New Square-Root Algorithms for Kalman Filtering

PooGyeon Park and Thomas Kailath

Abstract— **We present some new square-root algorithms that allow more reliable computation of the state estimates, using, as far as possible, quantities obtained via orthogonal operations. New algorithms are given for covariance quantities and information quantities, and a new combined algorithm is also presented.**

I. Introduction

For numerical reasons, square-root algorithms are generally preferred for implementation of the recursive Kalman filtering and smoothing formulas; see, for example, the discussion in [1]–[4] and several other studies. Apart from numerical advantages, we may mention that square-root algorithms appear to be better suited to parallel implementation and to very large scale integration (VLSI) implementation. Though there is by now a long history of such algorithms, most recently described in Grewal and Andrews' textbook [5], we have recently discovered some useful new square-root algorithms:

1) A new square-root covariance algorithm that permits easier propagation of the state estimates, along with the square-root factors of the state error covariance matrix.
2) A new square-root information filter that allows computation of the state-estimates without the need for a back-substitution calculation.
3) A new combined covariance and information form square-root algorithm.
4) A new combined covariance and information form square-root algorithm separated into measurement and time updates.

For convenience we first introduce some notational conventions.

Square-Root Factors: Given a positive definite matrix A, $A > 0$, a square-root factor will be defined as any matrix, say $A^{1/2}$, in a such way that $A = (A^{1/2})(A^{1/2})^*$, where the "*" denotes matrix transpose. Such square-root factors are clearly not unique. They can be made unique, e.g., by insisting that the factors be symmetric or that they be triangular (with positive diagonal elements). In most applications, the triangular form is preferred. For convenience we shall also write $(A^{1/2})^* = A^{*/2}$, $(A^{1/2})^{-1} = A^{-1/2}$, $(A^{-1/2})^* = A^{-*/2}$. Thus, let us note the expressions $A = A^{1/2}\ A^{*/2}$, $A^{-1} = A^{-*/2}\ A^{-1/2}$.

Pre-Arrays and Post-Arrays: Given a matrix A, assume that we apply a unitary operator Θ to the A so as to get some special form of a matrix B such as $A\Theta = B$, then we shall call the A a pre-array and the B a post-array.

State-Space Model: $x_{i+1} = F_i x_i + G_i u_i$ and $y_i = H_i x_i + v_i$, where $i \geq 0$, $F_i \in \mathcal{C}^{n\times n}$, $H_i \in \mathcal{C}^{p\times n}$, $G_i \in \mathcal{C}^{n\times m}$, and $\{x_0, u_i, v_i\}$ are random variables with the properties

$$E\begin{bmatrix} x_0 \\ u_i \\ v_i \end{bmatrix}\begin{bmatrix} x_0^* & u_j^* & v_j^* & 1\end{bmatrix} = \begin{bmatrix} \Pi_0 & 0 & 0 & 0 \\ 0 & Q_i\delta_{ij} & S_i\delta_{ij} & 0 \\ 0 & S_i^*\delta_{ij} & R_i\delta_{ij} & 0 \end{bmatrix}.$$

Manuscript received January 19, 1994; revised June 13, 1994. This work was supported in part by the Advanced Research Projects Agency of the Department of Defense and a fellowship of the Jawaharlal Nehru Centre for Advanced Scientific Research, Bangalore, India and was monitored by the Air Force Office of Scientific Research.

The authors are with the Information Systems Laboratory, Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA.

IEEE Log Number 9408788.

The matrices $\{F_i, G_i, H_i, Q_i, S_i, R_i, \Pi_0\}$ are assumed to be known. For the sake of simplicity, we shall assume henceforth that $S_i \equiv 0$. The modifications necessary to handle the more general case will be described in the Appendix.

We shall define

$$\hat{x}_{i|j} \triangleq \text{the linear least-squares estimate of } x_i \text{ given } \{y_0, \cdots, y_j\},$$

$$P_{i|j} \triangleq E[x_i - x_{i|j}][x_i - x_{i|j}]^*, \text{ the error covariance of the estimate } \hat{x}_{i|j}.$$

When $j = i-1$, $\hat{x}_{i|i-1}$ is called the one-step predicted estimate, $j = i$, $\hat{x}_{i|i}$ is called the filtered estimate. A corresponding terminology will be used for the error-variance matrices. For compactness we shall write, unless necessary for emphasis or comparison, $\hat{x}_{i|i-1} \triangleq \hat{x}_i$, $P_{i|i-1} \triangleq P_i$.

The above problems will be well posed if and only if the covariance matrix of the stacked output $\{y_0, \cdots, y_j\}$ is strictly positive definite. We may note that this condition may hold even if the matrices $\{R_i\}$ are themselves only positive-semi-definite, i.e., $R_i \geq 0$. While this possibility exists, it is also well known that the numerical condition of the problem is much improved if the model is defined so that not only $R_i > 0$ and but also, in fact, the R_i are as well conditioned as possible, i.e., roughly speaking the attempt is made to move them closer to the identity matrix. No such constraint needs to be imposed on the Q_i, which are, of course, also such that $Q_i \geq 0$; in fact, they can be identically zero, as happens in parameter estimation and adaptive filtering problems.

II. Kalman Filtering

Given the standard state-space model described above, the basic Kalman Filter (KF) formulas are the following algorithm.

Algorithm II.1—Kalman Filtering: The one-step predicted estimates obey the recursion

$$\hat{x}_{i+1} = F_i\hat{x}_i + K_i R_{e,i}^{-1}(y_i - H_i\hat{x}_i), \hat{x}_0 = 0 \tag{1}$$

where $K_i \triangleq F_i P_i H_i^*$, $R_{e,i} \triangleq R_i + H_i P_i H_i^*$, and the P_i are propagated via the difference Riccati equation (recursion) (DRE)

$$P_{i+1} = F_i P_i F_i^* + G_i Q_i G_i^* - K_i R_{e,i}^{-1} K_i^*, P_0 = \Pi_0. \tag{2}$$

It is useful to define the innovations $e_i \triangleq y_i - H_i\hat{x}_i$, and note that $R_{e,i} = Ee_ie_i^*$, while $K_i = Ex_{i+1}e_i^*$. An assumption about the well-posedness of the problem that the covariance of the stacked output $\{y_0, \cdots, y_N\}$ is strictly positive definite can be shown to ensure that, for $0 \leq i \leq N$, $R_{e,i} > 0$. Though not necessary for Algorithm II.1, it is well known that for numerical reasons it is desirable to choose state-space models in which $R_i > 0$. Then we can write, when $P_i > 0$

$$K_{p,i} \triangleq K_i R_{e,i}^{-1} = F_i P_i (I + H_i^* R_i^{-1} H_i P_i)^{-1} H_i^* R_i^{-1}$$
$$= F_i (P_i^{-1} + H_i^* R_i^{-1} H_i)^{-1} H_i^* R_i^{-1}.$$

Under the assumptions $R_i > 0$, $\Pi_0 > 0$, and nonsingular F_i, it can be proved that $P_i > 0$. A simple proof is by induction. If $P_i > 0$, we can rearrange the DRE as

$$P_{i+1} = G_i Q_i G_i^* + F_i(P_i^{-1} + H_i^* R_i^{-1} H_i)^{-1} F_i^*.$$

Therefore, $P_{i+1} \geq F_i(P_i^{-1} + H_i^* R_i^{-1} H_i)^{-1} F_i^* > 0$, and $\Pi_0 > 0$ allows the induction to begin.

Information Filter Forms

When Π_0 is very large, it is often preferable to propagate P_i^{-1}, which can be done when $\Pi_0 > 0$, $R_i > 0$, and the F_i are invertible. The resulting so-called information filter formulas are rather complicated in the general case, so they are generally represented separately in terms of what are called measurement and time updates.

Measurement Updates: Here we incorporate the measurement y_i to update the predicted estimate. The appropriate formulas are

$$\begin{aligned} \hat{x}_{i|i} &= \hat{x}_i + P_i H_i^* R_{e,i}^{-1} e_i \\ &= \text{the measurement-updated estimate of } \hat{x}_i, \\ P_{i|i} &= P_i - P_i H_i^* R_{e,i}^{-1} H_i P_i \\ &= E(x_i - \hat{x}_{i|i})(x_i - \hat{x}_{i|i})^*. \end{aligned}$$

To obtain the information filter formulas, we can use a matrix inversion formula to obtain

$$\begin{aligned} P_{i|i}^{-1} \hat{x}_{i|i} &= P_i^{-1} \hat{x}_i + H_i^* R_i^{-1} y_i, \\ P_{i|i}^{-1} &= P_i^{-1} + H_i^* R_i^{-1} H_i. \end{aligned} \tag{3}$$

Time Updates: The step of going from $\hat{x}_{i|i}$ to $\hat{x}_{i+1}$ is called a time-update. The appropriate formulas are

$$\begin{aligned} \hat{x}_{i+1} &= F_i \hat{x}_{i|i}, \\ P_{i+1} &= F_i P_{i|i} F_i^* + G_i Q_i G_i^*. \end{aligned}$$

To get the information filter formulas, we again apply a matrix inversion formula to obtain

$$\begin{aligned} P_{i+1}^{-1} &= A_i - A_i G_i (Q_i^{-1} + G_i^* A_i G_i)^{-1} G_i^* A_i, \\ P_{i+1}^{-1} \hat{x}_{i+1} &= (I + A_i G_i Q_i G_i^*)^{-1} (F_i^{-*} P_{i|i}^{-1} \hat{x}_{i|i}) \end{aligned} \tag{4}$$

where $A_i \triangleq F_i^{-*} P_{i|i}^{-1} F_i^{-1}$. We can also write

$$Q_i^{-1} + G_i^* A_i G_i = \tilde{Q}_i^{-1}, \quad \tilde{Q}_i = Q_i - Q_i G_i^* P_{i+1}^{-1} G_i Q_i$$

where $\tilde{Q}_i$ is encountered in the study of smoothed estimates.

III. Square-Root Algorithms

A. Square-Root Covariance Filtering (SRCF)

Because of the accumulation of numerical errors, the matrices P_i propagated by the Riccati equation can lose their theoretically required positive-definiteness. In some situations, even the diagonal entries of P_i may become negative, resulting in totally meaningless state estimates. To protect against such circumstances, it is now widely recommended to propagate square-root factors, $P_i^{1/2}$. While numerical effects will still be present, forming $(P_i^{1/2})(P_i^{*/2})$ is much more likely to lead to a positive definite matrix; in fact, the diagonal elements of the product will now always be positive. Propagation of square-root factors also has the advantage of better conditioning, reduced dynamic range for fixed-point implementation, and easier parallel and systolic array implementation.

There are several forms of square-root algorithms. The basic so-called square-root covariance filter (SRCF) algorithm is the following algorithm.

Let us form the pre-array

$$\mathcal{A} = \begin{bmatrix} R_i^{1/2} & H_i P_i^{1/2} & 0 \\ 0 & F_i P_i^{1/2} & G_i Q_i^{1/2} \end{bmatrix}$$

and block triangularize it by applying any orthogonal rotation Θ so that

$$\mathcal{A}\Theta = \begin{bmatrix} X & 0 & 0 \\ Y & Z & 0 \end{bmatrix}.$$

We may regard this as saying that the rotation Θ sets up a conformal (i.e., a norm- and angle-preserving) mapping between the (block) rows of $\mathcal{A}$ and the rows of the post-array. So, for example

$$\begin{aligned} &\langle [R_i^{1/2} \quad H_i P_i^{1/2} \quad 0], [R_i^{1/2} \quad H_i P_i^{1/2} \quad 0] \rangle \\ &\quad = \langle [X \quad 0 \quad 0], [X \quad 0 \quad 0] \rangle, \\ &\langle [R_i^{1/2} \quad H_i P_i^{1/2} \quad 0], [0 \quad F_i P_i^{1/2} \quad G_i Q_i^{1/2}] \rangle \\ &\quad = \langle [X \quad 0 \quad 0], [Y \quad Z \quad 0] \rangle. \end{aligned}$$

Therefore, we have the relations $R_i + H_i P_i H_i^* = XX^*$ and $H_i P_i F_i^* = XY^*$, from which we can identify

$$X = R_{e,i}^{1/2}, \quad Y = F_i P_i H_i^* R_{e,i}^{-*/2} (\triangleq \bar{K}_{p,i})$$

recalling that $K_{p,i} = K_i R_{e,i}^{-1} = F_i P_i H_i^* (R_{e,i}^{-*/2} R_{e,i}^{-1/2})$. Finally, from the second block rows we obtain

$$F_i P_i F_i^* + G_i Q_i G_i^* = \bar{K}_{p,i} \bar{K}_{p,i}^* + ZZ^*$$

so that, by the Riccati equation, we can identify $Z = P_{i+1}^{1/2}$. Therefore, we have established the following algorithm.

Algorithm III.1—Conventional SRCF: Given $P_0^{1/2} = \Pi_0^{1/2}$

$$\begin{bmatrix} R_i^{1/2} & H_i P_i^{1/2} & 0 \\ 0 & F_i P_i^{1/2} & G_i Q_i^{1/2} \end{bmatrix} \Theta_i = \begin{bmatrix} R_{e,i}^{1/2} & 0 & 0 \\ \bar{K}_{p,i} & P_{i+1}^{1/2} & 0 \end{bmatrix}$$

where $\bar{K}_{p,i} = K_i R_{e,i}^{-*/2}$ and Θ_i is any orthogonal rotation, $\Theta_i \Theta_i^* = I$, that block-triangularizes the pre-array.

Remark III-A.1: The rotation Θ_i can be implemented in various ways—as a sequence of elementary circular (Givens) rotations, producing zeros one entry at a time, or as a sequence of elementary Householder reflections, producing a row of zeros at a time, or various combinations thereof. We could also use weighted orthogonal matrices obeying $\Theta_i D_{1,i} \Theta_i^* = D_{2,i}$, where $D_{1,i}$ and $D_{2,i}$ are diagonal matrices chosen so as to avoid arithmetic square-roots and division operations in the triangularization procedure. See, for example, the recent comprehensive study of such matters by Hsieh *et al.* [6] and Götze and Schweigelshohn [7], following the early work [8] and [9]. Here we may only mention that the so-called U-D algorithms of Bierman and Thornton [4] arise from just such choices (see, e.g., [10]).

Remark III-A.2: Note that the triangularization could also be affected by the use of block rotations and reflections, which, however, will generally be more complicated than by the use of only scalar operations as mentioned above. We may also note that in the conventional Riccati recursions, special sequential updating algorithms are used to avoid the $p \times p$ matrix inversion of the $R_{e,i}$. These "scalarization" results come for "free" in the square-root algorithms.

Remark III-A.3: Though numerically reliable orthogonal operations are used to propagate the covariance and to compute the $R_{e,i}^{1/2}$ and $\bar{K}_{p,i}$, the state estimates are still to be obtained as, for $\hat{x}_0 = 0$

$$\hat{x}_{i+1} = (F_i)(\hat{x}_i) + (\bar{K}_{p,i})(R_{e,i}^{1/2})^{-1}(y_i - H_i \hat{x}_i). \tag{5}$$

We still have a $p \times p$ matrix inversion, despite a triangular matrix $R_{e,i}^{1/2}$ (so that the inversion step can be replaced by solving linear equations by backward substitution).

The following extended algorithm mitigates both of the above problems. We augment the pre-array of the SRCF by an additional row and apply the same orthogonal rotation as determined in the SRCF to obtain the following array form

$$\begin{bmatrix} R_i^{1/2} & H_iP_i^{1/2} & 0 \\ 0 & F_iP_i^{1/2} & G_iQ_i^{1/2} \\ \hline -y_i^*R_i^{-*/2} & \hat{x}_i^*P_i^{-*/2} & 0 \end{bmatrix} \Theta_i = \begin{bmatrix} R_{e,i}^{1/2} & 0 & 0 \\ \bar{K}_{p,i} & P_{i+1}^{1/2} & 0 \\ \hline \alpha & \beta & \gamma \end{bmatrix}.$$

Comparing the inner products of the first and third block rows on both sides yields

$$(R_{e,i}^{1/2})(\alpha^*) = (R_i^{1/2})(-R_i^{-1/2}y_i) + (H_iP_i^{1/2})(P_i^{-1/2}\hat{x}_i)$$

so that we can identify $\alpha = -e_i^*R_{e,i}^{-*/2} = -\bar{e}_i^*$, the normalized innovations. Similarly, the inner product of the second and the third block rows provides $F_i\hat{x}_i = \bar{K}_{p,i}\alpha^* + P_{i+1}^{1/2}\beta^*$ so that we can identify

$$\beta^* = P_{i+1}^{-1/2}(F_i\hat{x}_i + K_iR_{e,i}^{-1}e_i) = P_{i+1}^{-1/2}\hat{x}_{i+1}.$$

The quantity γ appears to be messy and to have no special value in Kalman filtering, so we shall leave it undetermined (however, see Remark IIIA.6). We can summarize these results in the following algorithm

Algorithm III.2—Extended SRCF (eSRCF): Assume that $R_i > 0$. Given $P_0^{1/2} = \Pi_0^{1/2}$ and $P_0^{-1/2}\hat{x}_0 = 0$

$$\begin{bmatrix} R_i^{1/2} & H_iP_i^{1/2} & 0 \\ 0 & F_iP_i^{1/2} & G_iQ_i^{1/2} \\ \hline -y_i^*R_i^{-*/2} & \hat{x}_i^*P_i^{-*/2} & 0 \end{bmatrix} \Theta_i = \begin{bmatrix} R_{e,i}^{1/2} & 0 & 0 \\ \bar{K}_{p,i} & P_{i+1}^{1/2} & 0 \\ \hline -e_i^*R_{e,i}^{-*/2} & \hat{x}_{i+1}^*P_{i+1}^{-*/2} & (*) \end{bmatrix} \tag{6}$$

where "(*)" indicates a "don't care" entry and Θ_i is any orthogonal rotation that block-triangularizes the first two (block) rows of the pre-array. The estimate $\hat{x}_{i+1}$ can now be constructed as $\hat{x}_{i+1} = (P_{i+1}^{1/2})(P_{i+1}^{-1/2}\hat{x}_{i+1})$.

Remark III-A.4: Since the state estimates are now found as products of quantities that are available from the post-array, this algorithm is more amendable for the parallel implementation than the conventional SRCF algorithm. This method may employ a modified version of (5), which uses the normalized innovation $(R_{e,i}^{-1/2}e_i)$ of the post-array, $\hat{x}_{i+1} = (F_i)(\hat{x}_i) + (\bar{K}_{p,i})(R_{e,i}^{-1/2}e_i)$. We note that the latter method may entail a little less computation when the matrix F_i is sparse. We may also note that if our major interest is in updating the estimates, then some computation may be reduced in the eSRCF by not computing the entry $\bar{K}_{p,i}$ in the post-array.

Remark III-A.5: We notice from Algorithm III.2 that the only quantities for which inverses are required are the matrices R_i or, rather, their (Cholesky) triangular factors $R_i^{1/2}$. (If $Ex_0 \neq 0$, then we shall require $\Pi_0^{-1/2}$.) This shows that the numerical behavior of the algorithms depends heavily on the properties of the R_i, which, in fact, has been pointed out in the few (partial) numerical analysis studies of the KF (see, e.g., [3], [4]). The importance of proper modeling in which we seek to properly scale the R_i to be as near identity matrices as possible is also seen from the above algorithm, where the only inverses required are of the matrices $R_i^{1/2}$.

Remark III-A.6: With some effort, an explicit formula can be formed for the (*) quantity, viz. $\gamma = -\hat{x}_{i+1}^*K_{b,i}^*\tilde{Q}_i^{-*/2}$, where $K_{b,i} = Q_iG_iP_{i+1}^{-1}$ and $\tilde{Q}_i = Q_i - Q_iG_i^*P_{i+1}^{-1}G_iQ_i$. These apparently complicated quantities actually arise in the Rauch–Tung–Striebel formulas for the smoothed estimate $\hat{x}_{i|N}$.

Remark III-A.7: The reason for the appearance of $P_{i+1}^{-1/2}$ and $R_{e,i}^{-1/2}$ goes back to the (geometric) derivation of the square-root algorithms, in which we are required to orthogonalize a basis $\{e_i, x_{i+1} - \hat{x}_{i+1}\}$ (see [2]).

B. Square-Root Information Filtering (SRIF)

As mentioned before, there are situations in which it is desirable to propagate the inverse quantities P_i^{-1}, $P_i^{-1/2}$. One convenient way of obtaining such algorithms is to augment the SRCF to construct nonsingular arrays that can be inverted. Thus, with any nonsingular $W \in C^{m\times m}$, let us form the augmented arrays

$$\begin{bmatrix} R_i^{1/2} & H_iP_i^{1/2} & 0 \\ 0 & F_iP_i^{1/2} & G_iQ_i^{1/2} \\ 0 & 0 & W \end{bmatrix} \Theta_i = \begin{bmatrix} R_{e,i}^{1/2} & 0 & 0 \\ \bar{K}_{p,i} & P_{i+1}^{1/2} & 0 \\ Y_1 & Y_2 & Y_3 \end{bmatrix}$$

where forming inner products yields $Y_1 = 0$, $Y_2 = WQ_i^{*/2}G_i^* P_{i+1}^{-*/2}$, $Y_3 = (WW^* - Y_2Y_2^*)^{1/2}$. Convenient choices of W are $W = I$ or $W = Q_i^{1/2}$. Making the first choice, we can invert and transpose the above equation to obtain the following result.

Algorithm III.3—Square-Root Information Filtering (SRIF): Assume that $R_i > 0$, $\Pi_0 > 0$, and the F_i are invertible. Given $P_0^{-*/2} = \Pi_0^{-*/2}$ and $P_0^{-*/2}\hat{x}_0 = 0$

$$\begin{bmatrix} R_i^{-*/2} & 0 & 0 \\ -F_i^{-*}H_i^*R_i^{-*/2} & F_i^{-*}P_i^{-*/2} & 0 \\ Q_i^{*/2}G_i^*F_i^{-*}H_i^*R_i^{-*/2} & -Q_i^{*/2}G_i^*F_i^{-*}P_i^{-*/2} & I \\ \hline -y_i^*R_i^{-*/2} & \hat{x}_i^*P_i^{-*/2} & 0 \end{bmatrix} \Theta_i = \begin{bmatrix} R_{e,i}^{-*/2} & -K_{p,i}^*P_{i+1}^{-*/2} & (*) \\ 0 & P_{i+1}^{-*/2} & (*) \\ 0 & 0 & (*) \\ \hline -e_i^*R_{e,i}^{-*/2} & \hat{x}_{i+1}^*P_{i+1}^{-*/2} & (*) \end{bmatrix}$$

where we note that Θ_i can be the same rotation as in the SRCF algorithms. Alternatively, we could define Θ_i directly as any orthogonal rotation that upper-triangularizes the second and third two (block) rows of the pre-array.

Remark III-B.1: In the above array form, we have added the same "data" row as in the eSRCF algorithm here; since Θ_i is the same in both arrays, the resulting right-hand sides are also the same. (As a matter of interest, we note that the "data" rows have always traditionally been included with the information form arrays—see, e.g. [1], [2].)

Remark III-B.2: In the SRIF, the first (block) row can be ignored unless we are interested in finding $R_{e,i}$.

Remark III-B.3: The predicted estimate can be found from the entries of the post-array by solving the triangular system $(P_{i+1}^{-1/2})(\hat{x}_{i+1}) = (P_{i+1}^{-1/2}\hat{x}_{i+1})$. This backward substitution calculation can be avoided, however, by adding an additional (fourth block) row to the SRIF, as shown below.

Algorithm III.4—Modified Square-Root Information Filtering (mSRIF): Assume that $R_i > 0$, $\Pi_0 > 0$, and the F_i are invertible. Given $P_0^{-*/2} = \Pi_0^{-*/2}$, $P_0^{1/2} = \Pi_0^{1/2}$, and $P_0^{-*/2}\hat{x}_0 = 0$

$$\left[\begin{array}{ccc} R_i^{-*/2} & 0 & 0 \\ -F_i^{-*}H_i^*R_i^{-*/2} & F_i^{-*}P_i^{-*/2} & 0 \\ Q_i^{*/2}G_i^*F_i^{-*}H_i^*R_i^{-*/2} & -Q_i^{*/2}G_i^*F_i^{-*}P_i^{-*/2} & I \\ 0 & F_iP_i^{1/2} & G_iQ_i^{1/2} \\ \hline -y_i^*R_i^{-*/2} & \hat{x}_i^*P_i^{-*/2} & 0 \end{array}\right] \times \Theta_i = \left[\begin{array}{ccc} R_{e,i}^{-*/2} & -K_{p,i}^*P_{i+1}^{-*/2} & (*) \\ 0 & P_{i+1}^{-*/2} & (*) \\ 0 & 0 & (*) \\ \bar{K}_{p,i} & P_{i+1}^{1/2} & 0 \\ \hline -e_i^*R_{e,i}^{-*/2} & \hat{x}_{i+1}^*P_{i+1}^{-*/2} & (*) \end{array}\right]$$

where Θ_i is any orthogonal rotation that upper-triangularizes the second and third (block) rows of the pre-array.

Now we have available the term $(P_{i+1}^{1/2})$, so that we can directly form $\hat{x}_{i+1} = (P_{i+1}^{1/2})(P_{i+1}^{-1/2}\hat{x}_{i+1})$.

C. Combined Arrays

The reader may have noticed that the pre-array in Algorithm III.4 is a partial combination of the covariance and information-form arrays. In fact, we can have a full combination as illustrated below.

Algorithm III.5—Combined Square-Root Filtering (cSRF): Assume that $R_i > 0$, $\Pi_0 > 0$, and the F_i are invertible. Given $P_0^{-*/2} = \Pi_0^{-*/2}$, $P_0^{1/2} = \Pi_0^{1/2}$, and $P_0^{-*/2}\hat{x}_0 = 0$

$$\left[\begin{array}{ccc} R_i^{1/2} & H_iP_i^{1/2} & 0 \\ 0 & F_iP_i^{1/2} & G_iQ_i^{1/2} \\ \hline -F_i^{-*}H_i^*R_i^{-*/2} & F_i^{-*}P_i^{-*/2} & 0 \\ Q_i^{*/2}G_i^*F_i^{-*}H_i^*R_i^{-*/2} & -Q_i^{*/2}G_i^*F_i^{-*}P_i^{-*/2} & I \\ \hline -y_i^*R_i^{-*/2} & \hat{x}_i^*P_i^{-*/2} & 0 \end{array}\right] \times \Theta_i = \left[\begin{array}{ccc} R_{e,i}^{1/2} & 0 & 0 \\ \bar{K}_{p,i} & P_{i+1}^{1/2} & 0 \\ \hline 0 & P_{i+1}^{-*/2} & -K_{b,i}^*\tilde{Q}_i^{-*/2} \\ 0 & 0 & \tilde{Q}_i^{-*/2} \\ \hline -e_i^*R_{e,i}^{-*/2} & \hat{x}_{i+1}^*P_{i+1}^{-*/2} & -\hat{x}_{i+1}^*K_{b,i}^*\tilde{Q}_i^{-*/2} \end{array}\right]$$

where Θ_i is any orthogonal rotation that either lower-triangularizes the first two (block) rows or upper-triangularizes the third and fourth (block) rows of the pre-array. As before, $\hat{x}_{i+1}$ can be constructed as one of

$$\hat{x}_{i+1} = (P_{i+1}^{1/2})(P_{i+1}^{-1/2}\hat{x}_{i+1}), \tag{7}$$

$$\hat{x}_{i+1} = (F_i)(\hat{x}_i) + (\bar{K}_{p,i})(R_{e,i}^{-1/2}e_i). \tag{8}$$

Remark III-C.1—Numerical Properties: The rotation Θ_i may be determined from either the first two (block) rows or the third and fourth. Theoretically, either alternative will give the same Θ_i, but numerically the two possibilities will no longer agree. We can get some insights about numerical properties from Verhaegen and Van Dooren's paper [3], which shows that the condition number of the innovation variance matrix, $R_{e,i}$, is the key quantity in determining the numerical behavior of the covariance algorithm, while the condition number of the (sort of a covariance) matrix, $\tilde{Q}_i$, is what controls the numerical behavior of the information algorithm.

Remark III-C.2—Computational Complexity: Since all rows of the combined square-root array will not generally be used, the computational complexity depends on the situation. As far as the complexity of the SRCF and the SRIF are concerned, the Verhaegen and Van Dooren's paper [3] treats in details.

Remark III-C.3—Usefulness of the cSRF: We do not have to use the full array form of the combined square-root algorithm. That is, we employ only parts of the square-root array, according to the problem; for example, in smoothing problems, we need several combinations of the rows of the square-root array, as shown in [11].

IV. Measurement and Time Updates

As noted in Section II, the general information filter equations are somewhat complicated, so that they are usually separated into measurement and time update formulas. It is interesting to note that *this is not really necessary with square-root forms—the complication* is absorbed into the "don't care" terms! (see Algorithm III.3). It may be useful to note, however, the corresponding measurement and time update formulas.

Measurement Updates:

$$\left[\begin{array}{cc} H_i^*R_i^{-*/2} & P_i^{-*/2} \\ \hline -y_i^*R_i^{-*/2} & \hat{x}_i^*P_i^{-*/2} \end{array}\right]\Theta_{i,1} = \left[\begin{array}{cc} 0 & P_{i|i}^{-*/2} \\ \hline -e_i^*R_{e,i}^{-*/2} & \hat{x}_{i|i}^*P_{i|i}^{-*/2} \end{array}\right]$$

where $\Theta_{i,1}$ is any orthogonal rotation that zeros out the (1, 1) block element in the post-array.

Time Updates:

$$\left[\begin{array}{cc} -G_i^*F_i^{-*}P_{i|i}^{-*/2} & Q_i^{-*/2} \\ F_i^{-*}P_{i|i}^{-*/2} & 0 \\ \hline \hat{x}_{i|i}^*P_{i|i}^{-*/2} & 0 \end{array}\right]\Theta_{i,2} = \left[\begin{array}{cc} 0 & \tilde{Q}_i^{-*/2} \\ P_{i+1}^{-*/2} & -F_i^{-*}P_{i|i}^{-1}F_i^{-1}G_i\tilde{Q}_i^{1/2} \\ \hline \hat{x}_{i+1}^*P_{i+1}^{-*/2} & -\hat{x}_{i|i}^*P_{i|i}^{-1}F_i^{-1}G_i\tilde{Q}_i^{1/2} \end{array}\right]$$

where $\Theta_{i,2}$ is any orthogonal rotation that zeros out the (1, 1) block element in the post-array. We may note that the (3, 2) element $-\tilde{Q}_i^{*/2}G_i^*F_i^{-*}P_{i|i}^{-1}\hat{x}_{i|i}$ in the post-array is

$$\begin{aligned} &-\tilde{Q}_i^{*/2}G_i^*F_i^{-*}P_{i|i}^{-1}F_i^{-1}\hat{x}_{i+1} \\ &= -\tilde{Q}_i^{*/2}G_i^*(P_{i+1} - G_iQ_iG_i^*)^{-1}\hat{x}_{i+1} \\ &= -\tilde{Q}_i^{*/2}(Q_i - Q_iG_i^*P_{i+1}^{-1}G_iQ_i)^{-1}Q_iG_iP_{i+1}^{-1}\hat{x}_{i+1} \end{aligned}$$

which is the same as the quantity γ^* identified in Remark III-A.6.

Similarly, we can derive the square-root covariance filtering algorithms into measurement and time update form. We note the corresponding combined forms below.

Combined Measurement Update Square-Root Filtering:

$$\left[\begin{array}{cc} R_i^{1/2} & H_iP_i^{1/2} \\ 0 & F_iP_i^{1/2} \\ \hline -H_i^*R_i^{-*/2} & P_i^{-*/2} \\ \hline -y_i^*R_i^{-*/2} & \hat{x}_i^*P_i^{-*/2} \end{array}\right]\Theta_{i,1} = \left[\begin{array}{cc} R_{e,i}^{1/2} & 0 \\ \bar{K}_{p,i} & F_iP_{i|i}^{1/2} \\ \hline 0 & P_{i|i}^{-*/2} \\ \hline -e_i^*R_{e,i}^{-*/2} & \hat{x}_{i|i}^*P_{i|i}^{-*/2} \end{array}\right]$$

where $\Theta_{i,1}$ is any orthogonal rotation that zeros out the (1, 2) or the (3, 1) entry of the pre-array.

Combined Time Update Square-Root Filtering:

$$\left[\begin{array}{cc} F_iP_{i|i}^{1/2} & G_iQ_i^{1/2} \\ \hline F_i^{-*}P_{i|i}^{-*/2} & 0 \\ -G_i^*F_i^{-*}P_{i|i}^{-*/2} & Q_i^{-*/2} \\ \hline \hat{x}_{i|i}^*P_{i|i}^{-*/2} & 0 \end{array}\right]\Theta_{i,2} = \left[\begin{array}{cc} P_{i+1}^{1/2} & 0 \\ \hline P_{i+1}^{-*/2} & -K_{b,i}^*\tilde{Q}_i^{-*/2} \\ 0 & \tilde{Q}_i^{-*/2} \\ \hline \hat{x}_{i+1}^*P_{i+1}^{-*/2} & -\hat{x}_{i+1}^*K_{b,i}^*\tilde{Q}_i^{-*/2} \end{array}\right]$$

where $\Theta_{i,2}$ is any orthogonal rotation that zeros out the (1, 2) or (3, 1) entry of the pre-array.

Appendix
The Case of $S_i \not\equiv 0$

As promised earlier, we note ways of handling nonzero S_i.

Case 1) $Q_i > 0$ and $\hat{R}_i \triangleq R_i - S_i^*Q_i^{-1}S_i > 0$,

Case 2) $R_i > 0$ and $\hat{Q}_i \triangleq Q_i - S_iR_i^{-1}S_i^* > 0$.

Algorithm V.1. (cSRF) Case 1: $Q_i > 0$ and $\hat{R}_i > 0$: Assume that $\Pi_0 > 0$ and the F_i are invertible. Given $P_0^{1/2} = \Pi_0^{1/2}$,, $P_0^{-*/2} = \Pi_0^{-*/2}$, and $P_0^{-*/2}\hat{x}_0 = 0$

$$\left[\begin{array}{ccc} \hat{R}_i^{1/2} & H_iP_i^{1/2} & S_i^*Q_i^{-*/2} \\ 0 & F_iP_i^{1/2} & G_iQ_i^{1/2} \\ \hline -F_i^{-*}H_i^*\hat{R}_i^{-*/2} & F_i^{-*}P_i^{-*/2} & 0 \\ E_i^*\hat{R}_i^{-*/2} & -G_i^*F_i^{-*}P_i^{-*/2} & Q_i^{-*/2} \\ \hline -y_i^*\hat{R}_i^{-*/2} & \hat{x}_i^*P_i^{-*/2} & 0 \end{array}\right]\Theta_i = \left[\begin{array}{ccc} R_{e,i}^{1/2} & 0 & 0 \\ \bar{K}_{p,i} & P_{i+1}^{1/2} & 0 \\ \hline 0 & P_{i+1}^{-*/2} & (*) \\ 0 & 0 & (*) \\ \hline -e_i^*R_{e,i}^{-*/2} & \hat{x}_{i+1}^*P_{i+1}^{-*/2} & (*) \end{array}\right]$$

where $E_i \triangleq H_iF_i^{-1}G_i - S_i^*Q_i^{-1}$ and Θ_i is any orthogonal rotation that either lower-triangularizes the first two (block) rows or upper-triangularizes the middle two (block) rows of the pre-array. The predicted estimates $\hat{x}_{i+1}$ can be constructed as (7) or (8).

Algorithm V.2 (cSRF) Case 2: $R_i > 0$ and $\hat{Q}_i > 0$: Assume that $\Pi_0 > 0$ and $\hat{F}_i$ are invertible. Given $P_0^{1/2} = \Pi_0^{1/2}$, $P_0^{-*/2} = \Pi_0^{-*/2}$, and $P_0^{-*/2}\hat{x}_0 = 0$

$$\left[\begin{array}{ccc} R_i^{1/2} & H_iP_i^{1/2} & 0 \\ 0 & \hat{F}_iP_i^{1/2} & G_i\hat{Q}_i^{1/2} \\ \hline -\hat{F}_i^{-*}H_i^*R_i^{-*/2} & \hat{F}_i^{-*}P_i^{-*/2} & 0 \\ G_i^*\hat{F}_i^{-*}H_i^*R_i^{-*/2} & -G_i^*\hat{F}_i^{-*}P_i^{-*/2} & \hat{Q}_i^{-*/2} \\ \hline -y_i^*R_i^{-*/2} & \hat{x}_i^*P_i^{-*/2} & y_i^*R_i^{-1}S_i^*\hat{Q}_i^{-*/2} \end{array}\right]\Theta_i = \left[\begin{array}{ccc} R_{e,i}^{1/2} & 0 & 0 \\ \hat{K}_{p,i} & P_{i+1}^{1/2} & 0 \\ \hline 0 & P_{i+1}^{-*/2} & (*) \\ 0 & 0 & (*) \\ \hline -e_i^*R_{e,i}^{-*/2} & \hat{x}_{i+1}^*P_{i+1}^{-*/2} & (*) \end{array}\right]$$

where $\hat{F}_i \triangleq F_i - G_iS_iR_i^{-1}H_i$, $\hat{K}_{p,i} \triangleq \hat{F}_iP_iH_i^*R_{e,i}^{-*/2}$, and Θ_i is any orthogonal rotation that either lower-triangularizes the first two (block) rows or upper-triangularizes the middle two (block) rows of the pre-array. The predicted estimates $\hat{x}_{i+1}$ can be constructed as (7) or as $\hat{x}_{i+1} = (\hat{F}_i)(\hat{x}_i) + (\hat{K}_{p,i})(R_{e,i}^{-1/2}e_i) + G_iS_iR_i^{-1}y_i$.

An interesting distinction between Case 1) and Case 2) (if both apply) is that only the algorithm in Case 2) can be separated into measurement and time update steps.

References

[1] P. G. Kaminski, A. E. Bryson, and S. F. Schmidt, "Discrete square-root filtering—A survey of current techniques," *IEEE Trans. Automat. Contr.*, vol. AC-16, pp. 727–736, 1971.

[2] M. Morf and T. Kailath, "Square root algorithms for least squares estimation," *IEEE Trans. Automat. Contr.*, vol. AC-20, pp. 487–497, Aug. 1975.

[3] M. Verhaegen and P. V. Dooren, "Numerical aspects of different Kalman filter implementation," *IEEE Trans. Automat. Contr.*, vol. AC-31, pp. 907–917, Oct. 1986.

[4] G. J. Bierman and C. L. Thornton, "Numerical comparison of Kalman filter algorithms: Orbit determination case study," *Automatica*, vol. 13, p. 23, 1977.

[5] M. S. Grewal and A. P. Andrews, *Kalman Filtering*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[6] S. F. Hsieh, K. J. R. Liu, and K. Yao, "A unified square-root-free approach for QRD-based recursive least squares estimation," *IEEE Trans. Sig. Proces.*, vol. 41, pp. 1405–1409, Mar. 1993.

[7] J. Götze and U. Schweigelsohn, "A square root and division free givens rotation for solving least squares problems on systolic arrays," *SIAM J. Sci. Stat. Comp.*, vol. 12, pp. 800–807, July 1991.

[8] W. M. Gentleman, "Least squares computations by givens transformation without square roots," *J. Inst. Math. Appl.*, vol. 12, pp. 329–336, 1973.

[9] S. Hammarling, "A note on modifications to the givens plane rotation," *J. Inst. Math. Appl.*, vol. 13, pp. 215–218, 1974.

[10] J. M. Jover and T. Kailath, "A parallel architecture for Kalman filter measurement update and parameter estimation," *Automatica*, vol. 22, no. 1, pp. 43–57, 1986.

[11] P. Park and T. Kailath, "New square-root smoothing algorithms," in *Proc. IEEE Conf. Decis. Contr.*, Orlando, FL, Dec. 1994.