ESTIMATION OF SONOBUOY POSITION RELATIVE TO AN AIRCRAFT USING EXTENDED KALMAN FILTERS

Nicholas Mason Brownsberger

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

Estimation of Sonobuoy Position
Relative to an Aircraft
Using Extended Kalman Filters

by

Nicholas Mason Brownsberger

September 1979

Thesis Advisor: D. J. Collins

Approved for public release; distribution unlimited

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)

Estimation of Sonobuoy Position
Relative to an Aircraft
Using Extended Kalman Filters

5. TYPE OF REPORT & PERIOD COVERED

Master's and Engineer's
Thesis; September 1979

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)

Nicholas Mason Brownsberger

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS

Naval Postgraduate School
Monterey, California 93940

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS

Naval Postgraduate School
Monterey, California 93940

12. REPORT DATE

September 1979

13. NUMBER OF PAGES

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

Naval Postgraduate School
Monterey, California 93940

15. SECURITY CLASS. (of this report)

Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

SRS (Sonobuoy Reference System)
Kalman Filter
Sonobuoy

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

In airborne anti-submarine warfare there is a need to more accurately determine the positions of sonobuoys on the surface of the water. This report develops two algorithms which employ extended Kalman filters to determine estimated position. The bearing from the aircraft to the sonobuoy is the primary measurement. Range information is not available. The first algorithm is a six-state filter which was reduced from the

13-state system developed by the Orincon Corporation. Its states include relative position, relative velocity, and inertial misalignments. The second algorithm includes two cascaded Kalman filters. The primary two-state filter estimates sonobuoy position. A secondary filter estimates drift from information obtained from the primary filter. Both algorithms successfully estimated sonobuoy position for simulated aircraft data. The effect of aircraft-to-sonobuoy range, the frequency of measurement, and changes in altitude are also analyzed.

Estimation of Sonobuoy Position
Relative to an Aircraft
Using Extended Kalman Filters

by

Nicholas Mason Brownsberger
Lieutenant, United States Navy
B.S., United States Naval Academy, 1972

Submitted in partial fulfillment of the
requirements for the degrees of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING
AND
AERONAUTICAL ENGINEER

from the

NAVAL POSTGRADUATE SCHOOL
September 1979

ABSTRACT

In airborne anti-submarine warfare there is a need to more accurately determine the positions of sonobuoys on the surface of the water. This report develops two algorithms which employ extended Kalman filters to determine estimated position. The bearing from the aircraft to the sonobuoy is the primary measurement. Range information is not available. The first algorithm is a six-state filter which was reduced from the 13-state system developed by the Orincon Corporation. Its states include relative position, relative velocity, and inertial misalignments. The second algorithm includes two cascaded Kalman filters. The primary two-state filter estimates sonobuoy position. A secondary filter estimates drift from information obtained from the primary filter. Both algorithms successfully estimated sonobuoy position for simulated aircraft data. The effect of aircraft-to-sonobuoy range, the frequency of measurement, and changes in altitude are also analyzed.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENT

I would like to thank Dr. Birnbaum and Ms. Peggy Pembroke of NADC for their responsiveness in answering questions and supplying computer information. And to Dr. Collins I would like to express my gratitude for his help and guidance without which this thesis could not have been completed. To my wife I would like to offer my special thanks and love for her patience, understanding, and fortitude during a time which tested both of us.

Nick Brownsberger

I. INTRODUCTION

The P3 Orion is the U.S. Navy´s primary long range anti-submarine warfare (ASW) aircraft. It is outfitted with equipment which allows it to search for, locate, and track submarines. The aircraft carries a Univac digital computer, the CP 901, which performs much of the navigational and tactical plotting chores. The primary sensor used by this and many other ASW aircraft is an airdropped listening device known as a sonobuoy. The Orion generally deploys several sonobuoys (4 to 20) in patterns which can cover a thousand square miles of ocean while searching for the submarine. Once contact has been made, these sonobuoys provide information which locates the submarine. The target can then be tracked (or attacked if required) until the Orion´s mission is complete.

The sonobuoy is dropped from the aircraft at the geographical location designated by the aircrew. Once in the water the sonobuoy floats and deploys a hydrophone to depths varying from 50 to 500 feet. The information picked up is transmitted back to the aircraft where it is analyzed. This information can consist of the relative intensity of target noise, bearings and sometimes ranges to the target, all of which are used to fix the current position of the submarine. In order to maintain close, accurate tracking and be able to launch an attack, the submarine´s position must

be accurate to within several hundred yards. The information received from the sonobuoys has some error inherent in the nature of the measurements made by the hydrophone. Sonobuoy position error also contributes significantly to the submarine tracking inaccuracies.

Historically, the positions of the sonobuoys were determined and updated by "mark-on-top"s. This required the aircraft to home on the transmitting sonobuoy until the buoy was overflown. At that instant the aircraft's position was entered into the on-board computer which slewed the buoy to this updated position. After many of these updates the computer was able to develop a bias which was applied to the sonobuoy positions in the computer effectively allowing them to drift. The method had several disadvantages. Error in the updated position was at least as great as the aircraft altitude at the time of the "mark" which could vary from 300 to 20,000 feet. The accuracy also depended on the consistency of the several pilots who might be making the mark-on-tops during the flight. The updating was not continuous in that it was several minutes between consecutive marks on one buoy at best, and more likely 30 to 60 minutes. Not all buoys were even updated. Furthermore, this method required the aircraft to overfly the submarine many times in order to make the mark-on-tops. This should be avoided.

The purpose of this thesis was to investigate some alternative methods for accurately fixing the position of

sonobuoys. They should allow the aircraft to stand-off from the sonobuoy field and still produce more accurate fixing than the historical method provided. The Naval Air Development Center (NADC) at Warminster, PA. had already partially developed such a system. This thesis was undertaken in support of their work but was conducted independently. Their system, the Sonobuoy Reference System (SRS), was already installed on the aircraft and had the capability of measuring the relative bearing to any transmitting sonobuoy. Additional information available for use included aircraft heading, altitude, and airspeed as well as doppler velocity and drift angle. Also, the aircraft's Inertial Navigational System (INS) provided geographical position although the Schuler cycle and inertial drifts could make this position several nautical miles in error. On the other hand, aircraft-to-sonobuoy range and sea surface drift information were assumed not to be available. An attempt was made to determine sonobuoy position at least relative to the aircraft with a less accurate geographical position as a secondary objective. Kalman filtering techniques were used based primarily on measurements of bearing from the aircraft to the sonobuoy.

II. FILTERS

Kalman filtering is a recursive technique for estimating the state of a system. It was developed in the 1960's by R. E. Kalman and improved upon previous methods by Wiener and others. The Wiener filter is based on frequency domain designs which are statistically optimal but are only applicable to stationary processes. The Kalman filter is based on state-space, time domain formulations and is especially suited to digital computers. From a simplistic, one-dimensional point of view, the Kalman filter recursively averages noisy measurements to provide a more precise estimate of the actual value.

Assume that a system can be linearly modeled with state equations in matrix form as

$$X_k = \underline{\Phi}_{k-1} X_{k-1} + \Lambda_{k-1} U_{k-1} + \Gamma_{k-1} W_{k-1} \qquad (1)$$

where X_k represents the states of the system at the Kth interval. (Only the discrete case was be considered in this study.) $\underline{\Phi}_k$ is the transition matrix and is used to "propagate" the system from K to K+1. U_k represents the control input to the system and W_k represents white, guassian noise with Ø mean and Q_k variance, written $N(Ø, Q_k)$. Measurements are required to update the system and are

described by the linear matrix equation

$$Z_k = H_k X_k + V_k \tag{2}$$

where Z_k is the measurement. H_k describes the relationship between the states and the measurement and V_k is measurement noise described by $N(\emptyset, R_k)$.

The discrete Kalman filter equations are:

Propagate

$$\hat{X}_k(-) = \Phi_{k-1} \hat{X}_{k-1}(+) \tag{3}$$

$$P_k(-) = \Phi_{k-1} P_{k-1}(+) \Phi^T_{k-1} + Q_{k-1} \tag{4}$$

Update

$$G_k = P_k(-) H^T_k \left[H_k P_k(-) H^T_k + R_k \right]^{-1} \tag{5}$$

$$\hat{X}_k(+) = \hat{X}_k(-) + G_k \left[Z_k - H_k \hat{X}_k(-) \right] \tag{6}$$

$$P_k(+) = \left[I - G_k H_k \right] P_k(-) \tag{7}$$

where

$\hat{X}_k$ = estimate of the state X_k

P_k = covariance

G_k = Kalman gain

The estimate prior to the measurement is denoted by $\hat{X}_K(-)$ and can be updated after the measurement to a new estimate denoted $\hat{X}_K(+)$. This notation is shown in figure 1. The covariance matrix provides a statistical measure of the uncertainty of $\hat{X}$. Consider a 2 x 2 covariance matrix where the error $\tilde{x}$ in the estimate is defined as $\hat{x} - x$:

$$P = \begin{bmatrix} E(\tilde{x}_1^2) & E(\tilde{x}_1 \tilde{x}_2) \\ E(\tilde{x}_1 \tilde{x}_2) & E(\tilde{x}_2^2) \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & \sigma_1 \sigma_2 \\ \sigma_1 \sigma_2 & \sigma_2^2 \end{bmatrix} \qquad (8)$$

The diagonal elements represent the mean square errors of the corresponding state variables x_1 and x_2. The off-diagonal elements are indicators of cross-correlation between the states. The Kalman gain G is an optimal gain choosen so as to minimize the sum of the diagonal terms of the covariance matrix.

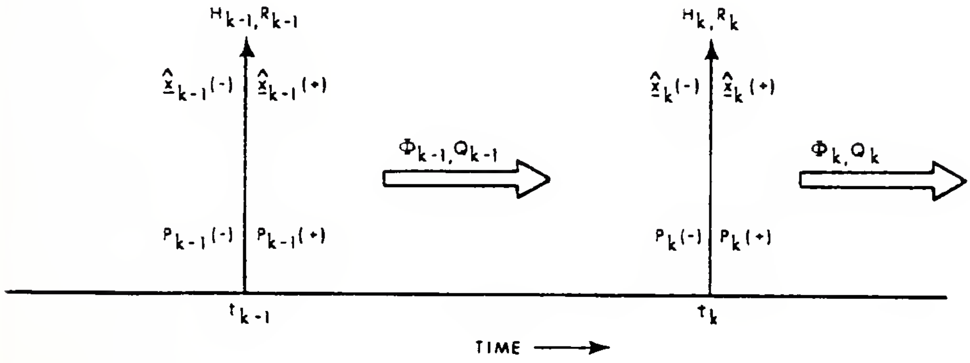


Figure 1. Kalman filter notation

If either the system model or the measurement model is non-linear, then non-linear filters must be used. Let the non-linear system be described by

$$X_k = \phi(X_{k-1}) + W_{k-1} \quad ; \; W_k \sim N(\emptyset, Q_k) \qquad (9)$$

$$Z_k = h(X_k) + V_k \quad ; \; V_k \sim N(\emptyset, R_k) \qquad (1\emptyset)$$

where $\phi(\hat{X}_{k-1})$ represents the non-linear equations of the system. They can be linearized by a Taylor series expansion around the latest estimate of the state to obtain $\Phi(\hat{X}_{k-1})$. Likewise, $h(X_k)$ represents the non-linear measurement equations and by a Taylor series expanicns yields $H(\hat{X}_k)$.

$$\phi(X_k) = \phi(\hat{X}_k) + \Phi(\hat{X}_k)\left[X_k - \hat{X}_k\right] + \ldots \qquad (11)$$

$$h(X_k) = h(\hat{X}_k) + H(\hat{X}_k)\left[X_k - \hat{X}_k\right] + \ldots \qquad (12)$$

where

$$\Phi(\hat{X}_k) = \frac{\partial}{\partial x}\left[\phi(X_k)\right]_{x=\hat{x}} \qquad (13)$$

$$H(\hat{X}_k) = \frac{\partial}{\partial x}\left[h(X_k)\right]_{x=\hat{x}} \qquad (14)$$

Second order Taylor series terms are neglected. Now, the extended Kalman filter can be implemented with the following equations:

Propagate

$$\hat{X}_k(-) = \phi(\hat{X}_{k-1}) \tag{15}$$

$$P_k(-) = \Phi(\hat{X}_{k-1}(-))\ P_{k-1}(+)\ \Phi(\hat{X}_{k-1}(-))^T + Q_{k-1} \tag{16}$$

Update

$$G = P_k(-)\ H(\hat{X}_k(-))^T \left[H(\hat{X}_k(-))\ P_k(-)\ H(\hat{X}_k(-))^T + R_k\right]^{-1} \tag{17}$$

$$\hat{X}_k(+) = \hat{X}_k(-) + G_k\left[Z_k - H(\hat{X}_k(-))\right] \tag{18}$$

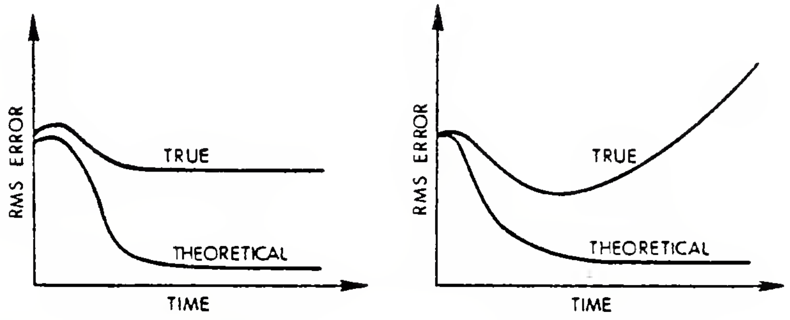
$$P_k(+) = \left[I - G_k H(\hat{X}_k(-))\right]\ P_k(-) \tag{19}$$

Higher order filters can be used if the linearization errors are large. The second order Kalman filter employs one more term in the Taylor series expansion by modifing the update equations of the extended Kalman filter to account for this term. The iterated extended Kalman filter uses the same equations as does the extended Kalman filter. However, the calculations are repeated, each time linearizing about the most recent estimate, until there is little further improvement with each new iteration. The iterated extended filter can greatly reduce the errors due to non-linearities, more so than the second order filter.

Kalman filters should be based on correctly modeled systems and accurate noise statistics to ensure proper

performance. This is not always possible either due to ignorance about the system or lack of sufficient statistical information. A filter which is not operating properly may diverge. Apparent divergence describes that situation where the true estimation errors are larger than the filter predicted errors although they are bounded. True divergence is characterized by errors which continue to grow with time and eventually become infinite. These divergence phenomenon are depicted in figure 2.

There are several ways to overcome the divergence problem when the modeling is not completely accurate such as adding fictitious noise. This allows the filter a little more freedom to adjust to whatever modeling inconsistencies may exist, but makes the filter estimate appear more erratic. Another method which helps overcome divergence is finite memory filtering. Since Kalman gains tend to grow smaller and smaller as time passes, they may reach a point



(a) Apparent Divergence

(b) True Divergence

Figure 2. Divergence

where new measurement information has no effect on the estimates. Finite memory filtering effectively eliminates old data which is no longer useful by keeping the gains significant. This is sometimes called a "moving window".

Some simplifing techniques used in Kalman filters include precomputed gains. Although forfeiting the optimal Kalman gains, this has the advantage of reducing computational burden. More importantly, the gains can be controlled to overcome modeling weaknesses if necessary. Another technique which can be used when more than one measurement is provided to a filter is processing them one at a time. This avoids taking the inverse of more than a scalar when computing the updated covariance matrix $P_K(+)$. This is possible if the simultaneous measurements are considered to be taken sequentially over a zero time span.

III. MODELS

A. GENERAL APPROACHES TO SYSTEM MODELING

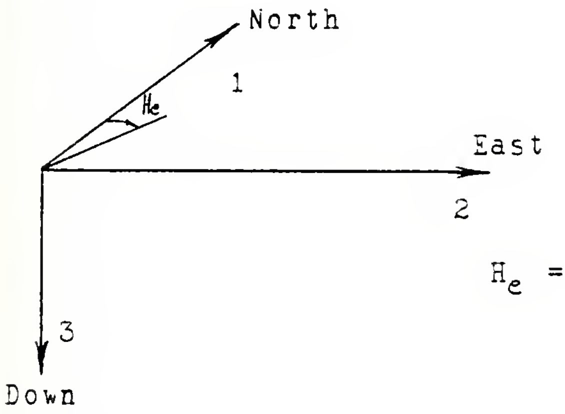
The aircraft-sonobuoy system must be modeled in state-space for use with the Kalman filter. There are at least two approaches to the modeling of this problem depending on the point of view. One intuitive approach is to assume the sonobuoy drifts at a constant velocity and then use aircraft-to-sonobuoy bearing measurements to locate the sonobuoy. The states become sonobuoy position and sonobuoy velocity. Unfortunately, this problem is not observable. The bearing measurements provide only information about position; there is no rate of change information in the bearings themselves. In addition, the aircraft must maintain a track of its geographical position between updates in order to determine the next expected measurement for the Kalman filter. As mentioned before, this aircraft position is subject to non-linear as well as linear navigational drifts which are not taken into account in this model. However, an observable system can be obtained by reducing the number of states to sonobuoy position only and introducing fictitious process noise to account for the drift. This noise effectively allows sonobuoy position to update so as to keep up with the drift. One approach developed by this thesis is a variation of this concept.

Another approach considers only the relative position of the sonobuoy with respect to the aircraft. Relative velocities and relative accelerations must be taken into account and these change radically as the aircraft flies in the tactical situation. Sonobuoy position is not obtained directly. (When only the word "position" is used it will indicate the position relative to an earth fixed coordinate system, such as latitude and longitude. "Relative position" will always mean the location with respect to an aircraft fixed coordinate system.)

There are some other considerations which should be addressed. The sonobuoy drift is generally slow (less than 5 NM/Hr most of the time) and constant. It is not unreasonable to assume that the entire sonobuoy field drifts at the same velocity. Another point is that aircraft navigational drift can not be distinguished from sonobuoy drift. In other words, the drift that is preceived by the aircraft is the combination of sonobuoy drift and aircraft navigational drift. If this navigational drift is linear and not excessive it causes few problems. However, non-linear navigational drifts, such as the Schuler cycle, can cause large errors.

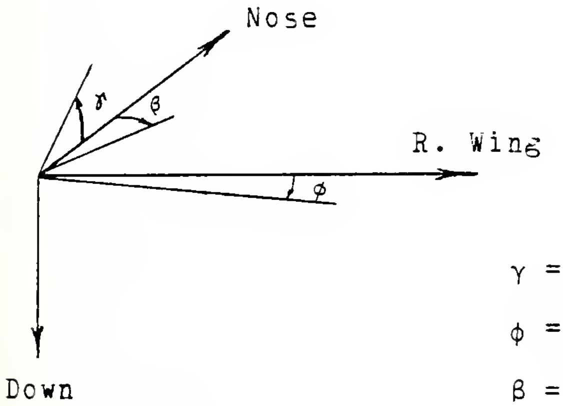
B. THE COORDINATE SYSTEM

A right-handed coordinate system was chosen as depicted in figures 3 and 4. This is a departure from the work of Orincon Corporation which is described in the next section. This system allows all angles to be measured positive in the direction they are normally defined, i.e., aircraft heading measured clockwise from north. It also coincides with the usual aerodynamic coordinate system.



H_e = Aircraft heading, + clockwise

Figure 3. Earth fixed



γ = Aircraft pitch, + up

ϕ = Aircraft roll, + r. wing down

β = Doppler drift angle, + r. drift

Figure 4. Aircraft fixed

C. THE SIX-STATE SYSTEM

Orincon Corporation of La Jolla, CA., completed a technical report in December of 1978 concerning the Sonobuoy Reference System. They were contracted by NADC to investigate the effect of errors in the aircraft navigational system on the performance of the sonobuoy tracking procedure. As a result of their work they developed a 13-state mathematical model which successfully determined sonobuoy positions in spite of these navigational errors. The complete state vector was

$$X^T = \left[\Delta x_1 \; \Delta x_2 \; \Delta v_1 \; \Delta v_2 \; x_1 \; x_2 \; v_1 \; v_2 \; \psi_1 \; \psi_2 \; \psi_3 \; m_1 \; m_2\right] \quad (20)$$

where

Δx = relative position of the sonobuoy from the aircraft;

Δv = relative velocity of the sonobuoy from the aircraft;

x = position of the sonobuoy on the ocean surface;

v = velocity of the sonobuoy on the ocean surface;

ψ = inertial system's platform azimuth alignment errors

m = inertial system's gyro drift rates assumed constant.

(Subscripts 1, 2, and 3 represent North, East, and down respectively.) The model was developed using state equations which described sonobuoy motion relative to the aircraft and relative to the ocean's surface. These were derived by

Orincon from

$$\Delta X = X - (X_a^I + E) \tag{21}$$

and other mechanization equations to obtain

$$\begin{aligned}
\Delta \dot{x}_1 &= \Delta v_1 \\
\Delta \dot{x}_2 &= \Delta v_2 \\
\Delta \dot{v}_1 &= -\ddot{R}_1 + \omega^2 (x_1 - x_{a1}^I - \Delta x_1) - \psi_3 A_2 + \psi_2 g \\
\Delta \dot{v}_2 &= -\ddot{R}_2 + \omega^2 (x_2 - x_{a2}^I - \Delta x_2) - \psi_3 A_1 - \psi_1 g \\
\dot{x}_1 &= v_1 \\
\dot{x}_2 &= v_2 \\
\dot{v}_1 &= \emptyset \\
\dot{v}_2 &= \emptyset
\end{aligned} \tag{22}$$

where

X_a^I = aircraft inertial navigation system position

E = aircraft inertial navigation system error

R = vector from the center of the earth to the aircraft

g = acceleration due to gravity (assumed constant)

ω = Schuler frequency $\sqrt{g/R}$

A = aircraft accelerometer outputs

The sonobuoys were assumed to drift at a constant velocity and the aircraft was assumed to fly at a constant altitude. The inertial alignment errors were described by

$$\begin{aligned} \dot{\psi}_1 &= \psi_2\Omega \sin\lambda - \psi_3\Omega\cos\lambda + m_1 \\ \dot{\psi}_2 &= -\psi_1\Omega \sin\lambda + m_2 \\ \dot{\psi}_3 &= \psi_1\Omega \cos\lambda \end{aligned} \tag{23}$$

where

Ω = earth rate

λ = latitude

There were several sources of measurement information. Aircraft to sonobuoy bearing was available from the aircraft interferometer system in the form of $\cos\theta$ where θ is the angle between the direction to the sonobuoy and the base line of the antenna. Doppler velocity and drift angle from the aircraft doppler radar system provided information about Δv_1 and Δv_2. The only other relevant source of measurement information would have come from GPS (Global Positioning System) which would have provided error free aircraft position.

Orincon performed a numerical observability analysis on this system based on the observability matrix M_k given by

$$M_k = \sum_{i=1}^{N} \Phi_k^T H_k^T R^{-1} H_k \Phi_k \tag{24}$$

If this matrix is positive definite for $N \geq 2$ then the system is considered observable. In addition, an eigenvalue analysis provided information as to the conditioning of the system. The result of their analysis indicated that the

system was not fully observable with measurements from the interferometer and doppler systems alone. Only ten states were observable; sonobuoy position most likely was not observable and ψ_3, m_1, m_2, v_1 and v_2 were weakly observable at best. They also noted that "observability is reduced when the aircraft pursues a straight flight path". By removing sonobuoy positions and gyro drifts from the system they found the observability improved. Further, the system became fully observable if sonobuoy velocity was also removed from the state vector.

The Orincon report presented the results of a Monte-Carlo simulation on the 13-state system. However, they concluded that a reduced state filter made up of

$$X^T = \left[\Delta x_1 \ \Delta x_2 \ \Delta v_1 \ \Delta v_2 \ \psi_1 \ \psi_2 \right] \tag{25}$$

would provide a workable solution to the problem although they noted that it had a tendency toward divergence. It was partly the intention of this report to further the investigation of this reduced state filter.

The state equations for the reduced six-state filter can be obtained directly from Equations (22) and (23).

$$\begin{aligned}
\Delta\dot{x}_1 &= \Delta v_1 \\
\Delta\dot{x}_2 &= \Delta v_2 \\
\Delta\dot{v}_1 &= -\ddot{R}_1 + \omega^2(x_1 - x^I_{a1} - \Delta x_1) + \psi_2 g \\
\Delta\dot{v}_2 &= -\ddot{R}_2 + \omega^2(x_2 - x^I_{a2} - \Delta x_2) - \psi_1 g \\
\dot{\psi}_1 &= \psi_2 \sin\lambda \\
\dot{\psi}_2 &= -\psi_1 \sin\lambda
\end{aligned} \qquad (26)$$

where ψ_3, m_1, and m_2 are set to zero. In matrix form

$$\dot{X} = \begin{bmatrix} \Delta\dot{x}_1 \\ \Delta\dot{x}_2 \\ \Delta\dot{v}_1 \\ \Delta\dot{v}_2 \\ \dot{\psi}_1 \\ \dot{\psi}_2 \end{bmatrix} = \left[\begin{array}{cccc|cc} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ -\omega^2 & 0 & 0 & 0 & 0 & -g \\ 0 & -\omega^2 & 0 & 0 & g & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & -\Omega\sin\lambda \\ 0 & 0 & 0 & 0 & \Omega\sin\lambda & 0 \end{array}\right] \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta v_1 \\ \Delta v_2 \\ \psi_1 \\ \psi_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -\ddot{R}_1 + \omega^2(x_1 - x^I_{a1}) \\ -\ddot{R}_2 + \omega^2(x_2 - x^I_{a2}) \\ 0 \\ 0 \end{bmatrix}$$

which can be abreviated

$$\dot{X} = \left[\begin{array}{c|c} A^{\Delta} & A^{\Delta\psi} \\ \hline 0 & A^{\psi} \end{array}\right] X + B \qquad (27)$$

Expressed in discrete time the state equation is

$$X_{k+1} = \Phi X_k + \Lambda_k + W_k \qquad (28)$$

The discrete formulation of the plant matrix can be obtained from

$$\Phi = \mathcal{L}^{-1}\left\{\left|sI - A\right|^{-1}\right\} = e^{At} \tag{29}$$

Developing this

$$\left[sI - A\right] = \left[\begin{array}{cccc|cc} s & 0 & 1 & 0 & 0 & 0 \\ 0 & s & 0 & 1 & 0 & 0 \\ \omega^2 & 0 & s & 0 & 0 & g \\ 0 & \omega^2 & 0 & s & -g & 0 \\ \hline 0 & 0 & 0 & 0 & s & \Omega\sin\lambda \\ 0 & 0 & 0 & 0 & -\Omega\sin\lambda & s \end{array}\right] \tag{30}$$

from which the determinant of the system is

$$\left|sI - A\right| = (s^2 + \omega^2)^2(s^2 + \Omega^2\sin^2\lambda)^2 \tag{31}$$

The inverse of this matrix and its Laplace transform are presented on the following pages as Equations (32) and (33) respectively.

$$\left[sI - A\right]^{-1} = \left[\begin{array}{cccc|cc}
\frac{s}{s^2 + \omega^2} & 0 & \frac{1}{s^2 + \omega^2} & 0 & \frac{-g\ \Omega\sin\lambda}{(s^2 + \omega^2)(s^2 + \Omega^2\sin^2\lambda)} & \frac{-g\ s}{(s^2 + \omega^2)(s^2 + \Omega^2\sin^2\lambda)} \\
0 & \frac{s}{s^2 + \omega^2} & 0 & \frac{1}{s^2 + \omega^2} & \frac{g\ s}{(s^2 + \omega^2)(s^2 + \Omega^2\sin^2\lambda)} & \frac{-g\ \sin\lambda}{(s^2 + \omega^2)(s^2 + \Omega^2\sin^2\lambda)} \\
\frac{-\omega}{s^2 + \omega^2} & 0 & \frac{s}{s^2 + \omega^2} & 0 & \frac{-g\ s\ \Omega\sin\lambda}{(s^2 + \omega^2)(s^2 + \Omega^2\sin^2\lambda)} & \frac{-g\ s^2}{(s^2 + \omega^2)(s^2 + \Omega^2\sin^2\lambda)} \\
0 & \frac{-\omega^2}{s^2 + \omega^2} & 0 & \frac{s}{s^2 + \omega^2} & \frac{g\ s^2}{(s^2 + \omega^2)(s^2 + \Omega^2\sin^2\lambda)} & \frac{-g\ s\ \Omega\sin\lambda}{(s^2 + \omega^2)(s^2 + \Omega^2\sin^2\lambda)} \\
\hline
0 & 0 & 0 & 0 & \frac{s}{s^2 + \Omega^2\sin^2\lambda} & \frac{-\Omega\sin\lambda}{s^2 + \Omega^2\sin^2\lambda} \\
0 & 0 & 0 & 0 & \frac{\Omega\sin\lambda}{s^2 + \Omega^2\sin^2\lambda} & \frac{s}{s^2 + \Omega^2\sin^2\lambda}
\end{array}\right] \qquad (32)$$

$$\Phi = \left[\begin{array}{cccc|cc}
\cos\omega t & 0 & \frac{1}{\omega}\sin\omega t & 0 & \frac{-g\left[\frac{\Omega\sin\lambda}{\omega}\sin\omega t-\sin(\Omega\sin\lambda)t\right]}{\Omega^2\sin^2\lambda-\omega^2} & \frac{-g\left[\cos\omega t-\cos(\Omega\sin\lambda)t\right]}{\Omega^2\sin^2\lambda-\omega^2} \\
0 & \cos\omega t & 0 & \frac{1}{\omega}\sin\omega t & \frac{g\left[\cos\omega t-\cos(\Omega\sin\lambda)t\right]}{\Omega^2\sin^2\lambda-\omega^2} & \frac{-g\left[\frac{\Omega\sin\lambda}{\omega}\sin\omega t-\sin(\Omega\sin)t\right]}{\Omega^2\sin^2\lambda-\omega^2} \\
-\omega\sin\omega t & 0 & \cos\omega t & 0 & \frac{-g\Omega\sin\lambda\left[\cos\omega t-\cos(\Omega\sin\lambda)t\right]}{\Omega^2\sin^2\lambda-\omega^2} & \frac{g\left[\omega\sin\omega t-\Omega\sin\lambda\sin(\Omega\sin\lambda)t\right]}{\Omega^2\sin^2\lambda-\omega^2} \\
0 & -\omega\sin\omega t & 0 & \cos\omega t & \frac{-g\left[\omega\sin\omega t-\Omega\sin\lambda\sin(\Omega\sin\lambda)t\right]}{\Omega^2\sin^2\lambda-\omega^2} & \frac{-g\Omega\sin\lambda\left[\cos\omega t-\cos(\Omega\sin\lambda)t\right]}{\Omega^2\sin^2\lambda-\omega^2} \\
\hline
0 & 0 & 0 & 0 & \cos(\Omega\sin\lambda)t & -\sin(\Omega\sin\lambda)t \\
0 & 0 & 0 & 0 & \sin(\Omega\sin\lambda)t & \cos(\Omega\sin\lambda)t
\end{array}\right] \qquad (33)$$

The discrete formulation of the continuous matrix B is

$$\Lambda_k = \int_{t_k}^{t_{k+1}} e^{At}\, d\lambda\, B \qquad \text{where } \lambda = \Delta t - \tau \tag{34}$$

Hence

$$\Lambda_k = \begin{bmatrix} \{-\frac{\ddot{R}_1}{\omega^2} + (x_1 - x^I_{a1})\}\ (1 - \cos\omega\Delta t) \\ \{-\frac{\ddot{R}_2}{\omega^2} + (x_2 - x^I_{a2})\}\ (1 - \cos\omega\Delta t) \\ \{-\frac{\ddot{R}_1}{\omega} + \omega(x_1 - x^I_{a1})\}\ \sin\omega\Delta t \\ \{-\frac{\ddot{R}_2}{\omega} + \omega(x_2 - x^I_{a2})\}\ \sin\omega\Delta t \\ \emptyset \\ \emptyset \end{bmatrix} \tag{35}$$

The noise term is $W_k \sim N(\emptyset, Q)$. The variances are estimated to be .Ø1 NM for relative position, 1.Ø (NM/Hr) for relative velocity, and .ØØ1 radians for inertial misalignments. As a result, the following constant diagonal matrix is used for Q:

$$Q = \begin{bmatrix} .\emptyset 1 & & & & & \\ & .\emptyset 1 & & & 0 & \\ & & 1.\emptyset & & & \\ & & & 1.\emptyset & & \\ & 0 & & & .\emptyset\emptyset 1 & \\ & & & & & .\emptyset\emptyset 1 \end{bmatrix} \tag{36}$$

The same measurement information is available as before

in the larger system, namely interferometer bearings and doppler velocity and drift angle. These introduce non-linearities and must be modeled as in Equation (10).

$$Z_k = h(X_k) + V_k$$

A Taylor series expansion as in Equation (11) about the latest estimate of sonobuoy position provides a method of linearization. As a result

$$H = \left[\left.\frac{\partial z}{\partial \Delta x_1}\right|_{x=\hat{x}} \left.\frac{\partial z}{\partial \Delta x_2}\right|_{x=\hat{x}} \left.\frac{\partial z}{\partial \Delta v_1}\right|_{x=\hat{x}} \left.\frac{\partial z}{\partial \Delta v_2}\right|_{x=\hat{x}} \left.\frac{\partial z}{\partial \psi_1}\right|_{x=\hat{x}} \left.\frac{\partial z}{\partial \psi_2}\right|_{x=\hat{x}} \right] \tag{37}$$

To determine z and its partial derivatives, recall that

$$z_1 = \cos\theta = \overrightarrow{\Delta X} \cdot \overrightarrow{RB} \tag{38}$$

where $\overrightarrow{\Delta X}$ is the unit vector representation of a line from the aircraft to the sonobuoy and $\overrightarrow{RB}$ is the unit vector describing the base line between the antenna pairs of the interferometer.

$$\hat{z}_1 = \cos\theta\Big|_{x=\hat{x}} = \frac{\Delta\hat{x}_1}{|\Delta\hat{x}|} RB_1 + \frac{\Delta\hat{x}_2}{|\Delta\hat{x}|} RB_2 + \frac{\Delta\hat{x}_3}{|\Delta\hat{x}|} RB_3$$

$$\hat{z}_1 = \frac{\Delta\hat{x}_1 RB_1 + \Delta\hat{x}_2 RB_2 + h\ RB_3}{(\Delta\hat{x}_1{}^2 + \Delta\hat{x}_2{}^2 + h^2)^{\frac{1}{2}}} \tag{39}$$

where h is the aircraft altitude.

$$H = \left[\left. \frac{\partial z_1}{\partial \Delta x_1} \right|_{X=\hat{X}} \left. \frac{\partial z_1}{\partial \Delta x_2} \right|_{X=\hat{X}} \quad 0 \quad 0 \quad 0 \quad 0 \right] \tag{40}$$

where

$$\left. \frac{\partial}{\partial \Delta x_1}(z_1) \right|_{X=\hat{X}} = \frac{1}{(\Delta\hat{x}_1^2+\Delta\hat{x}_2^2+h^2)^{1/2}} \left[RB_1 - \frac{\Delta\hat{x}_1(\Delta\hat{x}_1 RB_1 + \Delta\hat{x}_2 RB_2 + hRB_3)}{(\Delta\hat{x}_1^2 + \Delta\hat{x}_2^2 + h^2)} \right] \tag{41}$$

$$\left. \frac{\partial}{\partial \Delta x_2}(z_1) \right|_{X=\hat{X}} = \frac{1}{(\Delta\hat{x}_1^2+\Delta\hat{x}_2^2+h^2)^{1/2}} \left[RB_2 - \frac{\Delta\hat{x}_2(\Delta\hat{x}_1 RB_1 + \Delta\hat{x}_2 RB_2 + hRB_3)}{(\Delta\hat{x}_1 + \Delta\hat{x}_2^2 + h^2)} \right] \tag{42}$$

The doppler velocity, V_d, is a measurement of aircraft speed relative to the ocean surface and doppler drift angle, β, is a measure of the angle between aircraft track and aircraft heading. These measurements can be related to the state vector by assuming the sonobuoys drift at the same velocity as the ocean surface. If this is true then

$$z_2 = V_d = (\Delta v_1^2 + \Delta v_2^2)^{1/2} \tag{43}$$

Expanding in a Taylor series yields

$$z_2 = (\Delta\hat{v}_1^2 + \Delta\hat{v}_2^2)^{1/2} \tag{44}$$

$$H_2 = \left[0 \quad 0 \quad \left. \frac{\partial z_2}{\partial \Delta v_1} \right|_{X=\hat{X}} \quad \left. \frac{\partial z_2}{\partial \Delta v_2} \right|_{X=\hat{X}} \quad 0 \quad 0 \right] \tag{45}$$

where

$$\left. \frac{\partial z_2}{\partial \Delta v_1} \right|_{X=\hat{X}} = \frac{\Delta\hat{v}_1}{(\Delta\hat{v}_1^2 + \Delta\hat{v}_2^2)^{1/2}} \tag{46}$$

$$\left. \frac{\partial z_2}{\partial \Delta v_2} \right|_{X=\hat{X}} = \frac{\Delta\hat{v}_2}{(\Delta\hat{v}_1^2 + \Delta\hat{v}_2^2)^{1/2}} \tag{47}$$

Similarly,

$$z_3' = \beta = \text{Track} - \text{Heading}$$
$$= \tan^{-1}\left(\frac{\Delta v_2}{\Delta v_1}\right) - H_e \qquad (48)$$

or since the aircraft heading, H_e, is deterministic

$$z_3 = H_e + \beta = \tan^{-1}\left(\frac{\Delta v_2}{\Delta v_1}\right) \qquad (49)$$

$$H_3 = \left[\begin{matrix} 0 & 0 & \left.\frac{\partial z_3}{\partial \Delta v_1}\right|_{X=\hat{X}} & \left.\frac{\partial z_3}{\partial \Delta v_2}\right|_{X=\hat{X}} & 0 & 0 \end{matrix}\right] \qquad (50)$$

where

$$\left.\frac{\partial z_3}{\partial \Delta v_1}\right|_{X=\hat{X}} = \frac{-\Delta \hat{v}_2}{\Delta \hat{v}_1^2 + \Delta \hat{v}_2^2} \qquad (51)$$

$$\left.\frac{\partial z_3}{\partial \Delta v_2}\right|_{X=\hat{X}} = \frac{\Delta \hat{v}_1}{\Delta \hat{v}_1 + \Delta \hat{v}_2} \qquad (52)$$

Finally,

$$Z = \begin{bmatrix} \cos\theta \\ V_d \\ H_e - \beta \end{bmatrix} \qquad (53)$$

$$H = \begin{bmatrix} \frac{\partial(\cos\theta)}{x_1} & \frac{\partial(\cos\theta)}{x_2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\partial(v_d)}{\partial \Delta v_1} & \frac{\partial(v_d)}{\partial \Delta v_2} & 0 & 0 \\ 0 & 0 & \frac{\partial(H_e+\beta)}{\partial \Delta v_1} & \frac{\partial(H_e+\beta)}{\partial \Delta v_2} & 0 & 0 \end{bmatrix} \qquad (54)$$

The measurement noise, V_d, is described by N(Ø,R). It is assumed that the measurement noise R and the plant noise Q are white, gaussian, and uncorrelated. Based on information obtained from NADC, R was assigned the following values:

$$
\begin{aligned}
R_1 &= .0007 \text{ rad} && \text{(interferometer)} \\
R_2 &= 1.0 \text{ (NM/Hr)} && \text{(doppler velocity)} \qquad (55) \\
R_3 &= .01 \text{ deg} && \text{(doppler drift angle)}
\end{aligned}
$$

It should be pointed out that the doppler measurements are not always available. The doppler system freezes the last value of groundspeed and drift angle anytime the doppler radar is not receiving good information. This happens whenever the aircraft is above 5000′ altitude, or can happen when flying above a cloud cover or when the sea surface is too smooth. As a result, the doppler could be inaccurate during a significant portion of the flight.

The six-state system was programmed and tested for several conditions. Flow charts for significant portions of the program and the entire program listing are presented in Appendix A. The main program has two purposes: first, to control the simulation; and second, to read and prepare measurement data for the subroutine FILTER. This subroutine performs all the Kalman filter computations as specified in the above equations. The Carlson square root technique is used to ensure a positive definite covariance matrix. Also, in order to avoid inverting a matrix in the calculations for the Kalman gain, the three measurements are processed one at

a time. The results of the simulation are presented in section IV.B.

D. THE TWO-STATE SYSTEM

An attempt was made earlier on in this research to model sonobuoy motion as

$$\begin{aligned} X_{k+1} &= X_k + U_k \Delta t \\ U_{k+1} &= U_k \end{aligned} \tag{56}$$

with a state vector made up of x_1, x_2, u_1, and u_2. This system proved to be unobservable although the simple approach was appealing. Since sonobuoy drift rates are generally slow, another attempt was made to model the system without velocity. The equations reduced to

$$X_{k+1} = X_k + W_k \tag{57}$$

With only x_1 and x_2 as states, this system was observable. By describing large statistical values of system noise there would hopefully be enough freedom in the "update" to compensate for any sonobuoy drifts. Unfortunately, this approach was not completely successful; however, it was observed that the general direction of sonobuoy drift was correct. As a result of these investigations a technique of cascading Kalman filters was used to solve the tracking problem. The intent was to use results from one filter as measurement for a second filter. Then, the result of the

second filter could be used as a deterministic control input to better propagate the first filter.

Two Kalman filters are used in this approach. Figure 5 shows how they are related (or cascaded). The first filter models sonobuoy position and is based on the discrete state equation

$$X_k = X_k + U_k\Delta t + W_k \tag{58}$$

The state vector consists of the sonobuoy positions, x_1 and x_2. U_k is a deterministic input accounting for sonobuoy drift and W_k is system noise. The interferometer measurement, $\cos\theta$, is used to update this filter.

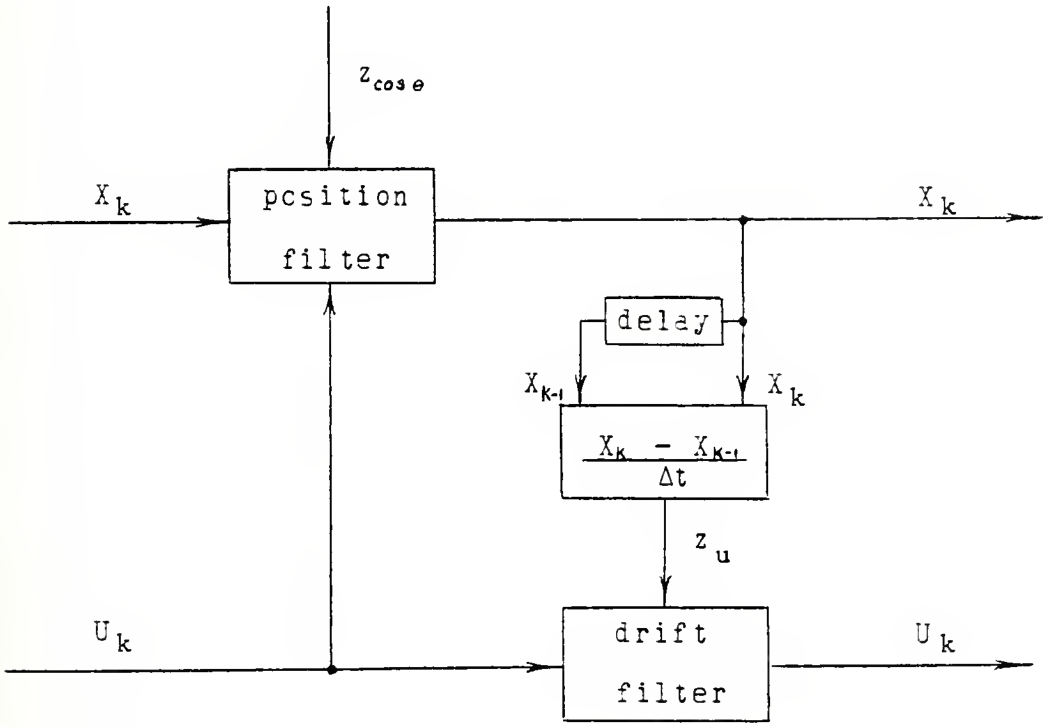


Figure 5. Cascaded Kalman filters.

The second filter models the sonobuoy drift with the state equation

$$U_{k+1} = U_k \tag{59}$$

As the sonobuoy's position changes with each update of the position filter, the velocity required to move from X_k to X_{k+1} in the time interval Δt can be calculated. This value is used as a measurement for the drift filter. All sonobuoys contribute to this filter which produces one overall estimate of drift for the sonobuoy field. This value is then used as an input to the position filter when the next interferometer measurement occurs.

The technique is understandably sensitive to the output of the drift filter. If the estimated drift is too much in error, the estimated positions will be affected. Divergence could result. For this reason, a simplified Kalman filter is used to estimate the drift. It is of the form

$$U_{k+1} = U_k + G_k(Z_k - U_k) \tag{60}$$

The gain is precomputed from 1/(n+1) where n is the total number of measurements and is limited to a value at least as great as Δt/(84 min.). This limit is reached after the first 84 minutes of tracking time. From that point on the gain remains constant effectively averaging the drift over one period of the Schuler cycle. It must be understood that

the drift this filter estimates consists not only of sonobuoy drift but aircraft drift as well and is, in fact, the vector sum of these. An estimate which attempts to follow the changing velocities through the Schuler cycle tends to cause the position filters to diverge. This might be the case if larger values of gain are used, averaging only the last few measurements of drift. Therefore, many measurements of drift are averaged which provides an estimate for linear drift only. The lower limit of gain prevents new drift measurements from being ignored and also provides for flexibility in estimating the drift.

Unlike the six-state system, the interferometer measurement is the only measurement required by the position filter. The measurement equation is essentially the same as in the preceeding section where z is determined as shown in Equation (39). Since there is only one term in z, the H matrix is a row vector consisting of

$$H = \begin{bmatrix} \frac{\partial z}{\partial x_1} & \frac{\partial z}{\partial x_2} \end{bmatrix} \tag{61}$$

where the terms are given in Equations (41) and (42). It is necessary to determine the aircraft-to-sonobuoy range for these values from

$$\Delta X = \hat{X} - X_a^I \tag{62}$$

where $\hat{X}$ is the estimated sonobuoy position. X_a^I represents

the aircraft position and is considered deterministic. Undoubtedly, X_a^I is in error due to aircraft navigational drifts and therefore causes errors in the estimated geographical position of the sonobuoy. But, since the measurement of bearing is based on relative positions, the estimate is relatively correct.

An observability analysis on the 2-state position filter was performed again using

$$M_k = \sum_{k=1}^{N} \Phi_k^T H_k^T R^{-1} H_k \Phi_k \qquad (63)$$

Subtitution into this formula with N=2 yields

$$M = \begin{bmatrix} \left(\frac{\partial z}{\partial x}\right)^2_{1/i} & \left(\frac{\partial z}{\partial x}\right)_{1/i}\left(\frac{\partial z}{\partial x}\right)_{2/i} \\ \left(\frac{\partial z}{\partial x}\right)_{1/i}\left(\frac{\partial z}{\partial x}\right)_{2/i} & \left(\frac{\partial z}{\partial x}\right)^2_{2/i} \end{bmatrix} + \begin{bmatrix} \left(\frac{\partial z}{\partial x}\right)^2_{1/i+1} & \left(\frac{\partial z}{\partial x}\right)_{2/i+1}\left(\frac{\partial z}{\partial x}\right)_{1/i+1} \\ \left(\frac{\partial z}{\partial x}\right)_{1/i+1}\left(\frac{\partial z}{\partial x}\right)_{2/i+1} & \left(\frac{\partial z}{\partial x}\right)^2_{2/i+1} \end{bmatrix} \qquad (64)$$

Expanding,

$$\begin{aligned} M = {} & \left(\frac{\partial z}{\partial x}\right)^2_{1/i+1} \left(\frac{\partial z}{\partial x}\right)^2_{2/i} - \left(\frac{\partial z}{\partial x}\right)_{1/i+1}\left(\frac{\partial z}{\partial x}\right)_{2/i+1}\left(\frac{\partial z}{\partial x}\right)_{1,i}\left(\frac{\partial z}{\partial x}\right)_{2/i} \\ & + \left(\frac{\partial z}{\partial x}\right)^2_{1/i} \left(\frac{\partial z}{\partial x}\right)^2_{2/i+1} - \left(\frac{\partial z}{\partial x}\right)_{1/i}\left(\frac{\partial z}{\partial x}\right)_{2/i}\left(\frac{\partial z}{\partial x}\right)_{1/i+1}\left(\frac{\partial z}{\partial x}\right)_{2/i+1} \end{aligned} \qquad (65)$$

This matrix is positive definite for all except the case when

$$\left(\frac{\partial z}{\partial x_1}\right)_i = \left(\frac{\partial z}{\partial x_1}\right)_{i+1} \text{ and } \left(\frac{\partial z}{\partial x_2}\right)_i = \left(\frac{\partial z}{\partial x_2}\right)_{i+1}$$

This occurs whenever the relative bearing from the aircraft to the sonobuoy is not changing (i.e. when flying directly toward or away from the sonobuoy). This is intuitively correct since two or more bearings must cross in order to determine a position.

The two-state system was programmed and tested to determine the usefulness of this simplified filter. The majority of the program is the same as the one used for the six-state system. The significant differences occur in subroutine FILTER. The Carlson square root technique is used again but only one measurement instead of three is processed. The drift filter is programmed in this subroutine along with the position filter. A flow chart and program listing are presented in Appendix A.

IV. ANAYLSIS

A. THE SIMULATION

Actual data from the Orion was not easily obtainable for this research. Consequently, a computer program had to be written to generate the information required by these algorithms. A flow chart in Appendix B describes the program. An aircraft track was created by alternating lines and curves of various lengths and then a determination of noise free measurements was made as the track was flown. (Measurement noise was added later during the simulations.) Sonobuoys were allowed to drift at a constant velocity, and the aircraft's navigational drift was modeled with a constant velocity and Schuler cycle variations as

$$X_a^I = X_a + K_x \Delta t + A_x \sin \omega t \qquad (66)$$

where

X_a^I = aircraft inertial position

X_a = aircraft true position

ω = Schuler frequency $\sqrt{g/R}$

The constant drift rate, K_x, was found by NADC to have 0 mean and a standard deviation of 2.5 NM/HR. Likewise, the amplitude of the Schuler cycle, A_x, had 0 mean and a standard deviation of .5 NM. The effect of wind was also available in the program but was never included for analysis.

It was obvious that both algorithms were extremely sensitive to the aircraft flight path. Since there is no "typical" flight path for an Orion during its on-station period, several uniform patterns were selected which would provide meaningful information on each algorithm's performance. These base parameters were chosen:

airspeed	180 kts
altitude	3000 ft
range	15 NM aircraft-to-sonobuoy
frequency	20 sec between measurements on one sonobuoy

A circular pattern was flown around a sonobuoy at a range of about 15 NM. Initial sonobuoy placement was at (1,1)[1] with the aircraft flying clockwise starting at (15,0). This is shown in Figure 6 which depicts a portion of the aircraft's true track. Sonobuoy drift was west at 1 NM/HR; aircraft navigational drift was south at 1 NM/HR and included a Schuler cycle (amplitude .5 NM) when specified. (This drift would be indicated in Figures 12 thru 30 by the following notation: DR=(a/c 180-1 + Schuler, b 270-1) .) The sonobuoy remained near the center of the pattern allowing the range to remain relatively constant. This was desirable since this basic flight path was used to test the response of the algorithms to variations in altitude and range. Similarly, a square pattern was flown counter-clockwise around a

[1] (x_1, x_2) measured in nautical miles from an arbitrary origin near the aircraft's starting point.

sonobuoy at a range of 15 NM. In this case the aircraft began on-top the initial position of the sonobuoy at (-3,-3) and preceeded to fly the track shown in Figure 7. The sonobuoy was allowed to drift to the east at 2 NM/HR and the aircraft's navigational system to the north at 2.5 NM/HR. This pattern was used eventually to analyze performance at different measuring frequencies. Figure 8 shows the output of the aircraft's navigational system when it has a northerly drift of 2.5 NM/HR while the aircraft is flying the square pattern. Figure 9 shows this same pattern when a Schuler cycle with an amplitude of .5 NM is also present.

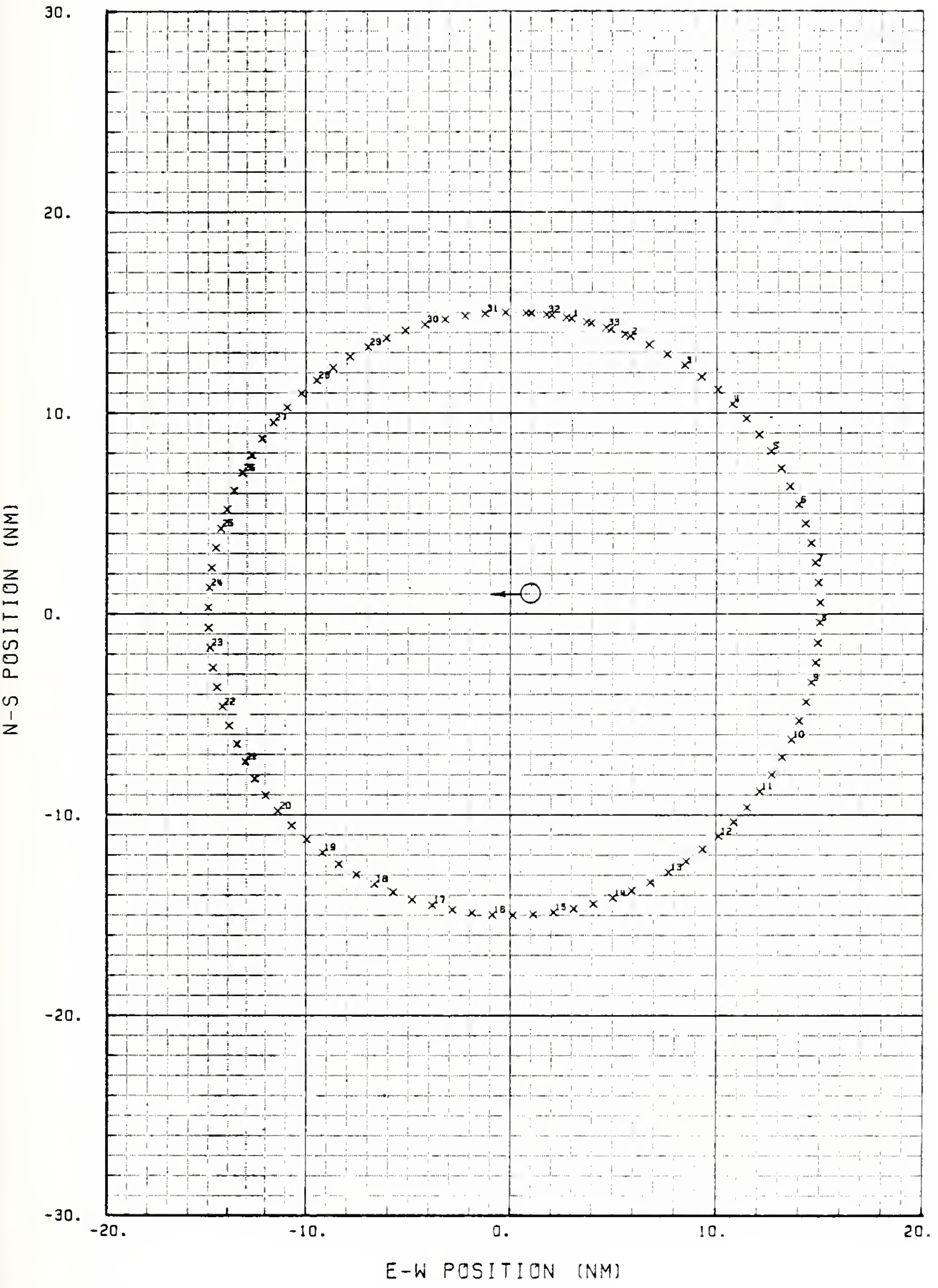


Figure 6. Aircraft track for the 15 NM circular pattern with initial sonobuoy location and direction of drift indicated.

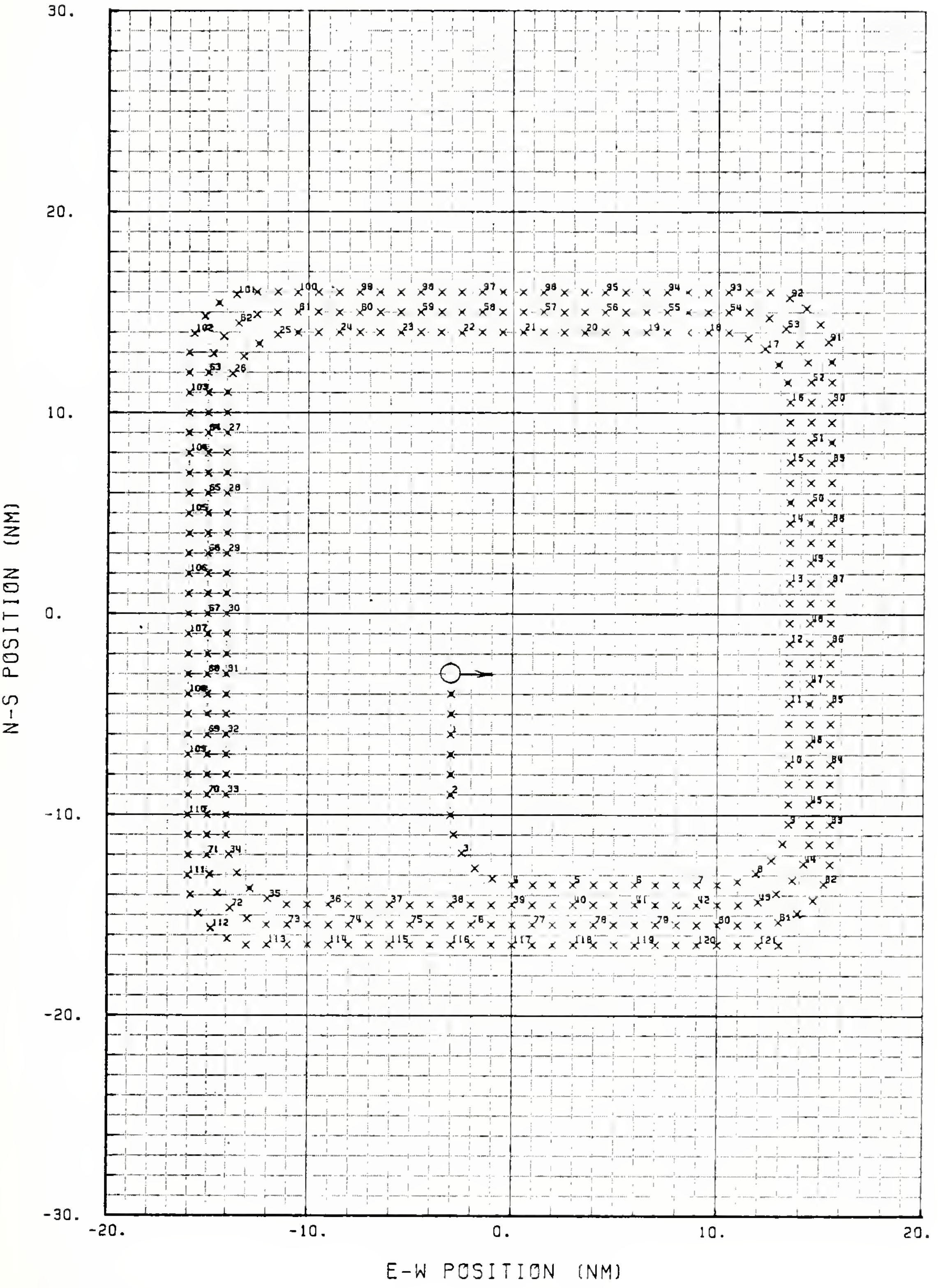


Figure 7. Aircraft track for the 15 NM square pattern with initial sonobuoy location and direction of drift indicated.

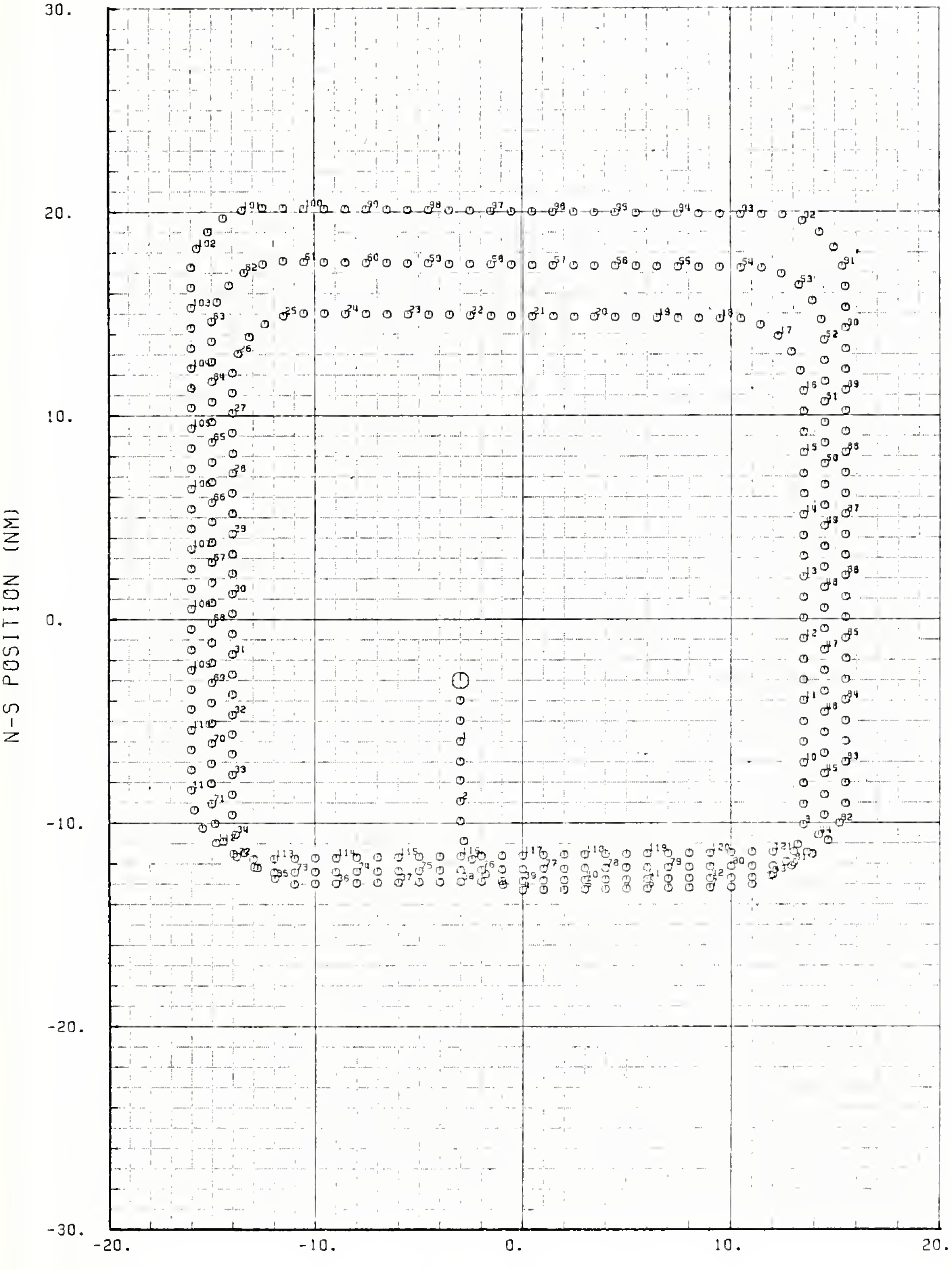


Figure 8. Aircraft's navigational output for the 15 NM square; navigational drift is 2.5 NM/Hr to the north.

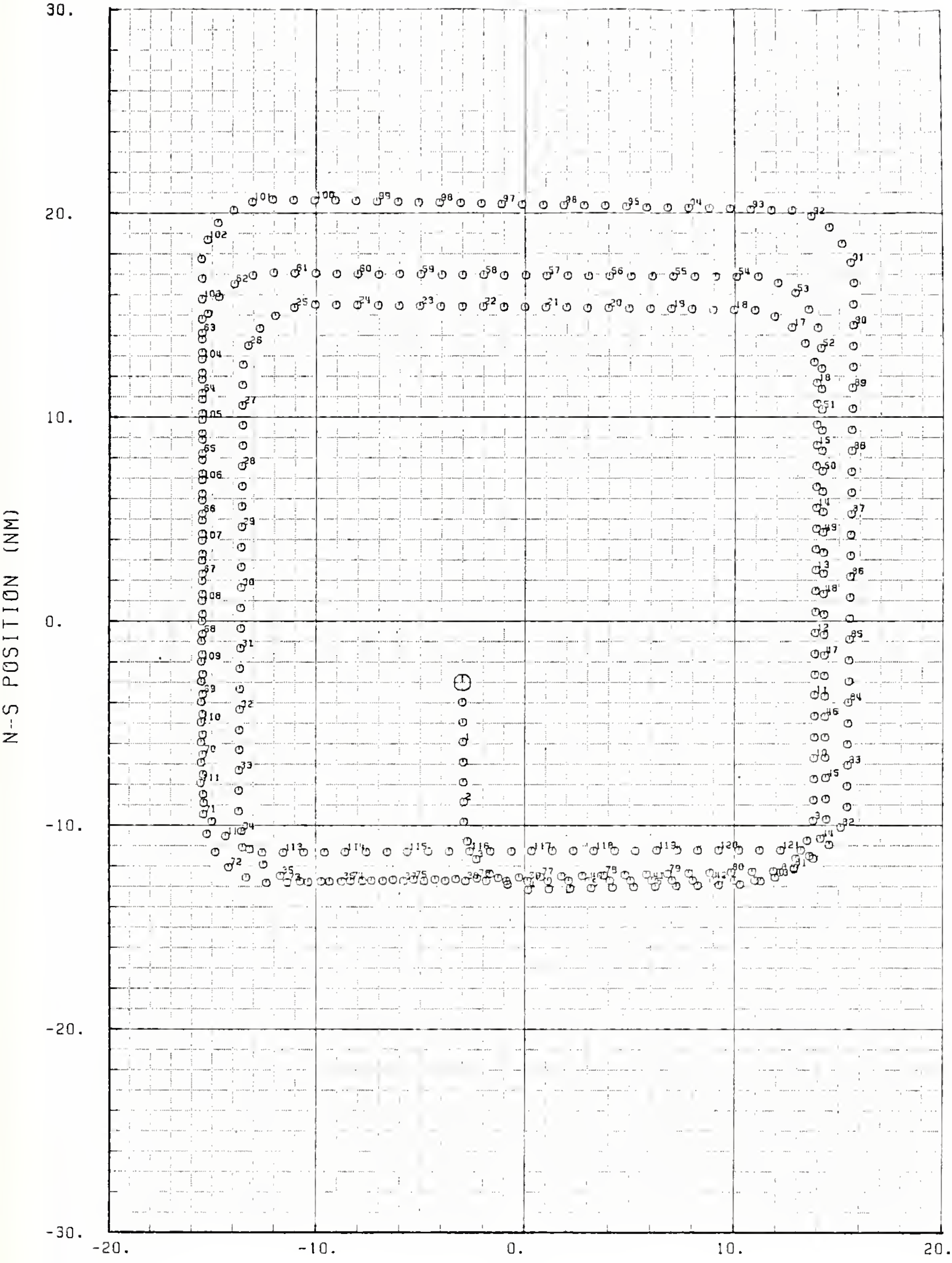


Figure 9. Aircraft's navigational output for the 15 NM square; navigational drift is 2.5 NM/Hr north with a Schuler cycle.

These patterns were used to generate the data which included sonobuoy number, time of measurement and measurement, aircraft latitude and longitude, altitude, heading, pitch angle, roll angle, N-S acceleration, E-W acceleration, and the antenna used for the measurement. Also determined, but used only for error analysis, were sonobuoy position and true aircraft position. This information was computed for the entire simulation period and stored in a file to be used when needed.

Initial estimates of the state and the square root of the covariance matrices were obtained as follows:

Six-state

$$X(\emptyset) = \begin{bmatrix} x_1(\emptyset)-x_{a_1}(\emptyset) \\ x_2(\emptyset)-x_{a_2}(\emptyset)+.5 \\ -V_a \cos H \\ -V_a \sin H \\ \emptyset \\ \emptyset \end{bmatrix} \qquad S_D(\emptyset) = \begin{bmatrix} 5. \\ 5. \\ 5. \\ 5. \\ .\emptyset\emptyset25 \\ .\emptyset\emptyset25 \end{bmatrix} \tag{67}$$

Two-state

$$X(\emptyset) = \begin{bmatrix} x_1(\emptyset) \\ x_2(\emptyset)+.5 \end{bmatrix} \qquad S_D(\emptyset) = \begin{bmatrix} \mathrm{Alt} \\ \mathrm{Alt} \end{bmatrix} \tag{68}$$

where $S_D(\emptyset)$ indicates the diagonal elements of $S(\emptyset)$.

For the circular pattern

$$x_1(Ø) = 1. \qquad\qquad x_{a_1}(Ø) = 15.$$
$$x_2(Ø) = 1. \qquad\qquad x_{a_2}(Ø) = Ø.$$

and for the square pattern

$$x_1(Ø) = -3. \qquad\qquad x_{a_1}(Ø) = -3.$$
$$x_2(Ø) = -3. \qquad\qquad x_{a_2}(Ø) = -3.$$

The initial values of the covariance were chosen so as to describe the errors in the state associated with dropping a sonobuoy from an aircraft. The two-state system operated better with a lower initial variance than did the six-state system.

Simulations were run for a nominal period of two hours. Each unit of data was sequentially read into the simulation program from the storage file. Measurement noise was added with the following normal distributions:

$\cos\theta$	N(Ø,.ØØØ7)	in radians
doppler velocity	N(Ø,1.Ø)	in NM/HR
doppler drift angle	N(Ø,.Ø1)	in degrees

These values agree with the ones chosen for R as described previously. A random number generator using an initial seed was used to create the noise from the proper distributions. After the estimate was made, errors were measured and manipulated in subroutine RESULT. This cycle was repeated

until the simulation was complete. (Refer again to Appendix A for a flow diagram of the simulation process.)

Information was gathered and is presented primarily in two ways. First, for both the square and circular patterns the results of the first run of the simulation are plotted. (Figure 13 is an example.) The top portion of the results shows the estimated positions of the sonobuoy on the aircraft's navigational plot. It should be noted that these are not true positions since inertial errors may be causing this plot to drift. The bottom portion shows the North-South and East-West errors in the estimated positions. These errors are the differences between the true and the estimated relative positions of the sonobuoy. They are depicted in Figure 10 and were computed from

$$\Delta\tilde{X}_k = \Delta\hat{X}_k - \Delta X_k \qquad (69)$$

$$\Delta\tilde{X}_k = (\hat{X} - X_a^I)_k - (X - X_a)_k \qquad (70)$$

where $\Delta\tilde{X}_k$ is the relative error at time k. The true aircraft

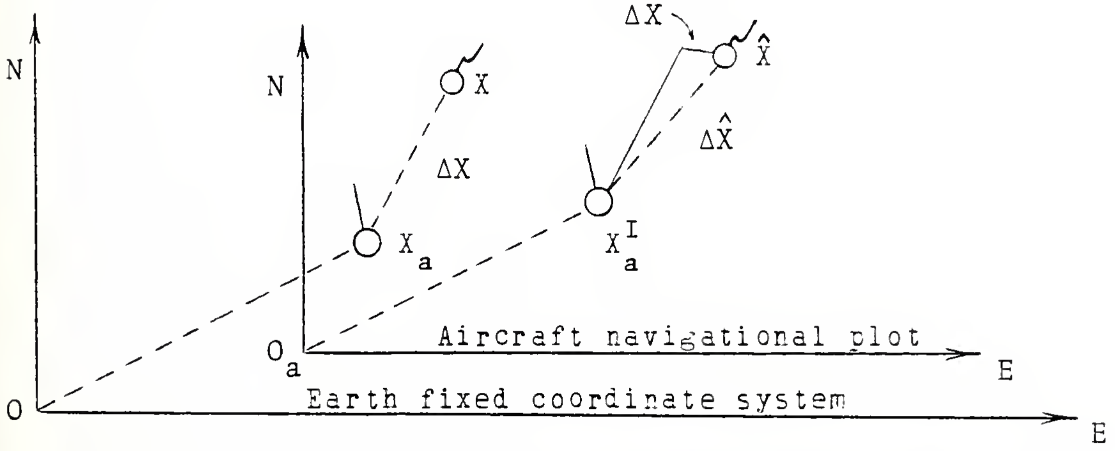


Figure 10. The effect of the aircraft navigational plot

position is denoted by X_a and the true geographical position of the sonobuoy is denoted by X. These values are known from the simulated data. $\hat{X}$ is the filter's estimate of position and X_a^I is a deterministic input from the aircraft's navigational system. The errors indicated in these plots are different from those perceived in the upper plot whenever the aircraft's navigational picture is drifting. The positive and negative values of the square root of the covariance are also shown on these plots as solid lines. Specifically,

$$\text{N-S error} = \pm\sqrt{P_{11}}$$

$$\text{E-W error} = \pm\sqrt{P_{22}}$$

(Only position errors are analyzed in this report.)

The second way in which the information was gathered consisted of a shortened Monte-Carlo simulation. For each scenerio under study, 20 two-hour simulations were run, each with new values of measurement noise provided by the random number generator from their respective distributions. The objective was to compare the RMS errors predicted by the covariance matrix in the filters to the actual RMS errors observed in the simulations. Three RMS statistics were collected and plotted versus time as follows:

1. The square root of the covariance computed by the Kalman filter is represented by a solid line on the plots. It was computed from

$$\sigma_{f_k} = \sqrt{\frac{1}{n}\sum P_{11} + \frac{1}{n}\sum P_{22}}\text{, } n=20\text{, } k = 1,2,3... \qquad (71)$$

This standard deviation is the filter´s estimate of its accuracy.

2. The mean and standard deviation of the error in the estimated sonobuoy position relative to the aircraft was determined by

$$\mu_k = \frac{1}{n}\sum_{i=1}^{n} \Delta\tilde{X}_{k,i}, \qquad n = 20,\ k = 1,2,3... \tag{72}$$

$$\sigma^2_{x_k} = \frac{1}{n}\sum_{i=1}^{n} \Delta\tilde{X}^2_{k,i} - \mu^2_k\ , \qquad n = 20,\ k = 1,2,3... \tag{73}$$

   This standard deviation is represented by an ´X´ on the plot and is a measure of the variability in the filter´s estimated position.

3. The RMS value of the actual relative error was also also determined. It was computed as

$$\sigma^2_{a_k} = \frac{1}{n}\sum_{i=1}^{n} \Delta\tilde{X}^2_{ki}\ , \qquad n = 20,\ k = 1,2,3... \tag{74}$$

   and is represented by an ´O´ on the plot. This value is a measure of the error which occurred between the actual relative position of the sonobuoy and the estimated one. (See Figure 11.)

These values are plotted over the two hour simulation period (k = 1,2,3...) and provide a measure of the accuracy obtained by each of the algorithms.

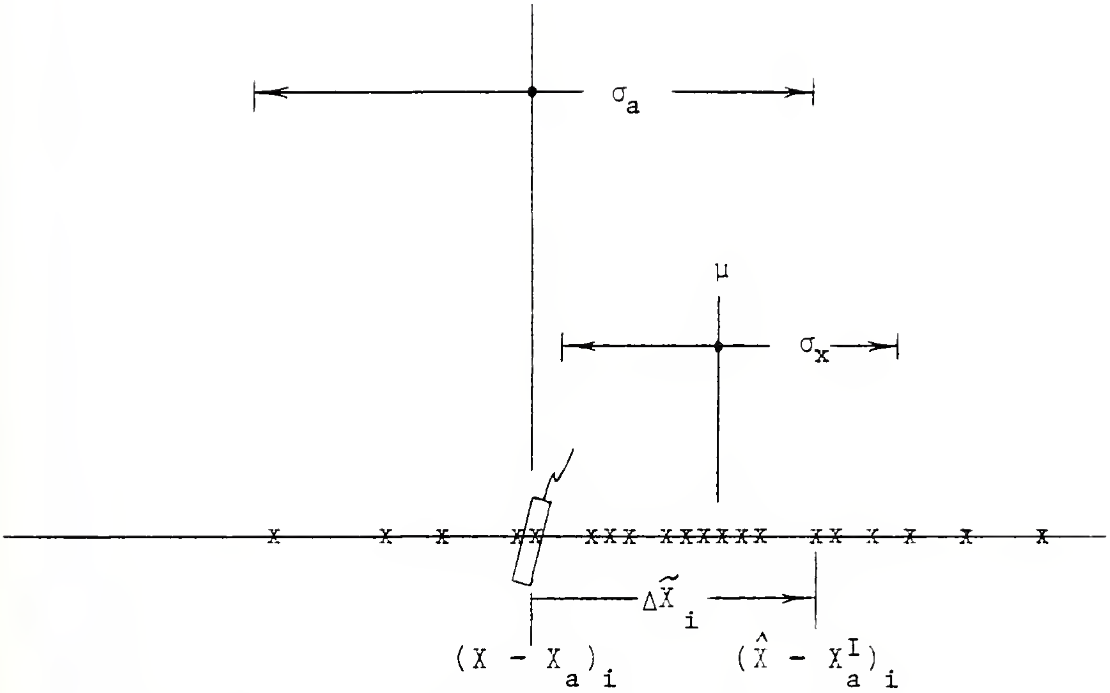


Figure 11. RMS statistics at time = k.

B. THE RESULTS

Orincon Corporation performed much of their analysis using the small circular pattern shown in Figure 12. The aircraft sped around this track at 388 knots taking a measurement every 20 seconds. Altitude was constant at 3000 feet. The initial position of the sonobuoy was at (2,2) and it drifted on a heading of 045 degrees at 5.5 NM/HR while the aircraft's navigational system drifted in the opposite direction at 5.2 NM/HR. A Schuler cycle was also superimposed on this drift.

The initial estimates of sonobuoy position and variances used by the algorithms were different for this particular pattern in order to coincide with the Orincon simulation as much as possible. They were

Two-state

$$X(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad S_D(0) = \begin{bmatrix} 2. \\ 2. \end{bmatrix} \qquad (74)$$

Six-state

$$X(0) = \begin{bmatrix} -5.4 \\ 0 \\ -V_a \cos H \\ -V_a \sin H \\ 0 \\ 0 \end{bmatrix} \qquad S_D(0) = \begin{bmatrix} 2. \\ 2. \\ 5. \\ 5. \\ .0025 \\ .0025 \end{bmatrix} \qquad (75)$$

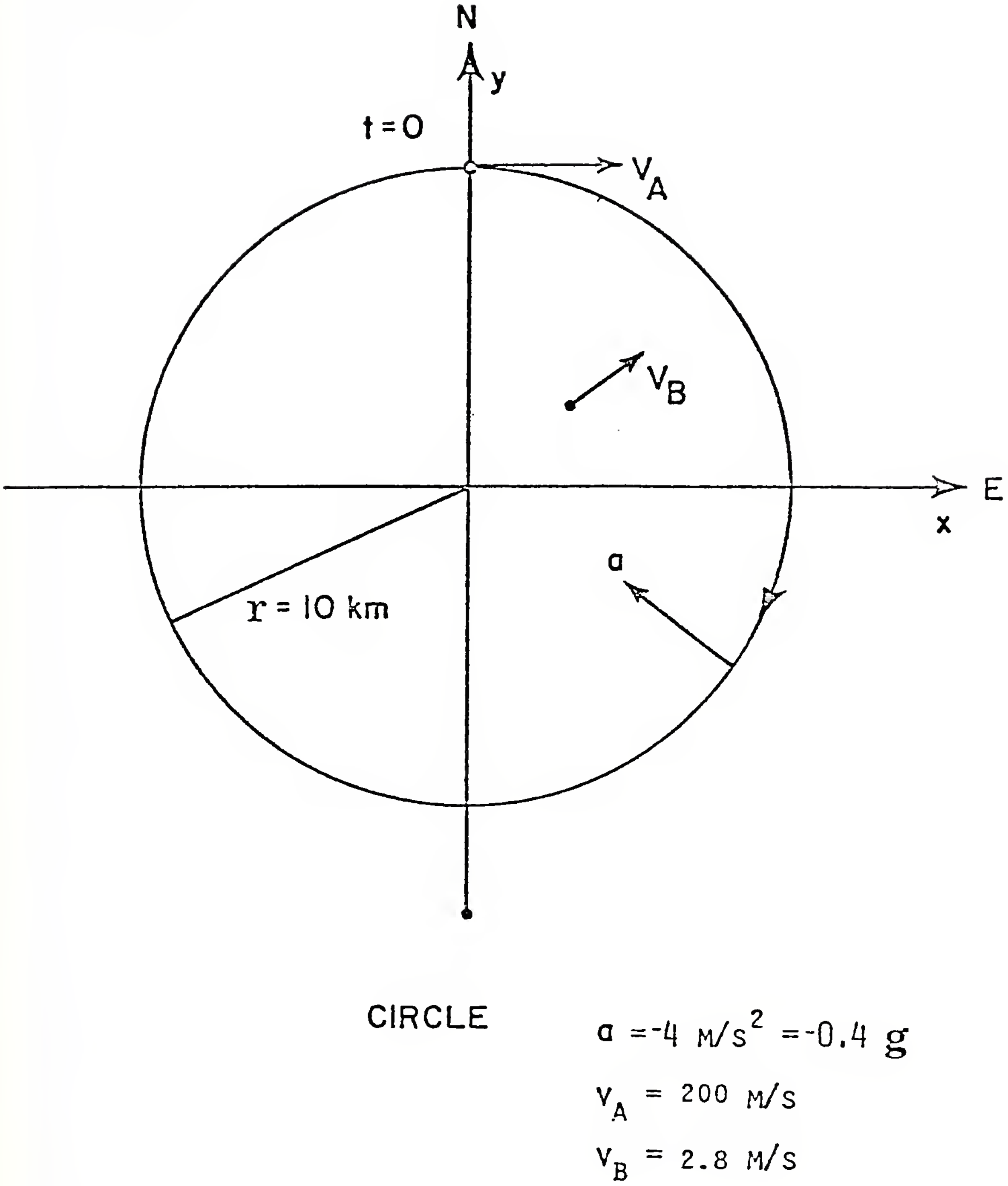

N
y
t = 0
V_A
V_B
E
x
a
r = 10 km
CIRCLE
$a = -4\ \mathrm{M/S}^2 = -0.4\ g$
$V_A = 200\ \mathrm{M/S}$
$V_B = 2.8\ \mathrm{M/S}$

Figure 12. Orincon's simulation flight path.

Initially, both the six-state and the two-state algorithms were tested using this flight path and comparisons were made with the results of Orincon's 13-state system. It was discovered that the six-state system's results did not change as a consequence of whether or not a Schuler cycle existed in the navigational system. Therefore, the six-state system was run only with the Schuler cycle active. The two-state system was run with and without a Schuler cycle.

The results of the first run of the simulation are shown in Figures 13, 14, and 15. The sawtooth shape of the covariance (solid lines) is the result of the circular pattern and is caused in two ways. Primarily, since the sonobuoy remains closer to one side of the circle, there is a minimum value of the variance each time the aircraft passes on this side. A second reason for a change in variance is the aircraft's location in the pattern. However, these modulations are not as apparent since the range variation dominates. RMS errors were obtained by repeating these runs 20 times and are shown in Figures 16 thru 18. The fluctuations generated by the flight path are apparent, more so in the six-state system than in the two-state system. Also, steady state values of the error gradually increase. This is true because the average range to the sonobuoy increases as the buoy moves farther away

from the center of the pattern. (It will be demonstrated later that the errors are range dependent.) The covariance of the six-state system increases from 400 yards at 25 minutes to 800 yards at the end of the simulation. The two-state system increases slowly from 250 yards until it is hampered by the Schuler cycle. It is worthy of note that without the navigational errors stemming from the Schuler cycle the RMS errors of the two-state system are considerably lower than those of the six-state system.

Table 1 compares the RMS errors of these two systems to the results obtained by Orincon for their 13-state system. (The values for the two-state and six-state systems are taken from the covariance at a point 35 minutes into the simulation.) The first line considers measurements of bearing, doppler velocity, and drift angle. Navigational errors are present including a Schuler cycle. As expected there is an increase in relative position error because of the reduction from 13 to 6 states. The next line considers the effect of the Global Positioning System (GPS), with very accurate aircraft positioning information, on the relative

	13-state	6-state	2-state
Schuler SRS+dv+DA	394	450	
No Schuler SRS+DV+DA+GPS		450	200

Table 1. Comparison of RMS results in yards.

position errors of the sonobuoy. In the 13-state system Orincon observed that the errors in the states pertaining to the aircraft inertials improved markedly; relative position errors most likely remained the same. This is exactly what was found in the analysis of the six-state system: that the relative position errors were unaffected by navigational errors. However, the two-state system is very much affected by navigational errors. Relative position errors were significantly less for the two-state system when accurate aircraft positions were known.

To obtain a more realistic analysis of their performance, each algorithm was tested using the circular and square flight patterns of Figures 6 and 7. The results of the first simulation run for the circular path at a range of 15 NM are shown in Figures 19 thru 21. In this case, the wavy nature of the covariance is due to the location of the aircraft in the pattern since the range to the sonobuoy remains relatively constant. For instance, as the aircraft passes directly north or south of the sonobuoy the ability to correct E-W errors is greatest. Therefore, the E-W variance reaches a minimum value at this time. The N-S variance is affected in a similar manner. The two-state filter is not as sensitive to this as the six-state filter. Also, the effect of the Schuler cycle on the two-state filter can be seen in Figure 20.

The RMS errors are shown in Figures 22 thru 24. The steady state RMS value of error predicted by the filter is

425 yards for the two-state and 750 yards for the six-state system. The actual RMS error in the relative position of the sonobuoy is 250 to 1200 yards for the two-state and 400 yards for the six-state system. Without Schuler cycle error the two-state system drops to a steady 250 yards, better than the six-state system. It is interesting to note that the measured errors are significantly less than those predicted by the filter. Both measured values of RMS error, σ_x and σ_a, are in close agreement; that is, the deviation of the filter's estimated position about its mean is generally the same as the deviation of the estimated position about the actual relative position of the sonobuoy. The closer these values are to one another, the more confidence can be placed in the analysis. The only exception to this is Figure 23 which shows the reaction of the two-state filter to the Schuler cycle. One complete cycle with a period of 84 minutes is obvious. The Kalman filter does not recognize this cycle since the modeling equations do not account for it. The dip in the center of the two peaks is once again the result of flight path geometry. The aircraft is in such a position relative to the sonobuoy that the measurements provide enough information to correct for Schuler cycle errors. However, it is not in this position long enough to influence the errors anymore than it does.

The same analysis was performed using the square pattern of Figure 7. This scenerio has more drift than the other and in a different direction providing the algorithms

with another motion to track. The outcome was generally the same as can be seen in Figures 25 thru 30. There is no measurable difference in the results from the two different flight paths. Flying a straight path does not adversely affect the results as was concluded by the Oricon Corporation for their 13-state system. The wavy nature of the covariance is again a result of the flight path. It is not as smooth as before because a square pattern was flown instead of a circular one. Another look at errors caused in the two-state system by the Schuler cycle can be seen in Figure 30. The algorithm does make some corrections to these errors but they are not as effective as before. In this case the time the aircraft was in a position to make the corrections did not coincide with the time the peak errors occurred. There appears to be no way to predict the optimum time and place for the aircaft to be without knowing when and how the Schuler cycle is occurring.

The distance the aircraft is from the sonobuoy is directly related to the accuracy the algorithms can achieve. Simulations were performed at ranges of 5, 15, 30, and 45 nautical miles using a circular flight path with the sonobuoy in the center. (The RMS plots can be found in Appendix C.) The steady state errors were observed and are plotted in Figure 31. Solid lines represent the RMS values of the covariance and dashed lines represent the actual errors. In all cases the errors increase with distance which is intuitively satisfying. Mathematically, the

covariance increases because the H matrix becomes smaller with an increase in range. Consequently, the updated value of the covariance, P(+), is larger from Equation (19). For the six-state system there is an increase in actual error from 240 yards at 5 NM to 750 yards at 45 NM. And for the two-state system, actual errors increase from 130 yards at 5 NM to 500 yards at 45 NM. (Note that for the two-state system, Schuler cycle errors, indicated by circles, peak approximately 1000 yards above these plotted values; the lower values, indicated by x´s, were used since they were more reliable for comparison between ranges.) There is a slight decrease in the slopes of all the error curves as distance increases. Hovever, this decrease is small and the curves might well have been interpeted as linear within the limits of the analysis.

The frequency at which measurements are made also affects the accuracy of the estimate. Each algorithm was tested with measurement intervals of 4, 10, 20, and 30 seconds on one sonobuoy. The outcomes of the steady state RMS errors are plotted in Figure 32. (Once again the two-state system´s errors do not show the effect of the Schuler cycle.) The actual errors in the six-state system increase steadily as the measurement interval increases. However, the covariance decreases rapidly at first to a minimum value somewhere around 10 seconds and then increases with increasing interval. An abbreviated run with a two second measurement interval confirmed these results.

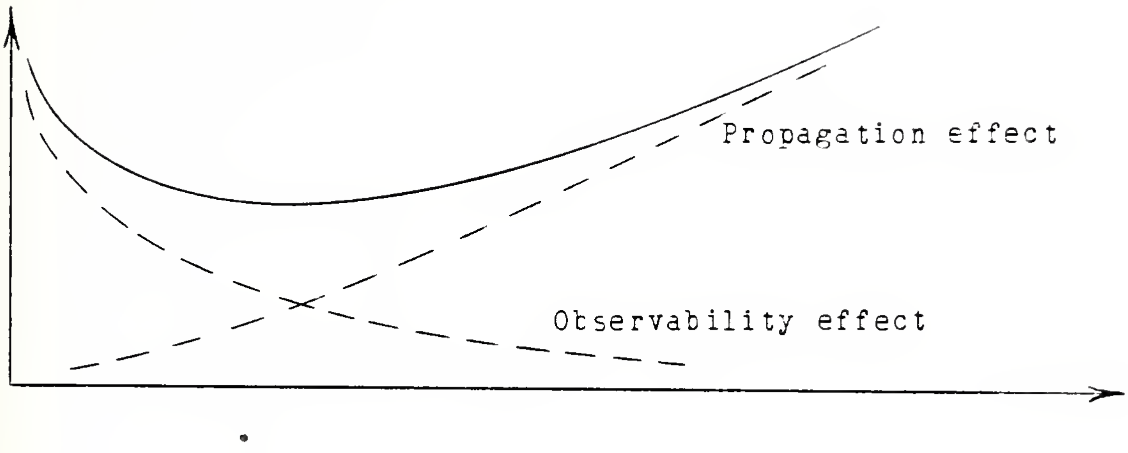


Figure 33. Factors which influence the covariance.

Figure 33 helps to explain this outcome. There are two conditions which affect the covariance. The most obvious is the increase in covariance due to an increase in the propagation interval. This occurs because the plant matrix, which is a function of time, affects the covariance in Equation (16). The second condition is a consequence of the model's observability. If the relative bearing between the aircraft and the sonobuoy does not change then the state is not observable. In other words, two successive bearings must intersect to determine an estimated position. By allowing the interval between measurements to become too small, the aircraft is unable to make a significant change relative to the sonobuoy. The covariance increases as the conditions approach those which make the problem unobservable. There appears to be an optimum frequency with which to make measurements on one buoy, namely 10 seconds for this range of 15 NM and speed of 180 knots.

The two-state system exhibited similar characteristics in Figure 32. In this case the covariance never does increase in the range of intervals studied. The actual errors remain constant. Since the plant matrix for this system is equal to the identity matrix and the Q matrix is prescribed to be constant, the propagation interval, Δt, does not effect the covariance. However, the effects of the observability conditions do cause the covariance to increase when Δt becomes too small. It is believed that as Δt becomes smaller the actual errors would begin to increase also.

The statistical plots for the range tests found in Appendix B show that the Schuler cycle causes larger peaks in the actual error as the range from the sonobuoy increases. However, the peaks are decreasing with time. Further simulation revealed that the peak errors decrease to a steady value about 1000 yards greater than non-Schuler cycle errors. This coincides with the amplitude of the Schuler cycle as it was programmed for this particular flight path. Therefore, in the steady state a Schuler cycle may cause additional estimation errors approximately equal to its amplitude.

Altitude was also tested for any effect it might have on these algorithms. Using the circular pattern at a range of 15 NM, altitudes of 300, 3000, 10,000, and 20,000 feet were tested. Neither the six-state nor the two-state system showed any effect for these changes in altitude. It is

believed that at very small ranges (less than two nautical miles) altitude might hamper good position estimation. However, for the majority of the time altitude is of no concern.

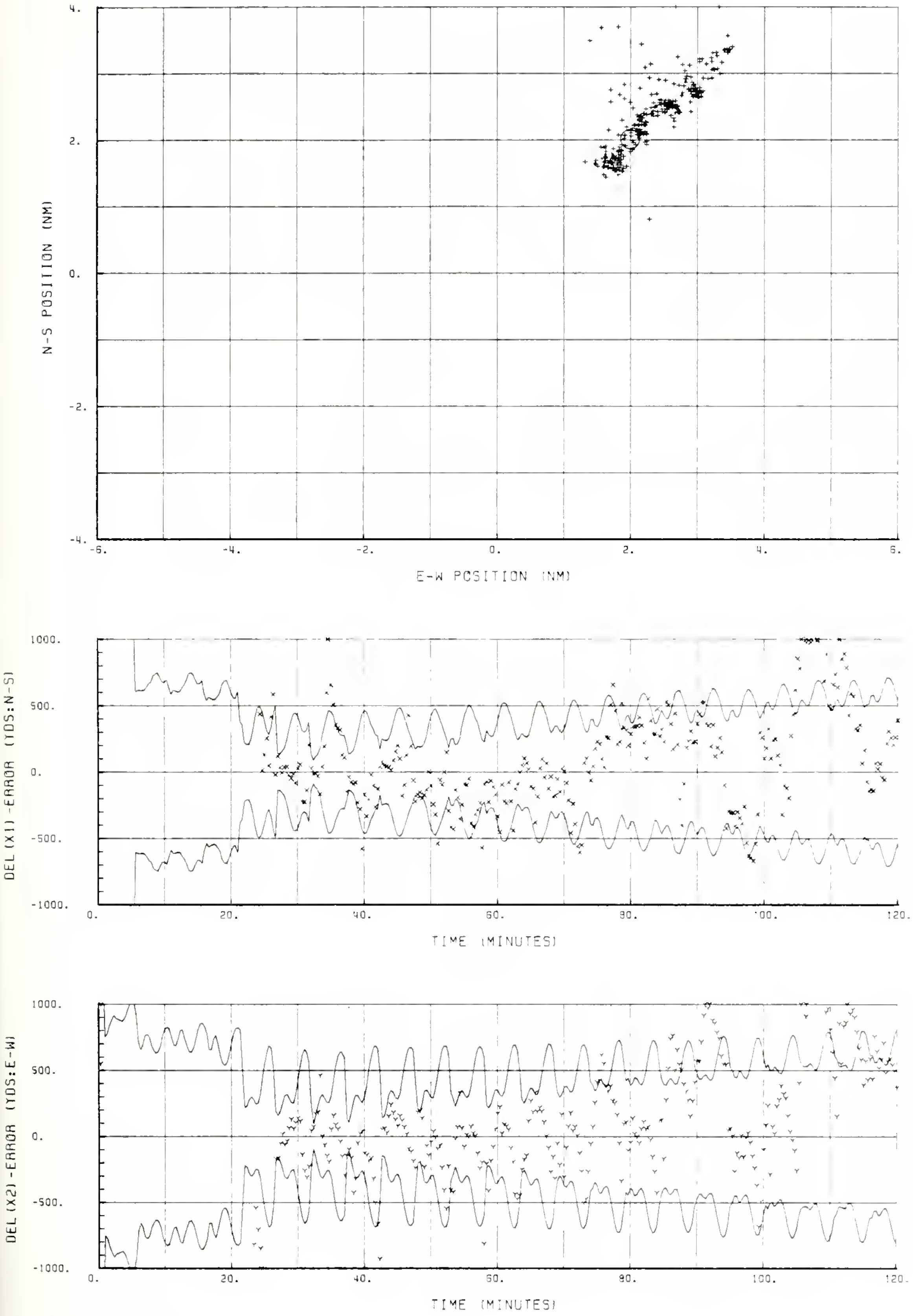


Figure 13. Est. position and relative errors for the six-state system using Orincon's pattern. DR=(a/c 225-5.2 + Schuler, b 045-5.5), Δt=20s, Alt=3000'.

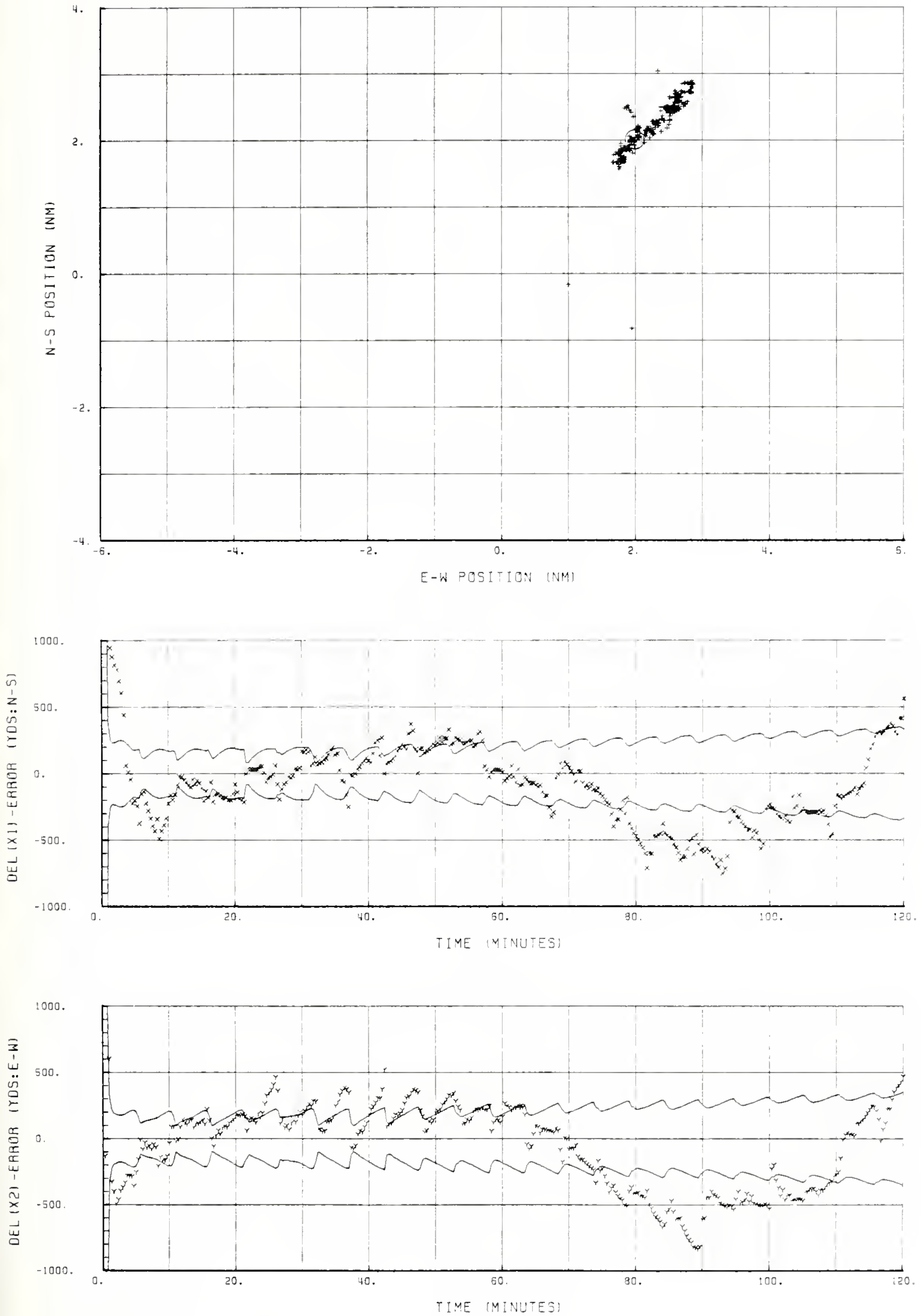


Figure 14. Est. position and relative errors for the two-state system using Orincon's pattern. DR=(a/c 225-5.2 + Schuler, b 045-5.5), Δt=20s, Alt=3000'.

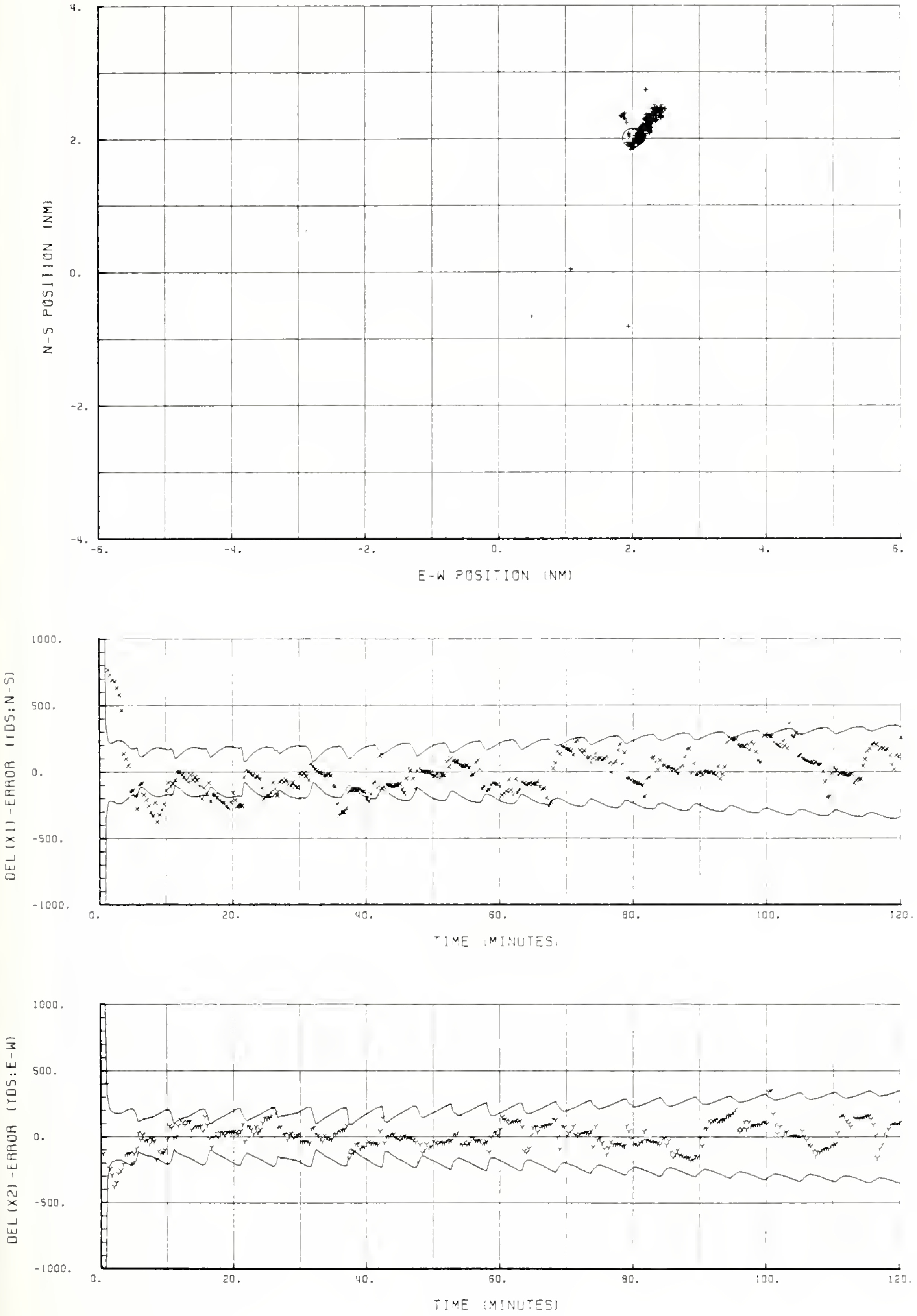


Figure 15. Est. position and relative errors for the two-state system using Orincon's pattern. DR=(a/c 225-5.2 + No Sch., b 045-5.5), Δt=20s, Alt=3000'.

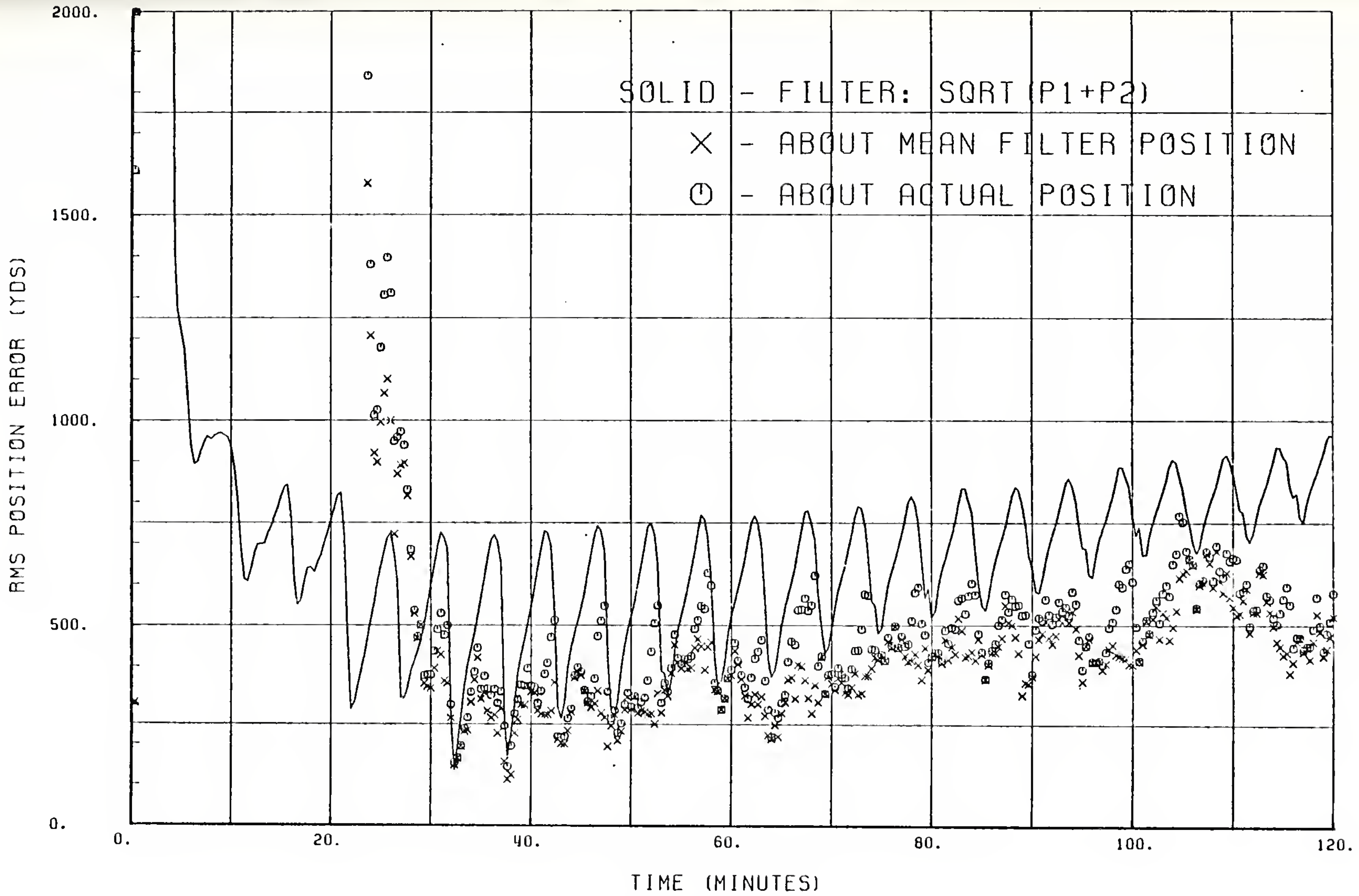


Figure 16. RMS position errors for the six-state system while flying the circular pattern developed by Orincon Corp. DR=(a/c 225-5.2 + Schuler, b 045-5.5), Δt=20s, Alt=3000'.

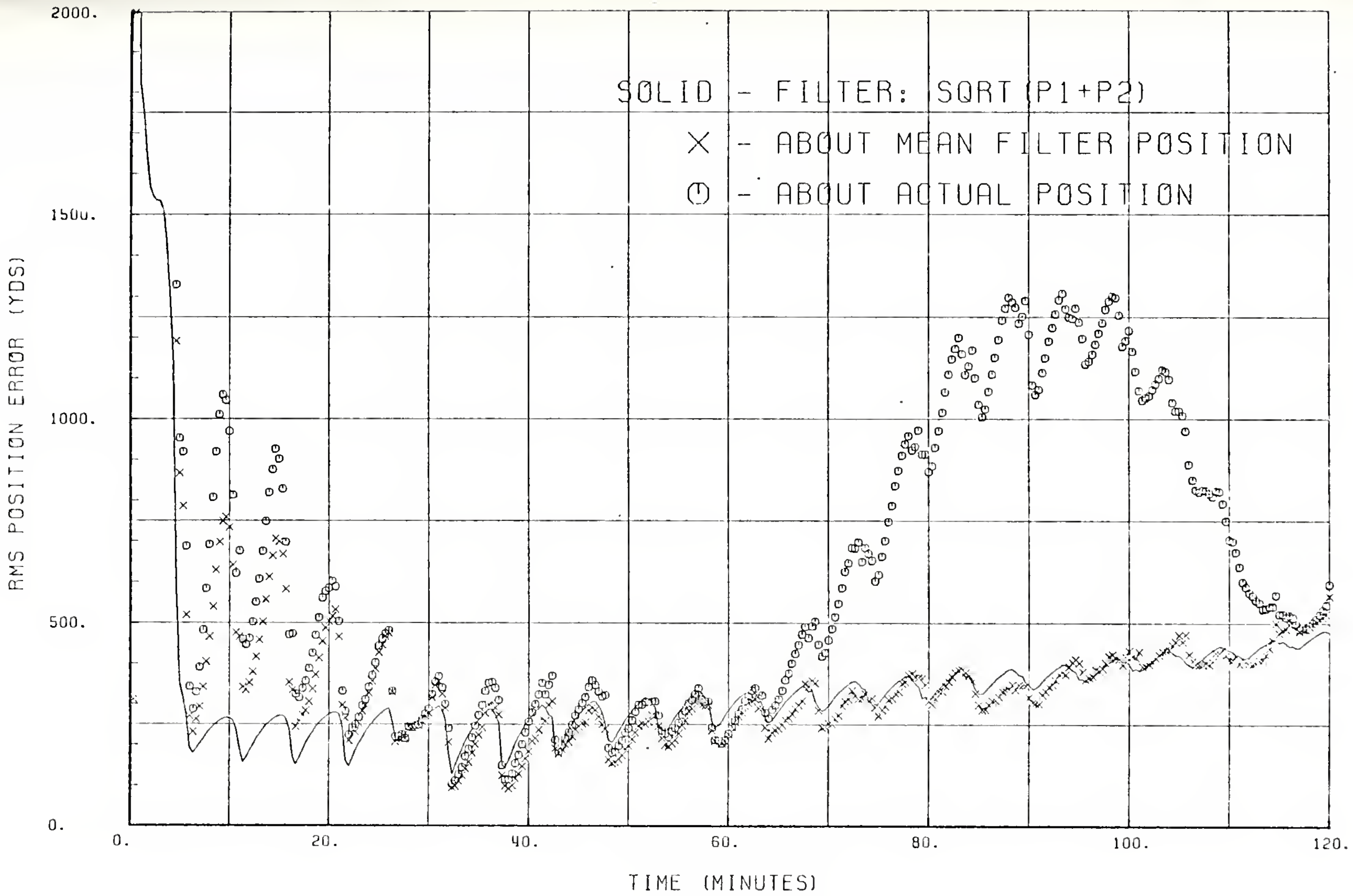


Figure 17. RMS position errors for the two-state system while flying the circular pattern developed by Orincon Corp. DR=(a/c 225-5.2 + Schuler, b 045-5.5), Δt=20s, Alt=3000'.

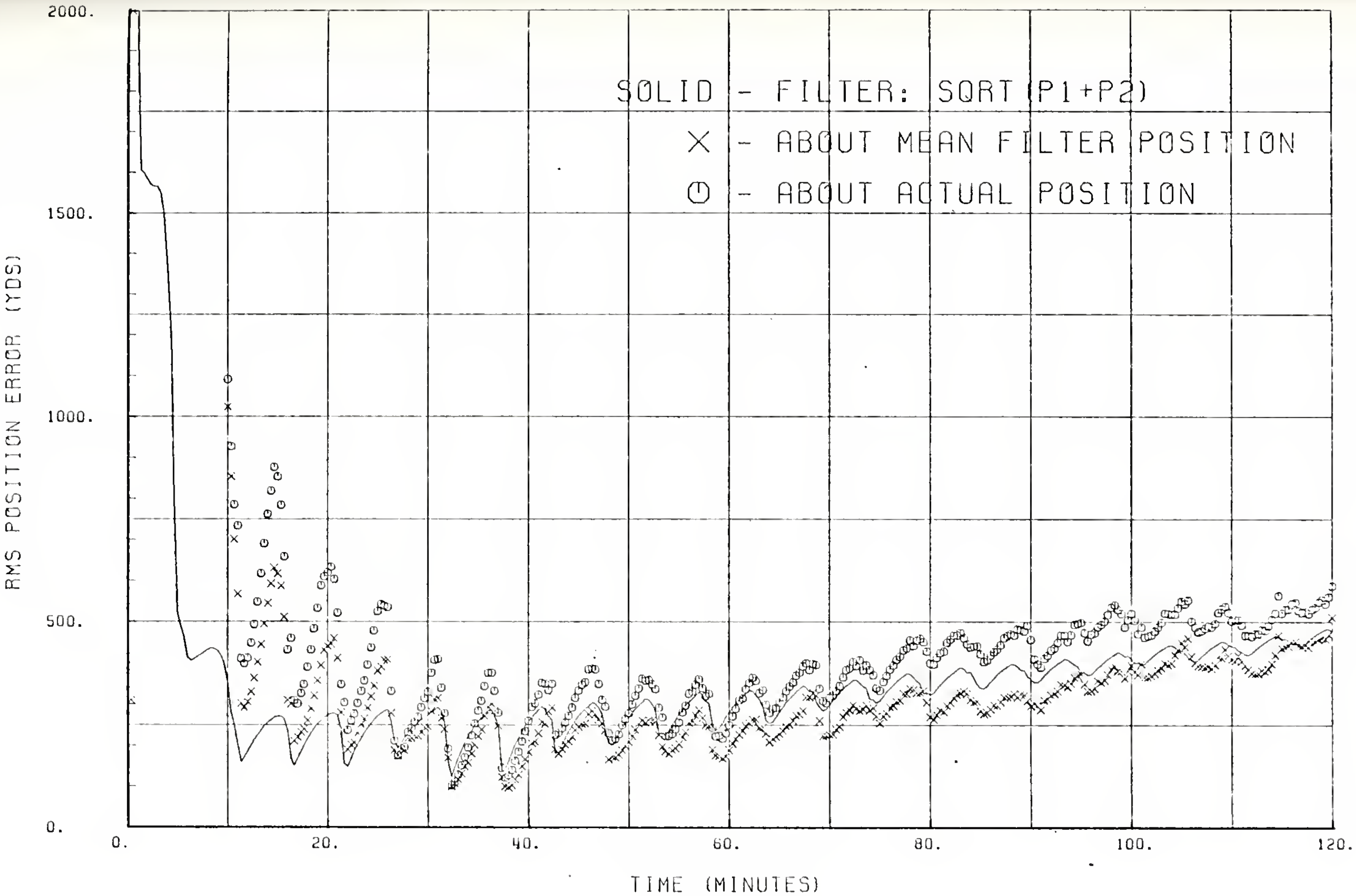


Figure 18. RMS position errors for thr two-state system while flying the circular pattern developed by Orincon Corp. DR=(a/c 225-5.2 No Sch., b 045-5.5), Δt=20s, Alt=3000'.

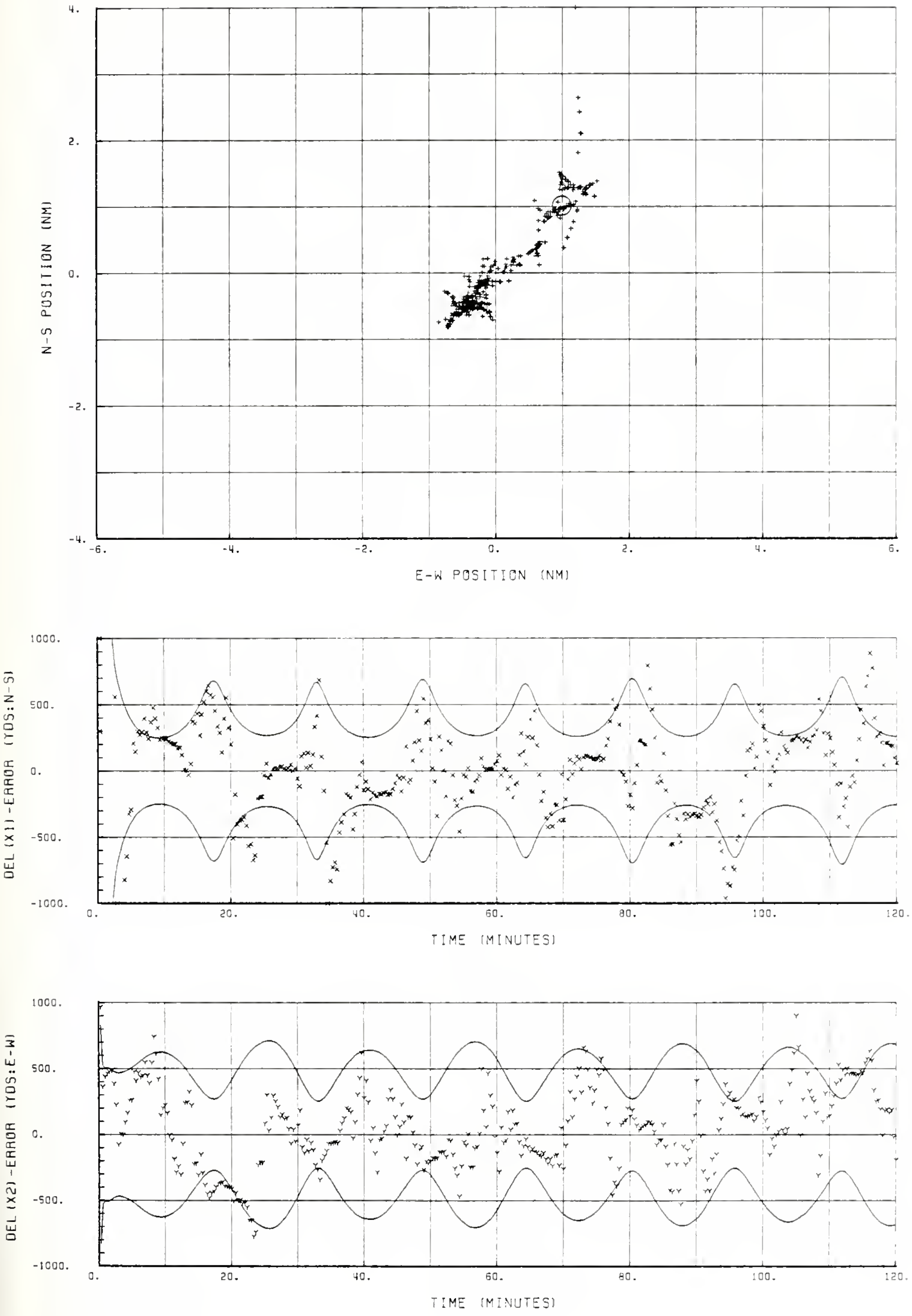


Figure 19. Est. position and relative errors for the six-state system using circular pattern at 15NM. DR=(a/c 180-1 + Schuler, b 270-1), Δt=20s, Alt=3000'

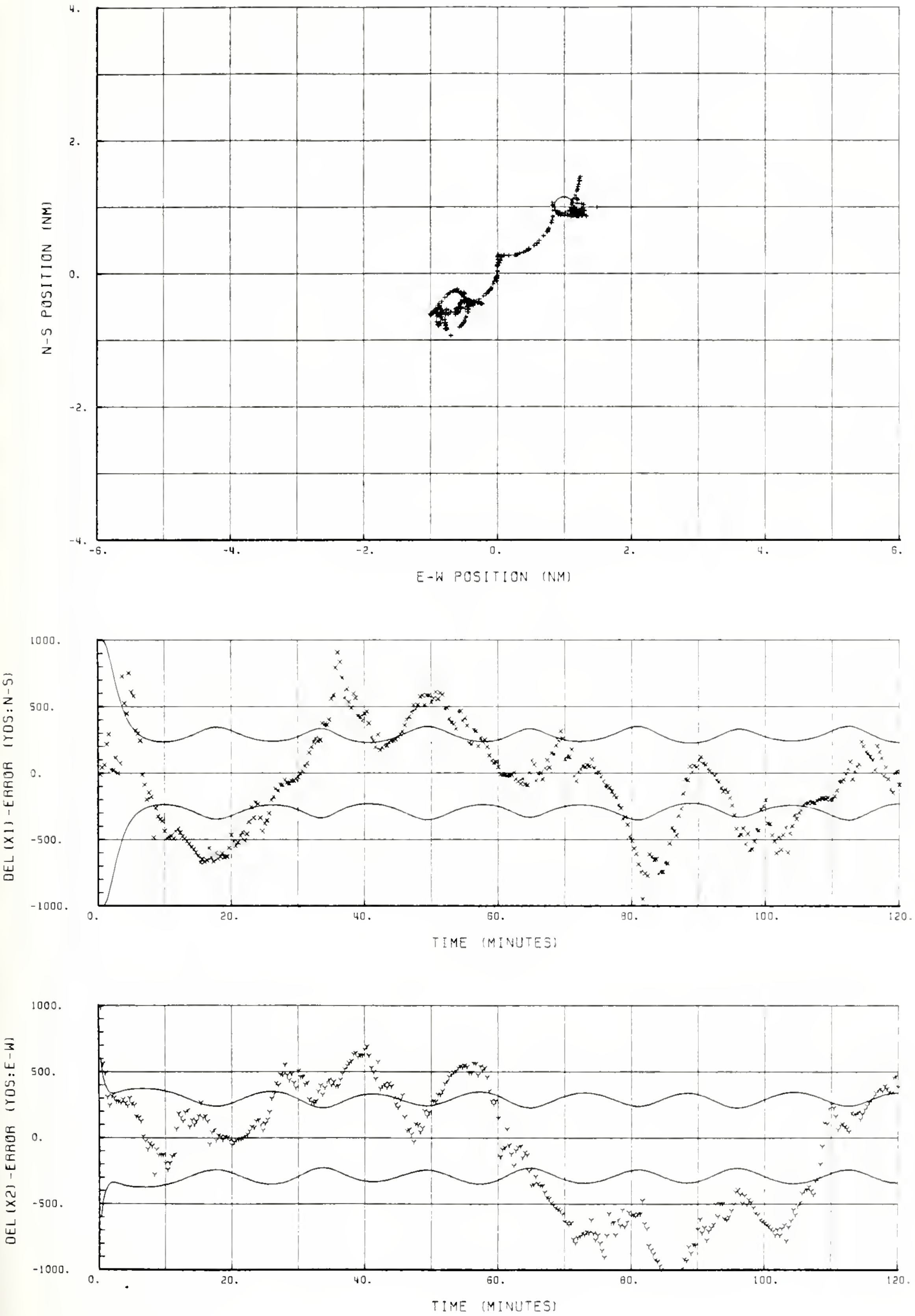


Figure 20. Est. position and relative errors for the two-state system using circular pattern at 15NM. DR=(a/c 180-1 + Schuler, b 270-1), Δt=20s, Alt=3000'

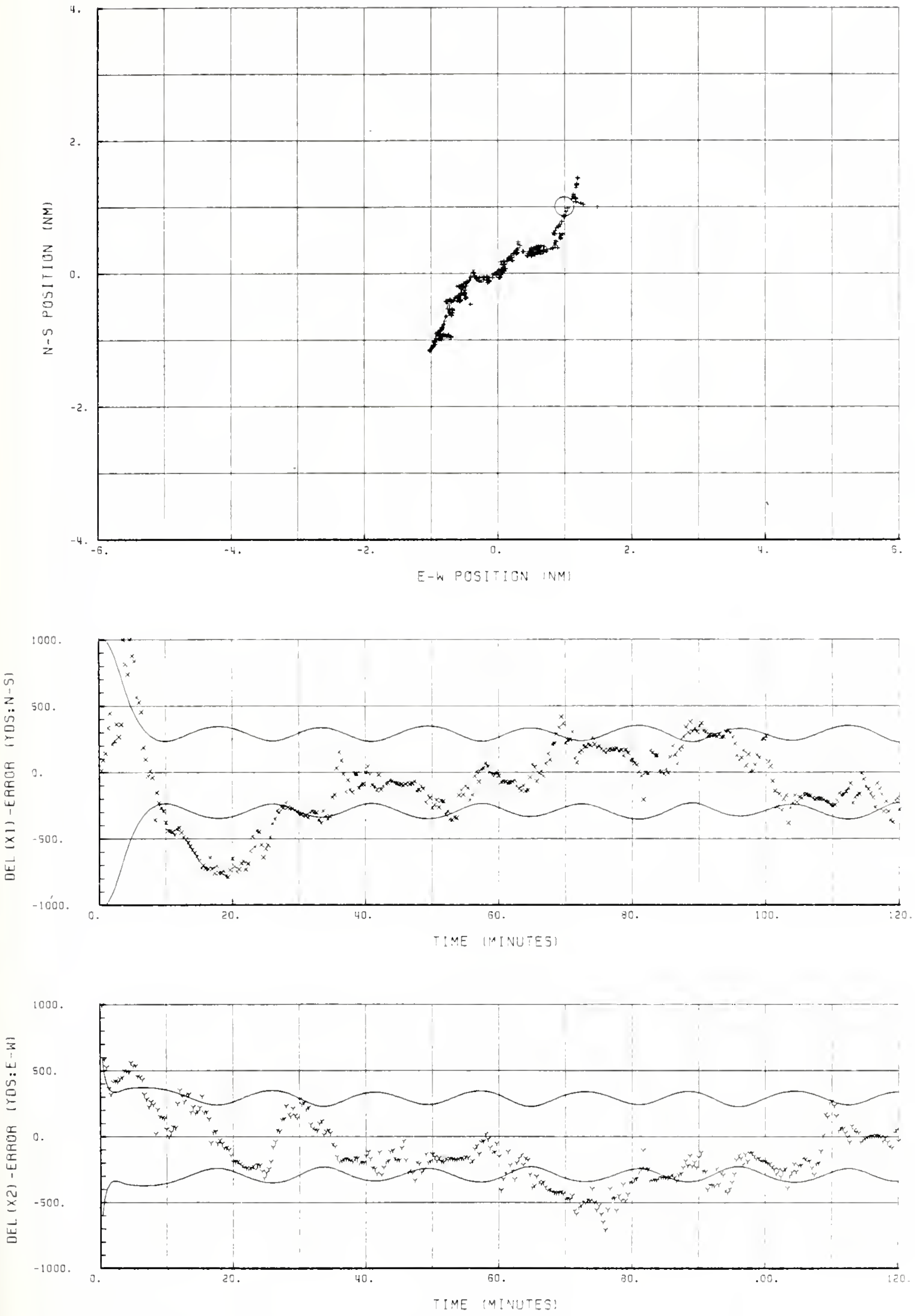


Figure 21. Est. position and relative errors for the two-state system using circular pattern at 15NM. DR=(a/c 180-1 + No Sch., b 270-1), Δt=20s, Alt=3000'

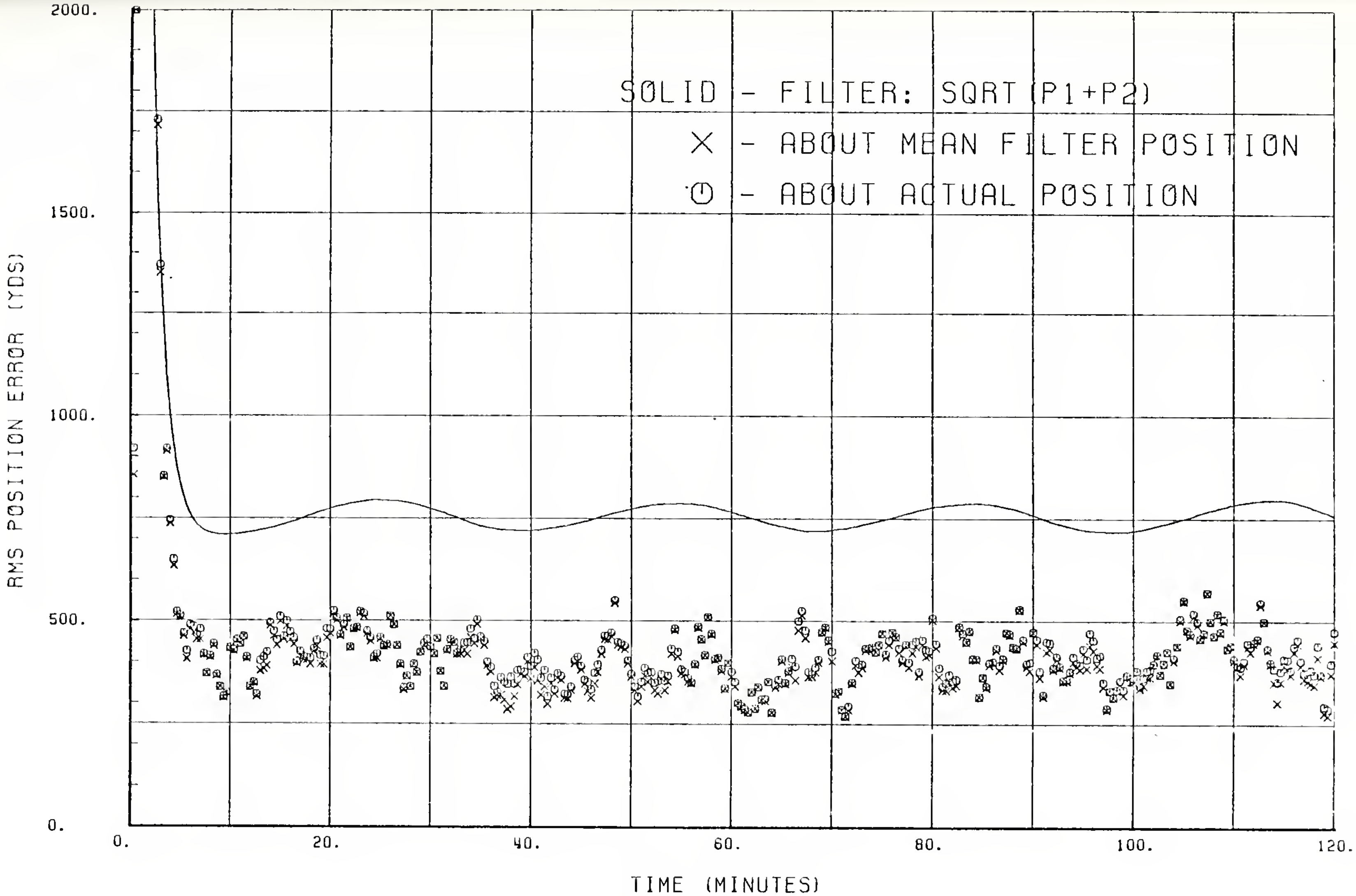


Figure 22. RMS position errors for the six-state system while flying a 15 NM circular pattern around the sonobuoy. DR=a/c 180-1 + Schuler, b 270-1), ∆t=20s, Alt=3000'.

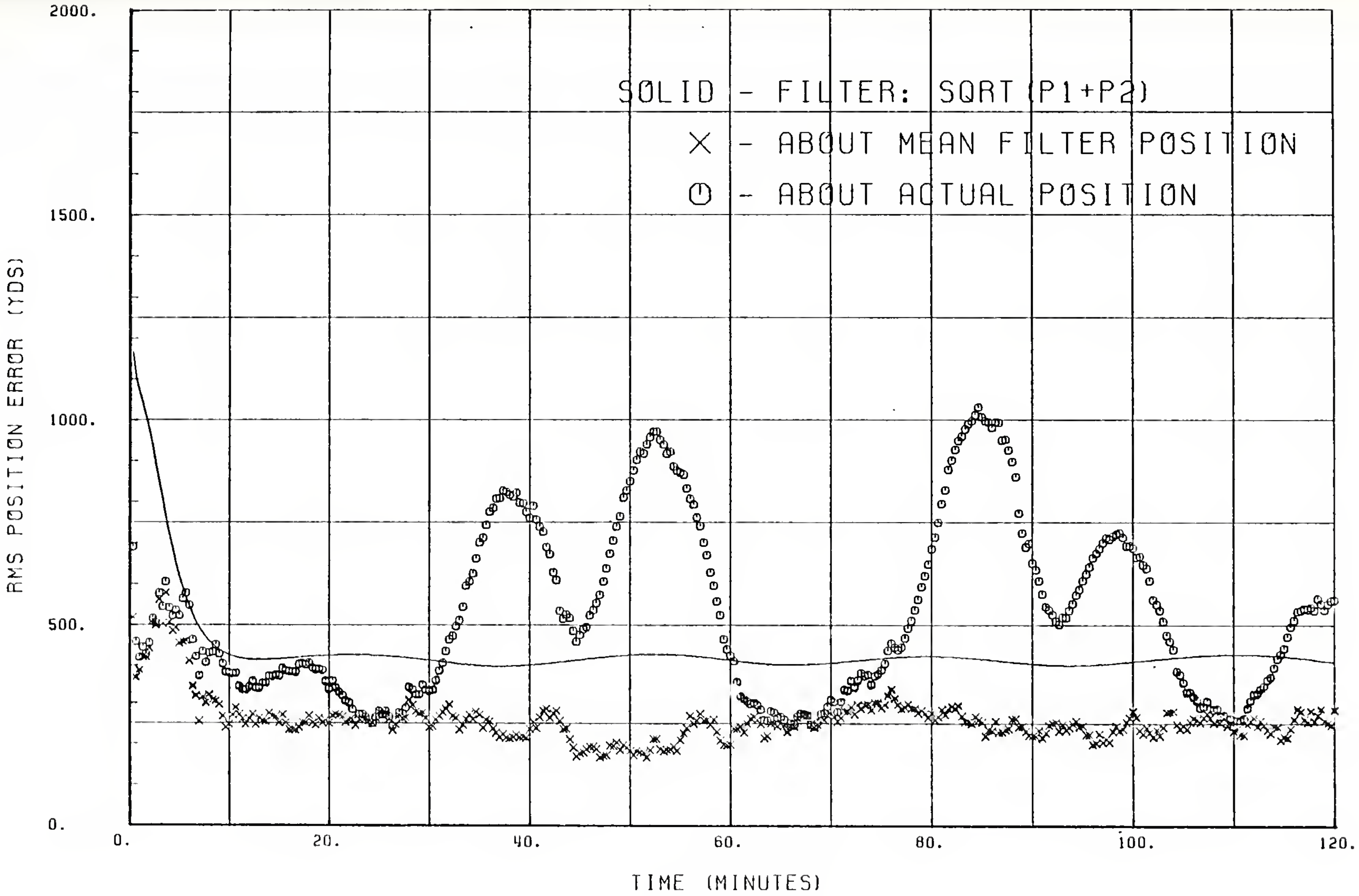


Figure 23. RMS position errors for the two-state system while flying a 15 NM circular pattern around the sonobuoy. DR=(a/c 180-1 + Schuler, b 270-1), Δt=20s, Alt=3000'.

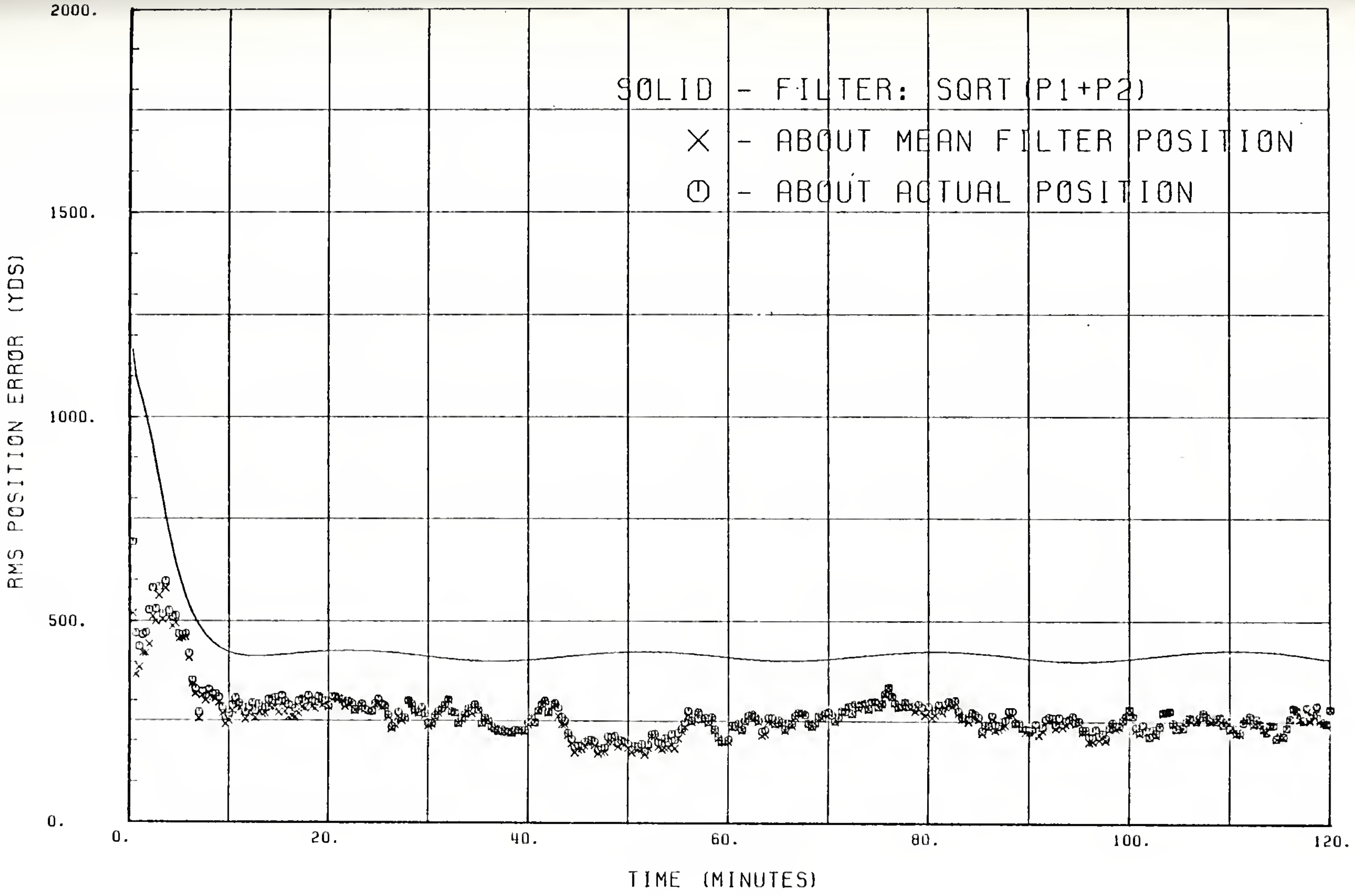


Figure 24. RMS position errors for the two-state system while flying a 15 NM circular pattern around the sonobuoy. DR=(a/c 180-1 + No Sch., b 270-1), t=20s, Alt=3000'.

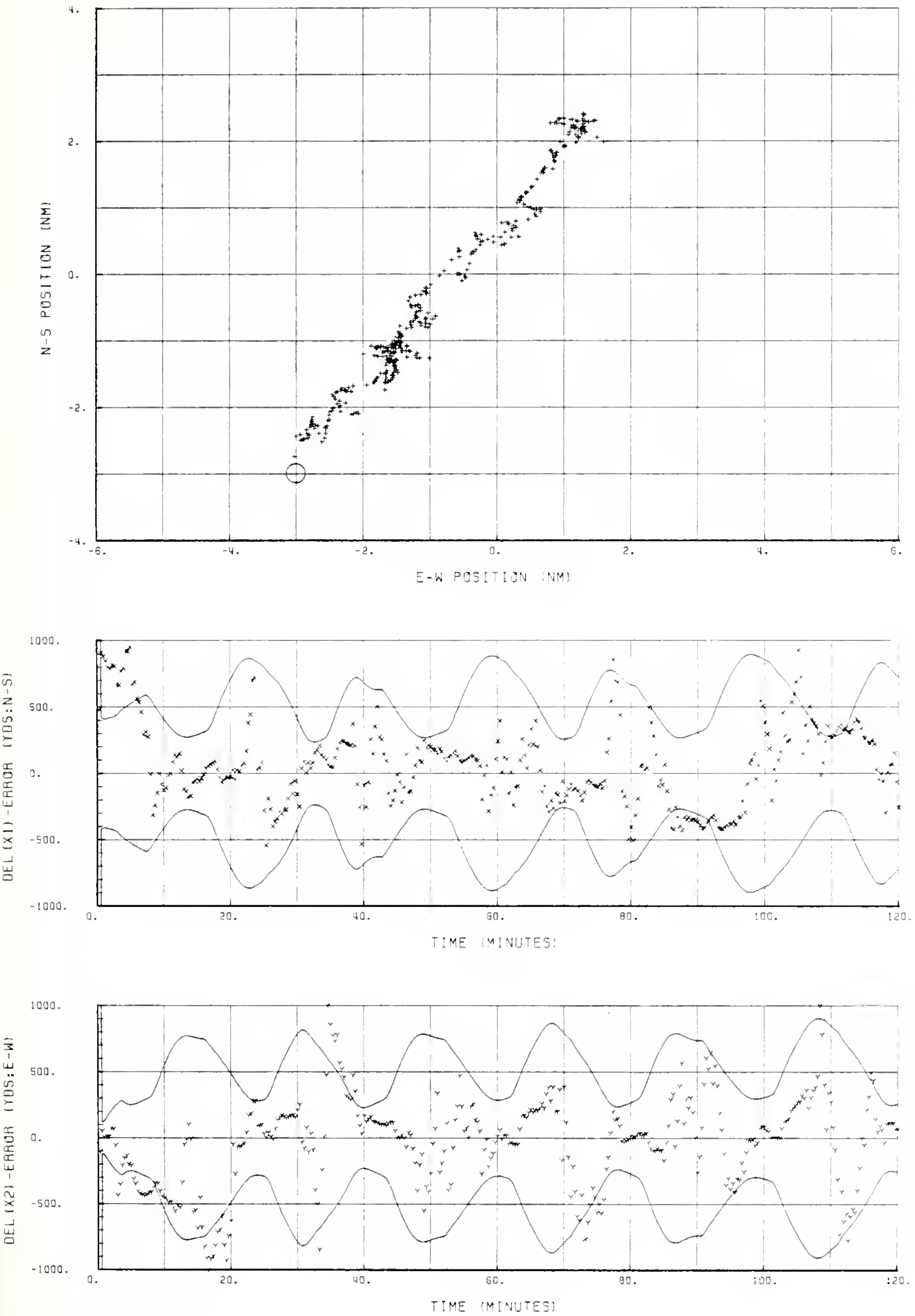


Figure 25. Est. position and relative errors for the six-state system using square pattern at 15NM. DR=(a/c 000-2.5 + Schuler, b 090-2), Δt=20s, Alt=3000'

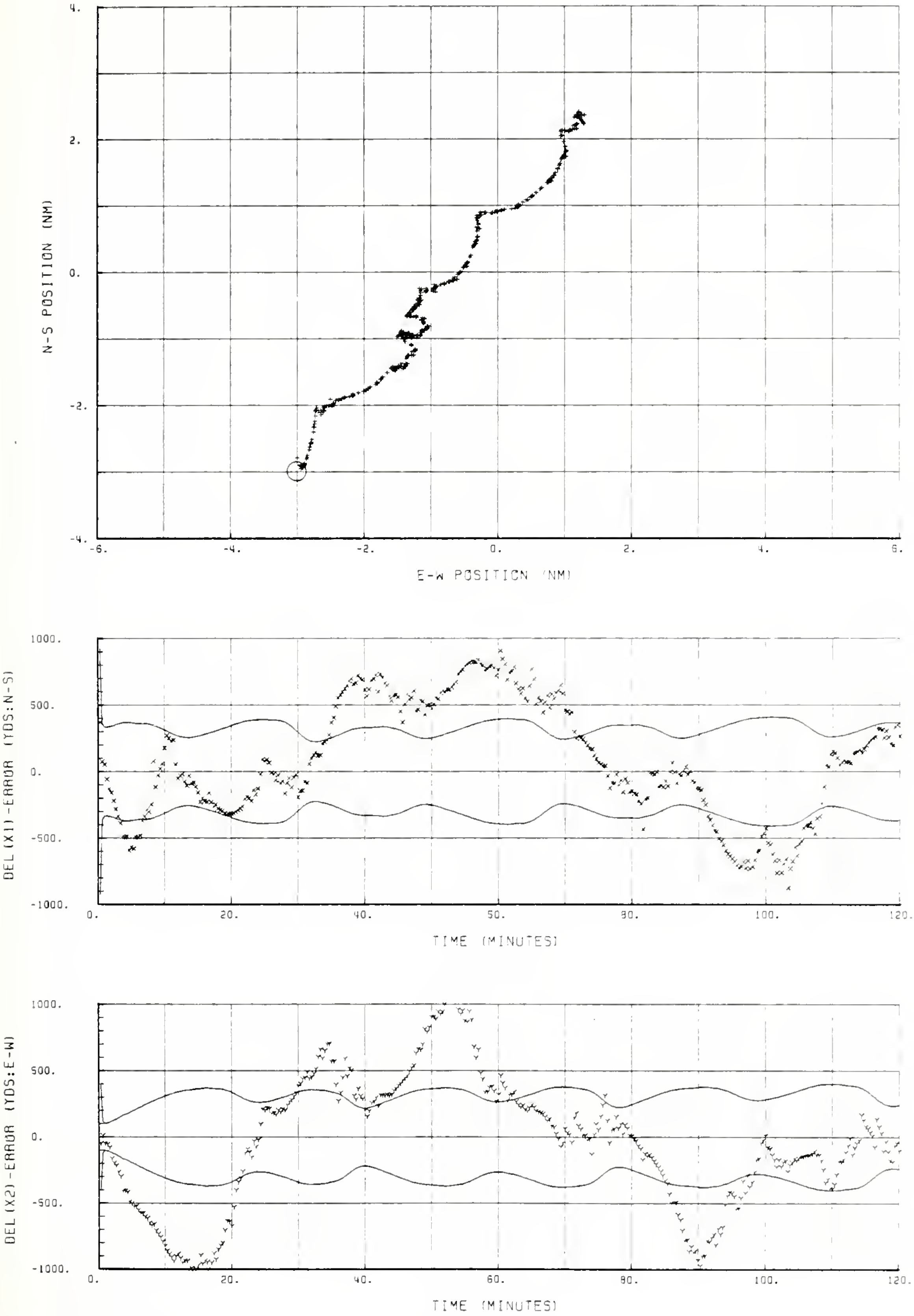


Figure 26. Est. position and relative errors for the two-state system using square pattern at 15NM. DR=(a/c 000-2.5 + Schuler, b 090-2), Δt=20s, Alt=3000'

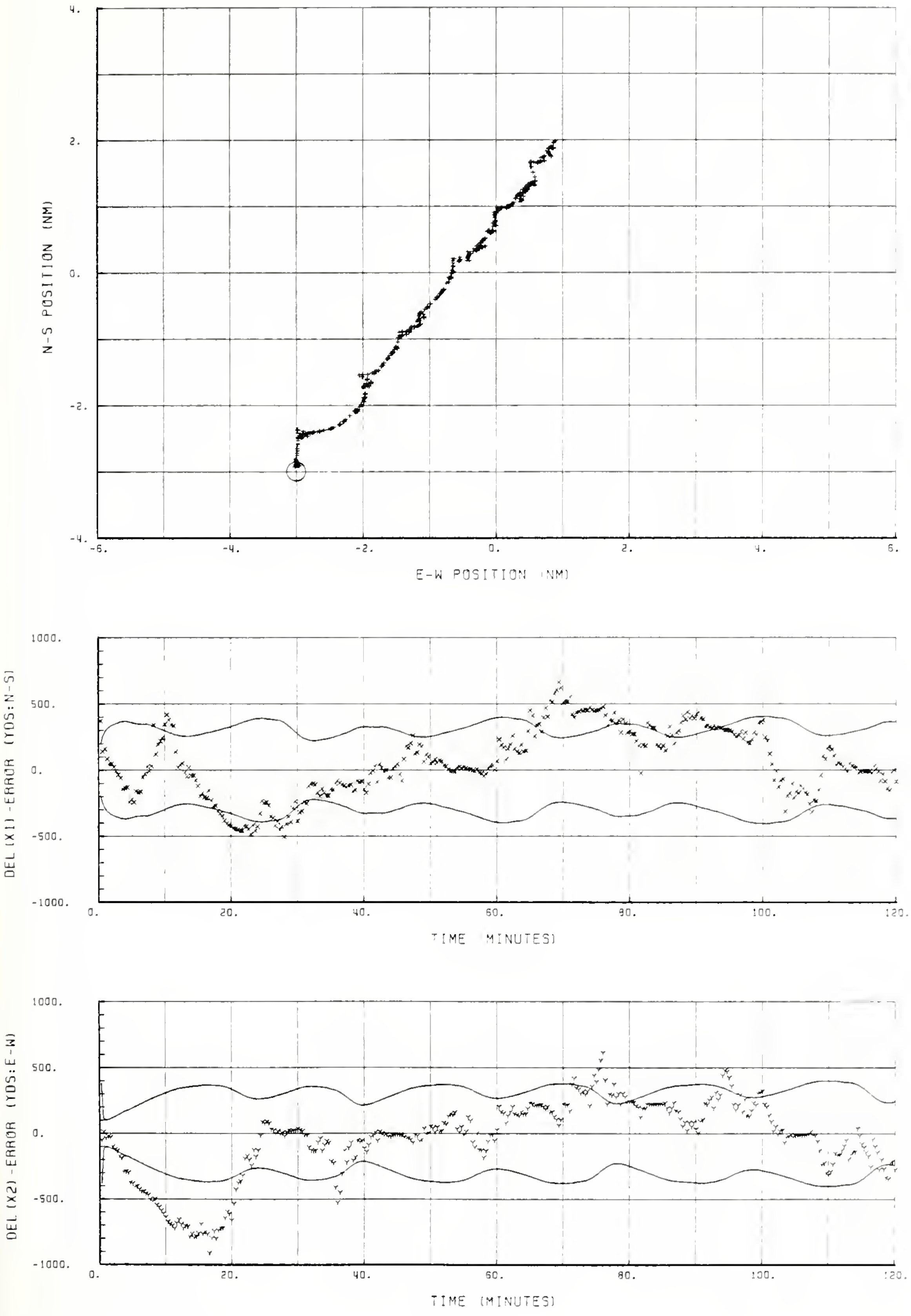


Figure 27. Est. position and relative errors for the two-state system using square pattern at 15NM. DR=(a/c 000-2.5 + No Sch., b 090-2), Δt=20s, Alt=3000'

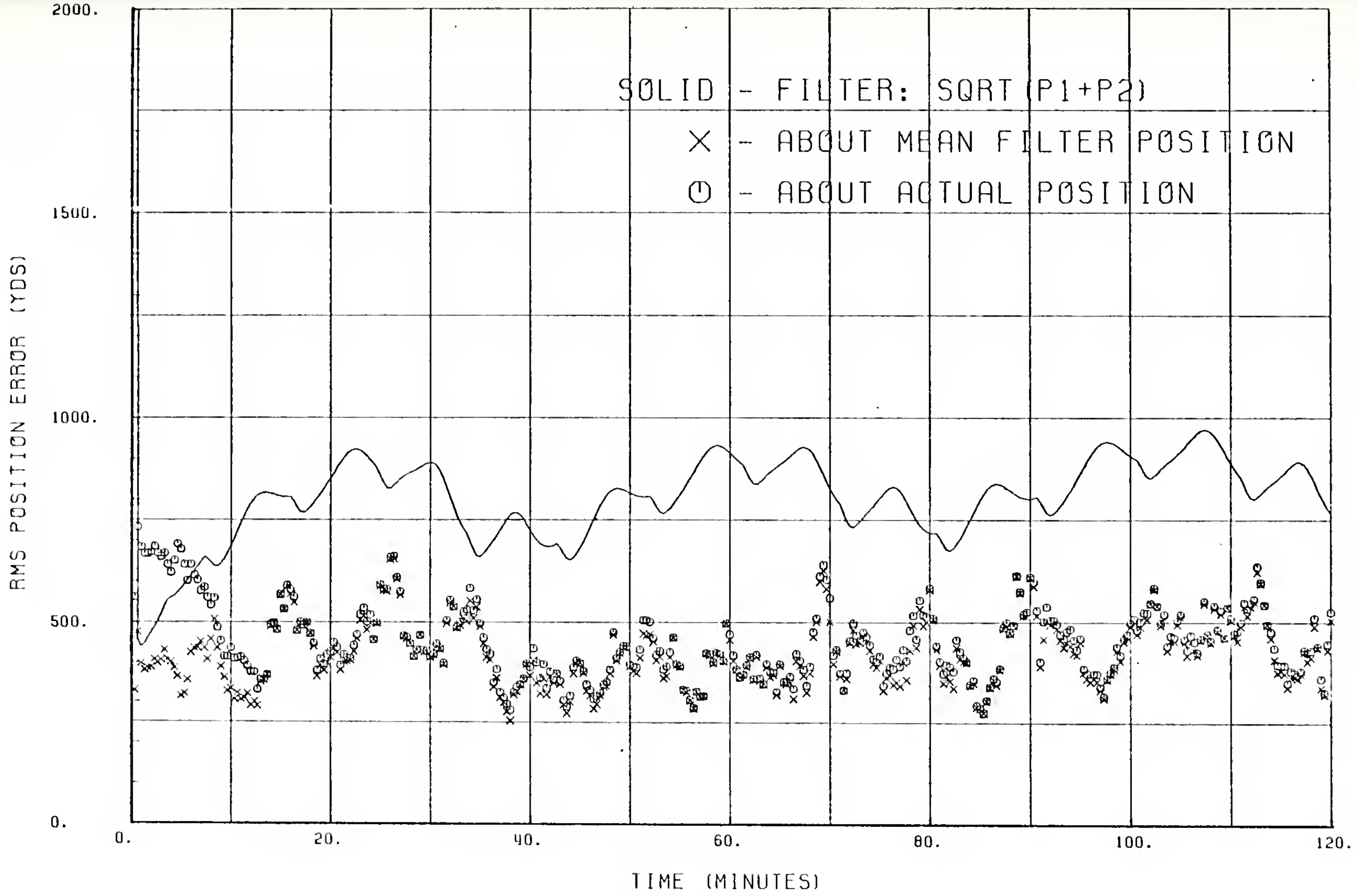


Figure 28. RMS position errors for the six-state system while flying a 15 NM square pattern around the sonobuoy. DR=(a/c 000-2.5 + Schuler, b 090-2), t=20s, Alt=3000'.

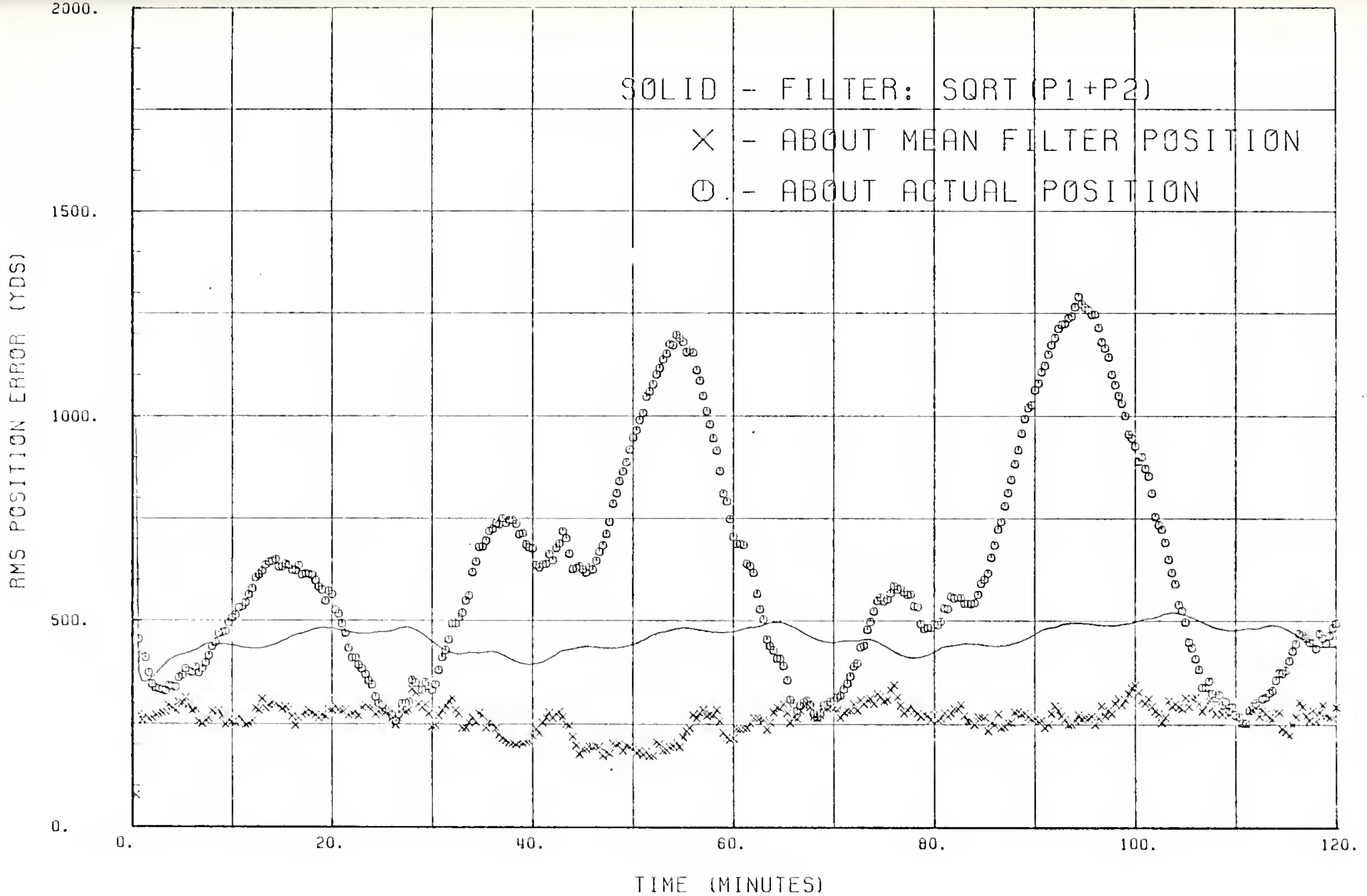


Figure 29. RMS position errors for the two-state system while flying a 15 NM square pattern around the sonobuoy. DR=(a/c 00-2.5 + Schuler, b 090-2), Δt=20s, Alt=3000'.

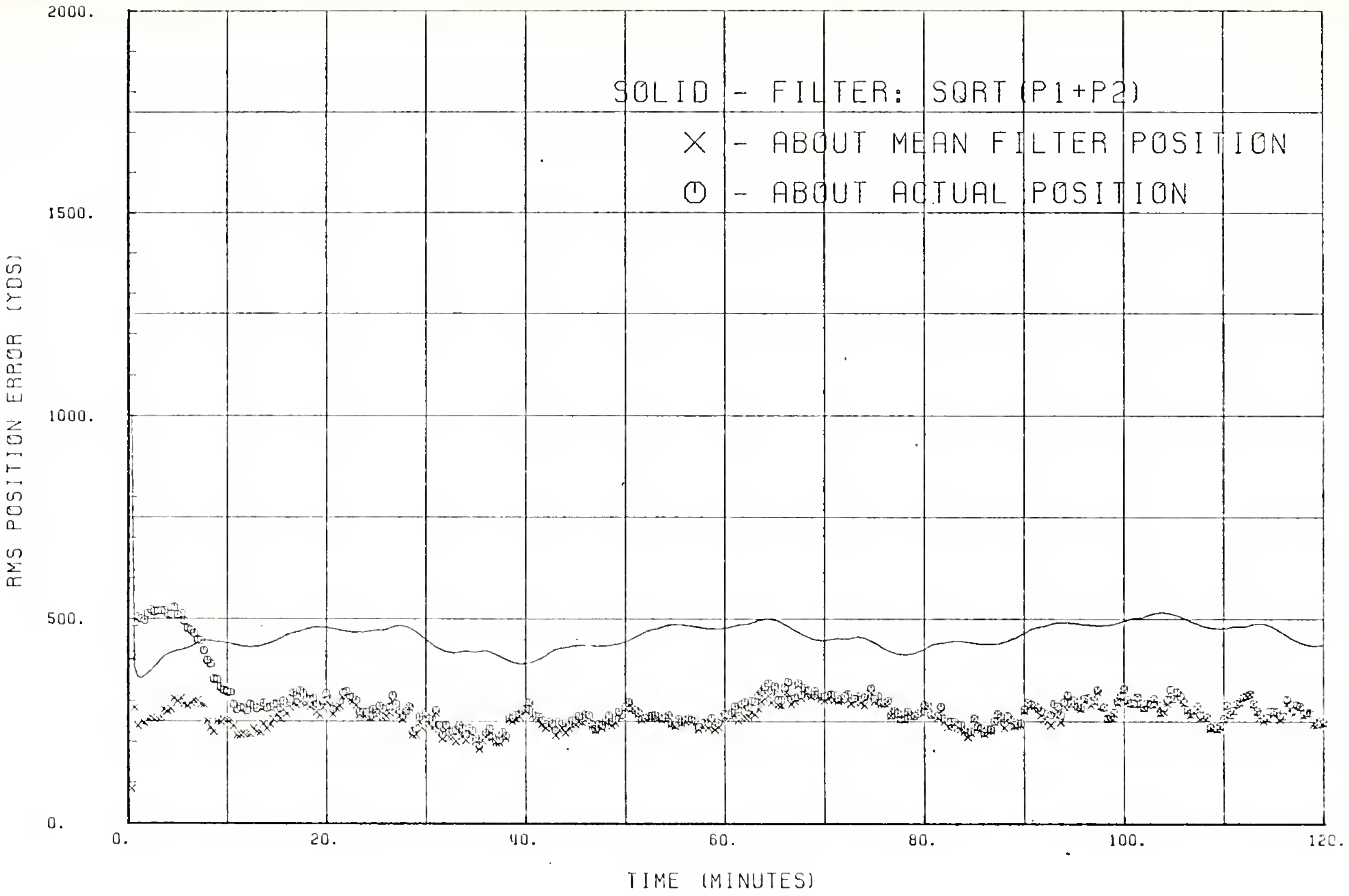


Figure 30. RMS position errors for the two-state system while flying a 15 NM square pattern around the sonobuoy. DR(a/c 000-2.5 + No Sch., b 090-2), Δt=20s, Alt=3000'.

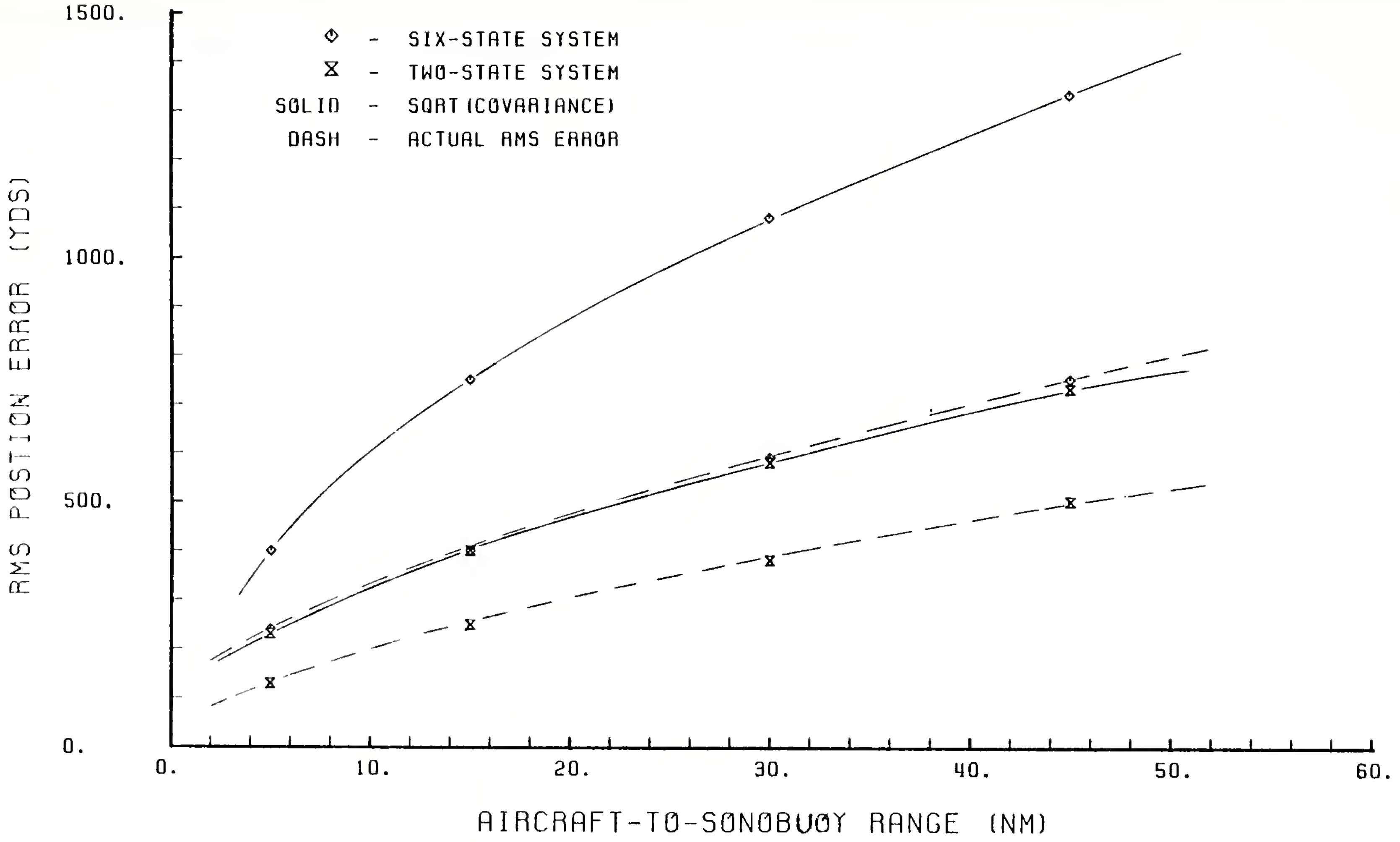


Figure 31. RMS position error as a function of aircraft-to-sonobuoy range, Δx, for the six-state and two-state systems.

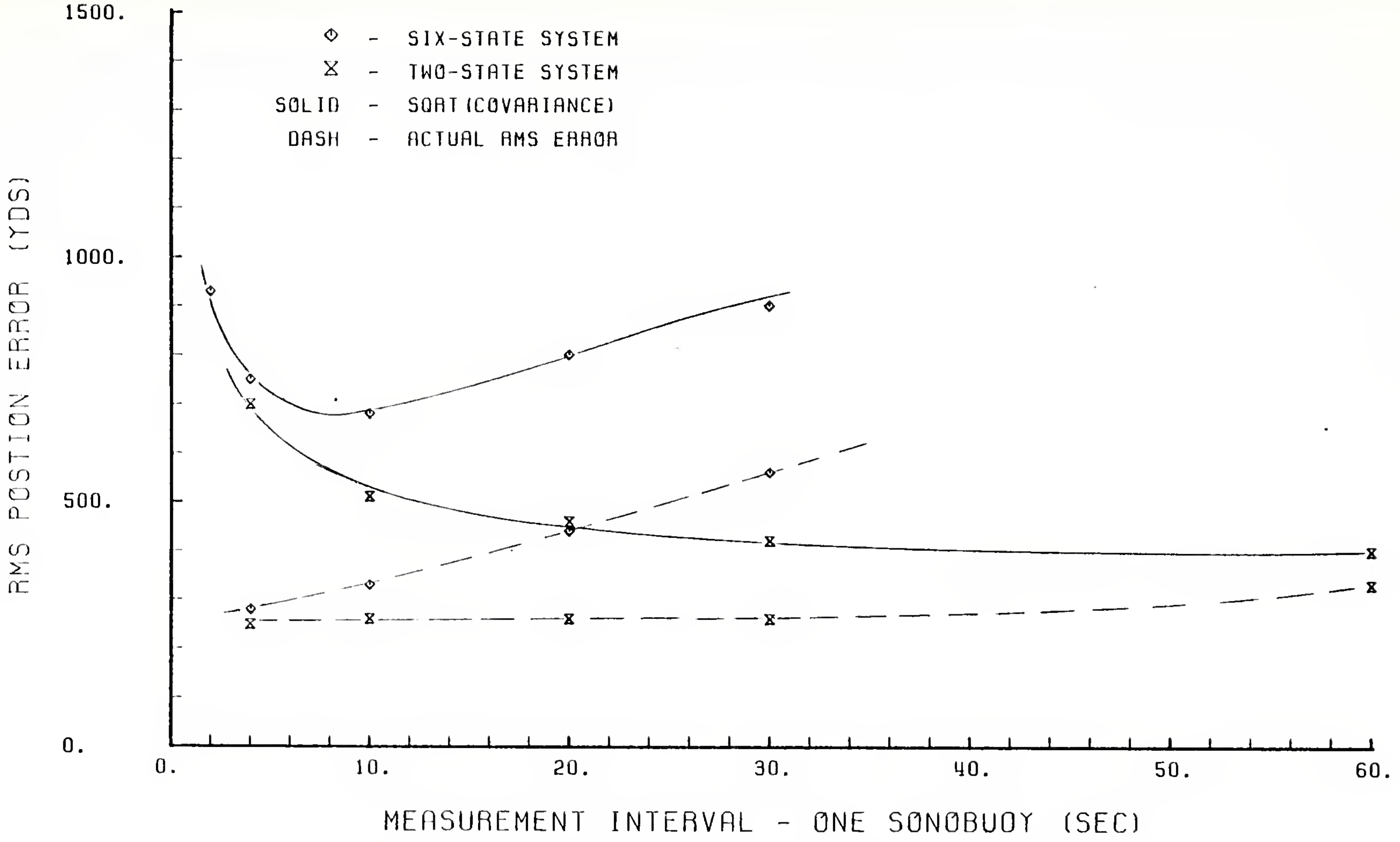


Figure 32. RMS position error as a function of measurement interval, Δt, for the six-state and two-state systems.

V. CONCULSION

The following are the conculsions reached at the end of this research:

1. The six-state algorithm developed in this report produced estimates of relative sonobuoy position with a standard deviation in the error of 500 yards at a range of 20 NM. Aircraft drift, which included the Schuler cycle, had no effect on this system since the states were defined relative to the aircraft.
2. The two-state algorithm developed in this report produced estimates of relative sonobuoy position with a standard deviation of 300 yards plus an amount equal to the amplitude of the Schuler cycle error at a range of 20 NM. When no Schuler cycle navigational errors were present, the two-state system out-performed the six-state system.
3. Both algorithms showed an increase in steady state RMS error with an increase in range.
4. The six-state system showed an increase in actual steady state RMS error for an increase in the time interval between measurements; however, the covariance of this system indicated minimum error occurred with a 10 second interval on one sonobuoy at a range of 15 NM and an aircraft speed of 180 kts. (This translates to a 1½ degree change in relative bearing.) The two-state

system showed no change in the actual steady state RMS error for intervals between 4 and 30 seconds and only a slight increase in these errors as a 60 second interval was approached. The covariance of the system actually decreased with increasing interval.

5. Neither system showed any dependence on altitudes within the operational limits at a range of 15 NM.
6. No adverse effect was observed while flying straight tracks as opposed to curved ones.
7. Steady state RMS error was affected by two conditions as measurement intervals changed. The longer the interval the more the error tended to increase, especially for the six-state algorithm, as a result of system propagation. The shorter the interval the more the error tended to increase as the systems encroached upon their non-observability condition. An optimum measurement interval existed for each system which was a function of range.

VI. SUMMARY

Both the six-state and the two-state algorithms were able to successfully track the sonobuoy. However, the six-state system was the best choice given the navigational system in use by the Orion. It was capable of providing estimated sonobuoy positions at a range of 2Ø NM such that the errors had a standard deviation of 5ØØ yards. The computer time required to process one measurement was on the order of 2ØØ milliseconds. On the other hand, the much simpler two-state system required only 28 milliseconds or one seventh ($\frac{1}{7}$) the amount of computer time. However, its estimate of sonobuoy position was degraded by Schuler cycle errors in the navigational system. Without these, this system was subject to errors having a standard deviation of 3ØØ yards at a range of 2Ø NM. The Schuler cycle periodically increased these errors by an amount equal to its amplitude.

The algorithms were very much affected by the aircraft´s flight path including the relative location of the sonobuoy, the range to the sonobuoy, and the amount of angular change in the bearing to the sonobuoy as measurements were taken. Since the sonobuoy position was to be estimated in two-dimensions, the aircraft was required to maneuver in such a way so as to provide information in both dimensions. For example, as the aircraft crossed north or south of the

sonobuoy the E-W errors decreased while the N-S errors increased. The trend reversed when the aircraft flew east or west of the sonobuoy. It was also observed that steady state RMS errors grew with range at a slightly decreasing rate. They were twice as large at 45 NM as they were at 15 NM. Finally, the sonobuoy position was not observable if the relative bearing to the sonobuoy did not change implying that two lines of bearing must cross in order to obtain an estimated position. If the amount of angular change between each measurement was too small it began to affect the errors because this condition was being approached. Anytime the aircraft flew in such a way as to reduce the rate of bearing change, or the frequency at which measurements were made was too high, this condition was prevalent. It was observed that at least 1½ degrees of change was required to avoid this problem.

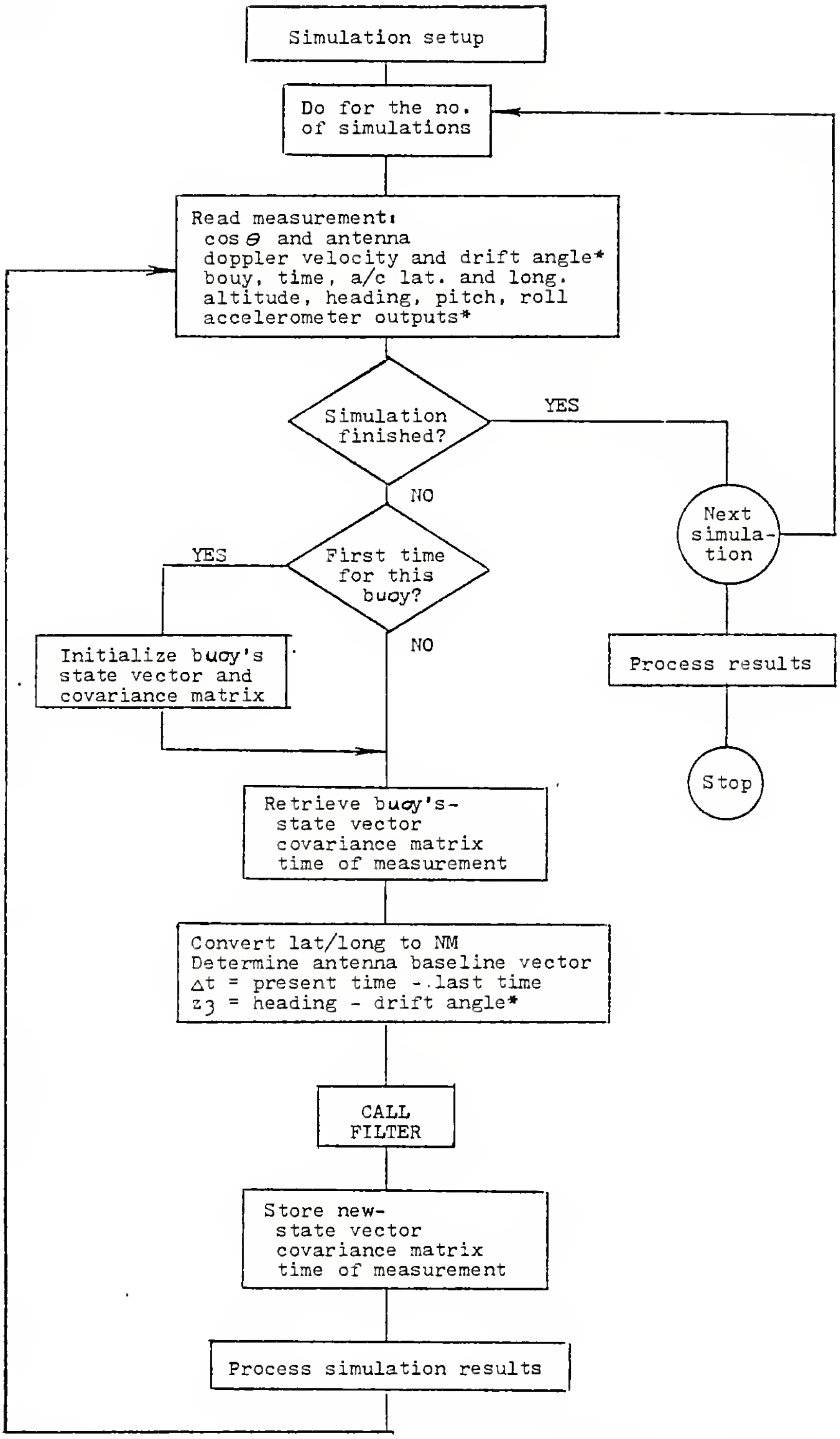
The six-state system showed no tendency toward divergence as long as initial conditions were reasonable. The two-state system, on the other hand, did have a tendency to diverge when the gains on the drift filter were not adjusted correctly. For example, if the gain was to high the estimated drift was heavily dependent on the last few measurements and was too quickly affected by changes in the position filter. But, it was also necessary for the drift filter to have non-zero gains in the steady state in order to provide some insurance against an initially wrong estimate of sonobuoy drift. With these modifications made

the algorithm operated correctly during all tests and, in fact, had less error in some cases than did the six-state system.

It is the opinion of the author that either one of these algorithms is an improvement on the historical method of "marking-on-top" the sonobuoy. Further, it adds a great deal more uncertainty as to exactly how much accuracy could be obtained in the old way. Non-linear navigational effects were not even considered and the number of measurements made to determine sonobuoy position was on the order of 10 or less for any one sonobuoy pattern. Nevertheless, in an operational sense the accuracies achieved here would need to be improved. Without the non-linear navigational errors the accuracies could have been much better. However, the results of this research do provide another perspective for the problem at hand.

APPENDIX A
THE COMPUTER PROGRAMS

MAIN



Simulation setup
Do for the no. of simulations
Read measurement:
cos θ and antenna
doppler velocity and drift angle*
bouy, time, a/c lat. and long.
altitude, heading, pitch, roll
accelerometer outputs*
Simulation finished?
YES
NO
Next simula-tion
Process results
Stop
First time for this buoy?
YES
NO
Initialize buoy's state vector and covariance matrix
Retrieve buoy's-
state vector
covariance matrix
time of measurement
Convert lat/long to NM
Determine antenna baseline vector
Δt = present time - last time
z3 = heading - drift angle*
CALL FILTER
Store new-
state vector
covariance matrix
time of measurement
Process simulation results

Figure 34. Flow chart: main program

* Required for six-state system only.

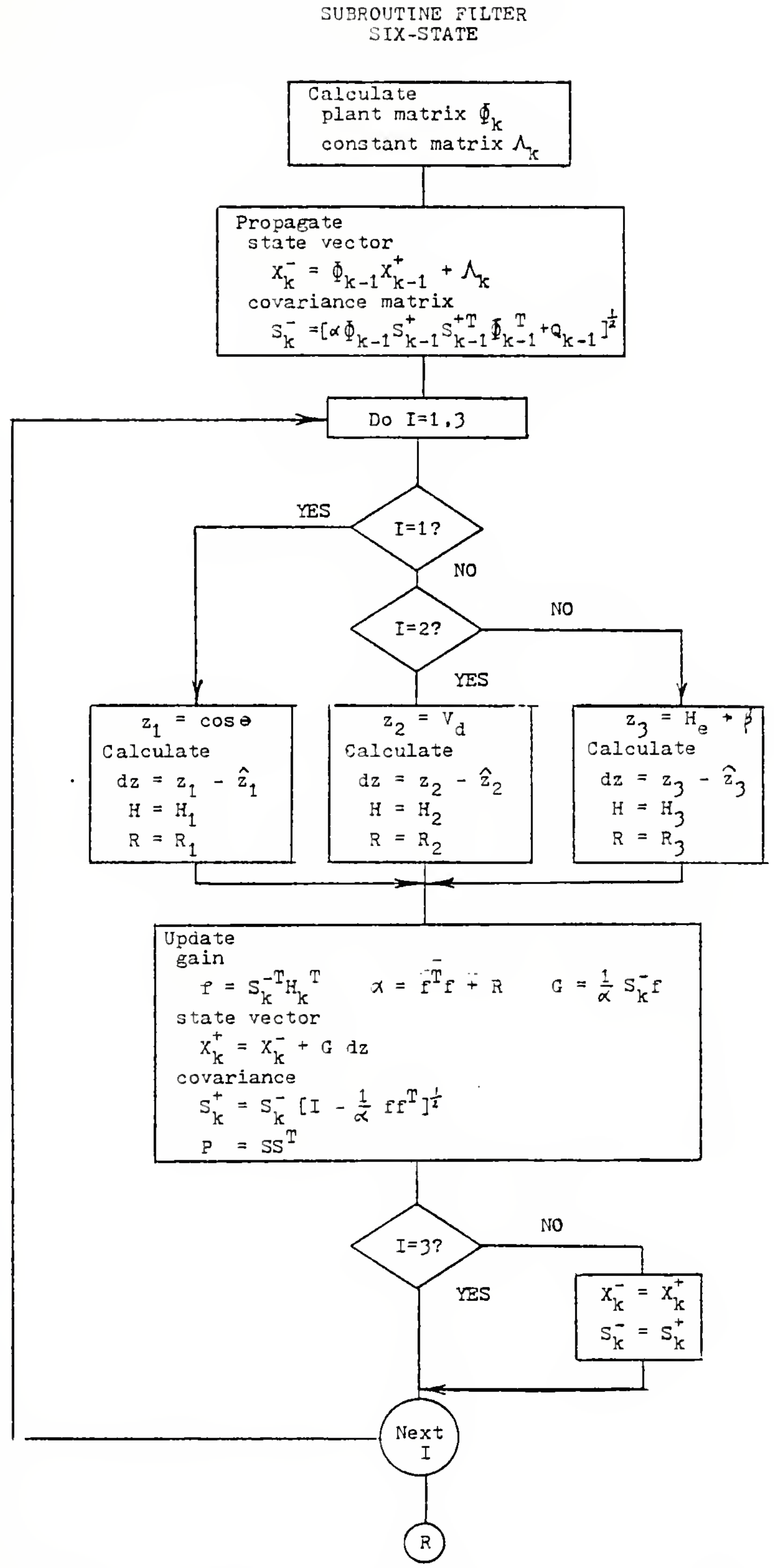


Figure 35. Flow chart: Subroutine FILTER six-state

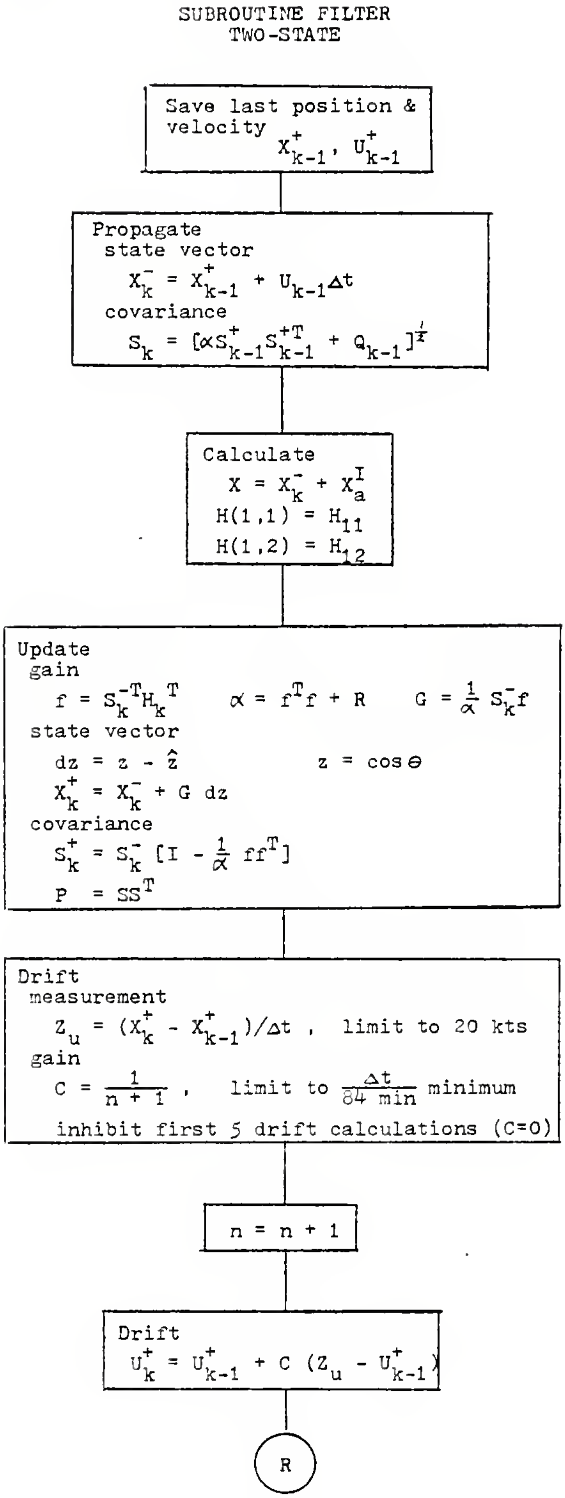

SUBROUTINE FILTER
TWO-STATE
Save last position & velocity X^+_{k-1}, U^+_{k-1}
Propagate
state vector
$X^-_k = X^+_{k-1} + U_{k-1}\Delta t$
covariance
$S_k = [\alpha S^+_{k-1} S^{+T}_{k-1} + Q_{k-1}]^{\frac{1}{2}}$
Calculate
$X = X^-_k + X^I_a$
$H(1,1) = H_{11}$
$H(1,2) = H_{12}$
Update
gain
$f = S^{-T}_k H^T_k$
$\alpha = f^T f + R$
$G = \frac{1}{\alpha} S^-_k f$
state vector
$dz = z - \hat{z}$
$z = \cos\theta$
$X^+_k = X^-_k + G\ dz$
covariance
$S^+_k = S^-_k [I - \frac{1}{\alpha} ff^T]$
$P = SS^T$
Drift
measurement
$Z_u = (X^+_k - X^+_{k-1})/\Delta t$, limit to 20 kts
gain
$C = \frac{1}{n+1}$, limit to $\frac{\Delta t}{84 \text{ min}}$ minimum
inhibit first 5 drift calculations (C=0)
$n = n + 1$
Drift
$U^+_k = U^+_{k-1} + C\ (Z_u - U^+_{k-1})$
R

Figure 36. Flow chart: Subroutine FILTER two-state

GLOSSARY OF COMPUTER VARIABLES

SIMULATION PROGRAM

A	Vector of aircraft accelerations
ALPHAI	Result of $\frac{1}{f^T f + R}$ or $\frac{1}{\alpha}$
ALT	Altitude
ANGLE	Direction of sonobuoy drift from North
ANT	Antenna
AVEL	Initial aircraft velocity
AS	Matrix result of $I - \frac{ff^T}{\alpha}$
B	Matrix result of $\{ \varepsilon SS^T + Q\}^{\frac{1}{2}}$
BOUY	Sonobuoy RF channel or number
C	Gain of drift filter
CA	Result of Δt/(84 min.)
CK	Constant matrix in six-state system´s equations
CKL	Rate of change of nautical miles for a change in degrees of longitude
D	Array which holds all items requiring storage from one measurement to the next
DELT	Interval between measurements, Δt
DELX1	Relative position, Δx_1
DELX2	Relative position, Δx_2
DELX3	Relative position, Δx_3
DRIFT	Magnitude of sonobuoy drift in yards
DSEED	Seed for the random number generator
DTIME	Array of initial drop times for the sonobuoy
DZ	Residual, dz
EDELX1	Error in the est. relative position of sonobuoy, Δx_1

EDELX2	Error in the est. relative position of sonobuoy, Δx_2
EOSSO	Matrix result of $\varepsilon\ \Phi^T S^T S\ \Phi$
EPSLN	Age weighting factor,
ESST	Matrix result of $\varepsilon\ S\ S^T$
EX1	Error in est. geographical position of sonobuoy, Δx_1
EX2	Error in est. geographical position of sonobuoy, Δx_2
F	f vector from Carlson square root technique
FFT	Matrix result of ff^T
FFTA	Matrix result of $\frac{ff^T}{\alpha}$
FT	Transpose of f
G	Gain matrix
GAMMA	Aircraft pitch angle, γ
GDZ	Vector result of G dz
GR	Gamma in radians
H	Aircraft heading
H	The H matrix when used in subroutine FILTER
HOUR	Time in hours
HR	Heading in radians
HT	Transpose of the H matrix
I	Index
I	Identity matrix when used in subroutine FILTER
IEND	Indicator of EOF
IFFTA	Matrix result of $I - \frac{ff^T}{\alpha}$
IN	File from which data is read
INRES	File from which true scnobuoy positions are read
IOUT	Device on which print is made
IOUTP	File in which results are written

IOUTZ	File in which random vectors are inspected
IRUN	Index for run number
IZ	Index for measurement, $z = 1,2,3$
K	Incremental measure of time
LAT	Latitude
LATI	The latitude of the origin of the local earth fixed coordinate system
LONG	Longitude
LONGI	The longitude of the origin of the local earth fixed coordinate system
MARK	Indicates whether information is to be stored or retrieved
MEAN1	Mean estimated sonobuoy position, μ_1
MEAN2	Mean estimated sonobuoy position, μ_2
MIN	Time in minutes
N	Total number of measurements made on all sonobuoys
N	Index when used in subroutines INITAL and DATA
NRUNS	Integer equivalent of RUNS
O	Plant matrix, Φ
OS	Matrix result of Φ S
OST	Transpose of Φ S^T
OSSO	Matrix result of $\Phi^T S^T S$ Φ
OX	Matrix result of Φ X
P	Covariance matrix
PHI	Aircraft roll angle, ϕ
PI	π
PR	Phi in radians
Q	Variance of plant noise from N(Ø,Q)
R	Variance of measurement noise from N(Ø,R)

RAD	π /180
RANGE	Range to sonobuoy from aircraft
RANGE2	Squared value of range
RB1	Unit vector component of antenna baseline
RB2	Unit vector component of antenna baseline
RB3	Unit vector component of antenna baseline
RBX	Result of $(RB_1 \Delta x_1 + RB_2 \Delta x_2 + RB_3 \Delta x_3)$
RNO	Normally distributed random number from N(0,1)
RUNS	Number of simulation runs to be made
S	Array in Carlson's square root technique which represents square root of covariance matrix
S1	Square root of variance, $\sqrt{P_{11}}$
S2	Square root of variance, $\sqrt{P_{22}}$
SEC	Time in seconds
SF	Matrix result of S f
SIGMA	RMS error in the estimate about the actual location
SIGMAF	RMS value from the Kalman filter, $\sqrt{P_{11} + P_{22}}$
SIGMAX	RMS error in the estimate about the mean estimate
SKL	Rate of change of nautical miles for a change in degrees of latitude
SNEXT	Propagated matrix of S
SST	Matrix result of S S^T
ST	Transpose of S matrix
SUMP1	Array for the sum of P(1,1) over NRUNS
SUMP2	Array for the sum of P(2,2) over NRUNS
SUMSQ1	Array for sum of $\Delta\tilde{x}_1^2$ over NRUNS
SUMSQ2	Array for sum of $\Delta\tilde{x}_2^2$ over NRUNS
SUMX1	Array for sum of $\Delta\tilde{x}_1$ over NRUNS

SUMX2	Array for sum of Δx_2 over NRUNS
T	Last time in seconds the sonobuoy was processed by the filter
TIME	Time from aircraft clock in seconds
U	Sonobuoy drift vector
U1	Old value of U(1)
U2	Old value of U(2)
VAR1	Variance of the estimated sonobuoy position about the actual location σ_1^2
VAR2	Variance of the estimated sonobuoy position about the actual location σ_2^2
VARX1	Variance of the estimated sonobuoy position about the mean estimated position, σ^2_{x1}
VARX2	Variance of the estimated sonobuoy position about the mean estimated position, σ^2_{x2}
VELCT	Relative velocity, ΔV
VELCT2	Squared value of relative veclocity
X	State vector
XAI	Aircraft inertial position, X_a^I
XAT	Aircraft true position, X_a
X1	Old value of X(1)
X2	Old value of X(2)
XB	Vector including XB1, XB2, velocity of sonobuoy, and two zeros used in the six-state system
XB1	True geographical location of sonobuoy, x_1
XB2	True geographical location of sonobuoy, x_2
XB1I	Array of initial sonobuoy positions
XB2I	Array of initial sonobuoy positions
XNEXT	Propagated value of X
Z	Measurement vector

ZHAT	Estimate of z
ZN	Normally distributed random number from N(z,R)
ZU	Vector of velocity measurement for drift filter

```
C     ************************************
C     *          MAIN PROGRAM             *
C     *              FOR                  *
C     *       SIX-STATE SYSTEM            *
C     ************************************

      COMMON P(500)
      DIMENSION S(6,6),P(6,6),X(6),XAT(2),XB(6),Z(3),MARK(6),A(2)
      REAL LAT,LONG
      INTEGER BDAY,HOUR,MIN,SEC,TIME,T,ANT
      DOUBLE PRECISION DSEED
      DATA MARK, PI/6*0, 3.1415926/
C
C        SIMULATION SETUP
C
      DSEED=7654321.D00
      NRUNS=2
      RUNS=FLOAT(NRUNS)
C
      IN=5
      IOUT=6
      WRITE(IOUTP,9005)
      IN=7
      INRES=IN+1
      DO 70 IRUN=1,NRUNS
      WRITE(IOUT,9006) IRUN
      DO 10 I=1,500
   10 P(I)=0.
      DO 20 I=1,6
   20 MARK(I)=0
      READ (IN,8000)
      READ (INRES,8000)
 1000 WRITE(IOUTP,9007)
      AVEL=130.
C
C        READ MEASUREMENT DATA
C
 1000 WRITE(IOUT,9003)
   40 READ(IN,8001)BDAY,HOUR,MIN,SEC,Z,LAT,LONG,ALT,H,GAMMA,PHI,A,ANT
      WRITE(IOUT,8001)BDAY,HOUR,MIN,SEC,Z,LAT,LONG,ALT,H,GAMMA,PHI,A,ANT
      IF (BDAY.EQ.9) GO TO 80
C
C        RETRIEVE OLD DATA
C
   50 IF (MARK(BDAY).NE.0) GO TO 60
      CALL INITL(P,XB,ALT,H,AVEL)
```

```
   50 CALL DATA(BUOY,S,X,XB,T,K,MARK(BUOY))
C        PREPARE FOR FILTER
      CALL LATLON(LAT,LONG,XAI)
      CALL NOISE(DSEED,Z(1),.026)
      CALL NOISE(DSEED,Z(2),2.)
      CALL NOISE(DSEED,Z(3),.008)
      CALL BASE(ANT,H,GAMMA,PHI,KB1,KB2,KB3)
      TIME=3600*HOUR+60*MIN+SEC
      DELT=FLOAT(TIME-T)/3600.
      Z(3)=H+Z(3)
C        USE FILTER AND STORE NEW DATA
C9603 WRITE(IOUT,9002) BUOY,K,HOUR,MIN,SEC
      CALL FILTER(S,P,X,XB,Z,DELT,XAI,LAT,ALT,KB1,KB2,KB3,Q,BUOY,K)
      CALL DATA(BUOY,S,X,XB,TIME,K,MARK(BUOY))
      CALL RESULT(BUOY,K,X,XB,P,IN,RUNS)
      GO TO 40
C        CONTINUE SIMULATION
   60 REWIND IN
   70 REWIND INRES
 9701 CALL RESULT(S,P,X,XB,P,IN,RUNS)
      IEND=66400
      WRITE(IOUT,9004) IEND
 9000 FORMAT(////)
 9001 FORMAT(I3,3I2,F7.3,F6.1,F5.1,2F9.5,F6.0,F6.1,F4.1,F5.1,
     * 2F7.3,I2)
 9010 FORMAT(1H1)
 9011 FORMAT(I3,3I2,F7.3,F6.1,F5.1,2F9.5,F6.0,F6.1,F4.1,F5.1,
     * 2F7.3,I2)
 9002 FORMAT(' BUOY NO. ',I2,'    K=',I4,'     TIME ',I3,2(':',I2))
 9003 FORMAT(' RUN=',I2)
 9004 FORMAT(I8)
 9005 FORMAT(/,' 6-STATE: RUN 3/00, TT,TT, 5000FT,  TO=2S',/
     ,' 6-STATE: 15 F CIRCLE, DR (S/C 18)-1 AND SCHULER, R 270-1)')
 9006 FORMAT(' TIME=',F6.2)
 9007 FORMAT(' BUOY  K',4X,'XB1',6X,'XB2',6X,'LAB1',5X,'LAB2',
     *5X,'SIGLX1',5X,'SIGLX2',3X,'SIGMA1',3X,'SIGMA2')

      STOP
      END
```

```
C     ****************************************************
C                 SUBROUTINES FIRST LEVEL                *
C     ****************************************************


      SUBROUTINE LATLON(LAT,LONG,XAI)

      REAL LAT,LONG,LATI,LONGI
      DIMENSION XAI(2)
      DATA PI,A1,A2,A3,A4/3.1415926,2.9035546E-04,5.7582098E-07,
     *     1.0067682,0.7681702E-13/
      LATI=36.
      LONGI=70.
      RP=1./(A1-A2*(SIN(LAT))**2)
      RM=RP/(A3-A4*(SIN(LAT))**2)
      SKL=RP*PI/180.
      CKL=RM*PI/180.
      XAI(1)=(LAT-LATI)*SKL
      XAI(2)=(LONG-LONGI)*CKL
      RETURN
      END



      SUBROUTINE BASE(ANT,H,GAMMA,PHI,RB1,RB2,RB3)

      INTEGER ANT
      RAD=3.1415926/180.
      GR=GAMMA*RAD
      PR=PHI*RAD
      HR=H*RAD
      IF(ANT.EQ.3) GO TO 1
      RB1=-COS(PR)*SIN(HR)
      RB2=COS(PR)*COS(HR)
      RB3=SIN(PR)
      GO TO 5
 1    RB1=COS(GR)*COS(HR)
      RB2=COS(GR)*SIN(HR)
      RB3=-SIN(GR)
 5    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INITAL(BUOY,ALT,H,AVEL)                                DAT01490
                                                                        DAT01500
      COMMON D(500)                                                     DAT01510
      DIMENSION XB1I(6),XB2I(6),DTIME(6)                                DAT01520
      INTEGER BUOY,T                                                    DAT01530
      DATA PI/3.1415926/                                                DAT01540
      K=64*BUOY-47                                                      DAT01550
                                                                        DAT01560
C     ** INITIAL BUOY DATA **                                           DAT01570
                                                                        DAT01580
      DATA XB1I/-3.,-5.,-10.,-15.,-24.,0./                              DAT01590
      DATA XB2I/-3.5,-5.5,-10.5,-15.5,0.,0./                            DAT01600
      DATA DTIME/0.,140.,280.,420.,3460.,0./                            DAT01610
                                                                        DAT01620
C     ** INITIAL VARIANCES **                                           DAT01630
                                                                        DAT01640
C     D(K)=ALT/6076.115                                                 DAT01650
C     D(K+7)=ALT/6076.115                                               DAT01660
      D(K)=5.                                                           DAT01670
      D(K+7)=5.                                                         DAT01680
      D(K+14)=5.                                                        DAT01690
      D(K+21)=5.                                                        DAT01700
      D(K+28)=.0025                                                     DAT01710
      D(K+35)=.0025                                                     DAT01720
                                                                        DAT01730
C     ** INITIAL REL. POSITION **                                       DAT01740
                                                                        DAT01750
      D(K+36)=0.                                                        DAT01760
      D(K+37)=.5                                                        DAT01770
                                                                        DAT01780
C     ** INITIAL REL. VELOCITY **                                       DAT01790
                                                                        DAT01800
      D(K+38)=-AVEL*COS(H*PI/180.)                                      DAT01810
      D(K+39)=-AVEL*SIN(H*PI/180.)                                      DAT01820
                                                                        DAT01830
C     ** INITIAL BUOY POSITION **                                       DAT01840
                                                                        DAT01850
      D(K+42)=XB1I(BUOY)                                                DAT01860
      D(K+43)=XB2I(BUOY)                                                DAT01870
                                                                        DAT01880
C     ** INITIAL BUOY VELOCITY **                                       DAT01890
                                                                        DAT01900
      D(K+44)=0.                                                        DAT01910
      D(K+45)=0.                                                        DAT01920
                                                                        DAT01930
C     ** DELTA TIME **                                                  DAT01940
                                                                        DAT01950
      D(K+46)=DTIME(BUOY)                                               DAT01960
```

```
      RETURN
      END



      SUBROUTINE DATA(BUOY,S,X,XB,T,K,MARK)

      COMMON D(500)
      DIMENSION S(6,6),X(6),XB(6)
      INTEGER BUOY,T
      N=54*BUOY-47
      IF(MARK.GE.2) GO TO 2

C     ** RETRIEVE **

    1 DO 10 I=1,6
      DO 20 J=1,6
      S(I,J)=D(N)
   20 N=N+1
   10 CONTINUE
      X(1)=D(N)
      X(2)=D(N+1)
      X(3)=D(N+2)
      X(4)=D(N+3)
      X(5)=D(N+4)
      X(6)=D(N+5)
      XB(1)=D(N+6)
      XB(2)=D(N+7)
      XB(3)=D(N+8)
      XB(4)=D(N+9)
      XB(5)=0.
      XB(6)=0.
      T=IFIX(D(N+10))
      K=IFIX(D(N+11)) + 1
      MARK=2
      GO TO 50

C     ** STORE **

    2 DO 30 I=1,6
      DO 40 J=1,6
      D(N)=S(I,J)
   40 N=N+1
   30 CONTINUE
      D(N)=X(1)
      D(N+1)=X(2)
      D(N+2)=X(3)
```

```
      D(M+3)=X(4)                                                       DAT02450
      D(M+4)=X(5)                                                       DAT02460
      D(M+5)=X(6)                                                       DAT02470
      D(M+6)=XB(1)                                                      DAT02480
      D(M+7)=XB(2)                                                      DAT02490
      D(M+8)=XB(3)                                                      DAT02500
      D(M+9)=XB(4)                                                      DAT02510
      D(M+10)=FLOAT(T)                                                  DAT02520
      D(M+11)=FLOAT(K)                                                  DAT02530
      MARK=1                                                            DAT02540
   90 CONTINUE                                                          DAT02550
      RETURN                                                            DAT02560
      END                                                               DAT02570
                                                                        DAT02580
                                                                        DAT02590
                                                                        DAT02600
                                                                        DAT02610
      SUBROUTINE FILTER(S,P,X,XB,Z,DELT,XAI,LAT,ALT,RB1,RB2,RB4,A,BODY,KDAT02620
     *)                                                                 DAT02630
                                                                        DAT02640
      DIMENSION P(6,6),G(6),X(6),HT(6),XNEXT(6),GDZ(6),SNEXT(6,6),      DAT02650
     *   H(1,6),Q(6,6),I(6,6),S(6,6),                                   DAT02660
         F(6),FT(1,6),FFT(6,6),R(6,6),BETA(6,6),IBETA(6,6),AS(6,6),     DAT02670
     *   ST(6,6),SST(6,6),SF(6),J(6,6),DX(6),CK(6),DS(6,6),             DAT02680
         JJT(6,6),QSSQ(6,6),Z(3),XB(6),XAT(2),A(2),FQSSQ(6,6)           DAT02690
      REAL I,IBETA,LAT,LONG,LAM                                         DAT02700
      INTEGER BODY                                                      DAT02710
      DATA Q/36*0./                                                     DAT02720
      DATA I/36*0./                                                     DAT02730
      DATA C/36*0./                                                     DAT02740
      DATA CK/6*0./                                                     DAT02750
      DATA PI,EPSLN/3.1415926,1./                                       DAT02760
      IPUN=0                                                            DAT02770
                                                                        DAT02780
    1 XB1=XB(1)                                                         DAT02790
      XB2=XB(2)                                                         DAT02800
      XB3=XB(3)                                                         DAT02810
      XB4=XB(4)                                                         DAT02820
                                                                        DAT02830
      GM=32.2*3600.**2/6076.115                                         DAT02840
      LAM=LAT*PI/180.                                                   DAT02850
      W=.00124*3600.                                                    DAT02860
      WEAR=.7292E-04*3600.                                              DAT02870
      WT=W*DELT                                                         DAT02880
      SWT=WEAR*SIN(LAM)                                                 DAT02890
      SHT=WEAR*SIN(LAM)*DELT                                            DAT02900
      DI=(WEAR*SIN(LAM))**2-W**2                                        DAT02910
      Q(1,1)=.001                                                       DAT02920
```

```
      Q(2,2)=.001                                                       DAT02930
      Q(3,3)=.25                                                        DAT02940
      Q(4,4)=.25                                                        DAT02950
      Q(5,5)=.001                                                       DAT02960
      Q(6,6)=.001                                                       DAT02970
      DO 2  L=1,6                                                       DAT02980
      I(L,L)=1.0                                                        DAT02990
                                                                        DAT03000
C        PLANT MATRIX                                                   DAT03010
                                                                        DAT03020
    2 O(L,L)=COS(WT)                                                    DAT03030
      O(1,3)=SIN(WT)/W                                                  DAT03040
      O(1,5)=-GR*(SNL/W*SIN(WT)-SIN(SNLT))/DEN                          DAT03050
      O(1,6)=-GR*(COS(WT)-COS(SNLT))/DEN                                DAT03060
      O(2,4)=O(1,3)                                                     DAT03070
      O(2,5)=-O(1,6)                                                    DAT03080
      O(2,6)=O(1,5)                                                     DAT03090
      O(3,1)=-W*SIN(WT)                                                 DAT03100
      O(3,5)=SNL*O(1,6)                                                 DAT03110
      O(3,6)=GR*(W*SIN(WT)-SNL*SIN(SNLT))/DEN                           DAT03120
      O(4,2)=O(3,1)                                                     DAT03130
      O(4,5)=-O(3,6)                                                    DAT03140
      O(4,6)=O(3,5)                                                     DAT03150
      O(5,5)=COS(SNLT)                                                  DAT03160
      O(5,6)=-SIN(SNLT)                                                 DAT03170
      O(6,5)=-O(5,6)                                                    DAT03180
      O(6,6)=O(5,5)                                                     DAT03190
                                                                        DAT03200
      R2DOT1=A(1)                                                       DAT03210
      R2DOT2=A(2)                                                       DAT03220
      CK(1)=(-R2DOT1/W**2+XB(1)-XAI(1))*(1-COS(WT))                     DAT03230
      CK(2)=(-R2DOT2/W**2+XB(2)-XAI(2))*(1-COS(WT))                     DAT03240
      CK(3)=(-R2DOT1/W+W*(XB(1)-XAI(1)))*SIN(WT)                        DAT03250
      CK(4)=(-R2DOT2/W+W*(XB(2)-XAI(2)))*SIN(WT)                        DAT03260
                                                                        DAT03270
                                                                        DAT03280
C       PROPAGATE                                                       DAT03290
C        **X(K/K-1)**                                                   DAT03300
                                                                        DAT03310
   10 CALL  GMPRD(O,X,OX,6,6,1)                                         DAT03320
      CALL  GMADD(OX,CK,XNEXT,6,1)                                      DAT03330
                                                                        DAT03340
C        ** S(K/K-1)**                                                  DAT03350
                                                                        DAT03360
   20 CALL  GMPRD(O,S,OS,6,6,6)                                         DAT03370
      CALL  GMTRA(O,OT,6,6)                                             DAT03380
      CALL  GMPRD(OS,OT,OSSC,6,6,6)                                     DAT03390
      CALL  GMSPA(OSS ,EPSLN,OSS ,6,6)                                  DAT03400
```

```
      CALL GMADD(POSSO,Q,B,6,6)                                         DAT03410
      CALL GMSRT(B,SNEXT,6)                                             DAT03420
                                                                        DAT03430
   30 DO 79 IZ=1,3                                                      DAT03440
      RANGE2=XNEXT(1)**2+XNEXT(2)**2+(ALT/6076.115)**2                  DAT03450
      RANGE=SQRT(RANGE2)                                                DAT03460
      RBX=RB1*XNEXT(1)+RB2*XNEXT(2)+RB3*(ALT/6076.115)                  DAT03470
      VV2=(XNEXT(3)**2+XNEXT(4)**2)                                     DAT03480
      VV=SQRT(VV2)                                                      DAT03490
      XB(1)=X(1)+XAI(1)                                                 DAT03500
      XB(2)=X(2)+XAI(2)                                                 DAT03510
                                                                        DAT03520
C        H MATRIX AND Z                                                 DAT03530
                                                                        DAT03540
   40 DO 41 L=1,6                                                       DAT03550
   41 H(1,L)=0.                                                         DAT03560
                                                                        DAT03570
      IF(IZ.NE.1) GO TO 42                                              DAT03580
      DZ=Z(1)-RBX/RANGE                                                 DAT03590
      H(1,1)=(RB1-XNEXT(1)*RBX/RANGE2)/RANGE                            DAT03600
      H(1,2)=(RB2-XNEXT(2)*RBX/RANGE2)/RANGE                            DAT03610
      R=.0007                                                           DAT03620
      GO TO 50                                                          DAT03630
                                                                        DAT03640
   42 IF(IZ.NE.2) GO TO 43                                              DAT03650
      DZ=Z(2)-VV                                                        DAT03660
      H(1,3)=XNEXT(3)/VV                                                DAT03670
      H(1,4)=XNEXT(4)/VV                                                DAT03680
      R=1.0                                                             DAT03690
      GO TO 50                                                          DAT03700
                                                                        DAT03710
   43 IF(XNEXT(4).NE.0.) GO TO 44                                       DAT03720
      ZHAT=0.                                                           DAT03730
      IF(XNEXT(3).LT.0.) ZHAT=PI                                        DAT03740
      GO TO 45                                                          DAT03750
   44 ZHAT=ATAN(XNEXT(3)/XNEXT(4))                                      DAT03760
      IF(XNEXT(4).LE.0.) ZHAT=PI/2.-ZHAT                                DAT03770
      IF(XNEXT(4).GT.0.) ZHAT=3.*PI/2.-ZHAT                             DAT03780
   45 DZ=Z(3)*PI/180.-ZHAT                                              DAT03790
      IF(DZ.GT.PI) DZ=2*PI-DZ                                           DAT03800
      IF(DZ.LT.-PI) DZ=2*PI+DZ                                          DAT03810
      H(1,3)=-XNEXT(4)/VV2                                              DAT03820
      H(1,4)=XNEXT(3)/VV2                                               DAT03830
      R=.01                                                             DAT03840
                                                                        DAT03850
                                                                        DAT03860
                                                                        DAT03870
                                                                        DAT03880
```

```
C        UPDATE                                                          DAT03890
C         **G(K)**                                                       DAT03900
                                                                         DAT03910
   50   CALL GMTRA(H,HT,1,6)                                             DAT03920
        CALL GMTRA(SMEXT,ST,6,6)                                         DAT03930
        CALL GMPRD(ST,HT,F,6,6,1)                                        DAT03940
        CALL GMPRD(SMEXT,F,SF,6,6,1)                                     DAT03950
        CALL GMTRA(F,FT,6,1)                                             DAT03960
        CALL GMPRD(FT,F,FTF,1,6,1)                                       DAT03970
        ALPHAI=1/(P+FTF)                                                 DAT03980
        CALL GMSPR(SF,ALPHAI,G,6,1)                                      DAT03990
                                                                         DAT04000
C         **X(K/K)**                                                     DAT04010
                                                                         DAT04020
   60   CALL GMPRD(G,DZ,GDZ,6,1,1)                                       DAT04030
        CALL GMADD(XNEXT,GDZ,X,6,1)                                      DAT04040
C9814 WRITE(IOUT,901) DZ                                                  DAT04050
C9815 WRITE(IOUT,941) (XNEXT(L),(P(L,J),J=1,6),G(L),H(1,L),L=1,6)        DAT04060
                                                                         DAT04070
C         **S(K/K)**                                                     DAT04080
                                                                         DAT04090
   70   CALL GMPRD(F,FT,FFT,6,1,6)                                       DAT04100
        CALL GMSPR(FFT,ALPHAI,FFTA,6,6)                                  DAT04110
        CALL GMSUB(I,FFTA,IFFTA,6,6)                                     DAT04120
        CALL GMSRT(IFFTA,AS,6)                                           DAT04130
        CALL GMPRD(SMEXT,AS,S,6,6,6)                                     DAT04140
        CALL GMTRA(S,ST,6,6)                                             DAT04150
        CALL GMPRD(S,ST,P,6,6,6)                                         DAT04160
        IF(IZ.EQ.3) GO TO 73                                             DAT04170
        DO 72 L=1,6                                                      DAT04180
        DO 71 J=1,6                                                      DAT04190
   71   SMAT(L,J)=S(L,J)                                                 DAT04200
   72   XMAT(L)=X(L)                                                     DAT04210
   73   CONTINUE                                                         DAT04220
C9804 WRITE(IOUT,902)                                                     DAT04230
C9805 WRITE(IOUT,942) (X(L),(P(L,J),J=1,6),L=1,6)                         DAT04240
                                                                         DAT04250
C       COMPUTATION OF BUOY POSITION                                     DAT04260
                                                                         DAT04270
   80   XB(1)=X(1)+XAI(1)                                                DAT04280
        XB(2)=X(2)+XAI(2)                                                DAT04290
                                                                         DAT04300
  941   FORMAT(/,6(5X,F10.5,10X,6F10.5,10X,F10.5,10X,F10.5,/))           DAT04310
  942   FORMAT(/,6(5X,F10.5,10X,6F10.5,10X,F10.5,/))                     DAT04320
  901   FORMAT(/,'   X(K/K-1)-MATRIX:',9X,'P(K/K-1)-MATRIX:',53X,'DZ=',   DAT04330
       * F9.5,/1X,'G(K)-MATRIX:',12X,'H',8X)                             DAT04340
  902   FORMAT(/,'   X(K/K)-MATRIX:',11X,'P(K/K)-MATRIX:',53X,'BUOY POS ANDAT04350
       * VEL:')                                                          DAT04360
```

```
      RETURN
      END

      SUBROUTINE RESULT(BUDY,K,X,XB,P,IN,RUNS)

      DIMENSION X(6),P(6,6),XB(6),SUMP1(1800),SUMP2(1800),XAT(2),
     *SUMX1(1800),SUMX2(1800),SUMSQ1(1800),SUMSQ2(1800)
      REAL MEAN1,MEAN2
      INTEGER BUDY
      DATA SUMX1,SUMX2,SUMSQ1,SUMSQ2 /1800*0.,1800*0.,1800*0.,1800*0./
      DATA SUMP1,SUMP2 /1800*0.,1800*0./
      INRES=IN+1
      IOUTP=10
      IF(BUDY.EQ.9) GO TO 30

   10 READ(INRES,9) XB1,XB2,XAT
      EXB1=XB(1)-XB1
      EXB2=XB(2)-XB2
      EDELX1=X(1)-XB1+XAT(1)
      EDELX2=X(2)-XB2+XAT(2)
      IF(BUDY.NE.1) GO TO 20
      SUMX1(K)=SUMX1(K)+EDELX1
      SUMX2(K)=SUMX2(K)+EDELX2
      SUMSQ1(K)=SUMSQ1(K)+EDELX1**2
      SUMSQ2(K)=SUMSQ2(K)+EDELX2**2
      SUMP1(K)=SUMP1(K)+P(1,1)
      SUMP2(K)=SUMP2(K)+P(2,2)
   20 CONTINUE
      S1=SQRT(P(1,1))
      S2=SQRT(P(2,2))
 9667 WRITE(IOUTP,3) BUDY,K,XB(1),XB(2),EXB1,EXB2,EDELX1,EDELX2,S1,S2
      RETURN

   30 READ(INRES,6)
 9702 WRITE(IOUTP,4)
      DO 50 I=1,K
      READ(INRES,9) XB1,XB2,XAT
      MEAN1=SUMX1(I)/RUNS
      MEAN2=SUMX2(I)/RUNS
      VARX1=SUMSQ1(I)/RUNS-MEAN1**2
      VARX2=SUMSQ2(I)/RUNS-MEAN2**2
      VAR1=SUMSQ1(I)/RUNS
      VAR2=SUMSQ2(I)/RUNS
      SIGMAX=SQRT(VARX1+VARX2)
      SIGMA=SQRT(VAR1+VAR2)
      SIGMAF=SQRT(SUMP1(I)/RUNS+SUMP2(I)/RUNS)
 9705 WRITE(IOUTP,5) I,SIGMAX,SIGMA,SIGMAF
   50 CONTINUE
```

```
    4 FORMAT('          K',4X,'SIGMAX',4X,'SIGMA',5X,'SIGMAF')          DAT04850
    5 FORMAT(I6,3F20.5)                                                 DAT04860
    7 FORMAT(////)                                                      DAT04870
    8 FORMAT(2I4,3F9.5)                                                 DAT04880
    9 FORMAT(3X,4F9.4)                                                  DAT04890
      RETURN                                                            DAT04900
      END                                                               DAT04910
                                                                        DAT04920
                                                                        DAT04930
                                                                        DAT04940
                                                                        DAT04950
      SUBROUTINE NOISE(DSEED,Z,STD)                                     DAT04960
                                                                        DAT04970
      DOUBLE PRECISION DSEED                                            DAT04980
      IOUTZ=9                                                           DAT04990
      RND=GGNQF(DSEED)                                                  DAT05000
      ZN=RND*STD+Z                                                      DAT05010
CC    WRITE(IOUTZ,9200) Z,ZN,RND                                        DAT05020
      Z=ZN                                                              DAT05030
 9200 FORMAT(3F20.8)                                                    DAT05040
      RETURN                                                            DAT05050
      END                                                               DAT05060
```

```
C     ******************************************
C     *      SUBROUTINES SECOND LEVEL          *
C     ******************************************


      SUBROUTINE GMTRA(A,R,N,M)

      DIMENSION A(N,M), R(M,N)
      DO 10 I=1,N
      DO 20 J=1,M
   20 R(J,I)=A(I,J)
   10 CONTINUE
      RETURN
      END


      SUBROUTINE GMADD(A,B,R,N,M)

      DIMENSION A(1),B(1),R(1)
      NM=N*M
      DO 10 I=1,NM
   10 R(I)=A(I)+B(I)
      RETURN
      END


      SUBROUTINE GMSUB(A,B,R,N,M)

      DIMENSION A(1),B(1),R(1)
      NM=N*M
      DO 10 I=1,NM
   10 R(I)=A(I)-B(I)
      RETURN
      END


      SUBROUTINE GMPRD(A,B,R,N,M,L)

      DIMENSION A(1),B(1),R(1)
      IR=0
      IK=-M
      DO 10 K=1,L
      IK=IK+M
      DO 10 J=1,N
      IR=IR+1
      JI=J-N
      IB=IK
```

```
      R(IR)=0
      DO 10 I=1,M
      JI=JI+N
      IB=IB+1
10    R(IR)=R(IR)+A(JI)*B(IB)
      RETURN
      END


      SUBROUTINE GMSPR(A,S,AS,N,M)

      DIMENSION A(N,M),AS(N,M)
      DO 10 I=1,N
      DO 20 J=1,M
20    AS(I,J)=A(I,J)*S
10    CONTINUE
      RETURN
      END


      SUBROUTINE GMSRT(B,A,N)

      DIMENSION B(N,N),A(N,N)
      DO 8 JJ=1,N
      J=N+1-JJ
      JP1=J+1
      DO 7 II=1,N
      I=N+1-II
      IF(I.GT.J) GO TO 6
      IF(I.EQ.J) GO TO 3
      AA=0.
      IF(JP1.GT.N) GO TO 2
      DO 1 L=JP1,N
1     AA=AA+A(I,L)*A(J,L)
2     A(I,J)=(B(I,J)-AA)/A(J,J)
      GO TO 7
3     AIK2=0.
      IF(JP1.GT.N) GO TO 5
      DO 4 K=JP1,N
4     AIK2=AIK2+A(I,K)**2
5     A(I,I)=SQRT(B(I,I)-AIK2)
      GO TO 7
6     A(I,J)=0.
7     CONTINUE
8     CONTINUE
      RETURN
      END
```

```
C     SETUP:                                                    DAT06050
C      056   IOUTP FOR ERRORS                                   DAT06060
C       57   IOUTP FOR SPLOT                                    DAT06070
C      058   PRINTS                                             DAT06080
C      IN=7                                                     DAT06090
C      NRUNS=2                                                  DAT06100
C      AVEL=180.                                                DAT06110
C      DATA XF1I...-3.,-5.,....                                 DAT06120
C      DATA XF2I...-3.5,-5.5,...                                DAT06130
C      E(N+56)=0.                                               DAT06140
C      E(N+57)=.5                                               DAT06150
```

```
C           ****************************************
C           *          MAIN PROGRAM                *
C           *              FOR                     *
C           *        TWO-STATE SYSTEM              *
C           ****************************************

      COMMON L(50)
      DIMENSION S(2,2),P(2,2),X(2),U(2),XAI(2),MARK(6)
      REAL LAT,LONG
      INTEGER BUOY,HOUR,MIN,SEC,TIME,T,ANT,CLOCK(6)
      DOUBLE PRECISION DSEED
      DATA PI/3.1415926/

C        SIMULATION SETUP

      DSEED=7654321.0D0
      NRUNS=20
      RUNS=FLOAT(NRUNS)

      IOUT=6
      IOUTP=10
      WRITE(IOUTP,9005)
      IN=7
      INMES=IN+1
      DO 70 IRUN=1,NRUNS
      WRITE(IOUT,9001) IRUN
      K=0
      N=1
      U(1)=0.
      U(2)=0.
      DO 10 I=1,50
   10 L(I)=0.
      DO 20 I=1,6
   20 MARK(I)=0
      READ (IN,6000)
      READ (INMES,8000)
C9001 WRITE(IOUTP,9007)

C        READ MEASUREMENT DATA

C9002 WRITE(IOUT,9000)
   40 READ(IN,8001) BUOY,HOUR,MIN,SEC,Z,LAT,LONG,ALT,H,SIGMA,PHI,ANT
C9013 WRITE(IOUT,9011)BUOY,HOUR,MIN,SEC,Z,LAT,LONG,ALT,H,SIGMA,PHI,ANT
      IF (BUOY .GE. 9) GO TO 60



C        RETRIEVE THE DATA
```

```
      IF(MARK(BUOY).EQ.0) GO TO 50
      CALL INITAL(BUOY,U,ALT)
   50 CALL DATA(BUOY,S,X,U,T,K,MARK(BUOY))
C
C         PREPARE FOR FILTER
C
      CALL LATLON(LAT,LONG,XAI)
      CALL NOISE(DSEED,Z,.026)
      CALL BASE(ANT,U,GAMMA,PHI,RB1,RB2,RB3)
      TIME=3600*HOUR+60*MIN+SEC
      DELT=FLOAT(TIME-T)/3600.
C
C         USE FILTER AND STORE NEW DATA
C
 9800 WRITE(IOUT,9002) BUOY,K,HOUR,MIN,SEC
      CALL FILTER(S,P,X,U,Z,DELT,XAI,ALT,RB1,RB2,RB3,BUOY,K,M)
      CALL DATA(BUOY,S,X,U,TIME,K,MARK(BUOY))
      CALL RESULT(BUOY,P,X,U,XAI,P,TU,FUIS)
      GO TO 40
C
C         CONTINUE SIMULATION
C
   60 REWIND IN
   70 REWIND INFES
 9701 CALL RESULT(9,P,X,U,XAI,P,IN,FUIS)
      IEND=60400
      WRITE(IOUTP,9006) IEND
C
 6100 FORMAT(////)
 6101 FORMAT(I1,3I2,F7.3,11X,2F9.5,F6.0,F6.1,F4.1,F5.1,13X,I2)
 6102 FORMAT(I3)
 9000 FORMAT(1H1)
 9001 FORMAT(I2,3I2,F7.3,11X,2F9.5,F6.0,F6.1,F4.1,F5.1,13X,I2)
 9002 FORMAT(' BUOY ',I1,' ',I2,'   K=',I3,'      TIME ',I3,2(':',I2))
 9003 FORMAT(' NUM=',I3)
 9004 FORMAT(I8)
 9005 FORMAT(/,' Z-STATE: 9/DD, TT.TT, 3000FT, FO=10%',/,
     ' Z-STATE: 15MN CIRCLE, RS (A/C 000-1 AND SCHOOL), F 270-1)')
 9006 FORMAT(' TIME=',F6.2)
 9007 FORMAT(' BUOY  K',4X,'XB1',6X,'XB2',6X,'EXB1',5X,'EXB2',
     4X,'DELX1',3X,'DELX2',3X,'SIGMA1',3X,'SIGMA2')
C
      STOP
      END
```

```
C                   ************************************************
C                   *       SUBROUTINES FIRST LEVEL                *
C                   ************************************************

      SUBROUTINE LATLON(LAT,LONG,XAT)

      DIMENSION XAT(2)
      REAL LAT,LONG,LATI,LONGI
      DATA PI,A1,A2,A3,A4/3.1415926,2.9035548E-04,9.7532048E-07,
     *     1.0067682,6.7681702E-13/
      LATI=30.
      LONGI=70.
      RP=1./(A1-A2*(SIN(LAT))**2)
      RM=RP/(A3-A4*(SIN(LAT))**2)
      SKL=RP*PI/180.
      CKL=RM*PI/180.
      XAT(1)=(LAT-LATI)*SKL
      XAT(2)=(LONG-LONGI)*CKL
      RETURN
      END



      SUBROUTINE BASE(ANT,H,GAMMA,PHI,RB1,RB2,RB3)

      INTEGER ANT
      RAD=3.1415926/180.
      GR=GAMMA*RAD
      PR=PHI*RAD
      HR=H*RAD
      IF(ANT.EQ.1) GO TO 10
      RB1=-COS(PR)*SIN(HR)
      RB2=COS(PR)*COS(HR)
      RB3=SIN(PR)
      GO TO 20
   10 RB1=COS(GR)*COS(HR)
      RB2=COS(GR)*SIN(HR)
      RB3=-SIN(GR)
   20 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INITAL(BUOY,U,ALT)

      COMMON B(50)
      DIMENSION XB1I(6),XB2I(6),PTIME(6),U(2)
      INTEGER BUOY,K
      DATA PI/3.1415926/
      K=6*BUOY-5

C        INITIAL BUOY DATA

      DATA XB1I/-3.,-5.,-10.,-15.,-24.,0./
      DATA XB2I/-3.5,-5.5,-10.5,-15.5,0.,0./
      DATA PTIME/0.,140.,280.,420.,3500.,0./

C        INITIAL BUOY VELOCITY

      B(1)=U(1)
      B(2)=U(2)

C        INITIAL VARIANCES

      B(K)=ALT/6076.115
      B(K+3)=ALT/6076.115
      B(K+1)=2.
      B(K+2)=2.

C        INITIAL BUOY POSITION

      B(K+4)=XB1I(BUOY)
      B(K+5)=XB2I(BUOY)

C        DROP TIME

      B(K+6)=PTIME(BUOY)
      RETURN
      END
```

```
      SUBROUTINE DATA(BODY,S,X,U,T,K,MARK)

      COMMON D(50)
      DIMENSION S(2,2),X(2),U(2)
      INTEGER BODY,T
      N=8*BODY-5
      IF (MARK .GE. 2) GO TO 2

C        RETRIEVE

    1 U(1)=D(1)
      U(2)=D(2)
      DO 20 I=1,2
      DO 10 J=1,2
      S(I,J)=D(N)
   10 N=N+1
   20 CONTINUE
      X(1)=D(N)
      X(2)=D(N+1)
      T=IFIX(D(N+2))
      K=IFIX(D(N+3)) + 1
      MARK=2
      GO TO 50

C        STORE

    2 D(1)=U(1)
      D(2)=U(2)
      DO 40 I=1,2
      DO 30 J=1,2
      D(N)=S(I,J)
   30 N=N+1
   40 CONTINUE
      D(N)=X(1)
      D(N+1)=X(2)
      D(N+2)=FLOAT(T)
      D(N+3)=FLOAT(K)
      MARK=1
   50 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FILTER(S,P,X,U,Z,DELT,XAI,ALT,RB1,RB2,RB3,BODY,K,N)     EXT02290
                                                                          EXT02300
      DIMENSION P(2,2),C(2),X(2),U(2),HT(2),XNEXT(2),GDZ(2),SNEXT(2,2),  EXT02310
     *  H(1,2),Q(2,2),I(2,2),S(2,2),XAT(2),                              EXT02320
     *  F(2),FT(1,2),FFT(2,2),B(2,2),BETA(2,2),IBETA(2,2),A(2,2),        EXT02330
     *  ST(2,2),SST(2,2),SF(2),ZH(2),ESST(2,2)                           EXT02340
      INTEGER BODY                                                        EXT02350
      REAL I,IBETA                                                        EXT02360
      DATA Q/4*0./                                                        EXT02370
      DATA I/1.,0.,0.,1./                                                 EXT02380
      DATA EPSLN/1./                                                      EXT02390
      IOUT=6                                                              EXT02400
                                                                          EXT02410
      Q(1,1)=.001                                                         EXT02420
      Q(2,2)=.001                                                         EXT02430
      U1=U(1)                                                             EXT02440
      U2=U(2)                                                             EXT02450
      X1=X(1)                                                             EXT02460
      X2=X(2)                                                             EXT02470
                                                                          EXT02480
C     PROPAGATE                                                           EXT02490
C        ** X(K/K-1)**                                                    EXT02500
                                                                          EXT02510
      XNEXT(1)=X(1)+U(1)*DELT                                             EXT02520
      XNEXT(2)=X(2)+U(2)*DELT                                             EXT02530
                                                                          EXT02540
C        ** S(K/K-1)**                                                    EXT02550
                                                                          EXT02560
   20 CALL GMTRA(S,ST,2,2)                                                EXT02570
      CALL GMPRD(S,ST,SST,2,2,2)                                          EXT02580
      CALL GMSCR(SST,EPSLN,ESST,2,2)                                      EXT02590
      CALL GMADD(ESST,Q,B,2,2)                                            EXT02600
      CALL GMSFT(B,SNEXT,2)                                               EXT02610
                                                                          EXT02620
   30 DELX1=XNEXT(1)-XAT(1)                                               EXT02630
      DELX2=XNEXT(2)-XAT(2)                                               EXT02640
      DELALT=ALT/6076.115                                                 EXT02650
      RANGE2=DELX1**2+DELX2**2+DELALT**2                                  EXT02660
      RANGE=SQRT(RANGE2)                                                  EXT02670
      RBX=RB1*DELX1+RB2*DELX2+RB3*DELALT                                  EXT02680
      H(1,1)=(RB1-DELX1*RBX/RANGE2)/RANGE                                 EXT02690
      H(1,2)=(RB2-DELX2*RBX/RANGE2)/RANGE                                 EXT02700
      R=.0067                                                             EXT02710
                                                                          EXT02720
                                                                          EXT02730
                                                                          EXT02740
                                                                          EXT02750
                                                                          EXT02760
```

```
C     UPDATE                                                                                             EXT02770
C          **G(K)**                                                                                      EXT02780
                                                                                                         EXT02790
   50 CALL GMTRA(H,HT,1,2)                                                                               EXT02800
      CALL GMTRA(SNEXT,ST,2,2)                                                                           EXT02810
      CALL GMPRD(ST,HT,F,2,2,1)                                                                          EXT02820
      CALL GMPRD(SNEXT,F,SF,2,2,1)                                                                       EXT02830
      CALL GMTRA(F,FT,2,1)                                                                               EXT02840
      CALL GMPRD(FT,F,FTF,1,2,1)                                                                         EXT02850
      ALPHAI=1/(FTF+FTI)                                                                                 EXT02860
      CALL GMSPR(SF,ALPHAI,G,2,1)                                                                        EXT02870
                                                                                                         EXT02880
C          **X(K/K)**                                                                                    EXT02890
                                                                                                         EXT02900
   60 DZ=Z-EBX/PANGL                                                                                     EXT02910
      CALL GMPRD(G,DZ,GDZ,2,1,1)                                                                         EXT02920
      CALL GMADD(XNEXT,GDZ,X,2,1)                                                                        EXT02930
C9814 WRITE(IOUT,901) DZ                                                                                 EXT02940
C9815 WRITE(IOUT,941) (XNEXT(L),(P(L,J),J=1,2),G(L),GDZ(L),H(1,L),L=1,2)                                 EXT02950
                                                                                                         EXT02960
C          **S(K/K)**                                                                                    EXT02970
                                                                                                         EXT02980
   70 CALL GMPRD(F,FT,FFT,2,1,2)                                                                         EXT02990
      CALL GMSPR(FFT,ALPHAI,FFTA,2,2)                                                                    EXT03000
      CALL GMSUB(I,FFTA,IFFTA,2,2)                                                                       EXT03010
      CALL GMSRT(IFFTA,A,2)                                                                              EXT03020
      CALL GMPRD(SNEXT,A,S,2,2,2)                                                                        EXT03030
      CALL GMTRA(S,ST,2,2)                                                                               EXT03040
      CALL GMPRD(S,ST,P,2,2,2)                                                                           EXT03050
                                                                                                         EXT03060
C     COMPUTATION OF EDDY VELOCITIES (DRIFT)                                                             EXT03070
                                                                                                         EXT03080
   80 ZU(1)=(X(1)-X1)/DELT                                                                               EXT03090
      IF (ZU(1).GT.20.) ZU(1)=10.                                                                        EXT03100
      IF (ZU(1).LT.-20.) ZU(1)=-20.                                                                      EXT03110
      ZU(2)=(X(2)-X2)/DELT                                                                               EXT03120
      IF (ZU(2).GT.20.) ZU(2)=10.                                                                        EXT03130
      IF (ZU(2).LT.-20.) ZU(2)=-20.                                                                      EXT03140
      CA=1./(1.+EP)                                                                                      EXT03150
      C=DELT*CA./84.                                                                                     EXT03160
      IF (C.GT.1.) C=1.                                                                                  EXT03170
      IF (C.LT.0.5) C=0.                                                                                 EXT03180
C9810 WRITE(IOUT,908) C                                                                                  EXT03190
      U(1)=U1+C*(ZU(1)-U1)                                                                               EXT03200
      U(2)=U2+C*(ZU(2)-U2)                                                                               EXT03210
C9804 WRITE(IOUT,903)                                                                                    EXT03220
C9805 WRITE(IOUT,942) (X(L),(P(L,J),J=1,2),ZU(L),L=1,2)                                                  EXT03230
        CURT=SQRT(U(1)**2+U(2)**2)*.025.3//60.                                                           EXT03240
```

```
      IF(U(2).NE.0.) GO TO 91
      IF(U(1).GE.0.) ANGLE=0.
      IF(U(1).LT.0.) ANGLE=180.
      GO TO 92
   91 ANGLE=ATAN(U(1)/U(2))*180./3.1415926
      IF(U(2).GE.0.) ANGLE=90.-ANGLE
      IF(U(2).LT.0.) ANGLE=270.-ANGLE
   92 SIGMA1=S(1,1)*2025.37
      SIGMA2=S(2,2)*2025.37
C980C WRITE(IOUT,904) DRIFT,ANGLE,SIGMA1,SIGMA2
      N=N+1


  904 FORMAT(/,'    DRIFT:',F10.1,' YDS/MIN AT ',F4.0,' DEG    ',
     1    'SIGMA1=',F7.1,' YDS    SIGMA2=',F7.1,' YDS',//)
  941 FORMAT(/,2('  ',F10.5,15X,2F10.5,15X,F10.5,15X,2F10.5,/))
  942 FORMAT(/,2('  ',F10.5,15X,2F10.5,15X,F10.5,/))
  901 FORMAT(/,'    X(K/K-1)-MATRIX:',9X,'P(K/K-1)-MATRIX:',10X,'G(K)-MAT
     *RIX:',17X,'DX',3X,'H',8X,'DZ=',F7.5)
  902 FORMAT(/,'    X(K/K)-MATRIX:',9X,'P(K/K)-MATRIX:',21X,'BUOY VELOCIT
     *Y:')
  908 FORMAT(/,F20.10)
      RETURN
      END



      SUBROUTINE RESULT(BUOY,K,X,U,XAI,P,IN,BUOS)

      DIMENSION X(2),U(2),P(2,2),XAI(2),XAT(2),SUMP1(625),
     *SUMX1(625),SUMX2(625),SUMSQ1(625),SUMSQ2(625),SUMP2(625)
      REAL IBEAT1,IBEAT2
      INTEGER BUOY
      DATA SUMX1,SUMX2,SUMSQ1,SUMSQ2 /625*0.,625*0.,625*0.,625*0./
      DATA SUMP1,SUMP2 /625*0.,625*0./
      INBUS=IN+1
      IOUTP=10
      IF(BUOY.EQ.9) GO TO 30

   10 READ(INBUS,*) XB1,XB2,XAT
      DX1=X(1)-XB1
      DX2=X(2)-XB2
      DELXA1=X(1)-XAT(1)-XB1+XAT(1)
      DELXA2=X(2)-XAT(2)-XB2+XAT(2)
      IF(IBUOY.NE.1) GO TO 20
      SUMX1(K)=SUMX1(K)+DELX1
```

```
      SUMX2(K)=SUMX2(K)+FDELX2                                          EXT03730
      SUMSQ1(K)=SUMSQ1(K)+FDELX1**2                                     EXT03740
      SUMSQ2(K)=SUMSQ2(K)+FDELX2**2                                     EXT03750
      SUMP1(K)=SUMP1(K)+P(1,1)                                          EXT03760
      SUMP2(K)=SUMP2(K)+P(2,2)                                          EXT03770
   20 CONTINUE                                                          EXT03780
      S1=SQRT(P(1,1))                                                   EXT03790
      S2=SQRT(P(2,2))                                                   EXT03800
 9607 WRITE(IOUTP,8) BUOY,K,X,FX1,FX2,FDELX1,FDELX2,S1,S2               EXT03810
      RETURN                                                            EXT03820
                                                                        EXT03830
   30 READ(INREFS,6)                                                    EXT03840
 9702 WRITE(IOUTP,4)                                                    EXT03850
      DO 50 I=1,K                                                       EXT03860
      READ(INREFS,9) XB1,XB2,XAT                                        EXT03870
      MEAN1=SUMX1(I)/RUNS                                               EXT03880
      MEAN2=SUMX2(I)/RUNS                                               EXT03890
      VARX1=SUMSQ1(I)/RUNS-MEAN1**2                                     EXT03900
      VARX2=SUMSQ2(I)/RUNS-MEAN2**2                                     EXT03910
      VAR1=SUMSQ1(I)/RUNS                                               EXT03920
      VAR2=SUMSQ2(I)/RUNS                                               EXT03930
      SIGMAX=SQRT(VARX1+VARX2)                                          EXT03940
      SIGMA=SQRT(VAR1+VAR2)                                             EXT03950
      SIGMAF=SQRT(SUMP1(I)/RUNS+SUMP2(I)/RUNS)                          EXT03960
 9705 WRITE(IOUTP,5) I,SIGMAX,SIGMA,SIGMAF                              EXT03970
   50 CONTINUE                                                          EXT03980
                                                                        EXT03990
    4 FORMAT('          K',4X,'SIGMAX',4X,'SIGMA',5X,'SIGMAF')          EXT04000
    5 FORMAT(I8,3F10.5)                                                 EXT04010
    7 FORMAT(///)                                                       EXT04020
    8 FORMAT(2I4,3F9.5)                                                 EXT04030
    9 FORMAT(3X,4F9.4)                                                  EXT04040
      STOP                                                              EXT04050
      END                                                               EXT04060
                                                                        EXT04070
                                                                        EXT04080
                                                                        EXT04090
      SUBROUTINE NOISE(OSEED,Z,STD)                                     EXT04100
                                                                        EXT04110
      DOUBLE PRECISION OSEED                                            EXT04120
    1 IDEZ=5                                                            EXT04130
      ZN=GGNQF(OSEED)                                                   EXT04140
      Z1=ZN*STD+Z                                                       EXT04150
 9613 WRITE(IOUTZ,9610) Z,Z1,ZN                                         EXT04160
      Z=Z1                                                              EXT04170
 9610 FORMAT(3F20.8)                                                    EXT04180
      RETURN                                                            EXT04190
      END                                                               EXT04200
```

```
C             ************************************                      EXT04210
C             *     SUBROUTINES SECOND LEVEL     *                      EXT04220
C             ************************************                      EXT04230
                                                                        EXT04240
                                                                        EXT04250
                                                                        EXT04260
      SUBROUTINE GMTRA(A,R,N,M)                                         EXT04270
                                                                        EXT04280
      DIMENSION A(N,M), R(M,N)                                          EXT04290
      DO 10 I=1,N                                                       EXT04300
      DO 20 J=1,M                                                       EXT04310
  20  R(J,I)=A(I,J)                                                     EXT04320
  10  CONTINUE                                                          EXT04330
      RETURN                                                            EXT04340
      END                                                               EXT04350
                                                                        EXT04360
                                                                        EXT04370
      SUBROUTINE GMADD(A,B,R,N,M)                                       EXT04380
                                                                        EXT04390
      DIMENSION A(1),B(1),R(1)                                          EXT04400
      NM=N*M                                                            EXT04410
      DO 10 I=1,NM                                                      EXT04420
  10  R(I)=A(I)+B(I)                                                    EXT04430
      RETURN                                                            EXT04440
      END                                                               EXT04450
                                                                        EXT04460
                                                                        EXT04470
      SUBROUTINE GMSUB(A,B,R,N,M)                                       EXT04480
                                                                        EXT04490
      DIMENSION A(1),B(1),R(1)                                          EXT04500
      NM=N*M                                                            EXT04510
      DO 10 I=1,NM                                                      EXT04520
  10  R(I)=A(I)-B(I)                                                    EXT04530
      RETURN                                                            EXT04540
      END                                                               EXT04550
                                                                        EXT04560
                                                                        EXT04570
      SUBROUTINE GMPRD(A,B,R,N,M,L)                                     EXT04580
                                                                        EXT04590
      DIMENSION A(1),B(1),R(1)                                          EXT04600
      IR=0                                                              EXT04610
      IK=-M                                                             EXT04620
      DO 10 K=1,L                                                       EXT04630
      IK=IK+M                                                           EXT04640
      DO 10 J=1,N                                                       EXT04650
      IR=IR+1                                                           EXT04660
      JI=J-N                                                            EXT04670
      IB=IK                                                             EXT04680
```

```
      K(IK)=0                                                           EXT04690
      DO 10 I=1,M                                                       EXT04700
      JI=JI+N                                                           EXT04710
      IB=IB+1                                                           EXT04720
   10 K(IK)=K(IK)+A(JI)*B(IB)                                           EXT04730
      RETURN                                                            EXT04740
      END                                                               EXT04750
                                                                        EXT04760
                                                                        EXT04770
      SUBROUTINE GMSCP(A,S,AS,N,M)                                      EXT04780
                                                                        EXT04790
      DIMENSION A(N,M),AS(N,M)                                          EXT04800
      DO 10 I=1,N                                                       EXT04810
      DO 20 J=1,M                                                       EXT04820
   20 AS(I,J)=A(I,J)*S                                                  EXT04830
   10 CONTINUE                                                          EXT04840
      RETURN                                                            EXT04850
      END                                                               EXT04860
                                                                        EXT04870
                                                                        EXT04880
      SUBROUTINE GMSRT(B,A,N)                                           EXT04890
                                                                        EXT04900
      DIMENSION B(N,N),A(N,N)                                           EXT04910
      DO 8 JJ=1,N                                                       EXT04920
      J=N+1-JJ                                                          EXT04930
      JP1=J+1                                                           EXT04940
      DO 7 II=1,N                                                       EXT04950
      I=N+1-II                                                          EXT04960
      IF(I.GT.J) GO TO 6                                                EXT04970
      IF(I.EQ.J) GO TO 3                                                EXT04980
      AA=0.                                                             EXT04990
      IF(JP1.GT.N) GO TO 2                                              EXT05000
      DO 1 L=JP1,N                                                      EXT05010
    1 AA=AA+A(I,L)*A(J,L)                                               EXT05020
    2 A(I,J)=(B(I,J)-AA)/A(J,J)                                         EXT05030
      GO TO 7                                                           EXT05040
      AIK2=0.                                                           EXT05050
      IF(JP1.GT.N) GO TO 5                                              EXT05060
      DO 4 K=JP1,N                                                      EXT05070
    4 AIK2=AIK2+A(I,K)**2                                               EXT05080
    5 A(I,I)=SQRT(B(I,I)-AIK2)                                          EXT05090
      GO TO 7                                                           EXT05100
    6 A(I,J)=0.                                                         EXT05110
    7 CONTINUE                                                          EXT05120
    8 CONTINUE                                                          EXT05130
      RETURN                                                            EXT05140
      END                                                               EXT05150
                                                                        EXT05160
```

```
C     SETUP:                                                             EXT05170
C       C96   ICUTE FOR SERIES                                           EXT05180
C        97   ICUTE FOR SPLIT                                            EXT05190
C       C98   PRINTS                                                     EXT05200
C       IN=7                                                             EXT05210
C       NRUNS=20                                                         EXT05220
C       DATA XB11...-3.,-5.,....                                         EXT05230
C       DATA XB21...-3.5,-5.5,...                                        EXT05240
```

APPENDIX B
COMPUTER PROGRAM FOR DATA GENERATION

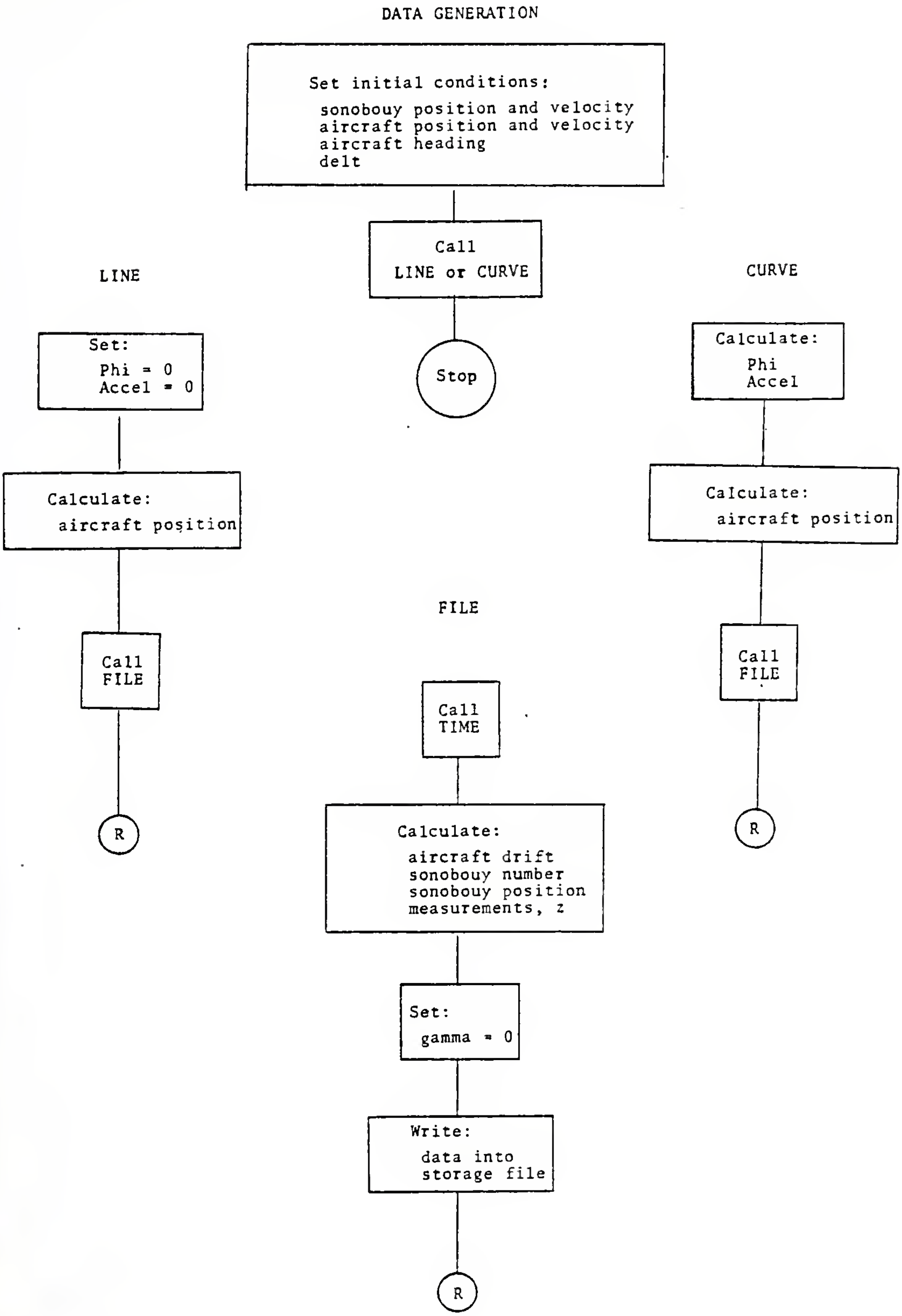


Figure 37. Flow chart: data generation program.

GLOSSARY OF COMPUTER VARIABLES

DATA GENERATION PROGRAM

A	Vector of aircraft accelerations
AL	Altitude in nautical miles
ALT	Altitude
ANT	Antenna
AVEL	Initial aircraft velocity
AX1	Amplitude of Schuler cycle, A_{x_1}
AX2	Amplitude of Schuler cycle, A_{x_2}
B	Intercept on N-S axis of straight track
BANG	Sonobuoy drift direction
BOUY	Sonobuoy RF channel or number
BVEL	Sonobuoy drift direction
CKL	Rate of change of nautical miles for a change in degrees of longitude
D	Length of straight track
DELT	Interval between measurements, Δt
DELV1	Relative velocity, Δv_1
DELV2	Relative velocity, Δv_2
DELX1	Relative position, Δx_1
DELX2	Relative position, Δx_2
DH	Total change in heading while on curved track
DSEED	Seed for the random number generator
END	Indicator of EOF
G	Gravitational acceleration
GAMMA	Aircraft pitch angle, γ

GR	Gamma in radians
H	Aircraft heading
HNEW	Final heading on curved track
HOUR	Time in hours
HR	Heading in radians
IOUT	Device on which print is made
IOUTP	File in which results are written
IOUTR	File in which true sonobuoy positions are written
IOUTZ	File in which random vectors are inspected
ITIME	Time from aircraft clock in seconds
KX1	Rate of linear navigational drift, K_{x1}
KX2	Rate of linear navigational drift, K_{x2}
LAT	Latitude
LATI	The latitude of the origin of the local earth fixed coordinate system
LONG	Longitude
LONGI	The longitude of the origin of the local earth fixed coordinate system
LTIME	Array of initial drop times for the sonobuoy
M	Slope of straight track
MARK	Indicates the first time a buoy is to be proccessed
MIN	Time in minutes
N	Index
NBOUY	Total number of sonobuoys
NSTOP	End of line or curve series
PHI	Aircraft roll angle, ϕ
PI	π
PI2	2π

PR	Phi in radians
R	Radius of curved track
RB1	Unit vector component of antenna baseline
RB2	Unit vector component of antenna baseline
RB3	Unit vector component of antenna baseline
S	Distance aircraft travels in time Δt
SEC	Time in seconds
SKL	Rate of change of nautical miles for a change in degrees of latitude
TRACK	Direction of aircraft travel over the ground
U	Direction to center of curved track
V	Direction from center of curved track to aircraft
VA1	Velocity of aircraft
VA2	Velocity of aircraft
VB1	Sonobuoy drift
VB2	Sonobuoy drift
WANG	Wind direction
WVEL	Wind velocity
XAI	Aircraft inertial position, X_a^I
XAT	Aircraft true position, X_a
XB1	True geographical location of sonobuoy, x_1
XB2	True geographical location of sonobuoy, x_2
XC1	Location of the center of the curved track
XC2	Location of the center of the curved track
XX1	Incremental change in X_1
XX2	Incremental change in X_2
Z	Measurement vector

```
                                                      CRI00520
4     CALL  CURVEL(XAT,3.5,H,000.,S,MARK)             CRI00530
                                                      CRI00540
5     CALL  LINE(XAT,H,21.,S,MARK)                    CRI00550
                                                      CRI00560
6     CALL  CURVEL(XAT,3.5,H,270.,S,MARK)             CRI00570
                                                      CRI00580
7     CALL  LINE(XAT,H,21.,S,MARK)                    CRI00590
                                                      CRI00600
8     CALL  CURVEL(XAT,3.5,H,180.,S,MARK)             CRI00610
                                                      CRI00620
9     CALL  LINE(XAT,H,22.,S,MARK)                    CRI00630
                                                      CRI00640
10    CALL  CURVEL(XAT,3.5,H,090.,S,MARK)             CRI00650
                                                      CRI00660
11    CALL  LINE(XAT,H,22.,S,MARK)                    CRI00670
                                                      CRI00680
12    CALL  CURVEL(XAT,3.5,H,0.,S,MARK)               CRI00690
                                                      CRI00700
13    CALL  LINE(XAT,H,23.,S,MARK)                    CRI00710
                                                      CRI00720
14    CALL  CURVEL(XAT,3.5,H,270.,S,MARK)             CRI00730
                                                      CRI00740
15    CALL  LINE(XAT,H,23.,S,MARK)                    CRI00750
                                                      CRI00760
16    CALL  CURVEL(XAT,3.5,H,180.,S,MARK)             CRI00770
                                                      CRI00780
17    CALL  LINE(XAT,H,24.,S,MARK)                    CRI00790
                                                      CRI00800
18    CALL  CURVEL(XAT,3.5,H,090.,S,MARK)             CRI00810
                                                      CRI00820
19    CALL  LINE(XAT,H,24.,S,MARK)                    CRI00830
                                                      CRI00840
20    CALL  CURVEL(XAT,3.5,H,000.,S,MARK)             CRI00850
                                                      CRI00860
21    CALL  LINE(XAT,H,25.,S,MARK)                    CRI00870
                                                      CRI00880
22    CALL  CURVEL(XAT,3.5,H,270.,S,MARK)             CRI00890
                                                      CRI00900
23    CALL  LINE(XAT,H,25.,S,MARK)                    CRI00910
                                                      CRI00920
24    CALL  CURVEL(XAT,3.5,H,180.,S,MARK)             CRI00930
                                                      CRI00940
25    CALL  LINE(XAT,H,26.,S,MARK)                    CRI00950
                                                      CRI00960
26    CALL  CURVEL(XAT,3.5,H,090.,S,MARK)             CRI00970
                                                      CRI00980
27    CALL  LINE(XAT,H,26.,S,MARK)                    CRI00990
```

```
      END=99999.9
      IEND=99
      WRITE(IOUT,9001) IEND
      WRITE(IOUTP,9000) END
9000  FORMAT(10X,F10.4)
9001  FORMAT(I2)

      STOP
      END


      SUBROUTINE LINE(XAT,H,D,S,MARK)

      DIMENSION XAT(2),A(2)
      REAL M
      NSTOP=IFIX(D/S)
      PHI=0.
      A(1)=0.
      A(2)=0.
      IF(H.EQ.0.) GO TO 20
      IF(H.EQ.180.) GO TO 40
      HR=H*3.1415926/180.
      M=COTAN(HR)
      B=XAT(1)-M*XAT(2)
      XX2=S*SIN(HR)

      DO 10 I=1,NSTOP
      XAT(2)=XAT(2)+XX2
      XAT(1)=M*XAT(2)+B
10    CALL FILE(XAT,H,PHI,A,MARK)
      GO TO 99
20    DO 30 K=1,NSTOP
      XAT(1)=XAT(1)+S
30    CALL FILE(XAT,H,PHI,A,MARK)
      GO TO 99
40    DO 50 N=1,NSTOP
      XAT(1)=XAT(1)-S
50    CALL FILE(XAT,H,PHI,A,MARK)
99    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CURVEL(XAT,R,H,HNEW,S,MARK)

      DIMENSION XAT(2),A(2)
      DATA PI,PI2,G /3.1415926,6.2831852,32.2/
      AVEL=180.
      U=(H-90.)*PI/180.
      XC1=R*COS(U)+XAT(1)
      XC2=R*SIN(U)+XAT(2)
      V=(H+90.)*PI/180.
      DH=H+360.-HNEW
      IF(DH.GT.360.) DH=DH-360.
      NSTOP=IFIX(DH*PI/180.*R/S)
      PHI=-ATAN((AVEL*1.6878)**2/(G*R*6076.115))*180./PI

      DO 10 K=1,NSTOP
      V=V-S/R
      IF(V.GT.PI2) V=V-PI2
      IF(V.LT.0.) V=V+PI2
      A(1)=-COS(V)*AVEL**2/R
      A(2)=-SIN(V)*AVEL**2/R
      H=V*180./PI-90.
      IF(H.LT.0.) H=H+360.
      XAT(1)=R*COS(V)+XC1
      XX2=SQRT(R**2-(XAT(1)-XC1)**2)
      IF(V.LE.PI) XAT(2)=XC2+XX2
      IF(V.GT.PI) XAT(2)=XC2-XX2
   10 CALL FILE(XAT,H,PHI,A,MARK)
      H=HNEW
      RETURN
      END


      SUBROUTINE CURVER(XAT,R,H,HNEW,S,MARK)

      DIMENSION XAT(2),A(2)
      DATA PI,PI2,G /3.1415926,6.2831852,32.2/
      AVEL=180.
      U=(H+90.)*PI/180.
      XC1=R*COS(U)+XAT(1)
      XC2=R*SIN(U)+XAT(2)
      V=(H-90.)*PI/180.
      DH=H+360.-HNEW
      IF(DH.GT.360.) DH=DH-360.
      NSTOP=IFIX(DH*PI/180.*R/S)
      PHI=ATAN((AVEL*1.6878)**2/(G*R*6076.115))*180./PI

      DO 10 K=1,NSTOP
      V=V+S/R
```

```
      IF(V.GT.PI2) V=V-PI2
      IF(V.LT.0.) V=V+PI2
      A(1)=-COS(V)*AVEL**2/R
      A(2)=-SIN(V)*AVEL**2/R
      H=V*180./PI+90.
      IF(H.GE.360.) H=H-360.
      XAT(1)=R*COS(V)+XC1
      XX2=SQRT(R**2-(XAT(1)-XC1)**2)
      IF(V.LT.PI) XAT(2)=XC2+XX2
      IF(V.GT.PI) XAT(2)=XC2-XX2
10    CALL FILE(XAT,H,PHI,A,MARK)
      T=TNEW
      RETURN
      END


      SUBROUTINE FILE(XAT,H,PHI,A,MARK)

      DIMENSION Z(3),XAI(2),XAT(2),A(2)
      REAL LAT,LONG
      INTEGER HOUR,SEC,BUOY,ANT
      DATA PI/3.1415926/
      IOUT=7
      IOUTR=IOUT+1
      IOUTP=10
      AVEL=160.
      WVEL=0.
      WARC=0.
      IF(MARK.NE.1) GO TO 1
      WRITE(IOUT,9094)
      WRITE(IOUTP,9095)
      WRITE(IOUTR,9096)
1     CALL TIME(HOUR,MIN,SEC,MARK)
      ITIME=3600*HOUR+60*MIN+SEC
      CALL ACFNIF(XAT,XAI,ITIME)
      CALL BUOYNO(BUOY,MARK)
2     CALL BUOYPO(BUOY,ITIME,XB1,XB2,VB1,VB2,MARK)
      ANT=1
      ALT=3000.
      AL=ALT/6076.115
      GAMMA=0.
      DELX1=XB1-XAT(1)
      DELX2=XB2-XAT(2)

C     Z(1) - COS(THETA) *

3     CALL  (ANT,H,GAMMA,PHI,EB1,EB2,EB3)
      Z(1)=(DELX1*EB1+DELX2*EB2+AL*EB3)/SQRT(DELX1**2+DELX2**2+AL**2)
```

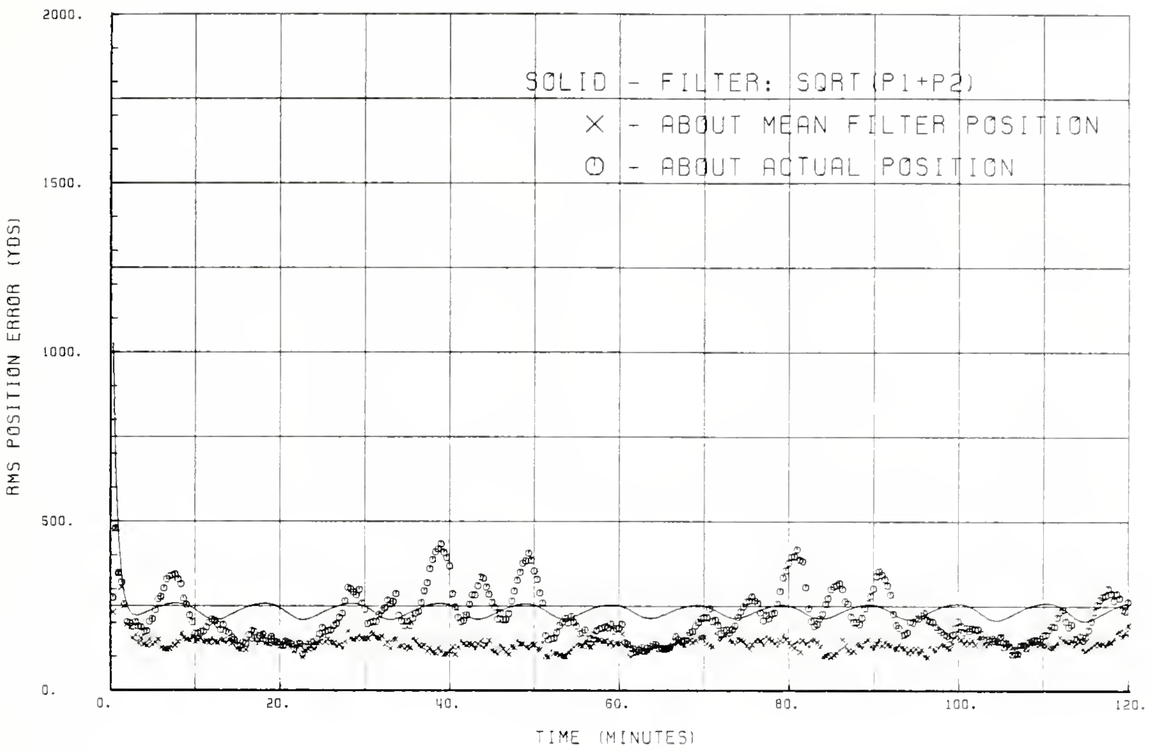
```
      IF(ABS(Z(1)).LE..866) GO TO 20                                   CRI02390
      IF(ANT.EQ.0) GO TO 15                                             CRI02400
      ANT=0                                                             CRI02410
      GO TO 15                                                          CRI02420
   15 WRITE(IOUT,9082)                                                  CRI02430
   20 IF(MARK.EQ.1) DSEED=7654321                                       CRI02440
                                                                        CRI02450
C     ** Z(2) - VELOCITY RELATIVE TO WATER SURFACE **                   CRI02460
                                                                        CRI02470
      VA1=AVEL*COS(H*PI/180.) + WVEL*COS(WANG*PI/180.)                  CRI02480
      VA2=AVEL*SIN(H*PI/180.) + WVEL*SIN(WANG*PI/180.)                  CRI02490
      DELV1=VB1-VA1                                                     CRI02500
      DELV2=VB2-VA2                                                     CRI02510
      Z(2)=SQRT(DELV1**2+DELV2**2)                                      CRI02520
                                                                        CRI02530
C     ** Z(3) - DRIFT ANGLE **                                          CRI02540
                                                                        CRI02550
      IF(DELV2.NE.0.) GO TO 30                                          CRI02560
      IF(DELV1.GE.0.) TRACK=0.                                          CRI02570
      IF(DELV1.LT.0.) TRACK=180.                                        CRI02580
      GO TO 31                                                          CRI02590
   30 TRACK=ATAN(DELV1/DELV2)*180./PI                                   CRI02600
      IF(DELV2.LE.0.) TRACK=90.-TRACK                                   CRI02610
      IF(DELV2.GT.0.) TRACK=270.-TRACK                                  CRI02620
   31 Z(3)=TRACK-H                                                      CRI02630
      IF(Z(3).GT.180.) Z(3)=Z(3)-360.                                   CRI02640
      IF(Z(3).LT.-180.) Z(3)=Z(3)+360.                                  CRI02650
                                                                        CRI02660
      WRITE(IOUTP,9081) MARK,XAI,XAT                                    CRI02670
      WRITE(IOUTP,9083) MARK,XB1,XB2,XAT,DELV1,DELV2,TRACK              CRI02680
      CALL LATLOP(XAI(1),XAI(2),LAT,LONG)                               CRI02690
      WRITE(IOUT,9080)BUOY,HOUR,MIN,SEC,Z,LAT,LONG,ALT,H,GAMMA,PHI,A,ANTCRI02700
      MARK=MARK+1                                                       CRI02710
                                                                        CRI02720
 9080 FORMAT(I1,3I2,F7.3,F6.3,F5.3,2F9.5,F6.0,F6.1,F4.1,F5.1,           CRI02730
     $ 2F7.0,I2)                                                        CRI02740
 9081 FORMAT(I10,4F10.4)                                                CRI02750
 9082 FORMAT(' BAD BEARING')                                            CRI02760
 9083 FORMAT(I3,4F9.4,3F10.4)                                           CRI02770
 9094 FORMAT(/,' DAT'12: RUN 9709, DELT=90',//,'  H  M  S',            CRI02780
     $ 2X,'Z(1)',2X,'Z(2)',2X,'Z(3)',3X,'XAI(1)',3X,'XAI(2)',2X,'ALT',2XCRI02790
     $ ,'BEAR',2X,'GAM',2X,'PHI',3X,'AX1',3X,'AX2',1X,'AN')             CRI02800
 9095 FORMAT(/,' DAT'12: RUN 9709, DELT=90, ',                          CRI02810
     $ //,11X,'XAI(1)',4X,'XAI(2)',4X,'XAT(1)',4X,'XAT(2)')             CRI02820
 9096 FORMAT(/,' DAT'12: RUN 9709, DELT=90, DC 090-2, .5,.5,2.5,0.',    CRI02830
     $ //,6X,'XB1',6X,'XB2',6X,'XAT(1)',3X,'XAT(2)',3X,'DELV1',5X,'DELV2CRI02840
     $',5X,'TRACK')                                                     CRI02850
                                                                        CRI02860
```

```
      RETURN
      END


      SUBROUTINE TIME(H,M,S,MARK)

      INTEGER H,S
      IF(MARK.NE.1) GO TO 10
      S=0
      M=0
      H=0

 10   IDELT=50
      S=S+IDELT
      IF(S.LT.60) GO TO 20
      S=0
      M=M+1
      IF(M.LT.60) GO TO 20
      M=0
      H=H+1

 20   CONTINUE
      RETURN
      END


      SUBROUTINE ACORTE(XAT,XAI,ITIME)

      DIMENSION XAT(2),XAI(2)
      REAL KX1,KX2
      WVEL=0.
      WANG=0.

C     ** WIND **

      XAT(1)=XAT(1)+WVEL*COS(WANG*PI/180.)
      XAT(2)=XAT(2)+WVEL*SIN(WANG*PI/180.)

C     ** INITIAL **

      DATA AX1,AX2,KX1,KX2 /.5,.5,2.5,0./
      XAI(1)=XAT(1)+AX1*SIN(.00124*ITIME)+KX1*ITIME/3600.
      XAI(2)=XAT(2)+AX2*SIN(.00124*ITIME)+KX2*ITIME/3600.
      RETURN
      END
```

```
      SUBROUTINE BUOYNO(BUOY,MARK)

      INTEGER BUOY
C9801 NBUOY=4
C9802 IF(MARK.GT.64) GO TO 10
C9803 NBUOY=3
C9804 IF(MARK.GT.42) GO TO 10
C9805 NBUOY=2
C9806 IF(MARK.GT.21) GO TO 10
      NBUOY=1
   C8 IF(MARK.EQ.1) BY=0
   10 BUOY=MARK+BY
   11 IF (BUOY.LE.NBUOY) GO TO 20
   12 BY=BY-NBUOY
   13 GO TO 10
   20 CONTINUE
      RETURN
      END


      SUBROUTINE BUOYPO(BUOY,ITIME,X1,X2,X3,X4,MARK)

      DIMENSION XB1(31),XB2(31),LTIME(31)
      INTEGER BUOY
      BVEL=2.
      BANG=390.
      IF (MARK.NE.1) GO TO 2
      XB1(1)=-3.
      XB2(1)=-3.
      LTIME(1)=0
      XB1(2)=-5.
      XB2(2)=-5.
      LTIME(2)=630
      XB1(3)=-10.
      XB2(3)=-10.
      LTIME(3)=1260
      XB1(4)=-15.
      XB2(4)=-15.
      LTIME(4)=1920
      VB1=BVEL*COS(BANG*3.1415926/180.)
      VB2=BVEL*SIN(BANG*3.1415926/180.)
      DELT=FLOAT(ITIME-LTIME(BUOY))/3600.
      X1=XB1(BUOY)+DELT*VB1
      X2=XB2(BUOY)+DELT*VB2
      X3=VB1
      X4=VB2
      LTIME(BUOY)=ITIME
      XB1(BUOY)=X1
```

```
              ****************************************
              *          DATA GENERATION             *
              *                FOR                   *
              *          SQUARE PATTERN              *
              ****************************************

C   TO ALTER, CHECK:

C     IOUT=7
C     IOUTP=10
C     IOUTZ=9
C     XAT(1)=-3.
C     XAT(2)=-3.
C     AVEL=180.
C     H=180.
C     DELT=90
C     ALT=3000.
C     GAMMA=0.
C     DSEED=7654221.0D0
C     AX1,AX2,KX1,KX2 IN ACDRIF
C     WVEL=0.
C     WANG=0.
C     BVEL=2.
C     BANG=090.
C     LINE AND CURVE CALLS
C     FOR MULTIPLE BUOYS-ADJUST 'C98' IN BUOYNO
C           AND INITIAL BUOY POSITIONS IN BUOYPO

C       ** WARNING: H CANNOT BE GREATER THAN 360. **


      DIMENSION XAT(2)
      IOUT=7
      IOUTP=10
      IOUTZ=9
      MARK=1
      XAT(1)=-3.
      XAT(2)=-3.
      H=180.
      AVEL=180.
      DELT=90.
      S=AVEL*DELT/3600.

 1    CALL LINE(XAT,H,7.5,S,MARK)

 2    CALL CURVE1(XAT,3.5,H,090.,S,MARK)

 3    CALL LINE(XAT,H,10.,S,MARK)
```

```
      A32(POLY)=X2                                                      CRI03830
      RETURN                                                            CRI03840
      END                                                               CRI03850
                                                                        CRI03860
                                                                        CRI03870
      SUBROUTINE RAS (ANT,H,GAMMA,PHI,RB1,RB2,RB3)                      CRI03880
                                                                        CRI03890
      INTEGER ANT                                                       CRI03900
      RAD=3.1415926/180.                                                CRI03910
      GR=GAMMA*RAD                                                      CRI03920
      PR=PHI*RAD                                                        CRI03930
      HR=H*RAD                                                          CRI03940
      IF(ANT.EQ.1) GO TO 1                                              CRI03950
      RB1=-COS(PR)*SIN(HR)                                              CRI03960
      RB2=COS(PR)*COS(HR)                                               CRI03970
      RB3=SIN(PR)                                                       CRI03980
      GO TO 5                                                           CRI03990
    1 RB1=COS(GR)*COS(HR)                                               CRI04000
      RB2=COS(GR)*SIN(HR)                                               CRI04010
      RB3=-SIN(GR)                                                      CRI04020
    5 CONTINUE                                                          CRI04030
      RETURN                                                            CRI04040
      END                                                               CRI04050
                                                                        CRI04060
                                                                        CRI04070
      SUBROUTINE LATLON(X,Y,LAT,LONG)                                   CRI04080
                                                                        CRI04090
      REAL LAT,LONG,LATI,LONGI                                          CRI04100
      DATA PI,A1,A2,A3,A4/3.1415926,2.9035543E-04,9.7532043E-07,        CRI04110
     *     1.0067682,6.7681702E-13/                                     CRI04120
      LATI=30.                                                          CRI04130
      LONGI=70.                                                         CRI04140
      LAT=LATI+X/60.                                                    CRI04150
      RP=1./(A1-A2*(SIN(LAT))**2)                                       CRI04160
      RM=RP/(A3-A4*(SIN(LAT))**2)                                       CRI04170
      SKL=RP*PI/180.                                                    CRI04180
      CKL=RM*PI/180.                                                    CRI04190
      LAT=LATI+X/SKL                                                    CRI04200
      LONG=LONGI+Y/CKL                                                  CRI04210
      RETURN                                                            CRI04220
      END                                                               CRI04230
```



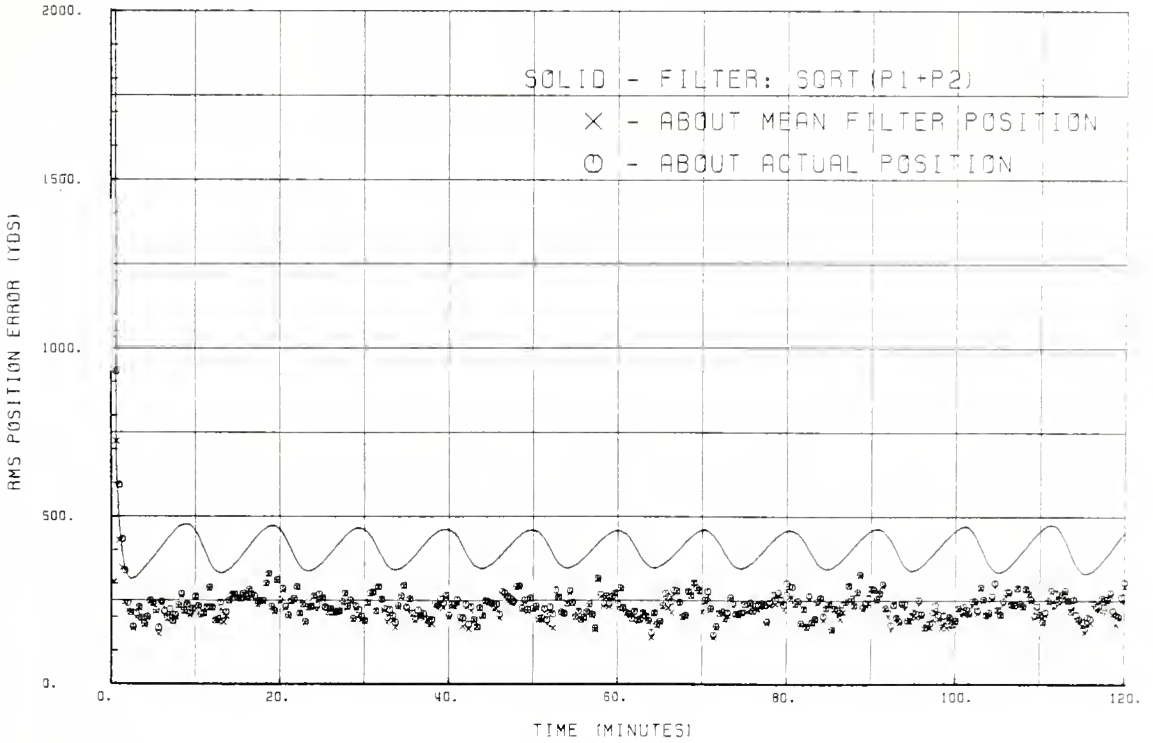


Figure 38. RMS errors for two-state system (top) and six-state system (bottom) using circular pattern at 5 NM.

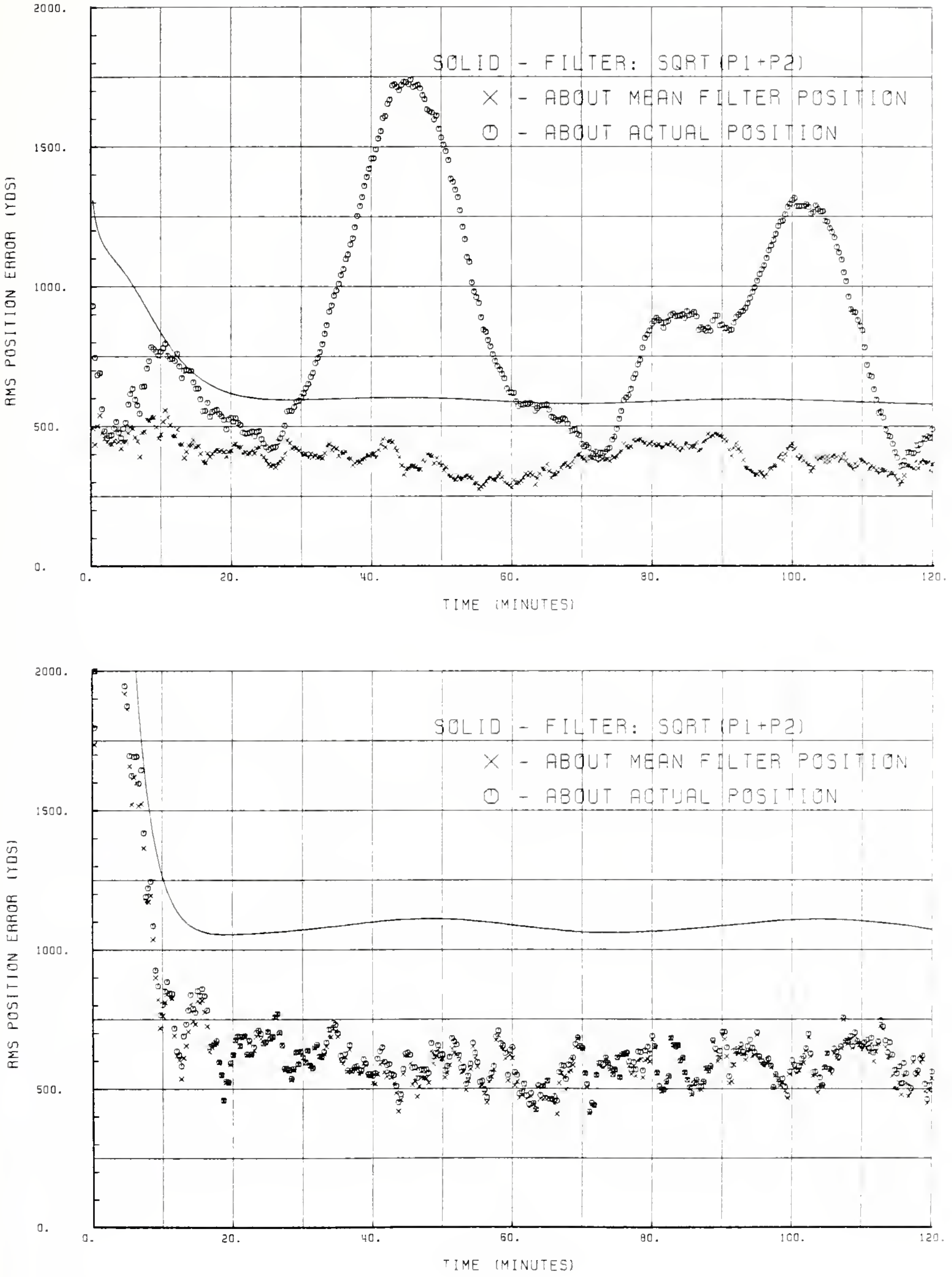


Figure 39. RMS errors for two-state system (top) and six-state system (bottom) using circular pattern at 30 NM.

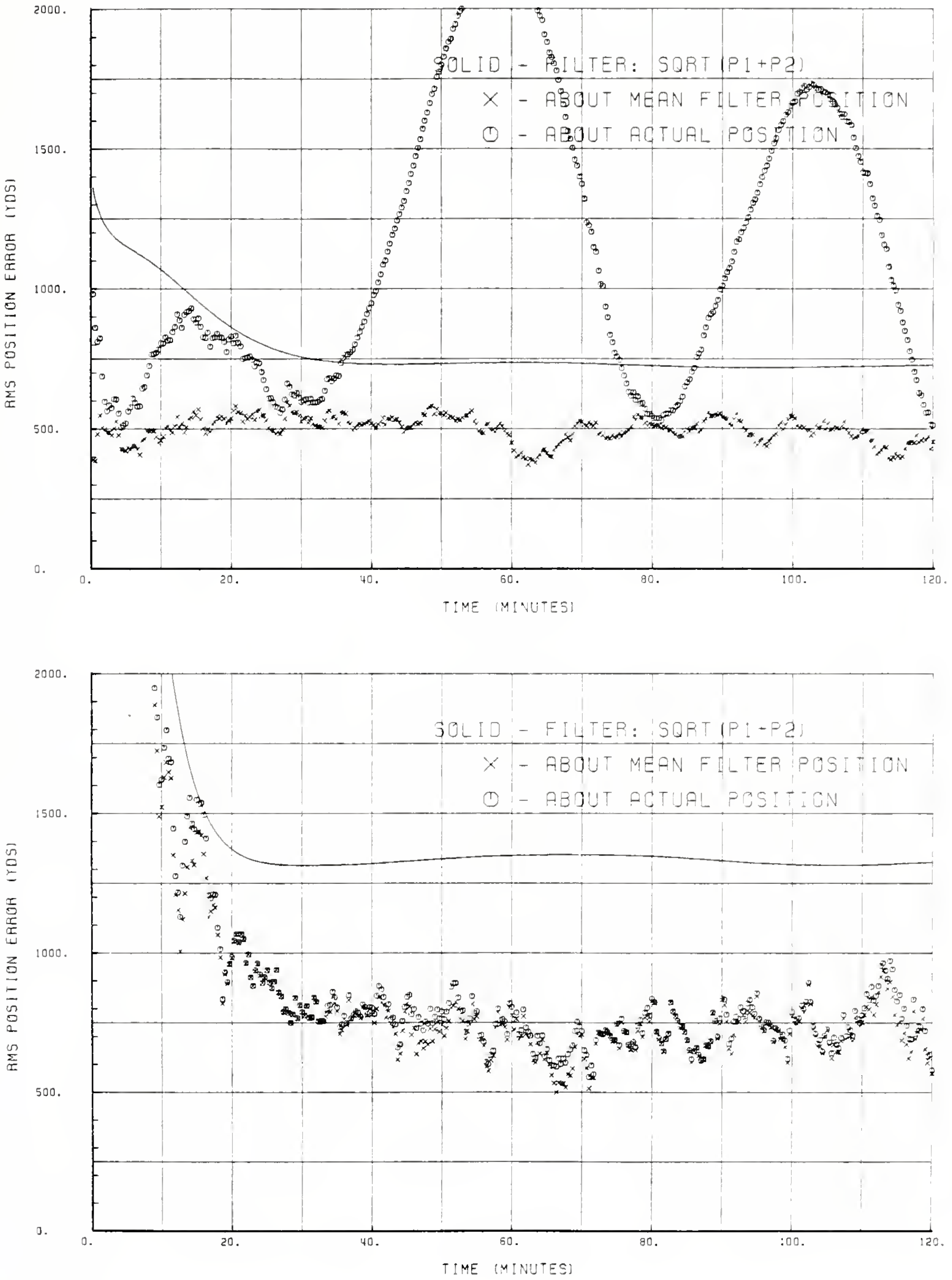
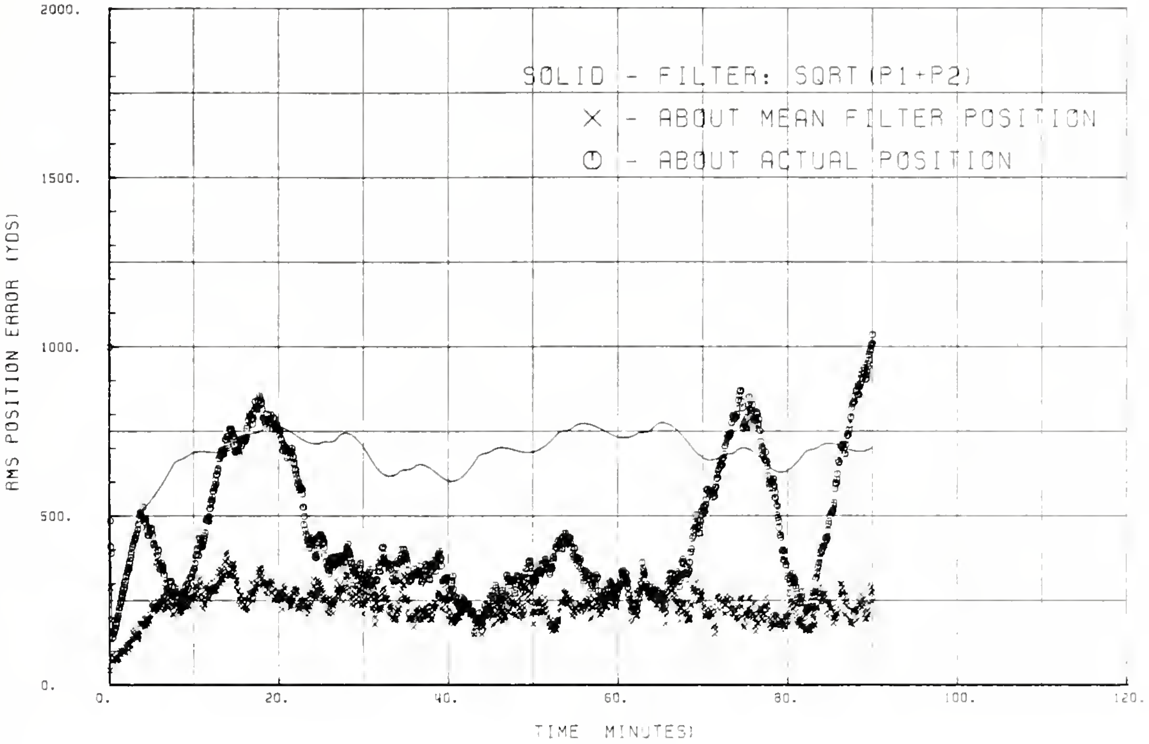


Figure 40. RMS errors for two-state system (top) and six-state system (bottom) using circular pattern at 45 NM.



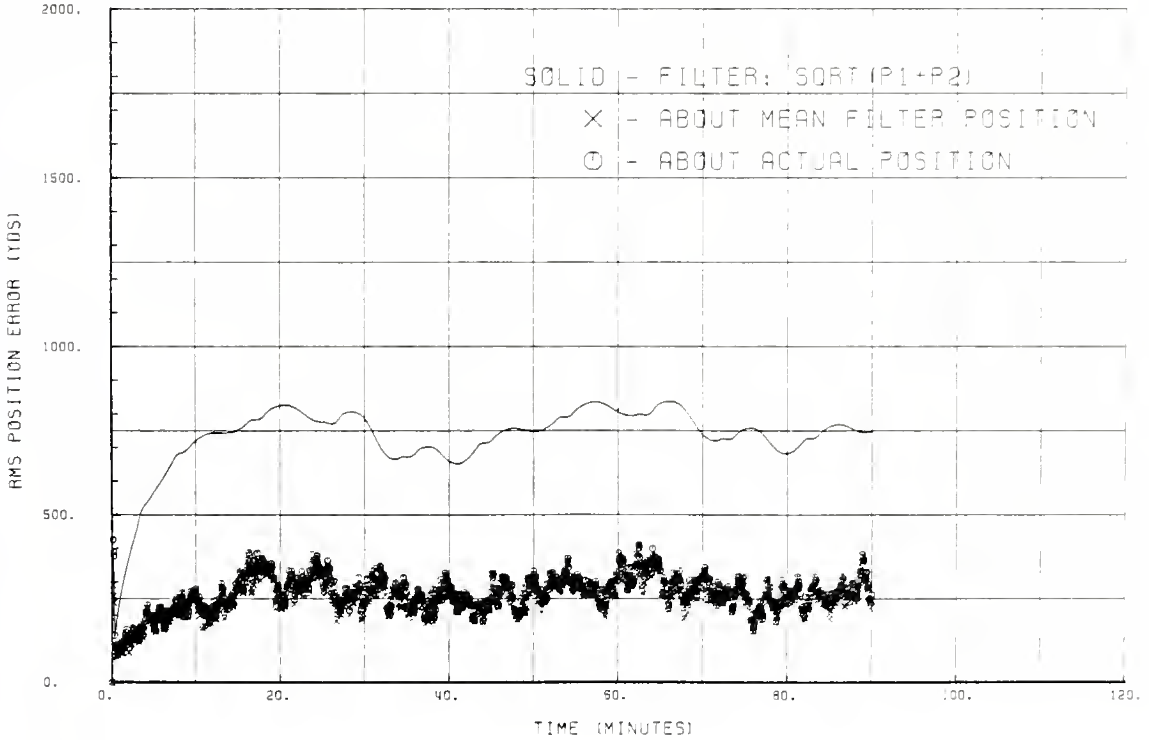
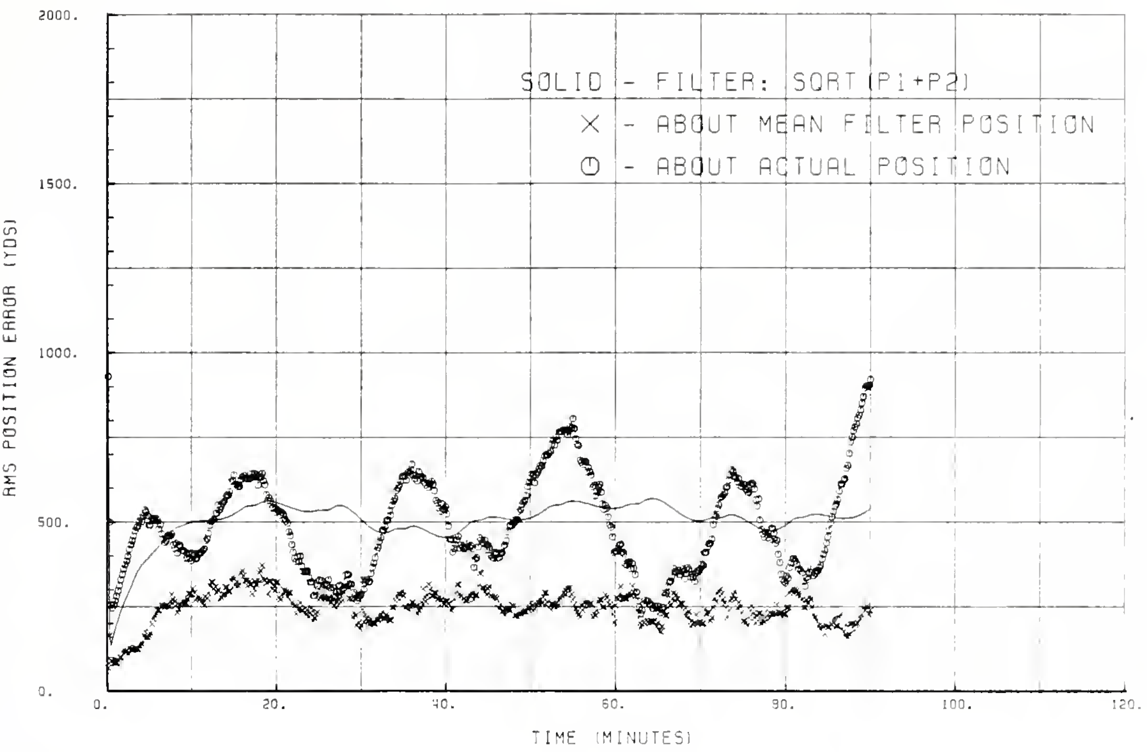


Figure 41. RMS errors for two-state system (top) and six-state system (bottom) using square pattern with t=4 sec.



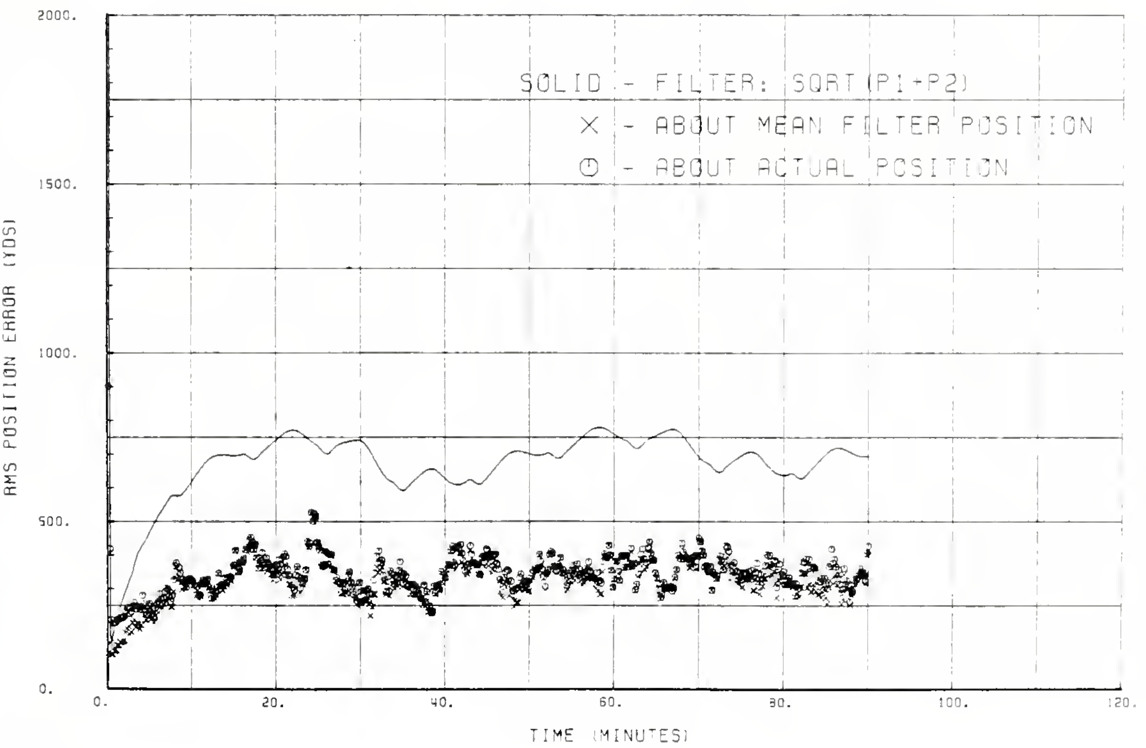


Figure 42. RMS errors for two-state system (top) and six-state system (bottom) using square pattern with t=10 sec.

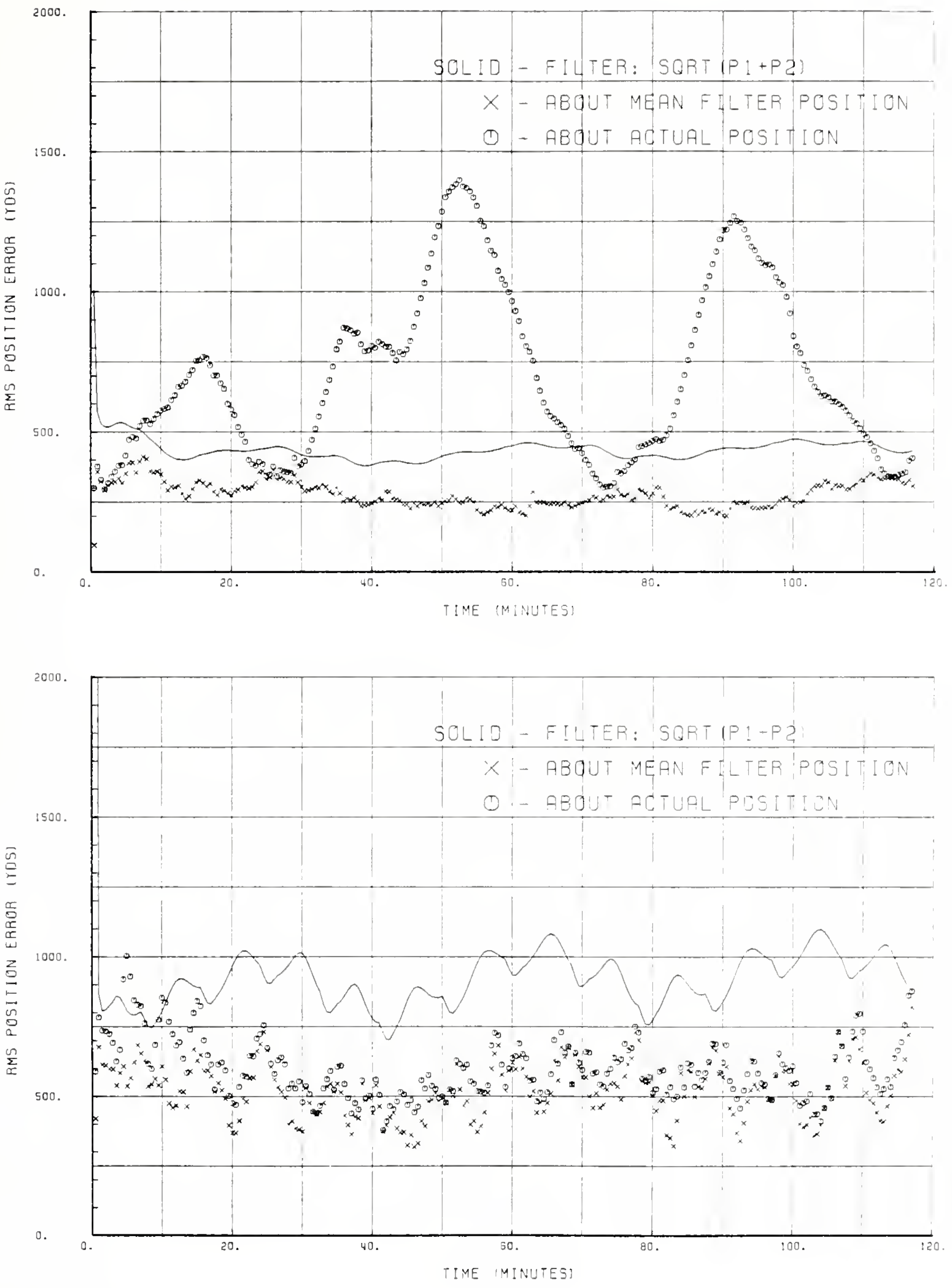
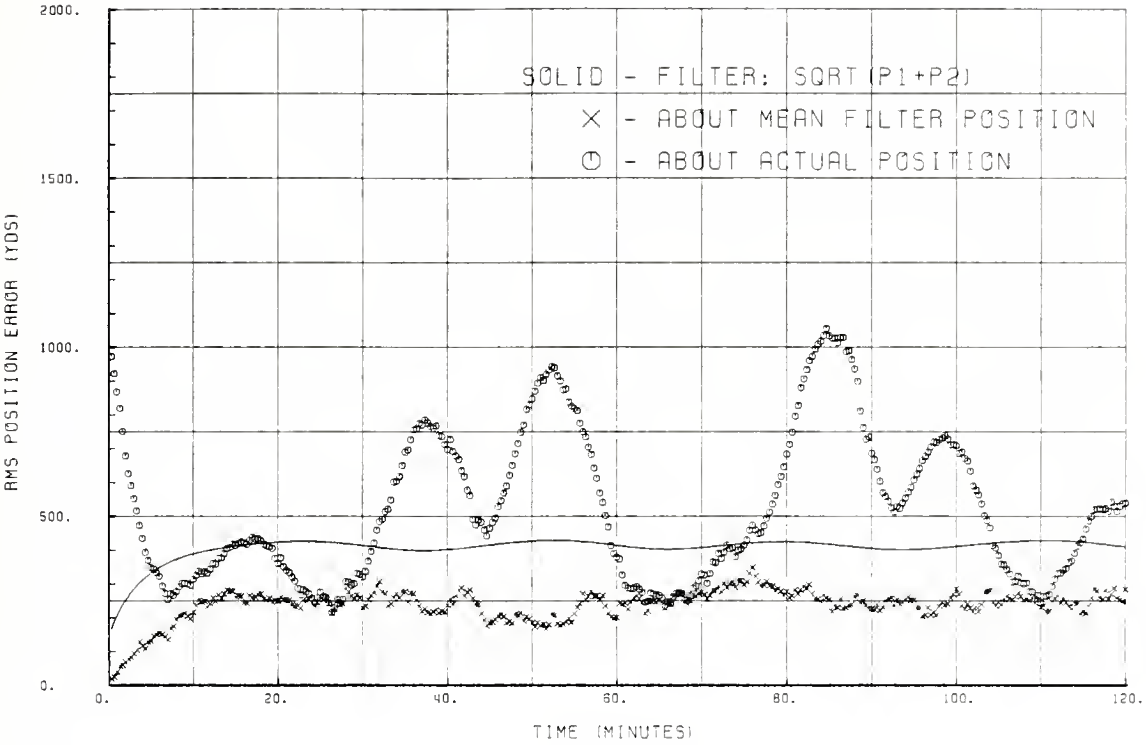


Figure 43. RMS errors for two-state system (top) and six-state system (bottom) using square pattern with t=30 sec.



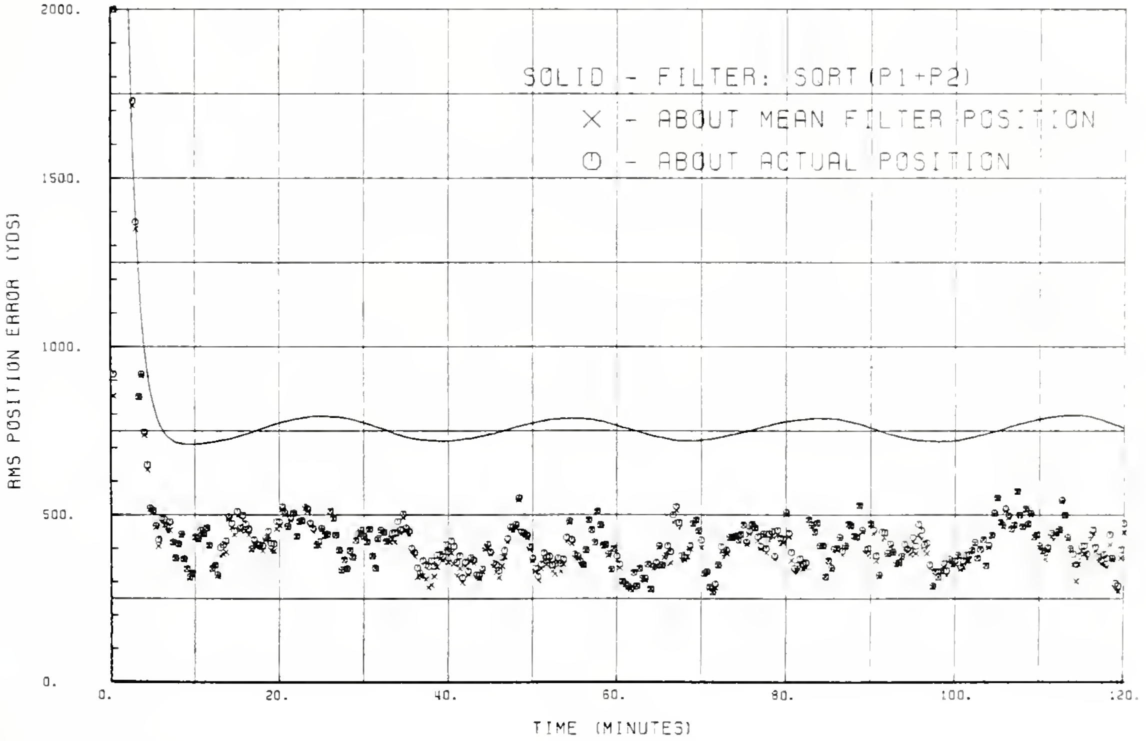
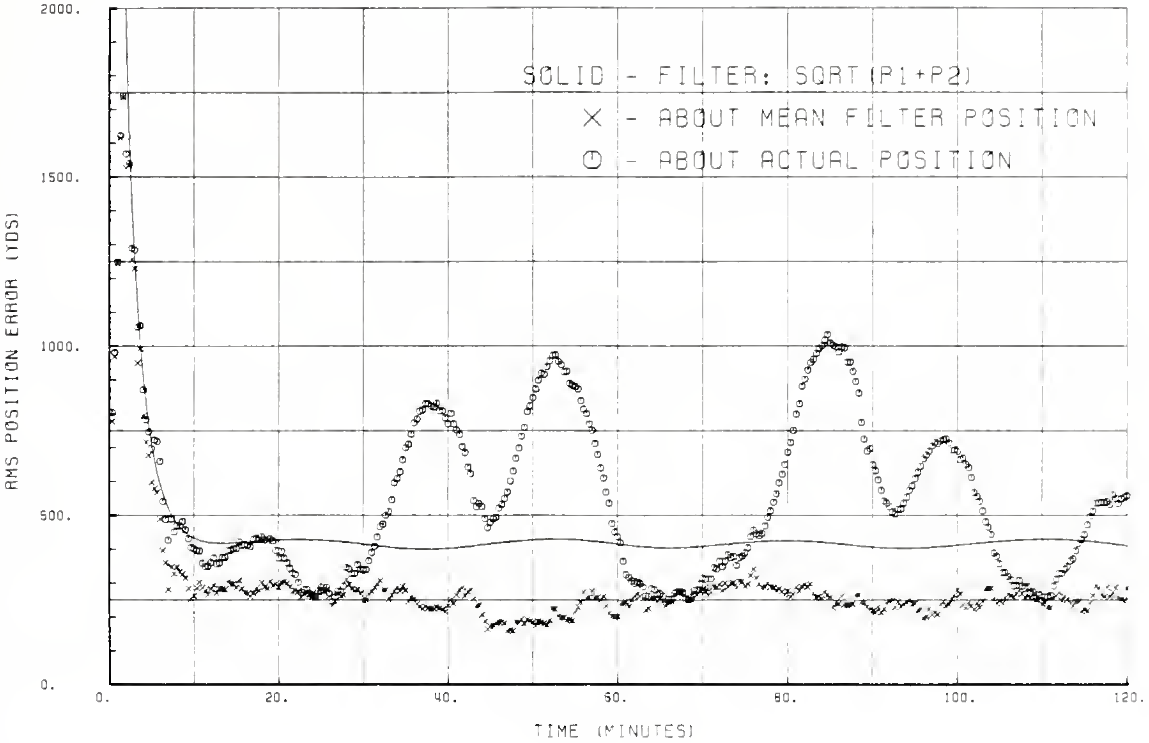


Figure 43. RMS errors for two-state system (top) and six-state system (bottom) using circular pattern, Alt=300'.



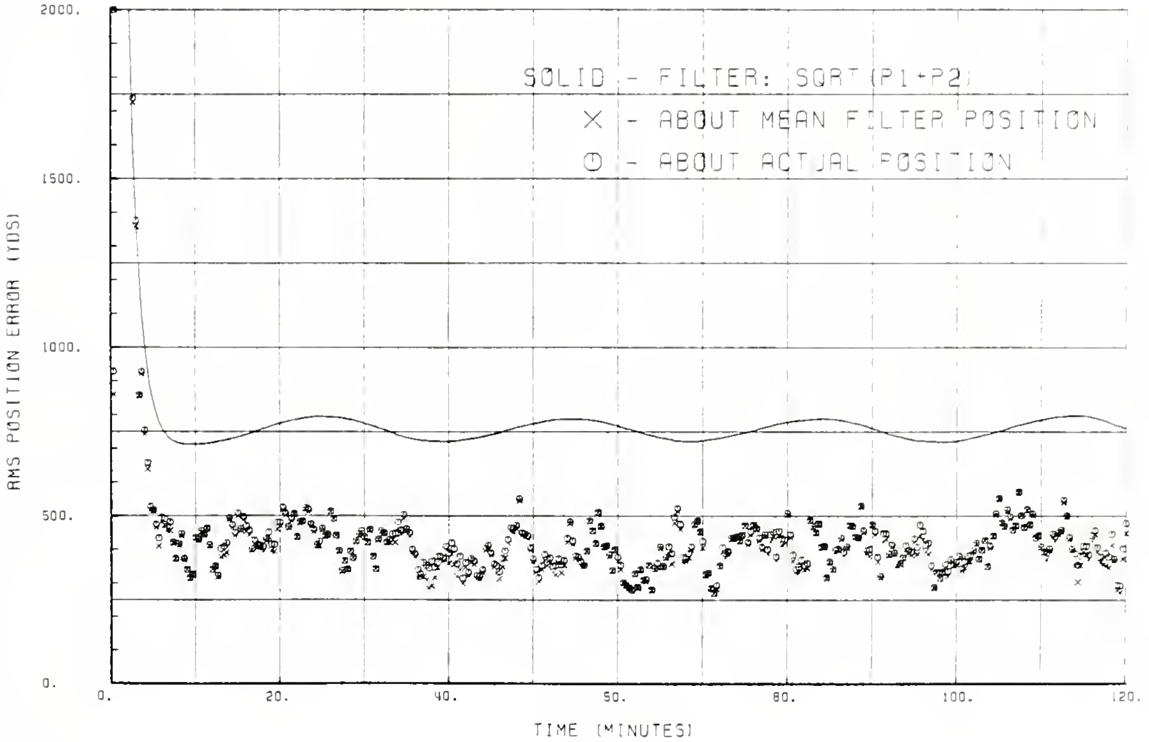
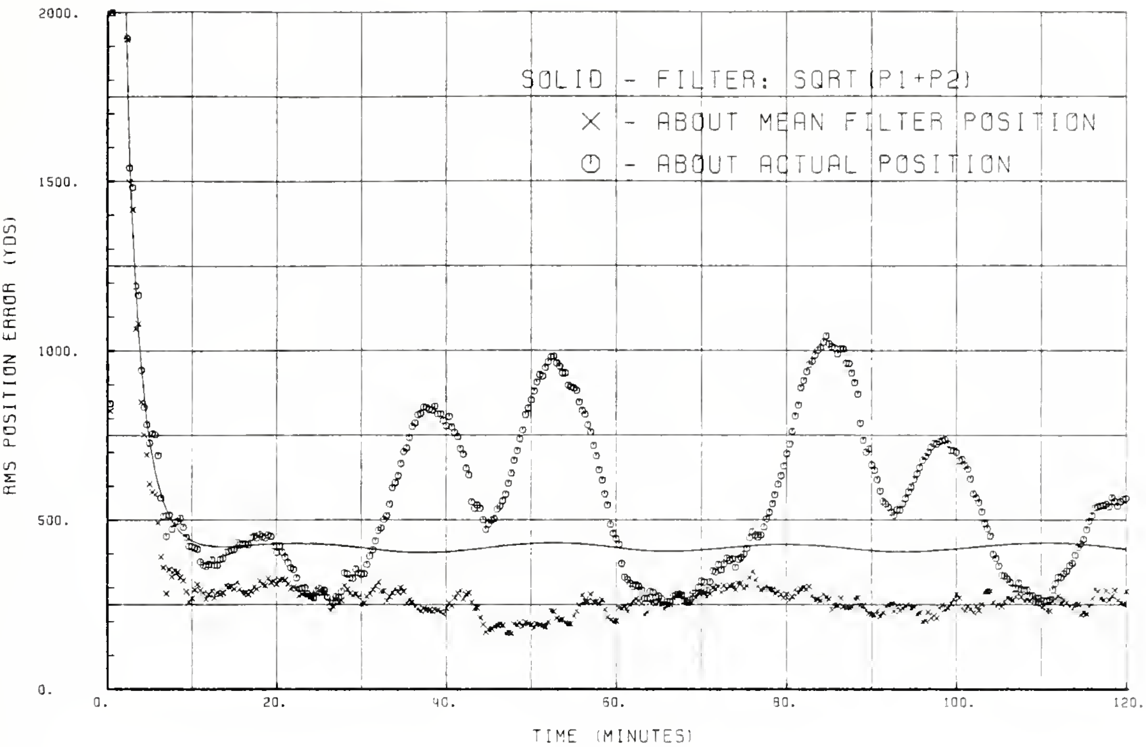


Figure 45. RMS errors for two-state system (top) and six-state system (bottom) using circular pattern, Alt=10,000'.



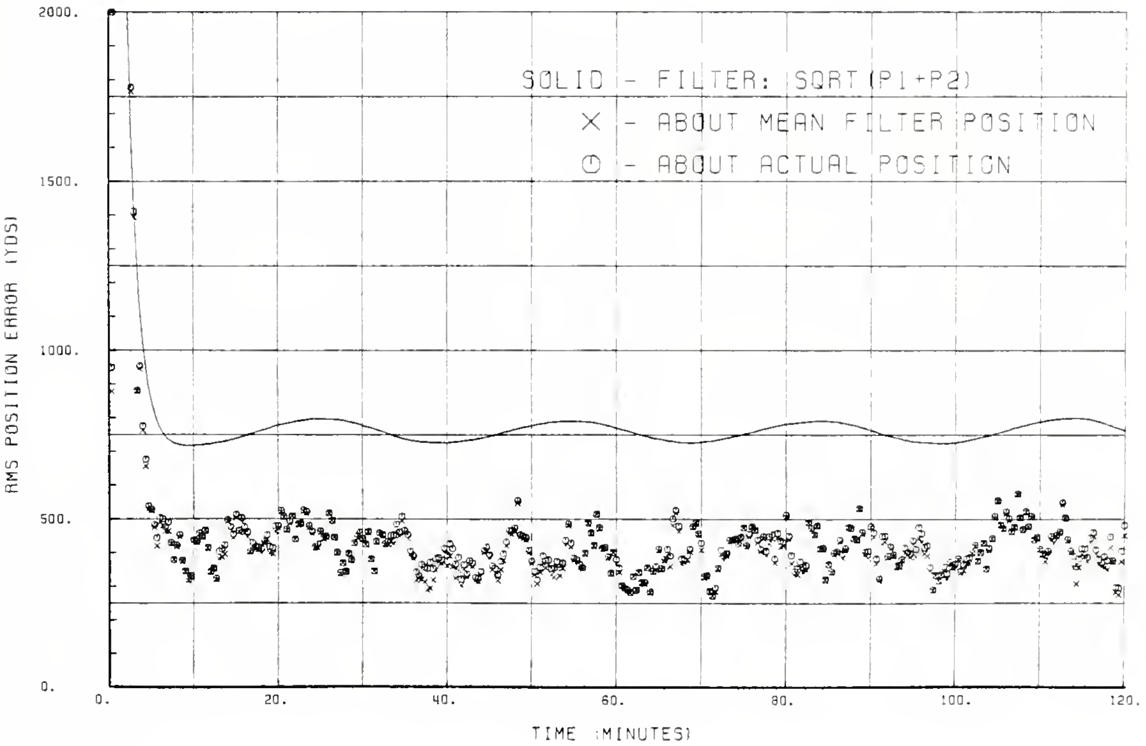


Figure 46. RMS errors for two-state system (top) and six-state system (bottom) using circular pattern, Alt=20,000'.

BIBLIOGRAPHY

Bryson, Jr., A. E. and Yu-Chi Ho, Applied Optimal Control, Hemisphere Publishing Corp. 1975.

Carlson, N. A., Fast Triangular Formulation of the Square Root Filter, Vol. II, No. 9, AIAA Journal, September 1973.

Demetry, J. S., Notes on the Theory and Applications of Optimal Estimation, Naval Postgraduate School, Monterey, California 1970.

Gelb, A. and others, Applied Optimal Estimation, M.I.T. Press 1974.

Leondes, C. T. Editor, Control and Dynamic Systems, Vol. 12, Academic Press, 1976.

Orincon Corporation Report OC-R-78-A003-1 Sonobuoy Reference System, by Asher, R. M., Judge, R., and Sorenson, H. W., December 1978.

System Functional Description for Sonobuoy Reference System, FD-17 (I 4.3), Naval Air Development Center, June 1978.

System Functional Description for Navigation, FD-9 (I 3/4.2) Naval Air Development Center, November 1977.

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Documentation Center 2
Cameron Station
Alexandria, Virginia 22314

2. Library, Code 0142 2
Naval Postgraduate School
Monterey, California 93940

3. Department Chairman, Code 62 1
Department of Aeronautics
Naval Postgraduate School
Monterey, California 93940

4. Prof. D. J. Collins, 67Co 3
Department of Aeronautics
Naval Postgraduate School
Monterey, California 93940

5. LT N. M. Brownsberger 2
3802 La Miranda Pl
Pittsburg, California 94565

6. Dr. D. Birnbaum, Code 2031 1
Naval Air Development Center
Warminister, Pennsylvania 18974

Thesis 184310
B8243 Brownsberger
c.1 Estimation of sonobuoy position relative to an aircraft using extended kalman filters.