H∞ Kalman Filtering for Rectangular Descriptor Systems with Unknown Inputs

Chien-Shu Hsieh

Abstract—This paper considers H∞ filtering for rectangular descriptor systems with unknown inputs that affect both the system and the output. An optimal H∞ filter is developed based on the maximum likelihood descriptor Kalman filtering (DKF) method. The developed H∞ filter serves as a unified solution to solve H∞ and Kalman filtering for descriptor systems and standard systems with or without unknown inputs, which, however, may also suffer from computational complexity problem. Three computationally efficient alternatives to the developed H∞ filter are further proposed based on a novel matrix transformation and the recently proposed gain-covariance matrix (GCM) concept to remedy the computational problem. Simulation results are given to illustrate the usefulness of the proposed results.

I. INTRODUCTION

Unknown input filtering (UIF) serves as a useful technique to solve many practical state estimation problems that often arise in systems subject to disturbances, modeling errors, system uncertainties, and reduced-order filtering (see [1] and the references therein). As a most general case, there is no prior information about the unknown input. A general approach to solve for the state estimation of systems with unknown inputs that have arbitrary statistics is to apply unknown input decoupled state estimation, which yields unbiased minimum-variance filters (UMVF). Recently, the global optimality of the UMVF has been established [2].

Apart from the UMVF, optimal state estimation for descriptor systems also serves as a useful method to solve the UIF problem [3]-[4]. It is shown that standard systems with unknown inputs can be treated as descriptor systems through the definition of an extended state that contains the unknown input. However, these results only apply to unknown inputs that enter into the system dynamics. Maximum likelihood (ML) estimation and least-squares data fitting (see, e.g., [5]-[7]) also serve as useful means to estimate the optimal system state for standard systems with unknown inputs. Among these, a generalized Kalman filter, the “3-block” form of a descriptor Kalman filter (DKF), has been proposed to optimally estimate the system state for descriptor systems [5]. The connection between the DKF and the standard Kalman filter can be found in [8]. However, it is noticed that the DKF cannot be directly applied to solve the addressed UIF problem (see [9] for details). Recently, to remedy the problem, a 5-block extended DKF (EDKF) was proposed in [9] to optimally estimate the system state for standard systems with unknown inputs. It is shown that the 5-block EDKF is equivalent to the recently developed globally optimal state estimator in [1].

Descriptor systems serve as natural descriptions of systems that involve both dynamics and constraints among variables and are a natural starting point to describe noncausal phenomena [10]-[11]. More importantly, the descriptor formulation can treat the standard state-space system as a special case. More recently, we have extended the 5-block EDKF to optimally estimate the system state for descriptor systems with unknown inputs [12]. Specifically, through the proposed gain-covariance matrix (GCM), two compact versions of the 5-block EDKF, named as the least-squares data-fitting filter (LSDFF) and the descriptor recursive three-step filter (DRTSF), are further proposed. It is shown that the LSDFF and the DRTSF serve as extensions of the works done by Ishihara et al. [7] and Hsieh [1], respectively.

Recently, some attention has been focused on H∞ filtering for descriptor systems (see, e.g., [13]-[14] and the references therein). It is noticed that the results developed in [13] and [14] are derived based on data fitting arguments combined with two-players game theory and unbiased filtering technique, respectively. Nevertheless, all these results are not directly applicable for descriptor systems with unknown inputs that affect both the system state and the output.

In this paper, we continue the previous works done by Hsieh in [9] and [12] and further consider H∞ filtering for descriptor systems with unknown inputs that affect both the system and the output. It is shown that the recursive ML estimation method developed in [5] serves as a unified filtering framework to yield H∞ and Kalman filtering for descriptor systems and standard systems with or without unknown inputs. In the sequel, an H∞ version of the 5-block EDKF is developed, which, however, may suffer from computational complexity problem. To remedy this problem, three computationally efficient alternatives are further proposed based on a novel matrix transformation and the GCM.

II. PROBLEM FORMULATION

Consider a general class of descriptor systems with unknown inputs as follows:

\[ E_{k+1} x_{k+1} = A_k x_k + G_k d_k + w_k, \]
\[ y_k = C_k x_k + H_k d_k + v_k, \]
\[ z_k = L_k x_k + u_k, \]
where $x_k \in \mathbb{R}^n$ is the descriptor vector, $d_k \in \mathbb{R}^q$ is the unknown input, $y_k \in \mathbb{R}^p$ is the measured output, and $z_k \in \mathbb{R}^r$ is the signal to be estimated. Here, the row numbers of matrices $E_k$ and $A_k$ are equivalent ($= m$) and may not equal $n$, $w_k$, $v_k$, and $u_k$ are uncorrelated white noises sequences of zero-mean and with covariance matrices $Q_k > 0$, $R_k > 0$, and $S_k > 0$, respectively. The initial state $x_0$ is with unbiased mean $\hat{x}_0$ and covariance matrix $P_0$ and is independent of $w_k$, $v_k$, and $u_k$.

The problem of interest in this paper is to consider the optimal recursive estimation of $z_k$, i.e., finding $\hat{z}_k$, for the descriptor system (1)-(3) with the ML filtering method such that the following min-max optimization problem is achieved for all $k$:

$$\min \max_{\tilde{x}_k, d_k} J_k(y_k, \tilde{z}_k, \tilde{x}_k, d_k) > 0,$$

where

$$J_k = ||E_k \tilde{x}_k - A_{k-1} \tilde{x}_{k-1} - G_{k-1} \tilde{d}_{k-1}||_Q^{-1},$$

$$+ ||y_k - C_k \tilde{x}_k - H_k \tilde{d}_k||_R^{-1},$$

$$- \gamma^{-2} ||z_k - L_k \tilde{x}_k||_S^{-1},$$

(5)

where $\gamma > 0$ is the $H_\infty$ tuning parameter. Here, the notation $\tilde{x}_k$ represents the filtered estimate $\hat{x}_k$, and $||x||^2_W$ is used to represent $x^T W x$.

For descriptor systems without unknown inputs, i.e., $G_k = 0$ and $H_k = 0$, and assuming that matrix $[E_k^T C_k^T]$ has full-row rank, the $H_\infty$ descriptor data fitting filter (HDDFF) developed in [13] may serve as an optimal solution of the addressed min-max optimization problem. For easy reference, the HDDFF is listed as follows:

$$\hat{z}_k = L_k \tilde{x}_k,$$

(6)

$$\tilde{x}_k = (\tilde{P}_k^x)^{-1} + \gamma^{-2} L_k^T S_k^{-1} L_k)^{-1}$$

$$\times \left( E_k^T \tilde{P}_k^x + C_k^T R_k^{-1} y_k \right),$$

(7)

$$\tilde{P}_k^x = \left( E_k^T (E_k^T \tilde{P}_k^x E_k + C_k^T R_k^{-1} C_k - \gamma^{-2} L_k^T S_k^{-1} L_k) \right)^{-1},$$

(8)

where

$$\tilde{x}_k = A_{k-1} \tilde{x}_{k-1},$$

(9)

$$\tilde{P}_k^x = A_{k-1} P_k^x A_{k-1}^T + Q_{k-1},$$

(10)

Note that, if $\gamma$ goes to infinity the above HDDFF tends to the least-squares data-fitting filter (LSDFF) developed in [7].

On the other hand, for descriptor systems with unknown inputs and assuming that the system state is estimable and $\gamma$ goes to infinity, the 5-block EDKF$^u$ (or refined EDKF) in [9], where the superscript $u$ denotes the unrammed approach, can be easily generalized to yield the optimal ML estimate of $x_k$ due to a direct extension.

The main aim of this paper is to extend the 5-block EDKF$^u$ to present an $H_\infty$ version, which is named as the 6-block HEDKF$^u$, in order to solve the addressed min-max optimization problem.

### III. RE-Derivation of the HDDFF

To facilitate the derivation of the 6-block HEDKF$^u$, in this section we present an $H_\infty$ version of the (3-block) DKF, which is named as the 4-block HDDFF, that solves the optimization problem (4) for descriptor systems without unknown inputs, i.e., $d_k = 0$.

Viewing the dynamics of (1) and (3) as additional measurements, one can transform the original system of (1)-(3) into the following augmented output equation (AOE):

$$\begin{bmatrix} \tilde{x}_k \\ y_k \\ \tilde{z}_k \end{bmatrix} = \begin{bmatrix} E_k \\ C_k \\ L_k \end{bmatrix} x_k + \begin{bmatrix} \eta_k \\ v_k \\ \mu_k \end{bmatrix},$$

(11)

where $\tilde{x}_k$ is given by (9), $\tilde{z}_k$ remains to be determined, and

$$\eta_k = -A_{k-1}(x_{k-1} - \hat{x}_{k-1}) - w_{k-1}.$$  

(12)

From (12), it is clear that $\text{cov}(\eta_k) = \tilde{P}_k^x$. Note that to account for the negative cost in (5) we assume that $\mu_k$ in (11) is a fictitious zero-mean signal with the following negative covariance matrix:

$$\text{cov}(\mu_k) = -\gamma^2 S_k.$$  

(13)

Assuming that matrix $[E_k^T C_k^T]$ has full-row rank and using the recursive ML estimation procedure in Nikoukhah et al. [5] based on (11)-(13), one can obtain the following 4-block HDDF:

$$\hat{x}_k = \begin{bmatrix} 0 & 0 & 0 & I \end{bmatrix} \left( \Lambda_k^x \right)^{-1} \times \begin{bmatrix} \tilde{x}_k^T \\ y_k^T \\ \tilde{z}_k^T \\ 0 \end{bmatrix}^T,$$

(14)

$$\tilde{P}_k^x = -\begin{bmatrix} 0 & 0 & 0 & I \end{bmatrix} \left( \Lambda_k^x \right)^{-1} \times \begin{bmatrix} 0 & 0 & 0 & I \end{bmatrix}^T,$$

(15)

where

$$\Lambda_k^x = \begin{bmatrix} \tilde{P}_k^x & 0 & 0 & E_k \\ 0 & R_k & 0 & C_k \\ 0 & 0 & -\gamma^2 S_k & L_k \\ E_k^T C_k^T & L_k^T & 0 \end{bmatrix},$$

(16)

in which $\tilde{P}_k^x$ is given by (10).

In the following, we show that the 4-block HDDF can be equivalent to the HDDFF if $\tilde{z}_k$ is chosen as in (6). Denoting

$$\left( \Lambda_k^x \right)^{-1} = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \Omega_k & \Delta_k & \Upsilon_k & \Sigma_k \end{bmatrix},$$

(17)

where the entries marked as “$\times$” are irrelevant to the discussion, we have

$$\Omega_k = -\Sigma_k E_k^T (\tilde{P}_k^x)^{-1},$$

(18)

$$\Delta_k = -\Sigma_k C_k^T R_k^{-1},$$

(19)

$$\Upsilon_k = \gamma^{-2} \Sigma_k L_k^T S_k^{-1},$$

(20)

$$\Omega_k E_k + \Delta_k C_k + \Upsilon_k L_k = I.$$  

(21)

Using (18)-(20) in (21) and solving for $\Sigma_k$, we obtain

$$\Sigma_k = -(E_k^T (\tilde{P}_k^x)^{-1} E_k + C_k^T R_k^{-1} C_k - \gamma^{-2} L_k^T S_k^{-1} L_k)^{-1}.$$  

(22)
Using (17)-(20) in (14)-(15) yields
\[ \hat{x}_k = P_k^x \left( E_k^T (P_k^x)^{-1} \hat{x}_k + C_k^T R_k^{-1} y_k \right) - \gamma - 2 P_k^x \sum_{k=1}^{L_k} \hat{x}_k, \] (23)
\[ P_k^x = -\Sigma_k. \] (24)

Finally, using (6) in (23) and solving for \( \hat{x}_k \), we obtain
\[ \hat{x}_k = \left( (E_k^T P_k^x)^{-1} + \gamma^2 L_k^T S_k^{-1} L_k \right)^{-1} \times \left[ E_k^T (P_k^x)^{-1} \hat{x}_k + C_k^T R_k^{-1} y_k \right], \] (25)
which has the same expression as that given in (7). Thus, we conclude that the 4-block HDKF is equivalent to the HDDF.

Using (6) and (14), the system state estimate (14) can be rewritten as
\[ \hat{x}_k = \begin{bmatrix} 0 & 0 & 0 & I \end{bmatrix} (A_k^6)^{-1} \begin{bmatrix} \hat{x}_k^T & y_k^T & 0 & 0 \end{bmatrix}^T + \begin{bmatrix} 0 & 0 & 0 & I \end{bmatrix}^T L_k \hat{x}_k, \] (26)

Using (17), (20), and (24), and solving (26) for \( \hat{x}_k \), we have
\[ \hat{x}_k = \left( I + \gamma P_k^x L_k^T S_k^{-1} L_k \right)^{-1} \times \left( \begin{bmatrix} \hat{x}_k^T & y_k^T & 0 & 0 \end{bmatrix}^T \right). \] (27)

IV. DERIVATION OF THE 6-BLOCK HEDKF

In this section, we extend the 4-block HDKF, presented in Section III, to account for descriptor systems with unknown inputs, and the obtained filter is named as the 6-block HEDKF.

First, using the augmented state \( X_k = \left[ \begin{array}{c} x_k^T \ d_k^T \end{array} \right]^T \), we extend the AOE (11) as follows:
\[ \begin{bmatrix} \tilde{x}_k \\ y_k \\ \tilde{z}_k \end{bmatrix} = \begin{bmatrix} E_k & 0 & -\Pi_{k-1} \\ C_k & H_k & 0 \\ L_k & 0 & 0 \end{bmatrix} \begin{bmatrix} X_k \\ d_{k-1} \end{bmatrix} + \begin{bmatrix} \eta_k^T \\ \psi_k^T \\ \mu_k^T \end{bmatrix}, \] (28)
where
\[ \tilde{x}_k = \begin{bmatrix} A_{k-1} & G_{k-1} \end{bmatrix} \tilde{X}_{k-1}, \] (29)
\[ \Pi_{k-1} = G_{k-1} (I - H_{k-1}^+ H_{k-1}), \] (30)
\[ \eta_k = -A_{k-1} (x_{k-1} - \tilde{x}_{k-1}) - G_{k-1} (\Phi_{k-1} d_{k-1} - \tilde{d}_{k-1}) - w_{k-1}. \] (31)

Second, solving (28) for \( \tilde{X}_k \) using the recursive ML estimation method, we obtain the 6-block HEDKF as:
\[ \tilde{X}_k = \left( T_k^6 \right)^T (A_k^6)^+ \begin{bmatrix} \tilde{x}_k^T & y_k^T & \tilde{z}_k^T & 0 & 0 & 0 \end{bmatrix}^T, \] (32)
\[ P_k^X = -\left( T_k^6 \right)^T (A_k^6)^+ + T_k^6, \] (33)
where \( M^+ \) is the Moore-Penrose pseudo-inverse of \( M \),
\[ T_k^6 = \begin{bmatrix} 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 \end{bmatrix}^T, \] (34)
\[ A_k^6 = \begin{bmatrix} P_k^x & 0 & 0 & E_k & 0 & -\Pi_{k-1} \\ 0 & R_k & 0 & C_k & H_k & 0 \\ 0 & 0 & -\gamma^2 S_k & L_k & 0 & 0 \\ E_k^T & C_k^T & L_k^T & 0 & 0 & 0 \\ 0 & H_k^T & 0 & 0 & 0 & 0 \\ -\Pi_{k-1} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \] (35)
in which
\[ \tilde{P}_k^x = \begin{bmatrix} A_{k-1} & G_{k-1} \end{bmatrix} P_{k-1} \begin{bmatrix} A_{k-1}^T \\ C_{k-1}^T \end{bmatrix} + Q_{k-1}. \] (36)

Third, using (6) in (32) and solving for \( \tilde{X}_k \) yields the following augmented system state estimate:
\[ \tilde{X}_k = \left( I - (T_k^6)^T (A_k^6)^+ \begin{bmatrix} 0 & 0 & I & 0 & 0 & 0 \end{bmatrix}^T \tilde{L}_k \right)^{-1} \times (T_k^6)^T (A_k^6)^+ \begin{bmatrix} \tilde{x}_k^T & y_k^T & 0 & 0 & 0 & 0 \end{bmatrix}^T, \] (37)
where \( \tilde{L}_k = \begin{bmatrix} 0 \ L_k \end{bmatrix} \). The estimated signal \( \tilde{z}_k \) is obtained as follows:
\[ \tilde{z}_k = \tilde{L}_k \tilde{X}_k. \] (38)

Finally, we address the filter initialization issue. As noted in [13], the 6-block HEDKF can be initialized with the following substitutions:
\[ \tilde{x}_0 \rightarrow \tilde{x}_0, \quad \tilde{P}_0^x \rightarrow \tilde{P}_0, \quad E_0 \rightarrow I, \quad \Pi_{-1} \rightarrow 0. \] (39)

Remark 1: The proposed 6-block HEDKF serves as an asynchronous solution to solve \( H_{\infty} \) and Kalman filtering for descriptor systems and standard systems with or without unknown inputs. It can be easily checked that the HEDKF generalizes the results in [12] and [13].

V. PRACTICAL ISSUES OF IMPLEMENTING THE 6-BLOCK HEDKF

It is noticed that the 6-block HEDKF may not be practical for implementation due to its tremendous computational load in implementing a \( (2(n+q)+p+r) \times (2(n+q)+p+r) \) pseudo-inverse (see Section VI for an illustration). In this section, we present three computationally efficient alternatives to the 6-block HEDKF. Without loss of generality, we assume that matrix \( \Pi_k \) is not null for all \( k \geq 0 \).

A. The 6-Block HEDKF

The basic idea for deriving a computationally efficient algorithm of the 6-block HEDKF is to implement the pseudo-inverse operation \( (A_k^6)^+ \) by using an alternative matrix inverse \( (A_k^6)^{-1} \).

Denote the following full-rank factorizations:
\[ H_k = \tilde{H}_k \hat{H}_k, \quad \Pi_{k-1} = \tilde{\Pi}_{k-1} \hat{\Pi}_{k-1}, \] (40)
where \( \tilde{H}_k \) and \( \tilde{\Pi}_{k-1} \) are full-column rank matrices and \( \hat{H}_k \) and \( \hat{\Pi}_{k-1} \) are full-rank matrices. Then, using (40), one has the following relationship:
\[ \begin{bmatrix} 0 & -\Pi_{k-1} \end{bmatrix} = \begin{bmatrix} 0 & -\tilde{\Pi}_{k-1} \end{bmatrix} \times diag\{\tilde{H}_k, \tilde{\Pi}_{k-1}\}, \] (41)
where
\[ \Psi_k = diag\{I, I, I, I, \tilde{H}_k, \tilde{\Pi}_{k-1}\}, \] (42)
\[ \Lambda^6_k = \begin{bmatrix}
\tilde{P}_k^* \\
0 & 0 & E_k & 0 & -\tilde{\Pi}_{k-1} \\
0 & R_k & 0 & C_k & \tilde{H}_k \\
0 & 0 & -\gamma^2 S_k & L_k & 0 \\
E_k^T & C_k^T & L_k^T & 0 & 0 \\
0 & \tilde{H}_k^T & 0 & 0 & 0 \\
-\tilde{\Pi}_{k-1}^T & 0 & 0 & 0 & 0
\end{bmatrix}. \] (43)
In the following discussions, we assume that the following matrix:
\[ \Lambda^6_k \]
has full-column rank.

Thus, using (41) and the relationship \((AB)^+ = B^+A^+\),
where \(A\) is of full-column rank and \(B\) is of full-row rank, one has
\[ (\Lambda^6_k)^+ = \Psi_k^T (\Lambda^6_k)^{-1} (\Psi_k^T)^T. \] (45)
Moreover, we have the following relationship:
\[ (T_k^6)^T \Psi_k^+ = \begin{bmatrix} 0 & 0 & 0 & I & 0 & 0 \\
0 & 0 & 0 & 0 & H_k^+ & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \equiv (T_k^6)^T. \] (46)

Finally, using (45)-(46) in (33) and (37) yields the following 6-block HEDKF,
where the superscript \(t\) denotes the transformed approach:
\[ \hat{\Delta}_k = \left( I - (T_k^6)^T (\Lambda^6_k)^{-1} \left[ \begin{array}{l}
0 \\
0 \\
0 \\
0 \\
\end{array} \right]
\right)^{-1}
\times (T_k^6)^T (\Lambda^6_k)^{-1} \left[ \begin{array}{l}
\bar{x}_k^T \\
\bar{y}_k^T \\
0 \\
0 \\
0 \\
\end{array} \right]. \] (47)
\[ P_k^{X} = -(T_k^6)^T (\Lambda^6_k)^{-1} T_k^6. \] (48)

B. The HDDFF

The 6-block HEDKF can be further simplified by using the approach given in Section III.

Denote
\[ (\Lambda^6_k)^{-1} = \begin{bmatrix}
\tilde{P}_k^* & 0 & 0 & \varepsilon_k \\
0 & R_k & 0 & C_k \\
0 & 0 & -\gamma^2 S_k & L_k \\
\varepsilon_k^T & C_k^T & L_k^T & 0
\end{bmatrix}^{-1} = \begin{bmatrix}
\times & \times & \times & \times \\
\times & \times & \times & \times \\
\times & \times & \times & \times \\
\Omega_k & \Delta_k & \Upsilon_k & \Sigma_k
\end{bmatrix}, \] (49)
where
\[ \varepsilon_k = \left[ \begin{array}{c}
E_k \\
0 \\
0 \\
0
\end{array} \right], \] (50)
\[ C_k = \left[ \begin{array}{c}
C_k \\
\tilde{H}_k \\
0 \\
0
\end{array} \right], \] (51)
\[ L_k = \left[ \begin{array}{c}
L_k \\
0 \\
0
\end{array} \right]. \] (52)
Using (49), we have
\[ \Omega_k = -\Sigma_k \varepsilon_k^T (\tilde{P}_k^*)^{-1}, \] (53)
\[ \Delta_k = -\Sigma_k \varepsilon_k^T R_k^{-1}, \] (54)
\[ \Upsilon_k = \gamma^{-2} \Sigma_k L_k \varepsilon_k T_k^{-1}. \] (55)

where
\[ \Sigma_k = -\left( \varepsilon_k^T (\tilde{P}_k^*)^{-1} \varepsilon_k + C_k^T R_k^{-1} C_k - \gamma^{-2} L_k T_k^{-1} S_k^{-1} L_k \right)^{-1}. \] (56)
Substituting (49) and (52)-(55) into (47)-(48) and using the following relationship:
\[ L_k^T = \begin{bmatrix} 0 & 0 & 0 \\
0 & \tilde{H}_k^+ & 0 \\
0 & 0 & 0
\end{bmatrix} \]
we obtain the following HDDFF corresponding to (1)-(3), which is a generalization of (6)-(8):
\[ \hat{\Delta}_k = \left( I + \gamma^{-2} P_k^{X} T_k^6 \varepsilon_k^{-1} L_k^{-1} \right)^{-1} \begin{bmatrix} 0 & 0 \\
0 & \tilde{H}_k^+
\end{bmatrix} \times \Sigma_k \left[ \begin{array}{c}
I \\
0
\end{array} \right] \] (57)
\[ P_k^{X} = \left[ \begin{array}{c}
I \\
0
\end{array} \right] \begin{bmatrix} 0 & 0 \\
0 & \tilde{H}_k^+
\end{bmatrix} \] (58)

Remark 2: If \(\gamma\) goes to infinity, the above HDDFF tends to the LSDFF developed in [12]. Moreover, in the special case \(G_k = 0\) and \(H_k = 0\), the HDDFF will be equivalent to the original work in [13].

C. The HDRFSF

The HDDFF can be slightly simplified via further reducing the dimension of matrix inverse. This is achieved by determining the gain-covariance matrix (GCM) [9] corresponding to the 6-block HEDKF as follows:
\[ (\Lambda^6_k)^{-1} = \begin{bmatrix}
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times \\
\Gamma_k & K_k & K_k^z - P_k^X & -P_k^{x^{d}} & \times \\
\Xi_k & M_k & M_k^z - (P_k^{x^{d}})^T & -P_k^{x^{d}} & \times \\
\times & \times & \times & \times & \times
\end{bmatrix}, \] (59)
where \(\bar{\Delta}_k\) is part of the original unknown input vector that is estimable.

From (43) and (59), we have the following relationships:
\[ \Gamma_k E_k = I - K_k C_k - K_k^z L_k, \] (60)
\[ \Xi_k P_k^X = (P_k^{x^{d}})^T E_k^T + \{\bullet\}_{(5,6)} \Pi_{k-1}, \] (61)
\[ K_k = P_k^X C_k^T R_k^1, \] (62)
\[ M_k = H_k^+ (I - R_k \Theta_k - C_k K_k), \] (63)
\[ K_k^z = -\gamma^{-2} P_k^X L_k^T S_k^{-1}, \] (64)
\[ M_k^z = -\gamma^{-2} (P_k^{x^{d}})^T L_k^T S_k^{-1}, \] (65)
\[ P_k^X E_k^T = \Gamma_k P_k^X - \{\bullet\}_{(4,6)} \Pi_{k-1}, \] (66)
\[ P_k^{x^{d}} = (K_k R_k - P_k^X C_k^T) (H_k^+)^T, \] (67)
\[ P_k^{x^{d}} = H_k^+ (R_k M_k^T - C_k P_k^{x^{d}}), \] (68)
\[ \Theta_k H_k = 0, \] (69)
where \(\{\bullet\}_{(i,j)}\) is the \(i,j\) entry of the GCM and
\[ R_k^1 = R_k^{-1} (I - \tilde{H}_k H_k^+), \] (70)
\[ H_k^* = (H_k^+ R_k^1 \tilde{H}_k)^{-1} H_k R_k^{-1}. \] (71)
Now, we are in place to further simplify the expressions in (63), (67), and (68). This is achieved by using the following relationships:
\[ \begin{align*}
\tilde{H}_k^* R_k \Theta_k &= 0, \\
\tilde{H}_k^* R_k K_k^T &= 0,
\end{align*} \]
by which we obtain
\[ \begin{align*}
M_k &= \tilde{H}_k^* (I - C_k K_k), \quad (71) \\
P_k^{zd} &= -P_k^z C_k^T (\tilde{H}_k^*)^T, \quad (72) \\
P_k^d &= \tilde{H}_k^* (C_k P_k^z C_k^T + R_k)(\tilde{H}_k^*)^T. \quad (73)
\end{align*} \]

Then, we determine the unspcific matrices: \( \Gamma_k, \Xi_k, \) and \( P_k^z, \) given by (60), (61), and (66), respectively. Using the following notations:
\[ \begin{align*}
(P_k^x)^\dagger &= (P_k^x)^{-1}(I - \Pi_{k-1} \bar{\Pi}_{k-1}), \\
\Pi_{k-1}^z &= (\Pi_{k-1}^z - P_k^z)^{-1} \Pi_{k-1}^z - (P_k^z)^{-1}, \quad (74)
\end{align*} \]
and the relationship \( \Gamma_k \Pi_{k-1} = 0 \) in (66) yields
\[ \Gamma_k = P_k^x E_k^T (P_k^x)^\dagger. \quad (76) \]
Similarly, we can determine the matrix \( \Xi_k \) in (61) as
\[ \Xi_k = (P_k^{zd})^T E_k^T (P_k^z)^\dagger = -\tilde{H}_k^* C_k \Gamma_k, \quad (77) \]
where (72) and (76) are used. Using (62), (64), and (76) in (60), and solving for \( P_k^z \) we obtain
\[ P_k^z = \left( E_k^T (P_k^x)^\dagger E_k + C_k^T R_k C_k - \gamma^2 L_k^T S_k^{-1} L_k \right)^{-1}. \quad (78) \]

Next, comparing (49) with (59) and using (71)-(73) and (77), (57) and (58) can be expressed as:
\[ \begin{align*}
\hat{X}_k &= \left[ \begin{array}{c}
I + \gamma^{-2} P_k^z L_k^T S_k^{-1} L_k \\
-\gamma^{-2} \tilde{H}_k^* C_k \Gamma_k P_k^{zd} L_k^T S_k^{-1} L_k \end{array} \right]^{-1} \\
&\times \left[ \begin{array}{c}
\tilde{H}_k^* H_k^* \ \\
\Gamma_k \bar{X}_{k-1} + K_k y_k
\end{array} \right], \\
P_k^x &= \text{diag}(I, \tilde{H}_k^*) \\
&\times \left[ \begin{array}{c}
-H_k^* C_k P_k^z C_k^T (\tilde{H}_k^*)^T \\
-H_k^* C_k P_k^z C_k^T + R_k (H_k^*)^T
\end{array} \right] \\
&\times \text{diag}(I, \tilde{H}_k^*). \quad (79)
\end{align*} \]

Finally, summarizing the above results, we obtain the following \( H_{\infty} \) descriptor recursive four-step filter (HDFSF):

Step 1: Establish the system state \( x_k \)
\[ \hat{x}_k = (P_k^{zd})^{-1} + \gamma^{-2} L_k^T S_k^{-1} L_k \\
\times \left( E_k^T (P_k^x)^\dagger \hat{x}_k + C_k^T R_k^\dagger y_k \right), \quad (81) \]
\[ P_k^x = \left( E_k^T (P_k^x)^\dagger E_k + C_k^T R_k^\dagger C_k - \gamma^2 L_k^T S_k^{-1} L_k \right)^{-1}. \quad (82) \]

Step 2: Estimate the predicted system state \( \bar{x}_{k+1} \)
\[ \bar{x}_{k+1} = A_k \hat{x}_k + G_k d_k, \quad (87) \]
\[ P_{k+1}^x = \left[ \begin{array}{c}
A_k \ \\
G_k
\end{array} \right] \left[ \begin{array}{c}
P_k^z \\
P_k^{zd}
\end{array} \right] \left[ \begin{array}{c}
A_k^T \ \\
G_k^T
\end{array} \right] + Q_k. \quad (88) \]

Step 4: Estimate the predicted system state \( \tilde{x}_{k+1} \)
\[ \tilde{x}_{k+1} = A_k \hat{x}_k + G_k d_k, \quad (87) \]
\[ P_{k+1}^x = \left[ \begin{array}{c}
A_k \ \\
G_k
\end{array} \right] \left[ \begin{array}{c}
P_k^z \\
P_k^{zd}
\end{array} \right] \left[ \begin{array}{c}
A_k^T \ \\
G_k^T
\end{array} \right] + Q_k. \quad (88) \]

Remark 3: If \( \gamma \) goes to infinity, the HDFSF tends to the DRTSF developed in [12]. Moreover, in the special case \( E_k = I \), the HDFSF will be equivalent to the ERTSF [1].

VI. AN ILLUSTRATIVE EXAMPLE

As an illustrative example, we consider a numerical example adapted form [7], which is given as follows:
\[ \begin{align*}
E_{k+1} &= \left[ \begin{array}{c}
1 \\
0
\end{array} \right], \\
A_k &= \left[ \begin{array}{c}
0.3 \\
0.2
\end{array} \right], \\
\gamma &= \left[ \begin{array}{c}
0.01 \\
-1.25
\end{array} \right], \\
H_k &= \left[ \begin{array}{c}
0 \\
0
\end{array} \right], \\
L_k &= \left[ \begin{array}{c}
1.4 \\
0.8
\end{array} \right], \\
C_k &= I_2, \\
S_k &= 1, \\
Q_k &= \left[ \begin{array}{c}
9.3 \\
290
\end{array} \right], \\
R_k &= \left[ \begin{array}{c}
0.1 \\
0.001
\end{array} \right].
\end{align*} \]

In the simulation, we set \( \bar{x}_0 = 0, \bar{P}_0 = I_2, \) and
\[ \begin{align*}
d_k &= \left[ \begin{array}{c}
5u_k[k] - 5u_k[k - 20] + 5u_k[k - 79] \\
4u_k[k] - 4u_k[k - 30] + 4u_k[k - 65]
\end{array} \right],
\end{align*} \]
where \( u_k[k] \) is the unit-step function. The simulation time is 100 time steps with a Monte Carlo simulation of 100 runs.

In the simulation, the 6-block HEDKFs \([37] \) and \( [33] \), the 6-block HEDKF \([47]-[48] \), the HDDFF \([57]-[58] \), the HDFSF \([81]-[88] \), and the DRTSF in [12] are considered. Note that the DRTSF can be seen as a special case of the HDFSF with \( \gamma \to \infty \). We illustrate the root-mean-square-errors (rmse) in the estimated signal \( z_k \) of the HDFSF and the DRTSF in Fig. 1, from which we observe that the filtering performance of the HDFSF will reduce to that of the DRTSF when \( \gamma \) goes to infinity as expected. Moreover, the lower the value of \( \gamma \) is the better the performance of the HDFSF will achieve. As shown in [13], the lower bound of \( \gamma \) can be estimated from (82) as follows:
\[ \gamma^2 \left( E_k^T Q_k^{-1} E_k + C_k^T R_k^\dagger C_k \right) \geq L_k^T S_k^{-1} L_k, \quad (89) \]
where
\[ Q_k^{-1} = Q_k^{-1} - Q_k^{-1} \Pi_k (S_k^\dagger S_k^{-1} \Pi_k)^{-1} S_k^\dagger Q_k^{-1}. \]
By (89), we obtain that the lower bound of \( \gamma \) in this simulation is 0.3720.

In illustrating the filtering performance of the proposed results, we choose the \( H_{\infty} \) tuning parameter as \( \gamma = 0.545 \) based on Fig. 1. Because the matrix in (44) has full-column rank, we can deduce that all the new developed filters, i.e., the 6-block HEDKFs, the 6-block HEDKF, the HDDFF, and the HDFSF, will yield the same optimal filtering
TABLE I

PERFORMANCE OF THE HEDKF , HDDFF , HDRFSF , AND DRTSF .

<table>
<thead>
<tr>
<th>Filter</th>
<th>$\tilde{z}_k$</th>
<th>$\tilde{z}_1$</th>
<th>$\tilde{z}_2$</th>
<th>flops</th>
</tr>
</thead>
<tbody>
<tr>
<td>6-block HEDKF</td>
<td>3.2011</td>
<td>0.4542</td>
<td>4.4218</td>
<td>28175</td>
</tr>
<tr>
<td>6-block HDDFF</td>
<td>3.2011</td>
<td>0.4542</td>
<td>4.4218</td>
<td>1673</td>
</tr>
<tr>
<td>HDRFSF</td>
<td>3.2011</td>
<td>0.4542</td>
<td>4.4218</td>
<td>938</td>
</tr>
<tr>
<td>DRTSF</td>
<td>3.2533</td>
<td>0.4556</td>
<td>4.4926</td>
<td>837</td>
</tr>
</tbody>
</table>

performance. Moreover, due to that the feedthrough matrix of the unknown input to the output is not of full-column rank, the unknown input vector is not completely estimable. It can be checked that only the second component of the unknown input vector is estimable. Thus, the dedicated filter should decouple the first component of the unknown input vector from the filtering error in order to achieve the optimal performance. We list the rmse errors in the state estimates of the considered filters in Table 1, from which we obtain that all the new proposed $H_\infty$ filters yield the optimal estimated signal and system state estimates, which are slightly better than those obtained of the previously proposed DRTSF.

Then, we address complexity: we use floating point operations, or “flops,” in Matlab as a measure of the computational complexity of the aforementioned filters. The results are listed in Table 1, from which we have the following findings: 1) the 6-block HEDKF is most computationally intensive than the others due to the heavy load of the pseudo-inverse operation, which, however, can be effectively solved by using the alternative 6-block HEDKF; 2) the HDDFF and the HDRFSF both serve as compact versions of the 6-block HEDKF, whereas the latter has the less computational complexity than the former due to the GCM concept; and 3) the DRTSF has the least computational complexity, however, at the expense of sacrificing filtering performance.

VII. CONCLUSION

In this paper, we derived $H_\infty$ Kalman filtering recursions for rectangular discrete-time descriptor systems with unknown inputs that affect both the system and the output. An alternative to the data fitting approach for deriving an $H_\infty$ filter is developed based on the descriptor Kalman filtering method. A 6-block HEDKF is proposed serving as a unified solution to solve the addressed $H_\infty$ unknown input filtering problem for descriptor systems. Due to computational complexity consideration, three simplified versions of the 6-block HEDKF are presented. Using a novel matrix transformation, a more efficient 6-block EDKF is proposed as an alternative to the HEDKF, where the pseudo-inverse operation is implemented by using an alternative matrix inverse. Moreover, based on the recently developed GCM concept, two compact versions of the 6-block HEDKF, i.e., the HDDFF and the HDRFSF, are further proposed to reduce matrix inversion dimension; specifically, the latter is less computationally intensive than the former. Finally, simulation results verify the usefulness of the proposed results.

VIII. ACKNOWLEDGMENT

The author gratefully acknowledges the contribution of the National Science Council, Taiwan and reviewers’ comments.

REFERENCES