Optimal tracking control for robotic manipulator using actor-critic network

*Note: Sub-titles are not captured in Xplore and should not be used

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

3rd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—This paper proposes an optimal control scheme based on the action-critic neural network(NN) for the complex mechanical manipulator system with dynamic disturbance. The actor's goal is to optimize control behavior, while the critic's goal is to evaluate control performance. The optimal control update law in the scheme can guarantee the system error and the weight estimation error SGUUB, and its stability and convergence are proved based on the direct Lyapunov method. Finally, the connecting rods on two degrees of freedom are tested to verify the effectiveness of the proposed optimal control scheme.**

***Index Terms*—Optimized tracking control; neural network (NN); adaptive dynamic programming; actor-critic**

I. Introduction

With the rapid development of computer technology and the increasing demand for automation, robotic manipulator have been used in manufacturing, military, transportation, medical industry and other special environments become more and more widely. In the face of today's increasingly serious energy crisis, we are more faced with using the least resources to complete tasks. However, due to the strong coupling, nonlinear dynamics and time-varying effects of the manipulator [1], it is very difficult to design a control method for it. In fact, the current mainstream control method of robots is still PID control based on feedforward torque compensation because of its simple calculation and low cost of setup.

To solve this problem, optimal control has been introduced [2], [3]. Generally, the optimal control of a nonlinear system such as a manipulator is obtained by solving the Hamilton-Jacobi-Bellman (HJB) equation [4]. However, due to the strong nonlinearity in the HJB equation, it is difficult to obtain an analytical solution [5]. In order to achieve optimal control, Bellman et al. proposed an effective method called DP (dynamic programming) [6]. The core of this method is the Bellman optimality principle, that is, the optimal strategy of the multi-level decision process regardless of the initial state and the initial decision, the rest of the decisions must also be an optimal strategy for the state formed by the initial decision. But DP method is difficult to dimensional explosion problem.

Identify applicable funding agency here. If none, delete this.

In order to overcome this disadvantage, Werbos proposed the framework of the ADP (adaptive dynamic programming). The main idea of the method is to estimate the cost function by using approximate structures such as neural networks, fuzzy models, and polynomials [7]. Recently, the combination of dynamic programming (ADP) and neural network-based predictive control concepts has proposed a new ADP-based model-free predictive control strategy for nonlinear systems [8]. Adaptive dynamic programming algorithm to solve the stability control problem of nonlinear system with unknown actuator saturation [9]. Aiming at the reconfigurable manipulator with saturated actuators, an optimal active fault-tolerant control method based on ADP is introduced in [10], and the best performance index function can be estimated by establishing a commenter neural network (NN) [11].

In what follows, aiming at the complex mechanical manipulator system with dynamic disturbance, we combine the optimal control technology and neural network learning method, and propose a nonlinear optimization design method. Theoretical proof and computer simulation results show that the proposed optimization method can make the output state follow the desired trajectory. The main contributions of this paper are as follows.

1) This paper uses neural network to construct evaluation factors to estimate performance indicators, and solves the HJB equation through the actor-critic strategy.
2) Based on the Lyapunov stability theory, this paper strictly proves the boundedness and tracking performance of the trajectory tracking error of the manipulator.

II. SYSTEM STATEMENT

Considering a n-DOF robot manipulator, the dynamic of robot system can be expressed as [12]

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + d(t) = \tau(t) \tag{1}$$

with $q, \dot{q}, \ddot{q} \in R^n$ the vector of joint generalized position, velocity, and acceleration, respectively. $M(q) \in R^{n\times n}$ denotes symmetric positive definite inertia matrix. $C(q,\dot{q}) \in R^{n\times n}$ denotes Coriolis and centrifugal matrix. $G(q) \in R^n$ is the

gravitational force, $\|d(t)\| \leq b_d$ is the external disturbance which is boundness. And $\tau(t) \in R^n$ is the external control torque to each joint.

Property 1. *[13] The matrix $\dot{M}(q) - 2C(q,\dot{q})$ is a skew symmetric matrix.*

Property 2. *The inertia matrix $M(q)$ is uniformly bounded, which satisfies the following inequality.*

$$m_1\|x\|^2 \leq x^T M(q) x \leq m_2\|x\|^2 \tag{2}$$

In order to reduce the initial position error between the actual position of the end of the manipulator with the desired position effectively. The tracking error is defined as

$$e(t) = q_d(t) - q(t) \tag{3}$$

where $q_d(t) \in R^n$ is the desired trajectory. In order to eliminate the error existing at the end of the manipulator, that is, let the Equation (3) converge to 0. Design the reference error as

$$r(t) = \dot{e}(t) - \Lambda e(t) \tag{4}$$

where $\Lambda \in R^{n\times n}$ is the constant gain matrix which denotes the convergence rate of the position error, $\dot{e}(t)$ denotes derivative of terminal error over time.

Then the dynamic model (1) can be written as

$$M(q)\dot{r}(t) = -C(q,\dot{q})r(t) - \tau(t) + h(x) \tag{5}$$

where $h(x) = M(q)(\ddot{q}_d + \Lambda\dot{e}) + C(q,\dot{q})(\ddot{q}_d + \Lambda\dot{e}) + G(q) + d(t)$.

Define an auxiliary control input $u(t) = h(x) - \tau(t)$. Then Equation (5) becomes

$$\dot{r}(t) = -M(q)^{-1}C(q,\dot{q})r(t) + M^{-1}u(t) \tag{6}$$

In view of Equations (4) and (6). The following augmented coupling system including thedynamics of space manipulator and estimation error is obtained

$$\begin{aligned}\dot{x}(t) &= \begin{bmatrix} -\Lambda & I \\ 0 & -M(q)^{-1}C(q,\dot{q}) \end{bmatrix} x + \begin{bmatrix} 0 \\ M(q)^{-1} \end{bmatrix} u(t) \\ &= f(x) + g(x)u(t)\end{aligned} \tag{7}$$

where $x = \left[\dot{e}^{\mathrm{T}}, \dot{r}^{\mathrm{T}}\right]^{\mathrm{T}}$, with $f(x) \in R^{2n\times 2n}$, $g(x) \in R^{2n\times n}$

Our control objective is to find optimal controller for Equation (7) to guarantee

1) the system asymptotic and the tracking errors are SGUUB.
2) the actual trajectory of manipulator can track desired trajectory with a predetermined accuracy.

III. CONTROLLER DESIGN

A. Optimal Control

In order to achieve accurate trajectory tracking of the robot manipulator, we want to obtain the control policy $u \in \Psi(\Omega)$ where $\Psi(\Omega)$ is the set of admissible control [14] to minimize the performance index which defined as

$$J(x) = \int_t^\infty l(x,u)d\tau \tag{8}$$

where the cost function $l(x,u) = x(t)^{\mathrm{T}}Q(x)x(t) + u^{\mathrm{T}}u$, with $Q(x) = g(x)g(x)^{\mathrm{T}}$.

Assuming that u^* is the optimal controller that minimize the performance index. Then performance index $J(x)$ can be yield as

$$\begin{aligned}J^*(x) &= \min_{u\in\Psi(\Omega)} \int_t^\infty l(x,u)d\tau \\ &= \int_t^\infty l(x,u^*)d\tau\end{aligned} \tag{9}$$

For the continuous affine nonlinear system (7), and the corresponding performance index is shown in Equation (8). The corresponding Hamiltonian is as follows

$$\begin{aligned}H(x,u,\nabla J(x)) &= \nabla J(x)^T\dot{x}(t) + l(x,u) \\ &= \nabla J(x)^T(f(x) + g(x)u) + x(t)^{\mathrm{T}}Q(x)x(t) \\ &\quad + u^{\mathrm{T}}u\end{aligned} \tag{10}$$

where $\nabla J(x) = \partial J(x)/\partial x$ is partial derivative of $J(x)$ with regard to $x(t)$.

When both the control policy u and the cost function $J(x)$ take the optimal value. The Hamilton-Jacobi-Bellman (HJB) equation is as follows

$$\min_{u\in\Psi(\Omega)}\{H(x,u,\nabla J(x))\} = H\left(x,u^*,\nabla J^*(x)\right) = 0 \tag{11}$$

Equation (11) is a function of the control variable u, then we can derive that the derivation of both sides of the equation with respect to u as

$$\partial H(x,u,\nabla J(x))/\partial u = 2u^{\mathrm{T}} + \nabla J(x)^T g(x) = 0 \tag{12}$$

Equation (12) has a unique solution u^* as

$$u^* = -\frac{1}{2}g(x)^{\mathrm{T}}\nabla J^*(x) \tag{13}$$

Substituting (13) into (11), the HJB equation turns into

$$\begin{aligned}H\left(x,u^*,\nabla J^*(x)\right) =& x^T(t)Q(x)x(t) + \nabla J(x)^T f(x) \\ &- \frac{1}{4}\nabla J(x)^T Q(x)\nabla J(x)^T = 0\end{aligned} \tag{14}$$

It is necessary to obtain the optimal control policy u^* by solving the HJB equation. However, since nonlinearity of the HJB equation, it is hard to solve it by analytical methods. In addition, with the increase dimensions of the x and u, the amount of calculation and storage has increased dramatically, which is what we usually call the "curse of dimensionality" problem. To overcome these weaknesses, We need to use neural networks to estimate the performance index.

B. Actor-Critic Neural Network

Since neural networks have good ability to approach unknown nonlinearities, they have been widely used in robotic control.

In order to obtain the gradient term $\nabla J(x)$ by solving the HJB function (14). We break down $J(x)$ into two parts

$$J^*(x) = \alpha\|x(t)\|^2 + {J_0}^*(x) \tag{15}$$

where $J_0(x) = J(x) - \alpha\|x(t)\|^2$ with $\alpha > 0$ a constant.

Employing the RBF-NN, $J_0(x)$ can be approximate on desired accuracy, which is expressed as

$$J_0(x) = W^{\mathrm{T}}\phi(x) + \varepsilon_x \tag{16}$$

where $W \in R^m$ is the target weight vector of the neural networks which m is the neuraon number, $\phi(x)$ is the activation function, ε_x is the approximation error of neural network. The following assumption should be considered

Assumption 1. *The weight vector $W \in R^m$, the activation function $\phi(x)$, and the approximation error are upper-bounded.*

Therefore, In view of (13), the optimal control policy u^* can be rewritten as

$$u^* = -\alpha g(x)^{\mathrm{T}} x(t) - \frac{1}{2} g(x)^{\mathrm{T}} \nabla\phi(x) W^* - \frac{1}{2} g(x)^{\mathrm{T}} \frac{\partial \varepsilon_x}{\partial x} \tag{17}$$

where $\nabla\phi(x) = \frac{\partial \phi(x)}{\partial x}$ and W^* is the optimal weight matrix.

Finally, the evaluation neural network and control neural network are established to approximate the performance index (8) and control policy u respectively.

$$\hat{J}^*(x) = \alpha\|x(t)\|^2 + \hat{W}_c{}^{\mathrm{T}}\phi(x) \tag{18}$$

$$\hat{u} = \alpha g(x)^{\mathrm{T}} x(t) - \frac{1}{2} g(x)^{\mathrm{T}} \nabla\phi(x) \hat{W}_a \tag{19}$$

where $\hat{J}^*(x), \hat{u}$ denotes estimation of $J^*(x), u$, $\hat{W}_a, \hat{W}_c$ are estimation of actor and critic NN weight.

Bring the actual output of the evaluation network into the Hamilton function. We can derive that

$$\begin{aligned} H\left(x, u, \hat{W}_c\right) = {} & x^T(t)Q(x)x(t) + u^T u + (2\alpha x(t) + \nabla\phi(x) \\ & \times \hat{W}_c(t)\Big)^T (f(x) + g(x)u(t)) = e_c \end{aligned} \tag{20}$$

The goal of the evaluation network is to make $e_c = 0$, so as to satisfy the HJB equation. We define $E(t) = \frac{1}{2} {e_c}^T e_c$, in order to reduce the error e_c, we need to cauculate the negative gradient $\dot{\hat{W}}_c(t) = -\frac{k_c \gamma(t)}{1+\|\gamma(t)\|^2} \frac{\partial E(t)}{\partial \hat{W}_c}$ to satisfy $\dot{E} < 0$. Then the output of the evaluation network can approximate the performance index optimal value. The goal of a control network is to make the actual output of the control network approximate to the optimal control policy determined by the output of the evaluation network.

The actor and critic updateing law are desigened respectively as

$$\begin{aligned} \dot{\hat{W}}_c(t) = {} & -\frac{k_c \gamma(t)}{1+\|\gamma(t)\|^2} \Big(\gamma^T(t)\hat{W}_c(t) - \left(\alpha^2 - 1\right) \\ & \times x^T(t)Q(x)x(t) + 2\alpha x^T(t) f(x) \\ & + \frac{1}{4}\hat{W}_a^T(t)\Phi(x)\hat{W}_a(t)\Big) \end{aligned} \tag{21}$$

$$\begin{aligned} \dot{\hat{W}}_a(t) = {} & \frac{1}{2}\nabla\phi^T(x)Q(x)x(t) - k_a\Phi(x)\hat{W}_a(t) \\ & + \frac{k_c}{4\left(1+\|\gamma(t)\|^2\right)}\Phi(x)\hat{W}_a(t)\gamma^T(t)\hat{W}_c(t) \end{aligned} \tag{22}$$

where $\Phi(x) = \nabla\phi(x)g(x)g(x)^T$, $\gamma(t) = \nabla\phi^T(x)\Big(f(x) - \alpha Q(x)x(t) - \frac{1}{2}Q(x)\nabla\phi(x)\hat{W}_a(t)\Big)$, and $k_c, k_a > 0$ denote the updateing rate.

C. Stability Analysis

Theoerm 1. *For the manipulator system (1) the following objectives can be achieved*

1) *the state variables x and weights $\tilde{W}_c, \tilde{W}_a$ are semi ulobally uniformly ultimately bounded(SGUUB).*
2) *the actual trajectory of manipulator can track desired trajectory $q_d(t)$ with a predetermined accuracy.*

Proof. Choose the following barrier Lyapunov function as

$$V(t) = \frac{1}{2}x^T(t)x(t) + \frac{1}{2}\tilde{W}_a^T(t)\tilde{W}_a(t) + \frac{1}{2}\tilde{W}_c^T(t)\tilde{W}_c(t) \tag{23}$$

In view of optiaml control policy (19), we have the derivative of V as

$$\begin{aligned} \dot{V}(t) \le {} & \alpha x^T(t)Q(x)x(t) + x^T(t)f(x) \\ & - \frac{1}{2}x^T(x)Q(x)\nabla\phi(x)\hat{W}_a + \tilde{W}_a^T\dot{\hat{W}}_a + \tilde{W}_c^T\dot{\hat{W}}_c \end{aligned} \tag{24}$$

Applying the following fact that

$$x^T(t)f(x) \le \frac{1}{2}\|x(t)\|^2 + \frac{1}{2}\|f(x)\|^2 \tag{25}$$

$$\begin{aligned} -k_a\tilde{W}_a^T(t)\Phi(x)\hat{W}_a(t) = {} & -\frac{k_a}{2}\tilde{W}_a^T(t)\Phi(x)\tilde{W}_a(t) \\ & -\frac{k_a}{2}\hat{W}_a^T(t)\Phi(x)\hat{W}_a(t) + \frac{k_a}{2}W^{*T}\Phi(x)W^* \end{aligned} \tag{26}$$

The derivative of V can be derived as

$$\begin{aligned} \dot{V}(t) \le {} & -\chi_1(t) - \chi_2(t) - \chi_4(t) \\ & + \xi_1(t) + \tilde{W}_c^T\dot{\hat{W}}_c + \eta_1(t) + \eta_2(t) \end{aligned} \tag{27}$$

where $\chi_1(t) = x^T(t)((\alpha - \frac{1}{2})Q(x) - E)x(t)$, $\chi_2(t) = \frac{k_a}{2}{\tilde{W}_a}^T\Phi(x){\tilde{W}_a}^T$, $\chi_4(t) = \frac{k_a}{2}{\hat{W}_a}^T\Phi(x)\hat{W}_a$, $\eta_1(t) = \frac{k_a+1}{2}W^{*T}\Phi(x)W^*$, $\eta_2(t) = \frac{1}{2}\|f(x)\|^2$, $\xi_1(t) = \frac{k_c}{4(1+\|\gamma(t)\|^2)}\tilde{W}_a^T\Phi(x)\hat{W}_a(t)\gamma^T(t)\hat{W}_c(t)$

Substituting the $\tilde{W}_c^T\dot{\hat{W}}_c$, the derivative of V can be rewritten as

$$\begin{aligned} \dot{V}(t) \le {} & -\chi_1(t) - \chi_2(t) - \chi_3(t) - \frac{k_a}{2}{\hat{W}_a}^T\Phi(x)\hat{W}_a \\ & + \xi_1(t) + \xi_2(t) + \xi_3(t) + \eta_1(t) + \eta_2(t) \end{aligned} \tag{28}$$

where $\xi_2(t) = -\frac{k_c}{4(1+\|\gamma(t)\|^2)}\tilde{W}_c^T\gamma(t)\tilde{W}_a^T\Phi(x)\hat{W}_a(t)$, $\chi_3(t) = \frac{k_c}{1+\|\gamma(t)\|^2}\tilde{W}_c^T(t)\gamma(t)\gamma^T(t)\tilde{W}_c(t)$, $\xi_3(t) = \frac{k_c}{4(1+\|\gamma(t)\|^2)}\tilde{W}_c^T\gamma(t)W^{*T}\Phi(x)\tilde{W}_a(t)$.

Considering $\xi_1(t) + \xi_2(t) = \xi_4(t)$, where $\frac{k_c}{4(1+\|\gamma(t)\|^2)}\tilde{W}_a^T\nabla\phi^T(x)g(x)W^{*T}\gamma(t)g^T(x)\nabla\phi(x)\hat{W}_a(t)$.

$$\begin{aligned} \dot{V}(t) \le {} & -\chi_1(t) - \chi_2(t) - \chi_3(t) - \frac{k_a}{2}{\hat{W}_a}^T\Phi(x)\hat{W}_a \\ & + \xi_4(t) + \xi_3(t) + \eta_1(t) + \eta_2(t) \end{aligned} \tag{29}$$

Notice that

$$\xi_4(t) \le \xi_5(t) + \xi_6(t) \tag{30}$$

TABLE I
PARAMETERS FOR ROBOTIC SYSTEM

Parameter	Description	value
m_1	mass of link 1	1.0Kg
m_2	mass of link 2	1.0Kg
l_1	length of link 1	0.8m
l_2	length of link 2	0.7m
I_1	Inertia of link 1	$0.25{m_1}{l_1}^2$
I_2	Inertia of link 2	$0.25{m_2}{l_2}^2$

where $\xi_6(t) = \frac{k_c^2}{2}\hat{W}_a^T(t)\Phi(x)\hat{W}_a(t)$, $\xi_5(t) = \frac{1}{32}\tilde{W}_a^T(t)\nabla\phi^T(x)g(x)W^{*T}\gamma(t)\gamma^T(t)W^*g^T(x)\nabla\phi(x)\tilde{W}_a(t)$.

$$\xi_3(t) \leq \xi_7(t) + \xi_8(t) \tag{31}$$

where $\xi_8(t) = \frac{k_c^2}{2}\tilde{W}_a^T(t)\Phi(x)\tilde{W}_a(t)$, $\xi_7(t) = \frac{1}{32(1+\|\gamma(t)\|^2)}\tilde{W}_c^T(t)\gamma(t)W^{*T}\Phi(x)W^*\gamma^T(t)\hat{W}_c(t)$.

Considering the above inequalities (30) (31), we can derive that

$$\begin{aligned}\dot{V}(t) \leq & -\chi_1(t) - (\chi_2(t) - \xi_5(t) - \xi_8(t)) - (\chi_3(t) - \xi_7(t)) \\ & - (\chi_4(t) - \xi_6(x)) + \eta_1(t) + \eta_2(t)\end{aligned} \tag{32}$$

According to the property 2. we can derive that $Q(x)$ is bounded, let $\alpha_1 = \alpha - \frac{1}{2}\|Q\|_{min} - 1$, $\alpha_2 = (\left(\frac{k_a}{2} - \frac{k_c^2}{2} - \frac{1}{32}\|W^*\|^2\right)\|Q\|_{min})$, $\alpha_3 = \left(\frac{k_c}{2} - \frac{\|Q\|_{max}}{32}\|W^*\|^2\right)$.

Then Equation (32) can be rewritten as

$$\begin{aligned}\dot{L}(t) \leq & -\alpha_1 x^T(t)x(t) - \alpha_2\tilde{W}_a^T(t)\tilde{W}_a(t) - \alpha_3\tilde{W}_c^T(t)\tilde{W}_c(t) \\ & + \eta_1(t) + \eta_2(t) \\ & \leq -\alpha_{min}V(t) + L\end{aligned} \tag{33}$$

where $\alpha_{min} = min\{\alpha_1, \alpha_3, \alpha_3\}$, $L = max\{\eta_1(t) + \eta_2(t)\}$

Inequality (33) indicate the state variable $x(t)$ $\tilde{W}_c$, and$\tilde{W}_a$ are SGUUB

The proof is completed.

IV. SIMULATION RESULTS

In this section, a 2-DOF manipulator is carried out to verify the effectiveness of the proposed control scheme. The parameters for robotic system is defined as Table. I.

In addition, the weighting matric (4) is $\Lambda = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$, the control Parameter β in (19) is chosen as $\beta = 50$.

In order to estimate $J_0(x)$, a RBF-NN is employed, the number of neural network nodes we selected as $n = 16$. And the basis function based on Gaussian radial function is as $\phi(x) = \exp\left[\frac{-(x-\mu_k)^1(x-\mu_k)}{\eta_k^2}\right], k = 1, 2, \ldots 16$ where the variance $\eta_k = 1$, $\mu_k = [\mu_{k1}, \mu_{k2}, \mu_{k3}, \mu_{k4}]^T$. The k_c, k_a in updateing laws (21) (22) are defined as $k_c = 5$, $k_a = 30$ respectively. The initial weight $\hat{W}_c = 0.3(i = 1, 2, \ldots 16)$, $\hat{W}_a = 0.5(i = 1, 2, \ldots 16)$.

The problem we consider is that the manipulator tracks a circle with a center at $x = y = 0$ and a radius of $0.1m$. The desired position and external disturbance are given as follows

$$d(\dot{q}) = \begin{bmatrix} \dot{q}_1 + 0.1\sin(\dot{q}_1) \\ \dot{q}_2 + 0.1\sin(\dot{q}_2) \end{bmatrix} \quad q_d(t) = \begin{bmatrix} 0.1 \times \sin(\pi t) \\ 0.1 \times \cos(\pi t) \end{bmatrix} \tag{34}$$

The Figs. (1-4) display the simuliation results. Fig. (1) and Fig. (3) illustrate the trajectory tracking performance, the trajectory under the proposed controller coincides with the desired trajectory precisely. Fig. (2) shows the tracking errors of link1 and link2. In addition, Fig. (4) is the control input signals with the trajectory tracking control. Fig. (5) illustrate the boundness of actor and critic weight. It can be seen from the above figures the trajectory under the proposed controller coincides with the desired trajectory precisely.

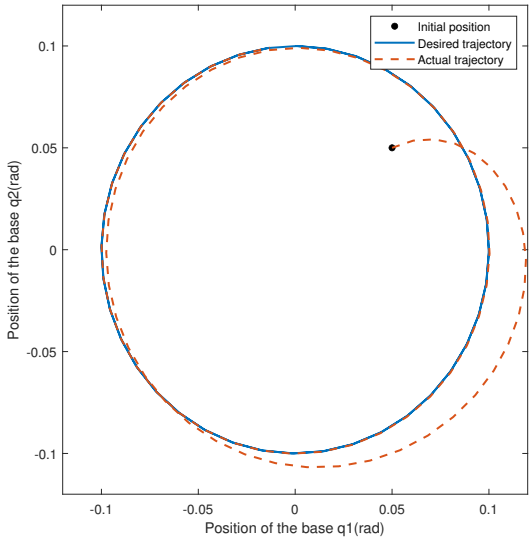

Fig. 1. Performance of trajectory tracking

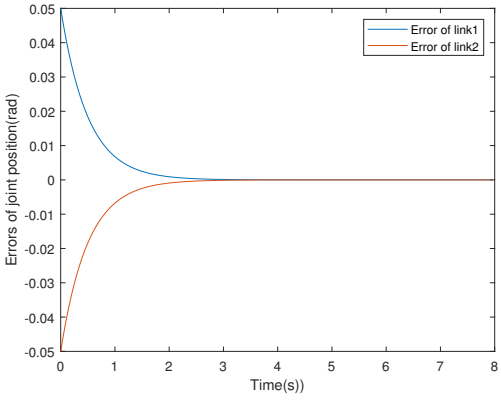

Fig. 2. Tracking errors of the link1 and link2

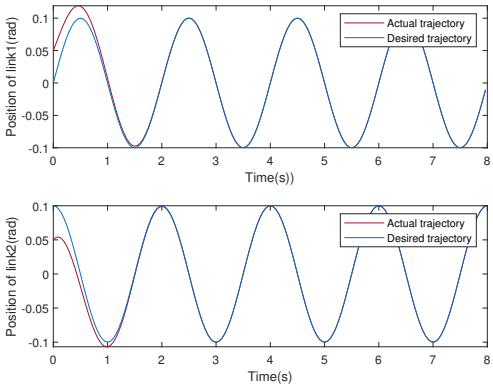

Fig. 3. Tracking performance of optimal controller

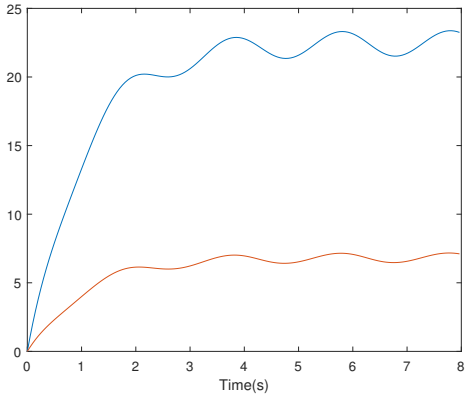

Fig. 4. Control signal for the trajectory tracking

V. CONCLUSION

In this paper, we propose an optimal control algorithm based on the actor-critic neural network for the complex mechanical manipulator system with dynamic disturbance. The goal of the actor is to optimize the control behavior, and the goal of the critic is to evaluate the control performance and return to the evaluation to improve the actor. By proving its stability and convergence, the optimal control update law can guarantee the system error and the weight estimation error SGUUB. Finally, an example is used to verify the effectiveness of the proposed optimal control scheme.

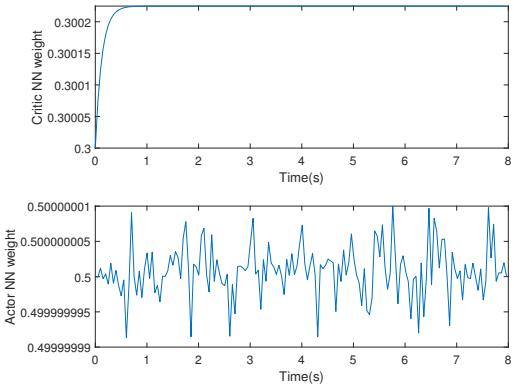

Fig. 5. The norm of actor and critic weight

REFERENCES

[1] Guangran Cheng and Lu Dong. Optimal control for robotic manipulators with input saturation using single critic network. In *2019 Chinese Automation Congress (CAC)*, pages 2344–2349. IEEE, 2019.

[2] Young H Kim, Frank L Lewis, and Darren M Dawson. Intelligent optimal control of robotic manipulators using neural networks. *Automatica*, 36(9):1355–1364, 2000.

[3] Bo Dong, Tianjiao An, Fan Zhou, Keping Liu, Weibo Yu, and Yuanchun Li. Actor-critic-identifier structure-based decentralized neuro-optimal control of modular robot manipulators with environmental collisions. *IEEE Access*, 7:96148–96165, 2019.

[4] Hamidreza Modares, Frank L Lewis, and Mohammad-Bagher Naghibi-Sistani. Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems. *Automatica*, 50(1):193–202, 2014.

[5] Guoxing Wen, CL Philip Chen, Shuzhi Sam Ge, Hongli Yang, and Xiaoguang Liu. Optimized adaptive nonlinear tracking control using actor–critic reinforcement learning strategy. *IEEE transactions on industrial informatics*, 15(9):4969–4977, 2019.

[6] R Bellman. Dynamic programming, princeton, nj: Princeton univ. *versity Press. BellmanDynamic Programming1957*, 1957.

[7] David A White and Donald A Sofge. *Handbook of intelligent control: Neural, fuzzy, and adaptive approaches*. Van Nostrand Reinhold Company, 1992.

[8] Na Dong and Zengqiang Chen. A novel adp based model-free predictive control. *Nonlinear Dynamics*, 69(1-2):89–97, 2012.

[9] Zhao, Bo, Jia, Lihao, Xia, Hongbing, Li, and Yuanchun. Adaptive dynamic programming-based stabilization of nonlinear systems with unknown actuator saturation. *Nonlinear Dynamics*, 2018.

[10] Fujie Nie, Fan Zhou, Bo Dong, Tianjiao An, and Yuanchun Li. Optimal fault tolerant control of reconfigurable manipulator with actuator saturation. In *2020 Chinese Control And Decision Conference (CCDC)*, pages 428–433. IEEE, 2020.

[11] Biao Luo, Derong Liu, and Huai-Ning Wu. Adaptive constrained optimal control design for data-based nonlinear discrete-time systems with critic-only structure. *IEEE Transactions on Neural Networks and Learning Systems*, 29(6):2099–2111, 2017.

[12] Jean-Jacques E Slotine and Weiping Li. Composite adaptive control of robot manipulators. *Automatica*, 25(4):509–519, 1989.

[13] Rastko R Selmic and Frank L Lewis. Deadzone compensation in motion control systems using neural networks. *IEEE Transactions on Automatic Control*, 45(4):602–613, 2000.

[14] Danil V Prokhorov and Donald C Wunsch. Adaptive critic designs. *IEEE transactions on Neural Networks*, 8(5):997–1007, 1997.