F.S.A.I., Vol. 8, Nos. 1–3, pp. 5–23, 2002

An On-line Approach to Fuzzy Modeling

Euntai Kim, Heesung Lee

School of Electrical and Electronic Engr.,
Yonsei University
134 Shinchon-dong, Seodaemun-gu, Seoul, Korea, 120-749.

Abstract: This paper presents an explanation regarding on-line identification of a fuzzy system. The fuzzy system to be identified is assumed to be in the type of singleton consequent parts (type) and be represented by a linear combination of fuzzy basis functions (FBF's). For on-line identification, squared-cosine (SCOS) membership functions are introduced to facilitate the parameter tuning. Then the parameters of the fuzzy system are identified on-line by the gradient search method (GS), the extended Kalman filter (EKF) and the hybrid approach (HYB). This paper differs from the previous works regarding the fuzzy modeling in that (1) it is concerned with the on-line identification and (2) not only the consequent parts but also the premise parts are jointly tuned during the parameter adaptation. Finally computer simulations are used to help explain the suggested methods and also differentiate between previous works.

Key word: fuzzy system, on-line identification, squared-cosine membership function, gradient search, extended Kalman filter

1. Introduction

Over the last two decades, the fuzzy theory suggested by Zadeh has received significant attention from many researchers and has been successfully applied to various fields. In recent years, the fuzzy theory has grown to the concept of 'Soft computing' [1] in association with neural network, evolutionary algorithm, artificial intelligence, etc. In essence, we have come a long way since the conventional concept of 'hard computing' or 'numerical method'.

In most applications of the fuzzy theory (e.g., the application in control systems or prediction systems), the main design objective is to construct a fuzzy system to approximate a desired control system or process. However, the parameters of the system are usually decided by the skilled and experienced operators in heuristic manners. Until now, despite the fast development and wide application of the fuzzy system theory, only a few studies on the automatic identification of the fuzzy system have been conducted.

For example, Pedrycz [2, 3] suggested the identification algorithms of fuzzy relational model. Sugeno and his colleagues proposed the identification of so-called TSK (Takagi-Sugeno-Kang) fuzzy system [4-6]. Recently, other researchers also participate in the identification of the TSK fuzzy system [7-12]. Sugeno and Yasukawa reported qualitative modeling of a fuzzy system [13] and some researchers attempted to identify the fuzzy system via the neural-network-based approaches [14-17].

However, most of these are the off-line algorithms and it takes much time for them to identify the fuzzy system. Owing to the off-line property, the algorithms mentioned above couldn't be applied to the situations where real-time or on-line data processing is required such as in adaptive control and signal processing. Even though on-line successive fuzzy modeling was suggested in [18], it cannot be viewed as an on-line algorithm since it requires an initial fuzzy model, which is fully constructed in advance by other algorithms.

To solve this problem, this paper proposes on-line identification algorithms of a fuzzy system. First, squared-cosine (SCOS) membership functions (MFs) are introduced to make the fuzzy system smoother (more differentiable) than the ones with the triangular MFs and to prevent the redundant overlapping of MFs in the universe of discourse, which may happen in case of gaussian MFs. Then, the parameters of the fuzzy system are identified on-line by three different methods: the gradient search method, the extended Kalman filter and their hybrid approach.

This paper is composed as follows: In Section II, the fuzzy system is reviewed briefly and the structure of the fuzzy system and the SCOS MF are proposed. In Section III, the on-line algorithms are presented. In Section IV, some benchmark examples and the results of the computer simulation are provided to compare the algorithms suggested herein with the previous works and to demonstrate the validity of the suggested algorithms. Finally, in Section V, the concluding remarks are presented.

2. Fuzzy Systems and some properties

A. Types of Fuzzy System and Notation

Since Mamdani applied fuzzy logic to a practical system [19], many different fuzzy systems have been used with different structures, MFs, etc. Generally, these systems can be categorized into the following three types according to their consequent parts:

Type Ⅰ: (Type of the linguistic consequent parts)

$$\mathrm{R}^{n}\text{: If } x_1 \text{ is } \mathrm{A}_1^n \text{ and } x_2 \text{ is } \mathrm{A}_2^n, \cdots, x_m \text{ is } \mathrm{A}_m^n, \text{ then y is } C^n$$

Type Ⅱ: (Type of the singleton consequent parts)

$$\mathrm{R}^{n}\text{: If } x_1 \text{ is } \mathrm{A}_1^n \text{ and } x_2 \text{ is } \mathrm{A}_2^n, \cdots, x_m \text{ is } \mathrm{A}_m^n, \text{ then y is } \theta^n$$

Type Ⅲ: (Type of the linear consequent parts)

$$\mathrm{R}^{n}\text{: If } x_1 \text{ is } \mathrm{A}_1^n \text{ and } x_2 \text{ is } \mathrm{A}_2^n, \cdots, x_m \text{ is } \mathrm{A}_m^n,$$
$$\text{then } y = a_0^n + a_1^n x_1 + a_2^n x_2 + \cdots + a_m^n$$

where A_i^n and C^n are fuzzy variables, θ^n is a singleton and a_i^n's are the coefficients of consequent parts.

Remark 1.

(1)Type Ⅱ can be viewed as the intermediate type of type Ⅰ and type Ⅲ.

(2)Type Ⅲ is suggested by Sugeno and his colleagues [4], and usually called TSK (Takagi-Sugeno-Kang) fuzzy system or Sugeno-type fuzzy system.

In this paper, the following type $\prod$ fuzzy system $\Im$ with the singleton consequent parts is considered:

$$\mathrm{R}^{n_1 n_2 \cdots n_m}\text{: If } x_1 \text{ is } \mathrm{A}_1^{n_1} \text{ and } x_2 \text{ is } \mathrm{A}_2^{n_2}, \cdots, x_m \text{ is } \mathrm{A}_m^{n_m},$$
$$\text{then } y \text{ is } \theta^{n_1, n_2, \cdots, n_m} \tag{1}$$

$$(n_1 = 1, \cdots \mathrm{M}_1,\ n_2 = 1, \cdots \mathrm{M}_2,\ \cdots\cdots,\ n_m = 1, \cdots \mathrm{M}_m)$$

where $x \equiv [x_1, x_2, \ldots, x_m]$ and y are the input and the output variables of the fuzzy system $\Im$, respectively. As noted in (1), the fuzzy system $\Im$ is assumed to be $\Im : \bigcup \subset \mathbf{R}^m \to V \subset \mathbf{R}$, where $U = U_1 \times U_2 \times \ldots \times U_m \subset \mathbf{R}^m$ and the number of fuzzy rules is $\prod_{i=1}^{m} M_i$. Generally speaking, a fuzzy system consists of four principal components: a fuzzifier, a fuzzy rule base (implications), a fuzzy inference engine and a defuzzifier [20-22]. If the fuzzifier is a singleton, the T-norm in fuzzy implication and inference is a product inference and the defuzzifier is the center average, then the fuzzy system of (1) can be formulated by (2-1) through (2-8).

$$y_m = \Im(x) = \Im(x_1, x_2, \ldots, x_m)$$

$$\sum_{n_1}^{M_1} \sum_{n_{21}}^{M_2} \cdots \sum_{n_m}^{M_m} \left\{ \frac{\prod_{i}^{m} A_i^{n_i}(x_i)}{\sum_{n_1}^{M_1} \sum_{n_2}^{M_2} \cdots \sum_{n_m}^{M_{m1}} \prod_{i=1}^{m} A_i^{n_i}(x_i)} \right\} \theta^{n_1, n_2, \ldots, n_m} \tag{2-1}$$

$$= \sum_{n=1}^{m} \left\{ \frac{\Omega^n(x)}{\sum_{n=1}^{m} \Omega^n(x)} \right\} \theta^n = \sum_{n=1}^{m} \{\xi^n(x)\} \theta^n \tag{2-2}$$

where the following notations are introduced for simplification:

$$n \equiv [n_1, n_2, \cdots, n_m] \tag{2-3}$$

$$M \equiv \left[M_1, M_2, \ldots, M_m\right] \tag{2-4}$$

$$\sum_{n_1=1}^{M} \equiv \sum_{n_1=1}^{M} \sum_{n_2=1}^{M} \cdots \sum_{n_m=1}^{M} \tag{2-5}$$

$$\Omega^n(x) \equiv \prod_{i=1}^{m} A_1^n(x_i) \tag{2-6}$$

$$\xi^n(x) \equiv \frac{\Omega^n(x)}{\sum_{n-1}^{M} \Omega^n(x)} \quad \text{(FBF (Fuzzy Basis Function))} \tag{2-7}$$

B. Squared Cosine Membership Functions

Type II fuzzy system usually uses gaussian, triangular or trapezoidal MFs. For example, Wang used gaussian MFs in [23] and Zeng and Singh adopted triangular and trapezoidal MFs in [21, 22]. As an alternative choice for the MFs, SCOS MF is proposed in this subsection. The SCOS MF is defined as in (3) and is shown with the universe of discourse in Fig. 1.

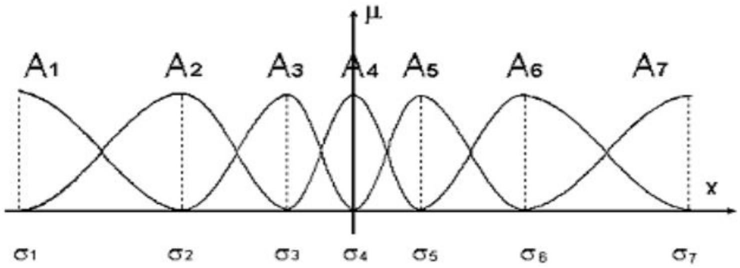

Fig. 1 – The universe of discourse and squared-cosine membership functions

$$A_i^{n_i}(x_i) = \begin{cases} \cos^2 \dfrac{\pi\left(x_i - \sigma_i^n\right)}{2d_i^{m-1}} & if \quad \sigma_i^{n-1} \le x_i < \sigma_i^n \\ \cos^2 \dfrac{\pi\left(x_i - \sigma_i^n\right)}{2d_i^{m-1}} & if \quad \sigma_i^n \le x_i < \sigma_i^{n+1} \\ 0 & elsewhere \end{cases} \tag{3}$$

where $d_i^m \equiv \sigma_i^{n+1} - \sigma_i^n$ (n = 1, ⋯, M_i)

Remark 2.

(1) Compared to the triangular or the trapezoidal MFs, the SCOS MF is more smooth at every point and differentiable even at σ_i^{n-1}, σ_i^n and σ_i^{n+1}. The differentiability at σ_i^{n-1}, σ_i^n and σ_i^{n+1} can be clearly demonstrated by showing that

$$\lim_{x_i \to \sigma_i^n -} \frac{d A_i^n(x_i)}{d x_i} = \lim_{x_i \to \sigma_i^n +} \frac{d A_i^n(x_i)}{d x_i}$$

$$\lim_{x_i \to \sigma_i^{n-1} -} \frac{d A_i^n(x_i)}{d x_i} = \lim_{x_i \to \sigma_i^{n-1} +} \frac{d A_i^n(x_i)}{d x_i}$$

$$\lim_{x_i \to \sigma_i^{n+1} -} \frac{d A_i^n(x_i)}{d x_i} = \lim_{x_i \to \sigma_i^{n+1} +} \frac{d A_i^n(x_i)}{d x_i}$$

(2) In case of the SCOS MF, FBF, $\xi^n(x)$ turns to be the product of MFs as in the triangular MFs, that is,

$$\sum_{n=1}^{M} \Omega^n(\mathrm{x}) = \sum_{n_1=1}^{M_1} \sum_{n_2=1}^{M_2} \cdots \sum_{n_m=1}^{M_m} \prod_{i=1}^{m} A_i^{n_i}(x_i) = 1$$

and

$$\xi^n(\mathrm{x}) = \frac{\Omega^n(\mathrm{x})}{\sum_{n=1}^{M} \Omega^n(\mathrm{x})} = \Omega^n(\mathrm{x}) = \prod_{i=1}^{m} A_i^{n_i}(x_i)$$

Then the output is reduced to (4) as in the case of the triangular MFs.

$$
\begin{aligned}
y_m &= \Im(\mathrm{x}) = \Im(x_1, x_2, \cdots, x_m) = \sum_{n=1}^{M} \left\{ \xi^n(\mathrm{x}) \right\} \theta^n \\
&= \sum_{n_1=1}^{M_1} \sum_{n_2=1}^{M_2} \cdots \sum_{n_m=1}^{M_m} \left\{ \frac{\prod_{i=1}^{m} A_i^{n_i}(x_i)}{\sum_{n_1=1}^{M_1} \sum_{n_2=1}^{M_2} \cdots \sum_{n_m=1}^{M_m} \prod_{i=1}^{m} A_i^{n_i}(x_i)} \right\} \theta^{n_1, n_2, \cdots, n_m} \\
&= \sum_{n=1}^{M} \left\{ \frac{\Omega^n(\mathrm{x})}{\sum_{n=1}^{M} \Omega^n(\mathrm{x})} \right\} \theta^n = \sum_{n=1}^{M} \left\{ \xi^n(\mathrm{x}) \right\} \theta^n
\end{aligned} \tag{4}
$$

Compared to (2-2), the reduced representation of (4) is more suitable for both the fast fuzzy inference and the on-line identification. The proof of this remark is given in appendix.

(3) While the gaussian MF is specified by the center and the width, the SCOS MF is determined by the boundary values (σ_i^n and σ_i^{n+1}). Thus compared to gaussian MF, the SCOS MF is more consistent and can prevent the overlapping of the redundant MFs.

3. On-Line Identification

Among a panoply of various learning schemes, the gradient search and the extended Kalman filter (EKF) approaches are the most popular and widely used algorithms in the field of identification [24, 25]. In this paper, three different strategies are used to identify a fuzzy system adopting SCOS MFs on-line:

(1) the gradient search algorithm (GS)
(2) the extended Kalman filter (EKF)
(3) the hybrid approach of (1) and (2) (HYB).

In the following subsections, these three algorithms are explained briefly one by one and their characteristics including merits and demerits are presented in comparison with those of each other. Before that, some initial construction of a fuzzy system to be identified is discussed and some simple preprocessing is explained.

A. Initial Fuzzy System Construction

The numbers of fuzzy sets for each coordinate (*i.e.*, M_i for the i ’th coordinate) are assumed to be given in advance. Then, M_i SCOS fuzzy MFs $A_i^{n_i}$'s are defined, which uniformly cover U_i, the projection of U onto the i ’th coordinate. In other words, the premise parameters (σ_i^n, $\cdots$, $\sigma_i^{M_i}$) of the fuzzy system adopting SCOS MFs are positioned in this order with the identical intervals between the neighboring points to cover U_i. During the on-line identification, the order should be kept and cannot be changed.

After the aforementioned initial construction is completed, the given initial fuzzy system with initialized premise parameters and arbitrary consequent parameters is represented as in (5):

$$\begin{aligned} &\mathrm{R}^{n_1 n_2 \cdots n_m}\text{: If } x_1 \text{ is } \mathrm{A}_1^{n_1} \text{ and } x_2 \text{ is } \mathrm{A}_2^{n_2}, \cdots, x_m \text{ is } \mathrm{A}_m^{n_m}, \\ &\qquad \text{then } \ y \ \text{ is } \ \theta^{n_1, n_2, \cdots, n_m} \\ &(n_1=1, \cdots \mathrm{M}_1, \ \ n_2=1, \cdots \mathrm{M}_2, \ \ \cdots\cdots, \ \ n_m=1, \cdots \mathrm{M}_m) \end{aligned} \tag{5}$$

or in input-output representation as in (6):

$$\begin{aligned} y_m &= \Im(\mathrm{x}, \theta) = \Im(x_1, x_2, \cdots, x_m, \theta) = \sum_{n=1}^{M} \left\{ \xi^n(\mathrm{x}) \right\} \theta^n \\ &= \sum_{n_1=1}^{M_1} \sum_{n_2=1}^{M_2} \cdots \sum_{n_m=1}^{M_m} \theta^{n_1, n_2, \cdots, n_m} \prod_{i=1}^{m} A_i^{n_i}(x_i, \sigma_i^{n_i-1}, \sigma_i^{n_i}, \sigma_i^{n_i+1}) \end{aligned} \tag{6}$$

where the parameters to be identified on-line are consequent parameters $\theta^{n_1, n_2, \cdots, n_m}$ $(n_i=1, \cdots \mathrm{M}_i, \ i=1, \cdots, m)$ and premise parameters $\sigma_i^{k_i}$ $(i=1, \cdots, m, \ k_i=1, \cdots \mathrm{M}_i)$. For simplicity, the parameters are collected to form a vector as follows:

$$\theta = (\theta_c^T, \ \theta_p^T)^T (\theta^{n_1, n_2, \cdots, n_m}) \qquad (L \times 1 \text{ matrix})$$

where $\theta_c^T = (\theta^{1,1,\cdots,1}, \cdots, \ \theta^{M1, M2, \cdots, Mm})$

- consequent parameters ($L_c \equiv \prod_{i=1}^{m} M_i$ elements)

$$\theta_p^T = (\sigma_1^1, \cdots, \sigma_1^{M1}, \sigma_2^1, \cdots, \sigma_2^{M2}, \ \cdots \ , \sigma_m^1, \cdots, \sigma_m^{Mm})$$

- premise parameters ($L_p \equiv \prod_{i=1}^{m} (M-2)$ elements), (σ_1^1's and σ_m^{Mi}'s are fixed to cover the domain of interest.)

$$L \equiv L_c + L_p = \prod_{i=1}^{m} M_i + \prod_{i=1}^{m} (M-2) \quad \text{(the size of } \theta\text{)}.$$

In the following subsections, how to adjust the premise and consequent parameters is presented.

B. Gradient Search Algorithm

The gradient search algorithm is widely used parameter tuning algorithm [24]. In this paper, gradient search algorithm is used to adapt the parameters on-line in the direction of the negative gradient as shown in (7)

$$\hat{\theta}(t+1) = \hat{\theta}(t) - \eta \frac{\partial E}{\partial \theta(t)} + \alpha(\hat{\theta}(t) - \hat{\theta}(t-1)) \tag{7}$$

where t refers to the number of iterations (time), η is the learning rate and α is the momentum rate. For $\hat{\theta}_p(t)$ and $\hat{\theta}_c(t)$, different learning rates are used and they are denoted by η_p and η_c, respectively. The cost function is given by:

$$E = \frac{1}{2}(y_d(t) - y_m(t))^2$$

where y_d is the desired value and y_m is the output from the fuzzy model. In (7), the gradient of the cost function with respect to each parameter is expressed as

$$\frac{\partial E}{\partial \theta(t)} = -(y_d(t) - y_m(t)) \frac{\partial y_m(t)}{\partial \theta(t)}$$

and $\frac{\partial y_m(t)}{\partial \theta(t)} = \frac{\partial \Im}{\partial \theta(t)}$ can be derived numerically. The gradient search algorithm involves only some element-by-element additions and multiplications and is computationally simple, compared to other parameter tuning algorithms. However, the gradient search algorithm has some problems as follows:

(1) The convergence is inherently slow since the learning rate is fixed and the algorithm is subject to the effects of local minims.

(2) The performance is sensitive to the learning rate and the momentum, which are chosen in a heuristic manner.

(3) Furthermore, since it weighs the current measurement too much rather than past measurements, the algorithm is sensitive to measurement noise.

C. Extended Kalman Filter (EKF)

Another approach to nonlinear estimation is to linearize the nonlinear model around a given operating point and apply optimal linear estimation to the linearized system. The extended Kalman filter is such an approximate algorithm and can be viewed as an extended version of the linear Kalman filter to the nonlinear case. It computes an estimate at each sampling instant by the use of the linear Kalman filter on the linearized system of the nonlinear system [26-28]. In this subsection, the EKF is briefly explained and it is applied to the on-line identification of a fuzzy system represented as in (6).

Assume a nonlinear discrete-time system represented by the following equations:

$$x(t+1) = f(x(t)) + w(t) \tag{8}$$

$$y(t+1) = h(x(t)) + v(t) \tag{9}$$

where $x(t)$ and $y(t)$ are the state and observation vectors, respectively, and $f(\cdot)$ and $h(\cdot)$ are time-invariant nonlinear functions. Also $w(t)$ and $v(t)$ are independent Gaussian white noise vectors with zero mean and known covariance $Q(t)$ and $R(t)$. The EKF updates the current estimated state vector $\hat{x}(t|t-1)$ based on the observations up to time $t-1$ by the following recursive equation [25]:

$$\hat{x}(t+1|t) = f(\hat{x}(t|t)) \tag{10-1}$$

$$\hat{x}(t|t) = \hat{x}(t|t-1) + K(t)[y(t) - h(\hat{x}(t|t-1))] \tag{10-2}$$

$$K(t) = P(t|t-1)\ H(t)^{T}\ [H(t)\ P(t|t-1)\ H(t)^{T} + R(t)]^{-1} \tag{10-3}$$

$$P(t+1|t) = F(t)\ P(t|t)\ F(t)^{T} + Q(t) \tag{10-4}$$

$$P(t|t) = P(t|t-1) - K(t)\ H(t)\ P(t|t-1) \tag{10-5}$$

where the matrix $F(t)$ and the $H(t)$ have the appropriate size and are defined by

$$F(t) = \left(\frac{\partial f}{\partial x}\right), \quad x = \hat{x}(t/t) \tag{11-1}$$

$$H(t) = \left(\frac{\partial h}{\partial x}\right), \quad x = \hat{x}(t/t-1) \tag{11-2}$$

Since the EKF is just the approximate estimate method based on the linearization of $f(x(t))$ and $h(x(t))$ around the estimated points, the EKF may also stay at local minims as in the case of the gradient search. Nevertheless, many successful applications have been reported due to its excellent convergence properties [25, 27]

To apply the EKF to the on-line identification of the fuzzy system of (6), the fuzzy system is recast to the configuration of the state-space and observation system as follows:

$$\theta(t+1) = \theta(t) \tag{12-1}$$

$$\begin{aligned} y_d(t) &= \Im\ (\theta(t))\ +\ v(t) \\ &= y_m(t)\ +\ v(t) \end{aligned} \tag{12-2}$$

where t denotes time. $v(t)$ is assumed to be a white noise with variance λ. The application of the EKF to the identification of the fuzzy system of (12) provides the following on-line identification algorithm:

$$\widehat{\theta}(t) = \widehat{\theta}(t-1) + \mathrm{K}(t)\,[y_d(t) - y_m(t)] \qquad (L\times 1) \tag{13-1}$$

$$\mathrm{K}(t) = \frac{\mathrm{P}(t\text{-}1)\ \mathrm{H}(t)^{\mathrm{T}}}{\mathrm{H}(t)\ \mathrm{P}(t\text{-}1)\ \mathrm{H}(t)^{\mathrm{T}} + \lambda} \qquad (L\times 1) \tag{13-2}$$

$$\mathrm{P}(t) = \mathrm{P}(t\text{-}1) - \mathrm{K}(t)\ \mathrm{H}(t)\ \mathrm{P}(t\text{-}1) \qquad (L\times L) \tag{13-3}$$

where, for simplicity, $\mathrm{P}(t) = \mathrm{P}(t|t)$ and $\hat{\theta}(t) = \widehat{\theta}(t|t)$ since $\mathrm{P}(t|t) = \mathrm{P}(t+1|t)$ and $\widehat{\theta}(t|t) = \hat{\theta}(t+1|t); \hat{y}_m(t)$ is the output from the fuzzy model based on the $t-1$ parameter estimations, *i.e.*, $\hat{y}_m(t) = \Im(x(t), \theta(t-1))$.

From (11-2), $\mathrm{H}(t)$ is given as follows:

$$\mathbf{H}(t) = \left(\frac{\partial \Im}{\partial \theta}\right)_{\theta(t) = \widehat{\theta}(t-1)}$$

$$= \left(\frac{\partial \Im}{\partial \theta_1},\ \frac{\partial \Im}{\partial \theta_2},\ \cdots,\ \frac{\partial \Im}{\partial \theta_L}\right)_{\theta(t) = \widehat{\theta}(t-1)} \ (1\times L)$$

where $\frac{\partial \Im}{\partial \theta_i}$ $(i = (1, \cdots, L))$ can be derived numerically. Further, usually the variance λ is unknown *a priori* and it should be estimated as in [28]. In this paper, however, λ is set to 1, for simplicity, which means all sample data are given unit weights and all of them are considered to be equally important.

As far as the implementation of the EKF is concerned, the following problem should be taken into consideration. When the number of the inputs or the number of the fuzzy rules involved are high, the EKF recursive equations represented by (13) may become computationally intractable because the size of the $\mathrm{P}(t)$ may be too large $(L\times L)$. In this case, θ $(L\times 1$ matrix$)$ cannot be updated all together at a time. To facilitate the implementation, the parameter vector θ is divided into several subvectors and the subvectors are identified in series. For example, in

$\theta = (\theta_c^T,\ \theta_p^T)^T$, the previous estimates of $\hat{\theta}_c$ are used for the update of $\hat{\theta}_p$ in the present iteration and then $\hat{\theta}_c$ are updated assuming that the newly updated $\hat{\theta}_p$ is the true value of θ_p.

D. Hybrid Approach (HYB)

The gradient search consists of only element-wise additions and multiplications and is computationally simple but the performance is very sensitive to the learning rate and the momentum chosen in a heuristic manner. On the contrary, the EKF is more reliable than the gradient search but computationally more expensive. The hybrid approach is the combined version of the two algorithms; it is computationally simpler than the EKF and outperforms the gradient search.

As noted in (6), the output of the fuzzy model is *linear in the consequent parameters and nonlinear in the premise parameters*. In the hybrid approach, for the consequent parameters, the EKF is used for the update because the Kalman filter guarantees the faster convergence than the gradient search for the linear parameters. For the premise parameters, on the other hand, the gradient search is adopted to lessen the computational burden because even the EKF gives only approximate estimates of the nonlinear parameters. In summary, the hybrid approach is given as follows:

For $\theta = (\theta_c^T,\ \theta_p^T)^T$

$$\theta_p(t+1) = \theta_p(t) - \eta_p \frac{\partial E}{\partial \theta_p(t)} + \alpha(\theta_p(t) - \theta_p(t-1)) \quad (L_p \times 1) \qquad (14\text{-}1)$$

$$\theta_c(t) = \theta_c(t-1) + \mathrm{K_c}(t)\ [y_d(t) - y_m(t)] \quad (L_c \times 1) \qquad (14\text{-}2)$$

$$\mathrm{K_c}(t) = \frac{\mathrm{P_c}(t-1)\ \mathrm{H_c}(t)^T}{\mathrm{H_c}(t)\ \mathrm{P_c}(t-1)\ \mathrm{H_c}(t)^T + 1} \quad (L_c \times 1) \qquad (14\text{-}3)$$

$$\mathrm{P_c}(t) = \mathrm{P_c}(t-1) - \mathrm{K_c}(t)\ \mathrm{H_c}(t)\ \mathrm{P_c}(t-1) \quad (L_c \times L_c) \qquad (14\text{-}4)$$

4. Computer Simulation

In what follows, several illustrative examples are provided to illustrate the validity of the suggested methods. Some examples are taken from the previous works [8, 13, 16, 29] to compare the performance of the proposed on-line fuzzy system with those of conventional fuzzy systems. For comparison, the performance measures used in the original papers are adopted such that the performance measures differ from example to example. In every example, however, a lower number means better performance.

A. Example 1

The target system to be modeled is a nonlinear static function expressed by (15) taken from [8] and [13].

$$y = (1 + x_1^{-2} + x_2^{-1.5})^2 , \quad 1 \le x_1,\ x_2 \le 5 \tag{15}$$

One hundred points are taken randomly from $1 \le x_1,\ x_2 \le 5$ to form input-output data and the fuzzy models are built on-line by the suggested three algorithms one by one. In this example and the following examples, each universe of discourse is assumed to be covered by five fuzzy MFs $(M_1 = M_2 = \cdots = M_m = 5)$.

In the GS-based method, η_p, η_c and α are set to 0.2, 0.6, and 0.2, respectively, because they provide the fastest convergence in several trials. In the EKF-based identification, $\mathrm{P}(0)$ is initialized by $10\,\mathrm{I}$. In the hybrid approaches, η_p, α, and $\mathrm{P}_c(0)$ are set to 0.5, 0.2 and $10\,\mathrm{I}$. If not specified in the following examples, $\mathrm{P}(0)$ is initialized by $10\,\mathrm{I}$. In this example, the MSE (Mean Squared Error) is used as a performance measure as follows:

$$\mathrm{PM} = \mathrm{MSE} = \overline{e^2} = \frac{1}{n}\sum_{i=1}^{n}(y_d(i) - y_m(i))^2$$

The performance measures of the suggested three on-line algorithms are given in Table 1 with those of other conventional methods. While fifty sample data are trained off-line for thousands of epochs in [8] and [13], one hundred data are used on-line (one epoch) in the suggested three algorithms.

Table 1

COMPARISON OF PERFORMANCE (EXAMPLE 1)

Model	PM
Kim et al [8]	0.0197
Sugeno and Yasukawa [13]	0.0790
Gradient search	0.0247
Extended Kalman filter	0.0193
Hybrid	0.0255

It is quite noticeable that one of the suggested on-line implemented fuzzy model (the EKF-based model) even outperforms the conventional off-line models. Fig. 2 shows the plot of the performance measure vs. time. In this figure, although the identification is carried out on-line assuming that only the present sample is available, the MSE calculated using a batch of all sample data is drawn for reference.

Fig. 3 compares the original system to the fuzzy systems reconstructed by the suggested algorithms.

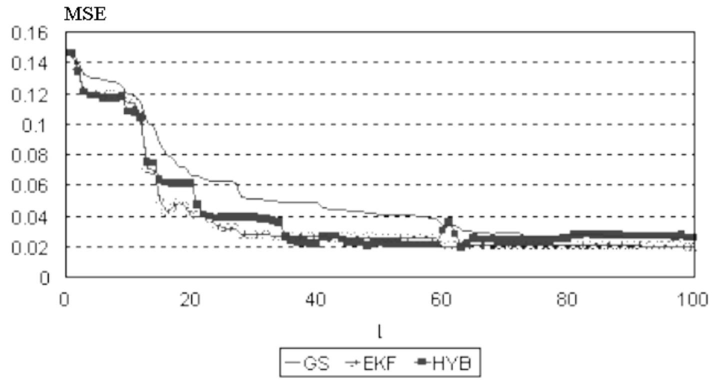

Fig. 2 – MSE vs. time (Example 1)

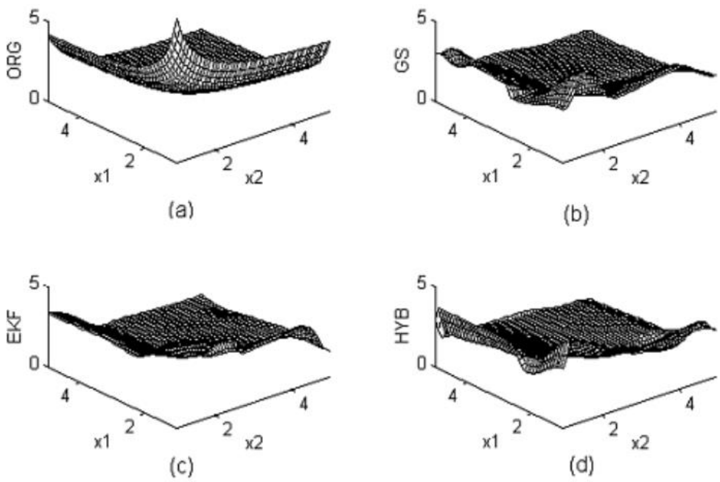

Fig. 3 – Original system and on-line reconstructed fuzzy systems

(Example 1) (a) Original (b) GS (c) EKF (d) HYB

B. Example 2

In this example, a nonlinear system with two inputs (x_1 and x_2) and a single output y defined by

$$y = (2 + x_1^{1.5} - 1.5\sin(3x_2)^2 \ , \quad 1 \le x_1, \ x_2 \le 3 \tag{16}$$

is used. This example is taken from [16]. As in Example 1, one hundred points are taken randomly from $1 \le x_1, \ x_2 \le 3$ to form input-output data and the fuzzy models are formed on-line by the suggested three algorithms. In the GS-based method, η_p, η_c and α are set to 0.005, 0.085 and 0.005, respectively, which showed the fastest convergence in several trials. In the hybrid approaches, both of η_p and α are set to 0.001. In this example, the performance measure, which was adopted in [16] is used for comparison. The performance measure is defined as follows and the comparison of performance is given in Table 2.

$$\mathrm{PM} = \frac{\sqrt{\sum_{i=1}^{n} (y_d(i) - y_m(i))^2}}{\sum_{i=1}^{n} \left| y_d(i) \right|}$$

Table 2

COMPARISON OF PERFORMANCE (EXAMPLE 2)

Model	PM
Lin and Cunningham [16]	0.00497
Gradient search	0.08400
Extended Kalman filter	0.02090
Hybrid	0.02500

In this example, none of the suggested algorithms outperforms the conventional method. However, when taking into account that the parameters in [16] were updated for 2800 epochs and the parameters of the suggested methods are identified on-line (only for one epoch), we can say that the results are acceptable and the suggested algorithms are appropriate for the applications requiring the real-time processing. Fig. 4 is the plot of the performance measure vs. time and Fig. 5 is the figure, which compares the original system and the fuzzy systems reconstructed by the suggested algorithms.

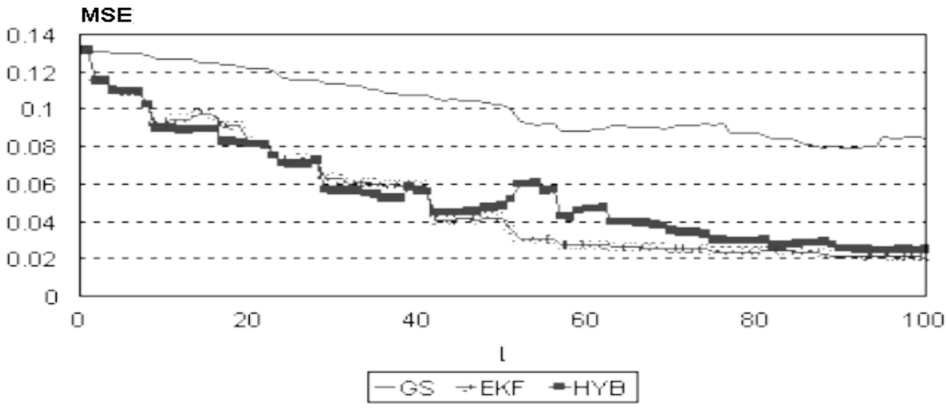

Fig. 4 – PM vs. time (Example 2)

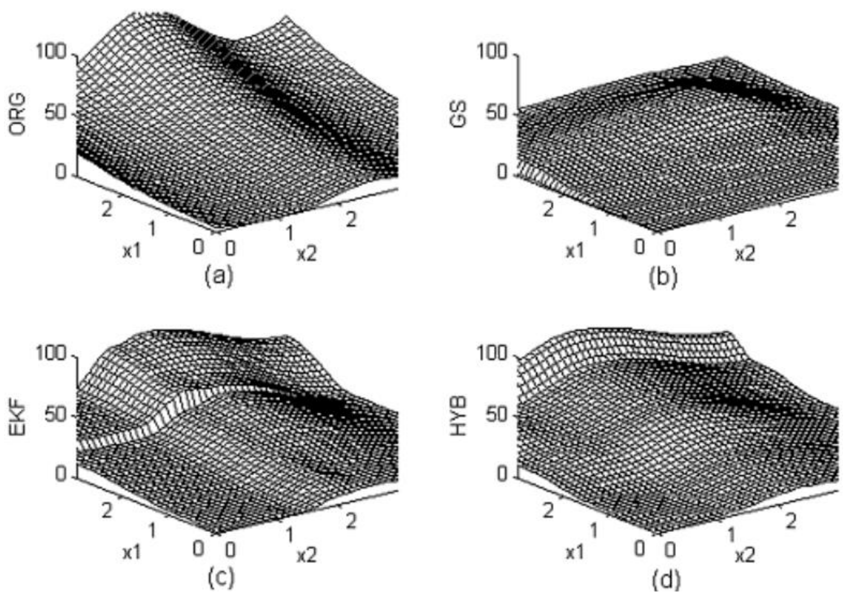

Fig. 5 – Original system and on-line reconstructed fuzzy systems

(Example 2) (a) Original (b) GS (c) EKF (d) HYB

C. Example 3

The system to be studied in this example originally appears in [29] and is described by the (17).

$$y = 10\left\{\exp\left(-\frac{(x_1-0.5)^2}{0.75}-\frac{(x_2-2.5)^2}{3.75}\right)+\exp\left(-\frac{(x_1-2.5)^2}{0.05}-\frac{(x_2-3.5)^2}{0.05}\right)\right.$$

$$+\exp\left(-\frac{(x_1-3.0)^2}{0.05}-\frac{(x_2-3.5)^2}{0.05}\right)+\exp\left(-\frac{(x_1-4.5)^2}{1.50}-\frac{(x_2-1.0)^2}{1.00}\right) \qquad (17)$$

$$\left.+\exp\left(-\frac{(x_1-4.5)^2}{0.50}-\frac{(x_2-4.5)^2}{0.50}\right)\right\} \quad , \quad 0 \le x_1,\ x_2 \le 5$$

One hundred samples are taken from the function to form the fuzzy models. In the GS-based method, η_p, η_c and α are chosen to be 0.01, 0.2, and 0.05, respectively, by several trials. In the hybrid approach, the same η_p and α are used. In this example, only the performance measures of the suggested algorithms are compared with each other. This is because the modeling suggested in [29] is a model-based strategy, as opposed to the sample-databased strategies of this paper, and therefore the performance comparison with [29] is meaningless. In this example, the MSE (Mean Squared Error) is used as a performance measure:

$$\mathrm{PM} = \mathrm{MSE} = \overline{e^2} = \frac{1}{n}\sum_{i=1}^{n}(y_d(i) - y_m(i))^2$$

The performance measures are compared with each other in Table 3. It can be noted that the performance measure of the hybrid approach is almost as good as that of the EKF-based method in this example.

Table 3

COMPARISON OF PERFORMANCE

(EXAMPLE 3)

Model	PM
Gradient search	2.8857
Extended Kalman filter	1.0388
Hybrid	1.1821

The MSE vs. time is monitored in Fig. 6. The original system and the fuzzy systems reconstructed by the suggested algorithms are compared in Fig. 7.

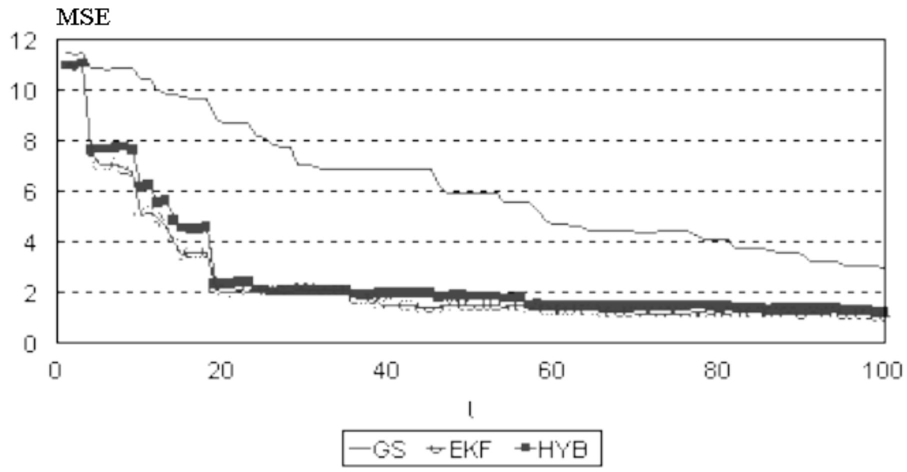

Fig. 6 – MSE vs. time (Example 3)

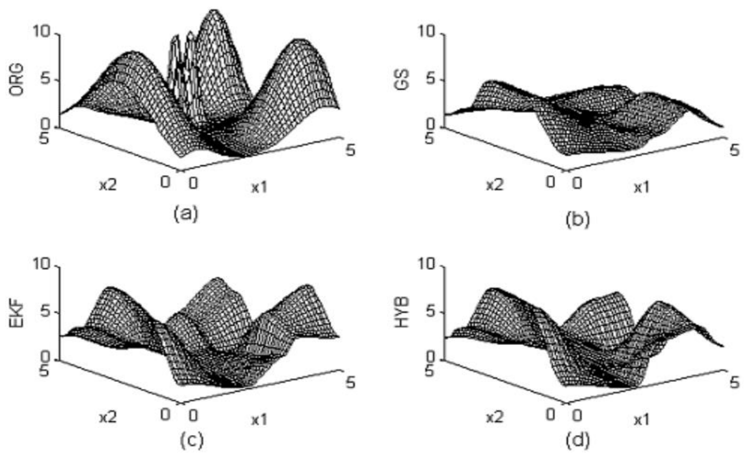

Fig. 7 – Original system and on-line reconstructed fuzzy systems

(Example 3) (a) Original (b) GS (c) EKF (d) HYB

5. CONCLUSION

In this paper, on-line identification methodologies for a fuzzy system are proposed and their validity is verified through computer simulations. First squared-cosine MFs is introduced to reduce the number of parameters and to make on-line identification tractable. Then the on-line identification of the fuzzy system adopting SCOS MFs is carried out by the gradient search method (GS), the extended Kalman filter (EKF) and the hybrid (HYB) approach of the GS and the

EKF. The applications of the discussed on-line identification methodologies are numerous. The direct use of identification arises in nonlinear adaptive control and signal processing with little prior knowledge available. However, as the number of input increases, the suggested algorithms may lead to the combinatorial explosion of the fuzzy rules and the further studies regarding the problem are needed.

APPENDIX

For $x'(x_1, x_2, \ldots, x_m)$, assume $\sigma_1^{k_1} \le x_1 < \sigma_1^{k_1 * 1}$, $\sigma_2^{k_2} \le x_2 < \sigma_2^{k_2 * 1}$, $\sigma_m^{k_m} \le x_m < \sigma_m^{k_m * 1}$. For i th component x_i, only two memberships are fired at an instant as follows:

$$A_i^{k_i}(x_i) = \cos^2 \frac{\pi\left(x_i - \sigma_i^{k_i}\right)}{2\left(\sigma_i^{k_i} - \sigma_i^{k_i}\right)}$$

$$A_i^{k_i+1}(x_i) = \cos^2 \frac{\pi\left(x_i - \sigma_i^{k_i+1}\right)}{2\left(\sigma_i^{k_i+1} - \sigma_i^{k_i}\right)}$$

$$A_i^{P}(x_i) = 0 \quad (p = 1, \ldots, M_i,\ p \ne k_i, k_i + 1)$$

Now,

$$\begin{aligned}
\sum_{n_i=1}^{M_i} A_i^{n_i}(x_i) &= A_i^{k_i}(x_i) + A_i^{k_i+1}(x_i) = \cos^2 \frac{\pi(x_i - \sigma_i^{k_i})}{2(\sigma_i^{k_i+1} - \sigma_i^{k_i})} + \cos^2 \frac{\pi(x_i - \sigma_i^{k_i+1})}{2(\sigma_i^{k_i+1} - \sigma_i^{k_i})} \\
&= \cos^2 \frac{\pi(x_i - \sigma_i^{k_i})}{2(\sigma_i^{k_i+1} - \sigma_i^{k_i})} + \cos^2 \frac{\pi(x_i - \sigma_i^{k_i+1})}{2(\sigma_i^{k_i+1} - \sigma_i^{k_i})} \\
&= \cos^2 \frac{\pi(x_i - \sigma_i^{k_i})}{2(\sigma_i^{k_i+1} - \sigma_i^{k_i})} + \left\{ -\sin \frac{\pi(x_i - \sigma_i^{k_i})}{2(\sigma_i^{k_i+1} - \sigma_i^{k_i})} \right\}^2 = 1
\end{aligned}$$

Therefore:

$$\begin{aligned}
\sum_{n_1=1}^{M_1} \sum_{n_2=1}^{M_2} \cdots \sum_{n_m=1}^{M_m} \prod_{i=1}^{m} A_i^{n_i}(x_i) &= \sum_{n_1=1}^{M_1} A_1^{n_1}(x_1) \left(\sum_{n_2=1}^{M_2} \cdots \sum_{n_m=-M_m}^{M_m} \prod_{i=2}^{m} A_i^{n_i}(x_i) \right) \\
&= \sum_{n_1=1}^{M_1} A_1^{n_1}(x_1) \sum_{n_2=1}^{M_2} A_2^{n_2}(x_2) \left(\sum_{n_3=1}^{M_3} \cdots \sum_{n_m=1}^{M_m} \prod_{i=2}^{m} A_i^{n_i}(x_i) \right) = \qquad \cdots\cdots \\
&= \prod_{i=1}^{m} \left\{ \sum_{n_1=1}^{M_i} A_i^{n_i}(x_i) \right\} = 1
\end{aligned}$$

References

[1] L. A. ZADEH: The roles of soft computing and fuzzy logic in the conception, design, and deployment of intelligent systems, opening lecture, Proc. of the 4th Int. Conf. on Soft Computing, Iizuka, pp. 3-4, 1996.

[2] W. PEDRYCZ: *An identification algorithm in fuzzy relational systems*, Fuzzy Sets and Systems, vol. **13**, pp.153-167, 1984.

[3] W. PEDRYCZ: *Fuzzy Control and Fuzzy Systems*, John Willey and Sons, New York, 1993.

[4] T. TAKAGI, M. SUGENO: *Fuzzy identification of systems and its applications to modeling and control*, IEEE Trans. Systems Man Cybernet, vol. **15**, No. 1, pp. 116-132, 1985.

[5] M. SUGENO, G. T. KANG: *Structure identification of fuzzy model*, Fuzzy Sets and Systems, vol. **28**, pp. 15-33, 1988.

[6] M. SUGENO, G. T. KANG: *Fuzzy modeling and control of multilayer incinerator*, Fuzzy Sets and Systems, Vol. **18**, pp. 329-346, 1986.

[7] S. K. SIN, R. J. P. deFIGUEIREDO: *Fuzzy system design through fuzzy clustering and optimal predefuzzification*, 2nd IEEE Int. Conf on Fuzzy Systems, San Francisco, Calif., pp.190-195, 1993.

[8] E. KIM, M. PARK, S. JI: *A new approach to fuzzy modeling*, IEEE Trans. Fuzzy Systems, vol. **5**, No. 3, pp. 328-337, Aug. 1997.

[9] E. KIM, H. LEE, M. PARK: *A simply-identified Sugeno-type fuzzy model via double clustering*, Information Sciences, vol. **110**, pp., 25-39, 1998.

[10] E. KIM, M. PARK, S. KIM, M. PARK: *A transformed input-domain approach to fuzzy modeling*, IEEE Trans. Fuzzy Systems, vol. **6**, no. 4, pp. 596-604, 1998.

[11] M. PARK, S. JI, E. KIM, M. PARK: *Identification of Takagi-Sugeno fuzzy models via clustering and Hugh transform*, in H. Hellendoorn and D. Driankov, eds., Fuzzy Model Identification: Selected Approaches, Springer- Verlag:Berlin, pp. 91-119, 1997.

[12] L. WANG, R. LANGARI: Building Sugeno-type models using fuzzy discretization and orthogonal parameter estimation techniques, IEEE Trans. Fuzzy Systems, vol. **3**, No. 4, pp. 454-458, Nov. 1995.

[13] M. SUGENO, T. YASUKAWA: *A fuzzy-logic-based approach to qualitative modeling*, IEEE Trans. Fuzzy Systems, Vol. **1**, No. 1, pp. 7-31, Feb. 1993.

[14] S. HORIKAWA, T. FURUHASHI, Y. UCHIKAWA: *On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm*, IEEE Trans. Neural Networks, vol. **3**, No. 5, pp. 801-806, 1992.

[15] C. T. LIN, C. S. G. LEE: *Neural-network-based fuzzy logic control and decision system*, IEEE Trans. Computer, vol. **40**, No. 12, pp. 1320-1336, Dec. 1990.

[16] Y. LIN, G. A. CUNNINGHAM III: *A new approach to fuzzy-neural modeling*, IEEE Trans. Fuzzy Systems, Vol. **3**, No. 2, pp. 190-197, May 1995.

[17] W. A. FARAG, V. H. QUINTANA, G. LAMBERT-TORRES: *A genetic-based neuro-fuzzy approach for modeling and control of dynamical systems*, IEEE Trans. Neural Networks, vol. **9**, no. 5, pp. 756-765, 1998.

[18] M. SUGENO, K. TANAKA: Successive identification of a fuzzy model and its applications to prediction of a complex system, Fuzzy Sets and Systems, vol. **42**, pp. 315-334, 1991.

[19] E. H. MAMDANI: Applications of fuzzy algorithms for control of simple dynamic plant, Proc. IEE, vol.**121**, No.2, pp. 1585-1588, 1974.

[20] C. C. LEE: *Fuzzy logic in control systems: Fuzzy logic control - Part I and II*, IEEE Trans. Sys., Man and Cyber., vol. **20**, pp. 404-435, 1990.

[21] X. ZENG, M. G. SINGH: *Approximation theory of fuzzy systems-SISO case*, IEEE Trans. Fuzzy Systems, vol. **2**, No. 2, pp. 162-176, May 1994.

[22] X. ZENG, M. G. SINGH: *Approximation theory of fuzzy systems-MIMO case*, IEEE Trans. Fuzzy Systems, vol. **3**, No. 2, pp. 219-235, May 1995.

[23] L. X. WANG: Adaptive Fuzzy Systems And Control: Design and Stability Analysis, Englewood Cliffs, NJ: Prentice-Hall, 1994.

[24] B. WIDROW, S. D. STEARNS: *Adaptive Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1985.

[25] A. H. JAZWINSKI: *Stochastic Processes and Filtering Theory*, New York: Academic, 1970.
[26] G. CHEN: *Approximate Kalman Filtering*, World Scientific Publishing Co., 1993.
[27] B. D. O. ANDERSON, J. B. MOORE: *Optimal Filtering*, Englewood Cliffs, NJ: Prentice-Hall, 1979.
[28] L. LJUNG, T. SÖDERSTRÖM: *Theory and Practice of Recursive Identification*, Cambridge, MA: MIT, 1983.
[29] W. PEDRYCZ, M. REFORMAT: *Rule-based modeling of nonlinear relationships*, IEEE Trans. Fuzzy Systems, vol. **5**, No. 2, pp. 256-269, May 1997.