Research Article

Interval Type-2 Recurrent Fuzzy Neural System for Nonlinear Systems Control Using Stable Simultaneous Perturbation Stochastic Approximation Algorithm

Ching-Hung Lee and Feng-Yu Chang

Department of Electrical Engineering, Yuan Ze University, Chung-Li 32003, Taiwan

Correspondence should be addressed to Ching-Hung Lee, chlee@saturn.yzu.edu.tw

This paper proposes a new type fuzzy neural systems, denoted IT2RFNS-A (interval type-2 recurrent fuzzy neural system with asymmetric membership function), for nonlinear systems identification and control. To enhance the performance and approximation ability, the triangular asymmetric fuzzy membership function (AFMF) and TSK-type consequent part are 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 results for the chaotic system identification and the control of Chua's chaotic circuit are shown to illustrate the feasibility and effectiveness of the proposed method.

1. Introduction

In the 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 (MFs) 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 reduce the 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 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 the interval type-2 fuzzy sets (IT2 FSs) are proposed [10]. The computations associated with IT2 FSs are very manageable which makes it quite practical [11]. In our previous research [12–15], we proposed a type-2 fuzzy neural network with asymmetric membership function (T2FNN-A) which combines interval type-2 fuzzy logic system with neural network. Then, in order to improve the efficiency of the T2FNN-A, we proposed an interval type-2 fuzzy neural system with AFMFs (IT2FNS-A) which utilizes several enhanced techniques such as Takagi-Sugeno-Kang fuzzy logic system (TSK FLS) and embedded type-reduction network layer [16]. 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. Thus, using IT2FNS-A to process dynamic problems is inefficient. Many results have been shown that recurrent system can learn and memorize information to provide better performance [2, 17–20]. In this paper, we propose an interval type-2 recurrent fuzzy neural system with AFMFs (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 neural networks, the back-propagation (BP) algorithm is widely used. However, the differential information of system is difficult to obtain due to the piece-wise continuous property of triangular AFMFs and since there are many adjustable parameters in the IT2RFNS-A. Herein, we adopt the simultaneous perturbation stochastic approximation (SPSA) algorithm to derive the update laws of the proposed IT2RFNS-A. The SPSA algorithm is based on the approximation to gradient information from the measurements of the objective function [21–23]. Hence, a great deal of computational effort is saved. However, due to the stochastic characteristic of SPSA algorithm, we cannot guarantee that every searching 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 2, the construction of triangular AFMFs and the SPSA algorithm are introduced. Section 3 illustrates the proposed IT2RFNS-A and the stable SPSA learning algorithm. The simulation results on chaotic system identification and control of the Chua's chaotic circuit are shown in Section 4. Finally, the conclusion is given.

2. Preliminaries

In this section, we briefly introduce some prerequisite material including the interval type-2 asymmetric fuzzy membership function (IT2 AFMF) and simultaneous perturbation stochastic approximation (SPSA) algorithm.

2.1. Interval Type-2 Asymmetric Fuzzy Membership Function (IT2 AFMF)

The interval type-2 membership function (IT2 MF) is considered to be a special case of a general type-2 membership function (T2 MF) which simplifies the computational effort
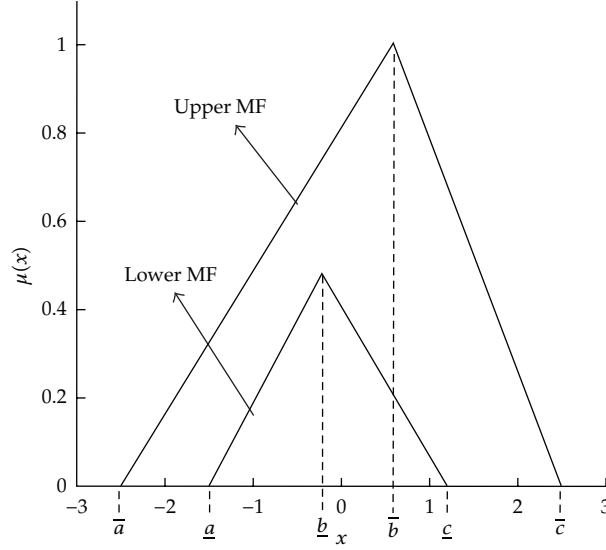
Figure 1: Construction of interval type-2 asymmetric fuzzy membership function.

significantly [10, 11]. In general, the symmetric MF is used for simplification. However, the symmetric MF only holds either uncertain mean or uncertain variance. In addition, tuning the parameter of MFs symmetrically may result in low precision. The AFMF can treat these problems.

In this paper, the triangular fuzzy MFs are used to construct the interval type-2 asymmetric fuzzy membership functions (IT2 AFMFs) due to their lower computational effort. Each IT2 AFMF consists of upper and lower MFs, as shown in Figure 1. The upper MF is defined as

$$
\overline{\mu}_{\widetilde{F}}(x) = \begin{cases} \dfrac{x - \overline{a}}{\overline{b} - \overline{a}}, & \overline{a} \leq x \leq \overline{b}, \\[2mm] \dfrac{\overline{c} - x}{\overline{c} - \overline{b}}, & \overline{b} \leq x \leq \overline{c}, \\[2mm] 0, & \text{otherwise}, \end{cases} \tag{2.1}
$$

where $\overline{a}$, $\overline{b}$, and $\overline{c}$ denote the positions of three corners satisfying $\overline{a} \leq \overline{b} \leq \overline{c}$. Similarly, the lower MF is defined as

$$
\underline{\mu}_{\widetilde{F}}(x) = \begin{cases} \lambda \cdot \dfrac{x - \underline{a}}{\underline{b} - \underline{a}}, & \underline{a} \leq x \leq \underline{b}, \\[2mm] \lambda \cdot \dfrac{\underline{c} - x}{\underline{c} - \underline{b}}, & \underline{b} \leq x \leq \underline{c}, \\[2mm] 0, & \text{otherwise}, \end{cases} \tag{2.2}
$$

where $\underline{a}$, $\underline{b}$, and $\underline{c}$ denote the positions of three corners satisfying $\underline{a} \le \underline{b} \le \underline{c}$ and λ denotes the magnitude of lower MF which should be limited between 0.5 and 1 to avoid the invalid result (a small firing strength). As above description, the following restrictions should be constrained to avoid unreasonable IT2 AFMFs: $\overline{a} \le \overline{b} \le \overline{c}$, $\underline{a} \le \underline{b} \le \underline{c}$, $\overline{a} \le \underline{a}$, $\underline{c} \le \overline{c}$, $\overline{a} + \lambda(\overline{b} - \overline{a}) \le \underline{b} \le \overline{c} - \lambda(\overline{c} - \overline{b})$.

2.2. Simultaneous Perturbation Stochastic Approximation (SPSA) Algorithm

This section introduces the SPSA algorithm briefly. The detailed description can be found in literature [23]. Consider the optimization problem with an objective function $f(W)$, the SPSA algorithm updates W by

$$W(k+1) = W(k) - a_k g(W(k)), \tag{2.3}$$

where $g(\cdot)$ is the estimated gradient result of objective function $f(\cdot)$ with respect to W, that is, $\partial f(W)/\partial W \approx g(W)$. a_k denotes the learning step length which is decreased over iterations with $a_k = a/(k+A)^\alpha$, where a, A, and α are positive configuration coefficients [23]. The SPSA approach estimates the gradient, $g(\cdot)$, using the following method. Assume that the dimension of parameter W is p. Let $\Delta_k = [\Delta_{k1}\Delta_{k2}\cdots\Delta_{kp}]$ be a p-dimensional vector whose element is mutually independent zero-mean random variable. Then, the estimation of the gradient at kth iteration can be computed by

$$g(W(k)) = \frac{f(W(k) + c_k\Delta_k) - f(W(k))}{c_k}\left[\Delta_{k1}^{-1}\Delta_{k2}^{-1}\cdots\Delta_{kp}^{-1}\right]^T, \tag{2.4}$$

where c_k is gain sequence that is also decreased with $c_k = c/(k+1)^\gamma$, where c and γ are nonnegative configuration coefficients [23]. Obviously, all elements of W are perturbed simultaneously and only two measurements 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 piecewise continuous property of AFMFs and large number of adjustable parameters. Herein, we adopt the SPSA algorithm to derive the stable learning for guaranteeing the convergence and stability of the closed-loop systems.

3. Interval Type-2 Recurrent Fuzzy Neural System with Asymmetric Membership Function (IT2RFNS-A)

3.1. Fuzzy Reasoning of IT2RFNS-A

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

$$R^j : \text{ if } u_{1j} \text{ is } \widetilde{F}_{1j}, \ldots, u_{nj} \text{ is } \widetilde{F}_{nj}, \quad \text{then } Y_j = C_{j0} + C_{j1}x_1 + C_{j2}x_j + \cdots + C_{jn}x_n, \tag{3.1}$$

where u_{ij} is input linguistic variable of the jth rule, $\tilde{F}_{ij}$ are interval type-2 antecedent fuzzy sets, C_{ji} are consequent interval fuzzy set, x_i is the network input, and Y_j is the output of the jth rule. Note that the input linguistic variable u_{ij} has the system input term and the past information which is introduced in Section 3.2. As the above description of T2 FLSs, the membership grade is an interval valued set which consists of the lower and upper membership grades, that is,

$$\mu_{\tilde{F}_{ij}}(u_{ij}) = \left[\underline{\mu}_{\tilde{F}_{ij}}(u_{ij}) \ \ \overline{\mu}_{\tilde{F}_{ij}}(u_{ij})\right] \tag{3.2}$$

and the consequent part is

$$C_{ji} = \left[c_{ji} - s_{ji} \ \ c_{ji} + s_{ji}\right], \tag{3.3}$$

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

$$F_j(\mathbf{u}) = \left[\underline{f}_j(\mathbf{u}) \ \ \overline{f}_j(\mathbf{u})\right], \tag{3.4}$$

where $\underline{f}_j(\mathbf{u}) = \underline{\mu}_{\tilde{F}_{1j}}(u_{1j}) \times \cdots \times \underline{\mu}_{\tilde{F}_{nj}}(u_{nj})$ and $\overline{f}_j(\mathbf{u}) = \overline{\mu}_{\tilde{F}_{1j}}(u_{1j}) \times \cdots \times \overline{\mu}_{\tilde{F}_{nj}}(u_{nj})$. Thus, the consequent part of the jth rule is

$$
\begin{aligned}
y_j^l &= \left(c_{j0} + \sum_{i=1}^{n} c_{ji} u_{ij}\right) - \left(s_{j0} + \sum_{i=1}^{n} s_{ji} |u_{ij}|\right), \\
y_j^r &= \left(c_{j0} + \sum_{i=1}^{n} c_{ji} u_{ij}\right) + \left(s_{j0} + \sum_{i=1}^{n} s_{ji} |u_{ij}|\right).
\end{aligned}
\tag{3.5}
$$

By applying the Extension principle [24], the output of the FLS is

$$Y_{\text{TSK}}(\mathbf{u}) = \left[y^l y^r\right] = \frac{\int_{y_1 \in [y_1^l \ y_1^r]} \cdots \int_{y_M \in [y_M^l \ y_M^r]} \int_{f_1 \in [\underline{f}_1 \ \overline{f}_1]} \cdots \int_{f_M \in [\underline{f}_M \ \overline{f}_M]} 1}{\sum_{j=1}^{M} f_j y_j / \sum_{j=1}^{M} f_j}. \tag{3.6}$$

To compute $Y_{\text{TSK}}(\mathbf{u})$, we need to compute its two end-points y^l and y^r by type-reduction operation. Karnik and Mendel developed an iterative algorithm which is known as KM algorithm to compute these two end-points [24, 25]. A type reducer combines all fired-rule output sets in some way, just like a type-2 defuzzifier combines the type-1 rule output sets, which leads to a T1 FS that is called a type-reduced set. Herein, we need to compute the left-end point y^l and right-end point y^r by utilizing the KM algorithm [24, 25]. In the KM algorithm, the left-end and right-end points are represented as

$$y^l = \frac{\sum_{i=1}^{L} \overline{f}_i y_i + \sum_{L+1}^{M} \underline{f}_i y_i}{\sum_{i=1}^{L} \overline{f}_i + \sum_{L+1}^{M} \underline{f}_i}, \qquad y^r = \frac{\sum_{i=1}^{R} \underline{f}_i y_i + \sum_{R+1}^{M} \overline{f}_i y_i}{\sum_{i=1}^{R} \underline{f}_i + \sum_{R+1}^{M} \overline{f}_i}. \tag{3.7}$$

Then, we have to find the proper switch point value L and R by iterative procedure where more detail can be referred to [24, 25]. However, this iterative procedure for finding switch point values is time wasting. Hence, in the proposed IT2RFNS-A, a simple weight-average method is used to approximate the L and R so that the iterative procedure is unnecessary. That is, we can calculate the left-most and right-most point of firing strength by

$$f_j^l = \frac{\overline{\omega}_j^l \overline{f_j} + \underline{\omega}_j^l \underline{f}_j}{\overline{\omega}_j^l + \underline{\omega}_j^l}, \qquad f_j^r = \frac{\overline{\omega}_j^r \overline{f_j} + \underline{\omega}_j^r \underline{f}_j}{\overline{\omega}_j^r + \underline{\omega}_j^r}, \tag{3.8}$$

where $\overline{\omega}_j^l$, $\underline{\omega}_j^l$, $\overline{\omega}_j^r$, and $\underline{\omega}_j^r$ are adjustable weights. Then, we can obtain the following left-end point and right-end point of the output of interval type-2 fuzzy inference system

$$y^l = \frac{\sum_{j=1}^{M} f_j^l y_j^l}{\sum_{j=1}^{M} f_j^l}, \qquad y^r = \frac{\sum_{j=1}^{M} f_j^r y_j^r}{\sum_{j=1}^{M} f_j^r}. \tag{3.9}$$

Note that the above simplified type-reduction technique is adopted in layer 4 (left-most and right-most layer) and feedback layer of IT2RFNS-A system. This simplifies the computational effort in type-reduction procedure. Finally, we defuzzify the type-reduced set to get a crisp output, that is,

$$y(\mathbf{u}) = \frac{y^l + y^r}{2}. \tag{3.10}$$

3.2. Network Structure of IT2RFNS-A

The proposed IT2RFNS-A consists of six feed-forward layers and a feedback one. Layer 1 accepts input variables. Layer 2 is used to calculate the IT2 AFMF grade. The feedback layer embedded in Layer 2 is used to store the past information. Layer 3 forms the fuzzy rule base. Layer 4 introduces the simplified type-reduction scheme, called left-most and right-most layer. The TSK-type consequent part is implemented into Layer 5. Layer 6 is the output layer.

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

Layer 1: Input Layer

For the ith node of layer 1, the net input and output are represented as
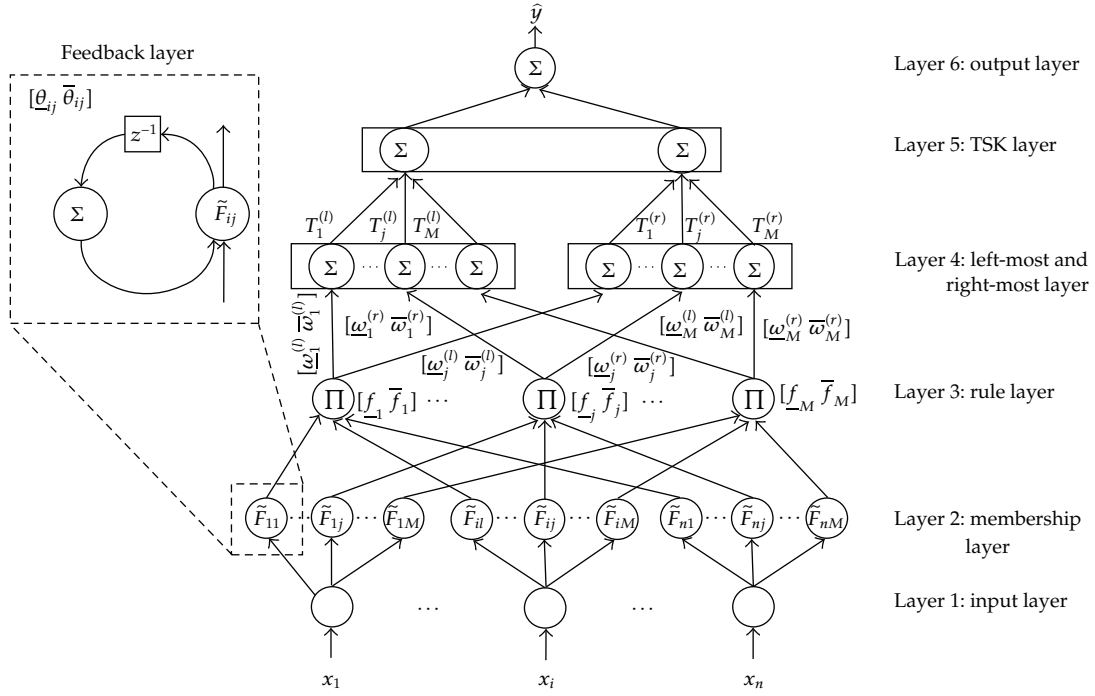
$$O_i^{(1)} = x_i, \tag{3.11}$$

Figure 2: Diagram of the proposed IT2RFNS-A system.

where x_i represents the ith input to the ith node. The nodes in this layer only transmit input to the next layer directly.

Layer 2: Membership Layer

In this layer, each node performs a triangular IT2 AFMF, that is,

$$
\begin{aligned}
O_{ij}^{(2)} &= \mu_{\widetilde{F}_{ij}}\left(O_i^{(1)} + O_i^{(f)}\right) \\
&= \left[\underline{O}_{ij}^{(2)} \quad \overline{O}_{ij}^{(2)}\right]^T \\
&= \left[\underline{\mu}_{\widetilde{F}_{ij}}\left(O_i^{(1)} + O_i^{(f)}\right) \quad \overline{\mu}_{\widetilde{F}_{ij}}\left(O_i^{(1)} + O_i^{(f)}\right)\right]^T,
\end{aligned}
\tag{3.12}
$$

where the subscript ij indicates the jth term of the ith input and $\mu_{\widetilde{F}_{ij}}$ is an IT2 AFMF as shown in Figure 1 and (3.2). Note that the output of layer 2 and the feedback weight are interval values. According to the above description and the results of literature [26], the type reduction is embedded in the network. Herein, the output of feedback layer is expressed as

$$
O_i^{(f)}(k) = \frac{\underline{O}_{ij}^{(2)}(k-1)\cdot\underline{\theta}_{ij} + \overline{O}_{ij}^{(2)}(k-1)\cdot\overline{\theta}_{ij}}{\underline{\theta}_{ij} + \overline{\theta}_{ij}},
\tag{3.13}
$$

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

Layer 3: Rule Layer

This layer is used for computing firing strength of fuzzy rule. From (3.4), we obtain

$$\underline{f}_j = \underline{\mu}_{\tilde{F}_{1j}}\left(O_1^{(1)} + O_1^{(f)}\right) \times \cdots \times \underline{\mu}_{\tilde{F}_{nj}}\left(O_n^{(1)} + O_n^{(f)}\right),$$
$$\overline{f}_j = \overline{\mu}_{\tilde{F}_{1j}}\left(O_1^{(1)} + O_1^{(f)}\right) \times \cdots \times \overline{\mu}_{\tilde{F}_{nj}}\left(O_n^{(1)} + O_n^{(f)}\right),$$
(3.14)

where $\underline{\mu}_{\tilde{F}_{ij}}(\cdot)$ and $\overline{\mu}_{\tilde{F}_{ij}}(\cdot)$ are the lower and upper membership grades. Therefore, the operation function in this layer is

$$O_j^{(3)} = \left[\underline{f}_j \ \ \overline{f}_j\right]^T = \left[\underline{O}_j^{(3)} \ \ \overline{O}_j^{(3)}\right]^T = \left[\prod_{i=1}^{n}\underline{O}_{ij}^{(2)} \ \ \prod_{i=1}^{n}\overline{O}_{ij}^{(2)}\right]^T.$$
(3.15)

Layer 4: Left-Most & Right-Most layer

Similar to the feedback layer, the type reduction is integrated into the network structure by calculating their left-most and right-most values. That is, the complicated type-reduction method such as KM algorithm can be reduced as

$$O_j^{(4)} = \left[O_{jl}^{(4)}O_{jr}^{(4)}\right]^T = \left[\frac{\overline{\omega}_j^l\overline{O}_j^{(3)} + \underline{\omega}_j^l\underline{O}_j^{(3)}}{\overline{\omega}_j^l + \underline{\omega}_j^l} \ \ \frac{\overline{\omega}_j^r\overline{O}_j^{(3)} + \underline{\omega}_j^r\underline{O}_j^{(3)}}{\overline{\omega}_j^r + \underline{\omega}_j^r}\right]^T,$$
(3.16)

where the link weights are $\underline{\omega}^l = [\underline{\omega}_1^l \cdots \underline{\omega}_M^l]^T$, $\overline{\omega}^l = [\overline{\omega}_1^l \cdots \overline{\omega}_M^l]^T$, $\underline{\omega}^r = [\underline{\omega}_1^r \cdots \underline{\omega}_M^r]^T$, and $\overline{\omega}^r = [\overline{\omega}_1^r \cdots \overline{\omega}_M^r]^T$. These link weights are adjusted by the proposed stable SPSA learning algorithm. Note that $\underline{\omega}_j^l < \overline{\omega}_j^l$ and $\underline{\omega}_j^r < \overline{\omega}_j^r$.

Layer 5: TSK Layer

From (3.5), the TSK-type consequent part is

$$T_j = \left[T_j^l \ T_j^r\right]^T$$
$$= \left[\left(c_{j0} + \sum_{i=1}^{n} c_{ji}x_i\right) - \left(s_{j0} + \sum_{i=1}^{n} s_{ji}|x_i|\right) \left(c_{j0} + \sum_{i=1}^{n} c_{ji}x_i\right) + \left(s_{j0} + \sum_{i=1}^{n} s_{ji}|x_i|\right)\right]^T.$$
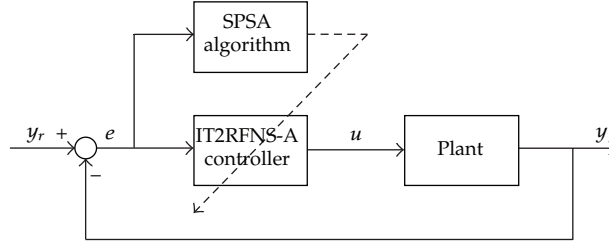(3.17)

Figure 3: Adaptive control scheme for nonlinear system.

Then, the output of this layer is

$$O^{(5)} = \begin{bmatrix} O_l^{(5)} & O_r^{(5)} \end{bmatrix}^T = \begin{bmatrix} \dfrac{\sum_{j=1}^{M} O_{jl}^{(4)} T_j^l}{\sum_{j=1}^{M} O_{jl}^{(4)}} & \dfrac{\sum_{j=1}^{M} O_{jr}^{(4)} T_j^r}{\sum_{j=1}^{M} O_{jr}^{(4)}} \end{bmatrix}^T. \tag{3.18}$$

Layer 6: Output Layer

Layer 6 is the output layer which is used to implement the defuzzification operation. Therefore, the crisp output is

$$O^{(6)} = \frac{O_l^{(5)} + O_r^{(5)}}{2}. \tag{3.19}$$

Obviously, the total design parameters of IT2RFNS-A are $\overline{a}$, $\overline{b}$, $\overline{c}$, $\underline{a}$, $\underline{b}$, $\underline{c}$, λ, $\underline{\omega}_j^l$, $\overline{\omega}_j^l$, $\underline{\omega}_j^r$, $\overline{\omega}_j^r$, c_{ji}, and s_{ji}. These parameters are adjusted by the proposed stable SPSA learning algorithm which guarantees the convergence of IT2RFNS-A systems.

3.3. Training of IT2RFNS-A by the Stable SPSA Algorithm

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

$$e(k) = y_r(k) - y(k). \tag{3.20}$$

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

$$E(k) = \frac{1}{2}e^2(k) = \frac{1}{2}(y_r(k) - y(k))^2. \tag{3.21}$$

The control objective is to generate the control signal $u(k)$ such that the tracking error approaches to zero, that is, to minimize the objective function $E(k)$. From the well-known gradient descent method, the parameter update law can be written as

$$W(k+1) = W(k) + \Delta W(k) = W(k) + a_k\left(-\frac{\partial E(k)}{\partial W}\right), \qquad (3.22)$$

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

$$\frac{\partial E(k)}{\partial W} = -e(k)\frac{\partial e(k)}{\partial W}. \qquad (3.23)$$

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

$$W(k+1) = W(k) + a_k e(k) g(W(k)), \qquad (3.24)$$

where

$$g(W(k)) = \frac{e(W(k) + c_k\Delta_k) - e(W(k))}{c_k}\left[\Delta_{k1}^{-1}\Delta_{k2}^{-1}\cdots\Delta_{kp}^{-1}\right]^T, \qquad (3.25)$$

where $e(W(k) + c_k\Delta_k)$ denotes the tracking error between the desired output and system output resulted by the IT2RFNS-A with perturbed tuning parameters. Note that (3.25) does not calculate the system sensitivity or gradient functions; this simplifies the computational effort.

Note that, in training neural networks, it may not be possible to update all estimated parameters with a single gradient approximation function (3.25). Thus, we should partition the estimated parameters W into several parts, that is, each kind of parameter has its corresponding estimated parameter (e.g., $W^{\overline{\theta}}$ as the estimated link weight of feedback layer). Then, the estimated parameters are updated separately by the SPSA algorithm.

Remark 3.1. As the previous results of [1, 2, 4], the adaptive control laws can be obtained by multiplying system sensitivity $\partial y/\partial u$, that is,

$$\frac{\partial E(k)}{\partial W} = e(k)\frac{\partial(y_r(k) - y(k))}{\partial W} = e(k)(-1)\frac{\partial y(k)}{\partial W} = e(k)\cdot(-1)\cdot\frac{\partial y}{\partial u}\cdot\frac{\partial u}{\partial W}, \qquad (3.26)$$

where u is the control input produced by the IT2RFNS-A, that is, $u = O^{(6)}$. Hence, the gradient of the IT2RFNS-A should be calculated. However, an additional identifier (using a FNN or an IT2RFNS-A) should be developed to find the unknown system's sensitivity (details can be found in [1, 2]). In this way, the approximation accuracy should be guaranteed for training
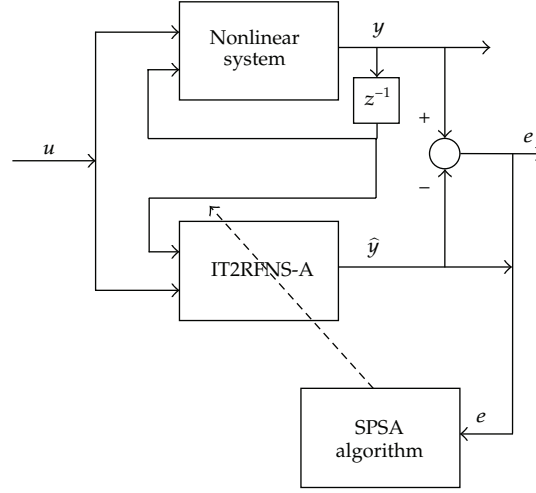
Figure 4: Series-parallel training architecture for system identification.

the IT2RFNS-A controller and the computational effort is complex and huge. Therefore, an approximation term of $\partial E(k)/\partial W \cong -(e + \Delta e)\partial u/\partial W$ and the optimal learning step length are derived to enhance the efficiency and guarantee the stability, where Δe denotes $\dot{e}$ and $e(k) - e(k-1)$ for continuous and discrete time case, respectively.

Remark 3.2. Our proposed approach is also valid for the nonlinear system identification. The series-parallel architecture shown in Figure 4 is adopted. Hence, the inputs of IT2RFNS-A are u and $y(k-1)$ and the IT2RFNS-A output is the estimated output $\hat{y}(k)$. The parameters of IT2RFNS-A are tuned by the proposed stable SPSA algorithm. Thus, the parameter update law is

$$W(k+1) = W(k) + a_k \left[y(k) - \hat{y}(k) \right] \cdot \frac{O^{(6)}(W(k) + c_k\Delta_k) - O^{(6)}(W(k))}{c_k} \left[\Delta_{k1}^{-1}\Delta_{k2}^{-1} \cdots \Delta_{kp}^{-1} \right]^T, \tag{3.27}$$

where $O^{(6)}(W(k) + c_k\Delta_k)$ denotes the output of the IT2RFNS-A with perturbed tuning parameters.

3.4. Stability Analysis

In this section, the convergence theorem for selecting appropriate learning step length a_k is introduced. The choice of learning step length is very important for convergence. If a small value is given for the learning step length, then the convergence of the IT2RFNS-A is guaranteed. However, the convergent speed may be slow. On the other hand, if a large value is selected, then the system may be unstable. Hence, we employ the Lyapunov stability approach to have the condition for convergence and find the optimal learning step length for IT2RFNS-A.

Theorem 3.3. *Let a_k be the learning step length of tuning parameters for the IT2RFNS-A controller. Consider the nonlinear control problem using IT2RFNS-A (shown in Figure 3), the asymptotic convergence of the closed-loop system is guaranteed if the learning step length is chosen satisfying*

$$0 < a_k < \frac{2}{|g(k)|^2}, \quad \forall k, \tag{3.28}$$

where $g(W(k)) = -(e + \Delta e/e) \cdot (O^{(6)}(W(k) + c_k \Delta_k) - O^{(6)}(W(k))/c_k)[\Delta_{k1}^{-1}\Delta_{k2}^{-1}\cdots\Delta_{kp}^{-1}]^T$ is the gradient estimation using SPSA approach. In addition, the faster convergence can be obtained by the following optimal time-varying learning step length

$$a_k^* = \frac{1}{|g(k)|^2}. \tag{3.29}$$

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

$$V(k) = E(k) = \frac{1}{2}e^2(k) = \frac{1}{2}(y_r(k) - y(k))^2, \tag{3.30}$$

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

$$\Delta V(k) = V(k+1) - V(k) = \frac{1}{2}\left(e^2(k+1) - e^2(k)\right). \tag{3.31}$$

According to the Lyapunov stability theorem, if the change of the positive definite Lyapunov function, denoted $\Delta V(k)$, satisfies the condition $\Delta V(k) < 0$, for all k, then the asymptotical stability is guaranteed [1, 2, 27]. Hence, our objective is to select the proper learning step length such that $\Delta V(k) < 0$, for all 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

$$\Delta e(k) = e(k+1) - e(k) \cong \left[\frac{\partial e(k)}{\partial W}\right]^T \Delta W, \tag{3.32}$$

where ΔW denotes the change of W. From (3.22) and (3.24), we obtain

$$\Delta W \equiv -a_k e(k)\frac{\partial e(k)}{\partial W} \cong a_k e(k)g(k). \tag{3.33}$$

Thus,

$$
\begin{aligned}
\Delta V(k) &= \frac{1}{2}\left[e^2(k+1) - e^2(k)\right] \\
&= \frac{1}{2}[e(k+1) - e(k)] \cdot [e(k+1) + e(k)] \\
&= \frac{1}{2}\Delta e(k) \cdot [2e(k) + \Delta e(k)] \\
&= \Delta e(k) \cdot \left(e(k) + \frac{1}{2}\Delta e(k)\right) \\
&= \left[\frac{\partial e(k)}{\partial W}\right]^T a_k e(k) g(k) \cdot \left\{e(k) + \frac{1}{2}\left[\frac{\partial e(k)}{\partial W}\right]^T a_k e(k) g(k)\right\} \\
&= -a_k e^2(k)|g(k)|^2\left(1 - \frac{1}{2}a_k|g(k)|^2\right).
\end{aligned}
\tag{3.34}
$$

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

$$
\Delta V(k) = -e^2(k)P(k)\left(1 - \frac{1}{2}P(k)\right).
\tag{3.35}
$$

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

$$
0 < a_k < \frac{2}{|g(k)|^2}.
\tag{3.36}
$$

The asymptotic stability is guaranteed if a_k is chosen to satisfy (3.36). In addition, we would like to find a condition for a_k that guarantees the fast convergence. From (3.34) and (3.35), we have

$$
\begin{aligned}
e^2(k+1) &= e^2(k) - e^2(k)P(k)(2 - P(k)) \\
&= e^2(k)\left[1 - 2P(k) + P^2(k)\right] \\
&= e^2(k)[P(k) - 1]^2.
\end{aligned}
\tag{3.37}
$$

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

$$a_k^* = \frac{1}{|g(k)|^2}. \tag{3.38}$$

$\square$

For the system identification problem, a similar convergence theorem can be obtained.

Theorem 3.4. *Let a_{Ik} be the learning step length of tuning parameters for the IT2RFNS-A. Consider the nonlinear identification problem by series-parallel architecture using IT2RFNS-A (shown in Figure 4), and the parameters update laws are shown in (3.27). The asymptotic convergence of the IT2RFNS-A system is guaranteed if the chosen learning step length is satisfied:*

$$0 < a_{Ik} < \frac{2}{|g_I(k)|^2}, \quad \forall k, \tag{3.39}$$

where $g_I(k) = (O^{(6)}(W(k) + c_k\Delta_k) - O^{(6)}(W(k))/c_k)[\Delta_{k1}^{-1}\Delta_{k2}^{-1}\cdots\Delta_{kp}^{-1}]^T$ is the gradient estimation of IT2RFNS-A by SPSA approach. In addition, the faster convergence can be obtained by using the following optimal time-varying learning step length:

$$a_{Ik}^* = \frac{1}{|g_I(k)|^2}. \tag{3.40}$$

Proof. Herein, we omitted it due to that the proof of Theorem 3.4 is similar to the proof of Theorem 3.3. Only the express of the estimated gradient function $g(k)$ is replaced by $g_I(k)$.

$\square$

Remark 3.5. As above description, the SPSA algorithm has the stochastic property in the estimation of gradient functions $g(k)$ and $g_I(k)$ due to the random values c_k and Δ_k. Therefore, the stability conditions (3.28) and (3.39) are important for training. In addition, the time-varying optimal learning step lengths a_k^* and a_{Ik}^*, shown in (3.29) and (3.40), have the ability of guaranteeing the high-speed convergence. This reduces the effect of random values. Simulation results in Section 4 demonstrate the performance of our approach.

4. Simulation Results

Simulation results and comparisons, including the nonlinear system identification and the control of nonlinear system, are introduced to verify the performance of our proposed approach.

4.1. Example 1: Nonlinear System Identification of Chaotic System

Consider a nonlinear system described as

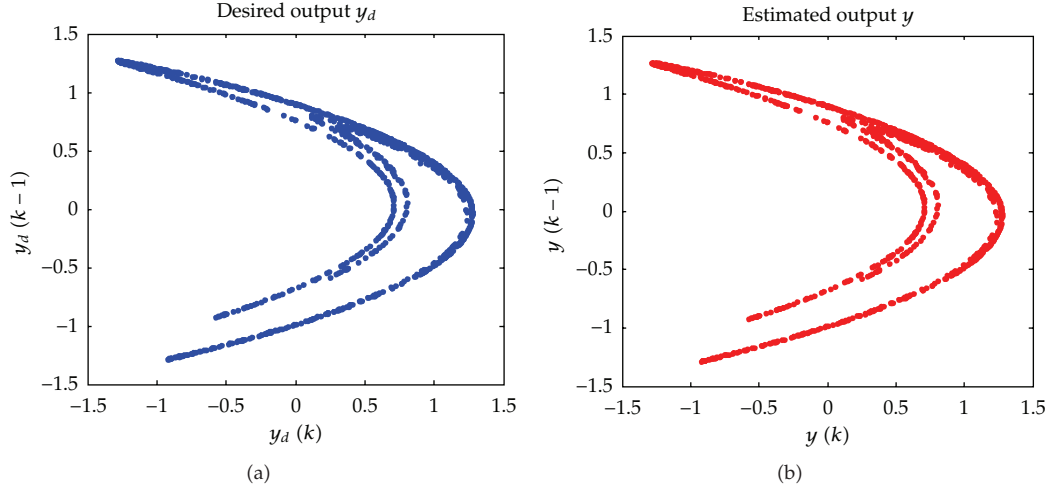$$y(k) = f(y(k-1), \ldots, y(k-n); u(k-1), \ldots, u(k-m)), \tag{4.1}$$

Figure 5: Phase plane plot of (a) the chaotic system and (b) the result of IT2RFNS-A with optimal learning step length.

where u denotes the input, y is the output, m and n are positive integers, and $f(\cdot)$ is the unknown function $f : \mathfrak{R}^{n+m} \rightarrow \mathfrak{R}$. Our purpose is to identify the nonlinear system (4.1) using IT2RFNS-A with the proposed stable SPSA algorithm. From literature [2], the chaotic system is described as

$$y(k+1) = -P \cdot y^2(k) + Q \cdot y(k-1) + 1.0, \tag{4.2}$$

where $P = 1.4$ and $Q = 0.3$ and produces a chaotic strange attractor as shown in Figure 5(a). We first choose 1000 pairs training data randomly from the system over the interval $[-1.5\ 1.5]$ and the initial point is set to $[y(1)\ y(0)] = [0.4\ 0.4]$. Then, the IT2RFNS-A is used to approximate the chaotic system using the stable SPSA algorithm with the following coefficients listed in Table 1. After training for 100 epochs, the phase plane of the chaotic system is shown in Figure 5(b). In order to show the superiority of using optimal learning step length, the same simulation is done without using optimal learning step length for comparison. In addition, we also compare it with other neural networks such as fuzzy neural network (FNN) [1], recurrent fuzzy neural network (RFNN) [2], and interval type-2 fuzzy logic system with asymmetric membership function (IT2FNS-A) [16]. For statistical analysis, the learning process is repeated for 10 independent runs. The descending trend of the mean square error (MSE) is shown in Figure 6, solid-black line: IT2RFNS-A with the optimal learning step length a_{Ik}^*; solid-red line: IT2RFNS-A without the optimal learning step length; dashed line: IT2FNS-A; dotted line: RFNN; dash-dotted line: FNN. The worst, average, and best MSE that each neural network achieved are shown in Table 2. Obviously, the IT2RFNS-A with optimal learning step length achieves the best performance and if we focus on using optimal learning step length or not, then we can find that there is a great improvement. From Figure 6, we can observe that using optimal learning step length results in better performance in convergence. Hence, we accumulate the number of epochs that MSE decreased in 100 epochs and listed in Table 3. The result, evidently, shows that the optimal learning step length can guarantee the efficient training of IT2RFNS-A.

Table 1: Configuration coefficients for Example 1: system identification.

		FNN	RFNN	IT2FNS-A	IT2RFNS-A	IT2RFNS-A
SPSA configuration coefficients	a	0.1	0.1	0.01	0.01	optimal learning step length (3.40)
	α	0.602	0.602	0.602	0.602	
	A	10	10	10	10	
	c	0.1	0.1	0.1	0.1	0.1
	γ	0.101	0.101	0.101	0.101	0.101
Neural network coefficient	rule number	8	8	2	2	2
epochs		100				

Table 2: Comparison results in MSE for Example 1.

MSE	FNN	RFNN	IT2FNS-A	IT2RFNS-A	IT2RFNS-A (optimal a_{Ik}^*)
Worst	0.0048	0.0017	6.9579×10^{-5}	1.9464×10^{-5}	5.0267×10^{-6}
Average	0.0028	6.6921×10^{-4}	3.4154×10^{-5}	4.8760×10^{-6}	1.3319×10^{-6}
Best	0.0018	2.2731×10^{-4}	1.1265×10^{-5}	6.0900×10^{-8}	1.0091×10^{-27}

4.2. Nonlinear System Control of Chua's Chaotic Circuit

Consider an nth-order nonlinear dynamic system in the companion form or controllability canonical from

$$x^{(n)} = f(x) + g(x)u + d,$$

$$y = x, \tag{4.3}$$

where u and y are the control input and output of the nonlinear system, $f(\cdot)$ and $g(\cdot)$ are unknown nonlinear and continuous functions, and d is the external disturbance or system uncertainty. 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.

In this paper, we consider the typical Chua's chaotic circuit which consists of one inductor, two capacitors, one linear resistor, and one piecewise-linear resistor [29, 30]. 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

$$\frac{dv_{C_1}}{dt} = (C_1)^{-1} \left(\frac{v_{C_2} - v_{C_1}}{R} - \gamma \right),$$

$$\frac{dv_{C_2}}{dt} = (C_2)^{-1} \left(\frac{v_{C_1} - v_{C_2}}{R} - i_L \right), \tag{4.4}$$

$$\frac{di_L}{dt} = (L)^{-1} (-v_{C_1} - R_O i_L),$$

Table 3: Comparison results of using optimal learning step length (number of epochs that MSE decreases in 100 epochs, 10 independent runs).

Number of epochs	IT2RFNS-A	IT2RFNS-A with optimal a^*_{lk}
Worst	38	68
Average	36	89
Best	55	98
Total epochs		100
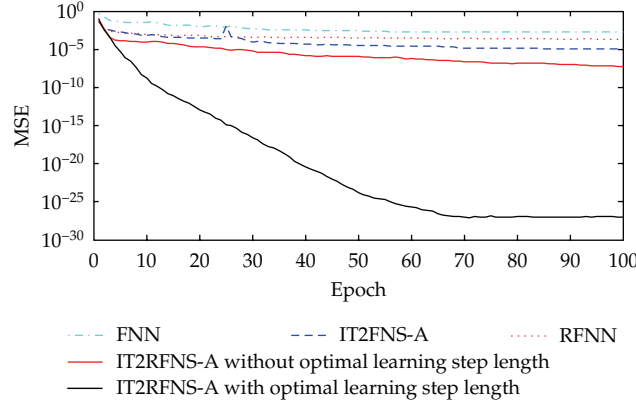
Figure 6: Comparison results in MSE of Example 1.

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

$$\gamma = aV_{C_1} + c(v_{C_1})^3 \quad (a > 0, \ c < 0). \tag{4.5}$$

According to [30], (4.4) can be transferred into the canonical form as

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

$$
y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \tag{4.6}
$$

where $f = (14/1805)x_1 - (168/9025)x_2 + (1/38)x_3 - (2/45)((28/361)x_1 + (7/95)x_2 + x_3)^3$, $g = 1$, 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 4. The

Table 4: The selected coefficients for Example 2.

		FNN	RFNN	IT2FNS-A	IT2RFNS-A	IT2RFNS-A
SPSA configuration coefficients	a	0.01	0.01	0.01	0.01	optimal learning step length (3.29)
	α	0.602	0.602	0.602	0.602	
	A	200	200	200	200	
	c	1	1	1	1	1
	γ	0.101	0.101	0.101	0.101	0.101
Neural network coefficient	rule number	12	12	4	4	4
Time (sec.)				20		

Table 5: Comparison results of control performance for Example 2.

| MSE	FNN	RFNN	IT2FNS-A	IT2RFNS-A	IT2RFNS-A (optimal a_k)
Worst	2.6046	2.5321	0.3906	0.7264	0.0682
Average	1.2580	1.2005	0.3146	0.2768	0.0476
Best	0.7599	0.6532	0.2605	0.0713	0.0307

simulation results of trajectories are shown in Figure 7. Figure 7(a) shows the phase plant trajectory after being controlled; Figures 7(b)–7(d) shows the state trajectories x_1, x_2, and x_3, respectively (solid line: actual system output; dashed line: reference trajectory). Figure 8 depicts the control effort of the IT2RFNS-A controller. We can observe that the proposed method is valid to control a nonlinear dynamic system. The comparison results are shown in Table 5 which is obtained by 10 independent runs. From Table 5, we can easily find that the IT2RFNS-A performs better than others. 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 5, we again demonstrate that adopting optimal learning step length guarantees the efficient training of IT2RFNS-A.

5. Conclusion

In this paper, we have proposed an interval type-2 recurrent fuzzy neural system with triangular asymmetric membership functions (IT2RFNS-A) and its training scheme using the proposed stable SPSA algorithm. 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 and guarantee the convergence of IT2RFNS-A for system identification. In addition, we also obtain the optimal learning step length that ensures the efficient training for IT2RFNS-A. The feasibility and the effectiveness of the proposed method have been demonstrated by two illustration examples. The simulations on the chaotic system identification and the control of the Chua's chaotic circuit are done, and both of them show the following advantages of proposed approach, (a) the IT2RFNS-A having few rules can achieve the specific performance or even better; (b) the IT2RFNS-A can catch the dynamic response of the system; that is, it is capable of coping with dynamic problems; (c) by using the proposed stable 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
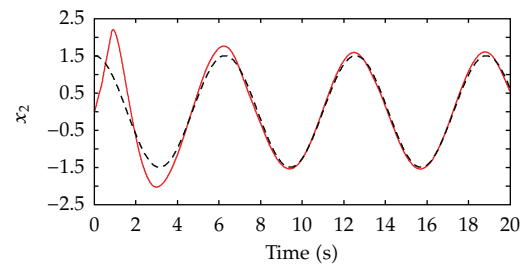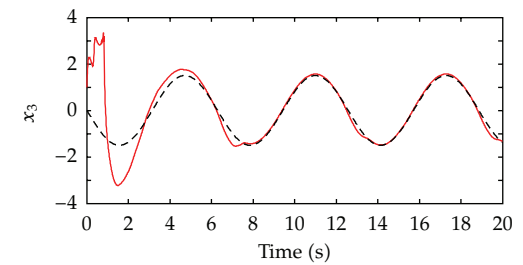
(a)

(b)

(c)
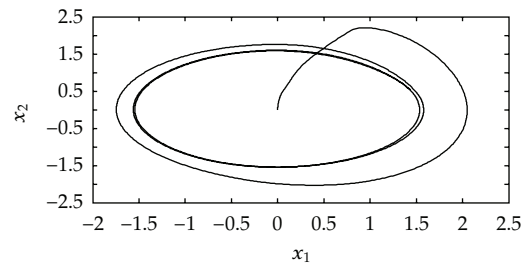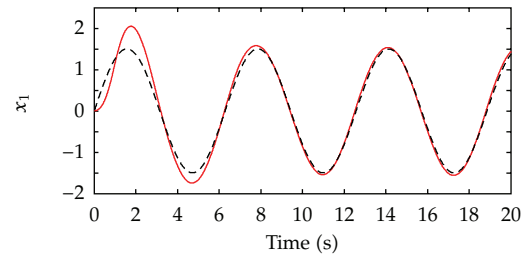
(d)

Figure 7: Simulation results of Example 2: (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 $\ddot{y}_r$.
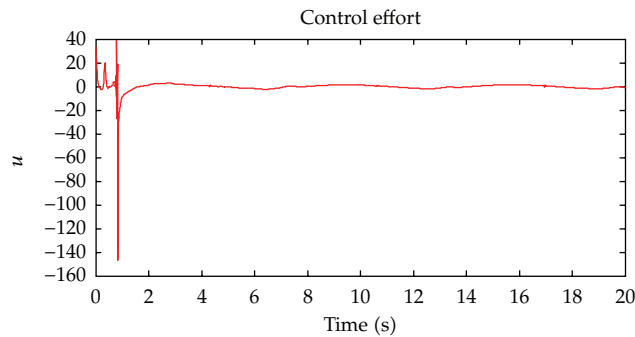
Figure 8: Control effort of IT2RFNS-A controller of Chua's chaotic circuit.

avoids approximating the system sensitivity which leads to inaccuracy; (e) adopting optimal learning step length improves the performance of training IT2RFNS-A.

Acknowledgments

References

[1] Y. C. Chen and C. C. Teng, "A model reference control structure using a fuzzy neural network," *Fuzzy Sets and Systems*, vol. 73, no. 3, pp. 291–312, 1995.

[2] C. H. Lee and C. C. Teng, "Identification and control of dynamic systems using recurrent fuzzy neural networks," *IEEE Transactions 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 Transactions on Fuzzy Systems*, vol. 11, no. 4, pp. 462–477, 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 Transactions on Fuzzy Systems*, vol. 12, no. 4, pp. 552–559, 2004.

[5] A. Lotfi and A. C. Tsoi, "Learning fuzzy inference systems using an adaptive membership function scheme," *IEEE Transactions 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, no. 3, pp. 199–249, 1975.

[10] Q. Liang and J. M. Mendel, "Interval type-2 fuzzy logic systems: theory and design," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 5, pp. 535–550, 2000.

[11] J. M. Mendel, "On the importance of interval sets in type-2 fuzzy logic systems," in *Proceedings of the Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, pp. 1647–1652, Vancouver, BC, Canada, July 2001.

[12] 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.

[13] C. H. Lee, Y. C. Lin, and W. Y. Lai, "Systems identification using type-2 fuzzy neural network (type-2 FNN) systems," in *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA '03)*, vol. 3, pp. 1264–1269, 2003.

[14] C. H. Lee and Y. C. Lin, "Control of nonlinear uncertain systems using type-2 fuzzy neural network and adaptive filter," in *Proceedings of the IEEE International Conference on Networking, Sensing and Control*, vol. 2, pp. 1177–1182, March 2004.

[15] C. H. Lee and Y. C. Lin, "An adaptive type-2 fuzzy neural controller for nonlinear uncertain systems," *Control and Intelligent Systems*, vol. 12, no. 1, pp. 41–50, 2005.

[16] 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," in *Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS '10)*, Lecture Notes in Engineering and Computer Science, pp. 140–145, Hong Kong, March 2010.

[17] G. C. Mouzouris and J. M. Mendel, "Nonsingleton fuzzy logic systems: theory and application," *IEEE Transactions on Fuzzy Systems*, vol. 5, no. 1, pp. 56–71, 1997.

[18] J. Zhang and A. J. Morris, "Recurrent neuro-fuzzy networks for nonlinear process modeling," *IEEE Transactions on Neural Networks*, vol. 10, no. 2, pp. 313–326, 1999.

[19] C. F. Juang, "A TSK-type recurrent fuzzy neural network for dynamic systems processing by neural network and genetic algorithm," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 155–170, 2002.

[20] J. B. Theocharis, "A high-order recurrent neuro-fuzzy system with internal dynamics: application to the adaptive noise cancellation," *Fuzzy Sets and Systems*, vol. 157, no. 4, pp. 471–500, 2006.

[21] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Transactions on Automatic Control*, vol. 37, no. 3, pp. 332–341, 1992.

[22] J. C. Spall and J. A. Cristion, "Model free control of nonlinear stochastic systems with discrete-time measurements," *IEEE Transactions on Automatic Control*, vol. 43, no. 9, pp. 1198–1210, 1998.

[23] J. C. Spall, *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*, John Wiley & Sons, Hoboken, NJ, USA, 2003.

[24] N. N. Karnik and J. M. Mendel, "Centroid of a type-2 fuzzy set," *Information Sciences*, vol. 132, no. 1–4, pp. 195–220, 2001.

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

[26] J. R. Castro, O. Castillo, P. Melin, and R. Díaz, "A hybrid learning algorithm for a class of interval type-2 fuzzy neural networks," *Information Sciences*, vol. 179, no. 13, pp. 2175–2193, 2009.

[27] J. J. E. Slotine and W. Li, *Applied Nonlinear Control*, Prentice-Hall, Upper Saddle River, NJ, USA, 1991.

[28] T. Yabuta and T. Yamada, "Learning control using neural networks," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '91)*, pp. 740–745, Sacramento, Calif, USA, April 1991.

[29] Y. H. Joo, L. S. Shieh, and G. Chen, "Hybrid state space fuzzy model based controller with dual-rate sampling for digital control of chaotic systems," *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 4, pp. 394–408, 1999.

[30] C. H. Wang, T. C. Lin, T. T. Lee, and H. L. Liu, "Adaptive hybrid intelligent control for uncertain nonlinear dynamical systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 32, no. 5, pp. 583–597, 2002.