THESIS

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

PASSIVE ACOUSTIC TARGET MOTION ANALYSIS

by

George Edward Olcovich

June 1986

Thesis Advisor: H.A. Titus

JRITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

REPORT SECURITY CLASSIFICATION CLASSIFIED	1b. RESTRICTIVE MARKINGS
SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited
DECLASSIFICATION / DOWNGRADING SCHEDULE	
ERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)

NAME OF PERFORMING ORGANIZATION val Postgraduate School	6b OFFICE SYMBOL (If applicable) 62	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
ADDRESS (City, State, and ZIP Code) nterey, California 93943-5Q00		7b. ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
NAME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
ADDRESS (City, State, and ZIP Code)		10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO.

ITLE (Include Security Classification)
SSIVE ACOUSTIC TARGET MOTION ANALYSIS

ERSONAL AUTHOR(S)
orge Edward Olcovich

TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
ster's Thesis	FROM ______ TO ______	1986 June 20	210

UPPLEMENTARY NOTATION

COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
IELD	GROUP	SUB-GROUP	Extended Kalman Filter; Passive Acoustic Target Tracking; Nonlinear Filtering; Tracking from Airbourne Platform

BSTRACT (Continue on reverse if necessary and identify by block number)

:he technique of Extended Kalman filtering is applied to a passive acous-
: target motion analysis problem. A multisensor tracking algorithm is
reloped to provide a solution to the manuevering target
oblem based on noisy passive bearing and doppler shifted frequency
asurements. An adaptive control method is used to allow for
neuvering. The performance of the filter is also evaluated using
nputer generated doppler frequency and bearing data. The
formance of the filter is found to be acceptable under the tested
nditions.

STRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED

AME OF RESPONSIBLE INDIVIDUAL f H. A. Titus	22b TELEPHONE (Include Area Code) (408)646-2156	22c OFFICE SYMBOL 62Ts

DRM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete — SECURITY CLASSIFICATION OF THIS PAGE

Approved for public release; distribution is unlimited.

Passive Acoustic Target Motion Analysis

by

George Edward Olcovich
Lieutenant Commander, United States Navy
B.S., University of Redlands, 1974

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June, 1986

ABSTRACT

The technique of Extended Kalman filtering is applied to a passive acoustic target motion analysis problem. A multisensor tracking algorithm is developed to provide a solution to the manuevering target problem based on noisy passive bearing and doppler shifted frequency measurements. An adaptive control method is used to allow for maneuvering. The performance of the filter is also evaluated using computer generated doppler frequency and bearing data. The performance of the filter is found to be acceptable under the tested conditions.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

The author is indebted to the United States Navy for providing the opportunity to pursue this research. Many thanks to Professor Hal Titus for his counsel and professional guidance.

I. INTRODUCTION

The passive target motion analysis problem is defined as the estimation of the velocity and position of a target based on passive noisy measurements acquired from sonobuoys. Sonobuoys are capable of receiving a noise corrupted acoustic signature emitted by a target. This signature is transmitted to an Anti-Submarine Warfare (ASW) aircraft where various combinations of amplitude, frequency, bearing and time delay information can be obtained. Two of the primary information sources used for passive target tracking are relative Doppler from the target's radiated frequencies and bearing information from directional sonobuoys. If the sonobuoy is a Directional Finding Acoustic Receiver (DIFAR) buoy, a bearing can be obtained from the buoy to the target. If the sonobuoy is either a DIFAR or a Low Frequency Acoustic Receiver (LOFAR) buoy, the frequency spectrum of the acoustic signature can be determined. These two types of noisy measurements are nonlinear functions of the target's position, course and speed.

The purpose of this research was to develop an operational algorithm for tracking submarines from an airbourne platform by observing these noisy measurements. This algorithm assumes a close tracking environment (i.e., direct path). In order to operate, the algorithm requires an initial estimated position, course, speed, frequency and an area of probability for the target of interest. In a close tracking environment this information would be available through human operators, received from other tracking routines, or received from other external sensors.

Since the noisy measurement equations are nonlinear, nonlinear filter theory must be applied to the tracking algorithm. The Extended Kalman filter theory represents, in principle, an ideal solution to this type of problem. The filter provides

1. For the use of any number, combination and sequence of external measurements,

2. Its own error analysis,

3. A structure which is recursive,

4. The ability to reconstruct the entire state vector from a noisy measurement .

5. A form of linearization

The discrete time version of the Extended Kalman filter is developed in this research and was chosen because of the prevalence of digital computers.

Before the filter theory of Kalman can be applied to the problem, the system and measurement models must be developed. From the geometry of the target-sonobuoy problem, the target's equations of motion and noise-free measurement equations are defined in Chapter II.

In Chapter III, the theoretical background and assumptions used in deriving the discrete Extended Kalman Filter are presented. The discrete Extended Kalman Filter is then applied to the passive acoustic tracking problem. In the final subsection of Chapter III, the problem of maneuvering and divergence control is presented. System modeling errors occur when the target undergoes a maneuver. These modeling errors may cause the filter to diverge. In order to prevent divergence and to accommodate maneuver, an adaptive control method is developed.

In Chapter IV, the concept of error ellipsoids is developed. Error ellipsoids provide significant information about the estimated position. Also, the error ellipsoids are useful in visualizing the estimation error.

Chapter V discusses the development and implementation of the multiple sonobuoy tracking algorithm. The algorithm is divided into three modules.

1. The track module generates noise-free track data.

2. The observation module receives the noise-free track data generated by the track module and simulates noisy measurement data for input to the tracking algorithm module.

3. The tracking algorithm module receives noisy measurement data and generates estimates and predictions.

In Chapter VI, five scenarios are presented to demonstrate the performance of the algorithm. The final Chapter summarizes the results of this research and presents the conclusions and recommendations for further study.

II. PROBLEM STATEMENT

A. SYSTEM MODEL

Consider the target(submarine)-observer(sonobuoy) encounter in the two dimensional plane as shown in Figure (2-1).[Ref. 1:p. 3] The x and y components of the target velocity, v_t are denoted v_{xt} and v_{yt}, while the x and y components of the target and observer positions are denoted x_t, y_t, x_b, and y_b respectively. The course of the target is θ_t and the bearing from the observer to the target with respect to North is θ. The distance from the observer to the target is

$$r = \sqrt{(x_t - x_b)^2 + (y_t - y_b)^2} \quad . \tag{2-1}$$

Since the doppler shifted frequency is used as an observable, it is also necessary to estimate the rest frequency of the emitter on the target. In this North-East oriented Cartesian coordinate system, a fifth order state variable is chosen.

$$\underline{x} = \begin{bmatrix} x_t \\ v_{xt} \\ y_t \\ v_{yt} \\ f_o \end{bmatrix} \tag{2-2}$$

Note that other reference frames or state vectors can be used to solve this problem. This model was chosen because it is simple enough to work with mathematically, yet sufficiently detailed to describe the target's motion. The notation used throughout the thesis is that vector quantities are lower case letters and underlined and matrices are given upper case letters.

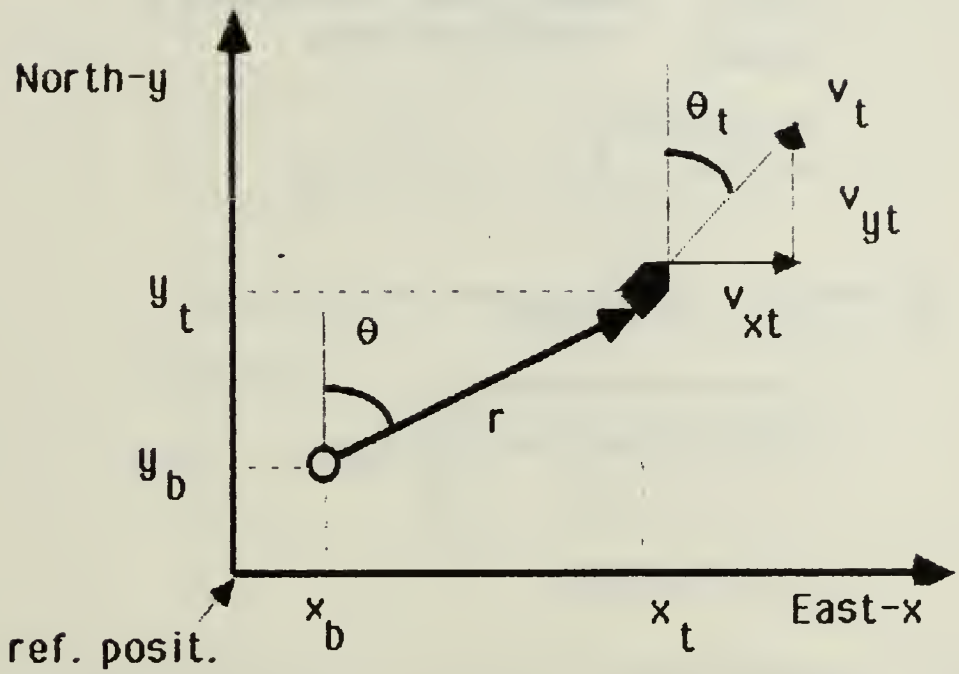

Figure 2-1 Target- Observer Encounter

1. Discrete Time Estimation

The advances in digital computer technology and development of Kalman filtering techniques make it possible to obtain a straight forward recursive solution to a real-time estimation problem. Three types of estimation problems are shown in Figure (2-2). The discrete time system involves estimating the state variables of the system at time t_k based upon a sequence of observations taken up to an including time t_j. The problem is termed filtering when $t_k = t_j$; smoothing when $t_k < t_j$; and prediction when $t_k > t_j$. [Ref. 2:p. 3] Since the primary purpose of this work is to develop an algorithm for tracking submarines, only the filtering and prediction problems are considered. Let the time interval between discrete time t_k and t_{k+1} be defined as

$$T = t_{k+1} - t_k \quad . \tag{2-3}$$

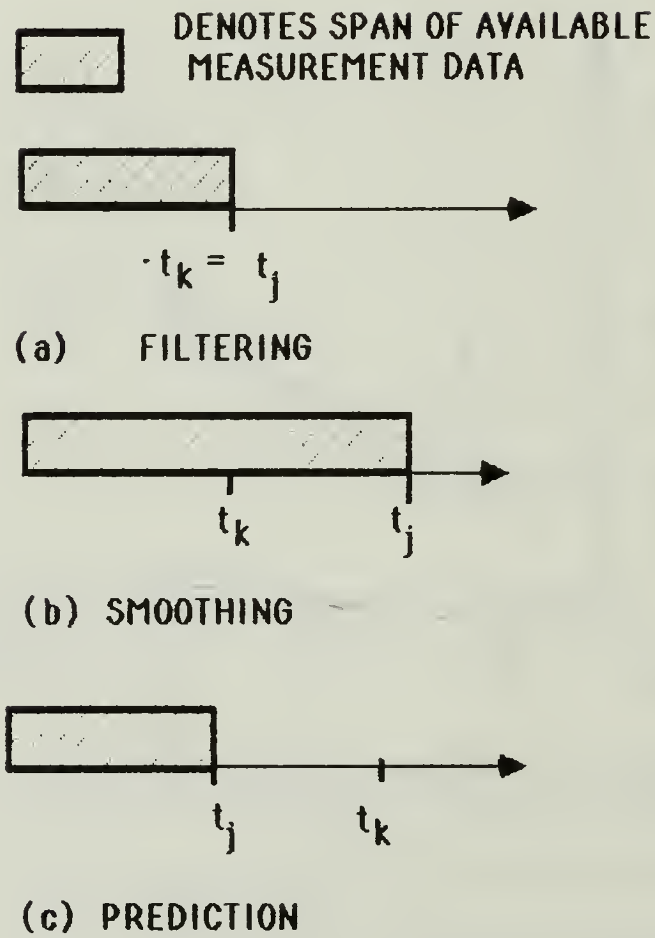

Figure 2-2 Three Types of Estimation Problems
(estimate desired at time t_k)

2. Target Maneuvers

It is assumed that all maneuvers are imparted by white random forcing functions. As depicted in Figure (2-3), let the random variables δ_{vt} and $\delta_{\theta t}$ represent the following:

δ_{vt} = acceleration along the target's course

$\delta_{\dot{\theta}t}$ = angular velocity or turn rate

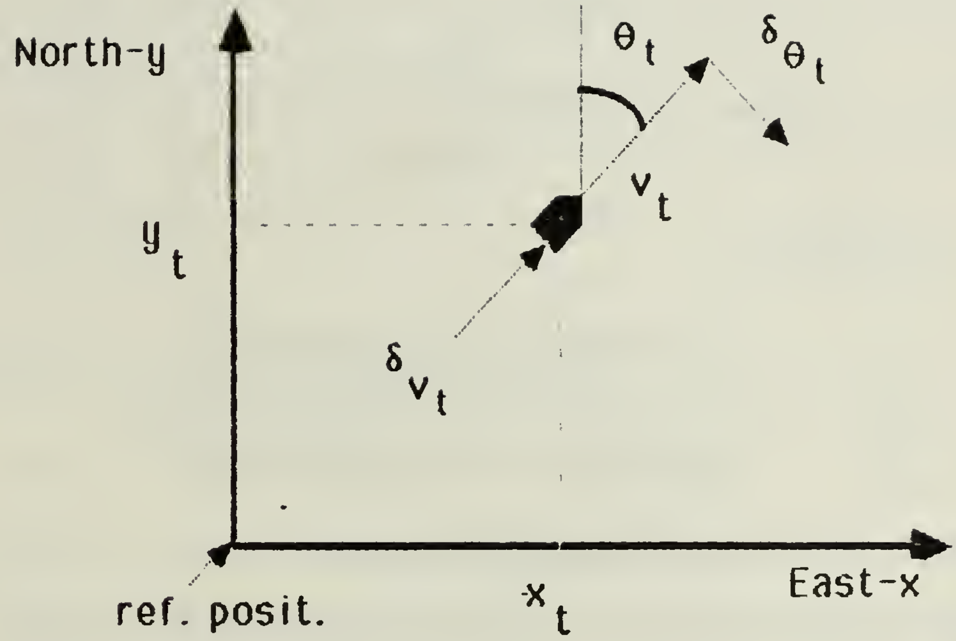

Figure 2-3 Geometry of Target Maneuvers

Also let the random variable δ_{f0} represent the change in frequency. The quantities δ_{vt}, $\delta_{\theta t}$,and δ_{f0} are random changes of the target which are assumed to be independent and zero mean

$$E[\delta_{vt}(k)] = \text{expected value of } \delta_{vt}(k)$$
$$= E[\delta_{\theta t}(k)] = E[\delta_{f0}(k)] = 0 \quad \text{for all } k \geqslant 0 . \qquad (2-4)$$

The variances of δ_{vt}, $\delta_{\theta t}$, δ_{f0} are define by

$$\sigma_{\dot{v}t}^2 \equiv E[\delta_{vt}^2]$$
$$\sigma_{\dot{\theta}t}^2 \equiv E[\delta_{\theta t}^2] \qquad (2-5)$$
$$\sigma_{\dot{f}0}^2 \equiv E[\delta_{f0}^2] .$$

Further it is assumed that the random variables remain constant over the discrete time interval T (i.e., a random walk). The following values for the standard deviations were taken from estimated maneuvering parameters for the target. [Ref. 3:p. 39]

$$\sigma_{\dot{v}t} = 0.01 \text{ kts/sec}$$
$$\sigma_{\dot{\theta}t} = 0.1 \text{ degree/sec} \quad (2\text{-}6)$$
$$\sigma_{\dot{f}_0} = 0.001 \text{ hz/sec} .$$

Hence the variances are

$$\sigma_{\dot{v}t}^2 = (0.01\text{kts/sec})^2 = 410.8 \text{ yds}^2/\text{min}^4$$
$$\sigma_{\dot{\theta}t}^2 = (0.1\text{deg/sec})^2 = 0.01096 \text{ rads}^2/\text{min}^2 \quad (2\text{-}7)$$
$$\sigma_{\dot{f}0}^2 = (0.001\text{hz/sec})^2 = 0.0036 \text{ hz}^2/\text{min}^2$$

These values are used in the scenarios in Section VI .

Using the definition of T and the equations of motion in the x-y plane, the difference equations can be obtained from Figure (2-1) and Figure (2-3)

$$\underline{x}(k+1) = \begin{bmatrix} x_t(k+1) \\ v_{xt}(k+1) \\ y_t(k+1) \\ v_{yt}(k+1) \\ f_0(k+1) \end{bmatrix} = \begin{bmatrix} x_t(k) + T\cdot v_{xt}(k) + g_1(\delta_{vt},\delta_{\theta t},k) \\ v_{xt}(k) + g_2(\delta_{vt},\delta_{\theta t},k) \\ y_t(k) + T\cdot v_{yt}(k) + g_3(\delta_{vt},\delta_{\theta t},k) \\ v_{yt}(k) + g_4(\delta_{vt},\delta_{\theta t},k) \\ f_0(k) + g_5(\delta_{f0}) \end{bmatrix} \quad (2\text{-}8)$$

The random forcing function terms g_i trough g_5 are included to account for random changes in speed, heading and frequency which can occur for a maneuvering target. Writing equation (2-8) in more familiar terms

$$\begin{bmatrix} x_t(k+1) \\ v_{xt}(k+1) \\ y_t(k+1) \\ v_{yt}(k+1) \\ f_0(k+1) \end{bmatrix} = \begin{bmatrix} 1 & T & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & T & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_t(k) \\ v_{xt}(k) \\ y_t(k) \\ v_{yt}(k) \\ f_0(k) \end{bmatrix} + \begin{bmatrix} T^2/2 & 0 & 0 \\ T & 0 & 0 \\ 0 & T^2/2 & 0 \\ 0 & T & 0 \\ 0 & 0 & T \end{bmatrix} \begin{bmatrix} w_1(k) \\ w_2(k) \\ w_3(k) \end{bmatrix} \quad (2\text{-}9)$$

where w_1 represents the rate of change of speed and heading with respect to the x component.
w_2 represents the rate of change of speed and heading with respect to the y component.
w_3 represents the rate of change of the frequency component.

Hence, our system model is in the linear form

$$\underline{x}(k+1) = \Phi(k) \cdot \underline{x}(k) + \Gamma(k) \cdot \underline{w}(k) \qquad (2-10)$$

where $x(k)$ is the N x 1 dimensional state vector,
$\Phi(k)$ is the N x N dimensional state transition matrix,
$w(k)$ is the M x 1 dimensional vector of random forcing functions,
$\Gamma(k)$ is the N x M dimensional state forcing matrix,
k is the discrete time index representing time t_k.

B. NOISE-FREE MEASUREMENT EQUATION

As a vessel moves through the ocean, it radiates an acoustic signature. Information about the vessel can be obtained, if the sonobuoy receives the acoustic signature. As indicated in section I, a bearing to the target can be determined from a DIFAR sonobuoy. The frequency spectrum of the acoustic signature can be found, if the sonobuoy is either a DIFAR or LOFAR. The doppler effect relates the change in received frequency at a sensor to the relative motion between the target and sensor. Thus from the received acoustic signature, two different types of measurements are available for determining position, course, and speed of the target. In Cartesian coordinates, from Figure (2-1) the noise-free Doppler equation, [Ref. 4:p. 258] and [Ref. 5:p. 24] may be written as

$$f_d = \frac{f_0 \cdot v_p}{v_p + \frac{(x_t - x_b) \cdot v_{xt} + (y_t - y_b) \cdot v_{yt}}{\sqrt{(x_t - x_b)^2 + (y_t - y_b)^2}}} \qquad (2-11)$$

where v_p is the speed of sound in the water,
f_0 is the frequency radiated by the target,
$v_{xt} = v_t \cdot \sin\theta_t$ is the speed along the x axis,
$v_{yt} = v_t \cdot \cos\theta_t$ is the speed along the y axis.

For underwater tracking the observed doppler shifts are quite small. The speed of sound, v_p, is approximately 3000 knots in the water so a 5 knot target has a velocity/speed of sound ratio of 5/3000 or approximately 0.16%. This corresponds to ± 0.5 hertz for a radiated frequency of 300 hertz.

The noise-free bearing equation obtain from Figure (2-1) is

$$\theta = \tan^{-1}\left[\frac{x_t - x_b}{y_t - y_b}\right] \qquad (2-12)$$

where $\tan^{-1}[\ ]$ is the inverse tangent function.

C. MULTIPLE SONOBUOY PROBLEM

Figure (2-4) illustrates the geometry of the multiple sonobuoy problem [Ref. 4:p. 256]. As discussed in Subsection II.B, a DIFAR buoy can provide a bearing and frequency measurement. While a LOFAR buoy can provide only a frequency measurement. The reference point in Figure (2-4) can represent any latitude(y coordinate) and longititude(x coordinate). The target's position and sonobuoy positions are in relation to the reference point. Hence, the latitude and longitude for any of these positions can easily be obtained.

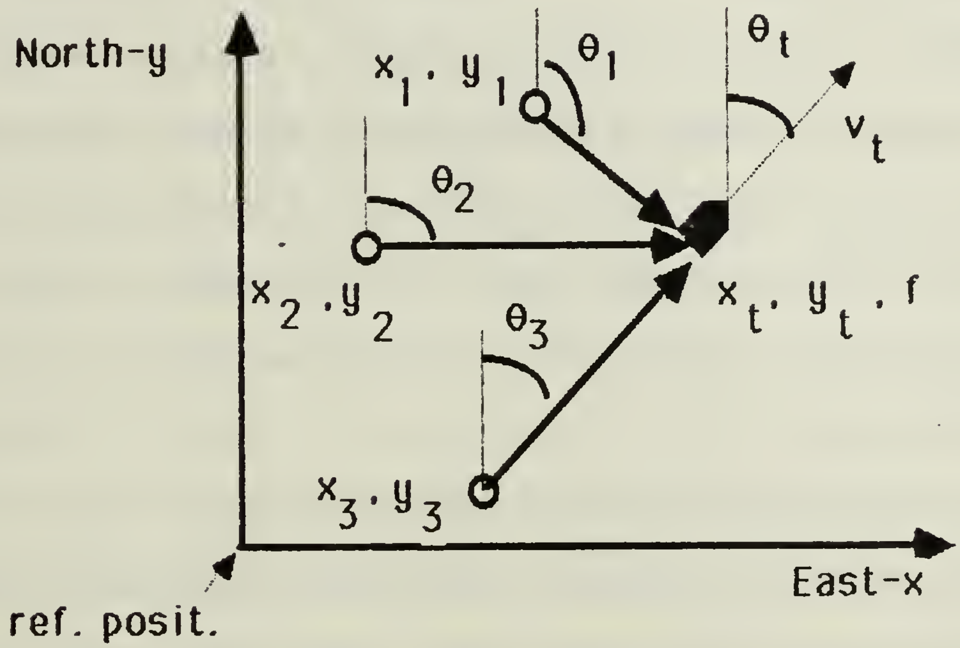

Figure 2-4 Multiple Sonobuoy Problem Geometry

III. KALMAN FILTERS AND EXTENDED KALMAN FILTERS

A. THE KALMAN FILTER

The purpose of the Kalman filter is to keep track of the state of the system by means of a sequence of noisy measurements. Conceptually, this technique may be viewed as a method to systematically reduce the measurement errors associated with the observations of the target's acoustic signature to determine an optimal estimate of the target's position and motion.

When both the system and measurement models are linear functions of the state variables the Kalman filter is the appropriate technique to use. As given in section II, the discrete time Kalman filter dynamic system is described by the state equation

$$\underline{x}(k+1) = \Phi(k) \cdot \underline{x}(k) + \Gamma(k) \cdot \underline{w}(k) \qquad (2-10)$$

and the measurement equation is related to the state by

$$\underline{z}(k) = H(k) \cdot \underline{x}(k) + \underline{v}(k) \qquad (3-1)$$

where $\underline{x}(k)$ is the N x 1 dimensional state vector,
$\Phi(k)$ is the N x N dimensional transition matrix,
$\underline{w}(k)$ is the M x 1 dimensional vector of random forcing functions,
$\Gamma(k)$ is the N x M dimensional state forcing matrix,
$\underline{z}(k)$ is the P x 1 dimensional measurement vector,
$H(k)$ is the P x n dimensional measurement matrix,
$\underline{v}(k)$ is the P x 1 measurement noise, and
k is the discrete time index representing time t_k.

B. THE EXTENDED KALMAN FILTER

Looking at equations (2-11) and (2-12) it can be seen that our measurement equations are nonlinear functions of the state variables. The nonlinear measurement model becomes

$$\underline{z}(k) = \underline{h}(\underline{x}(k),k) + \underline{v}(k) \qquad (3-2)$$

where the measurement, $\underline{z}(k)$, is a function, $\underline{h}(\underline{x}(k),k)$ of the state variables plus some noise (i.e. error), $\underline{v}(k)$. The $\underline{h}(\underline{x}(k),k)$ must be "linearized." This is accomplished by expanding $\underline{h}$ in a Taylor series about the best estimate of the state at the time and only the first order terms are kept [Ref. 3:p.34] and [Ref. 2:p.182]. The application of the Kalman filter to the nonlinear case is called the Extended Kalman filter. Higher order, more precise approximations to the optimal nonlinear filter can be acheived using more terms of the Taylor Series expansion for the nonlinearities, and by derviving recursive relations for the higher moments of state variables. For a derivation and additional discussion of Extended Kalman filtering the reader can refer to [Ref. 2:pp. 180-225] or [Ref. 6:pp. 219-292]. The following will be a summarized discussion of the discrete Extended Kalman filter equations. The linear form of equation (3-2) yields

$$\underline{z}(k) = H(k)\cdot\underline{x}(k) + \underline{v}(k) \qquad (3-3a)$$

where

$$H(k) = \frac{\partial h(\underline{x}(k),k)}{\partial \underline{x}(k)} \bigg|_{\underline{x}(k) = \hat{\underline{x}}(k|k-1)} \qquad (3-3b)$$

$\hat{\underline{x}}(k)$ is the N x 1 dimensional estimate state after the kth measurement. $\hat{\underline{x}}(k|k-1)$ is the N-dimensional predicted values of the state vector before the kth measurement. That is

$$\hat{\underline{x}}(k|k-1) = \Phi(k)\cdot\hat{\underline{x}}(k-1|k-1) \ . \qquad (3-4)$$

The following assumptions are made:

1. The measurement noise is zero mean and is uncorrelated with covariance $R(k)$

$$E[\underline{v}(k)] = 0, \text{ for all } k \geq 0 \tag{3-5a}$$

$$E[\underline{v}(k)\cdot\underline{v}(j)^T] = \begin{cases} 0, & k \neq j \\ R(k), & k = j \end{cases} \tag{3-5b}$$

$$\equiv R(k)\cdot\delta_{kj}, \text{ for all } k,j \geq 0.$$

2. The random forcing function is zero mean and uncorrelated with covariance $Q'(k)$

$$E[\underline{w}(k)] = 0, \text{ for all } k \geq 0 \tag{3-6a}$$

$$E[\underline{w}(k)\cdot\underline{w}(j)^T] = Q'(k)\cdot\delta_{kj} \text{ for all } k,j \geq 0. \tag{3-6b}$$

3. The random forcing input and measurement noise are uncorrelated

$$E[\underline{w}(k)\cdot\underline{v}(j)^T] = E[\underline{v}(j)\cdot\underline{w}(j)^T] = 0 \text{ for all } k,j \geq 0. \tag{3-7}$$

4. The initial state is a random variable with known mean and covariance

$$E[\underline{x}(0)] = \hat{x}_0 \ , \tag{3-8}$$

and

$$E\{[\underline{x}(0) - \hat{\underline{x}}_0]\cdot[\underline{x}(0) - \hat{\underline{x}}_0]^T\} = P(0|-1) = P_0 \ . \tag{3-9}$$

5. The initial state and measurement noise are uncorrelated

$$E[\underline{x}(0)\cdot\underline{v}(k)^T] = E[\underline{v}(k)\cdot\underline{x}(0)^T] = 0 \ , \text{for all } k \geq 0 \ . \tag{3-10}$$

6. The random forcing input and initial state are uncorrelated

$$E[\underline{w}(k)\cdot\underline{x}(0)^T] = E[\underline{x}(0)\cdot\underline{w}(k)^T] = 0 \text{ for all } k \geq 0 \ . \tag{3-11}$$

The state estimation error vector $\tilde{\underline{x}}(k)$ is define by the estimated state vector minus the true state vector

$$\tilde{\underline{x}}(k) \equiv \hat{\underline{x}}(k|k) - \underline{x}(k) \ . \tag{3-12}$$

and the predicted state estimation error vector is defined by the predicted state vector minus the true state vector

$$\tilde{\underline{x}}(k|k-1) \equiv \hat{\underline{x}}(k|k-1) - \underline{x}(k) \ . \tag{3-13}$$

The covariance of estimation error matrix is given by

$$P(k|k-1) = E\{\tilde{\underline{x}}(k)\cdot\tilde{\underline{x}}(k)^T\} \; . \qquad (3-14)$$

and the predicted covariance of state error matrix is

$$P(k|k-1) = E\{\tilde{\underline{x}}(k|k-1)\cdot\tilde{\underline{x}}(k|k-1)^T\} \qquad (3-15)$$

The state excitation covariance matrix is given by

$$Q(k) = \Gamma(k)\cdot E\{\underline{w}(k)\cdot\underline{w}(k)^T\}\cdot\Gamma(k)^T \qquad (3-16)$$
$$= \Gamma(k)\cdot Q'(k)\cdot\Gamma(k)^T \; .$$

If the estimate is selected to have the form

$$\hat{\underline{x}}(k|k) = \hat{\underline{x}}(k|k-1) + G(k)\cdot[\underline{z}(k) - \underline{h}(\hat{\underline{x}}(k|k-1))] \qquad (3-17)$$

and the optimal estimator is defined as the one that minimizes the sum of the variances of estimation error, i.e.

$$E\{\tilde{\underline{x}}_1(k)^2\} + E\{\tilde{\underline{x}}_2(k)^2\} + \ldots + E\{\tilde{\underline{x}}_n(k)^2\}$$

then the optimal estimation gains are those which satisfy the equations

$$G(k) = P(k|k-1)\cdot H^T(k)\cdot[H(k)\cdot P(k|k-1)\cdot H^T(k) + R(k)]^{-1} \qquad (3-18)$$

$$P(k|k) = [I - G(k)\cdot H(k)]\cdot P(k|k-1) \qquad (3-19)$$

$$P(k+1|k) = \Phi(k)\cdot P(k|k)\cdot\Phi(k)^T + Q(k) \qquad (3-20)$$

If the estimation equation (3-17) is initialized with the value

$$\hat{\underline{x}}(0|-1) = \hat{\underline{x}}_0 \, , \qquad (3-21)$$

it can be shown that the optimal estimate $\hat{x}(k|k)$ is unbiased i.e.

$$E\{[\hat{\underline{x}}(k|k) - \hat{\underline{x}}(k)]\} = 0 \quad \text{for all } k \geqslant 0 \; . \qquad (3-22)$$

and the initial condition is

$$E\{[\underline{x}(0) - \hat{\underline{x}}_0]\cdot[\underline{x}(0) - \hat{\underline{x}}_0]^T\} = P(0|-1) = P_0 \quad . \qquad (3-23)$$

In summary, Table 3-1 defines the discrete Extended Kalman filter algorithm for the linear state equation (2-10) and the linearized measurement equation (3-3). Equations (3-17), (3-18, (3-19), (3-4), and (3-20) comprise the Extended Kalman filter recursive equations.[Ref. 2:p. 190] Once the loop is entered, it can be continue ad infinitum, in principle at least. The pertinent equations and the sequence of computational steps are shown in Figure (3-1).

Table 3-1 SUMMARY OF KALMAN EQUATIONS

SYSTEM MODEL: $\underline{x}(k+1) = \Phi(k)\underline{x}(k) + \Gamma(k)\underline{w}(k)$ (2-10)

where $\underline{w}(k) \sim N[0, \mathbf{Q}'(k)]$

MEASURE MODEL: $\underline{z}(k) = \underline{h}(x(k),k) + \underline{v}(k)$ (3-2)

where $\underline{v}(k) \sim N[\underline{0}\ R(k)]$

INITIAL CONDITIONS: $\underline{x}(0) \sim N[\hat{\underline{x}}_0\ , \underline{P}_0]$ (3-8) and (3-9)

OTHER ASSUMPATIONS: $E[\underline{w}(k)\cdot\underline{v}(j)^T] = 0$ for all $k, j \geq 0$. (3-7)

GAIN EQUATION:

$$G(k) = P(k|k-1)\cdot H^T(k)\cdot[H(k)\cdot P(k|k-1)\cdot H^T(k) + R(k)]^{-1} \quad (3\text{-}18)$$

ERROR COVARIANCE UPDATE EQUATION:

$$P(k|k) = [I - G(k)\cdot H(k)]\cdot P(k|k-1) \quad (3\text{-}19)$$

STATE ESTIMATE UPDATE EQUATION:

$$\hat{\underline{x}}(k|k) = \hat{\underline{x}}(k|k-1) + G(k)\cdot[\underline{z}(k) - \underline{h}(\hat{x}(k|k-1))] \quad (3\text{-}17)$$

ERROR COVARIANCE PROPAGATION EQUATION:

$$P(k+1|k) = \Phi(k)\cdot P(k|k)\cdot\Phi(k)^T + Q(k) \quad (3\text{-}20)$$

STATE ESTIMATE PROPAGATION:

$$\hat{\underline{x}}(k+1|k) = \Phi(k)\cdot\hat{\underline{x}}(k|k) \quad (3\text{-}4)$$

DEFINITIONS:

$$H(k) = \left.\frac{\partial h(\underline{x}(k),k)}{\partial \underline{x}(k)}\right|_{\underline{x}(k) = \hat{\underline{x}}(k|k-1)} \quad (3\text{-}3b)$$

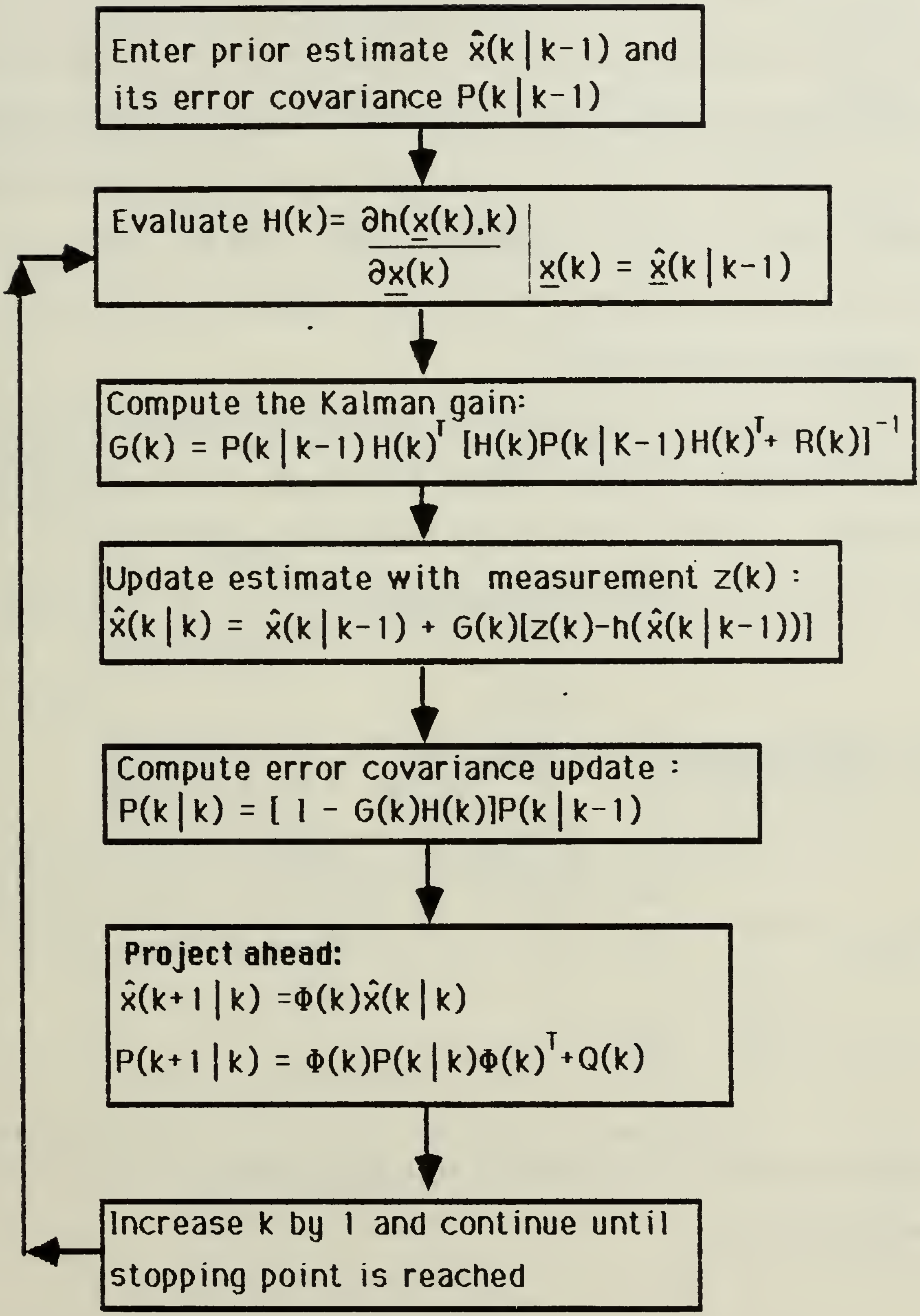

Figure 3-1 Kalman Filter Recursive Loop

C. FUNCTIONS, MATRICES, AND EQUATIONS

In the following paragraphs the application of the discrete Extended Kalman filter to the passive acoustic tracking problem will be discussed. Given the system and noise-free measurement models developed in Section II, we will derive the random forcing function $\underline{w}(k)$, the state excitation covariance matrix Q(k), the measurement equation $\underline{z}(k)$, and the measurement noise covariance matrix R(k).

1. Random forcing function

From equation (2-10) the vector $\underline{w}(k)$ represents the effect on the states of the random forcing function and may be calculated from equations relating v_{xt} and v_{yt} to the target heading, θ_t, and velocity, v_t. From Figure (2-1) the velocity in the x direction is

$$\dot{x}_t = v_{xt} = v_t \cdot \sin\theta_t . \qquad (3\text{-}24)$$

Differentiateted equation (2-9) gives

$$\ddot{x}_t = \dot{v}_{xt} = v_t \cdot \dot{\theta}_t \cdot \cos\theta_t + \dot{v}_t \cdot \sin\theta_t \quad . \qquad (3\text{-}25)$$

where $\sin\theta_t = \dfrac{v_{xt}}{v_t}$

$\cos\theta_t = \dfrac{v_{yt}}{v_t}$.

Letting $\dot{\theta}_t = \delta_{\theta t}$ and $\dot{v}_t = \delta_{vt}$ and substituting into equation (3-25), the acceleration becomes

$$w_1(k) = \ddot{x}_t = v_{yt} \cdot \delta_{\theta t} + \frac{(v_{xt})}{v_t} \cdot \delta_{vt} \quad . \qquad (3\text{-}26)$$

Similarly,

$$\dot{y}_t = v_{yt} = v_t \cdot \cos \theta_t \text{ , and} \qquad (3\text{-}27)$$

$$\ddot{y}_t = \dot{v}_{yt} = - v_t \cdot \dot{\theta}_t \cdot \sin \theta_t + \dot{v}_t \cdot \cos \theta_t \ . \qquad (3\text{-}28)$$

and after substitution

$$w_2(k) = \ddot{y}_t = - v_{xt} \cdot \delta_{\theta t} + \frac{(v_{yt})}{v_t} \cdot \delta_{vt} \ . \qquad (3\text{-}29)$$

The frequency term becomes

$$w_3(k) = \dot{f}_c = \delta_{fo} \ . \qquad (3\text{-}30)$$

From equation (2-4) and the assumptions on the δ's, we conclude that $\underline{w}$ is zero mean

$$E[\underline{w}(k)] = 0 \ . \qquad (3\text{-}31)$$

The random forcing functions covariance matrix is

$$Q'(k) = E[\underline{w}(k) \cdot \underline{w}^T(k)]$$

$$= \begin{bmatrix} E[w_1(k)^2] & E[w_1(k) \cdot w_2(k)] & E[w_1(k) \cdot w_3(k)] \\ E[w_2(k) \cdot w_1(k)] & E[w_2 k)^2] & E[w_2(k) \cdot w_3(k)] \\ E[w_3(k) \cdot w_1(k)] & E[w_3(k) \cdot w_2(k)] & E[w_3(k)^2] \end{bmatrix} \qquad (3\text{-}32)$$

Since δ_{vt}, $\delta_{\theta t}$, and δ_{fo} are independent and zero mean

$$E[w_1(k) \cdot w_3(k)] = E[w_3(k) \cdot w_1(k)] = 0 \qquad (3\text{-}33a)$$

and

$$E[w_2(k) \cdot w_3(k)] = E[w_3(k) \cdot w_2(k)] = 0 \qquad (3\text{-}33b)$$

Let the variances of $w_1(k)$, $w_2(k)$, and $w_3(k)$ be defined as

$$\begin{aligned} \sigma_{\ddot{x}}^2 &\equiv E[w_1(k)^2] \\ \sigma_{\ddot{y}}^2 &\equiv E[w_2(k)^2] \\ \sigma_{\dot{f}o}^2 &\equiv E[w_3(k)^2] \\ \sigma_{\ddot{x}\ddot{y}} &\equiv E[w_1(k) \cdot w_2(k)] \end{aligned} \qquad (3\text{-}34)$$

Substituting equation (3-26) for $w_1(k)$ and cancelling the cross terms, the variance of $w_1(k)$ becomes

$$\sigma_{\dot{x}}^2 = \frac{(v_{xt})^2}{v_t} \cdot E[\delta_{vt}^2] + v_{yt}^2 \cdot E[\delta_{\theta t}^2] \quad . \qquad (3\text{-}35a)$$

From equation (2-5) the variance simplifies to

$$\sigma_{\dot{x}}^2 = \frac{(v_{xt})^2}{v_t} \cdot \sigma_{\dot{v}t}^2 + v_{yt}^2 \cdot \sigma_{\dot{\theta}t}^2 \quad . \qquad (3\text{-}35b)$$

The variance of $w_2(k)$ and $w_3(k)$ can be found in similar fashion to give

$$\sigma_{\dot{y}}^2 = \frac{(v_{yt})^2}{v_t} \cdot \sigma_{\dot{v}t}^2 + v_{xt}^2 \cdot \sigma_{\dot{\theta}t}^2 \quad . \qquad (3\text{-}36)$$

Upon substituting for $w_1(k)$, and $w_2(k)$, and simplifying the covariance term $E[w_1(k) \cdot w_2(k)]$ becomes

$$\sigma_{\dot{x}\dot{y}} = v_{xt} \cdot v_{yt} \cdot ((\sigma_{\dot{v}t}^2/v_t^2) - \sigma_{\dot{\theta}t}^2) \qquad (3\text{-}37)$$

Substituting equations (2-21) through (2-25) into equation (2-19) the $Q'(k)$ matrix result is

$$Q'(k) = E[\underline{w}(k) \cdot \underline{w}^T(k)] = \begin{bmatrix} \sigma_{\dot{x}}^2 & \sigma_{\dot{x}\dot{y}} & 0 \\ \sigma_{\dot{x}\dot{y}} & \sigma_{\dot{y}}^2 & 0 \\ 0 & 0 & \sigma_{\dot{f}o}^2 \end{bmatrix} \qquad (3\text{-}38)$$

where $\sigma_{\dot{x}}^2$, $\sigma_{\dot{y}}^2$, $\sigma_{\dot{x}\dot{y}}$ are evaluated at the predicted values of v_{xt} and v_{yt}.

2. State Excitation Covariance Matrix

To compute the error covariance propagation equation (3-19) the state excitation covariance matrix $Q(k)$ must be known. The size of $Q(k)$ has a direct bearing on the magnitude of the $P(k+1|k)$ [Ref. 2:p. 76]. The possibility of a maneuvering target and model inaccuracies are taken into account by the state excitation covariance matrix. As more and more data is processed, $Q(k)$ prevents the Kalman gains from approaching zero by insuring some uncertainty in the predicted state vector. From equation (3-16), the state excitation covariance matrix is

$$Q(k) = \Gamma(k) \cdot Q'(k) \cdot \Gamma(k)^T \quad . \qquad (3\text{-}16)$$

Substituting from equation (2-9) for $\Gamma(k)$ and (3-38) for $Q'(k)$ the state excitation covariance matrix is

$$Q(k) = \begin{bmatrix} \frac{T^4 \cdot \sigma_{\ddot{x}}^2}{4} & \frac{T^3 \cdot \sigma_{\ddot{x}}^2}{2} & \frac{T^4 \cdot \sigma_{\ddot{x}\ddot{y}}}{4} & \frac{T^3 \cdot \sigma_{\ddot{x}\ddot{y}}}{2} & 0 \\ \frac{T^3 \cdot \sigma_{\ddot{x}}^2}{2} & T^2 \cdot \sigma_{\ddot{x}}^2 & \frac{T^3 \cdot \sigma_{\ddot{x}\ddot{y}}}{2} & T^2 \cdot \sigma_{\ddot{x}\ddot{y}} & 0 \\ \frac{T^4 \cdot \sigma_{\ddot{x}\ddot{y}}}{4} & \frac{T^3 \cdot \sigma_{\ddot{x}\ddot{y}}}{2} & \frac{T^4 \cdot \sigma_{\ddot{y}}^2}{4} & \frac{T^3 \cdot \sigma_{\ddot{y}}^2}{2} & 0 \\ \frac{T^3 \cdot \sigma_{\ddot{x}\ddot{y}}}{2} & T^2 \cdot \sigma_{\ddot{x}\ddot{y}} & \frac{T^3 \cdot \sigma_{\ddot{y}}^2}{2} & T^2 \cdot \sigma_{\ddot{y}}^2 & 0 \\ 0 & 0 & 0 & 0 & T^2 \cdot \sigma_{\dot{f}_o}^2 \end{bmatrix} \qquad (3\text{-}39)$$

3. Measurement Equation

From equation (3-2) the nonlinear measurement equation is

$$\underline{z}(k) = \underline{h}(\underline{x}(k),k) + \underline{v}(k) \qquad (3\text{-}2)$$

As indicated in equation (3-3a) and (3-3b) the linear form of this equation is

$$\underline{z}(k) = H(k) \cdot \underline{x}(k) + \underline{v}(k) \qquad (3\text{-}3a)$$

where $$H(k) = \left. \frac{\partial \underline{h}(\underline{x}(k),k)}{\partial \underline{x}(k)} \right|_{\underline{x}(k) = \hat{\underline{x}}(k|k-1)} \qquad (3\text{-}3b)$$

Substituting the doppler equation (2-11) and bearing equation (2-12) into $\underline{h}(\underline{x}(k),k)$, the measurement model becomes

$$\underline{z}(k) = \begin{bmatrix} f_d(k) \\ \\ \theta(k) \end{bmatrix} = \begin{bmatrix} \dfrac{f_o(k)\cdot v_p}{v_p + \dfrac{(x_t(k)-x_b)\cdot v_{xt}(k) + (y_t(k)-y_b)\cdot v_{yt}(k)}{\sqrt{(x_t(k)-x_b)^2 + (y_t(k)-y_b)^2}}} \\ \\ \tan^{-1}\left(\dfrac{x_t(k)-x_b}{y_t(k)-y_b}\right) \end{bmatrix} + \begin{bmatrix} v_f(k) \\ \\ v_\theta(k) \end{bmatrix} \qquad (3\text{-}40)$$

The linearized measurement matrix is derived from equation (3-3b) and is shown below

$$H(k) =$$

$$\begin{bmatrix} \dfrac{\partial f_d(k|k-1)}{\partial x_t(k|k-1)} & \dfrac{\partial f_d(k|k-1)}{\partial v_{xt}(k|k-1)} & \dfrac{\partial f_d(k|k-1)}{\partial y_t(k|k-1)} & \dfrac{\partial f_d(k|k-1)}{\partial v_{yt}(k|k-1)} & \dfrac{\partial f_d(k|k-1)}{\partial f_o(k|k-1)} \\ \\ \dfrac{\partial \theta(k|k-1)}{\partial x_t(k|k-1)} & \dfrac{\partial \theta(k|k-1)}{\partial v_{xt}(k|k-1)} & \dfrac{\partial \theta(k|k-1)}{\partial y_t(k|k-1)} & \dfrac{\partial \theta(k|k-1)}{\partial v_{yt}(k|k-1)} & \dfrac{\partial \theta(k|k-1)}{\partial f_o(k|k-1)} \end{bmatrix} \qquad (3\text{-}41)$$

To simplify notation let

$$u(k|k-1) = [\hat{x}_t(k|k-1)-x_b(i)]\cdot\hat{v}_{xt}(k|k-1)+[\hat{y}_t(k|k-1)-y_b(i)]\cdot\hat{v}_{yt}(k|k-1) \,. \qquad (3\text{-}42)$$

and using from equation (2-1) the estimated range

$$r(k|k-1) = \sqrt{[\hat{x}_t(k|k-1)-x_b(i)]^2+[\hat{y}_t(k|k-1)-y_b(i)]^2} \,. \qquad (3\text{-}43)$$

where $x_b(i)$ and $y_b(i)$ is the x and y components of the the ith sonobuoy in the buoy pattern.

For the frequency measurement, the $H_{1,5}(k)$ component will be evaulated first in order to reduce notation. The results of the partial derivatives of the frequency measurement evaluated at the predicted states values $\hat{x}(k|k-1)$ are

$$H_{1,5}(k) = \frac{\partial f_d(k|k-1)}{\partial f_o(k|k-1)} = \frac{V_p}{V_p + \frac{u(k|k-1)}{r(k|k-1)}} \qquad (3\text{-}44e)$$

Let $Ak \equiv - \frac{f_o(k|k-1)\cdot[H_{1,5}(k)]^2}{V_p + [r(k|k-1)]^2}$ then

$$H_{1,1}(k) = Ak\cdot\left(\hat{v}_{xt}(k|k-1)\cdot r(k|k-1) - \frac{u(k|k-1)\cdot[\hat{x}_t(k|k-1)-x_b(i)]}{r(k|k-1)}\right) \qquad (3\text{-}44a)$$

$$H_{1,2}(k) = Ak\cdot\left(r(k|k-1)\cdot[\hat{x}_t(k|k-1)-x_b(i)]\right) \qquad (3\text{-}44b)$$

$$H_{1,3}(k) = Ak\cdot\left(\hat{v}_{yt}(k|k-1)\cdot r(k|k-1) - \frac{u(k|k-1)\cdot[\hat{y}_t(k|k-1)-x_b(i)]}{r(k|k-1)}\right) \qquad (3\text{-}44c)$$

$$H_{1,4}(k) = Ak\cdot\left(r(k|k-1)\cdot[\hat{y}_t(k|k-1)-y_b(i)]\right) \qquad (3\text{-}44d)$$

Similarly, the partial derivatives of the bearing angle measurement evaluated at the predicted state values $\hat{x}(k|k-1)$ follows:

$$H_{2,1}(k) = [\hat{y}_t(k|k-1)-y_b(i)] \qquad (3\text{-}45a)$$

$$H_{2,2}(k) = 0 \qquad (3\text{-}45b)$$

$$H_{2,3}(k) = -[\hat{x}_t(k|k-1)-x_b(i)] \qquad (3\text{-}45c)$$

$$H_{2,4}(k) = 0 \qquad (3\text{-}45d)$$

$$H_{2,5}(k) = 0 \qquad (3\text{-}45e)$$

4. <u>Measurement Noise Covariance Matrix</u>

To compute the Kalman filter gains, the measurement noise covariance matrix R(k) must be known. From Table 3-1 and equations (3-5a and (3-5b) the measurement noises, $v_f(k)$ and $v_\theta(k)$ are assumed to be zero mean and uncorrelated. The noise covariance matrix is define as

$$R(k) = \begin{bmatrix} \sigma_f^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix} . \tag{3-46}$$

where σ_f is the standard deviation of the frequency measurement noise
σ_θ is the standard deviation of the bearing measurement noise

As indicated by Mitschang [Ref. 3:p. 43], the resolution of the frequency measurement is equal to the inverse of the record length of the time signal. Various numbers from 0.02 to 1.0 hertz were tested in the simulations. A typical value of the frequency standard deviation is

$$\sigma_f = 0.04 \text{ hertz} . \tag{3-47}$$

The magnitude of the bearing measurement noise is a function of the signal-to-noise ratio at the sonobuoy, which is influenced by several environmental factors and a function of the signal processor. Since most of the simultations runs were close tracking scenarios, a typical bearing standard deviation value is

$$\sigma_\theta = \pm 5 \text{ degrees} . \tag{3-48}$$

D. MANEUVERING AND DIVERGENCE CONTROL

An unprecise model may cause the filter to diverge. Divergence occurs when the calculated error covariance does not bound the actual error covariance. In otherwords when the calculated covariance matrix error covariance. In otherwords when the calculated covariance matrix becomes too small or optimistic. When the calculated covariance matrix becomes small, hence the filter gain is small, subsequent measurements have little effect on the estimate. So the estimated state and the actual state diverge because the system model in the filter is different than the actual system model. Such model errors are due to the following:

1. Approximations that might be made to simplify the filter computations
2. Limited knowledge of the physical system.
3. Computational errors resulting from the use of finite precision arithmetic.

In order to prevent divergence and to accommodate maneuvering, the basic idea is to increase the calculated covariance matrix, $P(k|k)$; since model errors are compensated by a larger calculated covariance matrix. However, this increase in the calculated error covariance makes the filter more sensitive to random errors in the measurement process, often resulting in poorer target estimates when the target does not undergo a maneuver. As a result, an adaptive control method has been devised to increase the calculated error covariance only when the target has maneuvered. Further details on divergence is delineated in by Jazwinski [Ref.7:pp. 301-305] and Gelb [Ref. 2:pp. 277-311].

The first procedure used to prevent divergence and to allow for maneuvering is the random forcing function $\underline{w}(k)$. As indicated in Subsection III.C.2, $Q(k)$ prevents the Kalman gains from approaching zero by insuring some uncertainty in the predicted state vector.

The second, maneuver and divergence control procedure involves the development of an adaptive gate. From equation (3-17) the difference between the actual measurement and the predicted measurement, is defined as the predicted measurement residual error (called hereafter predicted residual)

$$e(k|k-1) \equiv \underline{z}(k) - \underline{h}(\hat{\underline{x}}(k|k-1) \quad . \qquad (3\text{-}49)$$

Jazwinski [Ref. 7:p. 271] provides the statiscal properties of the predicted residual which are

$$E[e(k|k-1)] = 0 \quad , \qquad (3\text{-}50)$$

and

$$Z(k|k-1) \equiv E[e(k|k-1 \cdot e(k|k-1)^T]$$
$$= H(k) \cdot P(k|k-1) \cdot H(k)^T + R(k) \quad , \qquad (3\text{-}51)$$

Therefore the predicted residual standard deviation is defined as

$$\sigma_z = \sqrt{Z(k|k-1)} = \sqrt{H(k) \cdot P(k|k-1) \cdot H(k)^T + R(k)} \, . \qquad (3\text{-}52)$$

In order to allow for target maneuvers, we define an adaptive gate as three times the predicited residual standard deviation

$$\text{Gate3}(k) \equiv 3 \cdot \sigma_z \, . \qquad (3\text{-}53)$$

We can judge the performance of the filter by comparing the predicted residual to the adaptive gate. For each measurement the algorithm tests the residual and lets the residual itself determine the appropriate random forcing function

$$|e(k|k-1)| < 3 \cdot \sigma_z \quad . \qquad (3\text{-}54)$$

The adaptive gate is adaptive in the following sense. As long as the predicted residual remains less than the $3 \cdot \sigma_z$ value, the random forcing function covariance matrix Q'(k) remains as calculated by equation (3-38).. When the predicted residual becomes larger than the $3\cdot\sigma_z$ value the filter is diverging. To prevent divergence the variances of the random forcing function (equation (3-34)) are increased as follows:

$$\begin{aligned} \sigma_{\ddot{x}}^2 \text{ new} &= 10\cdot\sigma_{\ddot{x}}^2 \text{ old}, \\ \sigma_{\ddot{y}}^2 \text{ new} &= 10\cdot\sigma_{\ddot{y}}^2 \text{ old}, \text{and} \\ \sigma_{\dot{f}_0}^2 \text{ new} &= 2\cdot\sigma_{\dot{f}_0}^2 \text{ old}. \end{aligned} \qquad (3\text{-}55)$$

The constants in equation (3-55) were found by trial and error. The random forcing function covariance matrix increases, which increases Q(k). When the state excitation matrix, Q(k), increases, the covariance matrix, $P(k|k-1)$ increases. The covariance matrix causes the adaptive gate to increase. Note H(k) and R(k) remain the same. Hence, the gate opens the filter to the incoming measurement. At the next iteration the variances of the random forcing function reverts back to equation (3-34) to calculate random forcing function covariance matrix and the state excitation covariance matrix for the next measurement.

If the predicted residual exceeds the adaptive gate three consecutive times, the algorithm is reinitialized to the original estimated error covariance matrix. The reinitialized error covariance for the simulations in Section VI is

$$P(k|k-1) = \begin{bmatrix} (0.5\text{ nm})^2 & 0 & 0 & 0 & 0 \\ 0 & (3\text{ kts})^2 & 0 & 0 & 0 \\ 0 & 0 & (0.5\text{ nm})^2 & 0 & 0 \\ 0 & 0 & 0 & (3\text{ kts})^2 & 0 \\ 0 & 0 & 0 & 0 & (1\text{hz})^2 \end{bmatrix} \qquad (3\text{-}56)$$

IV. ERROR ELLIPSOIDS

A. INTRODUCTION

Error ellipsoids are useful in visualizing the estimation error. With them we can consider the true state value to lie within a certain region surrounding the estimate .. This uncertainy is expressed in the covariance of error matrix $P(k|k)$. The concept of the error ellipsoid is summarized.

> DEFINITION. Suppose the n -dimensional vector random variable $\underline{x}$ has a multivariate gaussian distribution with a mean value of zero and covariance $E[\underline{x}\cdot\underline{x}^T]=P$. The "error ellipsoids" are defined as n-dimensional surfaces of constant probability density. [Ref. 6:p. 252]

The probability density function of $\underline{x}$ has the multivariate gaussian form

$$f_x(x) = (2\pi)^{-n/2}\left|\det P\right|^{-1/2}\cdot\exp[-1/2(\underline{x}^T\cdot P^{-1}\cdot\underline{x})] \qquad (4-1)$$

From which the surface of constant probability density is described as

$$\underline{x}^T\cdot P^{-1}\cdot\underline{x} = c^2 \qquad (4-2)$$

where c^2 is an arbitrary constant. The name "error ellipsoid " comes from the surface of constant probability density. The surface is an ellipsoid, if P is a nonnegative definite matrix. The ellipsoids have a simple probabilistic interpretation. For a specified value of c the probability that $\underline{x}$ lies within or on the ellipsoid is obtained by integrating the probability density function over the surface of the ellipsoid. Table 4-1 lists the probabilities for a few values of n and c [Ref. 8:p. 4-49].

From the equations given in Table 3-1 we can assume that the initial state of the system $\hat{\underline{x}}_0$ and the random noise processes $\underline{w}(k)$ and $\underline{v}(k)$ are

Table 4-1. PROBABILITIES FOR ERROR ELLIPSES

	c		
n	1	2	3
1	.683	.955	.997
2	.394	.865	.989
3	.200	.739	.971

Gaussian. If these Gaussian assumptions are satisfied then the following are also Gaussian:

1. The state, $\underline{x}(k)$, since it is a linear function of $\underline{w}(k-1)$.
2. The estimate, $\hat{\underline{x}}(k|k)$, which is a linear combination of $\underline{x}(k)$ and $\underline{v}(k)$.
3. The estimate error, $\tilde{\underline{x}}(k) = \hat{\underline{x}}k|k) - \underline{x}(k)$

As indicated in Section III, the state estimation error is zero mean with covariance of error $P(k|K)$. If $P(k|k)$ is nonnegative definite the surface is an ellipsoid.

B. APPLICATION

The first and third components of the state vector (equation (2-1)) represent position, the second and fourth components represent velocity and the fifth component represent frequency. A ellipsoid for the total matrix is a conglomerate mess; so it is logical to examine a submatrix relating the state variables of the position components or the velocity components. A discussion on the position components follows, a similar examination can be made with the velocity components.

The position components submatrix of the error covariance matrix $P(k|k)$ is defined as

$$P_{xtyt}(k|k) \equiv \begin{bmatrix} P_{11}(k|k) & P_{13}(k|k) \\ P_{31}(k|k) & P_{33}(k|k) \end{bmatrix} = \begin{bmatrix} \text{var } x & \text{cov}(x,y) \\ \text{cov}(x,y) & \text{var } y \end{bmatrix} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy}^2 \\ \sigma_{xy}^2 & \sigma_y^2 \end{bmatrix} \qquad (4-3)$$

The diagonal terms $P_{11}(k|k)$ and $P_{33}(k|k)$ of the error covariance matrix represents the variances of the estimate in the x and y positions respectfully. The off diagonal term $P_{13}(k|K)$ represents the covariance of the estimate in x and y. This term describes the degree of coupling and the orientation of the uncertainty in the x-y plane.

The multivariante Gaussian distribution reduces to a bivariate Gaussian distribution with a probability density function given by

$$f_{xy}(xy) = (2\pi\sigma_x\sigma_y)^{-1}(1-r^2)^{-1/2} \cdot \exp\left[-\frac{\frac{x^2}{\sigma_x^2} - \frac{2\cdot r\cdot x\cdot y}{\sigma_x\sigma_y} + \frac{y^2}{\sigma_y^2}}{2(1-r^2)}\right] \qquad (4-4)$$

where the means of the random variables x and y are 0
the parameter r is called the correlation coefficient of the random variables x and y.

From which a curve of constant probability is described by

$$\frac{x^2}{\sigma_x^2} - \frac{2\cdot r\cdot x\cdot y}{\sigma_x\sigma_y} + \frac{y^2}{\sigma_y^2} = c^2 . \qquad (4-5)$$

This curve is an ellipse. The term error ellipsoid often refers to the specific case when c is set equal to one. From Table (4-1) for the case when n=2 and c=1 the probability that a sample point will be within the ellipsoid is .394 . The major and minor axis of this ellipse are not aligned with the coordinate system. Since the error estimate $\tilde{x}(k|k)$ is normally distributed, the coordinate system can be rotated in such a way that in the new system position components are uncorrelated. Let the matrix A represent a rotation of the axes through an angle θ

$$A = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} . \qquad (4-6)$$

By applying a linear transformation

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \qquad (4-7)$$

and picking the angle θ so that

$$\tan 2\theta = \frac{2 \cdot r \cdot \sigma_x \cdot \sigma_y}{\sigma_x^2 - \sigma_y^2} = \frac{2 \cdot \mathrm{cov}(x,y)}{\mathrm{var}\ x - \mathrm{var}\ y} \qquad (4-8)$$

hence,

$$\theta = \frac{1}{2} \cdot \tan^{-1} \left(\frac{2 \cdot P13(k|k)}{P_{11}(k|k) - P_{33}(k|k)} \right) \qquad (4-9)$$

$$= \frac{1}{2} \cdot \tan^{-1} \left(\frac{2 \cdot r \cdot \sigma_x \cdot \sigma_y}{\sigma_x^2 - \sigma_y^2} \right)$$

we obtain uncorrelated random variables x' and y' [Ref. 9:p. 159]. The variances in this system are calculated by:

$$\sigma_x'^2 = \frac{\sigma_x^2 + \sigma_y^2}{2} + \frac{\mathrm{cov}(x,y)}{\sin 2\theta} , \qquad (4-10)$$

and

$$\sigma_y'^2 = \frac{\sigma_x^2 + \sigma_y^2}{2} - \frac{cov(x,y)}{\sin 2\theta} \quad . \qquad (4-11)$$

In order to get a better understanding an example will be given. In Figure (4-1), the target is on a course of 180 degs at 10 kts. The DIFAR buoy is providing frequency and bearing measurements, while the LOFAR buoy is providing only frequency measurements. The error ellipsoids for this figure are twenty times their actual size. At time t_k = 10 min the DIFAR buoy's bearing measurement's position components submatrix of the error covariance matrix is

$$P_{xtyt}(k|k) \equiv \begin{bmatrix} 2.39 \times 10^4 & 3.96 \times 10^4 \\ 3.96 \times 10^4 & 7.34 \times 10^4 \end{bmatrix} \qquad (4-12)$$

Hence,

σ_x = 155 yds,
σ_y = 270 yds, and
σ_{xy} = 199 yds .

Substituting these numbers into equation (4-9) gives

$$\theta = \frac{1}{2} \cdot \tan^{-1}\left[\frac{2 \cdot 3.96 \times 10^4}{2.39 \times 10^4 - 7.34 \times 10^4}\right] = -29 \text{ degs} \qquad (4-13)$$

From equations (4-10) and (4-11) the new uncorrelated variances are

$$\sigma_x'^2 = \frac{2.39 \times 10^4 + 7.34 \times 10^4}{2} + \frac{3.96 \times 10^4}{\sin 2(-29)} , \approx 1.95 \times 10^3 \text{ yds}^2$$

and

$$\sigma_y'^2 = \frac{2.39 \times 10^4 + 7.34 \times 10^4}{2} - \frac{3.96 \times 10^4}{\sin 2(-29)} . \approx 9.53 \times 10^4 \text{ yds}^2$$

The new standard deviations are

$\sigma_x' \approx$ 44 yds, and
$\sigma_y' \approx$ 309 yds .

If these values are multiple by the scale factor, they agree with the values from Figure (4-1). Note the majority of the error is along the bearing line.

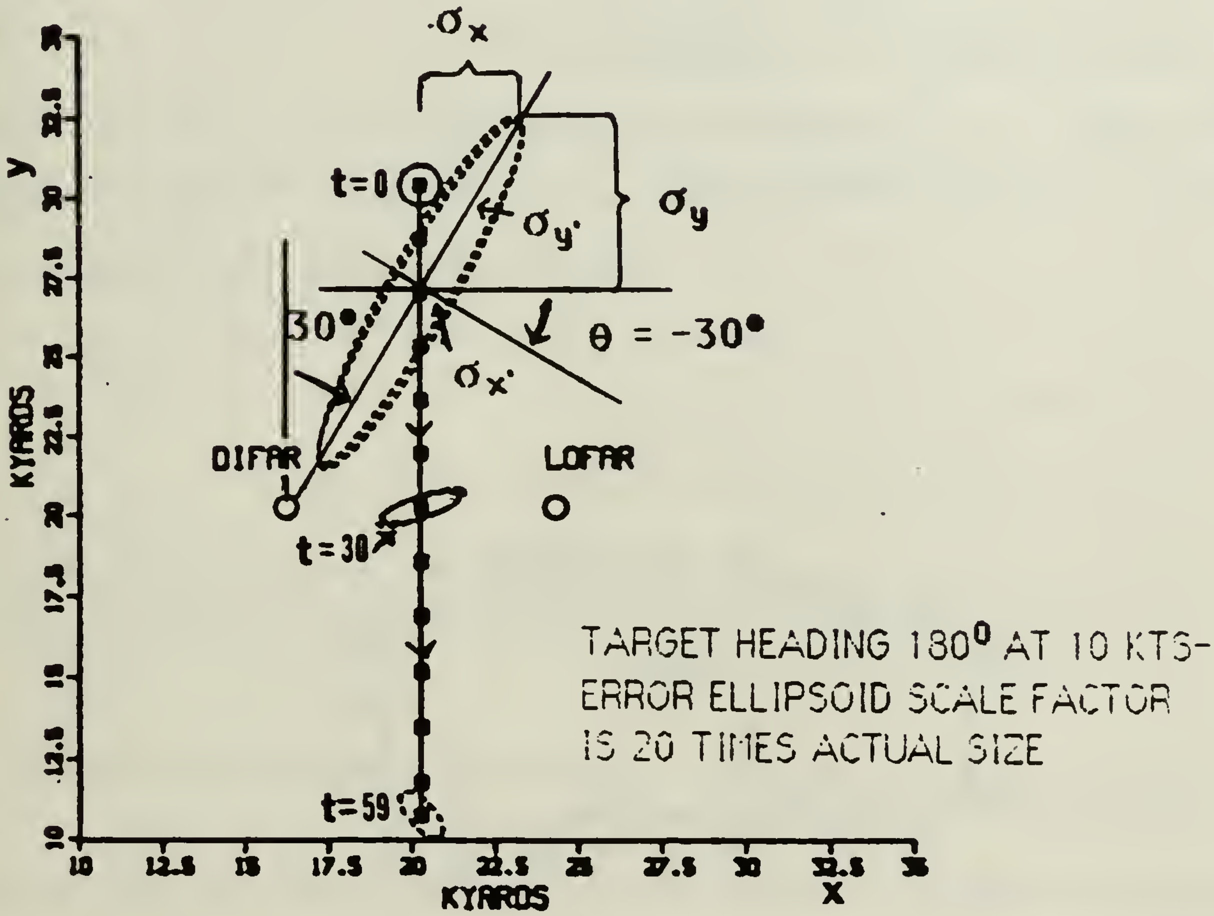

Figure 4-1. Error Ellipsoid Example

V. THE ALGORITHM

A. INTRODUCTION

This section discusses the development of the tracking algorithm. The tracking algorithm was designed to:

1. Require a priori target information
2. Minimize the data storage necessary
3. Produce in all quadrants of the coordinate system an accurate estimate of the target position in a reasonably short period of time
4. Require no human intervention once the algorithm is initiated.

To accomplish this, the algorithm was divided into three modules. The modules performed the following tasks:

1. The track module generates a noise-free track data.
2. The observation module receives the noise-free track data generated by the track module and simulates noisy measurement data for input to the tracking algorithm module.
3. The tracking algorithm module receives noisy measurement date and generates estimates and predictions.

All computer programs were written in FORTRAN 77 and executed on the IBM 3033 located at the Naval Postgraduate School, Monterey, California.

B. TARGET'S TRACK

To evalute the performance of the tracking algorithm a pattern similar to the one illustrated by O'Connor, Findley, and Nitsche [Ref. 40: 245] was developed. A typical track for a 5 kt target is shown in Figure (5-1). The target's initial position, speed and course are

x-coordinate = 10 nm (20253.7 yds)
y-coordinate = 12 nm (24304.4 yds)
speed = 5 kts
course = 180 degs .

The solid line is the positional time history of the target's track. The small circles indicate the position every five minutes. The target algorithm and output is contained in Appendix A. Table (5-1) lists the times that the various maneuvers occur in Figure (5-1). Other target algorithms were developed and tested, but Figure (5-1) pattern is used in all simulations presented in Section VI.

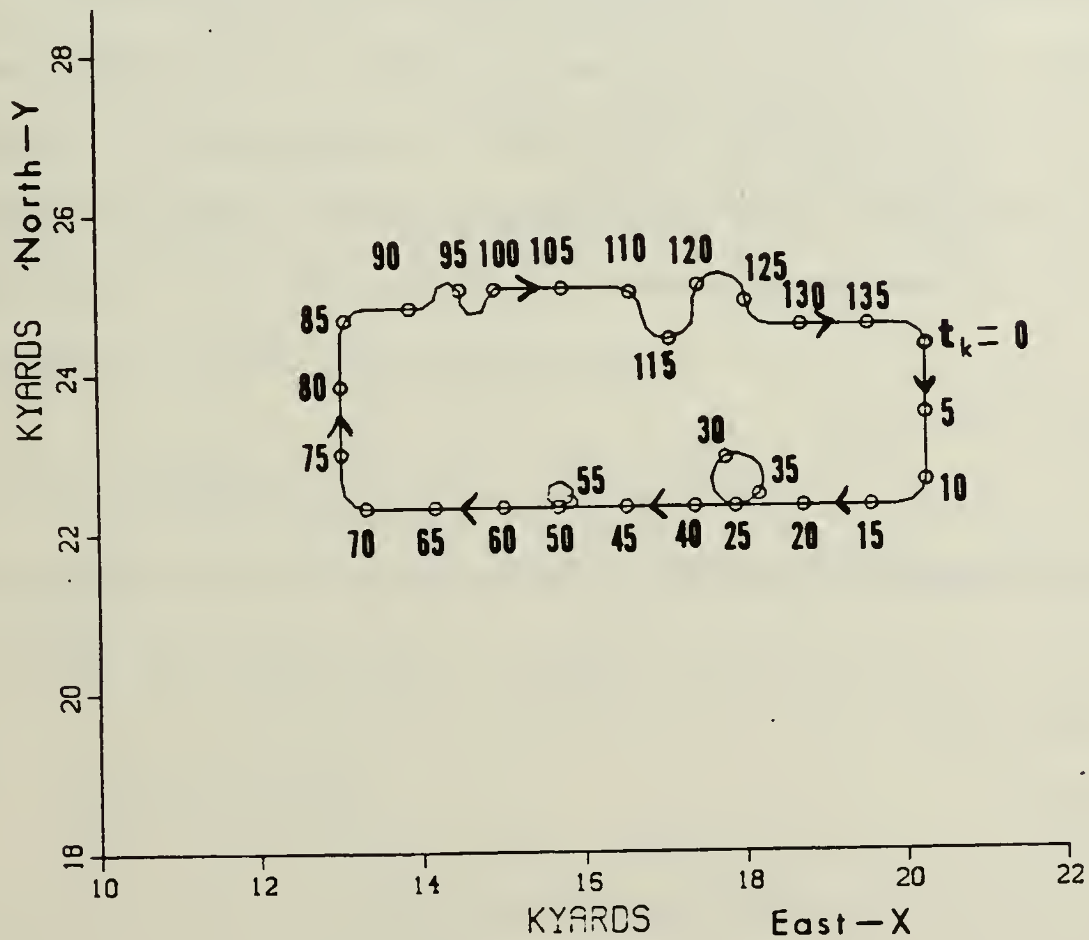

Figure 5-1 Typical Track of the Target.

Table 5-1. MANEUVER TIMES FOR FIGURE 5-1.

Time (mins)	Maneuver
0-10	straight
11-13	90° starboard turn
14-25	straight
26-37	large 360° starboard turn
38-50	straight
51-56	small 360° starboard turn
57-70	straight
71-73	90° starboard turn
74-84	straight
85-87	90° starboard turn
88-91	straight
92-101	small "s" turn
102-109	straight
110-128	large "s" turn
129-137	straight
138-140	90° starboard turn

C. OBSERVATION PACKET

Based on the target's position and the sonobuoy pattern, the noisy frequency and bearing measurements are computed. It is convenient to think of each observation as arriving in a packet(called the observation packet) containing

1. the time,
2. identification of the sonobuoys having contact,
3. the type of measurement (frequency or bearing),

4. the measurement, and

5. its standard deviation.

1. Time

The tracking algorithm makes no assumption concerning the regularity of the measurement arrival times. Measurements are likely to occur at random times. These times are determine by the target algorithm. For the simulations in Section VI, the target algorithm updates the target's position every minute.

2. Sonobuoy Pattern

The sonobuoy pattern must be known to the algorithm at the start of the simulation. The sonobuoy positions are assumed to remain stable throughout the simulation. There are existing methods to estimate sonobuoy position drift. An input data set provides the following information:

1. Number of sonobuoys in the pattern

2. Type of sonobuoy (ie DIFAR or LOFAR), and

3. The sonobuoy position (x coordinate and y coordinate in nautical miles).

An example of this data set is contained in Appendix A.

3. Noisy Measurement

Since the noise-free target position and the various sonobuoy positions are known, the noise-free bearing measurements can be obtained from equation (2-12). To calculate the noise-free frequency measurement from the Doppler equation (2-11), the following quantities must be known:

1. The center frequency, f_0, (the true radiated frequency) against which all relative Doppler inputs are compared, and

2. An estimate of the speed of sound in the water, v_p.

These quantities are supplied by the user at the start of the simulation. The measurement noise is assumed to have a Gaussian distribution with zero mean and known variance. We assume that the amount of the measurement noise depends on the distance the target is from the sonobuoy. So the value of the measurement noise standard deviation, depends on the range which is obtained from equation (2-1). Table (5-2) lists two sets of typical standard deviations used for the frequency measurement noise. Two sets of bearing measurement noise standard deviations are listed in Table (5-3).

The IMSL routines GAUSS and GGUBS are combined to provide a Gaussian pseudo-random number generator. Therefore, by inputting the noise-free measurement and its measurement noise standard deviation into the Gaussian pseudo-random number generator, a noisy measurement is obtained.

Table 5-2. FREQUENCY MEASUREMENT STANDARD DEVIATIONS

Range	Set 1	Set 2
r < 2 nm	σ_f = 0.02 hz	σ_f = 0.04 hz
2 nm < r < 5 nm	σ_f = 0.04 hz	σ_f = 0.06 hz
5 nm < r < 10 nm	σ_f = 0.08 hz	σ_f = 0.08 hz
r > 10 nm	σ_f = 0.1 hz	σ_f = 0.1 hz

Table 5-3. BEARING MEASUREMENT STANDARD DEVIATIONS

Range	Set 1	Set 2
$r < 2$ nm	$\sigma_b = 2$ degs	$\sigma_b = 5$ degs
2 nm $< r < 5$ nm	$\sigma_b = 5$ degs	$\sigma_b = 10$ degs
5 nm $< r < 10$ nm	$\sigma_b = 10$ degs	$\sigma_b = 15$ degs
$r > 10$ nm	$\sigma_b = 15$ degs	$\sigma_b = 15$ degs

D. MULTIPLE SONOBUOY TRACKING ALGORITHM

The structure of the multiple sonobuoy tracking algorithm is diagramed in Figure (5-2). The required inputs are

1. The initial state estimation vector $\hat{\underline{x}}(0|-1)$ and the initial estimated covariance of error matrix $P(0|-1)$, and

2. The observation packet .

The multiple sonobuoy tracking algorithm is divided into three stages. The first stage is the initialization of the Kalman filter. The Extended Kalman filter algorithm is contained in the second stage. Stages one, and two, are combined with the observation packet into one FORTRAN program which is contained in Appendix B. The last stage consists of the filter's graphical output, which is contained in Appendix C.

1. Initialization

In the first series of simulations, different combinations of the initial state estimate $\hat{\underline{x}}(0|-1)$ and the initial estimated error covariance $P(0|-1)$ were tested. Initially the choices of these values were somewhat arbitrary, but using common sense and through trial and error, an educated guess can be obtained for $\hat{\underline{x}}(0|-1)$ and $P(0|-1)$.

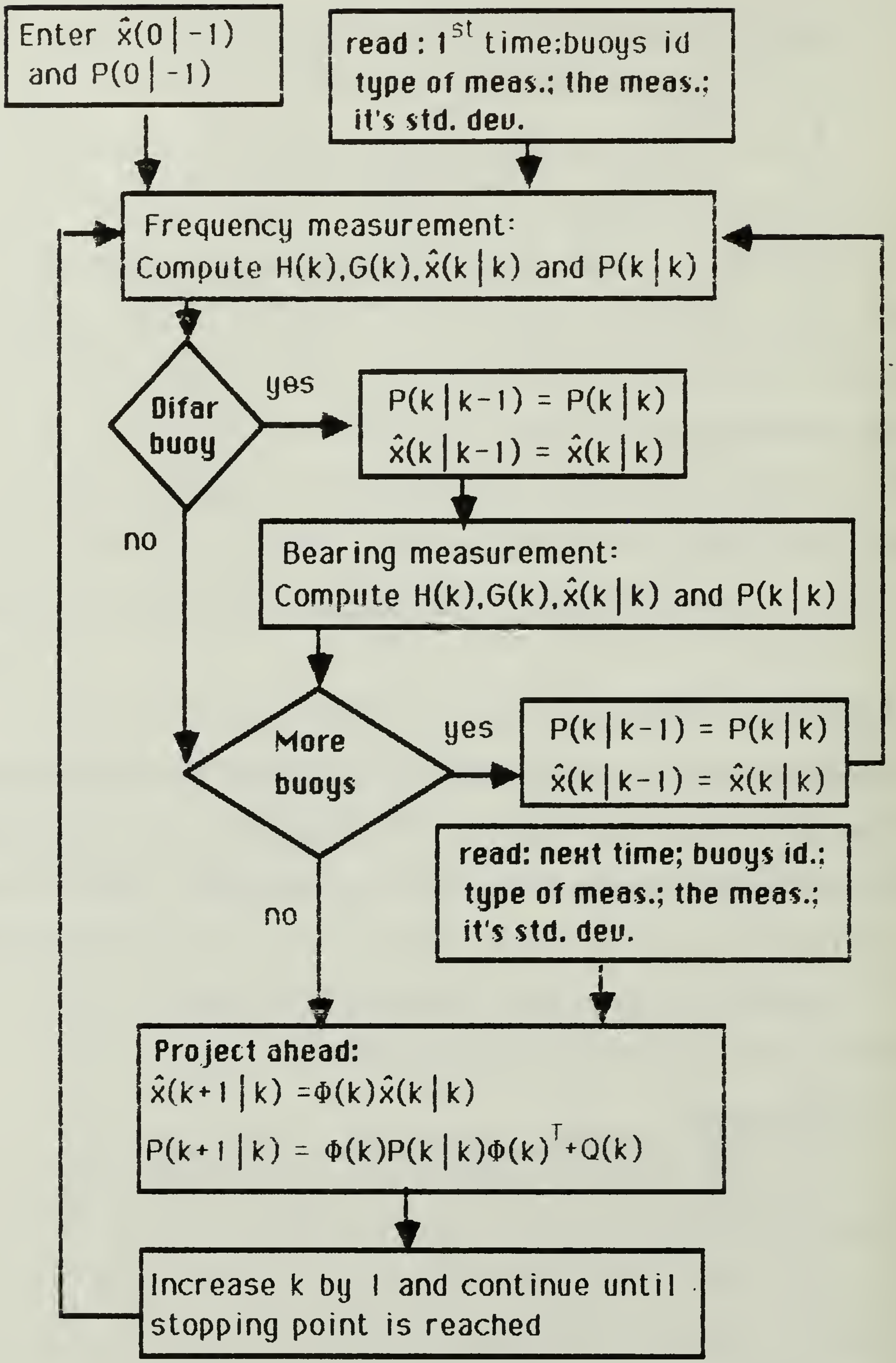

Figure 5-2 Multiple Sonobuoy Tracking Algorithm

The current version of the tracking algorithm requires a priori information in order to operate. This information is assumed to be available through human operators, received from other tracking routines, or received from some external sensor. Since this algorithm was design for close range tracking, the a priori information is assumed to be accurate (i.e., position is know within 2 nm, velocity is known within 5 kts, and course within 15 degs). To start the cyclic process the following quantities are required:

1. The target's estimated position, speed, and course,
2. Knowledge of the accuracy (i.e., standard deviation) of the above estimates,
3. The center frequency, f_o,
4. An estimate of the speed of sound in the water, v_p, and
5. The sonobuoy pattern entered in terms of buoy type of and position.

The initial estimated state vector $\underline{x}(0|-1)$ is obtained from f_o, v_p, and the target's estimated position, speed, and course. The target's estimated position, speed, and course are defined as

$x_e \equiv$ the x coordinate initial estimated position in nm,
$y_e \equiv$ the y coordinate initial estimated position in nm,
$v_e \equiv$ the estimated speed in kts, and
$h_e \equiv$ the estimated course in degs .

The a priori state estimation vector $\hat{\underline{x}}(0|-1)$ is defined as

$$\hat{x}(0|-1) \equiv \begin{bmatrix} x_e \\ v_{xe} \\ y_e \\ v_{ye} \\ f_e \end{bmatrix} \tag{5-1}$$

where $v_{xe} = v_e \sin h_e$

$v_{ye} = v_e \cos h_e$

$$f_e = \frac{f_0 \cdot v_p}{v_p + \dfrac{(x_e - x_b(1)) \cdot v_{xe} + (y_e - y_b(1)) \cdot v_{ye}}{\sqrt{(x_e - x_b(1))^2 + (y_e - y_b(1))^2}}}$$

The initial estimated covariance of error matrix is obtained from the accuracy of the target's estimated position, speed and course. The standard deviation of estimated position, speed, course, and frequency are defined as

$\sigma_p \equiv$ the standard deviation of the estimated position in nm,
$\sigma_{ve} \equiv$ the standard deviation of the estimated speed in kts,
$\sigma_{he} \equiv$ the standard deviation of the estimated course in degs, and
$\sigma_{fe} \equiv$ the standard deviation of the frequency in hz.

Typical values are

$\sigma_p = 0.5$ nm,
$\sigma_{ve} = 3$ kts,
$\sigma_{he} = 10$ degs, and
$\sigma_{fe} = 1$ hz

The a priori estimated covariance of error matrix is defined as

$$P(0|-1) \equiv \begin{bmatrix} \sigma_p^2 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{vxe}^2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_p^2 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{vye}^2 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{fe}^2 \end{bmatrix} \tag{5-2}$$

where $\sigma_{xe} = \sigma_{ve}\sin(h_e + \sigma_{he})$

$\sigma_{ye} = \sigma_{ve}\cos(h_e + \sigma_{he})$

2. <u>The Filter</u>

In order to gain a better understanding of the tracking algorithm recursive process the following examples will be discussed.

a. Example 1

Consider the case of only one DIFAR buoy having contact on the target. The algorithm reads the first observation packet which consists of

1. Initial time.
2. DIFAR buoy position.
3. The frequency measurement, and
4. The bearing measurement.

From the initialization stage we have the initial state estimate $\hat{x}(0|-1)$ and its error covariance $P(0|-1)$. Following Figures (3-2) and (3-1) the sequence of calculations goes like this

1. Evaluate $H(0)$ by applying the frequency measurement and $\hat{x}(0|-1)$ to equation (3-3b).
2. Compute the Kalman gain $G(0)$ by applying $P(0|-1)$ to equation (3-18).
3. Compute the updated error covariance for the frequency measurement $P_f(0|0)$ from equation (3-19).
4. Compute the updated state estimate for the frequency measurement $\hat{x}_f(0|0)$ from equation (3-17).

5. Since there is also a bearing measurement for this buoy, the update state estimate $\underline{x}_f(0|0)$ and error covariance $P_f(0|0)$ become the inputs to the bearing measurement estimates. Hence,

$$\underline{x}_b(0|-1) = \underline{x}_f(0|0), \text{ and}$$
$$P_b(0|-1) = P_f(0|0)$$

6. Evaluate $H(0)$ by applying the bearing measurement and $\hat{\underline{x}}_b(0|-1)$ to equation (3-30).

7. Compute the Kalman gain $G(0)$ by applying $P_b(0|-1)$ to equation (3-18).

8. Compute the updated error covariance for the bearing measurement $P_b(0|0)$ from equation (3-19).

9. Compute the updated state estimate for the bearing measurement $\hat{\underline{x}}_b(0|0)$ from equation (3-17).

10. The next step is to project ahead, but in order to project ahead the next time must be known. So,when the next observation packet comes available the algorithm computes $\hat{\underline{x}}(1|0)$ from equation (3-4) and $P(1|0)$ from equation (3-20).

The process is repeated by re-cycling through steps 1-10.

b. Example 2

Consider a two sonobuoy pattern, one DIFAR buoy (called DIFAR 1) and one LOFAR buoy (called LOFAR 2). The algorithm reads the first observation packet which consists of

1. Initial time,

2. DIFAR 1's position,

3. DIFAR 1's frequency measurement,

4. DIFAR 1's bearing measurement,

5. LOFAR 2's position.

6. LOFAR 2's frequency measurement.

From the initialization stage we have the initial state estimate $\underline{x}(0|-1)$ and its error covariance $P(0|-1)$. The sequence of calculations goes like this

1. From the first example the algorithm computes steps 1-9.

2. In order to compute a estimate from LOFAR 2's frequency measurement, the algorithm repeats step 5 from example one

$$\hat{\underline{x}}_f(0|-1) = \hat{\underline{x}}_b(0|0), \text{ and}$$
$$P_f(0|-1) = P_b(0|0).$$

3. From the first example Steps 1-4 are computed to obtain LOFAR 2's updated state estimate $\hat{\underline{x}}(0|0)$ and error covariance $P(0|0)$.

4. When the next observation packet comes available the algorithm projects ahead and the recursive process continues.

3. Adaptive Control

As discussed in Subsection III.D, an adaptive control method has been devised to increase the error covariance only when the filter detects divergence. The algorithm compares the predicted residual with the adaptive gate by implementing the adaptive control decision rule, equation (3-54),

$$|e(k|k-1)| < 3\cdot\sigma_z .$$

This test occurs during steps 4 and 9 in example one above. If the predicted residual is less than the adaptive gate the algorithm continues to compute the updated state estimate. If the predicted residual is greater than the adaptive gate the sequence of calculations goes like this

1. From equation (3-55) the random forcing function is increased.

2. Calculate a new state excitation matrix using equation (3-16)

3. Compute a new error covariance using
$$P(k|k-1)new = P(k|k-1)old + Q(k).$$

4. Compute the Kalman gain G(k) by applying P(k|k-1)new to equation (3-18).

5. Compute the predicted residual variance σ_z^2 from equation (3-51).

6. Compute the updated error covariance from equation (3-19).

7. Compare the predicted residual with the new adaptive gate using equation (3-54).

If the predicted residual is less than the adaptive gate the algorithm continues with step 4 or 9. If the predicted residual is greater than the adaptive gate repeat steps 1-7 above. If the adaptive gate is exceeded three consecutive times the error covariance matrix P(k|k-1) is reinitialized. The algorithm continues the process starting with step 4 above.

VI. SIMULATION RESULTS

A. INTRODUCTION

One of the objectives of this research was to produce a operational computer program for passively tracking a target from an airbourne platform. The purpose of this section is to demonstrate through scenarios the performance of the algorithm described in Section V. In the following pages five scenarios are presented. Scenarios 1 and 2 are presented to demonstrate a best-case performance. Changing the type of sonobuoys and increasing the measurement noise used in Scenarios 1 and 2, Scenario 3 and 4 are presented to demonstrate a worst-case performance. Scenario 5 demonstrates the initialization process.

B. SCENARIO 1

Scenario 1 consists of eight simulations that demonstrate the effects of the measurement noise, the state excitation covariance matrix, and the adaptive control method on the filters performance. In order to examine the effects of a state excitation covariance matrix and the adaptive control individually, two random forcing function covariance matrix are developed. The first random forcing function covariance matrix $Q'_1(k)$ is from equation (3-39). This covariance matrix, developed in Subsection III.C, is used in the majority of the simulations. A summary of $Q'_1(k)$ is given for completeness. From equation (3-39) $Q'_1(k)$ is defined as

$$Q'_1(k) \equiv Q'(k) = E[w(k)\cdot w^T(k)] = \begin{bmatrix} \sigma_{\ddot{x}}^2 & \sigma_{\ddot{x}\ddot{y}} & 0 \\ \sigma_{\ddot{x}\ddot{y}} & \sigma_{\ddot{y}}^2 & 0 \\ 0 & 0 & \sigma_{\dot{f}_0}^2 \end{bmatrix} \qquad (3\text{-}38)$$

where $\sigma_{\ddot{x}}^2$, $\sigma_{\ddot{y}}^2$, and $\sigma_{\ddot{x}\ddot{y}}$ are evaluated at the predicted values of v_t, v_{xt}, and v_{yt}. Recall from subsection III.C

$$\sigma_{\ddot{x}}^2 = \frac{(v_{xt})^2}{v_t} \cdot \sigma_{\dot{v}t}^2 + v_{yt}^2 \cdot \sigma_{\dot{\theta}t}^2 \;, \qquad (3\text{-}35a)$$

$$\sigma_{\ddot{y}}^2 = \frac{(v_{yt})^2}{v_t} \cdot \sigma_{\dot{v}t}^2 + v_{xt}^2 \cdot \sigma_{\dot{\theta}t}^2 \;\text{, and} \qquad (3\text{-}36)$$

$$\sigma_{\ddot{x}\ddot{y}} = v_{xt} \cdot v_{yt} \cdot \left(\left(\frac{\sigma_{\dot{v}t}}{v_t} \right)^2 - \sigma_{\dot{\theta}t}^2 \right) \qquad (3\text{-}37)$$

where the values of $\sigma_{\dot{v}t}^2$, $\sigma_{\dot{\theta}t}^2$, and $\sigma_{\dot{f}_0}^2$ are given in equation (2-7).

The intent of second random forcing function covariance matrix is to produce a state excitation covariance matrix that has little effect on the performance of the filter. In this situation only the adaptive control method or measurement noise is effecting the filter's performance. Let the second random forcing function $Q'_2(k)$ be defined as

$$Q'_2(k) = \begin{bmatrix} 10\ \text{yd}^2/\text{mins}^4 & 0 & 0 \\ 0 & 10\ \text{yd}^2/\text{mins}^4 & 0 \\ 0 & 0 & 0.01\ \text{hz}^2/\text{mins}^2 \end{bmatrix} \qquad (E\text{-}1)$$

Since $Q'_s(k)$ has small values, the state excitation covariance matrix will have small values. Hence, the state excitation covariance matrix will have minimal effect on the filter performance.

Scenario 1 is a three DIFAR buoy scenario. A geographic plot of the sonobuoy pattern, target's track and the filter's estimated track are shown in Figure (6-1). Enlarged plots of the eight simulations are shown in Figures (6-2)-(6-9). Table (6-1) lists the differences in the simulations. Four noise-free simulations are illustrated in Figures (6-2)-(6-5). In Figures (6-6) - (6-9) measurement noise from Table (5-2) set 1 and Table (5-3) set 1 is applied to the frequency and bearing measurements, respectively.

The target track is described in Subsection V.B . The dashed line is the filter's estimated track. Each "x" represents target's estimated position every five minutes (i.e., the first "x" is the target's estimated position from the updated state estimate $\hat{x}(0|0)$, the next "x" is from $\hat{x}(5|5)$, etc...). The sonobuoy positions are shown by circles. The algorithm generates esimates and predictions from the measurements in the following order:

1. Frequency measurement from DIFAR 1

2. Bearing measurement from DIFAR 1

3. Frequency measurement from DIFAR 2

4. Bearing measurement from DIFAR 2

5. Frequency measurement from DIFAR 3

6. Bearing measurement from DIFAR 3

The a priori information follows

x_e = 10 nm σ_D = 0.5 nm

y_e = 12 nm

v_e = 5 kts σ_{ve} = 3 kts

h_e = 180 degs σ_{he} = 10 degs

f_o = 300 hz σ_{fe} = 1 hz

v_s = 4860 ft/sec

The one standard deviation error ellipsoids shown are due to the covariance of error matrix position components [P11(k|k), P33(k|k), and P13(k|k)]. The error ellipsoids are displayed for times t_k = 0, t_k = 10, and t_k = 30 mins. The large dashed circle is the error ellipsoid from DIFAR 1's frequency measurement. Similarly, the large horizontal ellipse is due to DIFAR 1's bearing measurement. As can be seen the error ellipsoids become smaller as more measurements are taken.

Table 6-1 SCENARIO 1 SIMULATIONS

Figure	Sim.	Noise Applied	Q'(k) Applied	Adapt. Cont. Applied
6-2	1	No	$Q'_2(k)$	No
6-3	2	No	$Q'_2(k)$	Yes
6-4	3	No	$Q'_1(k)$	No
6-5	4	No	$Q'_1(k)$	Yes
6-6	5	Yes	$Q'_2(k)$	No
6-7	6	Yes	$Q'_2(k)$	Yes
6-8	7	Yes	$Q'_1(k)$	No
6-9	8	Yes	$Q'_1(k)$	Yes

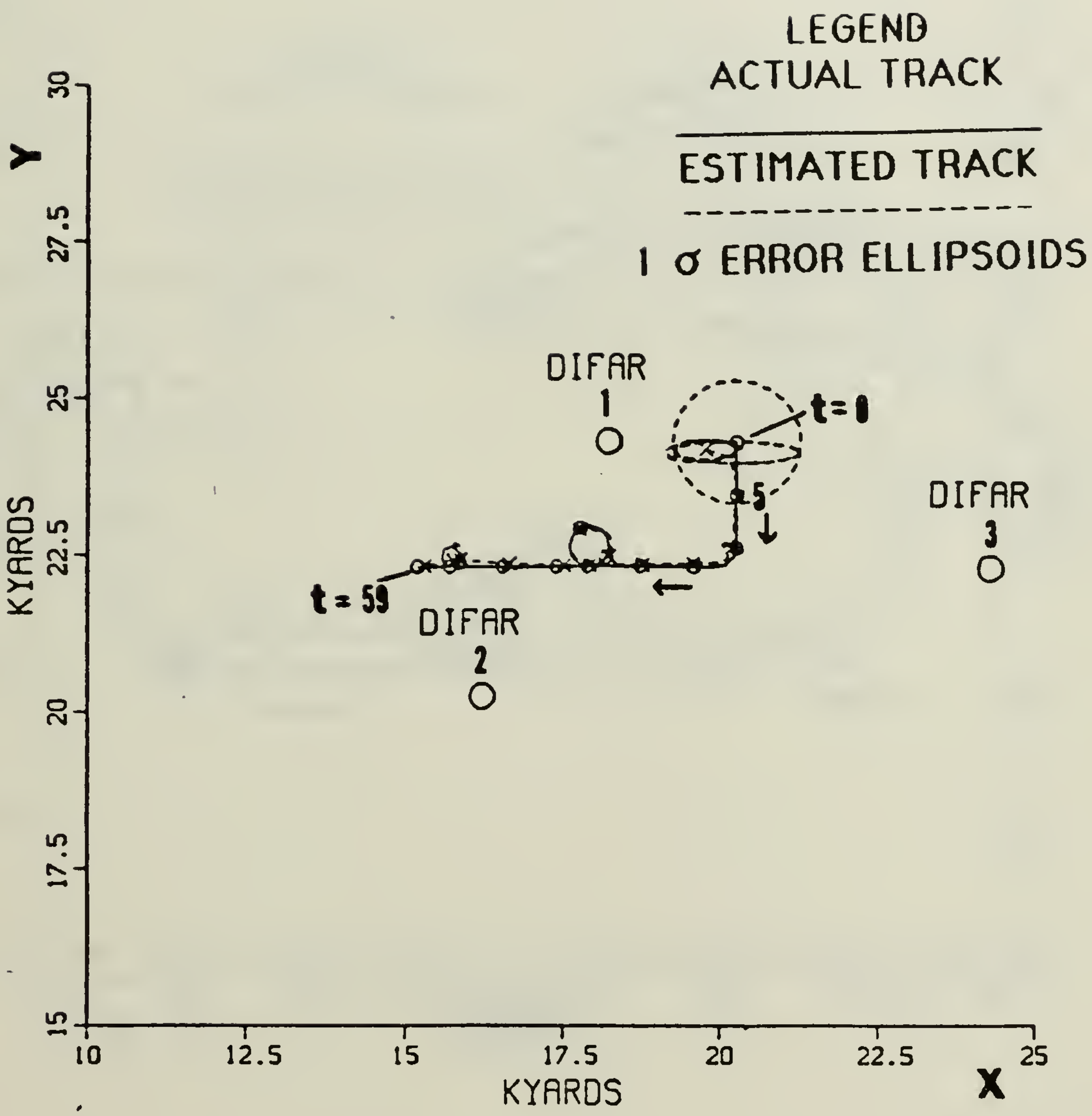

Figure 6-1. Scenario 1: Geographic Plot-Noise, $Q'_1(k)$, and Adaptive Control Applied

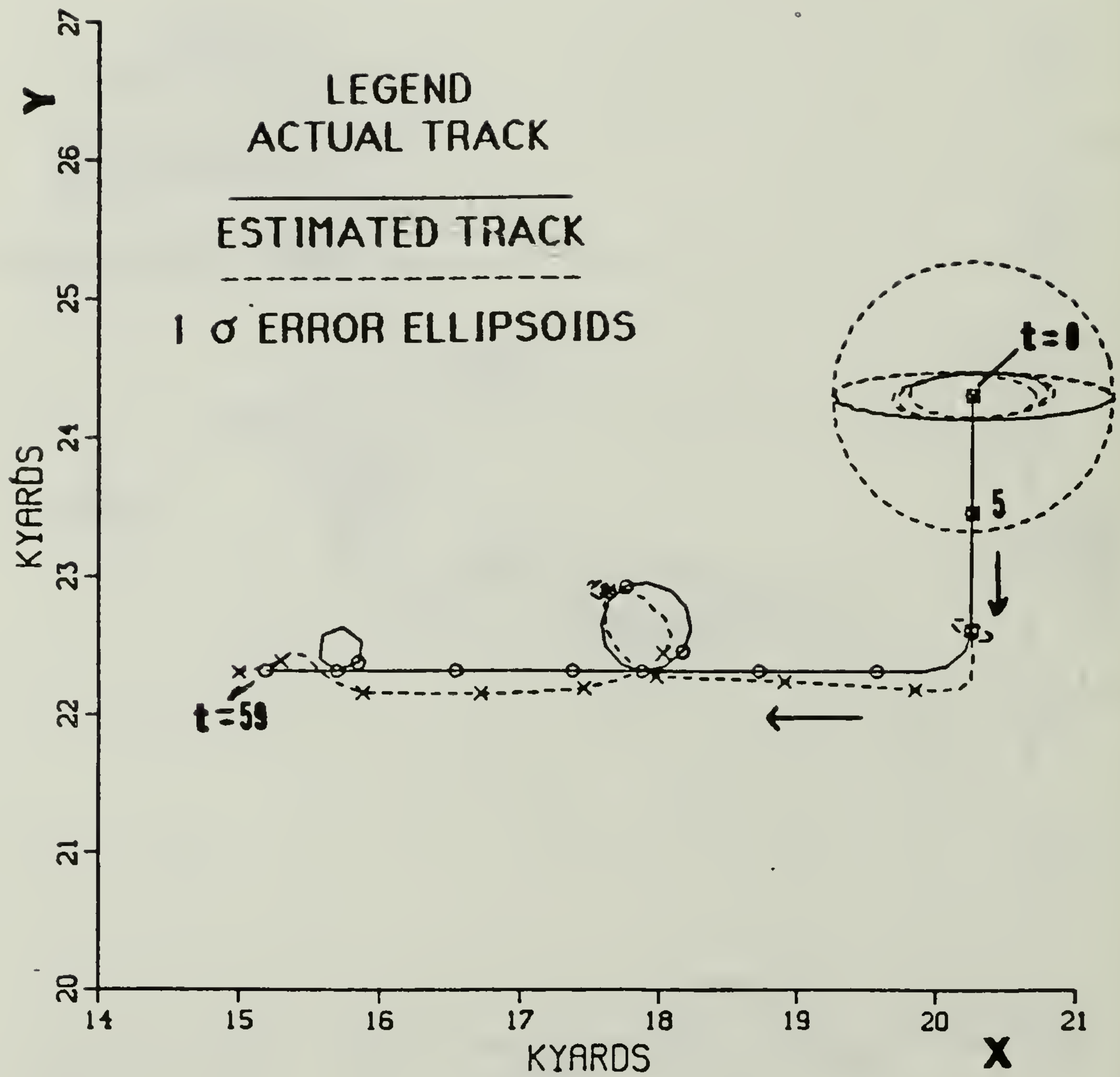

Figure 6-2. Scenario 1-Simulation 1: Enlarged Geographic Plot- $Q'_2(k)$ Applied

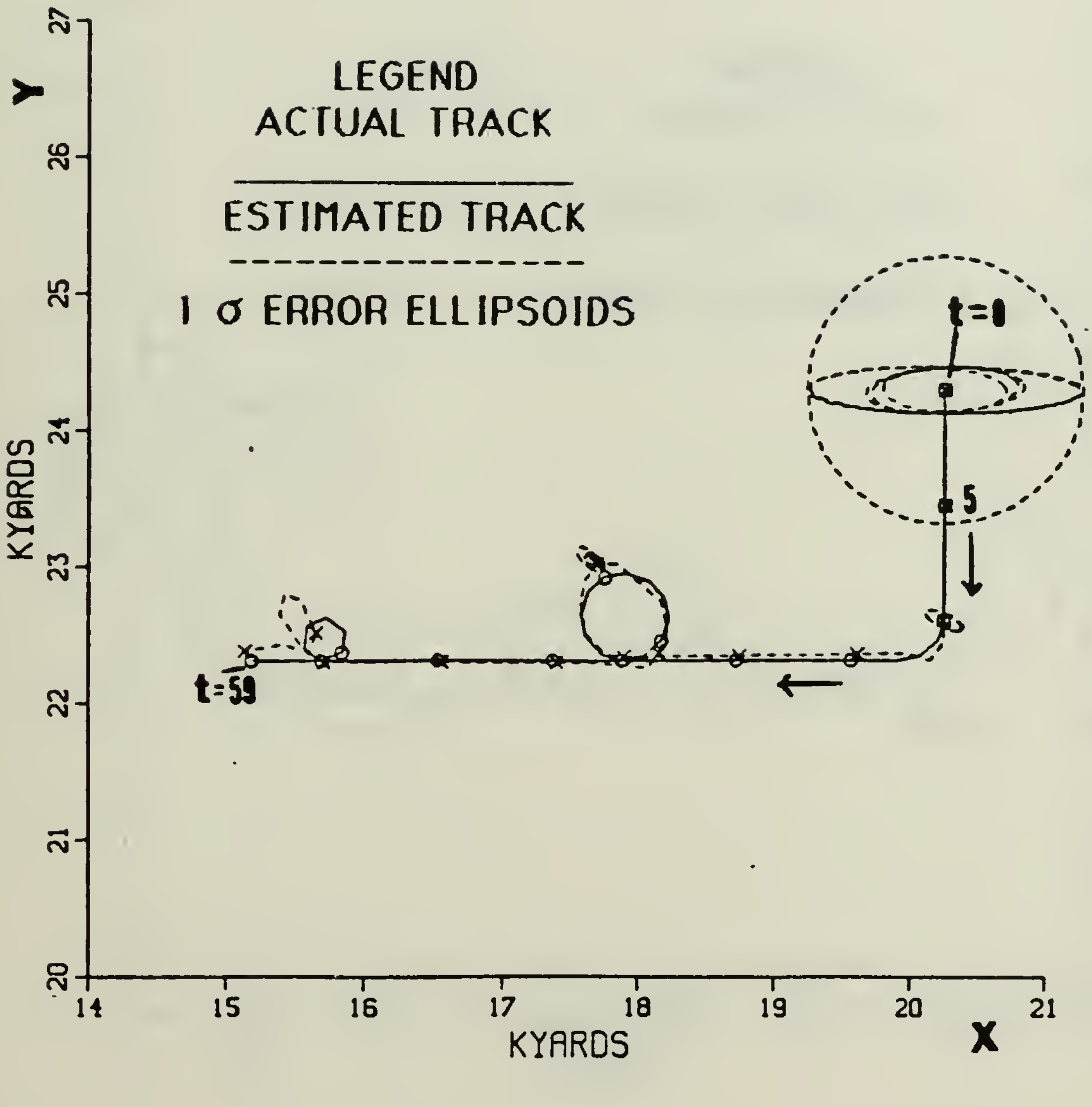

Figure 6-3. Scenario 1-Simulation 2: Enlarged Geographic Plot- $Q'_2(k)$ and Adaptive Control Applied

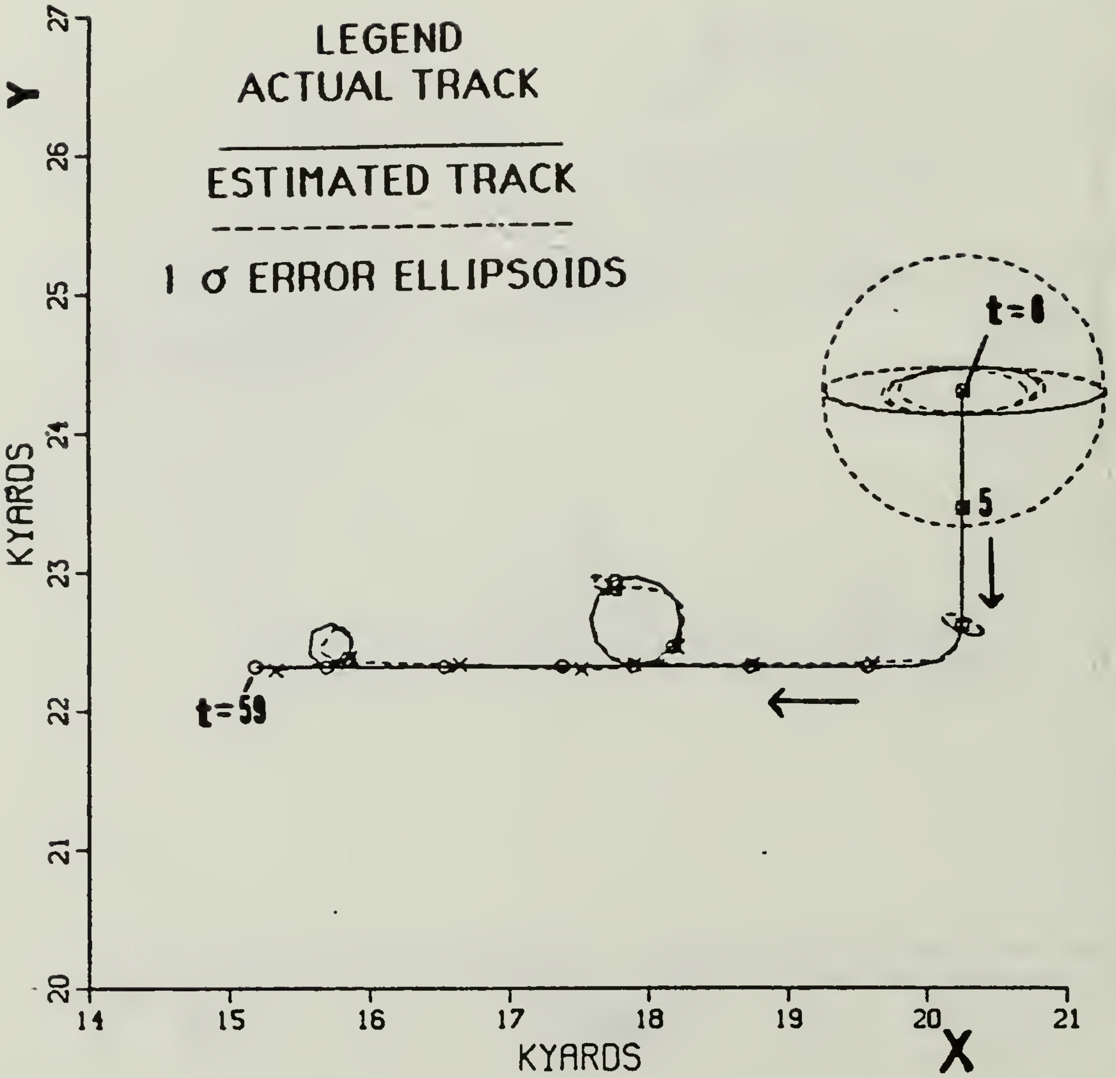

Figure 6-4. Scenario 1-Simulation 3:
Enlarged Geographic Plot- $Q'_1(k)$ Applied

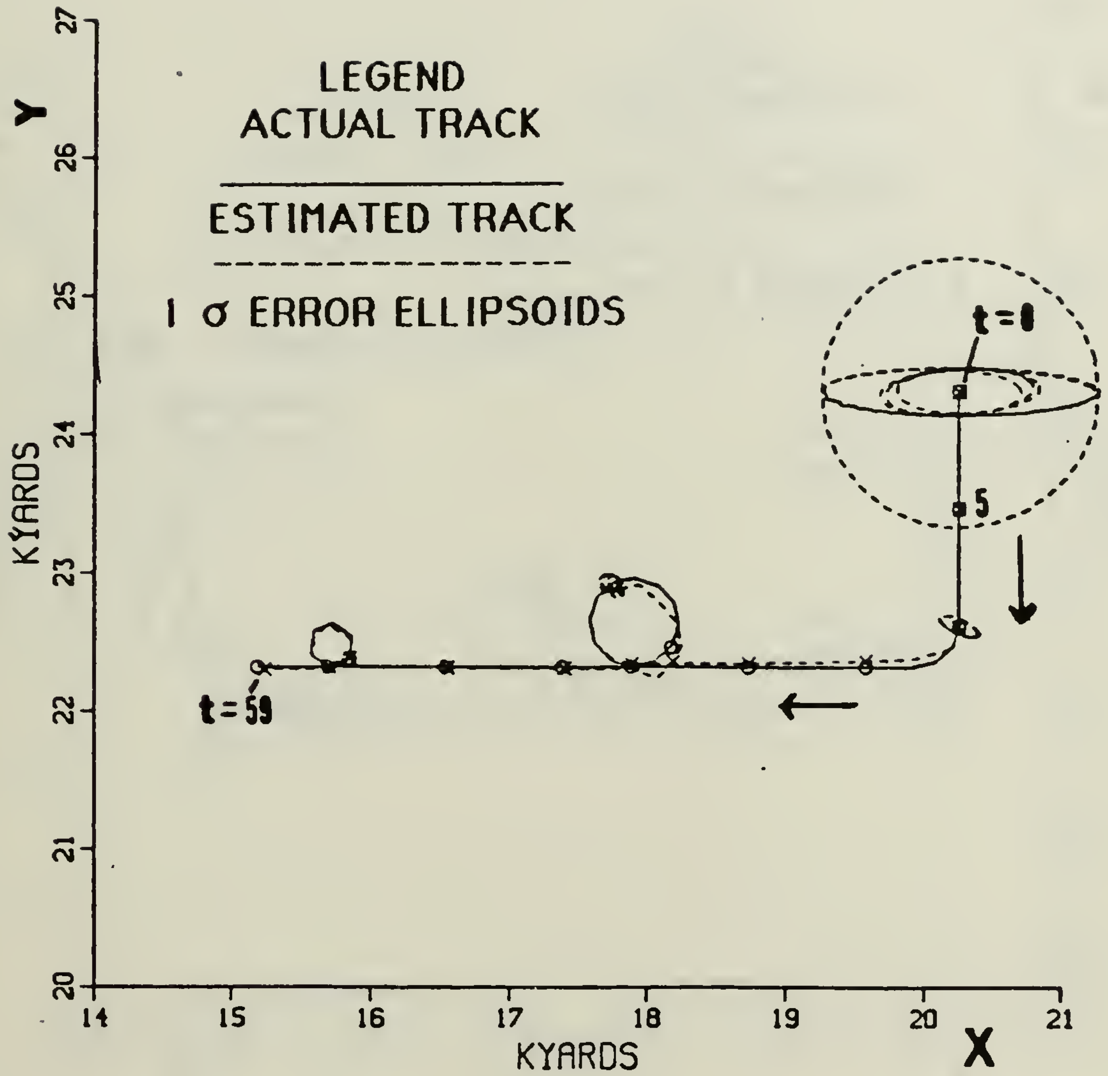

Figure 6-5. Scenario 1-Simulation 4: Enlarged Geographic Plot- $Q'_i(k)$ and Adaptive Control Applied

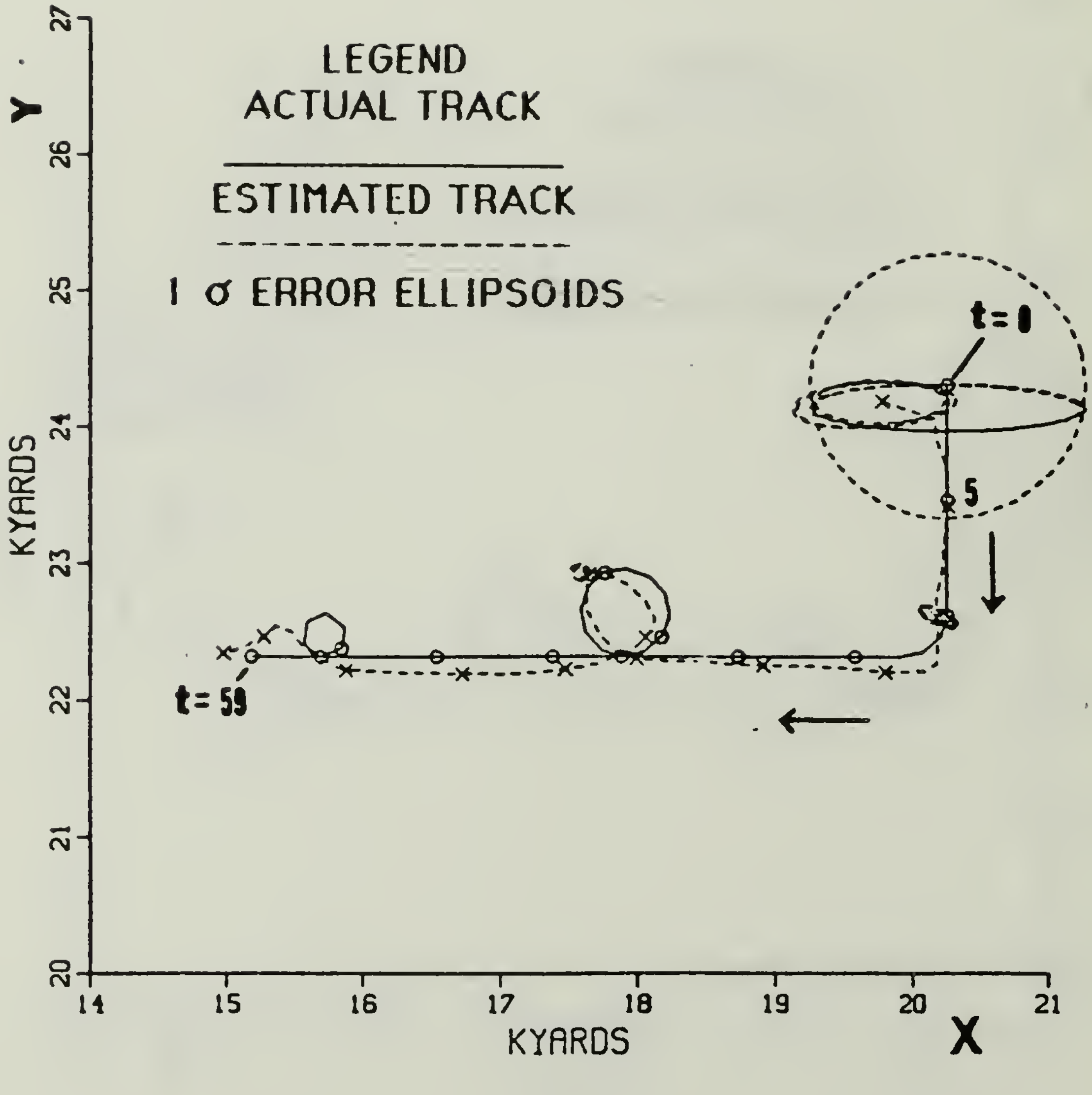

Figure 6-6. Scenario 1-Simulation 5:Enlarged Geographic Plot-Noise and $Q'_2(k)$ Applied

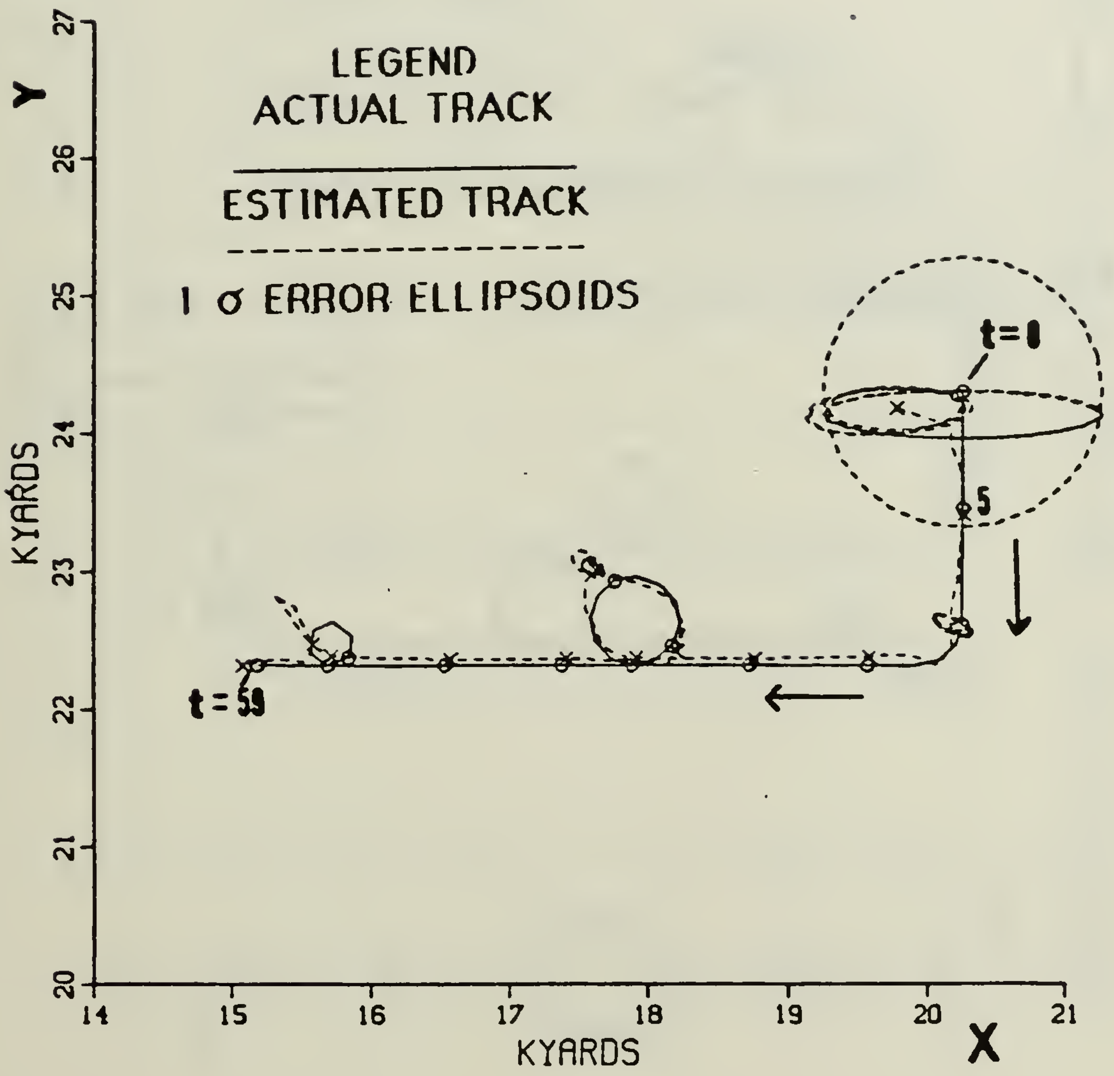

Figure 6-7. Scenario 1-Simulation 6: Enlarged Geographic Plot-Noise, $Q'_2(k)$, and Adaptive Control Applied

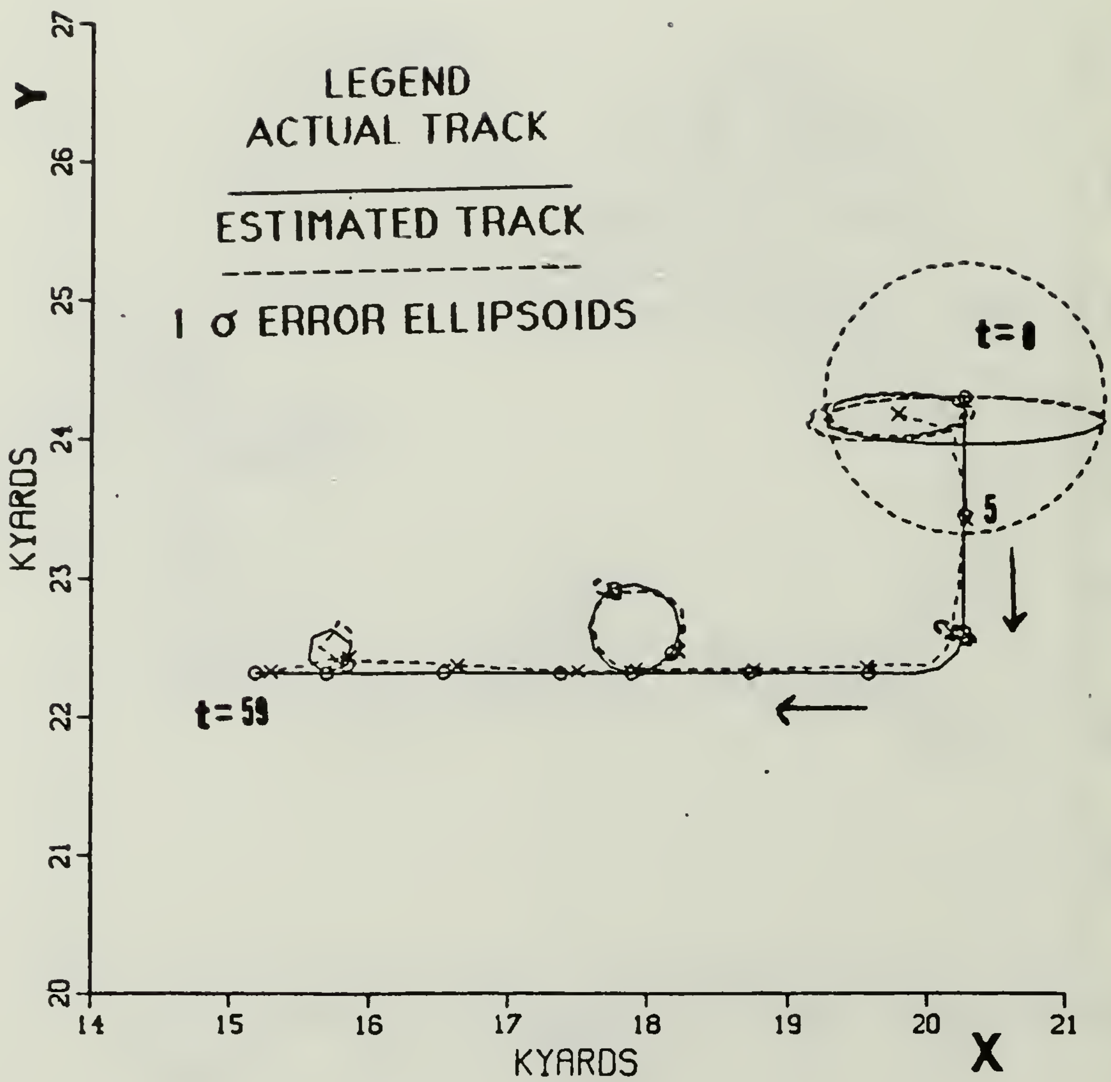

Figure 6-8. Scenario 1-Simulation 7:Enlarged Geographic Plot-Noise and $Q'_1(k)$ Applied

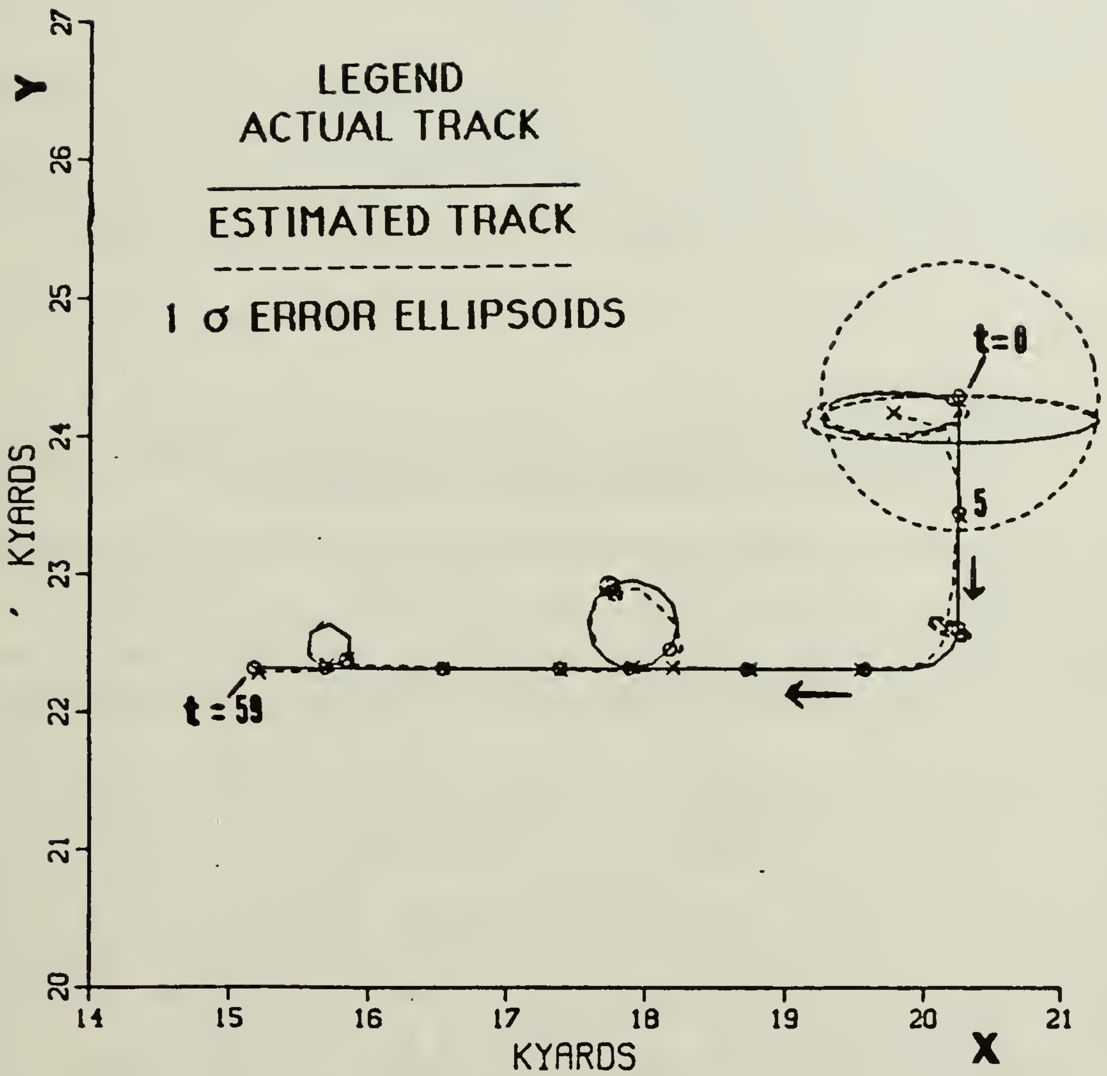

Figure 6-9. Scenario 1-Simulation 8:Enlarged Geographic Plot-Noise, $Q'_1(k)$, and Adaptive Control Applied.

1. Noise-free Simulations

As indicated in Table (6-1), measurement noise is not applied to the simulations 1-4, Figures (6-2)-(6-5) respectively. For each of the noise-free simulations, Table (6-2) lists the maximum position error and time of the error for each of the target maneuvers. The left hand column of Table (6-2) is taken from Table (5-1). As indicated from the a priori information, the initial estimated state vector $\hat{\underline{x}}(0|-1)$ values are the same as the actual state vector values. All four simulations reconstructed the target's track to within 2.5 ft. until the first maneuver. Each of the simulations will be discussed in the following paragraphs.

Table 6-2. SCENARIO 1: NOISE-FREE SIMULATIONS-
MAXIMUM ERROR FOR EACH MANEUVER

Time	Sim. 1		Sim. 2		Sim. 3		Sim. 4	
(Mins)	Time	Max Error	Time	Max Error	Time	Max Error	Time	Max Error
0-10	10	2 ft	10	2 ft	10	2.5 ft	10	2.5 ft
11-13	13	290 yds	12	80 yds	13	59 yds	12	57 yds
14-25	15	317 yds	14	73 yds	14	53 yds	14	49 yds
26-37	32	212 yds	30	114 yds	37	69 yds	34	135 yds
38-50	47	249 yds	39	31 yds	40	137 yds	39	29 yds
51-56	55	545 yds	54	394 yds	52	127 yds	53	43 yds
57-59	57	355 yds	57	169 yds	59	150 yds	55	35 yds

According to Table (6-1), $Q'_2(k)$, the random forcing function covariance matrix, is used in Figure (6-2). The estimated track lags behind every maneuver and completely misses the small 360 deg turn.

In Figure (6-3) the adaptive control method and $Q'_2(k)$ are applied. As can be seen from Figure (6-3) and Table (6-2), the filter estimates are slightly off during the large 360 deg turn and off by as much as 394 yds in the small 360 deg turn. Table (6-3) lists the number of times the $3 \cdot \sigma_z$ adaptive gate was exceeded for each measurement. Note that only the frequency measurements cause the adaptive gate to be exceeded.

In Figure (6-4) the random forcing function $Q'_1(k)$ is applied to the filter. In this simulation the filter estimates accurately reconstructs the track until the end of the large 360 deg turn (about time t_k= 36 mins). The estimates lag slightly behind the actual track from t_k=38 mins to the end with an average error of approximately 95 yds.

Both $Q'_1(k)$ and the adaptive control method are applied to the filter in Figure (6-5). The filter estimates are slightly off (greater than 100 yds for 5 mins) during the large 360 deg turn. Table (6-4) lists the number of times the adaptive gate was exceeded for each measurement. Again only the frequency measurement adaptive gate is exceeded. Comparing Table (6-3) to Table (6-4), it can seen that by increasing the random forcing function covarinace matrix, the adaptive gate is exceeded far less. As expected this simulation reconstructs the actual track better than the Simulations 1-3.

Table 6-3. SCENARIO 1-SIMULATION 2: NUMBER OF TIMES THE ADAPTIVE GATE IS EXCEEDED FOR EACH MEASUREMENT

Time	Buoy 1		Buoy 2		Buoy 3	
(Mins)	freq meas	brg meas	freq meas	brg meas	freq meas	brg meas
11					1	
12	2				1	
26			2			
27			1		1	
28					1	
30	1				1	
31	2				1	
33			1		1	
34			1		1	
35					1	
36	1				1	
37	1				1	
51	1		2			
52					1	
53	1		1			
54	2		1			
55	2					
56			2			

Table 6-4. SCENARIO 1-SIMULTATION 4: NUMBER OF TIMES THE ADAPTIVE GATE IS EXCEEDED FOR EACH MEASUREMENT

Time	Buoy 1		Buoy 2		Buoy 3	
(Mins)	freq meas	brg meas	freq meas	brg meas	freq meas	brg meas
11					1	
13	1					
27	1					
28					1	
29					1	
33			1			
34			1			
36					1	
51	1					
52	1					
53			1			
54	1					
55	1					
56			1			

2. <u>Noisy Simulations</u>

As indicated in Table (6-1), measurement noise is applied to simulations 5-8, Figures (6-6)-(6-9) respectively. From Table (5-2) set 1 frequency measurement standard deviations and from Table (5-3) set 1 bearing measurement standard deviations are applied to the noise-free measurements. Like Table (6-2) for the noise-free simulations, Table (6-5) lists the maximum position error for each maneuver in Simulations 5-8. The a priori information is the same as the noise-free simulations.

Table 6-5. SCENARIO 1 SIMULATIONS WITH NOISE
MAXIMUM ERROR FOR EACH MANEUVER

Time	Sim. 5		Sim. 6		Sim. 7		Sim. 8	
(Mins)	Time	Max Error	Time	Max Error	Time	Max Error	Time	Max Error
0-10	0	494 yds	0	494 yds	0	494 yds	0	494 yds
11-13	13	203 yds	13	74 yds	11	82 yds	11	82 yds
14-25	15	249 yds	14	77 yds	14	77 yds	14	52 yds
26-37	32	183 yds	30	157 yds	37	90 yds	34	150 yds
38-50	45	226 yds	49	77 yds	41	131 yds	47	32 yds
51-56	55	574 yds	53	477 yds	52	161 yds	53	60 yds
57-59	57	388 yds	57	191 yds	59	113 yds	57	53 yds

In all four simulations, the first estimate $\underline{\hat{x}}(0|0)$ is 494 yds southwest of the actual starting position. Examining the error ellipsoids gives us some insight to this error. The center of error ellipsoid is the measurement 's estimated position. The large dashed circle error ellipsoid is due to DIFAR 1 frequency measurement. Its estimated position is the actual starting position. DIFAR 1 bearing measurement contributes the large horizontal ellipse. Its estimated position is approximately 180 yds south of the actual staring position. DIFAR 2 bearing measurement is the major contributor to the estimated position shifting to the west of the target's position. While, DIFAR 3 estimated position is a little to west of DIFAR 2 estimated position. The shape of the error ellipsoids indicate that the greatest error is along the bearing measurement from DIFAR 1. In all four simulations, the filter's estimated position is less than 50 yds from the actual position within 4 mins.

Simulation 5, Figure (6-6) is the same as Simulation 1 except noise is applied to the measurements. Comparing Simulation 5 errors in Table (6-5) to Simulation 1 errors in Table (6-2), it can be seen that the only major difference is in the first leg, time t_k=0-10. As indicated above this difference is due to the noisy initial measurements.

In Figure (6-7) the adaptive control is applied to the filter. The results of Figure (6-7) are similar to that of Figure (6-3), except for the small 360 deg turn. Table (6-6) lists the number of times the adaptive gate was exceeded for each measurement.

The random forcing function covariance matrix $Q'_1(k)$ and noise is applied to the filter in Figure (6-8). The results are similar to the Figure (6-4), in that the filter's estimated position lags behind the actual position from time t_k = 38 mins to the end. The average errors in Figure (6-8) are slightly higher than in the noise-free case.

In Figure (6-9) the random forcing function covariance matrix $Q'_1(k)$, the adaptive gate and noise are applied to the filter. Like Figure (6-5), the filter estimates are slightly off during the large 360 deg starboard turn in figure (6-9). The errors are less than 70 yds except from t_k = 32 to t_k = 35 where the errors are greater than 100 yds. Table (6-7) lists the number of times the adaptive gate is exceeded for each measurement.

Table 6-6. SCENARIO 1-SIMULTAION 8:
NUMBER OF TIMES THE ADAPTIVE GATE IS
EXCEEDED FOR EACH MEASUREMENT

Time	Buoy 1		Buoy 2		Buoy 3	
(Mins)	freq meas	brg meas	freq meas	brg meas	freq meas	brg meas
11					1	
12	2				1	
26			1			
27	1		2			
28					1	
29					2	
30	1					
31	2					
32			1			
33	1		1			
34			1		1	
35					1	
36	1				1	
37	1					
51	1		2			
52					1	
53	1				1	
54	2		1			
55	2					
56			2			

As in the noise-free simulations, Simulation 8 reconstructs the target's track better than Simulations 5-7. Simulation 8's configuration is used in the rest of the simulations presented in this text.

Table 6-7 SCENARIO 1-SIMULTAION 8: NUMBER OF TIMES THE ADAPTIVE GATE IS EXCEEDED FOR EACH MEASUREMENT

Time	Buoy 1		Buoy 2		Buoy 3	
(Mins)	freq meas	brg meas	freq meas	brg meas	freq meas	brg meas
12	1					
13	1					
27	1					
28					1	
29					1	
31					1	
33			1			
34			1			
35					1	
51	1					
52	1					
53			1			
54	1					
55	1					
56			1			

3. Simulation 8 in Detail

The following discussion and figures will be a detailed examination of simulation 8, figure (6-9). The frequency and bearing measurement predicted residuals are illustrated along with its adaptive gate in figures (6-10) and (6-11) respectively. As shown the dashed line is the predicted residual. The solid line is the $3 \cdot \sigma_z$ adaptive gate. The small triangles represent the number of times the predicted residual exceeded the adaptive gate at that particular time. Figure (6-10) agrees with Table (6-7) information. We can see in Figure (6-10) that during the 90 deg turn at time t_k = 12 and t_k = 13 the adaptive gate increases to admit the frequency measurement from buoy 1 to the filter. Similarly, the adaptive gate increases during the large 360 deg and small 360 deg turns.

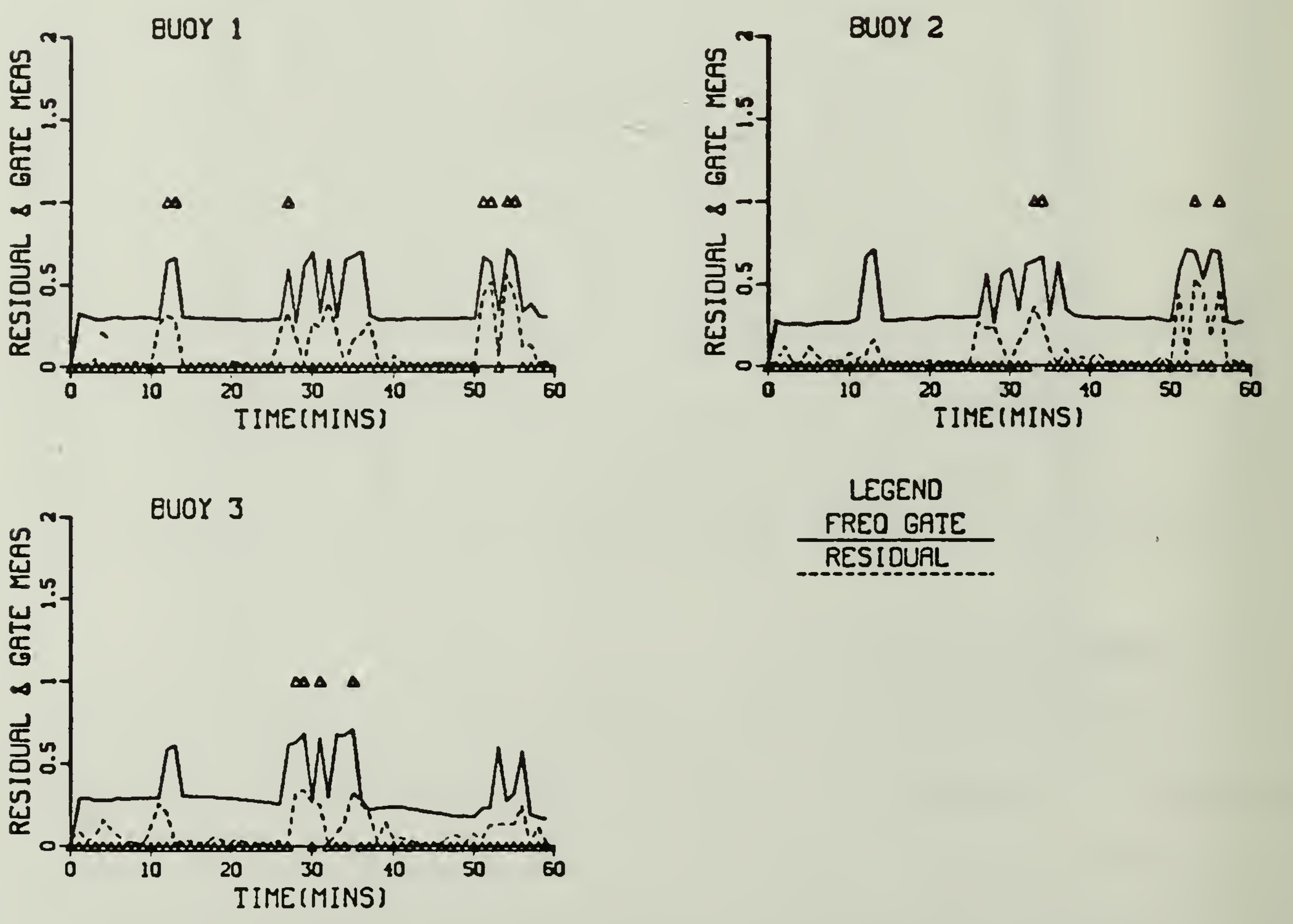

Figure 6-10. Scenario 1-Simulation 8:
Frequency Measurement-Predicted Residual and Adaptive Gate

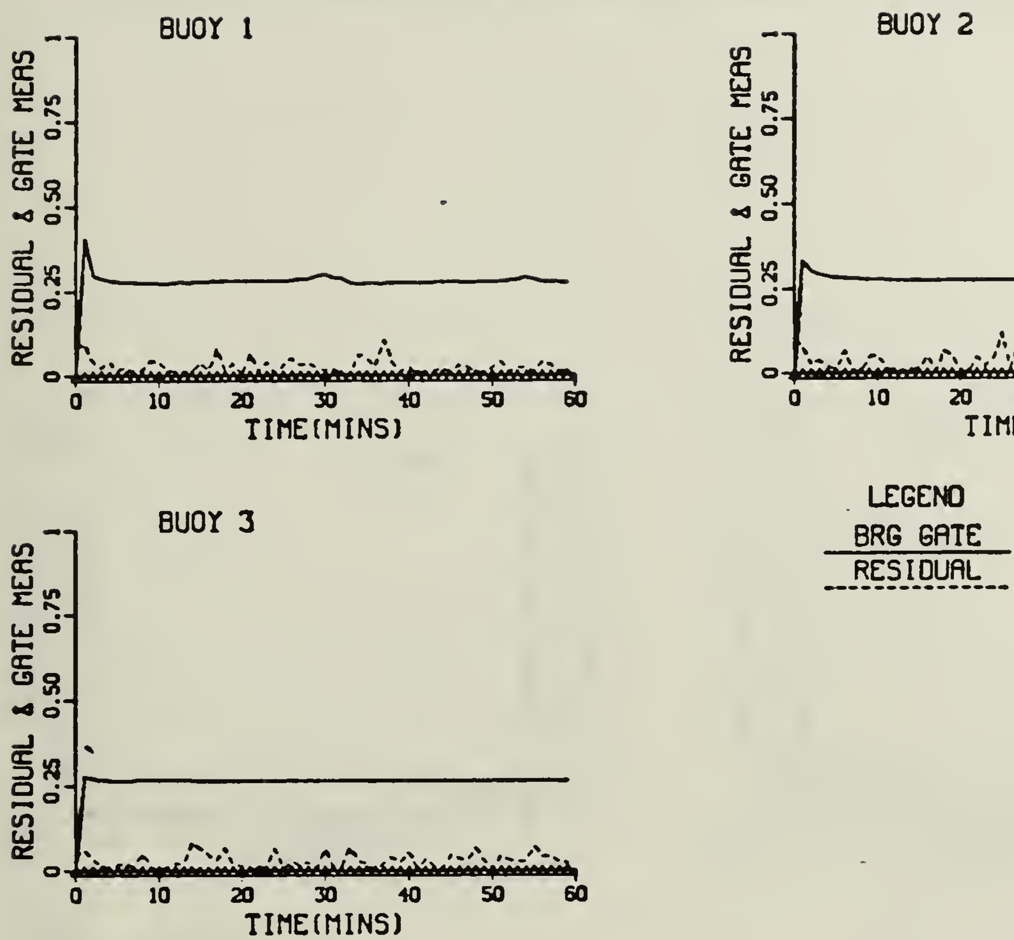

Figure 6-11. Scenario 1-Simulation 8:
Bearing Measurement-Predicted Residual and Adaptive Gate

Figures (6-12) and (6-13) show the variances of the position components, x component is $P_{11}(k|k)$ and y component is $P_{33}(k|k)$, from the error covariance matrix for the frequency and bearing measurement respectively. Both figures show the x and y component variances decreasing rapidly. This is due to the a priori position information, σ_p

equals 0.5 nm (1012 yds). Hence the variance σ_p^2 is approximately 10^6 yds^2. Figures (6-14) and (6-15) are the result of expanding Figures (6-12) and (6-13). In these graphs we can see how the position variances change during each of the target's maneuvers. Note that the frequency and bearing measurement position variances are nearly the same for each buoy.

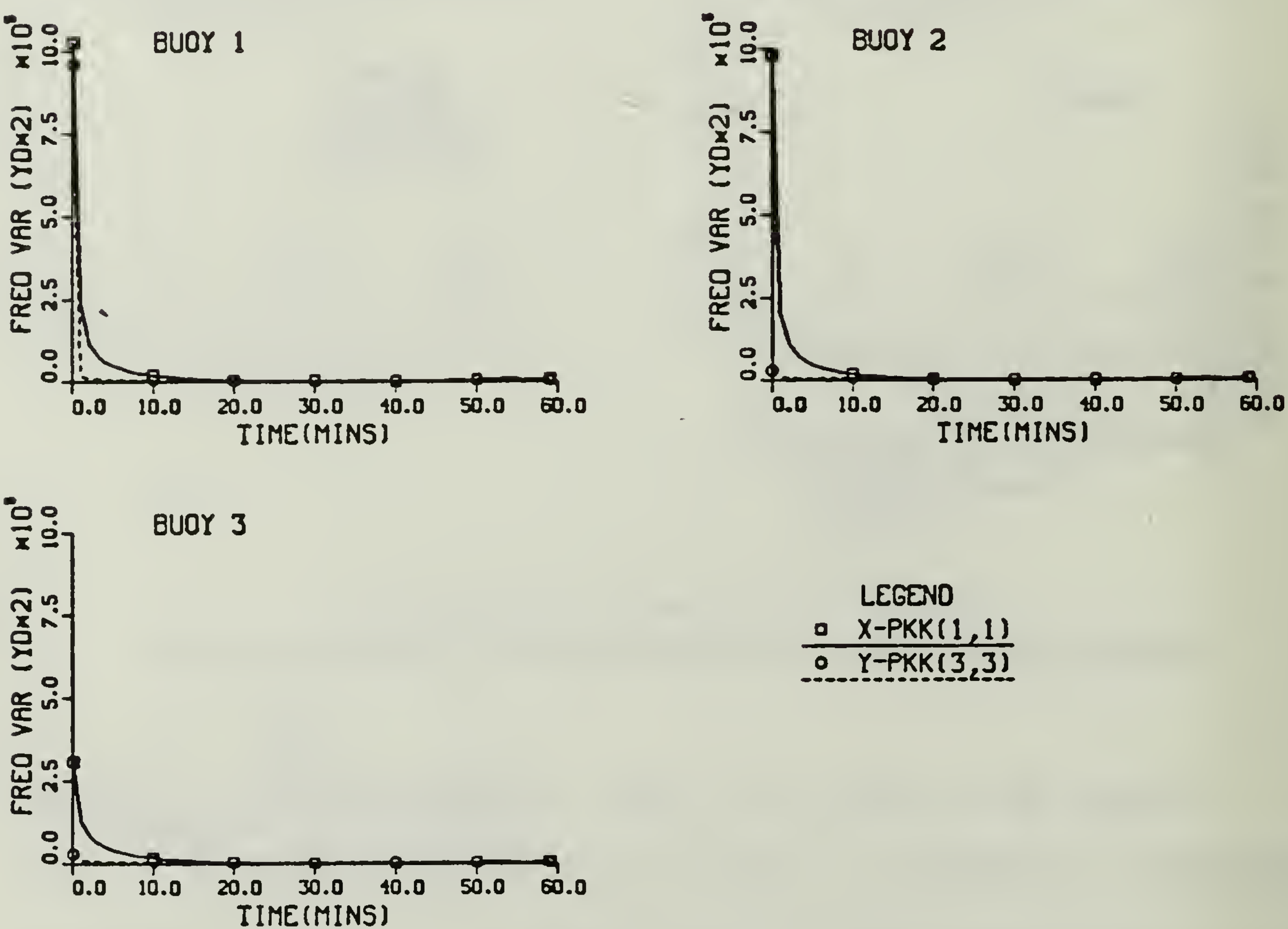

Figure 6-12. Scenario 1-Simulation 8:
Frequency Measurement-Position Variances

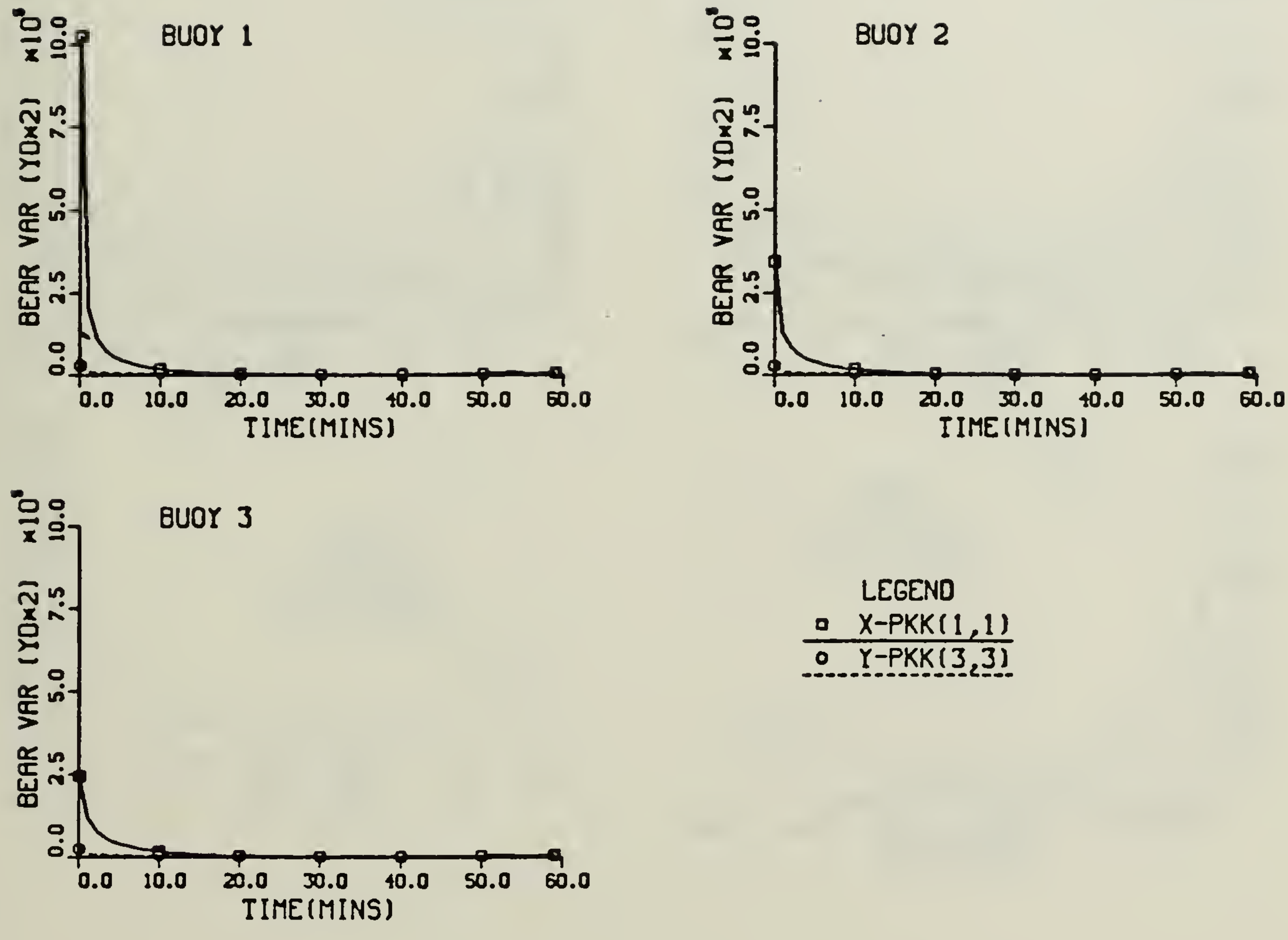

Figure 6-13. Scenario 1-Simulation 8:
Bearing Measurement-Position Variances

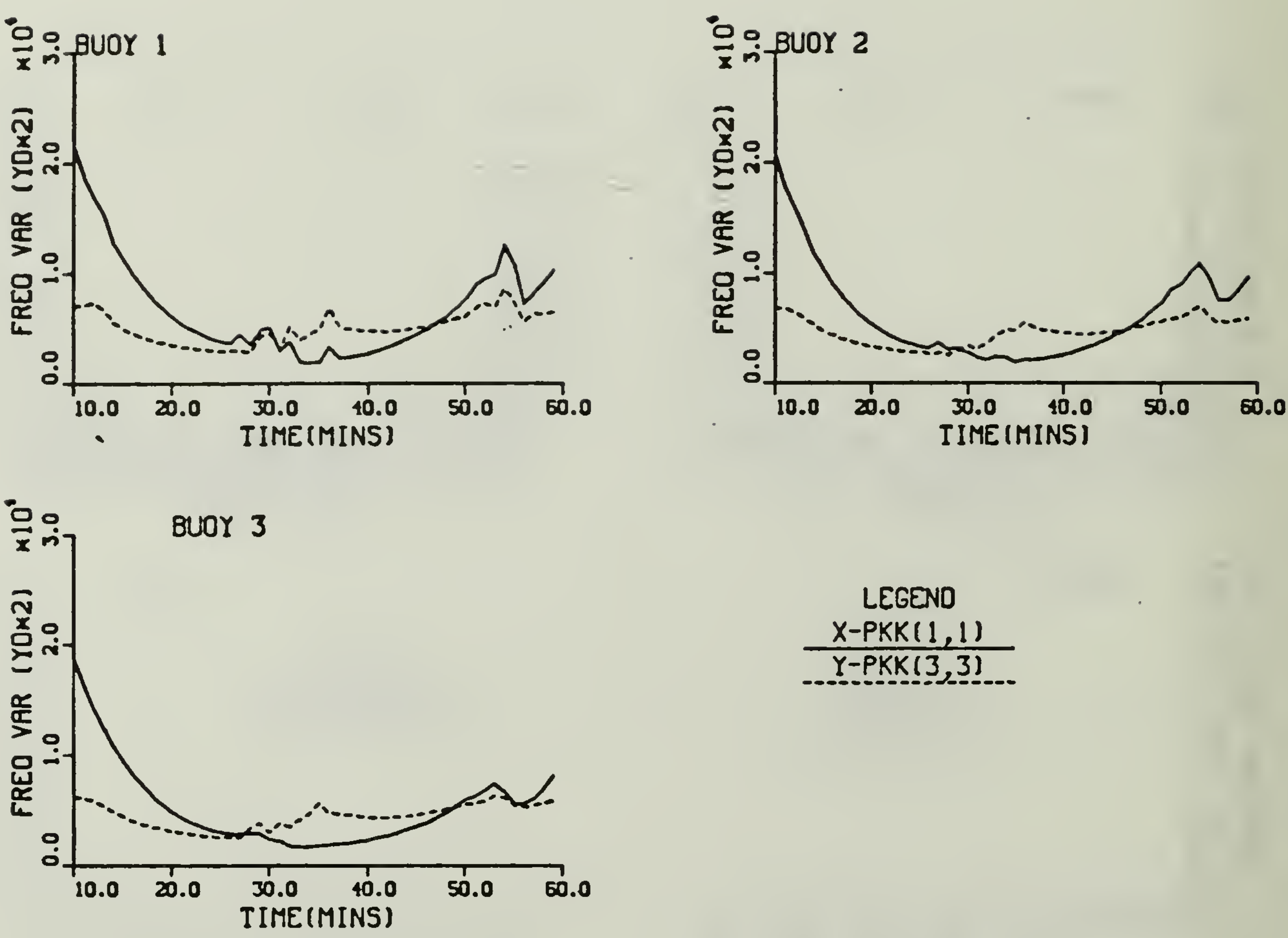

Figure 6-14. Scenario 1-Simulation 8: Expanded View of the Frequency Measurement Position Variances

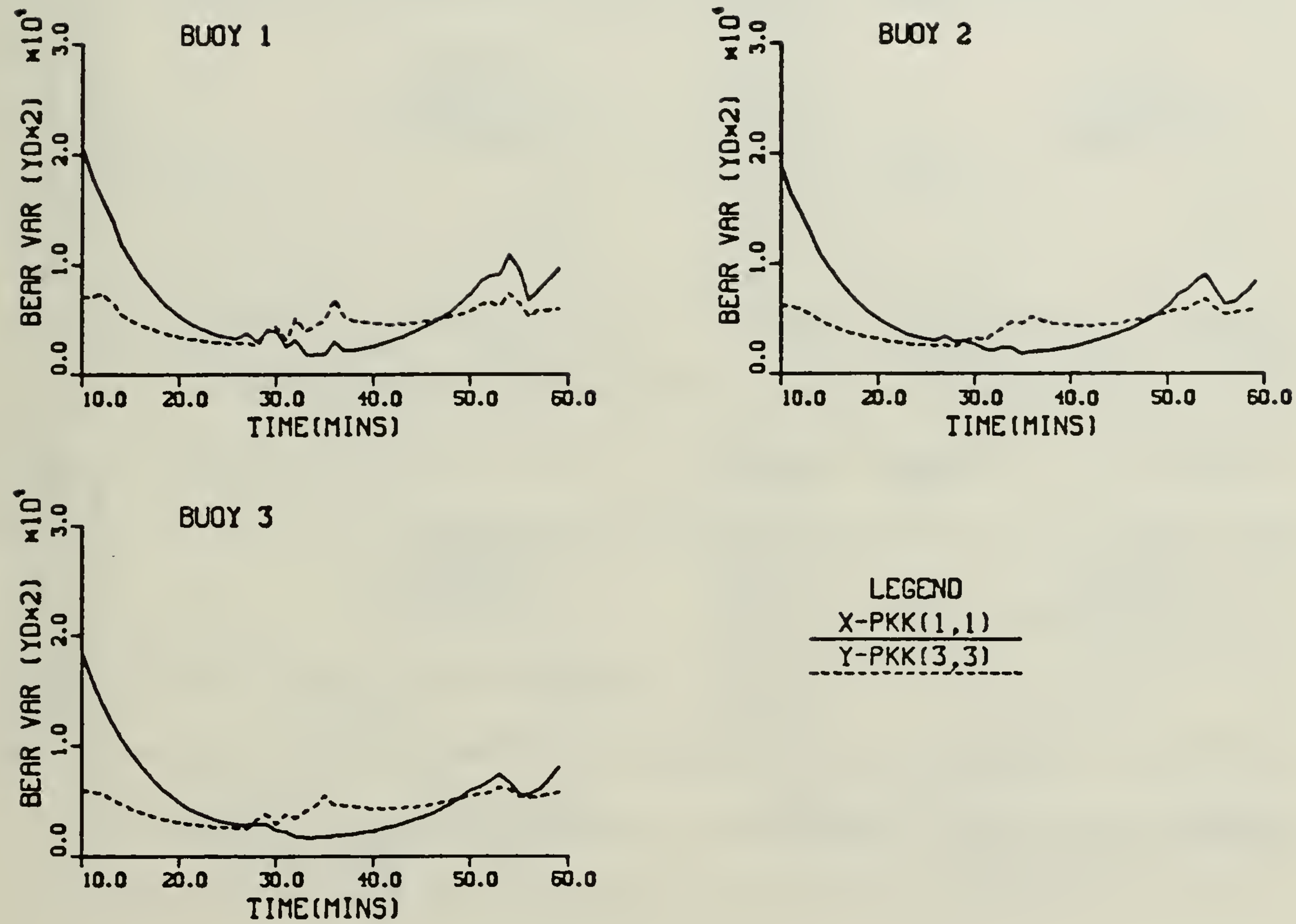

Figure 6-15. Scenario 1-Simulation 8: Expanded View of the Bearing Measurement Position Variances

Next we compared the Kalman filter's position variances to an experimental position variance. The experimental position variance consisted of a ten term moving window. From Helstrom, [Ref. 9:p. 219] the most convenient form for calculating the sample variance is

$$\text{var} = \frac{1}{n-1} \cdot \sum_{k=1}^{n} [\underline{x}(k|k) - \hat{\underline{x}}(k|k)]^2 \quad . \qquad (6\text{-}1)$$

In order to have a ten term moving window equation (6-1) becomes

$$\text{exp var} = \frac{1}{9} \cdot \sum_{i=k-10}^{k} [\underline{x}(k|k) - \hat{\underline{x}}(k|k)]^2 \qquad (6\text{-}2)$$

Figures (6-16) - (6-18) illustrates the Kalman position variances and the experimental position variances. The experimental variance's y component increases for all three buoys from t_k = 30 to t_k =35 mins because the y component error is greatest during this time(from t_k = 32 to t_k =35 mins the error is greater than 100 yds).

Figures (6-19) and (6-20) illustrates the change in the velocity variances, $v_x = P_{22}(k|k)$ and $v_y = P_{44}(k|k)$, for simulation 8. Note that the frequency and bearing measurement velocity variances are very similar. The velocity variances of buoy 1 increases during each of the maneuvers. Buoy 2 variances are only slightly effected by the 90 deg turn, but as the target moves toward the buoy the variances increase, especially v_x. During the large 360 deg turn v_y has a greater increase for buoy 3. This is expected due to DIFAR 3's location.

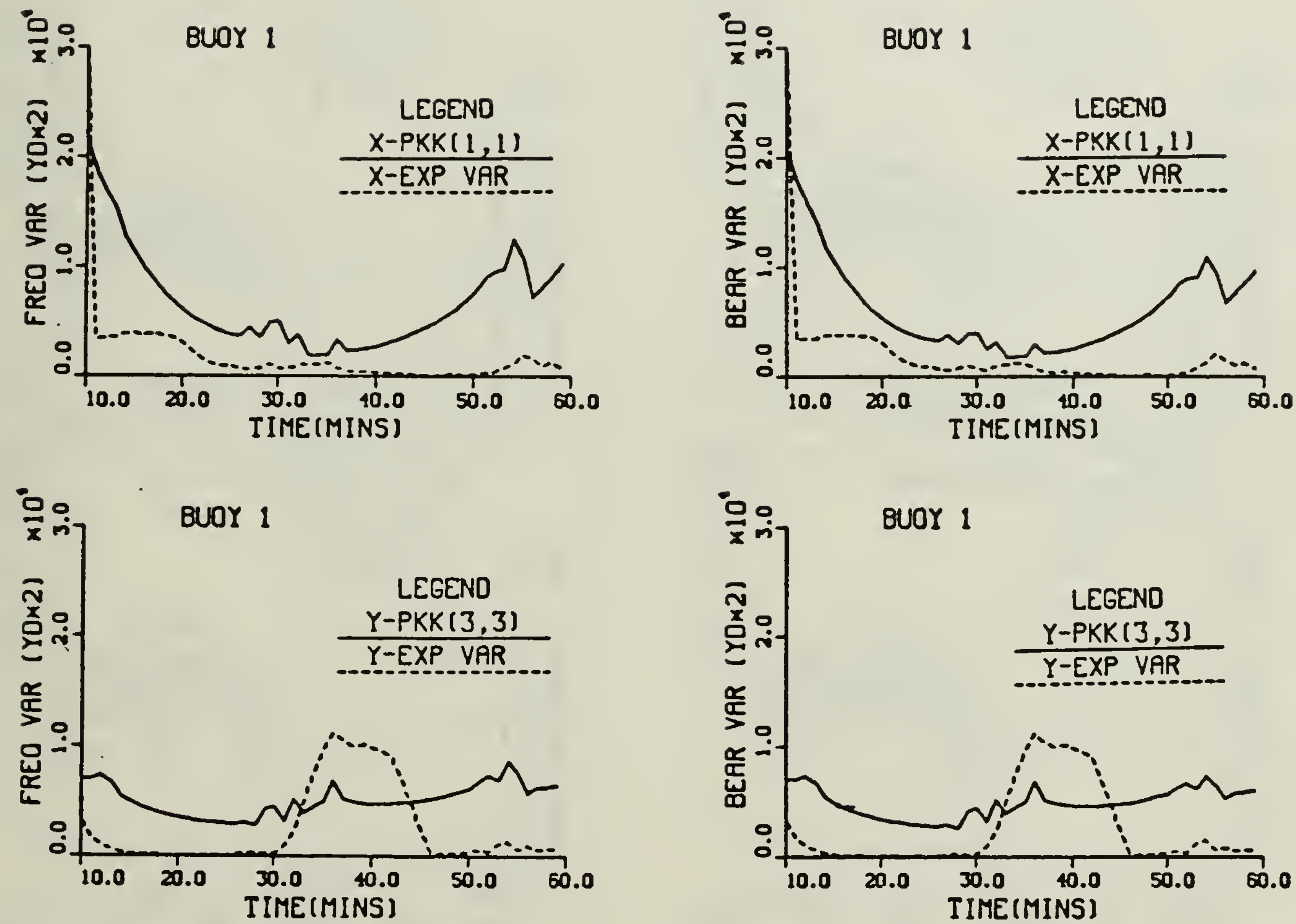

Figure 6-16 Scenario 1-Simulation 8: Buoy 1's Position Variances and Experimental Variances

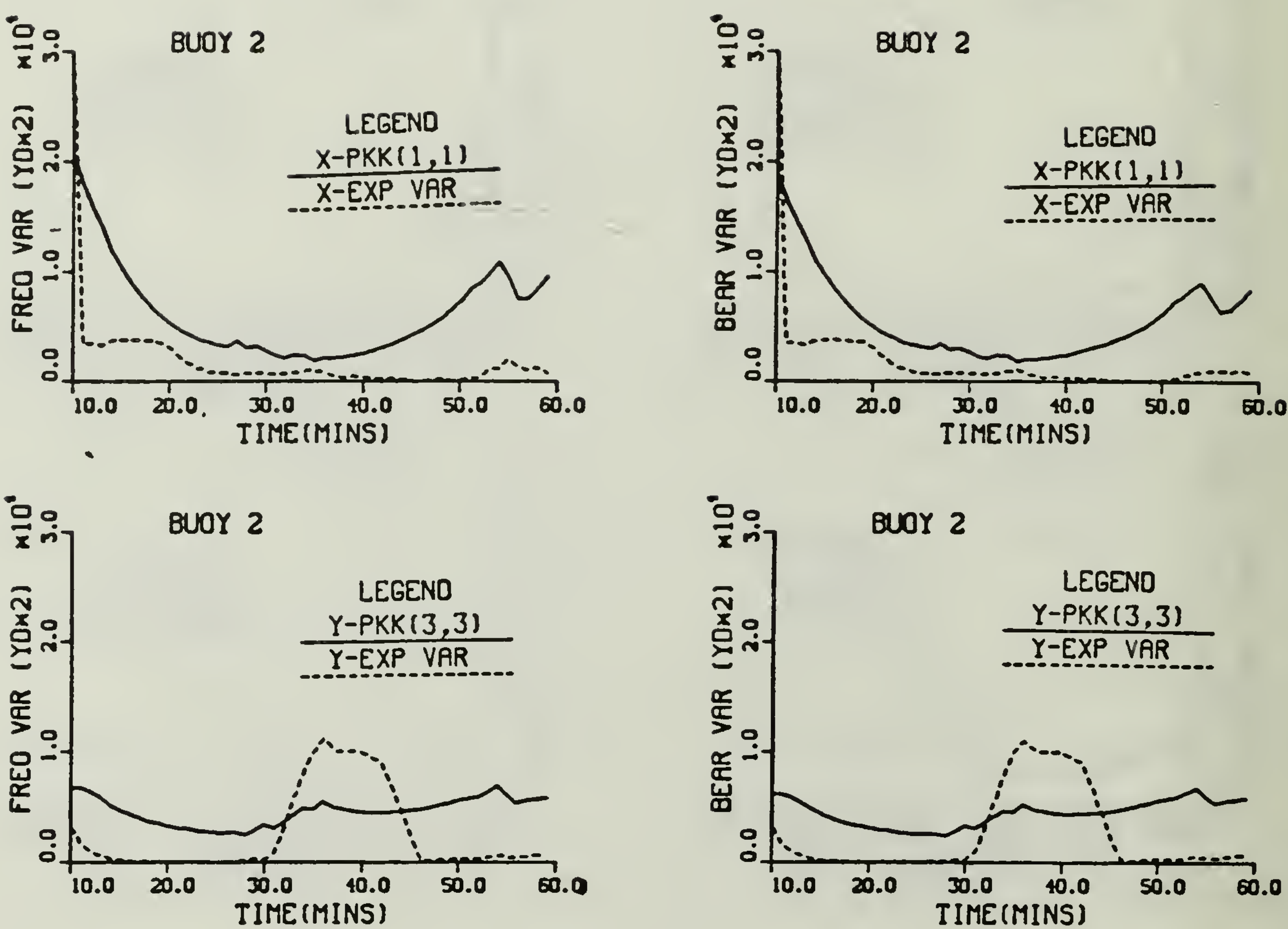

Figure 6-17. Scenario 1-Simulation 8: Buoy 2's Position Variances and Experimental Variances

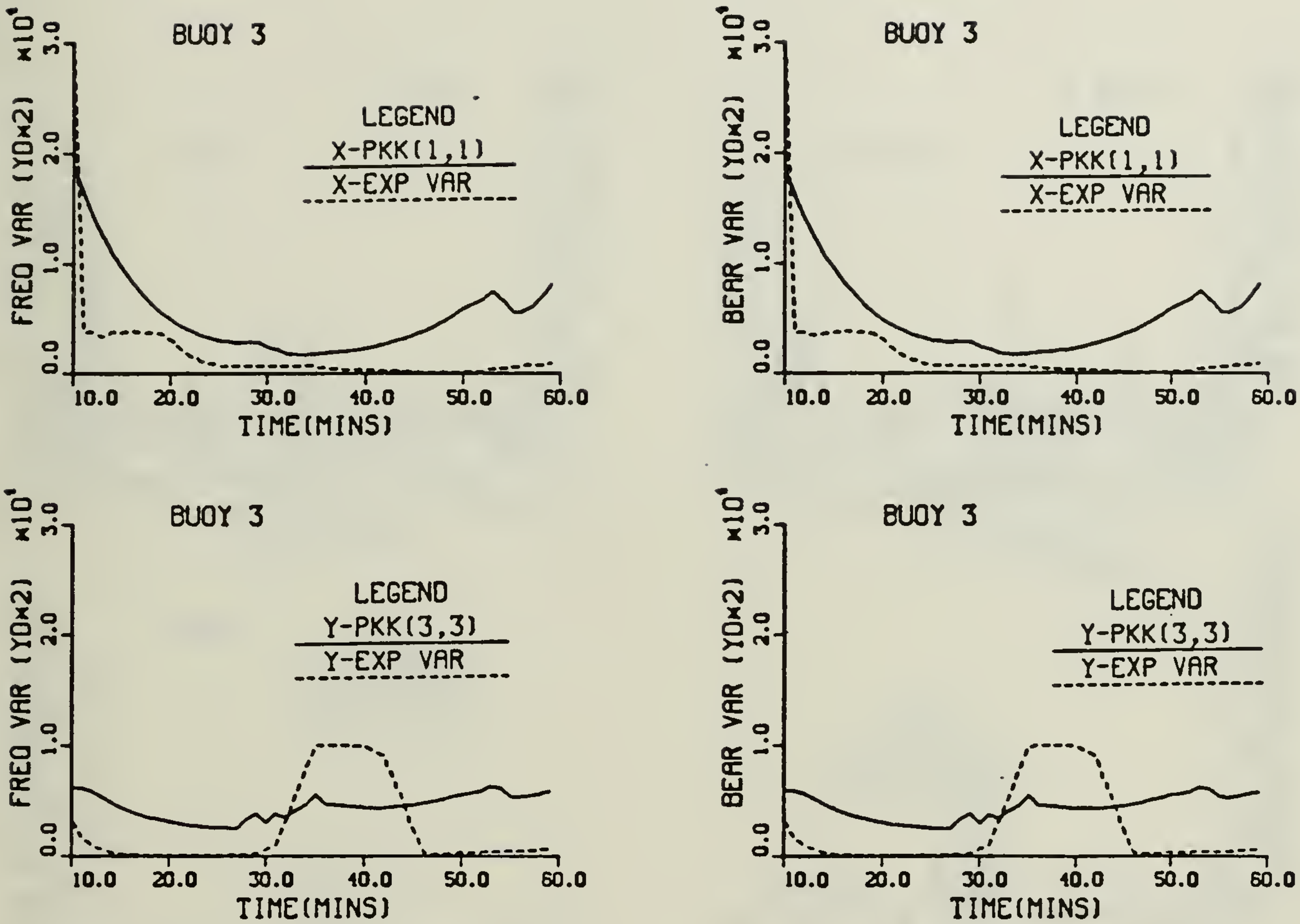

Figure 6-18. Scenario 1-Simulation 8: Buoy 3's Position Variances and Experimental Variances

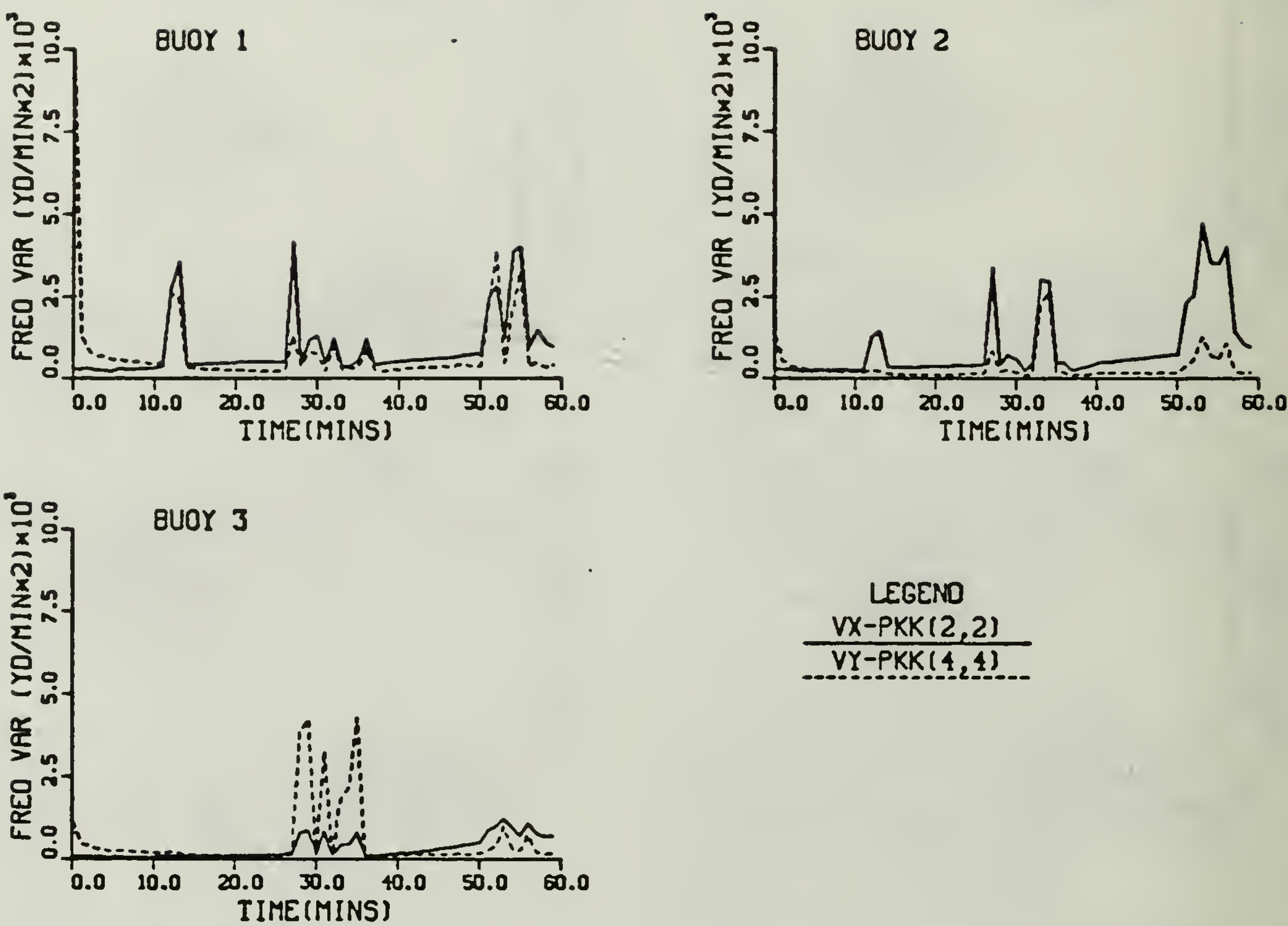

Figure 6-19. Scenario 1-Simulation 8:
Frequency Measurement-Velocity Variances

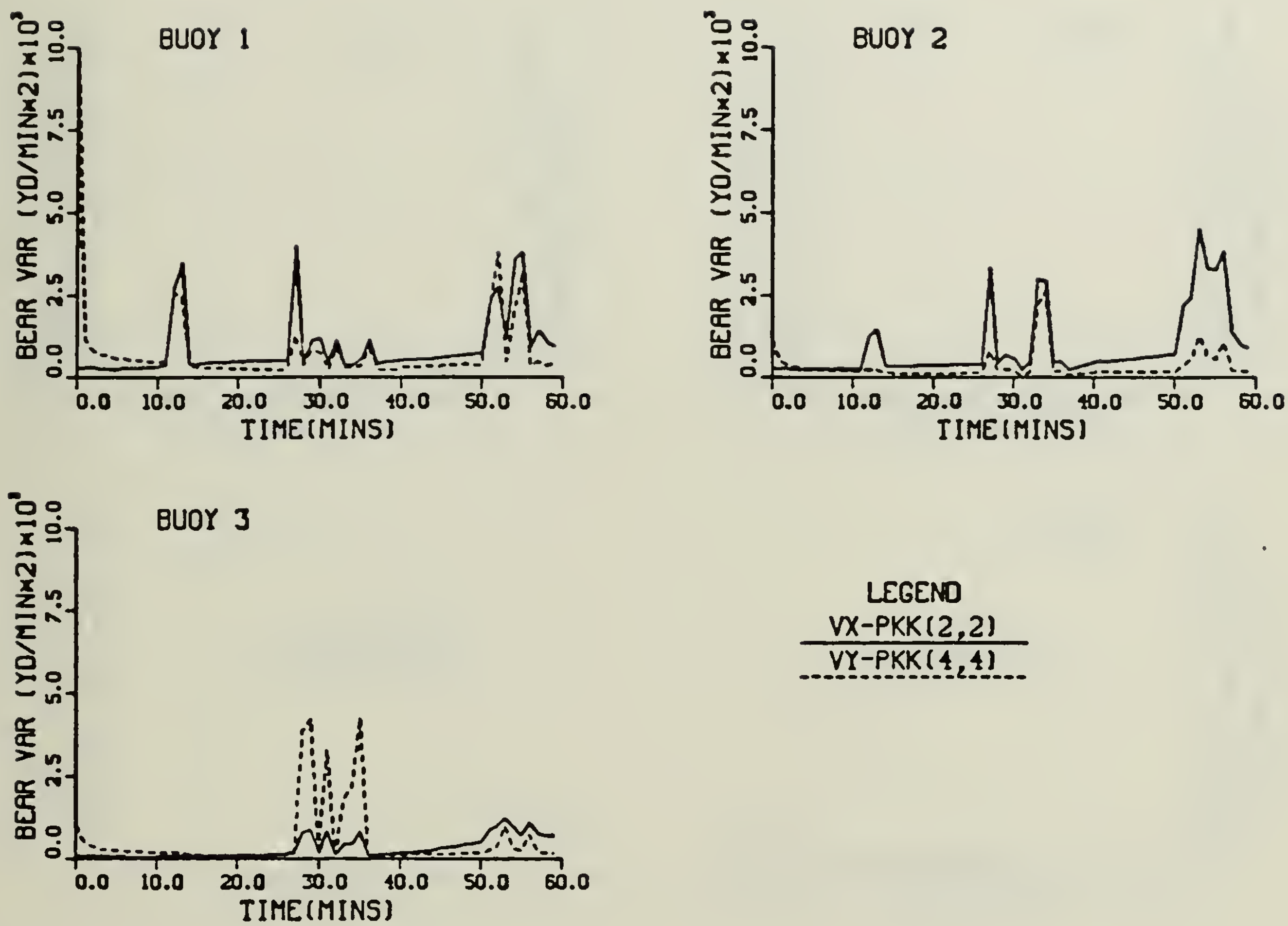

Figure 6-20. Scenario 1-Simulation 8: Bearing Measurement-Velocity Variances

The frequency variance for the frequency and bearing measurements are illustrated in Figures (6-21) and (6-22). Like the velocity variances, the frequency variance for the bearing measurement is very similar to the frequency variance for frequency measurement.

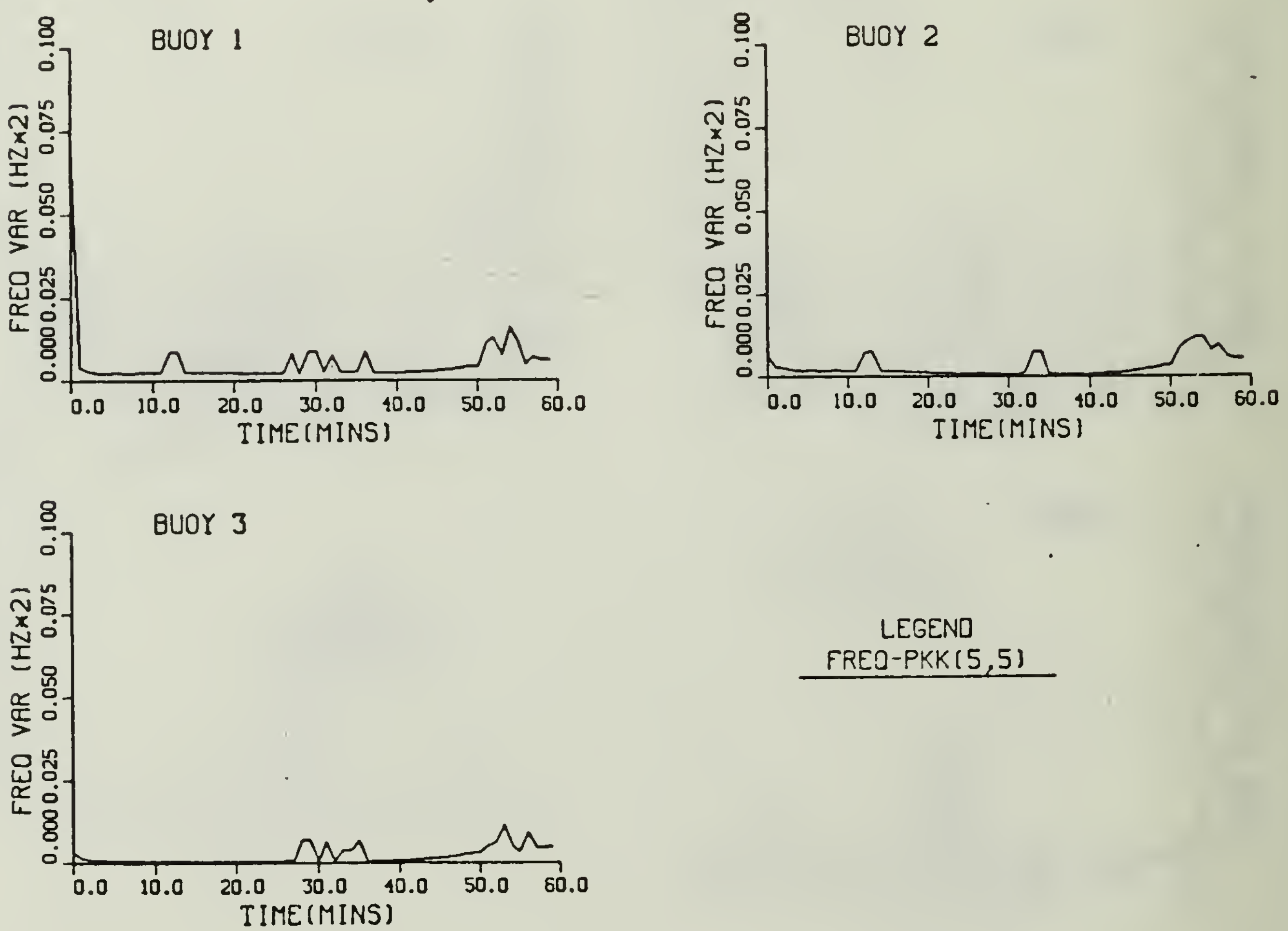

Figure 6-21 Scenario 1-Simulation 8:
Frequency Measurement-Frequency Variances

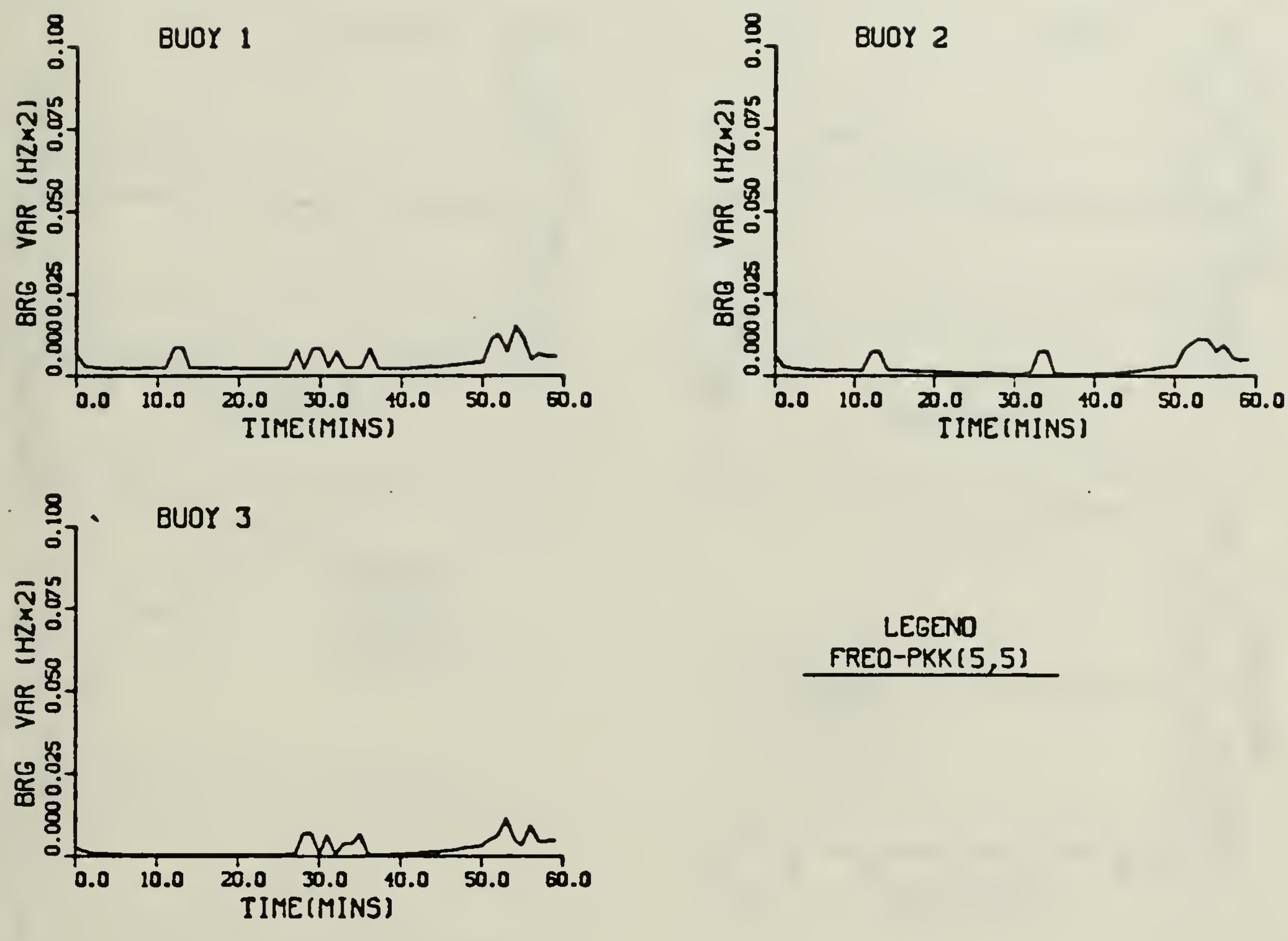

Figure 6-22. Scenario 1-Simulation 8:
Bearing Measurement-Frequency Variances

Figures (6-23)-(6-28) illustrates the position, velocity and frequency components Kalman gains for the frequency and bearing measurements. In Figures (6-23) and (6-24) the position gains decrease towards zero with small deviations for the turns. The velocity and frequency gains, Figures (6-25) - (6-28) are very erratic during the turns.

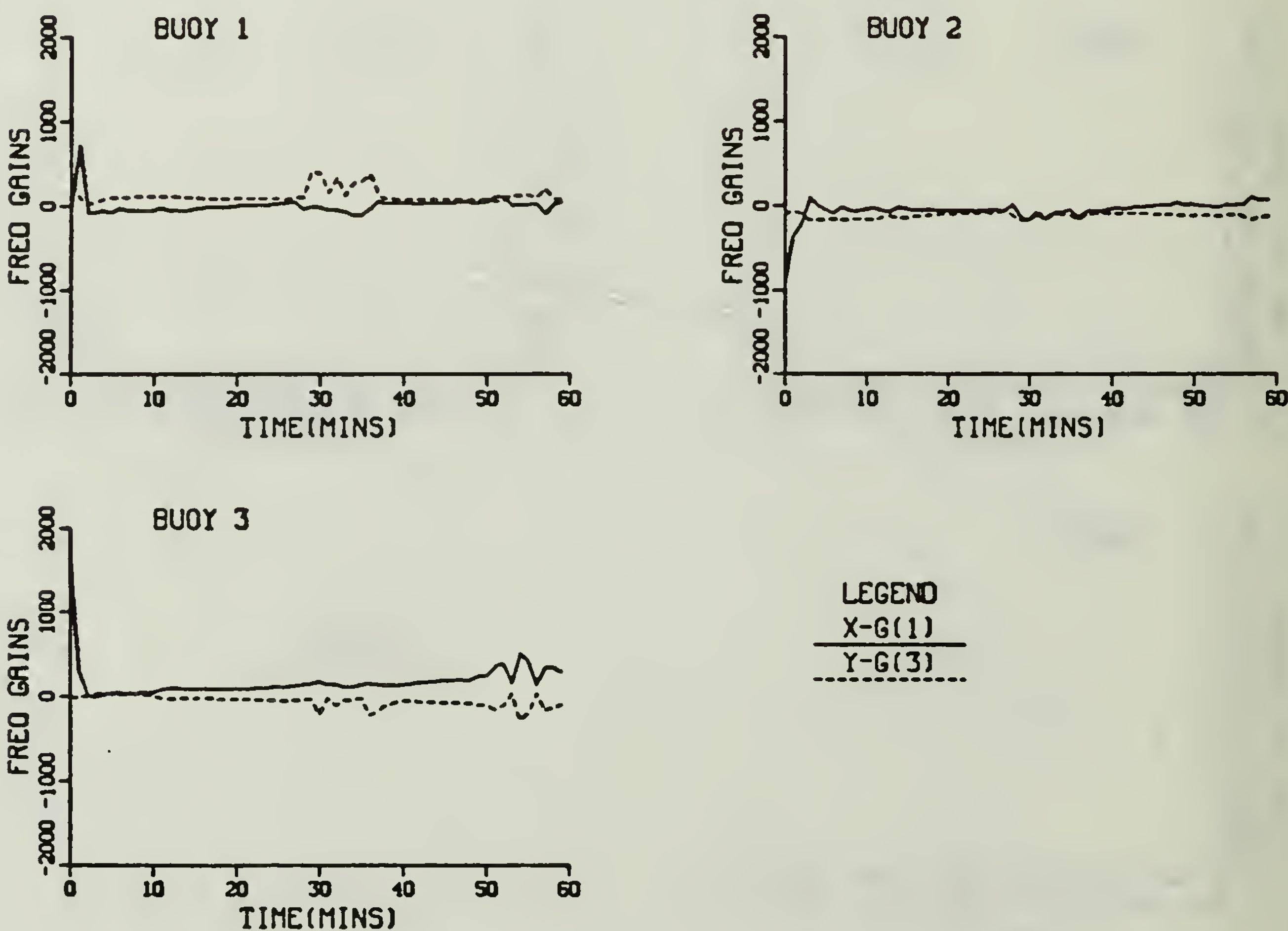

Figure 6-23. Scenario 1-Simulation 8:
Frequency Measurement - Kalman Gains for Position Components

We can now explain why the bearing measurement's variances are similar to the frequency measurement's variances. Recall the error covariance update equation (3-19) from Table 3-1 is

$$P(k|k) = (I - G(k) \cdot H(k)) \cdot P(k|k-1)$$

and that in this case $P(k|k-1)$ is really the frequency measurement's covariance of error matrix. In Subsection V.D.2 it was shown that

$$P_b(k|k-1) = P_f(k|k).$$

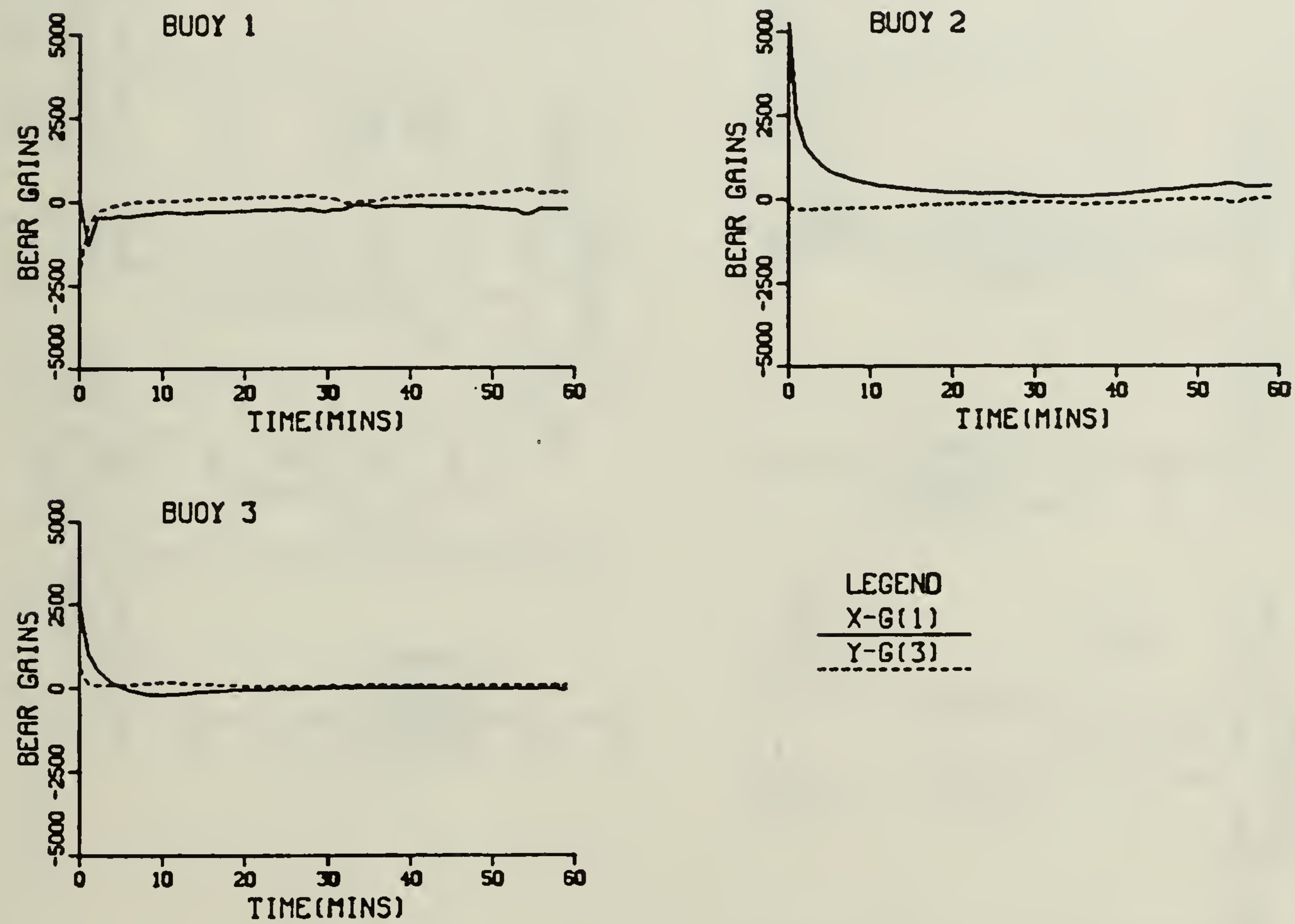

Figure 6-24. Scenario 1-Simulation 8:
Bearing Measurement - Kalman Gains for Position Components

From equations (3-46b), (3-46d), and (3-46e), the partial derivatives of the bearing measurement with respect to the velocity and frequency components are zero. Hence, only the position components of H(k) are used in the calculation of G(k)·H(k). It can be seen in figures (6-23)-(6-28), that the Kalman gains are not always small numbers. But, examining the output data indicates that the position components of H(k) are very small compared to G(k), so G(k)·H(k) is be small. Therefore the error covariance matrix for the bearing measurement P(k|k) will be approximately equal to the error covariance of the frequency measurement P(k|k-1).

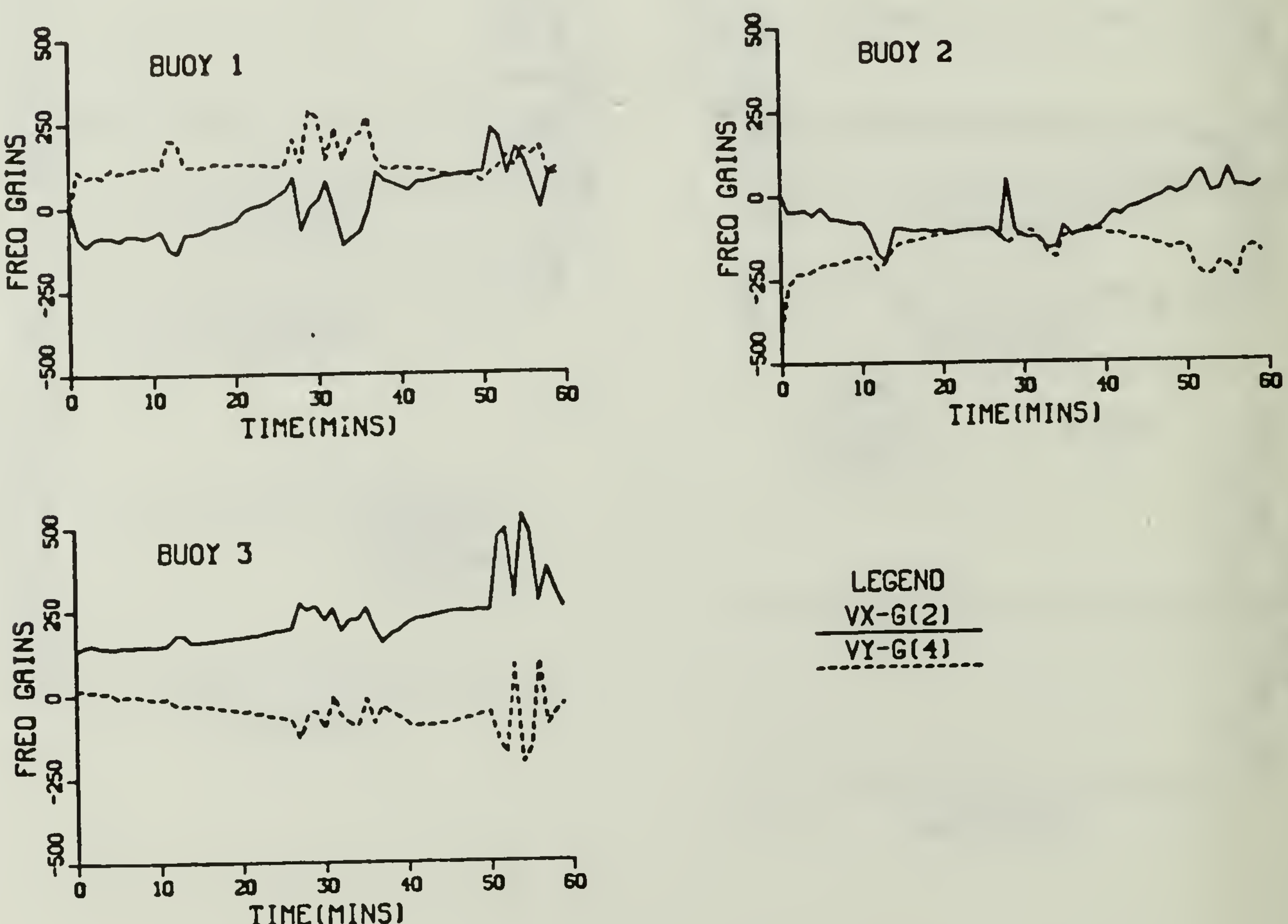

Figure 6-25. Scenario 1-Simulation 8:
Frequency Measurement - Kalman Gains for Velocity Components

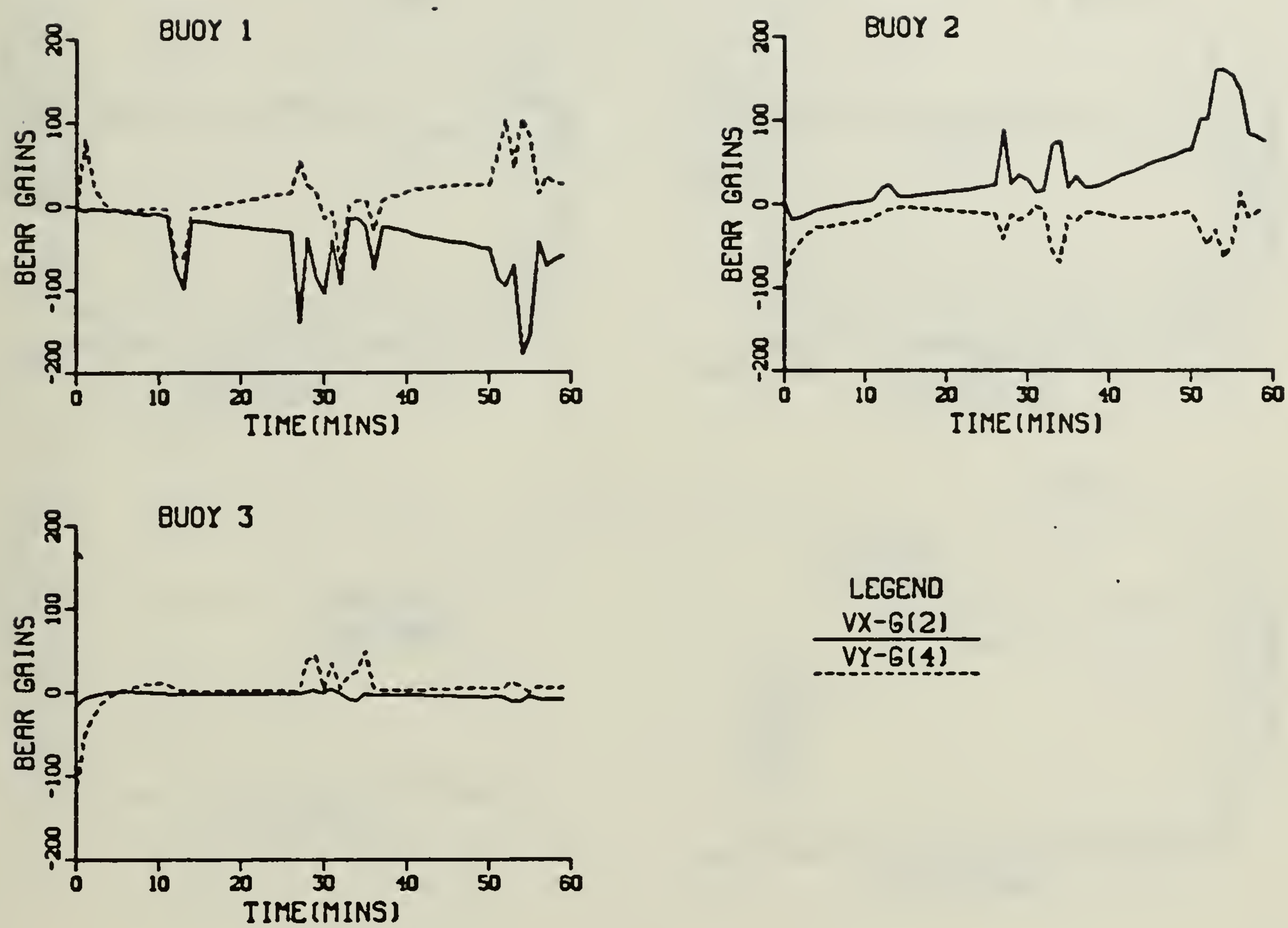

Figure 6-26. Scenario 1-Simulation 8:
Bearing Measurement - Kalman Gains for Velocity Components

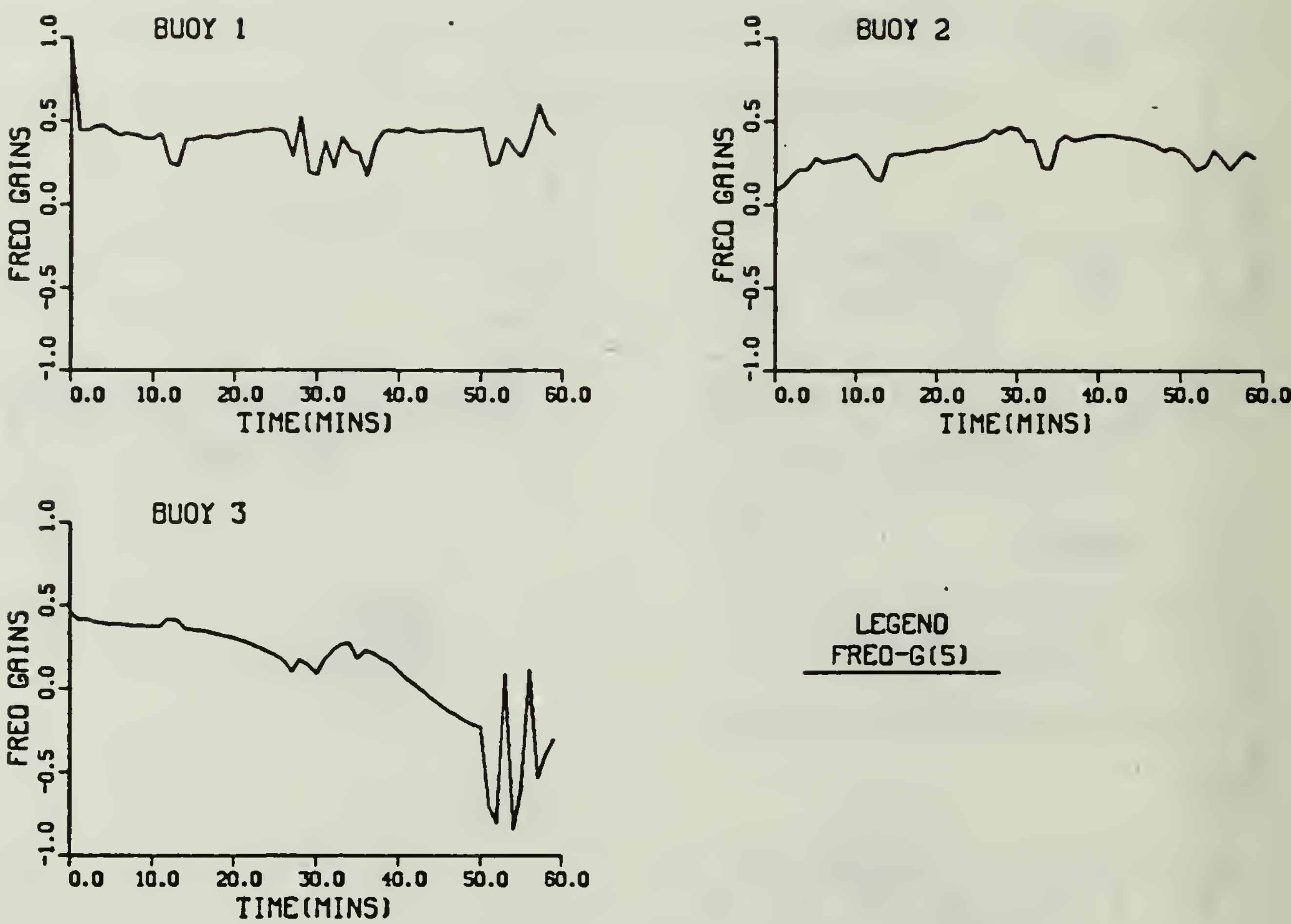

Figure 6-27. Scenario 1-Simulation 8:
Frequency Measurement - Kalman Gain for Frequency Component

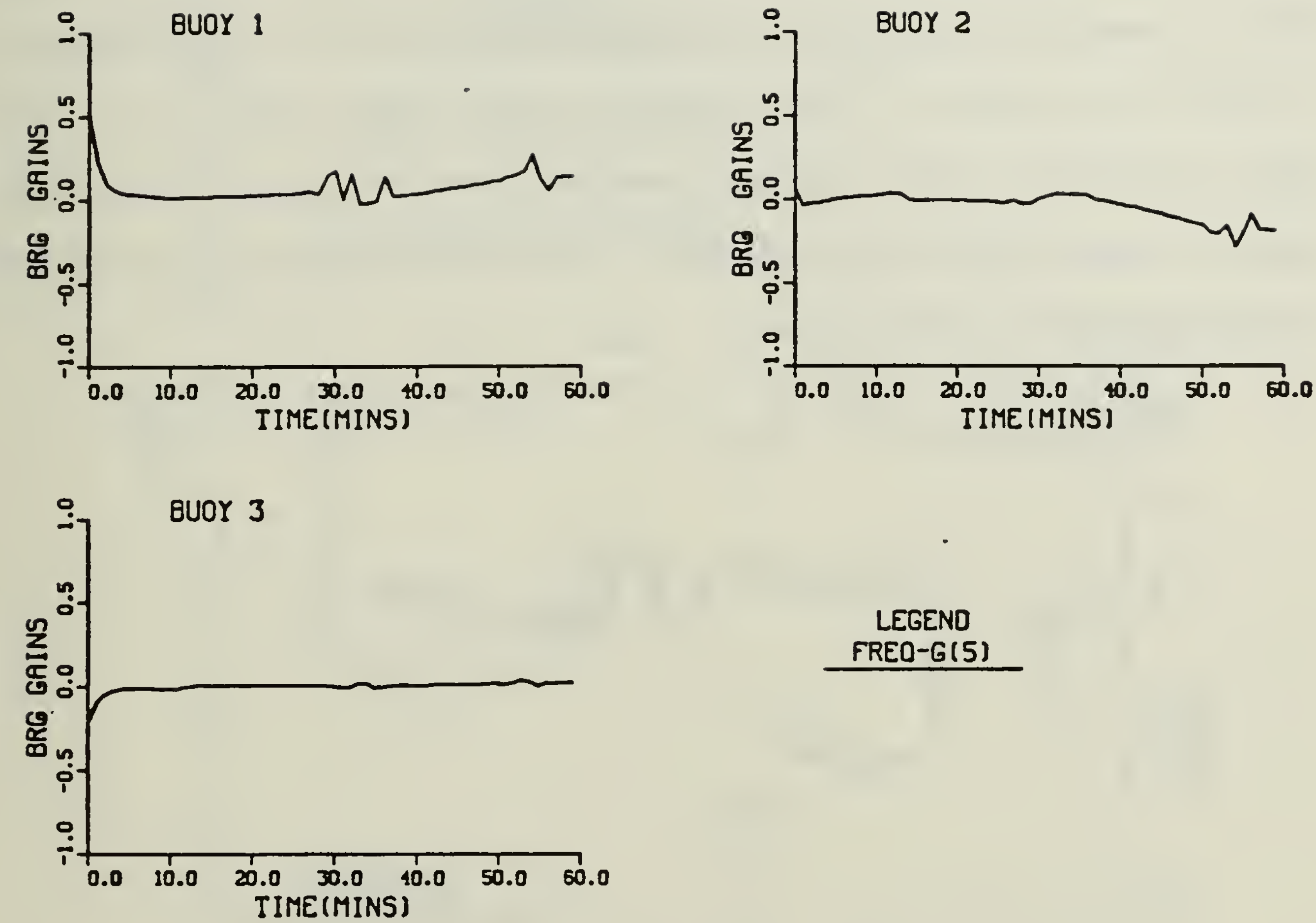

Figure 6-28. Scenario 1-Simulation 8:
Bearing Measurement - Kalman Gain for Frequency Component

C. SCENARIO 2

Scenario 2 is a continuation of Scenario 1-Simulation 8. The last state estimate $\hat{\underline{x}}(59|59)$ and last error covariance matrix P(59|59) from Simulation 8 is used to initialize Scenario 2. DIFAR 3 from Scenario 1 is dropped and another DIFAR 3 is placed west of the target's track as illustrated in figure (6-29). An enlarged geographic plot is shown in Figure (6-30). Table (6-8) lists the maximum position error for each maneuver. The left hand column of Table (6-30) is taken from Table (5-1). The filter's estimated track accurately reconstructs the actual track. As shown in figure (6-30) and Table (6-8) the maximum errors occur during the last segment of both "s" turns.

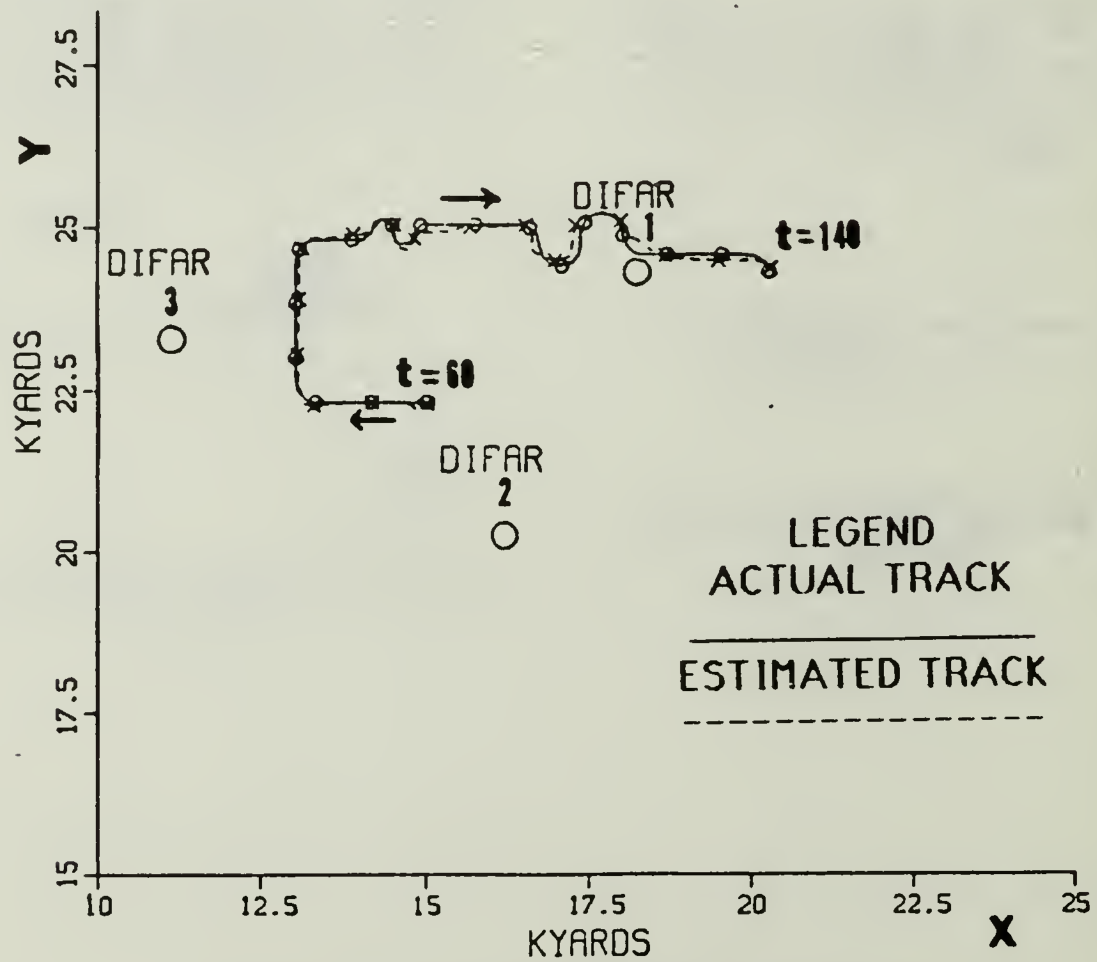

Figure 6-29. Scenario 2: GeographicPlot-Noise, $Q'_1(k)$, and Adaptive Control Applied

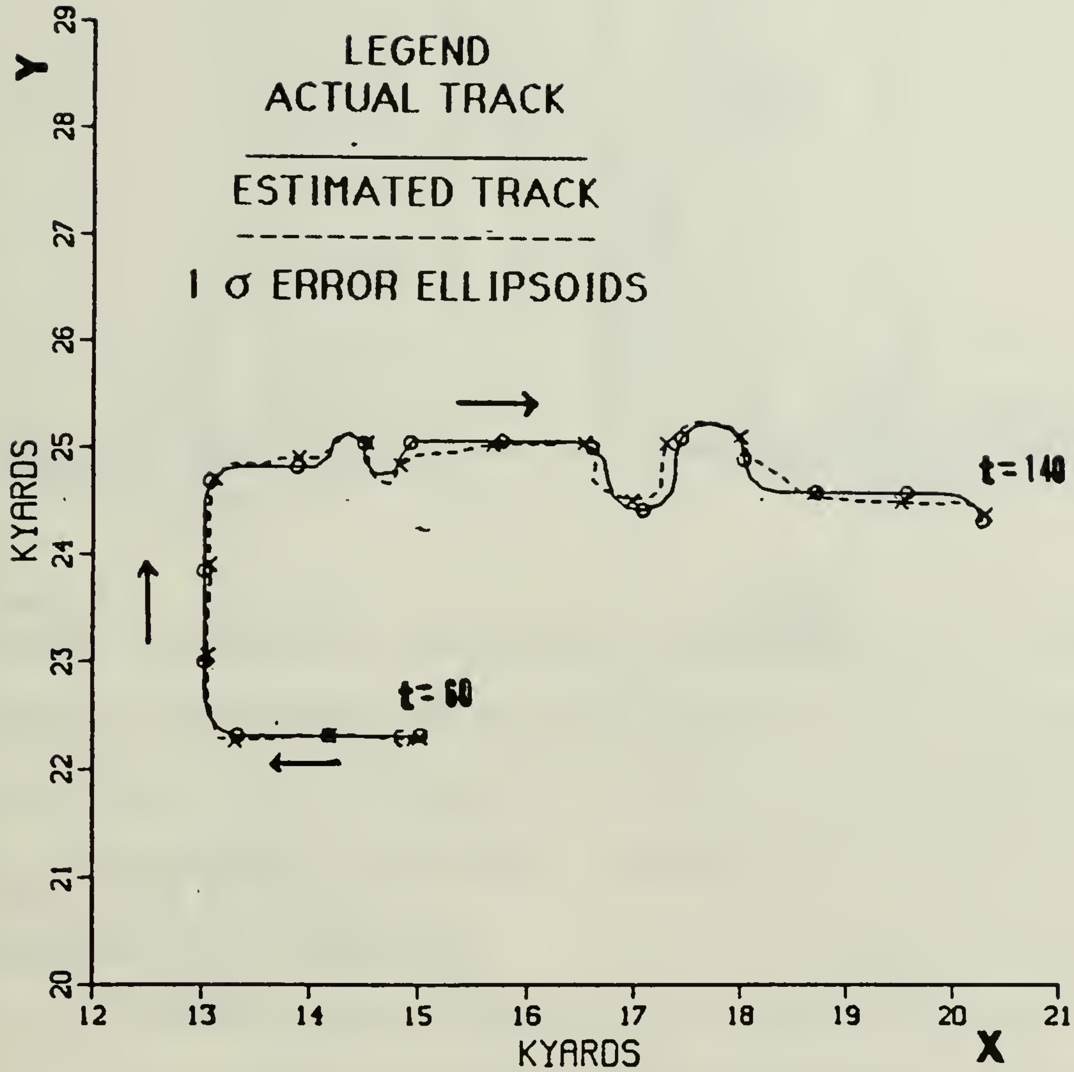

Figure 6-30 Scenario 2: Enlarged Geographic Plot-
Noise, $Q'_i(k)$, and Adaptive Control Applied

Table 6-8. SCENARIO 2: MAXIMUM ERROR FOR EACH MANEUVER

Time (Mins)	Time	Max Error
60-70	70	55 yds
71-73	71	75 yds
74-84	79	85 yds
85-87	86	59 yds
88-91	90	89 yds
92-101	99	250 yds
102-109	102	388 yds
110-128	127	239 yds
129-137	133	130 yds
138-140	138	71 yds

Figures (6-31) and (6-32) illustrates the predicted residual and adaptive gate for the frequency and bearing measurements respectively. Note that adaptive gate for Buoy 1 in Figure (6-31) is exceeded two times at t_k = 100 mins. Like Scenario 1, the bearing measurement's adaptive gate is never exceeded.

The position variances for the frequency measurement are illustrated in Figure (6-33). The x component variance increases from t_k = 130 mins to the end. This is due to the target's track being very close to DIFAR 1's location and the noise in DIFAR 1 frequency measurement. At t_k = 140 mins buoy 1's x component variance is 5.2×10^4 yds², which gives a standard deviation of 228 yds. The bearing measurement's position variances are shown in Figure (6-34). As expected, its graphs are similar to

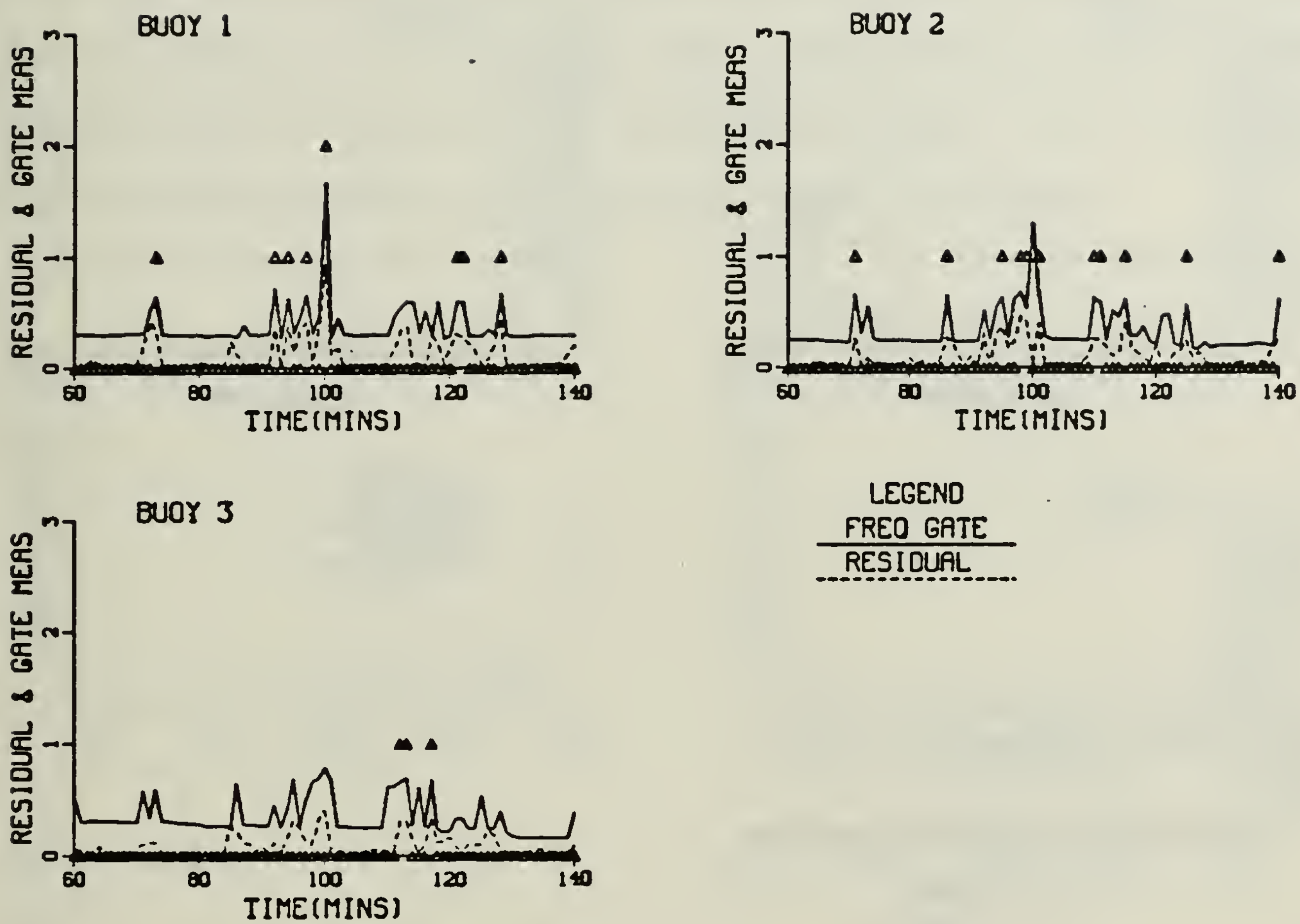

Figure 6-31. Scenario 2: Frequency Measurement-Predicted Residual and Adaptive Gate

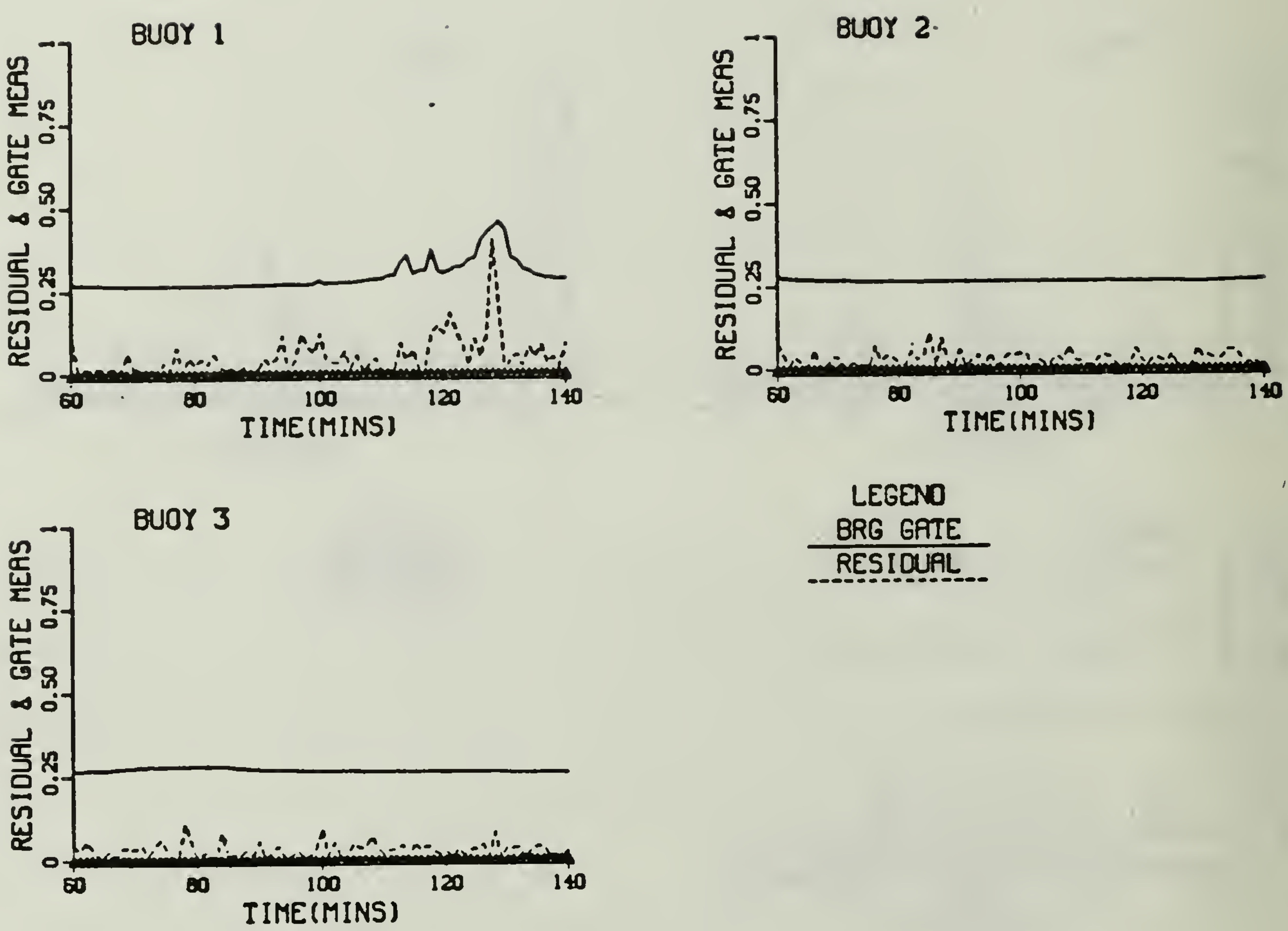

Figure 6-32. Scenario 2: Bearing Measurement-
Predicted Residual and Adaptive Gate

Figure (6-33). The values of the bearing measurement's positions variances are slightly less than the frequency measurement's position variances values.

Figure (6-35) illustrates the velocity components variances for the frequency and bearing measurement. As indicated for Scenario 1, the bearing measurement's velocity variances are very similar to the frequency measurement's velocity variances. Hence, only one graph is displayed. Note the large value for v_y variance for buoy 1 at t_k = 100 mins. This is when the target is completing the small "s" maneuver.

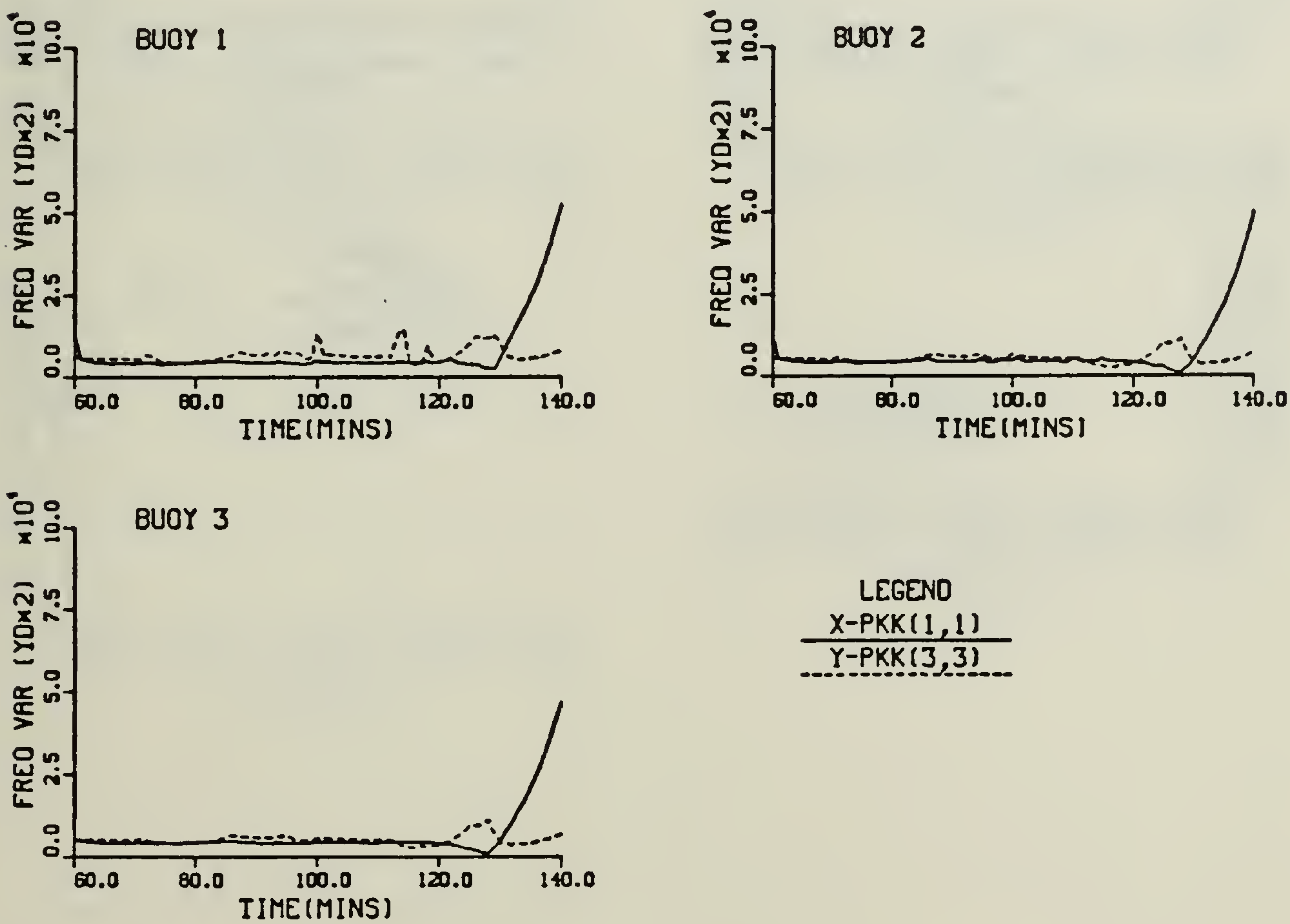

Figure 6-33. Scenario 2: Frequency Measurement-
Position Variances

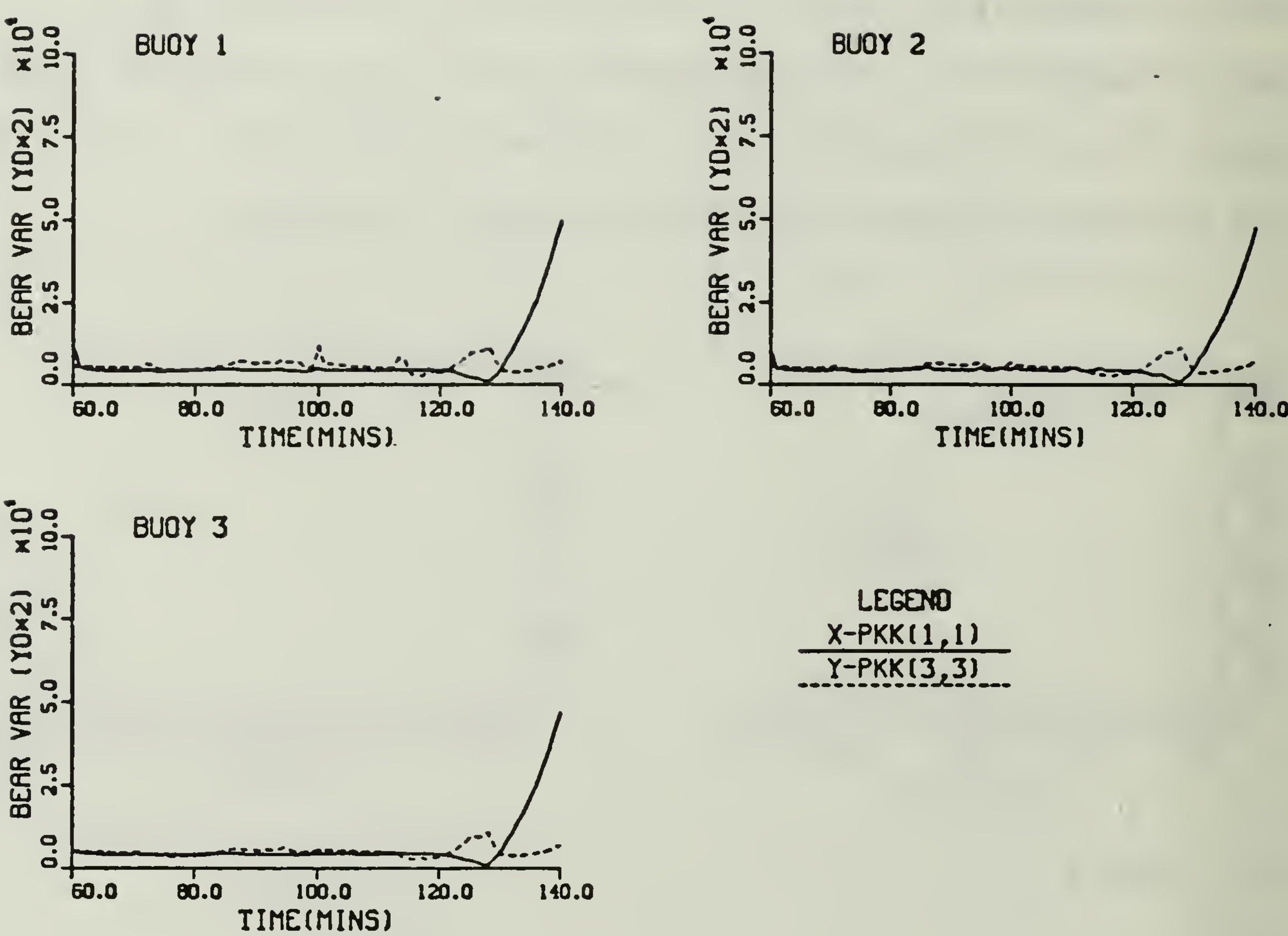

Figure 6-34. Scenario 2: Bearing Measurement-
Position Variances

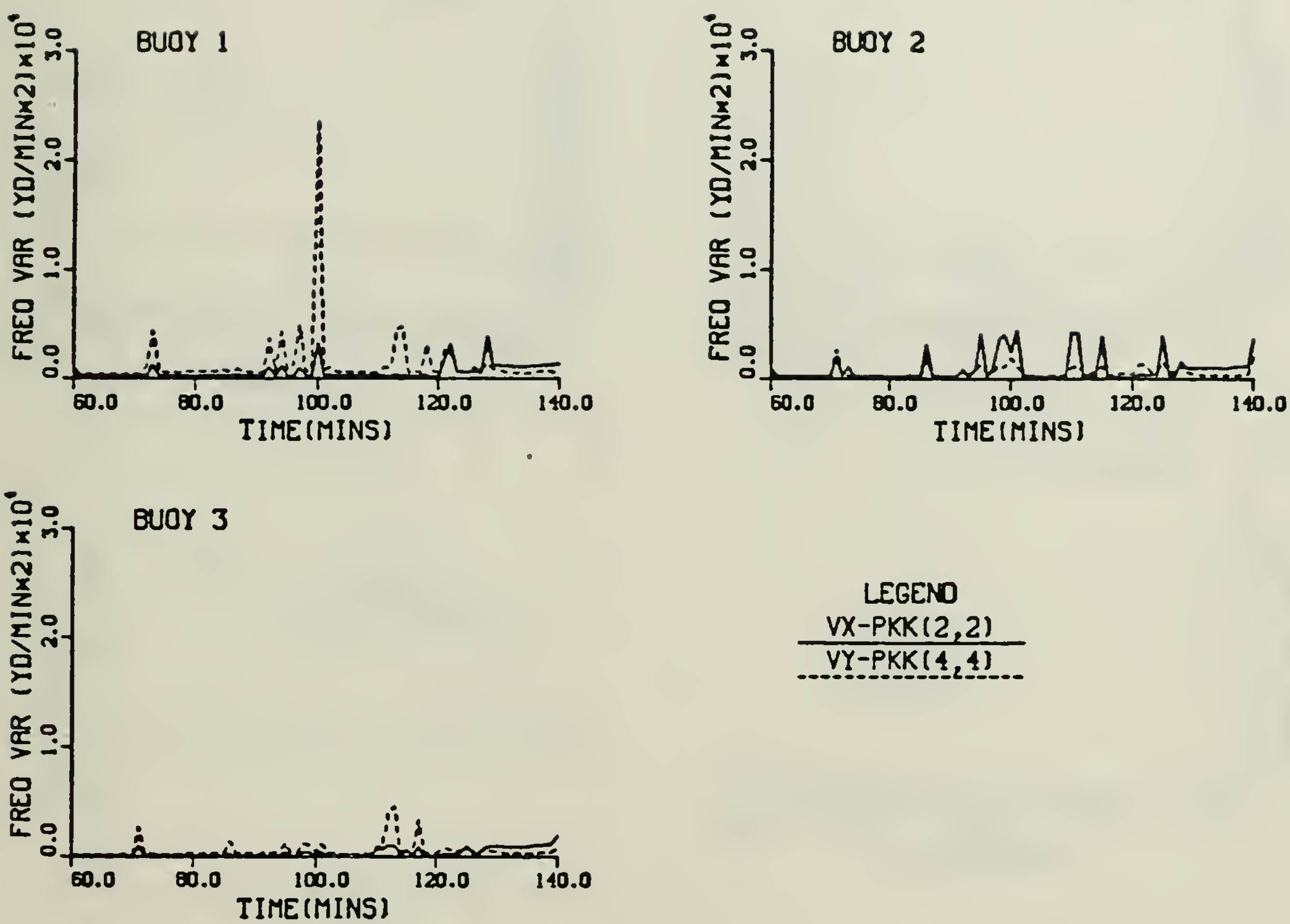

Figure 6-35. Scenario 2: Frequency and Bearing Measurement - Velocity Variances

The frequency component variance is shown in Figure (6-36). Since the bearing and frequency measurement's frequency component variance is nearly the same, only one graph is shown.

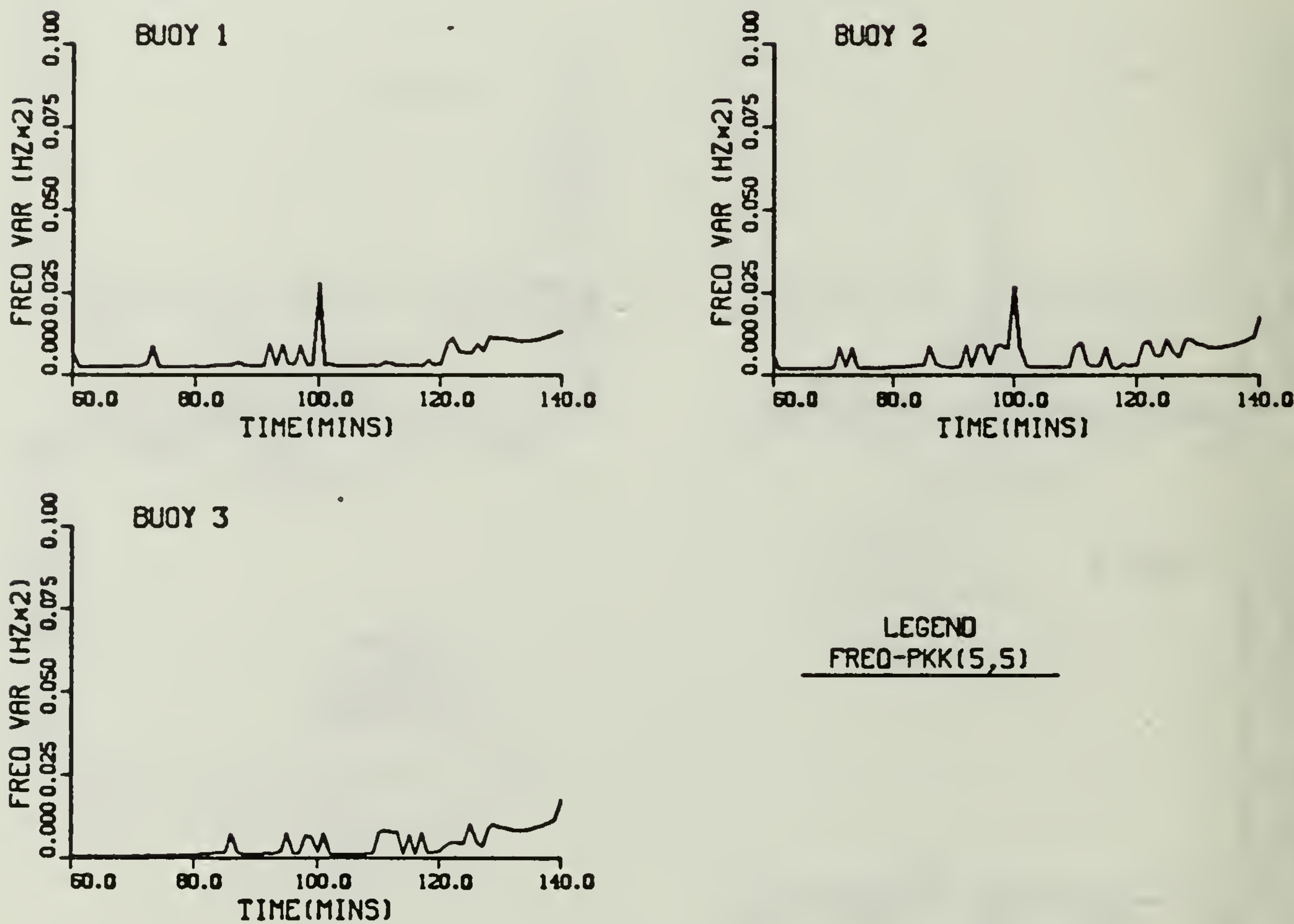

Figure 6-36. Scenario 2: Frequency and Bearing Measurement-Frequency Variance

Figures (6-37)-(6-42) illustrates the Kalman gains for the position, velocity and frequecny components. The position gains, Figures (6-37) and (6-38), are usually less than ± 100, but there are deviations when the target maneuvers. Again the velocity and frequency gains, Figures (6-39)-(6-42), are very erratic during the maneuvers.

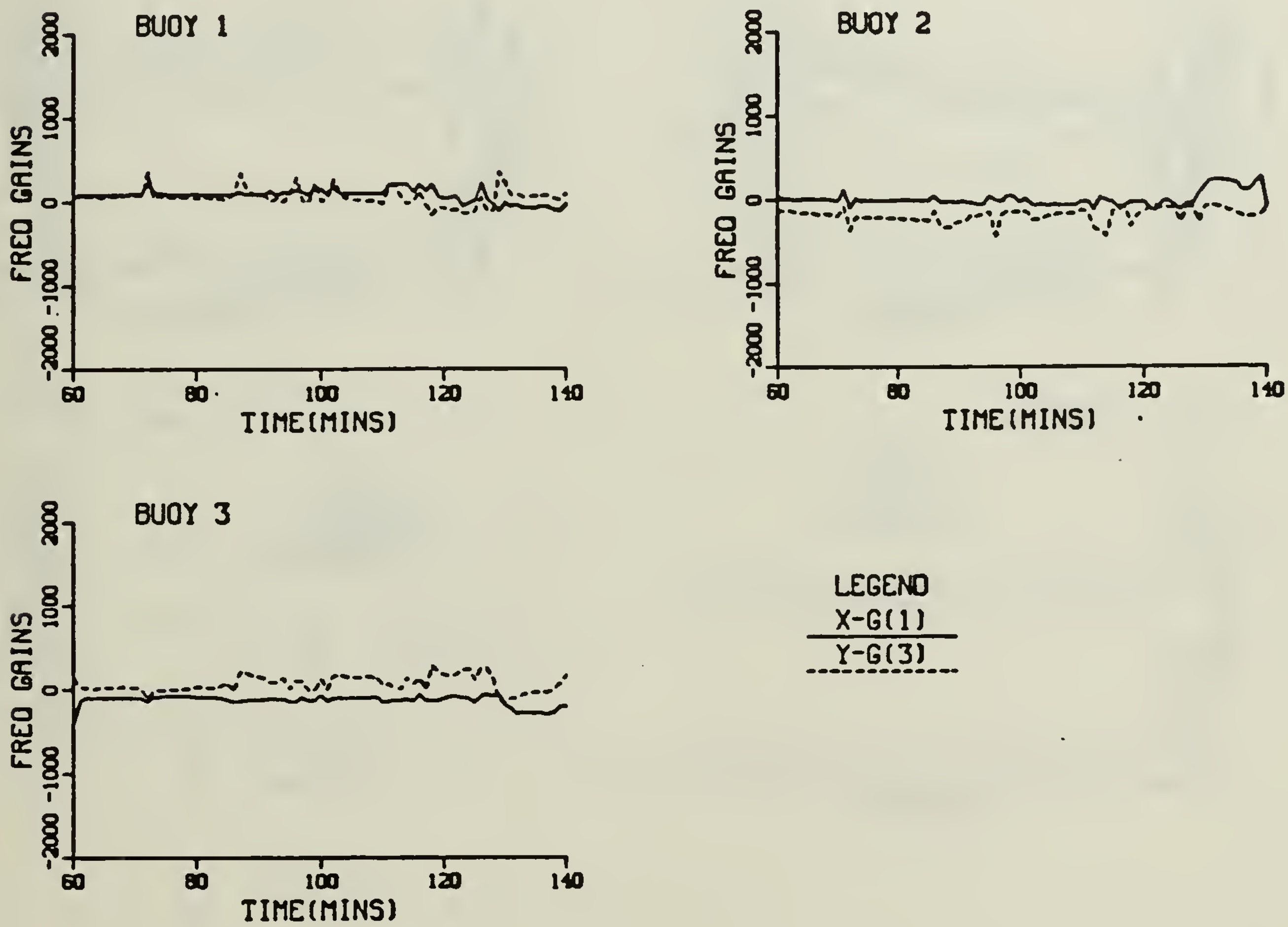

Figure 6-37. Scenario 2: Frequency Measurement-
Kalman Gains for Position Components

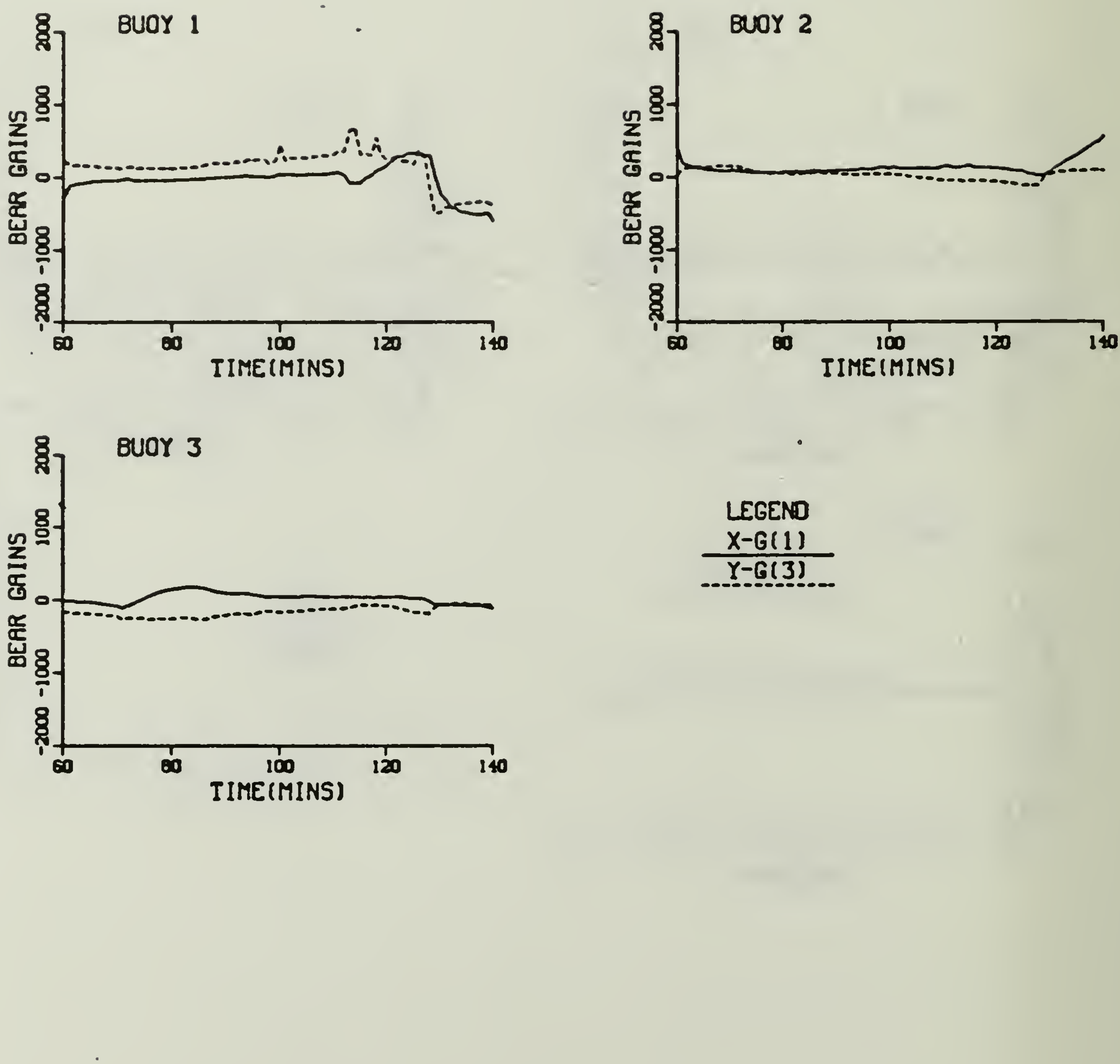

Figure 6-38. Scenario 2: Bearing Measurement-
Kalman Gains for Position Components

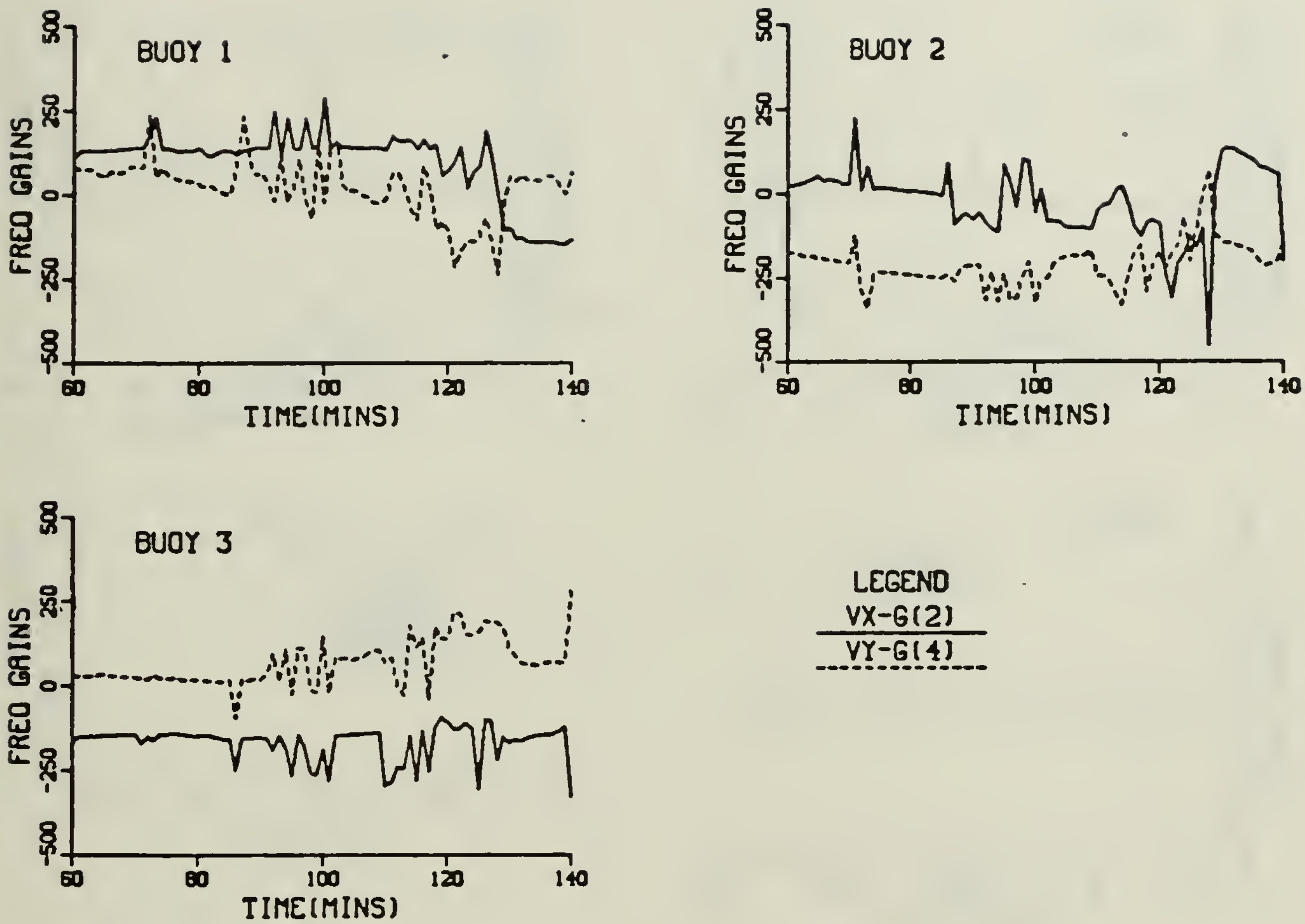

Figure 6-39 Scenario 2: Frequency Measurement-
Kalman Gains for Velocity Components

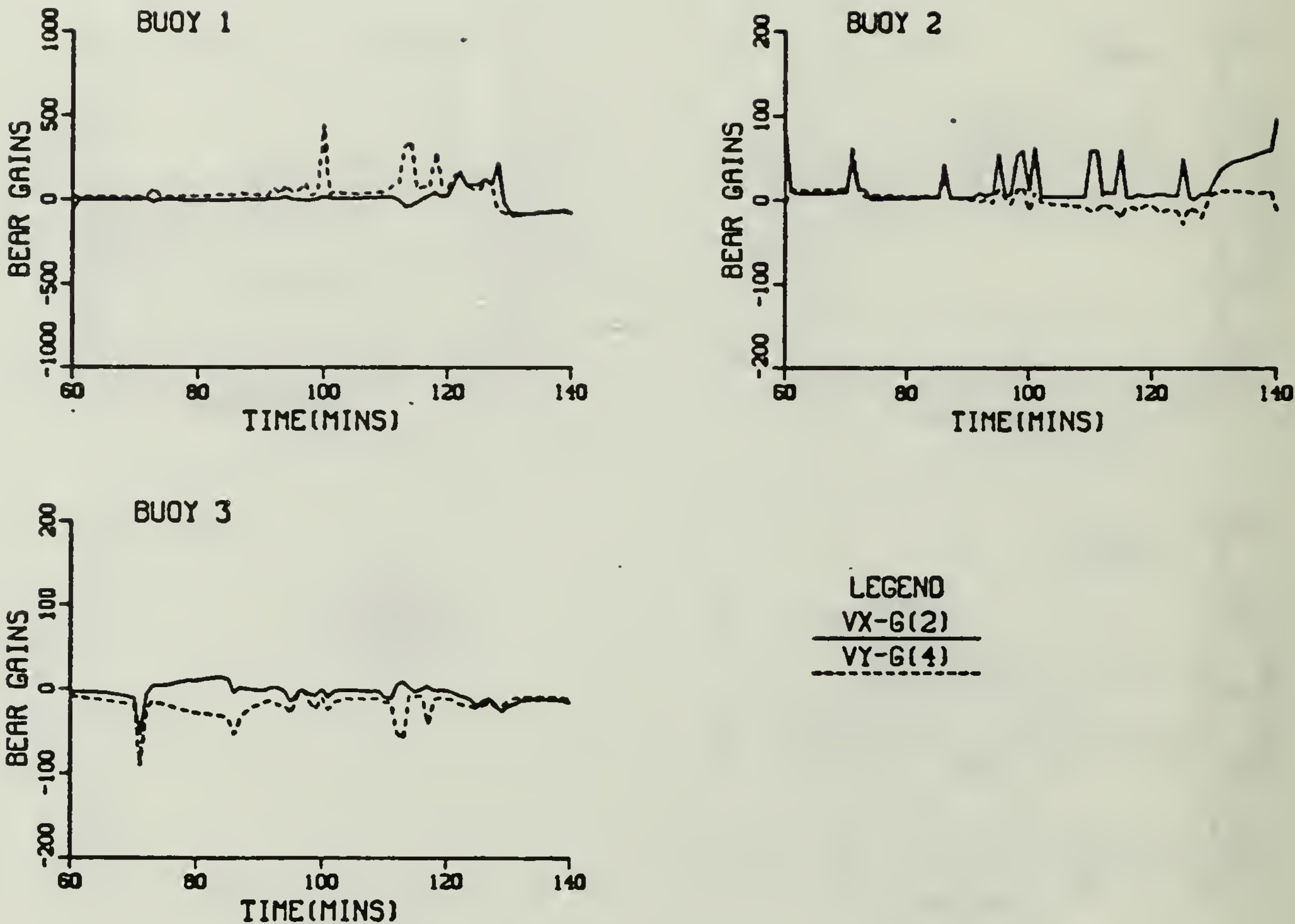

Figure 6-40. Scenario 2: Bearing Measurement - Kalman Gains for Velocity Components

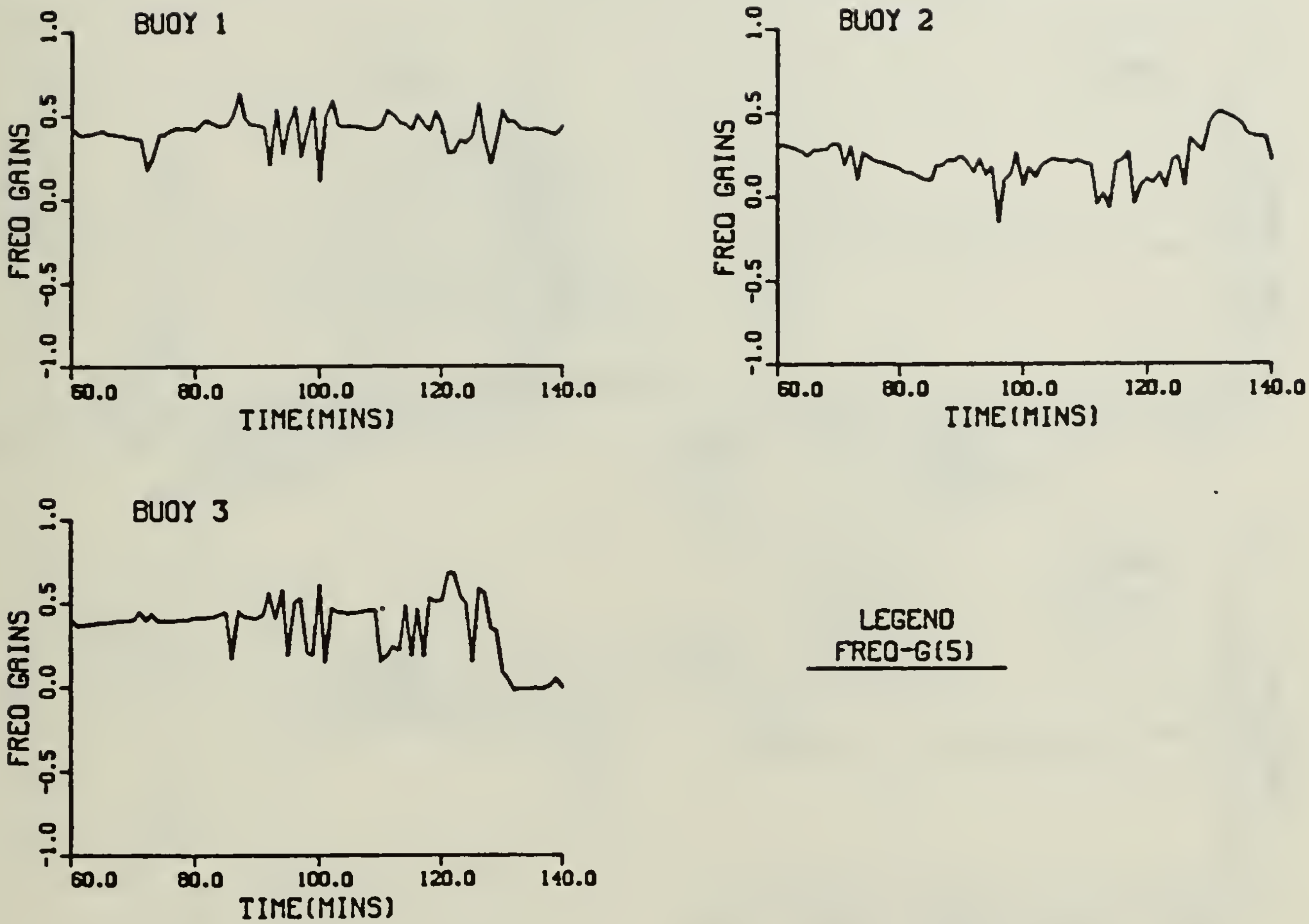

Figure 6-41. Scenario 2: Frequency Measurement-
Kalman Gain for Frequency Component

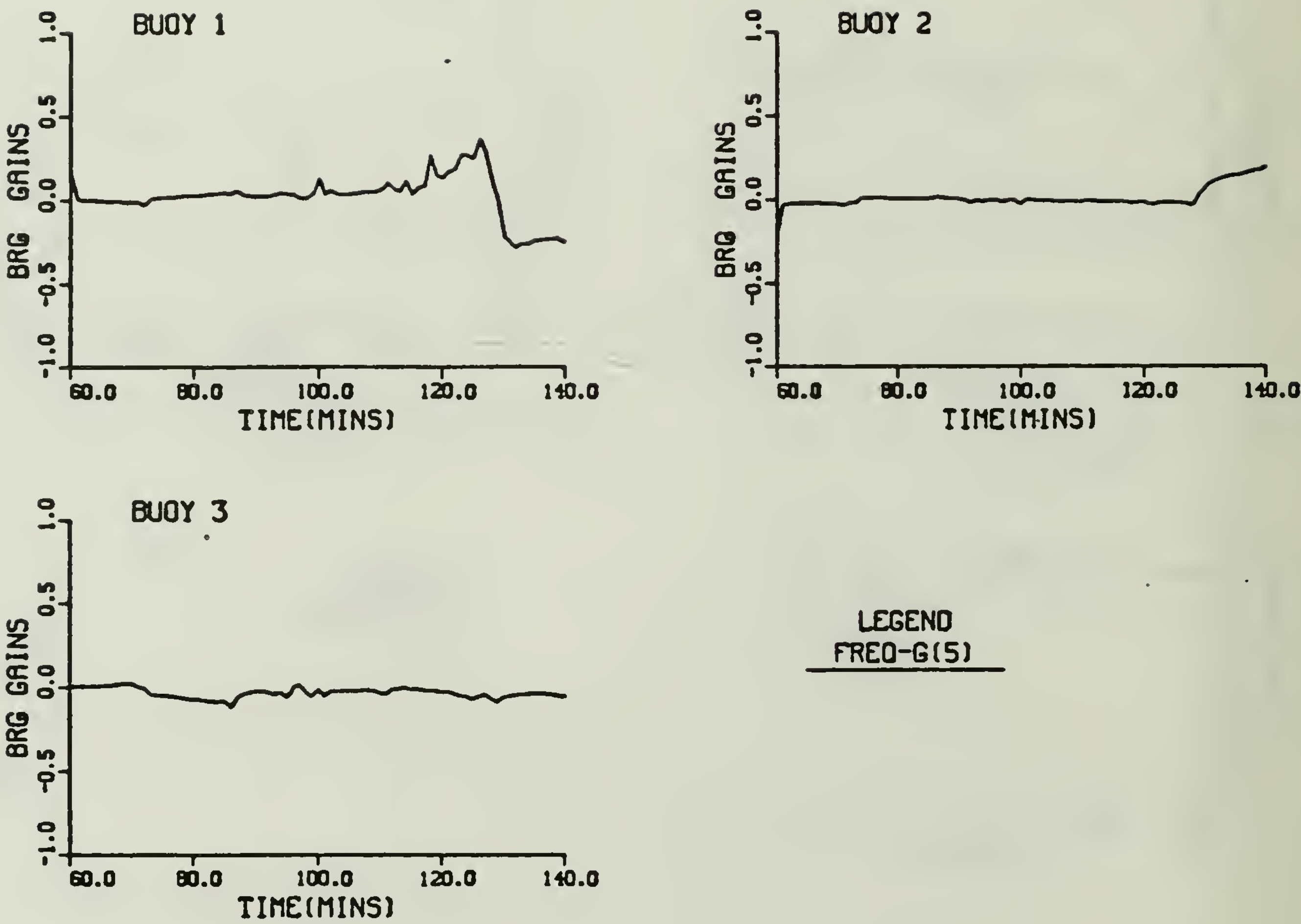

Figure 6-42. Scenario 2-: Bearing Measurement - Kalman Gain for Frequency Component

D. SCENARIO 3

Scenario 3 is a three sonobuoy scenario, where two of the buoys are LOFAR. A geographic plot of the target's track, sonobuoy pattern and the filter's estimated track are shown in Figure (6-43). An enlarged geographic plot is shown in Figure (6-44). The target's track is the same

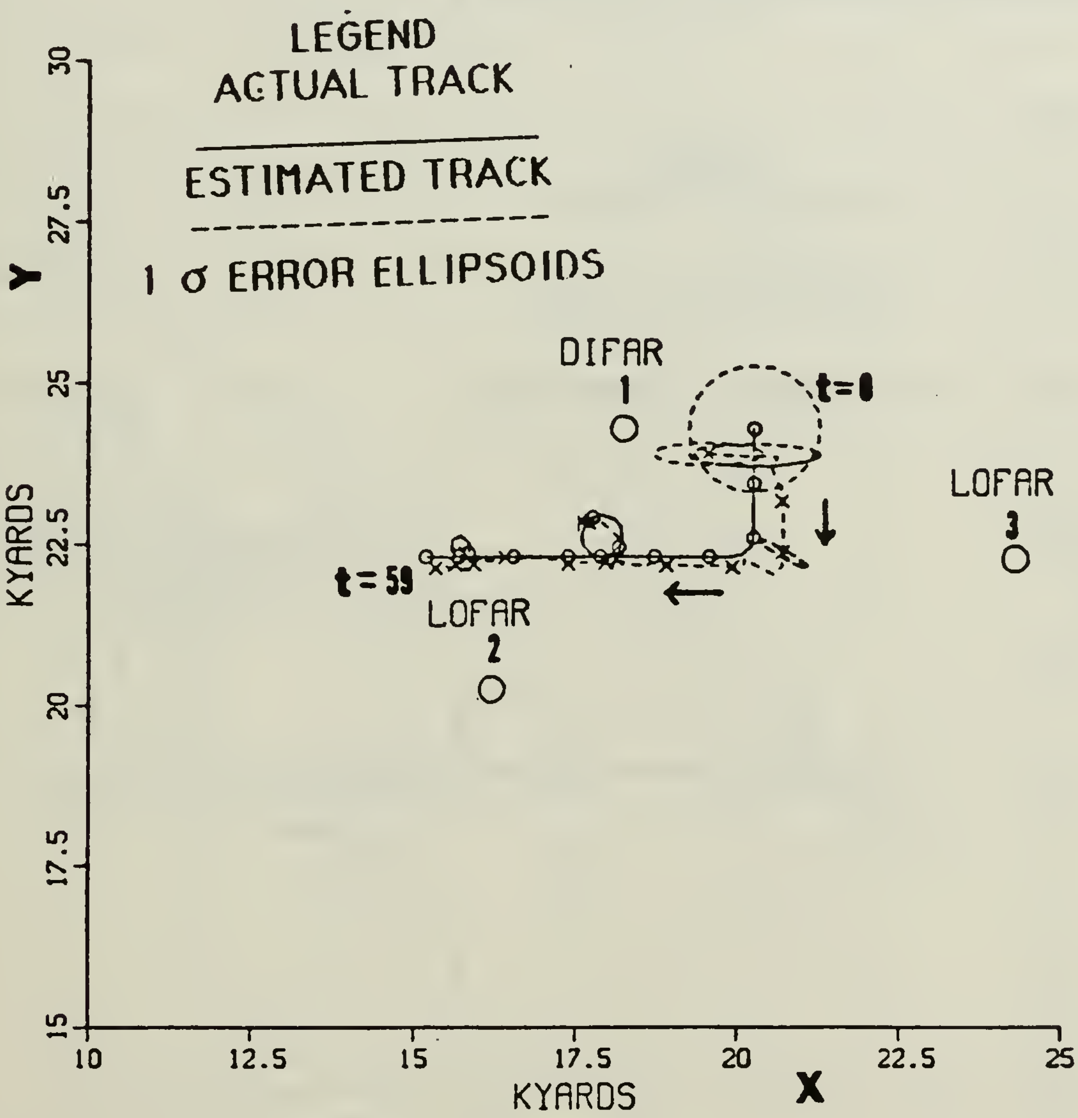

Figure 6-43. Scenario 3: Geographic Plot-Noise, $Q'_1(k)$, and Adaptive Control Applied

as the track in Scenario 1 and is described in Subsection V.B . The algorithm generates the estimates and predictions from the measurements in the following order:

1. Frequency measurement from DIFAR 1
2. Bearing measurement from DIFAR 1
3. Frequency measurement from LOFAR 2
4. Frequency measurement from LOFAR 3

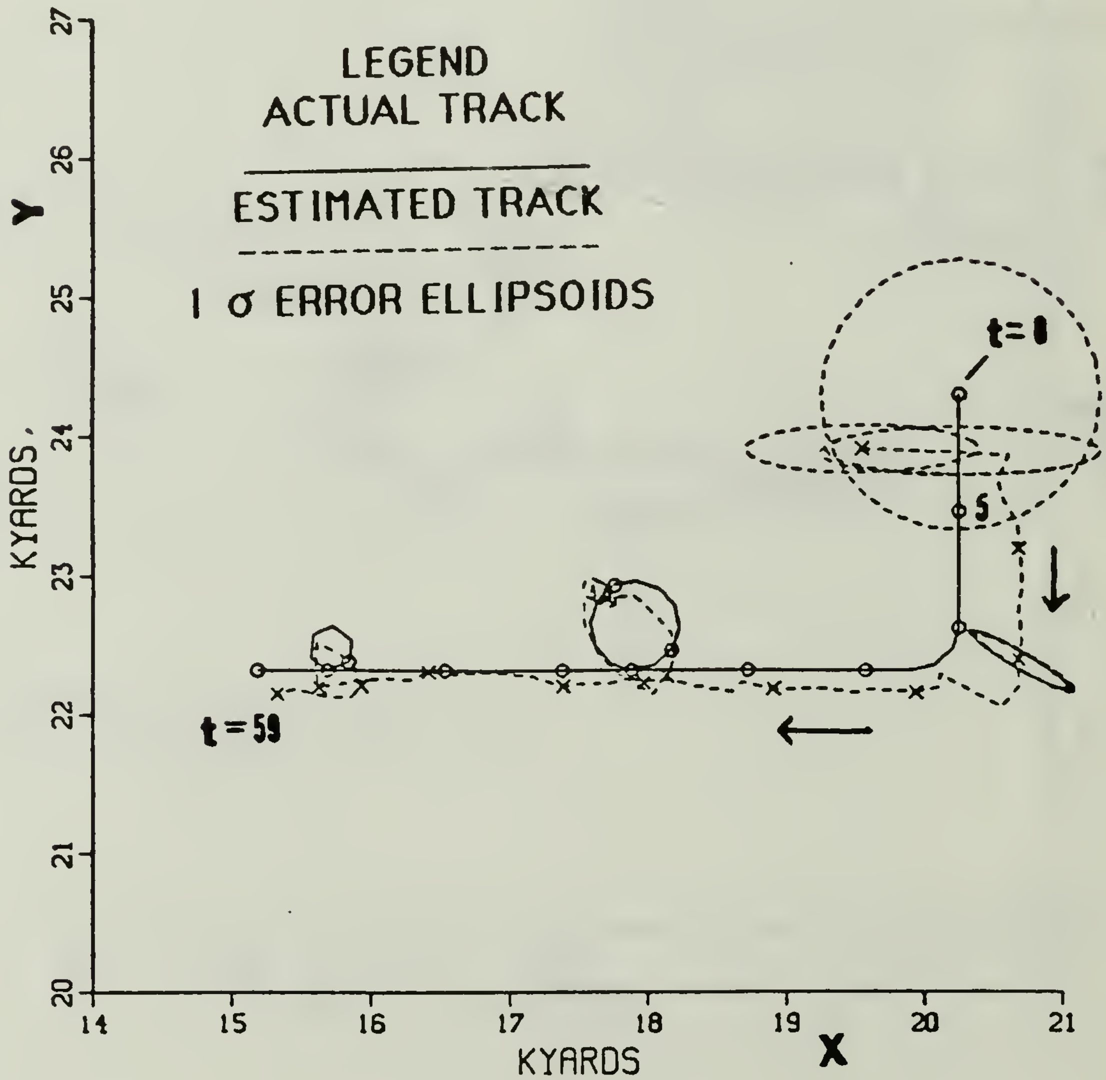

Figure 6-44 Scenario 3: Enlarged Geographic Plot- Noise, $Q'_1(k)$, and Adaptive Control Applied

The a priori information is the same as the information used in Scenario 1. The random forcing function covariance matrix $Q'_j(k)$, the adaptive gate, and noise are applied to the filter. From Table (5-2) set 2 frequency measurement standard deviations and from Table (5-3) set 2 bearing measurement standard deviations are applied to the noise-free measurements. The measurement noise standard deviations used in Scenario 3 are two times the measurement noise standard deviations used in Scenario 1. Table (6-9) lists the maximum position error for each maneuver. The errors are expected to be larger because twice the measurement noise has been applied to the simulation and there is only one bearing measurement per time interval.

Table 6-9. SCENARIO 3: MAXIMUM ERROR FOR EACH MANEUVER

Time (Mins)	Time	Max Error
0-10	0	808 yds
11-13	12	577 yds
14-25	15	394 yds
26-37	36	209 yds
38-50	49	138 yds
51-56	56	258 yds
57-59	57	239 yds

Examining the error ellipsoids at t_k = 0 mins gives us some insight into the large initial error. The large dashed circle error ellipsoid is from DIFAR 1's frequency measurement. Its estimated position is the target's actual starting position. The right horizontal ellipse is due to DIFAR 1's bearing measurement. This noisy bearing measurement from DIFAR 1 is 102 degs, 12 degs more than the noise-free measurement. This causes the estimated position to be 410 yds south of the target's actual starting position. LOFAR 2 frequency measurement estimated position is nearly the same as DIFAR 1's bearing measurement. LOFAR 3 noisy frequency measurement is 0.134 lower than the noise-free doppler frequency measurement. Its estimated position is 808 yds southwest of the actual target. LOFAR 3 shifts the error ellipsoid to the left. Hence, the large initial error is due to DIFAR 1's noisy bearing measurement and LOFAR 3's noisy frequency measurement. The shape of the error ellipsoid indicates that the majority of the error is along the bearing measurement of DIFAR 1. The error ellipsoids for t_k = 10 mins and t_k = 30 mins are align with DIFAR 1's bearing measurement to the actual position, but the actual position is outside of the error ellipsoids. The filter takes approximately twenty minutes to lock on to the target with errors less than 250 yds. As can be seen in figure (6-44) the filter detects maneuvers and tracks the target through all the maneuvers.

Figures (6-45) illustrates the predicted residual and adaptive gate for the frequency and bearing measurements. Note at t_k = 41 mins, the bearing measurement for buoy 1 causes the predicted residual to exceed the adaptive gate two times. The increase in the adaptive gate causes the large peak in buoy 2's and buoy 3's adaptive gate at t_k = 41 mins.

The variances of the position, velocity and frequency components are shown in figures (6-46)- (6-48). Figures (6-50)-(6-52) illustrates the Kalman gains for the position, velocity and frequency components.

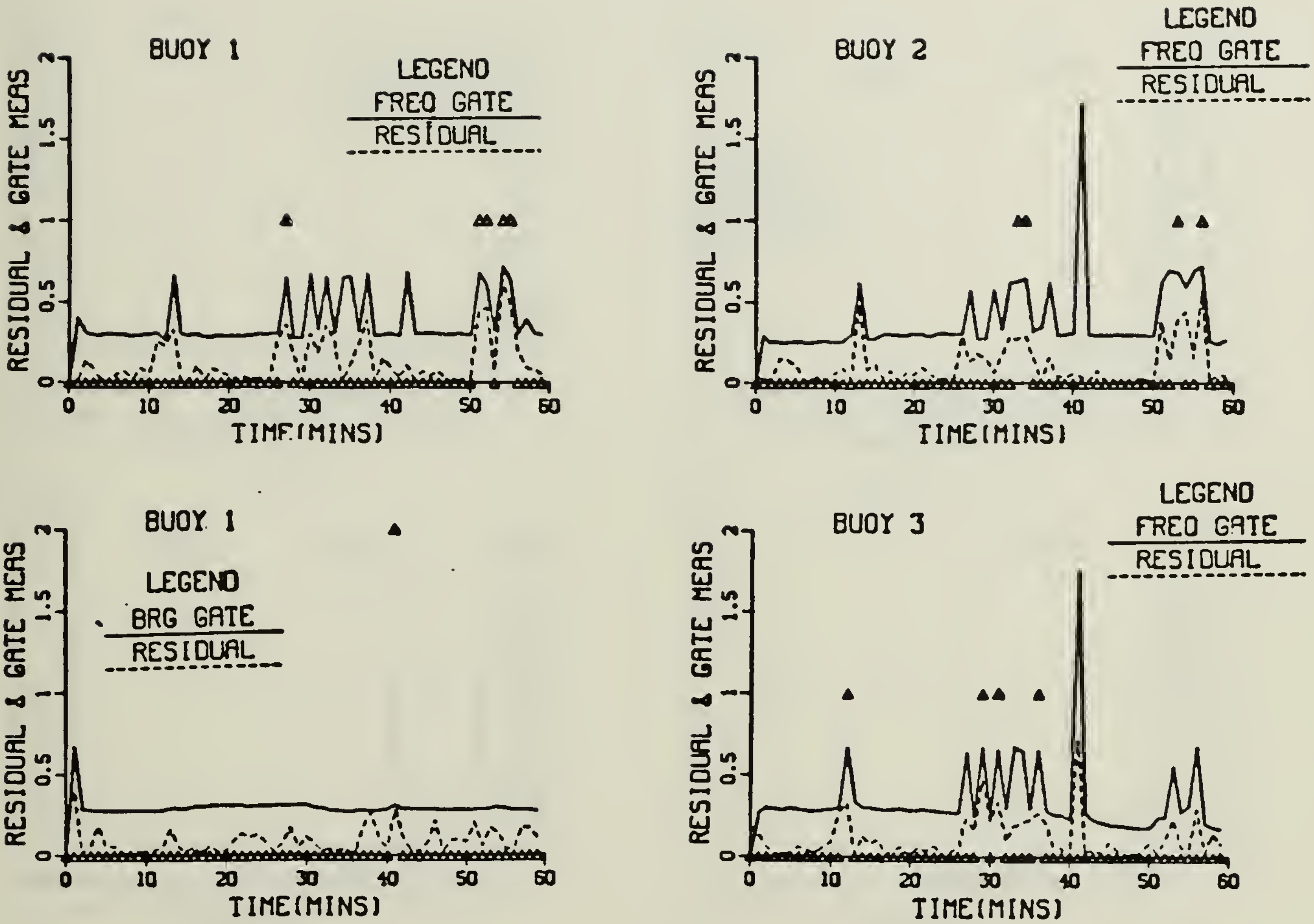

Figure 6-45. Scenario 3: Frequency and Bearing Measurement- Predicted Residual and Adaptive Gate

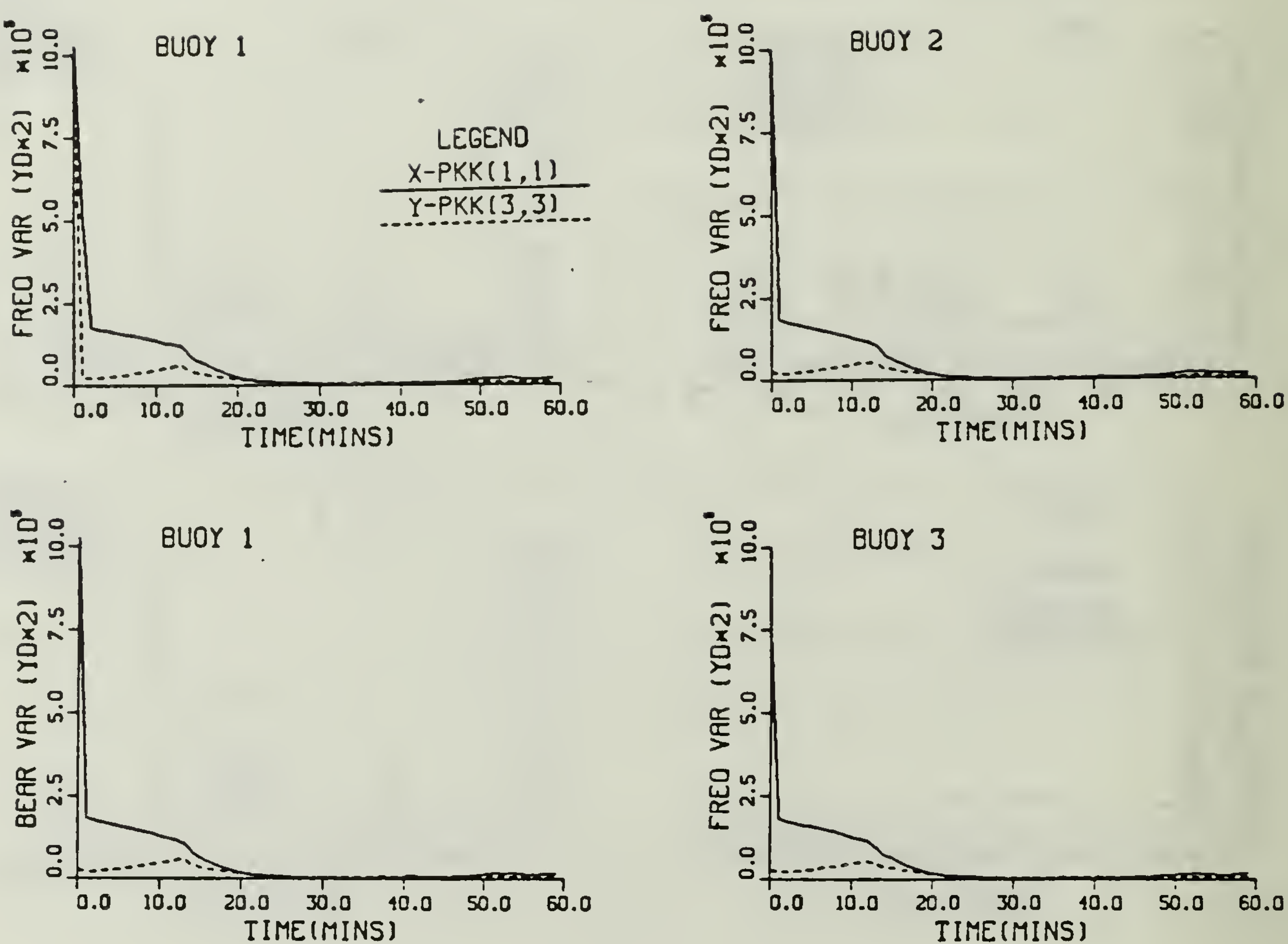

Figure 6-46. Scenario 3: Frequency and Bearing Measurement- Position Variances

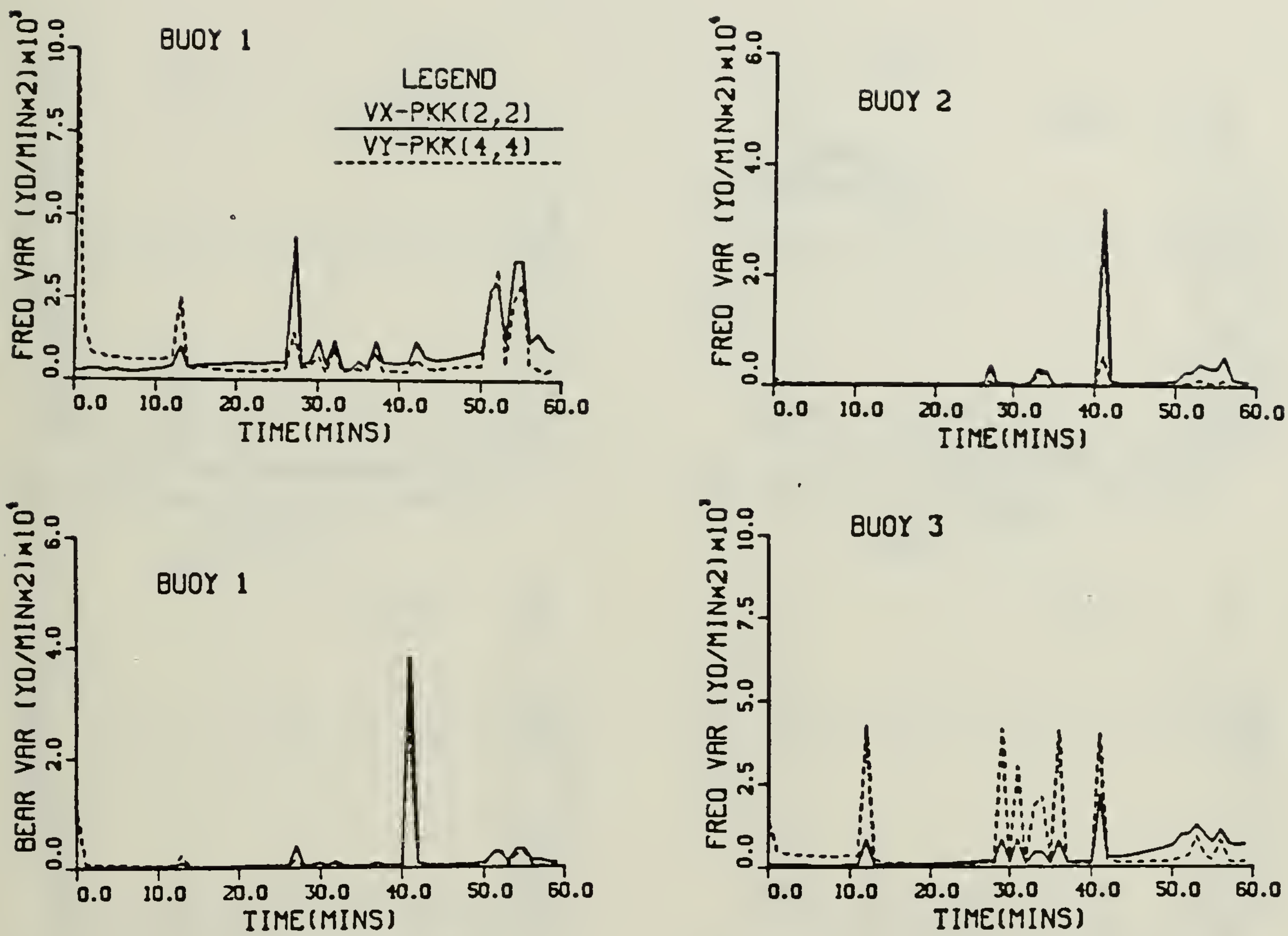

Figure 6-47. Scenario 3: Frequency and Bearing Measurement - Velocity Variances

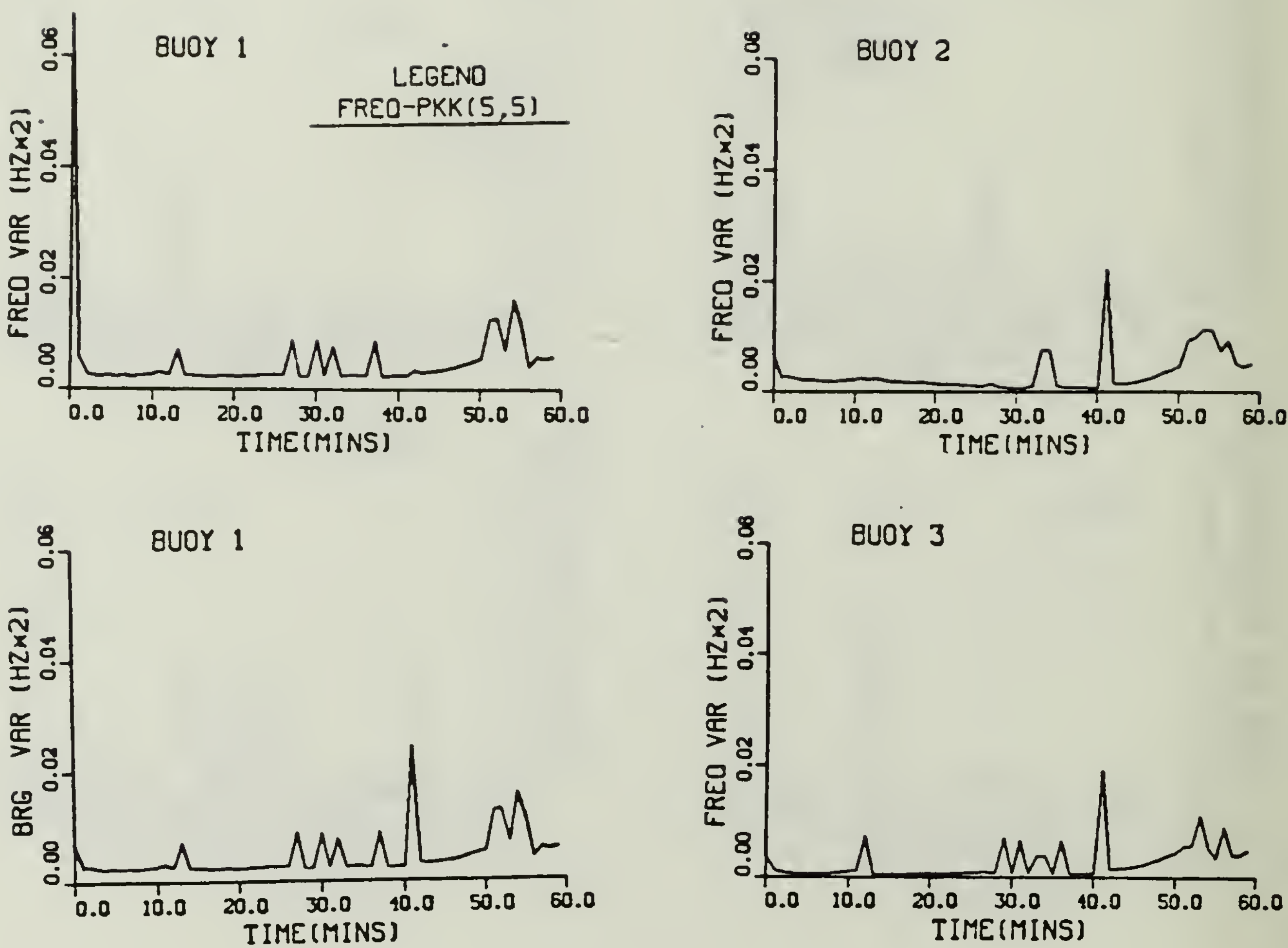

Figure 6-48. Scenario 3: Frequency and Bearing Measurement-Frequency Variance

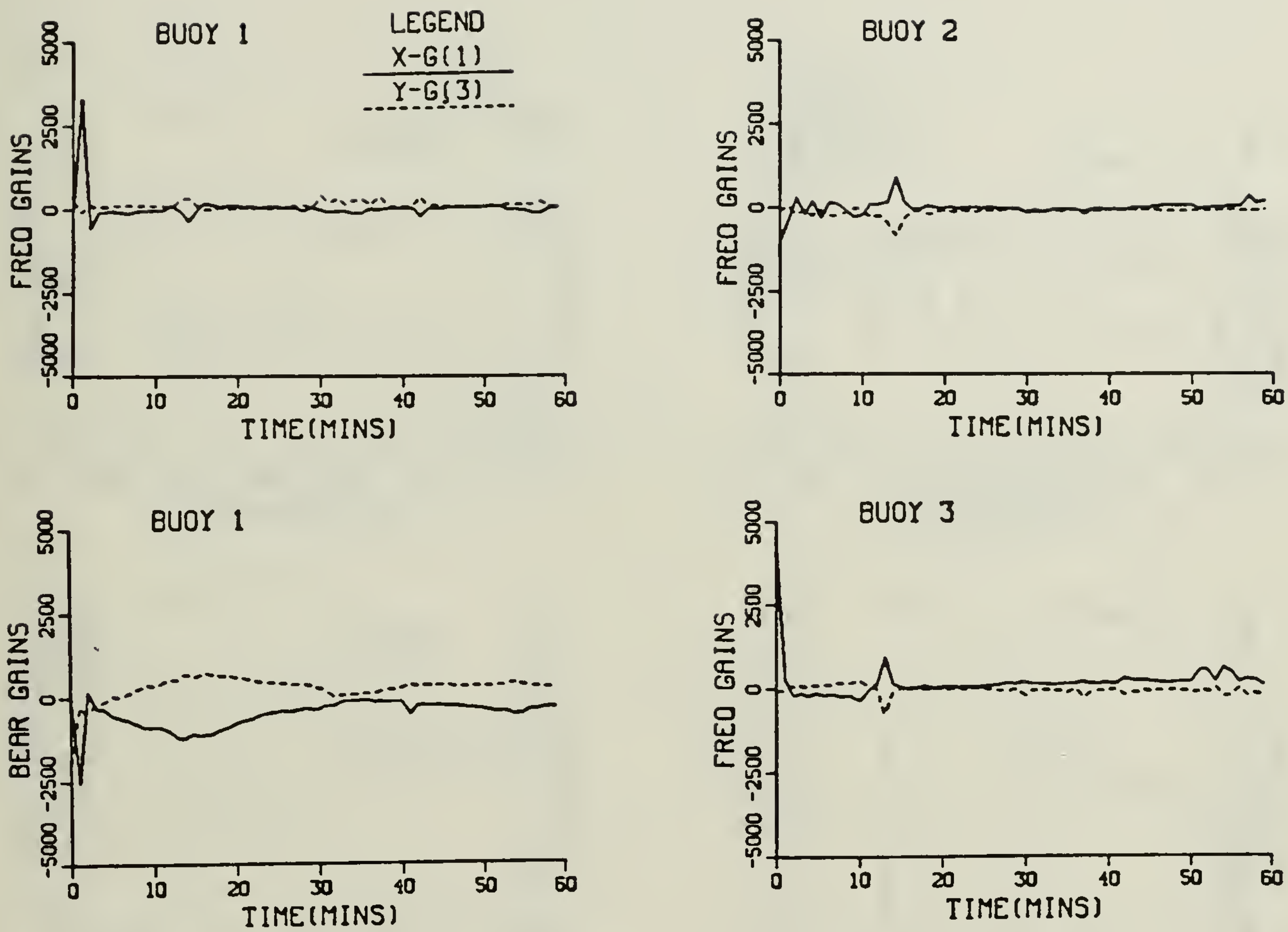

Figure 6-49. Scenario 3: Frequency and Bearing Measurement-Kalman Gains for Position Components

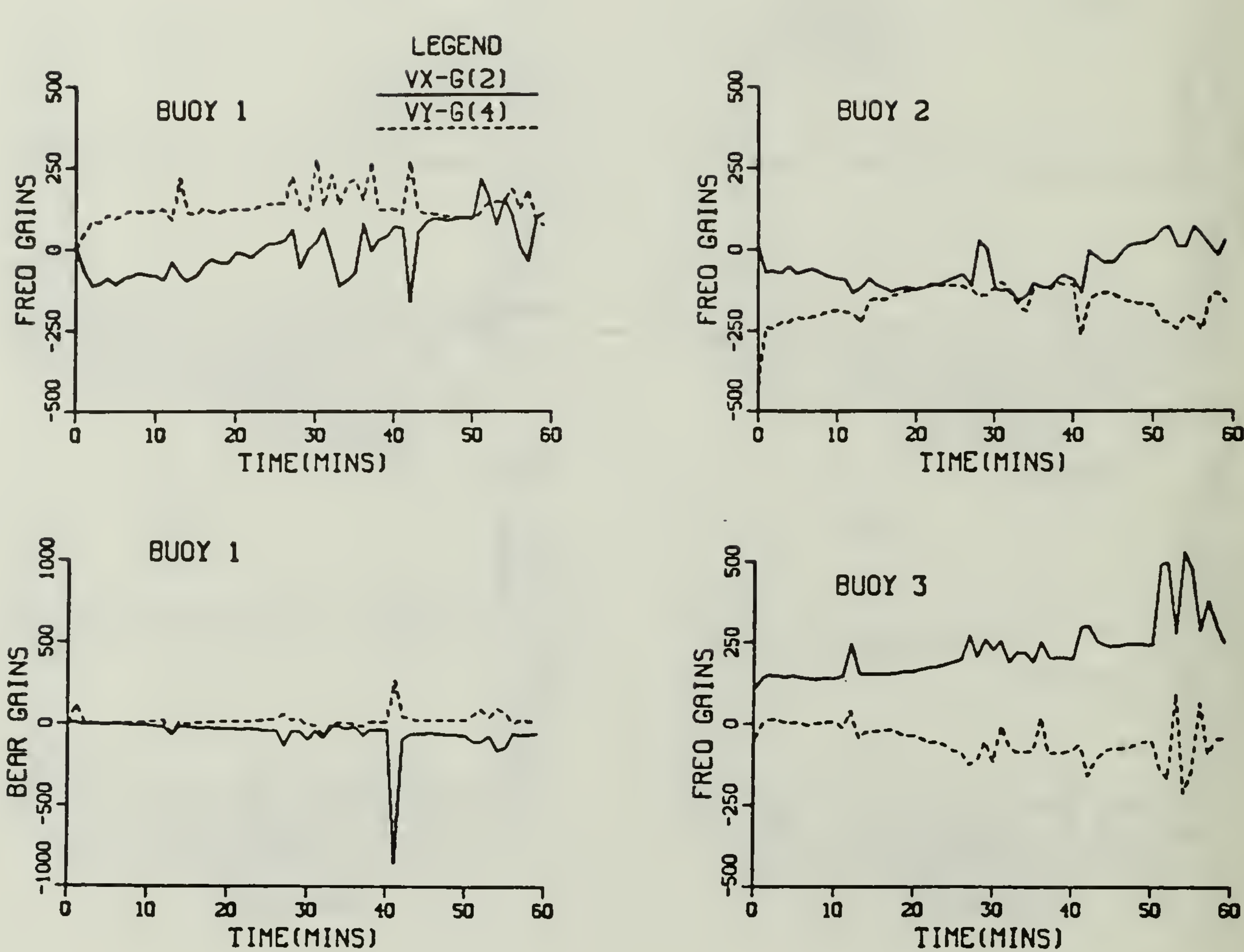

Figure 6-50 Scenario 3: Frequency and Bearing Measurement- Kalman Gains for Velocity Components

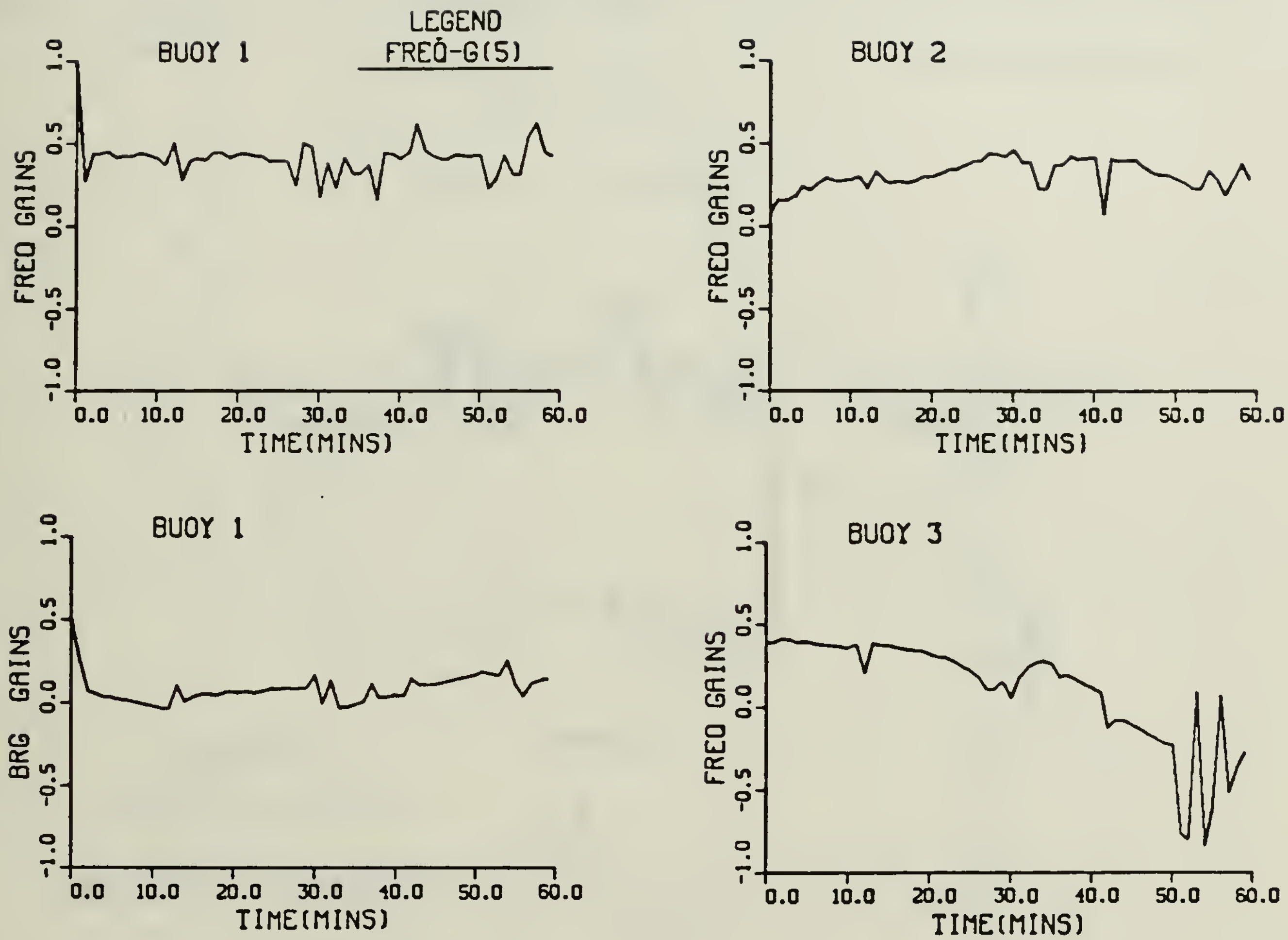

Figure 6-51. Scenario 3: Frequency and Bearing Measurement - Kalman Gain for Frequency Component

E. SCENARIO 4

Scenario 4 is a continuation of Scenario 3. Like Scenario 2, the last state estimate, $\hat{x}(59|59)$, and last error covariance matrix, P(59|59), from Scenario 3 is used to initialize Scenario 4. As illustrated if Figure (6-52), LOFAR 3 from Scenario 3 is dropped and another LOFAR 3 is placed west of the target's track. An enlarged geographic plot is shown in Figure (6-53). Table (6-10) lists the maximum position error for each segment of the targets track.

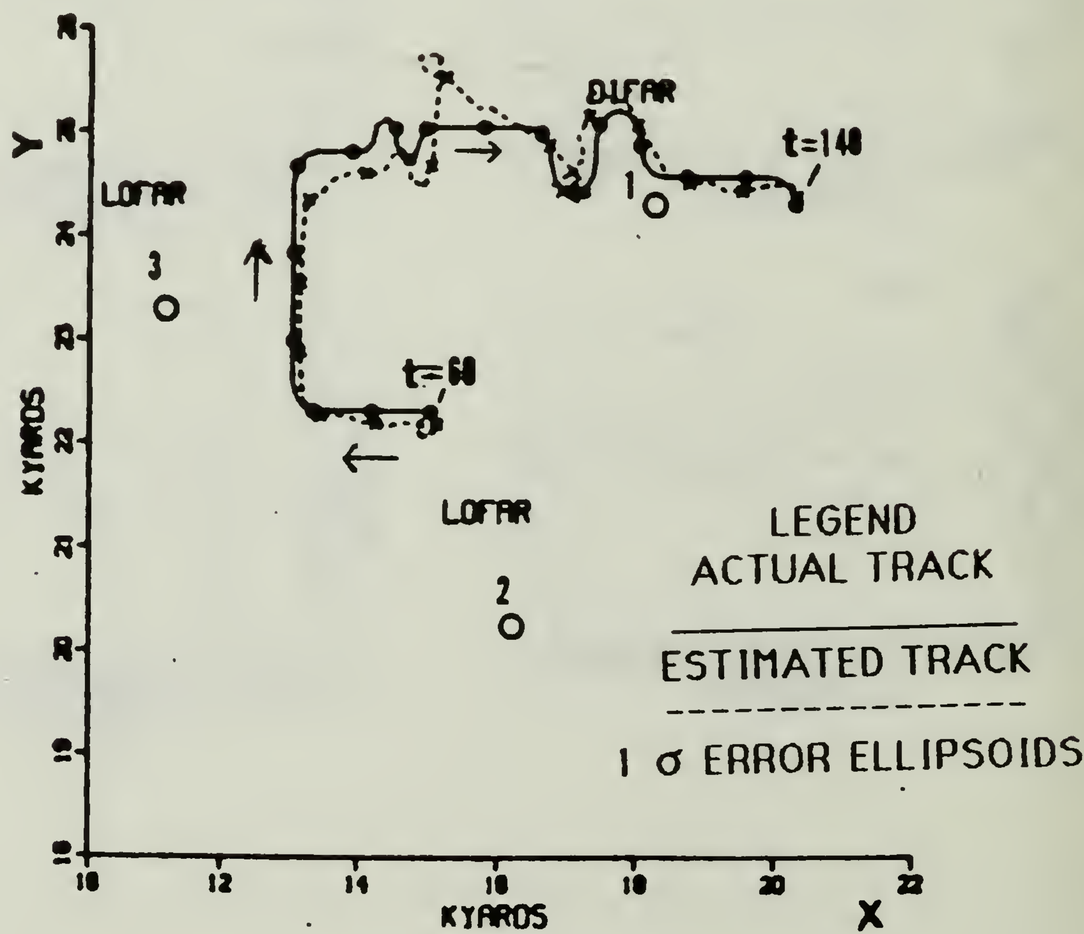

Figure 6-52. Scenario 4: Geographic Plot-Noise, $Q'_1(k)$, and Adaptive Control Applied

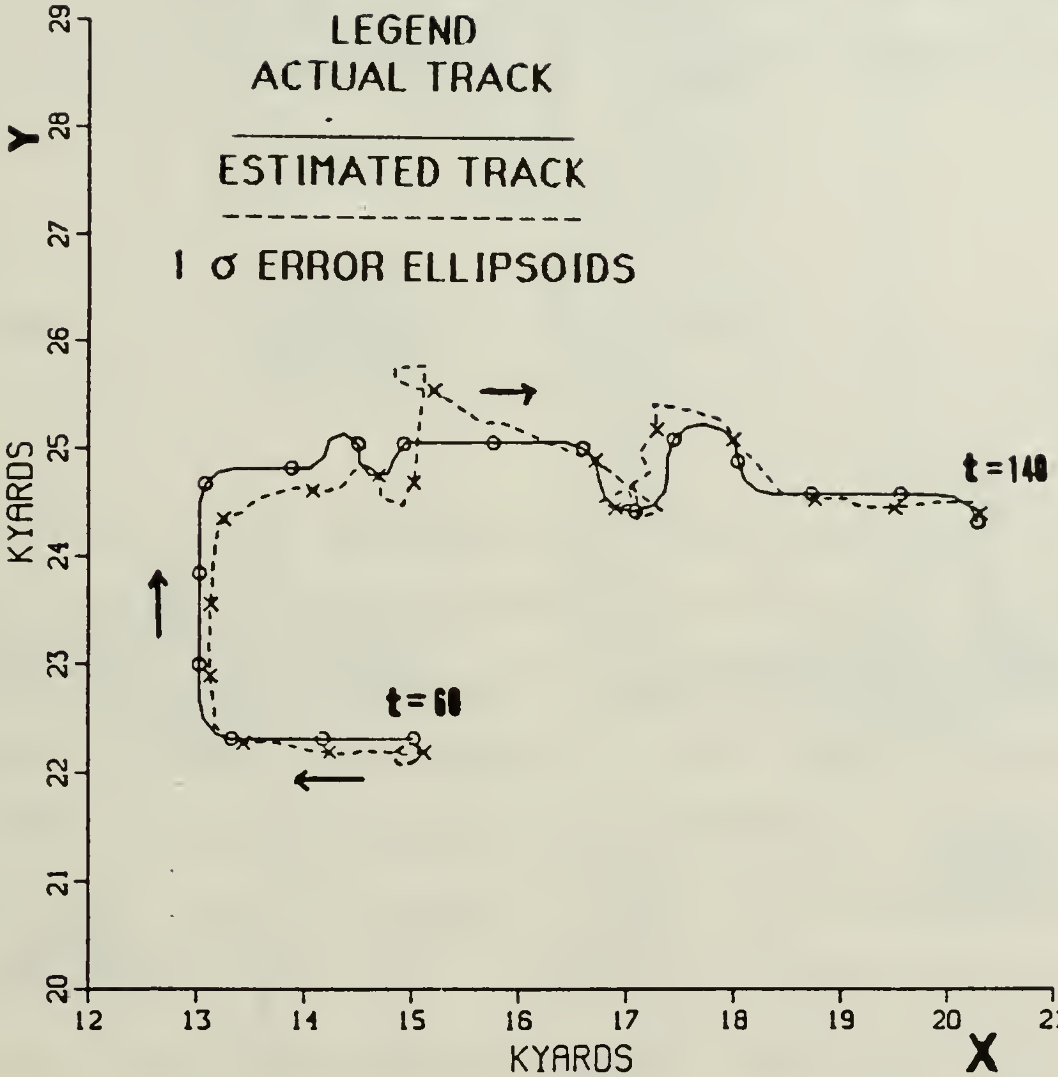

Figure 6-53 Scenario 4: Enlarged Geographic Plot-
Noise, $Q'_1(k)$, and Adaptive Control Applied

Table 6-10. SCENARIO 4: MAXIMUM ERROR FOR EACH MANEUVER

Time (Mins)	Time	Max Error
60-70	61	177 yds
71-73	73	132 yds
74-84	83	351 yds
85-87	86	436 yds
88-91	88	309 yds
92-101	101	718 yds
102-109	103	847 yds
110-128	111	770 yds
129-137	134	145 yds
138-140	140	85 yds

The filter's estimated track accurately reconstructs the actual track from t_k = 60 mins to t_k = 75 mins. At t_k = 73 mins the frequency measurement from buoy 1 causes the predicted residual to exceed the adaptive gate. Figure (6-54) illustrates the predicted residual and adaptive gate for the frequency and bearing measurements. From t_k= 74 mins to t_k = 84 mins the target is heading 360 degs. The filter lags behind the target during this segment. The noisy bearing measurement from DIFAR 1 is less than the noise-free measurement eight times out of the ten measurements. Hence, the filter thinks the target is going slower. Due to the position of DIFAR 1, the y coordinate is sensitive to the bearing measurements from t_k = 90 mins to t_k = 125 mins. The filter detects the small "s" turn and tracks the target with an average error of approximately 360 yds. At t_k = 101 mins the bearing measurement from buoy 1 causes the predicted residual to exceed the adaptive gate three times and the filter is reinitialized. DIFAR 1's noisy bearing measurement for t_k = 101 mins is 15 degs greater than the noise-free measurement. The distance from the target's actual position to buoy 1 is 3250 yds. A 15 deg error at this distance is a very large error. After restarting, the filter locks on to the target at t_k = 108 mins with an error of 80 yds. At t_k =111 mins the bearing measurement is 13 degs less than the noise-free

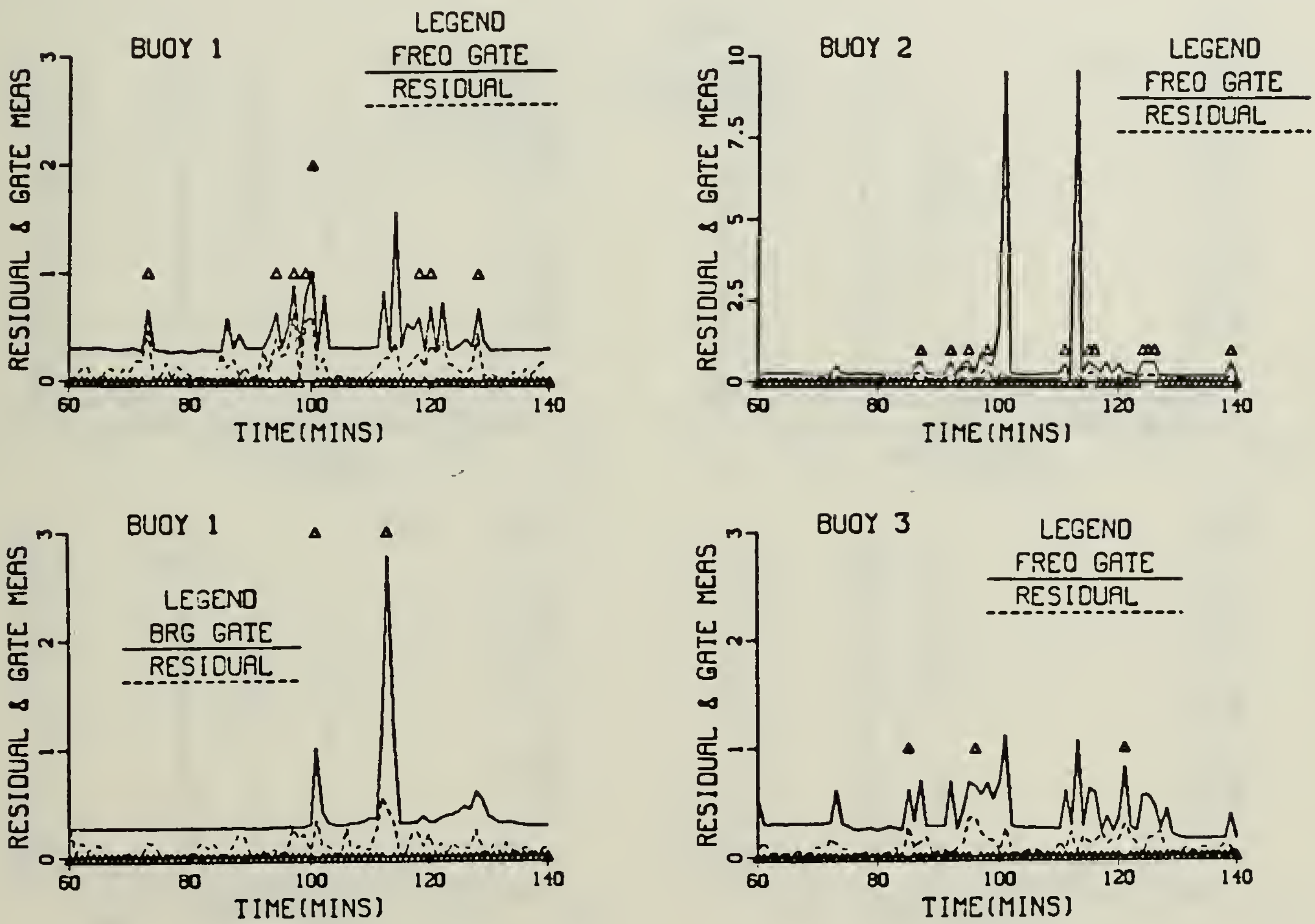

Figure 6-54. Scenario 4: Frequency and Bearing Measurement- Predicted Residual and Adaptive Gate

measurement and the filter's estimated position is southeast of the actual position by 770 yds. The filter reinitializes again at t_k = 113 mins. DIFAR 1's noisy bearing measurement is 10 degs greater than the noise-free measurement and the target is less than 2000 yds from buoy 1. After restarting the filter continues to track the target.

Figure (6-55) illustrates the position variances. Note the two peaks at t_k =101 mins and t_k = 113 mins, this is when the filter is reinitialized. From equation (3-57) the position components are reinitialize to $(0.5\ nm)^2$ or approximately 10^6 yds^2 .

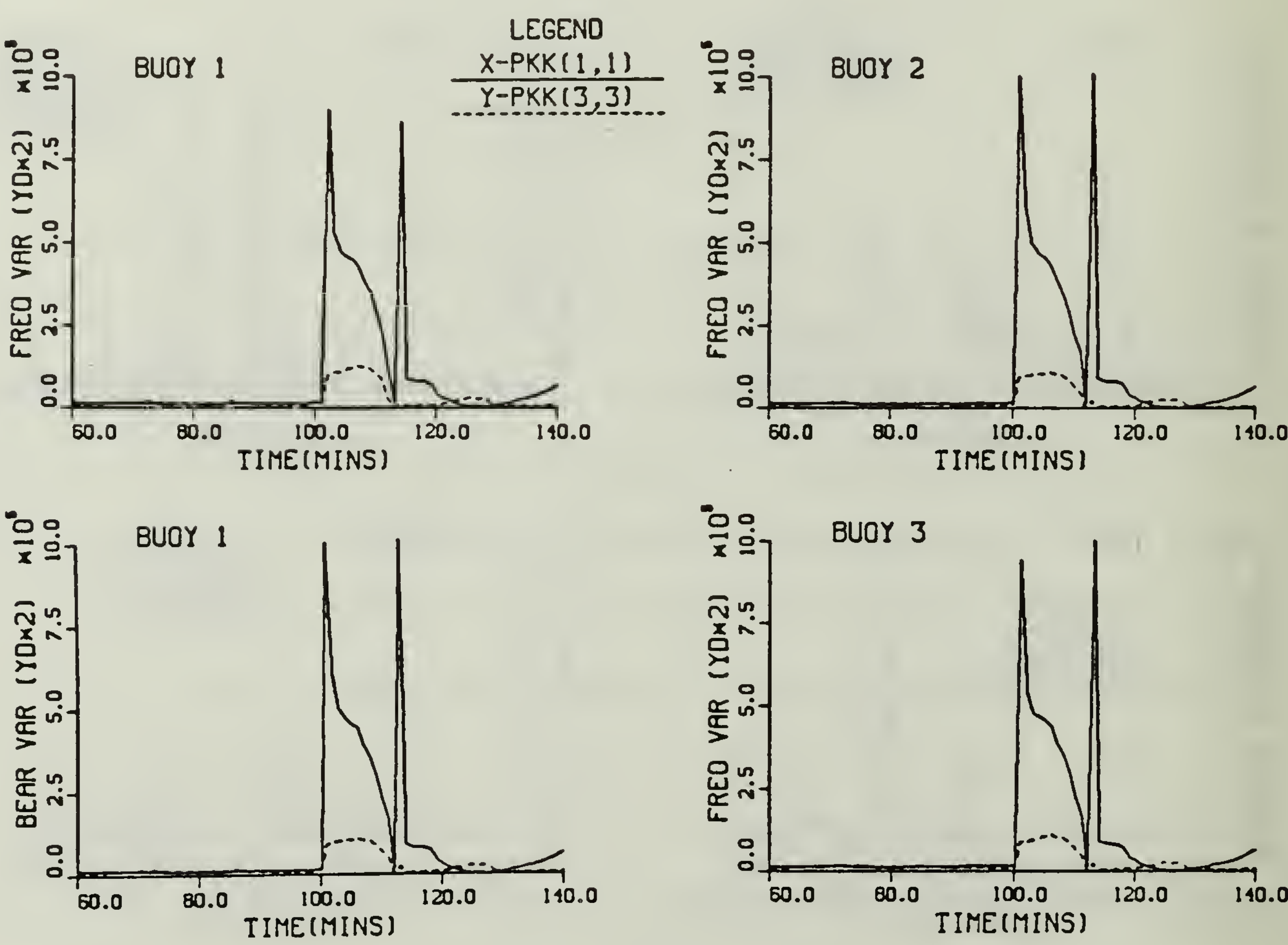

Figure 6-55. Scenario 4: Frequency and Bearing Measurement- Position Variances

The velocity variances are shown in Figure (6-56). The v_y variance for buoy 1 at t_k = 100 is 4.32 × 10^4 (yds/mins)^2. This is at the completion of the small "s" maneuver. The velocity components are reinitialized to (3kts)^2 or approximately 10^4 (yds/mins)^2 at t_k =101 mins and t_k = 113 mins. Figure (6-57) illustrates the frequency variances. Instead of reinitializing the frequency component to 1 hz^2 as indicated in equation (3-57) the algorithm used 10 hz^2 for this scenario. The Kalman gains for the position, velocity, and frequency components are shown in Figures (6-58) - (6-60).

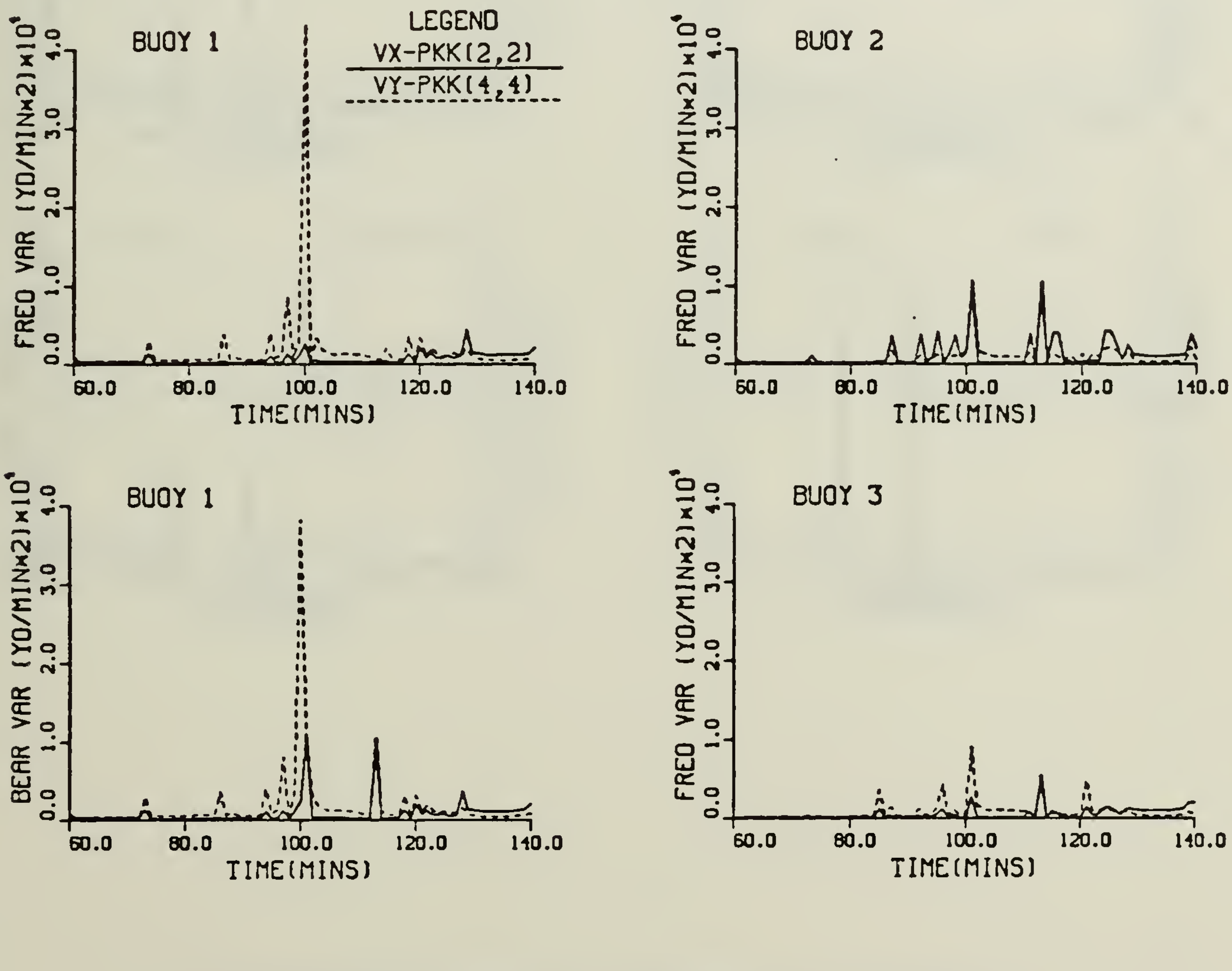

Figure 6-56. Scenario 4: Frequency and Bearing Measurement - Velocity Variances

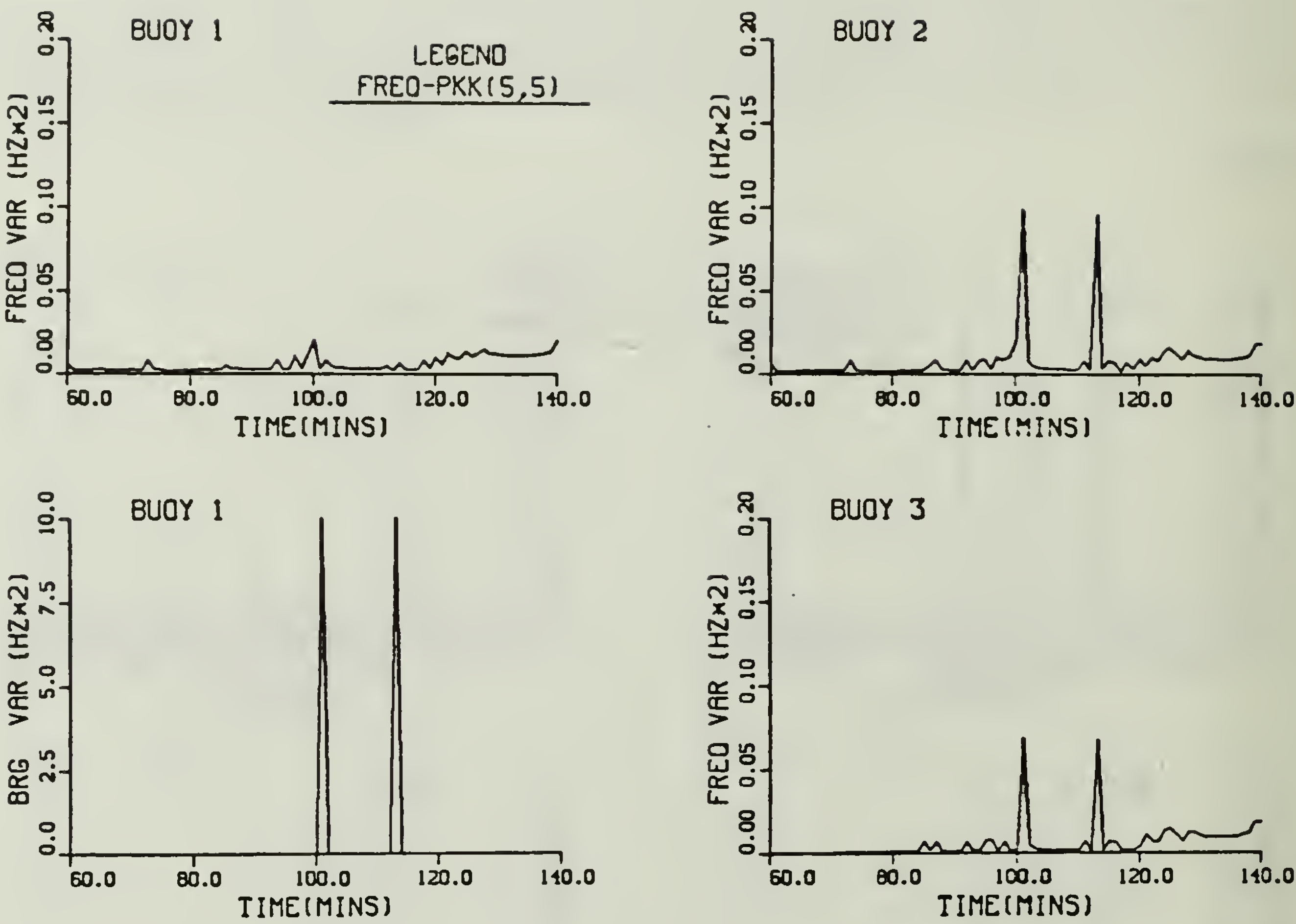

Figure 6-57. Scenario 4: Frequency and Bearing
Measurement-Frequency Component

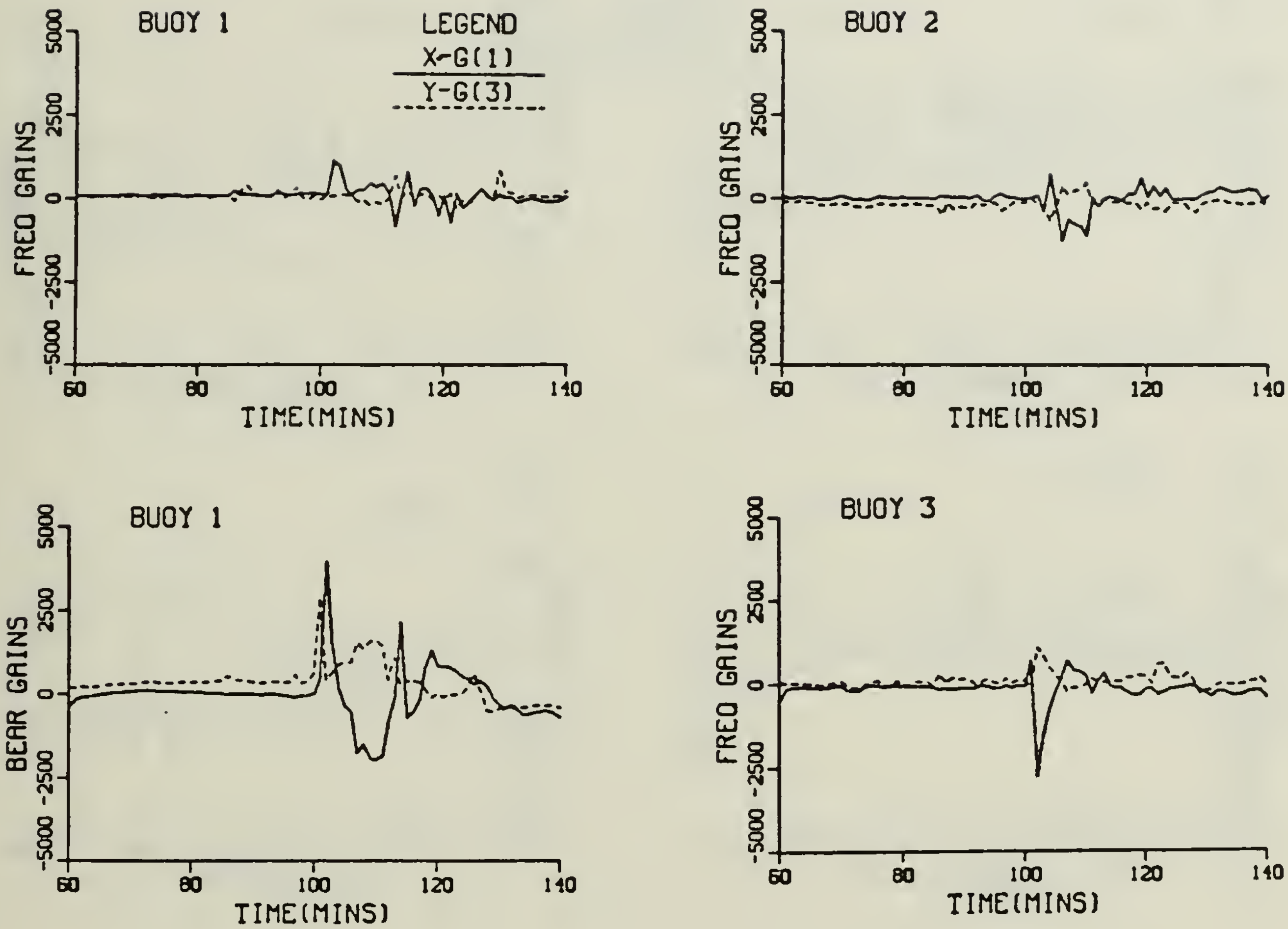

Figure 6-58. Scenario 4: Frequency and Bearing Measurement-Kalman Gains for Position Components

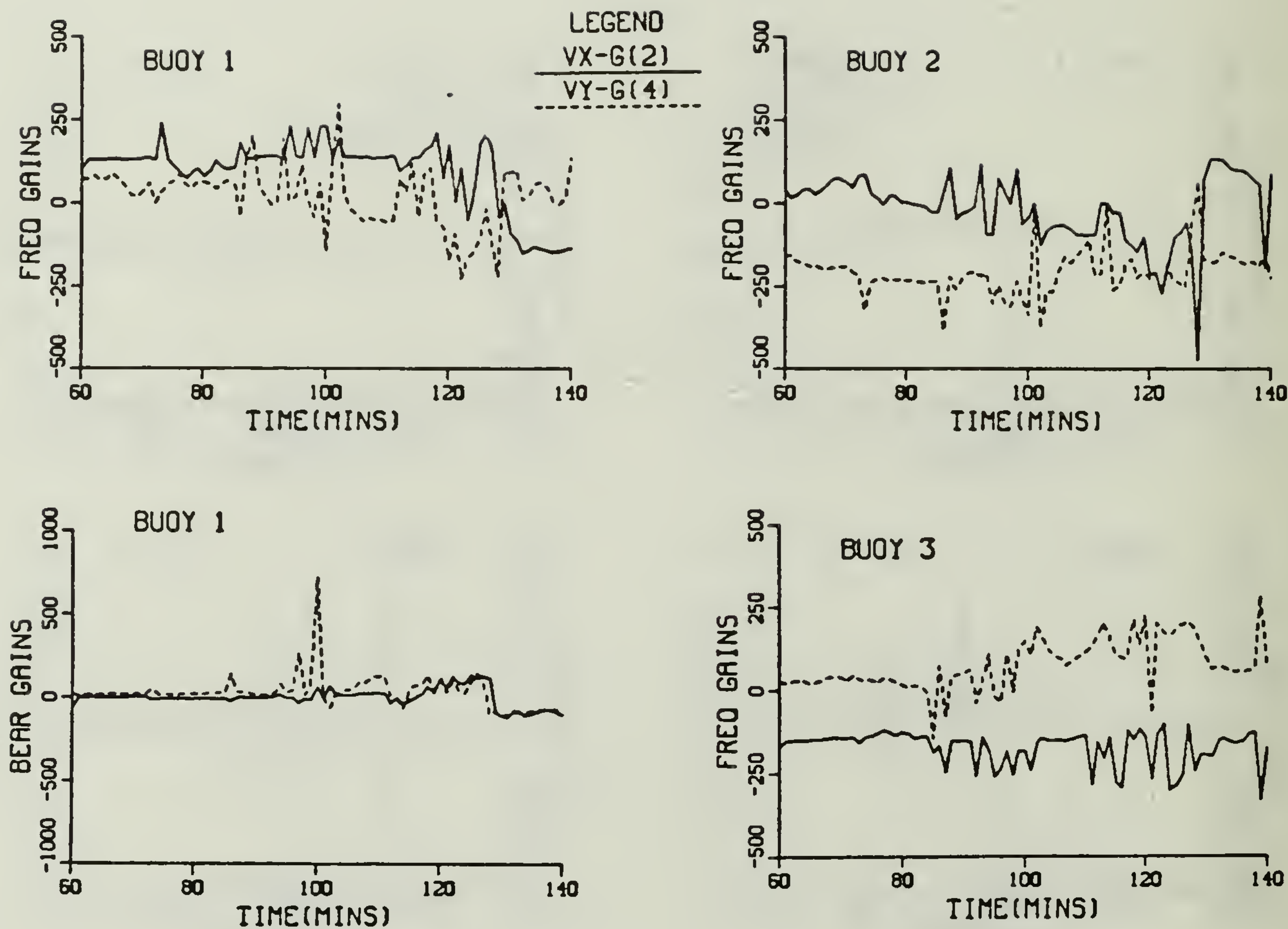

Figure 6-59. Scenario 4: Frequency and Bearing Measurement-Kalman Gains for Velocity Components

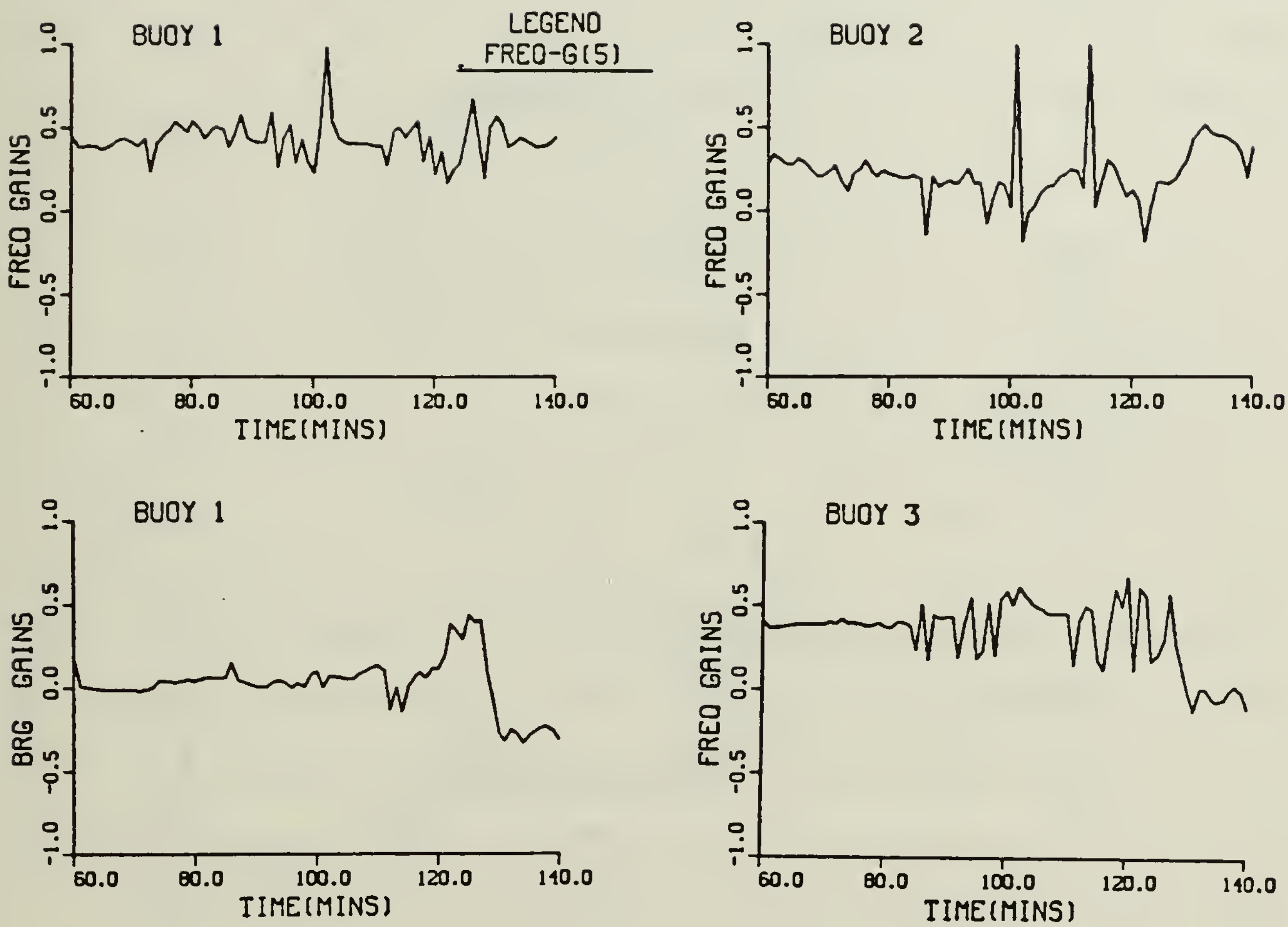

Figure 6-60. Scenario 4: Frequency and Bearing Measurement - Kalman Gain for Frequency Component

F. INITIALIZATION EXAMPLES

A couple examples of the initialization process are illustrated in Scenario 5 - Simulations 1 and 2. Scenario 5 is a three buoy scenario where one of the buoys is a DIFAR. Scenario 5 is similar to Scenario 3, but Scenario 5 uses set 1 measurement noise standard deviations from Table (5-2) and Table (5-3). Simulation 1, Figure (6-61) shows a geographic plot of the sonobuoy pattern, target's track and the filter's estimated track. The a priori information follows:

x_e = 10 nm σ_p = 0.5 nm
y_e = 12 nm
v_e = 5 kts σ_{ve} = 3 kts
h_e = 180 degs σ_{he} = 10 degs
f_o = 300 hz σ_{fe} = 1 hz
v_p = 4860 ft/sec

The error ellipsoids shown are due to the covariance of error matrix position components $P_{11}(k|k)$, $P_{33}(k|k)$, and $P_{13}(k|k)$. The large dashed circle is the error ellipsoid from DIFAR 1's frequency measurement. Similarly, the horizontal ellipse is due to DIFAR 1's bearing measurement. LOFAR 2 and 3 contribute very little to reducing the error ellipsoid.

The filter takes approximately 5 mins. to lock on to the target with and error of less than 500 yards. This is partial due to the inaccuracies in the initial estimated covariance of error matrix $P(0|-1)$.

Simulation 2, Figure (6-62), is the same as Simulation 1 except for the target's initial estimated position, speed, and course has been changed to

x_e = 9.75 nm
y_e = 11.75 nm
v_e = 6 kts
h_e = 185 degs

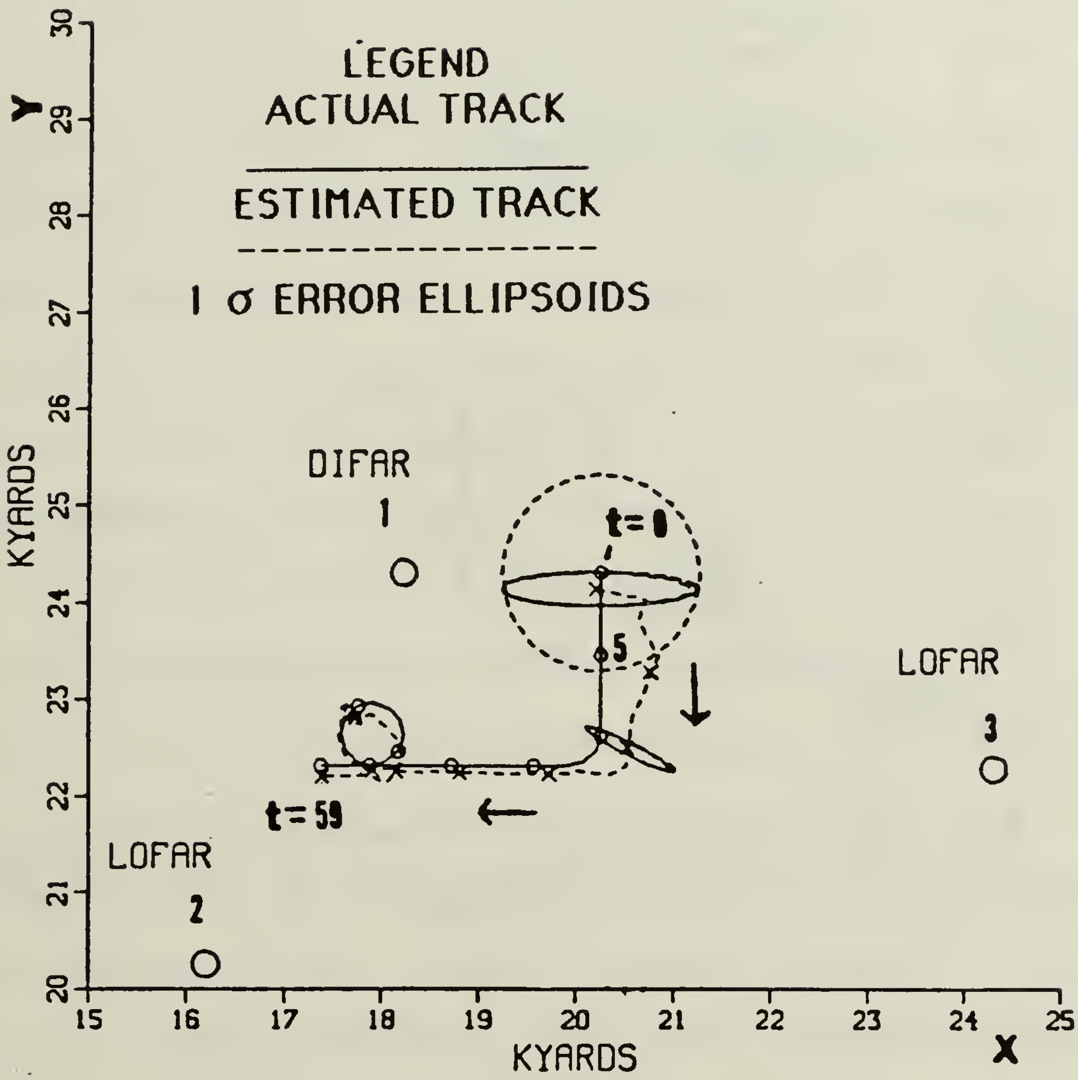

Figure 6-61. Scenario 5-Simulation 1: Geographic Plot-Noise, $Q'_1(k)$, and Adaptive Control Applied

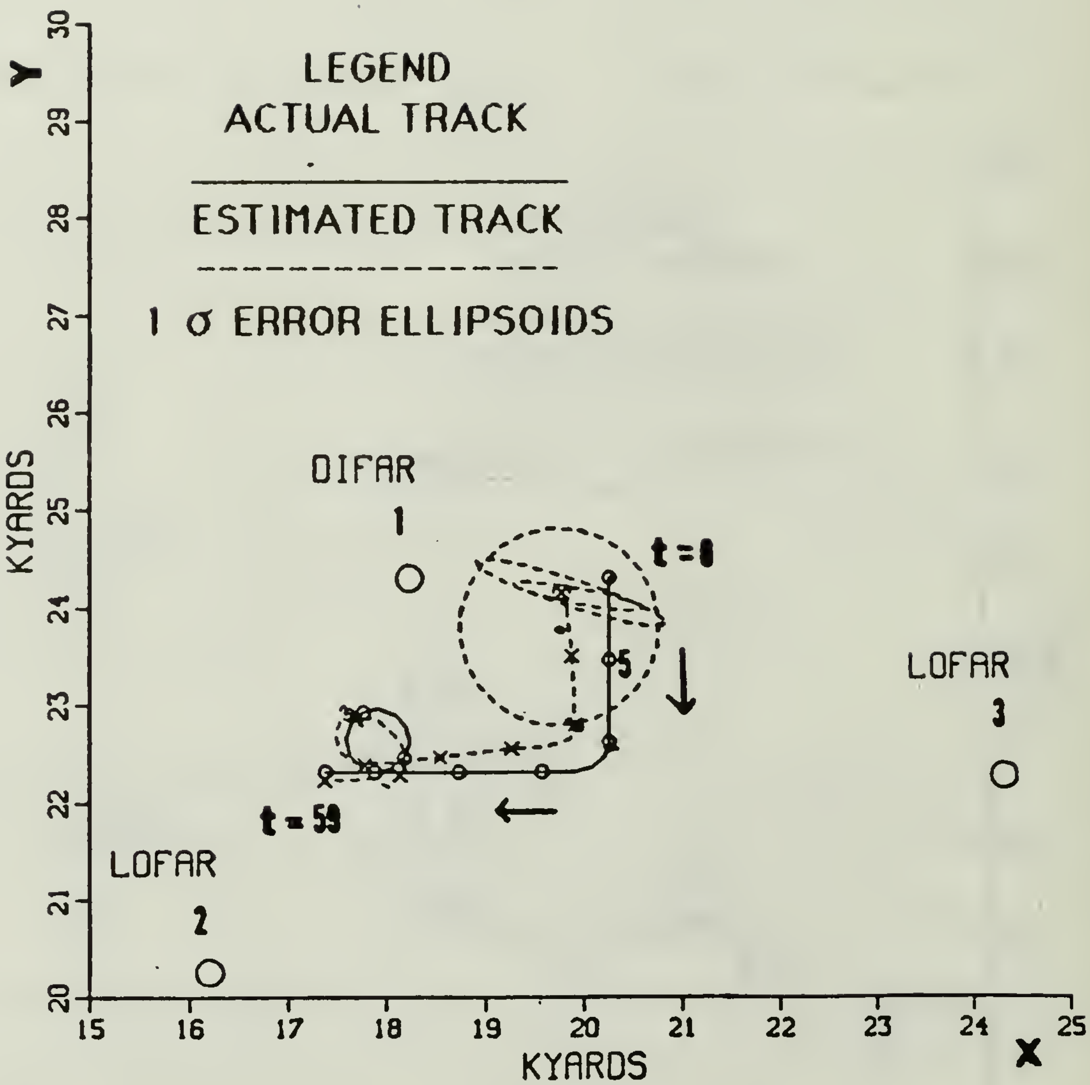

Figure 6-62 Scenario 5-Simulation 2: Geographic Plot-Noise, $Q'_1(k)$, and Adaptive Control Applied

Also, the a priori estimated covariance of error matrix has been changed to include some off diagonal terms

$$P(0|-1) = \begin{bmatrix} \sigma_p^2 & \sigma_p \cdot \sigma_{vxe} & 0 & 0 & 0 \\ \sigma_p \cdot \sigma_{vxe} & \sigma_{vxe}^2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_p^2 & \sigma_p \cdot \sigma_{vye} & 0 \\ 0 & 0 & \sigma_p \cdot \sigma_{vye} & \sigma_{vye}^2 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{fe}^2 \end{bmatrix} \qquad (6\text{-}3)$$

The black dot in the center of the dashed circle indicates the target's initial estimated position from DIFAR 1 frequency measurement. This position is approximately 0.35 nm (720 yds) southwest of the target's actual position. The initial estimated speed is 1 kt. faster than the actual speed and the initial estimated course is off by 5 degs., but all of estimates are within their standard deviations. The estimated position $\hat{\underline{x}}(0|0)$ is less than 500 yards from the actual position. The filter establishes a consistent track estimate that reduces the position error with each iteration. The error ellipsoids decrease very rapidly, due to the addition of the off diagonal terms in $P(0|-1)$.

Several other simulations were run with different combinations of the a priori information, $\hat{\underline{x}}(0|-1)$ values, buoy types, buoy position, and number of buoys. As expected the better the a priori information is the quicker the filter will lock on to target's track.

II. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

The goal of this thesis was to develop an operational program for tracking submarines using Extended Kalman Filtering techniques. Numerous simulation tests were conducted and modifications made during the study. Only a small representative sampling of the results have been presented here. The results demonstrated that the filter is capable of tracking an maneuvering target. The accuracy of the tracking is highly dependent on the following:

1. The type and number of sonobuoys in the pattern.
2. The amount of measurement noise.
3. The a priori information.

The predicted residuals of the filter were found to be a good indication of the filter's performance. The $3\sigma_z$ adaptive gate demonstrated a satisfactory method to track a maneuvering target. Through, the adaptive control method we implemented may have a defect. The value of the state excitation matrix Q(k) is based on only one predicted residual and therefore has little statistical significance. A simply way to remedy this is by replacing the one predicted residual by the sample mean of N predicted residuals. The disadvantages of using the sample mean is the requirement for more computer memory and the cost (time) of calculating the mean.

The following practical problems were not tested in the simulations. One problem that is encountered in an actual tracking scenario is sonobuoy fade out. Since the measurements will often be taking place at the limit of the range of signal reception, the possibility of a buoy becoming available for a few measurements, only to drop out either temporarily or permanently as the signal weakens, must be taken into account. A second practical problem is due to the measurement signal processors. Measurements are likely to be available at random times. And the third problem is bad measurements. How do we handle a bad measurement? For instance, the bearing measurement is 90 degs from the expected measurement. The tracking algorithm is design to allow for sonobuoy fade out and asynchronous measurements. As indicated in Subsection V.C, the observation packet consists of the following:

1. the time,
2. identification of the sonobuoy having contact
3. the measurement, and
4. its standard deviation.

For a given time only those buoys that have measurements are used as input data. The current version of the tracking algorithm assumes that all measurements are good. The predicted residual can be used to detect a bad measurement. A method needs to be developed to determined and discard the bad measurements.

B. RECOMMENDATIONS

This study is by no means complete. The program needs to be tested with realistic data. The assumptions (i.e., measurement noise, standard deviations, target accelarations to name a few) made were to allow the simulations to create a pseudo-realistic tracking environment. The tracking algorithm will need to be modified to accept the other data. Further study is needed in the following areas:

1. Implement the tracking program on microcomputer. If algorithm works reasonably well on a microcomputer then it could be implemented on the an aircraft computer

2. Test the tracking algorithm with an existing initialization algorithm Or develop an initialization algorithm that provides the tracking algorithm with the a priori information.

3. Conduct a comparison of this algorithm with other existing algorithms. Or develop a filter using polar coordinates, modified polar coordinates or other state vector and compare it to the tracking algorithm. Note in polar coordinates both the system model and measurement model are nonlinear.

4. Include the depth of the target as a state variable.

5. Apply the tracking algorithm to a data fusion problem. For instance, at the same frequency we have measurements to a submarine and a surface target from the same buoy. Or in addition to the acoustic information, their is also nonacoustic information on the target's location.

6. Will including higher order terms in the linearization bring about a better solution given the cost (time) of calculating them?

7. The algorithm needs to be made interactive. There should be a means for the operator to insert different frequencies along with its confidence level or estimated variance into the algorithm. The operator may have a several dominant frequencies for a target. Also,there should be a means to insert into the algorithm the velocity of sound for the water.

APPENDIX A: FORTRAN PROGRAM FOR THE TARGET AND INPUT DATA

The FORTRAN program used to implement the target's track is presented in this appendix. The program is written in FORTRAN 77 and executed on the IBM 3033 located at the Naval Postgraduate, Monterey, California. The program generates an output file that is used as input file for the tracking program presented in Appendix B.

The sonobuoy pattern data set is, also, presented in this appendix. It is used as input file to the tracking program.

```
C     *** PURPOSE ***
C     PROGRAM COMPUTES THE TARGET TRACK
C  ******************************************
C     THE OUTPUT IS IN FILE DEVICE 3
C  ******************************************
C
C     *** VARIABLE DEFINITIONS ***
C     DT        =    SAMPLE INTERVAL IN SECS.
C     HDG       =    MATRIX OF TARGET'S HEADINGS
C     I         =    COUNTER
C     II        =    I-1
C     ITIME     =    INTERVAL TO TIME TURNS ITIME =12,
C                    EX. ITIME X DT X TRATE = DEGREES OF TURN
C                         12 X 15 SEC X 0.5 DEG/SEC = 90 DEGS
C     J         =    COUNTER
C     PI180     =    PI X 180 DEGREES
C     T         =    TIME USED IN SECONDS AND CONVERTED TO MINUTES FOR
C                    OUTPUT
C     TACC      =    HORIZ ACCELERATION- YARDS/SEC**2
C     TRATE     =    TURN RATE-INPUT IN DEG/SEC, CONVERTED TO RAD/SEC
C                    + CLOCKWISE,- COUNTER CLOCKWISE
C     TURN      =    SUBROUTINE TO CALCULATE TURNS
C     VEL       =    MATRIX OF TARGET'S VELOCITIES-KNOTS AND YARDS/SEC
C     VX        =    X COMPONENT OF VEL
C     VY        =    Y COMPONENT OF VEL
C     X         =    X COMPONENT-USED IN NM AND YDS
C     Y         =    Y COMPONENT-USED IN NM AND YDS
C
C     *** VARIABLE DECLARATIONS ***
C
```

```
      REAL HDG
      DIMENSION T(2000),VEL(2000),TRATE(2000),X(2000),Y(2000),HDG(2000)
C
C     INITIALIZE TARGET DATA
      PI180=0.0174533
      X(1)=10.
      Y(1)=12.
      VEL(1)=5.0
      HDG(1)=180.0
      TRATE(1)=0.5
      DT=15.
      TACC=2.
      T(1)=0.
      ITIME=15
C     CONVERT DIST TO YARDS
      X(1)=X(1)*2025.3667
      Y(1)=Y(1)*2025.3667
C     CONVERT TURN RATE FROM DEG/SEC TO RAD/SEC
      TRATE(1)=TRATE(1)*PI180
C     CONVERT VEL FROM KTS TO YARDS/SEC
      VEL(1)=VEL(1)*0.562605
C     CONVERT HEADING TO RAD/SEC
      HDG(1)=HDG(1)*PI180
C     CONVERT HORIZ ACC FROM YARDS/SEC**2 TO YARDS/MIN**2
C     TDATA(8)=TDATA(8)*3600.0
C      WRITE(6,1000) T,X,Y,VEL,HDG
C      WRITE(3,1000) T,X,Y,VEL,HDG
       I=1
34    IF(T(I).EQ.600.) THEN
C     ITIME=12 90 DEG AT 0.5 DEG/SEC
C     ITIME=24 180 DEG AT 0.5 DEG/SEC
C     ITIME=48 360 DEG AT 0.5 DEG/SEC
      ITIME=12
C     90 DEG -STARBOARD TURN
      CALL TURN(X,Y,TRATE,VEL,HDG,DT,ITIME,T,I)
```

```
      GO TO 800
      ELSEIF(T(I).EQ.1500.) THEN
C     LARGE 360 DEG STARBOARD TURN
      ITIME=48
      CALL TURN(X,Y,TRATE,VEL,HDG,DT,ITIME,T,I)
      GO TO 800
      ELSEIF(T(I).EQ.3000.) THEN
C     SMALL 360 DEG STARBOARD TURN
      ITIME=24
      TRATE(I)=1*PI180
      CALL TURN(X,Y,TRATE,VEL,HDG,DT,ITIME,T,I)
      GO TO 800
      ELSEIF(T(I).EQ.4200.) THEN
C     90 DEG STARBOARD TURN
      ITIME=12
      TRATE(I)=0.5*PI180
      CALL TURN(X,Y,TRATE,VEL,HDG,DT,ITIME,T,I)
      GO TO 800
      ELSEIF(T(I).EQ.5040.) THEN
C     90 DEG STARBOARD TURN
      ITIME=12
      TRATE(I)=0.5*PI180
      CALL TURN(X,Y,TRATE,VEL,HDG,DT,ITIME,T,I)
      GO TO 800
      ELSEIF(T(I).EQ.5460.) THEN
C     90 DEG STARBOARD TURN, BEGINNING OF SMALL S TURN
      ITIME=6
      TRATE(I)=-1*PI180
      CALL TURN(X,Y,TRATE,VEL,HDG,DT,ITIME,T,I)
      GO TO 800
      ELSEIF(T(I).EQ.5550.) THEN
C     180 DEG STARBOARD TURN
      ITIME=12
      TRATE(I)=1*PI180
      CALL TURN(X,Y,TRATE,VEL,HDG,DT,ITIME,T,I)
```

```
      GO TO 800
      ELSEIF(T(I).EQ.5760.) THEN
C     180 DEG PORT TURN
      ITIME=12
      TRATE(I)=-1*PI180
      CALL TURN(X,Y,TRATE,VEL,HDG,DT,ITIME,T,I)
      GO TO 800
      ELSEIF(T(I).EQ.5940.) THEN
C     90 DEG STARBOARD TURN, ENDING OF SMALL S TURN
      ITIME=6
      TRATE(I)=1.0*PI180
      CALL TURN(X,Y,TRATE,VEL,HDG,DT,ITIME,T,I)
      GO TO 800
      ELSEIF(T(I).EQ.6540.) THEN
C     90 DEG STARBOARD TURN, BEGINNING OF LARGE S TURN
      ITIME=12
      TRATE(I)=0.5*PI180
      CALL TURN(X,Y,TRATE,VEL,HDG,DT,ITIME,T,I)
      GO TO 800
      ELSEIF(T(I).EQ.6720.) THEN
C     180 DEG PORT TURN
      ITIME=24
      TRATE(I)=-0.5*PI180
      CALL TURN(X,Y,TRATE,VEL,HDG,DT,ITIME,T,I)
      GO TO 800
      ELSEIF(T(I).EQ.7140.) THEN
C     180 DEG STARBOARD TURN
      ITIME=24
      TRATE(I)=0.5*PI180
      CALL TURN(X,Y,TRATE,VEL,HDG,DT,ITIME,T,I)
      GO TO 800
      ELSEIF(T(I).EQ.7500.) THEN
C     90 DEG PORT TURN
      ITIME=12
      TRATE(I)=-0.5*PI180
```

```
      CALL TURN(X,Y,TRATE,VEL,HDG,DT,ITIME,T,I)
      GO TO 800
      ELSEIF(T(I).EQ.8250.) THEN
C     90 DEG STARBOARD TURN
      ITIME=12
      TRATE(I)=0.5*PI180
      CALL TURN(X,Y,TRATE,VEL,HDG,DT,ITIME,T,I)
      GO TO 800
      ENDIF
      TRATE(I+1)=TRATE(I)
      HDG(I+1)=HDG(I)
      VEL(I+1)=VEL(I)
      VX=VEL(I)*SIN(HDG(I+1))
      VY=VEL(I)*COS(HDG(I+1))
      X(I+1)=X(I)+VX*DT
      Y(I+1)=Y(I)+VY*DT
      T(I+1)=T(I)+DT
800   I=I+1
      IF(I.LT.564) GO TO 34
      II=I-1
      DO 20 J=1,II,4
C     CONVERT HEADING FROM RADS TO DEGS.
       HDG(J)=HDG(J)/PI180
C     CONVERT VELOCITY FROM YARDS/SEC TO KTS
       VEL(J)=VEL(J)/0.5626
C     CONVERT X AND Y TO NM
      X(J)=X(J)/2025.3667
      Y(J)=Y(J)/2025.3667
      T(J)=T(J)/60.
      WRITE(6,1500) T(J),X(J),Y(J),VEL(J),HDG(J)
      WRITE(3,1500) T(J),X(J),Y(J),VEL(J),HDG(J)
20    CONTINUE
1000  FORMAT (F8.2,2(4X,F11.4),4X,F7.2,4X,F7.2)
100   FORMAT (F10.5,2X,F8.2)
1500  FORMAT (F8.2,4X,F11.4,4X,F11.4,4X,F11.4,4X,F7.2)
```

```
200   FORMAT (/,2X,'TIME SLOT = ',I4)
2000  FORMAT (/,5X,F8.2,5X,F11.5,5X,F11.5)
      STOP
      END
C
C
C
C
      SUBROUTINE TURN(X,Y,TRATE,VEL,HDG,DT,ITIME,T,I)
C
C     *** PURPOSE ***
C     CALCULATE X AND Y COMPONENTS IN TURNS
C
C     *** VARIABLE DEFINITIONS ***
C
C     DT       =    SAMPLE INTERVAL IN SECS.
C     HDG      =    MATRIX OF TARGET'S HEADINGS
C     I        =    COUNTER
C     ISTOP    =    TIME INTERVAL TURNSTOPS
C     ITIME    =    INTERVAL TO TIME TURNS,ITIME =12,
C                   EX. ITIME X DT X TRATE = DEGREES OF TURN
C                        12 X 15 SEC X 0.5 DEG/SEC = 90 DEGS
C     J        =    COUNTER
C     T        =    TIME USED IN SECONDS AND CONVERTED TO MINUTES FOR
C                   OUTPUT
C     TRATE    =    TURN RATE-INPUT IN DEG/SEC, CONVERTED TO RAD/SEC
C                   + CLOCKWISE,- COUNTER CLOCKWISE
C     TWOPI    =    2 X PI
C     VEL      =    MATRIX OF TARGET'S VELOCITIES-KNOTS AND YARDS/SEC
C     VX       =    X COMPONENT OF VEL
C     VY       =    Y COMPONENT OF VEL
C     X        =    X COMPONENT-USED IN NM AND YDS
C     Y        =    Y COMPONENT-USED IN NM AND YDS
C
C     *** VARIABLE DECLARATIONS ***
```

```
C
      REAL HDG
      DIMENSION T(2000),VEL(2000)
      DIMENSION TRATE(2000),X(2000),Y(2000),HDG(2000)
      TWOPI=6.2831853
      ISTOP=I+ITIME-1
      DO 10 J=I,ISTOP
      TRATE(J+1)=TRATE(J)
      HDG(J+1)=HDG(J)+TRATE(J)*DT
      IF(HDG(J+1).GT.TWOPI) HDG(J+1)=HDG(J+1)-TWOPI
      VEL(J+1)=VEL(J)
      VX=VEL(J)*SIN(HDG(J+1))
      VY=VEL(J)*COS(HDG(J+1))
      X(J+1)=X(J)+VX*DT
      Y(J+1)=Y(J)+VY*DT
      T(J+1)=T(J)+DT
10    CONTINUE
      I=ISTOP
      RETURN
      END
```

```
c    output from target track, input to tracking algorithm.
c    file device 3
c    time         x comp          y comp          velocity    heading
c    mins         nm              nm               kts          degs
      0.00        10.0000         12.0000          5.0000      180.00
      1.00        10.0000         11.9167          5.0000      180.00
      2.00        10.0000         11.8333          5.0000      180.00
      3.00        10.0000         11.7500          5.0000      180.00
      4.00        10.0000         11.6667          5.0000      180.00
      5.00        10.0000         11.5833          5.0000      180.00
      6.00        10.0000         11.5000          5.0000      180.00
      7.00        10.0000         11.4167          5.0000      180.00
      8.00        10.0000         11.3333          5.0000      180.00
      9.00        10.0000         11.2500          5.0000      180.00
     10.00        10.0000         11.1667          5.0000      180.00
     11.00         9.9735         11.0886          5.0000      210.00
     12.00         9.9115         11.0342          5.0000      240.00
     13.00         9.8306         11.0181          5.0000      270.00
     14.00         9.7473         11.0181          5.0000      270.00
     15.00         9.6640         11.0181          5.0000      270.00
     16.00         9.5806         11.0181          5.0000      270.00
     17.00         9.4973         11.0181          5.0000      270.00
     18.00         9.4140         11.0181          5.0000      270.00
     19.00         9.3306         11.0181          5.0000      270.00
     20.00         9.2473         11.0181          5.0000      270.00
     21.00         9.1640         11.0181          5.0000      270.00
     22.00         9.0806         11.0181          5.0000      270.00
     23.00         8.9973         11.0181          5.0000      270.00
     24.00         8.9140         11.0181          5.0000      270.00
     25.00         8.8306         11.0181          5.0000      270.00
```

```
c sample sonobuoy pattern input file
c for time t=0 to t= 59
c this pattern was used in scenario 3
c a priori information used for p(k/k-1) and x(k/k-1)
c
c input data follows:
c number of buoys in pattern
c buoy   bflag  x comp    y comp
c type          in nm     in nm
c
  3
  DIFAR   1.0  9.0        12.0
  LOFAR   2.0  8.0        10.0
  LOFAR   2.0  12.0       11.0
c
c
c   sample sonobuoy pattern input file
c   for time t=60 to t=140
c   using x(k/k) and P(k/k) from previous run
c   pattern and data is similar to that used in scenario 4
c
c   input data follows:
c
c   number of buoys in pattern
c   buoy  bflag  x comp    y comp
c   type         in nm     in nm
c   time
c   p(k/k) = p(59/59) is a 5 x 5 matrix
c   x(k/k) = x(59/59) is a 1 x 5 matrix
c
```

```
 3
 DIFAR   1.0  9.0        12.0
 LOFAR   2.0  8.0        10.0
 LOFAR   2.0  5.5        11.5
 5.9000E+01
 1.7942E+04  2.0082E+03  3.9967E+02  3.8684E+01 -5.9835E+00
 2.0082E+03  6.9507E+02 -3.2991E+02 -1.2371E+02 -1.7275E+00
 3.9968E+02 -3.2991E+02  7.5759E+03  4.3116E+02  7.8573E-01
 3.8682E+01 -1.2371E+02  4.3116E+02  1.5582E+02  3.1255E-01
-5.9835E+00 -1.7275E+00  7.8573E-01  3.1255E-01  4.8693E-03
 1.5318E+04 -1.4382E+02  2.2142E+04 -3.6561E+00  2.9992E+02
```

APPENDIX B: FORTRAN PROGRAM FOR FILTER

The FORTRAN program used to implement the tracking program is presented in this appendix. The program is written in FORTRAN 77 and executed on the IBM 3033 located at the Naval Postgraduate, Monterey, California. The program generates an output file that is used as input file for the graphics program presented in Appendix C. The executive routine used to run the tracking program is included on pages 149 and 150.

```
c   exec's used to run the tracking algorithm
c
c   example of temp exec used to reserve space on disk b
c   as indicated in buoy exec file device 7,8 10,12,18, and 19
c   are on disk b.  in order to see disk b type "flist * * b"
c
c temp exec
CP DEFINE T3350 AS 193 20
FORMAT 193 B
c
c   example of buoy exec
c   filedef 2 is sonobouy pattern input file
c   filedef 3 is the target track input file
c   filedef 4 is the graphic input for geographic and enlarged
c   geographic plots
c   filedef 7 is the graphic input for the kalman gains plots
c   filedef 8 is the output of the tracking algorithm. Each
c   matrices and calculation can be checked if desired.
c   filedef 9 is the graphic input for the variance plots
c   filedef 10 is the track position data and the filter's estimated
c   position output (in yards)
c   filedef 12 is the graphic input for the kalman position
c   variances and the experimental position variances.
c   filedef 18 is the graphic input for the predicted residual
c   and adaptive gating plots
c   filedef 19 is the error ellipsoid data
c
&TRACE ON
CP TERMINAL LINES 80
FILEDEF 02 DISK BUOY2 DATA A
```

```
FILEDEF 03 DISK TGT INPUT A
FILEDEF 04 DISK FILE FT04F001 (PERM
FILEDEF 07 DISK FILE FT07F001 B
FILEDEF 08 DISK OUTPUT DATA B
FILEDEF 09 DISK FILE FT09F001 (PERM
FILEDEF 10 DISK TRACK DATA B
FILEDEF 12 DISK FILE FT12F001 B
FILEDEF 18 DISK FILE FT18F001 B
FILEDEF 19 DISK ELLIP DATA B
FILEDEF 06 TERM
 LOAD BUOY (START
```

```
C     *** PURPOSE ***
C     THIS IS A MULTISENSOR TRACKING PROGRAM. PROGRAM USES EXTENDED
C     KALMAN FILTER TECHNIQUES TO TRACK A MANEUVERING TARGET BASED
C     ON NOISY PASSIVE BEARING AND DOPPLER SHIFTED FREQUENCY
C     MEASUREMENTS.
C
C     IN ORDER TO RUN PROGRAM SEE THE TWO EXEC FILES DEFINE
C     ON THE PREVIOUS PAGES. TO RUN THE PROGRAM USE THE
C     FOLLOWING STEPS.
C     1. RESERVE SPACE ON B DISK BY USING TEMP EXEC.
C     2. COMPILE THIS PROGRAM.
C     3. EXECUTE BUOY EXEC.
C
C     *** VARIABLE DEFINITIONS ***
C     AFLAG       =    ADAPTIVE GATING FLAG; 0.0-OFF,1.0-ON.
C     BEAR        =    SUBROUTINE TO CALC BEARING MEASUREMENT
C     BFLAG       =    DEFINES THE BUOY TYPE; 1.0- DIFAR, 2.0-LOFAR
C     BMEAS       =    SUBROUTINE TO CALC MEASUREMENT MATRIX H(K)
C                      FOR THE BEARING MEASUREMENT
C     BRG         =    BEARING IN RADS
C     BRKKM1      =    PREDICTED BEARING MEAS. IN RADS,BRG(K/K-1)
C     BTYPE       =    BUOY TYPE, DIFAR OR LOFAR
C     D           =    DISTANCE OR RANGE, USED IN NM AND YARDS
C     DBKKM1      =    PREDICTED BEARING MEAS IN DEGS, DBRG(K/K-1)
C     DBRG        =    BEARING IN DEGS
C     DOPP        =    SUBROUTINE TO CALC DOPPLER FREQUENCY
C     DSEED       =    NUMBER USED IN PSEUDO RANDOM NUMBER GENERATOR
C     DT          =    TIME DIFFERENCE,T(K)- T(K-1)
C     E           =    PREDICTED RESIDUAL
C     EXT         =    ESTIMATED X COMP, XKK(1)
```

```
C     EYT        =    ESTIMATED Y COMP, XKK(3)
C     FD         =    ARRAY OF DOPP FREQS, ONE FOR EACH FREQ MEAS
C     FDKKM1     =    PREDICTED FREQ RESIDUAL FD(K/K-1)
C     FDOPP      =    THE DOPP FREQ, OUTPUT OF DOPP SUBROUTINE
C     FLAG       =    INDICATES MORE MEASUREMENTS IN THE TIME INTERVAL
C                     1.0-REMAIN IN SAME TIME SLOT,0.0-CAN READ NEXT
C                     TIME
C     FMEAS      =    SUBROUTINE TO CALC MEASUREMENT MATRIX H(K)
C                     FOR FREQ MEASUREMENT.
C     G          =    KALMAN GAINS
C     GAIN       =    SUBROUTINE TO CALC KALMAN EQUATIONS
C     GAM        =    GAMMA MATRIX, STATE FORCING MATRIX (5 X 3)
C     GAMT       =    TRANSPOSE OF GAMMA
C     GATE       =    H(K)*P(K/K-1)*H(K)T
C     GATE3      =    3 SIGMA ADAPTIVE GATE, 3*((GATE+R)**1/2)
C     GAUSS      =    SUBROUTINE TO CALC GAUSSIAN PSEUDO-RANDOM
C                     NUMBER GENERATOR
C     GE         =    G*E
C     GFLAG      =    USED WITH ADAPTIVE GATE,0-CONTINUE
C                     1-GATE3 EXCEEDED, 2- REINITIALIZE PROBLEM
C     H          =    MEASUREMENT MATRIX (5 X 1)
C     HE         =    A PRIORI HEADING INFORMATION
C     HT         =    TRANSPOSE OF H
C     I          =    COUNTER
C     INIT       =    INITIALIZATION FLAG TO DETERMINE WHICH P(0/-1)
C                     TO USE
C     J          =    COUNTER
C     K          =    ITERATION INTERVAL
C     KK         =    K-1, COUNTER
C     KOUNT      =    COUNTER TO DETERMINE THE TIMES THE ADAPT GATE
C                     IS EXCEEDED
C     L          =    ROW OR COLUMN OF MATRIX, 1 IN THIS CASE
C     LD         =    MAX NUMBER OF ROWS OR COLUMNS
C     M          =    ROW OR COLUMNS OF MATRIX
C     MD         =    MAX NUMBER OF ROWS OR COLUMNS
```

```
C     MFLAG        =     USE MANEUVERING EQUATIONS, Q1(K)
C     MZ           =     LAST TRACK VALUE OF DATA
C     MZZ          =     MZ-1
C     N            =     ROWS OR COLUMNS OF MATRIX
C     NB           =     COUNTER FOR BUOY NUMBER
C     NBUOY        =     TOTAL NUMBER OF BUOYS IN THE PATTERN
C     ND           =     MAX NUMBER OF ROWS AND COLUMNS
C     NFLAG        =     INDICATES  0.0- NO NOISE IS ADDED TO
C                        MEAS, 1.0 NOISE IS INCLUDED
C     NM           =     USED WITH EXP VAR, 10 TERM MOVING WINDOW
C                        NM=K-10
C     NNZ          =     NZ + 10
C     NZ           =     FIRST TRACK INTERVAL VALUE,USUALLY 1 OR 60
C     NZZ          =     NZ-1
C     OFF          =     ERROR IN POSITION, IN YARDS
C     PFLAG        =     0.0-CALC P(0/-1) FROM A PRIORI INFORMATION
C                        1.0-CALC P(K/K-1) FROM PREVIOUS SIMULATION
C     PHI          =     TRANSITION MATRIX (5 X 5)
C     PHIGAM       =     SUBROUTINE TO CALC PHI AND GAMMA
C     PHIT         =     TRANSPOSE OF PHI
C     PI           =     3.1415927
C     PI180        =     PI/180 DEGS
C     PKK          =     P(K/K) COVARIANCE OF ERROR MATRIX
C     PKKM1        =     P(K/K-1) PREDICTED COVARIANCE OF ERROR MATRIX
C     PLOTB        =     SUBROUTINE TO PLOT BEARING MEAS. GAINS AND
C                        COV OF ERROR
C     PLOTF        =     SUBROUTINE TO PLOT FREQ MEAS GAINS AND COV
C                        OF ERROR
C     PLOTGB       =     SUBROUTINE TO PLOT ADAPT GATE AND PREDICTED
C                        RESIDUAL FROM BEARING MEAS
C     PLOTGF       =     SUBROUTINE TO PLOT ADAPT GATE AND PREDICTED
C                        RESIDUAL FROM FREQ MEAS
C     PROD         =     SUBROUTINE TO MULTIPLY MATRICES
C     Q            =     STATE EXCITATION MATRIX (5 X 5)
C     QFIND        =     SUBROUTINE TO CALC Q
```

```
C     R           =    MEAS NOISE COVARIANCE OF ERROR MATRIX
C     RFLAG       =    CALCS THE NUMBER OF TIMES THE PROBLEM IS
C                      REINITIALIZED FOR A GIVEN MEASUREMENT
C     RFREQ       =    TRUE RADIATED FREQUENCY
C     RSTART      =    SUBROUTINE TO REINITIALIZE THE PROBLEM
C     SIGB        =    STANDARD DEVIATION FOR BEARING MEAS NOISE,RADS
C     SIGDG       =    STD DEV FOR BEARING MEAS NOISE,DEGS
C     SIGF        =    STD DEV FOR FREQ NOISE, HZ
C     SIGFE       =    STD DEV FOR A PRIORI FREQ,HZ
C     SIGHE       =    STD DEV FOR A PRIORI HEADING, DEGS
C     SIGMA       =    VECTOR OF 3 STD DEVS FROM QFIND
C     SIGPOS      =    STD DEV FOR A PRIORI POSITION, NM
C     SIGVE       =    STD DEV FOR A PRIORI VELOCITY, KTS
C     SIGVXE      =    STD DEV FOR VX COMP OF VE,KTS
C     SIGVYE      =    STD DEV FOR VY COMP OF VE,KTS
C     SUMX        =    SUMMATION FOR 10 TERM MOVING WINDOW FOR
C                      X COMP
C     SUMY        =    SUMMATION FOR 10 TERM MOVING WINDOW FOR
C                      Y COMP
C     TEMP        =    MATRIX USED FOR CALCULATIONS
C     TEMP1       =    MATRIX USED FOR CALCULATIONS

C     TEMP2       =    MATRIX USED FOR CALCULATIONS
C     TEMP3       =    MATRIX USED FOR CALCULATIONS
C     TGATE       =    COUNTS THE OF TIMES GATE3 IS EXCEEDED
C     THDG        =    TRUE HEADING OF TARGET
C     TIME        =    TIME OF MEAS
C     TWOPI       =    2*PI
C     UNIT        =    IDENTITY MATRIX
C     VARX        =    X COMP EXPERIMENTAL VARIANCE
C     VARY        =    Y COMP EXPERIMENTAL VARIANCE
C     VE          =    A PRIORI VELOCITY INFORMATION
C     VMEAS       =    MEAS VELOCITY IN DOPPLER EQUATION
C     VP          =    VELOCITY OF SOUND IN WATER, FT/SEC
C     VS          =    MATRIX OF THE VMEAS
```

```
C     VT          =    TARGET'S TRUE VELOCITY
C     VX          =    TARGET'S TRUE X COMP VELOCITY
C     VXE         =    A PRIORI X COMP VELOCITY INFORMATION
C     VY          =    TARGET'S TRUE Y COMP VELOCITY
C     VYE         =    A PRIORI Y COMP VELOCITY INFORMATION
C     W           =    RANDOM FORCING FUNCTION COVARIANCE MATRIX
C     XB          =    BUOY'S X COMP POSITION,NM AND YARDS
C     XD          =    DISTANCE IN TARGET'S X COMP AND BUOY'S X COMP,
C                      XT-XB
C     XDE         =    DISTANCE IN XE-XB
C     XE          =    A PRIORI X COMP INFORMATION
C     XKK         =    ESTIMATED STATE VECTOR, X(K/K)
C     XKKM1       =    PREDICTED STATE VECTOR, X(K/K-1)
C     XT          =    TARGET'S TRUE X COMP
C     YB          =    BUOY'S Y COMP POSITION, NM AND YARDS
C     YD          =    DISTANCE YT-YB
C     YDE         =    DISTANCE YE-YB
C     YE          =    A PRIORI Y COMP INFORMATION
C     YT          =    TARGET'S TRUE Y.COMP
C     Z           =    MEAS MODEL
C     ZDBRG       =    BEARING MEAS IN DEGS
C     ZFREQ       =    MATRIX OF ZZFREQ'S
C     ZGATE       =    SUBROUTINE TO CALC GATE3 AND TEST PREDICTED
C                      RESIDUAL
C     ZZBRG       =    NOISY BEARING MEAS IN RADS
C     ZZFREQ      =    NOISY FREQ MEAS IN HZ
C
C     *** VARIABLE DECLARATIONS ***
C
      DIMENSION TIME(200),XT(200),YT(200),VT(200),THDG(200)
      DIMENSION XB(10),YB(10),BTYPE(10),BFLAG(10)
      DIMENSION VS(5,200),VX(200),VY(200)
      DIMENSION H(5,5),HT(5,5),G(5,5),PHI(5,5),TEMP(5,5),TEMP1(5,5)
      DIMENSION Q(5,5), PKK(5,5),PKKM1(5,5),TEMP2(5,5),TEMP3(5,5)
      DIMENSION GAM(5,5),UNIT(5,5),SIGMA(3)
```

```
      DIMENSION Z(5),E(5),GE(5),W(5,5),EXT(200),EYT(200)
      DIMENSION GAMT(5,5),PHIT(5,5),XKK(5),XKKM1(5)
      DIMENSION ZDBRG(5,200),DBRG(5,200),FD(5,200),ZFREQ(5,200)
      REAL*8 DSEED
      CHARACTER*5 BTYPE
C
C     INITIALIZE TERMS
C     K=DISCRETE PT IN TIME, THE STAGE OF THE PROCESS
C     L=1 ROW OR COLUMN
C     M=2 ROW OR COLUMNS
C     N=5 ROWS OR COLUMNS
C     LD=MD=ND=MAX # OF ROWS OR COLUMNS
C     DT=DELAY TIME
C     MZ=NO. OF TRACK VALUES
C     MZZ=MZ-1 USE IN DO LOOP
      NZ=1
      MZ=60
      PFLAG=0.0
      NZZ=NZ-1
      MZZ=MZ-1
      NNZ=NZ+10
      L=1
      M=3
      N=5
      LD=5
      MD=5
      ND=5
      PI180=0.0174533
      PI=3.1415927
      TWOPI=2*PI
      RFREQ=300.
      VP=4860.
      FDOP=0.
      DSEED=50519
C     FLAGS
```

```
C     NOISE FLAG-0.0 NOISE OFF,1.0 NOISE ON
      NFLAG=1.0
C     MANV FLAG-0.0 MANV OFF,1.0 MANV ON
      MFLAG=1.0
C     ADAPT GATING FLAG-0.0 OFF,1.0 ON
      AFLAG=1.0
C     INITIALIZE FLAGS
      GFLAG=0.
      FLAG=0.
C     NUMBER OF INITIAL PKKM1 MATRIX TO USE
      INIT=4
C     INITIAL ESTIMATE OF TARGET POSITION, VELOCITY, AND COURSE IN
C     NM, KTS, AND DEGS.
      IF(PFLAG.EQ.0.0) THEN
      XE=10.
      YE=12.0
      VE=5.0
      HE=180.
C     STD. DEV. FOR PKKM(0/-1)
C     POSIT=0.5NM,VEL=3KTS,HEADING=10DEG,FREQ=1 HZ
      SIGPOS=0.5
      SIGVE=3.0
      SIGHE=10
      SIGFE=1.0
C     CONVERT XE AND YE TO YARDS
      XE=XE*2025.3667
      YE=YE*2025.3667
      SIGPOS=SIGPOS*2025.3667
C     KTS TO YARDS/MIN
      VE=VE*33.75633
      SIGVE=SIGVE*33.75633
C     CALC VXE AND VYE OF THE TARGET
         VXE=VE*SIN(HE*PI180)
         VYE=VE*COS(HE*PI180)
         SIGVXE=SIGVE*SIN((HE+SIGHE)*PI180)
```

```
         SIGVYE=SIGVE*COS((HE+SIGHE)*PI180)
      ENDIF
C     3 SIGMA GATE AND RESTART COUNTER FOR ADAPT. GATING
      GATE3=0.
      KOUNT=0
C
C
C     CONVERT VP (VEL OF SOUND IN WATER) FROM FT/SEC TO YARDS/MIN
      VP=VP*60.0/3.0
C
C
C     READ IN ACTUAL TARGET DATA
C
      READ(3,1)(TIME(I),XT(I),YT(I),VT(I),THDG(I),I=NZ,MZ)
1     FORMAT(F8.2,4X,F11.4,4X,F11.4,4X,F11.4,4X,F7.2)
C     WRITE ACTUAL TRACK VALUES
C     WRITE(8,1000)
1000  FORMAT(4X,'TIME',12X,'XT',12X,'YT',10X,'TGT VEL',6X,'TGT HDG')
C     DO LOOP CONVERTS VT(I) FROM KTS TO OTHER UNITS
      DO 800 I=NZ,MZ
C     CONVERT VT TO AGREE WITH VP
C     KTS TO YARDS/MIN
      VT(I)=VT(I)*33.75633
C     CONVERT XT AND YT TO YARDS
      XT(I)=XT(I)*2025.3667
      YT(I)=YT(I)*2025.3667
800   CONTINUE
C
      WRITE(8,1)(TIME(I),XT(I),YT(I),VT(I),THDG(I),I=NZ,MZ)
C1001  FORMAT(F8.2,3(4X,F11.4),4X,F7.2)
C
C     READ IN NUMBER OF BUOYS, THEN READ TYPE AND LOCATION
      READ(2,5) NBUOY
5     FORMAT(I3)
C     BFLAG=1.0 BUOY GIVES BEARING AND FREQ
```

```
C     BFLAG=2.0 BUOY GIVES FREQ ONLY
C     READ(4,6)(BTYPE(I),XB(I),YB(I),I=1,NBUOY)
      READ(2,6)(BTYPE(I),BFLAG(I),XB(I),YB(I),I=1,NBUOY)
6     FORMAT(2X,A5,2X,F2.0,2X,F8.4,4X,F8.4)
C     WRITE ACTUAL BUOYS -TYPES AND LOCATIONS
C     CONVERT XB AND YB TO YARDS
      DO 1004 I=1,NBUOY
      XB(I)=XB(I)*2025.3667
      YB(I)=YB(I)*2025.3667
1004  CONTINUE
      WRITE(8,1005)
1005  FORMAT(//,10X,'BUOYS')
      WRITE(8,1006)
1006  FORMAT(3X,'TYPE',3X,'FLAG',8X,'XB',13X,'YB')
      WRITE(8,7) (BTYPE(I),BFLAG(I),XB(I),YB(I),I=1,NBUOY)
7     FORMAT(3X,A5,3X,F2.0,3X,F11.4,4X,F11.4)
C
C     CALC. THE ACTUAL BEARINGS FROM BUOYS TO TARGET TRACKS AND THE
C     FREQS. RECEIVED
C
C     WRITE BEARINGS AND FREQS FOR EACH BUOY
      DO 102 I=NZ,MZ
C        WRITE(10,1008)
1008     FORMAT(//,' POSIT ',4X,'XT',11X,'YT')
C        WRITE(10,1009) I,XT(I),YT(I)
1009     FORMAT(I4,2X,2(F11.4,2X))
C        WRITE(10,1010)
1010     FORMAT(/,7X,'XB',11X,'YB',7X,'BRG',8X,'FREQ')
      DO 103 J=1,NBUOY
         XD=XT(I)-XB(J)
         YD=YT(I)-YB(J)
         CALL BEAR(XD,YD,BRG)
         DBRG(J,I)=BRG/PI180
C     CALC VX AND VY OF THE TARGET
         VX(I)=VT(I)*SIN(THDG(I)*PI180)
```

```
         VY(I)=VT(I)*COS(THDG(I)*PI180)
         CALL DOPP(VX,VY,XD,YD,VP,RFREQ,VMEAS,FDOP,I)
         FD(J,I)=FDOP
         VS(I,J)=VMEAS
C        WRITE(10,1010) XB(J),YB(J),XT(I),YT(I),DBRG(J,I),FD(J,I)
C        WRITE(10,1011) XB(J),YB(J),DBRG(J,I),FD(J,I)
1011     FORMAT(2(F11.4,2X),F7.2,2X,F10.4)
103   CONTINUE
102   CONTINUE
C     STARTS KALMAN FILTER PART OF PROGRAM
C
      K=NZ
      DO 899 I=1,M
      DO 899 J=1,M
899   W(I,J)=0.0
      DO 900 I=1,N
      DO 900 J=1,N
       PHI(I,J)=0.
       PHI(I,I)=1.0
       PKKM1(I,J)=0.0
900   CONTINUE
       DO 20 I=1,N
          DO 25 J=1,M
           GAM(I,J)=0.
25    CONTINUE
20    CONTINUE
      IF(PFLAG.EQ.0.0) THEN
      XKKM1(1)=XE
      XKKM1(2)=VXE
      XKKM1(3)=YE
      XKKM1(4)=VYE
      XDE=XE-XB(1)
      YDE=YE-YB(1)
      CALL DOPP(VXE,VYE,XDE,YDE,VP,RFREQ,VMEAS,FDOP,1)
      XKKM1(5)=FDOP
```

```
C
C     4 INITIAL PKKM1 MATRICES ARE SET UP USE 1
C     IF CONTINUING FROM A PREVIOUS RUN PFLAG =1.0
C     AND THE PKKM1 MATRIX WILL BE READ IN
      IF(INIT.EQ.1) THEN
C     INITIAL #1
      PKKM1(1,1)=1000.
      PKKM1(2,2)=500.
      PKKM1(3,3)=1000.
      PKKM1(4,4)=500.
C
      ELSEIF(INIT.EQ.2) THEN
C     INITIAL #2
C     POSITION (1NM)**2, VEL. (2KTS)**2
      PKKM1(1,1)=4.0E6
      PKKM1(2,2)=4.0E3
      PKKM1(1,2)=1.4E4
      PKKM1(2,1)=PKKM1(1,2)
      PKKM1(3,3)=4.0E6
      PKKM1(3,4)=1.4E4
      PKKM1(4,3)=PKKM1(3,4)
      PKKM1(4,4)=4.0E3
      ELSE
C     INITIAL #3
C     POSITION (.5NM)**2, VEL. (3KTS)**2
      PKKM1(1,1)=SIGPOS
      PKKM1(2,2)=SIGVXE
      PKKM1(3,3)=SIGPOS
      PKKM1(4,4)=SIGVYE
      ENDIF
C
      PKKM1(5,5)=SIGFE
      ELSE
      READ(2,91) TIME(NZZ)
      DO 7235 I=1,N
```

```
7235  READ(2,*) (PKK(I,J),J=1,N)
      WRITE(8,656)
      DO 7323 I=1,N
7323  WRITE(8,92) (PKK(I,J),J=1,N)
      READ(2,*) (XKK(I),I=1,N)
      WRITE(8,8011)
      WRITE(8,92)(XKK(J),J=1,N)
      GO TO 67
      ENDIF
C
C      WRITE(8,7177) K
7177  FORMAT(//,'############  K=',I4)
      WRITE(8,555)
555   FORMAT(/'    PKKM1 MATRIX  ')
      DO 3022 I=1,N
3022  WRITE(8,91) (PKKM1(I,J),J=1,N)
      WRITE(8,8812)
8812  FORMAT (/'    XKKM1  ')
      WRITE(8,91) (XKKM1(J),J=1,N)
C
C
91    FORMAT(8(1PE12.4))
92    FORMAT(2X,8(1PE12.5,2X))
67    CONTINUE
C     NB IS COUNTER INDICATING BUOY NUMBER
      NB=1
C
      IF(K.EQ.1) GO TO 3
      WRITE(8,7177) K
      DT=TIME(K)-TIME(K-1)
      WRITE(8,7905) DT
7905  FORMAT(5X,'    DT=  ',F10.2)
      CALL PHIGAM(DT,N,M,PHI,GAM,K)
C     WRITE PHI MATRIX
      WRITE(8,979)
```

```
979   FORMAT(/'    PHI MATRIX  ')
      DO 3580 I=1,N
3580  WRITE(8,92) (PHI(I,J),J=1,N)
C     WRITE GAMMA MATRIX
      WRITE(8,978)
978   FORMAT(/'   GAMMA MATRIX  ')
      DO 3581 I=1,N
3581  WRITE(8,92) (GAM(I,J),J=1,M)
      CALL PROD(PHI,XKK,N,N,L,XKKM1,N,M,L)
       WRITE(8,8812)
       WRITE(8,92) (XKKM1(J),J=1,N)
      CALL QFIND(DT,GAM,XKKM1,W,Q,N,M,ND,MD,SIGMA,K,MFLAG,GFLAG)
C     WRITE(8,544)
544   FORMAT(/'    W ')
C      DO 3021 I=1,3
C3021  WRITE(8,92) (W(I,J),J=1,3)
      WRITE(8,799)
799   FORMAT(/'    Q MATRIX  ')
      DO 3123 I=1,N
3123  WRITE(8,92) (Q(I,J),J=1,N)
C
3     IF((BFLAG(NB).EQ.1.0).OR.(BFLAG(NB).EQ.2.0)) THEN
      RFLAG=0.0
      TGATE=0.0
      WRITE(8,3583) NB
3583  FORMAT(//,5X,'FREQ MEAS FROM BUOY ',I2)
      CALL FMEAS(XB,YB,XKKM1,VP,H,R,N,NB,K,FDKKM1,D)
8     CALL GAIN(PKK,PKKM1,Q,R,PHI,H,N,L,G,ND,MD,LD,K,FLAG,GATE,GFLAG,NZ)
      WRITE(8,2400) GATE
2400  FORMAT(/'GATE =',1PE12.5)
C
      WRITE(8,656)
656   FORMAT(/'    PKK    ')
      DO 3023 I=1,N
3023  WRITE(8,92) (PKK(I,J),J=1,N)
```

```
C
C     SOLVE XKK=XKKM1+G(K)(Z(1)-FDKKM1)
C
      IF(GFLAG.EQ.1.) GO TO 5110
C     CALC FREQ. CONFIDENCE LEVELS (IE STD DEV)
      D=D/2025.3667
      IF(D.LE.2.)THEN
C     SIGF=0.02
      SIGF=0.04
      ELSEIF(D.LE.5.0) THEN
C     SIGF=0.04
      SIGF=0.06
      ELSEIF(D.LE.10.0) THEN
      SIGF=0.08
      ELSE
      SIGF=0.1
      ENDIF
      WRITE(8,5009) SIGF
5009  FORMAT(/,'FREQ. STD DEV =',F4.2)
C     ADD NOISE TO FREQ MEAS.
      CALL GAUSS(DSEED,SIGF,FD(NB,K),ZZFREQ,NFLAG)
      ZFREQ(NB,K)=ZZFREQ
C
5110  E(1)=ZFREQ(NB,K)-FDKKM1
        WRITE(8,8811)
C       WRITE(10,8811)
8811   FORMAT(/,8X,'Z',10X,'ZKKM1',8X,'ACTUAL')
      WRITE(8,92) ZFREQ(NB,K),FDKKM1,FD(NB,K)
C     WRITE(10,92) ZFREQ(NB,K),FDKKM1,FD(NB,K)
      WRITE(8,3029)
C     PRINT OUT ERROR
      WRITE(8,92) E(1)
      GFLAG=0.
C
C     AFLAG=1.0 USE ADAPTIVE FILTER
```

```
C     AFLAG=0.  DON'T USE ADAPTIVE FILTER
      IF(AFLAG.EQ.1.) THEN
C     DETERMINES IF RESIDUAL IS INSIDE GATE3
C     GATE1=(H*PKKM1*H'+R)**0.5
C     GATE3=3*GATE1
      IF(K.NE.1) THEN
      CALL ZGATE(E(1),GATE,R,W,GFLAG,KOUNT,GATE3)
      IF(GFLAG.EQ.1.0) TGATE=TGATE+1.0
      IF(KOUNT.EQ.3) THEN
      CALL RSTART(XB,XKKM1,PKKM1,RFLAG)
      GFLAG=2.
      RFLAG=RFLAG+1.0
      WRITE(8,987)
987   FORMAT(/,'********* RESTART THE PROBLEM **********')
      KOUNT=0
C     WRITE XKKM1
      WRITE(8,8812)
      WRITE(8,92) (XKKM1(J),J=1,N)
C     WRITE PKKM1
      WRITE(8,555)
      DO 3024 I=1,N
3024  WRITE(8,92) (PKKM1(I,J),J=1,N)
      ENDIF
      IF(GFLAG.NE.0.) THEN
      CALL QFIND(DT,GAM,XKKM1,W,Q,N,M,ND,MD,SIGMA,K,MFLAG,GFLAG)
C     WRITE(8,544)
C     544   FORMAT(/'    W ')
C     DO 3025 I=1,3
C3025  WRITE(8,92) (W(I,J),J=1,3)
      WRITE(8,799)
      DO 3124 I=1,N
3124  WRITE(8,92) (Q(I,J),J=1,N)
      GFLAG=1.
      GO TO 8
      ENDIF
```

```
      ENDIF
      IF(KOUNT.GT.0) KOUNT=KOUNT-1
      ENDIF
      CALL PROD(G,E,N,L,L,GE,ND,MD)
      CALL ADD(XKKM1,GE,N,L,XKK,ND,MD)
      WRITE(8,8011)
8011   FORMAT(/'   XKK          ')
      WRITE(8,92)(XKK(J),J=1,N)
C     WRITE(8,9897)  K,NB
9897  FORMAT(//' ***** SUMMARY FOR K= ',I4,'  FROM BUOY ',I2,'*****')
C     WRITE(8,9899)
      EXT(K)=XKK(1)
      EYT(K)=XKK(3)
      VARX=0.0
      VARY=0.0
      SUMX=0.0
      SUMY=0.0
      IF(K.GE.NNZ) THEN
      NM=K-10
      DO 3033 I=NM,K
         SUMX=SUMX+(XT(I)-EXT(I))**2
         SUMY=SUMY+(YT(I)-EYT(I))**2
3033  CONTINUE
      ENDIF
      VARX=SUMX/9.
      VARY=SUMY/9.
C
C     SET UP ARRAYS TO COLLECT GAINS, VARIANCES, AND ERROR ELLIPSOID
C     DATA FOR PLOTS.
C
      CALL PLOTF(TIME,EXT,EYT,G,PKK,K,NB,NBUOY,NZ,MZ,VARX,VARY)
C     STORE RESIDUAL AND GATE3 VS TIME FOR PLOTS
      CALL PLOTGF(TIME,K,NB,NBUOY,NZ,MZ,ABS(E(1)),GATE3,TGATE,RFLAG)
C
9899  FORMAT(5X,'TIME',9X,'XT',11X,'YT',12X,'EST XT',11X,'EST YT')
```

```
C     WRITE(8,393) (TIME(I),XT(I),YT(I),EXT(I),EYT(I),I=1,K)
      WRITE(8,393) TIME(K),XT(K),YT(K),EXT(K),EYT(K)
      WRITE(10,393) TIME(K),XT(K),YT(K),EXT(K),EYT(K)
393   FORMAT(/,F8.2,4X,F11.3,4X,F11.3,4X,F11.3,4X,F11.3)
      ENDIF
C
C     BEARING MEASUREMENT
C
      IF((BFLAG(NB).EQ.1.0).OR.(BFLAG(NB).EQ.3.0)) THEN
C     WRITE(8,3584) NB
C3584  FORMAT(//,5X,'BEARING MEAS. FROM BUOY ',I2)
      IF(BFLAG(NB).EQ.1.0) THEN
      FLAG=1.0
C     FLAG=1.0 MEANS REMAIN IN SAME TIME SLOT IE. K REMAINS THE SAME
C
      DO 4000 I=1,N
4000  XKKM1(I)=XKK(I)
      WRITE(8,8812)
      WRITE(8,92) (XKKM1(J),J=1,N)
      DO 4001 I=1,N
      DO 4001 J=1,N
4001  PKKM1(I,J)=PKK(I,J)
       WRITE(8,555)
       DO 4002 I=1,N
4002  WRITE(8,92) (PKKM1(I,J), J=1,N)
      ELSE
       FLAG=0.0
      ENDIF
      WRITE(8,3584) NB
3584  FORMAT(//,5X,'BEARING MEAS. FROM BUOY ',I2)
      CALL BMEAS(XB,YB,XKKM1,VP,H,R,N,NB,K,BRKKM1,D)
      RFLAG=0.0
      TGATE=0.0
4     CALL GAIN(PKK,PKKM1,Q,R,PHI,H,N,L,G,ND,MD,LD,K,FLAG,GATE,GFLAG,NZ)
      WRITE(8,2400) GATE
```

```
C
C
       WRITE(8,656)
       DO 4003 I=1,N
4003  WRITE(8,92) (PKK(I,J),J=1,N)
C
C     SOLVE XKK=XKKM1+G(K)(Z(1)-BRKKM1)
C
      IF(GFLAG.EQ.1.) GO TO 5111
C     CALC BEARING CONFIDENCE LEVELS
      D=D/2025.3667
      IF(D.LE.2.)THEN
C     SIGDB=2.0
      SIGDB=5.0
      ELSEIF(D.LE.5.0) THEN
C     SIGDB=5.0
      SIGDB=10.0
      ELSEIF(D.LE.10.0) THEN
      SIGDB=10.0
      ELSE
      SIGDB=15.0
      ENDIF
      SIGB=SIGDB*PI180
      WRITE(8,5010) SIGDB
5010  FORMAT(/,'BEARING STD DEV = ',F4.2)
C     ADD NOISE TO BRG MEAS.
      BRG=DBRG(NB,K)*PI180
      CALL GAUSS(DSEED,SIGB,BRG,ZZBRG,NFLAG)
      Z(1)=ZZBRG
      IF(BRKKM1.LT.0.) BRKKM1=BRKKM1+TWOPI
      E(1)=Z(1)-BRKKM1
      IF(E(1).GT.PI) E(1)=E(1)-TWOPI
      IF(E(1).LT.-PI) E(1)=E(1)+TWOPI
5111    WRITE(8,8811)
C       WRITE(10,8811)
```

```
        WRITE(8,92) Z(1),BRKKM1,BRG
      DBKKM1=BRKKM1/PI180
      ZDBRG(NB,K)=Z(1)/PI180
      WRITE(8,93) ZDBRG(NB,K),DBKKM1,DBRG(NB,K)
C     WRITE(10,94) ZDBRG(NB,K),DBKKM1,DBRG(NB,K)
93    FORMAT(/,5X,F7.2,5X,F7.2,5X,F7.2)
94    FORMAT(/,5X,F7.2,5X,F7.2,5X,F7.2)
        WRITE(8,3029)
3029   FORMAT(/ '   ERROR    ************************* ')
        WRITE(8,92) E(1)
      GFLAG=0.
C     AFLAG=1.0 USE ADAPTIVE FILTER
C     AFLAG=0.  DON'T USE ADAPTIVE FILTER
      IF(AFLAG.EQ.1.) THEN
C     DETERMINES IF RESIDUAL IS INSIDE GATE3
C     GATE1=(H*PKKM1*H'+R)**0.5
C     GATE3=3*GATE1
      IF(K.NE.1) THEN
      CALL ZGATE(E(1),GATE,R,W,GFLAG,KOUNT,GATE3)
      IF(GFLAG.EQ.1.0) TGATE=TGATE+1.0
      IF(KOUNT.EQ.3) THEN
      CALL RSTART(XB,XKKM1,PKKM1)
      GFLAG=2.0
      RFLAG=RFLAG+1.
      WRITE(8,987)
C     987   FORMAT(/,'********* RESTART THE PROBLEM **********')
      KOUNT=0
C     WRITE XKKM1
      WRITE(8,8812)
      WRITE(8,92) (XKKM1(J),J=1,N)
C     WRITE PKKM1
      WRITE(8,555)
      DO 3026 I=1,N
3026  WRITE(8,92) (PKKM1(I,J),J=1,N)
      ENDIF
```

```
      IF(GFLAG.NE.0.) THEN
      CALL QFIND(DT,GAM,XKKM1,W,Q,N,M,ND,MD,SIGMA,K,MFLAG,GFLAG)
C     WRITE(8,544)
C     544   FORMAT(/'    W ')
C     DO 3027 I=1,3
C3027  WRITE(8,92) (W(I,J),J=1,3)
      WRITE(8,799)
      DO 3125 I=1,N
3125  WRITE(8,92) (Q(I,J),J=1,N)
      GFLAG=1.
      GO TO 4
      ENDIF
      ENDIF
      IF(KOUNT.GT.0) KOUNT=KOUNT-1
      ENDIF
      CALL PROD(G,E,N,L,L,GE,ND,MD)
      CALL ADD(XKKM1,GE,N,L,XKK,ND,MD)
C     PRINT OUT XKK
        WRITE(8,8011)
        WRITE(8,92)(XKK(J),J=1,N)
C     WRITE(8,9897)  K,NB
C     WRITE(8,9899)
      EXT(K)=XKK(1)
      EYT(K)=XKK(3)
C
      VARX=0.0
      VARY=0.0
      SUMX=0.0
      SUMY=0.0
      IF(K.GE.NNZ) THEN
      NM=K-10
      DO 3034 I=NM,K
         SUMX=SUMX+(XT(I)-EXT(I))**2
         SUMY=SUMY+(YT(I)-EYT(I))**2
3034  CONTINUE
```

```
      ENDIF
      VARX=SUMX/9.
      VARY=SUMY/9.
C     SET UP ARRAYS TO COLLECT GAINS, VARIANCES, AND ERROR ELLIPSOID
C     DATA FOR PLOTS.
C     COMMENT PLOTB OUT IF WANT ONLY VEL. ERROR ELLIPSES ONLY FROM
C     PLOTF.
C
      CALL PLOTB(TIME,EXT,EYT,G,PKK,K,NB,NBUOY,NZ,MZ,BFLAG,VARX,VARY)
C
C     STORE RESIDUAL AND GATE3 VS TIME FOR PLOTS
      CALL PLOTGB(TIME,K,NB,NBUOY,NZ,MZ,ABS(E(1)),GATE3,TGATE,RFLAG)
C
C     WRITE(8,393) (TIME(I),XT(I),YT(I),EXT(I),EYT(I),I=NZ,K)
      WRITE(8,393) TIME(K),XT(K),YT(K),EXT(K),EYT(K)
      WRITE(10,393) TIME(K),XT(K),YT(K),EXT(K),EYT(K)
      ENDIF
      IF(NB.LT.NBUOY) THEN
      NB=NB+1
        FLAG=1.0
      DO 5000 I=1,N
5000  XKKM1(I)=XKK(I)
       WRITE(8,8812)
       WRITE(8,92) (XKKM1(J),J=1,N)
      DO 5001 I=1,N
      DO 5001 J=1,N
5001  PKKM1(I,J)=PKK(I,J)
       WRITE(8,555)
       DO 5002 I=1,N
5002  WRITE(8,92) (PKKM1(I,J), J=1,N)
C
C     LOOP BACK TO CALC FREQ AND BEARING FROM NEXT BUOY
        GO TO 3
      ENDIF
C     COMMENT OUT PLOTF AND PLOTB ABOVE. THE CALL BELOW WILL WRITE
```

```
C     THE LAST ELLIPSE CALC. BY PLOTF OR PLOTB.
      IF(BFLAG(NB).EQ.2.) THEN
      CALL PLOTF(TIME,EXT,EYT,G,PKK,K,NB,NBUOY,NZ,MZ,VARX,VARY)
      ELSE
      CALL PLOTB(TIME,EXT,EYT,G,PKK,K,NB,NBUOY,NZ,MZ,BFLAG,VARX,VARY)
      ENDIF
      K=K+1
      KK=K-1
      IF(K.GT.MZ) GO TO 888
      GO TO 67
C
C     SET UP FILES TO PLOT
888   WRITE(8,9897) KK,NB
      DO 104 I=NZ,MZ
         WRITE(10,1008)
         WRITE(10,1009) I,XT(I),YT(I)
         WRITE(10,1013)
1013  FORMAT(/,6X,'XB',9X,'YB',9X,'BRG',6X,'ZDBRG',7X,'FREQ',7X,
     *'ZFREQ')
      DO 105 J=1,NBUOY
      WRITE(10,1014) XB(J),YB(J),DBRG(J,I),ZDBRG(J,I),FD(J,I),ZFREQ(J,I)
 1014  FORMAT(2(F11.4,2X),F7.2,2X,F7.2,2X,F10.4,2X,F10.4)
 105   CONTINUE
 104   CONTINUE
      WRITE(8,9899)
      DO 6000 I=NZ,KK
      OFF=((XT(I)-EXT(I))**2+(YT(I)-EYT(I))**2)**0.5
      WRITE(8,393) TIME(I),XT(I),YT(I),EXT(I),EYT(I)
C     WRITE(10,92) XT(I),YT(I),EXT(I),EYT(I),OFF
      WRITE(4,92) XT(I),YT(I),EXT(I),EYT(I),OFF
6000  CONTINUE
C     PRINT OUT GAINS, VARIANCES  DATA FOR PLOTS
      CALL PLOTF(TIME,EXT,EYT,G,PKK,K,NB,NBUOY,NZ,MZ,VARX,VARY)
C     PLOT THE RESIDUAL AND GATE3 VS TIME
      CALL PLOTGF(TIME,K,NB,NBUOY,NZ,MZ,E(1),GATE3,TGATE,RFLAG)
```

```
      CALL PLOTB(TIME,EXT,EYT,G,PKK,K,NB,NBUOY,NZ,MZ,BFLAG,VARX,VARY)
C     PLOT THE RESIDUAL AND GATE3 VS TIME
      CALL PLOTGB(TIME,K,NB,NBUOY,NZ,MZ,E(1),GATE3,TGATE,RFLAG)
      WRITE(4,5) NBUOY
      WRITE(4,6001) (BFLAG(I),XB(I),YB(I),I=1,NBUOY)
6001  FORMAT(3X,F2.0,3X,F11.4,4X,F11.4)
      WRITE(4,5) INIT
C     WRITE(5,343)
      WRITE(2,91) TIME(KK)
343   FORMAT(/'    PKK    ')
      DO 3323 I=1,N
3323  WRITE(2,91) (PKK(I,J),J=1,N)
      WRITE(2,91)(XKK(J),J=1,N)
C
      STOP
      END
C
C
C
      SUBROUTINE BEAR (XD,YD,BRG)
C     *** PURPOSE ***
C     THIS GIVES ACTUAL BEARINGS FROM A BUOY TO THE TARGET IN RADIANS.
C     USING NORTH AS 360 DEGS., EAST AS 90 DEGS, SOUTH AS 180 DEGS.,
C     WEST AS 270 DEGS.
C
C     *** VARIABLE DEFINITIONS ***
C     SAME AS MAIN PROGRAM
C
      PI180=0.0174533
      PI=3.1415927
      TWOPI=6.283185
      IF(YD.EQ.0.0) THEN
         IF(XD.GT.0.0) THEN
           BRG=90.0*PI180
         ELSE
```

```
            BRG=270.0*PI180
          ENDIF
       GO TO 1
       ENDIF
       BRG=ATAN2(XD,YD)
       IF(BRG.LT.0.0) BRG=TWOPI+BRG
1      RETURN
       END
C
       SUBROUTINE DOPP (VX,VY,XD,YD,VP,RFREQ,VMEAS,FD,I)
C      *** PURPOSE ***
C      SUBROUTINE CALCS. THE DOPPLER FREQ.
C
C      *** VARIABLE DEFINITIONS ***
C      SAME AS MAIN PROGRAM, EXCEPT
C      R        =    RANGE(DISTANCE)
C
C      *** VARIABLE DECLARATIONS ***
       DIMENSION VX(200),VY(200)
       R=(XD*XD+YD*YD)**0.5
       VMEAS=(XD*VX(I)+YD*VY(I))/R
        FD=RFREQ/(1+(VMEAS/VP))
       RETURN
       END
C
C
C
C
C
       SUBROUTINE PHIGAM(T,N,M,PHI,GAM,K)
C      *** PURPOSE ***
C      CALCULATE THE PHI AND GAMMA MATRICES
C
C
C      *** VARIABLE DEFINITIONS ***
```

```
C     SAME AS MAIN PROGRAM
C
C     *** VARIABLE DECLARATIONS***
      DIMENSION PHI(5,5),GAM(5,5)
C
C     SET UP PHI MATRIX
92    FORMAT(2X,8(1PE12.5,2X))
      PHI(1,2)=T
      PHI(3,4)=T
      GAM(1,1)=(T*T)/2
      GAM(3,2)=GAM(1,1)
      GAM(2,1)=T
      GAM(4,2)=T
      GAM(5,3)=T
C     REMOVE C'S TO GET PRINTOUT IF DESIRED
C      WRITE(8,35)
C35    FORMAT(/,5X,'   PHI MATRIX  ')
C      DO 100 I=1,N
C100   WRITE(8,92) (PHI(I,J),J=1,N)
C     WRITE(8,40)
40    FORMAT(/,5X,' GAMMA MATRIX  ')
C     DO 101 I=1,N
C101   WRITE(8,92) (GAM(I,J),J=1,M)
      RETURN
      END
C
C
C
C
C
      SUBROUTINE GAIN(PKK,PKKM1,Q,R,PHI,H,N,L,G,ND,MD,LD,K,FLAG,GATE,
     *GFLAG,NZ)
      *** PURPOSE ***
C     THIS SUBROUTINE COMPUTES THE OPTIMUM GAIN MATRIX AND THE
C     COVARIANCE
```

```
C
C     *** VARIABLE DEFINITIONS ***
C     SAME AS MAIN PROGRAM
C
C     *** VARIABLE DECLARATIONS ****
      DIMENSION H(5,5),HT(5,5),G(5,5),PHI(5,5),TEMP(5,5),TEMP1(5,5)
      DIMENSION Q(5,5), PKK(5,5),PKKM1(5,5),TEMP2(5,5),TEMP3(5,5)
      DIMENSION GAM(5,5),UNIT(5,5)
      DIMENSION Z(5),E(5),GE(5)
      DIMENSION GAMT(5,5),PHIT(5,5),XKK(5),XKKM1(5)
C
C
      IF(K.EQ.NZ) THEN
      DO 900 I=1,N
         DO 900 J=1,N
         UNIT(I,J)=0.0
900      UNIT(I,I)=1.0
      ENDIF
C        REMAIN IN SAME TIME SO PHI AND GAM MATRICES ARE THE SAME
      IF((K.EQ.1).OR.(FLAG.EQ.1.0).OR.(GFLAG.EQ.1.0)) GO TO 8889
C       NOTE HERE PKKM1(I,J) = P(K/K-1)
C       WHERE P(K/K-1)=PHI*P(K-1/K-1)*PHIT + Q
C
C     CALC PKKM1
C
        CALL TRANS(PHI,N,N,PHIT,ND,MD)
        CALL PROD(PKK,PHIT,N,N,N,TEMP,ND,MD,LD)
        CALL PROD(PHI,TEMP,N,N,N,TEMP1,ND,MD,LD)
        CALL ADD(TEMP1,Q,N,N,PKKM1,ND,MD)
C
8889    CONTINUE
      IF(GFLAG.EQ.1.0) THEN
      CALL ADD(PKKM1,Q,N,N,PKKM1,ND,MD)
      ENDIF
      WRITE(8,8777) FLAG,GFLAG
```

```
      WRITE(8,555)
555   FORMAT(/'    MATRIX PKKM1 ')
      DO 3022 I=1,N
3022  WRITE(8,92)(PKKM1(I,J),J=1,N)
8777 FORMAT(/' FLAG= ',F4.2,2X,'GFLAG =',F4.2)
C
C     CALC GAIN G(K)
C
C     G(K) = P(K/K-1)*HT*(H*P(K/K-1)*HT + R)**-1
      CALL TRANS(H,L,N,HT,ND,MD)
      WRITE(8,39)
39    FORMAT('     H    ')
      DO 22 I=1,L
22    WRITE(8,92)(H(I,J),J=1,N)
92    FORMAT(2X,8(1PE12.5,2X))
C     WRITE(8,36)
36    FORMAT('     HT   ')
C     DO 23 I=1,N
C23    WRITE(8,92)(HT(I,J),J=1,L)
      CALL PROD(PKKM1,HT,N,N,1,TEMP,ND,MD,LD)
       WRITE(8,20)
20    FORMAT('   PKKM1*HT   ')
       DO 21 I=1,N
21    WRITE(8,92) (TEMP(I,J),J=1,L)
      CALL PROD(H,TEMP,L,N,L,TEMP1,ND,MD,LD)
      WRITE(8,38)
38    FORMAT('  H P HT ')
      DO 50 I=1,L
      DO 50 J=1,L
      GATE=TEMP1(I,J)
      TEMP2(I,J)=TEMP1(I,J)
50    TEMP3(I,J)=TEMP1(I,J)
      DO 24 I=1,L
24    WRITE(8,92)(TEMP3(I,J),J=1,L)
      TEMP3(1,1)=TEMP3(1,1)+R
```

```
      WRITE(8,338)
338   FORMAT('  H P HT + R')
      DO 224 I=1,L
224   WRITE(8,92)(TEMP3(I,J),J=1,L)
C      WRITE(8,31)
C31    FORMAT('    (HPHT + R)-1')
      DET=1/TEMP3(1,1)
C      DO 27 I=1,L
C27    WRITE(8,92) DET
      CALL CONST(DET,TEMP,N,L,G,ND,LD)
      WRITE(8,99)
99    FORMAT(/'   MATRIX G  ')
       DO 3024 I=1,N
3024    WRITE(8,92)(G(I,J),J=1,L)
C       NOTE HERE PKK(I,J) = P(K/K) WHERE P(K/K) = (I-G(K)*H)*P(K/K-1)
      CALL PROD(G,H,N,L,N,TEMP,ND,MD,LD)
C     WRITE(8,30)
C
30    FORMAT('    GH   ')
C     DO 25 I=1,N
C25    WRITE(8,92)(TEMP(I,J),J=1,N)
C     WRITE(8,37)
37    FORMAT('    IDENTITY MATRIX   ')
C     DO 45 I=1,N
C45    WRITE(8,92)(  UNIT(I,J),J=1,N)
C
C
      CALL SUB(UNIT,TEMP,N,N,TEMP1,ND,MD)
       WRITE(8,33)
33    FORMAT('  I -GH  ')
       DO 35 I=1,N
35    WRITE(8,92)(TEMP1(I,J),J=1,N)
      CALL PROD(TEMP1,PKKM1,N,N,N,PKK,ND,MD,LD)
      FLAG=0.0
      RETURN
```

```
      END
C
C
      SUBROUTINE QFIND(DT,GAM,XKKM1,W,Q,N,M,ND,MD,SIGMA,K,MFLAG,GFLAG)
C     *** PURPOSE ***
C     CALCULATES THE STATE EXCITATION COVARIANCE OF ERROR MATRIX
C
C
C     *** VARIABLE DEFINITIONS ***
C     SAME AS MAIN PROGRAM,EXCEPT
C     SIGVT2=(0.01 KTS/SEC)**2 = 410.8 YDS**2/MIN**4
C     SIGTH2=(0.1DEG/SEC)**2 = 0.01096 RADS**2/MIN**2
C     SIGF02=(0.001HZ/SEC)**2= 0.0036 HZ**2/MIN**2
C     CALC W MATRIX WHERE W= E(W(K)*W'(K))
C     W(1,1)=(SIGX)**2
C     W(2,2)=(SIGY)**2
C     W(1,2)=(SIGXY)
C     W(3,3)=(SIGFO)**2=SIGF02
C
C     *** VARIABLE DECLARATIONS ***
      DIMENSION GAM(5,5),GAMT(5,5),W(5,5),Q(5,5),TEMP(5,5)
      DIMENSION XKKM1(5),SIGMA(3)
C     SET UP W MATRIX
      IF(GFLAG.NE.1.) THEN
      IF(MFLAG.EQ.1) THEN
C     MFLAG = 0 NO MANUEVERING
C     MFLAG = 1 MANUEVERING
      SIGVT2 =410.8
      SIGTH2 = 0.01096
      SIGF02 = 0.0036
      SIGMA(1)=SIGVT2
      SIGMA(2)=SIGTH2
      SIGMA(3)=SIGF02
C
C
```

```
      EVT2 = XKKM1(2)**2+XKKM1(4)**2
      W(1,1)=((XKKM1(2)**2)/EVT2)*SIGVT2+(XKKM1(4)**2)*SIGTH2
      W(2,2)=((XKKM1(4)**2)/EVT2)*SIGVT2+(XKKM1(2)**2)*SIGTH2
      W(1,2)=((XKKM1(2)*XKKM1(4))/EVT2)*SIGVT2+(XKKM1(2)*XKKM1(4))*
     #SIGTH2
      W(2,1)=W(1,2)
      W(3,3)=SIGF02
      ELSE
      W(1,1)=10.
      W(2,2)=10.
      W(3,3)=0.01
      ENDIF
      ENDIF
543   WRITE(8,544)
544   FORMAT(/'    W ')
      DO 3021 I=1,3
3021  WRITE(8,92) (W(I,J),J=1,3)
C
      CALL TRANS(GAM,N,M,GAMT,ND,MD)
C     WRITE(8,1)
1     FORMAT(/,5X,'  GAMT MATRIX  ')
C     DO 100 I=1,M
C100   WRITE(8,92) (GAMT(I,J),J=1,N)
92    FORMAT(2X,8(1PE12.5,2X))
      CALL PROD(GAM,W,N,M,M,TEMP,ND,MD,MD)
C      WRITE(8,2)
2     FORMAT(/,5X,'  TEMP MATRIX  ')
C     DO 101 I=1,N
C101   WRITE(8,92) (TEMP(I,J),J=1,M)
      CALL PROD(TEMP,GAMT,N,M,N,Q,ND,MD,MD)
C     REMOVE C'S FOR PRINTOUT IF DESIRED
C     WRITE(8,3)
3     FORMAT(/,5X,'  Q MATRIX  ')
C     DO 102 I=1,N
C102   WRITE(8,92) (Q(I,J),J=1,N)
```

```
      RETURN
      END
C
C
C
C
      SUBROUTINE FMEAS(XB,YB,XKKM1,VP,H,R,N,NB,K,FDKKM1,D)
C     *** PURPOSE ***
C     SUBROUTINE CALCS THE H MATRIX FOR THE FREQ. MEASUREMENTS
C     AND SELECTS AND R.
C
C     *** VARIABLE DECLARATIONS ***
      DIMENSION XB(10),YB(10),XKKM1(5),H(5,5)
C     FREQ NOISE STD DEV SFREQ=0.04HZ.
      SFREQ= 0.04
      R=SFREQ**2
      WRITE(8,40)
40    FORMAT(/'  R FOR FREQ ')
      WRITE(8,92) R
C     WRITE(8,41)
41    FORMAT(/'   VP ')
C     WRITE(8,92) VP
92    FORMAT(2X,8(1PE12.5,2X))
      U=(((XKKM1(1)-XB(NB))*XKKM1(2))+((XKKM1(3)-YB(NB))*XKKM1(4)))
C     WRITE(8,32)
32    FORMAT(/'   U  ')
C     WRITE(8,92) U
      D=((XKKM1(1)-XB(NB))**2+(XKKM1(3)-YB(NB))**2)**0.5
C     WRITE(8,43)
43    FORMAT(/'   D  ')
C     WRITE(8,92) D
      H(1,5)=1/(1+(U/(VP*D)))
      AK=(XKKM1(5)*(H(1,5)**2))/(VP*D*D)
C     WRITE(8,34)
34    FORMAT(/'   AK ')
```

```
C     WRITE(8,92) AK
      H(1,1)=-AK*((XKKM1(2)*D)-(U*(XKKM1(1)-XB(NB))/D))
      H(1,2)=-AK*D*(XKKM1(1)-XB(NB))
      H(1,3)=-AK*((XKKM1(4)*D)-(U*(XKKM1(3)-YB(NB)))/D)
      H(1,4)=-AK*D*(XKKM1(3)-YB(NB))
      FDKKM1=XKKM1(5)*H(1,5)
C     WRITE(8,44)
44    FORMAT(/'    FDKKM1  ')
C     WRITE(8,92) FDKKM1
C
      RETURN
      END
C
C
C
C
      SUBROUTINE BMEAS(XB,YB,XKKM1,VP,H,R,N,NB,K,BRKKM1)
C     SUBROUTINE CALCS THE H MATRIX FOR THE BEARING MEASUREMENTS
C     AND SELECTS AND R.
      DIMENSION XB(10),YB(10), XKKM1(5),H(5,5)
      PI180=0.0174533
C     BEARING NOISE STD DEV SDBRG=5 DEGS.
      SDBRG=5.0
      R=(SDBRG*PI180)**2
      WRITE(8,40)
40    FORMAT(/'  R FOR BEAR.  ')
      WRITE(8,92) R
      A=XKKM1(1)-XB(NB)
      B=XKKM1(3)-YB(NB)
      D=SQRT(A**2+B**2)
92    FORMAT(2X,8(1PE12.5,2X))
C     WRITE(8,32)
32    FORMAT(/,5X,'A',12X,'B')
C     WRITE(8,92) A,B
C     WRITE(8,43)
```

```
43    FORMAT(/'   D  ')
C     WRITE(8,92) D
      H(1,1)=B/(D*D)
      H(1,2)=0.0
      H(1,3)=-A/(D*D)
      H(1,4)=0.0
      H(1,5)=0.0
      CALL BEAR(A,B,BRKKM1)
C     WRITE(8,44)
44    FORMAT(/'   BRKKM1  ')
C     WRITE(8,92) BRKKM1
      RETURN
      END
C
C
C
C
      SUBROUTINE PLOTF(TIME,XT,YT,G,PKK,K,NB,NBUOY,NZ,MZZ,VARX,VARY)
C     *** PURPOSE ***
C     CONTAINS GRAPHIC DATA FOR KALMAN GAINS, COVARIANCE MATRIX,AND
C     ERROR ELLIPSOIDS FOR FREQ MEAS
C
C     *** VARIABLE DEFINITIONS ***
C     TERMS SAME AS MAIN PROGRAM, EXCEPT FOR
C     ELLIP          =    SUBROUTINE  TO CALC ERROR ELLIPSOIDS
C     EVARX          =    MATRIX OF VARX
C     EVARY          =    MATRIX OF VARY
C     GF             =    MATRIX OF FREQ GAINS,STORE FOR PLOTTING
C     GX             =    MATRIX OF X COMP GAINS, STORE FOR PLOTTING
C     GXD            =    MATRICX OF VX COMP GAINS
C     GY             =    MATRIX OF Y COMP GAINS, STORE FOR PLOTTING
C     GYD            =    MATRIX OF VY COMP GAINS
C     PF             =    MATRIX OF FREQ COMP COVARIANCE OF ERROR
C     PVV            =    USED FOR PLOTTING ERROR ELLIPSOIDS
C     PX             =    X COMP VARIANCE, USED WITH ERROR ELLIPSOIDS
```

```
C     PXD           =    VX COMP VARIANCE,
C     PXY           =    COVARIANCE OF X AND Y COMPS
C     PY            =    Y COMP VARIANCE, USED WITH ERROR ELLIPSOIDS
C     PYD           =    VY COMP VARIANCE
C
C     *** VARIABLE DECLARATIONS ***
      DIMENSION TIME(200),XT(200),YT(200),G(5,5),PKK(5,5)
      DIMENSION YH(120),XH(120),XP(200),YP(200)
      DIMENSION GX(6,200),GXD(6,200),GY(6,200),GYD(6,200),GF(6,200)
      DIMENSION PX(6,200),PXD(6,200),PY(6,200),PYD(6,200),PF(6,200)
      DIMENSION PXY(6,200),PVV(6,200),EVARX(6,200),EVARY(6,200)
C     WRITE(7,1) NB
C     WRITE(9,1) NB
C1    FORMAT(/,5X,' FREQ.MEAS. FROM BUOY ',I2)
      KK=K-1
      IF(K.GT.MZZ) GO TO 888
      GX(NB,K)=G(1,1)
      GXD(NB,K)=G(2,1)
      GY(NB,K)=G(3,1)
      GYD(NB,K)=G(4,1)
      GF(NB,K)=G(5,1)
C     SET UP PLOTS OF PKKM1 COMPONENTS
      PX(NB,K)=PKK(1,1)
      PXD(NB,K)=PKK(2,2)
      PY(NB,K)=PKK(3,3)
      PYD(NB,K)=PKK(4,4)
      PF(NB,K)=PKK(5,5)
      PXY(NB,K)=PKK(1,3)
      PVV(NB,K)=PKK(2,4)
      EVARX(NB,K)=VARX
      EVARY(NB,K)=VARY
      PFLAG=0.0
C     COMMENT OUT ONE OF CALLS IN EACH IF-THEN, THE
C     FIRST CALL ELLIP COMPUTES POSITION ERROR ELLIPSES, SECOND ONE
C     THE VELOCITY ERROR ELLIPSES
```

```
C
      IF(K.EQ.1) THEN
      CALL ELLIP(XT,YT,PX,PY,PXY,K,NB,PFLAG)
C     CALL ELLIP(XT,YT,PXD,PYD,PVV,K,NB,PFLAG)
      ELSEIF(K.EQ.11) THEN
      CALL ELLIP(XT,YT,PX,PY,PXY,K,NB,PFLAG)
C     CALL ELLIP(XT,YT,PXD,PYD,PVV,K,NB,PFLAG)
      ELSEIF(K.EQ.31) THEN
      CALL ELLIP(XT,YT,PX,PY,PXY,K,NB,PFLAG)
C     CALL ELLIP(XT,YT,PXD,PYD,PVV,K,NB,PFLAG)
      ELSEIF(K.EQ.61) THEN
      CALL ELLIP(XT,YT,PX,PY,PXY,K,NB,PFLAG)
C     CALL ELLIP(XT,YT,PXD,PYD,PVV,K,NB,PFLAG)
      ENDIF
      GO TO 900
C
888   DO 6001 I=1,NBUOY
      WRITE(7,80) I
      WRITE(9,81) I
      WRITE(19,81) I
80    FORMAT(/,5X,' FREQ MEAS. GAINS FROM BUOY ',I2)
81    FORMAT(/,5X,' FREQ MEAS. VAR. FROM BUOY ',I2)
      DO 6002 J=NZ,KK
      WRITE(7,95) TIME(J),GX(I,J),GXD(I,J),GY(I,J),GYD(I,J),GF(I,J)
      WRITE(9,95) TIME(J),PX(I,J),PXD(I,J),PY(I,J),PYD(I,J),PF(I,J)
      WRITE(19,95) TIME(J),PX(I,J),PY(I,J),PXY(I,J)
      IF(J.GE.11) THEN
      WRITE(12,95) TIME(J),EVARX(I,J),EVARY(I,J),PX(I,J),PY(I,J)
      ENDIF
6002  CONTINUE
6001  CONTINUE
95    FORMAT(/F7.2,2X,5(1PE12.5,2X))
C95    FORMAT(/,F8.2,4X,F11.3,4X,F11.3,4X,F11.3,4X,F11.3)
900   RETURN
      END
```

```
C
C
C
      SUBROUTINE PLOTB(TIME,XT,YT,G,PKK,K,NB,NBUOY,NZ,MZZ,BFLAG,VARX,
     *VARY)
C     *** PURPOSE ***
C     CONTAINS GRAPHIC DATA FOR KALMAN GAINS, COVARIANCE MATRIX,AND
C     ERROR ELLIPSOIDS FOR BEARING MEAS
C
C     *** VARIABLE DEFINITIONS ***
C     SAME AS MAIN PROGRAM AND/OR  PLOTF
C
C     *** VARIABLE DECLARATIONS ***
      DIMENSION TIME(200),XT(200),YT(200),G(5,5),PKK(5,5)
      DIMENSION BFLAG(10),GF(6,200),PF(6,200)
      DIMENSION GX(6,200),GXD(6,200),GY(6,200),GYD(6,200)
      DIMENSION PX(6,200),PXD(6,200),PY(6,200),PYD(6,200)
      DIMENSION PXY(6,200),EVARX(6,200),EVARY(6,200)
C     WRITE(7,1) NB
C     WRITE(9,1) NB
C1    FORMAT(/,5X,' BEARING MEAS. FROM BUOY ',I2)
      KK=K-1
      IF(K.GT.MZZ) GO TO 888
      GX(NB,K)=G(1,1)
      GXD(NB,K)=G(2,1)
      GY(NB,K)=G(3,1)
      GYD(NB,K)=G(4,1)
      GF(NB,K)=G(5,1)
C     SET UP PLOTS OF PKKM1 COMPONENTS
      PX(NB,K)=PKK(1,1)
      PXD(NB,K)=PKK(2,2)
      PY(NB,K)=PKK(3,3)
      PYD(NB,K)=PKK(4,4)
      PF(NB,K)=PKK(5,5)
      PXY(NB,K)=PKK(1,3)
```

```
      EVARX(NB,K)=VARX
      EVARY(NB,K)=VARY
      PFLAG=1.0
      IF(K.EQ.1) THEN
      CALL ELLIP(XT,YT,PX,PY,PXY,K,NB,PFLAG)
      ELSEIF(K.EQ.11) THEN
      CALL ELLIP(XT,YT,PX,PY,PXY,K,NB,PFLAG)
      ELSEIF(K.EQ.31) THEN
      CALL ELLIP(XT,YT,PX,PY,PXY,K,NB,PFLAG)
      ELSEIF(K.EQ.61) THEN
      CALL ELLIP(XT,YT,PX,PY,PXY,K,NB,PFLAG)
      ENDIF
      GO TO 900
C
888   DO 6001 I=1,NBUOY
C     DON,T PRINT GAINS AND VAR IF ITS A LOFAR BUOY (THERE 0 ANYWAY)
      IF(BFLAG(I).EQ.2.) GO TO 6001
      WRITE(7,80) I
      WRITE(9,81) I
      WRITE(19,81) I
80    FORMAT(/,5X,' BEARING MEAS. GAINS FROM BUOY ',I2)
81    FORMAT(/,5X,' BEARING MEAS. VAR. FROM BUOY ',I2)
      DO 6002 J=NZ,KK
      WRITE(7,95) TIME(J),GX(I,J),GXD(I,J),GY(I,J),GYD(I,J),GF(I,J)
      WRITE(9,95) TIME(J),PX(I,J),PXD(I,J),PY(I,J),PYD(I,J),PF(I,J)
      WRITE(19,95) TIME(J),PX(I,J),PY(I,J),PXY(I,J)
      IF(J.GE.11) THEN
      WRITE(12,95) TIME(J),EVARX(I,J),EVARY(I,J),PX(I,J),PY(I,J)
      ENDIF
6002  CONTINUE
6001  CONTINUE
95    FORMAT(/F7.2,2X,5(1PE12.5,2X))
900   RETURN
      END
C
```

```
C
C
C
      SUBROUTINE ELLIP(XT,YT,P1,P3,P13,K,NB,PFLAG)
C     *** PURPOSE ***
C     *** ROUTINE TO PLACE ELLIPSE DATA IN FILE
C
C     *** VARIABLE DECLARATIONS ***
      DIMENSION XT(200),YT(200),XP(200),YP(200)
      DIMENSION P1(6,200),P3(6,200),P13(6,200)
      A=2*P13(NB,K)
      B=P1(NB,K)-P3(NB,K)
      IF((A.EQ.0.0).AND.(B.EQ.0.0)) B=0.0001
      THE1=.50*ATAN2(A,B)
      A=(P1(NB,K)+P3(NB,K))/2.
      B=0.0
      IF(P13(NB,K).EQ.0.0) GO  TO 10
      B=P13(NB,K)/SIN(2.*THE1)
10    SIG2X=(A+B)
      SIG2Y=(A-B)
      SX=((SIG2X)**.5)
      SY=((SIG2Y)**.5)
      PT=3.14159265/12
      CT=COS(THE1)
      ST=SIN(THE1)
C      WRITE(4,9897)  K,NB
9897  FORMAT(//' ***** SUMMARY FOR K= ',I4,'  FROM BUOY ',I2,'*****')
      IF(PFLAG.EQ.1.) THEN
C     WRITE(4,9898)
9898  FORMAT(/' BEARING MEAS. ERROR ELLIPSE ')
      ELSE
C     WRITE(4,9999)
9999  FORMAT(/' FREQ. MEAS. ERROR ELLIPSE ')
      ENDIF
      DO 1981 IELLIP=1,25
```

```
      XI=IELLIP
      XP(IELLIP)=SX*COS(PT*XI)*CT-SY*SIN(PT*XI)*ST+XT(K)
      YP(IELLIP)=SX*COS(PT*XI)*ST+SY*SIN(PT*XI)*CT+YT(K)
 1981 WRITE(4,1982)XP(IELLIP),YP(IELLIP)
 1982 FORMAT(2F14.4)
      RETURN
      END
C     *** END OF ELLIPSE CALCULATION
C
C
C
      SUBROUTINE PLOTGF(TIME,K,NB,NBUOY,NZ,MZ,E,GATE3,TGATE,RFLAG)
C     *** PURPOSE ***
C     GRAHICS INPUT FILE ADAPT GATE AND PREDICTED
C     RESIDUAL FROM FREQ MEAS
C
C     *** VARIABLE DEFINITIONS ***
C     SAME AS MAIN EXCEPT FOR
C     ERR     =    MATRIX TO STORE PREDICTED RESIDUALS
C     TRIP    =    MATRIX OF NUMBER OF TIMES GATE3 IS
C                  EXCCEDED
C
C     *** VARIABLE DECLARATIONS ***
      DIMENSION ERR(6,200),GATE(6,200),TIME(200)
      DIMENSION TRIP(5,200),RESTR(5,200)
C
C
      IF(K.GT.MZ) GO TO 888
      ERR(NB,K)=E
      GATE(NB,K)=GATE3
      TRIP(NB,K)=TGATE
      RESTR(NB,K)=RFLAG
      GO TO 900
888   DO 6000 I=1,NBUOY
      WRITE(18,80) I
```

```
80    FORMAT(/,5X,'FREQ. RESIDUAL AND GATE FROM BUOY ',I2)
      DO 6001 J=NZ,MZ
      WRITE(18,95) TIME(J),TRIP(I,J),RESTR(I,J),ERR(I,J),GATE(I,J)
6001  CONTINUE
6000  CONTINUE
95    FORMAT(F7.2,2X,F5.2,2X,F5.2,2X,2(1PE12.5,2X))
900   RETURN
      END
C
C
C
      SUBROUTINE PLOTGB(TIME,K,NB,NBUOY,NZ,MZ,E,GATE3,TGATE,RFLAG)
C     PLOTGB        =    SUBROUTINE TO PLOT ADAPT GATE AND PREDICTED
C                        RESIDUAL FROM BEARING MEAS
C     GRAHICS INPUT FILE ADAPT GATE AND PREDICTED
C     RESIDUAL FROM BEARING MEAS
C
C     *** VARIABLE DEFINITIONS ***
C     SAME AS MAIN EXCEPT FOR
C     ERR      =    MATRIX TO STORE PREDICTED RESIDUALS
C     TRIP     =    MATRIX OF NUMBER OF TIMES GATE3 IS
C                   EXCCEDED
C     RESTR    =    MATRIX OF THE NUMBER OF RESTARTS FOR EACH
C                   MEAS

C     *** VARIABLE DECLARATIONS ***
      DIMENSION ERR(6,200),GATE(6,200),TIME(200)
      DIMENSION TRIP(5,200),RESTR(5,200)
C
C
      IF(K.GT.MZ) GO TO 888
      ERR(NB,K)=E
      GATE(NB,K)=GATE3
      TRIP(NB,K)=TGATE
      RESTR(NB,K)=RFLAG
```

```
      GO TO 900
888   DO 6000 I=1,NBUOY
      WRITE(18,80) I
80    FORMAT(/,5X,'BRG. RESIDUAL AND GATE FROM BUOY ',I2)
      DO 6001 J=NZ,MZ
      WRITE(18,95) TIME(J),TRIP(I,J),RESTR(I,J),ERR(I,J),GATE(I,J)
6001  CONTINUE
6000  CONTINUE
95    FORMAT(F7.2,2X,F5.2,2X,F5.2,2X,2(1PE12.5,2X))
900   RETURN
      END
C
C
C
      SUBROUTINE ZGATE(E,GATE,R,W,GFLAG,KOUNT,GATE3,TGATE)
C     *** PURPOSE ***
C     CALCS THE GATE3, AND RANDOM FORCING FUNC COV MATRIX VALUES
C
C     *** VARIABLE DEFINITIONS ***
C     SAME AS MAIN PROGRAM,EXCEPT FOR
C     GATE1   =    ONE SIGMA- (GATE + R) **0.5
C
C     *** VARIABLE DECLARATIONS **
      DIMENSION W(5,5)
      GATE1=(GATE+R)**0.5
      GATE3=3*GATE1
      WRITE(8,96) GATE3
96    FORMAT(/,'GATE3 = ',1PE12.5)
      IF(ABS(E)-GATE3.GT.0.) THEN
      GFLAG=1.
      TGATE =TGATE+1
      KOUNT=KOUNT+1
      W(1,1)=10.*W(1,1)
      W(2,2)=10.*W(2,2)
      W(3,3)=2*W(3,3)
```

```
      WRITE(8,97)
97    FORMAT(/,' GATE3 HAS BEEN EXCEEDED ')
      ENDIF
      RETURN
      END
C
C
C
C
      SUBROUTINE RSTART(XB,XKKM1,PKKM1)
C     *** PURPOSE ***
C     REINITIALIZES THE PROGRAM
C
C     *** VARIABLE DEFINITONS ***
C     SAME AS MAIN PROGRAM
C
C     *** VARIABLE DECLARATIONS ***
      DIMENSION XKKM1(5),PKKM1(5,5),XB(10)
      DO 118 I=1,5
         DO 118 J=1,5
118      PKKM1(I,J)=0.0
      PKKM1(1,1)=1.025E6
      PKKM1(3,3)=PKKM1(1,1)
      PKKM1(2,2)=1.025E4
      PKKM1(4,4)=PKKM1(2,2)
      PKKM1(5,5)=1.0
      RETURN
      END
C
C
C
C
      SUBROUTINE GAUSS(DSEED,SIG,MEAN,Z,NFLAG)
C     *** PURPOSE ***
C     GAUSSIAN PSEUDO- RANDOM NUMBER GENERATOR
```

```
C
C
C     *** VARIABLE DECLARATIONS ***
      DOUBLE PRECISION DSEED
      REAL MEAN
      IF(NFLAG.EQ.1) THEN
      NR=1
      TEMP=0.0
      DO 10 I=1,12
      CALL GGUBS(DSEED,NR,R)
      TEMP=TEMP+R
10    CONTINUE
      Z=(TEMP-6.0)*SIG+MEAN
      ELSE
      Z=MEAN
      ENDIF
C     WRITE(6,92) (MEAN,Z(I),I=1,NR)
C92    FORMAT(/,2X,'MEAN= ',F8.3,' Z= ',F8.3)
      RETURN
      END
C
C
C
      SUBROUTINE GGUBS (DSEED,NR,R)
C     *** PURPOSE ***
C     BASIC UNIFORM (0,1) PSEUDO-RANDOM NUMBER GENERATOR
C
C
C     *** VARIABLE DECLARATIONS ***
      INTEGER            NR
      REAL               R(NR)
      DOUBLE PRECISION   DSEED
C     SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            I
      DOUBLE PRECISION   D2P31M,D2P31
```

```
C     D2P31M=(2**31) - 1
C     D2P31 =(2**31)(OR AN ADJUSTED VALUE)
      DATA                  D2P31M/2147483647.D0/
      DATA                  D2P31/2147483648.D0/
C     FIRST EXECUTABLE STATEMENT
      DO 5 I=1,NR
         DSEED = DMOD(16807.D0*DSEED,D2P31M)
    5 R(I) = DSEED / D2P31
      RETURN
      END
C
C
C
C
        SUBROUTINE ADD(A,B,N,M,C,ND,MD)
C     *** PURPOSE ***
C     SUBROUTINE ADDS TWO MATRICES
C     *** VARIABLE DECLARATIONS ***
      DIMENSION A(ND,MD),B(ND,MD),C(ND,MD)
C
        DO 152 I = 1,N
        DO 152 J = 1,M
  152   C(I,J) = A(I,J) + B(I,J)
      RETURN
        END
C
C
C
C
      SUBROUTINE SUB(A,B,N,M,C,ND,MD)
C     *** PURPOSE ***
C     SUBROUTINE SUBTRACTS TWO MATRICES
C     *** VARIABLE DECLARATIONS ***
        DIMENSION A(ND,MD),B(ND,MD),C(ND,MD)
        DO 152 I = 1,N
```

```
      DO 152 J = 1,M
  152 C(I,J) = A(I,J) - B(I,J)
      RETURN
      END
C
C
C
C
     SUBROUTINE PROD(A,B,N,M,L,C,ND,MD,LD)
C    *** PURPOSE ***
C    SUBROUTINE MULTIPLES TWO MATRICES
C    *** VARIABLE DECLARATIONS ***
      DIMENSION A(ND,MD),B(MD,LD),C(ND,LD)
      DO 1 I = 1,N
      DO 1 J = 1,L
    1 C(I,J) = 0.
      DO 151 I = 1,N
      DO 151 J = 1,L
      DO 151 K = 1,M
  151 C(I,J) =C(I,J) + A(I,K)*B(K,J)
      RETURN
      END
C
C
C
C
      SUBROUTINE TRANS(A,N,M,C,ND,MD)
C    *** PURPOSE ***
C    SUBROUTINE TRANSPOSES A MATRIX
C    *** VARIABLE DECLARATIONS ***
      DIMENSION  A(ND,MD),C(MD,ND)
      DO 153 I = 1,N
      DO 153 J = 1,M
  153 C(J,I) = A(I,J)
      RETURN
```

```
      END
C
C
C
C
      SUBROUTINE CONST(Q,A,N,M,C,ND,MD)
C    *** PURPOSE ***
C    SUBROUTINE MULTIPLES A MATRIX BY A CONSTANT
C    *** VARIABLE DECLARATIONS ***
      DIMENSION A(ND,MD),C(ND,MD)
      IF(Q) 11,10,11

 10   DO 100 I = 1,N
      DO 100 J = 1,M
 100  C(I,J) = 0.0
      RETURN
 11   IF (Q-1.0) 13,12,13
 12   DO 120 I = 1,N
      DO 120 J = 1,M
 120  C(I,J) = A(I,J)
      RETURN
 13   IF (Q+1.0) 15,14,15
 14   DO 140 I = 1,N
      DO 140 J = 1,M
 140  C(I,J) = -A(I,J)
      RETURN
 15   DO 150 I = 1,N
      DO 150 J = 1,M
 150  C(I,J) = Q*A(I,J)
      RETURN
      END
C
C
```

APPENDIX C. FORTRAN PROGRAMS FOR THE GRAPHICS

The two FORTRAN programs in this appendix are samples of the plotting routines. These routines were used to generate some of the graphics presented in this research. The programs are written in Display Integrated Software System and Plotting Language (DISSPLA) and executed on the IBM 3033 located at the Naval Postgraduate, Monterey, California.

```
C     SAMPLE OF GRAPHICS PROGRAM.
C     *** PURPOSE ***
C     THIS PROGRAM PLOTS THE TARGET'S ACTUAL TRACK, THE FILTER
C     ESTIMATED TRACK, THE BUOY PATTERN AND THE ERROR ELLIPSOIDS
C
C     *** VARIABLE DEFINITIONS ***
C     MAJORITY OF THE VARIABLES ARE SAME AS IN THE MAIN PROGRAM
C     XP      =    ERROR ELLIPSOIDS X COMP
C     YP      =    ERROR ELLIPSOIDS Y COMP
C     MZZ     =    NUMBER OF TRACK POINTS
C     NN      =    NUMBER OF ERROR ELLIPSOID POINTS(25 POINTS/ELLIPSE)
C     DISSPLA TERMS ARE DEFINE IN THE DISSPLA BOOK
C
C
C     *** VARIABLE DECLARATIONS ***
      REAL T(200),XT(200),YT(200),EXT(200),EYT(200),OFF(200)
      REAL XP(500),YP(500)
      REAL BFLAG(6),XB(6),YB(6)
C
      MZZ=60
      NN=525
      NE=NN/25
      XORIG=10.0
      XMAX=25.0
      YORIG=15.0
      YMAX=30.0
C      CALL TEK618
      CALL COMPRS
      CALL NOBRDR
      CALL PAGE(8.5,11.)
```

```
      CALL HEIGHT(0.15)
      CALL MX1ALF('STANDART','&')
      CALL MX2ALF('L/CSTD','#')
C     CALL PHYSOR(2.0,2.0)
      CALL AREA2D(6.0,6.0)
C     GRAPHICS ROUTINE TO TRACK,BUOYS AND ERROR ELLIPSES
C
      CALL INTAXS
       CALL XNAME('KYARDS$',100)
      CALL YNAME('KYARDS$',100)
      CALL GRAF(XORIG,'SCALE',XMAX,YORIG,'SCALE',YMAX)
      DO 10 I=1,NE
      DO 11 J=1,25
      READ(4,*) XP(J),YP(J)
      XP(J)=XP(J)/1000.0
      YP(J)=YP(J)/1000.0
11    CONTINUE
        CALL DASH
        CALL CURVE(XP,YP,25,0)
10    CONTINUE
      READ(4,*) (XT(I),YT(I),EXT(I),EYT(I),OFF(I),I=1,MZZ)
      READ(4,*) NBUOY
      READ(4,*) (BFLAG(I),XB(I),YB(I),I=1,NBUOY)
      DO 1 I=1,MZZ
      XT(I)=XT(I)/1000.0
      YT(I)=YT(I)/1000.0
      EXT(I)=EXT(I)/1000.0
      EYT(I)=EYT(I)/1000.0
1     CONTINUE
      CALL HEIGHT(0.15)
      CALL RESET('DASH')
      CALL MARKER(16)
      CALL CURVE(XT,YT,MZZ,5)
      CALL DASH
      CALL MARKER(4)
```

```
      CALL CURVE(EXT,EYT,MZZ,5)
      CALL HEIGHT(0.15)
      DO 1000 I=1,NBUOY
      XB(I)=XB(I)/1000.
      YB(I)=YB(I)/1000.
      TITX=XB(I)-1
      TITY=YB(I)+1
      IF(BFLAG(I).EQ.1.) THEN
      CALL RLMESS('DIFAR$',100,TITX,TITY)
      CALL MARKER(16)
      CALL SCLPIC(2)
      CALL CURVE(XB(I),YB(I),1,5)
      ENDIF
      IF(BFLAG(I).EQ.2.) THEN
       CALL RLMESS('LOFAR$',100,TITX,TITY)
       CALL MARKER(16)
      CALL SCLPIC(2)
      CALL CURVE(XB(I),YB(I),1,5)
      ENDIF
1000  CONTINUE
      CALL ENDPL(0)
      CALL DONEPL
      STOP
      END
```

```
C     *** PURPOSE ***
C     SAMPLE GRAPHICS PROGRAM PLOTS THE PREDICTED RESIDUAL,
C     ADAPTIVE GATE, AND THE NUMBER OF TIMES THE GATE IS EXCEEDED
C     FIRST THE PROGRAM PLOTS THE GRAPHS FOR THE FREQ MEAS, THEN ON
C     ANOTHER PAGE PLOTS THE GRAPHS FOR THE BEARING MEAS.
C     HENCE AS SHOWN IN FIG. 6-10
C
C     *** VARIABLE DEFINITIONS ***
C     SAME AS MAIN PROGRAM, AND STANDARD DISSPLA TERMS
C
C     DRAW RESIDUAL AND GATE FOR FREQ MEASUREMENTS
      REAL T(500),ERR(500),GATE(500),TRIP(500),FLAG(500)
      XORIG=0.0
      XMAX=60.0
C     XORIG=60.0
C     XMAX=140.0
      YORIG=0.0
      YMAX=3.0
C      CALL TEK618
       CALL COMPRS
      CALL NOBRDR
      CALL PAGE(11.,8.5)
      CALL HEIGHT(0.15)
      CALL MX1ALF('STANDART','&')
      CALL MX2ALF('L/CSTD','#')
      READ(4,*) (T(I),TRIP(I),FLAG(I),ERR(I),GATE(I),I=1,60)
C     READ(4,*) (T(I),TRIP(I),FLAG(I),ERR(I),GATE(I),I=1,81)
      CALL PHYSOR(.75,5.0)
      CALL AREA2D(4.0,2.75)
      CALL INTAXS
```

```
      CALL XNAME('TIME(MINS)$',100)
      CALL YNAME('RESIDUAL & GATE MEAS$',100)
      CALL GRAF(XORIG,'SCALE',XMAX,YORIG,'SCALE',YMAX)
       CALL HEIGHT(0.15)
        CALL RLMESS('BUOY 1$',100,70.0,YMAX)
       CALL HEIGHT(0.15)
      CALL THKCRV (0.01)
      CALL CURVE(T,GATE,60,0)
C     CALL CURVE(T,GATE,81,0)
      CALL DASH
      CALL CURVE(T,ERR,60,0)
C     CALL CURVE(T,ERR,81,0)
      CALL RESET('DASH')
      CALL MARKER(2)
      CALL CURVE(T,TRIP,60,-1)
C     CALL CURVE(T,TRIP,81,-1)
      CALL RESET('MARKER')
      CALL ENDGR(0)
      XORIG=0.0
      XMAX=60.0
C     XORIG=60.0
C     XMAX=140.0
      YORIG=0.0
      YMAX=3.0
      READ(4,*) (T(I),TRIP(I),FLAG(I),ERR(I),GATE(I),I=1,60)
C     READ(4,*) (T(I),TRIP(I),FLAG(I),ERR(I),GATE(I),I=1,81)
      CALL PHYSOR(6.5,5.0)
      CALL AREA2D(4.0,2.75)
       CALL HEIGHT(0.15)
      CALL XNAME('TIME(MINS)$',100)
      CALL YNAME('RESIDUAL & GATE MEAS$',100)
      CALL GRAF(XORIG,'SCALE',XMAX,YORIG,'SCALE',YMAX)
       CALL HEIGHT(0.15)
      CALL RLMESS('BUOY 2$',100,70.,YMAX)
       CALL HEIGHT(0.15)
```

```
      CALL THKCRV (0.01)
      CALL CURVE(T,GATE,60,0)
C     CALL CURVE(T,GATE,81,0)
      CALL DASH
      CALL CURVE(T,ERR,60,0)
C     CALL CURVE(T,ERR,81,0)
      CALL RESET('DASH')
      CALL MARKER(2)
      CALL CURVE(T,TRIP,60,-1)
C     CALL CURVE(T,TRIP,81,-1)
      CALL RESET('MARKER')
      CALL ENDGR(0)
      XORIG=0.0
      XMAX=60.0
      XORIG=60.0
      XMAX=140.0
      YORIG=0.0
      YMAX=3.0
      READ(4,*) (T(I),TRIP(I),FLAG(I),ERR(I),GATE(I),I=1,60)
C     READ(4,*) (T(I),TRIP(I),FLAG(I),ERR(I),GATE(I),I=1,81)
      CALL PHYSOR(.75,1.0)
      CALL AREA2D(4.0,2.75)
       CALL HEIGHT(0.15)
      CALL XNAME('TIME(MINS)$',100)
      CALL YNAME('RESIDUAL & GATE MEAS$',100)
      CALL GRAF(XORIG,'SCALE',XMAX,YORIG,'SCALE',YMAX)
       CALL HEIGHT(0.15)
      CALL RLMESS('BUOY 3$',100,70.,YMAX)
       CALL HEIGHT(0.15)
      CALL LINESP (2.0)
      CALL LINES ('FREQ GATE$',IPAK,1)
      CALL LINES ('RESIDUAL$',IPAK,2)
      XW=XLEGND (IPAK,2)
      YW=YLEGND (IPAK,2)
      CALL LEGLIN
```

```
      CALL THKCRV (0.01)
      CALL CURVE(T,GATE,60,0)
C     CALL CURVE(T,GATE,81,0)
      CALL DASH
      CALL CURVE(T,ERR,60,0)
C     CALL CURVE(T,ERR,81,0)
      CALL RESET('DASH')
      CALL MARKER(2)
      CALL CURVE(T,TRIP,60,-1)
C     CALL CURVE(T,TRIP,81,-1)
      CALL RESET('MARKER')
      CALL LEGEND(IPAK,2,6.,2.25)
C
      CALL ENDPL(0)
      XORIG=0.0
      XMAX=60.0
C     XORIG=60.0
C     XMAX=140.0
      YORIG=0.0
      YMAX=1.0
      CALL NOBRDR
      CALL PAGE(11.,8.5)
      CALL HEIGHT(0.15)
      READ(4,*) (T(I),TRIP(I),FLAG(I),ERR(I),GATE(I),I=1,60)
      CALL PHYSOR(.75,5.0)
      CALL AREA2D(4.0,2.75)
      CALL INTAXS
      CALL XNAME('TIME(MINS)$',100)
      CALL YNAME('RESIDUAL & GATE MEAS$',100)
      CALL GRAF(XORIG,'SCALE',XMAX,YORIG,'SCALE',YMAX)
       CALL HEIGHT(0.15)
        CALL RLMESS('BUOY 1$',100,70.0,YMAX)
       CALL HEIGHT(0.15)
      CALL THKCRV (0.01)
      CALL CURVE(T,GATE,60,0)
```

```
C     CALL CURVE(T,GATE,81,0)
      CALL DASH
      CALL CURVE(T,ERR,60,0)
C     CALL CURVE(T,ERR,81,0)
      CALL RESET('DASH')
      CALL MARKER(2)
      CALL CURVE(T,TRIP,60,-1)
C     CALL CURVE(T,TRIP,81,-1)
      CALL RESET('MARKER')
      CALL ENDGR(0)
C     PLOT FOR BUOY 2
C     PLOT NEXT GRAPH
      XORIG=0.0
      XMAX=60.
C     XORIG=60.0
C     XMAX=140.0
      YORIG=0.0
      YMAX=1.0
      READ(4,*) (T(I),TRIP(I),FLAG(I),ERR(I),GATE(I),I=1,60)
C     READ(4,*) (T(I),TRIP(I),FLAG(I),ERR(I),GATE(I),I=1,81)
      CALL PHYSOR(6.5,5.0)
      CALL AREA2D(4.0,2.75)
       CALL HEIGHT(0.15)
      CALL XNAME('TIME(MINS)$',100)
      CALL YNAME('RESIDUAL & GATE MEAS$',100)
      CALL GRAF(XORIG,'SCALE',XMAX,YORIG,'SCALE',YMAX)
       CALL HEIGHT(0.15)
      CALL RLMESS('BUOY 2$',100,10.,YMAX)
C     CALL RLMESS('BUOY 2$',100,70.,YMAX)
       CALL HEIGHT(0.15)
      CALL THKCRV (0.01)
      CALL CURVE(T,GATE,60,0)
C     CALL CURVE(T,GATE,81,0)
      CALL DASH
      CALL CURVE(T,ERR,60,0)
```

```
C     CALL CURVE(T,ERR,81,0)
      CALL RESET('DASH')
      CALL MARKER(2)
      CALL CURVE(T,TRIP,60,-1)
C     CALL CURVE(T,TRIP,81,-1)
      CALL RESET('MARKER')
C     CALL LEGEND(IPAK,2,1.,2.25)
      CALL ENDGR(0)
C     PLOT FOR BUOY 3
C     PLOT NEXT GRAPH
      XORIG=0.0
      XMAX=60.
C     XORIG=60.0
C     XMAX=140.0
      YORIG=0.0
      YMAX=1.0
      READ(4,*) (T(I),TRIP(I),FLAG(I),ERR(I),GATE(I),I=1,60)
C     READ(4,*) (T(I),TRIP(I),FLAG(I),ERR(I),GATE(I),I=1,81)
      CALL PHYSOR(.75,1.0)
      CALL AREA2D(4.0,2.75)
C     CALL INTAXS
       CALL HEIGHT(0.15)
      CALL XNAME('TIME(MINS)$',100)
      CALL YNAME('RESIDUAL & GATE MEAS$',100)
C     CALL CROSS
      CALL GRAF(XORIG,'SCALE',XMAX,YORIG,'SCALE',YMAX)
       CALL HEIGHT(0.15)
C     CALL RLMESS('BUOY 3$',100,10.,YMAX)
      CALL RLMESS('BUOY 3$',100,70.,YMAX)
       CALL HEIGHT(0.15)
      CALL LINESP (2.0)
      CALL LINES ('BRG GATE$',IPAK,1)
      CALL LINES ('RESIDUAL$',IPAK,2)
      CALL LINES ('EXCEED$',IPAK,3)
      XW=XLEGND (IPAK,2)
```

```
      YW=YLEGND (IPAK,2)
C     CALL MYLEGN ('BUOY 3$',100)
      CALL LEGLIN
      CALL THKCRV (0.01)
      CALL CURVE(T,GATE,60,TRIP)
C     CALL CURVE(T,GATE,81,0)
      CALL DASH
      CALL CURVE(T,ERR,60,0)
C     CALL CURVE(T,ERR,81,0)
      CALL RESET('DASH')
      CALL MARKER(2)
      CALL CURVE(T,TRIP,60,-1)
C     CALL CURVE(T,TRIP,81,-1)
      CALL RESET('MARKER')
      CALL LEGEND(IPAK,2,6.,2.25)
C
      CALL ENDPL(0)
      CALL DONEPL
      STOP
      END
```

LIST OF REFERENCES

1. Naval Underwater Systems Center Report 5766, Passive Target Motion Analysis in a Maneuvering Target Environment, by B. W. Guimond, 16 May 1978.

2. Gelb, A., et al, Applied Optimal Estimation, 8th ed., The M.I.T. Press, 1984.

3. Mitschang, G. A., An Application of Nonlinear Filtering Theory to Passive Target Location and Tracking, Ph. D. Dissertation, Naval Postgraduate School, Monterey, California, June 1974.

4. Naval Postgraduate School Report NPS-62Ts77071, Advances in Passive Target Tracking, by H. A. Titus, et al, Naval Postgraduate School, Monterey, California, June 1977.

5. Gauss, J. A., Passive Acoustic Tracking Using Nonlinear Estimation Theory, Ph. D. Dissertation, Naval Postgraduate School, Monterey, California, March 1980.

6. Sorenson, H. W., "Kalman Filtering Techniques," Advances in Control Systems, Vol. 3, Academic Press, 1966.

7. Jazwinski, A. H., Stochastic Processes and Filtering Theory, Academic Press, 1970.

8. Kirk, D. E., Optimal Estimation: An Introduction to the Theory and Applications, Naval Postgraduate School, Monterey, California, 1975, (unpublished).

9. Helstrom, C. W., Probability and Stochastic Processes for Engineers, Macmillan, Inc., 1984.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Department Chairman, Code 62 Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	[illegible]
4.	Dr. H. [illegible]. Titus, Code 62Ts Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1
5.	LCDR George E. Olcovich, USN 2087 N. Via Tomas, Camarillo, California 93010	5

219310

Thesis
03675 Olcovich
c.1 Passive acoustic target motion analysis.