Technical Notes and Correspondence

Array Algorithms for H^∞ Estimation

Babak Hassibi, Thomas Kailath, and Ali H. Sayed

***Abstract*—In this paper we develop array algorithms for H^∞ filtering. These algorithms can be regarded as the Krein space generalizations of H^2 array algorithms, which are currently the preferred method for implementing H^2 filters. The array algorithms considered include two main families: square-root array algorithms, which are typically numerically more stable than conventional ones, and fast array algorithms which, when the system is time-invariant, typically offer an order of magnitude reduction in the computational effort. Both have the interesting feature that one does not need to explicitly check for the positivity conditions required for the existence of H^∞ filters, as these conditions are built into the algorithms themselves. However, since H^∞ square-root algorithms predominantly use J-unitary transformations, rather than the unitary transformations required in the H^2 case, further investigation is needed to determine the numerical behavior of such algorithms.**

***Index Terms*—Array algorithms, estimation, fast algorithms, robustness, H^∞.**

I. Introduction

Ever since its inception in 1960, the celebrated Kalman filter has played a central role in estimation. Although first expressed as a recursive algorithm which required the propagation of a certain Riccati recursion, current implementations of the Kalman filter are most often expressed in (what is called) an *array* form, and do not propagate this Riccati recursion directly. The square-root array algorithms devised in the late 1960's [1]–[3] are closely related to the QR method for solving systems of linear equations and have the properties of better conditioning, reduced dynamical range, and the use of orthogonal transformations, which typically lead to more stable algorithms. Furthermore, for constant systems, or in fact for systems where the time-variation is structured in a certain way, the Riccati and square-root recursions, both of which take $O(n^3)$ elementary computations (flops) per iteration (where n is the state-space dimension) can be replaced by more efficient fast recursions, which require only $O(n^2)$ flops per iteration [4]–[7].

Recently, there has been growing interest in worst case, or H^∞, estimation with the belief that the resulting estimators will have more robust performance in the face of model uncertainty and lack of statistical knowledge on the exogenous signals (see, e.g., [8]–[10]). The resulting H^∞ estimators involve propagating a Riccati recursion and bear a striking resemblance to the conventional Kalman filter. In a series of papers [11], we have recently shown that H^∞ filters are indeed Kalman filters, provided we set up estimation problems, not in the usual Hilbert space of random variables, but in an indefinite-metric (or so-called Krein) space. This observation leads to a unified approach to H^2 and H^∞ theory and shows a way to apply to the H^∞ setting many of the results developed for Kalman filtering and LQG control over the last three decades.

One immediate fall-out is that it allows one to generalize the square-root and fast array algorithms of H^2 estimation to the H^∞ setting. The hope is that the resulting H^∞ array algorithms will be more attractive for actual implementations of H^∞ filters and controllers. As we shall see, the H^∞ array algorithms have several interesting features. They 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.

II. H^2 Square-Root Array Algorithms

In state-space estimation problems we begin with the (possibly) time-variant state-space model

$$\begin{cases} x_{j+1} = F_j x_j + G_j u_j, \quad x_0 \\ y_j = H_j x_j + v_j \\ s_j = L_j x_j \end{cases} \tag{2.1}$$

where $F_j \in \mathcal{C}^{n\times n}$, $G_j \in \mathcal{C}^{n\times m}$, $H_j \in \mathcal{C}^{p\times n}$ and $L_j \in \mathcal{C}^{q\times n}$ are known, $\{u_j, v_j\}$ are the unknown disturbances, $\{y_j\}$ is the observed output, and $\{s_j\}$ is the signal we intend to estimate. We are typically interested in obtaining filtered and predicted estimates, denoted by $\hat{s}_{j\,|\,j}$ and $\hat{s}_j$, that use the observations $\{y_k, k \le j\}$ and $\{y_k, k < j\}$, respectively.

In conventional Kalman filtering the $\{x_0, u_j, v_j\}$ are assumed to be uncorrelated zero-mean random variables with variances $\Pi_0 \ge 0$, $Q_i \ge 0$, and $R_i > 0$, respectively, and the goal is to determine the estimates $\hat{s}_{j\,|\,j}$ and $\hat{s}_j$ so as to respectively minimize the expected squared estimation errors $(s_i - \hat{s}_{i\,|\,i})^*(s_i - \hat{s}_{i\,|\,i})$ and $(s_i - \hat{s}_i)^*(s_i - \hat{s}_i)$. The solutions to these problems are given by

$$\begin{cases} \hat{x}_{j+1} = F_j \hat{x}_j + K_{p,j}(y_j - H_j \hat{x}_j) \\ \hat{s}_j = L_j \hat{x}_j, \quad \hat{x}_0 = 0 \end{cases}$$
$$\begin{cases} \hat{x}_{j+1\,|\,j+1} = F_j \hat{x}_{j\,|\,j} + K_{f,j+1}(y_{j+1} - H_{j+1} F_j \hat{x}_{j\,|\,j}) \\ \hat{s}_{j\,|\,j} = L_j \hat{x}_{j\,|\,j}, \quad \hat{x}_{-1\,|\,-1} = 0 \end{cases} \tag{2.2}$$

where the gain matrices $K_{p,j}$ and $K_{f,j}$ can be computed via

$$\begin{aligned} K_{f,j} &= P_j H_j R_{e,j}^{-1}, \qquad K_{p,j} = F_j K_{f,j} \\ R_{e,j} &= R_j + H_j P_j H_j^* \end{aligned} \tag{2.3}$$

where P_j satisfies the Riccati recursion

$$\begin{aligned} P_{j+1} &= F_j P_j F_j^* + G_j Q_j G_j^* - K_{p,j} R_{e,j} K_{p,j}^* \\ P_0 &= \Pi_0. \end{aligned} \tag{2.4}$$

The matrix P_j appearing in the Riccati recursion (2.4) 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 nonnegative definite. Round-off errors can cause a loss of positive-definiteness, thus throwing all the

Manuscript received October 9, 1997. Recommended by Associate Editor, T. Katayama. This work was supported in part by DARPA through the Department of Air Force under Contract F49620-95-1-0525-P00001 and by the Joint Service Electronics Program at Stanford under Contract DAAH04-94-G-0058-P00003.

B. Hassibi and T. Kailath are with the Information Systems Laboratory, Stanford University, Stanford, CA USA.

A. H. Sayed is with the Department of Electrical Engineering, University of California, Los Angeles, CA USA.

Publisher Item Identifier S 0018-9286(00)04092-7.

obtained results into doubt. For this, and other reasons (reduced dynamic range, better conditioning, more stable algorithms, etc.) attention has moved in the Kalman filtering community to algorithms that propagate square-root factors of P_j, i.e. a matrix, $P_j^{1/2}$ say, such that $P_j = P_j^{1/2}(P_j^{1/2})^* = P_j^{1/2}P_j^{*/2}$ [1], [2].

Algorithm 1 (Conventional Square-Root Algorithm): The gain matrix $K_{p,j}$ necessary to obtain the predicted state estimates in the conventional Kalman filter (2.2) can be updated as

$$\begin{bmatrix} R_j^{1/2} & H_jP_j^{1/2} & 0 \\ 0 & F_jP_j^{1/2} & G_jQ_j^{1/2} \end{bmatrix} \Theta_j = \begin{bmatrix} R_{e,j}^{1/2} & 0 & 0 \\ K_{p,j}R_{e,j}^{1/2} & P_{j+1}^{1/2} & 0 \end{bmatrix}$$
$$P_0^{1/2} = \Pi_0^{1/2} \tag{2.5}$$

and the gain matrix $K_{f,j}$ necessary to obtain the filtered state estimates in (2.2) can be updated as

$$\begin{cases} \begin{bmatrix} R_j^{1/2} & H_jP_j^{1/2} \\ 0 & P_j^{1/2} \end{bmatrix} \Theta_j^{(1)} = \begin{bmatrix} R_{e,j}^{1/2} & 0 \\ K_{f,j}R_{e,j}^{1/2} & P_{j\,|\,j}^{1/2} \end{bmatrix}, \quad P_0^{1/2} = \Pi_0^{1/2} \\ [\, F_jP_{j\,|\,j}^{1/2} \quad G_jQ_j^{1/2} \,]\, \Theta_j^{(2)} = [\, P_{j+1}^{1/2} \quad 0 \,] \end{cases} \tag{2.6}$$

where Θ_j, $\Theta_j^{(1)}$, and $\Theta_j^{(2)}$ are any unitary matrices that triangularize the above pre-arrays.

Note 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 above algorithms can be verified by squaring both sides of (2.5) and (2.6), using the fact that $\Theta_j\Theta_j^* = I$, and comparing the entries of both sides of the result.

III. H^∞ Square-Root Array Algorithms

A. H^∞ Filtering

Let us begin by quoting the standard solutions to the so-called suboptimal H^∞ estimation problems using the notation of [11]. (See also [9] and [10].)

Theorem 1 (Suboptimal H^∞ Filters): Consider the standard state-space model (2.1) and define the matrices

$$R_j = \begin{bmatrix} -\gamma^2 I_q & 0 \\ 0 & I_p \end{bmatrix}$$

and

$$R_{e,j} = \begin{bmatrix} -\gamma^2 I_q & 0 \\ 0 & I_p \end{bmatrix} + \begin{bmatrix} L_j \\ H_j \end{bmatrix} P_j \, [\, L_j^* \quad H_j^* \,] \tag{3.1}$$

where the P_j satisfy the Riccati recursion

$$P_{j+1} = F_jP_jF_j^* + G_jG_j^* - F_jP_j\,[\, L_j^* \quad H_j^* \,]\, R_{e,j}^{-1} \begin{bmatrix} L_j \\ H_j \end{bmatrix} P_jF_j^*$$
$$P_0 = \Pi_0. \tag{3.2}$$

i) Given $\gamma > 0$, an H^∞ *a priori* filter that achieves

$$\sup_{x_0,\{u_j\},\{v_j\}} \frac{\sum_{j=0}^{i}(s_j - \hat{s}_j)^*(s_j - \hat{s}_j)}{x_0^*\Pi_0^{-1}x_0 + \sum_{j=0}^{i}u_j^*u_j + \sum_{j=0}^{i}v_j^*v_j} < \gamma^2 \tag{3.3}$$

exists if, and only if, *all leading submatrices* of R_j and $R_{e,j}$ have the same inertia[1] for $0 \le j \le i$. If this is the case, then all possible H^∞ *a priori* estimators that achieve (3.3) are given by those $\hat{s}_j = \mathcal{F}_{p,j}(y_0, \cdots y_{j-1})$ that satisfy

$$\sum_{j=0}^{k-1} \begin{bmatrix} \hat{s}_{j\,|\,j} - L_j\hat{x}_j \\ y_j - H_j\hat{x}_j \end{bmatrix}^* R_{e,j}^{-1} \begin{bmatrix} \hat{s}_{j\,|\,j} - L_j\hat{x}_j \\ y_j - H_j\hat{x}_j \end{bmatrix}$$
$$- (\hat{s}_{k\,|\,k} - L_k\hat{x}_k)^*(\gamma^2 I_q - L_kP_kL_k^*)^{-1}(\hat{s}_{k\,|\,k} - L_k\hat{x}_k) > 0 \tag{3.4}$$

for all $0 \le k \le i$, where $\hat{x}_j$ is given by the recursion

$$\hat{x}_{j+1} = F_j\hat{x}_j + K_{p,j} \begin{bmatrix} \hat{s}_j - L_j\hat{x}_j \\ y_j - H_j\hat{x}_j \end{bmatrix}, \qquad \hat{x}_0 = 0 \tag{3.5}$$

and $K_{p,j} = F_jP_j[L_j^* \;\; H_j^*]R_{e,j}^{-1}$.

ii) Given $\gamma > 0$, an H^∞ *a posteriori* filter that achieves

$$\sup_{x_0,\{u_j\},\{v_j\}} \frac{\sum_{j=0}^{i}(s_j - \hat{s}_{j\,|\,j})^*(s_j - \hat{s}_{j\,|\,j})}{x_0^*\Pi_0^{-1}x_0 + \sum_{j=0}^{i}u_j^*u_j + \sum_{j=0}^{i}v_j^*v_j} < \gamma^2 \tag{3.6}$$

exists if, and only if, all the *reversely leading* submatrices of R_j and $R_{e,j}$ have the same inertia for $0 \le j \le i$. If this is the case, then all possible H^∞ *a posteriori* estimators that achieve (3.6) are given by those $\hat{s}_{j\,|\,j} = \mathcal{F}_{f,j}(y_0, \cdots y_j)$ that satisfy

$$\sum_{j=0}^{k} \begin{bmatrix} \hat{s}_{j\,|\,j} - L_j\hat{x}_j \\ y_j - H_j\hat{x}_j \end{bmatrix}^* R_{e,j}^{-1} \begin{bmatrix} \hat{s}_{j\,|\,j} - L_j\hat{x}_j \\ y_j - H_j\hat{x}_j \end{bmatrix} > 0, \qquad 0 \le k \le i \tag{3.7}$$

where $\hat{x}_j$ is given by the same recursion as (3.5) with $\hat{s}_j$ replaced by $\hat{s}_{j\,|\,j}$.

Important special choices that guarantee (3.4) and (3.7) are the so-called *central* a filters, $\hat{s}_j = L_j\hat{x}_j$ and $\hat{s}_{j\,|\,j} = L_j\hat{x}_{j\,|\,j}$, where $\hat{x}_j$ satisfies the recursion

$$\hat{x}_{j+1} = F_j\hat{x}_j + K_{a,j}(y_j - F_j\hat{x}_j), \qquad \hat{x}_0 = 0 \tag{3.8}$$

with

$$K_{a,j} = \tilde{P}_jH_j^*(I_p + H_j\tilde{P}_jH_j^*)^{-1}$$

and

$$\tilde{P}_j = \left[P_j^{-1} - \gamma^{-2}L_j^*L_j\right]^{-1}$$

and where $\hat{x}_{j\,|\,j}$ satisfies the recursion

$$\hat{x}_{j+1\,|\,j+1} = F_j\hat{x}_{j\,|\,j} + K_{s,j+1}(y_{j+1} - H_{j+1}F_j\hat{x}_{j\,|\,j})$$
$$\hat{x}_{-1\,|\,-1} = 0 \tag{3.9}$$

with $K_{s,j} = P_jH_j^*(I_p + H_jP_jH_j^*)^{-1}$.

B. A Krein Space Formulation

The central H^∞ filters of (3.8) and (3.9) look very similar to their Kalman filter counterparts. In fact, we have recently shown that these H^∞ filters can be obtained as certain Kalman filters, not in an H^2 (Hilbert) space, but in an indefinite metric (so-called Krein) space [11], [12]. We shall not go into the details of estimation in indefinite-metric spaces here. Instead, we shall use the above observation as a guideline for generalizing the square-root array algorithms of Section II to the H^∞ setting.

[1]By the inertia of a Hermitian matrix, we mean the number of its positive, negative, and zero eigenvalues.

The first problem that occurs if one wants to extend (2.5) to the Krein space setting (of which the H^∞ filtering problem is a special case) is that the matrices R_i, Q_i, P_i and $R_{e\cdot i}$ need not be nonnegative definite. We thus need to employ the notion of an *indefinite* square-root

$$R_i = R_i^{1/2} S_i^{(1)} R_i^{*/2}, \qquad Q_i = Q_i^{1/2} S_i^{(2)} Q_i^{*/2}$$
$$P_i = P_i^{1/2} S_i^{(3)} P_i^{*/2}, \qquad R_{e,i} = R_{e,i}^{1/2} S_i^{(4)} R_{e,i}^{*/2}$$

where the $\{S_i^{(k)}\}$ are signature[2] matrices representing the inertia of $\{R_i, Q_i, P_i, R_{e,i}\}$. In the H^∞ setting we know that, when a solution exists, $P_i \geq 0$ [11], and that $R_{e,i}$ and R_i have the same inertia. Moreover, since $Q_i = I_m > 0$ and $R_i = I_p \oplus (-\gamma^2 I_q)$, we may write

$$R_i = R_i^{1/2} J R_i^{*/2}, \qquad Q_i = Q_i^{1/2} Q_i^{*/2}$$
$$P_i = P_i^{1/2} P_i^{*/2}, \qquad R_{e,i} = R_{e,i}^{1/2} J R_{e,i}^{*/2} \tag{3.10}$$

with $R_i^{1/2} = (\gamma I_q) \oplus I_p$ and $J = (-I_q) \oplus I_p$. This suggests that in the H^∞ filtering problem, the pre-array in (2.5) should be replaced by

$$\begin{bmatrix} \begin{bmatrix} \gamma I_q & 0 \\ 0 & I_p \end{bmatrix} & \begin{bmatrix} L_j \\ H_j \end{bmatrix} P_j^{1/2} & 0 \\ 0 & F_j P_j^{1/2} & G_j \end{bmatrix}. \tag{3.11}$$

Since the H^∞ estimation problem is most naturally formulated in a Krein space, it seems plausible that we should attempt to triangularize (3.11), not by a unitary transformation, but by a J-unitary transformation, i.e., a matrix Θ such that

$$\Theta J \Theta^* = \Theta^* J \Theta = J. \tag{3.12}$$

Unitary transformations (or ordinary rotations) preserve the length (or ordinary norm) of vectors. J-unitary transformations (or hyperbolic rotations), on the other hand, preserve the (indefinite) J-norm of vectors. Indeed, if $b = a\Theta$, with Θ J-unitary, then $bJb^* = a\Theta J \Theta^* a^* = aJa^*$.

Now it is well known that it is always possible to triangularize arrays using unitary transformations. But is this also true of J-unitary transformations? Thus, suppose we are given the (two-element) row vector $[a \;\; b]$ and are asked to hyperbolically rotate it so that the resulting vector lies along the direction of the x-axis. If such a transformation exist, then we can write

$$[\, a \quad b \,]\, \Theta = [\, c \quad 0 \,] \tag{3.13}$$

where $\Theta J \Theta^* = J$ and $J = 1 \oplus (-1)$. Since Θ is J-unitary this implies that

$$|a|^2 - |b|^2 = |c|^2 \geq 0. \tag{3.14}$$

Thus, $[a \;\; b]$ must have nonnegative J-norm. In other words, if the given $[a \;\; b]$ has negative J-norm (i.e., $|a|^2 - |b|^2 < 0$) then it is impossible to hyperbolically rotate it to lie along the x-axis.

Thus it is quite obvious that it is not always possible to triangularize arrays using J-unitary transformations. The precise condition follows.

Lemma 1 (J-Unitary Matrices and 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 $[A \;\; B]$ can be triangularized by a J-unitary transformation Θ as

$$[\, A \quad B \,]\, \Theta = [\, C \quad 0 \,]$$

with C lower (upper) triangular, if and only if, all leading (reversely leading) submatrices of S_1 and $AS_1A^* + BS_2B^*$ have the same inertia.

[2]A diagonal matrix with diagonal entries either $+1$ or -1.

Proof: The proof will be omitted for brevity and can be found, for example, in [12, Ch. 5]. ■

C. Square-Root Array Algorithms

We can now apply the result of Lemma 1 to the triangularization of the pre-array (3.11) using a J-unitary transformation with $J = (-I_q) \oplus I_p \oplus I_n \oplus I_m$. In fact, we need only consider the condition for triangularizing the first block row (since setting the block $(2, 3)$ entry of the post array to be zero can always be done via a standard unitary transformation). Thus we need only consider

$$\begin{bmatrix} \begin{bmatrix} I_q & 0 \\ 0 & \gamma I_p \end{bmatrix} & \begin{bmatrix} L_j \\ H_j \end{bmatrix} P_j^{1/2} \end{bmatrix} \Theta_j = [\, A \quad 0 \,] \tag{3.15}$$

using a J-unitary Θ_j, with $J = (-I_q) \oplus I_p \oplus I_n$. If we insist on a lower (upper) triangular A, from Lemma 1, the condition is that all leading (reversely leading) submatrices of J and

$$\underbrace{\begin{bmatrix} \gamma I_q & 0 \\ 0 & I_p \end{bmatrix} \begin{bmatrix} -I_q & 0 \\ 0 & I_p \end{bmatrix} \begin{bmatrix} \gamma I_q & 0 \\ 0 & I_p \end{bmatrix}}_{R_j} + \begin{bmatrix} L_j \\ H_j \end{bmatrix} P_j^{1/2} P_j^{*/2} [\, L_j^* \quad H_j^* \,] = R_{e,j} \tag{3.16}$$

should have the same inertia. But this is precisely the condition required for the existence of an H^∞ *a priori* (*a posteriori*) filter! (see Theorem 1). This result is quite useful—it states that an H^∞ filter exists if, and only if, the pre-array can be triangularized.

Now that we have settled the existence question, we can triangularize the pre-array (3.11) as

$$\begin{bmatrix} \begin{bmatrix} I_p & 0 \\ 0 & \gamma I_q \end{bmatrix} & \begin{bmatrix} H_j \\ L_j \end{bmatrix} P_j^{1/2} & 0 \\ 0 & F_j P_j^{1/2} & G_j \end{bmatrix} \Theta_j = \begin{bmatrix} A & 0 & 0 \\ B & C & 0 \end{bmatrix}. \tag{3.17}$$

To identify the elements A, B, and C in the post array we can square both sides of (3.17), use the fact that Θ_j is J-unitary, and compare the block entries of both sides. This leads to the following result.

Theorem 2 (H^∞ Square-Root Algorithm): The H^∞ *a priori* (*a posteriori*) filtering problem with level γ has a solution if, and only if, for all $j = 0, \cdots, i$ there exist J-unitary matrices (with $J = (-I_q) \oplus I_p \oplus I_n \oplus I_m$), Θ_j, such that

$$\begin{bmatrix} \begin{bmatrix} \gamma I_q & 0 \\ 0 & I_p \end{bmatrix} & \begin{bmatrix} L_j \\ H_j \end{bmatrix} P_j^{1/2} & 0 \\ 0 & F_j P_j^{1/2} & G_j \end{bmatrix} \Theta_j = \begin{bmatrix} R_{e,j}^{1/2} & 0 & 0 \\ K_{p,j} R_{e,j}^{1/2} & P_{j+1}^{1/2} & 0 \end{bmatrix}$$
$$P_0^{1/2} = \Pi_0^{1/2} \tag{3.18}$$

with $R_{e,j}^{1/2}$ lower (upper) triangular. If this is the case, then all possible H^∞ *a priori* (*a posteriori*) filters are given by (3.4) and (3.5) [(3.7) and (3.5)] with $K_{p,j}$ and $R_{e,j}$ as found above.

Note that, as in the H^2 case, the quantities necessary to update the square-root array and to calculate the desired estimates may all be found from the triangularized post-array. An interesting aspect of Theorem 2 is that there is no need to explicitly check for the existence conditions required of H^∞ filters. These conditions are built into the square-root algorithms themselves: an H^∞ estimator of the desired level exists if, and only if, the algorithms can be performed.

D. The Central Filters

Perhaps the most important filters in the parametrization of Theorem 1 are the so-called central filters, which we described in (3.8) and (3.9). In this section we shall show how the observer gains, $K_{a,j}$ and $K_{s,j}$, for the central filters can be readily obtained from the square-root array algorithms of Theorem 2.

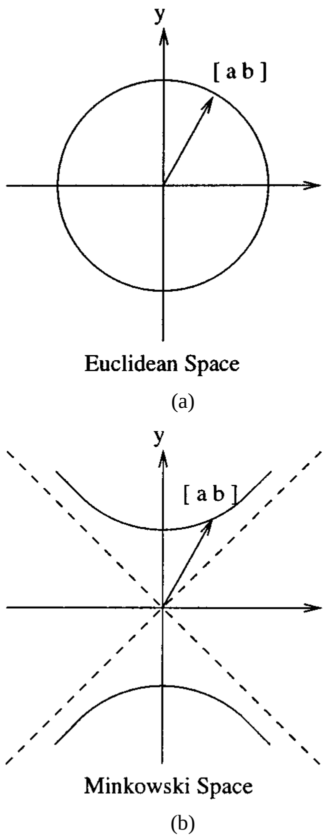

Fig. 1. Standard rotations versus hyperbolic rotations.

In fact, in [12] it is shown that, when $R_{e,j}^{1/2}$ is lower triangular, then

$$K_{a,j} = \left(\text{second block column of } K_{p,j} R_{e,j}^{1/2}\right) \cdot \left(\text{(2, 2) block entry of } R_{e,j}^{1/2}\right)^{-1}. \quad (3.19)$$

However, finding an expression for $K_{s,j}$ requires us to rewrite the *a posteriori* square-root algorithm of Theorem 2 via the following two-step procedure, which is the H^∞ analog of (2.6):

$$\begin{cases} \begin{bmatrix} \begin{bmatrix} \gamma I_q & 0 \\ 0 & I_p \end{bmatrix} & \begin{bmatrix} L_j \\ H_j \end{bmatrix} P_j^{1/2} \\ 0 & P_j^{1/2} \end{bmatrix} \Theta_j^{(1)} = \begin{bmatrix} R_{e,j}^{1/2} & 0 \\ K_{f,j} R_{e,j}^{1/2} & P_{j|j}^{1/2} \end{bmatrix} \\ [\, F_j P_{j|j}^{1/2} \quad G_j \,] \, \Theta_j^{(2)} = [\, P_{j+1}^{1/2} \quad 0 \,] \end{cases} \quad (3.20)$$

where $\Theta_j^{(1)}$ is J-unitary, with $J = (-I_q) \oplus I_p \oplus I_n$, $\Theta_j^{(2)}$ is unitary, and $R_{e,j}^{1/2}$ is upper triangular. In the above recursions, we of course have $K_{f,j} = P_j [L_j^* \;\; H_j^*] R_{e,j}^{-1}$. In [12] we have shown that, when $R_{e,j}^{1/2}$ is upper triangular, the gain matrix $K_{s,j}$ is given by

$$K_{s,j} = \left(\text{second block column of } K_{f,j} R_{e,j}^{1/2}\right) \cdot \left(\text{(2, 2) block entry of } R_{e,j}^{1/2}\right)^{-1}. \quad (3.21)$$

IV. H^2 Fast Array Algorithms

The conventional Kalman filter and square-root array recursions of Section II both require $O(n^3)$ operations per iteration. However, when the state-space model is time-invariant (or if the time-variation is structured in a certain way), there exist fast recursions that require only $O(n^2)$ operations per iteration [4]–[7]. In what follows we shall assume that the state-space model (2.1) is time-invariant, i.e., that $F_j = F$, $G_j = G$, $H_j = H$, and, that in the H^2 case, $Q_j = Q \geq 0$ and $R_j = R > 0$, for all j.

Under the aforementioned assumptions, it turns out that we can write

$$P_{j+1} - P_j = M_j S M_j, \qquad \text{for all } j \quad (4.1)$$

where M_j is a $n \times d$ matrix and S is a $d \times d$ signature matrix. Thus, for time-invariant state-space models, $P_{j+1} - P_j$ has rank d for all j and in addition has constant inertia. In several important cases, d can be much less than n. Two such cases are $P_0 = 0$, and $P_0 = \Pi$ (the solution to $\Pi = F\Pi F^* + GQG^*$). In any case, when $d < n$, propagating the smaller matrices M_j, which is equivalent to propagating the P_j, can offer significant computational reductions.

Algorithm 2 (Fast H^2 Recursions): The gain matrix $K_{p,j} = \overline{K}_{p,j} R_{e,j}^{-1/2}$ necessary to obtain the state estimates in the conventional Kalman filter (2.2) can be computed using

$$\begin{bmatrix} R_{e,j}^{1/2} & HM_j \\ \overline{K}_{p,j} & FM_j \end{bmatrix} \Theta_j = \begin{bmatrix} R_{e,j+1}^{1/2} & 0 \\ \overline{K}_{p,j+1} & M_{j+1} \end{bmatrix} \quad (4.2)$$

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

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

The validity of the above algorithm can be readily verified by squaring both sides of (4.2), using the fact that $\Theta_j J \Theta_j^* = J$, and comparing the entries on both sides of the result.

Note that compared to the square-root formulas, the size of the pre-array in the fast recursions has been reduced from $(p+n) \times (p+n+m)$ to $(p+n) \times (p+d)$. 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.

V. H^∞ Fast Array Algorithms

A. The General Case

Generalizing the fast array algorithms to the H^∞ case is now straightforward and follows the same pattern presented for the square-root algorithms. We shall therefore omit the details and refer the reader to [12, Chapter 5]. The result is given below.

Theorem 3 (H^∞ Fast Array Algorithm): The H^∞ *a priori* (*a posteriori*) filtering problem with level γ has a solution if, and only if, for all $j = 0, \cdots, i$ there exist J-unitary matrices (with $J = (-I_q) \oplus I_p \oplus S$), Θ_j, such that

$$\begin{bmatrix} R_{e,j}^{1/2} & \begin{bmatrix} L \\ H \end{bmatrix} M_j \\ K_{p,j} R_{e,j}^{1/2} & FM_j \end{bmatrix} \Theta_j = \begin{bmatrix} R_{e,j+1}^{1/2} & 0 \\ K_{p,j+1} R_{e,j+1}^{1/2} & M_{j+1} \end{bmatrix} \quad (5.1)$$

with $R_{e,j}^{1/2}$ lower (upper) triangular, and where the algorithm is initialized with

$$R_{e,0} = \begin{bmatrix} -\gamma^2 I_q & 0 \\ 0 & I_p \end{bmatrix} + \begin{bmatrix} L \\ H \end{bmatrix} \Pi_0 \, [\, L^* \quad H^* \,]$$
$$K_{p,0} = F\Pi_0 \, [\, L^* \quad H^* \,] \, R_{e,0}^{-1}$$

and

$$\begin{aligned} M_0 S M_0^* &= P_1 - \Pi_0 \\ &= F\Pi_0 F^* + GQG^* - K_{p,0} R_{e,0} K_{p,0}^* - \Pi_0. \end{aligned} \quad (5.2)$$

If this is the case, then all possible H^∞ *a priori* (*a posteriori*) filters are given by (3.4), (3.5) and (3.5)–(3.7).

Note that compared to the H^∞ square-root formulas, the size of the pre-array in the H^∞ fast recursions has been reduced from $(p+q+n)\times(p+q+n+m)$ to $(p+q+n)\times(p+q+d)$ where m, p, and q 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 fast recursions do not require explicitly checking the positivity conditions of Theorem 1—if the recursions can be carried out then an H^∞ estimator of the desired level exists, and if not, such an estimator does not exist.

B. The Central Filters

We finally remark that fast array algorithms can also be developed for the central H^∞ filters (3.8) and (3.9). The resulting statements are straightforward and will be omitted for brevity.

VI. Conclusion

In this paper, we developed square-root and fast array algorithms for the H^∞ *a priori* and *a posteriori* and filtering problems. These algorithms involve propagating the indefinite square-roots of the quantities of interest and have the interesting property that the appropriate inertia of these quantities is preserved. Moreover, 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 and fast array algorithms are preferred because of their better numerical behavior (in the case of square-root arrays) and their reduced computational complexity (in the case of the fast recursions). Since the H^∞ square-root and fast array algorithms are the direct analogs of their conventional counterparts, they may be more attractive for numerical implementations of H^∞ filters. However, since J-unitary rather than unitary operations are involved, further numerical investigation is needed.

Our derivation of the H^∞ square-root and fast array algorithms demonstrates a virtue of the Krein space approach to H^∞ estimation and control; the results appear to be more difficult to conceive and prove in the traditional H^∞ approaches. We should also mention that there are many variations of the conventional square-root and fast array algorithms, e.g. for control problems, and the methods given here are directly applicable to extending these variations to the H^∞ setting as well. Finally, the algorithms presented here are equally applicable to risk-sensitive estimation and control problems and to quadratic dynamic games.

References

[1] P. Dyer and S. McReynolds, "Extension of square-root filtering to include process noise," *J. Optimiz. Theory Appl.*, vol. 3, pp. 444–459, 1969.

[2] P. G. Kaminski, A. E. Bryson, and S. F. Schmidt, "Discrete square-root filtering: A survey of current techniques," *IEEE Trans. Automat. Contr.*, vol. 16, pp. 727–735, Dec. 1971.

[3] H. L. Harter, "The Method of Least-Squares and Some Alternatives," Aerospace Res. Lab., Air Force Systems Command, Wright-Patterson AFB, Ohio, Tech. Rep. ARL 72-0129, Sept. 1972.

[4] T. Kailath, "Some new algorithms for recursive estimation in constant linear systems," *IEEE Trans. Inform. Theory*, vol. 19, pp. 750–760, Nov. 1973.

[5] M. Morf, G. S. Sidhu, and T. Kailath, "Some new algorithms for recursive estimation in constant, linear, discrete-time systems," *IEEE Trans. Automat. Contr.*, vol. 19, pp. 315–323, 1974.

[6] A. Lindquist, "Some reduced-order non-Riccati equations for linear least-squares estimation: The stationary, single-output case," *Int. J. Contr.*, vol. 24, no. 6, pp. 821–42, Dec. 1976.

[7] A. H. Sayed and T. Kailath, "Extended Chandrasekhar recursions," *IEEE Trans. Automat. Contr.*, vol. 39, pp. 619–623, Mar. 1994.

[8] T. Basar, "Optimum performance levels for minimax filters, predictors and smoothers," *Syst. Contr. Lett.*, vol. 16, pp. 309–317, 1991.

[9] P. P. Khargonekar and K. M. Nagpal, "Filtering and smoothing in an H^∞ setting," *IEEE Trans. Automat. Contr.*, vol. 36, pp. 151–166, 1991.

[10] U. Shaked and Y. Theodor, "H^∞-optimal estimation: A tutorial," in *Proc. IEEE Conf. Decision and Control*, Tucson, AZ, 1992, pp. 2278–2286.

[11] B. Hassibi, A. H. Sayed, and T. Kailath, "Linear estimation in Krein spaces—Parts I and II: Theory and applications," *IEEE Trans. Automat. Contr.*, vol. 41, pp. 18–49, Jan. 1996.

[12] B. Hassibi, "Indefinite metric spaces in estimation, control and adaptive filtering," Ph.D. dissertation, Stanford Univ., CA, 1996.

[13] S. Bittante, A. J. Laub, and J. C. Willems, Eds., *The Riccati Equation*. Berlin, Germany: Springer-Verlag, 1991.

Reliable Control of Nonlinear Systems

Yew-Wen Liang, Der-Cherng Liaw, and Ti-Chung Lee

***Abstract*—In this paper, we extend Veillette's results (1995) to the study of reliable linear-quadratic regulator problem for nonlinear systems. This is achieved by employing the Hamilton–Jacobi inequality in the nonlinear case instead of algebraic Riccati equation in the linear one. The proposed state-feedback controllers are shown to be able to tolerate the outage of actuators within a prespecified subset of actuators. Both the gain margins of guaranteeing system stability and retaining a performance bound are estimated.**

***Index Terms*—Algebraic Riccati equation, Hamilton–Jacobi inequality, linear-quadratic regulator problem.**

I. Introduction

The study of the design of reliable control systems which can tolerate the failure of the control components and retain the desired system performance has recently attracted considerable attention (see e.g., [1] and [6]–[10]). Several approaches for the design of the reliable controllers have been proposed; however, most of those efforts are focused on linear control systems [1], [6]–[8] rather than nonlinear ones. For instance, Veillette employed the algebraic Riccati equation approach to develop a procedure for the design of a state-feedback controllers, which could tolerate the outage within a selected subset of actuators while retaining the stability and the known quadratic performance bound [7]. Both the gain margins for guaranteeing system's stability and preserving system performance were also estimated in [7]. Two recent papers employed the Hamilton–Jacobi inequality approach to investigate the nonlinear reliable control problem. One studied the design of controllers that could guarantee locally asymptotic stability and H_∞ performance even when some components failed within a prespecified subset of control components

Manuscript received August 27, 1997, revised April 5, 1998. Recommended by Associate Editor, A. J. van der Schaft. This work was supported by the National Science Council, Taiwan R.O.C. under Grant NSC-83-0404-E-009-001 and Grant NSC 84-2212-E-009-002.

Y.-W. Liang and D.-C. Liaw are with the Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu 30039, Taiwan, R.O.C. (e-mail: dcliaw@cc.nctu.edu.tw).

T.-C. Lee is with the Department of Electrical Engineering, Ming Hsin Institute of Technology, Hsin-Fong, Hsinchu 304, Taiwan, R.O.C.

Publisher Item Identifier S 0018-9286(00)04097-6.