Neural Adaptive Backstepping Control of a Robotic Manipulator with Prescribed Performance Constraint

Qing Guo[†], *Member, IEEE*, Yi Zhang[†]*, *Member, IEEE*,
Branko G. Celler, *Fellow, IEEE*, Steven W. Su, *Senior Member, IEEE*

***Abstract*—This paper presents an adaptive neural network control of two-degree-of-freedom (Two-DOF) manipulator driven by electrohydraulic actuator. To restrict the system output in a prescribed performance constraint (PPC), a weighted performance function is designed to guarantee the dynamic and steady tracking errors of joint angle in a required accuracy. Then a Radial-Basis-Function neural network (RBFNN) is constructed to train the unknown model dynamics of manipulator by traditional backstepping control (TBC) and obtain the preliminary estimated model, which can replace the pre-known dynamics in the backstepping iteration. Furthermore, an adaptive estimation law is adopted to self-tune every trained-node weight and the estimated model is online optimized to enhance the robustness of the neural network controller. The effectiveness of the proposed control is verified by comparative simulation and experimental results with PID and TBC methods.**

***Index Terms*—Two-DOF manipulator, Adaptive neural network control, Prescribed performance constraint, Weighted performance function, Adaptive estimation law.**

I. Introduction

MANIPULATOR is a typical robotic plant that is widely used in industrial, aeronautics and astronautics engineering as it has a superior load-lifting capability and can replace the human operation in high risk environment. Generally, manipulator has two types of driven modes, i.e., full-actuated joints [1][2] and under-actuated joints [3]. The latter is more difficult than the former since both kinematic and dynamic constraints are integrated in the under-actuated motion system, which cannot directly adopt conventional Euler-Lagrange method. Thus, model complexity [4] and uncertainty degrade the controller robustness. To handle model constraint and system uncertainty, neural network control are popular to be used in discrete nonlinear systems [5], hypersonic flight vehicle [6], mobile manipulators [7] and fixed-base manipulators [8]. Karakaşoğlu *et al.* [9] originated a supervised learning scheme that employs a simple distributed updating rule for the online identification and decentralized adaptive control. Subsequently, Man *et al.* [10] proposed a robust adaptive sliding mode control with RBFNN for rigid robotic manipulators to achieve the robustness and asymptotic error convergence. Yang *et al.* [11] investigated the teleoperated robot systems and presented many novel control techniques such as RBFNN, wave variable, and variable gain control to eliminate the negative effects caused by communication delays and dynamics uncertainties existing in robot systems and human operators. Cao *et al.* [12] presented an approximate optimal control integrating with neural network to realize the path control of underactuated snake robots. Chen [13] proposed a robust adaptive control based on dynamic neural-fuzzy structure in a fixed-base manipulator to avoid the problems of overfitting and underfitting existed in the trained network. Subsequently, Wai *et al.* [14] presented a fuzzy-neural-network controller with backstepping and sliding-mode to improve the position tracking performance of a two-link robotic manipulator driven by DC servomotor. Patiño *et al.* [15] proposed a feedback adaptive neurocontroller for PUMA-560 robot which combines feedforward neural networks with adaptive and robust control techniques. The advantage of this neuro-controller is that the parameter adaptation is faster than that in the case where the learning capability of the full neural network (NN) is used for the adaptation task. Yue *et al.* [16] presented a NN with terminal sliding mode control used in wheeled mobile robots. In this reference, the uncertain ground friction model is identified according to the required performance. Then He *et al.* [17][18] proposed an adaptive NN control to estimate the unknown modelling uncertainty and environmental disturbance. Chen *et al.* [19] presented an adaptive neural control strategy for multiple input multiple output nonlinear systems to handle the nonsymmetric input nonlinearity and the constrained states. Meanwhile, Sun *et al.* [20] employed an adaptive neural network in flexible robotic manipulator to suppress vibrations. Yang *et al.* [21] adopted adaptive RBFNN control to improve the position tracking performance of the coupled motor drives system. Dutta *et al.* [22] presented a single-network adaptive critic-based controller for continuous-time systems with unknown dynamics in a policy iteration framework. This control algorithm is verified in a

Q. Guo is with the School of Aeronautics and Astronautics, University of Electronic Science and Technology of China, Chengdu, 611731, China, and with State Key Laboratory of Fluid Power & Mechatronic Systems, Zhejiang University, Hangzhou, 310027, China (e-mail: guoqinguestc@uestc.edu.cn).

Y. Zhang is with the School of Aeronautics and Astronautics, and with Center for Information in BioMedicine, and with Key Laboratory for NeuroInformation of Ministry of Education, School of Life Science and Technology, University of Electronic Science and Technology of China, Chengdu 611731, China. (e-mail: yi.zhang@uestc.edu.cn).

B. Celler is with the School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, NSW, Australia (e-mail: b.celler@unsw.edu.au).

S. Su is with Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, NSW, Australia (e-mail: Steven.Su@uts.edu.au).

This work was supported by National Natural Science Foundation of China (No. 51775089, No. 61305092, No. 51205045), the Fundamental Research Funds for the Central Universities, China (No. ZYGX2015J118, No. ZYGX2016J160), the Open Foundation of the State Key Laboratory of Fluid Power & Mechatronic Systems (No. GZKF-201515), and the China Postdoctoral Science Foundation funded project (2017M612950).

[†]Both of authors equally contribute to this work. *The corresponding author (e-mail: yi.zhang@uestc.edu.cn).

commercial robotic manipulator experiment.

In order to constrain state and output of nonlinear systems, Tee *et al.* [23][24] proposed Barrier Lyapunov Function (BLF) to restrict system states in prescribed constraint domain. Then Ren *et al.* [25] adopted BLF integrated with adaptive neural controller to improve the dynamic behavior of strict-feedback systems. Subsequently, He *et al.* [26] employed BLF to restrict the tracking error in arbitrary accuracy of robotic manipulator and improved the robustness of neural network controller. Meanwhile, he also adopted adaptive neural control and distributed parameter control in robotic manipulator [27]. The authors have successfully proposed novel adaptive NN controllers for the robots with constraints [17][27][28], where the stability of the closed loop system is also proved. Then Yang *et al.* [29] constructed a prescribed tracking performance requirement function integrated with BLF to guarantee both transient and steady-state tracking performance of the dual-arm robot. Zhang studied the multiloop integral controllability of multiple-input single-output system [30] to guarantee decentralized unconditional stability under control loop failure as well as to achieve offset-free tracking performance. Guo *et al.* [31] derived an observer bandwidth constraint to compromise between the dynamic performance and the maximal load capability of EHS. However, the input constraint is not considered in control design, which may result into larger control magnitude than traditional controller. In practice, perhaps the input saturation is a serious problem, which should be avoided in most physical systems. Thus, the designed controller has to compromise between the dynamic response performance and the input saturation, especially the initial dynamic response with large state error.

On the other hand, to guarantee the tracking error with convergence rate no less than a pre-specified value, Bechlioulis *et al.* [32] proposed an adaptive control with prescribed performance constraint to overcome the loss of controllability issue with input saturation. Subsequently, Zhang *et al.* [33] adopted prescribed performance constraint to restrict the angle-of-attack of hypersonic vehicle, which can be more easily applied to handle both static and time-varying constraints than the BLF-based methods. The advantage of prescribed performance constraint is that converts the original constrained system into an equivalent one without constraints by a weighted performance function. All the system states in the closed loop are uniformly ultimately bounded while the prescribed output constraints are held. Hence, the differences of two mentioned constraint holding techniques are summarized as follows: 1) The PPC is constructed by an adjustable weighted performance function [32], which is positive and monotonically decreasing, while the BLF is a clear logarithm function form [23]; 2) Both techniques are suitable for both time-invariant output/state constraint and the time-varying constraint [23][29]. The function derivation for time-varying constraint of PPC is more convenient than the logarithm derivation of BLF. Nevertheless, for time-invariant constraint, the difference of structural complexity can be neglected; 3) Considering the physical control saturation, the output constraint of PPC needs low control magnitude by the appropriate design of weighted performance function. 4) The initial output error of PPC can be allowed in a larger scope and the negative effect to the system stability is not relatively sensitive.

Thanks to the research development of motion control of NN based manipulator control [34][35], the study is supplied valuable intention. In this study, inspired by the adaptive neural network control proposed in [28] and the prescribed tracking performance function in [29], an adaptive neural network control is used in Two-DOF manipulator driven by electrohydraulic actuators. Different from these references, the electrohydraulic actutor model is considered in the robotic systems, which implies that the model order is increased from two to three. In this condition, model uncertainties caused by the mechanical structure are more obvious than that of without actuator model, which will decline the output performance of manipulator. Furthermore, to the best of the authors' knowledge, the robotic manipulator has not been driven by electrohydraulic actuator with prescribed performance constraint technique until now. Simultaneously, a RBFNN is adopted to train the unknown model dynamics emerged in backstepping iteration. Furthermore, considering the parametric uncertainty existed in manipulator model, an adaptive estimation law is designed to self-tune every trained-node weights of the RBFNN to enhance the proposed controller robustness. The comparison simulations and experimental results with the other two common controllers have verified the effectiveness of the proposed controller in terms of the tracking angle performance and the control current output by the servo valve of electrohydraulic actuator.

The remainder of this paper is organized as follows. The manipulator plant is described in section II. The adaptive neural network control is designed in section III including prescribed performance constraint, traditional backstepping control design, the model-training by RBFNN and the adaptive estimation law of node weights. The simulation and experimental results demonstrated on the joint motion of Two-DOF robotic manipulator are given in section IV and section V respectively. Finally, the conclusion is drawn in section VI.

II. Plant Description

A Two-DOF robotic manipulator is comprised by an upper arm, a forearm, a disc load and a fixed torso as shown as in Fig. 1. The shoulder and elbow joints can be driven to rotation by two electrohydraulic actuators (EHAs).

In the pressure loop of this EHA, the load pressure p_L of the hydraulic cylinder is controlled by the spool position x_v of servo valve. Since the cut-off frequency of servo valve is far greater than the control system bandwidth, the valve dynamics can be neglected in model construction [36] as follow:

$$x_v = K_{sv} u, \tag{1}$$

where K_{sv} and u are the gain and control current of servo valve respectively.

Then the load pressure p_L output by the hydraulic cylinder [37] is given by:

$$\frac{p_L}{x_v} = \frac{a_1 s + a_0}{s + b_0}, \tag{2}$$

where a_1, a_0, and b_0 are hydraulic model parameters.

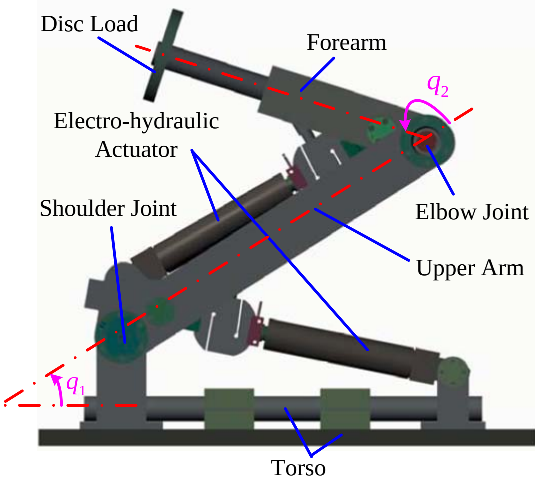

Fig. 1. The single-rod EHA control mechanism

Additionally, the driven joint torque [38] can be computed by:

$$\tau_i = p_{Li} A_p l_i(q_i), \quad i = 1, 2 \tag{3}$$

where $l_i(q_i)$ $(i = 1, 2)$ are two dynamic force arms, and A_p is the cylinder ram areas.

According to triangle geometry rule, the corresponding force arm $l_i(q_i)$ is computed by:

$$\begin{cases} c_i(q_i) = \sqrt{\kappa_i^2 + \varsigma_i^2 - 2\kappa_i\varsigma_i \cos(q_i + \epsilon_i)} \\ l_i(q_i) = \kappa_i \sin(\arccos((\kappa_i^2 + c_i^2 - \varsigma_i^2)/2\kappa_i c_i)) \end{cases}, \tag{4}$$

where $c_i(q_i)$ is the dynamic cylinder length, κ_i and ς are installing locations of two cylinders, and ϵ_i is the bias angle of cylinder for $i = 1, 2$.

Definition 1: For convenient expression, two operators '$\odot$' and '$\oslash$' are defined as follows:

$$\begin{aligned} \beta \odot \gamma &= [\beta_1, \beta_2, \ldots, \beta_n]^T \odot [\gamma_1, \gamma_2, \ldots, \gamma_n]^T \\ &= [\beta_1\gamma_1, \beta_2\gamma_2, \ldots, \beta_n\gamma_n]^T, \\ \beta \oslash \gamma &= [\beta_1, \beta_2, \ldots, \beta_n]^T \oslash [\gamma_1, \gamma_2, \ldots, \gamma_n]^T \\ &= [\beta_1/\gamma_1, \beta_2/\gamma_2, \ldots, \beta_n/\gamma_n]^T. \end{aligned}$$

The dynamic model of Two-DOF robotic manipulator [39] is described as follows:

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + f_v(q) \odot l(q) = \tau, \tag{5}$$

$$\begin{aligned} H(q) &= \begin{bmatrix} n_1 + n_2 + 2n_3 \cos q_2 & \star \\ n_2 + n_3 \cos q_2 & n_2 \end{bmatrix}, \\ C(q, \dot{q}) &= \begin{bmatrix} -n_3\dot{q}_2 \sin q_2 & -n_3(\dot{q}_1 + \dot{q}_2) \sin q_2 \\ n_3\dot{q}_1 \sin q_2 & 0 \end{bmatrix}, \\ G(q) &= \begin{bmatrix} n_4 g \cos q_1 + n_5 g \cos(q_1 + q_2) \\ n_5 g \cos(q_1 + q_2) \end{bmatrix}, \end{aligned} \tag{6}$$

$$\begin{cases} n_1 = m_1 L_{c2} + m_2 L_1 + m_f L_1^2 \\ n_2 = m_2 L_{c2}^2 + I_2 + m_f L_2^2 \\ n_3 = m_2 L_1 L_{c2} + m_f L_2^2 \\ n_4 = m_1 L_{c2} + m_2 L_1 + m_f L_1 \\ n_5 = m_2 L_{c2} + m_f L_2 \end{cases}, \tag{7}$$

where $q = [q_1, q_2]^T$, $\dot{q}$, $\ddot{q}$ are the angular position, velocity and acceleration, $\tau = [\tau_1, \tau_2]^T$ are the joint torque provided by two EHAs, $f_v(q) = \mu[\mathrm{sgn}(\dot{c}_1(q_1)), \mathrm{sgn}(\dot{c}_2(q_2))]^T$ is the viscous resistance, μ is the viscous coefficient of hydraulic oil, $H(q)$ is the symmetric positive definite inertia matrix, '$\star$' denotes the same transpose element of H, $C(q, \dot{q})\dot{q}$ is the Coriolis force element, $G(q)$ is the gravitational force, m_1 is the upper arm mass including cylinder, m_2 is the forearm mass, m_f is the disc load mass, L_i is the corresponding link length, L_{c1} is the distance from shoulder joint to the centre of mass of upper arm, L_{c2} is the distance from elbow joint to the centre of mass of forearm, I_i is the moment of inertia rotating respective centre of mass of i link, and g is the gravity constant.

From (1)-(5), if these states are defined as $x_1 = [x_{11}, x_{21}]^T = [q_1, q_2]^T$, $x_2 = [x_{12}, x_{22}]^T = [\dot{q}_1, \dot{q}_2]^T$, and $x_3 = [x_{13}, x_{23}]^T = [p_{L1}, p_{L2}]^T$, then the state space model of this manipulator system is given by:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = H^{-1}(x_1)(x_3 \odot l(x_1)A_p - f_v(x_1, x_2) \odot l(x_1) \\ \qquad - C(x_1, x_2)x_2 - G(x_1)) \\ \dot{x}_3 = v - b_0 x_3 \end{cases}, \tag{8}$$

where $v = K_{sv}(a_1\dot{u} + a_0 u)$ is considered to be the indirect control variable of this manipulator system.

Remark 1: Without loss of generality, some model parameters in (8) such as n_i $(i = 1, \ldots, 5)$ and μ are usually uncertain or inaccurate, which can be handled by NN learning in control design.

Property 1: [39] Since $H(q)$ is symmetric positive definite matrix, the matrix $\dot{H}(q) - 2C(q, \dot{q})$ is skew-symmetric.

Problem 1: Considering the dynamic model (8) of manipulator with parametric uncertainty mentioned in *Remark 1*, an adaptive neural network controller is designed to guarantee the output state x_1 tracking the demand input x_{1d}. Furthermore, the tracking error $x_1 - x_{1d}$ is restricted in a prescribed performance.

III. Control Design

The proposed controller is designed based on the traditional backstepping control method together with the output constraint holding technique and the node weights adaptive estimation law of RBFNN trained-model.

A. Prescribed Performance Constraint

Different from the Barrier Lyapunov function, the tracking error constraint is realized by a designed weighted performance function $\rho(t)$, which can guarantee the control variable u not beyond required saturation.

The two tracking errors of this manipulator are defined as:

$$e_i(t) = x_{i1}(t) - x_{i1d}(t), \quad i = 1, 2. \tag{9}$$

If x_{i1} is constrained in $x_{i1}(t) \in (x_{i1\min}, x_{i1\max})$, and x_{i1d} has a definite boundary $x_{i1d\min} \le x_{i1d} \le x_{i1d\max}$, then

$$\underline{k}_i < e_i(t) < \bar{k}_i \quad i = 1, 2, \tag{10}$$

where $\underline{k}_i = x_{i1\min} - x_{i1d\max} < 0$, and $\bar{k}_i = x_{i1\max} - x_{i1d\min} > 0$ are two constants.

Definition 2: [40] A smooth function $\rho(t) = (\rho_i(0) - \rho_i(\infty))e^{-\lambda t} + \rho_i(\infty)$ is called a weighted performance function if

(I) $\rho(t)$ is positive and monotonically decreasing;

(II) $\lim\limits_{x\to\infty} \rho(t) = \rho_\infty > 0$;

(III) $\rho(\infty) < \rho(0) < 1$.

Lemma 1: [32] If a weighted performance function $\rho(t)$ is designed such that

$$\underline{k}_i < e_i(t)/\rho_i(t) < \bar{k}_i, \tag{11}$$

then $e_i(t)$ is constrained in $(\underline{k}_i, \bar{k}_i)$.

In fact, if $e_i(t) \geq 0$, then $e_i(t) \leq e_i(t)/\rho_i(t) < \bar{k}_i$. On the other hand, if $e_i(t) < 0$, then $\underline{k}_i < e_i(t)/\rho_i(t) < e_i(t)$.

Thus from *Lemma 1*, the prescribed performance constraint $\underline{k}_i\rho_i(t) < e_i(t) < \bar{k}_i\rho_i(t)$ can be used to define the system state errors as follows:

$$z_{i1}(t) = T^{-1}(\frac{e_i(t)}{\rho_i(t)}) = \ln(\frac{\bar{k}_i(\underline{k}_i - e_i/\rho_i)}{\underline{k}_i(\bar{k}_i - e_i/\rho_i)}), \quad i = 1, 2 \tag{12}$$

where $T(\cdot)$ is a smooth function, $T^{-1}(\cdot)$ is its inverse function.

Theorem 1: [33] The smooth function $T(\cdot)$ is monotonically increasing, and holds the following properties:

$$\begin{aligned} \underline{k}_i < &T(z_{i1}) < \bar{k}_i & T(0) &= 0 \\ \lim_{z_{i1}\to-\infty} &T(z_{i1}) = \underline{k}_i & \lim_{z_{i1}\to+\infty} T(z_{i1}) &= \bar{k}_i \end{aligned} .$$

Proof. From (11), the inverse function of z_{i1} is described as:

$$T(z_{i1}) = \frac{e_i(t)}{\rho_i(t)} = \frac{\underline{k}_i\bar{k}_i(e^{z_{i1}} - 1)}{\underline{k}_i e^{z_{i1}} - \bar{k}_i}. \tag{13}$$

Then the derivative of $T(z_{i1})$ is given by:

$$\frac{dT}{dz_{i1}} = \frac{\underline{k}_i(\underline{k}_i - \bar{k}_i)e^{z_{i1}}}{\bar{k}_i(\frac{\underline{k}_i}{\bar{k}_i}e^{z_{i1}} - 1)^2}. \tag{14}$$

Since $\underline{k}_i < 0$ and $\bar{k}_i > 0$, $dT/dz_{i1} > 0$. Hence, $T(z_{i1})$ is monotonically increasing. Furthermore, due to the prescribed performance constraint $\underline{k}_i\rho_i(t) < e_i(t) < \bar{k}_i\rho_i(t)$, $\underline{k}_i < T(z_{i1}) < \bar{k}_i$ is held. As $z_{i1} \to \pm\infty$, $T(z_{i1})$ approaches its up and down boundary $\bar{k}_i$ and $\underline{k}_i$ respectively. If $z_{i1} = 0$ is substituted into (12), then $T(0) = 0$. ■

Remark 2: From *Theorem 1*, the prescribed performance constraint $\underline{k}_i\rho_i(t) < e_i(t) < \bar{k}_i\rho_i(t)$ of the tracking error $e_i(t)$ can be transformed into the system state error arbitrarily varying on the scale $-\infty < z_{i1}(t) < +\infty$.

B. Traditional Backstepping Controller

The other two state errors are defined as follows:

$$\begin{aligned} z_{i2} &= x_{i2} - \alpha_{i1} \\ z_{i3} &= x_{i3} - \alpha_{i2} \end{aligned}, \quad i = 1, 2 \tag{15}$$

where α_{ij} is the virtual control variable [41] emerged in backstepping control design.

For convenient derivation, many 2-dimensional vectors are defined as $z_j = [z_{1j}, z_{2j}]^T$ $(j = 1, 2, 3)$, $\alpha_j = [\alpha_{1j}, \alpha_{2j}]^T$ $(j = 1, 2)$, $e = [e_1, e_2]^T$, and $\rho = [\rho_1, \rho_2]^T$. Then the candidate Lyapunov functions of (8) are given by:

$$\begin{cases} V_1 = z_1^T z_1/2 \\ V_2 = V_1 + z_2^T H(x_1) z_2/2 \\ V_3 = V_2 + z_3^T z_3/2 \end{cases} . \tag{16}$$

Step 1: From (11), the derivative of z_1 is given by:

$$\dot{z}_1 = R(x_2 - \dot{x}_{1d} - e \odot \dot{\rho} \oslash \rho), \tag{17}$$

where $R = \text{diag}(r_1, r_2) \in \mathbb{R}^{2\times2}$,

$$r_i = \frac{\partial T^{-1}}{\partial(e_i/\rho_i)}\frac{1}{\rho_i} = \frac{\bar{k}_i - \underline{k}_i}{(\bar{k}_i - e_i/\rho_i)(e_i/\rho_i - \underline{k}_i)\rho_i}, \quad i = 1, 2.$$

If the virtual control α_1 is designed as:

$$\alpha_1 = \dot{x}_{1d} + e \odot \dot{\rho} \oslash \rho - R^{-1}C_1 z_1 \tag{18}$$

where $C_1 = \text{diag}(c_{11}, c_{21}) \in \mathbb{R}^{2\times2}$ is a positive definite constant matrix, then the derivative of V_1 in (16) is given by:

$$\dot{V}_1 = -z_1^T C_1 z_1 + z_1^T R z_2. \tag{19}$$

Step 2: Substituting *Property 1* into the derivative of V_2, we obtain that

$$\begin{aligned} \dot{V}_2 =& \dot{V}_1 + z_2^T H(x_1)\dot{z}_2 + z_2^T \dot{H}(x_1) z_2/2 \\ =& -z_1^T C_1 z_1 + z_2^T(R z_1 + z_3 \odot l A_p + \alpha_2 \odot l A_p \\ & - H\dot{\alpha}_1 - f_v \odot l - C\alpha_1 - G) \end{aligned} . \tag{20}$$

If the virtual control α_2 is designed as:

$$\alpha_2 = A_p^{-1}(-R z_1 + H\dot{\alpha}_1 + f_v \odot l + C\alpha_1 + G - C_2 z_2) \oslash l, \tag{21}$$

where $C_2 = \text{diag}(c_{12}, c_{22})$ is similar to C_1, then

$$\dot{V}_2 = -z_1^T C_1 z_1 - z_2^T C_2 z_2 + z_2^T z_3 \odot l A_p \tag{22}$$

Step 3: The derivative of V_3 is given by:

$$\begin{aligned} \dot{V}_3 =& \dot{V}_2 + z_3^T \dot{z}_3 \\ =& -z_1^T C_1 z_1 - z_2^T C_2 z_2 \\ & + z_3^T(A_p l \odot z_2 + v - b_0 x_3 - \dot{\alpha}_2) \end{aligned} . \tag{23}$$

If the final control variable v is designed as:

$$v = -C_3 z_3 - A_p l \odot z_2 + b_0 x_3 + \dot{\alpha}_2, \tag{24}$$

then

$$\dot{V}_3 \leq -z_1^T C_1 z_1 - z_2^T C_2 z_2 - z_3^T C_3 z_3 < 0. \tag{25}$$

Remark 3: From (16), (25), the control variable v (24) integrated with the virtual controls α_1 (18) and α_2 (21) cannot only guarantee all the system errors z_{ij} $(i = 1, 2,\ j = 1, 2, 3)$ asymptotic to zero but also restrict the dynamic errors e_1 and e_2 in the prescribed performance constraint $\underline{k}_i\rho_i(t) < e_i(t) < \bar{k}_i\rho_i(t)$.

C. Adaptive Neural Network Controller

According to *Remark 1*, some parametric uncertainties exist in the matrices H, C, G and f_v, which lead to negative effect in the virtual control α_2 (21) and the final control variable v (24). Thus, an adaptive neural network is adopted to handle these unknown dynamics.

Definition 3: [28] A class of Radial Basis Function Neural Network (RBFNN) is usually used to estimate an unknown continuous function $f_i(X)$ as follows:

$$f_i(X) = W_i^T S_i(X) + \varepsilon_i(X), \quad i = 1, 2 \tag{26}$$

where X is the input vector, W_i is weight vector with k_i nodes of RBFNN, $S_i(X) = [s_1, s_2, \ldots, s_{k_i}]^T$ is Gaussian basis function vector, and $\varepsilon_i(X)$ is the estimation error of RBFNN which is bounded by $|\varepsilon_i(X)| < \varepsilon_{\max}$ for $i = 1, 2$, and $\varepsilon_{\max}$ is an unknown boundary.

The radial basis element s_j of $S_i(X)$ is given by:

$$s_j(X) = \exp\left(\frac{-(X-\mu_j)^T(X-\mu_j)}{\sigma_j^2}\right), \, j = 1, \ldots, k_i, \tag{27}$$

where μ_j is the center of the receptive field, and σ_j the width of the Gaussian function [42].

If the input vector X of RBFNN is defined as $X = [x_1, x_2, \alpha_1, \dot{\alpha}_1]^T$, then the RBFNN estimation $W^T S(X)$ can be trained by sufficient sample data to approximate the following uncertain dynamic model:

$$\begin{aligned} W^T S(X) + \varepsilon(X) =& H(x_1)\dot{\alpha}_1 + C(x_1, x_2)\alpha_1 \\ &+ G(x_1) + f_v(x_1) \odot l(x_1) \\ =& F(X, \delta n_i, \delta\mu) \end{aligned}, \tag{28}$$

where F is the model uncertainty by model parameters n_i ($i = 1, \ldots, 5$) and μ mentioned in *Remark 1.*

Simultaneously, to enhance the robustness of the RBFNN controller, an adaptive estimation law for the weight vector W_i is designed as follows:

$$\dot{\hat{W}}_i = -\Gamma_i(S_i(X)z_{2i} + \eta_i \hat{W}_i), \quad i = 1, 2, \tag{29}$$

where $\hat{W}_i$ is the adaptive estimated value of the ideal weight W_i, $\Gamma_i > 0$ and $\eta_i > 0$ are diagonal matrix and constant gains of the adaptive estimation law.

Remark 4: The weight estimation $\hat{W}_i$ is online self-tuning by (29) based on a preliminary training value $W_i(0)$ of the designed RBFNN, which can eliminate the negative effect of parametric uncertainties existed in H, C, G and f_v.

If the adaptive weight estimation law (29) is considered in backstepping iteration, then the virtual control α_2 (21) is revised as follow:

$$\alpha_2 = A_p^{-1}(-Rz_1 + \sum_{i=1}^{2} \hat{W}_i^T S_i(X) - C_2 z_2) \oslash l. \tag{30}$$

Theorem 2: Consider the manipulator model (8), if the RBFNN controller is designed as (18), (24), (28) and (30), together with the adaptive weight estimation law (29), then the system state errors (12) and (15) are all ultimate boundary [43] and the error convergence domain is an hypersphere H_r, i.e.,

$$H_r \in \left\{ \begin{array}{c} z_1^T z_1 + z_2^T H z_2 + z_3^T z_3 + \sum\limits_{i=1}^{2} \tilde{W}_i^T \Gamma_i^{-1} \tilde{W}_i \\ = 2V_3(0)e^{-\lambda^* t_f} + 2\delta/\lambda^* \end{array} \right\}, \tag{31}$$

where λ^* and δ are positive constants, and $V_3(0)$ is the initial system state error, $\forall t > t_f$ (t_f is a finite time).

Proof. Consider the candidate Lyapunov functions of (8) as follows:

$$\left\{ \begin{array}{l} V_1 = z_1^T z_1/2 \\ V_2 = V_1 + z_2^T H(x_1) z_2/2 + \sum\limits_{i=1}^{2} \tilde{W}_i^T \Gamma_i^{-1} \tilde{W}_i/2 \\ V_3 = V_2 + z_3^T z_3/2 \end{array} \right., \tag{32}$$

where $\tilde{W}_i = W_i - \hat{W}_i$ is the self-tuning weight error of W_i.

Then the virtual control α_i ($i = 1, 2$) and the final control variable v can be also derived by the derivatives of V_i ($i = 1, 2, 3$).

Different from the backstepping iteration in section B, substituting (28) into (20), the derivative $\dot{V}_2$ is given by:

$$\begin{aligned} \dot{V}_2 =& \dot{V}_1 + z_2^T H(x_1)\dot{z}_2 + z_2^T \dot{H}(x_1) z_2/2 - \sum_{i=1}^{2} \tilde{W}_i^T \Gamma_i^{-1} \dot{\hat{W}}_i \\ =& - z_1^T C_1 z_1 + z_2^T (Rz_1 + z_3 \odot lA_p + \alpha_2 \odot lA_p \\ & - \sum_{i=1}^{2} \hat{W}_i^T S_i - \sum_{i=1}^{2} \tilde{W}_i^T S_i - \varepsilon) - \sum_{i=1}^{2} \tilde{W}_i^T \Gamma_i^{-1} \dot{\hat{W}}_i \end{aligned} \tag{33}$$

By Young's inequality, we can obtain:

$$\begin{array}{c} z_2^T \varepsilon \le (z_2^T z_2 + \|\varepsilon\|^2)/2 \\ \tilde{W}_i^T W_i \le (\tilde{W}_i^T \tilde{W}_i + W_i^T W_i)/2 \end{array}. \tag{34}$$

Substituting the revised virtual control α_2 (30), the adaptive weight estimation law (29), and the Young's inequalities (34) into (33), the derivative $\dot{V}_2$ is given by:

$$\begin{aligned} \dot{V}_2 \le& - z_1^T C_1 z_1 - z_2^T (C_2 - \frac{1}{2} I_{2\times 2}) z_2 - \sum_{i=1}^{2} \frac{\gamma_i}{2} \left\|\tilde{W}_i\right\|^2 \\ & + \frac{1}{2}\|\varepsilon\|^2 + \sum_{i=1}^{2} \frac{\gamma_i}{2} \|W_i\|^2 + z_2^T z_3 \odot lA_p \end{aligned} \tag{35}$$

Similar to *Step 3*, if the control variable v (24) is substituted into the derivative of V_3 in (32), then

$$\begin{aligned} \dot{V}_3 \le& - z_1^T C_1 z_1 - z_2^T (C_2 - \frac{1}{2} I_{2\times 2}) z_2 - z_3^T C_3 z_3 \\ & - \sum_{i=1}^{2} \frac{\gamma_i}{2} \left\|\tilde{W}_i\right\|^2 + \frac{1}{2}\|\varepsilon\|^2 + \sum_{i=1}^{2} \frac{\gamma_i}{2} \|W_i\|^2 \end{aligned} \tag{36}$$

If a positive constant λ^* is defined as:

$$\lambda^* = \min \left\{ \begin{array}{ll} 2\lambda_{\min}(C_1), & \dfrac{2\lambda_{\max}(C_2 - \frac{1}{2} I_{2\times 2})}{\lambda_{\min}(H)}, \\ 2\lambda_{\min}(C_3), & \min\limits_{i=1,2}(\dfrac{\sigma_i}{\lambda_{\max}(\Gamma_i^{-1})}) \end{array} \right\}, \tag{37}$$

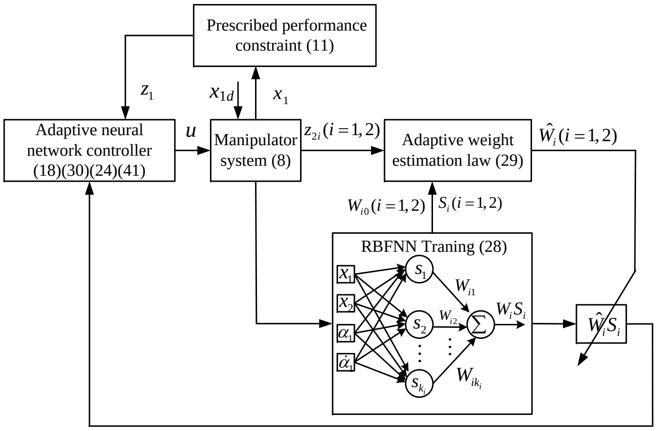

Fig. 2. The block diagram of the adaptive neural network control scheme

then (36) becomes:

$$\dot{V}_3 \leq -\lambda^* V_3 + \delta, \tag{38}$$

where the positive constant $\delta = \|\varepsilon\|^2/2 + \sum_{i=1}^{2} \sigma_i \|W_i\|^2/2$.

When pre-multiply and post-multiply the inequality (38) by $e^{\lambda^* t}$, the following equality holds:

$$\frac{d(V_3 e^{\lambda^* t})}{dt} = \delta e^{\lambda^* t}. \tag{39}$$

Integrating (39) and the following inequality holds:

$$V_3(t) \leq (V_3(0) - \frac{\delta}{\lambda^*})e^{-\lambda^* t} + \frac{\delta}{\lambda^*} \leq V_3(0)e^{-\lambda^* t} + \frac{\delta}{\lambda^*}. \tag{40}$$

Now substituting (32) into (40), and let $t > t_f$, the error convergence domain H_r in (31) is obtained. Furthermore, the size of H_r mainly depends on the element δ/λ^*. Thus, the increased control gains C_i $(i = 1, 2, 3)$ and the parametric estimation gains Γ_i $(i = 1, 2)$ can arbitrarily shrink the size of H_r as $t \to \infty$. ■

From (1), (2) and the indirect control variable v in (8), the actual control current u of servo valve is given by:

$$u = \frac{v}{K_{sv}(a_1 s + a_0)}. \tag{41}$$

Fig.2 shows the block diagram of the proposed adaptive neural network control scheme. The RBFNN (28) is trained by the system states x_1, x_2, the virtual control α_1 and its derivative $\dot{\alpha}_1$ in the traditional backstepping iteration from (18), (21), (24), which can obtain the preliminary weight value W_{i0} and the RBF $S_i(X)$ for $i = 1, 2$. Then the actual weights $\hat{W}_i$ $(i = 1, 2)$ are online self-tuning by the adaptive weight estimation law (29), which compensates the inaccurate dynamic model in (8). According to the prescribed performance constraint (11), the adaptive neural network controller u (41) is constructed by (18), (30) and (24) to guarantee the dynamic performance of the manipulator system (8).

IV. Simulation

To verify the proposed adaptive neural network controller (ANNC) in simulation, some nominal mechanical and hydraulic parameters of this robotic manipulator are shown in Tab. 1. The hydraulic parameters a_1, a_0 and b_0 are simplified from the linear load pressure model p_L controlled by the valve spool position x_v [44].

TABLE I
Some Mechanical and hydraulic parameters

Parameter	Value	Parameter	Value
m_1	6.012 kg	m_2	1.479 kg
m_f	1.068 kg	I_1	$0.118\ \text{kg}\cdot\text{m}^2$
I_2	$0.017\text{kg}\cdot\text{m}^2$	L_{c1}	0.177 m
L_{c2}	0.114 m	L_1	0.438 m
L_2	0.345 m	κ_i	0.403 m
ς_i	0.055 m	ϵ_1	6.1°
ϵ_2	13.8°	b_0	124
a_1	7	a_0	2.2×10^8
μ	500 s/m	A_p	$4.91 \times 10^{-4} m^2$
K_{sv}	0.125 mm/mA	g	$9.81\ \text{m}^2/\text{s}$

The motion ranges of two joint angles are $29.84° \leq q_1 \leq 115.76°$, and $47.85° \leq q_2 \leq 135.92°$. The prescribed performance boundaries of two tracking errors $\underline{k}_1 = \underline{k}_2 = -100°$ and $\bar{k}_1 = \bar{k}_2 = 100°$. Three parameters of the weighted performance function $\rho(0) = [0.95, 0.95]^T$, $\rho(\infty) = [0.02, 0.02]^T$ and $\lambda = 0.5$. The control gains are designed as $C_1 = \text{diag}\{1, 1\}$, $C_2 = \text{diag}\{10, 10\}$, and $C_3 = \text{diag}\{2000, 2000\}$. The approximate order of magnitude for these control gains is determined by the virtual controls (18), (21) and the final control (24). For the RBFNN estimation, 68 and 57 nodes are used for each $S_i(X)$ with centers selected in the area of [-1,1] with 8-dimensional grids. The variances of centers are $\sigma_j^2 = 1$, $(j = 1, \ldots, k_i)$. Two diagonal matrix gains of the adaptive estimation law are $\Gamma_1 = 20 \times I_{68}$, and $\Gamma_2 = 20 \times I_{57}$, where I_n denotes the $n \times n$ identity matrix, $\eta_1 = \eta_2 = 0.02$. These matrix gains are well-tuned considering both the fast convergent weights and the desirable tracking performance.

To illustrate the problem, the proposed adaptive neural network controller based on (18), (30), (24) and (41) is compared with the following two controllers:

1) Proportional-integral-derivative (PID) controller $u = k_p(y_d - x_1) + k_i \int (y_d - x_1)dt + k_d(\dot{y}_d - x_2)$, where the control gains k_p = 140, k_i = 17 and k_d = 7 have been well tuned to guarantee fast responses of two robotic joint angles.

2) The traditional backstepping controller (TBC) based on (18), (21), (24) and (41).

A. Comparison with PID control

Two sinusoidal demands of the joint angles are $q_{1d} = 33\sin(1\pi t) + 72.8°$ and $q_{2d} = 34\sin(2\pi t) + 91.9°$. The comparison results of two controllers are shown in Figs. 3-5. As the time $t < 5$ s, the dynamic tracking performances Δq_i $(i = 1, 2)$ of PID are better than ANNC. However, the two control currents u_i $(i = 1, 2)$ of PID controller are more consumed than ANNC, which surpass 100 mA near the initial zero time. Since the weighted performance function $\rho(t)$ is exponential attenuated, the dynamic tracking error of ANNC is restricted in $\underline{k}_i\rho_i(t) < e_i(t) < \bar{k}_i\rho_i(t)$ mentioned in (11). As two joint angles approach their steady states, the steady state errors of ANNC $|e_i| < 2°$ (i.e., $\underline{k}_i\rho_i(\infty) < e_i < \bar{k}_i\rho_i(\infty)$),

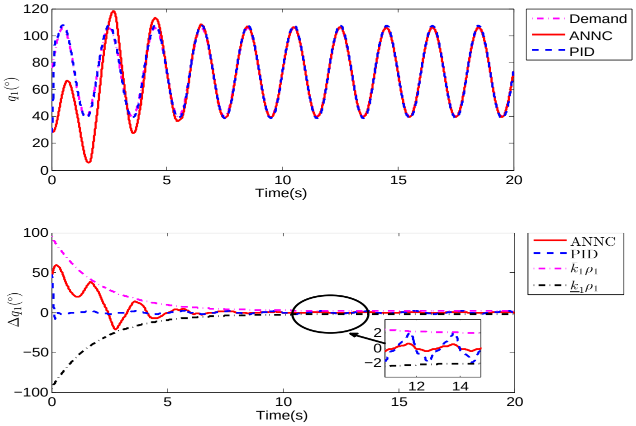

Fig. 3. The comparison results with PID controller, q_1 is the shoulder joint angle, $\Delta q_1 = q_{1d} - q_1$ is the tracking angle error.

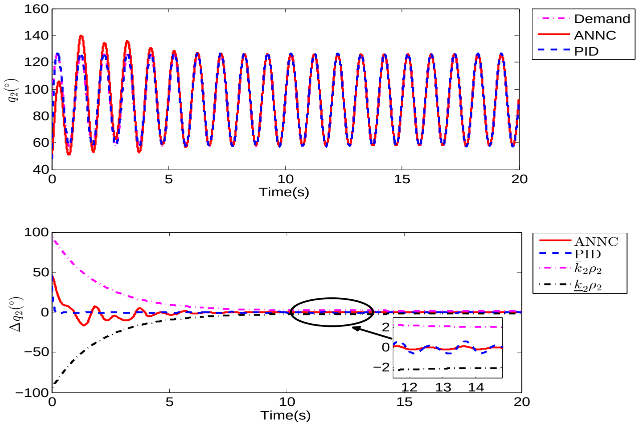

Fig. 4. The comparison results with PID controller, q_2 is the elbow joint angle, $\Delta q_2 = q_{2d} - q_2$ is the tracking angle error.

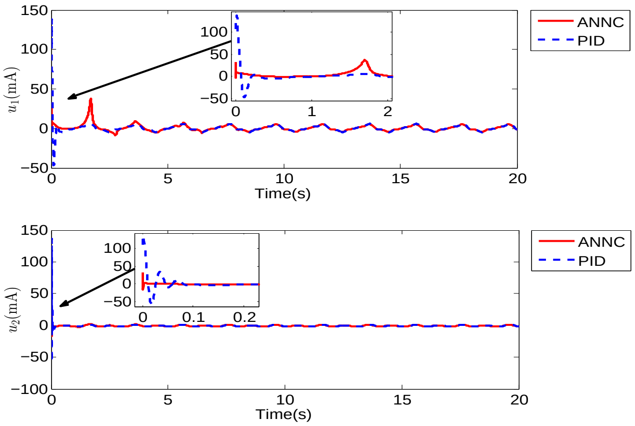

Fig. 5. The comparison results with PID controller, u_1 and u_2 are the control currents of two joint electro-hydraulic actuators.

which has better performances than PID as shown in Fig. 5. Certainly, the steady tracking error of PID can be further reduced by increasing PID gains. However, the system stability margin may be degraded.

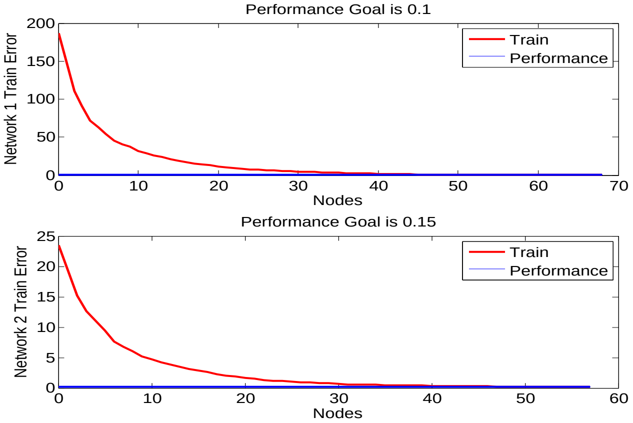

Fig. 6. The train results of two neural networks by (28).

B. *Comparison with TBC*

The two joint angle demands are the same to Section A. To realize the ANNC algorithm, the model (28) are trained by two RBFNNs $W_i^T S_i(X)$ $(i = 1, 2)$. If two performance goals are selected as 0.1 and 0.15, the model trained errors are asymptotic convergence by the selected 68 and 57 network nodes respectively as shown in Fig. 6. Then the well-trained weights of two RBFNNs are obtained and used as the initial value of the adaptive weight estimation law (29). Figs. 7-10 show the comparison results of ANNC with TBC. Since the prescribed performance constraint is both considered by two controllers, the dynamic and steady tracking errors of ANNC is similar to TBC, which indicates the favorable optimal performance of the model (28) by the RBFNN. After 5 s, the tracking errors of two joint angles are restricted in $1°$. However, the control current magnitude of TBC is still larger than ANNC, which surpasses 20 mA as shown in Fig. 10. Thus, the prescribed performance constraint (11) consumes obvious control capability by TBC without adaptive weight estimation law. The model estimation results by adaptive neural network (28) and (29) are shown in Fig. 9. The model estimation error about $H\dot{\alpha}_1 + C\alpha_1 + G + f_v \odot l$ includes two elements, i.e., the RBFNN estimation error $\varepsilon_i(X)$ and the self-tuning weight error $\tilde{W}_i$. From Fig. 9, the model estimation error can be constrained in a satisfactory neighbourhood by ANNC.

C. *Comparative results with Parametric Uncertainty*

To verify the robustness of ANNC, two sinusoidal demands of the joint angles are chosen as $q_{1d} = 33\sin(1.3\pi t) + 72.8°$ and $q_{2d} = 34\sin(1.5\pi t) + 91.9°$. Some parametric uncertainties are assumed as $m_1' = 1.5m_1$, $m_2' = 1.5m_2$, $m_f' = 1.5m_f$, $L_1' = 1.3L_1$, $L_2' = 0.7L_2$, $I_1' = 0.7I_1$, $I_2' = 1.3I_2$, and $\mu' = 1.5\mu$. The control current saturation is $u_{\max} = \pm 20$ mA. Then the comparison results are shown in Figs. 11-13. The dynamic and steady tracking errors can be also restricted in the prescribed performance constraint (11) by TBC and ANNC. However, the control saturation emerges in the shoulder joint motion by TBC as shown in Fig. 13. Two control currents of TBC u_1 and u_2 are larger than that of ANNC. Due to the

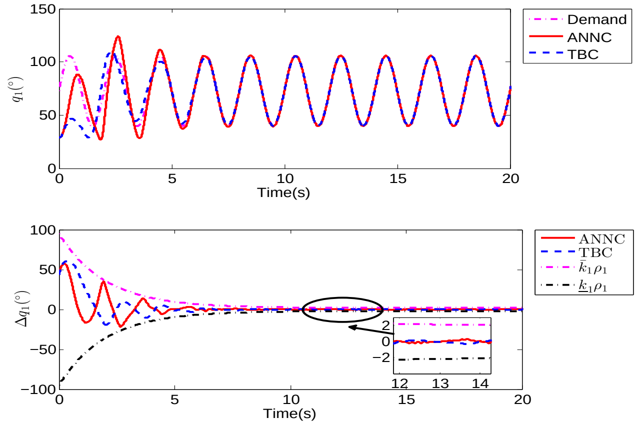

Fig. 7. The comparison results with traditional backstepping controller, q_1 is the shoulder joint angle, $\Delta q_1 = q_{1d} - q_1$ is the tracking angle error.

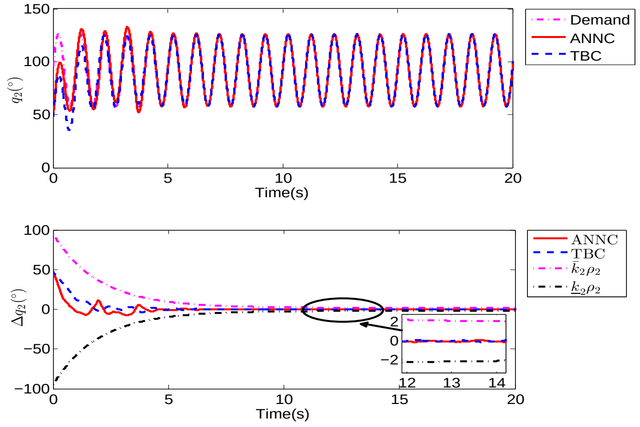

Fig. 8. The comparison results with traditional backstepping controller, q_2 is the elbow joint angle, $\Delta q_2 = q_{2d} - q_2$ is the tracking angle error.

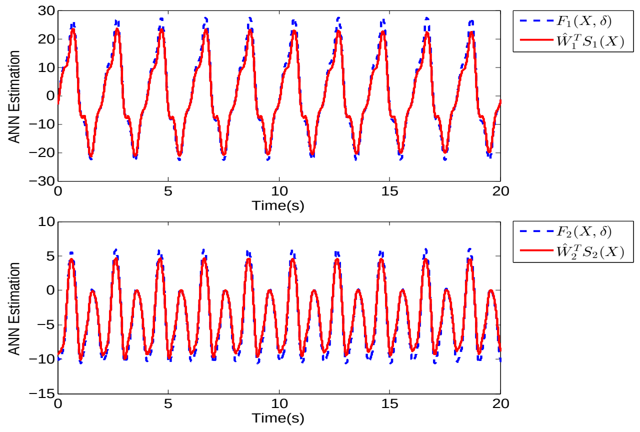

Fig. 9. The ANN estimation for model uncertainty, where $F_i(X, \delta)$ is the model uncertainty and $\hat{W}_i^T S_i$ is the corresponding estimation of ANN.

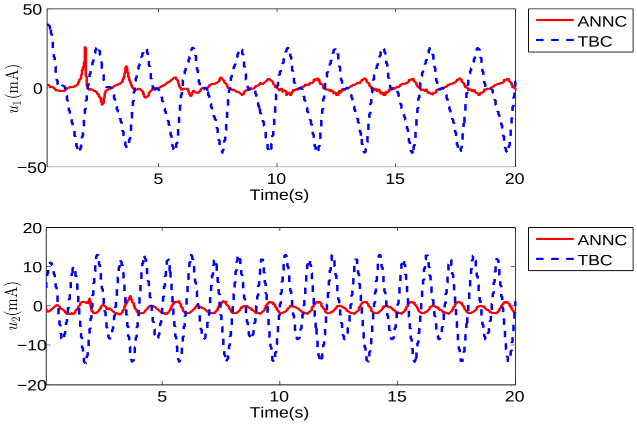

Fig. 10. The comparison results with traditional backstepping controller, u_1 and u_2 are the control currents of two joint electro-hydraulic actuators.

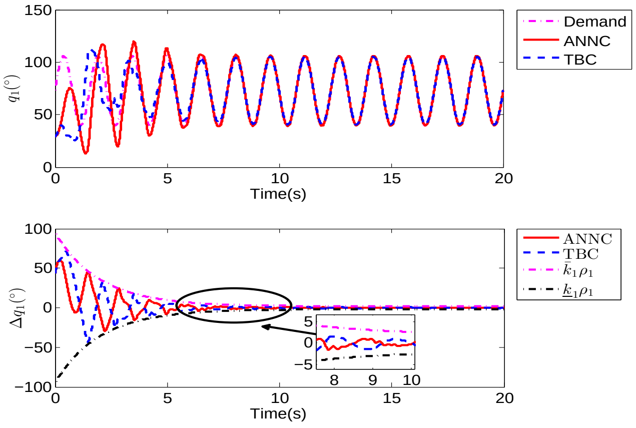

Fig. 11. The comparison results with traditional backstepping controller, q_1 is the shoulder joint angle, $\Delta q_1 = q_{1d} - q_1$ is the tracking angle error.

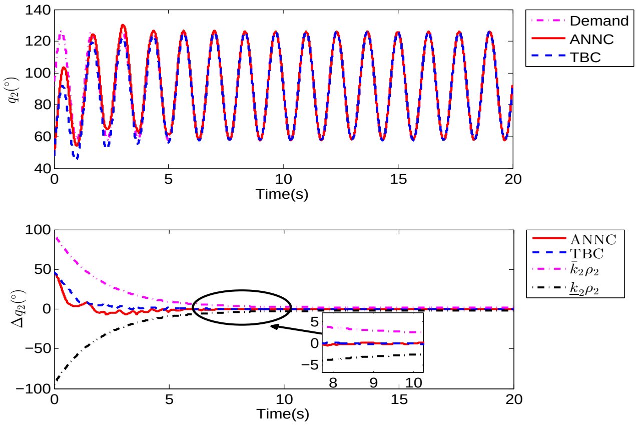

Fig. 12. The comparison results with traditional backstepping controller, q_2 is the elbow joint angle, $\Delta q_2 = q_{2d} - q_2$ is the tracking angle error.

parametric uncertainties injected in the manipulator system (8), TBC consumed obvious control cost to compensate the model error and to guarantee two tracking angle errors in the prescribed performance constraint. However, it is necessary for ANNC to adopt the adaptive NN estimation model (30) rather than the uncertain model (21) by TBC. Even though there exists modelling uncertainties, the dynamic and steady tracking errors of two joint angles can be still restricted in the prescribed performance constraint ($\underline{k}_i\rho_i(t) < e_i(t) < \bar{k}_i\rho_i(t)$). Furthermore, the control currents output by two servo valves of EHAs are satisfactory as the adaptive weight estimation law is adopted to self-tune every node weight as shown in Fig. 14.

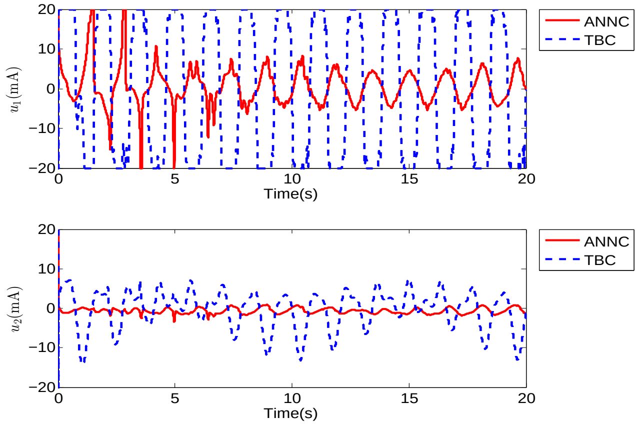

Fig. 13. The comparison results with traditional backstepping controller, u_1 and u_2 are the control currents of two joint electro-hydraulic actuators.

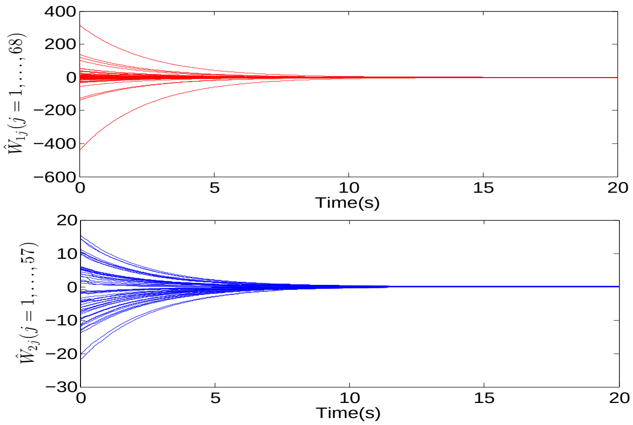

Fig. 14. The weight estimations by (29), $\hat{W}_{1j}$ $(j = 1, \ldots, 68)$ and $\hat{W}_{2j}$ $(j = 1, \ldots, 57)$ are respective estimation of two numbers of nodes.

V. Experiment

The experimental bench of Two-DOF robotic manipulator is set up as shown in Fig. 15 to verify the effectiveness of the proposed ANNC. Two electrohydraulic actuators are composed of two servo valves (FF-102/03021T240), two cylinders (UG1511R25/16-80), a pump station (HY-36CC-01/11kw), and two accumulators (NXQ1-L1.0/31.5H). Four cylinder pressures are measured by the pressure transducer (BD-Sensors-DMP-331). Two joint angles are measured by the incremental encoder (HENGSILL ALN722R7LSDN13188).

The control implementation of this experiment is shown in Fig. 16. The encoder and the pressure transducer data are sampled by NI card as the feedback information. The control algorithm is realized by Matlab/Simulink tool in the host PC computer and the control demand is returned to NI card, which drives the servo valve to regulate the cylinder pressure supplied by the pump station. The interval of the whole algorithm execution is 5 ms. Due to the variable load pressure, the joint motion control of the robotic manipulator is driven by this electrohydraulic actuator. The screen display shows the measurement and computed signals in real-time.

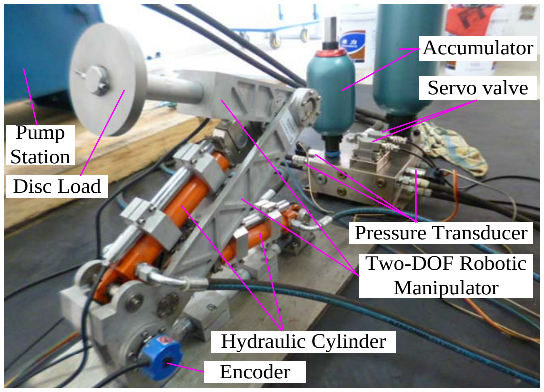

Fig. 15. The experimental bench of the robotic manipulator driven by electrohydraulic actuator.

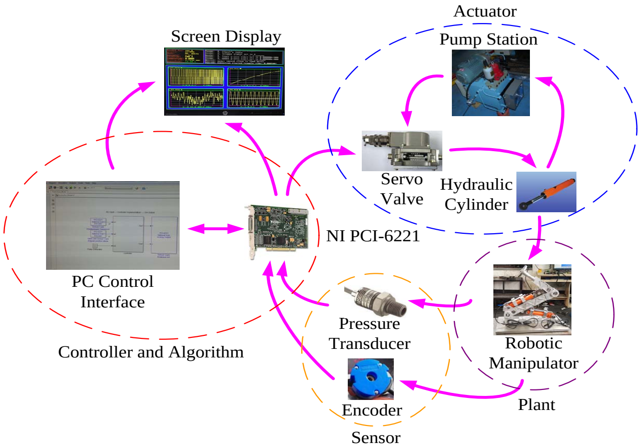

Fig. 16. The control implementation diagram of the robotic manipulator

A. Experimental Results of ANNC

Two joint angle demands are chosen as $q_{1d} = 33\sin(0.5\pi t) + 72.8^\circ$ and $q_{2d} = 34\sin(\pi t) + 91.9^\circ$. The experimental results of ANNC are shown as Figs. 17-20. The dynamic and steady tracking errors of two joint angles are restricted in the prescribed performance constraint $(\underline{k}_i\rho_i(t) < e_i(t) < \bar{k}_i\rho_i(t))$. The steady tracking errors Δq_i $(i = 1, 2)$ are less than 4° as shown in Fig. 17-18. From Fig. 20, since the load pressure of shoulder electrohydraulic actuator p_{L1} is larger than that of elbow actuator p_{L2}, the dynamic response of shoulder angle is slower than the elbow angle. Thus, two hydraulic accumulators are used to store energy and improve the flow velocity in the hydraulic control loop. However, the duration of energy storage integrated with pressure preparation is approximate 2-5 seconds as shown in Fig. 20, which results in the control current saturation ± 20 mA in initial response time of ANNC as shown in Fig. 19. After 10s, two joint angles approach the steady state and the control current magnitudes of two servo valves are periodically regulated to guarantee two joint angles q_i $(i = 1, 2)$ track the corresponding demands q_{id} $(i = 1, 2)$. Meanwhile, the cylinder chamber pressures p_{ai} and p_{bi} $(i = 1, 2)$ of two electrohydraulic actuators are less than 50 bar, which are constrained by the supply pressure p_s of the pump station.

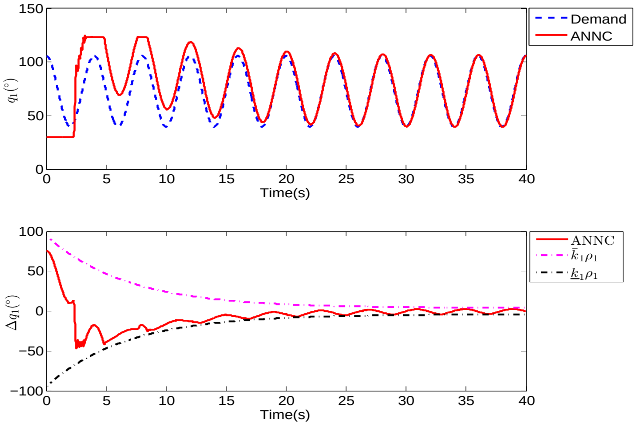

Fig. 17. The experimental results of ANNC, q_1 is the shoulder joint angle, $\Delta q_1 = q_{1d} - q_1$ is the tracking angle error.

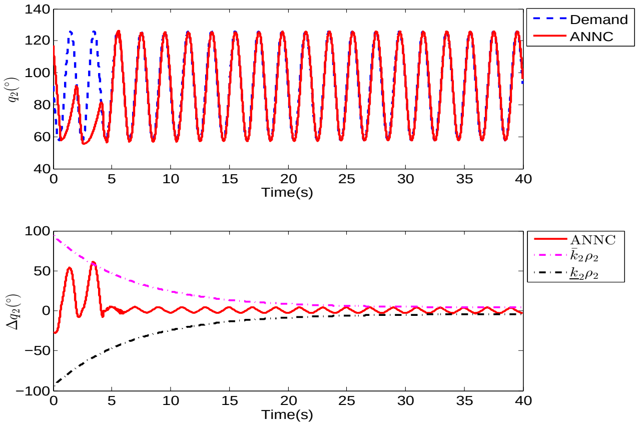

Fig. 18. The experimental results of ANNC, q_2 is the elbow joint angle, $\Delta q_2 = q_{2d} - q_2$ is the tracking angle error.

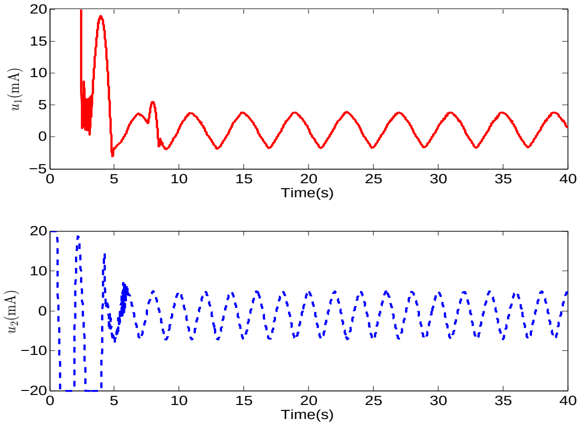

Fig. 19. The experimental results of ANNC, u_1 and u_2 are the control currents of two joint electrohydraulic actuators.

B. Comparison Results

Then two joint demands are chosen as $q_{1d} = 33\sin(0.8\pi t) + 72.8°$ and $q_{2d} = 34\sin(\pi t) + 91.9°$. The experimental results of three controllers are shown as Figs. 21-24. Although the steady tracking errors of PID are less than the other two controllers, some angle chatters emerges in two joint responses as shown in Figs. 21-22. Of course, the control gains of PID can be reduced to eliminate these angle chatters, which may degrade the dynamic and steady performance. Two steady errors of ANNC are less than 4°, which has more favorable performances than TBC. From Fig. 23, the control current of elbow actuator approaches the control saturation since the elbow motion frequency is larger than the shoulder. Furthermore, some control chatters also emerges in PID rather than the other two controllers. The model estimation results $\hat{W}_i^T S_i(X)$ $(i = 1, 2)$ by the ANN (28) and (29) are shown in Fig. 24, which indicates the robustness of ANNC when two joint demands vary with different motion frequency.

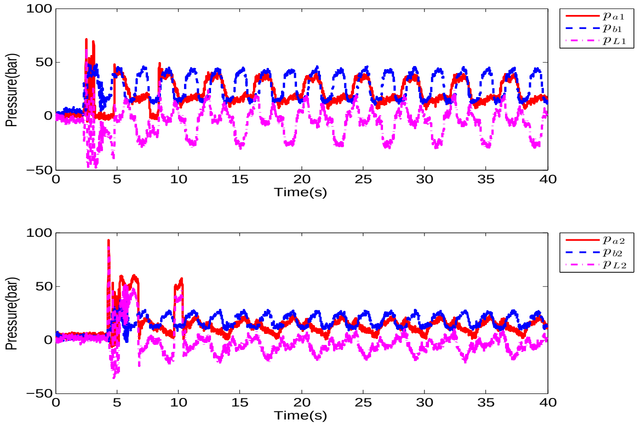

Fig. 20. The experimental results of ANNC, p_{ai} and p_{bi} are two cylinder chamber pressures, $p_{Li} = p_{ai} - p_{bi}$ is the load pressure of the corresponding electrohydraulic actuator for $i = 1, 2$.

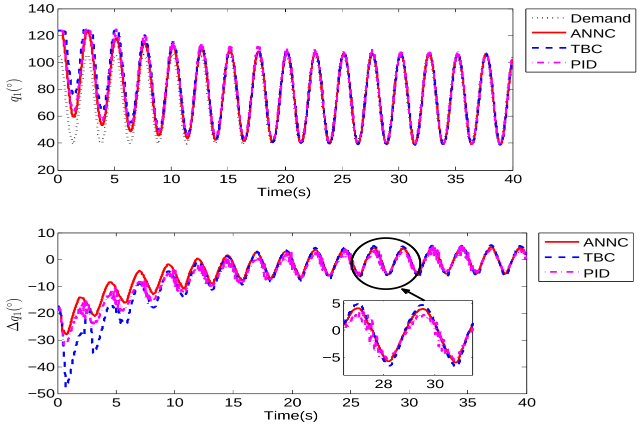

Fig. 21. The experimental results of three controllers, q_1 is the shoulder joint angle, $\Delta q_1 = q_{1d} - q_1$ is the tracking angle error.

VI. Conclusion

In this study, an adaptive neural network control with backstepping was proposed for Two-DOF manipulator driven by the electrohydraulic actuator. Considering the output constrained problem, a weighted performance function was designed to restrict the tracking angle errors of two joints in a prescribed performance constraint. To avoid the unknown dynamics in model-based control design, a RBF neural network was constructed to train the unknown model dynamics.

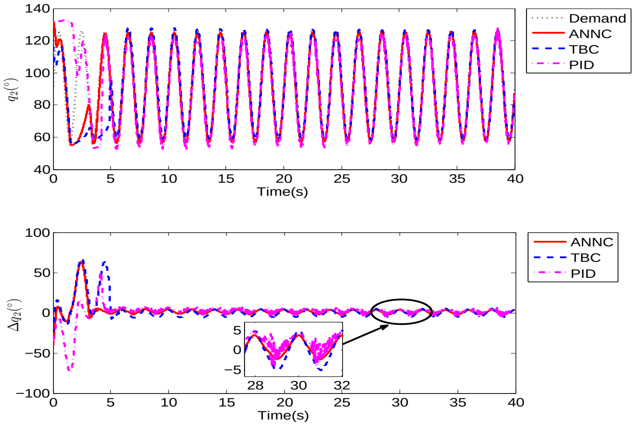

Fig. 22. The experimental results of three controllers, q_2 is the elbow joint angle, $\Delta q_2 = q_{2d} - q_2$ is the tracking angle error.

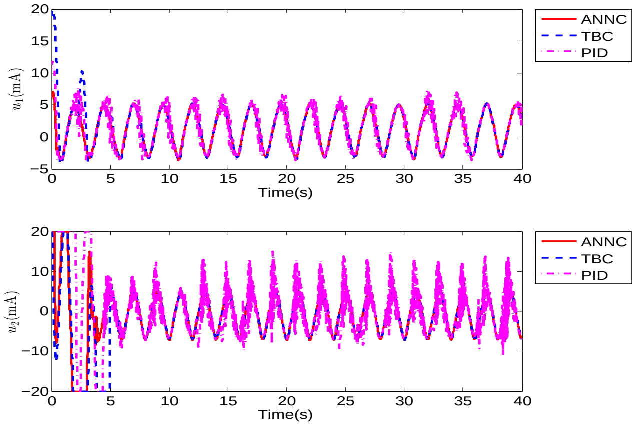

Fig. 23. The experimental results of three controllers, u_1 and u_2 are the control currents of two joint electro-hydraulic actuators.

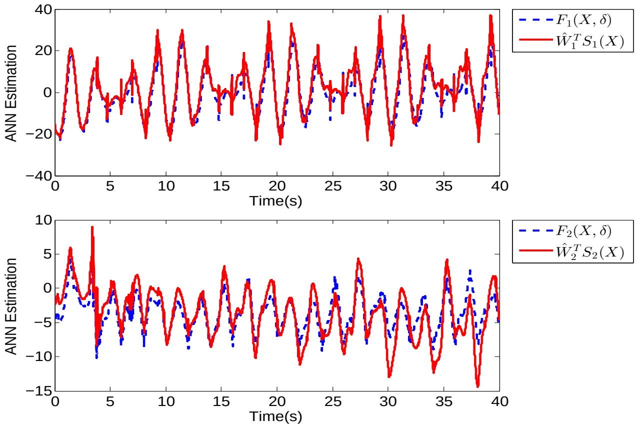

Fig. 24. The ANN estimation for model uncertainty, where $F_i(X,\delta)$ is the model uncertainty and $\hat{W}_i^T S_i$ is the corresponding estimation of ANN.

Although the network estimated model depended on different training samples, the node weights of RBFNN can be self-tuned by an adaptive estimation law according to the system state errors. The comparison results with PID and traditional backstepping controller indicated that the proposed ANNC had three advantages as follows: 1) the unknown dynamic model parameters need not to be pre-known. By the RBFNN learning, the complicated nonlinear model can be well identified. 2) the control sensitivity will be relaxed and the robustness is improvement when some model dynamics and parametric uncertainty exited in the robotic manipulator. 3) the controller based on RBFNN will not easy to emerge chatters when the demand frequency and magnitude increased and guarantee the desirable tracking performance by PPC, which indicates the adaptation capability of RBFNN to address uncertain parameters and disturbance. To be honest, due to the control saturation of the hydraulic actuator and the load pressure delay caused by hydraulic pipe transmission, the control gains cannot be big enough and the experimental dynamic response of the proposed controller is certainly lower than the corresponding simulation results. To further improve the performance of this manipulator, the controller should consider the time-delay model of electro-hydraulic actuator and the hydraulic elements configuration may be optimized in the future.

References

[1] Z. Li, S. S. Ge, M. Adams, and W. S. Wijesoma, "Adaptive robus output-feedback motion/force control of electrically driven nonholonomic mobile manipulators," *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 6, pp. 1308–1315, Nov 2008.

[2] W. Dong, "On trajectory and force tracking control of constrained mobile manipulators with parameter uncertainty," *Automatica*, vol. 38, no. 9, pp. 1475–1484, Sep 2002.

[3] Z. Li, C. Yang, C. Y. Su, and W. Ye, "Adaptive fuzzy-based motion generation and control of mobile under-actuated manipulators," *Eng. Appl. Artif. Intel.*, vol. 30, pp. 86–95, Apr 2014.

[4] Y. Yan, J. Xu, and M. Wiercigroch, "Basins of attraction of the bistable region of time-delayed cutting dynamics," *Phys. Rev. E*, vol. 96, no. 3, Sep 2017.

[5] Y. Liu, L. Tang, S. Tong, and C. L. P. Chen, "Adaptive nn controller design for a class of nonlinear mimo discrete-time systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 5, pp. 1007–1018, May 2015.

[6] B. Xu, C. Yang, and Z. Shi, "Reinforcement learning output feedback nn control using deterministic learning technique," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 635–641, Mar 2014.

[7] S. Lin and A. A. Goldenberg, "Neural-network control of mobile manipulators," *IEEE Trans. Neural Netw.*, vol. 12, no. 5, pp. 1121–1133, May 2001.

[8] L. Wang, T. Chai, and C. Yang, "Neural-network-based contouring control for robotic manipulators in operational space," *IEEE Trans. Control Syst. Technol.*, vol. 20, no. 4, pp. 1073–1080, Jul 2012.

[9] A. Karakaşoğlu, S. I. Sudharsanan, and M. K. Sundareshan, "Identification and decentralized adaptive control using dynamical neural networks with application to robotic manipulators," *IEEE Trans. Neural Netw.*, vol. 4, no. 6, pp. 919–930, Nov 1993.

[10] Z. Man, X. H. Yut, K. Eshraghian, and M. Pdaniswamiz, "A robust adaptive sliding mode tracking control using an rbf neural network for robotic manipulators," in *Proc. IEEE International Conference on Neural Networks*, 1995, pp. 2403–2408.

[11] C. Yang, X. Wang, Z. Li, Y. Li, and C. Y. Su, "Teleoperation control based on combination of wave variable and neural networks," *IEEE Trans. Syst., Man, Cybern. A, Syst.*, vol. 47, no. 8, pp. 2125–2136, Aug 2017.

[12] Z. Cao, Q. Xiao, R. Huang, and M. Zhou, "Robust neuro-optimal control of underactuated snake robots with experience replayn," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 208–217, Jan 2018.

[13] C.-S. Chen, "Dynamic structure neural-fuzzy networks for robust adaptive control of robot manipulators," *IEEE Trans. Ind. Electron.*, vol. 55, no. 9, pp. 3402–3414, Sep 2008.

[14] R. J. Wai and R. Muthusamy, "Fuzzy-neural-network inherited sliding-mode control for robot manipulator including actuator dynamics," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 2, pp. 274–287, Feb 2013.

[15] H. D. Patiño, R. Carelli, and B. R. Kuchen, "Neural networks for advanced control of robot manipulators," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 343–354, Mar 2002.

[16] M. Yue, L. Wang, and T. Ma, “Neural network based terminal sliding mode control for wmrs affected by an augmented ground friction with slippage effect,” *IEEE/CAA J. Automat. Sinica*, vol. 4, no. 3, pp. 498–506, Jul 2017.

[17] W. He, A. O. David, Z. Yin, and C. Sun, “Neural network control of a robotic manipulator with input deadzone and output constraint,” *IEEE Trans. Syst., Man, Cybern. A, Syst.*, vol. 46, no. 6, pp. 759–770, Jun 2016.

[18] W. He, Z. Yan, C. Sun, and Y. Chen, “Adaptive neural network control of a flapping wing micro aerial vehicle with disturbance observer,” *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3452–3465, Oct 2017.

[19] Z. Chen, Z. Li, and C. L. P. Chen, “Adaptive neural control of uncertain mimo nonlinear systems with state and input constraints,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 6, pp. 1318–1330, Jun 2017.

[20] C. Sun, W. He, and J. Hong, “Neural network control of a flexible robotic manipulator using the lumped spring-mass model,” *IEEE Trans. Syst., Man, Cybern. A, Syst.*, vol. 47, no. 8, pp. 1863–1874, Aug 2017.

[21] H. Yang and J. Liu, “An adaptive rbf neural network control method for a class of nonlinear systems,” *IEEE/CAA J. Automat. Sinica*, vol. 5, no. 2, pp. 457–462, Mar 2018.

[22] S. Dutta, P. K. Patchaikani, and L. Behera, “Near-optimal controller for nonlinear continuous-time systems with unknown dynamics using policy iteration,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 7, pp. 1537–1549, Jul 2016.

[23] K. P. Tee, S. S. Ge, and E. H. Tay, “Barrier lyapunov functions for the control of output-constrained nonlinear systems,” *Automatica*, vol. 45, no. 1, pp. 918–927, Apr 2009.

[24] K. P. Tee, B. Ren, and S. S. Ge, “Control of nonlinear systems with time varying output constraints,” *Automatica*, vol. 47, no. 11, pp. 2511–2516, Nov 2011.

[25] B. Ren, S. S. Ge, K. P. Tee, and T. H. Lee, “Adaptive neural control for output feedback nonlinear systems using a barrier lyapunov function,” *IEEE Trans. Neural Netw.*, vol. 21, no. 8, pp. 1339–1345, Aug 2010.

[26] W. He, H. Huang, and S. S. Ge, “Adaptive neural network control of a robotic manipulator with time-varying output constraints,” *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3136–3147, Oct 2017.

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

[28] W. He, Y. Chen, and Z. Yin, “Adaptive neural network control of an uncertain robot with full-state constraints,” *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 620–629, Mar 2016.

[29] C. Yang, Y. Jiang, Z. Li, W. He, and C. Y. Su, “Neural control of bimanual robots with guaranteed global stability and motion precision,” *IEEE Trans. Ind. Informat.*, vol. 13, no. 3, pp. 1162–1171, Jun 2016.

[30] Y. Zhang, S. Su, A. Savkin, B. Celler, and N. Hung, “Multiloop integral controllability analysis for nonlinear multiple-input single-output processes,” *Ind. Eng. Chem. Res.*, vol. 56, no. 28, pp. 8054–8065, Jul 2017.

[31] Q. Guo, Y. Zhang, B. G. Celler, and S. W. Su, “Backstepping control of electro-hydraulic system based on extended-state-observer with plant dynamics largely unknown,” *IEEE Trans. Ind. Electron.*, vol. 63, no. 11, pp. 6909–6920, Nov 2016.

[32] C. P. Bechlioulis and G. A. Rovithakis, “Adaptive control with guaranteed transient and steady state tracking error bounds for strict feedback systems,” *Automatica*, vol. 45, no. 2, pp. 532–538, Feb 2009.

[33] Z. Zhang, G. Duan, and M. Hou, “Longitudinal attitude control of a hypersonic vehicle with angle of attack constraints,” in *Proc. 10th Asian Control Conference*, 2015, pp. 1–6.

[34] M. Tokita, T. Mituoka, T. Fukuda, T. Shibata, and F. Arai, “Position and force hybrid control of robotic manipulator by neural network (adaptive control of 2 dof manipulators),” in *Proc. IEEE International Joint Conference on Neural Networks*, 1991, pp. 113–121.

[35] Y. Jiang, C. Yang, J. Na, G. Li, Y. Li, and J. Zhong, “A brief review of neural networks based learning and control and their applications for robots,” *Complexity*, vol. 2017, no. 4, pp. 1–14, Oct 2017.

[36] W. Kim, D. Shin, D. Won, and C. C. Chung, “Disturbance-observer-based position tracking controller in the presence of biased sinusoidal disturbance for electrohydraulic actuators,” *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 6, pp. 2290–2298, Nov 2013.

[37] H. Merritt, *Hydraulic control systems*. USA: John Wiley & Sons, New York, NY, 1967.

[38] Q. Guo, T. Yu, and D. Jiang, “Robust H$_\infty$ positional control of 2-dof robotic arm driven by electro-hydraulic servo system,” *ISA Trans.*, vol. 59, pp. 55–64, Nov 2015.

[39] S. S. Ge, T. H. Lee, and C. J. Harris, *Adaptive Neural Network Control of Robotic Manipulators*. Singapore: World Scientific Publishing, 1998.

[40] Z. Zhang, G. Duan, and M. Hou, “Robust adaptive dynamic surface control of uncertain non-linear systems with output constraints,” *IET Control Theory & Appl.*, vol. 11, no. 1, pp. 110–121, Jan 2017.

[41] M. Krstic, I. Kanellakopoulos, and P. V. Kokotovic, *Nonlinear and Adaptive Control Design*. USA: John Wiley & Sons, New York, NY, 1995.

[42] R. M. Sanner and J. E. Slotine, “Gaussian networks for direct adaptive control,” *IEEE Trans. Neural. Netw.*, vol. 3, no. 6, pp. 837–863, Nov 1992.

[43] H. K. Khalil, *Nonlinear systems*, 3rd ed. USA: Prentice Hall, Upper Saddle River, NJ, 2002.

[44] Z. Wu, *Hydraulic control system*, ser. 1. China: Higher Education Press, Beijing, China, 2008.

Qing Guo (M’16) received the B.E. degree in Automation from Harbin Institute of Technology, Harbin, China, in 2003, and the M.S. and Ph.D. degrees in Navigation, Guidance and Control from Harbin Institute of Technology, in 2005 and 2008.

Now He is an associate professor in School of Aeronautics and Astronautics, University of Electronic Science and Technology of China. Dr. Guo is also the member of Youth Working Committee, Fluid transmission and control branch, Chinese Mechanical Engineering Society. From December 2013 to December 2014, he was an academic visitor with Center for Power Transmission and Motion Control, Department of Mechanical Engineering, University of Bath, UK. His research interests include robust and adaptive control, electrohydraulics, exoskeleton & rehabilitation robot.

Yi Zhang (M’16) received the B.E. degree from the Automation College of the Chongqing University of China in 2007. Then he held his two M.S. degrees of Engineering from the University of Technology, Sydney in 2008 and the University of Sydney in 2009. He received the Ph.D. degree in 2014 at the University of Technology Sydney, and awarded Australian Postgraduate Award scholarship and Top-up scholarship in 2011.

Currently, he is a lecturer in Aeronautics and Astronautics, University of Electronic Science and Technology. His research interests include exoskeleton & rehabilitation robot, biomedical system modeling and control, and fault tolerant control.

Branko G. Celler (M’87-SM’12-F’14) graduated from the University of New South Wales (UNSW), Sydney, Australia, with the B.Sc. degree in 1969 and the B.E.E. degree (Hons) in 1971. He received the Ph.D. degree in 1978 at the same university. He undertook a Postdoctoral Fellowship at Johns Hopkins Baltimore, MD. He returned to the UNSW in 1981 and was Foundation Director of the Biomedical Systems Laboratory, the Human Performance Laboratory, and the Centre for Health Informatics.

He is a Fellow of the IEEE. During his career has published more than 220 journal papers and refereed conference proceedings. His research interests include biomedical instrumentation and systems, system modeling and control, noninvasive modelling of cardiovascular performance, home telecare and remote monitoring, and medical informatics.

Steven W. Su (M'99-SM'17) received both the B.S. and M.S. degrees from Harbin Institute of Technology, Harbin, China, in 1990 and 1993, respectively, and the Ph.D. degree from the Research School of Information Sciences and Engineering, Australian National University (ANU), in 2002. He was a postdoctoral research fellow in the Faculty of Engineering, University of New South Wales (UNSW) from 2002 to 2006.

Now he is currently an Associate Professor in the Faculty of Engineering and IT, University of Technology, Sydney (UTS). His main research interests include biomedical system modeling and control, nonlinear robust control, fault tolerant control, and wearable monitoring system.