H∞ Fuzzy Estimation for a Class of Nonlinear Discrete-Time Dynamic Systems

Chung-Shi Tseng, Member, IEEE, and Bor-Sen Chen, Senior Member, IEEE

Abstract—This paper studies H∞ fuzzy filtering design for state estimation of nonlinear discrete-time systems with bounded but unknown disturbances. First, the Takagi and Sugeno fuzzy model is proposed to represent a nonlinear system. Next, using a linear matrix inequality (LMI) approach, the H∞ fuzzy filtering problems are characterized in terms of a linear matrix inequality problem (LMI). The LMI can be efficiently solved using convex optimization techniques. Simulation examples are given to illustrate the design procedure and the applicability of the proposed method. The results indicate that the proposed method is suitable for practical applications.

Index Terms—H∞ fuzzy filtering, LMI.

I. INTRODUCTION

In general, it is difficult to design an efficient filter for state estimation of nonlinear systems. A nonlinear estimation algorithm, known as the extended Kalman filter (EKF), is often used for state estimation. With this algorithm, linearization of the nonlinear systems around the present estimate makes it possible to apply the linear Kalman filter theory [1]–[5]. The design of an EKF relies on an exact knowledge of plant dynamics and the second-order statistical properties of external disturbances and measurement noises. In some practical systems, however, the statistical properties of external disturbances and measurement noises are rarely known. Recently, H∞ filtering has been used to address this issue of uncertain external disturbances and measurement noises [6], [7]. In [7], model uncertainties are additionally dealt with via a backward Riccati equation. In this paper, the H∞ filtering design problem for nonlinear systems is studied.

The advantage of using an H∞ filter in comparison with a Kalman filter is that no statistical assumption on the exogenous signals is needed. The H∞ filter is designed by minimizing state estimation error for the worst-case bounded disturbances and noises. As a result, H∞ filters are more robust than the Kalman filter. Furthermore, with an H∞ filtering design, the induced L2 norm of the operator from the disturbances to the estimation error can be less than a prescribed attenuation level [7], [10]. Nonlinear H∞ filter design (or control design) needs to solve a nonlinear Hamilton–Jacobi equation, which cannot be easily solved, except for some special cases [10]–[15]. In [13], an H∞ estimation algorithm is proposed for nonlinear systems based on a linearized model around the suboptimal state estimate with an adaptive performance bound. The adaptive process increases the computational burden. In [15], necessary and sufficient conditions are given for the solvability of a standard H∞ suboptimal control and estimation problem that involve the solvability of a pair of partial differential equations of the Hamilton–Jacobi type. In this paper, the nonlinear H∞ filtering problem is studied using a fuzzy approach.

Recently, there have been many applications of fuzzy systems theory in various fields, such as control systems, communication systems, and signal processing. In most of these applications, fuzzy systems are considered to be universal approximators for certain nonlinear systems. The Takagi and Sugeno fuzzy model [16], which has been proved to be a good representation for a certain class of nonlinear dynamic systems, was extensively studied in control systems [16]–[21]. In this study, a Takagi and Sugeno fuzzy model is proposed to interpolate a class of nonlinear systems. Based on the fuzzy model, the H∞ fuzzy filtering design problem is characterized by making the prescribed attenuation level as small as possible, subject to some linear matrix inequality (LMI) constraints. This convex feasibility problem is also called the linear matrix inequalities problem (LMIP). LMIP can be efficiently solved by the convex optimization algorithm [22].

The paper is organized as follows. An H∞ discrete-time fuzzy filtering design for the state estimation of nonlinear systems is introduced in Section II. In Section III, simulation examples are provided to demonstrate the design procedure. Finally, concluding remarks are made in Section IV.

II. H∞ FUZZY FILTERING FOR NONLINEAR DISCRETE-TIME SYSTEMS

Consider a class of nonlinear discrete-time system

\[
\begin{align*}
    x(k+1) &= F(x(k)) + Bu(k) \\
    y(k) &= H(x(k)) + Gv(k) \\
    s(k) &= Dx(k)
\end{align*}
\]

where \( x(k) = [x_1(k), x_2(k), \ldots, x_n(k)]^T \in \mathbb{R}^{n\times1} \) denotes the state vector; \( y(k) \in \mathbb{R}^{m\times1} \) and \( s(k) \in \mathbb{R}^{l\times1} \) denote output and a linear combination of the state variables to be estimated, respectively; \( u(k) \in \mathbb{R}^{n\times1} \) and \( v(k) \in \mathbb{R}^{m\times1} \) are assumed to be bounded external disturbance and measurement noise, respectively; \( B \in \mathbb{R}^{n\times n}, G \in \mathbb{R}^{m\times m} \) and \( D \in \mathbb{R}^{l\times n} \).
are constant matrices; and \( F(x(k)) \in \mathbb{R}^{n \times 1} \) and \( H((x(k)) \in \mathbb{R}^{m \times 1} \) are vector fields with \( F(0) = 0 \) and \( H(0) = 0 \).

The purpose of this study is to find an estimator: a mapping whose input is the initial estimate \( \hat{x}_0 \) as well as the observation signals \( y(0), \ldots, y(k_f - 1) \) and whose output is sequence of estimates \( \hat{x}(k), k = 1, \ldots, k_f \), such that the \( H_\infty \) performance

\[
\begin{align*}
&\sum_{k=1}^{k_f} e^T(k)Qe(k) \\
\leq & \rho^2 \sum_{k=0}^{k_f-1} (u^T(k)u(k) + v^T(k)v(k))
\end{align*}
\] (2)

is achieved for all \( u(k) \) and \( v(k) \) [7]–[10]. Here, \( e(k) = s(k) - \hat{s}(k) = D(x(k) - \hat{x}(k)) \). \( P_0 \) is an initial weighting matrix that is assumed to be symmetric and positive definite, and \( Q \) is a symmetric semi-positive definite matrix. Note that (2) can be written in the following form:

\[
\begin{align*}
&\sum_{k=1}^{k_f} e^T(k)Qe(k) \\
= & \sum_{i=1}^{k_f} (D(x(k) - \hat{x}(k)))^TQ(D(x(k) - \hat{x}(k))) \\
\leq & \rho^2 \left[ e^T(0)P_0e(0) + \sum_{k=0}^{k_f-1} (u^T(k)u(k) + v^T(k)v(k)) \right].
\end{align*}
\] (3)

Remark 1: The meaning of (2) is that \( e(k) \) must be less than a prescribed level \( \rho \) for all possible \( e(0), u(k), \) and \( v(k) \) from the energy point of view, i.e., the induced \( \ell_2 \) norm of the operator from \( e(0), u(k), \) and \( v(k) \) to \( e(k), k \in [1, k_f] \) must be less than \( \rho \). For stable LTI systems with an infinite-time horizon, the induced \( \ell_2 \) norm is identical to the \( H_\infty \) norm in the frequency domain [7]. \( H_\infty \) filter design, by protecting against the worst case, is more suitable for systems with unknown (or uncertain) driving noise and measurement noise. \( H_\infty \) optimality is achieved with the lowest possible value of \( \rho^2 \) in (2).

The \( i \)th rule of the Takagi–Sugeno fuzzy model for the discrete-time nonlinear system in (1) is proposed as the following form [16]–[18]:

**Rule i:**

If \( z_i(k) \) is \( F_{ij} \) and \( z_j(k) \) is \( F_{ij} \)

Then \( x(k+1) = A_ix(k) + Bu(k) \)

\[
\begin{align*}
y(k) & = C_i x(k) + Gv(k) \\
\dot{s}(k) & = D_ix(k)
\end{align*}
\] (4)

for \( i = 1, 2, \ldots, L \), where \( F_{ij} \) is the fuzzy set; \( A_i \in \mathbb{R}^{n \times n}; C_i \in \mathbb{R}^{m \times n}; L \) is the number of If–Then rules; \( z_1(k), z_2(k), \ldots, z_L(k) \) are the premise variables; and \( k = 0, \ldots, k_f - 1 \). The state dynamic and the output of the fuzzy system are inferred, respectively, as follows:

\[
x(k+1) = \sum_{i=1}^{L} \mu_i(z(k))(A_ix(k) + Bu(k))
\]

\[
y(k) = \sum_{i=1}^{L} \mu_i(z(k))(C_i x(k) + Gv(k))
\]

where

\[
\mu_i(z(k)) = \prod_{j=1}^{g} F_{ij}(z_j(k))
\]

and \( F_{ij}(z_j(k)) \) is the grade of membership of \( z_j(k) \) in \( F_{ij} \). Finally

\[
h_i(z(k)) = \frac{\mu_i(z(k))}{\sum_{i=1}^{L} \mu_i(z(k))}
\]

\[
z(k) = [z_1(k), z_2(k), \ldots, z_L(k)].
\] (6)

In this paper, we assume [16]

\[
\mu_i(z(k)) \geq 0, \quad \text{for } i = 1, 2, \ldots, L
\]

and

\[
\sum_{i=1}^{L} \mu_i(z(k)) > 0 \quad \text{for } k = 0, \ldots, k_f - 1.
\]

Therefore, we have

\[
h_i(z(k)) \geq 0 \quad \text{for } i = 1, 2, \ldots, L
\] (7)

and

\[
\sum_{i=1}^{L} h_i(z(k)) = 1.
\] (8)

The fuzzy model in (5) can be interpreted as an interpolation of \( L \) linear systems through the membership function \( h_i(z(k)) \) to approximate the nonlinear system in (1). Therefore, the nonlinear system in (1) can be described as

\[
x(k+1) = F(x(k)) + Bu(k)
\]

\[
y(k) = \sum_{i=1}^{L} h_i(z(k))(A_ix(k) + F(x(k))
\]

\[
= \sum_{i=1}^{L} h_i(z(k))A_ix(k) + Bu(k)
\] (9)
where $k = 0, \ldots, k_f - 1$, and
\begin{align*}
\Delta F(x(k)) &= F(x(k)) - \sum_{i=1}^{L} h_i(z(k)) A_i x(k) \quad (11) \\
\Delta H(x(k)) &= H(x(k)) - \sum_{i=1}^{L} h_i(z(k)) C_i x(k) \quad (12)
\end{align*}
denote the approximation (or interpolation) errors between the nonlinear system in (1) and the fuzzy model in (5).

Assumption: There exist $\Omega$ and $\Phi$ such that
\begin{align*}
\Delta F(x(k))^T \Delta F(x(k)) &
\leq x(k)^T \Omega x(k) \quad (13) \\
\Delta H(x(k))^T \Delta H(x(k)) &
\leq x(k)^T \Phi x(k) \quad (14)
\end{align*}
for all trajectories $x(k)$.

Remark 2: The assumptions in $F(0) = 0$ and $H(0) = 0$ are related to the assumptions in (13) and (14). Under these assumptions, the assumptions in (13) and (14) are still valid for $z(k) = 0$. In other words, the system under consideration is not universally applicable and restricted on the system with $F(0) = 0$ and $H(0) = 0$.

Remark 3: The T–S fuzzy model is a piecewise interpolation of several linear models at different operating points through fuzzy membership functions. It is only an approximation model for a nonlinear system. The approximation error between the fuzzy model and the original nonlinear system does exist. The idea behind the assumptions in (13) and (14) is that if the approximation error can be covered by an upper bound, then we can design a robust $H_\infty$ fuzzy filter for the nonlinear system to tolerate the approximation error based on the upper bound. The upper bound can be thought of as the worst-case approximation error.

Remark 4: In general, the premise variables $z(k)$ depend on the state variables, and the premise variables $\hat{z}(k)$ depend on the estimated state variables. If $z_1(k) = x_1(k)$, $z_2(k) = x_2(k)$, $\ldots$, $z_p(k) = x_p(k)$, then $\hat{z}_1(k) = \hat{x}_1(k)$, $\hat{z}_2(k) = \hat{x}_2(k)$, $\ldots$, $\hat{z}_p(k) = \hat{x}_p(k)$.

The overall fuzzy estimator is written as
\begin{align*}
\hat{x}(k+1) &= \sum_{j=1}^{L} h_j(\hat{z}(k)) (A_j \hat{x}(k) + B_j \hat{u}(k)) \\
\hat{z}(k+1) &= \sum_{j=1}^{L} h_j(\hat{z}(k)) \left( \sum_{i=1}^{L} h_i(\hat{z}(k)) A_i x(k) + \Delta F(x(k)) \right) \nonumber \\
&\quad \times \left( \sum_{i=1}^{L} h_i(\hat{z}(k)) C_i x(k) + \Delta H(x(k)) \right) + B_j \hat{u}(k) \nonumber \\
&\quad + \sum_{i=1}^{L} h_i(\hat{z}(k)) \sum_{j=1}^{L} h_j(\hat{z}(k)) \sum_{s=1}^{L} h_s(\hat{z}(k)) (K_j C_i x(k) + (A_j - K_j C_i) \hat{x}(k)) + K_j \Delta H + K_j G \hat{u}(k). \quad (17)
\end{align*}

Then, from (9) and (17), the augmented system can be written as the following form:
\begin{equation}
\begin{bmatrix}
\dot{x}(k+1) \\
\dot{z}(k+1)
\end{bmatrix}
= \begin{bmatrix}
\sum_{i=1}^{L} h_i(\hat{z}(k)) A_i x(k) + \Delta F(x(k)) + B_j \hat{u}(k) \\
\sum_{i=1}^{L} h_i(\hat{z}(k)) \sum_{j=1}^{L} h_j(\hat{z}(k)) \sum_{s=1}^{L} h_s(\hat{z}(k)) (K_j C_i x(k) + (A_j - K_j C_i) \hat{x}(k)) + K_j \Delta H + K_j G \hat{u}(k)
\end{bmatrix}.
\end{equation}

Let us define
\begin{align*}
\eta(k) &= \begin{bmatrix}
x(k) \\
\hat{z}(k)
\end{bmatrix}, \quad \hat{\eta}_{in} = \begin{bmatrix}
A_i & 0 \\
K_j C_i & A_j - K_j C_i
\end{bmatrix} \\
\hat{E}_{ij} &= \begin{bmatrix}
I & 0 \\
0 & K_j
\end{bmatrix}, \quad \Delta \Gamma(x(k)) = \begin{bmatrix}
\Delta F(x(k)) \\
\Delta H(x(k))
\end{bmatrix} \\
\hat{G}_j &= \begin{bmatrix}
B & 0 \\
0 & K_j G
\end{bmatrix}, \quad \varpi(k) = \begin{bmatrix}
\nu(k) \\
\nu(k)
\end{bmatrix}.
\end{align*}

Then, the augmented system in (18) can be rewritten as the following form:
\begin{align*}
\eta(k+1) &= \sum_{i=1}^{L} h_i(\hat{z}(k)) \sum_{j=1}^{L} h_j(\hat{z}(k)) \sum_{s=1}^{L} h_s(\hat{z}(k)) \times \left( \hat{\eta}_{in}(k) + \hat{E}_{ij} \Delta \Gamma(x(k)) + \hat{G}_j \varpi(k) \right). \quad (19)
\end{align*}
Therefore, the $H_\infty$ estimation performance in (3) can be modified as follows:

\[
\sum_{k=1}^{k_f} e^T(k)Qe(k) = \sum_{k=1}^{k_f} (D(x(k) - \hat{x}(k))^T Q (D(x(k) - \hat{x}(k)))
\]

\[
= \sum_{k=1}^{k_f} \left( \eta^T(k) \tilde{Q} \eta(k) \right)
\]

\[
\leq \rho^2 \left[ \eta(0)^T \tilde{P}\eta(0) + \sum_{k=0}^{k_f-1} (w^T(k)w(k) + \nu^T(k)v(k)) \right].
\]

(20)

where $\tilde{P}_0$ is an initial weighting matrix, and

\[
\tilde{Q} = \begin{bmatrix}
D^T Q D & -D^T Q D \\
-D^T Q D & D^T Q D
\end{bmatrix}.
\]

The following well-known lemmas are useful for our design.

**Lemma 1 (Schur Complements for Nonstrict Inequalities)**[22]: The linear matrix inequality (LMI)

\[
\begin{bmatrix}
M(\bar{x}) & S(\bar{x}) \\
S^T(\bar{x}) & N(\bar{x})
\end{bmatrix} \geq 0
\]

\[(21)\]

where $M(\bar{x}) = M^T(\bar{x})$, $N(\bar{x}) = N^T(\bar{x})$, and $S(\bar{x})$ depends on $\bar{x}$, is equivalent to

\[
N(\bar{x}) \geq 0, \quad M(\bar{x}) - S(\bar{x})N(\bar{x})S^T(\bar{x}) \geq 0
\]

and

\[
S(\bar{x})(I - N(\bar{x})N^T(\bar{x})) = 0
\]

(22)

where $N^T(\bar{x})$ denotes the Moore–Penrose inverse of $N(\bar{x})$. [22]

**Lemma 2 (S-Procedure for Nonstrict Inequalities)**[22]: Consider the following condition on $T_0, \ldots, T_p$:

\[
\zeta^T T_0 \zeta \geq 0 \quad \text{for all } \zeta \quad \text{such that } \zeta^T T_i \zeta \geq 0
\]

\[(23)\]

If there exists $\tau_1 \geq 0, \ldots, \tau_p \geq 0$ such that

\[
T_0 - \sum_{i=1}^{p} \tau_i T_i \geq 0
\]

(24)

then (23) holds.

Then, we get the following result.

**Theorem 1**: In the nonlinear system (1), if there exist $\tau \geq 0$ and a symmetric positive definite matrix $\hat{P} = \hat{P}^T > \hat{Q}$ such that the matrix inequalities

\[
\begin{bmatrix}
\hat{A}_{ij} + \hat{A}_{skj} \\
\hat{E}_{ij} \\
\end{bmatrix}
\]

\[
\frac{2}{\hat{Q}^{1/2}} \\
\hat{G}_{ij} \\
\end{bmatrix}^T \\
\begin{bmatrix}
\hat{P} - \tau \Psi \\
0 \\
\tau I \\
\end{bmatrix} \leq 0
\]

\[(25)\]

hold for $h_i(z(k)), h_j(z(k)), h_s(z(k)) \neq 0$ and $i, j, s = 1, 2, \ldots, L$, where $\Psi = \begin{bmatrix} \hat{\Phi} & 0 \\ 0 & 0 \end{bmatrix}$, then the $H_\infty$ estimation performance in (20) is guaranteed for a prescribed $\rho^2$ at steady state.

**Proof**: From (20), we obtain

\[
\sum_{k=1}^{k_f} e^T(k)Qe(k)
\]

\[
= \sum_{k=1}^{k_f} (\eta^T(k) \tilde{Q} \eta(k))
\]

\[
= \eta^T(0) \tilde{P}(0) \eta(0) - \eta^T(0) \tilde{Q} \eta(0)
\]

\[
+ \eta^T(k_f) \left[ \tilde{Q} - \tilde{P}(k_f) \right] \eta(k_f)
\]

\[
+ \sum_{k=0}^{k_f-1} \left\{ \eta^T(k) \tilde{Q} \eta(k) + \eta^T(k+1) \tilde{P}(k+1) \right\}.
\]

By (19) and with $\tilde{P}_0 = (1/\rho^2) \tilde{P}(0)$, we get

\[
\sum_{k=1}^{k_f} e^T(k)Qe(k)
\]

\[
\leq \rho^2 \left( \eta^T(0) \tilde{P}(0) \eta(0) + \eta^T(k_f) \left[ \tilde{Q} - \tilde{P}(k_f) \right] \eta(k_f) \right)
\]

\[
+ \sum_{k=0}^{k_f-1} \left\{ \eta^T(k) \tilde{Q} \eta(k) \right\}
\]

\[
+ \left( \sum_{i=1}^{L} h_i(z(k)) \sum_{j=1}^{L} h_j(z(k)) \sum_{s=1}^{L} h_s(z(k)) \right)
\]

\[
\times \left( \hat{A}_{ij} \eta(k) + \hat{E}_j \Delta \Gamma(k) + \hat{G}_{ij} \varpi(k) \right)^T
\]

\[
\times \hat{P}(k+1) \left( \sum_{i=1}^{L} h_i(z(k)) \sum_{j=1}^{L} h_j(z(k)) \sum_{m=1}^{L} \right)
\]

\[
\times \left. \left. \left. h_m(z(k)) \left( \hat{A}_{km} \eta(k) + \hat{E}_k \Delta \Gamma(k) \right) + \hat{G}_{kj} \varpi(k) \right) \right\} \eta(k_f)
\]

\[
= \rho^2 \left( \eta^T(0) \tilde{P}(0) \eta(0) + \eta^T(k_f) \left[ \tilde{Q} - \tilde{P}(k_f) \right] \eta(k_f) \right)
\]

\[
+ \frac{1}{\rho^2} \left( \sum_{i=1}^{L} h_i(z(k)) \sum_{j=1}^{L} h_j(z(k)) \sum_{s=1}^{L} h_s(z(k)) \right)
\]

\[
\times \left( \hat{A}_{ij} \eta(k) + \hat{E}_j \Delta \Gamma(k) + \hat{G}_{ij} \varpi(k) \right)^T
\]

\[
\times \hat{P}(k+1)
\]
\[
\sum_{k=0}^{k_f-1} \left\{ \sum_{j=1}^{L} h_i(z(k)) \sum_{j=1}^{L} h_j(\hat{x}(k)) \sum_{s=1}^{L} h_s(\hat{\hat{x}}(k)) \times \left[ \left( \hat{\hat{A}}_{ij} + \hat{\hat{A}}_{isi} \right) \eta(k) + \hat{\hat{E}}_j \Delta \Gamma(k) + \hat{\hat{C}}_j \varpi(k) \right) \right] \right\} - \rho^2 v(k) T v(k) + \rho^2 w(k) T w(k) \}
\]

From (26), if \( \hat{Q} - \hat{P}(k_f) < 0 \) and \( \sum_{k=1}^{k_f-1} \eta^T(k) \hat{Q} \eta(k) \leq \rho^2 \eta^T(0) \hat{P}_0 \eta(0) \) + \( \eta^T(k_f) \left[ \hat{Q} - \hat{P}(k_f) \right] \eta(k_f) \), we can then by the property of membership function in (7), (8), we obtain

\[
\begin{bmatrix}
\eta(k) \\
\Delta \Gamma(k) \\
\varpi(k)
\end{bmatrix}^T \times \left[ \begin{bmatrix}
\frac{\hat{A}_{isi} + \hat{A}_{jis}}{2} & \hat{E}_j & \hat{C}_j \\
\hat{P}(k+1) & 0 & 0 \\
\hat{E}_j & \hat{E}_j & \hat{C}_j
\end{bmatrix} - \begin{bmatrix}
\hat{P}(k) & 0 & 0 \\
0 & \hat{Q}^{1/2} & 0 \\
0 & 0 & \rho^2 I
\end{bmatrix} \right] \begin{bmatrix}
\eta(k) \\
\Delta \Gamma(k) \\
\varpi(k)
\end{bmatrix} \leq 0 \quad (27)
\]

Hence, the \( H_\infty \) estimation performance is achieved with a prescribed \( \rho^2 \). Under this situation, we will encounter the constraint on the variables \( \hat{P}(k) \) satisfying (27) and the constraint in (15). Note that (15) is equivalent to

\[
\sum_{k=0}^{k_f-1} \eta^T(k) \hat{Q} \eta(k) \leq \rho^2 \eta^T(0) \hat{P}_0 \eta(0) + \sum_{k=0}^{k_f-1} (w^T(k) w(k) + v^T(k) v(k)) \]

(28)

where \( \hat{P}_0 \) is an initial weighting matrix.
Applying the S-procedure in Lemma 2, (27) and (29) are equivalent to the existence of $\tau \geq 0$ such that

$$
\begin{bmatrix}
\frac{\hat{A}_{ij} + \hat{A}_{kj}}{2} & \hat{E}_j & \hat{G}_j \\
\hat{P}(k) - \tau \Psi & 0 & 0 \\
0 & \tau I & 0 \\
0 & 0 & \rho^2 I
\end{bmatrix}
\leq 0. \quad (30)
$$

For the convenience of design, only the steady-state case (i.e., $k_f \to \infty$) is considered in this study. Then, let $\hat{P}(k) = \hat{P}(k + 1) = \hat{P}$ for $k \to \infty$ (and thus, $\hat{P} > \hat{Q}$). In this situation, (30) becomes

$$
\begin{bmatrix}
\hat{A}_{ij} + \hat{A}_{kj} & \hat{E}_j & \hat{G}_j \\
\hat{P} - \tau \Psi & 0 & 0 \\
0 & \tau I & 0 \\
0 & 0 & \rho^2 I
\end{bmatrix}
\leq 0. \quad (31)
$$

This completes the proof.

From the analysis above, the most important work associated with the fuzzy $H_{\infty}$ filtering problem consists of solving $\hat{P} = P^T > \hat{Q}$ from (25). In general, it is not easy to analytically determine the common solution $\hat{P} = P^T > \hat{Q}$ for (25). Fortunately, (25) can be converted into linear matrix inequality problem (LMIP) [22]. The LMIP can be solved in a computationally efficient manner using a convex optimization technique such as the interior point method. First, the matrix inequalities in (25) are converted into LMIs using Schur complements as in Lemma 1 by the following procedures. From (25), we obtain

$$
\begin{bmatrix}
\frac{\hat{A}_{ij} + \hat{A}_{kj}}{2} & \hat{E}_j & \hat{G}_j \\
\hat{P} - \tau \Psi & 0 & 0 \\
0 & \tau I & 0 \\
0 & 0 & \rho^2 I
\end{bmatrix}
\leq 0. \quad (32)
$$

For the convenience of design, let

$$
\hat{P} = \begin{bmatrix} P & 0 \\ 0 & P \end{bmatrix} \quad (33)
$$

where $P = P^T > 0$.

With $Z_j = PK_j$, (32) is equivalent to

$$
\begin{bmatrix}
M_{r1} & 0 & P & PB & 0 \\
M_{s1} & M_{s2} & 0 & Z_j & 0 \\
Q_{11} & Q_{12} & 0 & 0 & 0 \\
Q_{21} & Q_{22} & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
P & 0 & 0 & 0 \\
0 & P & 0 & 0 \\
0 & 0 & P & 0 \\
0 & 0 & 0 & P
\end{bmatrix}
\begin{bmatrix}
M_{r1} & 0 & P & PB & 0 \\
M_{s1} & M_{s2} & 0 & Z_j & 0 \\
Q_{11} & Q_{12} & 0 & 0 & 0 \\
Q_{21} & Q_{22} & 0 & 0 & 0
\end{bmatrix}
\leq 0 \quad (34)
$$

where

$$
\begin{align*}
M_{11} &= P - \tau(\Omega + \Phi)_r \\
M_{s1} &= \frac{P\hat{A}_i + \hat{A}_i C_j}{2} \\
M_{s2} &= \frac{P\hat{A}_j - \hat{A}_j C_i}{2} \\
M_{71} &= PA_i \\
M_{72} &= A_j C_i \\
M_{81} &= Z_j C_i + Z_j C_i \\
M_{82} &= Z_j C_i + Z_j C_i
\end{align*}
$$

and

$$
\hat{Q}^{1/2} = \begin{bmatrix} Q_{11} & Q_{12} \\
Q_{21} & Q_{22} \end{bmatrix}.
$$
By the Schur complements in Lemma 1, (34) is equivalent to the following LMIs:

\[
\begin{bmatrix}
M_{11} & 0 & 0 & 0 & 0 \\
0 & P & 0 & 0 & 0 \\
0 & 0 & \tau I & 0 & 0 \\
0 & 0 & 0 & \tau I & 0 \\
0 & 0 & 0 & 0 & \rho^2 I \\
0 & 0 & 0 & 0 & 0 \\
M_{T1} & 0 & P & 0 & PB \\
M_{S1} & M_{S2} & 0 & Z_j & 0 \\
Q_{11} & Q_{12} & 0 & 0 & 0 \\
Q_{21} & Q_{22} & 0 & 0 & 0 \\
M_{T1}^T & M_{S1}^T & Q_{11}^T & Q_{12}^T & Q_{21}^T \\
0 & M_{S2}^T & Q_{12}^T & Q_{22}^T & 0 \\
0 & Z_j^T & 0 & 0 & 0 \\
(PB)^T & 0 & 0 & 0 & 0 \\
0 & (Z_j G)^T & 0 & 0 & 0 \\
P & 0 & 0 & 0 & 0 \\
0 & P & 0 & 0 & 0 \\
0 & 0 & I & 0 & 0 \\
0 & 0 & 0 & I & 0 
\end{bmatrix} \succeq 0. 
\] (35)

To obtain a better robust filtering performance (against disturbances), the attenuation level \( \rho^2 \) can be reduced to the minimum possible value such that (3) is satisfied. Therefore, the design procedure is summarized as follows.

**Design Procedure:**

1. **Step 1)** Construct the fuzzy model in (4).
2. **Step 2)** Evaluate \( \Omega \) and \( \Phi \), given an initial attenuation level \( \rho^2 \).
3. **Step 3)** Solve the LMIP in (35) to obtain \( P \) and \( Z_j \) (thus, obtain \( K_j = D^{-1} Z_j \)).
4. **Step 4)** Decrease the attenuation level \( \rho^2 \), and repeat Step 3–Step 4) until a positive definite \( P \) is not obtained.
5. **Step 5)** Obtain the fuzzy estimator from (17).

**Remark 5:** The LMIP can be solved very efficiently by a convex optimization technique such as the interior point algorithm [22]–[26].

**Remark 6:** \( \Omega \) and \( \Phi \) can be evaluated by off-line simulations. Note that if \( \Omega \) and \( \Phi \) are chosen as

\[
\Omega = \text{diag}(\overline{\tau}) \quad \text{and} \quad \Phi = \text{diag}(\overline{\beta})
\] (36)

then

\[
\begin{align*}
\Delta F(x(k))^T \Delta F(x(k)) & \leq \overline{\tau} \cdot x^T(k) x(k) \\
\Delta H(x(k))^T \Delta H(x(k)) & \leq \overline{\beta} \cdot x^T(k) x(k).
\end{align*}
\] (37)

Therefore, \( \overline{\tau} \) and \( \overline{\beta} \) can be chosen as follows:

\[
\overline{\tau} = \max_{w(k)} \left\{ \frac{\Delta F(x(k))^T \Delta F(x(k))}{x^T(k) x(k)} \right\}
\] (38)

\[
\overline{\beta} = \max_{w(k)} \left\{ \frac{\Delta H(x(k))^T \Delta H(x(k))}{x^T(k) x(k)} \right\}
\] (39)

for all \( x(k) \neq 0 \). By the choice of \( \overline{\tau} \) and \( \overline{\beta} \) in (38) and (39), a worst-case design philosophy is adopted, and the assumptions in (13) and (14) can be guaranteed.

**Remark 7:** The problem of choosing \( A_i \) and \( C_i \) in the fuzzy plant rules can be considered to be that of constructing a Takagi–Sugeno fuzzy model for nonlinear systems and can be described by the following two steps.

1. **Step 1)** Specify the number of fuzzy sets and membership functions for the premise variables in advance.
2. **Step 2)** The weighted recursive least-squares algorithm is applied to determine the parameters of the fuzzy model [27]–[30].

With the above steps, one can obtain \( A_i \) and \( C_i \) to construct the fuzzy model.

### III. Simulation Examples

**Example 1:** Consider the following discrete-time nonlinear chaotic system [31]:

\[
x_1(k+1) = 0.24 x_1(k) - 0.97 (x_2(k) - x_1^2(k)) + w_1(k) \\
x_2(k+1) = 0.97 x_1(k) + 0.24 (x_2(k) - x_1^2(k)) + w_2(k) \\
y(k) = x_1(k) + 0.1 x_1^2(k) + x_2(k) + 0.1 x_2^2(k) + v(k) \\
s(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}
\] (40)

where \( w(k) = [w_1(k), w_2(k)]^T \). In this example, for the convenience of simulation, it is assumed that \( w_1(k) = 0.1 \sin(k) \), \( w_2(k) = 0.1 \cos(k) \), and \( v(k) \) is normal distribution noise with zero mean and variance 0.01, respectively. By simulations, we observe that \( x_1 \) and \( x_2 \in [-1, 1] \). The design purpose is to estimate the state variables of the chaotic system.

Following the design procedure in the above section, the \( H_\infty \) fuzzy filtering design for the discrete-time system in (40) is given by the following steps.

1. **Step 1)**
   - **Rule 1:**
     - If \( x_1 \) is about \(-1.0 \) and \( x_2 \) is about \(-1.0 \), THEN \( x(k+1) = A_1 x(k) + B w(k) \), \( y(k) = C_1 x(k) + G v(k) \).
   - **Rule 2:**
     - If \( x_1 \) is about \(-1.0 \) and \( x_2 \) is about \( 0 \), THEN \( x(k+1) = A_2 x(k) + B w(k) \), \( y(k) = C_2 x(k) + G v(k) \).
   - **Rule 3:**
     - If \( x_1 \) is about \(-1.0 \) and \( x_2 \) is about \( 1.0 \), THEN \( x(k+1) = A_3 x(k) + B w(k) \), \( y(k) = C_3 x(k) + G v(k) \).
   - **Rule 4:**
     - If \( x_1 \) is about \( 0 \) and \( x_2 \) is about \(-1.0 \), THEN \( x(k+1) = A_4 x(k) + B w(k) \), \( y(k) = C_4 x(k) + G v(k) \).
   - **Rule 5:**
     - If \( x_1 \) is about \( 0 \) and \( x_2 \) is about \( 1.0 \), THEN \( x(k+1) = A_5 x(k) + B w(k) \), \( y(k) = C_5 x(k) + G v(k) \).
   - **Rule 6:**
     - If \( x_1 \) is about \( 0 \) and \( x_2 \) is about \( 1.0 \), THEN \( x(k+1) = A_6 x(k) + B w(k) \), \( y(k) = C_6 x(k) + G v(k) \).
Rule 7: 
IF \( x_1 \) is about 1.0 and \( x_2 \) is about -1.0. THEN \( x(k + 1) = A_2x(k) + Bu(k), \ y(k) = C_2x(k) + Du(k) \).

Rule 8: 
IF \( x_1 \) is about 1.0 and \( x_2 \) is about 0. THEN \( x(k + 1) = A_3x(k) + Bu(k), \ y(k) = C_3x(k) + Du(k) \).

Rule 9: 
IF \( x_1 \) is about 1.0 and \( x_2 \) is about 1.0. THEN \( x(k + 1) = A_4x(k) + Bu(k), \ y(k) = C_4x(k) + Du(k) \).

Here and Triangular membership functions are used for Rules 1 to 9 in this example.

Step 2) \( \Omega = 0.01734I \) and \( \Phi = 0.03512I \) are chosen according to Remark 6.

Steps 3 and 4) Solve the LMIP. In this case, we obtain \( \rho = 0.02 \) and the fuzzy estimation gains are found to be

\[
A_1 = \begin{bmatrix} -0.73 & -0.9 \\ 1.21 & 0.24 \end{bmatrix}, \quad A_2 = \begin{bmatrix} -0.73 & -0.9 \\ 1.21 & 0.24 \end{bmatrix}, \quad \ldots
\]

\[
B = I
\]

\[
C_1 = \begin{bmatrix} 0.9 \\ 0.9 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 0.9 \\ 1.0 \end{bmatrix}, \quad C_3 = \begin{bmatrix} 1.0 \\ 1.1 \end{bmatrix}, \quad C_4 = \begin{bmatrix} 1.0 \\ 0.9 \end{bmatrix}, \quad C_5 = \begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix}, \quad C_6 = \begin{bmatrix} 1.0 \\ 1.1 \end{bmatrix}, \quad C_7 = \begin{bmatrix} 1.1 \\ 0.9 \end{bmatrix}, \quad C_8 = \begin{bmatrix} 1.1 \\ 1.0 \end{bmatrix}, \quad C_9 = \begin{bmatrix} 1.1 \\ 1.1 \end{bmatrix}, \quad \text{and} \quad G = 1.
\]

The initial condition in the simulation is assumed to be \( (x_1(0), x_2(0), \dot{x}_1(0), \dot{x}_2(0))^T = (0.1, -0.1, 0, 0)^T \). Fig. 1 shows the estimation errors (squared) for \( x_1(k) \) and \( x_2(k) \) using the proposed \( H_\infty \) fuzzy filter and the EKF. In this example, the statistical conditions \( E[u(k)u(k)^T] = 0.006I \) and \( E[v(k)v(k)^T] = 0.01 \) are used for the design of the EKF. The ratios of error to state of the proposed fuzzy filter and the EKF are listed in Table I. The proposed \( H_\infty \) fuzzy filter is obtained without any information about external disturbances and measurement noise, as long as they are bounded. In fact, the statistical properties of \( u(k) \) and \( v(k) \) change with time and are rarely known beforehand. Obviously, the proposed \( H_\infty \) fuzzy filter is more robust.

Example 2: A trajectory estimation of re-entry vehicles (RV) by radar is shown in Fig. 2. The RV flies over several hundred kilometers along a ballistic trajectory above the Earth. The RV model in radar coordinates centered at the radar site can be expressed as [32, 33]

\[
\begin{align*}
\dot{v}_x &= -\frac{\rho(v_x^2 + v_y^2 + v_z^2)/2\beta}{g} \cos \gamma_1 \sin \gamma_2 + d_x \\
\dot{v}_y &= -\frac{\rho(v_x^2 + v_y^2 + v_z^2)/2\beta}{g} \cos \gamma_1 \cos \gamma_2 + d_y \\
\dot{v}_z &= \frac{\rho(v_x^2 + v_y^2 + v_z^2)/2\beta}{g} \sin \gamma_1 - g + d_z
\end{align*}
\]

where \( v_x, v_y, \text{ and } v_z \) velocity components (m/s) along \( X_R, Y_R, \) and \( Z_R \), respectively;

\( d_x, d_y, \text{ and } d_z \) disturbances;

\( \rho \text{ and } \beta \) air density and ballistic coefficient, respectively;

\( g \) (9.8 (m/sec^2)) gravity force.

\( \gamma_1 \) and \( \gamma_2 \) are defined as follows:

\[
\gamma_1 = \tan^{-1} \left( \frac{-v_y}{\sqrt{v_x^2 + v_z^2}} \right), \quad \gamma_2 = \tan^{-1} \left( \frac{v_x}{v_y} \right).
\]

In this example, \( \beta = 2440 \text{ Kgm/m}^2 \) is assumed to be a constant. The air density is a function of altitude and described as \( \rho = 0.00237 e^{-z/10000} \).
Let the state vector be $x = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6]^T = [x \ y \ z \ v_x \ v_y \ v_z]^T$, where $x$, $y$, and $z$ denote the RV position along $X_R$, $Y_R$, and $Z_R$ respectively. The RV is measured by a precision radar with a sampling rate of 4 $H_z$ [sampling time $T = 0.25$ (s)]. The nonlinear discrete-time state equation can be written as

$$
\begin{align*}
    x_1(k+1) &= x_1(k) + T x_4(k) + u_3(k) \\
    x_2(k+1) &= x_2(k) + T x_3(k) + u_2(k) \\
    x_3(k+1) &= x_3(k) + T x_6(k) + u_3(k) \\
    x_4(k+1) &= x_4(k) - T\left(\frac{[p x_2^2(k) + x_3^2(k) + x_6^2(k)]}{2\beta}\right) \\
    &\times g \cos \gamma_1 \sin \gamma_2 + u_4(k) \\
    x_5(k+1) &= x_5(k) - T\left(\frac{[p x_2^2(k) + x_3^2(k) + x_6^2(k)]}{2\beta}\right) \\
    &\times g \cos \gamma_1 \cos \gamma_2 + u_2(k) \\
    x_6(k+1) &= x_6(k) + T\left(\frac{[p x_2^2(k) + x_3^2(k) + x_6^2(k)]}{2\beta}\right) \\
    &\times g \sin \gamma_1 + u_6(k)
\end{align*}
$$

where

$$
\begin{align*}
    \rho &= 0.0002376e^{-x_3/10000} \\
    \gamma_1 &= \tan^{-1}\left(\frac{-x_6}{\sqrt{x_4^2 + x_5^2}}\right), \quad \gamma_2 = \tan^{-1}\left(\frac{x_4}{x_5}\right) \\
    u_1(k) &= T C_1, \quad u_2(k) = T C_2, \quad u_3(k) = T C_3 \\
    u_4(k) &= T (d_4 + C_4), \quad u_5(k) = T (d_5 + C_5) \\
    u_6(k) &= -T (g - d_6 - C_6)
\end{align*}
$$

and $C_i$ for $i = 1, \ldots, 6$ stand for the process noises [32].

For conventional air defense missile systems, the ground radar is the major instrument for detecting the RV. The measurement equation is

$$
y(k) = C x(k) + v(k)
$$

where $C = I$, and $v(k)$ denotes the measurement noise and jamming. The purpose of this problem is to produce an estimate of the trajectory $x(k)$ from radar output $y(k)$.

Observe that the state variables corresponding to the nonlinear terms are $x_3$, $x_4$, $x_5$, and $x_6$. It is assumed that $x_3 \in [0, 30000]$, $x_4$, and $x_5 \in [-800, 0]$, and $x_6 \in [-2000, 0]$. Therefore, the fuzzy rules are given as follows:

Rule 1: IF $x_3$ is about 0 and $x_4$ is about $-800$ and $x_5$ is about $0$ and $x_6$ is about $-2000$. THEN $x(k+1) = A_1 x(k) + B w(k)$, $y(k) = C_1 x(k) + G v(k)$.

Rule 2: IF $x_3$ is about 0 and $x_4$ is about $-800$ and $x_5$ is about $0$ and $x_6$ is about $0$. THEN $x(k+1) = A_2 x(k) + B w(k)$, $y(k) = C_2 x(k) + G v(k)$.

Rule 3: IF $x_3$ is about 0 and $x_4$ is about $-800$ and $x_5$ is about $0$ and $x_6$ is about $-2000$. THEN $x(k+1) = A_3 x(k) + B w(k)$, $y(k) = C_3 x(k) + G v(k)$.

Rule 4: IF $x_3$ is about 0 and $x_4$ is about $-800$ and $x_5$ is about $0$ and $x_6$ is about 0. THEN $x(k+1) = A_4 x(k) + B w(k)$, $y(k) = C_4 x(k) + G v(k)$.

Rule 5: IF $x_3$ is about 0 and $x_4$ is about 0 and $x_5$ is about $-800$ and $x_6$ is about $-2000$. THEN $x(k+1) = A_5 x(k) + B w(k)$, $y(k) = C_5 x(k) + G v(k)$.

Rule 6: IF $x_3$ is about 0 and $x_4$ is about 0 and $x_5$ is about 0 and $x_6$ is about $0$. THEN $x(k+1) = A_6 x(k) + B w(k)$, $y(k) = C_6 x(k) + G v(k)$.

Rule 7: IF $x_3$ is about 0 and $x_4$ is about 0 and $x_5$ is about $-800$ and $x_6$ is about $-2000$. THEN $x(k+1) = A_7 x(k) + B w(k)$, $y(k) = C_7 x(k) + G v(k)$.

Rule 8: IF $x_3$ is about 0 and $x_4$ is about 0 and $x_5$ is about 0 and $x_6$ is about 0. THEN $x(k+1) = A_8 x(k) + B w(k)$, $y(k) = C_8 x(k) + G v(k)$.

Rule 9: IF $x_3$ is about 30000 and $x_4$ is about $-800$ and $x_5$ is about $-800$ and $x_6$ is about $-2000$. THEN $x(k+1) = A_9 x(k) + B w(k)$, $y(k) = C_9 x(k) + G v(k)$.

Rule 10: IF $x_3$ is about 30000 and $x_4$ is about $-800$ and $x_5$ is about 0 and $x_6$ is about 0. THEN $x(k+1) = A_{10} x(k) + B w(k)$, $y(k) = C_{10} x(k) + G v(k)$.

Rule 11: IF $x_3$ is about 30000 and $x_4$ is about $-800$ and $x_5$ is about 0 and $x_6$ is about 0. THEN $x(k+1) = A_{11} x(k) + B w(k)$, $y(k) = C_{11} x(k) + G v(k)$.

Rule 12: IF $x_3$ is about 30000 and $x_4$ is about $-800$ and $x_5$ is about 0 and $x_6$ is about 0. THEN $x(k+1) = A_{12} x(k) + B w(k)$, $y(k) = C_{12} x(k) + G v(k)$.

Rule 13: IF $x_3$ is about 30000 and $x_4$ is about 0 and $x_5$ is about $-800$ and $x_6$ is about $-2000$. THEN $x(k+1) = A_{13} x(k) + B w(k)$, $y(k) = C_{13} x(k) + G v(k)$.

Rule 14: IF $x_3$ is about 30000 and $x_4$ is about 0 and $x_5$ is about 0 and $x_6$ is about $-2000$. THEN $x(k+1) = A_{14} x(k) + B w(k)$, $y(k) = C_{14} x(k) + G v(k)$.

Rule 15: IF $x_3$ is about 30000 and $x_4$ is about 0 and $x_5$ is about 0 and $x_6$ is about 0. THEN $x(k+1) = A_{15} x(k) + B w(k)$, $y(k) = C_{15} x(k) + G v(k)$.

Rule 16: IF $x_3$ is about 30000 and $x_4$ is about 0 and $x_5$ is about 0 and $x_6$ is about 0. THEN $x(k+1) = A_{16} x(k) + B w(k)$, $y(k) = C_{16} x(k) + G v(k)$.

Where $A_i$ and $C_i$ (for $i = 1, \ldots, 16$) are shown in Appendix A, $B = I$, $G = I$, $w(k) = [w_1(k) \ w_2(k) \ w_3(k) \ w_4(k) \ w_5(k) \ w_6(k)]^T$, and $v(k) = [v_1(k) \ v_2(k) \ v_3(k) \ v_4(k) \ v_5(k) \ v_6(k)]^T$. 

![Tactical ballistic missile geometry.](image-url)
For the convenience of simulations, the disturbances $d_i$ ($i = 4, 5, 6$), the process noises $\zeta_i$ ($i = 1, \ldots, 6$), and measurement noises $v_i$ ($i = 1, \ldots, 6$) are assumed as follows [32]:

$$
\begin{align*}
    d_i &= \mathcal{N}(0, 10) \quad \text{for } i = 4, 5, 6 \\
    \zeta_i &= \mathcal{N}(0, 1) \quad \text{for } i = 1, \ldots, 6 \\
    v_i &= \mathcal{N}(0, 1) \quad \text{for } i = 1, \ldots, 6
\end{align*}
$$

where $\mathcal{N}(\ast, \ast)$ represents the normal distribution with mean $\ast$ and variance $\ast$.

Triangular membership functions are also used for Rules 1 to 16. $\alpha_i$ and $\beta_i$ are chosen according to Remark 6. Solving the corresponding LMIP, we obtain $\rho = 0.0245$

$$
P = \begin{bmatrix}
    9.6 \times 10^{-4} & 7.7 \times 10^{-10} & 5.1 \times 10^{-11} \\
    7.7 \times 10^{-10} & 9.6 \times 10^{-4} & 1.4 \times 10^{-13} \\
    5.1 \times 10^{-11} & 1.4 \times 10^{-13} & 9.6 \times 10^{-4} \\
    -1.2 \times 10^{-4} & -8.1 \times 10^{-8} & -3.5 \times 10^{-8} \\
    -7.9 \times 10^{-9} & -1.2 \times 10^{-4} & -3.5 \times 10^{-8} \\
    3.8 \times 10^{-8} & 3.8 \times 10^{-8} & -1.2 \times 10^{-4} \\
    -1.2 \times 10^{-4} & -7.9 \times 10^{-9} & 3.8 \times 10^{-8} \\
    -8.1 \times 10^{-9} & -1.2 \times 10^{-4} & 3.8 \times 10^{-8} \\
    -3.5 \times 10^{-8} & -3.5 \times 10^{-8} & -1.2 \times 10^{-4} \\
    1.0 \times 10^{-3} & -8.0 \times 10^{-8} & 6.1 \times 10^{-8} \\
    -8.0 \times 10^{-8} & 1.0 \times 10^{-3} & 6.3 \times 10^{-8} \\
    6.1 \times 10^{-8} & 6.3 \times 10^{-8} & 1.0 \times 10^{-8}
\end{bmatrix}
$$

and the fuzzy estimation gains $K_j$ (for $j = 1, \ldots, 16$) are shown in Appendix B. Therefore, we can construct the $H_\infty$ fuzzy estimator as follows:

$$
\hat{x}(k + 1) = \sum_{j=1}^{16} h_j(\hat{x}_3(k), \hat{x}_4(k), \hat{x}_5(k), \hat{x}_6(k)) \\
\times (A_j\hat{x}(k) + K_j(y(k) - g(k))).
$$

In this example, the proposed $H_\infty$ fuzzy filter is employed to deal with the nonlinear estimation problem in radar detecting systems. The estimation errors (squared) of the proposed $H_\infty$ fuzzy filtering and extended Kalman filtering are shown in Fig. 3. In this example, the statistical conditions $E[\varepsilon_i^2(k)] = 10$ ($i = 4, 5, 6$), $E[\zeta_i^2(k)] = 1$ ($i = 1, \ldots, 6$), and $E[v_i^2(k)] = 1$ ($i = 1, \ldots, 6$) are used for the design of the EKF. The ratios of error to state of the proposed fuzzy filter and the EKF are listed in Table II. From the results of this simulation, it also shows that the performance of the proposed method is also more robust than that of EKF.

### IV. CONCLUSIONS

In this paper, based on a Takagi and Sugeno fuzzy model, $H_\infty$ fuzzy filtering problems for nonlinear discrete-time systems with uncertain noises are studied.

An LMI-based design procedure for the $H_\infty$ fuzzy filtering problems of the nonlinear discrete-time systems is developed. The proposed design procedure is very simple. Simulation examples are given to illustrate the design procedure, and the results are satisfactory. Therefore, the proposed method is very suitable for practical filtering applications.

### APPENDIX A

$$
A_1 = \begin{bmatrix}
    1 & 0 & 0 & 0.25 \\
    0 & 1 & 0 & 0 \\
    0 & 0 & 1 & 0 \\
    0 & 0 & 4.5165 \times 10^{-5} & 1.0011 \\
    0 & 0 & 4.4299 \times 10^{-5} & 6.2207 \times 10^{-5} \\
    0 & 0 & -1.625 \times 10^{-4} & -3.0459 \times 10^{-4} \\
    0 & 0.25 & 0 & 0 \\
    0 & 0 & 0.25 & 6.0868 \times 10^{-5} \\
    0 & 0 & 2.5194 \times 10^{-4} & 1.0010 \\
    0 & 0 & 2.4922 \times 10^{-4} & 9.9841 \times 10^{-1}
\end{bmatrix}
$$

### TABLE II

<table>
<thead>
<tr>
<th>$i$</th>
<th>EKF</th>
<th>Fuzzy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$4.3462 \times 10^{-4}$</td>
<td>$3.3112 \times 10^{-4}$</td>
</tr>
<tr>
<td>2</td>
<td>$3.5539 \times 10^{-4}$</td>
<td>$2.7872 \times 10^{-4}$</td>
</tr>
<tr>
<td>3</td>
<td>$3.9780 \times 10^{-4}$</td>
<td>$1.9474 \times 10^{-4}$</td>
</tr>
<tr>
<td>4</td>
<td>0.0047</td>
<td>0.0039</td>
</tr>
<tr>
<td>5</td>
<td>0.0033</td>
<td>0.0028</td>
</tr>
<tr>
<td>6</td>
<td>0.0044</td>
<td>0.0038</td>
</tr>
</tbody>
</table>
\begin{align*}
A_2 &= \begin{bmatrix}
1 & 0 & 0 & 0.25 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 3.0231 \times 10^{-5} & 1.0008 \\
0 & 0 & 2.9483 \times 10^{-5} & 1.5197 \times 10^{-4} \\
0 & 0 & -3.7200 \times 10^{-5} & -1.7810 \times 10^{-4} \\
0.25 & 0 & 0 \\
0 & 0.25 \\
1.5188 \times 10^{-1} & 2.0011 \times 10^{-1} \\
1.0008 & 1.9751 \times 10^{-1} \\
-1.7889 \times 10^{-1} & 9.9897 \times 10^{-1}
\end{bmatrix}
\end{align*}

\begin{align*}
A_3 &= \begin{bmatrix}
1 & 0 & 0 & 0.25 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 4.3143 \times 10^{-5} & 1.0010 \\
0 & 0 & 1.8527 \times 10^{-5} & 1.8905 \times 10^{-5} \\
0 & 0 & -1.1098 \times 10^{-4} & -3.4566 \times 10^{-1} \\
0 & 0.25 & 0 \\
0 & 0.25 \\
3.9519 \times 10^{-5} & 2.6198 \times 10^{-1} \\
1.0010 & 1.1043 \times 10^{-1} \\
-1.8974 \times 10^{-1} & 9.9842 \times 10^{-1}
\end{bmatrix}
\end{align*}

\begin{align*}
A_4 &= \begin{bmatrix}
1 & 0 & 0 & 0.25 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 2.6781 \times 10^{-5} & 1.0008 \\
0 & 0 & 1.1003 \times 10^{-5} & 6.9047 \times 10^{-5} \\
0 & 0 & -3.3992 \times 10^{-5} & -2.2252 \times 10^{-1} \\
0 & 0.25 & 0 \\
0 & 0.25 \\
1.0497 \times 10^{-1} & 2.2236 \times 10^{-1} \\
1.0007 & 9.2939 \times 10^{-5} \\
-1.964 \times 10^{-1} & 9.9001 \times 10^{-1}
\end{bmatrix}
\end{align*}

\begin{align*}
A_5 &= \begin{bmatrix}
1 & 0 & 0 & 0.25 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 1.9047 \times 10^{-5} & 1.0010 \\
0 & 0 & 4.2281 \times 10^{-5} & 3.9901 \times 10^{-5} \\
0 & 0 & -1.1100 \times 10^{-4} & -1.8792 \times 10^{-1} \\
0 & 0.25 & 0 \\
0 & 0.25 \\
1.7910 \times 10^{-5} & 1.1249 \times 10^{-1} \\
1.0001 & 2.3937 \times 10^{-1} \\
-3.4698 \times 10^{-1} & 9.9842 \times 10^{-1}
\end{bmatrix}
\end{align*}

\begin{align*}
A_6 &= \begin{bmatrix}
1 & 0 & 0 & 0.25 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 1.1991 \times 10^{-5} & 1.0007 \\
0 & 0 & 2.6087 \times 10^{-5} & 1.0361 \times 10^{-4} \\
0 & 0 & -3.3906 \times 10^{-5} & -1.1841 \times 10^{-1}
\end{bmatrix}
\end{align*}

\begin{align*}
A_7 &= \begin{bmatrix}
1 & 0 & 0 & 0.25 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 1.8213 \times 10^{-5} & 1.0009 \\
0 & 0 & 1.7716 \times 10^{-5} & 1.0423 \times 10^{-5} \\
0 & 0 & -1.0515 \times 10^{-4} & -2.2054 \times 10^{-1} \\
0 & 0.25 & 0 \\
0 & 0.25 \\
9.6696 \times 10^{-6} & 1.1449 \times 10^{-1} \\
1.0009 & 1.262 \times 10^{-1} \\
-2.2243 \times 10^{-1} & 9.8841 \times 10^{-1}
\end{bmatrix}
\end{align*}

\begin{align*}
A_8 &= \begin{bmatrix}
1 & 0 & 0 & 0.25 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 1.0261 \times 10^{-5} & 1.0006 \\
0 & 0 & 9.9017 \times 10^{-6} & 5.0204 \times 10^{-5} \\
0 & 0 & -2.9528 \times 10^{-5} & -1.6973 \times 10^{-1} \\
0 & 0.25 & 0 \\
0 & 0.25 \\
5.0451 \times 10^{-5} & 1.6851 \times 10^{-1} \\
1.0006 & 1.0632 \times 10^{-1} \\
-1.7144 \times 10^{-1} & 9.9004 \times 10^{-1}
\end{bmatrix}
\end{align*}

\begin{align*}
A_9 &= \begin{bmatrix}
1 & 0 & 0 & 0.25 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 1.7973 \times 10^{-5} & 1.0005 \\
0 & 0 & 1.7402 \times 10^{-5} & 1.5374 \times 10^{-4} \\
0 & 0 & -4.5257 \times 10^{-5} & -4.4626 \times 10^{-1} \\
0 & 0.25 & 0 \\
0 & 0.25 \\
1.5746 \times 10^{-4} & 1.3707 \times 10^{-1} \\
1.0005 & 1.3440 \times 10^{-1} \\
-4.4925 \times 10^{-1} & 9.9935 \times 10^{-1}
\end{bmatrix}
\end{align*}

\begin{align*}
A_{10} &= \begin{bmatrix}
1 & 0 & 0 & 0.25 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 1.1397 \times 10^{-5} & 1.0004 \\
0 & 0 & 1.0946 \times 10^{-5} & 1.5944 \times 10^{-4} \\
0 & 0 & -1.3734 \times 10^{-5} & -1.9582 \times 10^{-1} \\
0 & 0.25 & 0 \\
0 & 0.25 \\
1.6364 \times 10^{-4} & 8.5203 \times 10^{-5} \\
1.0004 & 8.3142 \times 10^{-5} \\
-1.9708 \times 10^{-1} & 9.9964 \times 10^{-1}
\end{bmatrix}
\end{align*}
\[ A_{11} = \begin{bmatrix}
1 & 0 & 0 & 0.25 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & -1.6335 \times 10^{-5} & 1.0005 \\
0 & 0 & 6.9401 \times 10^{-6} & 7.6354 \times 10^{-5} \\
0 & 0 & -4.0060 \times 10^{-5} & -5.2400 \times 10^{-1} \\
0 & 0.25 & 0 & 0 \\
0 & 0 & 0.25 & 0 \\
8.5246 \times 10^{-5} & 1.5100 \times 10^{-1} & 1.0004 & 6.3072 \times 10^{-5} \\
-2.4254 \times 10^{-1} & 9.9932 \times 10^{-1} & 0 & 0
\end{bmatrix} \]

\[ A_{12} = \begin{bmatrix}
1 & 0 & 0 & 0.25 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 9.2203 \times 10^{-6} & 1.0004 \\
0 & 0 & 3.9552 \times 10^{-6} & 8.0439 \times 10^{-5} \\
0 & 0 & -1.1360 \times 10^{-5} & -2.3782 \times 10^{-1} \\
0 & 0.25 & 0 & 0 \\
0 & 0 & 0.25 & 0 \\
9.3198 \times 10^{-5} & 9.5933 \times 10^{-5} & 1.0002 & 4.0195 \times 10^{-5} \\
-1.1144 \times 10^{-1} & 9.9965 \times 10^{-1} & 0 & 0
\end{bmatrix} \]

\[ A_{13} = \begin{bmatrix}
1 & 0 & 0 & 0.25 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 7.2306 \times 10^{-6} & 1.0004 \\
0 & 0 & 1.5829 \times 10^{-5} & 8.1566 \times 10^{-5} \\
0 & 0 & -4.9988 \times 10^{-5} & -2.3911 \times 10^{-1} \\
0 & 0.25 & 0 & 0 \\
0 & 0 & 0.25 & 0 \\
7.8669 \times 10^{-5} & 6.5705 \times 10^{-5} & 1.0005 & 1.4807 \times 10^{-1} \\
-5.2769 \times 10^{-1} & 9.9932 \times 10^{-1} & 0 & 0
\end{bmatrix} \]

\[ A_{14} = \begin{bmatrix}
1 & 0 & 0 & 0.25 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 4.1507 \times 10^{-6} & 1.0002 \\
0 & 0 & 8.8478 \times 10^{-6} & 8.9070 \times 10^{-5} \\
0 & 0 & -1.1375 \times 10^{-5} & -1.0988 \times 10^{-1} \\
0 & 0.25 & 0 & 0 \\
0 & 0 & 0.25 & 0 \\
8.3027 \times 10^{-5} & 4.1506 \times 10^{-5} & 1.0004 & 9.3567 \times 10^{-5} \\
-2.3895 \times 10^{-1} & 9.9965 \times 10^{-1} & 0 & 0
\end{bmatrix} \]

\[ A_{15} = \begin{bmatrix}
1 & 0 & 0 & 0.25 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 6.3493 \times 10^{-6} & 1.0003 \\
0 & 0 & 6.0877 \times 10^{-6} & 4.2057 \times 10^{-5} \\
0 & 0 & -3.3314 \times 10^{-5} & -2.9444 \times 10^{-1} \\
0 & 0 & 0.25 & 0 \\
0 & 0 & 0.25 & 0 \\
8.3027 \times 10^{-5} & 4.1506 \times 10^{-5} & 1.0004 & 9.3567 \times 10^{-5} \\
-2.3895 \times 10^{-1} & 9.9965 \times 10^{-1} & 0 & 0
\end{bmatrix} \]

\[ C_i = I \text{ (for } i = 1, \ldots, 16) \]

**APPENDIX B**

\[ K_1 = \begin{bmatrix}
3.55 \times 10^{-1} & 2.87 \times 10^{-7} & 2.92 \times 10^{-7} \\
2.67 \times 10^{-7} & 3.55 \times 10^{-1} & 2.80 \times 10^{-7} \\
8.36 \times 10^{-8} & 6.43 \times 10^{-8} & 3.55 \times 10^{-1} \\
8.91 \times 10^{-3} & 2.21 \times 10^{-6} & 1.73 \times 10^{-5} \\
-5.01 \times 10^{-6} & 9.91 \times 10^{-3} & 1.69 \times 10^{-5} \\
9.58 \times 10^{-2} & 3.63 \times 10^{-6} & -4.51 \times 10^{-6} \\
3.70 \times 10^{-6} & 9.58 \times 10^{-2} & -4.60 \times 10^{-6} \\
-3.20 \times 10^{-6} & -3.27 \times 10^{-6} & 9.57 \times 10^{-2} \\
3.47 \times 10^{-1} & 2.89 \times 10^{-5} & 7.96 \times 10^{-5} \\
2.94 \times 10^{-5} & 3.47 \times 10^{-1} & 7.85 \times 10^{-5} \\
-1.14 \times 10^{-1} & -1.15 \times 10^{-1} & 3.47 \times 10^{-1}
\end{bmatrix} \]

\[ K_2 = \begin{bmatrix}
3.55 \times 10^{-1} & 7.11 \times 10^{-7} & 9.71 \times 10^{-7} \\
6.98 \times 10^{-7} & 3.55 \times 10^{-1} & 9.61 \times 10^{-7} \\
-4.68 \times 10^{-7} & -4.24 \times 10^{-7} & 3.55 \times 10^{-1} \\
8.91 \times 10^{-8} & 3.87 \times 10^{-6} & 1.45 \times 10^{-5} \\
3.82 \times 10^{-6} & 8.91 \times 10^{-3} & 1.42 \times 10^{-5} \\
-5.16 \times 10^{-6} & -5.22 \times 10^{-6} & 8.84 \times 10^{-3} \\
9.58 \times 10^{-2} & 1.32 \times 10^{-6} & -7.13 \times 10^{-6} \\
1.37 \times 10^{-6} & 9.58 \times 10^{-2} & -7.24 \times 10^{-6} \\
-1.50 \times 10^{-6} & -1.58 \times 10^{-6} & 9.57 \times 10^{-2} \\
3.47 \times 10^{-1} & 4.92 \times 10^{-5} & 5.67 \times 10^{-5} \\
4.92 \times 10^{-5} & 3.47 \times 10^{-1} & 5.56 \times 10^{-5} \\
-7.49 \times 10^{-5} & -7.54 \times 10^{-5} & 3.47 \times 10^{-1}
\end{bmatrix} \]

\[ K_3 = \begin{bmatrix}
3.55 \times 10^{-1} & 7.87 \times 10^{-8} & 9.76 \times 10^{-8} \\
1.69 \times 10^{-7} & 3.55 \times 10^{-1} & 9.60 \times 10^{-7} \\
1.00 \times 10^{-7} & -1.00 \times 10^{-7} & 3.55 \times 10^{-1} \\
8.91 \times 10^{-8} & 1.51 \times 10^{-6} & 1.62 \times 10^{-5} \\
1.65 \times 10^{-6} & 8.91 \times 10^{-3} & 1.04 \times 10^{-5} \\
-5.06 \times 10^{-6} & -6.49 \times 10^{-6} & 8.81 \times 10^{-3}
\end{bmatrix} \]
\[
K_1 = \begin{bmatrix}
9.58 \times 10^{-2} & 4.66 \times 10^{-6} & -3.64 \times 10^{-6} \\
4.33 \times 10^{-6} & 9.58 \times 10^{-2} & -6.69 \times 10^{-6} \\
-3.14 \times 10^{-6} & 1.11 \times 10^{-6} & 9.57 \times 10^{-7} \\
3.47 \times 10^{-1} & 2.55 \times 10^{-5} & 8.50 \times 10^{-3} \\
1.84 \times 10^{-5} & 3.47 \times 10^{-1} & 3.14 \times 10^{-3} \\
-1.26 \times 10^{-1} & -7.29 \times 10^{-5} & 3.47 \times 10^{-1}
\end{bmatrix}
\]

\[
K_8 = \begin{bmatrix}
3.55 \times 10^{-1} & 2.29 \times 10^{-7} & 1.09 \times 10^{-6} \\
2.17 \times 10^{-7} & 3.55 \times 10^{-1} & 1.07 \times 10^{-6} \\
-1.14 \times 10^{-6} & -1.15 \times 10^{-6} & 3.55 \times 10^{-1} \\
8.90 \times 10^{-3} & 1.99 \times 10^{-6} & 8.43 \times 10^{-6} \\
1.94 \times 10^{-6} & 8.90 \times 10^{-3} & 8.21 \times 10^{-6} \\
-6.42 \times 10^{-6} & -6.47 \times 10^{-6} & 8.84 \times 10^{-3}
\end{bmatrix}
\]

\[
K_9 = \begin{bmatrix}
9.58 \times 10^{-2} & 3.93 \times 10^{-6} & -6.99 \times 10^{-6} \\
3.55 \times 10^{-1} & 9.58 \times 10^{-2} & -7.08 \times 10^{-6} \\
-1.82 \times 10^{-6} & 1.70 \times 10^{-6} & 9.57 \times 10^{-7} \\
3.47 \times 10^{-1} & 2.67 \times 10^{-5} & 3.00 \times 10^{-3} \\
2.66 \times 10^{-5} & 3.47 \times 10^{-1} & 2.91 \times 10^{-3} \\
-6.66 \times 10^{-5} & -6.74 \times 10^{-5} & 3.47 \times 10^{-1}
\end{bmatrix}
\]

\[
K_{10} = \begin{bmatrix}
3.55 \times 10^{-1} & 7.19 \times 10^{-7} & -2.87 \times 10^{-7} \\
7.25 \times 10^{-7} & 3.55 \times 10^{-1} & -3.08 \times 10^{-7} \\
-8.54 \times 10^{-7} & -8.68 \times 10^{-7} & 3.55 \times 10^{-1} \\
8.90 \times 10^{-3} & 3.92 \times 10^{-6} & 6.96 \times 10^{-6} \\
3.91 \times 10^{-6} & 8.90 \times 10^{-3} & 6.66 \times 10^{-6} \\
-7.29 \times 10^{-6} & -7.37 \times 10^{-6} & 8.84 \times 10^{-3}
\end{bmatrix}
\]

\[
K_{11} = \begin{bmatrix}
9.58 \times 10^{-2} & 1.26 \times 10^{-6} & -9.58 \times 10^{-7} \\
1.24 \times 10^{-6} & 9.58 \times 10^{-2} & -1.01 \times 10^{-6} \\
1.65 \times 10^{-6} & 1.56 \times 10^{-6} & 9.57 \times 10^{-2} \\
3.47 \times 10^{-1} & 5.07 \times 10^{-5} & 5.56 \times 10^{-5} \\
4.94 \times 10^{-5} & 3.47 \times 10^{-1} & 5.46 \times 10^{-5} \\
-1.47 \times 10^{-1} & -1.48 \times 10^{-1} & 3.47 \times 10^{-1}
\end{bmatrix}
\]

\[
K_{12} = \begin{bmatrix}
3.55 \times 10^{-1} & 7.46 \times 10^{-7} & 9.67 \times 10^{-7} \\
7.54 \times 10^{-7} & 3.55 \times 10^{-1} & 9.53 \times 10^{-7} \\
-1.24 \times 10^{-6} & -1.25 \times 10^{-6} & 3.55 \times 10^{-1} \\
8.90 \times 10^{-3} & 4.03 \times 10^{-6} & 8.29 \times 10^{-6} \\
4.02 \times 10^{-6} & 8.90 \times 10^{-3} & 8.06 \times 10^{-6} \\
-6.86 \times 10^{-6} & -6.92 \times 10^{-6} & 8.86 \times 10^{-3}
\end{bmatrix}
\]

\[
K_3 = \begin{bmatrix}
3.55 \times 10^{-1} & 1.84 \times 10^{-7} & 9.76 \times 10^{-7} \\
6.08 \times 10^{-1} & 3.55 \times 10^{-1} & 8.53 \times 10^{-8} \\
-9.97 \times 10^{-7} & 9.11 \times 10^{-8} & 3.55 \times 10^{-1} \\
8.91 \times 10^{-3} & 1.70 \times 10^{-6} & 1.07 \times 10^{-5} \\
1.45 \times 10^{-6} & 8.91 \times 10^{-3} & 1.58 \times 10^{-5} \\
-6.43 \times 10^{-6} & -5.13 \times 10^{-6} & 8.81 \times 10^{-3}
\end{bmatrix}
\]

\[
K_6 = \begin{bmatrix}
3.55 \times 10^{-1} & 4.87 \times 10^{-7} & 1.34 \times 10^{-6} \\
3.00 \times 10^{-7} & 3.55 \times 10^{-1} & 7.52 \times 10^{-7} \\
-1.19 \times 10^{-6} & -3.59 \times 10^{-7} & 3.55 \times 10^{-1} \\
8.91 \times 10^{-3} & 2.82 \times 10^{-6} & 9.58 \times 10^{-6} \\
2.45 \times 10^{-6} & 8.91 \times 10^{-3} & 1.27 \times 10^{-5} \\
-6.37 \times 10^{-6} & -5.13 \times 10^{-6} & 8.84 \times 10^{-3}
\end{bmatrix}
\]

\[
K_7 = \begin{bmatrix}
3.55 \times 10^{-1} & 4.17 \times 10^{-8} & 8.20 \times 10^{-7} \\
2.49 \times 10^{-8} & 3.55 \times 10^{-1} & 8.02 \times 10^{-7} \\
-9.94 \times 10^{-7} & -1.00 \times 10^{-6} & 3.55 \times 10^{-1} \\
8.91 \times 10^{-3} & 1.25 \times 10^{-6} & 1.00 \times 10^{-5} \\
1.19 \times 10^{-6} & 8.91 \times 10^{-3} & 9.80 \times 10^{-6} \\
-6.55 \times 10^{-6} & -6.61 \times 10^{-6} & 8.81 \times 10^{-3}
\end{bmatrix}
\]

\[
K_9 = \begin{bmatrix}
3.55 \times 10^{-1} & 3.70 \times 10^{-7} & 7.67 \times 10^{-7} \\
4.22 \times 10^{-7} & 3.55 \times 10^{-1} & 1.40 \times 10^{-6} \\
-1.21 \times 10^{-6} & -1.57 \times 10^{-6} & 3.55 \times 10^{-1} \\
8.90 \times 10^{-3} & 2.61 \times 10^{-6} & 7.12 \times 10^{-6} \\
2.68 \times 10^{-6} & 8.90 \times 10^{-3} & 7.01 \times 10^{-6} \\
-6.91 \times 10^{-6} & -7.13 \times 10^{-6} & 8.86 \times 10^{-3}
\end{bmatrix}
\]
REFERENCES


Chung-Shi Tseng (M’01) received the B.S. degree from the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, R.O.C., the M.S. degree from the Department of Electrical Engineering and Computer Engineering, University of New Mexico, Albuquerque, and the Ph.D. degree in electrical engineering, from National Tsing-Hua University, Hsinchu, Taiwan.

He is now an Associate Professor at Ming Hsin Institute of Technology and Commerce, Hsinchu. His research interests are in signal processing and control.

Bor-Sen Chen (M’82–SM’89) received the B.S. degree from Tatung Institute of Technology, Taipei, Taiwan, R.O.C., in 1970, the M.S. degree from National Central University, Taipei, in 1973, and the Ph.D. degree from the University of Southern California, Los Angeles, in 1982.

He was a Lecturer, Associate Professor, and Professor at Tatung Institute of Technology from 1973 to 1987. He is now a Professor at National Tsing Hua University, Hsinchu, Taiwan. His current research interests include control and signal processing.

Dr. Chen has received the Distinguished Research Award from National Science Council of Taiwan four times. He is a Research Fellow of the National Science Council and the Chair of the Outstanding Scholarship Foundation.