Applied Soft Computing 23 (2014) 355–365

Contents lists available at ScienceDirect

Applied Soft Computing

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

Intelligent complementary sliding-mode control with dead-zone parameter modification

Chun-Fei Hsu [a,*], Tzu-Chun Kuo [b]

[a] Department of Electrical Engineering, Tamkang University, No. 151, Yingzhuan Road, Tamsui District, New Taipei City 25137, Taiwan
[b] Department of Electrical Engineering, Chien Hsin University of Science and Technology, No. 229, Chien Hsin Road, Zhongli City, Taoyuan County 32097, Taiwan

ARTICLE INFO

Article history:
Received 25 August 2013
Received in revised form 27 April 2014
Accepted 3 June 2014
Available online 5 July 2014

Keywords:
Neural control
Sliding-mode control
Neural fuzzy inference network
Recurrent neural network

ABSTRACT

This paper proposes an intelligent complementary sliding-mode control (ICSMC) system which is composed of a computed controller and a robust controller. The computed controller includes a neural dynamics estimator and the robust compensator is designed to prove a finite L_2-gain property. The neural dynamics estimator uses a recurrent neural fuzzy inference network (RNFIN) to approximate the unknown system term in the sense of the Lyapunov function. In traditional neural network learning process, an over-trained neural network would force the parameters to drift and the system may become unstable eventually. To resolve this problem, a dead-zone parameter modification is proposed for the parameter tuning process to stop when tracking performance index is smaller than performance threshold. To investigate the capabilities of the proposed ICSMC approach, the ICSMC system is applied to a one-link robotic manipulator and a DC motor driver. The simulation and experimental results show that favorable control performance can be achieved in the sense of the L_2-gain robust control approach by the proposed ICSMC scheme.

© 2014 Elsevier B.V. All rights reserved.

Introduction

Modeling inaccuracies can have strong adverse effects on control system. One of the most important approaches to dealing with model uncertainty is robust control. If all system uncertainties are bounded, a sliding-mode control (SMC) can be applied. This control method has been studied extensively for over 50 years due to its invariance property to system uncertainties on the sliding surface [1]. Meanwhile, many applications of the SMC system have been presented [2–6]. Since the system uncertainties unavoidably exist in the practical systems, a switching control term that consists of a sign function is required to maintain the sliding variable on the sliding surface. The gain of the switching control term is chosen based on the system uncertainty bound; however, a high gain will result the undesirable chattering phenomenon. A tradeoff between the chattering phenomenon and the control accuracy arises. The chattering problem is one of the most critical handicaps for applying the SMC system to real applications.

To overcome the chattering phenomenon, a boundary layer technique is adopted by replacing the sign function with a saturation function in the switching control term. However, an indefinite steady state error would cause depending on the width of the boundary layer [7]. To deal with this problem, an adaptive sliding-mode control with uncertainty bound estimator is proposed to online estimate the bound of system uncertainties [8,9]. But, the uncertainty bound estimator cannot avoid growing unboundedly. On the other hand, some researchers proposed a dynamic sliding-mode control (DSMC) system [10,11]. Due to the integration method for obtaining the practical control input, the chattering phenomenon can be improved effectively. Recently, some researchers proposed a complementary sliding-mode control (CSMC) system [12,13] which not only alleviates the chattering phenomena but also possesses favorable control accuracy. The nominal dynamics of control plants is required in the DSMC and CSMC schemes. Since the nominal dynamics may be unknown or perturbed in real applications, the DSMC and CSMC systems cannot be obtained.

To overcome this problem, many studies on neural fuzzy networks for controller development and for describing the system dynamics of unknown nonlinear systems have been published [14–17]. A main property of the neural networks regarding feedback control purpose is a universal function approximation property. It is known that the neural networks with a finite number of neurons can only estimate a function with a limited accuracy.

* Corresponding author. Tel.: +886 226215656x2615.
E-mail addresses: fei@ee.tku.edu.tw (C.-F. Hsu), tck@uch.edu.tw (T.-C. Kuo).

http://dx.doi.org/10.1016/j.asoc.2014.06.008
1568-4946/© 2014 Elsevier B.V. All rights reserved.

Because of the existence of approximation errors, the learning laws could make the adjustable parameters of neural fuzzy networks drift and the closed-loop system might become unstable eventually. A common approach to solve this problem is to employ the projection modification [18,19]. However, the projection modification is only able to limit the drift of adjustable parameters in a given region. It implies that the goal of parameters learning cannot be achieved when the adjustable parameters are fixed after a period of learning [20]. Meanwhile, in the derivation of the adaptive law, the derivatives of the optimal adjustable parameters are assumed to be zeros. However, when the adjustable parameters are not fixed in a finite time, it is difficult to say that the derivatives of the optimal adjustable parameters are zero.

Since the output of a dynamic nonlinear plant is a function of past output, past input or both, identification and control of this dynamic nonlinear plant is not a static control problem. Though the favorable control performance can be achieved using feedforward neural networks in [14–17], the usage of the long tapped delay input will result in a large network size. To deal with this problem, interest in using recurrent neural networks for processing dynamic nonlinear plants has been steadily growing [21–24]. Due to the recurrent neural networks have an internal feedback loop, they can capture the dynamic response and information storing ability. As a result, this paper proposes a recurrent neural fuzzy inference network (RNFIN) which is inherently a recurrent multilayered connectionist neural network for realizing the basic elements and functions of a dynamic fuzzy inference system.

The dead-zone nonlinearity is common in mechanical connections, hydraulic servovalves, piezoelectric translators, and electric servomotors. The effects of the dead-zone nonlinearities cannot be neglected in controller design [25,26]. In this paper, an intelligent complementary sliding-mode control (ICSMC) system which is composed of a computed controller and a robust controller is proposed. A RNFIN is utilized to online approximate the unknown nonlinear term in the system dynamics. To avoid the parameters-drift problem, a dead-zone parameter modification is considered in which the parameter tuning process will be stop when tracking performance index is smaller than performance threshold. Thus, the parameter-drift problem in the traditional parameter learning algorithm can be reduced. Meanwhile, to show the effectiveness of the ICSMC system, a one-link robotic manipulator and a DC motor driver are given as the example studies. The simulation and experimental results verify that the proposed ICSMC system can achieve favorable control performance in the sense of the L_2-gain robust control approach.

Problem formulation

For the applicability of the proposed controller design, a second-order nonlinear plant is expressed in the following form as

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

where $\mathbf{x} = [x, \dot{x}]^{\mathrm{T}}$ is the state vector, $f(\mathbf{x}, t)$ and $g(\mathbf{x}, t)$ are the system dynamics, $d(t)$ is the external disturbance, and u is the control input. Without losing generality, it is assume $g(\mathbf{x}, t) \neq 0$ for all $\mathbf{x}$. The control objective is to find a control law so that the state trajectory x can track a state command x_c closely. To achieve this control objective, a tracking error is defined as

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

Rewriting (1), the $\ddot{e}$ can be given as

$$\ddot{e} = z(\mathbf{x}, t) + u \tag{3}$$

where the nonlinear system term $z(\mathbf{x}, t) = -\ddot{x}_c + (1 - (1/g(\mathbf{x}, t)))\ddot{x} + ((f(\mathbf{x}, t) + d(t))/g(\mathbf{x}, t))$. To design a CSMC system, a sliding surface is designed as [12,13]

$$s = \dot{e} + 2\lambda e + \lambda^2 \int_0^t e(\tau)d\tau \tag{4}$$

where λ is a positive constant. We can obtain that

$$\begin{aligned} \dot{s} &= \ddot{e} + 2\lambda\dot{e} + \lambda^2 e \\ &= z(\mathbf{x}, t) + u + 2\lambda\dot{e} + \lambda^2 e \end{aligned} \tag{5}$$

Next, a complementary sliding surface is designed as

$$s_c = \dot{e} - \lambda^2 \int_0^t e(\tau)d\tau \tag{6}$$

A significant result concerning the relationship between s and s_c can be obtained as [12,13]

$$\dot{s}_c = \dot{s} - \lambda(s + s_c) \tag{7}$$

In (7), it shows that the two sliding surfaces s and s_c satisfied the reaching condition simultaneously. Thus, the control variables $\dot{e}$, e and $\int_0^t e(\tau)d\tau$ will move toward the neighborhood of the intersection of the two sliding surfaces ($s = s_c = 0$) and will slide along it toward the origin $\dot{e} = e = \int_0^t e(\tau)d\tau = 0$ in finite time. To guarantee the stability of the CSMC system, consider the candidate Lyapunov function in the following form as

$$V_1(t) = \frac{1}{2}s^2 + \frac{1}{2}s_c^2. \tag{8}$$

Differentiating (8) with respect to time and using (5) and (7) yields

$$\begin{aligned} \dot{V}_1(t) &= s\dot{s} + s_c\dot{s}_c \\ &= (s + s_c)(\dot{s} - \lambda s_c) \\ &= (s + s_c)[z(\mathbf{x}, t) + u + 2\lambda\dot{e} + \lambda^2 e - \lambda s_c] \end{aligned} \tag{9}$$

Assuming that all the parameters in $z(\mathbf{x}, t)$ are known, an ideal CSMC system is given as

$$u^* = -z(\mathbf{x}, t) - \lambda(2\dot{e} + \lambda e + s) \tag{10}$$

Imposing the control law $u = u^*$ into (9) yields

$$\begin{aligned} \dot{V}_1(t) &= (s + s_c)(-\lambda s - \lambda s_c) \\ &= -\lambda(s + s_c)^2 \leq 0. \end{aligned} \tag{11}$$

It implies that $s \to 0$ and $s_c \to 0$ as $t \to \infty$ [27]. Then, the stability of the ideal CSMC system can be guaranteed. Since the nonlinear system term $z(\mathbf{x}, t)$ may be unknown or perturbed in practical applications, the ideal CSMC system cannot be precisely obtained.

ICSMC system design

The proposed ICSMC system is shown in Fig. 1, where the controller output is defined as

$$u_{ics} = u_{cc} + u_{rc}. \tag{12}$$

The neural dynamic estimator $\hat{z}_{nn}$ is designed to approximate the nonlinear system term $z(\mathbf{x}, t)$. The computed controller u_{cc} is designed as $-\hat{z}_{nn} - k(2\dot{e} + ke + s)$ and the robust compensator u_{rc} is designed to eliminate the effect of the approximation error introduced by the neural dynamic estimator.

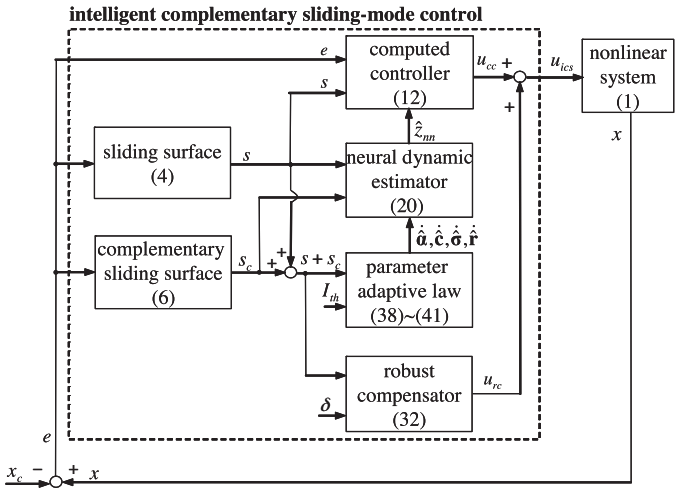

Fig. 1. The block diagram of the ICSMC system.

Description of a RNFIN

Consider a RNFIN with two external inputs and a single output as shown in Fig. 2 [28,29]. The jth fuzzy rule can be described as

$$\begin{aligned}\text{Rule } j: \quad & \text{IF } s \text{ is } A_{1j} \text{ and } s_c \text{ is } A_{2j} \text{ and } h_j^{pre} \text{ is } G, \\ & \text{THEN } z_{nn} \text{ is } \alpha_j \text{ and } h_1 \text{ is } r_{ji} \text{ and } \ldots \text{ and } h_n \text{ is } r_{jn}\end{aligned} \tag{13}$$

where h_j^{pre} is the jth internal variable in the previous time, n is the total number of fuzzy rules, A and G are the fuzzy sets, α and r are the consequent parameters for inference output z_{nn} and h, respectively. The external inputs of the RNFIN are $x_1 = s$ and $x_2 = s_c$ in this study. In the RNFIN, two types of membership functions are used. For external input, the following Gaussian membership function is used as

$$\phi_{ij} = \exp\left(-\frac{(x_i - c_{ij})^2}{(\sigma_{ij})^2}\right) \quad j = 1, 2, \ldots, n \tag{14}$$

where c_{ij} and σ_{ij} are the center and the width of the Gaussian membership function of the jth term of the ith input variable, respectively. For internal variable, the following sigmoid membership function is used as

$$\varsigma_k = \frac{1}{1 + \exp(-h_k^{pre})} \quad k = 1, 2, \ldots, n \tag{15}$$

where h_k denotes the output signal of the kth recurrent neuron. The network output is obtained as

$$z_{nn} = \sum_{j=1}^{n} \alpha_j \Theta_j \tag{16}$$

where α_j is the consequent parameter for the inference output z_{nn} and $\Theta_j = \varsigma_j \prod_{i=1}^{2} \phi_{ij}$ is the firing weight of the jth fuzzy rule. The recurrent output is obtained as

$$h_k = \sum_{j=1}^{n} r_{jk} \Theta_j, \quad k = 1, 2, \ldots, n \tag{17}$$

where r_{jk} is the consequent parameter for inference output h_k. Then, the output of the RNFIN can be represented in a vector form as

$$z_{nn} = \alpha^{\mathrm{T}} \boldsymbol{\Theta}(\mathbf{c}, \boldsymbol{\sigma}, \mathbf{r}) \tag{18}$$

where $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_n]^{\mathrm{T}}$, $\boldsymbol{\Theta} = [\Theta_1, ..., \Theta_n]^{\mathrm{T}}$, $\mathbf{c} = [c_{11}, c_{21}, \cdots, c_{1n}, c_{2n}]^{\mathrm{T}}$, $\boldsymbol{\sigma} = [\sigma_{11}, \sigma_{21}, ..., \sigma_{1n}, \sigma_{2n}]^{\mathrm{T}}$ and $\mathbf{r} = [r_{11}, r_{21}, \cdots, r_{n1}, \cdots, r_{1n}, r_{2n}, \cdots, r_{nn}]^{\mathrm{T}}$. By the universal function approximation property, an ideal RNFIN z_{nn}^* can be obtained as [28,29]

$$z(\mathbf{x}, t) = z_{nn}^* + \Delta = \boldsymbol{\alpha}^{*\mathrm{T}} \boldsymbol{\Theta}(\mathbf{c}^*, \boldsymbol{\sigma}^*, \mathbf{r}^*) + \Delta = \boldsymbol{\alpha}^{*\mathrm{T}} \boldsymbol{\Theta}^* + \Delta \tag{19}$$

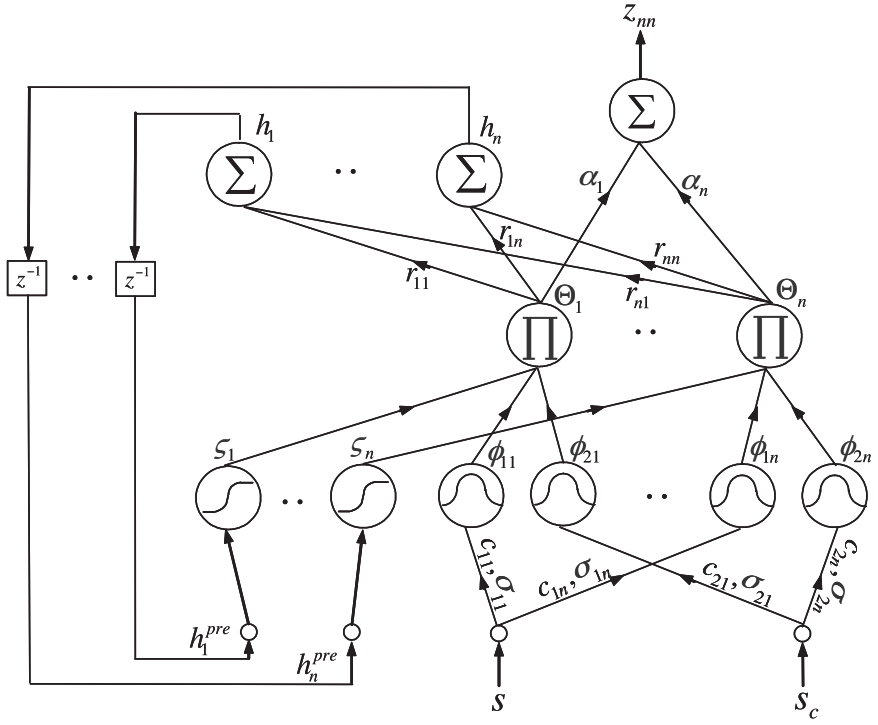

Fig. 2. Network structure of RNFIN.

where Δ denotes the approximation error, $\boldsymbol{\alpha}^*$ and $\boldsymbol{\Theta}^*$ are the optimal parameter vectors of $\boldsymbol{\alpha}$ and $\boldsymbol{\Theta}$, respectively, and $\mathbf{c}^*$, $\boldsymbol{\sigma}^*$ and $\mathbf{r}^*$ are the optimal parameter vectors of $\mathbf{c}$, $\boldsymbol{\sigma}$ and $\mathbf{r}$, respectively. In fact, the optimal parameter vectors that are needed for the best approximation cannot be determined. An estimated RNFIN $\hat{z}_{nn}$ is defined as

$$\hat{z}_{nn} = \hat{\alpha}^{\mathrm{T}}\boldsymbol{\Theta}(\hat{\mathbf{c}}, \hat{\boldsymbol{\sigma}}, \hat{\mathbf{r}}) = \hat{\boldsymbol{\alpha}}^{\mathrm{T}}\hat{\boldsymbol{\Theta}} \tag{20}$$

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

$$\begin{aligned}\tilde{z} &= z(\mathbf{x}, t) - \hat{z}_{nn} \\ &= \tilde{\boldsymbol{\alpha}}^{\mathrm{T}}\hat{\boldsymbol{\Theta}} + \hat{\boldsymbol{\alpha}}^{\mathrm{T}}\tilde{\boldsymbol{\Theta}} + \tilde{\boldsymbol{\alpha}}^{\mathrm{T}}\tilde{\boldsymbol{\Theta}} + \Delta\end{aligned} \tag{21}$$

where $\tilde{\boldsymbol{\alpha}} = \boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}}$ and $\tilde{\boldsymbol{\Theta}} = \boldsymbol{\Theta}^* - \hat{\boldsymbol{\Theta}}$. Take the expansion of $\tilde{\boldsymbol{\Theta}}$ in a Taylor series to obtain [30]

$$\tilde{\Theta} = \mathbf{A}^{\mathrm{T}}\tilde{\mathbf{c}} + \mathbf{B}^{\mathrm{T}}\tilde{\boldsymbol{\sigma}} + \mathbf{C}^{\mathrm{T}}\tilde{\mathbf{r}} + \mathbf{h} \tag{22}$$

where $\tilde{\mathbf{c}} = \mathbf{c}^* - \hat{\mathbf{c}}$, $\tilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}^* - \hat{\boldsymbol{\sigma}}$, $\tilde{\mathbf{r}} = \mathbf{r}^* - \hat{\mathbf{r}}$, $\mathbf{h}$ is a vector of high order terms, $\mathbf{A} = \left[\frac{\partial\Theta_1}{\partial\mathbf{c}} \cdots \frac{\partial\Theta_n}{\partial\mathbf{c}}\right]|_{\mathbf{c}=\hat{\mathbf{c}}}$, $\mathbf{B} = \left[\frac{\partial\Theta_1}{\partial\boldsymbol{\sigma}} \cdots \frac{\partial\Theta_n}{\partial\boldsymbol{\sigma}}\right]|_{\boldsymbol{\sigma}=\hat{\boldsymbol{\sigma}}}$ and $\mathbf{C} = \left[\frac{\partial\Theta_1}{\partial\mathbf{r}} \cdots \frac{\partial\Theta_n}{\partial\mathbf{r}}\right]|_{\mathbf{r}=\hat{\mathbf{r}}}$. Substituting (22) into (21) yields

$$\begin{aligned}\tilde{z}_{nn} &= \tilde{\boldsymbol{\alpha}}^{\mathrm{T}}\hat{\boldsymbol{\Theta}} + \hat{\boldsymbol{\alpha}}^{\mathrm{T}}(\mathbf{A}^{\mathrm{T}}\tilde{\mathbf{c}} + \mathbf{B}^{\mathrm{T}}\tilde{\boldsymbol{\sigma}} + \mathbf{C}^{\mathrm{T}}\tilde{\mathbf{r}} + \mathbf{h}) + \tilde{\boldsymbol{\alpha}}^{\mathrm{T}}[\mathbf{A}^{\mathrm{T}}(\mathbf{c}^* - \hat{\mathbf{c}}) + \mathbf{B}^{\mathrm{T}}(\boldsymbol{\sigma}^* - \hat{\sigma}) + \mathbf{C}^{\mathrm{T}}(\mathbf{r}^* - \hat{\mathbf{r}}) + \mathbf{h}] + \Delta \\ &= \tilde{\boldsymbol{\alpha}}^{\mathrm{T}}(\hat{\boldsymbol{\Theta}} - \mathbf{A}^{\mathrm{T}}\hat{\mathbf{c}} - \mathbf{B}^{\mathrm{T}}\hat{\boldsymbol{\sigma}} - \mathbf{C}^{\mathrm{T}}\hat{\mathbf{r}}) + \tilde{\mathbf{c}}^{\mathrm{T}}\mathbf{A}\hat{\boldsymbol{\alpha}} + \tilde{\boldsymbol{\sigma}}^{\mathrm{T}}\mathbf{B}\hat{\boldsymbol{\alpha}} + \tilde{\mathbf{r}}^{\mathrm{T}}\mathbf{C}\hat{\boldsymbol{\alpha}} + \varepsilon\end{aligned} \tag{23}$$

where the uncertain term $\varepsilon = \boldsymbol{\alpha}^{*\mathrm{T}}\mathbf{h} + \tilde{\boldsymbol{\alpha}}^{\mathrm{T}}(\mathbf{A}^{\mathrm{T}}\mathbf{c}^* + \mathbf{B}^{\mathrm{T}}\boldsymbol{\sigma}^* + \mathbf{C}^{\mathrm{T}}\mathbf{r}^*) + \Delta$ and $\hat{\boldsymbol{\alpha}}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\tilde{\mathbf{c}} = \tilde{\mathbf{c}}^{\mathrm{T}}\mathbf{A}\hat{\boldsymbol{\alpha}}$, $\hat{\boldsymbol{\alpha}}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\tilde{\boldsymbol{\sigma}} = \tilde{\boldsymbol{\sigma}}^{\mathrm{T}}\mathbf{B}\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\alpha}}^{\mathrm{T}}\mathbf{C}^{\mathrm{T}}\tilde{\mathbf{r}} = \tilde{\mathbf{r}}^{\mathrm{T}}\mathbf{C}\hat{\boldsymbol{\alpha}}$ are used since they are scalars.

Parameter learning of the ICSMC system

Imposing the control law $u = u_{ics}$ into (5), it is obtained that

$$\begin{aligned}\dot{s} &= z(\mathbf{x}, t) - \hat{z}_{nn} - \lambda(2\dot{e} + ke + s) + u_{rc} + 2\lambda\dot{e} + \lambda^2 e \\ &= \tilde{z} - \lambda s + u_{rc}\end{aligned} \tag{24}$$

where the estimation error $\tilde{z}$ is $z(\mathbf{x}, t) - \hat{z}_{nn}$. Imposing (23) into (24) yields

$$\begin{aligned}\dot{s} = \tilde{\boldsymbol{\alpha}}^{\mathrm{T}}(\hat{\boldsymbol{\Theta}} - \mathbf{A}^{\mathrm{T}}\hat{\mathbf{c}} - \mathbf{B}^{\mathrm{T}}\hat{\boldsymbol{\sigma}} - \mathbf{C}^{\mathrm{T}}\hat{\mathbf{r}}) + \tilde{\mathbf{c}}^{\mathrm{T}}\mathbf{A}\hat{\boldsymbol{\alpha}} + \tilde{\boldsymbol{\sigma}}^{\mathrm{T}}\mathbf{B}\hat{\boldsymbol{\alpha}} \\ + \tilde{\mathbf{r}}^{\mathrm{T}}\mathbf{C}\hat{\boldsymbol{\alpha}} + \varepsilon - \lambda s + u_{rc}.\end{aligned} \tag{25}$$

To guarantee the stability of the ICSMC system, consider a Lyapunov function candidate in the following form,

$$V_2(t) = \frac{1}{2}s^2 + \frac{1}{2}s_c^2 + \frac{\tilde{\boldsymbol{\alpha}}^{\mathrm{T}}\tilde{\boldsymbol{\alpha}}}{2\eta_\alpha} + \frac{\tilde{\mathbf{c}}^{\mathrm{T}}\tilde{\mathbf{c}}}{2\eta_c} + \frac{\tilde{\sigma}^{\mathrm{T}}\tilde{\boldsymbol{\sigma}}}{2\eta_\sigma} + \frac{\tilde{\mathbf{r}}^{\mathrm{T}}\tilde{\mathbf{r}}}{2\eta_r} \tag{26}$$

where η_α, η_c, η_σ and η_r are positive constants. Differentiating (26) with respect to time and using (25), it is obtained that

$$\begin{aligned}\dot{V}_2(t) &= s\dot{s} + s_c\dot{s}_c + \frac{\tilde{\boldsymbol{\alpha}}^{\mathrm{T}}\dot{\tilde{\boldsymbol{\alpha}}}}{\eta_\alpha} + \frac{\tilde{\mathbf{c}}^{\mathrm{T}}\dot{\tilde{\mathbf{c}}}}{\eta_c} + \frac{\tilde{\boldsymbol{\sigma}}^{\mathrm{T}}\dot{\tilde{\boldsymbol{\sigma}}}}{\eta_\sigma} + \frac{\tilde{\mathbf{r}}^{\mathrm{T}}\dot{\tilde{\mathbf{r}}}}{\eta_r} \\ &= (s + s_c)\left[\tilde{\boldsymbol{\alpha}}^{\mathrm{T}}(\hat{\boldsymbol{\Theta}} - \mathbf{A}^{\mathrm{T}}\hat{\mathbf{c}} - \mathbf{B}^{\mathrm{T}}\hat{\boldsymbol{\sigma}} - \mathbf{C}^{\mathrm{T}}\hat{\mathbf{r}}) + \tilde{\mathbf{c}}^{\mathrm{T}}\mathbf{A}\hat{\boldsymbol{\alpha}} + \tilde{\boldsymbol{\sigma}}^{\mathrm{T}}\mathbf{B}\hat{\boldsymbol{\alpha}} + \tilde{\mathbf{r}}^{\mathrm{T}}\mathbf{C}\hat{\boldsymbol{\alpha}} + \varepsilon - \lambda(s + s_c) + u_{rc}\right] + \frac{\tilde{\boldsymbol{\alpha}}^{\mathrm{T}}\dot{\tilde{\boldsymbol{\alpha}}}}{\eta_\alpha} + \frac{\tilde{\mathbf{c}}^{\mathrm{T}}\dot{\tilde{\mathbf{c}}}}{\eta_c} + \frac{\tilde{\boldsymbol{\sigma}}^{\mathrm{T}}\dot{\tilde{\boldsymbol{\sigma}}}}{\eta_\sigma} + \frac{\tilde{\mathbf{r}}^{\mathrm{T}}\dot{\tilde{\mathbf{r}}}}{\eta_r} \\ &= \tilde{\boldsymbol{\alpha}}^{\mathrm{T}}\left[(s + s_c)(\hat{\boldsymbol{\Theta}} - \mathbf{A}^{\mathrm{T}}\hat{\mathbf{c}} - \mathbf{B}^{\mathrm{T}}\hat{\boldsymbol{\sigma}} - \mathbf{C}^{\mathrm{T}}\hat{\mathbf{r}}) + \frac{\dot{\tilde{\boldsymbol{\alpha}}}}{\eta_\alpha}\right] + \tilde{\mathbf{c}}^{\mathrm{T}}\left[(s + s_c)\mathbf{A}\hat{\boldsymbol{\alpha}} + \frac{\dot{\tilde{\mathbf{c}}}}{\eta_c}\right] + \tilde{\boldsymbol{\sigma}}^{\mathrm{T}}\left[(s + s_c)\mathbf{B}\hat{\boldsymbol{\alpha}} + \frac{\dot{\tilde{\boldsymbol{\sigma}}}}{\eta_\sigma}\right] + \tilde{\mathbf{r}}^{\mathrm{T}}\left[(s + s_c)\mathbf{C}\hat{\boldsymbol{\alpha}} + \frac{\dot{\tilde{\mathbf{r}}}}{\eta_r}\right] \\ &\quad + (s + s_c)(\varepsilon + u_{rc}) - \lambda(s + s_c)^2\end{aligned} \tag{27}$$

Choose the parameter adaptive laws as

$$\dot{\hat{\alpha}} = -\dot{\tilde{\boldsymbol{\alpha}}} = \eta_\alpha(s + s_c)(\hat{\Theta} - \mathbf{A}^{\mathrm{T}}\hat{\mathbf{c}} - \mathbf{B}^{\mathrm{T}}\hat{\boldsymbol{\sigma}} - \mathbf{C}^{\mathrm{T}}\hat{\mathbf{r}}) \tag{28}$$

$$\dot{\hat{\mathbf{c}}} = -\dot{\tilde{\mathbf{c}}} = \eta_c(s + s_c)\mathbf{A}\hat{\boldsymbol{\alpha}} \tag{29}$$

$$\dot{\hat{\sigma}} = -\dot{\tilde{\boldsymbol{\sigma}}} = \eta_\sigma(s + s_c)\mathbf{B}\hat{\boldsymbol{\alpha}} \tag{30}$$

$$\dot{\hat{\mathbf{r}}} = -\dot{\tilde{\mathbf{r}}} = \eta_r(s + s_c)\mathbf{C}\hat{\boldsymbol{\alpha}} \tag{31}$$

and the robust compensator is chosen as

$$u_{rc} = -\frac{\delta^2 + 1}{2\delta^2}(s + s_c) \tag{32}$$

where δ is a small positive constant, then (27) can be obtained

$$\begin{aligned}\dot{V}_2(t) &\le (s + s_c)\varepsilon - \left(\frac{\delta^2 + 1}{2\delta^2}\right)(s + s_c)^2 \\ &= -\frac{1}{2}(s + s_c)^2 - \frac{1}{2}\left(\frac{s + s_c}{\delta} - \delta\varepsilon\right)^2 + \frac{1}{2}\delta^2\varepsilon^2 \\ &\le -\frac{1}{2}(s + s_c)^2 + \frac{1}{2}\delta^2\varepsilon^2.\end{aligned} \tag{33}$$

Integrating the inequality (33) in $[0, t_f]$, it is obtained that

$$V_2(t_f) - V_2(0) \le -\frac{1}{2}\int_0^{t_f}(s + s_c)^2 dt + \frac{1}{2}\delta^2\int_0^{t_f}\varepsilon^2 dt. \tag{34}$$

If the system starts with initial condition $V_2(0) = 0$ and $V_2(t_f) \ge 0$, the L_2-gain inequality can be written as

$$\frac{||s + s_c||_{L_2}^2}{||\varepsilon||_{L_2}^2} \le \delta^2\big|_{V_2(0)=0} \tag{35}$$

where $||s + s_c||_{L_2}^2 = \int_0^{t_f}(s + s_c)^2 dt$ and $||\varepsilon||_{L_2}^2 = \int_0^{t_f}\varepsilon^2 dt$. The attenuation constant δ can be specified by the designer to achieve the desired attenuation ratio between $||s + s_c||$ and $||\varepsilon||$ [31–33]. If $V_2(0) \neq 0$, the L_2-gain inequality (34) can be given as

$$||(s + s_c)||_{L_2} \le \delta||\varepsilon||_{L_2} + r_{bias} \tag{36}$$

where $r_{bias} = \sqrt{2V_2(0)}$. It can be found that $\delta||\varepsilon||_{L_2}$ provides an attraction region around the origin of the state space and its area is controlled by δ. It is implied that the proposed ICSMC system is finite L_2-gain stable [31–33].

Dead-zone parameter modification

In the parameter adaptive laws (28)–(31), $\dot{\boldsymbol{\alpha}}^* = \dot{\mathbf{c}}^* = \dot{\boldsymbol{\sigma}}^* = \dot{\mathbf{r}}^* = 0$ is necessary. Since the existence of approximation errors, the learning laws could make the parameters drift constantly and the system might become unstable eventually. To avoid the parameters-drift problem, a dead-zone parameter modification is proposed in which the parameter tuning process will stop when tracking performance index is smaller than performance threshold. Assume that the trajectory is cyclic and the cycle time is given by t_Δ, thus the tracking performance index I_{index} during the last cycle time is defined as

$$I_{index} = \int_{t_f - t_\Delta}^{t_f} |e|^2 dt \tag{37}$$

In this paper, the parameter adaptive laws in (28)–(31) can be modified as following

$$\dot{\hat{\boldsymbol{\alpha}}} = \begin{cases} \eta_\alpha(s+s_c)(\hat{\boldsymbol{\Theta}} - \mathbf{A}^{\mathrm{T}}\hat{\mathbf{c}} - \mathbf{B}^{\mathrm{T}}\hat{\boldsymbol{\sigma}} - \mathbf{C}^{\mathrm{T}}\hat{\mathbf{r}}), & \text{if } I_{index} \geq I_{th} \\ 0, & \text{if } I_{index} < I_{th} \end{cases} \tag{38}$$

$$\dot{\hat{\mathbf{c}}} = \begin{cases} \eta_c(s+s_c)\mathbf{A}\hat{\boldsymbol{\alpha}}, & \text{if } I_{index} \geq I_{th} \\ 0, & \text{if } I_{index} < I_{th} \end{cases} \tag{39}$$

$$\dot{\hat{\boldsymbol{\sigma}}} = \begin{cases} \eta_\sigma(s+s_c)\mathbf{B}\hat{\boldsymbol{\alpha}}, & \text{if } I_{index} \geq I_{th} \\ 0, & \text{if } I_{index} < I_{th} \end{cases} \tag{40}$$

$$\dot{\hat{\mathbf{r}}} = \begin{cases} \eta_\sigma(s+s_c)\mathbf{C}\hat{\boldsymbol{\alpha}}, & \text{if } I_{index} \geq I_{th} \\ 0, & \text{if } I_{index} < I_{th} \end{cases} \tag{41}$$

where I_{th} is a performance threshold. The performance threshold I_{th} can be selected according to the desired control performance. The modification parameter adaptive laws (38)–(41) mean that if the performance threshold I_{th} is properly selected as $I_{index} < I_{th}$ where the learning is sufficient during the last cycle time, then the parameter learning will be stopped. By choosing $I_{th} = 0$ means that the parameter adaptation is without dead-zone parameter modification which can be found in many published papers. If I_{th} is chosen smaller, the better learning performance of the RNFIN can be achieved, but this may lead to an overtraining problem. To attack this problem, a first-order transfer function of the following form is designed for the performance threshold as

$$\dot{I}_{th} = \tau(I_{final} - I_{th}) \tag{42}$$

where τ is a designed time constant. Eq. (42) shows that the performance threshold starts $I_{th} = 0$ and rises to I_{final}. Thus, the time-varying performance threshold not only can achieve better learning performance but also can avoid an overtraining problem. When the learning stopped, it can be assumed that the learning performance is reached. In this case, we can obtain that $\hat{\boldsymbol{\alpha}} = \boldsymbol{\alpha}^*$, $\hat{\mathbf{c}} = \mathbf{c}^*$, $\hat{\boldsymbol{\sigma}} = \boldsymbol{\sigma}^*$ and $\hat{\mathbf{r}} = \mathbf{r}^*$ (or $\tilde{\boldsymbol{\alpha}} = 0$, $\tilde{\mathbf{c}} = 0$, $\tilde{\boldsymbol{\sigma}} = 0$ and $\tilde{\mathbf{r}} = 0$) and $\dot{\boldsymbol{\alpha}}^* = \dot{\mathbf{c}}^* = \dot{\boldsymbol{\sigma}}^* = \dot{\mathbf{r}}^* = 0$. To guarantee the stability of the ICSMC system with dead-zone parameter modification, the Lyapunov function candidate is defined as (26). Differentiating (26) with respect to time and using (38)–(41), we can obtain as

$$\dot{V}_2(t) = (s+s_c)(\varepsilon + u_{rc}). \tag{43}$$

The robust compensator is designed as (32), thus (43) is exactly the same as (33). Therefore, the proposed ICSMC system without parameter adaptations still has the finite L_2-gain property when tracking performance index is smaller than performance threshold.

Simulation and experimental results

Example 1: One-link robotic manipulator

The one-link robotic manipulator is studied in this work. The system dynamics of the one-link robotic manipulator is referred as follows [34]

$$\ddot{q} = \frac{-b\dot{q} - mlg_V\cos(q)}{ml^2} + \frac{1}{ml^2}u \tag{44}$$

where l is the link length, m is the mass, and q is the angular position with initial condition $q(0) = \dot{q}(0) = 0$. It is assumed that the parameters are given by $m = l = b = g_V = 1$. To illustrate the effectiveness of the ICSMC system, an external disturbance $d(t) = 5\sin(2t)\cos(8t)$ is generated starting from $t = 20$ s. To show the effectiveness of the ICSMC system, a comparison between the robust sliding-mode control (RSMC) system [35] and the proposed ICSMC system is made.

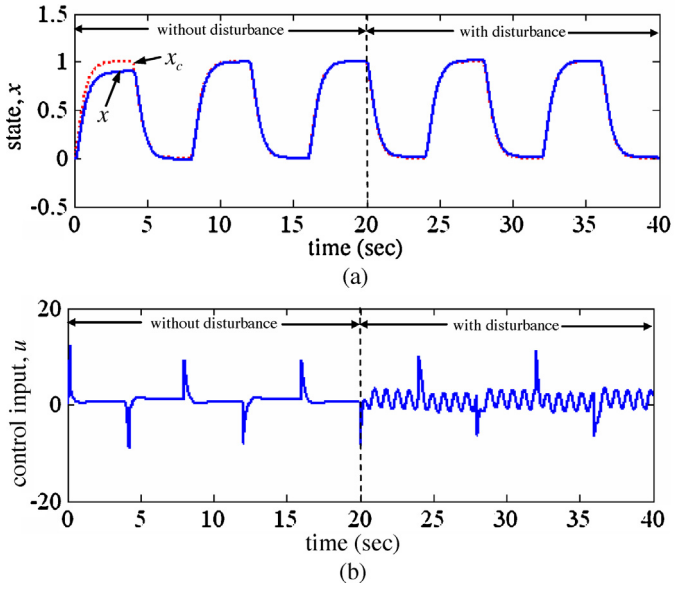

Fig. 3. Simulation results of the RSMC system in Ref. [35] for small learning rate.

First, the RSMC system [35] is applied to the one-link robotic manipulator. A bound estimation law is utilized to estimate the bound of the model uncertainties and external disturbances in real time. The simulation results of the RSMC system for small learning rates are shown in Fig. 3. The tracking response of the state q is shown in Fig. 3(a) and the control input u is shown in Fig. 3(b). The simulation results show that the favorable control performance can be achieved after the learning process for the controller parameters. However, this results in slow learning speed. On the contrary, the simulation results of the RSMC system for large learning rates are shown in Fig. 4. The tracking response of the state q is shown in Fig. 4(a) and the control input u is shown in Fig. 4(b). The simulation results show that the chattering control input are resulted due to the bound of the system uncertainties overtraining problem occurring.

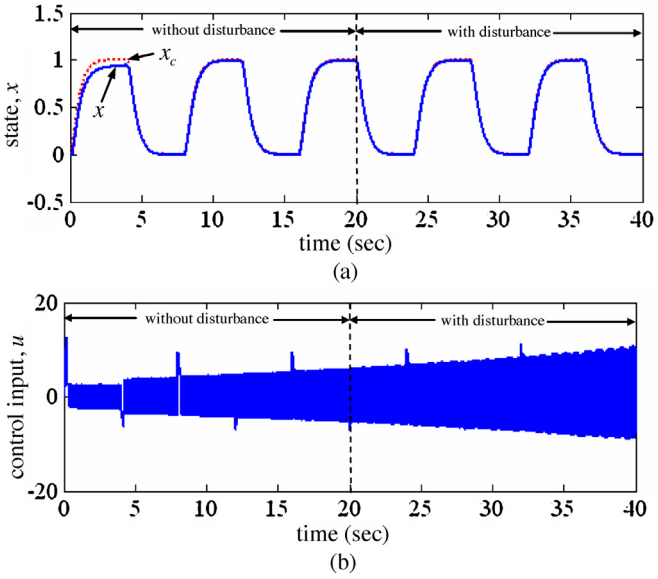

Fig. 4. Simulation results of the RSMC system in Ref. [35] for large learning rate.

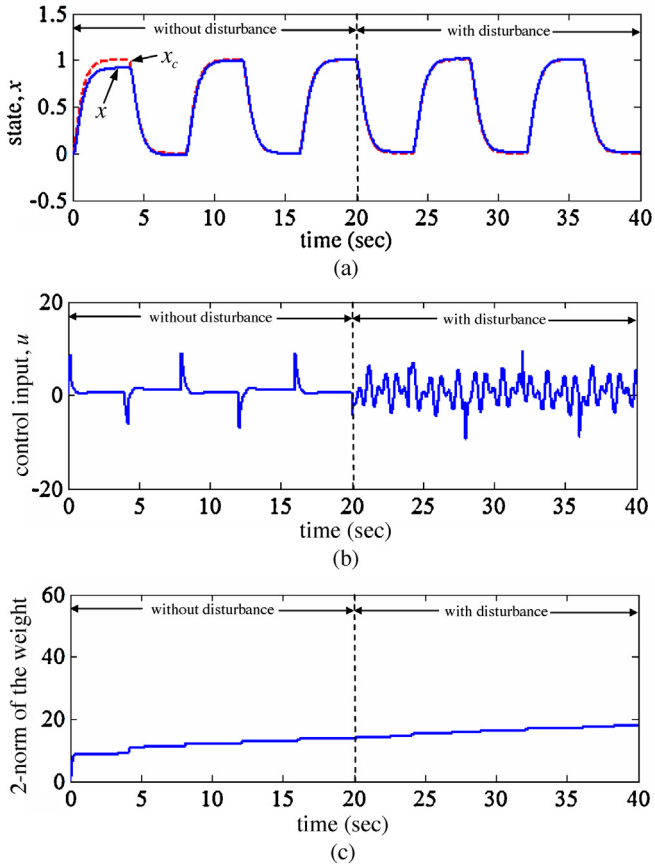

Fig. 5. Simulation results of the ICSMC system without dead-zone parameter modification for small learning rate.

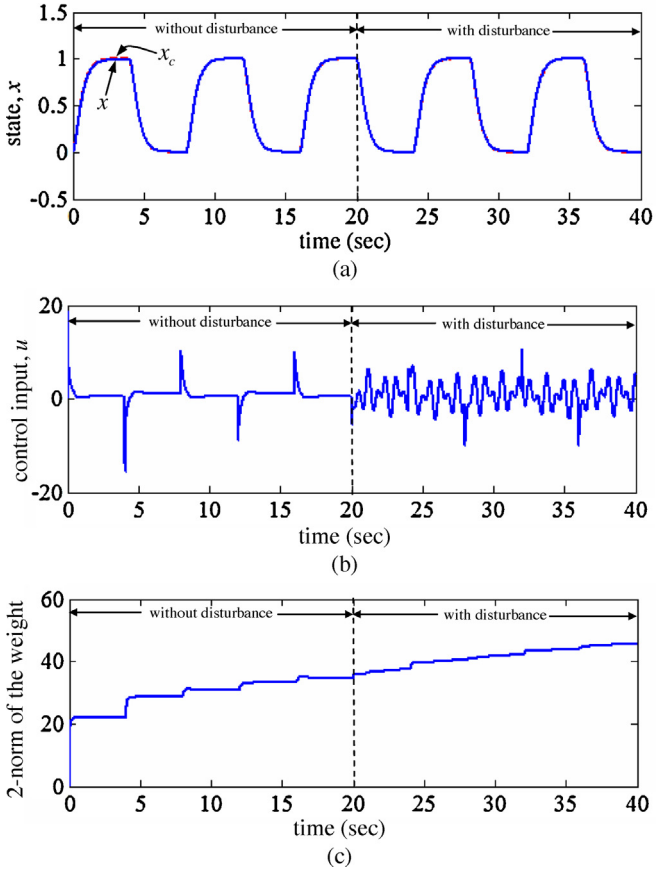

Fig. 6. Simulation results of the ICSMC system without dead-zone parameter modification for large learning rate.

Then, the proposed ICSMC system is applied to the one-link robotic manipulator again. The parameters of the ICSMC system are selected as $\lambda = 0.2$, $\eta_\alpha = \eta_r = 20$, $\eta_c = \eta_\sigma = 1$, and $\delta = 0.5$. These parameters are chosen to achieve a desired system response. By choosing the value of λ properly, the desired system specifications can be easily ensured in (4). Next, the parameters η_α, η_r, η_c and η_σ are the leaning rates of the tuning parameters in the RNFIN. Generally speaking, if the leaning rates are chosen to be small, parameter convergence can be easily achieved; however, this will result in slow learning speed. On the contrary, if the leaning rates are too large, the learning speed can be increased; yet, the system may become unstable. Moreover, the prescribed attenuation constant δ will influence the convergent speed of the tracking error. A better tracking performance of the ICSMC system can be achieved if δ is chosen smaller, but this will lead to large control signal. A better learning performance of RNFIN can be achieved, if I_{th} is chosen smaller, but this may lead to an overtraining problem.

The simulation results of the ICSMC system without dead-zone parameter modification for small learning rates are shown in Fig. 5. The tracking response of the state q is shown in Fig. 5(a), the control input u is shown in Fig. 5(b), and the learning histories of the network parameter is shown in Fig. 5(c). The simulation results show that the favorable control performance can be achieved after the learning process for the controller parameters. It can be seen that the constant optimal controller parameters cannot be achieved. But, the derivatives of the optimal parameters are assumed to be zeros in the derivation of the parameter learning process.

On the contrary, the ICSMC system without the dead-zone parameter modification for large learning rates is applied to the one-link robotic manipulator. The parameters of the ICSMC system are selected as $\lambda = 0.2$, $\eta_\alpha = \eta_r = 1000$, $\eta_c = \eta_\sigma = 10$, and $\delta = 0.5$. The simulation results of the ICSMC system without dead-zone parameter modification for large learning rates are shown in Fig. 6. The tracking response of the state q is shown in Fig. 6(a), the control input u is shown in Fig. 6(b), and the learning histories of the network parameter is shown in Fig. 6(c). The simulation results show that the better control performance can be achieved but the constant optimal controller parameters cannot be achieved. When these adjustable controller parameters are not fixed in a finite time, it is difficult to say that the derivatives of the optimal parameter vectors are zero.

Next, the proposed ICSMC system with dead-zone parameter modification for small learning rates is applied. The parameters of the ICSMC system are selected as $\tau = 1$, $I_{final} = 20$, $\lambda = 0.2$, $\eta_\alpha = \eta_r = 20$, $\eta_c = \eta_\sigma = 1$, and $\delta = 0.5$. These parameters are chosen to achieve a desired system response. The simulation results of the ICSMC system with dead-zone parameter modification for small learning rates are shown in Fig. 7. The tracking response of state q is shown in Fig. 7(a), the control input u is shown in Fig. 7(b), and the learning histories of the network parameter is shown in Fig. 7(c). Fig. 7(c) shows that the parameter adaptation is stopped ($\dot{\boldsymbol{\alpha}}^* = \dot{\mathbf{c}}^* = \dot{\boldsymbol{\sigma}}^* = \dot{\mathbf{r}}^* = 0$) in a finite time. The simulation results verify that not only robust tracking performance can be achieved, but also the parameter overtraining problem can be avoided by the proposed dead-zone parameter modification.

Finally, the proposed ICSMC system with dead-zone parameter modification for large learning rates is applied, and the parameters of the ICSMC system are selected as $\tau = 1$, $I_{final} = 20$, $\lambda = 0.2$, $\eta_\alpha = \eta_r = 1000$, $\eta_c = \eta_\sigma = 10$, and $\delta = 0.5$. The simulation results of the ICSMC system with dead-zone parameter modification for large learning rates are shown in Fig. 8. The tracking response of state q is shown in Fig. 8(a), the control input u is shown in Fig. 8(b),

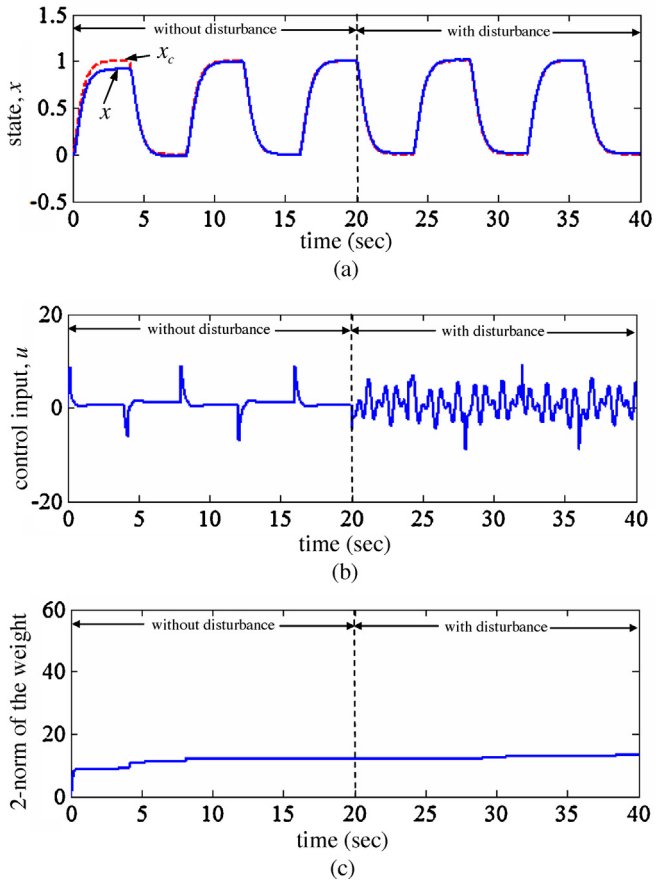

Fig. 7. Simulation results of the ICSMC system with dead-zone parameter modification for small learning rate.

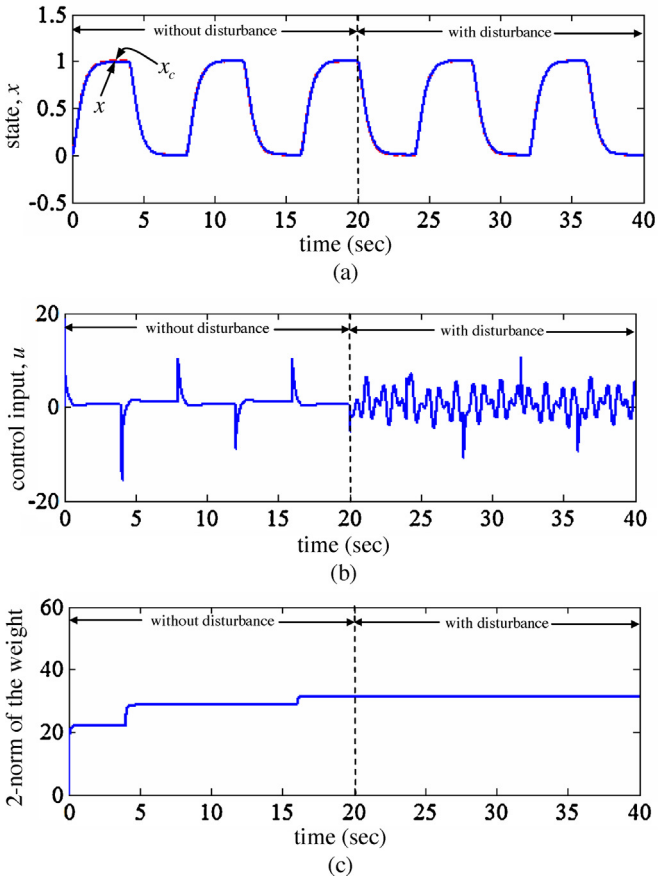

Fig. 8. Simulation results of the ICSMC system with dead-zone parameter modification for large learning rate.

and the learning histories of the network parameter is shown in Fig. 8(c). Figure 8(c) shows that the parameter adaptation is stopped ($\dot{\boldsymbol{\alpha}}^* = \dot{\mathbf{c}}^* = \dot{\boldsymbol{\sigma}}^* = \dot{\mathbf{r}}^* = 0$) in a finite time. The simulation results show that the proposed dead-zone parameter modification indeed not only can achieve favorable tracking performances via the finite L_2-gain property but also can reduce the parameter overtraining problem.

Example 2: DC motor driver

Because a microcontroller would be more suitable for the real-time control applications than a PC and a DSP, this paper proposed a microcontroller-based experimental setup for a DC motor driver as shown in Fig. 9 for possible low-cost and high-performance industrial applications. In this paper, the control board is 32-bit Arduino DUE [36]. Using an Arduino DUE simplifies the amount of hardware and software development which we need to do in order to get a system running. On the hardware side, the microcontroller (Atmel® SAM3X8E) runs at 84 MHz and features 512 KB of Flash and 100 KB of SRAM based on the ARM® Cortex™-M3 processor. The Arduino DUE board already has the power and reset circuitry setup as well as circuitry to program and communicate with the microcontroller over USB. In addition, the I/O pins of the microcontroller are typically already fed out to sockets/headers for easy access. On the software side, the open-source Arduino Software Development Environment makes it easy to write code and upload it to the Arduino DUE board. It runs on Windows, Mac OS X, and Linux. Arduino also provides a number of libraries to make programming the microcontroller easier.

L298 motor driver is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads. HCTL-2032 decoder performs the quadrature decoder, 32-bit counter and bus interface function. In this paper, L298 motor

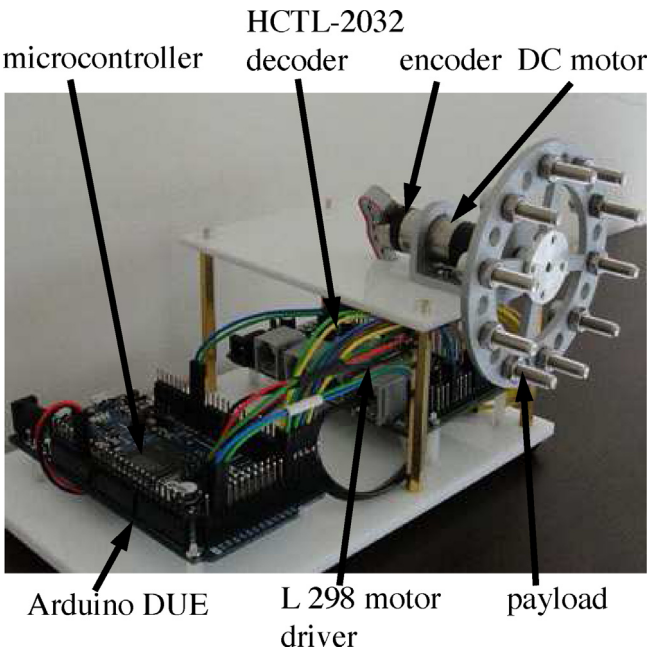

Fig. 9. Microcontroller-based experimental setup for a DC motor driver.

driver and HCTL-2032 decoder are used to improve system performance in digital closed loop motion control. Meanwhile, the DC motor driver can be represented in the following form [37,38]

$$\ddot{\theta} = -\left(\frac{B}{J} + \frac{K_t K_b}{JR_a}\right)\dot{\theta} - \frac{T_l}{J} + \frac{K_t}{JR_a}u \tag{45}$$

where θ is the rotor position, J is the moment of inertia, B is the damping coefficient, T_l denotes the external disturbance torque, R_a is the motor resistance, K_t is the torque constant, K_b is the back electromotive force coefficient, and u is a control input. To investigate the effectiveness of the proposed ICSMC system, two conditions are tested. One is the nominal condition and the other one is the payload condition that has one disk as the payload.

First, the proposed ICSMC system without dead-zone parameter modification for small learning rates is applied to the DC motor driver. The parameters are selected as $\lambda = 1$, $\eta_\alpha = \eta_r = 8$, $\eta_c = \eta_\sigma = 1$, and $\delta = 0.8$. These parameters are selected through some trials. The experimental results of the ICSMC system without dead-zone parameter modification for small learning rates are shown in Fig. 10. The tracking responses of the rotor position θ are shown in Fig. 10(a) and (d), the control inputs u are shown in Fig. 10(b) and (e), and the learning histories of the network parameter are shown in Fig. 10(c) and (f) for nominal condition and payload condition, respectively. The experimental results show that favorable control performance can be achieved with a payload variation after the controller learning process. However, it can be seen that the constant optimal controller parameters ($\lim_{t\to t_f}\tilde{\boldsymbol{\alpha}} = 0$, $\lim_{t\to t_f}\tilde{\mathbf{c}} = 0$, $\lim_{t\to t_f}\tilde{\boldsymbol{\sigma}} = 0$, and $\lim_{t\to t_f}\tilde{\mathbf{r}} = 0$) are not reachable.

To achieve a better control performance, the proposed ICSMC system without dead-zone parameter modification for large learning rates is applied. The parameters are selected as $\lambda = 1$, $\eta_\alpha = \eta_r = 200$, $\eta_c = \eta_\sigma = 10$, and $\delta = 0.8$. The experimental results of the ICSMC system without dead-zone parameter modification for large learning rates are shown in Fig. 11. The tracking responses of the rotor position θ are shown in Fig. 11(a) and (d), the control inputs u are shown in Fig. 11(b) and (e), and the learning histories of the network parameter are shown in Fig. 11(c) and (f) for nominal condition and payload condition, respectively. The experimental results show that favorable control performance can be achieved after the controller learning process. However, the chattering control input are resulted due to the controller parameters overtraining problem occurring.

Next, the proposed ICSMC system with dead-zone parameter modification for small learning rates is applied again. The parameters are selected as $\tau = 1$, $I_{final} = 10$, $\lambda = 1$, $\eta_\alpha = \eta_r = 8$, $\eta_c = \eta_\sigma = 1$, and $\delta = 0.8$. The experimental results of the ICSMC system with dead-zone parameter modification for small learning rates are shown in Fig. 12. The tracking responses of the rotor position θ are shown in Fig. 12(a) and (d), the control inputs u are shown in Fig. 12(b) and (e), and the learning histories of the network parameter are shown in Fig. 12(c) and (f) for nominal condition and payload condition, respectively. Fig. 12(c) and (f) show that parameter learning is stopped after 8.24 s and 3.18 s for the nominal condition and the payload condition, respectively. The experimental results can be seen that not only robust tracking performance can be achieved but also the parameter adaptation is stopped.

Finally, the proposed ICSMC system with dead-zone parameter modification for large learning rates is applied to the DC motor driver. The parameters are selected as $\tau = 1$, $I_{final} = 10$, $\lambda = 1$, $\eta_\alpha = \eta_r = 200$, $\eta_c = \eta_\sigma = 10$, and $\delta = 0.8$. The experimental results of the ICSMC system with dead-zone parameter modification for large learning rates are shown in Fig. 13. The tracking responses of the rotor position θ are shown in Fig. 13(a) and (d), the control inputs u are shown in Fig. 13(b) and (e), and the learning histories of

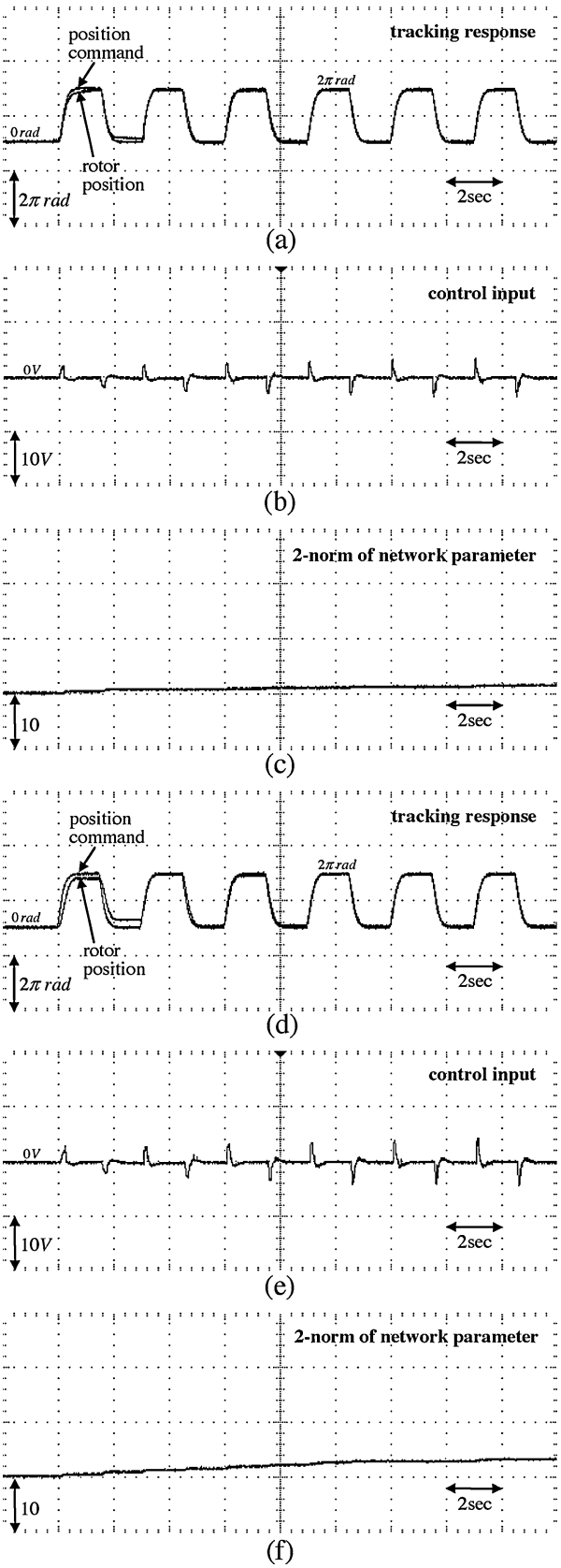

Fig. 10. Experimental results of the ICSMC system without dead-zone parameter modification for small learning rate.

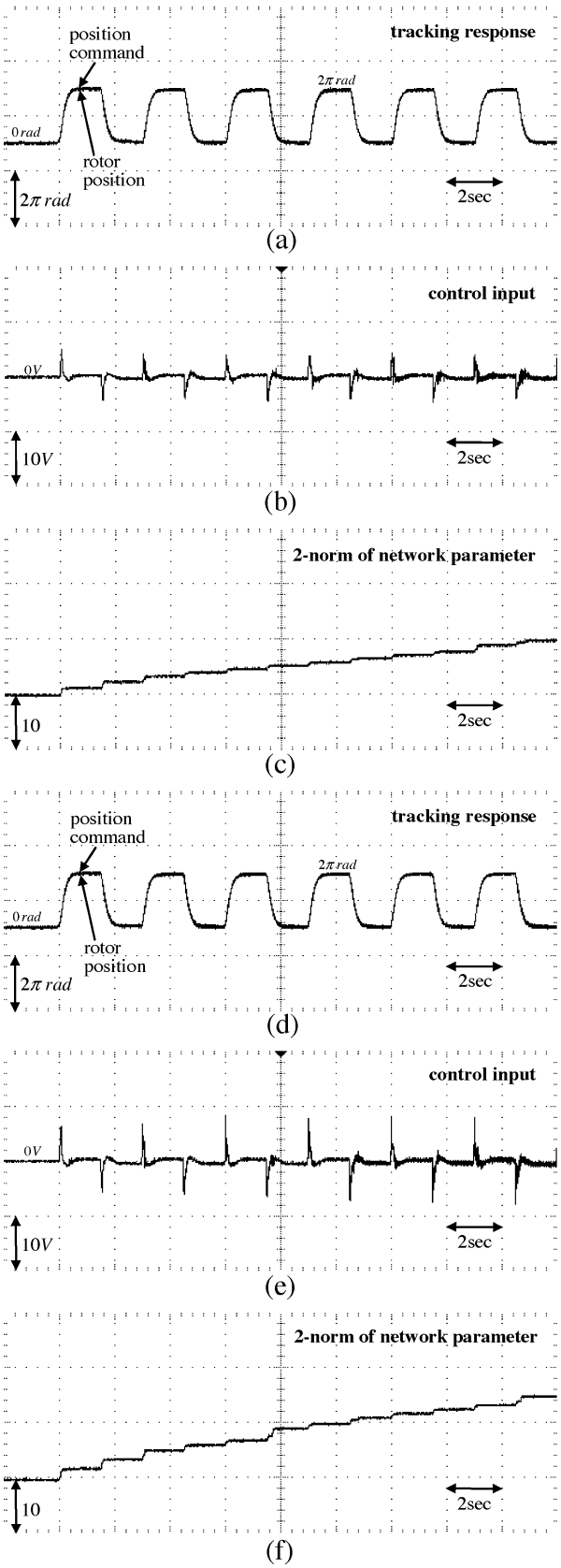

Fig. 11. Experimental results of the ICSMC system without dead-zone parameter modification for large learning rate.

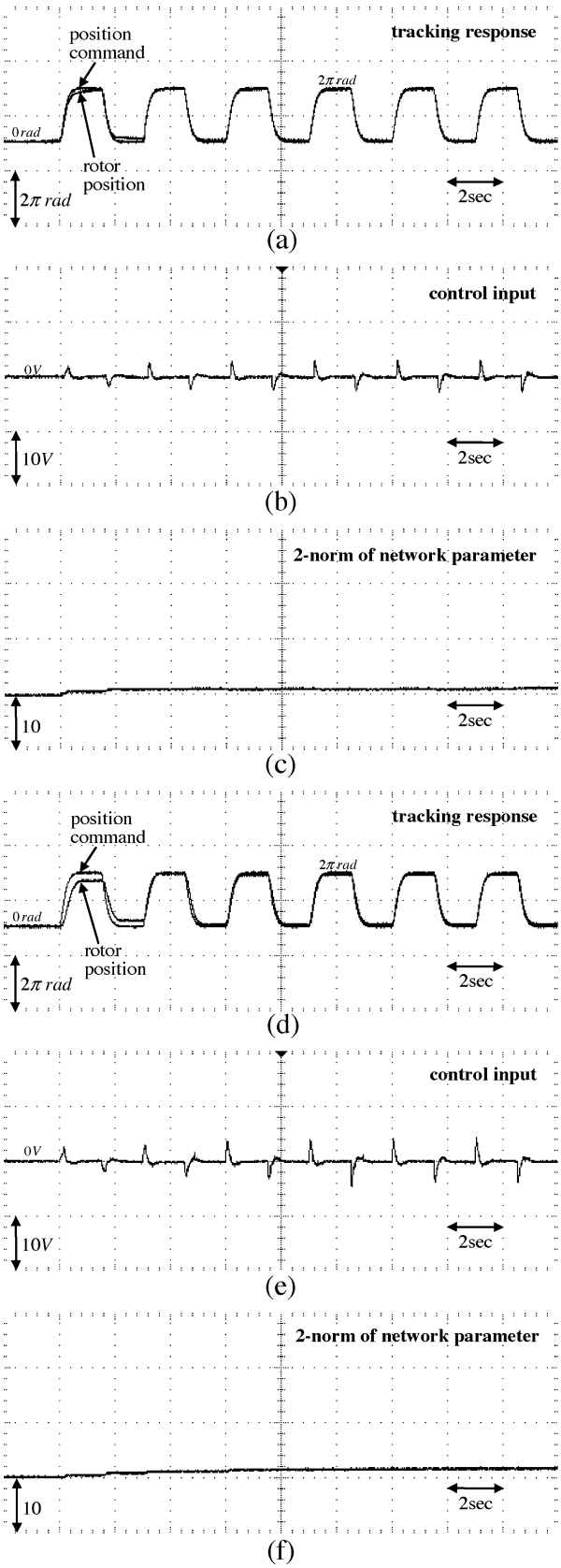

Fig. 12. Experimental results of the ICSMC system with dead-zone parameter modification for small learning rate.

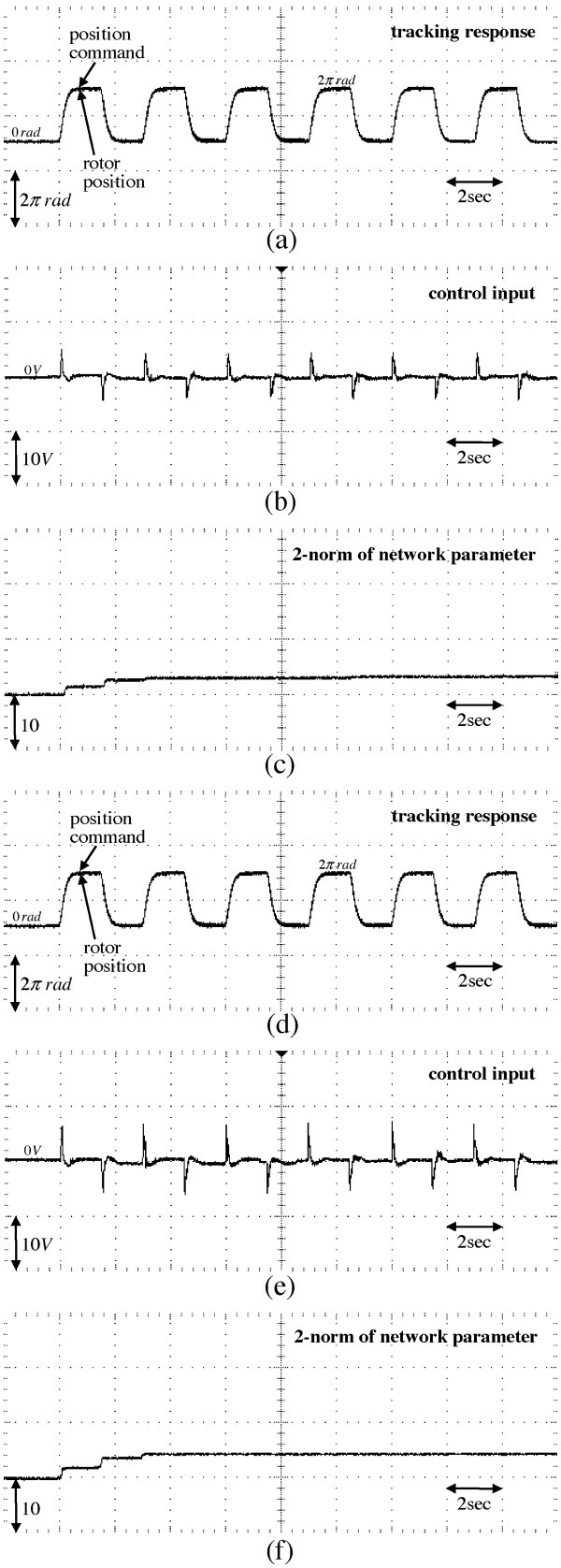

Fig. 13. Experimental results of the ICSMC system with dead-zone parameter modification for large learning rate.

the network parameter are shown in Fig. 13(c) and (f) for nominal condition and payload condition, respectively. Fig. 13(c) and (f) show that parameter learning is stopped after 5.28 s and 1.58 s for the nominal condition and the payload condition, respectively. It demonstrated that the proposed ICSMC approach with dead-zone parameter indeed not only can have favorable tracking performances via the finite L_2-gain property and stable learning results but also can reduce the parameter overtraining problem.

Conclusions

This paper proposes an ICSMC system which is composed of a computed controller and a robust compensator. The computed controller uses a RNFIN to online approximate the unknown system term and the robust compensator is designed to eliminate the effect of the approximation error. By employing the L_2 control theory, it is possible to attenuate the effects of the approximation errors between the RNFIN and the system dynamic to a prescribed level. To avoid the drift, parameters are limited in a given region, and a dead-zone parameter modification in which the parameter tuning process will stop when a learning performance index is smaller than a pre-specified threshold. Finally, some simulation and experimental results demonstrate that the proposed ICSMC approach indeed can have satisfactory tracking performances via the finite L_2-gain property and stable learning results. The proposed ICSMC system is only applicable to a second-order nonlinear system. The proposed ICSMC system can be easily extended to other nonlinear control problem liking as robotic manipulators and linear ultrasonic motors.

Acknowledgments

The authors are grateful to the associate editor and 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 100-2628-E-032-003.

References

[1] V.I. Utkin, Sliding Modes and theirs Applications in Variable Structure Systems, MIR, Moscow, 1978.
[2] W. Chen, M. Saif, Output feedback controller design for a class of MIMO nonlinear systems using high-order sliding-mode differentiators with application to a laboratory 3-D crane, IEEE Trans. Ind. Electron. 55 (11) (2008) 3985–3997.
[3] T.R. Oliveira, A.J. Peixoto, L. Hsu, Sliding mode control of uncertain multivariable nonlinear systems with unknown control direction via switching and monitoring function, IEEE Trans. Autom. Control 55 (4) (2010) 1028–1034.
[4] S.Z. Chen, N.C. Cheung, K.C. Wong, J. Wu, Integral variable structure direct torque control of doubly fed induction generator, IET Renew. Power Gener. 5 (1) (2011) 18–25.
[5] M. Nayeripour, M.R. Narimani, T. Niknam, S. Jam, Design of sliding mode controller for UPFC to improve power oscillation damping, Appl. Soft Comput. 11 (8) (2011) 4766–4772.
[6] C.L. Chen, C.W. Chang, H.T. Yau, Terminal sliding mode control for aeroelastic systems, Nonlinear Dyn. 70 (3) (2012) 2015–2026.
[7] C.M. Lin, C.F. Hsu, Neural network hybrid control for antilock braking systems, IEEE Trans. Neural Netw. 14 (2) (2003) 351–359.
[8] C.M. Lin, C.F. Hsu, Supervisory recurrent fuzzy neural network control of wing rock for slender delta wings, IEEE Trans. Fuzzy Syst. 12 (5) (2004) 733–742.
[9] S. Mondal, C. Mahanta, Adaptive second-order sliding mode controller for a twin rotor multi-input-multi-output system, IET Control Theory Appl. 6 (14) (2012) 2157–2167.
[10] A.J. Koshkouei, K.J. Burnham, A.S.I. Zinober, Dynamic sliding mode control design, IEE Control Theory Appl. 152 (4) (2005) 392–396.
[11] J.Z. Tsai, C.F. Hsu, C.J. Chiu, K.L. Peng, FPGA-based adaptive dynamic sliding-mode neural control for a BLDC motor, Asian J. Control 13 (6) (2011) 845–857.
[12] F.J. Lin, J.C. Hwang, P.H. Chou, Y.C. Hung, FPGA-based intelligent-complementary sliding-mode control for PMLSM servo-drive system, IEEE Trans. Power Electron. 25 (10) (2010) 2573–2587.
[13] C.F. Hsu, Adaptive neural complementary sliding-mode control via functional-linked wavelet neural network, Eng. Appl. Artif. Intell. 26 (4) (2013) 1221–1229.
[14] C.K. Lin, Radial basis function neural network-based adaptive critic control of induction motors, Appl. Soft Comput. 11 (3) (2011) 3066–3074.

[15] H.C. Lu, M.H. Chang, C.H. Tsai, Adaptive self-constructing fuzzy neural network controller for hardware implementation of an inverted pendulum system, Appl. Soft Comput. 11 (5) (2011) 3962–3975.
[16] I. Chairez, Differential neuro-fuzzy controller for uncertain nonlinear systems, IEEE Trans. Fuzzy Syst. 21 (2) (2013) 369–384.
[17] C.F. Hsu, Nonlinear system control using a self-organizing functional-linked neuro-fuzzy network, Nonlinear Dyn. 73 (3) (2013) 1631–1643.
[18] L.X. Wang, Adaptive Fuzzy Systems and Control: Design and Stability Analysis, Prentice-Hall, Englewood Cliffs, NJ, 1994.
[19] W.Y. Wang, Y.G. Leu, C.C. Hsu, Robust adaptive fuzzy-neural control of nonlinear dynamical systems using generalized projection update law and variable structure controller, IEEE Trans. Syst. Man Cybern. B 31 (1) (2001) 140–147.
[20] Y.C. Hsueh, S.F. Su, C.W. Tao, C.C. Hsiao, Robust L_2-gain compensative control for direct-adaptive fuzzy-control-system design, IEEE Trans. Fuzzy Syst. 18 (4) (2010) 661–673.
[21] K. Nouri, R. Dhaouadi, N.B. Braiek, Adaptive control of a nonlinear dc motor drive using recurrent neural networks, Appl. Soft Comput. 8 (1) (2008) 371–382.
[22] C.H. Chen, C.F. Hsu, Recurrent wavelet neural backstepping controller design with a smooth compensator, Neural Comput. Appl. 19 (7) (2010) 1089–1100.
[23] C.H. Chiu, The design and implementation of a wheeled inverted pendulum using an adaptive output recurrent cerebellar model articulation controller, IEEE Trans. Ind. Electron. 57 (5) (2010) 1814–1822.
[24] Y. Pan, J. Wang, Model predictive control of unknown nonlinear dynamical systems based on recurrent neural networks, IEEE Trans. Ind. Electron. 59 (8) (2012) 3089–3101.
[25] K.K. Shyu, W.J. Liu, K.C. Hsu, Design of large-scale time-delayed systems with dead-zone input via variable structure control, Automatica 41 (7) (2005) 1239–1246.
[26] M.P. Aghababa, Adaptive control for electromechanical systems considering dead-zone phenomenon, Nonlinear Dyn. 75 (1) (2014) 157–174.
[27] J.J.E. Slotine, W.P. Li, Applied Nonlinear Control, Prentice Hall, Englewood Cliffs, NJ, 1991.
[28] C.F. Juang, C.H. Hsu, Temperature control by chip-implemented adaptive recurrent fuzzy controller designed by evolutionary algorithm, IEEE Trans. Circuit Syst. 52 (11) (2005) 2376–2384.
[29] C.F. Juang, J.S. Chen, Water bath temperature control by a recurrent fuzzy controller and its FPGA implementation, IEEE Trans. Ind. Electron. 53 (3) (2006) 941–949.
[30] C.F. Hsu, Adaptive backstepping Elman-based neural control for unknown nonlinear systems, Neurocomputing 136 (20) (2014) 170–179.
[31] C.K. Lin, H^{∞} reinforcement learning control of robot manipulators using fuzzy wavelet networks, Fuzzy Sets Syst. 160 (12) (2009) 1765–1786.
[32] Y.F. Peng, Development of robust intelligent tracking control system for uncertain nonlinear systems using H^{∞} control technique, Appl. Soft Comput. 11 (3) (2011) 3135–3146.
[33] C.F. Hsu, Adaptive dynamic CMAC neural control of nonlinear chaotic systems with L_2 tracking performance, Eng. Appl. Artif. Intell. 25 (5) (2012) 997–1008.
[34] T.F. Wu, P.S. Tsai, F.R. Chang, L.S. Wang, Adaptive fuzzy CMAC control for a class of nonlinear systems with smooth compensation, IEE Control Theory Appl. 153 (6) (2006) 647–657.
[35] M.P. Aghababa, H.P. Aghababa, Chaos suppression of a class of unknown uncertain chaotic systems via single input, Commun. Nonlinear Sci. Numer. Simul. 17 (9) (2012) 3533–3538.
[36] http://www.arduino.cc/
[37] C.F. Hsu, B.K. Lee, FPGA-based adaptive PID control of a DC motor driver via sliding-mode approach, Expert Syst. Appl. 38 (9) (2011) 11866–11872.
[38] C.F. Hsu, Intelligent tracking control of a DC motor driver using self-organizing TSK type fuzzy neural networks, Nonlinear Dyn. 67 (1) (2012) 587–600.