NRL Report 8212

A Simple Means of Updating the SRIF Filter When the State Equations are in Triangular Form

B. H. CANTRELL

Radar Analysis Staff
Radar Division

March 29, 1978

NAVAL RESEARCH LABORATORY
Washington, D.C.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NRL Report 8212	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A SIMPLE MEANS OF UPDATING THE SRIF FILTER WHEN THE STATE EQUATIONS ARE IN TRIANGULAR FORM		5. TYPE OF REPORT & PERIOD COVERED Interim report on a continuing NRL problem 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Ben H. Cantrell		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Research Laboratory Washington, D.C. 20375		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NRL Problem R12-18 Project SF12-133-401 Program Element 62712N
11. CONTROLLING OFFICE NAME AND ADDRESS Department of the Navy Naval Sea Systems Command Washington, D.C. 20362		12. REPORT DATE March 29, 1978 13. NUMBER OF PAGES 14
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)

Kalman filters
Tracking
Least-square filters
SRIF filters

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

The square-root information filter (SRIF), which is a numerical method of implementing the Kalman filter, was investigated. Under an important special case of no process noise and an upper triangular form of the state transition matrix the prediction process of the filter is of simple form. The transformed smoothed and predicted states are identical, and the smoothed and predicted covariance factors are related by a simple matrix transform.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

CONTENTS

A SIMPLE MEANS OF UPDATING THE SRIF FILTER WHEN THE STATE EQUATIONS ARE IN TRIANGULAR FORM

INTRODUCTION

Estimating the state of a system from a set of uncertain measurements has been a problem for a long time. Kalman in the early sixties provided a simple recursive estimation procedure by introducing the concept of state and state transition. This procedure in some instances provided simpler implementation than batching techniques. Since Kalman's work a number of numerical procedures have been developed. An excellent account of these procedures as well as historical notes can be found in Bierman's book [1]. The square-root information filter (SRIF filter) is the numerical method of solving the Kalman-filter equations, which is of interest in this report.

There are a number of problems which involve a state transition matrix which is in upper triangular form. Prominent examples of problems involving the condition are most tracking problems. This report describes a simple means of updating the prediction process of the filter under this condition. A secondary but important result is that the SRIF filter lends itself to parallel hardware implementation.

REVIEW OF THE SRIF FILTER

The SRIF filter is a numerical method of implementing the Kalman filter [1]. The Kalman filter is obtained from modeling the process as state equations, defining a measurement procedure, and best estimating the states of the systems. The state equation and measurement process are defined as

$$X(k) = \Phi(k)X(k-1) + \Gamma(k)W(k) \quad .$$

and

$$X_M(k) = H(k)X(k) + V(k),$$

where it is desired to best estimate the n-by-1 state vector $X(k)$. The remaining quantities are an n-by-n state transition matrix $\Phi(k)$, an n-by-p matrix $\Gamma(k)$, an m-by-n measurement matrix $H(k)$, and an m-by-1 measurement vector $X_M(k)$. $W(k)$ and $V(k)$ are independent Gaussian noises with the properties

$$E[W(k)] = 0,$$
$$E[W(k)W'(j)] = S(k)\delta_{jk},$$
$$E[V(k)] = 0,$$

*Manuscript submitted February 16, 1978.

$$E[V(k)V'(j)] = Q(k)\delta_{jk},$$

and

$$E[W(k)V'(j)] = 0,$$

where δ_{jk} is 1 when $j = k$ and is 0 otherwise. The covariance matrices $S(k)$ and $Q(k)$ are of dimension p by p and m by m respectively.

The best estimate of $X(k)$, denoted by $\tilde{X}(k)$ in the standard Kalman-filter format, is

$$\tilde{X}(k) = \hat{X}(k) + K(k)[X_m(k) - H(k)\hat{X}(k)], \tag{1}$$

where $K(k)$ is the filter gain, given by

$$K(k) = \tilde{P}(k)H'(k)Q^{-1}(k), \tag{2}$$

in which $\tilde{P}(k)$ is the smoothed covariance matrix, with

$$\tilde{P}^{-1}(k) = \hat{P}^{-1}(k) + H'(k)Q^{-1}(k)H(k). \tag{3}$$

$\hat{P}(k)$ is the predicted covariance matrix, with

$$\hat{P}(k+1) = \Phi(k+1)\tilde{P}(k)\Phi'(k+1) + \Gamma(k+1)S(k+1)\Gamma'(k+1), \tag{4}$$

and $\hat{X}(k+1)$ is the prediction:

$$\hat{X}(k+1) = \Phi(k+1)\tilde{X}(k), \tag{5}$$

The filter operates in a predict-and-correct fashion. This suggests a simple derivation, outlined below.

Equation (1) is the least-square estimate between the prediction and the measurement at the kth sample which is obtained by minimizing the cost function

$$J(k) = [\hat{X}(k) - X(k)]'\hat{P}^{-1}(k)[\hat{X}(k) - X(k)] + [X_M(k) - HX(k)]'Q^{-1}(k)[X_M(k) - HX(k)] \tag{6}$$

with respect to $X(k)$. The value of $X(k)$ which minimizes $J(k)$ is denoted by $\tilde{X}(k)$, is the best estimate of $X(k)$, and is given in equations (1) through (3). Given the best estimate of $X(k)$, the best prediction is simply equation (5) with the covariance of (4). The process is then simply repeated recursively, with equations (4) and (5) being the prediction and equations (1) through (3) being the correction.

The SRIF filter is a means of implementing the Kalman filter which depends heavily on Cholesky decomposition and the Householder matrix triangulation algorithm [1]. The Cholesky decomposition is performed on a symmetric positive-definite matrix by factoring it into the product of a lower triangular matrix L and its transpose:

$$Q = L\,L'$$

and

$$Q^{-1} = (L')^{-1}L^{-1}.$$

The algorithm for obtaining L, found in reference 1, is

$$\ell_{jj} = \sqrt{q_{jj}} \qquad \text{for } j = 1, \ldots, n-1,$$

$$\ell_{kj} = q_{kj}/\ell_{jj} \qquad \text{for } k = j+1, \ldots, n,$$

and

$$q_{ik} = q_{ik} - \ell_{ij}\ell_{kj} \qquad \text{for } k = j+1, \ldots, n \text{ and } i = k, \ldots, n.$$

The cost function in equation (6) can be written as

$$J = (\hat{X} - X)'\hat{R}\hat{R}'(\hat{X} - X) + (X_M - HX)'(L')^{-1}L^{-1}(X_M - HX), \tag{7}$$

where the parenthetical k has been dropped for notational covenience, $\hat{P}^{-1}$ is factored into $\hat{R}\hat{R}'$, and $Q(k)$ is factored into $L\,L'$ (note that $Q^{-1}(k) = (L')^{-1}L^{-1}$). Equation (7) can be rewritten as

$$J = (\hat{Z} - \hat{R}'X)'(\hat{Z} - \hat{R}'X) + (Z_M - H_W X)'(Z_M - H_W X),$$

where

$$\hat{Z} = \hat{R}'X,$$

$$Z_M = L^{-1}X_M,$$

and

$$H_W = L^{-1}H.$$

Equation (7) can then be rewritten more compactly as

$$J = \left[\begin{bmatrix} \hat{R}' \\ H_W \end{bmatrix} X - \begin{bmatrix} \hat{Z} \\ Z_M \end{bmatrix}\right]' \left[\begin{bmatrix} \hat{R}' \\ H_W \end{bmatrix} X - \begin{bmatrix} \hat{Z} \\ Z_M \end{bmatrix}\right]. \tag{8}$$

The cost J is unaltered if an orthogonal transform T, where $T'T = I$, is multiplied by the new resulting vector in (8). Consequently using

$$\begin{bmatrix} \hat{R}' \\ H_W \end{bmatrix} X - \begin{bmatrix} \hat{Z} \\ Z_M \end{bmatrix} = C$$

in $J = C'\,C$ yields the same cost J as

$$J = C'T'T\,C.$$

In addition, if T, which is a $n + m$ square matrix, is chosen such that

$$T \begin{bmatrix} \hat{R}' \\ H_W \end{bmatrix} = \begin{bmatrix} \tilde{R}' \\ 0 \end{bmatrix} \tag{9}$$

and

$$T \begin{bmatrix} \hat{Z} \\ Z_M \end{bmatrix} = \begin{bmatrix} \tilde{Z} \\ e \end{bmatrix}, \tag{10}$$

then the cost J becomes

$$J = (\tilde{R}'X - \tilde{Z})'(\tilde{R}'X - \tilde{Z}) + e'e.$$

By inspection the least-square estimate of X is

$$\tilde{R}'\tilde{X} = \tilde{Z} \text{ or } \tilde{X} = \tilde{R}^{-1}\tilde{Z},$$

$e'e$ is the minimum value of the cost J, and the smoothed covariance is $\tilde{P}(k) = (\tilde{R}')^{-1}\tilde{R}^{-1}$. For simplification (10) is argumented to (9), yielding

$$T \begin{bmatrix} \hat{R}' & \hat{Z} \\ H_W & Z_M \end{bmatrix} = \begin{bmatrix} \tilde{R}' & \tilde{Z} \\ 0 & e \end{bmatrix}. \tag{11}$$

The transform T triangulizes the matrix.

To show (11) is equivalent to the smoothing portion of the Kalman filter, (11) can be written as

$$\begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \begin{bmatrix} \hat{R}' & \hat{Z} \\ H_W & Z_M \end{bmatrix} = \begin{bmatrix} \tilde{R}' & \tilde{Z} \\ 0 & e \end{bmatrix}.$$

Equating terms, one has

$$T_{11}\hat{R}' + T_{12}H_W = \tilde{R}' \tag{12}$$

and

$$T_{11}\hat{Z} + T_{12}Z_M = \tilde{Z}. \tag{13}$$

If one chooses

$$T_{11} = \tilde{R}^{-1}\hat{R} \tag{14}$$

and

$$T_{12} = \tilde{R}^{-1}H_W', \tag{15}$$

equation (12) becomes

$$\hat{R}\,\hat{R}' + H_W' H_W = \tilde{R}\,\tilde{R}'.$$

Using previous definitions for H_W and using $\hat{P}^{-1} = \hat{R}\,\hat{R}'$ and $\tilde{P}^{-1} = \tilde{R}\,\tilde{R}'$, one obtains equation (3) of the Kalman filter. Similarly, substituting (14) and (15) into (13), one obtains equation (1) of the Kalman filter.

The Householder algorithm can be used to triangulize the matrix represented in (11) without ever computing the transform T directly. Only the basic results are sketched, and an example is given. Detailed information may be found in reference 1. The algorithm is based on reflection. Let the vector U be normal to the plane $U_\perp$. An arbitrary vector Y can be represented by

$$Y = (Y'\hat{U})\hat{U} + \nu, \tag{16}$$

where $\hat{U} = U/(U \cdot U)^{1/2}$ and ν is that part of Y that is orthogonal to U. The reflection of Y denoted by Y_r in the plane $U_\perp$ is

$$Y_r = -(Y'\hat{U})\hat{U} + \nu, \tag{17}$$

and the results are represented in Fig. 1.

Eliminating ν from (16) and (17) yields

$$Y_r = Y - 2\,\frac{Y'U}{U'U}\,U = (I - \beta U\,U')Y = TY, \tag{18}$$

where

$$\beta = \frac{2}{U'U}\;.$$

The matrix T is an elementary Householder transform with properties $T' = T$ and $T\,T' = I$. Equation (18) can be shown to triangulize a matrix by first setting the elements of the vector U by

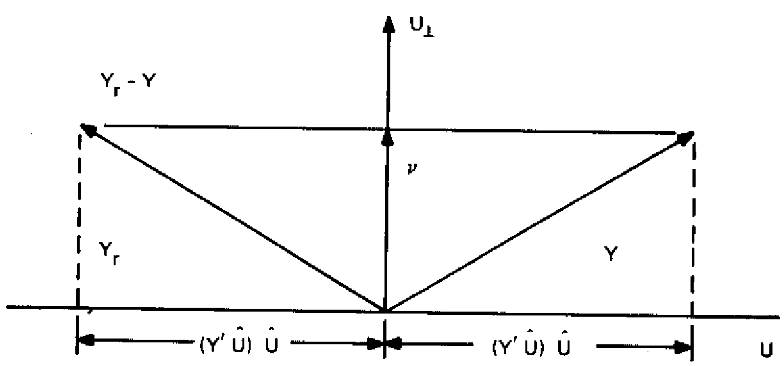

Fig. 1 — Geometry of the Householder algorithm

$$u(1) = y(1) + \sigma,$$

$$u(2) = y(2),$$

and

$$u(j) = y(j),$$

where $\sigma = \text{sgn}\, y(1)\sqrt{Y'Y}$. The transform TY yields $y_r(1) = \sigma$ and $y_r(j) = 0$ for $j = 1, 2, \ldots$. The first column of the matrix is chosen as $y(j)$ in order to set $u(j)$. Equation (18) is then applied successively. The sign on (18) is changed to yield positive diagonal elements of σ, and the notation $\beta = 2/U'U$ is introduced. The algorithm operating on successive columns of the matrix is

$$Y_r = -Y + \beta(Y'U)U.$$

For example

$$T_1 \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix} = \begin{bmatrix} \sigma_1 & b_{12} & b_{13} \\ 0 & b_{22} & b_{23} \\ 0 & b_{32} & b_{33} \\ 0 & b_{42} & b_{43} \end{bmatrix},$$

where

$$U = \begin{bmatrix} a_{11} + \sigma_1 \\ a_{21} \\ a_{31} \\ a_{41} \end{bmatrix},$$

$$\sigma_1 = \text{sgn}(a_{11})\sqrt{a_{11}^2 + a_{21}^2 + a_{31}^2 + a_{41}^2},$$

$$\beta = 2/U'U,$$

$$\gamma_{1j} = a_{1j}u(1) + a_{2j}u(2) + a_{3j}u(3) + a_{4j}u(4),$$

and

$$b_{ij} = -a_{ij} + \beta\gamma_{1j}u(i) \quad \text{for } j = 1, 2, \text{ and } 3 \text{ and } i = 1, \ldots, 4.$$

The process is repeated for each successive submatrix. The next step is

$$\begin{bmatrix} 1 & 0 \\ 0 & T_2 \end{bmatrix} \begin{bmatrix} \sigma_1 & b_{12} & b_{13} \\ 0 & b_{22} & b_{23} \\ 0 & b_{32} & b_{33} \\ 0 & b_{42} & b_{43} \end{bmatrix} = \begin{bmatrix} \sigma_1 & b_{12} & b_{13} \\ 0 & \sigma_2 & c_{23} \\ 0 & 0 & c_{33} \\ 0 & 0 & c_{43} \end{bmatrix}, \tag{19}$$

where

$$U = \begin{bmatrix} b_{22} + \sigma_2 \\ b_{32} \\ b_{42} \end{bmatrix},$$

$$\sigma_2 = \text{sign}(b_{22})\sqrt{b_{22}^2 + b_{32}^2 + b_{42}^2},$$

$$\beta = 2/U'U,$$

$$\gamma_{2j} = b_{2j}u(1) + b_{3j}u(2) + b_{4j}u(3),$$

and

$$c_{ij} = -b_{ij} + \beta\gamma_{2j}u(j-1) \quad \text{for } j = 2 \text{ and } 3 \text{ and } i = 2, 3, \text{ and } 4.$$

Equation (19) is the desired triangular form required of equation (11) for the example. The correspondence is

$$\hat{R}' = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \; \tilde{R}' = \begin{bmatrix} \sigma_1 & b_{12} \\ 0 & \sigma_2 \end{bmatrix},$$

$$H_W = \begin{bmatrix} a_{31} & a_{32} \\ a_{41} & a_{42} \end{bmatrix}, \; Z_M = \begin{bmatrix} a_{33} \\ a_{43} \end{bmatrix},$$

$$\hat{Z} = \begin{bmatrix} a_{13} \\ a_{23} \end{bmatrix}, \; \tilde{Z} = \begin{bmatrix} b_{13} \\ c_{23} \end{bmatrix}, \text{ and } e = \begin{bmatrix} c_{33} \\ c_{43} \end{bmatrix}.$$

The Householder algorithm just described can be compactly encoded in Fortran for general computer operation. In some cases a hardware implementation is desirable and is shown schematically in Fig. 2.

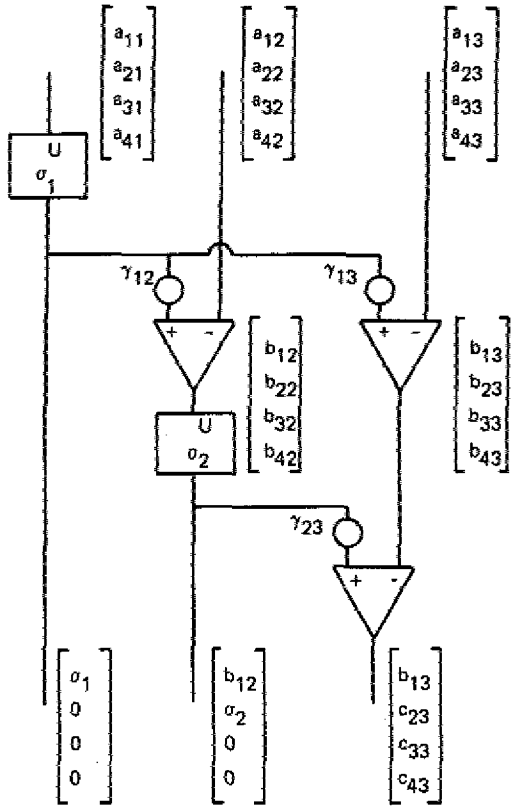

Fig. 2 — Schematic of operations performed with the Householder algorithm

The Kalman and SRIF filters were briefly reviewed to set the notation and acquaint those readers not familiar with algorithms with the salient features. A simple means of obtaining the prediction portion of the SRIF filter under an important special case is next considered.

PREDICTION PROCESS

The smoothing portion of the Kalman filter using SRIF implementation updates the factorization of the smoothed covariance and the transformed best estimate. It is desirable to update the prediction process in a commensurable form. Only an important special case is considered.

The process noise $W(k)$ is assumed to be zero, and the state transition matrix is assumed to be in upper triangular form. Equation (4) updating the prediction covariance then becomes

$$\hat{P} = \Phi\tilde{P}\Phi', \tag{20}$$

where the noise $W(k)$ is removed and the sample k has been dropped for notational convenience. The inverse of (20) is taken, yielding

$$\hat{P}^{-1} = (\Phi')^{-1}\tilde{P}^{-1}\Phi^{-1}. \tag{21}$$

The covariances are replaced with their factorization

$$\hat{R}\,\hat{R}' = (\Phi')^{-1}\tilde{R}\,\tilde{R}'\Phi^{-1},$$

which can be rewritten as

$$\hat{R}\,\hat{R}' = \left[(\Phi')^{-1}\tilde{R}\right]\left[(\Phi')^{-1}\tilde{R}\right]'.$$

Note that $(\Phi')^{-1}\tilde{R}$ is in lower triangular form, which means that

$$\hat{R} = (\Phi')^{-1}\tilde{R}\,. \tag{22}$$

Equation (22) shows the simple form of updating the factor of the prediction covariance.

The predicted state given by

$$\hat{X} = \Phi\tilde{X}$$

from equation (5) is transformed by

$$(\hat{R}')^{-1}\hat{Z} = \Phi(\tilde{R}')^{-1}\tilde{Z}\,,$$

where $\hat{X} = (\hat{R}')^{-1}\hat{Z}$ and $\tilde{X} = (\tilde{R}')^{-1}\tilde{Z}$. Solving for $\hat{Z}$ yields

$$\hat{Z} = (\hat{R}')\Phi(\tilde{R}')^{-1}\tilde{Z}\,. \tag{23}$$

Substituting $\hat{R}$ from (22) into (23) yields

$$\hat{Z} = \tilde{Z}\,. \tag{24}$$

The transformed smoothed and predicted states are seen to be identical.

Sometimes it is desirable to implement a fading-memory filter by making the smoothed covariance larger. This is accomplished by rewriting equation (21) as

$$\hat{P}^{-1} = (\Phi')^{-1}a\tilde{P}^{-1}\Phi^{-1}.$$

The parameter is a scaler representing a time fading by

$$a = e^{-t/\tau},$$

where τ is the time constant and t is time. Equation (22) is modified by

$$\hat{R} = \sqrt{a}(\Phi')^{-1}\tilde{R},$$

and equation (24) remains the same under the fading-memory condition.

IMPLEMENTATION

As an example a tracking problem is taken into consideration. The state transition matrix

$$\Phi = \begin{bmatrix} 1 & t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & t \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{ where } \Phi^{-1} = \begin{bmatrix} 1 & -t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -t \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

represents a target moving in a straight line in a two-dimensional Cartesian coordinate system. The components of the state vector $X(k)$ to be estimated are $X_1(k)$, the position in the ith direction; $X_2(k)$, the velocity in the ith direction; $X_3(k)$, the position in the jth direction; and $X_4(k)$, the velocity in the jth direction. Only the positions are measured; consequently the measurement matrix H is

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

The functional flow of the filter is shown in Fig. 3. The measurement is prewhitened using the Cholesky factorization. In most tracking problems the inverse required can simply be written in closed form using the Cramer rule. The prediction variables are updated with no more than a matrix multiplication. These steps can be mechanized with several degrees of parallelism in hardware. Finally the smoothing is performed using the Householder algorithm shown schematically in Fig. 2. The output of the filter in normal tracking is the statistical distance [2, 3] $J = \hat{e}'e$ which is required for correlation (a direct consequence of the filter) and the predicted position $\tilde{X}$ used in correlation and for display. The outputs are easily obtained, including $\tilde{X}$, because $\tilde{R}^{-1}$ need not be found. The best estimate $\hat{X}$ can be obtained from $\tilde{Z}$ and $\tilde{R}'$ directly by back substitution, since $\tilde{R}'$ is in triangular form. All the operations described including the Householder algorithm are simple operations easily mechanized with parallelism in the hardware.

SUMMARY

The SRIF filter was briefly reviewed, including the Cholesky factorization and Householder algorithm. The smoothing portion of the SRIF filter is claimed to have good numerical characteristics and lends itself to parallel hardware operation. The prediction process under an important simple case was examined. The state transition matrix was assumed to be in upper triangular form, and the process noise was assumed to be zero. Most tracking problems can be formulated in this form. Under this special case it was shown that the transformed smoothed and predicted states were identical and that the smoothed and predicted covariance factors were related by a simple matrix transform. Consequently the entire SRIF filter including both the smoothing and prediction lends itself to hardware implementation.

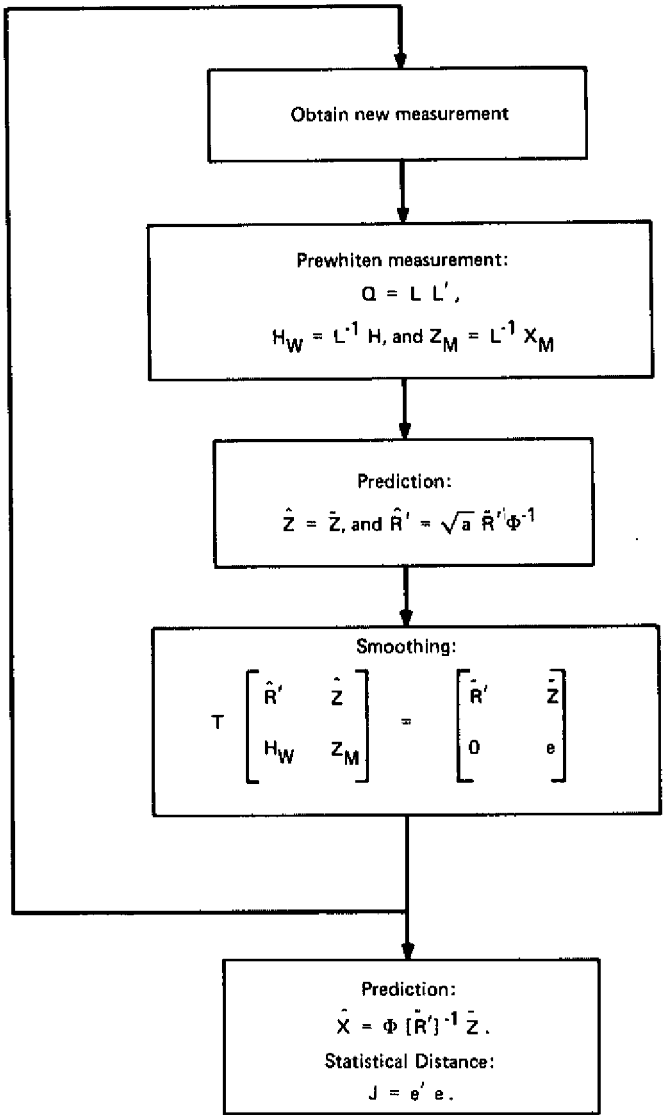

Fig. 3 — Functional flow of the SRIF filter

REFERENCES

1. G. J. Bierman, *Factorization Methods for Discrete Sequential Estimation*, Academic Press, New York, 1976.

2. B. H. Cantrell, "Statistical Correlation Distance for Platform-to-Platform Radar Integration," NRL Report 8123, May 18, 1977 (AD A041429).

3. G. V. Trunk and J. D. Wilson, "Tracking Filters for Multiple-Platform Radar Integration," NRL Report 8087, Dec. 14, 1976 (AD A034608).