

Research Article

Robust Trajectory Tracking of Uncertain Systems via Adaptive Critic Learning

Ziliang Zhao , Qinglin Zhu, and Bin Guo 

College of Transportation, Shandong University of Science and Technology, Qingdao 266590, China

Correspondence should be addressed to Bin Guo; gb_bb@126.com

Received 8 February 2022; Accepted 13 April 2022; Published 24 May 2022

Academic Editor: Carlos Aguilar-Ibanez

Copyright © 2022 Ziliang Zhao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This study develops an adaptive dynamic programming (ADP) scheme for uncertain systems to achieve the robust trajectory tracking. In this framework, the augmented state is first established via combining the tracking error and reference trajectory, where the robust tracking control problem can be resolved using the regulation control strategy. Then, the robust control problem of uncertain system can be represented as an optimal control problem of nominal system, which provides a new pathway to address the robust control problem. To realize the optimal control, the derived Hamilton–Jacobi–Bellman equation (HJBE) is solved by training a critic neural network (CNN). Finally, two innovative critic learning techniques are suggested to calculate the unknown NN weights, where the convergence of NN weights can be guaranteed. Simulations are carried out to demonstrate the effectiveness of the proposed method.

1. Introduction

Considering the various and complex working conditions, some systems cannot avoid suffering from the uncertainties and nonlinearities, which brings challenge for higher control performance of these systems [1, 2]. Designing a robust controller for uncertain systems that can accommodate such uncertainties has always been an important yet challenging problem. To retain the robustness property, various robust control methods have been developed against model uncertainties [3–8]. Conventional robust control methods, however, were studied mainly via *offline* means. Moreover, the majority of existing robust control designs were devoted to addressing regulation issues [9], while the tracking problems of dynamic systems are even more difficult to be tackled. Although some satisfactory results have been yielded, the developed optimal control is implemented *offline* [10, 11].

Adaptive dynamic programming (ADP) [12] is a new subject formed by the development of artificial intelligence and control. One of the well-known merits of the ADP method is that it can yield an approximation solution [13, 14] due to the employed critic neural network (CNN).

Hence, it has been widely tailored and applied to solve the unknown cost function for optimal control. However, it is worth mentioning that the classical ADP framework requires an extra actor NN [12] to estimate the optimal control action, which results in a more complex ADP structure. To further reduce the computational burden, an innovative identifier-critic-based ADP structure was reported in [15], where only one CNN is used, and a new adaptive law is introduced to retain the convergence of CNN weights. Owing to its learning ability, recent work has also been carried out to solve specific robust control problems [16, 17]. Nevertheless, most existing ADP methods aim at the optimal or robust control regulation problem only.

In fact, the trajectory tracking performance of uncertain systems is an important index to realize the stable and reliable work such as robotic systems. The tracking control of the robotic systems is mainly reflected in the tracking reference signal of each joint, so as to realize the end tracking to the desired trajectory. Therefore, it is necessary to design an appropriate controller for the uncertain nonlinear system to ensure the optimal tracking performance. Recently, some ADP-based tracking control results have been studied. In [18], an optimal tracking controller designed by the ADP

was incorporated into steady-state control, by which the steady-state tracking response can be guaranteed, but the suboptimal control problem was obtained. In [19], an optimal tracking control scheme with an iterative learning method is introduced. Nevertheless, the aforementioned results are devoted to realizing the optimal tracking control without considering the uncertain system dynamics.

Inspired by the above discussions, we will develop a novel ADP-based online adaptive critic method for robust trajectory tracking of systems with uncertain dynamics. To realize this purpose, the augmented state is established in terms of the tracking error and reference trajectory simultaneously, such that we can design the tracking controller from the perspective of regulation control problem. Then, the robust control problem is represented as an optimal control problem of the nominal systems with an appropriate index function; thus, a Hamilton–Jacobi–Bellman equation (HJBE) can be derived. Furthermore, two critic learning laws are presented to online calculate the unknown CNN weights; thus, the solution of HJBE can be resolved and the convergence of the CNN weights can be retained simultaneously. Finally, simulation results are provided to exemplify efficiency of the proposed method.

The main contributions of this study are summarized as follows:

- (1) Different from the classical robust control synthesis, the robust trajectory tracking problem is represented as an equivalent optimal regulation problem. Then, we can obtain the online solution using the critic learning algorithms based on ADP scheme.
- (2) Based on the ADP framework, a simple CNN is used to reconstruct the cost function on the premise of ensuring convergence, by which the actor NN applied in the existing ADP structure is eliminated.
- (3) A new adaptive critic learning law driven via CNN weights error is extracted to online training the CNN weights, where the convergence of the updated CNN weights can be guaranteed.

The rest of the study is arranged as follows. In Section 2, the system dynamics and problem formulation are described. Section 3 gives the robust tracking control design using the ADP scheme. Simulation results are provided in Section 4. Section 5 summarizes the main content of this study.

2. Problem Description

Consider the following system with uncertain dynamics:

$$\dot{x} = f(x) + g(x)(u + k(x)u) + g(x)d(x), \quad (1)$$

where $x \in \mathbb{R}^n$ denotes the state variables, $u \in \mathbb{R}^n$ is the control input, $f(x)$ and $g(x)$ are the differentiable functions with $f(0) = 0$, and $k(x)$ and $d(x)$ are the uncertain terms with $k(0) = 0$ and $d(0) = 0$. In this study, we presume the uncertainties $k(x) \geq 0$ and with $f_{\max}(0) = 0$ [10, 11].

The objective of this study is achieving the trajectory tracking control of uncertain system (1); hence, the tracking

error e_x between the system output x and the reference trajectory $x_d \in \mathbb{R}^n$ is designed as

$$e_x \triangleq x - x_d, \quad (2)$$

where the reference trajectory x_d is Lipschitz continuous.

According to system (1) and reference trajectory x_d , the time derivative of e_x is calculated as

$$\begin{aligned} \dot{e}_x &= \dot{x} - \dot{x}_d \\ &= f(x) + g(x)(u + k(x)u) + g(x)d(x) - \dot{x}_d. \end{aligned} \quad (3)$$

Furthermore, define augmented state $\mathcal{Y} = [e_x^T, x_d^T]^T \in \mathbb{R}^{2n}$, and then, an augmented system can be derived along with (3):

$$\begin{aligned} \dot{\mathcal{Y}} &= \begin{bmatrix} f(x) - \dot{x}_d \\ \dot{x}_d \end{bmatrix} + \begin{bmatrix} g(x) \\ 0 \end{bmatrix} u + \begin{bmatrix} g(x) \\ 0 \end{bmatrix} k(x)u, \\ &\quad + \begin{bmatrix} g(x) \\ 0 \end{bmatrix} d(x), \\ &= \mathcal{A}(\mathcal{Y}) + \mathcal{B}(\mathcal{Y})u + \mathcal{B}(\mathcal{Y})K(\mathcal{Y})u + \mathcal{B}(\mathcal{Y})F(\mathcal{Y}), \end{aligned} \quad (4)$$

where

$$\begin{aligned} \mathcal{A}(\mathcal{Y}) &= \begin{bmatrix} f(x) - \dot{x}_d \\ \dot{x}_d \end{bmatrix}, \mathcal{B}(\mathcal{Y}) = \begin{bmatrix} g(x) \\ 0 \end{bmatrix}, \\ K(\mathcal{Y}) &= k(x), F(\mathcal{Y}) = d(x). \end{aligned} \quad (5)$$

Then, we have new uncertainties $K(\mathcal{Y})$ and $F(\mathcal{Y})$ as

$$\begin{aligned} \|\mathcal{K}(\mathcal{Y})\| &\triangleq \|k(x)\| \geq 0, \\ \|\mathcal{F}(\mathcal{Y})\| &\triangleq \|f(x)\| \leq f_{\max}(x) \triangleq \lambda_d(\mathcal{Y}). \end{aligned} \quad (6)$$

According to the above development, one can find that the original tracking control problem can be transformed into a regulation problem of system (4) [20]. Different from the existed result [20], there exists the uncertain terms $K(\mathcal{Y})$ and $F(\mathcal{Y})$ in (4), making a robust control problem. Inspired by [10, 21], we will further exploit the robust trajectory tracking control action via the optimal equation.

Now, the nominal plant of system (4) can be defined:

$$\dot{\mathcal{Y}} = \mathcal{A}(\mathcal{Y}) + \mathcal{B}(\mathcal{Y})u. \quad (7)$$

Then, the following cost function (8) should be minimized via an optimal control u^* [20, 22, 23]:

$$\begin{aligned} \mathcal{V}(\mathcal{Y}(t)) &= \int_t^{\infty} e^{-\gamma(s-t)} [\mathcal{F}(\mathcal{Y}) + \mathcal{Y}^T(s)Q_T\mathcal{Y}(s) \\ &\quad + u^T(s)Ru(s)] ds, \end{aligned} \quad (8)$$

where $\gamma > 0$ is the discount factor; this can guarantee the boundedness of the cost function even if the reference trajectory and the embedded feedforward control action do not converge to zero. $Q_T = \text{diag}\{Q, 0_{n \times n}\}$, where $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{m \times m}$ denote the appropriate symmetric matrices.

The cost function, defined by \mathcal{V}^* , is the optimal value of \mathcal{V} denoted in (8), i.e.,

$$\mathcal{V}^*(\mathcal{Y}(t)) = \min_u \mathcal{V}(\mathcal{Y}(t)). \quad (9)$$

Using Bellman's optimality principle to $\mathcal{V}^*(\mathcal{Y})$ in (9), we know the derivative $\mathcal{V}_{\mathcal{Y}}^*(\mathcal{Y})$ of $\mathcal{V}^*(\mathcal{Y})$ satisfies

$$0 = \min_u H(\mathcal{Y}, u, \mathcal{V}_{\mathcal{Y}}^*) = H(\mathcal{Y}, u^*, \mathcal{V}_{\mathcal{Y}}^*), \quad (10)$$

with

$$H(\mathcal{Y}, u^*, \mathcal{V}_{\mathcal{Y}}^*) = \mathcal{F}(\mathcal{Y}) + \mathcal{Y}^T Q_T \mathcal{Y} + u^{*T} R u^* + \mathcal{V}_{\mathcal{Y}}^{*T} (\mathcal{A}(\mathcal{Y}) + \mathcal{B}(\mathcal{Y})u^*) - \gamma \mathcal{V}^*(\mathcal{Y}). \quad (11)$$

Applying the optimal control theory, the control solution of the optimization problem is

$$u^*(\mathcal{Y}) = -\frac{1}{2}R^{-1}\mathcal{B}^T(\mathcal{Y})\mathcal{V}_{\mathcal{Y}}^*(\mathcal{Y}). \quad (12)$$

To implement the tracking control, we choose the utility function $\mathcal{F}(\mathcal{Y}) \in \mathbb{R}$ as

$$\mathcal{F}(\mathcal{Y}) = \frac{1}{4}(\mathcal{V}_{\mathcal{Y}}^* \mathcal{B}(\mathcal{Y}))^T (\mathcal{V}_{\mathcal{Y}}^* \mathcal{B}(\mathcal{Y})) + \lambda_d^2(\mathcal{Y}). \quad (13)$$

Remark 1. As discussed in [22], this study constructs an equivalence between the robust trajectory tracking problem and an optimal control problem of system (4) with the cost function (9), which gives a novel idea to resolve the robust trajectory tracking problem.

3. Solving Robust Tracking Control via ADP

This section will find an optimal solution for $\mathcal{V}_{\mathcal{Y}}^*(\mathcal{Y})$ in (12). Within the ADP framework, a CNN is first constructed to update the optimal cost function; then, a CNN self-learning is proposed to learn robust tracking control solution. To improve the convergence of the CNN weights and transient performance, an adaptive learning scheme based on the estimated CNN weight error is developed to approximate the CNN weights; thus, the robust tracking control policy can be obtained.

3.1. CNN Design. Applying the universal approximation property, the estimated cost function is constructed via an CNN given in (14), which can be considered as a continuous function:

$$\mathcal{V}^*(\mathcal{Y}) = W^T \bar{\sigma}(\mathcal{Y}) + \varepsilon_v(\mathcal{Y}), \quad (14)$$

where $W \in \mathbb{R}^l$ is the ideal CNN weight, $\bar{\sigma}(\mathcal{Y}) \in \mathbb{R}^l$ denotes the activation function, l is the number of hidden neurons, and $\varepsilon_v(\mathcal{Y})$ is the construction error. Its derivative along \mathcal{Y} gives

$$\mathcal{V}_{\mathcal{Y}}^*(\mathcal{Y}) = (\nabla \bar{\sigma}(\mathcal{Y}))^T W + \nabla \varepsilon_v(\mathcal{Y}), \quad (15)$$

with $\nabla \bar{\sigma}(\mathcal{Y}) = \partial \bar{\sigma}(\mathcal{Y}) / \partial \mathcal{Y} \in \mathbb{R}^{l \times 2n}$ and $\nabla \varepsilon_v(\mathcal{Y}) = \partial \varepsilon_v(\mathcal{Y}) / \partial \mathcal{Y} \in \mathbb{R}^{2n}$.

Generally, the optimal CNN weight W is to resolve; thus, the approximated \hat{W} involved in cost function is expressed via

$$\hat{\mathcal{V}}(\mathcal{Y}) = \hat{W}^T \bar{\sigma}(\mathcal{Y}), \quad (16)$$

with \hat{W} being the approximated CNN weight. Its derivative is

$$\hat{\mathcal{V}}_{\mathcal{Y}}(\mathcal{Y}) = (\nabla \bar{\sigma}(\mathcal{Y}))^T \hat{W}. \quad (17)$$

Based on (14) and (15), we can rewrite (12) as

$$u^*(\mathcal{Y}) = -\frac{1}{2}R^{-1}\mathcal{B}^T(\mathcal{Y})[(\nabla \bar{\sigma}(\mathcal{Y}))^T W + \nabla \varepsilon_v(\mathcal{Y})]. \quad (18)$$

The practical optimal controller becomes

$$u(\mathcal{Y}) = -\frac{1}{2}R^{-1}\mathcal{B}^T(\mathcal{Y})(\nabla \bar{\sigma}(\mathcal{Y}))^T \hat{W}. \quad (19)$$

From (19), one can claim that the tracking performance (i.e., $x \rightarrow x_d$) heavily relies on the estimation performance of \hat{W} . Nevertheless, most of the traditional learning algorithms based on ADP cannot make the CNN weights converge to the true values. Therefore, an actor NN needs to be introduced to stabilize the system. To overcome this issue, we will present a novel self-learning algorithm to online learn the CNN weights without using the actor NN.

3.2. Self-Learning Robust Tracking Control. This section will develop a self-learning algorithm to online learn the CNN weights, so as to obtain the robust tracking control. To this end, based on Section 2, we have the estimated Hamiltonian function as

$$H(\mathcal{Y}, u, \hat{W}) = \mathcal{F}(\mathcal{Y}) + \mathcal{Y}^T Q_T \mathcal{Y} + u^T R u + \hat{W}^T \nabla \bar{\sigma}(\mathcal{Y}) \dot{\mathcal{Y}} - \gamma \hat{W}^T \bar{\sigma}(\mathcal{Y}) = e_c. \quad (20)$$

To realize the self-learning for CNN weights, we define the objective function $E_c = 1/2e_c^T e_c$. Then, based on the standard steepest descent algorithm, the critic learning law can be designed as

$$\dot{\hat{W}} = -\delta \left[\frac{\partial E_c}{\partial \hat{W}} \right], \quad (21)$$

with $\delta > 0$ being the learning gain.

From (20), we rewrite Hamiltonian function as

$$H(\mathcal{Y}, u, W) = \mathcal{F}(\mathcal{Y}) + \mathcal{Y}^T Q_T \mathcal{Y} + u^T R u + W^T \nabla \bar{\sigma}(\mathcal{Y}) \dot{\mathcal{Y}} - \gamma W^T \bar{\sigma}(\mathcal{Y}) = e_{ci}, \quad (22)$$

where $e_{ci} = -(\nabla \varepsilon_v(\mathcal{Y}))^T \dot{\mathcal{Y}} + \gamma \varepsilon_v(\mathcal{Y})$ is the residual error due to the neural network approximation.

Denote $\tilde{W} = W - \hat{W}$; then, recalling (20) and (22), we have $e_{ci} - e_c = \tilde{W}^T (\nabla \bar{\sigma}(\mathcal{Y}) - \gamma \bar{\sigma})$; let $\Theta = \nabla \bar{\sigma}(\mathcal{Y}) - \gamma \bar{\sigma}$; then, we have $e_{ci} - e_c = \tilde{W} \Theta$ with $\|\Theta\| \leq \Theta_M$, $\Theta \geq 0$. Thus, the weight estimation error dynamic can be given as

$$\begin{aligned}\dot{\tilde{W}} &= -\dot{\hat{W}} \\ &= \delta \left(e_{ci} - \tilde{W}^T \Theta \right) \Theta.\end{aligned}\quad (23)$$

Then, we include some results as follows.

Theorem 1. Consider system (7) with the learning law (21); the CNN weights' error is uniform ultimate boundedness (UUB).

Proof. One chooses a Lyapunov function as $J_1 = 1/\delta \text{tr}(\tilde{W}^T \tilde{W})$; then, we can derive its time derivative:

$$\begin{aligned}\dot{J}_1 &= \frac{2}{\delta} \text{tr} \left(\tilde{W}^T \dot{\tilde{W}} \right) \\ &= \frac{2}{\delta} \text{tr} \left(\tilde{W}^T \delta \left(e_{ci} - \tilde{W}^T \Theta \right) \Theta \right).\end{aligned}\quad (24)$$

Applying some mathematical operation on (24), we have

$$\dot{J}_1 \leq - (2 - \delta) \|\tilde{W}^T \Theta\|^2 + \frac{1}{\delta} e_{ci}^2. \quad (25)$$

Consider the Cauchy-Schwarz inequality and assumption $\|\Theta\| \leq \Theta_M$; we can claim that $\dot{J}_1 < 0$ for $0 < \delta < 2$ and

$$\|\tilde{W}\| > \sqrt{\frac{e_{ci}^2}{\delta(2-\delta)\Theta_M^2}}. \quad (26)$$

From the Lyapunov theory, we can obtain the CNN weights error is UUB.

In this case, using the same Lyapunov method, we can get $\|u - u^*\| \leq \kappa$ with $\kappa > 0$. This UUB stability can refer to [24]. \square

3.3. Adaptive Learning Robust Tracking Control. To realize the fast convergence of CNN weights, this section will develop a novel adaptive learning algorithm. To this end, we first substitute (15) into (10) as

$$\begin{aligned}\mathcal{F}(\mathcal{Y}) + \mathcal{Y}^T Q_T \mathcal{Y} + u^T R u \\ + W^T \{-\gamma \bar{\sigma} + \nabla \bar{\sigma} [\mathcal{A}(\mathcal{Y}) + \mathcal{B}(\mathcal{Y})u]\} + \varepsilon_{HJB} = 0,\end{aligned}\quad (27)$$

with $\varepsilon_{HJB} = \nabla \varepsilon_v^T [\mathcal{A}(\mathcal{Y}) + \mathcal{B}(\mathcal{Y})u] - \gamma \varepsilon_v$ being the HJBE approximate error.

To approximate the CNN weight W with the novel adaptive critic learning algorithm, we first introduce two variables to redefine (27), such that

$$\begin{cases} \mathcal{X} = \nabla \bar{\sigma}(\mathcal{Y}) [\mathcal{A}(\mathcal{Y}) + \mathcal{B}(\mathcal{Y})u], \\ \mathcal{Z} = \mathcal{F}(\mathcal{Y}) + \mathcal{Y}^T Q_T \mathcal{Y} + u^T R u.\end{cases}\quad (28)$$

Then, based on (28), we rewrite (27) as

$$\mathcal{Z} = -W^T \mathcal{X} - \varepsilon_{HJB}. \quad (29)$$

According to (29), the unknown CNN weights W are involved. Thus, the adaptive critic learning law can be

designed to online approximate the CNN weights W . For this purpose, two auxiliary matrices $\mathcal{U} \in \mathbb{R}^{l \times l}$ and $\mathcal{W} \in \mathbb{R}^l$ are denoted as

$$\begin{cases} \dot{\mathcal{U}} = -\ell \mathcal{U} + \mathcal{X} \mathcal{X}^T, & \mathcal{U}(0) = 0 \\ \dot{\mathcal{W}} = -\ell \mathcal{W} + \mathcal{X} \mathcal{Z}, & \mathcal{W}(0) = 0,\end{cases}\quad (30)$$

with $\ell > 0$ being the positive constant. Thus, the solution of \mathcal{U} and \mathcal{W} can be calculated based on (30) as

$$\begin{cases} \mathcal{U} = \int_0^t e^{-\ell(t-s)} \mathcal{X}(s) \mathcal{X}^T(s) ds, \\ \mathcal{W} = \int_0^t e^{-\ell(t-s)} \mathcal{X}(s) \mathcal{Z}(s) ds.\end{cases}\quad (31)$$

Furthermore, another auxiliary vector $M \in \mathbb{R}^l$ is defined along with \mathcal{U} and \mathcal{W} as

$$M = \mathcal{U} \hat{W} + \mathcal{W}. \quad (32)$$

Based on (29) and (31), one can derive $\mathcal{W} = -\mathcal{U}W + v$ with $v = -\int_0^t e^{-\ell(t-s)} \varepsilon_{HJB}(s) \mathcal{X}(s) ds$ a bounded variable, i.e., $\|v\| \leq \varepsilon_{1v}$ for a positive constant $\varepsilon_{1v} > 0$. Then, the auxiliary vector $M \in \mathbb{R}^l$ from (30)–(32) can be rewritten as

$$M = -\mathcal{U} \tilde{W} + v, \quad (33)$$

where $\tilde{W} = W - \hat{W}$ is the estimation error of NN weights.

Therefore, an adaptive critic learning law can be designed via (33) as

$$\dot{\tilde{W}} = -\mathcal{F} M, \quad (34)$$

where $\mathcal{F} > 0$ is the learning gain.

It is noted that the critic learning law (34) contains the weights \tilde{W} , which can drive the estimate \tilde{W} toward to its true values, i.e., $W \rightarrow \tilde{W}$. This helps to achieve x tracking x_d .

Before proving the convergence property, the property of matrix \mathcal{X} is examined:

Lemma 1 (see [25, 26]). *If the variable \mathcal{X} given in (28) satisfies the persistent excitation (PE) condition, then the matrix \mathcal{U} presented in (31) is positive definite.*

We can include the advantage of the designed learning technique as follows.

Theorem 2. For (29) with critic learning law (34), the weights error \tilde{W} will exponentially converge to a compact set around zero provided that the vector \mathcal{Z} is persistently excited.

Proof. According to Lemma 1, we have the minimal eigenvalue of \mathcal{U} as $\lambda_{\min}(\mathcal{U}) \varrho > 0$. One chooses a Lyapunov function as $\mathcal{V}_1 = 1/2 \tilde{W}^T \mathcal{F}^{-1} \tilde{W}$; then, its time derivative can be calculated as

$$\dot{\mathcal{V}}_1 = \tilde{W}^T \mathcal{F} \dot{\tilde{W}} = -\tilde{W}^T \mathcal{U} \tilde{W} + \tilde{W}^T v \leq -\|\tilde{W}\| (\varrho \|\tilde{W}\| - \varepsilon_{1v}). \quad (35)$$

From (35), we can conclude that the estimation error \tilde{W} converge to Ω : $\{\tilde{W} \mid \|\tilde{W}\| \leq \varepsilon_{1v}/\varrho\}$, and the magnitude is

decided by the construction error ε_v and the excitation level ϱ . Ideally, $\varepsilon_{HJB} = 0$; we have that the convergence of estimated error \tilde{W} can be guaranteed, i.e., $\tilde{W} = 0$. This completes the proof.

Consider system (7) with practical optimal control (19) and critic learning law (34); the error \tilde{W} and augmented system state \mathcal{Y} are uniformly ultimately bounded (UUB) provided that the vector \mathcal{G} satisfies the PE condition. Then, the practical optimal control u in (19) is convergent to its optimal solution u^* given in (18), i.e., $\|u - u^*\| \leq \varepsilon_u$. \square

3.4. Analysis for Robust Trajectory Tracking Performance. This section will show the tracking performance of controlled system.

Theorem 3. *Considering the augmented plant (7) and cost function (8), the optimal control action (19) can guarantee the tracking error is UUB.*

Proof. Based on (10) and (12), we have

$$\begin{aligned} \mathcal{V}^{*T}(\mathcal{Y})\mathcal{A}(\mathcal{Y}) &= -\mathcal{F}(\mathcal{Y}) - \mathcal{Y}^T Q_T \mathcal{Y} \\ &\quad - u^{*T} R u^* - \mathcal{V}_{\mathcal{Y}}^{*T} \mathcal{B}(\mathcal{Y}) u^* + \gamma \mathcal{V}^*(\mathcal{Y}) \\ &= -\mathcal{F}(\mathcal{Y}) - \mathcal{Y}^T Q_T \mathcal{Y} + u^{*T} R u^* + \gamma \mathcal{V}^*(\mathcal{Y}). \end{aligned} \quad (36)$$

Using (36), we apply the optimal control (19) to the uncertain system (4). Because we can consider the cost function $\mathcal{V}^*(\mathcal{Y})$ as a Lyapunov function and the fact $K(\mathcal{Y}) > 0$, then its time derivative can be derived as

$$\begin{aligned} \dot{\mathcal{V}}^*(\mathcal{Y}) &= \mathcal{V}_{\mathcal{Y}}^T \dot{\mathcal{Y}} \\ &= -\mathcal{F}(\mathcal{Y}) - \mathcal{Y}^T Q_T \mathcal{Y} + u^{*T} R u^* + \gamma \\ &\quad \mathcal{V}^*(\mathcal{Y}) - 2u^{*T} R u - 2u^{*T} R K(\mathcal{Y}) u \\ &\quad + \mathcal{V}_{\mathcal{Y}}^* \mathcal{B}(\mathcal{Y}) F(\mathcal{Y}) \\ &\leq -\mathcal{F}(\mathcal{Y}) - \mathcal{Y}^T Q_T \mathcal{Y} + u^{*T} R u^* + \gamma \\ &\quad \mathcal{V}^*(\mathcal{Y}) - 2u^{*T} R u + \mathcal{V}_{\mathcal{Y}}^* \mathcal{B}(\mathcal{Y}) F(\mathcal{Y}). \end{aligned} \quad (37)$$

Then, we have

$$\begin{aligned} \dot{\mathcal{V}}^*(\mathcal{Y}) &\leq -\mathcal{Y}^T Q_T \mathcal{Y} + u^{*T} R u^* + \gamma \mathcal{V}^*(\mathcal{Y}) - 2u^{*T} R u \\ &\quad - \left[\frac{1}{2} \mathcal{V}_{\mathcal{Y}}^* \mathcal{B}(\mathcal{Y}) - F(\mathcal{Y}) \right]^T \left[\frac{1}{2} \mathcal{V}_{\mathcal{Y}}^* \mathcal{B}(\mathcal{Y}) - F(\mathcal{Y}) \right] \\ &\quad - \lambda_d^2 + (F(\mathcal{Y}))^T (F(\mathcal{Y})). \end{aligned} \quad (38)$$

Consider the fact $\mathcal{Y}^T Q_T \mathcal{Y} = e_x^T Q e_x$, and $u^{*T} R u^* - 2u^{*T} R u \leq (u - u^*)^T R (u - u^*)$. We denote $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ as the minimal and maximal eigenvalues of a matrix, respectively. From the fact $\|F(\mathcal{Y})\| \leq \lambda_d^2(\mathcal{Y})$ and the optimal cost function being bounded by $B_v > 0$, we can finally derive (38):

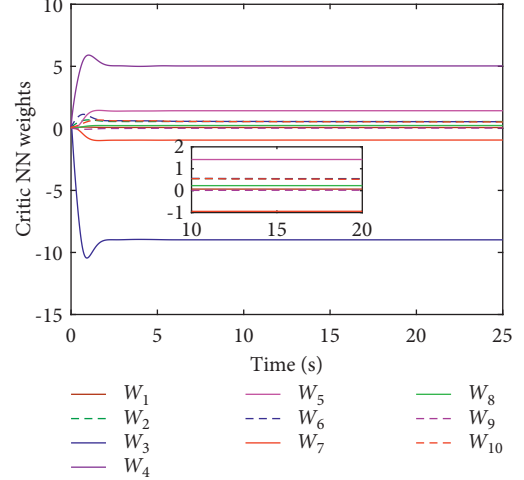


FIGURE 1: Profiles of CNN weights with (34).

$$\begin{aligned} \dot{\mathcal{V}}^*(\mathcal{Y}) &\leq e_x^T Q e_x + (u - u^*)^T R (u - u^*) + \gamma \mathcal{V}^* \\ &\leq -\lambda_{\min}(Q) \|e_x\|^2 + \lambda_{\max}(R) \Lambda^2 + \gamma B_v. \end{aligned} \quad (39)$$

To obtain $\dot{\mathcal{V}}^*(\mathcal{Y}) < 0$, the tracking error e_x should satisfy the following condition $\left\{ e_x: \|e_x\| \leq \sqrt{\lambda_{\max}(R) \Lambda^2 + \gamma B_v / \lambda_{\min}(Q)} = B_{e_x} \right\}$ with $B_{e_x} > 0$ being the constant. Moreover, we denote $\Lambda = \varepsilon_v \text{ or } \kappa$. Then, we have that applying the optimal action (19) to the uncertain system (4), the tracking error is UUB. This completes the proof. \square

4. Simulations' Verification

This section will first present numerical simulations to show the effectiveness of proposed method and then apply proposed methods to the 2-DOF robotic system to illustrate the applicability.

4.1. Numerical Simulations. In this section, we will provide numerical simulation results to illustrate the effectiveness of the developed method in terms of an uncertain system [27]:

$$\dot{x} = \begin{bmatrix} x_2 \\ -0.5(x_1 + x_2) + 0.5x_1^2 x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} (1+k)u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} d, \quad (40)$$

where $x = [x_1, x_2]^T \in \mathbb{R}^2$ is the system state, $u \in \mathbb{R}$ denotes the system input, and the terms $k(x) = p_1 x^2$ with $p_1 = [0, 1]$ and $d(x) = p_2 x_1 \sin(x_2 + p_3)$ with $p_2 = [-1, 1]$ and $p_3 = [-1, 1]$ are the uncertain parameters.

The system initial state is set as $x_0 = [-0.5, 0.5]^T$ and the desired trajectory is chosen as $x_{d1} = 0.5 \sin(t)$ and $x_{d2} = 0.5 \cos(t)$. Then, from the fact $e_x = x - x_d$ and $x_d = [x_{d1}, x_{d2}]^T$, we can construct the augmented system as (4). Hence, we can obtain

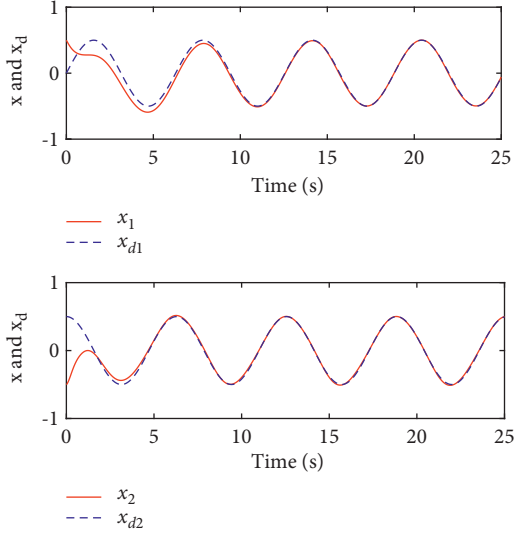


FIGURE 2: Tracking performance for uncertain system (44) with (34).

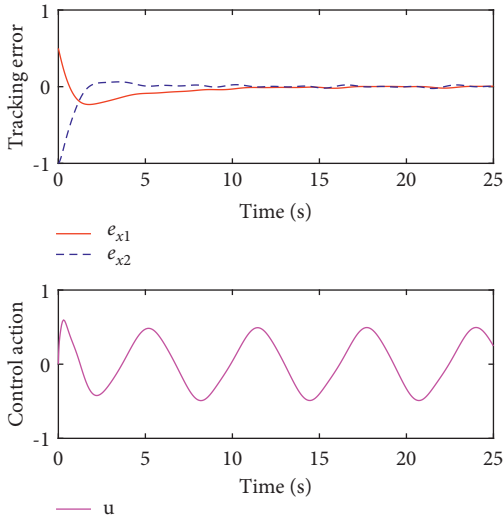


FIGURE 3: Tracking error and proposed control action with (34).

$$\begin{aligned} \|K(\mathcal{Y})\| &\triangleq \|k(x)\| \geq 0 \\ \|F(\mathcal{Y})\| &\triangleq \|d(x)\| \leq f_{\max}(x) \triangleq \lambda_d(\mathcal{Y}) = \|x_1\|. \end{aligned} \quad (41)$$

To obtain the optimal control solution, we set matrices $Q_T = \text{diag}\{8I_2, 0_{2 \times 2}\}$, $R = 1$. Moreover, the discounted factor in cost function (8) is set as $\lambda = 0.05$.

The other parameters in this simulation are selected as $\Gamma = 18.5$ and $\ell = 15.8$; the parameters in uncertain terms are selected as $p_1 = 1$, $p_2 = 0.5$, and $p_3 = 1$.

To obtain the optimal control action, we apply a single NN to reformulate HJBE (27), i.e., estimate \mathcal{V}^* ; its convergence of weights can be guaranteed using the learning law (34). In the process of simulation, we set the activation function of CNN as $\bar{\sigma} = [\mathcal{Y}_1^2, \mathcal{Y}_1 \mathcal{Y}_2, \mathcal{Y}_1 \mathcal{Y}_3, \mathcal{Y}_1 \mathcal{Y}_4, \mathcal{Y}_2^2, \mathcal{Y}_2 \mathcal{Y}_3, \mathcal{Y}_2 \mathcal{Y}_4, \mathcal{Y}_3^2, \mathcal{Y}_3 \mathcal{Y}_4, \mathcal{Y}_4^2]^T$ and the initial values for the CNN weights are chosen as $\bar{W}(0) = 0$. After the online

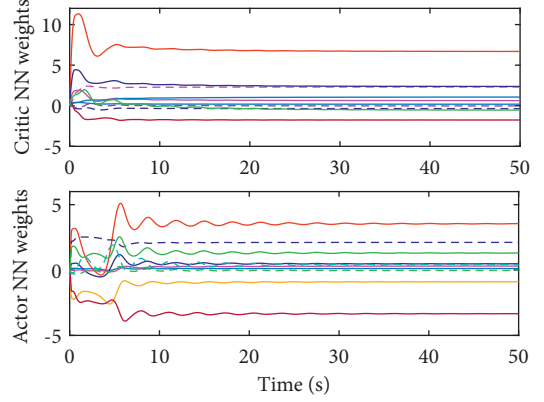


FIGURE 4: Profiles of critic-actor NN weights [18].

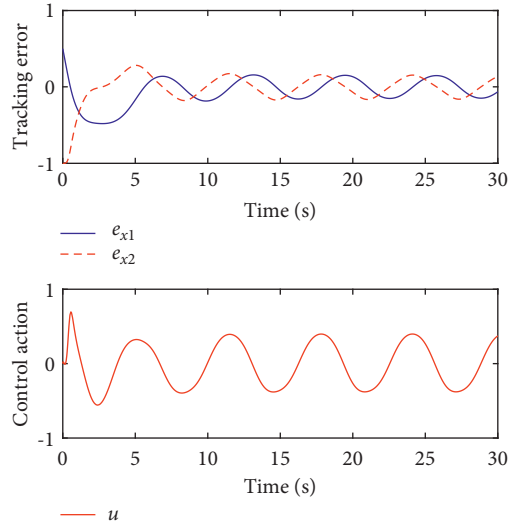


FIGURE 5: Tracking error and control action u for nonlinear systems [18].

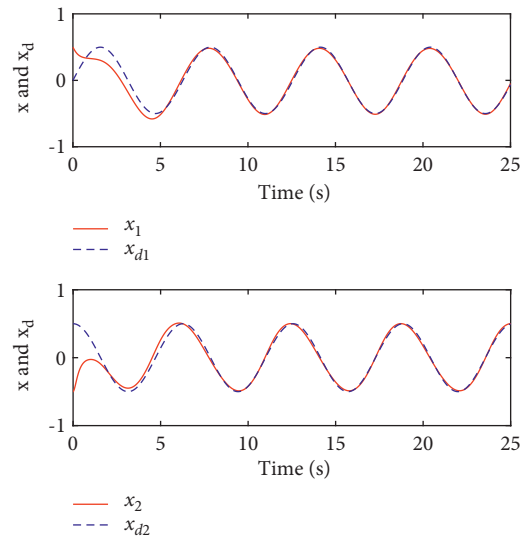


FIGURE 6: Tracking performance for uncertain system (44) with (34) (case 1).

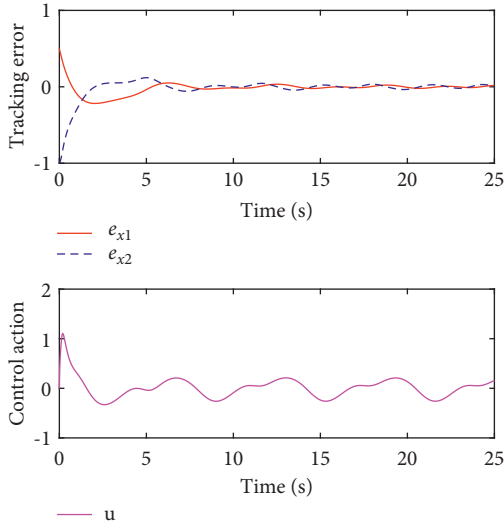


FIGURE 7: Tracking error and proposed control action with (34) (case 1).

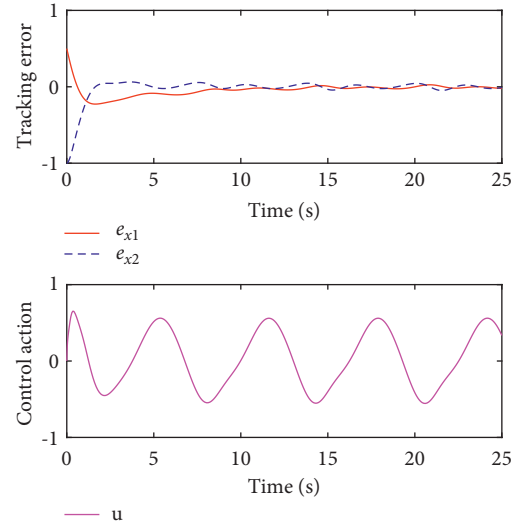


FIGURE 9: Tracking error and proposed control action with (34) (case 2).

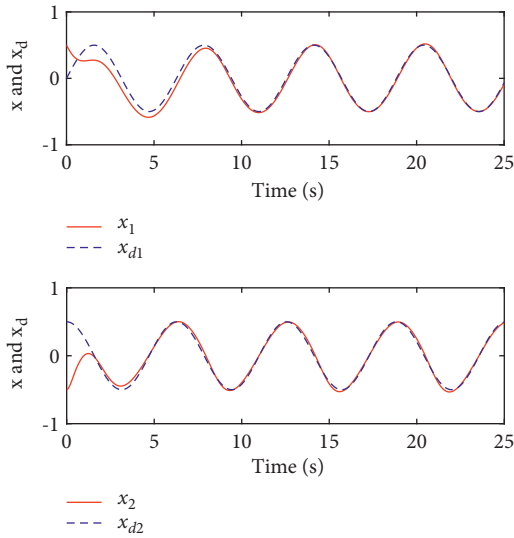


FIGURE 8: Tracking performance for uncertain system (44) with (34) (case 2).

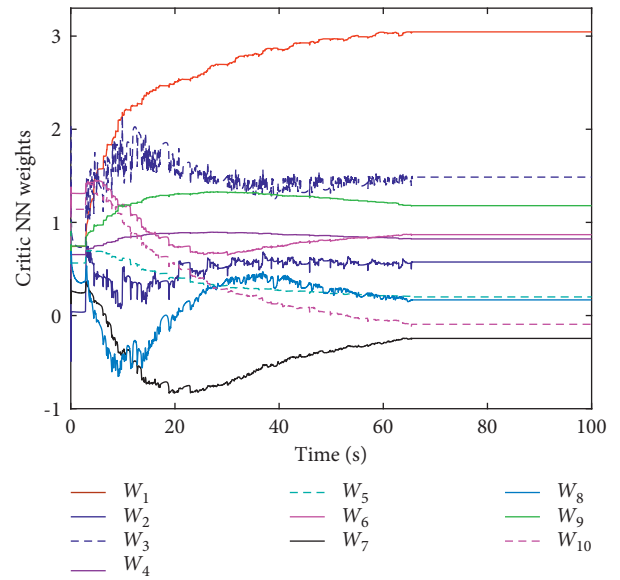


FIGURE 10: Profiles of CNN weights with (21).

learning of $t = 25s$, the CNN weights are convergence to its certain values, which are displayed in Figure 1. Then, the online updated CNN weights are applied to robust trajectory tracking implementation. Figure 2 gives the tracking performance of uncertain system (44). To better show the effectiveness of the developed method, the tracking error between practical system output x and desired trajectory x_d is provided in Figure 3, which can converge to zero, and the proposed tracking control action is also displayed in Figure 3, which is bounded and smooth.

To show the effectiveness of the proposed single critic NN-based ADP, a critic-actor NN-based ADP method [18] is proposed to use for comparison. The critic-actor weights can be found in Figure 4, and the corresponding control

performances are given in Figure 5. We have that the proposed single critic NN-based ADP has the advantages of fast convergence speed and high control accuracy.

To further show the robustness of the suggested control, we will give other two cases: (1) $p_1 = 1$, $p_2 = 1$, and $p_3 = 1$ and (2) $p_1 = 0.5$, $p_2 = 1$, and $p_3 = 0.5$. Then, the tracking performance and control action can be yielded in Figures 6–9, where we have that even under different uncertainties, the system can effectively track the reference trajectory; i.e., the robustness of the designed controller can be guaranteed.

Finally, some simulations for (21) are given in Figures 10 and 11. From these results, we have that the convergence of CNN weights in Figure 10 is more than that in Figure 1 and

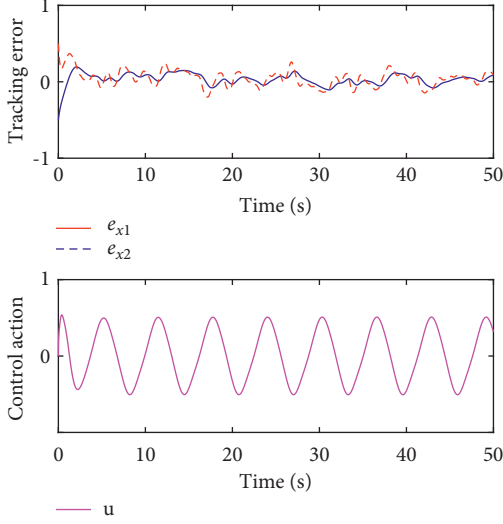


FIGURE 11: Tracking error and proposed control action with (21).

TABLE 1: Dynamic parameters of 2-DOF robot.

Symbol	Meaning	Values
J_1	The moment of inertia of link 1	0.0856
J_2	The moment of inertia of link 2	0.0707
$b_1\dot{q}_1$	The friction at joint 1	$0.014\dot{q}_1$
$b_2\dot{q}_2$	The friction at joint 2	$0.035\dot{q}_2$
m_1	Mass of joint 1	4.467
m_2	Mass of joint 2	1.694
l_1	Length 1	0.250
l_2	Length 2	0.250
r_1	The center of mass	0.079
r_2	The center of mass	0.085
m_L	Mass of load	[0, 3]
g	Gravitational acceleration	9.8

used adaptive learning [34]; this led to a poor tracking performance, as shown in Figure 11.

4.2. Application to Robotic Systems. In this section, comparative simulations are tested out via a 2-DOF robotic system, to illustrate the feasibility of the presented adaptive robust tracking control method. We first give the dynamics of the robotic system as [28]

$$\mathcal{H}(q)\ddot{q} + \mathcal{C}_\cdot(q, \dot{q}) + F(q) + G(q) = \tau, \quad (42)$$

where $q \in \mathbb{R}^n$ denotes the joint variables, $\tau \in \mathbb{R}^n$ is the generalized forces, $\mathcal{H}(q) \in \mathbb{R}^{n \times n}$ is the inertia matrix, $\mathcal{C}_\cdot(q, \dot{q}) \in \mathbb{R}^n$ represents the Coriolis/centripetal vector, $G(q) \in \mathbb{R}^n$ is the gravity vector, and $F(\dot{q}) \in \mathbb{R}^n$ is the friction vector. This study denotes $\mathcal{N}(q, \dot{q}) = \mathcal{C}_\cdot(q, \dot{q}) + F(\dot{q}) + G(q)$. $\mathcal{H}(q)$ and $\mathcal{C}_\cdot(q, \dot{q})$ are uncertain because of the unknown load and unmodeled frictions. The inertia matrix is

$$\mathcal{H}(q) = \begin{bmatrix} \mathcal{H}_{11} & \mathcal{H}_{12} \\ \mathcal{H}_{21} & \mathcal{H}_{22} \end{bmatrix}, \quad (43)$$

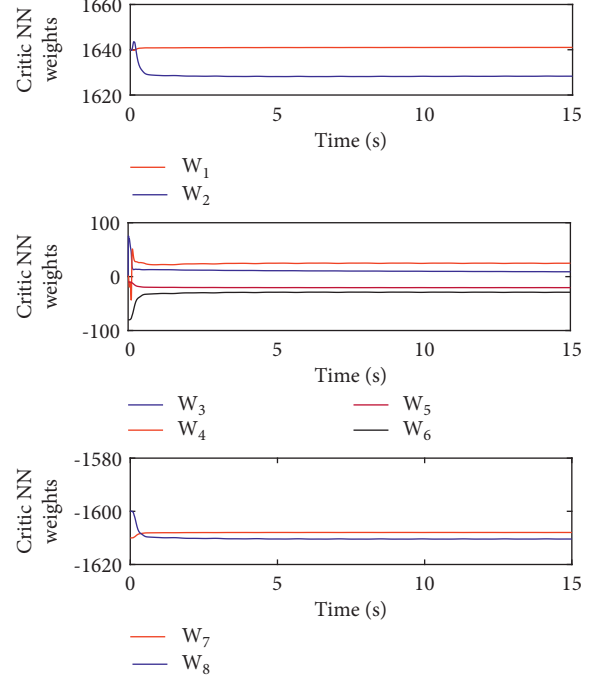
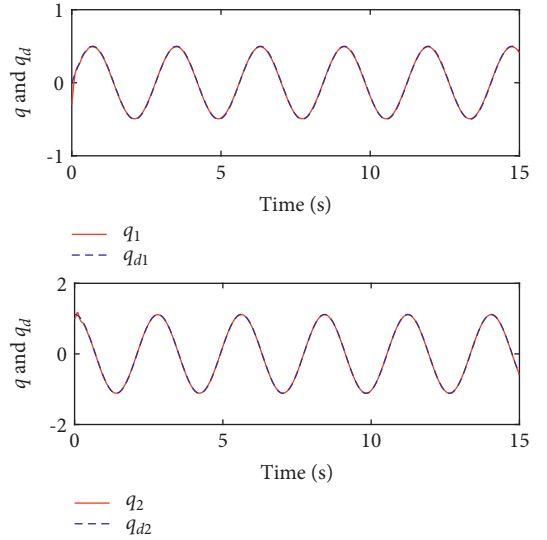


FIGURE 12: Profiles of CNN weights.

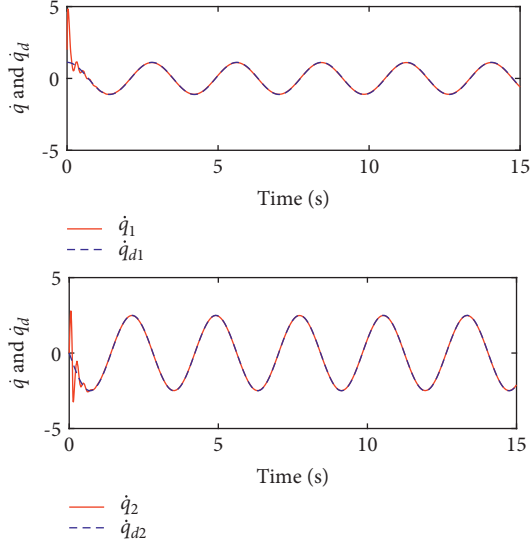
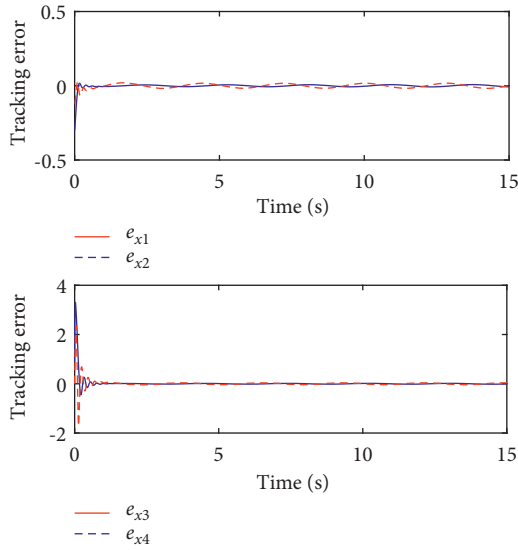
FIGURE 13: Tracking performance for q_1 and q_2 .

where $\mathcal{H}_{11} = J_1 + J_2 + m_1 r_1^2 + m_2 l_1^2 + m_2 r_2^2 + 2m_2 l_1 r_2 \cos(q_2) + m_L l_1^2 + m_L l_2^2 + 2m_L l_1 l_2 \cos(q_2)$, $\mathcal{H}_{12} = J_2 + m_2 r_2^2 + m_2 l_1 r_2 \cos(q_2) + m_L l_2^2 + m_L l_1 l_2 \cos(q_2)$, and $\mathcal{H}_{22} = J_2 + m_2 r_2^2 + m_L l_2^2$.

The centripetal vector is

$$\mathcal{C}_\cdot(q, \dot{q}) = \begin{bmatrix} \mathcal{C}_{\cdot 1} \\ \mathcal{C}_{\cdot 2} \end{bmatrix}. \quad (44)$$

where $\mathcal{C}_{\cdot 1} = (m_2 l_1 r_2 + m_L l_1 l_2)(2\dot{q}_1 - \dot{q}_2)\dot{q}_2 \sin(q_2)$ and $\mathcal{C}_{\cdot 2} = (m_2 l_1 r_2 + m_L l_1 l_2)\dot{q}_1^2 \sin(q_2)$.

FIGURE 14: Tracking performance for \dot{q}_1 and \dot{q}_2 .FIGURE 15: Tracking errors of q_1 and q_2 .

The friction vector and gravity matrix denote

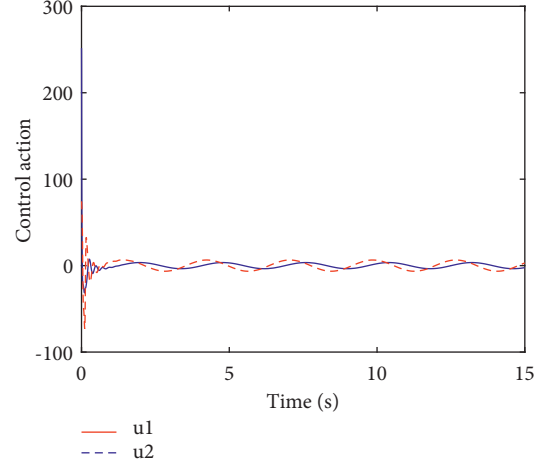
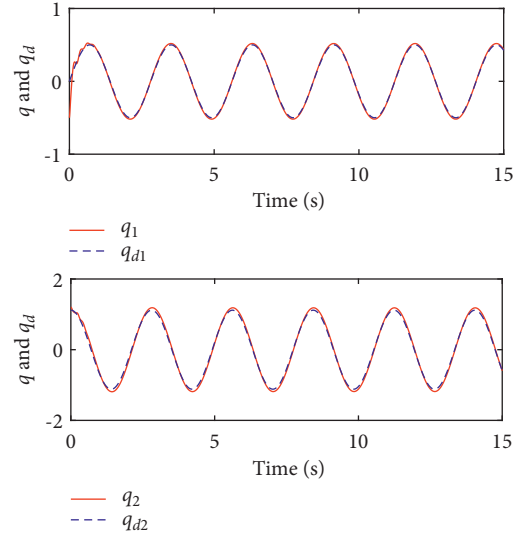
$$F(\dot{q}) = \begin{bmatrix} b_1 \dot{q}_1 \\ b_2 \dot{q}_2 \end{bmatrix}; G(q) = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix}, \quad (45)$$

where $G_1 = (m_1 g r_1 + m_2 g r_1 + m_L g r_1) \sin(q_1) + (m_2 g r_2 + m_L g l_2) \sin(q_1 + q_2)$ and $G_2 = (m_2 g r_2 + m_L g l_2) \sin(q_1 + q_2)$.

Based on the above facts, we can define the state equation of system (42) as

$$\dot{x} = f(x) + g(x)(u + k(x)u) + g(x)d(x), \quad (46)$$

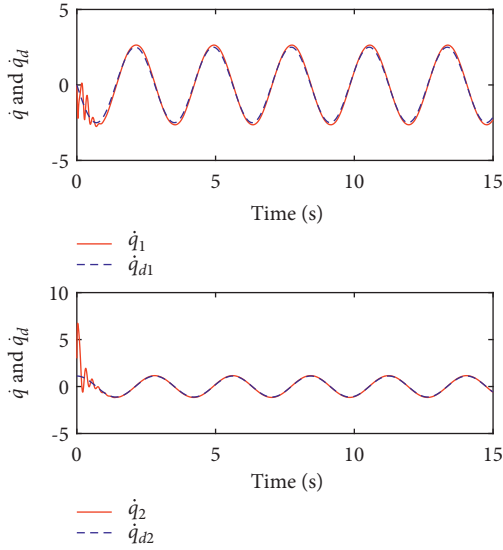
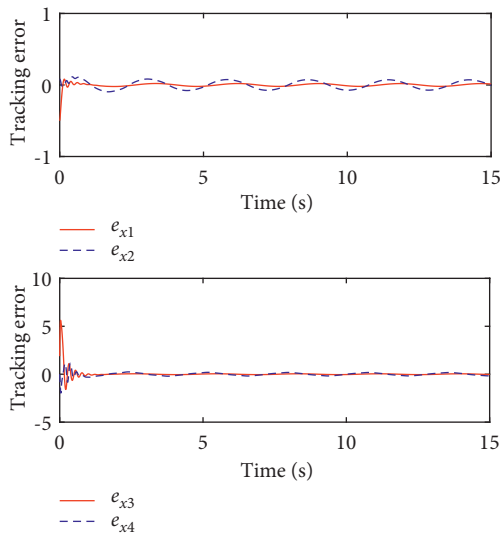
where $x = [x_1, x_2, x_3, x_4]^T = [q_1, q_2, \dot{q}_1, \dot{q}_2]^T$, $f(x) = [x_2, 0]^T$, $g(x) = [0, I]^T$, $k(x) = \mathcal{H}(x_1)^{-1} \mathcal{H}_0(x_1) - I \geq 0$, and $d(x) = \mathcal{H}^{-1}(q)(\mathcal{N}_0(q, \dot{q}) - \mathcal{N}(q, \dot{q}))$.

FIGURE 16: The required control input for q_1 and q_2 .FIGURE 17: Tracking performance for q_1 and q_2 .

To realize experimental verification, the dynamic parameters of this 2-DOF robot are given in Table 1. The upper bound of uncertainty $d(x)$ is given:

$$\begin{cases} \|d(x)\| = \|\mathcal{H}^{-1}(x_1)(\mathcal{N}_0(x_1, x_2) - \mathcal{N}(x_1, x_2))\| \\ \leq \|\mathcal{H}(x_1)\|^{-1} \|(\mathcal{N}_0(x_1, x_2) - \mathcal{N}(x_1, x_2))\| \\ \leq \|\mathcal{H}_{\min}(x_1)\|^{-1} \bar{n}_{\max}(x_1, x_2) \\ = f_{\max}(x). \end{cases} \quad (47)$$

To complete the tracking control, the sinusoidal signals $q_{1d} = 0.5 \sin(\sqrt{5}t)$, $q_{2d} = 0.5\sqrt{5} \cos(\sqrt{5}t)$, $\dot{q}_{1d} = \dot{q}_{2d}$, and $\dot{q}_{2d} = -5\dot{q}_{1d}$ are chosen as the desired trajectories. The regressor vector of CNN is designed as $\bar{\sigma} = [\mathcal{Y}_1^2, \mathcal{Y}_2^2, \mathcal{Y}_3^2, \mathcal{Y}_4^2, \mathcal{Y}_5^2, \mathcal{Y}_6^2, \mathcal{Y}_7^2, \mathcal{Y}_8^2]^T$. Because the uncertainties in $\mathcal{H}(q)$ and $\mathcal{E}_0(q, \dot{q})$ are existed due to issues such as the different load to be picked, then we consider the following cases.

FIGURE 18: Tracking performance for q and q_d .FIGURE 19: Tracking errors of q_1 and q_2 .

Case 1. In this case, we will test the tracking response under the load $m_L = 0.5kg$. The control parameters are selected as $\lambda = 0.05$, $\ell = 0.011$, $\Gamma = 0.019$, $Q = 1$, and $R = 1$.

Figure 12 shows the learning results of the online updated weights \hat{W} using the proposed method (34). Then, based on the solved results, the profiles of the tracking performance are given in Figures 13 and 14, in which one can find that both the proposed control method (34) can make both q_1 , q_2 and \dot{q}_1 , \dot{q}_2 track the given command perfectly. This can be further found in Figure 15, where the tracking errors are given. Figure 16 shows the control action, which is bounded.

Since the robotic system uncertainty mostly comes from the uncertainty of end load, then we set a Case 2; the maximum end load is given in this case.

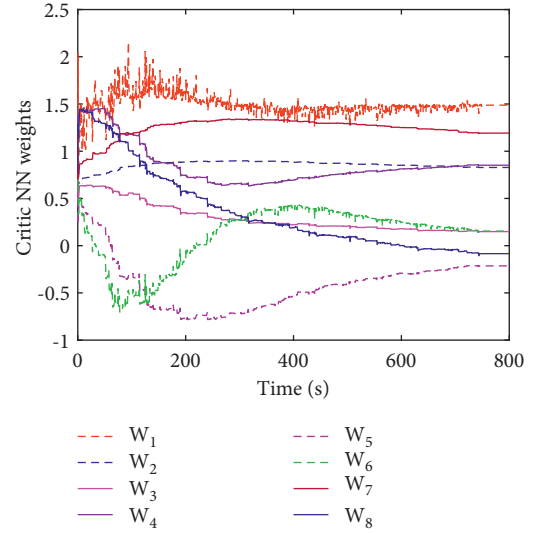
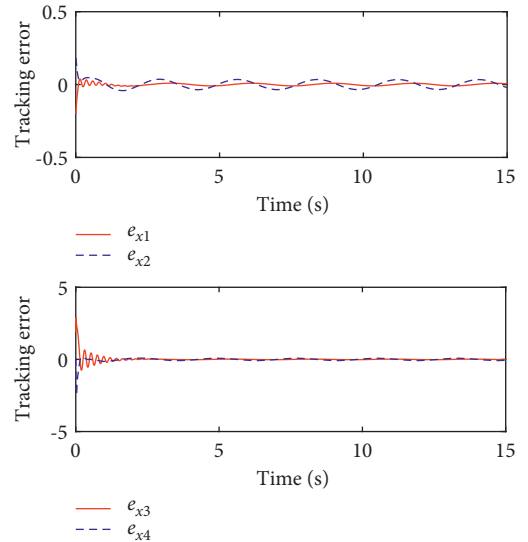


FIGURE 20: Profiles of CNN weights with (21).

FIGURE 21: Tracking errors of q_1 and q_2 .

Case 2. In this case, we will test the tracking response under the load $m_L = 3kg$, i.e., maximum end load. The control parameters are given as: $\lambda = 0.05$, $\ell = 0.011$, $\Gamma = 0.019$, $Q = 1$, and $R = 1$.

Based on the learning results in Figure 12, the tracking response in this case is given in Figures 17 and 18, and the corresponding tracking errors are given Figure 19. From Figures 17–19, we have that even if the end load of the robot becomes larger, axis 1 and axis 2 of the robotic system can track accurately on the basis of previous learning.

Finally, we give some results for proposed learning (21), as shown Figures 20 and 21, in which we have that although the accurate tracking of the robot can finally be realized, compared with the learning law (34), the learning time of neural network weight is too long, which will reduce the transient of tracking.

The above simulation results show the effectiveness of the proposed control method and the feasibility of the learning algorithm.

5. Conclusion

This study aims at resolving the robust trajectory tracking of uncertain systems, which adopts the ideas of ADP presented for optimal control. The basic concept is to represent the robust control problem as an equivalent optimal control problem for the nominal system, where the augmented state is considered. Thus, we can resolve this robust tracking control problem using an optimal control method. To obtain the optimal control solution, two adaptive critic learning techniques are developed via ADP scheme, where the parameter estimation method is adopted. The closed-loop system stability and robust tracking performance are rigorously proved; simulation results are given to exemplify the feasibility of the developed method. In our future work, we will extend the proposed idea to address the robust tracking control problem with several unmatched dynamics cases, which allows to carry out practical experimental validations based on existing test rigs in our lab.

Data Availability

The codes used to support the findings of this study are available from corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by National Key R&D Program (863 Program) (2017YFB0102800).

References

- [1] S. Wang, "Asymptotic tracking control for nonaffine systems with disturbances," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 2, pp. 479–483, 2022.
- [2] M. Bagheri, I. Karafyllis, P. Naseradinmousavi, and M. Krstic, "Adaptive control of a two-link robot using batch least-square identifier," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 1, pp. 86–93, 2021.
- [3] J. A. Meda-Campaña, J. O. Escobedo-Alva, J. D. J. Rubio et al., "On the rejection of random perturbations and the tracking of random references in a quadrotor," *Complexity*, vol. 2022, pp. 1–16, 2022.
- [4] J. D. J. Rubio, E. Orozco, D. A. Cordova et al., "Modified linear technique for the controllability and observability of robotic arms," *IEEE Access*, vol. 10, pp. 3366–3377, 2022.
- [5] C. Aguilar-Ibanez, J. Moreno-Valenzuela, O. Garcia-Alarcon, M. Martinez-Lopez, J. A. Acosta, and M. S. Suarez-Castanon, "PI-type controllers and Σ - Δ modulation for saturated DC-DC buck power converters," *IEEE Access*, vol. 9, Article ID 20346, 346 pages, 2021.
- [6] L. A. Soriano, J. D. J. Rubio, E. Orozco et al., "Optimization of sliding mode control to save energy in a scara robot," *Mathematics*, vol. 9, no. 24, p. 3160, 2021.
- [7] L. A. Soriano, E. Zamora, J. M. Vazquez-Nicolas, G. Hernández, J. A. Barraza Madrigal, and D. Balderas, "PD control compensation based on a cascade neural network applied to a robot manipulator," *Frontiers in Neurorobotics*, vol. 14, Article ID 577749, 2020.
- [8] R. Silva-Ortigoza, E. Hernandez-Marquez, A. Roldan-Caballero et al., "Sensorless tracking control for a "Full-Bridge buck inverter-DC motor" system: passivity and flatness-based design," *IEEE Access*, vol. 9, Article ID 132191, 191 pages, 2021.
- [9] K. Zhou and J. Doyle, *Essentials of Robust Control*, Prentice hall NJ, USA, 1997.
- [10] F. Lin, *Robust Control Design: An Optimal Control Approach*, John Wiley & Sons, Chichester, England, 2007.
- [11] F. Lin, R. D. Brandt, and J. Sun, "Robust control of nonlinear systems: compensating for uncertainty," *International Journal of Control*, vol. 56, no. 6, pp. 1453–1459, 1992.
- [12] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits and Systems Magazine*, vol. 9, no. 3, pp. 32–50, 2009.
- [13] P. J. Werbos, "Approximate dynamic programming for real-time control and neural modeling," *Handbook of Intelligent Control Neural Fuzzy & Adaptive Approaches*, 1992.
- [14] Y. Yang, K. G. Vamvoudakis, H. Modares, Y. Yin, and D. C. Wunsch, "Hamiltonian-driven hybrid adaptive dynamic programming," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 10, pp. 6423–6434, 2021.
- [15] J. Na and H. Guido, "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, no. 4, pp. 412–422, 2014.
- [16] 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.
- [17] D. Wang, "Robust policy learning control of nonlinear plants with case studies for a power system application," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1733–1741, 2020.
- [18] H. Huaguang Zhang, L. Lili Cui, X. Xin Zhang, and Y. Yanhong Luo, "Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 2226–2236, 2011.
- [19] A. Heydari and S. N. Balakrishnan, "Fixed-final-time optimal tracking control of input-affine nonlinear systems," *Neurocomputing*, vol. 129, pp. 528–539, 2014.
- [20] 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.
- [21] D. Wang and C. Mu, "Adaptive-critic-based robust trajectory tracking of uncertain dynamics and its application to a spring-mass-damper system," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 1, pp. 654–663, 2018.
- [22] D. Wang, L. Cheng, and J. Yan, "Self-learning robust control synthesis and trajectory tracking of uncertain dynamics," *IEEE Transactions on Cybernetics*, vol. 52, no. 1, pp. 278–286, 2020.
- [23] Y. Zhu, D. Zhao, and X. Li, "Using reinforcement learning techniques to solve continuous-time non-linear optimal tracking problem without system dynamics," *IET Control Theory & Applications*, vol. 10, no. 12, pp. 1339–1347, 2016.

- [24] D. Wang, D. Liu, C. Mu, and Y. Zhang, "Neural network learning and robust stabilization of nonlinear systems with dynamic uncertainties," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 4, pp. 1342–1351, 2018.
- [25] J. Na, S. Wang, Y. J. Liu, Y. Huang, and X. Ren, "Finite-time convergence adaptive neural network control for nonlinear servo systems," *IEEE Transactions on Cybernetics*, vol. 50, no. 6, pp. 2568–2579, 2020.
- [26] J. Zhao, Y. Lv, and Z. Zhao, "Adaptive learning based output-feedback optimal control of CT two-player zero-sum games," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 3, pp. 1437–1441, 2022.
- [27] F. Lin, "An optimal control approach to robust control design," *International Journal of Control*, vol. 73, no. 3, pp. 177–186, 2000.
- [28] C. Zhang, J. Na, J. Wu, Q. Chen, and Y. Huang, "Proportional-integral approximation-free control of robotic systems with unknown dynamics," *IEEE*, vol. 26, no. 4, pp. 2226–2236, 2021.