Fuzzy Identification Using Fuzzy Neural Networks With Stable Learning Algorithms

Wen Yu and Xiaoou Li

***Abstract*—In general, fuzzy neural networks cannot match nonlinear systems exactly. Unmodeled dynamic leads parameters drift and even instability problem. According to system identification theory, robust modification terms must be included in order to guarantee Lyapunov stability. This paper suggests new learning laws for Mamdani and Takagi–Sugeno–Kang type fuzzy neural networks based on input-to-state stability approach. The new learning schemes employ a time-varying learning rate that is determined from input–output data and model structure. Stable learning algorithms for the premise and the consequence parts of fuzzy rules are proposed. The calculation of the learning rate does not need any prior information such as estimation of the modeling error bounds. This offer an advantage compared to other techniques using robust modification.**

***Index Terms*—Fuzzy neural networks, identification, stability.**

I. Introduction

BOTH NEURAL networks and fuzzy logic are universal estimators, they can approximate any nonlinear function to any prescribed accuracy, provided that sufficient hidden neurons and fuzzy rules are available. Resent results show that the fusion procedure of these two different technologies seems to be very effective for nonlinear systems identification [1], [4], [13]. Gradient descent and backpropagation are always used to adjust the parameters of membership functions (fuzzy sets) and the weights of defuzzification (neural networks) for fuzzy neural networks. Slow convergence and local minimum are main drawbacks of these algorithms [14]. Some modifications were derived in recently published literatures. [3] suggested a robust backpropagation law to resist the noise effect and reject errors drift during the approximation. [24] used B-spline membership functions to minimize a robust object function, their algorithm can improve convergence speed. In [21], radial basis function (RBF) neural networks were applied to fuzzy systems, a novel approach of determining structure and parameters of fuzzy neural systems was proposed.

The stability problem of fuzzy neural identification is very important in applications. It is well known that normal identification algorithms (for example, gradient descent and least square) are stable in ideal conditions. In the presence of unmodeled dynamics, they might become unstable. The lack of robustness of the parameter identification was demonstrated in [5] and became a hot issue in the 1980s, when some robust modification techniques were suggested [8]. The learning procedure of fuzzy neural networks can be regarded as a type of parameter identification. Gradient descent and backpropagation algorithms are stable, if fuzzy neural models can match nonlinear plants exactly. However, some robust modifications must be applied to assure stability with respect to uncertainties. Projection operator is an effective tool to guarantee fuzzy modeling bounded [23]. It was also used by many fuzzy-neural systems [11]. Another general approach is to use robust adaptive techniques [8] in fuzzy neural modeling. For example, [25] applied a switch σ-modification to prevent parameters drift.

Fuzzy neural identification uses input–output data and model structure. It can be regarded as black-box approximation. All uncertainties can be considered as parts of the black-box, i.e., unmodeled dynamics are within the black-box model, not as structured uncertainties. Therefore, the robustifying techniques usually employed are not necessary. In [22], the authors suggested stable and optimal learning rate without robust modification, a genetic search algorithm was proposed to find the optimal rate. However the algorithm is complex, and difficult to realize. By using the passivity theory, we successfully proved that for continuous-time recurrent neural networks, gradient descent algorithms without robust modification were stable and robust to any bounded uncertainties [26], and for continuous-time identification they were also robustly stable [27]. Nevertheless, do fuzzy neural networks have the similar characteristics? To the best of our knowledge, input-to-state stability (ISS) approach for fuzzy neural system was not still applied in the literature.

In this paper, the ISS approach is applied to system identification via fuzzy neural networks. Two cases are considered: 1) the premise memberships are assumed to be known, predetermined somehow in advance and learning is carried on only on the consequence parameters; and 2) weight update concerns both the premise and the consequent parameters. The new stable algorithms with time-varying learning rates are applied to two types of fuzzy neural models, namely, the traditional Mamdani's type model and Takagi–Sugeno–Kang's (TSK) model. Two examples are given to illustrate the effectiveness of the suggested algorithms.

II. Preliminaries

ISS is another elegant approach to analyze stability besides Lyapunov method. It can lead to general conclusions on stability by using input and state characteristics. Consider the following discrete-time nonlinear system:

$$x(k+1) = f[x(k), u(k)] \quad y(k) = h[x(k)] \tag{1}$$

Manuscript received October 2, 2002; revised February 19, 2003 and August 4, 2003. This work was supported by CONACyT under Grants 38505A and 39151A.

W. Yu is with the Departamento de Control Automático, CINVESTAV-IPN, México City 07360, México (e-mail: yuw@ctrl.cinvestav.mx).

X. Li is with the Sección de Computación, Departamento de Ingeniería Eléctrica, CINVESTAV-IPN, México City 07360, México.

Digital Object Identifier 10.1109/TFUZZ.2004.825067

where $u(k) \in \Re^m$ is the input vector, $x(k) \in \Re^n$ is a state vector, and $y(k) \in \Re^m$ is the output vector. f and h are general nonlinear smooth function $f, h \in C^\infty$. Let us now recall the following definitions.

Definition 1:

a) If a function $\gamma(\cdot)$ is continuous and strictly increasing with $\gamma(0) = 0$, we call $\gamma(\cdot)$ as $\mathcal{K}$-function

b) If a function $\beta(\cdot)$ is $\mathcal{K}$-function and $\lim_{s_k \to \infty} \beta(s_k) = 0$, we call $\beta(\cdot)$ as $\mathcal{KL}$-function.

c) If a function $\alpha(\cdot)$ is $\mathcal{K}$-function and $\lim_{s_k \to \infty} \alpha_i(s_k) = \infty$, we call $\alpha(\cdot)$ as $\mathcal{K}_\infty$-functions.

Definition 2:

a) System (1) is said to be *input-to-state stable* if there exists a $\mathcal{K}$-function $\gamma(\cdot)$ and $\mathcal{KL}$ -function $\beta(\cdot)$, such that, for each $u \in L_\infty$, *ie.,* $\sup\{\|u(k)\|\} < \infty$, and each initial state $x^0 \in R^n$, it holds that

$$\left\|x(k, x^0, u(k))\right\| \le \beta(\left\|x^0\right\|, k) + \gamma(\|u(k)\|).$$

b) A smooth function $V: \Re^n \to \Re \ge 0$ is called an ISS-Lyapunov function for (1) if there exist $\mathcal{K}_\infty$-functions $\alpha_1(\cdot)$, $\alpha_2(\cdot)$, $\alpha_3(\cdot)$, and $\mathcal{K}$-function $\alpha_4(\cdot)$ such that for any $s \in \Re^n$, each $x(k) \in \Re^n$, $u(k) \in \Re^m$

$$\begin{aligned}\alpha_1(s) &\le V(s) \le \alpha_2(s)\\ V_{k+1} - V_k &\le -\alpha_3(\|x(k)\|) + \alpha_4(\|u(k)\|).\end{aligned}$$

Theorem 1: If a discrete-time nonlinear system admit a ISS-Lyapunov function, it is input-to-state stable, and the behavior of the system remains bounded when its inputs are bounded [9].

From (1) we have

$$\begin{aligned}y(k) &= h[x(k)] := F_1[x(k)]\\ y(k+1) &= h[f[x(k), u(k)]] := F_2[x(k), u(k)]\\ y(k+n-1) &:= F_n[x(k), u(k)\\ &\qquad u(k+1)\dots u(k+n-2)].\end{aligned} \tag{2}$$

Denoting

$$\begin{aligned}Y(k) &= [y(k), y(k+1), \dots y(k+n-1)]^T\\ U(k) &= [u(k), u(k+1), \dots u(k+n-2)]^T\end{aligned}$$

so $Y(k) = F[x(k), U(k)]$, $F = [F_1 \dots F_n]^T$. Since (1) is smooth nonlinear system, (2) can be expressed as $x(k+1) = g[Y(k+1), U(k+1)]$. This leads to the multi-variable NARMA model [2]

$$\begin{aligned}y(k) &= h[x(k)]\\ &= \Psi[y(k-1), y(k-2), \dots u(k-1), u(k-2), \dots]\\ &= \Psi[X(k)]\end{aligned} \tag{3}$$

where

$$X(k) = [y(k-1), y(k-2) \dots u(k-d), u(k-d-1), \dots]^T \tag{4}$$

$\Psi(\cdot)$ is an unknown nonlinear difference equation representing the plant dynamics, $u(k)$ and $y(k)$ are measurable scalar input and output, d is time delay. One can see that Definitions 1 and 2 and Theorem 1 do not depend on the exact expression of nonlinear systems. In this paper, we will apply ISS to the NARMA model (3).

A generic fuzzy model is presented as a collection of fuzzy rules in the following form (Mamdani fuzzy model [16]):

$$\text{R}^i: \text{IF } x_1 \text{ is } A_{1i} \text{ and } x_2 \text{ is } A_{2i} \text{ and} \cdots x_n \text{ is } A_{ni} \quad \text{THEN } \widehat{y}_1 \text{ is } B_{1i} \text{ and} \cdots \widehat{y}_m \text{ is } B_{mi}. \tag{5}$$

We use $l(i = 1, 2, \dots, l)$ fuzzy IF–THEN rules to perform a mapping from an input linguistic vector $X = [x_1 \dots x_n] \in \Re^n$ to an output linguistic vector $\widehat{Y}(k) = [\widehat{y}_1 \dots \widehat{y}_m]^T \in R^{m\times 1}$. $A_{1i}, \dots A_{ni}$ and $B_{1i}, \dots B_{mi}$ are standard fuzzy sets. Each input variable x_i has l_i fuzzy sets. In the case of full connection, $l = l_1 \times l_2 \times \cdots l_n$. From [23] we know, by using product inference, center-average and singleton fuzzifier, the pth output of the fuzzy logic system can be expressed as

$$\widehat{y}_p = \frac{\left(\sum_{i=1}^{l} w_{pi}\left[\prod_{j=1}^{n} \mu_{A_{ji}}\right]\right)}{\left(\sum_{i=1}^{l}\left[\prod_{j=1}^{n} \mu_{A_{ji}}\right]\right)} = \sum_{i=1}^{l} w_{pi}\phi_i \tag{6}$$

where $\mu_{A_{ji}}$ is the membership functions of the fuzzy sets A_{ji}, w_{pi} is the point at which $\mu_{B_{pi}} = 1$. If we define

$$\phi_i = \frac{\prod_{j=1}^{n} \mu_{A_{ji}}}{\sum_{i=1}^{l}\prod_{j=1}^{n} \mu_{A_{ji}}} \tag{7}$$

(6) can be expressed in matrix form

$$\widehat{Y}(k) = W(k)\Phi[X(k)] \tag{8}$$

where parameter $W(k) = \begin{bmatrix} w_{11} & & w_{1l}\\ & \ddots & \\ w_{m1} & & w_{ml}\end{bmatrix} \in R^{m\times l}$, data vector $\Phi[X(k)] = [\phi_1 \cdots \phi_l]^T \in R^{l\times 1}$. The structure of the fuzzy neural system is shown in Fig. 1. This four layers fuzzy neural networks was discussed [7], [11], [13], and [22]. Each node of layer II represent the value of the membership function of the linguistic variable. Nodes at layer III represent fuzzy rules. Layer IV is the output layer, the links between layer III and layer IV are full connected by the weight matrix $W(k)$.

For Takagi–Sugeno–Kang fuzzy model [19]

$$\text{R}^i: \text{IF } x_1 \text{ is } A_{1i} \text{ and } x_2 \text{ is } A_{2i} \text{ and} \cdots x_n \text{ is } A_{ni} \quad \text{THEN } \widehat{y}_j = p_{j0}^i + p_{j1}^i x_1 + \cdots p_{jn}^i x_n \tag{9}$$

where $j = 1 \dots m$. The qth output of the fuzzy logic system can be expressed as

$$\widehat{y}_q = \sum_{i=1}^{l}\left(p_{q0}^i + p_{q1}^i x_1 + \cdots p_{qn}^i x_n\right)\phi_i \tag{10}$$

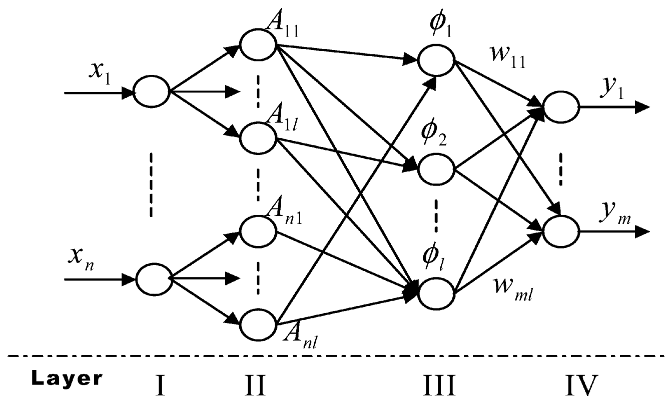


Fig. 1. Architecture of fuzzy neural system.

where ϕ_i is defined as in (7). Equation (10) can be also expressed in the form of the Mamdani-type (8)

$$\widehat{Y}(k) = W(k)\Phi[X(k)] \tag{11}$$

where $\widehat{Y}(k) = [\widehat{y}_1 \cdots \widehat{y}_m]^T$

$$W(k) = \begin{bmatrix} p_{10}^1 \cdots p_{10}^l & p_{11}^1 \cdots p_{11}^l & \cdots & p_{1n}^1 \cdots p_{1n}^l \\ \vdots & \vdots & & \vdots \\ p_{m0}^1 \cdots p_{m0}^l & p_{m1}^1 \cdots p_{m1}^l & \cdots & p_{mn}^1 \cdots p_{mn}^l \end{bmatrix}$$
$$\Phi[X(k)] = [\,\phi_1 \ldots \phi_l \quad x_1\phi_1 \ldots x_1\phi_l \quad \ldots \quad x_n\phi_1 \ldots x_n\phi_l\,]^T.$$

III. Fuzzy Neural Modeling With Known Premise Membership Functions

When we have some prior information of the identified plant, we can construct fuzzy rules as (5) or (9). In this section, we assume the premise membership functions $A_{1i} \ldots A_{ni}$ are given by prior knowledge, i.e., $\phi_i = \prod_{j=1}^n \mu_{A_{ji}} / \sum_{i=1}^l \prod_{j=1}^n \mu_{A_{ji}}$ is known (see [4], [21], and [25]). Mamdani (8) and the TSK (11) models have the same forms because $\Phi\,[X\,(k)]$ is known, the only different is the definition of $W(k)$.

The object of fuzzy neural modeling is to find the center values of $B_{1i} \ldots B_{mi}$ (the weights between Layer III and Layer IV in Fig. 1), such that the output $\widehat{Y}(k)$ of fuzzy neural networks (8) can follow the output $Y(k)$ of nonlinear plant (3). Let us define identification error vector $e(k) \in R^{m\times 1}$ as

$$e(k) = \widehat{Y}(k) - Y(k). \tag{12}$$

We will use the modeling error $e(k)$ to train the fuzzy neural networks (8) online such that $\widehat{Y}(k)$ can approximate $Y(k)$. According to function approximation theories of fuzzy logic [22] and neural networks [6], the identified nonlinear process (3) can be represented as

$$Y(k) = W^*\Phi[X(k)] - \mu(k) \tag{13}$$

where W^* is unknown weights which can minimize the unmodeled dynamic $\mu(k)$. The identification error can be represented by (12) and (13)

$$e(k) = \widetilde{W}(k)\Phi[X(k)] + \mu(k) \tag{14}$$

where $\widetilde{W}(k) = W(k) - W^*$. In this paper we are only interested in open-loop identification, we assume that the plant (3) is bounded-input–bounded-output (BIBO) stable, i.e., $y(k)$ and $u(k)$ in (3) are bounded. By the bound of the membership function Φ, $\mu(k)$ in (13) is bounded. The following theorem gives a stable gradient descent algorithm for fuzzy neural modeling.

Theorem 2: If we use the fuzzy neural networks (8) to identify nonlinear plant (3), the following gradient descent algorithm with a time-varying learning rate can make identification error $e(k)$ bounded

$$W(k+1) = W(k) - \eta_k e(k)\Phi^T[X(k)] \tag{15}$$

where the scalar $\eta_k = (\eta/1 + \|\Phi\,[X\,(k)]\|^2)$, $0 < \eta \le 1$. The normalized identification error

$$e_N(k) = \frac{e(k)}{1 + \max_k(\|\Phi[X(k)]\|^2)}$$

satisfies the following average performance:

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{k=1}^{T} \|e_N(k)\|^2 \le \overline{\mu} \tag{16}$$

where $\overline{\mu} = \max_k \left[\|\mu(k)\|^2\right]$.

Proof: We selected a positive defined scalar L_k as

$$L_k = \left\|\widetilde{W}(k)\right\|^2. \tag{17}$$

By the updating law (15), we have

$$\widetilde{W}(k+1) = \widetilde{W}(k) - \eta_k e(k)\Phi^T[X(k)].$$

Using the inequalities

$$\|a-b\| \ge \|a\| - \|b\| \qquad 2\,\|ab\| \le a^2 + b^2$$

for any a and b. By using (14) and $0 \le \eta_k \le \eta \le 1$, we have

$$\begin{aligned} \Delta L_k &= L_{k+1} - L_k \\ &= \left\|\widetilde{W}(k) - \eta_k e(k)\Phi^T(X)\right\|^2 - \left\|\widetilde{W}(k)\right\|^2 \\ &= \left\|\widetilde{W}(k)\right\|^2 - 2\eta_k \left\|e(k)\Phi^T(X)\widetilde{W}(k)\right\| \\ &\quad + \eta_k^2 \|e(k)\Phi[X(k)]\|^2 - \left\|\widetilde{W}(k)\right\|^2 \\ &= \eta_k^2 \|e(k)\|^2 \|\Phi[X(k)]\|^2 \\ &\quad - 2\eta_k \|e(k)[e(k) - \mu(k)]\| \\ &\le \eta_k^2 \|e(k)\|^2 \|\Phi[X(k)]\|^2 \\ &\quad - 2\eta_k \|e(k)\|^2 + 2\eta_k \|e(k)\mu(k)\| \\ &\le \eta_k^2 \|e(k)\|^2 \|\Phi[X(k)]\|^2 \\ &\quad - 2\eta_k \|e(k)\|^2 + \eta_k \|e(k)\|^2 + \eta_k \|\mu(k)\|^2 \\ &= -\eta_k \|e(k)\|^2 \left(1 - \eta_k \left\|\Phi^T(X)\right\|^2\right) \\ &\quad + \eta_k \|\mu(k)\|^2. \end{aligned} \tag{18}$$

Since $\eta_k = \eta / 1 + \|\Phi[X(k)]\|^2$

$$\begin{aligned}\eta_k(1-\eta_k\|\Phi[x(k)]\|^2) &= \eta_k\Big(1-\frac{\eta}{1+\|\Phi[X(k)]\|^2}\\ &\qquad\times\|\Phi[X(k)]\|^2\Big)\\ &\ge \eta_k\left(1-\eta\frac{\max_k(\|\Phi[X(k)]\|^2)}{1+\max_k(\|\Phi[X(k)]\|^2)}\right)\\ &\ge \eta_k\left(1-\frac{\max_k(\|\Phi[X(k)]\|^2)}{1+\max_k(\|\Phi[X(k)]\|^2)}\right)\\ &= \frac{\eta_k}{1+\max_k(\|\Phi[X(k)]\|^2)}\\ &\ge \frac{\eta}{[1+\max_k(\|\Phi[X(k)]\|^2)]^2}.\end{aligned}$$

So

$$\Delta L_k \le -\pi\|e(k)\|^2 + \eta\|\mu(k)\|^2 \tag{19}$$

where π is defined as

$$\pi = \frac{\eta}{\left[1+\max_k(\|\Phi[X(k)]\|^2)\right]^2}.$$

Because

$$n\min\left(\widetilde{w}_i^2\right) \le L_k \le n\max\left(\widetilde{w}_i^2\right)$$

where $n\min\left(\widetilde{w}_i^2\right)$ and $n\max\left(\widetilde{w}_i^2\right)$ are $\mathcal{K}_\infty$-functions, and $\pi\|e(k)\|^2$ is an $\mathcal{K}_\infty$-function, $\eta\|\mu(k)\|^2$ is a $\mathcal{K}$-function. So L_k admits a ISS-Lyapunov function as in Definition 2. By Theorem 1, the dynamic of the identification error is input-to-state stable. From (14) and (17), we know L_k is the function of $e(k)$ and $\mu(k)$. The "INPUT" corresponds to the second term of (19), i.e., the modeling error $\mu(k)$. The "STATE" corresponds to the first term of (18), i.e., the identification error $e(k)$. Because the "INPUT" $\mu(k)$ is bounded and the dynamic is ISS, the "STATE" $e(k)$ is bounded.

Equation (18) can be rewritten as

$$\begin{aligned}\Delta L_k \le& -\eta\frac{\|e(k)\|^2}{\left[1+\max_k(\|\Phi[X(k)]\|^2)\right]^2} + \eta\|\mu(k)\|^2\\ \le& -\eta\frac{\|e(k)\|^2}{\left[1+\max_k(\|\Phi[X(k)]\|^2)\right]^2} + \eta\overline{\mu}.\end{aligned} \tag{20}$$

Summarizing (20) from 1 up to T, and by using $L_T > 0$ and L_1 is a constant, we obtain

$$L_T - L_1 \le -\eta\sum_{k=1}^{T}\|e_N(k)\|^2 + T\eta\overline{\mu}$$

$$\eta\sum_{k=1}^{T}\|e_N(k)\|^2 \le L_1 - L_T + T\eta\overline{\mu} \le L_1 + T\eta\overline{\mu}$$

(16) is established. ■

Remark 1: In general, a fuzzy neural model cannot match any nonlinear system exactly. The parameters of the fuzzy neural network will not converge to its optimal values. The idea of online identification proposed in this paper is to force the output of the fuzzy neural networks to follow the output of the plant. Although the parameters cannot converge to their optimal values, (16) shows that the normalized identification error will converge to a ball radius $\overline{\mu}$. If the fuzzy neural networks (8) can match the nonlinear plant (3) exactly $(\mu(k) = 0)$, i.e., we can find the best membership function $\mu_{A_{ji}}$ and W^* such that the nonlinear system can be written as $Y(k) = W^*\Phi\left[\mu_{A_{ji}}\right]$. Since $\|e(k)\|^2 > 0$, the same learning law (15) makes the identification error $\|e(k)\|$ asymptotically stable

$$\lim_{k\to\infty}\|e(k)\| = 0. \tag{21}$$

Remark 2: Normalizing learning rate η_k in (15) is time-varying in order to insure the stability of identification error. These learning rates are easier to be decided than [22] (for example we may select $\eta = 1$), without requiring any prior information. Time-varying learning rates can be found in some standard adaptive schemes [8]. But they need robust modifications to guarantee stability of the identification. Equation (15) is similar to the results of [15], however, the approaches are different. In this paper, the algorithm is derived from stability analysis (or ISS-Lyapunov function), the algorithm of [15] was obtained from minimization of the cost function. We focus on the bound of identification error, [15] focused on convergence analysis. It is interested to see that the two different methods can get similar results.

IV. Fuzzy Neural Modeling With Premise Membership Functions Learning

When we regard the plant as a black-box, neither the premise nor the consequent parameters are known (see [20], [22], and [24]). Now, the object of the fuzzy neural modeling is to find the center values of $B_{1i}\cdots B_{mi}$, as well as the membership functions $A_{1i}\ldots A_{ni}$, such that the fuzzy neural networks (8) can follow the nonlinear plant (3).

Gaussian membership function is exploited to identify fuzzy rules in this paper, which is defined by

$$\mu_{A_{ji}} = \exp\left(-\frac{(x_j - c_{ji})^2}{\sigma_{ji}^2}\right). \tag{22}$$

The qth output of the fuzzy neural model can be expressed as

$$\widehat{y}_q = \frac{\sum_{i=1}^{l} w_{qi}\prod_{j=1}^{n}\exp\left(-\frac{(x_j-c_{ji})^2}{\sigma_{ji}^2}\right)}{\left[\sum_{i=1}^{l}\prod_{j=1}^{n}\exp\left(-\frac{(x_j-c_{ji})^2}{\sigma_{ji}^2}\right)\right]}. \tag{23}$$

Let us define

$$z_i = \prod_{j=1}^{n}\exp\left(-\frac{(x_j-c_{ji})^2}{\sigma_{ji}^2}\right) \quad a_q = \sum_{i=1}^{l} w_{qi}z_i$$

$$b = \sum_{i=1}^{l} z_i.$$

So

$$\widehat{y}_q = \frac{a_q}{b}.$$

Similar to (13), the nonlinear plant (3) can be represented as

$$y_q = \frac{\sum_{i=1}^{l} w_{qi}^* \prod_{j=1}^{n} \exp\left(-\frac{\left(x_j - c_{ji}^*\right)^2}{\sigma_{ji}^{*2}}\right)}{\left[\sum_{i=1}^{l} \prod_{j=1}^{n} \exp\left(-\frac{\left(x_j - c_{ji}^*\right)^2}{\sigma_{ji}^{*2}}\right)\right]} - \mu_q \tag{24}$$

where w_{qi}^*, c_{ji}^* and σ_{ji}^{*2} are unknown parameters which may minimize the unmodeled dynamic μ_q.

In the case of three independent variables, a smooth function f has Taylor formula as

$$f(x_1, x_2, x_3) = \sum_{k=0}^{l-1} \frac{1}{k!} \left[\left(x_1 - x_1^0\right) \frac{\partial}{\partial x_1} + \left(x_2 - x_2^0\right) \frac{\partial}{\partial x_2} + \left(x_3 - x_3^0\right) \frac{\partial}{\partial x_3}\right]_0^k f + R_l$$

where R_l is the remainder of the Taylor formula. If we let x_1, x_2, and x_3 correspond to w_{pi}^*, c_{ji}^*, and σ_{ji}^{*2}, x_1^0, x_2^0, and x_3^0 correspond to w_{pi}, c_{ji}, and σ_{ji}^2

$$\begin{aligned} y_q + \mu_q = {} & \widehat{y}_q + \sum_{i=1}^{l} \frac{\left(w_{qi}^* - w_{qi}\right) z_i}{b} \\ & + \sum_{i=1}^{l} \sum_{j=1}^{n} \frac{\partial}{\partial c_{ji}} \left(\frac{a_q}{b}\right) \left(c_{ji}^* - c_{ji}\right) \\ & + \sum_{i=1}^{l} \sum_{j=1}^{n} \frac{\partial}{\partial \sigma_{ji}} \left(\frac{a_q}{b}\right) \left(\sigma_{ji}^* - \sigma_{ji}\right) + R_{1q} \end{aligned} \tag{25}$$

where R_{1q} is a second-order approximation error of the Taylor series, $q = 1 \dots m$. Using the chain rule, we get

$$\begin{aligned} \frac{\partial}{\partial c_{ji}} \left(\frac{a_q}{b}\right) &= \frac{\partial}{\partial z_i} \left(\frac{a_q}{b}\right) \frac{\partial z_i}{\partial c_{ji}} \\ &= \left(\frac{1}{b} \frac{\partial a_q}{\partial z_i} + \frac{\partial}{\partial z_i} \left(\frac{1}{b}\right) a_q\right) \left(2 z_i \frac{x_j - c_{ji}}{\sigma_{ji}^2}\right) \\ &= \left(\frac{w_{qi}}{b} - \frac{a_q}{b^2}\right) \left(2 z_i \frac{x_j - c_{ji}}{\sigma_{ji}^2}\right) \\ &= 2 z_i \frac{w_{qi} - \widehat{y}_q}{b} \frac{x_j - c_{ji}}{\sigma_{ji}^2} \\ \frac{\partial}{\partial \sigma_{ji}} \left(\frac{a_q}{b}\right) &= \frac{\partial}{\partial z_i} \left(\frac{a_q}{b}\right) \frac{\partial z_i}{\partial \sigma_{ji}} \\ &= 2 z_i \frac{w_{qi} - \widehat{y}_q}{b} \frac{\left(x_j - c_{ji}\right)^2}{\sigma_{ji}^3}. \end{aligned}$$

In matrix form

$$y_q + \mu_q = \widehat{y}_q - Z(k) \widetilde{W}_q - D_{Zq} \overline{C}_k E - D_{Zq} \overline{B}_k E + R_{1q} \tag{26}$$

where the equation shown at the bottom of the page holds. The identification error is defined as $e_q = \widehat{y}_q - y_q$, using (26) we have

$$e_q = Z(k) \widetilde{W}_q + D_{Zq} \overline{C}_k E + D_{Zq} \overline{B}_k E + \mu_q - R_{1q}. \tag{27}$$

In vector form

$$e(k) = \widetilde{W}_k Z(k) + D_z(k) \overline{C}_k E + D_z(k) \overline{B}_k E + \zeta(k) \tag{28}$$

where $e(k) = [e_1 \dots e_m]^T$

$$\widetilde{W}_k = \begin{bmatrix} w_{11} - w_{11}^* & & w_{m1} - w_{m1}^* \\ & \ddots & \\ w_{1l} - w_{1l}^* & & w_{ml} - w_{ml}^* \end{bmatrix}$$

$$D_z(k) = \begin{bmatrix} 2 z_1 \frac{w_{11} - \widehat{y}_1}{b} & & 2 z_l \frac{w_{1l} - \widehat{y}_1}{b} \\ & \ddots & \\ 2 z_1 \frac{w_{m1} - \widehat{y}_m}{b} & & 2 z_l \frac{w_{ml} - \widehat{y}_m}{b} \end{bmatrix}$$

$\zeta(k) = \mu - R_1$, $\mu = [\mu_1 \cdots \mu_m]^T$, $R_1 = [R_{11} \cdots R_{1m}]^T$.

By the bound of the Gaussian function ϕ and the plant is BIBO stable, μ and R_1 in (24) and (25) are bounded. So $\zeta(k)$ in (28) is bounded. The following theorem gives a stable algorithm for discrete-time Mamdani-type fuzzy neural networks.

$$Z(k) = \left[\frac{z_1}{b} \dots \frac{z_l}{b} y\right]^T \quad W_q = [w_{q1} \dots w_{ql}] \quad \widetilde{W}_q = W_q - W_q^*$$

$$D_{Zq} = \left[2 z_1 \frac{w_{q1} - \widehat{y}_q}{b}, \dots, 2 z_l \frac{w_{ql} - \widehat{y}_q}{b}\right] \quad E = [1, \dots 1]^T$$

$$\overline{C}_k = \begin{bmatrix} \frac{x_1 - c_{11}}{\sigma_{11}^2} \left(c_{11} - c_{11}^*\right) & & \frac{x_n - c_{n1}}{\sigma_{n1}^2} \left(c_{n1} - c_{n1}^*\right) \\ & \ddots & \\ \frac{x_1 - c_{1l}}{\sigma_{1l}^2} \left(c_{1l} - c_{1l}^*\right) & & \frac{x_n - c_{nl}}{\sigma_{nl}^2} \left(c_{nl} - c_{nl}^*\right) \end{bmatrix}$$

$$\overline{B}_k = \begin{bmatrix} \frac{\left(x_1 - c_{11}\right)^2}{\sigma_{11}^3} \left(\sigma_{11} - \sigma_{11}^*\right) & & \frac{\left(x_n - c_{n1}\right)^2}{\sigma_{n1}^3} \left(\sigma_{n1} - \sigma_{n1}^*\right) \\ & \ddots & \\ \frac{\left(x_1 - c_{1l}\right)^2}{\sigma_{1l}^3} \left(\sigma_{1l} - \sigma_{1l}^*\right) & & \frac{\left(x_n - c_{nl}\right)^2}{\sigma_{nl}^3} \left(\sigma_{nl} - \sigma_{nl}^*\right) \end{bmatrix}.$$

Theorem 3: If we use the Mamdani-type fuzzy neural network (23) to identify nonlinear plant (3), the following backpropagation algorithm makes identification error $e(k)$ bounded

$$\begin{aligned} W_{k+1} &= W_k - \eta_k e(k) Z(k)^T \\ c_{ji}(k+1) &= c_{ji}(k) \\ &\quad - 2\eta_k z_i \frac{w_{pi} - \widehat{y}_p}{b} \frac{x_j - c_{ji}}{\sigma_{ji}^2} (\widehat{y}_q - y_q) \\ \sigma_{ji}(k+1) &= \sigma_{ji}(k) \\ &\quad - 2\eta_k z_i \frac{w_{pi} - \widehat{y}_p}{b} \frac{(x_j - c_{ji})^2}{\sigma_{ji}^3} (\widehat{y}_q - y_q) \end{aligned} \tag{29}$$

where $(\eta_k = \eta/1 + \|Z\|^2 + 2\|D_z\|^2)$, $0 < \eta \le 1$. The average of the identification error satisfies

$$J = \limsup_{T\to\infty} \frac{1}{T} \sum_{k=1}^{T} \|e(k)\|^2 \le \frac{\eta}{\pi} \overline{\zeta} \tag{30}$$

where $\pi = (\eta/(1+\kappa)^2) > 0$, $\kappa = \max_k \left(\|Z\|^2 + 2\|D_z\|^2 \right)$, $\overline{\zeta} = \max_k \left[\|\zeta(k)\|^2 \right]$.

Proof: Let use define $\widetilde{c}_{ji}(k) = c_{ji}(k) - c_{ji}^*$, $\widetilde{b}_{ji}(k) = \sigma_{ji}(k) - \sigma^*(k)$, the element of $\widetilde{C}_k$ is expressed as $\widetilde{c}_{ji}(k) = \left[\widetilde{C}_k\right]$. So

$$\left[\widetilde{C}_{k+1}\right] = \left[\widetilde{C}_k\right] - 2\eta_k z_i \frac{w_{qi} - \widehat{y}_q}{b} \frac{x_j - c_{ji}}{\sigma_{ji}^2} (\widehat{y}_q - y_q).$$

We selected a positive defined scalar L_k as

$$L_k = \left\|\widetilde{W}_k\right\|^2 + \left\|\widetilde{C}_k\right\|^2 + \left\|\widetilde{B}_k\right\|^2. \tag{31}$$

From the updating law (29)

$$\widetilde{W}_{k+1} = \widetilde{W}_k - \eta_k e(k) Z(k)^T.$$

By using (28), we have

$$\begin{aligned} \Delta L_k &= \left\| \widetilde{W}_k - \eta_k e(k) Z(k)^T \right\|^2 \\ &\quad + \left\| \widetilde{C}_k - \left[2\eta_k z_i \frac{w_{qi} - \widehat{y}_q}{b} \frac{x_j - c_{ji}}{\sigma_{ji}^2} (\widehat{y}_q - y_q) \right] \right\|^2 \\ &\quad + \left\| \widetilde{B}_k - \left[2\eta_k z_i \frac{w_{qi} - \widehat{y}_q}{b} \frac{(x_j - c_{ji})^2}{\sigma_{ji}^3} (\widehat{y}_q - y_q) \right] \right\|^2 \\ &\quad - \left\|\widetilde{W}_k\right\|^2 - \left\|\widetilde{C}_k\right\|^2 - \left\|\widetilde{B}_k\right\|^2 \\ &= \eta_k^2 \|e(k)\|^2 \left(\left\|Z(k)^T\right\|^2 + 2\left\|D_z^T\right\|^2 \right) - 2\eta_k \|e(k)\| \\ &\quad \times \left\| \widetilde{W}_k Z(k)^T + D_z^T \overline{C}_k E + D_z^T \overline{B}_k E \right\| \\ &= \eta_k^2 \|e(k)\|^2 \left(\|Z\|^2 + 2\|D_z\|^2 \right) \\ &\quad - 2\eta_k \|e(k) [e(k) - \zeta(k)]\| \\ &\le -\eta_k \|e(k)\|^2 \left[1 - \eta_k \left(\|Z\|^2 + 2\|D_z\|^2 \right) \right] \\ &\quad + \eta \|\zeta(k)\|^2 \\ &\le -\pi \|e(k)\|^2 + \eta \|\zeta(k)\|^2 \end{aligned} \tag{32}$$

where π is defined as

$$\pi = \frac{\eta}{\left[1 + \max_k \left(\|Z\|^2 + 2\|D_z\|^2 \right) \right]^2}.$$

Because

$$\begin{aligned} n\left[\min\left(\widetilde{w}_i^2\right) + \min\left(\widetilde{c}_{ji}^2\right) + \min\left(\widetilde{b}_{ji}^2\right) \right] &\le L_k \\ &\le n\left[\max\left(\widetilde{w}_i^2\right) + \max\left(\widetilde{c}_{ji}^2\right) + \max\left(\widetilde{b}_{ji}^2\right) \right] \end{aligned}$$

where $n\left[\min\left(\widetilde{w}_i^2\right) + \min\left(\widetilde{c}_{ji}^2\right) + \min\left(\widetilde{b}_{ji}^2\right) \right]$ and $n\left[\max\left(\widetilde{w}_i^2\right) + \max\left(\widetilde{c}_{ji}^2\right) + \max\left(\widetilde{b}_{ji}^2\right) \right]$ are $\mathcal{K}_\infty$-functions, and $\pi\|e(k)\|^2$ is an $\mathcal{K}_\infty$-function, $\eta\|\zeta(k)\|^2$ is a $\mathcal{K}$-function. So L_k admits a ISS-Lyapunov function as in Definition 2. From Theorem 1, the dynamic of the identification error is input-to-state stable. From (28) and (31) we know V_k is the function of $e(k)$ and $\zeta(k)$. Because the "INPUT" $\zeta(k)$ is bounded and the dynamic is ISS, the "STATE" $e(k)$ is bounded.

Equation (32) can be rewritten as

$$\Delta L_k \le -\pi \|e(k)\|^2 + \eta \|\zeta(k)\|^2 \le \pi \|e(k)\|^2 + \eta \overline{\zeta}. \tag{33}$$

Summarizing (33) from 1 up to T, and by using $L_T > 0$ and L_1 is a constant, we obtain

$$\begin{aligned} L_T - L_1 &\le -\pi \sum_{k=1}^{T} \|e(k)\|^2 + T\eta\overline{\zeta} \\ \pi \sum_{k=1}^{T} \|e(k)\|^2 &\le L_1 - L_T + T\eta\overline{\zeta} \le L_1 + T\eta\overline{\zeta} \end{aligned}$$

(30) is established. ■

For TSK-type fuzzy neural model (9), we select A_{ji} as Gaussian functions. The qth output of the fuzzy logic system can be expressed as

$$\widehat{y}_q = \frac{\sum_{i=1}^{l} \left(\sum_{k=0}^{n} p_{qk}^i x_k \right) \prod_{j=1}^{n} \exp\left(-\frac{(x_j - c_{ji})^2}{\sigma_{ji}^2} \right)}{\left[\sum_{i=1}^{l} \prod_{j=1}^{n} \exp\left(-\frac{(x_j - c_{ji})^2}{\sigma_{ji}^2} \right) \right]} \tag{34}$$

where $x_0 = 1$ The part $\sum_{i=1}^{l} \left(w_{qi}^* - w_{qi} \right) z_i / b_q$ in (25) is changed as

$$\sum_{i=1}^{l} \left(\sum_{k=0}^{n} \left(p_{qk}^{i*} - p_{qk}^i \right) x_k \right) \frac{z_i}{b}.$$

The following theorem gives a stable algorithm for TSK-type fuzzy neural networks.

Theorem 4: If we use TSK-type fuzzy neural network (34) to identify nonlinear plant (3), the following algorithm makes identification error $e(k)$ bounded

$$\begin{aligned} p_{qk}^i(k+1) &= p_{qk}^i(k) - \eta_k (\widehat{y}_q - y_q) \frac{z_i}{b} x_k \\ c_{ji}(k+1) &= c_{ji}(k) - 2\eta_k z_i \frac{w_{pi} - \widehat{y}_p}{b} \frac{x_j - c_{ji}}{\sigma_{ji}^2} (\widehat{y}_q - y_q) \\ \sigma_{ji}(k+1) &= \sigma_{ji}(k) - 2\eta_k z_i \frac{w_{pi} - \widehat{y}_p}{b} \frac{(x_j - c_{ji})^2}{\sigma_{ji}^3} (\widehat{y}_q - y_q) \end{aligned}$$

where $\eta_k = (\eta/1 + \|Z\|^2 + 2\|D_z\|^2)$, $0 < \eta \le 1$.

Proof: The proof is the same as Theorem 3. ■

One can see that TSK requires only a different formulation compared to traditional Mamdani model and the difference between Theorems 3 and 4 is the inclusion of the update law for the $p_{qk}^{i}(k)$.

Remark 3: Like the second Lyapunov method, the condition $0 < \eta \leq 1$ is a necessary but not sufficient condition ensuring ISS of the learning process. It is possible to speed up convergence of the algorithms by considering values of η larger than unity. However, we cannot ensure the stability in whole identification period. The contradiction in stability $(\eta \leq 1)$ and fast convergence $(\eta > 1)$ still exists in the gradient-like algorithms. In this paper we are focus exclusively on the first issue rather on the last one.

Remark 4: If we select η as a dead-zone function

$$\begin{cases} \eta = 0, & \text{if } |e(k)| \leq \overline{\zeta} \\ \eta = \eta_0, & \text{if } |e(k)| > \overline{\zeta} \end{cases}$$

(29) is the same as [23]. If a σ-modification term or modified δ-rule term are added in k in (15) or (29), it becomes that of [25] or [12]. However, all of them need the upper bound of modeling error $\overline{\zeta}$, and the identification error is enlarged by the robust modifications [8].

Remark 5: Even if the input is persistent exciting, the modeling error $\zeta(k)$ will not make the weights converge to their optimal values. It is possible that the output error is convergent, but the weight errors are very high when the fuzzy rules are well defined. The relations of the output error and the weight errors are shown in (14) and (28). Simpler case is that we use(14) and fuzzy neural networks can match the nonlinear plant exactly

$$\begin{aligned} \text{plant: } & y = W^{*}\Phi[X(k)] \\ \text{fuzzy neural networks: } & \widehat{y} = W(k)\Phi[X(k)] \\ \text{output error: } & (y - \widehat{y}) = (W^{*} - W(k))\Phi[X(k)]. \end{aligned}$$

If $\Phi[X(k)]$ is large, small output error $(y - \widehat{y})$ does not mean good convergence of the weight error $W^{*} - W(k)$. This means that the weights do not gradually converge to certain values. In other words, the update laws should be always in alert for online identification. Several algorithms suggested in fuzzy online modeling follow this standard procedure [10], [23], [25].

V. SIMULATION

In this section, the suggested stable learning algorithms are applied to function approximation and nonlinear system identification.

A. Two-Dimensional Function Approximation

We the fuzzy neural networks (8) to approximate following nonlinear function:

$$f(x_1, x_2) = 0.52 + 0.1x_1 + 0.28x_2 - 0.6x_1x_2.$$

This example is taken from [23] where the following fuzzy system was used:

$$\widehat{f} = \frac{\sum_{x_1}\sum_{x_2} f(x_1, x_2)\mu_{A_1}\mu_{A_2}}{\sum_{x_1}\sum_{x_2} \mu_{A_1}\mu_{A_2}}. \tag{35}$$

In this paper, we use fuzzy neural network as

$$\widehat{f} = \sum_{x_1}\sum_{x_2} W_{x_1,x_2} \frac{\mu_{A_1}\mu_{A_2}}{\sum_{x_1}\sum_{x_2} \mu_{A_1}\mu_{A_2}}. \tag{36}$$

Compared with (35), the advantage of (36) is that it has adaptive ability. The number of fuzzy sets for each input variable is 2. The membership functions for x_1 and x_2 are triangular function in $[-1, 1]$, see Fig. 2. As in Fig. 1, $n = 2$, $l_1 = l_2 = 2$, $l = 4$, $m = 1$. We use the gradient descent learning (15) with $\eta = 1$. We first use 300 data $(T = 300)$ to train the model. The training data is selected as $x_1(k) = -1 + 2k/T$, $x_2(k) = 1 - 2k/T$, $k = 1, 2, \ldots T$. Then we fixed the weights W_{x_1,x_2}, use another 300 data to test the model. The testing data is $x_1(k) = 1 - 2k/T$, $x_2(k) = -1 - 2k/T$, $k = 1, 2, \ldots T$. The identification results are shown in Fig. 3. Let us define the mean squared error for finite time is

$$J_1(N) = \frac{1}{2N}\sum_{k=1}^{N} e^2(k). \tag{37}$$

In the training phase $J_1(300) = 0.0016$, in the testing phase $J_1(300) = 0.0018$. Following Remark 5, the online identification algorithm cannot make the weights converge to the optimal values after a certain period training. Even for this simple nonlinear function, the testing result is not quit encouraging.

In this example, we found that the stability limit for η is about 2. This limit depends on model parameters, such as initial condition of W_{x_1,x_2} the number of fuzzy set, etc. Although $1 < \eta < 2$ can speedup the learning process, we cannot guarantee the stability for any condition and whole learning period. Theorem 2 insures the modeling error is stable for any condition when $0 < \eta \leq 1$.

Modeling errors $\mu(k)$ in (13) and $\zeta(k)$ in (28) depend on the complexity of the particular model selected and how close it is to the actual plant. In this example, if we select $l_1 = l_2 = 8$, all the other conditions do not change. The training result is shown in Fig. 4. The identification error is enlarged as $J_1(300) = 0.0021$. The worse results obtained are due to the redundant fuzzy rules, or the rules (memberships) are not well defined. From the point of identification, it is because the model is not close to the plant. We should mention that model structure influences modeling error, but does not destroy stability of identification process.

B. Nonlinear System Identification

We use a nonlinear system to illustrate the backpropagation algorithm (29). The identified plant is [18, Ex. 2], which was also discussed in [10], [17], and [22]

$$y(k) = \frac{y(k-1)y(k-2)[y(k-1) + 2.5]}{1 + y(k-1)^2 + y(k-2)^2} + u(k-1). \tag{38}$$

The input signal is selected as random number in the interval $[0, 1]$ We use the fuzzy neural network (23) to identify (38)

$$X(k) = [y(k-1), y(k-2), u(k-1)]^T.$$

In order to avoid to computation burden, we use single connection, i.e., the input for ϕ_k is only

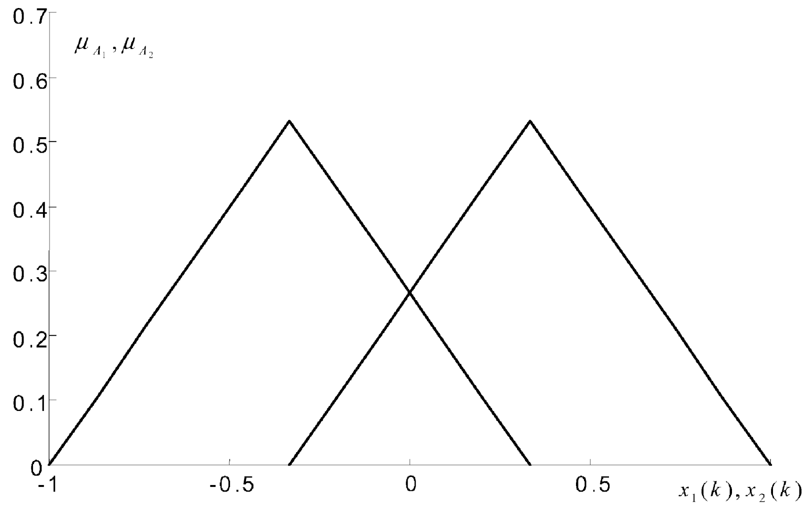


Fig. 2. Membership functions.

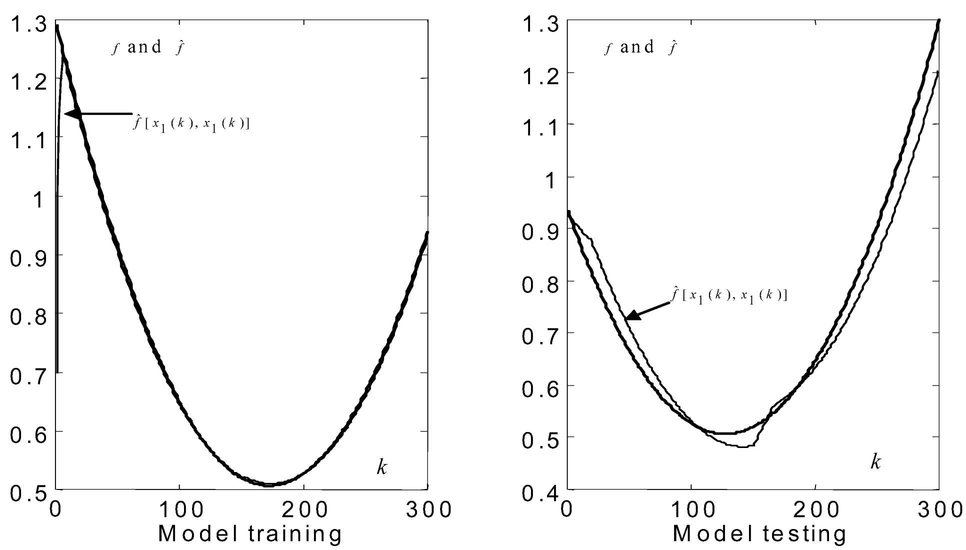


Fig. 3. Function approximation.

$A_{1k}, A_{2k}, \ldots A_{nk}$ $(k = 1 \ldots l)$, $l = 6$, $m = 1$. The membership functions are chosen as Gaussian functions

$$\mu_{A_{ji}}(k) = \exp\left(-\frac{(x_j - c_{ji})^2}{\sigma_{ji}^2}\right).$$

We assume the membership function A_{ji} is unknown. We use (29) with $\eta = 1$ to update W_k, c_{ji}, and σ_{ji}. The initial conditions for W_k, c_{ji}, and σ_{ji} are random delected in the range $[0, 1]$. The online identification results are shown in Fig. 5.

Now, we compare our algorithm with normal backpropagation algorithm [18] and optimal learning [22]. We use the same multilayer neural networks as [18], it is $\Pi_{3,20,10,1}$ (two hidden layers with 20 and ten nodes), and a fixed learning rate $\eta = 0.05$. In this simulation, we found after $\eta > 0.1$ the neural networks become unstable. We also repeated simulation of [22, Ex. 3]. The performance comparison can be realized by mean squared errors (37). The comparison results are shown in Fig. 6.

We can see that the optimal learning for fuzzy neural networks [22] is the best with respect to identifications error. However, it is difficult to realize because we have to solve a equation $A\beta^2 + B\beta = 0$, or use genetic algorithm to find the optimal learning rate. The stable algorithm proposed in this paper has almost the same convergence rate as the optimal learning. Although the identification error is bigger, $J_1(200) = 0.051$, but it is simple and easy to realize. Normal backpropagation algorithm for multilayer neural networks has a slow convergence speed and a big identification error, $J_1(200) = 0.1078$.

VI. Conclusion

This paper applies ISS approach to Mamdani- and TSK-type fuzzy neural networks and proposes robust learning algorithms which can guarantee the stability of training process. The proposed algorithms are effective. The main contributions are as follows.

1) By using ISS approach, we conclude that the commonly-used robustifying techniques in discrete-time fuzzy neural modeling, such as projection and dead-zone, are not necessary.
2) New algorithms with time-varying learning rates are proposed, which are robust to any bounded uncertainty.

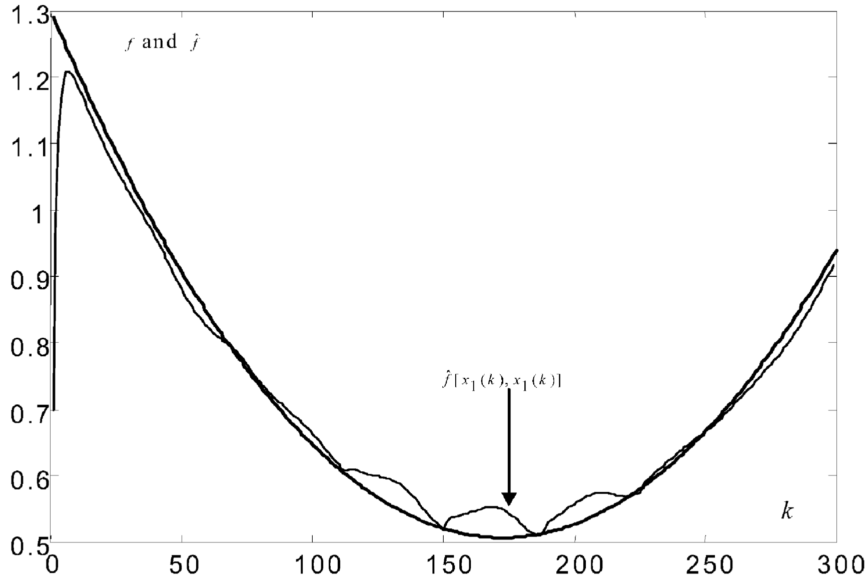


Fig. 4. Function approximation with redundant fuzzy rules.

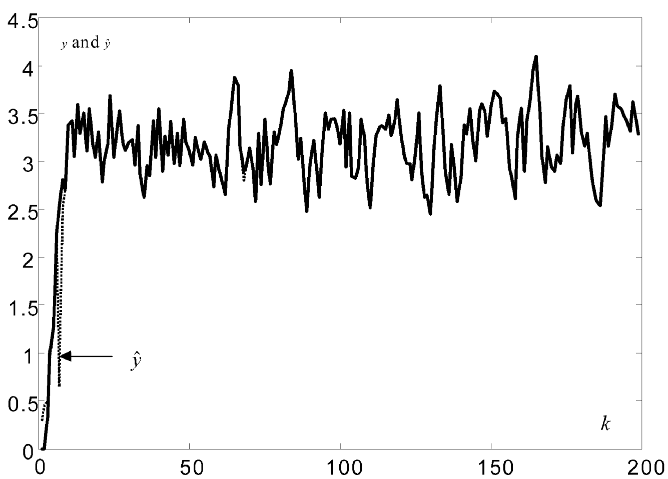


Fig. 5. Nonlinear system online identification.

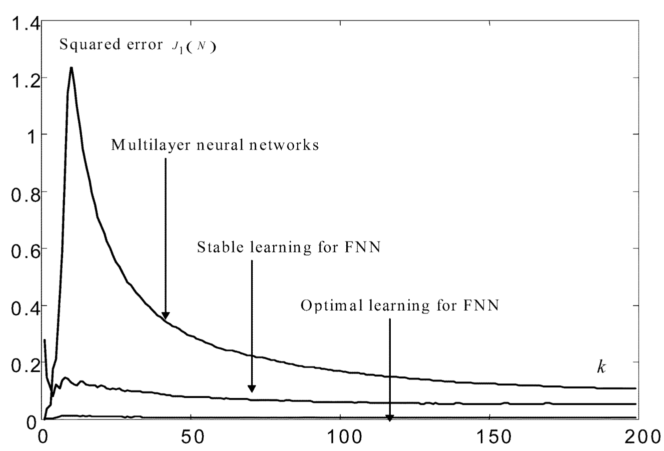


Fig. 6. Perform comparision.

Further works will be done on structure learning. It will assure that the fuzzy regions and respective fuzzy sets are well posed within the premise space according to the particular system under consideration. Combined with such a scheme, the weight update laws will provide a smaller modeling error within a certain small ball.

References

[1] M. Brown and C. J. Harris, *Neurofuzzy Aadaptive Modeling and Control*. Upper Saddle River, NJ: Prentice-Hall, 1994.

[2] W. C. Chan, C. W. Chan, K. C. Cheung, and Y. Wang, "Modeling of nonlinear stochastic dynamical systems using neurofuzzy networks," in *Proc. 38th IEEE Conf. Decision Control*, Phoenix, AZ, 1999, pp. 2643–2648.

[3] D. S. Chen and R. C. Jain, "A robust back propagation learning algorithm for fuunction approximation," *IEEE Trans. Neural Networks*, vol. 5, pp. 467–479, May 1994.

[4] M. Y. Chen and D. A. Linkensm, "A systematic neuro-fuzzy modeling framework with application to material property prediction," *IEEE Trans. Syst., Man, Cybern. B*, vol. 31, pp. 781–790, Sept. 2001.

[5] B. Egardt, *Stability of Adaptive Controllers*. Berlin, Germany: Springer-Verlag, 1979, vol. 20, Lecture Notes in Control and Information Sciences.

[6] S. Haykin, *Neural Networks—A Comprehensive Foundation*. New York: MacMillan, 1994.

[7] S. I. Horikawa, T. Furuhashi, and Y. Uchikawa, "On fuzzy modeling using fuzzy neual networks with the back-propagation algorithm," *IEEE Trans. Neural Networks*, vol. 3, pp. 801–806, Sept. 1992.

[8] P. A. Ioannou and J. Sun, *Robust Adaptive Control*. Upper Saddle River, NJ: Prentice-Hall, 1996.

[9] Z. P. Jiang and Y. Wang, "Input-to-state stability for discrete-time nonlinear systems," *Automatica*, vol. 37, no. 2, pp. 857–869, 2001.

[10] C. F. Juang, "A TSK-type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithm," *IEEE Trans. Fuzzy Syst.*, vol. 10, pp. 155–170, Apr. 2002.

[11] Y. G. Leu, T. T. Lee, and W. Y. Wang, "Observer-based adaptive fuzzy-neural control for unknown nonlinear dynamical systems," *IEEE Trans. Syst., Man, Cybern. B*, vol. 29, pp. 583–591, June 1999.

[12] F. L. Lewis, A. Yesildirek, and K. Liu, "Multilayer neural-net robot controller with guaranteed tracking performance," *IEEE Trans. Neural Networks*, vol. 7, pp. 388–399, Mar. 1996.

[13] C. T. Lin and G. Lee, *Neural Fuzzy Systems: A Neural-Fuzzy Synergism to Intelligent Systems*. Upper Saddle River, NJ: Prentice-Hall, 1996.

[14] C. T. Lin, "A neual fuzzy control system with structure and parameter learning," *Fuzzy Sets Syst.*, vol. 70, pp. 183–212, 1995.

[15] D. P. Mandic, A. I. Hanna, and M. Razaz, "A normalized gradient descent algorithm for nonlinear adaptive filters using a gradient adaptive step size," *IEEE Signal Processing Lett.*, vol. 8, pp. 295–297, Feb. 2001.

[16] E. H. Mamdani, "Application of fuzzy algorithms for control of simple dynamic plant," *Proc. Inst. Elect. Eng.*, vol. 121, no. 12, pp. 1585–1588, 1976.

[17] P. A. Mastorocostas and J. B. Theocharis, "A recurrent fuzzy-neural model for dynamic system identification," *IEEE Trans. Syst., Man, Cybern. B*, vol. 32, pp. 176–190, Feb. 2002.

[18] K. S. Narendra and S. Mukhopadhyay, "Adaptive control using neural networks and approximate models," *IEEE Trans. Neural Networks*, vol. 8, pp. 475–485, May 1997.

[19] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, pp. 116–132, Jan. 1985.

[20] H. H. Tsai and P. T. Yu, "On the optimal design of fuzzy neural networks with robust learning for function approximation," *IEEE Trans. Syst., Man, Cybern. B*, vol. 30, pp. 217–223, Apr. 2000.

[21] S. Wu and M. J. Er, "Dynamic fuzzy neural networks- a novel approach to function approximation," *IEEE Trans. Syst., Man, Cybern. B*, vol. 30, pp. 358–364, 2000.

[22] C. H. Wang, H. L. Liu, and C. T. Lin, "Dynamic optimal learning rates of a certain class of fuzzy neural networks and its applications with genetic algorithm," *IEEE Trans. Syst., Man, Cybern. B*, vol. 31, pp. 467–475, June 2001.

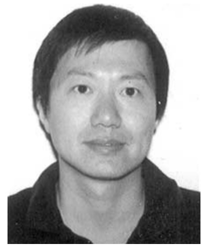
[23] L. X. Wang, *Adaptive Fuzzy Systems and Control*. Upper Saddle River, NJ: Prentice-Hall, 1994.

[24] W. Y. Wang, T. T. Lee, C. L. Liu, and C. H. Wang, "Function approximation using fuzzy neural networks with robust learning algorithm," *IEEE Trans. Syst., Man, Cybern. B*, vol. 27, pp. 740–747, Sept. 1997.

[25] W. Y. Wang, Y. G. Leu, and C. C. Hsu, "Robust adaptive fuzzy-neural control of nonlinear dynamical systems using generalized projection updated law and variable structure controller," *IEEE Trans. Syst., Man, Cybern. B*, vol. 31, pp. 140–147, 2001.

[26] W. Yu and X. Li, "Some stability properties of dynamic neural networks," *IEEE Trans. Circuits Syst. I*, vol. 48, pp. 256–259, Jan. 2001.

[27] ——, "Some new results on system identification with dynamic neural networks," *IEEE Trans. Neural Networks*, vol. 12, pp. 412–417, Mar. 2001.



Wen Yu received the B.S. degree from Tsinghua University, Beijing, China, in 1990 and the M.S. and Ph.D. degrees, both in electrical engineering, from Northeastern University, Shenyang, China, in 1992 and 1995, respectively.

From 1995 to 1996, he served as a Lecturer in the Department of Automatic Control at Northeastern University, Shenyang, China. In 1996, he joined CINVESTAV-IPN, Mexico, where he is a Professor in the Departamento de Control Automático. He has held a research position with the Instituto Mexicano del Petroleo from December 2002 to November 2003. His research interests include adaptive control, neural networks, and fuzzy control.

Xiaoou Li received the B.S. and Ph.D. degrees in applied mathematics and electrical engineering from Northeastern University, Shenyang, China, in 1991 and 1995, respectively.

From 1995 to 1997, she was a Lecturer in the Department of Automatic Control of Northeastern University. From 1998 to 1999, she was an Associate Professor with Centro de Instrumentos, UNAM, Mexico. Since 2000, she has been a Professor with Sección de Computación, Departamento de Ingeniería Eléctrica, CINVESTAV-IPN, Mexico. Her research interests include Petri net theory and application, neural networks, advanced database systems, computer integrated manufacturing, and fuzzy systems.