

## Research Article

# Neural Learning Control of Flexible Joint Manipulator with Predefined Tracking Performance and Application to Baxter Robot

Min Wang, Huiping Ye, and Zhiguang Chen

*School of Automation Science and Engineering, Guangzhou Key Laboratory of Brain Computer Interaction and Applications, South China University of Technology, Guangzhou 510641, China*

Correspondence should be addressed to Min Wang; [auwangmin@scut.edu.cn](mailto:auwangmin@scut.edu.cn)

Received 20 July 2017; Accepted 14 September 2017; Published 31 October 2017

Academic Editor: Yanan Li

Copyright © 2017 Min Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper focuses on neural learning from adaptive neural control (ANC) for a class of flexible joint manipulator under the output tracking constraint. To facilitate the design, a new transformed function is introduced to convert the constrained tracking error into unconstrained error variable. Then, a novel adaptive neural dynamic surface control scheme is proposed by combining the neural universal approximation. The proposed control scheme not only decreases the dimension of neural inputs but also reduces the number of neural approximators. Moreover, it can be verified that all the closed-loop signals are uniformly ultimately bounded and the constrained tracking error converges to a small neighborhood around zero in a finite time. Particularly, the reduction of the number of neural input variables simplifies the verification of persistent excitation (PE) condition for neural networks (NNs). Subsequently, the proposed ANC scheme is verified recursively to be capable of acquiring and storing knowledge of unknown system dynamics in constant neural weights. By reusing the stored knowledge, a neural learning controller is developed for better control performance. Simulation results on a single-link flexible joint manipulator and experiment results on Baxter robot are given to illustrate the effectiveness of the proposed scheme.

## 1. Introduction

Due to the great demands in industrial applications, the tracking control problem for flexible joint robot (FJR) manipulator has attracted much attention in recent years. Unlike rigid joint robot, the joint flexibility of FJR results in complex control situation, so that the control problem of FJR becomes much more difficult. In the past few decades, lots of efforts have been made on the research of FJR systems. Based on the model of FJR presented in [1], multifarious nonlinear control methods are presented such as backstepping method [2–4], sliding-mode control [5–8], switching control [9], fuzzy control [10], and neural network control [11, 12]. In consideration of the problem caused by the inherent structure of FJR under practical circumstance, such as friction, time delay, and variable stiffness, some researchers proposed effective strategies to solve such problem [13–15]. Moreover, the teleoperation control method is also widely used in robot research [16, 17].

The backstepping control [18] is known as one of the popular method for designing the control scheme of FJR. Nevertheless, it should be pointed out that this method has a drawback called “explosion of complexity” [19]. This problem generally occurs in the design of neural networks (NNs) during backstepping procedure. To overcome this problem, some researchers used the intermediate variables as neural inputs to reduce the dimension of neural network input vector [20]. The method in [20] did work well but the problem remained unsolved. Then other researchers proposed a dynamic surface control (DSC) method by introducing a first-order filter at each step of the backstepping procedure [21]. Due to the property of DSC method, many researchers presented their control schemes combined with DSC method [22–27]. In [24], a new robust output feedback control approach for flexible joint electrically driven robots via the observer-based dynamic surface method was proposed, which only requires position measurement of the system.

Besides, the transient and steady-state tracking performance constraints of system's output are an important issue that needs to be taken into consideration [28, 29]. According to the practical operating environment, the manipulator is not only demanded to trace the reference trajectory accurately but also required to keep the tracking error within a specified range. To satisfy this condition, a performance function transformation was used to convert the "constrained" system into the "unconstrained" one [30]. Based on the idea in [30], further researches on prescribed performance for a variety of systems are proposed [31–36]. Authors in [31, 32] presented novel controllers for FJRs to achieve tracking control of link angles with any prescribed performance requirements. By combining neural learning control scheme, further results are given in [33–35]. In [36], an adaptive prescribed performance tracking control scheme is investigated for a class of output feedback nonlinear systems with input unmodeled dynamics based on dynamic surface control method.

In addition, adaptive neural control of nonlinear system has been widely studied for decades, but most of the traditional works focus on the system stability through online adjustment of neural weights and less works discuss the knowledge acquisition, storage, and utilization of optimal neural weights. To achieve such learning ability, the key problem is to verify the persistent excitation (PE) condition. Thanks to the results in [37], a deterministic learning mechanism is proposed, which proved the satisfaction of PE condition for the localized radial basis function (RBF) NN centered in a neighborhood along recurrent orbits. The result was extended to nonlinear systems satisfying matching conditions [38–40]. By combining recursive design technologies such as backstepping control and the system decomposition strategy, the deterministic learning was also applied to solve learning problem of accurate identification of ocean surface ship and robot manipulation in uncertain dynamical environments [41–44]. However, due to the recursive property of backstepping control, the convergence of neural weights has to be recursively verified based on the system decomposition strategy. It would be a tedious and complex process since the intermediate variables grow drastically as the order of system increases. Therefore, it is difficult to prove all neural weights convergence for high-order system by existing works.

This paper focuses on learning from adaptive neural control of flexible joint manipulator with unknown dynamics under the prescribed constraints. A performance function is introduced to transform the constrained tracking error into the unconstrained variable. To avoid the curse of dimensionality of RBF NN, first-order filters are introduced to reduce the number of NN approximators and decrease the dimension of NN inputs. The control law is constructed based on Lyapunov stability, which guarantees the closed-loop stability and the tracking error satisfying the prescribed performance during the transient process. Subsequently, due to the property of DSC and structure features of the considered manipulator, a system decomposition strategy is employed to decompose the stable closed-loop system into two linear time-varying (LTV) perturbed subsystems on the basis of the number of NNs in the whole system. Through the recursive

design, the recurrent properties of NN input variables are easily proven. Consequently, with the satisfaction of the PE condition of RBF NNs, the convergence of partial neural weights is verified, and the unknown dynamics of system are approximated accurately in a local region along recurrent orbits. By utilization of the constant neural weights stored, a neural learning controller is developed to achieve the closed-loop stability and better control performance under the prescribed constraints for the same or similar control task. Compared with the existing neural learning results, the proposed neural learning control scheme not only achieves better control performance with specified transient and steady-state constraints but also reduces the dimension of NN inputs and the number of NNs significantly.

This paper is organized as follows. In Section 2, the problem formulation and preliminaries are stated before the control scheme design. In Section 3, a novel adaptive neural dynamic surface control scheme is proposed to guarantee that the constrained tracking error converges to a small neighborhood around zero with the prescribed performance in a finite time, and all the signals in the closed-loop system are uniformly ultimately bounded. Section 4 shows that the knowledge acquisition, expression, storage, and utilization of the manipulator's unknown dynamics can be achieved after the steady-state control process. To verify the effectiveness of the proposed control scheme, simulation results on a single-link flexible joint manipulator and experiment results on Baxter robot are given in Section 5. Last but not least, the conclusions are drawn in Section 6.

## 2. Problem Formulation and Preliminaries

*2.1. System Formulation.* In this paper, we consider an  $n$ -link manipulator with flexible joints, whose model is described by [1]

$$\begin{aligned} M(q_1) \ddot{q}_1 + C(q_1, \dot{q}_1) \dot{q}_1 + g(q_1) + K(q_1 - q_2) &= 0, \\ J \ddot{q}_2 - K(q_1 - q_2) &= u, \end{aligned} \quad (1)$$

where  $q_1 \in R^n$  is the vector of links' angle positions and  $q_2 \in R^n$  is the vector of motors' angle positions.  $M(q_1) \in R^{n \times n}$  is the link inertia matrix and  $J \in R^{n \times n}$  is the diagonal and positive definite motor inertia matrix. Moreover,  $C(q_1, \dot{q}_1) \in R^{n \times n}$  denotes the Coriolis and centrifugal matrix and  $g(q_1) \in R^n$  represents the gravitational terms.  $K \in R^{n \times n}$  is a diagonal and positive definite matrix of joint spring constants; thus  $K^{-1}$  is also positive definite. Finally,  $u \in R^n$  is the control input of system (1), and the output of system (1) is  $q_1$ .

*Property 1* (see [2]). The inertia matrix  $M(q_1)$  is symmetric and positive definite; both  $M(q_1)$  and  $M^{-1}(q_1)$  are uniformly bounded.

*Property 2* (see [2]). The Coriolis and centrifugal matrix  $C(q_1, \dot{q}_1)$  can be defined such that  $\dot{M}(q_1) - 2C(q_1, \dot{q}_1)$  is skew symmetric; that is,  $x^T(\dot{M}(q_1) - 2C(q_1, \dot{q}_1))x = 0, \forall x \in R^n$ .

The reference trajectory vector  $y_d \in R^n$  is generated by the following smooth and bounded reference model:

$$\begin{aligned} \dot{x}_{di} &= x_{d,i+1}, \quad i = 1, \dots, m-1, \\ \dot{x}_{dm} &= f_d(x_d), \\ y_d &= x_{d1}, \end{aligned} \quad (2)$$

where  $x_d = [x_{d1}^T, \dots, x_{dm}^T]^T \in R^{m \cdot n}$  and  $y_d$  is the system outputs vector.  $f_d(\cdot)$  is a smooth known nonlinear function. Assume  $x_{di} \in R^n$ ,  $1 \leq i \leq m$  are recurrent signals and the reference orbit (denoted as  $\psi_d(x_d(0))$ ) is a recurrent motion. Moreover,  $M^{-1}(q_1)$ ,  $C(q_1, \dot{q}_1)$ ,  $g(q_1)$ ,  $K$ , and  $J$  are assumed as unknown terms.

Our goal is to design a neural learning controller, which forces the tracking error vector (i.e.,  $e_1 = q_1 - y_d$ ) converges to a small neighborhood around zero with prescribed performance in a finite time. Before the design of learning control (LC) scheme, a stable adaptive neural dynamic surface controller with prescribed performance is developed to verify the feasibility of ANC scheme. According to the deterministic learning theory, the unknown system dynamics are accurately approximated by localized RBF networks along the recurrent orbits of NN inputs. Then, based on the ANC scheme and the approximation of localized RBF networks, the knowledge on unknown system dynamics is stored in static neural weights, which is also reused to develop a neural learning controller. This neural learning controller is verified to achieve the closed-loop system stability and better control performance with prescribed constraints for the same or similar tasks.

**2.2. Prescribed Tracking Performance.** In this paper, the output error vector of system (1) is defined as  $e_1 = q_1 - y_d = [e_{11}, e_{12}, \dots, e_{1n}]^T \in R^n$ . To achieve the prescribed performance (i.e., overshoot, convergence rate, and convergence accuracy), each element in  $e_1$  is constrained into the following prescribed region:

$$-\delta_{1k}\beta(t) < e_{1k} < \delta_{2k}\beta(t), \quad k = 1, 2, \dots, n, \quad (3)$$

where  $\delta_{1k}$  and  $\delta_{2k}$  are positive design constants.  $\beta(t)$  is a bounded, smooth, strictly positive, and decreasing performance function. In addition,  $\beta(t)$  is chosen as the following form by setting  $\lim_{t \rightarrow 0} \beta(t) = \beta_0$  and  $\lim_{t \rightarrow \infty} \beta(t) = \beta_\infty$ :

$$\beta(t) = (\beta_0 - \beta_\infty)e^{-\kappa t} + \beta_\infty, \quad (4)$$

where  $\beta_0$ ,  $\beta_\infty$ , and  $\kappa$  are positive constants. With (3) and (4), it can be concluded that the convergence rate of  $e_{1k}(t)$  is constrained by the decreasing rate  $\kappa$  of  $\beta(t)$ , while its maximum bound of overshoot at initial moment is constrained by  $-\delta_{1k}\beta_0$  and  $\delta_{2k}\beta_0$ , and its steady error is constrained within a range from  $-\delta_{1k}\beta_\infty$  to  $\delta_{2k}\beta_\infty$ .

**2.3. RBF Neural Network.** According to [45], RBF NN can approximate any continuous function vector  $F(Z) \in R^n$  over a compact set  $\Omega_Z \in R^m$  to any arbitrary accuracy as

$$F(Z) = W^{*T}S(Z) + \epsilon(Z), \quad \forall Z \in \Omega_Z \in R^m, \quad (5)$$

where  $W^* \in R^{l \times n}$  is the ideal weights matrix,  $l$  is the NN node number,  $S(Z) = [s_1(Z), s_2(Z), \dots, s_l(Z)]^T \in R^l$  is the basis function vector with  $s_i(Z)$  ( $i = 1, 2, \dots, l$ ) chosen as the Gaussian function  $s_i(Z) = \exp[-(Z - \mu_i)^T(Z - \mu_i)/2\eta_i^2]$ , and  $\epsilon(Z) = [\epsilon_1(Z), \epsilon_2(Z), \dots, \epsilon_n(Z)]^T \in R^n$  is the approximation error vector which satisfies  $\|\epsilon_i(Z)\| \leq \epsilon_i^*$ ,  $i = 1, 2, \dots, n$ , with constant  $\epsilon_i^* > 0$ .

On the other hand, it has been shown in [46] that, for any bounded trajectory  $Z(t)$  over the compact set  $\Omega_Z$ , the continuous and smooth function  $F(Z)$  can be approximated to an arbitrary accuracy using the localized RBF NNs with a limited number of neurons located in a local region along the trajectory:

$$F(Z) = W_\zeta^{*T}S_\zeta(Z) + \epsilon_\zeta(Z), \quad (6)$$

where  $S_\zeta(Z) \in R^l$  is the subvector of  $S(Z)$  and  $W_\zeta^{*T} \in R_\zeta^l$  with  $l_\zeta < l$ .  $\epsilon_\zeta(Z)$  is the approximation error which is close to  $\epsilon(Z)$ .

**Lemma 1** ((partial PE condition for RBF NNs) [46]). *Consider any continuous recurrent trajectory  $Z(t)$ . Assume that  $Z(t)$  is a continuous map from  $[0, \infty)$  into  $R^q$ , and  $Z(t)$  remains in a bounded compact set  $\Omega_Z$  with  $\Omega_Z \subset R^q$ . Then, for the RBF NN  $W^T S(Z)$  with centers placed on a regular lattice (large enough to cover the compact set  $\Omega_Z$ ), the regression subvector  $S_\zeta(Z)$  consisting of RBFs with centers located in a small neighborhood of  $Z(t)$  is persistently exciting.*

### 3. Adaptive Neural DSC Design with Predefined Tracking Performance

In this section, performance function is introduced for describing constraints of system (1). Then an adaptive neural DSC is developed, with the design of adaptive control law based on the transformed error. Meanwhile, RBF NN is used to approximate the unknown dynamics.

*Step 1.* Similarly to the traditional backstepping design, we set

$$e_1 = q_1 - y_d = [e_{11}, e_{12}, \dots, e_{1n}]^T \in R^n. \quad (7)$$

It should be pointed out that any errors set in previously traditional design are under unrestricted condition [20], while  $e_1$  is constrained to satisfy condition (3) in this paper, which is rewritten as

$$-\delta_{1k} < \frac{e_{1k}}{\beta(t)} < \delta_{2k}, \quad k = 1, 2, \dots, n. \quad (8)$$

It implies that  $e_1$  can not be used for design directly due to the limitation of the traditional design method. To solve the constrained tracking control problem, the constrained error should be transformed into the unconstrained one equivalently. Therefore, a new error vector is defined as  $z_1 = [z_{11}, z_{12}, \dots, z_{1n}]^T \in R^n$  called transformed error vector. Define a smooth transformed functions vector  $\Phi(z_1) = [\phi(z_{11}), \phi(z_{12}), \dots, \phi(z_{1n})]^T$  with  $\phi(z_{1k})$  being chosen as

$$\phi(z_{1k}) = \delta_{ak} \arctan\left(z_{1k} - \tan\left(\frac{\delta_{bk}}{\delta_{ak}}\right)\right) + \delta_{bk}, \quad (9)$$

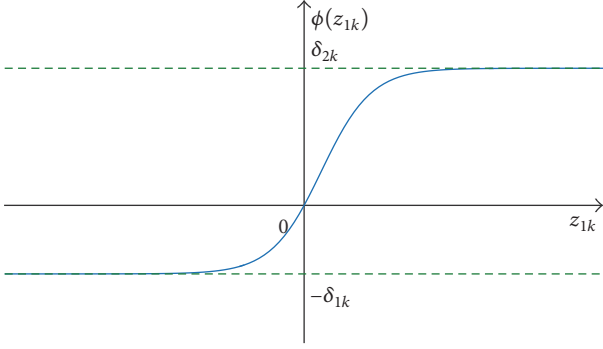


FIGURE 1: Transformed function  $\phi(z_{1k})$ .

where  $\delta_{ak} = (\delta_{2k} + \delta_{1k})/\pi$ , and  $\delta_{bk} = (\delta_{2k} - \delta_{1k})/2$ . Moreover, for the symmetric tracking error constraints  $-\delta_k \beta(t) < e_{1k} < \delta_k \beta(t)$ ,  $k = 1, 2, \dots, n$ , the transformation function (9) can be constructed as  $\phi(z_{1k}) = (2/\pi)\delta_k \arctan(z_{1k})$ . For clarity, Figure 1 illustrates the relationship between  $\phi(z_{1k})$  and  $z_{1k}$ .

It can be concluded from (9) and Figure 1 that the transformation function  $\phi(z_{1k})$  is smooth and strictly increasing while possessing the following properties:

$$\begin{aligned} -\delta_{1k} < \phi(z_{1k}) < \delta_{2k}, \quad \forall z_{1k} \in L_\infty, \\ \lim_{z_{1k} \rightarrow +\infty} \phi(z_{1k}) &= \delta_{2k}, \\ \lim_{z_{1k} \rightarrow -\infty} \phi(z_{1k}) &= -\delta_{1k}. \end{aligned} \quad (10)$$

By combining (8) with (10),  $e_{1k}$  can be rewritten as

$$e_{1k} = \phi(z_{1k}) \beta(t), \quad k = 1, 2, \dots, n. \quad (11)$$

Since  $\phi(\cdot)$  is a strictly monotonic increasing function and  $\beta(t) \neq 0$ , the inverse function  $z_{1k}$  of  $\phi(\cdot)$  exists, which can be expressed as

$$z_{1k} = \tan\left(\frac{e_{1k}}{\delta_{ak}\beta(t)} - \frac{\delta_{bk}}{\delta_{ak}}\right) + \tan\left(\frac{\delta_{bk}}{\delta_{ak}}\right). \quad (12)$$

Noting that  $z_1 = [z_{11}, z_{12}, \dots, z_{1n}]^T$ , its derivative can be presented as

$$\dot{z}_1 = \Upsilon \left( \dot{q}_1 - \dot{y}_d - \frac{\dot{\beta}(t)}{\beta(t)} e_1 \right), \quad (13)$$

where  $\Upsilon = \text{diag}(\gamma_1, \gamma_2, \dots, \gamma_n)$ , with  $\gamma_k$  ( $k = 1, 2, \dots, n$ ) being presented as

$$\gamma_k = \frac{1}{\delta_{ak}\beta(t)} \left[ 1 + \tan^2\left(\frac{e_{1k}}{\delta_{ak}\beta(t)} - \frac{\delta_{bk}}{\delta_{ak}}\right) \right]. \quad (14)$$

It is clear that  $\Upsilon$  is positive definite, which is helpful for the stability analysis. By introducing a new filter variable  $\alpha_{1f} \in R^n$  and noting  $z_2 = \dot{q}_1 - \alpha_{1f}$ , then the virtual controller  $\alpha_1 \in R^n$  is constructed as

$$\alpha_1 = -(\Upsilon^{-1}C_1 + \Upsilon)z_1 + \dot{y}_d + \frac{\dot{\beta}(t)}{\beta(t)}e_1, \quad (15)$$

where  $C_1 \in R^{n \times n}$  is a diagonal and positive design matrix. Take  $\alpha_1$  as the input of a first-order filter and  $\alpha_{1f}$  as the output of it, a differential equation is constructed as

$$\tau_1 \dot{\alpha}_{1f} + \alpha_{1f} = \alpha_1, \quad \alpha_{1f}(0) = \alpha_1(0), \quad (16)$$

where  $\tau_1$  is the filter time constant and set  $y_1 = \alpha_{1f} - \alpha_1$ ; then (13) can be rewritten as

$$\dot{z}_1 = -(C_1 + \Upsilon^T \Upsilon)z_1 + \Upsilon z_2 + \Upsilon y_1. \quad (17)$$

Step 2. Let  $z_2 = \dot{q}_1 - \alpha_{1f}$ , its derivative can be obtained:

$$\begin{aligned} \dot{z}_2 &= M^{-1}(q_1)K(q_2 - q_1 - K^{-1}C(q_1, \dot{q}_1)\alpha_{1f} \\ &\quad - K^{-1}g(q_1) - K^{-1}M(q_1)\dot{\alpha}_{1f} - K^{-1}C(q_1, \dot{q}_1)z_2). \end{aligned} \quad (18)$$

Define the unknown dynamics in (18) as

$$\begin{aligned} H_2(\Psi_2) &= -K^{-1}C(q_1, \dot{q}_1)\alpha_{1f} - K^{-1}g(q_1) \\ &\quad - K^{-1}M(q_1)\dot{\alpha}_{1f}, \end{aligned} \quad (19)$$

where

$$\Psi_2 = [q_1^T, \dot{q}_1^T, \dot{\alpha}_{1f}^T]^T \in R^{3n}. \quad (20)$$

According to the property of RBF NN,  $H_2(\Psi_2)$  can be approximated accurately by RBF NN and (19) is rewritten as

$$H_2(\Psi_2) = W_2^{*T}S_2(\Psi_2) + \epsilon_2, \quad (21)$$

where  $\epsilon_2 \in R^n$  is the bounded approximation error vector which satisfies  $\|\epsilon_2\| \leq \epsilon_2^*$ . Define  $\widehat{W}_2$  as the estimate of  $W_2^*$  and set  $\widetilde{W}_2 = \widehat{W}_2 - W_2^*$ . Then a new filter variable  $\alpha_{2f} \in R^n$  is introduced and note  $z_3 = z_2 - \alpha_{2f}$ ; (18) can be rewritten by combining (19) and (21):

$$\begin{aligned} \dot{z}_2 &= M^{-1}(q_1)K(z_3 + \alpha_{2f} - q_1 + W_2^{*T}S_2(\Psi_2) + \epsilon_2 \\ &\quad - K^{-1}C(q_1, \dot{q}_1)z_2). \end{aligned} \quad (22)$$

Then the virtual controller  $\alpha_2 \in R^n$  is constructed as

$$\alpha_2 = -\Upsilon z_1 - C_2 z_2 - \widehat{W}_2^T S_2(\Psi_2) + q_1, \quad (23)$$

where  $C_2 \in R^{n \times n}$  is a diagonal and positive design matrix. The updated law of NN weights is given by

$$\dot{\widehat{W}}_2 = \Gamma_2 [S_2(\Psi_2)z_2^T - \sigma_2 \widehat{W}_2] \quad (24)$$

with diagonal matrix  $\Gamma_2 = \Gamma_2^T > 0$  and small value  $\sigma_2 > 0$  for enhancing the robustness of the controller (23).

Take  $\alpha_2$  as the input of a first-order filter and  $\alpha_{2f}$  as the output of it, a differential equation is constructed as

$$\tau_2 \dot{\alpha}_{2f} + \alpha_{2f} = \alpha_2, \quad \alpha_{2f}(0) = \alpha_2(0), \quad (25)$$

where  $\tau_2$  is the filter time constant and set  $y_2 = \alpha_{2f} - \alpha_2$ ; then (22) can be rewritten as

$$\begin{aligned} \dot{z}_2 &= M^{-1}(q_1)K(-\Upsilon z_1 - C_2 z_2 + z_3 + y_2 \\ &\quad - \widehat{W}_2^T S_2(\Psi_2) + \epsilon_2 - K^{-1}C(q_1, \dot{q}_1)z_2). \end{aligned} \quad (26)$$



Step 3. Define  $z_3 = q_2 - \alpha_{2f}$ , its derivative can be obtained:

$$\dot{z}_3 = \dot{q}_2 - \dot{\alpha}_{2f}. \quad (27)$$

Introduce a new filter variable  $\alpha_{3f} \in R^n$  and note  $z_4 = \dot{q}_2 - \alpha_{3f}$ . Then the virtual controller  $\alpha_3 \in R^n$  is constructed as

$$\alpha_3 = -z_2 - C_3 z_3 + \dot{\alpha}_{2f}, \quad (28)$$

where  $C_3 \in R^{n \times n}$  is a diagonal and positive design matrix. Take  $\alpha_3$  as the input of a first-order filter and  $\alpha_{3f}$  as the output of it, a differential equation is constructed as

$$\tau_3 \dot{\alpha}_{3f} + \alpha_{3f} = \alpha_3, \quad \alpha_{3f}(0) = \alpha_3(0), \quad (29)$$

where  $\tau_3$  is the filter time constant and set  $y_3 = \alpha_{3f} - \alpha_3$ ; then (27) can be rewritten as

$$\dot{z}_3 = -z_2 - C_3 z_3 + z_4 + y_3. \quad (30)$$

Step 4. Let  $z_4 = \dot{q}_2 - \alpha_{3f}$ ; the following can be obtained:

$$\dot{z}_4 = J^{-1} [u + K(q_1 - q_2) - J\dot{\alpha}_{3f}]. \quad (31)$$

Define the unknown dynamics of the system as

$$H_4(\Psi_4) = K(q_1 - q_2) - J\dot{\alpha}_{3f}, \quad (32)$$

where

$$\Psi_4 = [q_1^T, q_2^T, \dot{\alpha}_{3f}^T]^T \in R^{3n}. \quad (33)$$

According to the property of RBF NN,  $H_4(\Psi_4)$  can be approximated accurately by RBF NN and (32) is rewritten as

$$H_4(\Psi_4) = W_4^{*T} S_4(\Psi_4) + \epsilon_4, \quad (34)$$

where  $W_4^* \in R^{j \times n}$  is the ideal constant weight matrix with  $j$  being the NN node number,  $S_4(\Psi_4) \in R^j$  is the basis function vector, and  $\epsilon_4$  is the bounded approximation error vector which satisfies  $\|\epsilon_4\| \leq \epsilon_4^*$ . Define  $\widehat{W}_4$  as the estimate of  $W_4^*$  and let  $\widetilde{W}_4 = \widehat{W}_4 - W_4^*$ . Then (31) can be rewritten by combining (32) and (34):

$$\dot{z}_4 = J^{-1} [u + W_4^{*T} S_4(\Psi_4) + \epsilon_4]. \quad (35)$$

Then the control input  $u$  is constructed as

$$u = -z_3 - C_4 z_4 - \widehat{W}_4^T S_4(\Psi_4), \quad (36)$$

where  $C_4 \in R^{n \times n}$  is a diagonal and positive design matrix. The updated law of NN weights is given by

$$\dot{\widehat{W}}_4 = \Gamma_4 [S_4(\Psi_4) z_4^T - \sigma_4 \widehat{W}_4] \quad (37)$$

with diagonal matrix  $\Gamma_4 = \Gamma_4^T > 0$  and small value  $\sigma_4 > 0$  for enhancing the robustness of the controller (36). Then (35) can be rewritten as

$$\dot{z}_4 = J^{-1} [-z_3 - C_4 z_4 - \widetilde{W}_4^T S_4(\Psi_4) + \epsilon_4]. \quad (38)$$

Let us construct the following Lyapunov function candidate:

$$V = \frac{1}{2} \sum_{i=1}^4 z_i^T K_i z_i + \frac{1}{2} \sum_{k=1}^2 \text{tr} [\widehat{W}_{2k}^T \Gamma_{2k}^{-1} \widehat{W}_{2k}] + \frac{1}{2} \sum_{l=1}^3 y_l^T y_l, \quad (39)$$

where  $K_1 = I, K_2 = K^{-1} M(q_1), K_3 = I$ , and  $K_4 = J$ .

*Remark 2.* It should be pointed out that, in the adaptive neural backstepping design [20], the derivative  $\dot{\alpha}_2$  of the virtual control  $\alpha_2$  in Step 2 is usually used to design the virtual control  $\alpha_3$  in Step 3. However, according to (1) and (23), it can be seen clearly that  $\dot{\alpha}_2$  is not available because the unknown terms, such as  $M^{-1}(q_1), C(q_1, \dot{q}_1)$ , and  $g(q_1)$ , are included in  $\dot{\alpha}_2$ . Therefore, a neural network has to be employed in Step 3 of backstepping to approximate the unknown dynamics in  $\dot{\alpha}_2$ . However, too many neural networks employed make the control scheme implemented difficultly. To solve this problem, this paper introduces a new variable  $\alpha_{2f}$  in (25) to design virtual control  $\alpha_2$  using a first-order filter. From (25), it is easy to calculate that  $\dot{\alpha}_{2f} = (\alpha_2 - \alpha_{2f})/\tau_2 = -y_2/\tau_2$ . The advantage of the proposed approach is that the unknown dynamics in the previous step does not affect the design of virtual control in next step, so that the number of NNs employed can be greatly reduced. Moreover, the proposed method uses  $\dot{\alpha}_{if}$ , instead of the intermediate variables used in [20], as neural input variable, so that there are only  $3n$  neural input variables for neural networks  $W_2^{*T} S_2(\Psi_2)$  and  $W_4^{*T} S_4(\Psi_4)$ , respectively, which reduces significantly the number of neural input variables used in [20], where  $4n$  and  $8n$  neural input variables are used in  $W_2^{*T} S_2(\Psi_2)$  and  $W_4^{*T} S_4(\Psi_4)$ .

**Theorem 3.** Consider the manipulator model (1), the reference trajectory  $y_d$ , the prescribed performance bounds (8), the state transformation (11), the adaptive NN control law (36), the weight updated law (24), (37), and Lyapunov function (39). Given any constant  $\mu > 0$ , for any bounded initial conditions satisfying the prescribed performance (8) and  $V(0) < \mu$ , there exists design parameters  $C_1, C_2, C_3, C_4, \Gamma_2, \sigma_2, \Gamma_4, \sigma_4, \tau_1, \tau_2$ , and  $\tau_3$ , such that the proposed control scheme guarantees that (1) all the signals in the closed-loop system are uniformly ultimately bounded and (2) the tracking error converges to a small neighborhood around zero with the prescribed performance (8) in a finite time  $T_1$ .

*Proof.* See Appendix A.  $\square$

## 4. Dynamic Neural Learning

In this section, we will show the learning ability of RBF NNs for unknown system dynamics  $H_2(\Psi_2)$  and  $H_4(\Psi_4)$  in the case of the manipulator with predefined tracking performance. Subsequently, the stored knowledge on the unknown dynamics will be utilized to design a neural learning controller achieving better control performance, while satisfying the prescribed performance.

*4.1. Learning from Adaptive Neural DSC.* According to Lemma 1, it can be concluded that a recurrent orbit can make

regression subvectors  $S_\zeta(Z)$  satisfy the partial PE condition, which is a key condition to ensure the accurate convergence of neural weights.

**Theorem 4.** Consider the closed-loop system consisting of the flexible joint manipulator model (1), the reference trajectory  $y_d$ , the prescribed performance bounds (8), the state transformation (11), the adaptive NN control law (36), and the weight updated law (24) and (37). Then, for any recurrent orbit  $\psi_d(x_d(t))|_{t \geq 0}$  and initial condition  $x(0) \in \Omega_0$  ( $\Omega_0$  is an appropriately chosen compact set) satisfying the prescribed performance (8) and  $\widehat{W}_2(0) = \widehat{W}_4(0) = 0$ , we have that the neural weights  $\widehat{W}_i$  converge to small neighborhoods around optimal values  $W_i^*$ , and the locally accurate approximation of the system dynamics  $H_i(\Psi_i)$  is obtained by the stored knowledge  $\overline{W}_i$ :

$$\overline{W}_i = \text{mean}_{t \in [t_{a_i}, t_{b_i}]} \widehat{W}_i(t) \quad (40)$$

with  $i = 2, 4$ ,  $T_1 \leq t_{a_2} \leq t_{b_2} \leq T_2 \leq t_{a_4} \leq t_{b_4}$ .  $[t_{a_i}, t_{b_i}]$  and  $[t_{a_1}, t_{b_1}]$  are time segments after the steady-state control process.

*Proof.* From Theorem 3, all the signals in the closed-loop system are uniformly ultimately bounded and the tracking error vector  $e_1 = q_1 - y_d$  converges to a small neighborhood around zero with the prescribed performance in a finite time  $T_1$ . Thus, the state  $q_1$  converges closely to the recurrent signals  $y_d$  for all  $t \geq T_1$ . In addition, it can be obtained from the proof of Theorem 3 that the transformed error vector  $z_1$  converges exponentially to a small neighborhood around zero in a finite time  $T_1$ . From (15), it can be concluded that virtual control  $\alpha_1$  is recurrent with the same period as  $\dot{y}_d$  by combining the convergence of  $z_1$  and  $e_1$ . Noting  $\dot{q}_1 = z_2 + y_1 + \alpha_1$  with  $z_2$  and  $y_1$  being close to a small neighborhood around zero based on Theorem 3, then  $\dot{q}_1$  is a recurrent signal with the same period as  $\alpha_1$ . In addition,  $\alpha_{1f}$  is also a recurrent signal with the same period as  $\alpha_1$  since  $y_1 = \alpha_{1f} - \alpha_1$  and  $y_1$  is a small value. From (16), it can be obtained that  $\dot{\alpha}_{1f}$  is a recurrent signal as well. Therefore, the NN inputs  $\Psi_2 = [q_1^T, \dot{q}_1^T, \dot{\alpha}_{1f}^T]^T$  are recurrent for all  $t \geq T_1$ , and a partial PE condition of  $S_2(\Psi_2)$  is satisfied according to Lemma 1. By combining the convergence of  $z_1$  and the localized RBF NN along the recurrent signals  $\Psi_2(t)$  ( $t > T_1$ ), it can be obtained from (24) and (26) that

$$\dot{z}_2 = M^{-1}(q_1)K \left[ -Yz_1 - C_2z_2 + z_3 + y_2 - \overline{W}_{2\zeta}^T S_{2\zeta}(\Psi_2) + \epsilon_{2\zeta} - K^{-1}C(q_1, \dot{q}_1)z_2 \right] \quad (41)$$

$$\begin{aligned} \dot{\overline{W}}_{2\zeta} &= \Gamma_{2\zeta} \left[ S_{2\zeta}(\Psi_2)z_2^T - \sigma_2 \overline{W}_{2\zeta} \right], \\ \dot{\overline{W}}_{\overline{2\zeta}} &= \Gamma_{\overline{2\zeta}} \left[ S_{\overline{2\zeta}}(\Psi_2)z_2^T - \sigma_2 \overline{W}_{\overline{2\zeta}} \right], \end{aligned} \quad (42)$$

where the subscript  $\zeta$  stands for the region near the orbits  $\Psi_2(t)$ ,  $S_{2\zeta}(\Psi_2)$  is the subvector of  $S_2(\Psi_2)$  consisting of the corresponding RBFs, and  $\overline{W}_{2\zeta}$  is the corresponding weight submatrix of  $\overline{W}_2$ . Moreover, the subscript  $\overline{\zeta}$  represents the region

away from the orbits  $\Psi_2(t)$ , and  $\epsilon_{2\zeta} = \epsilon_2(\Psi_2) - \overline{W}_{2\zeta}^T S_{2\zeta}(\Psi_2)$  is the NN approximation error along the orbits  $\Psi_2(t)$ . Since  $\|\overline{W}_{2\zeta}^T S_{2\zeta}(\Psi_2)\|$  is small,  $\|\epsilon_{2\zeta}\|$  is close to  $\|\epsilon_2(\Psi_2)\|$ .  $\square$

It would be shown that the perturbation term  $M^{-1}(q_1)K\epsilon_{2\zeta}$  may be large, which will make the accurate convergence of neural weights become difficult. To solve this problem, a state transformation  $z_{2s} = K^{-1}M(q_1)z_2$  is introduced to eliminate the influence of  $M^{-1}(q_1)K\epsilon_{2\zeta}$ . Subsequently, set  $\overline{W}_{2\zeta} = [\overline{W}_{21\zeta}, \overline{W}_{22\zeta}, \dots, \overline{W}_{2n\zeta}] \in R^{\zeta \times n}$  and  $\overline{W}_{2\overline{\zeta}} = [\overline{W}_{21\overline{\zeta}}, \overline{W}_{22\overline{\zeta}}, \dots, \overline{W}_{2n\overline{\zeta}}] \in R^{\zeta \times n}$ ; (41) can be transformed into the following form:

$$\begin{aligned} \begin{bmatrix} \dot{z}_{2s} \\ \dot{\overline{W}}_{21\zeta} \\ \vdots \\ \dot{\overline{W}}_{2n\zeta} \end{bmatrix} &= \begin{bmatrix} A_2(t) & B_2(t) \\ C_2(t) & 0 \end{bmatrix} \begin{bmatrix} z_{2s} \\ \overline{W}_{21\zeta} \\ \vdots \\ \overline{W}_{2n\zeta} \end{bmatrix} \\ &+ \begin{bmatrix} \epsilon'_{2\zeta} \\ -\sigma_2 \Gamma_{2\zeta} \overline{W}_{21\zeta} \\ \vdots \\ -\sigma_2 \Gamma_{2\zeta} \overline{W}_{2n\zeta} \end{bmatrix}, \end{aligned} \quad (43)$$

where

$$\begin{aligned} A_2(t) &= -FM^{-1}(q_1)K \in R^{n \times n} \\ B_2(t) &= -\text{diag} \{S_{2\zeta}^T, \dots, S_{2\zeta}^T\} \in R^{n \times \zeta n} \\ C_2(t) &= \overline{\Gamma}_{2\zeta} \overline{S}_{2\zeta} M^{-1}(q_1)K \in R^{\zeta \times n \times n} \\ F &= C_2 - K^{-1} [\dot{M}(q_1) - C(q_1, \dot{q}_1)] \in R^{n \times n} \\ \overline{\Gamma}_{2\zeta} &= \text{diag} \{\Gamma_{2\zeta}, \dots, \Gamma_{2\zeta}\} \in R^{\zeta n \times \zeta n} \\ \overline{S}_{2\zeta} &= \text{diag} \{S_{2\zeta}, \dots, S_{2\zeta}\} \in R^{\zeta n \times n} \\ \epsilon'_{2\zeta} &= Yz_1 + z_3 + y_2 + \epsilon_{2\zeta}. \end{aligned} \quad (44)$$

According to Theorem 3,  $\epsilon'_{2\zeta}$  is a small value and  $-\sigma_2 \Gamma_{2\zeta} \overline{W}_{2i\zeta}$  ( $i = 1, 2, \dots, n$ ) can be made as a small value by choosing a small design parameter  $\sigma_2$ . Therefore, system (43) can be considered as a linear time-varying (LTV) system with a small perturbation term. Choose  $P(t) = M^{-1}(q_1)K$ ; then

$$\begin{aligned} \dot{P}(t) + P(t)A_2(t) + A_2^T(t)P(t) \\ = \dot{M}^{-1}(q_1)K - (F_1 C_2 + F_2)M^{-1}(q_1)K, \end{aligned} \quad (45)$$

where  $F_1 = M^{-1}(q_1)K + KM^{-1}(q_1)$ ,  $F_2 = KM^{-1}(q_1)[\dot{M}(q_1) - C(q_1, \dot{q}_1)]K^{-1} - M^{-1}(q_1)[\dot{M}(q_1) - C(q_1, \dot{q}_1)]$ . Choose  $C_2$  appropriately such that  $\dot{P}(t) + P(t)A_2(t) + A_2^T(t)P(t) < 0$ . Subsequently, based on the perturbation theory from Lemma 4.6 in [47], both  $z_{2s}$  and  $\overline{W}_{2\zeta}$  converge exponentially to a small

neighborhood around zero in a finite time  $T_1$ , and the size of neighborhood is determined by  $\|\dot{\epsilon}_{2\zeta}\|$  and  $\|\sigma_2 \Gamma_{2\zeta} \widehat{W}_{2\zeta}\|$ , respectively. Noting  $\widehat{W}_{2\zeta} = \widehat{W}_{2\zeta} - W_{2\zeta}^*$ , it is clear that  $\widehat{W}_{2\zeta}$  can converge to a small neighborhood of optimal weights  $W_{2\zeta}^*$  in a finite time  $T_1$ , and the constant weights  $\overline{W}_2$  can be obtained from (40). According to the localization property of RBF NN, the system dynamics  $H_2(\Psi_2)$  can be described by

$$\begin{aligned} H_2(\Psi_2) &= \widehat{W}_{2\zeta}^T S_{2\zeta}(\Psi_2) + \epsilon_{2\zeta}(\Psi_2) \\ &= \overline{W}_2^T S_2(\Psi_2) + \bar{\epsilon}_2(\Psi_2), \end{aligned} \quad (46)$$

where both  $\epsilon_{2\zeta}(\Psi_2)$  and  $\bar{\epsilon}_2(\Psi_2)$  are close to  $\epsilon_2(\Psi_2)$  due to the convergence of  $\widehat{W}_{2\zeta}$ .

According to the above analysis and noting  $\alpha_2 = -\Upsilon z_1 - C_2 z_2 - \overline{W}_2^T S_2(\Psi_2) + q_1$ , there exists a constant  $T_2 > T_1$ , such that the virtual control  $\alpha_2$  can be rewritten as

$$\alpha_2 = -\Upsilon z_1 - C_2 z_2 - \overline{W}_2^T S_2(\Psi_2) + q_1 + \varepsilon_2 \quad (47)$$

for all  $t > T_2$ , where  $\varepsilon_2 = [\overline{W}_2 - \widehat{W}_2]^T S_2(\Psi_2)$  is a small value because of the convergence of  $\widehat{W}_2$ . According to Theorem 3,  $\alpha_2$  is a recurrent signal. Since  $y_2 = \alpha_{2f} - \alpha_2$  and  $y_2$  is a small value,  $\alpha_{2f}$  is also recurrent with the same period as  $\alpha_2$ . From (25), it can be obtained that  $\dot{\alpha}_{2f}$  is also a recurrent signal as well. From (28), by combining the convergence of  $z_2$  and  $z_3$ , virtual control  $\alpha_3$  is a recurrent signal with the same period as  $\dot{\alpha}_{2f}$ . Since  $y_3 = \alpha_{3f} - \alpha_3$  and  $y_3$  is a small value,  $\alpha_{3f}$  is also recurrent with the same period as  $\alpha_3$ . From (29), it can be obtained that  $\dot{\alpha}_{3f}$  is also a recurrent signal as well. Therefore, the NN inputs  $\Psi_4 = [q_1^T, q_2^T, \dot{\alpha}_{3f}^T]^T$  are recurrent for all  $t \geq T_2$ , and a partial PE condition of  $S_4(\Psi_4)$  is satisfied according to Lemma 1. By combining the convergence of  $z_4$  and the localized RBF NN along the recurrent signals  $\Psi_4(t)$  ( $t > T_2$ ), it can be obtained from (37) and (38) that

$$\begin{aligned} \dot{z}_4 &= J^{-1} [-z_3 - C_4 z_4 - \widehat{W}_{4\zeta}^T S_{4\zeta}(\Psi_4) + \epsilon_{4\zeta}] \\ \dot{\widehat{W}}_{4\zeta} &= \Gamma_{4\zeta} [S_{4\zeta}(\Psi_4) z_4^T - \sigma_4 \widehat{W}_{4\zeta}], \\ \dot{\widehat{W}}_{4\bar{\zeta}} &= \Gamma_{4\bar{\zeta}} [S_{4\bar{\zeta}}(\Psi_4) z_4^T - \sigma_4 \widehat{W}_{4\bar{\zeta}}]. \end{aligned} \quad (48)$$

Similarly,  $S_{4\zeta}(\Psi_4)$  is the subvector of  $S_4(\Psi_4)$  consisting of the RBFs near the orbits  $\Psi_4(t)$ , and  $\widehat{W}_{4\zeta}$  is the corresponding weight submatrix of  $\widehat{W}_4$ . Moreover,  $\epsilon_{4\zeta} = \epsilon_4(\Psi_4) - \widehat{W}_{4\zeta}^T S_{4\zeta}(\Psi_4)$  is the NN approximation error along the orbits  $\Psi_4(t)$ . Since  $\|\widehat{W}_{4\zeta}^T S_{4\zeta}(\Psi_4)\|$  is small,  $\|\epsilon_{4\zeta}\|$  is close to  $\|\epsilon_4(\Psi_4)\|$ .

It would be shown that the perturbation term  $J^{-1}\epsilon_{4\zeta}$  may be large, which will make the accurate convergence of neural weights become difficult. To solve this problem, a state transformation  $z_{4s} = Jz_4$  is introduced to eliminate the influence

of  $J^{-1}\epsilon_{4\zeta}$ . Subsequently, set  $\widehat{W}_{4\zeta} = [\widehat{W}_{41\zeta}, \widehat{W}_{42\zeta}, \dots, \widehat{W}_{4n\zeta}] \in R^{\zeta' \times n}$  and  $\widehat{W}_{4\bar{\zeta}} = [\widehat{W}_{41\bar{\zeta}}, \widehat{W}_{42\bar{\zeta}}, \dots, \widehat{W}_{4n\bar{\zeta}}] \in R^{\zeta' \times n}$ ; (46) can be transformed into the following form:

$$\begin{aligned} \begin{bmatrix} \dot{z}_{4s} \\ \dot{\widehat{W}}_{41\zeta} \\ \vdots \\ \dot{\widehat{W}}_{4n\zeta} \end{bmatrix} &= \begin{bmatrix} A_4(t) & B_4(t) \\ C_4(t) & 0 \end{bmatrix} \begin{bmatrix} z_{4s} \\ \widehat{W}_{41\zeta} \\ \vdots \\ \widehat{W}_{4n\zeta} \end{bmatrix} \\ &+ \begin{bmatrix} \epsilon'_{4\zeta} \\ -\sigma_4 \Gamma_{4\zeta} \widehat{W}_{41\zeta} \\ \vdots \\ -\sigma_4 \Gamma_{4\zeta} \widehat{W}_{4n\zeta} \end{bmatrix}, \end{aligned} \quad (49)$$

where

$$\begin{aligned} A_4(t) &= -C_4 J^{-1} \in R^{n \times n} \\ B_4(t) &= -\text{diag}\{S_{4\zeta}^T, \dots, S_{4\zeta}^T\} \in R^{n \times \zeta' n} \\ C_4(t) &= \bar{\Gamma}_{4\zeta} \bar{S}_{4\zeta} J^{-1} \in R^{\zeta' n \times n} \\ \bar{\Gamma}_{4\zeta} &= \text{diag}\{\Gamma_{4\zeta}, \dots, \Gamma_{4\zeta}\} \in R^{\zeta' n \times \zeta' n} \\ \bar{S}_{4\zeta} &= \text{diag}\{S_{4\zeta}, \dots, S_{4\zeta}\} \in R^{\zeta' n \times n} \\ \epsilon'_{4\zeta} &= -z_3 + \epsilon_{4\zeta}. \end{aligned} \quad (50)$$

Using the similar step and choosing  $P'(t) = J$ , we have

$$P'(t) + P'(t) A_4(t) + A_4^T(t) P'(t) = -2J C_4 J^{-1} < 0 \quad (51)$$

with  $C_4 > 0$ . Then, it can be proven that  $\widehat{W}_{4\zeta}$  converges to a small neighborhood of optimal weights  $W_{4\zeta}^*$  in a finite time  $T_2$ , and the constant weights  $\overline{W}_4$  can be obtained from (40). According to the localization property of RBF NN, the system dynamics  $H_4(\Psi_4)$  can be described by

$$H_4(\Psi_4) = \overline{W}_4^T S_4(\Psi_4) + \bar{\epsilon}_4(\Psi_4), \quad (52)$$

where  $\bar{\epsilon}_4(\Psi_4)$  is close to  $\epsilon_4(\Psi_4)$  due to the convergence of  $\widehat{W}_{4\zeta}$ . Therefore, the dynamics  $H_i(\Psi_i)$ ,  $i = 1, 2$ , can be accurately approximated by the constant RBF NN  $\overline{W}_i^T S_i(\Psi_i)$  with the stored knowledge  $\overline{W}_i$  obtained in (40).

**4.2. Neural Learning Control Using the Stored Knowledge.** Since the locally accurate NN approximation can be achieved by the constant RBF NN  $\overline{W}_i^T S_i(\Psi_i)$ , for a similar control task, we will reuse the knowledge  $\overline{W}_i$  to design a neural learning controller for system (1):

$$u = -z_3 - C_4 z_4 - \overline{W}_4^T S_4(\Psi_4) \quad (53)$$

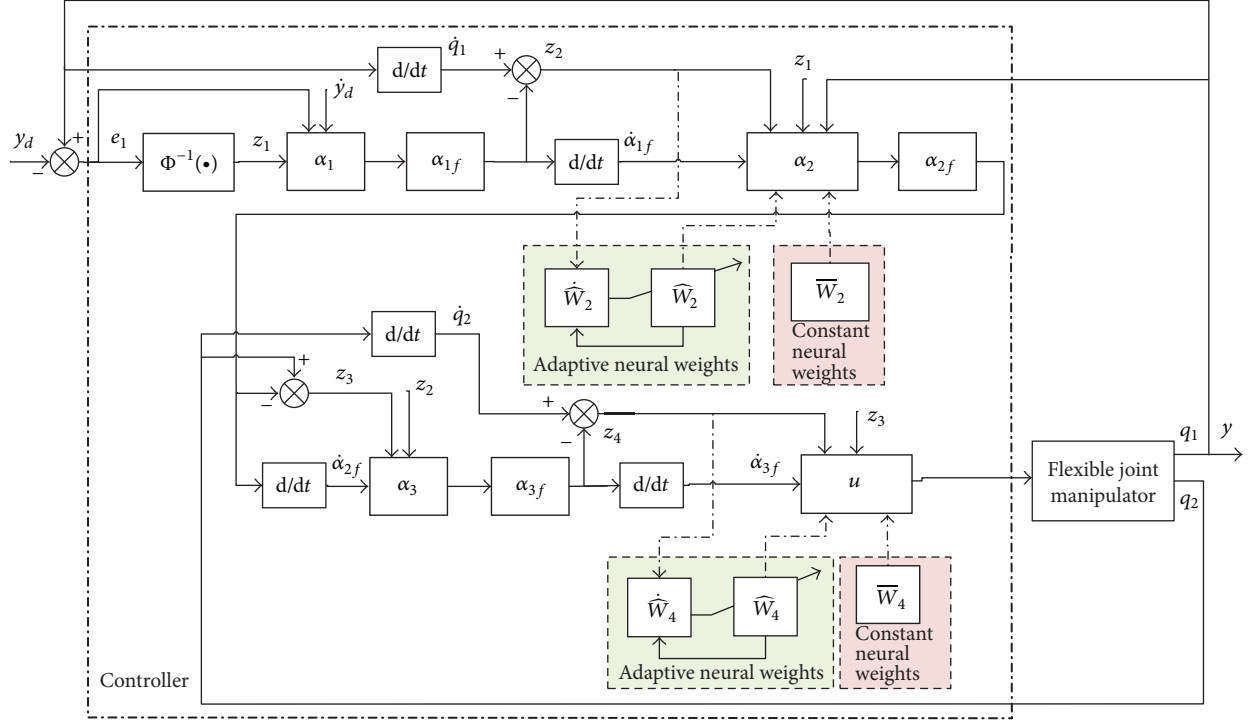


FIGURE 2: Block diagram of the proposed learning control scheme.

with virtual controller

$$\begin{aligned}\alpha_1 &= -(\Upsilon^{-1}C_1 + \Upsilon)z_1 + \dot{y}_d + \frac{\hat{\beta}(t)}{\beta(t)}e_1 \\ \alpha_2 &= -\Upsilon z_1 - C_2 z_2 - \bar{W}_2^T S_2(\Psi_2) + q_1 \\ \alpha_3 &= -z_2 - C_3 z_3 + \dot{\alpha}_{2f},\end{aligned}\quad (54)$$

where  $\bar{W}_2^T S_2(\Psi_2)$  and  $\bar{W}_4^T S_4(\Psi_4)$  are the accurate neural approximators of the unknown system dynamics  $H_2(\Psi_2)$  and  $H_4(\Psi_4)$ , respectively.  $\bar{W}_2$  and  $\bar{W}_4$  are constant weight matrices, which are obtained from the previous control process. Similar to the proof of Theorem 3, another Lyapunov function candidate is constructed as

$$V = \frac{1}{2} \sum_{i=1}^4 z_i^T K_i z_i + \frac{1}{2} \sum_{l=1}^3 y_l^T y_l, \quad (55)$$

where  $K_1 = I$ ,  $K_2 = K^{-1}M(q_1)$ ,  $K_3 = I$ , and  $K_4 = J$ .

**Theorem 5.** Consider the closed-loop system consisting of the manipulator model (1), the reference trajectory  $y_d$ , the prescribed performance bounds (8), the state transformation (11), the neural learning controller (53) with the constant weights given by (40), and Lyapunov function (55). Then, for initial conditions  $x_d(0)$  which generate the same or similar recurrent reference orbit  $\psi_d(x_d(0))$  as in Theorem 4, and initial conditions  $x(0)$  satisfying the prescribed performance (8) and  $V(0) \leq \mu$  with given  $\mu > 0$ , such that all the closed-loop signals remain uniformly ultimately bounded, and the tracking

error converges to a small neighborhood around zero with the prescribed performance (8) when initial conditions  $x(0)$  are in a close vicinity of  $\psi_d(x_d(0))$ .

*Proof.* See Appendix B.  $\square$

**Remark 6.** For clarity, a block diagram of the proposed schemes is shown in Figure 2. From Figure 2, the main difference between adaptive neural control and learning neural control lies in the adaptation of NN weights. The neural weights  $\hat{W}_2$  and  $\hat{W}_4$  are updated online in the adaptive neural control process, while the stored constant weights  $\bar{W}_2$  and  $\bar{W}_4$  are reused in the neural learning process for the same or similar control task. Without the repeat adjustment of the neural weights, the neural learning controller (53) and (54) can achieve the better control performance with faster tracking convergence rate and smaller tracking error.

## 5. Simulation and Experiment

In this section, to illustrate the effectiveness of the proposed approach, a single-link manipulator system with flexible joint is considered by the following form:

$$\begin{aligned}I\ddot{q}_1 + MgL \sin(q_1) + K(q_1 - q_2) &= 0 \\ J\ddot{q}_2 - K(q_1 - q_2) &= u,\end{aligned}\quad (56)$$

where  $M$  is the mass,  $g$  is the gravitational acceleration, and  $L$  is the length of link. Figure 3 illustrates the structure of a single-link flexible joint manipulator. In the simulation, the system parameter is chosen exactly as  $M = 2.3$ ,  $L = 1$ ,



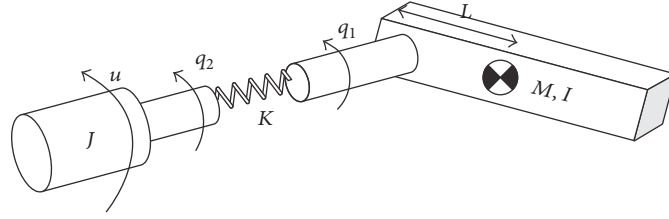


FIGURE 3: Single-link flexible joint manipulator.

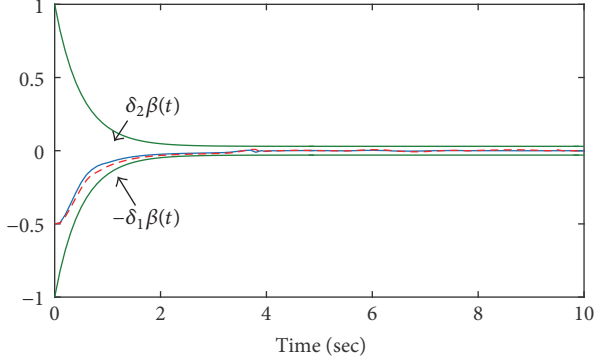


FIGURE 4: Tracking error: LC (-) and ANC (- -).

$g = 9.8$ ,  $I = ML^2$ ,  $J = 0.5$ , and  $K = 15$ . Then the desired reference trajectory  $y_d$  is generated from the following equation:

$$2\ddot{y}_d + 3\dot{y}_d + y_d = 4 \sin(1.5t). \quad (57)$$

**5.1. Numerical Simulation Results.** Based on Theorem 3, the adaptive control law is chosen as (36), and the virtual controllers are chosen as (15), (23), and (28) with the virtual control laws (24) and (37). In the performance function,  $\beta_0 = 1$ ,  $\beta_\infty = 0.05$ ,  $\kappa = 2$ , and  $\delta_1 = \delta_2 = 1$ .  $\widehat{W}_2^T S_2(\Psi_2)$  is constructed by 121 neurons whose centers evenly spaced on  $[-0.8, 0.8] \times [-2, 2]$  and widths  $\eta_{21} = 0.2$ ,  $\eta_{22} = 0.5$ . Similarly,  $\widehat{W}_4^T S_4(\Psi_4)$  is constructed by 2783 neurons whose centers evenly spaced on  $[-0.8, 0.8] \times [-2, 2] \times [-11, 11]$  and widths  $\eta_{41} = 0.2$ ,  $\eta_{42} = 0.5$ ,  $\eta_{43} = 1.25$ . Other design parameters are  $c_1 = 2$ ,  $c_2 = 5$ ,  $c_3 = 2$ ,  $c_4 = 30$ ,  $\Gamma_2 = 5$ ,  $\Gamma_4 = 60$ , and  $\sigma_2 = \sigma_4 = 0.0001$ . The initial states are  $[q_1(0), \dot{q}_1(0), q_2(0), \dot{q}_2(0)]^T = [0.5, 0, 0.2, 0]^T$  and  $[y_d(0), \dot{y}_d(0)]^T = [1, 0]^T$ . The initial weights  $\widehat{W}_2(0)$  and  $\widehat{W}_4(0)$  are both set as zero vectors.

According to (40), the constant neural weights  $\overline{W}_2$  and  $\overline{W}_4$  in (53) and (54) are obtained as  $\overline{W}_2 = \text{mean}_{t \in [200, 300]} \widehat{W}_2(t)$  and  $\overline{W}_4 = \text{mean}_{t \in [400, 500]} \widehat{W}_4(t)$ . In order to be compared with the adaptive neural DSC results, in this simulation, the parameters in performance function and the initial states are set the same as the values set in adaptive neural DSC, while the control gain are set as  $c_1 = 1$ ,  $c_2 = 4$ ,  $c_3 = 2$ , and  $c_4 = 5$ .

The related simulation results are shown in Figures 4–10. Figure 4 illustrates that, for the proposed adaptive neural DSC scheme, the tracking error  $e_1$  can ultimately

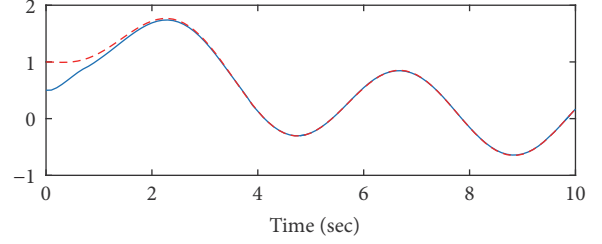
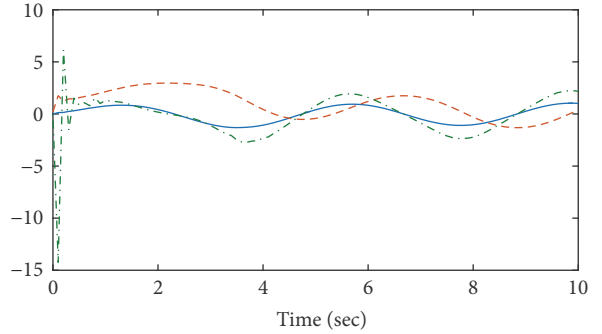
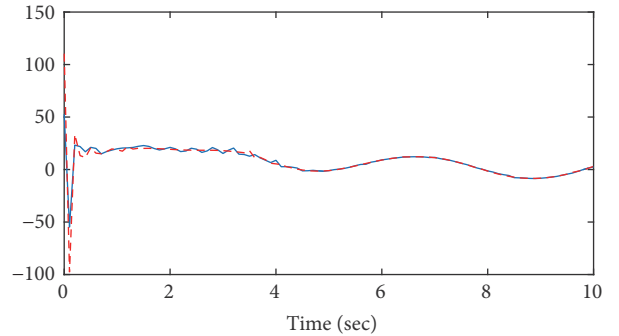
FIGURE 5: System output:  $x_1$  (-) and  $y_d$  (- -).FIGURE 6: State variables  $x_2$  (-),  $x_3$  (- -),  $x_4$  (-.-).

FIGURE 7: Control input: LC (-) and ANC (- -).

converge to a small neighborhood of zero besides satisfying the prescribed performance; for the proposed learning control scheme, not only does the tracking error satisfy the prescribed performance, but also the convergence rate is faster under the similar control input amplitude shown in Figure 7. Moreover, the time consumption is decreased by 2/3 in contrast with adaptive neural DSC scheme. Figure 5 shows that the output  $x_1$  of system (56) tracks to the reference

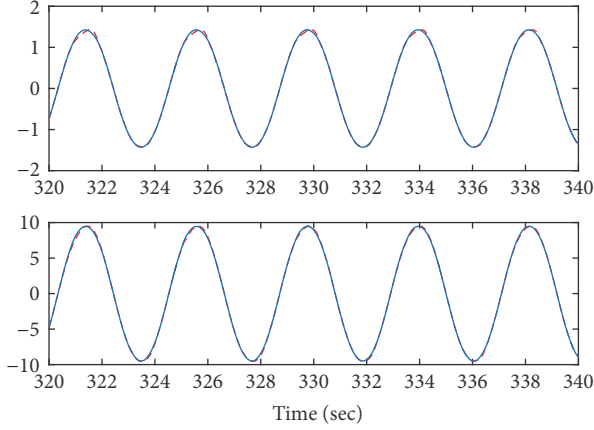


FIGURE 8: Function approximation:  $\widehat{W}_i^T S_i(\Psi_i)$  (- -) and  $h_i(\Psi_i)$  (-) ( $i = 2, 4$ ).

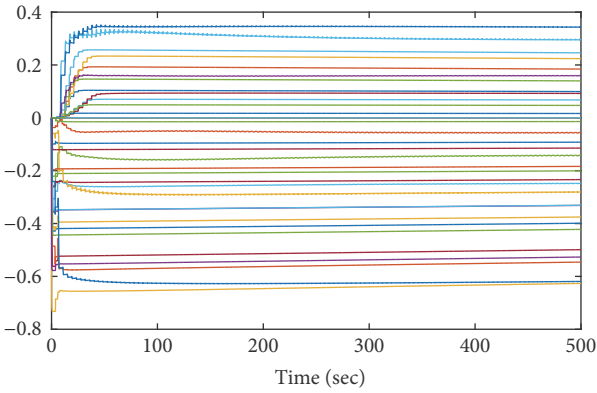


FIGURE 9: The convergence of partial NN weights  $\widehat{W}_2$ .

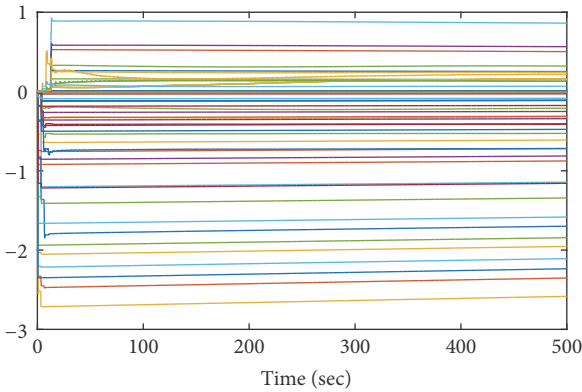


FIGURE 10: The convergence of partial NN weights  $\widehat{W}_4$ .

trajectory  $y_d$  quickly. Figure 6 shows that other state variables are bounded for the proposed scheme. Figure 8 gives the NN approximation ability for the unknown dynamics. Figures 9 and 10 show that the NN weights converge to certain values along with the updated laws of NN weights.

*Remark 7.* In order to compare with the difference of tracking performance between different parameter selection in (3),

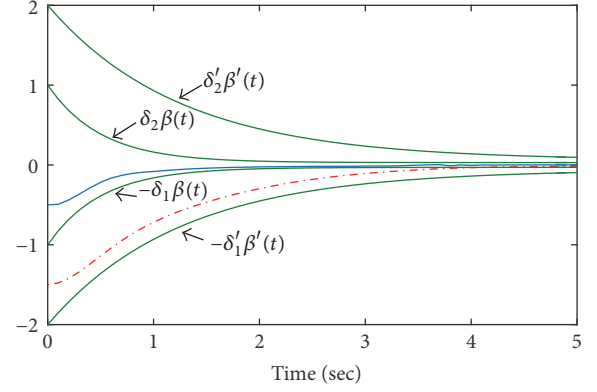


FIGURE 11: Tracking error:  $e$  (-) and  $e'$  (-.-).

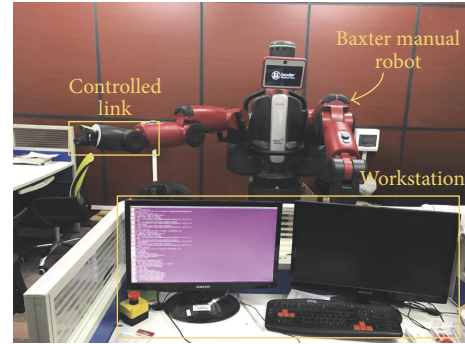


FIGURE 12: Overview of the experimental platform.

these parameters in (3) are chosen as  $\beta_0 = 1$ ,  $\beta_\infty = 0.05$ ,  $\kappa = 2$ , and  $\delta_1 = \delta_2 = 1$ ; the others are set as  $\beta'_0 = 1$ ,  $\beta'_\infty = 0.05$ ,  $\kappa' = 1$ , and  $\delta'_1 = \delta'_2 = 2$ . Define the tracking error of the former as  $e$  and the latter one as  $e'$ . Figure 11 illustrates two different tracking performances of the system's output by selecting different parameters of constraints. It is evident that the tracking performance is affected by the parameter selection of performance function. And as the range of permissible error gets smaller, the tracking error is forced to converge faster and the tracking accuracy gets higher.

*5.2. Experiment Results on Baxter Robot.* Moreover, in order to validate the effectiveness of the proposed control scheme, the Baxter bimanual robot is used in the experiment, as shown in Figure 12. It is of two 7-DOF arms and advanced sensing technologies, including position, force, and torque sensors and control at every joint. The resolution for the joint sensors is 14 bits with 360 degrees (0.022 degrees per tick resolution), while the maximum joint torques that can be applied to the joints are 50 Nm (the first four joints) and 15 Nm (the last three joints). To compare with the simulation's results, the desired reference trajectory in the experiment is the same as one in the simulation, which is also generated from (57). In the experiment, one of the Baxter robot's links (such as the wrist's link of robot's right arm) is commanded to track the desired reference trajectory, and the tracking

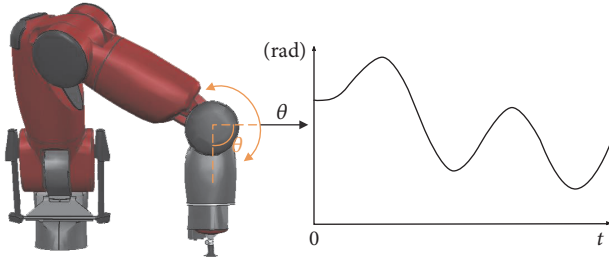


FIGURE 13: The desired motion of the robot's link.

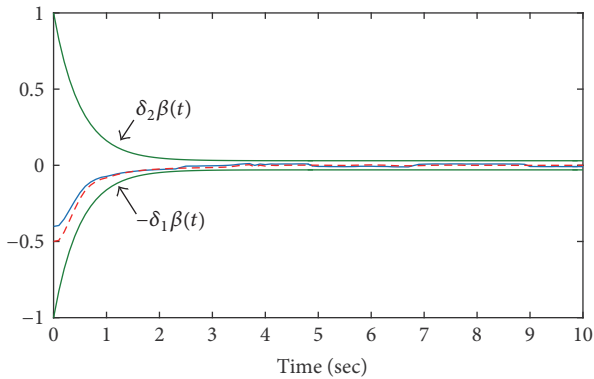


FIGURE 14: Tracking error: LC (-) of Baxter robot and LC (-) of simulation.

error between link's angle position and reference trajectory is forced to converge to a small neighborhood around zero with prescribed performance in a finite time, while other links of the robot stay still. Figure 13 illustrates the desired motion of the robot's link.

To verify the effectiveness of neural learning control scheme for Baxter robot, the constant weights  $\bar{W}_2$  and  $\bar{W}_4$  in Section 5.1 are reused in experiment. For comparison, the parameters in performance function and the initial states are set the same as these values in neural learning control, while the control gains are set as  $c_1 = 1.5$ ,  $c_2 = 3$ ,  $c_3 = 2$ , and  $c_4 = 4$ .

The related results are shown in Figures 14–16. Figure 14 shows the difference between LC of simulation and LC of Baxter robot. Although the vibration of the robot affects the tracking performance, it is evident that the tracking error of Baxter robot can also ultimately converge to a small neighborhood of zero besides satisfying the prescribed performance. Figure 15 show that the output of Baxter robot's link (i.e.,  $q_1$  of system (56)) tracks to the reference trajectory  $y_d$  quickly, and other state variables are bounded for the proposed learning control scheme. Figure 16 shows that the control input of Baxter robot is also bounded with small overshoot and small mechanical vibration.

## 6. Conclusion

In this paper, we studied learning from adaptive neural dynamic surface control for a class of flexible joint manipulator with unknown dynamics under the prescribed constraint.

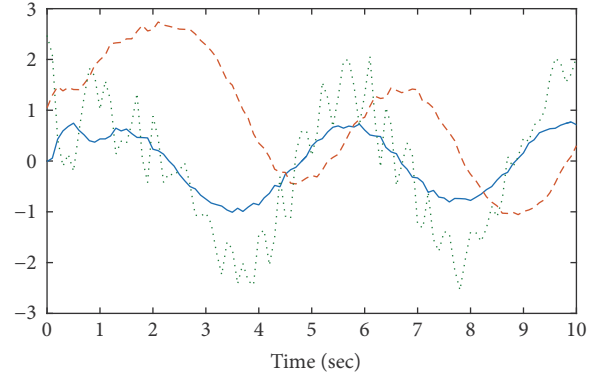
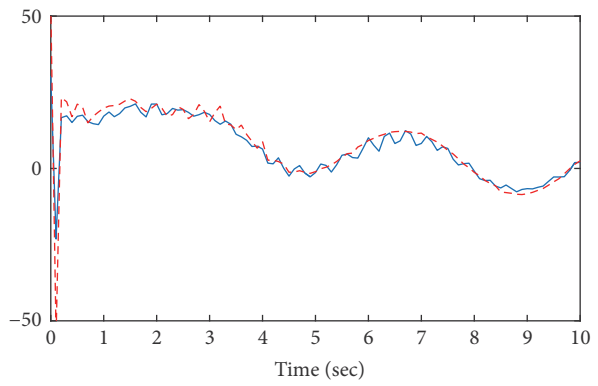
FIGURE 15: State variables  $q_1$  (-),  $q_2$  (-), and  $q_3$  (--) of Baxter robot.

FIGURE 16: Control input: LC (-) of Baxter robot and LC (-) of simulation.

A novel error transformed function was utilized to transform the constrained tracking problem into the equivalent unconstrained one so as to facilitate the controller design. Furthermore, by combining DSC method, which was used to reduce the number of NN approximators and decrease the dimension of NN inputs, a novel adaptive neural control scheme was proposed to guarantee the prescribed performance during the transient process. Then, the closed-loop stability and control performance were achieved according to the construction of the Lyapunov function. After the stable control process, since two NNs were used in the controller design, the recurrent property of the NN input variables and the partial PE condition of RBF NNs were proved recursively. Therefore, the locally accurate approximations of unknown system dynamics by RBF NNs were achieved, and the proposed control scheme was verified to be capable of storing the learned knowledge in constant RBF NNs. Finally, the stored knowledge was reused to develop the neural learning controller for the same system model and the same or similar control task, so that the closed-loop stability and better control performance were achieved under the prescribed constraint. Simulation results for a single-link flexible joint manipulator and experiment results for Baxter robot were presented to prove the effectiveness of the proposed control scheme.

## Appendix

### A. Proof of Theorem 3

Firstly, the derivative of  $y_1$ ,  $y_2$ , and  $y_3$  can be expressed as

$$\dot{y}_l = \dot{a}_{lf} - \dot{a}_l = -\frac{y_l}{\tau_l} + D_l(\cdot), \quad l = 1, 2, 3, \quad (\text{A.1})$$

where  $D_1(\cdot) := D_1(z_1, z_2, y_1, y_d, \dot{y}_d, \ddot{y}_d) = C_1 \dot{z}_1 - \dot{y}_d - \dot{\beta}(t)/\beta(t)e_1 - \dot{\beta}(t)/\beta(t)[\dot{e}_1 - \dot{\beta}(t)/\beta(t)e_1]$ . Similarly,  $D_2(\cdot) := D_2(z_1, z_2, z_3, y_1, y_2, \widehat{W}_2, y_d, \dot{y}_d, \ddot{y}_d)$ ,  $D_3(\cdot) := D_3(z_1, z_2, z_3, z_4, y_1, y_2, y_3, \widehat{W}_2, y_d, \dot{y}_d, \ddot{y}_d)$ , and they are continuous functions.  $\dot{V}$  can be expressed as

$$\begin{aligned} \dot{V} = & -\sum_{i=1}^4 z_i^T K_i^T z_i + z_1^T Y y_1 + \sum_{j=2}^3 z_j^T y_j + z_2^T \epsilon_2 + z_4^T \epsilon_4 \\ & - \sum_{k=1}^2 \text{tr} \left[ \sigma_{2k} \widehat{W}_{2k}^T \widehat{W}_{2k} \right] \\ & + \sum_{l=1}^3 \left[ -\frac{y_l^T y_l}{\tau_l} + y_l^T D_l(\cdot) \right], \end{aligned} \quad (\text{A.2})$$

where  $K_1' = C_1 + Y^T Y$ ,  $K_2' = C_2 + K^{-1} C(q_1, \dot{q}_1) - K^{-1} \dot{M}(q_1)/2$ ,  $K_3' = C_3$ ,  $K_4' = C_4$ . From  $\widehat{W}_{2k} = \widehat{W}_{2k} - W_{2k}^*$  ( $k = 1, 2$ ), the following inequality can be obtained:

$$\begin{aligned} -\text{tr} \left[ \sigma_{2k} \widehat{W}_{2k}^T \widehat{W}_{2k} \right] \leq & -\frac{1}{2} \text{tr} \left[ \sigma_{2k} \widehat{W}_{2k}^T \widehat{W}_{2k} \right] \\ & + \frac{1}{2} \text{tr} \left[ \sigma_{2k} W_{2k}^{*T} W_{2k}^* \right]. \end{aligned} \quad (\text{A.3})$$

In addition, since  $y_d$ ,  $\dot{y}_d$ , and  $\ddot{y}_d$  are recurrent signals, there exists a compact set  $\Omega_d := \{(y_d, \dot{y}_d, \ddot{y}_d) : y_d^2 + \dot{y}_d^2 + \ddot{y}_d^2 \leq B_0\}$  with  $B_0 > 0$ . Furthermore, the set  $\Omega := \{V \leq \mu\}$  is also compact by function (39), which implies that  $\Omega_d \times \Omega$  is compact as well. Consequently,  $D_1(\cdot)$ ,  $D_2(\cdot)$ , and  $D_3(\cdot)$  are in the compact set  $\Omega_d \times \Omega$ . Therefore, there exists a maximum constant  $N_l$  for  $D_l(\cdot)$  on  $\Omega_d \times \Omega$  such that  $\|D_l(\cdot)\| \leq N_l$ ,  $l = 1, 2, 3$ . According to Young's inequality, the following inequalities are obtained:

$$\begin{aligned} z_1^T Y y_1 & \leq z_1^T Y Y^T z_1 + \frac{y_1^T y_1}{4} \\ z_j^T y_j & \leq z_j^T z_j + \frac{y_j^T y_j}{4}, \quad j = 2, 3 \\ z_k^T \epsilon_k & \leq z_k^T z_k + \frac{\epsilon_k^{*2}}{4}, \quad k = 2, 4 \\ y_l^T D_l(\cdot) & \leq \frac{y_l^T y_l}{(4\rho)} + N_l^2 \rho, \quad l = 1, 2, 3, \end{aligned} \quad (\text{A.4})$$

where  $\rho$  is a positive design parameter. Let  $C_2 > 2I$ ,  $C_3 > I$ , and  $C_4 > I$ . Then the inequality for  $\dot{V}$  can be obtained:

$$\begin{aligned} \dot{V} \leq & -\sum_{i=1}^4 c_i z_i^T K_i z_i - \sum_{k=1}^2 \text{tr} \left[ \frac{\sigma_{2k}}{2} \widehat{W}_{2k}^T \widehat{W}_{2k} \right] - \sum_{l=1}^3 \xi_l y_l^T y_l \\ & + \sum_{k=1}^2 \text{tr} \left[ \frac{\sigma_{2k}}{2} W_{2k}^{*T} W_{2k}^* \right] + (N_1^2 + N_2^2 + N_3^2) \rho \\ & + \frac{\epsilon_2^{*2}}{4} + \frac{\epsilon_4^{*2}}{4}, \end{aligned} \quad (\text{A.5})$$

where  $c_1 = \lambda_{\min}(C_1)$ ,  $c_2 = \lambda_{\min}[(C_2 - 2I)M^{-1}(q_1)K]$ ,  $c_3 = \lambda_{\min}(C_3 - I)$ ,  $c_4 = \lambda_{\min}[(C_4 - I)J^{-1}]$ ,  $\xi_l = 1/\tau_l - N_l^2/(4\rho) - 1/4$  ( $l = 1, 2, 3$ ). Choose  $\tau_l$  ( $l = 1, 2, 3$ ) appropriately such that  $\xi_l > 0$ . Moreover, set values of  $C_1, C_2, C_3, C_4, \Gamma_2, \sigma_2, \Gamma_4, \sigma_4$  appropriately and (A.5) can be rewritten as

$$\dot{V} \leq -2k_0 V + b_0, \quad (\text{A.6})$$

where  $b_0 = \text{tr}[\sigma_2 W_2^{*T} W_2^*]/2 + \text{tr}[\sigma_4 W_4^{*T} W_4^*]/2 + (N_1^2 + N_2^2 + N_3^2)\rho + \epsilon_2^{*2}/4 + \epsilon_4^{*2}/4$  and  $k_0 = \min\{c_1, c_2, c_3, c_4, \sigma_2 \lambda_{\min}(\Gamma_2)/2, \sigma_4 \lambda_{\min}(\Gamma_4)/2, \xi_1, \xi_2, \xi_3\} > 0$ . Let  $\nu = b_0/(2k_0)$ ; then

$$V \leq [V(0) - \nu] e^{-2k_0 t} + \nu \quad (\text{A.7})$$

which means that, as  $t \rightarrow \infty$ ,  $V \leq \nu$ , thus

$$\|z_i\| \leq \sqrt{\frac{2\nu}{\lambda_{\min}(K_i)}}, \quad i = 1, 2, 3, 4. \quad (\text{A.8})$$

Since  $\nu$  can be made arbitrarily small by choosing appropriate design parameters  $k_0, \rho, \sigma_2$ , and  $\sigma_4$ . Thus, for given  $\nu' > \nu$ , there exists a finite time  $T_1$  so that  $\|z_i\| \leq \sqrt{2\nu'/\lambda_{\min}(K_i)}$  for all  $t \geq T_1$ . Based on the error transformation (11), the tracking error  $e_1$  converges to a small neighborhood around zero with guaranteed prescribed performance (3) in a finite time  $T_1$  as well, and all the signals in the closed-loop system are uniformly ultimately bounded.

### B. Proof of Theorem 5

Similar to the adaptive neural DSC design in the Section 3, the derivative of  $z_i$  ( $i = 1, 2, 3, 4$ ) along (53) and (54) yields

$$\begin{aligned} \dot{z}_1 & = -(C_1 + Y^T Y) z_1 + Y z_2 + Y y_1. \\ \dot{z}_2 & = M^{-1}(q_1) K \left( -Y z_1 - C_2 z_2 + z_3 + y_2 \right. \\ & \quad \left. - \overline{W}_2^T S_2(\Psi_2) + H_2(\Psi_2) - K^{-1} C(q_1, \dot{q}_1) z_2 \right) \\ \dot{z}_3 & = -z_2 - C_3 z_3 + z_4 + y_3 \\ \dot{z}_4 & = J^{-1} \left[ -z_3 - C_4 z_4 - \overline{W}_4^T S_4(\Psi_4) + H_4(\Psi_4) \right] \end{aligned} \quad (\text{B.1})$$

According to Theorem 4, there exists a small positive constant  $\epsilon_i'$  for the recurrent orbit  $\psi_{d_i}(x_{d_i}(t))$  ( $i = 2, 4$ ), such



that  $\|\overline{W}_i^T S_i(\Psi_i) - H_i(\Psi_i)\| < \epsilon'_i$ . Thus, by Young's inequality, the derivative of  $V$  along (A.1) and (B.1) yields

$$\begin{aligned} \dot{V} \leq & -\sum_{i=1}^4 c'_i z_i^T K_i z_i - \sum_{l=1}^3 \xi_l y_l^T y_l + (N_1^2 + N_2^2 + N_3^2) \rho \\ & + \frac{\epsilon_2'^2}{4} + \frac{\epsilon_4'^2}{4}, \end{aligned} \quad (\text{B.2})$$

where  $c'_1 = \lambda_{\min}(C_1)$ ,  $c'_2 = \lambda_{\min}[(C_2 - 2I)M^{-1}(q_1)K]$ ,  $c'_3 = \lambda_{\min}(C_3 - I)$ ,  $c'_4 = \lambda_{\min}[(C_4 - I)J^{-1}]$ . Let  $C_2 > 2I$ ,  $C_3 > I$ , and  $C_4 > I$ ; (B.2) can be rewritten as

$$\dot{V} \leq -2k'_0 V + b'_0, \quad (\text{B.3})$$

where  $b'_0 = (N_1^2 + N_2^2 + N_3^2)\rho + \epsilon_2'^2/4 + \epsilon_4'^2/4$  and  $k'_0 = \min\{c'_1, c'_2, c'_3, c'_4, \xi_1, \xi_2, \xi_3\} > 0$ . Let  $\nu' = b'_0/(2k'_0)$ ; then, based on the similar analysis in Theorem 3, it can be concluded that the tracking error  $e_1$  converges to a small neighborhood around zero with the prescribed performance, and all the signals in the closed-loop system are uniformly ultimately bounded.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

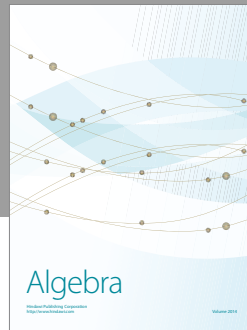
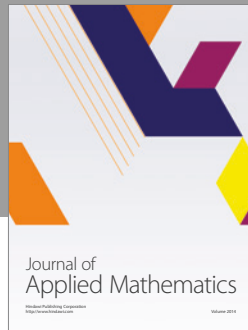
## Acknowledgments

The authors would like to thank Professor Chenguang Yang for the experiment platform. Moreover, this work was partially supported by the National Natural Science Foundation of China (nos. 61773169, 61374119, 61473121, and 61611130214), the Royal Society Newton Mobility Grant IE150858, the Guangdong Natural Science Foundation under Grants 2017A030313369 and 2017A030313381, the youth talent of Guangdong Province, and the Fundamental Research Funds for the Central Universities.

## References

- [1] M. W. Spong and M. Vidyasagar, *Robot Dynamics and Control*, Chap. 13, Wiley, New York, NY, USA, 1989.
- [2] J. H. Oh and J. S. Lee, "Control of flexible joint robot system by backstepping design approach," in *Proceedings of International Conference on Robotics and Automation*, vol. 4, pp. 3435–3440, 1997.
- [3] J. Huang and J. Lin, "Backstepping Control Design of a Single-Link Flexible Robotic Manipulator," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 11775–11780, 2008.
- [4] Y. Soukkou and S. Labiod, "Adaptive backstepping control using combined direct and indirect adaptation for a single-link flexible-joint robot," *International Journal of Industrial Electronics and Drives*, vol. 2, no. 1, p. 11, 2015.
- [5] J.-J. E. Slotine and S. Hong, "Two-time scale sliding control of manipulators with flexible joints," in *Proceedings of the 1986 American Control Conference*, pp. 805–810, 1986.
- [6] A.-C. Huang and Y.-C. Chen, "Adaptive sliding control for single-link flexible-joint robot with mismatched uncertainties," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 5, pp. 770–775, 2004.
- [7] B. Xu and P. Zhang, "Minimal-learning-parameter Technique Based Adaptive Neural Sliding Mode Control of MEMS Gyroscope," *Complexity*, Article ID 6019175, 8 pages, 2017.
- [8] B. Xu and P. Zhang, "Composite Learning Sliding Mode Control of Flexible-link Manipulator," *Complexity*, Article ID 9430259, 6 pages, 2017.
- [9] C.-I. Morarescu and B. Brogliato, "Passivity-based switching control of flexible-joint complementarity mechanical systems," *Automatica*, vol. 46, no. 1, pp. 160–166, 2010.
- [10] Y. Li, S. Tong, and T. Li, "Adaptive fuzzy output feedback control for a single-link flexible robot manipulator driven DC motor via backstepping," *Nonlinear Analysis: Real World Applications*, vol. 14, no. 1, pp. 483–494, 2013.
- [11] F. Abdollahi, H. A. Talebi, and R. V. Patel, "A stable neural network-based observer with application to flexible-joint manipulators," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 17, no. 1, pp. 118–129, 2006.
- [12] S. J. Yoo, "Distributed adaptive containment control of networked flexible-joint robots using neural networks," *Expert Systems with Applications*, vol. 41, no. 2, pp. 470–477, 2014.
- [13] I. F. Jasim and P. W. Plapper, "Robust direct adaptive fuzzy control of flexible joints robots with time-varying stiffness/damping parameters," in *Proceedings of the Joint Conference of ISR 2014 - 45th International Symposium on Robotics and Robotik 2014 - 8th German Conference on Robotics, ISR/ROBOTIK 2014*, pp. 149–156, June 2014.
- [14] E. Psomopoulou, A. Theodorakopoulos, Z. Doulgeri, and G. A. Rovithakis, "Prescribed Performance Tracking of a Variable Stiffness Actuated Robot," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 5, pp. 1914–1926, 2015.
- [15] L. Zhang, Z. Li, and C. Yang, "Adaptive neural network based variable stiffness control of uncertain robotic systems using disturbance observer," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 3, pp. 2236–2245, 2017.
- [16] C. Yang, J. Luo, Y. Pan, Z. Liu, and C. Su, "Personalized variable gain control with tremor attenuation for robot teleoperation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–12, 2017.
- [17] C. Yang, X. Wang, Z. Li, Y. Li, and C. Su, "Teleoperation control based on combination of wave variable and neural networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017.
- [18] M. Krstic, I. Kanellakopoulos, and P. Kokotovic, *Nonlinear and Adaptive Control Design*, Wiley, NY, USA, 1995.
- [19] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice-Hall, Upper Saddle River, NJ, USA, 2nd edition, 1999.
- [20] S. S. Ge and C. Wang, "Direct adaptive NN control of a class of nonlinear systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 13, no. 1, pp. 214–221, 2002.
- [21] D. Swaroop, J. K. Hedrick, P. P. Yip, and J. . Gerdes, "Dynamic surface control for a class of nonlinear systems," *Institute of Electrical and Electronics Engineers Transactions on Automatic Control*, vol. 45, no. 10, pp. 1893–1899, 2000.
- [22] D. Wang and J. Huang, "Neural network-based adaptive dynamic surface control for a class of uncertain nonlinear systems in strict-feedback form," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 16, no. 1, pp. 195–202, 2005.
- [23] S. Gao, H. Dong, and B. Ning, "Observer-based nonlinear feedback decentralized neural adaptive dynamic surface control

- for large-scale nonlinear systems," *International Journal of Adaptive Control and Signal Processing*, 2017.
- [24] S. J. Yoo, J. B. Park, and Y. H. Choi, "Adaptive output feedback control of flexible-joint robots using neural networks: Dynamic surface design approach," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 19, no. 10, pp. 1712–1726, 2008.
- [25] S. Gao, H. Dong, S. Lyu, and B. Ning, "Truncated adaptation design for decentralised neural dynamic surface control of interconnected nonlinear systems under input saturation," *International Journal of Control*, vol. 89, no. 7, pp. 1447–1466, 2016.
- [26] S. Gao, B. Ning, and H. Dong, "Fuzzy dynamic surface control for uncertain nonlinear systems under input saturation via truncated adaptation approach," *Fuzzy Sets and Systems*, vol. 290, pp. 100–117, 2016.
- [27] M. Wang, X. Liu, and P. Shi, "Adaptive neural control of pure-feedback nonlinear time-delay systems via dynamic surface technique," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 41, no. 6, pp. 1681–1692, 2011.
- [28] W. He, Y. Chen, and Z. Yin, "Adaptive neural network control of an uncertain robot with full-state constraints," *IEEE Transactions on Cybernetics*, vol. 46, no. 3, pp. 620–629, 2016.
- [29] W. He, A. O. David, Z. Yin, and C. Sun, "Neural network control of a robotic manipulator with input deadzone and output constraint," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 6, pp. 759–770, 2016.
- [30] C. P. Bechlioulis and G. A. Rovithakis, "Robust adaptive control of feedback linearizable MIMO nonlinear systems with prescribed performance," *Institute of Electrical and Electronics Engineers Transactions on Automatic Control*, vol. 53, no. 9, pp. 2090–2099, 2008.
- [31] A. K. Kostarigka, Z. Doulgeri, and G. A. Rovithakis, "Prescribed performance tracking for flexible joint robots with unknown dynamics and variable elasticity," *Automatica*, vol. 49, no. 5, pp. 1137–1147, 2013.
- [32] M. Wang and H. Ye, "Adaptive neural dynamic surface control for flexible joint manipulator with prescribed performance," in *Proceedings of the 2017 29th Chinese Control And Decision Conference (CCDC)*, pp. 5311–5316, Chongqing, China, May 2017.
- [33] M. Wang, C. Wang, P. Shi, and X. Liu, "Dynamic learning from neural control for strict-feedback systems with guaranteed predefined performance," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 12, pp. 2564–2576, 2016.
- [34] C. Yang, X. Wang, L. Cheng, and H. Ma, "Neural-learning-based telerobot control with guaranteed performance," *IEEE Transactions on Cybernetics*, vol. 47, no. 10, pp. 3148–3159, 2017.
- [35] M. Wang, Y. Zhang, and H. Ye, "Dynamic Learning from Adaptive Neural Control of Uncertain Robots with Guaranteed Full-State Tracking Precision," *Complexity*, Article ID 5860649, 14 pages, 2017.
- [36] X. Xia, T. Zhang, Y. Yi, and Q. Shen, "Adaptive prescribed performance control of output feedback systems including input unmodeled dynamics," *Neurocomputing*, vol. 190, pp. 226–236, 2016.
- [37] C. Wang and D. J. Hill, *Deterministic Learning Theory for Identification, Recognition, and Control*, CRC Press, Boca Raton, FL, USA, 2009.
- [38] T. Liu, C. Wang, and D. J. Hill, "Learning from neural control of nonlinear systems in normal form," *Systems & Control Letters*, vol. 58, no. 9, pp. 633–638, 2009.
- [39] S.-L. Dai, C. Wang, and M. Wang, "Dynamic learning from adaptive neural network control of a class of nonaffine nonlinear systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 1, pp. 111–123, 2014.
- [40] M. Wang, C. Wang, and X. Liu, "Dynamic learning from adaptive neural control with predefined performance for a class of nonlinear systems," *Information Sciences*, vol. 279, pp. 874–888, 2014.
- [41] S.-L. Dai, C. Wang, and F. Luo, "Identification and learning control of ocean surface ship using neural networks," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 4, pp. 801–810, 2012.
- [42] M. Wang and A. Yang, "Dynamic learning from adaptive neural control of robot manipulators with prescribed performance," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 8, pp. 2244–2255, 2017.
- [43] C. Yang, K. Huang, H. Cheng, Y. Li, and C. Su, "Haptic identification by ELM-controlled uncertain manipulator," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 8, pp. 2398–2409, 2017.
- [44] S.-L. Dai, M. Wang, and C. Wang, "Neural learning control of marine surface vessels with guaranteed transient tracking performance," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 3, pp. 1717–1727, 2016.
- [45] R. M. Sanner and J. E. Slotine, "Gaussian networks for direct adaptive control," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 3, no. 6, pp. 837–863, 1992.
- [46] C. Wang and D. J. Hill, "Learning from neural control," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 17, no. 1, pp. 130–146, 2006.
- [47] H. K. Khalil, *Nonlinear Systems*, Prentice-Hall, Englewood Cliffs, NJ, USA, 2002.



# Hindawi

Submit your manuscripts at  
<https://www.hindawi.com>

