Neural Processing Letters (2005) 22:391–404
DOI 10.1007/s11063-005-1523-4

State-Space Recurrent Fuzzy Neural Networks for Nonlinear System Identification

WEN YU
Departamento de Control Automatico, CINVESTAV-IPN, A.P. 14-740, Av.IPN 2508, México, D.F. 07360, México. e-mail: yuw@ctrl.cinvestav.mx

Abstract. In this paper, we propose a new recurrent fuzzy neural network, which has the standard state space form, we call it state-space recurrent neural networks. Input-to-state stability is applied to access robust training algorithms for system identification. Stable learning algorithms for the premise part and the consequence part of fuzzy rules are proved.

Key words. fuzzy neural networks, stability, system identification

1. Introduction

Neural networks can be classified as feedforward and recurrent ones. Feedforward networks have been shown to obtain successful results in system identification [4]. But they use static mapping schemes, the weights' updating do not utilize information on the local data structure, so the function approximation is sensitive to the training data [7]. Recurrent networks incorporate feedback, they have powerful representation capability and can overcome disadvantages of feedforward networks [4]. It is widely known, both neural networks and fuzzy logic are universal estimators, they can approximate any nonlinear function to any prescribed accuracy, provided that sufficient hidden neurons or fuzzy rules are available. Recent results show that the fusion procedure of the two different technologies, called fuzzy neural network (FNN), seems to be very effective for system identification, when we do not have complete plant information [2, 12]. A major drawback of FNN is that its application domain is limited to static problems due to its feedforward networks structure. To deal with this problem, interest in using recurrent fuzzy neural networks (RFNNs) has been steadily growing in recent years. RFNNs can be divided into several classes. For example, fuzzy neural networks with external feedback [24], or fuzzy models with internal recurrence [9, 15], the feedback loop can be put in the premise part [9], in the consequence part [8, 15] or in the output of the whole fuzzy system [24].

Gradient decent learning and backpropagation algorithm are always used to adjust the parameters of the membership functions and the weights of defuzzification. The slow rate of learning and local minimum are the main drawbacks of these learning algorithms [13]. Some modifications were derived. Chen and Jain [1] suggested a robust BP learning in order to resist the noise effect and reject errors drift during the approximation. Wang et al. [21] used B-spline membership functions to minimize robust object function, and the convergence speed was improved. RBF neural networks were applied in [18] to determine the structure and parameters of fuzzy neural systems. Despite their successful applications, the above algorithms exhibit a problem during training as they lack stability analysis.

For engineers it is very important to ensure stability in the theory before they would like to apply fuzzy neural modeling technique in a real system. It is well known that normal identification algorithms (e.g., gradient decent and lease square) are stable for ideal conditions. In the presence of unmodeled dynamics, these adaptive procedures can go to instability easily. The lack of robustness of the parameter identification was demonstrated in [3] and became a hot issue in 1980s, when some robust modification techniques for adaptive identification was suggested [5].

The membership functions' updating of the neuro-fuzzy modeling can be regarded as parameters identification. The normal gradient decent learning and backpropagation algorithm are stable when the fuzzy neural model can match the nonlinear plant exactly. Generally we have to apply some modifications to these algorithms such that the learning processes are robust stable. The projection operator is very effective to insure the parameters bounded for fuzzy modeling [20] and fuzzy-neural systems [11]. Another generalized method is to use robust modification techniques [5] in fuzzy neural modeling. For example, Wang et al. [21] applied switch σ-modification, in order to prevent parameter drift. In [19], the authors suggested a stable and optimal learning rate without robust modification, but this optimal rate is difficult to be found.

In this paper we propose a new RFNNs, namely state-space recurrent fuzzy neural networks (SRFNNs), where the feedback is in the output of the whole fuzzy system. In [24], the whole state feedback are in linear form, it is Takagi–Sugeno type [16] recurrent fuzzy neural. This paper give a Mamdani-type [14] RFNNs, the final form is a nonlinear state-space equation, so it is very suitable to model a nonlinear process which is expressed in state space framework. When the membership functions of the consequence part (THEN part) are unknown, gradient decent learning is used. When then membership functions of the premise part (IF part) are unknown, backpropagation-like algorithm is used. Time-varying learning rates are obtained by input-to-state stability (ISS) approach to update the parameters of the membership functions, these learning laws can assure stability in the training process. Compared with [19], the algorithms proposed in this paper is more simple, which can be calculated directly by input/output data.

2. State-Space Recurrent Fuzzy Neural Networks

In order to identify following state-space nonlinear system

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

where $u(k) \in \Re^m$ is the input vector, $x(k) \in \Re^n$ is a state vector, $u(k) \in \Re^l$ is the output vector, and f is a general nonlinear smooth function $f \in C^\infty$, we use a fuzzy model which is presented as a collection of fuzzy rules in the following form:

Rule i is

without control input IF $x_1(k)$ is A_{1i} and $x_2(k)$ is A_{2i} and $\cdots x_n(k)$ is A_{ni} THEN $\alpha\widehat{x_1}(k+1)$ is B_{1i} and $\cdots$ is $\alpha\widehat{x_n}(k+1)$ is B_{ni}

or

with control input IF $x_1(k)$ is D_{1i} and $x_2(k)$ is D_{2i} and $\cdots x_n(k)$ is D_{ni} THEN $\alpha\widehat{x_1}(k+1)$ is u_1C_{1i} and $\cdots$ is $\alpha\widehat{x_n}(k+1)$ is u_mC_{mi}

or

$$\text{recurrent} \quad \widehat{x}(k+1) = \alpha A\widehat{x}(k) \tag{2}$$

where matrix $A \in \Re^{n\times n}$ is a stable matrix which will be specified later, α is positive constant, the following theorem will show how to select it. $\widehat{x}_j(k)$ is the state of the fuzzy model, A_{ji}, B_{ji}, C_{ji} and D_{ji} are standard fuzzy sets [20]. We use l fuzzy IF–THEN rules $(i = 1, 2, \ldots, l)$ to perform a mapping from an input linguistic vector $[\widehat{x_1}(k)\cdots\widehat{x_n}(k), u_1\cdots u_m] \in \Re^{n+m}$ to an output linguistic vector $[\widehat{x_1}(k+1)\cdots\widehat{x_n}(k+1)]$. For each input variable x_i there are l_i fuzzy sets. In the case of full connection, $l = l_1 \times l_2 \cdots \times l_n$. By using product inference, center-average and singleton fuzzifier [20], the pth output of the fuzzy logic system can be expressed as

$$\begin{aligned}
\widehat{x}_p(k+1) &= \alpha\left(\sum_{i=1}^{l} w_{1pi}\left[\prod_{j=1}^{n}\mu_{A_{ji}}\right]\right)\Big/\left(\sum_{i=1}^{l}\left[\prod_{j=1}^{n}\mu_{A_{ji}}\right]\right) \\
&= \alpha\sum_{i=1}^{l} w_{1pi}\phi_{1i} \\
\widehat{x}_p(k+1) &= \alpha\left(\sum_{i=1}^{l} u_i w_{2pi}\left[\prod_{j=1}^{n}\mu_{D_{ji}}\right]\right)\Big/\left(\sum_{i=1}^{l}\left[\prod_{j=1}^{n}\mu_{D_{ji}}\right]\right) \\
&= \alpha\sum_{i=1}^{l} w_{2pi}\phi_{2i}u_i
\end{aligned} \tag{3}$$

where $\mu_{A_{ji}}$ and $\mu_{D_{ji}}$ are the membership functions of the fuzzy sets A_{ji} and D_{ji}, w_{1pi} is the point at which $\mu_{B_{pi}} = 1$, w_{2pi} is the point at which $\mu_{C_{pi}} = 1$. If we define

$$\begin{aligned}
\phi_{1i} &= \prod_{j=1}^{n}\mu_{A_{ji}}\Big/\textstyle\sum_{i=1}^{l}\displaystyle\prod_{j=1}^{n}\mu_{A_{ji}} \\
\phi_{2i} &= \prod_{j=1}^{n}\mu_{D_{ji}}\Big/\textstyle\sum_{i=1}^{l}\displaystyle\prod_{j=1}^{n}\mu_{D_{ji}}
\end{aligned} \tag{4}$$

(3) can be expressed in matrix form

$$\beta\widehat{x}(k+1) = A\widehat{x}(k) + W_1\sigma_1[x(k)] + W_2\sigma_2[x(k)]U(k) \tag{5}$$

where $\beta = \frac{1}{\alpha}$, the parameter

$$W_1 = \begin{bmatrix} w_{111} & & w_{11l} \\ & \ddots & \\ w_{1n1} & & W_{1nl} \end{bmatrix}, \quad W_2 = \begin{bmatrix} w_{211} & & w_{21l} \\ & \ddots & \\ w_{2n1} & & W_{2nl} \end{bmatrix}$$

data vectors $\sigma_1[x(k)] = [\phi_{11} \cdots \phi_{1l}]^T$, $\sigma_2[x(k)]U(k) = [\phi_{21}u_1 \cdots \phi_{2m}u_m 0, \cdots 0]^T$. This is series-parallel model (the input is from state of the plant) [12]. The structure of the series-parallel SRFNNs is shown in Figure 1.

If the rules are changed as

IF $\widehat{x}_1(k)$ is A_{1i} and $\widehat{x}_2(k)$ is A_{2i} and $\cdots \widehat{x}_n(k)$ is A_{ni}
THEN $\alpha\widehat{x_1}(k+1)$ is B_{1i} and $\cdots \alpha\widehat{x_n}(k+1)$ is B_{ni}

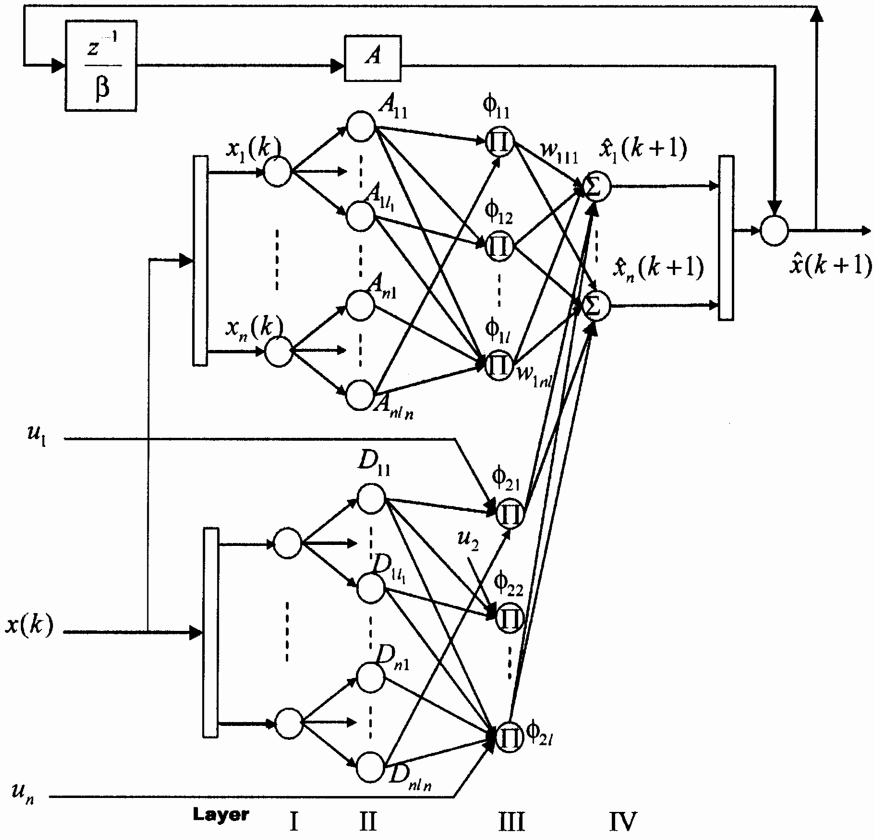

Figure 1. Series-parallel SRFNNs.

or

$$\text{IF } \widehat{x}_1(k) \text{ is } D_{1i} \text{ and } \widehat{x}_2(k) \text{ is } D_{2i} \text{ and } \cdots \widehat{x}_n(k) \text{ is } D_{ni}$$
$$\text{THEN } \alpha\widehat{x}_1(k+1) \text{ is } u_1 C_{1i} \text{ and } \cdots \alpha\widehat{x}_n(k+1) \text{ is } u_m C_{mi}$$

or

$$\widehat{x}(k+1) = \alpha A\widehat{x}(k) \tag{6}$$

The RFNNs is

$$\beta\widehat{x}(k+1) = A\widehat{x}(k) + W_1\sigma_1[\widehat{x}(k)] + W_2\sigma_2[\widehat{x}(k)]U(k) \tag{7}$$

(7) is parallel model (the input is from state of the model). The structure of the parallel SRFNNs is shown in Figure 2. In this paper, we only discuss series-parallel model, some similar results can be obtained for the parallel model.

The feedforward part is a four layers fuzzy neural networks which is discussed in many papers [12, 19]. Layer I accepts input linguistic vector $x(k)$. Each node

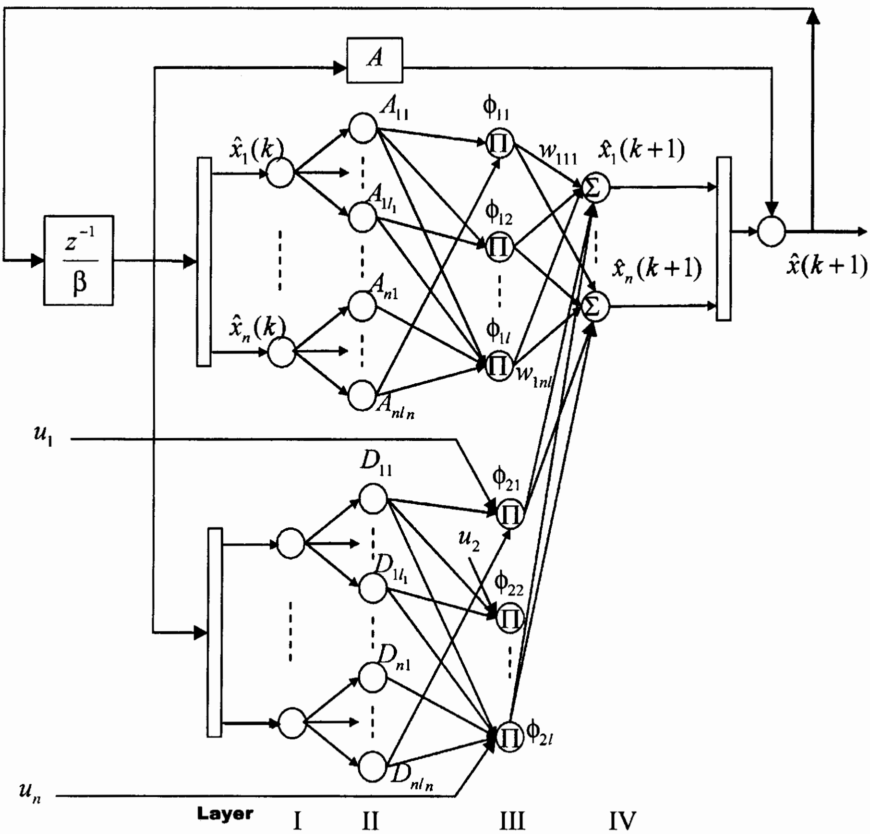

Figure 2. Parallel SRFNNs.

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 layers III and IV are full connected by the weight matrix W. Layers I and II are the premise part. Layers III and IV are the consequent part.

3. System Identification via SRFNN

The identified nonlinear system is represented as in (1). We assume the plant is BIBO stable, i.e., $x(k)$ and $u(k)$ are bounded. According to the Stone–Weierstrass theorem [4], this general nonlinear smooth function can be written as

$$\beta x(k+1) = Ax(k) + W_1^*\sigma_1[x(k)] + W_2^*\sigma_2[x(k)]U(k) + \mu(k) \tag{8}$$

where W_1^* and W_2^* are constant weights which can minimize the modeling error $\mu(k)$. Since σ_1 and σ_2 are bounded functions, $\mu(k)$ is bounded as $\mu^2(k) \leq \overline{\mu}$, where $\overline{\mu}$ is a positive constant. The neuro identification error is defined as

$$e(k) = \widehat{x}(k) - x(k) \tag{9}$$

From (8) and (9) we get

$$\begin{aligned}\beta e(k+1) &= Ae(k) + \widetilde{W}_1(k)\sigma_1[x(k)] \\ &\quad + \widetilde{W}_2(k)\sigma_2[x(k)]U(k) + \zeta(k)\end{aligned} \tag{10}$$

where $\widetilde{W}_1(k) = W_1(k) - W_1^*$, $\widetilde{W}_2(k) = W_2(k) - W_2^*$, $\varepsilon_1(k)$ is the second-order approximation error. σ' is the derivative of nonlinear activation function $\sigma(.)$ at the point of $W_1(k)$. Since ϕ is a sigmoid activation function, $\varepsilon(k)$ is bounded as $\|\varepsilon_1(k)\|^2 \leq \overline{\varepsilon}_1, \overline{\varepsilon}_1$ is an unknown positive constant, where $\zeta(k) = \varepsilon_1(k) + \varepsilon_2(k) - \mu(k)$. The following theorem gives a stable learning algorithm of discrete-time single-layer neural network.

THEOREM 1. *If we use RFNN (5) to identify nonlinear plant (1) and the eigenvalues of A are selected as $-1 < \lambda(A) < 0$, the following gradient updating law without robust modification can make identification error $e(k)$ bounded (stable in an L_∞ sense)*

$$\begin{aligned} W_1(k+1) &= W_1(k) - \eta(k)\sigma_1 e(k)^{\mathrm{T}} \\ W_2(k+1) &= W_2(k) - \eta(k)u(k)\sigma_2 e^{\mathrm{T}}(k) \end{aligned} \tag{11}$$

where if $\beta\|e(k+1)\| \geq \|e(k)\|, \eta(k) = \eta/1 + \|\sigma\|^2 + 1 + \|u\sigma\|^2$; if $\beta\|e(k+1)\| < \|e(k)\|, \eta(k) = 0.0 < \eta \leq 1$.

Proof. We select Lyapunov function as

$$V(k) = \|\widetilde{W}_1(k)\|^2 + \|\widetilde{W}_1(k)\|^2 \tag{12}$$

where $\|\widetilde{W}_1(k)\|^2 = \sum_{i=1}^{n} \widetilde{w}_1(k)^2 = \mathrm{tr}\{\widetilde{W}_1^{\mathrm{T}}(k)\widetilde{W}_1(k)\}$. From the updating law (11)

$$\widetilde{W}_1(k+1) = \widetilde{W}_1(k) - \eta(k)\sigma_1 e(k)^{\mathrm{T}} \tag{13}$$

So,

$$\begin{aligned}\Delta V(k) &= V(k+1) - V(k)\\ &= \|\widetilde{W}_1(k) - \eta(k)\sigma_1 e(k)^{\mathrm{T}}\|^2 - \|\widetilde{W}_1(k)\|^2\\ &\quad + \|\widetilde{W}_2(k) - \eta(k)u(k)\sigma_2 e^{\mathrm{T}}(k)\|^2 - \|\widetilde{W}_2(k)\|^2\\ &= \eta^2(k)\|e(k)\|^2\|\sigma_1\|^2 - 2\eta(k)\|\sigma_1\widetilde{W}_1(k)e^{\mathrm{T}}(k)\|\\ &\quad + \eta^2(k)\|e(k)\|^2\|u(k)\sigma_2\|^2 - 2\eta(k)\|u(k)\sigma_2\widetilde{W}_2(k)e^{\mathrm{T}}(k)\|\end{aligned} \tag{14}$$

Using (10) and $\eta(k)\ge 0$, there exist a small $\beta > 0$, such that $\|\beta e(k+1)\| \ge \|e(k)\|$

$$\begin{aligned}&-2\eta(k)\|\sigma_1\widetilde{W}_1 e^{\mathrm{T}}(k)\| - 2\eta(k)\|u(k)\sigma_2\widetilde{W}_2 e^{\mathrm{T}}(k)\|\\ &\quad \le -2\eta(k)\|e^{\mathrm{T}}(k)\|\|\beta e(k+1) - Ae(k) - \zeta(k)\|\\ &\quad = -2\eta(k)\|e^{\mathrm{T}}(k)\beta e(k+1) - e^{\mathrm{T}}(k)Ae(k) - e^{\mathrm{T}}(k)\zeta(k)\|\\ &\quad \le -2\eta(k)\|e^{\mathrm{T}}(k)\beta e(k+1)\|\\ &\qquad + 2\eta(k)e^{\mathrm{T}}(k)Ae(k) + 2\eta(k)\|e^{\mathrm{T}}(k)\zeta(k)\|\\ &\quad \le -2\eta(k)\|e(k)\|^2 + 2\eta(k)\lambda_{\max}(A)\|e(k)\|^2\\ &\qquad + \eta(k) + \|e(k)\| + \eta(k)\|\zeta(k)\|^2\end{aligned} \tag{15}$$

Since $0 < \eta \le 1$,

$$\begin{aligned}\Delta V(k) &\le \eta^2(k)\|e(k)\|^2\|\sigma' x(k)\|^2\\ &\quad + \eta^2(k)\|e(k)\|^2\|u(k)\phi' x(k)\|^2 + 2\eta(k)\lambda_{\max}(A)\|e(k)\|^2\\ &\quad - \eta(k)\|e(k)\|^2 + \eta(k)\|\zeta(k)\|^2\\ &= -\eta(k)[1 - 2\lambda_{\max}(A)]\\ &\quad + \eta(k)\frac{\|\sigma' x(k)\|^2 + \|u(k)\phi' x(k)\|^2}{1 + \|\sigma' x(k)\|^2 + \|u(k)\phi' x(k)\|^2}\eta e^2(k)\\ &\quad + \eta_k\zeta^2(k) \le -\pi e^2(k) + \eta\zeta^2(k)\end{aligned} \tag{16}$$

where

$$\pi = \frac{\eta}{1+k}\left[1 - \frac{k}{1+k}\right], \quad k = \max_k\left(\|\sigma' x(k)\|^2 + \|u(k)\phi' x(k)\|^2\right).$$

Since $-1 \le \lambda(A) < 0, \pi > 0$

$$n\min(\widetilde{w}_i^2) \le V_k \le n\max(\widetilde{w}_i^2) \tag{17}$$

where $n \times \min(\widetilde{w}_i^2)$ and $n \times \max(\widetilde{w}_i^2)$ are $\kappa_\infty - functions$, and $\pi e^2(k)$ is an κ_∞-function, $\eta\zeta^2(k)$ is a κ-function, so V_k admits the smooth ISS–Lyapunov function as in *Definition 2*. From Theorem 1, the dynamic of the identification error is input-to-state stable. The "INPUT" is corresponded to the second term of the last line in (16), i.e., the modeling error $\zeta(k)=\varepsilon_1(k)+\varepsilon_2(k)-\mu(k)$, the "STATE" is corresponded to the first term of the last line in (16), i.e., the identification error $e(k)$. Because the "INPUT" $\zeta(k)$ is bounded and the dynamic is ISS, the "STATE"$e(k)$ is bounded. When $\beta\|e(k+1)\|<\|e(k)\|, \Delta V(k)=0$. V_k is constant, the constant of $W_1(k)$ means $e(k)$ is bounded (see (10)). □

Remark 1. (11) is the gradient descent algorithm, which the normalizing learning rate η_k is time-varying in order to assure the identification process is stable. This learning law is simpler to use, because we do not need to care about how to select a better learning rate to assure both fast convergence and stability. No previous information is required.

Remark 2. The condition $\beta\|e(k+1)\|\geq\|e(k)\|$ is a dead-zone. If β is selected big enough, the dead-zone becomes small.

When the identified plant is regarded as black-box, we cannot construct fuzzy rules for the premise part $\mu_{A_{ji}}$ and $\mu_{D_{ji}}$ (see [17, 19, 21]). The object of the fuzzy neural modeling is to find the center values W_1 of $B_{1i}\cdots B_{mi}$ and W_2 of $C_{1i}\cdots C_{mi}$, as well as the membership functions $A_{1i}\cdots A_{ni}$ and $D_{1i}\cdots C_{ni}$, such that the fuzzy neural networks (5) can follow the nonlinear plant (1). In order to simplify the calculation, we only discuss the case of $U=0$, so D and C are not exist. In this paper, Gaussian membership functions are exploited to identify fuzzy rules, which are defined by

$$\mu_{A_{ji}}=\exp\left(-\frac{(x_j-c_{ji})^2}{\rho_{ji}^2}\right) \tag{18}$$

The fuzzy neural model can be expressed as

$$\beta\widehat{x}(k+1)=A\widehat{x}(k)+W_1(k)\sigma_1[x(k)] \tag{19}$$

Let us define

$$z_i=\prod_{j=1}^{n}\exp\left(-\frac{(x_j-c_{ji})^2}{\rho_{ji}^2}\right)$$
$$a_q=\alpha\sum_{i=1}^{l}w_{1qi}z_i,\quad b=\sum_{i=1}^{l}z_i \tag{20}$$

So

$$\widehat{x}_q=\frac{a_q}{b} \tag{21}$$

The identified nonlinear process (1) can be represented as

$$\widehat{x}_q = \frac{\sum_{i=1}^{l} w_{qi}^* \prod_{j=1}^{n} \exp\left(-(x_j - c_{ji}^*)^2/\rho_{ji}^{*2}\right)}{\left[\sum_{i=1}^{l} \prod_{j=1}^{n} \exp\left(-(x_j - c_{ji}^*)^2/\rho_{ji}^{*2}\right)\right]} - \mu_q \tag{22}$$

where w_{qi}^*, c_{ji}^* and ρ_{ji}^* 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

$$f(x_1, x_2, x_3) = \sum_{k=0}^{l-1} \frac{1}{k!}[(x_1 - x_1^0)\frac{\partial}{\partial x_1} + (x_2 - x_2^0)\frac{\partial}{\partial x_2} + (x_3 - x_3^0)\frac{\partial}{\partial x_3}]_0^k f + R_l \tag{23}$$

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

$$\begin{aligned} x_q + \mu_q = \widehat{x}_q + \sum_{i=1}^{l}(w_{qi}^* - w_{qi})z_i \Big/ b + \sum_{i=1}^{l}\sum_{j=1}^{n} \frac{\partial}{\partial c_{ji}}\left(\frac{a_q}{b}\right)(c_{ji}^* - c_{ji}) \\ + \sum_{i=1}^{l}\sum_{j=1}^{n} \frac{\partial}{\partial \rho_{ji}}\left(\frac{a_q}{b}\right)(\rho_{ji}^* - \rho_{ji}) + R_{1q} \end{aligned} \tag{24}$$

Using the chain rule, we get

$$\begin{aligned} \frac{\partial}{\partial C_{ji}}\left(\frac{a_q}{b}\right) &= \frac{\partial}{\partial z_i}\left(\frac{a_q}{b}\right)\frac{\partial z_i}{\partial C_{ji}} \\ &= \left(\frac{1}{b}\frac{\partial a_q}{\partial z_i} + \frac{\partial}{\partial z_i}\left(\frac{1}{b}\right)a_q\right)\left(2z_i\frac{x_j - c_{ji}}{\sigma_{ji}^2}\right) \\ &= \left(\frac{w_{qi}}{b} - \frac{a_q}{b^2}\right)\left(2z_i\frac{x_j - c_{ji}}{\sigma_{ji}^2}\right) = 2z_i\frac{w_{qi} - \widehat{y}_q}{b}\frac{x_j - c_{ji}}{\sigma_{ji}^2} \\ \frac{\partial}{\partial \rho_{ji}}\left(\frac{a_q}{b}\right) &= \frac{\partial}{\partial z_i}\left(\frac{a_q}{b}\right)\frac{\partial z_i}{\partial \rho_{ji}} \\ &= 2z_i\frac{w_{qi} - \widehat{x}_q(k+1)}{b}\frac{(x_j - c_{ji})^2}{\rho_{ji}^3} \end{aligned} \tag{25}$$

In matrix form,

$$x_q(k+1) + \mu_q = \widehat{x}_q(k+1) - \widetilde{W}_q Z(k) - D_{zq}\overline{C}_k E - D_{zq}\overline{B}_k E \tag{26}$$

Let us define the identification error as is defined as

$$\begin{aligned} e_q &= \widehat{x}_q(k+1) - x_q(k+1) \\ e_q &= Z(k)\widehat{W}_q + D_{zq}\overline{C}_k E + D_{zq}\overline{B}_k E + \mu_q - R_{1q} \end{aligned} \tag{27}$$

where R_{1q} is the second-order approximation error of the Taylor series, $q = 1 \cdots m$. We define $e(k) = [e_1 \cdots e_m]^{\mathrm{T}}$, so

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

By the boundness of the Gaussian function ϕ we may assume that μ in (22) is bounded, also R_1 is bounded. So $\zeta(k)$ in (28) is bounded. The following theorem gives a stable backpropagation algorithm for discrete-time FNNs.

THEOREM 2. *If we use the fuzzy neural network (19) to identify nonlinear plant (1), the eigenvalues of A are selected as* $-1<\lambda(A)<0$, *and the following backpropagation algorithm can make identification error* $e(k)$ *bounded*

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

where if $\beta\|e(k+1)\| \geq \|e(k), \eta(k) = \frac{\eta}{1+\|Z\|^2+2\|D_z\|^2}$; *if* $\beta\|e(k+1)\| < \|e(k)\|, \eta(k) = 0.0 < \eta \leq 1$. *The average of the identification error satisfies*

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

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

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

$$[\widetilde{C}_{k+1}] = [\widetilde{C}_k] - 2\eta_k z_i \frac{w_{qi}-\widehat{y}_q}{b} \frac{x_j - c_{ji}}{\rho_{ji}^2} (\widehat{x}_q - x_q) \tag{31}$$

We selected a positive defined matrix L_k as

$$L_k = \|\widetilde{W}_k\|^2 + \|\widetilde{C}_k\|^2 + \|\widetilde{B}_k\|^2 \tag{32}$$

Similar as the proof of Theorem 1, by using (28) we have

$$\begin{aligned} \Delta L_k = & \|\widetilde{W}_k - \eta_k e(k) Z(k)^{\mathrm{T}}\|^2 \\ & + \left\| \widetilde{C}_k - \left[2\eta_k z_i \frac{w_{qi}-\widehat{y}_q}{b} \frac{x_j - c_{ji}}{\rho_{ji}^2} (\widehat{x}_q - x_q) \right] \right\|^2 \\ & + \left\| \widetilde{B}_k - \left[2\eta_k z_i \frac{w_{qi}-\widehat{y}_q}{b} \frac{(x_j - c_{ji})^2}{\rho_{ji}^2} (\widehat{x}_q - x_q) \right] \right\|^2 \\ & - \|\widetilde{W}_k\|^2 - \|\widetilde{C}_k\|^2 - \|\widetilde{B}_k\|^2 \end{aligned}$$

$$
\begin{aligned}
&= \eta_k^2 e^2(k)\left(\|Z(k)^{\mathrm{T}}\|^2 + 2\|D_z^{\mathrm{T}}\|^2\right) \\
&\quad -2\eta_k\|e(k)\|\left\|\widetilde{W}_k Z(k)^{\mathrm{T}} + D_z^{\mathrm{T}}\overline{C}_k E + D_z^{\mathrm{T}}\overline{B}_k E\right\| \\
&= \eta_k^2 e^2(k)(\|Z\|^2 + 2\|D_z\|^2) - 2\eta_k\|e(k)[e(k) - \zeta(k)]\| \\
&\leq -\eta_k e^2(k)[1 - \eta_k(\|Z\|^2 + 2\|D_z\|^2)] + \eta\zeta^2(k) \\
&\leq -\pi e^2(k) + \eta\zeta^2(k)
\end{aligned}
\tag{33}
$$

where π is defined as

$$
\pi = \frac{\eta}{1 + \max_k(\|Z\|^2 + 2\|D_z\|^2)} \tag{34}
$$

Because

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

where $n\left[\min(\widetilde{w}_i^2) + \min(\widetilde{c}_{ji}^2) + \min(\widetilde{b}_{ji}^2)\right]$ and $n\left[\max(\widetilde{w}_i^2) + \max(\widetilde{c}_{ji}^2) + \max(\widetilde{b}_{ji}^2)\right]$ are $\mathcal{K}_\infty$-functions, and $\pi e^2(k)$ is an $\mathcal{K}_\infty$-function, $\eta\zeta^2$ is a $\mathcal{K}$-function. From (28) and (32), we know V_k is the function of $e(k)$ and $\zeta(k)$, so L_k admits a smooth ISS–Lyapunov function as in Definition 2. From Theorem 1, the dynamic of the identification error is input-to-state stable. Because the "INPUT" $\zeta(k)$ is bounded and the ISS is dynamic, the "STATE" $e(k)$ is bounded.

(33) can be rewritten as

$$
\Delta L_k \leq -\pi e^2(k) + T\eta\zeta^2(k) \leq \pi e^2(k) + \eta\bar{\zeta} \tag{35}
$$

Summarizing (35) from 1 up to T, and by using $L_{\mathrm{T}} > 0$ and L_1 is a constant, we obtain

$$
L_{\mathrm{T}} - L_1 \leq -\pi\sum_{k=1}^{\mathrm{T}} e^2(k) + T\eta\bar{\zeta}\pi\sum_{k=1}^{\mathrm{T}} e^2(k) \leq L_1 - L_{\mathrm{T}} + \mathrm{T}\eta\bar{\zeta} \leq L_1 + \mathrm{T}\eta\bar{\zeta} \tag{36}
$$

(30) is established. □

4. Simulation

In this section, one typical chaotic system, Lorenz model, is chosen to demonstrate the abilities of the RFNNs. Lorenz model is used for the fluid conviction description especially for some feature of atmospheric dynamic. The uncontrolled model is given by

$$
\begin{aligned}
\dot{x}_1 &= -\beta x_1 + x_2 x_3 \\
\dot{x}_2 &= \omega(x_3 - x_2) \\
\dot{x}_3 &= -x_1 x_2 + \rho x_2 - x_3
\end{aligned}
\tag{37}
$$

with $x_0 = [50, 10, 20]^T$. We use following difference technique to get the discrete-time states of the system (37). We define $s_1 = Ax_k, s_2 = A(x_k + s_1), s_3 = A(x_k + \frac{s_1+s_2}{4})$. If $|\frac{s_1-2*s_3+s_2}{3}| \le \frac{|x_k|}{1000}$ or $|\frac{s_1-2*s_3+s_2}{3}| < 1$, then $x_{k+1} = x_k + \frac{s_1+4s_3+s_2}{6}$, $k = 0, 1, 2, \cdots$. First we use FNN identification without premise membership functions learning

$$\beta\widehat{x}(k+1) = A\widehat{x}(k) + W_1(k)\sigma_1[x(k)] \tag{38}$$

where $\beta = 4, \sigma_1(\cdot) = \tanh(\cdot), \widehat{x} = [\widehat{x}_1, \widehat{x}_2, \widehat{x}_3]^T, A = \text{diag}\,[.8, .8, .8], W_1(k) \in \Re^{3\times 3}$. The elements of $W_1(0)$ are random number between [0, 1]. We use (11) to update the weight

$$W_1(k+1) = W_1(k) - \frac{1}{1+\|\sigma\|^2} e(k)\sigma^T \tag{39}$$

Figure 3 shows the on-line identification results.

Next, we use FNN with premise membership functions learning. The on-line identification result is shown in Figure 4. Model complexity is important in the context of system identification, which is corresponded to the number of rules. In this simulation we try to test different rule number, we find that after rule number

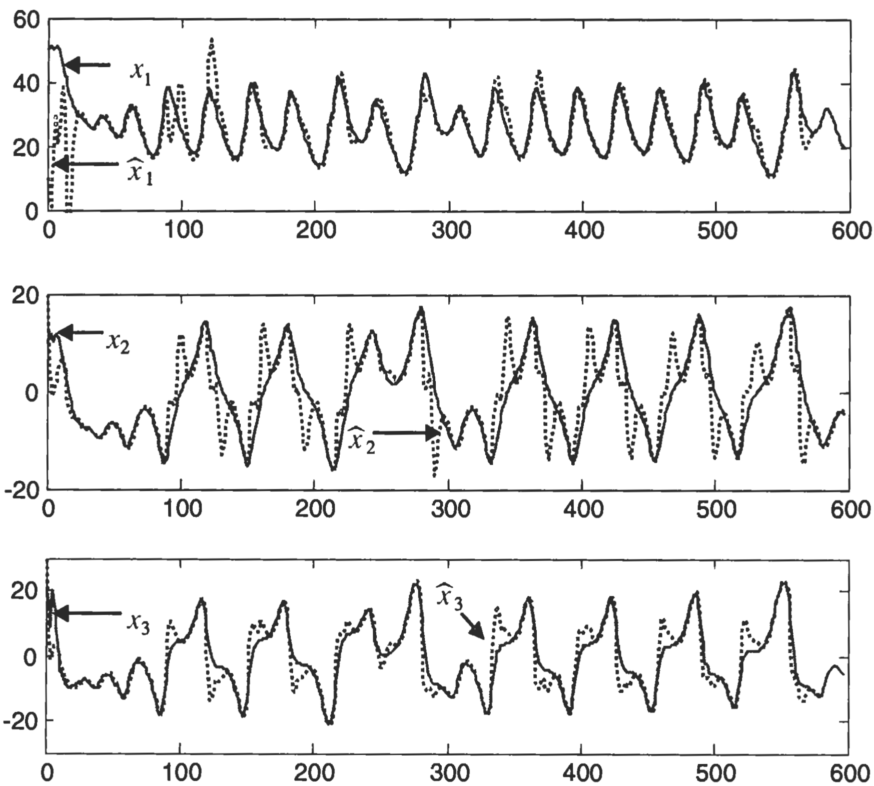

Figure 3. FNN identification without premise membership learning.

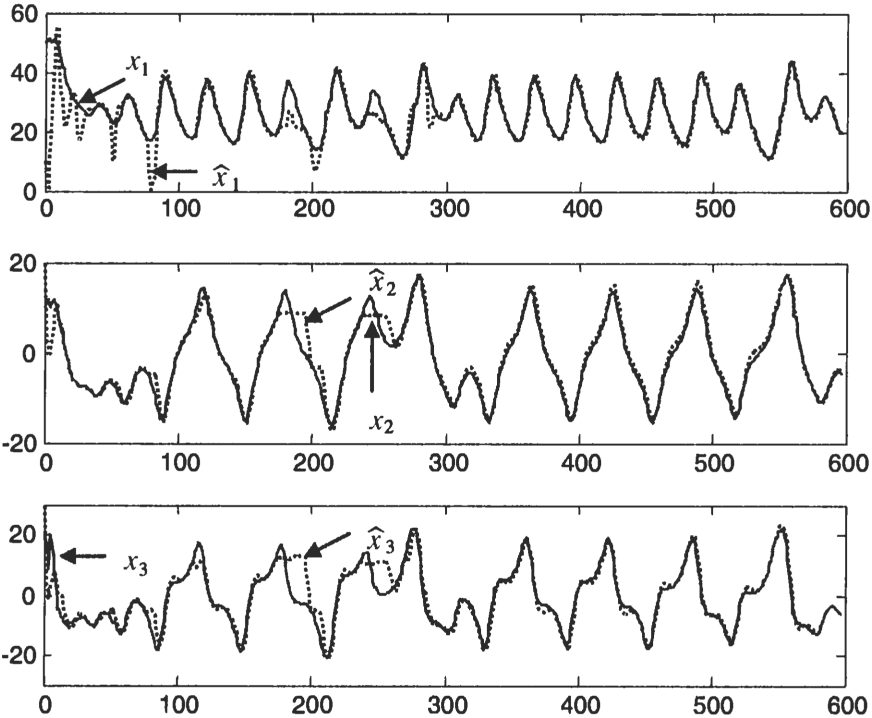

Figure 4. FNN identification with premise membership learning.

l is more than 20, the identification accuracy will not be improved a lot. The total simulation time is 600 there is 1 time for $\beta \| e(k+1) \| < \| e(k) \|$. Theorem 1 gives a necessary condition of η for stable learning, $\eta \leq 1$. In this example, we found that if $\eta \geq 2.5$, the learning process becomes unstable. The total simulation time is 600, there are 14 times for $\beta \| e(k+1) \| < \| e(k) \|$.

5. Conclusion

In this paper we propose a new FNNs – SRFNNs. The main contribution is that by using ISS approach stable updating laws for the membership functions of the fuzzy neural networks are proposed.

References

1. Chen, D. S. and Jain, R. C.: A robust back propagation learning algorithm for function approximation, *IEEE Trans. Neural Networks* **5**(3) (1994).
2. Chen, M. Y. and Linkensm, D. A.: A systematic neuro-fuzzy modeling framework with application to material property prediction, *IEEE Trans. Syst. Man Cybern. B* **31** (2001), 781–790.
3. Egardt, B.: *Stability of Adaptive Controllers, Lecture Notes in Control and Information Sciences*, Vol.20, Springer, Berlin, 1979.

4. Haykin, S.: *Neural Networks – A Comprehensive Foundation*, Macmillan New York, 1994.
5. Ioannou, P. A. and Sun, J.: *Robust Adaptive Control*, Upper Saddle River, NJ, Prentice-Hall, 1996.
6. Jiang, Z. P. and Wang, Y.: Input-to-State Stability for discrete-time nonlinear systems, *Automatica* **37**(2) (2001), 857–869.
7. Jin, L. and Gupta, M. M.: Stable dynamic backpropagation learning in recurrent neural networks, *IEEE Trans. Neural Networks* **10**(6) (1999), 1321–1334.
8. Juang, C. F.: A TSK-type recurrent fuzzy networks for dynamic systems processing by neural network and genetic algorithms, *IEEE Trans. Fuzzy Syst.* **10**(2) (2002), 155–170.
9. Lee, C. H. and Teng, C. C.: Identification and control of dynamic system using recurrent fuzzy neural networks, *IEEE Trans. Fuzzy Syst.* **8**(4) (2000), 349–366.
10. Lewis, F. L. Yesildirek, A. and Liu, K.: Multilayer neural-net robot controller with guaranteed tracking performance, *IEEE Trans. Neural Networks* **7**(2) (1996), 388–399.
11. Leu, Y. G. Lee, T. T. and Wang, W. Y.: Observer-based adaptive fuzzy-neural control for unknown nonlinear dynamical systems, *IEEE Trans. Syst. Man Cybern. B* **29** (1999), 583–591.
12. Lin, C. T. and Lee, G.: Neural fuzzy systems: A neural-fuzzy synergism to intelligent systems, Prentice-Hall Inc., NJ, 1996.
13. Lin, C. T.: A neual fuzzy control system with structure and parameter learning, *Fuzzy Sets anc Syst.* **70** (1995), 183–212.
14. Mamdani, E. H.: Application of fuzzy algorithms for control of simple dynamic plant, *IEE Proc. Control Theory Appl.* **121**(12) (1976), 1585–1588.
15. Mastorocostas, P. A. and Theocharis, J. B.: A recurrent fuzzy-neural model for dynamic system identification, *IEEE Trans. Syst. Man Cybern. B* **32**(2) (2002), 176–190.
16. Takagi, T. and Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control, *IEEE Trans. Syst. Man Cybern.* **15** (1985), 116–132.
17. Tsai, H. H. and Yu, P. T.: On the optimal design of fuzzy neural networks with robust learning for function approximation, *IEEE Trans. Syst. Man Cybern. B* **30** (2000), 217–223.
18. Wu, S. and Er, M. J.: Dynamic fuzzy neural networks- a novel approach to function approximation, *IEEE Trans. Syst. Man Cybern. B* **30** (2000), 358–364.
19. Wang, C. H. Liu, H. L. and Lin, C. T.: Dynamic optimal learning rates of a certain class of fuzzy neural networks and its applications with genetic algorithm, *IEEE Trans. Syst. Man Cybern. B* **31** (2001), 467–475.
20. Wang, L. X.: *Adaptive Fuzzy Systems and Control*, Englewood Cliffs, NJ, Prentice-Hall, 1994.
21. Wang, W. Y. Leu, Y. G. and Hsu, C. C.: Robust adaptive fuzzy-neural control of nonlinear dynamical systems using generalized projection updated law and variable structure controller, *IEEE Trans. Syst. Man Cybern. B* **31** (2001), 140–147.
22. Yu, W. and Li, X.: Some stability properties of dynamic neural networks, *IEEE Trans. Circuits Syst.* **48**(1) (2001), 256–259.
23. Yu, W. and Li, X.: Some new results on system identification with dynamic neural networks, *IEEE Trans. Neural Networks* **12**(2) (2001), 412–417.
24. Zhang, J. and Morris, A.J.: Recurrent neuro-fuzzy networks for nonlinear process modeling, *IEEE Trans. Neural Networks* **10**(2) (1999), 313–326.