Proceedings of the 33rd
Conference on Decision and Control
Lake Buena Vista, FL - December 1994

Square-Root Arrays and Chandrasekhar Recursions for H^∞ Problems*

Babak Hassibi[†] Ali H. Sayed[‡] and Thomas Kailath

Information Systems Laboratory
Stanford University, Stanford CA 94305

Abstract

Using our recent observation that H^∞ filtering coincides with Kalman filtering in Krein space we develop square-root arrays and Chandraskhar recursions for H^∞ filtering problems. The H^∞ square-root algorithms involve propagating the *indefinite* square-root of the quantities of interest and have the property that the appropriate inertia of these quantities is preserved. For systems that are constant, or whose time-variation is structured in a certain way, the Chandraskhar recursions allow a reduction in the computational effort per iteration from $O(n^3)$ to $O(n^2)$, where n is the number of states. The H^∞ square-root and Chandrasekhar recursions both have the interesting feature that one does not need to explicitly check for the positivity conditions required of the H^∞ filters. These conditions are built into the algorithms themselves so that an H^∞ estimator of the desired level exists if, and only if, the algorithms can be executed.

1 Introduction

Classical results in linear least-squares estimation and Kalman filtering are based on an H^2 criterion and require apriori knowledge of the statistical properties of the noise signals. In some applications, however, one is faced with model uncertainties and lack of statistical information on the exogenous signals, which has led to an increasing interest in minimax estimation, with the belief that the resulting so-called H^∞ algorithms will be more robust and less sensitive to parameter variations (see *e.g.* [1,2,3,4]). Interestingly enough, the H^∞ filters obtained in this fashion involve propagating a Riccati variable and bear a striking resemblance to the classical Kalman filter. Nevertheless there are enough key differences that ingenious new methods seem to have been necessary to tackle H^∞ problems.

In [5] we have shown that by introducing variables in an indefinite metric (Krein) space, rather than in the Hilbert spaces common in conventional stochastic theory, the H^∞ filters could be obtained from the theory of Kalman filters in Krein space. The point is that although Hilbert spaces and Krein spaces share many characteristics, they differ in special ways that turn out to mark the differences between the LQG or H^2 theories and the more recent H^∞ theories.

[†]This research was supported by the Advanced Research Projects Agency of the Department of Defense monitored by the Air Force Office of Scientific Research under Contract F49620-93-1-0085.

[‡]Dr. Sayed is currently an Assistant Professor in the Dept. of Electrical and computer Engineering, University of California, Santa Barbara, CA 93106

The main point of our approach is that, apart from rather more transparent derivations of existing results, it shows a way to apply to the H^∞ setting many of the results developed for Kalman filtering over the last three decades. In particular, for several reasons, certain square-root arrays are now more often used to implement the conventional Kalman filter. Furthermore for constant systems, or in fact for systems where the time-variation is structured in a certain way, the Riccati recursions and the square-root recursions, both of which take $O(n^3)$ elementary computations (flops) per iteration (where n is the dimension of the state-space), can be replaced by the more efficient Chandrasekhar recursions, which require only $O(n^2)$ flops per iteration [6,7].

One immediate fall-out of our appraoch is that it allows us to generalize these square-root arrays and Chandrasekhar recursions to the H^∞ setting. Both these algorithms involve propagating (indefinite) square-roots of the quantities of interest and guarantee that the proper inertia of these quantities is preserved. Furthermore the condition required for the existence of the H^∞ filters is built into the algorithms: if the algorithms can be carried out, then an H^∞ filter of the desired level exists, and if they cannot be executed then such H^∞ filters do not exist. This can be a significant simplification of the existing algorithms.

A brief remark on the notation used in this paper. To avoid confusion between the various *gain vectors* used in this paper, we shall employ the following convention: $K_{p,i}$ will denote the gain vector in the usual Krein space (or Hilbert space) Kalman filter, $K_{f,i}$ the gain vector in the *filtered* form of the Krein space Kalman filter, and $K_{s,i}$ and $K_{a,i}$ will denote the gain vectors in the H^∞ aposteriori and apriori filters, respectively.

2 Inertia Conditions for H^∞ Filtering

Consider a state-space model of the form

$$\begin{cases} x_{i+1} = F_i x_i + G_i u_i, & x_0 \\ y_i = H_i x_i + v_i & i \geq 0 \end{cases} \qquad (2.1)$$

where x_0, $\{u_i\}$, and $\{v_i\}$ are *unknown* quantities and y_i is the measured output. Note that we shall make no assumption on the nature of the disturbances (such as uncorrelated, normally distributed, etc.) Let z_i be linearly related to the state x_i via a given matrix L_i, viz.,

$$z_i = L_i x_i.$$

We shall be interested in the following two cases. Let $\check{z}_{i|i} = \mathcal{F}_f(y_0, y_1, \ldots, y_i)$ denote the estimate of z_i given observations $\{y_j\}$ from time 0 up to and including time

i, and $\check{z}_i = \mathcal{F}_p(y_0, y_1, \ldots, y_{i-1})$ denote the estimate of z_i given observations $\{y_j\}$ from time 0 to time $i-1$. We then have the following two estimation errors: the *filtered* error $e_{f,i} = \check{z}_{i|i} - L_i x_i$, and the *predicted* error $e_{p,i} = \check{z}_i - L_i x_i$.

Let $T_{f,i}$ denote the transfer operator that maps the unknown disturbances $\{x_0, \{u_j\}_{j=0}^{i}, \{v_j\}_{j=0}^{i}\}$ to the filtered errors $\{e_{f,j}\}_{j=0}^{i}$. Likewise, let $T_{p,i}$ denote the transfer operator from $\{x_0, \{u_j\}_{j=0}^{i-1}, \{v_j\}_{j=0}^{i-1}\}$ to the predicted errors $\{e_{p,j}\}_{j=0}^{i}$. The H^∞ estimation problem can now be stated as follows.

Problem 2.1 (Sub-optimal H^∞ Problem) *Given scalars $\gamma_f > 0$ and $\gamma_p > 0$, find H^∞-suboptimal estimation strategies $\check{z}_{i|i} = \mathcal{F}_f(y_0, y_1, \ldots, y_i)$ and $\check{z}_i = \mathcal{F}_p(y_0, y_1, \ldots, y_{i-1})$ that respectively achieve $\| T_f \|_\infty < \gamma_f$ and $\| T_p \|_\infty < \gamma_p$. In other words*

$$\sup_{x_0, u\in h_2, v\in h_2} \frac{\sum_{j=0}^{i} |e_{f,j}|^2}{\| x_0 \|_{\Pi_0^{-1}} + \sum_{j=0}^{i} |u_j|^2 + \sum_{j=0}^{i} |v_j|^2} < \gamma_f^2 \tag{2.2}$$

and

$$\sup_{x_0, u\in h_2, v\in h_2} \frac{\sum_{j=0}^{i} |e_{p,j}|^2}{\| x_0 \|_{\Pi_0^{-1}} + \sum_{j=0}^{i-1} |u_j|^2 + \sum_{j=0}^{i-1} |v_j|^2} < \gamma_p^2 \tag{2.3}$$

where Π_0 is a positive definite weighting matrix that reflects apriori knowledge as to how close x_0 is to the initial guess $\check{x}_0$.

We now state the solution to Problem 2.1 [3,5].

Theorem 2.1 (The H^∞ Aposteriori Filter) *For a given $\gamma > 0$, if the F_i are nonsingular then an estimator with $\|T_{f,i}\|_\infty < \gamma$ exists if, and only if,*

$$P_j^{-1} + H_j^* H_j - \gamma^{-2} L_j^* L_j > 0, \qquad j = 0, \ldots, i, \tag{2.4}$$

where $P_0 = \Pi_0$ and P_j satisfies the Riccati recursion

$$P_{j+1} = F_j P_j F_j^* + G_j G_j^* - F_j P_j \begin{bmatrix} H_j^* & L_j^* \end{bmatrix} R_{e,j}^{-1} \begin{bmatrix} H_j \\ L_j \end{bmatrix} P_j F_j^*, \tag{2.5}$$

with $R_{e,j} = \begin{bmatrix} I & 0 \\ 0 & -\gamma^2 I \end{bmatrix} + \begin{bmatrix} H_j \\ L_j \end{bmatrix} P_j \begin{bmatrix} H_j^* & L_j^* \end{bmatrix}$. *If this is the case, then one possible H^∞ filter with level γ is given by $\check{z}_{j|j} = L_j \hat{x}_{j|j}$, where $\hat{x}_{j|j}$ is recursively computed as*

$$\hat{x}_{j+1|j+1} = F_j \hat{x}_{j|j} + K_{s,j}(y_{j+1} - H_{j+1} F_j \hat{x}_{j|j}), \quad \hat{x}_{-1|-1}, \tag{2.6}$$

and $K_{s,j} = P_{j+1} H_{j+1}^* (I + H_{j+1} P_{j+1} H_{j+1}^*)^{-1}$.

Theorem 2.2 (The H^∞ Apriori Filter) *For a given $\gamma > 0$, if the F_i are nonsingular then an estimator with $\|T_{p,i}\|_\infty < \gamma$ exists if, and only if,*

$$\tilde{P}_j^{-1} = P_j^{-1} - \gamma^{-2} L_j^* L_j > 0, \qquad j = 0, \ldots, i, \tag{2.7}$$

where P_j is the same as in Theorem 2.1. If this is the case, then one possible H^∞ filter with level γ is given by $\check{z}_j = L_j \hat{x}_j$, where

$$\hat{x}_{j+1} = F_j \hat{x}_j + K_{a,j}(y_j - H_j \hat{x}_j), \quad \hat{x}_0, \tag{2.8}$$

and where $K_{a,j} = F_j \tilde{P}_j H_j^* (I + H_j \tilde{P}_j H_j^*)^{-1}$.

Remark: These look very much like a Kalman filter solution, except that the Riccati recursion differs from that of the Kalman filter, since

- we have indefinite *covariance* matrices, $R_{e,j}$.
- the L_i (of the quantity to be estimated) enters the Riccati equation.
- we have an additional condition, (2.4), that must be satisfied for the filter to exist; in the Kalman filter problem the L_i would not appear, and the P_i would be positive definite, so that (2.4) is immediate.

Despite these differences, we have shown that the filters of Theorems 2.1 and 2.2 can in fact be obtained as certain Kalman filters, not in an H^2 (Hilbert) space, but in a certain indefinite vector space, called a Krein space [5]. The indefinite covariances and the appearance of L_i in the Riccati equation are easily explained in this framework. The additional conditions (2.4) and (2.7) arise from the fact that in Krein space, unlike as in the usual Hilbert space context, quadratic forms need not always have minima or maxima, unless certain additional conditions are met. Moreover, this approach provides simpler and more general alternatives to the tests (2.4) and (2.7).

Lemma 2.1 (New Existence Tests) *The conditon (2.4) can be replaced by the condition that all leading submatrices of*

$$R_j = \begin{bmatrix} I & 0 \\ 0 & -\gamma^2 I \end{bmatrix} \quad \text{and}$$

$$R_{e,j} = \begin{bmatrix} I & 0 \\ 0 & -\gamma^2 I \end{bmatrix} + \begin{bmatrix} H_j \\ L_j \end{bmatrix} P_j \begin{bmatrix} H_j^* & L_j^* \end{bmatrix}$$

have the same inertia. Likewise the condition (2.7) can be replaced by the condition that all leading submatrices of

$$\tilde{R}_j = \begin{bmatrix} -\gamma^2 I & 0 \\ 0 & I \end{bmatrix} \quad \text{and}$$

$$\tilde{R}_{e,j} = \begin{bmatrix} -\gamma^2 I & 0 \\ 0 & I \end{bmatrix} + \begin{bmatrix} L_j \\ H_j \end{bmatrix} P_j \begin{bmatrix} L_j^* & H_j^* \end{bmatrix}$$

have the same inertia. The nonsingularity of the $\{F_j\}$ is no longer required here and the size of the matrices involved is generally smaller than in (2.4) and (2.7).

By the inertia of a Hermitian matrix, we mean the number of its positive, negative and zero eigenvalues. A simple way of calculating the inertia of a strongly regular Hermitian matrix R (*i.e.* one whose leading minors are all nonzero), is by computing its LDU decomposition

$$R = LDL^*,$$

where L is a lower triangular matrix with unit diagonal and D is a diagonal matrix: the number of positive and negative elements of D give the number of positive and negative eigenvalues of R and hence the inertia. Therefore to check whether all leading submatrices of two strongly regular Hermitian matrices R_1 and R_2 have the same inertia, we can compute the LDU decompositions $R_1 = L_1 D_1 L_1^*$ and $R_2 = L_2 D_2 L_2^*$ and check whether the corresponding diagonal entries of D_1 and D_2 have the same sign.

The condition of Lemma 2.1 is easier to check than that of (2.4) since it involves $R_{e,j}$ which is used to propagate the Riccati recursion, whereas we must invert P_j at each step to check (2.4). Moreover, in

Lemma 2.1 we must check for the inertia properties of a $(p_1+p_2)\times(p_1+p_2)$ matrix (where p_1 and p_2 are the number of outputs and the number of linear combinations of the states given to estimate) whereas in (2.4) we must check for the positivity of a $n\times n$ matrix where n is the number of states. In most applications p_1+p_2 will be less than n.

In fact, the need for checking separate conditions as in (2.4) or Lemma 2.1 can be completely avoided by going to a new square-root form of the H^∞ filtering algorithm - these conditions are built into the the square-root recursions. The H^∞ square-root algorithms are very naturally suggested by the close connection we make between H^2 (Kalman) filtering and H^∞ filtering.

Due to lack of space, we shall not go into the details of the connection between Krein space Kalman filtering and H^∞ filtering here (the interested reader is referred to [5,9]). We shall just mention that the H^∞ filters of Theorems 2.1 and 2.2 can be derived from the Krein space Kalman filter corresponding to the following state-space model

$$\left\{\begin{array}{rcl} \mathbf{x_{i+1}} &=& F_i\mathbf{x_i}+G_i\mathbf{u_i} \\ \begin{bmatrix}\mathbf{y_i}\\ \mathbf{z_i}\end{bmatrix} &=& \begin{bmatrix}H_i\\ L_i\end{bmatrix}\mathbf{x_i}+\mathbf{w_i}\end{array}\right. \tag{2.9}$$

where the $\{\mathbf{u_i},\mathbf{v_i},\mathbf{x_0}\}$ are elements in a Krein space $\mathcal{K}$, such that

$$<\begin{bmatrix}\mathbf{u_i}\\ \mathbf{w_i}\\ \mathbf{x_0}\end{bmatrix}\begin{bmatrix}\mathbf{u_j}\\ \mathbf{w_j}\\ \mathbf{x_0}\end{bmatrix}>_{\mathcal{K}}=\begin{bmatrix} I\delta_{ij} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \begin{bmatrix} I & \mathbf{0}\\ \mathbf{0} & -\gamma_f^2 I\end{bmatrix}\delta_{ij} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \Pi_0\end{bmatrix} \tag{2.10}$$

and where $<.,.>_{\mathcal{K}}$ is the Krein space inner product. Note that we need to consider a Krein space because

$$R_j=\begin{bmatrix} I & \mathbf{0}\\ \mathbf{0} & -\gamma^2 I\end{bmatrix}\quad\text{is indefinite.}$$

3 Square-Root Algorithms

3.1 H^2 Square-Root Arrays

In the conventional Kalman filter

$$\hat{x}_{j+1}=F_j\hat{x}_j+K_{p,j}(y_j-H_j\hat{x}_j),\qquad \hat{x}_0=0,$$

the gain vector $K_{p,j}$ is updated using a Riccati recursion as follows

$$K_{p,j}=F_jP_jH_j^*R_{e,j}^{-1}\quad,\quad R_{e,j}=R_j+H_jP_jH_j^* \tag{3.11}$$

and

$$P_{j+1}=F_jP_jF_j^*-K_{p,j}R_{e,j}K_{p,j}^*+G_jQ_jG_j^*,\quad P_0=\Pi_0. \tag{3.12}$$

The matrix P_j appearing in this Riccati recursion has the physical meaning of being the variance of the state prediction error, $\tilde{x}_j=x_j-\hat{x}_j$, and therefore has to be positive (semi)definite. Round-off errors can cause a loss of positive-definiteness, so that for this, and other reasons (reduced dynamic range, better conditioning, stabler algorithms, etc.) attention has moved in the Kalman filtering community to the so-called square-root array algorithms that propagate square-root factors of P_j, *i.e.* a matrix, $P_j^{\frac12}$ say, such that

$$P_j=P_j^{\frac12}(P_j^{\frac12})^*=P_j^{\frac12}P_j^{\frac{*}{2}}.$$

Now the following algorithm can be introduced. Apply any *orthogonal* transformation, say Θ_j, that triangularizes the pre-array shown below

$$\begin{bmatrix} R_i^{\frac12} & H_jP_j^{\frac12} & 0\\ 0 & F_jP_j^{\frac12} & G_jQ_j^{\frac12}\end{bmatrix}\Theta_j=\begin{bmatrix} X & 0 & 0\\ Y & Z & 0\end{bmatrix}.$$

The resulting post-array entries can be identified as

$$Z=P_{j+1}^{\frac12}\quad,\quad X=R_{e,j}^{\frac12}$$

and

$$Y=F_jP_jH_j^*R_{e,j}^{-\frac{*}{2}}=K_{p,j}R_{e,j}^{\frac12}=\bar{K}_{p,j},\quad\text{say.}$$

This can be checked by taking *squares* of both sides (and using the orthogonality of Θ_j) to get

$$\begin{aligned} XX^* &= R_j+H_jP_jH_j^*=R_{e,j}\\ YX^* &= F_jP_jH_j^*\\ ZZ^* &= F_jP_jF_j^*+G_jQ_jG_j^*-YY^*\\ &= F_jP_jF_j^*+G_jQ_jG_j^*-F_jP_jH_j^*R_{e,j}^{-1}H_jP_jF_j^*\\ &= P_{j+1}\end{aligned}$$

Note that the quantities necessary to update the square-root array and to calculate the state estimates may all be found from the triangularized array.

Theorem 3.1 **(H^2 Square-Root Algorithm)** *Quantities of interest in the conventional Kalman filter*

$$\hat{x}_{j+1}=F_j\hat{x}_j+K_{p,j}(y_j-H_j\hat{x}_j),\qquad \hat{x}_0=0,$$

can be updated as follows:

$$\begin{bmatrix} R_j^{\frac12} & H_jP_j^{\frac12} & 0\\ 0 & F_jP_j^{\frac12} & G_jQ_j^{\frac12}\end{bmatrix}\Theta_j=\begin{bmatrix} R_{e,j}^{\frac12} & 0 & 0\\ K_{p,j}R_{e,j}^{\frac12} & P_{j+1}^{\frac12} & 0\end{bmatrix}, \tag{3.13}$$

where Θ_j can be chosen to be any unitary matrix that triangularizes the above pre-array. The algorithm is initialized with $P_0=\Pi_0$.

In practice Θ_j is implemented via a sequence of elementary unitary (Givens) rotations or (Householder) reflections. We also quote the filtered form of the square-root array algorithm which can be verified similarly.

Theorem 3.2 **(Filtered Form)** *Quantities of interest in the filtered form of the conventional Kalman filter*

$$\hat{x}_{j+1|j+1}=F_j\hat{x}_{j|j}+K_{f,j}(y_{j+1}-H_{j+1}F_j\hat{x}_{j|j}),\quad \hat{x}_{-1|-1}=0,$$

can be updated as follows

$$\begin{bmatrix} R_j^{\frac12} & H_jP_j^{\frac12}\\ 0 & P_j^{\frac12}\end{bmatrix}\Theta_j^{(1)}=\begin{bmatrix} R_{e,j}^{\frac12} & 0\\ K_{f,j-1}R_{e,j}^{\frac12} & P_{j|j}^{\frac12}\end{bmatrix}, \tag{3.14}$$

$$\begin{bmatrix} F_jP_{j|j}^{\frac12} & G_jQ_j^{\frac12}\end{bmatrix}\Theta_j^{(2)}=\begin{bmatrix} P_{j+1}^{\frac12} & 0\end{bmatrix} \tag{3.15}$$

where $\Theta_j^{(1)}$ and $\Theta_j^{(2)}$ can be chosen to be any unitary matrices that triangularize the above pre-arrays. The algorithm is initialized with $P_0=\Pi_0$.

3.2 H^∞ Square-Root Arrays

In the H^∞ aposteriori filtering problem with level γ, the R_j and $R_{e,j}$ (of the Krein space model (2.9)) are no longer positive-definite, but indefinite. Therefore we must consider the indefinite square roots of these quantities, namely,

$$R_j^{\frac{1}{2}} = \begin{bmatrix} I & 0 \\ 0 & \gamma I \end{bmatrix} \quad \text{and} \quad R_{e,j}^{\frac{1}{2}},$$

where

$$R_j^{\frac{1}{2}} J R_j^{\frac{*}{2}} = \begin{bmatrix} I & 0 \\ 0 & -\gamma^2 I \end{bmatrix} \quad \text{and} \quad R_{e,j}^{\frac{1}{2}} J R_{e,j}^{\frac{*}{2}} = R_{e,j}$$

with $J = (I \oplus -I)$. In the square-root form of the H^∞ aposteriori filtering problem with level γ, it is therefore plausible to begin with the pre-array

$$\begin{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \gamma I \end{bmatrix} & \begin{bmatrix} H_j \\ L_j \end{bmatrix} P_j^{\frac{1}{2}} & 0 \\ 0 & F_j P_j^{\frac{1}{2}} & G_j \end{bmatrix} \tag{3.16}$$

where as before $P_j^{\frac{1}{2}} P_j^{\frac{*}{2}} = P_j$ (since P_j can be shown to be positive). We then attempt to triangularize this pre-array with a J-unitary matrix Θ_j (*i.e.* one for which $\Theta_j J \Theta_j^* = J$), where now

$$J = I \oplus -I \oplus I \oplus I. \tag{3.17}$$

However, triangularization via J-unitary matrices is not always possible and requires certain inertia properties. The precise statement follows.

Lemma 3.1 (*J*-unitary Triangularization)
Let A and B be arbitrary $n \times n$ and $n \times m$ matrices, respectively, and suppose $J = S_1 \oplus S_2$, where S_1 and S_2 are $n \times n$ and $m \times m$ signature matrices. Then $\begin{bmatrix} A & B \end{bmatrix}$ can be triangularized by a J-unitary transformation Θ as

$$\begin{bmatrix} A & B \end{bmatrix} \Theta = \begin{bmatrix} L & 0 \end{bmatrix}$$

with L lower triangular, if and only if, all leading submatrices of

$$S_1 \quad \textit{and of} \quad A S_1 A^* + B S_2 B^*$$

have the same inertia.

Thus from Lemma 3.1, triangularization of (3.16) via a J-unitary transformation is possible if, and only, if all leading submatrices of

$$\begin{bmatrix} I & 0 \\ 0 & \gamma I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \gamma I \end{bmatrix}^* + \begin{bmatrix} H_j \\ L_j \end{bmatrix} P_j \begin{bmatrix} H_j^* & L_j^* \end{bmatrix} = R_{e,j} \quad \text{and} \quad \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix},$$

have the same inertia. But this is precisely the inertia condition given for the existence of H^∞ aposteriori filters in Lemma 2.1.

We are thus led to the following square-root array algorithm for the H^∞ aposteriori filtering problem. The proof essentially follows from *squaring* both sides and comparing terms. For more details see [9].

Theorem 3.3 (H^∞ **Square-Root Algorithm**)
The H^∞ aposteriori filtering problem with level γ has a solution if, and only if, for all $j = 0, \ldots, i$ there exist J-unitary (with $J = I \oplus -I \oplus I$) matrices $\Theta_j^{(1)}$ such that

$$\begin{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \gamma I \end{bmatrix} & \begin{bmatrix} H_j \\ L_j \end{bmatrix} P_j^{\frac{1}{2}} \\ 0 & P_j^{\frac{1}{2}} \end{bmatrix} \Theta_j^{(1)} = \begin{bmatrix} R_{e,j}^{\frac{1}{2}} & 0 \\ K_{f,j-1} R_{e,j}^{\frac{1}{2}} & P_{j|j}^{\frac{1}{2}} \end{bmatrix}, \tag{3.18}$$

$$\begin{bmatrix} F_j P_{j|j}^{\frac{1}{2}} & G_j Q_j^{\frac{1}{2}} \end{bmatrix} \Theta_j^{(2)} = \begin{bmatrix} P_{j+1}^{\frac{1}{2}} & 0 \end{bmatrix} \tag{3.19}$$

with $R_{e,j}^{\frac{1}{2}}$ and $P_{j+1}^{\frac{1}{2}}$ lower block triangular, and with $\Theta_j^{(2)}$ unitary. The gain vector $K_{s,j}$ needed to update the estimates in

$$\hat{x}_{j+1|j+1} = F_j \hat{x}_{j|j} + K_{s,j}(y_{j+1} - H_{j+1} F_j \hat{x}_{j|j}), \qquad \hat{x}_{-1|-1} = 0,$$

is equal to $K_{s,j} = \bar{K}_{s,j}(I + H_{j+1} P_{j+1} H_{j+1}^*)^{-\frac{1}{2}}$, where $\bar{K}_{s,j}$ is given by the first block column of $\bar{K}_{f,j} = K_{f,j} R_{e,j+1}^{\frac{1}{2}}$, and $(I + H_{j+1} P_{j+1} H_{j+1}^*)^{\frac{1}{2}}$ is given by the $(1,1)$ block entry of $R_{e,j+1}^{\frac{1}{2}}$. The algorithm is initialized with $P_0 = \Pi_0$.

Note, as before, that the quantities necessary to update the square-root array and to calculate the state estimates may all be found from the triangularized post-array. The reason for choosing sub-blocks of $\bar{K}_{f,j}$ and $R_{e,j+1}^{\frac{1}{2}}$ follows from the fact that we can only use y_i, whereas the output equation for the Krein state-space model (2.9) has both a y_i and a z_i component.

In conventional Kalman filtering, square-root arrays are preferred since the positive-definiteness of the matrices is guaranteed, and since the Θ_j are unitary, which improves the numerical stability of the algorithm. In the H^∞ setting the square-root arrays guarantee that the various matrices have their appropriate inertia (see Lemma 2.1); however, the Θ_j are no longer unitary but J-unitary. Therefore the numerical aspects need further investigation.

An interesting aspect of Theorem 3.3 is that there is no need to explicitly check for the positivity condition (2.4). This condition is built into the algorithm itself: if the algorithm can be performed an H^∞ estimator of the desired level exists, and if it cannot be performed such an estimator does not exist.

Comparing Theorem 3.2 with Theorem 3.3 reveals the formal similarities between the H^2 and H^∞ square-root array algorithms. The H^∞ algorithms are essentially the Krein space generalization of the H^2 algorithms (which for example explains why unitary matrices are replaced by J-unitary matrices), and it is this approach that is used to derive Theorem 3.3 and similar results.

We close this section by giving the square-root version of the H^∞ apriori filtering algorithm, which can be derived in a similar fashion. Note that the major difference with the aposteriori square-root algorithm of Theorem 3.3 is the change in the order of $\{H_j, L_j\}$ and $\{I, \gamma I\}$ in the various arrays. The reader at this point may want to note this order reversal in Lemma 2.1 as well.

Theorem 3.4 (Apriori Case) *The H^∞ apriori filtering problem with level γ has a solution if, and only if,* *for all $j = 0, \ldots, i$ there exist J-unitary matrices (with J given by (3.17)) such that*

$$\begin{bmatrix} \begin{bmatrix} \gamma I & 0 \\ 0 & I \end{bmatrix} & \begin{bmatrix} L_j \\ H_j \end{bmatrix} P_j^{\frac{1}{2}} & 0 \\ 0 & F_j P_j^{\frac{1}{2}} & G_j \end{bmatrix} \Theta_j = \begin{bmatrix} \tilde{R}_{e,j}^{\frac{1}{2}} & 0 & 0 \\ \bar{K}_{p,j} & P_{j+1}^{\frac{1}{2}} & 0 \end{bmatrix} \quad (3.20)$$

with $\tilde{R}_{e,j}^{\frac{1}{2}}$ and $P_{j+1}^{\frac{1}{2}}$ lower block triangular. The gain vector $K_{a,i}$ needed to update the estimates in

$$\hat{x}_{j+1} = F_j \hat{x}_j + K_{a,j}(y_j - F_j \hat{x}_j), \qquad \hat{x}_0 = 0,$$

is equal to $K_{a,j} = \bar{K}_{a,j}(I + H_j \tilde{P}_j H_j^*)^{-\frac{1}{2}}$, where $\bar{K}_{a,j}$ is given by the second block column of $\bar{K}_{p,j}$, and $(I + H_j \tilde{P}_j H_j^*)^{\frac{1}{2}}$ is given by the $(2,2)$ block entry of $\tilde{R}_{e,j}^{\frac{1}{2}}$. The algorithm is initialized with $P_0 = \Pi_0$.

4 Chandrasekhar Recursions

4.1 H^2 Chandrasekhar Recursions

In what follows we shall assume a *constant* state-space model of the form

$$\begin{cases} x_{j+1} = F x_j + G u_j, & x_0 \\ y_j = H x_j + v_j \end{cases} \quad (4.21)$$

In the H^2 case, we shall also assume that the covariances of the $\{u_j, v_j\}$ are constant, *i.e.* $Q_j = Q$ and $R_j = R$, for all j. As before, we are interested in obtaining estimates of the states, denoted by $\hat{x}_j$, using the observations $\{y_k\}_{k=0}^{i-1}$.

Under the aforementioned assumptions, it is possible to choose the matrix Π_0 such that

$$P_{j+1} - P_j = M_j S M_j, \qquad \forall j, \quad (4.22)$$

where M_j is a $n \times d$ matrix (often $d \ll n$) and S is a $d \times d$ signature matrix (*i.e.* a diagonal matrix with $+1$ and -1 on the diagonal). Thus for time-invariant state-space models, $P_{j+1} - P_j$ has rank d for all j and in addition has constant inertia. If this is true, note that propagating the (smaller) matrices M_j is equivalent to propagating the P_j. This what is done by the Chandrasekhar recursions (see [8,6], App. II).

In the conventional Chandrasekhar recursions, one triangularizes the following pre-array with a J-unitary matrix Θ_j

$$\begin{bmatrix} R_{e,j}^{\frac{1}{2}} & H M_j \\ \bar{K}_{p,j} & F M_j \end{bmatrix} \Theta_j = \begin{bmatrix} X & 0 \\ Y & Z \end{bmatrix}$$

where $R_{e,j}^{\frac{1}{2}} R_{e,j}^{\frac{*}{2}} = R_{e,j} = R + H P_j H^*$ and $\bar{K}_{p,j} = K_{p,j} R_{e,j}^{\frac{1}{2}}$, and where J is given by $\begin{bmatrix} I & \\ & S \end{bmatrix}$. Squaring both sides of the above equality and using (4.22) and the J-orthogonality of Θ_j allows us to identify the elements in the post array as follows.

$$\begin{aligned} XX^* &= R_{e,j}^{\frac{1}{2}} R_{e,j}^{\frac{*}{2}} + H M_j S M_j^* H^* = R_{e,j+1} \\ YX^* &= \bar{K}_{p,j} R_{e,j}^{\frac{*}{2}} + F M_j S M_j^* H^* = F P_{j+1} H^* \\ ZSZ^* &= \bar{K}_{p,j} \bar{K}_{p,j}^* + F M_j S M_j^* - YY^* \\ &= P_{j+2} - P_{j+1} \end{aligned}$$

Therefore we can identify

$$X = R_{e,j+1}^{\frac{1}{2}}, \qquad Y = \bar{K}_{p,j+1} \qquad \text{and} \qquad Z = M_{j+1}.$$

We have thus verified the following result.

Theorem 4.1 (**H^2 Chandrasekhar Recursions**) *Quantities of interest in the conventional Kalman filter*

$$\hat{x}_{j+1} = F_j \hat{x}_j + K_{p,j}(y_j - H_j \hat{x}_j), \qquad \hat{x}_0 = 0,$$

can be updated using

$$\begin{bmatrix} R_{e,j}^{\frac{1}{2}} & H M_j \\ \bar{K}_{p,j} & F M_j \end{bmatrix} \Theta_j = \begin{bmatrix} R_{e,j+1}^{\frac{1}{2}} & 0 \\ \bar{K}_{p,j+1} & M_{j+1} \end{bmatrix}. \quad (4.23)$$

where Θ_j is any J-unitary matrix (with $J = I \oplus S$) that triangularizes the above pre-array. The algorithm is initialized with $R_{e,0} = R + H \Pi_0 H^*$, $\bar{K}_{p,0} = F \Pi_0 H^* R_{e,0}^{\frac{1}{2}}$, and

$$P_1 - \Pi_0 = F \Pi_0 F^* + GQG^* - K_{p,0} R_{e,0} K_{p,0}^* - \Pi_0 = M_0 S M_0^*.$$

Thus, once more, the quantities necessary to update the arrays and to calculate the state estimates are all found from the triangularized post array.

In the filtered case, one noramally computes the gain vector using the equation $K_{f,i} = F^{-1} K_{p,i+1}$. We shall see this in the H^∞ algorithms of the next section.

4.2 H^∞ Chandrasekhar Recursions

Our earlier results suggest that in the H^∞ aposteriori filtering problem with level γ, we need to start with the pre-array

$$\begin{bmatrix} R_{e,j}^{\frac{1}{2}} & \begin{bmatrix} H \\ L \end{bmatrix} M_j \\ \bar{K}_{p,j} & F M_j \end{bmatrix}, \quad (4.24)$$

where

$$R_{e,j}^{\frac{1}{2}} \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix} R_{e,j}^{\frac{*}{2}} = R_{e,j} = R_j + \begin{bmatrix} H \\ L \end{bmatrix} P_j \begin{bmatrix} H^* & L^* \end{bmatrix}, \quad (4.25)$$

$\bar{K}_{p,j} = K_{p,j} R_{e,j}^{\frac{1}{2}}$ and $P_{j+1} - P_j = M_j S M_j^*$. The next step is to triangularize the pre-array (4.24) using a J-unitary matrix, where

$$J = (I \oplus -I \oplus S). \quad (4.26)$$

The condition for the existence of such a triangularization (as given by Lemma 3.1) is precisely the condition for the existence of the H^∞ aposteriori filters. We may thus prove the following result using the method mentioned for deriving the conventional recursions. For more details see [9].

Theorem 4.2 (**H^∞ Chandrasekhar Recursions**) *The H^∞ aposteriori filtering problem with level γ has a solution if, and only if, all leading submatrices of*

$$R = \begin{bmatrix} I & 0 \\ 0 & -\gamma^2 I \end{bmatrix} \quad \text{and} \quad R_{e,0} = R + \begin{bmatrix} H \\ L \end{bmatrix} \Pi_0 \begin{bmatrix} H^* & L^* \end{bmatrix}$$

have the same inertia, and if for all $j = 0, \ldots, i$ there exist J-unitary matrices Θ_j (where J is given by (4.26)) such that

$$\begin{bmatrix} R_{e,j}^{\frac{1}{2}} & \begin{bmatrix} H \\ L \end{bmatrix} M_j \\ \bar{K}_{p,j} & FM_j \end{bmatrix} \Theta_j = \begin{bmatrix} R_{e,j+1}^{\frac{1}{2}} & 0 \\ \bar{K}_{p,j+1} & M_{j+1} \end{bmatrix} \tag{4.27}$$

with $R_{e,j}^{\frac{1}{2}}$ and $R_{e,j+1}^{\frac{1}{2}}$ lower block triangular. The gain vector $\bar{K}_{s,j}$ needed to update the estimates in

$$\hat{x}_{j+1|j+1} = F_j \hat{x}_{j|j} + K_{s,j}(y_{j+1} - H_{j+1} F_j \hat{x}_{j|j}), \quad \hat{x}_{-1|-1} = 0,$$

is equal to $K_{s,j} = \bar{K}_{s,j}(I + H_{j+1}P_{j+1}H_{j+1}^*)^{-\frac{1}{2}}$, where $\bar{K}_{s,j}$ is given by the first block column of $F^{-1}\bar{K}_{p,j+1}$, and $(I + H_{j+1}P_{j+1}H_{j+1}^*)^{\frac{1}{2}}$ is given by the $(1,1)$ block entry of $R_{e,j+1}^{\frac{1}{2}}$.

Note that compared to the square-root formulas, the size of the pre-array in the Chandrasekhar recursions has been reduced from $(p_1 + p_2 + n) \times (p_1 + p_2 + n + m)$ to $(p_1 + p_2 + n) \times (p_1 + p_2 + d)$ where m, p_1 and p_2 are the dimensions of the driving disturbance, output and states to be estimated, respectively, and where n is the number of the states. Thus the number of operations for each iteration has been reduced from $O(n^3)$ to $O(n^2 d)$ with d typically much less than n.

As in the square-root case, the Chandrasekhar recursions do not require explicitly checking the positivity condition (2.4). We can also obtain H^∞ apriori Chandrasekhar recursions in a similar fashion. The result is given below. Note, once more, the change in the order of $\{H, L\}$ and $\{I, \gamma I\}$ in the arrays, and the absence of F^{-1} to compute the gain.

Theorem 4.3 (Apriori Case) *The H^∞ apriori filtering problem with level γ has a solution if, and only if, all leading submatrices of*

$$\tilde{R} = \begin{bmatrix} -\gamma^2 I & 0 \\ 0 & I \end{bmatrix} \quad \text{and} \quad \tilde{R}_{e,0} = \tilde{R} + \begin{bmatrix} L \\ H \end{bmatrix} \Pi_0 \begin{bmatrix} L^* & H^* \end{bmatrix}$$

have the same inertia, and if for all $j = 0, \ldots, i$ there exist J-unitary matrices Θ_j (where J is given by (??)) such that

$$\begin{bmatrix} \tilde{R}_{e,j}^{\frac{1}{2}} & \begin{bmatrix} L \\ H \end{bmatrix} M_j \\ \bar{K}_{p,j} & FM_j \end{bmatrix} \Theta_j = \begin{bmatrix} \tilde{R}_{e,j+1}^{\frac{1}{2}} & 0 \\ \bar{K}_{p,j+1} & M_{j+1} \end{bmatrix} \tag{4.28}$$

with $\tilde{R}_{e,j}^{\frac{1}{2}}$ and $\tilde{R}_{e,j+1}^{\frac{1}{2}}$ lower block triangular. The gain vector $K_{a,i}$ needed to update the estimates in

$$\hat{x}_{j+1} = F_j \hat{x}_j + K_{a,j}(y_j - F_j \hat{x}_j), \qquad \hat{x}_0 = 0,$$

is equal to $K_{a,j} = \bar{K}_{a,j}(I + H_j \tilde{P}_j H_j^*)^{-\frac{1}{2}}$, where $\bar{K}_{a,j}$ is given by the second block column of $\bar{K}_{p,j}$, and $(I + H_j \tilde{P}_j H_j^*)^{\frac{1}{2}}$ is given by the $(2,2)$ block entry of $\tilde{R}_{e,j}^{\frac{1}{2}}$. The algorithm is initialized with $R_{e,0}$, $\bar{K}_{p,0} = F\Pi_0 [\, L^* \;\; H^* \,]$ and

$$P_1 - \Pi_0 = F\Pi_0 F^* + GQG^* - \bar{K}_{p,0} S \bar{K}_{p,0}^* - \Pi_0 = M_0 S M_0^*.$$

5 Conclusion

We have obtained square-root array algorithms and Chandrasekhar recursions for the H^∞ aposteriori and apriori filtering problems. These have the important prrperty that the conditions for the existence of the H^∞ filters are built into the algorithms, so that filter solutions will exist if,and only if, the algorithms can be executed.

The conventional square-root arrays and Chandrasekhar recursions are preferred because of their numerical stability (in the case of square-root arrays) and their reduced computational complexity (in the case of the Chandrasekhar recursions). Since the H^∞ square-root arrays and Chandrasekhar recursions are the direct analogs of their conventional counterparts, they may be more attractive for numerical implementation of H^∞ filters. However, since J-unitary rather than unitary operations are involved, further investigation is needed.

Our derivation of the H^∞ square-root arrays and Chandrasekhar recursions demonstrates a virtue of our Krein space approach to H^∞ estimation; the results appear to be more difficult to conceive and prove in the traditional H^∞ approaches. There are many variations of the conventional square-root array and Chandrasekhar recursions, *e.g.* for control problems, and the methods given here are directly applicable to extending these variations to the H^∞ setting as well.

References

[1] P.P. Khargonekar and K. M. Nagpal. Filtering and smoothing in an H^∞– setting. *IEEE Trans. on Automatic Control*, AC-36:151–166, 1991.

[2] T. Basar. Optimum performance levels for minimax filters, predictors and smoothers. *Systems and Control Letters*, 16:309–317, 1991.

[3] U. Shaked and Y. Theodor. H_∞–optimal estimation: A tutorial. In *Proc. IEEE Conference on Decision and Control*, pages 2278–2286, Tucson, AZ, Dec. 1992.

[4] M. J. Grimble. Polynomial matrix solution of the H^∞ filtering problem and the relationship to Riccati equation state-space results. *IEEE Trans. on Signal Processing*, 41(1):67–81, January 1993.

[5] B. Hassibi, A. H. Sayed, and T. Kailath. Recursive linear estimation in Krein spaces - Parts I & ii. In *Proc. IEEE Conference on Decision and Control*, pages 3489-3501, San Antonio, TX, Dec. 1993.

[6] M. Morf, G. S. Sidhu, and T. Kailath. Some New Algorithms for Recursive Estimation in Constant, Linear, Discrete-Time Systems. *IEEE Transactions on Automatic Control*, Vol. 19, pp. 315-323, 1974.

[7] A. H. Sayed and T. Kailath. Extended Chandrasekhar recursions. *IEEE Transactions on Automatic Control*, Feb 1994.

[8] T. Kailath. *Lecture Notes in Wiener and Kalman Filtering.* 2nd edition, Springer-Verlag, Berlin, 1980.

[9] B. Hassibi, A. H. Sayed and T. Kailath. H^∞ Square-Root and Chandrasekhar Recursions. Submitted to *IEEE Transactions on Automatic Control*, 1994.

æ