Nonlinear System Identification Using Takagi-Sugeno-Kang Type Interval-valued Fuzzy Systems via Stable Learning Mechanism

Ching-Hung Lee and Yi-Han Lee

Abstract— **In this paper, we propose a stable learning mechanism for novel Takagi-Sugeno-Kang type interval-valued neural fuzzy systems with asymmetric fuzzy membership functions (called TIVNFS-A). The TIVNFS-A consists of asymmetric fuzzy membership functions and Takagi-Sugeno-Kang type consequent part to enhance the performance. The corresponding type reduction procedure is simplified and 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 optimized by the back-propagation (BP) algorithm having an optimal learning rate (adaptive learning rate) to guarantee the stable and faster convergence. Finally, the TIVNFS-A with the optimal stable learning mechanism is applied in nonlinear system identification to demonstrate the effectiveness and performance.**

Index Terms—**interval-valued fuzzy set, Lyapunov approach, asymmetric, TSK type, nonlinear system**

I. Introduction

In recent years, fuzzy neural networks (FNNs) are used successfully in many applications [5, 7-9, 10, 13-18, 20-22, 27, 31-32], such as classifications, prediction, nonlinear system identification, nonlinear control, and intelligent control. FNNs combine the advantages of fuzzy system and neural network to perform a fuzzy logic system in neural network structure. Interval type-2 fuzzy logic systems (IT2FLSs) have got lots of attention in many applications due to their ability to model uncertainties. Besides, many literatures have shown that an IT2FLS is the same as an interval-valued fuzzy logic system (IVFLS) [1-2, 24, 30]. 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 neural fuzzy systems (IVNFS) systems are presented to handle the system uncertainty [6, 16, 18].

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).

Yi-Han Lee is with Department of Electrical Engineering, Yuan-Ze University, Chung-li, Taoyuan 320, Taiwan. (e-mail: s994609@mail.yzu.edu.tw)

By designing the fuzzy logic systems in fuzzy partition and rule engine, symmetric and fixed membership functions (MFs), like Gaussian or triangular ones, 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, 17-18, 20-21, 25]. These results demonstrated that using AFMFs can improve the modeling capability and enhance the performance of approximation accuracy. The AFMFs provide the ability of high flexibility and more accurate [18]. In addition, it is well known that the Takagi-Sugeno-Kang (TSK) type FLS has the ability of universal approximation capability [4]. By the combination of the above advantages, the TSK-type interval-valued neural fuzzy system with AFMFs (called TIVNFS-A) is introduced and applied in the nonlinear system identification. Note that the corresponding type reduction procedure is simplified and integrated in the adaptive network layers to reduce the amount of computation in the system. Therefore, the Karnik-Mendel type-reduction procedure can be removed. This reduces the computational complexity.

For training and optimizing the neural fuzzy systems, the back-propagation algorithm is widely used and a powerful training technique [5, 8, 27]. 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. That is, the adaptive time-varying learning rate which ensures the stable and faster convergence is derived. Otherwise, for simulations, we used a simple nonlinear system to implement the basic performance of MISO TIVNFS-A. The other is used the Continuous Stirred Tank Reactor (CSTR) system as the plant of simulation because many papers [24, 28-29] had used that to shown the capability and performance of the neural network.

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.

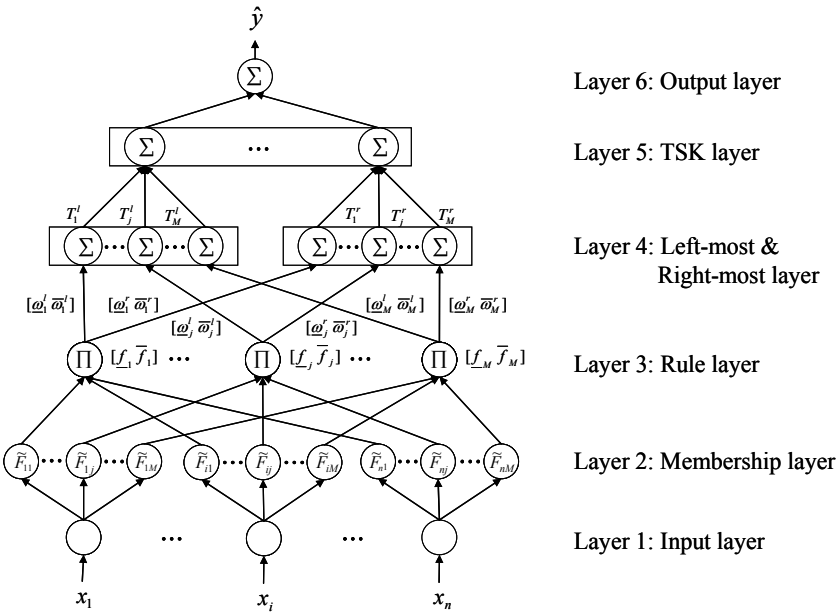

Figure 1: Diagram of the proposed TIVNFS-A with *M* rules.

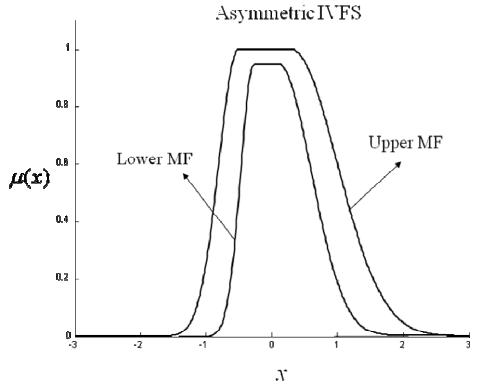

Figure 2: Constructed asymmetric interval valued fuzzy set (IVFS) [18].

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 fuzzy 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. A TIVNFS-A with M fuzzy rules is implemented as the six-layer network shown in Fig. 1. 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 $\widetilde{F}_{1j}$ and … and x_n is $\widetilde{F}_{nj}$, 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 $\widetilde{F}_{ij}$ is the antecedent fuzzy set. The fuzzy MFs of the antecedent part $\widetilde{F}_{ij}$ are the asymmetric interval-valued fuzzy sets (IVFSs) that are shown in Fig. 2, which are different from typical Gaussian MFs. Several approaches indicate that using asymmetric MFs can improve approximation accuracy [3, 12, 18, 20-22, 27]. The asymmetric IVFSs are generated from four Gaussian functions, as shown in Fig. 2. The construction of asymmetric IVFSs is introduced in Appendix A. Subsequently, the fuzzy sets of the consequent part C_{j0} and C_{ji} are designed to be convex, normal type-1 fuzzy subsets. 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.

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

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

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. 2), where the subscript "ij" indicates the jth term of the ith input, j=1, …, M. We use superscripts ^{l} and ^{r} to denote the left and right curves of the Gaussian membership function. The parameters of the lower and upper MFs are denoted by $\underline{\ }$ and $\bar{\ }$, respectively. Therefore, the output of layer two is

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

where

$$\overline{\mu}_{\widetilde{F}}(x) = \begin{cases} e^{-\frac{1}{2}\left(\frac{x-\overline{m}^l}{\overline{\sigma}^l}\right)^2}, & \text{for } x \le \overline{m}^l \\ 1, & \text{for } \overline{m}^l < x < \overline{m}^r \\ e^{-\frac{1}{2}\left(\frac{x-\overline{m}^r}{\overline{\sigma}^r}\right)^2}, & \text{for } \overline{m}^r \le x \end{cases}$$

and

$$\underline{\mu}_{\widetilde{F}}(x) = \begin{cases} \gamma \cdot e^{-\frac{1}{2}\left(\frac{x-\underline{m}^l}{\underline{\sigma}^l}\right)^2}, & \text{for } x \le \underline{m}^l \\ \gamma, & \text{for } \underline{m}^l < x < \underline{m}^r \\ \gamma \cdot e^{-\frac{1}{2}\left(\frac{x-\underline{m}^r}{\underline{\sigma}^r}\right)^2}, & \text{for } \underline{m}^r \le x \end{cases}.$$

Herein, $\overline{m}^l$, $\overline{m}^r$, $\underline{m}^l$, and $\underline{m}^r$ denote the uncertain means of Gaussian MFs, and $\overline{\sigma}^l$, $\overline{\sigma}^r$, $\underline{\sigma}^l$, and $\underline{\sigma}^r$ denote the uncertain deviations (width) of Gaussian MFs. To avoid a too small lower MFs value, γ is chosen between 0.5 and 1.

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 *j*th rule is calculated by

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

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

where $\underline{\mu}_{\tilde{F}_{ij}}(\cdot)$ and $\overline{\mu}_{\tilde{F}_{ij}}(\cdot)$ are the lower and upper membership grades of $\mu_{\tilde{F}}(\cdot)$, 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 $\left[\underline{\omega}_j^l \ \ \overline{\omega}_j^l\right]^T$ and $\left[\underline{\omega}_j^r \ \ \overline{\omega}_j^r\right]^T$, 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: $\underline{\omega}^l = [\underline{\omega}_1^l \cdots \underline{\omega}_M^l]^T$, $\overline{\omega}^l = \left[\overline{\omega}_1^l \ \cdots \ \overline{\omega}_M^l\right]^T$, $\underline{\omega}^r = [\underline{\omega}_1^r \cdots \underline{\omega}_M^r]^T$, and $\overline{\omega}^r = \left[\overline{\omega}_1^r \ \cdots \ \overline{\omega}_M^r\right]^T$. Therefore, the output of layer 4 is

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

This expression calculates the left-most points, $O_{jl}^{(4)}$, and right-most points, $O_{jr}^{(4)}$. This simplifies the calculation of the left and right end points and they are used in layers 4 and 5 to approximate the operation of type-reduction.

Layer 5 (TSK Layer): Since 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, $C_{ji}=[c_{ji}\text{-}s_{ji} \ c_{ji}+ s_{ji}]^T$, where i=1,…, n, and j=1,…, M. In this expression, c_{ji} denotes the center of C_{ji}, and s_{ji} denotes the spread of C_{ji}, i=1, 2, …, n and j=1, 2, …, M. Therefore, the TSK-type consequent part of *Rule j* is

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

where $s_{ji} \geq 0$. Therefore, the output of layer 5 is

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

Note that we need to compute the left-end point $O_l^{(5)}$ and right-end point $O_r^{(5)}$ by utilizing the KM algorithm. In the KM algorithm, we have to find the proper switch point value L and R by iterative procedure, and the left-end and right-end point should be calculated by equation (9). According to previously results [6], the type reduction is integrated into the layers 4 and 5. Therefore, the Karnik-Mendel type-reduction procedure can be removed. Since the iterative procedure for finding coefficients R and L is not necessary, the computational effort can be reduced effectively.

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 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. In addition, the above description can be extended to the multi-output TIVNFS-A system easily. Section IV introduces a two-input-two-output TIVNFS-A application for the identification of a continuous stirred tank reactor (CSTR) system.

III. Stable Learning Mechanism 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, 27]. The back-propagation algorithm is based on the gradient descent method to find 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 $e(k) = y_p(k) - \hat{y}(k) = y_p(k) - O^{(6)}(k)$, that $\hat{y}(k)$ and $y_p(k)$ are the TIVNFS-A's identified system 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 weighting vector, $\underline{\mathbf{W}}$ is the parameters of lower AFMFs, $\overline{\mathbf{W}}$ is upper AFMFs' parameters, and γ is the maximum value of lower AFMFs, i.e.,

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

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

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

thus

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \eta \cdot e(k) \cdot \frac{\partial O^{(6)}(k)}{\partial \mathbf{W}}. \tag{15}$$

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 $\mathbf{C}$. According to (11)-(15), we can obtain the update rule of each parameter.

-The Derivation of update law for $\mathbf{C}$

The update law of the TIVNFS-A's parameters $\mathbf{C}$ is

$$c_j(k+1) = c_j(k) + \eta \cdot e(k)\frac{\partial \hat{y}(k)}{\partial c_j} \tag{16}$$

$$s_j(k+1) = s_j(k) + \eta \cdot e(k)\frac{\partial \hat{y}(k)}{\partial s_j} \tag{17}$$

where the partial derivative with respect to c_{ji} and s_{ji} are

$$\frac{\partial \hat{y}(k)}{\partial c_{j0}} = \frac{1}{2}\left[\frac{O_{jl}^{(4)}}{\sum_{j=1}^{M} O_{jl}^{(4)}} + \frac{O_{jr}^{(4)}}{\sum_{j=1}^{M} O_{jr}^{(4)}}\right] \tag{18}$$

$$\frac{\partial \hat{y}}{\partial c_{ji}} = \frac{1}{2}\cdot x_i \cdot \left[\frac{O_{jl}^{(4)}}{\sum_{j=1}^{M} O_{jl}^{(4)}} + \frac{O_{jr}^{(4)}}{\sum_{j=1}^{M} O_{jr}^{(4)}}\right] \tag{19}$$

$$\frac{\partial \hat{y}}{\partial s_{j0}} = \frac{1}{2}\left[-\frac{O_{jl}^{(4)}}{\sum_{j=1}^{M} O_{jl}^{(4)}} + \frac{O_{jr}^{(4)}}{\sum_{j=1}^{M} O_{jr}^{(4)}}\right] \tag{20}$$

$$\frac{\partial \hat{y}}{\partial s_{ji}} = \frac{1}{2}\cdot |x_i| \cdot \left[-\frac{O_{jl}^{(4)}}{\sum_{j=1}^{M} O_{jl}^{(4)}} + \frac{O_{jr}^{(4)}}{\sum_{j=1}^{M} O_{jr}^{(4)}}\right]. \tag{21}$$

-The Derivation of update law for $\mathbf{W}_\omega$

The update law of the TIVNFS-A's parameters $\mathbf{W}_\omega$ is

$$\underline{\omega}_j(k+1) = \underline{\omega}_j(k) + \eta \cdot e(k)\frac{\partial \hat{y}(k)}{\partial \underline{\omega}_j} \tag{22}$$

$$\overline{\omega}_j(k+1) = \overline{\omega}_j(k) + \eta \cdot e(k)\frac{\partial \hat{y}(k)}{\partial \overline{\omega}_j} \tag{23}$$

where the partial derivative with respect to $\overline{\omega}_j$ and $\underline{\omega}_j$ are

$$\frac{\partial \hat{y}(k)}{\partial \underline{\omega}_j^l} = \frac{1}{2}\left[\frac{T_j^l - O_l^{(5)}}{\sum_{j=1}^{M} O_{jl}^{(4)}} \cdot \frac{\underline{O}_j^{(3)} - O_{jl}^{(4)}}{\underline{\omega}_j^l + \overline{\omega}_j^l}\right] \tag{24}$$

$$\frac{\partial \hat{y}(k)}{\partial \underline{\omega}_j^r} = \frac{1}{2}\left[\frac{T_j^r - O_r^{(5)}}{\sum_{j=1}^{M} O_{jr}^{(4)}} \cdot \frac{\underline{O}_j^{(3)} - O_{jr}^{(4)}}{\underline{\omega}_j^r + \overline{\omega}_j^r}\right] \tag{25}$$

$$\frac{\partial \hat{y}(k)}{\partial \overline{\omega}_j^l} = \frac{1}{2}\left[\frac{T_j^l - O_l^{(5)}}{\sum_{j=1}^{M} O_{jl}^{(4)}} \cdot \frac{\overline{O}_j^{(3)} - O_{jl}^{(4)}}{\underline{\omega}_j^l + \overline{\omega}_j^l}\right] \tag{26}$$

$$\frac{\partial \hat{y}(k)}{\partial \overline{\omega}_j^r} = \frac{1}{2}\left[\frac{T_j^r - O_r^{(5)}}{\sum_{j=1}^{M} O_{jr}^{(4)}} \cdot \frac{\overline{O}_j^{(3)} - O_{jr}^{(4)}}{\underline{\omega}_j^r + \overline{\omega}_j^r}\right]. \tag{27}$$

-The Derivation of update law for $\underline{\mathbf{W}}$

The update law of the TIVNFS-A's parameters $\underline{\mathbf{W}}$ is

$$\underline{m}_{ij}(k+1) = \underline{m}_{ij}(k) + \eta \cdot e(k)\frac{\partial \hat{y}(k)}{\partial \underline{m}_{ij}} \tag{28}$$

$$\underline{\sigma}_{ij}(k+1) = \underline{\sigma}_{ij}(k) + \eta \cdot e(k)\frac{\partial \hat{y}(k)}{\partial \underline{\sigma}_{ij}} \tag{29}$$

where the partial derivative with respect to $\underline{m}_{ij}$ and $\underline{\sigma}_{ij}$ are

$$\begin{aligned}\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}} = -\frac{1}{4}\gamma\Bigg[&\frac{T_j^l - O_l^{(5)}}{\sum_{j=1}^{M} O_{jl}^{(4)}} \cdot \frac{\underline{\omega}_j^l \cdot \underline{O}_j^{(3)}}{\underline{\omega}_j^l + \overline{\omega}_j^l}\\ &+ \frac{T_j^r - O_r^{(5)}}{\sum_{j=1}^{M} O_{jr}^{(4)}} \cdot \frac{\underline{\omega}_j^r \cdot \underline{O}_j^{(3)}}{\underline{\omega}_j^r + \overline{\omega}_j^r}\Bigg] \cdot \frac{\partial\left(\frac{x - \underline{m}}{\underline{\sigma}}\right)^2}{\partial \underline{\mathbf{W}}}\end{aligned} \tag{30}$$

where

$$\frac{\partial}{\partial \underline{m}_{ij}^l}\left(\frac{x_i - \underline{m}_{ij}^l}{\underline{\sigma}_{ij}^l}\right)^2 = -2\frac{\left(x_i - \underline{m}_{ij}^l\right)}{\left(\underline{\sigma}_{ij}^l\right)^2} \tag{31}$$

$$\frac{\partial}{\partial \underline{m}_{ij}^r}\left(\frac{x_i - \underline{m}_{ij}^r}{\underline{\sigma}_{ij}^r}\right)^2 = -2\frac{\left(x_i - \underline{m}_{ij}^r\right)}{\left(\underline{\sigma}_{ij}^r\right)^2} \tag{32}$$

$$\frac{\partial}{\partial \underline{\sigma}_{ij}^l}\left(\frac{x_i - \underline{m}_{ij}^l}{\underline{\sigma}_{ij}^l}\right)^2 = -2\frac{\left(x_i - \underline{m}_{ij}^l\right)^2}{\left(\underline{\sigma}_{ij}^l\right)^3} \tag{33}$$

$$\frac{\partial}{\partial \underline{\sigma}_{ij}^r}\left(\frac{x_i - \underline{m}_{ij}^r}{\underline{\sigma}_{ij}^r}\right)^2 = -2\frac{\left(x_i - \underline{m}_{ij}^r\right)^2}{\left(\underline{\sigma}_{ij}^r\right)^3}. \tag{34}$$

-The Derivation of update law for $\overline{\mathbf{W}}$

The update law of the TIVNFS-A's parameters $\overline{\mathbf{W}}$ is

$$\overline{m}_{ij}(k+1) = \overline{m}_{ij}(k) + \eta \cdot e(k)\frac{\partial \hat{y}(k)}{\partial \overline{m}_{ij}} \tag{35}$$

$$\overline{\sigma}_{ij}(k+1)=\overline{\sigma}_{ij}(k)+\eta\cdot e(k)\frac{\partial\hat{y}(k)}{\partial\overline{\sigma}_{ij}} \tag{36}$$

where the partial derivative with respect to $\overline{m}_{ij}$ and $\overline{\sigma}_{ij}$ are

$$\frac{\partial\hat{y}(k)}{\partial\overline{\mathbf{W}}}=\left[-\frac{1}{4}\left(\frac{T_j^l-O_l^{(5)}}{\sum_{j=1}^{M}O_{jl}^{(4)}}\cdot\frac{\overline{\omega}_j^l\cdot\overline{O}_j^{(3)}}{\underline{\omega}_j^l+\overline{\omega}_j^l}+\frac{T_j^r-O_r^{(5)}}{\sum_{j=1}^{M}O_{jr}^{(4)}}\cdot\frac{\overline{\omega}_j^r\cdot\overline{O}_j^{(3)}}{\underline{\omega}_j^r+\overline{\omega}_j^r}\right)\right]\cdot\frac{\partial\left(\frac{x-\overline{m}}{\overline{\sigma}}\right)^2}{\partial\overline{\mathbf{W}}} \tag{37}$$

where

$$\frac{\partial}{\partial\overline{m}_{ij}^l}\left(\frac{x_i-\overline{m}_{ij}^l}{\overline{\sigma}_{ij}^l}\right)^2=-2\frac{\left(x_i-\overline{m}_{ij}^l\right)}{\left(\overline{\sigma}_{ij}^l\right)^2} \tag{38}$$

$$\frac{\partial}{\partial\overline{m}_{ij}^r}\left(\frac{x_i-\overline{m}_{ij}^r}{\overline{\sigma}_{ij}^r}\right)^2=-2\frac{\left(x_i-\overline{m}_{ij}^r\right)}{\left(\overline{\sigma}_{ij}^r\right)^2} \tag{39}$$

$$\frac{\partial}{\partial\overline{\sigma}_{ij}^l}\left(\frac{x_i-\overline{m}_{ij}^l}{\overline{\sigma}_{ij}^l}\right)^2=-2\frac{\left(x_i-\overline{m}_{ij}^l\right)^2}{\left(\overline{\sigma}_{ij}^l\right)^3} \tag{40}$$

$$\frac{\partial}{\partial\overline{\sigma}_{ij}^r}\left(\frac{x_i-\overline{m}_{ij}^r}{\overline{\sigma}_{ij}^r}\right)^2=-2\frac{\left(x_i-\overline{m}_{ij}^r\right)^2}{\left(\overline{\sigma}_{ij}^r\right)^3}. \tag{41}$$

-The Derivation of update law for γ

The update law of the TIVNFS-A's parameters γ is

$$\gamma_{ij}(k+1)=\gamma_{ij}(k)+\eta\cdot e(k)\frac{\partial\hat{y}(k)}{\partial\gamma_{ij}} \tag{42}$$

where the partial derivative with respect to γ is

$$\frac{\partial\hat{y}(k)}{\partial\gamma_{ij}}=\frac{1}{2}\left[\frac{T_j^l-O_l^{(5)}}{\sum_{j=1}^{M}O_{jl}^{(4)}}\cdot\frac{\underline{\omega}_j^l}{\underline{\omega}_j^l+\overline{\omega}_j^l}+\frac{T_j^r-O_r^{(5)}}{\sum_{j=1}^{M}O_{jr}^{(4)}}\cdot\frac{\underline{\omega}_j^r}{\underline{\omega}_j^r+\overline{\omega}_j^r}\right]\cdot\frac{\underline{O}_j^{(3)}}{\gamma_{ij}}. \tag{43}$$

Equations (16)-(43) introduce the training of TIVNFS-A for nonlinear system identification by gradient-descent method.

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 important but it is not easy to choose properly. Thus, we use the Lyapunove function to find the optimal learning rate to avoid the local minimum and enhance the convergence performance [5, 15, 33]. At first, we defined the positive Lyapunov candidate

$$V(k)=E(k)=\frac{1}{2}e^2(k). \tag{44}$$

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

$$\begin{aligned}\Delta V(k)&=V(k+1)-V(k)\\&=\frac{1}{2}\left[(e(k+1)-e(k))\cdot(e(k+1)+e(k))\right]\\&=\frac{1}{2}\left[\Delta e(k)\cdot(2e(k)+\Delta e(k))\right]\\&=\frac{1}{2}\left[\Delta e^2(k)+2e(k)\cdot\Delta e(k)\right].\end{aligned} \tag{45}$$

Note that $\Delta e(k)\approx(\partial e/\partial\mathbf{W})\Delta\mathbf{W}$ and according to the parameter update laws (12) $\Delta\mathbf{W}=-\eta(\partial E/\partial\mathbf{W})$, we have

$$\Delta e(k)\approx\frac{\partial e}{\partial\mathbf{W}}\left(-\eta\frac{\partial E}{\partial\mathbf{W}}\right)=-\eta\cdot e(k)\cdot\left(\frac{\partial\hat{y}}{\partial\mathbf{W}}\right). \tag{46}$$

Thus,

$$\begin{aligned}\Delta V(k)&=\frac{1}{2}\left[-2\eta\cdot e^2\cdot\left(\frac{\partial\hat{y}}{\partial\mathbf{W}}\right)^2+\eta^2\cdot e^2\cdot\left(\frac{\partial\hat{y}}{\partial\mathbf{W}}\right)^4\right]\\&=-\frac{1}{2}\eta\cdot e^2\cdot\left(\frac{\partial\hat{y}}{\partial\mathbf{W}}\right)^2\left[2-\eta\cdot\left(\frac{\partial\hat{y}}{\partial\mathbf{W}}\right)^2\right].\end{aligned} \tag{47}$$

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

$$0<\eta\left(\frac{\partial\hat{y}}{\partial\mathbf{W}}\right)^2<2. \tag{48}$$

Define $\lambda=\eta\cdot\left(\frac{\partial\hat{y}}{\partial\mathbf{W}}\right)^2$ and rewrite (47) as

$$\Delta V(k)=V(k)\cdot\lambda\cdot(-2+\lambda)\le 0. \tag{49}$$

Then we have

$$V(k+1)-V(k)=V(k)\cdot\lambda\cdot(-2+\lambda)\le 0 \tag{50}$$

and

$$V(k+1)=V(k)\cdot(1-2\lambda+\lambda^2)\le 0. \tag{51}$$

Thus, the optimal learning rate η can be obtained

$$\eta^*=\left(\frac{\partial\hat{y}}{\partial\mathbf{W}}\right)^{-2} \tag{52}$$

such that λ=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

$$\eta^*=\frac{1}{D}\left(\frac{\partial\hat{y}}{\partial\mathbf{W}}\right)^{-2}. \tag{53}$$

The optimal learning rates of TIVNFS-A's parameters are introduced as the following.

- Optimal learning of C

$$\eta_{c_{j0}}=\frac{1}{D}\left(\frac{\partial\hat{y}(k)}{\partial c_{j0}}\right)^{-2}=\frac{1}{D}\left[\frac{1}{2}\left(\frac{O_{jl}^{(4)}}{\sum_{j=1}^{M}O_{jl}^{(4)}}+\frac{O_{jr}^{(4)}}{\sum_{j=1}^{M}O_{jr}^{(4)}}\right)\right]^{-2} \tag{54}$$

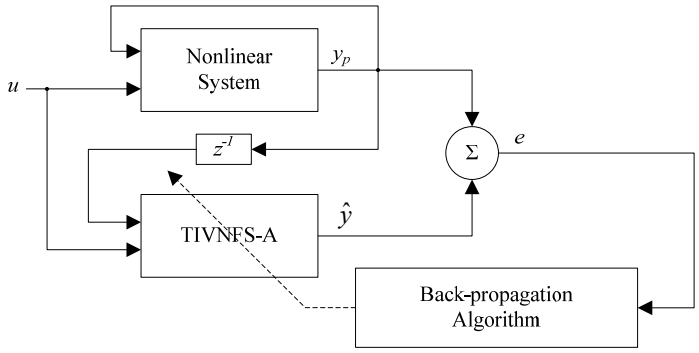

Figure 3: Series-parallel model with the TIVNFS-A for nonlinear systems identification.

$$\eta_{c_{ji}} = \frac{1}{D}\left(\frac{\partial \hat{y}(k)}{\partial c_{ji}}\right)^{-2} = \frac{1}{D}\left[\frac{x_i}{2}\cdot\left(\frac{O_{jl}^{(4)}}{\sum_{j=1}^{M} O_{jl}^{(4)}} + \frac{O_{jr}^{(4)}}{\sum_{j=1}^{M} O_{jr}^{(4)}}\right)\right]^{-2} \quad (55)$$

$$\eta_{s_{j0}} = \frac{1}{D}\left(\frac{\partial \hat{y}(k)}{\partial s_{j0}}\right)^{-2} = \frac{1}{D}\left[\frac{1}{2}\left(-\frac{O_{jl}^{(4)}}{\sum_{j=1}^{M} O_{jl}^{(4)}} + \frac{O_{jr}^{(4)}}{\sum_{j=1}^{M} O_{jr}^{(4)}}\right)\right]^{-2} \quad (56)$$

$$\eta_{s_{ji}} = \frac{1}{D}\left(\frac{\partial \hat{y}(k)}{\partial s_{ji}}\right)^{-2} = \frac{1}{D}\left[\frac{|x_i|}{2}\cdot\left(\frac{O_{jl}^{(4)}}{\sum_{j=1}^{M} O_{jl}^{(4)}} + \frac{O_{jr}^{(4)}}{\sum_{j=1}^{M} O_{jr}^{(4)}}\right)\right]^{-2}. \quad (57)$$

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

$$\eta_{\underline{\mathbf{W}}} = \frac{1}{D}\left(\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}}\right)^{-2} = \left[-\frac{1}{4}\gamma\left(\frac{T_j^l - O_l^{(5)}}{\sum_{j=1}^{M} O_{jl}^{(4)}}\cdot\frac{\underline{\omega}_j^l \cdot \underline{O}_j^{(3)}}{\underline{\omega}_j^l + \overline{\omega}_j^l} + \frac{T_j^r - O_r^{(5)}}{\sum_{j=1}^{M} O_{jr}^{(4)}}\cdot\frac{\underline{\omega}_j^r \cdot \underline{O}_j^{(3)}}{\underline{\omega}_j^r + \overline{\omega}_j^r}\right)\cdot\frac{\partial}{\partial \underline{\mathbf{W}}}\left(\frac{x-\underline{m}}{\underline{\sigma}}\right)^2\right]^{-2}. \quad (58)$$

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

$$\eta_{\overline{\mathbf{W}}} = \frac{1}{D}\left(\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}}\right)^{-2} = \left[-\frac{1}{4}\gamma\left(\frac{T_j^l - O_l^{(5)}}{\sum_{j=1}^{M} O_{jl}^{(4)}}\cdot\frac{\overline{\omega}_j^l \cdot \overline{O}_j^{(3)}}{\underline{\omega}_j^l + \overline{\omega}_j^l} + \frac{T_j^r - O_r^{(5)}}{\sum_{j=1}^{M} O_{jr}^{(4)}}\cdot\frac{\overline{\omega}_j^r \cdot \overline{O}_j^{(3)}}{\underline{\omega}_j^r + \overline{\omega}_j^r}\right)\cdot\frac{\partial}{\partial \overline{\mathbf{W}}}\left(\frac{x-\overline{m}}{\overline{\sigma}}\right)^2\right]^{-2}. \quad (59)$$

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

$$\eta_{\underline{\omega}_j^l} = \frac{1}{D}\left(\frac{\partial \hat{y}(k)}{\partial \underline{\omega}_j^l}\right)^{-2} = \frac{1}{D}\left[\frac{1}{2}\left(\frac{T_j^l - O_l^{(5)}}{\sum_{j=1}^{M} O_{jl}^{(4)}}\cdot\frac{\underline{O}_j^{(3)} - O_{jl}^{(4)}}{\underline{\omega}_j^l + \overline{\omega}_j^l}\right)\right]^{-2} \quad (60)$$

$$\eta_{\underline{\omega}_j^r} = \frac{1}{D}\left(\frac{\partial \hat{y}(k)}{\partial \underline{\omega}_j^r}\right)^{-2} = \frac{1}{D}\left[\frac{1}{2}\left(\frac{T_j^r - O_r^{(5)}}{\sum_{j=1}^{M} O_{jr}^{(4)}}\cdot\frac{\underline{O}_j^{(3)} - O_{jr}^{(4)}}{\underline{\omega}_j^r + \overline{\omega}_j^r}\right)\right]^{-2} \quad (61)$$

$$\eta_{\overline{\omega}_j^l} = \frac{1}{D}\left(\frac{\partial \hat{y}(k)}{\partial \overline{\omega}_j^l}\right)^{-2} = \frac{1}{D}\left[\frac{1}{2}\left(\frac{T_j^l - O_l^{(5)}}{\sum_{j=1}^{M} O_{jl}^{(4)}}\cdot\frac{\overline{O}_j^{(3)} - O_{jl}^{(4)}}{\underline{\omega}_j^l + \overline{\omega}_j^l}\right)\right]^{-2} \quad (62)$$

$$\eta_{\overline{\omega}_j^r} = \frac{1}{D}\left(\frac{\partial \hat{y}(k)}{\partial \overline{\omega}_j^r}\right)^{-2} = \frac{1}{D}\left[\frac{1}{2}\left(\frac{T_j^r - O_r^{(5)}}{\sum_{j=1}^{M} O_{jr}^{(4)}}\cdot\frac{\overline{O}_j^{(3)} - O_{jr}^{(4)}}{\underline{\omega}_j^r + \overline{\omega}_j^r}\right)\right]^{-2}. \quad (63)$$

- Optimal learning of γ

$$\eta_{\gamma_{ij}} = \frac{1}{D}\left(\frac{\partial \hat{y}(k)}{\partial \gamma_{ij}}\right)^{-2} = \frac{1}{D}\left[\frac{1}{2}\left(\frac{T_j^l - O_l^{(5)}}{\sum_{j=1}^{M} O_{jl}^{(4)}}\cdot\frac{\underline{\omega}_j^l}{\underline{\omega}_j^l + \overline{\omega}_j^l} + \frac{T_j^r - O_r^{(5)}}{\sum_{j=1}^{M} O_{jr}^{(4)}}\cdot\frac{\underline{\omega}_j^r}{\underline{\omega}_j^r + \overline{\omega}_j^r}\right)\cdot\frac{\underline{O}_j^{(3)}}{\gamma_{ij}}\right]^{-2}. \quad (64)$$

IV. Simulation Results

In this section, two illustrated examples of nonlinear system identification are introduced to show the performance and effectiveness of 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.

Example 1: Nonlinear system identification

Consider the following nonlinear system [15]

$$y_p(k+1) = f\left(y_p(k), y_p(k-1), y_p(k-2), u(k), u(k-1)\right) \quad (65)$$

where

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

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

$$e(k) \equiv y_p(k) - \hat{y}(k), \tag{66}$$

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

$$u(k) = 0.3\sin(\frac{\pi k}{25}) + 0.1\sin(\frac{\pi k}{32}) + 0.6\sin(\frac{\pi k}{10}) \tag{67}$$

and the testing input is

$$u(k) = \begin{cases} \sin(\frac{\pi k}{25}) & 0 < k < 250, \\ 1.0 & 250 \le k < 500, \\ -1.0 & 500 \le k < 750, \\ 0.3\sin(\frac{\pi k}{25}) + 0.1\sin(\frac{\pi k}{32}) & \\ \quad + 0.6\sin(\frac{\pi k}{10}) & 750 \le k < 1000. \end{cases} \tag{68}$$

The following RMSE is adopted to be the performance index

$$RMSE : (\sum_{k} e^2(k)/N)^{1/2} \tag{69}$$

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 training and testing results of nonlinear system identification after 50 epochs. Figures 4(a) and 4(b) shows the actual system output and TIVNFS-A's output after training, respectively. In addition, to illustrate the generalization performance of TIVNFS-A, Figs. 5(a) and 5(b) show the testing results, (a) actual output and (b) TIVNFS-A's output). From Figures 4 and 5, we can observe that the training and testing results are almost the same as the actual outputs. These demonstrate the effectiveness of TIVNFS-A. The convergence of RMSE in training error is shown in Fig. 6. The average RMSE of ten independent runs after 50 epochs is 2.377×10^{-3}.

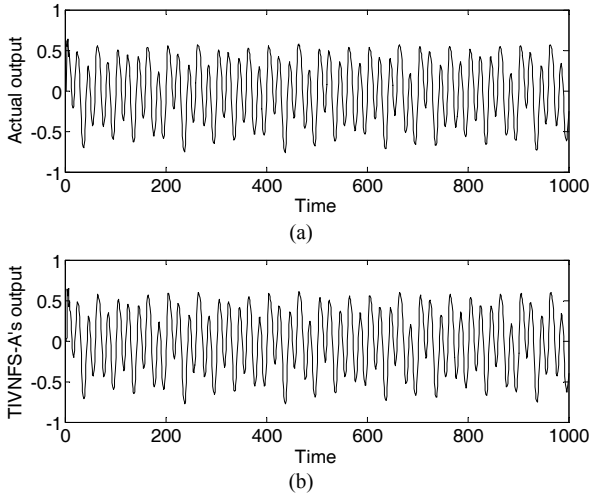

Figure 4: (a)The actual output of nonlinear system, (b)Training results of nonlinear system identification.

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^* (solid-line) and fixed learning rate η (dashed-line). Obviously, we obtain the performance of BP algorithm with optimal learning rate is better than one without optimal learning rate. The average RMSE of BP algorithm with η^* and ones with the fixed learning rate is 0.002377 and 0.003203, respectively. In addition, the performance of the best and the worst in RMSE are more stable.

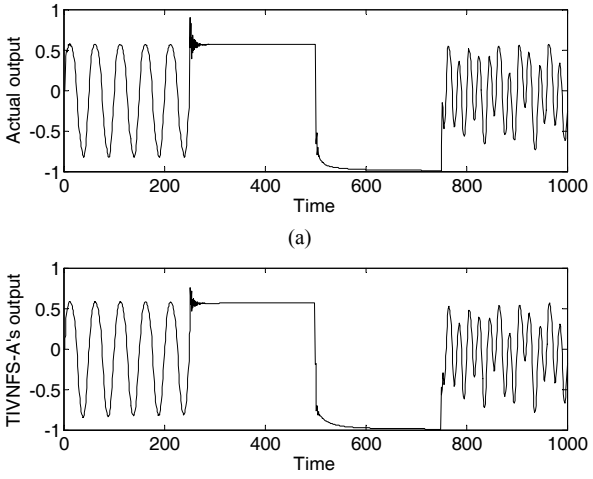

Figure 5: (a)The actual output of nonlinear system, (b)Testing results of nonlinear system identification.

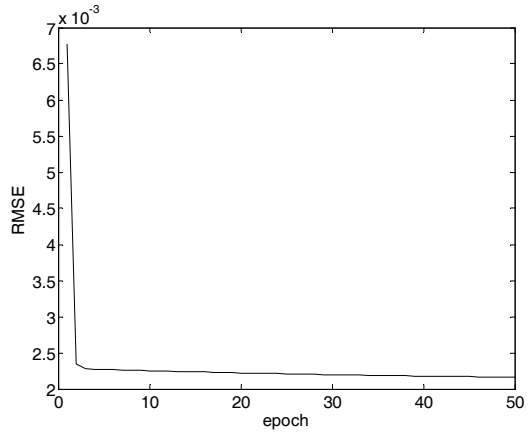

Figure 6: The values of RMSE after training.

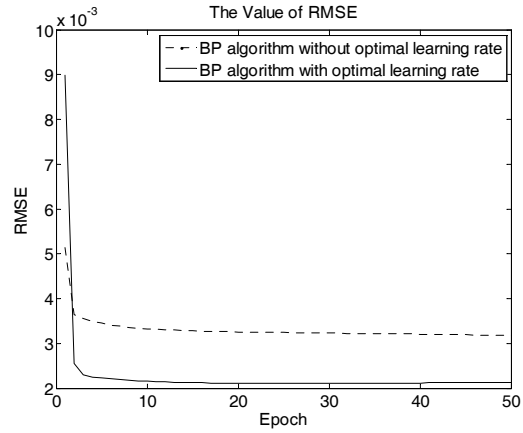

Figure 7: The values of RMSE of BP algorithm with optimal learning rate and without optimal learning rate.

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 performance of TIVNFS-A is better than RFNN's in comparison of RMSE. In addition, the convergent speed of TIVNFS-A is also faster than ones of 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 better than by RFNN. We solve the nonlinear system identification by TIVNFS-A successfully and the simulation result the better performance.

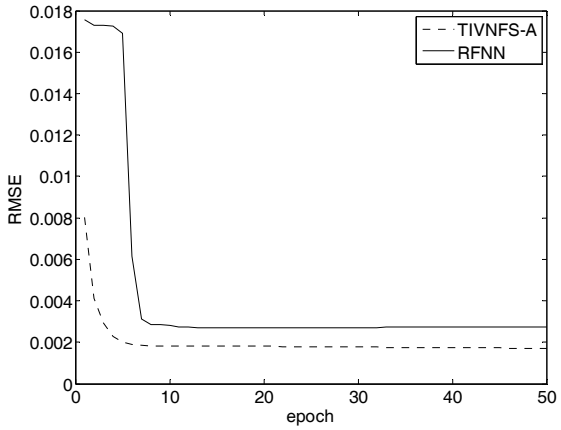

Figure 8: The values of RMSE of TIVNFS-A and RFNN.

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
imes(sec)	15.808	1.050

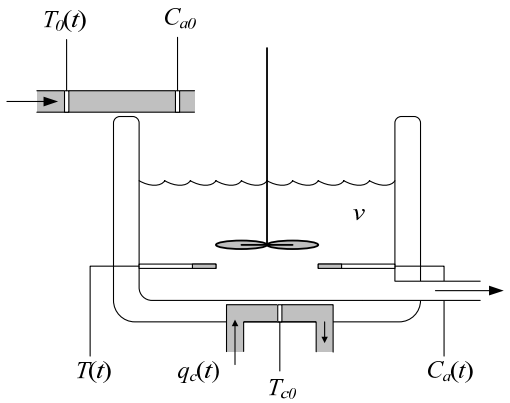

Figure 9: Continuous Stirred Tank Reactor.

TABLE III
THE PARAMETER OF CSTR PLANT

Parameter	Description	Value
C_{a0}	The concentration of inlet feed	1 (mol·l^{-1})
T_0	The temperature of inlet feed	350 (K)
T_{c0}	The temperature of inlet coolant	350 (K)
v	The volume in the reactor tank	100 (l)
k_0	Reaction rate constant	7.2×1010 (min^{-1})
E/R	Energy of activation	9.95×103 (K)
ΔH	Heat of reaction	-2×105 (cal·mol^{-1})
C_p, C_{pc}	Specific heats	1 (cal·g-1·K^{-1})
ρ, ρ_c	Liquid densities	1×103 (g·l-1)
h_a	Heat transfer coefficient	7×105 (cal·min^{-1}·K^{-1})

Example 2: Identification of nonlinear CSTR system

In Example 1, we have shown that the TIVNFS-A has capability of nonlinear system identification and it has better performance than RFNN's results. In this example, we will use the TIVNFS-A to identify an actual system in chemistry and show the TIVNFS-A system. The Continuous Stirred Tank Reactor (CSTR) system was chosen as an example problem by application of identification. Figure 9 shows the illustration of CSTR system that is common chemical system. The process describes the reactions two reactants which are react and generate a compound in the tank. The plant of CSTR system is described by [29]

$$\dot{C}_a(t) = \frac{q(t)}{v}\left(C_{a0} - C_a(t)\right) - k_0 C_a(t) e^{-\frac{E}{RT(t)}} \tag{70}$$

$$\begin{aligned}\dot{T}(t) = {} & \frac{q(t)}{v}\left(T_0 - T(t)\right) - k_1 C_a(t) e^{-\frac{E}{RT(t)}} \\ & + k_2 q_c(t)\left(1 - e^{-\frac{k_3}{q_c(t)}}\right)\left(T_{c0} - T(t)\right)\end{aligned} \tag{71}$$

where k_1, k_2, and k_3 are thermodynamic and chemical as

$$k_1 = -\frac{\Delta H k_0}{\rho C_p},\ k_2 = \frac{\rho_c C_{pc}}{\rho C_p v} \text{ and } k_3 = \frac{h_a}{\rho_c C_{pc}}.$$

$C_a(t)$ and $T(t)$ are the concentration and temperature of produce in the tank. $q(t)$ and $q_c(t)$ are the flow rate of inlet feed and inlet coolant. Other parameters of above equations are fixed and have been described and determined in TABLE III.

For system identification, the flow rate of inlet feed and inlet coolant $q(t)$ and $q_c(t)$ as the inputs of TIVNFS-A. The concentration and temperature of produce $C_a(t)$ and $T(t)$ as the outputs of TIVNFS-A. Figure 10 shows the series-parallel scheme for CSTR system identification. In order to make more real simulation, we added the white noise in the inputs data. Since the inputs of the TIVNFS-A, $q(t)$ and $q_c(t)$ contain the white noise, therefore we use a simple filter before TIVNFS-A system. Figures 11 and 12 show the inputs and the desire outputs data for the CSTR system identification, respectively. Note that the non-smooth phenomenon of input and output signals exists since the white noises are added in the input signals.

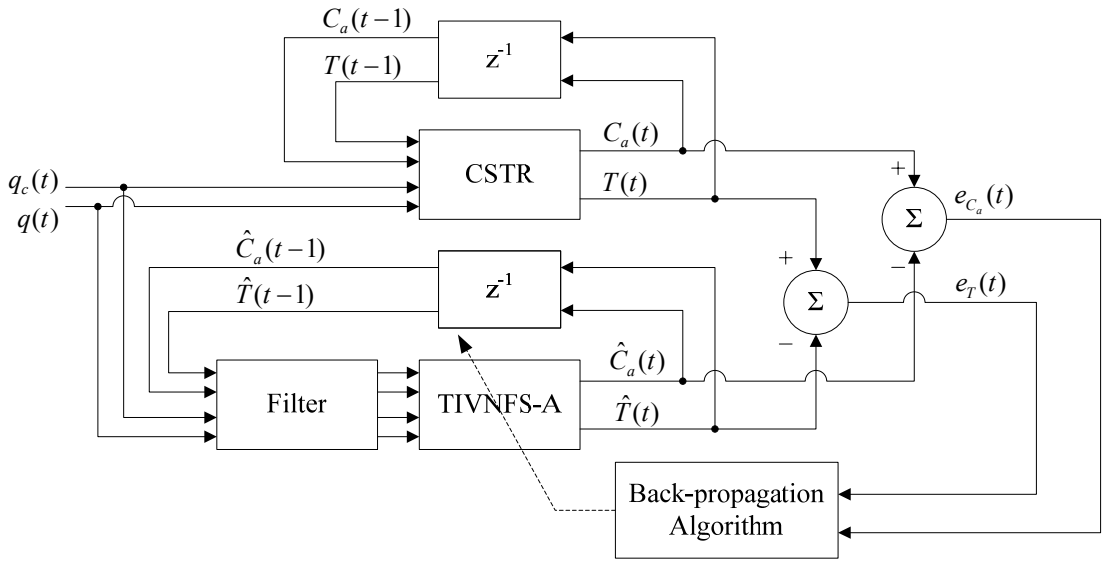

Figure 10: Series-parallel model with the TIVNFS-A for CSTR system identification.

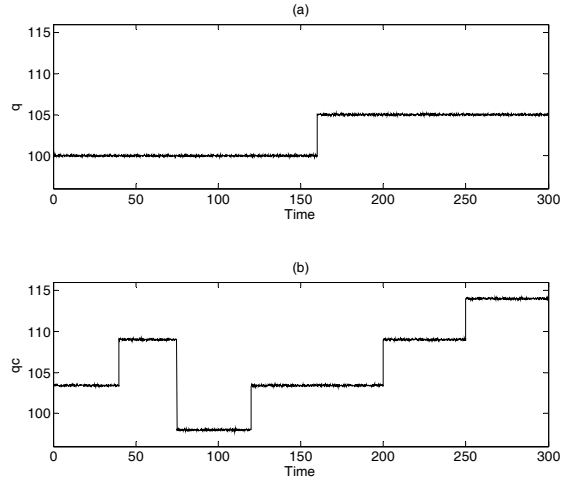

Figure11: Input of the flow rate for identification process (a) inlet feed and (b) inlet coolant.

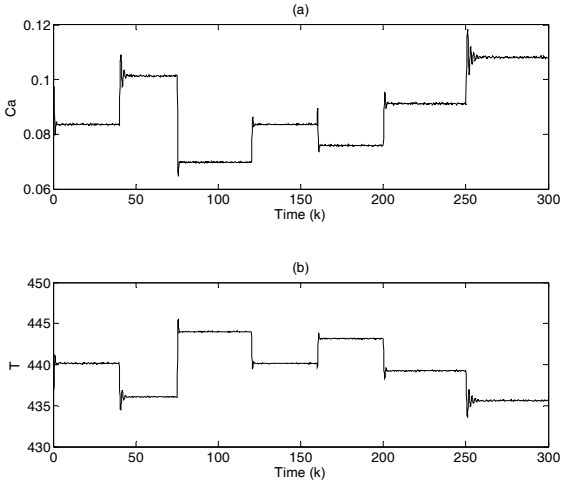

Figure12: Desire output for identification process, (a) the concentration of produce and (b) temperature of produce.

In this example, we chose the suitable parameters of TIVNFS-A and back-propagation algorithm and are following: The parameters of TIVNFS-A are $\overline{m}^l, \underline{m}^l, \overline{m}^r, \underline{m}^r$ that are chosen randomly between [0, 8], $\overline{\sigma}^l, \underline{\sigma}^l, \overline{\sigma}^r, \underline{\sigma}^r$ that are chosen randomly between [0, 25], and $\overline{\omega}, \underline{\omega}, c, s$ are chosen randomly between [-1, 1]. The numbers of the TIVNFS-A's rule is set to be 3, then the structure of TIVNFS-A is 4-12-3-6-4-2, and the numbers of parameter of TIVNFS-A is 180.

Figures 13 (a) and (b) show the results of the concentration and temperature of produce $C_a(t)$ and $T(t)$, respectively. Otherwise, Figure 14 shows the result of RMSE with 30 epochs. We can obtain that the performance of identification is accurate and well. In TABLE IV, we averaged the RMSE value using 10 independent runs, and it can be found that the performance is good and stable. According to above simulations of two illustration examples, we can obtain the TIVNFS-A systems have capable of system identification for actual system with multi-input-multi-output.

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 in nonlinear system identification. The TIVNFS-A consists of asymmetric membership functions and TSK-type consequent part to enhance the performance. The corresponding type reduction procedure has been simplified and integrated in the network layers to reduce the amount of computation in TIVNFS-A 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 stable learning mechanism.

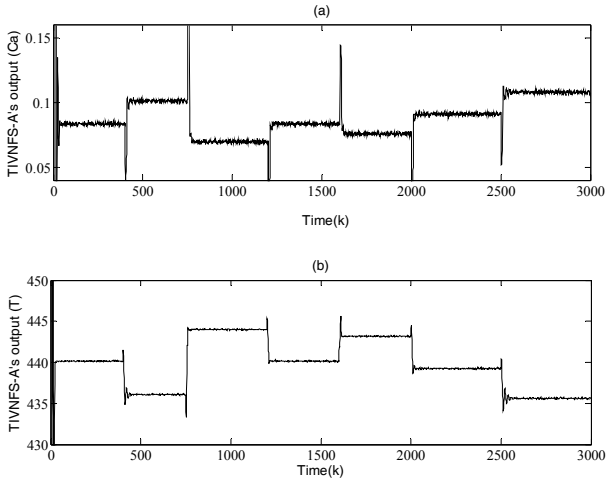

Figure 13: Identification results of CSTR system, (a) Concentration, $C_a(t)$; (b) Temperature, $T(t)$.

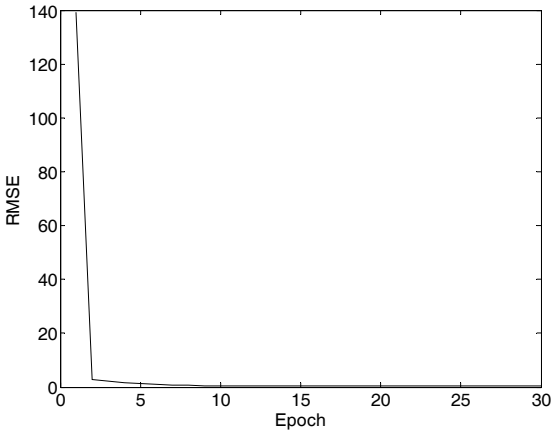

Figure 14: Results of RMSE value.

TABLE IV
THE RESULTS OF CSTR IDENTIFICATION USING TIVNFS-A IN 10 INDEPENDENT RUNS.

	TIVNFS-A
Best	0.42827
Average	0.62242
Worst	0.77573
Times(sec)	15.808

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] Y. H. Lee and C. H. Lee, "Stable Learning Mechanism for Novel Takagi-Sugeno-Kang Type Interval-valued Fuzzy Systems," *Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2011, IMECS 2011*, 16-18 March, 2011, Hong Kong, pp. 1-6.

[20] 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.

[21] 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.

[22] C. T. Lin and C. S. G. Lee, *Neural Fuzzy Systems*, Prentice Hall: Englewood Cliff, 1996.

[23] 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.

[24] G. Ligntbody and G. W. Irwin, "Nonlinear Control Structures Based on Embedded Neural System Models," *IEEE Transactions on Neural Networks*, Vol. 8, pp. 553-567, 1997.

[25] J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*, Upper Saddle River, Prentice-Hall, NJ, 2001.

[26] 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.

[27] 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.

[28] K. Salahshoor, A. S. Kamalabady, "On-line Multivariable Identification by Adaptive RBF Neural Networks Based on UKF Learning Algorithm," *Chinese Control and Decision Conference*, pp.4754-4759, 2008.

[29] L. S. Saoud and A. Khellaf, "Nonlinear Dynamic Systems Identification Based on Dynamic Wavelet Neural Units," *Neural Computing and Applications*, Vol. 19, pp. 997-1002, 2010.

[30] I. B. Turksen, "Interval Valued Fuzzy Sets based on Normal Forms," *Fuzzy Sets and Systems*, Vol. 20, Issue 2, pp. 191-210, 1986.
[31] 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.
[32] 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.
[33] Yu, X. H. et al., "Dynamic Learning Rate Optimization of the Back Propagation Algorithm," *IEEE Trans. On Neural Networks*, vol. 6, pp. 669-677, May 1995.

Ching-Hung Lee was born in Taipei County, Taiwan, R.O.C., in 1969. He received the B.S. and M.S. degree in Control Engineering from the National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 1992 and 1994, respectively, and the Ph.D. degree in Electrical and Control Engineering from the same University, in 2000. He is currently an Associate Professor of the Department of Electrical Engineering at Yuan Ze University. He is the Director for the Intelligent Control and Applications Laboratory. He is a chapter author of Recurrent Neural Network (I-Tech, 2008). He has authored numerous published journal papers in the area of intelligent control systems and applications. Dr. Lee received the Wu Ta-Yu Medal and Young Researcher Award in 2008 from the National Science Council, R.O.C. He also received the 2009 Youth Automatic Control Engineering Award from Chinese Automatic Control Society. He is also the recipient of the 2009 YZU Award Excellence in Research and 2009 YZU Award Excellence in Teaching from Yuan Ze University, R.O.C. His main research interests are fuzzy neural systems, neural network, evolutionary algorithm, signal processing, nonlinear control systems, image processing, and robotics control.

Yi-Han Lee was born in Taipei City, Taiwan, R.O.C., in 1988. He received the B.S. degree in electronic engineering at Huafan University, Taipei, Taiwan, R.O.C., in 2010. He is currently working toward the Master degree in electrical engineering at the Yuan Ze University. His research interests include neural network, algorithm, image process, and intelligent control.