REAL TIME KALMAN FILTERING FOR TORPEDO RANGE TRACKING

Dennis Michael Dwyer

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

REAL TIME KALMAN FILTERING FOR
TORPEDO RANGE TRACKING

by

Dennis Michael Dwyer

December 1978

Thesis Advisor: H. A. Titus

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)

REAL TIME KALMAN FILTERING FOR TORPEDO RANGE TRACKING

5. TYPE OF REPORT & PERIOD COVERED

Master's Thesis; December 1978

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)

Dennis Michael Dwyer

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

December 1978

13. NUMBER OF PAGES

98

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

15. SECURITY CLASS. (of this report)

Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

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

18. SUPPLEMENTARY NOTES

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

Torpedo Tracking
Kalman Filter

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

Two Extended Kalman filter routines, one using a one-step estimation/prediction and the other a sequential approach, were developed and compared to provide real time estimates of target positions on the three dimensional underwater tracking range at Naval Underwater Weapons Engineering Station, Keyport, Washington. Inputs to the routines were acoustic pulse transit times from the target to receiving array elements which are non-linear functions of the position coordinates. These inputs were linearized and the

filter gains calculated on-line. Simulated runs were conducted for tracks in the area of one hydrophone array and for tracks that transited through multiple arrays. It was found that the sequential estimate routine exhibited better performance in recovering from transients caused by random measurement noise or target movement.

REAL TIME KALMAN FILTERING
FOR
TORPEDO RANGE TRACKING

by

Dennis Michael Dwyer
Lieutenant, United States Navy
B.S., United States Naval Academy, 1973

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1978

ABSTRACT

Two Extended Kalman filter routines, one using a one-step estimation/prediction and the other a sequential approach, were developed and compared to provide real time estimates of target positions on the three dimensional underwater tracking range at Naval Underwater Weapons Engineering Station, Keyport, Washington. Inputs to the routines were acoustic pulse transit times from the target to receiving array elements which are non-linear functions of the position coordinates. These inputs were linearized and the filter gains calculated on-line. Simulated runs were conducted for tracks in the area of one hydrophone array and for tracks that transited through multiple arrays. It was found that the sequential estimate routine exhibited better performance in recovering from transients caused by random measurement noise or target movement.

TABLE OF CONTENTS

ACKNOWLEDGEMENT

The author is deeply indebted to Professor Hal Titus for his counsel during this project and also to Lieutenant Commander John Gauss for his professional guidance and assistance.

I. INTRODUCTION

The Naval Underwater Weapons Engineering Station, Keyport, Washington currently operates two three-dimensional underwater tracking ranges with the capability of acoustically tracking torpedoes. Underwater tracking employs an acoustic device installed in the object to be tracked. This device transmits timed acoustic pulses which are received by bottom mounted hydrophone arrays and then relayed via cable to a computer at the observation site which calculates the position of the object for each pulse and plots its path.

The measured data, which is the time elapsed from transmission of a pulse until its receipt at the hydrophone array, is corrupted by noise due to the combined effects of environmental factors and the measurement instruments.

These noisy tracks are later analyzed, and measurements judged most inaccurate on the basis of total track statistics are removed in order to obtain a smooth representation of the track.

As stated in Reference 1, the computer system at the Dabob range of the station will be up-graded, and will consist of three MODCOMP IV computers by DATACOM, Inc. A software conversion project will take place with applications software being developed for the new computers. The bulk of this development will consist of converting the current tracking programs and other related programs to FORTRAN.

An opportunity exists for expanding the real-time capability of the system by applying a Kalman filter routine which can take as an input the transit times of the acoustic pulses, and produce the best estimate of the position of the tracked object at a particular time.

II. THREE DIMENSIONAL RANGE DESCRIPTION

The three dimensional range described in Reference 2 is an acoustic system capable of determining the trajectory of suitably instrumented underwater objects in the vicinity of a transducer array placed at the bottom of the bay. The tracked unit (torpedo) carries a synchronous clock and an acoustic transducer. A hydrophone array defines a rectangular coordinate system to which measurements are referred. Positional information is obtained from the transit times of a periodic pulsed acoustic signal traveling from the torpedo to four independent hydrophones located on each array. The geometry of the hydrophones and the coordinate system is illustrated in Figure 1. On each array, the four hydrophones R_C, R_X, R_Y and R_Z are on four adjacent vertices of a cube. Each hydrophone is separated by a distance d= 30 feet along the edges. The origin of the coordinate system is at the center of the cube.

The transit times of the acoustic pulse from the tracked object to each of the four independent hydrophones can be expressed as follows:

$$T_C = 1/VEL \left[(X+d/2)^2 + (Y+d/2)^2 + (Z+d/2)^2 \right]^{1/2}$$

$$T_X = 1/VEL \left[(X-d/2)^2 + (Y+d/2)^2 + (Z+d/2)^2 \right]^{1/2}$$

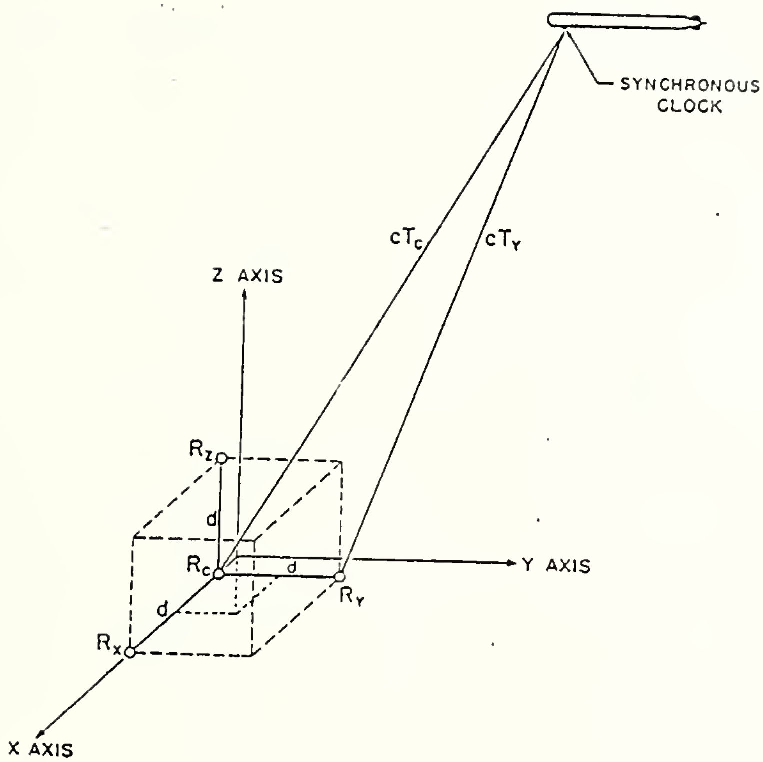


FIGURE 1: The Geometry Used in the Calculation of Positional Coordinates (The origin is at the center of the dashed cube).

$$T_Y = 1/VEL \left[(X+d/2)^2 + (Y-d/2)^2 + (Z+d/2)^2 \right]^{1/2}$$

$$T_Z = 1/VEL \left[(X+d/2)^2 + (Y+d/2)^2 + (Z-d/2)^2 \right]^{1/2}$$

Where VEL equals the velocity of propagation of sound in water and C, X, Y and Z are the four hydrophones on each array. It is essential as part of this fundamental calculation to know when the tracked object emits an acoustic pulse in order to measure the transit times to the hydrophones. For submerged objects two stable crystal-controlled clocks, one in the tracked unit and the other at the computer are used. Prior to a run the two clocks are synchronized by radio.

The range is a high frequency, short-baseline facility with the acoustic tracking pulses emitted at 75 kHz. The range layout consists of six in-line hydrophone arrays spaced 2000 yards apart.

III. THEORY

In Reference 3, a Kalman filter application was used assuming a linear system and filtering on the corrupted X, Y and Z positions that the computer had already calculated from the received transit times of the acoustic signals.

A. THE EXTENDED KALMAN FILTER

Since the transit times were readily available and are non-linear functions of position, these equations can be linearized and Kalman filter theory applied using the extended Kalman filter. This procedure produces a real-time system, filtering on the corrupted transit times T_C, T_X, T_Y and T_Z, without the necessity of converting these times to positions.

For tracking, a fifth order state vector was chosen:

$$\underset{\sim}{X} = \begin{bmatrix} X \\ \dot{X} \\ Y \\ \dot{Y} \\ Z \end{bmatrix}$$

The target was assumed to maintain constant depth and any velocity in the Z direction ($\dot{Z}$) was considered a random excitation.

The states were characterized by the following difference equation:

$$\underset{\sim}{X}(K+1) = \underset{\sim}{\Phi}\ \underset{\sim}{X}(K) + \underset{\sim}{\Gamma}\ \underset{\sim}{W}(K)$$

and the noisy measurement equation:

$$\underset{\sim}{Z}(K) = M(\underset{\sim}{X}) + \underset{\sim}{V}(K)$$

where

$\underset{\sim}{X}(K)$ is the N-dimensional state vector at time K

$\underset{\sim}{W}(K)$ is the M-dimensional random forcing input at time K

$\underset{\sim}{Z}(K)$ is the J-dimensional measurement vector at time K

$\underset{\sim}{V}(K)$ is the J-dimensional random noise vector at time K and noise is assumed white and zero-mean Gaussian

Φ and Γ are constant matrices

$M(\underset{\sim}{X})$ is a matrix of the non-linear measurement equations which are a function of the states.

The estimator equations are given by:

$$\underset{\sim}{\hat{X}}(K/K) = \underset{\sim}{\hat{X}}(K/K-1) + G(K) * \left[\underset{\sim}{Z}(K) - M(\underset{\sim}{X}) * \underset{\sim}{\hat{X}}(K/K-1) \right]$$

and

$$\underset{\sim}{P}(K/K) = \left[\underset{\sim}{I} - \underset{\sim}{G}(K)\underset{\sim}{H}(K) \right] \underset{\sim}{P}(K/K-1)$$

where

$\underset{\sim}{\hat{X}}(K/K)$ is the estimate of the state at time K given K measurements

$\underset{\sim}{P}(K/K)$ is the covariance of estimation error matrix at time K

$\underset{\sim}{G}(K)$ is the Kalman filter gain at time K

which is defined as:

$$\underset{\sim}{G}(K) = \underset{\sim}{P}(K/K-1)\ \underset{\sim}{H}(K)^T \left[\underset{\sim}{H}(K)\underset{\sim}{P}(K/K-1)\underset{\sim}{H}(K)^T + \underset{\sim}{R}(K) \right]^{-1}$$

$\underset{\sim}{R}(K)$ is the covariance of random measurement noise matrix

$\underset{\sim}{H}(K)$ is the matrix used to linearize the non-linear measurement equations. It is defined as:

$$H(K) = \frac{\partial M(\underset{\sim}{X})}{\partial \underset{\sim}{X}} \Bigg|_{\hat{\underset{\sim}{X}}(K/K-1)}$$

The measurement equation matrix $M(\underset{\sim}{X})$ is expanded in a Taylor series and linearized around the one step prediction based on the last estimate. Only first order terms are kept.

The prediction equations are given by:

$$\hat{\underset{\sim}{X}}(K+1/K) = \underset{\sim}{\Phi}\hat{\underset{\sim}{X}}(K/K) + \Gamma\, \underset{\sim}{W}(K)$$

$$\underset{\sim}{P}(K+1/K) = \underset{\sim}{\Phi}\underset{\sim}{P}(K/K)\underset{\sim}{\Phi}^T + \underset{\sim}{Q}(K)$$

where

$\underset{\sim}{Q}(K)$ is the covariance of random excitation matrix found by:

$$\underset{\sim}{Q}(K) = \Gamma\ COV(W)\Gamma^T$$

and COV(W) is the covariance of the random forcing input or acceleration.

B. THE SEQUENTIAL EXTENDED KALMAN FILTER

The prediction and estimation equations just mentioned represent a system that is characterized by a one-step estimation and prediction for each set of new measurements. Some important points in this one step process must be stressed:

1. New estimates are only available after the prediction and estimation equations are calculated.
2. A major part of the time period for the estimation calculation is spent in the gain equation because of the inversion of the (JXJ) matrix that is required, where J is the number of observations.

3. The one-step prediction and estimation for the extended Kalman filter is based on a linearization about the predicted value over a period t_K to t_{K+1} which might be extended.

When the measurements come from statistically independent sources, like the hydrophones in the torpedo tracking problem, a sequential approach can be taken to process each arrival time separately.

If the measurements are assumed to occur simultaneously, they can be processed one at a time and the result of processing one measurement component is used in the following computation to process the next measurement component.

IV. PROBLEM DEFINITION - TORPEDO TRACKING WITH THE EXTENDED KALMAN FILTER

In the torpedo tracking problem, the non-linear observations are the four independent transit times from the tracked object to the hydrophones, T_C, T_X, T_Y and T_Z. Thus the non-linear measurement matrix $M(\underset{\sim}{X})$ is defined as:

$$M(\underset{\sim}{X}) = \begin{bmatrix} T_C \\ T_X \\ T_Y \\ T_Z \end{bmatrix} \begin{bmatrix} \frac{1}{VEL} \left[(X+d/2)^2 + (Y+d/2)^2 + (Z+d/2)^2 \right]^{\frac{1}{2}} \\ \frac{1}{VEL} \left[(X-d/2)^2 + (Y+d/2)^2 + (Z+d/2)^2 \right]^{\frac{1}{2}} \\ \frac{1}{VEL} \left[(X+d/2)^2 + (Y-d/2)^2 + (Z+d/2)^2 \right]^{\frac{1}{2}} \\ \frac{1}{VEL} \left[(X+d/2)^2 + (Y+d/2)^2 + (Z-d/2)^2 \right]^{\frac{1}{2}} \end{bmatrix}$$

$M(\underset{\sim}{X})$ is expanded into a Taylor series and linearly approximated by:

$$M(\underset{\sim}{X}) \approx \frac{\partial M(\underset{\sim}{X})}{\partial \underset{\sim}{X}} \Bigg|_{\hat{\underset{\sim}{X}}(K/K-1)} \underset{\sim}{X} = H(\underset{\sim}{X}) \Bigg|_{\hat{\underset{\sim}{X}}(K/K-1)} \underset{\sim}{X}$$

The linearizing H matrix is evaluated around the best information available at the time which is the prediction $\hat{\underset{\sim}{X}}(K/K-1)$, and is used in the calculation of the gain $\underset{\sim}{G}(K)$ and estimated covariance of error $\underset{\sim}{P}(K/K)$ equations.

The torpedo dynamics used for the tracking problem are assumed to be $1/S^2$ with extimations on five states X position,

X velocity, Y position, Y velocity and Z position (height of torpedo above hydrophone array). The mean of the random excitation is assumed to be zero $\left[E(W(K))=0\right]$, and this simplifies the estimate prediction equation to:

$$\hat{\underset{\sim}{X}}(K+1/K) = \underset{\sim}{\Phi}\ \hat{\underset{\sim}{X}}(K/K)$$

Also the mean of the random ncise is assumed to be zero $\left[E(V(K))=0\right]$. In forming the measurement equation, the best guess of v(K) is used which is the mean producing:

$$\underset{\sim}{Z}(K) = M(\underset{\sim}{X})$$

Four measurements are taken every 1.31 seconds, which is one time slot, and with this sampling time the $1/_{S}2$ plant has state transition (PHI) and gamma matrices equal to:

$$\underset{\sim}{\Phi} = \begin{bmatrix} 1 & 1.31 & 0 & 0 & 0 \\ 0 & 1. & 0 & 0 & 0 \\ 0 & 0 & 1. & 1.31 & 0 \\ 0 & 0 & 0 & 1. & 0 \\ 0 & 0 & 0 & 0 & 1. \end{bmatrix}$$

and

$$\underset{\sim}{\Gamma} = \begin{bmatrix} .86 & 0 & 0 \\ 1.31 & 0 & 0 \\ 0 & .86 & 0 \\ 0 & 1.31 & 0 \\ 0 & 0 & 1.31 \end{bmatrix}$$

A. THE SEQUENTIAL APPROACH

In the sequential approach, the basic Kalman filter equations

have been modified to circumvent the matrix inversion in the gain equation and to obtain a more accurate estimate. Calculations are performed on each of the four independent transit times in the following order T_C, T_X, T_Y and T_Z for each 1.31 second time slot.

The estimate of the states, $\hat{X}(K/K)$, based on one time measurement is used as the prediction $\hat{X}(K/K-1)$ for the calculations on the next measurement.

In this manner only parts of the linearizing H matrix and gain matrices are used in each calculation.

After the linearizing H matrix is formed from

$$H(X) = \frac{\partial M(X)}{\partial X} \Bigg|_{\hat{X}(K/K-1)}$$

evaluated around the initial states $\hat{X}(1/O)$ for K=1, the first gain column corresponding to the first time measurement T_C is calculated from:

$$G_{iCOL} = \frac{P(K/K-1)H^T_{iROW}}{H_{iROW}P(K/K-1)H^T_{iROW}+R_{ii}}$$

where i = 1 to J, and J is the number of observations. In the tracking problem J = 4 corresponding to the four measured times.

Thus, the first row of the H matrix is used to calculate the first column of the gain matrix with both corresponding to the first measured time T_C.

Next, an estimate of the particular observation time $\hat{M}(\underset{\sim}{X})$ is calculated

$$\hat{M}(\underset{\sim}{X}) = \begin{bmatrix} \hat{T}_C \\ \hat{T}_X \\ \hat{T}_Y \\ \hat{T}_Z \end{bmatrix} = \begin{bmatrix} \frac{1}{VEL} \left[(X+d/2)^2 + (Y+d/2)^2 + (Z+d/2)^2 \right]^{1/2} \\ \frac{1}{VEL} \left[(X-d/2)^2 + (Y+d/2)^2 + (Z+d/2)^2 \right]^{1/2} \\ \frac{1}{VEL} \left[(X+d/2)^2 + (Y-d/2)^2 + (Z+d/2)^2 \right]^{1/2} \\ \frac{1}{VEL} \left[(X+d/2)^2 + (Y+d/2)^2 + (Z-d/2)^2 \right]^{1/2} \end{bmatrix}$$

Using the predicted values of X, Y, Z from $\hat{\underset{\sim}{X}}(K/K-1)$.

The difference between the observed transit time Z_i and the estimated transit time $\hat{T}_i$ forms the residual Z_{DIFF} which is used in the estimate equation.

$$\hat{X}_i = \hat{X}(K/K-1) + G_i COL \left[Z_{DIFF} \right]$$

This equation gives an estimate of the states based on one measurement.

Next, the covariance of estimation error is calculated based on one measurement using:

$$P_i = \left[\underset{\sim}{I} - G_{iCOL} H_{iROW} \right] P_{i-1}$$

where

$\underset{\sim}{I}$ equals the identity matrix

P_{i-1} is the theoretical covariance of estimation error from the previous measurement or if i=1, the prediction $\underset{\sim}{P}(K/K-1)$

After the first iteration, $\underset{\sim}{X}_1$ becomes $\underset{\sim}{X}(K/K-1)$ and P_1 becomes $\underset{\sim}{P}(K/K-1)$ for the second iteration which calculates the estimate of the states based on the second measurement T_X.

After four iterations (i=4), $\underset{\sim}{X}_4$ becomes the estimate for the time slot, $\hat{X}(K/K)$ and P_4 becomes the updated covariance of error $\underset{\sim}{P}(K/K)$.

Then the predictions for the next time slot are calculated using:

$$\underset{\sim}{\hat{X}}(K+1/K) = \underset{\sim}{\Phi}\ \underset{\sim}{\hat{X}}(K/K)$$

and

$$\underset{\sim}{P}(K+1/K) = \underset{\sim}{\Phi}\ \underset{\sim}{P}(K/K)\Phi^T + \underset{\sim}{Q}(K)$$

The entire process is repeated for the next set of measurements forming a sequential, extended Kalman filter to produce real time estimates of the torpedo track.

A listing of the FORTRAN program designed for the sequential extended Kalman filter is contained in Appendix B. The program is in modular form and well documented by comments for ease of implementation. All repetitive calculations and utility routines are separated into subroutines and are listed in Appendix D.

B. THE MATRIX INVERSION APPROACH

All results obtained using the sequential, extended Kalman filter were compared to results from a traditional extended Kalman filter using the equations delineated in the THEORY section. This includes an inversion of a 4x4 matrix in the gain equation.

$$\underset{\sim}{G}(K) = \underset{\sim}{P}(K/K-1)\underset{\sim}{H}(K)^{T}\left[\underset{\sim}{H}(K)\underset{\sim}{P}(K/K-1)\underset{\sim}{H}(K)^{T}+\underset{\sim}{R}(K)\right]^{-1}$$

The portion of the equation to be inverted is always symmetric and the IBM-360 library subroutine SINV was used to perform this operation.

A listing of the FORTRAN program designed for the traditional extended Kalman filter is contained in Appendix C.

V. TESTING AND SIMULATION

Both the sequential and traditional Kalman filter routines were tested first using deterministic tracks at speeds of 5.0 to 25.0 knots and no measurement noise with a single hydrophone array. The only errors allowed in the first phase of testing were in position and velocity in the initialization of the filter.

Computer generated tracks were tested in the first series of straight running, constant depth and constant velocity torpedoes. A variety of track scenerios were used transiting through multiple quadrants including:

1. crossing north of the array
2. crossing south of the array
3. inbound to the array
4. outbound from the array
5. crossing over top of the array

All runs were made with a variety of initialization errors in position and velocity.

In the second series of tests, white, zero-mean Gaussian noise was added to corrupt the observed transit times.

The noise was added to the straight running, constant depth tracks. Before this series of tests could be conducted a gating scheme was designed to protect the filter from spurious erroneous time or positional data.

In the third series of tests, a number of torpedo maneuvers

were added to the target tracks. One-third, two-thirds and one-G turns were used. These tracks were tested with and without noise.

The torpedo velocities were increased to the 40 to 50 knot range in the fourth series of tests for straight running tracks with and without noise corruption. Maneuvers were then added to these higher velocity tracks.

In the last series of tests, the handoff routine described at the end of this section was added to the filters and tracks that traversed through the areas of multiple arrays were tested.

A. THE GATING SCHEME

The operation of the filter may be adversely affected by large measurement noise. One error of a relatively large magnitude could invalidate the filtered output for many subsequent time slots. Before random measurement noise and random excitations could be added to the observed times for testing, a form of protection was designed to guard against catastrophic failure. This protection is provided by establishing limits of acceptability for each of the measurements.

Measurement errors can occur because of many factors including an error in the transit time of the acoustic pulse primarily due to the receipt of multipath signals from previous time slots that have bounced off the surface, bottom or different density layers, or large errors in the estimates of position or velocity.

A three-sigma gate was deisgned using the covariance of measurement noise (R) and the covariance of estimation error (P(K/K)).

For each calculation of a state estimate ($\underset{\sim}{X}(K/K)$), the largest positional covariance of error was used, either X, Y or Z, and converted to time in seconds using the average velocity of sound in water for Dabob bay, 4860 ft/sec. The gate then was written for each time measurement i = 1 to 4:

$$\text{GATE} = \sqrt{\frac{P(K/K)_{\text{largest}}}{4860.} + R_{ii}}$$

The gate expands or decreases depending on the confidence level of the transit time and position estimate. If ZDIFF which is the difference between the actual transit time received and the predicted transit time to a particular hydrophone exceeds the gate, the measurement is considered unacceptable and the filter gain is set to zero causing the filter to ignore the data and take the prediction of the states as the estimate.

$$\hat{\underset{\sim}{X}}(K/K) = \hat{\underset{\sim}{X}}(K/K-1)$$

For the sequential extended Kalman filter, because of the iterative aspect, a large erroneous time measurement zeros only the gain column for that particular hydrophone causing only that hydrophone's data to be ignored. In the traditional one-step filter, an erroneous input from any hydrophone zeros the entire gain matrix causing the filter to ignore data for the whole time slot.

B. MULTIPLE ARRAY TRACKING

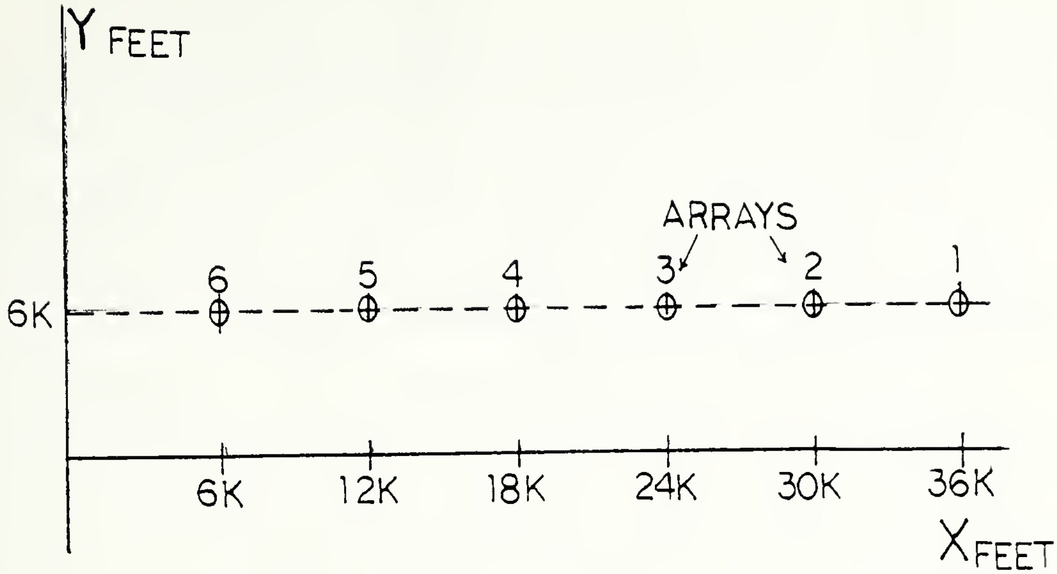
Initial tests were performed on tracks in the area of one array. In order to more closely simulate a typical run on the range, a scheme was designed to track a target through multiple arrays.

First, a coordinate system was defined as shown in Figure 2. The center of the coordinate system is geographically near the entrance to Dabob bay in the simulation. Array number 6 is the closest array to the coordinate center. Each hydrophone is a particular array has an X, Y, Z position. In the simulation array 1 was at 36,000 feet from coordinate center and array 6 was 6000 feet. The C hydrophone was assumed to be the axis location of each array. Then each X position for the X hydrophone in each array was X_C+30, each Y position for the Y hydrophone was Y_C+30 and each Z position for the Z hydrophone was Z_C+30. These 72 positions, an XYZ position for each of 4 hydrophones in 6 arrays, were placed into a 6x12 matrix HYDRO and referenced throughout the routine. The geometry centered on each array was taken out of the problem and the target position was based on a central reference.

The non-linear time equation became:

$$T = 1/VEL \sqrt{(X-X_0)^2+(Y-Y_0)^2+(Z-Z_0)^2}$$

where X_0, Y_0 or Z_0 is the position of a particular hydrophone and array being used. In the filter routine X, Y and Z were the predicted positions $\hat{\underset{\sim}{X}}(K/K-1)$, and the time equation was



Coordinate System for Multiple Array Tracking

	C Hydro			X Hydro			Y Hydro			Z Hydro		
ARRAYS	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z
1	36000	6000	0	36030	6000	0	36000	6030	0	36000	6000	30
2	30000	6000	0	30030	6000	0	30000	6030	0	30000	6000	30
3	24000	6000	0	24030	6000	0	24000	6030	0	24000	6000	30
4	18000	6000	0	18030	6000	0	18000	6030	0	18000	6000	30
5	12000	6000	0	12030	6000	0	12000	6030	0	12000	6000	30
6	6000	6000	0	6030	6000	0	6000	6030	0	6000	6000	30

HYDRO -- Hydrophone Location Matrix

FIGURE 2

used to calculate the estimate of the measurement times $\hat{M}(\underset{\sim}{X})$. The decision parameter used to determine the switching from array to array was a straight handoff. If the predicted X position was greater than 3000 feet from the array in use, then an index (I8) was incremented and the next row of HYDRO was implemented. This placed into the routine the $X_0Y_0Z_0$ positions of the hydrophones in the next array. The handoff can easily be utilized in real range operations, as the transit times from adjacent arrays are present at the computer for a particular time slot.

For simulation, it was assumed that in all the arrays each axis pointed in the same direction. In range operations, the positions of the particular hydrophones referenced to the central coordinate system can be input into the matrix HYDRO to correct for OFF AXIS discrepancies.

VI. SIMULATION RESULTS

A. SERIES ONE

This series of tests included straight running, constant depth, constant velocity tracks with no noise. Various target speeds were tested ranging from 5.0 to 25.0 knots.

The only induced errors in this series of tests were in initial target position and velocity. Both the sequential and traditional filter routines effectively handled initial position errors in the X AND/OR Y direction from 0 to 25 feet and initial velocity errors from 0 to 10 ft/sec. The filter estimate was within 3 feet and 1 ft/sec in a maximum of 3 time slots. In a number of worst case tests, initial position errors of up to 50 feet and initial velocity errors of 60 ft/sec and 80 ft/sec were used. Both filter routines had the estimate on track within seven time slots.

Figure 3 is a geographical plot of a typical series one test using the sequential extended Kalman filter. The initialization of the filter was 14 feet off in X and 21 feet off in Y with no error in Z. For a 25 knot target initial velocity errors were 3 ft/sec. Figures 4 through 6 depict the deviation in feet between the estimated and true positions, $X_A(K) - \hat{X}_A(K/K)$, where A = 1, 3 or 5.

There was no great difference in performance between the sequential and traditional filter routines in this series of tests.

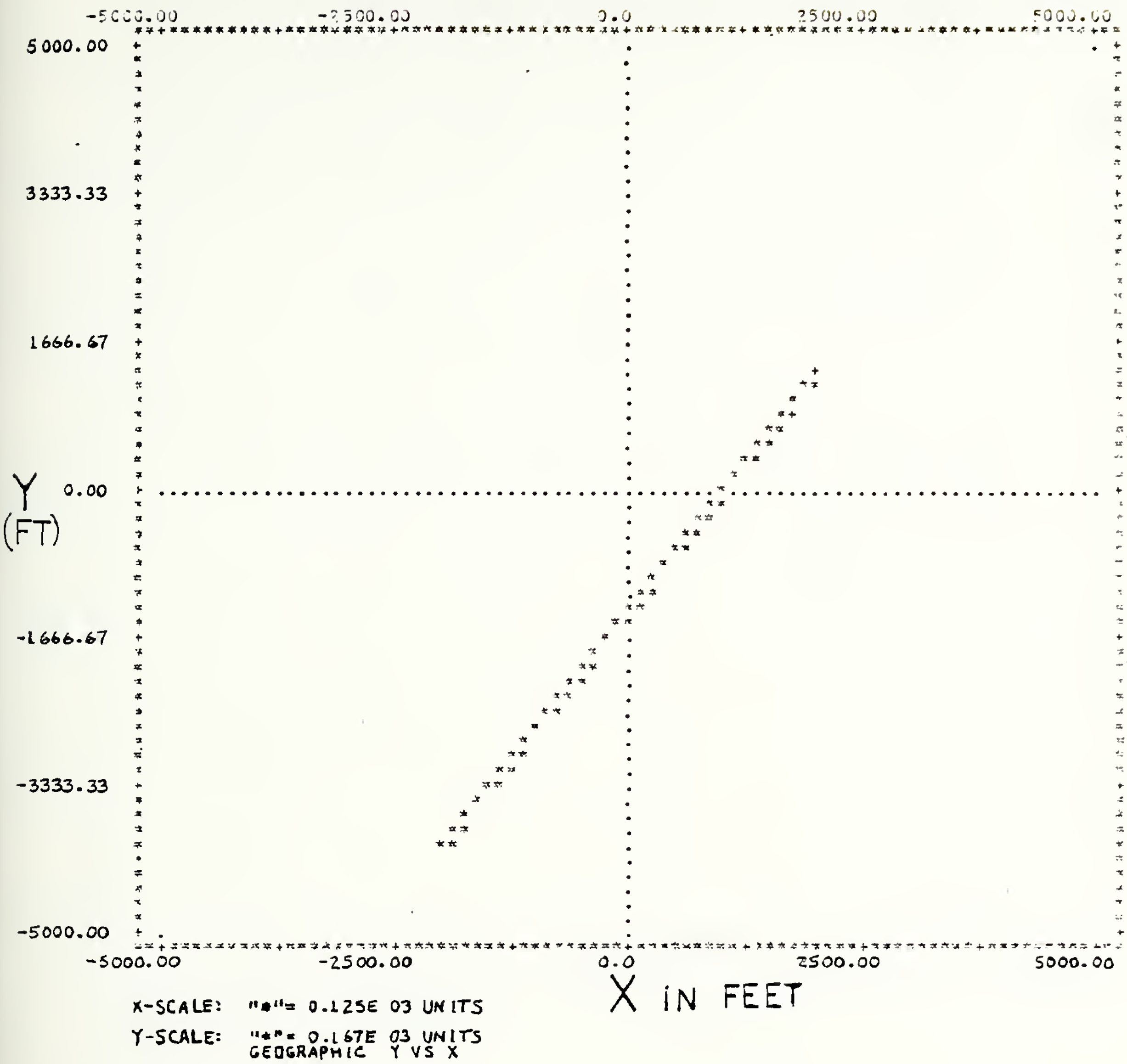


FIGURE 3: Geographic Plot Straight Running Track in the area of a single Array

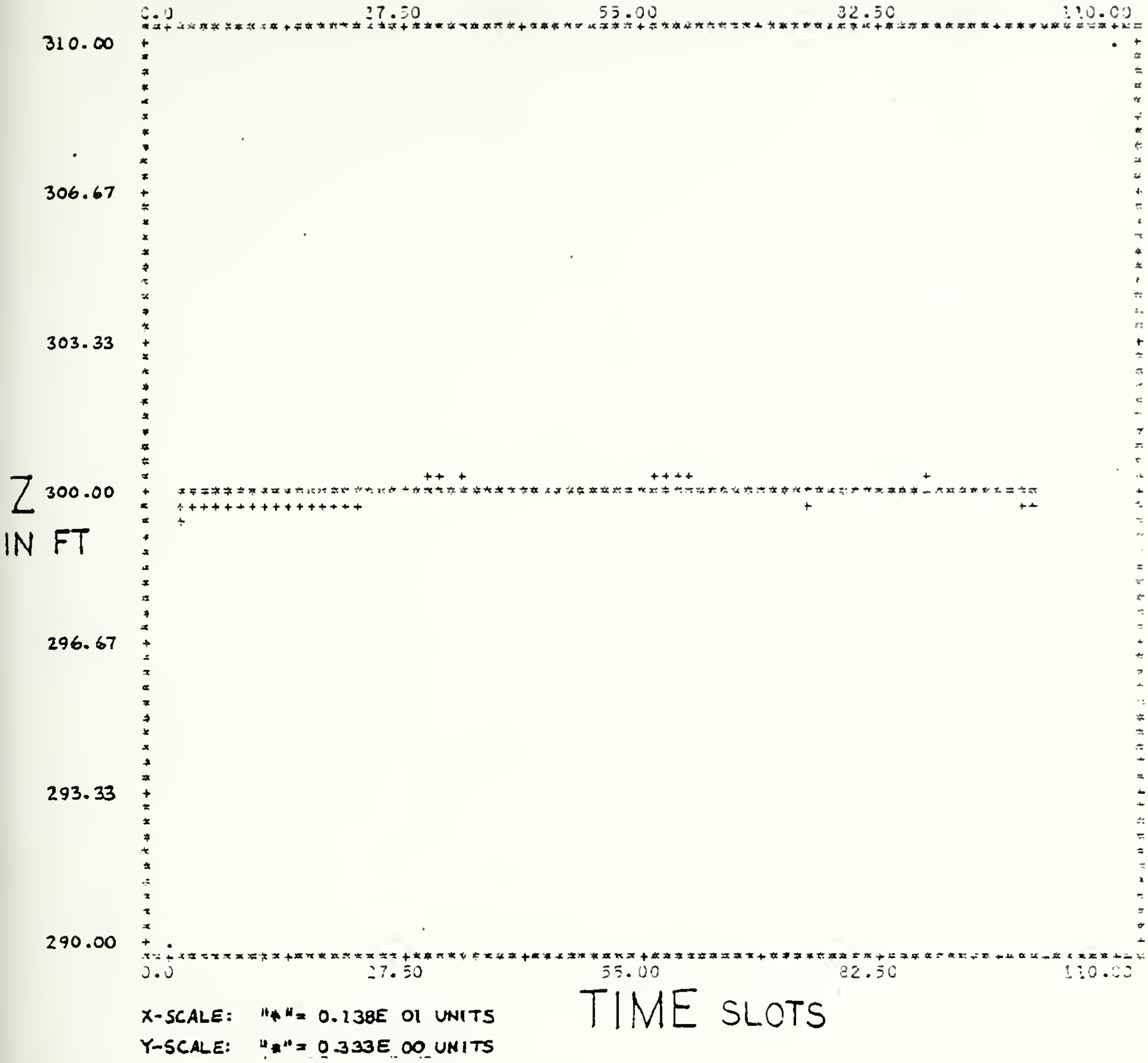


FIGURE 4: Z Position versus Time for a Straight Running Track with No Noise in the Area of a Single Array

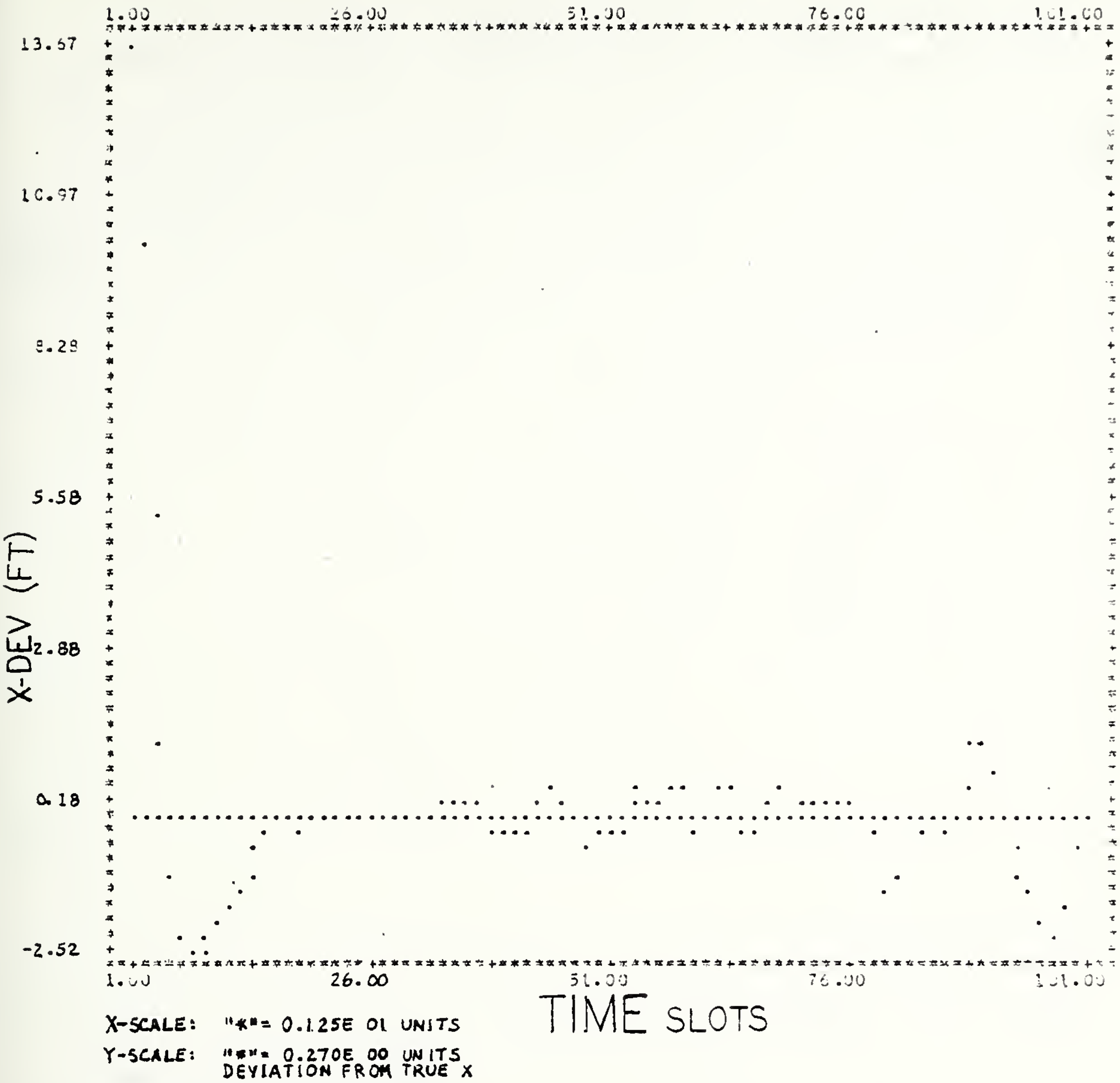


FIGURE 5: X Deviation from True versus Time for a Straight Running Track without Noise in the Area of a Single Array

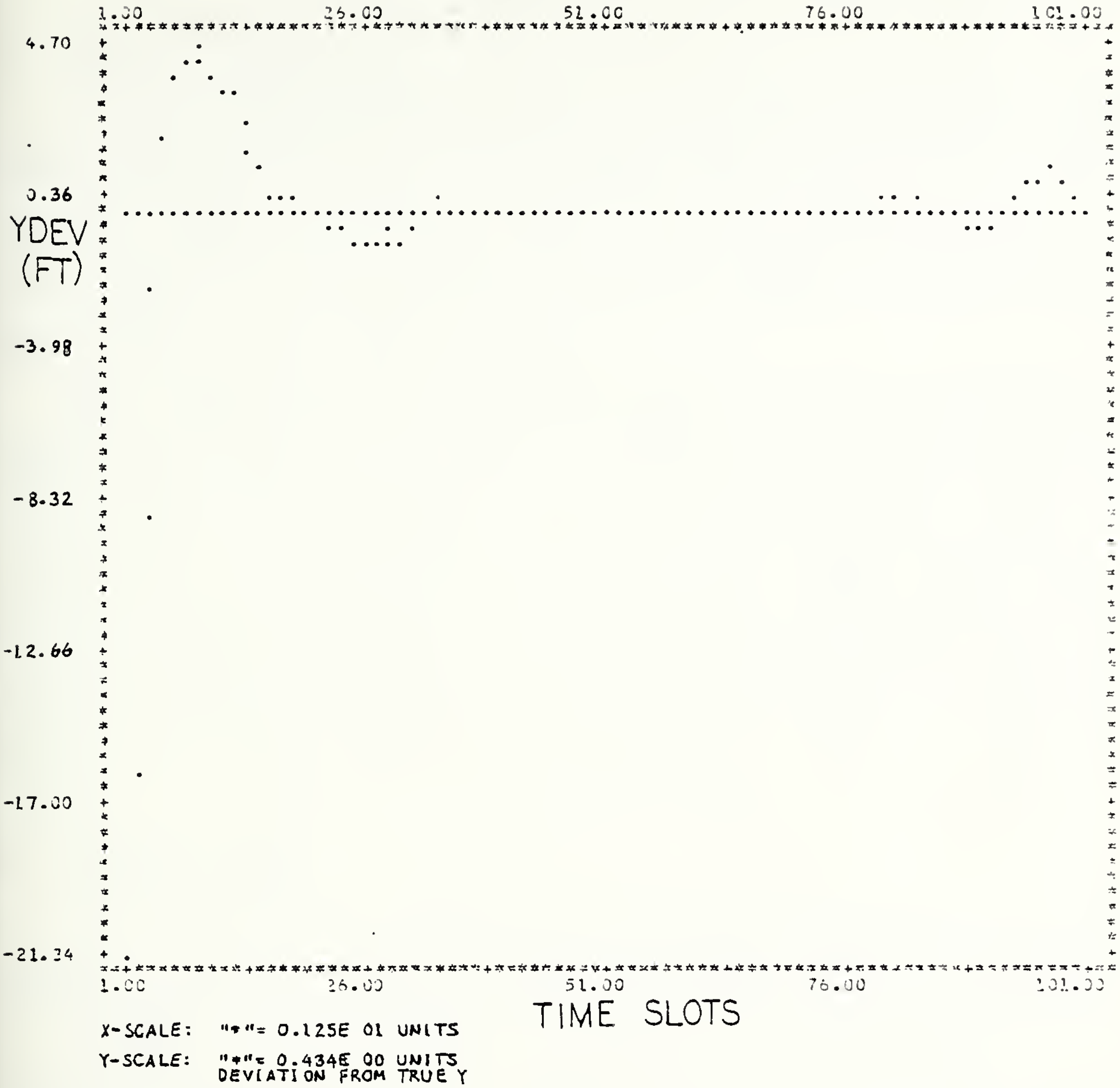


FIGURE 6: Y Deviation from True versus Time for a straight Running Track without Noise in the Area of a Single Array

B. SERIES TWO

In the second series of tests random white Gaussian noise was added to the transit time measurements for straight running tracks. Target velocities varied from 5.0 to 25.0 knots. Initial position errors ranged from 0 to 25 feet and initial velocity errors from 0 to 10 ft/sec.

Both the sequential and traditional routines performed well against the random noise. Estimate deviations from true track in X and Y positions were not observed greater than 3 feet after lock-on was acquired. Figures 7 and 8 depict the estimate deviation from true track for the sequential routine. Initial positions were 9 feet off in X, 13 feet in Y and 2 feet in Z.

C. SERIES THREE

In this series of tests maneuvers were added to the torpedo tracks and the filter was tested with and without noise.

Figure 9 is a geographical plot of a 25 knot target in a noiseless environment with 1/3-G turns at time slots 25 and 65. Figure 10 depicts the Z position versus the time of run. In the initialization of the filter X was off 13 feet and Y was off 21 feet. Figure 11 depicts the deviation by the estimate from the true track in the X direction, for the traditional inversion approach. Figure 12 shows the deviation for the same track using the sequential extended Kalman filter. When the target initiated the turn at time slots 25 and 65, the

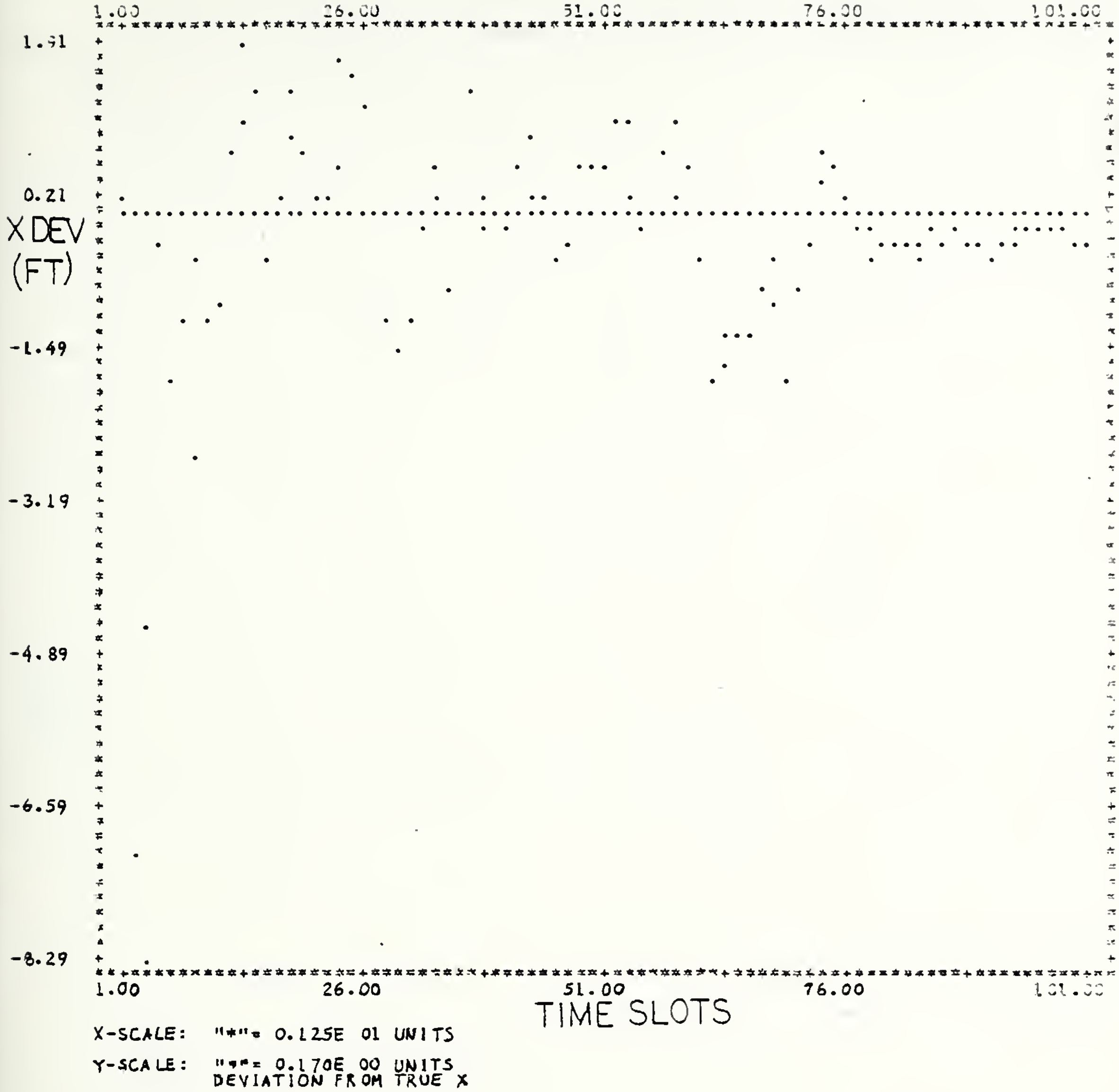


FIGURE 7: X Deviation from True versus Time for a straight Running Track with Noise in the Area of a Single Array -- Sequential Routine

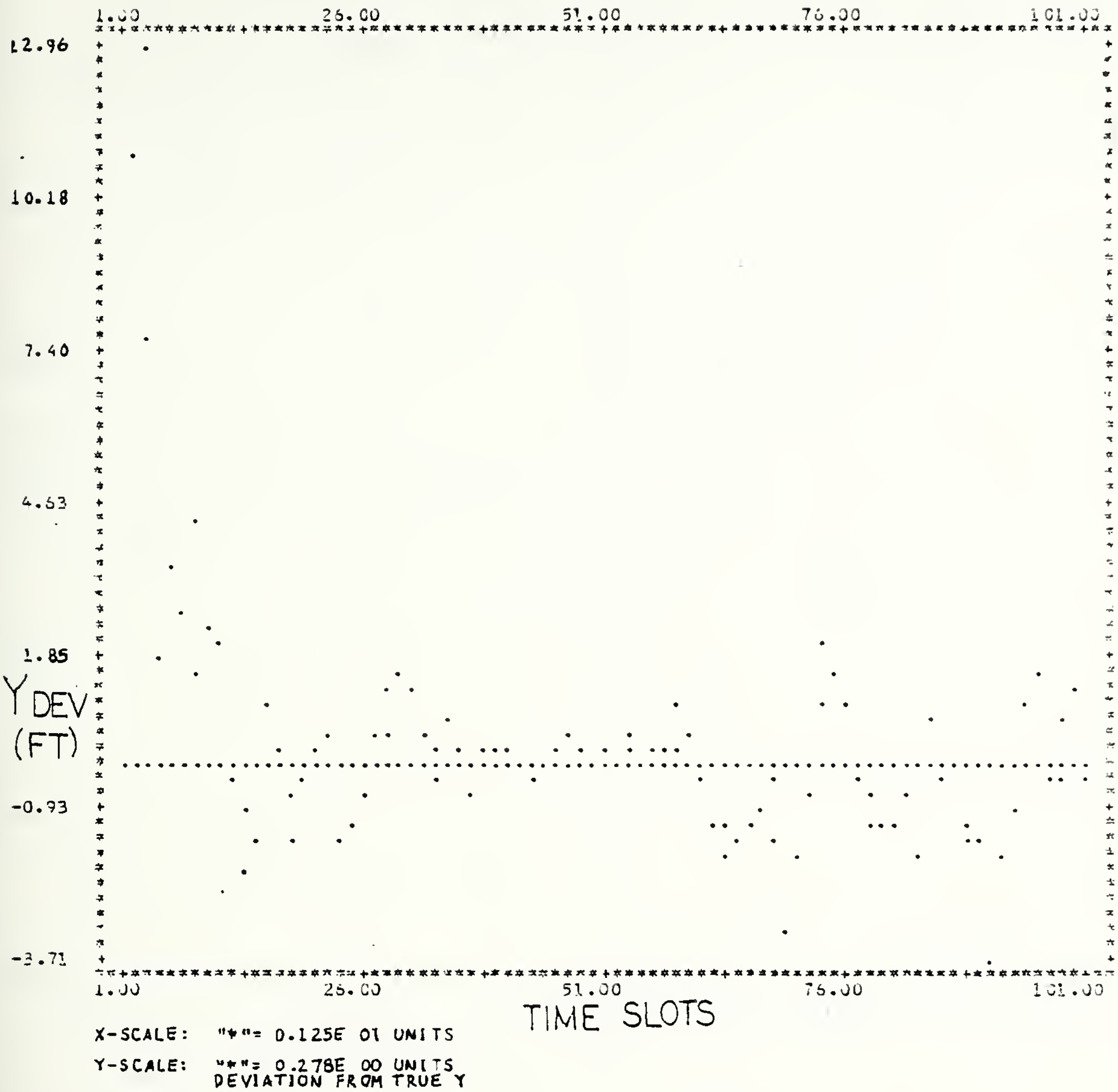


FIGURE 8: Y Deviation from True versus Time for a straight Running Track with Noise in the Area of a single Array -- Sequential Routine

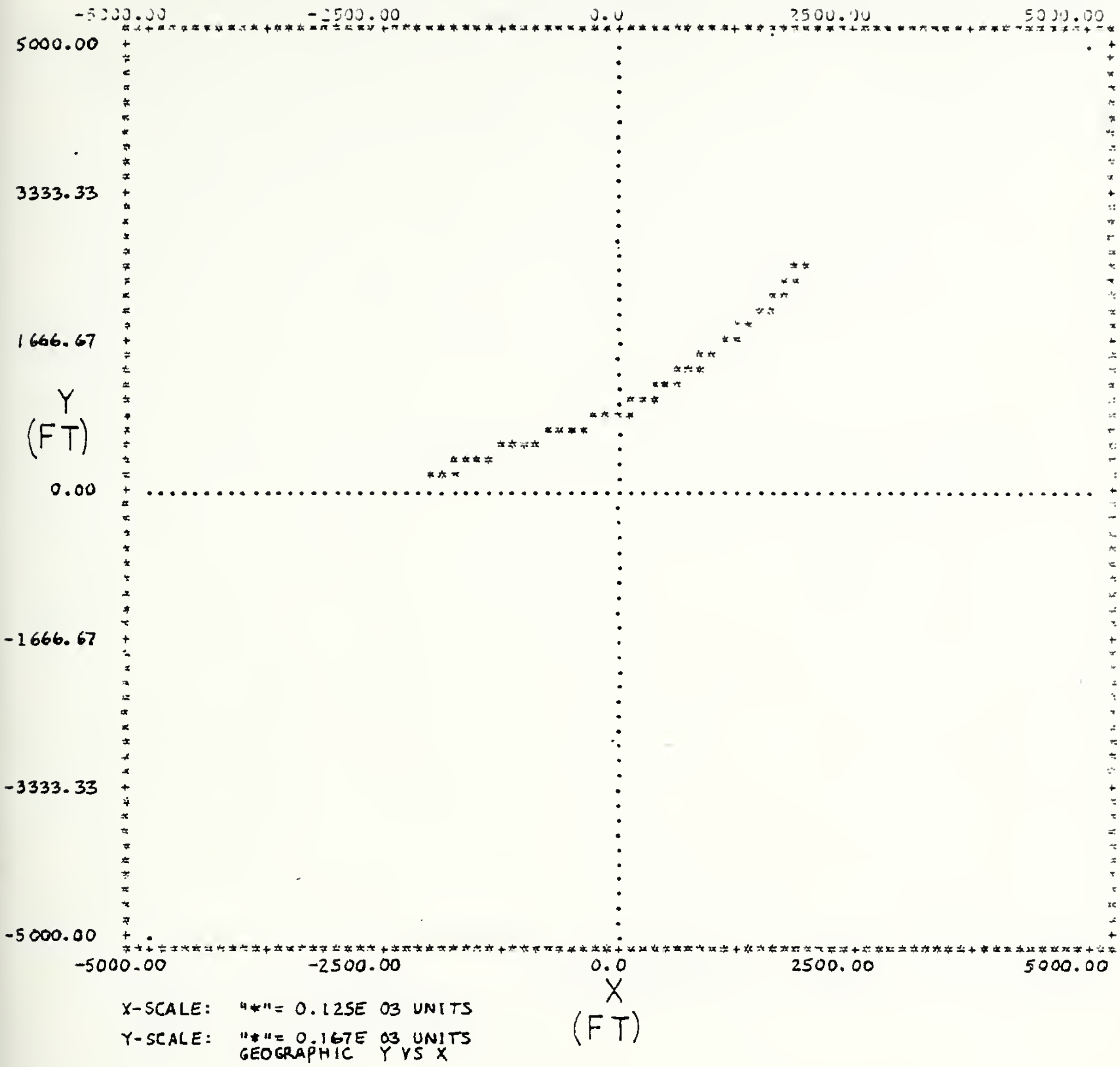


FIGURE 9: Geographic Plot of a Track with 1/3-G Turns in the Area of a Single Array

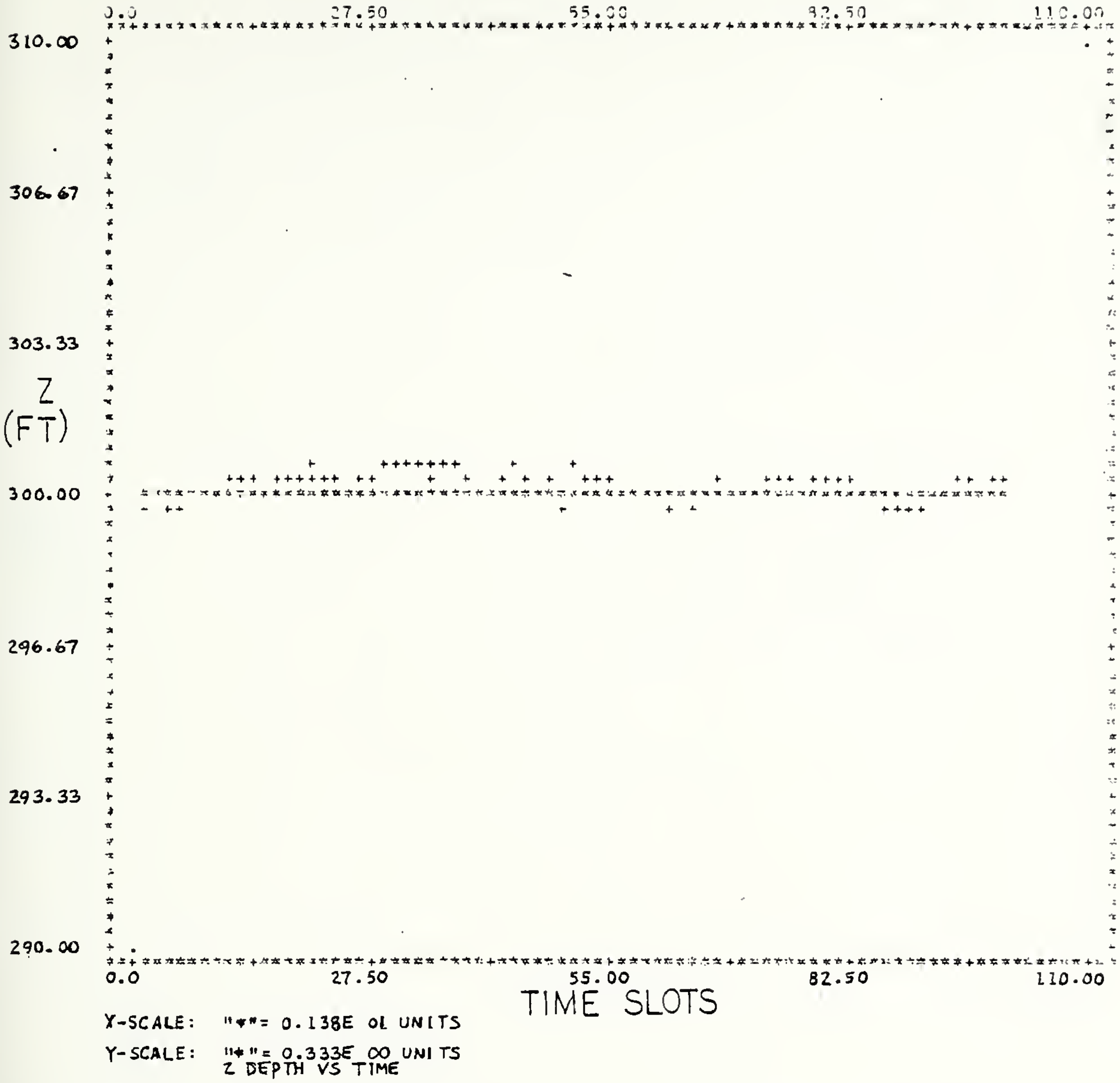


FIGURE 10: Z Position versus Time for a Track with 1/3-G Turns without Noise in the Area of a single Array

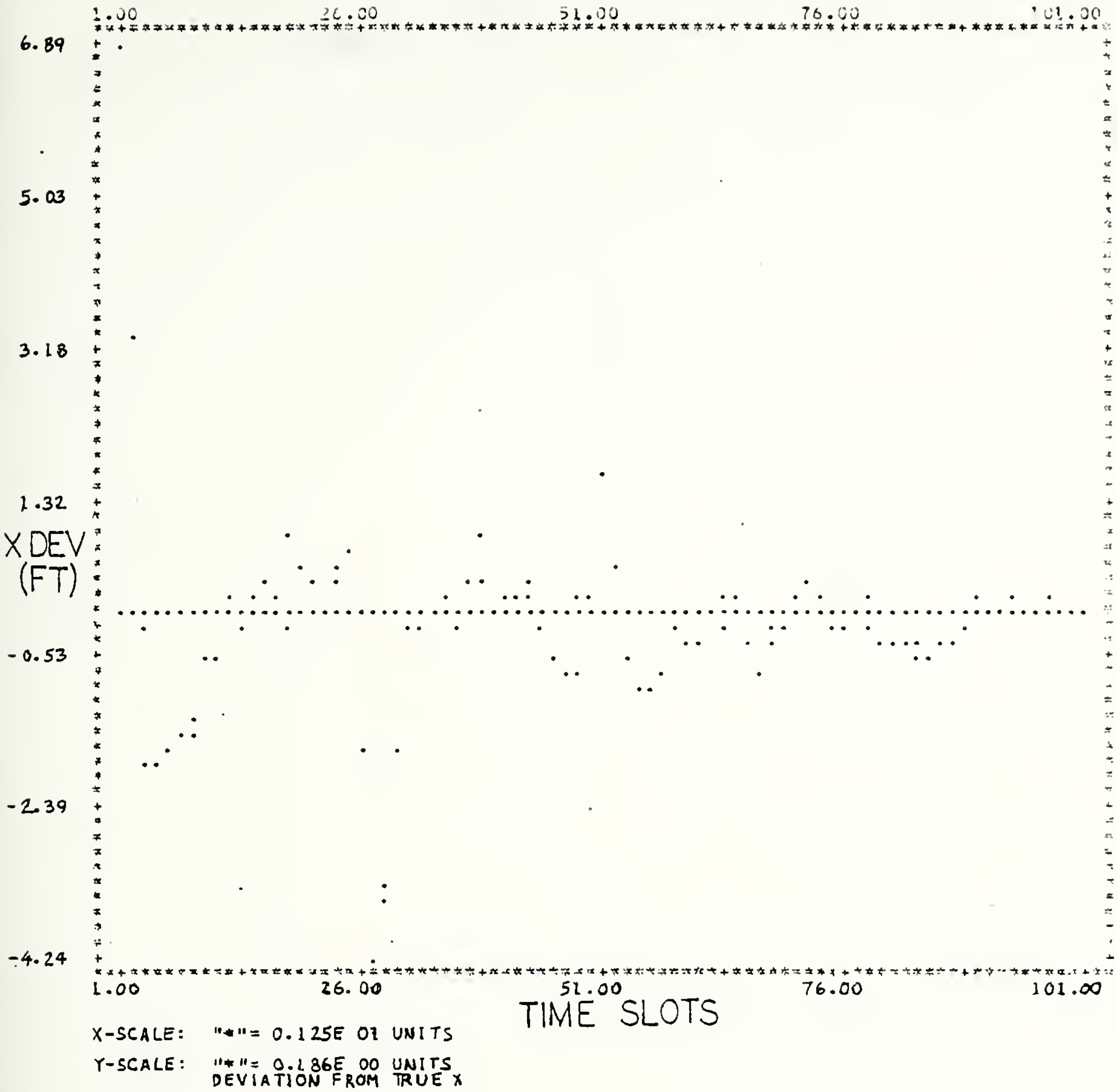


FIGURE 11: X Deviation from True for a Track with 1/3-G Turns without Noise in the Area of a Single Array -- Traditional Routine

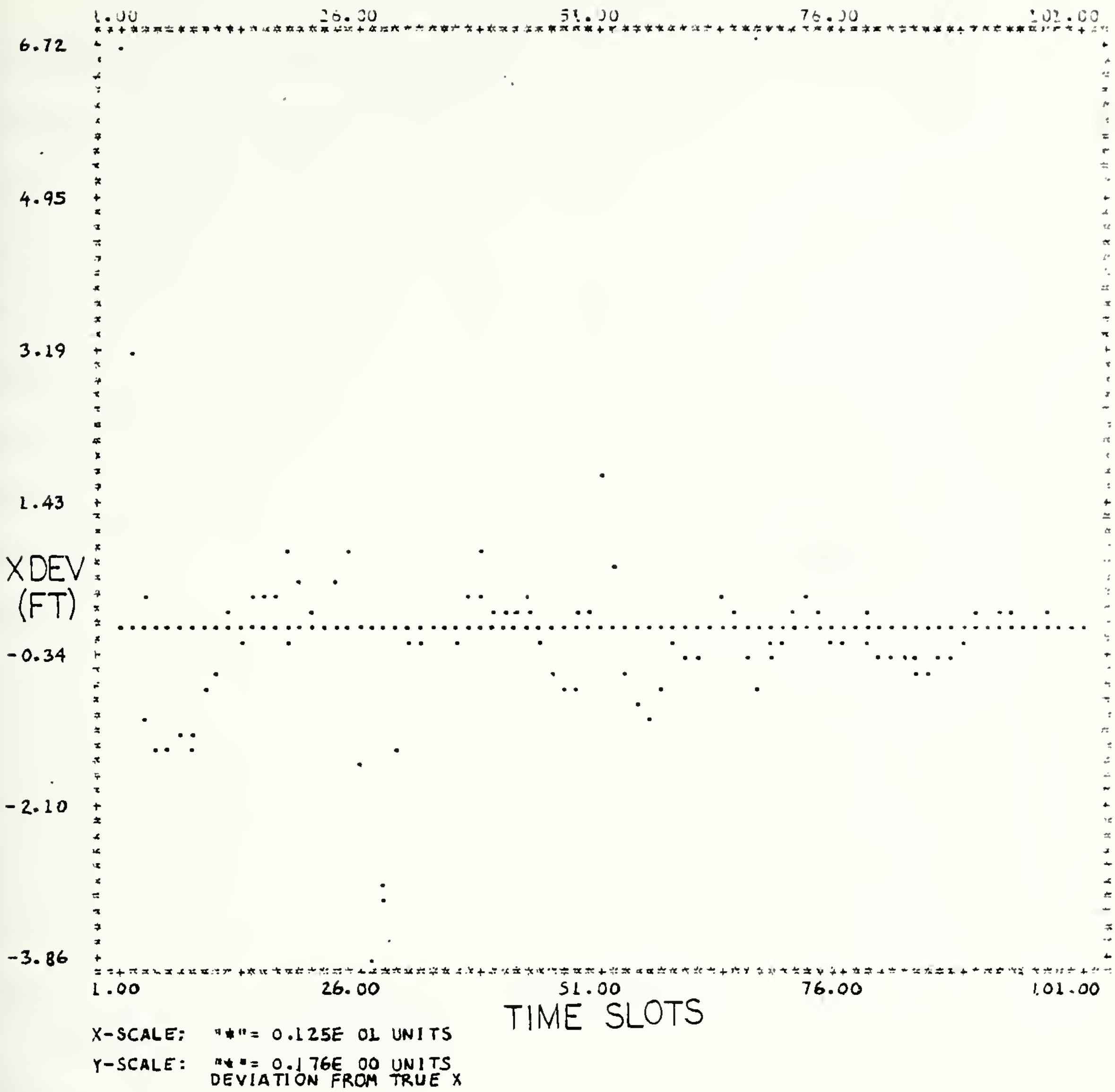


FIGURE 12: X Deviation from True for a Track with 1/3-G Turns without Noise in the Area of a Single Array -- Sequential Routine

covariance of estimation error and thus the filter gains increased allowing more weight to be placed on the incoming data (transit times). In this manner the filter was able to adjust the velocities in the X and Y directions and force the estimate back on track. Results from both routines were comparable as is shown in Figures 11 and 12. The routine using the sequential extended Kalman filter deviated a maximum of four feet at the turn and was back on track within 3 time slots, while the traditional filter routine had a slightly greater deviation and required 2 more time slots to acquire lock again. Figure 13 shows the deviation of the estimate from true track in the Y direction for this same run for the sequential filter and Figure 14 for the traditional filter. Both estimates deviated a maximum of 3 feet at the turn and re-acquired lock-on in 4 time slots with the sequential routine reacting slightly better to the turns.

Next, target tracks with 2/3 and 1-G turns were tested. Figure 15 is a geographic plot of this track with a 1-G turn at time slot 25 and a 2/3-G turn at time slot 65. Figure 16 depicts the Z position vs time for this track. With an initialization error of 13 feet in X and 21 feet in Y, the maximum deviation after lock-on was again at the turn points. Figure 17 depicts this deviation from true track in the X direction, for the sequential routine with a maximum deviation of approximately 5.5 feet for the 1-G turn. Lock-on was reacquired in a maximum of 4 time slots. Figure 18 shows the

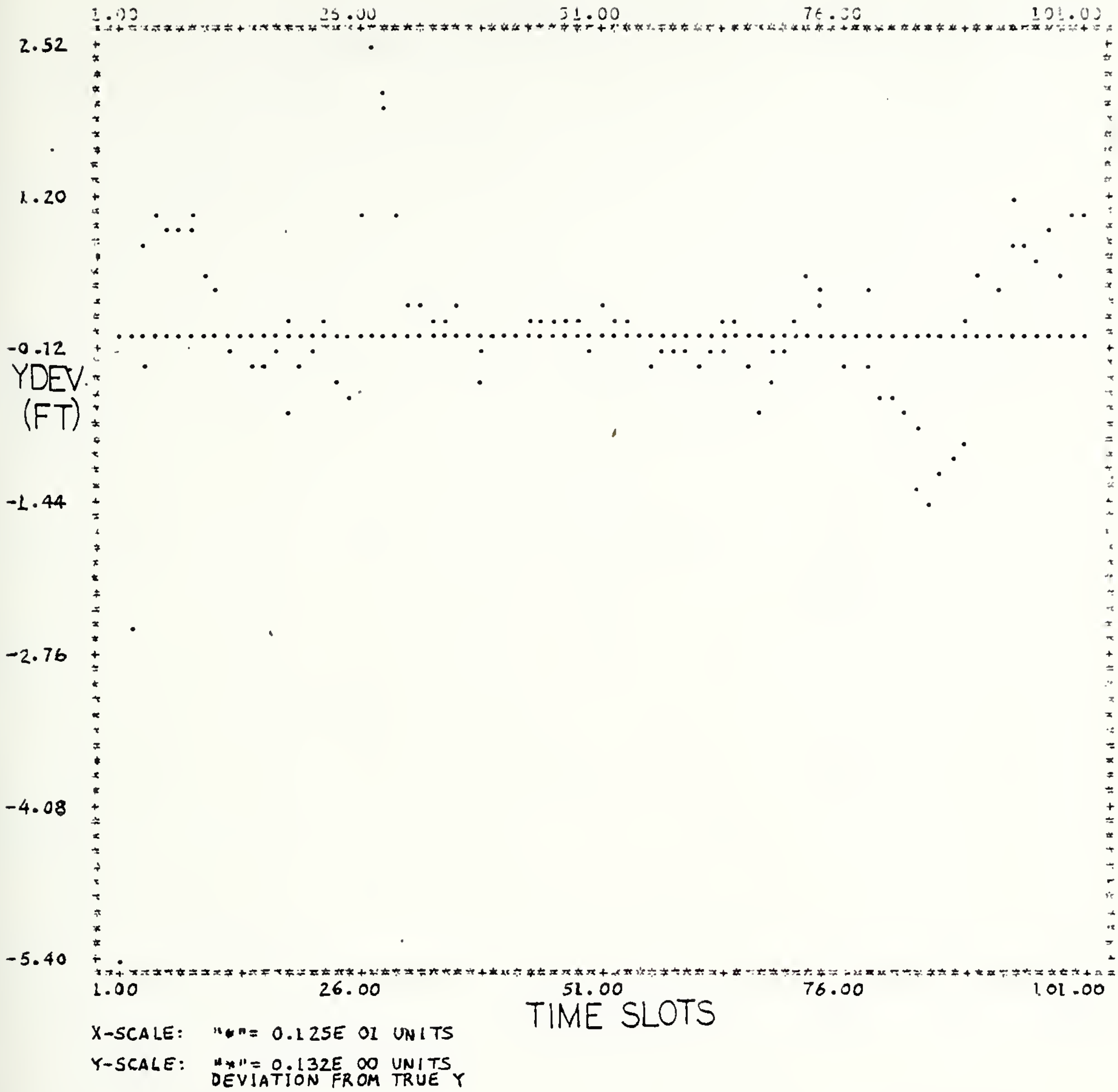


FIGURE 13: Y Deviation from True for a Track with 1/3-G Turns without Noise in the Area of a Single Array -- Sequential Routine

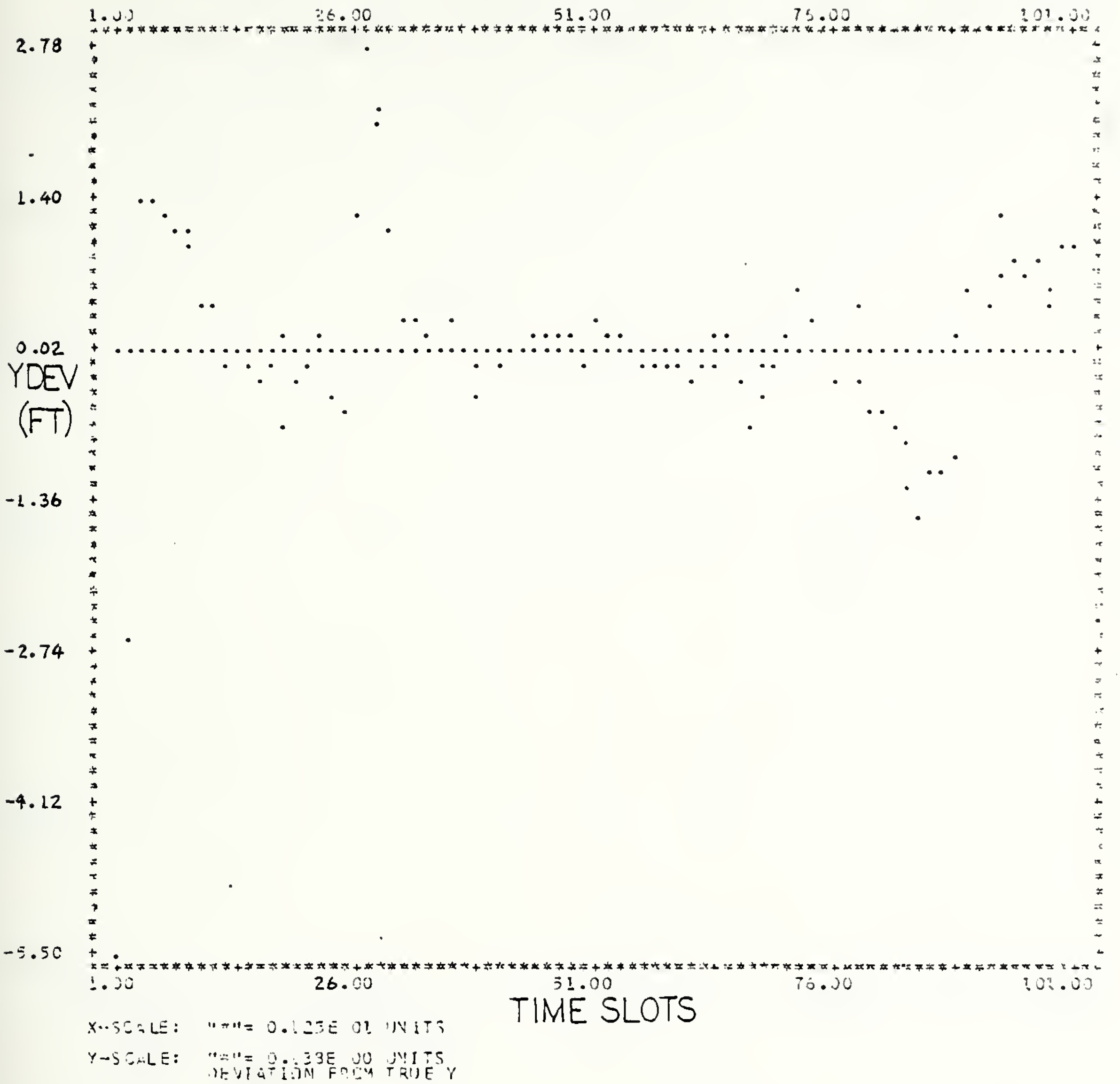


FIGURE 14: Y Deviation from True for a Track with 1/3-G Turns without Noise in the Area of a Single Array -- Traditional Routine

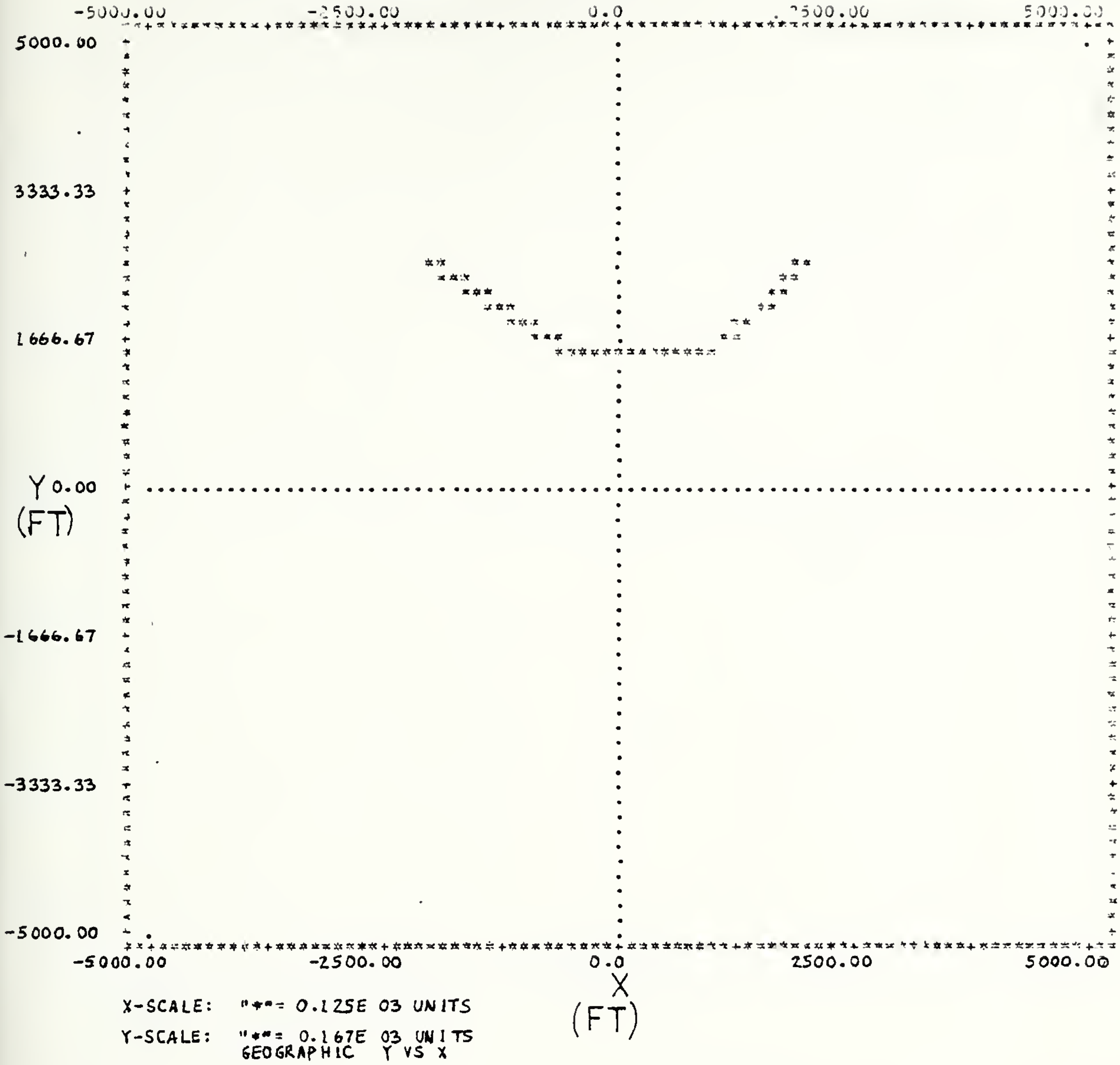


FIGURE 15: Geographic Plot for a Track with 2/3-G and 1-G Turns in the Area of a Single Array

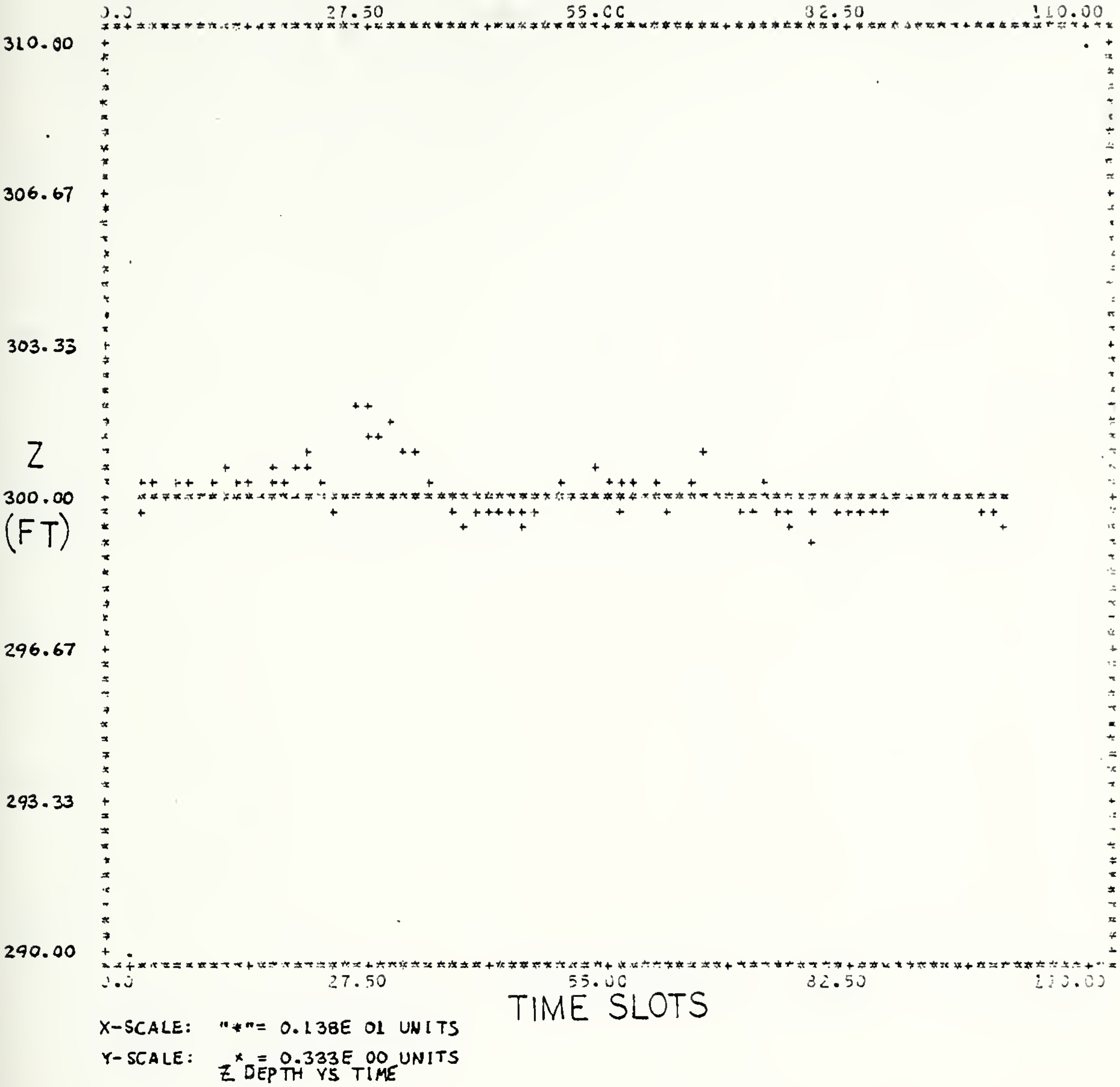


FIGURE 16: Z Position versus Time for a Track with 2/3-G and 1-G Turns without Noise in the Area of a Single Array

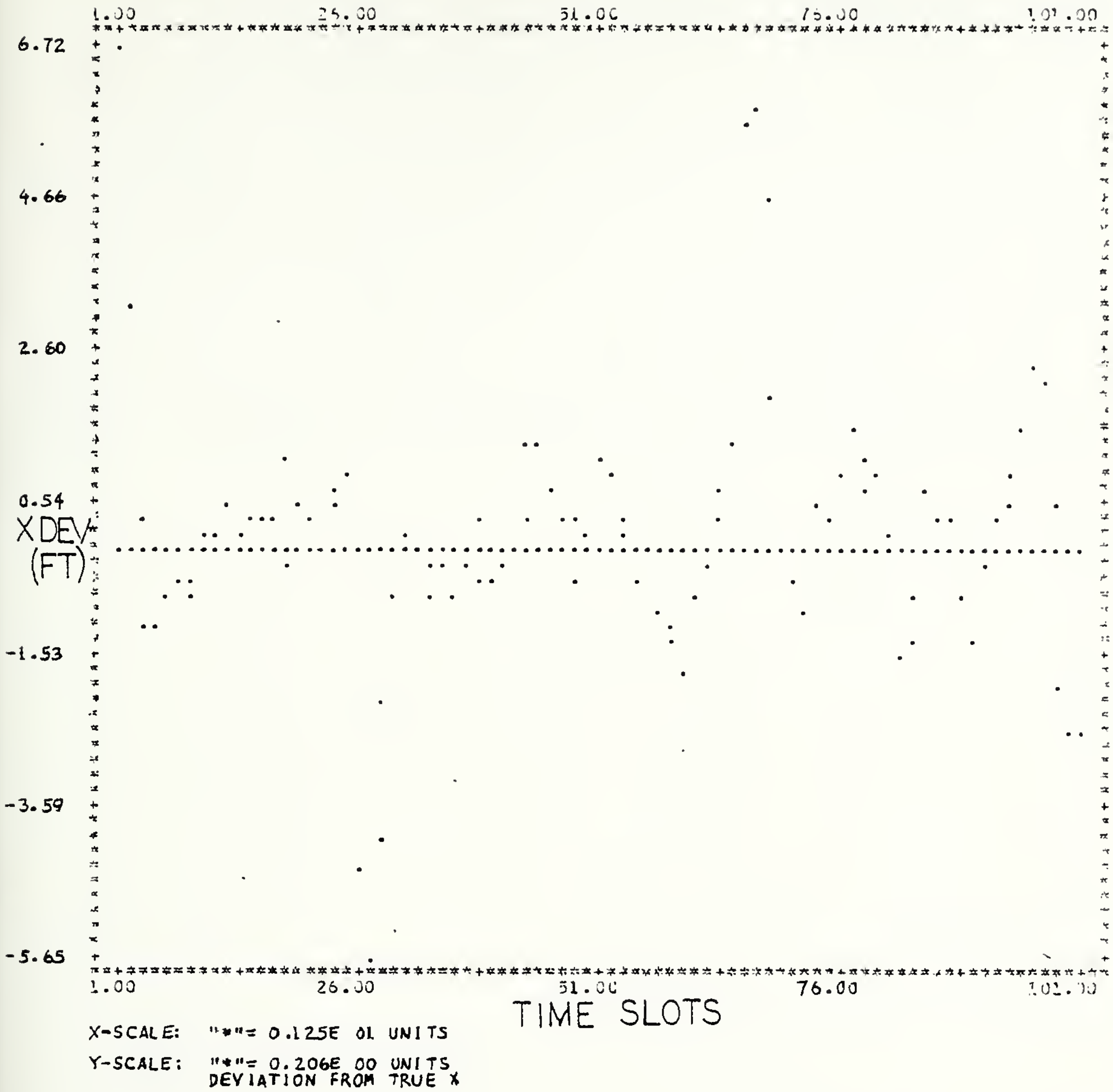


FIGURE 17: X Deviation from True for a Track with 2/3-G and 1-G Turns without Noise in the Area of a Single Array -- Sequential Routine

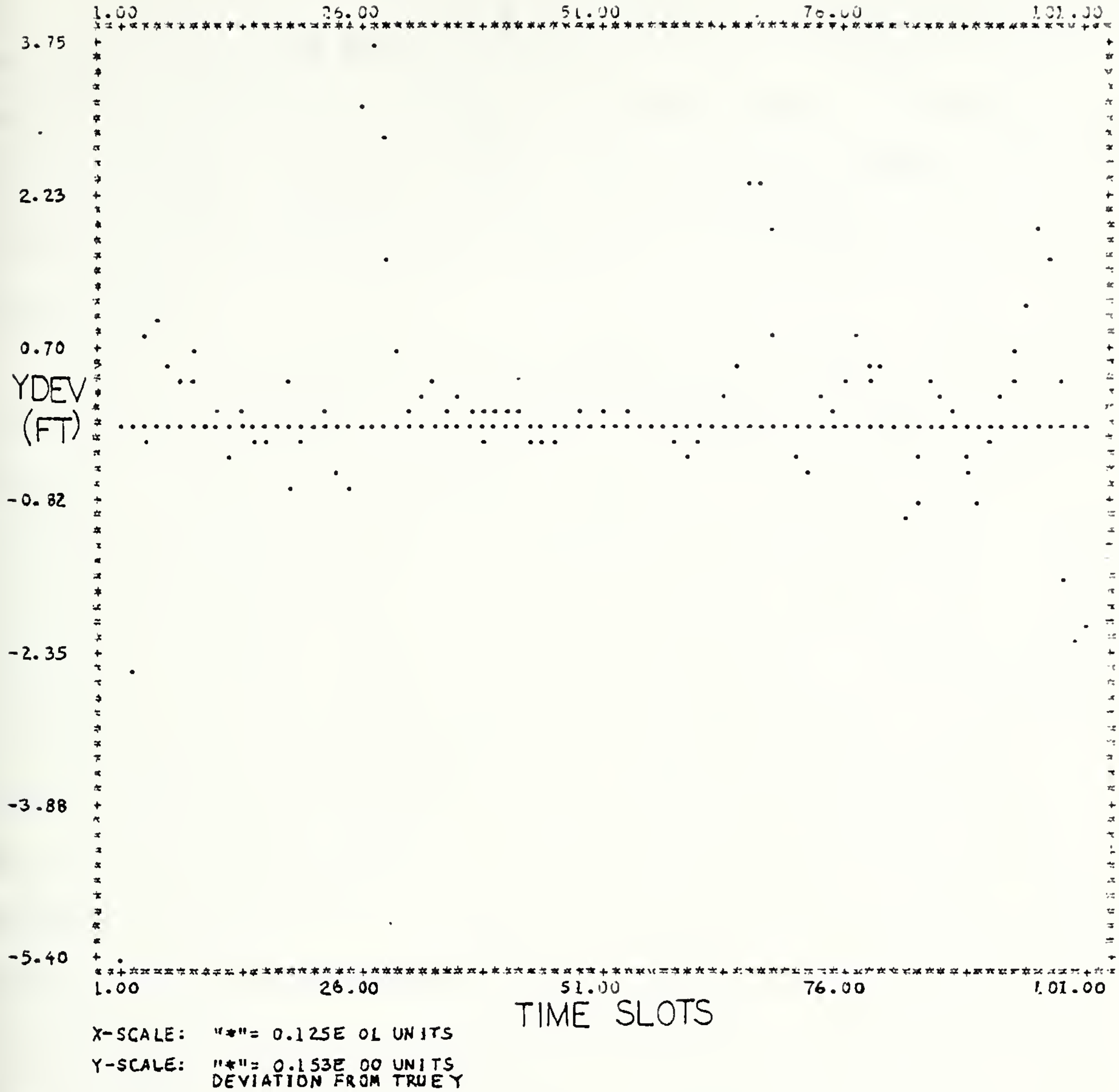


FIGURE 18: Y Deviation from True for a Track with 2/3-G and 1-G Turns without Noise in the Area of a Single Array -- Sequential Routine

deviation from true in the Y direction with a maximum deviation after lock-on for the 1-G turn of under 4 feet. Runs with the traditional extended Kalman filter routine showed slightly larger deviations with more reaction time required to re-acquire lock-on.

In the final tests for this series, zero mean, white Gaussian noise was added to corrupt the observed transit times. Figure 19 depicts the estimate's deviation from true track in the X direction using the traditional filter routine. The target experienced 1/3-G turns at time slots 25 and 65. Figure 20 shows the same track in noise with the same filter parameters when run with the sequential extended Kalman filter routine. Comparison shows that the latter had approximately 2 feet less maximum deviation and that while the traditional routine had trouble re-acquiring lock-on after the turn, the sequential routine regained track within 4 time slots. Figures 21 and 22 depict the same type of behavior for deviation in the Y direction. Maximum deviation again was slightly less and lock-on more efficiently re-acquired using the sequential routine.

In the cases involving more radical maneuvers, the sequential routine continued to perform better against the modeled noise. Figure 23 shows the estimate's deviation from true track in the X direction for a run with a 1-G maneuver at time slot 25 and a 2/3-G maneuver at time 65 in noise. This graph for the traditional routine shows maximum deviation at

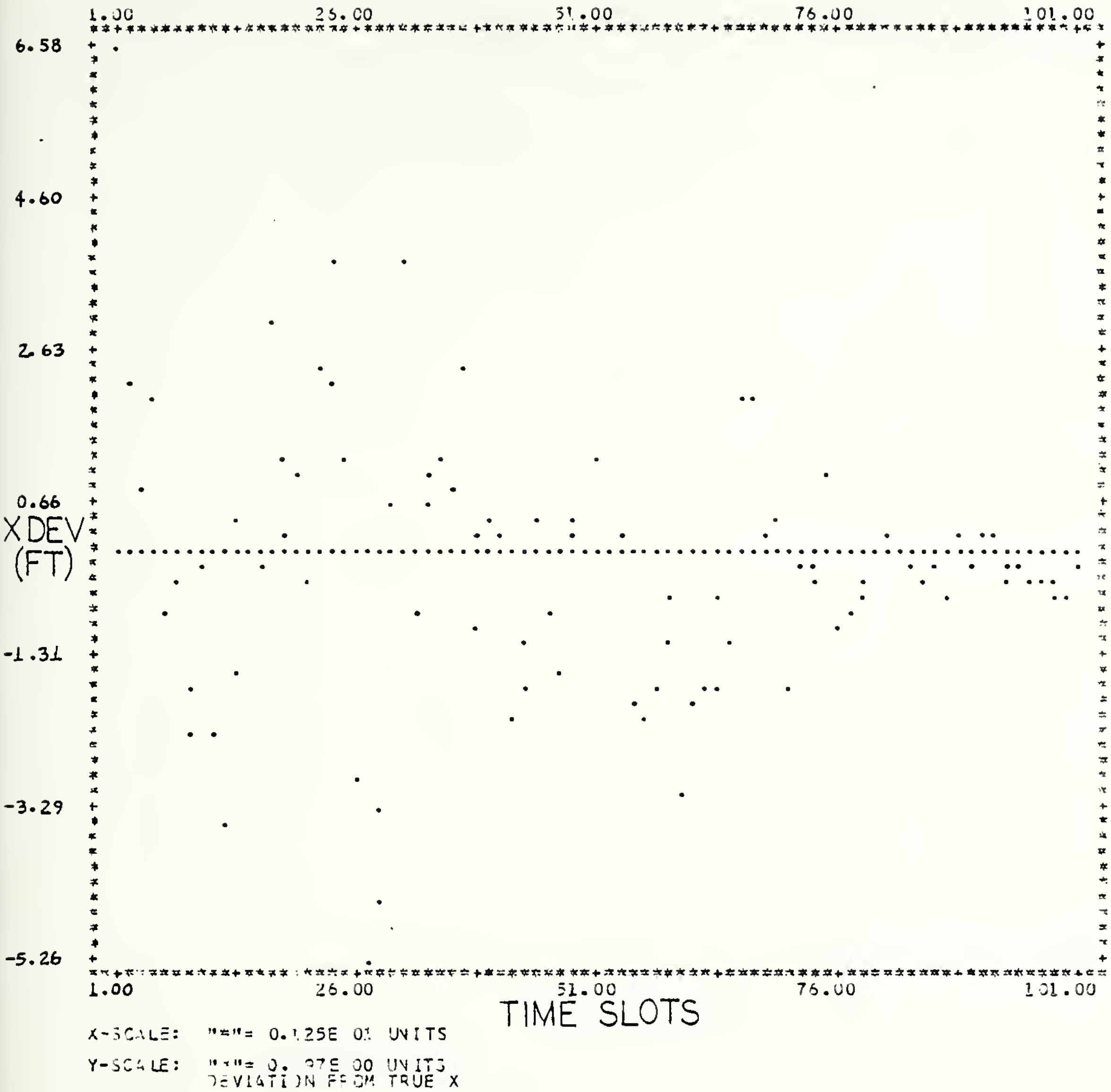


FIGURE 19: X Deviation from True for a Track with 1/3-G Turns with Noise in the Area of a Single Array -- Traditional Routine

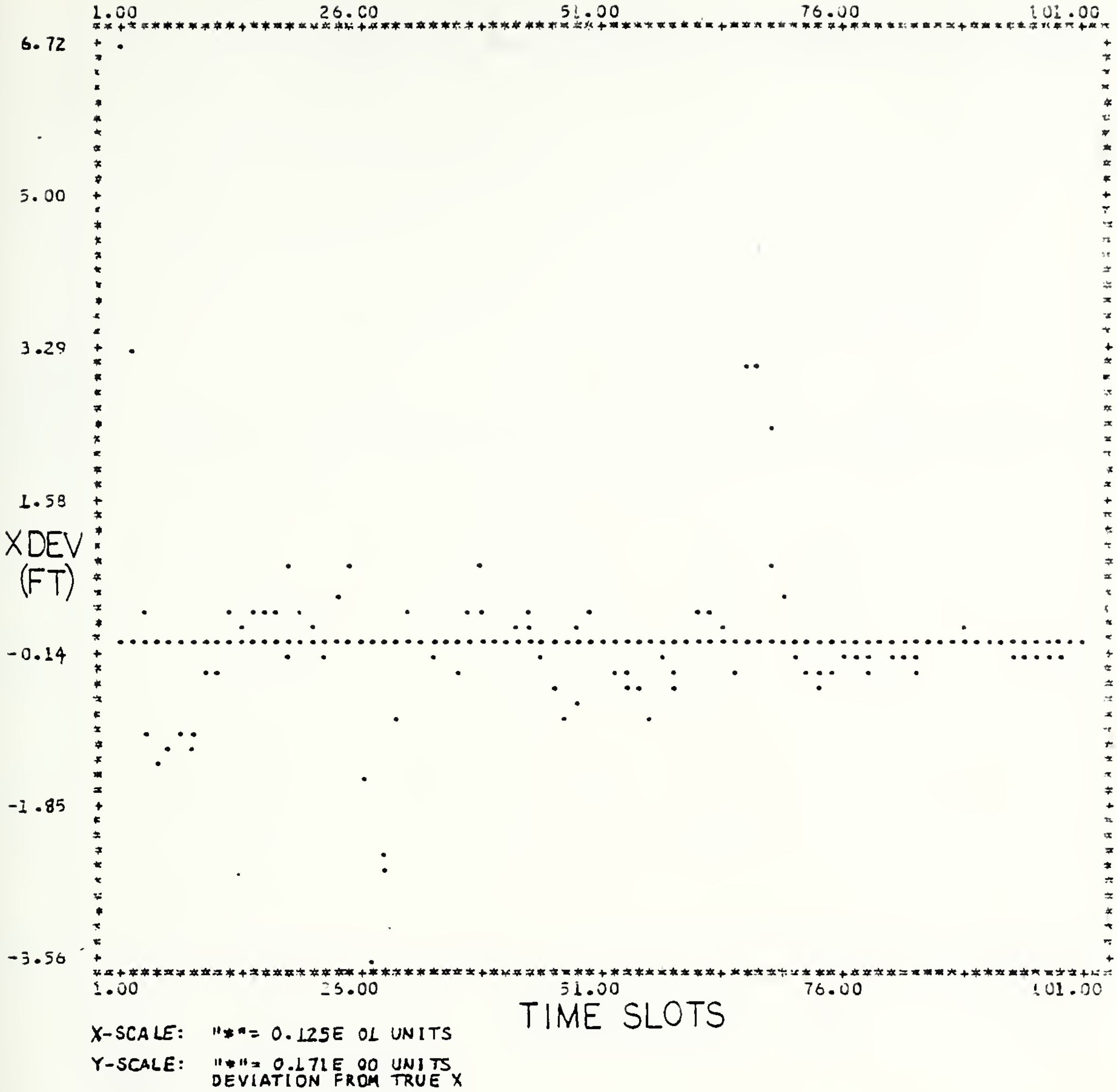


FIGURE 20: X Deviation from True for a Track with 1/3-G Turns with Noise in the Area of a Single Array -- Sequential Routine

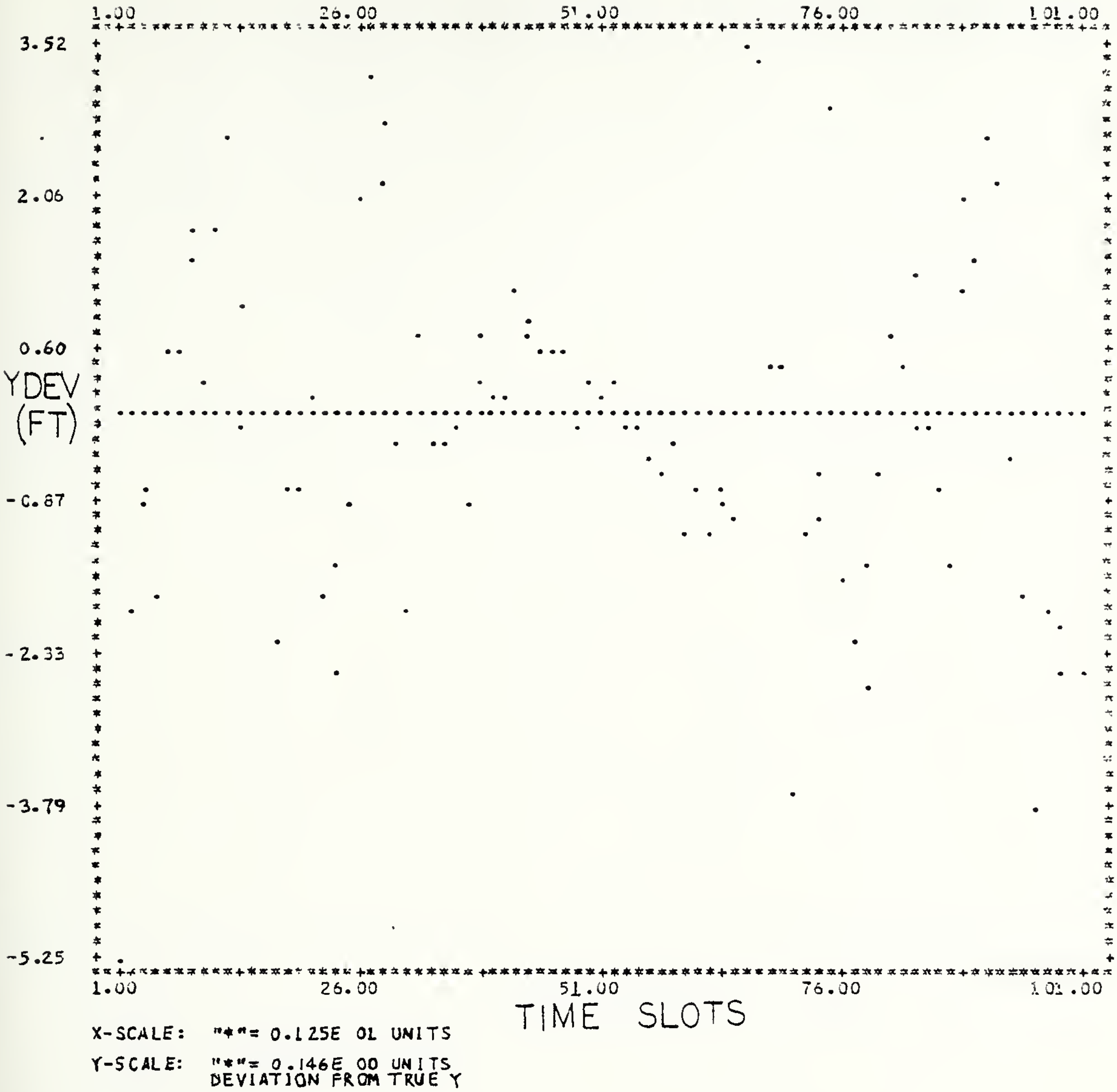


FIGURE 21: Y Deviation from True for a Track with 1/3-G Turns with Noise in the Area of a Single Array -- Traditional Routine

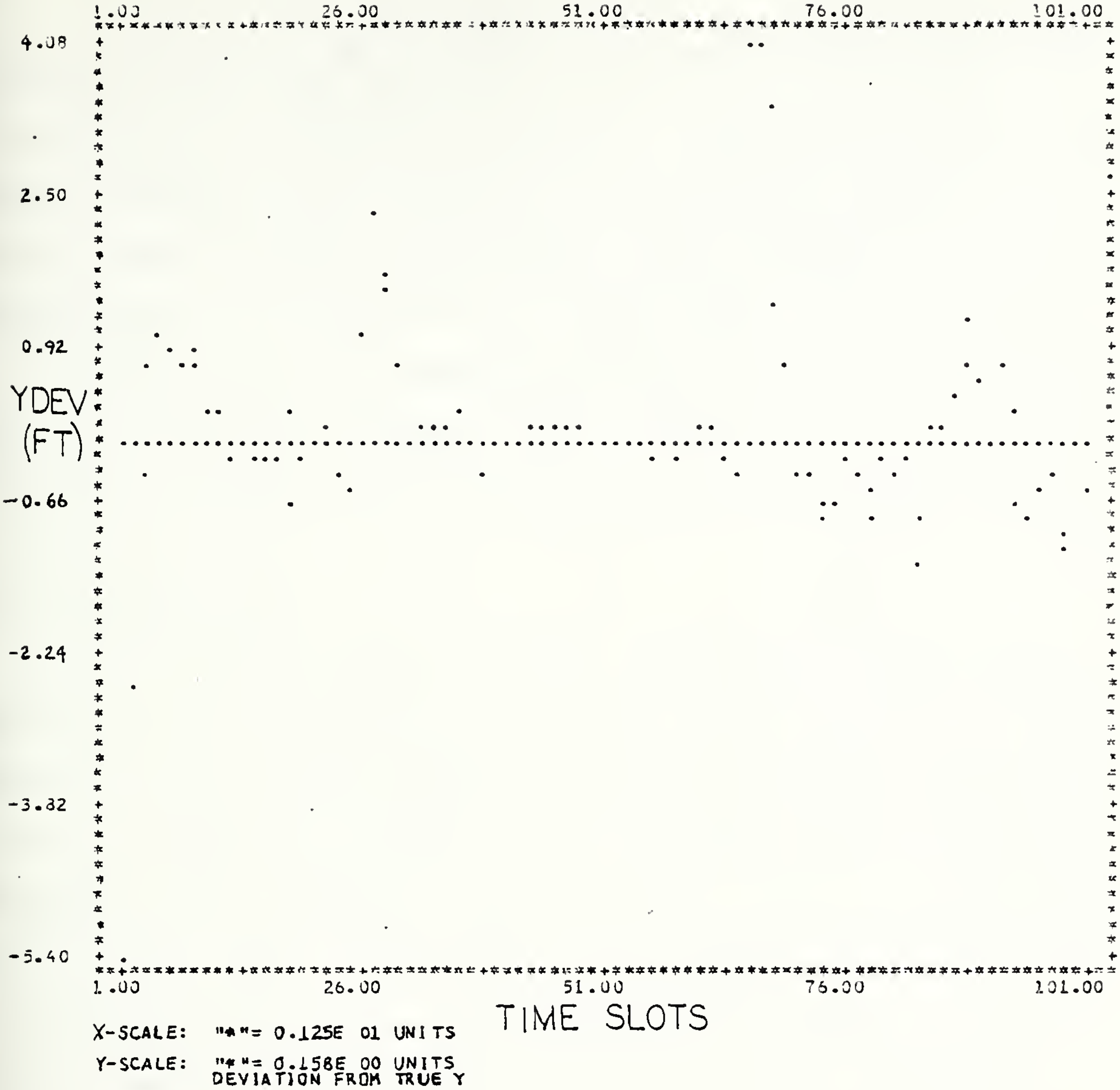


FIGURE 22: Y Deviation from True for a Track with 1/3-G Turns with Noise in the Area of a Single Array -- Sequential Routine

the 1-G turn of approximately 9 feet and the difficulty the filter had in re-acquiring lock-on. Figure 24, which is the same track and filter parameters run with the sequential routine, shows a marked improvement in both maximum deviation and efficiency in re-acquiring the track after a turn in noise.

Figures 25 and 26 demonstrate again the better performance of the sequential extended filter routine regarding deviation in the Y direction for the more radically maneuvering track.

D. SERIES FOUR

In the fourth series of tests, the target speeds were increased to the 40 to 50 knot range in order to bring the simulation in line with speeds actually experienced on the range. In a noiseless environment, both routines showed similar performance with a maximum deviation from true track during the run under 2 feet. With random noise added this deviation increased and Figure 27 is a plot of the deviation in the X direction using the traditional routine. When compared to Figure 28 which is the same track using the sequential routine with the same parameters, a marked improvement in maximum deviation and lock-on efficiency is noticeable. Better performance by the sequential routine was also present regarding deviation in the Y direction.

When 1/3-G turns were introduced, the routines performed similarly in a noiseless environment. With random noise and maneuvers the routines were again comparable with the traditional routine performing slightly better as is shown in

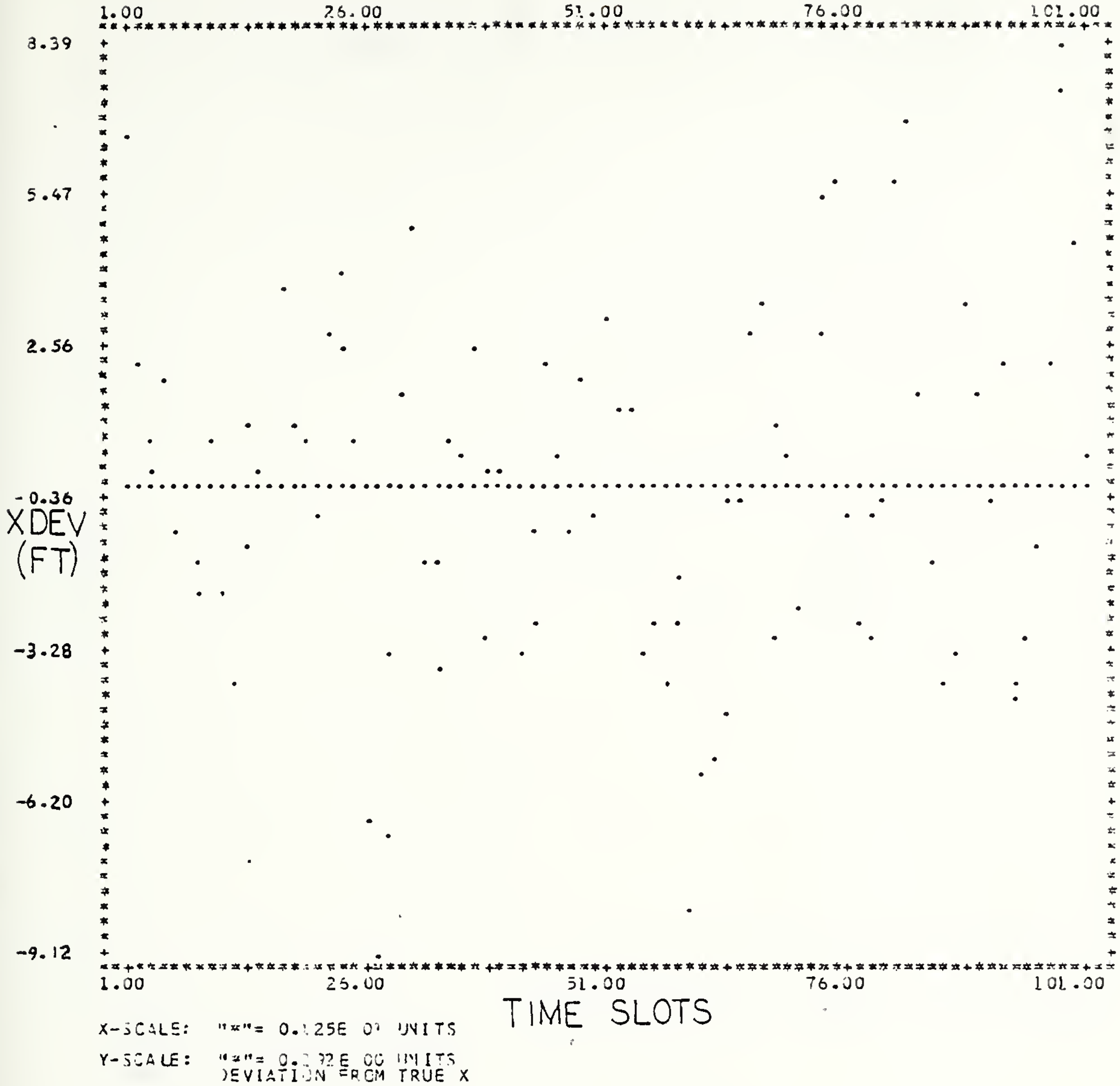


FIGURE 23: X Deviation from True for a Track with 2/3-G and 1-G Turns with Noise in the Area of a Single Array -- Traditional Routine

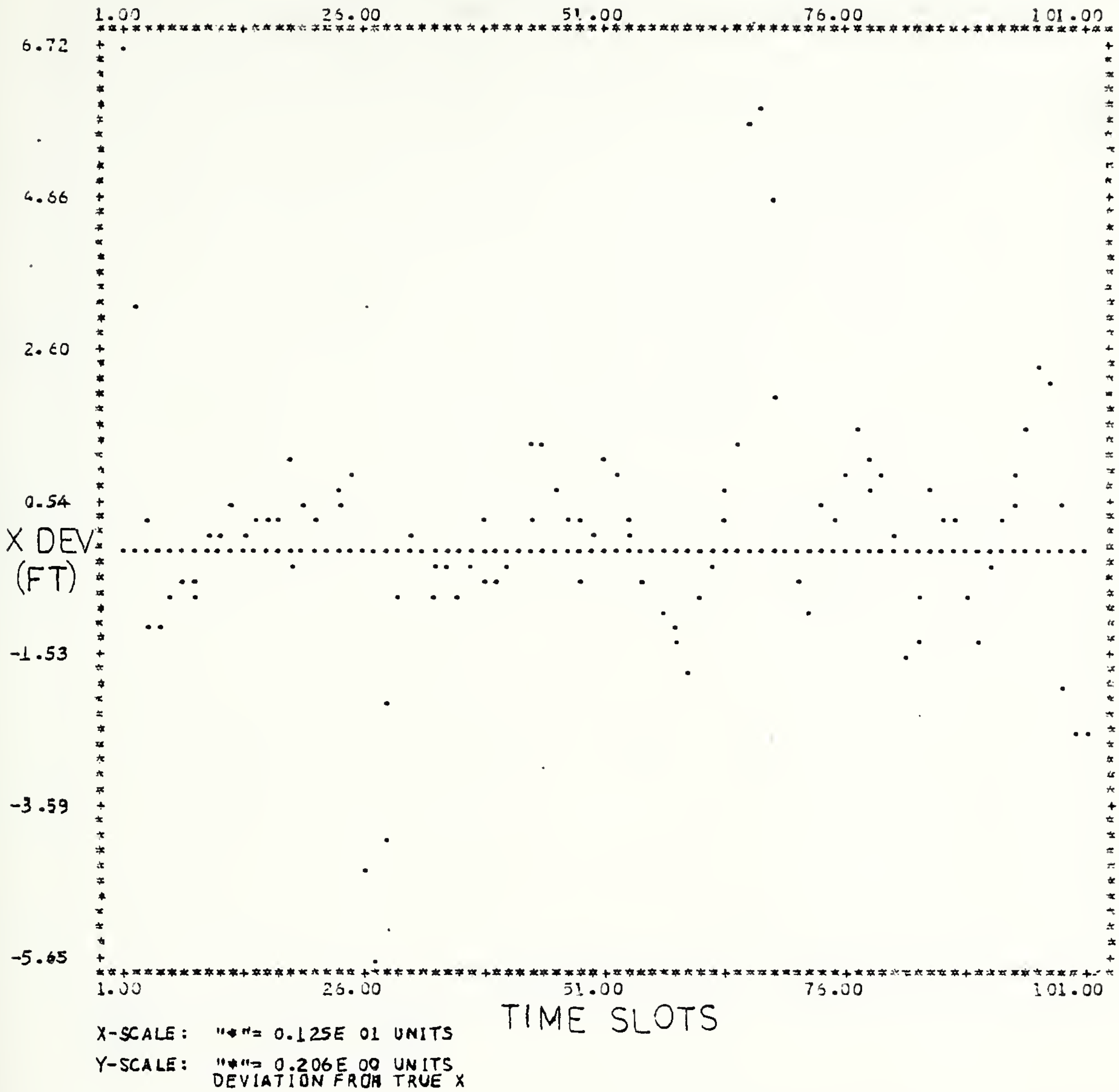


FIGURE 24: X Deviation from True for a Track with 2/3-G and 1-G Turns with Noise in the Area of a Single Array -- Sequential Routine

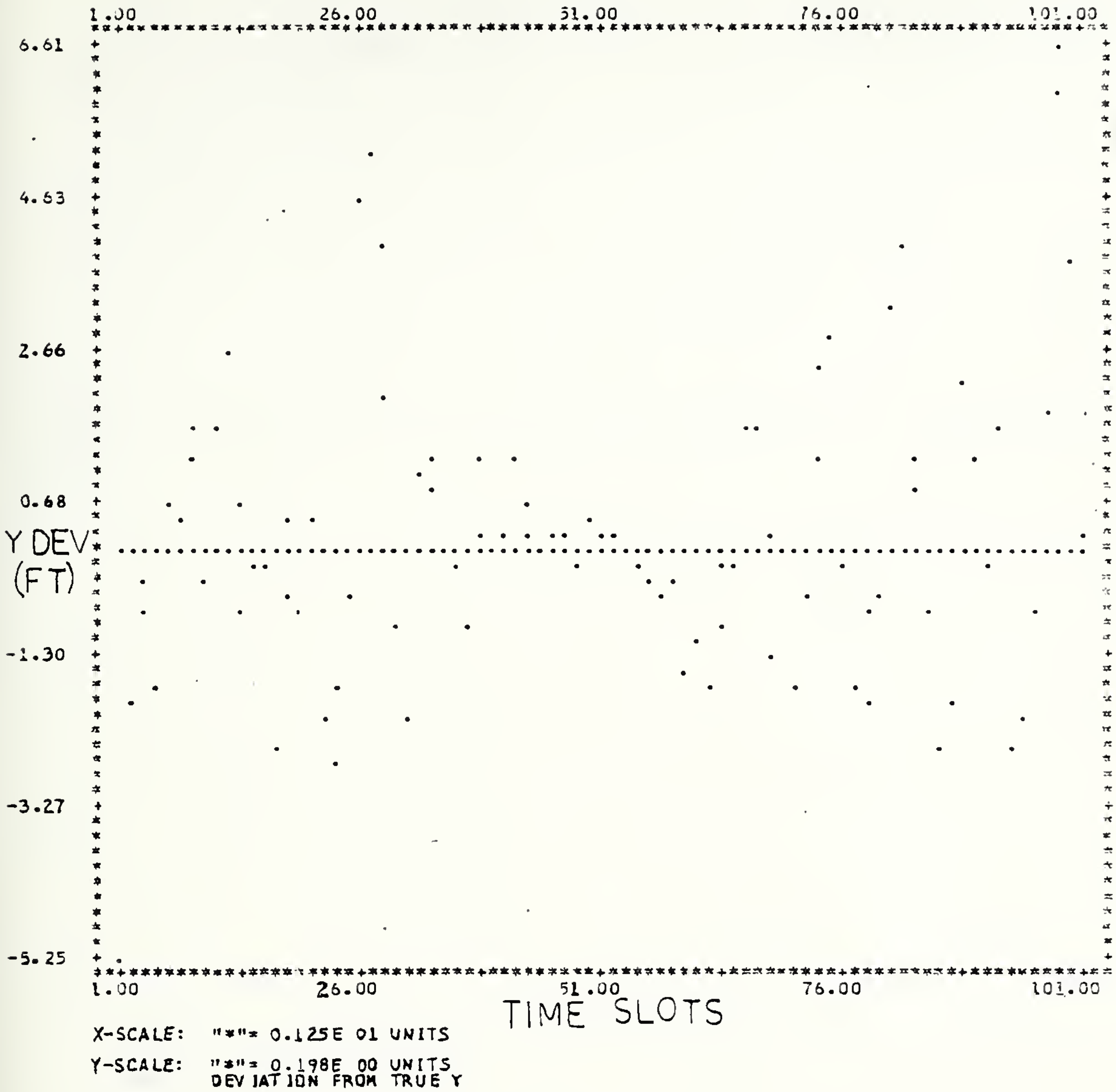


FIGURE 25: Y Deviation from True for a Track with 2/3-G and 1-G Turns in the Area of a Single Array with Noise --Traditional Routine

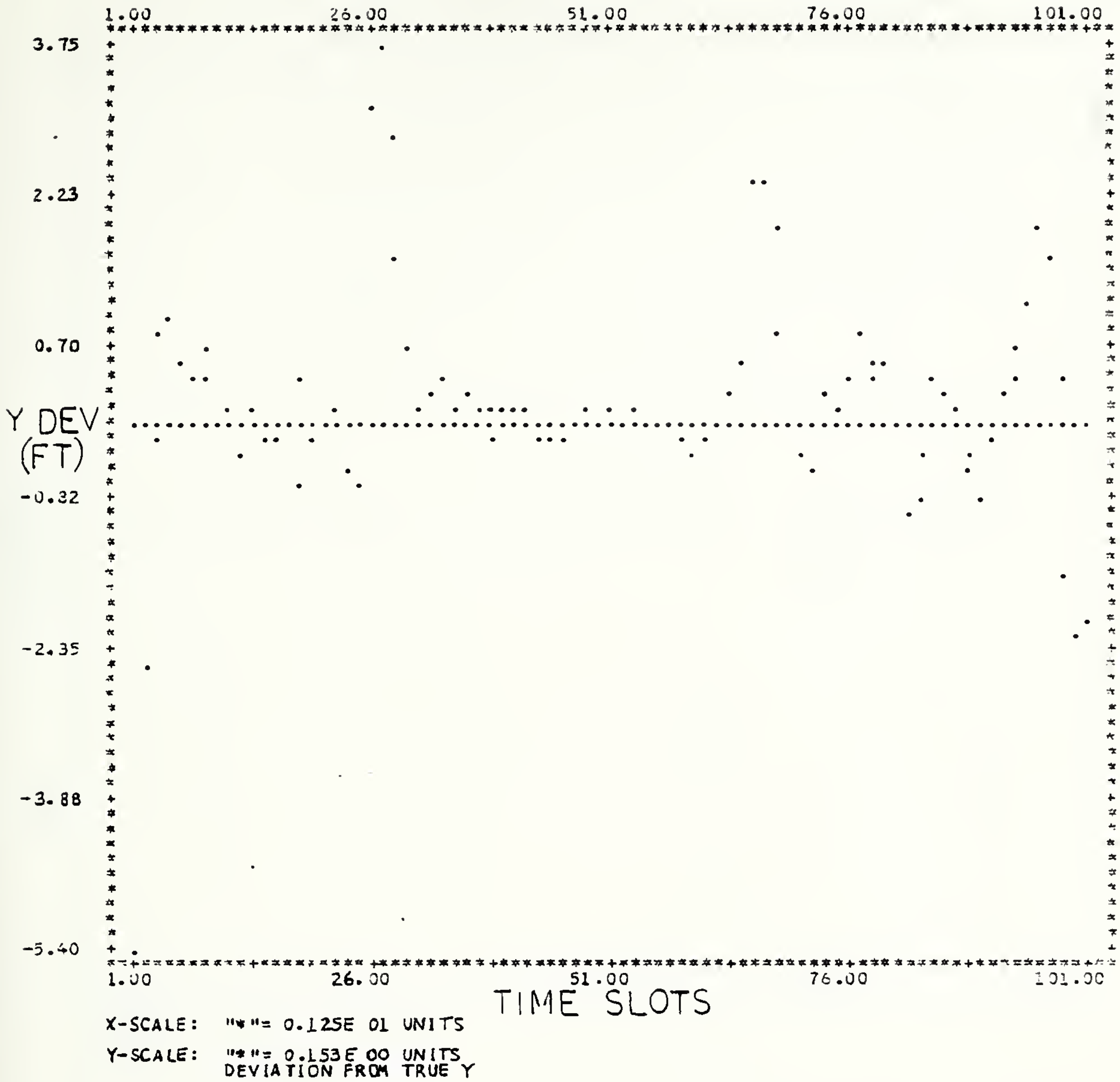


FIGURE 26: Y Deviation from True for a Track with 2/3-G and 1-G Turns with Noise in the Area of a Single Array -- Sequential Routine

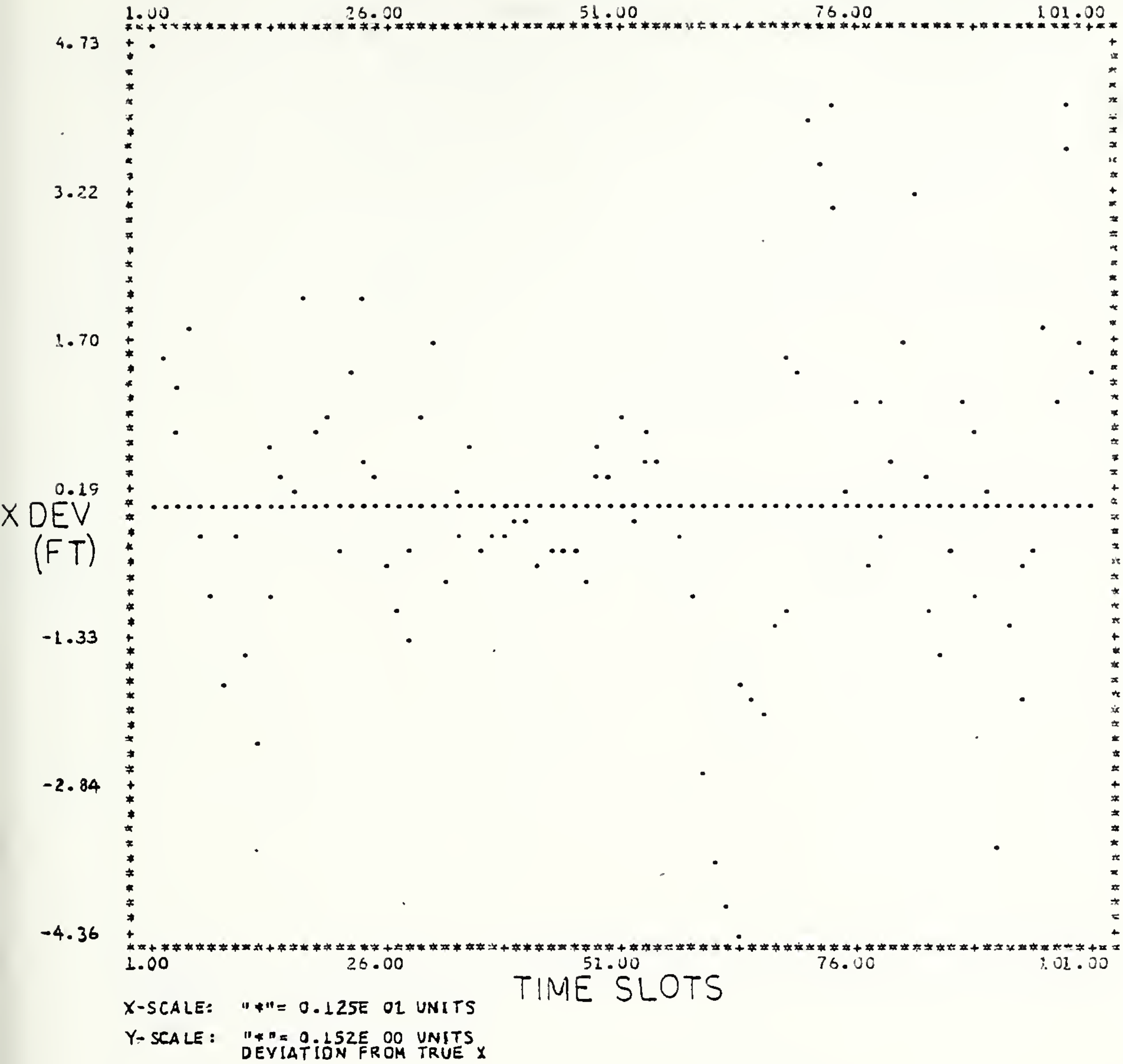


FIGURE 27: X Deviation from True for a Straight Running 42 Knot Target with Noise in the Area of a Single Array -- Traditional Routine

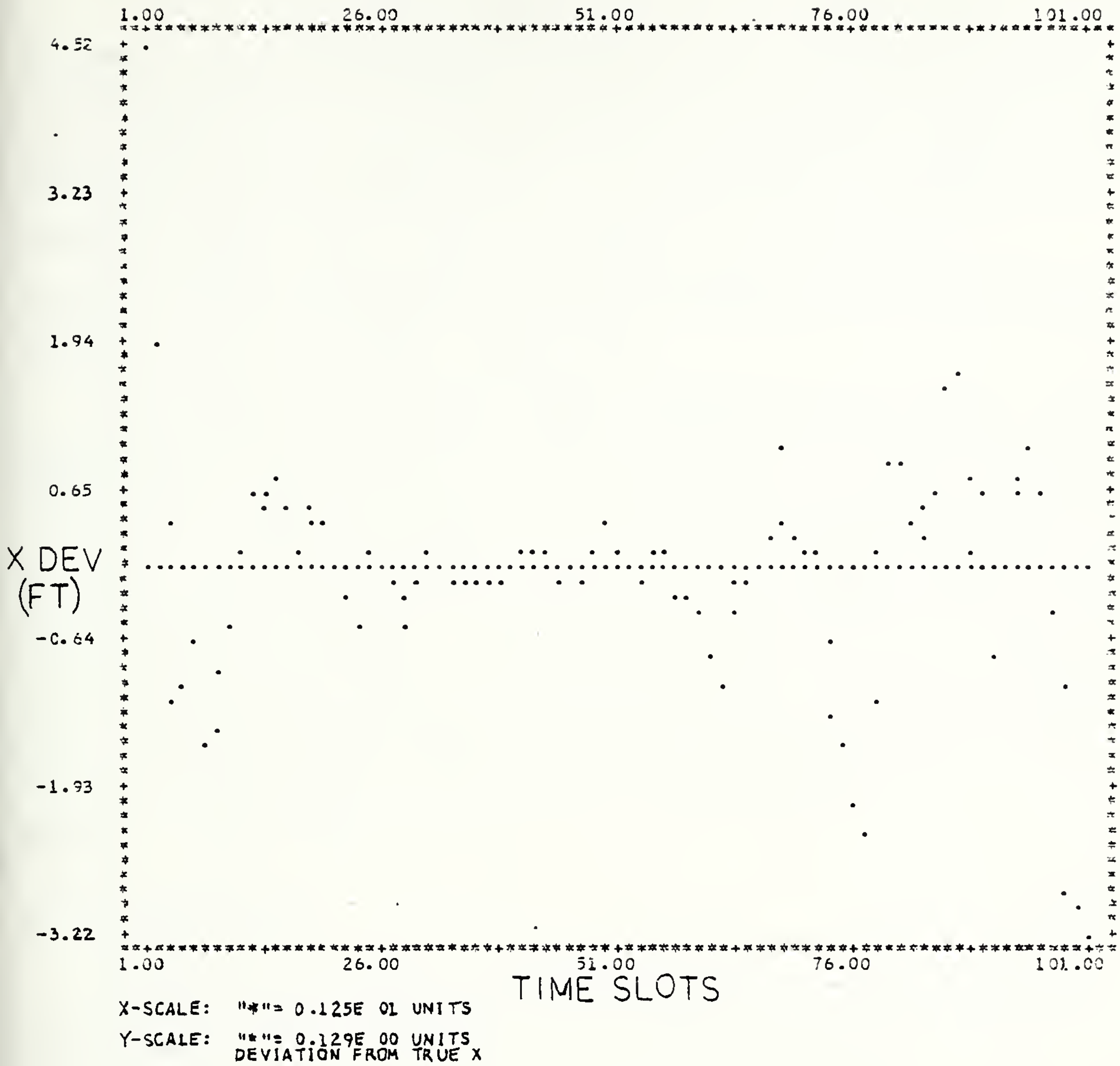


FIGURE 28: X Deviation from True for a Straight Running 42 Knot Target with Noise in the Area of a Single Array -- Sequential Routine

Figures 29 and 30. Figure 29 depicts the estimate's deviation from true in the X direction for the traditional routine and Figure 30 for the sequential. In this track the target maneuvered, using 1/3-G turns, at time slots 25 and 65. A slightly better performance was also exhibited by the traditional routine in the Y direction.

In tests in which more radical maneuvers were made, the traditional routine again exhibited slightly better performance in noise. Figure 31, shows the deviation in the X direction for a track with a 1-G turn at time 25 and a 2/3-G turn at 65 using the sequential routine. Comparision with Figure 32, which is the same track using the traditional routine, indicates that the performances were similar with the traditional routine having a slight edge.

E. SERIES FIVE

In the last series of tests, the targets were tracked through multiple arrays using the handoff scheme described in the previous seciton. Figure 33 is a geographic plot of a typical track including hydrophone positions. Figure 34 depicts the estimates deviation from true in the X direction using the traditional routine for a straight running target in noise. Figure 35 is the X deviation for the same track using the sequential routine. Through the multiple arrays, both routines were comparable, and the handoff from array to array was smooth with no glitches. Figures 36 and 37 also show comparable results for deviation in the Y direction for this same

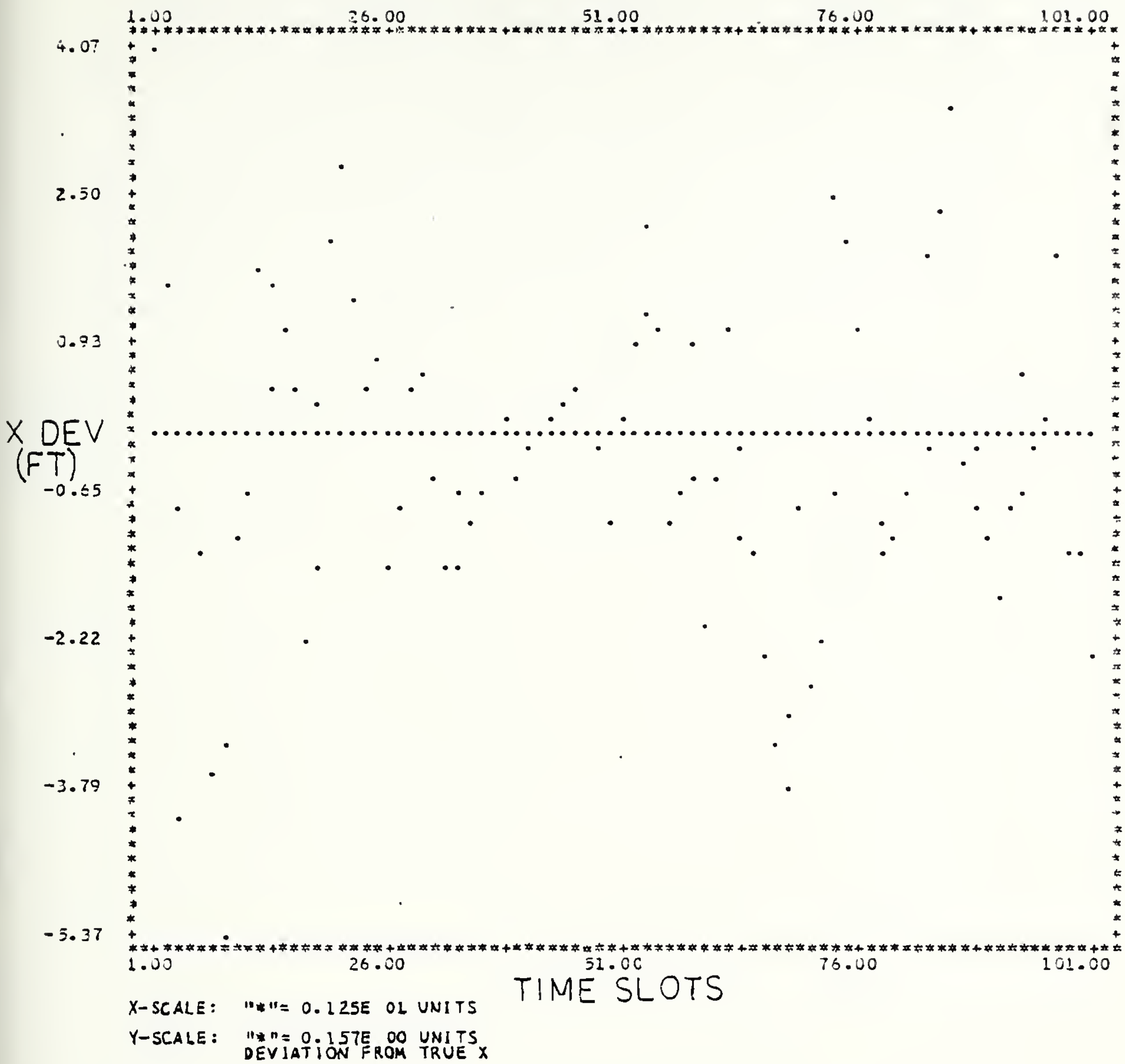


FIGURE 29: X Deviation from True for a Track with 1/3-G Turns at 42 Knots with Noise in the Area of a Single Array -- Traditional Routine

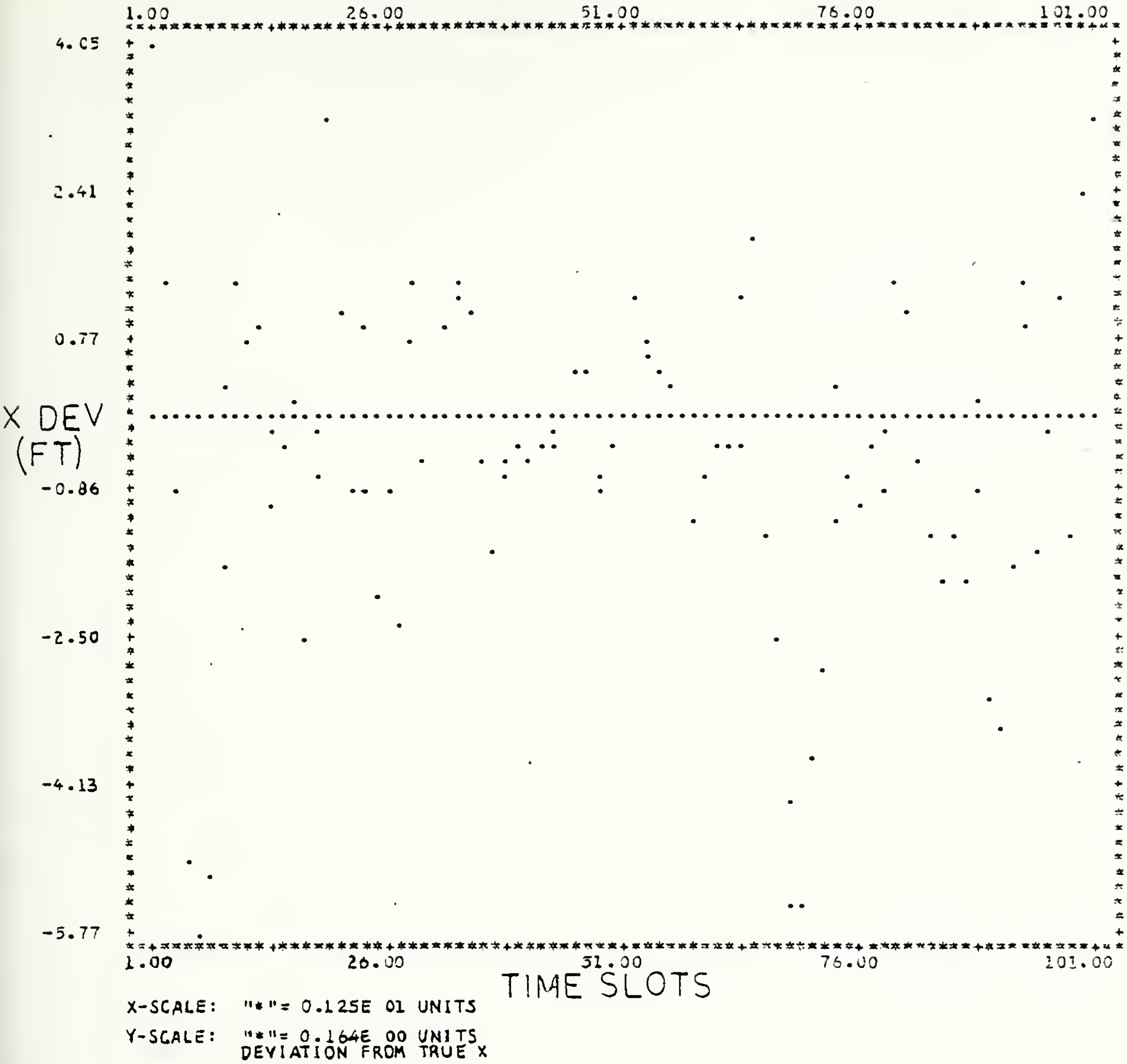


FIGURE 30: X Deviation from True for a Track with 1/3-G Turns at 42 Knots with Noise in the Area of a Single Array -- Sequential Routine

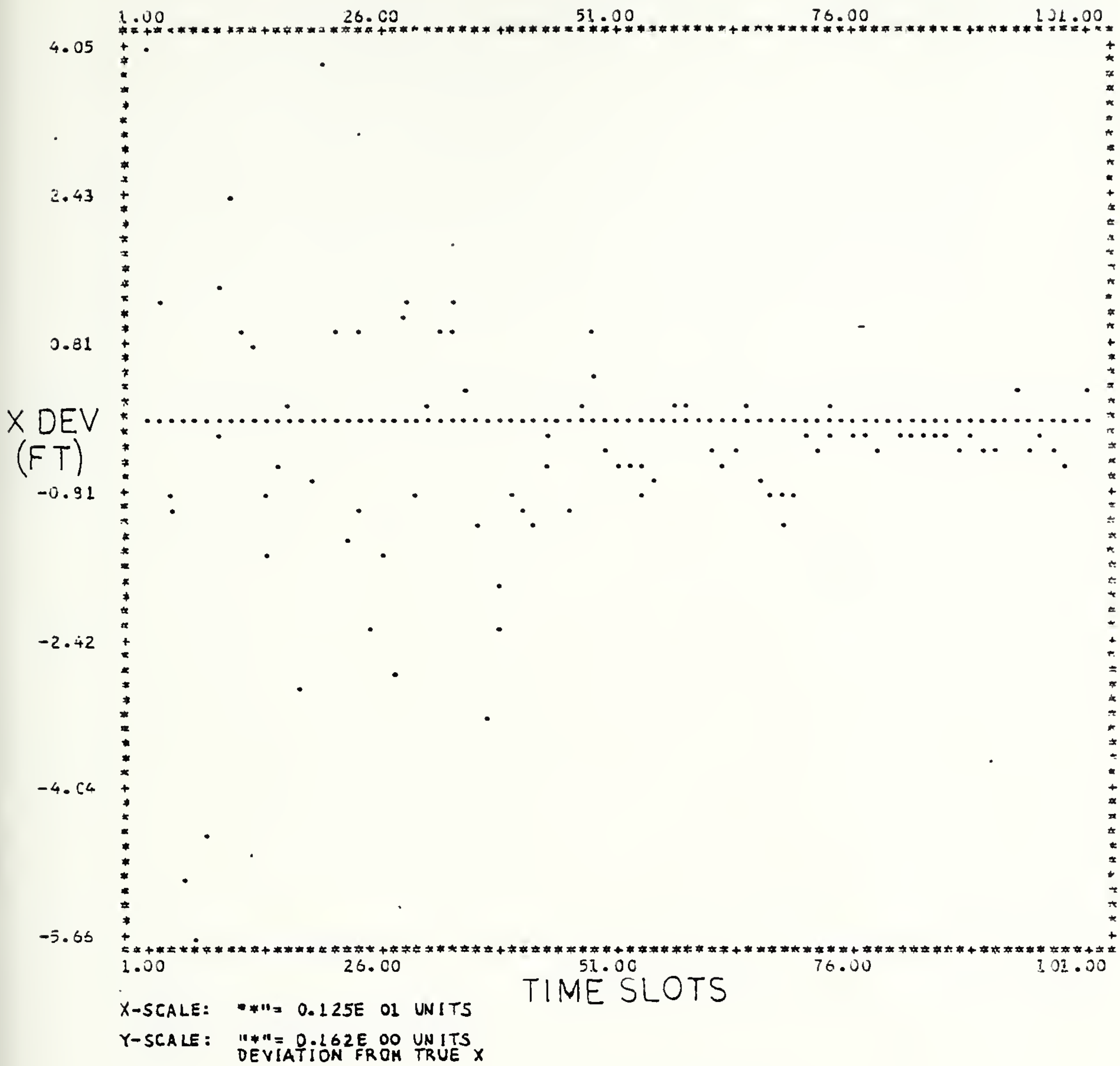


FIGURE 31: X Deviation from True for a Track with 2/3-G and 1-G Turns at 42 Knots with Noise in the Area of a Single Array -- Sequential Routine

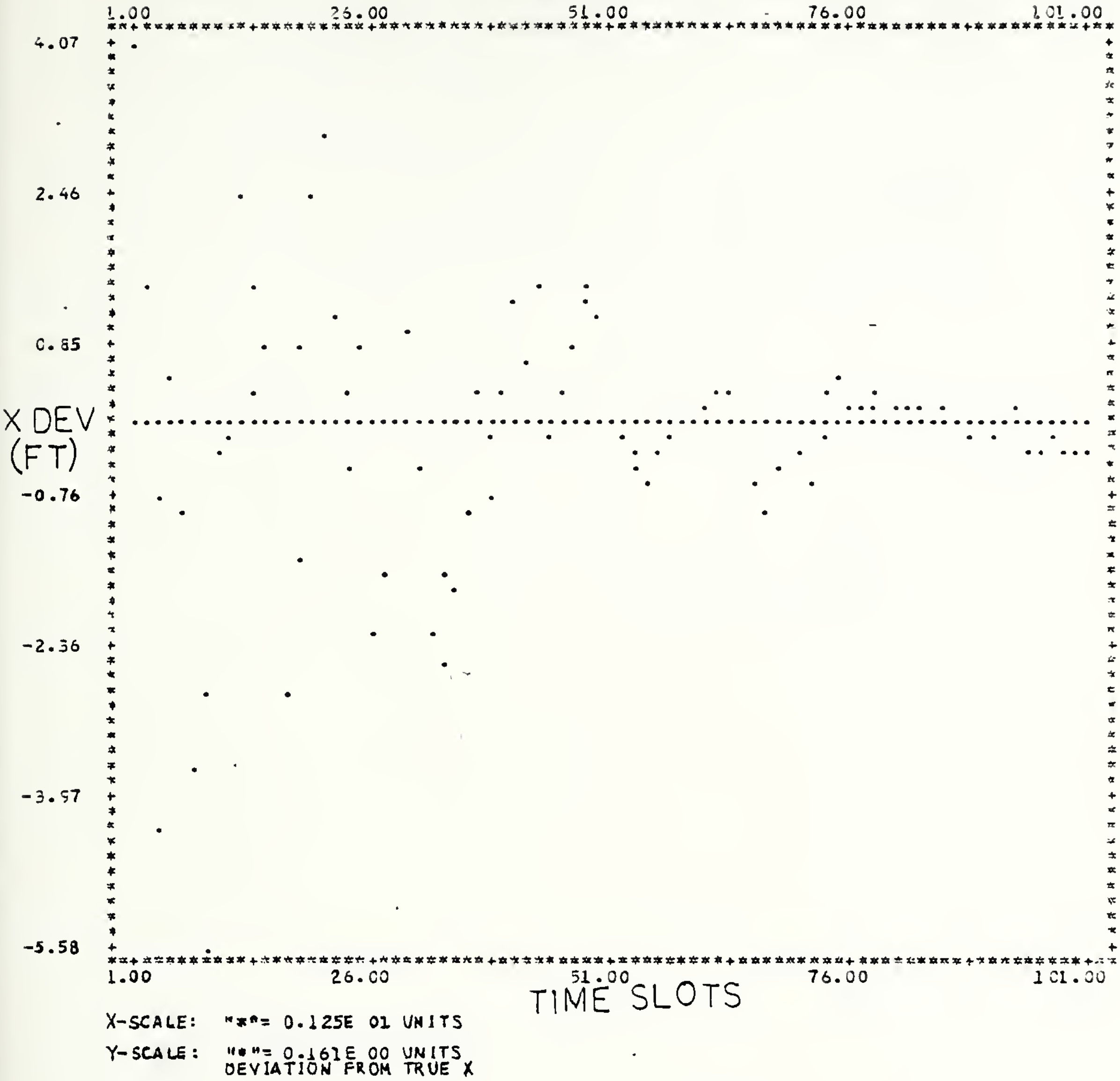


FIGURE 32: X Deviation from True for a Track with 2/3-G and 1-G Turns at 42 Knots with Noise in the Area of a Single Array -- Traditional Routine

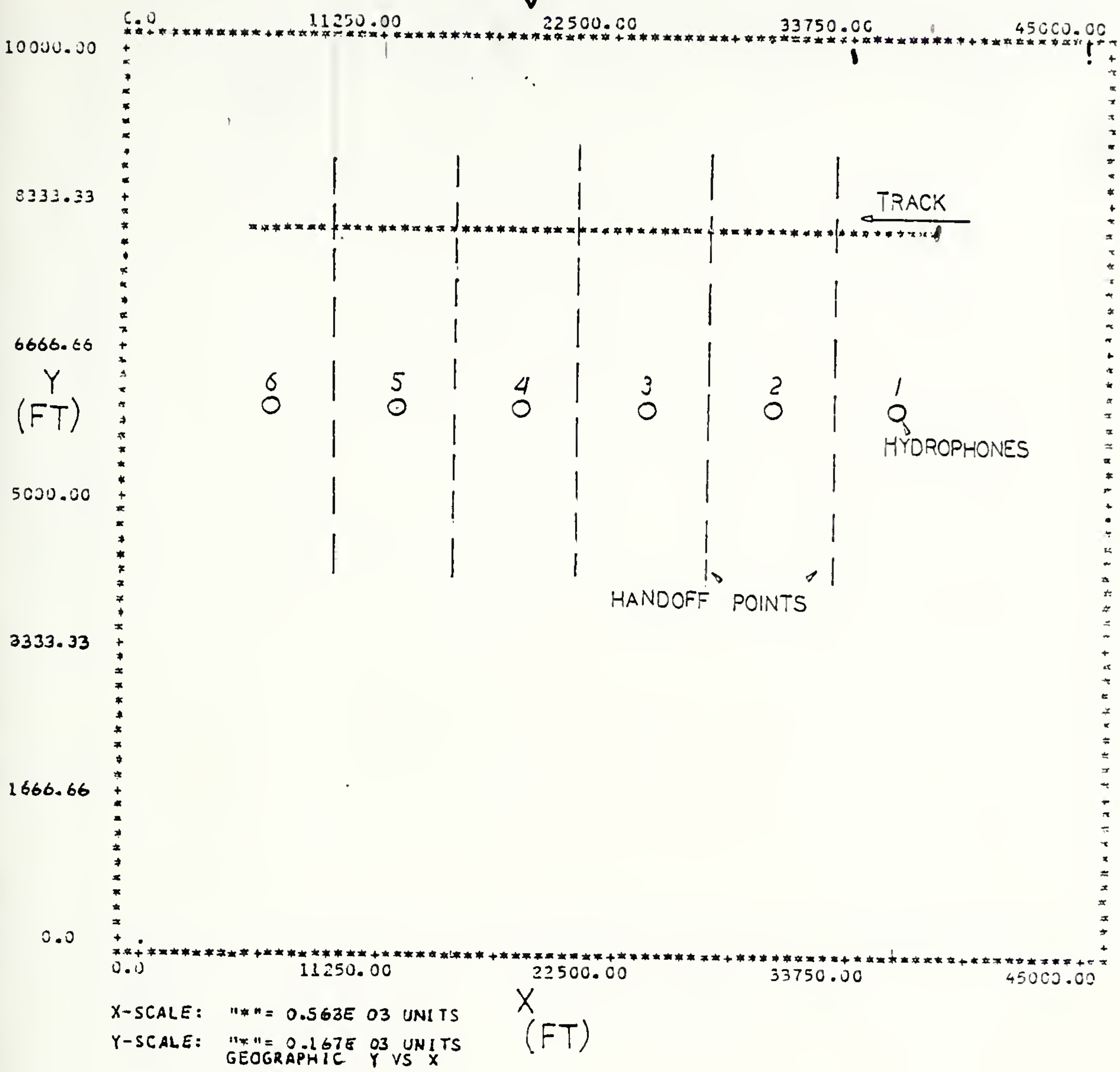


FIGURE 33: Geographic Plot of a Straight Running Track through Multiple Arrays

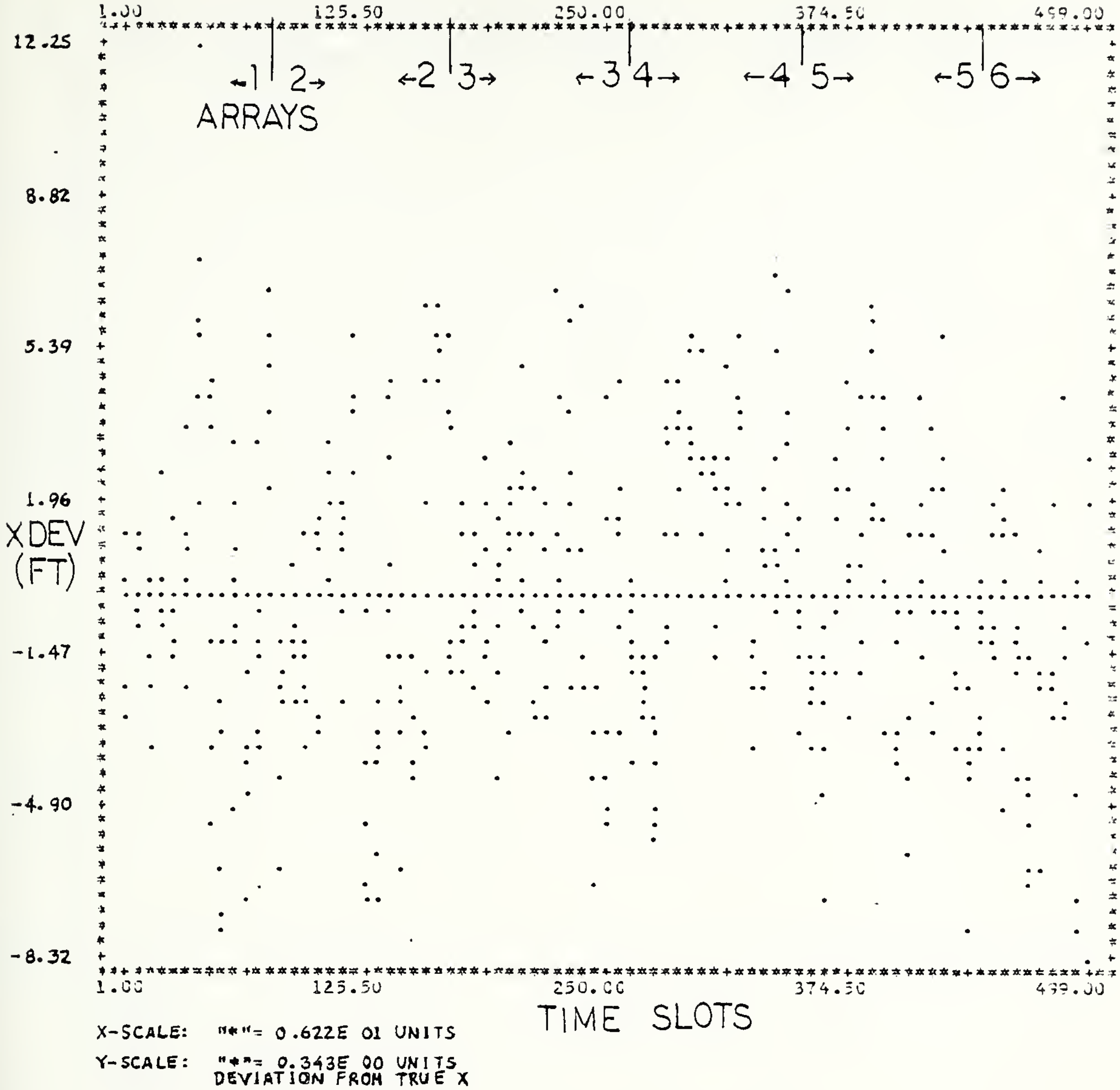


FIGURE 34: X Deviation from True for a Straight Running Track through Multiple Arrays with Noise -- Traditional Routine

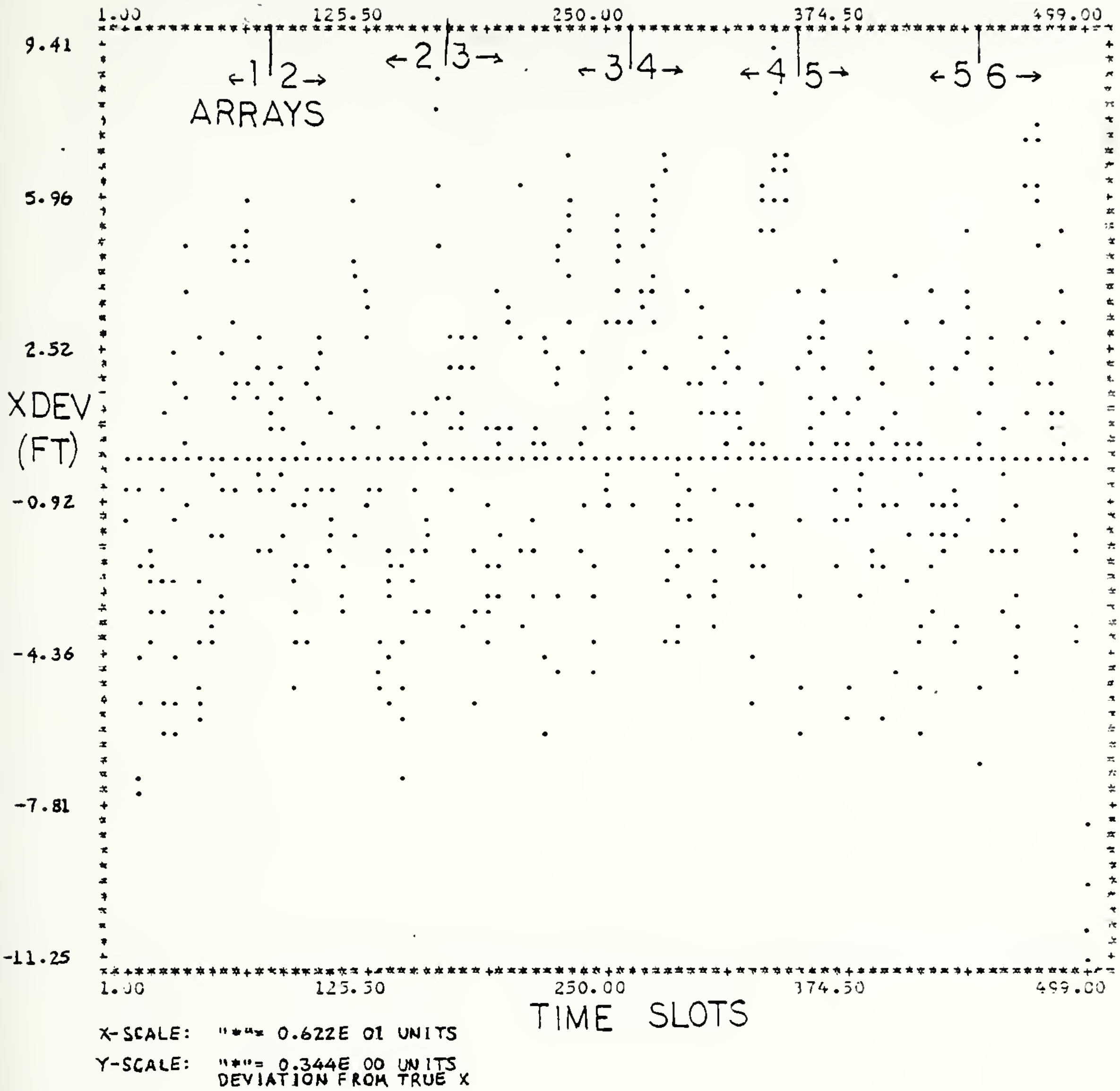


FIGURE 35: X Deviation from True for a Straight Running Track through Multiple Arrays with Noise -- Sequential Routine

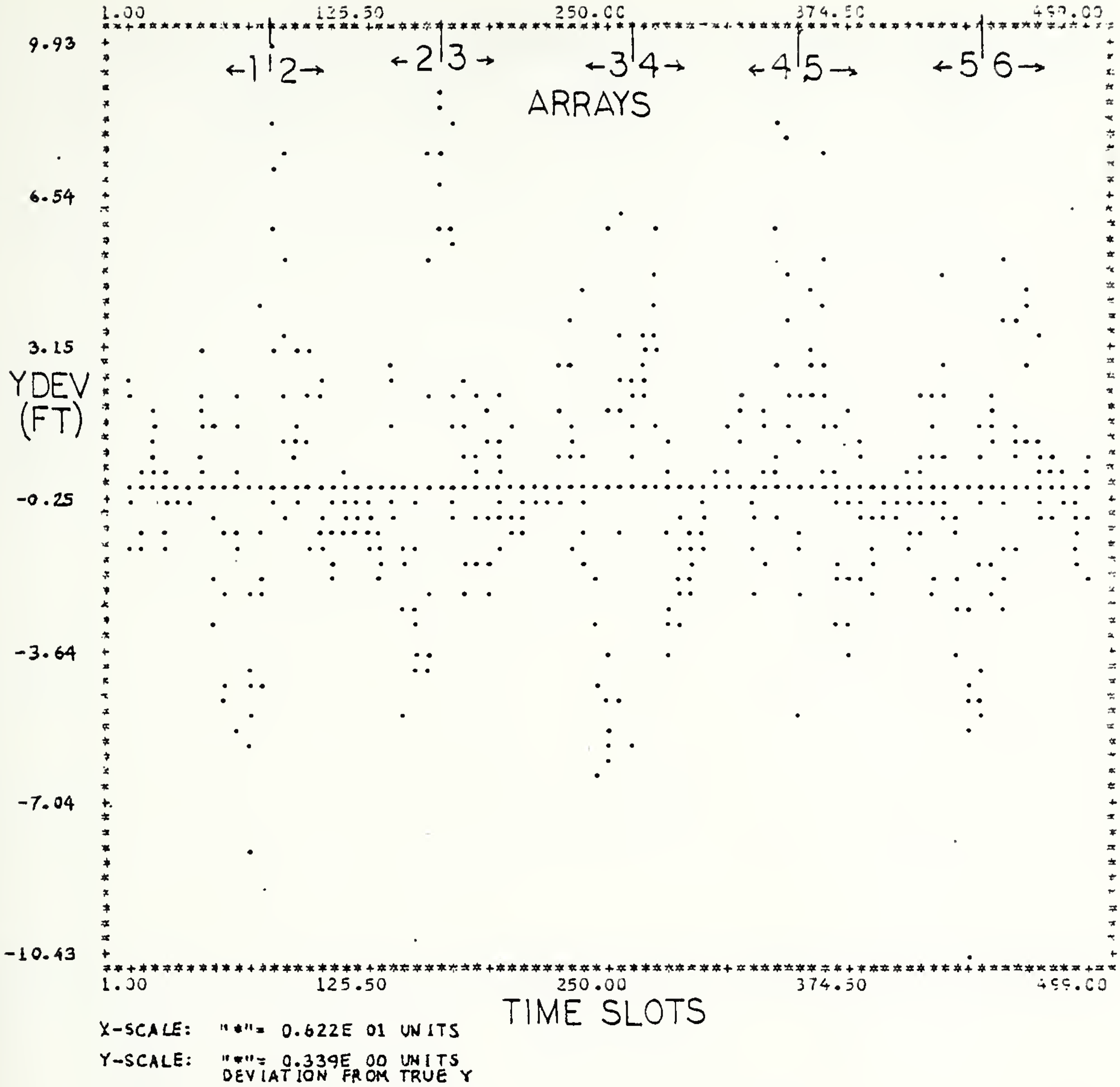


FIGURE 36: Y Deviation from True for a Straight Running Track through Multiple Arrays with Noise -- Traditional Routine

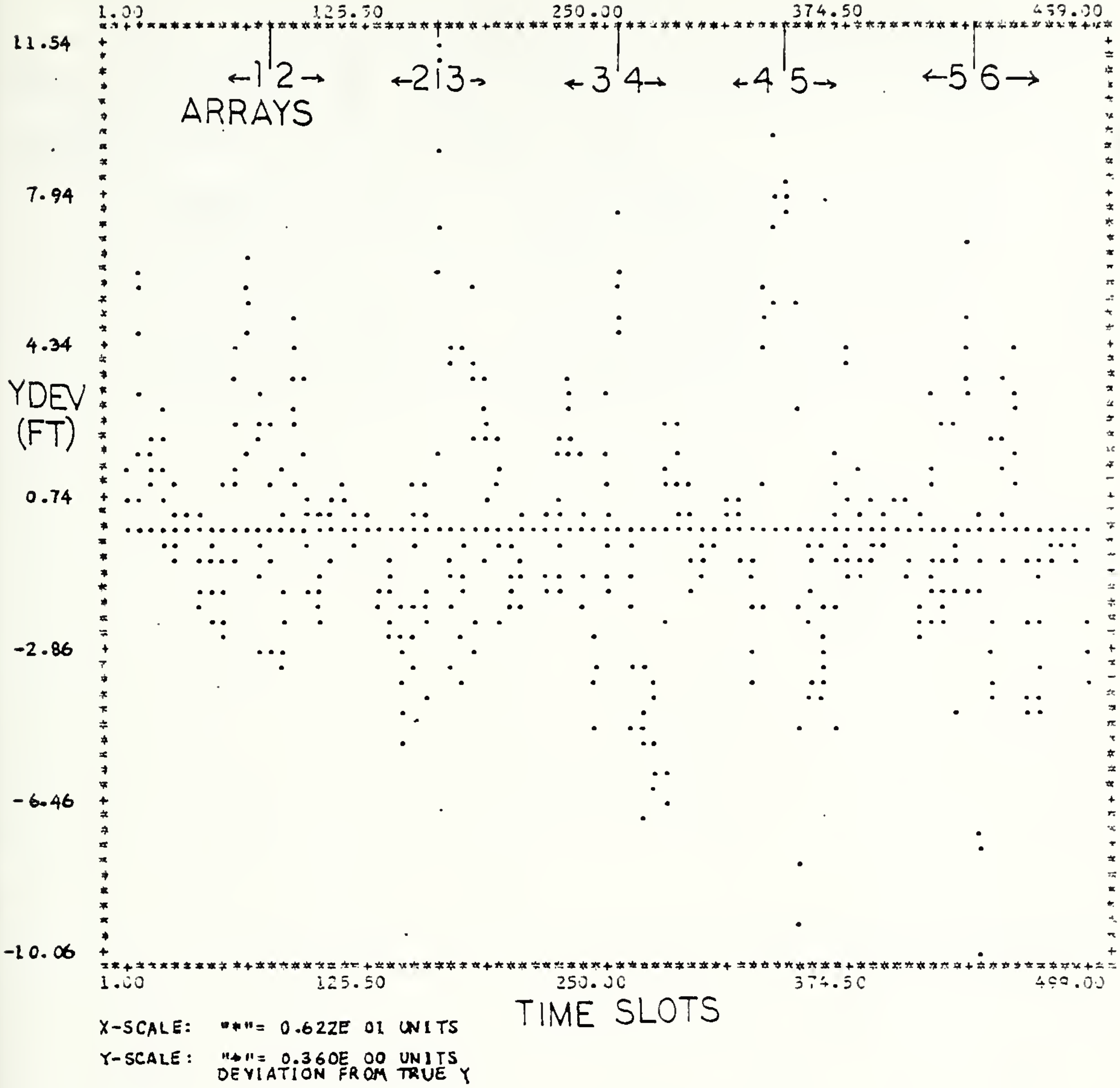


FIGURE 37: Y Deviation from True for a Straight Running Track through Multiple Arrays with Noise -- Sequential Routine

run. The target was traveling at 40 knots and handoff was accomplished at the following times:

Handoff 1 to 2 --- Time Slot 77
Handoff 2 to 3 --- Time Slot 168
Handoff 3 to 4 --- Time Slot 260
Handoff 4 to 5 --- Time Slot 352
Handoff 5 to 6 --- Time Slot 443

VII. CONCLUSIONS

Both extended Kalman filter routines designed will provide on-line, real time estimates of targets with various maneuvers up to 1-G turns.

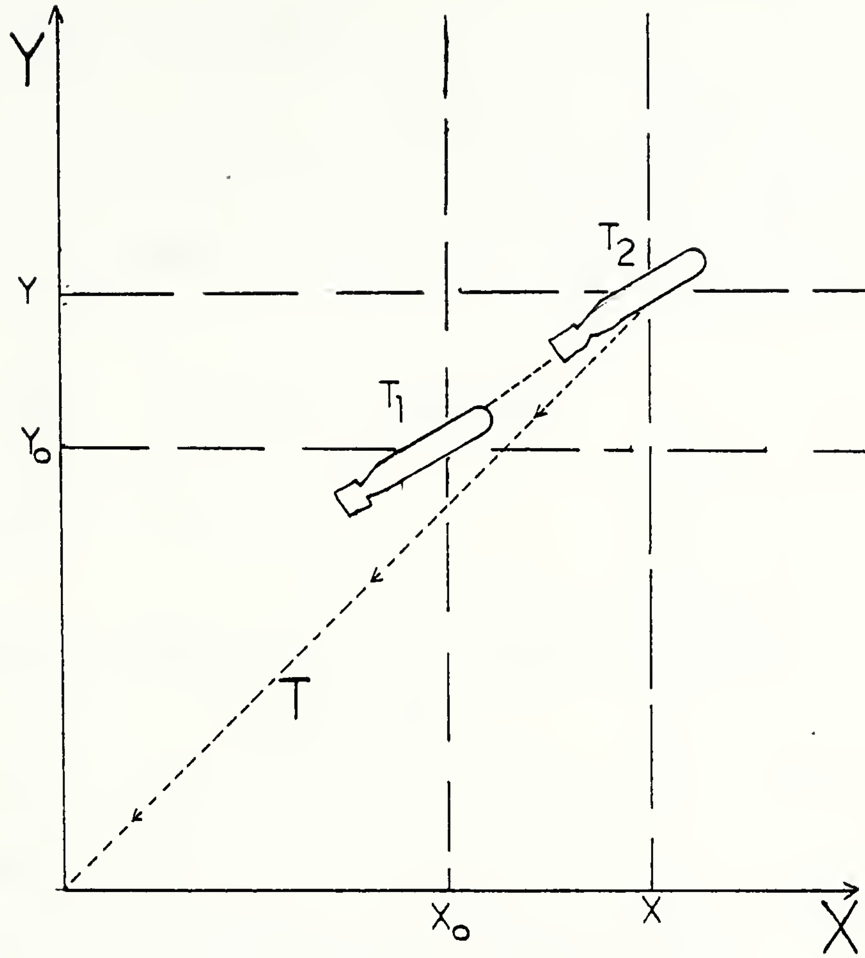
Implementation at the range computer facilities can be accomplished by loading the received transit times into a file in memory and reading them into the routine with subroutine DREAD. The hydrophone positions can also be read into a file to be referenced during operation. The initial covariance of estimation error P(1/0), read in as a constant matrix, can be varied with the uncertainty of the targets initial positions and velocities. Any greatly erroneous time measurements caused by multipath signals or from noise spikes will cause the gains to zero, making the estimated position equal to the predicted, thus putting the filter in 'coast', and preventing catastrophic failure.

The system was remodeled to include the X and Y velocities in the linearizing matrix H. This model is depicted in Figure 38. With a constant Z, the targets position at time t_1 is X_0Y_0. By the time the transmitted signal has reached the hydrophone array, the target with X velocity (V_X) and Y velocity (V_Y) has moved to the position at t_2:

$$X = X_0+V_XT$$

$$Y = Y_0+V_YT$$

where T is the transit time of the signal



MOVING TARGET in the X-Y PLANE

$$X = X_0 + V_x T$$

$$Y = Y_0 + V_y T$$

FIGURE 38: Geometry used to remodel the System to include X and Y Velocities in the H Matrix

$$T = 1/VEL \sqrt{X_0^{\,2} + Y_0^{\,2}}$$

and VEL is the velocity of propagation of sound in water.

Including the new positions corresponding to the target location when the signal is received gives a new equation for the transit time.

$$T = \frac{1}{VEL} \sqrt{(X-V_X T)^2 + (Y-V_Y T)^2}$$

solving for T produces:

$$T = \frac{(XV_X+YV_Y) - \sqrt{(XV_X+YV_Y)^2-(V_X^{\,2}+V_Y^{\,2}-VEL)(X^2+Y^2+Z^2)}}{(V_X^{\,2}+V_Y^{\,2}-VEL)}$$

Upon testing, this added complexity did not give a corresponding increase in the quality of the estimate.

In general, both routines were comparable with the sequential filter having the following advantages:

1. Less deviation and quicker lock-on time after a modest maneuver

2. With the three-sigma gate utilized a noise spike encountered will negate data from only that particular hydrophone

3. A matrix inversion is not necessary.

APPENDIX A

PROGRAM DESCRIPTION AND FEATURES

Two programs were written to implement the extended Kalman filter routine for torpedo tracking. The first, THEFIV, utilizes the sequential approach described in section IV-A, and the second a traditional matrix inversion approach as described in section IV-B. Both routines use the same utility programs and are modularized for ease of implementation.

1. THEFIV - Sequential Routine

This program is general in nature and many of the parameters of the Kalman routine are variable including:

a. The number of states in the routine - N

b. The number of random forcing functions - M

c. The number of measurements - J

d. Data rate or sample time - T0

and $\Phi(1,2)$, $\Phi(3,4)$

e. Number of time slots - JTIME

The constant matrices PHI,R,COVW and GAMMA are read in using subroutines in the utility program AUX. The filter is initialized with $\underset{\sim}{P}(1/0)$ and $\underset{\sim}{X}(1/0)$ (initial covariances of estimation error and states) also using AUX. The first state estimate is at time 1 and continues until ITIME = JTIME+1. True measurement times (ZI) are read in, four for each time slot (T_C, T_X, T_Y, T_Z), using the subroutine DREAD listed in AUX and corrupted by zero-mean, white Gaussian noise using the

IBM-360 subroutine SNORM. For each of the four time measurements the corresponding row of the linearizing H matrix is calculated in the utility subroutine CHROW, and the corresponding gain matrix column GI is found. These row and column values are then utilized in forming the covariance of extimation error for the particular time measurement PI. Next the estimate of the observation time $\hat{M}(\underset{\sim}{X})$ from that particular hydrophone called ZHAT is formed using the subroutine CZHAT and the residual ZDIFF = ZI-ZHAT. Finally, the estimate of the states XI based on one time measurement is calculated, and the process is repeated for the next measurement. After four iterations, XI becomes the state estimate and PI becomes the updated covariance of estimation error PKK, and the predictions of the states and covariances XKKM1 and PKKM1 are formed.

For testing, this program used the IBM-360 library subroutine PLOTP to obtain plots of the states and covariances versus time, estimate deviation from true versus time and a geographic track.

2. THESIS - Traditional Routine

THESIS utilized the same format as THEFIV. The parameters N,M,JS, sample time and number of time slots were still variable, and the filter initial conditions and constant matrices were read in by the subroutines located in AUX. True time measurements were again read in for each time slot through DREAD and corrupted with noise from SNORM. The linearizing H matrix was formed row by row using subroutine CHROW and used

to calculate the gain matrix G. The symmetric matrix inversion in the gain equation was done by the IBM-360 library subroutine SINV. The estimate vector of the observations ZHAT is formed iteratively by CZHAT and used to calculate the residual vector ZDIFF. Next the estimate XKK and updated covariance of error matrix PKK is calculated once for each time slot. Finally, the predictions XKKM1 and PKKM1 are formed before the process is repeated for the next time slot. PLOTP was again utilized for the output graphs.

3. Utility Programs

These subroutines were designed to be used for repetitive calculations and processes. The first, AUX performs all data input functions and matrix manipulations including:

a. PROD - multiplying two matrices

b. MMULT - multiplying a matrix and a vector

c. VMULT - multiplying two vectors

d. MREAD - reading in a matrix

e. TRANS - transposing a matrix

f. ADD - adding two matrices

g. VREAD - reading in a vector

h. DREAD - reads in a matrix containing observed time data

i. TRREAD - reads in a matrix containing true positional data for comparison

The second utility subroutine CZHAT calculates the estimate of the observation times (T_C, T_X, T_Y or T_Z) using the predicted state values ($\hat{\underset{\sim}{X}}(K/K-1)$).

The subroutine CHROW calculates a row of the linearizing H matrix, each row corresponding to a particular observation time measurement.

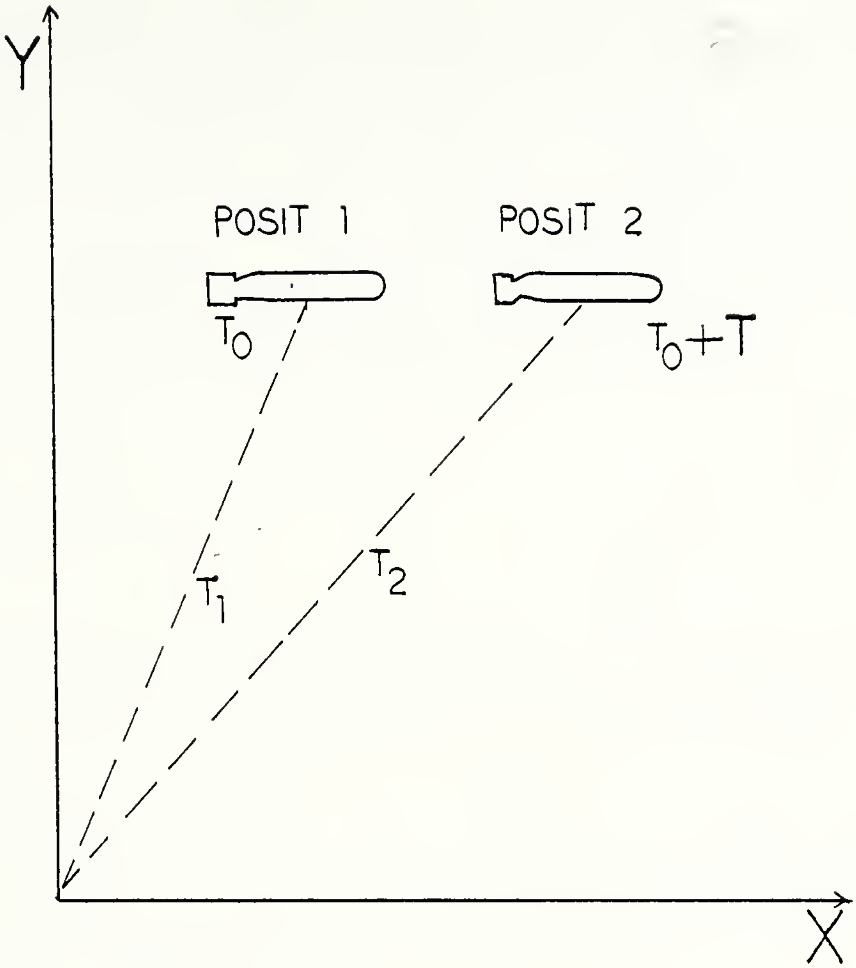
A. VARYING STATE TRANSITION MATRIX

In the $1/S^2$ model that represents the dynamics of the Kalman filter, the State Transition and GAMMA matrices are as follows:

$$\Phi = \begin{bmatrix} 1 & T & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & T & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\Gamma = \begin{bmatrix} T2/2 & 0 & 0 \\ T & 0 & 0 \\ 0 & T2/2 & 0 \\ 0 & T & 0 \\ 0 & 0 & T \end{bmatrix}$$

where T is the sampling time. In the tracking problem, this time is not always constant. As is shown in Figure 39 it varies with the transit times required from positions in adjacent time slots. In Figure 39, the target is at position 1 at arbitrary time t_0, and is at position 2, the next time slot, at t_0+T where T is the time between pings of the target transducer = 1.31 seconds. The times that the transmitted



Moving Target in the X-Y Plane

$$T_s = (t_0 + T + t_2) - (t_0 + t_1)$$

$$T_s = T + (t_2 - t_1)$$

FIGURE 39: Geometry used for the Time Varying State Transition Matrix

signals are received by the hydrophone array are:

Time signal received from position 1 = t_0+t_1

Time signal received from position 2 = t_0+T+t_2

where t_1 and t_2 are the transit times from position 1 and position 2 respectively. The difference between these two times signals are received is the sampling time T_S.

$$T_S = (t_0+T+t_2) - (t_0+t_1)$$

$$T_S = T + (t_2-t_1)$$

Therefore, the sampling time differs each time slot from the clocked ping time by the difference in the adjacent slot transit times.

In the programs THEFIV and THESIS, this difference is calculated using the average measured times from the four hydrophones for each time slot.

B. ADAPTIVE Q

The Q matrix which appears in the predicted covariance of error equation

$$\underset{\sim}{P}(K+1/K) = \underset{\sim}{\Phi}\underset{\sim}{P}(K/K)\underset{\sim}{\Phi}^T+\underset{\sim}{Q}(K)$$

and is formed by

$$\underset{\sim}{Q}(K) = \underset{\sim}{\Gamma}COVW\underset{\sim}{\Gamma}^T$$

is a measure of the amount of target maneuverability that can be handled by the filter. If more random excitations (or accelerations) by the target is expected, Q is increased which

in turn increases the covariances of estimation error $\underset{\sim}{P}(K+1/K)$ and the filter gains G. Corresponding the filter puts more emphasis on the incoming data and is better able to see and react to target turns. However, if the filter gains are increased the filter bandwidth is widened, which lets in more noise, and makes the filter more susceptible to error. The adaptive Q routine (Reference 4) varies the Q matrix as the velocities in the X and Y directions are increased or decreased. This routine was implemented in the subroutine QFIND with inputs of:

SIGACC = expected maximum acceleration in the X or Y direction in ft/sec^2

SIGDIV = expected maximum acceleration in the Z direction in ft/sec^2

SIGCC = expected maximum target course change in degrees/sec

QFIND is listed in Appendix D.

APPENDIX B
SEQUENTIAL EXTENDED KALMAN FILTER
THEFIV

```
      COMMON XKKM1(5),PKKM1(5,5),PKK(5,5),XKK(5)                      THE00010
      DIMENSION PHI(5,5),GAMMA(5,3),R(4,4),PHIPKK(5,5),PHIT(5,5),      THE00020
     1PKTEMP(5,5),HROW(5),PI(5,5),PDUM(5,5),QTEMP(5,3),X1(510),        THE00030
     1GNUM(5),GI(5),XI(5),ZI(4),GAMMAT(3,5),Q(5,5),COVW(3,3),          THE00040
     1X5(510),KOUNT(510),AKOUNT(510),PVEC(510),P1(510,5,5),X3(510),    THE00050
     1TRUX(510),TRUY(510),TRUZ(510),TD(3),C1(2),C2(2),C3(2),           THE00060
     1C4(2),C5(2),C6(2),C7(2),XP(5,510),XDIFF1(510),XDIFF3(510),       THE00070
     1XDIFF5(510),ZC(4),HYDRO(6,12),XB(4),YB(4),ZB(4)                  THE00080
      N=5                                                              THE00090
      M=3                                                              THE00100
      JS=4                                                             THE00110
      I8=1                                                             THE00120
      TO=1.31                                                          THE00130
      READ(5,200)JTIME                                                 THE00140
  200 FORMAT(I3)                                                       THE00150
      READ(5,201)SIGACC,SIGDIV,SIGCC                                   THE00160
  201  FORMAT(3F4.2)                                                   THE00170
      SIGCC=SIGCC*3.14159/180.                                         THE00180
      DO 799 I=1,6                                                     THE00190
  799  READ(7,739)(HYDRO(I,J),J=1,12)                                  THE00200
  739  FORMAT(12F6.1)                                                  THE00210
C LOAD X(0/-1)                                                         THE00240
      CALL VREAD(XKKM1,N)                                              THE00260
C LOAD F(0/-1)                                                         THE00270
      CALL MREAD(PKKM1,N,N)                                            THE00280
C READ IN CONSTANT MATRICES-GET TRANSPOSES                             THE00300
      CALL MREAD(PHI,N,N)                                              THE00310
      CALL MREAD(R,JS,JS)                                              THE00320
      CALL MREAD (COVW,M,M)                                            THE00340
      CALL MREAD(GAMMA,N,M)                                            THE00350
      CALL TRANS(GAMMA,N,M,GAMMAT)                                     THE00360
      CALL TRANS(PHI,N,N,PHIT)                                         THE00370
C                                                                      THE00380
C START THE TIME SLOT LOOP                                             THE00390
C                                                                      THE00400
      ITIME=JTIME+1                                                    THE00610
      XT=36000.                                                        THE00620
      SW=XT-3000.                                                      THE00630
        DO 99 KK=1,ITIME                                               THE00640
      DO 600 I3=1,4                                                    THE00650
      XB(I3)=HYDRC(I8,3*I3-2)                                          THE00660
      YB(I3)=HYDRO(I8,3*I3-1)                                          THE00670
```

```
 600  ZB(I3)=HYDRC(I8,3*I3)                                             THE00710
      IF(XKKM1(1).GT.SW)GO TO 610                                       THE00720
      I8=I8+1                                                           THE00730
      WRITE(6,759)I8,KK                                                 THE00740
 759   FORMAT(1X,'ARRAY',2X,I2,'STARTS TRACKING AT TIME',2X,I3)         THE00750
      SW=SW-6000.                                                       THE00760
      XT=XT-6000.                                                       THE00770
C                                                                       THE00880
C COMPUTE THE TRUE TIMES                                                THE00890
C                                                                       THE00900
 610     CALL DREAD(ZI,JS)                                              THE00910
C                                                                       THE00920
C TIME VARY THE STATE TRANSITION MATRIX PHI                             THE00930
C                                                                       THE00940
      T2=(.25)*(ZI(1)+ZI(2)+ZI(3)+ZI(4))                                THE00950
      IF(KK.EQ.1)T1=T2                                                  THE00960
      PHI(1,2)=T0-(T2-T1)                                               THE00970
      PHI(3,4)=PHI(1,2)                                                 THE00980
      PHIT(2,1)=PHI(1,2)                                                THE00990
      PHIT(4,3)=PHI(1,2)                                                THE01000
       T1=T2                                                            THE01010
      A14=PHI(1,2)                                                      THE01020

         CALL TRREAD(TD,M)                                              THE01030
         TRUX(KK)=TD(1)                                                 THE01040
         TRUY(KK)=TD(2)                                                 THE01050
         TRUZ(KK)=TD(3)                                                 THE01060
C FIRST GET HROW-CALCULATE GAIN,ESTIMATE,COVARIANCE OF                  THE01070
C ERROR BASED ON ONE TIME MEASUREMENT-TC,TX,TY OR TZ
                                                                        THE01100
 711        DO 97 I=1,JS                                                THE01110
           CALL CHROW3(I,HROW,XB,YB,ZB)                                 THE01120
           CALL MMULT(PKKM1,HROW,N,N,GNUM)                              THE01130
           CALL VMULT(HROW,GNUM,N,GDTEMP)                               THE01140
           GDENOM=GDTEMP+R(I,I)                                         THE01150
              DO 16 IX=1,N                                              THE01160
 16           GI(IX)=GNUM(IX)/GDENOM                                    THE01170
C THIS IS THE FIRST GAIN COLUMN                                         THE01180
C CALCULATE THE COVARIANCE OF ERROR - PI
                                                                        THE01210
            DO 77 IP=1,N                                                THE01220
               DO 79 JP=1,N                                             THE01230
               PDUM(IP,JP)=(-1.*GI(IP))*HROW(JP)                        THE01240
              IF(IP.EQ.JP)PDUM(IP,JP)=1.+PDUM(IP,JP)                    THE01250
 79           CONTINUE                                                  THE01260
 77         CONTINUE                                                    THE01270
```

```
          CALL PROD (PDUM,PKKM1,N,N,N,PI)                                THE01280
                                                                         THE01340
C CALCULATE THE FIRST MEASUREMENT PREDICTION
                                                                         THE01360
          CALL CZHAT3(I,ZHAT,XB,YB,ZB)                                   THE01370
 122      ZDIFF=ZIC-ZHAT                                                 THE01430

C SET THE GATE
       PK1=ABS(PI(1,1))                                                  THE01460
       PK3=ABS(PI(3,3))                                                  THE01470
       PK5=ABS(PI(5,5))                                                  THE01480
       IF((PK1.GE.PK3).AND.(PK1.GE.PK5))P=PK1                            THE01490
       IF((PK3.GE.PK1).AND.(PK3.GE.PK5))P=PK3                            THE01500
       IF((PK5.GE.PK1).AND.(PK5.GE.PK3))P=PK5                            THE01510
       PGATE=P/4860.
       RGATE=SQRT(ABS(R(I,I)))                                           THE01540
       GATE=3.*SQRT(PGATE+RGATE)
       IF(ZDIFF.LT.GATE)GO TO 500                                        THE01560
       WRITE(6,501)KK                                                    THE01570
 501   FORMAT('0','GATE HAS BEEN EXCEEDED TIME',I4)                      THE01580
       DO 502 LG=1,N                                                     THE01590
 502   GI(LG)=0.0                                                        THE01600
                                                                         THE01610
C CALCULATE THE ESTIMATE BASED ON ONE MEASUREMENT
 500            DO 17 IZ=1,N                                             THE01630
 17            XI(IZ)=XKKM1(IZ)+GI(IZ)*ZDIFF                             THE01640
                                                                         THE01650
                                                                         THE01880
          IF(I.EQ.4)GO TO 56                                             THE01890
   12       DO 19  IQ=1,N                                                THE01900
 19          XKKM1(IQ)=XI(IQ)                                            THE01910
              DO 23 IQ=1,N                                               THE01920
                DO 18 JQ=1,N                                             THE01930
 18               PKKM1(IQ,JQ)=PI(IQ,JQ)                                 THE01940
 23             CONTINUE                                                 THE01950
 97         CONTINUE                                                     THE01980
                                                                         THE01990
C ORIGINAL X(0/-1) WAS XKKM1, UPDATED AFTER 1
C MEASUREMENT AND CALLED IT XI, THEN XKKM1=XI, WENT
C THRU ITERATION AGAIN, AFTER XKKM1 UPDATED FOR EACH
C MEASUREMENT XKK=XI AND PKK=PI
                                                                         THE02040
 56       DO 57 ID=1,N                                                   THE02050
          XKK(ID)=XI(ID)                                                 THE02060
            DO 58 JD=1,N                                                 THE02070
 58         PKK(ID,JD)=PI(ID,JD)                                         THE02080
 57       CONTINUE                                                       THE02090
C                                                                        THE02200
```

```
      XDIFF1(KK)=TRUX(KK)-XKK(1)                                        THE02210
      XDIFF3(KK)=TRUY(KK)-XKK(3)                                        THE02220
      XDIFF5(KK)=TRUZ(KK)-XKK(5)                                        THE02230
                                                                        THE02240
C CALCULATE THE PREDICTIONS FOR PKKM1
      CALL QFIND(KK,SIGACC,SIGDIV,SIGCC,A14,Q)                          THE02260
        CALL PROD(PHI,PKK,N,N,V,PHIPKK)                                 THE02330
        CALL PROD(PHIPKK,PHIT,N,N,N,PKTEMP)                             THE02340
        CALL ADD(PKTEMP,Q,N,N,PKKM1)                                    THE02350
                                                                        THE02360
C CALCULATE THE PREDICTIONS FOR XKKM1
                                                                        THE02380
        CALL MMULT(PHI,XKK,N,N,XKKM1)                                   THE02390
         DO 41 IG=1,N                                                   THE02400
41       XP(IG,KK)=XKK(IG)                                              THE02410
        DO 38 II=1,N                                                    THE02420
          DO 39 JJ=1,N                                                  THE02430
39        P1(KK,II,JJ)=PKK(II,JJ)                                       THE02440
38      CONTINUE                                                        THE02450
99    CONTINUE                                                          THE02480
      C1(1)=290.                                                        THE02490
      C1(2)=310.                                                        THE02500
      C3(1)=0.                                                          THE02510
      C3(2)=510.                                                        THE02520
      C4(1)=0.                                                          THE02530
      C4(2)=45000.                                                      THE02540
      C5(1)=0.                                                          THE02550
      C5(2)=10000.                                                      THE02560
      DO 91 KK=1,ITIME                                                  THE02570
      X1(KK)=XP(1,KK)                                                   THE02580
      X3(KK)=XP(3,KK)                                                   THE02590
91    X5(KK)=XP(5,KK)                                                   THE02600
          DO 92 IK=1,ITIME                                              THE02610
92        KOUNT(IK)=IK                                                  THE02620
             DO 113 IF=1,ITIME                                          THE02630
113          AKOUNT(IF)=FLOAT(KOUNT(IF))                                THE02640
      DO 238 II=1,N                                                     THE02650
      JJ=II                                                             THE02660
        DO 240 KK=1,ITIME                                               THE02670
240     PVEC(KK)=P1(KK,II,JJ)                                           THE02680
        WRITE(6,900)                                                    THE02690
        CALL PLOTP(AKOUNT,PVEC,ITIME,0)                                 THE02700
238    WRITE(6,800)II,JJ                                                THE02710
800    FORMAT(23X,'PKK(',I1,',',I1,')','VS TIME')                       THE02720
900    FORMAT('1')                                                      THE02730
      WRITE(6,900)                                                      THE02740
      CALL PLOTP(C3,C4,2,1)                                             THE02750
```

```
      CALL  FLOTP(AKOUNT,X1,ITIME,2)
      CALL  PLOTP(AKOUNT,TRUX,ITIME,3)
      WRITE(6,801)
801   FORMAT(25X,'XPOSIT VS TIME')
      WRITE(6,900)
      CALL  PLOTP(AKOUNT,XDIFF1,ITIME,0)
      WRITE(6,805)
 805  FORMAT(25X,'DEVIATION FROM TRUE X')
      WRITE(6,900)
      CALL  PLOTP(C3,C5,2,1)
      CALL  PLOTP(AKOUNT,X3,ITIME,2)
      CALL  PLOTP(AKOUNT,TRUY,ITIME,3)
      WRITE(6,802)
802   FORMAT(25X,'YPOSIT VS TIME')
      WRITE(6,900)
      CALL  PLOTP(AKOUNT,XDIFF3,ITIME,0)
      WRITE(6,806)
 806  FORMAT(25X,'DEVIATION FROM TRUE Y')
      WRITE(6,900)
      CALL  PLOTP(C3,C1,2,1)
      CALL  PLOTP(AKOUNT,X5,ITIME,2)
      CALL  PLOTP(AKOUNT,TRUZ,ITIME,3)
      WRITE(6,803)
803   FORMAT(25X,'Z DEPTH VS TIME')
      WRITE(6,900)
      CALL  PLOTP(C4,C5,2,1)
      CALL  PLOTP(X1,X3,ITIME,2)
      CALL  PLOTP(TRUX,TRUY,ITIME,3)
      WRITE(6,804)
804   FORMAT(25X,'GEOGRAPHIC  Y VS X')
      WRITE(6,900)
      END
```

APPENDIX C

TRADITIONAL EXTENDED KALMAN FILTER

THESIS

```
      COMMON XKKM1(5),PKKM1(5,5),PKK(5,5),XKK(5)                             THE00010
      DIMENSION PHI(5,5),GAMMA(5,3),R(4,4),PHIPKK(5,5),PHIT(5,5),            THE00020
     1PKTEMP(5,5),H(4,5),HT(5,4),AA(20),CTEMP(5,3),X1(510),X3(510),          THE00030
     1GNUM(5,4),G(5,4),ZI(4),GAMMAT(3,5),Q(5,5),COVW(3,3),                   THE00040
     1X5(510),KOUNT(510),AKOUNT(510),PVEC(510),P1(510,5,5),                  THE00050
     1TRUX(510),TRUY(510),TRUZ(510),TD(3),C1(2),C2(2),                       THE00060
     1XP(5,510),C4(2),C5(2),HROW(5),GDTEM1(4,5),GDTEM2(4,4),                 THE00070
     1GDENOM(4,4),ZDIFF(4),GZ(5),PTEM1(5,5),PTEM2(5,5),                      THE00080
     1EI(5,5),C3(2),ZH(4),C6(2),C7(2),L(4),MM(4),ZC(4),ZIC(4),ZCS(4),        THE00090
     1XDIFF1(510),XDIFF3(510),XDIFF5(510),HYDRO(6,12),                       THE00100
     1XB(4),YB(4),ZB(4)                                                      THE00110
      N=5                                                                    THE00120
      M=3                                                                    THE00130
      JS=4                                                                   THE00140
      I8=1                                                                   THE00150
      T0=1.31                                                                THE00160
      READ(5,200)JTIME                                                       THE00170
  200 FORMAT(I3)                                                             THE00180
      READ(5,201)SIGACC,SIGDIV,SIGCC                                         THE00190
  201  FORMAT(3F4.2)                                                         THE00200
      SIGCC=SIGCC*3.14159/180.                                               THE00210
      DO 799 I=1,6                                                           THE00220
  799 READ(7,739)(HYDRO(I,J),J=1,12)                                         THE00230
  739  FORMAT(12F6.1)                                                        THE00240
C                                                                            THE00270
C FORM AN IDENTITY MATRIX EI                                                 THE00280
C                                                                            THE00290
      DO 140 I=1,N                                                           THE00300
             DO 140 J=1,N                                                    THE00310
  140 EI(I,J)=0.0                                                            THE00320
      DO 141 I=1,N                                                           THE00330
  141 EI(I,I)=1.0                                                            THE00340
C LOAD X(0/-1)                                                               THE00350
      CALL VREAD(XKKM1,N)                                                    THE00370
C LOAD P(0/-1)                                                               THE00380
      CALL MREAD(PKKM1,N,N)                                                  THE00390
C READ IN CONSTANT MATRICES-GET TRANSPOSES                                   THE00410
      CALL MREAD(PHI,N,N)                                                    THE00420
      CALL MREAD(R,JS,JS)                                                    THE00430
      CALL MREAD (COVW,M,M)                                                  THE00450
      CALL MREAD(GAMMA,N,M)                                                  THE00460
                                                                             THE00470
                                                                             THE00480
                                                                             THE00490
```

```
      CALL TRANS(GAMMA,N,M,GAMMAT)                                      THE00500
      CALL TRANS(PHI,N,N,PHIT)                                          THE00510
C                                                                       THE00620
CC                                                                      THE00720
CCSTART THE LOOP TO ITERATE THROUGH EACH TIME SLOT                      THE00730
CC                                                                      THE00740
      ITIME=JTIME+1                                                     THE00750
      XT=36000.                                                         THE00760
      SW=XT-3000.                                                       THE00770
        DO 99 KK=1,ITIME                                                THE00780
        DO 600 I3=1,4                                                   THE00790
      XB(I3)=HYDRO(I8,3*I3-2)                                           THE00800
      YB(I3)=HYDRO(I8,3*I3-1)                                           THE00810
 600  ZB(I3)=HYDRO(I8,3*I3)                                             THE00820
      IF(XKKM1(1).GT.SW)GO TO 610                                       THE00830
      I8=I8+1                                                           THE00840
      WRITE(6,759)I8,KK                                                 THE00850
 759   FORMAT(1X,'ARRAY',2X,I2,'STARTS TRACKING AT TIME',2X,I3)         THE00860
      SW=SW-6000.                                                       THE00870
      XT=XT-6000.                                                       THE00880

 610    CALL DREAD(ZI,JS)                                               THE00990
C                                                                       THE01000
C                                                                       THE01010
C TIME VARY THE STATE TRANSITION MATRIX PHI                             THE01020
C                                                                       THE01030
      T2=(.25)*(ZI(1)+ZI(2)+ZI(3)+ZI(4))                                THE01040
      IF(KK.EQ.1)T1=T2                                                  THE01050
      PHI(1,2)=T0+(T2-T1)                                               THE01060
      PHI(3,4)=PHI(1,2)                                                 THE01070
      PHIT(2,1)=PHI(1,2)                                                THE01080
      PHIT(4,3)=PHI(1,2)                                                THE01090
      T1=T2                                                             THE01100
      A14=PHI(1,2)                                                      THE01110
C                                                                       THE01120
C                                                                       THE01140
        CALL TRREAD(TD,M)                                               THE01190
        TRUX(KK)=TD(1)                                                  THE01200
        TRUY(KK)=TD(2)                                                  THE01210
        TRUZ(KK)=TD(3)                                                  THE01220
      EPS=.00001                                                        THE01230
C CALCULATE THE H MATRIX                                                THE01240
                                                                        THE01260
      DO 120 KY=1,JS                                                    THE01270
      CALL CHROW3(KY,HROW,XB,YB,ZB)                                     THE01280
      DO 121 J=1,N                                                      THE01290
      HT(J,KY)=HROW(J)                                                  THE01300
```

```
121   H(KY,J)=HROW(J)                                              THE01310
120   CONTINUE                                                     THE01320
                                                                   THE01330
C CALCULATE THE GAIN MATRIX
                                                                   THE01350
      CALL PROD(PKKM1,HT,N,N,JS,GNUM)                              THE01360
      CALL PROD(H,PKKM1,JS,N,N,GDTEM1)                             THE01370
      CALL PROD(GDTEM1,HT,JS,N,JS,GDTEM2)                          THE01380
      CALL ADD(GDTEM2,R,JS,JS,GDENOM)                              THE01390
C     II=0                                                         THE01460
C     DO 23 IA=1,JS                                                THE01470
C        DO 23 JA=1,IA                                             THE01480
C           II=II+1                                                THE01490
C23      AA(II)=GDENOM(JA,IA)                                      THE01500
C     CALL SINV(AA,JS,EPS,IER)                                     THE01510
C     II=0                                                         THE01540
C     DO 24 IA=1,JS                                                THE01550
C        DO 24 JA=1,IA                                             THE01560
C           II=II+1                                                THE01570
C        GDENOM(JA,IA)=AA(II)                                      THE01580
C24      GDENOM(IA,JA)=AA(II)                                      THE01600
      CALL PROD(GNUM,GDENOM,N,JS,JS,G)                             THE01670
                                                                   THE01660
                                                                   THE01590
C CALCULATE THE COVARIANCE OF ESTIMATION ERROR P(K/K)
                                                                   THE01780
      CALL PROD(G,H,N,JS,N,PTEM1)                                  THE01790
      DO 127 ID=1,N                                                THE01830
         DO 127 JD=1,N                                             THE01840
127   PTEM1(ID,JD)=-1.*PTEM1(ID,JD)                                THE01850
      CALL ADD (EI,PTEM1,N,N,PTEM2)                                THE01860
      CALL PROD(PTEM2,PKKM1,N,N,N,PKK)                             THE01930
                                                                   THE01940
C GET AN ESTIMATE OF THE MEASUREMENT
C CALCULATE THE RESIDUAL
C                                                                  THE01970
      DO 122 IB=1,JS                                               THE01980
      CALL CZHAT3(IB,ZHAT,XB,YB,ZB)                                THE01990
      ZH(IB)=ZHAT                                                  THE02000
122   ZDIFF(IB)=ZI(IB)-ZHAT                                        THE02010
C                                                                  THE02020
C SET THE GATE
      PK1=ABS(PKK(1,1))                                            THE02040
      PK3=ABS(PKK(3,3))                                            THE02050
      PK5=ABS(PKK(5,5))                                            THE02060
      IF((PK1.GE.PK3).AND.(PK1.GE.PK5))P=PK1                       THE02070
      IF((PK3.GE.PK1).AND.(PK3.GE.PK5))P=PK3                       THE02080
      IF((PK5.GE.PK1).AND.(PK5.GE.PK3))P=PK5                       THE02090
```

```
CC                                                                       THE02100
      DO 500 LG=1,N                                                      THE02110
      PGATE=P/4860.                                                      THE02120
      RGATE=ABS(R(LG,LG))                                                THE02130
      GATE=3.*(SQRT(PGATE+RGATE))                                        THE02140
      IF(ZDIFF(LG).LT.GATE)GO TO 500                                     THE02150
      WRITE(6,501)LG,KK                                                  THE02160
 501  FORMAT('0','GATE HAS BEEN EXCEEDED STATE VAR',I1,'TIME',I2)       THE02170
      DO 502  LH=1,N                                                     THE02180
         DO 502  LM=1,JS                                                 THE02190
 502  G(LH,LM)=0.0                                                       THE02200
      GO TO 503                                                          THE02210
 500  CONTINUE                                                           THE02220
CC                                                                       THE02230
C CALCULATE THE ESTIMATE
C                                                                        THE02250
 503   CALL MMULT(G,ZDIFF,N,JS,GZ)                                       THE02260
      DO 124 IC=1,N                                                      THE02270
 124  XKK(IC)=XKKM1(IC)+GZ(IC)                                           THE02280
      XDIFF1(KK)=TRUX(KK)-XKK(1)                                         THE02290
      XDIFF3(KK)=TRUY(KK)-XKK(3)                                         THE02300
      XDIFF5(KK)=TRUZ(KK)-XKK(5)                                         THE02310
C                                                                        THE02330
C                                                                        THE02340
                                                                         THE02620
C CALCULATE THE PREDICTIONS FOR PKKM1
      CALL QFIND(KK,SIGACC,SIGDIV,SIGCC,A14,Q)                           THE02640
        CALL PROD(PHI,PKK,N,N,N,PHIPKK)                                  THE02710
        CALL PROD(PHIPKK,PHIT,N,N,N,PKTEMP)                              THE02720
        CALL ADD(PKTEMP,Q,N,N,PKKM1)                                     THE02730
                                                                         THE02740
C CALCULATE THE PREDICTIONS FOR XKKM1
                                                                         THE02760
        CALL MMULT(PHI,XKK,N,N,XKKM1)                                    THE02770
         DO 41 IG=1,N                                                    THE02780
 41      XP(IG,KK)=XKK(IG)                                               THE02790
        DO 38  IZ=1,N                                                    THE02800
          DO 39 JJ=1,N                                                   THE02810
 39       P1(KK,IZ,JJ)=PKK(IZ,JJ)                                        THE02820
 38     CONTINUE                                                         THE02830
 99   CONTINUE                                                           THE02860
      C1(1)=290.                                                         THE02870
      C1(2)=310.                                                         THE02880
      C3(1)=0.                                                           THE02890
      C3(2)=510.                                                         THE02900
      C4(1)=0.                                                           THE02910
      C4(2)=45000.                                                       THE02920
      C5(1)=0.                                                           THE02930
```

```
      C5(2)=10000.
      DO 91  KK=1,ITIME
      X1(KK)=XP(1,KK)
      X3(KK)=XP(3,KK)
91    X5(KK)=XP(5,KK)
         DO 92 IK=1,ITIME
92       KOUNT(IK)=IK
            DO 113  IF=1,ITIME
113         AKOUNT(IF)=FLOAT(KOUNT(IF))
      DO 238  IZ=1,N
      JJ=IZ
        DO 240  KK=1,ITIME
240     PVEC(KK)=P1(KK,IZ,JJ)
        WRITE(6,900)
        CALL  PLOTP(AKOUNT,PVEC,ITIME,0)
238    WRITE(6,800)IZ,JJ
800    FORMAT(23X,'PKK(',I1,',',I1,')','VS TIME')
900    FORMAT('1')
      WRITE(6,900)
      CALL  PLOTP(C3,C4,2,1)
      CALL  PLOTP(AKOUNT,X1,ITIME,2)
      CALL  PLOTP(AKOUNT,TRUX,ITIME,3)
      WRITE(6,801)
801   FORMAT(25X,'XPOSIT VS TIME')
      WRITE(6,900)
      CALL  PLOTP(AKOUNT,XDIFF1,ITIME,0)
      WRITE(6,805)
  805 FORMAT(25X,'DEVIATION FROM TRUE X')
      WRITE(6,900)
      CALL  PLOTP(C3,C5,2,1)
      CALL  PLOTP(AKOUNT,X3,ITIME,2)
      CALL  PLOTP(AKOUNT,TRUY,ITIME,3)
      WRITE(6,802)
802   FORMAT(25X,'YPOSIT VS TIME')
      WRITE(6,900)
      CALL  PLOTP(AKOUNT,XDIFF3,ITIME,0)
      WRITE(6,806)
  806 FORMAT(25X,'DEVIATION FROM TRUE Y')
      WRITE(6,900)
      CALL  PLOTP(C3,C1,2,1)
      CALL  PLOTP(AKOUNT,X5,ITIME,2)
      CALL  PLOTP(AKOUNT,TRUZ,ITIME,3)
      WRITE(6,803)
803   FORMAT(25X,'Z DEPTH VS TIME')
      WRITE(6,900)
      CALL  PLOTP(C4,C5,2,1)
      CALL  PLOTP(X1,X3,ITIME,2)
      CALL  PLOTP(TRUX,TRUY,ITIME,3)
```

```
      WRITE(6,804)                                                      THE03420
  804 FORMAT(25X,'GEOGRAPHIC  Y VS X')                                  THE03430
      WRITE(6,900)                                                      THE03440
      END                                                               THE03450
```

APPENDIX D - UTILITY SUBROUTINES

SUBROUTINE AUX

```
      SUBROUTINE PROD(A,B,N,M,L,C)                                      AUX00010
      DIMENSION A(N,M),B(M,L),C(N,L)                                    AUX00020
      DO 1 I=1,N                                                        AUX00030
      DO 1 J=1,L                                                        AUX00040
1     C(I,J)=0.                                                         AUX00050
      DO 2 I=1,N                                                        AUX00060
      DO 2 J=1,L                                                        AUX00070
      DO 2 K=1,M                                                        AUX00080
2     C(I,J)=C(I,J)+A(I,K)*B(K,J)                                       AUX00090
      RETURN                                                            AUX00100
      END                                                               AUX00110
      SUBROUTINE MMULT(A,B,N,C)                                         AUX00120
      DIMENSION A(N,N),B(N),C(N)                                        AUX00130
      DO 3 I=1,N                                                        AUX00140
      C(I)=0.                                                           AUX00150
      DO 4 J=1,N                                                        AUX00160
4     C(I)=C(I)+A(I,J)*B(J)                                             AUX00170
3     CONTINUE                                                          AUX00180
      RETURN                                                            AUX00190
      END                                                               AUX00200
      SUBROUTINE VMJLT(A,B,N,C)                                         AUX00210
      DIMENSION A(N),B(N)                                               AUX00220
      C=0.                                                              AUX00230
      DO 6 I=1,N                                                        AUX00240
6     C=C+A(I)*B(I)                                                     AUX00250
      RETURN                                                            AUX00260
      END                                                               AUX00270
      SUBROUTINE MREAD(A,N,M)                                           AUX00280
      DIMENSION A(N,M)                                                  AUX00290
      DO 10 I=1,N                                                       AUX00300
10    READ(3,11)(A(I,J),J=1,M)                                          AUX00310
11    FORMAT (5F10.5)                                                   AUX00320
      DO 20 I=1,N                                                       AUX00330
20    WRITE(6,21)(A(I,J),J=1,M)                                         AUX00340
21    FORMAT(5F12.8)                                                    AUX00350
      RETURN                                                            AUX00360
      END                                                               AUX00370
      SUBROUTINE TRANS(A,N,M,B)                                         AUX00380
      DIMENSION A(N,M),B(M,N)                                           AUX00390
      DO 13 I=1,N                                                       AUX00400
      DO 13 J=1,M                                                       AUX00410
13    B(J,I)=A(I,J)                                                     AUX00420
      RETURN                                                            AUX00430
      END                                                               AUX00440
      SUBROUTINE ADD (A,B,N,M,C)                                        AUX00450
      DIMENSION A(N,M),B(N,M),C(N,M)                                    AUX00460
      DO 15 I=1,N                                                       AUX00470
      DO 15 J=1,M                                                       AUX00480
```

```
15     C(I,J)=A(I,J)+B(I,J)                                    AUX00490
       RETURN                                                  AUX00500
       END                                                     AUX00510
       SUBROUTINE VREAD(A,N)                                   AUX00520
       DIMENSION A(N)                                          AUX00530
       READ(4,23) (A(I),I=1,N)                                 AUX00540
23     FORMAT(5F10.5)                                          AUX00550
       RETURN                                                  AUX00560
       END                                                     AUX00570
       SUBROUTINE CREAD(A,JS)                                  AUX00580
       DIMENSION A(JS)                                         AUX00590
       READ(2,33) (A(I),I=1,JS)                                AUX00600
33     FORMAT(4F10.5)                                          AUX00610
       RETURN                                                  AUX00620
       END                                                     AUX00630
       SUBROUTINE TRREAD(A,M)                                  AUX00640
       DIMENSION A(M)                                          AUX00650
       READ(1,43) (A(I),I=1,M)                                 AUX00660
                                                               AUX00670
43     FORMAT(3F10.5)                                          AUX00680
       RETURN                                                  AUX00690
       END                                                     AUX00700
```

```
      SUBROUTINE CZHAT(I,ZHAT)
      CCMMON XKKM1(5)
      VEL=4860.
      IF(I.EQ.1)ZHAT=1./VEL*(((XKKM1(1)+15.)**2)+((XKKM1(3)+15.)**2)
     1+((XKKM1(5)+15.)**2))**0.5
      IF(I.EQ.2)ZHAT=1./VEL*(((XKKM1(1)-15.)**2)+((XKKM1(3)+15.)**2)
     1+((XKKM1(5)+15.)**2))**0.5
      IF(I.EQ.3)ZHAT=1./VEL*(((XKKM1(1)+15.)**2)+((XKKM1(3)-15.)**2)
     1+((XKKM1(5)+15.)**2))**0.5
      IF(I.EQ.4)ZHAT=1./VEL*(((XKKM1(1)+15.)**2)+((XKKM1(3)+15.)**2)
     1+((XKKM1(5)-15.)**2))**0.5
      RETURN
      END
```

CZHAT - USED IN SINGLE ARRAY TRACKING

```
      SUBROUTINE CZHAT3 (I,ZHAT,XB,YB,ZB)
      CCMMON XKKM1(5)
      DIMENSION XB(4),YB(4),ZB(4)
      VEL=4860.
      XO=XB(I)
      YO=YB(I)
      ZO=ZB(I)
      ZHAT=(1./VEL)*(((XKKM1(1)-XO)**2)+((XK<M1(3)-YO)**2)+((XKKM1(5)
     1-ZO)**2))**0.5
      RETURN
      END
```

CZHAT3 - USED IN MULTIPLE ARRAY TRACKING

SUBROUTINE CHROW

```
      SUBROUTINE CHROW(I,HROW)
      COMMON XKKM1(5)
      DIMENSION HROW(5)
      VEL=4860.
      DENOM1=(((XKKM1(1)+15.)**2)+((XKKM1(3)+15.)**2)+((XKKM1
     1(5)+15.)**2))**0.5
      DENOM2=(((XKKM1(1)-15.)**2)+((XKKM1(3)+15.)**2)+((XKKM1
     1(5)+15.)**2))**0.5
      DENOM3=(((XKKM1(1)+15.)**2)+((XKKM1(3)-15.)**2)+((XKKM1
     1(5)+15.)**2))**0.5
      DENOM4=(((XKKM1(1)+15.)**2)+((XKKM1(3)+15.)**2)+((XKKM1
     1(5)-15.)**2))**0.5
      A1=1.
      A2=1.
      A3=1.
      DENOM=DENOM1
      IF(I.EQ.2)DENOM=DENOM2
      IF(I.EQ.3)DENOM=DENOM3
      IF(I.EQ.4)DENOM=DENOM4
      IF(I.EQ.2)A1=-1.
      H1=(1./VEL)*((XKKM1(1)+A1*15.)/DENOM)
      IF(I.EQ.3)A2=-1.
      H3=(1./VEL)*((XKKM1(3)+A2*15.)/DENOM)
      IF(I.EQ.4)A3=-1.
      H5=(1./VEL)*((XKKM1(5)+A3*15.)/DENOM)
      HROW(1)=H1
      HROW(3)=H3
      HROW(5)=H5
      DO 27 J=2,4,2
   27 HROW(J)=0.
      RETURN
      END
```

CHROW - USED IN SINGLE ARRAY TRACKING

```
      SUBROUTINE CHROW3 (I,HROW,XB,YB,ZB)
      COMMON XKKM1(5)
      DIMENSION HROW(5),XB(4),YB(4),ZB(4)
      VEL=4860.
      XO=XB(I)
      YO=YB(I)
      ZO=ZB(I)
      DENOM=(((XKKM1(1)-XO)**2)+((XKKM1(3)-YO)**2)+((XKKM1(5)-ZO)
     1**2))**0.5
      HROW(1)=(1./VEL)*((XKKM1(1)-XO)/DENOM)
      HROW(3)=(1./VEL)*((XKKM1(3)-YO)/DENOM)
      HROW(5)=(1./VEL)*((XKKM1(5)-ZO)/DENOM)
      DO 27 J=2,4,2
   27  HROW(J)=0.
      RETURN
      END
```

CHROW3 - USED IN MULTIPLE ARRAY TRACKING

```
      SUBROUTINE QFIND(K,SIGACC,SIGDIV,SIGCC,A,Q)                       QFI00010
      CCMMON XHKKM1(5),PKKM1(5,5),PKK(5,5),XHKK(5)                      QFI00020
      DIMENSION Q(5,5)                                                  QFI00030
      IF(K.NE.1) GO TO 15                                               QFI00040
      DO 10 I=1,5                                                       QFI00050
           DO 10 J=1,5                                                  QFI00060
   10   Q(I,J)=0.0                                                      QFI00070
      SIGACC=SIGACC**2                                                  QFI00080
      Q(5,5)=(SIGDIV*A)**2                                              QFI00090
      SIGCC=SIGCC**2                                                    QFI00100
      G1=(A**2)/2.0                                                     QFI00110
      G2=G1**2                                                          QFI00120
      G3=A*G1                                                           QFI00130
      A2=A**2                                                           QFI00140
   15   A1=XHKK(2)**2+XHKK(4)**2                                        QFI00150
      A3=XHKK(2)/SQRT(A1)                                               QFI00160
      B=XHKK(4)                                                         QFI00170
      C=XHKK(4)/SCRT(A1)                                                QFI00180
      D=XHKK(2)                                                         QFI00190
      E1=(A3**2)*SIGACC+(B**2)*SIGCC                                    QFI00200
      E12=A3*C*SIGACC-B*D*SIGCC                                         QFI00210
      E2=(C**2)*SIGACC+(D**2)*SIGCC                                     QFI00220
      Q(1,1)=E1*G2                                                      QFI00230
      Q(1,2)=G3*E1                                                      QFI00240
      Q(1,3)=E12*G2                                                     QFI00250
      Q(1,4)=G3*E12                                                     QFI00260
      Q(2,2)=A2*E1                                                      QFI00270
      Q(2,3)=G3*E12                                                     QFI00280
      Q(2,4)=A2*E12                                                     QFI00290
      Q(3,3)=G2*E2                                                      QFI00300
      Q(3,4)=G3*E2                                                      QFI00310
      Q(4,4)=A2*E2                                                      QFI00320
      DO 27 I=1,4                                                       QFI00330
           DO 27 J=1,I                                                  QFI00340
   27   Q(I,J)=Q(J,I)                                                   QFI00350
      RETURN                                                            QFI00360
      END                                                               QFI00370
```

LIST OF REFERENCES

1. NAVAL TORPEDO STATION REPORT 1382, NAVTORPSTA 3-D RANGE COMPUTER SOFTWARE CONVERSION PLAN, J. J. Hirschfelder, November 1977.

2. NAVAL TORPEDO STATION REPORT 1030, Principle of the Three dimensional Range, Revision 1, Enclosures (1) and (2).

3. Benson, Eric J., An Application of Kalman Filtering To Underwater Tracking, Masters Thesis, Naval Postgraduate School, December 1976.

4. Mitschang, George W., An Application of Nonlinear Filtering Theory to Passive Target Location and Tracking, PhD Thesis, Naval Postgraduate School, June 1974.

5. Kirk, Donald E., EE 4413 Optimal Control Systems Class notes (unpublished), Naval Postgraduate School, 1975.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 62 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
4.	Professor H. A. Titus, Code 62Ts Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	5
5.	Professor G. L. Sackman, Code 62Sa Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
6.	Commanding Officer Naval Underwater Weapons Engineering Station Kayport, Washington 98345	5
7.	Lieutenant Dennis M. Dwyer 4953 Klein Court Woodbridge, Virginia 22193	2

Thesis 179401
D94 Dwyer
c.1 Real time Kalman filtering for torpedo range tracking.