We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Recurrent Interval Type-2 Fuzzy Neural Network Using Asymmetric Membership Functions

Ching-Hung Lee and Tzu-Wei Hu
Department of Electrical Engineering
Yuan-Ze University
Taiwan

1. Introduction

The fuzzy systems and control are regarded as the most widely used application of fuzzy logic systems in recent years (Jang, 1993; John & Coupland, 2007; Lin & Lee, 1006; Mendel, 2001; Wang, 1994). The structure of traditional fuzzy system models that is characterized by using type 1 fuzzy sets, which are defined on a universe of discourse, map an element of the universe of discourse onto a precise number in the unit interval [0, 1]. The concept of type-2 fuzzy sets was initially proposed by Zadeh as an extension of typical fuzzy sets (called type-1) (Zadeh, 1975). Mendel and Karnik developed a complete theory of interval type-2 fuzzy logic systems (iT2FLSs) (Karnik et al, 1999; Liang & Mendel, 2000; Mendel, 2001). Recently, T2FLSs have attracted more attention in many literatures and special issue of IEEE Transactions on Fuzzy systems (Baldwin & Karake, 2003; John & Coupland, 2007; Lee & Lin, 2005; Liang & Mendel, 2000; Mendel, 2001, Hagras, 2007; Ozen & Garibaldi, 2004; Pan et al, 2007; Wang et al, 2004).

T2FLSs are more complex than type-1 ones, the major difference being the present of type- is their antecedent and consequent sets. T2FLSs result better performance than type-1 Fuzzy Logic Systems (T1FLSs) on the applications of function approximation, modeling, and control. In addition, neural networks have found numerous practical applications, especially in the areas of prediction, classification, and control (Lee & Teng, 2000; Lin & Lee, 1996; Narendra & Parthasarathy, 1990). The main aspect of neural networks lies in the connection weights which are obtained by training process. Based on the advantages of T2FLSs and neural networks, the type-2 neural fuzzy systems are presented to handle the system uncertainty and reduce the rule number and computation (Castillo & Melin, 2004; Lee & Lin, 2005; Mendel, 2001; Pan et al, 2007; Wang et al, 2004). Besides, recurrent neural network has the advantages of store past information and speed up convergence (Lee & Teng, 2000).

The design of a fuzzy partition and rules engine normally affects system performance. To simplify the design procedure, we usually use the symmetric and fixed membership functions (MFs), such as Gaussian, triangular. However, a large rule number should be used to achieve the specified approximation accuracy (or result larger approximated error) (Lee & Teng, 2001; Lotfi & Tsoi, 1996). Several approaches have been introduced to optimize fuzzy MFs and choose an efficient scheme for structure and parameter learning. Nevertheless, asymmetric fuzzy MFs (AFMFs) has been discussed and analyzed for this problem (Baldwin

& Karake, 2003; Kim et al, 2003; Lee & Teng, 2001; Li et al, 2005; Lin & Ho, 2005; Ozen & Garibaldi, 2004; Pan et al, 2007). The results showed that using AFMFs can improve the approximation capability. According to the results above, our purpose is to introduce a recurrent interval type-2 fuzzy neural network with asymmetric membership functions (RiT2FNN-A). The asymmetric Gaussian function is a new type of membership function due to excellent approximation results. It also provides a fuzzy-neural network with higher flexibility to easily approach the optimum result more accurately. Literature (Lee & Pan, 2007; Pan et al, 2007) proposed that a T2FNN with AFMFs (T2FNN-A) can improve the system performance and obtain better approach ability. However, the structure of network was a static model. In this article, we proposed a combining interval type-2 fuzzy asymmetric membership functions with recurrent neural network system, called RiT2FNNA. The proposed RiT2FNN-A is a modified version of the T2FNN (Lee & Lin, 2005; Lee et al, 2003; Lee & Pan, 2007; Pan et al, 2007; Wang et al, 2004), which provides memory elements to capture system dynamic information (Lee & Teng, 2000). The RiT2FNN-A system capability for temporarily storing information allowed us to extend the application domain to include temporal problem. Simulations are shown to illustrate the effectiveness of the RiT2FNN-A system.

This article is organized as follows. Section 2 introduces the interval type-2 fuzzy neural systems and construction of interval type-2 AFMFs. The proposed RiT2FNN-A system is described in Section 3. Simulation results about handling nonlinear system identification is done and introduced in Section 4. Finally, conclusion is given.
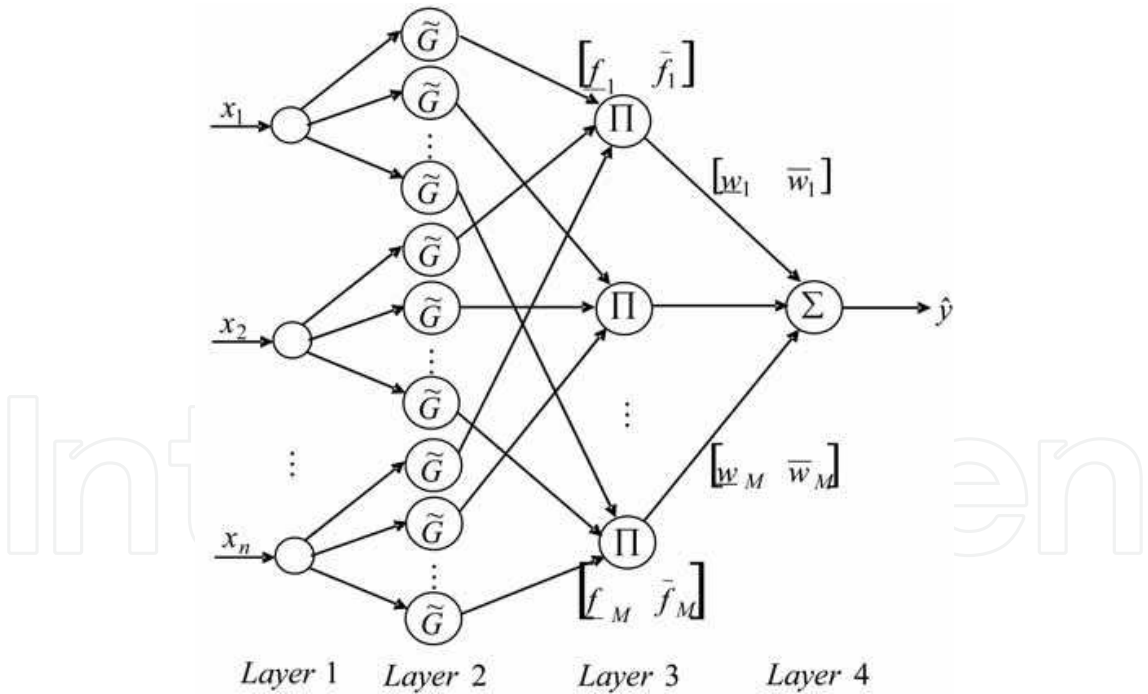

Figure1, Diagram of MISO T2FNN system with *M* fuzzy rules (Lee & Lin, 2005).

2. Interval type-2 fuzzy neural systems

The concept of type-2 fuzzy set was initially proposed as an extension of ordinary one (called type-1) by Zadeh (Zadeh 1975). In recent years, Mendel and Karnik have developed a

complete theory of T2FLSs (John & Coupland, 2007; Karnik et al, 1999; Lee & Lin, 2005; Mendel, 2001). These systems are as an extension of general FLSs (called type-1) which is characterized by IF-THEN rules (Lin & Lee, 1996). The computation of iT2FLSs is more complex than the T1FLSs because of the antecedent and consequent type-2 fuzzy sets (John & Coupland, 2007; Karnik et al, 1999; Mendel, 2001). In our previous results, we successfully constructed the T2FNN to identify the nonlinear system (Lee & Lin, 2005; Lee et al, 2003; Lee & Pan, 2007). They perform as well as the general T1FNNs, even better. In this section, we first introduce the interval type-2 fuzzy neural network (iT2FNN) systems, a type of fuzzy inference system in neural network structures, followed by the construction of interval type-2 AFMFs (iT2AFMFs) which is used to develop the recurrent interval type-2 fuzzy neural network (RiT2FNN).
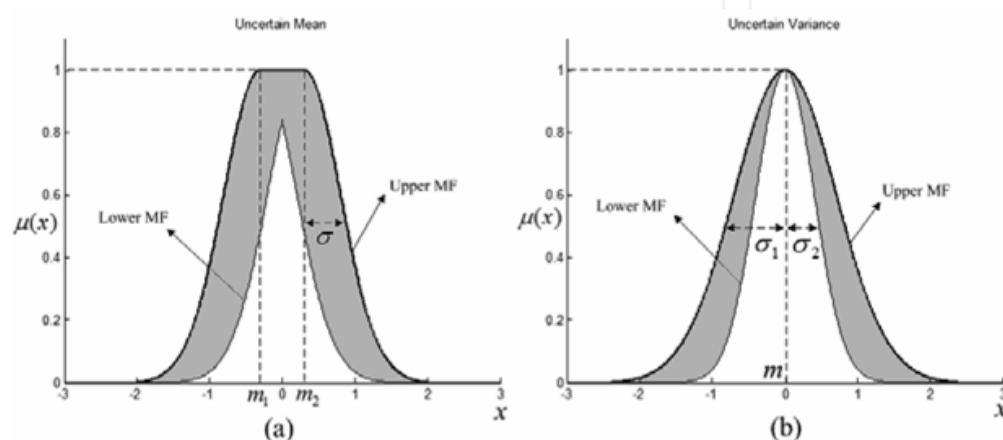

Figure 2. Symmetric interval type-2 fuzzy MFs: (a) Gaussian MF with uncertain mean and (b) Gaussian MF with uncertain variance.

2.1 Interval type-2 fuzzy neural network systems

In general, given an system input data set x_i , i=1, 2, …, n, and the desired output y_p , p=1, 2, …, m, the jth type-2 fuzzy rule has the form

$$\text{Rule } j: \text{ IF } x_1 \text{ is } \widetilde{G}_1^j \text{ and } \dots x_n \text{ is } \widetilde{G}_n^j$$
$$\text{THEN } y_1 \text{ is } \widetilde{w}_1^j \text{ and } \dots y_m \text{ is } \widetilde{w}_m^j, \tag{1}$$

where j is the number of rules, $\widetilde{G}_i^j$ represents the linguistic term of the antecedent part, $\widetilde{w}_p^j$ represents the real number of the consequent part; n and m are the numbers of the input and output dimensions, respectively. Based on the iT2FLSs, the construction of multi-inputsingle- output (MISO) type of the iT2FNN system is shown in Fig. 1 (Lee & Lin, 2005).

Obviously, it is a static model and the structure uses interval type-2 fuzzy sets ($\widetilde{G}$ and $\widetilde{w}$). Figure 2 shows a commonly used two-dimensional interval type-2 Gaussian MF. Figure 2(a) is an interval type-2 Gaussian MF with an interval mean in [m_1,m_2] and fixed variance σ, and Fig. 2(b) is an interval type-2 Gaussian MF with an interval variance in [σ$_1$, σ$_2$] and fixed mean m.

It can be found that the iT2FNN uses the interval type-2 fuzzy sets and it implements the FLS in a four layer neural network structure. Layer-1 nodes are input nodes representing

input linguistic variables, and layer-4 nodes are output nodes. The nodes in layer 2 are term nodes that act as T2MFs. All of the layer-3 nodes together formulate a fuzzy rule basis, and the links between layers 3 and 4 function as a connectionist inference engine. Herein, we introduce the iT2FNN system.

Layer 1: Input Layer

For the ith node of layer 1, the net input and the net output are represented as:

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

where $x_i^{(1)}$ represents the ith input to the ith node of layer 1. The subscript $_i$ denotes the ith input and the super-script $^{(1)}$ denotes the first layer.

Layer 2: Membership Layer

In this layer, each node performs a type-2 membership function (T2MF). Two kinds of T2MF are introduced (Liang & Mendel, 2000; Mendel, 2001). For case 1- Gaussian MFs with uncertain mean, shown in Fig. 2(a), we have

$$O_{ij}^{(2)} = \exp\left[-\frac{1}{2}\frac{\left(O_i^{(1)} - m_{ij}\right)^2}{\left(\sigma_{ij}\right)^2}\right] = \begin{cases} \overline{O}_{ij}^{(2)} & as \quad m_{ij} = \overline{m}_{ij} \\ \underline{O}_{ij}^{(2)} & as \quad m_{ij} = \underline{m}_{ij}. \end{cases} \tag{3}$$

Case 2- Gaussian MFs with uncertain variance, shown in Fig. 2(b), we have

$$O_{ij}^{(2)} = \exp\left[-\frac{1}{2}\frac{\left(O_i^{(1)} - m_{ij}\right)^2}{\left(\sigma_{ij}\right)^2}\right] = \begin{cases} \overline{O}_{ij}^{(2)} & as \quad \sigma_{ij} = \overline{\sigma}_{ij} \\ \underline{O}_{ij}^{(2)} & as \quad \sigma_{ij} = \underline{\sigma}_{ij}, \end{cases} \tag{4}$$

where m_{ij} and σ_{ij} represent the center (or mean) and the width (or variance), respectively The subscript $_{ij}$ indicates the jth term of the ith input $O_i^{(1)}$, where $j=1, \ldots, M$, and the superscript $_{(2)}$ means the secondary layer. Therefore, the output $O_{ij}^{(2)}$ is represented as $[\underline{O}_{ij}^{(2)} \ \overline{O}_{ij}^{(2)}]$.

Layer 3: Rule Layer

In this layer, the operation is chosen as simple PRODUCT operation, i.e.,

$$O_j^{(3)} = \prod_{i=1}^{n}\left(O_{ij}^{(2)}\right) = \begin{cases} \overline{O}_j^{(3)} = \prod_{i=1}^{n}\left(\overline{w}_{ij}^{(3)}\overline{O}_{ij}^{(2)}\right) \\ \underline{O}_j^{(3)} = \prod_{i=1}^{n}\left(\underline{w}_{ij}^{(3)}\underline{O}_{ij}^{(2)}\right), \end{cases} \tag{5}$$

where the weights $w_{ij}^{(3)}$ are assumed to be unity, and the subscript $_j$ indicates the jth rule, $j = 1, \ldots, M$, and the super-script $^{(3)}$ means the third layer. Thus, the output $O_{ij}^{(3)}$ is represented as $[\underline{O}_j^{(3)}, \overline{O}_j^{(3)}]$.

Layer 4: Output Layer

Links in this layer are used to implement the consequence matching, type-reduction and defuzzification (Lee & Lin, 2005; Mendel, 2001). Thus,

$$\hat{y} = O^{(4)} = \frac{O_R^{(4)} + O_L^{(4)}}{2}, \tag{6}$$

where

$$O_R^{(4)} = \sum_{j=1}^{M} \left(f_j^R \overline{w}_j^{(4)} \right) = \sum_{j=1}^{R} \left(\underline{O}_j^{(3)} \overline{w}_j^{(4)} \right) + \sum_{j=R+1}^{M} \left(\overline{O}_j^{(3)} \overline{w}_j^{(4)} \right), \tag{7}$$

$$O_L^{(4)} = \sum_{j=1}^{M} \left(f_j^L \underline{w}_j^{(4)} \right) = \sum_{j=1}^{L} \left(\overline{O}_j^{(3)} \underline{w}_j^{(4)} \right) + \sum_{j=L+1}^{M} \left(\underline{O}_j^{(3)} \underline{w}_j^{(4)} \right), \tag{8}$$

and

$$R = \underset{j \in [1, \cdots, M-1]}{\arg \ \max} \left(\overline{O}_R^{(4)} \right), \qquad L = \underset{j \in [1, \cdots, M-1]}{\arg \ \min} \left(\underline{O}_L^{(4)} \right).$$

In order to get $O_L^{(4)}$ and $O_R^{(4)}$, we first need to find coefficients R and L. Without loss of generality, we assume that the pre-computed $\overline{w}_j^{(4)}$ and $\underline{w}_j^{(4)}$ are arranged in ascending order, i.e., $\overline{w}_1^{(4)} \leq \overline{w}_2^{(4)} \leq \cdots \leq \overline{w}_M^{(4)}$ and $\underline{w}_1^{(4)} \leq \underline{w}_2^{(4)} \leq \cdots \leq \underline{w}_M^{(4)}$ (Mendel, 2001). Then,

R1: Compute $O_R^{(4)}$ in (7) by initial setting $f_j^R = \frac{1}{2} \left(\overline{O}_j^{(3)} + \underline{O}_j^{(3)} \right)$ for $j = 1, \ldots, M$, and let

$y_r \equiv O_R^{(4)}$.

R2: Find $R \left(1 \leq R \leq M-1 \right)$ such that $\overline{w}_R^{(4)} \leq y_r \leq \overline{w}_{R+1}^{(4)}$.

R3: Compute $O_R^{(4)}$ in (7) with $f_j^R = \underline{O}_j^{(3)}$ for $j \leq R$ and $f_j^R = \overline{O}_j^{(3)}$ for $j > R$, and let

$y_r' = O_R^{(4)}$.

R4: If $y_r' \neq y_r$, then go to step R5. If $y_r' = y_r$, then stop and set $O_R^{(4)} = y_r'$.

R5: Set y_r' to be y_r and return to step R2.

Subsequently, the computation of $O_L^{(4)}$ is similar to the above procedure.

L1: Compute $O_L^{(4)}$ in (8) by initial setting $f_j^L = \frac{1}{2} \left(\overline{O}_j^{(3)} + \underline{O}_j^{(3)} \right)$ for $i = 1, \cdots, M$, and let

$y_r \equiv O_L^{(4)}$.

L2: Find $L \left(1 \leq L \leq M-1 \right)$ such that $\underline{w}_L^{(4)} \leq y_l \leq \underline{w}_{L+1}^{(4)}$.

L3: Compute $O_L^{(4)}$ in (2.7) with $f_j^L = \overline{O}_j^{(3)}$ for $j \leq L$ and $f_j^L = \underline{O}_j^{(3)}$ for $j > L$, and let

$y_l' = O_L^{(4)}$.

L4: If $y_l' \neq y_l$, then go to step L5. If $y_l' = y_l$, then stop and set $O_L^{(4)} = y_l'$.

L5: Set y_l' to be y_l and return to step L2.

This five-step iterative procedure is called the Karnik-Mendel procedure (Liang & Mendel, 2000; Mendel, 2001).Thus, the input/output representation of iT2FNN system with uncertain mean is

$$\hat{y}\left(\overline{m}_{ij}, \underline{m}_{ij} \sigma_{ij}, \overline{w}_j, \underline{w}_j \right) = \frac{1}{2} \left[\sum_{j=1}^{R} \left(\underline{O}_j^{(3)} \overline{w}_j^{(4)} \right) + \sum_{k=R+1}^{M} \left(\overline{O}_k^{(3)} \overline{w}_k^{(4)} \right) + \sum_{j=1}^{L} \left(\overline{O}_j^{(3)} \underline{w}_j^{(4)} \right) + \sum_{k=L+1}^{M} \left(\underline{O}_k^{(3)} \underline{w}_k^{(4)} \right) \right]. \tag{9}$$

Similarly, the iT2FNN using T2MFs with uncertain variance can be simplified as (Lee & Lin, 2005)

$$\hat{y}\left(m_{ij}, \overline{\sigma}_{ij}, \underline{\sigma}_{ij}, w_j\right) = \frac{1}{2} \sum_{j=1}^{M} \left[\left(\underline{O}_j^{(3)} + \overline{O}_j^{(3)}\right) w_j^{(4)}\right]$$
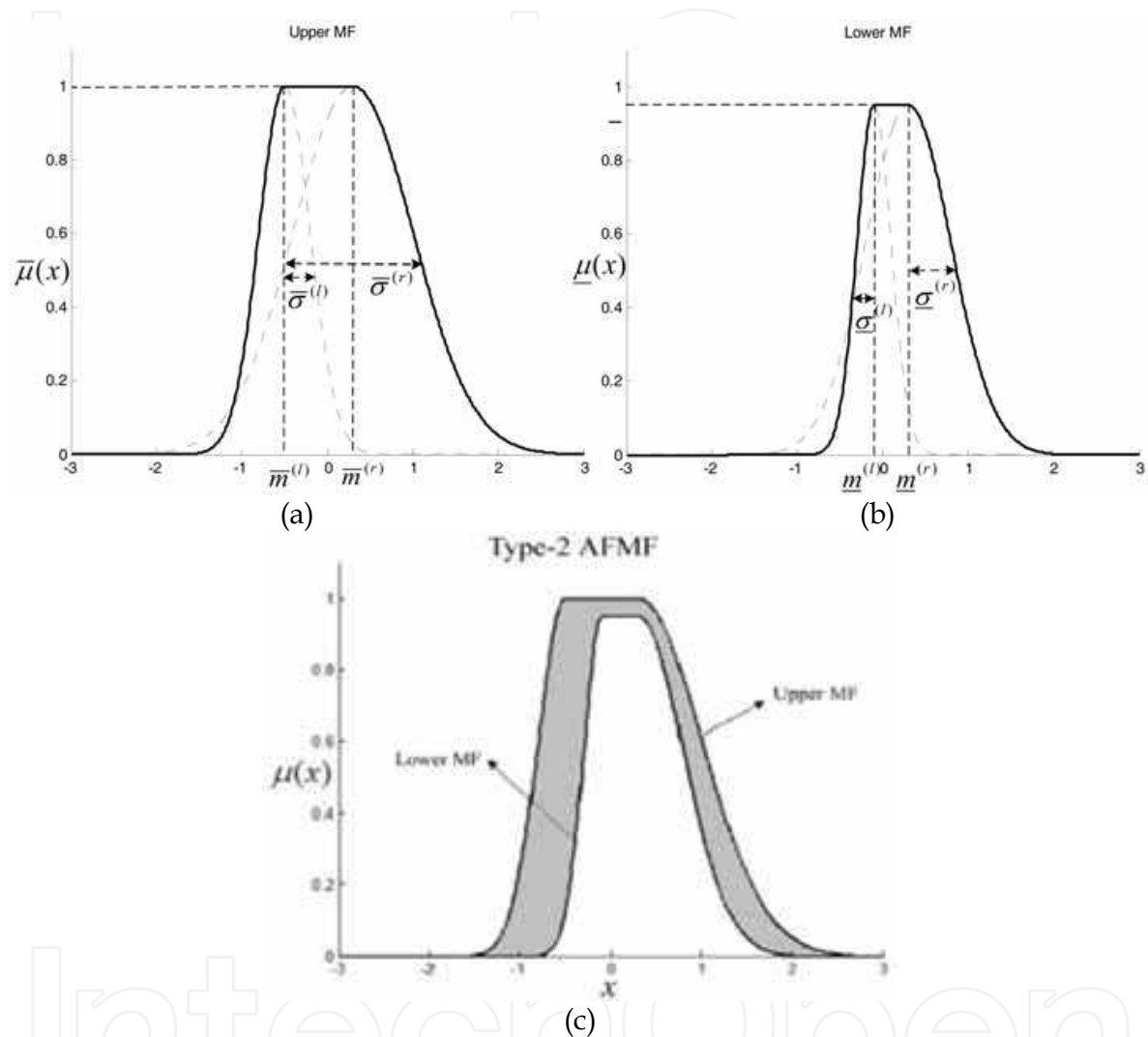
(10)

(a)

(b)

(c)

Figure 3. Construction of a type-2 AFMF: (a) upper MF (solid line), (b) lower MF (solid line), and (c) constructed iT2AFMF.

2.2 Construction of interval type-2 asymmetric fuzzy membership functions

The interval T2MFs of the precondition part discussed in this article are of asymmetric type, iT2AFMFs, as described below (see Fig. 3). Each MF is replaced by an asymmetric one constructed from parts of four Gaussian functions; that is, each upper and lower MF is constructed by two Gaussian MFs and one segment. Here we use the superscripts (l) and (r) to denote the left and right curves of a Gaussian MF. The parameters of lower and upper MFs are denoted by an underline (_) and bar (¯), respectively. Thus, the upper MF is constructed as

$$\overline{\mu}_{\tilde{G}}(x) = \begin{cases} \exp\left[-\frac{1}{2}\left(\dfrac{x-\overline{m}^{(l)}}{\overline{\sigma}^{(l)}} \right)^2 \right] & , \text{ for } x \le \overline{m}^{(l)} \\ \qquad\qquad 1 & , \text{ for } \overline{m}^{(l)} \le x \le \overline{m}^{(r)} \\ \exp\left[-\frac{1}{2}\left(\dfrac{x-\overline{m}^{(r)}}{\overline{\sigma}^{(r)}} \right)^2 \right] & , \text{ for } \overline{m}^{(r)} \le x \end{cases} \tag{11}$$

where $\overline{m}^{(l)}$ and $\overline{m}^{(r)}$ denote the means of two Gaussian MFs satisfying $\overline{m}^{(l)} \le \overline{m}^{(r)}$, and $\overline{\sigma}^{(l)}$ and $\overline{\sigma}^{(r)}$ denotes the deviation (i.e., width) of two Gaussian MFs. Figure 3(a) shows the upper iT2AFMF constructed using $\overline{m}^{(l)}$, $\overline{m}^{(r)}$, $\overline{\sigma}^{(l)}$, and $\overline{\sigma}^{(r)}$. Similarly, the lower asymmetric MF is defined as

$$\underline{\mu}_{\tilde{G}}(x) = \begin{cases} \underline{r}\cdot\exp\left[-\frac{1}{2}\left(\dfrac{x-\underline{m}^{(l)}}{\underline{\sigma}^{(l)}} \right)^2 \right] & , \text{ for } x \le \underline{m}^{(l)} \\ \qquad\qquad \underline{r} & , \text{ for } \underline{m}^{(l)} \le x \le \underline{m}^{(r)} \\ \underline{r}\cdot\exp\left[-\frac{1}{2}\left(\dfrac{x-\underline{m}^{(r)}}{\underline{\sigma}^{(r)}} \right)^2 \right] & , \text{ for } \underline{m}^{(r)} \le x \end{cases} \tag{12}$$

where $\underline{m}^{(l)} \le \underline{m}^{(r)}$ and $0.5 \le r \le 1$. The corresponding widths of the MFs are $\underline{\sigma}^{(l)}$ and $\underline{\sigma}^{(r)}$. To avoid unreasonable MFs, the following constrains are added:

$$\begin{cases} \overline{m}^{(l)} \le \underline{m}^{(l)} \le \underline{m}^{(r)} \le \overline{m}^{(r)} \\ \underline{\sigma}^{(l)} \le \overline{\sigma}^{(l)} \; , \;\; \underline{\sigma}^{(r)} \le \overline{\sigma}^{(r)} \\ 0.5 \le \underline{r} \le 1 \end{cases} \tag{13}$$

Figure 3(b) sketches the lower type-2 AFMF. The corresponding constructed iT2AFMF is shown in Fig. 3(c). This introduces the properties of uncertain mean and variance (Karnik et al, 1999). Additionally, we can construct other iT2AFMFs by tuning the parameters. The corresponding tuning algorithm is derived to improve system accuracy and approximation ability.

3. RiT2FNN-A system and learning

3.1 Network structure of RiT2FNN-A system
In this section, the structure of RiT2FNN-A system is introduced. The MISO case I considered here for convenience. The proposed RiT2FNN-A is modified and extended from previous results of literature (Juang, 2002; Karnik et al, 1999; Lee & Lin, 2005; Lee & Pan, 2007; Lin & Ho, 2005). It uses the interval asymmetric type-2 fuzzy sets and it implements the FLS in a five-layer neural network structure which contains four-layer forward network and a feedback layer. Layer-1 nodes are input nodes representing input linguistic variables,

and layer-4 nodes are output nodes representing output linguistic variables. The nodes in layer 2 are term nodes that act as MFs, where each membership node is responsible for mapping an input linguistic variable into a corresponding linguistic value for that variable. All of the layer-3 nodes together formulate a fuzzy rule basis, and the links between layers 3 and 4 function as a connectionist inference engine. The rule nodes reside in layer 3, and layer 5 is the recurrent part in type-2 fuzzy sets.

In general, given system input data x_i, $i = 1, 2,…, n$, the internal variables g_j, $j = 1, 2,…, M$, and the desired output y_p, $p = 1, 2,…, m$, the jth type-2 fuzzy rule for RiT2FNN-A has the form:

$$\text{Rule } j: \text{IF } x_1 \text{ is } \widetilde{G}_{1j} \text{ and } … x_n \text{ is } \widetilde{G}_{nj} \text{ and } g_j \text{ is } \widetilde{G}_j^F$$
$$\text{THEN} \quad y_1 \text{ is } \widetilde{w}_1^j \text{ and } … y_m \text{ is } \widetilde{w}_m^j, g_1 \text{ is } \widetilde{a}_1^j, g_2 \text{ is } \widetilde{a}_2^j, …, \text{ and } g_M \text{ is } \widetilde{a}_M^j. \tag{14}$$

where $\widetilde{G}$ represents the linguistic term of the antecedent part, $\widetilde{w}$ and $\widetilde{a}$ represents the interval real number of the consequent part; and M is the total rule number. Here the fuzzy MFs of the antecedent part $\widetilde{G}$ are of iT2AFMFs, which represent the different from typical Gaussian MFs. The diagram of RiT2FNN-A is shown in Fig. 4. Below we indicate the signal propagation and the operation functions of the nodes in each layer. In the following description, $O_i^{(l)}$ denotes the ith output of a node in the lth layer.
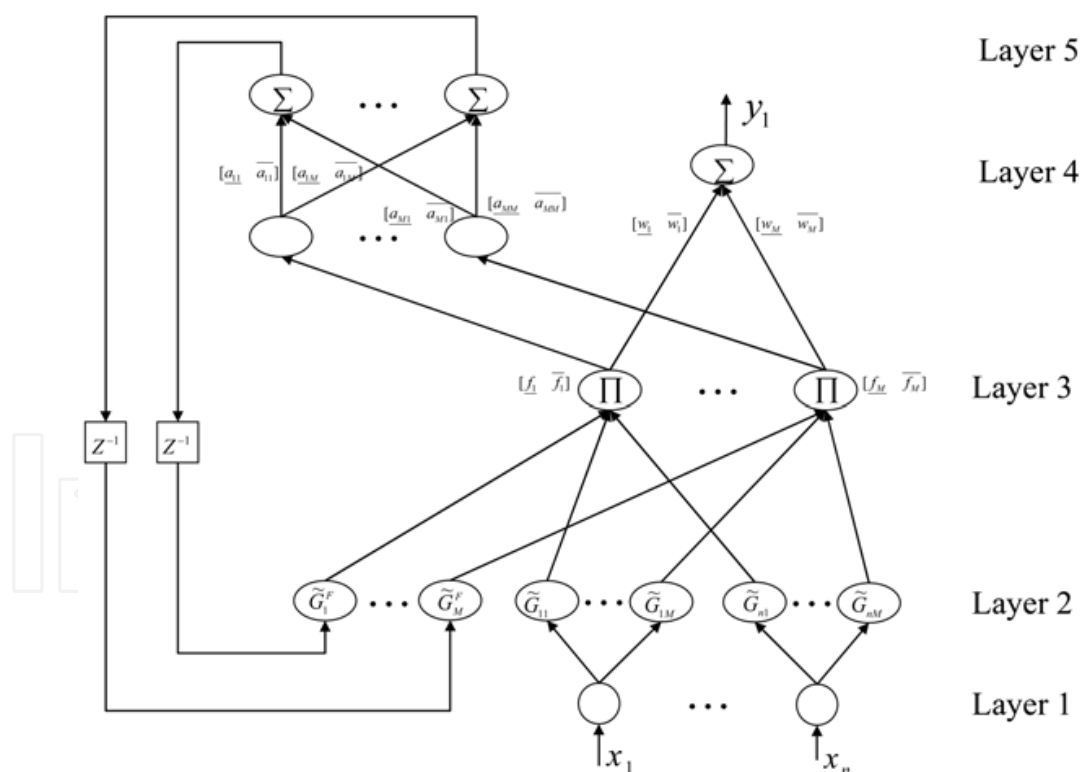

Fig. 4. Diagram of the proposed RiT2FNN-A system.

Layer 1: Input Layer

For the ith node of layer 1, the net input and output are represented as

$$O_j^{(1)} = x_i^{(1)}, \tag{15}$$

where $x_i^{(1)}$ represents the ith input to the jth node. Obviously, the nodes in this layer only transmit input values to the next layer directly.

Layer 2: Membership Layer

In layer 2, each node performs an iT2AFMF introduced by (11)–(13) (shown in Fig. 3). The following simplified notation is adopted

$$O_{ij}^{(2)} = \tilde{\mu}_{\tilde{G}_i^j}(O_i^{(1)}) \tag{16}$$

It is clear that there are two parts in this layer, regular nodes and feedback nodes. Their input are $O_j^{(1)}$ and $g_j(k)$. Therefore, for network input x_i, the output is

$$O_{ij}^{(2)} = \begin{bmatrix} \underline{O}_{ij}^{(2)} & \overline{O}_{ij}^{(2)} \end{bmatrix}^T = \begin{bmatrix} \underline{\mu}_{\tilde{G}_{ij}}(O_i^{(1)}) & \overline{\mu}_{\tilde{G}_{ij}}(O_i^{(1)}) \end{bmatrix}^T \tag{17}$$

For internal or feedback variable g_j,

$$O_j^{F(2)} = \begin{bmatrix} \underline{O}_j^{F(2)} & \overline{O}_j^{F(2)} \end{bmatrix}^T = \begin{bmatrix} \underline{\mu}_{\tilde{G}_j^F}(g_j(k)) & \overline{\mu}_{\tilde{G}_j^F}(g_j(k)) \end{bmatrix}^T, \tag{18}$$

where the subscript ij indicates the jth term of the ith input $O_i^{(1)}$. The superscript F indicates the feedback layer.

Layer 3: Rule Layer

The links in layer 3 are used to implement the antecedent matching, and these are equal to the work in the rule layer. Using the product t-norm, the firing strength associated with the jth rule is

$$\underline{f}^j = \underline{\mu}_{\tilde{G}_1^j}(x_1) * \cdots * \underline{\mu}_{\tilde{G}_n^j}(x_n) * \underline{\mu}_{\tilde{G}_j^F}(\cdot) \tag{19}$$

$$\overline{f}^j = \overline{\mu}_{\tilde{G}_1^j}(x_1) * \cdots * \overline{\mu}_{\tilde{G}_n^j}(x_n) * \overline{\mu}_{\tilde{G}_j^F}(\cdot) \tag{20}$$

where $\underline{\mu}(\cdot)$ and $\overline{\mu}(\cdot)$ are the lower and upper membership grades of $\tilde{G}(\cdot)$, respectively. Therefore, a simple product operation is used. Then, for the jth input rule node:

$$O_j^{(3)} = \begin{cases} \overline{O}_j^{F(2)} \prod_i w_{ij}^{(3)} \cdot \overline{O}_{ij}^{(2)} \\ \underline{O}_j^{F(2)} \prod_i w_{ij}^{(3)} \cdot \underline{O}_{ij}^{(2)} \end{cases} \tag{21}$$

where weights $w_{ij}^{(3)}$ are assumed to be unity and

$$O_j^{(3)} = \begin{bmatrix} \underline{O}_j^{(3)} & \overline{O}_j^{(3)} \end{bmatrix}^T = \begin{bmatrix} \underline{O}_j^{F(2)} \prod_{i=1}^n \underline{O}_{ij}^{(2)} & \overline{O}_j^{F(2)} \prod_{i=1}^n \overline{O}_{ij}^{(2)} \end{bmatrix}^T. \tag{22}$$

Layer 4: Output Layer

Without loss of generality, the consequent part of the iT2FLS is $\tilde{w}_j = \begin{bmatrix} \underline{w}_j & \overline{w}_j \end{bmatrix}^T$, $\underline{w}_j \leq \overline{w}_j$. The vector notations $\underline{w} = [\underline{w}_1 \quad \cdots \quad \underline{w}_M]^T$ and $\overline{w} = [\overline{w}_1 \quad \cdots \quad \overline{w}_M]^T$ are used for clarity. The

remaining works are type reduction and defuzzification. For type reduction, we should calculate the lower and upper bounds [y_l, y_r] (Karnik et al, 1999; Mendel, 2001). Modifying from the Karnik-Mendel procedure (Karnik et al, 1999; Mendel, 2001), let

$$O_{TR}^{(4)} = \int_{w_i \in [\underline{w}_1, \overline{w}_1]} \cdots \int_{w_M \in [\underline{w}_M, \overline{w}_M]} \int_{f_1 \in [\underline{f}_1, \tilde{f}_1]} \cdots \int_{f_M \in [\underline{f}_M, \tilde{f}_M]} 1 \bigg/ \sum_{i=1}^{M} f_i w_i . \tag{23}$$

Note that the normalization ($\sum_{i=1}^{M} f_i$) is removed here to simplify the type reduction procedure, computation, and the derivation of the learning algorithm by the gradient method. We denote the maximum and minimum of $\sum_{i=1}^{M} f_i w_i$ as $\overline{O}^{(4)}$ and $\underline{O}^{(4)}$,

$$\underline{O}^{(4)} = \underline{w}^T f_L = \sum_{j=1}^{L} \left(\overline{O}_j^{(3)} \underline{w}_j \right) + \sum_{j=L+1}^{M} \left(\underline{O}_j^{(3)} \underline{w}_j \right) \tag{24}$$

$$\overline{O}^{(4)} = \overline{w}^T f_R = \sum_{j=1}^{R} \left(\underline{O}_j^{(3)} \overline{w}_j \right) + \sum_{j=R+1}^{M} \left(\overline{O}_j^{(3)} \overline{w}_j \right), \tag{25}$$

where

$$f_L = \left[\tilde{f}_1, \cdots, \tilde{f}_L, \underline{f}_{L+1}, \cdots, \underline{f}_M \right]^T = \left[\overline{O}_1^{(3)}, \cdots, \overline{O}_L^{(3)}, \underline{O}_{L+1}^{(3)}, \cdots, \underline{O}_M^{(3)} \right]^T, \tag{26}$$

$$f_R = \left[\underline{f}_1, \cdots, \underline{f}_R, \tilde{f}_{R+1}, \cdots, \tilde{f}_M \right]^T = \left[\underline{O}_1^{(3)}, \cdots, \underline{O}_R^{(3)}, \overline{O}_{R+1}^{(3)}, \cdots, \overline{O}_M^{(3)} \right]^T . \tag{27}$$

It is obvious that R and L should be calculated first. The weights are arranged in order as $\underline{w}_1 \leq \underline{w}_2 \leq \cdots \underline{w}_M$ and $\overline{w}_1 \leq \overline{w}_2 \leq \cdots \overline{w}_M$. According to the Karnik-Mendel procedure (Karnik et al, 1999; Liang & Mendel, 2000; Mendel, 2001), L and R are

$$L = \arg_{j \in [1, \cdots, M-1]} \min \left(\underline{O}^{(4)} \right), \qquad R = \arg_{j \in [1, \cdots, M-1]} \max \left(\overline{O}^{(4)} \right). \tag{28}$$

According to the above introduction, only the minimum of $\underline{O}^{(4)}$ and the maximum of $\overline{O}^{(4)}$ should be calculated; which therefore simplifies the type-reduction computation. Finally, the crisp output is

$$O^{(4)} = \frac{\underline{O}^{(4)} + \overline{O}^{(4)}}{2}. \tag{29}$$

Layer 5: Feedback Layer

This layer contains the context nodes, which is used to produce the internal variable $O_j^{(5)}$. Each rule is associated with a particular internal variable. Hence, the number of the context nodes is equal to the number of rules. The same operations (type-reduction and defuzzifcation) as layer 4 are performed here.

$$g_j(k+1) = O_j^{(5)}(k+1) = \frac{1}{2} \left[\underline{O}_j^{(5)}(k+1) + \overline{O}_j^{(5)}(k+1) \right] \tag{30}$$

$$\underline{O}_j^{(5)}(k+1) = \underline{a}_j^T f_L = \sum_{h=1}^{L_j^F} \left(\overline{O}_h^{(3)} \underline{a}_{jh} \right) + \sum_{h=L_j^F+1}^{M} \left(\underline{O}_h^{(3)} \underline{a}_{jh} \right) \tag{31}$$

$$\overline{O}_j^{(5)}(k+1) = \overline{a}_j^{-T} f_R = \sum_{h=1}^{R_j^F} \left(\underline{O}_h^{(3)} \overline{a}_{jh} \right) + \sum_{h=R_j^F+1}^{M} \left(\overline{O}_h^{(3)} \overline{a}_{jh} \right) \tag{32}$$

$$L_j^F = \arg_{j \in [1,\cdots,M-1]} \min \left(\underline{O}_j^{(5)}(k+1) \right) \tag{33}$$

$$R_j^F = \arg_{j \in [1,\cdots,M-1]} \max \left(\overline{O}_j^{(5)}(k+1) \right). \tag{34}$$

Note that the delayed value of g_j is fed into layer 2, and it acts as an input variable to the precondition part of a rule. Each fuzzy rule has the corresponding internal variable g_j which is used to decide the influence degree of temporal history to the current rule.

3.2 Learning algorithm for RiT2FNN-A

The gradient descent method is adopted to derive learning algorithm of the RiT2FNN-A system. For clarification, we consider the single-output system and define the error cost function as

$$E(k) = \frac{1}{2}[y_d(k) - \hat{y}(k)]^2 \tag{35}$$

where y_d is the desired output and $\hat{y}$ is the RiT2FNN-A's output. Using the gradient descent algorithm, the parameters updated law is

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \Delta \mathbf{W}(k) = \mathbf{W}(k) + \eta \left(-\frac{\partial E(k)}{\partial \mathbf{W}(k)} \right) \tag{36}$$

in which η is the learning rate ($0 < \eta \le 1$). $\mathbf{W} = [\mathbf{W}_w \quad \underline{\mathbf{W}} \quad \mathbf{W}^F \quad \overline{\mathbf{W}} \quad \overline{\mathbf{W}}^F \quad \mathbf{W}_a \quad \underline{r} \quad \underline{r}^F]$ are the adjustable parameters, where $\mathbf{W}_w$ is consequent weights, $\underline{\mathbf{W}}$ and $\mathbf{W}^F$ are parameters of lower MFs, $\overline{\mathbf{W}}$ and $\overline{\mathbf{W}}^F$ are upper MFs parameters, $\mathbf{W}_a$ is parameter in feedback layer, and $\underline{r}$ and $\underline{r}^F$ are the column vectors, i.e.,

$$\mathbf{W}_w = \begin{bmatrix} \underline{w} & \overline{w} \end{bmatrix}^T \tag{37}$$

$$\mathbf{W}_a = \begin{bmatrix} \underline{a} & \overline{a} \end{bmatrix}^T \tag{38}$$

$$\underline{\mathbf{W}} = \begin{bmatrix} \underline{m}^{(l)} & \underline{m}^{(r)} & \underline{\sigma}^{(l)} & \underline{\sigma}^{(r)} \end{bmatrix}^T \tag{39}$$

$$\overline{\mathbf{W}} = \begin{bmatrix} \overline{m}^{(l)} & \overline{m}^{(r)} & \overline{\sigma}^{(l)} & \overline{\sigma}^{(r)} \end{bmatrix}^T \tag{40}$$

$$\underline{\mathbf{W}}^F = \left[\underline{m}^{F(l)} \quad \underline{m}^{F(r)} \quad \underline{\sigma}^{F(l)} \quad \underline{\sigma}^{F(r)}\right]^T \tag{41}$$

$$\overline{\mathbf{W}}^F = \left[\overline{m}^{F(l)} \quad \overline{m}^{F(r)} \quad \overline{\sigma}^{F(l)} \quad \overline{\sigma}^{F(r)}\right]^T. \tag{42}$$

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

$$\frac{\partial E(k)}{\partial \mathbf{W}(k)} = \frac{\partial E(k)}{\partial \hat{y}(k)}\frac{\partial \hat{y}(k)}{\partial \mathbf{W}(k)} = -\left[y_d(k) - \hat{y}(k)\right]\frac{\partial \hat{y}(k)}{\partial \mathbf{W}(k)} \tag{43}$$

Thus, (36) can be rewritten as

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \eta e(k)\frac{\partial \hat{y}(k)}{\partial \mathbf{W}(k)} \tag{44}$$

where $e(k) = y_d(k) - \hat{y}(k)$. The remaining work involves finding the corresponding partial derivatives with respect to each parameter.

Observing equation (24) and if $j \le L$, only the term of $\sum_{j=1}^{L}\left(\overline{O}_j^{(3)}\underline{w}_j\right)$ should be considered, and only consider $\sum_{j=L+1}^{M}\left(\underline{O}_j^{(3)}\underline{w}_j\right)$ if $j > L$. Moreover, we consider $\sum_{j=1}^{R}\left(\underline{O}_j^{(3)}\overline{w}_j\right)$ if $j \le R$ in (25), as well as $\sum_{j=R+1}^{M}\left(\overline{O}_j^{(3)}\overline{w}_j\right)$ where $j > R$. Thus, we should notice the values of j, R, and L in deriving the update laws.

In order to avoid the unnecessary tuning, we must also consider the firing regions of MFs for input variable xi. For example, considering an upper MF as shown in Fig. 5, region (I)- $x_i \le \overline{m}_{ij}^{(l)}$, only $\overline{m}_{ij}^{(l)}$ and $\overline{\sigma}_{ij}^{(l)}$ are updated; region (II)- $\overline{m}_{ij}^{(r)} \le x_i$, only $\overline{m}_{ij}^{(r)}$ and $\overline{\sigma}_{ij}^{(r)}$ must be updated as well. Finally, region (III)- $\overline{m}_{ij}^{(l)} < x_i < \overline{m}_{ij}^{(r)}$, nothing should be done. Therefore, we can tune one side of MF for each training pattern. The results of lower MFs are the same as above discussion. Besides, parameter r must be updated for all three regions. Owing the recurrent property, the real time recurrent learning algorithm (RTRL) is used.
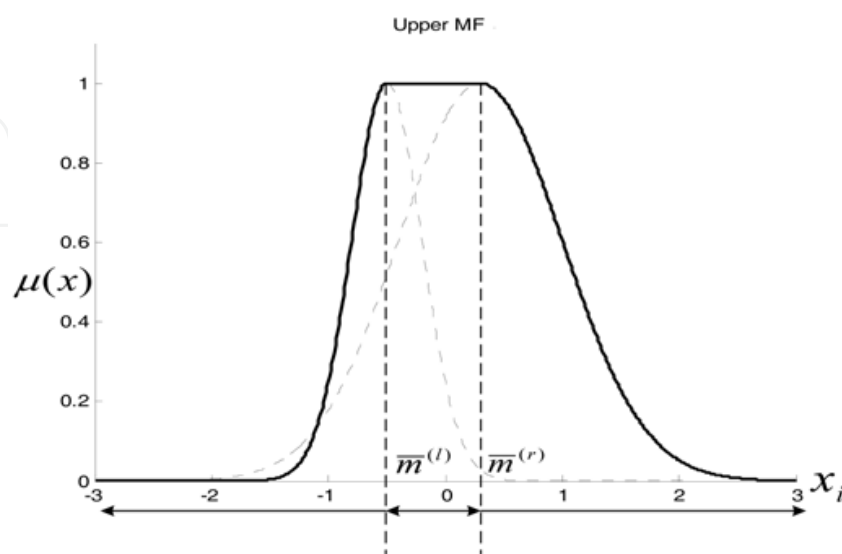

Fig. 5. Definitions of firing regions of input variable xi (upper MF).

By the gradient method, we derive the parameters update laws. Consider equations (24), (25), and (29), the output of RiT2FNN-A is rewritten as

$$O^{(4)} = \frac{1}{2}\left[\underline{O}_1^{(4)} + \overline{O}_1^{(4)} \right] = \frac{1}{2}\left[\sum_{j=1}^{L}\left(\overline{O}_j^{(3)}\underline{w}_j\right) + \sum_{j=L+1}^{M}\left(\underline{O}_j^{(3)}\underline{w}_j\right) + \sum_{j=1}^{R}\left(\underline{O}_j^{(3)}\overline{w}_j\right) + \sum_{j=R+1}^{M}\left(\overline{O}_j^{(3)}\overline{w}_j\right) \right]. \tag{45}$$

From equations (36) and (44), our major work is to find the partial derivation of RiT2FNN-A with respect to each parameter which can be obtained using the chain rule. We will show the update rule of $\mathbf{W}_w$ and $\overline{\mathbf{W}}$ only. Other parameter's updated rule can be derived the same way and are omitted.

-*Parameters* $\mathbf{W}_w$

$$\begin{bmatrix} \underline{w}_j(k+1) \\ \overline{w}_j(k+1) \end{bmatrix} = \begin{bmatrix} \underline{w}_j(k) \\ \overline{w}_j(k) \end{bmatrix} + \eta_w e(k)\begin{bmatrix} f_{L,j} \\ f_{R,j} \end{bmatrix} \tag{46}$$

where f_{Lj}, and f_{Rj}, are introduced previously in (26) and (27), and η_w is the corresponding learning rate.

-*Parameters* $\overline{\mathbf{W}}$

Region (I): $x_i \le \overline{m}_{ij}^{(l)}$

$$\begin{bmatrix} \overline{m}_{ij}^{(l)}(k+1) \\ \overline{\sigma}_{ij}^{(l)}(k+1) \end{bmatrix} = \begin{bmatrix} \overline{m}_{ij}^{(l)}(k) \\ \overline{\sigma}_{ij}^{(l)}(k) \end{bmatrix} + \frac{1}{2}\overline{\eta}e(k)\begin{cases} \underline{w}_j\begin{bmatrix} \overline{P}_{ij}^{(l)} + \overline{\omega}_j^{F(l)} \times \overline{H}_{jh}^{P(l)} \\ \overline{Q}_{ij}^{(l)} + \overline{\omega}_j^{F(l)} \times \overline{H}_{jh}^{Q(l)} \end{bmatrix}, & \text{if } j \le L \\[3ex] \overline{w}_j\begin{bmatrix} \overline{P}_{ij}^{(l)} + \overline{\omega}_j^{F(l)} \times \overline{H}_{jh}^{P(l)} \\ \overline{Q}_{ij}^{(l)} + \overline{\omega}_j^{F(l)} \times \overline{H}_{jh}^{Q(l)} \end{bmatrix}, & \text{if } j > R \end{cases} \tag{47}$$

where $\overline{\eta}$ denotes the corresponding learning rate,

$$\overline{P}_{ij}^{(l)} = \overline{O}_j^{(3)}\frac{\left(x_i - \overline{m}_{ij}^{(l)}\right)}{\left(\overline{\sigma}_{ij}^{(l)}\right)^2}, \quad \overline{Q}_{ij}^{(l)} = \overline{O}_j^{(3)}\frac{\left(x_i - \overline{m}_{ij}^{(l)}\right)^2}{\left(\overline{\sigma}_{ij}^{(l)}\right)^3},$$

$$\overline{\omega}_j^{F(l)} = \overline{O}_j^{(3)}\frac{\left(g_j - \overline{m}_j^{F(l)}\right)}{\left(\overline{\sigma}_j^{F(l)}\right)^2},$$

$$\overline{H}_{jh}^{P(l)} = \sum_{h=1}^{M}\overline{C}_{jh}^{F}\left(\overline{P}_{ij}^{(l)}(k-1) + \omega_h^{F(l)}(k-1)\cdot\overline{H}_{jh}^{P(l)}(k-1)\right),$$

$$\overline{H}_{jh}^{Q(l)} = \sum_{h=1}^{M}\overline{C}_{jh}^{F}\left(\overline{Q}_{ij}^{(l)}(k-1) + \overline{\omega}_j^{F(l)}(k-1)\cdot\overline{H}_{jh}^{Q(l)}(k-1)\right), \text{ and}$$

$$\overline{C}_{jh}^{F} = \begin{cases} 0.5\times\underline{a}_{jh}, & \text{for } (h \le L_j^F) \text{ and } (h \le R_j^F) \\ 0.5\times\overline{a}_{jh}, & \text{for } (h > L_j^F) \text{ and } (h > R_j^F) \\ 0, & \text{for } (h > L_j^F) \text{ and } (h \le R_j^F) \\ 0.5\times(\underline{a}_{jh}+\overline{a}_{jh}), & \text{else.} \end{cases}$$

Region (II): $\overline{m}_{ij}^{(r)} \leq x_i$

$$\begin{bmatrix} \overline{m}_{ij}^{(r)}(k+1) \\ \overline{\sigma}_{ij}^{(r)}(k+1) \end{bmatrix} = \begin{bmatrix} \overline{m}_{ij}^{(r)}(k) \\ \overline{\sigma}_{ij}^{(r)}(k) \end{bmatrix} + \frac{1}{2}\overline{\eta}e(k) \begin{cases} \underline{w}_j \begin{bmatrix} \overline{P}_{ij}^{(r)} + \overline{\omega}_j^{F(r)} \times \overline{H}_{jh}^{P(r)} \\ \overline{Q}_{ij}^{(r)} + \overline{\omega}_j^{F(r)} \times \overline{H}_{jh}^{Q(r)} \end{bmatrix}, \text{ if } j \leq L \\ \overline{w}_j \begin{bmatrix} \overline{P}_{ij}^{(r)} + \overline{\omega}_j^{F(r)} \times \overline{H}_{jh}^{P(r)} \\ \overline{Q}_{ij}^{(r)} + \overline{\omega}_j^{F(r)} \times \overline{H}_{jh}^{Q(r)} \end{bmatrix}, \text{ if } j > R \end{cases} \qquad (48)$$

where

$$\overline{P}_{ij}^{(r)} = \overline{O}_j^{(3)} \frac{\left(x_i - \overline{m}_{ij}^{(r)}\right)}{\left(\overline{\sigma}_{ij}^{(r)}\right)^2}, \overline{Q}_{ij}^{(r)} = \overline{O}_j^{(3)} \frac{\left(x_i - \overline{m}_{ij}^{(r)}\right)^2}{\left(\overline{\sigma}_{ij}^{(r)}\right)^3},$$

$$\overline{\omega}_j^{F(r)} = \overline{O}_j^{(3)} \frac{\left(g_j - \overline{m}_j^{F(r)}\right)}{\left(\overline{\sigma}_j^{F(r)}\right)^2},$$

$$\overline{H}_{jh}^{P(r)} = \sum_{h=1}^{M} \overline{C}_{jh}^F \left(\overline{P}_{ij}^{(r)}(k-1) + \omega_j^{F(r)}(k-1) \cdot H_{jh}^{P(r)}(k-1)\right), \text{ and}$$

$$\overline{H}_{jh}^{Q(r)} = \sum_{h=1}^{M} \overline{C}_{jh}^F \left(\overline{Q}_{ij}^{(r)}(k-1) + \overline{\omega}_j^{F(r)}(k-1) \cdot \overline{H}_{jh}^{Q(r)}(k-1)\right).$$

Region (III): $\overline{m}_{ij}^{(l)} < x_i < \overline{m}_{ij}^{(r)}$

$$\begin{bmatrix} \overline{m}_{ij}^{(l)}(k+1) \\ \overline{\sigma}_{ij}^{(l)}(k+1) \\ \overline{m}_{ij}^{(r)}(k+1) \\ \overline{\sigma}_{ij}^{(r)}(k+1) \end{bmatrix} = \begin{bmatrix} \overline{m}_{ij}^{(l)}(k) \\ \overline{\sigma}_{ij}^{(l)}(k) \\ \overline{m}_{ij}^{(r)}(k) \\ \overline{\sigma}_{ij}^{(r)}(k) \end{bmatrix}. \qquad (49)$$

Note that $\overline{H}_{jh}^{P(l)}, \overline{H}_{jh}^{P(r)}, \overline{H}_{jh}^{Q(l)},$ and $\overline{H}_{jh}^{Q(r)}$ are recurrent factors and equal to zero initially and are reset to zero after a period of time. $\overline{C}_{jh}^F$ is the recurrent weighting factor.

By using the Lyapunov stability approach, we have the following convergence theorem.

Theorem 1: Let $[\eta_w \quad \underline{\eta} \quad \overline{\eta} \quad \eta_r]$ be the learning rates of the tuning parameters for RiT2FNNA The asymptotic convergence of RiT2FNN-A is guaranteed if proper learning rates $[\eta_w \quad \underline{\eta} \quad \overline{\eta} \quad \eta_r]$ are chosen satisfying the following condition

$$\left(\lambda_w + \underline{\lambda} + \overline{\lambda} + \lambda_r + \underline{\lambda}_a + \overline{\lambda}_a + \underline{\lambda}^F + \overline{\lambda}^F + \lambda_{r^r}\right) < 2 \qquad (50)$$

where

$$\lambda_w = \eta_w \left[\frac{\partial \hat{y}(k)}{\partial \mathbf{W}_w}\right]^T \left[\frac{\partial \hat{y}(k)}{\partial \mathbf{W}_w}\right] > 0, \underline{\lambda} = \underline{\eta} \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}}\right]^T \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}}\right] > 0$$

$$\overline{\lambda} = \overline{\eta} \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}}\right]^T \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}}\right] > 0, \lambda_r = \eta_r \left[\frac{\partial \hat{y}(k)}{\partial \underline{r}}\right]^T \left[\frac{\partial \hat{y}(k)}{\partial \underline{r}}\right] > 0$$

$$\underline{\lambda}_a = \eta_a \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}_a} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}_a} \right] > 0, \overline{\lambda}_a = \eta_a \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}_a} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}_a} \right] > 0$$

$$\underline{\lambda}^F = \underline{\eta} \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}^F} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}^F} \right] > 0, \overline{\lambda}^F = \overline{\eta} \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}^F} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}^F} \right] > 0$$

$$\lambda_{\underline{r}^F} = \eta_r \left[\frac{\partial \hat{y}(k)}{\partial \underline{r}^F} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \underline{r}^F} \right] > 0.$$

Proof:

First, we define the Lyapunov function as follows:

$$V(k) = \frac{1}{2} \left[y_d(k) - \hat{y}(k) \right]^2 = \frac{1}{2} e^2(k) \tag{51}$$

where $\hat{y}(k)$ is RIT2FNN-A's system output, $y_d(k)$ is desired output and $e(k)$ denotes the approximated error. Thus, the change of V(k) is

$$\Delta V(k) = \frac{1}{2} \left[e^2(k+1) - e^2(k) \right] = \frac{1}{2} \left[e(k+1) + e(k) \right] \left[e(k+1) - e(k) \right]. \tag{52}$$

The error difference due to the learning can be represented by

$$\Delta e(k) = e(k+1) - e(k) \approx \left[\frac{\partial e(k)}{\partial \mathbf{W}} \right]^T \Delta \mathbf{W} \tag{53}$$

where $\left[\frac{\partial e(k)}{\partial \mathbf{W}} \right]^T = \left[\frac{\partial e(k)}{\partial \mathbf{W}_w} \quad \frac{\partial e(k)}{\partial \underline{\mathbf{W}}} \quad \frac{\partial e(k)}{\partial \overline{\mathbf{W}}} \quad \frac{\partial e(k)}{\partial \underline{r}} \quad \frac{\partial e(k)}{\partial \underline{\mathbf{W}}_a} \quad \frac{\partial e(k)}{\partial \overline{\mathbf{W}}_a} \quad \frac{\partial e(k)}{\partial \underline{\mathbf{W}}^F} \quad \frac{\partial e(k)}{\partial \overline{\mathbf{W}}^F} \quad \frac{\partial e(k)}{\partial \underline{r}^F} \right]^T$,

$\Delta \mathbf{W}^T = [\Delta \mathbf{W}_w \; \Delta \underline{\mathbf{W}} \; \Delta \overline{\mathbf{W}} \; \Delta r \; \Delta \underline{\mathbf{W}}_a \; \Delta \overline{\mathbf{W}}_a \; \Delta \underline{\mathbf{W}}^F \; \Delta \overline{\mathbf{W}}^F \; \Delta r^F]$

$$\Delta \mathbf{W}_w = e(k) \eta_w \left[\frac{\partial \hat{y}(k)}{\partial \mathbf{W}_w} \right]^T = e(k) \eta_w \begin{bmatrix} \frac{\partial \hat{y}(k)}{\partial w} \\ \frac{\partial \hat{y}(k)}{\partial \overline{w}} \end{bmatrix}, \qquad \Delta \underline{\mathbf{W}} = e(k) \underline{\eta} \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}} \right]^T = e(k) \underline{\eta} \begin{bmatrix} \frac{\partial \hat{y}(k)}{\partial \underline{m}^{(l)}} \\ \frac{\partial \hat{y}(k)}{\partial \underline{m}^{(r)}} \\ \frac{\partial \hat{y}(k)}{\partial \underline{\sigma}^{(l)}} \\ \frac{\partial \hat{y}(k)}{\partial \underline{\sigma}^{(r)}} \end{bmatrix},$$

$$\Delta \overline{\mathbf{W}} = e(k) \overline{\eta} \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}} \right]^T = e(k) \overline{\eta} \begin{bmatrix} \frac{\partial \hat{y}(k)}{\partial \overline{m}^{(l)}} \\ \frac{\partial \hat{y}(k)}{\partial \overline{m}^{(r)}} \\ \frac{\partial \hat{y}(k)}{\partial \overline{\sigma}^{(l)}} \\ \frac{\partial \hat{y}(k)}{\partial \overline{\sigma}^{(r)}} \end{bmatrix}, \qquad \Delta \underline{r} = e(k) \eta_r \frac{\partial \hat{y}(k)}{\partial \underline{r}},$$

$$\Delta \mathbf{W}_a = e(k)\eta_a \left[\frac{\partial \hat{y}(k)}{\partial \mathbf{W}_a} \right]^T = e(k)\eta_a \begin{bmatrix} \dfrac{\partial \hat{y}(k)}{\partial \underline{W}_a} \\ \dfrac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}_a} \end{bmatrix}, \qquad \Delta \overline{\mathbf{W}}_a = e(k)\eta_a \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}_a} \right]^T,$$

$$\Delta \underline{\mathbf{W}}^F = e(k)\underline{\eta} \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}^F} \right]^T = e(k)\underline{\eta} \begin{bmatrix} \dfrac{\partial \hat{y}(k)}{\partial \underline{m}^{F(l)}} \\ \dfrac{\partial \hat{y}(k)}{\partial \underline{m}^{F(r)}} \\ \dfrac{\partial \hat{y}(k)}{\partial \underline{\sigma}^{F(l)}} \\ \dfrac{\partial \hat{y}(k)}{\partial \underline{\sigma}^{F(r)}} \end{bmatrix}, \qquad \Delta \overline{\mathbf{W}}^F = e(k)\overline{\eta} \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}^F} \right]^T = e(k)\overline{\eta} \begin{bmatrix} \dfrac{\partial \hat{y}(k)}{\partial \overline{m}^{F(l)}} \\ \dfrac{\partial \hat{y}(k)}{\partial \overline{m}^{F(r)}} \\ \dfrac{\partial \hat{y}(k)}{\partial \overline{\sigma}^{F(l)}} \\ \dfrac{\partial \hat{y}(k)}{\partial \overline{\sigma}^{F(r)}} \end{bmatrix},$$

$$\Delta \underline{r}^F = e(k)\eta_r \frac{\partial \hat{y}(k)}{\partial \underline{r}^F}.$$

Therefore, the change in the Lyapunov function is

$$\Delta V(k) = \frac{1}{2}\Delta e(k)[2e(k) + \Delta e(k)]$$

$$= \frac{1}{2}\left\{ e(k)\eta_w \left[\frac{\partial e(k)}{\partial \mathbf{W}_w} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \mathbf{W}_w} \right] + e(k)\underline{\eta} \left[\frac{\partial e(k)}{\partial \underline{\mathbf{W}}} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}} \right] + e(k)\overline{\eta} \left[\frac{\partial e(k)}{\partial \overline{\mathbf{W}}} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}} \right] \right.$$

$$+ e(k)\eta_r \left[\frac{\partial e(k)}{\partial \underline{r}} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \underline{r}} \right] + e(k)\eta_a \left[\frac{\partial e(k)}{\partial \underline{\mathbf{W}}_a} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}_a} \right] + e(k)\eta_a \left[\frac{\partial e(k)}{\partial \overline{\mathbf{W}}_a} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}_a} \right]$$

$$\left. + e(k)\underline{\eta} \left[\frac{\partial e(k)}{\partial \underline{\mathbf{W}}^F} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}^F} \right] + e(k)\overline{\eta} \left[\frac{\partial e(k)}{\partial \overline{\mathbf{W}}^F} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}^F} \right] + e(k)\eta_r \left[\frac{\partial e(k)}{\partial \underline{r}^F} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \underline{r}^F} \right] \right\}$$

$$\times \left\{ 2e(k) + e(k)\eta_w \left[\frac{\partial e(k)}{\partial \mathbf{W}_w} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \mathbf{W}_w} \right] + e(k)\underline{\eta} \left[\frac{\partial e(k)}{\partial \underline{\mathbf{W}}} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}} \right] \right.$$

$$+ e(k)\overline{\eta} \left[\frac{\partial e(k)}{\partial \overline{\mathbf{W}}} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}} \right] + e(k)\eta_r \left[\frac{\partial e(k)}{\partial \underline{r}} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \underline{r}} \right] + e(k)\eta_a \left[\frac{\partial e(k)}{\partial \underline{\mathbf{W}}_a} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}_a} \right]$$

$$+ e(k)\eta_a \left[\frac{\partial e(k)}{\partial \overline{\mathbf{W}}_a} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}_a} \right] + e(k)\underline{\eta} \left[\frac{\partial e(k)}{\partial \underline{\mathbf{W}}^F} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}^F} \right] + e(k)\overline{\eta} \left[\frac{\partial e(k)}{\partial \overline{\mathbf{W}}^F} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}^F} \right]$$

$$\left. + e(k)\eta_r \left[\frac{\partial e(k)}{\partial \underline{r}^F} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \underline{r}^F} \right] \right\}$$

$$= -[e(k)]^2 \frac{1}{2}\left\{ \eta_w \left[\frac{\partial \hat{y}(k)}{\partial \mathbf{W}_w} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \mathbf{W}_w} \right] + \underline{\eta} \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}} \right] + \overline{\eta} \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}} \right] \right.$$

$$+ \eta_{\underline{r}} \left[\frac{\partial \hat{y}(k)}{\partial \underline{r}} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \underline{r}} \right] + \eta_a \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}_a} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}_a} \right] + \eta_a \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}_a} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}_a} \right]$$

$$+ \underline{\eta} \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}^F} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}^F} \right] + \overline{\eta} \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}^F} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}^F} \right] + \eta_{\underline{r}} \left[\frac{\partial \hat{y}(k)}{\partial \underline{r}^F} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \underline{r}^F} \right] \Bigg\}$$

$$\times \Bigg\{ 2 - \eta_w \left[\frac{\partial \hat{y}(k)}{\partial \mathbf{W}_w} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \mathbf{W}_w} \right] - \underline{\eta} \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}} \right] - \overline{\eta} \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}} \right]$$

$$- \eta_{\underline{r}} \left[\frac{\partial \hat{y}(k)}{\partial \underline{r}} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \underline{r}} \right] - \eta_a \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}_a} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}_a} \right] - \eta_a \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}_a} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}_a} \right]$$

$$- \underline{\eta} \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}^F} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}^F} \right] - \overline{\eta} \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}^F} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}^F} \right] - \eta_{\underline{r}} \left[\frac{\partial \hat{y}(k)}{\partial \underline{r}^F} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \underline{r}^F} \right] \Bigg\}$$

$$= -[e(k)]^2 \frac{1}{2} \left(\lambda_w + \underline{\lambda} + \overline{\lambda} + \lambda_{\underline{r}} + \underline{\lambda}_a + \overline{\lambda}_a + \underline{\lambda}^F + \overline{\lambda}^F + \lambda_{\underline{r}^F} \right) \times \left(2 - \lambda_w - \underline{\lambda} \right.$$

$$\left. - \overline{\lambda} - \lambda_{\underline{r}} - \underline{\lambda}_a - \overline{\lambda}_a - \underline{\lambda}^F - \overline{\lambda}^F - \lambda_{\underline{r}^F} \right) \tag{54}$$

$$= -[e(k)]^2 \frac{1}{2} \lambda$$

where

$$\lambda_w = \eta_w \left[\frac{\partial \hat{y}(k)}{\partial \mathbf{W}_w} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \mathbf{W}_w} \right] > 0, \qquad \underline{\lambda} = \underline{\eta} \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}} \right] > 0,$$

$$\overline{\lambda} = \overline{\eta} \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}} \right] > 0, \qquad \lambda_{\underline{r}} = \eta_{\underline{r}} \left[\frac{\partial \hat{y}(k)}{\partial \underline{r}} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \underline{r}} \right] > 0,$$

$$\underline{\lambda}_a = \eta_a \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}_a} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}_a} \right] > 0, \qquad \overline{\lambda}_a = \eta_a \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}_a} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}_a} \right] > 0,$$

$$\underline{\lambda}^F = \underline{\eta} \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}^F} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \underline{\mathbf{W}}^F} \right] > 0, \qquad \overline{\lambda}^F = \overline{\eta} \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}^F} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \overline{\mathbf{W}}^F} \right] > 0, \text{ and}$$

$$\lambda_{\underline{r}^F} = \eta_{\underline{r}} \left[\frac{\partial \hat{y}(k)}{\partial \underline{r}^F} \right]^T \left[\frac{\partial \hat{y}(k)}{\partial \underline{r}^F} \right] > 0.$$

Let

$$\lambda = \left(\lambda_w + \underline{\lambda} + \overline{\lambda} + \lambda_{\underline{r}} + \underline{\lambda}_a + \overline{\lambda}_a + \underline{\lambda}^F + \overline{\lambda}^F + \lambda_{\underline{r}^F} \right) \times \left(2 - \lambda_w - \underline{\lambda} - \overline{\lambda} - \lambda_{\underline{r}} \right.$$
$$\left. - \underline{\lambda}_a - \overline{\lambda}_a - \underline{\lambda}^F - \overline{\lambda}^F - \lambda_{\underline{r}^F} \right).$$

The convergence of RiT2FNN-A is guaranteed if $\Delta V(k) < 0$, i.e., $\lambda > 0$, and

$$\left(\lambda_w + \underline{\lambda} + \overline{\lambda} + \lambda_r + \underline{\lambda}_a + \overline{\lambda}_a + \underline{\lambda}^F + \overline{\lambda}^F + \lambda_{r^F} \right) < 2 .$$
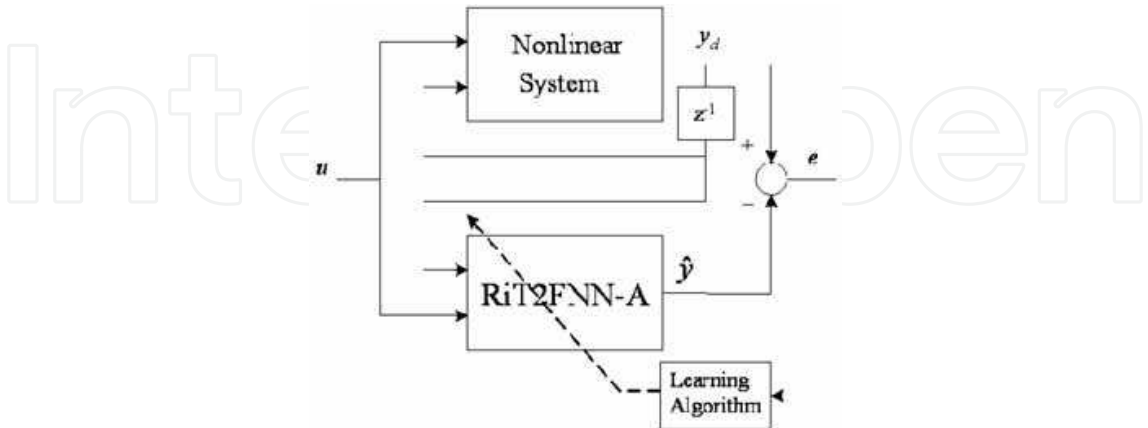
This completes the proof.

Fig. 7. Series-parallel identification scheme using RiT2FNN-A.

(a)

(b)
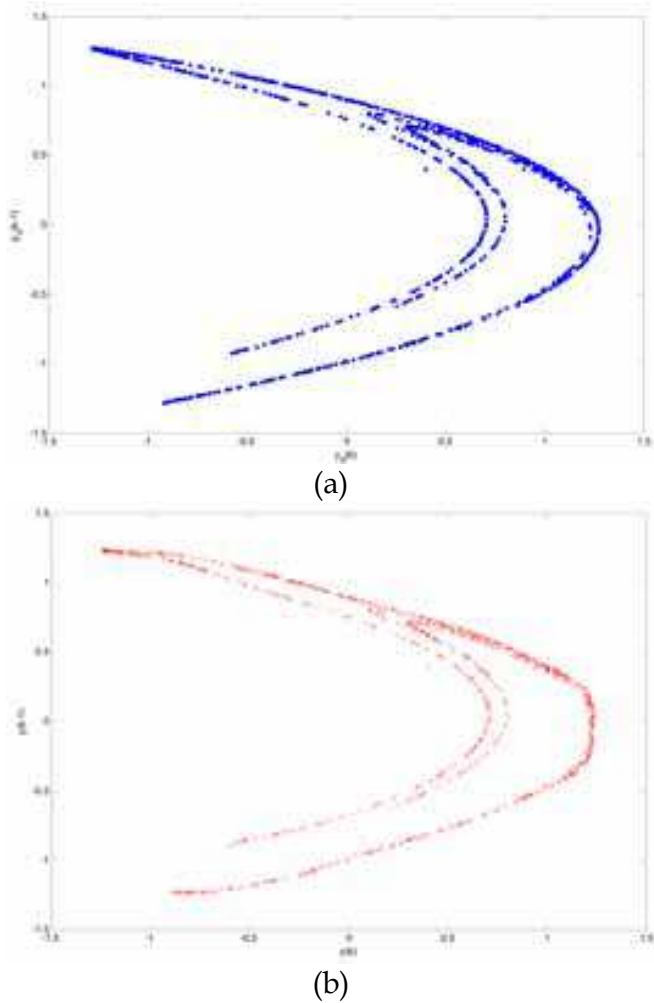
Fig. 8. Phase plane plot of chaotic system; (a) the chaotic system, (b) identification result of RiT2FNN-A.

4. System identification using RiT2FNN-A system

Consider the following non-linear system

$$y_d(k+1) = f(y_d(k), \cdots, y_d(k-n+1), u(k), \cdots, u(k-m+1)) \tag{55}$$

where u and y_d are systematic input and output; function f(.) is the unknown function which is approximated by the RiT2FNN-A. And then m and n are all positive integer number. Here, the series-parallel training scheme is adopted, as shown in Fig. 7. The approximated error is defined as follows

$$e(k) \equiv y_d(k) - \hat{y}(k) \tag{56}$$

where $\hat{y}(k)$ denotes the RiT2FNN-A's output. Clearly, the inputs of RiT2FNN-A are contro input u and system past input $y_d(k\text{-}1)$. If a static network system (or feed-forward neural network) is used, such as, neural network, fuzzy neural network, T2FNN, T2FNN-A, the input number of $n+m$ should be used. This is due to the dynamic property (feedback layer) of RiT2FNN-A system.

In general, the following Training-Mean-Square-Error (TMSE) is adopted to be the performance index.

$$TMSE \equiv \frac{1}{N} \sum_{k=1}^{N} e^2(k) \tag{57}$$

where N is the number of training pattern.

In this article, the following nonlinear chaotic system is considered

$$y_d(k) = -P \cdot y_d^2(k-1) + Q \cdot y_d(k-2) + 1.0 \tag{58}$$

where P=1.4 and Q=0.3.

The feed-forward type-2 fuzzy neural network- T2FNN and T2FNN-A, are used to have comparisons in nonlinear system identification for illustrating the performance of RiT2FNNA. It is clear that the feed-forward T2FNN with three input nodes for feeding appropriate past values of yd and u were used. In this article, only two values, $y_d(k\text{-}1)$ and u(k), are fed into the RiT2FNN-A to predict the system output. In training the RiT2FNN-A, we first randomly choose the training data (1000 pairs) from system over the interval [-1.5 1.5]. Then, the RiT2FNN-A is used to approximate the chaotic system. In this simulation, we use 3 rules to construct the RiT2FNN-A. Learning rate is selected as 0.1.

The simulation results are described in Figs. 8 and 9. Figure 8(a) shows the phase plane of this chaotic system, whereas Fig. 8(b) shows the result of RiT2FNN-A system after training (10 epochs). The initial point is $[y_d(1), y_d(0)]T=[0.4, 0.4]T$ and the TMSE is 0.00019886, which is less than the results of T2FNN-A and T2FNN (as shown in Fig. 9). The initial interval T2MFs for input and internal variables x and g are empirically designed as Figs. 10(a) and 10(b), respectively. After training, the final iT2AFMFs are shown in Figs. 10(c) and 10(d). Obviously, the iT2AFMFs are obtained for better performance.

In order to make sure RiT2FNN-A system to be stable in training, we need to check the condition (50). Figure 11 shows the values of $\beta = \lambda_w + \underline{\lambda} + \overline{\lambda} + \lambda_t + \underline{\lambda}_a + \overline{\lambda}_a + \underline{\lambda}^F + \overline{\lambda}^F + \lambda_{t^F}$

which were introduced previously in (50), the stable condition hold if $\beta<2$. Obviously, condition (50) holds in training epochs.
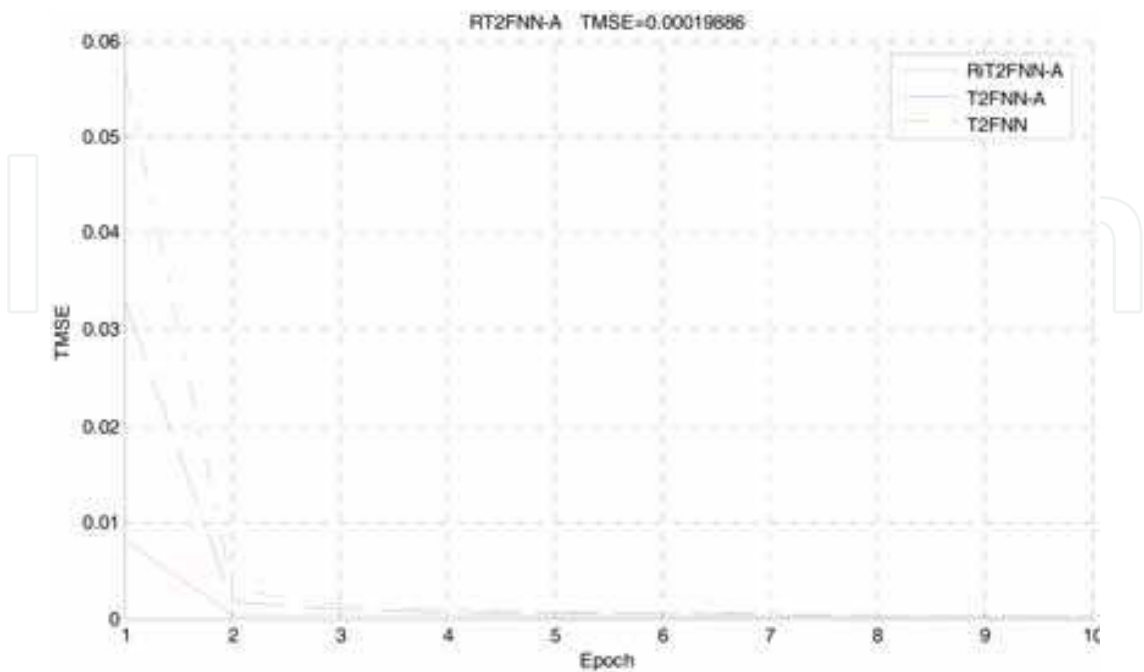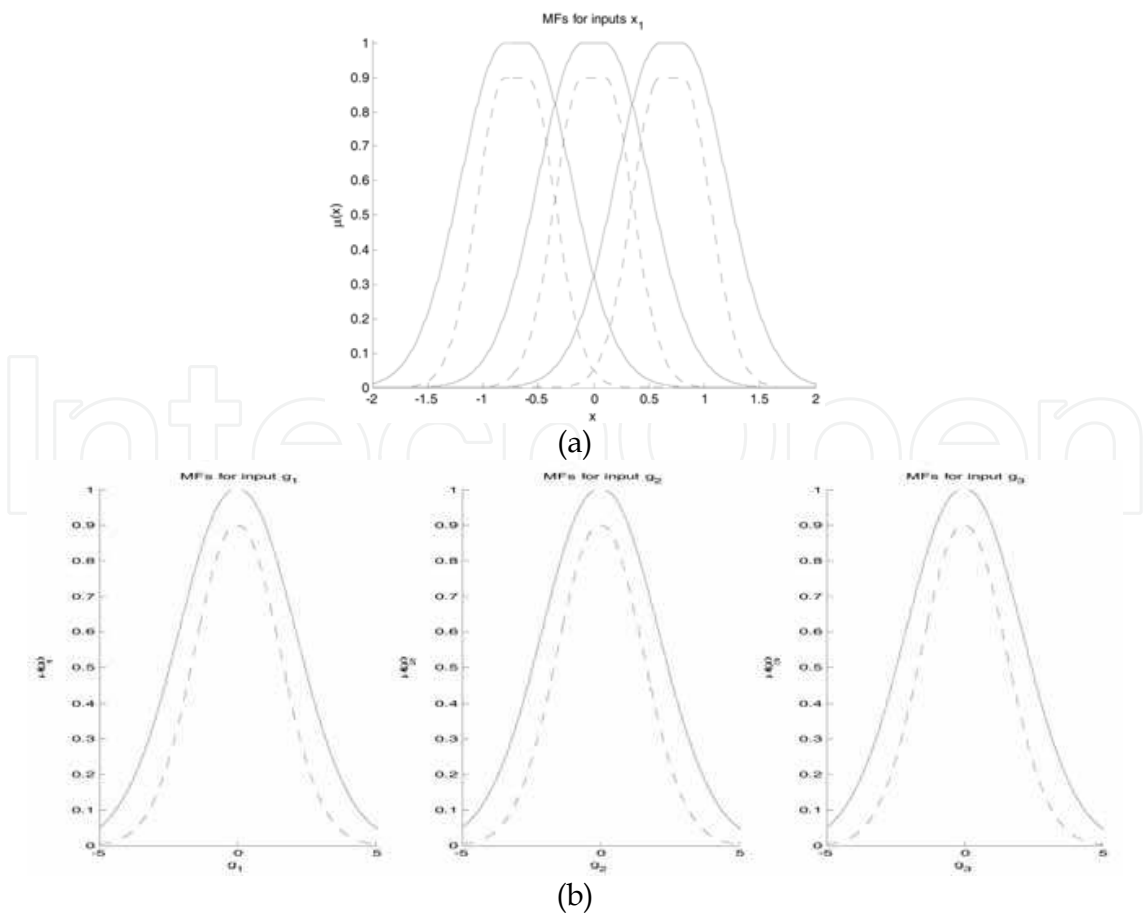

Fig. 9. Simulation results of system identification; (a) system output, (b) learning curves of the T2FNN (dotted-line), T2FNN-A (dashed-line) and RiT2FNN-A (solid-line).
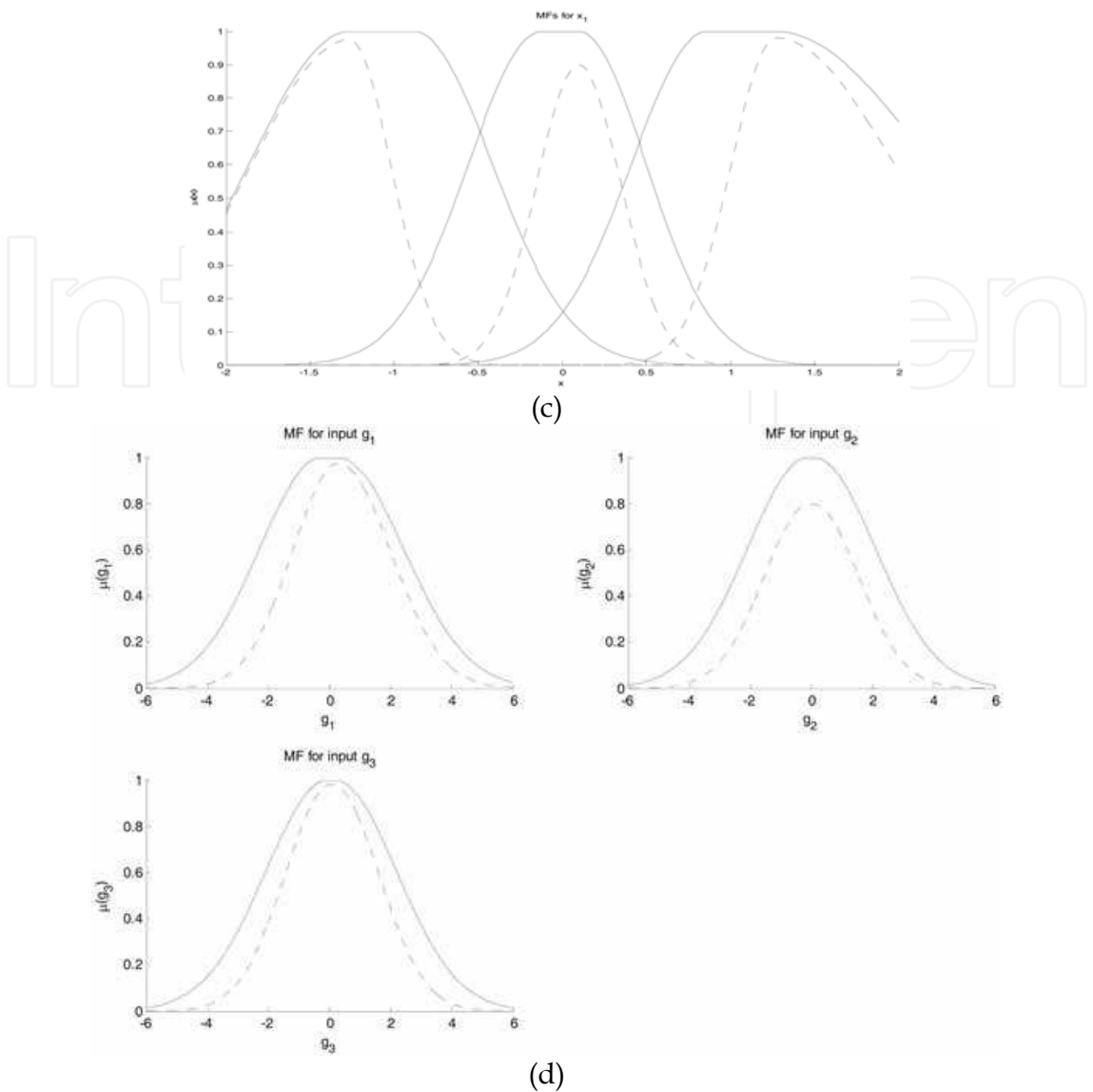

(a)

(b)

(c)

(d)

Fig. 10. Membership functions; (a) initial MFs for x_1, (b) initial MFs for g_1, g_2, and g_3, (c) MFs for x_1 after training, and (d) MFs for g_1, g_2, and g_3 after training.

Fig. 11. Condition checking of (50).

This simulation demonstrates that the RiT2FNN-A has the smaller network structure for identification. In addition, we observe that the identification error of the RiT2FNN-A is less than that of T2FNN-A for each epoch.

	Network structure	Rule number (M)	Parameter number	TMSE ($\times 10^{-3}$)
T2FNN (Lee & Lin, 2005)	2-10-5-1	5	40	0.35
	1-3-3-1	3	15	215.5
T2FNN-A (Lee & Pan, 2007)	2-6-3-1	3	36	0.34
	1-3-3-1	3	21	195.9
RiT2FNN-A	1-3-3-1	3	24	0.20

Table 1. Comparison results of network structure, rule number, parameter number, and TMSE.

Table 1 shows the comparison results of network structure, rule number, parameter number, and TMSE. Obviously, the asymmetric MFs improve the approximation accuracy of the iT2FLSs. On the other hand, for a given approximation accuracy, RiT2FNN-A can achieve by using less fuzzy rules and tuning parameters with simplified structure.

5. Conclusion

This article has introduced a novel recurrent interval type-2 fuzzy neural network with asymmetric membership functions, which utilizes Lyapunov stability theorem to prove the stability of the system. The novel RiT2FNN-A use the interval asymmetric type-2 fuzzy sets implements the FLS in a five-layer neural network structure which contains four layer forward network and a feedback layer. According to the Lyapunov theorem and gradient descent method, the convergence of RiT2FNN-A is guaranteed and the corresponding learning algorithm is derived. Moreover, the RIT2FNN-A capability to temporarily store information allowed us to extend the application domain to include temporal problem. In application, We have found that the proposed RiT2FNN-A can use a smaller network structure and a small number of tuning parameters than the feed-forward fuzzy neural networks to obtain similar or better performance. It can successfully also approximate to a dynamic system mapping as accurately as desired.

6. References

Baldwin, J. F. & Karake, S. B. (2003). Asymmetric Triangular Fuzzy Sets for Classification Models, *Lecture Notes in Artificial Intelligence*, Vol. 2773, pp. 364-370, 2003.

Castillo, O. & Melin, P. (2004). Adaptive Noise Cancellation Using Type-2 Fuzzy Logic and Neural Networks, *IEEE International Conf. on Fuzzy Systems*, Vol. 2, pp. 1093-1098, 2004.

Jang, J. S. (1993). ANFIS: Adaptive-Network-Based Fuzzy Inference System, *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 23, No. 3, pp. 665-685, 1993.

John, R. & Coupl, S. (1999). Geometric Type-1 and Type-2 Fuzzy Logic Systems, *IEEE Trans. on Fuzzy Systems, Special Issue on Type-2 Fuzzy Systems*, Vol. 15, No. 1, pp. 3-15, 2007.

Juang, C. F. (2002), A TSK-type recurrent fuzzy network for dynamic systems processingby neural network and genetic algorithms, *IEEE trans. on Fuzzy Systems*, Vol. 10, No. 2, pp. 155- 170, 2002.

Karnik, N. N., Mendel, J. & Liang, Q. (1999). Type-2 Fuzzy Logic Systems, *IEEE Trans. on Fuzzy Systems*, Vol. 7, No. 6, pp. 643-658, 1999.

Kim, M. S., Kim, C. H. & Lee, J. J. (2003). 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.

Lee, C. H. & Lin, Y. C. (2005). An Adaptive Type-2 Fuzzy Neural Controller for Nonlinear Uncertain Systems, *International Journal of Control and Intelligent*, Vol. 12, No. 1, pp. 41-50, 2005.

Lee, C. H. & Teng, C. C. (2000). Identification and Control of Dynamic Systems Using Recurrent Fuzzy Neural Networks, *IEEE Trans. on Fuzzy Systems*, Vol. 8, No. 4, pp. 349-366, 2000.

Lee, C. H. & Teng, C. C. (2001).Fine Tuning of Membership Functions for Fuzzy Neural Systems, *Asian Journal of Control*, Vol. 3, No. 3, pp. 216-225, 2001.

Lee, C. H. Lin, Y. C. & Lai, W. Y. (2003). Systems Identification Using Type-2 Fuzzy Neural Network (Type-2 FNN) Systems, *IEEE International Sym. on Computational Intelligence in Robotics and Automation*, Vol. 3, pp. 1264-1269, 2003.

Lee, C. H. & Pan, H. Y. (2007). Enhancing the Performance of Neural Fuzzy Systems Using Asymmetric Membership Functions, Revised in *Fuzzy Sets and Systems*, 2007.

Li, C. Cheng, K. H. & Lee, J. D. (2005). 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.

Li, C. & Lee, C. Y. (2003). Self-organizing Neuro-fuzzy System for Control of Unknown Plants, *IEEE Trans. on Fuzzy Systems*, Vol. 11, No. 1, pp. 135-150, 2003.

A. Q. Liang & J. M. Mendel, Interval Type-2 Fuzzy Logic Systems: Theory and Design, *IEEE Trans. on Fuzzy Systems*, Vol. 8, No. 5, pp. 535-550, 2000.

Lin, C. T. & Lee, C. S. G. *Neural Fuzzy Systems: A Neuro-fuzzy Synergism to Intelligent Systems*, Prentice-Hall, Englewood Cliffs, 1996.

Lin, C. J. & Ho, W. H. (2005). An Asymmetric-similarity-measure-based Neural Fuzzy Inference System, *Fuzzy Sets and Systems*, Vol. 152, pp. 535-551, 2005.

Lotfi, A. & Tsoi, A. C. (1996). Learning Fuzzy Inference Systems Using An Adaptive Membership Function Scheme, *IEEE Trans. on Systems, Man, and Cybernetics, Part-B*, Vol. 26, No. 2, pp. 326-331, 1996.

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

Hagras, H. (2007). Type-2 FLCs: A New Generation of Fuzzy Controllers, *IEEE Computational Intelligence Magazine*, Vol. 2, No. 1, pp. 30-43, 2007.

Narendra, K. S. & Parthasarathy, K. (1990). Identification and Control of Dynamical Systems Using Neural Networks, *IEEE Trans. on Neural Networks*, Vol. 1, No. 1, pp. 4-27, 1990.

Ozen, T. & Garibaldi, J. M. (2004). Effect of Type-2 Fuzzy Membership Function Shape on Modeling Variation in Human Decision Making, *IEEE International Conf. on Fuzzy Systems*, Vol.2, pp. 971-976, 2004.

Pan, H. Y., Lee, C. H., Chang, F. K., & Chang, S. K. (2007). Construction of Asymmetric Type-2 Fuzzy Membership Functions and Application in Time Series Prediction, *International Conf. on Machine Learning and Cybernetics,* Vol. 4, pp. 2024-2030, 2007.

Wang, C. H. Cheng, C. S. & Lee, T. T. (2004). Dynamical Optimal Training for Interval Type-2 Fuzzy Neural Network (T2FNN), *IEEE Trans. on Systems, Man, Cybernetics Part-B*, Vol. 34, No. 3, pp. 1462-1477, 2004.

Wang, L. X. (1994). *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*, Prentice -Hall, Englewood Cliffs, NJ, 1994.

Zadeh, L. A. (1975). The Concept of A Linguistic Variable and Its Application to Approximate Reasoning, *Information Sciences*, Vol. 8, No.3, pp. 199-249, 1975.
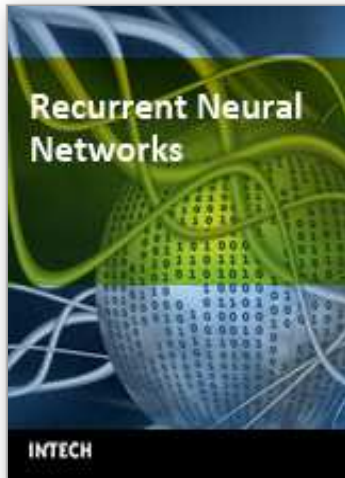
Recurrent Neural Networks

Edited by Xiaolin Hu and P. Balasubramaniam

The concept of neural network originated from neuroscience, and one of its primitive aims is to help us understand the principle of the central nerve system and related behaviors through mathematical modeling. The first part of the book is a collection of three contributions dedicated to this aim. The second part of the book consists of seven chapters, all of which are about system identification and control. The third part of the book is composed of Chapter 11 and Chapter 12, where two interesting RNNs are discussed, respectively.The fourth part of the book comprises four chapters focusing on optimization problems. Doing optimization in a way like the central nerve systems of advanced animals including humans is promising from some viewpoints.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Ching-Hung Lee and Tzu-Wei Hu (2008). Recurrent Interval Type-2 Fuzzy Neural Network Using Asymmetric Membership Functions, Recurrent Neural Networks, Xiaolin Hu and P. Balasubramaniam (Ed.), ISBN: 978-953-7619-08-4, InTech, Available from:
http://www.intechopen.com/books/recurrent_neural_networks/recurrent_interval_type-2_fuzzy_neural_network_using_asymmetric_membership_functions

INTECH
open science | open minds