

Adaptive Dual Network Design for a Class of SIMO Systems with Nonlinear Time-variant Uncertainties

LIU Bo¹ HE Hai-Bo^{1,2} CHEN Sheng¹

Abstract A novel adaptive dual network design consisting of a rough adjustment network (RAN) and a fine adjustment network (FAN) is proposed to eliminate the unknown time-variant uncertainties of servo system. To accomplish this objective, a RAN is proposed based on the combination of sliding mode control, function approximation, and error compensation technique. Then, an FAN is proposed to compensate the tracking error. In our current design, the FAN includes a critic network based on a neural network model and a prediction network based on an online curve fitting scheme. Theoretical analysis followed by detailed design strategies are presented in this work. Simulation results and comparative study of this method with those of existing approaches demonstrate the effectiveness of the proposed adaptive dual network design for position tracking.

Key words Adaptive control, critic network, sliding mode control (SMC), neural network

A long-lasting focus on the electromechanical system is how to design a simple and effective controller for such nonlinear systems. This problem is deeply rooted in the nonlinearity and time variance of the systems due to the limitations of modern modeling theory. Therefore, most of these problems are too complicated to be solved by the traditional linear control methods. Furthermore, in addition to the desired robustness to the uncertainty of linear model parameters, the controller must be able to adapt to the time-variant uncertainty. However, because uncertainties are usually nonsmooth and nonlinear, the conventional approaches for such challenging problems usually fail or far from satisfactory. Therefore, time-variant electric-mechanical system remains a great challenge to the research community^[1].

Generally speaking, adaptive control systems include two closed loops: the system-controller loop and the adaptive adjustment loop^[2-3]. The breakthroughs in nonlinear control, especially the backstepping^[4-5] and linearization techniques^[6-7], have solved a class of problems with nonlinear time-invariant uncertainties. Sliding mode control (SMC)^[2, 8], or variable structure control (VSC), because of its excellent robustness, is often applied in systems with uncertainties defined in the compact set, e.g., the electromechanical system with uncertainties. Meanwhile, with the capacity to learn and approximate nonlinear functions, conventional intelligence methods, such as artificial neural networks (ANNs), are widely used in situations where system identification cannot be implemented successfully, or the system model is highly time-variant with unknown uncertainties^[9-11]. The theorem that neural networks can approximate any rational function is the main reason of its wide applications in highly uncertain systems^[12-13]. A combination of function approximation, adaptive control mechanisms, and Lyapunov design might have positive impacts on the nonlinear time-variant electric-mechanical real-time systems.

Huang et al. proposed a novel approach to deal with nonlinear systems with a class of time-variant uncertainties with unknown boundaries, namely the uncertainty can

be approximated by finite Fourier series^[14-15]. The adaptive law based on Lyapunov approach was presented to update the coefficients of the series^[16]. In [17], an adaptive SMC controller applied to active suspension systems with time-variant loadings was proposed, in which the system uncertainty can be lumped into two unknown time-variant functions if such functions can be approximated by Fourier series.

In this paper, we propose a dual network design approach including a Fourier series based sliding mode adaptive control (FSSMAC) network for rough adjustment and an adaptive prediction critic network for fine adjustment. First, we unify the time-variant unknown uncertainties including the modeling error, nonlinear friction, unknown dead zone as well as other disturbances into an integrated term. Based on this, FSSMAC is introduced to develop the rough adjustment network to reduce the tracking error. Then, a fine adjustment network (FAN) with two major components is proposed to compensate the tracking error. In our current implementation, the FAN includes an online critic network and a polynomial curve fitting network. Both components have self-learning capability. Simulation results validate the effectiveness of the proposed controller applied in the position tracking applications.

The rest of this paper is organized as follows. Section 1 gives the problem description. Section 2 introduces the design of rough adjustment network (RAN) and Section 3 elaborates design of FAN. In Section 4, detailed simulation results of the proposed method are presented. Furthermore, comparative studies of our method with those of existing approaches are also discussed in this section. Finally, a conclusion and a brief discussion of the future research directions are given in Section 5.

1 Problem description

Consider a class of single input multiple output (SIMO) systems with time-variant unknown uncertainties.

$$\begin{cases} \dot{x}_1(t) = a_1 x_2(t) + d_1(\mathbf{X}, t) \\ \dot{x}_2(t) = a_2 x_3(t) + d_2(\mathbf{X}, t) \\ \vdots \\ \dot{x}_{n-1}(t) = a_{n-1} x_n(t) + d_{n-1}(\mathbf{X}, t) \\ \dot{x}_n(t) = a_n x_n(t) + bu(t) + d_n(\mathbf{X}, t) \\ \mathbf{Y}(t) = [x_1(t) \quad x_2(t) \quad \cdots \quad x_{n-1}(t) \quad x_n(t)] \end{cases} \quad (1)$$

Manuscript received December 18, 2008; accepted February 27, 2009

Supported by the Center for Intelligent Networked Systems (iNetS), Stevens Institute of Technology, United States of America

1. Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken NJ 07030, United States of America
2. Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, Kingston RI 02881, United States of America

DOI: 10.1016/S1874-1029(09)60023-9

where $d_i(\mathbf{X}, t)$ denotes the unknown time-variant uncertainties, $x_i(t)$ ($i = 1, 2, \dots, n$) and $\mathbf{Y}(t)$ are the system's states and output, respectively. Let $\mathbf{X} = [x_1(t) \ \dots \ x_n(t)]^T$ and rewrite (1) into state space equation form as shown in (2):

$$\begin{cases} \dot{\mathbf{X}} = \mathbf{A}\mathbf{X} + \mathbf{B}(\mathbf{X})u + \mathbf{D}(\mathbf{X}, t) \\ \mathbf{Y} = \mathbf{X} \end{cases} \quad (2)$$

where \mathbf{A}, \mathbf{B} are known matrices, and $\mathbf{D}(\mathbf{X}, t)$ is unknown time variant uncertainty term, as shown in (3).

$$\mathbf{A} = \begin{bmatrix} 0 & a_1 & 0 & 0 & \dots & 0 \\ 0 & 0 & a_2 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & \dots & 0 & a_{n-1} \\ 0 & 0 & \dots & \dots & 0 & a_n \end{bmatrix} \quad (3)$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ b \end{bmatrix}$$

$$\mathbf{D}(\mathbf{X}, t) = \begin{bmatrix} d_1(\mathbf{X}, t) \\ \vdots \\ d_{n-1}(\mathbf{X}, t) \\ d_n(\mathbf{X}, t) \end{bmatrix}$$

Assumption 1. From [18], \exists constant vector $\mathbf{C} = [c_1 \ \dots \ c_n] \in \mathbf{R}^{1 \times n}$, $\forall c_i \neq 0, i = 1, 2, \dots, n$ for the perturbation system, $\mathbf{C}\mathbf{B}(\mathbf{X}) \neq 0$ and the time-variant uncertainty $\mathbf{C}\mathbf{D}(\mathbf{X}, t)$ is quadratically integrable in any limited time frame T , namely $\mathbf{C}\mathbf{D}(\mathbf{X}, t) \in L^2[\mathbf{R}^+]$.

Fig. 1 shows the system-level diagram of the proposed approach. In this figure, the RAN includes two components: the SMC component and the online uncertainty approximation component; the FAN includes the prediction network and the critic network. The RAN and FAN are designed to satisfy two different primary objectives: RAN targets at the effectiveness of the system while FAN is used to meet the efficiency requirement of the system. In this way, asymptotic stability is guaranteed in the RAN design, while FAN aims to meet the real-time requirement of adaptive online control, and the transient performance of the system is also guaranteed.

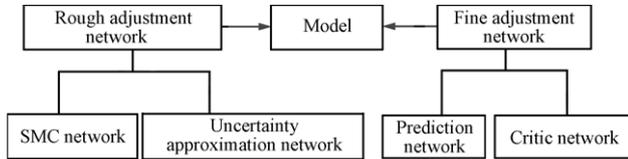


Fig. 1 The system-level diagram of the proposed approach

2 Design of rough adjustment network

From (2), one can see, the term $\mathbf{D}(\mathbf{X}, t)$ represents the unknown time-variant nonlinear uncertainty. It is mainly caused by the nonlinear friction, dead zone, and other factors. Conventional model-based control methods are not effective to solve this problem due to the difficulty in obtaining the precise model of $\mathbf{D}(\mathbf{X}, t)$. Therefore, it is necessary to design an adaptive controller for online approximation to improve control accuracy. We use Fourier series combined with SMC to design the controller that can adap-

tively approximate time-variant uncertainties $\mathbf{D}(\mathbf{X}, t)$ and can automatically offset the approximation error.

Given the desired position $y_r(t)$, we assume

$$\begin{aligned} x_{d1}(t) &= y_r(t), \quad x_{di}(t) = \frac{1}{a_{i-1}} \dot{x}_{d(i-1)}(t) \\ \mathbf{X}_d(t) &= [x_{d1}(t) \ \dots \ x_{dn}(t)]^T \end{aligned} \quad (4)$$

In this way, the error function is defined in (5):

$$\mathbf{E}(t) = \mathbf{X}(t) - \mathbf{X}_d(t) = [e_1(t) \ \dots \ e_n(t)]^T \quad (5)$$

If one chooses the sliding surface in (7)

$$s(t) = \mathbf{C}(\mathbf{X}(t) - \mathbf{X}_d(t)) = \mathbf{C} \cdot \mathbf{E}(t) \quad (6)$$

Then, the derivative of (6) can be calculated as given in (7):

$$\begin{aligned} \frac{\partial s(t)}{\partial t} &= \mathbf{C}\dot{\mathbf{X}}(t) - \mathbf{C}\dot{\mathbf{X}}_d(t) = \\ &\mathbf{C}(\mathbf{A}\mathbf{X}(t) + \mathbf{B}u(t) + \mathbf{D}(\mathbf{X}, t)) - \mathbf{C}\dot{\mathbf{X}}_d(t) \end{aligned} \quad (7)$$

On the basis of the sliding function (6) and its derivative (7), a controller is proposed including an undetermined time variant uncertainty term u_m and an undetermined compensation term $u_r(t)$. These uncertainty terms are expressed by finite Fourier series; let $u_m(t) = \mathbf{C}\mathbf{D}(\mathbf{X}, t)$. Based on Assumption 1, the time-variant $u_m(t)$ is estimated by $\hat{u}_m(t)$ with finite Fourier series as formulated in (8).

$$u_m(t) = \mathbf{W}^T \mathbf{Z}(t) + \delta, \quad \hat{u}_m(t) = \hat{\mathbf{W}}^T \mathbf{Z}(t) \quad (8)$$

where $\mathbf{Z}(t)$ and \mathbf{W} are Fourier basis function vector and coefficients vector, respectively. δ denotes the approximation error. The function approximation is used to approximate $u_m(t)$ by estimating the coefficient vector \mathbf{W} with $\hat{\mathbf{W}}$. Let us define

$$\tilde{\mathbf{W}} = \mathbf{W} - \hat{\mathbf{W}} \quad (9)$$

Similar as [18], one can get

$$\frac{\partial s(t)}{\partial t} = \tilde{\mathbf{W}}^T \mathbf{Z}(t) - k(\text{sgn}(s(t)) + s(t)) + u_r(t) \quad (10)$$

where $u_r(t)$ is used to compensate $\tilde{\mathbf{W}}$. From (2) ~ (10), $u_{\text{RAN}}(t)$ can be deduced as

$$\begin{aligned} u_{\text{RAN}}(t) &= -(\mathbf{C}\mathbf{B})^{-1} \mathbf{C} \left(\mathbf{A}\mathbf{X}(t) - \dot{\mathbf{X}}_d(t) \right) - \\ &(\mathbf{C}\mathbf{B})^{-1} k(\text{sgn}(s(t)) + s(t)) - \\ &(\mathbf{C}\mathbf{B})^{-1} \hat{u}_m(t) + (\mathbf{C}\mathbf{B})^{-1} u_r(t) \end{aligned} \quad (11)$$

In (11), $(\mathbf{C}\mathbf{B})^{-1} \mathbf{C} \left(\mathbf{A}\mathbf{X}(t) - \dot{\mathbf{X}}_d(t) \right)$ is based on nominal model, $(\mathbf{C}\mathbf{B})^{-1} k(\text{sgn}(s(t)) + s(t))$ is the reaching law term, k is a positive real number, $(\mathbf{C}\mathbf{B})^{-1} \hat{u}_m(t)$ is the undetermined time variant uncertainty term, $\hat{u}_m(t)$ is the approximation of $u_m(t) = \mathbf{C}\mathbf{D}(t)$, and $(\mathbf{C}\mathbf{B})^{-1} u_r(t)$ is the undetermined compensation term. Suppose $u_m(t)$ and $\hat{u}_m(t)$ are quadratically integrable as in Assumption 1 and $\mathbf{C}\mathbf{B}(\mathbf{X}) \neq 0$. Furthermore, if the update law are chosen as

in (12) by using Lyapunov function with $\hat{\mathbf{W}}$ and $s(t)$, one can prove the system is asymptotically stable. The proof follows the same procedure in [18] with modifications in reaching law. Due to space considerations, we refrain from providing the detailed proof instead direct the interested readers to [18] for details.

$$\frac{\partial \hat{\mathbf{W}}}{\partial t} = \frac{\eta_1}{\eta_2} s(t) \mathbf{Z}(t), \quad u_r(t) = -\eta_3 \eta_1 s(t) \quad (12)$$

Now, we will prove that the sliding hypersurface asymptotically converges to zero in each period of constant motor velocity direction.

Proof. Define Lyapunov function as

$$\mathbf{V}(s(t), \tilde{\mathbf{W}}) = \frac{1}{2} \eta_1 (s(t))^2 + \frac{1}{2} \eta_2 \tilde{\mathbf{W}}^T \tilde{\mathbf{W}} \geq 0 \quad (13)$$

Then, we have two expressions of $\dot{\mathbf{V}}(s(t), \tilde{\mathbf{W}})$

$$\frac{\partial \mathbf{V}(s(t), \tilde{\mathbf{W}})}{\partial t} = \eta_1 s(t) \frac{\partial s(t)}{\partial t} - \eta_2 \tilde{\mathbf{W}}^T \frac{\partial \hat{\mathbf{W}}}{\partial t} \quad (14)$$

$$\frac{\partial \mathbf{V}(s(t), \tilde{\mathbf{W}})}{\partial t} = -\eta_3 (\eta_1 s(t))^2 - k \eta_1 (|s(t)| + s^2(t)) < 0 \quad (15)$$

Then, we have:

$$\eta_1 s(t) \frac{\partial s(t)}{\partial t} - \eta_2 \tilde{\mathbf{W}}^T \frac{\partial \hat{\mathbf{W}}}{\partial t} < -k \eta_1 |s| \quad (16)$$

From the update law (12), one can get:

$$\eta_1 s(t) \frac{\partial s(t)}{\partial t} - \eta_2 \tilde{\mathbf{W}}^T \frac{\eta_1}{\eta_2} s(t) \mathbf{Z}(t) < -k \eta_1 |s| \quad (17)$$

Considering (17), if $s(t) > 0$, then one can get

$$\frac{\partial s(t)}{\partial t} < -k + \tilde{\mathbf{W}}^T \mathbf{Z} \quad (18)$$

Since $\tilde{\mathbf{W}}^T \mathbf{Z}$ is bounded, by taking integral with respect to t in the range $[0, t_r]$ ($s_{t_r} = 0$, t_r is the time when the system first reaches the sliding hypersurface), one can get $\exists M$ satisfying

$$\int_0^{t_r} \tilde{\mathbf{W}}^T \mathbf{Z} dt < M \quad (19)$$

Considering (18) and (19), one can get:

$$-s_0 < M - k t_r \quad (20)$$

Therefore, one can get:

$$t_r < \frac{M + s_0}{k} \quad (21)$$

On the other hand, if $s(t) < 0$, one can get the same result in a similar way. Therefore, the sliding hypersurface will converge to zero in a finite time. \square

3 Design of fine adjustment network

3.1 General design principle

The RAN proposed above can successfully approximate the time-variant uncertainty. However, the theoretical proof in [18] has demonstrated that the tracking error can be reduced into a very small bound rather than asymptotically converges to zero. It is rather intriguing on how to introduce new design approaches to reduce the error bound. In this section, we propose to use an adaptive FAN to compensate the tracking error. The FAN includes two major components: the critic network and the prediction network. In the following, we will show that with a careful design of these two components, even very simple compensator such as proportional-integral-derivative (PID) controller can reach satisfactory results.

We now present the detailed design strategy of the proposed approach. We use the prediction component and the critic component to predict \mathbf{E}_{RAN} , the tracking error of RAN, i.e., the error without the FAN. We would like to note that \mathbf{E}_{RAN} is a virtual term and does not really exist after adding the FAN. The objective of the prediction component and the critic component is to approximate this virtual \mathbf{E}_{RAN} . Here, we define the approximated \mathbf{E}_{RAN} as $\hat{\mathbf{E}}_{\text{RAN}}$. At sampling time k , we define

$$\begin{aligned} \mathbf{E}(k) &= \mathbf{X}(k) - \mathbf{X}_d(k) \\ \mathbf{X}(k) &= \mathbf{X}_{\text{RAN}}(k) + \mathbf{X}_{\text{FAN}}(k) \\ \mathbf{E}_{\text{RAN}}(k) &= \mathbf{X}_{\text{RAN}}(k) - \mathbf{X}_d(k) \end{aligned} \quad (22)$$

From (22), we can deduce $\mathbf{E}_{\text{RAN}}(k) = \mathbf{E}(k) - \mathbf{X}_{\text{FAN}}(k)$. $\mathbf{E}(k)$, $\mathbf{X}(k)$, and $\mathbf{X}_d(k)$ are observable, whereas $\mathbf{X}_{\text{RAN}}(k)$, $\mathbf{X}_{\text{FAN}}(k)$, and $\mathbf{E}_{\text{RAN}}(k)$ are not. Equation (23) is used to estimate the value of $\mathbf{E}_{\text{RAN}}(k)$ online, where $\mathbf{X}_{\text{FAN}}(k)$ is substituted by $-\hat{\mathbf{E}}_{\text{RAN}}(k-1, k)$.

$$\hat{\mathbf{E}}_{\text{RAN}}(k) = \mathbf{E}(k-1) + \hat{\mathbf{E}}_{\text{RAN}}(k-1, k) \quad (23)$$

Equation (23) serves as the adaptive estimation law of the FAN. For the critic network, (23) is used to set the online training data $\langle \text{set_train}(k), \text{target}(k) \rangle$:

$$\begin{aligned} \text{set_train}(k) &= \langle \mathbf{E}(k-2), \mathbf{X}(k-2), u_{\text{RAN}}(k-2), \\ &\quad \mathbf{X}_d(k-1) \rangle \\ \text{target}(k) &= \mathbf{E}(k-1) + \hat{\mathbf{E}}_{\text{RAN_critic}}(k-1) \end{aligned} \quad (24)$$

For the prediction network, (23) is transformed into the form formulated by (25):

$$\hat{\mathbf{E}}_{\text{RAN_predict}}(k) = \mathbf{E}(k-1) + \hat{\mathbf{E}}_{\text{RAN_predict}}(k-1, k) \quad (25)$$

3.2 Critic network design

A nonlinear multilayer feedforward neural network is used as the critic network. Similar to [19–21], we use one hidden layer in our current design. Fig. 2 shows the network architecture. Here, we focus on the critic network architecture design and its mathematical learning foundation. The convergence of this critic network can be proved according to the Robbins-Monro algorithm. In fact, [19] proved that this critic network actually converges to a (local) minimum from statistical perspective. Due to space considerations, we refrain from providing the detailed proof on this and interested readers can refer to [19] for details. Also please note that $\mathbf{E}(k)$ and the related variables are scalars hereafter in accordance with the application of the

SIMO system.

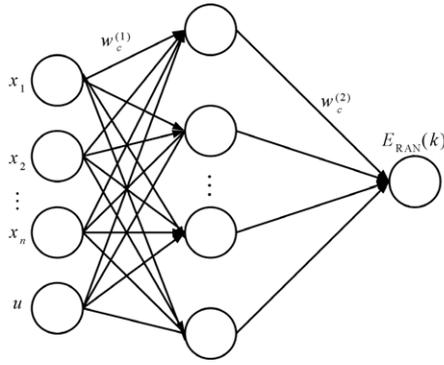


Fig. 2 Schematic diagram of the critic network

The learning process in Fig. 2 can be summarized as (26) ~ (33)^[19–21], where $J(k)$ is the critic network's output, and $T(k)$ is the abbreviation for $\hat{E}_{\text{RAN_predict}}(k)$:

$$\begin{cases} \hat{e}_{\text{critic}}(k) = \frac{1}{2} (J(k) - T(k))^2 \\ J(k) = k_{\text{out}} \frac{1 - e^{-v(k)}}{1 + e^{-v(k)}} \end{cases} \quad (26)$$

$$v(k) = \sum_{i=1}^{N_h} w_{c_i}^{(2)}(k) g_i(k) \quad (27)$$

$$g_i(k) = \frac{1 - e^{-h_i(k)}}{1 + e^{-h_i(k)}}, \quad i = 1, \dots, N_h \quad (28)$$

$$h_i(k) = \sum_{j=1}^{n+1} w_{c_{ij}}^{(1)}(k) x_j(k), \quad i = 1, \dots, N_h \quad (29)$$

where v is the input to the action node, and g_i and h_i are the output and the input of the hidden nodes of the critic network, respectively.

1) $\Delta w_c^{(2)}$ (hidden to output layer)

$$\Delta w_c^{(2)c} = l_c(k) \left[-\frac{\partial \hat{e}_{\text{critic}}(k)}{\partial \Delta w_{c_i}^{(2)}(k)} \right] \quad (30)$$

$$\left[\frac{\partial \hat{e}_{\text{critic}}(k)}{\partial w_{c_i}^{(2)}(k)} \right] = \frac{\partial \hat{e}_{\text{critic}}(k)}{\partial J(k)} \frac{\partial J(k)}{\partial v(k)} \frac{\partial v(k)}{\partial w_{c_i}^{(2)}(k)} = (J(k) - T(k)) k_{\text{out}} \left[\frac{1}{2} (1 - v^2(k)) \right] g_i(k) \quad (31)$$

2) $\Delta w_c^{(1)}$ (input to hidden layer)

$$\Delta w_{c_{ij}}^{(1)} = l_c(k) \left[-\frac{\partial \hat{e}_{\text{critic}}(k)}{\partial \Delta w_{c_{ij}}^{(1)}(k)} \right] \quad (32)$$

$$\begin{aligned} \left[\frac{\partial \hat{e}_{\text{critic}}(k)}{\partial w_{c_{ij}}^{(1)}(k)} \right] &= \frac{\partial \hat{e}_{\text{critic}}(k)}{\partial J(k)} \frac{\partial J(k)}{\partial v(k)} \frac{\partial v(k)}{\partial g_i(k)} \frac{\partial g_i(k)}{\partial h_i(k)} \frac{\partial h_i(k)}{\partial w_{c_{ij}}^{(1)}(k)} = \\ &= (J(k) - T(k)) k_{\text{out}} \left[\frac{1}{2} (1 - v^2(k)) \right] w_{c_i}^{(2)}(k) \times \\ &\quad \left[\frac{1}{2} (1 - g_i^2(k)) \right] x_j(k) \end{aligned} \quad (33)$$

Similar to [19–20], we also used the normalization procedure to properly scale the weights. This is defined in (34):

$$w_c(k+1) = \frac{w_c(k) + \Delta w_c(k)}{\|w_c(k) + \Delta w_c(k)\|_1} \quad (34)$$

3.3 Prediction network design

Due to the characteristics of the system, polynomial curve fitting is used here to approximate the tracking error. We would like to note that other curve fitting functions can also be applied here. At every k -th sample time, two predicting methods are used to predict $E_{\text{RAN}}(k)$: the critical network output $E_{\text{RAN}}(k)$ and the polynomial curve fitting output $\hat{E}_{\text{RAN_predict}}(k)$. Then, we compare the absolute difference of these two predictions with the threshold ε , defined in (35) as

$$\gamma = \text{abs}(\hat{E}_{\text{RAN_predict}}(k) - \hat{E}_{\text{RAN_critic}}(k)) \quad (35)$$

Two scenarios are considered based on the result of (35). If $\gamma \leq \varepsilon$, we directly use the current polynomial curve fitting output to compensate the tracking error as an input to the system. On the other hand, if $\gamma > \varepsilon$, then we take the average of these two prediction results as in (36) and combine this information with the historical data to update the $p(k)$.

$$\hat{E}_{\text{RAN_predict}}(k) \leftarrow \frac{1}{2} (\hat{E}_{\text{RAN_critic}}(k) + \hat{E}_{\text{RAN_predict}}(k)) \quad (36)$$

Fig.3 shows the architecture of the proposed dual network. We now give the detailed control algorithm as follows.

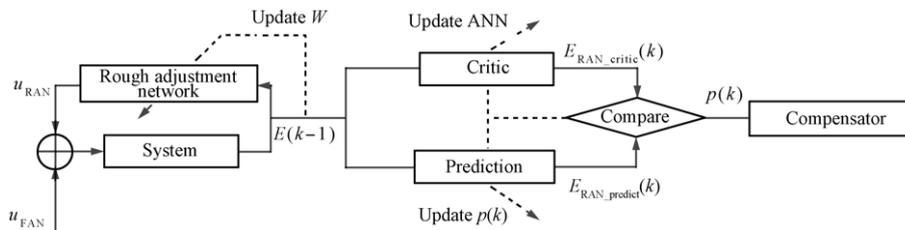


Fig. 3 The proposed dual network architecture

Notations.

- 1) Critic network's parameters:
 N_{in} : total number of input nodes;
 N_h : total number of hidden nodes;
 N_{out} : total number of output nodes;
 L_c : learning rate;
 N_{cycle} : internal back-propagation cycle.
 2) Prediction network's parameters:
 ε : the threshold of judging γ ;
order: the order of the curve fitting polynomial's regression;
steps: the number of backtracking steps for the regression algorithm.

Algorithm.**Initialization.**

- 1) Initialize RAN's uncertainty approximation network, choose proper orthogonal basis functions;
 2) Initialize the coefficient vector: $\hat{w}_i(0) = 0$ ($i = 1, 2, \dots, N$);
 3) Initialize the SMC parameters, the critic network parameters, and the prediction network parameters as in Table 4.

Do for $k = 1, 2, \dots, T$:

- 1) Observe the system state $\mathbf{X}(k)$ and calculate $\hat{\mathbf{X}}_d(k), \hat{\mathbf{X}}_d(k)$;
 2) Compute $u_{RAN}(k)$ as follows:
 a) Compute $s(k)$ based on (6);
 b) Compute $\hat{\mathbf{W}}(k)$ based on (12);
 c) Compute $\hat{u}_m(k)$ and $u_r(k)$ based on (8) and (12);
 d) Compute $u_{RAN}(k)$ based on (11);
 3) Compute $u_{FAN}(k)$ as follows:
 a) Use (24) as input value to train the neural network, and output $\hat{E}_{RAN_critic}(k)$;
 b) Use $p(k-1)$ to output $\hat{E}_{RAN_predict}(k)$, then $\hat{E}_{RAN_predict}(k) \leftarrow \hat{E}_{RAN_predict}(k) + E(k-1)$.
 c) Compute γ based on (35). If $\gamma > \varepsilon$, update $\hat{E}_{RAN_predict}(k)$ based on (36) and update $p(k)$;
 d) Use $p(k)$ and the controller component to compute $u_{FAN}(k)$;
 4) Update $u(k) = u_{FAN}(k) + u_{RAN}(k)$.

End Do

4 System simulation

In this section, we demonstrate the performance of the proposed model based on a DC motor and a tachogenerator for speed feedback application. The simplified DC motor open-loop system is shown in Fig. 4, where U_f represents the virtual input voltage equivalent to the unknown nonlinear time variant friction torque (the model is effective only if $|u| > |U_f|$), x_1 represents the position of the motor, x_2 represents the velocity, T_m is the time constant of DC motor, and K_e is the speed feedback coefficient. The outputs of the system are position and velocity. The time-variant model is nonlinear due to the friction when the motor is operating at a relatively low speed, especially when the motor is changing its velocity direction and model parameters.

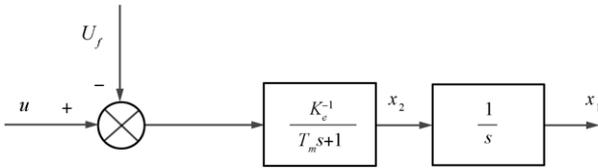


Fig. 4 Simplified open-loop model of linear motor

The state space equation of Fig. 4 is given in (37):

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\frac{1}{T_m}x_2 + \frac{1}{T_m K_e}u(t) - \frac{1}{T_m K_e}U_f(t) \end{cases} \quad (37)$$

For comparison, we choose the same model parameters as in [18], which are provided in Table 1. Here, f is the abbreviation for forward model and b is the abbreviation for the backward model. Sample time T_s is 0.01 s.

Table 1 The forward/backward models including nonlinear friction torque which are described by (38) and (39)

$T_m(f)$	$K_e(f)$	$T_m(b)$	$K_e(b)$
3.0787	5.3658	2.7746	5.3169

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\frac{1}{3.0787}x_2 + \frac{5.3658}{3.0787}u(t) - \frac{5.3658}{3.0757}U_{f1} \end{cases} \quad (38)$$

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\frac{1}{2.7746}x_2 + \frac{5.3169}{2.7746}u(t) - \frac{5.3169}{2.7746}U_{f2} \end{cases} \quad (39)$$

The tribology study demonstrated that the friction in electromechanical system mainly composes of static friction, Coulomb friction, and viscous friction, in which viscous friction is proportional to the motor's speed^[5]. The nonlinear friction torque including the static friction and Coulomb friction has various kinds of expressions, of which Stribeck^[22] model of (40) has been widely used in the community. In our current study, we also adopted the model of (40) with the parameters presented in Table 2.

$$U_f(\omega) = U_c \text{sgn}(\omega) + (U_s - U_c) \cdot e^{-\alpha|\omega|} \text{sgn}(\omega) \quad (40)$$

Table 2 Stribeck friction model parameters

$U_f(f)$	$U_c(f)$	$\alpha(f)$	$U_f(b)$	$U_c(b)$	$\alpha(b)$
250	465	0.002	50	240	0.003

Based on the experimental results in [18], we define the nonlinear friction torque U_{f1} and U_{f2} as:

$$\begin{cases} U_{f1} = 250 + 215 \times e^{-0.002|x_1|} \\ U_{f2} = 50 + 190 \times e^{-0.003|x_1|} \end{cases} \quad (41)$$

The system model is shown in Fig. 5, where f_1 and f_2 represent U_f in the forward and backward model, respectively.

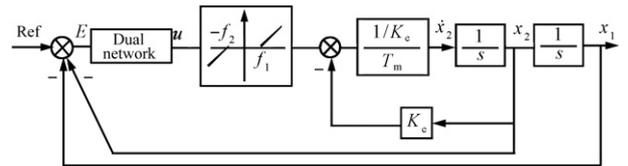


Fig. 5 System with equivalent nonlinearity and dual network

The RAN parameters are set as in Table 3, where N is the number of basis functions. C are the coefficients to construct the sliding surface, and μ_i denote the parameters in \hat{u}_m and u_r , $i = 1, 2, 3$.

Table 3 RAN parameters

N	C	η_1	η_2	η_3	k
8	$10 * [30, 0.4]$	3	1	20	17

Case 1. For performance comparison, we use the position tracking function $y(t) = 150 \sin(2\pi \times 0.0667t)$ as in [18]. The FAN and PID controller's parameters are shown in Tables 4~6. Fig. 6 shows the position tracking error performance of the proposed dual network in this paper. The tracking error is mainly caused by the time-variant nonlinear friction when the motor is operated at a relatively low speed, especially when the motor is changing its velocity directions. This is confirmed in our simulation result as shown in Fig. 6. To compare our result with those of existing methods, Fig. 6 also illustrates the position tracking error of the methods in [14] and [18]. Table 7 shows the corresponding numerical values of these results.

Table 4 System parameters

Algorithms	Parameter			
	*	K_p	K_i	K_d
PID	Case 1	120	2	0.3
	Case 2	200	6	1

Table 5 System parameters

Algorithms	Parameter		
Prediction network	Threshold ε	order	steps
	0.01	3	50

Table 6 System parameters

Algorithms	Parameter				
	Critic network	N_{in}	N_{hidden}	N_{out}	L_c
	4	3	1	0.001	500

Table 7 Comparison of position tracking

Tracking error	Main period	Direction changing period
Dual network	$[-0.8, 0.8]$	$[-1.5, 1.5]$
Reference [14]	$[-10, 10]$	$[-40, 20]$
Reference [18]	$[-7, 7]$	$[-7, 7]$

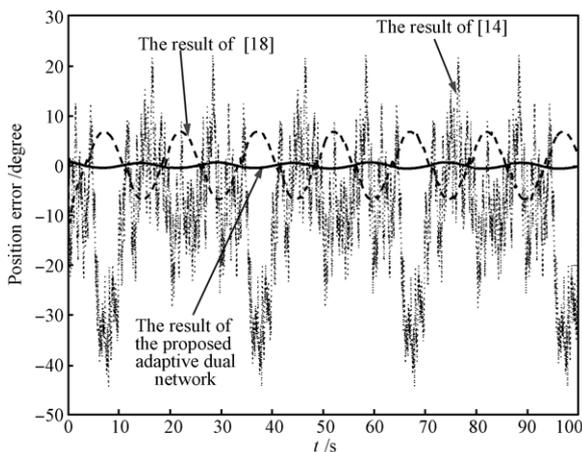


Fig. 6 Tracking error comparison of Case 1

From Fig. 6 and Table 7, one can see the tracking error of the method in [14] mainly remains within $[-10, 10]$,

and such errors can jump to 20 or -40 when the motor is changing its velocity direction. The method in [18] improves the results so that the tracking errors are mostly located within $[-7, 7]$. In our proposed method, the tracking error mainly remains within the bound of $[-0.8, 0.8]$, and such errors never goes beyond the range of $[-1.5, 1.5]$ even in the worst situations. These results suggest that the proposed method can achieve significantly better results compared to the existing methods.

To have a detailed analysis of the proposed system, Fig. 7 shows the fitting error of the E_{RAN} , in which we can see that the fitting error remains within the bound of $[-0.15, 0.15]$. Further analysis suggests that at each period of identical velocity direction, the fitting error asymptotically converges to zero, which confirms the effectiveness of the proposed FAN. Fig. 8 shows the total control input u and Fig. 9 represents the u_{RAN} of the system. By comparing u and u_{RAN} , we can find that after adding the FAN, the total control input u is much better than u_{RAN} . This proves the effectiveness of the proposed dual network from the input perspective.

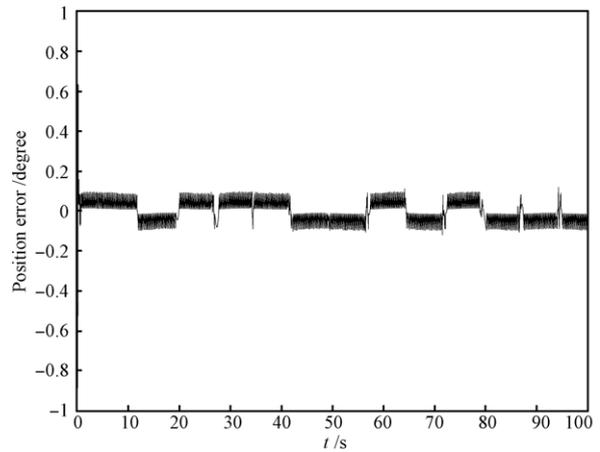


Fig. 7 Fitting error of Case 1

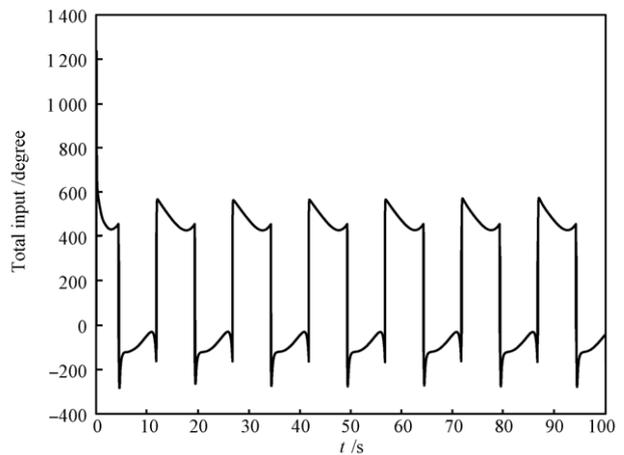


Fig. 8 Total control input to the system: u

Case 2. We now examine the performance of the proposed dual network design on more complicated situations. To be realistic, we implemented noises on three places similar to those considered in [19], $u(t)$, U_f , and x_1 . The actuator noise through $u = (1 + \rho) \times u$, where ρ is a uniformly distributed random variable. For the sensor noise on x_1 , we experimented with adding both

uniform and Gaussian random variable to x_1 . The uniform state sensor noise was implemented through $x_1 = (1 + \text{noise percentage}) \times x_1$. Gaussian sensor noise was zero mean with specified variance σ^2 . Meanwhile, the nonlinear friction part is also added with noise simultaneously through $U_f = (1 + \text{noise percentage}) \times U_f$. The noise percentage part is the same as that added with x_1 , namely the noise is both uniform and Gaussian. The position tracking function is $150 \sin(0.2\pi \times 0.667t) + 300 \sin(2t)e^{-0.05t}$. Here, we compare the results of our method with those approaches given in [14] and [18], in which only the Stribeck friction is considered. The parameters of the FAN and PID controller are shown in Table 8.

see that the tracking error of the method in [14] mainly remains within $[-10, 10]$. The method in [18] improves the results so that the tracking errors are mostly located within $[-6, 6]$. In our proposed method, the tracking error mainly remains within the bound of $[-0.7, 0.7]$, and such errors never go beyond the range of $[-2.5, 1.8]$. Fig. 12 shows the fitting error of E_{RAN} , in which one can see that the error remains within the bound $[-0.15, 0.15]$ after convergence. These results suggest that the proposed method can achieve significantly better results compared to the existing approaches. We would also like to note that while the PID parameters seem to work fine in this example, one may need to adjust such parameters if it is used in different application scenarios.

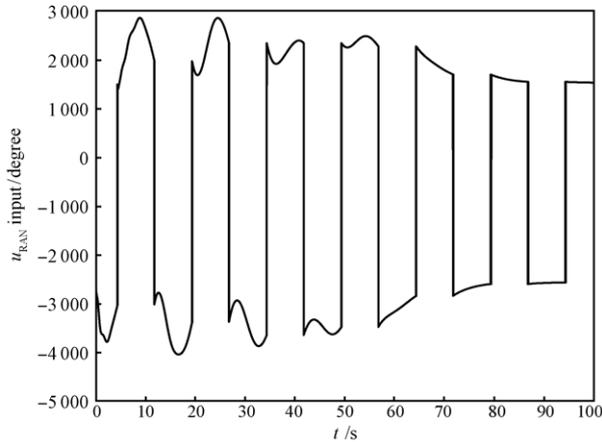


Fig. 9 The output of RAN: u_{RAN}

Here, we show a snapshot of the simulation results with uniform noise on u ($\rho = 5\%$), Gaussian noise on x_1 and U_f ($\sigma^2 = 0.1$). From Fig. 10, Fig. 11, and Table 9, one can

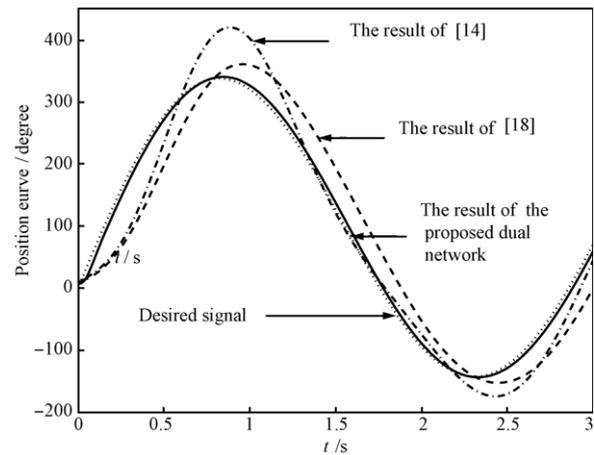


Fig. 11 Tracking error comparison of Case 2 in time frame $[0\text{s}, 3\text{s}]$

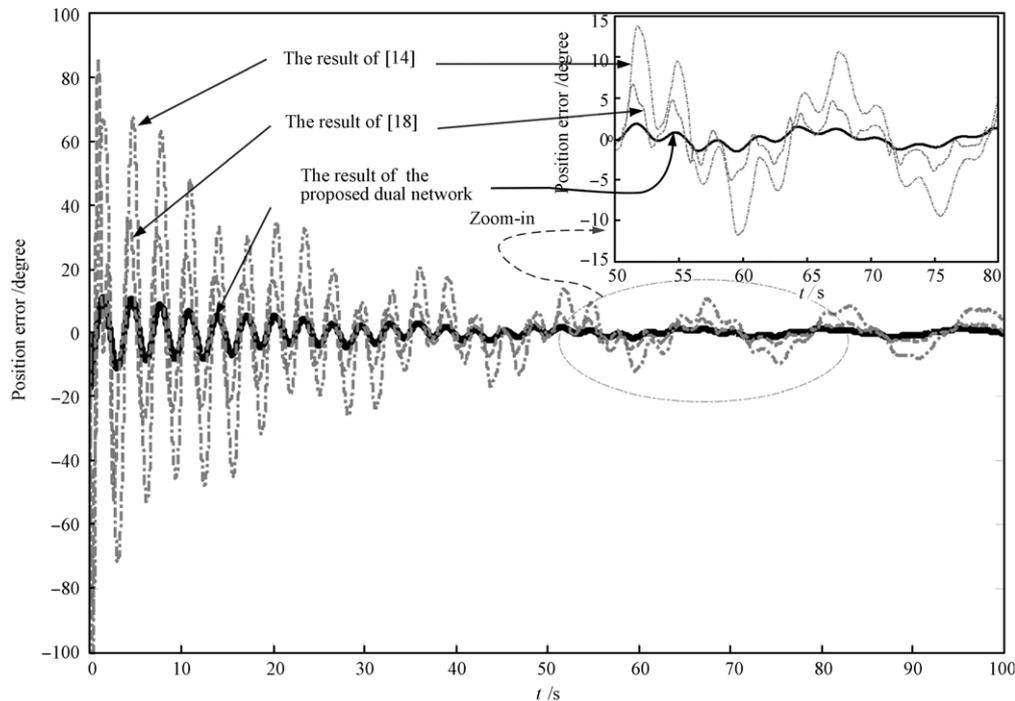


Fig. 10 Tracking error comparison of Case 2

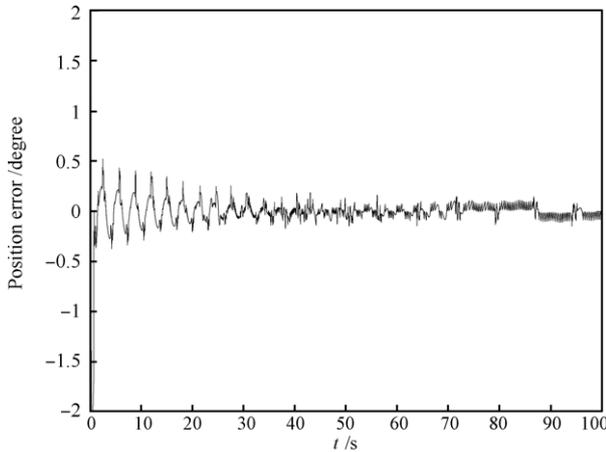


Fig. 12 Fitting error of Case 2

Table 8 Performance evaluation of dual network under different noise conditions

Noise type	Place to add	Parameters
Noise free	*	*
Uniform	u	$\rho = 5\%, 10\%$
Uniform	x_1	$\rho = 5\%, 10\%$
Gaussian	x_1	$\sigma^2 = 0.1, 0.2$
Uniform	U_f	$\rho = 5\%, 10\%$
Gaussian	U_f	$\sigma^2 = 0.1, 0.2$

Table 9 Comparison of position tracking

Tracking error	Main period	Direction changing period
Dual network	$[-0.7, 0.7]$	$[-2.5, 1.8]$
Reference [14]	$[-10, 10]$	$[-30, 25]$
Reference [18]	$[-6, 6]$	$[-6, 6]$

Finally, Fig. 10 also suggests that the convergence speed of the proposed dual network system is faster than those of [14] and [18]. If the tracking signal is periodical, or “simple” in some sense such as Case 1, the advantage of our proposed method is not obvious as shown in Fig. 6. In such a case, the tracking signal can reach its steady state after only about 10 time steps (0.1s). However, when the tracking signal becomes complicated, our method exposes its superiority over existing approaches as clearly shown in Fig. 10.

5 Conclusions

In this paper, a novel adaptive dual network structure based on a RAN and an FNA is proposed to eliminate the unknown time-variant uncertainties of a servo system. In this design, the RAN is based on the combination of SMC, function approximation, and error compensation technique, and the FAN includes a critic network based on a neural network model and a prediction network based on an online curve fitting scheme. The major advantage of the proposed dual network structure is that it can reduce the impact of the uncertainties into a bounded tracking error. We demonstrate the applications of this approach to a DC motor position tracking problem with different tracking signals. Simulation results and comparative study of this method with those of existing approaches demonstrate the effectiveness of this design strategy.

There are several interesting future research directions of this topic. For instance, the design principle and mutual relationships between the RAN and FAN is yet to be deeply explored. Furthermore, in addition to the curve fitting techniques used in this paper, other methods can also be used for function approximation. The theoretical and empirical study of such effects need to be fully analyzed. Furthermore, in our current study, we adopted the neural network model to design the critic network. We would like to note that this is possible to have other alternative design approaches for such critic networks. For instance, the online value system proposed in [23] might be an effective technique to implement the critic network. Generally speaking, the optimal strategy of critic network design is a challenging task across different application domains. Our simulation results in this paper demonstrated that neural network can be a powerful approach for such critic network design, which is consistent with many of the existing literature results as presented in [19–21]. Finally, as a novel control scheme, it would be interesting to test this idea across a wide range of application domains. Motivated by our initial results in this paper, we believe that the proposed dual network structure may provide a powerful method for the adaptive control problems.

References

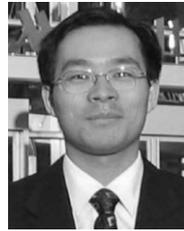
- Han H G, Su C Y, Stepanenko Y S. Adaptive control of a class of nonlinear systems with nonlinearly parameterized fuzzy approximators. *IEEE Transactions on Fuzzy Systems*, 2001, **9**(2): 315–323
- Young K D, Utkin V I, Ozguner U. A control engineer’s guide to sliding mode control. *IEEE Transactions on Control Systems Technology*, 1999, **7**(3): 328–342
- Krstic M, Kanellakopoulos I, Kokotovic P V. *Nonlinear and Adaptive Control Design*. New York: John Wiley and Sons, 1995
- Kokotovic P V. The joy of feedback: nonlinear and adaptive. *IEEE Control Systems Magazine*, 1992, **12**(3): 7–17
- Khalil H K. *Nonlinear Systems*. New York: Prentice Hall, 2002
- Isidori A. *Nonlinear Control Systems: an Introduction*. Berlin: Springer-Verlag, 1989
- Lindlau J D, Knospe C R. Feedback linearization of an active magnetic bearing with voltage control. *IEEE Transactions on Control Systems Technology*, 2002, **10**(1): 21–31
- Kwatny H G, Teolisb C, Mattice M. Variable structure control of systems with uncertain nonlinear friction. *Automatica*, 2002, **38**(7): 1251–1256
- Oya M, Su C Y, Katoh R. Robust adaptive motion/force tracking control of uncertain nonholonomic mechanical systems. *IEEE Transactions on Robotics and Automation*, 2003, **19**(1): 175–181
- Calise A J, Hovakimyan N, Idan M. Adaptive output feedback control of nonlinear systems using neural networks. *Automatica*, 2001, **37**(8): 1201–1211
- Calise A J, Rysdyk R T. Nonlinear adaptive flight control using neural networks. *IEEE Control Systems Magazine*, 1998, **18**(6): 14–25
- Boling J M, Seborg D E, Hespanha J P. Multi-model adaptive control of a simulated pH neutralization process. *Control Engineering Practice*, 2007, **15**(6): 663–672
- Lin C M, Peng Y F. Missile guidance law design using adaptive cerebellar model articulation controller. *IEEE Transactions on Neural Networks*, 2005, **16**(3): 636–644

- 14 Huang A C, Kuo Y S. Sliding control of non-linear systems containing time-varying uncertainties with unknown bounds. *International Journal of Control*, 2001, **74**(3): 252–264
- 15 Huang A C, Chen Y C. Adaptive multiple-surface sliding control for non-autonomous systems with mismatched uncertainties. *Automatica*, 2004, **40**(11): 1939–1945
- 16 Huang S N, Tan K K, Lee T H. Adaptive motion control using neural network approximations. *Automatica*, 2002, **38**(2): 227–233
- 17 Chen P C, Huang A C. Adaptive sliding control of active suspension systems with uncertain hydraulic actuator dynamics. *Vehicle System Dynamics*, 2006, **44**(5): 357–368
- 18 Liang Y Y, Cong S, Shang W W. Function approximation-based sliding mode adaptive control. *Nonlinear Dynamics*, 2008, **54**(3): 223–230
- 19 Si J, Wang Y T. Online learning control by association and reinforcement. *IEEE Transactions on Neural Networks*, 2001, **12**(2): 264–276
- 20 Si J, Barto A G, Powell W B, Wunsch D. *Handbook of Learning and Approximate Dynamic Programming*. New York: IEEE, 2004. 125–153
- 21 Prokhorov D V, Wunsch D C. Adaptive critic designs. *IEEE Transactions on Neural Networks*, 1997, **8**(5): 997–1007
- 22 He H, Starzyk J A. Online dynamic value system for machine learning. In: *Proceedings of the 4th International Symposium on Neural Networks*. Nanjing, China: Springer, 2007. 441–448
- 23 Dupont P, Hayward V, Armstrong B, Altpeter F. Single state elasto-plastic friction models. *IEEE Transactions on Automatic Control*, 2002, **47**(5): 787–792



LIU Bo Received his B.S. degree in electrical engineering from Nanjing University of Post and Telecommunications, and M.S. degree from University of Science and Technology of China, China in 2005 and 2008, respectively. He is currently a Ph.D. candidate in the Department of Electrical and Computer Engineering, Stevens Institute of Technology, USA.

His research interest covers cybernetics, machine learning, data mining, reinforcement learning, and self-adaptive intelligent systems. E-mail: bliu@stevens.edu



HE Hai-Bo Received his B.S. and M.S. degrees in electrical engineering from Huazhong University of Science and Technology, China in 1999 and 2002, respectively, and the Ph.D. degree in electrical engineering from Ohio University, USA in 2006. From 2006 to 2009, he was an assistant professor in the Department of Electrical and Computer Engineering, Stevens Institute of Technology, New Jersey, USA. He is currently an assistant professor in the Department of Electrical, Computer, and

Biomedical Engineering at the University of Rhode Island, USA. His research interest covers machine intelligence, self-adaptive systems, computational intelligence and applications, VLSI and FPGA design, and embedded systems design. Corresponding author of this paper. E-mail: he@ele.uri.edu



CHEN Sheng Received his B.S. and M.S. degrees in control science and engineering from Huazhong University of Science and Technology, China in 2004 and 2007, respectively. He is currently a Ph.D. candidate in the Department of Electrical and Computer Engineering, Stevens Institute of Technology, USA.

His research interest covers computational intelligence and applications, machine learning, data mining, reinforcement learning, and self-adaptive intelligent systems. E-mail: schen5@stevens.edu