Robust Self-Organizing Fuzzy-Neural Control Using Asymmetric Gaussian Membership Functions

Ping-Zong Lin and Tsu-Tian Lee

Abstract

A robust self-organizing fuzzy-neural control (RSOFNC) system is proposed in this paper. The RSOFNC system is comprised of a self-structuring fuzzy neural network (SFNN) controller and a robust controller. The SFNN controller is the principal controller and the robust controller is designed to achieve L_2 tracking performance. In the SFNN controller design, a SFNN with the asymmetric Gaussian membership functions is used to online approximate an ideal controller via the structure and parameter learning phases. The structure learning phase consists of the growing of membership functions and the pruning of fuzzy rules, and thus the SFNN can avoid the time-consuming trial-and-error tuning procedure for determining the network structure of fuzzy neural network. Finally, the proposed RSOFNC system is applied to control a second-order chaotic system. The simulation results show that the proposed RSOFNC system can achieve favorable tracking performance.

Keywords: Fuzzy neural network, asymmetric Gaussian membership function, structure adaptation algorithm, adaptive control, robust control.

1. Introduction

With the learning ability of neural networks, they have widely been recognized a powerful tool in industrial control, commercial, image processing applications and etc. [1], [2]. Within control engineering, neural networks are attractive because they hold the promise of solving problems that have so far been difficult to handle with classical analytical methods. Recently, the fuzzy neural network (FNN) which incorporates the advantages of fuzzy inference and neuro-learning has been an interesting topic. The FNN possesses the merits of the low-level learning and computational power of neural network, and the high-level human knowledge representation and thinking of fuzzy theory [3], [4]. The most useful property of the FNNs is the ability to arbitrarily approximate linear or nonlinear mappings through learning. The FNNs are increasingly receiving attention in solving the control problems [5]-[8]. For the FNN-based control approaches in [5]-[8], the structure of the FNN should be determined by trial-and-error in advance because it is difficult to consider the balance between the rule number and the desired performance. If the number of fuzzy rules is chosen too large, the computation loading is heavy so that FNNs are not suitable for practical applications. If the number of fuzzy rules is chosen too small, the control performance may be not good enough to achieve desired performance.

To solve the problem of the structure determination in FNN approaches, much interest has been focused on the self-structuring fuzzy neural network (SFNN) approach [9]-[13]. The self-structuring approach demonstrates the properties of generating the rules of FNN automatically without needing the preliminary knowledge. In general, the mathematical description of the existing rules can be expressed as a cluster. As usually seen in other self-structuring approaches, the new membership function is generated when a new input signal is too far away from the current clusters, and an existing rule is canceled when the fuzzy rule is insignificant. The SFNNs also have been adopted widely for the control of complex dynamical systems owing to its good generalization capability, structure adaptation, and simple computation [14]-[16]. Some of them use the gradient descent method to derive the parameter learning algorithms; however, they can not guarantee the system stability. Some of them use the Lyapunov function to derive the parameter learning algorithms; however, the design procedure is extremely complex.

Recently, some researches are considered the asymmetric Gaussian membership function, also called Pseudo-Gaussian (PG), to act as the membership term node [17]-[19]. Because the variability and malleability of the asymmetric Gaussian membership function are higher than those of the traditional membership function,

Corresponding Author: Ping-Zong Lin is with the Department of Electrical and Control Engineering, National Chiao-Tung University, Hsinchu 300, Taiwan, Republic of China.
E-mail: u9212820@cn.nctu.edu.tw
Tsu-Tian Lee is with the Department of Electrical Engineering, National Taipei University of Technology, Taipei 106, Taiwan, Republic of China.
E-mail: ttlee@ntut.edu.tw
Manuscript received 25 March 2007; accepted 11 June. 2007.

the asymmetric Gaussian membership function provides a fuzzy-neural network with higher flexibility to approach easily the optimum result more accurately. The asymmetric Gaussian function is a new type of membership function, which is applied here due to the excellent approximation results it provides.

In this paper, a novel SFNN, which includes the growing of membership function and the pruning of fuzzy rules, is developed. The idea for these algorithms is given as follows. The growing method can be considered when the input values will cross the range of existing membership functions. After several executing iterations, the error between the real function and the estimation function gradually converges to a tolerant bound. Some fuzzy rules, which are generated in the transient period, may be less or never used in this situation. They make no meaningful contributions to the output of the SFNN and thus are obviously not significant. The pruning algorithm would eliminate these rules which make a very small contribution to the fuzzy neural network output. The asymmetric Gaussian function is considered as membership functions of the proposed SFNN to increase the learning capability. Then, a robust self-organizing fuzzy-neural control (RSOFNC) system, which is comprised of a SFNN controller and a robust controller, is proposed. The SFNN controller with structure adaptation and asymmetric membership functions is the principal controller, and the robust controller is designed to achieve L_2 tracking performance with desired attenuation level. The proposed structure adaptation will solve the annoyance which is difficult to consider the balance between the rule number and the desired performance. Finally, to investigate the effectiveness of the proposed control scheme, the proposed RSOFNC system is applied to control a second-order chaotic system. Simulation results demonstrate that the proposed RSOFNC scheme can achieve favorable tracking performances.

2. Problem Formulation and Ideal Controller

Consider the nth-order nonlinear dynamic system of the form

$$x^{(n)} = f(\mathbf{x}) + u \tag{1}$$

where $\mathbf{x} = [x\ \dot{x} \cdots x^{(n-1)}]^T$, which is assumed to be available for measurement, is the state vector of the system, $f(\mathbf{x})$ is the system dynamics equation, and u is the control effort. The control objective is to find a control law so that the state trajectory x can track a trajectory command x_c. A tracking error is defined as

$$e = x_c - x\,. \tag{2}$$

If the system dynamics $f(\mathbf{x})$ is well known, there exists an ideal controller as follows [20]

$$u_{id} = -f(\mathbf{x}) + x_c^{(n)} + \mathbf{k}^T\mathbf{e} \tag{3}$$

where $\mathbf{e} = [e\ \dot{e}\ \cdots\ e^{(n-1)}]^T$ is the tracking error vector and $\mathbf{k} = [k_1\ k_2\ \cdots\ k_n]^T$. Substituting (3) into (1) and using (2) yield

$$e^{(n)} + k_n e^{(n-1)} + ... + k_2\dot{e} + k_1 e = 0\,. \tag{4}$$

If k_i, $i = 1, 2, \cdots, n$ are chosen to correspond to the coefficients of a Hurwitz polynomial whose roots lie strictly in the open left half of complex plane, then $\lim_{t\to\infty} e = 0$ can be implied for any starting initial conditions. However, because the system dynamics $f(\mathbf{x})$ may be unknown or perturbed in practice, the ideal control law u_{id} in (3) cannot be implemented easily.

3. Self-Structuring Fuzzy Neural Network

A. *Structure of SFNN*

The structure of the SFNN has four layers of neural network included the input, the membership function, the rule, and the output layers. Nodes at layer 1 are input nodes that represent input linguistic variables. Nodes at layer 2 are term nodes which act as membership functions to represent the terms of the respective linguistic variables. The asymmetric Gaussian membership function which consists of a center, a left-side variance, and a right-side variance is considered. Each node at layer 3 is a fuzzy rule. Layer 4 is the output layer, where the node in this layer is the output of SFNN. The interactions for those layers are given as follows. Note that the superscript q in x_i^q denotes the layer number.

Layer 1 - Input layer: For every node i in this layer, the net input and the net output are represented as

$$net_i^1 = x_i^1 \tag{5}$$

$$y_i^1 = f_i^1\left(net_i^1\right) = net_i^1\,,\quad i = 1, 2, \cdots, L \tag{6}$$

where x_i^1 represents the i-th input to the node of layer 1.

Layer 2 - Membership layer: In this layer, each node performs a membership function and acts as a unit of memory. The asymmetric Gaussian function is adopted as the membership function. For the i-th input and the j-th node, the corresponding input and output are described as

$$net_{ij}^2 = \begin{cases} -\left(x_i^2 - m_{ij}\right)^2 \big/ \left(\sigma_{ij}^l\right)^2, & if\ \ -\infty < x_i^2 \le m_{ij} \\ -\left(x_i^2 - m_{ij}\right)^2 \big/ \left(\sigma_{ij}^r\right)^2, & if\ \ m_{ij} \le x_i^2 < \infty \end{cases} \tag{7}$$

$$y_{ij}^2 = f_{ij}^2\left(net_{ij}^2\right) = \mathbf{exp}\left(net_{ij}^2\right),\quad j = 1, 2, \cdots, M \tag{8}$$

where M is the total number of membership functions with respect to the respective input node, and m_{ij}, σ_{ij}^{l}, and σ_{ij}^{r} are the mean, left-side variance, and right-side variance of the asymmetric Gaussian function in the j-th term of the i-th input linguistic variable x_i^2.

Layer 3 - Rule layer: The links in this layer are used to implement the antecedent matching. Each node k in this layer is denoted by $\prod$ which multiplies the incoming signals and outputs the result of the product. For the k-th rule node, the operation is presented as

$$net_k^3 = \prod w_{ij}^3 x_{ij}^3 \tag{9}$$

$$y_k^3 = f_k^3\left(net_k^3\right) = net_k^3,\quad k = 1, 2, \cdots, N \tag{10}$$

where x_{ij}^3 represents the i, j-th input to the k-th node of layer 3, the weights w_{ij}^3 between the membership and the rule layers are assumed as unity.

Layer 4 - Output layer: The single node o in this layer is labeled as Σ which computes the overall output as the summation of all incoming signals. The operation is described as

$$net_o^4 = \sum_k w_k^4 x_k^4 \tag{11}$$

$$y_o^4 = f_o^4\left(net_o^4\right) = net_o^4,\quad o = 1 \tag{12}$$

where w_k^4 is the output action strength of the output associated with the k-th rule, x_k^4 represents the k-th input to the node of layer 4, and y_o^4 is the output of SFNN. The vectors $\mathbf{m}$, $\boldsymbol{\sigma}_l$, and $\boldsymbol{\sigma}_r$ collecting all parameters of membership layer are defined as

$$\mathbf{m} = [m_{11} \cdots m_{L1}\ m_{12} \cdots m_{L2} \cdots\cdots m_{1M} \cdots m_{LM}]^T \tag{13}$$

$$\boldsymbol{\sigma}_l = [\sigma_{11}^l \cdots \sigma_{L1}^l\ \sigma_{12}^l \cdots \sigma_{L2}^l \cdots\cdots \sigma_{1M}^l \cdots \sigma_{LM}^l]^T \tag{14}$$

$$\boldsymbol{\sigma}_r = [\sigma_{11}^r \cdots \sigma_{L1}^r\ \sigma_{12}^r \cdots \sigma_{L2}^r \cdots\cdots \sigma_{1M}^r \cdots \sigma_{LM}^r]^T. \tag{15}$$

Hence, the output of the SFNN can be represented in a vector form

$$y_o^4 = \mathbf{w}^T \boldsymbol{\xi}(\mathbf{x}, \mathbf{m}, \boldsymbol{\sigma}_l, \boldsymbol{\sigma}_r) \tag{16}$$

where $\boldsymbol{\xi} = [x_1^4\ x_2^4\ \cdots\ x_N^4]^T = [\xi_1\ \xi_2\ \cdots\ \xi_N]^T$ and $\mathbf{w} = [w_1^4\ w_2^4\ \cdots\ w_N^4]^T$.

B. *Structure Adaptation of SFNN*

For the FNN approaches in [5]-[8], the structure of the FNN was determined by trial-and-error in advance for the reason that it is difficult to consider the balance between the rule number and the desired performance. In this paper, the structure learning algorithm which includes the growing of membership functions and the pruning of fuzzy rules is proposed to solve this problem. The descriptions for those algorithms are given as follows.

In the process of the growing of membership functions, the concept which decides whether to add a new node (membership function) in layer 2 and the associated fuzzy rule in layer 3 will be introduced. For constructing the initial fuzzy rules of the SFNN, the fuzzy clustering method is used to partition a set of data into a number of overlapping clusters based on the distance in a metric space between the data points and the cluster prototypes. Each cluster in the product space of the input-output data represents a rule in the rule base. The firing strength of a rule for each incoming data x_i^1 can be represented as the degree that the incoming data belongs to the cluster. If the value of firing strength is too small, it represents that the input value is on the edge of range of the existing membership functions. In this situation, the output may cause a bad control performance. Therefore, a new node (membership function) should be added at this moment.

The firing strength obtained from (9) is used as the degree measure

$$\beta_k = y_k^3,\quad k = 1, 2, ..., N \tag{17}$$

where N is the number of the existing rules. Find the maximum degree $\beta_{\max}$ defined as

$$\beta_{\max} = \max_{1 \le k \le N} \beta_k. \tag{18}$$

We observe that if $\beta_{\max} \le G_{th}$ is satisfied, where $G_{th} \in (0,1)$ is a pre-given threshold, the incoming data is far away from the edge of range of the existing membership functions. Hence, a new membership function is generated. The mean and the standard deviation of the new membership function and the weight are selected as follows

$$m_i^{new} = x_i^1, \tag{19}$$

$$\sigma_i^{l,new} = \sigma_i, \tag{20}$$

$$\sigma_i^{r,new} = \sigma_i, \tag{21}$$

$$w^{new} = 0 \tag{22}$$

where x_i is the new incoming data and σ_i is a pre-specified constant. If the unknown control system dynamics is too complex, we can choose the larger G_{th} so that many membership functions can be created.

Next, to avoid the unbounded growing of network structure and the overload computation, the algorithm for the pruning of fuzzy rules is developed to eliminate irrelevant fuzzy rules. When the r-th firing strength β_r is smaller than the pre-defined threshold value P_{th}, it means that the relationship becomes weak between the input and the r-th rule. This fuzzy rule may be less or never used. Then, we will gradually reduce the value of the r-th used frequency index when the r-th firing strength β_r satisfies our setting condition continuously.

However, we do not ignore this situation that the r-th firing strength β_r is bigger than the threshold value P_{th} again. Therefore, when the incoming inputs fall into the range of the r-th fuzzy rule, we have to make the value of the r-th used frequency index rise. The rise and decay curves of the used frequency index show in Fig. 1, which is determined as

$$I_r = \begin{cases} I_r^p \cdot \exp(-\tau_1), & \text{if } \beta_r < P_{th} \\ I_r^p \cdot \left[2 - \exp\left(-\tau_2\left(1 - I_r^p\right)\right)\right], & \text{if } \beta_r \geq P_{th} \end{cases}, \quad r = 1, 2, \cdots, N \tag{23}$$

where I_r is the used frequency index of the r-th rule and its initial value is 1, P_{th} is the pruning threshold value, τ_1 and τ_2 are the designed constant, and I_r^p denotes the most recent I_r. Note that the maximum value of the used frequency index equals 1. If $I_r \leq I_{th}$ is satisfied, where I_{th} is another pre-given threshold, the r-th fuzzy rule is pruned. If the computation loading is the important issue for practical implement, we can choose the larger P_{th} so that many fuzzy rules can be pruned. Hence, the computation load should be decreased. In summary, the flow chart of the structure learning algorithm is shown in Fig. 2. The major contributions of using these algorithms for the SFNN structure adaptation are: 1) SFNN can be operated directly without spending much time on pre-determining membership functions and fuzzy rules; and 2) the computation load can be reduced simultaneously.

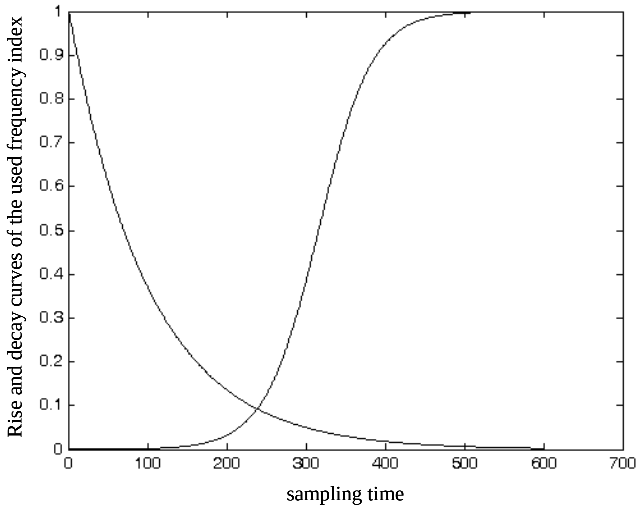

Figure 1. The rise and decay curves of the used frequency index.

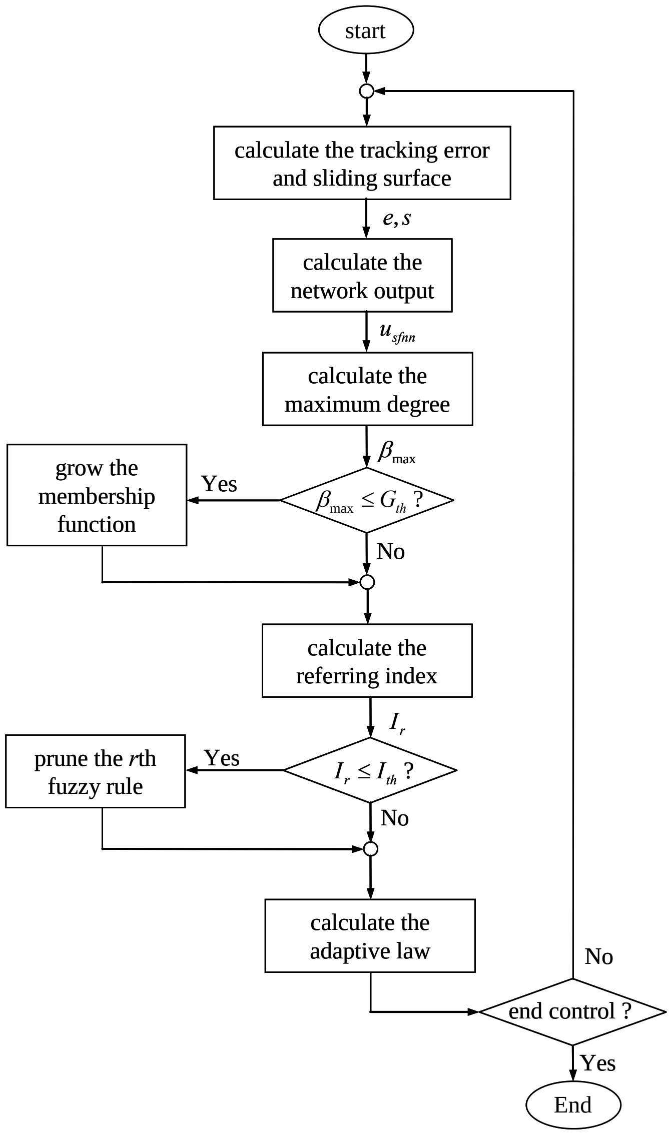

Figure 2. The flow chart of the structure adaptation algorithm.

4. Robust Self-Organizing Fuzzy-Neural Control Design

The RSOFNC system is proposed to tackle the control problems which we discussed before. The control law of the RSOFNC is developed as follows

$$u_{rsofnc} = u_{sfnn} + u_{rb} \tag{24}$$

where the SFNN controller u_{sfnn} using asymmetric Gaussian membership functions is designed to mimic the ideal controller, and the robust controller u_{rb} is designed to compensate for the modeling error between the SFNN controller and the ideal controller. Define a sliding surface is defined as

$$s = e^{(n-1)} + k_n e^{(n-2)} + \cdots + k_2 e + k_1 \int_0^t e \, d\tau \, . \tag{25}$$

Substituting (24) into (1) and using (2), (3), and (25) yield

$$\dot{s} = u_{id} - u_{sfnn} - u_{rb} \, . \tag{26}$$

A. *Approximation of SFNN*

By the universal approximation theorem, an optimal

SFNN controller can be designed to approximate the ideal controller, such that [21]

$$u_{id} = \mathbf{w}^{*T}\boldsymbol{\xi}(\mathbf{x},\mathbf{m}^*,\boldsymbol{\sigma}_l^*,\boldsymbol{\sigma}_r^*) + \Delta = \mathbf{w}^{*T}\boldsymbol{\xi}^* + \Delta \quad (27)$$

where $\mathbf{w}^{*T}\boldsymbol{\xi}(\mathbf{x},\mathbf{m}^*,\boldsymbol{\sigma}_l^*,\boldsymbol{\sigma}_r^*) = \mathbf{w}^{*T}\boldsymbol{\xi}^*$, $\mathbf{m}^*$, $\boldsymbol{\sigma}_l^*$, $\boldsymbol{\sigma}_r^*$ are the optimal vectors of $\mathbf{m}$, $\boldsymbol{\sigma}_l$, $\boldsymbol{\sigma}_r$, and Δ is the approximation error. However, the optimal SFNN controller is unknown. Therefore, a SFNN controller will be introduced to mimic the ideal controller as

$$u_{sfnn} = \hat{\mathbf{w}}^T\boldsymbol{\xi}(\mathbf{x},\hat{\mathbf{m}},\hat{\boldsymbol{\sigma}}_l,\hat{\boldsymbol{\sigma}}_r) = \hat{\mathbf{w}}^T\hat{\boldsymbol{\xi}} \quad (28)$$

where $\hat{\mathbf{w}}^T\boldsymbol{\xi}(\mathbf{x},\hat{\mathbf{m}},\hat{\boldsymbol{\sigma}}_l,\hat{\boldsymbol{\sigma}}_r) = \hat{\mathbf{w}}^T\hat{\boldsymbol{\xi}}$ and $\hat{\mathbf{m}}$, $\hat{\boldsymbol{\sigma}}_l$, $\hat{\boldsymbol{\sigma}}_r$ are the estimated vectors of $\mathbf{m}$, $\boldsymbol{\sigma}_l$, $\boldsymbol{\sigma}_r$. Define an modeling error, $\tilde{u}$, as

$$\begin{aligned}\tilde{u} &= u_{id} - u_{sfnn} \\ &= \tilde{\mathbf{w}}^T\tilde{\boldsymbol{\xi}} + \hat{\mathbf{w}}^T\tilde{\boldsymbol{\xi}} + \tilde{\mathbf{w}}^T\hat{\boldsymbol{\xi}} + \Delta \end{aligned} \quad (29)$$

where $\tilde{\mathbf{w}} = \mathbf{w}^* - \hat{\mathbf{w}}$ and $\tilde{\boldsymbol{\xi}} = \boldsymbol{\xi}^* - \hat{\boldsymbol{\xi}}$. In the following, the linearization technique is employed to transform the nonlinear fuzzy function into a partially linear form so that the expansion $\tilde{\boldsymbol{\xi}}$ can be expressed as [22]

$$\begin{aligned}\tilde{\boldsymbol{\xi}} = \begin{bmatrix}\tilde{\xi}_1\\ \tilde{\xi}_2\\ \vdots\\ \tilde{\xi}_N\end{bmatrix} &= \begin{bmatrix}\partial\xi_1/\partial\mathbf{m}\\ \partial\xi_2/\partial\mathbf{m}\\ \vdots\\ \partial\xi_N/\partial\mathbf{m}\end{bmatrix}\Big|_{\mathbf{m}=\hat{\mathbf{m}}}(\mathbf{m}^* - \hat{\mathbf{m}}) \\ &+ \begin{bmatrix}\partial\xi_1/\partial\boldsymbol{\sigma}_l\\ \partial\xi_2/\partial\boldsymbol{\sigma}_l\\ \vdots\\ \partial\xi_N/\partial\boldsymbol{\sigma}_l\end{bmatrix}\Big|_{\boldsymbol{\sigma}_l=\hat{\boldsymbol{\sigma}}_l}(\boldsymbol{\sigma}_l^* - \hat{\boldsymbol{\sigma}}_l) \\ &+ \begin{bmatrix}\partial\xi_1/\partial\boldsymbol{\sigma}_r\\ \partial\xi_2/\partial\boldsymbol{\sigma}_r\\ \vdots\\ \partial\xi_N/\partial\boldsymbol{\sigma}_r\end{bmatrix}\Big|_{\boldsymbol{\sigma}_l=\hat{\boldsymbol{\sigma}}_l}(\boldsymbol{\sigma}_r^* - \hat{\boldsymbol{\sigma}}_r) + \mathbf{h} \\ &= \boldsymbol{\xi}_m^T\tilde{\mathbf{m}} + \boldsymbol{\xi}_{\sigma_l}^T\tilde{\boldsymbol{\sigma}}_l + \boldsymbol{\xi}_{\sigma_r}^T\tilde{\boldsymbol{\sigma}}_r + \mathbf{h}\end{aligned} \quad (30)$$

where $\mathbf{h}$ is a vector of higher-order terms, $\tilde{\mathbf{m}} = \mathbf{m}^* - \hat{\mathbf{m}}$, $\tilde{\boldsymbol{\sigma}}_l = \boldsymbol{\sigma}_l^* - \hat{\boldsymbol{\sigma}}_l$, $\tilde{\boldsymbol{\sigma}}_r = \boldsymbol{\sigma}_r^* - \hat{\boldsymbol{\sigma}}_r$,

$$\boldsymbol{\xi}_m = [\partial\xi_1/\partial\mathbf{m} \quad \partial\xi_2/\partial\mathbf{m} \quad \cdots \quad \partial\xi_N/\partial\mathbf{m}]|_{\mathbf{m}=\hat{\mathbf{m}}}, \quad (31)$$

$$\boldsymbol{\xi}_{\sigma_l} = [\partial\xi_1/\partial\boldsymbol{\sigma}_l \quad \partial\xi_2/\partial\boldsymbol{\sigma}_l \quad \cdots \quad \partial\xi_N/\partial\boldsymbol{\sigma}_l]|_{\boldsymbol{\sigma}_l=\hat{\boldsymbol{\sigma}}_l}, \quad (32)$$

and

$$\boldsymbol{\xi}_{\sigma_r} = [\partial\xi_1/\partial\boldsymbol{\sigma}_r \quad \partial\xi_2/\partial\boldsymbol{\sigma}_r \quad \cdots \quad \partial\xi_N/\partial\boldsymbol{\sigma}_r]|_{\boldsymbol{\sigma}_r=\hat{\boldsymbol{\sigma}}_r}. \quad (33)$$

The $\partial\xi_i/\partial\mathbf{m}$, $\partial\xi_i/\partial\boldsymbol{\sigma}_l$ and $\partial\xi_i/\partial\boldsymbol{\sigma}_r$ are defined as

$$\partial\xi_i/\partial\mathbf{m} = [\partial\xi_i/\partial m_{i1} \quad \partial\xi_i/\partial m_{i2} \quad \cdots \quad \partial\xi_i/\partial m_{iM}]^T, \quad (34)$$

$$\partial\xi_i/\partial\boldsymbol{\sigma}_l = [\partial\xi_i/\partial\sigma_{i1}^l \quad \partial\xi_i/\partial\sigma_{i2}^l \quad \cdots \quad \partial\xi_i/\partial\sigma_{iM}^l]^T, \quad (35)$$

and

$$\partial\xi_i/\partial\boldsymbol{\sigma}_r = [\partial\xi_i/\partial\sigma_{i1}^r \quad \partial\xi_i/\partial\sigma_{i2}^r \quad \cdots \quad \partial\xi_i/\partial\sigma_{iM}^r]^T. \quad (36)$$

Substituting (30) into (29), we can rewrite $\tilde{u}$ as

$$\begin{aligned}\tilde{u} &= \tilde{\mathbf{w}}^T\tilde{\boldsymbol{\xi}} + \hat{\mathbf{w}}^T(\boldsymbol{\xi}_m^T\tilde{\mathbf{m}} + \boldsymbol{\xi}_{\sigma_l}^T\tilde{\boldsymbol{\sigma}}_l + \boldsymbol{\xi}_{\sigma_r}^T\tilde{\boldsymbol{\sigma}}_r + \mathbf{h}) + \tilde{\mathbf{w}}^T\hat{\boldsymbol{\xi}} + \Delta \\ &= \tilde{\mathbf{w}}^T\tilde{\boldsymbol{\xi}} + \tilde{\mathbf{m}}^T\boldsymbol{\xi}_m\hat{\mathbf{w}} + \tilde{\boldsymbol{\sigma}}_l^T\boldsymbol{\xi}_{\sigma_l}\hat{\mathbf{w}} + \tilde{\boldsymbol{\sigma}}_r^T\boldsymbol{\xi}_{\sigma_r}\hat{\mathbf{w}} + \varepsilon\end{aligned} \quad (37)$$

where $\hat{\mathbf{w}}^T\boldsymbol{\xi}_m^T\tilde{\mathbf{m}} = \tilde{\mathbf{m}}^T\boldsymbol{\xi}_m\hat{\mathbf{w}}$, $\hat{\mathbf{w}}^T\boldsymbol{\xi}_{\sigma_l}^T\tilde{\boldsymbol{\sigma}}_l = \tilde{\boldsymbol{\sigma}}_l^T\boldsymbol{\xi}_{\sigma_l}\hat{\mathbf{w}}$, $\hat{\mathbf{w}}^T\boldsymbol{\xi}_{\sigma_r}^T\tilde{\boldsymbol{\sigma}}_r = \tilde{\boldsymbol{\sigma}}_r^T\boldsymbol{\xi}_{\sigma_r}\hat{\mathbf{w}}$, and the uncertain term $\varepsilon = \hat{\mathbf{w}}^T\mathbf{h} + \tilde{\mathbf{w}}^T\tilde{\boldsymbol{\xi}} + \Delta$.

B. *Parameter Learning*

Selection of parameters for the asymmetric membership functions and network weights has a significant effect on the network performance. If inappropriate values are given for the asymmetric Gaussian membership functions and network weights, the network will converge at a low speed or even get a bad control performance. In order to train the parameters of the SFNN effectively, an online parameter learning methodology, which is derived on the basis of Lyapunov theorem and Taylor expansion, is developed. This training scheme will increase the learning capability of the SFNN. By using (37), (26) can be rewritten as

$$\dot{s} = \tilde{\mathbf{w}}^T\tilde{\boldsymbol{\xi}} + \tilde{\mathbf{m}}^T\boldsymbol{\xi}_m\hat{\mathbf{w}} + \tilde{\boldsymbol{\sigma}}_l^T\boldsymbol{\xi}_{\sigma_l}\hat{\mathbf{w}} + \tilde{\boldsymbol{\sigma}}_r^T\boldsymbol{\xi}_{\sigma_r}\hat{\mathbf{w}} + \varepsilon - u_{rb}. \quad (38)$$

If ε exists, a robust controller will be considered to satisfy a specified L_2 tracking performance [22], [23]

$$\begin{aligned}\int_0^T s^2(t)dt \le s^2(0) + \delta^2\int_0^T \varepsilon^2(t)dt + (1/\eta_w)\tilde{\mathbf{w}}^T(0)\tilde{\mathbf{w}}(0) \\ + (1/\eta_m)\tilde{\mathbf{m}}^T(0)\tilde{\mathbf{m}}(0) + (1/\eta_{\sigma_l})\tilde{\boldsymbol{\sigma}}_l^T(0)\tilde{\boldsymbol{\sigma}}_l(0) \\ + (1/\eta_{\sigma_r})\tilde{\boldsymbol{\sigma}}_r^T(0)\tilde{\boldsymbol{\sigma}}_r(0)\end{aligned} \quad (39)$$

where δ is a prescribed attenuation constant. If the system starts with a set of initial conditions $s(0) = 0$, $\hat{\mathbf{w}}(0) = \mathbf{w}^*(0)$, $\hat{\mathbf{m}}(0) = \mathbf{m}^*(0)$, $\hat{\boldsymbol{\sigma}}_l(0) = \boldsymbol{\sigma}_l^*(0)$, and $\hat{\boldsymbol{\sigma}}_r(0) = \boldsymbol{\sigma}_r^*(0)$, the L_2 tracking performance in (39) can be rewritten as

$$\sup_{\varepsilon\in L_2[0,T]} \|s\|/\|\varepsilon\| \le \delta \quad (40)$$

where $\|s\|^2 = \int_0^T s^2(t)dt$ and $\|\varepsilon\|^2 = \int_0^T \varepsilon^2(t)dt$. If $\delta = \infty$, this is the case of minimum error tracking control without disturbance attenuation. To determine the adaptive laws of the parameters of SFNN appropriately and guarantee the closed-loop system stability, the Lyapunov function candidate is defined as

$$\begin{aligned}V = (1/2)s^2 + \tilde{\mathbf{w}}^T\tilde{\mathbf{w}}/2\eta_w + \tilde{\mathbf{m}}^T\tilde{\mathbf{m}}/2\eta_m + \tilde{\boldsymbol{\sigma}}_l^T\tilde{\boldsymbol{\sigma}}_l/2\eta_{\sigma_l} \\ + \tilde{\boldsymbol{\sigma}}_r^T\tilde{\boldsymbol{\sigma}}_r/2\eta_{\sigma_r}\end{aligned} \quad (41)$$

where η_w, η_m, η_{σ_l}, and η_{σ_r} are the learning rates and positive constants. Differentiating (41) with respect to time and using (38) yield

$$\dot{V} = s\dot{s} + \tilde{\mathbf{w}}^T\dot{\tilde{\mathbf{w}}}/\eta_w + \tilde{\mathbf{m}}^T\dot{\tilde{\mathbf{m}}}/\eta_m + \tilde{\boldsymbol{\sigma}}_l^T\dot{\tilde{\boldsymbol{\sigma}}}_l/\eta_{\sigma_l} + \tilde{\boldsymbol{\sigma}}_r^T\dot{\tilde{\boldsymbol{\sigma}}}_r/\eta_{\sigma_r}$$

$$= s(\tilde{\mathbf{w}}^T\tilde{\boldsymbol{\xi}} + \tilde{\mathbf{m}}^T\boldsymbol{\xi}_m\hat{\mathbf{w}} + \tilde{\boldsymbol{\sigma}}_l^T\boldsymbol{\xi}_{\sigma_l}\hat{\mathbf{w}} + \tilde{\boldsymbol{\sigma}}_r^T\boldsymbol{\xi}_{\sigma_r}\hat{\mathbf{w}} + \varepsilon - u_{rb})$$
$$+ \tilde{\mathbf{w}}^T\dot{\tilde{\mathbf{w}}}/\eta_w + \tilde{\mathbf{m}}^T\dot{\tilde{\mathbf{m}}}/\eta_m + \tilde{\boldsymbol{\sigma}}_l^T\dot{\tilde{\boldsymbol{\sigma}}}_l/\eta_{\sigma_l} + \tilde{\boldsymbol{\sigma}}_r^T\dot{\tilde{\boldsymbol{\sigma}}}_r/\eta_{\sigma_r}$$
$$= \tilde{\mathbf{m}}^T(s\boldsymbol{\xi}_m\hat{\mathbf{w}} + (1/\eta_m)\dot{\tilde{\mathbf{m}}}) + \tilde{\boldsymbol{\sigma}}_l^T(s\boldsymbol{\xi}_{\sigma_l}\hat{\mathbf{w}} + (1/\eta_{\sigma_l})\dot{\tilde{\boldsymbol{\sigma}}}_l)$$
$$+ \tilde{\boldsymbol{\sigma}}_r^T(s\boldsymbol{\xi}_{\sigma_r}\hat{\mathbf{w}} + (1/\eta_{\sigma_r})\dot{\tilde{\boldsymbol{\sigma}}}_r) + \tilde{\mathbf{w}}(s\hat{\boldsymbol{\xi}} + (1/\eta_w)\dot{\tilde{\mathbf{w}}})$$
$$+ s(\varepsilon - u_{rb}) . \quad (42)$$

Choose the adaptive laws as

$$\dot{\tilde{\mathbf{w}}} = -\dot{\hat{\mathbf{w}}} = -\eta_w s\hat{\boldsymbol{\xi}} , \quad (43)$$
$$\dot{\tilde{\mathbf{m}}} = -\dot{\hat{\mathbf{m}}} = -\eta_m s\boldsymbol{\xi}_m\hat{\mathbf{w}} , \quad (44)$$
$$\dot{\tilde{\boldsymbol{\sigma}}}_l = -\dot{\hat{\boldsymbol{\sigma}}}_l = -\eta_{\sigma_l} s\boldsymbol{\xi}_{\sigma_l}\hat{\mathbf{w}} , \quad (45)$$

and

$$\dot{\tilde{\boldsymbol{\sigma}}}_r = -\dot{\hat{\boldsymbol{\sigma}}}_r = -\eta_{\sigma_r} s\boldsymbol{\xi}_{\sigma_r}\hat{\mathbf{w}} \quad (46)$$

and the robust controller is designed as

$$u_{rb} = ((\delta^2 + 1)/2\delta^2)s . \quad (47)$$

Thus, (42) can be rewritten as

$$\dot{V} = s(\varepsilon - ((\delta^2 + 1)/2\delta^2)s)$$
$$= s\varepsilon - s^2/2 - s^2/2\delta^2$$
$$= -s^2/2 - (1/2)(s/\delta - \varepsilon\delta)^2 + (1/2)\varepsilon^2\delta^2$$
$$\le -(1/2)s^2 + (1/2)\varepsilon^2\delta^2 . \quad (48)$$

Assume $\varepsilon \in L_2[0,T]$, $\forall T \in [0,\infty)$. Integrating the above equation from $t = 0$ to $t = T$ yields

$$V(T) - V(0) \le -(1/2)\int_0^T s^2 dt + (1/2)\delta^2 \int_0^T \varepsilon^2 dt . \quad (49)$$

Since $V(t) \ge 0$, we can arrange (49) as follows

$$(1/2)\int_0^T s^2 dt \le V(0) + (1/2)\delta^2 \int_0^T \varepsilon^2 dt . \quad (50)$$

which is equivalent to inequality (39) , i.e., L_2 tracking performance. Hence, assume $\varepsilon \in L_2$, then the sliding surface s will converge to a certain boundary. It is implied that the tracking error e will also converge to a certain boundary.

To summarize, Fig. 3 shows the overall scheme of the robust self-organizing fuzzy-neural control proposed in this paper.

5. Simulation Results

In this section, the proposed RSOFNC is applied to control a second-order chaotic system to verify its effectiveness. The scheme emphasizes that the parameter and network structure of the SFNN can be online tuned by the proposed algorithm. Moreover, the appropriate network structure will be obtained by using the methods of the growing of membership functions and the pruning of fuzzy rules during the process of training adjustable parameters. Consider a second-order chaotic system such as the Duffing's equation describing a special nonlinear circuit or a pendulum moving in a viscous medium as follows [24]

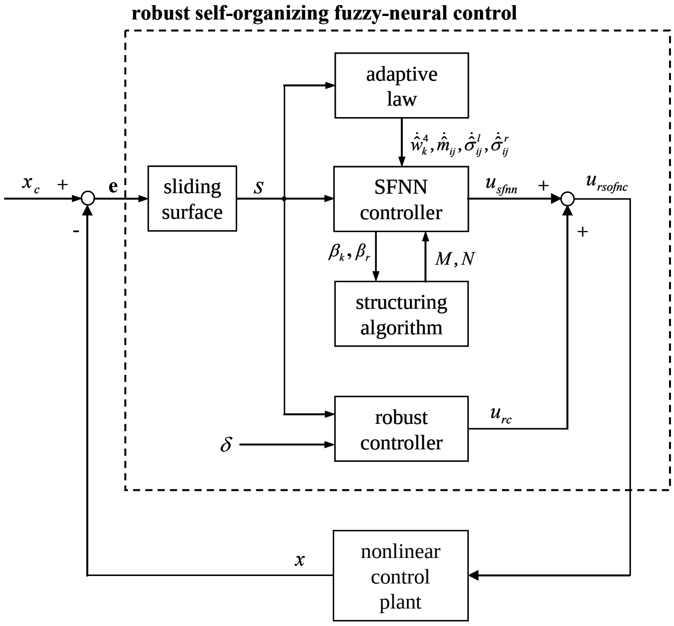

Figure 3. The block diagram of RSOFNC system.

$$\ddot{x} = f(\mathbf{x}) + u \quad (51)$$

where $f(\mathbf{x}) = -p\dot{x} - p_1 x - p_2 x^3 + q\cos(\omega t)$ is the system dynamics, t is the time variable, ω is the frequency, u is the control force, and p , p_1, p_2, and q are real constants. The system is known that the solutions of (51) may exhibit periodic depending on the choice of these constants, i.e., it is almost periodic and chaotic behavior. In order to observe the chaotic unpredictable behavior, the open-loop system behavior, i.e., $u = 0$, is simulated with $p = 0.4$, $p_1 = -1.1$, $p_2 = 1.0$, and $\omega = 1.8$. The phase plane plots with an initial condition point (-0.5,0) are shown in Figs. 4(a) and 4(b) for $q = 1.95$ and $q = 7.00$. The uncontrolled chaotic system has different trajectories for different q.

The parameters of RSOFNC system are selected as $k_1 = 2$, $k_2 = 1$, $\eta_w = 80$, $\eta_m = \eta_{\sigma_l} = \eta_{\sigma_r} = 0.2$, $G_{th} = 0.5$, $I_{th} = 0.1$, $P_{th} = 0.1$, $\tau_1 = 0.01$, $\tau_2 = 0.05$, and $\delta = 0.6$. All the gains in the proposed control system are chosen to achieve the best transient control performance considering the requirement of stability and possible operating conditions. The simulation results of RSOFNC for $q = 1.95$ and $q = 7.00$ are shown in Figs. 5 and 6. The tracking responses of state x are shown in Figs. 5(a) and 6(a). The tracking responses of state $\dot{x}$ are shown in Figs. 5(b) and 6(b). The associated control efforts of the RSOFNC are shown in Figs. 5(c) and 6(c). The number of fuzzy rules is shown in Figs. 5(d) and

6(d). The final shapes of membership functions are shown in Figs. 5(e) and 6(e). From the simulation results, the proposed RSOFNC system which includes SFNN with the asymmetric Gaussian membership function can perform successful control and achieve desired performance for different q of the second-order chaotic system. Meanwhile, a concise SFNN structure can be obtained by the proposed self-structuring mechanism and the online learning algorithms.

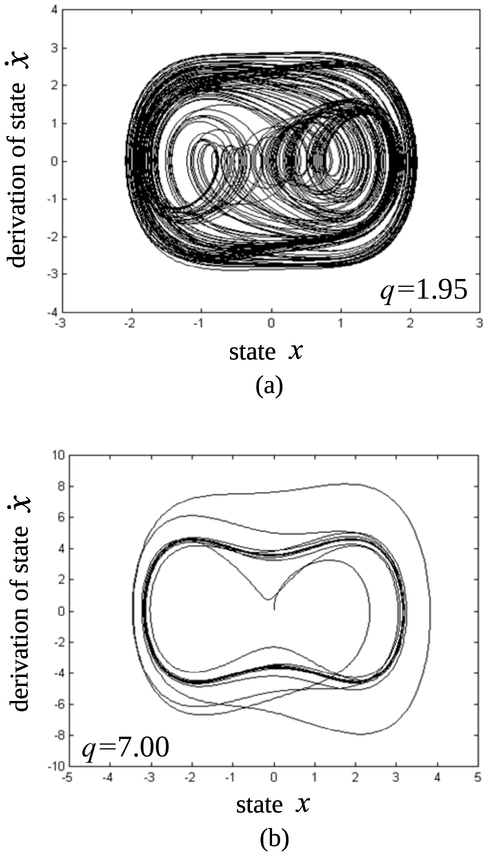

Figure 4. Phase plane of uncontrolled chaotic system.

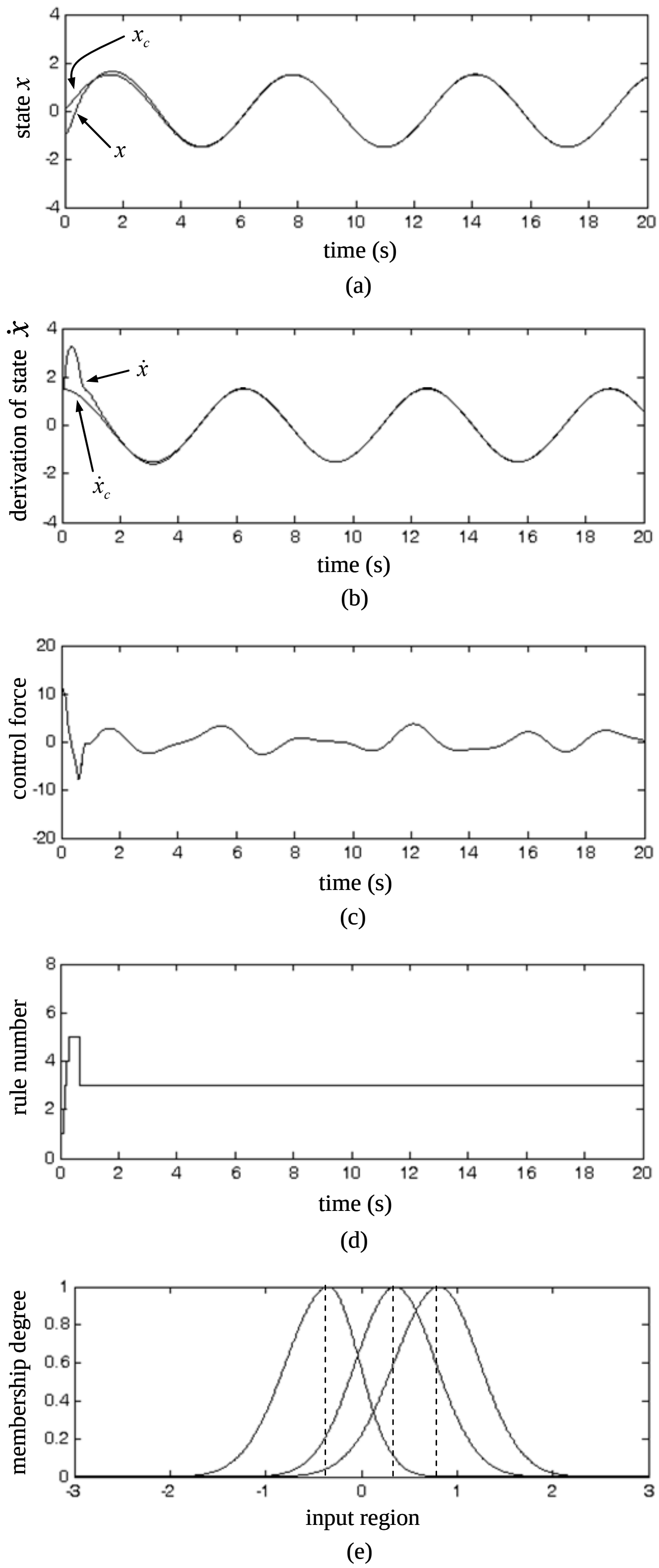

Figure 5. Simulation results of chaotic system for $q = 1.95$.

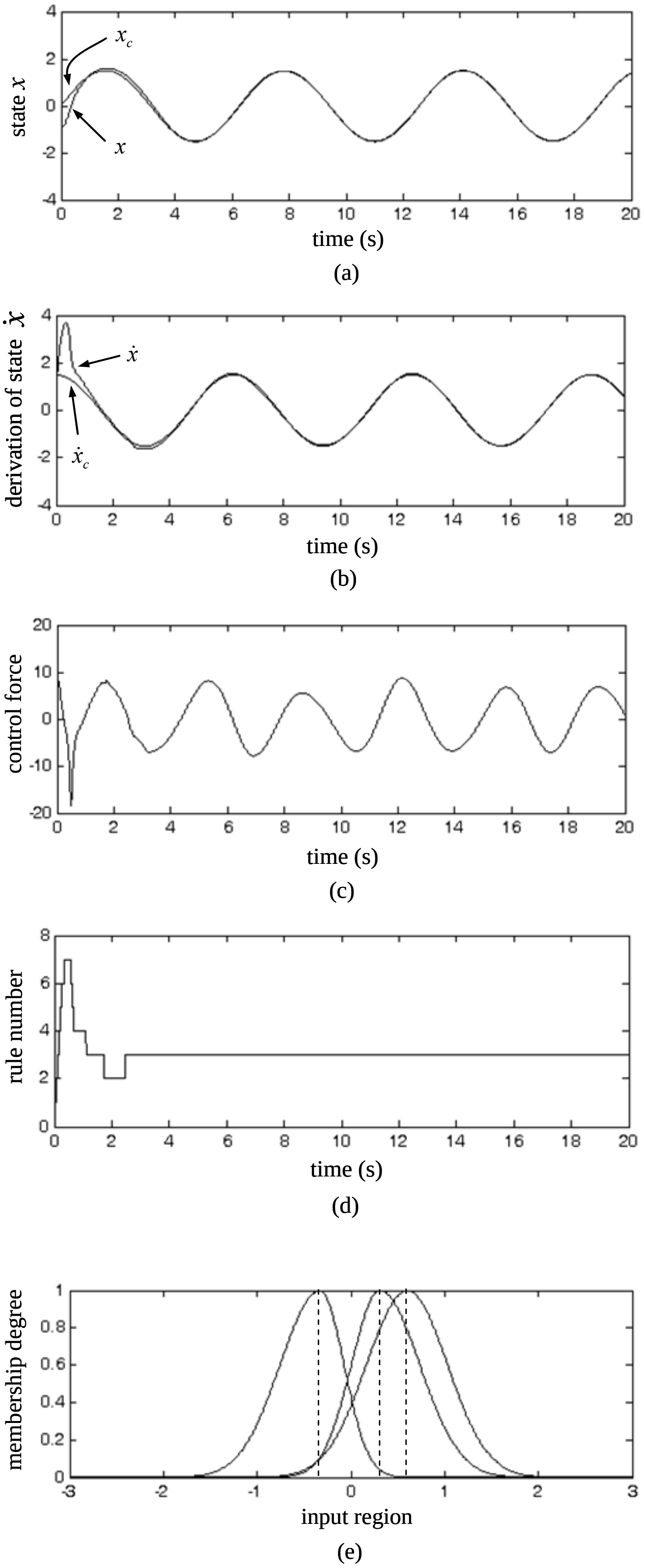

Figure 6. Simulation results of chaotic system for $q = 7.00$.

6. Conclusions

This paper develops a robust self-organizing fuzzy-neural control (RSOFNC) system, which comprises of a self-structuring fuzzy neural network (SFNN) controller and a robust controller. The SFNN controller with the structure and parameter learning and the asymmetric Gaussian functions is utilized to online approximate an ideal controller, and the robust controller is designed to achieve L_2 tracking performance with desired attenuation level. The major contributions of this paper are: 1) the structure learning algorithm which includes the growing and pruning of the membership functions and fuzzy rules has been developed to achieve favorable learning performance of network structure; 2) the combination of the Lyapunov theorem and the method of Taylor expansion derives the online parameter learning algorithms well; and 3) the RSOFNC is developed and applied successfully to control a second-order chaotic system.

7. Acknowledgment

The authors appreciate the partial financial support from the National Science Council of Republic of China under grant NSC 92-2213-E-009-009.

8. References

[1] J. M. Mendel, *A Prelude to Neural Networks: Adaptive and Learning Systems*, Englewood Cliffs, NJ: Prentice-Hall, 1994.
[2] O. Omidvar and D. L. Elliott, *Neural Systems for Control*, New York: Academic, 1997.
[3] C. T. Lin and C. S. G. Lee, *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*, Englewood Cliffs, NJ: Prentice -Hall, 1996.
[4] D. Nauck, F. Klawonn, and R. Kruse, *Foundations of Neuro-Fuzzy Systems*, New York: Wiley, 1997.
[5] C. H. Wang, H. L. Liu, and T. C. Lin, "Direct adaptive fuzzy-neural control with state observer and supervisory controller for unknown nonlinear dynamical systems," *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 1, pp. 39-49, Feb. 2002.
[6] C. M. Lin and C. F. Hsu, "Supervisory recurrent fuzzy neural network control of wing rock for slender delta wings," *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 5, pp. 733-742, Oct. 2004.
[7] Y. G. Leu, W. Y. Wang, and T. T. Lee, "Observer-based direct adaptive fuzzy-neural control for nonaffine nonlinear systems," *IEEE Trans. Neural Networks*, vol. 16, no. 4, pp. 853-861, July 2005.

[8] A. Chatterjee, K. Pulasinghe, K. Watanabe, and K. Izumi, “A particle-swarm-optimized fuzzy-neural network for voice-controlled robot systems,” *IEEE Trans. Industrial Electronics*, vol. 52, no. 6, pp. 1478-1489, Dec. 2005.

[9] S. Wu and M. J. Er, “Dynamic fuzzy neural networks - a novel approach to function approximation,” *IEEE Trans. Syst., Man, and Cybern., Pt. B: Cybern.*, vol. 30, no. 2, pp. 358-364, April 2000.

[10] S. Wu, M. J. Er, and Y. Gao, “A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks,” *IEEE Trans. Fuzzy Syst.*, vol. 9, no. 4, pp. 578-594, Aug. 2001.

[11] G. B. Huang, P. Saratchandran, and N. Sundararajan, “An efficient sequential learning algorithm for growing and pruning RBF (GAP-RBF) networks,” *IEEE Trans. Syst., Man, and Cybern., Pt. B: Cybern.*, vol. 34, no. 6, pp. 2284-2292, Dec. 2004.

[12] C. Li, C. Y. Lee, and K. H. Cheng, “Pseudo-error-based self-organizing neuro-fuzzy system,” *IEEE. Trans. Fuzzy Syst.*, vol. 12, no. 6, pp. 812-819, Dec. 2004.

[13] C. T. Lin, W. C. Cheng, and S. F. Liang, “An on-line ICA-mixture-model-based self-constructing fuzzy neural network.” *IEEE Trans. Circuits Syst. I*, vol. 52, no. 1, pp. 207-221, Jan. 2005.

[14] F. J. Lin, C. H. Lin, and P. H. Shen, “Self-constructing fuzzy neural network speed controller for permanent-magnet synchronous motor drive,” *IEEE Trans. Fuzzy Syst.*, vol. 9, no. 5, pp. 751-759, Oct. 2001.

[15] 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.

[16] F. J. Lin and C. H. Lin, “A permanent-magnet synchronous motor servo drive using self-constructing fuzzy neural network controller,” *IEEE Trans. Energy Conversion*, vol. 19, no. 1, pp. 66-72, March 2004.

[17] C. S. Velayutham and S. Kumar, “Some applications of an asymmetric subsethood product fuzzy neural inference system,” *IEEE FUZZ '03*, Vol. 1, pp. 202-207, May 2003.

[18] C. S. Velayutham and S. Kumar, “Asymmetric subsethood-product fuzzy neural inference system (ASuPFuNIS),” *IEEE Trans. on Neural Networks*, Vol. 16, pp. 160-174, Jan. 2005.

[19] C. J. Lin, T. C. Lin, and C. L. Lee, “An asymmetry subsethood-based neural fuzzy network,” *IJCNN '06*, pp. 2852-2858, July 2006.

[20] J. J. E. Slotine and W. P. Li, *Applied Nonlinear Control*, Englewood Cliffs, NJ: Prentice Hall, 1991.

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

[22] C. F. Hsu, C. M. Lin, and T. T. Lee, “Wavelet adaptive backstepping control for a class of nonlinear systems,” *IEEE Trans. Neural Networks*, vol. 17, no. 5, pp. 1175-1183, Sept. 2006.

[23] W. Y. Wang, M. L. Chan, C. C. J. Hsu, and T. T. Lee, “H^{∞} tracking-based sliding mode controller for uncertain nonlinear systems via an adaptive fuzzy-neural approach,” *IEEE Trans. Syst., Man, and Cybern., Part B*, vol. 32, no. 4, pp. 483-492, Aug. 2002.

[24] A. Loria, E. Panteley, and H. Nijmeijer, “Control of the chaotic Duffing equation with uncertainty in all parameter,” *IEEE Trans. Circuits Syst. I*, vol. 45, no. 6, pp. 1252-1255, Dec. 1998.

Ping-Zong Lin was born in Taipei, Taiwan, R.O.C., in 1978. He received the B.S. and M.S. degrees in electronic engineering from Fu-Jen Catholic University, Taipei, in 2001 and 2003, respectively. Since September 2003, he has been pursuing the Ph.D. degree in the Department of Electrical and Control Engineering at Chiao Tung University, Taiwan.

His research interests include sliding mode control, fuzzy logic system, neural network, adaptive control, genetic algorithms, and intelligent control.

Tsu-Tian Lee was born in Taipei, Taiwan, R.O.C., in 1949. He received the B.S. degree in control engineering from the National Chiao Tung University (NCTU), Hsinchu, Taiwan, in 1970, and the M.S. and Ph.D. degrees in electrical engineering from the University of Oklahoma, Norman, OK in 1972 and 1975, respectively.

In 1975, he was appointed Associate Professor and in 1978 Professor and Chairman of the Department of Control Engineering at NCTU. In 1981, he became Professor and Director of the Institute of Control Engineering, NCTU. In 1986, he was a Visiting Professor and in 1987, a Full Professor of Electrical Engineering at University of Kentucky, Lexington. In 1990, he was a Professor and Chairman of the Department of Electrical Engineering, National Taiwan University of Science and Technology (NTUST). In 1998, he became the Professor and Dean of the Office of Research and Development, NTUST. In 2000, he was with the Department of Electrical and Control Engineering, NCTU, where he served as a Chair Professor. Since 2004, he has been with National Taipei University of Technology (NTUT), where he is now the President of NTUT. He has published more than 200 refereed journal and conference papers in the areas of automatic control, robotics, fuzzy systems, and neural networks. His current research involves motion planning, fuzzy and neural control, optimal control theory and application, and walking machines.

Prof. Lee received the Distinguished Research Award from National Science Council, R.O.C., in 1991-1992, 1993-1994, 1995-1996, and 1997-1998, respectively, the TECO Sciences and Technology Award from TECO Foundation in 2003, the Academic Achievement Award in Engineering and Applied Science from the Ministry of Education, Republic of China, in 1998, and the National Endow Chair from Ministry of Education, Republic of China, in 2003. He was elected to the grade of IEEE Fellow in 1997 and IEE Fellow in 2000, respectively. He became a Fellow of New York Academy of Sciences (NYAS) in 2002. His professional activities include serving on the Advisory Board of Division of Engineering and Applied Science, National Science Council, serving as the Program Director, Automatic Control Research Program, National Science Council, and serving as an Advisor of Ministry of Education, Taiwan, and numerous consulting positions.

Prof. Lee has been actively involved in many IEEE activities. He has served as Member of Technical Program Committee and Member of Advisory Committee for many IEEE sponsored international conferences. He is now the Vice President for Conference and Meeting of the IEEE Systems, Man and Cybernetics Society.