Received June 11, 2019, accepted June 19, 2019, date of publication June 24, 2019, date of current version July 15, 2019.
Digital Object Identifier 10.1109/ACCESS.2019.2924478

RBF Networks-Based Weighted Multi-Model Adaptive Control for a Category of Nonlinear Systems With Jumping Parameters

QING LI, YUHANG YANG, YUZHEN ZHANG, YUKUN LIU, WEICUN ZHANG, AND JIARUI CUI
School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China
Corresponding author: Yuzhen Zhang (zhangyz_ustb@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61520106010 and Grant 61741302, and in part by the Beijing Municipal Natural Science Foundation under Grant 3182027.

ABSTRACT This paper studies the tracking control problem for nonlinear system with largely jumping parameters. To deal with the large uncertainties of system parameters, an RBF networks-based weighted multi-model adaptive control (WMMAC) strategy is designed, in which the model set can be constructed to cover and approximate the variation range of the plant parameters. Correspondingly, the RBF network controller set generates the appropriate global control signals when parameters jumping, and better transient performance is obtained. Different from existing neural network learning methods, two novel learning rules based on the difference value between the current value of an objective function and its optimal value for the RBF networks are developed to achieve a fast convergence rate. The stability of the overall closed-loop system is validated by the virtual equivalent system (VES) theory, and favorable performance of the proposed control strategy is demonstrated by the numerical simulations.

INDEX TERMS Multi-model adaptive control, RBF network control, learning rule, weighting algorithm, virtual equivalent system.

I. INTRODUCTION

It is well known that multi-model adaptive control strategy, as a method of divide and conquer, aims to deal with the control performance issues for the nonlinear systems with large parameter uncertainties. The multi-model idea was originally proposed by D. T. Magill in the 1960s and applied to the state estimation of stochastic systems with uncertain parameters [1]. Then, D. G. Lainiotis and Athans developed multiple model adaptive control (MMAC) in the 1970s [2]–[4], which had attracted many attentions from scholars.

With the development of nonlinear control, neural network attempts to mimic the information processing of the human brain through a widely interconnected structure and an effective learning mechanism, which is an important method to control nonlinear systems [5]. The radial basis function (RBF) neural network has shown superior self-learning ability and fast learning rate, which is suitable for real-time dynamic control [6]–[8]. Furthermore, a wave of deep learning is set off in many practical applications lately [9], [10]. In the control field, neural networks have been widely used to approximate the unknown dynamics of nonlinear systems [11], [12], which is mainly in the following ways: (1) as the plant model in the model-based control system; (2) as an iterative learning controller in the adaptive control [13], [14]; and (3) combine with other control methods, such as expert fuzzy control [15], fault-tolerant control [16], *et al.* to form new intelligent controllers. Neural network control has been applied to robotic [17], servo mechanisms [18], [19]. And in order to improve the approximation ability, the multilayer neural networks are developed for hydraulic system [20]. In [21], some investigations focusing on deep learning in control field have been given, which includes control target identification, system parameter identification, and control strategy calculation, *et al.*

As the plant system with uncertain parameters becomes more and more complex, some scholars combine multi-model idea with neural networks to improve the control performance [22], [23]. Subsequently, this method is gradually developed in [24]–[26]. Lately, the multi-model adaptive

The associate editor coordinating the review of this manuscript and approving it for publication was Jianyong Yao.

control systems based on different neural network structures are presented [27], [28], in which neural networks are used to identify the unknown dynamic of the system.

However, to the best of our knowledge, tracking control for complex nonlinear system with jumping parameters have not well investigated in the literature. The main difficulties are that (1) the conventional adaptive control makes the transient performance of the plant system poor when parameters jumping; and (2) the existing learning rules, such as the gradient descent method, *et al*. make the neural network factors converge slowly for the adaptive control system, which cannot meet the performance requirements for the systems with largely jumping parameters.

To deal with these issues above, a WMMAC strategy is developed for the nonlinear system with largely jumping parameters. Meanwhile, two novel neural network learning rules are developed to improve the learning rate and fast adaptability. Compared with existing methods, the RBF networks are directly used as local controllers in WMMAC strategy by virtue of their strong self-learning ability, which simplifies the complicated mathematical calculation in the controller design process to some extent. The main contributions of this paper are summarized as follows.

(1) A WMMAC method based on multiple RBF networks is developed for nonlinear system with largely jumping parameters, which can effectively improve transient performance of the plant system;

(2) Two novel learning rules are studied to achieve a faster convergence rate than conventional neural network, which can improve the adaptability.

The rest of this paper is organized as follows. Section II presents the nonlinear system with jumping parameters. The WMMAC strategy based on multiple RBF networks with novel learning rules is presented in Section III. Section IV gives the stability proof of the overall closed-loop system. The simulations are carried out for the proposed control strategy in Section V. The conclusions and future works are drawn in Section VI.

II. DESCRIPTION OF THE PLANT SYSTEM

Consider the following nonlinear discrete-time system with largely jumping parameters

$$y(k) = f\left(y(k-1), u(k-1), \varepsilon(k)\right) \tag{1}$$

where $u(k-1)$ and $y(k-1)$ are the input and output sequences of the system respectively, $\varepsilon(k)$ is a largely uncertain parameter.

A class of jumping parameter is studied in this paper, which can be described as piecewise function

$$\varepsilon(k) = \varepsilon_\chi, k_{\chi-1} < k \leq k_\chi \tag{2}$$

where ε_χ is a constant for time sequences from $k_{\chi-1}$ to k_χ, $\chi = 1, 2, \ldots, N$. All the constants are not completely equal and even quite different from each other, which causes the large parameter uncertainties of the discrete system.

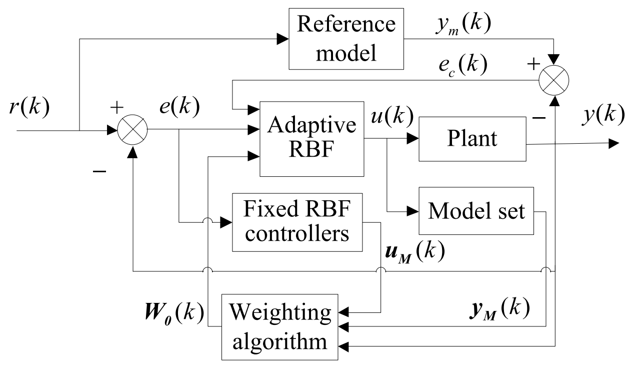


FIGURE 1. Simplified block diagram of the control system.

Remark 1: It is well known that the adaptive ability or robustness of conventional controllers is within a certain range. When the system parameters change beyond the control range, the control performance will deteriorate. This situation is known as large parameter uncertainties.

III. RBF NETWORKS BASED WMMAC SYSTEM

In this section, to achieve the tracking control objective with good transient performance for the nonlinear system when parameters jumping, a novel RBF networks based WMMAC is designed. Firstly, the large uncertainties of system parameters are approximately divided into several parts with constant parameters to form a model set. Secondly, corresponding fixed local RBF network controllers are designed to make the local closed-loop systems stable respectively. Meanwhile, two novel learning rules are developed to accelerate the learning rate of neural network factors. Then, one RBF network adaptive controller that can be reassigned initial values is designed to control the real plant all the time. When the uncertain parameters jump, some excellent values of RBF network factors are assigned to the RBF network adaptive controller as its initial values, such that the adaptive ability can be improved. The good factor values are obtained by fusing the multiple fixed local RBF network controllers by the weighting algorithm [29], [30]. Next, RBF network adaptive controller generate appropriate signals online through the learning of network factors. Thus, transient performance can be guaranteed.

Remark 2: In practice, the accurate jumping parameter values are unknown, and only the model closest to the real system exists in the model set. Therefore, the multi-model control based on the weighting algorithm is more reasonable than the direct switching method in this study.

Remark 3: The Gaussian function of a RBF network is non-zero in the finite range of the input space. Compared with BP and other networks, it has a fast learning rate, which is suitable for the real-time requirements of neural network adaptive control [7].

In order to illustrate the control strategy conveniently, a concise block diagram of the proposed control system is shown in Figure 1.

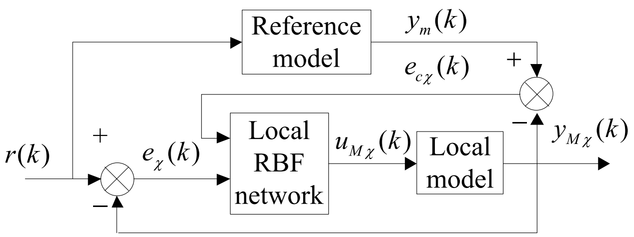


FIGURE 2. Local RBF network control for its local model.

where $y_m(k)$ is the output of the reference model, $e_c(k)$ is the system output tracking error, $\boldsymbol{y_M}(k) = \left[y_{M_1}(k), \cdots, y_{M_N}(k)\right]^{\mathrm{T}}$ and $\boldsymbol{u_M}(k) = \left[u_{M_1}(k), \cdots, u_{M_N}(k)\right]^{\mathrm{T}}$ are the output vectors of the local models and fixed local controllers, $\boldsymbol{W_o}(k)$ is the network factor values after fusion.

A. MODEL SET AND RBF NETWORK CONTROLLERS

Based on the system operation data, several local models can be established by clustering method [31] to form the model set $\Omega = \left\{M_\chi | \chi = 1, 2, \ldots, N\right\}$, where M_χ is the χth local model. Besides, the Vinnicombe distance termed as ν-gap metric [32] is presented to calculate the distance among models. In fact, the purpose of model set is to effectively cover and approximate the plant system.

Remark 4: The constructing process of model set is omitted, because it is not the focus in this study. For conciseness, it includes a finite number of fixed parameter models.

Based on the model set, multiple local RBF network controllers can be designed well for the local models. Correspondingly, the controller set is $C = \left\{C_\chi | \chi = 1, 2, \cdots, N\right\}$, where C_χ is the χth fixed local RBF network controller. In order to reduce the dynamic performance impacts on the plant system caused by the local controllers, we design a RBF network adaptive controller that can be reassigned initial values to control the plant all the time. When the plant parameters jump, the good factor values are reasonably assigned to the RBF network adaptive controller as its initial values. The good factor values are obtained by fusing the fixed local RBF network controllers. Thus, it can quickly converge to the ideal values.

Then, a same design method is adopted for the local RBF network controllers. The structure of resulting local system is shown in Figure 2 [7].

where $e_{c\chi}(k)$ is the tracking error for the local control system, $\chi = 1, 2, \ldots, N$.

For the local control system, the control objective requires that the output of the local model $y_{M\chi}(k)$ can stably track the output of the reference model $y_m(k)$. Then the tracking error is

$$e_{c\chi}(k) = y_m(k) - y_{M\chi}(k) \tag{3}$$

The control objective function is

$$E_{M\chi}(k) = \frac{1}{2} e_{c\chi}(k)^2 \tag{4}$$

The local control law is the output of the RBF network, that is

$$u_{M\chi}(k) = \boldsymbol{W}_\chi^{\mathrm{T}} \cdot \boldsymbol{H}_\chi \tag{5}$$

where $\boldsymbol{W}_\chi = \left[w_1 \cdots w_m\right]^{\mathrm{T}}$ is the connection weight vector, and $\boldsymbol{H}_\chi = \left[h_1 \cdots h_m\right]^{\mathrm{T}}$ is the radial basis vector, m is the number of hidden layer neurons in RBF network, w_j is the connection weight between the jth hidden layer neuron and the output layer of the network, h_j is the output of the jth hidden layer neuron.

$$h_j = \exp\left(-\frac{\left\|\boldsymbol{X}_\chi - \boldsymbol{C}_{\chi j}\right\|^2}{2b_j^2}\right) \tag{6}$$

where $\boldsymbol{X}_\chi = \left[x_1 \cdots x_n\right]^{\mathrm{T}}$ is the input vector, b_j is the basis width factor of the jth node and $b_j > 0$, $\boldsymbol{C}_{\chi j}$ is the center vector of the jth node for the χth local RBF network controller, $\boldsymbol{C}_{\chi j} = \left[c_{j1}, \cdots, c_{ji}, \cdots, c_{jn}\right]^{\mathrm{T}}$, $\boldsymbol{B}_\chi = [b_1, \cdots, b_m]^{\mathrm{T}}$, $j = 1, \cdots, m$.

The learning rules of RBF network factors are designed as follows.

Based on the gradient descent method and chain rule [7], the b_j and c_{ji} can be obtained.

$$\begin{aligned}\Delta b_j(k) &= -\eta \frac{\partial E_{M\chi}(k)}{\partial b_j} \\ &= \eta e_{c\chi}(k) \frac{\partial y_{M\chi}(k)}{\partial u_{M\chi}} w_j h_j \frac{\left\|\boldsymbol{X}_\chi - c_{ji}\right\|^2}{b_j^3}\end{aligned} \tag{7}$$

$$b_j(k) = b_j(k-1) + \Delta b_j(k) + \alpha\left(b_j(k-1) - b_j(k-2)\right) \tag{8}$$

$$\begin{aligned}\Delta c_{ji}(k) &= -\eta \frac{\partial E_{M\chi}(k)}{\partial c_{ji}} \\ &= \eta e_{c\chi}(k) \frac{\partial y_{M\chi}(k)}{\partial u_{M\chi}} w_j h_j \frac{\boldsymbol{X}_\chi - c_{ji}}{b_j^2}\end{aligned} \tag{9}$$

$$\begin{aligned}c_{ji}(k) = {} & c_{ji}(k-1) + \Delta c_{ji}(k) \\ & + \alpha\left(c_{ji}(k-1) - c_{ji}(k-2)\right)\end{aligned} \tag{10}$$

where η is the learning rate, α is the momentum factor, $\partial y_{M\chi}(k) / \partial u_{M\chi}$ is Jacobian value.

The learning process can be consider as an optimization problem to minimize the objective function. For the conventional optimization problem, the optimal value of control objective function is not known. Thus, the equations that partial derivatives equal to zero are solved to learn the optimal value. The gradient descent method is based on this idea, which may result in the low learning rate, because $\Delta \boldsymbol{W}_\chi(k)$ in each step is calculated based on gradient. Fortunately, the neural network has specific training objective, that is the optimal value of $E_{M\chi}(k)$ is zero in the ideal case. Therefore, the goal of the learning rule is to make it be zero.

The neural network adjusts its connection weights to make $E_{M\chi}(k)$ reach the optimal value from the current value as soon as possible. To learning $\Delta \boldsymbol{W}_\chi(k)$ in each step, novel learning rules based on the difference value between current value of

$E_{M\chi}(k)$ and its optimal value for general neural networks are developed as follows.

When $E_{M\chi}(k) < \sigma$, where σ is a threshold value, the learning rules are designed based on gradient descent method, that is

$$\Delta w_j(k) = \eta e_{c\chi}(k)\partial y_{M\chi}(k)/\partial u_{M\chi}h_j + \alpha\left(w_j(k-1) - w_j(k-2)\right) \quad (11)$$

When $E_{M\chi}(k) \geq \sigma$, two novel learning methods are developed.

Learning rule 1: In order to make $E_{M\chi}(k)$ reach the optimal value from the current value, the proportional-derivative (PD) idea is introduced to learn $\Delta w_j(k)$, that is

$$\Delta w_j(k) = \operatorname{sign}\left(e_{c\chi}(k)\right)\left|K_P E_{M\chi}(k) + K_D \Delta E_{M\chi}(k)\right| \quad (12)$$

where K_P and K_D are proportional and derivative coefficients.

Learning rule 2: A correlation between $\Delta w_j(k)$ and objective function value $E_{M\chi}(k)$ is established as follow

$$\Delta w_j(k) = \operatorname{sign}\left(e_{c\chi}(k)\right)\left|\frac{K_1}{K_2 + e^{-E^2_{M\chi}(k)}}\right| \quad (13)$$

where K_1 and K_2 are adjustable coefficients, $K_1 > 0, K_2 > 0$.

Then, the connection weight can be obtained,

$$w_j(k) = w_j(k-1) + \Delta w_j(k)$$

Remark 5: Only the ideal situation without disturbance is considered in this paper. Due to the influence of noise, the setting of the optimal value needs to be considered according to the actual situation in the application. Besides, the learning rules are not based on any restrictions and specifications of RBF networks, it is suitable for general neural networks.

Remark 6: Both learning rules attempt to directly adjust the objective function to its optimal value based on the real-time difference value, instead of calculating partial derivatives in the traditional methods. Learning rule 1 introduces PD idea (not the PD control algorithm) to learning neural network connection weight, while learning rule 2 learns it by establishing the correlation between neural network connection weight and objective function value.

B. PERFORMANCE INDEX AND WEIGHTING ALGORITHM

As shown in Figure 1, the performance index and a simple weighting algorithm can be designed based on the output errors $e_{M\chi}(k)$ between local models and the plant [30].

Considering the combination of historical and current output errors, the following performance index is proposed.

$$l'_i(k) = \zeta + \beta\left[y(k) - y_{M\chi}(k)\right]^2 + \gamma\sum_{j=1}^{L}\theta^j\left[y(k-j) - y_{M\chi}(k-j)\right]^2 \quad (14)$$

where ζ is a small positive value to prevent that the error performance index is equal to zero, when it is as the denominator; β and γ indicate the influences of the present and historical moments on the performance index, $\beta > 0, \gamma > 0$; $0 < \theta \leq 1$ is forgetting factor; L is the time length of influence; $\chi = 1, 2, \cdots, N$.

Then, the weights for local controllers can be obtained.

$$l'_{\min}(k) = \min l'_\chi(k) \quad (15)$$

$$l_\chi(k) = \begin{cases} l_\chi(k-1) & g_\chi(k) = 1 \\ l_\chi(k-1)g_\chi(k)^{\operatorname{ceil}\left(\frac{1}{1-g_\chi(k)}\right)} & g_\chi(k) < 1 \end{cases} \quad (16)$$

$$p_\chi(k) = \frac{l_\chi(k)}{\sum_{\chi=1}^{N} l_\chi(k)} \quad (17)$$

where $l_\chi(k)$ is the iterative process variable for weighting algorithm, $g_\chi(k) = l'_{\min}(k)\big/ l'_\chi(k)$, ceil$(*)$ is the ceiling function that generates the smallest integer not less than the value $(*)$, and $p_\chi(k)$ is the weight value of the χth local controller.

Remark 7: All the limit operations in this paper are in the sense of probability one. Thus, there is a weighting calculation situation should be stated, that is when the weight of a local model is 1, the others are zero. Then, the unselected models cannot be recalculated. Therefore, a threshold value should be designed to prevent weights from being zero in the application.

Remark 8: The time complexity is $O(N \cdot m^2 \cdot n)$, which means that the time complexity is related to the number of local models, hidden and input layer neurons of RBF network.

IV. STABILITY ANALYSIS

Based on weighting algorithm convergence and VES theory [29], this section shows the stability analysis of the RBF networks based WMMAC system of parameter jump plant.

Primarily, it should be explained that the stability of control system means the boundedness of its input-output signals and the convergence of its performance index to that of the local control systems [29].

Lemma 1 [30]: If a weighted multiple model adaptive control system has the following properties:

(1) There is a model, say $M_z \in \Omega$, which is closest to the true plant in the following sense with probability 1

$$\sum_{r=1}^{k} e_{Mz}(r)^2 < \sum_{r=1}^{k} e_{M\chi}(r)^2, \quad \forall k \geq d,\ \chi \neq z$$

(2) Each local controller is well defined such that C_χ is stabilizing M_χ, and the resulting closed-loop system $\{C_\chi, M_\chi\}$ is tracking the reference signal $y_m(k)$;

(3) The model closest to the true plant, that is M_z, its output approximates the true plant output in the sense that

$$\left\|e'(k)\right\| = o(\|\phi(k-d)\|)$$

Then it is stable and convergent.

Theorem 1: Regarding the RBF networks based WMMAC system of parameter jump plant in Figure 1, it is bounded-input bounded-output stable if the following properties are satisfied:

(1) Variation range of plant parameters can be covered and approximated by model set Ω, and the approximation error is bounded;

(2) Each local RBF network controller is well defined, such that the neural network is convergent in finite time;

(3) Among any jumping time sequence, there is a model $M_z \in \Omega$ closest to the current plant in the following sense with probability one

$$\begin{cases} \sum_{r=1}^{k} \| e_{Mz}(T_l+r) \|^2 < \sum_{r=1}^{k} \| e_{M\chi}(T_l+r) \|^2, \\ \qquad \forall k \geq d+1 \\ \frac{1}{k}\sum_{r=1}^{k} \| e_{Mz}(T_l+r) \|^2 \to S_z, \\ \qquad \frac{1}{k}\sum_{r=1}^{k} \| e_{M\chi}(T_l+r) \|^2 \to S_\chi \end{cases}$$

where d is the system delay, S_z is a constant, S_χ is a positive constant or infinity, and $S_z < S_\chi$, $\chi \neq z$, T_l, $l = 0, 1, 2, \cdots$ is parameter jumping time sequence.

Proof: Considering the complexity of Lyapunov function and its derivative for the stability analysis of WMMAC system, Lyapunov method is not used in this paper. The VES theory is adopted, which is relatively simple. The proof is as follows.

Firstly, the convergence of the weighting algorithm is ensured easily. According to property (3) in Theorem 1, we obtain

$$l'_z(k) = l'_{\min}(k) \tag{18}$$

Then,

$$\begin{cases} g_z(k) = l'_{\min}(k)/l'_z(k) = 1 \\ g_\chi(k) = l'_{\min}(k)/l'_\chi(k) < 1 \end{cases} \tag{19}$$

where $\chi \neq z$.

Thus, $\lim_{k\to\infty} g_\chi(k)^{\mathrm{ceil}\left(\frac{1}{1-g_\chi(k)}\right)} = \frac{1}{e} < 1$. Considering Equations (16) (17), $p_z(k)$ tends to 1 and $p_\chi(k)$, $\chi = 1, \cdots, N$, $\chi \neq z$ tend to 0.

Secondly, according to the learning rules, one is inspired by PD idea, and the other one is bounded,

$$\begin{cases} \lim\limits_{E_{M\chi}(k)\to 0} \left|\Delta w_{2j}(k)\right| = \lim\limits_{E_{M\chi}(k)\to 0} \dfrac{K_1}{K_2 + e^{-E_{M\chi}(k)^2}} = \dfrac{K_1}{1+K_2} \\ \lim\limits_{E_{M\chi}(k)\to \infty} \left|\Delta w_{2j}(k)\right| = \lim\limits_{E_{M\chi}(k)\to \infty} \dfrac{K_1}{K_2 + e^{-E_{M\chi}(k)^2}} = \dfrac{K_1}{K_2} \end{cases} \tag{20}$$

that is $K_1/(1+K_2) < \left|\Delta w_{2j}(k)\right| < K_1/K_2$. Based on the gradient descent methods, chain rule, and the Equation (20), the convergence of the RBF network is guaranteed.

In fact, the structures of the RBF network adaptive controller and fixed local RBF network controllers are designed in the same method. The RBF networks based WMMAC strategy and the weighted multi-model adaptive control strategy [30] are equivalent in the sense of input and output. Then, the VES of the control system is given in Figure 3.

where $e'(k)$ and $\Delta u'(k)$ are the equivalent output and control errors. For the RBF network adaptive controller, the influence of other fixed local controllers on system performance is included in $\Delta u'(k)$.

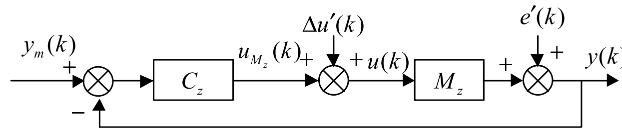


FIGURE 3. VES for the control system.

According to the property (2) in Theorem 1 and the Lemma 1, the stability can be obtained for one time period as shown in Equation (2).

Finally, parameters jump cannot be infinitely fast in reality, which is regarded as slow switching among constants. Based on the switching system theory [33], the RBF networks based WMMAC system of parameter jump plant is globally stable.

V. NUMERICAL SIMULATION

In this section, two examples are studied to illustrate the effectiveness of the proposed RBF networks based WMMAC strategy.

A. EXAMPLE 1

In order to analyze the advantages of the proposed algorithm concisely, a traditional nonlinear discrete-time plant [7] is considered,

$$y(k) = (-0.10y(k-1) + u(k-1) + \varepsilon(k))/(1 + y(k-1)^2) \tag{21}$$

where $\varepsilon(k)$ is a time-varying parameter,

$$\varepsilon(k) = \begin{cases} 0 & 0 < k \leq 1250 \\ 3 & 1250 < k \leq 3000 \end{cases} \tag{22}$$

and the sampling time is $t_s = 1$ms.

For simplicity, it's known that there are two local models to cover the parameter variations, whose parameters have been obtained as $\varepsilon(k) = 0$ in model 1 and $\varepsilon(k) = 3$ in model 2. Then, the corresponding RBF network controllers set can be designed. Thus, the RBF networks based WMMAC system shown in Figure 1 is obtained.

1) PART 1: LOCAL RBF NETWORK CONTROL SYSTEM

For $y_{M1}(k) = (-0.10y_{M1}(k-1) + u_{M1}(k-1))/(1 + y_{M1}(k-1)^2)$, the RBF network controller is designed as shown in Figure 2. Reference model is $y_m(k) = 1$ for $R \leq k \leq R + 500$ and $y_m(k) = -1$ for $R + 500 \leq k \leq R + 1000$, $R = 0, 1000$; the driving signal is $r(k) = 0.50\sin(2\pi k \cdot t_s)$. Then, $r(k)$, $e_{c1}(k)$ and $y_{M1}(k)$ are selected as the inputs of the RBF network with 3-6-1 structure. Based on the learning rule 1 as shown in Equation (11), we can obtained the values of neural network factors $\eta = 0.35$, $\alpha = 0.05$, $K_P = 0.4$ and $K_D = 0.001$ by simulation debugging.

The simulation results, including the local model output $y_{M1}(k)$ following reference signal $y_m(k)$, the tracking error and the control signal $u_{M1}(k)$ are shown in Figure 4 and Figure 6. As shown in Figure 4, the system takes about 15 steps to track reference signal, which illustrates the control system can realize target tracking.

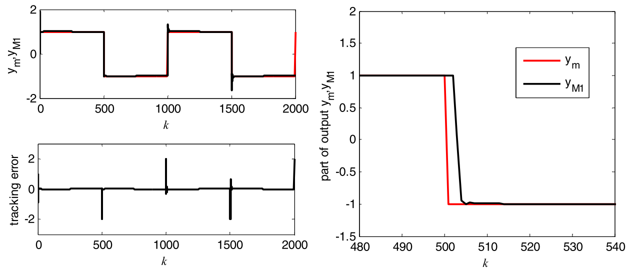


FIGURE 4. Local model output based on learning rule 1.

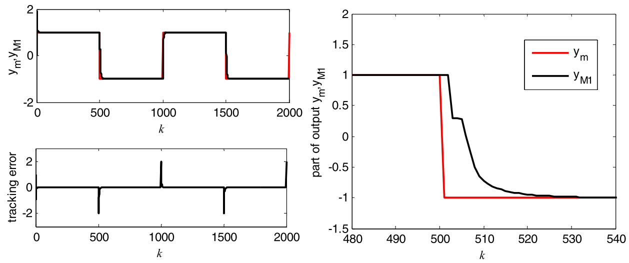


FIGURE 5. System output using gradient descent method.

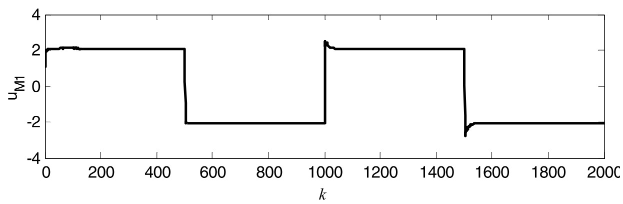


FIGURE 6. Control signal of the local system with learning rule 1.

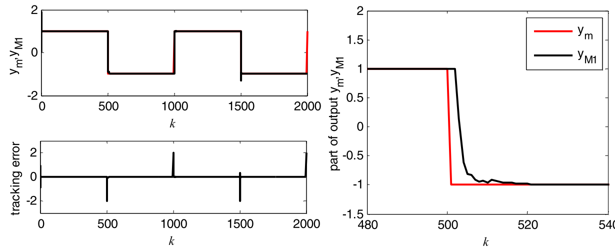


FIGURE 7. Local model output based on learning rule 2.

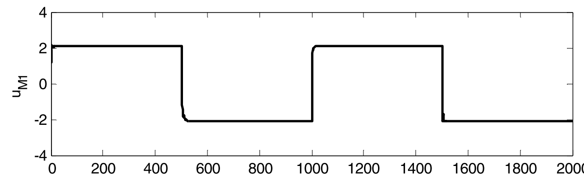


FIGURE 8. Control signal of the local system with learning rule 2.

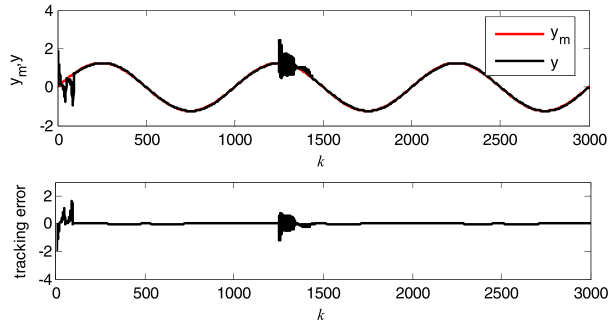


FIGURE 9. RBF networks based WMMAC system output.

In order to demonstrate the improvement of learning rate, the simulation results using gradient descent method are given in Figure 5. It shows that the system takes about 30 steps to track the output of the reference model. Comparing the output curves of Figure 4 and 5, the increase of learning rate is obvious, which strongly illustrates the effectiveness of the learning rule 1.

Remark 9: The gradient descent method is widely used in machine learning, but its learning rate is slow for real-time adaptive control system. Then, we design the new learning rules to improve it. Thus, the performance comparison between them is reasonable.

The control signal of the local RBF network control system based on learning rule 1 is shown in Figure 6.

On the other side, consider the same local model 1 based on the learning rule 2 in Equation (12) (13), the control system simulation can be obtained. By simulation debugging, the values of neural network factors are $\eta = 0.35$, $\alpha = 0.05$, $K_1 = 0.085$ and $K_2 = 0.1$. The simulation results including the closed-loop output $y_{M1}(k)$ tracking reference signal $y_m(k)$ and the control signal $u_{M1}(k)$ are shown in Figure 7 and Figure 8.

According to the Figure 7, the system takes about 20 steps to track the output of the reference model. Comparing the output curves of Figure 7 and 5, we can clearly see the increase of learning rate, which illustrates the effectiveness of the learning rule 2.

Remark 10: Two new learning rules are both to improve the learning rate of RBF network in different methods. Since they have the problem of coefficient selection, which is explored by numerical simulations, we cannot ensure the best performance is obtained in theory. Thus, their performances are not compared in this study.

2) PART 2: RBF NETWORKS BASED WMMAC SYSTEM

Consider the nonlinear discrete-time plant in Equation (21) (22), the multi-model adaptive controller can be developed based on weighting algorithm as shown in Section III and the simulation in Part 1. The RBF network adaptive controller that can be reassigned initial values is designed by the same method, whose reference model is $y_m(k) = 0.6y_m(k-1) + r(k)$, where the driving signal is $r(k) = 0.50\sin(2\pi k \cdot t_s)$.

The simulation results are presented as follows. As shown in Figure 9, the system output can closely track the reference output without frequent vibration, even when the plant parameter jumps at $k = 1250$, whose output errors are relatively small. In addition, to illustrate the improvement of control performance, the output of plant controlled by conventional single RBF network controller is shown in Figure 10. It is obvious that the control performance of the RBF networks based WMMAC strategy is better, especially when the plant

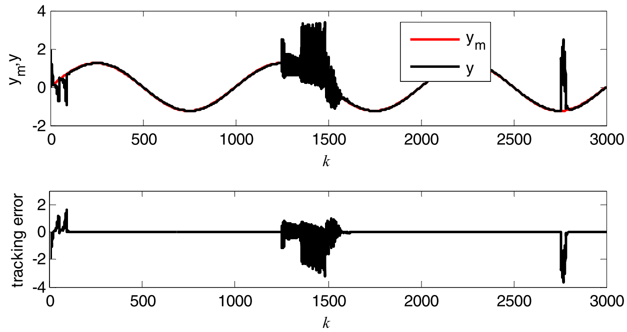


FIGURE 10. System output based on conventional controller.

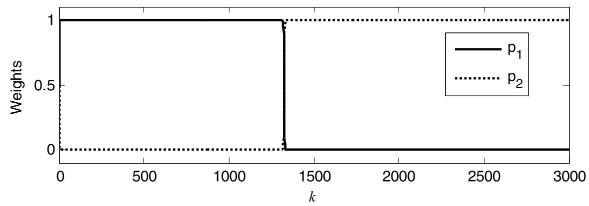


FIGURE 11. Weights for the multiple local controllers.

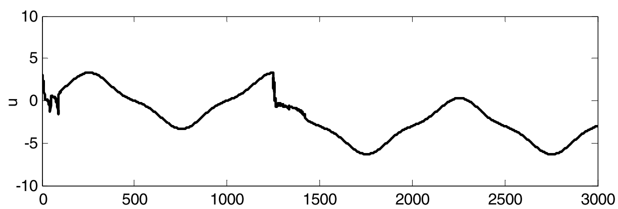


FIGURE 12. Global control signal for the real plant.

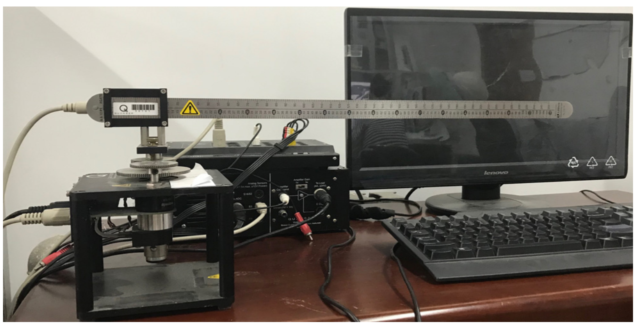


FIGURE 13. The rotary flexible link system.

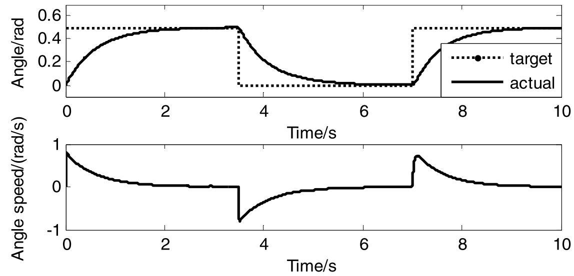


FIGURE 14. Joint angle tracking.

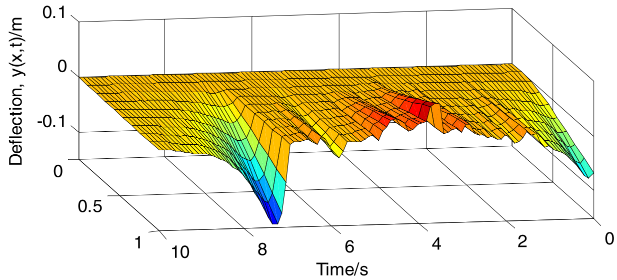


FIGURE 15. Flexible deflection of rotary link.

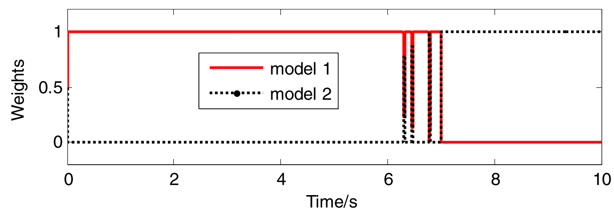


FIGURE 16. Weights for local controllers.

parameters change largely at $k = 1250$. The transient performance is improved. Besides, the weights of multiple local RBF network controllers are shown in Figure 11, whose convergence process takes about 80 steps when the plant parameter jumps. The Figure 12 shows the global control signal for the real plant system.

B. EXAMPLE 2

In this subsection, a single-link rotary flexible arm with jumping parameter is studied, as shown in Figure 13. This plant is used for the study of angle control and vibration suppression.

Based on the real rotary flexible link system, the dynamic model can be built as follow [34]

$$\begin{cases} \rho\ddot{w}(x,t) = -EI(t)\cdot w''''(x,t) \\ \tau(t) = I_h\ddot{\theta}(t) - EI(t)\cdot w''(0,t) \\ F(t) + d(t) = m\ddot{w}(L,t) - EI(t)\cdot w'''(L,t) \\ w(0,t) = w''(L,t) = 0 \end{cases} \tag{23}$$

where $\rho = 0.2\ \mathrm{kg\cdot m^{-1}}$, is the uniform mass per unit length; $w(x,t)$ is the arc approximation at point x of the link; $\tau(t)$ is the control torque of joint motor; $F(t)$ is the tip control input; $d(t) = 0.1\cdot\mathrm{randn}(1)$ is the disturbances; $I_h = 0.1\ \mathrm{kg\cdot m^2}$, is the joint motor inertia; $m = 0.1$ kg, is the tip payload mass; $\theta(t)$ is the joint angle; $L = 1$m, is the link length; $EI(t)$ jumping from 3 to 0.5 $\mathrm{N\cdot m^2}$ at 7th s, is the link flexural rigidity; $(\dot{*}) = \partial(*)/\partial t$, $(*)' = \partial(*)/\partial x$.

To carry out the simulation, the dynamic model is discretized, the steps of time and link space are $\Delta t = 0.5$ms, $\Delta x = 0.01$m. The reference angle is that $\theta_d = 0.5$rad, $0 \le t < 3.5$ and $7 \le t < 10$, $\theta_d = 0$rad, $3.5 \le t < 7$.

To verify the method proposed in this paper, two local models are built based on the jumping parameter $EI(t)$, that is $EI_1(t) = 3$ of model 1 and $EI_2(t) = 0.5$ of model 2. As the angle control is simple relatively, the $\tau(t)$ can be design as PD controller, $k_p = 50$ and $k_d = 30$. The $F(t)$ designed by RBF networks based WMMAC is to suppress the vibration.

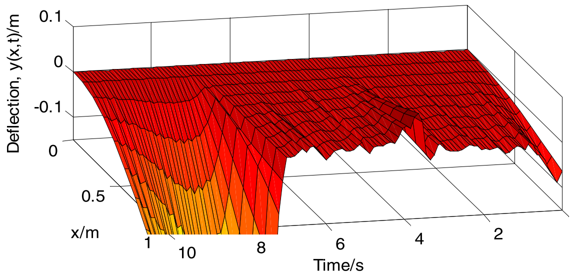


FIGURE 17. Flexible deflection using conventional controller.

Learning rule 1 is adopted, whose factors are the same as the values in Part 1, and RBF network owns 3 inputs and 62 hidden layer neurons.

Figures 14 to 16 show the angle control and vibration suppression results by using the RBF networks based WMMAC. The flexible deformation of the link with a single conventional RBF network controller is shown in Figure 17. The angle position and the reference angle are shown in Figure 14, which shows that the output tracking performance is satisfactory. Weights for two local controllers are shown in Figure 16. From the comparison between Figure 15 and 17, it can be seen that the proposed method in this paper receives better vibration suppression effect, especially after the system parameter jumps at 7th s.

VI. CONCLUSIONS AND FUTURE WORKS

This paper is concerned with the tracking control problem of nonlinear systems subjected to the large parameter uncertainties. By combining WMMAC and RBF network adaptive control based on novel learning rules, the control system can receive better transient performance. The simulation results illustrate the efficiency of the proposed method. The mainly advantages of the proposed method are that (1) the RBF networks based WMMAC can improve the transient performance when parameters jumping; (2) the novel neural network learning rules can improve the learning rate.

The future works focus on dynamic optimization of model set, the improvements of neural network approximation ability and learning rules, time complexity problem of neural networks, and the application of real control system.

REFERENCES

[1] D. T. Magill, "Optimal adaptive estimation of sampled stochastic processes," *IEEE Trans. Autom. Control*, vol. AC-10, no. 4, pp. 434–439, Oct. 1965.

[2] D. G. Lainiotis, "Partitioning: A unifying framework for adaptive systems, I: Estimation," *Proc. IEEE*, vol. 64, no. 8, pp. 1126–1143, Aug. 1976.

[3] D. G. Lainiotis, "Partitioning: A unifying framework for adaptive systems, II: Control," *Proc. IEEE*, vol. 64, no. 8, pp. 1182–1198, Aug. 1976.

[4] M. Athans, D. Castanon, K. Dunn, C. Greene, W. Lee, N. Sandell, and A. Willsky, "The stochastic control of the F-8C aircraft using a multiple model adaptive control (MMAC) method—Part I: Equilibrium flight," *IEEE Trans. Autom. Control*, vol. AC-22, no. 5, pp. 768–780, Oct. 1977.

[5] Z. Zhou, *Machine Learning*. Beijing, China: Tsinghua Univ. Press, 2016.

[6] J. Moody and C. Darken, "Learning with localized receptive fields," in *Connectionist Models Summer School*, D. S. Touretzky and G. T. Hinton, Eds. Burlington, MA, USA: Morgan Kauffman, 1988, pp. 133–134.

[7] J. Liu, *Intelligent Control*, 4th ed. Beijing, China: Publishing House of Electronics Industry, 2017.

[8] W. He, Y. Dong, and C. Sun, "Adaptive neural impedance control of a robotic manipulator with input saturation," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 3, pp. 334–344, Mar. 2016.

[9] J. Lian and J. Wang, "Passivity of switched recurrent neural networks with time-varying delays," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 2, pp. 357–366, Feb. 2015.

[10] Z. Wang, S. Ding, Q. Shan, and H. Zhang, "Stability of recurrent neural networks with time-varying delay via flexible terminal method," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2456–2463, Oct. 2017.

[11] Z. Li, Y. Xia, C.-Y. Su, J. Deng, J. Fu, and W. He, "Missile guidance law based on robust model predictive control using neural-network optimization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 8, pp. 1803–1809, Aug. 2015.

[12] S. Li, Y. Zhang, and L. Jin, "Kinematic control of redundant manipulators using neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2243–2254, Oct. 2017.

[13] J. Zhu, P. Xie, C. Huang, and B. Qi, "Ball and beam system position control based on feed forward neural network control," *Control Eng. China*, vol. 24, no. 6, pp. 1186–1193, 2017.

[14] C. Li, G. Wang, Y. Li, Y. Fan, and A. Xu, "Direct adaptive neural network control of switched reluctance motors," *Electr. Mach. Control*, vol. 22, no. 1, pp. 29–36, 2018.

[15] M.-L. Zhou, J.-A. Zhang, Y. Zhao, and W. Gao, "Hybrid control for piezoelectric micro positioning platform based on BP neural network and expert fuzzy control," *Control Decis.*, vol. 33, no. 1, pp. 95–100, 2018.

[16] M. Chen, P. Shi, and C. Lim, "Adaptive neural fault-tolerant control of a 3-DOF model helicopter system," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 46, no. 2, pp. 260–270, Feb. 2016.

[17] J. Liu, S. Duan, T. Li, and L. Wang, "Design of RBF neural network control system based on spintronic memristor for robotic manipulator," *Chin. J. Sci. Instrum.*, vol. 39, no. 8, pp. 212–219, 2018.

[18] S. Wang, H. Yu, J. Yu, J. Na, and X. Ren, "Neural-network-based adaptive funnel control for servo mechanisms with unknown dead-zone," *IEEE Trans. Cybern.*, to be published. doi: 10.1109/TCYB.2018.2875134.

[19] S. Wang, J. Na, and X. Ren, "Rise-based asymptotic prescribed performance tracking control of nonlinear servo mechanisms," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 12, pp. 2359–2370, Dec. 2018.

[20] Z. Yao, J. Yao, and W. Sun, "Adaptive rise control of hydraulic systems with multilayer neural-networks," *IEEE Trans. Ind. Electron.*, to be published. doi: 10.1109/TIE.2018.2886773.

[21] Y. J. Duan, Y. S. Lv, J. Zhang, X. L. Zhao, and F. Y. Wang, "Deep learning for control: The state of the art and prospects," *Acta Autom. Sinica*, vol. 42, no. 5, pp. 643–654, 2016.

[22] L. Chen and K. S. Narendra, "Nonlinear adaptive control using neural networks and multiple models," *Automatica*, vol. 37, no. 8, pp. 1245–1255, Aug. 2001.

[23] Y. Fu, T. Chai, and H. Yue, "Intelligent control using multiple models and neural networks," *Int. J. Adapt. Control Signal Process.*, vol. 22, no. 5, pp. 495–509, Jun. 2008.

[24] Y. Fu and T. Chai, "Nonlinear multivariable adaptive control using multiple models and neural networks," *Automatica*, vol. 43, no. 6, pp. 1101–1110, Jun. 2007.

[25] Y. Fu and T. Chai, "Indirect self-tuning control using multiple models for non-affine nonlinear systems," *Int. J. Control*, vol. 84, no. 6, pp. 1031–1040, 2011.

[26] M. Huang, X. Wang, and Z. Wang, "Nonlinear adaptive controller using multiple models and neural networks based on zero order proximity boundedness," *Control Decis.*, vol. 28, no. 9, pp. 1315–1321, 2013.

[27] X.-L. Li, C. Jia, D.-X. Liu, and D.-W. Ding, "Nonlinear adaptive control using multiple models and dynamic neural networks," *Neurocomputing*, vol. 136, pp. 190–200, Jul. 2014.

[28] X.-L. Li, C. Jia, K. Wang, and J. Wang, "Trajectory tracking of nonlinear system using multiple series-parallel dynamic neural networks," *Neurocomputing*, vol. 168, pp. 1–12, Nov. 2015.

[29] W. Zhang, "Stable weighted multiple model adaptive control: Discrete-time stochastic plant," *Int. J. Adapt. Control Signal Process.*, vol. 27, no. 7, pp. 562–581, 2013.

[30] W. Zhang, "Further results on stable weighted multiple model adaptive control: Discrete-time stochastic plant," *Int. J. Adapt. Control Signal Process.*, vol. 29, no. 12, pp. 1497–1514, 2015.

[31] M. Huang, X. Wang, and Z.-L. Wang, "Multiple models adaptive control based on time series for a class of nonlinear systems," *Acta Autom. Sinica*, vol. 39, no. 5, pp. 581–586, 2013.

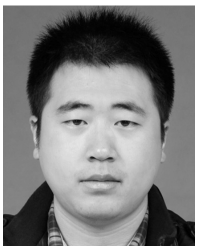
[32] H. Mahdianfar, S. Ozgoli, and H. R. Momeni, "Robust multiple model adaptive control: Modified using ν-gap metric," *Int. J. Robust Nonlinear Control*, vol. 21, pp. 2027–2063, Dec. 2011.

[33] R. Shorten, F. Wirth, O. Mason, K. Wulff, and C. King, "Stability criteria for switched and hybrid systems," *SIAM Rev.*, vol. 49, no. 4, pp. 545–592, 2017.

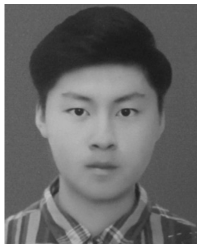
[34] W. He and C. Sun, "Boundary feedback stabilisation of a flexible robotic manipulator with constraint," *Int. J. Control*, vol. 89, no. 3, pp. 635–651, 2016.

QING LI received the B.E. degree from the North China University of Science and Technology, Tangshan, China, in 1993, and the Ph.D. degree in control theory and its applications from the University of Science and Technology Beijing, Beijing, China, in 2000, where he is currently a Professor with the School of Automation and Electrical Engineering. He was a Visiting Scholar with Ryerson University, Toronto, Canada, from 2006 to 2007. His research interests include intelligent control and intelligent optimization.

YUHANG YANG is currently pursuing the bachelor's degree with the University of Science and Technology Beijing, China. He has attended the student research training program on controller design for unmanned aerial vehicles, in 2018. His current research interest includes adaptive control.



YUZHEN ZHANG received the B.E. degree from the China University of Petroleum, Dongying, China, in 2011, and the M.E. degree in control engineering from the University of Science and Technology Beijing, Beijing, China, in 2014, where he is currently pursuing the Ph.D. degree in control theory and its applications with the School of Automation and Electrical Engineering. His research interest includes adaptive control and its applications.



YUKUN LIU is currently pursuing the bachelor's degree with the University of Science and Technology Beijing, China. He has attended the student research training program on path planning for manipulator, in 2018. His current research interest includes intelligent optimization.

WEICUN ZHANG received the Ph.D. degree in control theory and applications from Tsinghua University, Beijing, China, in 1993. He is currently an Associate Professor with the School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing, China. His research interests include adaptive control, multiple model adaptive control, and intelligent control and their applications.

JIARUI CUI received the B.E. degree in electric engineering from Liaocheng University, China, in 2004, and the M.E. degree in signal and information processing from the Harbin University of Science and Technology, China, in 2007, and the Ph.D. degree from the University of Science and Technology Beijing, China, in 2011. He was a Visiting Scholar with the University of Strathclyde, Britain, in 2010, 2016, and 2018. He is currently a Senior Engineer with the University of Science and Technology Beijing. His research interests include two-dimensional stochastic control and its applications in machine vision, smart grids and V2G, and embedded systems and its applications in industrial fields.