N O T I C E

THIS DOCUMENT HAS BEEN REPRODUCED FROM MICROFICHE. ALTHOUGH IT IS RECOGNIZED THAT CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED IN THE INTEREST OF MAKING AVAILABLE AS MUCH INFORMATION AS POSSIBLE

80-FM-32

JSC-16700

JUL 24 1980

Smoothing Filters

(NASA-TM-81033) SMOOTHING FILTERS (NASA) 58 p HC A04/MF A01 CSCL 12A

N80-29075

Unclas

G3/64 26929

Mission Planning and Analysis Division

July 1980

National Aeronautics and
Space Administration

Lyndon B. Johnson Space Center
Houston, Texas

AUG 1980
RECEIVED
NASA STI FACILITY
ACCESS DEPT.

80FM32

80-FM-32

JSC-16700

SHUTTLE PROGRAM

SMOOTHING FILTERS

By William M. Lear, TRW

JSC Task Monitor: Paul T. Pixley, Mathematical Physics Branch

Approved: Emil R. Schiesser
Emil R. Schiesser, Chief
Mathematical Physics Branch

Approved: Ronald L. Berry
Ronald L. Berry, Chief
Mission Planning and Analysis Division

Mission Planning and Analysis Division

National Aeronautics and Space Administration

Lyndon B. Johnson Space Center

Houston, Texas

July 1980

CONTENTS

Section		Page
1.0	INTRODUCTION	1
2.0	DEFINITIONS	1
3.0	THE FORWARD ESTIMATE	2
4.0	THE BACKWARD (SMOOTHED) ESTIMATE	5
5.0	THE BACKWARD EQUATIONS OF MOTION	18
6.0	EXAMPLE 1	21
7.0	EXAMPLE 2	32
8.0	MATRIX INVERSION ALGORITHMS	48
8.1	FORTRAN ALGORITHM FOR $H = H^{-1}$ WHERE $H = H^T$	49
8.2	FORTRAN ALGORITHM FOR $G = H^{-1}$ WHERE $H = H^T$	50
8.3	FORTRAN ALGORITHM FOR $H = H^{-1}$ WHERE $H \neq H^T$	51
8.4	FORTRAN ALGORITHM FOR $G = H^{-1}$ WHERE $H \neq H^T$	52
9.0	CONCLUDING REMARKS	53

PRECEDING PAGE BLANK NOT FILMED

TABLES

Table		Page
I	SMOOTHING EQUATIONS FOR OPTIMAL OR SUBOPTIMAL G_{N-i} AND $D_{N-i} = I - W_{N-i}P_{N-i}$	12
II	VERSION 1 OF THE OPTIMAL SMOOTHING EQUATIONS FOR OPTIMAL FORWARD (KALMAN) EQUATIONS	13
III	VERSION 2 OF THE OPTIMAL SMOOTHING EQUATIONS FOR OPTIMAL FORWARD (KALMAN) EQUATIONS	14
IV	VERSION 3 OF THE OPTIMAL SMOOTHING EQUATIONS FOR OPTIMAL FORWARD (KALMAN) EQUATIONS	15
V	SMOOTHING RESIDUAL COVARIANCE MATRICES	16
VI	OPTIMAL SMOOTHING EQUATIONS FOR A FIXED EPIC TIME, t_n	17

FIGURES

Figure		Page
1	Kalman filter solution for scale factor	28
2	Smoothing filter solution for scale factor	29
3	Kalman filter velocity error	30
4	Smoothing filter velocity error	31
5	Shuttle tracking geometry	32
6	Magnitude of the velocity error vector for Kalman filter	42
7	Magnitude of the velocity error vector for smoothing filter	43
8	Velocity error across staging for Kalman filter	44
9	Velocity error across staging for smoothing filter	45
10	Kalman filter solution for $\ddot{x}$ across staging	46
11	Smoothing filter solution for $\ddot{x}$ across staging	47

1.0 INTRODUCTION

A forward sequential digital filter, such as a Kalman filter, estimates a state vector from measurements that were made preceding a timepoint in question. This type of filter is suitable for processing measurement data in real time because data following current time have not yet been received. However, it is clear that the best estimate of a state vector at a given time will be obtained by using data from both sides of the timepoint in question. Thus, optimal postfactum state vector evaluation requires the use of a "smoothing" filter.

2.0 DEFINITIONS

$\underline{x}_j$ = true state vector at time t_j

$\underline{y}_j^* = \underline{g}(\underline{x}_j) + \underline{q}_j$ = measurement vector at time t_j. $\underline{q}_j$ is zero-mean, timewise-uncorrelated measurement noise

$Q_j = E(\underline{q}_j, \underline{q}_j^T)$ = measurement noise covariance matrix

$\underline{x}_{j/k}$ = estimate of x_j based on $\underline{y}_1^*, \underline{y}_2^*, \ldots, \underline{y}_k^*$

$\underline{y}_{j/j-1} = \underline{g}(\underline{x}_{j/j-1})$ = estimate of $\underline{g}(\underline{x}_j)$ based on $j-1$ measurements

$P_j = \dfrac{\partial \underline{y}_{j/j-1}}{\partial \underline{x}_{j/j-1}}$ = measurement partials matrix at time t_j

$\underline{\varepsilon}_{j/k} = \underline{x}_{j/k} - \underline{x}_j$ = state estimation error

$C_{j/k} = E(\underline{\varepsilon}_{j/k}, \underline{\varepsilon}_{j/k}^T)$ = state error covariance matrix

$\underline{x}_{j+1} = \underline{f}(\underline{x}_j) + \underline{s}_j$ = forward integration equation for true state vector; $\underline{s}_j$ is zero-mean, timewise-uncorrelated state noise.

$S_j = E(\underline{s}_j, \underline{s}_j^T)$ = state noise covariance matrix

$\underline{x}_{j+1/j} = \underline{f}(\underline{x}_{j/j})$ = forward integration equation for the estimated state vector

$\hat{\underline{x}}_{j/N} = \underline{f}^{-1}(\underline{x}_{j+1,N})$ = backward integration equation for the estimated (based on N measurements) state vector

$$\Phi_{j+1/j} = \left.\frac{\partial \underline{f}}{\partial \underline{x}}\right|_{\underline{x} = \underline{x}_{j/j}}$$ = forward state transition matrix

$\Phi_{j/j+1} = \Phi^{-1}_{j+1/j}$ = backward state transition matrix

N = total number of timepoints at which data are available

3.0 THE FORWARD ESTIMATE

Let the forward estimation equations be given by

$$\boxed{\begin{aligned} &\underline{y}_{j/j-1} = \underline{g}(\underline{x}_{j/j-1}) \\ &\underline{x}_{j/j} = \underline{x}_{j/j-1} + W_j(\underline{y}^*_j - \underline{y}_{j/j-1}) \end{aligned}} \qquad \begin{aligned} &(3.1) \\ &(3.2) \end{aligned}$$

where W_j is the measurement weighting or forward gain matrix.

The error equation associated with equation (3.2) is obtained as follows.

$$\underline{y}^*_j = \underline{g}(\underline{x}_j) + \underline{q}_j = \underline{g}(\underline{x}_{j/j-1} - \underline{\varepsilon}_{j/j-1}) + \underline{q}_j$$

Assuming that $\underline{\varepsilon}_{j/j-1}$ is small, a first-order Taylor series expansion can be made.

$$\underline{y}^*_j = \underline{g}(\underline{x}_{j/j-1}) - P_j\,\underline{\varepsilon}_{j/j-1} + \underline{q}_j$$

But

$$\underline{g}(\underline{x}_{j/j-1}) = \underline{y}_{j/j-1}$$

so equation (3.2) can be written

$$\underline{x}_j + \underline{\varepsilon}_{j/j} = \underline{x}_j + \underline{\varepsilon}_{j/j-1} + W_j(\underline{q}_j - P_j\,\underline{\varepsilon}_{j/j-1})$$

or

Thus

$$\boxed{\begin{aligned} D_j &= I - W_j P_j \\ \underline{\varepsilon}_{j/j} &= D_j\,\underline{\varepsilon}_{j/j-1} + W_j \underline{q}_j \\ C_{j/j} &= D_j\,C_{j/j-1}\,D_j^T + W_j\,Q_j\,W_j^T \end{aligned}} \qquad \begin{aligned} &(3.3) \\ &(3.4) \\ &(3.5) \end{aligned}$$

The true state vector is propagated ahead by

$$\underline{x}_{j+1} = \underline{f}(\underline{x}_j) + \underline{s}_j = \underline{f}(\underline{x}_{j/j} - \underline{\varepsilon}_{j/j}) + \underline{s}_j$$

For small estimation errors

$$\underline{x}_{j+1} = \underline{f}(\underline{x}_{j/j}) - \Phi_{j+1/j}\,\underline{\varepsilon}_{j/j} + \underline{s}_j$$

But

$$\underline{f}(\underline{x}_{j/j}) = \underline{x}_{j+1/j}$$

and

$$\underline{x}_{j+1} = \underline{x}_{j+1/j} - \underline{\varepsilon}_{j+1/j}$$

so the propagation equations are

$$\boxed{\begin{aligned} \underline{x}_{j+1/j} &= \underline{f}(\underline{x}_{j/j}) \\ \underline{\varepsilon}_{j+1/j} &= \Phi_{j+1/j}\, \underline{\varepsilon}_{j/j} - \underline{s}_j \\ C_{j+1/j} &= \Phi_{j+1/j}\, C_{j/j}\, \Phi^T_{j+1/j} + S_j \end{aligned}}$$

(3.6)

(3.7)

(3.8)

The Kalman filter equations are now completed by minimizing the quadratic form $\underline{z}^T C_{j/j}\, \underline{z}$ with respect to W_j for all nonzero vectors $\underline{z}$. This will cause $\underline{x}_{j/j}$ to be a minimum variance estimate. Let

$$a = \underline{z}^T C_{j/j}\, \underline{z} \quad \text{(a scalar)}$$

Using equation 3.5

$$\begin{aligned} \frac{\partial a}{\partial W_j} &= \underline{z}^T\big(-P_j\, C_{j/j-1}(I - W_j\, P_j)^T - (I - W_j\, P_j)C_{j/j-1}\, P_j^T \\ &\quad + Q_j\, W_j^T + W_j\, Q_j\big)\, \underline{z} \\ &= \underline{z}^T\, \big(-P_j\, C_{j/j-1}(I - W_j\, P_j)^T + Q_j\, W_j^T\big)\, \underline{z} \\ &\quad + \underline{z}^T\, \big(-\,(I - W_j\, P_j)C_{j/j-1}\, P_j^T + W_j\, Q_j\big)\, \underline{z} \end{aligned}$$

This will equal zero if

$$(I - W_j\, P_j)\, C_{j/j-1}\, P_j^T = W_j\, Q_j \tag{3.9}$$

or if

$$\boxed{W_j = C_{j/j-1} P_j^T (P_j C_{j/j-1} P_j^T + Q_j)^{-1}} \tag{3.10}$$

which is the optimal forward weighting matrix. Note that the predicted residual variance is

$$\boxed{E\left((\underline{y}_j^* - \underline{y}_{j/j-1})(\underline{y}_j^* - \underline{y}_{j/j-1})^T\right) = P_j C_{j/j-1} P_j^T + Q_j} \tag{3.11}$$

This information, free of charge, can be used to edit, say, 6σ residuals.

Substituting equations (3.9) and (3.10) into equation (3.5) yields

$$C_{j/j} = D_j C_{j/j-1} = C_{j/j-1} D_j^T \tag{3.12}$$

Or, using equations (3.3) and (3.10) for the optimum W_j,

$$\boxed{C_{j/j} = C_{j/j-1} - C_{j/j-1} P_j^T (P_j C_{j/j-1} P_j^T + Q_j)^{-1} P_j C_{j/j-1}} \tag{3.13}$$

This is the equation for $C_{j/j}$ when W_j is optimum.

4.0 THE BACKWARD (SMOOTHED) ESTIMATE

For $i = 1, 2, \ldots, N-1$ let the backward, or smoothed, estimate of the state be given by

$$\hat{\underline{x}}_{N-i/N} = \underline{f}^{-1}(\underline{x}_{N-i+1/N}) \tag{4.1}$$

$$\underline{x}_{N-i/N} = \hat{\underline{x}}_{N-i/N} + B_{n-i}(\underline{x}_{N-i/N-i} - \hat{\underline{x}}_{N-i/N}) \tag{4.2}$$

where B_{N-i} is the backward gain matrix to be derived.

For the purpose of deriving error equations, equation (4.2) will be modified in the following manner. From equation (4.1), it is seen that

$$\underline{x}_{N-i/N} = \underline{f}^{-1}\left(\underline{x}_{N-i+1/N-i} + (\underline{x}_{N-i+1/N} - \underline{x}_{N-i+1/N-i})\right)$$

For small state estimation errors

$$\hat{\underline{x}}_{N-i/N} = \underline{x}_{N-i/N-i} + \Phi_{N-i/N-i+1}\,(\underline{x}_{N-i+1/N} - \underline{x}_{N-i+1/N-i}) \tag{4.3}$$

Substitution into equation (4.2) yields

$$\underline{x}_{N-i/N} = \hat{\underline{x}}_{N-i/N} + G_{N-i}(\underline{x}_{N-i+1/N-i} - \underline{x}_{N-i+1/N}) \tag{4.4}$$

or

$$\underline{x}_{N-i/N} = \underline{x}_{N-i/N-i} - (\Phi_{N-i/N-i+1} - G_{N-i})\,(\underline{x}_{N-i+1/N-i} - \underline{x}_{N-i+1/N}) \tag{4.5}$$

where the backward gain matrix G_{N-i} is

$$G_{N-i} = B_{N-i}\,\Phi_{N-i/N-i+1} \tag{4.6}$$

The error equation associated with $\underline{x}_{N-i/N}$ is obtained from equation (4.5)

$$\underline{\varepsilon}_{N-i/N} = \underline{\varepsilon}_{N-i/N-i} + (\Phi_{N-i/N-i+1} - G_{n-i})(\underline{\varepsilon}_{N-i+1/N} - \underline{\varepsilon}_{N-i+1/N-i})$$

But from equation (3.7), it is seen that

$$\underline{\varepsilon}_{N-i/N-i} = \Phi_{N-i/N-i+1}(\underline{\varepsilon}_{N-i+1/N-i} + \underline{s}_{N-i})$$

Thus, the state error propagation equation is

$$\boxed{\begin{aligned} \underline{\varepsilon}_{N-i/N} = {} & (\Phi_{N-i/N-i+1} - G_{N-i})\, \underline{\varepsilon}_{N-i+1/N} \\ & + G_{N-i}\, \underline{\varepsilon}_{N-i+1/N-i} \\ & + \Phi_{N-i/N-i+1}\, \underline{s}_{N-i} \end{aligned}}$$

In order to evaluate

$$C_{N-i/N} = E(\underline{\varepsilon}_{N-i/N}\, \underline{\varepsilon}^T_{N-i/N})$$

expressions for

$$E(\underline{\varepsilon}_{N-i+1/N}\, \underline{\varepsilon}^T_{N-i+1/N-i})$$

and

$$E(\underline{\varepsilon}_{N-i+1/N}\, \underline{s}^T_{N-i})$$

are needed. To obtain these, define the matrix H_{N-i} $(i = 2, 3, \ldots, N-1)$ by

$$H_{N-i} = \left((\Phi_{N-i+1/N-i+2} - G_{N-i+1})\, H_{N-i+1} + G_{N-i+1}\right) \Phi_{N-i+2/N-i+1}\, D_{N-i+1}$$

and

$$H_{N-i} = D_N = I - W_N P_N \qquad \text{for} \quad i = 1 \tag{4.8}$$

Using the state error propagation equations (3.4), (3.7) and (4.7), it can be shown that

$$E\left(\underline{\varepsilon}_{N-i+1/N}\, \underline{\varepsilon}^T_{N-i+1/N-i}\right) = H_{N-i}\, C_{N-i+1/N-i} \tag{4.9}$$

and

$$E\left(\underline{\varepsilon}_{N-i+1/N}\, \underline{s}^T_{N-i}\right) = -H_{N-i}\, S_{N-i} \tag{4.10}$$

It can now easily be shown that

$$\begin{aligned} C_{N-i/N} = {} & \Phi_{N-i/N-i+1}\left(C_{N-i+1/N} + (I - H_{N-i})\, S_{N-i} - S_{N-i}\, H^T_{N-i}\right) \Phi^T_{N-i/N-i+1} \\ & - G_{N-i}\left(C_{N-i+1/N} + (I - H_{N-i})\, S_{N-i} - C_{N-i+1/N-i}\, H^T_{N-i}\right) \Phi^T_{N-i/N-i+1} \\ & - \Phi_{N-i/N-i+1}\left(C_{N-i+1/N} + S_{N-i}(I - H^T_{N-i}) - H_{N-i}\, C_{N-i+1/N-i}\right) G^T_{N-i} \\ & + G_{N-i}\left(C_{N-i+1/N} + C_{N-i+1/N-i}\, (I - H^T_{N-i}) - H_{N-i}\, C_{N-i+1/N-i}\right) G^T_{N-i} \end{aligned} \tag{4.11}$$

The backward, or smoothing, equations for optimal or suboptimal W's and G's are summarized in table I.

The matrix G_{N-i} will now be chosen so as to minimize the quadratic form, $\underline{z}^T C_{N-i/N}\,\underline{z}$, for all $\underline{z}$. $\underline{x}_{N-i/N}$ will then be a minimum variance estimate of $\underline{x}_{N-i}$. The value of G_{N-i} (which accomplishes this) is easily seen to be

$$G_{N-i} = \Phi_{N-i/N-i+1}\left(C_{N-i+1/N} + S_{N-i}(I - H^T_{N-i}) - H_{N-i}\,C_{N-i+1/N-i}\right)$$
$$\times\left(C_{N-i+1/N} + C_{N-i+1/N-i}(I - H^T_{N-i}) - H_{N-i}\,C_{N-i+1/N-i}\right)^{-1} \tag{4.12}$$

It is also easily seen that

$$\Phi_{n-i/N-i+1} - G_{N-i} = \Phi_{N-i/N-i+1}\left(C_{N-i+1/N-i} - S_{N-i}\right)(I - H^T_{N-i})$$
$$\times\left(C_{N-i+1/N} + C_{N-i+1/N-i}\,(I - H^T_{N-i})\right.$$
$$\left. - H_{N-i}\,C_{N-i+1/N-i}\right)^{-1} \tag{4.13}$$

Substituting the optimum value of G_{N-i}, given by equation (4.12), into equation (4.11) yields

$$C_{N-i/N} = \Phi_{N-i/N-i+1}\left(C_{N-i+1/N} + (I - H_{N-i})S_{N-i} - S_{N-i}\,H^T_{N-i}\right)\Phi^T_{N-i/N-i+1}$$
$$- G_{N-i}\left(C_{N-i+1/N} + (I - H_{N-i})\,S_{N-i} - C_{N-i+1/N-i}\,H^T_{N-i}\right)\Phi^T_{N-i/N-i+1} \tag{4.14}$$

From the equations using the optimal forward gain matrix (equations (3.9), (3.10), and (3.12)) it can be shown, using proof by induction, that

$$H_{N-i} = C_{N-i+1/N}\,C^{-1}_{N-i+1/N-i} \tag{4.15}$$

The previous equations may be combined in several ways to give the optimal smoothing equations. Three versions are presented here in tables II, III, and IV.

Sometimes, as a check on the functioning of the filter, it is desired to see how well residual statistics agree with their predicted covariance matrix. Table V shows three forms of residuals with their associated covariance matrices.

Suppose that a smoothed estimate of the state is required at the fixed epic time, t_n. Using version 3 of the smoothing equations (table IV), it can be shown that the equations in table VI result. Note that these equations operate <u>forward</u> in time in conjunction with the normal forward (Kalman) filter equations.

The question arises as to which version of the smoothing equations is best. For problems with nonlinear dynamics, the author prefers version 1. Version 2 is simpler and faster, but it has a residual with a time tag that is not at the correction time. An additional linearity assumption has been made in this version, namely

$$\underline{x}_{N-i+1/N-i} - \underline{x}_{N-i+1/N} = \Phi_{N-i+1/N-i}(\underline{x}_{N-i/N-i} - \hat{\underline{x}}_{N-i/N}) \tag{4.16}$$

Version 3 also suffers from this defect. Version 3 is relatively simple and needs no backward integrator. However, take the limiting case where the te noise covariance is zero. Versions 1 and 2 correctly integrate the state backward by

$$\underline{x}_{N-i/N} = \underline{f}^{-1}(\underline{x}_{N-i+1/N}) \tag{4.17}$$

But, version 3 integrates backward by

$$\underline{x}_{N-i/N} = \underline{x}_{N-i/N-i} + \Phi_{N-i/N-i+1}(\underline{x}_{N-i+1/N} - \underline{x}_{N-i+1/N-i}) \tag{4.18}$$

which is not as accurate as equation (4.17). In the case of linear dynamics,

$$\underline{x}_{N-i/N} = \Phi_{N-i/N-i+1}\ \underline{x}_{N-i+1/N} \tag{4.19}$$

and all versions give the same answers except for computer roundoff error. In the case of linear dynamics, the author prefers version 3 of the smoothing equations.

A final case needs to be mentioned. The forward Kalman filter frequently processes successive scalar measurements at a single timepoint rather than the measurement vector. That is, a four-element measurement vector may be processed as four scalar measurements, all at the same timepoint. This is equivalent to setting $\Phi = I$ and $S = 0$ between measurements. For the smoothing

equations, save only the results after the last scalar measurement has been processed. The results at this time will be equivalent to processing all measurements at once as a vector.

TABLE I.- SMOOTHING EQUATIONS FOR OPTIMAL OR SUBOPTIMAL G_{N-i} and $D_{N-i} = I - W_{N-i}P_{N-i}$

<u>Initialize</u>

$$H_{N-i} = D_N = I - W_N P_N \qquad C_{N-i+1/N} = C_{N/N} \qquad \underline{x}_{N-i+1/N} = \underline{x}_{N/N}$$

<u>For $i = 1, 2, \ldots, N-1$</u>

$$\hat{\underline{x}}_{N-i} = \underline{f}^{-1}(\underline{x}_{N-i+1/N})$$

$$\Phi_{N-i+1/N-i} = \left.\frac{\partial \underline{f}}{\partial \underline{x}}\right|_{\underline{x} = \underline{x}_{N-i/N-i}} \qquad \Phi_{N-i/N-i+1} = \Phi^{-1}_{N-i+1/N-i}$$

$$\underline{x}_{N-i} = \hat{\underline{x}}_{N-i/N} + G_{N-i}\,\Phi_{N-i+1/N-i}(\underline{x}_{N-i/N-i} - \hat{\underline{x}}_{N-i/N})$$

$$\begin{aligned} C_{N-i/N} = {} & \Phi_{N-i/N-i+1}(C_{N-i+1/N} + (I - H_{N-i})S_{N-i} - S_{N-i}\,H^T_{N-i})\Phi^T_{N-i/N-i+1} \\ & - G_{N-i}(C_{N-i+1/N} + (I - H_{N-i})S_{N-i} - C_{N-i+1/N-i}\,H^T_{N-i})\Phi^T_{N-i/N-i+1} \\ & - \Phi_{N-i/N-i+1}(C_{N-i+1/N} + S_{N-i}(I - H^T_{N-i}) - H_{N-i}\,C_{N-i+1/N-i})G^T_{N-i} \\ & + G_{N-i}(C_{N-i+1/N} + C_{N-i+1/N-i}(I - H^T_{N-i}) - H_{N-i}\,C_{N-i+1/N-i})G^T_{N-i} \end{aligned}$$

$$H_{N-i-1} = ((\Phi_{N-i/N-i+1} - G_{N-i})\,H_{N-i} + G_{N-i})\Phi_{N-i+1/N-i}\,D_{N-i}$$

TABLE II.- VERSION 1 OF THE OPTIMAL SMOOTHING EQUATIONS FOR OPTIMAL FORWARD (KALMAN) EQUATIONS

Initialize

$$C_{N-i+1/N} = C_{N/N} \qquad \underline{x}_{N-i+1/N} = \underline{x}_{N/N}$$

For $i = 1, 2, \ldots, N-1$

$$\hat{\underline{x}}_{N-i/N} = \underline{f}^{-1}(\underline{x}_{N-i+1/N})$$

$$\Phi_{N-i+1/N-i} = \left.\frac{\partial \underline{f}}{\partial \underline{x}}\right|_{\underline{x} = \underline{x}_{N-i/N-i}} \qquad \Phi_{N-i/N-i+1} = \Phi^{-1}_{N-i+1/N-i}$$

$$B_{N-i} = \Phi_{N-i/N-i+1}\, S_{N-i}\, \Phi^{T}_{N-i/N-i+1}(C_{N-i/N-i} + \Phi_{N-i/N-i+1}\, S_{N-i}\, \Phi^{T}_{N-i/N-i+1})^{-1}$$

$$\underline{x}_{N-i/N} = \hat{\underline{x}}_{N-i/N} + B_{N-i}(\underline{x}_{N-i/N-i} - \hat{\underline{x}}_{N-i/N})$$

$$\begin{aligned} C_{N-i/N} &= C_{N-i/N-i} - (I - B_{N-i})\Phi_{N-i/N-i+1}(C_{N-i+1/N-i} - C_{N-i+1/N})\, \Phi^{T}_{N-i/N-i+1}(I - B_{N-i})^{T} \\ &= (I - B_{N-i})\Phi_{N-i/N-i+1}\, C_{N-i+1/N}\, \Phi^{T}_{N-i/N-i+1}(I - B_{N-i})^{T} + B_{N-i}\, C_{N-i/N-i} \\ &= (I - B_{N-i})\Phi_{N-i/N-i+1}\, C_{N-i+1/N}\, \Phi^{T}_{N-i/N-i+1}\, (I - B_{N-i})^{T} + (I - B_{N-i})\, \Phi_{N-i/N-i+1}\, S_{N-i}\, \Phi^{T}_{N-i/N-i+1} \end{aligned}$$

NOTE: $I - B_{N-i} = C_{N-i/N-i}\,(C_{N-i/N-i} + \Phi_{N-i/N-i+1}\, S_{N-i}\, \Phi^{T}_{N-i/N-i+1})^{-1}$

TABLE III.- VERSION 2 OF THE OPTIMAL SMOOTHING EQUATIONS FOR OPTIMAL FORWARD (KALMAN) EQUATIONS

<u>Initialize</u>

$$C_{N-i+1/N} = C_{N/N} \qquad \underline{x}_{N-i+1/N} = \underline{x}_{N/N}$$

<u>For $i = 1, 2, \ldots, N-1$</u>

$$\hat{\underline{x}}_{N-i/N} = \underline{f}^{-1}(\underline{x}_{N-i+1/N})$$

$$\Phi_{N-i+1/N-i} = \left.\frac{\partial \underline{f}}{\partial \underline{x}}\right|_{\underline{x} = \underline{x}_{N-i/N-i}} \qquad \Phi_{N-i/N-i+1} = \Phi^{-1}_{N-i+1/N-i}$$

$$G_{N-i} = \Phi_{N-i/N-i+1}\, S_{N-i}\, C^{-1}_{N-i+1/N-i}$$

$$\underline{x}_{N-i/N} = \hat{\underline{x}}_{N-i/N} + G_{N-i}(\underline{x}_{N-i+1/N-i} - \underline{x}_{N-i+1/N})$$

$$\begin{aligned} C_{N-i/N} &= C_{N-i/N-i} - (\Phi_{N-i/N-i+1} - G_{N-i})(C_{N-i+1/N-i} \\ &\quad - C_{N-i+1/N})\,(\Phi_{N-i/N-i+1} - G_{N-i})^T \\ &= (\Phi_{N-i/N-i+1} - G_{N-i})C_{N-i+1/N}(\Phi_{N-i/N-i+1} - G_{N-i})^T \\ &\quad + (\Phi_{N-i/N-i+1} - G_{N-i})\, S_{N-i}\, \Phi^T_{N-i/N-i+1} \end{aligned}$$

<u>NOTE</u>: $G_{N-i} = B_{N-i}\, \Phi_{N-i/N-i+1}$

TABLE IV.- VERSION 3 OF THE OPTIMAL SMOOTHING EQUATIONS FOR OPTIMAL FORWARD (KALMAN) EQUATIONS

Initialize

$$C_{N-i+1/N} = C_{N/N} \qquad \underline{x}_{N-i+1/N} = \underline{x}_{N/N}$$

For $i = 1, 2, \ldots, N-1$

$$\Phi_{N-i+1/N-i} = \left.\frac{\partial \underline{f}}{\partial \underline{x}}\right|_{\underline{x} = \underline{x}_{N-i/N-i}} \qquad \Phi_{N-i/N-i+1} = \Phi^{-1}_{N-i+1/N-i}$$

$$R_{N-i} = C_{N-i/N-i}\, \Phi^{T}_{N-i+1/N-i}\, C^{-1}_{N-i+1/N-i}$$

$$= \Phi_{N-i/N-i+1}(I - S_{N-i}\, C^{-1}_{N-i+1/N-i}) \quad \text{(preferred form)}$$

$$\underline{x}_{N-i/N} = \underline{x}_{N-i/N-i} + R_{N-i}(\underline{x}_{N-i+1/N} - \underline{x}_{N-i+1/N-i})$$

$$C_{N-i/N} = C_{N-i/N-i} - R_{N-i}(C_{N-i+1/N-i} - C_{N-i+1/N})R^{T}_{N-i}$$

$$= R_{N-i}\, C_{N-i+1/N}\, R^{T}_{N-i} + R_{N-i}\, S_{N-i}\, \Phi^{T}_{N-i/N-i+1}$$

NOTE: $R_{N-i} = (I - B_{N-i})\, \Phi_{N-i/N-i+1} = \Phi_{N-i/N-i+1} - G_{N-i}$

TABLE V.- SMOOTHING RESIDUAL COVARIANCE MATRICES

The covariance matrix of the residual $\underline{x}_{N-i/N-i} - \hat{\underline{x}}_{N-i/N}$ is

$$\Phi_{N-i/N-i+1}(C_{N-i+1/N-i} - C_{N-i+1/N})\,\Phi^{T}_{N-i/N-i+1}$$

The covariance matrix of the residual $\underline{x}_{N-i/N-i} - \underline{x}_{N-i/N}$ is

$$(I - B_{N-i})\Phi_{N-i/N-i+1}(C_{N-i+1/N-i} - C_{N-i+1/N})\,\Phi^{T}_{N-i/N-i+1}\,(I - B_{N-i})^{T}$$

$$= (\Phi_{N-i/N-i+1} - G_{N-i})(C_{N-i+1/N-i} - C_{N-i+1/N})(\Phi_{N-i/N-i+1} - G_{N-i})^{T}$$

The covariance matrix of the residual $\underline{x}_{N-i+1/N-i} - \underline{x}_{N-i+1/N}$ is

$$C_{N-i+1/N-i} - C_{N-i+1/N}$$

TABLE VI.- OPTIMAL SMOOTHING EQUATIONS[a] FOR A FIXED EPIC TIME, t_n

At time $t = t_n$ initialize

$$C_{n/n+j-1} = C_{n/n} \qquad \underline{x}_{n/n+j-1} = \underline{x}_{n/n} \qquad K_{n+j-1} = I$$

For $j = 1, 2, 3, \ldots$

$$\Phi_{n+j/n+j-1} = \left.\frac{\partial \underline{f}}{\partial \underline{x}}\right|_{\underline{x} = \underline{x}_{n+j-1/n+j-1}} \qquad \Phi_{n+j-1/n+j} = \Phi^{-1}_{n+j/n+j-1}$$

$$R_{n+j-1} = C_{n+j-1/n+j-1}\, \Phi^{T}_{n+j/n+j-1}\, C^{-1}_{n+j/n+j-1}$$

$$= \Phi_{n+j-1/n+j}\, (I - S_{n+j-1}\, C^{-1}_{n+j/n+j-1}) \qquad \text{(preferred form)}$$

$$K_{n+j} = K_{n+j-1}\, R_{n+j-1}$$

$$\underline{x}_{n/n+j} = \underline{x}_{n/n+j-1} + K_{n+j}\, (\underline{x}_{n+j/n+j} - \underline{x}_{n+j/n+j-1})$$

$$C_{n/n+j} = C_{n/n+j-1} - K_{n+j}\, (C_{n+j/n+j-1} - C_{n+j/n+j}) K^{T}_{n+j}$$

[a]Note that the above equations operate <u>forward</u> in time in conjunction with the normal forward (Kalman) filter equations.

5.0 THE BACKWARD EQUATIONS OF MOTION

Many (perhaps most) Kalman filters contain exponentially correlated random state variables in the state vector. In this case, backward integration is more involved than just setting ΔT negative. A simple example will illustrate the problem.

Let the acceleration of a point be $a_x(t)$. Let a_x be an exponentially correlated random variable, constant over each integration step. Position, velocity and acceleration states are integrated forward in the Kalman filter by

$$x_{i+1} = x_i + \dot{x}_i \, \Delta T + a_{x,i} \, \Delta T^2/2 \tag{5.1}$$

$$\dot{x}_{i+1} = \dot{x}_i + a_{x,i} \, \Delta T \tag{5.2}$$

$$a_{x,i+1} = a \, a_{x,i} \tag{5.3}$$

where

$$a = \exp(-\Delta T/\tau_{ax}) \quad (0 < a \leq 1) \tag{5.4}$$

Solving the above equations for x_i, $\dot{x}_i$ and $a_{x,i}$ yields the backward integration equations

$$x_i = x_{i+1} - \dot{x}_{i+1} \, \Delta T + \frac{1}{a} a_{x,i+1} \, \Delta T^2/2 \tag{5.5}$$

$$\dot{x}_i = \dot{x}_{i+1} - \frac{1}{a} a_{x,i+1} \, \Delta T \tag{5.6}$$

$$a_{x,i} = \frac{1}{a} a_{x,i+1} \tag{5.7}$$

where $\Delta T = t_{i+1} - t_i \geq 0$. To integrate backwards, the rules are this: for the nonexponentially correlated states

a. Exchange the i and $i+1$ subscripts

b. Set $\Delta T = -\Delta T$

c. Replace a_x with $\frac{1}{a}a_x$

For the exponentially correlated state, integrate backward by using

d. $$a_{x,i} = \frac{1}{a}a_{x,i+1}$$

The above example was for linear dynamical equations where the inverse solution was easily obtained. However, for complex nonlinear dynamics, the rules are precisely the same. For example, consider that the acceleration of a point is given by the nonlinear differential equation

$$\ddot{x} = a_x(t) - k(t)\dot{x}^2 \tag{5.9}$$

where $a_x(t)$ and $k(t)$ are exponentially correlated random variables, <u>constant</u> across the integration step. Note that $k(t)$ symbolizes a time variable drag coefficient. Note that

$$\dddot{x} = -2k\dot{x}\ddot{x} = -2k\dot{x}(a_x - k\dot{x}^2) \tag{5.10}$$

The forward equations of motion are easily seen to be

$$x_{i+1} = x_i + \dot{x}_i\,\Delta T + (a_{x,i} - k_i\,\dot{x}_i^2)\,\frac{\Delta T^2}{2} - 2k_i\,\dot{x}_i(a_{x,i} - k_i\,\dot{x}_i^2)\,\frac{\Delta T^3}{6} \tag{5.11}$$

$$\dot{x}_{i+1} = \dot{x}_i + (a_{x,i} - k_i\,\dot{x}_i^2)\,\Delta T - 2k_i\,\dot{x}_i\,(a_{x,i} - k_i\,\dot{x}_i^2)\,\Delta T^2/2 \tag{5.12}$$

$$a_{x,i+1} = a_1\,a_{x,i} \tag{5.13}$$

$$k_{i+1} = a_2\,k_i \tag{5.14}$$

where

$$a_1 = \exp(-\Delta T/\tau_{ax}) \tag{5.15}$$

$$a_2 = \exp(-\Delta T/\tau_k) \tag{5.16}$$

To obtain the backward equations of motion, merely apply the previously defined simple rules.

$$x_i = x_{i+1} - \dot{x}_{i+1}\,\Delta T + \left(\frac{1}{a_1}\,a_{x,i+1} - \frac{1}{a_2}\,k_{i+1}\,\dot{x}^2_{i+1}\right)\Delta T^2/2$$

$$+\,2\,\frac{1}{a_2}\,k_{i+1}\,\dot{x}_{i+1}\left(\frac{1}{a_1}\,a_{x,i+1} - \frac{1}{a_2}\,k_{i+1}\,\dot{x}^2_{i+1}\right)\Delta T^3/6 \tag{5.17}$$

$$\dot{x}_i = \dot{x}_{i+1} - \left(\frac{1}{a_1}\,a_{x,i+1} - \frac{1}{a_2}\,k_{i+1}\,\dot{x}^2_{i+1}\right)\Delta T$$

$$-\,2\frac{1}{a_2}\,k_{i+1}\,\dot{x}_{i+1}\left(\frac{1}{a_1}\,a_{x,i+1} - \frac{1}{a_2}\,k_{i+1}\,\dot{x}^2_{i+1}\right)\Delta T^2/2 \tag{5.18}$$

$$a_{x,i} = \frac{1}{a_1}\,a_{x,i+1} \tag{5.19}$$

$$k_i = \frac{1}{a_2}\,k_{i+1} \tag{5.20}$$

where ΔT above is a positive quantity.

Finally a word about the inverse of the state transition matrix. $\Phi_{N-i+1/N-i}$ is almost always of the form

$$\Phi_{N-i+1/N-i} = \begin{bmatrix} \Phi_1 & \Phi_2 \\ 0 & \Phi_4 \end{bmatrix} \tag{5.27}$$

where Φ_4 is frequently diagonal. The inverse is then easily seen to be

$$\Phi_{N-i/N-i+1} = \begin{bmatrix} \Phi_1^{-1} & -\Phi_1^{-1} \; \Phi_2 \; \Phi_4^{-1} \\ 0 & \Phi_4^{-1} \end{bmatrix} \tag{5.28}$$

6.0 EXAMPLE 1

Very precise altitude measurements will be made of a falling sphere in order to determine the density of the atmosphere. The sphere will be dropped from an altitude of $h = 11\,000$ meters. Altitude measurements will be made every 0.1 second. The altitude measurement random noise will have a standard deviation of

$$\sigma q = 0.1 \text{ meter} \tag{6.1}$$

The equation of motion is given by

$$x = -a_g + \frac{1}{2} C_D \frac{A}{W} \rho \dot{x}^2 \tag{6.2}$$

where the acceleration due to gravity is

$$a_g = 9.8 \text{ m/sec}^2$$

For $h \leq 11\,000$ meters the atmosphere density is assumed to be given by

$$\rho = (1 + \delta)\, \rho_0\, (1 - 0.0065h/288.15)^{4.2559} \tag{6.4}$$

where

$$\rho_0 = 1.2250 \text{ kg/m}^3 \tag{6.5}$$

Equation (6.4) with $\delta = 0$ is the equation for the 1962 Standard Atmosphere. 100δ is the percentage deviation from the standard. δ will be a state variable in the filter.

Let

$$C_D \frac{A}{W} = 0.01 \frac{m^2}{kg} \tag{6.6}$$

Then equation (6.2) becomes

$$\boxed{\ddot{x} = -9.8 + 0.006125(1 - 0.0065x/288.15)^{4.2559}(1 + \delta)\dot{x}^2} \tag{6.7}$$

From equation (6.7) (with $x = 0$) the ground ($x = h = 0$) impact velocity is seen to be about 40 m/sec = 89 mph. At an altitude of $x = h = 11\ 000$ meters, the steady state velocity is about 73.4 m/sec = 164 mph. The sphere takes about 207 seconds to reach the ground (with $\delta = 0$).

For the real world, δ will be chosen to be

$$\delta = 0.05 \cos\left(\frac{2\pi t}{200}\right) \tag{6.8}$$

Thus the real-world density will deviate from the 1962 Standard Atmosphere by ± 5 percent. For the filter world, δ will be taken as an exponentially correlated random variable whose standard deviation is 0.035 (3.5 percent) with a time constant of 100 seconds. That is, for the filter world

$$\delta_{i+1} = a\,\delta_i + \sigma_\delta \sqrt{1 - a^2}\ \eta_i \tag{6.9}$$

$$a = \exp(-\Delta T/\tau) \tag{6.10}$$

$$\tau = 100 \text{ seconds}$$

$$\sigma_\delta = 0.035$$

η is zero-mean, timewise-uncorrelated, state noise with a unit variance.

The state vector, $\underline{x}$, will be

$$\underline{x} = \begin{bmatrix} x = \text{altitude, } h \\ \dot{x} = \dot{h} \\ \delta = \text{scale factor error} \end{bmatrix} \tag{6.11}$$

For purposes of obtaining the state transition matrix, the following equations of motion are used.

$$x_{i+1} = x_i + \dot{x}_i \, \Delta T + \ddot{x}_i \, \Delta T^2/2 \tag{6.12}$$

$$\dot{x}_{i+1} = \dot{x}_i + \ddot{x}_i \, \Delta T \tag{6.13}$$

$$\delta_{i+1} = a \, \delta_i \tag{6.14}$$

where $\Delta T = 0.1$ second and $\ddot{x}$ was given by equation (6.7). Elements of the forward state transition matrix are

$$\Phi_{11} = \frac{\partial x_{i+1}}{\partial x_i} = 1 + \frac{\partial \ddot{x}_i}{\partial x_i} \Delta T^2/2 \tag{6.15}$$

$$\Phi_{12} = \frac{\partial x_{i+1}}{\partial \dot{x}_i} = \Delta T + \frac{\partial \ddot{x}_i}{\partial \dot{x}_i} \Delta T^2/2 \tag{6.16}$$

$$\Phi_{13} = \frac{\partial x_{i+1}}{\partial \delta_i} = \frac{\partial \ddot{x}_i}{\partial \delta_i} \Delta T^2/2 \tag{6.17}$$

$$\Phi_{21} = \frac{\partial \dot{x}_{i+1}}{\partial x_i} = \frac{\partial \ddot{x}_i}{\partial x_i} \Delta T \tag{6.18}$$

$$\Phi_{22} = \frac{\partial \dot{x}_{i+1}}{\partial \dot{x}_i} = 1 + \frac{\partial \ddot{x}_i}{\partial \dot{x}_i} \Delta T \tag{6.19}$$

$$\Phi_{23} = \frac{\partial \dot{x}_{i+1}}{\partial \delta_i} = \frac{\partial \ddot{x}_i}{\partial \delta_i} \Delta T \tag{6.20}$$

$$\Phi_{31} = \frac{\partial \delta_{i+1}}{\partial x_i} = 0 \tag{6.21}$$

$$\Phi_{32} = \frac{\partial \delta_{i+1}}{\partial \dot{x}_i} = 0 \tag{6.22}$$

$$\Phi_{33} = \frac{\partial \delta_{i+1}}{\partial \delta_i} = a \tag{6.23}$$

where

$$\frac{\partial \ddot{x}_i}{\partial x_i} \approx 0 \qquad \text{(x error contribution small)} \tag{6.24}$$

$$\frac{\partial \ddot{x}_i}{\partial x_i} = 0.01225 \quad (1 - 0.0065x/288.15)^{4.2559}(1 + \delta_i)\dot{x}_i \tag{6.25}$$

$$\frac{\partial \ddot{x}_i}{\partial \delta_i} = 0.006125 \ (1 - 0.0065x/288.15)^{4.2559}\dot{x}_i^2 \tag{6.26}$$

For both the real world and the filter, a fourth-order Nystrom-Lear integrator was used to integrate equation (6.7) to give x and $\dot{x}$. In the real world equation (6.8) gave δ as a function of time. In the filter integrator, δ is taken as a constant across the integration step. That is, for the forward filter integrator

$\Delta T = 0.1$

Integrate $\ddot{x} = f(x, \dot{x}, \delta_i)$ δ_i constant

$\delta_{i+1} = a\,\delta_i$

$T = T + \Delta T$

For the backward integrator in the filter

$\Delta T = -0.1$

$\delta_i = \delta_{i+1}/a$

Integrate $\ddot{x} = f(x, \dot{x}, \delta_i)$ δ_i constant

$T = T + \Delta T$

Note that the backward integration of $\ddot{x} = f(x, \dot{x}, \delta)$ is not "stable" and thus is a good test of the smoothing equations. The reason for the instability is that the downward fall of the sphere quickly reaches a "steady-state" condition, irrespective of the initial conditions. Because of computer roundoff error, the backward integration of the fall will generally yield absurd initial conditions.

From equation (6.9), the state noise covariance matrix is seen to be

$$S = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sigma_\delta^2(1-a^2) \end{bmatrix} \tag{6.27}$$

where

$$\sigma_\delta^2(1-a^2) = 0.24475\ 51634 \cdot 10^{-5} \tag{6.28}$$

Because measurements of x are made, the measurement partials matrix is

$$P = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \tag{6.29}$$

The logistics of the computer program are not simple. They are outlined below.

a Initialize: T = initial measurement time, $\underline{x}_{n+1/n}$, $C_{n+1/n}$, etc.

Process measurement to give $\underline{x}_{n/n}$, $C_{n/n}$.

Print T, $\underline{x}_{n/n}$, $C_{n/n}$, etc.

If last measurement go to b.

Compute $\Phi_{n+1/n}$.

Integrate $\underline{x}_{n/n}$ ahead to give $\underline{x}_{n+1/n}$.

Propagate $C_{n/n}$ ahead to give $C_{n+1/n}$.

Write tape of T, $\underline{x}_{n/n}$, $\underline{x}_{n+1/n}$, $C_{n/n}$, $C_{n+1/n}$, $\Phi_{n+1/n}$.

$T = T + \Delta T$.

GO to a.

b BACKSPACE TAPE.

Set $C_{N-i+1/N} = C_{n/n}$ and $\underline{x}_{N-i+1/N} = \underline{x}_{n/n}$.

$\Delta T = -\Delta T$.

c Read tape into: T, $\underline{x}_{N-i/N-i}$, $\underline{x}_{N-i+1/N-i}$, $C_{N-i/N-i}$, $C_{N-i+1/N-i}$, $\Phi_{N-i+1/N-i}$.

Go through smoothing equations to compute $\underline{x}_{N-i/N}$, $C_{N-i/N}$.

Print T, $\underline{x}_{N-i/N}$, $C_{N-i/N}$, etc.

If ($T = T_{initial}$) STOP.

BACKSPACE TAPE.

BACKSPACE TAPE.

GO TO c.

Figure 1 shows the actual scale factor, δ, and the Kalman filter solution for δ. The Kalman filter predicted standard deviation for the error in δ was about 0.0065, down from 0.035. Figure 2 shows the version 1 smoothing filter solution for the scale factor, δ. Its standard deviation was about 0.0027, a factor of 2.4 better than the Kalman filter solution. Figure 3 shows a plot of the Kalman filter velocity error versus time. Also shown on the plot is the predicted 1σ velocity error standard deviation, about 0.05 m/sec. Figure 4 shows the version 1 smoothing plot of velocity error and its predicted standard deviation of 0.016 m/sec, a factor of 3.1 improvement.

The predicted standard deviation of the position error for the Kalman filter was 0.043 meter, for the smoothing filter it was 0.020 meter - a factor of 2.2 improvement.

Version 3 of the smoothing filter equations was also run, and it showed very similar results compared to version 1. This was a little surprising because version 3 is more highly linearized than version 1, and the equations of motion are highly nonlinear, in addition to being unstable when integrated backward.

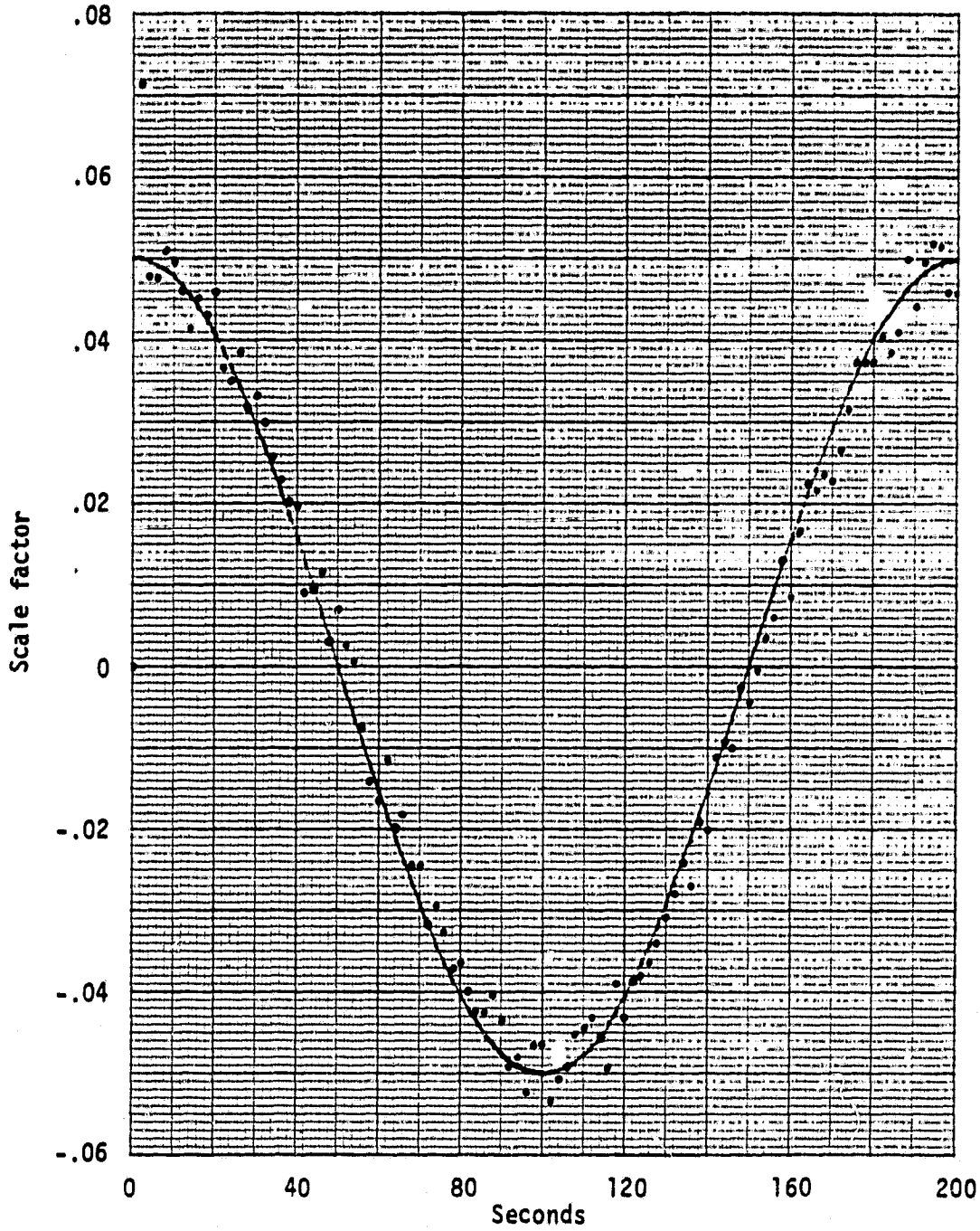

Figure 1.- Kalman filter solution for scale factor.

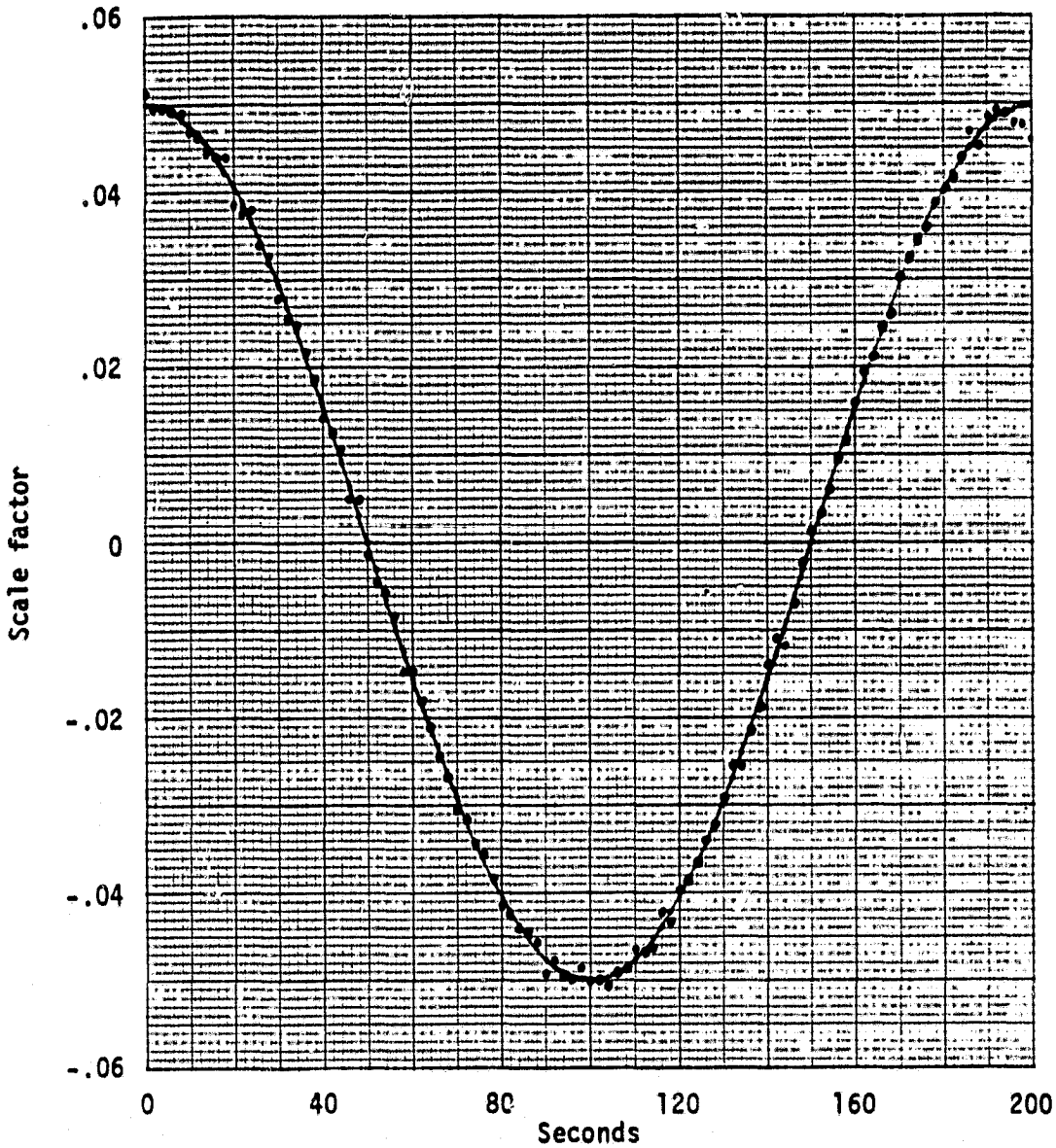

Figure 2.- Smoothing filter solution for scale factor.

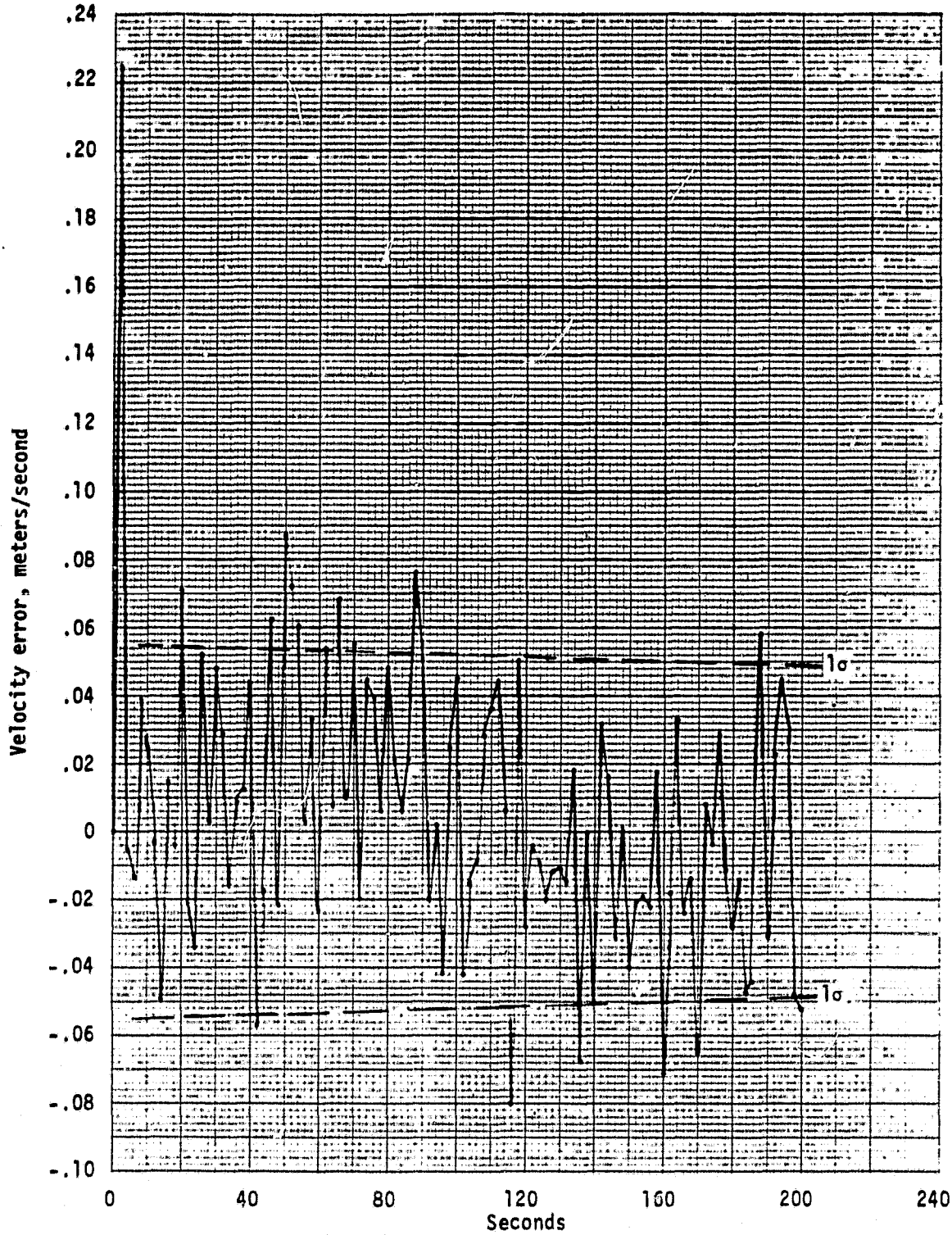

Figure 3.- Kalman filter velocity error.

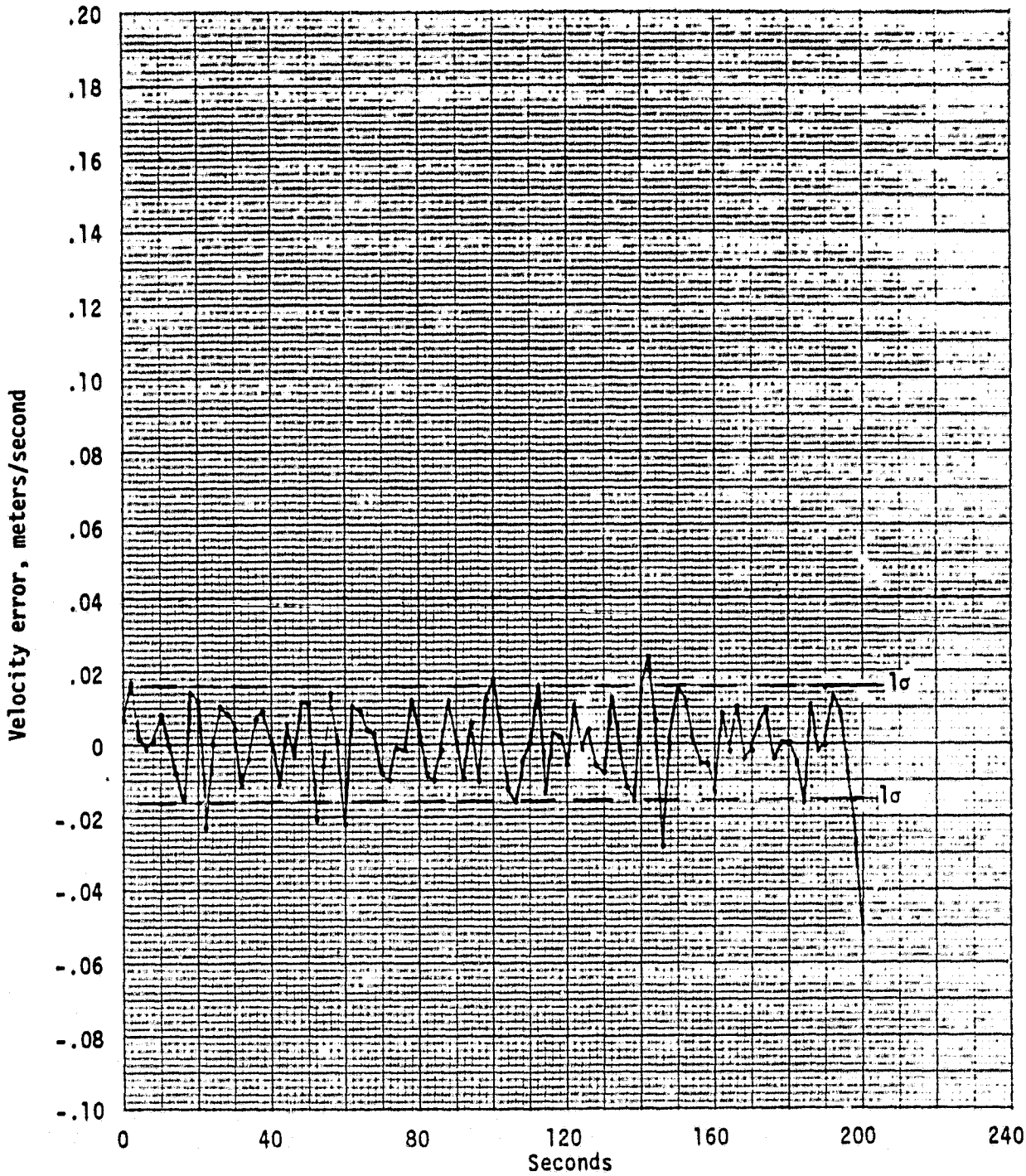

Figure 4.- Smoothing filter velocity error.

7.0 EXAMPLE 2

This example is for a highly simplified tracking version of a Space Shuttle launch. The two-dimensional tracking geometry is shown in figure 5.

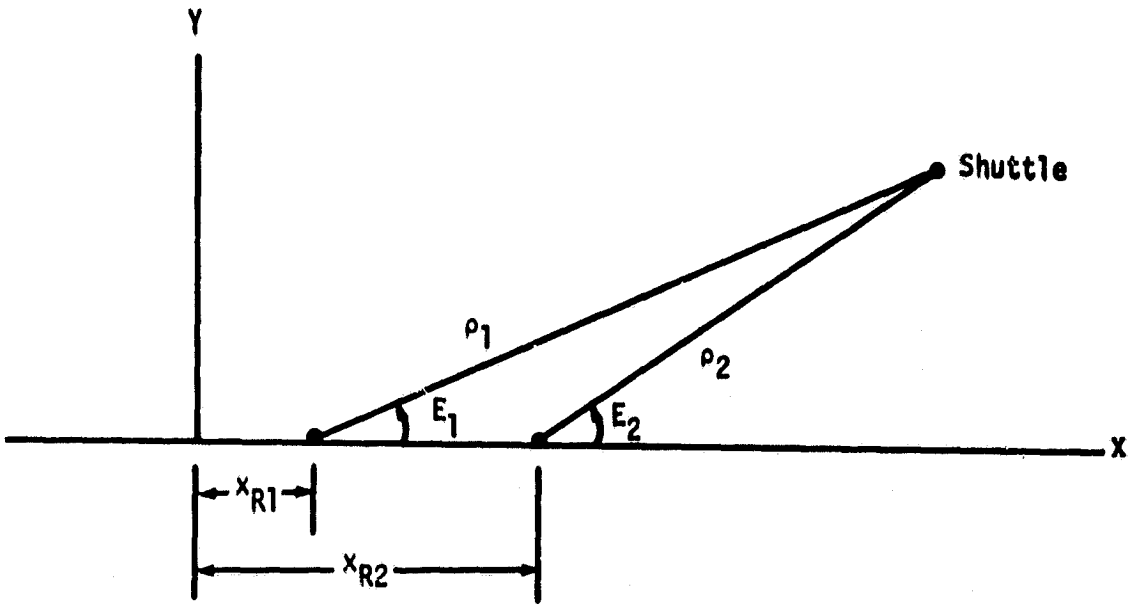

Figure 5.- Shuttle tracking geometry.

The values for the location of the radar stations will be

x_{R1} = -5000 meters

x_{R2} = 5000 meters

The actual biases adding to the radar measurements will be

$B_{A\rho 1}$ = 12 meters, constant

B_{AE1} = 0.15 milliradian, constant

$B_{A\rho 2}$ = 12 meters, constant

B_{AE2} = 0.15 milliradian, constant

The standard deviation of the actual random error adding to the measurements is

$$\sigma_{qA\rho 1} = \sigma_{qA\rho 2} = 9 \text{ meters}$$

$$\sigma_{qAE1} = \sigma_{qAE2} = 0.15 \text{ milliradian}$$

The measurement bias errors are modeled in the Kalman filter as exponentially correlated random variables with a time constant of 108 seconds. Their standard deviations are

$$\sigma_{B\rho 1} = \sigma_{B\rho 2} = 12 \text{ meters}$$

$$\sigma_{BE1} = \sigma_{BE2} = 0.15 \text{ milliradian}$$

The standard deviations of the random errors adding to the Kalman filter measurements are taken to be

$$\sigma_{q\rho 1} = \sigma_{q\rho 2} = 9 \text{ meters}$$

$$\sigma_{qE1} = \sigma_{qE2} = 0.15 \text{ milliradian}$$

They are the same as those used for the real world.

The true, real-world trajectory is generated by the following equations.

$$x_{T,i+1} = x_{Ti} + \dot{x}_{Ti}\,\Delta T + \ddot{x}_{Ti}\,\Delta T^2/2 + \dddot{x}_{Ti}\,\Delta T^3/6 \tag{7.1}$$

$$y_{T,i+1} = y_{Ti} + \dot{y}_{Ti}\,\Delta T + \ddot{y}_{Ti}\,\Delta T^2/2 + \dddot{y}_{Ti}\,\Delta T^3/6 \tag{7.2}$$

$$\dot{x}_{T,i+1} = \dot{x}_{Ti} + \ddot{x}_{Ti}\,\Delta T + \dddot{x}_{Ti}\,\Delta T^2/2 \tag{7.3}$$

$$\dot{y}_{T,i+1} = \dot{y}_{Ti} + \ddot{y}_{Ti}\,\Delta T + \dddot{y}_{Ti}\,\Delta T^2/2 \tag{7.4}$$

$$\ddot{x}_{T,i+1} = \ddot{x}_{Ti} + \dddot{x}_{Ti}\,\Delta T \tag{7.5}$$

$$\ddot{y}_{T,i+1} = \ddot{y}_{Ti} + \dddot{y}_{Ti}\,\Delta T \tag{7.6}$$

$$\dddot{x}_{T,i+1} = \dddot{x}_{Ti} = 0.1 \text{ m/sec}^3, \text{ constant} \tag{7.7}$$

$$\dddot{y}_{T,i+1} = \dddot{y}_{Ti} = 0.01 \text{ m/sec}^3, \text{ constant} \tag{7.8}$$

Initial values are, at T = 0,

x_T = -1.E6 meters

$\dot{x}_T = 0$

$\ddot{x}_T = 0$

$\dddot{x}_T = 0.1$ m/sec^3, constant

y_T = 100 000 meters

$\dot{y}_T = 0$

$\ddot{y}_T = 0$

$\dddot{y}_T = 0.01$ m/sec^3, constant

ΔT, the measurement sample interval, is

$\Delta T = 0.1$ second

At T = 300 seconds, $\ddot{x}_T = 30$ m/sec^2 = 3g, $\ddot{y}_T = 3$ m/sec^2. At this time $\ddot{x}_T$ and $\ddot{y}_T$ are set to zero because the maximum Shuttle acceleration is about 3 g's at staging.

At T = 300 seconds, the true state is

x_T = -.55E6 meter

$\dot{x}_T = 4500$ m/sec

$\ddot{x}_T = 30$ m/sec^2, reset to 0

y_T = 145 000 meters

$\dot{y}_T = 450$ m/sec

$\ddot{y}_T = 3$ m/sec^2, reset to 0

Tracking is terminated at $T = 600$ seconds. At this time,

$$x_T = 1.25E6 \text{ meters} = 675 \text{ n. mi.}$$

$$\dot{x}_T = 9000 \text{ m/sec}$$

$$\ddot{x}_T = 30 \text{ m/sec}^2$$

$$y_T = 325\ 000 \text{ meters} = 175.5 \text{ n. mi.}$$

$$\dot{y}_T = 900 \text{ m/sec}$$

$$\ddot{y}_T = 3 \text{ m/sec}^2$$

The state vector dynamics for the Kalman filter are given by

$$x_{i+1} = x_i + \dot{x}_i \Delta T + \ddot{x}_i \Delta T^2/2 \tag{7.9}$$

$$y_{i+1} = y_i + \dot{y}_i \Delta T + \ddot{y}_i \Delta T^2/2 \tag{7.10}$$

$$\dot{x}_{i+1} = \dot{x}_i + x_i \Delta T \tag{7.11}$$

$$\dot{y}_{i+1} = \dot{y}_i + y_i \Delta T \tag{7.12}$$

$$\ddot{x}_{i+1} = a_{ax}\ddot{x}_i + \sigma_{ax}\sqrt{1 - a_{ax}^2}\ \eta_{ax,i} \tag{7.13}$$

$$\ddot{y}_{i+1} = a_{ay}\ddot{y}_i + \sigma_{ay}\sqrt{1 - a_{ay}^2}\ \eta_{ay,i} \tag{7.14}$$

$$B\rho_{1,i+1} = a_{B\rho 1}\, B\rho_{1,i} + \sigma_{B\rho 1}\sqrt{1 - a_{B\rho 1}^2}\ \eta_{B\rho 1,i} \tag{7.15}$$

$$B_{E1,i+1} = a_{BE1}\, B_{E1,i} + \sigma_{BE1}\sqrt{1 - a_{BE1}^2}\ \eta_{BE1,i} \tag{7.16}$$

$$B_{\rho 2,i+1} = a_{B\rho 2}\, B_{\rho 2,i} + \sigma_{B\rho 2}\sqrt{1 - a_{B\rho 2}^{2}}\;\eta_{B\rho 2,i} \tag{7.17}$$

$$B_{E2,i+1} = a_{BE2}\, B_{E2,i} + \sigma_{BE2}\sqrt{1 - a_{BEe}^{2}}\;\eta_{BE2,i} \tag{7.18}$$

where the η_i are zero-mean, unit-variance, timewise-uncorrelated random variables. The terms containing the η_i are the Kalman filter state noise terms. Note that in integrating the state ahead, the best estimate of η_i is zero. The "a" values above are given by

$$a_{ax} = \exp(-\Delta T/\tau_{ax}) \qquad \tau_{ax} = 40 \text{ seconds} \tag{7.19}$$

$$a_{ay} = \exp(-\Delta T/\tau_{ay}) \qquad \tau_{ay} = 40 \text{ seconds} \tag{7.20}$$

$$a_{B\rho 1} = \exp(-\Delta T/\tau_{B\rho 1}) \qquad \tau_{B\rho 1} = 108 \text{ seconds} \tag{7.21}$$

$$a_{BE1} = \exp(-\Delta T/\tau_{BE1}) \qquad \tau_{BE1} = 108 \text{ seconds} \tag{7.22}$$

$$a_{B\rho 2} = \exp(-\Delta T/\tau_{B\rho 2}) \qquad \tau_{B\rho 2} = 108 \text{ seconds} \tag{7.23}$$

$$a_{BE2} = \exp(-\Delta T/\tau_{BE2}) \qquad \tau_{BE2} = 108 \text{ seconds} \tag{7.24}$$

It is seen that the dynamical equations are linear. That is, the state vector, $\underline{x}_i$, is propagated ahead by

$$\underline{x}_{i+1} = \Phi_{i+1/i}\, \underline{x}_i \tag{7.25}$$

where the state transition matrix $\Phi_{i+1/i}$ is given on the next page and its inverse on the page after that. Note the large number of zero elements in $\Phi_{i+1/i}$ and $\Phi_{i+1/i}^{-1}$. This is typical and may be used to speed the matrix arithmetic.

$$\Phi_{i+1/1} = \begin{bmatrix} 1 & 0 & \Delta T & 0 & \ell T^2/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \Delta T & 0 & \Delta T^2/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \Delta T & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta T & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{ax} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{ax} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{B\rho 1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_{BE1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_{B\rho 2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_{BE2} \end{bmatrix}$$

$$\Phi^{-1}_{i+1/1} = \begin{bmatrix} 1 & 0 & -\Delta T & 0 & \frac{1}{a_{ax}}\frac{\Delta T^2}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -\Delta T & 0 & \frac{1}{a_{ay}}\frac{\Delta T^2}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -\Delta T/a_{ax} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -\Delta T/a_{ay} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/a_{ax} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/a_{ay} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/a_{B\rho 1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/a_{BE1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/a_{B\rho 2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/a_{BE2} \end{bmatrix}$$

The range measurement from station 1 is given by

$$\rho_1 = \sqrt{(x - x_{R1})^2 + y^2} + B_{\rho 1} \tag{7.26}$$

And the elevation angle measurement from station 1 is

$$E_1 = \arctan \frac{y}{x - x_{R1}} + B_{E1} \tag{7.27}$$

The partial derivatives are

$$\frac{\partial \rho_1}{\partial x} = \frac{x - x_{R1}}{\sqrt{(x - x_{R1})^2 + y^2}} \tag{7.28}$$

$$\frac{\partial \rho_1}{\partial y} = \frac{y}{\sqrt{(x - x_{R1})^2 + y^2}} \tag{7.29}$$

$$\frac{\partial \rho_1}{\partial B_{\rho 1}} = 1 \tag{7.30}$$

$$\frac{\partial E_1}{\partial x} = - \frac{y}{(x - x_{R1})^2 + y^2} \tag{7.31}$$

$$\frac{\partial E_1}{\partial y} = \frac{x - x_{R1}}{(x - x_{R1})^2 + y^2} \tag{7.32}$$

$$\frac{\partial E_1}{\partial B_{E1}} = 1 \tag{7.33}$$

Similar equations apply to ρ_2 and E_2.

And finally, the state noise covariance matrix is given on the next page.

$$S_i = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{ax}^2(1-a_{ax}^2) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{ay}^2(1-a_{ay}^2) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{B\rho 1}^2(1-a_{B\rho 1}^2) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{BE1}^2(1-a_{BE1}^2) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{B\rho 2}^2(1-a_{B\rho 2}^2) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{BE2}^2(1-a_{BE2}^2) \end{bmatrix}$$

Figure 6 shows the magnitude of the velocity error vector versus time for the Kalman (forward) filter. Figure 7 shows the same thing for the smoothing filter. Note that staging occurred at 300 seconds. The smoothing filter results are much better (about 3 or 4 times better) than the Kalman filter results. Figures 8 and 9 show the velocity errors across staging in finer detail. Again it is seen that the smoothing filter results are superior to those of the Kalman filter. The smoothing filter has a much lower maximum error and a shorter transient time across staging. Also note how smooth the curve is for the smoothing filter in figure 9. The smothing filter is aptly named because its output appears to be very "smooth" compared to the Kalman filter.

Figures 10 and 11 show the Kalman filter and smoothing filter solutions for the x component of acceleration across staging. x is seen to go from 3g to zero at staging. Again it is seen that the smoothing filter solution is very smooth and improved over that of the Kalman filter.

The smoothing filter did not improve the position errors by much, maybe 5 percent. Also the radar bias errors in this example were not well determined by either the Kalman filter or the smoothing filter.

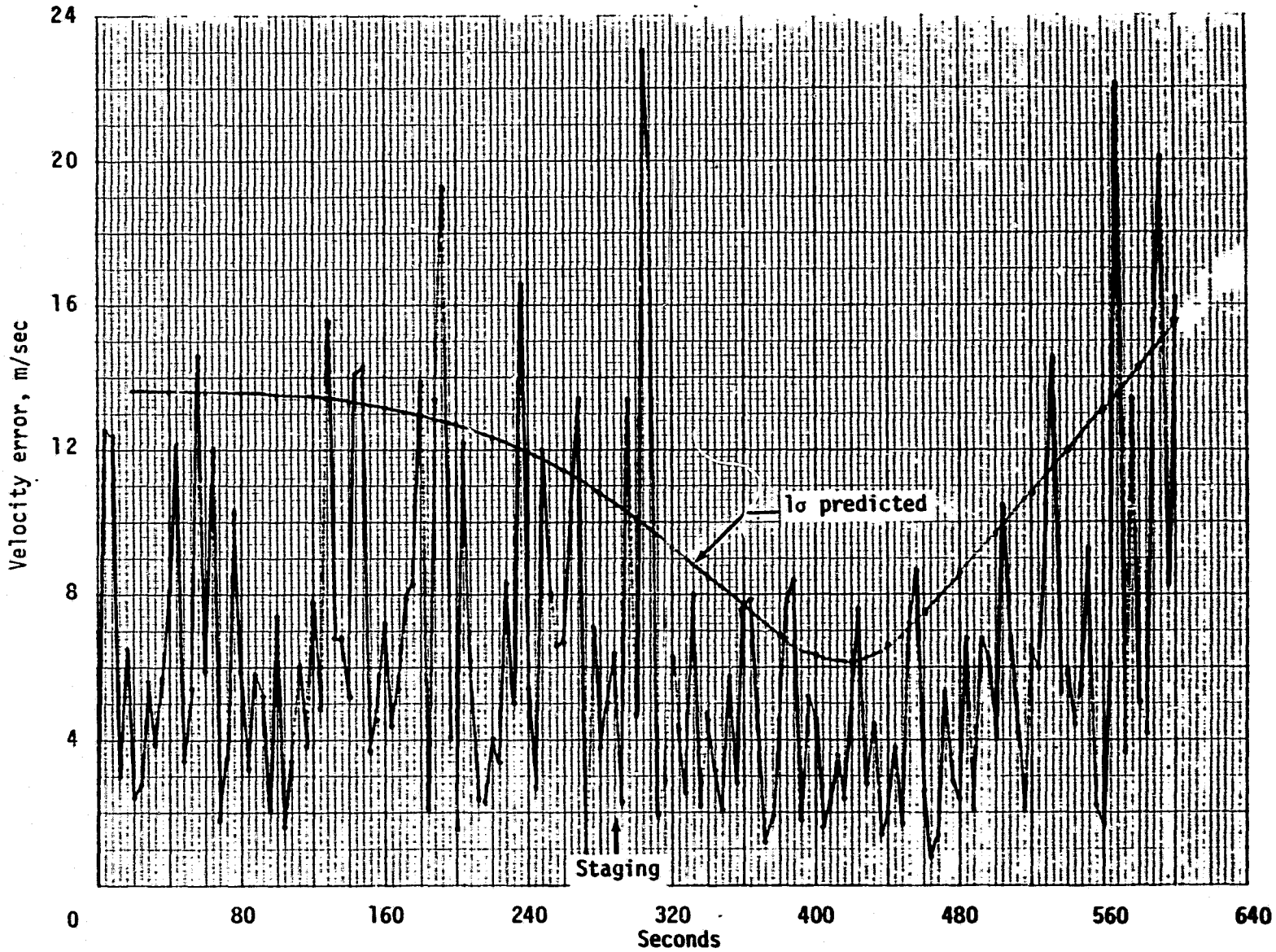

Figure 6.- Magnitude of velocity error vector for Kalman filter.

ORIGINAL PAGE IS OF POOR QUALITY

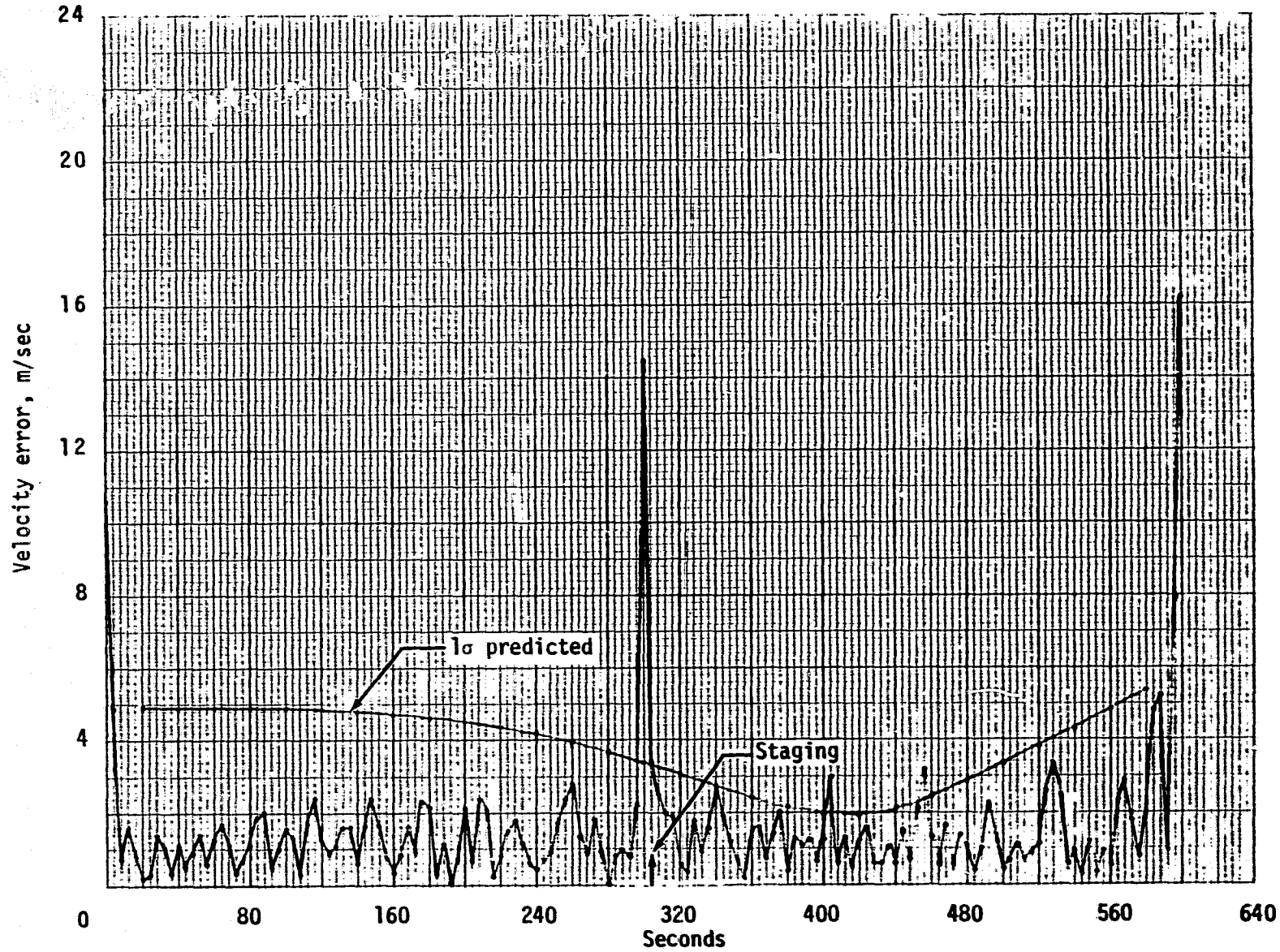

Figure 7.- Magnitude of velocity error vector for smoothing filter.

ORIGINAL PAGE IS
OF POOR QUALITY

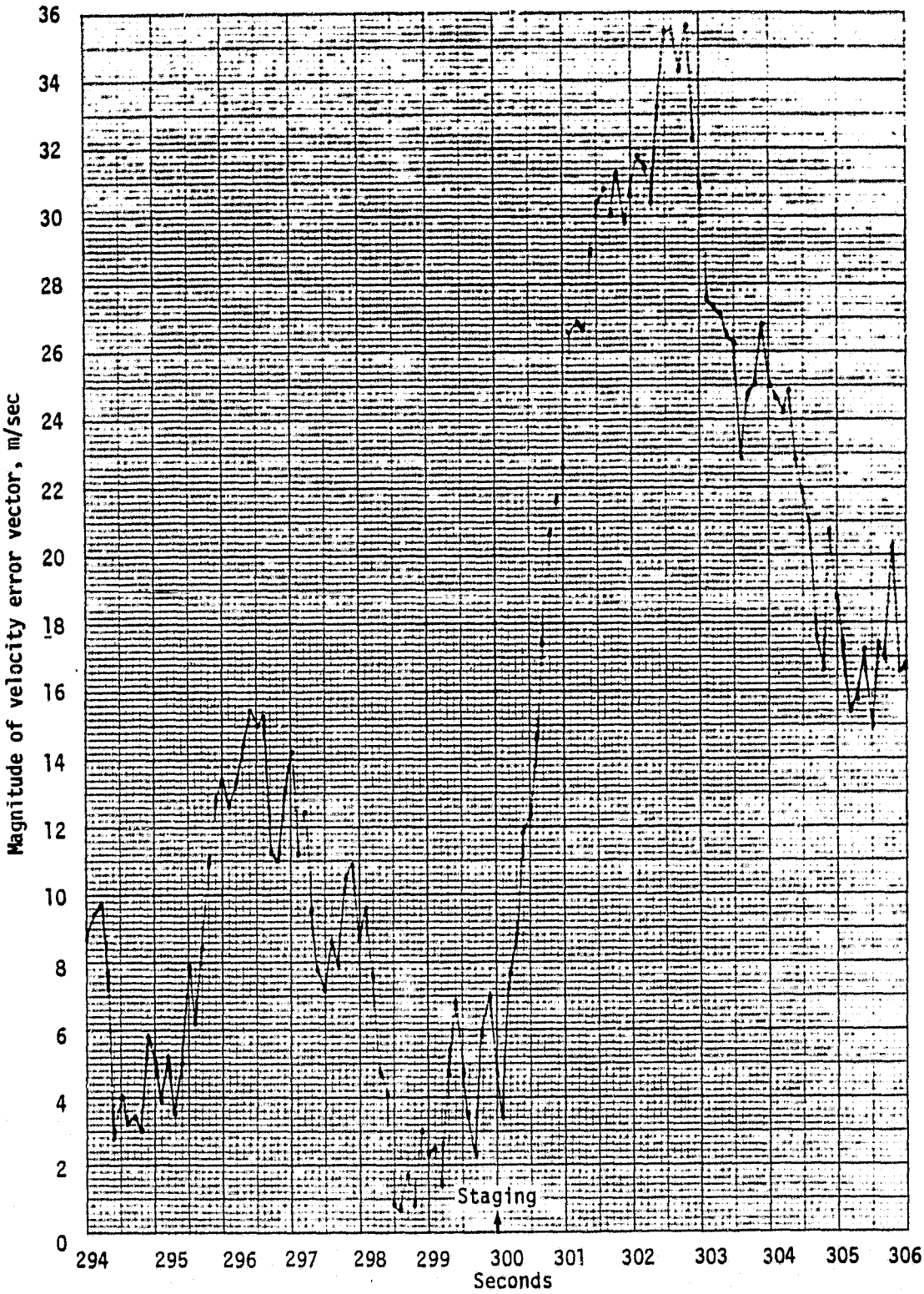

Figure 8.- Velocity error across staging for Kalman filter.

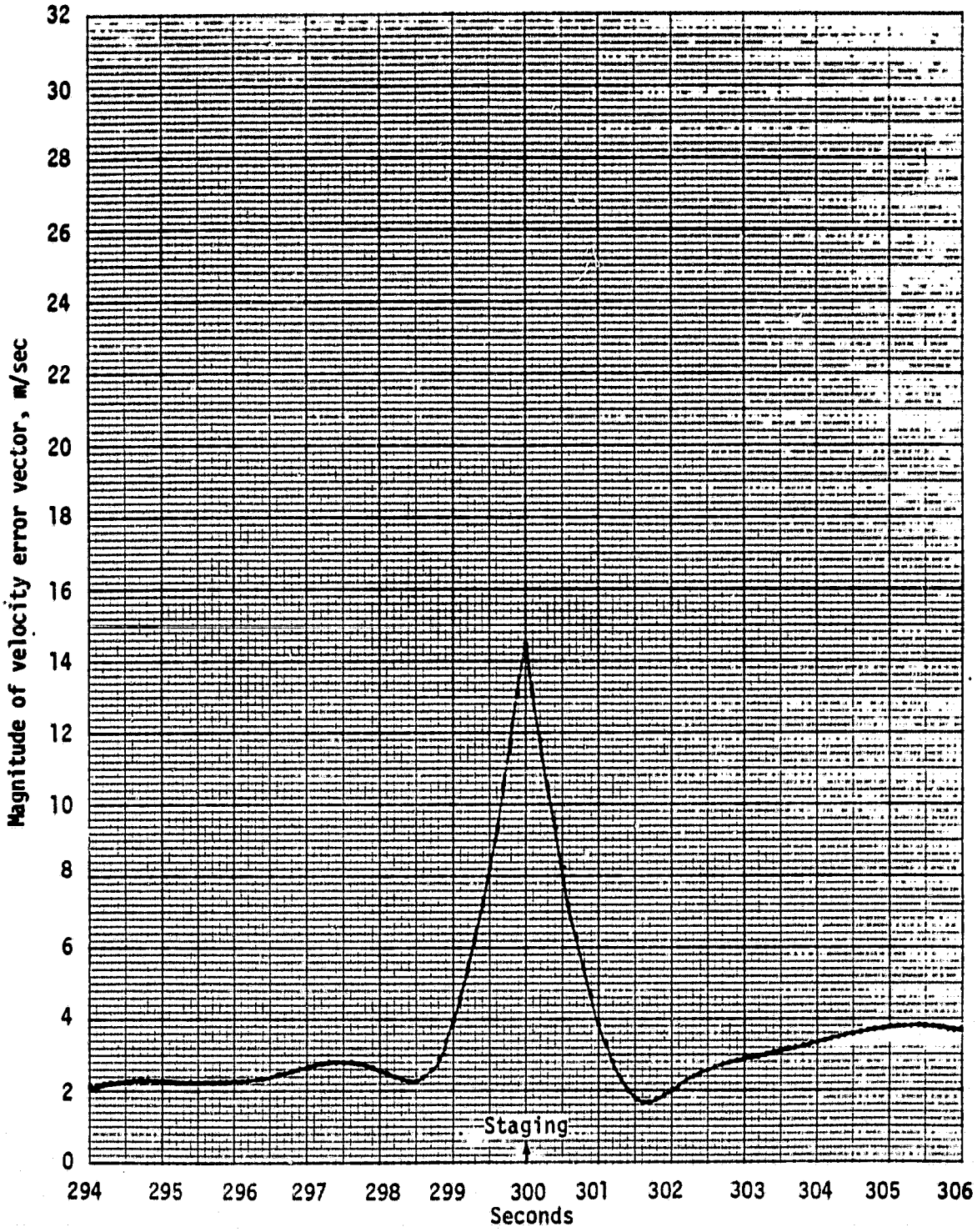

Figure 9.- Velocity error across staging for smoothing filter.

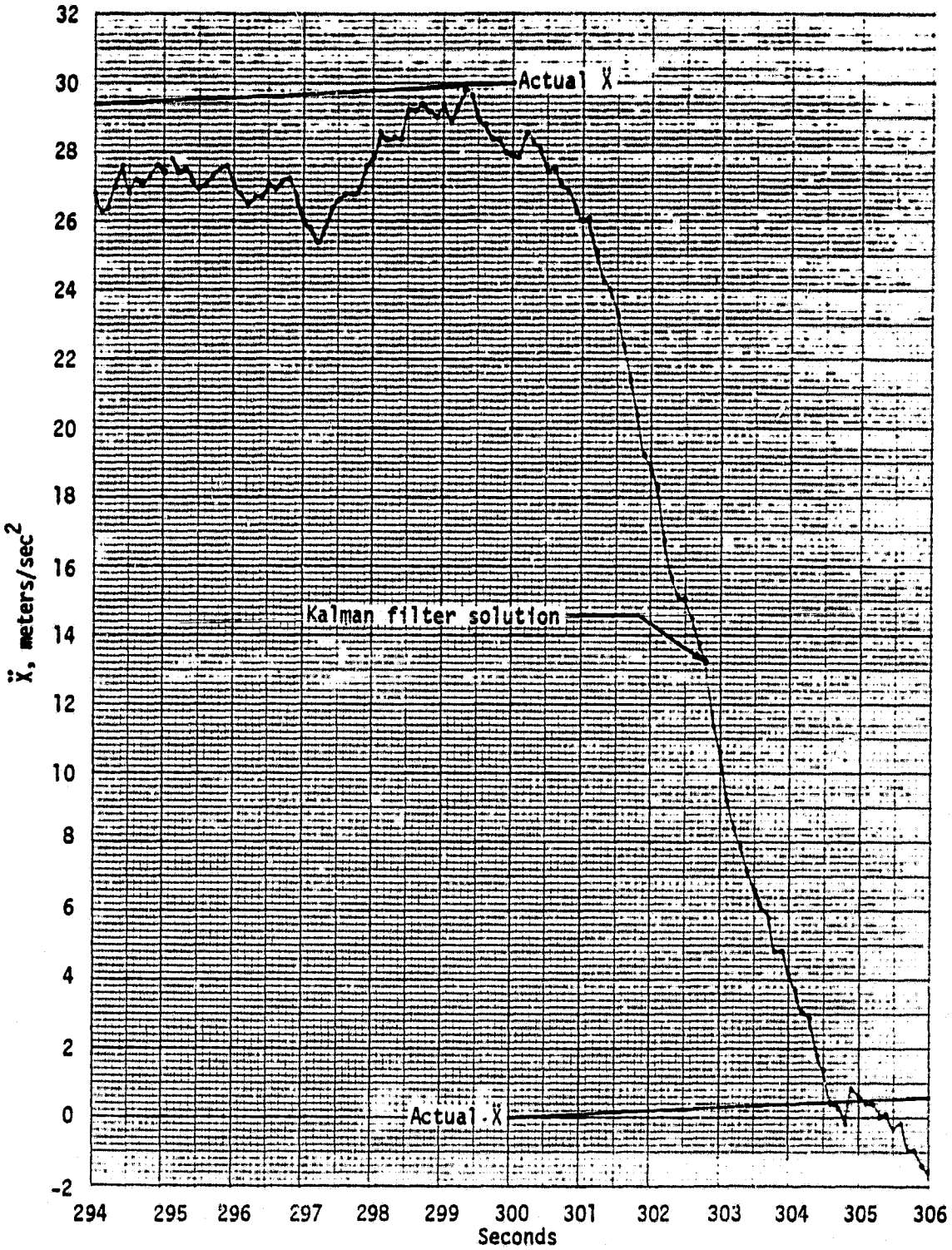

Figure 10.- Kalman filter solution for $\ddot{X}$ across staging.

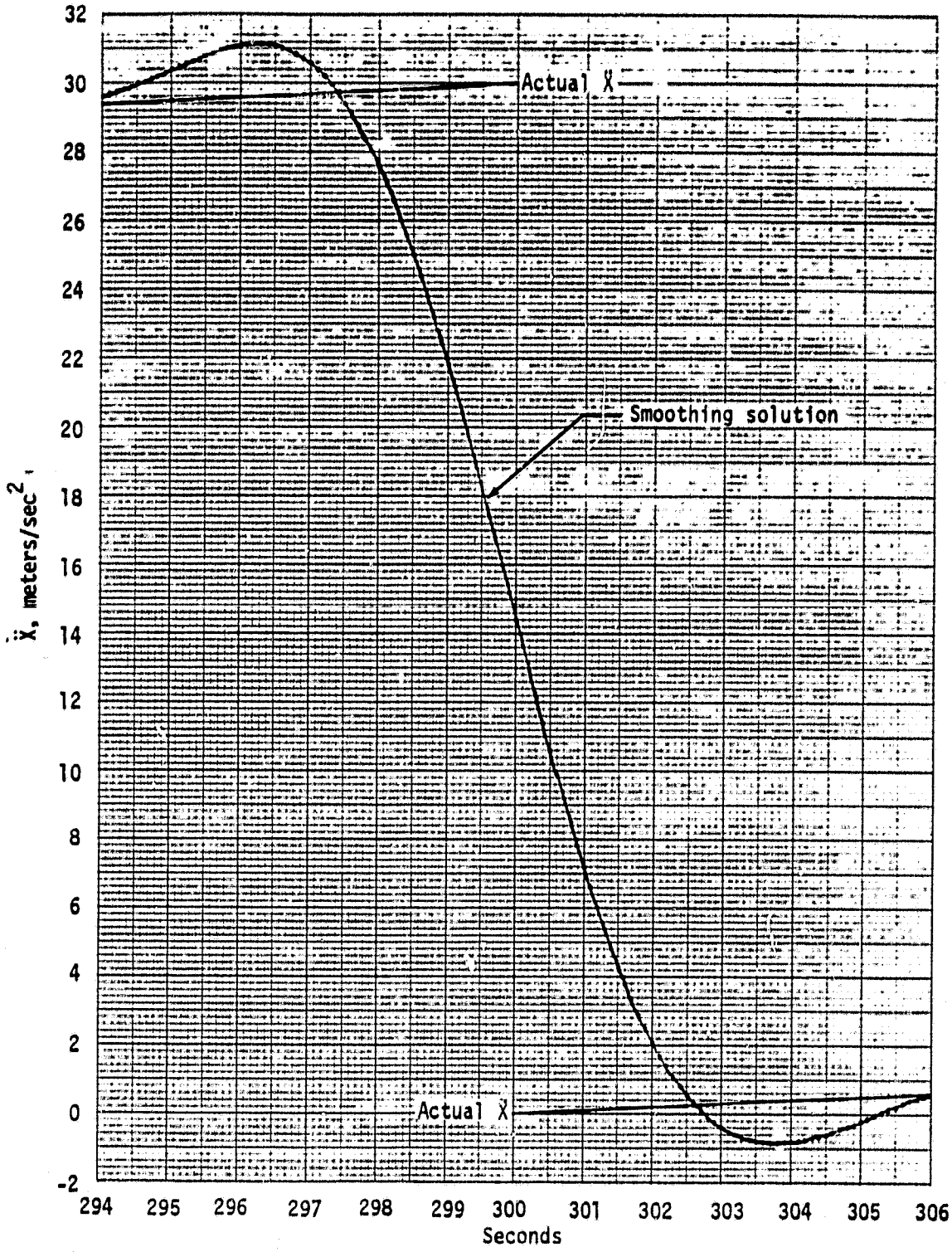

Figure 11.- Smoothing filter solution for $\ddot{X}$ across staging.

ORIGINAL PAGE IS
OF POOR QUALITY

8.0 MATRIX INVERSION ALGORITHMS

The smoothing equations require efficient matrix inversion algorithms. Four algorithms are presented here.[a] These algorithms are very fast and require a minimum of storage. The matrix to be inverted is H, an N by N matrix. If $H = H^T$ the inversion requires about $N^3/2$ additions and multiplications. If $H \neq H^T$ then the inversion requires about N^3 multiplications and additions. Note that multiplication of two N by N matrices requires about N^3 multiplications and additions. Note that the algorithms were written for clarity and may be speeded up even more by revised programing.

[a]These algorithms were obtained from William M. Lear, "Kalman Filtering Techniques", NASA/JSC Internal Note 78-FM-25, April 1978.

8.1 FORTRAN ALGORITHM FOR $H = H^{-1}$ WHERE $H = H^T$

```
      DIMENSION H(N,N)
      H(1,1) = 1./H(1,1)
      IF (N. EQ. 1) RETURN
      NM1 = N-1
      Dϕ 5 J=1, NM1
      Dϕ 1 I=1, J
      H(I,J+1) = 0.
      Dϕ 1 K=1, J
1     H(I,J+1) = H(I,J+1) - H(J+1,K)*H(I,K)
      Dϕ 2 K=1, J
2     H(J+1,J+1) = H(J+1,J+1) + H(J+1,K)*H(K,J+1)
      H(J+1,J+1) = 1./H(J+1,J+1)
      Dϕ 3 I=1, J
3     H(J+1,I) = H(I,J+1)*H(J+1,J+1)
      Dϕ 4 I=1, J
      Dϕ 4 K=I, J
      H(I,K) = H(I,K) + H(I,J+1)*H(J+1,K)
4     H(K,I) = H(I,K)
      Dϕ 5 I=1, J
5     H(I,J+1) = H(J+1,I)
      RETURN
      END
```

8.2 FORTRAN ALGORITHM FOR $G = H^{-1}$ WHERE $H = H^T$

```
      DIMENSION G(N,N), H(N,N)
      G(1,1) = 1./H(1,1)
      IF (N. EQ. 1) RETURN
      NM1 = N-1
      DΦ 5 J=1, NM1
      DΦ 1 I=1, J
      G(I,J+1) = 0.
      DΦ 1 K=1, J
1     G(I,J+1) = G(I,J+1) - H(J+1,K)*G(I,K)
      G(J+1,J+1) = H(J+1,J+1)
      DΦ 2 K=1, J
2     G(J+1,J+1) = G(J+1,J+1) + H(J+1,K)*G(K,J+1)
      G(J+1,J+1) = 1./G(J+1,J+1)
      DΦ 3 I=1, J
3     G(J+1,I) = G(I,J+1)*G(J+1,J+1)
      DΦ 4 I=1, J
      DΦ 4 K=I, J
      G(I,K) = G(I,K) + G(I,J+1)*G(J+1,K)
4     G(K,I) = G(I,K)
      DΦ 5 I=1, J
5     G(I,J+1) = G(J+1,I)
      RETURN
      END
```

8.3 FORTRAN ALGORITHM FOR $H = H^{-1}$ WHERE $H \neq H^T$

```
      DIMENSION F(N-1), H(N,N)
      H(1,1) = 1./H(1,1)
      IF (N. EQ. 1) RETURN
      NM1 = N-1
      Dϕ 5 J=1, NM1
      Dϕ 1 I=1, J
      F(I) = 0.
      Dϕ 1 K=1, J
1     F(I) = F(I) - H(I,K)*H(K,J+1)
      Dϕ 2 K=1, J
2     H(J+1,J+1) = H(J+1,J+1) + H(J+1,K)*F(K)
      H(J+1,J+1) = 1./H(J+1,J+1)
      Dϕ 3  I=1, J
      H(I,J+1) = F(I)*H(J+1,J+1)
      F(I) = 0.
      Dϕ 3 K=1, J
3     F(I) = F(I) - H(J+1,K)*H(K,I)
      Dϕ 4 I=1, J
      Dϕ 4 K=1, J
4     H(I,K) = H(I,K) + H(I,J+1)*F(K)
      Dϕ 5 I=1, J
5     H(J+1,I) = H(J+1,J+1)*F(I)
      RETURN
      END
```

8.4 FORTRAN ALGORITHM FOR $G = H^{-1}$ WHERE $H \neq H^T$

```
      DIMENSION G(N,N), H(N,N)
      G(1,1) = 1./H(1,1)
      IF (N. EQ. 1) RETURN
      NM1 = N-1
      Dϕ 5 J=1, NM1
      Dϕ 1 I=1, J
      G(I,N) = 0.
      Dϕ 1 K=1, J
1     G(I,N) = G(I,N) - G(I,K)*H(K,J+1)
      G(J+1,J+1) = H(J+1,J+1)
      Dϕ 2 K=1, J
2     G(J+1,J+1) = G(J+1,J+1) + H(J+1,K)*G(K,N)
      G(J+1,J+1) = 1./G(J+1,J+1)
      Dϕ 3 I=1, J
      G(I,J+1) = G(J+1,J+1)*G(I,N)
      G(N,I) = 0.
      Dϕ 3 K=1, J
3     G(N,I) = G(N,I) - H(J+1,K)*G(K,I)
      Dϕ 4 I=1, J
      Dϕ 4 K=1, J
4     G(I,K) = G(I,K) + G(I,J+1)*G(N,K)
      Dϕ 5 I=1, J
5     G(J+1,I) = G(J+1,J+1)*G(N,I)
```

9.0 CONCLUDING REMARKS

It has been seen that the smoothing filter can considerably improve accuracy over that of a Kalman filter. Factors of improvement for velocity errors of up to three or four times have been seen here when position measurements were being processed.

Three versions of the smoothing equations have been presented. Of the three, version 1 is theoretically the most accurate but requires the most computer time. If the equations of motion are linear, that is, if

$$\underline{x}_{i+1} = \Phi_{i+1/i}\, \underline{x}_i$$

then all three versions should give identical results except for computer roundoff error.

The state estimation equations for the smoothing filter appear to be quite stable. In example 1, the backward integration of the equations of motion was unstable but, in the presence of the smoothing equations, became stable. Example 2 showed that the smoothing filter output was indeed quite smooth when compared to the noisy Kalman filter estimates.

☆ U.S. GOVERNMENT PRINTING OFFICE: 1980—671-099/1849