Hindawi
Complexity
Volume 2021, Article ID 8839391, 7 pages
https://doi.org/10.1155/2021/8839391

Research Article

Online Optimal Control of Robotic Systems with Single Critic NN-Based Reinforcement Learning

Xiaoyi Long, Zheng He, and Zhongyuan Wang

School of Computer Science, Wuhan University, Wuhan 430072, China

Correspondence should be addressed to Zheng He; hezheng@whu.edu.cn

Received 29 August 2020; Revised 21 October 2020; Accepted 11 January 2021; Published 9 February 2021

Academic Editor: Jing Na

This paper suggests an online solution for the optimal tracking control of robotic systems based on a single critic neural network (NN)-based reinforcement learning (RL) method. To this end, we rewrite the robotic system model as a state-space form, which will facilitate the realization of optimal tracking control synthesis. To maintain the tracking response, a steady-state control is designed, and then an adaptive optimal tracking control is used to ensure that the tracking error can achieve convergence in an optimal sense. To solve the obtained optimal control via the framework of adaptive dynamic programming (ADP), the command trajectory to be tracked and the modified tracking Hamilton-Jacobi-Bellman (HJB) are all formulated. An online RL algorithm is the developed to address the HJB equation using a critic NN with online learning algorithm. Simulation results are given to verify the effectiveness of the proposed method.

1. Introduction

In the control field and practical applications, reinforcement learning (RL) [1, 2] and adaptive dynamic programming (ADP) [3, 4] play a critical role to address the optimal control problems. The purpose of optimal control is to design a stabilizing control law by minimizing a predefined performance function. A lot of work focusing on the regulation problem for optimal control using the RL/ADP algorithms has been reported [5, 6] in the past years. The objective is to solve an optimal control that can maximize or minimize the system output energy and control actions, where the associated optimal control equations can be numerically solved via neural networks (NNs). From the perspective of both the theoretical study and practical application, these results pave a new way to solve the optimal control problems. Relevant surveys about the recent developments on the RL and ADP can be referred to [7, 8].

RL was first developed in the intelligent control field, which was used to address the discrete-time Markov decision problems. Then, it has been extended to solve the continuous-time (CT) systems. With respect to optimal control designs, Abu-Khalaf et al. [9] suggested a policy iteration (PI) for the optimal regulation of CT nonlinear systems with actuator saturation. To overcome the problem of using the time derivatives of CT dynamics, Lewis et al. [10] developed an integral RL (IRL) technique for systems with partially known dynamics, where the full system information is avoided. In [11], the authors have employed an actor-critic structure and developed a synchronous PI algorithm for CT systems. In this framework, both the optimal cost function and control policy are estimated using NNs, whose weights are updated online simultaneously. For completely unknown system dynamics, the results in [12] showed that a model-free PI approach can be developed for CT linear systems, which can online calculate the optimal solutions using the input/output measurements. This principle was subsequently extended to nonlinear systems in [13, 14]. Another intelligent learning method used in the RL, experience replay (ER), was recently incorporated into the synthesis of ADP for optimal control in [15], where the past system observations are utilized together with the current information to enhance the convergence speed of the online learning.

Moreover, most existing results on the ADP-based optimal control designs focus on the optimal regulation

problems only. However, in the practical application, optimal tracking control problem (OTCP) is more widely used than the optimal regulation problem [16–18], in particular for robotic applications. However, the OTCP is more challenging to address than the regulation problem, since its solution is usually composed of a feedforward action to guarantee the perfect tracking and a feedback action to stabilize the closed-loop system dynamics [19]. For linear systems, the solution of OTCP is calculated by addressing the Riccati equations [20], while for nonlinear systems, the existing solution for the OTCP can be derived with a feedforward term by employing the dynamics inversion and a feedback term by calculating a complex HJB equation [16, 21]. However, it is well known that deriving the solution of OTCP is typically intractable, especially for the online tracking control. Hence, only few results have been reported to address the OTCP in the literature, in particular for robotic systems.

According to the above facts, we propose a new RL algorithm to realize the optimal tracking control of robotic systems. To this end, the system model is rewritten as a state-space form, which will contribute to the realization of optimal tracking control. Then, a steady-state control is designed, and then an adaptive optimal tracking control is used to retain that the tracking error converges in an optimal manner. To derive this optimal control, the command trajectory to be tracked and the modified tracking Hamilton–Jacobi–Bellman (HJB) can be formulated. Finally, an online RL method is used to address the derived HJB equation using a single critic NN approximation. Numerical simulations are also given to show the validity of the proposed approach. The contributions can be summarized as follows:

(1) To achieve the optimal tracking control, the robotic system model is transformed into a canonical form, which will contribute to the realization of optimal tracking control.

(2) A critic NN is applied to reconstruct the cost function with guaranteed convergence, such that the actor NN used in the existing ADP structures is avoided and the computational costs can be reduced.

(3) A RL algorithm is proposed to obtain the solution of the derived HJB equation, which can guarantee the convergence of critic NN weights to ensure the optimal tracking error convergence.

The paper is structured as follows: in Section 2, the system model is transformed into a canonical form, and a tracking performance function is constructed. In Section 3, an adaptive steady-state control is designed and an optimal control is developed with RL to make the tracking error dynamics convergent. For this purpose, a single critic NN is applied to estimate the solution of the HJB equation and update the optimal control action. Section 4 gives some simulation results to show the validity of the developed control and learning techniques. Conclusions are summarized in Section 5.

Notations: $\mathfrak{R}$ denotes the real number set. $\mathfrak{R}^n$ is the n-dimensional real vector. $\mathfrak{R}^{n\times m}$is the real matrices. $\|\cdot\|$ denotes the Euclidean norm of a vector in $\mathfrak{R}^n$ or a matrix in $\mathfrak{R}^{n\times m}$. I is the identity matrix, and 0×0 means the zero matrix. $\lambda_{\max}$ and $\lambda_{\min}$ are the maximal and minimum eigenvalues of a matrix, respectively. $\mathrm{diag}\{[a_1, a_2, a_3, \ldots, a_n]\}$ is a diagonal matrix with component $a_1, \ldots, a_n$. $(\cdot)_x = \partial(\cdot)/\partial(x)$ defines the partial differential operation.

2. Preliminaries and Problem Statement

In this paper, we consider the general rigid body dynamics for a nonlinear robotic manipulator. On the basis of the Lagrangian formulation, the manipulator dynamics can be formulated as [22, 23]

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

where $q = q(t) \in \mathfrak{R}^n$ denotes the generalized coordinates representing the joint position and $\dot{q}$, $\ddot{q}$ denote the derivatives of joint position (e.g., velocity and acceleration) with respect to time t. Let n denote the number of degrees of freedom and $M(q)$ denote a positive definite $n\times n$ inertia matrix which is invertible, $C(q,\dot{q}) \in \mathfrak{R}^n$ are the Coriolis/centripetal dynamics, and $G(q)$ represents the gravitational dynamics.

For the brevity of notation, we set $x_1 = q$, $x_2 = \dot{q}$, and $u = \tau$; then,

$$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = -M^{-1}Cx_2 - M^{-1}G + M^{-1}u. \end{cases} \tag{2}$$

Hence, we write system (1) as a state-space form as

$$\dot{x}(t) = f(x) + gu(t), \tag{3}$$

where $x(t) = \begin{bmatrix} q & \dot{q} \end{bmatrix}^T \in \mathfrak{R}^n$ is the system state, $f(x) = \begin{bmatrix} x_2 & -M^{-1}(Cx_2 + G) \end{bmatrix}^T$ is the known system dynamics, which is a continuous function with $f(0) = 0$, $g = \begin{bmatrix} 0 & M^{-1} \end{bmatrix}^T$is the control gain matrix, and $u(t) \in \mathfrak{R}^m$ denotes the control torque.

Assumption 1. The system dynamics $f(x)$ with $f(0) = 0$ are Lipschitz. Hence, system (3) is stable, i.e., a continuous control $u(t) \in \Omega$ can be found to stabilize the system, where Ω is an admissible set.

This paper aims to find an optimal control $u(t)$ to ensure that the system state $x(t)$ tracks a desired trajectory $x_d(t)$ by minimizing the following cost function:

$$V(u(t), e(t)) = \int_t^\infty \left[e^T(t)Qe(t) + u^T(t)Ru(t)\right]\mathrm{d}t, \tag{4}$$

where Q and R are positive definite symmetric matrices [2]. $\Psi(\Omega)$ is the set of admissible policies [9], $e^T(t)Qe(t) + u^T(t)Ru(t)$ is a positive utility function, and $e(t)$ is the tracking error defined as

$$e(t) = x(t) - x_d(t), \tag{5}$$

where $x_d(t) \in \mathfrak{R}^n$ is the reference trajectory. In this paper, the reference trajectory $x_d(t)$ and its derivative $\dot{x}_d(t)$ are assumed to be continuous and bounded.

Remark 1. There have been several methods developed to solve the tracking control problem of robotic systems, e.g., [24, 25]. However, most existing robotic controllers are not designed in the optimal manner, i.e., the required control actions may be large. Differing from these results, a novel RL algorithm is proposed in the following sections to design an optimal control for robotic systems to achieve trajectory tracking and reduce the required control energy.

Remark 2. For linear OTCP scheme, the derivation of optimal solution is shown in [18], i.e., $u = -R^{-1}B_0^T Sx + R^{-1}B_0^T v_{ss}$ with B_0 being the input matrix, and S can be solved by addressing a Riccati equation and v_{ss} is the feedforward action. Different from linear systems, the tracking control problem for nonlinear systems is not trivial since we cannot obtain a uniform formulation as the linear cases. This fact stimulated the current study.

3. Online Dynamic Tracking Control

To realize the optimal tracking control design, we decompose the control input u into two parts [18, 21] as

$$u = u_s + u_e, \tag{6}$$

where u_s and $u_e$are the steady-state control and optimal control, which are applied to make the steady-state trajectory tracking and stabilize the tracking error dynamics optimally, respectively. Figure 1 shows the proposed control system structure.

3.1. Steady-State Tracking Control. Since u_s can be adopted to ensure that the control error converges to zero in the steady-state, then we have from (3) that

$$u_s = g^+ \left[\dot{x}_d - f(x) - Ke\right], \tag{7}$$

where $K > 0$ is the feedback gain set by the desingers and $g^+ = [(g^T g)^{-1} g^T]$ is defined as the generalized inverse of g.

Then, based on (3) and (7), we have the tracking error e as

$$\dot{e} = f(x) + g\left(u_s + u_e\right) - \dot{x}_d = -Ke + gu_e. \tag{8}$$

From equations (7) and (8), we know that the tracking control of system (3) can be considered the regulation problem of (8). Hence, an optimal control u_e will be designed to stabilize tracking error dynamics (8) in an optimal manner.

3.2. Approximate Optimal Tracking Control. The controller u_e can be used to make (8) converge in an optimal sense. For this purpose, we can rewrite the infinite horizon cost function (4) as follows:

$$V(e(t)) = \int_t^\infty \left[e^T(t)Qe(t) + u_e^T(t)Ru_e(t)\right]\mathrm{d}t. \tag{9}$$

Then, an admissible control policy u_e should be found so that cost function (9) of system (8) can be minimized. To this end, the Lyapunov equation of (7) is given by

$$0 = V_e^T\left[-Ke + gu_e\right] + e^T Qe + u_e^T Ru_e, \tag{10}$$

with $V_e = (\partial V/\partial e)$ being the partial differential.

Then, the optimal cost function $V^*(e)$ is given as

$$V^*(e) = \min_{u_e}\left(\int_t^\infty \left[e^T(t)Qe(t) + u_e^{*T}(t)Ru_e^*(t)\right]\mathrm{d}t\right), \tag{11}$$

and the derived HJB equation is shown as

$$0 = \min_{u_e} H\left(e, u_e^*, V^*\right). \tag{12}$$

The optimal control u_e can be derived by solving $(\partial H(e, u_e^*, V^*)/\partial u_e^*) = 0$ from (10) as

$$u_e^* = -\frac{1}{2}R^{-1}g^T\frac{\partial V^*(e)}{\partial e}. \tag{13}$$

The problem to be finally addressed is to solve HJB equation (12) to obtain the optimal cost function $V^*(e)$ required in control (13).

3.2.1. Online Reinforcement Learning Algorithm. To calculate the above optimal control, we can recall the policy iteration (PI) method. Inspired by [1, 26], a policy iteration algorithm can be given as follows:

(1) Select a small positive constant κ. Let $i = 0$ and $V^{(0)} = 0$, then set an initial admissible control policy $u_e^{(0)}$.

(2) Solve the nonlinear Lyapunov equation using the control policy $u_e^{(i)}$

$$0 = e^T Qe + u_e^{T(i)}Ru_e^{(i)} + \left(\nabla V^{(i+1)}(e)\right)^T\left(-Ke + gu_e^i\right), \tag{14}$$

with $V^{(i+1)}(0) = 0$.

(3) Improve the control policy by

$$u_e^{(i+1)} = -\frac{1}{2}R^{-1}g^T\nabla V^{(i+1)}. \tag{15}$$

(4) If $|V^{(i+1)}(e) - V^{(i)}(e)| \le \kappa$, stop the iteration and take the approximate optimal control; else, let $i = i + 1$and go back to Step 2.

The above PI scheme can guarantee the convergence to the optimal cost function and control action, i.e., $V^{(i)} \longrightarrow V^*$ and $u_e^{(i)} \longrightarrow u^*$ as $i \longrightarrow \infty$. The convergence proof of the PI algorithm was detailed in [9].

3.2.2. Neural Network Approximation. The above policy iteration method is run *offline*. To implement online optimal control, we will introduce an online learning method in this section.

From HJB equation (12), it is generally difficult to derive its solution. As shown in [27, 28], we will use a critic NN to estimate the ideal cost function $V^*(e)$. In this paper, the cost

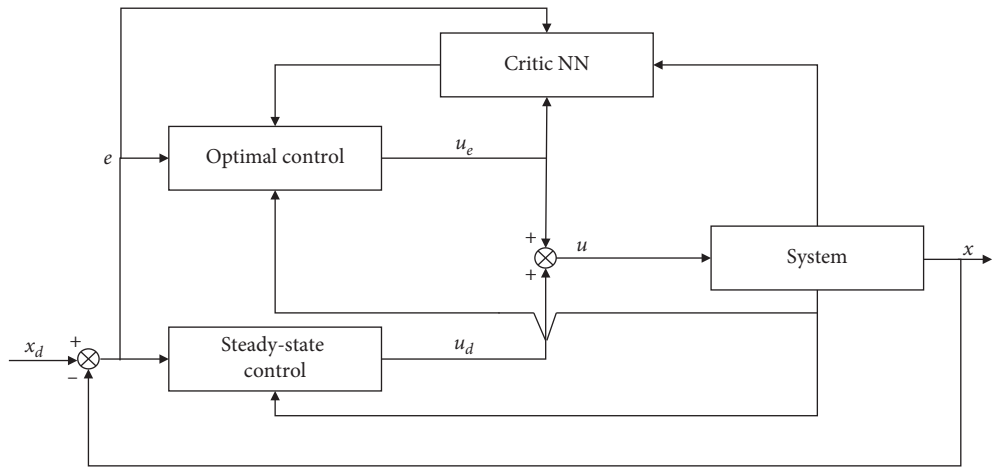

Figure 1: Schematic of the proposed control system.

function can be considered as smooth, then a critic NN [26, 28] is applied to approximate $V^*(e)$ as

$$V^*(e) = W^T\sigma(e) + \varepsilon(e), \tag{16}$$

where $W \in \mathfrak{R}^l$ is the ideal NN weight, $\sigma(e) \in \mathfrak{R}^l$ is the activation function, l is the number of neurons, and $\varepsilon(e)$ is the approximation error. Then, we have its derivative with respect to e as

$$\nabla V(e) = (\nabla\sigma(e))^T W + \nabla\varepsilon(e), \tag{17}$$

where $\nabla\sigma(e) = (\partial\sigma(e)/\partial e) \in \mathfrak{R}^{l\times n}$ and $\nabla\varepsilon(e) = (\partial\varepsilon(e)/\partial e) \in \mathfrak{R}^n$are the partial derivatives. Then, based on (17), equation (10) is represented as

$$0 = e^TQe + u_e^{*T}Ru_e^* + \left(W^T\nabla\sigma(e) + (\nabla\varepsilon(e))^T\right)\dot{e}. \tag{18}$$

Assumption 2 (see [9, 26]). Consider the critic NN weight W with the regressor $\nabla\sigma(e)$, and then the error $\varepsilon(e)$ and its gradient $\nabla\varepsilon(e)$ are all bounded. Moreover, we have $\varepsilon(e) \longrightarrow 0$ and $\nabla\varepsilon(e) \longrightarrow 0$ as $l \longrightarrow \infty$.

According to ideal cost function (16), the actual cost function can be given as

$$\widehat{V}(e) = \widehat{W}^T\sigma(e), \tag{19}$$

which can be used to estimate the practical cost function. For critic NN (19), we can select $\sigma(e)$ such that $\widehat{V}(e) > 0$ for $e \neq 0$ and $\widehat{V}(e) = 0$ for $e = 0$. Then, we have

$$\nabla\widehat{V}(e) = (\nabla\sigma(e))^T\widehat{W}, \tag{20}$$

where $\nabla\widehat{V}(e) = (\partial\widehat{V}(e)/\partial e)$.

Then, the approximated Hamiltonian function can be derived as

$$0 \triangleq H(x, u, \widehat{W}) = e^TQe + u_e^TRu_e + \widehat{W}^T\nabla\sigma(e)\dot{e} = e_c. \tag{21}$$

For training the critic NN to obtain the control action, it is expected to estimate $\widehat{W}$ to minimize the objective function $E = (1/2)e_c^Te_c$. Hence, the gradient descent algorithm can be used to update the critic NN weights $\widehat{W}$ by

$$\dot{\widehat{W}} = -\Gamma\left[\frac{\partial E}{\partial\widehat{W}}\right] = -\Gamma e_c\left[\frac{\partial e_c}{\partial\widehat{W}}\right], \tag{22}$$

where $\Gamma > 0$ is the learning gain.

Based on (18), the Hamiltonian function is

$$0 = e^TQe + u_e^{*T}Ru_e^* + W^T\nabla\sigma(e)\dot{e} = e_{c2}, \tag{23}$$

where $e_{c2} = -(\nabla\varepsilon(e))^T\dot{e}$ is the residual error.

Define $\vartheta = \nabla\sigma(e)\dot{e}$ and $\widetilde{W} = W - \widehat{W}$ as the estimation error of the critic NN weights, and a positive constant ϑ_M with $\|\vartheta\| \le \vartheta_M$. Then, from (21) and (23), we have $e_{c2} - e_c = \widetilde{W}^T\vartheta$. Hence, the estimation error dynamics are given by

$$\dot{\widetilde{W}} = -\dot{\widehat{W}} = \Gamma\left(e_{c2} - \widetilde{W}^T\vartheta\right)\vartheta. \tag{24}$$

The persistent excitation (PE) condition is required to retain the critic NN weights convergence and the condition $\|\vartheta\| \ge \vartheta_m$ with a positive constant ϑ_m. This condition can be satisfied in this paper since we consider the tracking control problem, and thus, the probing noise used in many existing ADP literature studies may be not necessary.

When implementing the online optimal control algorithm with critic NN (16), we have from (13) and (17) the optimal control as

$$u_e^* = -\frac{1}{2}R^{-1}g^T\left((\nabla\sigma(e))^TW + \nabla\varepsilon(e)\right). \tag{25}$$

Then, the approximated control action with critic NN (19) is formulated as

$$u_e^* = -\frac{1}{2}R^{-1}g^T(\nabla\sigma(e))^T\widehat{W}. \tag{26}$$

Equation (26) implies that with the updated critic NN weights $\widehat{W}$, the approximated control action can be calculated directly. Consequently, the widely used actor-critic structure can be simplified, and only the critic NN is adopted in this paper to reduce the computational cost.

Next, the stability of the proposed algorithm is given.

Lemma 1. *For error system (8) with control (26) and learning law (22), then the estimation error dynamics (24) are uniformly ultimately bounded (UUB).*

Proof. We select the Lyapunov function candidate as $J(t) = tr(\widetilde{W}^T \Gamma^{-1} \widetilde{W})$. Then, the time derivative of the Lyapunov function along the trajectory of error dynamics (24) is

$$\dot{J}(t) = 2tr\left(\widetilde{W}^T \Gamma^{-1} \dot{\widetilde{W}}\right) = 2tr\left(\widetilde{W}^T \left(e_{c2} - \widetilde{W}^T \vartheta\right)\vartheta\right). \quad (27)$$

After some mathematical manipulations, we have

$$\dot{J}(t) \leq -(2-\Gamma)\left\|\widetilde{W}^T \vartheta\right\|^2 + \frac{1}{\Gamma} e_{c2}^2. \quad (28)$$

Considering the Cauchy–Schwarz inequality and noticing the assumption $\|\vartheta\| \leq \vartheta_M$, we can conclude that $\dot{J}(t) < 0$ as long as $0 < \Gamma < 2$ and

$$\|\widetilde{W}\| > \sqrt{\frac{e_{c2}^2}{\Gamma(2-\Gamma)\vartheta_M^2}}. \quad (29)$$

According to the Lyapunov theory, we obtain that the estimation error is UUB.

4. Simulation

To demonstrate the validity of the developed method, a numerical simulation based on a SCARA robot plant is given. Consider the dynamics of a two degree-of-freedom SCARA robot system as

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -M^{-1}Cx_2 - M^{-1}G + M^{-1}u \end{bmatrix}, \quad (30)$$

where $x_1 = [q_1 \ q_2]$ and $x_2 = [\dot{q}_1 \ \dot{q}_2]$ are the SCARA robot's joint position and velocity vectors. The inverse of inertia matrix M^{-1}, Coriolis dynamics C, and gravity dynamics G is shown as

$$M^{-1} = \frac{\begin{bmatrix} b & -b-c \\ -b-c & a+b+2c \end{bmatrix}}{\left(ab - c^2\right)}, \quad C = \begin{bmatrix} -2\,d\dot{q}_2 & -d\dot{q}_2 \\ d\dot{q}_1 & 0 \end{bmatrix}, \text{ and } G = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad (31)$$

with $a = 0.613$, $b = 0.1173$, $c = 0.1584 \cos(q_2)$, and $d = 0.1584 \sin(q_2)$, and the detail modelling process can be found in [29].

Then, system (30) can be formulated as

$$\dot{x}(t) = f(x) + gu(t), \quad (32)$$

where $f(x) = [x_2 \ -M^{-1}(Cx_2 + G)]^T$ denote the drift dynamics and $g = [0 \ M^{-1}]^T$ denotes the control gain. To complete the optimal tracking control, equation (7) for system (30) can be given as $u_d = g^+[\dot{x}_d - f(x) - Ke]$ with $K = 0.8$. The initial critic NN weights are selected as

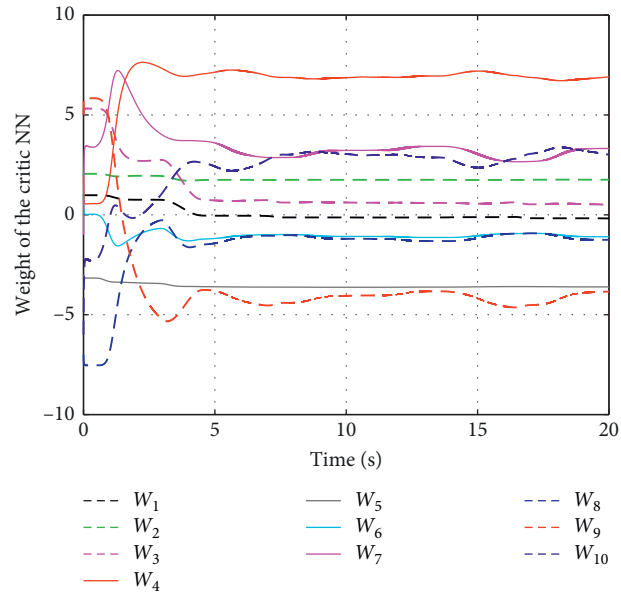

Figure 2: Convergence of the critic NN weights.

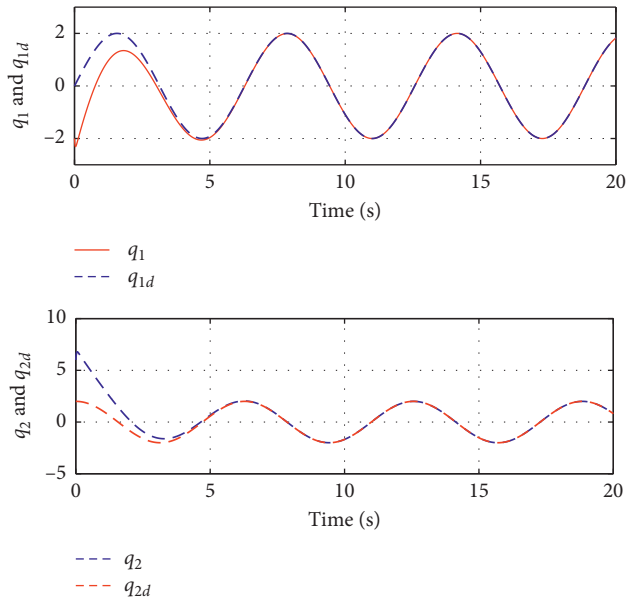

Figure 3: Position tracking of joint 1 and joint 2 of SCARA.

$\widehat{W} = [1 \ 2 \ 5 \ 2 \ -3 \ 1 \ -1 \ -7 \ 5 \ -6]^T$, and the initial system states are $x_1 = [-2, 6]^T$ and $x_2 = [-2.4, 1]^T$. In the learning process, we set the learning gain as $\Gamma = 0.02$. The activation function of the critic NN is chosen as $\sigma = [e_1^2, e_1e_2, e_1e_3, e_1e_4, e_2^2, e_2e_3, e_2e_4, e_3^2, e_3e_4, e_4^2]$, and the weighting matrices Q and R are selected as identity matrices as [29]. The desired trajectory x_d are given as $x_d = [q_{1d}, q_{2d}, dq_{1d}, dq_{2d}]$, where $q_{1d} = 2\sin(t)$, $q_{2d} = 2\cos(t)$, $dq_{1d} = q_{2d}$, and $dq_{2d} = -q_{1d}$. During the implementation of the policy iteration algorithm, we take the sinusoidal signals as the reference and thus the persistence excitation condition

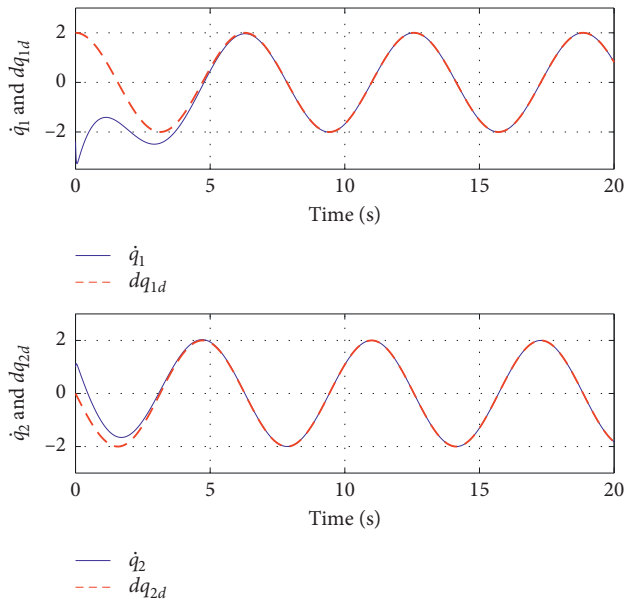

Figure 4: Velocity tracking of joint 1 and joint 2 of SCARA.

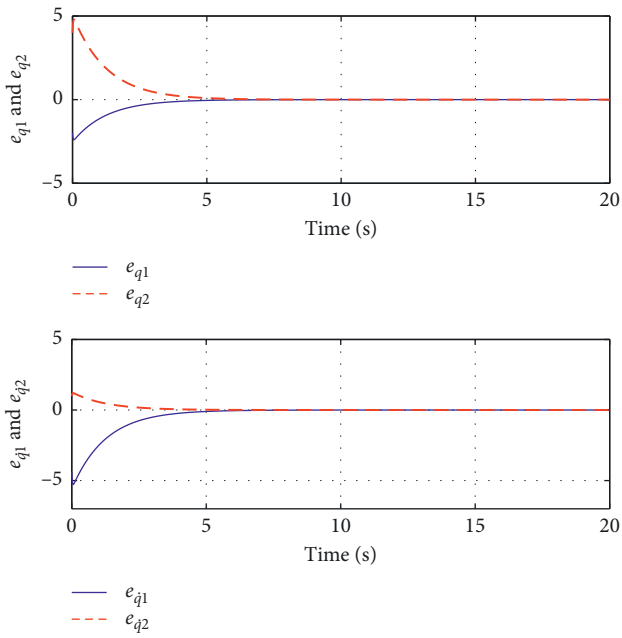

Figure 5: Tracking error convergence.

has been fulfilled. In this case, the probing noise introduced into the system can be avoided.

With these parameter configurations, numerical simulations are conducted and simulation results are given in Figures 2–5. The online profile of the approximated critic NN weights $\widehat{W}$ with proposed adaptive law (22) are displayed in Figure 2, which converge to the vector $[-0.4847, 1.6323, -0.4100, 6.0223, -3.8430, -1.3071, 1.9629, -2.2160, -3.7896, 2.9225]^T$ after a short transient stage. Clearly, we can find that the weights are convergent when

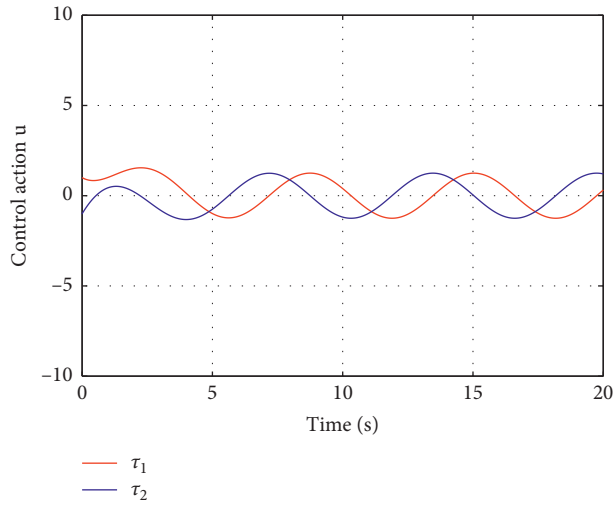

Figure 6: Control action (u).

performing the proposed online learning. With these online updated critic NN weights, the estimated control can converge to the ideal solution. As a consequence, the system tracking performances are given in Figures 3 and 4, which indicate that the system states can track the desired trajectories with the proposed optimal control. To better show the motion tracking results, the tracking errors are given in Figure 5, which converge to zero with very smooth profiles. Moreover, the control input is given as Figure 6, which is also bounded. It can be found from above simulations that the proposed optimal control can realize perfect tracking control. Specifically, the proposed RL algorithm can retain convergent response for the critic NN.

5. Conclusion

The optimal tracking control design for robotic systems using the RL algorithm is presented in this paper. The system model is first transformed into a canonical form, which can facilitate the realization of optimal tracking control design. To maintain the tracking response, a steady-state control is designed, and then an optimal tracking control is used to ensure the tracking error dynamics to be optimal. The online RL algorithm is dedicated to solving the HJB equation using a critic NN. Numerical simulations are given to show the effectiveness of the proposed technique. We will study the nonlinear optimal tracking problem with fully unknown system dynamics in the future work.

Data Availability

The data used to support the findings of this study were curated by the authors and are available upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Key R&D Project (2016YFE0202300), National Natural Science Foundation of China (U1903214, 62071339, 61671332, and U1736206), and Hubei Province Technological Innovation Major Project (2019AAA049).

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT press, Cambridge, MA, USA, 1998.

[2] F. L. Lewis and D. Liu, *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*, Vol. 17, John Wiley and Sons, Hoboken, NJ, USA, 2013.

[3] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, Vol. 703, John Wiley & Sons, Hoboken, NJ, USA, 2007.

[4] H. Zhang, D. Liu, Y. Luo, and D. Wang, *Adaptive Dynamic Programming for Control: Algorithms and Stability*, Springer Science and Business Media, Berlin, Germany, 2012.

[5] S. Bhasin, R. Kamalapurkar, M. Johnson, K. G. Vamvoudakis, F. L. Lewis, and W. E. Dixon, "A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems," *Automatica*, vol. 49, no. 1, pp. 82–92, 2013.

[6] Y. Lv, J. Na, Q. Yang, X. Wu, and Y. Guo, "Online adaptive optimal control for continuous-time nonlinear systems with completely unknown dynamics," *International Journal of Control*, vol. 89, no. 1, pp. 99–112, 2016.

[7] S. Chen Anthony and H. Guido, "Adaptive optimal control via continuous-time Q-learning for unknown nonlinear affine systems," in *Proceedings of the 2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 1007–1012, IEEE, Nice, France, December 2019.

[8] J. Y. Lee, J. B. Park, and Y. H. Choi, "Integral Q-learning and explorized policy iteration for adaptive optimal control of continuous-time linear systems," *Automatica*, vol. 48, no. 11, pp. 2850–2859, 2012.

[9] M. Abu-Khalaf and F. L. Lewis, "Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach," *Automatica*, vol. 41, no. 5, pp. 779–791, 2005.

[10] D. Vrabie and F. Lewis, "Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems," *Neural Networks*, vol. 22, no. 3, pp. 237–246, 2009.

[11] K. G. Vamvoudakis, D. Vrabie, and F. L. Lewis, "Online adaptive algorithm for optimal control with integral reinforcement learning," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 17, pp. 2686–2710, 2014.

[12] Y. Jiang and Z.-P. Jiang, "Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics," *Automatica*, vol. 48, no. 10, pp. 2699–2704, 2012.

[13] R. Song, F. L. Lewis, and Q. Wei, "Off-policy integral reinforcement learning method to solve nonlinear continuous-time multiplayer nonzero-sum games," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, pp. 704–713, 2016.

[14] R. Song, Q. Wei, and Q. Li, "Off-policy integral reinforcement learning method for multi-player non-zero-sum games," in *Adaptive Dynamic Programming: Single and Multiple Controllers*, pp. 227–249, Springer, Berlin, Germany, 2019.

[15] H. Modares, F. L. Lewis, and M.-B. Naghibi-Sistani, "Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems," *Automatica*, vol. 50, no. 1, pp. 193–202, 2014.

[16] H. Modares and F. L. Lewis, "Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning," *Automatica*, vol. 50, no. 7, pp. 1780–1792, 2014.

[17] Y. Lv, X. Ren, and J. Na, "Online Nash-optimization tracking control of multi-motor driven load system with simplified RL scheme," *ISA Transactions*, vol. 98, pp. 251–262, 2020.

[18] J. Na and G. Herrmann, "Online adaptive approximate optimal tracking control with simplified dual approximation structure for continuous-time unknown nonlinear systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 1, pp. 412–422, 2014.

[19] G. Xiao, Y. Luo, H. Zhang, and H. Jiang, "Data-driven optimal tracking control for a class of affine non-linear continuous-time systems with completely unknown dynamics," *IET Control Theory and Applications*, vol. 10, no. 6, pp. 700–710, 2016.

[20] H. Modares and F. L. Lewis, "Linear quadratic tracking control of partially-unknown continuous-time systems using reinforcement learning," *IEEE Transactions on Automatic Control*, vol. 59, no. 11, pp. 3051–3056, 2014.

[21] Y. Lv, X. Ren, and J. Na, "Adaptive optimal tracking controls of unknown multi-input systems based on nonzero-sum game theory," *Journal of the Franklin Institute*, vol. 356, no. 15, pp. 8255–8277, 2019.

[22] F. Piltan, N. Sulaiman, H. Nasiri, S. Allahdadi, and M. A. Bairami, "Novel robot manipulator adaptive artificial control: design a novel siso adaptive fuzzy sliding algorithm inverse dynamic like method," *International Journal of Engineering*, vol. 5, pp. 399–418, 2011.

[23] L. Peng and P.-Y. Woo, "Neural-fuzzy control system for robotic manipulators," *IEEE Control Systems Magazine*, vol. 22, pp. 53–63, 2002.

[24] Y.-C. Chang and B.-S. Chen, "A nonlinear adaptive H/sup/spl infin//tracking control design in robotic systems via neural networks," *IEEE Transactions on Control Systems Technology*, vol. 5, pp. 13–29, 1997.

[25] D. Zhao, S. Li, and Q. Zhu, "Adaptive synchronised tracking control for multiple robotic manipulators with uncertain kinematics and dynamics," *International Journal of Systems Science*, vol. 47, no. 4, pp. 791–804, 2016.

[26] D. Wang, D. Liu, and H. Li, "Policy iteration algorithm for online design of robust control for a class of continuous-time nonlinear systems," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 2, pp. 627–632, 2014.

[27] Y. Lv and X. Ren, "Approximate Nash solutions for multiplayer mixed-zero-sum game with reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, pp. 2739–2750, 2018.

[28] J. Zhao, J. Na, and G. Gao, "Adaptive dynamic programming based robust control of nonlinear systems with unmatched uncertainties," *Neurocomputing*, vol. 395, pp. 56–65, 2020.

[29] J. Na, M. N. Mahyuddin, G. Herrmann, X. Ren, and P. Barber, "Robust adaptive finite-time parameter estimation and control for robotic systems," *International Journal of Robust and Nonlinear Control*, vol. 25, no. 16, pp. 3045–3071, 2015.