

Interval Type-2 Recurrent Fuzzy Neural System Desing via Stable Simultaneous Perturbation Stochastic Approximation Algorithm

Feng-Yu Chang

Department of Electrical Engineering
Yuan Ze University
Chungli, Taoyuan, Taiwan

Ching-Hung Lee

Department of Electrical Engineering
Yuan Ze University
Chungli, Taoyuan, Taiwan
E-mail: chlee@saturn.yzu.edu.tw

Abstract— This paper proposes a new type fuzzy neural systems, denotes IT2RFNS-A (interval type-2 recurrent fuzzy neural system with asymmetric membership function), for nonlinear systems control. To enhance the performance and approximation ability, the TSK-type consequent part is adopted for IT2RFNS-A. The gradient information of the IT2RFNS-A is not easy to obtain due to the asymmetric membership functions and interval valued sets. The corresponding stable learning is derived by simultaneous perturbation stochastic approximation (SPSA) algorithm which guarantees the convergence and stability of the closed-loop systems. Simulation and comparison on the control of Chua's chaotic circuit is done to show the feasibility and effectiveness of proposed method.

Keywords- Nonlinear systems, fuzzy neural system, type-2 fuzzy system, Lyapunov theorem, SPSA algorithm

I. INTRODUCTION

In past few decades, the fuzzy neural network (FNN) which provides the advantages of both neural network and fuzzy system is successfully applied to nonlinear system identification and control [1-4]. In the FNN, the symmetric and fixed membership functions are commonly adopted to simplify the design procedure. However, a large number of fuzzy rules should be used to achieve the specified performance [5, 6]. Thus, an asymmetric fuzzy membership function (AFMF) has been proposed to solve this problem. The AFMF is discussed and analyzed that it can effectively improve the accuracy and achieve the specific performance with fewer fuzzy rules [7, 8].

Recently, the type-2 fuzzy sets (T2 FSs) are gaining more and more in popularity [9, 10]. The T2 FSs are described by membership functions (MFs) that are characterized by more parameters than the type-1 fuzzy sets (T1 FSs). Hence, the T2 FSs provide us with more design degree of freedom. Nevertheless, due to the computational complexity of using the T2 FS, most people only use interval type-2 fuzzy sets (IT2 FSs) [10]. The computations associated with IT2 FSs are very manageable which makes it quite practical [11]. Many results has been shown that the interval type-2 fuzzy logic system (IT2 FLS) can successfully apply to cope with the control problems and perform well [12-17]. In our previous

research [18-21], we proposed a type-2 fuzzy neural network with asymmetric membership function (T2FNN-A) which combines the IT2 FLS with neural network. Then, in order to improve the efficiency of the T2FNN-A, we proposed an interval type-2 fuzzy neural system with asymmetric membership function (IT2FNS-A) which utilizes several enhanced techniques such as Takagi-Sugeno-Kang fuzzy logic system (TSK FLS) and embedded type-reduction network layer [22]. These modifications not only improve the approximation accuracy of the T2FNN-A, but also achieve the specific performance with fewer fuzzy rules. However, a major drawback of the IT2FNS-A is that its application domain is limited to static problem due to its feedforward network structure. Using IT2FNS-A to process dynamic problems is inefficient. Thus, in this paper, we propose an interval type-2 recurrent fuzzy neural system with asymmetric membership functions (IT2RFNS-A) which provides the memory elements and extends the basic ability of the IT2FNS-A to include the dynamic problems. In addition, since the feedback layer captures the dynamic response of the system, the approximation accuracy of the network is improved.

In training of the neural network, the back-propagation (BP) algorithm is widely used. There is, however, a critical problem where the differential information of the system is very difficult to obtain because there are too many adjustable parameters in the IT2RFNS-A. Herein, we adopt the simultaneous perturbation stochastic approximation (SPSA) algorithm which is proposed by Spall [23-25]. This algorithm is based on the approximation to gradient formed from the measurements of the objective function. Hence, a great deal of computational effort is saved. Nevertheless, there still remains a problem. Because of the stochastic characteristic of SPSA algorithm, we cannot guarantee that every search step length is appropriate which may lead to the invalid search. To overcome this situation, we employ the Lyapunov stability analysis to derive the optimal learning step length for guaranteeing the stability of the closed-loop system. In addition, the efficient training is also ensured.

The remainder of this paper is organized as follows. In Section II, we cover the prerequisite material to be used in the study. Section III illustrates the proposed IT2RFNS-A. The

simulation results on chaotic system control of the Chua's chaotic circuit are shown in Section IV. Finally, the conclusion is given in Section V.

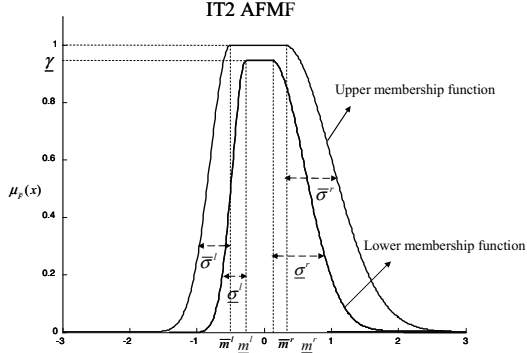


Figure 1. Construction of interval type-2 asymmetric fuzzy membership function.

II. PRELIMINARIES

A. Interval Type-2 Asymmetric Fuzzy Membership Function

Several literatures indicate that using AFMFs can improve the approximation accuracy and for effectively reducing the number of fuzzy rules [5-8]. It also provides better performance than traditional fuzzy systems on application of function approximation, modeling, and control. The learning capability and flexibility can be upgraded [5-8]. Here, we introduce the IT2 AFMFs. According to previous results [8], Gaussian functions are used to construct IT2 AFMFs, as shown in Fig. 1. Each IT2 AFMF is constructed by parts of four Gaussian functions. The upper and lower membership functions (MFs) are constructed by using two Gaussian MFs and one segment. The superscripts “*l*” and “*r*” denote the left and right curves of MF. The parameters of the lower and upper MFs are denoted by “*l*” and “*r*”, respectively. Accordingly, the upper AFMF is constructed as:

$$\bar{\mu}_F(x) = \begin{cases} e^{-\frac{1}{2}(\frac{x-\bar{m}^l}{\bar{\sigma}^l})^2}, & \text{for } x \leq \bar{m}^l \\ 1, & \text{for } \bar{m}^l < x < \bar{m}^r \\ e^{-\frac{1}{2}(\frac{x-\bar{m}^r}{\bar{\sigma}^r})^2}, & \text{for } \bar{m}^r \leq x \end{cases} \quad (1)$$

where $\bar{m}^l$ and $\bar{m}^r$ denote the means of two Gaussian MFs satisfying $\bar{m}^l \leq \bar{m}^r$, and $\bar{\sigma}^l$ and $\bar{\sigma}^r$ denote the deviations (width) of two Gaussian MFs. Similarly, the lower AFMF is

$$\underline{\mu}_F(x) = \begin{cases} \lambda \cdot e^{-\frac{1}{2}(\frac{x-\underline{m}^l}{\underline{\sigma}^l})^2}, & \text{for } x \leq \underline{m}^l \\ \lambda, & \text{for } \underline{m}^l < x < \underline{m}^r \\ \lambda \cdot e^{-\frac{1}{2}(\frac{x-\underline{m}^r}{\underline{\sigma}^r})^2}, & \text{for } \underline{m}^r \leq x \end{cases} \quad (2)$$

where $\underline{m}^l \leq \underline{m}^r$. To avoid a small activation value, λ is chosen to be $0.5 \leq \lambda \leq 1$. Thus, the following restrictions should be added to avoid unreasonable MFs:

$$\bar{m}^l \leq \underline{m}^l \leq \underline{m}^r \leq \bar{m}^r, \quad \underline{\sigma}^l \leq \bar{\sigma}^l, \quad \underline{\sigma}^r \leq \bar{\sigma}^r, \quad 0.5 \leq \lambda \leq 1. \quad (3)$$

B. Simultaneous Perturbation Stochastic Approximation Algorithm

This section introduces the SPSA algorithm briefly. The detail description can be found in literature [25]. Consider a problem of finding the minimum of objective function $f(W)$. The SPSA algorithm computes the parameter W at next iteration as

$$W(k+1) = W(k) - a_k g(W(k)) \quad (4)$$

where $g(\cdot)$ is estimated gradient result for objective function

$f(\cdot)$, i.e., $\frac{\partial f(W)}{\partial W} \approx g(W)$. a_k is the learning step length which

is decreased over iterations with $a_k = \frac{a}{(k+A)^\alpha}$, where a ,

A , and α are positive configuration coefficients [25]. The SPSA approach estimates the gradient $g(\cdot)$ using following method. Assume that the dimension of parameter W is p . Let $\Delta_k = [\Delta_{k1} \Delta_{k2} \dots \Delta_{kp}]$ be a p -dimensional vector whose element is mutually independent zero-mean random variable. Then, the estimation of the gradient at k th iteration can be computed by SPSA algorithm

$$g(W(k)) = \frac{f(W(k) + c_k \Delta_k) - f(W(k))}{c_k} [\Delta_{k1}^{-1} \Delta_{k2}^{-1} \dots \Delta_{kp}^{-1}]^T \quad (5)$$

$c_k = c/(k+1)^\gamma$ is gain sequence, where c and γ are nonnegative configuration coefficients. Obviously, all the elements of the parameter W are perturbed simultaneously, and only two measurement of the objective function are needed to estimate the gradient. In addition, Δ_k is usually obtained using Bernoulli ± 1 distribution with equal probability for each value.

In general, the gradient information of neural fuzzy system is not easy to obtain due to the asymmetric membership functions and interval valued sets. Herein, we adopt the SPSA algorithm to derive the stable learning rule which guarantees the convergence and stability of the closed-loop systems.

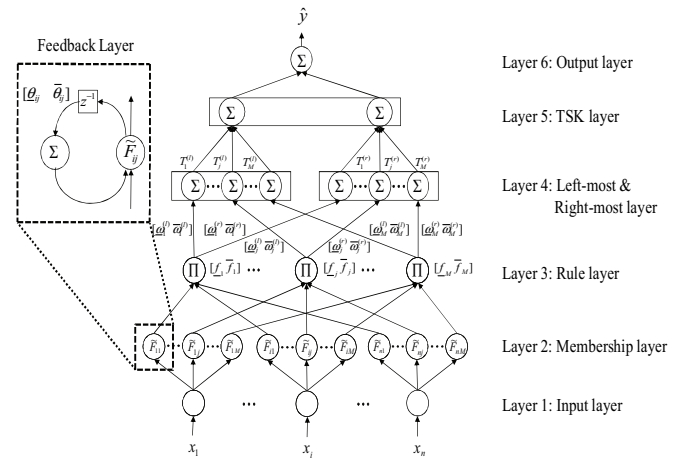


Figure 2. Diagram of the proposed interval type-2 recurrent fuzzy neural system with asymmetric membership functions.

III. INTERVAL TYPE-2 RECURRENT FUZZY NEURAL SYSTEM WITH ASYMMETRIC MEMBERSHIP FUNCTION

A. Fuzzy Reasoning of IT2RFNS-A

The proposed IT2RFNS-A realizes the fuzzy inference into the IT2RFNS-A systems. Assume that an IT2RFNS-A system has M rules and n inputs, the j th rule can be expressed as

$$R^j : \text{IF } z_{1j} \text{ is } \tilde{F}_{1j} \text{ and } \dots \text{ and } z_{nj} \text{ is } \tilde{F}_{nj}, \quad (6)$$

$$\text{THEN } Y_j = C_{j0} + C_{j1}x_1 + C_{j2}x_2 + \dots + C_{jn}x_n$$

where z_{ij} is input linguistic variable of the j th rule, $\tilde{F}_{ij}$ is interval type-2 antecedent fuzzy set, C_{ji} is type-1 consequent fuzzy set, x_i is the network input, and Y_j is the output of the j th rule and is also a type-1 fuzzy set (because it is the linear combination of type-1 fuzzy sets). Note that the input linguistic variable z_{ij} has the system input term and the past information which is introduced in Section III.B. Due to the IT2 FSs are used for the antecedent part and the interval type-1 fuzzy sets are used for the consequent part of the type-2 TSK rule, $\mu_{\tilde{F}_{ij}}$ and C_{ji} are interval sets, i.e.,

$$\mu_{\tilde{F}_{ij}}(z_{ij}) = [\underline{\mu}_{\tilde{F}_{ij}}(z_{ij}), \bar{\mu}_{\tilde{F}_{ij}}(z_{ij})] \quad (7)$$

$$C_{ji} = [c_{ji} - s_{ji}, c_{ji} + s_{ji}] \quad (8)$$

where c_{ji} denotes the center (mean) of C_{ji} , and s_{ji} denotes the spread of C_{ji} . Therefore, using the product t -norm, the firing strength associated with the j th rule is

$$F_j(\mathbf{z}) = [\underline{f}_j(\mathbf{z}), \bar{f}_j(\mathbf{z})] \quad (9)$$

where

$$\underline{f}_j(\mathbf{z}) = \underline{\mu}_{\tilde{F}_{1j}}(z_{1j}) * \dots * \underline{\mu}_{\tilde{F}_{nj}}(z_{nj}) \quad (10)$$

and

$$\bar{f}_j(\mathbf{z}) = \bar{\mu}_{\tilde{F}_{1j}}(z_{1j}) * \dots * \bar{\mu}_{\tilde{F}_{nj}}(z_{nj}). \quad (11)$$

The consequent of rule R^j , Y_j is also an interval set, i.e., $Y_j = [y_j^l, y_j^r]$, where

$$y_j^l = (c_{j0} + \sum_{i=1}^n c_{ji}x_{ij}) - (s_{j0} + \sum_{i=1}^n s_{ji}|x_{ij}|) \quad (12)$$

and

$$y_j^r = (c_{j0} + \sum_{i=1}^n c_{ji}x_{ij}) + (s_{j0} + \sum_{i=1}^n s_{ji}|x_{ij}|). \quad (13)$$

By applying the Extension principle [26], the output of the FLS is

$$Y_{TSK}(\mathbf{z}) = [y^l, y^r]$$

$$= \frac{\int \dots \int_{y_1 \in [\underline{y}_1^l, \underline{y}_1^r]} \int_{y_2 \in [\underline{y}_2^l, \underline{y}_2^r]} \dots \int_{y_M \in [\underline{y}_M^l, \underline{y}_M^r]} \int_{f_1 \in [\underline{f}_1, \bar{f}_1]} \int_{f_2 \in [\underline{f}_2, \bar{f}_2]} \dots \int_{f_M \in [\underline{f}_M, \bar{f}_M]} \prod_{j=1}^M f_j y_j}{\int \dots \int_{f_1 \in [\underline{f}_1, \bar{f}_1]} \int_{f_2 \in [\underline{f}_2, \bar{f}_2]} \dots \int_{f_M \in [\underline{f}_M, \bar{f}_M]} \prod_{j=1}^M f_j} \quad (14)$$

where $\underline{f}_j$, $\bar{f}_j$, y_j^l , and y_j^r are computed using (10)-(13). To compute $Y_{TSK}(\mathbf{z})$, we need to compute its two end-points y^l and y^r by type-reduction operation [26]. Finally, we defuzzify it

using the average of y^l and y^r . Hence, the defuzzified output of IT2RFNS-A is

$$y(\mathbf{z}) = \frac{y^l + y^r}{2}. \quad (15)$$

B. Network Structure of IT2RFNS-A

The IT2RFNS-A consists of six feed-forward layers and a feedback one. Layer 1 accepts input variables. Layer 2 is used to calculate the interval type-2 AFMF grade. Feedback layer which is embedded in Layer 2 is used to store the past information. Layer 3 forms the fuzzy rule base. Layer 4 is used for type-reduction. The TSK consequent part is implemented into Layer 5. Layer 6 is the output layer.

Next, we would indicate the signal propagation and the operation functions of the node in each layer. For convenience, the multi-input-single-output (MISO) case is considered here. The schematic diagram of the proposed IT2RFNS-A is shown in Fig. 2. In the following description, $O_i^{(l)}$ denotes the i th output node in layer l .

Layer 1: Input Layer

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

$$O_i^{(1)} = x_i. \quad (16)$$

The x_i represents the i th input to the i th 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 the interval type-2 asymmetric fuzzy membership function, i.e.,

$$O_{ij}^{(2)} = \mu_{\tilde{F}_{ij}}(O_i^{(1)} + O_i^{(f)})$$

$$= [\underline{O}_{ij}^{(2)}, \bar{O}_{ij}^{(2)}]^T \quad (17)$$

$$= [\underline{\mu}_{\tilde{F}_{ij}}(O_i^{(1)} + O_i^{(f)}), \bar{\mu}_{\tilde{F}_{ij}}(O_i^{(1)} + O_i^{(f)})]^T$$

where subscript ij indicates the j th term of the i th input $O_i^{(1)}$, $\mu_{\tilde{F}_{ij}}$ is an IT2 AFMF as shown in Fig. 1. Note that the output of layer 2 and the feedback weight are interval values. According to the results of [27], the type-reduction can be integrated into the network, $O_i^{(f)}$ is the output of feedback layer

$$O_i^{(f)}(k) = \frac{\underline{O}_{ij}^{(2)}(k-1) \cdot \underline{\theta}_{ij} + \bar{O}_{ij}^{(2)}(k-1) \cdot \bar{\theta}_{ij}}{\underline{\theta}_{ij} + \bar{\theta}_{ij}} \quad (18)$$

where $\underline{\theta}_{ij}$ and $\bar{\theta}_{ij}$ denote the link weight of the feedback unit. Clearly, the input of this layer contains the memory terms $\underline{O}_{ij}^{(2)}(k-1)$ and $\bar{O}_{ij}^{(2)}(k-1)$ which store the past information of the network.

Layer 3: Rule Layer

This layer is used for computing firing strength of the rule. From (9)-(11), we obtain

$$\underline{f}_j = \underline{\mu}_{\tilde{F}_{1j}}(O_1^{(1)} + O_1^{(f)}) * \dots * \underline{\mu}_{\tilde{F}_{nj}}(O_n^{(1)} + O_n^{(f)}) \quad (19)$$

$$\bar{f}_j = \bar{\mu}_{\bar{F}_j}(O_1^{(1)} + O_1^{(f)}) * \dots * \bar{\mu}_{\bar{F}_n}(O_n^{(1)} + O_n^{(f)}) \quad (20)$$

where $\underline{\mu}_{\bar{F}_j}(\cdot)$ and $\bar{\mu}_{\bar{F}_j}(\cdot)$ are the lower and upper membership grades of $\mu_{\bar{F}_j}(\cdot)$. Therefore, for the j th rule, the operation function of the node in this layer is

$$O_j^{(3)} = [\underline{O}_j^{(3)} \quad \bar{O}_j^{(3)}]^T = \left[\prod_{i=1}^n \underline{O}_{ij}^{(2)} \quad \prod_{i=1}^n \bar{O}_{ij}^{(2)} \right]^T \quad (21)$$

Layer 4: Left-most & Right-most layer

According to [27], the type-reduction can be integrated into the network layer by calculating their left-most and right-most value. That is, the complicated type-reduction method such as KM algorithm can be reduced as

$$O_j^{(4)} = [O_{jl}^{(4)} \quad O_{jr}^{(4)}]^T = \left[\frac{\bar{w}_j' \bar{O}_j^{(3)} + \underline{w}_j' \underline{O}_j^{(3)}}{\bar{w}_j' + \underline{w}_j'} \quad \frac{\bar{w}_j^r \bar{O}_j^{(3)} + \underline{w}_j^r \underline{O}_j^{(3)}}{\bar{w}_j^r + \underline{w}_j^r} \right]^T \quad (22)$$

where the link weights are $\underline{w}_j' = [\underline{w}_1' \dots \underline{w}_M']^T$, $\bar{w}_j' = [\bar{w}_1' \dots \bar{w}_M']^T$, $\underline{w}_j^r = [\underline{w}_1^r \dots \underline{w}_M^r]^T$, and $\bar{w}_j^r = [\bar{w}_1^r \dots \bar{w}_M^r]^T$. Note that $\underline{w}_j' < \bar{w}_j'$ and $\underline{w}_j^r < \bar{w}_j^r$.

Layer 5: TSK Layer

From (12) and (13), we can obtain

$$\begin{aligned} T_j &= [T_j' \quad T_j^r]^T \\ &= \left[(c_{j0} + \sum_{i=1}^n c_{ji} x_i) - (s_{j0} + \sum_{i=1}^n s_{ji} |x_i|) \right. \\ &\quad \left. (c_{j0} + \sum_{i=1}^n c_{ji} x_i) + (s_{j0} + \sum_{i=1}^n s_{ji} |x_i|) \right]^T. \end{aligned} \quad (23)$$

Then, the output of this layer is

$$O^{(5)} = [O_l^{(5)} \quad O_r^{(5)}]^T = \left[\frac{\sum_{j=1}^M O_{jl}^{(4)} T_j' \quad \sum_{j=1}^M O_{jr}^{(4)} T_j^r}{\sum_{j=1}^M O_{jl}^{(4)} \quad \sum_{j=1}^M O_{jr}^{(4)}} \right]^T. \quad (24)$$

Layer 6: Output Layer

Layer 6 is the output layer which is used to implement the defuzzification operation. Each node is the actual output that will be pumped out this system. From (15), the crisp output is

$$O^{(6)} = \frac{O_l^{(5)} + O_r^{(5)}}{2}. \quad (25)$$

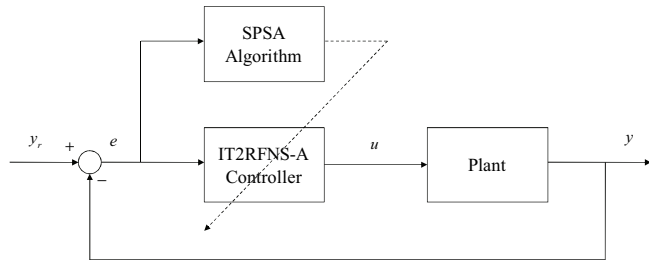


Figure 3. Control scheme for nonlinear system.

C. Training of IT2RFNS-A by A Stable SPSA Algorithm

Consider the nonlinear control problem, our goal is to generate a proper control signal $u(k)$ such that the system output $y(k)$ follows the desired trajectory $y_r(k)$, where k is the time index of discrete-time case. The IT2RFNS-A with the stable SPSA algorithm plays the role of controller for nonlinear plant, the adaptive control scheme is shown in Fig. 3. For convenience, we consider the single-output case and the tracking error is defined as

$$e(k) = y_r(k) - y(k). \quad (26)$$

Then, we define the objective function (error cost function) as

$$E(k) = \frac{1}{2} e^2(k) = \frac{1}{2} (y_r(k) - y(k))^2. \quad (27)$$

Our goal is to generate the control signal $u(k)$ such that the tracking error approaching to zero, i.e., minimize the objective function $E(k)$. By using the SPSA algorithm, the parameters of IT2RFNS-A is adjusted such that the error defined in equation (27) is minimized. From the well-known gradient descent method and equation (5), the parameter update law can be written as

$$W(k+1) = W(k) + \Delta W(k) = W(k) + a_k \left(-\frac{\partial E(k)}{\partial W} \right) \quad (28)$$

where W is the tuning parameters of IT2RFNS-A. Then,

$$\frac{\partial E(k)}{\partial W} = -e(k) \frac{\partial e(k)}{\partial W}. \quad (29)$$

Herein, we adopt the SPSA algorithm to reduce the computational complexity of parameters update. The parameters update laws can be represented as

$$W(k+1) = W(k) + a_k e(k) g(W(k)) \quad (30)$$

where

$$g(W(k)) = \frac{e(W(k) + c_k \Delta_k) - e(W(k))}{c_k} [\Delta_{k1}^{-1} \quad \Delta_{k2}^{-1} \quad \dots \quad \Delta_{kp}^{-1}]^T. \quad (31)$$

$e(W(k) + c_k \Delta_k)$ denotes the tracking error between desired output and system output resulted by the IT2RFNS-A with perturbed tuning parameters. Note that equation (31) does not need to calculate the system gradient information, this simplifies the computational effort of nonlinear control.

D. Stability Analysis

Herein, we develop the stable theorem for selecting appropriate learning step length a_k . The choice of learning step length is very important for convergence. Hence, we employ the Lyapunov stability analysis approach to have convergence condition and the optimal learning step length for IT2RFNS-A.

Theorem 1. Let a_k be the learning step length of tuning parameters for the IT2RFNS-A. Consider the nonlinear control problem using IT2RFNS-A (shown in Fig. 3), the asymptotic convergence of the closed-loop system is guarantee if the learning step length is chosen satisfying

$$0 < a_k < \frac{2}{|g(k)|^2}, \text{ for all } k \quad (32)$$

where $g(W(k)) = \frac{e(W(k) + c_k \Delta_k) - e(W(k))}{c_k} [\Delta_{k1}^{-1} \Delta_{k2}^{-1} \dots \Delta_{kp}^{-1}]^T$

is the gradient estimation using SPSA approach. In addition, the faster convergence can be obtained by using the following optimal time-varying learning step length

$$a_k^* = \frac{1}{|g(k)|^2}. \quad (33)$$

Proof: First, we define the discrete-time Lyapunov function as follows

$$V(k) = E(k) = \frac{1}{2} e^2(k) = \frac{1}{2} (y_r(k) - y(k))^2 \quad (34)$$

where $e(k)$ represents the tracking error. Then, the change of the Lyapunov function is

$$\Delta V(k) = V(k+1) - V(k) = \frac{1}{2} (e^2(k+1) - e^2(k)). \quad (35)$$

According to the Lyapunov stability theorem, if the change of the positive definite Lyapunov function, denoted $\Delta V(k)$, satisfies the condition $\Delta V(k) < 0, \forall k$, then the asymptotical stability is guaranteed [1, 2]. Hence, our objective is to select the proper learning step length such that $\Delta V(k) < 0, \forall k$. This implies that $V(k)$ will converge to zero when k approaches to infinity. By [1, 2, 28], the error difference can be represented as

$$\Delta e(k) = e(k+1) - e(k) \equiv \left[\frac{\partial e(k)}{\partial W} \right]^T \Delta W \quad (36)$$

where ΔW denotes the change of W . From equations (28) and (30), we obtain

$$\Delta W \equiv -a_k e(k) \frac{\partial e(k)}{\partial W} \equiv a_k e(k) g(k). \quad (37)$$

Thus,

$$\begin{aligned} \Delta V(k) &= \frac{1}{2} [e^2(k+1) - e^2(k)] \\ &= \frac{1}{2} [e(k+1) - e(k)] \cdot [e(k+1) + e(k)] \\ &= \frac{1}{2} \Delta e(k) \cdot [2e(k) + \Delta e(k)] \\ &= \Delta e(k) \cdot \left(e(k) + \frac{1}{2} \Delta e(k) \right) \\ &= \left[\frac{\partial e(k)}{\partial W} \right]^T a_k e(k) g(k) \cdot \left\{ e(k) + \frac{1}{2} \left[\frac{\partial e(k)}{\partial W} \right]^T a_k e(k) g(k) \right\} \\ &= -a_k e^2(k) |g(k)|^2 \left(1 - \frac{1}{2} a_k |g(k)|^2 \right). \end{aligned} \quad (38)$$

Note that $P(k)$ is positive for all $k > 0$ and let $P(k) = a_k |g(k)|^2$. Thus,

$$\Delta V(k) = -e^2(k) P(k) \left(1 - \frac{1}{2} P(k) \right). \quad (39)$$

Recall that the asymptotical stability of the IT2RFNS-A is guaranteed if $\Delta V(k) < 0, \forall k > 0$. Thus, $\left(1 - \frac{1}{2} P(k) \right)$ should be positive such that $\Delta V(k) < 0, \forall k$. Therefore, we obtain the stability condition for a_k

$$0 < a_k < \frac{2}{|g(k)|^2}. \quad (40)$$

The asymptotic stability is guaranteed if a_k is chosen to satisfy equation (40). In addition, we would like to find a condition for a_k that guarantees the fast convergence. From (38) and (39), we have

$$\begin{aligned} e^2(k+1) &= e^2(k) - e^2(k) P(k) (2 - P(k)) \\ &= e^2(k) [1 - 2P(k) + P^2(k)] \\ &= e^2(k) [P(k) - 1]^2. \end{aligned} \quad (41)$$

The minimum of $e^2(k+1)$ is achieved when $P(k) = 1$. Hence, the time-varying optimal learning step length is

$$a_k^* = \frac{1}{|g(k)|^2}. \quad (42)$$

■

IV. SIMULATION RESULTS

Consider the control problem of Chua's chaotic circuit. Our purpose is to develop an IT2RFNS-A controller to generate the proper control signal such that the system output y can follow a given reference trajectory y_r by the scheme as shown in the Fig. 3.

The typical Chua's chaotic circuit consists of one inductor, two capacitors, one linear resistor, and one piecewise-linear resistor [30, 31]. It has been shown that this circuit holds very rich nonlinear dynamics such as chaos and bifurcations. The dynamic equation of Chua's circuit are described as

$$\begin{aligned} \frac{dv_{C_1}}{dt} &= (C_1)^{-1} \left(\frac{v_{C_2} - v_{C_1}}{R} - \lambda \right) \\ \frac{dv_{C_2}}{dt} &= (C_2)^{-1} \left(\frac{v_{C_1} - v_{C_2}}{R} - i_L \right) \\ \frac{di_L}{dt} &= (L)^{-1} (-v_{C_1} - R_o i_L) \end{aligned} \quad (43)$$

where voltages v_{C_1} , v_{C_2} , and current i_L are state variables; R_o is a constant; λ denotes the nonlinear resistor which is a function of voltage across two terminals of C_1 . The is λ defined as a cubic function as

$$\lambda = a v_{C_1} + c (v_{C_1})^3 \quad (a > 0, c < 0). \quad (44)$$

According to [31], equation (43) can be transferred into the canonical form as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \cdot (f + g \cdot u + d) \quad (45)$$

$$y = x_1$$

where $f = \frac{14}{1805}x_1 - \frac{168}{9025}x_2 + \frac{1}{38}x_3 - \frac{2}{45}\left(\frac{28}{361}x_1 + \frac{7}{95}x_2 + x_3\right)^3$, $g = 1$, and d is the external disturbance and is assumed to be a square-wave with amplitude ± 0.5 and period 2π . Herein, the objective is to track the reference trajectory $y_r = 1.5\sin(t)$. We choose the initial state value as $x(0) = [0 \ 0 \ 1]^T$ and sampling time as 0.01 second.

For comparison, the same simulations are done by using IT2FNS-A and the case without optimal learning step length. The used coefficients are listed in Table I. The simulation results of trajectories are shown in Fig. 4. Figure 4(a) shows the phase plant trajectory after controlled, Fig. 4(b)-4(d) shows the state trajectories x_1 , x_2 , and x_3 , respectively (solid line: actual system output; dashed line: reference trajectory). Figure 5 depicts the control effort of the IT2RFNS-A controller. The final fuzzy rules are

R^1 : IF z_{11} is $\tilde{F}_{11}$ and z_{21} is $\tilde{F}_{21}$ and z_{31} is $\tilde{F}_{31}$,

$$\text{THEN } Y_1 = [-7.6414 - 0.2130] + [-5.6236 - 0.3833]x_1 \\ + [8.5389 \ 9.0126]x_2 + [-0.7853 \ 1.6596]x_3$$

where $\theta_{11} = [-4.3323 \ 9.3274]$, $\theta_{21} = [-2.0292 \ 2.6543]$, $\theta_{31} = [-8.0895 \ 4.3461]$, $\bar{\omega}_1 = [-10.0520 - 9.9033]$, and $\underline{\omega}_1 = [-10.2740 - 9.3170]$.

R^2 : IF z_{12} is $\tilde{F}_{12}$ and z_{22} is $\tilde{F}_{22}$ and z_{32} is $\tilde{F}_{32}$,

$$\text{THEN } Y_2 = [0.4022 \ 14.6956] + [-0.7847 \ 13.4972]x_1 \\ + [4.2989 \ 5.6034]x_2 + [14.6956 \ 2.0163]x_3$$

where $\theta_{12} = [-10.2843 \ 1.1492]$, $\theta_{22} = [-2.1777 \ 9.1312]$, $\theta_{32} = [-6.9073 - 0.5194]$, $\bar{\omega}_2 = [-9.6955 \ 2.2647]$, and $\underline{\omega}_2 = [-5.7276 - 7.5890]$.

R^3 : IF z_{13} is $\tilde{F}_{13}$ and z_{23} is $\tilde{F}_{23}$ and z_{33} is $\tilde{F}_{33}$,

$$\text{THEN } Y_3 = [-8.4356 - 6.3436] + [-9.0422 - 4.7307]x_1 \\ + [4.5459 \ 9.6610]x_2 + [-9.1732 \ 2.0163]x_3$$

where $\theta_{13} = [-2.0828 \ 3.3525]$, $\theta_{23} = [-6.9332 - 2.3628]$, $\theta_{33} = [-7.1283 - 7.4267]$, $\bar{\omega}_3 = [-9.5861 - 0.3531]$, and $\underline{\omega}_3 = [-9.9473 - 8.7061]$.

R^4 : IF z_{14} is $\tilde{F}_{14}$ and z_{24} is $\tilde{F}_{24}$ and z_{34} is $\tilde{F}_{34}$,

$$\text{THEN } Y_4 = [3.9646 \ 11.7177] + [5.1486 \ 9.3539]x_1 \\ + [-4.5256 \ 6.0764]x_2 + [-0.9234 \ 6.5402]x_3$$

where $\theta_{14} = [-7.7789 - 7.6991]$, $\theta_{24} = [-8.7967 - 5.7106]$, $\theta_{34} = [-7.1095 - 6.3978]$, $\bar{\omega}_4 = [-8.7774 \ 1.2302]$, and $\underline{\omega}_4 = [-7.3826 - 4.7145]$.

The corresponding MFs are depicted in Fig. 6.

We can observe that the proposed method is valid to control a nonlinear dynamic system. The comparison results are shown in Table II which is obtained by 10 independent runs. From Table II, we can easily find that the IT2RFNS-A performs better than IT2FNS-A. This result shows that the IT2RFNS-A has ability to cope with dynamic problems. Observe the performance of the case with optimal learning step length from Table II, we again demonstrate that adopting optimal learning step length guarantees the efficient training of IT2RFNS-A.

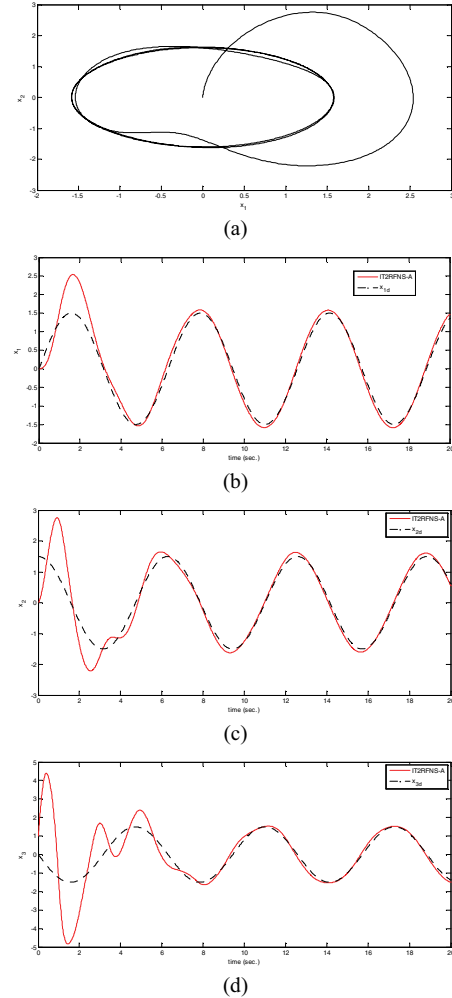


Figure 4. Simulation results: (a) phase plant trajectory of IT2RFNS-A, (b) state x_1 and reference trajectory y_r , (c) state x_2 and reference trajectory $\dot{y}_r$, and (d) state x_3 and reference trajectory $\ddot{y}_r$.

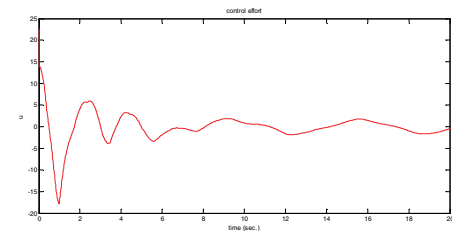


Figure 5. Control effort of IT2RFNS-A controller of Chua's chaotic circuit.

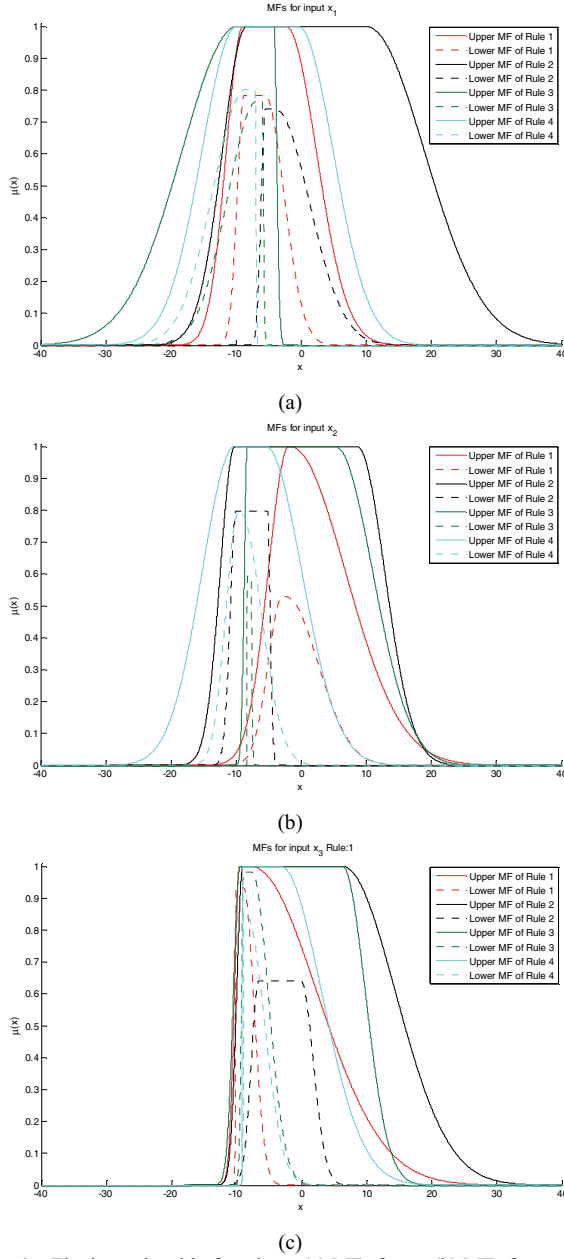


Figure 6. Final membership functions: (a) MFs for x_1 , (b) MFs for x_2 , and (c) MFs for x_3 .

TABLE I. THE USED COEFFICIENTS.

	IT2FNS-A	IT2RFNS-A	IT2RFNS-A
α	0.01	0.01	optimal learning step length (33)
α	0.602	0.602	
A	200	200	
c	1	1	1
γ	0.101	0.101	0.101
rule number	4	4	4

TABLE II. COMPARISON RESULTS OF CONTROL PERFORMANCE.

MSE	IT2FNS-A	IT2RFNS-A	IT2RFNS-A (optimal a_k)
Worst	2.5485	2.1833	0.0663
Average	0.9189	0.3265	0.0661
Best	0.2474	0.0675	0.0659

V. CONCLUSION

In this paper, we have proposed an interval type-2 recurrent fuzzy system with asymmetric membership functions (IT2RFNS-A) and its training scheme using the proposed stable SPSA algorithm. In addition, we adopt the Lyapunov theorem to derive the appropriate range of learning step length for SPSA to guarantee the stability of the closed-loop system for nonlinear control. In addition, we also derive the optimal learning step length which can ensure the efficient training for IT2RFNS-A. The feasibility and the effectiveness of the proposed method have been demonstrated by illustration example. The simulations on the control of the Chua's chaotic circuit are done and shows the following advantages of proposed approach; (a) The IT2RFNS-A can reduce the fuzzy rules, that is, only few rules can achieve the specific performance or even better; (b) The IT2RFNS-A can catch the dynamic response of the system, i.e., it is capable of coping with dynamic problems; (c) By using the proposed SPSA algorithm, the gradient information is unnecessary. In other words, a great deal of computational effort is saved. (d) For control problems, the proposed method avoids approximating the system sensitivity which leads to inaccuracy. (e) Adopting optimal learning step length improves the performance of training IT2RFNS-A.

ACKNOWLEDGMENT

This work was partially supported by the National Science Council, Taiwan, R.O.C., under contract No: NSC-97-2221-E-155-033-MY3.

REFERENCES

- [1] Y. C. Chen and C. C. Teng, "A model reference control structure using a fuzzy neural network," *Fuzzy Sets Syst.*, Vol. 73, pp. 291–312, 1995.
- [2] 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.
- [3] Y. Gao and M. J. Er, "Online adaptive fuzzy neural identification and control of a class of MIMO nonlinear systems," *IEEE Trans. Fuzzy Syst.*, Vol. 11, No. 4, pp. 462–477, Aug. 2003.
- [4] R. J. Wai and P. C. Chen, "Intelligent tracking control for robot manipulator including actuator dynamics via TSK-type fuzzy neural network," *IEEE Trans. Fuzzy Syst.*, Vol. 12, No. 4, pp. 552–560, Aug. 2004.
- [5] A. Lotfi and A. C. Tsoi, "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.
- [6] 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.
- [7] P. Z. Lin and T. T. Lee, "Robust self-organizing fuzzy-neural control using asymmetric Gaussian membership functions," *International Journal of Fuzzy Systems*, Vol. 9, No. 2, pp. 77–86, 2007.
- [8] 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.

- [9] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning-1," *Information Sciences*, Vol. 8, pp. 199-249, 1975.
- [10] Q. Liang and J. M. Mendel, "Interval type-2 fuzzy logic systems: Theory and design," *IEEE Trans. Fuzzy Syst.*, Vol. 8, No. 5, pp. 535-550, Oct. 2000.
- [11] J. M. Mendel, "On the importance of interval sets in type-2 fuzzy logic systems," in *Proc. Joint 9th IFSA World Congr. 20th NAFIPS Int. Conf.*, Vancouver, BC, Canada, pp. 1647-1652, Jul. 25-28, 2001.
- [12] O. Castillo, N. Cazarez, and P. Melin, "Design of Stable Type-2 Fuzzy Logic Controller based on a Fuzzy Lyapunov Approach," *IEEE International Conference on Fuzzy Systems*, pp. 2331-2336, 2006.
- [13] O. Castillo, "Interval Type-2 Fuzzy Logic for Control Application," *IEEE International Conference on Granular Computing*, pp. 79-84, 2010.
- [14] H. A. Hagra, "A Hierarchical Type-2 Fuzzy Logic Control Architecture for Autonomous Mobile Robot," *IEEE Trans. on Fuzzy Systems*, Vol. 12, No. 4, 2004.
- [15] H. A. Hagra, "Type-2 Fuzzy Control: A New Generation of Fuzzy Controllers," *IEEE Comput. Intell. Mag.*, Vol. 2, No. 1, pp. 30-43, Feb. 2007.
- [16] D. Wu and W. W. Tan, "A Simplified Architecture for Type-2 FLSs and Its Application to Nonlinear Control," *IEEE Conf. on Cybern. and Intell. Syst.*, pp. 485-490, Dec. 2004.
- [17] X. T. Chen and W. W. Tan, "A Type-2 Fuzzy Logic Controller for Dynamic Positioning Systems," *IEEE International Conference on Control and Automation*, pp. 1013-1018, June 2010.
- [18] C. H. Lee, J. L. Hong, Y. C. Lin, and W. Y. Lai, "Type-2 fuzzy neural network systems and learning," *International Journal of Computational Cognition*, Vol. 1, No. 4, pp. 79-90, 2003.
- [19] C. H. Lee, Y. C. Lin, and W. Y. Lai, "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.
- [20] C. H. Lee and Y. C. Lin, "Control of nonlinear uncertain system using type-2 fuzzy neural network and adaptive filter," *IEEE International Conf. on Networking, Sensing, and Control*, Vol. 2, pp. 1177-1182, 2004.
- [21] C. H. Lee, Y. C. Lin, "An adaptive type-2 fuzzy neural controller for nonlinear uncertain systems," *International Journal of Control and Intelligent systems*, Vol. 12, No. 1, pp. 41-50, 2005.
- [22] C. H. Lee, F. Y. Chang, and C. T. Lee, "Species-based hybrid of electromagnetism-like mechanism and back-propagation algorithms for an interval type-2 fuzzy system design," *Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2010*, IMECS 2010, 17-19 March, 2010, Hong Kong, pp. 140-145.
- [23] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Trans. Autom. Control*, Vol. 37, No. 3, pp. 332-341, Mar. 1992.
- [24] J. C. Spall and J. A. Cristion, "Model free control of nonlinear stochastic systems with discrete-time measurements," *IEEE Trans. Autom. Control*, Vol. 43, No. 9, pp. 1198-1210, Sep. 1998.
- [25] J. C. Spall, *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*, John Wiley and Sons, Hoboken, NJ, 2003.
- [26] J. M. Mendel, *Uncertain Rule-based Fuzzy Logic Systems: Introduction and New Directions*, Upper Saddle River, Prentice-Hall, NJ, 2001.
- [27] J. R. Castro, O. Castillo, P. Melin, and A. R. Diaz, "A hybrid learning algorithm for a class of interval type-2 fuzzy neural networks," *Information Sciences*, Vol. 179, No. 13, pp. 2175-2193, 2009.
- [28] T. Yabuta and T. Yamada, "Learning control using neural networks," *Proc. IEEE Int. Conf. Robotics and Automation*, Sacramento, CA, Apr. pp. 740-745, 1991.
- [29] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical system using neural network," *IEEE Trans. On Neural Networks*, Vol. 1, No. 1, pp. 4-27, 1990.
- [30] Y. H. Joo, L. S. Shieh, and G. R. Chen, "Hybrid state space fuzzy model based controller with dual-rate sampling for digital control of chaotic systems," *IEEE Trans. on Fuzzy Systems*, Vol. 7, No. 4, pp. 394-408, 1999.
- [31] C. H. Wang, T. C. Lin, T. T. Lee, and H. L. Liu, "Adaptive hybrid intelligent control for uncertain nonlinear dynamical systems," *IEEE Trans. on Systems, Man, Cybernetics Part-B*, Vol. 32, No. 5, pp. 583-597, 2002.