Finite-Horizon Robust Kalman Filter Design

Minyue Fu, *Senior Member, IEEE*, Carlos E. de Souza, *Senior Member, IEEE*, and Zhi-Quan (Tom) Luo, *Member, IEEE*

***Abstract*—In this paper, we study the problem of finite-horizon Kalman filtering for systems involving a norm-bounded uncertain block. A new technique is presented for robust Kalman filter design. This technique involves using multiple scaling parameters that can be optimized by solving a semidefinite program. The use of optimized scaling parameters leads to an improved design. A recursive design method that can be applied to real-time applications is also proposed.**

***Index Terms*—Adaptive filtering, Kalman filtering, robust filtering, robust signal processing.**

I. INTRODUCTION

FINITE-horizon Kalman filters, including recursive least-squares filters as a special case, are widely used in signal processing applications. Compared with infinite-horizon Kalman filters, the finite-horizon ones can offer a better transient performance, which is an important property for applications where signals are nonstationary.

One of the problems with Kalman filters, which has been well recognized now, is that they can be sensitive to system data, or in other words, they may lack robustness. A typical phenomenon is that the performance of the filter, although being optimal for a "nominal" system, may deteriorate very quickly as the system data drift; see, e.g., [4]. This is, of course, not acceptable for applications where a good system model is hard to obtain or the system drifts. Motivated by this problem, a number of papers have attempted to generalize the classical Kalman filter to systems involving a norm-bounded uncertain block; see [2]–[6]. Note that norm-bounded blocks are used to represent inaccuracies in the system model. The resulting filters are often called robust Kalman filters.

The design of robust Kalman filters faces a major obstacle in comparison with the classical Kalman filters. There are two prevailing properties possessed by classical finite-horizon Kalman filters. First, an optimal filter at time k leads to an optimal filter at $k+1$. That is, an optimal filter at k produces a minimum state estimation error at k (in the variance sense), which is the best initial condition for the filter design at $k+1$. Second, the optimal filter for state estimation is also optimal for estimation of any other signal, provided it is a linear function of the state. Unfortunately, neither of the two properties carries through when the system involves uncertainties. More precisely, a filter that produces a small state estimation error at time k may worsen the state estimation at time $k+1$. Similarly, a filter that minimizes the state estimation error may not be optimal for estimation of the signal of interest, even when it is a linear combination of the state.

A commonly used technique for robust Kalman filter design is to apply the so-called S-Procedure, which replaces the uncertainty block with a scaling parameter. This yields an upper bound for the covariance of the estimation error. Two types of scaling parameters have been used: constant and time-varying. A constant scaling parameter (τ) is used in [3], [4], and [6] and is most suitable for infinite-horizon or stationary filtering problems. One serious problem with using a constant scaling parameter is that the entailed conservatism can aggregate quickly as time evolves and may lead to a very poor estimator. Time-varying scaling parameters (τ_k) are more flexible, and if they are carefully chosen, the amount of conservatism can be reduced. Two papers have used time-varying scaling parameters. In [5], a simple formula is given, but the scaling parameter is not optimized in any way. In [2], the scaling parameter is chosen using a semidefinite program. However, as we will reveal later, the scaling parameter obtained at time k using [2] may lead to a poor estimation at future times. In addition, the semidefinite program to be solved in [2] is quite cumbersome.

In this paper, we intend to carry out some deeper study on finite-horizon Kalman filtering for systems involving a norm-bounded uncertain block. Our focus will be on how to choose scaling parameters. A summary of our results is given as follows.

- We show that optimal scaling parameters for time k may lead to poor estimation at future times. Subsequently, two types of scaling parameters are suggested: one optimal for time k and one used for the future. In fact, at each time k, all the scaling parameters $\tau_0, \ldots, \tau_k$ need to be reoptimized.
- The design of the estimator has the following separation properties.
  - — The covariance of the estimation error at $k+1$ depends only on the scaling parameters $\tau_0, \ldots, \tau_k$ and the system data and not on other parameters in the filter. Thus, the scaling parameters can be optimized first. In particular, we note that they depend on the signal to be estimated.
  - — Once the scaling parameters are determined, an optimal filter can be generated using an algebraic formula. In particular, we note that the optimal filter does not explicitly depend on the signal to be estimated. Implicit dependence happens only through the scaling parameters.

Manuscript received May 4, 1999; revised May 14, 2001. The associate editor coordinating the review of this paper and approving it for publication was Dr. Joseph M. Francos.

M. Fu is with the Department of Electrical and Computer Engineering, the University of Newcastle, Callaghan, Australia (e-mail: eemf@ee.newcastle.edu.au).

C. E. de Souza is with the Department of Systems and Control, Laboratório Nacional de Computação Científica—LNCC, Petrópolis, Brazil.

Z.-Q. Luo is with the Department of Electrical and Computer Engineering, MacMaster University, Hamilton, ON L8S 4K1 Canada.

Publisher Item Identifier S 1053-587X(01)07068-4.

- We show that optimal scaling parameters can be computed using a semidefinite program. The size of the program is moderate and grows at the rate k. An suboptimal scheme is also given that requires a constant amount of computation at each k.

II. Covariance Analysis

Consider the following uncertain system:

$$\begin{aligned} x_{k+1} &= (A_k + H_k F_k E_k) x_k + B_k w_k \\ z_k &= C_k x_k \end{aligned} \tag{2.1}$$

where $x_k \in \mathcal{R}^n$ is the state, $z_k \in \mathcal{R}^p$ is a linear combination of x_k; $A_k \in \mathcal{R}^{n\times n}$, $H_k \in \mathcal{R}^{n\times i}$, $E_k \in \mathcal{R}^{j\times n}$, $B_k \in \mathcal{R}^{n\times m}$ and $C_k \in \mathcal{R}^{p\times n}$ are given matrices with full row rank, $F_k \in \mathcal{R}^{i\times j}$ represents norm-bounded time-varying uncertainty, i.e.,

$$F_k F_k^t \le I, \qquad \forall\, k = 0, 1, \ldots \tag{2.2}$$

and w_k and x_0 are zero-mean and independent and satisfy the following second-order statistics:

$$\mathcal{E}(w_k w_l^t) = \begin{cases} I, & \text{if } k = l \\ 0, & \text{otherwise} \end{cases}; \quad \mathcal{E}(x_0 x_0^t) = \Sigma_0 > 0. \tag{2.3}$$

Without loss of generality, $E_k \ne 0$ for all k. To assure that the order of the system is not degenerate, we further assume

$$\text{rank}\,[A_k H_k B_k] = n \qquad \forall\, k. \tag{2.4}$$

Denote by Σ_k and $\Sigma_{z,k} = C_k \Sigma_k C_k^t$ the covariance matrices of x_k and z_k, respectively. The *(worst-case) covariance analysis problem* is as follows: Given $T \ge 0$, determine the worst-case $\Sigma_{z,T+1}$, i.e.,

$$L_{T+1} = \max\left\{L(\Sigma_{z,T+1}) \colon F_k F_k^t \le I,\ 0 \le k \le T\right\} \tag{2.5}$$

where $L(\Sigma)$ is any given linear function of Σ. In particular, it is common to choose $L(\Sigma) = \text{trace}(\Sigma)$.

We first introduce the so-called S-Procedure (see, e.g., [1]).

Lemma 2.1: Given $M, A, \Sigma \in \mathcal{R}^{n\times n}$, $H \in \mathcal{R}^{n\times i}$, and $E \in \mathcal{R}^{j\times n}$ with $\Sigma = \Sigma^t > 0$, the following inequality holds:

$$\begin{aligned} &M - (A + HFE)\Sigma(A + HFE)^t \ge 0 \\ &\quad \forall\, F \in \mathcal{R}^{i\times j},\ FF^t \le I \end{aligned} \tag{2.6}$$

if and only if there exists $\tau \ge 0$ such that

$$\begin{bmatrix} M & A & H \\ A^t & \Sigma^{-1} - \tau E^t E & 0 \\ H^t & 0 & \tau I \end{bmatrix} \ge 0. \tag{2.7}$$

Next, we give a solution to the covariance analysis problem for the case $T = 0$.

Theorem 2.1: Define

$$\Sigma_1(\tau_0) = A_0 S_0 A_0^t + B_0 B_0^t + \tau_0^{-1} H_0 H_0^t, \qquad \tau_0 \in \mathbf{R} \tag{2.8}$$

where

$$S_0 = \Sigma_0 + \Sigma_0 E_0^t (\tau_0^{-1} I - E_0 \Sigma_0 E_0^t)^{-1} E_0 \Sigma_0. \tag{2.9}$$

Then

$$\Sigma_1(\tau_0) \ge \Sigma_1,\ \Sigma_1(\tau_0) > 0, \qquad \forall\, 0 < \tau_0 < \|E_0 \Sigma_0 E_0^t\|^{-1}. \tag{2.10}$$

In addition

$$L_1 \le \overline{L}_1 = \inf\left\{L(C_1 \Sigma_1(\tau_0) C_1^t) \colon 0 < \tau_0 < \|E_0 \Sigma_0 E_0^t\|^{-1}\right\} \tag{2.11}$$

and $L_1 = \overline{L}_1$ if C_1 is a row vector. Further, the optimal τ_0 for $\overline{L}_1$ can be found by solving the following semidefinite program:

$$\begin{aligned} L_1 &= \min L(C_1 X C_1^t) \\ &\text{s.t.} \begin{bmatrix} X - B_0 B_0^t & A_0 & H_0 \\ A_0^t & \Sigma_0^{-1} - \tau_0 E_0^t E_0 & 0 \\ H_0^t & 0 & \tau_0 I \end{bmatrix} \ge 0 \\ X &= X^t,\ \tau_0 \ge 0. \end{aligned} \tag{2.12}$$

Proof: We first consider the general case where C_1 may not be a row vector. Obviously

$$\Sigma_1 = (A_0 + H_0 F_0 E_0)\Sigma_0 (A_0 + H_0 F_0 E_0)^t + B_0 B_0^t.$$

It follows that

$$\begin{aligned} L_1 &\le \min L(C_1 X C_1^t) \\ \text{s.t. } X &\ge (A_0 + H_0 F_0 E_0)\Sigma_0 (A_0 + H_0 F_0 E_0)^t \\ &\quad + B_0 B_0^t, \qquad \forall\, F_0 \\ X &= X^T. \end{aligned}$$

Using the S-Procedure (Lemma 2.1), the above is equivalent to (2.12). The equivalence between (2.12) and (2.11) follows from Schur's complements.

For the case where C_1 is a row vector

$$\begin{aligned} L_1 = \max_{F_0 F_0^t \le I} & C_1 (A_0 + H_0 F_0 E_0)\Sigma_0 \\ & \cdot (A_0 + H_0 F_0 E_0)^t C_1^t + C_1 B_0 B_0^t C_1^t. \end{aligned}$$

Given any $\ell > 0$, we have $L_1 \le \ell$ iff

$$\ell - C_1 B_0 B_0^t C_1^t - C_1 (A_0 + H_0 F_0 E_0)\Sigma_0 (A_0 + H_0 F_0 E_0)^t C_1^t \le 0$$

for all $F_0 F_0^t \le I$. Using the S-Procedure again, the above holds iff there exists some $\tau_0 \ge 0$ such that

$$\begin{bmatrix} \ell - C_1 B_0 B_0^t C_1^t & C_0 A_0 & C_0 H_0 \\ A_0^t C_0^t & \Sigma_0^{-1} - \tau_0 E_0^t E_0 & 0 \\ H_0^t C_0^t & 0 & \tau_0 I \end{bmatrix} \ge 0. \tag{2.13}$$

Obviously

$$L_1 = \min_{\tau_0 \ge 0} \ell$$

subject to (2.13). Writing $\ell = C_1 X C_1^t$ for some $X = X^t \ge 0$, it is easy to see the optimization problem above is equivalent to (2.12).

To show (2.10), we first note that X above is an upper bound for Σ_1. For each τ_0, the optimal X equals $\Sigma_1(\tau_0)$. Hence,

$\Sigma_1(\tau_0) \geq \Sigma_1$. Next, $\Sigma_1(\tau_0) > 0$ follows from $\Sigma_0 > 0$, (2.4), and a rewriting of (2.8) as follows:

$$\Sigma_1(\tau_0) = [A_0 \; H_0 \; B_0] \mathrm{diag}\,\{S_0, \tau_0^{-1}I, I\} \begin{bmatrix} A_0^t \\ H_0^t \\ B_0^t \end{bmatrix}.$$

■

The role of the scaling parameter τ_0 is to identify the worst-case F_0.

Returning to the problem in (2.5) for $T > 0$ where more than one F_k terms are involved, it turns out that they can be replaced by additional scaling parameters τ_k to compute L_{T+1}. This is detailed as follows.

Theorem 2.2: Denote $\tau = [\tau_0, \ldots, \tau_T]$ and define

$$\begin{aligned} \Sigma_0(\tau) &= \Sigma_0 \\ \Sigma_{k+1}(\tau) &= B_k B_k^t + \tau_k^{-1} H_k H_k^t + A_k S_k(\tau) A_k^t \end{aligned} \tag{2.14}$$

for $k = 0, 1, \ldots, T$, where

$$\begin{aligned} S_k(\tau) &= \Sigma_k(\tau) + \Sigma_k(\tau) E_k^t \\ &\quad \cdot \left(\tau_k^{-1} I - E_k \Sigma_k(\tau) E_k^t\right)^{-1} E_k \Sigma_k(\tau) \end{aligned} \tag{2.15}$$

with

$$S_k^{-1}(\tau) = \Sigma_k^{-1}(\tau) - \tau_k E_k^t E_k. \tag{2.16}$$

In addition, define

$$\Omega = \left\{\tau{:}0 < \tau_k < \left\|E_k \Sigma_k(\tau) E_k^t\right\|^{-1}, k = 0, \ldots, T\right\}. \tag{2.17}$$

Then

$$\Sigma_{k+1}(\tau) \geq \Sigma_{k+1}, \quad \Sigma_{k+1}(\tau) > 0, \qquad \forall \tau \in \Omega, \; 0 \leq k \leq T. \tag{2.18}$$

Next, an upper bound for L_{T+1} is given by

$$\overline{L}_{T+1} = \inf_{\tau \in \Omega} L\left(C_{T+1} \Sigma_{T+1}(\tau) C_{T+1}^t\right). \tag{2.19}$$

Further, the optimum above can be found by solving the following semi-definite program:

$$\begin{aligned} \overline{L}_{T+1} &= \min L\left(C_{T+1} X C_{T+1}^t\right) \\ &\quad \text{s.t.} \begin{bmatrix} X & U_{T+1} \\ U_{T+1}^t & \Pi_{T+1} \end{bmatrix} \geq 0 \\ X &= X^t, \qquad \tau \geq 0 \end{aligned} \tag{2.20}$$

where U_T and Π_T are defined recursively as

$$\begin{cases} U_0 = I \\ \Pi_0 = \Sigma_0^{-1} \\ \Pi_{k+1} = \mathrm{diag}\,\{\Pi_k - \tau_k U_k^t E_k^t E_k U_k, \tau_k I_i, I_m\} \\ U_{k+1} = [A_k U_k H_k B_k], & k \geq 0. \end{cases} \tag{2.21}$$

Proof: The problem in (2.5) is equivalent to

$$\begin{aligned} &\min_{X = X^t} \max\left\{L(C_{T+1} X C_{T+1}^t){:}\; F_k F_k \leq I, \; 0 \leq k \leq T\right\} \\ &\text{s.t. } X \geq \Sigma_{T+1}. \end{aligned} \tag{2.22}$$

Let

$$\begin{aligned} \Phi_{k,s} &= (A_k + H_k F_k E_k) \cdots (A_s + H_s F_s E_s) \\ &\quad 0 \leq s \leq k \end{aligned} \tag{2.23}$$

and

$$\begin{aligned} Y_k &= B_T B_T^t + \Phi_{T,T} B_{T-1} B_{T-1}^t \Phi_{T,T}^t \\ &\quad + \cdots + \Phi_{T,k+1} B_k B_k^t \Phi_{T,k+1}^t. \end{aligned} \tag{2.24}$$

Then

$$\begin{aligned} \Sigma_{T+1} &= \Phi_{T,1} \left[(A_0 U_0 + H_0 F_0 E_0 U_0) \Pi_0^{-1} \right. \\ &\quad \left. \cdot (A_0 U_0 + H_0 F_0 E_0 U_0)^t + B_0 B_0^t\right] \Phi_{T,1}^t + Y_1. \end{aligned}$$

It follows that $X \geq \Sigma_{T+1}$ iff

$$\begin{aligned} X &\geq \Phi_{T,1} \left[(A_0 U_0 + H_0 F_0 E_0 U_0) \Pi_0^{-1} \right. \\ &\quad \left. \cdot (A_0 U_0 + H_0 F_0 E_0 U_0)^t + B_0 B_0^t\right] \Phi_{T,1}^t + Y_1. \end{aligned} \tag{2.25}$$

Using the above and the S-Procedure (Lemma 2.1), (2.25) holds for all $F_0 F_0^t \leq I$ iff

$$\begin{bmatrix} \hat{X} & \Phi_{T,1} A_0 U_0 \\ U_0^t A_0^t \Phi_{T,1}^t & \Pi_0 - \tau_0 U_0^t E_0^t E_0 U_0 \end{bmatrix} \geq 0 \tag{2.26}$$

for some $\tau_0 > 0$, where

$$\hat{X} = X - Y_1 - \Phi_{T,1} B_0 B_0^t \Phi_{T,1}^t - \tau_0^{-1} \Phi_{T,1} H_0 H_0^t \Phi_{T,1}^t.$$

Then, use Schur's complements to convert the above into $\Pi_1 > 0$ and

$$\begin{aligned} X &\geq \Phi_{T,1} U_1 \Pi_1^{-1} U_1^t \Phi_{T,1}^t + Y_1 \\ &= \Phi_{T,2} \left[(A_1 U_1 + H_1 F_1 E_1 U_1) \Pi_1^{-1} \right. \\ &\quad \left. \cdot (A_1 U_1 + H_1 F_1 E_1 U_1)^t + B_1 B_1^t\right] \Phi_{T,2}^t + Y_2 \end{aligned} \tag{2.27}$$

and we return to a form similar to (2.25). In general, the optimal τ_0 depends on $F_1, F_2, \ldots$, but if we assume τ_0 is constant, the process above can be done recursively, and it will eventually give (2.20). The reason for (2.20) to give an upper bound only is because τ_k are assumed to be a constant in the recursion.

To show the equivalence between (2.20) and (2.19), we note that the optimal X for (2.20) is

$$X = U_{T+1} \Pi_{T+1}^{-1} U_{T+1}^t.$$

Thus, it suffices to show by induction that

$$\Sigma_k(\tau) = U_k \Pi_k^{-1} U_k^t, \qquad \forall k \geq 0.$$

This is certainly true for $k = 0$. Suppose that it is true for some k; then

$$\begin{aligned} &U_{k+1} \Pi_{k+1}^{-1} U_{k+1}^t \\ &\quad = A_k U_k (\Pi_k - \tau_k U_k^t E_k^t E_k U_k)^{-1} U_k^t A_k^t \\ &\qquad + B_k B_k^t + \tau_k^{-1} H_k H_k^t \\ &\quad = A_k U_k \left(\Pi_k^{-1} + \Pi_k^{-1} U_k^t E_k^t (\tau_k^{-1} I - E_k U_k \Pi_k^{-1} U_k^t E_k^t)^{-1} \right. \\ &\qquad \left. \cdot E_k U_k \Pi_k^{-1}\right) U_k^t A_k^t + B_k B_k^t + \tau_k^{-1} H_k H_k^t \\ &\quad = \Sigma_{k+1}(\tau). \end{aligned}$$

Finally, $\Sigma_{T+1}(\tau) \geq \Sigma_{T+1}$ for all $\tau \in \Omega$ because $X \geq \Sigma_{T+1}$. Since T is arbitrary, we must have $\Sigma_{k+1}(\tau) \geq \Sigma_{k+1}$ for all $\tau \in \Omega$. In addition, $\Sigma_{k+1}(\tau) > 0$ follows from $\Sigma_0 > 0$ and (2.4), as shown in the proof of Theorem 2.1. ■

Remark 2.1: Theorem 2.2 suggests that τ needs to be recomputed as T changes. This is indeed the case. In fact, we will show in Section VI that an optimal τ at a given time T may not be optimal at a different time. Because of this property, we will denote the optimal τ_k at time T by $\tau_{T,k}$, $k = 0, \ldots, T$ whenever necessary.

Remark 2.2: In Theorem 2.2, we have assumed nonsingularity of Σ_0. If Σ_0 is singular, we can always decompose it into $U_0 \Pi_0^{-1} U_0^t$ for some U_0 and Π_0. With these U_0 and Π_0, the recursion in (2.21) will still be valid.

III. Robust Filter Design: Problem Statement

We extend the system (2.1) to the following:

$$\begin{aligned} x_{k+1} &= (A_k + H_{1,k} F_k E_k) x_k + B_k w_k \\ y_k &= (C_{2,k} + H_{2,k} F_k E_k) x_k + v_k \\ z_k &= C_{1,k} x_k \end{aligned} \tag{3.28}$$

where $y_k \in \mathcal{R}^r$ is a measured output, $C_{2,k} \in \mathbf{R}^{r\times n}$, $H_{2,k} \in \mathbf{R}^{r\times i}$, v_k is a zero-mean measurement noise, which is independent of w_k, and with statistics

$$\mathcal{E}(v_k v_l^t) = \begin{cases} I, & \text{if } k = l \\ 0, & \text{otherwise.} \end{cases} \tag{3.29}$$

Other matrices are defined accordingly. In the design problem, z_k is a linear combination of x_k to be estimated. Similar to (2.4), it is assumed that

$$\text{rank} \begin{bmatrix} A_k & H_{1,k} & B_k \\ C_{2,k} & H_{2,k} & 0 \end{bmatrix} = n + r, \qquad \forall k. \tag{3.30}$$

We further assume that $C_{1,k}$ have full row rank for all k.

We consider an nth-order robust linear filter of the following form:

$$\begin{aligned} \hat{x}_{k+1} &= \hat{A}_k \hat{x}_k + \hat{B}_k (y_k - C_{2,k} \hat{x}_k) \\ \hat{x}_0 &= 0 \\ \hat{z}_k &= C_{1,k} \hat{x}_k. \end{aligned} \tag{3.31}$$

Note that the use of the same $C_{1,k}$ and $C_{2,k}$ does not lose any generality. More precisely, it can be easily shown that any nth-order linear time-invariant estimator for z_k with input y_k and zero initial state can have a state-space realization in the form of (3.31).

Given the filter above, the augmented system involving x_k and $\hat{x}_k$ is given by

$$\begin{aligned} \tilde{x}_{k+1} &= (\tilde{A}_k + \tilde{H}_k F_k \tilde{E}_k) \tilde{x}_k + \tilde{B}_k \tilde{w}_k \\ e_k &= \tilde{C}_k \tilde{x}_k \end{aligned} \tag{3.32}$$

where e_k is the estimation error, and

$$\begin{aligned} \tilde{x}_k &= \begin{bmatrix} x_k \\ \hat{x}_k \end{bmatrix}, \quad \tilde{w}_k = \begin{bmatrix} w_k \\ v_k \end{bmatrix} \\ \tilde{A}_k &= \begin{bmatrix} A_k & 0 \\ \hat{B}_k C_{2,k} & \hat{A}_k - \hat{B}_k C_{2,k} \end{bmatrix} \\ \tilde{B}_k &= \begin{bmatrix} B_k & 0 \\ 0 & \hat{B}_k \end{bmatrix}, \quad \tilde{H}_k = \begin{bmatrix} H_{1,k} \\ \hat{B}_k H_{2,k} \end{bmatrix} \\ \tilde{E}_k &= [E_k \ \ 0]; \quad \tilde{C}_k = C_{1,k}[I \ \ -I]. \end{aligned}$$

We will denote by $\tilde{\Sigma}_k$, $\Sigma_{x,k}$ and $\Sigma_{e,k}$ the covariance matrices of $\tilde{x}_k$, $x_k - \hat{x}_k$, and e_k.

Similar to the previous section, scaling parameters τ_k will be used to replace the uncertainty block F_k, which yields parameterized covariance matrices $\tilde{\Sigma}_{k+1}(\tau_k)$, $\Sigma_{x,k+1}(\tau_k)$, and $\Sigma_{e,k+1}(\tau_k)$ that serve as the upper bounds for $\tilde{\Sigma}_{k+1}$, $\Sigma_{x,k+1}$ and $\Sigma_{e,k+1}$, respectively. If this is done recursively, i.e., $\tau_0, \tau_1, \ldots, \tau_k$ are used, then these upper bounds depend on all these scaling parameters. In this case, we will denote these upper bounds by $\tilde{\Sigma}_{k+1}(\tau)$, $\Sigma_{x,k+1}(\tau)$, and $\Sigma_{e,k+1}(\tau)$, respectively.

With the discussion above, we propose a number of technical problems as follows:

P1) Given $\tilde{\Sigma}_k$ and τ_k, find the optimal filter at k (i.e., $\hat{A}_k$ and $\hat{B}_k$) such that $L(\Sigma_{e,k+1}(\tau_k))$ is minimized.

P2) Given $\tilde{\Sigma}_k$, find optimal τ_k, $\hat{A}_k$ and $\hat{B}_k$ such that $L(\Sigma_{e,k+1}(\tau_k))$ is minimized.

P3) Given T, Σ_0, find optimal τ_k and the optimal filter at all k, $k = 0, \ldots, T$ such that $L(\Sigma_{e,T+1}(\tau))$ is minimized.

Obviously, our aim is to solve P3, whereas P1 and P2 are immediate steps.

IV. Robust Filter Design: Solutions

Solutions to the three problems P1–P3 are given in this section.

Problem P1:

Theorem 3.1: Suppose

$$\begin{aligned} \tilde{\Sigma}_k &= \begin{bmatrix} \Sigma_{x,k} + \Sigma_{2,k} & \Sigma_{2,k} \\ \Sigma_{2,k} & \Sigma_{2,k} \end{bmatrix}, \qquad \text{for some } \Sigma_{x,k} > 0 \\ \Sigma_{2,k} &\geq 0 \end{aligned} \tag{4.33}$$

(which holds at $k = 0$), and $0 < \tau_k < \|E_k \Sigma_{1,k} E_k^t\|^{-1}$, where $\Sigma_{1,k} = \Sigma_{x,k} + \Sigma_{2,k}$. Then, the optimal solution to Problem 1 is given as follows:

$$\hat{A}_k = A_k + \Delta_k \tag{4.34}$$

$$\Delta_k = (A_k - \hat{B}_k C_{2,k}) \Sigma_{x,k} E_k^t V_k E_k \tag{4.35}$$

$$\begin{aligned} \hat{B}_k = &(\tau_k^{-1} H_{1,k} H_{2,k}^t + A_k S_k C_{2,k}^t) \\ &\cdot (I + \tau_k^{-1} H_{2,k} H_{2,k}^t + C_{2,k} S_k C_{2,k}^t)^{-1} \end{aligned} \tag{4.36}$$

where

$$V_k = (\tau_k^{-1}I - E_k\Sigma_{x,k}E_k^t)^{-1} > 0 \tag{4.37}$$

$$S_k = \Sigma_{x,k} + \Sigma_{x,k}E_k^tV_kE_k\Sigma_{x,k}. \tag{4.38}$$

In particular, the optimal filter is independent of $C_{1,k+1}$ (or the signal to be estimated). Further, the optimal filter given above preserves the structure in (4.33), i.e.,

$$\tilde{\Sigma}_{k+1}(\tau_k) = \begin{bmatrix} \Sigma_{x,k+1}(\tau_k) + \Sigma_{2,k+1}(\tau_k) & \Sigma_{2,k+1}(\tau_k) \\ \Sigma_{2,k+1}(\tau_k) & \Sigma_{2,k+1}(\tau_k) \end{bmatrix} \tag{4.39}$$

with

$$\begin{aligned}\Sigma_{x,k+1}(\tau_k) = B_kB_k^t + \tau_k^{-1}H_{1,k}H_{1,k}^t + A_kS_kA_k^t \\ - Z_k^t\Xi_k^{-1}Z_k\end{aligned} \tag{4.40}$$

and

$$\begin{aligned}&\Sigma_{x,k+1}(\tau_k) + \Sigma_{2,k+1}(\tau_k) \\ &\quad = B_kB_k^t + \tau_k^{-1}H_{1,k}H_{1,k}^t \\ &\qquad + A_k(\Sigma_{1,k}^{-1} - \tau_kE_k^tE_k)^{-1}A_k^t\end{aligned} \tag{4.41}$$

where

$$\Xi_k = I + \tau_k^{-1}H_{2,k}H_{2,k}^t + C_{2,k}S_kC_{2,k}^t \tag{4.42}$$

$$Z_k = \tau_k^{-1}H_{2,k}H_{1,k}^t + C_{2,k}S_kA_k^t. \tag{4.43}$$

Finally, we have

$$\begin{aligned}&\Sigma_{x,k+1}(\tau_k) > 0, \ \ \Sigma_{x,k+1}(\tau_k) \geq \Sigma_{x,k+1}, \ \ \Sigma_{2,k+1}(\tau_k) \geq 0 \\ &\forall\, 0 < \tau_k < \|E_k\Sigma_{1,k}E_k^t\|^{-1}.\end{aligned} \tag{4.44}$$

Proof: See Appendix A. ■

Problem P2:

Theorem 4.2: Under (4.33), the optimal solution to Problem 2 is given as in Theorem 4.1 with the optimal τ_k solving the following semidefinite program:

$$\begin{aligned}&\min L(C_{1,k+1}XC_{1,k+1}^t) \\ &\text{s.t.} \begin{bmatrix} X - B_kB_k^t & A_k & H_{1,k} \\ A_k^t & S_k^{-1} + C_{2,k}^tC_{2,k} & C_{2,k}^tH_{2,k} \\ H_{1,k}^t & H_{2,k}^tC_{2,k} & \tau_kI + H_{2,k}^tH_{2,k} \end{bmatrix} \geq 0 \\ &X = X^t, \qquad 0 \leq \tau_k \leq \|E_k^t\Sigma_{1,k}E_k\|^{-1}\end{aligned} \tag{4.45}$$

where S_k is given in (4.38) with

$$S_k^{-1} = \Sigma_{x,k}^{-1} - \tau_kE_k^tE_k. \tag{4.46}$$

Proof: See Appendix B. ■

Problem P3: Before we give the solution to P3, several key observations about Theorem 4.2 are needed.

- First, only $\Sigma_{x,k}$, rather than the whole $\tilde{\Sigma}_k$, is directly required for the filter design at time k. However, the term $\Sigma_{2,k}$ is used in constraining the range for τ_k.
- The optimal τ_k is solved independently of the optimal $\hat{A}_k$ and $\hat{B}_k$, although the latter depends on τ_k.
- The optimal X in (4.45) is indeed the optimal $\Sigma_{x,k+1}$. Further, if $\Sigma_{x,k}$ is replaced with any of its upper bound, the resulting optimal $\Sigma_{x,k+1}$ will be worsened.

These observations, together with the results in Section II, lead us to the main result of the paper.

Theorem 4.3: Let $\Sigma_0 > 0$ and $T \geq 0$. Denote

$$\begin{aligned}\tau &= [\tau_0, \ldots, \tau_T] \\ L_{e,k} &= C_{1,k}\Sigma_{x,k}C_{1,k}^t.\end{aligned}$$

Define $\Sigma_{x,0}(\tau) = \Sigma_0$. Let $\Sigma_{x,k+1}(\tau)$ and $\Sigma_{2,k+1}(\tau)$ be given as in (4.37), (4.38) and (4.40)–(4.43) for $k = 0, 1, \ldots$, except that $\Sigma_{x,k}$, $\Sigma_{2,k}$ and $\Sigma_{1,k}$ are replaced by $\Sigma_{x,k}(\tau)$, $\Sigma_{2,k}(\tau)$ and $\Sigma_{1,k}(\tau)$, respectively. Then, an upper bound for $L_{e,T+1}$ is given by

$$\overline{L}_{e,T+1} = \inf_{\tau\in\Omega_x} L\left(C_{1,T+1}\Sigma_{x,T+1}(\tau)C_{1,T+1}^t\right) \tag{4.47}$$

where

$$\Omega_x = \left\{\tau\colon 0 < \tau_k < \left\|E_k^t\Sigma_{1,k}(\tau)E_k\right\|^{-1}, \, k = 0, \ldots, T\right\} \tag{4.48}$$

In addition, define

$$\begin{cases} U_{x,0} = I; \\ \Pi_{x,0} = \Pi_{1,0} = \Sigma_0^{-1}; \\ W_{x,k} = \Pi_{x,k} - \tau_kU_{x,k}^tE_k^tE_kU_{x,k}; \\ W_{1,k} = \Pi_{1,k} - \tau_kU_{x,k}^tE_k^tE_kU_{x,k}; \\ \Pi_{x,k+1} = \text{diag}\left\{\begin{bmatrix} W_{x,k} & 0 \\ 0 & \tau_kI \end{bmatrix} + \begin{bmatrix} U_{x,k}^tC_{2,k}^t \\ H_{2,k}^t \end{bmatrix}\right. \\ \qquad\qquad \left. \cdot\, [C_{2,k}U_{x,k} \ \ H_{2,k}], \quad I_m\right\} \\ \Pi_{1,k+1} = \text{diag}\,\{W_{1,k}, \, \tau_kI, \, I_m\} \\ U_{x,k+1} = [A_kU_{x,k} \ \ H_{1,k} \ \ B_k]. \end{cases} \tag{4.49}$$

Note that $\Pi_{x,k}$ is affine in τ. Then, the optimal $\overline{L}_{e,T+1}$ can be found by solving the following semi-definite program:

$$\begin{aligned}\overline{L}_{e,T+1} &= \min L(C_{1,T+1}XC_{1,T+1}^t) \\ &\text{s.t.} \begin{bmatrix} X & U_{x,T+1} \\ U_{x,T+1}^t & \Pi_{x,T+1} \end{bmatrix} \geq 0 \\ W_{1,k} &\geq 0, \qquad k = 0, \ldots, T \\ X &= X^t, \qquad \tau \geq 0.\end{aligned} \tag{4.50}$$

Once the optimal τ is found, the optimal filter at time T is given as in Theorem 4.1, with $\Sigma_{x,T} = \Sigma_{x,T}(\tau)$.

Proof: See Appendix C. ■

Remark 4.1: Note that the optimal τ_k for each T may be different. Using a fixed τ_k may lead to conservative designs, but the optimal $\hat{A}_T$ and $\hat{B}_T$ does not explicitly depend on past filters, i.e., they depend on the optimal τ and the system data at T.

V. Recursive Robust Filter Design

There is one unpleasant feature about the solution in Theorem 4.3, that is, the size of the semidefinite program in (4.50) grows linearly in k. To avoid this, we propose a suboptimal solution, i.e., a recursive method that optimizes only a fixed number of most recent scaling parameters. The motivation for this approximate solution stems from a simple fact about Kalman filtering that the contribution of the initial covariance Σ_0 to the estimation error at time T decays as time evolves, provided that the augmented system (3.32) is asymptotically stable. The recursive method involves solving a semidefinite program of a constant size. Therefore, it is suitable for real-time applications where the information of the system dynamics (i.e., A_k, B_k, etc.) may not be available *a priori*.

The recursive algorithm given below is simply modified from Theorem 4.3.

Step 1) Let $N+1$ be the *window size* for recursion $N \geq 0$. For $0 \leq T \leq N$, apply (4.49) and (4.50).

Step 2) For $T > N$, still apply (4.49) and (4.50), but replace the constraint $W_{x,k} \geq 0,\ \forall k = 0, \ldots, T$ by $W_{x,k} \geq 0,\ \forall k = T-N, \ldots, T$ and reinitialize $U_{x,T-N} = I$ and $\Pi_{x,T-N} = (\Sigma^*_{T-N})^{-1}$, where Σ^*_{T-N} is the optimal $\Sigma_{T-N}(\tau)$ determined at $T-N$.

Remark 5.1: In general, all the matrices $\Phi_{x,k}$ and $U_{x,k}$, $k = T-N, \ldots, T+1$ need to be recomputed at each iteration $T > N$. This seems to require $O(N^2)$ amount of computation. We point out that it is possible to update $\Phi_{x,T+1}$ and $U_{x,T+1}$ using only a constant amount of computation, but this would require some messy notation and hence, will not be done here. However, for stationary systems where A_k, B_k, $C_{1,k}$, $C_{2,k}$, $H_{1,k}$, $H_{2,k}$, and E_k are constants (but F_k can still be time-varying), the update of $U_{x,T+1}$ and $\Phi_{x,T+1}$ is specially simple. This is because $U_{x,T+1}$ would remain constant and that $\Phi_{x,T+1}$ would have only one new element related to $\Sigma^*_{x,T-N}$.

VI. Example

To illustrate the results in this paper, we consider the following example, which has been used as a "benchmark" in [2], [5], [6]

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} 0 & -0.5 \\ 1 & 1+0.3\,\delta_k \end{bmatrix} x_k + \begin{bmatrix} -6 \\ 1 \end{bmatrix} w_k \\ y_k &= [-100 \ \ 10] x_k + v_k \\ z_k &= [1 \ \ 0] x_k \end{aligned} \tag{6.51}$$

where $|\delta_k| \leq 1$ is the uncertainty. We assume that the initial state covariance matrix $\Sigma_0 = I$.

To match the system description in (3.28), the uncertain term is represented by the matrices

$$H_1 = \begin{bmatrix} 0 \\ 3 \end{bmatrix}, \quad H_2 = 0, \quad E = [0 \ \ 0.03]. \tag{6.52}$$

Stationary filters are designed in [2], [5], and [6] to compare with the so-called "nominal" Kalman filter where the uncertainty is ignored. An infinite-horizon filter is used in [6] with guaranteed stability, which gives a great improvement over the nominal design. The design in [5] is based on finite horizon. In our setting, this design is similar to the recursive case with window size equal to one, except that the scaling parameter τ is preselected. The performance turns out to be superior to [6]. The design in [2] is similar to [5], except that the scaling parameter is optimized at each iteration using a semidefinite programming technique, yielding some small improvement over [5].

Design results using our new methods are given as follows.

1) (Design 1) For recursive design with window size equal to one, our design leads to a performance similar to [2], which is not surprising. Finite-horizon designs have the inherent instability problem, but this can usually be fixed by adding additional cost to the performance function.
2) (Designs 2 and 3) By increasing the window size, both the performance and stability can be improved dramatically with the tradeoff of more computation for the filter design.

Design 1: Recursive Design with $N = 1$: First, we design a filter using the given system data. The resulting filter turns out to be unstable. This demonstrates the inherent instability of finite-horizon designs. A main cause of instability is that the optimal τ_k may be such that it minimizes the cost function at $k+1$ but drives $\tilde{\Sigma}_{k+1}(\tau)$ too large, thus worsening the future costs. This problem has been recognized by other researchers. For example, [5] solves this problem by using a fixed (conservative) scaling parameter, whereas in [2], the covariance matrix $\tilde{\Sigma}_k$ is required to be bounded.

Alternatively, we treat the aforementioned problem by restricting the range of the scaling parameters as follows:

$$0 < \tau_k/\rho_k < \left\| E\Sigma_{1,k}(\tau)E^t \right\|^{-1}, \qquad \forall\, k \tag{6.53}$$

for some $0 < \rho_k < 1$. We also take an additional measure for improving stability by adding a term to the performance cost. Indeed, we take

$$C_1 = \begin{bmatrix} 1 & 0 \\ 0 & \varepsilon \end{bmatrix} \tag{6.54}$$

and the performance cost is trace$\{C_1\mathcal{E}(e_k e_k^t)C_1^t\}$. Effectively, the new term prevents $\tilde{\Sigma}_{k+1}(\tau)$ from getting too large, thus helping future costs. It is observed in the simulations that increasing ε can dramatically improve the stability and the steady-

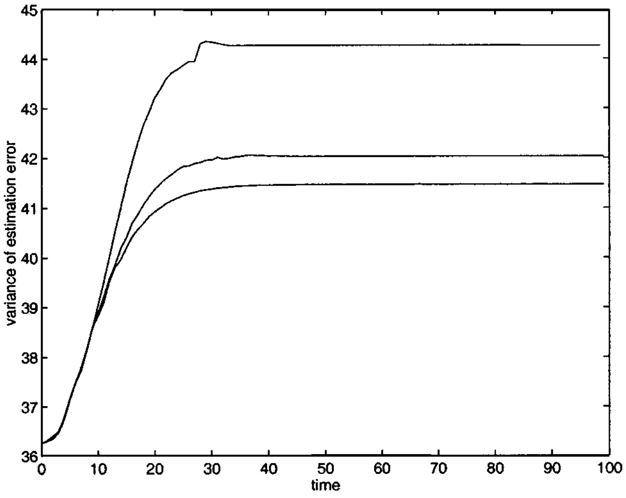


Fig. 1. Performances for recursive robust filter designs. Design 1: Top curve; Design 2: Middle curve; Design 3: Bottom curve.

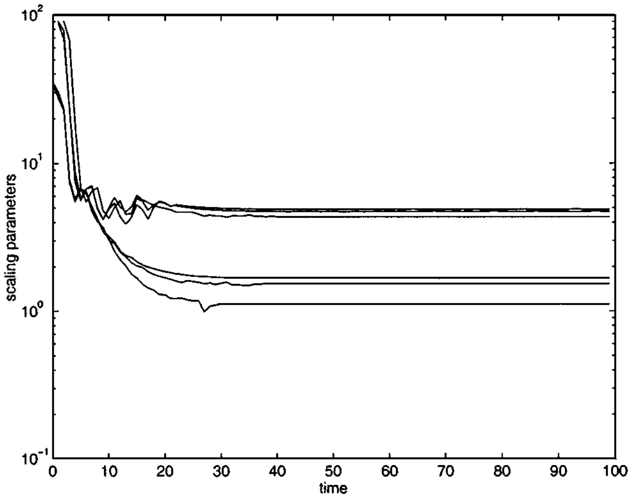


Fig. 2. Scaling parameters for recursive robust filter designs. Design 1: $\tau_{T,T}$—Bottom curve; Design 2: $\tau_{T,T-1}$—Third curve from the top; $\tau_{T,T}$—Second curve from the bottom; Design 3: $\tau_{T,T-2}$, $\tau_{T,T-1}$—Two top curves; $\tau_{T,T}$—Third curve from the bottom.

state performance with a minor tradeoff of the initial performance.

To demonstrate various recursive designs, we select $\rho = 0.7$ and $\varepsilon = 0.2$. The corresponding filter is stable and converges to a stationary one as $k \to \infty$, and it is given by (3.28) with

$$\hat{A}_k = \begin{bmatrix} 0 & -0.5166 \\ 1 & 1.0365 \end{bmatrix}, \quad \hat{B}_k = \begin{bmatrix} -0.003\,032 \\ -0.003\,330 \end{bmatrix}. \qquad (6.55)$$

The steady-state scaling parameter $\tau_{T,T} = 1.0981$. The results are shown in Figs. 1 and 2 as well as in Table I.

Design 2: Recursive Design with $N = 2$: Recall that with $N = 2$, two scaling parameters $\tau_{T,T-1}$ and $\tau_{T,T}$ are involved at each T. The first one is for estimating $\Sigma_{x,T}$ from $\Sigma_{x,T-1}$, and the second one is used to estimate $\Sigma_{x,T+1}$ and to design the filter.

TABLE I
STEADY-STATE PERFORMANCE COMPARISON

Filter	$\delta = -1$	$\delta = 0$	$\delta = 1$	Bound
Nominal Filter	551.2	36.0	8352.8	-
Robust Filter of [6]	64.0	61.4	64.4	98.7
Robust Filter of [5]	46.6	45.2	54.1	54.3
Robust Filter of [2]	50.8	49.4	53.5	N/A
Proposed Design 1	39.19	39.65	40.26	44.27
Proposed Design 2	38.11	38.66	39.40	42.03
Proposed Design 3	37.75	38.19	38.83	41.46

For the same fix of C_1 in (6.54), the steady-state filter has

$$\hat{A}_k = \begin{bmatrix} 0 & -0.5178 \\ 1 & 1.0391 \end{bmatrix}, \quad \hat{B}_k = \begin{bmatrix} -0.002\,561 \\ -0.004\,366 \end{bmatrix}. \qquad (6.56)$$

The steady-state scaling parameters are $\tau_{T,T-1} = 4.3966$ and $\tau_{T,T} = 1.5101$. The results are also shown in Figs. 1 and 2 as well as in Table I. Note that this filter gives a much better performance.

Design 3: Recursive Design with $N = 3$: The filter for $N = 3$ and the same fix of C_1 is also stable and has steady-state matrices

$$\hat{A}_k = \begin{bmatrix} 0 & -0.5544 \\ 1 & 1.1197 \end{bmatrix}, \quad \hat{B}_k = \begin{bmatrix} -0.002\,603 \\ -0.004\,272 \end{bmatrix}. \qquad (6.57)$$

Now, the scaling parameters converge to $\tau_{T,T-2} = 4.7172$, $\tau_{T,T-1} = 4.8874$, $\tau_{T,T} = 1.6699$. Once again, results are given in the same figures and table. However, the improvement in performance is virtually invisible. This means that $N = 2$ is sufficient for this example.

Remark 6.1: We would like to point out one interesting feature of our design. That is, for recursive designs without a relatively small window size, it is often quicker to compute the optimal scaling parameters using (4.47) rather than the semi-definite program (4.50). This is because (4.47) involves only a few scaling parameters and that the cost function is convex in τ. In fact, all the designs in this section are done using (4.47).

Remark 6.2: We suggested earlier that the motivation for re-optimizing the scaling parameters $\tau_{T-1}, \tau_{T-2}, \ldots$ is that their previous optimal values may not lead to an optimal solution at T. This fact is implicitly supported by Designs 2 and 3, where $\tau_{T,T-1}$ (the optimal τ_{T-1} at T) and $\tau_{T-1,T-1}$ (the optimal τ_{T-1} at $T-1$) are clearly different.

VII. CONCLUDING REMARKS

In this paper, we have proposed a new design technique for finite-horizon robust Kalman filters. This technique allows us to effectively treat systems with norm-bounded uncertainty blocks. The uncertainties are dealt with using the so-called S-Procedure, which yields a set of scaling parameters to optimize. The corresponding optimization problem is convex and can be solved either directly or via semidefinite program. A recursive design method that is mostly suitable to applications with nonstationary processes or signals is also presented. The proposed technique gives less conservative designs in comparison with existing techniques for robust Kalman filtering. This property has been demonstrated using an example.

Appendix A
Proof of Theorem 4.1

Let (4.33) hold. We first show that the optimal solution to Problem 1 is given by (4.34)–(4.38). Following Theorem 2.1, we consider the upper bound for $\tilde{\Sigma}_{k+1}$ given by

$$\tilde{\Sigma}_{k+1}(\tau_k) = \tilde{B}_k\tilde{B}_k^t + \tau_k^{-1}\tilde{H}_k\tilde{H}_k^t + \tilde{A}_k\Phi_k\tilde{A}_k^t \qquad \text{(A.58)}$$

where

$$\Phi_k = \tilde{\Sigma}_k + \tilde{\Sigma}_k\tilde{E}_k^t\tilde{V}_k\tilde{E}_k\tilde{\Sigma}_k \qquad \text{(A.59)}$$

with

$$\tilde{V}_k = (\tau_k^{-1}I - \tilde{E}_k\tilde{\Sigma}_k\tilde{E}_k^t)^{-1}. \qquad \text{(A.60)}$$

Note that

$$\tilde{V}_k = (\tau_k^{-1}I - E_k\Sigma_{1,\,k}E_k^t)^{-1} > 0. \qquad \text{(A.61)}$$

It follows from (A.58) that

$$\begin{aligned}
&\Sigma_{x,\,k+1}(\tau_k)\\
&\quad= [I \;\; -I]\tilde{\Sigma}_{k+1}(\tau)\begin{bmatrix} I \\ -I \end{bmatrix}\\
&\quad= B_kB_k^t + \hat{B}_k\hat{B}_k^t + \tau_k^{-1}(H_{1,\,k} - \hat{B}_kH_{2,\,k})\\
&\qquad\cdot (H_{1,\,k} - \hat{B}_kH_{2,\,k})^t + [A_k - \hat{B}_kC_{2,\,k} \;\; \hat{B}_kC_{2,\,k} - \hat{A}_k]\Phi_k\\
&\qquad\cdot \begin{bmatrix} A_k^t - C_{2,\,k}^t\hat{B}_k^t \\ C_{2,\,k}^t\hat{B}_k^t - \hat{A}_k^t \end{bmatrix}. \qquad \text{(A.62)}
\end{aligned}$$

Differentiating (A.62) with respect to $\hat{A}_k$ and $\hat{B}_k$ and setting the derivatives to zeros, we obtain

$$M_1 = [A_k - \hat{B}_kC_{2,\,k} \;\; \hat{B}_kC_{2,\,k} - \hat{A}_k]\Phi_k\begin{bmatrix} 0 \\ -I \end{bmatrix} = 0 \qquad \text{(A.63)}$$

and

$$\begin{aligned}
M_2 = {} & \hat{B}_k - \tau_k^{-1}(H_{1,\,k} - \hat{B}_kH_{2,\,k})H_{2,\,k}^t\\
& - [A_k - \hat{B}_kC_{2,\,k} \;\; \hat{B}_kC_{2,\,k} - \hat{A}_k]\Phi_k\begin{bmatrix} C_{2,\,k}^t \\ -C_{2,\,k}^t \end{bmatrix} = 0. \qquad \text{(A.64)}
\end{aligned}$$

Next, we want to compute optimal $\hat{A}_k$ from (A.63). To start with, we verify that

$$[I \;\; -I]\Phi_k\begin{bmatrix} 0 \\ I \end{bmatrix} = \Sigma_{x,\,k}E_k^t\tilde{V}_kE_k\Sigma_{2,\,k} \qquad \text{(A.65)}$$

and

$$[0 \;\; I]\Phi_k\begin{bmatrix} 0 \\ I \end{bmatrix} = \Gamma_k\Sigma_{2,\,k} \qquad \text{(A.66)}$$

where

$$\Gamma_k = I + \Sigma_{2,\,k}E_k^t\tilde{V}_kE_k. \qquad \text{(A.67)}$$

Substituting (4.34) and (A.65)–(A.66) into (A.63) yields

$$\Delta_k\Gamma_k\Sigma_{2,\,k} = (A_k - \hat{B}_kC_{2,\,k})\Sigma_{x,\,k}E_k^t\tilde{V}_kE_k\Sigma_{2,\,k} \qquad \text{(A.68)}$$

or equivalently

$$\Delta_k = (A_k - \hat{B}_kC_{2,\,k})\Sigma_{x,\,k}E_k^t\tilde{V}_kE_k\Gamma_k^{-1}. \qquad \text{(A.69)}$$

To show that (A.69) agrees with (4.35), we note that

$$\tilde{E}_k\tilde{\Sigma}_k\tilde{E}_k^t = E_k(\Sigma_{x,\,k} + \Sigma_{2,\,k})E_k^t. \qquad \text{(A.70)}$$

It follows that

$$\begin{aligned}
\tilde{V}_k &= (V_k^{-1} - E_k\Sigma_{2,\,k}E_k^t)^{-1}\\
&= V_k + V_kE_k(I - \Sigma_{2,\,k}E_k^tV_kE_k)^{-1}\Sigma_{2,\,k}E_k^tV_k \qquad \text{(A.71)}
\end{aligned}$$

with V_k given by (4.47). This implies

$$\begin{aligned}
&E_k^t\tilde{V}_kE_k\\
&\quad= E_k^tV_kE_k(I + (I - \Sigma_{2,\,k}E_k^tV_kE_k)^{-1}\Sigma_{2,\,k}E_k^tV_kE_k)\\
&\quad= E_k^tV_kE_k(I - \Sigma_{2,\,k}E_k^tV_kE_k)^{-1}. \qquad \text{(A.72)}
\end{aligned}$$

Subsequently

$$\begin{aligned}
&E_k^t\tilde{V}_kE_k(I + \Sigma_{2,\,k}E_k^t\tilde{V}_kE_k)^{-1}\\
&\quad= E_k^t\tilde{V}_kE_k(I + \Sigma_{2,\,k}E_k^tV_kE_k(I - \Sigma_{2,\,k}E_k^tV_kE_k)^{-1})^{-1}\\
&\quad= E_k^t\tilde{V}_kE_k(I - \Sigma_{2,\,k}E_k^tV_kE_k) = E_k^tV_kE_k. \qquad \text{(A.73)}
\end{aligned}$$

Hence, (A.69) is identical to (4.35).

Now we return to computing $\hat{B}_k$ from (A.69). Using (A.63) and (4.35), we rewrite (A.69) as

$$\begin{aligned}
&\hat{B}_k - \tau_k^{-1}(H_{1,\,k} - \hat{B}_kH_{2,\,k})H_{2,\,k}^t\\
&\qquad - (A_k - \hat{B}_kC_{2,\,k})[I \;\; -I - \Sigma_{x,\,k}E_k^tV_kE_k]\Phi_k\begin{bmatrix} C_{2,\,k}^t \\ 0 \end{bmatrix} = 0. \qquad \text{(A.74)}
\end{aligned}$$

To simplify the above, we verify that

$$[I \;\; -I]\Phi_k\begin{bmatrix} I \\ 0 \end{bmatrix} = \Sigma_{x,\,k} + \Sigma_{x,\,k}E_k^t\tilde{V}_kE_k(\Sigma_{x,\,k} + \Sigma_{2,\,k}) \qquad \text{(A.75)}$$

and

$$[0 \;\; I]\Phi_k\begin{bmatrix} I \\ 0 \end{bmatrix} = \Sigma_{2,\,k} + \Sigma_{2,\,k}E_k^t\tilde{V}_kE_k(\Sigma_{x,\,k} + \Sigma_{2,\,k}). \qquad \text{(A.76)}$$

From (A.73) and (A.75)–(A.76), we get

$$[I\ \ -I-\Sigma_{x,\,k}E_k^tV_kE_k]\Phi_k\begin{bmatrix} I \\ 0 \end{bmatrix}=S_k \tag{A.77}$$

where S_k is given in (4.38). Subsequently, (A.74) becomes

$$\hat{B}_k-\tau_k^{-1}(H_{1,\,k}-\hat{B}_kH_{2,\,k})H_{2,\,k}^t-(A_k-\hat{B}_kC_{2,\,k})S_kC_{2,\,k}^t=0 \tag{A.78}$$

which gives the optimal solution for $\hat{B}_k$ in (4.36). Substituting the optimal $\hat{A}_k$ and $\hat{B}_k$ into (A.62), we get (4.40).

The property in (4.39) is shown as follows:

$$\begin{aligned}
&[I\ \ -I]\tilde{\Sigma}_{k+1}(\tau)\begin{bmatrix} 0 \\ -I \end{bmatrix} \\
&\quad=\hat{B}_k\hat{B}_k^t-\tau_k^{-1}(H_{1,\,k}-\hat{B}_kH_{2,\,k})H_{2,\,k}^t\hat{B}_k^t \\
&\qquad+[A_k-\hat{B}_kC_{2,\,k}\ \ \hat{B}_kC_{2,\,k}-\hat{A}_k]\Phi_k\begin{bmatrix} -C_{2,\,k}^t\hat{B}_k^t \\ C_{2,\,k}^t\hat{B}_k^t-\hat{A}_k^t \end{bmatrix} \\
&\quad=M_1\hat{A}_k^t+M_2\hat{B}_k^t=0.
\end{aligned}$$

Equation (4.41) is verified as follows:

$$\begin{aligned}
&\Sigma_{x,\,k+1}(\tau_k)+\Sigma_{2,\,k+1}(\tau_k) \\
&\quad=[I\ \ 0]\tilde{\Sigma}_{k+1}(\tau_k)\begin{bmatrix} I \\ 0 \end{bmatrix} \\
&\quad=B_kB_k^t+\tau_k^{-1}H_{1,\,k}H_{1,\,k}^t+A_k[I\ \ 0]\Phi_k\begin{bmatrix} I \\ 0 \end{bmatrix}A_k^t \\
&\quad=B_kB_k^t+\tau_k^{-1}H_{1,\,k}H_{1,\,k}^t+A_k(\Sigma_{1,\,k}^{-1}-\tau_kE_k^tE_k)^{-1}A_k^t.
\end{aligned}$$

Finally, we focus on (4.44). Let $0<\tau_k<\|E_k^t\Sigma_{1,\,k}E_k\|^{-1}$. From (A.61), $\tilde{V}_k>0$. From Theorem 2.1, we have $\tilde{\Sigma}_{x,\,k+1}(\tau)\geq\tilde{\Sigma}_{x,\,k+1}\geq0$. Hence, $\Sigma_{x,\,k}(\tau)\geq\Sigma_{x,\,k}$ and $\Sigma_{2,\,k}(\tau)\geq0$. To show $\Sigma_{x,\,k+1}(\tau_k)>0$, we apply the Schur complement to (4.40), i.e., $\Sigma_{x,\,k+1}(\tau_k)>0$ iff

$$\begin{bmatrix} B_kB_k^t+\tau_k^{-1}H_{1,\,k}H_{1,\,k}^t+A_kS_kA_k^t & Z_k^t \\ Z_k & \Xi_k \end{bmatrix}>0$$

which can be rewritten as

$$\begin{bmatrix} A_k & H_{1,\,k} & B_k \\ C_{2,\,k} & H_{2,\,k} & 0 \end{bmatrix}\operatorname{diag}\{S_k,\,\tau_k^{-1}I,\,I\}\begin{bmatrix} A_k^t & C_{2,\,k}^t \\ H_{1,\,k}^t & H_{2,\,k}^t \\ B_k^t & 0 \end{bmatrix}>0$$

which is guaranteed due to $\Sigma_{x,\,k}>0$ and (3.30).

Appendix B
Proof of Theorem 4.2

Rewriting (4.40), we have

$$\Sigma_{x,\,k+1}(\tau_k)=B_kB_k^t+[A_k\ \ H_{1,\,k}]Y_k\begin{bmatrix} A_k^t \\ H_{1,\,k}^t \end{bmatrix}$$

where

$$\begin{aligned}
Y_k&=\begin{bmatrix} S_k & 0 \\ 0 & \tau_k^{-1}I \end{bmatrix}-\begin{bmatrix} S_k & 0 \\ 0 & \tau_k^{-1}I \end{bmatrix}\begin{bmatrix} C_{2,\,k}^t \\ H_{2,\,k}^t \end{bmatrix} \\
&\quad\cdot(I+C_{2,\,k}S_kC_{2,\,k}^t+\tau_k^{-1}H_{2,\,k}H_{2,\,k}^t)^{-1} \\
&\quad\cdot[C_{2,\,k}\ \ H_{2,\,k}]\begin{bmatrix} S_k & 0 \\ 0 & \tau_k^{-1}I \end{bmatrix} \\
&=\left(\begin{bmatrix} S_k^{-1} & 0 \\ 0 & \tau_kI \end{bmatrix}+\begin{bmatrix} C_{2,\,k}^t \\ H_{2,\,k}^t \end{bmatrix}[C_{2,\,k}\ \ H_{2,\,k}]\right)^{-1}.
\end{aligned}$$

The optimal τ_k is obtained by

$$\begin{aligned}
&\min L(C_{1,\,k+1}XC_{1,\,k+1}^t) \\
&\quad\text{s.t. } X\geq\Sigma_{x,\,k+1}(\tau_k) \\
&\quad X=X^t,\ 0\leq\tau_k\leq\|E_k\Sigma_{1,\,k}E_k^t\|^{-1}.
\end{aligned}$$

Using the Schur complement, the constraint $X\geq\Sigma_{x,\,k+1}(\tau_k)$ is equivalent to

$$\begin{bmatrix} X-B_kB_k^t & [A_k\ \ H_{1,\,k}] \\ \begin{bmatrix} A_k^t \\ H_{1,\,k}^t \end{bmatrix} & Y_k^{-1} \end{bmatrix}\geq0$$

which is the same as the first constraint in (4.45). Thus, the theorem has been proved.

Appendix C
Proof of Theorem 4.3

The first part of the theorem [(4.47)] follows from Theorem 4.1. To prove the second part [(4.50)], we first claim that it suffices to show

$$\begin{aligned}
\Sigma_{x,\,k}(\tau)&=U_{x,\,k}\Pi_{x,\,k}^{-1}U_{x,\,k}^t, \\
\Sigma_{1,\,k}(\tau)&=U_{x,\,k}\Pi_{1,\,k}^{-1}U_{x,\,k}^t,
\end{aligned}\qquad\forall k=0,\,\ldots,\,T;\ \tau\in\Omega_x. \tag{C.79}$$

Indeed, if (C.79) holds, it follows from the Schur complement that

$$\tau\in\Omega_x\Longleftrightarrow\tau>0,\ W_{x,\,k}>0,\qquad k=0\ldots,\,T. \tag{C.80}$$

Subsequently, the optimal X in (4.50) is achieved at

$$X=U_{T+1}\Pi_{x,\,T+1}^{-1}U_{T+1}^t.$$

To show (C.79), we apply reduction. Obviously, (C.79) holds for $k=0$ by definition. Suppose it holds for some k; then

$$\begin{aligned}
&U_{x,\,k+1}\Pi_{x,\,k+1}^{-1}U_{x,\,k+1}^t \\
&\quad=B_kB_k^t+[A_kU_{x,\,k}\ \ H_{1,\,k}]\hat{\Pi}_k^{-1}\begin{bmatrix} U_{x,\,k}^tA_k^t \\ H_{1,\,k}^t \end{bmatrix}
\end{aligned} \tag{C.81}$$

where

$$\hat{\Pi}_k=\begin{bmatrix} W_{x,\,k} & 0 \\ 0 & \tau_{x,\,k}I \end{bmatrix}+\begin{bmatrix} U_{x,\,k}^tC_{2,\,k}^t \\ H_{2,\,k}^t \end{bmatrix}[C_{2,\,k}U_{x,\,k}\ \ H_{2,\,k}].$$

We have

$$\hat{\Pi}_k^{-1}=\begin{bmatrix} W_{x,\,k}^{-1} & 0 \\ 0 & \tau_k^{-1}I \end{bmatrix}$$

$$
\begin{aligned}
&- \begin{bmatrix} W_{x,k}^{-1} & 0 \\ 0 & \tau_k^{-1} I \end{bmatrix} \begin{bmatrix} U_{x,k}^t C_{2,k}^t \\ H_{2,k}^t \end{bmatrix} \\
&\cdot (I + C_{2,k} U_{x,k} W_{x,k}^{-1} U_{x,k}^t C_{2,k}^t + \tau_k^{-1} H_{2,k} H_{2,k}^t)^{-1} \\
&\cdot [C_{2,k} U_{x,k} \;\; H_{2,k}] \begin{bmatrix} W_{x,k}^{-1} & 0 \\ 0 & \tau_k^{-1} I \end{bmatrix} \qquad \text{(C.82)}
\end{aligned}
$$

$$
\begin{aligned}
&U_{x,k} W_{x,k}^{-1} U_{x,k}^t \\
&\quad = U_{x,k} \Big(\Pi_{x,k}^{-1} - \Pi_{x,k}^{-1} U_{x,k} E_k^t \\
&\qquad\qquad \cdot (\tau_k^{-1} I - E_k U_{x,k} \Pi_{x,k}^{-1} U_{x,k}^t E_k^t)^{-1} \\
&\qquad\qquad \cdot E_k U_{x,k} \Pi_{x,k}^{-1} \Big) U_{x,k}^t \\
&\quad = \Sigma_{x,k}(\tau) - \Sigma_{x,k}(\tau) E_k^t \\
&\qquad \cdot \left(\tau_k^{-1} I - E_k \Sigma_{x,k}(\tau) E_k^t \right)^{-1} E_k \Sigma_{x,k}(\tau) = S_k. \qquad \text{(C.83)}
\end{aligned}
$$

Substituting (C.82) and (C.83) into (C.81), straightforward calculations yield

$$U_{k+1} \Pi_{x,k+1}^{-1} U_{k+1}^t = \Sigma_{x,k+1}(\tau)$$

Further

$$U_{x,k} W_{1,k}^{-1} U_{x,k}^t = \left(\Sigma_{1,k}^{-1}(\tau) - \tau_k E_k E_k^t \right)^{-1}. \qquad \text{(C.84)}$$

Subsequently

$$
\begin{aligned}
&U_{x,k+1} \Pi_{1,k+1}^{-1} U_{x,k+1}^t \\
&\quad = B_k B_k^t + \tau_k^{-1} H_{1,k} H_{1,k}^t A_k U_{x,k} W_{1,k}^{-1} U_{x,k}^t A_k^t \\
&\quad = \Sigma_{1,k+1}(\tau). \qquad \text{(C.85)}
\end{aligned}
$$

Thus, (C.79) is valid for all k.

ACKNOWLEDGMENT

The authors wish to thank Dr. L. Xie for his help in identifying a technical error in an earlier version of the paper.

REFERENCES

[1] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Philadelphia, PA: SIAM, 1994.

[2] L. Li, Z.-Q. Luo, K. M. Wong, and E. Bossé, "Semidefinite programming solutions to the robust state estimation problem with applications to multi-target tracking," McMaster Univ., Hamilton, ON, Canada, preprint.

[3] I. Petersen and D. C. McFarlane, "Optimal guaranteed cost filtering for uncertain discrete-time linear systems," *Int. J. Robust Nonlinear Contr.*, vol. 6, pp. 267–280, 1996.

[4] U. Shaked and C. E. de Souza, "Robust minimum variance filtering," *IEEE Trans. Signal Processing*, vol. 43, pp. 2474–2483, Nov. 1995.

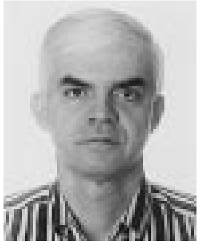
[5] Y. Theodor and U. Shaked, "Robust discrete-time minimum-variance filtering," *IEEE Trans. Signal Processing*, vol. 44, pp. 181–189, Feb. 1996.

[6] L. Xie, Y. C. Soh, and C. E. de Souza, "Robust Kalman filtering for uncertain discrete-time systems," *IEEE Trans. Automat. Contr.*, vol. 39, pp. 1310–1314, Nov. 1994.

Minyue Fu (SM'94) received the B.Sc. degree in electrical engineering from the China University of Science and Technology, Hefei, in 1982 and the M.S. and Ph.D. degrees in electrical engineering from the University of Wisconsin, Madison (UWM), in 1983 and 1987, respectively.

From 1983 to 1987, he held a teaching assistantship and a research assistantship at UWM. He worked as a Computer Engineering Consultant at Nicolet Instruments, Inc., Madison, during 1987. From 1987 to 1989, he served as an Assistant Professor with the Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI. For the summer of 1989, he was with the Universite Catholoque de Louvain, Louvain, Belgium, as a Maitre de Conferences Invited. He joined the Department of Electrical and Computer Engineering, the University of Newcastle, Australia, in 1989. Currently, he is an Associate Professor and Head of the Department. His main research interests include control systems, signal processing , and telecommunications. He is currently an Associate Editor of the *Journal of Optimization and Engineering.*

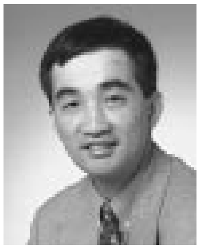
Dr. Fu has been an Associate Editor for the IEEE TRANSACTIONS ON AUTOMATIC CONTROL.



Carlos E. de Souza (M'81–SM'94) was born in João Pessoa, Brazil. He received the B.E. degree in electrical engineering (with highest honors) from the Universidade Federal de Pernambuco, Recife, Brazil, in 1976 and the Dr.-Ing. degree from the Université de Paris VI, Paris, France, in 1980.

From 1980 to 1984, he was a Lecturer with the Department of Electrical Engineering, Universidade Federal de Uberlndia, Uberlndia, Brazil. In 1985, he joined the Department of Electrical and Computer Engineering, University of Newcastle, Callaghan, Australia, as a Lecturer and became an Associate Professor in 1997. Since 1998, he has been a Professor with the Department of Systems and Control, Laboratório Nacional de Computação Científica (LNCC), Petrópolis, Brazil, and is the Director of the Research Center for Control of Dynamic Systems. During a sabbatical from 1992 to 1993, he was a Visiting Professor with the Laboratoire d'Automatique de Grenoble, Grenoble, France. He has also held numerous short-term visiting appointments at universities in several countries, including the United States, France, Switzerland, Israel, Australia, and Brazil. His research interests include robust signal estimation, optimal $\mathcal{H}_2$ and $\mathcal{H}_\infty$ estimation, robust control, optimal $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control, and time-delay systems. He has published over 170 research papers. He was Subject Editor for the *International Journal of Robust and Nonlinear Control* (IJRNC) and Guest Editor for the IJRNC Special Issue on $\mathcal{H}_\infty$ *and Robust Filtering.* He is currently a member of the Editorial Board of the IJRNC and Co-Chairman of the IFAC Technical Committee on Linear Systems.

Dr. de Souza is a Fellow of the Brazilian Academy of Sciences.



Zhi-Quan (Tom) Luo (M'90) was born in Nanchang, Jiangxi province, China. He received the B.Sc. degree in applied mathematics in 1984 from Peking University, Beijing, China. From 1984 to 1985, he studied at the Nankai Institute of Mathematics, Tianjin, China. He received the Ph.D. degree in operations research from the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, in 1989.

In 1989, he joined the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON, Canada, where he is now a Professor and Department Chair. His research interests lie in the union of large-scale optimization, information theory and coding, data communication, and signal processing. He is presently serving as an Associate Editor of the *Journal of Optimization Theory and Applications*, the *SIAM Journal on Optimization*, *Mathematics of Computation*, *Mathematics of Operations Research*, and *Optimization and Engineering*.

Prof. Luo is a member of SIAM and MPS. He is an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING.