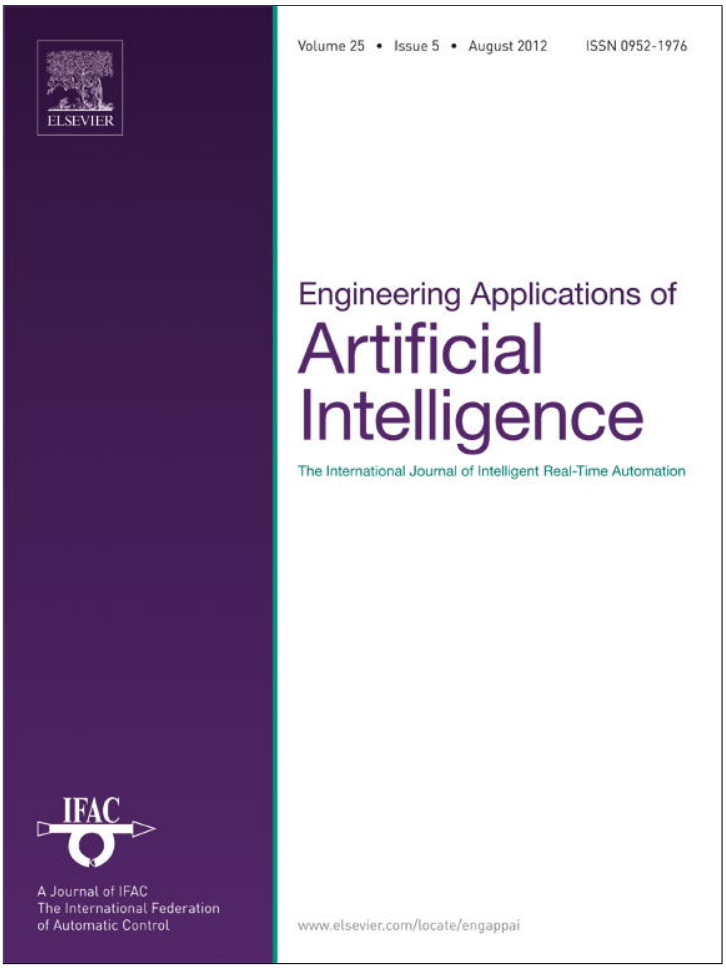

Engineering Applications of Artificial Intelligence 25 (2012) 997–1008

Contents lists available at SciVerse ScienceDirect

Engineering Applications of Artificial Intelligence

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

Adaptive dynamic CMAC neural control of nonlinear chaotic systems with L_2 tracking performance

Chun-Fei Hsu

Department of Electrical Engineering, Tamkang University, No. 151, Yingzhuan Rd., Tamsui Dist., New Taipei City 25137, Taiwan, ROC

ARTICLE INFO

Article history:
Received 27 September 2010
Received in revised form
7 January 2012
Accepted 23 March 2012
Available online 11 April 2012

Keywords:
Adaptive control
Neural control
Sliding-mode control
Chaotic system

ABSTRACT

The advantage of using cerebellar model articulation control (CMAC) network has been well documented in many applications. However, the structure of a CMAC network which will influence the learning performance is difficult to select. This paper proposes a dynamic structure CMAC network (DSCN) which the network structure can grow or prune systematically and their parameters can be adjusted automatically. Then, an adaptive dynamic CMAC neural control (ADCNC) system which is composed of a computation controller and a robust compensator is proposed via second-order sliding-mode approach. The computation controller containing a DSCN identifier is the principal controller and the robust compensator is designed to achieve L_2 tracking performance with a desired attenuation level. Moreover, a proportional–integral (PI)-type adaptation learning algorithm is derived to speed up the convergence of the tracking error in the sense of Lyapunov function and Barbalat's lemma, thus the system stability can be guaranteed. Finally, the proposed ADCNC system is applied to control a chaotic system. The simulation results are demonstrated that the proposed ADCNC scheme can achieve a favorable control performance even under the variations of system parameters and initial point.

1. Introduction

It is well known that the major advantage of a sliding-mode control (SMC) system is its insensitivity to parameter variations and external disturbance once the system trajectory reaches and stays on a sliding surface (Slotine and Li, 1991; Utkin, 1978). However, the SMC strategy produces a drawback associated with a large control chattering. It may wear coupled mechanisms and excite unstable system dynamics. To tackle this problem, a second-order sliding-mode control (SSC) system which is an effective control scheme for chattering eliminating is proposed (Koshkouei et al., 2005; Levant, 1993; Levant et al., 2000; Lin et al., 2009; Parra-Vega et al., 2003). The additional dynamics can be considered as compensators that are designed for improving sliding-mode stability. Due to the SSC approach using integration method to obtain a practical control input, the chattering phenomenon can be improved effectively. However, since the precise models of control plants are difficult to obtain, both of the SMC and SSC systems are difficult to be implemented in real-time applications.

During the past two decades, several adaptive neural controllers have been developed to compensate for the effects of nonlinearities and system uncertainties (Chen and Tian, 2009; Czarnigowski, 2010; Hsu, 2011; Miguel and Yu, 2009; Zhao and Yu, 2009). The basic issue of the adaptive neural control provides online learning algorithm that does not require preliminary off-line tuning. Some are based on the Lyapunov stability theorem and some are based on the gradient decent method; thus the stability, convergence and robustness of the closed-loop system can be guaranteed. Recently, a cerebellar model articulation control (CMAC) network was widely used due to its fast learning property and good generalization capability (Ananthraman and Gar, 1993; Hsu et al., 2009; Lin et al., 2007; Lin and Peng, 2005; Wu et al., 2011; Wu et al., 2006; Yeh, 2007). Though the control performances in the above CMAC network literatures are acceptable, the network structure of a CMAC network cannot be obtained automatically. If the amount of memory space is chosen too large, the computation loading is heavy so it is not suitable for real-time practical applications. If the amount of memory space is chosen too small, the learning performance may be not good enough to achieve a desired performance.

To tackle this problem, a dynamic structure CMAC network (DSCN) is proposed for the structure adaptation of a CMAC network (Lee et al., 2007; Lin and Chen, 2009; Yeh and Chang, 2006; Yen et al., 2012). However, some cannot avoid the CMAC network structure growing unboundedly and some requires overly complex design procedures. In this paper, a novel DSCN with online adjusting suitable memory space of a CMAC network structure is studied. The proposed self-constructing approach

E-mail address: fei@ee.tku.edu.tw

0952-1976/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.engappai.2012.03.014

demonstrates the properties of generating and pruning the input layers automatically. Then, the proposed DSCN is utilized to estimate the change of the system dynamics online owing to its good generalization capability, structure adaptation and simple computation.

In this paper, an adaptive dynamic CMAC neural control (ADCNC) system which is composed of a computation controller and a robust compensator is developed. The computation controller containing a DSCN identifier is the main controller and the robust compensator is designed to achieve L_2 tracking performance by attenuating the effect of the uncertain term caused by the DSCN identifier. The DSCN identifier does not require prior knowledge of a certain amount of memory space and the self-constructing approach demonstrates the properties of generating or pruning the layers automatically. The adaptive laws of the proposed ADCNC system are derived in the sense of Lyapunov function and Barbalat's lemma; thus the system can be guaranteed to be stable. It should be emphasized that the proposed ADCNC system requires no prior knowledge about the system dynamic and no offline learning phases. Finally, a chaotic system is provided as a simulation example. The simulation results verify that the system stabilization, favorable tracking performance and no chattering phenomena can be achieved by the proposed ADCNC system.

This paper is organized as follows. Problem formulation is described in Section 2. Section 3 expresses the design of the proposed ADCNC system. Simulation results are provided to validate the effectiveness of the proposed ADCNC system in Section 4. Finally, Section 5 concludes the paper.

2. Problem formulation

Chaotic systems have been studied and known to exhibit complex dynamical behavior (Hsu et al., 2009). The interest in chaotic systems lies mostly upon their complex, unpredictable behavior, and extreme sensitivity to initial conditions as well as parameter variations. Consider a second-order chaotic system, the well known Duffing's equation, which describes a special nonlinear circuit or a pendulum moving in a viscous medium under control. The dynamics of Duffing's equation is described as Hsu et al. (2009), Jiang (2002), Peng (2009)

$$\ddot{x}(t) = -p\dot{x}(t) - p_1 x(t) - p_2 x^3(t) + q\cos(wt) + u(t) = f(\mathbf{x},t) + u(t) \quad (1)$$

where t is the time variable, $\mathbf{x} = [x(t), \dot{x}(t)]^{\mathrm{T}}$ is the state vector, w is the frequency, $f(\mathbf{x},t) = -p\dot{x} - p_1 x - p_2 x^3 + q\cos(wt)$ is the system dynamic, $u(t)$ is the control input, and p, p_1, p_2 and q are real constants. Depending on the choices of these constants, the solutions of system (1) may display complex phenomena, including various periodic orbits behaviors and some chaotic behaviors (Hsu et al., 2009). To observe the complex phenomena, the open-loop system behavior for $u(t)=0$ was simulated as shown in Fig. 1 with $p=0.4$, $p_1=-1.1$, $p_2=1.0$ and $w=1.8$. 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. 1(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. Further, the time responses of the uncontrolled chaotic system with initial point (1,1) for $q=2.1$ and $q=7.0$ are shown in Fig. 1(c) and (d), respectively. An uncontrolled chaotic trajectory can be found in Fig. 1(c) and a period motion chaotic trajectory can be found in Fig. 1(d). In summary, the uncontrolled chaotic system sometimes has a chaotic trajectory but sometimes it does not. The uncontrolled chaotic system has extreme sensitivity to parameters change and initial points.

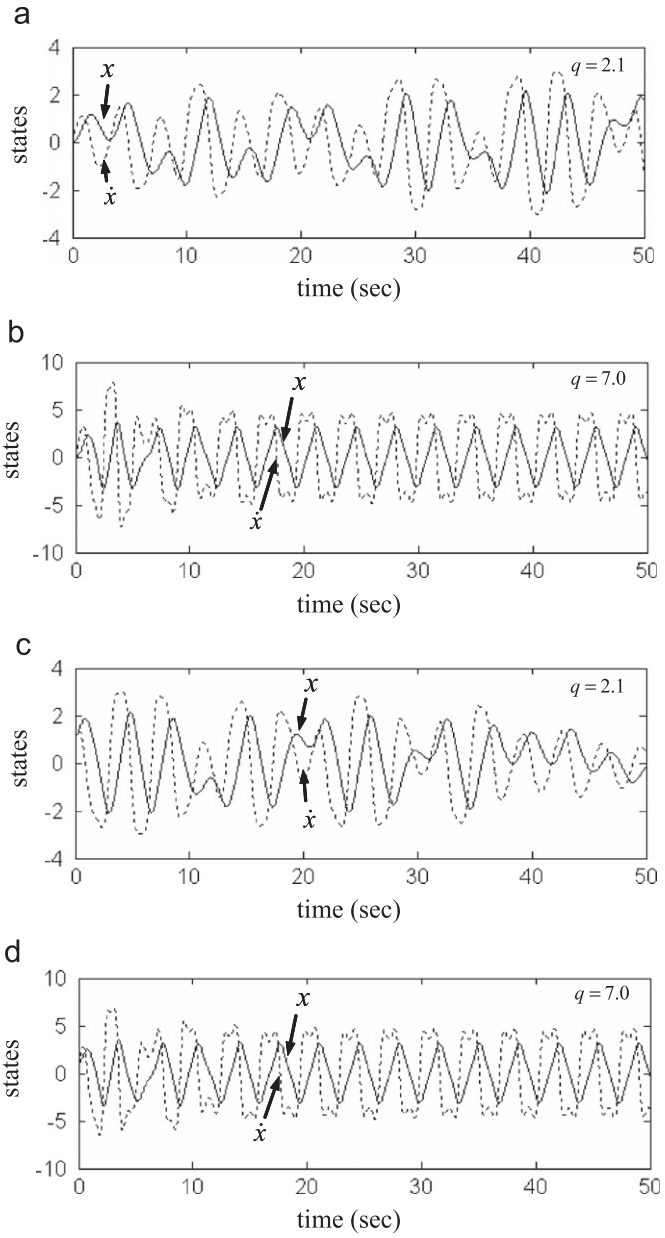

Fig. 1. Uncontrolled chaotic system.

Rewriting (1), the nominal model of the chaotic system can be represented as follows

$$\ddot{x}(t) = f_n(\mathbf{x},t) + u(t) \quad (2)$$

where $f_n(\mathbf{x},t)$ is a mapping that represents the nominal behavior of $f(\mathbf{x},t)$. If uncertainties occur, i.e., the parameters of the system deviate from the nominal value and a external disturbance is added into the system, the chaotic system can be modified as

$$\ddot{x}(t) = f_n(\mathbf{x},t) + \Delta f(\mathbf{x},t) + u(t) + d(t) = f_n(\mathbf{x},t) + u(t) + z(\mathbf{x},t) \quad (3)$$

where $\Delta f(\mathbf{x},t)$ denotes the system uncertainties, $d(t)$ is the external disturbance, and $z(\mathbf{x},t)$ is called the lumped uncertainty which is defined as $z(\mathbf{x},t) = \Delta f(\mathbf{x},t) + d(t)$. The control objective is to find a control law so that the state trajectory $x(t)$ can track a trajectory command $x_c(t)$ closely. Define a tracking error and a sliding

surface as Lin et al. (2009)

$$e(t)=x(t)-x_c(t) \tag{4}$$

$$s(t)=\dot{e}(t)+a_1e(t)+a_2\int_0^t e(\tau)\mathrm{d}\tau \tag{5}$$

where a_1 and a_2 are positive constants. Properly choosing the values of a_1 and a_2, the desired system dynamics such as rise time, overshoot, and settling time can be easily designed by the second-order system. Differentiating (5) with respect to time and using (3) and (4) obtain

$$\begin{aligned}\dot{s}(t)&=\ddot{e}(t)+a_1\dot{e}(t)+a_2e(t)\\&=f_n(\mathbf{x},t)+u(t)+z(\mathbf{x},t)-\ddot{x}_c(t)+a_1\dot{e}(t)+a_2e(t)\\&=c(\mathbf{x},t)+u(t)+z(\mathbf{x},t)+a_1\dot{e}(t)+a_2e(t)\end{aligned} \tag{6}$$

where $c(\mathbf{x},t)=f_n(\mathbf{x},t)-\ddot{x}_c(t)$. In order to reduce the chattering phenomenon, a SSC system is considered with a second-order sliding surface defined as Lin et al. (2009)

$$\sigma(t)=\dot{s}(t)+b_1s(t)+b_2\int_0^t s(\tau)\mathrm{d}\tau \tag{7}$$

where b_1 and b_2 are positive constants. Differentiating (7) with respect to time and using (3) and (6) obtain

$$\begin{aligned}\dot{\sigma}(t)&=\ddot{s}(t)+b_1\dot{s}(t)+b_2s(t)\\&=\dot{c}(\mathbf{x},t)+\dot{u}(t)+\dot{z}(\mathbf{x},t)+a_1\ddot{e}(t)+a_2\dot{e}(t)\\&\quad+b_1[\ddot{e}(t)+a_1\dot{e}(t)+a_2e(t)]\\&\quad+b_2[\dot{e}(t)+a_1e(t)+a_2\int_0^t e(\tau)\mathrm{d}\tau]\\&=\dot{c}(\mathbf{x},t)+\dot{u}(t)+\dot{z}(\mathbf{x},t)+(a_1+b_1)\ddot{e}(t)+(a_2+a_1b_1+b_2)\dot{e}(t)\\&\quad+(a_2b_1+a_1b_2)e(t)+a_2b_2\int_0^t e(\tau)\mathrm{d}\tau\\&=\dot{c}(\mathbf{x},t)+\dot{u}(t)+\dot{z}(\mathbf{x},t)+c_1\ddot{e}(t)+c_2\dot{e}(t)+c_3e(t)+c_4\int_0^t e(\tau)\mathrm{d}\tau\end{aligned} \tag{8}$$

where $c_1=a_1+b_1$, $c_2=a_2+a_1b_1+b_2$, $c_3=a_2b_1+a_1b_2$ and $c_4=a_2b_2$. The control law of the SSC system is given as Koshkouei et al. (2005), Lin et al. (2009)

$$u_{\mathrm{ssc}}(t)=\int_0^t \dot{u}_{\mathrm{ssc}}(\tau)\mathrm{d}\tau \tag{9}$$

$$\dot{u}_{\mathrm{ssc}}(t)=-\dot{c}(\mathbf{x},t)-c_1\ddot{e}(t)-c_2\dot{e}(t)-c_3e(t)-c_4\int_0^t e(\tau)\mathrm{d}\tau-Z_{\mathrm{ssc}}\mathrm{sgn}[\sigma(t)] \tag{10}$$

where Z_{ssc} is a given positive constant with the assumption $|\dot{z}(\mathbf{x},t)|\le Z_{\mathrm{ssc}}$. Imposing the control law $\dot{u}(t)=\dot{u}_{\mathrm{ssc}}(t)$ in (8) with (10) yields

$$\dot{\sigma}(t)=\dot{z}(\mathbf{x},t)-Z_{\mathrm{ssc}}\mathrm{sgn}[\sigma(t)]. \tag{11}$$

Consider the candidate Lyapunov function in the following form as

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

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

$$\begin{aligned}\dot{V}_1(t)&=\sigma(t)\,\dot{\sigma}(t)=\dot{z}(\mathbf{x},t)\sigma(t)-Z_{\mathrm{ssc}}|\sigma(t)|\\&\le|\dot{z}(\mathbf{x},t)|\,|\sigma(t)|-Z_{\mathrm{ssc}}|\sigma(t)|\\&=-(Z_{\mathrm{ssc}}-|\dot{z}(\mathbf{x},t)|)|\sigma(t)|\le 0.\end{aligned} \tag{13}$$

If the second-order sliding surface $\dot{\sigma}(t)=\ddot{s}(t)+b_1\dot{s}(t)+b_2s(t)=0$, then the sliding surface $s(t)=0$ for all time. Moreover, if $\dot{s}(t)=\ddot{e}(t)+a_1\dot{e}(t)+a_2e(t)=0$, the desired system dynamics such as rise time, overshoot and setting time can easily designed by a second-order system. The control law of the SSC system in (10) can guarantee the system stability in the sense of the Lyapunov theorem (Levant et al., 2000; Slotine and Li, 1991). However, in the most practical systems, the system dynamics $c(\mathbf{x},t)$ may be unknown or the bound of the lumped uncertainty Z_{ssc} is difficult to obtain, so the SSC system cannot be implemented in real-time applications.

3. Design of the ADCNC system

The proposed ADCNC system is composed of a computation controller and a robust compensator. The computation controller containing a DSCN identifier is the principal controller and the robust compensator is designed to achieve L_2 tracking performance with a desired attenuation level.

3.1. Network structure of DSCN

The architecture of DSCN is composed of input space, association memory space, receptive-field space, weight memory space and output space. The Gaussian function in association memory space is represented as Lin and Chen (2009)

$$\phi_{ij}(z_i,m_{ij},v_{ij})=\exp[-v_{ij}^2(z_i-m_{ij})^2],\quad \text{for } i=1,2 \tag{14}$$

where $\mathbf{z}=[z_1,z_2]^{\mathrm{T}}=[\sigma(t),\dot{\sigma}(t)]^{\mathrm{T}}$ is the input vector and $\phi_{ij}(z_i,m_{ij},v_{ij})$ presents the jth block of the ith input with two adjustable parameters m_{ij} and v_{ij}. The m_{ij} is the argument on which the Gaussian function has its maximum value, i.e., $\phi_{ij}(z_i,m_{ij},v_{ij})=1$ and the v_{ij} influences the Gaussian function profile. Assume that $N(k)$ is the number of the existing receptive-field functions at the kth sample interval, the multidimensional receptive-field function is defined as

$$\Theta_j(\mathbf{z},\mathbf{m}_j,\mathbf{v}_j)=\prod_{i=1}^{2}\phi_{ij}(z_i,m_{ij},v_{ij}),\quad \text{for } j=1,2,\ldots,N(k) \tag{15}$$

where Θ_j is associated with the jth receptive-field function, $\mathbf{m}_j=[m_{1j},m_{2j}]^{\mathrm{T}}$, and $\mathbf{v}_j=[v_{1j},v_{2j}]^{\mathrm{T}}$. The output of DSCN at the kth sample interval is given by

$$y=\sum_{j=1}^{N(k)}\alpha_j\Theta_j(\mathbf{z},\mathbf{m}_j,\mathbf{v}_j) \tag{16}$$

where α_j is the connecting weight value of the output associated with the jth receptive-field function and it is initialized from zero and is automatically adjusted during online operation. Then, the output represents in a vector form as

$$y=\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\Theta}(\mathbf{z},\mathbf{m},\mathbf{v}) \tag{17}$$

where $\boldsymbol{\alpha}=[\alpha_1,\alpha_2,\ldots,\alpha_{N(k)}]^{\mathrm{T}}$ and $\boldsymbol{\Theta}(\mathbf{z},\mathbf{m},\mathbf{v})=[\Theta_1,\Theta_2,\ldots,\Theta_{N(k)}]^{\mathrm{T}}$ with $\mathbf{m}=[\mathbf{m}_1^{\mathrm{T}},\mathbf{m}_2^{\mathrm{T}},\ldots,\mathbf{m}_{N(k)}^{\mathrm{T}}]^{\mathrm{T}}$, $\mathbf{v}=[\mathbf{v}_1^{\mathrm{T}},\mathbf{v}_2^{\mathrm{T}},\ldots,\mathbf{v}_{N(k)}^{\mathrm{T}}]^{\mathrm{T}}$.

A trade-off problem between the computation loading and learning performance arises. To tackle this problem, this paper proposes a structure learning algorithm including how to grow and prune the association memory space of DSCN is introduced. The first process of the structure learning is to determine whether to add a new Gaussian function in association memory and to create its hypercube and weight memory, simultaneously. In the generating process, the mathematical description of the existing Gaussian function can be expressed as clusters (Lin and Lee, 1996). The firing strength of a rule for each incoming data $\mathbf{z}$ can be represented as the degree to which the incoming data belong to the cluster. If a new input data falls $\mathbf{z}$ within the boundary of clusters, DSCN will not generate a new Gaussian function but update parameters of the existing rules. The distance between incoming data $\mathbf{z}$ and $\mathbf{m}_j$ as Lin and Chen (2009)

$$d_j=\|\mathbf{z}-\mathbf{m}_j\|. \tag{18}$$

Find a minimum distance which is defined as

$$d_{\min} = \min_{1 \le j \le N(k)} d_j. \tag{19}$$

If the distance is too large for the existing clusters, this means a new cluster should be generated a new input data. It implies that if $d_{\min} \ge d_{\text{th}}$ is satisfied, where dth a pre-given threshold, a new Gaussian function should be generated. Then, the number of receptive-field functions is increased at the next sample interval as follows

$$N(k+1) = N(k)+1. \tag{20}$$

The initial parameters of the new Gaussian function will be defined as

$$\alpha_{N(k+1)} = 0 \tag{21}$$

$$m_{iN(k+1)} = z_i \tag{22}$$

$$v_{iN(k+1)} = \overline{v} \tag{23}$$

where z_i is the input data and $\overline{v}$ is a pre-specified constant. To avoid the endless growing of DSCN structure and the overload computation loading, another self-constructing algorithm is considered to determine whether to delete the existing association memory space but is inappropriate. When the jth firing strength Θ_j is smaller than a elimination threshold ρ, it means that the relationship becomes weak between the input and the jth Gaussian function. This association memory may be less or never used. Then, it will gradually reduce the value of the jth significance index. A significance index for the importance of the jth receptive-field function at the kth sample interval is determined as Hsu (2007)

$$I_j(k) = \begin{cases} I_k(k-1)\exp(-\tau), & \text{if } \Theta_k < \rho \\ I_k(k-1), & \text{if } \Theta_k \ge \rho \end{cases} \tag{24}$$

where $I_j(k)$ is the significance index of the jth receptive-field basis function whose initial value is 1 and τ is the elimination speed constant. If $I_j(k) \le I$th is satisfied, where Ith a pre-given threshold, then the jth receptive-field basis function will be deleted. The computation loading should be decreased. In summary, the flow chart of the structure learning algorithm is shown in Fig. 2. The major contribution is DSCN can be operated directly without spending much time on pre-determining the receptive-field basis function.

3.2. Approximation of DSCN

Since the system dynamics $f(\mathbf{x},t)$ may be unknown in the most practical systems, the control law of the SSC system in (10) cannot be implemented. In this paper, a DSCN is utilized to estimate the change of the control system dynamics as Lin and Chen (2009)

$$\dot{f}(\mathbf{x},t) = \boldsymbol{\alpha}^{*\mathrm{T}}\boldsymbol{\Theta}(\mathbf{z},\mathbf{m}^*,\mathbf{v}^*) + \Delta = \boldsymbol{\alpha}^{*\mathrm{T}}\boldsymbol{\Theta}^* + \Delta \tag{25}$$

where Δ is the approximation error, $\boldsymbol{\alpha}^*$ and $\boldsymbol{\Theta}^*$ are the optimal parameter vectors of $\boldsymbol{\alpha}$ and $\boldsymbol{\Theta}$, respectively, and $\mathbf{m}^*$ and $\mathbf{v}^*$ are the optimal parameter vectors of $\mathbf{m}$ and $\mathbf{v}$, respectively. There exists Δ^* which is a finite positive constant such the inequality $|\Delta| \le \Delta^*$ can be held. Moreover, the optimal vectors can be further defined as Wang (1994)

$$(\boldsymbol{\alpha}^*,\mathbf{m}^*,\mathbf{v}^*) = \underset{\boldsymbol{\alpha}\in\boldsymbol{\Omega}_{\boldsymbol{\alpha}},\ \mathbf{m}\in\boldsymbol{\Omega}_{\mathbf{m}},\ \mathbf{v}\in\boldsymbol{\Omega}_{\mathbf{v}}}{\operatorname{argmin}} \left[\sup_{\mathbf{z}\in\boldsymbol{\Omega}_{\mathbf{z}}} |\dot{f} - \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\Theta}(\mathbf{z},\mathbf{m},\mathbf{v})|\right] \tag{26}$$

where $\boldsymbol{\Omega}_{\boldsymbol{\alpha}} = \{\boldsymbol{\alpha} : \|\boldsymbol{\alpha}\| \le D_{\boldsymbol{\alpha}}\}$, $\boldsymbol{\Omega}_{\mathbf{m}} = \{\mathbf{m} : \|\mathbf{m}\| \le D_{\mathbf{m}}\}$, and $\boldsymbol{\Omega}_{\mathbf{v}} = \{\mathbf{v} : \|\mathbf{v}\| \le D_{\mathbf{v}}\}$, respectively. The $D_{\boldsymbol{\alpha}}$, $D_{\mathbf{m}}$, and $D_{\mathbf{v}}$ are positive constants specified by designers. In fact, the optimal parameter vectors that are needed to best approximation cannot be determined. An estimation DSCN identifier is defined as

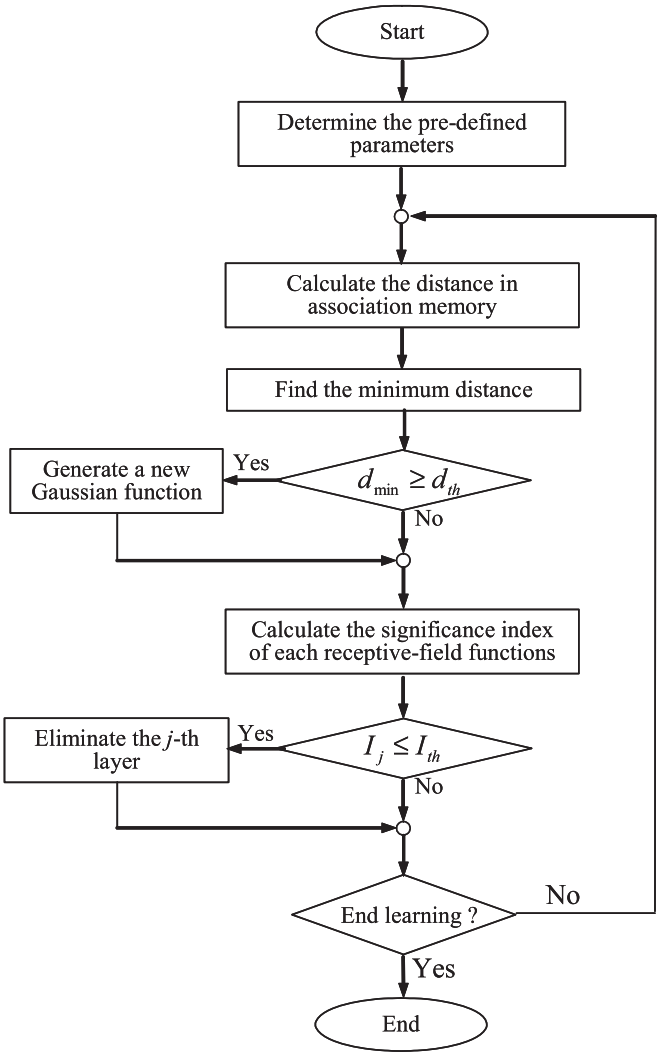

Fig. 2. Flow chart of the structure learning scheme.

$$\hat{\dot{f}} = \hat{\boldsymbol{\alpha}}^{\mathrm{T}}\boldsymbol{\Theta}(\mathbf{z},\hat{\mathbf{m}},\hat{\mathbf{v}}) = \hat{\boldsymbol{\alpha}}^{\mathrm{T}}\hat{\boldsymbol{\Theta}} \tag{27}$$

where $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\Theta}}$ are the estimated parameter vectors of $\boldsymbol{\alpha}$ and $\boldsymbol{\Theta}$, respectively; and $\hat{\mathbf{m}}$ and $\hat{\mathbf{v}}$ are the estimated parameter vectors of $\mathbf{m}$ and $\mathbf{v}$, respectively. Then, the estimation error is obtained as

$$\tilde{\dot{f}} = \dot{f} - \hat{\dot{f}} = \tilde{\boldsymbol{\alpha}}^{\mathrm{T}}\tilde{\boldsymbol{\Theta}} + \tilde{\boldsymbol{\alpha}}^{\mathrm{T}}\hat{\boldsymbol{\Theta}} + \hat{\boldsymbol{\alpha}}^{\mathrm{T}}\tilde{\boldsymbol{\Theta}} + \Delta \tag{28}$$

where $\tilde{\boldsymbol{\alpha}} = \boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}}$ and $\tilde{\boldsymbol{\Theta}} = \boldsymbol{\Theta}^* - \hat{\boldsymbol{\Theta}}$. To speed up the convergence of SOCM, the optimal parameter vector $\boldsymbol{\alpha}^*$ is decomposed into two parts as Golea et al. (2002), Hsu et al. (2009)

$$\boldsymbol{\alpha}^* = \eta_P\boldsymbol{\alpha}_{\mathbf{P}}^* + \eta_I\boldsymbol{\alpha}_{\mathbf{I}}^* \tag{29}$$

where $\boldsymbol{\alpha}_{\mathbf{P}}^*$ and $\boldsymbol{\alpha}_{\mathbf{I}}^*$ are the proportional and integral terms of $\boldsymbol{\alpha}^*$, respectively, η_P and η_I are positive coefficients, and $\boldsymbol{\alpha}_{\mathbf{I}}^* = \int_0^t \boldsymbol{\alpha}_{\mathbf{P}}^* \mathrm{d}\tau$. The proportional term of the optimal parameter vector satisfies $\|\boldsymbol{\alpha}_{\mathbf{P}}^*\| \le D_{\boldsymbol{\alpha}_{\mathbf{P}}}$, where $D_{\boldsymbol{\alpha}_{\mathbf{P}}}$ is a positive constant specified by designers. Similarly, the estimation parameter vector $\hat{\boldsymbol{\alpha}}$ is decomposed into two parts as Golea et al. (2002), Hsu et al. (2009)

$$\hat{\boldsymbol{\alpha}} = \eta_P\hat{\boldsymbol{\alpha}}_{\mathbf{P}} + \eta_I\hat{\boldsymbol{\alpha}}_{\mathbf{I}} \tag{30}$$

where $\hat{\boldsymbol{\alpha}}_{\mathbf{P}}$ and $\hat{\boldsymbol{\alpha}}_{\mathbf{I}}$ are the proportional and integral terms of $\hat{\boldsymbol{\alpha}}$, respectively, and $\hat{\boldsymbol{\alpha}}_{\mathbf{I}} = \int_0^t \hat{\boldsymbol{\alpha}}_{\mathbf{P}}\, \mathrm{d}\tau$. The integral term of the estimation parameter vector satisfies $\|\hat{\boldsymbol{\alpha}}_{\mathbf{I}}\| \le D_{\boldsymbol{\alpha}_{\mathbf{I}}}$, where $D_{\boldsymbol{\alpha}_{\mathbf{I}}}$ is a positive

constant specified by designers. Thus, $\tilde{\boldsymbol{\alpha}}$ can be expressed as

$$\tilde{\boldsymbol{\alpha}} = \eta_I \tilde{\boldsymbol{\alpha}}_{\mathbf{I}} - \eta_P \hat{\boldsymbol{\alpha}}_{\mathbf{P}} + \eta_P \boldsymbol{\alpha}_{\mathbf{P}}^* \tag{31}$$

where $\tilde{\boldsymbol{\alpha}}_{\mathbf{I}} = \boldsymbol{\alpha}_{\mathbf{I}}^* - \hat{\boldsymbol{\alpha}}_{\mathbf{I}}$. Substituting (31) into (28), it is obtained that

$$\begin{aligned}\dot{\tilde{f}} &= \tilde{\boldsymbol{\alpha}}^{\mathrm{T}}\tilde{\boldsymbol{\Theta}} + (\eta_I \tilde{\boldsymbol{\alpha}}_{\mathbf{I}} - \eta_P \hat{\boldsymbol{\alpha}}_{\mathbf{P}} + \eta_P \boldsymbol{\alpha}_{\mathbf{P}}^*)^{\mathrm{T}}\hat{\boldsymbol{\Theta}} + \hat{\boldsymbol{\alpha}}^{\mathrm{T}}\tilde{\boldsymbol{\Theta}} + \Delta \\ &= \eta_I \tilde{\boldsymbol{\alpha}}_{\mathbf{I}}^{\mathrm{T}}\hat{\boldsymbol{\Theta}} - \eta_P \hat{\boldsymbol{\alpha}}_{\mathbf{P}}^{\mathrm{T}}\hat{\boldsymbol{\Theta}} + \hat{\boldsymbol{\alpha}}^{\mathrm{T}}\tilde{\boldsymbol{\Theta}} + \tilde{\boldsymbol{\alpha}}^{\mathrm{T}}\tilde{\boldsymbol{\Theta}} + \eta_P \boldsymbol{\alpha}_{\mathbf{P}}^{*\mathrm{T}}\hat{\boldsymbol{\Theta}} + \Delta. \end{aligned} \tag{32}$$

Then, the Taylor expansion linearization technique is employed to transform the nonlinear function $\tilde{\boldsymbol{\Theta}}$ into a partially linear form (Peng and Lin, 2007), i.e.,

$$\tilde{\boldsymbol{\Theta}} = \mathbf{A}^{\mathrm{T}}\tilde{\mathbf{m}} + \mathbf{B}^{\mathrm{T}}\tilde{\mathbf{v}} + \mathbf{h} \tag{33}$$

where $\tilde{\mathbf{m}} = \mathbf{m}^* - \hat{\mathbf{m}}$, $\tilde{\mathbf{v}} = \mathbf{v}^* - \hat{\mathbf{v}}$, $\mathbf{h}$ is a vector of high order terms, $\mathbf{A} = [(\partial\Theta_1/\partial\mathbf{m})(\partial\Theta_2/\partial\mathbf{m})\cdots(\partial\Theta_N/\partial\mathbf{m})]\big|_{\mathbf{m}=\hat{\mathbf{m}}}$, and $\mathbf{B} = [(\partial\Theta_1/\partial\mathbf{v})(\partial\Theta_2/\partial\mathbf{v})\cdots(\partial\Theta_N/\partial\mathbf{v})]\big|_{\mathbf{v}=\hat{\mathbf{v}}}$. Substitute (33) into (32) yields

$$\begin{aligned}\dot{\tilde{f}} &= \eta_I \tilde{\boldsymbol{\alpha}}_{\mathbf{I}}^{\mathrm{T}}\hat{\boldsymbol{\Theta}} - \eta_P \hat{\boldsymbol{\alpha}}_{\mathbf{P}}^{\mathrm{T}}\hat{\boldsymbol{\Theta}} + \hat{\boldsymbol{\alpha}}^{\mathrm{T}}(\mathbf{A}^{\mathrm{T}}\tilde{\mathbf{m}} + \mathbf{B}^{\mathrm{T}}\tilde{\mathbf{v}} + \mathbf{h}) + \tilde{\boldsymbol{\alpha}}^{\mathrm{T}}\tilde{\boldsymbol{\Theta}} + \eta_P \boldsymbol{\alpha}_{\mathbf{P}}^{*\mathrm{T}}\hat{\boldsymbol{\Theta}} + \Delta \\ &= \eta_I \tilde{\boldsymbol{\alpha}}_{\mathbf{I}}^{\mathrm{T}}\hat{\boldsymbol{\Theta}} - \eta_P \hat{\boldsymbol{\alpha}}_{\mathbf{P}}^{\mathrm{T}}\hat{\boldsymbol{\Theta}} + \tilde{\mathbf{m}}^{\mathrm{T}}\mathbf{A}\hat{\boldsymbol{\alpha}} + \tilde{\mathbf{v}}^{\mathrm{T}}\mathbf{B}\hat{\boldsymbol{\alpha}} + \varepsilon \end{aligned} \tag{34}$$

where $\hat{\boldsymbol{\alpha}}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\tilde{\mathbf{m}} = \tilde{\mathbf{m}}^{\mathrm{T}}\mathbf{A}\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\alpha}}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\tilde{\mathbf{v}} = \tilde{\mathbf{v}}^{\mathrm{T}}\mathbf{B}\hat{\boldsymbol{\alpha}}$ are used since they are scalars, and $\varepsilon = \hat{\boldsymbol{\alpha}}^{\mathrm{T}}\mathbf{h} + \tilde{\boldsymbol{\alpha}}^{\mathrm{T}}\tilde{\boldsymbol{\Theta}} + \eta_P \boldsymbol{\alpha}_{\mathbf{P}}^{*\mathrm{T}}\hat{\boldsymbol{\Theta}} + \Delta$ denotes the uncertain term. It is an algebraic manipulation of the approximation via the Taylor expansion linearization technique. The higher-order term $\mathbf{h}$ is bounded by

$$\begin{aligned}\|\mathbf{h}\| &= \|\tilde{\boldsymbol{\Theta}} - \mathbf{A}^{\mathrm{T}}\tilde{\mathbf{m}} - \mathbf{B}^{\mathrm{T}}\tilde{\mathbf{v}}\| \\ &\le \|\tilde{\boldsymbol{\Theta}}\| + \|\mathbf{A}^{\mathrm{T}}\|\|\tilde{\mathbf{m}}\| + \|\mathbf{B}^{\mathrm{T}}\|\|\tilde{\mathbf{v}}\| \\ &\le \mathbf{c}_0 + \mathbf{c}_1\|\tilde{\mathbf{m}}\| + \mathbf{c}_2\|\tilde{\mathbf{v}}\| \end{aligned} \tag{35}$$

where c_0, c_1 and c_2 are bounded positive constants satisfying $\|\tilde{\boldsymbol{\Theta}}\| \le \mathbf{c}_0$, $\|\mathbf{A}^{\mathrm{T}}\| \le \mathbf{c}_1$, and $\|\mathbf{B}^{\mathrm{T}}\| \le \mathbf{c}_2$, respectively. The existence of c_0, c_1 and c_2 is assured due to the fact the Gaussian function and its derivative are always bounded by constants. Moreover, $\tilde{\boldsymbol{\alpha}}$, $\tilde{\mathbf{m}}$, and $\tilde{\mathbf{v}}$ satisfy

$$\|\tilde{\boldsymbol{\alpha}}\| = \|\boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}}\| \le \|\boldsymbol{\alpha}^*\| + \|\hat{\boldsymbol{\alpha}}\| \le D_{\boldsymbol{\alpha}} + \|\hat{\boldsymbol{\alpha}}\| \tag{36}$$

$$\|\tilde{\mathbf{m}}\| = \|\mathbf{m}^* - \hat{\mathbf{m}}\| \le \|\mathbf{m}^*\| + \|\hat{\mathbf{m}}\| \le D_{\mathbf{m}} + \|\hat{\mathbf{m}}\| \tag{37}$$

$$\|\tilde{\mathbf{v}} = \|\mathbf{v}^* - \hat{\mathbf{v}}\| \le \|\mathbf{v}^*\| + \|\hat{\mathbf{v}}\| \le D_{\mathbf{v}} + \|\hat{\mathbf{v}}\|. \tag{38}$$

Next, the uncertain term ε is bounded by

$$\begin{aligned}|\varepsilon| &= \|\hat{\boldsymbol{\alpha}}^{\mathbf{T}}\mathbf{h} + \tilde{\boldsymbol{\alpha}}^{\mathrm{T}}\tilde{\boldsymbol{\Theta}} + \eta_P \boldsymbol{\alpha}_{\mathbf{P}}^{*\mathrm{T}}\hat{\boldsymbol{\Theta}} + \Delta\| \\ &= \|\hat{\boldsymbol{\alpha}}^{\mathbf{T}}\mathbf{h} + \tilde{\boldsymbol{\alpha}}^{\mathrm{T}}(\mathbf{A}^{\mathrm{T}}\tilde{\mathbf{m}} + \mathbf{B}^{\mathrm{T}}\tilde{\mathbf{v}} + \mathbf{h}) + \eta_P \boldsymbol{\alpha}_{\mathbf{P}}^{*\mathrm{T}}\hat{\boldsymbol{\Theta}} + \Delta\| \\ &= \|\tilde{\boldsymbol{\alpha}}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\tilde{\mathbf{m}} + \tilde{\boldsymbol{\alpha}}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\tilde{\mathbf{v}} + \boldsymbol{\alpha}^{*T}\mathbf{h} + \eta_P \boldsymbol{\alpha}_{\mathbf{P}}^{*\mathrm{T}}\hat{\boldsymbol{\Theta}} + \Delta\| \\ &\le \mathbf{c}_1(D_{\boldsymbol{\alpha}} + \|\hat{\boldsymbol{\alpha}}\|)(D_{\mathbf{m}} + \|\hat{\mathbf{m}}\|) + \mathbf{c}_2(D_{\boldsymbol{\alpha}} + \|\hat{\boldsymbol{\alpha}}\|)(D_{\mathbf{v}} + \|\hat{\mathbf{v}}\|) \\ &\quad + D_{\boldsymbol{\alpha}}[\mathbf{c}_0 + \mathbf{c}_1(D_{\mathbf{m}} + \|\hat{\mathbf{m}}\|) + \mathbf{c}_2(D_{\mathbf{v}} + \|\hat{\mathbf{v}}\|)] + \eta_P\sqrt{N}D_{\boldsymbol{\alpha}_{\mathbf{P}}} + \Delta^* \\ &= [\xi_1, \xi_2, \xi_3, \xi_4, \xi_5, \xi_6][1, \|\hat{\boldsymbol{\alpha}}\|, \|\hat{\mathbf{m}}\|, \|\hat{\mathbf{v}}\|, \|\hat{\boldsymbol{\alpha}}\|\|\hat{\mathbf{m}}\|, \|\hat{\boldsymbol{\alpha}}\|\|\hat{\mathbf{v}}\|]^{\mathrm{T}} \\ &= \boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\Gamma} \end{aligned} \tag{39}$$

where $\boldsymbol{\xi} = [\xi_1, \xi_2, \xi_3, \xi_4, \xi_5, \xi_6]^{\mathrm{T}}$, $\boldsymbol{\Gamma} = [1, \|\hat{\boldsymbol{\alpha}}\|, \|\hat{\mathbf{m}}\|, \|\hat{\mathbf{v}}\|, \|\hat{\boldsymbol{\alpha}}\|\|\hat{\mathbf{m}}\|, \|\hat{\boldsymbol{\alpha}}\|\|\hat{\mathbf{v}}\|]^{\mathrm{T}}$, $\xi_1 = (\mathbf{c}_0 + 2\mathbf{c}_1 D_{\mathbf{m}} + 2\mathbf{c}_2 D_{\mathbf{v}})D_{\boldsymbol{\alpha}} + \eta_P\sqrt{N}D_{\boldsymbol{\alpha}_{\mathbf{P}}} + \Delta^*$, $\xi_2 = \mathbf{c}_1 D_{\mathbf{m}} + c_2 D_{\mathbf{v}}$, $\xi_3 = 2c_1 D_{\boldsymbol{\alpha}}$, $\xi_4 = 2c_2 D_{\boldsymbol{\alpha}}$, $\xi_5 = c_1$, and $\xi_6 = c_2$. Since $\boldsymbol{\xi}$ is a bounded vector, if $\boldsymbol{\Gamma}$ can be guaranteed to be bounded, the uncertain term ε is bounded.

3.3. ADCNC system design

The proposed ADCNC system is shown in Fig. 3, where the controller output is defined as

$$u_{\mathrm{adc}}(t) = \int_0^t \dot{u}_{\mathrm{adc}}(\tau)\mathrm{d}\tau \tag{40}$$

$$\dot{u}_{\mathrm{adc}} = \dot{u}_{\mathrm{cc}} + \dot{u}_{\mathrm{rc}} \tag{41}$$

where the computation controller is chosen as

$$\dot{u}_{\mathrm{cc}} = -\dot{\hat{f}} + \dddot{x}_c(t) - c_1\ddot{e}(t) - c_2\dot{e}(t) - c_3 e(t) - c_4\int_0^t e(\tau)\mathrm{d}\tau. \tag{42}$$

The computation controller $\dot{u}_{\mathrm{cc}}$ containing a DSCN identifier $\dot{\hat{f}}$ is the principal controller and the robust compensator $\dot{u}_{\mathrm{rc}}$ is designed to achieve L_2 tracking performance with a desired attenuation level. Differentiating (7) with respect to time and using (1) and (5) obtain

$$\begin{aligned}\dot{\sigma}(t) &= \dddot{s}(t) + b_1\ddot{s}(t) + b_2\dot{s}(t) \\ &= \dot{f}(\mathbf{x},t) + \ddot{u}(t) - \dddot{x}_c(t) + a_1\ddot{e}(t) + a_2\dot{e}(t) + b_1[\ddot{e}(t) + a_1\dot{e}(t) + a_2 e(t)] \\ &\quad + b_2[\dot{e}(t) + a_1 e(t) + a_2\int_0^i e(\tau)\mathrm{d}\tau] \\ &= \dot{f}(\mathbf{x},t) + \ddot{u}(t) - \dddot{x}_c(t) + (a_1 + b_1)\ddot{e}(t) + (a_2 + a_1 b_1 + b_2)\dot{e}(t) \end{aligned}$$

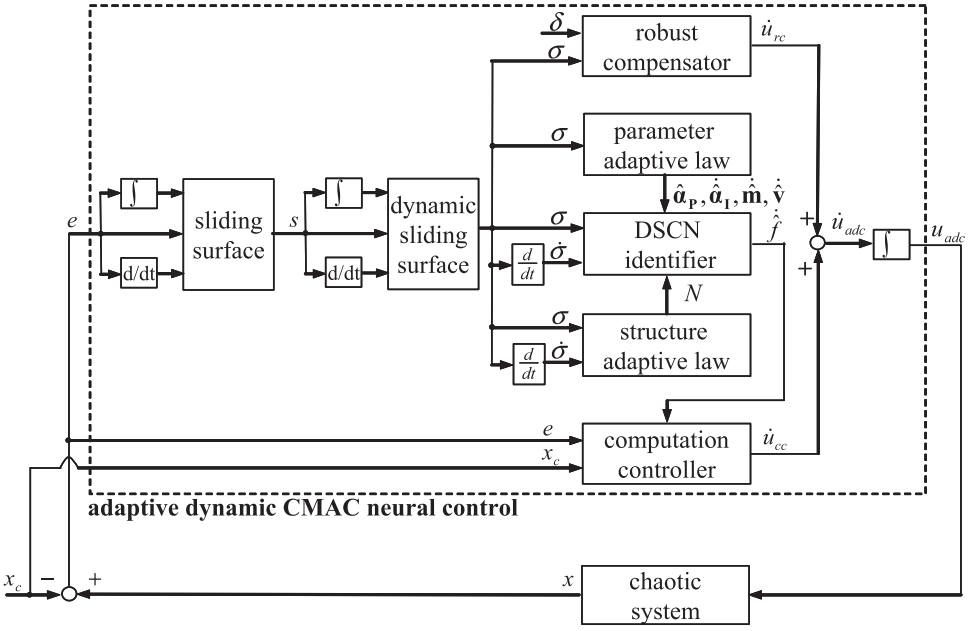

Fig. 3. Block diagram of the ADCNC system.

$$+(a_2b_1+a_1b_2)e(t)+a_2b_2\int_0^t e(\tau)\mathrm{d}\tau$$
$$=\dot{f}(\mathbf{x},t)+\dot{u}(t)-\dddot{x}_c(t)+c_1\ddot{e}(t)+c_2\dot{e}(t)+c_3e(t)+c_4\int_0^t e(\tau)\mathrm{d}\tau \tag{43}$$

where $c_1=a_1+b_1$, $c_2=a_2+a_1b_1+b_2$, $c_3=a_2b_1+a_1b_2$ and $c_4=a_2b_2$. Imposing the control law $\dot{u}(t)=\dot{u}_{\mathrm{adc}}(t)$ in (43) with (41) yields

$$\dot{\sigma}(t)=\dot{f}-\dot{\hat{f}}+\dot{u}_{\mathrm{rc}}. \tag{44}$$

By the approximation property (34), (44) can be rewritten as

$$\dot{\sigma}(t)=\eta_I\tilde{\boldsymbol{\alpha}}_{\mathbf{I}}^{\mathrm{T}}\hat{\boldsymbol{\Theta}}-\eta_P\hat{\boldsymbol{\alpha}}_{\mathbf{P}}^{\mathrm{T}}\hat{\boldsymbol{\Theta}}+\tilde{\mathbf{m}}^{\mathrm{T}}\mathbf{A}\hat{\boldsymbol{\alpha}}+\tilde{\mathbf{v}}^{\mathrm{T}}\mathbf{B}\hat{\boldsymbol{\alpha}}+\varepsilon+\dot{u}_{\mathrm{rc}}. \tag{45}$$

In case of the existence of ε, consider a specified L_2 tracking performance (Lin, 2009; Peng and Lin, 2007; Tseng, 2008; Wang et al., 2002)

$$\int_0^{\mathrm{T}}\sigma^2\mathrm{d}t\le\sigma^2(0)+\eta_I\tilde{\boldsymbol{\alpha}}_{\mathbf{I}}^{\mathrm{T}}(0)\tilde{\boldsymbol{\alpha}}_{\mathbf{I}}(0)+\frac{\tilde{\mathbf{m}}^{\mathrm{T}}(0)\tilde{\mathbf{m}}(0)}{\eta_m}+\frac{\tilde{\mathbf{v}}^{\mathrm{T}}(0)\tilde{\mathbf{v}}(0)}{\eta_v}+\delta^2\int_0^{\mathrm{T}}\varepsilon^2\mathrm{d}t \tag{46}$$

where η_I, η_m and η_v are positive learning rates and δ is a prescribed attenuation constant. If the system starts with initial conditions $\sigma(0)=0$, $\tilde{\boldsymbol{\alpha}}_{\mathbf{I}}(0)=0$, $\tilde{\mathbf{m}}(0)=0$ and $\tilde{\mathbf{v}}(0)=0$, the L_2 tracking performance in (46) can be rewritten as

$$\frac{\int_0^{\mathrm{T}}\sigma^2\mathrm{d}t}{\int_0^{\mathrm{T}}\varepsilon^2\mathrm{d}t}\le\delta^2. \tag{47}$$

The physical meaning of (47) is the effect of any uncertain term $\varepsilon(t)\in L_2$ on the second-order sliding surface $\sigma(t)$ must be attenuated below a desired level δ^2 from the viewpoint of energy. So the L_2 gain from $\varepsilon(t)$ to $\sigma(t)$ must be less than or equal to a prescribed value δ^2 (Peng and Lin, 2007; Tseng, 2008). If $\delta=\infty$, this is the case of minimum error tracking control without disturbance attenuation. To proof the stability of the ADCNC system, define a Lyapunov function candidate in the following form

$$V_2=\frac{1}{2}\sigma^2+\frac{\eta_I}{2}\tilde{\boldsymbol{\alpha}}_{\mathbf{I}}^{\mathrm{T}}\tilde{\boldsymbol{\alpha}}_{\mathbf{I}}+\frac{1}{2\eta_m}\tilde{\mathbf{m}}^{\mathrm{T}}\tilde{\mathbf{m}}+\frac{1}{2\eta_v}\tilde{\mathbf{v}}^{\mathrm{T}}\tilde{\mathbf{v}}. \tag{48}$$

Differentiating (48) with respect to time and using (45), it is obtained

$$\dot{V}_2=\sigma\dot{\sigma}+\eta_I\tilde{\boldsymbol{\alpha}}_{\mathbf{I}}^{\mathrm{T}}\dot{\tilde{\boldsymbol{\alpha}}}_{\mathbf{I}}+\frac{\tilde{\mathbf{m}}^{\mathrm{T}}\dot{\tilde{\mathbf{m}}}}{\eta_m}+\frac{\tilde{\mathbf{v}}^{\mathrm{T}}\dot{\tilde{\mathbf{v}}}}{\eta_v}$$
$$=\sigma(\eta_I\tilde{\boldsymbol{\alpha}}_{\mathbf{I}}^{\mathrm{T}}\hat{\boldsymbol{\Theta}}-\eta_P\hat{\boldsymbol{\alpha}}_{\mathbf{P}}^{\mathrm{T}}\hat{\boldsymbol{\Theta}}+\tilde{\mathbf{m}}^{\mathrm{T}}\mathbf{A}\hat{\boldsymbol{\alpha}}+\tilde{\mathbf{v}}^{\mathrm{T}}\mathbf{B}\hat{\boldsymbol{\alpha}}+\varepsilon+\dot{u}_{\mathrm{rc}})+\eta_I\tilde{\boldsymbol{\alpha}}_{\mathbf{I}}^{\mathrm{T}}\dot{\tilde{\boldsymbol{\alpha}}}_{\mathbf{I}}+\frac{\tilde{\mathbf{m}}^{\mathrm{T}}\dot{\tilde{\mathbf{m}}}}{\eta_m}+\frac{\tilde{\mathbf{v}}^{\mathrm{T}}\dot{\tilde{\mathbf{v}}}}{\eta_v}$$
$$=\eta_I\tilde{\boldsymbol{\alpha}}_{\mathbf{I}}^{\mathrm{T}}(\sigma\hat{\boldsymbol{\Theta}}+\dot{\tilde{\boldsymbol{\alpha}}}_{\mathbf{I}})+\tilde{\mathbf{m}}^{\mathrm{T}}(\sigma\mathbf{A}\hat{\boldsymbol{\alpha}}+\frac{\dot{\tilde{\mathbf{m}}}}{\eta_m})+\tilde{\mathbf{v}}^{\mathrm{T}}\left(\sigma\mathbf{B}\hat{\boldsymbol{\alpha}}+\frac{\dot{\tilde{\mathbf{v}}}}{\eta_v}\right)$$
$$-\sigma\eta_P\hat{\boldsymbol{\alpha}}_{\mathbf{P}}^{\mathrm{T}}\hat{\boldsymbol{\Theta}}+\sigma(\varepsilon+\dot{u}_{\mathrm{rc}}). \tag{49}$$

If the parameter adaptive laws are selected as

$$\hat{\boldsymbol{\alpha}}_{\mathbf{P}}=\sigma\hat{\boldsymbol{\Theta}} \tag{50}$$

$$\dot{\hat{\boldsymbol{\alpha}}}_{\mathbf{I}}=-\dot{\tilde{\boldsymbol{\alpha}}}_{\mathbf{I}}=\sigma\hat{\boldsymbol{\Theta}} \tag{51}$$

$$\dot{\hat{\mathbf{m}}}=-\dot{\tilde{\mathbf{m}}}=\eta_m\sigma\mathbf{A}\hat{\boldsymbol{\alpha}} \tag{52}$$

$$\dot{\hat{\mathbf{v}}}=-\dot{\tilde{\mathbf{v}}}=\eta_v\sigma\mathbf{B}\hat{\boldsymbol{\alpha}} \tag{53}$$

and design the robust compensator as

$$\dot{u}_{\mathrm{rc}}=-\frac{(\delta^2+1)\sigma}{2\delta^2} \tag{54}$$

then (49) becomes

$$\dot{V}_2=-\hat{\boldsymbol{\alpha}}_{\mathbf{P}}^{\mathrm{T}}\hat{\boldsymbol{\alpha}}_{\mathbf{P}}+\sigma\varepsilon-\frac{(\delta^2+1)\sigma^2}{2\delta^2}$$
$$\le-\frac{\sigma^2}{2}-\frac{1}{2}\left(\frac{\sigma}{\delta}-\delta\varepsilon\right)^2+\frac{1}{2}\delta^2\varepsilon^2$$
$$\le-\frac{\sigma^2}{2}+\frac{1}{2}\delta^2\varepsilon^2. \tag{55}$$

Integrating both sides of the above equation from $t=0$ to $t=T$, yields

$$V_2(T)-V_2(0)\le-\frac{1}{2}\int_0^{\mathrm{T}}\sigma^2\mathrm{d}t+\frac{1}{2}\delta^2\int_0^{\mathrm{T}}\varepsilon^2\mathrm{d}t \tag{56}$$

Since $V_2(T)\ge0$, the above inequality implies the following inequality

$$\frac{1}{2}\int_0^{\mathrm{T}}\sigma^2\mathrm{d}t\le V_2(0)+\frac{1}{2}\delta^2\int_0^{\mathrm{T}}\varepsilon^2\mathrm{d}t \tag{57}$$

Using (48), this inequality is equivalent to inequality (46). Thus, the L_2 tracking performance is achieved. Since $V_2(0)$ is finite and the uncertain term $\varepsilon\in L_2$, using the Barbalat's lemma, it implies that $\lim_{t\to\infty}\sigma=0$ (Slotine and Li, 1991).

3.4. Boundary analysis using projection algorithm

Though the stability of the ADCNC system can be guaranteed, the parameters $\hat{\boldsymbol{\alpha}}$, $\hat{\mathbf{m}}$, and $\hat{\mathbf{v}}$ cannot be guaranteed within a desired bound value by using the adaptive laws (51)–(53). Specifying the bounds $D_{\boldsymbol{\alpha}_{\mathbf{I}}}$, $D_{\mathbf{m}}$, and $D_{\mathbf{v}}$, the adaptive laws can be modified by the projection algorithm (Wang, 1994) as following:

$$\dot{\hat{\boldsymbol{\alpha}}}_{\mathbf{I}}=\begin{cases}\sigma\hat{\boldsymbol{\Theta}}, & \text{if } \|\dot{\hat{\boldsymbol{\alpha}}}_{\mathbf{I}}\|<D_{\boldsymbol{\alpha}_{\mathbf{I}}}\text{or}(\|\hat{\boldsymbol{\alpha}}_{\mathbf{I}}\|=D_{\boldsymbol{\alpha}_{\mathbf{I}}}\text{ and }\sigma\hat{\boldsymbol{\Theta}}\le0)\\ \mathbf{Pr}(\sigma\hat{\boldsymbol{\Theta}}), & \text{if } (\|\hat{\boldsymbol{\alpha}}_{\mathbf{I}}\|=D_{\boldsymbol{\alpha}_{\mathbf{I}}}\text{ and }\sigma\hat{\boldsymbol{\Theta}}>0)\end{cases} \tag{58}$$

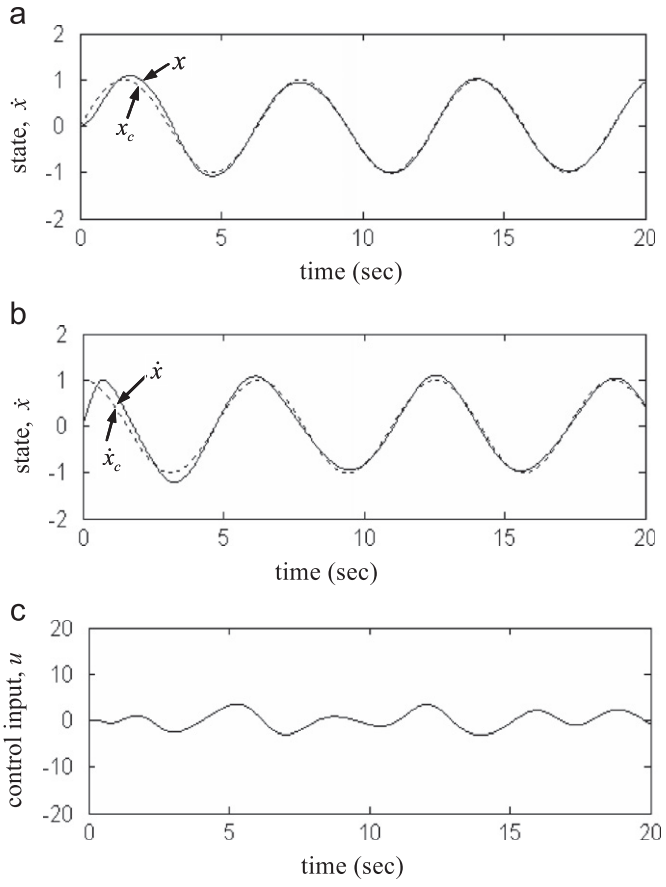

Fig. 4. Simulation results of ADCNC with RBF network for $q=2.1$.

$$\dot{\hat{\mathbf{m}}} = \begin{cases} \eta_m \sigma \mathbf{A}\hat{\boldsymbol{\alpha}}\ , & \text{if } \|\hat{\mathbf{m}}\| < D_{\mathbf{m}} \text{ or } (\|\hat{\mathbf{m}}\| = D_{\mathbf{m}} \text{ and } \sigma \hat{\mathbf{m}}^{\mathrm{T}} \mathbf{A}\hat{\boldsymbol{\alpha}} \le 0) \\ \mathbf{Pr}(\eta_m \sigma \mathbf{A}\hat{\boldsymbol{\alpha}}), & \text{if } (\|\hat{\mathbf{m}}\| = D_{\mathbf{m}} \text{ and } \sigma \hat{\mathbf{m}}^{\mathrm{T}} \mathbf{A}\hat{\boldsymbol{\alpha}} > 0) \end{cases} \tag{59}$$

$$\dot{\hat{\mathbf{v}}} = \begin{cases} \eta_v \sigma \mathbf{B}\hat{\boldsymbol{\alpha}}, \text{if } \|\hat{\mathbf{v}}\| < D_{\mathbf{v}} \text{ or } (\|\hat{\mathbf{v}}\| = D_{\mathbf{v}} \text{ and } \sigma \hat{\mathbf{v}}^{\mathrm{T}} \mathbf{B}\hat{\boldsymbol{\alpha}} \le 0) \\ \mathbf{Pr}(\eta_v \sigma \mathbf{B}\hat{\boldsymbol{\alpha}}) \text{if } (\|\hat{\mathbf{v}}\| = D_{\mathbf{v}} \text{ and } \sigma \hat{\mathbf{v}}^{\mathrm{T}} \mathbf{B}\hat{\boldsymbol{\alpha}} > 0) \end{cases} \tag{60}$$

where the projection operators are given as

$$\frac{\mathbf{Pr}(\sigma\hat{\boldsymbol{\Theta}}) = \sigma\hat{\boldsymbol{\Theta}} - \sigma\hat{\boldsymbol{\alpha}}^{\mathrm{T}}\hat{\boldsymbol{\Theta}}}{\|\hat{\boldsymbol{\alpha}}\|^2 \hat{\boldsymbol{\alpha}}_{\mathbf{I}}} \tag{61}$$

$$\frac{\mathbf{Pr}(\eta_m \sigma \mathbf{A}\hat{\boldsymbol{\alpha}}) = \eta_m \sigma \mathbf{A}\hat{\boldsymbol{\alpha}} - \eta_m \sigma \hat{\mathbf{m}}^{\mathrm{T}} \mathbf{A}\hat{\boldsymbol{\alpha}}}{\|\hat{\mathbf{m}}\|^2 \hat{\mathbf{m}}} \tag{62}$$

$$\frac{\mathbf{Pr}(\eta_v \sigma \mathbf{B}\hat{\boldsymbol{\alpha}}) = \eta_v \sigma \mathbf{B}\hat{\boldsymbol{\alpha}} - \eta_v \sigma \hat{\mathbf{v}}^{\mathrm{T}} \mathbf{B}\hat{\boldsymbol{\alpha}}}{\|\hat{\mathbf{v}}\|^2 \hat{\mathbf{v}}.} \tag{63}$$

If the initial value of $\hat{\boldsymbol{\alpha}}$ is bounded (i.e., $\hat{\boldsymbol{\alpha}}(0) \in \boldsymbol{\Omega}_{\boldsymbol{\alpha}}$), $\|\hat{\boldsymbol{\alpha}}\|$ is bounded by the constraint set $\boldsymbol{\Omega}_{\boldsymbol{\alpha}}$ for all $t \ge 0$. Similarly, the results can also be derived $\|\hat{\mathbf{m}}\|$ is bounded by the constraint set $\boldsymbol{\Omega}_{\mathbf{m}}$ if $\hat{\mathbf{m}}(0) \in \boldsymbol{\Omega}_{\mathbf{m}}$ and $\|\hat{\mathbf{v}}\|$ is bounded by the constraint set $\boldsymbol{\Omega}_{\mathbf{v}}$ if $\hat{\mathbf{v}}(0) \in \boldsymbol{\Omega}_{\mathbf{v}}$. Thus, the fact that the uncertain term ε is bounded can be guaranteed by the modified adaptive laws (58)–(60). Next, define variables as

$$V_{\boldsymbol{\alpha}_{\mathbf{I}}} = \tilde{\boldsymbol{\alpha}}_{\mathbf{I}}^{\mathrm{T}} (\sigma\hat{\boldsymbol{\Theta}} + \dot{\hat{\boldsymbol{\alpha}}}_{\mathbf{I}}) \tag{64}$$

$$V_{\mathbf{m}} = \tilde{\mathbf{m}}^{\mathrm{T}} \left(\sigma \mathbf{A}\hat{\boldsymbol{\alpha}} + \frac{\dot{\hat{\mathbf{m}}}}{\eta_m} \right) \tag{65}$$

and

$$V_{\mathbf{v}} = \tilde{\mathbf{v}}^{\mathrm{T}} \left(\sigma \mathbf{B}\hat{\boldsymbol{\alpha}} + \frac{\dot{\hat{\mathbf{v}}}}{\eta_v} \right). \tag{66}$$

If the projection algorithm is taken place, the property $\tilde{\mathbf{o}}^{\mathrm{T}}\hat{\mathbf{o}} = (1/2)(\|\mathbf{o}^*\|^2 - \|\hat{\mathbf{o}}\|^2 - \|\tilde{\mathbf{o}}\|^2) < 0$ (Lin and Peng, 2004) is applied according to $\|\hat{\mathbf{o}}\| = \boldsymbol{\Omega}_{\mathbf{o}} > \|\mathbf{o}^*\|$, where $\hat{\mathbf{o}} = \hat{\boldsymbol{\alpha}}_{\mathbf{I}}$, $\hat{\mathbf{m}}$, and $\hat{\mathbf{v}}$. Thus, the following equations can be obtained

$$V_{\boldsymbol{\alpha}_{\mathbf{I}}} = \frac{\sigma(\|\boldsymbol{\alpha}_{\mathbf{I}}^*\|^2 - \|\hat{\boldsymbol{\alpha}}_{\mathbf{I}}\|^2 - \|\tilde{\boldsymbol{\alpha}}_{\mathbf{I}}\|^2)}{\|\hat{\boldsymbol{\alpha}}\|^2} \hat{\boldsymbol{\alpha}}^{\mathrm{T}} \hat{\boldsymbol{\Theta}} \le 0, \quad \text{for } (\|\hat{\boldsymbol{\alpha}}_{\mathbf{I}}\| = D_{\boldsymbol{\alpha}_{\mathbf{I}}} \text{ and } \sigma\hat{\boldsymbol{\Theta}} > 0) \tag{67}$$

$$V_{\mathbf{m}} = \frac{\sigma(\|\mathbf{m}^*\|^2 - \|\hat{\mathbf{m}}\|^2 - \|\tilde{\mathbf{m}}\|^2)}{\|\hat{\mathbf{m}}\|^2} \hat{\mathbf{m}}^{\mathrm{T}} \mathbf{A}\hat{\boldsymbol{\alpha}} \le 0,$$

$$\text{for } (\|\hat{\mathbf{m}}\| = D_{\mathbf{m}} \text{ and } \sigma \hat{\mathbf{m}}^{\mathrm{T}} \mathbf{A}\hat{\boldsymbol{\alpha}} > 0) \tag{68}$$

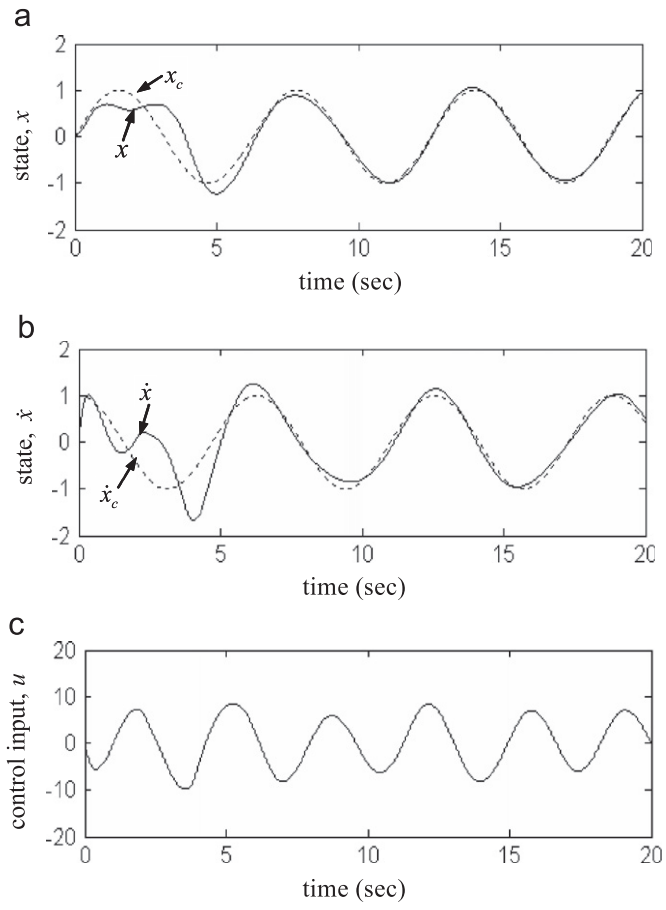

Fig. 5. Simulation results of ADCNC with RBF network for $q = 7.0$.

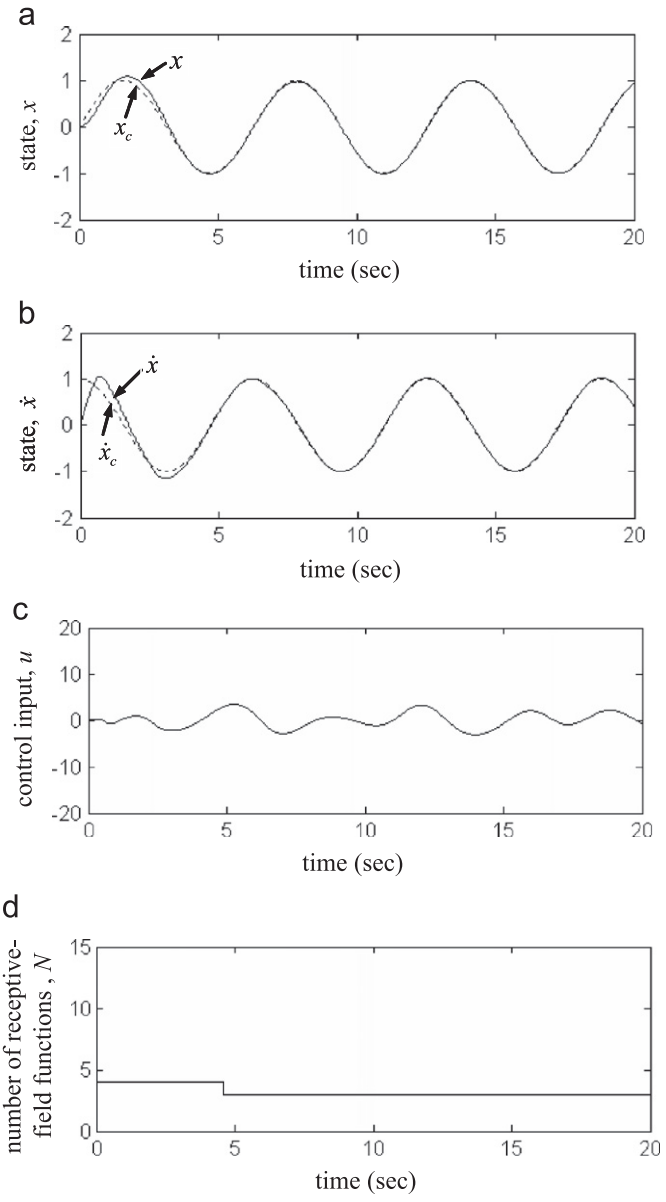

Fig. 6. Simulation results of ADCNC with I-type learning algorithm for $q = 2.1$.

and

$$V_{\mathbf{v}}=\frac{\sigma(\|\mathbf{v}^*\|^2-\|\hat{\mathbf{v}}\|^2-\|\tilde{\mathbf{v}}\|^2)}{\|\hat{\mathbf{v}}\|^2}\hat{\mathbf{v}}^{\mathrm{T}}\mathbf{B}\hat{\boldsymbol{\alpha}}\le 0,\quad \text{for}\quad (\|\hat{\mathbf{v}}\|=D_{\mathbf{v}}\text{and}\sigma\hat{\mathbf{v}}^{\mathrm{T}}\mathbf{B}\hat{\boldsymbol{\alpha}}>0). \tag{69}$$

Then, the derivative of Lyapunov function (49) can be rewritten as

$$\dot{V}_2=\eta_I V_{\boldsymbol{\alpha}_{\mathbf{I}}}+V_{\mathbf{m}}+V_{\mathbf{v}}-\sigma\eta_P\hat{\boldsymbol{\alpha}}_{\mathbf{p}}^{\mathrm{T}}\hat{\boldsymbol{\Theta}}+\sigma(\varepsilon+\dot{u}_{\mathrm{rc}})$$
$$\le\sigma(\varepsilon+\dot{u}_{\mathrm{rc}}). \tag{70}$$

By substituting the robust controller (54), (70) can be rewritten as

$$\dot{V}_2\le\sigma\varepsilon-\frac{(\delta^2+1)\sigma^2}{2\delta^2}\le-\frac{\sigma^2}{2}-\frac{1}{2}\left(\frac{\sigma}{\delta}-\delta\varepsilon\right)^2+\frac{1}{2}\delta^2\varepsilon^2$$
$$\le-\frac{\sigma^2}{2}+\frac{1}{2}\delta^2\varepsilon^2. \tag{71}$$

Using the same discussion in Section 3.3, the stability of the proposed ADCNC system with the projection algorithm can be guaranteed.

4. Simulation results

In this section, the proposed ADCNC system is applied to control a chaotic system to verify its effectiveness. It should be emphasized that the development of the ADCNC system does not need to know the system dynamics of the controlled system. In the following, the design steps of the ADCNC system are summarized as follows

Step 1: Initialize the pre-defined parameters of the ADCNC system.
Step 2: The tracking error e, sliding surface s and second-order sliding surface σ are given in (4), (5) and (7), respectively.
Step 3: Calculate the distance between incoming data $\mathbf{z}$ and $\mathbf{m}_j$. If $d_{\min}\ge d_{\mathrm{th}}$ is satisfied, a new Gaussian function is generated with initial parameters are give in (21)–(23). If condition is not satisfied, go to next step.

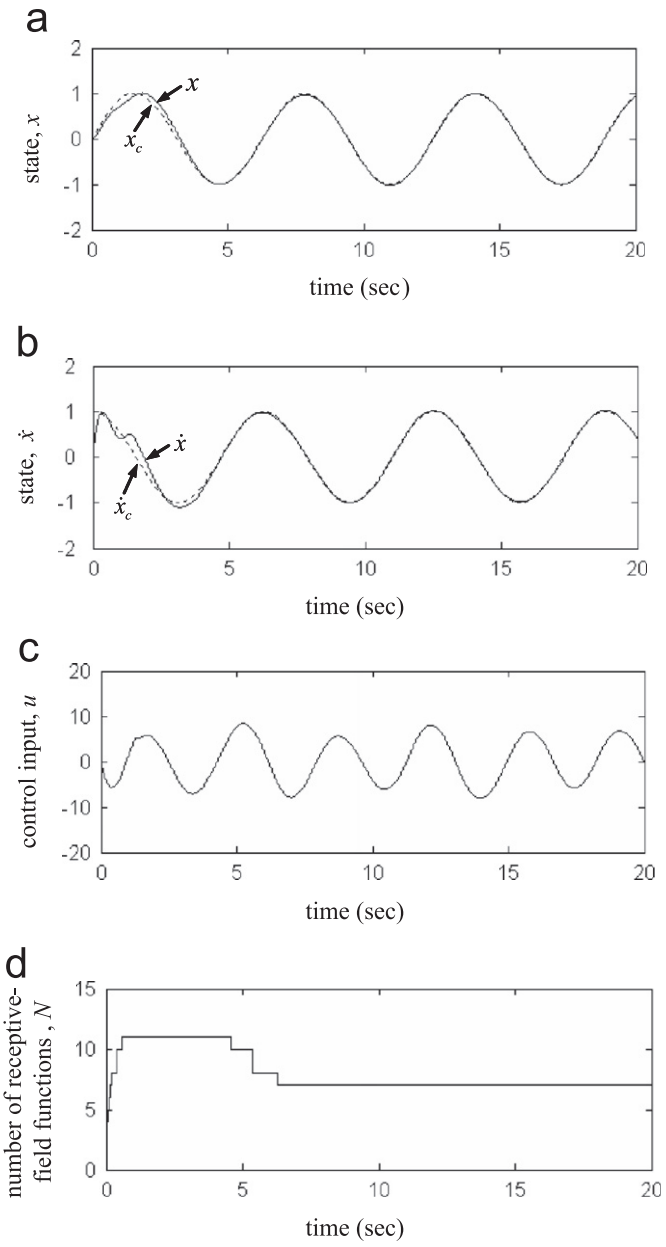

Fig. 7. Simulation results of ADCNC with I-type learning algorithm for $q=7.0$.

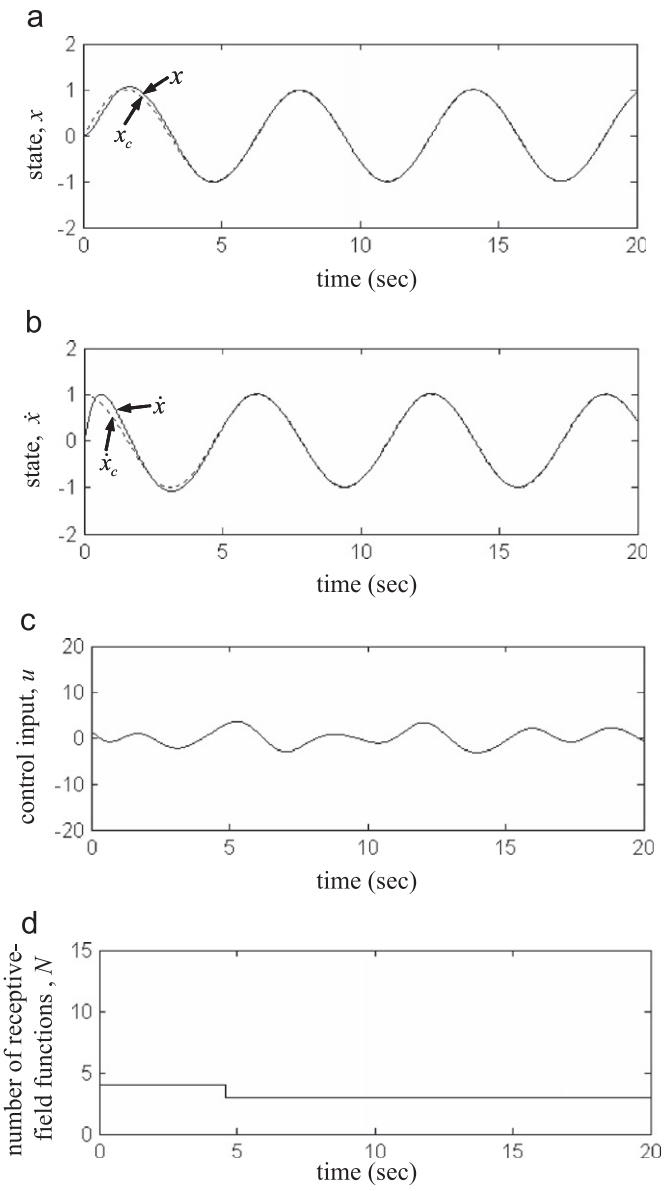

Fig. 8. Simulation results of ADCNC with PI-type learning algorithm for $q=2.1$.

Step 4: Update the significance index by (24). If $I_j \leq I\text{th}$ is satisfied, the jth receptive-field basis function is removed. If condition is not satisfied, go to Step 5.
Step 5: The output of DSCN identifier is given as $\hat{\boldsymbol{\alpha}}^T\hat{\boldsymbol{\Theta}}$, where $\hat{\boldsymbol{\alpha}}$, $\hat{\mathbf{m}}$ and $\hat{\mathbf{v}}$ are estimated by (50), (58), (59) and (60), respectively.
Step 6: The control law is given in (40), where the computation controller $\dot{u}_{cc}$ is given in (42) and the robust compensator is given in (54).
Step 7: If end learning is false, return to Step 2, but if end learning is true, go to STOP.

To show the effectiveness of the proposed DSCN which the network structure can grow or prune systematically and their parameters can be adjusted automatically, a fixing–structuring RBF network is applied to estimate the change of the control system dynamics. The simulation interval time is 0.001 s.The simulation results of the ADCNC system with RBF network are shown in Figs. 4 and 5 for $q=2.1$ and $q=7.0$, respectively. The tracking responses of state x are shown in Fig. 4(a) and Fig. 5(a), the tracking responses of state $\dot{x}$ are shown in Fig. 4(b) and Fig. 5(b), and the control inputs are shown in Fig. 4(c) and Fig. 5(c). From these simulation results, it can be seen that a favorable tracking performance can be achieved for $q=2.1$ but $q=7.0$ cannot. It implies that the RBF network cannot estimate the change of the control system dynamics well for $q=7.0$.

The simulation interval time is 0.001 s. The control parameters are selected as $a_1=b_1=2$, $a_2=b_2=1$, $\eta_P=20$, $\eta_I=20$, $\eta_m=\eta_\sigma=\eta_r=1$, $\delta=1.0$, $d\text{th}=0.3$, $\tau=0.01$, $\rho=0.2$, $I\text{th}=0.01$, and $\bar{v}=1.0$. These parameters are selected through trails. The parameters η_P, η_I, η_m, η_σ and η_r are the leaning rates. For small learning rates, convergence of tracking error can be easily guaranteed but with slow convergence speed. If the learning rates are too large, the parameter adaptation laws may lead to instability of the control systems. The parameters dth and Ith are the structure learning thresholds. If dth is smaller, more new layer can be easily generated to achieve a better approximation accuracy. If the computation loading is an important issue, Ith should be chosen larger, so that more existing receptive-field basis function can be canceled. To compare the tracking efficiency, the ADCNC with an integral (I)-type learning algorithm is

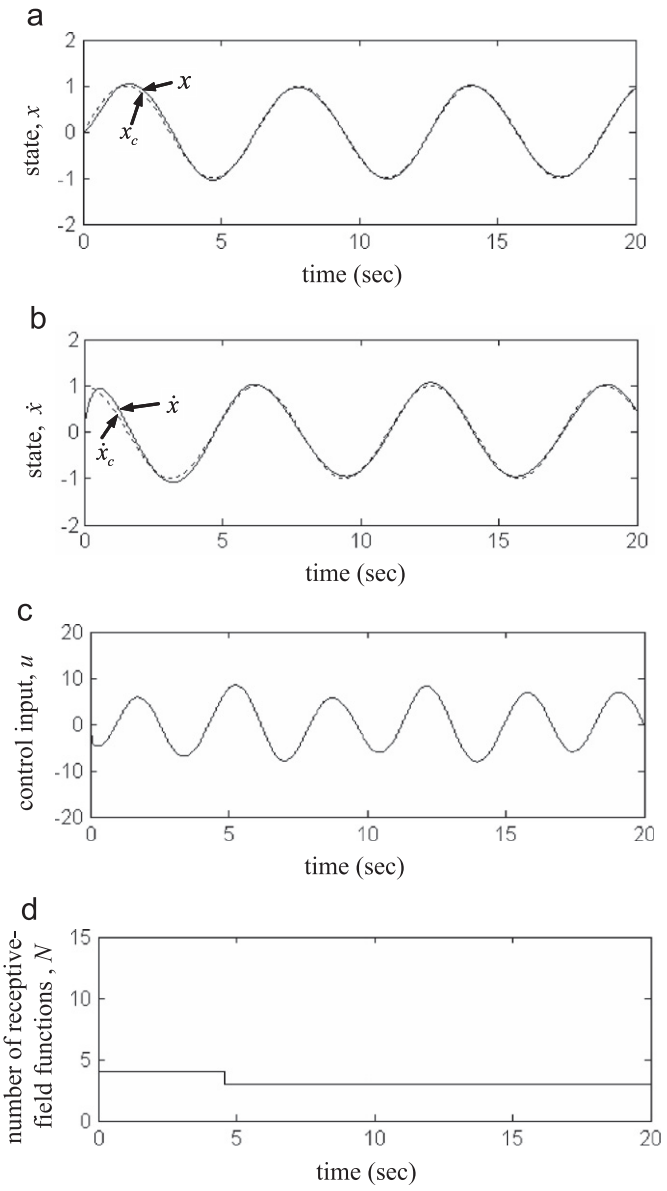

Fig. 9. Simulation results of ADCNC with PI-type learning algorithm for $q=7.0$.

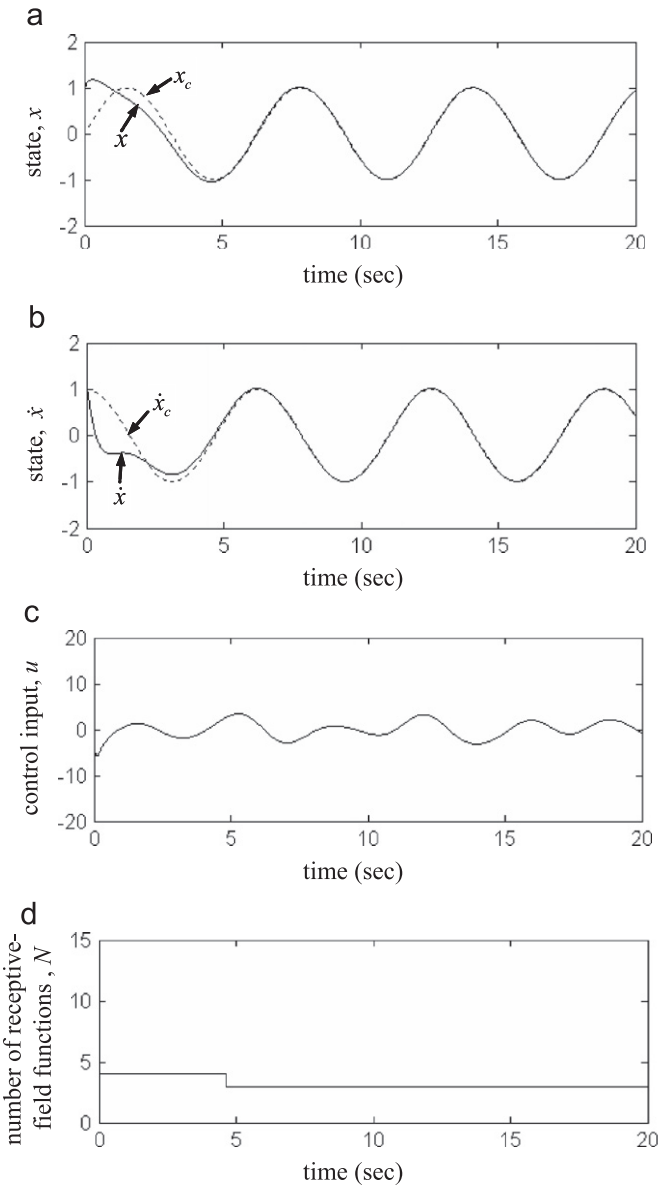

Fig. 10. Simulation results of ADCNC with difference initial point for $q=2.1$.

applied first. This is a special case of the developed ADCNC design method for $\eta_P=0$. The I-type learning algorithm can be found in previous research works (Lin and Chen, 2009; Yeh, 2007). The simulation results of the ADCNC system with I-type learning algorithm are shown in Figs. 6 and 7 for $q=2.1$ and $q=7.0$, respectively. The tracking responses of state x are shown in Fig. 6(a) and Fig. 7(a), the tracking responses of state $\dot{x}$ are shown in Fig. 6(b) and Fig. 7(b), the control inputs are shown in Fig. 6(c) and Fig. 7(c), and the number of the receptive-field basis function are shown in Fig. 6(d) and Fig. 7(d). From these simulation results, it can be seen that robust tracking performance can be achieved without any knowledge of system dynamic functions. However, the convergence of the controller parameter and tracking error is slow.

To speed up the convergence, the proposed PI-type learning algorithm is applied with $\eta_P=10$. The simulation results of the ADCNC system with PI-type learning algorithm are shown in Figs. 8 and 9 for $q=2.1$ and $q=7.0$, respectively. The tracking responses of state x are shown in Fig. 8(a) and Fig. 9(a), the tracking responses of state $\dot{x}$ are shown in Fig. 8(b) and Fig. 9(b), the control inputs are shown in Fig. 8(c) and Fig. 9(c), and the number of the receptive-field basis function are shown in Fig. 8(d) and Fig. 9(d). It is shown that the convergences of the tracking error and control parameter are accelerated by the PI-type learning algorithm. So the number of DSCN layer can be reduced comparing with Fig. 6(d) and Fig. 7(d). To demonstrate the robust performance of the proposed ADCNC system, a initial point changes to (1,1). The simulation results with initial point (1,1) are shown in Figs. 10 and 11 for $q=2.1$ and $q=7.0$, respectively. The tracking responses of state x are shown in Fig. 10(a) and Fig. 11(a), the tracking responses of state $\dot{x}$ are shown in Fig. 10(b) and Fig. 11(b), the control inputs are shown in Fig. 10(c) and Fig. 11(c), and the number of the receptive-field basis function are shown in Fig. 10(d) and Fig. 11(d). The simulation results show that not only the perfect tracking can be achieved but also the PI-type learning algorithm can speed up the convergence of the tracking error.

Finally, to achieve smaller attenuation level, the case for $\delta=0.1$ is reconsidered. The simulation results of the ADCNC system with smaller attenuation level are shown in Figs. 12 and 13 for $q=2.1$

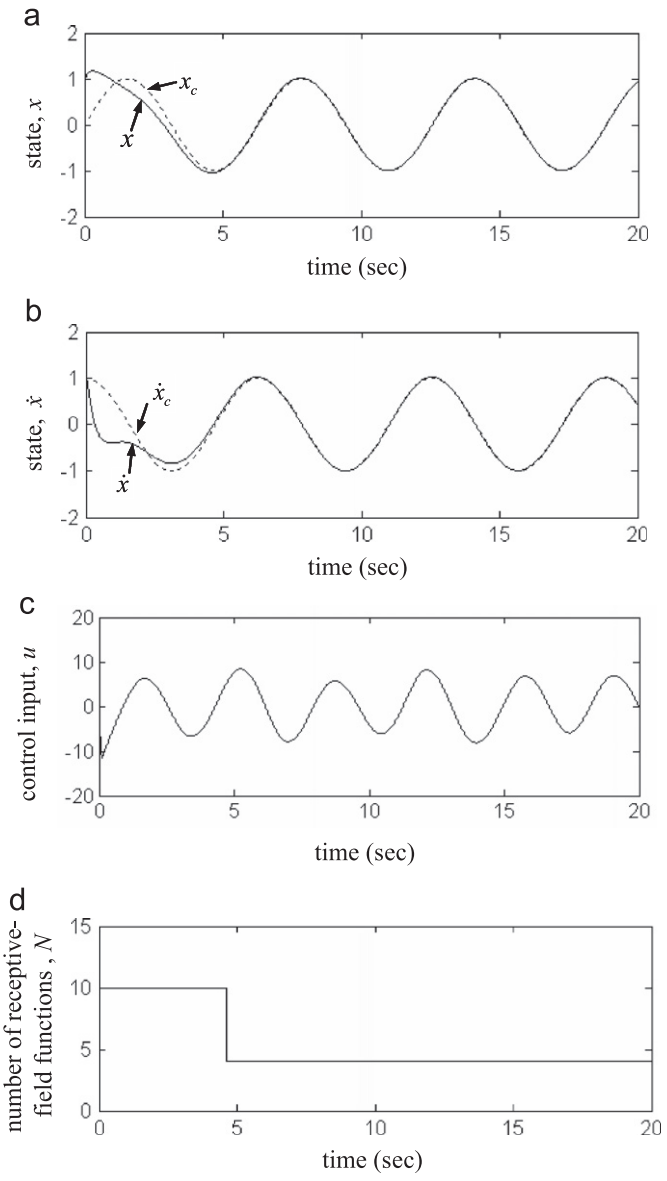

Fig. 11. Simulation results of ADCNC with difference initial point for $q=7.0$.

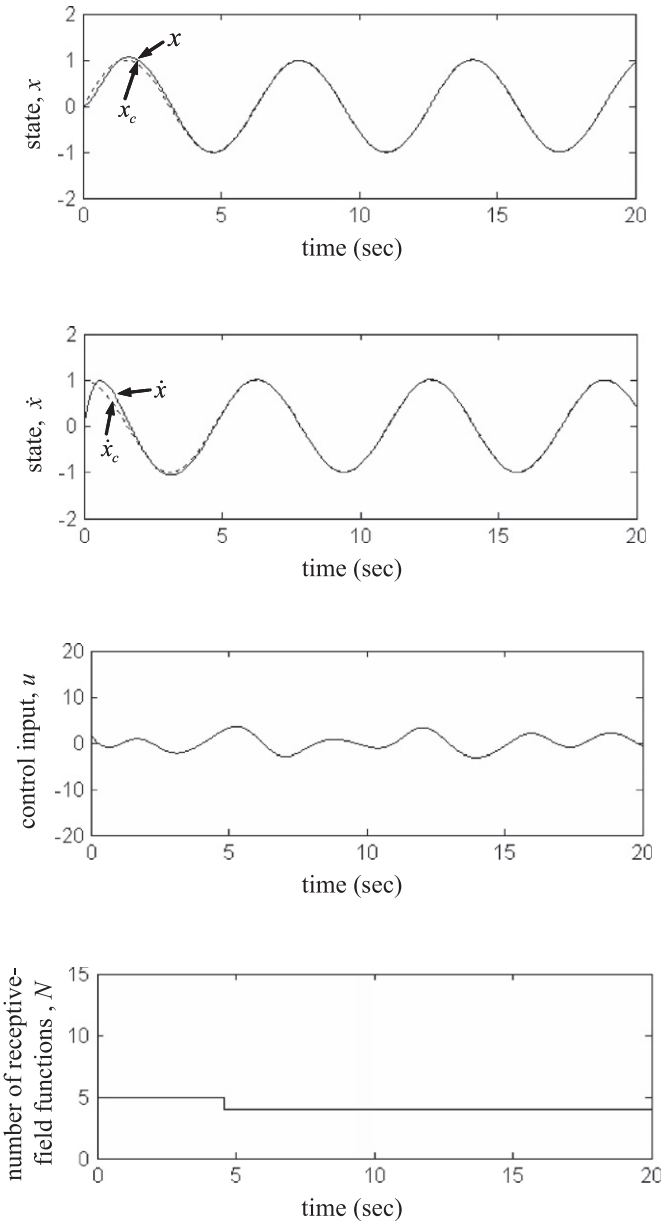

Fig. 12. Simulation results of ADCNC with smaller attenuation level for $q=2.1$.

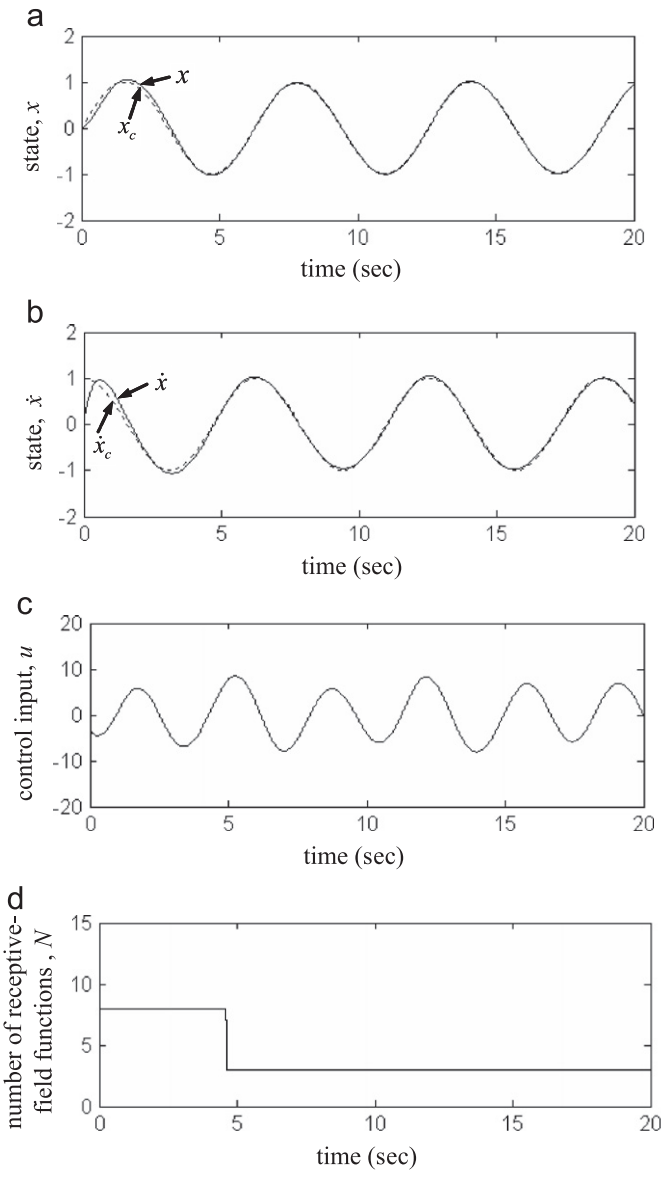

Fig. 13. Simulation results of ADCNC with smaller attenuation level for $q=7.0$.

and $q=7.0$, respectively. The tracking responses of state x are shown in Fig. 12(a) and Fig. 13(a), the tracking responses of state $\dot{x}$ are shown in Fig. 12(b) and Fig. 13(b), the control inputs are shown in Fig. 12(c) and Fig. 13(c), and the number of the receptive-field basis function are shown in Fig. 12(d) and Fig. 13(d). It is shown that the better tracking performance can be achieved as specified attenuation level δ is chosen smaller.

5. Conclusions

This paper has successfully developed a dynamic structure CMAC network (DSCN) with online adjusting suitable memory space of the CMAC network. In the memory reinforcement process, new associative memories will be generated when the current architecture is insufficient. On the other hand, the inefficient memories will be detected and reorganized in the memory reorganization process. Then, an adaptive dynamic CMAC neural control (ADCNC) system with a proportional–integral (PI)-type adaptation learning algorithm is proposed. The proposed ADCNC system is composed of computation controller and a robust compensator. The computation controller containing a DSCN identifier is the principal controller and the robust compensator is designed to achieve a L_2 tracking performance with desired attenuation level. The proposed adaptation laws are derived in the sense of Lyapunov function and Barbalat's lemma, thus the system can be guaranteed to be stable. Finally, some simulation results show the effectiveness of the proposed ADCNC system. The simulation results verify the proposed PI-type learning algorithm can achieve a faster convergence of the tracking error and controller parameters. And, the self-constructing approach demonstrates the properties of generating or pruning the layers in CMAC network automatically.

Acknowledgment

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

References

Ananthraman, S., Gar, D.P., 1993. Training backpropagation and CMAC neural networks for control of a SCARA robot. Eng. Appl. Artif. Intell. 6, 105–115.
Chen, W., Tian, Y.P., 2009. Neural network approximation for periodically disturbed functions and applications to control design. Neurocomputing 72, 3891–3900.
Czarnigowski, J., 2010. A neural network model-based observer for idle speed control of ignition in SI engine. Eng. Appl. Artif. Intell. 23, 1–7.
Golea, N., Golea, A., Benmahammed, K., 2002. Fuzzy model reference adaptive control. IEEE Trans. Fuzzy Syst. 10, 436–444.
Hsu, C.F., 2007. Self-organizing adaptive fuzzy neural control for a class of nonlinear systems. IEEE Trans. Neural Networks 18, 1232–1241.
Hsu, C.F., 2011. Adaptive fuzzy wavelet neural controller design for chaos synchronization. Expert Syst. Appl. 38, 10475–10483.
Hsu, C.F., Chung, C.M., Lin, C.M., Hsu, C.Y., 2009. Adaptive CMAC neural control of chaotic systems with a PI-type learning algorithm. Expert Syst. Appl. 36, 11836–11843.
Jiang, Z.P., 2002. Advanced feedback control of the chaotic Duffing equation. IEEE Trans. Circuits Syst. 49 (Pt-I), 244–249.
Koshkouei, A.J., Burnham, K.J., Zinober, A.S.I., 2005. Dynamic sliding mode control design. IEE Proc. Control Theory Appl. 152, 392–396.
Lee, C.Y., Lin, C.J., Chen, H.J., 2007. A self-constructing fuzzy CMAC model and its applications. Inf. Sci. 177, 264–280.
Levant, A., 1993. Sliding order and sliding accuracy in sliding mode control. Int. J. Control 58, 1247–1263.
Levant, A., Pridor, A., Gitizadeh, R., Yaesh, I., Ben-Asher, J.Z., 2000. Aircraft pitch control via second order sliding technique. AIAA J. Guid. Control Dyn. 23, 586–594.
Lin, C.K., 2009. Fuzzy-basis-function-network-based H_∞ tracking control for robotic manipulators using only position feedback. IEEE Trans. Fuzzy Syst. 17, 1208–1216.
Lin, C.M., Chen, L.Y., Chen, C.H., 2007. RCMAC hybrid control for MIMO uncertain nonlinear systems using sliding-mode technology. IEEE Trans. Neural Networks 18, 708–720.
Lin, C.M., Chen, T.Y., 2009. Self-organizing CMAC control for a class of MIMO uncertain nonlinear systems. IEEE Trans. Neural Networks 20, 1377–1384.
Lin, C.M., Peng, Y.F., 2004. Adaptive CMAC-based supervisory control for uncertain nonlinear systems. IEEE Trans. Syst. Man Cybern. 34 (Pt-B), 1248–1260.
Lin, C.M., Peng, Y.F., 2005. Missile guidance law design using adaptive cerebellar model articulation controller. IEEE Trans. Neural Networks 16, 636–644.
Lin, C.T., Lee, C.S.G., 1996. Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems. Prentice-Hall, Englewood Cliffs, NJ.
Lin, F.J., Chen, S.Y., Shyu, K.K., 2009. Robust dynamic sliding-mode control using adaptive RENN for magnetic levitation system. IEEE Trans. Neural Networks 20, 938–951.
Miguel, H., Yu, T., 2009. Adaptive output-feedback decentralized control of a class of second order nonlinear systems using recurrent fuzzy neural networks. Neurocomputing 73, 461–467.
Parra-Vega, V., Arimoto, S., Liu, Y.H., Hirzinger, G., Akella, P., 2003. Dynamic sliding PID control for tracking of robot manipulators: theory and experiments. IEEE Trans. Rob. Autom. 19, 967–976.
Peng, Y.F., Lin, C.M., 2007. Intelligent motion control of linear ultrasonic motor with H^∞ tracking performance. IET Control Theory Appl. 1, 9–17.
Peng, Y.F., 2009. Robust intelligent sliding model control using recurrent cerebellar model articulation controller for uncertain nonlinear chaotic systems. Chaos, Solitons Fractals 39, 150–167.

Slotine, J.J.E., Li, W.P., 1991. Applied Nonlinear Control. Prentice-Hall, Englewood Cliffs, NJ.

Tseng, C.S., 2008. A novel approach to H_∞ decentralized fuzzy-observer-based fuzzy control design for nonlinear interconnected systems. IEEE Trans. Fuzzy Syst. 16, 1337–1350.

Utkin, V.I., 1978. Sliding Modes and theirs Applications in Variable Structure Systems. MIR Editors, Moscow.

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

Wang, W.Y., Chan, M.L., Hsu, C.C.J., Lee, T.T., 2002. H^∞ tracking-based sliding mode control for uncertain nonlinear systems via an adaptive fuzzy-neural approach. IEEE Trans. Syst. Man Cybern. 32 (Pt-B), 483–492.

Wu, C.W., Hsu, C.F., Hwang, C.K., 2011. Master-slave chaos synchronization using adaptive TSK-type CMAC neural control. J. Franklin Inst. 348, 1847–1868.

Wu, T.F., Tsai, P.S., Chang, F.R., Wang, L.S., 2006. Adaptive fuzzy CMAC control for a class of nonlinear system with smooth compensation. IEE Proc. Control Theory Appl. 153, 647–657.

Yeh, M.F., 2007. Single-input CMAC control system. Neurocomputing 70, 2638–2644.

Yeh, M.F., Chang, K.C., 2006. A self-organizing CMAC network with gray credit assignment. IEEE Trans. Syst. Man Cybern. 36 (Pt-B), 623–635.

Yen, M.C., Hsu, C.F., Chung, I.H., 2012. Design of a CMAC-based smooth adaptive neural controller with a saturation compensator. Neural Comput. Appl. 21, 35–44.

Zhao, Y.J., Yu, D.L., 2009. Neural network model-based automotive engine air/fuel ratio control and robustness evaluation. Eng. Appl. Artif. Intell. 22, 171–180.