Proceedings of the International MultiConference of Engineers and Computer Scientists 2011 Vol I,
IMECS 2011, March 16 - 18, 2011, Hong Kong

Stable Learning Mechanism for Novel Takagi-Sugeno-Kang Type Interval-valued Fuzzy Systems

Yi-Han Lee and Ching-Hung Lee, *Member IAENG*

Abstract—**In this paper, we propose a novel Takagi-Sugeno-Kang type interval-valued neural fuzzy system with asymmetric fuzzy membership functions (called TIVNFS-A). In addition, the corresponding type reduction procedure is integrated in the adaptive network layers to reduce the amount of computation in the system. Based on the Lyapunov stability theorem, the TIVNFS-A system is trained by the back-propagation (BP) algorithm having an optimal learning rate (adaptive learning rate) to guarantee the stability and faster convergence. Finally, the TIVNFS-A with the optimal BP algorithm is applied in nonlinear system identification to demonstrate the effectiveness and performance.**

Index Terms—**interval-valued fuzzy system, Lyapunov stability theorem, asymmetric membership function, TSK type, nonlinear system**

I. Introduction

In recent years, fuzzy neural networks (FNNs) are used successfully in many applications [5, 7-9, 10, 13-21, 25, 27-28], such as classifications, prediction, nonlinear system and control. FNNs are combined the advantages of fuzzy system and neural network. Recently, interval type-2 fuzzy logic systems (T2FLSs) have got lots of attention in many applications due to their ability to model uncertainties. Besides, many literatures have shown that an interval T2FLS is the same as an interval-valued fuzzy logic system (IVFLS) [1-2, 23, 26]. IVFLSs are more complex than type-1 fuzzy logic systems (T1FLSs). IVFLSs have better performance than T1FLSs on the applications of function approximation, system modeling and control. Combining the advantages of IVFLSs and neural network, interval-valued fuzzy neural network (IVFNN) systems are presented to handle the system uncertainty [6, 16, 18].

By designing the fuzzy partition and rule engine, symmetric and fixed membership functions (MFs), like Gaussian or triangular, are usually used to simplify the design procedure. Accordingly, a large number of rules should be used to accomplish the explicit approximation accuracy [5, 15, 17]. In order to solve these problems, the asymmetric fuzzy MFs (AFMFs) have been adopted [3, 11, 17-20, 24]. These results demonstrated that using AFMFs can improve the modeling capability. The asymmetric fuzzy MFs provide the ability of high flexibility and more accurate [18]. In addition, the Takagi-Sugeno-Kang (TSK) type FLSs have the universal approximation capability [4]. By the combination of these above advantages, the TSK-type interval-valued neural fuzzy system with asymmetric membership functions (TIVNFS-A) is proposed in this paper. In addition, the corresponding type reduction procedure is integrated in the adaptive network layers to reduce the amount of computation in the system. Therefore, the Karnik-Mendel type-reduction procedure is removed.

For training and designing the neural fuzzy systems, the back-propagation algorithm is widely used and a powerful training technique [5, 8, 25]. For each training cycle, all parameters of neural fuzzy system are adjusted to reduce the error between the desired and actual output. Herein, based on the Lyapunov stability theorem, the TIVNFS-A system is trained by the back-propagation (BP) algorithm having an optimal learning rate (adaptive learning rate) to guarantee the stability and faster convergence.

The rest of this paper is as follows. Section II introduces the construction of AFMFs and TIVNFSF-A. The back-propagation algorithm with the optimal learning rate is introduced in Section III. Section IV shows the simulation results of nonlinear system identification using TIVNFS-A with optimal BP algorithm. Finally, the conclusion is given.

II. Takagi-Sugeno-Kang-Type Interval-Valued Neural Fuzzy Systems

In this paper, we propose a TSK-type interval-valued neural fuzzy system with asymmetric membership functions (TIVNFS-A), which is a modification of type-2 fuzzy neural networks (or interval-valued neural fuzzy systems). We adopted interval-valued asymmetric MFs and the TSK-type consequent part to develop the TIVNFS-A. We first introduce the network structure of TIVNFS-A. In general, given the system input data x_i, $i = 1, 2, \cdots, n$, and the TIVNFS-A's output $\hat{y}$, the jth fuzzy rule can be expressed as:

Rule j: IF x_1 is $\tilde{F}_{1j}$ and … and x_n is $\tilde{F}_{nj}$, THEN

$$Y_j = C_{j0} + C_{j1}x_1 + C_{j2}x_2 + \ldots + C_{jn}x_n, \quad (1)$$

where j=1, 2, …, M; C_{j0} and C_{ji} are the consequent fuzzy sets, Y_j is the output of the jth rule (a linear combination operation), and $\tilde{F}_{ij}$ is the antecedent fuzzy set. The fuzzy MFs of the antecedent part $\tilde{F}_{ij}$ are the asymmetric interval-valued fuzzy

This work was supported in part by the National Science Council, Taiwan, R.O.C., under contracts NSC-97-2221-E-155-033-MY3.
Ching-Hung Lee is with Department of Electrical Engineering, Yuan-Ze University, Chung-li, Taoyuan 320, Taiwan. (phone: +886-3-4638800, ext: 7119; fax: +886-3-4639355, e-mail: chlee@saturn.yzu.edu.tw).

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

sets (IVFSs) that are shown in Fig. 1, which are different from typical Gaussian MFs. Several approaches indicate that using asymmetric MFs can improve approximation accuracy [3, 12, 18, 19-21, 25]. The asymmetric IVFSs are generated from four Gaussian functions, as shown in Fig. 1. The construction of asymmetric IVFSs was introduced in literature [18]. Subsequently, the fuzzy sets of the consequent part C_{j0} and C_{ji} are designed to be convex, normal type-1 fuzzy subsets. A TIVNFS-A with M fuzzy rules is implemented as the six-layer network shown in Fig. 2. The signal propagation and the operation functions of the nodes are indicated in each layer. In the following description, $O_i^{(k)}$ denotes the ith output of a node in the kth layer.

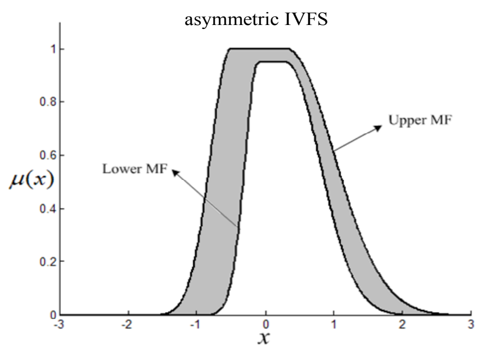

Figure 1: Constructed asymmetric interval valued fuzzy set (IVFS) [18].

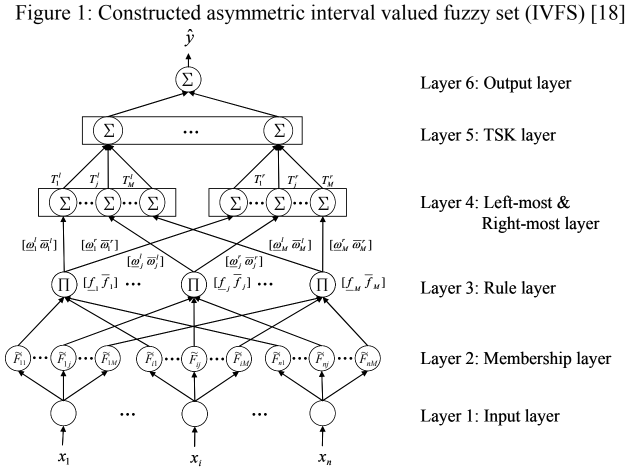

Figure 2: Diagram of the proposed TIVNFS-A with M rules.

Layer 1 (Input Layer): For the ith node of layer 1, the net input and the net output are written as

$$O_i^{(1)} = x_i \tag{2}$$

where i=1, 2, …, n, x_i represents the ith input to the ith node of layer 1. The nodes in this layer only transmit input values to the next layer directly.

Layer 2 (Membership Layer): In this layer, each node performs an asymmetric IVFS $\widetilde{F}_{ij}$, as shown in Fig. 1, i.e.,

$$O_{ij}^{(2)} = \mu_{\widetilde{F}_{ij}}(O_i^{(1)}) = [\underline{O}_{ij}^{(2)} \quad \overline{O}_{ij}^{(2)}]^T = [\underline{\mu}_{\widetilde{F}_{ij}}(O_i^{(1)}) \quad \overline{\mu}_{\widetilde{F}_{ij}}(O_i^{(1)})]^T \tag{3}$$

where the subscript "$_{ij}$" indicates the jth term of the ith input, where j=1, …, M.

Layer 3 (Rule Layer): The links in this layer are used to implement antecedent matching. We choose the product t-norm because it is easy to implement in a neural network. Thus, the firing strength associated with the jth rule is the following:

$$\underline{f}_j = \underline{\mu}_{\widetilde{F}_{1j}}(O_1^{(1)}) * \cdots * \underline{\mu}_{\widetilde{F}_{nj}}(O_n^{(1)}) \tag{4}$$

$$\overline{f}_j = \overline{\mu}_{\widetilde{F}_{1j}}(O_1^{(1)}) * \cdots * \overline{\mu}_{\widetilde{F}_{nj}}(O_n^{(1)}) \tag{5}$$

where $\underline{\mu}_{\widetilde{F}_{ij}}(\cdot)$ and $\overline{\mu}_{\widetilde{F}_{ij}}(\cdot)$ are the lower and upper membership grades of $\mu_{\widetilde{F}}(\cdot)$, respectively. Therefore, a simple product operation is used. Then,

$$O_j^{(3)} = \left[\underline{O}_j^{(3)} \quad \overline{O}_j^{(3)}\right]^T = \left[\prod_{i=1}^{n} \underline{O}_{ij}^{(2)} \quad \prod_{i=1}^{n} \overline{O}_{ij}^{(2)}\right]^T. \tag{6}$$

Layer 4 (Left-most and right-most layers): The weighting vectors of TIVNFS-A are interval-valued $\left[\underline{\omega}_j^l \quad \overline{\omega}_j^l\right]^T$ and $\left[\underline{\omega}_j^r \quad \overline{\omega}_j^r\right]^T$, where $\underline{\omega}_j^l < \overline{\omega}_j^l$ and $\underline{\omega}_j^r < \overline{\omega}_j^r$. The following vector notations are used for clarity: $\underline{\omega}^l = [\underline{\omega}_1^l \cdots \underline{\omega}_M^l]^T$, $\overline{\omega}^l = \left[\overline{\omega}_1^l \quad \cdots \quad \overline{\omega}_M^l\right]^T$, $\underline{\omega}^r = [\underline{\omega}_1^r \cdots \underline{\omega}_M^r]^T$, and $\overline{\omega}^r = \left[\overline{\omega}_1^r \quad \cdots \quad \overline{\omega}_M^r\right]^T$. Therefore, the output for layer 4 is

$$O_j^{(4)} = \left[O_{jl}^{(4)} \ O_{jr}^{(4)}\right]^T = \left[\frac{\overline{\omega}_j^l \overline{O}_j^{(3)} + \underline{\omega}_j^l \underline{O}_j^{(3)}}{\overline{\omega}_j^l + \underline{\omega}_j^l} \quad \frac{\overline{\omega}_j^r \overline{O}_j^{(3)} + \underline{\omega}_j^r \underline{O}_j^{(3)}}{\overline{\omega}_j^r + \underline{\omega}_j^r}\right]^T \tag{7}$$

This expression calculates the left-most points, $O_{jl}^{(4)}$, and right-most points, $O_{jr}^{(4)}$. According to previously reported results [6], the type reduction is integrated into the adaptive network layers. Therefore, the Karnik-Mendel type-reduction procedure is removed. Because the iterative procedure for finding coefficients R and L is not necessary, the computational effort can be reduced effectively.

Layer 5 (TSK Layer): Because the asymmetric IVFSs are used for the antecedents and because the interval sets are used for the consequent sets of the TSK rules, it is possible to state that the C_{ji} terms are interval sets. In other words, $C_{ji}=[c_{ji}-s_{ji} \ c_{ji}+ s_{ji}]^T$, where i=1,…, n, and j=1,…, M. In this expression, c_{ji} denotes the center (mean) of C_{ji}, and s_{ji} denotes the spread of C_{ji}, i=1, 2, …, n and j=1, 2, …, M. Therefore, the consequent part of *Rule j* is

$$T_j = [(c_{j0} + \sum_{i=1}^{n} c_{ji} x_i) - (s_{j0} + \sum_{i=1}^{n} s_{ji} |x_i|) \quad (c_{j0} + \sum_{i=1}^{n} c_{ji} x_i) + (s_{j0} + \sum_{i=1}^{n} s_{ji} |x_i|)]^T \tag{8}$$

where $s_{ji} \geq 0$. The output of layer 5 is

$$O^{(5)} = \left[O_l^{(5)} \ O_r^{(5)}\right]^T = \left[\frac{\sum_{j=1}^{M} O_{jl}^{(4)} T_j^l}{\sum_{j=1}^{M} O_{jl}^{(4)}} \quad \frac{\sum_{j=1}^{M} O_{jr}^{(4)} T_j^r}{\sum_{j=1}^{M} O_{jr}^{(4)}}\right]^T. \tag{9}$$

Layer 6 (Output Layer): Layer 6 is the output layer, which is used to implement the defuzzification operation. The output is the following:

$$O^{(6)} = \frac{O_l^{(5)} + O_r^{(5)}}{2}. \tag{10}$$

As the above introduction, the interval-valued fuzzy sets are used to design the antecedents and interval type-1 fuzzy sets are used to design the consequent sets of an interval-valued TSK rule. Directing our attention to (9), we can see that $O_j^{(4)}$ and T_j are interval type-1 fuzzy sets. Hence, $O_{TSK}^{(5)}$ is an interval type-1 fuzzy set. We only need to compute its two end-points $O_l^{(5)}$ and $O_r^{(5)}$ to compute $O_{TSK}^{(5)}$. Hence, we use the weights between layer 3 and layer 4 to regard as the firing interval which is determined by its left-most and

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

right-most points. Without operating the iterative procedure of KM algorithm described in [12] for finding coefficients R and L. Therefore, the TIVNFS-A can reduce the computational complexity successfully.

III. Learning of TIVNFS-A Systems

In this paper, we adjust the parameter of TIVNFS-A by the back-propagation (BP) algorithm to enhance performance [5, 8, 15, 25]. The back-propagation algorithm is based on the gradient descent method to fine the optimal solution of each parameter.

A. Back-propagation Algorithm

For clarification, we consider the signal-output system and define the error cost function

$$E(k)=\frac{1}{2}e(k)^2, \tag{11}$$

where $e(k)=y_d(k)-\hat{y}(k)=y_d(k)-O^{(6)}(k)$, that $\hat{y}(k)$ and $y_d(k)$ are the TIVNFS-A's output and desired output for discrete time k, respectively. Using the gradient descent method, the parameters updated law of the parameters is

$$\mathbf{W}(k+1)=\mathbf{W}(k)+\eta\left(-\frac{\partial E(k)}{\partial \mathbf{W}}\right), \tag{12}$$

where η is the learning rate. $\mathbf{W}=\left[\underline{\mathbf{W}},\overline{\mathbf{W}},\gamma,\mathbf{W}_\omega,\mathbf{C}\right]^T$ are the adjustable parameters, where $\mathbf{C}$ is the parameters of TSK layer, $\mathbf{W}_\omega$ is the consequent weights, $\underline{\mathbf{W}}$ is the parameters of lower MFs, $\overline{\mathbf{W}}$ is upper MFs parameters, and γ is the column vectors, i.e.,

$$\begin{aligned}
\mathbf{C}&=\left[c \quad s\right]^T,\\
\mathbf{W}_\omega&=\left[\underline{\omega}^l \quad \underline{\omega}^r \quad \overline{\omega}^l \quad \overline{\omega}^r\right]^T,\\
\underline{\mathbf{W}}&=\left[\underline{m}^l \quad \underline{m}^r \quad \underline{\sigma}^l \quad \underline{\sigma}^r\right]^T,\\
\overline{\mathbf{W}}&=\left[\overline{m}^l \quad \overline{m}^r \quad \overline{\sigma}^l \quad \overline{\sigma}^r\right]^T.
\end{aligned} \tag{13}$$

Considering $\partial E(k)/\partial \mathbf{W}$, we have

$$\frac{\partial E(k)}{\partial \mathbf{W}}=\frac{\partial E(k)}{\partial e(k)}\frac{\partial e(k)}{\partial \hat{y}(k)}\frac{\partial \hat{y}(k)}{\partial \mathbf{W}}=-e(k)\frac{\partial \hat{y}(k)}{\partial \mathbf{W}}, \tag{14}$$

thus,

$$\begin{aligned}
\mathbf{W}(k+1)&=\mathbf{W}(k)+\eta\cdot e(k)\cdot\frac{\partial \hat{y}(k)}{\partial \mathbf{W}}\\
&=\mathbf{W}(k)+\eta\cdot e(k)\cdot\frac{\partial O^{(6)}(k)}{\partial \mathbf{W}},
\end{aligned} \tag{15}$$

where $e(k)=y_d(k)-\hat{y}(k)$. The remaining work involves finding the corresponding partial derivative with respect to each parameter. For the BP algorithm, the remaining works are the derivations of gradient for parameters $\underline{\mathbf{W}},\overline{\mathbf{W}},\gamma,\mathbf{W}_\omega$, and $\mathbf{C}$. The derivations are omitted due to the writing space. Similar details can be found in literature [18].

B. Optimal Learning Rate

The learning rate plays an important role in BP algorithm. A small value of learning rate leads the speed of convergence will be slower. The large value of learning rate leads the speed of convergence is faster but it might produce local minimum. Hence, the selection of the learning rate is importantly but it is not easy to choosing suitably. Thus, we use the Lyapunove function to find the optimal learning rate [5, 15, 29]. At first, we defined the positive Lyapunov candidate

$$V(k)=E(k)=\frac{1}{2}e^2(k). \tag{16}$$

In general, $e(k+1)-e(k)=\Delta e(k)$. Thus, we have

$$\begin{aligned}
\Delta V(k)&=V(k+1)-V(k)\\
&=\frac{1}{2}\left[(e(k+1)-e(k))\cdot(e(k+1)+e(k))\right]\\
&=\frac{1}{2}\left[\Delta e(k)\cdot\left(2e(k)+\Delta e(k)\right)\right]\\
&=\frac{1}{2}\left[\Delta e^2(k)+2e(k)\cdot\Delta e(k)\right],
\end{aligned} \tag{17}$$

where $\Delta e(k)\approx(\partial e/\partial \mathbf{W})\Delta\mathbf{W}$,thus, according to the parameter update rule of BP algorithm, $\Delta\mathbf{W}=-\eta(\partial E/\partial \mathbf{W})$. And we can obtain

$$\Delta e(k)\approx\frac{\partial e}{\partial \mathbf{W}}\left(-\eta\frac{\partial E}{\partial \mathbf{W}}\right)=-\eta\cdot e(k)\cdot\left(\frac{\partial \hat{y}}{\partial \mathbf{W}}\right). \tag{18}$$

Thus,

$$\begin{aligned}
\Delta V(k)&=\frac{1}{2}\left[-2\eta\cdot e^2\cdot\left(\frac{\partial \hat{y}}{\partial \mathbf{W}}\right)^2+\eta^2\cdot e^2\cdot\left(\frac{\partial \hat{y}}{\partial \mathbf{W}}\right)^4\right]\\
&=-\frac{1}{2}\eta\cdot e^2\cdot\left(\frac{\partial \hat{y}}{\partial \mathbf{W}}\right)^2\left[2-\eta\cdot\left(\frac{\partial \hat{y}}{\partial \mathbf{W}}\right)^2\right].
\end{aligned} \tag{19}$$

Next, according to the Lyapunov stability theory, we should choose a proper value of η such that $\Delta V(k)\le 0$. Therefore, we can obtain the stability condition for learning rate

$$0<\eta\left(\frac{\partial \hat{y}}{\partial \mathbf{W}}\right)^2<2. \tag{20}$$

Define $\lambda=\eta\cdot\left(\frac{\partial \hat{y}}{\partial \mathbf{W}}\right)^2$ and rewrite (19) as

$$\Delta V(k)=V(k)\cdot\lambda\cdot\left(-2+\lambda\right)\le 0. \tag{21}$$

Then we have

$$V(k+1)-V(k)=V(k)\cdot\lambda\cdot\left(-2+\lambda\right)\le 0 \tag{22}$$

and

$$V(k+1)=V(k)\cdot\left(1-2\lambda+\lambda^2\right)\le 0. \tag{23}$$

Thus, the optimal learning rate η can be obtained

$$\eta^*=\left(\frac{\partial \hat{y}}{\partial \mathbf{W}}\right)^{-2} \tag{24}$$

such that $\lambda=1$. Note that $\mathbf{W}\in\Re^D$, where D is the dimension of the problem. Therefore, the choice of optimal learning rate guaranteed the faster convergence is

$$\eta^*=\frac{1}{D}\left(\frac{\partial \hat{y}}{\partial \mathbf{W}}\right)^{-2}. \tag{25}$$

The optimal learning of TIVNFS-A's parameters are as follows.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

- Optimal learning of C

$$\eta_{c_{j0}}=\frac{1}{D}\left(\frac{\partial\hat{y}(k)}{\partial c_{j0}}\right)^{-2}=\frac{1}{D}\left[\frac{1}{2}\left(\frac{O_{jl}^{(4)}}{\sum_{j=1}^{M}O_{jl}^{(4)}}+\frac{O_{jr}^{(4)}}{\sum_{j=1}^{M}O_{jr}^{(4)}}\right)\right]^{-2}, \quad (26)$$

$$\eta_{c_{ji}}=\frac{1}{D}\left(\frac{\partial\hat{y}(k)}{\partial c_{ji}}\right)^{-2}=\frac{1}{D}\left[\frac{x_i}{2}\cdot\left(\frac{O_{jl}^{(4)}}{\sum_{j=1}^{M}O_{jl}^{(4)}}+\frac{O_{jr}^{(4)}}{\sum_{j=1}^{M}O_{jr}^{(4)}}\right)\right]^{-2}, \quad (27)$$

$$\eta_{s_{j0}}=\frac{1}{D}\left(\frac{\partial\hat{y}(k)}{\partial s_{j0}}\right)^{-2}=\frac{1}{D}\left[\frac{1}{2}\left(-\frac{O_{jl}^{(4)}}{\sum_{j=1}^{M}O_{jl}^{(4)}}+\frac{O_{jr}^{(4)}}{\sum_{j=1}^{M}O_{jr}^{(4)}}\right)\right]^{-2}, \quad (28)$$

$$\eta_{s_{ji}}=\frac{1}{D}\left(\frac{\partial\hat{y}(k)}{\partial s_{ji}}\right)^{-2}=\frac{1}{D}\left[\frac{|x_i|}{2}\cdot\left(\frac{O_{jl}^{(4)}}{\sum_{j=1}^{M}O_{jl}^{(4)}}+\frac{O_{jr}^{(4)}}{\sum_{j=1}^{M}O_{jr}^{(4)}}\right)\right]^{-2}. \quad (29)$$

- Optimal learning of $\mathbf{W}_\omega$

$$\eta_{\underline{\omega}_j^l}=\frac{1}{D}\left(\frac{\partial\hat{y}(k)}{\partial\underline{\omega}_j^l}\right)^{-2}=\frac{1}{D}\left[\frac{1}{2}\left(\frac{T_j^l-O_l^{(5)}}{\sum_{j=1}^{M}O_{jl}^{(4)}}\cdot\frac{\underline{O}_j^{(3)}-O_{jl}^{(4)}}{\underline{\omega}_j^l+\overline{\omega}_j^l}\right)\right]^{-2}, \quad (30)$$

$$\eta_{\underline{\omega}_j^r}=\frac{1}{D}\left(\frac{\partial\hat{y}(k)}{\partial\underline{\omega}_j^r}\right)^{-2}=\frac{1}{D}\left[\frac{1}{2}\left(\frac{T_j^r-O_r^{(5)}}{\sum_{j=1}^{M}O_{jr}^{(4)}}\cdot\frac{\underline{O}_j^{(3)}-O_{jr}^{(4)}}{\underline{\omega}_j^r+\overline{\omega}_j^r}\right)\right]^{-2}, \quad (31)$$

$$\eta_{\overline{\omega}_j^l}=\frac{1}{D}\left(\frac{\partial\hat{y}(k)}{\partial\overline{\omega}_j^l}\right)^{-2}=\frac{1}{D}\left[\frac{1}{2}\left(\frac{T_j^l-O_l^{(5)}}{\sum_{j=1}^{M}O_{jl}^{(4)}}\cdot\frac{\overline{O}_j^{(3)}-O_{jl}^{(4)}}{\underline{\omega}_j^l+\overline{\omega}_j^l}\right)\right]^{-2}, \quad (32)$$

$$\eta_{\overline{\omega}_j^r}=\frac{1}{D}\left(\frac{\partial\hat{y}(k)}{\partial\overline{\omega}_j^r}\right)^{-2}=\frac{1}{D}\left[\frac{1}{2}\left(\frac{T_j^r-O_r^{(5)}}{\sum_{j=1}^{M}O_{jr}^{(4)}}\cdot\frac{\overline{O}_j^{(3)}-O_{jr}^{(4)}}{\underline{\omega}_j^r+\overline{\omega}_j^r}\right)\right]^{-2}. \quad (33)$$

-Optimal learning of $\underline{\mathbf{W}}$

$$\eta_{\underline{\mathbf{W}}}=\frac{1}{D}\left(\frac{\partial\hat{y}(k)}{\partial\underline{\mathbf{W}}}\right)^{-2}=\left[-\frac{1}{4}\gamma\left(\frac{T_j^l-O_l^{(5)}}{\sum_{j=1}^{M}O_{jl}^{(4)}}\cdot\frac{\underline{\omega}_j^l\cdot\underline{O}_j^{(3)}}{\underline{\omega}_j^l+\overline{\omega}_j^l}+\frac{T_j^r-O_r^{(5)}}{\sum_{j=1}^{M}O_{jr}^{(4)}}\cdot\frac{\underline{\omega}_j^r\cdot\underline{O}_j^{(3)}}{\underline{\omega}_j^r+\overline{\omega}_j^r}\right)\cdot\frac{\partial}{\partial\underline{\mathbf{W}}}\left(\frac{x-\underline{m}}{\underline{\sigma}}\right)^2\right]^{-2}. \quad (34)$$

- Optimal learning of $\overline{\mathbf{W}}$

$$\eta_{\overline{\mathbf{W}}}=\frac{1}{D}\left(\frac{\partial\hat{y}(k)}{\partial\overline{\mathbf{W}}}\right)^{-2}=\left[-\frac{1}{4}\gamma\left(\frac{T_j^l-O_l^{(5)}}{\sum_{j=1}^{M}O_{jl}^{(4)}}\cdot\frac{\overline{\omega}_j^l\cdot\overline{O}_j^{(3)}}{\underline{\omega}_j^l+\overline{\omega}_j^l}+\frac{T_j^r-O_r^{(5)}}{\sum_{j=1}^{M}O_{jr}^{(4)}}\cdot\frac{\overline{\omega}_j^r\cdot\overline{O}_j^{(3)}}{\underline{\omega}_j^r+\overline{\omega}_j^r}\right)\cdot\frac{\partial}{\partial\overline{\mathbf{W}}}\left(\frac{x-\overline{m}}{\overline{\sigma}}\right)^2\right]^{-2}. \quad (35)$$

- Optimal learning of γ

$$\eta_{\gamma_{ij}}=\frac{1}{D}\left(\frac{\partial\hat{y}(k)}{\partial\gamma_{ij}}\right)^{-2}=\frac{1}{D}\left[\frac{1}{2}\left(\frac{T_j^l-O_l^{(5)}}{\sum_{j=1}^{M}O_{jl}^{(4)}}\cdot\frac{\omega_j^l}{\underline{\omega}_j^l+\overline{\omega}_j^l}+\frac{T_j^r-O_r^{(5)}}{\sum_{j=1}^{M}O_{jr}^{(4)}}\cdot\frac{\omega_j^r}{\underline{\omega}_j^r+\overline{\omega}_j^r}\right)\cdot\frac{O_j^{(3)}}{\gamma_{ij}}\right]^{-2}. \quad (36)$$

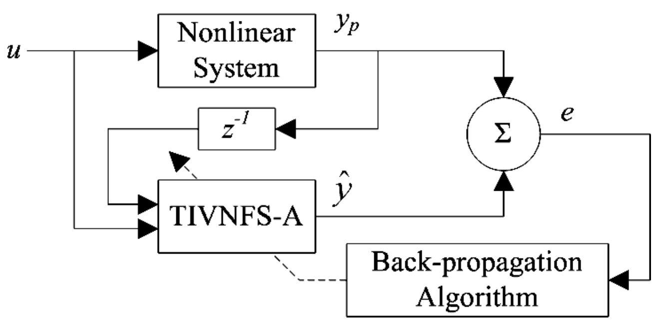

Fig. 3. Series-parallel model with the TIVNFS-A for nonlinear systems identification.

IV. Simulation Results

In this section, the example for nonlinear system identification is presented to show the performance of the TIVNFS-A. All simulations were done by MATLAB in Intel(R) CORE 2 QUAD computer with clock rate of 2.4GHz and 3GB of main memory.

Consider the following nonlinear system

$$y_p(k+1)=f\left(y_p(k),y_p(k-1),y_p(k-2),u(k),u(k-1)\right), \quad (37)$$

where

$$f\left(x_1,x_2,x_3,x_4,x_5\right)=\frac{x_1x_2x_3x_5(x_3-1)+x_4}{1+x_2^2+x_3^2}.$$

u and y_p are system's input and output. Herein, the series-parallel training scheme is adopted, as shown in Fig. 3. The approximated error is defined as follows

$$e(k)\equiv y_p(k)-\hat{y}(k), \quad (38)$$

where $\hat{y}(k)$ denotes the TIVNFS-A's output. Clearly, due to the static structure of TIVNFS-A, the input number should be set as 5. The training input is

$$u(k)=0.3\sin(\frac{\pi k}{25})+0.1\sin(\frac{\pi k}{32})+0.6\sin(\frac{\pi k}{10}), \quad (39)$$

and the testing input is

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

$$u(k)=\begin{cases}\sin(\frac{\pi k}{25}) & 0<k<250,\\ 1.0 & 250\le k<500,\\ -1.0 & 500\le k<750,\\ 0.3\sin(\frac{\pi k}{25})+0.1\sin(\frac{\pi k}{32}) & \\ \quad +0.6\sin(\frac{\pi k}{10}) & 750\le k<1000.\end{cases} \quad (40)$$

The following RMSE is adopted to be the performance index

$$RMSE:(\sum_k e^2(k)/N)^{1/2}\ , \quad (41)$$

where N is the number of training pattern.

The parameters of TIVNFS-A are $\overline{m}^l, \underline{m}^l, \overline{m}^r, \underline{m}^r, \overline{\sigma}^l, \underline{\sigma}^l, \overline{\sigma}^r, \underline{\sigma}^r, \gamma, \overline{\omega}, \underline{\omega}, c, s$ that are chosen randomly between [-1, 1]. The numbers of the TIVNFS-A's rule is set to be 2, then the structure of TIVNFS-A is 2-4-2-4-2-1, and the numbers of parameter of TIVNFS-A is 56.

Figures 4 and 5 show the results of nonlinear system identification after 50 epochs training and testing, respectively (solid line: actual output; dashed line: TIVNFS-A's output). The convergence of RMSE is shown in Fig. 6.

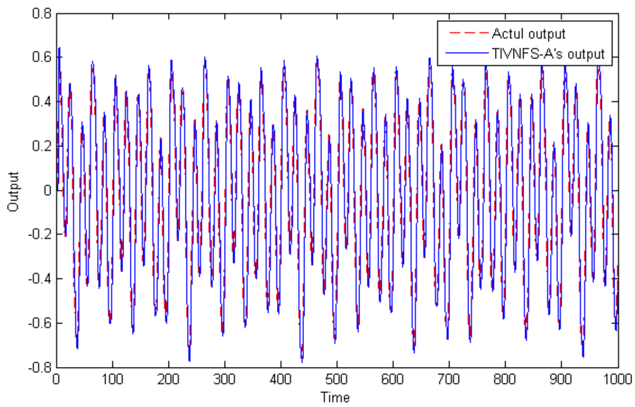

Fig. 4. Training results of nonlinear system identification.

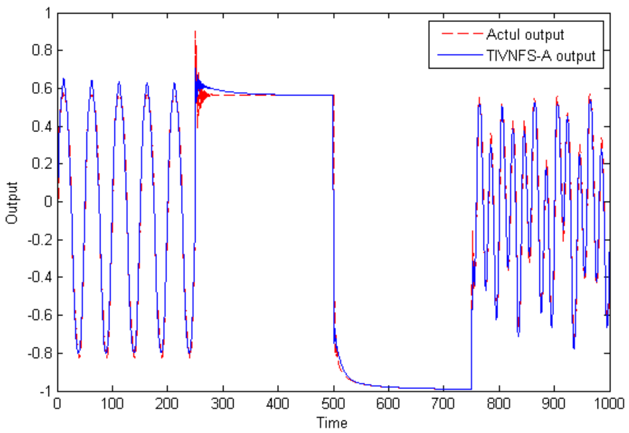

Fig. 5. Testing results of nonlinear system identification.

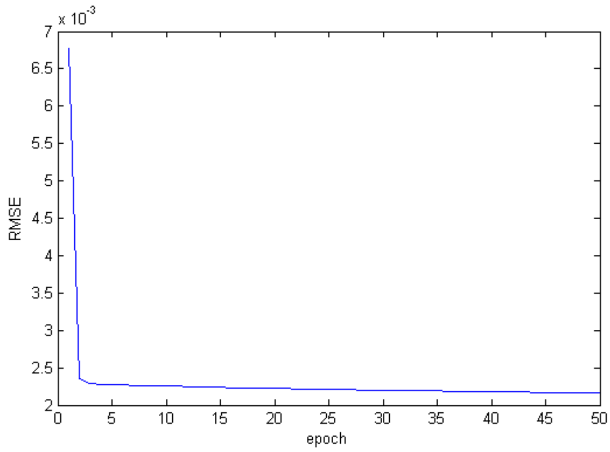

Fig. 6. The values of RMSE after training.

Illustration Comparison of the Optimal Learning Rate: Figure 7 and TABLE I show the comparison results between the TIVNFS-A system using the optimal learning rate η^* and fixed learning rate η. Obviously, we obtain the performance of BP algorithm with optimal learning rate is better than without optimal learning rate. The average RMSE of BP algorithm with η^* was 0.002377 and The average RMSE of BP algorithm with optimal learning rate was 0.003203. Thus, the performance is more stable with optimal learning from the best and the worst value of RMSE.

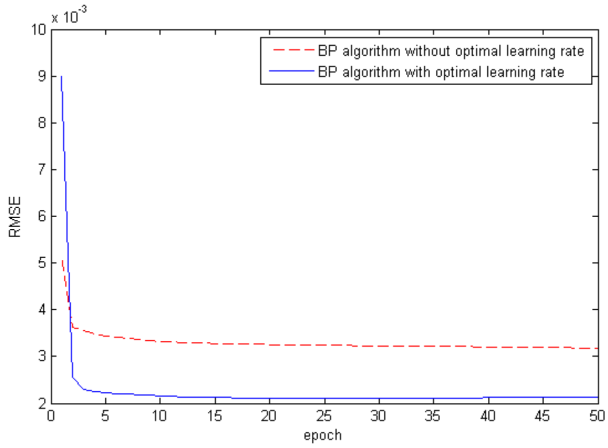

Fig. 7. The values of RMSE of BP algorithm with optimal learning rate and without optimal learning rate.

TABLE I
THE COMPARISON RESULTS IN 10 TIMES OF DIFFERENT LEARNING RATE

	BP algorithm with optimal learning rate	BP algorithm with fixed learning rate
Best	0.001671	0.001696
Average	0.002377	0.003203
Worst	0.003458	0.006690
Times(sec)	21.904	15.808

Illustration Comparison of TIVNFS-A system: In this comparison, we discuss the performance of TIVNFS-A and RFNN. Figure 8 shows the result of the values of RMSE after training with TIVNFS-A and RFNN. The RMSE of TIVNFS-A is better than RFNN at last epoch. In addition, the convergent speed of TIVNFS-A is also faster than using RFNN. TABLE II shows the best average and worst RMSE of TIVNFS-A and RFNN after training. According to the results, the performance of identification by TIVNFS-A is

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

better than by RFNN. We solve the nonlinear system identification by TIVNFS-A successfully and the simulation result the better performance.

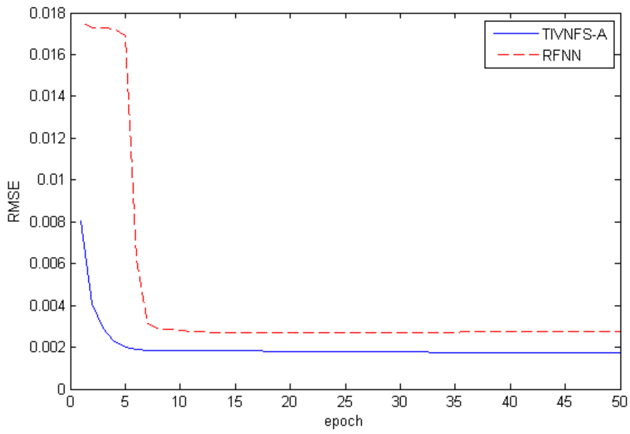

Fig. 8. The values of RMSE of TIVNFS-A and RFNN.

TABLE II
THE COMPARISON RESULTS IN 10 TIMES OF DIFFERENT NEURAL NETWORKS

	TIVNFS-A	RFNN
Best	0.001696	0.002169
Average	0.003203	0.010345
Worst	0.006690	0.017337
Times(sec)	15.808	1.050

V. CONCLUSION

In this paper, we have proposed a novel TSK-type interval-valued neural fuzzy system with asymmetric fuzzy membership functions (TIVNFS-A) for application of nonlinear system identification. In addition, the corresponding type reduction procedure is integrated in the adaptive network layers to reduce the amount of computation in the system. Based on the Lyapunov stability theorem, the TIVNFS-A system is trained by the back-propagation (BP) algorithm having an optimal learning rate (adaptive learning rate) to guarantee the stability and faster convergence. Illustration examples are shown to demonstrate the effectiveness and performance of the proposed TIVNFS-A with optimal BP learning algorithm.

REFERENCES

[1] H. Bustince, "Indicator of Inclusion Grade for Interval-valued Fuzzy Sets. Application to Approximate Reasoning based on Interval-valued Fuzzy Sets," *Int. J. of Approximate Reasoning*, Vol. 23, No. 3, pp. 137-209, 2000.
[2] H. Bustince and P Burillo, "Vague Sets are Intuitionistic Fuzzy Sets," *Fuzzy Sets andSystems*, Vol. 79, pp. 403-405, 1996.
[3] J. F. Baldwin and S. B. Karake, "Asymmetric Triangular Fuzzy Sets for Classification Models," *Lecture Notes in Artificial Intelligence*, Vol. 2773, pp. 364-370, 2003.
[4] M. Biglarbegian, W. W. Melek, and J. M. Mendel, "On the Stability of Interval Type-2 TSK Fuzzy Logic Control Systems," *IEEE Trans. on Systems, Man, and Cybernetics, Part B*, Vol. 10, No. 1, in press, 2010.
[5] Y. C. Chen and C. C. Teng, "A Model Reference Control Structure Using a Fuzzy Neural Network," *Fuzzy Sets and Systems*, Vol. 73, No.3, pp. 291-312, 1995.
[6] J. R. Castro, O. Castillo, P. Melin, and A. R. Díaz, "A Hybrid Learning Algorithm for A Class of Interval Type-2 Fuzzy Neural Networks," *Information Sciences*, Vol. 179, No. 13, pp. 2175-2193, 2009.
[7] W. A. Farag, V. H. Quintana, and L. T. Germano, "A Genetic-based Neuro-fuzzy Approach for Modeling and Control of Dynamical Systems," *IEEE Trans. on Neural Networks*, Vol. 9, No. 5, pp. 756-767, 1998.
[8] V. G. Gudise and G. K. Venayagamoorthy, "Comparison of Particle Swarm Optimization and Backpropagation as Training Algorithm for Neural Networks," *IEEE Swarm Intelligence Symp.*, pp. 110-117, April, 2003.
[9] S. Horikawa, T. Furuhashi, and Y. Uchikawa, "On Fuzzy Modeling Using Fuzzy Neural Networks with the Back-propagation Algorithm," *IEEE Trans. on Neural Networks*, Vol. 3, No. 5, pp. 801-806, 1992.
[10] D. H. Kim, "Parameter Tuning of Fuzzy Neural Networks by Immune Algorithm," *IEEE Int. Conf. on Fuzzy Systems*, Vol. 1, pp. 408-413, May, 2002.
[11] M. S. Kim, C. H. Kim, and J. J. Lee, "Evolutionary Optimization of Fuzzy Models with Asymmetric RBF Membership Functions Using Simplified Fitness Sharing," *Lecture Notes in Artificial Intelligence*, Vol. 2715, pp. 628-635, 2003.
[12] N. N. Karnik, J. Mendel, and Q. Liang, "Type-2 Fuzzy Logic Systems," *IEEE Trans. on Fuzzy Systems*, Vol. 7, No. 6, pp. 643-658, 1999.
[13] C. H. Lee, "Stabilization of Nonlinear Nonminimum Phase Systems: An Adaptive Parallel Approach Using Recurrent Fuzzy Neural Network," *IEEE Trans. on Systems, Man, Cybernetics- Part: B*, Vol. 34, No. 2, pp. 1075-1088, 2004.
[14] C. H. Lee and M. H. Chiu, "Adaptive Nonlinear Control Using TSK-type Recurrent Fuzzy Neural Network System," *Lecture Notes in Computer Science*, Vol. 4491, pp. 38-44, 2007.
[15] C. H. Lee and C. C. Teng, "Identification and Control of Dynamic Systems Using Recurrent Fuzzy Neural Networks," *IEEE Trans. on Fuzzy Systems*, Vol. 8, No. 4, pp. 349-366, 2000.
[16] C. H. Lee and Y. C. Lin, "An Adaptive Type-2 Fuzzy Neural Controller for Nonlinear Uncertain Systems," *Control and Intelligent systems*, Vol. 12, No. 1, pp. 41-50, 2005.
[17] C. H. Lee and C. C. Teng, "Fine Tuning of Membership Functions for Fuzzy Neural Systems," *Asian Journal of Control*, Vol. 3, No. 3, pp. 216-225, 2001.
[18] C. H. Lee and H. Y. Pan, "Performance Enhancement for Neural Fuzzy Systems Using Asymmetric Membership Functions," *Fuzzy Sets and Systems*, Vol. 160, No. 7, pp. 949-971, 2009.
[19] C. Li, K. H. Cheng, and J. D. Lee, "Hybrid Learning Neuro-fuzzy Approach for Complex Modeling Using Asymmetric Fuzzy Sets," *Proc. of the 17th IEEE International Conf. on Tools with Artificial Intelligence*, pp. 397-401, 2005.
[20] C. J. Lin and W. H. Ho, "An Asymmetric-similarity-measure-based Neural Fuzzy Inference System," *Fuzzy Sets and Systems*, Vol. 152, pp. 535-551, 2005.
[21] C. T. Lin and C. S. G. Lee, *Neural Fuzzy Systems*, Prentice Hall: Englewood Cliff, 1996.
[22] P. Z. Lin and T. T. Lee, "Robust Self-organizing Fuzzy-neural Control Using Asymmetric Gaussian Membership Functions," *Int. J. of Fuzzy Systems*, Vol. 9, No. 2, pp. 77-86, 2007.
[23] J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and NewDirections*, Upper Saddle River, Prentice-Hall, NJ, 2001.
[24] T. Ozen and J. M. Garibaldi, "Effect of Type-2 Fuzzy Membership Function Shape on Modeling Variation in Human Decision Making," *IEEE Int. Conf. on Fuzzy Systems*, Vol. 2, pp. 971-976, 2004.
[25] M. N. H. Siddique and M. O. Tokhi, "Training Neural Networks: Backpropagation vs. Genetic Algorithms," *Proc. of Int. J. Conf. on Neural Networks*, Vol. 4, pp. 2673-2678, 2001.
[26] I. B. Turksen, "Interval Valued Fuzzy Sets based on Normal Forms," *Fuzzy Sets and Systems*, Vol. 20, Issue 2, pp. 191-210, 1986.
[27] J. S. Wang and Y. P. Chen, "A Fully Automated Recurrent Neural Network for Unknown Dynamic System Identification and Control," *IEEE Trans. on Circuits and Systems-I*, Vol. 56, No. 6, pp. 1363-1372, 2006.
[28] P. J. Werbs, "Neurocontrol and Supervised Learning: An Overview and Evaluation," *Handbook of Intelligent Control*, D. A. White and D. A. Sofge, eds, Van Nostrand Reinhold, New York, 1992.
[29] X. H. Yu, et al., "Dynamic Learning Rate Optimization of the Back Propagation Algorithm," *IEEE Trans. On Neural Networks*, vol. 6, pp. 669-677, May 1995.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)