above. Reference 7 provides a detailed discussion of Kalman filtering and optimal estimation.

C. CONTINUOUS KALMAN FILTER

In order to go from a discrete to a continuous system of the form:

$$\dot{\underline{X}}(t) = \underline{F}(t) \underline{X}(t) + \underline{B}(t) \underline{W}(t) \quad (3.14)$$

$$\underline{Z}(t) = \underline{C}(t) \underline{X}(t) + \underline{V}(t) \quad (3.15)$$

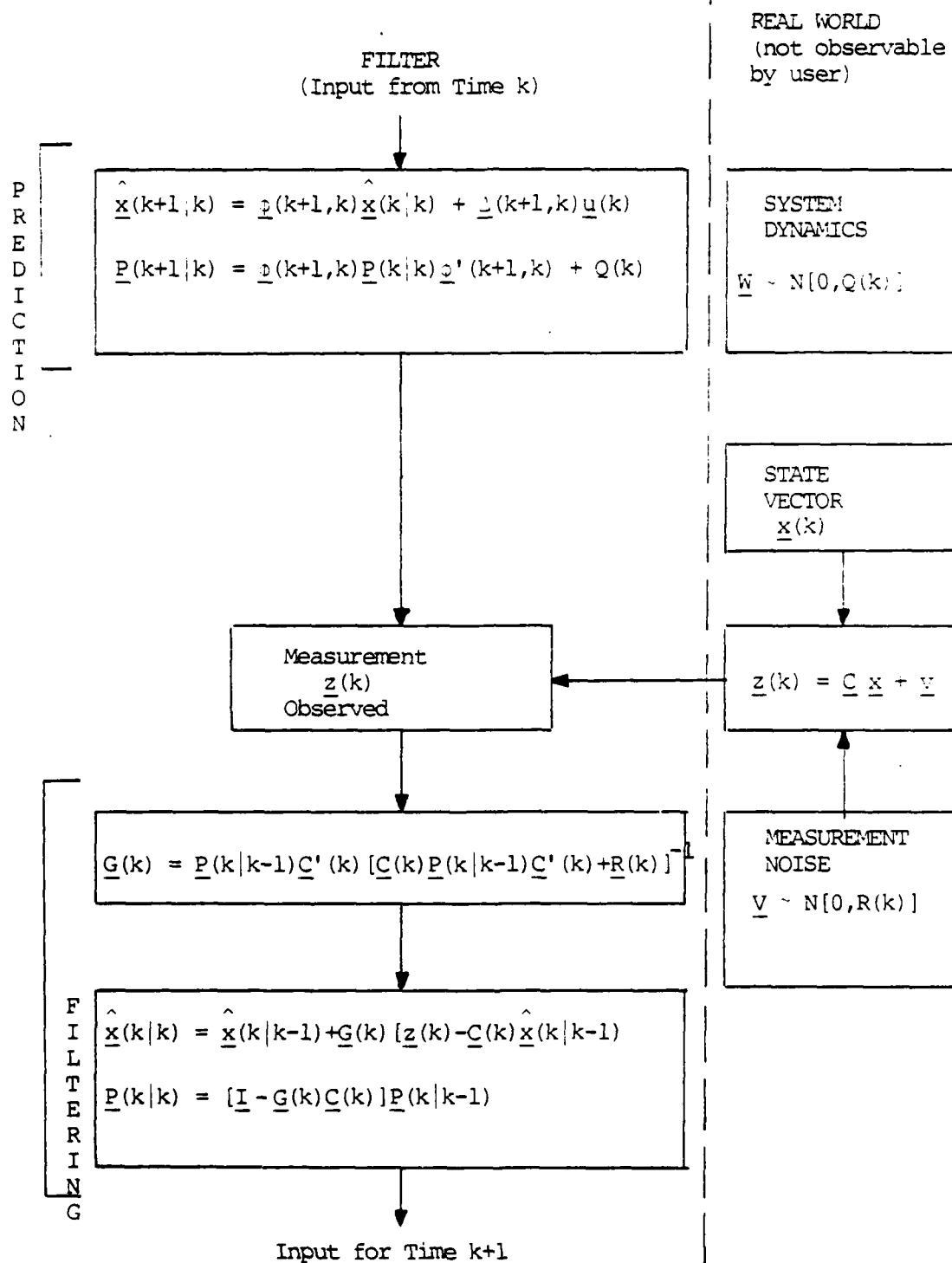


Figure 3.1 Schematic of the K.F. Algorithm

where W, V are zero mean, uncorrelated, white noise processes with covariance matrices Q and R respectively, we have to consider the situation at the limit as $t(k) - t(k-1) = \Delta t \rightarrow 0$. At this limit we have the following equivalences:

$$\Phi(k) \rightarrow I + F \Delta t \quad (3.16)$$

$$Q(k) \rightarrow B Q B' \Delta t \quad (3.17)$$

$$R(k) \rightarrow R/\Delta t \quad (3.18)$$

$R(k) = E(V(k) V'(k))$ is a covariance matrix, while:

$$R(t) = E(V(t) V'(\tau)) = R(t) \delta(t - \tau) \quad (3.19)$$

is a spectral density matrix. The covariance matrix $R(t) \delta(t - \tau)$ has infinite valued elements. To approximate the continuous white noise process by the discrete white noise sequence, the pulse lengths (Δt) may be shrunk and their amplitudes may be increased, such that $R(k) \rightarrow R/\Delta t$. In other words, as $\Delta t \rightarrow 0$, the discrete noise sequence tends to one of an infinite valued pulse of zero duration, such that the area under the "impulse" autocorrelation function is $R(k)\Delta t$, equal to the area R under the continuous white noise impulse autocorrelation function.

With the above expressions in mind, the procedure is to write the proper difference equations and to observe their

behavior in the limit as $\Delta t \rightarrow 0$. It is assumed that R is non-singular, i.e., R^{-1} exists.

Finally the continuous Kalman filter equations are summarized as follows:

a. System Model

$$\dot{\underline{X}}(t) = \underline{F}(t) \underline{X}(t) + \underline{B}(t) \underline{W}(t), \quad \underline{W}(t) \sim N(0, Q(t)) \quad (3.20)$$

b. Measurement Model

$$\underline{Z}(t) = \underline{C}(t) \underline{X}(t) + \underline{V}(t), \quad \underline{V}(t) \sim N(0, R(t)) \quad (3.21)$$

c. Initial Conditions

$$E[\underline{X}(0)] = \hat{\underline{X}}_0, \quad E[(\underline{X}(0) - \hat{\underline{X}}_0)(\underline{X}(0) - \hat{\underline{X}}_0)'] = \underline{P}_0 \quad (3.22)$$

d. Assumptions

$$R^{-1}(t) \text{ exists}$$

e. State Estimate

$$\dot{\hat{\underline{X}}}(t) = \underline{F}(t) \hat{\underline{X}}(t) + \underline{G}(t) [\underline{Z}(t) - \underline{C}(t) \hat{\underline{X}}(t)], \quad \hat{\underline{X}}(0) = \hat{\underline{X}}_0 \quad (3.23)$$

f. Error Covariance Propagation

$$\begin{aligned} \dot{\underline{P}}(t) &= \underline{F}(t) \underline{P}(t) + \underline{P}(t) \underline{F}'(t) + \underline{B}(t) \underline{Q}(t) \underline{B}'(t) - \underline{G}(t) \underline{R}(t) \underline{G}'(t) \\ &\text{where } \underline{P}(0) = \underline{P}_0 \end{aligned} \quad (3.24)$$

g. Kalman Gain Matrix

$$\underline{G}(t) = \underline{P}(t)\underline{C}'(t)\underline{R}^{-1}(t) \quad \text{when} \quad E[\underline{W}(t) \underline{V}'(\tau)] = 0 \quad (3.25)$$

$$\underline{G}(t) = [\underline{P}(t)\underline{C}'(t) + \underline{B}(t)\underline{J}(t)]\underline{R}^{-1}(t) \quad (3.26)$$

For more details, See Reference 7, pp. 119-124.

D. EXTENDED KALMAN FILTER

In target tracking it is usually needed to estimate the present target position, course and speed and to predict future target position based on the present estimate. The Kalman filter works very well for target tracking since the algorithm can provide an unbiased, minimum variance estimate of the target's state based on varied observations (filtering), predict future position using the prediction phase of the filter and provide an indicator of the estimate error through the estimate error covariance matrix.

Usually in practice, the state and/or measurement equations are not linear. Since the Kalman filter is applied to linear cases, it is necessary to find a "method" to use it in nonlinear estimation problems. One approach is to derive an optimal filter for the nonlinear problem. Another approach (more usual) is to linearize the problem and apply Kalman filter theory to the linearized problem. The highlights of the second method are presented in the following:

The system and measurement equations are assumed to be of the form:

Discrete

$$\underline{x}(k+1) = \underline{a}(\underline{x}(k), \underline{u}(k), k) + \underline{w}(k) \quad (3.27)$$

$$\underline{z}(k) = \underline{c}(\underline{x}(k)) + \underline{v}(k) \quad (3.28)$$

Continuous

$$\dot{\underline{x}}(t) = \underline{f}(\underline{x}(t), \underline{u}(t), t) + \underline{w}(t) \quad (3.29)$$

$$\underline{z}(t) = \underline{h}(\underline{x}(t)) + \underline{v}(t) \quad (3.30)$$

Here it is assumed that we deal with a discrete model.

It is assumed that we have available a trajectory $\underline{x}^{(0)}(k)$, $k = 0, 1, 2, \dots$. The vector function $\underline{a}(\underline{x}(k), \underline{u}(k), k)$ is expanded in a Taylor series about the nominal trajectory $\underline{x}^{(0)}(k)$. Then the linearized state equations can be written as:

$$\underline{x}(k+1) = \underline{A}(k) \underline{x}(k) + \underline{a}(\underline{x}^{(0)}(k), \underline{u}(k), k) - \underline{A}(k) \underline{x}^{(0)}(k) + \underline{w}(k) \quad (3.31)$$

where $\underline{A}(k)$ is defined to be the first partial derivatives of the nonlinear function $\underline{a}(\underline{x}^{(0)}(k), \underline{u}(k), k)$, with respect

to the state vector $\underline{x}(k)$. i.e.,

$$\underline{A}(k) = \left. \frac{\partial \underline{a}}{\partial \underline{x}} \right|_{(\underline{x}^{(0)}(k), \underline{u}(k), k)} \quad (3.32)$$

The accuracy of this approximation depends on the difference between $\underline{x}^{(0)}(k)$ and the actual trajectory $\underline{x}(k)$. The middle two terms are treated as deterministic inputs.

Now the measurement equation has been considered. We have

$$\underline{z}(k) = \underline{c}(\underline{x}(k)) + \underline{v}(k) \quad (3.33)$$

The nonlinear vector $\underline{c}$ is expanded again about the nominal trajectory $\underline{x}^{(0)}(k)$. Then the measurement equation can be written as

$$\underline{z}(k) = \underline{H}(k)\underline{x}(k) + \underline{c}(\underline{x}^{(0)}(k)) - \underline{H}(k)\underline{x}^{(0)}(k) + \underline{v}(k) \quad (3.34)$$

where $\underline{H}(k)$ is defined as the first partial derivatives of the measurement function $\underline{c}(\underline{x}^{(0)}(k))$ with respect to the state vector $\underline{x}(k)$, i.e.,

$$\underline{H}(k) \triangleq \left. \frac{\partial \underline{c}}{\partial \underline{x}} \right|_{\underline{x}^{(0)}(k)} \quad (3.35)$$

As in the linearized state equation, the two middle terms are known time-varying quantities which can be handled as if

they were a time-varying measurement bias. For simplification it is defined

$$\underline{u}'(k) \triangleq \underline{a}(\underline{x}^{(0)}(k), \underline{u}(k), k) - \underline{A}(k)\underline{x}^{(0)}(k) \quad (3.36)$$

$$\begin{aligned} \underline{z}'(k) &\triangleq \underline{z}(k) - \underline{c}(\underline{x}^{(0)}(k)) + \underline{H}(k)\underline{x}^{(0)}(k) \\ &= \underline{z}(k) - \underline{b}(k) \end{aligned} \quad (3.37)$$

so that

$$\underline{x}(k+1) = \underline{A}(k)\underline{x}(k) + \underline{u}'(k) + \underline{w}(k) \quad (3.38)$$

$$\underline{z}'(k) = \underline{H}(k)\underline{x}(k) + \underline{v}(k) \quad (3.39)$$

With these linearized equations, the appropriate Kalman filter equations are:

a. The gain equation

$$\underline{G}(k) = \underline{P}(k|k-1)\underline{H}'(k) [\underline{H}(k)\underline{P}(k|k-1)\underline{H}'(k) + \underline{R}(k)]^{-1} \quad (3.40)$$

b. The covariance of estimation error equation

$$\underline{P}(k|k-1) = \underline{A}(k-1) \underline{P}(k-1|k-1) \underline{A}'(k-1) + \underline{Q}(k-1) \quad (3.41)$$

$$\underline{P}(k|k) = [\underline{I} - \underline{G}(k)\underline{H}(k)] \underline{P}(k|k-1) \quad (3.42)$$

c. Filter update equation

$$\hat{\underline{x}}(k|k) = \hat{\underline{x}}(k|k-1) + \underline{G}(k) [\underline{z}(k) - \underline{c}(\hat{\underline{x}}(k|k-1))] \quad (3.43)$$

d. Prediction equation

$$\hat{\underline{x}}(k+1|k) = \underline{a}(\hat{\underline{x}}(k|k), \underline{u}(k), k) \quad (3.44)$$

A block diagram of the system and estimator is shown in Figure 3.2.

The gains can be pre-computed and stored if it is assumed that the nominal trajectory is known in advance.

To answer the question, "where does the nominal trajectory $\underline{x}(k)$ come from?", two approaches are usually used.

In the first, an approximate trajectory is used. This trajectory is known in advance from known data or may have been computed by solving the state equations

$$\underline{x}^{(0)}(k+1) = \underline{a}(\underline{x}^{(0)}(k), \underline{u}(k), k) \quad (3.45)$$

with the initial condition $\underline{x}^{(0)}(0) = E[\underline{x}(0)]$. If the uncertainty in $\underline{x}(0)$ is large, the solution of the above equation may be "far" from $\underline{x}(k)$ to make the linearization sufficiently accurate. If an accurate nominal trajectory is available, the gain can be computed off-line and stored.

In the second approach, the estimates produced by the filter are used as the nominal trajectory about which the

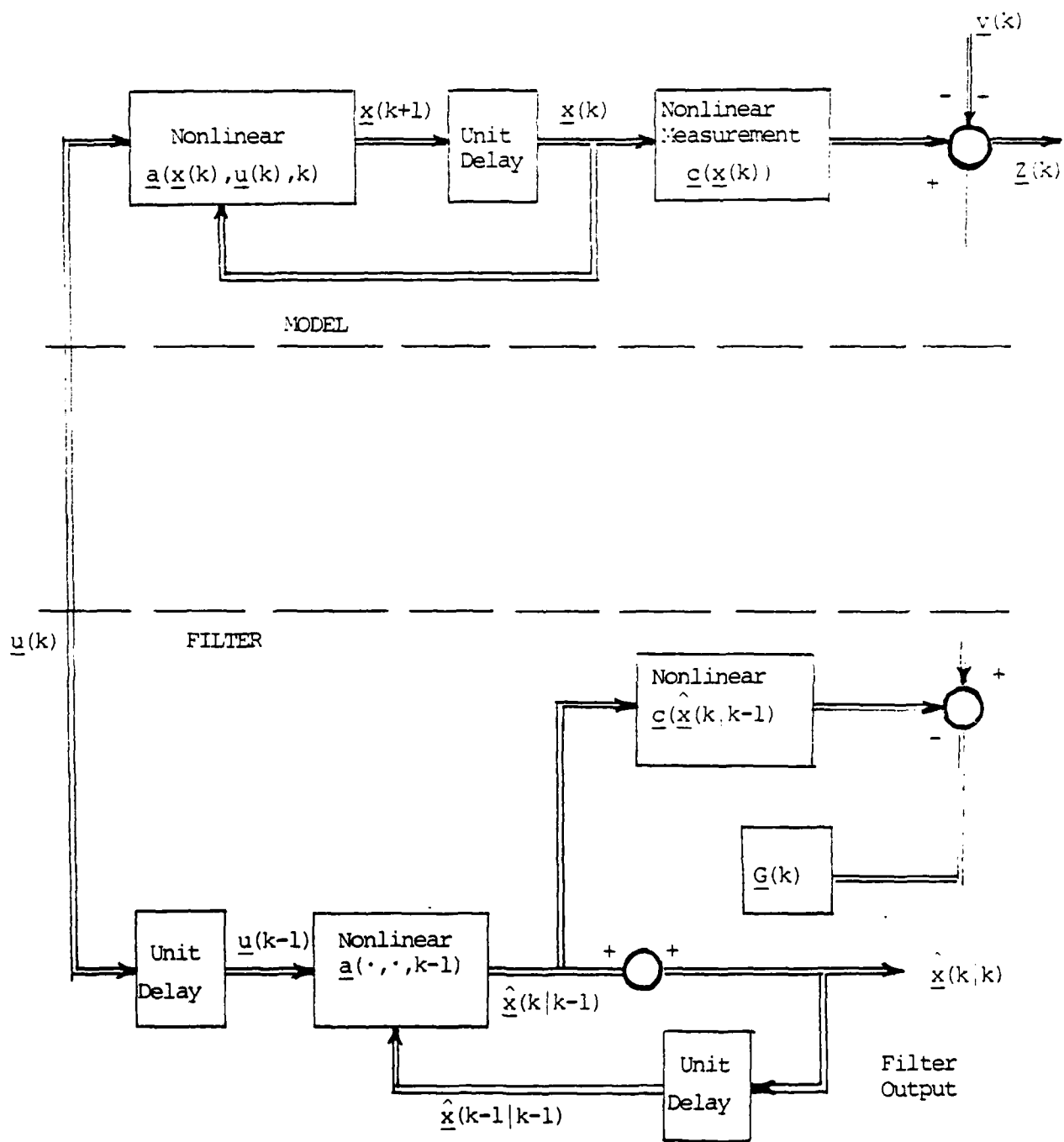


Figure 3.2 E.K.F. Block Diagram of the System and Estimator

linearization is performed. The matrices $\underline{A}(k)$ and $\underline{H}(k)$ must be used to generate $\underline{G}(k)$. The best trajectory information available when $\underline{H}(k)$ must be evaluated is $\hat{\underline{x}}(k|k-1)$; when $\underline{A}(k)$ is to be evaluated, $\hat{\underline{x}}(k|k)$ is available.

The filter that results from using this approach is called an Extended Kalman Filter. The gain and covariance equations must be solved on-line. The procedure of the filter's calculations is given below:

1. Start with $\hat{\underline{x}}(0|-1)$ and evaluate $\underline{H}(0)$ using:

$$\underline{H}(k) = \left. \frac{\partial \underline{c}}{\partial \underline{x}} \right|_{(\underline{x}(k|k-1))} \quad (3.46)$$

- 2.

$$\underline{P}(0|-1) = \underline{M} = E\{(\hat{\underline{x}}(0|-1) - \underline{x}(0))(\hat{\underline{x}}(0|-1) - \underline{x}(0))'\} \quad (3.47)$$

Use this matrix to compute $\underline{G}(0)$ given by

$$\underline{G}(k) = \underline{P}(k|k-1)\underline{H}'(k) [\underline{H}(k)\underline{P}(k|k-1)\underline{H}'(k) + \underline{R}(k)]^{-1} \quad (3.48)$$

3. Compute $\hat{\underline{x}}(0|0)$ from:

$$\hat{\underline{x}}(k|k) = \hat{\underline{x}}(k|k-1) + \underline{G}(k) [\underline{z}(k) - \underline{c}(\hat{\underline{x}}(k|k-1))] \quad (3.49)$$

or

$$\hat{\underline{x}}(0|0) = \hat{\underline{x}}(0|-1) + \underline{G}(0) [\underline{z}(0) - \underline{c}(\hat{\underline{x}}(0|-1))] \quad (3.50)$$

and $\hat{\underline{x}}(1|0)$ from:

$$\hat{\underline{x}}(k+1|k) = \underline{a}(\hat{\underline{x}}(k|k), \underline{u}(k), k) \quad (3.51)$$

4. Compute $\underline{P}(0|0)$ from:

$$\underline{P}(k|k) = [\underline{I} - \underline{G}(k) \underline{H}(k)] \underline{P}(k|k-1) \quad (3.52)$$

5. Since $\hat{\underline{x}}(0|0)$ is available before $\underline{A}(0)$ is calculated, the value of $\hat{\underline{x}}(0|0)$ is used, hence:

$$\underline{A}(0) = \left. \frac{\partial \underline{a}}{\partial \underline{x}} \right|_{(\hat{\underline{x}}(0|0), \underline{u}(0), 0)} \quad (3.53)$$

and then $\underline{P}(1|0)$ is computed from:

$$\underline{P}(k|k-1) = \underline{A}(k-1) \underline{P}(k-1|k-1) \underline{A}'(k-1) + \underline{Q}(k-1) \quad (3.54)$$

The process is repeated by recycling through steps 1 to 5.

The Extended Kalman Filter presented was obtained by a Taylor series expansion up to the first order terms. This filter obviously introduces errors in the equations where the higher order terms are neglected.

There are several ways of compensating for these errors.

Very roughly , an addition of "artificial process noise" covariance can be made, for compensation of the errors is the state prediction. Also a multiplication of the state covariance by a scalar $\lambda > 1$ at every sampling time can be carried out. This multiplication is equivalent to the filter having a "fading memory," i.e., at every sampling time the past data is "discounted" by attaching to it a higher covariance (lower accuracy).

The above subject is covered with more details in Reference 8, pp. 4-52--4.58 and in Reference 5, pp. 3.3-1--3.4-1.

IV. ERROR COVARIANCE MATRIX AND TARGET TRACKING QUALITY

A. DEFINITION OF ERROR COVARIANCE MATRIX

The error $\tilde{x}$ in the estimate of a state vector is defined to be the difference between the estimated ($\hat{x}$) and the actual (x) values:

$$\tilde{x} = \hat{x} - x \quad (4.1)$$

The above difference is known as the error vector or estimate error. The covariance of the estimate error is

$$P = E[(\hat{X}(t) - X(t)(\hat{X}(t) - X(t))')] \quad (4.2)$$

It provides a statistical measure of the uncertainty in x . It is possible to discuss the properties of the covariance matrix independently of the mean value of the state. The information contained is sufficient to generate indicators of the estimate quality.

B. INFORMATION CONTAINED IN THE ERROR COVARIANCE MATRIX

There are five important characteristics of the error covariance matrix which relate the matrix to the state vector and its estimate.

- (1) The error covariance matrix of an n -component state vector is an n by n symmetric matrix.

- (2) The diagonal elements of the error covariance matrix are the mean square errors of the error vector components.
- (3) The trace of the error covariance matrix is the mean square length of the error vector.
- (4) The off-diagonal terms of the matrix are correlations between the elements of the error vector $\hat{\underline{X}}(t) - \underline{X}(t)$.
- (5) The error covariance matrix $\underline{P}(k)$ tends to the system dynamics covariance matrix $\underline{Q}(k)$, as k goes to infinity. This means that as more information is available about the state vector (observations) the estimate uncertainty approaches the uncertainty of the environment in which the state vector exists.

C. ERROR ELLIPSOIDS ABOUT A TARGET POSITION

Frequently it is significant to have available an indication of the quality of the estimates. One approach to achieve this is the proper use of the error covariance matrix $\underline{P}(k|k)$. The outline of this approach is described below.

It is assumed that the initial state of the model $\underline{x}(0)$ and the random processes $\underline{v}(k)$, $\underline{w}(k)$ are Gaussian. If this assumption is satisfied then it can be shown that:

$$\underline{x}(k), \hat{\underline{x}}(k|k) \text{ and } \underline{e}(k|k) \triangleq \hat{\underline{x}}(k|k) - \underline{x}(k)$$

are Gaussian. The results obtained apply only to this case.

The probability density function for $\underline{e}(k|k)$ can be written as

$$f_E(\underline{e}(k|k)) = \frac{1}{(2\pi)^{n/2} |\underline{P}(k|k)|^{1/2}} \exp\left\{-\frac{1}{2} \underline{e}'(k|k) \underline{P}^{-1}(k|k) \underline{e}(k|k)\right\} \quad (4.3)$$

For a fixed time k this expression can be written as

$$f_E(\underline{e}) = \frac{1}{(2\pi)^{n/2} |\underline{P}|^{1/2}} \exp\left\{-\frac{1}{2} \underline{e}' \underline{w} \underline{e}\right\} \quad (4.4)$$

where $\underline{w} \triangleq \underline{P}^{-1}(k|k)$. In order to determine the locus of points where the density function $f_E(\underline{e})$, has a constant value the above equation has to be examined. It is seen that $f_E(\underline{e})$ has a constant value whenever

$$\frac{1}{2} \underline{e}' \underline{w} \underline{e} = \text{constant} \triangleq 1^2 \quad (4.5)$$

It can be shown that the points $\underline{e}$ which satisfy Equation (4.5) are hyperellipsoids (in two dimensions, ellipses).

If the left side of (4.5) is expanded for the two-dimensional case (the same approach can be extended to n dimensions), we have:

$$\frac{1}{2} w_{11} e_1^2 + w_{12} e_1 e_2 + \frac{1}{2} w_{22} e_2^2 = 1^2 \quad (4.6)$$

where the symmetry of $\underline{w}$ has been used ($w_{12} = w_{21}$). Equation (4.6) is an ellipse ($w_{11} > 0$, $w_{22} > 0$ and $w_{11} w_{22} > w_{12}^2$). It is not easy to be recognized as such because its principal axes do not coincide with the coordinate axes as shown in Figure 4.1.

First the principal axes must be determined and then the ellipse can be rewritten in terms of $\underline{y}^{(1)}$ and $\underline{y}^{(2)}$ as coordinate axes.

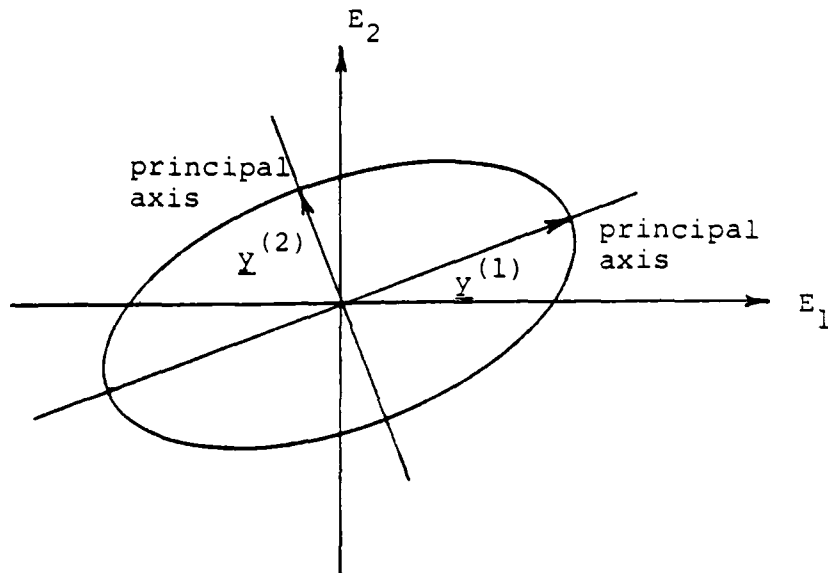


Figure 4.1 Error Ellipse

The ellipse in the new coordinate system is described by

$$\lambda_1 y_1^2 + \lambda_2 y_2^2 = \frac{y_1^2}{1/\lambda_1} + \frac{y_2^2}{1/\lambda_2} = 1^2 \quad (4.7)$$

where $y^{(1)}$ and $y^{(2)}$ are eigenvectors of $\underline{W}$ and λ_1 and λ_2 are the corresponding eigenvalues.

As it has already been mentioned, $\underline{W} = \underline{P}^{-1}$. Equation (4.7) is an ellipse in terms of the eigenvalues λ_1 and λ_2 and the eigenvectors $y^{(1)}$ and $y^{(2)}$ of $\underline{W}$. The expression which gives the ellipse in terms of the eigenvalues and eigenvectors of $\underline{P}$ is

$$\frac{y_1^2}{a_1} + \frac{y_2^2}{a_2} = 1^2 \quad (4.8)$$

where $y^{(i)}$ is an eigenvector for $\underline{P}$ and $a_i = 1/\lambda_i$ is the corresponding eigenvalue.

This result can be generalized to n dimensions as well.

Given the quadratic form

$$\frac{1}{2} \underline{e}' \underline{P}^{-1} \underline{e} = 1^2 \quad (4.9)$$

the eigenvalues of $\underline{P}$ are $a_1, a_2, \dots, a_n$ and the corresponding eigenvectors are $y^{(1)}, y^{(2)}, \dots, y^{(n)}$. The quadratic form (Equation (4.9)) describes a hyperellipsoid which can be written as

$$\frac{y_1^2}{a_1^2} + \frac{y_2^2}{a_2^2} + \dots + \frac{y_n^2}{a_n^2} = 1^2 \quad (4.10)$$

All vectors e in the n -dimensional space can be written as a linear combination of the eigenvectors $y^{(1)}, y^{(2)}, \dots, y^{(n)}$. The coefficient of $y^{(1)}$ in the linear combination is y_1 , the coefficient of $y^{(2)}$ is y_2 , etc.

So the surfaces of equal probability density are ellipses (or hyperellipsoids).

The problem can now be stated as follows:

For a specified value of λ , what is the probability that e lies within or on the ellipsoid?

These probabilities have been tabulated below for a few values of n and λ .

TABLE I
Probabilities for Error Ellipsoids

n	λ		
	1	2	3
1	.683	.955	.997
2	.394	.865	.989
3	.200	.739	.977

For example, for a system having $n = 2$:

Probability error inside $\lambda = 1$ ellipse = 0.394

Probability error inside $\lambda = 2$ ellipse = 0.865

Probability error inside $\lambda = 3$ ellipse = 0.989

Therefore, if the error covariance matrix P is given, the error ellipsoids can be determined by finding the eigenvalues and eigenvectors.

The error ellipses are useful in visualizing the estimation error. By using them we can consider the true state value to lie within a certain region surrounding the estimate.

For details and derivations, see Reference 8, pp. 4.44--4.51.

V. MANEUVERING TARGETS

A. GENERAL DESCRIPTION OF THE PROBLEM

The maneuvering target is generally described by the equation:

$$\underline{x}(k+1) = \underline{F}(k)\underline{x}(k) + \underline{G}(k)\underline{u}(k) + \underline{v}(k) \quad (3.1)$$

where the matrices $\underline{F}(k)$, $\underline{G}(k)$ are assumed known and the process noise $\underline{v}(k)$ is zero mean, white random sequence with covariance matrix $\underline{Q}(k)$. The main characteristic of the maneuvering target equation is that the inputs $\underline{u}(k)$ are unknown.

In the following, linear models are examined for simplicity but the same techniques can be applied to nonlinear cases.

A number of different approaches to the maneuvering target problem have appeared in the literature. The most commonly used model is, due to simplicity requirements, a kinematic model with states consisting of position, velocity, and in most cases also acceleration, driven by white noise.

It is possible to divide the different approaches into two broad categories:

- A. The unknown input (maneuver command) is modeled as a random process.
- B. The unknown input is estimated in real time.

The random process type models can be further classified into two categories, depending on their statistical properties:

A1. White noise

A2. Autocorrelated (Markov) noise

All these methods are approximations because in general, the maneuvers are not stochastic processes.

In the input estimation case the assumption of having a constant input over a certain period of time is usually made. The estimation can be based on the least squares criterion and can be used in the following ways:

B1. The estimated input corrects the state estimate.

B2. The input becomes an extra state component (state is augmented) that is reestimated within the state.

In the following section only one method is described. This method is illustrated by an example in Chapter VI. It was selected from the others, due to its simplicity.

For detailed descriptions of different approaches to the problem, see Reference 5.

B. THE WHITE NOISE MODEL WITH ADJUSTABLE LEVEL

In this method a certain low level of process noise is assumed in the filter.

A maneuver is considered as a large innovation. The detection procedure is based on the square norm of the innovations

$$\epsilon_v(k) = \underline{v}'(k) \underline{S}^{-1}(k) \underline{v}(k) \quad (5.2)$$

where

$$\underline{v}(k) = \underline{z}(k) - \hat{\underline{z}}(k|k-1) \quad (5.3)$$

Based on the target model (for a non-maneuvering target)
a threshold is set up

$$P\{\epsilon_v(k) < \epsilon_{\max}\} = 1 - a \quad (5.4)$$

where a is arbitrary. For example, $a = 0.05$.

If the threshold is exceeded, $Q(k-1)$ is multiplied by
a scaling factor ϕ until ϵ_v is reduced to the threshold
 $\epsilon_{\max}$.

When using the factor ϕ the covariance of the innovations
takes the form:

$$\underline{S}(k) = \underline{C}(k) [\underline{F}(k-1) \underline{P}(k-1|k-1) \underline{F}'(k-1) + \phi \underline{Q}(k-1)] \underline{C}'(k) + \underline{R}(k) \quad (5.5)$$

This obviously reduces the value of $\epsilon_v(k)$.

An analogous technique is possible to be used to lower
the process noise level after the maneuver.

VI. CHARACTERISTIC COMPUTER EXAMPLES

A. A LINEAR KALMAN FILTER EXAMPLE

1. Problem Description

The target is assumed to be described by the system:

$$\underline{x}(k+1) = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \underline{x}(k) + \begin{bmatrix} T \\ 1 \end{bmatrix} \underline{w}(k) \quad k = 0, 1, \dots \quad (6.1)$$

The available measurements are of the type:

$$\underline{z}(k) = [1 \ 0] \underline{x}(k) + \underline{v}(k) \quad k = 1, 2, \dots \quad (6.2)$$

where $\underline{w}(k) \sim N(0, Q)$, $\underline{v}(k) \sim N(0, R)$ are mutually independent, zero mean, white random sequences.

The initial state is

$$\underline{x}(0) = \begin{bmatrix} 5 \\ 1 \end{bmatrix} \quad (6.3)$$

Two measurements $z(0)$ and $z(1)$ are made to initialize a Kalman filter.

The sampling time is given as $T = 0.1$.

The process and measurement noise are to be yielded by a Gaussian random number generator, given that $Q = R = 0.02$.

Case 1

After the run of the Kalman filter for $k = 2, \dots, 100$, the following expressions are useful to be plotted:

- a. True trajectory vs. estimated trajectory in the x_1, x_2 plane.
- b. Position error variance

$$p_{11}(1|1), p_{11}(2|1), p_{11}(2|2), \dots$$

- c. Normalized position error

$$[x_1(k) - \hat{x}_1(k|k)] / [p_{11}(k|k)]^{1/2}, \quad k = 2, \dots, 100$$

- d. Velocity error variance

$$p_{22}(1|1), p_{22}(2|1), p_{22}(2|2), \dots$$

- e. Normalized velocity error

$$[x_2(k) - \hat{x}_2(k|k)] / [p_{22}(k|k)]^{1/2}$$

- f. Normalized state error squared

$$[\underline{x}(k) - \hat{\underline{x}}(k|k)]' P^{-1}(k) [\underline{x}(k) - \hat{\underline{x}}(k|k)]$$

- g. Normalized innovation error

$$[z(k) - \hat{z}(k|k-1)] / [p_{11}(k|k-1) + R]^{1/2}$$

Case 2

In order to see the effect of running the Kalman filter with a different Q than that of the model, it is helpful to plot the same expressions as in Case 1, using a different Q for the target's model, say $Q = 9 \times 10^{-2}$.

Case 3

Finally, it is also interesting to see the change on the same expressions as the Q of both the model and the filter increases. For this reason new plots of these expressions are to be obtained, using $Q = 9 \times 10^{-2}$ for both the model and the filter.

2. Analysis

a. True Trajectory

The matrix form of the given model is converted in the following equations:

$$\underline{x}_1(k+1) = \underline{x}_1(k) + T\underline{x}_2(k) + T\underline{w}(k) \quad (6.4)$$

$$\underline{x}_2(k+1) = \underline{x}_2(k) + \underline{w}(k) \quad (6.5)$$

where $T = 0.1$, $\underline{w}(k)$ is generated by a random number generator function and

$$\underline{x}(0) = \begin{bmatrix} 5 \\ 1 \end{bmatrix}.$$

b. Measurements

From the given measurement equation it is obvious that

$$\underline{z}(k) = \underline{x}_1(k) + \underline{v}(k) \quad (6.6)$$

where $x_1(0) = 5$ and $\underline{v}(k)$ is generated by a random number generator function.

c. Derivation of $x(1|1)$, $p(1|1)$

It is reasonable to start with the following estimations for the position and velocity vectors:

$$\hat{\underline{x}}(1|1) = \begin{bmatrix} \hat{x}_1(1|1) \\ \hat{x}_2(1|1) \end{bmatrix} = \begin{bmatrix} z(1) \\ \frac{z(1) - z(0)}{T} \end{bmatrix} \quad (6.7)$$

then

$$\hat{x}_1(1|1) = \underline{x}_1(1) + \underline{v}(1) \quad (6.8)$$

$$\hat{x}_2(1|1) = \frac{x_1(1) - x_1(0)}{T} + \frac{v(1) - v(0)}{T} \quad (6.9)$$

Using these values in the initial error covariance matrix

$$P(1|1) = E \begin{bmatrix} \tilde{x}_1^2(1|1) & \tilde{x}_1(1|1)\tilde{x}_2(1|1) \\ \tilde{x}_1(1|1)\tilde{x}_2(1|1) & \tilde{x}_2^2(1|1) \end{bmatrix} \quad (6.10)$$

where

$$\tilde{x}_1(1|1) = x_1(1|1) - \hat{x}_1(1|1) \quad (6.11)$$

$$\tilde{x}_2(1|1) = x_2(1|1) - \hat{x}_2(1|1) \quad (6.12)$$

It is obtained that

$$P(1|1) = \begin{bmatrix} R & \frac{R}{T} \\ \frac{R}{T} & \frac{2R}{T} \end{bmatrix} \quad (6.13)$$

d. Run of the Kalman Filter

(1) Calculation of Gains. The known equation of the Kalman Filter is:

$$\underline{P}(k|k-1) = \underline{\phi} \underline{P}(k-1|k-1) \underline{\phi}' + \underline{Q}(k-1) \quad (6.14)$$

It has been converted in matrix form and after some manipulation, the following equations are obtained:

$$\begin{aligned} \underline{P}_{11}(k|k-1) &= \underline{P}_{11}(k-1|k-1) + T \underline{P}_{21}(k-1|k-1) \\ &+ T \underline{P}_{12}(k-1|k-1) + T^2 \underline{P}_{22}(k-1|k-1) + T^2 \underline{Q}(k) \end{aligned} \quad (6.15)$$

$$\underline{P}_{12}(k|k-1) = \underline{P}_{12}(k-1|k-1) + T \underline{P}_{22}(k-1|k-1) + T \underline{Q}(k) \quad (6.16)$$

$$\underline{P}_{21}(k|k-1) = \underline{P}_{21}(k-1|k-1) + T \underline{P}_{22}(k-1|k-1) + T \underline{Q}(k) \quad (6.17)$$

$$\underline{P}_{22}(k|k-1) = \underline{P}_{22}(k-1|k-1) + \underline{Q}(k) \quad (6.18)$$

Keeping the same procedure, the Kalman Filter equation

$$\underline{G}(k) = \underline{P}(k|k-1)\underline{C}'[\underline{C}\underline{P}(k|k-1)\underline{C}' + \underline{R}(k)]^{-1} \quad (6.19)$$

takes the following form

$$\underline{G}_1(k) = \frac{\underline{P}_{11}(k|k-1)}{\underline{P}_{11}(k|k-1) + \underline{R}(k)} \quad (6.20)$$

$$\underline{G}_2(k) = \frac{\underline{P}_{21}(k|k-1)}{\underline{P}_{11}(k|k-1) + \underline{R}(k)} \quad (6.21)$$

Finally the equation

$$\underline{P}(k|k) = [\underline{I} - \underline{G}(k)\underline{C}]\underline{P}(k|k-1) \quad (6.22)$$

is analyzed in the following:

$$\underline{P}_{11}(k|k) = \underline{P}_{11}(k|k-1) - \underline{G}_1(k)\underline{P}_{11}(k|k-1) \quad (6.23)$$

$$\underline{P}_{12}(k|k) = \underline{P}_{12}(k|k-1) - \underline{G}_1(k)\underline{P}_{12}(k|k-1) \quad (6.24)$$

$$\underline{P}_{21}(k|k) = \underline{P}_{21}(k|k-1) - \underline{G}_2(k)\underline{P}_{11}(k|k-1) \quad (6.25)$$

$$\underline{P}_{22}(k|k) = \underline{P}_{22}(k|k-1) - \underline{G}_2(k)\underline{P}_{12}(k|k-1) \quad (6.26)$$

2. Estimated Trajectories

Following the procedure described above, the estimated trajectories given by the equations:

$$\hat{\underline{x}}(k|k-1) = \underline{F} \hat{\underline{x}}(k-1|k-1) + \underline{\Delta} u(k) \quad (6.27)$$

$$\hat{\underline{x}}(k|k) = \hat{\underline{x}}(k|k-1) + \underline{G}(k) [\underline{z}(k) - \underline{C}(k) \hat{\underline{x}}(k|k-1)] \quad (6.28)$$

take the form:

$$\hat{x}_1(k|k-1) = \hat{x}_1(k-1|k-1) + T \hat{x}_2(k-1|k-1) \quad (6.29)$$

$$\hat{\underline{x}}_2(k|k-1) = \hat{\underline{x}}_2(k-1|k-1) \quad (6.30)$$

$$\hat{\underline{x}}_1(k|k) = \underline{G}_1(k) \underline{z}(k) + (1 - \underline{G}(k)) \hat{\underline{x}}_1(k|k-1) \quad (6.31)$$

$$\hat{\underline{x}}_2(k|k) = \hat{\underline{x}}_2(k|k-1) + \underline{G}_2(k) \underline{z}(k) - \underline{G}_2(k) \hat{\underline{x}}_1(k|k-1) \quad (6.32)$$

e. Normalized State Error Squared

The normalized state error squared is given by the expression:

$$[\underline{x}(k) - \hat{\underline{x}}(k|k)]' \underline{P}^{-1}(k) [\underline{x}(k) - \hat{\underline{x}}(k|k)]$$

By manipulating this expression in its matrix form it is obtained that it is equal to the following:

$$\begin{aligned}
& \frac{1}{P_{11}P_{22} - P_{21}P_{12}} \{ [x_1(k) - \hat{x}_1(k|k)]^2 P_{22} + [x_2(k) - \hat{x}_2(k|k)]^2 P_{11} \\
& - 2P_{21} [x_2(k)x_1(k) - x_2(k)\hat{x}_1(k|k) - \hat{x}_2(k|k)x_1(k) \\
& + \hat{x}_2(k|k)\hat{x}_1(k|k)] \}
\end{aligned}$$

f. Normalized Innovation Error

By making use of the equation

$$\hat{z}(k|k-1) = \underline{c} \hat{x}(k|k-1) \quad (6.33)$$

the given expression for the innovation error becomes

$$\frac{[z(k) - \hat{z}(k|k-1)]}{[P_{11}(k|k-1) + R]^{1/2}} = \frac{[z(k) - \hat{x}_1(k|k-1)]}{[P_{11}(k|k-1) + R]^{1/2}} \quad (6.34)$$

All the above have been properly set in a FORTRAN program (pp. 178-180). The results corresponding to cases 1, 2, and 3 are indicated on pages 66-89, respectively. Some comments have been made on the computer program outputs on pages 65 and 90-94.

3. Comments on the Graphs

a. Normalized Expressions c, e, g of Cases 1, 2, and 3

The normal distribution has p.d.f.:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(x - \bar{x})^2/\sigma^2} \quad (6.35)$$

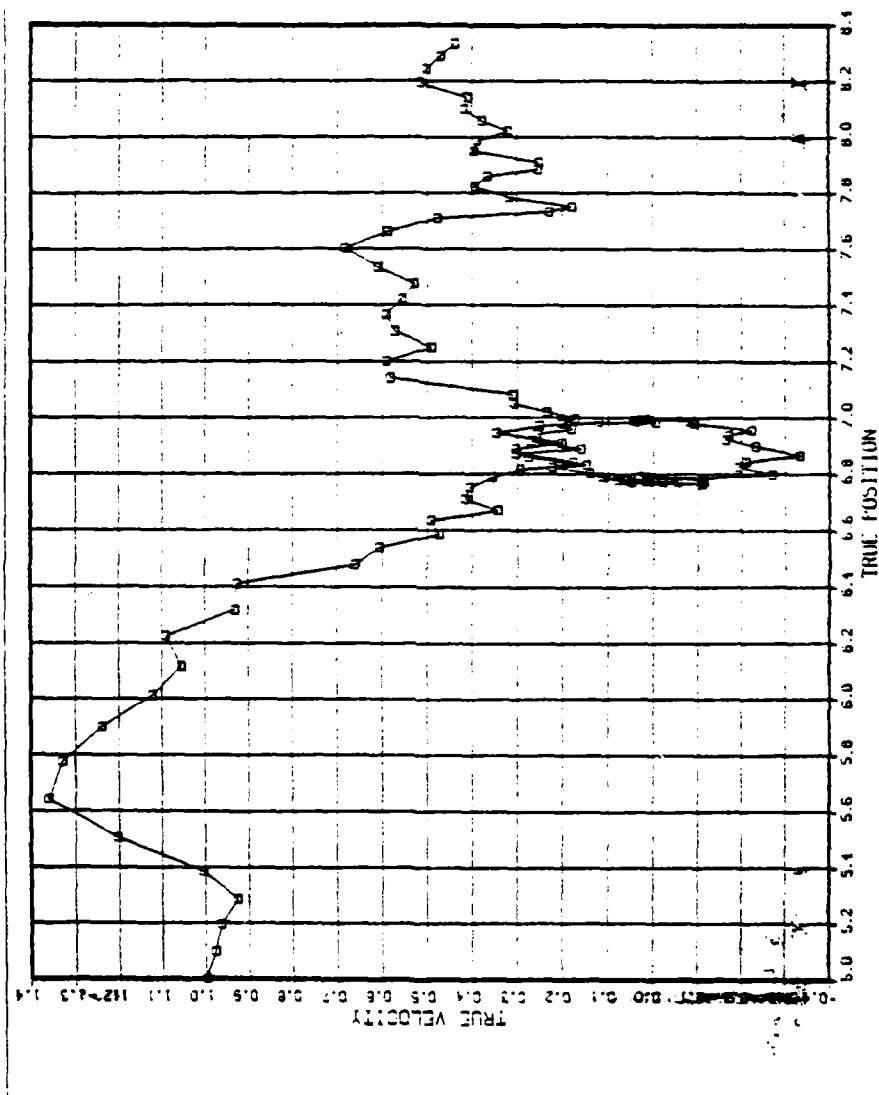


Figure 6.1 True Trajectory in x_1, x_2 Plane. Case 1(a)

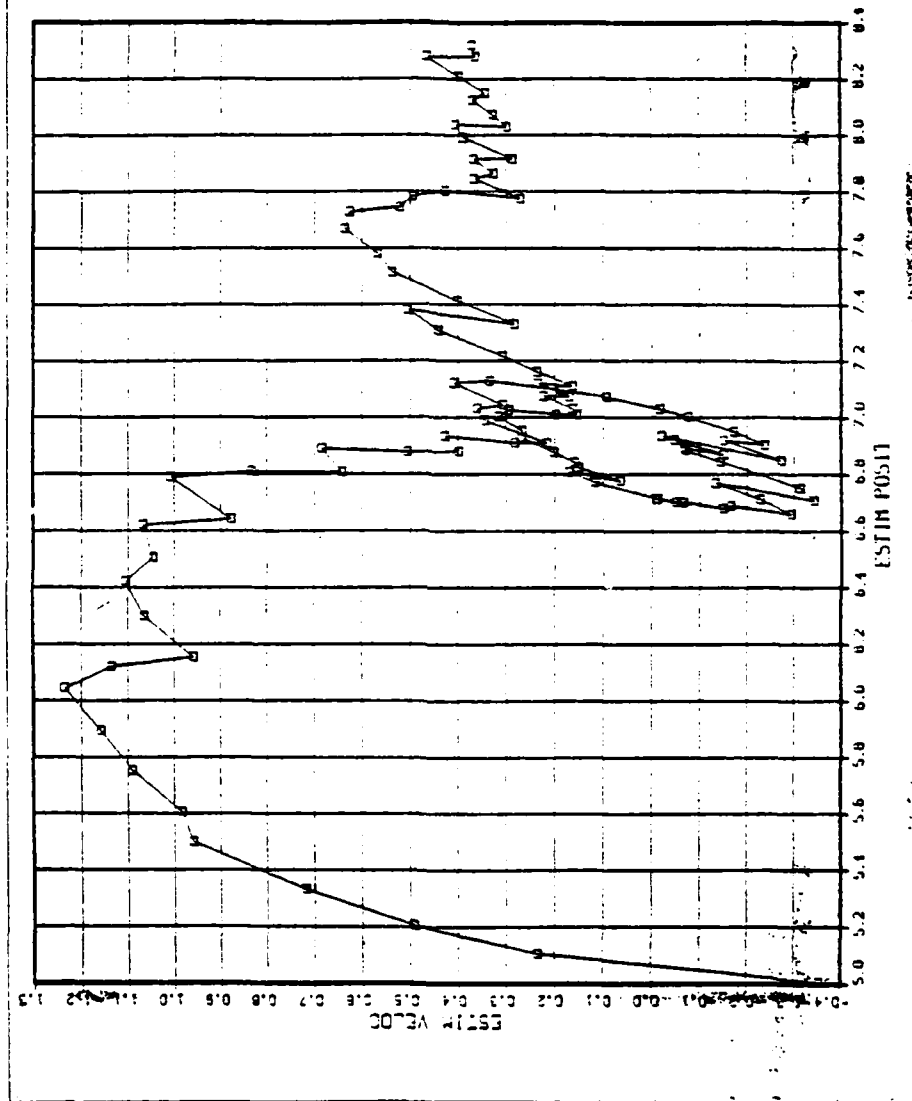


Figure 6.2 Estimated Trajectory in x_1, x_2 Plane, Case 1(a)

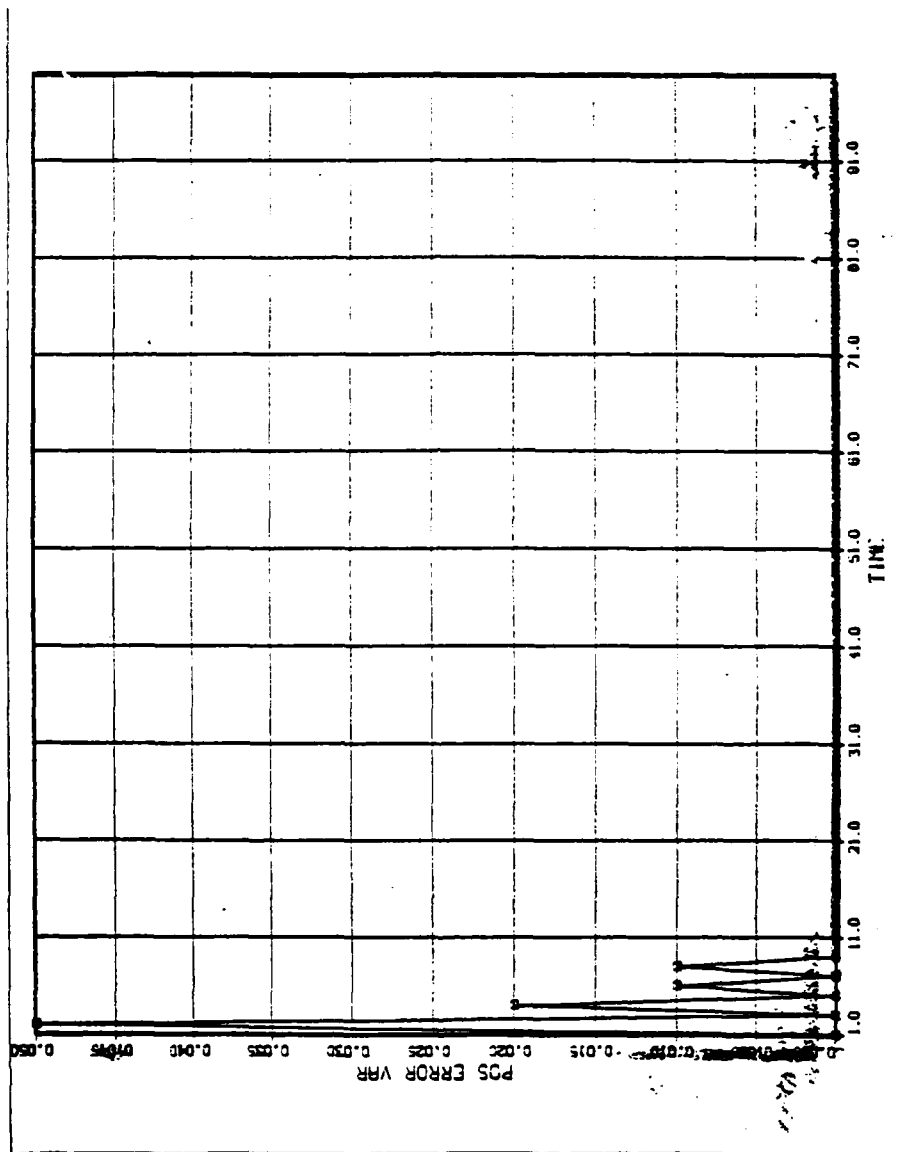


Figure 6.3 Position Error Variance. Case 1(b)

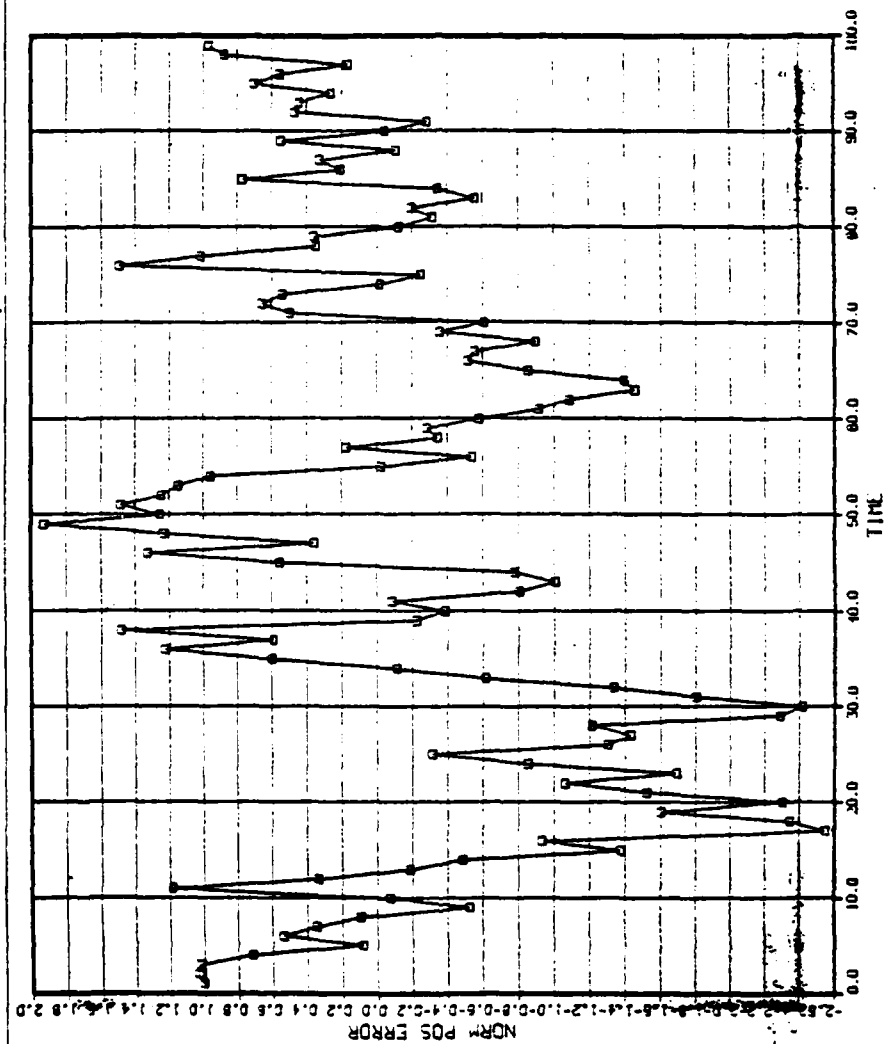


Figure 6.4 Normalized Position Error. Case 1(c)

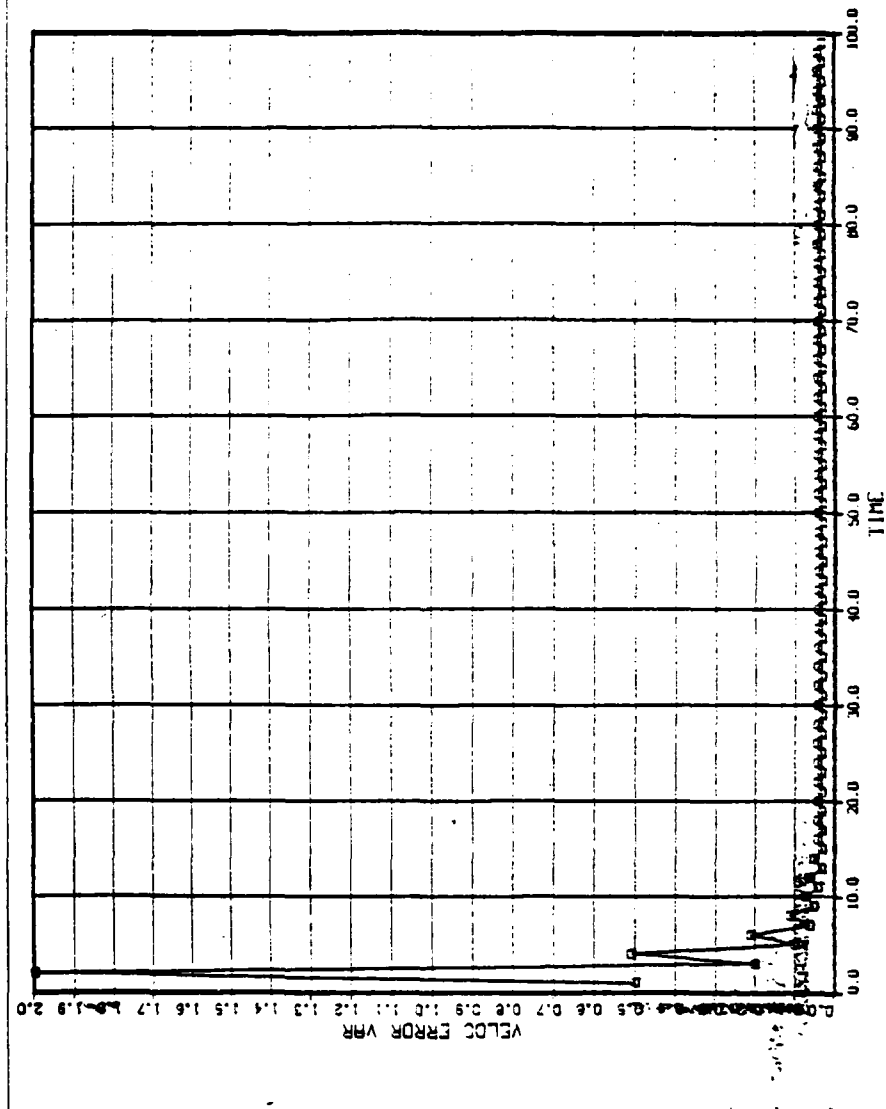


Figure 6.5 Velocity Error Variance. Case 1(d)

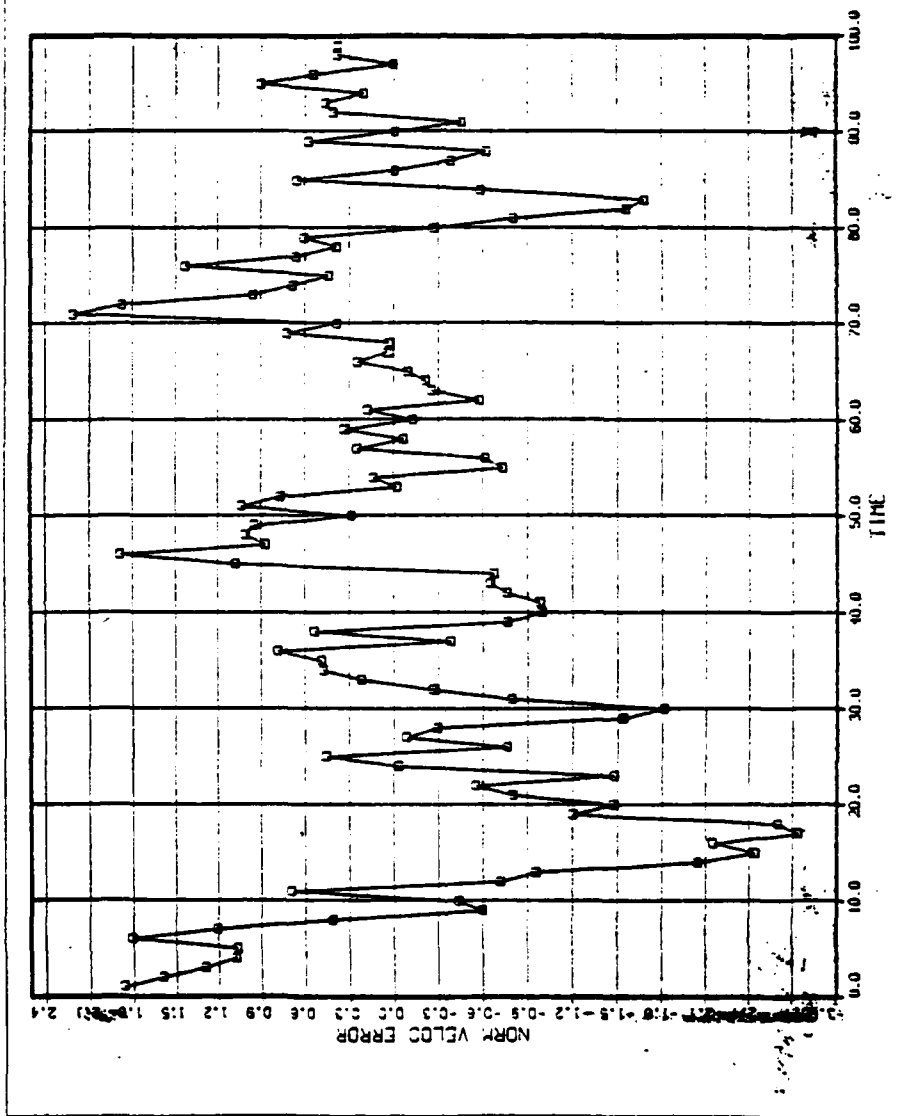


Figure 6.6 Normalized Velocity Error. Case 1(c)

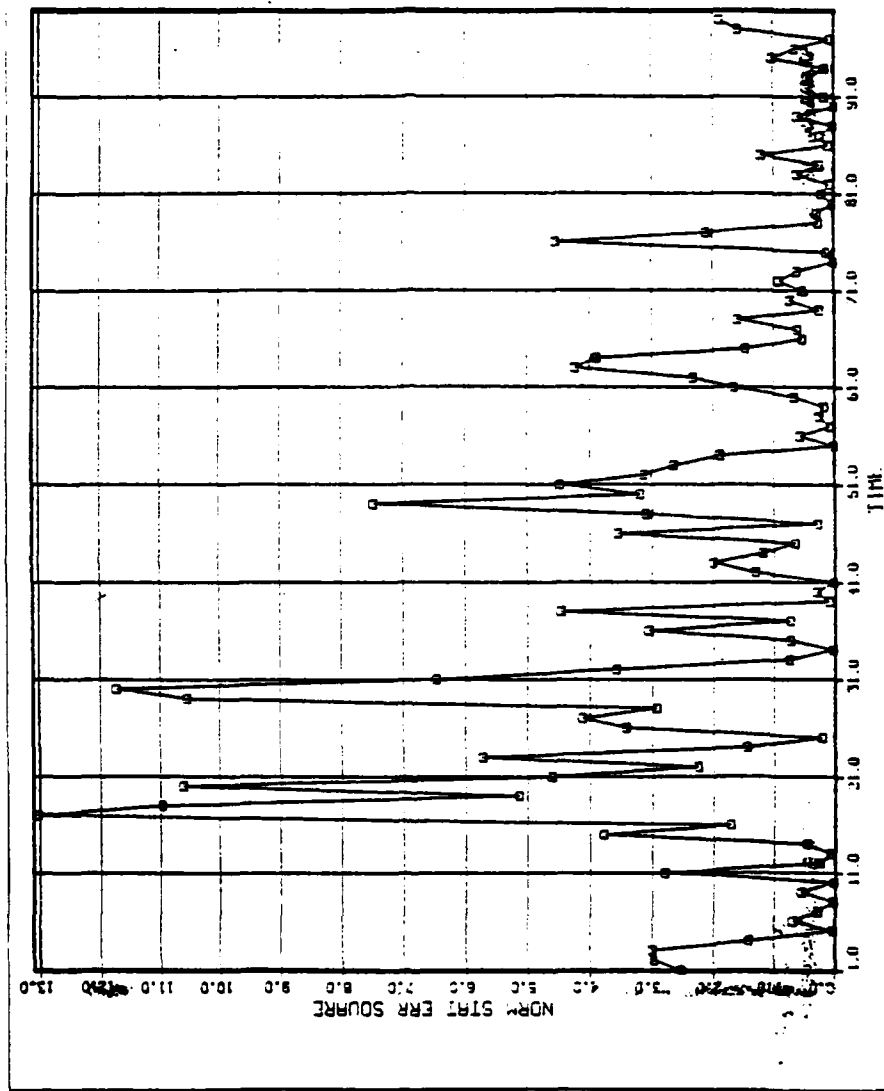


Figure 6.7 Normalized State Error Squared. Case 1(f)

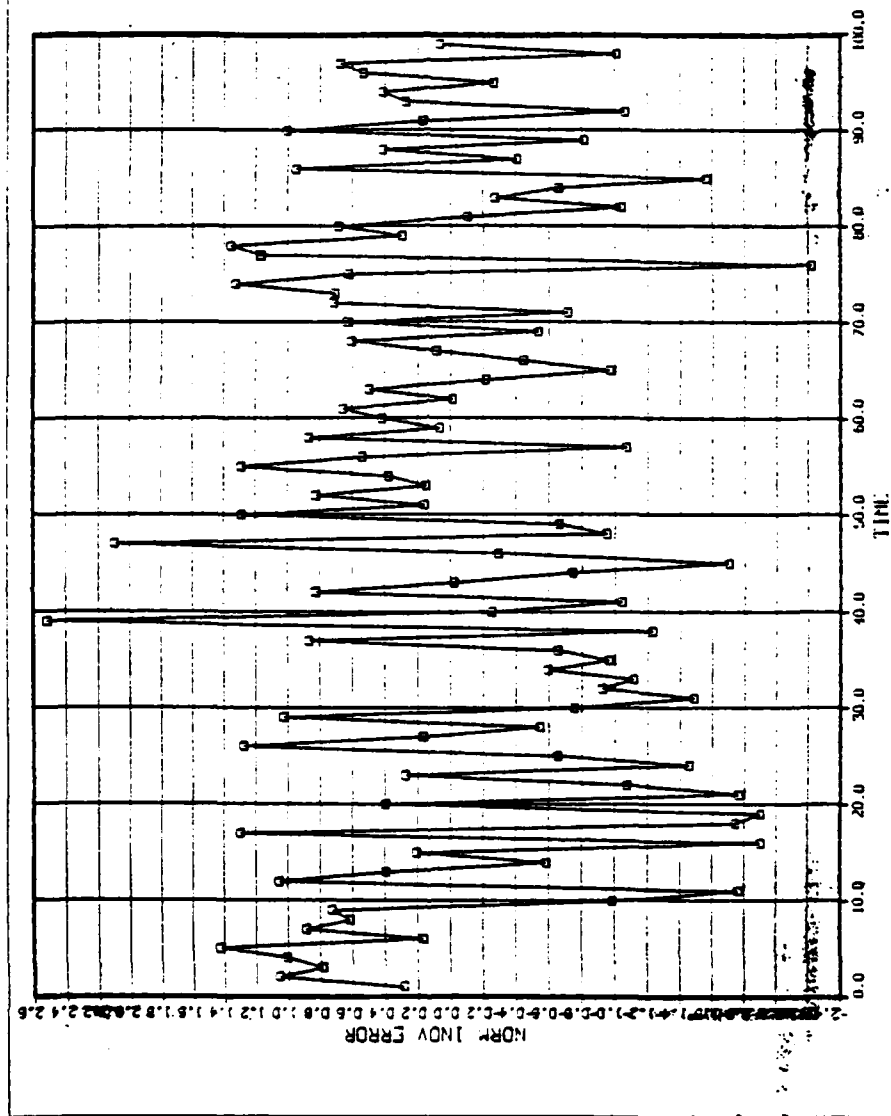


Figure 6.8 Normalized Innovation Error. Case 1(q)

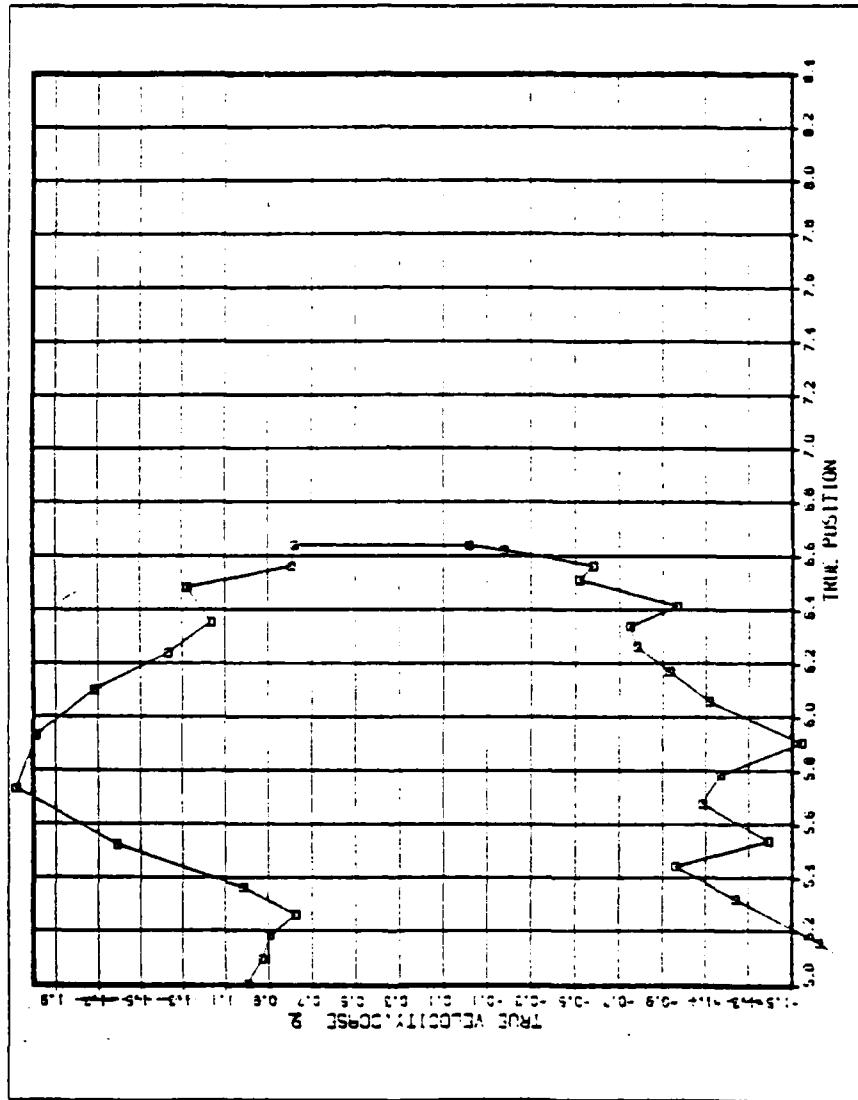


Figure 6.9 True Trajectory in x_1, x_2 Plane. Case 2(a)

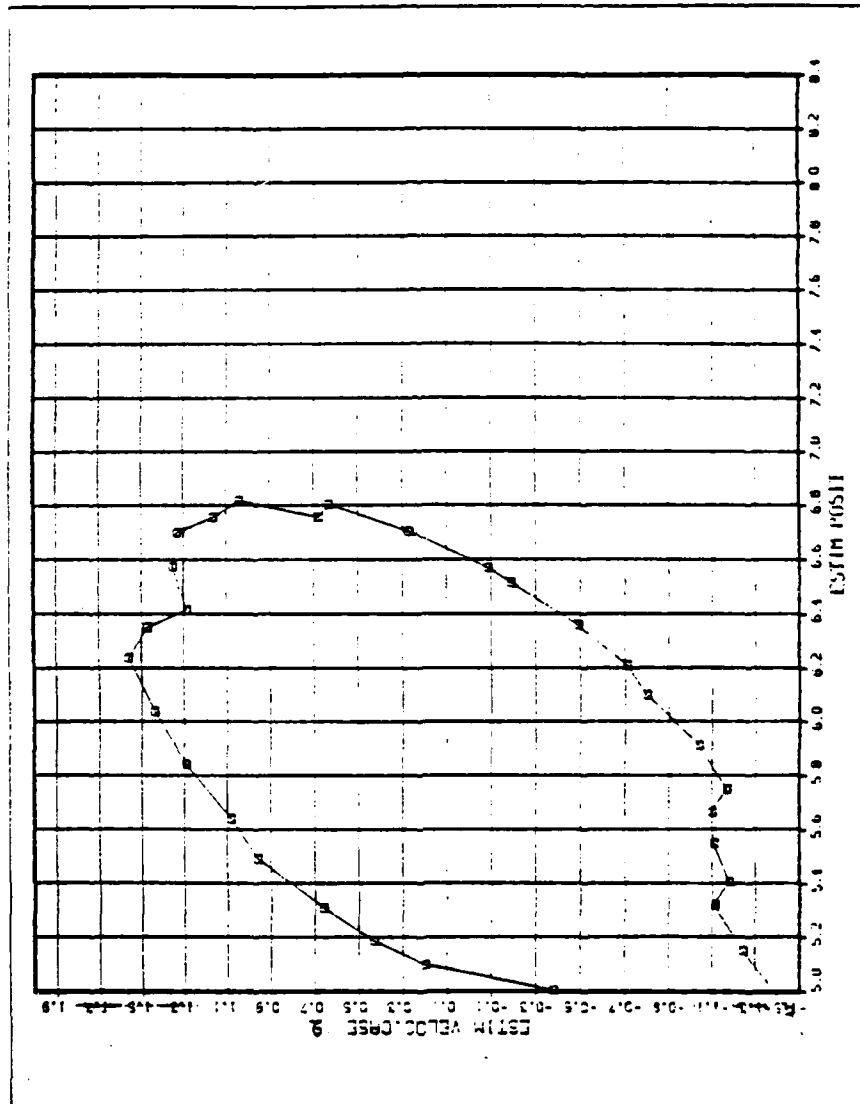


Figure 6.10 Estimated Trajectory in x_1, x_2 Plane. Case 2(a)

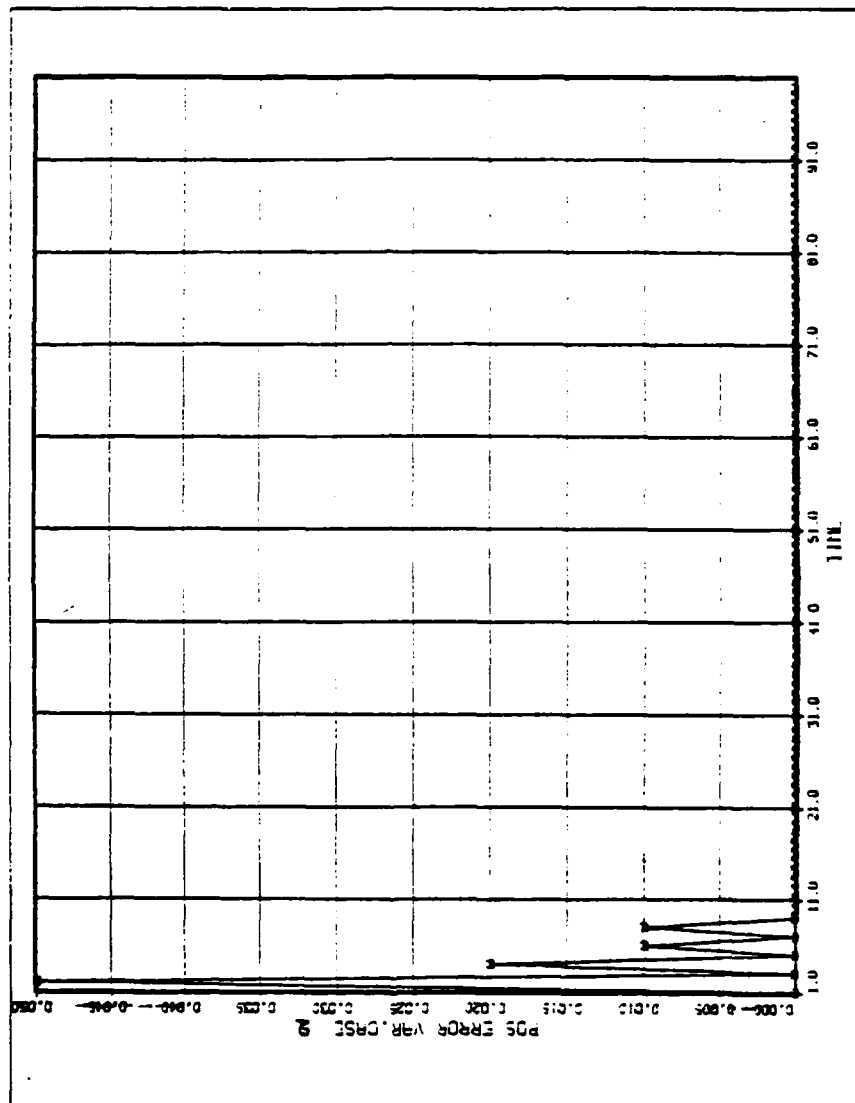


Figure 6.11 Position Error Variance. Case 2(b)

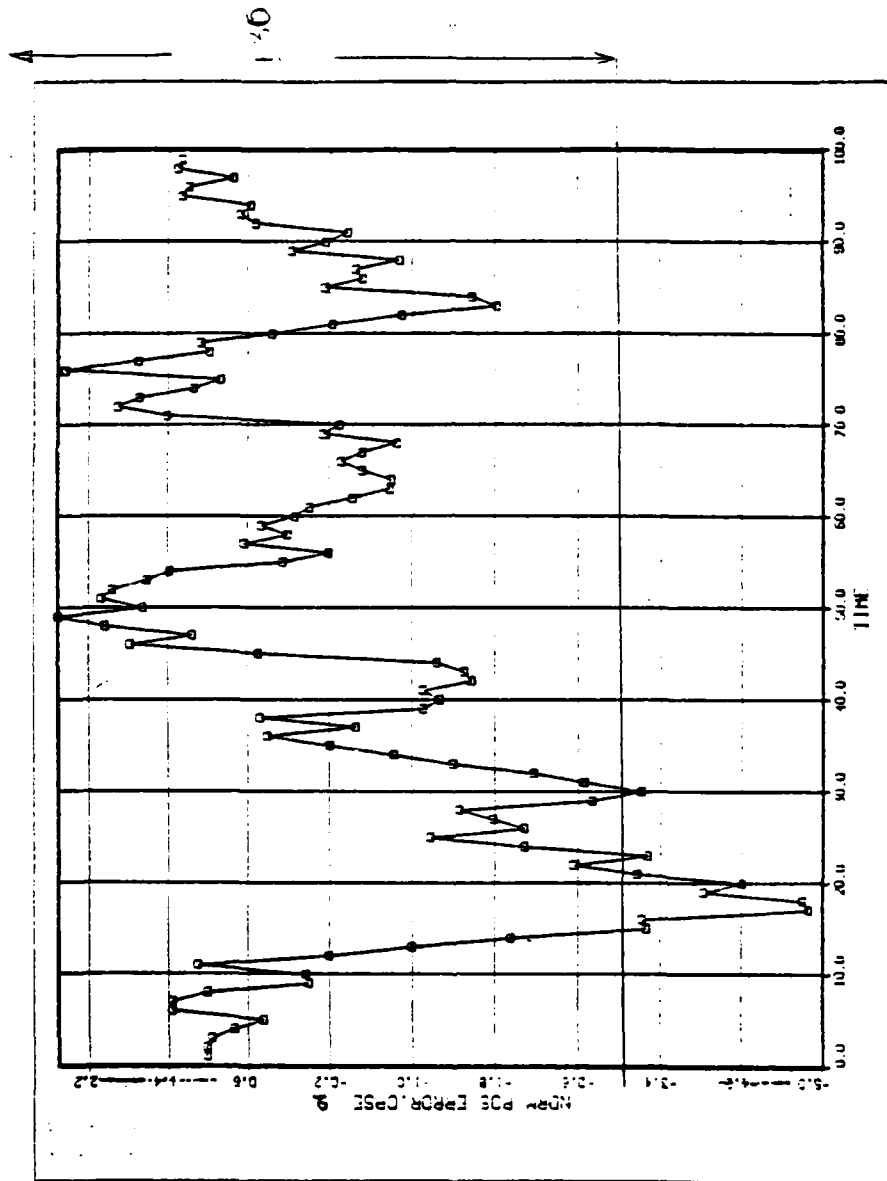


Figure 6.12 Normalized Position Error. Case 2(c)

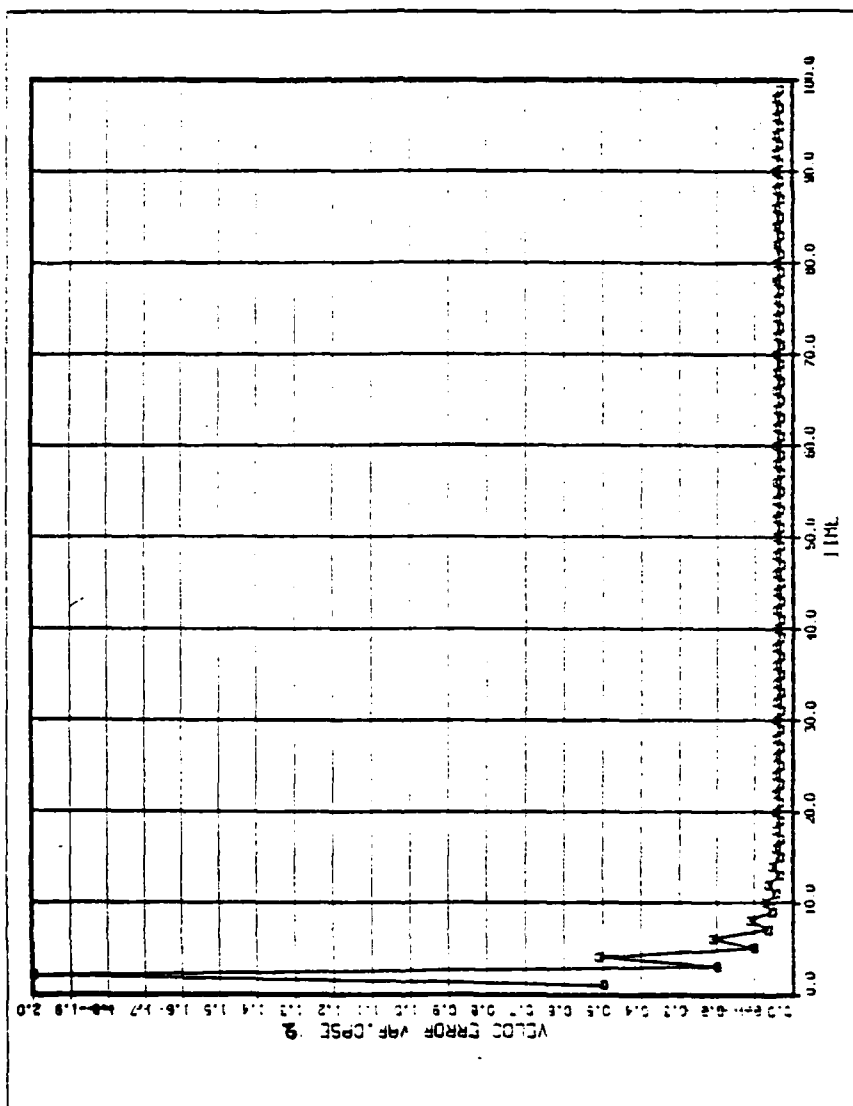


Figure 6.13 Velocity Error Variance. Case 2 (d)

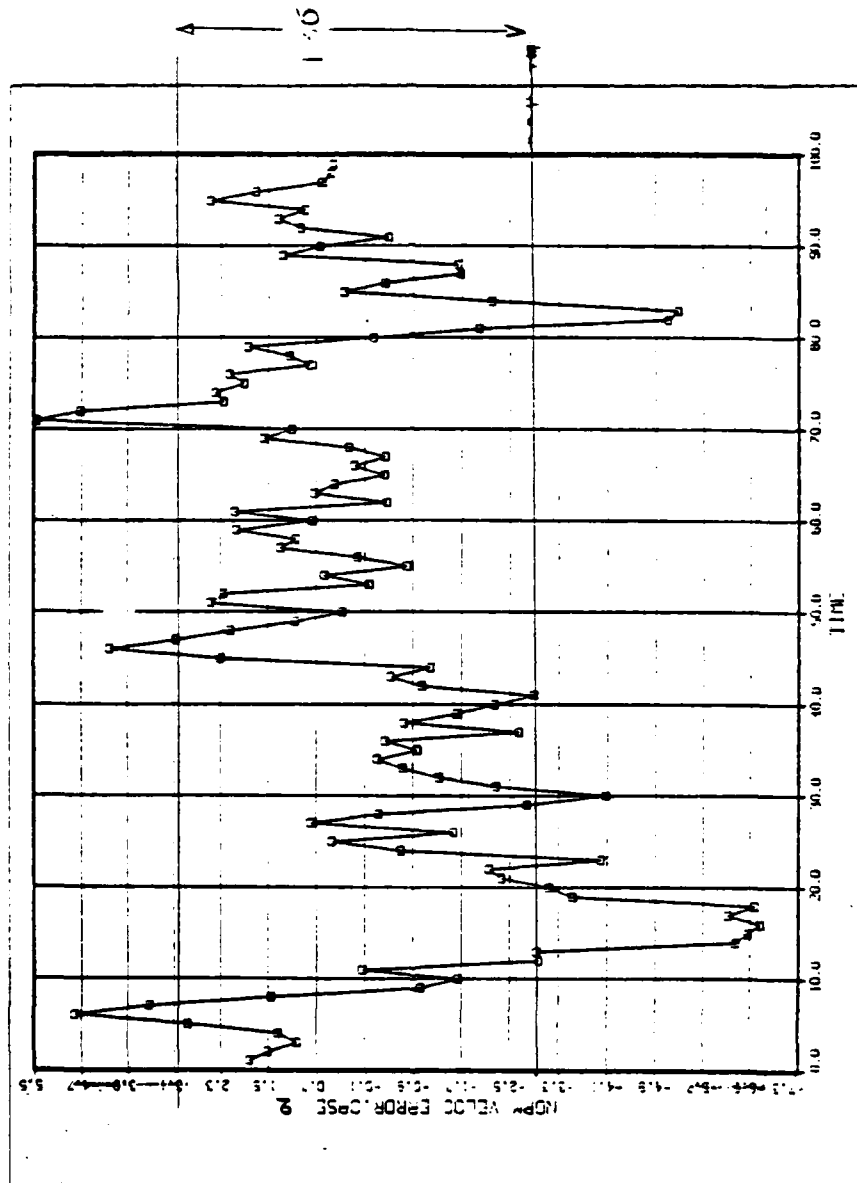


Figure 6.14 Normalized Velocity Error. Case 2(c)

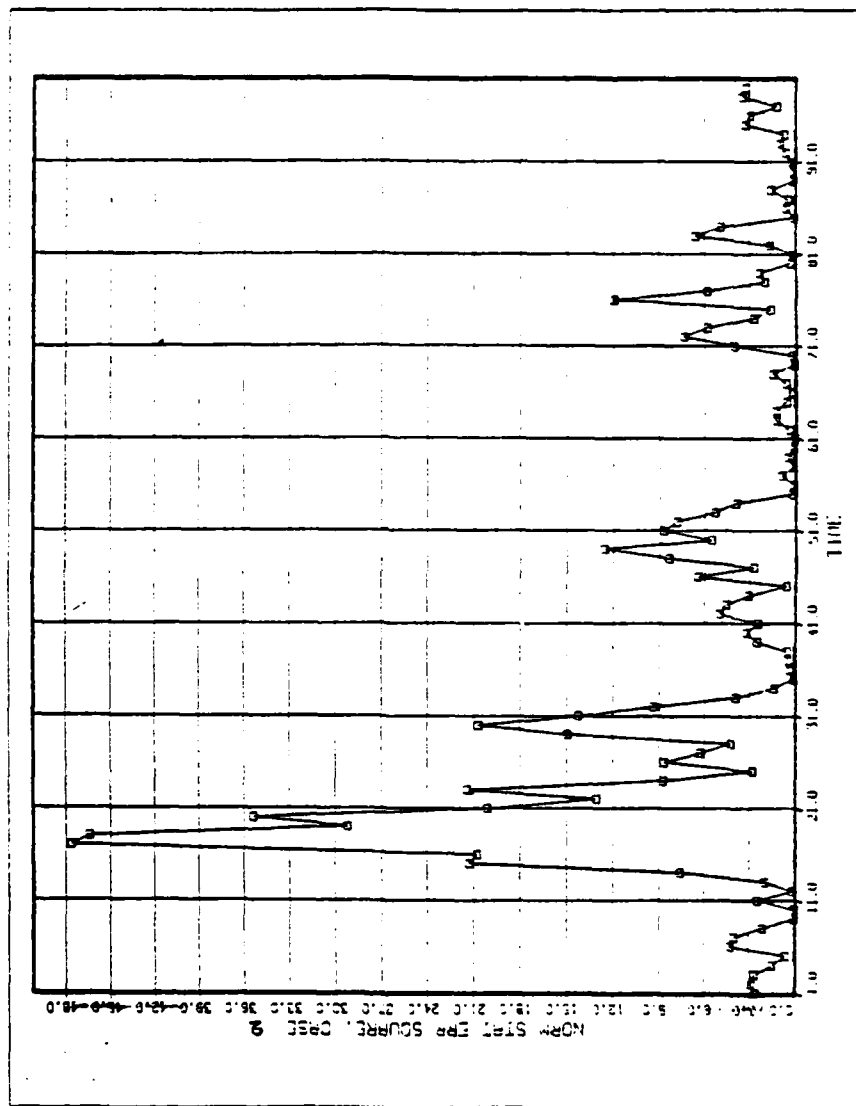


Figure 6.15 Normalized State Error Squared. Case 2(f)

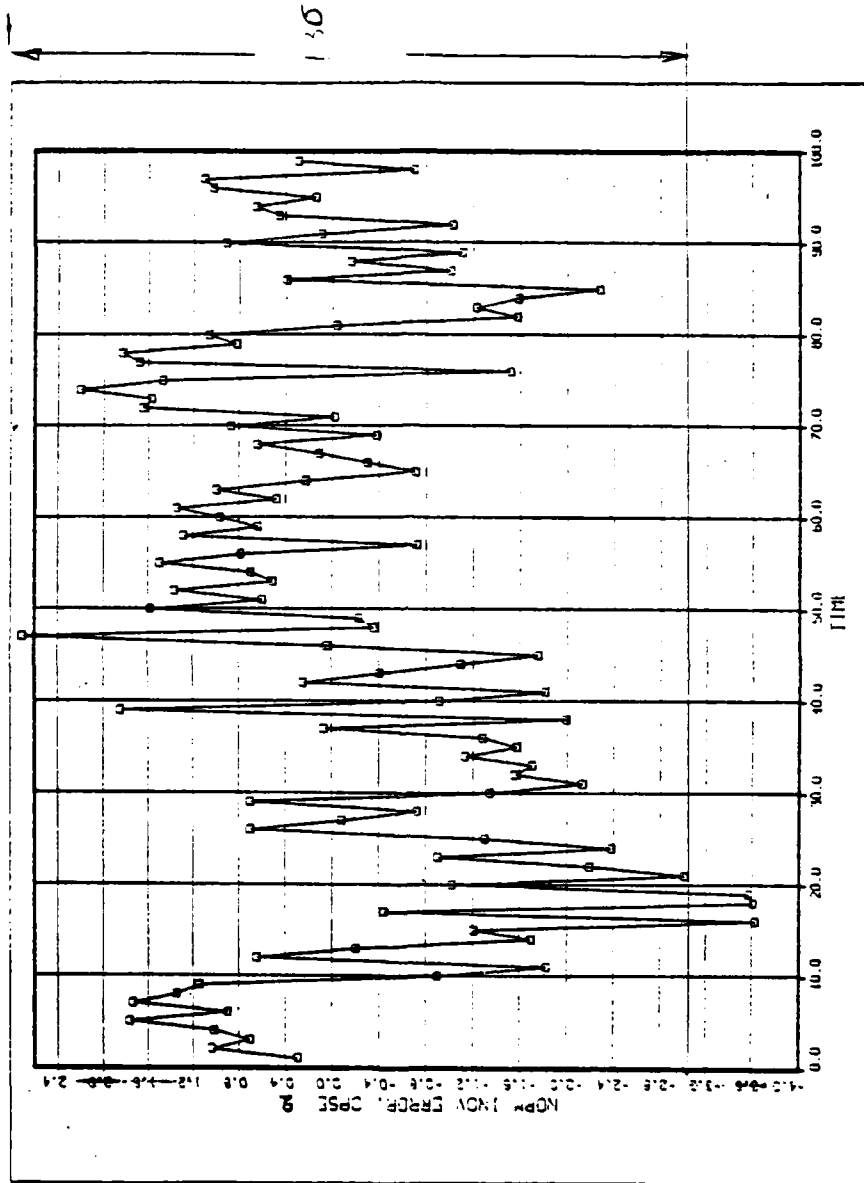


Figure 6.16 Normalized Innovation Error. Case 2(g)

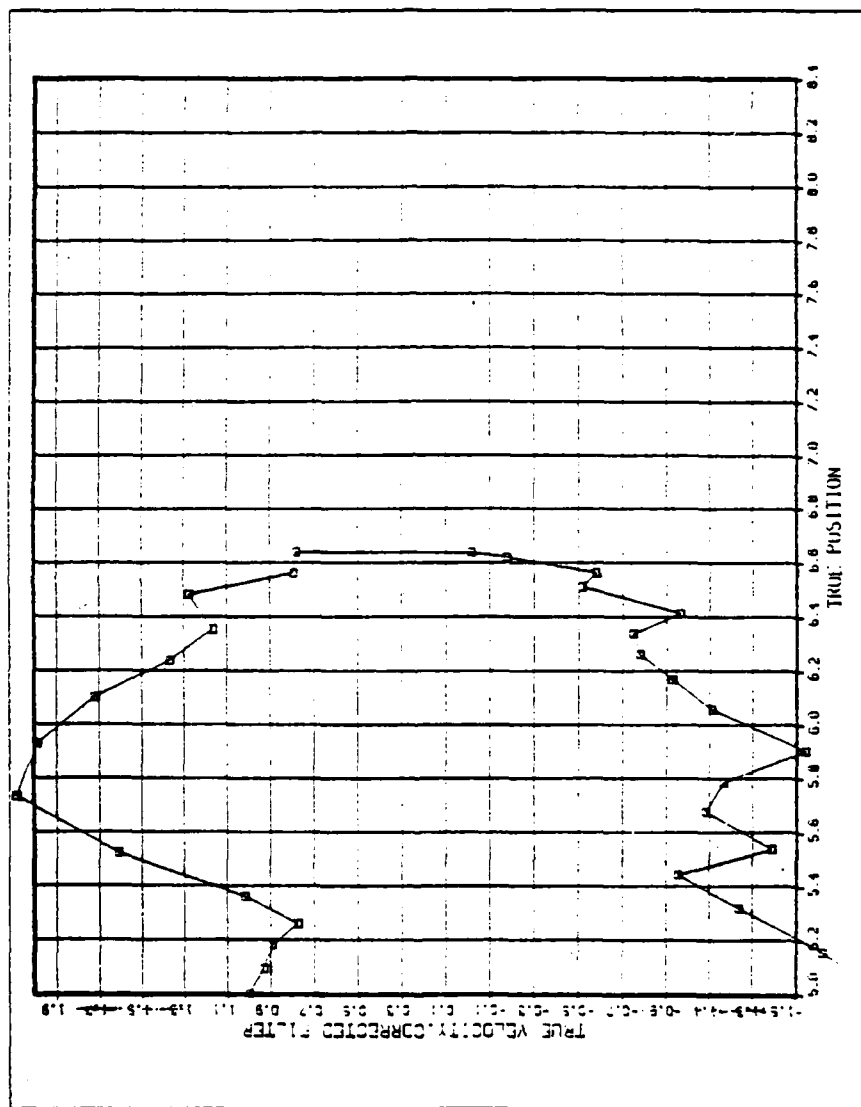


Figure 6.17 True Trajectory in x_1, x_2 Plane. Case 3(a)

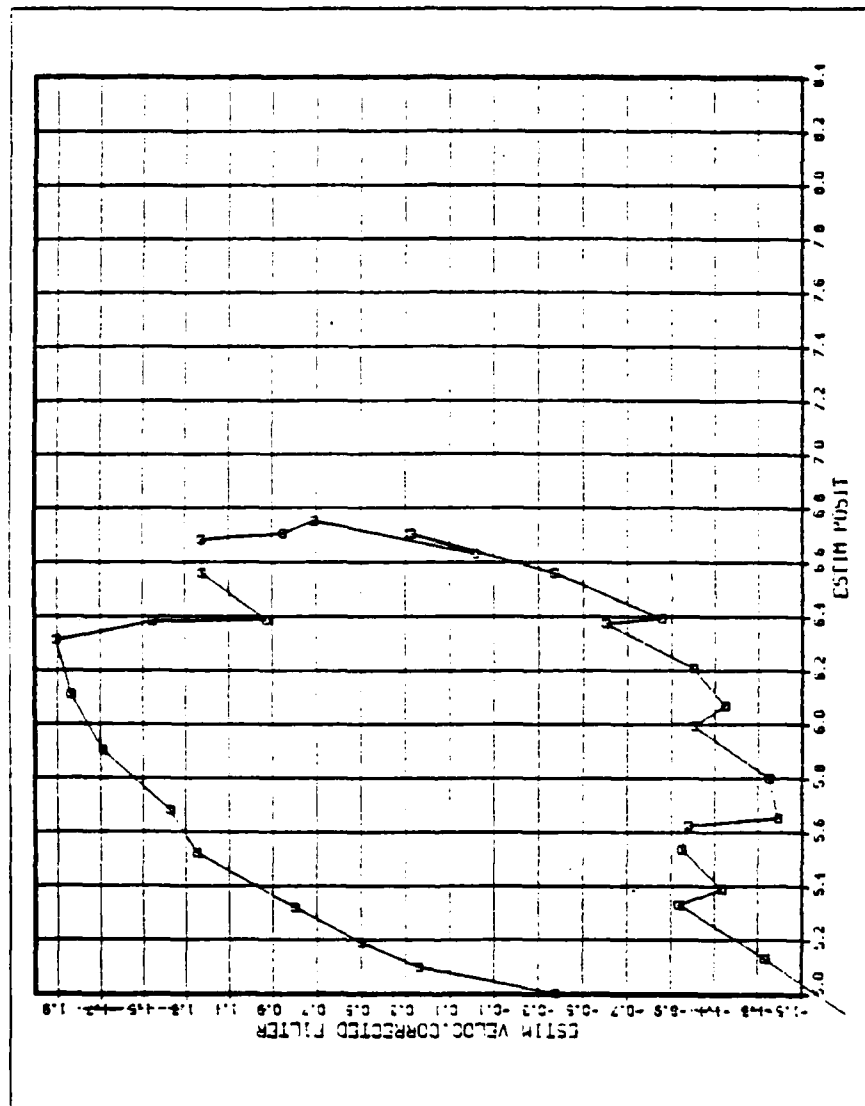


Figure 6.18 Estimated Trajectory in x_1, x_2 Plane. Case 3(a)

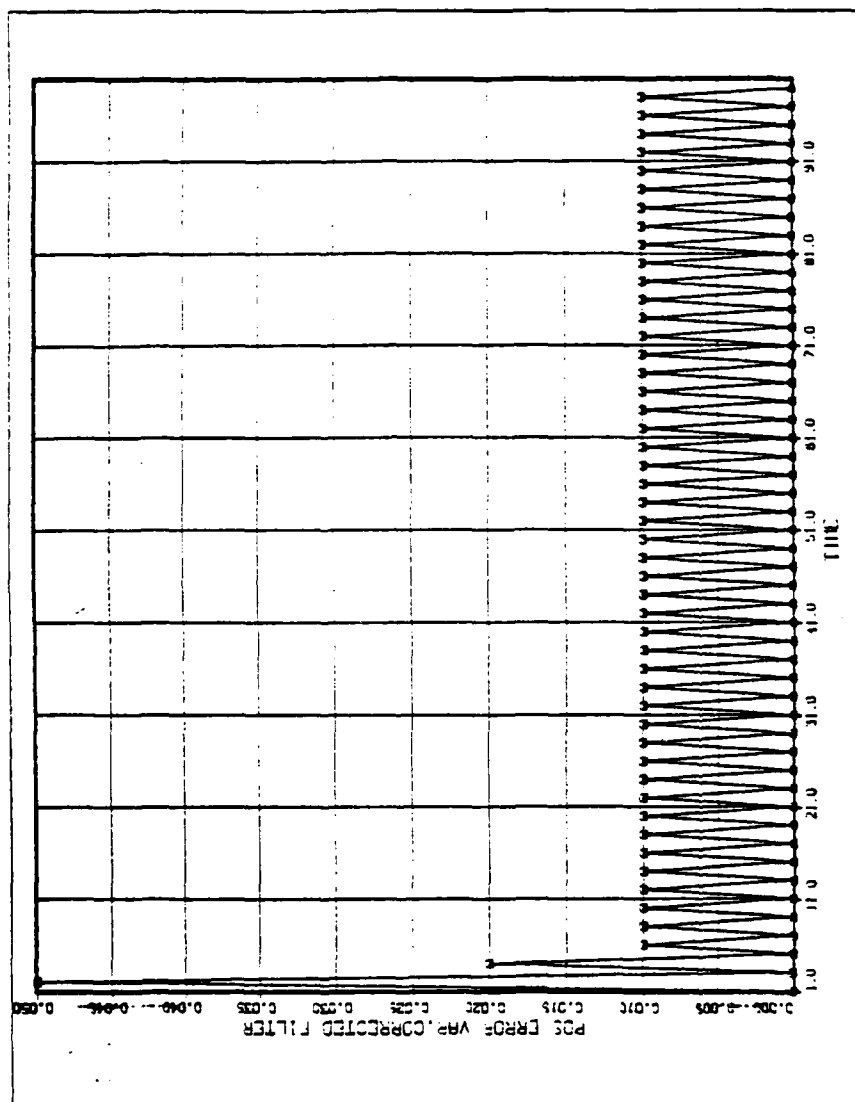


Figure 6.19 Position Error Variance, Case 3(b)

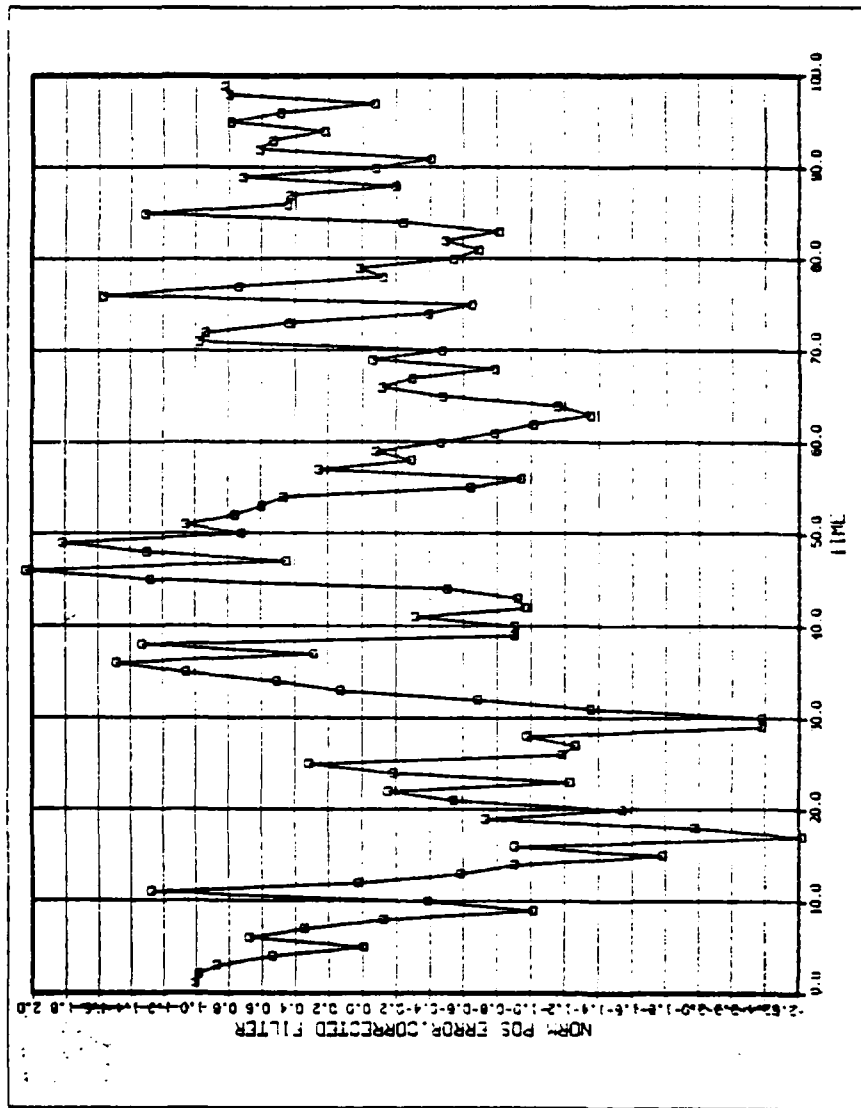


Figure 6.20 Normalized Position Error. Case 3(c)

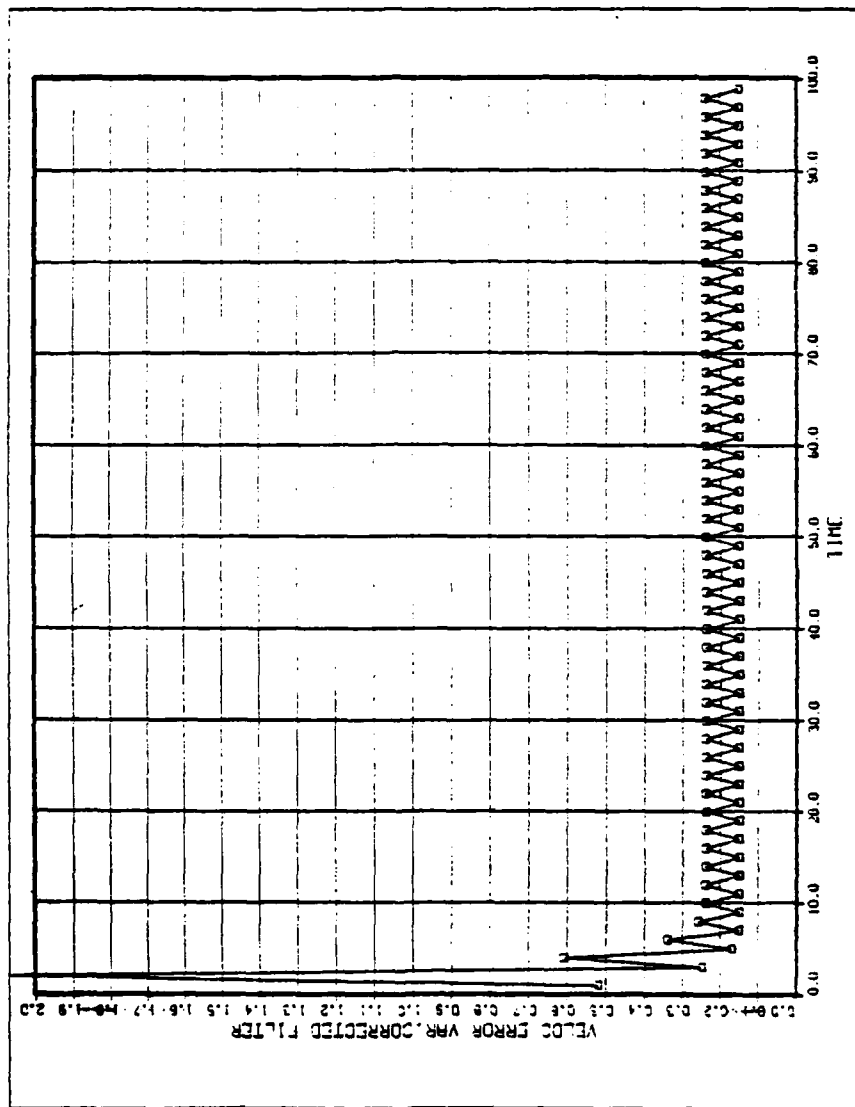


Figure 6.21 Velocity Error Variance. Case 3(d)

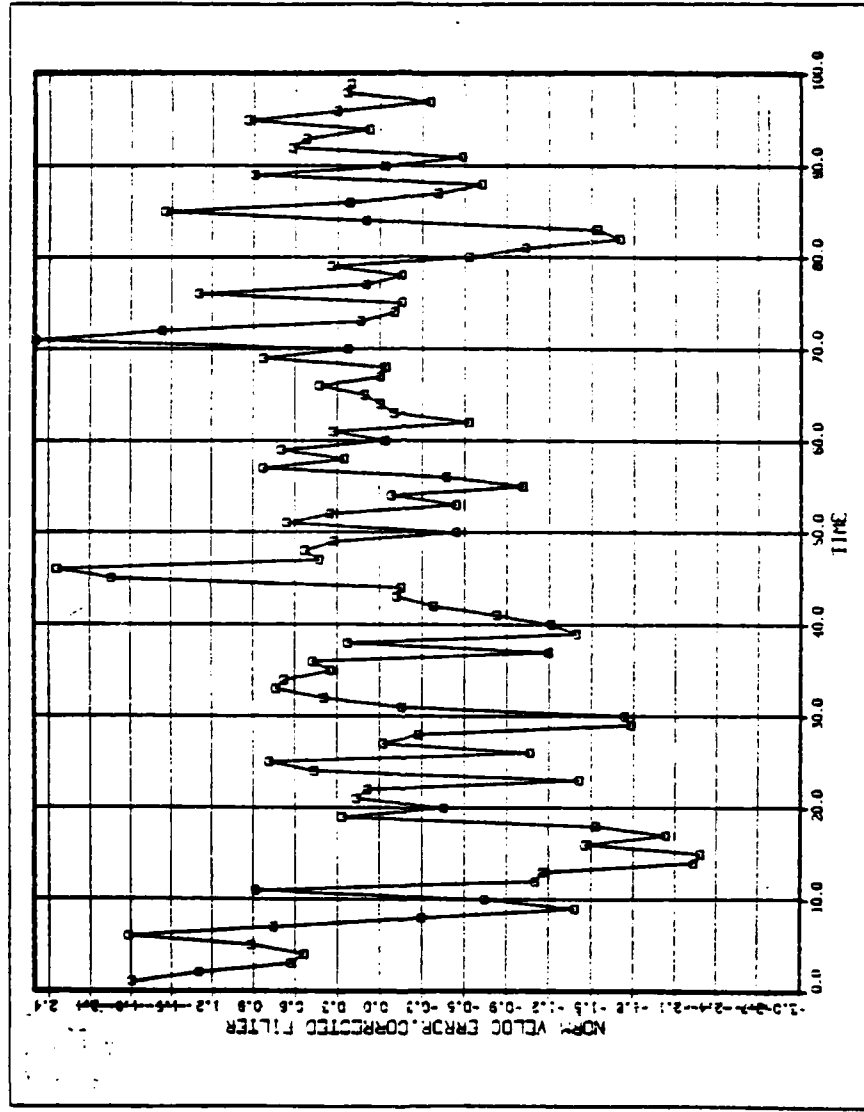


Figure 6.22 Normalized Velocity Error. Case 3(c)

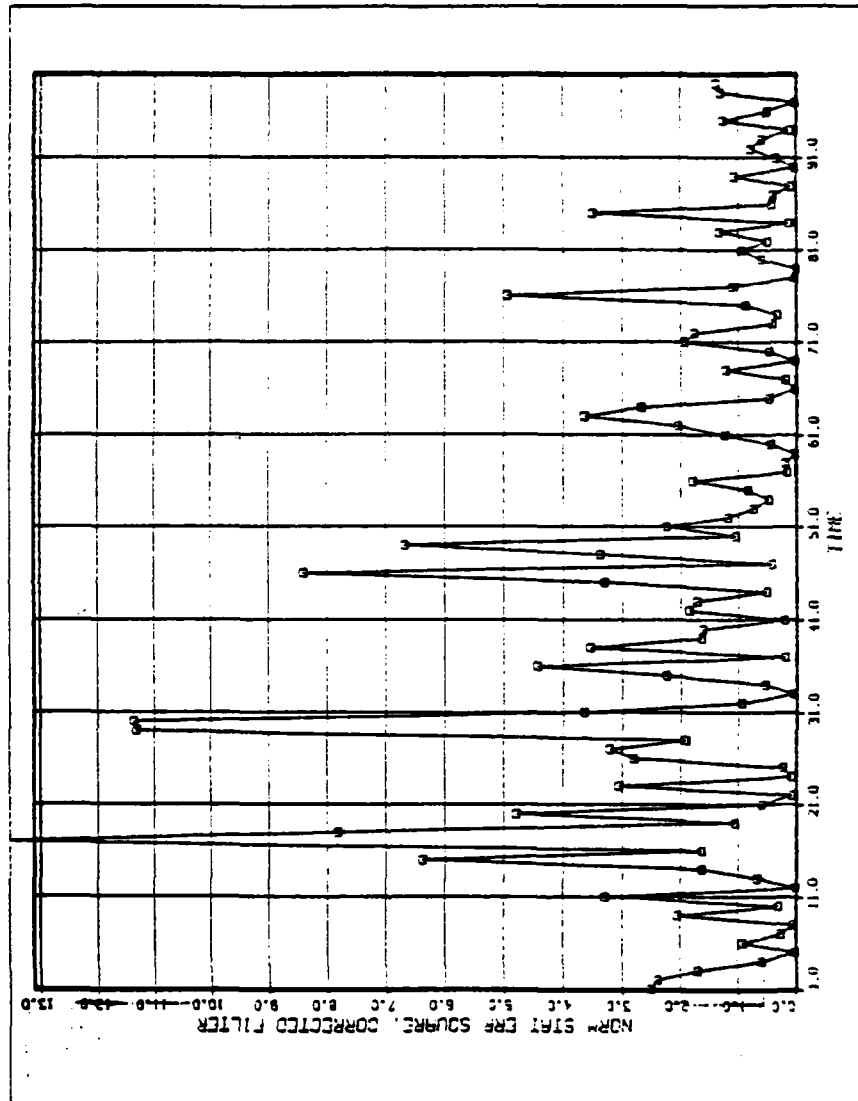


Figure 6.23 Normalized State Error Squared, Case 3(1)

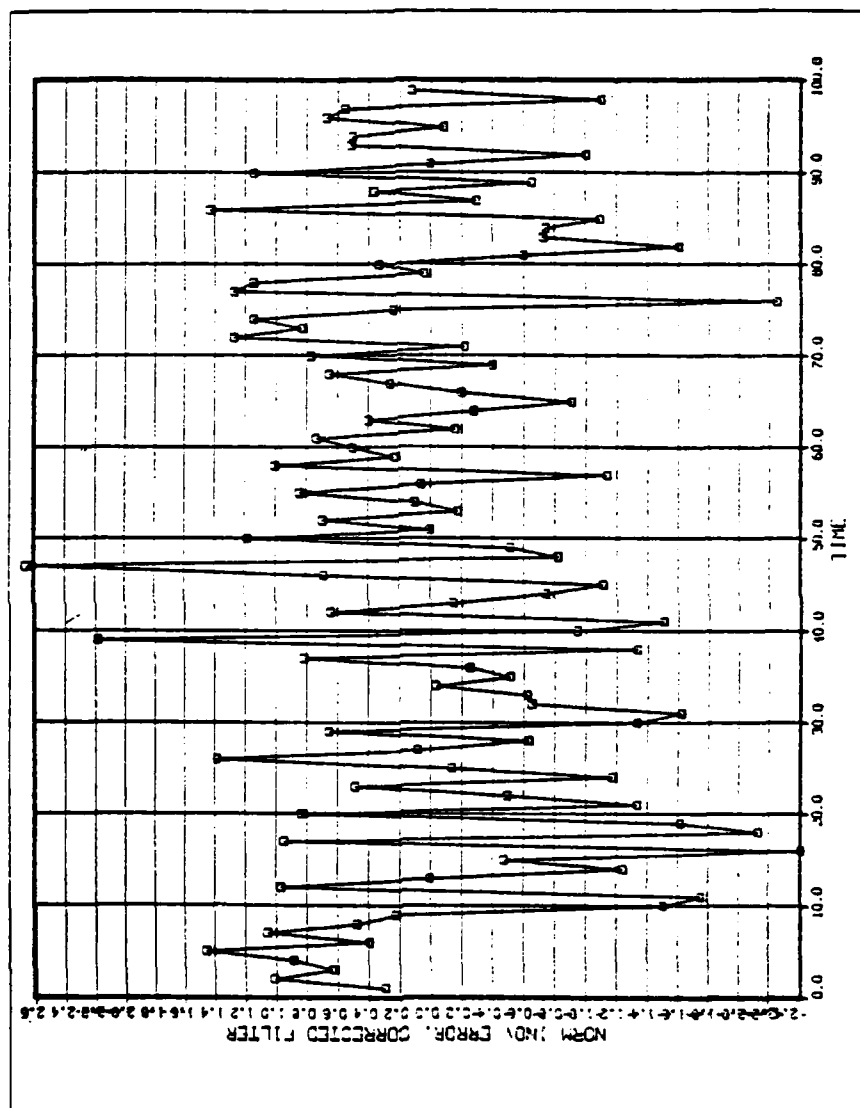


Figure 6.24 Normalized Innovation Error. Case 3(g)

By standardize the normal distribution curve we can have a single curve that may be adapted to all values of the mean as well as differing values of the standard deviation. This standardization may be accomplished by substituting $z = (x - \bar{x})/\sigma$ in Equation (6.35).

The above concept is the case for the subgraphs c, e, g. In Case c, for example,

$$z = [x_1(k) - \hat{x}_1(k, k)]/[P_{11}(k, k)]^{1/2}.$$

The standard normal distribution curve has a mean of zero and variance and standard deviation equal to one.

By integrating Equation (6.35) (after the substitution of z), it is found that:

Between $-1\sigma < z < +1\sigma$, 68.2% of the area under the curve is included.

Between $-2\sigma < z < +2\sigma$, 94.5% of the area under the curve is included.

Between $-3\sigma < z < +3\sigma$, 99.74% of the area under the curve is included.

With the above in mind, the probability that the normalized position, velocity and innovation errors lie between $\pm 2\sigma$ is 94.5% (or between $\pm 3\sigma$ is 99.74%).

From the graphs it is seen that in the two cases where the value of Q is common for the model and the

filter, the three normalized errors lie 100% between $\pm 3\sigma$.

In the case where Q is different between the model and the filter it is seen that these errors exceed at some points $\pm 3\sigma$. But generally it seems that the law of 99.74% between $\pm 3\sigma$ is applied satisfactory.

b. True Trajectory vs. Estimated Trajectory
(expr. a)

(1) $Q = 0.02$ for Model and Filter (pp. 66-67).

There is considerable difference between true and estimated curve at the beginning but as time progresses, there is a continuous improvement. Near the end, the two curves nearly coincide.

(2) $Q = 0.02$ for the Filter and $Q = 0.09$ for the Model (pp. 74-75). There is a significant difference between true and estimated curve at the beginning which is slightly improved at the end of the curves.

(3) $Q = 0.09$ for the Model and Filter (pp. 82-83). Same as in (1), with the only difference that at the end of the curves, the improvement is not nearly a coincidence but it is better than that of Case (2).

Generally the rate of the improvement is greater at small values of time because the gain is inversely proportional to time ($G(k) = 1/k+1$) and it weights the correction term $[z(k) - C(k)\hat{x}(k|k-1)]$ less heavily as time progresses.

The fact that the improvement between true and estimated curves is less in Case (3) than that of Case (1),

was expected. The reason is that Q in Case (3) has a greater value and as it is known, Q increases the uncertainty in the one step prediction:

$$\underline{P}(k|k-1) = \underline{P}(k-1|k-1) + Q \quad (6.36)$$

This affects also the $P(k|k)$:

$$\underline{P}(k|k) = [I - G(k)C(k)]\underline{P}(k|k-1) \quad (6.37)$$

The fact that the estimated curve in Case (2) differs from the true curve more than that of Cases (1) and (3) and also that the improvement is not so significant as in the other cases was also expected. The reason is that the Q 's of the mode-and the filter are different, so this difference affects the P and G of the filter negatively, resulting in a difficulty in approaching the true trajectories.

c. Normalized State Error Squared Error (expr. f)

It is known from the theory that $(\underline{x} - \bar{\underline{x}})'P^{-1}(\underline{x} - \bar{\underline{x}})$ is the sum of the squares of n independent zero-mean, unity variance Gaussian random variable, i.e., $N(0,1)$. This means that $(\underline{x} - \bar{\underline{x}})'P^{-1}(\underline{x} - \bar{\underline{x}})$ has a chi-square distribution with n degrees of freedom (n is the dimension of vector $\underline{x}$).

If the chi-square distribution (χ^2) equals zero, the true and estimated state vectors agree exactly. The larger the value of χ^2 the greater the discrepancy between the true and estimated state vectors.

What is expected to be seen in this problem is a greater discrepancy in the case where the Q's of the model and the filter are different, due to the poor estimations of the state vectors.

The above expectation seems to be the case. Specifically in the cases where the same Q for the filter and the model is used, it can be said roughly that the $\chi^2_{.025}$ and $\chi^2_{.975}$ are approximately 0.06 and 7.5 as they should be for two degrees of freedom (see proper tables).

d. Position and Velocity Error Variances (expr. b,d)

In Cases 1 and 2 (filter has $Q = 0.02$), we have identical position error variances and identical velocity error variances. After some initial variations the position error variance takes its steady state which is almost zero (expected), while the velocity error variance takes also its steady state which is approximately 0.04.

In Case 3, the increase in Q, increases $P_{11}(k|k-1)$ and reduces $P_{11}(k|k)$ (see proper equations). Then the steady state values of $P_{11}(k|k-1)$ and $P_{11}(k|k)$ are different and the result is that the position error variance oscillates between the values 0.0 and 0.01. For the same reason the velocity error oscillates between the values 0.15 and 0.24. The mean values of both then are greater than the corresponding values of Cases 1, 2. This is logical since now the Q is greater as it was derived:

Position Error Variance:

$$\underline{P}_{11}(k|k) = \underline{P}_{11}(k|k-1) - \underline{G}_1(k)\underline{P}_{11}(k|k-1) \quad (6.38)$$

$$\begin{aligned} \underline{P}_{11}(k|k-1) &= \underline{P}_{11}(k-1|k-1) + \underline{TP}_{21}(k-1|k-1) \\ &\quad + \underline{TP}_{12}(k-1|k-1) + T^2 \underline{P}_{22}(k-1|k-1) + T^2 \underline{Q} \end{aligned} \quad (6.39)$$

Velocity Error Variance:

$$\underline{P}_{22}(k|k) = -\underline{G}_2(k)\underline{P}_{12}(k|k-1) + \underline{P}_{22}(k|k-1) \quad (6.40)$$

$$\underline{P}_{22}(k|k-1) = \underline{P}_{22}(k-1|k-1) + \underline{Q} \quad (6.41)$$

So $\underline{Q}$ is proportional to $\underline{P}_{11}$ and $\underline{P}_{22}$.

B. AN EXAMPLE USING E.K.F. (NON-LINEAR CASE)

1. Problem Description

A stationary target is located at $x_1 = x_2 = 100$. Bearing measurements from a moving ship are taken at $k = 1, \dots, N$ from locations as indicated.

The following are given:

$$\underline{z}(k) = \underline{\theta}(k) + \underline{v}(k) \quad (6.42)$$

$$\theta(k) = \tan^{-1} \frac{x_2}{x_1 - a(k)} \quad k = 1, \dots, N \quad (6.43)$$

$$E[v(k)] = 0 \quad (6.44)$$

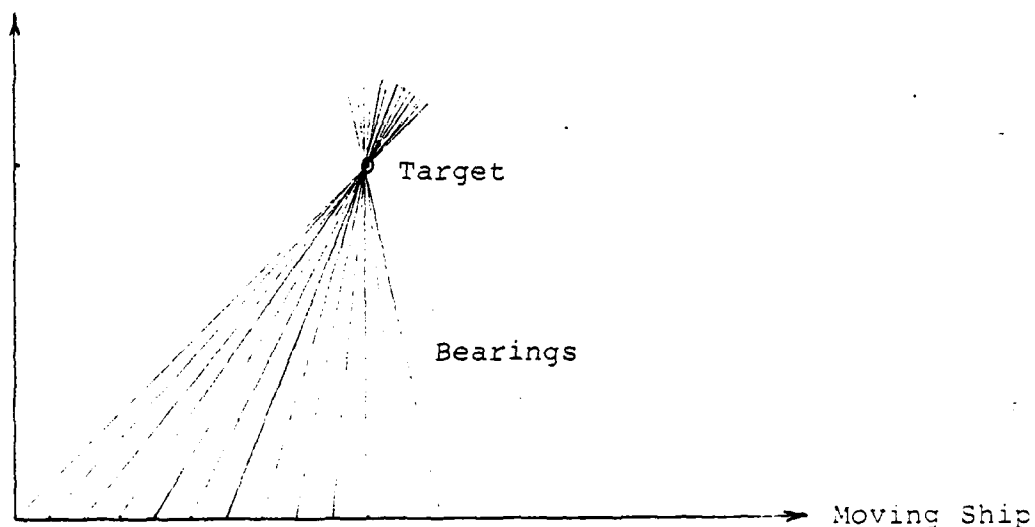


Figure 6.25 Descriptive Diagram for the Non-linear Problem

$$E[v(k)v(j)] = \sigma^2 \delta_{kj} \quad (6.45)$$

$$a(k) = 10k \quad \sigma^2 = (2^0)^2 \quad N = 12 \quad (6.46)$$

The initial estimate is:

$$\hat{x}(0|0) = \begin{bmatrix} \hat{x}_1(0|0) \\ \hat{x}_2(0|0) \end{bmatrix} = \begin{bmatrix} 80 \\ 120 \end{bmatrix} \quad (6.47)$$

$$P(0|0) = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix} \quad (6.48)$$

Based on the bearings of the moving ship, an Extended Kalman Filter can be used to improve the initial estimate of the target's position.

AD-A150 770

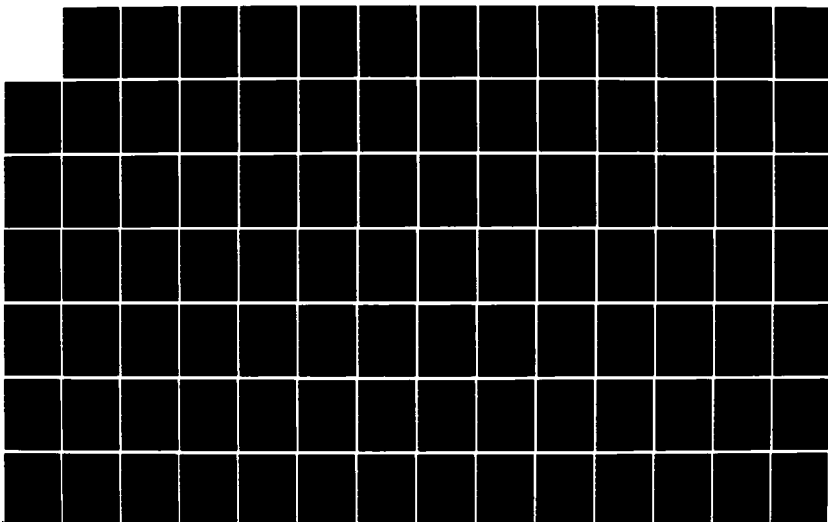
DIGITAL COMPUTER APPLICATIONS OF KALMAN FILTER IN
TARGET TRACKING(U) NAVAL POSTGRADUATE SCHOOL MONTEREY
CA V I MARTZOUCOS JUN 84

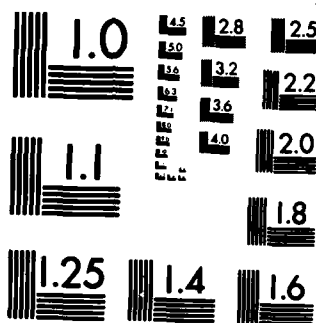
2/3

UNCLASSIFIED

F/G 9/2

NL





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

It is useful to obtain the values of $\hat{x}(N|N)$ and $\hat{P}(N|N)$ for observing the quality of the simulation.

2. Analysis

Since the target is stationary:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \quad (6.49)$$

The model equations for non-linear problems are:

$$\underline{x}(k+1) = \underline{a}(\underline{x}(k), \underline{u}(k), k) + \underline{w}(k) \quad (6.50)$$

$$\underline{z}(k) = \underline{c}(\underline{x}(k)) + \underline{v}(k) \quad (6.51)$$

For this problem

$$\underline{a}(\underline{x}(k), \underline{u}(k), k) = \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \quad (6.52)$$

so

$$A(k) = \left. \frac{\partial \underline{a}}{\partial \underline{x}} \right|_{(\underline{x}(k|k))} = \begin{bmatrix} \frac{\partial x_1}{\partial x_1} & \frac{\partial x_1}{\partial x_2} \\ \frac{\partial x_2}{\partial x_1} & \frac{\partial x_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (6.53)$$

$\underline{c}(\underline{x}(k))$ for this problem is given by:

$$\underline{c}(\underline{x}(k)) = \theta = \tan^{-1} \frac{x_2}{x_1 - a(k)} \quad (6.54)$$

so

$$H = \left. \frac{\partial \underline{c}}{\partial \underline{x}} \right|_{\underline{x}(k)} = \left. \frac{\partial \theta}{\partial \underline{x}} \right|_{\hat{\underline{x}}(k|k-1)} \quad (6.55)$$

and

$$H_1 = \frac{-x_2}{(x_1 - a(k))^2 + x_2^2} \quad (6.56)$$

$$H_2 = \frac{x_1 - a(k)}{(x_1 - a(k))^2 + x_2^2} \quad (6.57)$$

To calculate the gains of the filter, the following equation is used:

$$\underline{G}(k) = \underline{P}(k|k-1) \underline{H}'(k)$$

$$[\underline{H}(k) \underline{P}(k|k-1) \underline{H}'(k) + \underline{R}(k)]^{-1} \quad (6.58)$$

After some manipulations of the above equation in its matrix form, it is found that the gains are given by the following equations:

$$G_1(k) = \frac{P_{11}(k|k-1)H_1(k) + P_{12}(k|k-1)H_2(k)}{H_1^2(k)P_{11}(k|k-1) + H_1(k)H_2(k)[P_{21}(k|k-1) + P_{12}(k|k-1)] + H_2^2(k)P_{22}(k|k-1) + R(k)} \quad (6.59)$$

$$G_2(k) = \frac{P_{21}(k, k-1)H_1(k) + P_{22}(k, k-1)H_2(k)}{H_1^2(k)P_{11}(k|k-1) + H_1(k)H_2(k)[P_{21}(k|k-1) + P_{12}(k|k-1)] + H_2^2(k)P_{22}(k|k-1) + R(k)} \quad (6.60)$$

The estimation of the states is obtained by manipulating the following equation:

$$\hat{\underline{x}}(k|k) = \hat{\underline{x}}(k|k-1) + \underline{G}(k)[\underline{z}(k) - \underline{c}(\hat{\underline{x}}(k|k-1))] \quad (6.61)$$

The results are:

$$\hat{x}_1(k|k) = \hat{x}_1(k|k-1) + G_1(k)z(k) - G_1(k)\tan^{-1} \frac{\hat{x}_2(k|k-1)}{\hat{x}_1(k|k-1) - a(k)} \quad (6.62)$$

$$\hat{x}_2(k|k) = \hat{x}_2(k|k-1) + G_2(k)z(k) - G_2(k)\tan^{-1} \frac{\hat{x}_2(k|k-1)}{\hat{x}_1(k|k-1) - a(k)} \quad (6.63)$$

The error covariance matrix during the filtering phase of the Kalman Filter is given by the equation:

$$\underline{P}(k|k) = [\underline{I} \quad \underline{G}(k)\underline{H}(k)]\underline{P}(k|k-1) \quad (6.64)$$

By manipulating this equation as it was described previously, the following equations are obtained:

$$P_{11}(k|k) = P_{11}(k|k-1)[1 - G_1(k)H_1(k)] - P_{21}(k|k-1)G_1(k)H_2(k) \quad (6.65)$$

$$P_{12}(k|k) = [1 - G_1(k)H_1(k)]P_{12}(k|k-1) - G_1(k)H_2(k)P_{22}(k|k-1) \quad (6.66)$$

$$P_{21}(k|k) = [1 - G_2(k)H_2(k)]P_{21}(k|k-1) - G_2(k)H_1(k)P_{11}(k|k-1) \quad (6.67)$$

$$P_{22}(k|k) = [1 - G_2(k)H_2(k)]P_{22}(k|k-1) - G_2(k)H_1(k)P_{12}(k|k-1) \quad (6.68)$$

The error covariance matrix during the prediction phase of the Kalman Filter is given by the equation:

$$\underline{P}(k|k-1) = \underline{A}(k-1)\underline{P}(k-1|k-1)\underline{A}'(k-1) + \underline{Q}(k-1) \quad (6.69)$$

This yields the following results:

$$P_{11}(k|k-1) = P_{11}(k-1|k-1) \quad (6.70)$$

$$P_{12}(k|k-1) = P_{12}(k-1|k-1) \quad (6.71)$$

$$P_{21}(k|k-1) = P_{21}(k-1|k-1) \quad (6.72)$$

$$P_{22}(k, k-1) = P_{22}(k-1, k-1) \quad (6.73)$$

The normalized state error squared:

$$[\underline{x}(k) - \hat{\underline{x}}(k|k)]' \underline{P}^{-1}(k) [\underline{x}(k) - \hat{\underline{x}}(k|k)]$$

is given by the same expression as in the previous problem.

All the above results are used in a computer program (see Appendix A). This program basically follows the steps of an Extended Kalman Filter algorithm as it was described in Chapter III.

From the outputs it can be seen that the position variance on the horizontal axis is smaller than that on the vertical axis. This was expected since the bearings are crossed with relatively small angles and in this way, there is a larger uncertainty on the vertical axis.

Also from the results, it can be seen that the estimates for x_1 , x_2 , are improved as time goes on and finally, that they are very close to the real values.

C. MANEUVERING TARGET EXAMPLE

1. Problem Description

The target is modeled by the following equation:

$$\underline{x}(k+1) = \underline{F} \underline{x}(k) + \underline{G} \underline{w}(k) \quad (6.74)$$

 NON-LINEAR EXAMPLE RUN
 DSREQ = 32422
 ERROR COVARIANCE MATRIX ELEMENTS

P11(KIK)	P12(KIK)	P21(KIK)	P22(KIK)
100.00000	0.0	0.0	100.00000
39.55396	35.23686	35.23686	79.444514
37.55565	40.01921	40.01921	09.66414
34.47598	41.62045	41.62045	08.56604
32.06934	41.50189	41.50189	08.56154
28.06145	39.11339	39.11339	07.10515
22.77173	34.12053	34.12053	02.42247
17.48499	27.96136	27.96136	55.27096
12.34488	20.93508	20.93508	45.00004
8.35587	14.79344	14.79344	30.21820
5.55947	10.01808	10.01808	28.00339
3.60768	0.28326	0.28326	20.91670

 X1ESTK(1), X2ESTK(1): ESTIM. POSITION OF TARGET
 TH1A: BEARING ANGLE
 NSTESQ(1): NORMALIZED STATE ERROR SQUARED

X1ESTK(I)	X2ESTK(I)	TH1A	NSTESQ(I)
1	80.00000	0.23798	0.00000
2	102.13861	0.25006	0.67407
3	102.14719	0.96907	0.97375
4	103.65202	1.02036	0.23093
5	104.20103	1.10715	0.27533
6	106.25463	1.15029	0.25340
7	106.32711	1.47534	1.74208
8	105.41669	1.53540	1.53026
9	105.42439	1.47113	4.63523
10	105.05821	1.27060	3.72753
11	102.53960	1.60047	2.04271
12	100.40462	1.76813	0.70650

```

*****
**NON-LINEAR EXAMPLE RUN
**DSEED = 1111
**ERROR CCVARIANCE MATRIX ELEMENTS
*****
P11(KIK) P12(KIK) P21(KIK) P22(KIK)
-----
1 100.00000 35.23680 0.0 100.00000
2 35.59396 39.81206 35.23680 179.44514
3 37.74453 41.53508 39.81206 68.12425
4 35.09048 41.21964 41.53508 66.67914
5 32.58984 39.15721 41.21964 66.04700
6 28.79883 33.95612 39.15721 65.52493
7 23.19322 33.95612 33.95612 60.69914
8 16.22414 25.60390 25.60390 50.68945
9 10.37245 17.43547 17.43547 39.28093
10 6.25171 10.67679 10.67679 28.34793
11 3.97121 6.83913 6.83913 21.09922
12 2.74282 4.44173 4.44173 17.02052
*****
***** ESTIM. POSITION OF THE TARGET *****
X1ESTK(I),X2ESTK(I): ESTIM. POSITION OF THE TARGET
THETA: BEARING ANGLE
NSTESG(I): NORMALIZED STATE ERROR SQUARES
*****
X1ESTK(I) X2ESTK(I) THETA NSTESG(I)
-----
1 80.00000 120.00000 0.83798 8.00000
2 194.50216 105.70705 0.85006 0.57236
3 104.47229 105.78101 0.50007 0.57321
4 104.25208 105.93213 1.03038 0.56404
5 106.36147 106.14467 1.10715 1.50553
6 104.89732 105.32094 1.15029 0.59959
7 101.45752 102.15750 1.47534 0.59169
8 95.81422 100.18760 1.54540 0.57481
9 97.82155 97.40663 1.47113 0.57402
10 58.04329 57.75854 1.57080 0.57407
11 55.12383 59.67206 1.60347 0.57383
12 100.56055 102.51585 1.72019 0.57349

```

This equation is discretized over time intervals of length T . Using Cartesian coordinates, the state is:

$$\underline{x} = [\underline{x} \quad \dot{\underline{x}} \quad \underline{y} \quad \dot{\underline{y}}]' \quad (6.75)$$

and

$$\underline{F} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.76)$$

$$\underline{w} = [\underline{w}_1 \quad \underline{w}_2]' \quad (6.77)$$

$$\underline{G} = \begin{bmatrix} T/2 & 0 \\ 1 & 0 \\ 0 & T/2 \\ 0 & 1 \end{bmatrix} \quad (6.78)$$

$$E[\underline{w}(k)] = 0; \quad E[\underline{w}(k) \underline{w}'(j)] = \underline{Q} \delta_{kj} \quad (6.79)$$

The initial estimate is $\hat{\underline{x}}(0|0)$ with covariance $P(0|0)$.

It is assumed that only position measurements are available:

$$\underline{z}(k) = \underline{C} \underline{x}(k) + \underline{v}(k) \quad (6.80)$$

where

$$\underline{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (6.81)$$

$$E[\underline{v}(k)] = 0; \quad E[\underline{v}(k) \underline{v}'(j)] = \underline{R} \delta_{kj} \quad (6.82)$$

The following are given:

$$T = 10s, \quad Q = 0, \quad R_{11} = R_{22} = 10^4 m^2, \quad \text{and}$$

$$R_{12} = 500m^2.$$

The initial conditions of the target are:

$$x(0) = 200m, \quad \dot{x}(0) = 0, \quad y(0) = 10,000m, \quad \dot{y}(0) = -15m/s.$$

The target is on a constant course and speed until time $t = 400s$, when it maneuvers a slow 90° turn in the x direction with acceleration inputs $u^x = u^y = 0.075m/s^2$. It completes the turn at $t = 600s$ (after 20 sampling times) and from then on the accelerations are zero. The second turn, also of 90° , is fast. It starts at $t = 610s$ with acceleration of $0.3m/s^2$ and is completed at $t = 660s$, i.e., after 5 sampling times.

A simulation of the maneuvering target can be done in x coordinate only (the results for the y coordinate are similar), using the method of "White noise model with adjustable level."

The quality of the performance of the target's simulation can be observed from the following plots:

- a. True and estimated trajectories of the target.
- b. Monte Carlo runs of position root mean square error (r.m.s.e.):

$$\left\{ \frac{1}{50} \sum_{i=1}^{50} [\tilde{x}_1^{(i)}(k|k)]^2 \right\}^{1/2}$$

- c. The same as b. for the velocity r.m.s. error.

2. Analysis

The target's movement is presented in Figure 6.26.

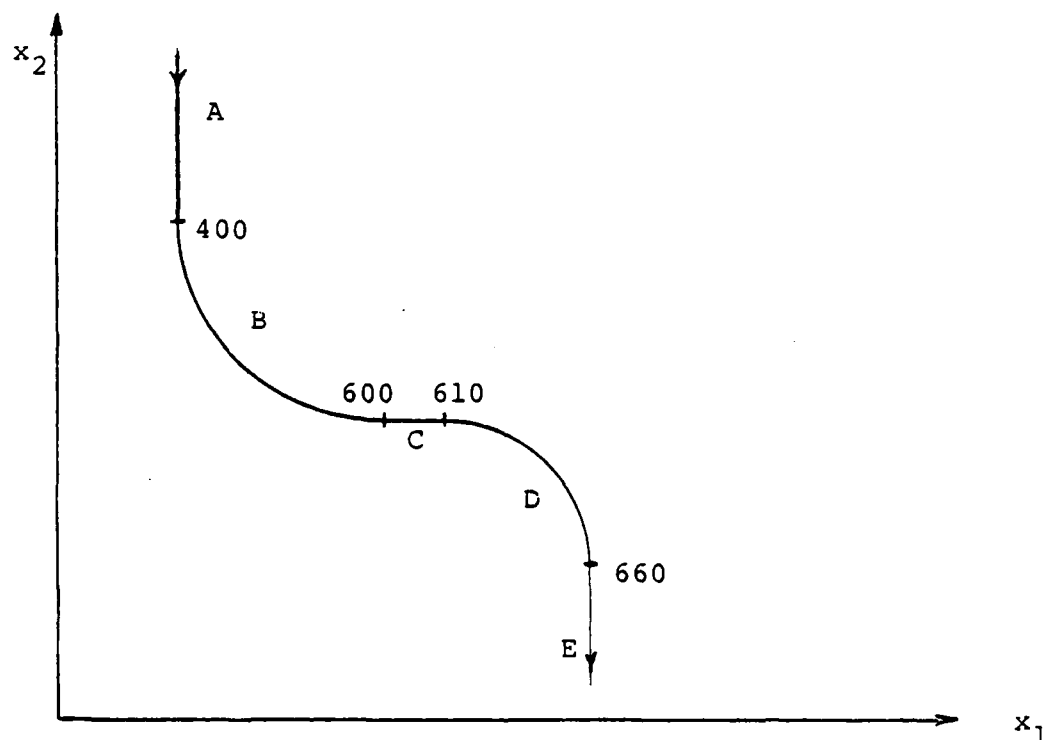


Figure 6.26 Movement of the Manuevering Target

For treating this problem, a linear Kalman Filter can be used. The only difference is that the actual trajectory of the target is now given by several formulas instead of one. The different phases of movement of the target A,B, C,D,E, can be formed in proper mathematical formulas, which can feed the position measurements one after the other, depending on the sampling time.

The initial estimates of the states in the x direction were obtained as:

$$\hat{x}(0|0) = \hat{x}_1(0|0) = z_1(0) \quad (6.83)$$

$$\hat{\dot{x}}(0|0) = \hat{\dot{x}}_2(0|0) = [z_1(0) - z_1(-1)]/T \quad (6.84)$$

where

$$z_1(-1) = x(0) - \dot{x}(0)T + v_1(-1) \quad (6.85)$$

$$z_1(0) = x(0) + v_1(0) \quad (6.86)$$

$$v \sim N(0, R_{11}) \quad (6.87)$$

The initial covariance matrix is then:

$$P(0|0) = \begin{bmatrix} R_{11} & R_{11}/T \\ R_{11}/T & 2R_{11}/T^2 \end{bmatrix} \quad (6.88)$$

The rest of the calculations needed for developing the Kalman Filter algorithm are very similar to those developed in the first example.

If the maneuvers are not taken into account and the filter runs as usual, the tracking results are very poor during the maneuvers (see pp. 114-115).

Using the "White noise model with adjustable level" method, a self-adjustment of the filter can be accomplished as follows:

After the square norm of innovations:

$$\underline{\varepsilon}_V(k) = \underline{V}'(k) S^{-1} \underline{V}(k)$$

exceeds a chosen value, the target is considered as maneuvering and a change (increase) in Q , is made. The increase in Q increases the gain G , so the state estimation:

$$\hat{\underline{x}}(k|k) = \hat{\underline{x}}(k|k-1) + \underline{G}(k) [\underline{z}(k) - \underline{c}(k) \hat{\underline{x}}(k|k-1)] \quad (6.89)$$

is more dependent on the second term since the state estimation $\hat{\underline{x}}(k|k-1)$ is less reliable during the maneuver. The filter continues its run with the new value of Q' until ε_V has again a lower value than that of the chosen threshold. Then the filter runs again with the given initial value of Q ($Q = 0.0$).

After the addition of the above provision in the tracking computer program, the results were much better than those of the non-adjustable filter.

Three categories of results were obtained. In each one, different values of Q's and threshold levels were used. The three categories are:

- a. Observed and estimated trajectories versus sampling time (50 Monte Carlo runs).
- b. Observed and estimated trajectories versus sampling time (single runs).
- c. Mean square position error (50 Monte Carlo runs).

The above results (plots) can be seen on pp. 116-149. A tabulation and qualification of these results has been developed as follows:

Symbols

S: Constant course and speed

1M: First Maneuver

2M: Second maneuver

Very Good: The estimated trajectory follows the time trajectory with great reliability (coincidence of the paths).

Good: The estimated trajectory is still very reliable, i.e., follows the true path closely but they don't coincide exactly.

F. Noise: The estimated trajectory follows the noise.

F. Noise E.: The estimated trajectory follows the noise exactly.

Deviation: There is deviation between the true and estimated trajectory.

S. Deviation: There is strong deviation between the true and estimated trajectory.

TABLE II
Observed and Estimated Trajectories
(50 Monte Carlo Runs)

Q	THR.		0.1	3	20
0.1	S		Very Good	Very Good	Very Good
	1M		Deviation	S. Deviation	S. Deviation
	2M		S. Deviation	S. Deviation	S. Deviation
3	S		Very Good	Very Good	Good
	1M		Good	Deviation	Deviation
	2M		Deviation	Deviation	Deviation
10	S		Very Good	Very Good	Good
	1M		Very Good	Good	Deviation
	2M		Good	Deviation	Deviation
100	S		Very Good	Very Good	Good
	1M		Very Good	Good	Deviation
	2M		Good	Good	Deviation
1000	S		Very Good	Very Good	Deviation
	1M		Very Good	Very Good	Deviation--F. Noise
	2M		Very Good	Good	Deviation--F. Noise

TABLE III
Observed and Estimated Trajectories
(Single Runs)

Q	THR.		0.1	3	20
0.1	S		Very Good	Very Good	Very Good
	1M		Deviation	S. Deviation	S. Deviation
	2M		S. Deviation	S. Deviation	S. Deviation
3	S		F. Noise	Very Good	Very Good
	1M		Good	Good	Deviation
	2M		Good	Deviation	Deviation
10	S		F. Noise E.	F. Noise	Good
	1M		Good	Good	Deviation
	2M		Good	Good	Deviation
100	S		F. Noise E.	F. Noise	Good
	1M		F. Noise E.	Good	Deviation--F. Noise
	2M		F. Noise E.	F. Noise	Deviation--F. Noise
1000	S		F. Noise E.	F. Noise	F. Noise
	1M		F. Noise E.	Good	Deviation--F. Noise E.
	2M		F. Noise E.	F. Noise	Deviation--F. Noise E.

Symbols

- 1MAX: Maximum value of m.s.e. during the first maneuver
- 2MAX: Maximum value of m.s.e. during the second maneuver
- 3MIN: Minimum value of m.s.e. during movement under constant course and speed.
- MAN: Maneuvers

TABLE IV

Root Mean Square Position Error (X Direction)

(50 Monte Carlo Runs)

Q	THR.	0.1	3	20
	1MAX	227	370	730
0.1	2MAX	257	300	370
	3MIN	32	30	30
	1MAX	82	147	455
4	2MAX	108	190	365
	3MIN	48	37	30
	1MAX	84	130	425
10	2MAX	92	168	340
	3MIN	52	37	40
	1MAX	102	107	375
100	2MAX	98	135	305
	3MIN	64	46	30
	1MAX	114	102	345
1000	2MAX	110	122	295
	3MIN	72	51	30
	1MAX		102	
3000	2MAX	—	124	
	3MIN		52	—

TABLE V
Conclusions From Tables II, III, IV

		TABLE II	TABLE III	TABLE IV
High Q	S	Good	F. Noise	r.m.s.e.
			F. Noise E.	increases
	MAN	Very Good	F. Noise	r.m.s.e.
			F. Noise E.	decreases
Low Q	S	Very Good	Very Good	r.m.s.e.
				decreases
	MAN	S. Deviation	Deviation	r.m.s.e.
			S. Deviation	increases
High Threshold	S	Deviation	Very Good	r.m.s.e.
			Good	decreases
	MAN	Deviation	Deviation	r.m.s.e.
			S. Deviation	increases
Low Threshold	S	Very Good	F. Noise	r.m.s.e.
			F. Noise E.	increases
	MAN	Very Good	Good	r.m.s.e.
			F. Noise	decreases

OBS. AND ESTIM. TRAJ. NON-ADJUST. FILTER

CONTIN. LINE: MEASUREMENTS TRAJECT.

DASHED LINE: FILTER S ESTIM. TRAJECT.

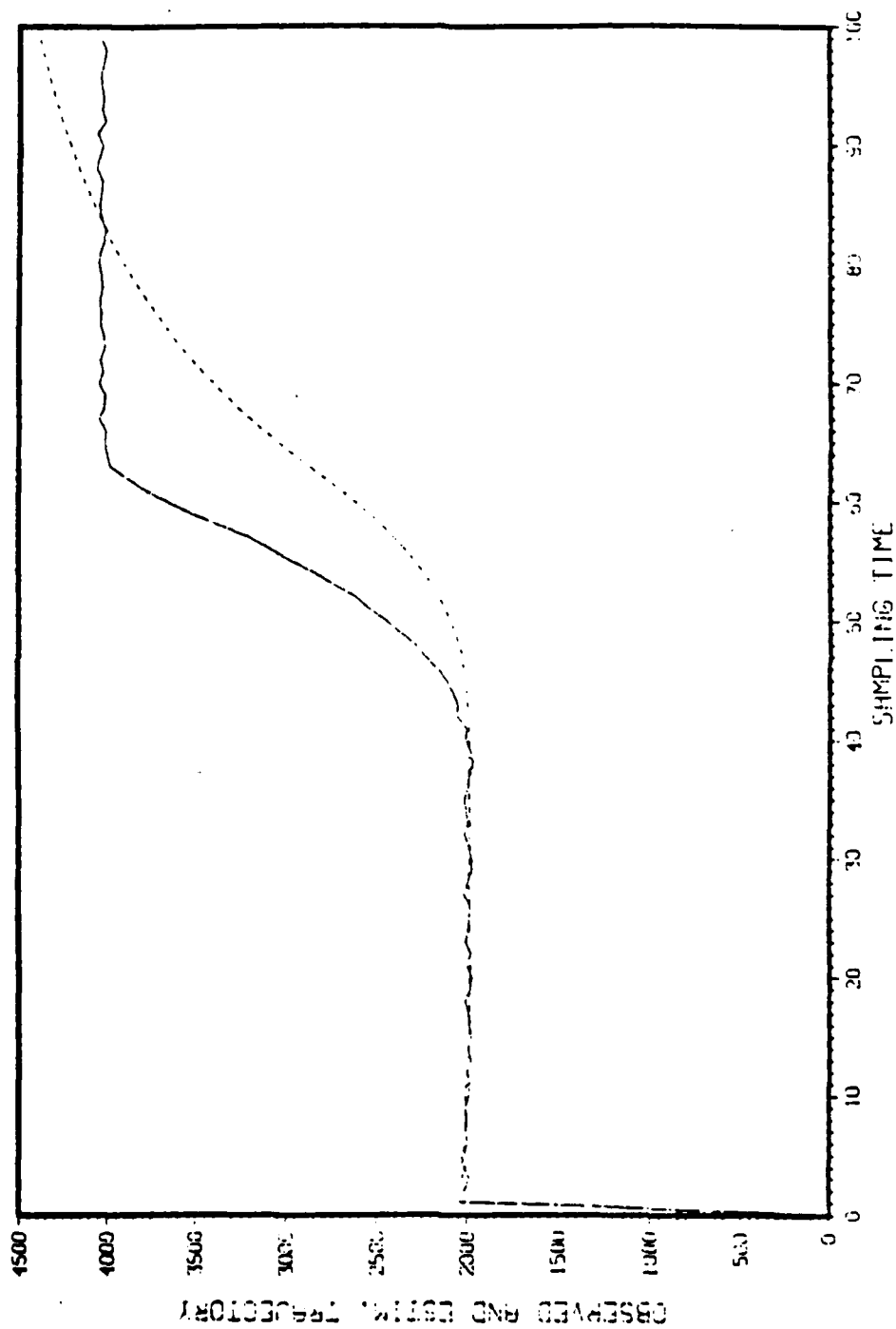


Figure 6.27 Obs. and Estim. Traj. Non-adjust. Filter
50 Monte Carlo Runs

OBS. AND ESTIM. TRAJ. NON-ADJUST. FILTER
 SINGLE RUN OF THE PROGRAM
 CONTIN. LINE: MEASUREMENTS TRAJECT.
 DASHED LINE: FILTER'S ESTIM. TRAJECT.

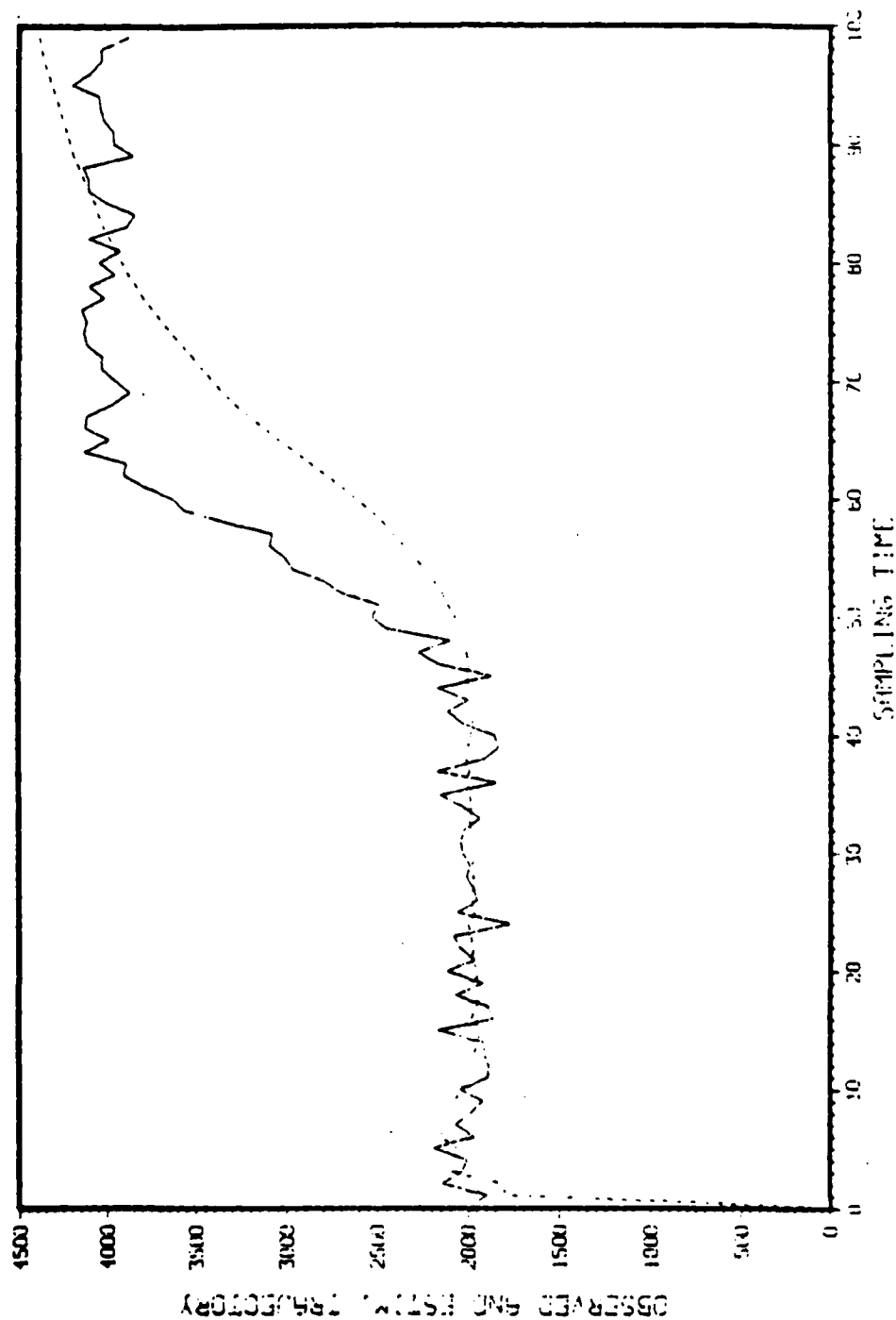


Figure 6.28 Obs. and Estim. Traj. Non-adjust. Filter
 Single Run of the Program

OBSERVED AND ESTIM. TRAJECT. 0-100 TR-.1

CONTIN. LINE: MEASUREMENTS TRAJECT.

DASHED LINE: FILTER S ESTIM. TRAJECT.

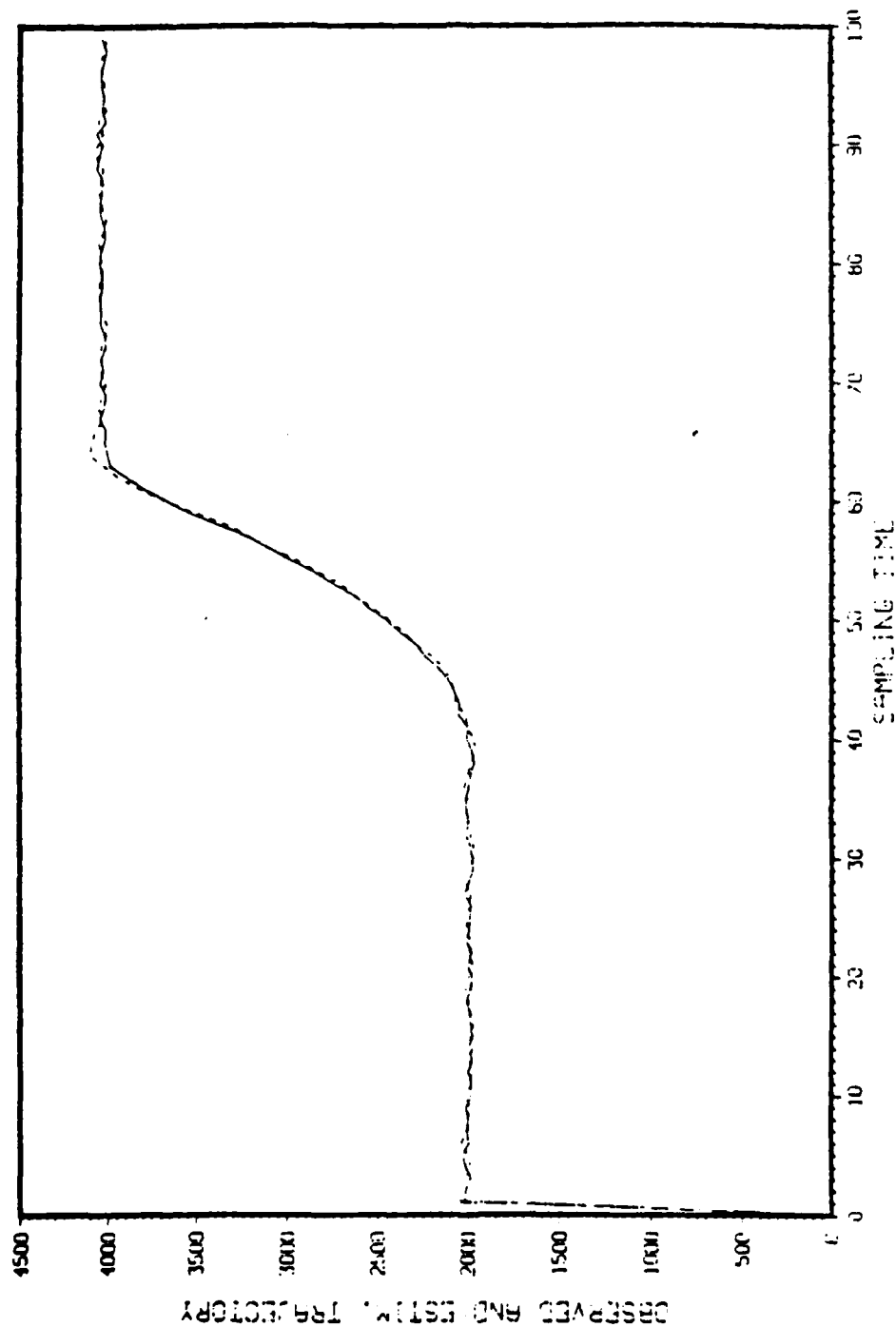


Figure 6.29 Observed and Estim. Traject. $Q^1 = 100$ TR-.1
50 Monte Carlo Runs

OBSERVED AND ESTIM. TRAJECT. 0-100 TR-3

CONTIN. LINE: MEASUREMENTS TRAJECT.

DASHED LINE: FILTER 5 ESTIM. TRAJECT.

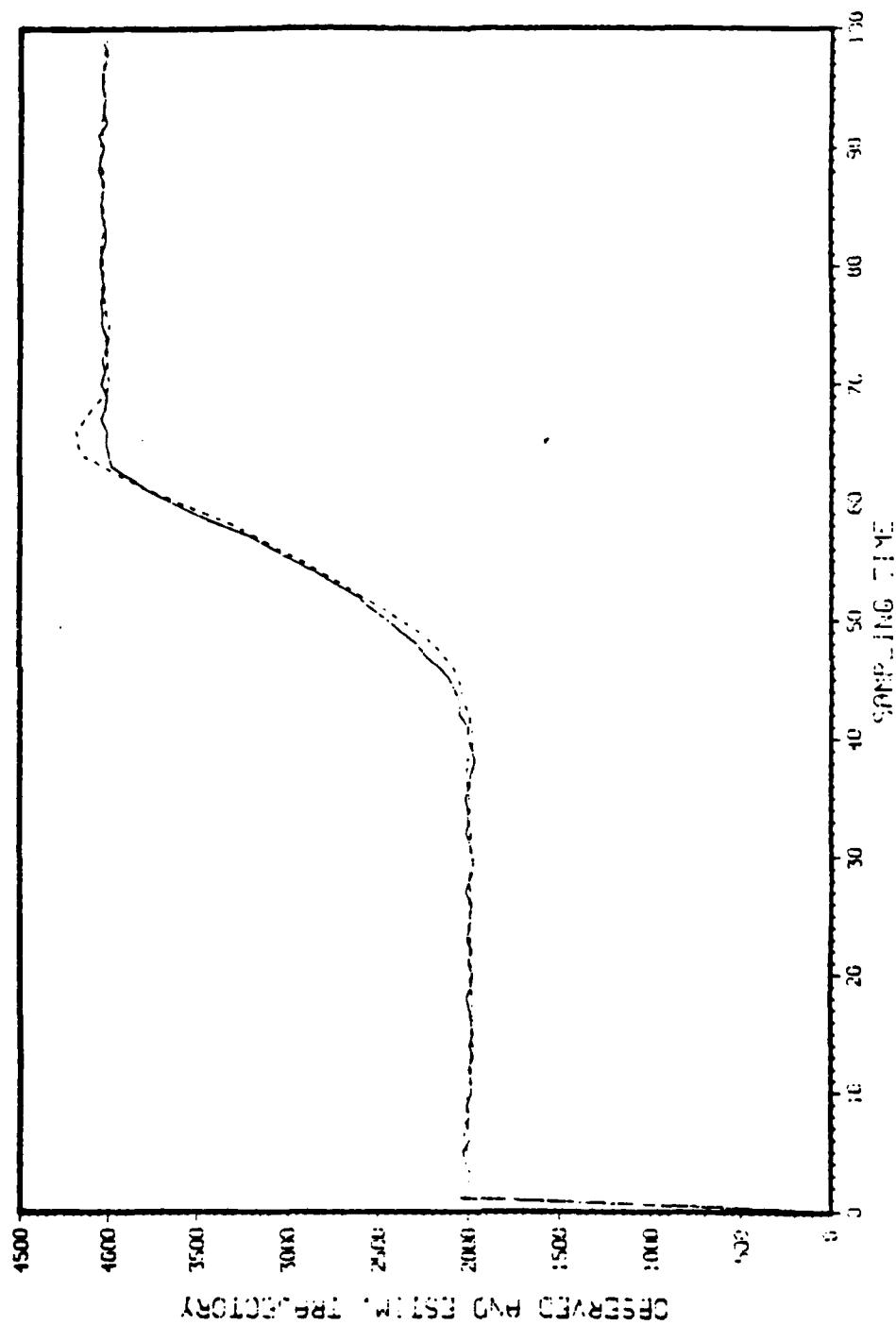


Figure 6.30 Observed and Estim. Traject. $Q' = 100$ TR = 3
50 Monte Carlo Runs

OBSERVED AND ESTIM. TRAJECT. 0-100 TR-20

CONTIN. LINE: MEASUREMENTS TRAJECT.

DASHED LINE: FILTER S ESTIM. TRAJECT.

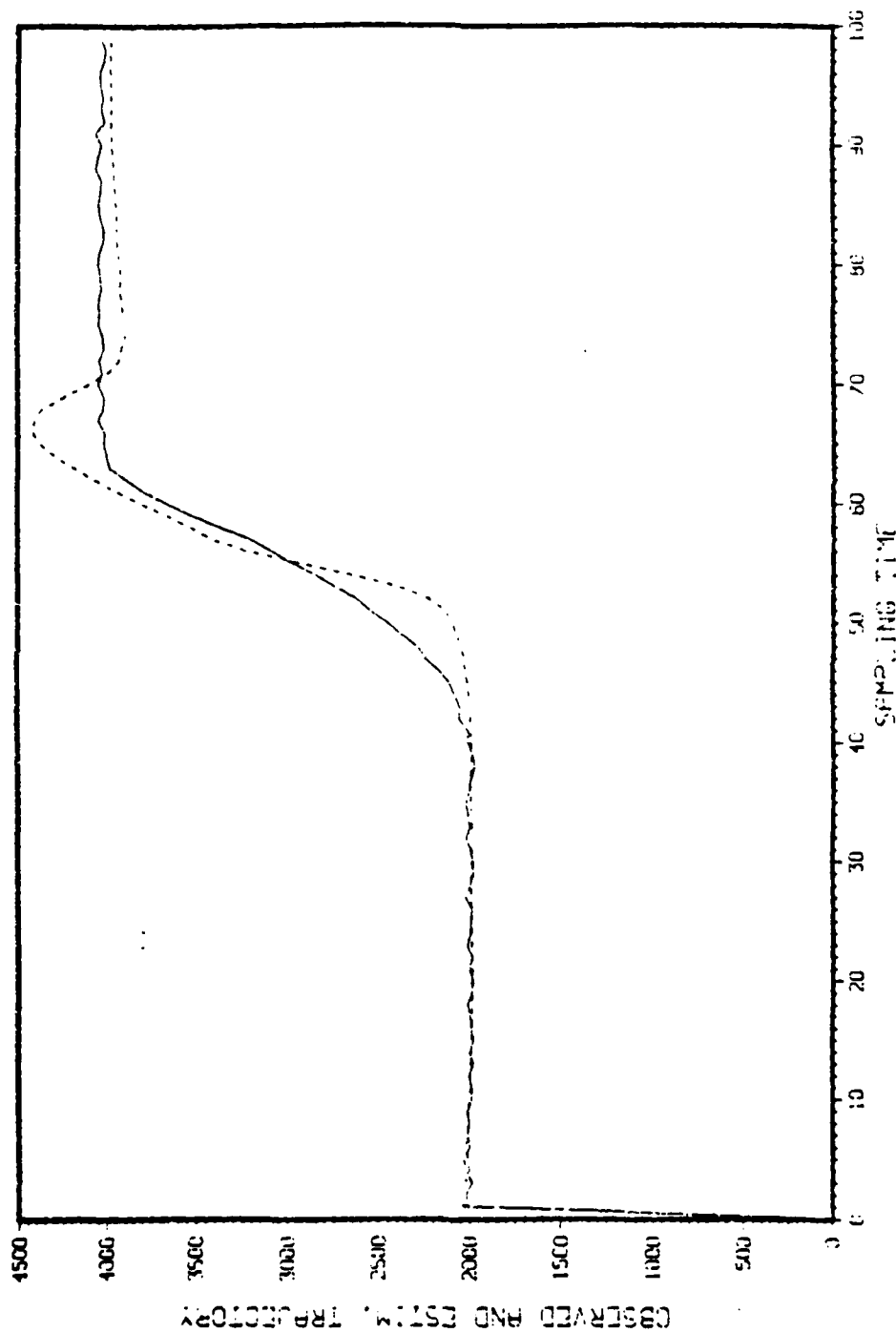


Figure 6.31 Observed and Estim. Traject. $Q' = 1.00$ TR = 20
50 Monte Carlo Runs

OBSERVED AND ESTIM. TRAJECT. G-.1 TR-.1
 SINGLE RUN OF THE PROGRAM
 CONTIN. LINE: MEASUREMENTS TRAJECT.
 DASHED LINE: FILTER'S ESTIM. TRAJECT.

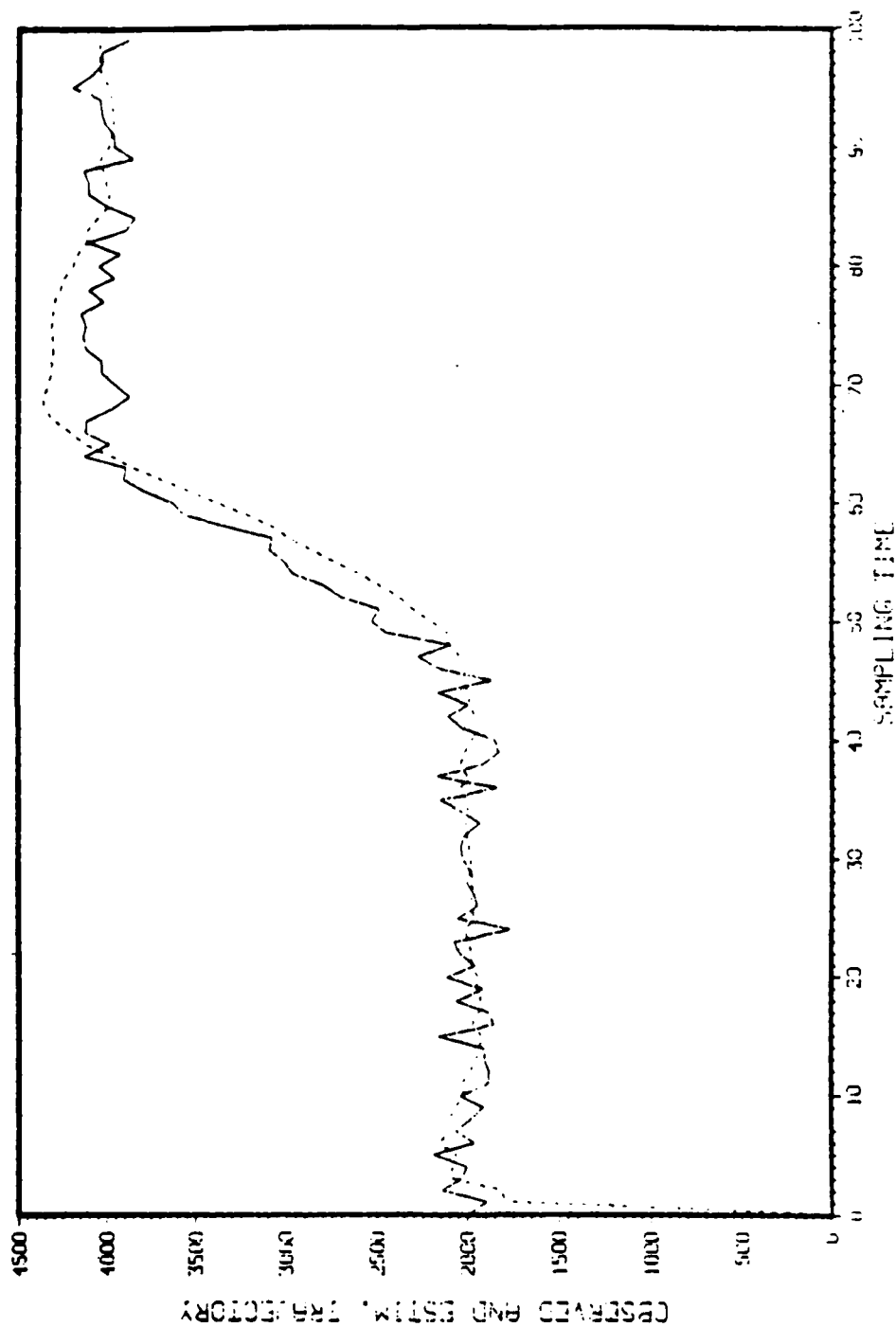


Figure 6.32 Observed and Estim. Traject. $Q' = .1$ $TR = .1$
 50 Monte Carlo Runs

OBSERVED AND ESTIM. TRAJECT. 0-3 TR-.1

SINGLE RUN OF THE PROGRAM

CONTIN. LINE: MEASUREMENTS TRAJECT.

DASHED LINE: FILTER S ESTIM. TRAJECT.

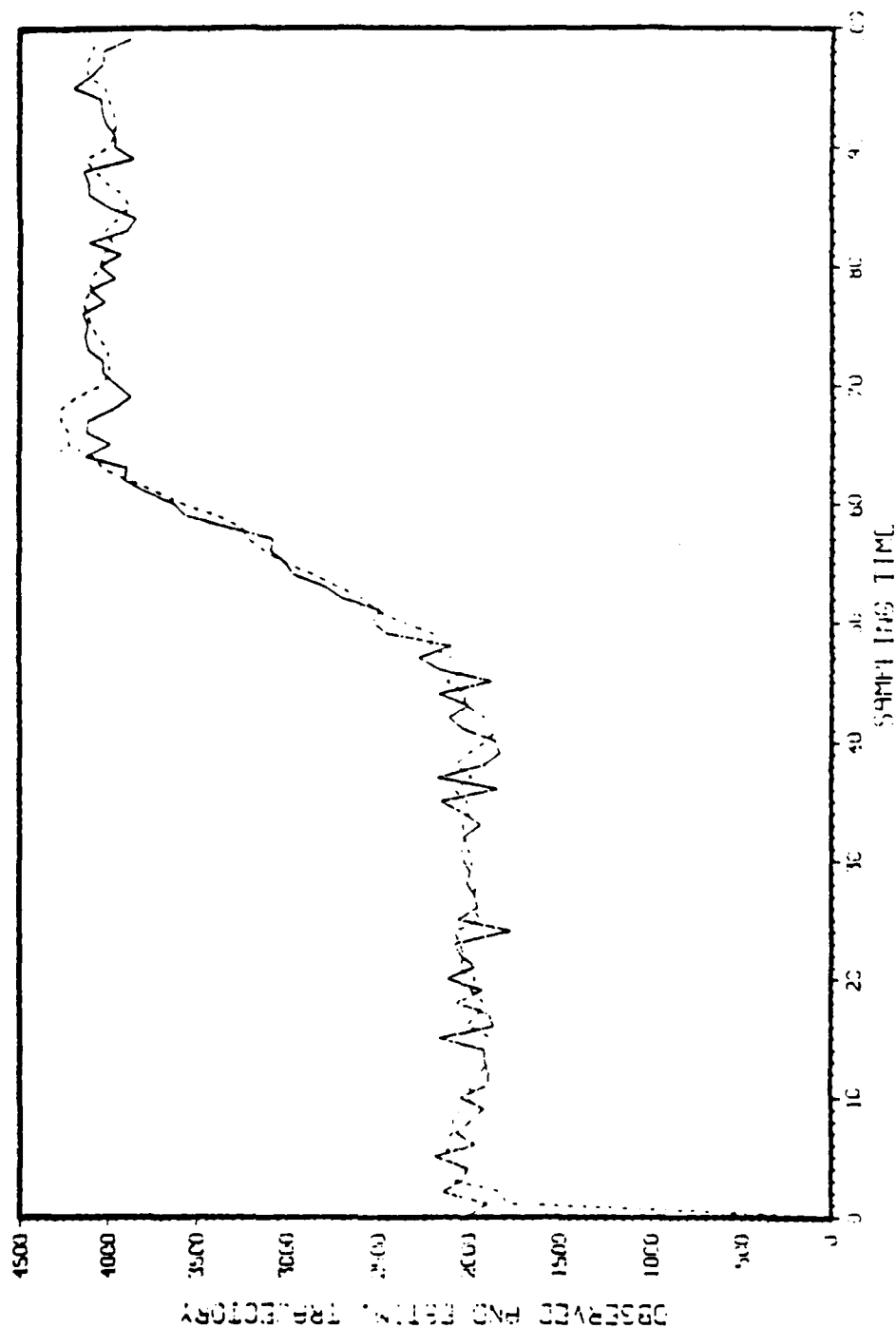


Figure 6.33 Observed and Estim. Traject. $Q' = 3 \text{ TR} \cdot 1$
Single Run of the Program

OBSERVED AND ESTIM. TRAJECT. 0-10 TR-1

SINGLE RUN OF THE PROGRAM

CONTIN. LINE: MEASUREMENTS TRAJECT.

DASHED LINE: FILTER S ESTIM. TRAJECT.

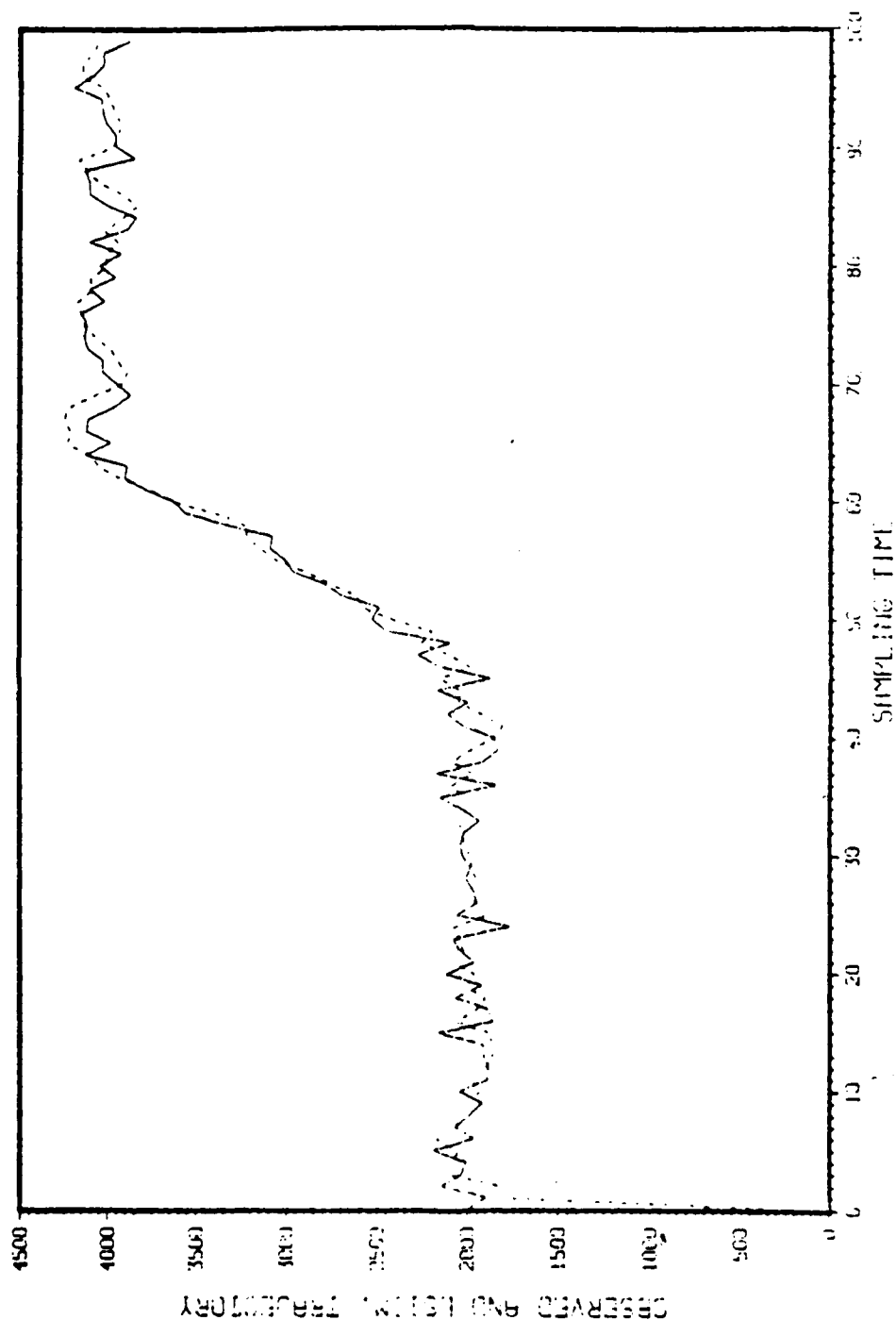


Figure 6.34 Observed and Estim. Traject. $Q' = 10$ TR = .1
Single Run of the Program

OBSERVED AND ESTIM. TRAJECT. 0-100 TR-.1

SINGLE RUN OF THE PROGRAM

CONTIN. LINE: MEASUREMENTS TRAJECT.

DASHED LINE: FILTER S ESTIM. TRAJECT.

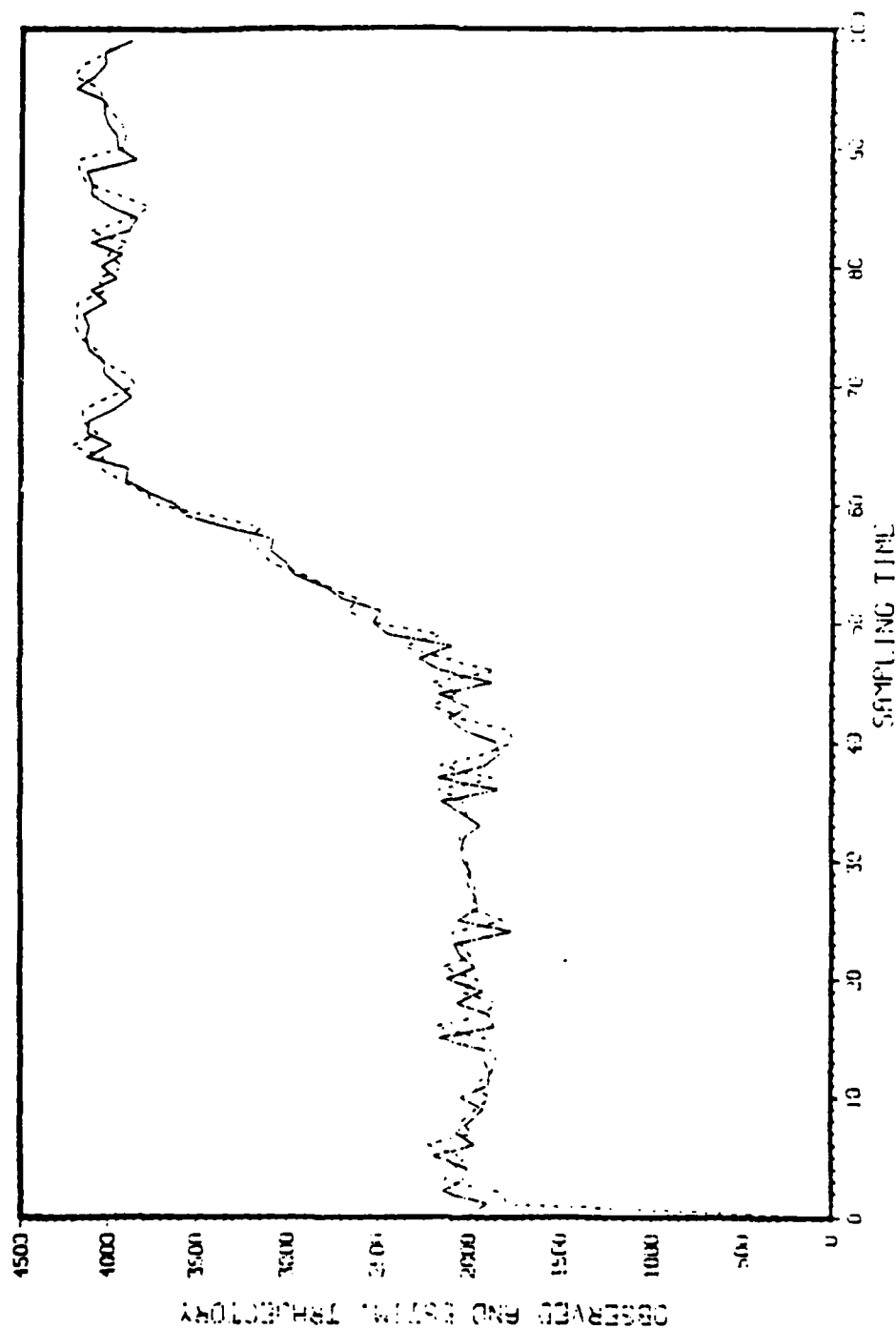


Figure 6.35 Observed and Estim. Traject. $Q' = 100$ TR = .1
Single Run of the Program

OBSERVED AND ESTIM. TRAJECT. C-1000 TR-.1

SINGLE RUN OF THE PROGRAM

CONTIN. LINE: MEASUREMENTS TRAJECT.

DASHED LINE: FILTER S ESTIM. TRAJECT.

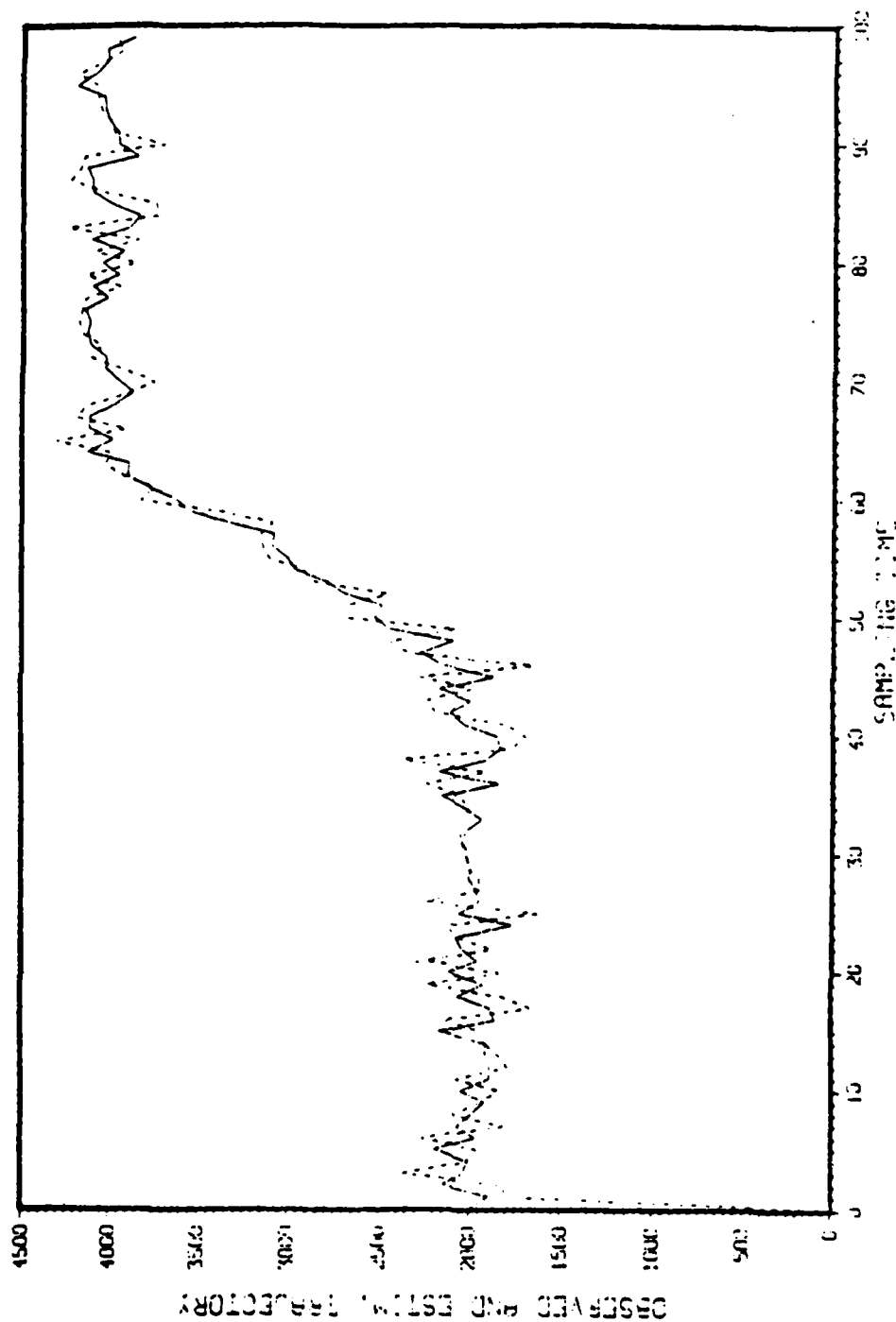


Figure 6.36 Observed and Estim. Traject. Q' - 1000 TR - .1
Single Run of the Program

OBSERVED AND ESTIM. TRAJECT. 0-1 TR-3

SINGLE RUN OF THE PROGRAM

CONTIN. LINE: MEASUREMENTS TRAJECT.

DASHED LINE: FILTER'S ESTIM. TRAJECT.

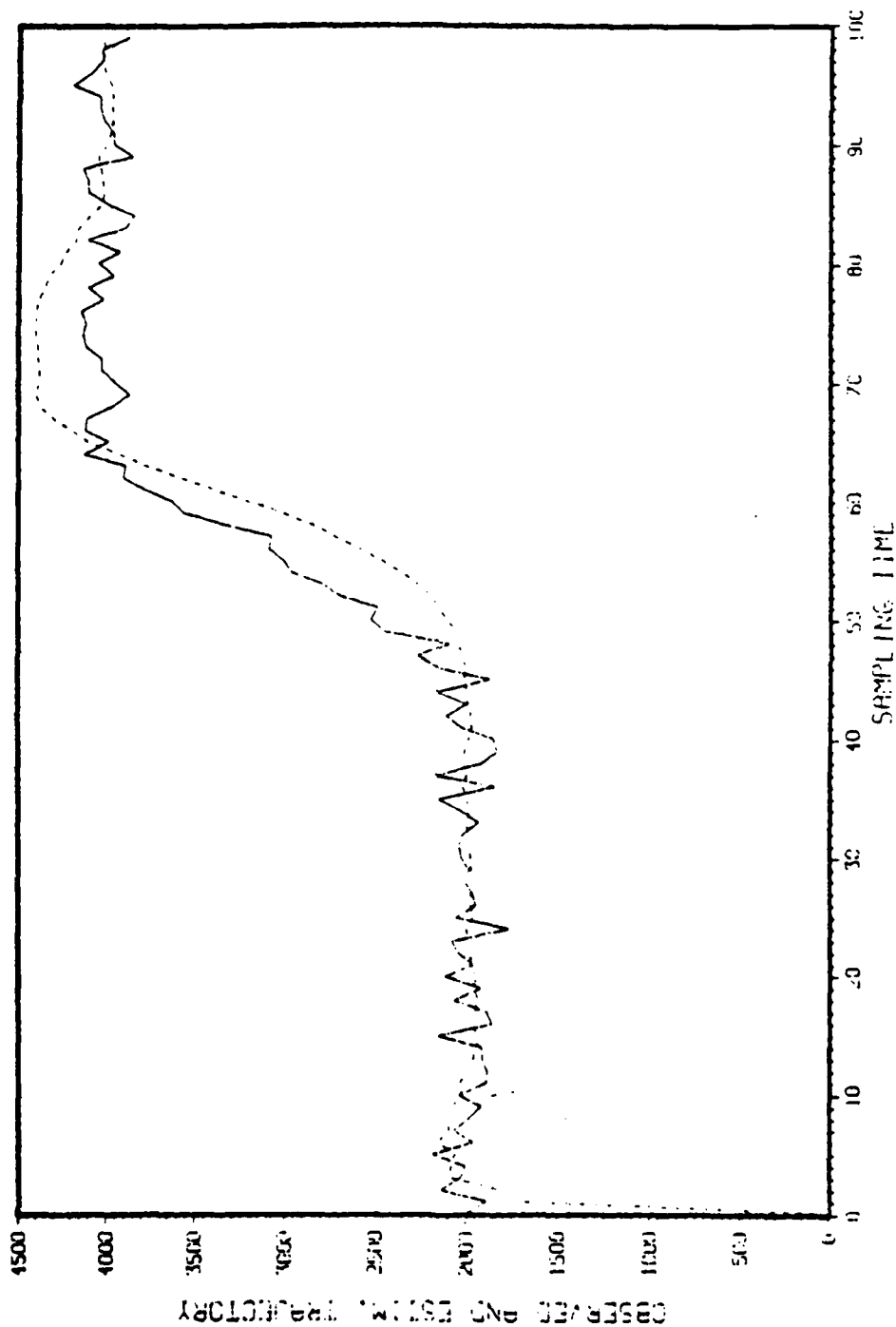


Figure 6.37 Observed and Estim. Traject. 0' .1 PR = 3
Single Run of the Program

OBSERVED AND ESTIM. TRAJECT. 0-3 TR-3

SINGLE RUN OF THE PROGRAM

CONTIN. LINE: MEASUREMENTS TRAJECT.

DASHED LINE: FILTER S ESTIM. TRAJECT.

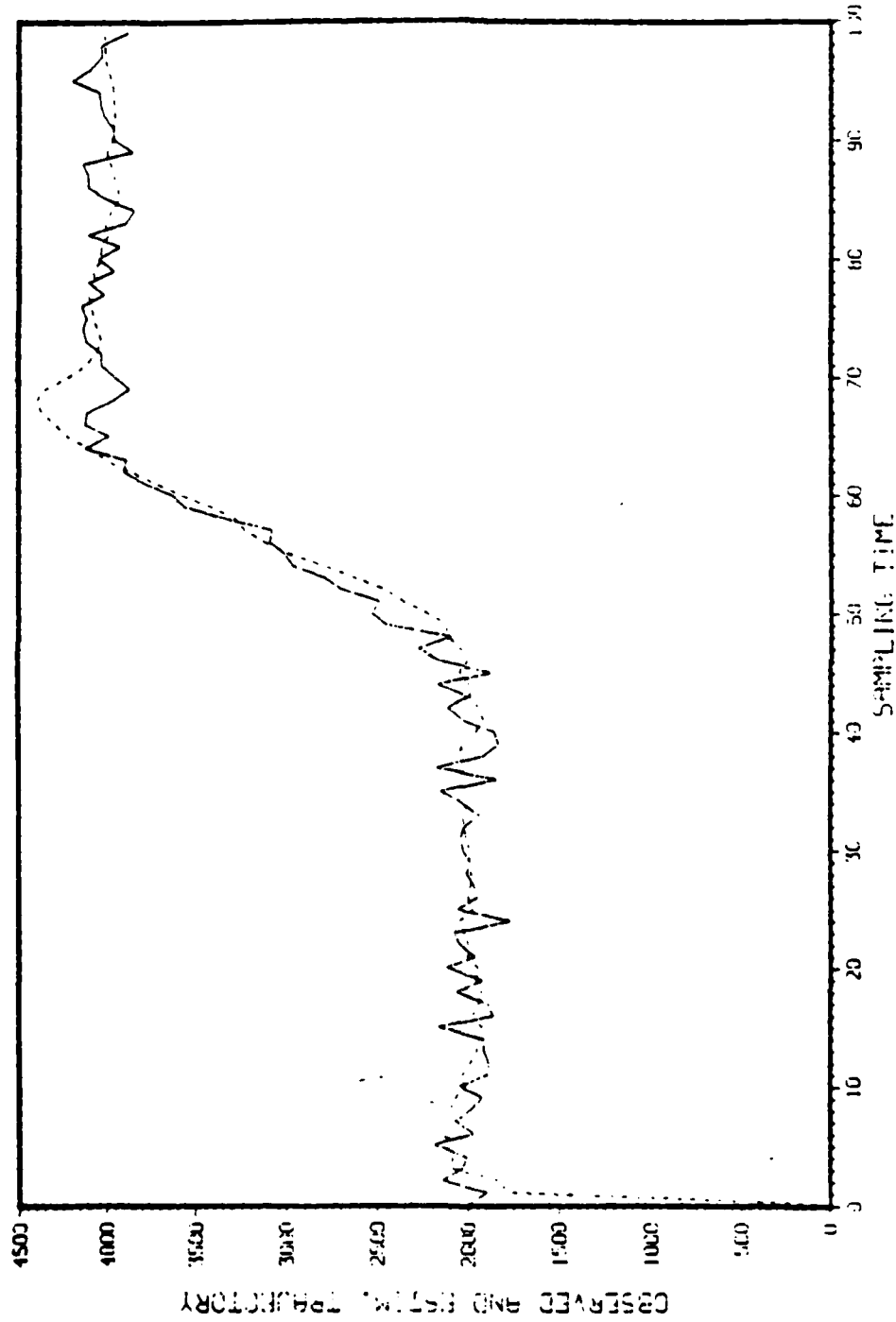


Figure 6.38 Observed and Estim. Traject. $Q^* = 3$ $TR = 3$
Single Run of the Program

OBSERVED AND ESTIM. TRAJECT. 0-10 TR-3

SINGLE RUN OF THE PROGRAM

CONTIN. LINE: MEASUREMENTS TRAJECT.

DASHED LINE: FILTER 5 ESTIM. TRAJECT.

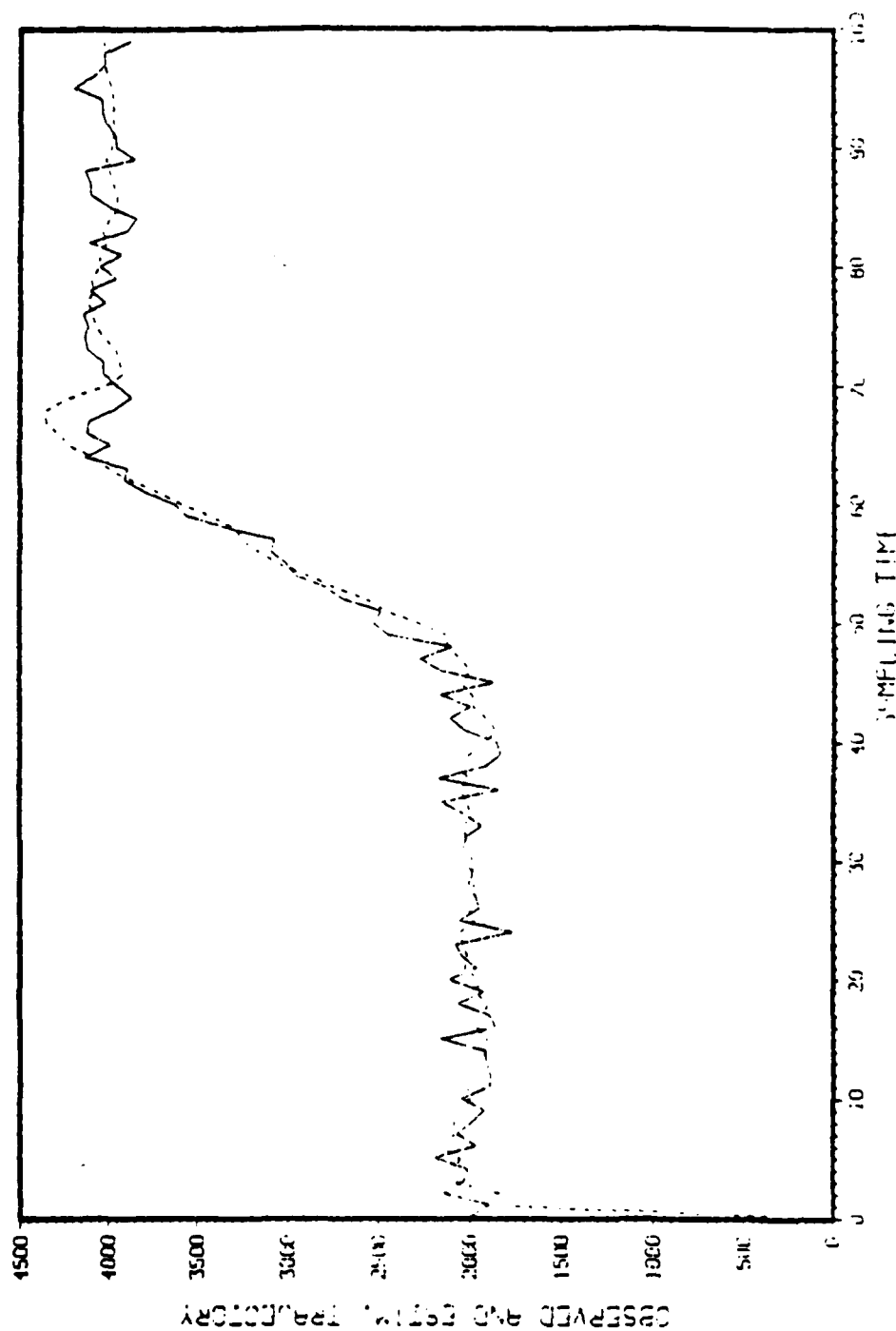


Figure 6.39 Observed and Estim. Traject. $Q' = 10$ TR 3
Single Run of the Program

OBSERVED AND ESTIM. TRAJECT. 0-100 TR-3

SINGLE RUN OF THE PROGRAM

CONTIN. LINE: MEASUREMENTS TRAJECT.

DASHED LINE: FILTER S ESTIM. TRAJECT.

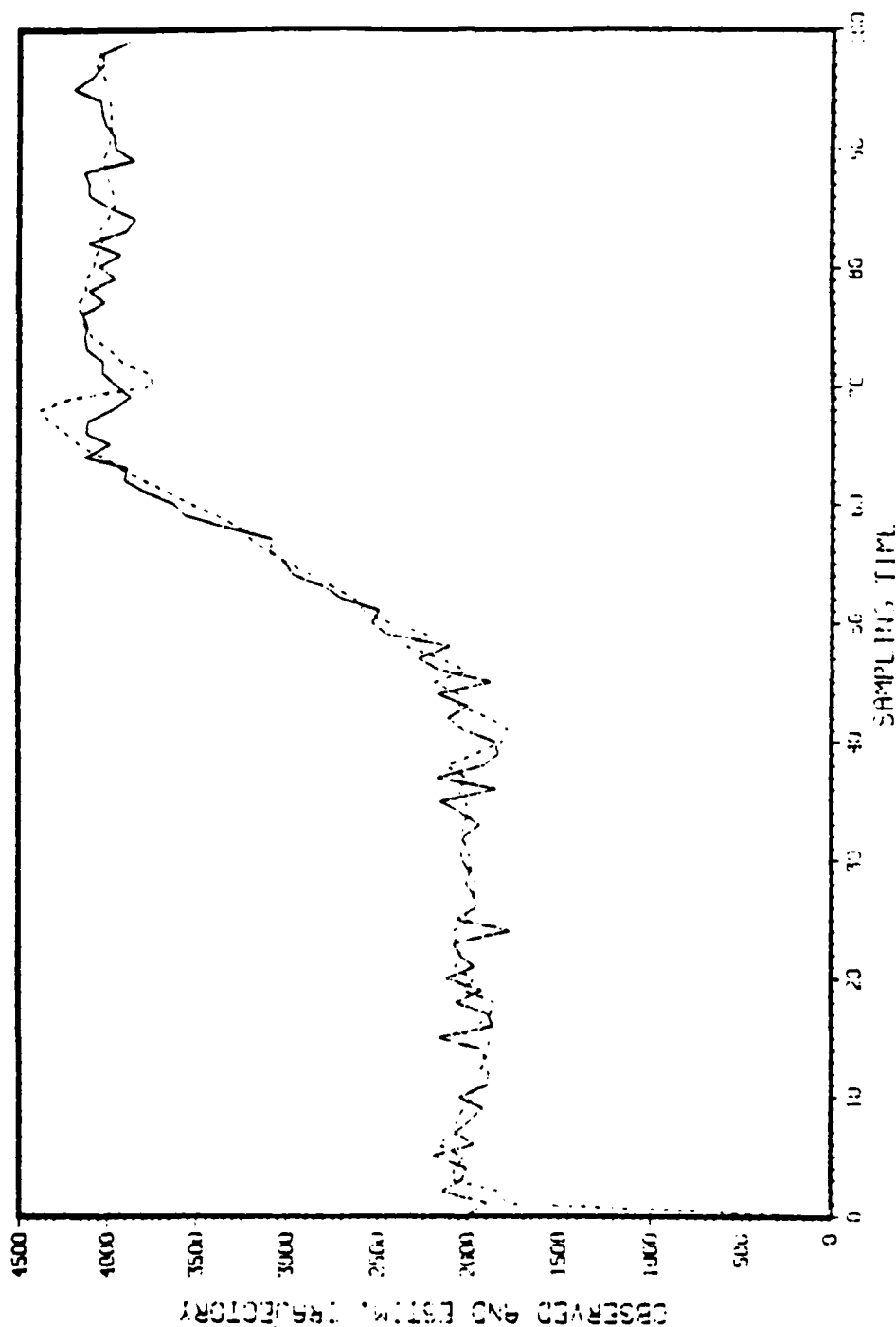


Figure 6.40 Observed and Estim. Traject. $Q^* = 100$ TR = 3
Single Run of the Program

OBSERVED AND ESTIM. TRAJECT. 0-1600 TR-3

SINGLE RUN OF THE PROGRAM

CONTIN. LINE: MEASUREMENTS TRAJECT.

DASHED LINE: FILTER S ESTIM. TRAJECT.

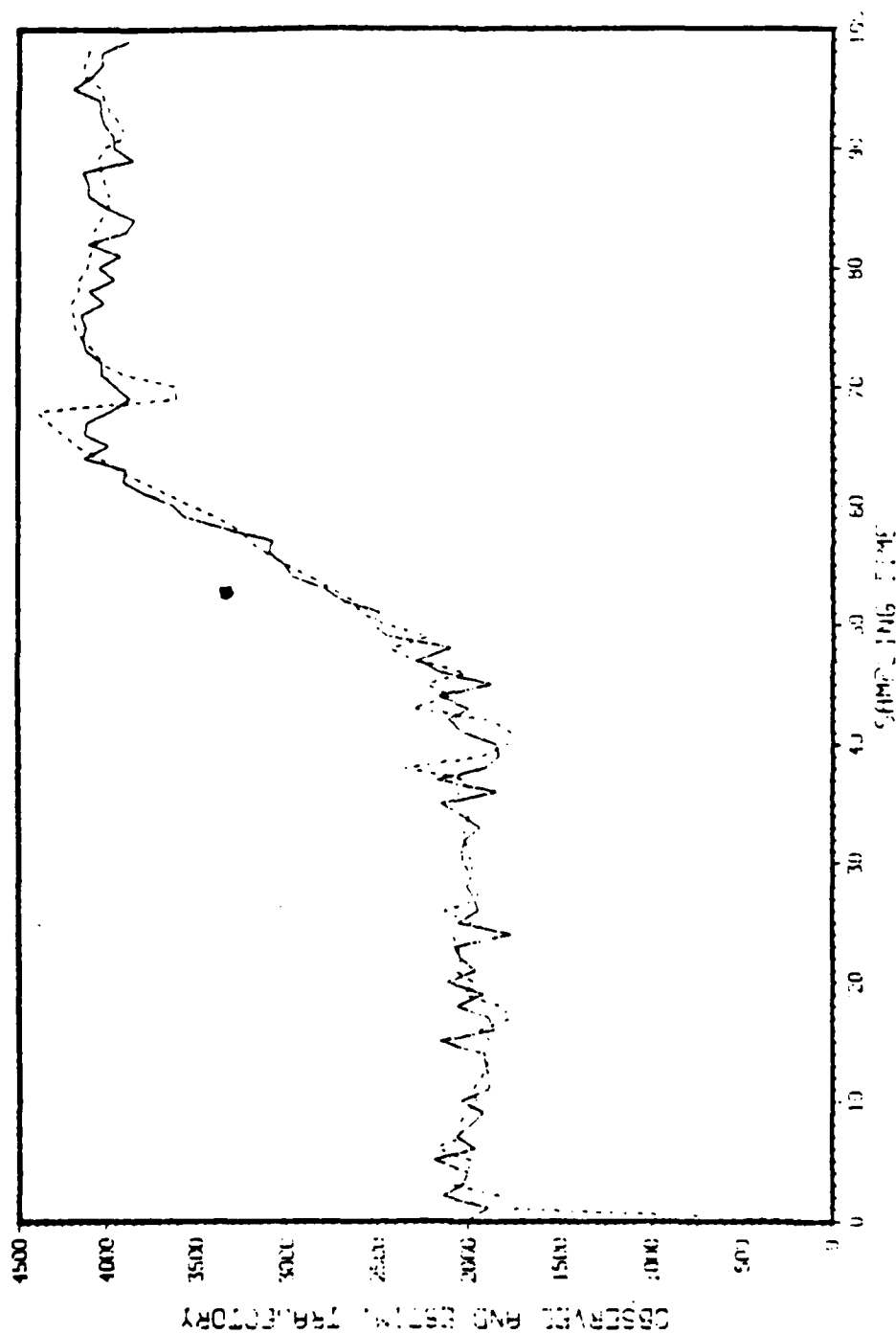


Figure 6.41 Observed and Estim. Traject. $Q' = 1000$ TR 3
Single Run of the Program

OBSERVED AND ESTIM. TRAJECT. 0-.1 TR-20

SINGLE RUN OF THE PROGRAM

CONTIN. LINE: MEASUREMENTS TRAJECT.

DASHED LINE: FILTER'S ESTIM. TRAJECT.

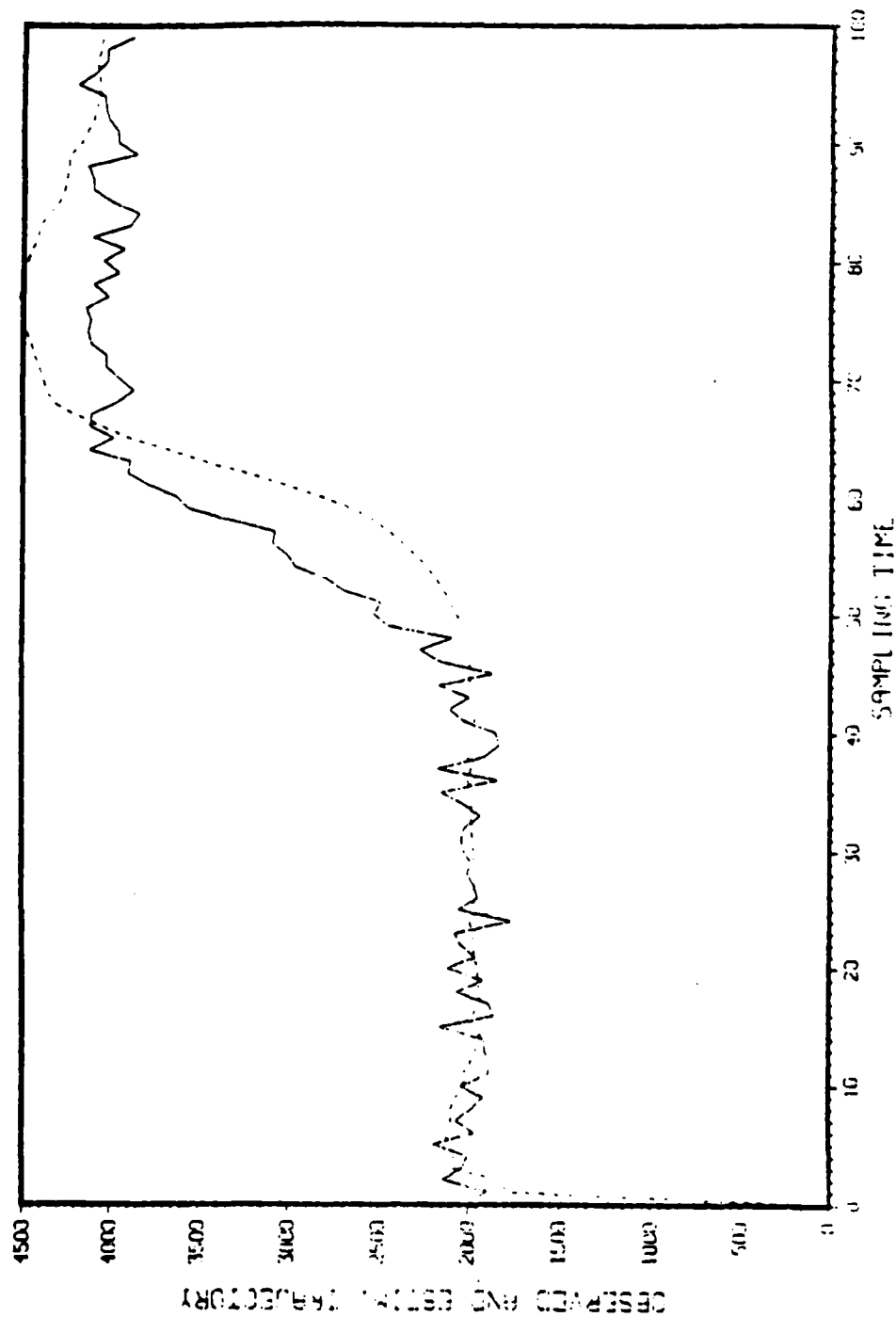


Figure 6.42 Observed and Estim. Traject. $Q' = .1$ $TR = 20$
Single Run of the Program

OBSERVED AND ESTIM. TRAJECT. Q-3 TR-20

SINGLE RUN OF THE PROGRAM

CONTIN. LINE: MEASUREMENTS TRAJECT.

DASHED LINE: FILTER S ESTIM. TRAJECT.

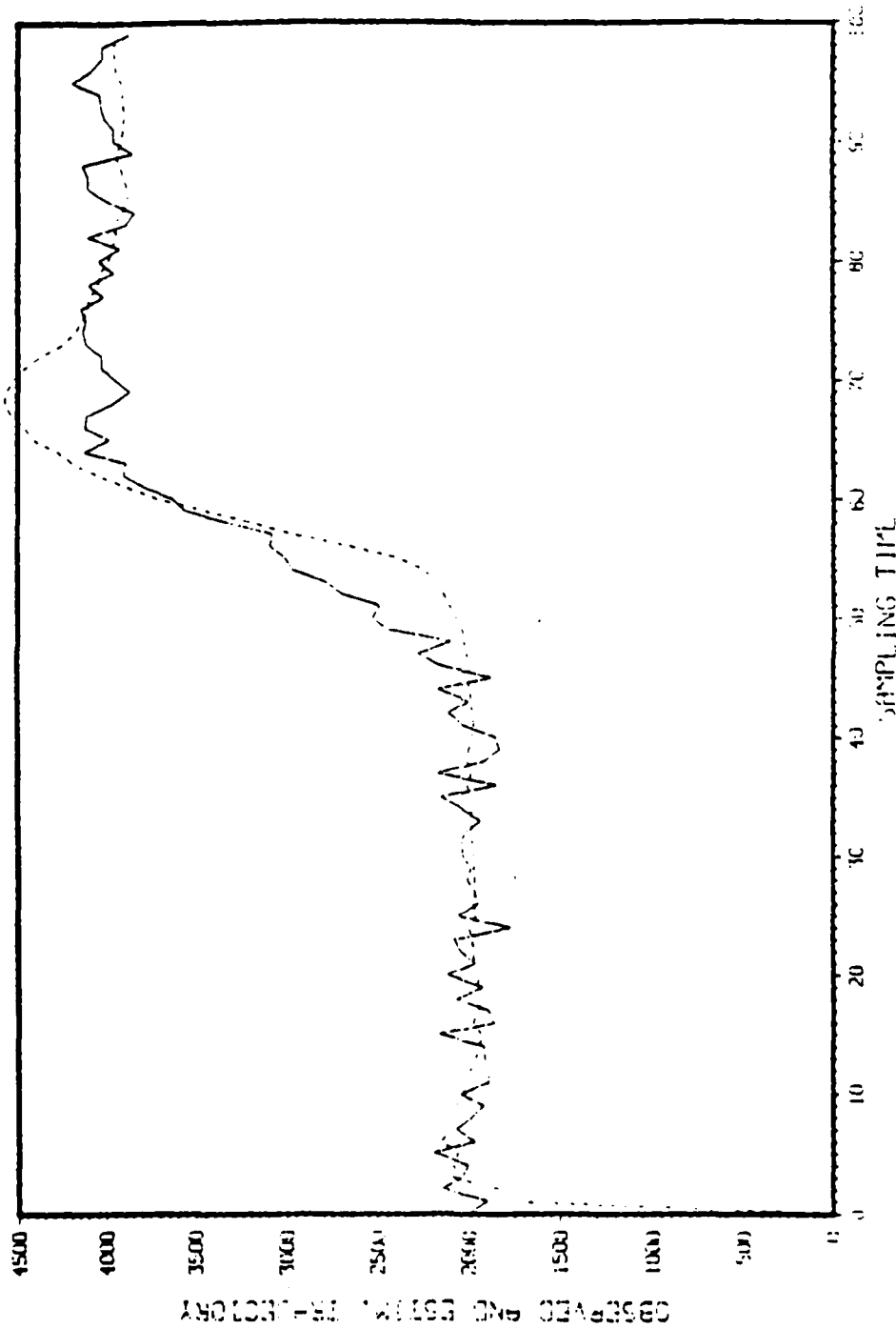


Figure 6.43 Observed and Estim. Traject. $Q^* = 3$ TR 20
Single Run of the Program

OBSERVED AND ESTIM. TRAJECT. 0-10 TR-20

SINGLE RUN OF THE PROGRAM

CONTIN. LINE: MEASUREMENTS TRAJECT.

DASHED LINE: FILTER S ESTIM. TRAJECT.

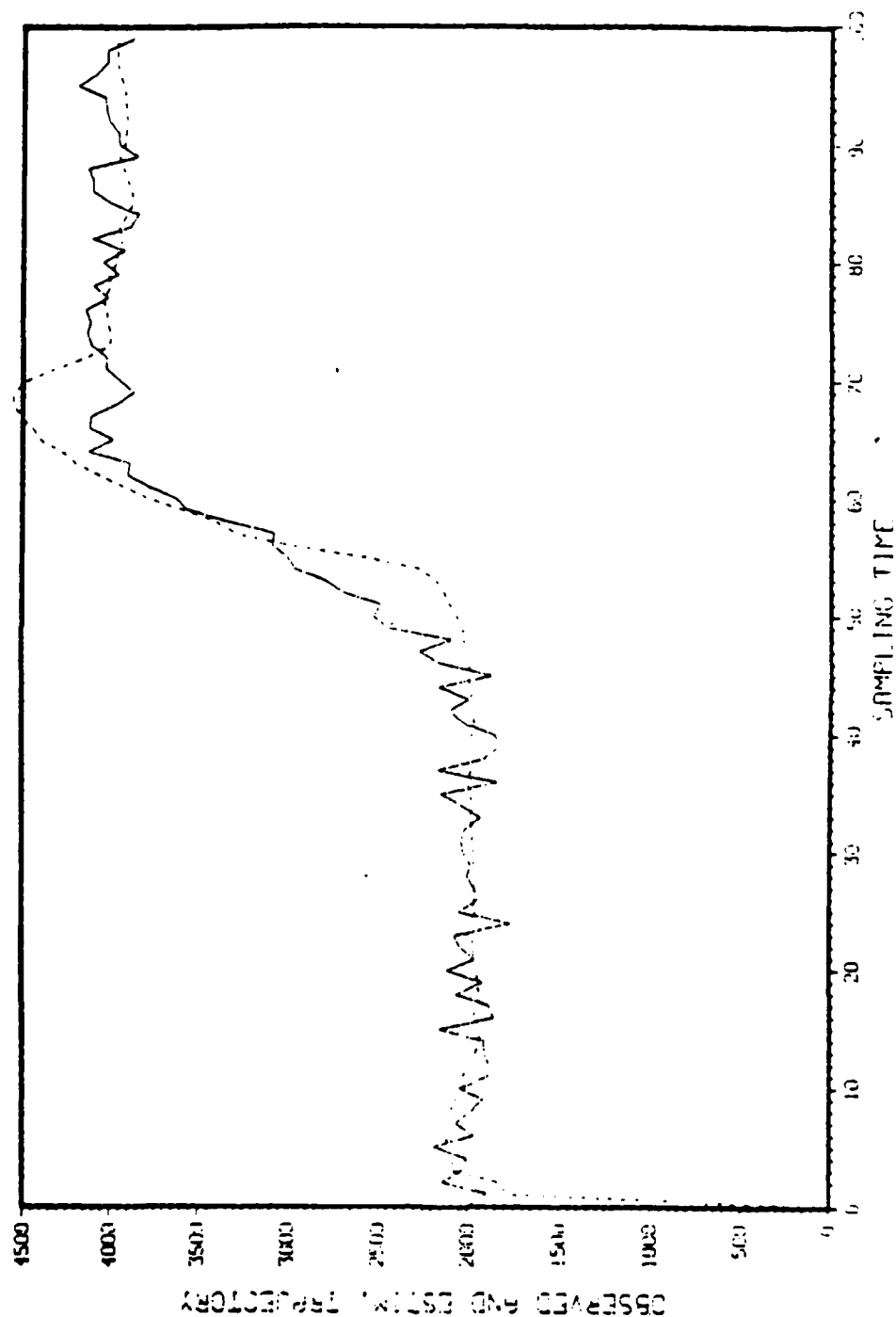


Figure 6.44 Observed and Estim. Traject. $Q' = 10$ $TR = 20$
Single Run of the Program

OBSERVED AND ESTIM. TRAJECT. 0-100 TR-20

SINGLE RUN OF THE PROGRAM

CONTIN. LINE: MEASUREMENTS TRAJECT.

DASHED LINE: FILTER S ESTIM. TRAJECT.

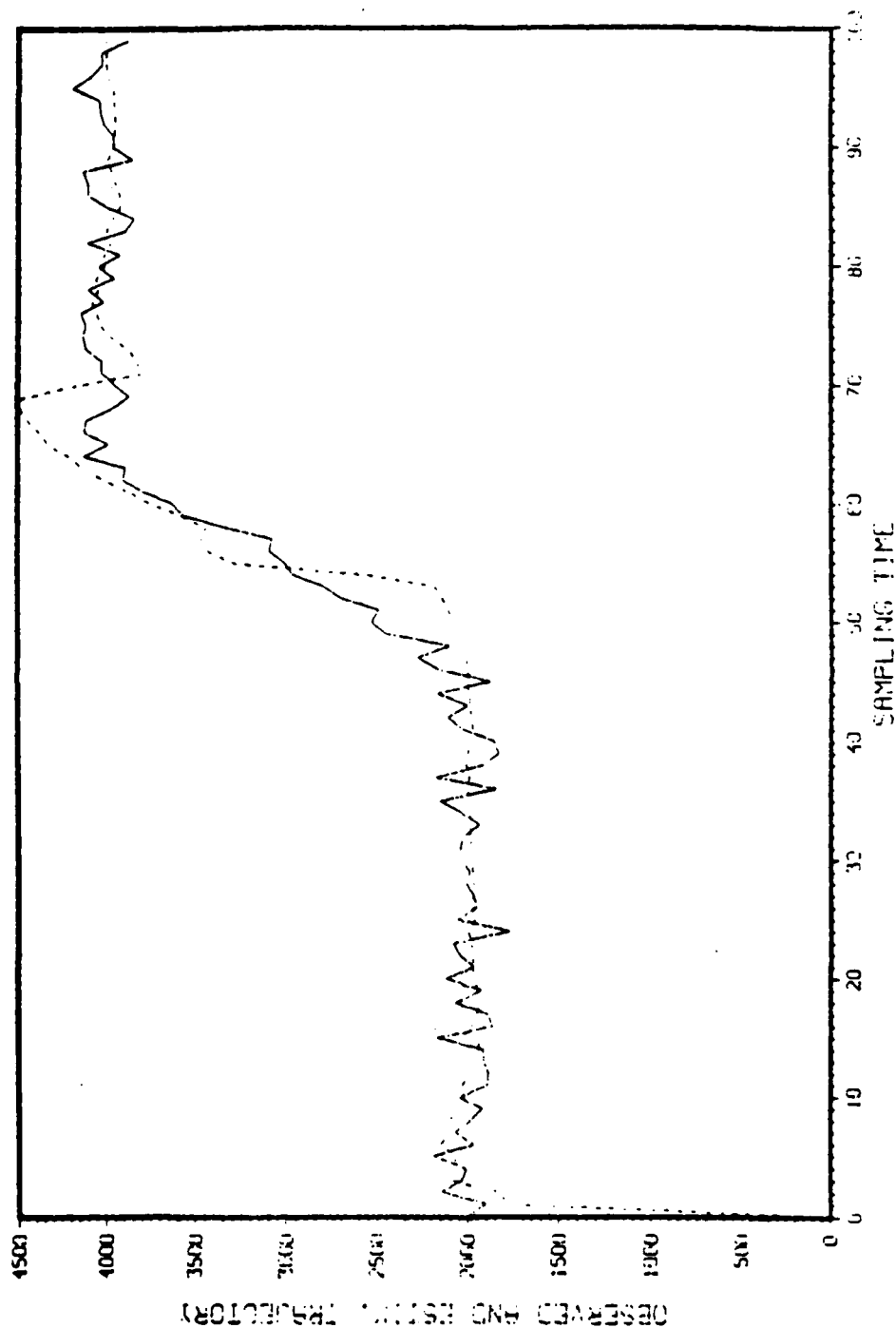


Figure 6.45 Observed and Estim. Traject. $Q' = 100$ TR = 20
Single Run of the Program

OBSERVED AND ESTIM. TRAJECT. 0-1000 TR-20

SINGLE RUN OF THE PROGRAM

CONTIN. LINE: MEASUREMENTS TRAJECT.

DASHED LINE: FILTER'S ESTIM. TRAJECT.

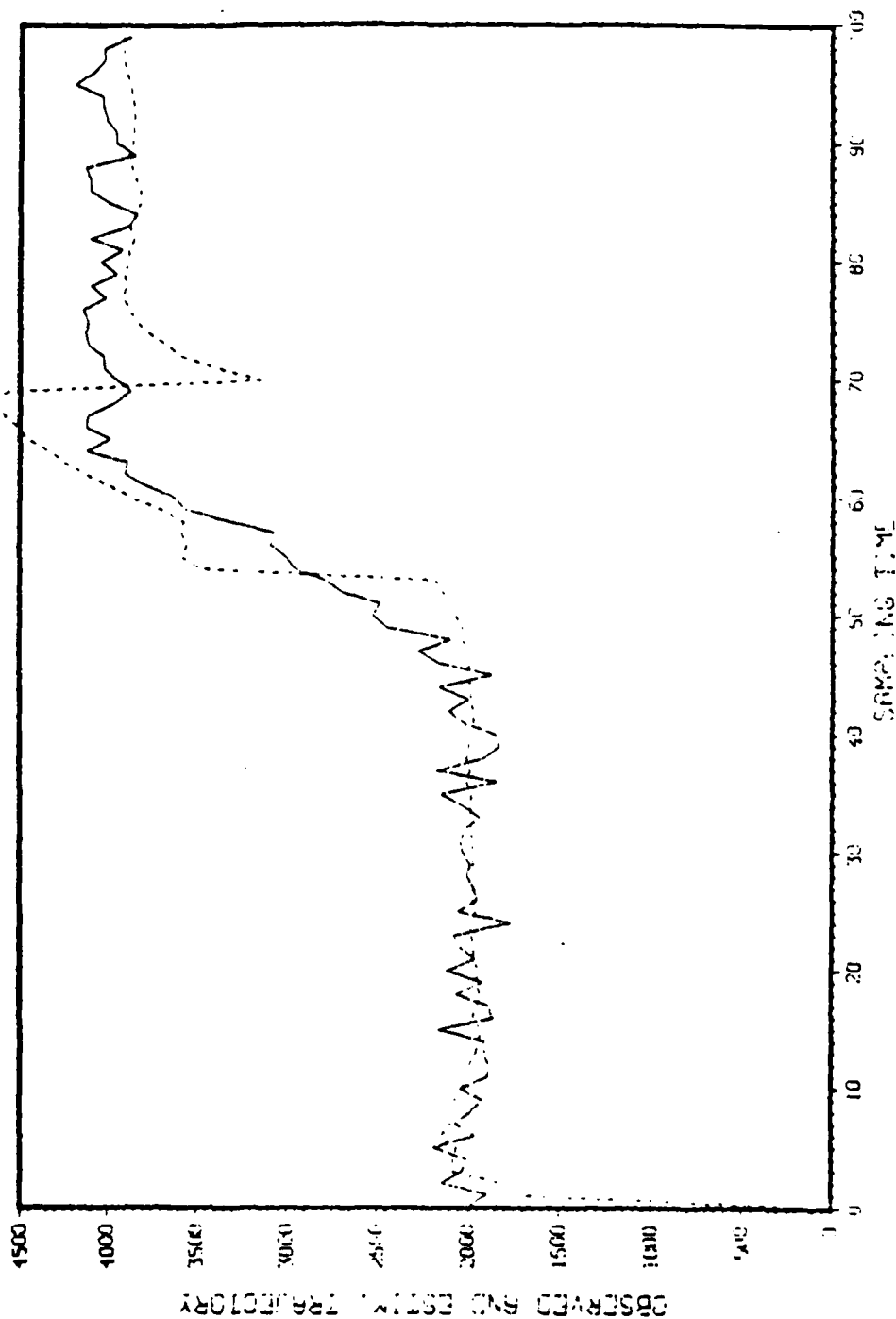


Figure 6.46 Observed and Estim. Traject. $Q' = 1000$ $TR = 20$
Single Run of the Program

MEAN SQUARE ERROR OF POS. $\sigma^2 = .1$ $TR = .1$

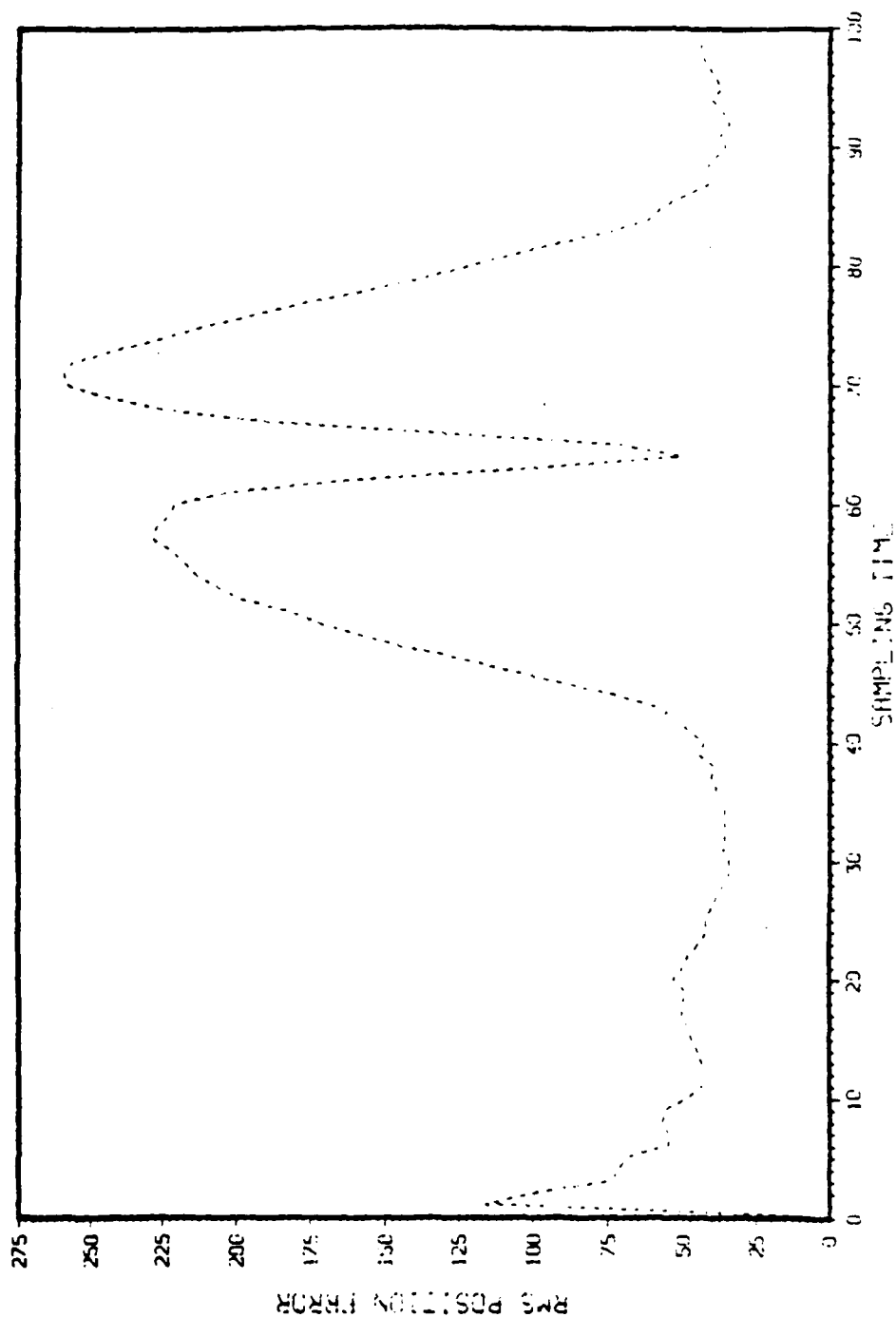


Figure 6.47 Mean Square Error of Pos. $\sigma^2 = .1$ $TR = .1$

MEAN SQUARE ERROR OF POS. $Q = 4$ $TR = .1$

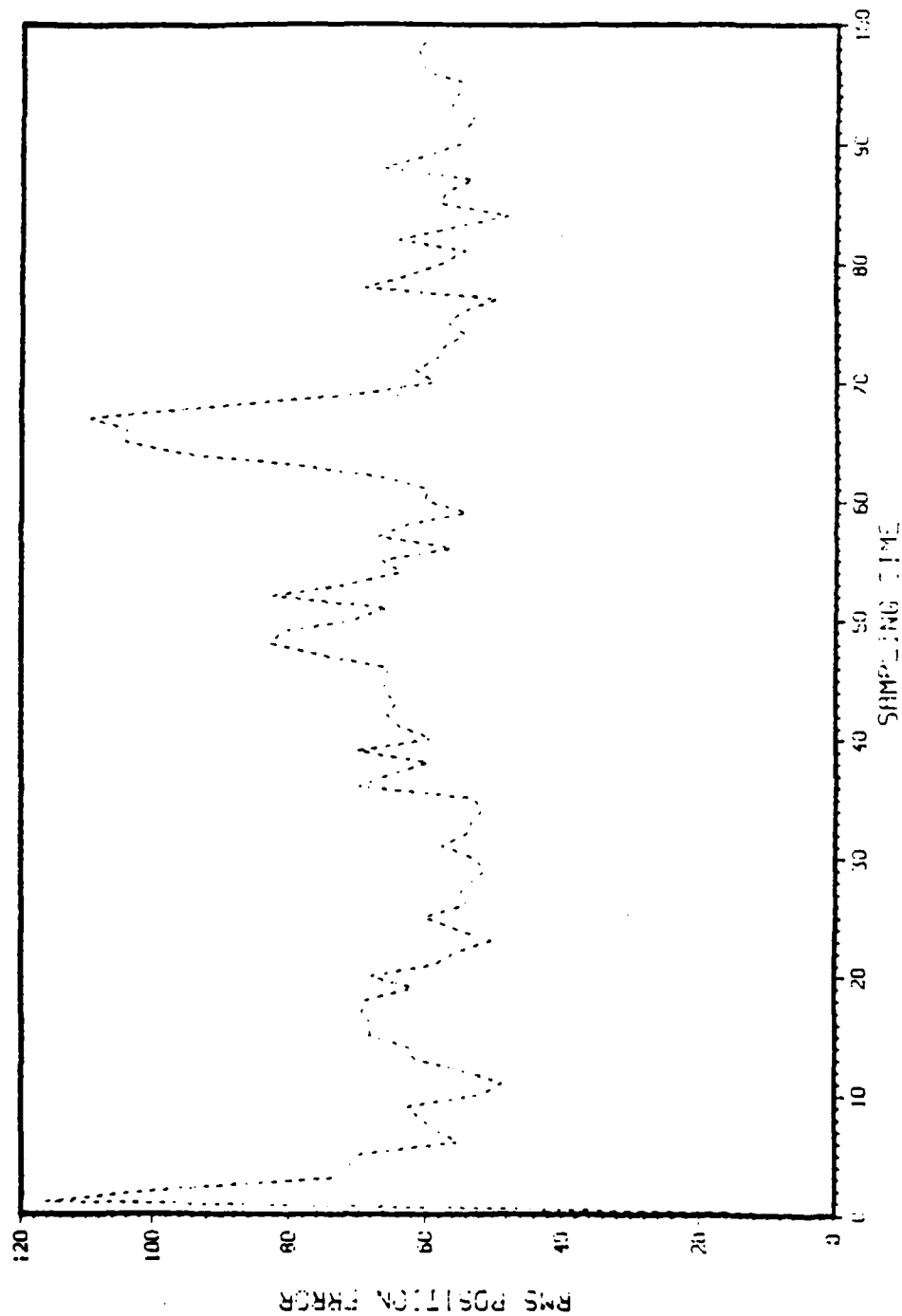


Figure 6.48 Mean Square Error of Pos. $Q = 4$ $TR = .1$

MEAN SQUARE ERROR OF POS. $Q = 10$ $TR = .1$

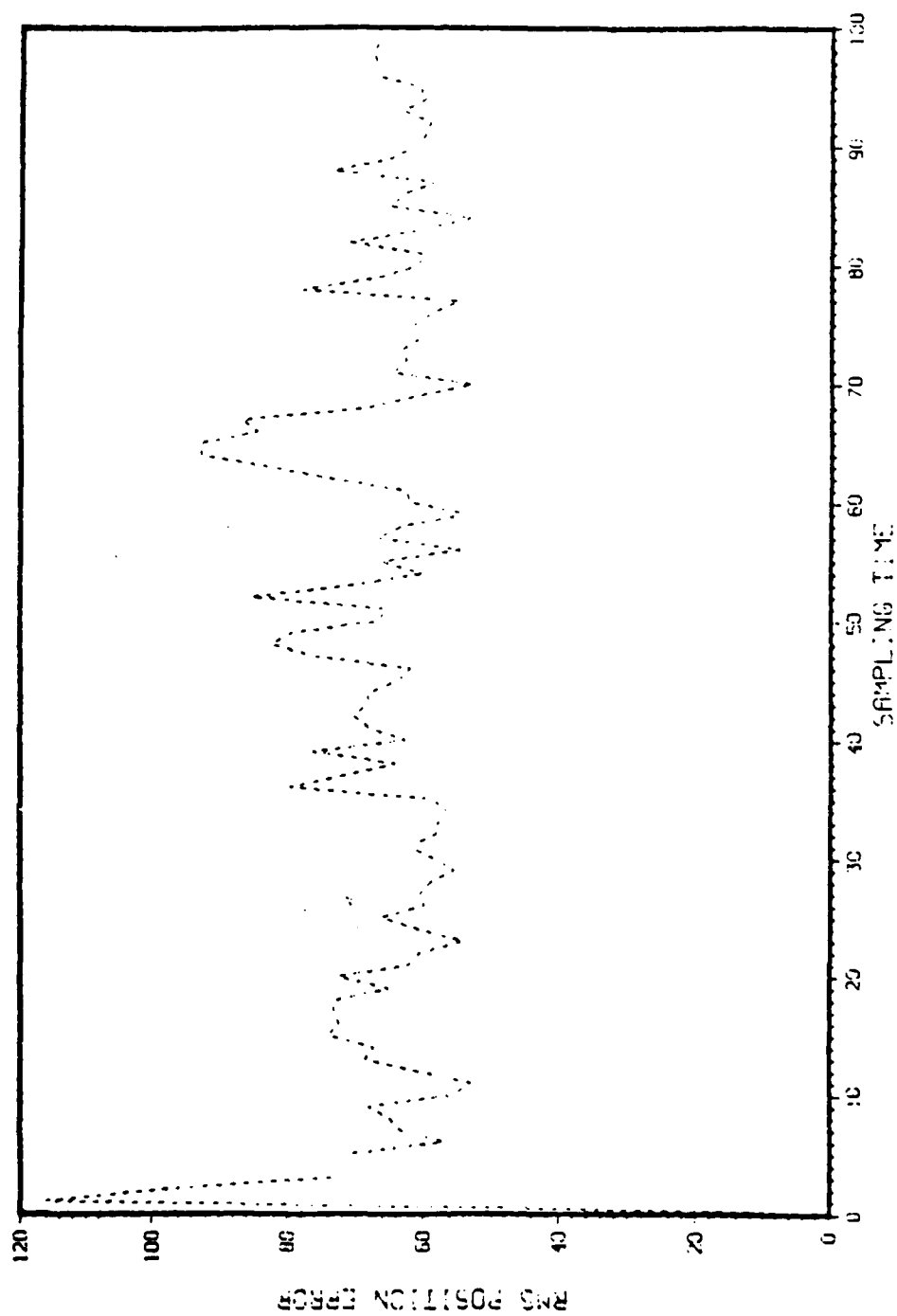


Figure 6.49 Mean Square Error of Pos. $Q = 10$ $TR = .1$

MEAN SQUARE ERROR OF POS. $Q=100$ $TR=.1$

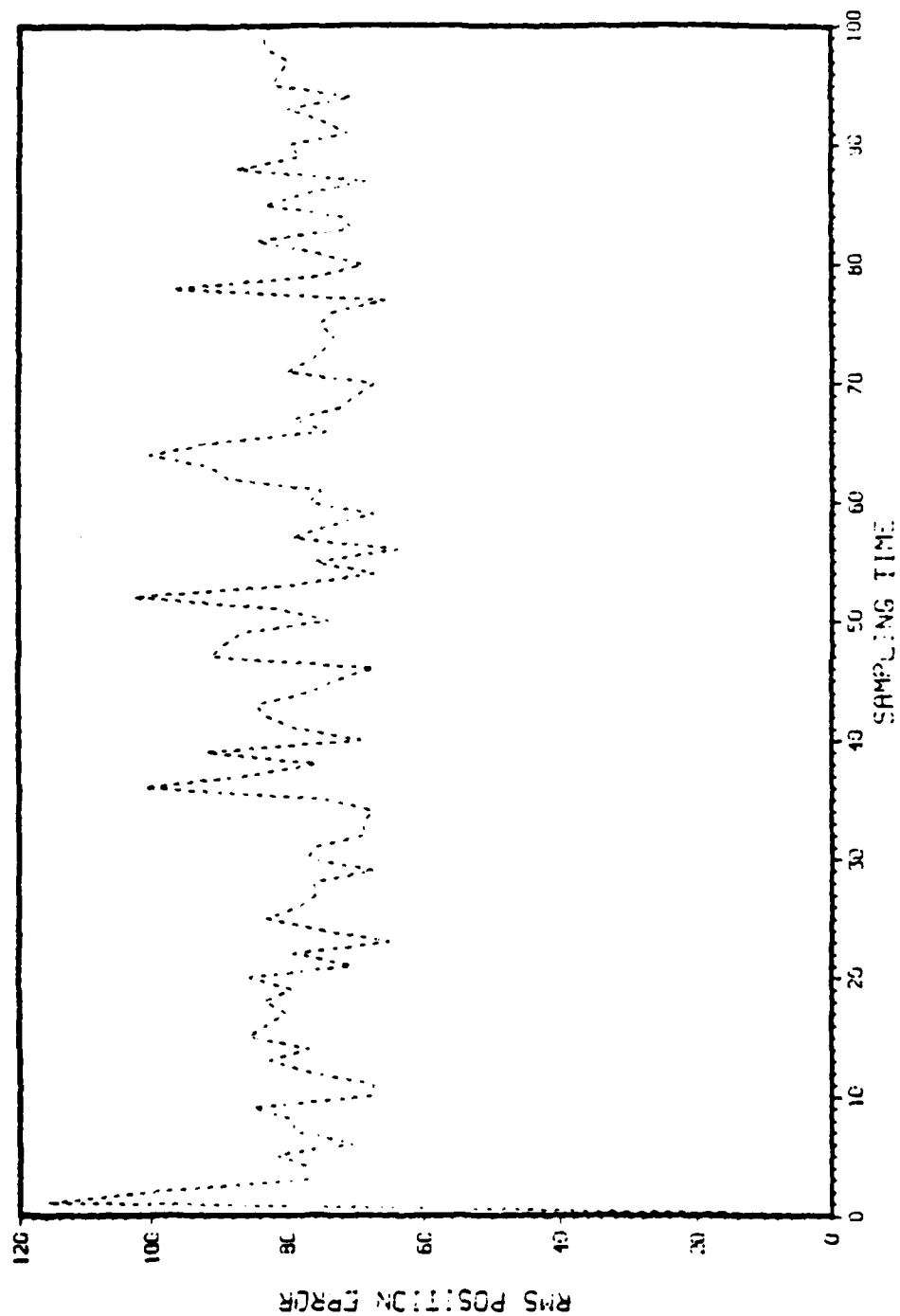


Figure 6.50 Mean Square Error of Pos. $Q' = 100$ $TR = .1$

MEAN SQUARE ERROR OF POS. $Q = 1000$ $TR = .1$

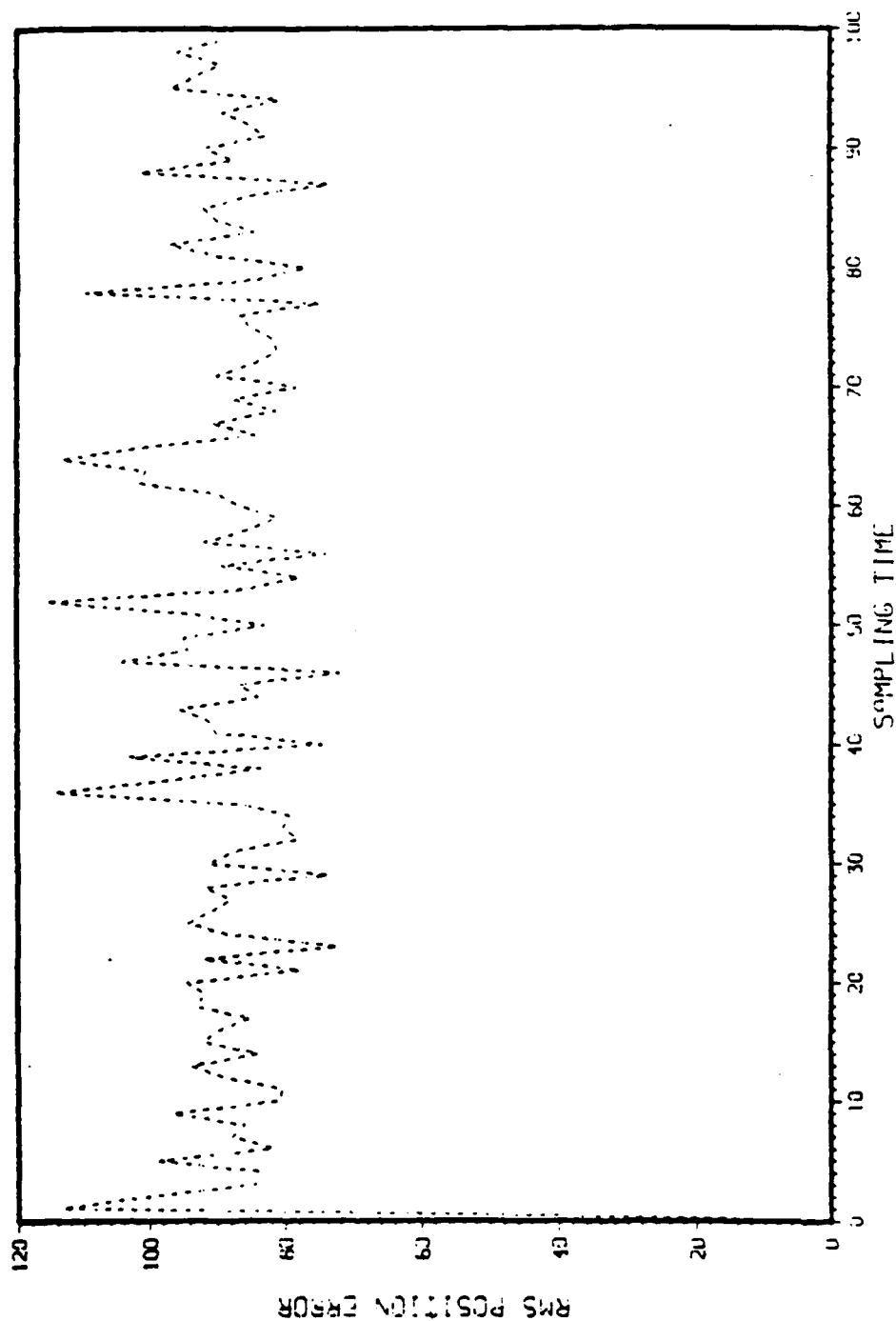


Figure 6.51 Mean Square Error of Pos. $Q' = 1000$ $TR = .1$

MEAN SQUARE ERROR OF POSIT. $Q = .1$ $TR = 3$

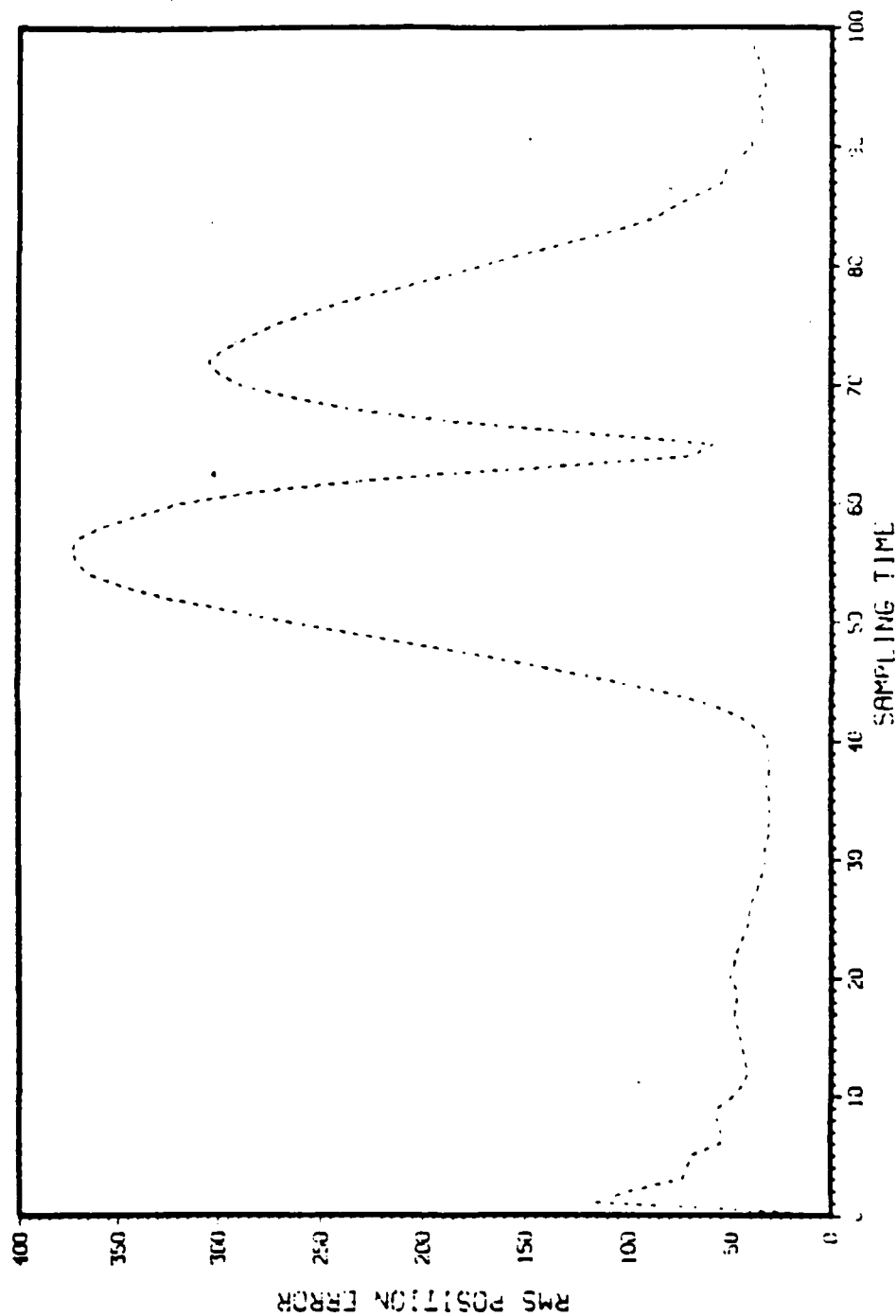


Figure 6.52 Mean Square Error of Pos. $Q' = .1$ $TR = 3$

MEAN SQUARE ERROR OF POSIT. $Q=4$ $TR=3$

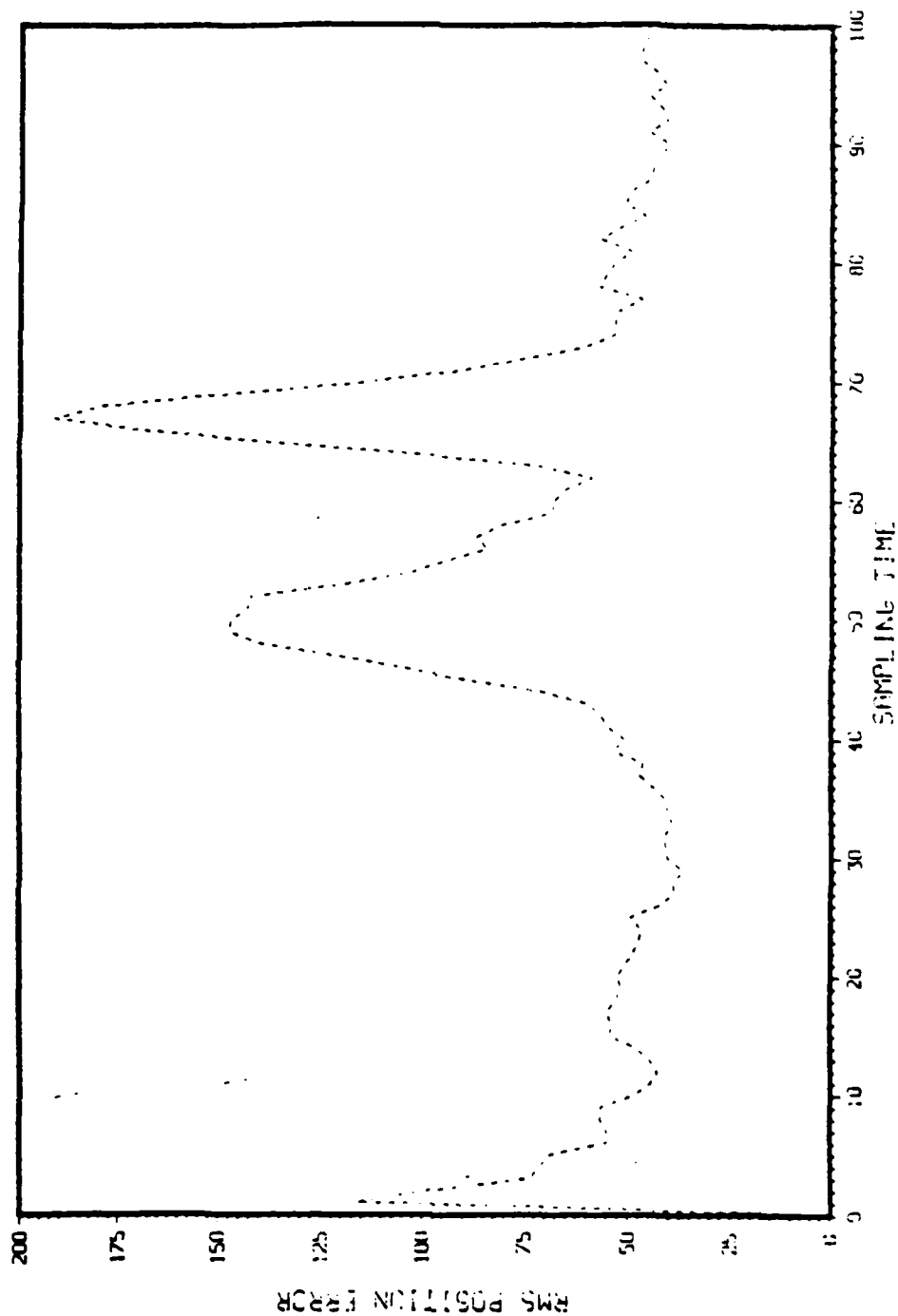


Figure 6.53 Mean Square Error of Pos. $Q' = 4$ $TR = 3$

MEAN SQUARE ERROR OF POSIT. $\sigma=10$ $TR=3$

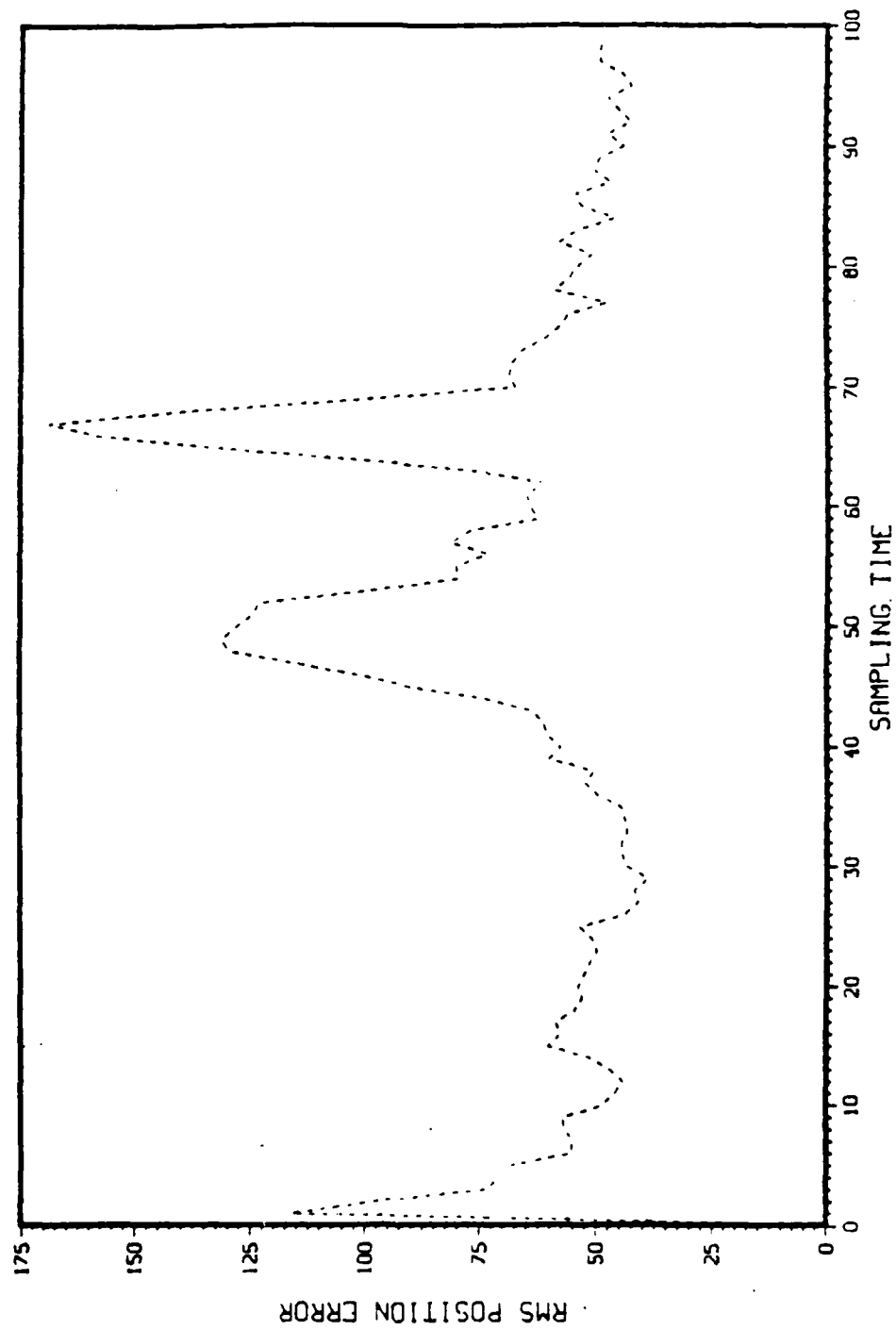


Figure 6.54 Mean Square Error of Pos. $\sigma=10$ $TR=3$

MEAN SQUARE ERROR OF POSIT. $Q=100$ $TR=3$

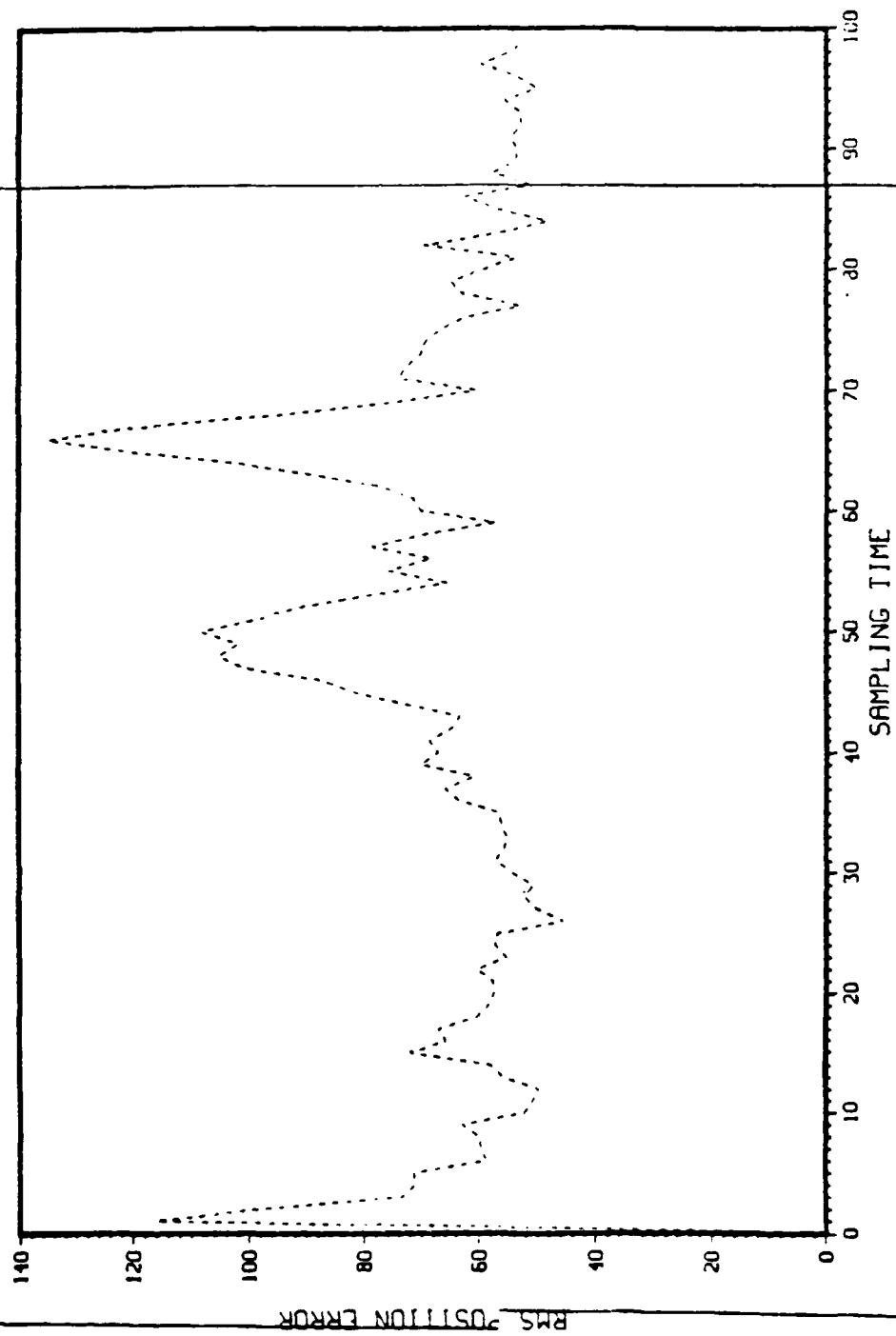


Figure 6.55 Mean Square Error of Pos. $Q=100$ $TR=3$

MEAN SQUARE ERROR OF POSIT. $Q=1000$ $TR=3$

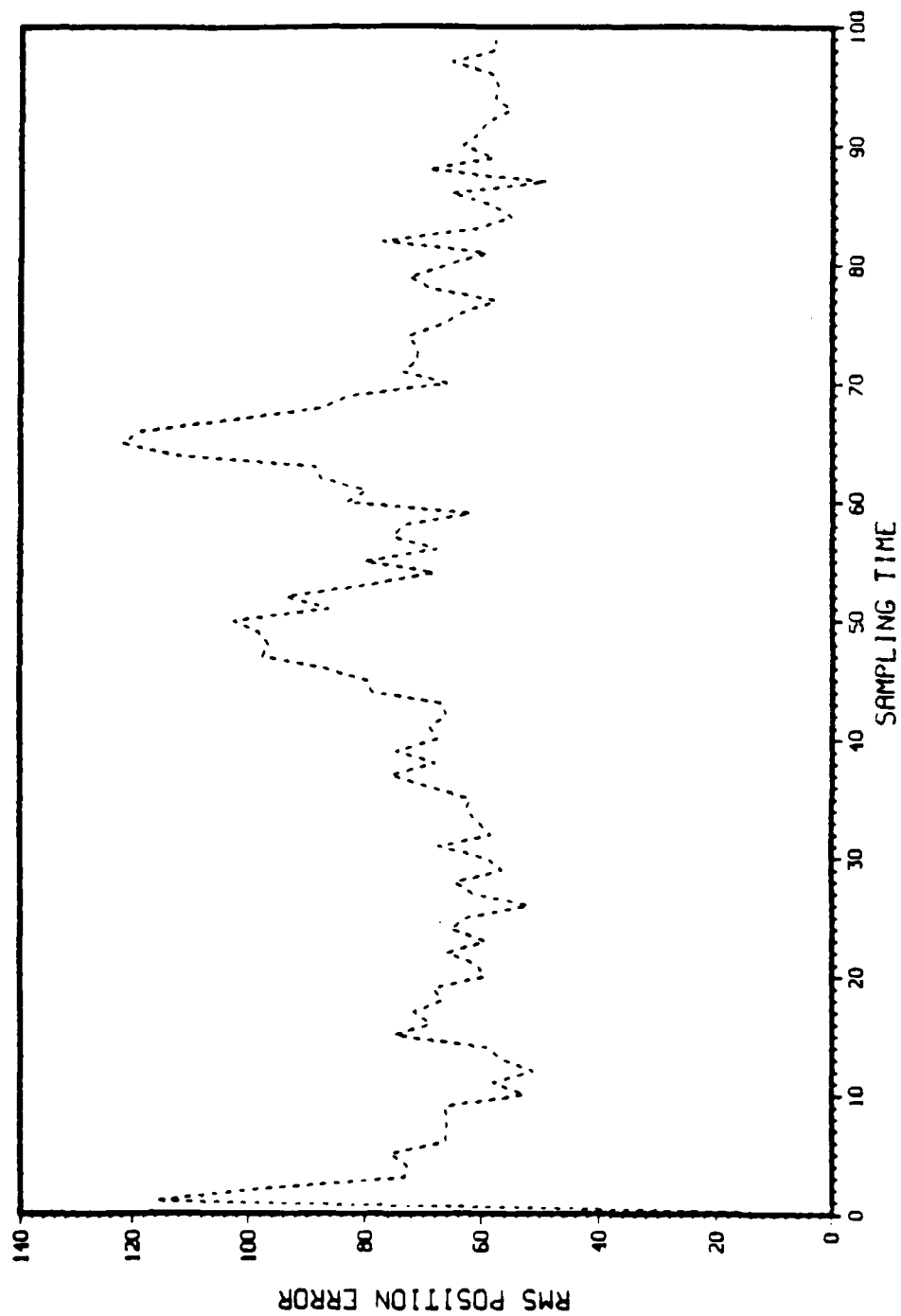


Figure 6.56 Mean Square Error of Pos. $Q=1000$ $TR=3$

MEAN SQUARE ERROR OF POSIT. $Q = 3000$ $TR = 3$

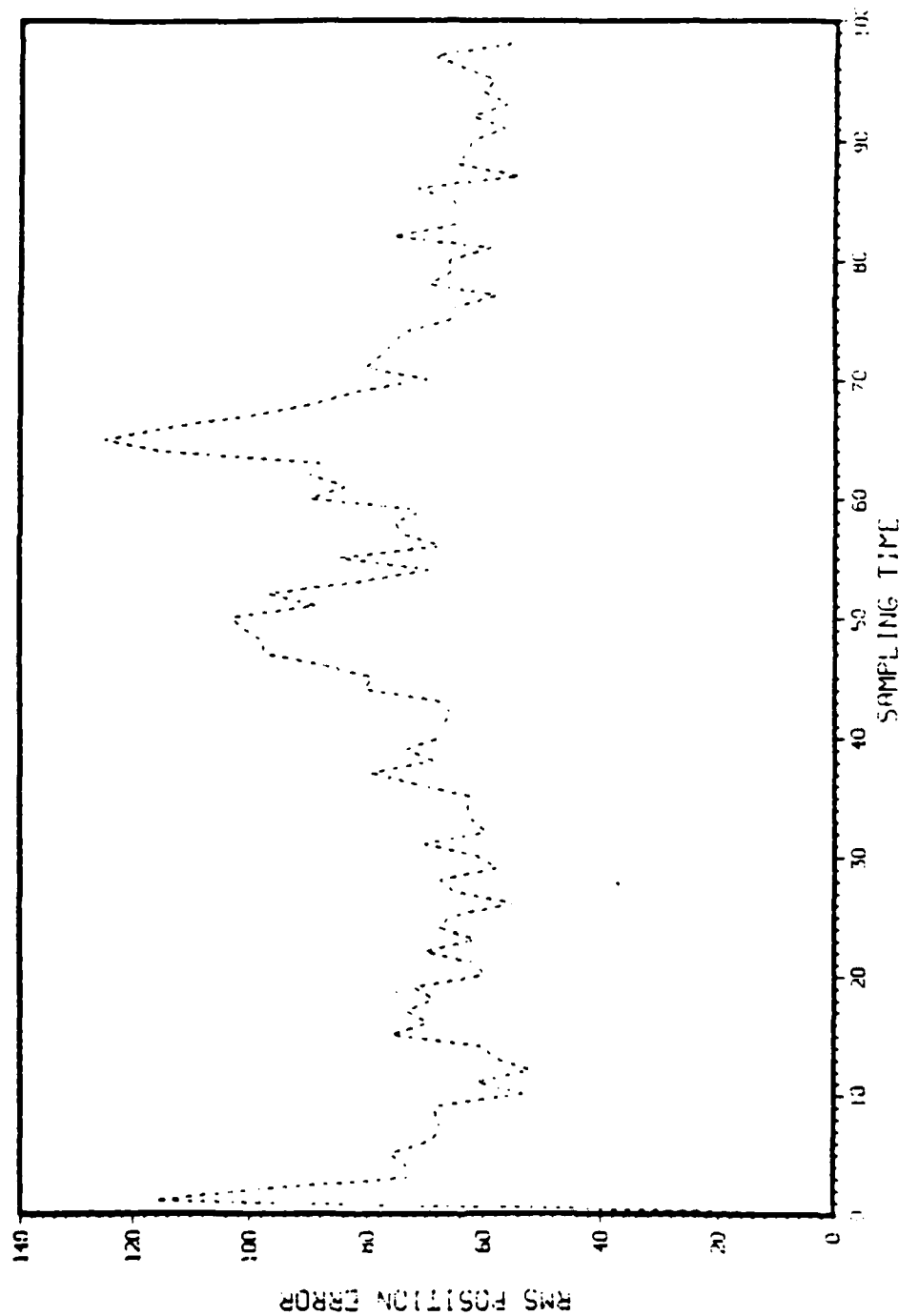


Figure 6.57 Mean Square Error of Pos. $Q = 3000$ $TR = 3$

MEAN SQUARE ERROR OF POS. $Q = .1$ $TR = 20$

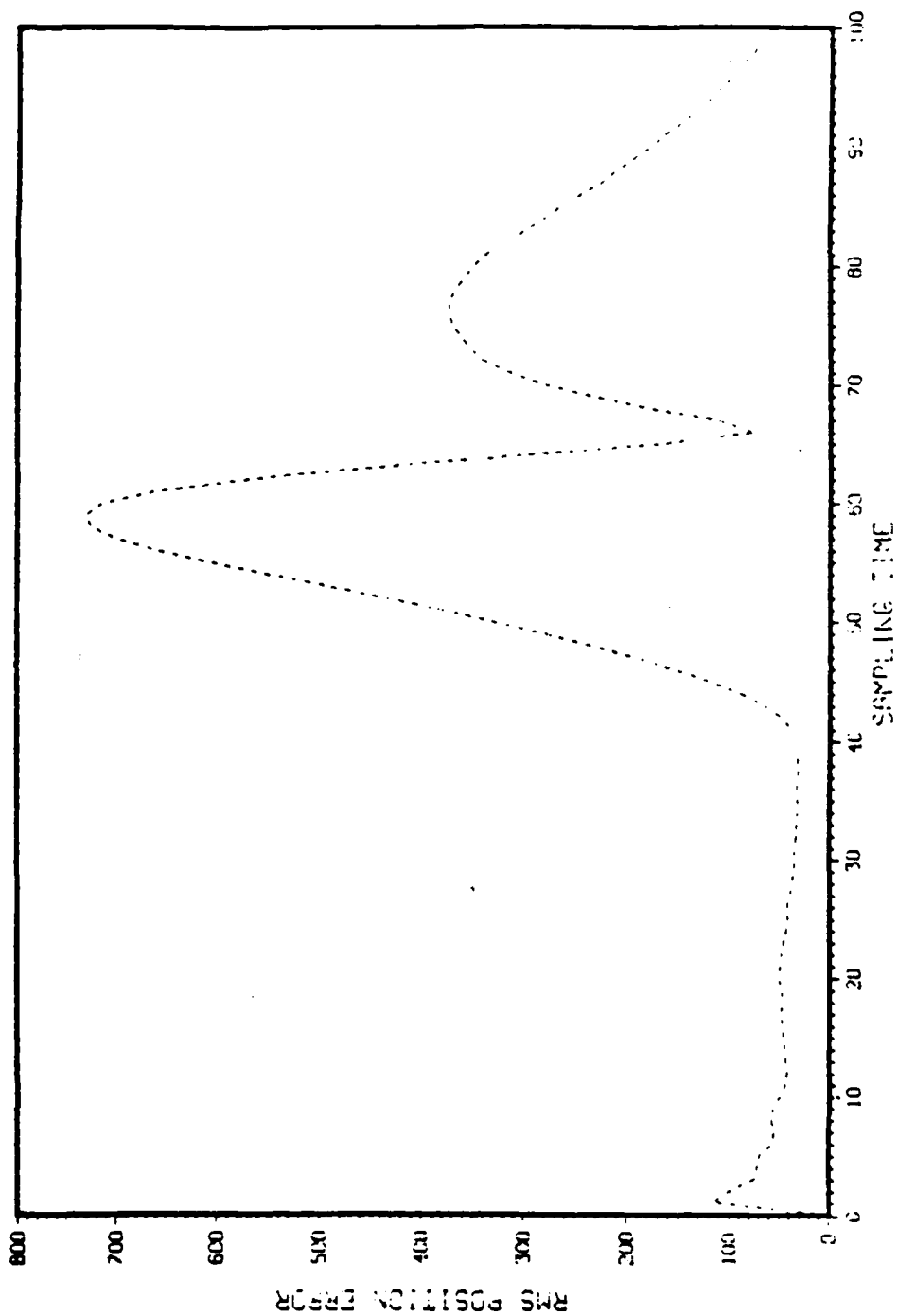


Figure 6.58 Mean Square Error of Pos. $Q = .1$ $TR = 20$

MEAN SQUARE ERROR OF POS. $Q=4$ $TR=20$

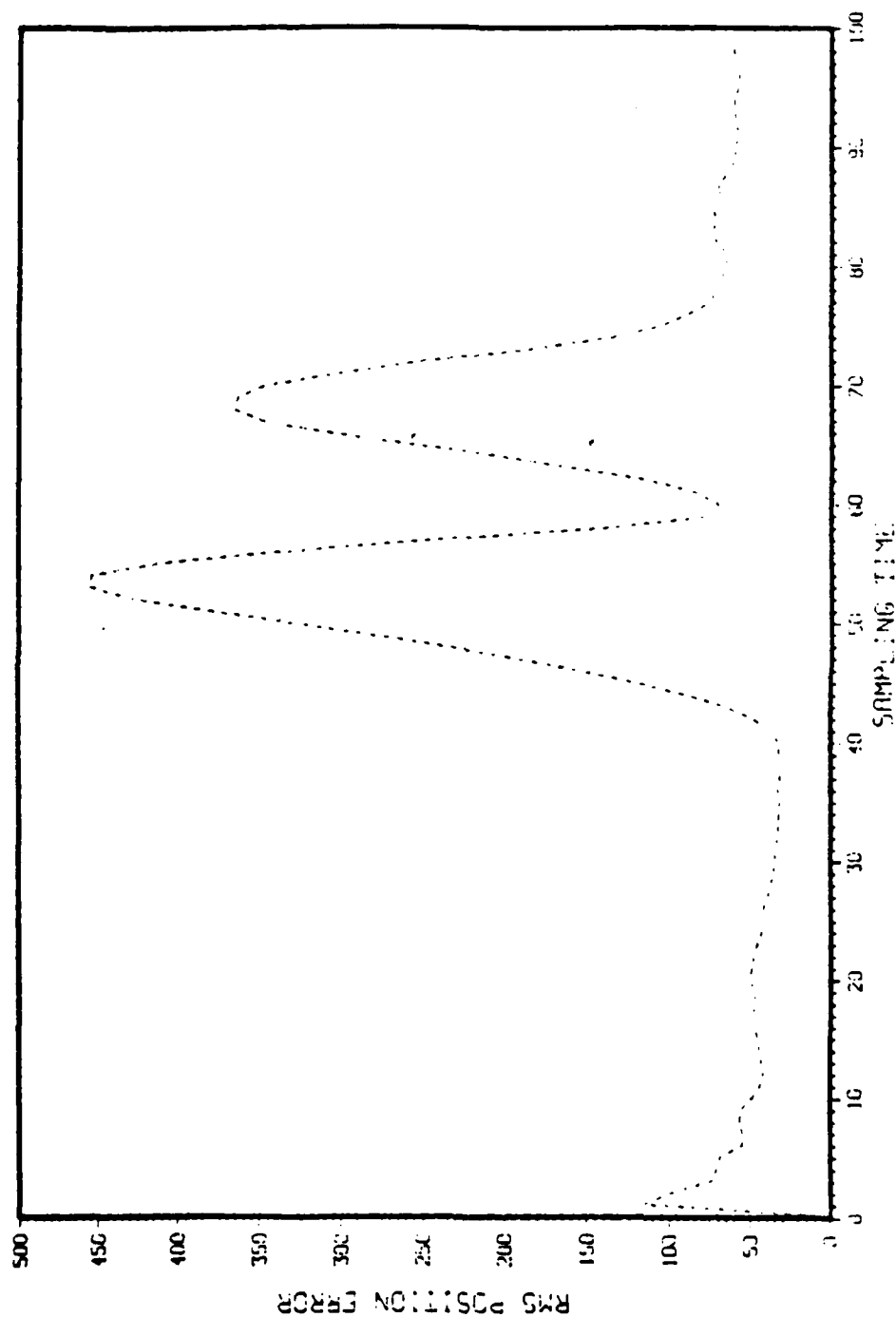


Figure 6.59 Mean Square Error of Pos. $Q=4$ $TR=20$

MEAN SQUARE ERROR OF POS. $Q=10$ $TR=20$

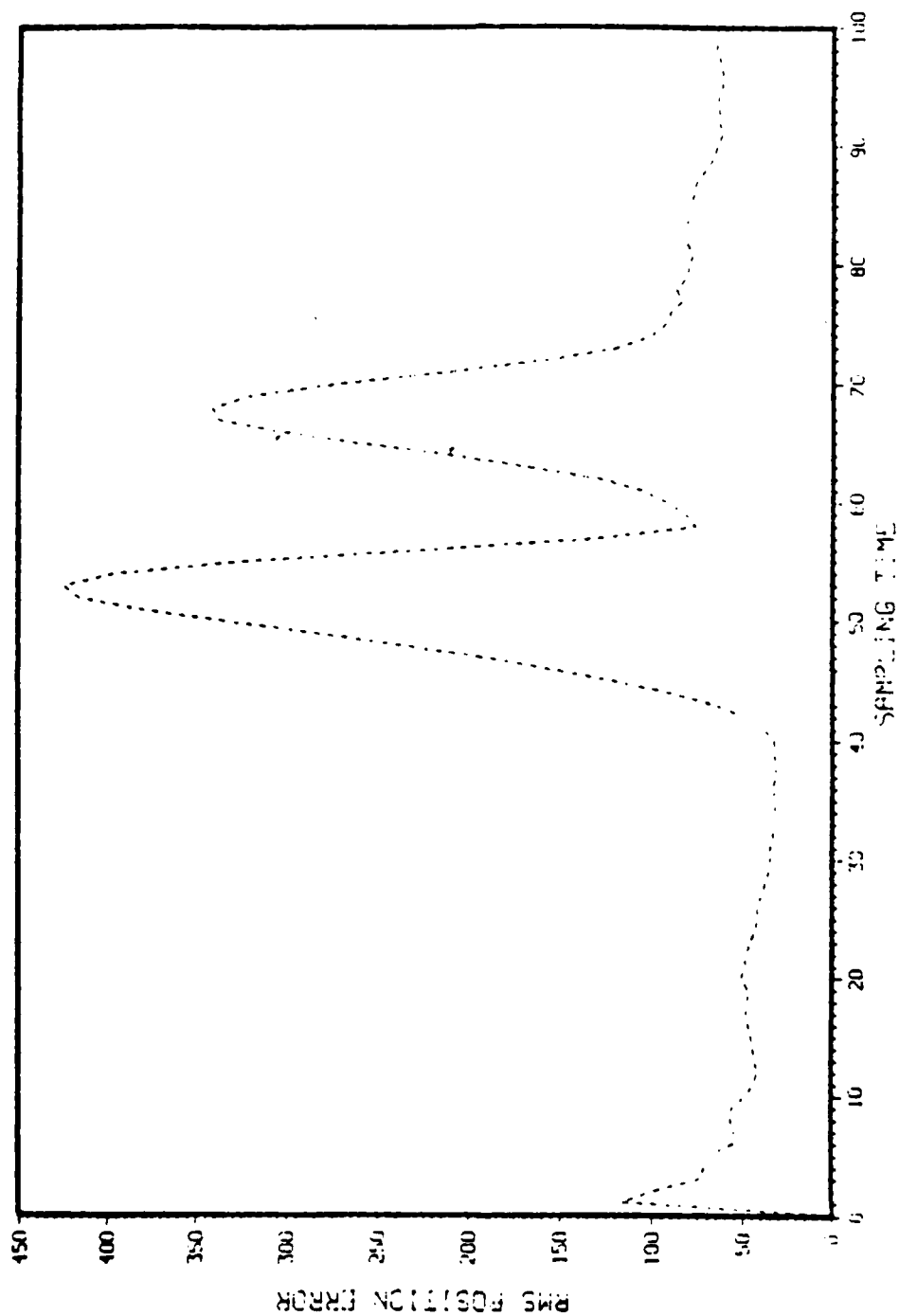


Figure 6.60 Mean Square Error of Pos. $Q=10$ $TR=20$

MEAN SQUARE ERROR OF POS. $C=100$ $TR=20$

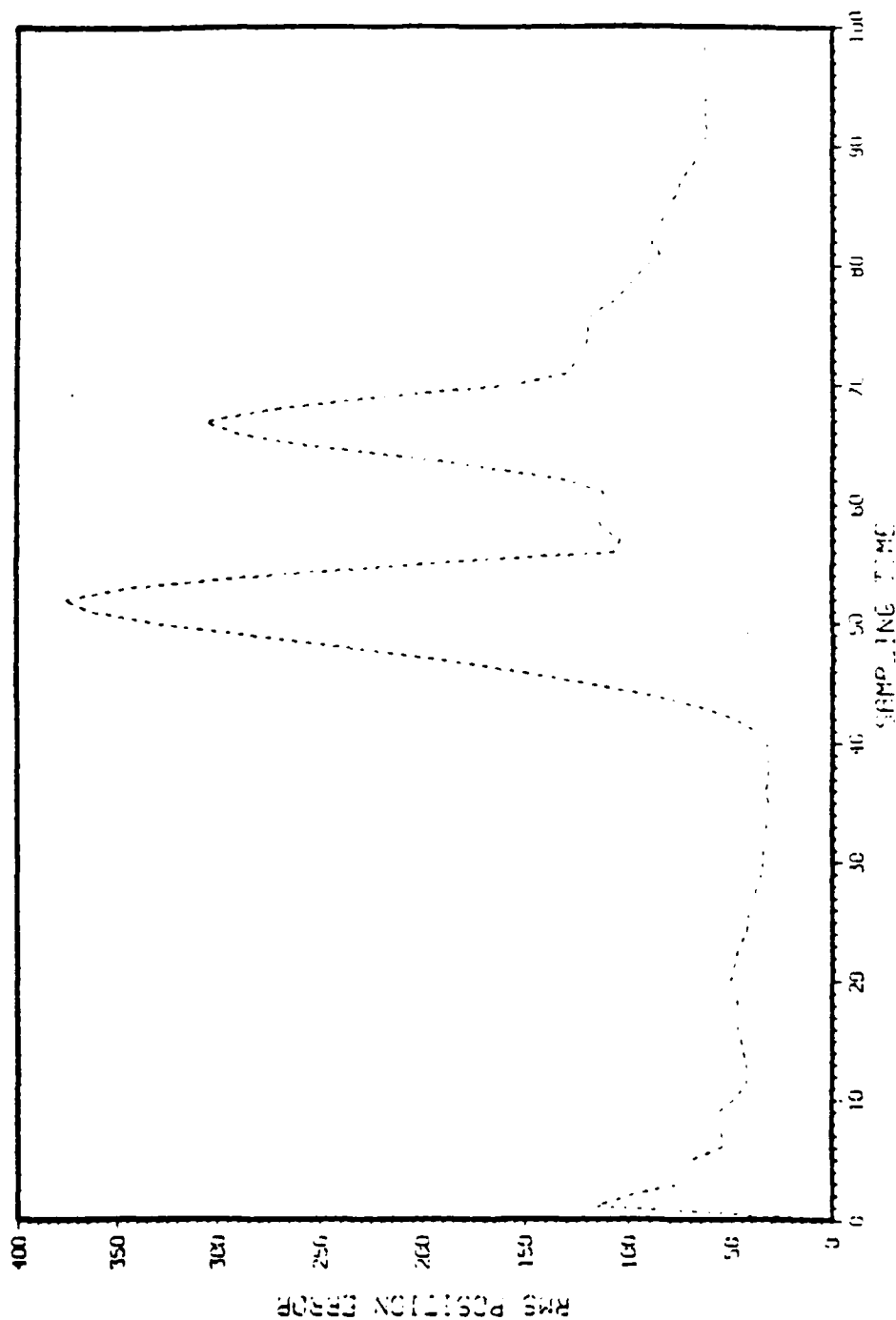


Figure 6.61 Mean Square Error of Pos. $C=100$ $TR=20$

MEAN SQUARE ERROR OF POS. $Q = 1000$ $TR = 20$

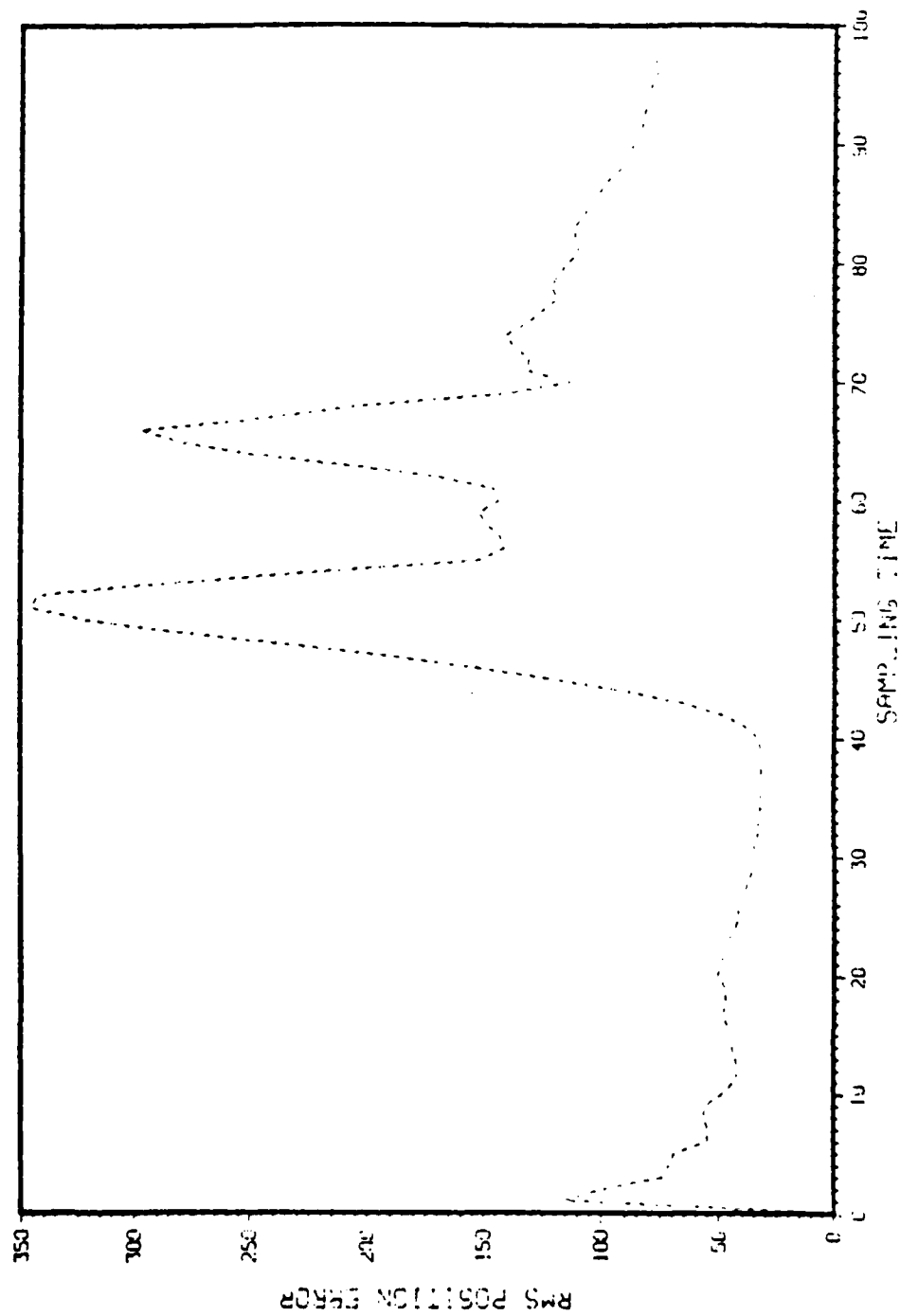


Figure 6.62 Mean Square Error of Pos. $Q = 1000$ $TR = 20$

Since the observed and estimated paths in the Monte Carlo case are averaged, the tracking results are better (smoother) and it is not easy to observe the various affections of extreme values of Q and the threshold level. For this reason, only three characteristic plots of this case are presented and the following conclusions are based mainly on the other two cases:

- a. When the target is moving with constant course and speed, it is desirable to have low values of Q and high threshold level.
- b. When the target is maneuvering, it is desirable to have medium or high values of Q and low threshold level.

Based on the above results, an attempt was made to run the filter using three levels of process noise Q (two threshold levels). In this way the two types of maneuvers (slow-fast) were treated separately. The threshold levels chosen were 1 and 5. The mean square error of position and velocity in the x direction over 50 Monte Carlo runs was plotted. The results (pp. 152-171) were very good especially for the position m.s.e. A tabulation of these results follows as Table VI.

TABLE VI
Multiple Process Noise Levels Results

Q	THR.	R.M.S. Position Error		R.M.S. Velocity Error	
		1-5		1-5	
0,0.1,100	1MAX	112		13.2	
	2MAX	140		14	
	3MIN	35		0.4	
0,3,100	1MAX	94		9.4	
	2MAX	124		13.6	
	3MIN	45		1.0	
0,10,100	1MAX	88		9.6	
	2MAX	102		13.9	
	3MIN	50		1.4	
0,0.1,10	1MAX	160		11.4	
	2MAX	208		13.6	
	3MIN	35		0.4	
0,1,10	1MAX	115		10	
	2MAX	168		13.2	
	3MIN	40		0.6	
0,3,10	1MAX	96		10	
	2MAX	140		13.4	
	3MIN	44		0.8	
0,0.1,1000	1MAX	136		21.5	
	2MAX	140		20.2	
	3MIN	37		0.25	
0,1,1000	1MAX	112		16	
	2MAX	130		17.4	
	3MIN	44		0.7	
0,3,1000	1MAX	100		13.5	
	2MAX	120		15.3	
	3MIN	46		1.0	
0,10,1000	1MAX	89		10.6	
	2MAX	106		15	
	3MIN	50		1.3	

MEAN SQUARE POSITION ERROR

MANY Q LEVELS

Q LEVELS: 0-.1-100

THRES. LEVELS: 1-5

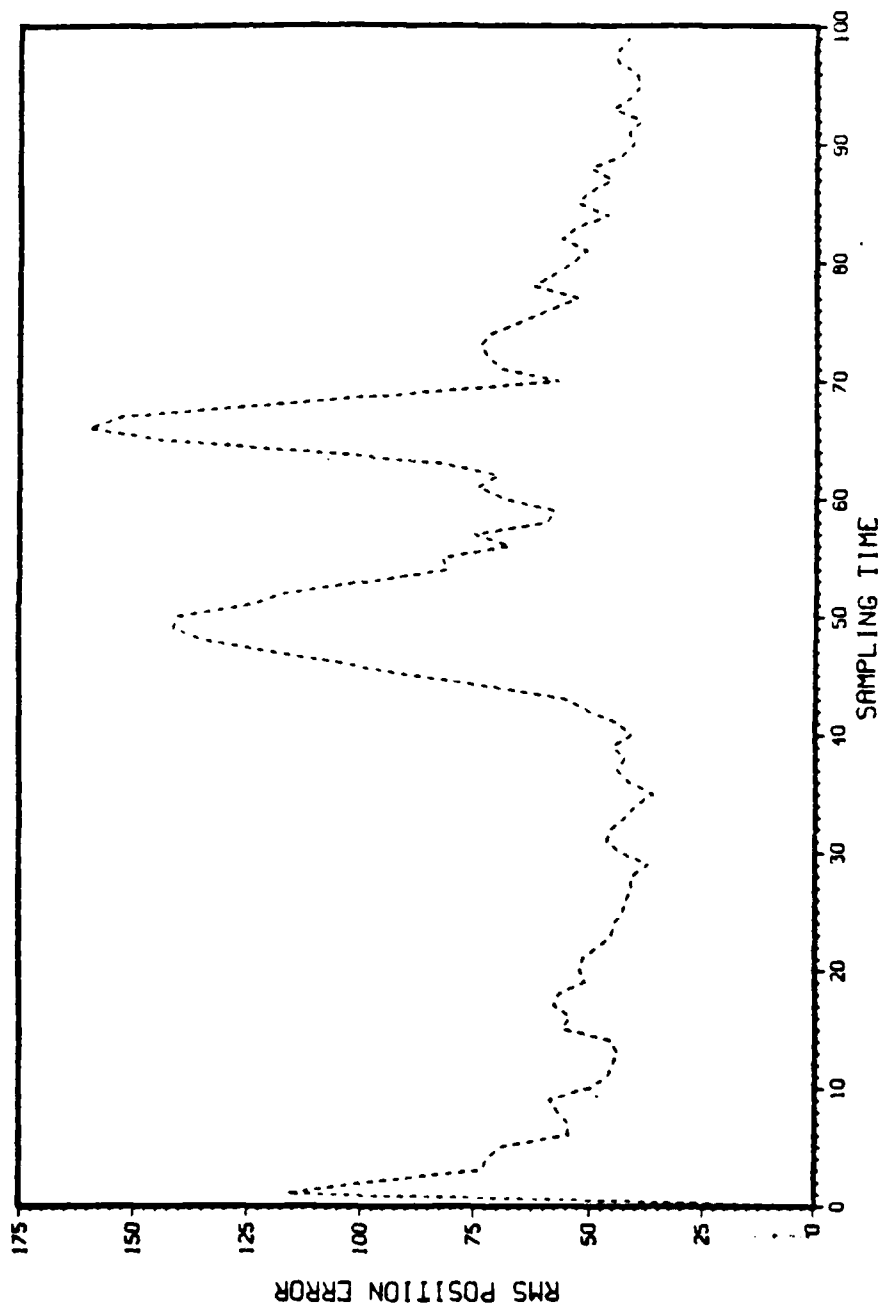


Figure 6.63 Mean Square Position Error; Many Q' Levels;
Q' Levels: 0,.1,100

MEAN SQUARE POSITION ERROR
 MANY Q LEVELS
 Q LEVELS: 0-3-100
 THRES. LEVELS: 1-5

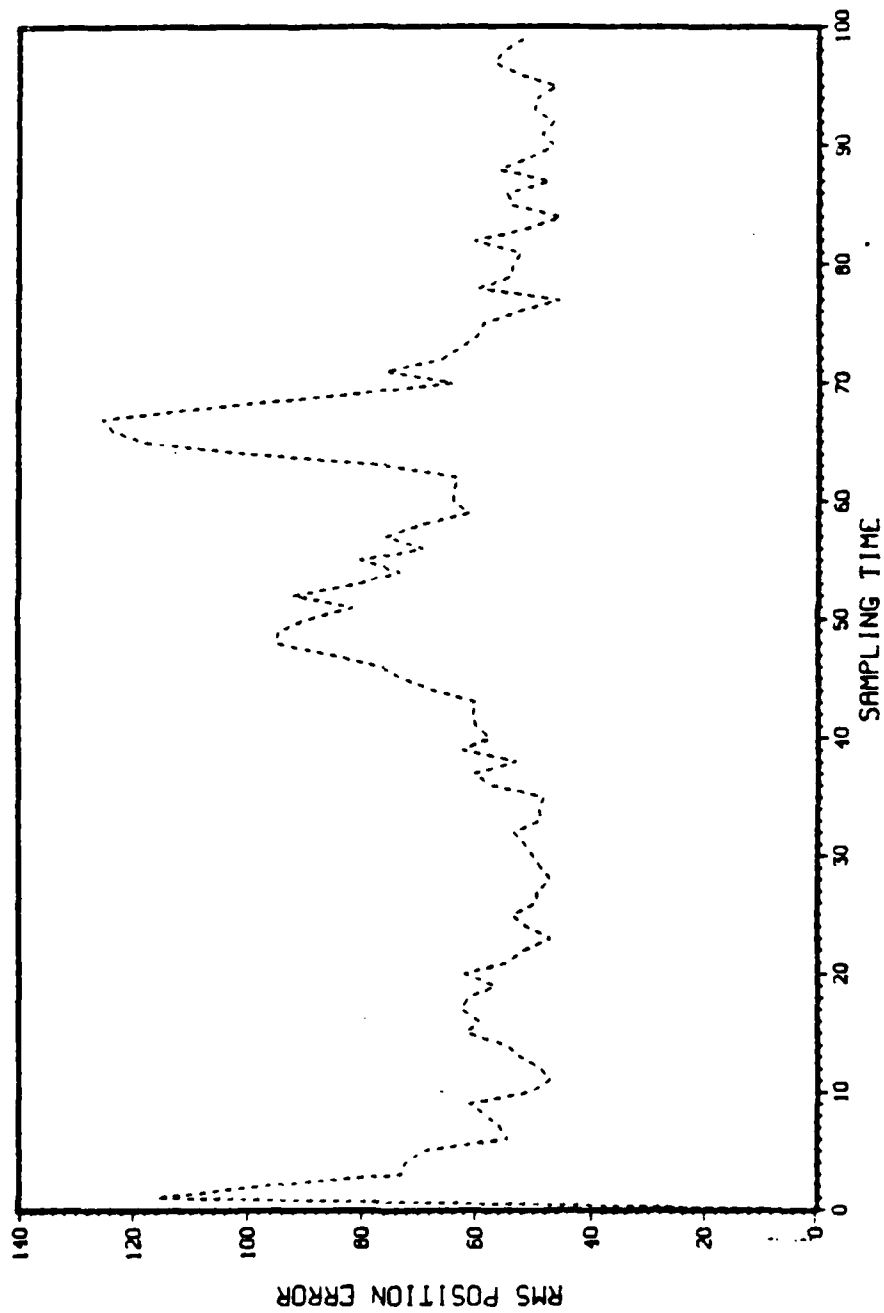


Figure 6.64 Mean Square Position Error; Many Q' Levels;
 Q' Levels: 0,3,100

MEAN SQUARE POSITION ERROR

MANY 0 LEVELS

0 LEVELS: 0-10-100

THRES. LEVELS: 1-5

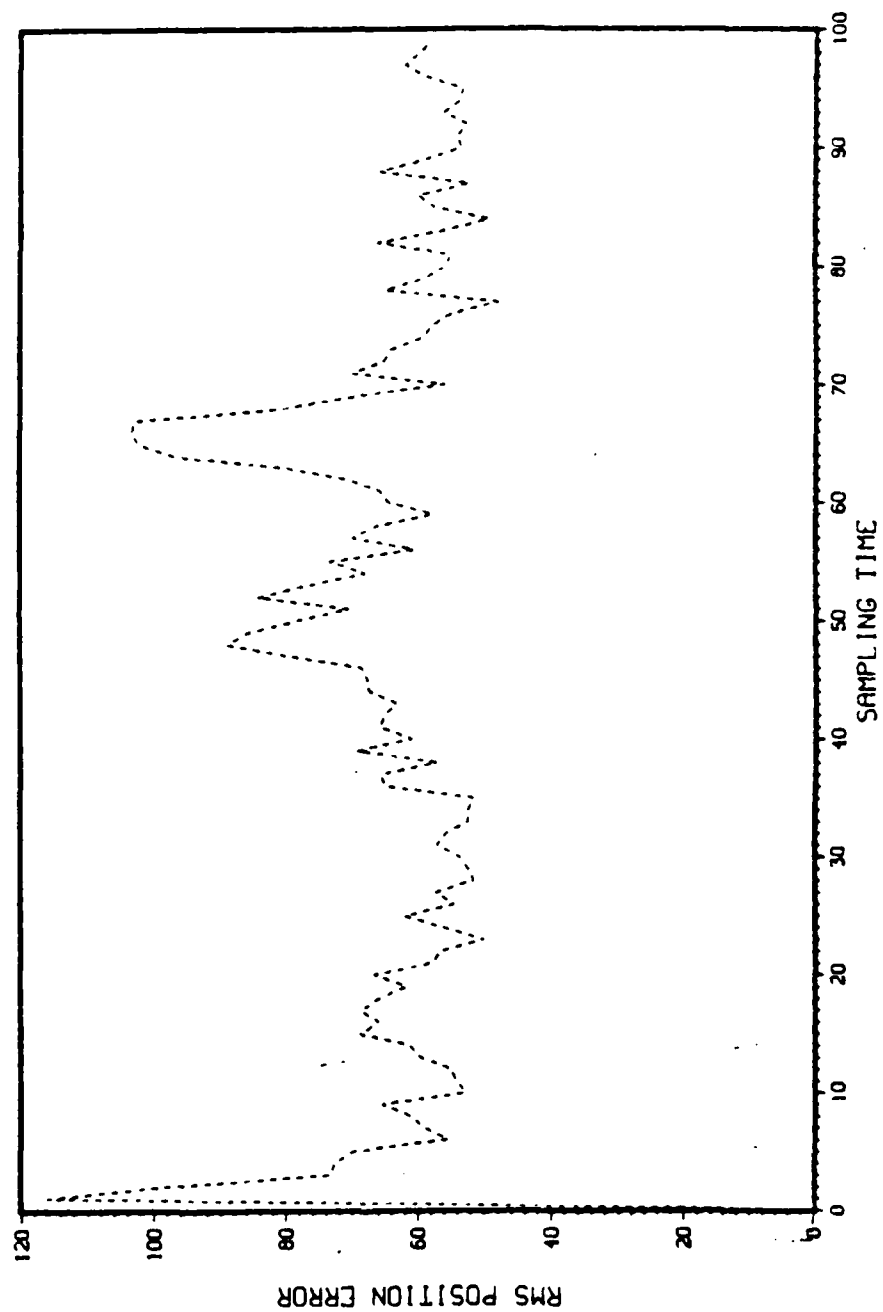


Figure 6.65 Mean Square Position Error; Many 0' Levels;
Q' Levels: 0,10,100

MEAN SQUARE POSITION ERROR

MANY Q LEVELS

Q LEVELS: 0-.1-10

THRES. LEVELS: 1-5

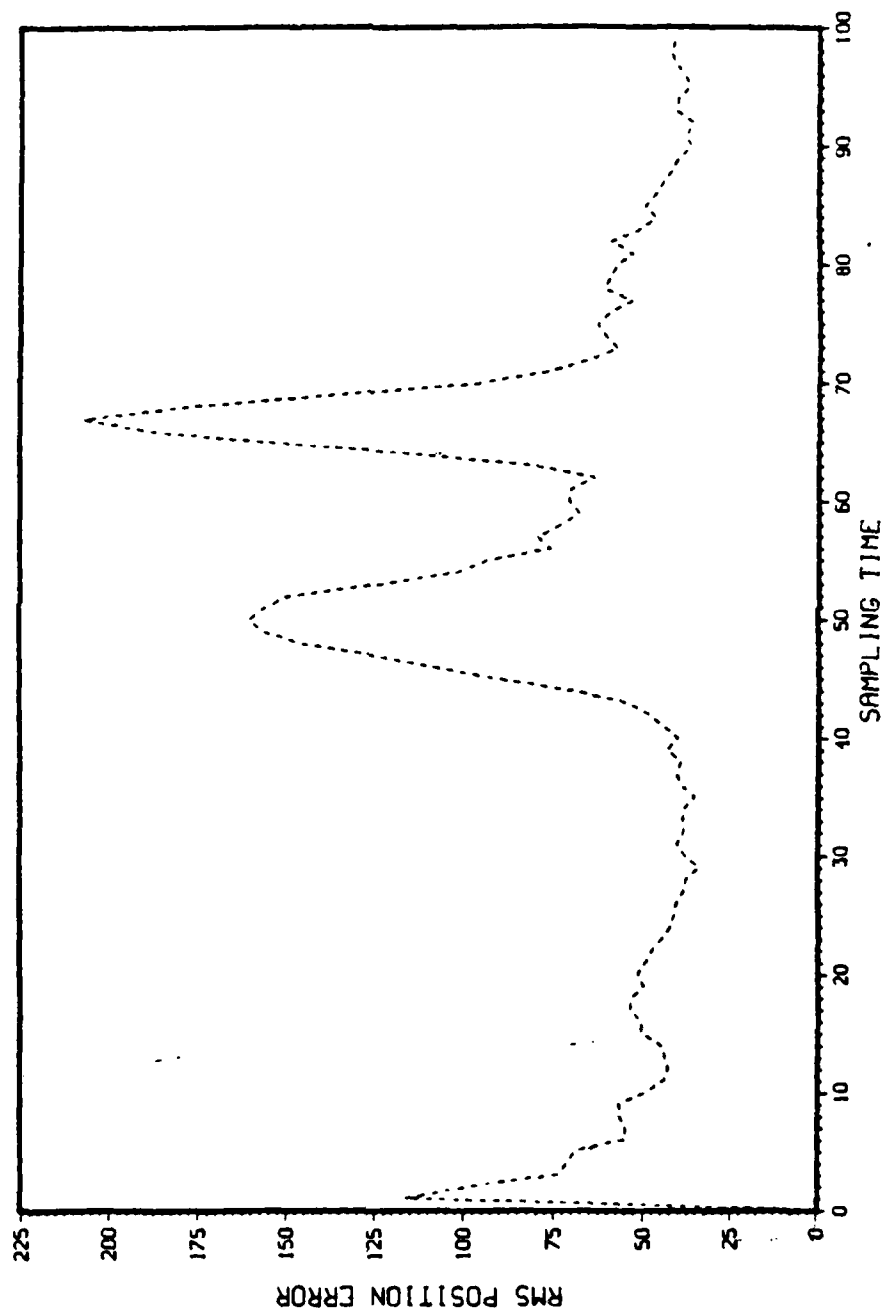


Figure 6.66 Mean Square Position Error; Many Q' Levels;
Q' Levels: 0,.1,10

MEAN SQUARE POSITION ERROR
MANY 0 LEVELS
0 LEVELS: 0-1-10
THRES. LEVELS: 1-5

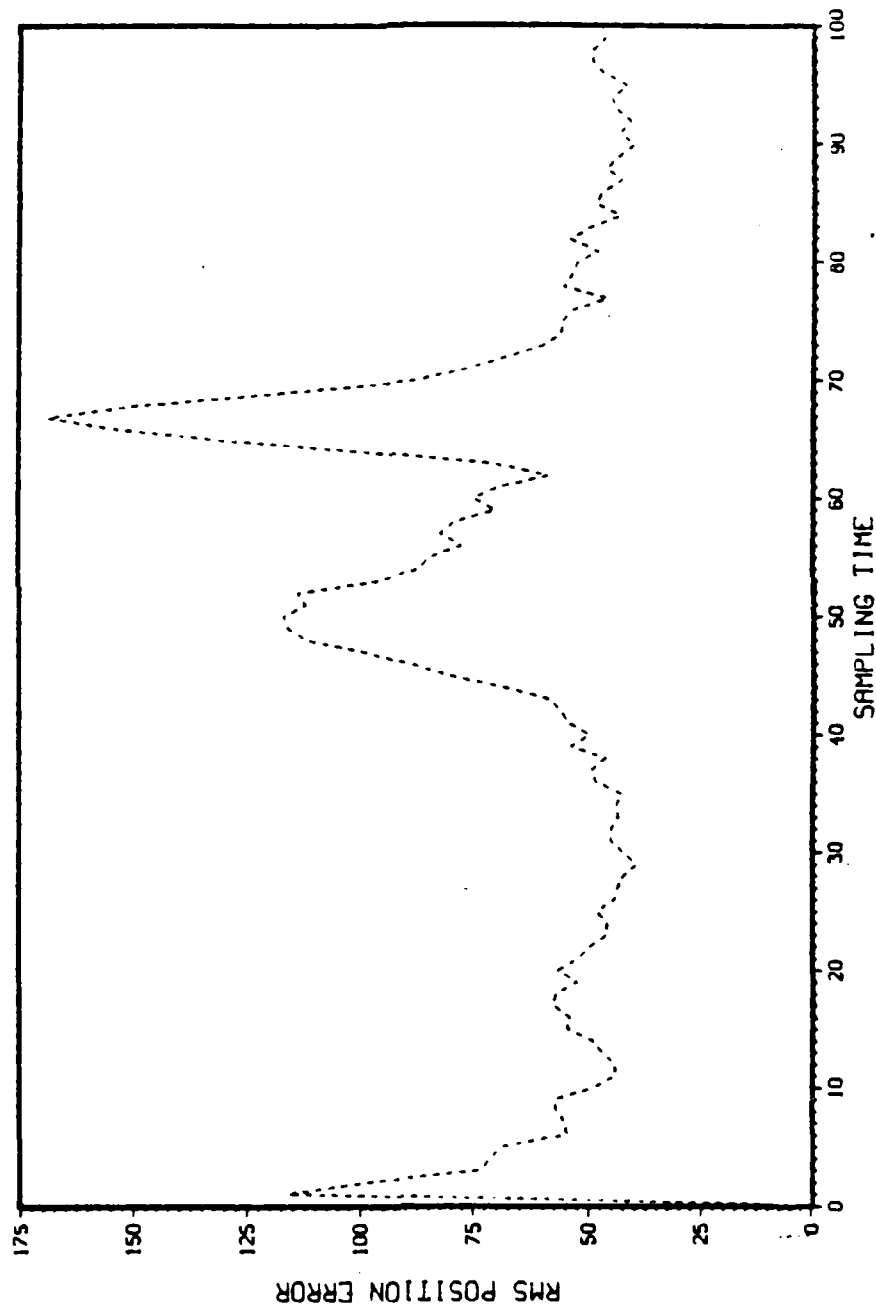


Figure 6.67 Mean Square Position Error; Many 0' Levels;
Q' Levels: 0,1,10

MEAN SQUARE POSITION ERROR

MANY 0 LEVELS

0 LEVELS: 0-3-10

THRES. LEVELS: 1-5

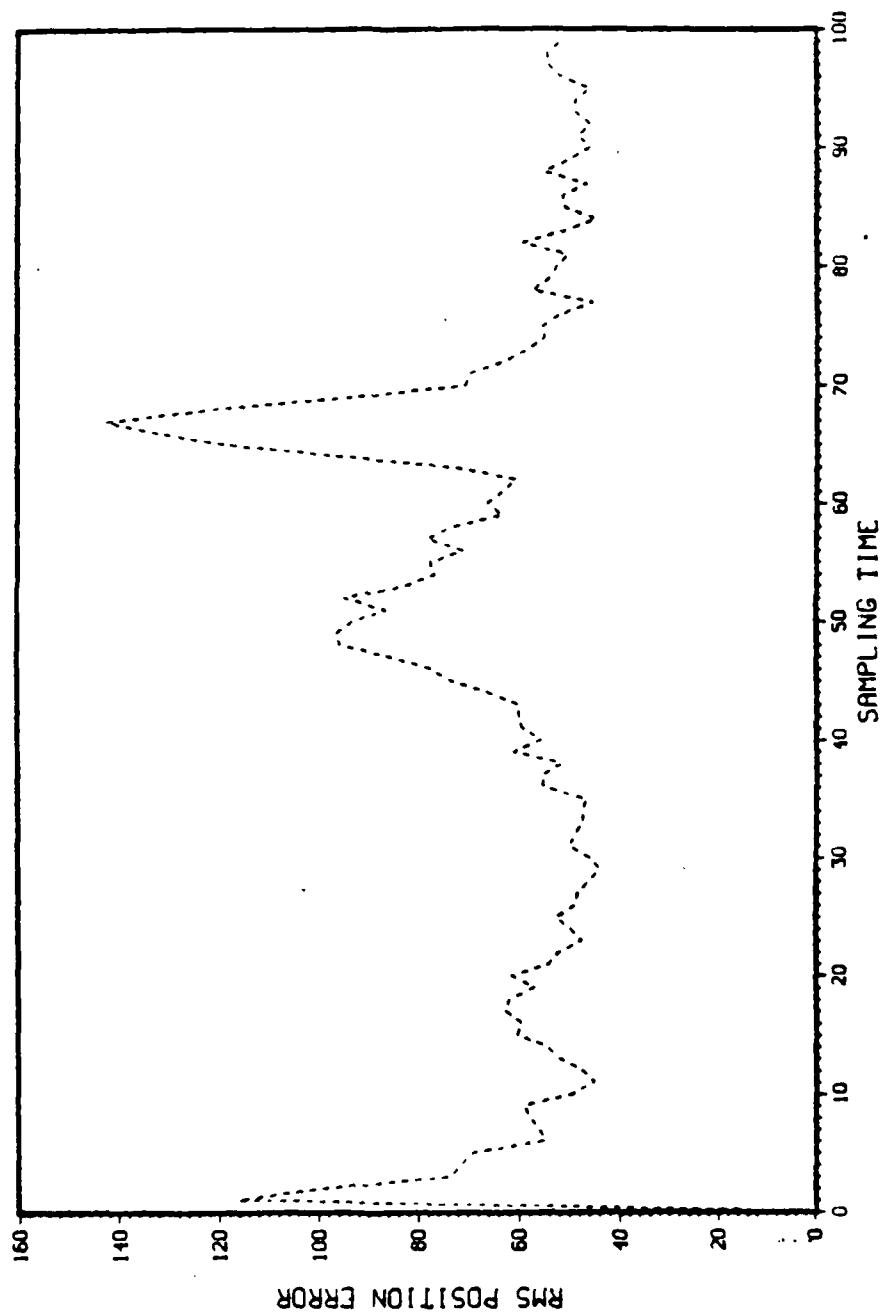


Figure 6.68 Mean Square Position Error; Many 0' Levels;
Q' Levels: 0,3,10

MEAN SQUARE POSITION ERROR
 MANY 0 LEVELS
 0 LEVELS: 0-.1-1000
 THRES. LEVELS: 1-5

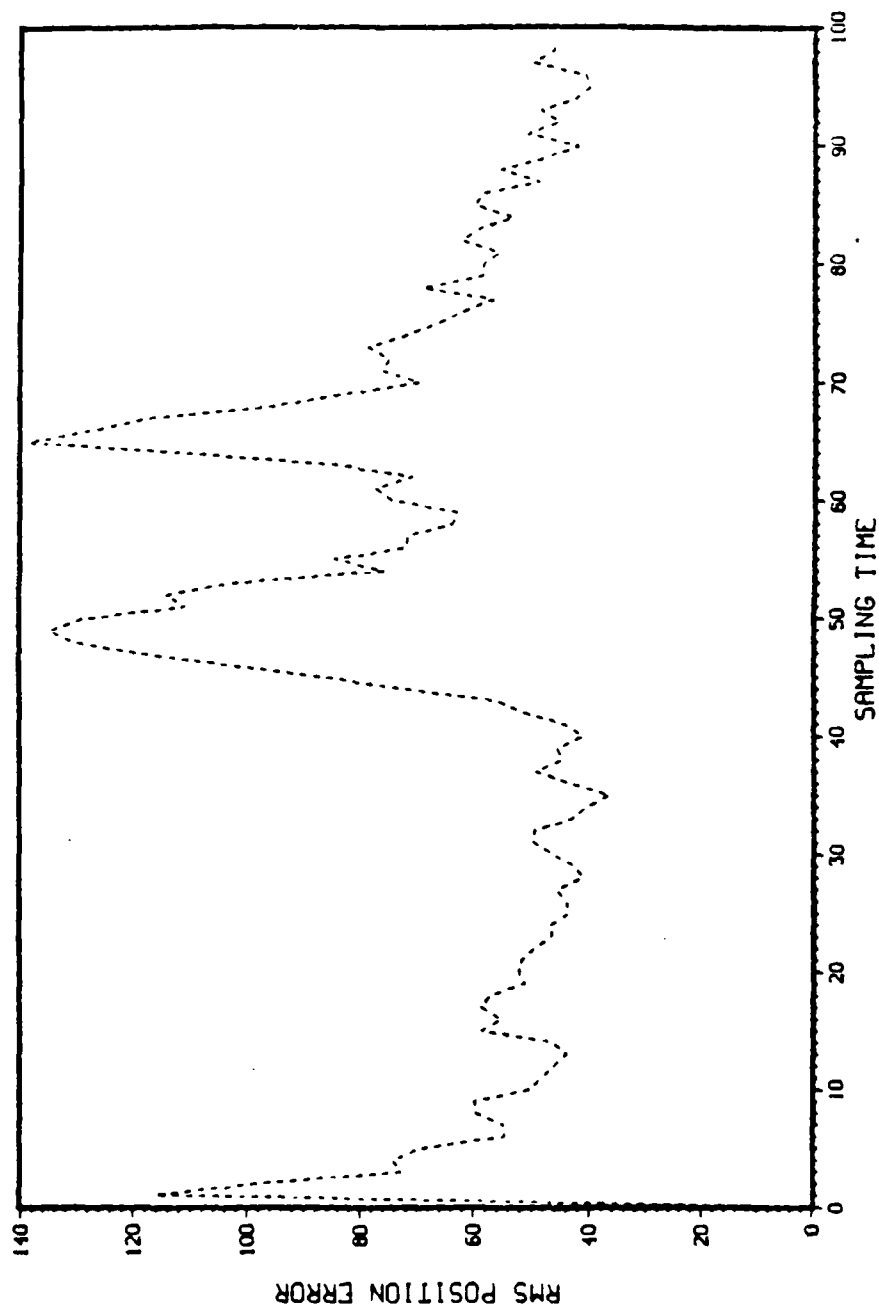


Figure 6.69 Mean Square Position Error; Many Q' Levels;
 Q' Levels: 0,.1,1000

MEAN SQUARE POSITION ERROR

MANY 0 LEVELS

Q LEVELS: 0-1-1000

THRES. LEVELS: 1-5

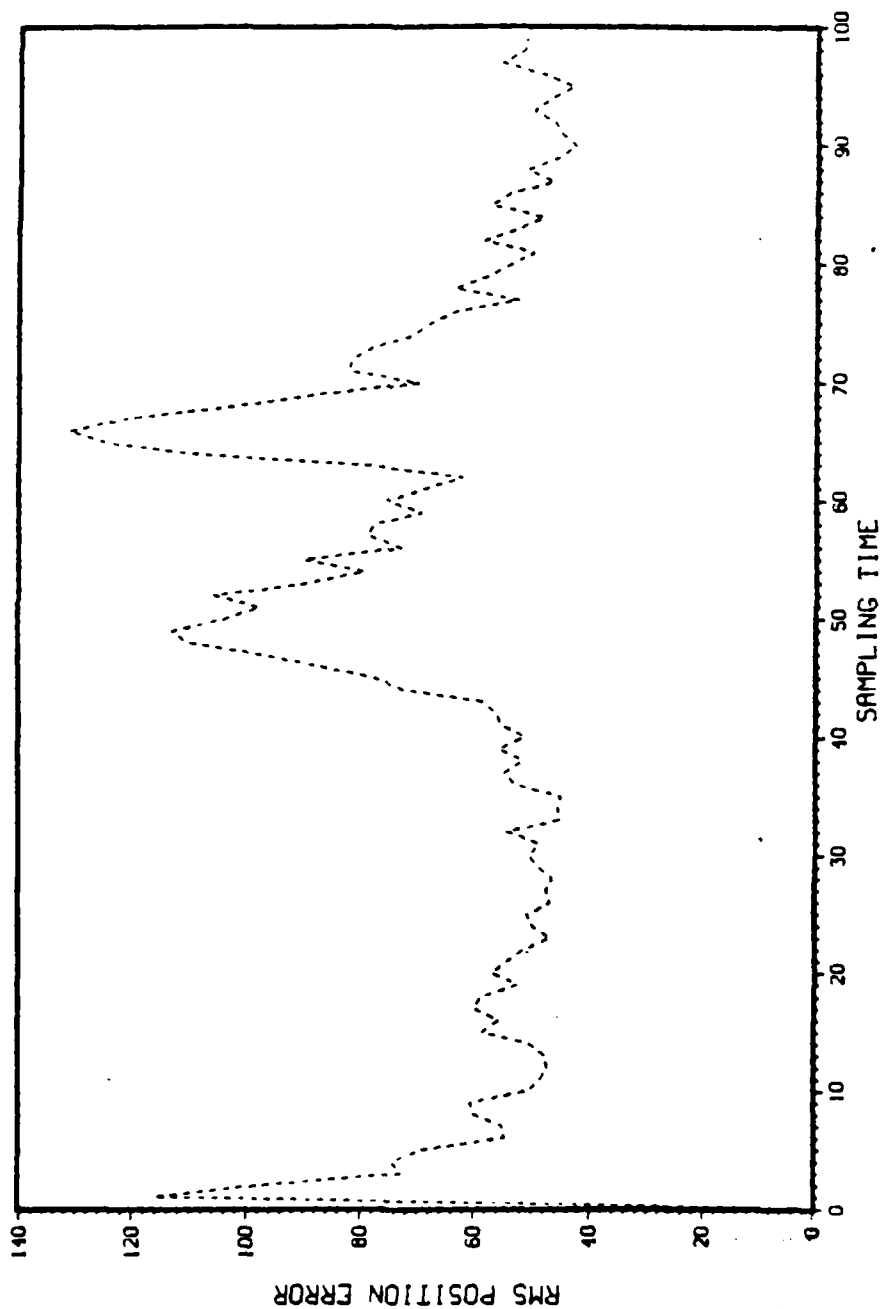


Figure 6.70 Mean Square Position Error; Many Q' Levels;
Q' Levels: 0,1,1000

MEAN SQUARE POSITION ERROR
MANY 0 LEVELS
0 LEVELS: 0-3-1000
THRES. LEVELS: 1-5

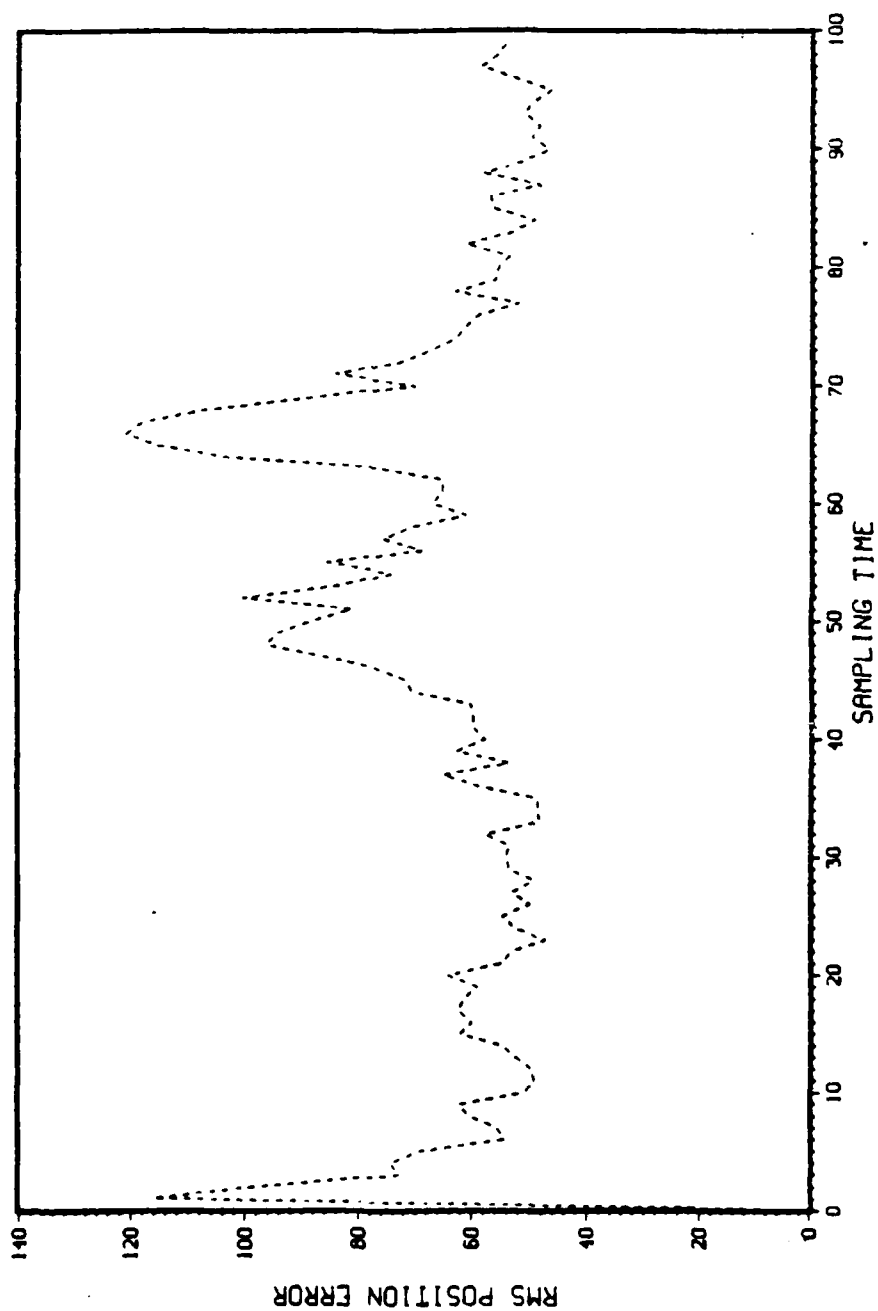


Figure 6.71 Mean Square Position Error; Many 0' Levels;
Q' Levels: 0,3,1000

MEAN SQUARE POSITION ERROR

MANY Q LEVELS

Q LEVELS: 0-10-1000

THRES. LEVELS: 1-5

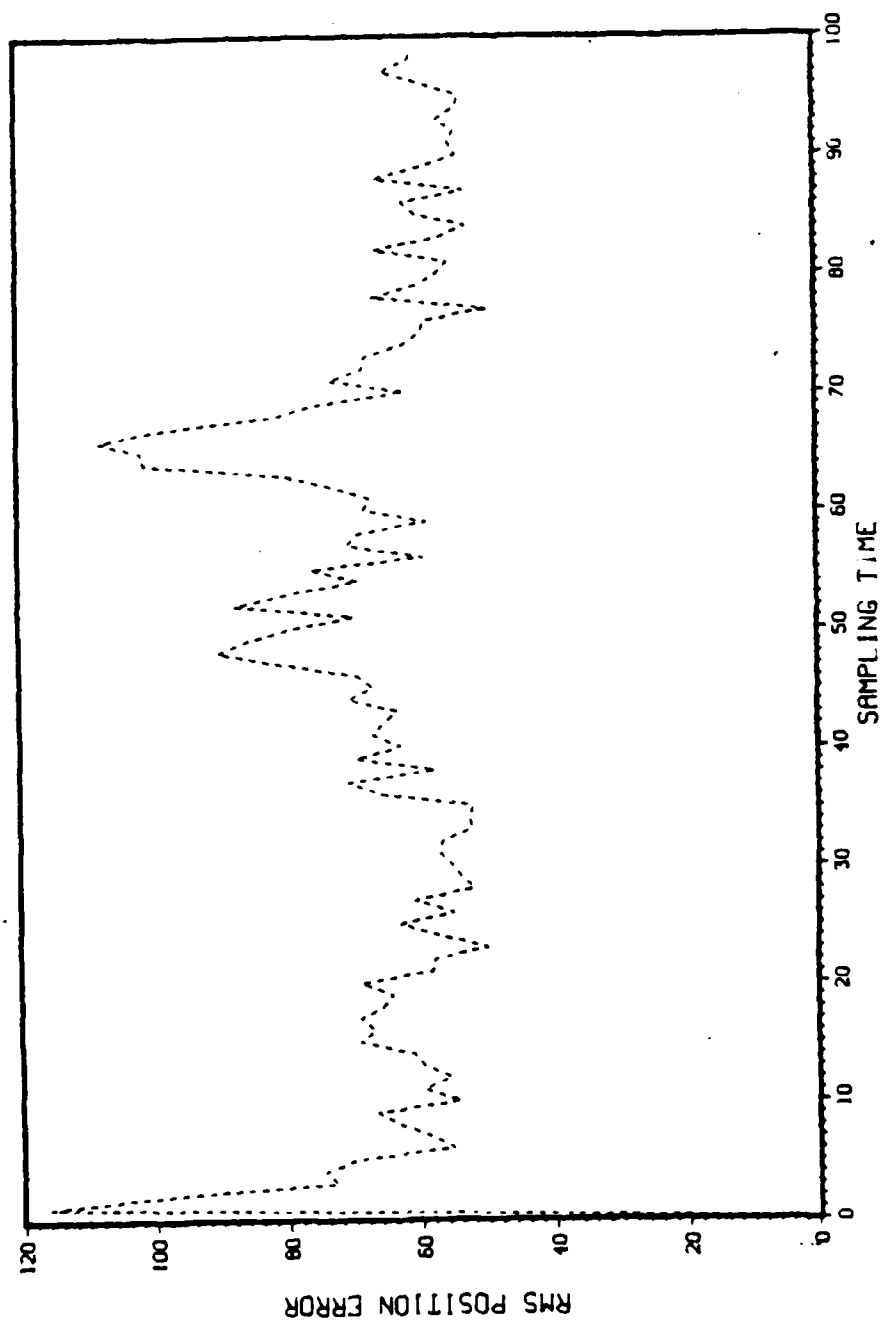


Figure 6.72 Mean Square Position Error; Many Q' Levels;
Q' Levels: 0,10,1000

MEAN SQUARE VELOCITY ERROR

MANY 0 LEVELS

Q LEVELS: 0-.1-100

THRES. LEVELS: 1-5

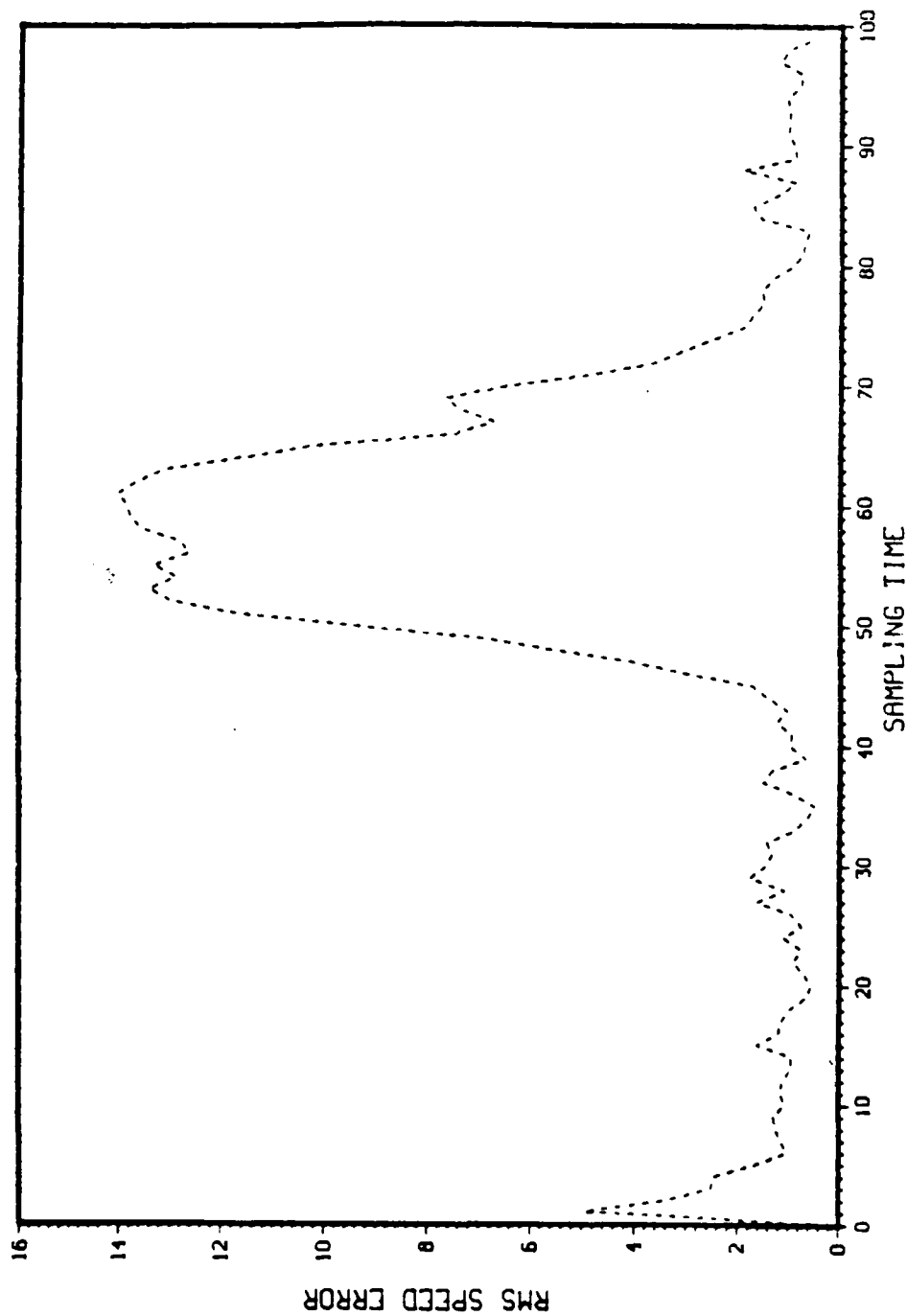


Figure 6.73 Mean Square Velocity Error; Many Q' Levels;
Q' Levels: 0,.1,100

MEAN SQUARE VELOCITY ERROR

MANY 0 LEVELS

0 LEVELS: 0-3-100

TWES. LEVELS: 1-5

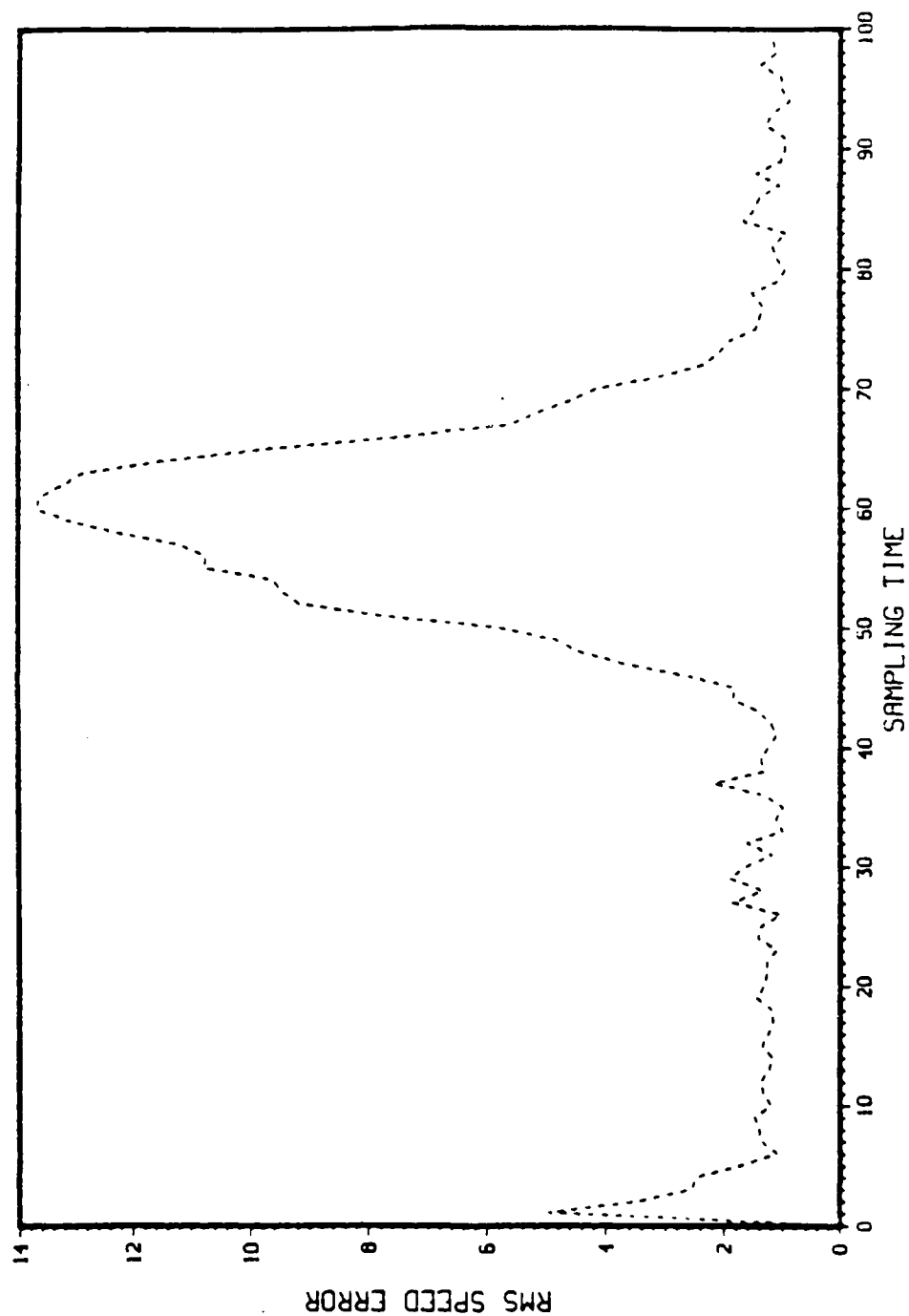


Figure 6.74 Mean Square Velocity Error; Many Q' Levels;
Q' Levels: 0,3,100

MEAN SQUARE VELOCITY ERROR

MANY 0 LEVELS

0 LEVELS: 0-10-100

THRES. LEVELS: 1-5

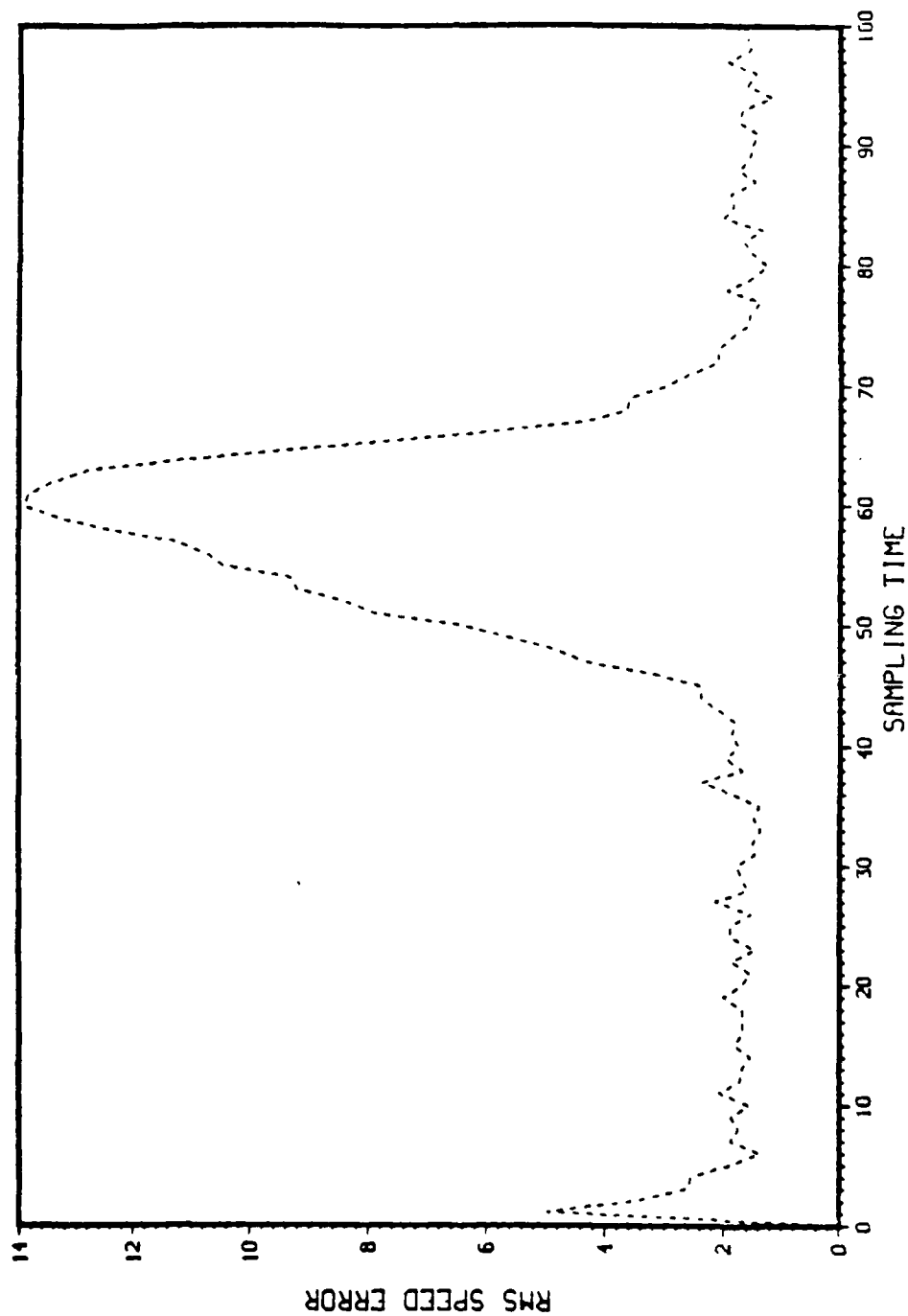


Figure 6.75 Mean Square Velocity Error; Many Q' Levels;
Q' Levels: 0,10,100

MEAN SQUARE VELOCITY ERROR

MANY 0 LEVELS

Q LEVELS: 0-.1-10

THRES. LEVELS: 1-5

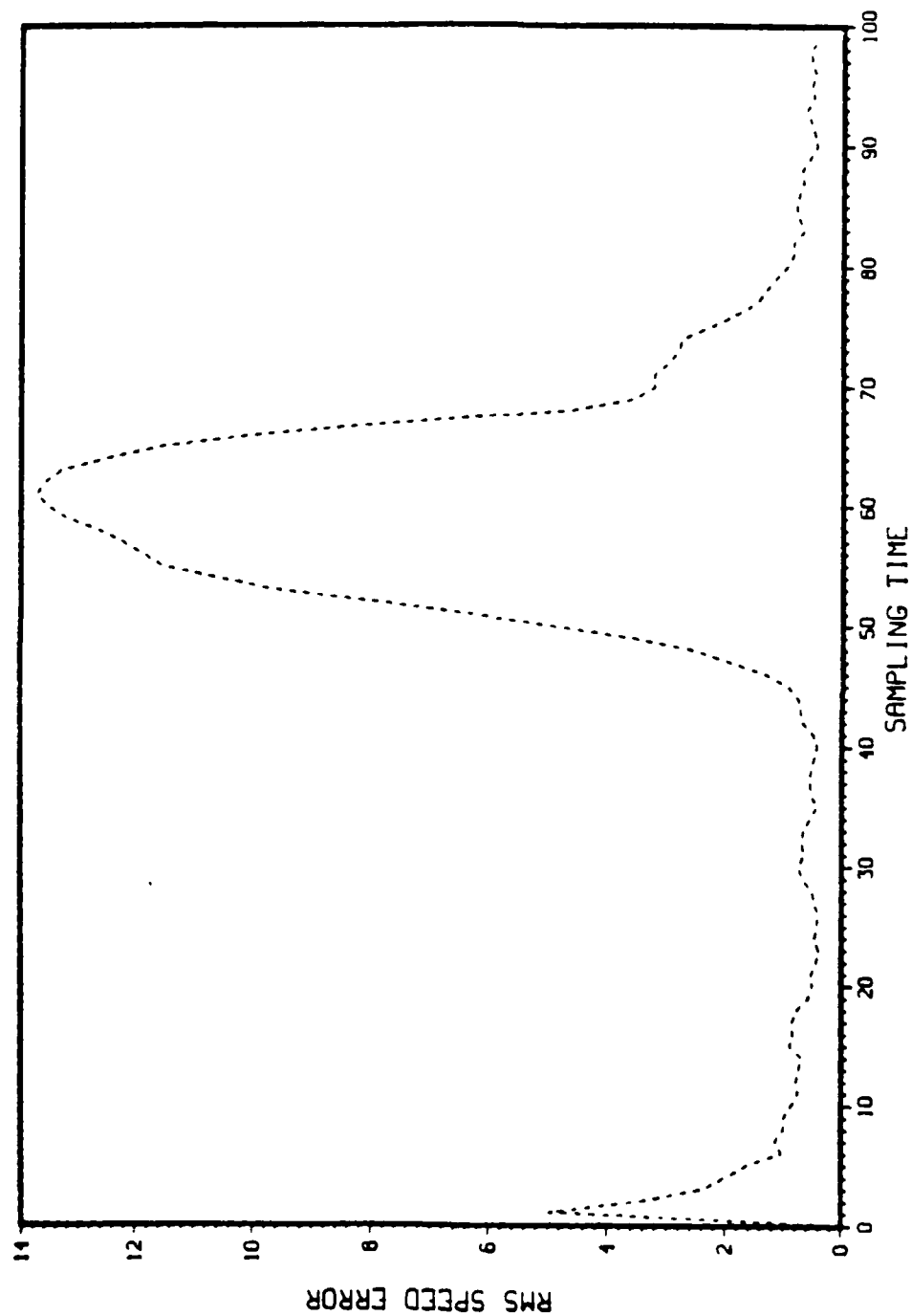


Figure 6.76 Mean Square Velocity Error; Many Q' Levels;
Q' Levels: 0,.1,10

MEAN SQUARE VELOCITY ERROR

MANY 0 LEVELS

0 LEVELS: 0-1-10

THRES. LEVELS: 1-5

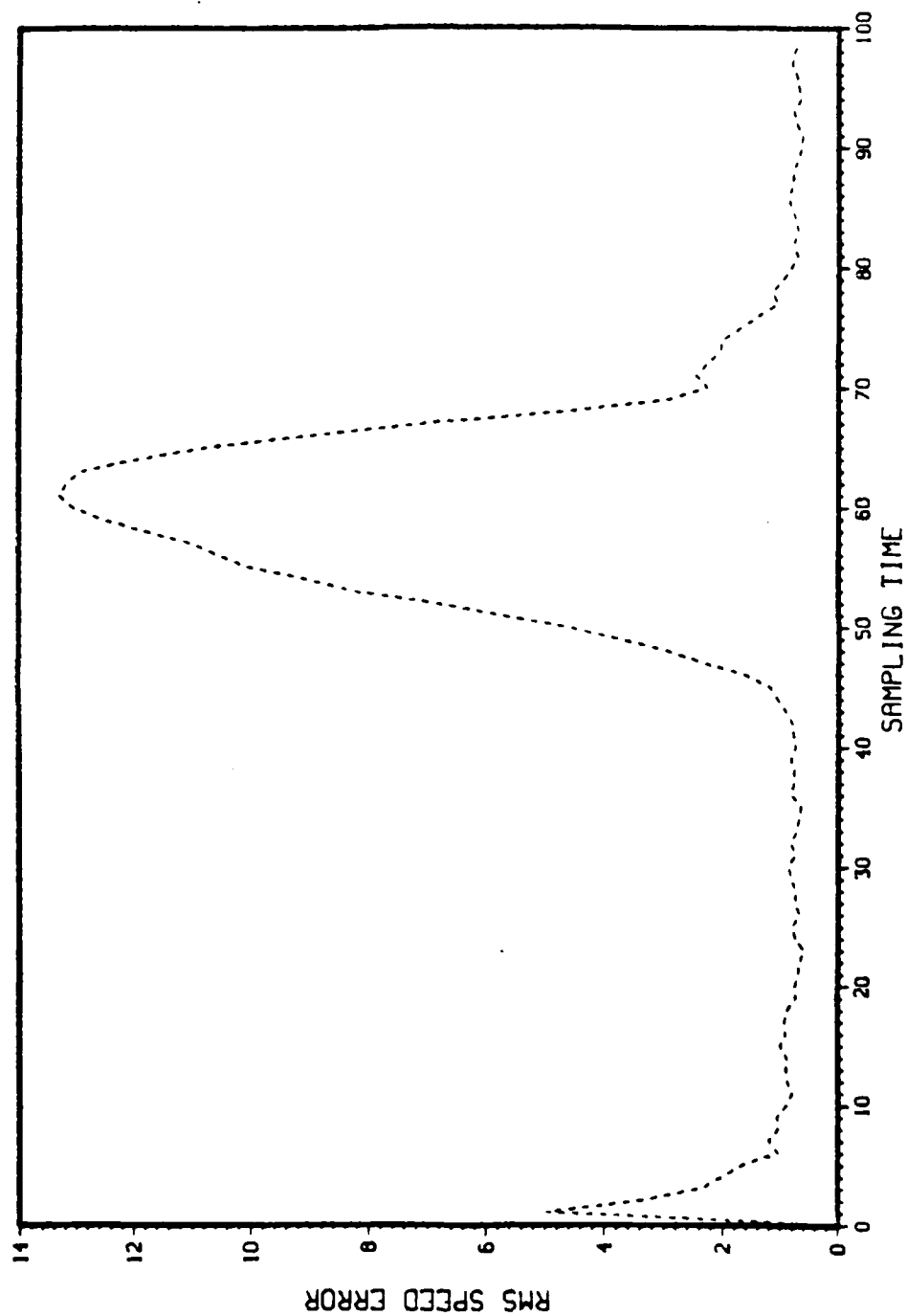


Figure 6.77 Mean Square Velocity Error; Many Q' Levels;
Q' Levels: 0,1,10

MEAN SQUARE VELOCITY ERROR

MANY 0 LEVELS

0 LEVELS: 0-3-10

THRES. LEVELS: 1-5

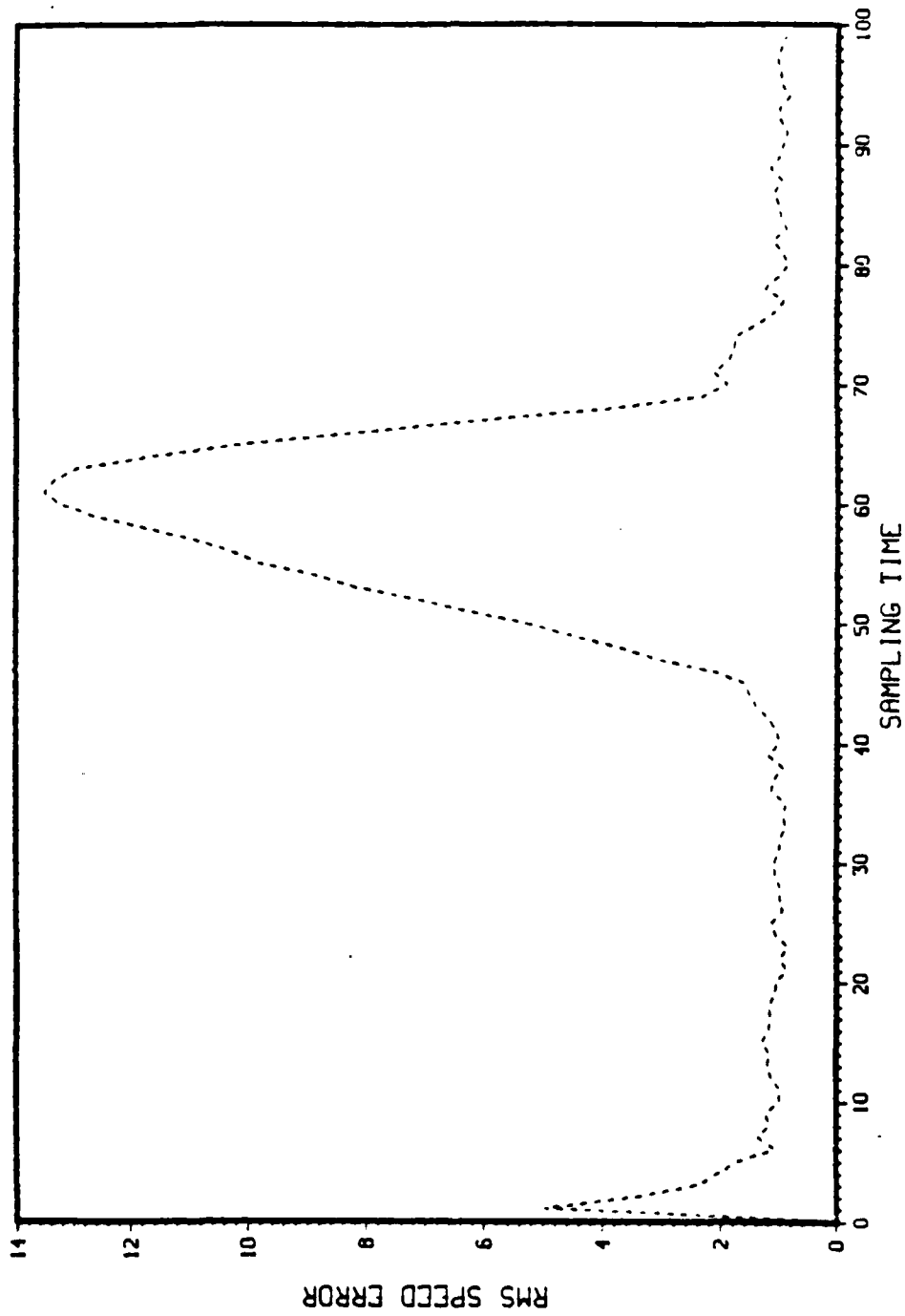


Figure 6.78 Mean Square Velocity Error; Many Q' Levels;
Q' Levels: 0,3,10

MEAN SQUARE VELOCITY ERROR

MANY 0 LEVELS

0 LEVELS: 0-1-1000

THRES. LEVELS: 1-5

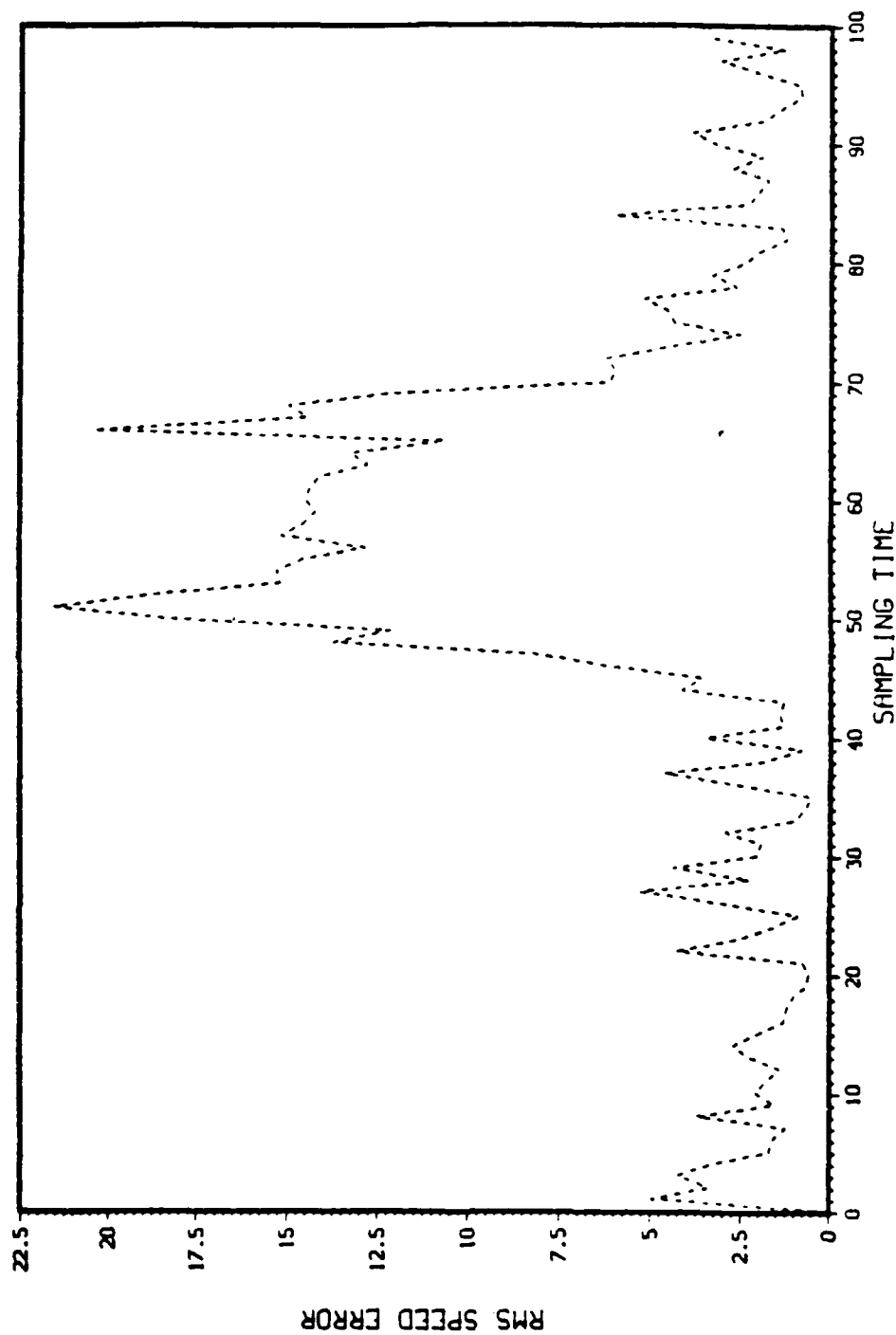


Figure 6.79 Mean Square Velocity Error; Many Q' Levels;
Q' Levels: 0,1,1000

MEAN SQUARE VELOCITY ERROR

MANY 0 LEVELS

0 LEVELS: 0-1-1000

THRES. LEVELS: 1-5

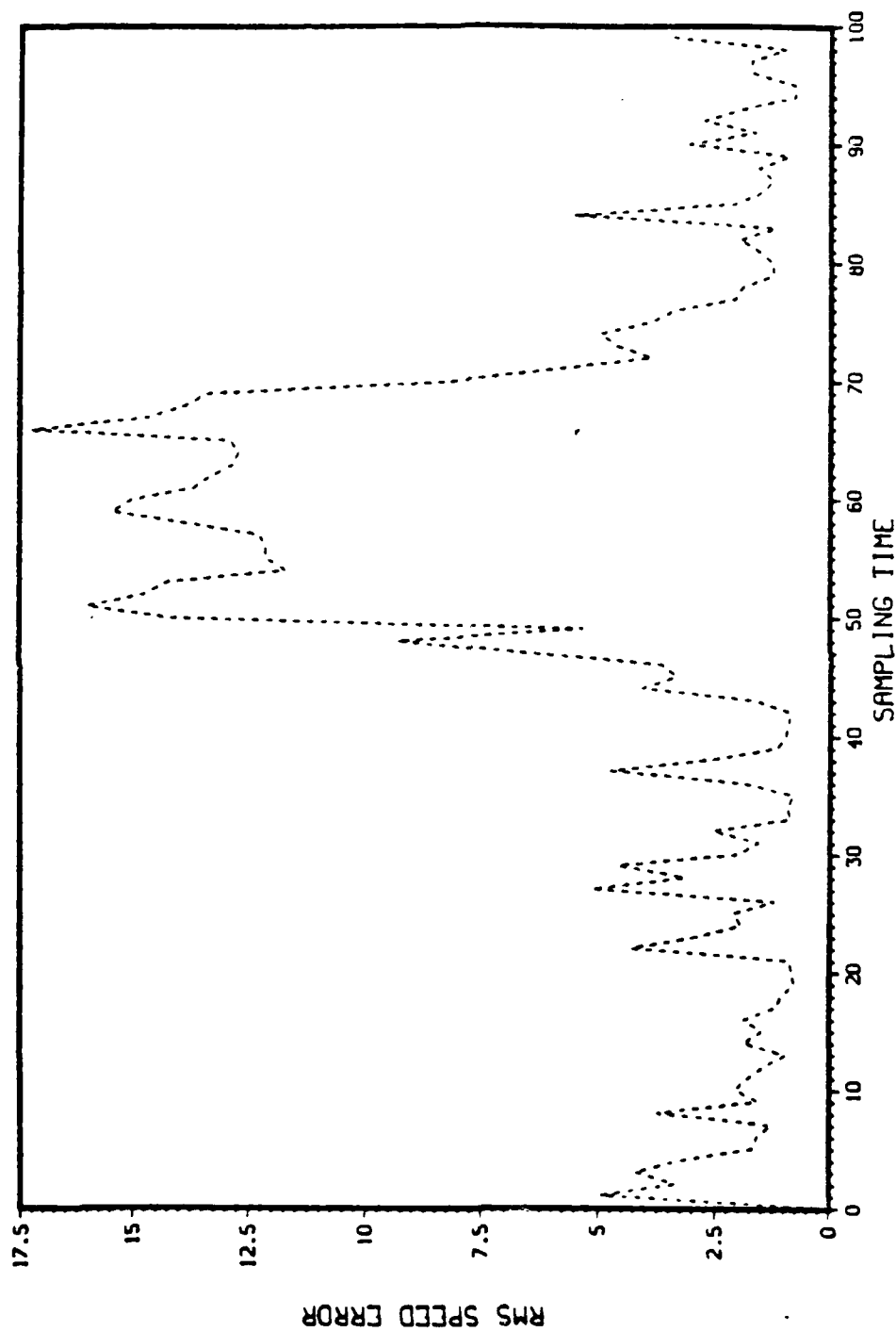


Figure 6.80 Mean Square Velocity Error; Many 0' Levels;
Q' Levels: 0,1,1000

MEAN SQUARE VELOCITY ERROR

MANY Q LEVELS

Q LEVELS: 0-3-1000

THRES. LEVELS: 1-5

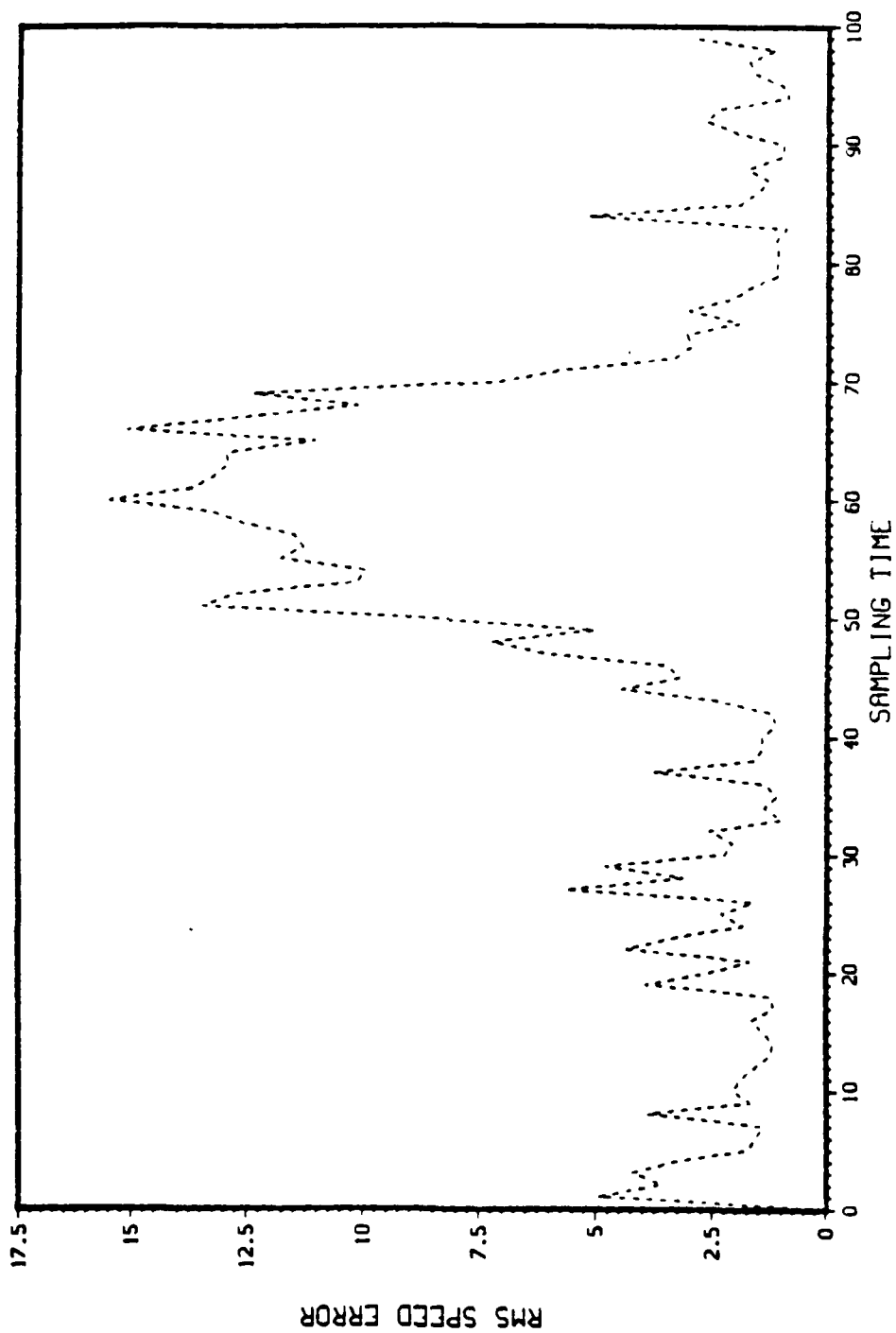


Figure 6.81 Mean Square Velocity Error; Many Q' Levels;
Q' Levels: 0,3,1000

MEAN SQUARE VELOCITY ERROR

MANY 0 LEVELS

Q LEVELS: 0-10-1000

THRES. LEVELS: 1-5

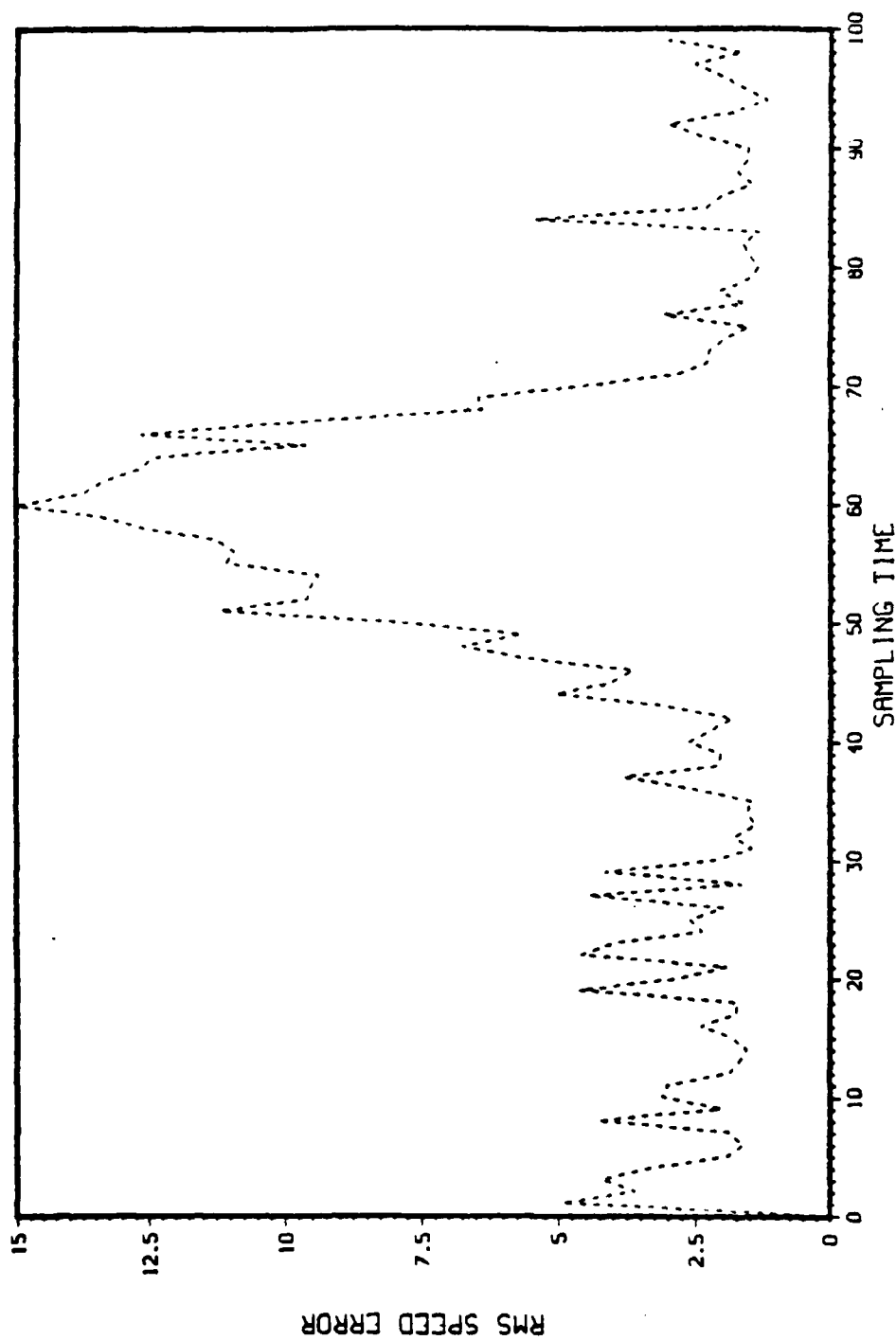


Figure 6.82 Mean Square Velocity Error; Many 0' Levels;
Q' Levels: 0,10,1000

3. Comparison With Other Approaches

Bar-Shalom [Ref. 9], has developed a different approach for the same problem. In this case the tracking filter operates in its normal mode in the absence of any maneuver. Once a maneuver is detected, a different state model is used by the filter. The acceleration is added as a new state component. The extent of the maneuver as detected, is then used to yield an estimate for the extra state component, and corrections are made on the other state components. The tracking is then done with the augmented state model until it will be converted to the normal model by another decision.

The results of this approach can be seen in Figure 6.83. The curve having the indication "VD" belongs to the above algorithm while the other curve (indication "IE") belongs to a different approach developed by Chan [Ref. 10]. The outline of the latter algorithm is the following:

When a maneuver is detected, the magnitude of the acceleration is identified in a least squares format. The result is used in conjunction with a standard Kalman Filter to estimate the state of the vehicle. The aim of the acceleration input estimation is to remove the filter bias caused by the target deviating from the assumed constant velocity, straight line motion.

It can be easily see (Figure 6.83) that the algorithm of Reference 9 is superior to that of Reference 10.

The "white noise model with adjustable level" method, modified with multiple levels of process noise Q (one for

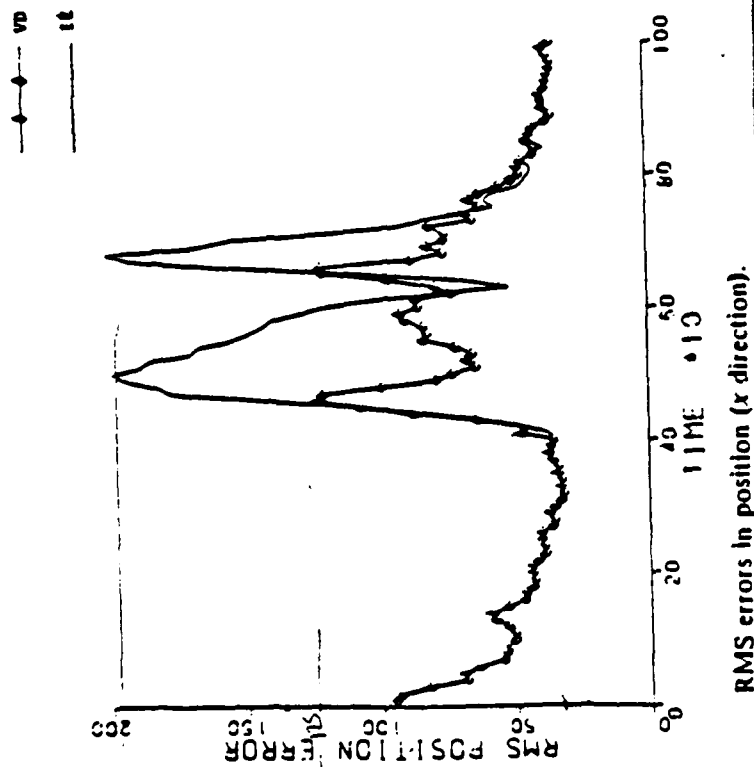
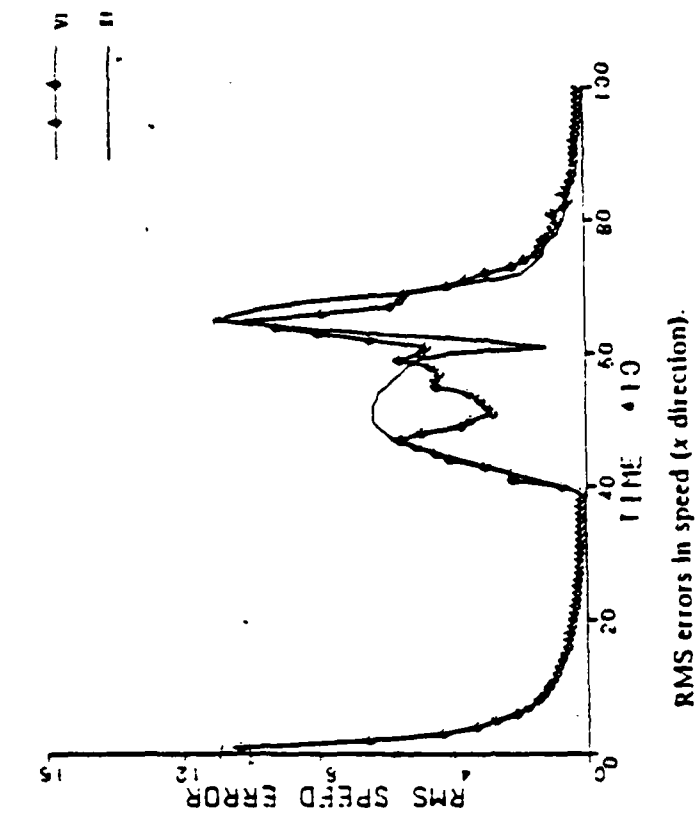


Figure 6.83 References 9 and 10 Results of Manoeuvring Target Example

each type of maneuver), can be compared with the algorithms . of References 9 and 10. Specifically this paper's algorithm is better than both the algorithms in the r.m.s. position error results and worse in the r.m.s. velocity error results. In the r.m.s. velocity error case, the approximate maximum value of the two other algorithms is 11, while this paper's algorithm maximum value is 13.3 (not too big a difference). In the r.m.s. position error case, the maximum values for References 9 and 10 algorithms are 200 and 125, respectively, while this paper's algorithm has achieved an approximate value of 95.

VII. CONCLUSIONS

In the examples presented, it was seen that the Kalman Filter has the ability and flexibility to treat various cases of the target tracking problem with very satisfactory results.

The Kalman Filter is always initialized by the user, providing the initial estimate $\hat{x}(k)$ and its corresponding estimate error covariance matrix $P(k)$. The initialization is of great importance for the filter's performance. A poor initialization needs more observations and time for the algorithm estimate to converge toward the value of the state vector.

In all cases the importance of the gain matrix $G(k)$ of the Kalman Filter was significant, i.e., since it is inversely proportional to time, it weights the correction term $[z(k) - \hat{x}(k|k-1)]$ less heavily as time progresses, and so the state estimation $\hat{x}(k|k)$ depends more on the state estimation of past time $\hat{x}(k|k-1)$.

It was also noted that the Q matrix accounts for any model inaccuracies. For a filter in a steady state condition, the Q also serves to prevent the gain matrix $G(k)$ from approaching zero by always insuring uncertainty in the predicted error covariance matrix. Q is also the key variable for treating the maneuvering target problem with the "White

noise model with adjustable level" method. As it was explained in the corresponding chapter, variations in Q cause proportional variations to the gain $G(k)$, which weights the correction factor in the equation:

$$\hat{\underline{x}}(k|k) = \hat{\underline{x}}(k|k-1) + G(k) [z(k) - c(k)\hat{\underline{x}}(k|k-1)]$$

Larger values of Q tend to cause the state estimate to favor the observation.

The error covariance matrix and specifically some of its elements (position and velocity error variance), were very useful as indicators of the filter's performance.

In the case of the maneuvering target, the "White noise model with adjustable level" method was very reliable and comparable to the algorithms of References 9 and 10. Additionally, it is much simpler than the other two methods. The only disadvantage was the greater value of the velocity r.m.s. error. In the example presented, the difference was not significant but it is estimated that it would increase in cases where the target maneuvers more drastically (greater accelerations).

The Monte Carlo simulation should be used to provide statistical results for any stochastic process which is represented by pseudorandom numbers. This simulation is a rigorous statistical procedure, to compare two or more algorithms.

More specific conclusions concerning the examples presented have already been provided in the corresponding analyses of them.

It is suggested that a continuation of this research should include the following:

a. Problem of maneuvering target which uses greater acceleration driving the maneuvers. The "White noise model with adjustable level" method should be applied and the results in position and velocity r.m.s. errors should be obtained.

b. Problem of multiple targets under various situations (maneuvering, in clutter, etc.).

c. Problem of single and multiple maneuvering targets in clutter. It is an interesting and important topic not included in this paper due to lack of time.

d. Study the implementation of a Kalman Filter algorithm using special purpose hardware.

APPENDIX A

COMPUTER ALGORITHMS

 LINEAR KALMAN FILTER EXAMPLE

```

XK1=X1(K,K-1)
XK2=X2(K,K-1)
X1ACT=X1(K,K)
X2ACT=X2(K,K)
PRR11=P11(K+1,K)
PRR12=P12(K+1,K)
PTT21=P21(K+1,K)
PTT22=P22(K+1,K)
PKK11=P11(K,K)
PKK12=P12(K,K)
PKK21=P21(K,K)
PKK22=P22(K,K)
  
```

DECLARATION OF VARIABLES

```

REAL PKK11(100),PKK12(100),PKK21(100),PKK22(100),XK1(100),XK2(100)
REAL X1EST(100),X2EST(100),G1(100),G2(100),PQER(100),VEER(100)
* PRR11(100),PTT21(100),PRR12(100),PTT22(100)
REAL X1ACT(100),X2ACT(100),W(100),L(100),M(101),Z(101),V(101)
*,NSTAER(100),NINER(100)
  
```

DEFINITION OF MATRIX P(1:1)

```

Q=1.
R=1.
T=1.
PKK11(1)=R
PKK12(1)=R/T
PKK21(1)=R/T
PKK22(1)=2*R/(T+1)
  
```

CALCULATION OF GAINS OF THE KALMAN FILTER

```

DO 10 K=2,100
  PRR11(K)=PKK11(K-1)+T*PKK21(K-1)+T*PKK12(K-1)+1+Z*PKK22(K-1)
  * +Q*0.25
  PRR12(K)=PKK12(K-1)+T*PKK22(K-1)+G-0.5*Q
  PTT21(K)=PKK21(K-1)+T*PKK22(K-1)+0.5*Q
  PTT22(K)=PKK22(K-1)+Q
  G1(K)=PRR11(K)/(PRR11(K)+K)
  G2(K)=PTT21(K)/(PKK11(K)+K)
  PKK11(K)=PRR11(K)-G1(K)*PKK11(K)
  
```

QA400010
 QA400020
 QA400030
 QA400040
 QA400050
 QA400060
 QA400070
 QA400080

QA400090
 QA400100
 QA400110
 QA400120
 QA400130

QA400140
 QA400150
 QA400160
 QA400170
 QA400180
 QA400190
 QA400200

QA400210
 QA400220
 QA400230
 QA400240
 QA400250
 QA400260
 QA400270
 QA400280
 QA400290

QA4000300
QA4000310
QA4000320
QA4000330
QA4000340
QA4000350

QA4000360
QA4000370

QA4000380
QA4000390
QA4000400
QA4000410
QA4000420
QA4000430
QA4000440

QA4000450
QA4000460
QA4000470

QA4000480
QA4000490
QA4000500

QA4000510
QA4000520
QA4000530
QA4000540

QA4000550
QA4000560
QA4000570
QA4000580
QA4000590

```

      PKK12(K)=PRR12(K)-G1(K)*PRK12(K)
      PKK21(K)=PTT21(K)-G2(K)*PKK11(K)
      PKK22(K)=PTT22(K)-G2(K)*PRK12(K)
      WRITE(5,100)PKK11(K),PRR11(K)
      WRITE(11,100)PKK22(K),PTT22(K)
10  CONTINUE
C
C  DEFINITION OF INITIAL STATE
C
      X1ACT(1)=0.
      X2ACT(1)=10.
C
C  GENERATION OF PROCESS NOISES "V"
C
      DSEED=5674312.
      NR=100
      CALL GGNML(DSEED,NR,L)
      DO 15 I=2,101
        K=I-1
        V(I-1)=SQRT(Q)*L(K)
        WRITE(11,200)X1ACT(K),X2ACT(K)
C
C  TRUE TRAJECTORY EQUATIONS
C
      X1ACT(I)=X1ACT(I-1)+T*X2ACT(I-1)+(T/2)*V(I-1)
      X2ACT(I)=X2ACT(I-1)+V(I-1)
15  CONTINUE
C
C  GENERATION OF MEASUREMENT NOISE "W"
C
      DSEED=231456.
      NR=101
      CALL GGNML(DSEED,NR,M)
C
C  GENERATION OF MEASUREMENTS
C
      DO 20 J=1,101
        W(J)=SQRT(R)*M(J)
        Z(J)=X1ACT(J)+W(J)
20  CONTINUE
C
C  ESTIMATED TRAJECTORIES EQUATIONS
C
      X1EST(1)=Z(2)
      X2EST(1)=(Z(2)-Z(1))/T
      DO 25 I=2,101
        XK1(I)=X1EST(I-1)+T*X2EST(I-1)
        XK2(I)=X2EST(I-1)
25  CONTINUE

```

QA400600
QA400610

QA400620
QA400630

QA400640
QA400650
QA400660
QA400670

QA400680
QA400690
QA400700
QA400710
QA400720
QA400730
QA400740
QA400750
QA400760
QA400770
QA400780

X1EST(I)=G1(I)*Z(I)+(1-G1(I))*XK1(I)
X2EST(I)=XK2(I)+G2(I)*Z(I)-G2(I)*XK1(I)

NORMALIZED POSITION AND VELOCITY ERRORS

POER(I)=(X1ACT(I)-X1EST(I))/(PKK11(I)**.5)
VEER(I)=(X2ACT(I)-X2EST(I))/(PKK22(I)**.5)

NORMALIZED STATE ERROR SQUARED

NSTAER(I)=(1/(PKK11(I)*PKK22(I)-PKK21(I)*PKK12(I))
(PKK22(I)(X1ACT(I)-X1EST(I))**2
+PKK11(I)*(X2ACT(I)-X2EST(I))**2-2*PKK21(I)
(X2ACT(I)-X2EST(I))(X1ACT(I)-X1EST(I)))

NORMALIZED INNOVATION ERROR

NINER(I)=(Z(I)-XK1(I))/(PRR11(I)*R)**.5)
WRITE(4,100)POER(I)
WRITE(12,100)VEER(I)
WRITE(2,200)X1EST(I),X2EST(I)
WRITE(8,100)NSTAER(I)
WRITE(10,100)NINER(I)

25 CONTINUE
100 FORMAT(F15.8)
200 FORMAT(2(F15.8))
STOP
END


```

15 TH1TA(10)=1.570796327
25 IF(I.GT.10)TH1TA(I)=3.141592654+TH1TA(I)
Z(I)=TH1TA(I)+V(I)*2*3.1415/180
DEQ(I)=((XKE1(I)-A(I))*2+XKE2(I))*2)
H1(I)=-XKE2(I)/DEQ(I)
H2(I)=(XKE1(I)-A(I))/DEQ(I)
PK11(I)=PK11(I)
PK12(I)=PK12(I)
PK21(I)=PK21(I)
PK22(I)=PK22(I)
35 CONTINUE
*PK22(I)=H1(I)*2*PK11(I)+H1(I)*H2(I)*(PK21(I)+PK12(I))+H2(I)**2*
DEN(I)=(PK11(I)*H1(I)+PK12(I)*H2(I))/DEN(I)
G1(I)=(PK21(I)*H1(I)+PK22(I)*H2(I))/DEN(I)
G2(I)=(PK11(I)*PK22(I)-PK21(I)*PK12(I))
NSIS(I)=1/(PK11(I)*((X2-X2ESTK(I))*2)
NSTESQ(I)=(PK11(I)*((X1-X1ESTK(I))*2)
+ PK22(I)*((X1-X1ESTK(I))*2)
- PK21(I)*((X1-X1ESTK(I))*2)
- PK12(I)*((X1-X1ESTK(I))*2)
PK11(K)=PK11(I)*((1-G1(I))*H1(I))*H1(I)*G1(I)*H2(I)
PK12(K)=(1-G1(I))*H2(I)*PK21(I)-G1(I)*H2(I)*PK22(I)
PK21(K)=(1-G2(I))*H2(I)*PK21(I)-G2(I)*H1(I)*PK11(I)
PK22(K)=(1-G2(I))*H2(I)*PK22(I)-G2(I)*H1(I)*PK12(I)
IF(XKE1(I).EQ.A(I))GO TO 45
ESTTHI(I)=(ATAN(XKE2(I)/(XKE1(I)-A(I))))
45 GU TO 55
55 ESTTHI(I)=1.570796327
IF(ESTTHI(I).LT.0)ESTTHI(I)=3.141592654+ESTTHI(I)
X1ESTK(K)=XKE1(I)+G1(I)*G1(I)*ESTTHI(I)
X2ESTK(K)=XKE2(I)+G2(I)*G2(I)*ESTTHI(I)
XKE1(K)=X1ESTK(K)
XKE2(K)=X2ESTK(K)
WRITE(6,50)I,X1ESTK(I),X2ESTK(I),TH1TA(I),NSTESQ(I)
C WRITE(6,100)PK11(I),PK12(I),PK21(I),PK22(I)
C WRITE(6,*)NSTESQ(I)
65 CONTINUE
50 FORMAT(5X,I5,F10.5,3(5X,F10.5))
100 FORMAT(5X,F10.5,3(5X,F10.5))
ST CP
END

```

```

CCCCCCCCCCCCCCCC
*****
MANEUVERING TARGET EXAMPLE
*****
PK1K11=F11(K+1|K)
PK1K12=F12(K+1|K)
PK1K21=F21(K+1|K)
PK1K22=F22(K+1|K)
PKK11 = F11(K|K)
PKK12 = F12(K|K)
PKK21 = F21(K|K)
PKK22 = F22(K|K)
DECLARATION OF VARIABLES
REAL*8 CSEED
REAL PKK11(100), PKK12(100), PKK21(100), PKK22(100), XK1(100), XK2(100)
REAL X1EST(100), X2EST(100), G1(100), G2(100), E(100), X1ACT1(100)
* PK1K11(100), PK1K21(100), PK1K12(100), PK1K22(100), PQ(100)
REAL X1ACT1(100), X2ACT1(100), W(100), L1(100), M(100), Z(100), V(100)
* PR(100), S7(100), X1ACT1(100), X2ACT1(100), A(100), X1ACT3(100)
* D(100), DBAR(100), XK1BAR(100), ZBAK(100), AVEG(100), AVSUMZ(100)
* AVEZ(100), X2SQ(100)
INTEGER F, I, J, IA, JA, L
CALL TEK618
CALL CCMPRS
INITIALIZATION OF VAR. FOR MONTE CARLO SIMULATION
DO 7 I=2,100
  LEAR(I)=0.
  AVEZ(I)=0.
  AVEX(I)=0.
  XK1EAR(I)=0.
  AVEZ(I)=0.
  ZBAR(I)=0.
  SUM(I)=0.
  AVSUM(I)=0.
  SUM2(I)=0.
  AVSUM2(I)=0.
7 CONTINUE
DEFINITION OF MATRIX P(111)
Q=0.0
R=10.**4
T=10.
SSSS00C10
SSSS00020
SSSS00030
SSSS00J40
SSSS00050
SSSS00C60
SSSS00070
SSSS00080
SSSS00C90
SSSS00100
SSSS00110
SSSS00120
SSSS00130
SSSS00140
SSSS00150
SSSS00160
SSSS00170
SSSS00180
SSSS00180
SSSS00200
SSSS00210
SSSS00220
SSSS00C230
SSSS00240
SSSS00250

```

```

CSEEC=5674312.D0
PKK11(1)=R
PKK12(1)=R/T
PKK21(1)=R/T
PKK22(1)=2*R/(1+T)
D(1)=C.

START CF MONTE CARLO LGOP
DO 666 F=2,51
START OF KALMAN FILTER LOOP
DO 599 L=2,100
D(L)=FLOAT(L)-1.
R=10**4
T=10.

CALCULATION OF GAINS OF THE KALMAN FILTER
*
PKK11(L)=PKK11(L-1)+T*PKK21(L-1)+T*PKK12(L-1)+T**2*PKK22(L-1)
+Q*(T/2)**2
PKK12(L)=PKK12(L-1)+T*PKK22(L-1)+(T/2)*Q
PKK21(L)=PKK21(L-1)+T*PKK22(L-1)+(T/2)*Q
PKK22(L)=PKK22(L-1)+Q
G1(L)=PKK11(L)/(PKK11(L)+R)
G2(L)=PKK12(L)/(PKK11(L)+R)
PKK11(L)=PKK11(L)-G1(L)*PKK11(L)
PKK12(L)=PKK12(L)-G1(L)*PKK12(L)
PKK21(L)=PKK21(L)-G2(L)*PKK11(L)
PKK22(L)=PKK22(L)-G2(L)*PKK12(L)
WRITE(3,100)PKK11(L),PKK11(L)
WRITE(11,100)PKK22(L),PKK12(L)

DEFINITION OF INITIAL STATE
X1ACT(1)=2000.
X2ACT(1)=0.0

GENERATION OF PROCESS NOISES "u"
USEED=5674312.
NR=100
CALL GGWML(DSEED,NR,A)
K=L-1
W(L-1)=0.0
WRITE(11,200)X1ACT(L),X2ACT(L)

```

```

C C
TRUE TRAJECTORY EQUATIONS
GAM1=.C15
GAM2=.C2
PO(1)=10.
PO(L)=PC(L-1)+10. GO TO 33
IF (PO(L)-LT.410) GO TO 33
IF (PO(L)-LT.610) AND. PO(L).GT.400) GO TO 43
IF (PO(L)-EQ.610) GO TO 53
IF (PO(L)-LT.670) AND. PO(L).GT.610) GO TO 63
IF (PO(L)-GT.660) AND. PO(L).LE.1000) GO TO 73
X2ACT(L)=X2ACT(L-1)+W(L-1)
X1ACT(L)=X1ACT(L-1)+10.*X2ACT(L-1)+5.*W(L-1)
GO TO 83
X1ACT1(L)=2000.+5.*GAM1*(PO(L)-400)**2+5.*W(L-1)
GO TO 83
X1ACT6=3650
GO TO 83
X1ACT2(L)=3650+15*(PO(L)-610)-.5.*GAM2*(PO(L)-610)**2+5.*W(L-1)
GO TO 83
X1ACT3(L)=4025.+5.*W(L-1)
CONTINUE
WRITE(5,200)X1ACT(L),X1ACT(L-1),X2ACT(L-1),W(L-1)
WRITE(5,100)X1ACT3(L)

C C C C C C C
GENERATION OF MEASUREMENT NOISE "V"
DSEED=231456.
NR=101
CALL GGNML(DSEED,NR,M)

C C C
GENERATION OF MEASUREMENTS
IF (L.LT.41) GO TO 241
IF (L.LT.61) GO TO 242
IF (L.EQ.61) GO TO 243
IF (L.GT.61) AND. L.LE.66) GO TO 246
IF (L.GT.66) GO TO 245
PR(L)=X1ACT(L)
GO TO 244
PR(L)=X1ACT1(L)
GO TO 244
PR(L)=X1ACT6
GO TO 244
PR(L)=X1ACT2(L)
GO TO 244
PR(L)=X1ACT3(L)
V(L)=100.*M(L)

```

```

SS00660
SS00670
SS00680
SS00690
SS00700
SS00710
SS00720
SS00730
SS00740
SS00750
SS00760
SS00770
SS00780
SS00790
SS00800
SS00810
SS00820
SS00830
SS00840
SS00850
SS00860
SS00870
SS00880
SS00890
SS00900
SS00910
SS00920
SS00930
SS00940
SS00950
SS00960
SS00970
SS00980
SS00990
SS01000
SS01010
SS01020
SS01030
SS01040
SS01050
SS01060
SS01070
SS01080
SS01090
SS01100
SS01110
SS01120
SS01130

```

SSSO1140
SSSO1150
SSSO1160
SSSO1170
SSSO1180

SSSO1190
SSSO1200
SSSO1210
SSSO1220
SSSO1230
SSSO1240

SSSO1250
SSSO1260
SSSO1270
SSSO1280
SSSO1290

SSSO1300
SSSO1310
SSSO1320
SSSO1330
SSSO1340
SSSO1350

SSSO1360
SSSO1370
SSSO1380

```

C      Z(L)=PR(L)+V(L)
C      WRITE (5,100)Z(L)
C
C      ESTIMATED TRAJECTORIES EQUATIONS
C
C      Z(1)=2000
C      XK1(1)=2000
C      X1EST(1)=Z(2)
C      X2EST(1)=Z(2)-Z(1))/T
C      XK1(L)=X1EST(L-1)+10.*X2EST(L-1)
C      XK2(L)=X2EST(L-1)
C      X1EST(L)=G1(L)*Z(L)+(1-G1(L))*XK1(L)
C      X2EST(L)=XK2(L)+G2(L)*Z(L)-G2(L)*XK1(L)
C      PRXSQ(L)=(PR(L)-X1EST(L))**2
C      X2SQ(L)=(X2ACT(L)-X2EST(L))**2
C
C      SQUARE ACRM OF THE INNOVATIONS
C
C      S7(L)=PKK1(L-1)+2*T*PKK21(L-1)+T**2*PKK22(L-1)+Q*(T/2)**2+K
C      E(L)=(Z(L)-XK1(L))**2/S7(L)
C
C      PROVISION FOR THE MANEUVERS (DIFFERENT Q LEVELS)
C
C      IF (E(L).GT.0.1) GO TO 11
C      IF (E(L).LE.0.1) GO TO 21
C      Q=100.
C      GO TO 31
C      Q=0.
C      CONTINUE
C      IF (E(L).LE.1.) GO TO 11
C      IF (E(L).GT.1.) AND (E(L).LE.3.) GO TO 21
C      IF (E(L).GT.3.) AND (E(L).LE.15.) GO TO 31
C      IF (E(L).GT.15.) GO TO 41
C      Q=0.
C      GO TO 51
C      Q=3.
C      GO TO 51
C      Q=100.
C      GO TO 51
C      Q=0.
C      CONTINUE
C      WRITE(5,100)E(L)
C      WRITE(5,200)PRXSQ(L),PR(L),X1EST(L),XK1(L),Z(L),X2EST(L)
C      999 CONTINUE
C
C      ENC OF KALMAN FILTER LOOP
C
C      DO 27 IA=2,100

```

```

DBAR(IA)=DBAR(IA)+D(IA)
SUM(IA)=SUM(IA)+PRXSQ(IA)
XK1BAR(IA)=XK1BAR(IA)+XK1(IA)
ZBAR(IA)=ZBAR(IA)+Z(IA)
SUM2(IA)=SUM2(IA)+X2SQ(IA)
27 CONTINUE
666 CONTINUE

```

```

C
C
C   END OF MGNT CARLO LOOP

```

```

DO 37 IA=2,100
  AVED(IA)=DBAR(IA)/50.
  AVSUM(IA)=SQRT(SUM(IA)/50.)
  AVEZ(IA)=XK1BAR(IA)/50.
  AVEZ(IA)=ZBAR(IA)/50.
  AVSUM2(IA)=SQRT(SUM2(IA)/50.)
  WRITE (5,200) AVED(IA), AVSUM(IA)

```

```

C   37 CONTINUE
100 FORMAT(F10.3)
200 FORMAT(6(F9.2))
C   DEFINE THE PLOTTING SCALE
SCALE=2.
C   PLOT SIGNAL, NOISE, VS. TIME
CALL FVST(AVED, AVSUM2, SCALE)
STOP
END

```

```

C   PLOTTING COMMANDS

```

```

C   SL6ROUTINE FVST PLOTS A FUNCTION .VS. TIME

```

```

SUBROUTINE FVST(TIME, VARI, SCALE)
REAL VARI(100), TIME(100)
CALL PAGE(11., 8.5)
CALL NCRDR
CALL BLCWUP(SCALE)
CALL AREA2D(9.0, 6.0)
CALL FRAME
CALL XNAME('SAMPLING TIME', 13)
CALL YNAME('RMS SPEED ERROR', 100)
CALL HEADIN('MEAN SQUARE VELOCITY ERROR', 100, 1., 4)
CALL HEADIN('MANY Q LEVELS', 100, 1., 4)
CALL HEADIN('Q LEVELS: 0-3-0', 100, 8, 4)
CALL HEADIN('THRES. LEVELS: 1-3', 100, 8, 4)
CALL HEADIN('CONTIN. LINE: MEASUREMENTS', 100, 8, 4)
CALL HEADIN('DASHED LINE: FILTER', 100, 8, 4)
CALL INTXS

```

```

SSSO1400
SSSO1410
SSSO1420
SSSO1430
SSSO1440
SSSO1450
SSSO1460
SSSO1470

```

```

SSSO1480
SSSO1490
SSSO1500
SSSO1510
SSSO1520
SSSO1530
SSSO1540
SSSO1550
SSSO1560
SSSO1570
SSSO1580
SSSO1590
SSSO1600

```

```

SSSO1610

```


LIST OF REFERENCES

1. Wiener, N., The Extrapolation Interpolation and Smoothing of Stationary Time Series with Engineering Applications, John Wiley and Sons, Inc., New York, 1949.
2. Kalman, R.E., "A New Approach to Linear Filtering on Prediction Problems," ASME Trans., J. Basic Eng., Vol. 83, March 1960, pp. 34-35.
3. Singer, R.A. and Stein, J.J., "An Optimal Tracking Filter for Processing Sensor Data of Imprecisely Determined Origin in Surveillance Systems," IEEE Proc. Conf. Decision and Control, December 1971, pp. 171-175.
4. Sea, R.G., "An Efficient Suboptimal Decision Procedure for Associating Sensor Data with Stored Tracks in Real-Time Surveillance Systems," Proc. IEEE Conf. on Decision and Control, December 1971, pp. 33-37.
5. Bar-Shalom, Y., Notes from NPS Course: EE 4416 (Multi-target Tracking), Spring Quarter, 1984.
6. Medith, Stochastic Optimal Linear Estimation and Control, McGraw-Hill Book Company, 1969.
7. Gelb, A., Ed., "Applied Optimal Estimation," The Analytic Sciences Corporation, 1974
8. Kirk, Donald E, Optimal Estimation: An Introduction to the Theory and Applications, Mimeographed Notes for Course EE4413, Naval Postgraduate School, Monterey, CA. Spring 1984.
9. Bar-Shalom, Y, and Bitminal, K., "Variable Dimension Filter for Maneuvering Target Tracking," IEEE T-AES., September 1982.
10. Chan, Y.T., Hu, A.G.C., and Plant, J.B., "A Kalman Filter Based Tracking Scheme with Input Estimation."

BIBLIOGRAPHY

Kirk, Donald E., Optimal Control Theory: An Introduction.
Prentice-Hall, 1970.

Maybeck, Peter S., Stochastic Models, Estimation, and
Control, Volume 1, New York: Academic Press, 1982.

Maybeck, Peter S., Stochastic Models, Estimation, and
Control, Volume 2, New York: Academic Press, 1982.

Mendel, Jerry M., Discrete Techniques of Parameter
Estimation, M. Dekker, 1973.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Library, Code 0142 Naval Postgraduate School Monterey, California 93943	2
2. Department Chairman, Code 62 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93943	1
3. Professor H. Loomis, Code 62Lm Department of Electrical Engineering Naval Postgraduate School Monterey, California 93943	4
4. Professor H. Titus, Code 62Ts Department of Electrical Engineering Naval Postgraduate School Monterey, California 93943	1
5. LT. V. Tsafaras, H.N. SMC #2355 Naval Postgraduate School Monterey, California 93943	1
6. LT. E. Chorianopoulos, H.N. SMC #1863 Naval Postgraduate School Monterey, California 93943	1
7. LT. S. Vitalis H.N. SMC #1334 Naval Postgraduate School Monterey, California 93943	1
8. LT. D. Kourkoulis H.N. SMC #1339 Naval Postgraduate School Monterey, California 93943	1
9. LT P. Andresakis H.N. SMC #2717 Naval Postgraduate School Monterey, California 93943	1

AD-A150 770

DIGITAL COMPUTER APPLICATIONS OF KALMAN FILTER IN
TARGET TRACKING(U) NAVAL POSTGRADUATE SCHOOL MONTEREY
CA V I MARTZOUCOS JUN 84

3/3

UNCLASSIFIED

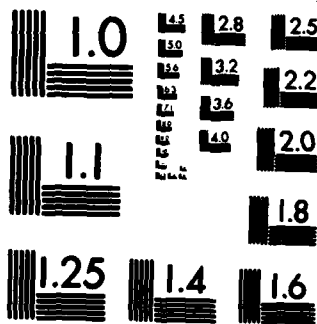
F/G 9/2

NL

END

FILED

DEC



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

- | | | |
|-----|--|---|
| 10. | LT. C. Miranda BA.
CTA/IAE/ESB
12200 Sao Jose Dos Campus
Sao Paulo, BRAZIL | 1 |
| 11. | Hellenic Navy General Staff
Education Department
Stratopedon Papagou
Holargos
Athens, GREECE | 3 |
| 12. | LT. V. Martzoucos, H.N.
Leoforos Galatsiou 115
Galatsi
Athens, GREECE | 5 |
| 13. | Defense Technical Information Center
Cameron Station
Alexandria, Virginia 22314 | 1 |

END

FILMED

4-85

DTIC