Applied Soft Computing 13 (2013) 1620–1626

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Supervisory adaptive dynamic RBF-based neural-fuzzy control system design for unknown nonlinear systems

Chun-Fei Hsu [a], Chih-Min Lin [b,*], Rong-Guan Yeh [b]

[a] Department of Electrical Engineering, Tamkang University, No. 151, Yingzhuan Rd., Tamsui Dist., New Taipei City 25137, Taiwan
[b] Department of Electrical Engineering, Yuan Ze University, No. 135, Far-Eastern Rd., Chung-Li, Taoyuan County 320, Taiwan

ARTICLE INFO

Article history:
Received 7 May 2012
Received in revised form 25 October 2012
Accepted 30 December 2012
Available online 12 January 2013

Keywords:
Adaptive control
Sliding-mode control
Neural-fuzzy system
Chaotic system
Inverted pendulum

ABSTRACT

Many published papers show that a TSK-type fuzzy system provides more powerful representation than a Mamdani-type fuzzy system. Radial basis function (RBF) network has a similar feature to the fuzzy system. As this result, this article proposes a dynamic TSK-type RBF-based neural-fuzzy (DTRN) system, in which the learning algorithm not only online generates and prunes the fuzzy rules but also online adjusts the parameters. Then, a supervisory adaptive dynamic RBF-based neural-fuzzy control (SADRNC) system which is composed of a DTRN controller and a supervisory compensator is proposed. The DTRN controller is designed to online estimate an ideal controller based on the gradient descent method, and the supervisory compensator is designed to eliminate the effect of the approximation error introduced by the DTRN controller upon the system stability in the Lyapunov sense. Finally, the proposed SADRNC system is applied to control a chaotic system and an inverted pendulum to illustrate its effectiveness. The stability of the proposed SADRNC scheme is proved analytically and its effectiveness has been shown through some simulations.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Since a mathematical model of control plants is difficult to develop accurately due to the lack of knowledge of some parameters or external disturbances, a controller design should take an appropriate action to counteract the presence of uncertainties and external disturbances. To attack this problem, there have been considerable interests in exploring the applications of a neural-fuzzy system to deal with the uncertain nonlinear systems [4,5,15,27]. The most important feature of the adaptive neural-fuzzy control schemes is the self-learning ability that neural-fuzzy systems are used to approximate arbitrary linear or nonlinear mappings through online learning algorithms without requiring preliminary offline tuning. The learning algorithms are based on the Lyapunov stability theorem or gradient decent method, so that the stability of the adaptive neural-fuzzy control system can be guaranteed.

Generally, the neural-fuzzy systems can be divided into two types which are Mamdani-type neural-fuzzy system and Takagi-Sugeno-Kang (TSK)-type neural-fuzzy system [20]. The output weights are equipped with singleton-type form in Mamdani-type neural-fuzzy systems but with functional-type form in TSK-type neural-fuzzy systems. Thus, the TSK-type neural-fuzzy systems provide more powerful representation than Mamdani-type neural-fuzzy systems. Recently, the adaptive TSK-type neural-fuzzy control schemes have grown rapidly in many previous published papers [2,11,14,17,21,25]. Though the control performance of the adaptive TSK-type neural-fuzzy control are acceptable, the learning algorithm only considers the parameter learning but does not consider the structure learning. The structure of the used neural-fuzzy system is determined by some trial-and-error tuning procedure. It is a trade-off between the approximation performance and the number of fuzzy rules.

An adaptive neuro fuzzy inference system (ANFIS) is a kind of neural network that is based on Takagi-Sugeno fuzzy inference system. Since it integrates both neural networks and fuzzy logic principles, it has potential to capture the benefits of both in a single framework. Its inference system corresponds to a set of fuzzy IF-THEN rules that have learning capability to approximate nonlinear functions [13]. Recently, the Radial basis function (RBF) network has been successfully applied in many fields, such as identification, control and pattern recognition [6,8,10,18,22,23]. RBF network can be viewed as a fuzzy rule base with specified membership functions and fuzzy inference operations. The output value is calculated using the weighted sum method. Then, the number of hidden nodes in the RBF network is the same as the number of fuzzy IF-THEN rules in the fuzzy system [22].

To attack the problem of determining the number of hidden neurons in an RBF network, several dynamic structure learning algorithms which can vary its structure dynamically to keep the

* Corresponding author.
E-mail addresses: fei@ee.tku.edu.tw (C.-F. Hsu), cml@saturn.yzu.edu.tw (C.-M. Lin).

1568-4946/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2012.12.028

prescribed approximation accuracy have been proposed [1,12,16]. Bortman and Aladjem [1] used a significance criterion and a growth criterion to design the structure learning algorithm for an RBF network. However, it lacks the real-time adaptation ability. Huang et al. [12] present a generalized growing and pruning algorithm. The growing and pruning strategy is based on linking the required learning accuracy with the significance of the nearest. However, the algorithm is only suitable for low dimensional applications. Leung and Tsoi [16] studied a structure learning algorithm for recurrent RBF networks. It can delete some unimportant RBF nodes based on the recursive least square approach; however, the computation loading is heavy.

This article proposes a dynamic TSK-type RBF-based neural-fuzzy (DTRN) system which can vary its structure dynamically to keep the prescribed approximation accuracy with a simple computation. Since the output value of the DTRN system is calculated with functional-type form, it can provide more powerful representation than neural-fuzzy system and RBF network. It starts with a small number of fuzzy rules and then adds new fuzzy rules if there is no existing fuzzy rules can efficiently represent the input data. Moreover, some of existing fuzzy rules can be removed if the removing condition is satisfied. Then, a supervisory adaptive dynamic RBF-based neural-fuzzy control (SADRNC) system which is composed of a DTRN controller and a supervisory compensator is proposed. To show the effectiveness of the proposed SADRNC system, it is applied to control a chaotic system and an inverted pendulum. In the simulation study, it is shown that the proposed SADRNC system can achieve a favorable control performance. In summary, according to the learning methods including structure learning and parameter learning, the proposed learning algorithms can be classified as online learning algorithm.

2. Problem statement

A model of nonlinear systems can be expressed in the following form as

$$\ddot{x}(t) = f(\mathbf{x}, t) + g(\mathbf{x}, t)u(t) \tag{1}$$

where $\boldsymbol{x} = [x(t), \dot{x}(t)]^T$ is the state vector of the system, $f(\mathbf{x}, t)$ and $g(\mathbf{x},t)$ are the system dynamics and $u(t)$ is the control input. Without losing generality, assume $g(\mathbf{x}, t) > 0$ for all $\mathbf{x}$. The control problem is to find a control law so the state $x(t)$ can track a command $x_c(t)$ closely. To determine a control law, define a tracking error as

$$e(t) = x_c(t) - x(t). \tag{2}$$

If the system dynamics are known, there exists an ideal controller as [24]

$$u^* = \frac{1}{g(x, t)}[-f(\mathbf{x}, t) + \ddot{x}_c(t) + k_1\dot{e}(t) + k_2 e(t)] \tag{3}$$

where k_1 and k_2 are nonzero positive constants. Applying (3) into (1) yields

$$\ddot{e}(t) + k_1\dot{e}(t) + k_2 e(t) = 0. \tag{4}$$

If k_1 and k_2 are chosen to correspond to the coefficients of a Hurwitz polynomial, it implies that $\lim_{t\to\infty} e(t) = 0$ [24]. Since the terms $f(\mathbf{x}, t)$ and $g(\mathbf{x}, t)$ may be unknown or perturbed, the ideal controller cannot be precisely obtained.

To attack this problem, many approaches under various hypotheses to account for the uncertainties have been introduced to treat the control problem in the last few decades. Sliding-mode control schemes have been developed as popular robust strategies to treat uncertain control systems [24]. But, designing a sliding-mode controller requires nominal model of the nonlinear system. Based on the linearization method, rewriting (1), the nominal model can be represented as follows

$$\ddot{x}(t) = f_n(\mathbf{x}, t) + g_n(\mathbf{x}, t)u(t) \tag{5}$$

where $f_n(\mathbf{x}, t)$ and $g_n(\mathbf{x}, t)$ represent the nominal behavior of $f(\mathbf{x}, t)$ and $g(\mathbf{x}, t)$, respectively. If the system uncertainties occur, i.e., parameters of system deviate from the nominal value, the controlled system can be modified as

$$\begin{aligned}\ddot{x}(t) &= [f_n(\mathbf{x}, t) + \Delta f(\mathbf{x}, t)] + [g_n(\mathbf{x}, t) + \Delta g(\mathbf{x}, t)]u(t) \\ &= f_n(\mathbf{x}, t) + g_n(\mathbf{x}, t)u(t) + w(\mathbf{x}, t)\end{aligned} \tag{6}$$

where $\Delta f(\mathbf{x}, t)$ and $\Delta g(\mathbf{x}, t)$ denote the system uncertainties and $w(\mathbf{x}, t)$ is called the lumped uncertainty which is defined as $w(\mathbf{x}, t) = \Delta f(\mathbf{x}, t) + \Delta g(\mathbf{x}, t)u(t)$. The lumped uncertainty is assumed to be bounded, i.e., $\left|w(\mathbf{x}, t)\right| \le W$ where W is a given positive constant. Then, a sliding surface is defined as

$$s(t) = \dot{e}(t) + k_1 e(t) + k_2 \int_0^t e(\tau)d\tau \tag{7}$$

where k_1 and k_2 are nonzero positive constants. The sliding-mode control law is given as [24]

$$u_{sm} = u_{eq} + u_{ht} \tag{8}$$

The equivalent controller u_{eq} is represented as

$$u_{eq} = \frac{1}{g_n(\mathbf{x}, t)}[-f_n(\mathbf{x}, t) + \ddot{x}_c(t) + k_1\dot{e}(t) + k_2 e(t)] \tag{9}$$

and the hitting controller u_{ht} is designed as

$$u_{ht} = \frac{1}{g_n(\mathbf{x}, t)}[W sgn(s(t))] \tag{10}$$

where sgn(·) is a sign function. Substituting (8) into (6) yields

$$\ddot{e}(t) + k_1\dot{e}(t) + k_2 e(t) = -w(\mathbf{x}, t) - W sgn(s(t)) = \dot{s}(t). \tag{11}$$

Consider the candidate Lyapunov function in the following form as

$$V_1(t) = \frac{1}{2}s^2(t). \tag{12}$$

Differentiating (12) with respect to time and using (11), we can obtain

$$\begin{aligned}\dot{V}_1(t) &= s(t)\dot{s}(t) = -w(\mathbf{x}, t)s(t) - W\left|s(t)\right| \le \left|w(\mathbf{x}, t)\right|\left|s(t)\right| - W\left|s(t)\right| \\ &= -(W - \left|w(\mathbf{x}, t)\right|)\left|s(t)\right| \le 0.\end{aligned} \tag{13}$$

In summary, the sliding-mode controller can guarantee the stability in the sense of the Lyapunov theorem [24]. However, a large control gain W often causes an outcome of a large amount of chattering. The chattering phenomena in control efforts will wear the bearing mechanism.

3. SADRNC system design

3.1. Structure learning of a DTRN system

In Fig. 1, the proposed DTRN system includes an input layer, a membership layer, a TSK layer and an output layer. The signal propagation and basic function in each space are introduced in the following.

Layer 1 – Input layer: No function is performed in this layer. The node only transmits input values to layer 2.

Layer 2 – Membership layer: In this layer, the Gaussian function is adopted as the membership function. The output of the i-th fuzzy rules is designed as

$$h_i = \exp[-\frac{(s(t) - c_i)^2}{\sigma_i^2}],\ i = 1, 2, \ldots, n(N) \tag{14}$$

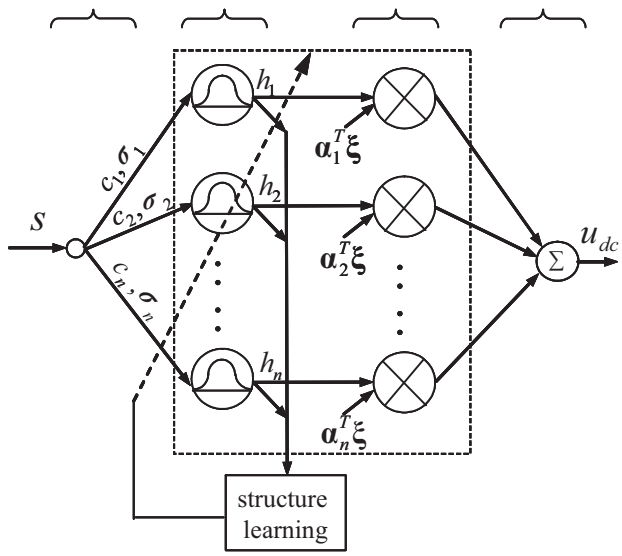

Fig. 1. Dynamic TSK-type RBF-based neural-fuzzy system.

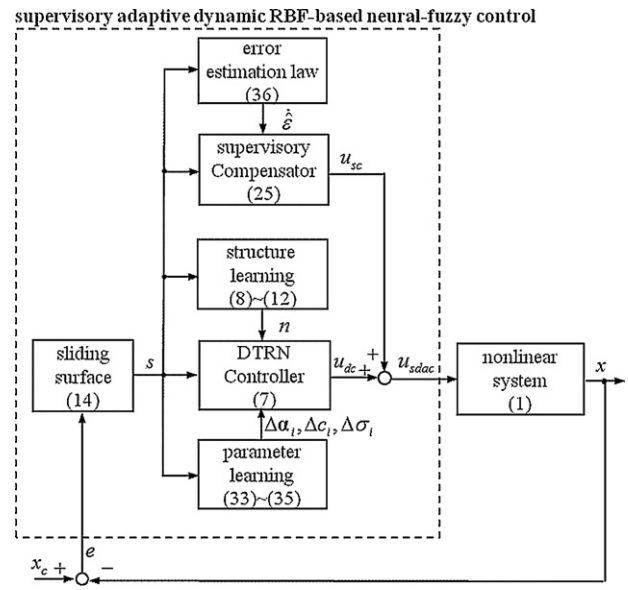

Fig. 2. The block diagram of the SADRNC system.

where $n(N)$ is the number of the fuzzy rules at the N-th sampling time and c_i and σ_i denote the center and width of the membership function, respectively.

Layer 3 – TSK layer: The TSK layer represents the linear combination function. Each node in this layer is denoted by

$$u_i = \alpha_{i0} + \alpha_{i1}s(t) = \alpha_i^T\xi \tag{15}$$

where α_{i0} and α_{i1} are the parameters designed by the designer, $\alpha_i = [\alpha_{i0}, \alpha_{i1}]^T$ and $\xi = [1, s(t)]^T$.

Layer 4 – Output layer: The output node together with links connected it act. The single node computes the overall output as the summation of all incoming signals. The output of the DTRN system can be represented as

$$u_{dc} = \sum_{i=1}^{n(N)} u_i h_i = \sum_{i=1}^{n(N)} \alpha_i^T \xi h_i. \tag{16}$$

Determining an appropriate number of fuzzy rules $n(N)$ is an important issue because of the trade-off between a computation loading and a learning performance. In general, a larger number of fuzzy rules results in a smaller approximation error.

To attack the problem, this article proposes an online structuring learning algorithm. The first step of the structural learning is to determine whether or not to add a new fuzzy rule. The weighting h_i in (14) can be represented as the degree to which the incoming data belong to the existing fuzzy rules. For an incoming data, find [9]

$$K = arg \max_{1 \le i \le n(N)} h_i. \tag{17}$$

It implies that the K-th fuzzy rule has the maximum firing weight. If $h_K \le G_{th}$ is satisfied, where $G_{th} \in (0, 1)$ a pre-given threshold, then a new fuzzy rule is generated. The initial parameters associated with the new fuzzy rule are given by

$$c_{n(N)+1} = s \tag{18}$$

$$\sigma_{n(N)+1} = \sigma \tag{19}$$

$$\alpha_{n(N)+1} = 0 \tag{20}$$

where σ is a pre-specified constant. To avoid the structure of the DTRN system growing unboundedly, an algorithm for pruning the inappropriate fuzzy rules is needed to prevent the endless growing of the DTRN system and the computation loading. A pruning algorithm based on the density is studied in this article, where the density is the number of times each fuzzy rule is used in the algorithm. Thus, each fuzzy rule has its own density. A density of the i-th fuzzy rule is defined as

$$d_i(N+1) = \begin{cases} d_i(N) & , if\ h_i < \rho \\ d_i(N) + 1 & , if\ h_i \ge \rho \end{cases} \tag{21}$$

where the initial value of d_i is 0 in each time iterations and ρ is a designed constant. The least-important fuzzy rule is the one that has the smallest density. After some iterations, if $d_i \le D_{th}$ are satisfied, where D_{th} a pre-specified threshold, then the i-th fuzzy rule is deleted. The proposed DTRN system can delete some unimportant fuzzy rules based on the density approach with a simple computation. Thus, it is more suitable for real-time implementation than other dynamic structure learning algorithms [1,12,16].

3.2. Parameter learning of a DTRN system

The proposed SADRNC system as shown in Fig. 2 is composed of a DTRN controller and a supervisory compensator, i.e.

$$u_{sdac} = u_{dc} + u_{sc}. \tag{22}$$

Substituting (22) into (1) and using (3), the error dynamic equation can be obtained as

$$\ddot{e}(t) + k_1\dot{e}(t) + k_2 e(t) = g(\mathbf{x}, t)(u^* - u_{dc} - u_{sc}) = \dot{s}(t). \tag{23}$$

Multiplying both sides of (23) by $s(t)$ gives

$$s(t)\dot{s}(t) = s(t)g(\mathbf{x}, t)(u^* - u_{dc} - u_{sc}). \tag{24}$$

The parameter learning algorithm is derived based on a gradient descent algorithm and aims to minimize $s(t)\dot{s}(t)$ for achieving fast convergence of s. According to the gradient descent method, the correction $\Delta\alpha_i$ applied to the weight α_i is given by [19]

$$\Delta\alpha_i(N) = -\eta_\alpha \frac{\partial s(t)\dot{s}(t)}{\partial \alpha_i} = -\eta_\alpha \frac{\partial s(t)\dot{s}(t)}{\partial u_{dc}} \frac{\partial u_{dc}}{\partial \alpha_i} = \eta_\alpha s(t) g(\mathbf{x}, t) h_i \xi \tag{25}$$

where η_α is a positive learning rate. The weight α_i is updated according to the following equation

$$\alpha_i(N+1) = \alpha_i(N) + \Delta\alpha_i(N). \tag{26}$$

Moreover, the corrections Δc_i and $\Delta\sigma_i$ applied to the center and width, respectively, for the i-th fuzzy rules can be obtained as [19]

$$\Delta c_i(N) = -\eta_c \frac{\partial s(t)\dot{s}(t)}{\partial c_i} = -\eta_c \frac{\partial s(t)\dot{s}(t)}{\partial u_{dc}} \frac{\partial u_{dc}}{\partial h_i} \frac{\partial h_i}{\partial c_i}$$
$$= 2\eta_c s(t) g(\mathbf{x}, t)\alpha_i \xi \frac{(s(t) - c_i)}{\sigma_i^2} h_i \tag{27}$$

$$\Delta\sigma_i(N) = -\eta_\sigma \frac{\partial s(t)\dot{s}(t)}{\partial \sigma_i} = -\eta_\sigma \frac{\partial s(t)\dot{s}(t)}{\partial u_{dc}} \frac{\partial u_{dc}}{\partial h_i} \frac{\partial h_i}{\partial \sigma_i}$$
$$= 2\eta_\sigma s(t) g(\mathbf{x}, t)\alpha_i \xi \frac{(s(t) - c_i)^2}{\sigma_i^3} h_i \tag{28}$$

where η_c and η_σ are positive learning rates. The center and width are updated as following

$$c_i(N+1) = c_i(N) + \Delta c_i(N) \tag{29}$$

$$\sigma_i(N+1) = \sigma_i(N) + \Delta\sigma_i(N). \tag{30}$$

3.3. Stability analysis

Since the number of fuzzy rules in the DTRN system is finite for real-time practical applications, there exists an approximation error between the ideal controller and the DTRN controller, i.e.

$$u^* - u_{nc} = \varepsilon(t) \tag{31}$$

where $\varepsilon(t)$ denotes the approximation error. Substituting (31) into (23) yields

$$\dot{s}(t) = g(\mathbf{x}, t)[\varepsilon(t) - u_{sc}] \tag{32}$$

The approximation error cannot be easily measured in practical applications. In this article, the supervisory compensator is designed as

$$u_{sc} = \hat{\varepsilon}(t) + ks(t) \tag{33}$$

where $\hat{\varepsilon}(t)$ denotes the estimated value of approximation error and k is a small positive constant. Substituting (33) into (32) yields

$$\dot{s}(t) = g(\mathbf{x}, t)[\varepsilon(t) - \hat{\varepsilon}(t) - ks(t)] = g(\mathbf{x}, t)[\tilde{\varepsilon}(t) - ks(t)] \tag{34}$$

where $\tilde{\varepsilon}(t) = \varepsilon(t) - \hat{\varepsilon}(t)$. To guarantee the stability of the SADRNC system, a Lyapunov function candidate is defined as

$$V_2(t) = \frac{1}{2}s^2(t) + \frac{1}{2\eta_\varepsilon}\tilde{\varepsilon}^2(t) \tag{35}$$

where η_ε is a positive learning rate. Differentiating (35) with respect to time and using (34) obtain

$$\dot{V}_2(t) = s(t)\dot{s}(t) + \frac{1}{\eta_\varepsilon}\tilde{\varepsilon}(t)\dot{\tilde{\varepsilon}}(t) = s(t)g(\mathbf{x}, t)[\tilde{\varepsilon}(t) - ks(t)] + \frac{1}{\eta_\varepsilon}\tilde{\varepsilon}(t)\dot{\tilde{\varepsilon}}(t)$$
$$= \tilde{\varepsilon}(t)[s(t)g(x, t) + \frac{1}{\eta_\varepsilon}\dot{\tilde{\varepsilon}}(t)] - kg(\mathbf{x}, t)s^2(t) \tag{36}$$

For achieving $\dot{V}_2(t) \le 0$, the error estimation law is designed as

$$\dot{\hat{\varepsilon}}(t) = -\dot{\tilde{\varepsilon}}(t) = \eta_\varepsilon s(t) g(\mathbf{x}, t) \tag{37}$$

then (36) can be rewritten as

$$\dot{V}_2(t) = -kg(\mathbf{x}, t)s^2(t) \le 0. \tag{38}$$

Since $\dot{V}_2(t)$ is negative semidefinite, that is $V_2(t) \le V_2(0)$, it implies that $s(t)$ and $\tilde{\varepsilon}(t)$ are bounded. Let function $\Omega(\tau) \equiv kg(\mathbf{x}, t)s^2(t) \le -\dot{V}_2(t)$, and integrate $\Omega(t)$ with respect to time, then it is obtained that

$$\int_0^t \Omega(\tau)d\tau \le V_2(0) - V_2(t). \tag{39}$$

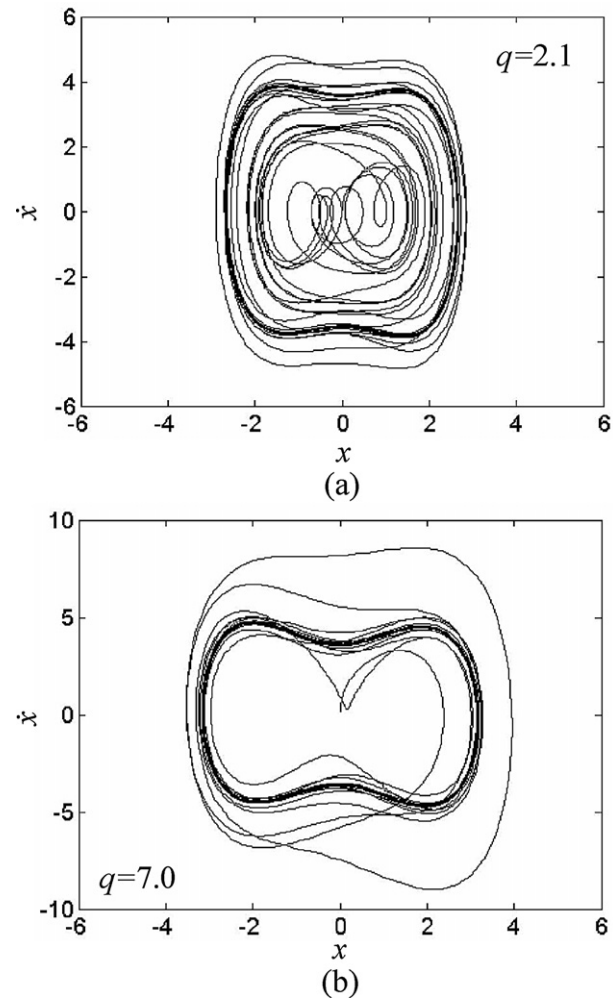

Fig. 3. Behavior of uncontrolled chaotic system.

Because $V_2(0)$ is bounded and $V_2(t)$ is nonincreasing and bounded, the following result can be obtained

$$\lim_{t\to\infty} \int_0^t \Omega(\tau)d\tau < \infty \tag{40}$$

Moreover, since $\dot{\Omega}(t)$ is bounded, by Barbalat's Lemma, $\lim_{t\to\infty} \Omega(t) = 0$ [24]. That is, $s(t) \to 0$ as $t \to \infty$. As a result, the stability of the proposed SADRNC system can be guaranteed.

Since the terms $g(\mathbf{x}, t)$ may be unknown or perturbed, the adaptive laws (25), (27), (28) and (37) cannot be precisely obtained. To deal with the unavailable system dynamics, $g(\mathbf{x}, t)$ is rewritten as $|g(\mathbf{x}, t)|\mathrm{sgn}(g(\mathbf{x}, t))$. Therefore, the update laws (25), (27) and (28) can be rewritten as follows

$$\Delta\alpha_i(N) = \beta_\alpha s(t) h_i \xi \tag{41}$$

$$\Delta c_i(N) = 2\beta_c s(t)\alpha_i \xi \frac{(s(t) - c_i)}{\sigma_i^2} h_i \tag{42}$$

$$\Delta\sigma_i = 2\beta_\sigma s(t)\alpha_i \xi \frac{(s(t) - c_i)^2}{\sigma_i^3} h_i \tag{43}$$

and the error estimation law (37) can be rewritten as follows

$$\dot{\hat{\varepsilon}} = -\dot{\tilde{\varepsilon}} = \beta_\varepsilon s(t) \tag{44}$$

where the terms $\eta_\alpha|g(\mathbf{x}, t)|$, $\eta_c|g(\mathbf{x}, t)|$, $\eta_\sigma|g(\mathbf{x}, t)|$ and $\eta_\varepsilon|g(\mathbf{x}, t)|$ are absorbed into the new positive learning rates β_α, β_c, β_σ and β_ε,

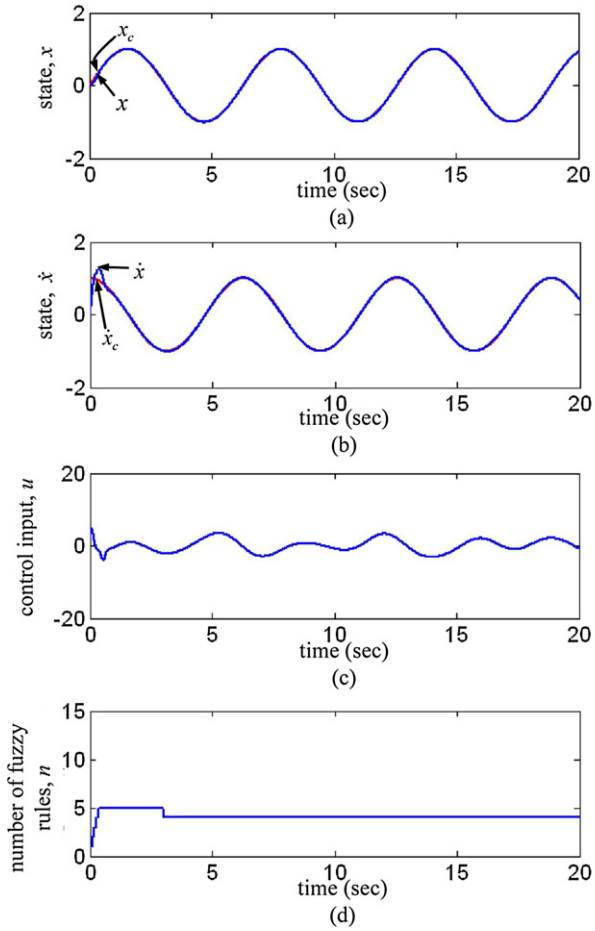

Fig. 4. Simulation results of the chaotic system for $q = 2.1$.

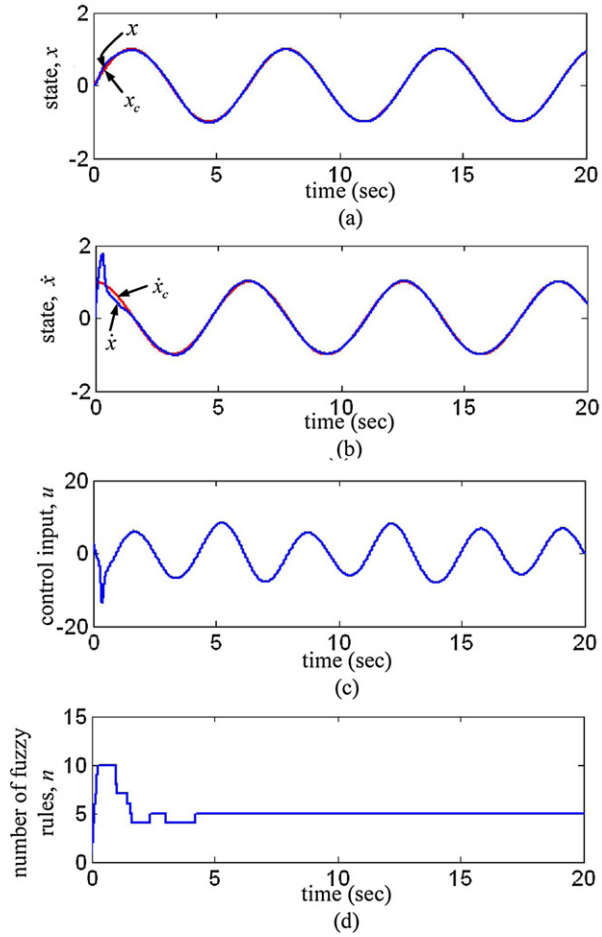

Fig. 5. Simulation results of the chaotic system for $q = 7.0$.

respectively. These new learning rates are specified by the users. Consequently, only the sign of $g(\mathbf{x}, t)$ is required in the design procedure and it can be easily obtained from the physical characteristic of the control plants.

4. Simulation results

In this section, the proposed SADRNC system is applied to control a chaotic system and an inverted pendulum. It should be emphasized that the development of the SADRNC system does not need to know the control system dynamics. For practical implementation, the parameters and the number of the fuzzy rules in the DTRN system can be online tuned by the proposed algorithm.

4.1. Chaotic system

Chaotic phenomena have been observed in numerous fields of science such as physics, chemistry, biology and ecology [3,7,26]. Chaotic behavior has been detected in some fields such as laser technology, plasma, mechanical engineering and chemistry. An interesting subject in chaos theory is to eliminate the chaotic behavior by means of control systems. However, in some occasions, it is desired to excite the chaotic mode of the system. It can be observed in many nonlinear circuits and mechanical systems. Consider a second-order chaotic system such as [3]

$$\begin{aligned}\ddot{x}(t) &= -p\dot{x}(t) - p_1 x(t) - p_2 x^3(t) + q\cos(\omega t) + u(t)\\ &= f(\mathbf{x}, t) + u(t)\end{aligned} \tag{45}$$

where $f(\mathbf{x}, t) = -p\dot{x}(t) - p_1 x(t) - p_2 x^3(t) + q\cos(\omega t)$ is the system dynamics, t is the time variable, ω is the frequency, $u(t)$ is the control input, and $p = 0.4$, $p_1 = -1.1$, $p_2 = 1.0$, $\omega = 1.8$ and q is a constant. Depending on the choices of these constants, the solutions of system (45) may display complex phenomena, including various periodic orbits behaviors and some chaotic behaviors. To observe the complex phenomena, the time responses of the uncontrolled chaotic system with initial point (0,0) for $q = 2.1$ and $q = 7.0$ are shown in Fig. 3(a) and (b), respectively. For the time responses with $q = 2.1$, an uncontrolled chaotic trajectory can be found, but a period motion chaotic trajectory happens with $q = 7.0$. It is shown that the uncontrolled chaotic system has different trajectories for different system parameters.

Then, the proposed SADRNC system is applied to control the chaotic system. The parameters of the SADRNC system are selected as $k_1 = 2$, $k_2 = 1$, $\beta_\alpha = 0.1$, $\beta_c = \beta_\sigma = 0.01$, $\eta_\varepsilon = 1.0$, $k = 1$, $\sigma = 2.0$, $G_{th} = 0.6$, $D_{th} = 300$, and $\rho = 0.3$. The choices of these values are through some trials considering the requirement of stability condition. The simulation results of the SADRNC system for $q = 2.1$ and $q = 7.0$ are shown in Figs. 4 and 5, respectively. The tracking responses of state x are shown in Figs. 4(a) and 5(a), the tracking responses of state $\dot{x}$ are shown in Figs. 4(b) and 5(b), the control inputs are shown in Figs. 4(c) and 5(c), and the numbers of fuzzy rules are shown in Figs. 4(d) and 5(d). The simulation results show that not only a satisfied tracking performance can be achieved after the controller parameters having been learned. Moreover, the structure learning approach demonstrates the properties of generating or pruning the fuzzy rules of the DTRN system automatically.

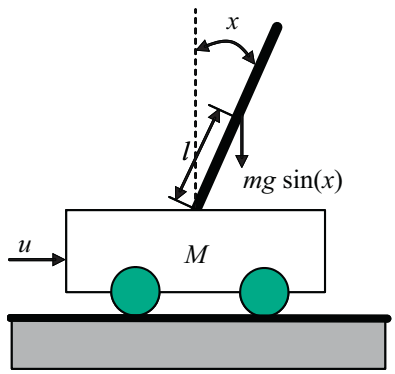

Fig. 6. The inverted pendulum system.

4.2. Inverted pendulum

Consider an inverted pendulum stabilizing problem as shown in Fig. 6 to demonstrate the robustness of the proposed control scheme. Let x be the angle of the pendulum with respect to the vertical line. The dynamic equations of the inverted pendulum system are [15,20]

$$\ddot{x}(t) = f(\mathbf{x}, t) + g(\mathbf{x}, t)u(t) \tag{46}$$

where $\mathbf{x} = [x(t), \dot{x}(t)]^T$ is the state vector, $f(\mathbf{x}, t) = \frac{ml\dot{x}\sin(x)\cos(x) - (M+m)g\sin(x)}{ml\cos^2(x) - \frac{4}{3}l(M+m)}$ and $g(\mathbf{x}, t) = \frac{-\cos(x)}{ml\cos^2(x) - \frac{4}{3}l(M+m)}$ are the system dynamics, $u(t)$ is the control input, M is the mass of cart, m is the mass of rod, $g = 9.8$ m/sec^2 is the acceleration due to gravity, and l is the half length of rod. In this example, it is assumed that $M = 1$ kg, $m = 0.1$ kg, and $l = 0.5$ m. The parameters of the SADRNC system are selected as $k_1 = 2$, $k_2 = 1$, $\beta_\alpha = 0.1$, $\beta_c = \beta_\sigma = 0.01$, $\eta_\varepsilon = 1.0$, $k = 1$, $\sigma = 2.0$, $G_{th} = 0.6$, $D_{th} = 600$, and $\rho = 0.3$. According to the initial states, two simulation cases are simulated. The setting of Case 1 is $(x, \dot{x}) = (0.25, 0)$ and this of Case 2 is $(x, \dot{x}) = (-0.25, 0)$. The simulation results of the SADRNC system are shown in Figs. 7 and 8 for Case 1 and Case 2, respectively. The tracking responses of the state $x(t)$ are shown in Figs. 7(a) and 8(a), the tracking responses of the state $\dot{x}(t)$ are shown in Figs. 7(b) and 8(b), the control inputs are shown in Figs. 7(c) and 8(c), and the numbers of fuzzy rules are shown in Figs. 7(d) and 8(d). The simulation results verify that the favorable tracking performance and no chattering phenomena can be achieved by the proposed SADRNC system. Moreover, the structure learning approach demonstrates the properties of generating or pruning the fuzzy rules of the DTRN system automatically.

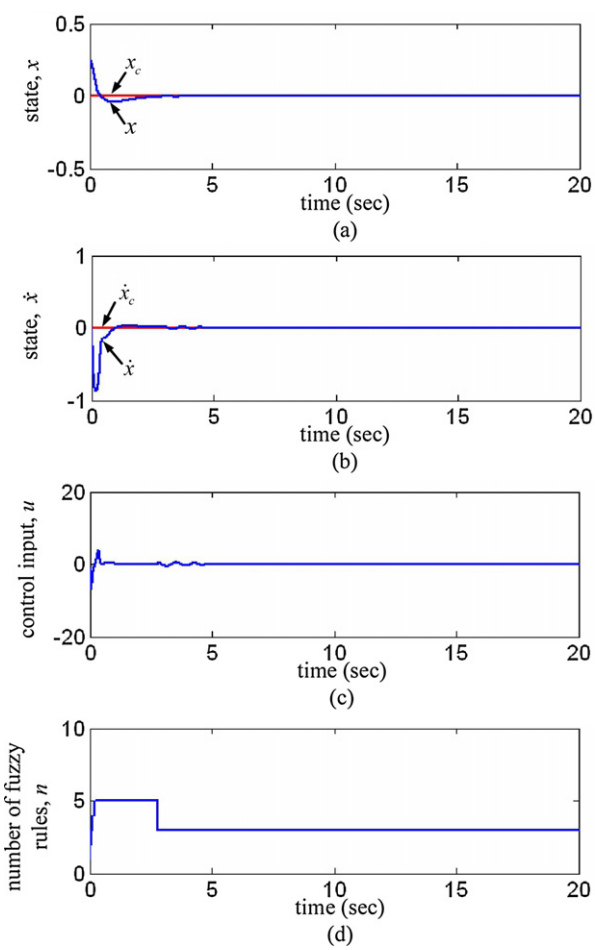

Fig. 7. Simulation results of the inverted pendulum system for Case 1.

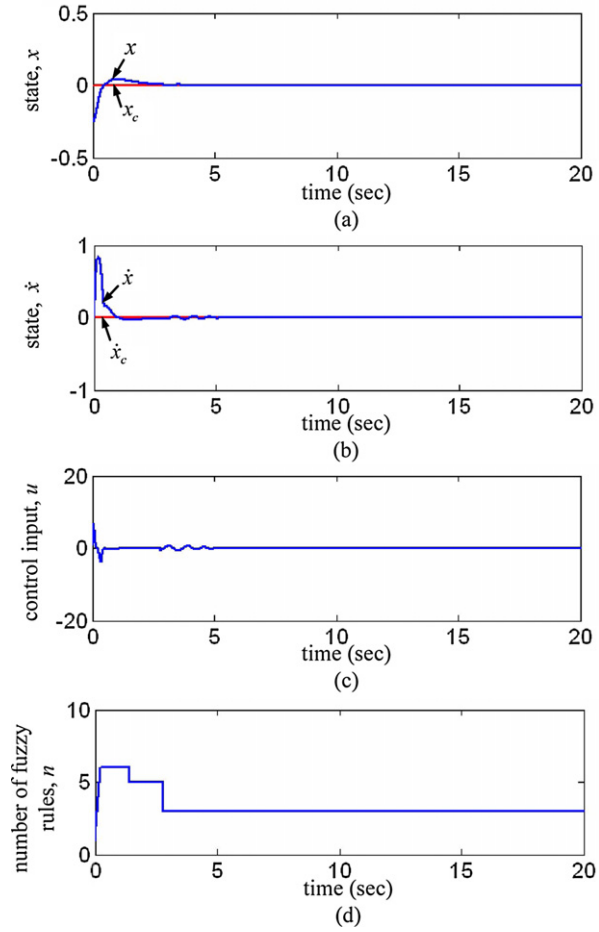

Fig. 8. Simulation results of the inverted pendulum system for Case 2.

5. Conclusions

A dynamic TSK-type RBF-based neural-fuzzy (DTRN) system is proposed in this article. In the proposed DTRN system, a new fuzzy rule will be generated if the incoming data does not belong to the existing fuzzy rules, and the existing fuzzy rules will be pruned if the used density is smaller than a pre-specified threshold. Then, a supervisory adaptive dynamic RBF-based neural-fuzzy control (SADRNC) system has been successfully developed. The proposed SADRNC system is composed of a DTRN controller and a supervisory compensator. The DTRN controller is used to approximate an

ideal controller and the supervisory compensator is designed to eliminate the effect of the approximation error introduced by the DTRN controller upon the system stability in the Lyapunov sense. Finally, the proposed SADRNC system is applied to control a chaotic system and an inverted pendulum. The simulation results verify that a favorable control performance can be achieved by the proposed SADRNC scheme after the controller parameters have been learned. The stability of the proposed SADRNC scheme has been shown and the effectiveness of the technique has been demonstrated through computer simulations.

Acknowledgments

The authors are grateful to the reviewers for their valuable comments. The authors appreciate the partial financial support from the National Science Council of Republic of China under the grant NSC 99-2628-E-216-002 and NSC 98-2221-E-155-058 -MY3.

References

[1] M. Bortman, M. Aladjem, A growing and pruning method for radial basis function networks, IEEE Transactions on Neural Networks 20 (6) (2009) 1039–1045.
[2] C.S. Chen, TSK-type self-organizing recurrent-neural-fuzzy control of linear microstepping motor drives, IEEE Transactions on Power Electronics 25 (9) (2010) 2253–2265.
[3] G. Chen, X. Dong, On feedback control of chaotic continuous time systems, IEEE Transactions on Circuits Systems I 40 (9) (1993) 591–601.
[4] K.H. Cheng, CMAC-based neuro-fuzzy approach for complex system modeling, Neurocomputing 72 (7) (2009) 1763–1774.
[5] C. Elmas, O. Ustun, H.H. Sayan, A neuro-fuzzy controller for speed control of a permanent magnet synchronous motor drive, Expert Systems with Applications 34 (1) (2008) 657–664.
[6] Y.Y. Fu, C.J. Wu, C.N. Ko, Radial basis function networks with hybrid learning for system identification with outliers, Applied Soft Computing 11 (3) (2011) 2278–2283.
[7] D. Hanbay, I. Turkoglu, Y. Demir, An expert system based on wavelet decomposition and neural network for modeling Chua's circuit, Expert Systems with Applications 34 (4) (2008) 2278–2283.
[8] S. Haykin, Neural Networks, a Comprehensive Foundation, Prentice-Hall, Upper Saddle River, NJ, 1999.
[9] C.F. Hsu, Self-organizing adaptive fuzzy neural control for a class of nonlinear systems, IEEE Transactions on Neural Networks 18 (4) (2007) 1232–1241.
[10] C.F. Hsu, Adaptive dynamic RBF neural controller design for a class of nonlinear systems, Applied Soft Computing 11 (8) (2011) 4607–4613.
[11] C.F. Hsu, Intelligent tracking control of a DC motor driver using self-organizing TSK type fuzzy neural networks, Nonlinear Dynamics 67 (1) (2012) 587–600.
[12] G.B. Huang, P. Saratchandran, N. Sundararajan, A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation, IEEE Transactions on Neural Networks 16 (1) (2005) 57–67.
[13] J.S.R. Jang, ANFIS: adaptive-network-based fuzzy inference system, IEEE Transactions on Systems, Man and Cybernetics 23 (3) (1993) 665–685.
[14] C.F. Juang, C. Lo, Zero-order TSK-type fuzzy system learning using a two-phase swarm intelligence, Fuzzy Sets and Systems 159 (21) (2008) 2910–2926.
[15] Y.G. Leu, W.Y. Wang, T.T. Lee, Observer-based direct adaptive fuzzy-neural control for nonaffine nonlinear systems, IEEE Transactions on Neural Networks 16 (4) (2005) 853–861.
[16] C.S. Leung, A.C. Tsoi, Combined learning and pruning for recurrent radial basis function networks based on recursive least square algorithms, Neural Computing and Applications 15 (1) (2005) 62–78.
[17] C.J. Lin, An efficient immune-based symbiotic particle swarm optimization learning algorithm for TSK-type neuro-fuzzy networks design, Fuzzy Sets and Systems 159 (21) (2008) 2890–2909.
[18] C.K. Lin, Radial basis function neural network-based adaptive critic control of induction motors? Applied Soft Computing 12 (6) (2011) 1640–1649.
[19] C.M. Lin, C.F. Hsu, Supervisory recurrent fuzzy neural network control of wing rock for slender delta wings, IEEE Transactions on Fuzzy Systems 12 (5) (2004) 733–742.
[20] C.T. Lin, C.S.G. Lee, Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems, Prentice-Hall, Upper Saddle River, NJ, 1996.
[21] F.J. Lin, P.H. Shen, P.H. Chou, S.L. Yang, TSK-type recurrent fuzzy network for dsp-based permanent-magnet linear synchronous motor servo drive? IEE Proceedings Electric Power Applications 153 (6) (2006) 921–931.
[22] F.J. Lin, L.T. Teng, M.H. Yu, Radial basis function network control with improved particle swarm optimization for induction generator system, IEEE Transactions on Power Electronics 23 (4) (2008) 2157–2169.
[23] S.N. Qasem, S.M. Shamsuddin, Radial basis function network based on time variant multi-objective particle swarm optimization for medical diseases diagnosis, Applied Soft Computing 11 (1) (2011) 1427–1438.
[24] J.E. Slotine, W. Li, Applied Nonlinear Control, Prentice-Hall, Englewood Cliffs, NJ, 1991.
[25] R.J. Wai, P.C. Chen, Intelligent tracking control for robot manipulator including actuator dynamics via TSK-type fuzzy neural network, IEEE Transactions on Fuzzy Systems 12 (4) (2004) 552–560.
[26] H.N. Wu, M.Z. Bai, Active fault-tolerant fuzzy control design of nonlinear model tracking with application to chaotic systems, IET Control Theory and Applications 3 (6) (2009) 642–653.
[27] S.J. Wu, C.T. Wu, Y.C. Chang, Neural-fuzzy gap control for a current/voltage-controlled 1/4-vehicle maglev system, IEEE Transactions on Intelligent Transportation Systems 9 (1) (2008) 122–136.