A brief—The fast Kalman filter provides very quick convergence at the computational complexity of the same order that the LMS algorithm requires. Nevertheless, its performance is still unsatisfactory in system identification because the conventional fast Kalman filter fails to track time-varying impulse responses of FIR systems. The failure of tracking is due to the absence of system noise in the state-space model to be used. However, according to the derivation of the fast Kalman filter, it is difficult to theoretically introduce the term of system noise into the algorithm. In this paper, we overcome the difficulties, a new fast filtering algorithm, called a fast $H_\infty$ filter, is derived based on the $H_\infty$ theory.

Keywords—$H_\infty$ filter, fast algorithm, Kalman filter, robust estimation, system identification, LMS

I. INTRODUCTION

Efficient linear estimation algorithms have been developed in the past based mainly on the minimization of the $L_2$-norm of the estimation error [1], [2]. In such approaches, the Kalman filter has played a central role among the minimum variance estimators, and, fortunately, the fast algorithm, called the fast Kalman filter, has been developed for a specific state-space model [3], [4].

Recently, a measure which differs from the $L_2$-norm has been introduced for optimal estimation [5], [6]. This measure is the $H_\infty$-norm of the operation that relates the exogenous disturbances (the initial state and noises) to the estimation error, and it has been used successfully in optimal estimation. As a solution to the suboptimal $H_\infty$ estimation problem in which the $H_\infty$-norm is less than a prescribed positive value, an $H_\infty$ filter has been derived from the game theory approach which permits a consideration of finite-time $H_\infty$ filtering problems for time-varying state-space models [7]. The $H_\infty$ filtering problem has also been solved as a linear estimation in an indefinite-metric space, called the Krein space [8]. The $H_\infty$ filters have attracted much attention in the field of robust estimation [5]-[9], since they attempt to optimally estimate the required combinations of the states against the worst disturbances. However, their computational complexity becomes intractable as the size of the state vector grows large. To solve this problem, some fast array algorithms have been developed using unitary transformations [8]. Unfortunately, they have been applicable only for time-invariant state-space models.

Focusing on the fast Kalman filter, it provides extremely fast convergences for system identification of FIR systems at a reasonable computational requirement, whose order is equal to that of the LMS algorithm. Nevertheless, its performance is still unsatisfactory because the standard fast Kalman filter fails to track the time-varying impulse responses. The failure of tracking in the fast Kalman filter is due to the absence of system noise in the state-space model to be used. However, according to the derivation of the fast Kalman filter, it is essentially difficult to theoretically introduce the term of system noise into the algorithm.

In this paper, a new fast filtering algorithm, which is called a fast $H_\infty$ filter, is derived in the sense of $H_\infty$ optimization, which possesses the capability of successfully tracking the time-varying impulse responses on real-time with a reasonable computational cost. Furthermore, a condition for the fast $H_\infty$ filter to exist is given, which makes it easy to use the filter in practical situations.

II. MODIFIED $H_\infty$ FILTERS

For simplicity, we consider the following specific time-varying state-space model:

$$x_{k+1} = x_k + w_k, \quad w_k, x_k \in \mathcal{R}^N \quad (1)$$

$$y_k = H_k x_k + v_k, \quad y_k, v_k \in \mathcal{R} \quad (2)$$

$$z_k = H_k x_k, \quad z_k \in \mathcal{R}, \quad H_k \in \mathcal{R}^{1 \times N} \quad (3)$$

which is often used for echo cancellers and system identification although it appears to be a very specialized model, where $z_k$ is the signal to be estimated and $H_k$ has a shift-structure such that $H_{k+1}(i+1) = H_k(i)$. $H_k = [u_{k-1} u_{k-2} \cdots u_{k-(N-1)}]$. Furthermore, to simplify the presentation, we will confine our attention in this paper to the case of real-valued data.

Theorem 1: For such a state-space model, a (level-$\gamma$) modified $H_\infty$ filter to achieve e

$$\sup_{a_0, \{w_k\}, \{v_k\}} \frac{\sum_{i=0}^{k} ||e_{f,i}||^2 / \rho}{||x_0 - x_{0-1}||^2 + \sum_{i=0}^{k} ||u_i||^2 \Sigma_{w_i} + \sum_{i=0}^{k} ||v_i||^2 / \rho} < \gamma^2 \frac{2}{\rho}$$

is represented by

$$z_k[k] = H_k x_k[k] \quad (5)$$

$$x_{k+1}[k+1] = x_k[k] + K_{k,k+1}(y_{k+1} - H_k x_k[k]) \quad (6)$$
\[
K_{s,k+1} = \Sigma_{k+1|k} H_{k+1}^T \\
\cdot (H_{k+1} \Sigma_{k+1|k} H_{k+1}^T + \rho)^{-1} \\
\Sigma_{k+1|k} = \Sigma_{k+1|k-1} - \Sigma_{k+1|k-1}[H_k^T H_k^T] \\
\cdot R_{\gamma,k}^{-1} [H_k^T H_k^T] \Sigma_{k+1|k-1} + \Sigma_{w,k} 
\]

where

\[
e_{f,i} = \hat{z}_{i,i} - H_i x_i \\
R_{\gamma,k} = R_k + \left[ H_k^T H_k \right] \Sigma_{k+1|k-1}[H_k^T H_k^T] \\
R_k = \begin{bmatrix} \rho & 0 \\ 0 & -\rho \gamma_f^2 \end{bmatrix}, \Sigma_{w,k} = \gamma_f^{-2} \Sigma_{k+1|k} \]

and the parameter \( \gamma_f \) should be sufficiently adjusted small for robustness, as long as the following existence condition is satisfied:

\[
\Sigma_{k+1|k}^{-1} + H_k^T H_k > 0, \quad i = 0, \ldots, k. 
\]  

\textit{Proof:} Supposing that the weight parameter \( \rho \) is independent of \( \gamma_f \), we can easily derive the above modified \( H_\infty \) filter and the existence condition using the standard \( H_\infty \) estimation scheme [8].

It should be noted here that since \( \rho \) appears on the left-hand side of (4) and depends on \( \gamma_f \), the above algorithm is a modified version of the ordinary central \( H_\infty \) filter, i.e., the present suboptimal \( H_\infty \) estimation problem is of a non-standard type.

If the condition of (10) is not satisfied, then any \( H_\infty \) filter to achieve (4) no longer exists, i.e., it is not guaranteed that the estimation level (the \( H_\infty \)-norm on the time interval \([0, k]\)) is less than \( \gamma_f \) for any case. So, the determination of \( \gamma_f \) is very important in both existence and performance. The requirement of (10) for existence is also equivalent to the condition that the matrices \( R_k \) and \( R_{\gamma,k} \) have the same inertia [8].

As seen in (4), the \( H_\infty \) boundness is modified in advance to theoretically produce a nontstationary system noise with the covariance \( \Sigma_{w,k} \) in to the state-space model through the parameter \( \rho = 1 - \gamma_f^{-2} \), which can be regarded as a forgetting factor. The system noise \( w_k \), whose individual entries are not necessarily independent, makes it possible for the \( H_\infty \) filter to track the variations in the dynamic behavior of unknown systems. Note that the smaller \( \gamma_f \) is chosen the larger the effect of system noise becomes. Moreover, one has no use for the determination of \( \Sigma_{w,k} \). Indeed, the update of the error covariance matrix is carried out as

\[
\Sigma_{k+1|k} = \left( \Sigma_{k+1|k-1} - \Sigma_{k+1|k-1}[H_k^T H_k^T] \cdot R_{\gamma,k}^{-1} [H_k^T H_k^T] \right) / \rho. 
\]

The major computational burden of the \( H_\infty \) filter lies in the update of \( \Sigma_{k+1|k} \in \mathcal{R}^{N \times N} \) for \( K_{s,k+1} \) which requires proportional-to-\( N^2 \), i.e., \( O(N^2) \) arithmetic operations per time step, where \( N \) is the dimension of \( x_k \). Hence, on increasing the dimension of \( x_k \), the computation time required to run the \( H_\infty \) filter increases rapidly.

To overcome this drawback, we must develop a fast algorithm of the \( H_\infty \) filter.

III. FAST ALGORITHM OF MODIFIED \( H_\infty \) FILTERS

In this section, by taking into account the shifting property of the sequences of \( H_k \), the number of arithmetic operations required to compute the filter gain \( K_{s,k} \) for each time step is reduced from \( O(N^2) \) to \( O(N) \). Before the derivation of the fast algorithm, some preparations are given.

A. Preparations

An alternative framework is given to efficiently derive a fast algorithm of the modified \( H_\infty \) filter as follows. As the first step, one starts with recursively determining

\[
K_k = P_k C_k^T \in \mathcal{R}^{N \times 2} 
\]

instead of \( K_{s,k} \in \mathcal{R}^{N \times 1} \) in (7), where \( K_k \) is called a gain matrix hereafter and

\[
P_k = [O_k^T \Omega_k \Omega_k]^{-1} = [\sum_{i=1}^{k} \rho^{k-i} C_i^T W_i C_i]^{-1} \\
\Omega_k = \begin{bmatrix} \rho \Omega_{k-1} & 0 \\ 0 & W_k \end{bmatrix}, \quad \Omega_1 = W_1 \\
W_i = \rho R_{\gamma,i}^{-1} = \rho \begin{bmatrix} 1 & 0 \\ 0 & -\gamma_f^{-2} \end{bmatrix} \in \mathcal{R}^{2 \times 2} \\
O_k = \begin{bmatrix} C_1 \\ \vdots \\ C_k \end{bmatrix}, \quad C_i = \begin{bmatrix} H_i \\ H_i \end{bmatrix} \in \mathcal{R}^{2 \times N}. 
\]

This is because \( K_k \) is more convenient than \( K_{s,k} \) to derive the fast algorithm in an iterative manner without the Riccati recursion, and moreover, \( P_k \) completely agrees with \( \Sigma_{k+1|k} \) in (8) under a certain initial condition.

Now we can prove the following.

\textit{Lemma 1:} The matrix \( P_k \) defined by (13) satisfies the Riccati recursion of (8).

\textit{Proof:} Taking the inverse of \( P_k \), we have

\[
P_k^{-1} = \rho O_k^T \Omega_k \Omega_k^{-1} + C_k^T W_k C_k \\
= \rho P_{k-1}^{-1} + C_k^T W_k C_k. 
\]

Furthermore, using the matrix inversion lemma, we obtain the recursion of \( P_k \) as

\[
P_k = [\rho P_{k-1}^{-1} + [H_k^T H_k] W_k H_k]^{-1} \\
= \rho^{-1} P_{k-1}^{-1} \rho^{-1} P_{k-1}^{-1} [H_k^T H_k]^{-1} \\
\cdot (W_k^{-1} + [H_k H_k] \rho^{-1} P_{k-1}^{-1} [H_k^T H_k])^{-1} 
\]
\[
\rho P_k = P_{k-1} - P_{k-1} [H_k^T H_k]^T
\]
\[
\cdot (R_k + [H_k H_k^T] P_{k-1} [H_k^T H_k]^T)^{-1}
\]
\[
\cdot [H_k H_k^T] P_{k-1},
\]
\[
P_k = P_{k-1} - P_{k-1} [H_k^T H_k]^T
\]
\[
\cdot (R_k + [H_k H_k^T] P_{k-1} [H_k^T H_k]^T)^{-1}
\]
\[
\cdot [H_k H_k^T] P_{k-1} + \gamma_f^{-2} P_k.
\]  

(15)

The above lemmas mean that our problem of determining the filter gain \( K_{k+1} \) is equivalent to the determination of the gain matrix \( K_k \).

According to (14), \( P_k^{-1} \) is also expressed by
\[
P_k^{-1} = \frac{\rho}{\gamma_0} I + \sum_{i=1}^k \rho^{i-1} C_i^T W_i C_i
\]
for \( k > 0 \) when it is set that \( P_0 = \gamma_0 I \). It follows that \( P_k \) in (14) tends to \( P_k \) in (13) when \( \gamma_0 \) approaches infinity \( (\gamma_0 \to \infty) \).

For further convenience, defining \( Q_k = P_k^{-1} \) and returning to (12), we find that our objective is to recursively determine the gain matrix \( K_k \) which satisfies
\[
Q_k K_k = C^T_k
\]

(20)

at the computational burden of \( O(N) \).

In the following, the fast calculation of \( K_k \) is given using the shifting property of \( C_k^T \in R^{N \times N} \), requiring \( O(N) \) arithmetic operations per time step.

\section*{B. Fast Calculation of the Gain Matrix}

From lemma 1 and lemma 2, our aim can be replaced by determining the gain matrix \( K_k \). The following lemma gives a solution to the problem.

**Lemma 3:** The gain matrix \( K_k \) is updated as
\[
K_{k+1} = m_k - B_k F_k^{-1} \mu_k \in R^{N \times 2}
\]

where \( m_k \in R^{N \times 2} \) and \( \mu_k \in R^{1 \times 2} \) are obtained, through the partition of \( K_k = \tilde{Q}_k^{-1} C_k \) from
\[
\begin{bmatrix}
\mu_k \\

\end{bmatrix}
\]

(22)

and the matrices \( A_k \in R^{N \times 1} \), \( S_k \in R \), and \( B_k F_k^{-1} \in R^{N \times 1} \) are given by Lemma 4 and Lemma 5.

**Proof:** Now, we assume that \( K_{i+1} = 0, \ldots, k \) has been given, and then solve the problem of determining \( K_{k+1} \), defined by
\[
Q_{k+1} K_{k+1} = C^T_{k+1}
\]

(23)

To take advantage of the shifting property of \( C_k \), we introduce
\[
\bar{C}_k^T = \begin{bmatrix}
C_k^T & C_{k+1}^T \\

\end{bmatrix} \in R^{(N+1) \times 2}
\]

(24)

and
\[
\bar{Q}_k = \sum_{i=1}^i \rho^{-i} \bar{C}_i^T W_i \bar{C}_i \in R^{(N+1) \times (N+1)}
\]

(25)

which is expressed in recursive fashion by
\[
\bar{Q}_k = \rho \bar{Q}_{k-1} + \bar{C}_k^T W_k \bar{C}_k
\]

(26)

Furthermore, it is partitioned as
\[
\bar{Q}_k = \begin{bmatrix}
M_k & T_k \\

\end{bmatrix} \in R^{(N+1) \times (N+1)}
\]

(27)

and using \( G_k = (\rho + H_k P_{k-1} H_k^T)/(1 + H_k P_{k-1} H_k^T) \) and \( H_k K_k = H_k P_{k-1} H_k^T/(1 + H_k P_{k-1} H_k^T) \), we can obtain from the first bloc column of \( K_k \) the transformation of (16).
With this notation, the equations (20) and (23) are contained in the following:

\[
\tilde{Q}_k \begin{bmatrix} 0 \\ K_k \end{bmatrix} = \begin{bmatrix} \alpha^T_k \\ C_k^T \end{bmatrix} = \bar{C}_k^T + \begin{bmatrix} \alpha^T_k - c_k^T \\ 0 \end{bmatrix} \tag{28}
\]

\[
\tilde{Q}_k \begin{bmatrix} K_{k+1} + 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \alpha^{T}_{k+1} \\ \beta_k \end{bmatrix} = \bar{C}_k^T + \begin{bmatrix} 0 \\ \beta_k - c_k^{T-N} \end{bmatrix} \tag{29}
\]

where

\[
\alpha^T_k = T_k^T K_k \in \mathcal{R}^{1 \times 2}, \quad \beta_k = T_k K_{k+1} \in \mathcal{R}^{1 \times 2}.
\]

In view of these expressions, it seems more reasonable to find \( \tilde{K}_k \in \mathcal{R}^{(N+1) \times 2} \) which satisfies

\[
\tilde{Q}_k \tilde{K}_k = \bar{C}_k^T \tag{30}
\]

as an intermediate step before determining \( K_k \), where

\[
\tilde{K}_k = \begin{bmatrix} k_{k+1}^T K_k^T \end{bmatrix} = \begin{bmatrix} k_{k+1}^T \end{bmatrix} \tag{31}
\]

To do so, using

\[
C_k^T = \tilde{Q}_k \begin{bmatrix} 0 \\ K_k \end{bmatrix} - \begin{bmatrix} \alpha^T_k - c_k^T \\ 0 \end{bmatrix} \tag{32}
\]

which is obtained from (28), we arrange \( \tilde{K}_k \in \mathcal{R}^{(N+1) \times 2} \) as

\[
\tilde{K}_k = \begin{bmatrix} m_k \\ \mu_k \end{bmatrix}
\]

\[
= Q_k^{-1} C_k^T = \begin{bmatrix} 0 \\ K_k \end{bmatrix} - \tilde{Q}_k^{-1} \begin{bmatrix} \alpha^T_k - c_k^T \\ 0 \end{bmatrix}
\]

\[
= \begin{bmatrix} 0 \\ K_k \end{bmatrix} - \begin{bmatrix} S_k^{-1} \end{bmatrix} \begin{bmatrix} \alpha^T_k - c_k^T \end{bmatrix} \tag{33}
\]

where \( \tilde{K}_k \) is partitioned with \( m_k \in \mathcal{R}^{N \times 2} \) and \( \mu_k \in \mathcal{R}^{1 \times 2} \). Note that \( \alpha^T_k - c_k^T = -(c_k^T + \alpha^T_k C_k^T) \) holds. Also, \( \tilde{Q}_k \) is invertible, and \( A_k \in \mathcal{R}^{N \times 1} \) and \( S_k \in \mathcal{R} \) satisfy

\[
\tilde{Q}_k \begin{bmatrix} 1 \\ A_k \end{bmatrix} = \begin{bmatrix} S_k \\ 0 \end{bmatrix} \left( \begin{bmatrix} 1 \\ A_k \end{bmatrix} S_k^{-1} = \tilde{Q}_k^{-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) \tag{34}
\]

whose lower block gives \( T_k \tilde{Q}_k + Q_k A_k = 0 \) or \( T_k^T = -A_k^T \tilde{Q}_k^T \).

To eliminate \( \mu_k \) in (33) without affecting the upper part of \( \tilde{C}_k^T \), introducing the matrices \( B_k \in \mathcal{R}^{N \times 1} \) and \( F_k \in \mathcal{R} \) such that

\[
\tilde{Q}_k \tilde{B}_k = \tilde{Q}_k \begin{bmatrix} B_k \\ F_k \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left( \tilde{B}_k = \begin{bmatrix} B_k \\ F_k \end{bmatrix} \right) \tag{35}
\]

and then subtracting \( \tilde{B}_k F_k^{-1} \mu_k \) from the partition of \( \tilde{K}_k \) in (33), we have

\[
\tilde{K}_k - \tilde{B}_k F_k^{-1} \mu_k = \begin{bmatrix} m_k \\ \mu_k \end{bmatrix} - \begin{bmatrix} B_k F_k^{-1} \\ 1 \end{bmatrix} \mu_k
\]

\[
= \begin{bmatrix} m_k - B_k F_k^{-1} \mu_k \\ 0 \end{bmatrix} \tag{36}
\]

Moreover, multiplying the left-hand side of (36) by \( \tilde{Q}_k \) from the left, we can arrange it as

\[
\tilde{Q}_k (\tilde{K}_k - \tilde{B}_k F_k^{-1} \mu_k) = \bar{Q}_k \tilde{K}_k - \tilde{Q}_k \tilde{B}_k F_k^{-1} \mu_k = \bar{C}_k^T - \begin{bmatrix} 0 \\ 1 \end{bmatrix} F_k^{-1} \mu_k
\]

\[
= \bar{C}_k^T - \begin{bmatrix} 0 \\ F_k^{-1} \mu_k \end{bmatrix} \tag{37}
\]

By substituting (36) into the above left-hand side, the equation (30) of interest is presented by

\[
\tilde{Q}_k (\tilde{K}_k - \tilde{B}_k F_k^{-1} \mu_k)
\]

\[
= \bar{C}_k^T - \begin{bmatrix} 0 \\ F_k^{-1} \mu_k \end{bmatrix}
\]

\[
= \begin{bmatrix} Q_{k+1} \end{bmatrix} \begin{bmatrix} T_k \\ M_k \end{bmatrix} \begin{bmatrix} m_k - B_k F_k^{-1} \mu_k \\ 0 \end{bmatrix}
\]

\[
= \begin{bmatrix} C_k^{T+1} \\ c_k^{-N} \end{bmatrix} + \begin{bmatrix} 0 \\ -F_k^{-1} \mu_k \end{bmatrix} \tag{38}
\]

which is the same form as (29). The upper block of (38) leads to

\[
Q_{k+1} (m_k - B_k F_k^{-1} \mu_k) = C_k^{T+1} \tag{39}
\]

With comparison of (23) and (39), we reach the update equation of the gain matrix \( K_k \).

\[\blacksquare\]

C. Determination of Unknown Matrices Appeared in the Update of Gain Matrix

It now remains to determine the matrices \( A_k \), \( S_k \), and \( D_k = B_k F_k^{-1} \), which are defined by (34) and (35). The determination of these matrices will be also carried out by means of induction.

Lemma 1: The matrices \( A_k \) and \( S_k \) are recursively computed as

\[
A_k = A_{k-1} - K_k W_k [c_k + C_k A_{k-1}] \in \mathcal{R}^{N \times 1} \tag{40}
\]

\[
S_k = \rho S_{k-1} + [c_k^T + A_k^T C_k^T] W_k [c_k + C_k A_{k-1}] \in \mathcal{R} \tag{41}
\]

where \( A_{-1} = 0 \) and \( S_{-1} = \rho/\rho_0 \).

Proof: Now, recalling that the matrices \( A_{k-1} \) and \( S_{k-1} \), by definition, satisfy

\[
\tilde{Q}_{k-1} \begin{bmatrix} 1 \\ A_{k-1} \end{bmatrix} = \begin{bmatrix} S_{k-1} \\ 0 \end{bmatrix} \tag{42}
\]

and using

\[
\tilde{Q}_k = \rho \tilde{Q}_{k-1} + C_k^T W_k C_k \tag{43}
\]

we obtain

\[
\tilde{Q}_k \begin{bmatrix} 1 \\ A_{k-1} \end{bmatrix} = \rho \tilde{Q}_{k-1} \begin{bmatrix} 1 \\ A_{k-1} \end{bmatrix} + C_k^T W_k [c_k + C_k A_{k-1}]
\]

\[
= \rho S_{k-1} + C_k^T \begin{bmatrix} c_k^T \\ C_k^T \end{bmatrix} W_k [c_k + C_k A_{k-1}] \tag{44}
\]
Whereas, multiplying both sides of (28) by $W_k[c_k + C_k A_{k-1}]$ yields
\[
\tilde{Q}_k \begin{bmatrix} 1 \\ K_k \end{bmatrix} W_k[c_k + C_k A_{k-1}] = \begin{bmatrix} \alpha_k^T \\ C_k^T \end{bmatrix} W_k[c_k + C_k A_{k-1}].
\] (45)

Subtracting (45) from (44) on both sides, we find that
\[
\tilde{Q}_k \begin{bmatrix} 1 \\ A_{k-1} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} W_k[c_k + C_k A_{k-1}] = \begin{bmatrix} \alpha_k^T \\ C_k^T \end{bmatrix} W_k[c_k + C_k A_{k-1}]
\]
\[
\tilde{Q}_k \begin{bmatrix} A_{k-1} - K_k W_k[c_k + C_k A_{k-1}] \\ 0 \end{bmatrix} = \left[ \begin{array}{c} \rho S_{k-1} \\
+ [C_k^T - \alpha_k^T] W_k[c_k + C_k A_{k-1}] \end{array} \right].
\] (46)

Then, comparing the above equation with (34), we obtain the recursions of (40) and (41) because $\alpha_k^T = T_k^T K_k = -A_k^T C_k$ holds. ⊣

Next, the matrix $D_k = B_k F_k^{-1}$ is determined instead of $B_k$ and $F_k$ since they are used only in such a combination.

**Lemma 5:** The matrix $D_k = B_k F_k^{-1}$ is recursively determined as
\[
D_k = [D_{k-1} - m_k W_k \eta_k][1 - \mu_k W_k \eta_k]^{-1} \in \mathbb{R}^{N \times N},
\] (47)
and $F_k$ is evolved by
\[
F_k = F_{k-1}[1 - \mu_k W_k \eta_k]/\rho \in \mathbb{R}
\] (48)
where $\eta_k = \tilde{C}_k \tilde{D}_{k-1} = e_{k-N} + C_{k+1} D_{k-1}$ and $D_{k-1} = 0$.

**Proof:** To update $\tilde{B}_k$ and $F_k$, using
\[
\tilde{Q}_{k-1} \tilde{B}_{k-1} = \tilde{Q}_{k-1} \begin{bmatrix} B_{k-1} \\ F_{k-1} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}
\] (49)
and (43), we find that
\[
\tilde{Q}_k \tilde{B}_{k-1} = \rho \tilde{Q}_{k-1} \tilde{B}_{k-1} + C_k^T W_k \tilde{C}_k \tilde{B}_{k-1}
\]
\[
= \rho \begin{bmatrix} 0 \\ 1 \end{bmatrix} + C_k^T W_k \tilde{C}_k \tilde{B}_{k-1}.
\] (50)

To change the above into the same form as (49), subtracting $\tilde{C}_k^T W_k \tilde{C}_k \tilde{B}_{k-1}$ from both sides of (50), we eliminate the last term on the above right-hand side as
\[
\tilde{Q}_k \tilde{B}_{k-1} = \tilde{C}_k^T W_k \tilde{C}_k \tilde{B}_{k-1}
\]
\[
= \tilde{Q}_{k-1} \tilde{B}_{k-1} - \tilde{Q}_{k-1} K_k W_k \tilde{C}_k \tilde{B}_{k-1} = \rho \begin{bmatrix} 0 \\ 1 \end{bmatrix},
\]
\[
\tilde{Q}_k \begin{bmatrix} \tilde{B}_{k-1} - K_k W_k \tilde{C}_k \tilde{B}_{k-1} \\ 0 \end{bmatrix} = \rho \begin{bmatrix} 0 \\ 1 \end{bmatrix}.
\] (51)

Then, comparing the last equation of (51) with (35), we achieve the recursion of $\tilde{B}_k$:
\[
\tilde{B}_k = (\tilde{B}_{k-1} - K_k W_k \tilde{C}_k \tilde{B}_{k-1})/\rho
\] (52)
which updates $B_k$ and $F_k$ iteratively.

However, since the rows of $\tilde{B}_k$ are used only in the combination such that $D_k = B_k F_k^{-1} \in \mathbb{R}^{N \times 1}$, it is probably more convenient to rewrite (35) and (52) in terms of
\[
D_k = B_k F_k^{-1},
\]
\[
\tilde{Q}_k \tilde{D}_k = \tilde{Q}_k B_k F_k^{-1} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} F_k^{-1},
\]
\[
\tilde{Q}_k \begin{bmatrix} D_k \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ F_k^{-1} \end{bmatrix}.
\] (54)

Then, multiplying both sides of (51) by $F_k^{-1}$, we have
\[
\tilde{Q}_k [B_{k-1} F_k^{-1} - K_k W_k \tilde{C}_k \tilde{B}_{k-1} F_k^{-1}] = \tilde{Q}_{k-1} [D_{k-1} - K_k W_k \tilde{C}_k \tilde{D}_{k-1}]
\]
\[
= \begin{bmatrix} 0 \\ \rho F_{k-1}^{-1} \end{bmatrix}.
\] (55)

Furthermore, using $\tilde{D}_{k-1} = \tilde{B}_{k-1} F_k^{-1}$, it is arranged as
\[
\tilde{Q}_k \begin{bmatrix} \tilde{D}_{k-1} - m_k W_k \tilde{C}_k \tilde{D}_{k-1} \\ 1 - m_k W_k \tilde{C}_k \tilde{D}_{k-1} \end{bmatrix} = \begin{bmatrix} 0 \\ \rho F_{k-1}^{-1} \end{bmatrix}.
\] (56)

Consequently, multiplying the last equation of (56) with $[1 - m_k W_k \tilde{C}_k \tilde{D}_{k-1}]^{-1}$ from the right, we find that
\[
\tilde{Q}_k \begin{bmatrix} m_k W_k \tilde{C}_k \tilde{D}_{k-1} \\ \rho F_{k-1}^{-1} [1 - m_k W_k \tilde{C}_k \tilde{D}_{k-1}]^{-1} \end{bmatrix} = \begin{bmatrix} 0 \\ \rho F_{k-1}^{-1} [1 - m_k W_k \tilde{C}_k \tilde{D}_{k-1}]^{-1} \end{bmatrix}.
\]

Comparing the above equation with (54) yields the update formulas of $D_k$ and $F_k$.

**D. A Fast $H_\infty$ Filter**

As a conclusion, we may summarize a fast algorithm of the $H_\infty$ filter.

**Theorem 2:** A fast algorithm of the modified $H_\infty$ filter is carried out with computational complexity of $O(N)$ per time step as follows.

**[Step 0]** Take the initial conditions for the recursions as
\[
K_0 = 0, A_{-1} = 0, S_{-1} = \rho/\sigma_0, D_{-1} = 0, \tilde{x}_{0:0} = 0
\]
where $\sigma_0$ is a sufficiently large positive number.
\[ \text{Step 1] } \text{ Determine } A_k \text{ and } S_k \text{ recursively as} \\
\begin{align*}
\tilde{e}_k &= c_k + C_k A_{k-1} \\
A_k &= A_{k-1} - K_k W_k \tilde{e}_k \\
\tilde{e}_k &= c_k + C_k A_k \\
S_k &= \rho S_{k-1} + e_k^T W_k \tilde{e}_k \\
\end{align*} \\
\in \mathcal{R}^{2 \times 1}, \quad \mathcal{R}^{N \times 1}, \quad \mathcal{R}^{2 \times 1}, \quad \mathcal{R}.
\]

\[ \text{Step 2] } \text{ Calculate } \hat{K}_k \text{ as} \\
\hat{K}_k &= \begin{bmatrix} S_k^{-1} e_k^T \\
K_k + A_k S_k^{-1} e_k^T \end{bmatrix} \\
\in \mathcal{R}^{(N+1) \times 2}.
\]

\[ \text{Step 3] } \text{ Partition } \hat{K}_k \text{ as} \\
\hat{K}_k &= \begin{bmatrix} m_k \\
\mu_k \end{bmatrix}, \quad m_k \in \mathcal{R}^{N \times 2}, \quad \mu_k \in \mathcal{R}^{1 \times 2}.
\]

\[ \text{Step 4] } \text{ Determine } D_k = B_k F_k^{-1} \text{ and then obtain the} \\
\text{ filter gain } K_{s,k+1} \text{ at next time step } k+1, \text{ through the} \\
\text{gain matrix } K_{s,k+1}, \text{ as} \\
\eta_k &= c_k + C_k + D_{k-1} \\
D_k &= [D_{k-1} - m_k W_k \eta_k][1 - \mu_k W_k \eta_k]^{-1} \\
K_{s,k+1} &= m_k - D_k \mu_k \\
\hat{K}_{s,k+1}(i) &= \rho \hat{K}_{s,k+1}(i,1), \quad i = 1, \ldots, N \\
K_{s,k+1} &= C_{s,k+1}^T K_{s,k+1}, \quad G_{k+1} = \rho + \gamma^2 H_{k+1} K_{s,k+1}
\]

\[ \text{where } \eta_k \in \mathcal{R}^{2 \times 1}, \quad D_k \in \mathcal{R}^{N \times 1}, \quad K_{s,k+1} \in \mathcal{R}^{N \times 2}, \quad K_{s,k+1} \in \mathcal{R}^{N \times 1}, \quad 0 < \rho = 1 - \gamma^2 \leq 1, \text{ and } \gamma_f > 1.
\]

\[ \text{Step 5] } \text{ Update the filter equation of the } H_\infty \text{ filter as} \\
x_{k+1|k+1} = x_{k|k} + K_{s,k+1}(y_{k+1} - H_{k+1} x_{k|k}).
\]

\[ \text{Step 6] } \text{ Increase time step } k \text{ } (k \to k + 1), \text{ and return to Step 1.} \\
\]

\[ \text{Proof: It follows from lemma 1 to lemma 5 that the} \\
\text{ above results are immediately proved.} \]

\[ \text{The derived fast algorithm has the following good asymptotical property.} \\
\]

\[ \text{Corollary 1: The fast algorithm when } \gamma_f = \infty \text{ agrees with that of} \\
\text{the Kalman filter [3].} \]

\[ \text{Proof: Setting } \gamma_f = \infty, \text{ we find that the statement} \\
\text{ is clear from the } H_\infty \text{ criterion in (4) or the algorithm described} \\
\text{ in Theorem 2.} \]

This means that the fast algorithm derived in this section \\
\text{ could be regarded as an extended version of the fast} \\
\text{ Kalman filter in the sense of } H_\infty \text{ optimization.} \]

E. Existence of Fast } H_\infty \text{ Filter

In general, it requires a considerably computational cost \\
\text{ to check the positivity condition of (10), especially for large} \\
\text{ } N. \text{ Therefore, it may be more favorable in this case to use the} \\
\text{ alternative existence condition that the matrices } R_k \\
\text{ and } R_{e,k} \text{ have the same inertia [8]. Here, what one means} \\
\text{ by the inertia of a matrix is the number of its positive,} \\
\text{ negative, and zero eigenvalues.} \\
\text{ Fortunately the inertia condition for the fast } H_\infty \text{ filter} \\
\text{ to exist can be reduced to a simpler form.} \\
\]

\[ \text{Lemma 6: The following condition allows a fair judgment} \\
\text{ for existence of the fast } H_\infty \text{ filter at the computational} \\
\text{ requirement of } O(N) \text{ per time step.} \\
\]

\[ \begin{align*}
-\tilde{e}_i + \rho \gamma_f^2 &> 0, \quad i = 0, \ldots, k \\
\theta &= 1 - \gamma_f^2, \quad \tilde{e}_i = \frac{H_k K_i}{1 - H_k K_i}
\end{align*} \]

\[ \text{Proof: Solving the characteristic equation } |I - R_{e,k}| = 0 \text{ of the } 2 \times 2 \text{ matrix } R_{e,k}, \text{ we obtain the eigenvalues } \lambda_i \text{ of } R_{e,k} \text{ as} \\
\lambda_i = \frac{\Phi \pm \sqrt{\Phi^2 - 4 \rho \theta H_k \Sigma_k |k-1| H_k^T + 4 \rho^2 \gamma_f^2}}{2}
\]

\[ \text{where } \Phi = 2 H_k \Sigma_k |k-1| H_k^T + \rho \theta \text{ and } \theta = 1 - \gamma_f^2. \text{ If and only if} \\
\text{ if } -4 \rho \theta H_k \Sigma_k |k-1| H_k^T + 4 \rho^2 \gamma_f^2 \geq 0, \text{ one of the two eigenvalues of } R_{e,k} \text{ becomes positive and the other negative,} \\
\text{ so that } R_k \text{ and } R_{e,k} \text{ have the same inertia. Hence, using} \\
H_k \Sigma_k |k-1| H_k^T = \frac{H_k K_i}{1 - H_k K_i}, \text{ we attain the more simplified inertia condition of (57), where the calculation of } H_k K_i \text{ requires } O(N) \text{ multiplications per time step.} \]

IV. CONCLUSIONS

A fast algorithm of } H_\infty \text{ filters with a modified } H_\infty \text{ boundness has been successfully derived in a recursive fashion} \\
\text{ using the shifting property of the sequences of the observation matrix } H_k. \text{ In addition, it was clarified that the computational} \\
\text{ complexity of the fast } H_\infty \text{ filter is of } O(N) \text{ per time step which is equal to that of the fast Kalman filter} \\
\text{ and the LMS algorithm, dramatically reducing the complexity, } O(N^2), \text{ of the corresponding } H_\infty \text{ filter. Furthermore,} \\
\text{ a simple method to judge the validity of the fast } H_\infty \text{ filter for a given } \gamma_f \text{ was developed, which requires } O(N) \text{ arithmetic operations per time step.} \]

REFERENCES