# Neural-Network-Based Online HJB Solution for Optimal Robust Guaranteed Cost Control of Continuous-Time Uncertain Nonlinear Systems

Derong Liu, *Fellow, IEEE*, Ding Wang, Fei-Yue Wang, *Fellow, IEEE*, Hongliang Li, *Student Member, IEEE,* and Xiong Yang

***Abstract*—In this paper, the infinite horizon optimal robust guaranteed cost control of continuous-time uncertain nonlinear systems is investigated using neural-network-based online solution of Hamilton–Jacobi–Bellman (HJB) equation. By establishing an appropriate bounded function and defining a modified cost function, the optimal robust guaranteed cost control problem is transformed into an optimal control problem. It can be observed that the optimal cost function of the nominal system is nothing but the optimal guaranteed cost of the original uncertain system. A critic neural network is constructed to facilitate the solution of the modified HJB equation corresponding to the nominal system. More importantly, an additional stabilizing term is introduced for helping to verify the stability, which reinforces the updating process of the weight vector and reduces the requirement of an initial stabilizing control. The uniform ultimate boundedness of the closed-loop system is analyzed by using the Lyapunov approach as well. Two simulation examples are provided to verify the effectiveness of the present control approach.**

***Index Terms*—Adaptive critic designs, adaptive/approximate dynamic programming (ADP), Hamilton–Jacobi–Bellman (HJB) equation, neural networks, optimal robust guaranteed cost control, uncertain nonlinear systems.**

## I. Introduction

THE adaptive or approximate dynamic programming (ADP) algorithm was first proposed by Werbos [1] as an effective method to solve optimization and optimal control problems. In general, it is implemented by solving the Hamilton–Jacobi–Bellman (HJB) equation based on function approximators, such as neural networks. It is one of the key directions for future researches in intelligent control and understanding brain intelligence [2], [3]. As a result, the ADP and related research have gained much attention from scholars across many disciplines (see [4]–[14] and the numerous references therein). Significantly, the ADP method has been often used in feedback control applications, both for discrete-time systems [15]–[36] and for continuous-time systems [37]–[54]. Besides, various traditional control problems, like robust control [55], [56], decentralized control [57], networked control [58], power system control [59], are studied under the new framework, which greatly extends the application scope of ADP methods.

Unavoidable discrepancies between system models and real-world dynamics may result in degradation of system performance including instability [60]–[62]. In this sense, the feedback control should be designed to be robust with respect to system uncertainties. The importance of robust control has been recognized by control scientists for several decades and various approaches have been proposed. In [63], it was shown that the robust control problem can be solved by studying the corresponding optimal control problem, hence the optimal control method can be employed to design robust controllers. However, the results are restricted to a class of systems with special form of uncertainties. Though Adhyaru *et al*. [55], [56] proposed an HJB equation-based optimal control algorithm to deal with the nonlinear robust control problem, the algorithm was constructed using the least square method and performed offline, not to mention the stability analysis of the closed-loop optimal control system was not conducted. On the other hand, when controlling a real plant, it is desirable to design a controller, which not only makes the closed-loop system asymptotically stable but also guarantees an adequate level of performance. The so-called guaranteed cost control approach [64] has the advantage of providing an upper bound on a given cost and thus the system performance degradation incurred by the model parameter uncertainties is guaranteed to be less than this bound [65], [66]. The optimal robust guaranteed cost control problem arises when discussing optimality of the guaranteed cost function. To the best of our knowledge, however, there are no results on optimal robust guaranteed cost control of uncertain

Manuscript received June 24, 2014; revised August 31, 2014; accepted September 1, 2014. Date of publication September 26, 2014; date of current version November 13, 2014. This work was supported in part by the National Natural Science Foundation of China under Grant 61034002, Grant 61233001, Grant 61273140, Grant 61304086, Grant 61374105, and Grant 71232006, in part by the Beijing Natural Science Foundation under Grant 4132078, and in part by the Early Career Development Award of State Key Laboratory of Management and Control for Complex Systems. This paper was recommended by Associate Editor D. Wang.

The authors are with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: dliu@ece.uic.edu; ding.wang@ia.ac.cn; feiyue.wang@ia.ac.cn; hongliang.li@ia.ac.cn; xiong.yang@ia.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2014.2357896

nonlinear systems using online ADP strategy. These motivate our research.

From a structural perspective, in many ADP-related literature, the main control strategy is implemented based on the actor-critic architecture, where two neural networks referred to as critic network and action network are taken to approximate the optimal cost function and the optimal control, respectively. In addition, from an algorithmic point of view, the value iteration and policy iteration are two important algorithms when designing the ADP-based optimal feedback control. It should be pointed out that the initial admissible control is necessary when employing the policy iteration algorithm. However, in many situations, finding the initial admissible control is not an easy task. Therefore, how to simplify the structure of ADP and relax the need for an initial stabilizing control are of great significance.

In this paper, we investigate the optimal robust guaranteed cost control of continuous-time uncertain nonlinear systems using neural-network-based online solution of HJB equation. The optimal robust guaranteed cost control problem is transformed into an optimal control problem by introducing an appropriate cost function. It can be proved that the optimal cost function of the nominal system is the optimal guaranteed cost of the controlled uncertain system. Then, a critic network is constructed for facilitating the solution of modified HJB equation. Moreover, inspired by the work of [45] and [46], an additional stabilizing term is introduced to verify the stability, which relaxes the need for an initial stabilizing control. The uniform ultimate boundedness (UUB) of the closed-loop system is also proved by using the well-known Lyapunov approach. The approximate control input can converge to the optimal control within a small bound.

In summary, the main contributions of this paper are as follows.

1) It is the first time that the infinite horizon optimal robust guaranteed cost control of uncertain nonlinear systems is investigated using the neural-network-based online HJB solution. The bounded function is introduced and the proper cost function is defined, then the optimal cost function of the nominal system is related to the optimal guaranteed cost of the original system.
2) Since the system uncertainties are not always considered in ADP-related literature, the control strategy established in this paper is significant to design robust controllers for uncertain nonlinear systems. In this sense, the conducted research extends the application scope of ADP method.

The rest of this paper is organized as follows. In Section II, the optimal robust guaranteed cost control problem of uncertain nonlinear systems is stated. In Section III, the studied problem is transformed into an optimal control problem with a modified cost function. In Section IV, a neural network is constructed to solve the modified HJB equation approximately. Then, the stability of the overall closed-loop system is proved. In Section V, two numerical examples are given to demonstrate the effectiveness of the established approach. In Section VI, concluding remarks and the discussion of future work are presented.

## II. Problem Statement

In this paper, we study a class of continuous-time uncertain nonlinear systems given by

$$\begin{aligned}\dot{x}(t) &= \overline{F}(x(t), u(t)) \\ &= f(x(t)) + g(x(t))u(t) + \Delta f(x(t))\end{aligned} \tag{1}$$

where $x(t) \in \mathbb{R}^n$ is the state vector and $u(t) \in \mathbb{R}^m$ is the control input. The known functions $f(\cdot)$ and $g(\cdot)$ are differentiable in their arguments with $f(0) = 0$, and $\Delta f(x(t))$ is the nonlinear perturbation of the corresponding nominal system

$$\dot{x}(t) = F(x(t), u(t)) = f(x(t)) + g(x(t))u(t). \tag{2}$$

Here, we let $x(0) = x_0$ be the initial state. In addition, as in many other literature, we assume that $f + gu$ is Lipschitz continuous on a set $\Omega$ in $\mathbb{R}^n$ containing the origin and that the system (2) is controllable.

Before proceeding, we assign an explicit structure to the system uncertainty. The following assumption is given, which has been used in [61] and [62].

*Assumption 1:* Assume that the uncertainty $\Delta f(x)$ has the form

$$\Delta f(x) = G(x)d(\varphi(x)) \tag{3}$$

where

$$d^{\mathsf{T}}(\varphi(x))d(\varphi(x)) \le h^{\mathsf{T}}(\varphi(x))h(\varphi(x)). \tag{4}$$

In (3) and (4), $G(\cdot) \in \mathbb{R}^{n\times r}$ and $\varphi(\cdot)$ satisfying $\varphi(0) = 0$ are known functions denoting the structure of the uncertainty, $d(\cdot) \in \mathbb{R}^r$ is an uncertain function with $d(0) = 0$, and $h(\cdot) \in \mathbb{R}^r$ is a given function with $h(0) = 0$.

Consider system (1) with infinite horizon cost function

$$\overline{J}(x_0, u) = \int_0^\infty U(x(\tau), u(\tau))\mathrm{d}\tau \tag{5}$$

where $U(x, u) = Q(x) + u^{\mathsf{T}}Ru$, $Q(x) \ge 0$, and $R = R^{\mathsf{T}} > 0$ is a constant matrix.

In this paper, the aim of solving the robust guaranteed cost control problem is to find a feedback control function $u(x)$ and determine a finite upper bound function $\Phi(u)$, i.e., $\Phi(u) < +\infty$, such that the closed-loop system is robustly stable and the cost function (5) satisfies $\overline{J} \le \Phi$. Here, the upper bound function $\Phi(u)$ is termed as a robust guaranteed cost function. Only when $\Phi(u)$ is minimized, it is named as the optimal robust guaranteed cost and is denoted as $\Phi^*$, i.e., $\Phi^* = \min_u \Phi(u)$. Additionally, the corresponding control function $\bar{u}^*$ is called the optimal robust guaranteed cost control, i.e., $\bar{u}^* = \arg\min_u \Phi(u)$.

In this paper, we will prove that the optimal robust guaranteed cost control problem of system (1) can be transformed into the optimal control problem of nominal system (2). The ADP technique can be employed to deal with the optimal control problem of system (2). Note that in this paper, the feedback control $u(x)$ is often written as $u$ for simplicity.

## III. Optimal Robust Guaranteed Cost Control of Uncertain Nonlinear Systems via HJB Solution

In this section, we show that the guaranteed cost of the uncertain nonlinear system is closely related to the modified cost function of the nominal system. The next theorem is derived by rechecking [60] with relaxed conditions.

*Theorem 1:* Assume that there exist a continuously differentiable and radially unbounded cost function $V(x)$ satisfying $V(x) > 0$ for all $x \neq 0$ and $V(0) = 0$, a bounded function $\Gamma(x)$ satisfying $\Gamma(x) \geq 0$, and a feedback control function $u(x)$ such that

$$(\nabla V(x))^{\mathsf{T}}\overline{F}(x,u) \leq (\nabla V(x))^{\mathsf{T}}F(x,u) + \Gamma(x) \tag{6}$$

$$(\nabla V(x))^{\mathsf{T}}F(x,u) + \Gamma(x) < 0, \ \ x \neq 0 \tag{7}$$

$$U(x,u) + (\nabla V(x))^{\mathsf{T}}F(x,u) + \Gamma(x) = 0 \tag{8}$$

where the symbol $\nabla V(x)$ denotes the partial derivative of the cost function $V(x)$ with respect to $x$, i.e., $\nabla V(x) = \partial V(x)/\partial x$. Then, with the feedback control function $u(x)$, there exists a neighborhood of the origin such that system (1) is locally asymptotically stable. Furthermore

$$\overline{J}(x_0,u) \leq V(x_0) = J(x_0,u) \tag{9}$$

where $J(x_0,u)$ is defined as

$$J(x_0,u) = \int_0^\infty \left\{U(x(\tau),u(x(\tau))) + \Gamma(x(\tau))\right\}\mathrm{d}\tau \tag{10}$$

and is termed as the modified cost function of system (2).

*Proof:* First, we show the asymptotic stability of system (1) under the feedback control $u(x)$. Let

$$\dot{V}(x) \triangleq \frac{\mathrm{d}V(x)}{\mathrm{d}t} = (\nabla V(x))^{\mathsf{T}}\overline{F}(x,u). \tag{11}$$

Considering (6) and (7), we obtain $\dot{V}(x(t)) < 0$ for any $x \neq 0$. This implies that $V(\cdot)$ is a Lyapunov function for system (1), which proves the local asymptotic stability.

Then, we show $\overline{J}(x_0,u)$ is upper bounded by a modified cost function corresponding to the nominal system (2).

For system (1), considering the fact that $\dot{V}(x) = (\nabla V(x))^{\mathsf{T}}\overline{F}(x,u)$, we have $U(x,u) = -\dot{V}(x) + (\nabla V(x))^{\mathsf{T}}\overline{F}(x,u) + U(x,u)$. According to (6) and (8), we have

$$\begin{aligned} U(x,u) &= -\dot{V}(x) + U(x,u) + (\nabla V(x))^{\mathsf{T}}\overline{F}(x,u) \\ &\leq -\dot{V}(x) + U(x,u) + (\nabla V(x))^{\mathsf{T}}F(x,u) + \Gamma(x) \\ &= -\dot{V}(x). \end{aligned} \tag{12}$$

Integrating over $[0,t)$ yields

$$\int_0^t U(x,u)\mathrm{d}\tau \leq -V(x(t)) + V(x_0). \tag{13}$$

Letting $t \to \infty$ and noting that $V(x(t)) \to 0$, we can obtain

$$\overline{J}(x_0,u) \leq V(x_0). \tag{14}$$

When $\Delta f(x) = 0$, we can still find that (6)–(8) are true since $\Gamma(x) \geq 0$. In this case, we derive that $\dot{V}(x) = (\nabla V(x))^{\mathsf{T}}F(x,u)$. Then, $U(x,u) + \Gamma(x) = -\dot{V}(x) + (\nabla V(x))^{\mathsf{T}}F(x,u) + U(x,u) + \Gamma(x)$. Based on (8), we obtain

$$\begin{aligned} U(x,u) + \Gamma(x) &= -\dot{V}(x) + U(x,u) \\ &\quad + (\nabla V(x))^{\mathsf{T}}F(x,u) + \Gamma(x) \\ &= -\dot{V}(x). \end{aligned} \tag{15}$$

Similarly, by integrating over $[0,t)$, we have

$$\int_0^t \left\{U(x,u) + \Gamma(x)\right\}\mathrm{d}\tau = -V(x(t)) + V(x_0). \tag{16}$$

Here, letting $t \to \infty$ yields

$$J(x_0,u) = V(x_0). \tag{17}$$

Based on (14) and (17), we can easily find that (9) is true. This completes the proof. ■

Theorem 1 shows that the bounded function $\Gamma(x)$ takes an important role in deriving the guaranteed cost of the controlled system. The following lemma presents a specific form of $\Gamma(x)$.

*Lemma 1:* For any continuously differentiable and radially unbounded function $V(x)$, define

$$\Gamma(x) = h^{\mathsf{T}}(\varphi(x))h(\varphi(x)) + \frac{1}{4}(\nabla V(x))^{\mathsf{T}}G(x)G^{\mathsf{T}}(x)\nabla V(x). \tag{18}$$

Then, we have

$$(\nabla V(x))^{\mathsf{T}}\Delta f(x) \leq \Gamma(x). \tag{19}$$

*Proof:* Considering (3), (4), and (18), since

$$\begin{aligned} 0 \leq\ & \left(d(\varphi(x)) - \frac{1}{2}G^{\mathsf{T}}(x)\nabla V(x)\right)^{\mathsf{T}}\left(d(\varphi(x)) \right. \\ & \left. - \frac{1}{2}G^{\mathsf{T}}(x)\nabla V(x)\right) \\ =\ & d^{\mathsf{T}}(\varphi(x))d(\varphi(x)) + \frac{1}{4}(\nabla V(x))^{\mathsf{T}}G(x)G^{\mathsf{T}}(x)\nabla V(x) \\ & - (\nabla V(x))^{\mathsf{T}}G(x)d(\varphi(x)) \\ \leq\ & h^{\mathsf{T}}(\varphi(x))h(\varphi(x)) + \frac{1}{4}(\nabla V(x))^{\mathsf{T}}G(x)G^{\mathsf{T}}(x)\nabla V(x) \\ & - (\nabla V(x))^{\mathsf{T}}\Delta f(x) \\ =\ & \Gamma(x) - (\nabla V(x))^{\mathsf{T}}\Delta f(x) \end{aligned} \tag{20}$$

we can see that (19) holds. ■

*Remark 1:* For any continuously differentiable and radially unbounded function $V(x)$, since

$$(\nabla V(x))^{\mathsf{T}}\overline{F}(x,u) = (\nabla V(x))^{\mathsf{T}}F(x,u) + (\nabla V(x))^{\mathsf{T}}\Delta f(x) \tag{21}$$

we can easily find that the bounded function (18) satisfies (6). Note that the Lemma 1 seems only imply (6), but in fact, it presents a specific form of $\Gamma(x)$ satisfying (6)–(8). The reason is that (7) and (8) are implicit assumptions of Theorem 1, noticing the framework of the generalized HJB equation [67] and the fact that $(\nabla V(x))^{\mathsf{T}}F(x,u) + \Gamma(x) = -U(x,u) < 0$ when $x \neq 0$. Hence, it can be used for problem transformation. In fact, based on (6) and (21), we can find that the positive semi-definite bounded function $\Gamma(x)$ gives an upper bound of the term $(\nabla V(x))^{\mathsf{T}}\Delta f(x)$, which facilitates us to solve

the optimal robust guaranteed cost control problem of a class of nonlinear systems with uncertainties.

*Remark 2:* It is important to note that Theorem 1 indicates the existence of the guaranteed cost of the uncertain nonlinear system (1). In addition, in order to derive the optimal guaranteed cost controller, we should minimize the upper bound $J(x_0, u)$ with respect to $u$. Therefore, we should solve the optimal control problem of system (2) with $V(x_0)$ considered as the cost function.

For optimal control problem, the designed feedback control must not only stabilize the controlled system on $\Omega$ but also guarantee that the cost function is finite. In other words, the control function must be admissible.

*Definition 1:* A control function $u(x)$ is said to be admissible with respect to (10) on $\Omega$, denoted by $u \in \Psi(\Omega)$ ($\Psi(\Omega)$ is the set of admissible controls on $\Omega$), if $u(x)$ is continuous on $\Omega$, $u(0) = 0$, $u(x)$ stabilizes system (2) on $\Omega$, and $J(x_0, u)$ is finite for all $x_0 \in \Omega$.

For system (2), observing that

$$\begin{aligned} V(x_0) &= \int_0^\infty \big\{U(x,u) + \Gamma(x)\big\}\mathrm{d}\tau \\ &= \int_0^T \big\{U(x,u) + \Gamma(x)\big\}\mathrm{d}\tau + V(x(T)) \end{aligned} \tag{22}$$

we have

$$\lim_{T\to 0}\frac{1}{T}\left(V(x(T)) - V(x_0) + \int_0^T \{U(x,u) + \Gamma(x)\}\,\mathrm{d}\tau\right) = 0. \tag{23}$$

Clearly, (23) is equivalent to (8). Hence, (8) is an infinitesimal version of the modified cost function (22) and is the so-called nonlinear Lyapunov equation.

For system (2) with modified cost function (22), define the Hamiltonian function of the optimal control problem as

$$H(x, u, \nabla V(x)) = U(x,u) + (\nabla V(x))^{\mathsf{T}} F(x,u) + \Gamma(x). \tag{24}$$

Define the optimal cost function of system (2) as $J^*(x_0) = \min_{u\in\Psi(\Omega)} J(x_0, u)$, where $J(x_0, u)$ is given in (10). Note that $J^*(x)$ satisfies the modified HJB equation

$$0 = \min_{u\in\Psi(\Omega)} H(x, u, \nabla J^*(x)) \tag{25}$$

where $\nabla J^*(x) = \partial J^*(x)/\partial x$. Assume that the minimum on the right hand side of (25) exists and is unique. Then, the optimal control of system (2) is

$$\begin{aligned} u^*(x) &= \arg\min_{u\in\Psi(\Omega)} H(x, u, \nabla J^*(x)) \\ &= -\frac{1}{2}R^{-1}g^{\mathsf{T}}(x)\nabla J^*(x). \end{aligned} \tag{26}$$

Hence, the modified HJB equation becomes

$$\begin{aligned} 0 = {} & U(x,u^*) + (\nabla J^*(x))^{\mathsf{T}} F(x,u^*) + h^{\mathsf{T}}(\varphi(x))h(\varphi(x)) \\ & + \frac{1}{4}(\nabla J^*(x))^{\mathsf{T}} G(x)G^{\mathsf{T}}(x)\nabla J^*(x) \end{aligned} \tag{27}$$

with $J^*(0) = 0$.

Substituting (26) into (27), we can obtain the formulation of the modified HJB equation in terms of $\nabla J^*(x)$ as follows:

$$\begin{aligned} 0 = {} & Q(x) + (\nabla J^*(x))^{\mathsf{T}} f(x) + h^{\mathsf{T}}(\varphi(x))h(\varphi(x)) \\ & - \frac{1}{4}(\nabla J^*(x))^{\mathsf{T}} g(x)R^{-1}g^{\mathsf{T}}(x)\nabla J^*(x) \\ & + \frac{1}{4}(\nabla J^*(x))^{\mathsf{T}} G(x)G^{\mathsf{T}}(x)\nabla J^*(x) \end{aligned} \tag{28}$$

with $J^*(0) = 0$.

Now, we give the following assumption, which is helpful to derive the optimal control with regard to system (2) and prove the stability of the closed-loop system.

*Assumption 2:* Consider system (2) with cost function (22) and the optimal feedback control function (26). Let $J_s(x)$ be a continuously differentiable Lyapunov function candidate formed as a polynomial and satisfying

$$\dot{J}_s(x) = (\nabla J_s(x))^{\mathsf{T}}\dot{x} = (\nabla J_s(x))^{\mathsf{T}}(f(x) + g(x)u^*) < 0 \tag{29}$$

where $\nabla J_s(x) = \partial J_s(x)/\partial x$. Assume there exists a positive definite matrix $\Lambda(x)$ such that the following relation holds:

$$(\nabla J_s(x))^{\mathsf{T}}(f(x) + g(x)u^*) = -(\nabla J_s(x))^{\mathsf{T}}\Lambda(x)\nabla J_s(x). \tag{30}$$

*Remark 3:* This is a common assumption that has been used in the literature, for instance [42], [45], and [46], to facilitate discussing the stability issue of closed-loop system. According to [45], we assume that the closed-loop dynamics with optimal control can be bounded by a function of system state on the compact set of this paper. Without loss of generality, we assume that $\|f(x) + g(x)u^*\| \le \eta\|\nabla J_s(x)\|$ with $\eta > 0$. Hence, we can further obtain $\|(\nabla J_s(x))^{\mathsf{T}}(f(x) + g(x)u^*)\| \le \eta\|\nabla J_s(x)\|^2$. Let $\lambda_m$ and $\lambda_M$ be the minimum and maximum eigenvalues of matrix $\Lambda(x)$, then we have

$$\lambda_m\|\nabla J_s(x)\|^2 \le (\nabla J_s(x))^{\mathsf{T}}\Lambda(x)\nabla J_s(x) \le \lambda_M\|\nabla J_s(x)\|^2. \tag{31}$$

Therefore, by noticing (29) and (31), we can conclude that the Assumption 2 is reasonable. Specifically, in this paper, $J_s(x)$ can be obtained by properly selecting a polynomial when implementing the ADP method.

The following theorem illustrates how to develop the optimal robust guaranteed cost control scheme for system (1).

*Theorem 2:* Consider system (1) with cost function (5). Suppose the modified HJB equation (28) has a continuously differentiable solution $J^*(x)$. Then, for any admissible control function $u$, the cost function (5) satisfies

$$\bar{J}(x_0, u) \le \Phi(u) \tag{32}$$

where

$$\Phi(u) \triangleq J^*(x_0) + \int_0^\infty (u - u^*)^{\mathsf{T}} R(u - u^*)\mathrm{d}\tau. \tag{33}$$

Moreover, the optimal robust guaranteed cost of the controlled uncertain nonlinear system is given by $\Phi^* = \Phi(u^*) = J^*(x_0)$. Accordingly, the optimal robust guaranteed cost control is given by $\bar{u}^* = u^*$.

*Proof:* For any admissible control function $u(x)$, the cost function (5) can be written as the following form:

$$\bar{J}(x_0, u) = J^*(x_0) + \int_0^\infty \big\{U(x,u) + \dot{J}^*(x)\big\}\mathrm{d}\tau. \tag{34}$$

Along the closed-loop trajectories of system (1) and according to (28), we find that

$$\begin{aligned} U(x,u)+&\dot{J}^*(x)\\ &= Q(x)+u^{\mathsf{T}}Ru+\left(\nabla J^*(x)\right)^{\mathsf{T}}\left(f(x)+g(x)u+\Delta f(x)\right)\\ &= u^{\mathsf{T}}Ru+\left(\nabla J^*(x)\right)^{\mathsf{T}}\left(g(x)u+\Delta f(x)\right)\\ &\quad -h^{\mathsf{T}}(\varphi(x))h(\varphi(x))-\frac{1}{4}\left(\nabla J^*(x)\right)^{\mathsf{T}}G(x)G^{\mathsf{T}}(x)\nabla J^*(x)\\ &\quad +\frac{1}{4}\left(\nabla J^*(x)\right)^{\mathsf{T}}g(x)R^{-1}g^{\mathsf{T}}(x)\nabla J^*(x). \end{aligned} \tag{35}$$

For the optimal cost function $J^*(x)$, in light of Lemma 1, we have the following inequality holds:

$$\begin{aligned} \left(\nabla J^*(x)\right)^{\mathsf{T}}\Delta f(x) \le\ & h^{\mathsf{T}}(\varphi(x))h(\varphi(x))\\ &+\frac{1}{4}\left(\nabla J^*(x)\right)^{\mathsf{T}}G(x)G^{\mathsf{T}}(x)\nabla J^*(x). \end{aligned} \tag{36}$$

Substituting (36) into (35), we can further obtain

$$\begin{aligned} U(x,u)+\dot{J}^*(x) \le\ & u^{\mathsf{T}}Ru+\left(\nabla J^*(x)\right)^{\mathsf{T}}g(x)u\\ &+\frac{1}{4}\left(\nabla J^*(x)\right)^{\mathsf{T}}g(x)R^{-1}g^{\mathsf{T}}(x)\nabla J^*(x). \end{aligned} \tag{37}$$

Considering the expression of the optimal control in (26), the (37) is in fact

$$U(x,u)+\dot{J}^*(x) \le\ (u-u^*)^{\mathsf{T}}R(u-u^*). \tag{38}$$

Thus, combining (34) with (38), we can find that

$$\bar{J}(x_0,u) \le J^*(x_0)+\int_0^\infty (u-u^*)^{\mathsf{T}}R(u-u^*)\mathrm{d}\tau \tag{39}$$

holds. Clearly, the optimal robust guaranteed cost can be obtained when setting $u = u^*$, i.e., $\Phi(u^*) = J^*(x_0)$. Furthermore, we can derive that $\Phi^* = \min_u \Phi(u) = J^*(x_0)$ and $\bar{u}^* = \arg\min_u \Phi(u) = u^*$. This completes the proof. ■

*Remark 4:* According to Theorem 2, the optimal robust guaranteed cost control of uncertain nonlinear system is transformed into the optimal control of nominal system, where the modified cost function is considered as the upper bound function. In other words, once the solution of the modified HJB equation (28) corresponding to nominal system (2) is derived, we can establish the optimal robust guaranteed cost control scheme of system (1).

## IV. Online HJB Solution of the Transformed Optimal Control Problem

For nonlinear system (2), the solution of optimal control problem can be obtained by solving the modified HJB equation (28) [9], [10], [12], [15], [38]. However, it is always difficult or even impossible to obtain the analytical solution. Thus, in many literature, the value iteration and policy iteration-based approaches are employed to get its approximate solution. The traditional ADP-based design methodology often utilizes critic network and action network without considering uncertainties of the controlled system. Besides, the design procedure is often performed with the requirement of an initial stabilizing control.

In this section, inspired by the excellent work of [39], [40], and [45], an improved online technique without utilizing the iterative strategy and an initial stabilizing control is developed by constructing a single network, namely, the critic network. Here, the ADP method is introduced to the framework of infinite horizon optimal robust guaranteed cost control of nonlinear systems with uncertainties.

### A. Neural Network Implementation

Assume that the cost function $V(x)$ is continuously differentiable. According to the universal approximation property of neural networks, $V(x)$ can be reconstructed by a single-layer neural network on a compact set $\Omega$ as

$$V(x) = \omega_c^{\mathsf{T}}\sigma_c(x)+\varepsilon_c(x) \tag{40}$$

where $\omega_c \in \mathbb{R}^l$ is the ideal weight, $\sigma_c(x) \in \mathbb{R}^l$ is the activation function, $l$ is the number of neurons in the hidden layer, and $\varepsilon_c(x)$ is the unknown approximation error of the neural network. Then

$$\nabla V(x) = \left(\nabla\sigma_c(x)\right)^{\mathsf{T}}\omega_c+\nabla\varepsilon_c(x) \tag{41}$$

is also unknown, where $\nabla\sigma_c(x) = \partial\sigma_c(x)/\partial x$ and $\nabla\varepsilon_c(x) = \partial\varepsilon_c(x)/\partial x$ are the gradient of the activation function and neural network approximation error, respectively. Based on (41), the Lyapunov equation (8) takes the following form:

$$\begin{aligned} 0 = \ & U(x,u)+\left(\omega_c^{\mathsf{T}}\nabla\sigma_c(x)+(\nabla\varepsilon_c(x))^{\mathsf{T}}\right)F(x,u)\\ &+h^{\mathsf{T}}(\varphi(x))h(\varphi(x))+\frac{1}{4}\left(\omega_c^{\mathsf{T}}\nabla\sigma_c(x)+(\nabla\varepsilon_c(x))^{\mathsf{T}}\right)\\ &\times G(x)G^{\mathsf{T}}(x)\left((\nabla\sigma_c(x))^{\mathsf{T}}\omega_c+\nabla\varepsilon_c(x)\right). \end{aligned} \tag{42}$$

Following the framework of [39], [40], and [45], we assume that the weight vector $\omega_c$, the gradient $\nabla\sigma_c(x)$, and the approximation error $\varepsilon_c(x)$ and its derivative $\nabla\varepsilon_c(x)$ are all bounded on a compact set $\Omega$.

Since the ideal weights are unknown, a critic neural network can be built in terms of the estimated weights as

$$\hat{V}(x) = \hat{\omega}_c^{\mathsf{T}}\sigma_c(x) \tag{43}$$

to approximate the cost function. Under the framework of ADP method, the selection of the activation function of the critic network is often a natural choice guided by engineering experience and intuition [37], [67]. Then, we have

$$\nabla\hat{V}(x) = (\nabla\sigma_c(x))^{\mathsf{T}}\hat{\omega}_c \tag{44}$$

where $\nabla\hat{V}(x) = \partial\hat{V}(x)/\partial x$.

According to (26) and (41), we have

$$u(x) = -\frac{1}{2}R^{-1}g^{\mathsf{T}}(x)\left((\nabla\sigma_c(x))^{\mathsf{T}}\omega_c+\nabla\varepsilon_c(x)\right) \tag{45}$$

which, in fact, represents the expression of optimal control $u^*(x)$ if the cost function in (40) is considered as the optimal one $J^*(x)$. Besides, in light of (26) and (44), the approximate control function can be given as

$$\hat{u}(x) = -\frac{1}{2}R^{-1}g^{\mathsf{T}}(x)(\nabla\sigma_c(x))^{\mathsf{T}}\hat{\omega}_c. \tag{46}$$

Applying (46) to system (2), the closed-loop system dynamics is expressed as

$$\dot{x} = f(x)-\frac{1}{2}g(x)R^{-1}g^{\mathsf{T}}(x)(\nabla\sigma_c(x))^{\mathsf{T}}\hat{\omega}_c. \tag{47}$$

Recalling the definition of the Hamiltonian function (24) and the modified HJB equation (25), we can easily obtain that $H(x, u^*, \nabla J^*) = 0$. The neural network expressions (41) and (45) imply that $u^*$ and $\nabla J^*$ can be formulated based on the ideal weight of the critic network, i.e., $\omega_c$. As a result, the Hamiltonian function becomes $H(x, \omega_c) = 0$, which specifically, can be written as

$$\begin{aligned} H(x, \omega_c) &= Q(x) + \omega_c^{\mathsf{T}} \nabla\sigma_c(x) f(x) \\ &\quad - \frac{1}{4}\omega_c^{\mathsf{T}} \nabla\sigma_c(x) g(x) R^{-1} g^{\mathsf{T}}(x) (\nabla\sigma_c(x))^{\mathsf{T}} \omega_c \\ &\quad + h^{\mathsf{T}}(\varphi(x)) h(\varphi(x)) \\ &\quad + \frac{1}{4}\omega_c^{\mathsf{T}} \nabla\sigma_c(x) G(x) G^{\mathsf{T}}(x) (\nabla\sigma_c(x))^{\mathsf{T}} \omega_c + e_{cH} \\ &= 0 \end{aligned} \tag{48}$$

where

$$\begin{aligned} e_{cH} &= (\nabla\varepsilon_c(x))^{\mathsf{T}} f(x) \\ &\quad - \frac{1}{2}(\nabla\varepsilon_c(x))^{\mathsf{T}} g(x) R^{-1} g^{\mathsf{T}}(x) (\nabla\sigma_c(x))^{\mathsf{T}} \omega_c \\ &\quad - \frac{1}{4}(\nabla\varepsilon_c(x))^{\mathsf{T}} g(x) R^{-1} g^{\mathsf{T}}(x) \nabla\varepsilon_c(x) \\ &\quad + \frac{1}{2}(\nabla\varepsilon_c(x))^{\mathsf{T}} G(x) G^{\mathsf{T}}(x) (\nabla\sigma_c(x))^{\mathsf{T}} \omega_c \\ &\quad + \frac{1}{4}(\nabla\varepsilon_c(x))^{\mathsf{T}} G(x) G^{\mathsf{T}}(x) \nabla\varepsilon_c(x). \end{aligned} \tag{49}$$

In (49), $e_{cH}$ denotes the residual error generated due to the neural network approximation.

Then, using the estimated weight vector, the approximate Hamiltonian function can be derived as

$$\begin{aligned} \hat{H}(x, \hat{\omega}_c) &= Q(x) + \hat{\omega}_c^{\mathsf{T}} \nabla\sigma_c(x) f(x) \\ &\quad - \frac{1}{4}\hat{\omega}_c^{\mathsf{T}} \nabla\sigma_c(x) g(x) R^{-1} g^{\mathsf{T}}(x) (\nabla\sigma_c(x))^{\mathsf{T}} \hat{\omega}_c \\ &\quad + h^{\mathsf{T}}(\varphi(x)) h(\varphi(x)) \\ &\quad + \frac{1}{4}\hat{\omega}_c^{\mathsf{T}} \nabla\sigma_c(x) G(x) G^{\mathsf{T}}(x) (\nabla\sigma_c(x))^{\mathsf{T}} \hat{\omega}_c. \end{aligned} \tag{50}$$

Letting $e_c = \hat{H}(x, \hat{\omega}_c) - H(x, \omega_c)$ and considering (48), we have $e_c = \hat{H}(x, \hat{\omega}_c)$. Let the weight estimation error of the critic network be

$$\tilde{\omega}_c = \omega_c - \hat{\omega}_c. \tag{51}$$

Then, based on (48), (50), and (51), we can obtain the formulation of $e_c$ in terms of $\tilde{\omega}_c$ as follows:

$$\begin{aligned} e_c &= \hat{H}(x, \hat{\omega}_c) - H(x, \omega_c) \\ &= -\tilde{\omega}_c^{\mathsf{T}} \nabla\sigma_c(x) f(x) \\ &\quad - \frac{1}{4}\tilde{\omega}_c^{\mathsf{T}} \nabla\sigma_c(x) g(x) R^{-1} g^{\mathsf{T}}(x) (\nabla\sigma_c(x))^{\mathsf{T}} \tilde{\omega}_c \\ &\quad + \frac{1}{2}\tilde{\omega}_c^{\mathsf{T}} \nabla\sigma_c(x) g(x) R^{-1} g^{\mathsf{T}}(x) (\nabla\sigma_c(x))^{\mathsf{T}} \omega_c \\ &\quad + \frac{1}{4}\tilde{\omega}_c^{\mathsf{T}} \nabla\sigma_c(x) G(x) G^{\mathsf{T}}(x) (\nabla\sigma_c(x))^{\mathsf{T}} \tilde{\omega}_c \\ &\quad - \frac{1}{2}\tilde{\omega}_c^{\mathsf{T}} \nabla\sigma_c(x) G(x) G^{\mathsf{T}}(x) (\nabla\sigma_c(x))^{\mathsf{T}} \omega_c - e_{cH}. \end{aligned} \tag{52}$$

For training the critic network, it is desired to design $\hat{\omega}_c$ to minimize the objective function

$$E_c = \frac{1}{2} e_c^{\mathsf{T}} e_c. \tag{53}$$

Here, the weights of the critic network are tuned based on the standard steepest descent algorithm with an additional term introduced to assure the boundedness of system state, that is

$$\begin{aligned} \dot{\hat{\omega}}_c &= -\alpha_c \left( \frac{\partial E_c}{\partial \hat{\omega}_c} \right) \\ &\quad + \frac{1}{2}\alpha_s \Pi(x, \hat{u}) \nabla\sigma_c(x) g(x) R^{-1} g^{\mathsf{T}}(x) \nabla J_s(x) \end{aligned} \tag{54}$$

where $\alpha_c > 0$ is the learning rate of the critic network, $\alpha_s > 0$ is the learning rate of the additional term, and $J_s(x)$ is the Lyapunov function candidate given in Assumption 2. In (54), the $\Pi(x, \hat{u})$ is the additional stabilizing term defined as

$$\Pi(x, \hat{u}) = \begin{cases} 0, \text{ if } \dot{J}_s(x) = (\nabla J_s(x))^{\mathsf{T}} F(x, \hat{u}) < 0 \\ 1, \text{ else.} \end{cases} \tag{55}$$

*Remark 5:* From the definition of the additional stabilizing term $\Pi(x, \hat{u})$, the second term in (54) is removed when the nonlinear system exhibits stable behavior. Hence, minimizing the approximate Hamiltonian becomes the primary objective of the weight update process. In contrast, when the controlled system exhibits signs of instability, i.e., $(\nabla J_s(x))^{\mathsf{T}} F(x, \hat{u}) > 0$, the second term of (54) is activated and is used to reinforce the training process of the weight vector until the system exhibits stable behavior. Hence, it can be seen that the term $\Pi(x, \hat{u})$ is defined based on the Lyapunov condition for stability. In this paper, we can obtain

$$\begin{aligned} -\frac{\partial\left((\nabla J_s(x))^{\mathsf{T}} F(x, \hat{u})\right)}{\partial \hat{\omega}_c} &= -\left( \frac{\partial \hat{u}}{\partial \hat{\omega}_c} \right)^{\mathsf{T}} \frac{\partial\left((\nabla J_s(x))^{\mathsf{T}} F(x, \hat{u})\right)}{\partial \hat{u}} \\ &= \frac{1}{2}\nabla\sigma_c(x) g(x) R^{-1} g^{\mathsf{T}}(x) \nabla J_s(x) \end{aligned} \tag{56}$$

which shows that the reinforced training process is carried out along the negative gradient direction of $(\nabla J_s(x))^{\mathsf{T}} F(x, \hat{u})$. When the case $(\nabla J_s(x))^{\mathsf{T}} F(x, \hat{u}) > 0$ occurs, the reinforced training process reduces the value of $(\nabla J_s(x))^{\mathsf{T}} F(x, \hat{u})$ to make it negative. To summarize, the second term in (54) is chosen for ensuring the stability of closed-loop system, and meanwhile, for facilitating the stability proof given in the sequel. Actually, it is in this sense that the requirement of an initial stabilizing control is relaxed. Therefore, the weight vector of critic network is initialized to zero during the neural network implementation process.

The structural diagram of the implementation process using neural network is displayed in Fig. 1.

Next, we will find the dynamics of the weight estimation error $\tilde{\omega}_c$. According to (50), we have

$$\begin{aligned} \frac{\partial e_c}{\partial \hat{\omega}_c} &= \nabla\sigma_c(x) f(x) \\ &\quad - \frac{1}{2}\nabla\sigma_c(x) g(x) R^{-1} g^{\mathsf{T}}(x) (\nabla\sigma_c(x))^{\mathsf{T}} \hat{\omega}_c \\ &\quad + \frac{1}{2}\nabla\sigma_c(x) G(x) G^{\mathsf{T}}(x) (\nabla\sigma_c(x))^{\mathsf{T}} \hat{\omega}_c. \end{aligned} \tag{57}$$

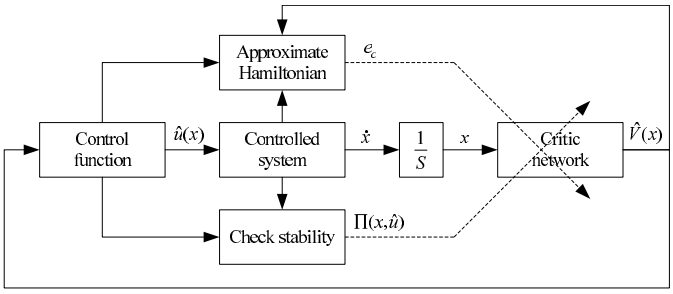



Fig. 1. Structural diagram of neural network implementation (the solid line represents the signal and the dashed line represents the back-propagating path).

In light of (51), (53), and (54), the dynamics of the weight estimation error is

$$\begin{aligned}\dot{\tilde{\omega}}_c &= -\dot{\hat{\omega}}_c \\ &= \alpha_c e_c\left(\frac{\partial e_c}{\partial \hat{\omega}_c}\right) \\ &\quad - \frac{1}{2}\alpha_s \Pi(x,\hat{u})\nabla\sigma_c(x)g(x)R^{-1}g^{\mathsf{T}}(x)\nabla J_s(x). \quad (58)\end{aligned}$$

Then, combining (51), (52), and (57), the error dynamics (58) becomes

$$\begin{aligned}\dot{\tilde{\omega}}_c = \alpha_c\Bigg( &- \tilde{\omega}_c^{\mathsf{T}}\nabla\sigma_c(x)f(x) \\ &- \frac{1}{4}\tilde{\omega}_c^{\mathsf{T}}\nabla\sigma_c(x)g(x)R^{-1}g^{\mathsf{T}}(x)(\nabla\sigma_c(x))^{\mathsf{T}}\tilde{\omega}_c \\ &+ \frac{1}{2}\tilde{\omega}_c^{\mathsf{T}}\nabla\sigma_c(x)g(x)R^{-1}g^{\mathsf{T}}(x)(\nabla\sigma_c(x))^{\mathsf{T}}\omega_c \\ &+ \frac{1}{4}\tilde{\omega}_c^{\mathsf{T}}\nabla\sigma_c(x)G(x)G^{\mathsf{T}}(x)(\nabla\sigma_c(x))^{\mathsf{T}}\tilde{\omega}_c \\ &- \frac{1}{2}\tilde{\omega}_c^{\mathsf{T}}\nabla\sigma_c(x)G(x)G^{\mathsf{T}}(x)(\nabla\sigma_c(x))^{\mathsf{T}}\omega_c - e_{cH}\Bigg) \\ \times\Bigg(&\nabla\sigma_c(x)f(x) \\ &- \frac{1}{2}\nabla\sigma_c(x)g(x)R^{-1}g^{\mathsf{T}}(x)(\nabla\sigma_c(x))^{\mathsf{T}}\omega_c \\ &+ \frac{1}{2}\nabla\sigma_c(x)g(x)R^{-1}g^{\mathsf{T}}(x)(\nabla\sigma_c(x))^{\mathsf{T}}\tilde{\omega}_c \\ &+ \frac{1}{2}\nabla\sigma_c(x)G(x)G^{\mathsf{T}}(x)(\nabla\sigma_c(x))^{\mathsf{T}}\omega_c \\ &- \frac{1}{2}\nabla\sigma_c(x)G(x)G^{\mathsf{T}}(x)(\nabla\sigma_c(x))^{\mathsf{T}}\tilde{\omega}_c\Bigg) \\ &- \frac{1}{2}\alpha_s\Pi(x,\hat{u})\nabla\sigma_c(x)g(x)R^{-1}g^{\mathsf{T}}(x)\nabla J_s(x). \quad (59)\end{aligned}$$

In the following, the stability analysis of the neural-network-based feedback control system is presented by using the Lyapunov theory.

### B. Stability Analysis

In this section, the error dynamics of the critic network and the closed-loop system based on the approximate optimal control will be proved to be UUB.

*Theorem 3:* Consider the nonlinear system given by (2). Let the control input be provided by (46) and the weights of the critic network be tuned by (54). Then, the state $x$ of the closed-loop system and the weight estimation error $\tilde{\omega}_c$ of the critic network are UUB.

*Proof:* See the Appendix. ■

*Corollary 1:* The approximate control input $\hat{u}$ in (46) converges to a neighborhood of optimal control input $u^*$ with finite bound.

*Proof:* According to (45) and (46), we have

$$u^* - \hat{u} = -\frac{1}{2}R^{-1}g^{\mathsf{T}}(x)(\nabla\sigma(x))^{\mathsf{T}}\tilde{\omega}_c - \frac{1}{2}R^{-1}g^{\mathsf{T}}(x)\nabla\varepsilon_c(x). \quad (60)$$

In light of Theorem 3, we have $\|\tilde{\omega}_c\| < \mathscr{A}$, where $\mathscr{A}$ is defined in the Appendix. Then, the terms $R^{-1}g^{\mathsf{T}}(x)(\nabla\sigma(x))^{\mathsf{T}}\tilde{\omega}_c$ and $R^{-1}g^{\mathsf{T}}(x)\nabla\varepsilon_c(x)$ are all bounded. Thus, we can further determine that

$$\begin{aligned}\|u^* - \hat{u}\| &\le \frac{1}{2}R_M^{-1}g_M\sigma_{dM}\mathscr{A} + \frac{1}{2}R_M^{-1}g_M\lambda_{10} \\ &\triangleq \varepsilon_u \quad (61)\end{aligned}$$

where $\lambda_{10}$ is given in the Appendix and $\varepsilon_u$ is the finite bound. This completes the proof. ■

### C. Design Procedure of the Optimal Robust Guaranteed Cost Control

For continuous-time uncertain nonlinear systems (1) satisfying (3) and (4), we summarize the design procedure of optimal robust guaranteed cost control as follows.

- Step 1: Select $G(x)$ and $\varphi(x)$, determine $h(\varphi(x))$, and conduct the problem transformation based on the bounded function $\Gamma(x)$.
- Step 2: Choose the Lyapunov function candidate $J_s(x)$, construct a critic network as (43), and set its initial weights to zero.
- Step 3: Solve the transformed optimal control problem via online solution of the modified HJB equation, using the expressions of approximate control function (46), approximate Hamiltonian function (50), and weights update criterion (54).
- Step 4: Derive the optimal robust guaranteed cost and optimal robust guaranteed cost control of original uncertain nonlinear system based on the converged weights of critic network.

*Remark 6:* It is observed from (43) and (50), both the approximate cost function and the approximate Hamiltonian become zero when $\|x\| = 0$. In this case, we can find that $\dot{\hat{\omega}}_c = 0$. Thus, when the system state converges to zero, the weights of the critic network are no longer updated. This can be viewed as a persistency of excitation requirement of the neural network inputs. In other words, the system state must be persistently exciting long enough in order to ensure the critic network to learn the optimal cost function as accurately as possible. In this paper, the persistency of excitation condition is satisfied by adding an exploration noise to the control input. The condition can be removed once the weights of the critic network converge to their target values. Actually, it is for this reason that there always exists a tradeoff between computational accuracy and time consumption for practical realization.

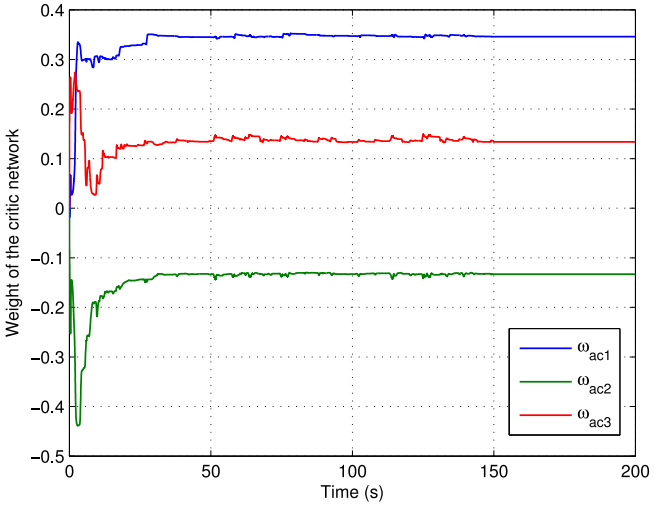



Fig. 2. Convergence of weight vector of the critic network ($\omega_{ac1}$, $\omega_{ac2}$, and $\omega_{ac3}$ represents $\hat{\omega}_{c1}$, $\hat{\omega}_{c2}$, and $\hat{\omega}_{c3}$, respectively).

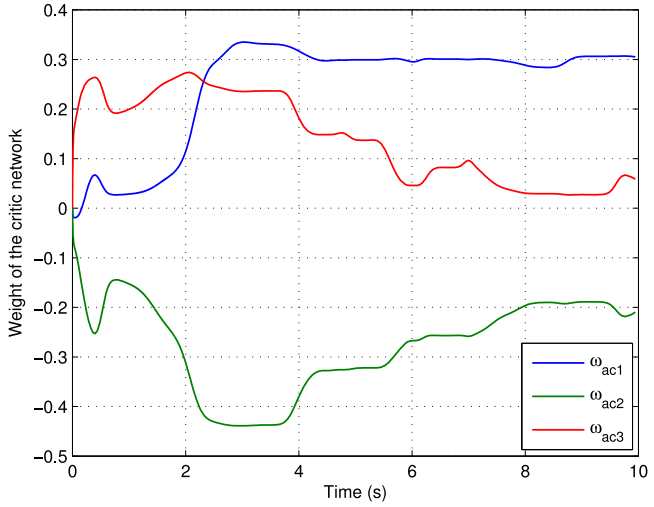



Fig. 3. Updating process of weight vector during the first 10 s ($\omega_{ac1}$, $\omega_{ac2}$, and $\omega_{ac3}$ represent $\hat{\omega}_{c1}$, $\hat{\omega}_{c2}$, and $\hat{\omega}_{c3}$, respectively).

## V. Simulation Studies

In this section, two simulation examples are provided to demonstrate the effectiveness of the optimal robust guaranteed cost control strategy derived based on the online HJB solution. We first consider a continuous-time linear system and then a nonlinear system, both with system uncertainty.

*Example 1:* Consider the continuous-time linear system

$$\dot{x} = \begin{bmatrix} -1 & -2 \\ 1 & -4 \end{bmatrix} x + \begin{bmatrix} 1 \\ -3 \end{bmatrix} u + \Delta f(x) \tag{62}$$

where $x = [x_1, x_2]^{\mathsf{T}}$ and $\Delta f(x) = [px_1 \sin x_2, 0]^{\mathsf{T}}$ with $p \in [-0.5, 0.5]$. According to the form of system uncertainty, we choose $G(x) = [1, 0]^{\mathsf{T}}$ and $\varphi(x) = x$. Then, we have $d(\varphi(x)) = px_1 \sin x_2$. Besides, we select $h(\varphi(x)) = 0.5x_1 \sin x_2$.

In this example, we first choose $Q(x) = x^{\mathsf{T}}x$, $R = I$, where $I$ is an identity matrix with suitable dimension. In order to solve the transformed optimal control problem, a critic network is constructed to approximate the modified cost function as

$$\hat{V}(x) = \hat{\omega}_{c1}x_1^2 + \hat{\omega}_{c2}x_1x_2 + \hat{\omega}_{c3}x_2^2. \tag{63}$$

Let the initial state of the controlled plant be $x_0 = [1, -1]^{\mathsf{T}}$. Select the Lyapunov function candidate of the weights tuning criterion as $J_s(x) = (1/2)x^{\mathsf{T}}x$. Let the learning rate of the critic network and the additional term be $\alpha_c = 0.8$ and $\alpha_s = 0.5$, respectively. During the neural network implementation process, we bring in an exploration noise $\mathcal{N}(t) = \sin^2(t)\cos(t) + \sin^2(2t)\cos(0.1t) + \sin^2(-1.2t)\cos(0.5t) + \sin^5(t) + \sin^2(1.12t) + \cos(2.4t)\sin^3(2.4t)$ to satisfy the persistency of excitation condition. It is introduced into the control input and thus affects the system state. After a learning session, the weights of the critic network converge to $[0.3461, -0.1330, 0.1338]^{\mathsf{T}}$ as shown in Fig. 2. Here, it is important to note that the initial weights of the critic network are all set as zero, which implies that no initial stabilizing control is needed for implementing the control strategy. This can be verified by observing Fig. 3, which displays the updating process of weight vector during the first 10 s.

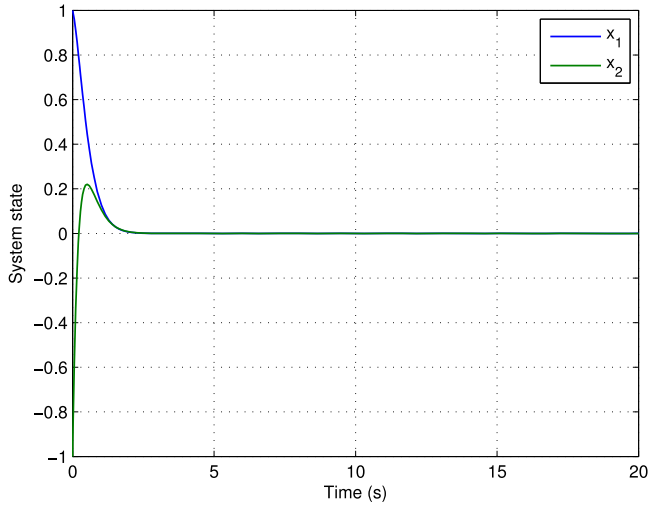



Fig. 4. System state ($p = 0.5$).

Based on the converged weight vector, the optimal robust guaranteed cost of the controlled system is $\Phi(u^*) = J^*(x_0) = 0.6129$. Next, the scalar parameter $p = 0.5$ is chosen for evaluating the control performance. Under the action of the obtained control function, the system trajectory during the first 20 s is presented in Fig. 4, which shows the good performance of the control approach.

Next, we set $Q(x) = 8x^{\mathsf{T}}x$, $R = 5I$, and conduct the neural network implementation again by increasing the learning rates of the critic network and the additional term properly. In this case, the weights of the critic network converge to $[5.4209, -3.5088, 1.2605]^{\mathsf{T}}$, which is depicted in Fig. 5. Similarly, the system trajectory during the first 20 s when choosing $p = 0.5$ is displayed in Fig. 6. These simulation results show that the parameters $Q(x)$ and $R$ play an important role in the design process. In addition, the power of the present control technique is demonstrated again.

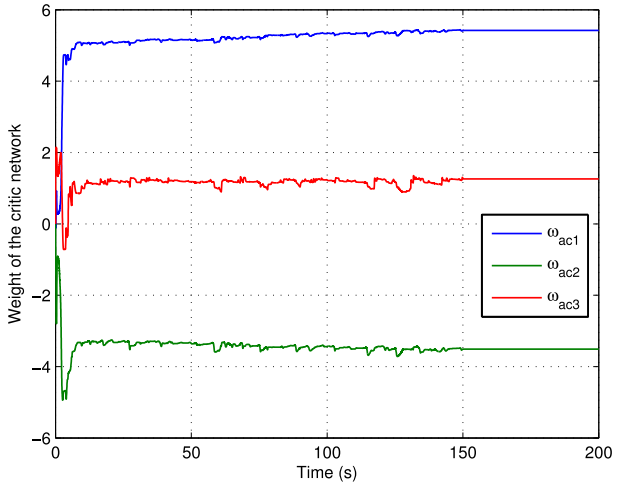



Fig. 5. Convergence of weight vector of the critic network ($\omega_{ac1}$, $\omega_{ac2}$, and $\omega_{ac3}$ represents $\hat{\omega}_{c1}$, $\hat{\omega}_{c2}$, and $\hat{\omega}_{c3}$, respectively).

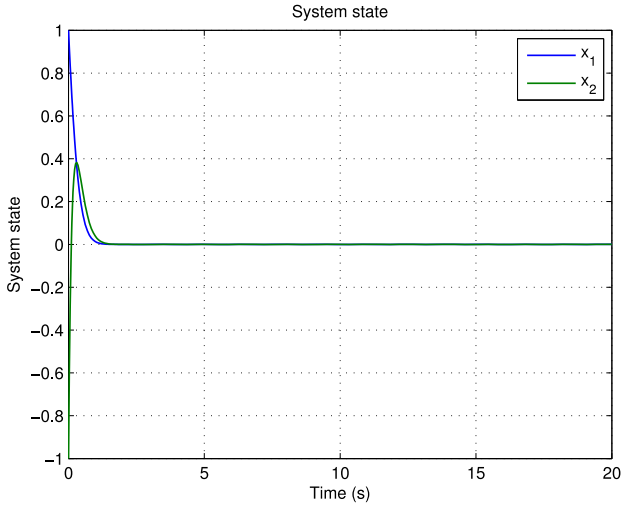



Fig. 6. System state ($p = 0.5$).

*Example 2:* Consider the following continuous-time nonlinear system:

$$\dot{x} = \begin{bmatrix} -x_1 + x_2 \\ 0.1x_1 - x_2 - x_1x_3 \\ x_1x_2 - x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} u + \Delta f(x) \qquad (64)$$

where $x = [x_1, x_2, x_3]^{\mathsf{T}}$, $\Delta f(x) = [0, 0, px_1 \sin x_2 \cos x_3]^{\mathsf{T}}$, and $p \in [-1, 1]$. Similarly, if we choose $G(x) = [0, 0, 1]^{\mathsf{T}}$ and $\varphi(x) = x$ based on the form of system uncertainty, then $d(\varphi(x)) = px_1 \sin x_2 \cos x_3$. Clearly, we can select $h(\varphi(x)) = x_1 \sin x_2 \cos x_3$.

In this example, $Q(x)$ and $R$ are chosen the same as the first case of Example 1. However, the critic network is constructed using the following form:

$$\begin{aligned} \hat{V}(x) = {} & \hat{\omega}_{c1}x_1^2 + \hat{\omega}_{c2}x_2^2 + \hat{\omega}_{c3}x_3^2 + \hat{\omega}_{c4}x_1x_2 + \hat{\omega}_{c5}x_1x_3 \\ & + \hat{\omega}_{c6}x_2x_3 + \hat{\omega}_{c7}x_1^4 + \hat{\omega}_{c8}x_2^4 + \hat{\omega}_{c9}x_3^4 \\ & + \hat{\omega}_{c10}x_1^2x_2^2 + \hat{\omega}_{c11}x_1^2x_3^2 + \hat{\omega}_{c12}x_2^2x_3^2 \\ & + \hat{\omega}_{c13}x_1^2x_2x_3 + \hat{\omega}_{c14}x_1x_2^2x_3 + \hat{\omega}_{c15}x_1x_2x_3^2 \\ & + \hat{\omega}_{c16}x_1^3x_2 + \hat{\omega}_{c17}x_1^3x_3 + \hat{\omega}_{c18}x_1x_2^3 \\ & + \hat{\omega}_{c19}x_2^3x_3 + \hat{\omega}_{c20}x_1x_3^3 + \hat{\omega}_{c21}x_2x_3^3. \end{aligned} \qquad (65)$$

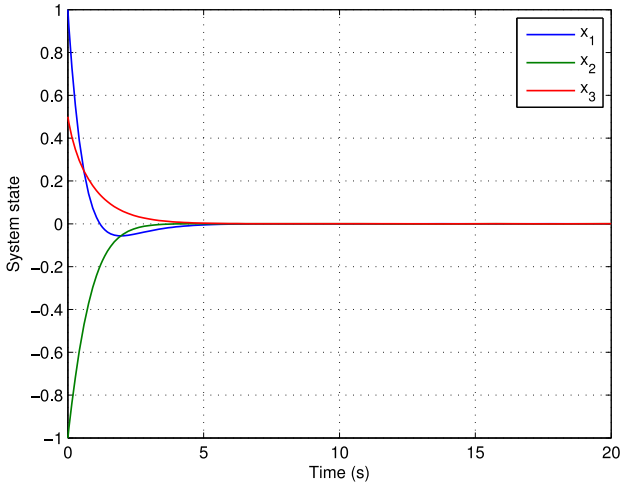



Fig. 7. System state ($p = -1$).

Here, let the initial state of the controlled system be $x_0 = [1, -1, 0.5]^{\mathsf{T}}$. Besides, let the learning rate of the critic network and the additional term be $\alpha_c = 0.3$ and $\alpha_s = 0.5$, respectively. Same as above, an exploration noise is added to satisfy the persistency of excitation condition during the neural network implementation process. Besides, all the elements of the weight vector of critic network are initialized to zero. After a sufficient learning session, the weights of the critic network converge to [0.4759, 0.5663, 0.1552, 0.4214, 0.0911, 0.0375, 0.0886, −0.0099, 0.0986, 0.1539, 0.0780, −0.0192, −0.1335, −0.0052, −0.0639, −0.1583, 0.0456, 0.0576, −0.0535, 0.0885, −0.0227]$^{\mathsf{T}}$.

Similarly, the optimal robust guaranteed cost of the nonlinear system is $\Phi(u^*) = J^*(x_0) = 1.1841$. In this example, the scalar parameter $p = -1$ is chosen for evaluating the robust control performance. The system trajectory is depicted in Fig. 7 when applying the obtained control to system (64) for 20 s. These simulation results verify the effectiveness of the developed control approach.

## VI. Conclusion

A novel strategy is developed to derive the optimal robust guaranteed cost control of uncertain nonlinear systems. This is accomplished by properly modifying the cost function to account for system uncertainty, so that the solution of the transformed optimal control problem serves as the optimal robust guaranteed cost of the original system. A critic network is constructed to solve the modified HJB equation online. Two simulation examples are presented to reinforce the theoretical results as well.

As for future works, we will study the optimal robust guaranteed cost control of uncertain nonlinear systems with constrained inputs based on single network ADP approach. In this case, we let all the elements of control input $u(t)$ in system (1) have lower and upper bounds, i.e., $u_{i\min} \le u_i \le u_{i\max}$, $i = 1, 2, \ldots, m$, where $u_{i\min}$ and $u_{i\max}$ are constants. Besides, how to deal with the problem when the dynamic knowledge of nominal system is unknown serves as another interesting direction of future research. Under such circumstance, functions $f(x)$ and $g(x)$ are assumed to be unknown, hence the

system identification will be employed by constructing neural networks. Remarkably, as an important part of machine learning community, reinforcement learning is characterized by finding optimal actions in unknown environment [10], [11]. Thus, it is of great significance to use more advanced idea of reinforcement learning to handle the optimal control problems under uncertain and unknown environment. Moreover, how to relax the restrictive condition of system uncertainty is also one of the directions of our future works. Additionally, the inverse optimal control [68], [69], which is featured by the fact that the meaningful cost function is determined from the stabilizing feedback control, serves as another effective strategy aimed at circumventing the challenging task of solving the HJB equation. Thus, the inverse optimal control approach will also be helpful for our future study.

## APPENDIX

*Proof of Theorem 3:* We choose the following Lyapunov function candidate:

$$L(t) = \frac{1}{2\alpha_c}\tilde{\omega}_c^{\mathsf{T}}\tilde{\omega}_c + \frac{\alpha_s}{\alpha_c}J_s(x) \tag{A.1}$$

where $J_s(x)$ is presented in Assumption 2. The derivative of the Lyapunov function candidate (A.1) with respect to time along the dynamics of (47) and (59) is

$$\dot{L}(t) = \frac{1}{\alpha_c}\tilde{\omega}_c^{\mathsf{T}}\dot{\tilde{\omega}}_c + \frac{\alpha_s}{\alpha_c}(\nabla J_s(x))^{\mathsf{T}}\dot{x}. \tag{A.2}$$

Substituting (47) and (59) into (A.2), we obtain

$$\begin{aligned}\dot{L}(t) = \tilde{\omega}_c^{\mathsf{T}}\Bigg( &-\tilde{\omega}_c^{\mathsf{T}}\nabla\sigma_c(x)f(x)\\ &-\frac{1}{4}\tilde{\omega}_c^{\mathsf{T}}\nabla\sigma_c(x)g(x)R^{-1}g^{\mathsf{T}}(x)(\nabla\sigma_c(x))^{\mathsf{T}}\tilde{\omega}_c\\ &+\frac{1}{2}\tilde{\omega}_c^{\mathsf{T}}\nabla\sigma_c(x)g(x)R^{-1}g^{\mathsf{T}}(x)(\nabla\sigma_c(x))^{\mathsf{T}}\omega_c\\ &+\frac{1}{4}\tilde{\omega}_c^{\mathsf{T}}\nabla\sigma_c(x)G(x)G^{\mathsf{T}}(x)(\nabla\sigma_c(x))^{\mathsf{T}}\tilde{\omega}_c\\ &-\frac{1}{2}\tilde{\omega}_c^{\mathsf{T}}\nabla\sigma_c(x)G(x)G^{\mathsf{T}}(x)(\nabla\sigma_c(x))^{\mathsf{T}}\omega_c - e_{cH}\Bigg)\\ &\times\Bigg(\nabla\sigma_c(x)f(x)\\ &-\frac{1}{2}\nabla\sigma_c(x)g(x)R^{-1}g^{\mathsf{T}}(x)(\nabla\sigma_c(x))^{\mathsf{T}}\omega_c\\ &+\frac{1}{2}\nabla\sigma_c(x)g(x)R^{-1}g^{\mathsf{T}}(x)(\nabla\sigma_c(x))^{\mathsf{T}}\tilde{\omega}_c\\ &+\frac{1}{2}\nabla\sigma_c(x)G(x)G^{\mathsf{T}}(x)(\nabla\sigma_c(x))^{\mathsf{T}}\omega_c\\ &-\frac{1}{2}\nabla\sigma_c(x)G(x)G^{\mathsf{T}}(x)(\nabla\sigma_c(x))^{\mathsf{T}}\tilde{\omega}_c\Bigg)\\ &-\frac{\alpha_s}{2\alpha_c}\Pi(x,\hat{u})\tilde{\omega}_c^{\mathsf{T}}\nabla\sigma_c(x)g(x)R^{-1}g^{\mathsf{T}}(x)\nabla J_s(x)\\ &+\frac{\alpha_s}{\alpha_c}(\nabla J_s(x))^{\mathsf{T}}\dot{x}.\end{aligned} \tag{A.3}$$

For simplicity, we denote

$$A = \nabla\sigma_c(x)g(x)R^{-1}g^{\mathsf{T}}(x)(\nabla\sigma_c(x))^{\mathsf{T}} \tag{A.4}$$

$$B = \nabla\sigma_c(x)G(x)G^{\mathsf{T}}(x)(\nabla\sigma_c(x))^{\mathsf{T}}. \tag{A.5}$$

Then, (A.3) becomes

$$\begin{aligned}\dot{L}(t) = &-\Bigg(\tilde{\omega}_c^{\mathsf{T}}\nabla\sigma_c(x)f(x) + \frac{1}{4}\tilde{\omega}_c^{\mathsf{T}}A\tilde{\omega}_c - \frac{1}{2}\tilde{\omega}_c^{\mathsf{T}}A\omega_c\\ &-\frac{1}{4}\tilde{\omega}_c^{\mathsf{T}}B\tilde{\omega}_c + \frac{1}{2}\tilde{\omega}_c^{\mathsf{T}}B\omega_c + e_{cH}\Bigg)\\ &\times\Bigg(\tilde{\omega}_c^{\mathsf{T}}\nabla\sigma_c(x)f(x) + \frac{1}{2}\tilde{\omega}_c^{\mathsf{T}}A\tilde{\omega}_c - \frac{1}{2}\tilde{\omega}_c^{\mathsf{T}}A\omega_c\\ &-\frac{1}{2}\tilde{\omega}_c^{\mathsf{T}}B\tilde{\omega}_c + \frac{1}{2}\tilde{\omega}_c^{\mathsf{T}}B\omega_c\Bigg)\\ &-\frac{\alpha_s}{2\alpha_c}\Pi(x,\hat{u})\tilde{\omega}_c^{\mathsf{T}}\nabla\sigma_c(x)g(x)R^{-1}g^{\mathsf{T}}(x)\nabla J_s(x)\\ &+\frac{\alpha_s}{\alpha_c}(\nabla J_s(x))^{\mathsf{T}}\dot{x}.\end{aligned} \tag{A.6}$$

Considering (47), we have

$$\begin{aligned}\dot{L}(t) = &-\Bigg(\tilde{\omega}_c^{\mathsf{T}}\nabla\sigma_c(x)\dot{x} - \frac{1}{4}\tilde{\omega}_c^{\mathsf{T}}A\tilde{\omega}_c - \frac{1}{4}\tilde{\omega}_c^{\mathsf{T}}B\tilde{\omega}_c\\ &+\frac{1}{2}\tilde{\omega}_c^{\mathsf{T}}B\omega_c + e_{cH}\Bigg)\\ &\times\Bigg(\tilde{\omega}_c^{\mathsf{T}}\nabla\sigma_c(x)\dot{x} - \frac{1}{2}\tilde{\omega}_c^{\mathsf{T}}B\tilde{\omega}_c + \frac{1}{2}\tilde{\omega}_c^{\mathsf{T}}B\omega_c\Bigg)\\ &-\frac{\alpha_s}{2\alpha_c}\Pi(x,\hat{u})\tilde{\omega}_c^{\mathsf{T}}\nabla\sigma_c(x)g(x)R^{-1}g^{\mathsf{T}}(x)\nabla J_s(x)\\ &+\frac{\alpha_s}{\alpha_c}(\nabla J_s(x))^{\mathsf{T}}\dot{x}.\end{aligned} \tag{A.7}$$

Noticing that $\dot{x}^* = f(x) + g(x)u^*$, where $u^*$ is given by (45), we can further obtain that

$$\begin{aligned}\dot{L}(t) = &-\Bigg(\tilde{\omega}_c^{\mathsf{T}}\nabla\sigma_c(x)\dot{x}^* + \frac{1}{4}\tilde{\omega}_c^{\mathsf{T}}A\tilde{\omega}_c\\ &+\frac{1}{2}\tilde{\omega}_c^{\mathsf{T}}\nabla\sigma_c(x)g(x)R^{-1}g^{\mathsf{T}}(x)\nabla\varepsilon_c(x)\\ &-\frac{1}{4}\tilde{\omega}_c^{\mathsf{T}}B\tilde{\omega}_c + \frac{1}{2}\tilde{\omega}_c^{\mathsf{T}}B\omega_c + e_{cH}\Bigg)\\ &\times\Bigg(\tilde{\omega}_c^{\mathsf{T}}\nabla\sigma_c(x)\dot{x}^* + \frac{1}{2}\tilde{\omega}_c^{\mathsf{T}}A\tilde{\omega}_c\\ &+\frac{1}{2}\tilde{\omega}_c^{\mathsf{T}}\nabla\sigma_c(x)g(x)R^{-1}g^{\mathsf{T}}(x)\nabla\varepsilon_c(x)\\ &-\frac{1}{2}\tilde{\omega}_c^{\mathsf{T}}B\tilde{\omega}_c + \frac{1}{2}\tilde{\omega}_c^{\mathsf{T}}B\omega_c\Bigg)\\ &-\frac{\alpha_s}{2\alpha_c}\Pi(x,\hat{u})\tilde{\omega}_c^{\mathsf{T}}\nabla\sigma_c(x)g(x)R^{-1}g^{\mathsf{T}}(x)\nabla J_s(x)\\ &+\frac{\alpha_s}{\alpha_c}(\nabla J_s(x))^{\mathsf{T}}\dot{x}.\end{aligned} \tag{A.8}$$

As in [45], we assume that $\lambda_{1m} > 0$ and $\lambda_{1M} > 0$ are the lower and upper bounds of the norm of matrix $A$. Similarly, assume that $\lambda_{2m} > 0$ and $\lambda_{2M} > 0$ are the lower and upper bounds of the norm of matrix $B$. Assume that $\|R^{-1}\| \le R_M^{-1}$, $\|g(x)\| \le g_M$, $\|\nabla\sigma(x)\| \le \sigma_{dM}$, $\|B\omega_c\| \le \lambda_4$, $\|\nabla\varepsilon_c(x)\| \le \lambda_{10}$, and $\|e_{cH}\| \le \lambda_{12}$, where $R_M^{-1}$, $g_M$, $\sigma_{dM}$, $\lambda_4$, $\lambda_{10}$, and $\lambda_{12}$ are positive constants. In addition, assume that $\|\nabla\sigma_c(x)\dot{x}^*\| \le \lambda_3$, where $\lambda_3$ is a positive constant. Let $\lambda_5 = (\sqrt{6}/2)\lambda_{12}$, $\lambda_9 = g_M^2R_M^{-1}$, and $\lambda_{11} = \sigma_{dM}g_M^2R_M^{-1}\lambda_{10}$, then $\|g(x)R^{-1}g^{\mathsf{T}}(x)\| \le \lambda_9$ and $\|\nabla\sigma(x)g(x)R^{-1}g^{\mathsf{T}}(x)\nabla\varepsilon_c(x)\| \le \lambda_{11}$. Using the relations

$$ab = \frac{1}{2}\left(-\left(\phi_+ a - \frac{b}{\phi_+}\right)^2 + \phi_+^2a^2 + \frac{b^2}{\phi_+^2}\right) \tag{A.9}$$

$$-ab = -\frac{1}{2}\left(\left(\phi_- a + \frac{b}{\phi_-}\right)^2 - \phi_-^2 a^2 - \frac{b^2}{\phi_-^2}\right) \quad \text{(A.10)}$$

we have

$$\begin{aligned}
&-\frac{3}{4}(\tilde{\omega}_c^{\mathsf{T}}\nabla\sigma_c(x)\dot{x}^*)(\tilde{\omega}_c^{\mathsf{T}}A\tilde{\omega}_c)\\
&\quad = -\frac{3}{8}\left(\left(\phi_1\tilde{\omega}_c^{\mathsf{T}}\nabla\sigma_c(x)\dot{x}^* + \frac{\tilde{\omega}_c^{\mathsf{T}}A\tilde{\omega}_c}{\phi_1}\right)^2\right.\\
&\qquad\qquad \left. - \phi_1^2(\tilde{\omega}_c^{\mathsf{T}}\nabla\sigma_c(x)\dot{x}^*)^2 - \frac{(\tilde{\omega}_c^{\mathsf{T}}A\tilde{\omega}_c)^2}{\phi_1^2}\right)\\
&\quad \le \frac{3}{8}\left(\phi_1^2(\tilde{\omega}_c^{\mathsf{T}}\nabla\sigma_c(x)\dot{x}^*)^2 + \frac{(\tilde{\omega}_c^{\mathsf{T}}A\tilde{\omega}_c)^2}{\phi_1^2}\right)\\
&\quad \le \frac{3}{8\phi_1^2}\lambda_{1M}^2\|\tilde{\omega}_c\|^4 + \frac{3}{8}\phi_1^2\lambda_3^2\|\tilde{\omega}_c\|^2 \qquad \text{(A.11)}
\end{aligned}$$

where $\phi_+$, $\phi_-$, and $\phi_1$ are nonzero constants. Other terms of (A.8) can be handled the same way. Then, we can find that

$$\begin{aligned}
\dot{L}(t) \le\ & -\lambda_7\|\tilde{\omega}_c\|^4 + \lambda_8\|\tilde{\omega}_c\|^2 + \lambda_5^2\\
& - \frac{\alpha_s}{2\alpha_c}\Pi(x,\hat{u})\tilde{\omega}_c^{\mathsf{T}}\nabla\sigma_c(x)g(x)R^{-1}g^{\mathsf{T}}(x)\nabla J_s(x)\\
& + \frac{\alpha_s}{\alpha_c}(\nabla J_s(x))^{\mathsf{T}}\dot{x} \qquad \text{(A.12)}
\end{aligned}$$

where

$$\begin{aligned}
\lambda_7 =\ & \frac{1}{8}\lambda_{1m}^2 + \frac{1}{8}\lambda_{2m}^2 - \frac{3}{8\phi_1^2}\lambda_{1M}^2 - \frac{3}{8\phi_2^2}\lambda_{2M}^2\\
& - \frac{3}{16}\phi_3^2\lambda_{1M}^2 - \frac{3}{16}\phi_4^2\lambda_{1M}^2 - \frac{3}{16}\phi_5^2\lambda_{11}^2 - \frac{3}{16}\phi_6^2\lambda_{2M}^2 \qquad \text{(A.13)}
\end{aligned}$$

$$\begin{aligned}
\lambda_8 =\ & \frac{3}{8}\phi_1^2\lambda_3^2 + \frac{3}{8}\phi_2^2\lambda_3^2 + \frac{3}{16\phi_3^2}\lambda_{11}^2\\
& + \frac{3}{16\phi_4^2}\lambda_4^2 + \frac{3}{16\phi_5^2}\lambda_{11}^2 + \frac{3}{16\phi_6^2}\lambda_4^2 \qquad \text{(A.14)}
\end{aligned}$$

and $\phi_i$, $i = 1, 2, \ldots, 6$, are nonzero constants chosen for the design purpose. Note that under the action of $\phi_i$, $i = 1, 2, \ldots, 6$, the relation $\lambda_7 > 0$ can be guaranteed.

In the following, the cases of $\Pi(x,\hat{u}) = 0$ and $\Pi(x,\hat{u}) = 1$ will be considered, respectively.

*Case 1:* $\Pi(x,\hat{u}) = 0$. Since $(\nabla J_s(x))^{\mathsf{T}}\dot{x} < 0$, we have $-(\nabla J_s(x))^{\mathsf{T}}\dot{x} > 0$. According to the density property of real numbers, there exists a positive constant $\lambda_6$ such that $0 < \lambda_6\|\nabla J_s(x)\| \le -(\nabla J_s(x))^{\mathsf{T}}\dot{x}$ holds for all $x \in \Omega$, i.e., $(\nabla J_s(x))^{\mathsf{T}}\dot{x} \le -\lambda_6\|\nabla J_s(x)\|$. Hence, the inequality (A.12) becomes

$$\begin{aligned}
\dot{L}(t) &\le -\lambda_7\|\tilde{\omega}_c\|^4 + \lambda_8\|\tilde{\omega}_c\|^2 + \lambda_5^2 + \frac{\alpha_s}{\alpha_c}(\nabla J_s(x))^{\mathsf{T}}\dot{x}\\
&\le -\lambda_7\|\tilde{\omega}_c\|^4 + \lambda_8\|\tilde{\omega}_c\|^2 + \lambda_5^2 - \frac{\alpha_s}{\alpha_c}\lambda_6\|\nabla J_s(x)\|. \qquad \text{(A.15)}
\end{aligned}$$

Therefore, given the following inequality:

$$\|\tilde{\omega}_c\| \ge \sqrt{\frac{\lambda_8 + \sqrt{4\lambda_5^2\lambda_7 + \lambda_8^2}}{2\lambda_7}} \triangleq \mathscr{A}_1 \qquad \text{(A.16)}$$

or

$$\|\nabla J_s(x)\| \ge \frac{\alpha_c\left(4\lambda_5^2\lambda_7 + \lambda_8^2\right)}{4\alpha_s\lambda_6\lambda_7} \triangleq \mathscr{B}_1 \qquad \text{(A.17)}$$

holds, we conclude $\dot{L}(t) < 0$.

*Case 2:* $\Pi(x,\hat{u}) = 1$. Adding and subtracting $\alpha_s(\nabla J_s(x))^{\mathsf{T}} g(x)R^{-1}g^{\mathsf{T}}(x)\nabla\varepsilon_c(x)/(2\alpha_c)$ to the right hand side of (A.12) and taking Assumption 2 into consideration yield

$$\begin{aligned}
\dot{L}(t) &\le -\lambda_7\|\tilde{\omega}_c\|^4 + \lambda_8\|\tilde{\omega}_c\|^2 + \lambda_5^2\\
&\quad - \frac{\alpha_s}{2\alpha_c}\tilde{\omega}_c^{\mathsf{T}}\nabla\sigma_c(x)g(x)R^{-1}g^{\mathsf{T}}(x)\nabla J_s(x)\\
&\quad + \frac{\alpha_s}{\alpha_c}(\nabla J_s(x))^{\mathsf{T}}(f(x) + g(x)\hat{u})\\
&= -\lambda_7\|\tilde{\omega}_c\|^4 + \lambda_8\|\tilde{\omega}_c\|^2 + \lambda_5^2\\
&\quad + \frac{\alpha_s}{\alpha_c}(\nabla J_s(x))^{\mathsf{T}}(f(x) + g(x)u^*)\\
&\quad + \frac{\alpha_s}{2\alpha_c}(\nabla J_s(x))^{\mathsf{T}}g(x)R^{-1}g^{\mathsf{T}}(x)\nabla\varepsilon_c(x)\\
&\le -\lambda_7\|\tilde{\omega}_c\|^4 + \lambda_8\|\tilde{\omega}_c\|^2 + \lambda_5^2\\
&\quad - \frac{\alpha_s}{\alpha_c}\lambda_m\|\nabla J_s(x)\|^2 + \frac{\alpha_s}{2\alpha_c}\lambda_9\lambda_{10}\|\nabla J_s(x)\|. \qquad \text{(A.18)}
\end{aligned}$$

Therefore, given the following inequality:

$$\|\tilde{\omega}_c\| \ge \sqrt{\frac{\lambda_8}{2\lambda_7} + \sqrt{\frac{\lambda_5^2}{\lambda_7} + \frac{\lambda_8^2}{4\lambda_7^2} + \frac{\alpha_s\lambda_9^2\lambda_{10}^2}{16\alpha_c\lambda_m\lambda_7}}} \triangleq \mathscr{A}_2 \qquad \text{(A.19)}$$

or

$$\|\nabla J_s(x)\| \ge \frac{\lambda_9\lambda_{10}}{4\lambda_m} + \sqrt{\frac{\alpha_c\left(4\lambda_5^2\lambda_7 + \lambda_8^2\right)}{4\alpha_s\lambda_m\lambda_7} + \frac{\lambda_9^2\lambda_{10}^2}{16\lambda_m^2}} \triangleq \mathscr{B}_2 \qquad \text{(A.20)}$$

holds, we obtain $\dot{L}(t) < 0$.

To summarize, if the inequality $\|\tilde{\omega}_c\| > \max(\mathscr{A}_1, \mathscr{A}_2) = \mathscr{A}$ or $\|\nabla J_s(x)\| > \max(\mathscr{B}_1, \mathscr{B}_2) = \mathscr{B}$ holds, then $\dot{L}(t) < 0$. Considering the fact that $J_s(x)$ is chosen as a polynomial and in accordance with the standard Lyapunov extension theorem [70], we can derive the conclusion that the state $x$ and the weight estimation error $\tilde{\omega}_c$ are UUB. This completes the proof. ■

## REFERENCES


[1] P. J. Werbos, "Approximate dynamic programming for real-time control and neural modeling," in *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches*, D. A. White and D. A. Sofge, Eds. New York, NY, USA: Van Nostrand Reinhold, 1992, ch. 13.

[2] P. J. Werbos, "ADP: The key direction for future research in intelligent control and understanding brain intelligence," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 898–900, Aug. 2008.

[3] P. J. Werbos, "Intelligence in the brain: A theory of how it works and how to build it," *Neural Netw.*, vol. 22, no. 3, pp. 200–212, Apr. 2009.

[4] D. V. Prokhorov and D. C. Wunsch, "Adaptive critic designs," *IEEE Trans. Neural Netw.*, vol. 8, no. 5, pp. 997–1007, Sep. 1997.

[5] D. P. Bertsekas, M. L. Homer, D. A. Logan, S. D. Patek, and N. R. Sandell, "Missile defense and interceptor allocation by neuro-dynamic programming," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 30, no. 1, pp. 42–51, Jan. 2000.

[6] J. Si and Y. T. Wang, "On-line learning control by association and reinforcement," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 264–276, Mar. 2001.

[7] S. Shervais, T. T. Shannon, and G. G. Lendaris, "Intelligent supply chain management using adaptive critic learning," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 33, no. 2, pp. 235–244, Mar. 2003.

[8] J. Varghese and S. Mukhopadhyay, "Automated web navigation using multiagent adaptive dynamic programming," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 33, no. 3, pp. 412–417, May 2003.

[9] F. Y. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: An introduction," *IEEE Comput. Intell. Mag.*, vol. 4, no. 2, pp. 39–47, May 2009.

[10] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits Syst. Mag.*, vol. 9, no. 3, pp. 32–50, Jul. 2009.

[11] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, "Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers," *IEEE Control Syst. Mag.*, vol. 32, no. 6, pp. 76–105, Dec. 2012.

[12] H. Zhang, X. Zhang, Y. Luo, and J. Yang, "An overview of research on adaptive dynamic programming," *Acta Autom. Sinica*, vol. 39, no. 4, pp. 303–311, Apr. 2013.

[13] D. Liu, H. Li, and D. Wang, "Data-based self-learning optimal control: Research progress and prospects," *Acta Autom. Sinica*, vol. 39, no. 11, pp. 1858–1870, Nov. 2013.

[14] P. Rakshit *et al.*, "Realization of an adaptive memetic algorithm using differential evolution and Q-learning: A case study in multirobot path planning," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 4, pp. 814–831, Jul. 2013.

[15] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 943–949, Aug. 2008.

[16] H. Zhang, Y. Luo, and D. Liu, "Neural-network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraints," *IEEE Trans. Neural Netw.*, vol. 20, no. 9, pp. 1490–1503, Sep. 2009.

[17] F. Y. Wang, N. Jin, D. Liu, and Q. Wei, "Adaptive dynamic programming for finite-horizon optimal control of discrete-time nonlinear systems with $\varepsilon$-error bound," *IEEE Trans. Neural Netw.*, vol. 22, no. 1, pp. 24–36, Jan. 2011.

[18] D. Wang, D. Liu, and Q. Wei, "Finite-horizon neuro-optimal tracking control for a class of discrete-time nonlinear systems using adaptive dynamic programming approach," *Neurocomputing*, vol. 78, no. 1, pp. 14–22, Feb. 2012.

[19] D. Liu, D. Wang, D. Zhao, Q. Wei, and N. Jin, "Neural-network-based optimal control for a class of unknown discrete-time nonlinear systems using globalized dual heuristic programming," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 3, pp. 628–634, Jul. 2012.

[20] D. Wang, D. Liu, Q. Wei, D. Zhao, and N. Jin, "Optimal control of unknown nonaffine nonlinear discrete-time systems based on adaptive dynamic programming," *Automatica*, vol. 48, no. 8, pp. 1825–1832, Aug. 2012.

[21] D. Wang, D. Liu, D. Zhao, Y. Huang, and D. Zhang, "A neural-network-based iterative GDHP approach for solving a class of nonlinear optimal control problems with control constraints," *Neural Comput. Appl.*, vol. 22, no. 2, pp. 219–227, Feb. 2013.

[22] D. Liu, H. Li, and D. Wang, "Neural-network-based zero-sum game for discrete-time nonlinear systems via iterative adaptive dynamic programming algorithm," *Neurocomputing*, vol. 110, pp. 92–100, Jun. 2013.

[23] H. Li and D. Liu, "Optimal control for discrete-time affine nonlinear systems using general value iteration," *IET Control Theory Appl.*, vol. 6, no. 18, pp. 2725–2736, Dec. 2012.

[24] A. Heydari and S. N. Balakrishnan, "Finite-horizon control-constrained nonlinear optimal control using single network adaptive critics," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 1, pp. 145–157, Jan. 2013.

[25] Z. Ni, H. He, and J. Wen, "Adaptive learning in tracking control based on the dual critic network design," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 6, pp. 913–928, Jun. 2013.

[26] H. He, Z. Ni, and J. Fu, "A three-network architecture for on-line learning and optimization based on adaptive dynamic programming," *Neurocomputing*, vol. 78, no. 1, pp. 3–13, Feb. 2012.

[27] Z. Ni, H. He, J. Wen, and X. Xu, "Goal representation heuristic dynamic programming on maze navigation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 12, pp. 2038–2050, Dec. 2013.

[28] X. Xu, C. Lian, L. Zuo, and H. He, "Kernel-based approximate dynamic programming for real-time online learning control: An experimental study," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 1, pp. 146–156, Jan. 2014.

[29] T. Dierks and S. Jagannathan, "Online optimal control of affine nonlinear discrete-time systems with unknown internal dynamics by using time-based policy update," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 7, pp. 1118–1129, Jul. 2012.

[30] Q. Wei and D. Liu, "Data-driven neuro-optimal temperature control of water gas shift reaction using stable iterative adaptive dynamic programming," *IEEE Trans. Ind. Electron.*, vol. 61, no. 11, pp. 6399–6408, Nov. 2014.

[31] D. Liu and Q. Wei, "Policy iterative adaptive dynamic programming algorithm for discrete-time nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 621–634, Mar. 2014.

[32] D. Wang and D. Liu, "Neuro-optimal control for a class of unknown nonlinear dynamic systems using SN-DHP technique," *Neurocomputing*, vol. 121, pp. 218–225, Dec. 2013.

[33] D. Liu, D. Wang, and X. Yang, "An iterative adaptive dynamic programming algorithm for optimal control of unknown discrete-time nonlinear systems with constrained inputs," *Inf. Sci.*, vol. 220, pp. 331–342, Jan. 2013.

[34] Q. Wei and D. Liu, "An iterative $\epsilon$-optimal control scheme for a class of discrete-time nonlinear systems with unfixed initial state," *Neural Netw.*, vol. 32, no. 6, pp. 236–244, Aug. 2012.

[35] D. Liu and Q. Wei, "Finite-approximation-error-based optimal control approach for discrete-time nonlinear systems," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 779–789, Apr. 2013.

[36] D. Liu, H. Javaherian, O. Kovalenko, and T. Huang, "Adaptive critic learning techniques for engine torque and air-fuel ratio control," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 988–993, Aug. 2008.

[37] 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, 779–791, May 2005.

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

[39] K. G. Vamvoudakis and F. L. Lewis, "Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, May 2010.

[40] S. Bhasin *et al.*, "A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems," *Automatica*, vol. 49, no. 1, pp. 82–92, Jan. 2013.

[41] H. N. Wu and B. Luo, "Neural network based online simultaneous policy update algorithm for solving the HJI equation in nonlinear $H_\infty$ control," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 12, pp. 1884–1895, Dec. 2012.

[42] H. Zhang, L. Cui, and Y. Luo, "Near-optimal control for nonzero-sum differential games of continuous-time nonlinear systems using single-network ADP," *IEEE Trans. Cybern.*, vol. 43, no. 1, pp. 206–216, Feb. 2013.

[43] D. Liu, X. Yang, and H. Li, "Adaptive optimal control for a class of continuous-time affine nonlinear systems with unknown internal dynamics," *Neural Comput. Appl.*, vol. 23, pp. 1843–1850, Dec. 2013.

[44] D. Liu, Y. Huang, D. Wang, and Q. Wei, "Neural-network-observer-based optimal control for unknown nonlinear systems using adaptive dynamic programming," *Int. J. Control*, vol. 86, no. 9, pp. 1554–1566, 2013.

[45] T. Dierks and S. Jagannathan, "Optimal control of affine nonlinear continuous-time systems," in *Proc. Amer. Control Conf.*, Baltimore, MD, USA, Jun. 2010, pp. 1568–1573.

[46] D. Nodland, H. Zargarzadeh, and S. Jagannathan, "Neural network-based optimal adaptive output feedback control of a helicopter UAV," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 7, pp. 1061–1073, Jul. 2013.

[47] D. Liu, D. Wang, and H. Li, "Decentralized stabilization for a class of continuous-time nonlinear interconnected systems using online learning optimal control approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 2, pp. 418–428, Feb. 2014.

[48] D. Liu, H. Li, and D. Wang, "Online synchronous approximate optimal learning algorithm for multi-player non-zero-sum games with unknown dynamics," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 8, pp. 1015–1027, Aug. 2014.

[49] 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 Trans. Autom. Sci. Eng.*, vol. 11, no. 2, pp. 627–632, Apr. 2014.

[50] D. Wang, D. Liu, H. Li, and H. Ma, "Neural-network-based robust optimal control design for a class of uncertain nonlinear systems via adaptive dynamic programming," *Inf. Sci.*, vol. 282, pp. 167–179, Oct. 2014.

[51] X. Yang, D. Liu, and D. Wang, "Reinforcement learning for adaptive optimal control of unknown continuous-time nonlinear systems with input constraints," *Int. J. Control*, vol. 87, no. 3, pp. 553–566, Mar. 2014.

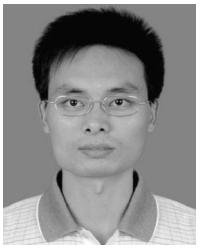
[52] H. Li, D. Liu, and D. Wang, "Integral reinforcement learning for linear continuous-time zero-sum games with completely unknown dynamics," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 3, pp. 706–714, Jul. 2014.

[53] H. Modares, F. L. Lewis, and M. B. Naghibi-Sistani, "Adaptive optimal control of unknown constrained-input systems using policy iteration and neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 10, pp. 1513–1525, Oct. 2013.
[54] B. Luo, H. N. Wu, and T. Huang, "Off-policy reinforcement learning for $H_\infty$ control design," *IEEE Trans. Cybern.*, to be published.
[55] D. M. Adhyaru, I. N. Kar, and M. Gopal, "Bounded robust control of nonlinear systems using neural network-based HJB solution," *Neural Comput. Appl.*, vol. 20, no. 1, pp. 91–103, 2011.
[56] D. M. Adhyaru, I. N. Kar, and M. Gopal, "Fixed final time optimal control approach for bounded robust controller design using Hamilton–Jacobi–Bellman solution," *IET Control Theory Appl.*, vol. 3, no. 9, pp. 1183–1195, Sep. 2009.
[57] S. Mehraeen and S. Jagannathan, "Decentralized optimal control of a class of interconnected nonlinear discrete-time systems by using online Hamilton–Jacobi–Bellman formulation," *IEEE Trans. Neural Netw.*, vol. 22, no. 11, pp. 1757–1769, Nov. 2011.
[58] H. Xu, S. Jagannathan, and F. L. Lewis, "Stochastic optimal control of unknown linear networked control system in the presence of random delays and packet losses," *Automatica*, vol. 48, no. 6, pp. 1017–1030, Jun. 2012.
[59] J. Liang, G. K. Venayagamoorthy, and R. G. Harley, "Wide-area measurement based dynamic stochastic optimal power flow control for smart grids with high variability and uncertainty," *IEEE Trans. Smart Grid*, vol. 3, no. 1, pp. 59–69, Mar. 2012.
[60] W. M. Haddad, V. S. Chellaboina, and J. L. Fausz, "Robust nonlinear feedback control for uncertain linear systems with nonquadratic performance criteria," *Syst. Control Lett.*, vol. 33, no. 5, pp. 327–338, 1998.
[61] W. M. Haddad, V. Chellaboina, J. L. Fausz, and A. Leonessa, "Optimal non-linear robust control for nonlinear uncertain systems," *Int. J. Control*, vol. 73, no. 4, pp. 329–342, 2000.
[62] W. M. Haddad and V. Chellaboina, *Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach*. Princeton, NJ, USA: Princeton Univ. Press, 2008.
[63] F. Lin, R. D. Brand, and J. Sun, "Robust control of nonlinear systems: Compensating for uncertainty," *Int. J. Control*, vol. 56, no. 6, pp. 1453–1459, 1992.
[64] S. S. L. Chang and T. K. C. Peng, "Adaptive guaranteed cost control of systems with uncertain parameters," *IEEE Trans. Autom. Control*, vol. 17, no. 4, pp. 474–483, Apr. 1972.
[65] L. Yu, Q. L. Han, and M. X. Sun, "Optimal guaranteed cost control of linear uncertain systems with input constraints," *Int. J. Control Autom. Syst.*, vol. 3, no. 3, pp. 397–402, Sep. 2005.
[66] L. Yu and J. Chu, "An LMI approach to guaranteed cost control of linear uncertain time-delay systems," *Automatica*, vol. 35, no. 6, pp. 1155–1159, Jun. 1999.
[67] R. W. Beard, G. N. Saridis, and J. T. Wen, "Galerkin approximations of the generalized Hamilton–Jacobi–Bellman equation," *Automatica*, vol. 33, no. 12, pp. 2159–2177, Dec. 1997.
[68] M. Krstic and Z. H. Li, "Inverse optimal design of input-to-state stabilizing nonlinear controllers," *IEEE Trans. Autom. Control*, vol. 43, no. 3, pp. 336–350, Mar. 1998.
[69] M. Krstic and P. Tsiotras, "Inverse optimal stabilization of a rigid spacecraft," *IEEE Trans. Autom. Control*, vol. 44, no. 5, pp. 1042–1049, May 1999.
[70] F. L. Lewis, S. Jagannathan, and A. Yesildirek, *Neural Network Control of Robot Manipulators and Nonlinear Systems*. London, U.K.: Taylor & Francis, 1999.

**Derong Liu** (S'91–M'94–SM'96–F'05) received the Ph.D. degree in electrical engineering from the University of Notre Dame, Notre Dame, IN, USA, in 1994.

He was a Staff Fellow at General Motors Research and Development Center, from 1993 to 1995 and an Assistant Professor at the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, USA, from 1995 to 1999. He was with the University of Illinois at Chicago, Chicago, IL, USA, in 1999, where he became a Full Professor of electrical and computer engineering, and computer science, in 2006. He was selected for the "100 Talents Program" by the Chinese Academy of Sciences, Beijing, China, in 2008. He has published 15 books (six research monographs and nine edited volumes).

Prof. Liu was the recipient of the Faculty Early Career Development Award from the National Science Foundation, in 1999, the University Scholar Award from the University of Illinois at Chicago, from 2006 to 2009, and the Overseas Outstanding Young Scholar Award from the National Natural Science Foundation of China, in 2008. He is currently a Distinguished Lecturer of the IEEE Computational Intelligence Society and also an Editor-in-Chief of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS. He is the General Chair of the 2014 IEEE World Congress on Computational Intelligence and the 2016 World Congress on Intelligent Control and Automation. He is a fellow of the International Neural Network Society.



**Ding Wang** received the B.S. degree in mathematics from the Zhengzhou University of Light Industry, Zhengzhou, China, the M.S. degree in operational research and cybernetics from Northeastern University, Shenyang, China, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2007, 2009, and 2012, respectively.

He is currently an Assistant Professor with The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences. His current research interests include adaptive dynamic programming, neural networks and learning systems, and complex systems and intelligent control.

**Fei-Yue Wang** (S'87–M'89–SM'94–F'03) received the Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990.

He was with the University of Arizona, Tucson, AZ, USA, in 1990, where he became a Professor and a Director of the Robotics and Automation Laboratory and the Program for Advanced Research in Complex Systems. He is the Founder of the Intelligent Control and Systems Engineering Center with the Chinese Academy of Sciences (CAS), Beijing, China, under the support of the Outstanding Overseas Chinese Talents Program, in 1999. Since 2002, he has been the Director of the Key Laboratory of Complex Systems and Intelligence Science with the CAS, and is currently the Director of The State Key Laboratory of Management and Control for Complex Systems. His current research interests include social computing, web science, complex systems, and intelligent control.

Dr. Wang was the recipient of the National Prize in Natural Sciences of China and was awarded the Outstanding Scientist by ACM for his work in intelligent control and social computing. He was an Editor-in-Chief of the *International Journal of Intelligent Control and Systems* and the *World Scientific Series in Intelligent Control and Intelligent Automation*, from 1995 to 2000. He is currently an Editor-in-Chief of the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS. He has served as Chair for over 20 IEEE, ACM, Institute for Operations Research and the Management Sciences, and American Society of Mechanical Engineers (ASME) conferences. He was the President of the IEEE Intelligent Transportation Systems Society, from 2005 to 2007, the Chinese Association for Science and Technology, New York, NY, USA, in 2005, and the U.S. Zhu Kezhen Education Foundation from 2007 to 2008. He is currently the Vice President of the ACM China Council and Vice President/Secretary-General of Chinese Association of Automation. He is the member of Sigma Xi and an Elected Fellow of the International Council on Systems Engineering, International Federation of Automatic Control, ASME, and American Association for the Advancement of Science.

**Hongliang Li** (S'13) received the B.S. degree in mechanical engineering and automation from the Beijing University of Posts and Telecommunications, Beijing, China, in 2010. He is currently pursuing the Ph.D. degree from The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing.

He is currently with the University of Chinese Academy of Sciences. His current research interests include machine learning, reinforcement learning, neural networks, intelligent control, and smart grid.

**Xiong Yang** received the B.S. degree in mathematics and applied mathematics from Central China Normal University, Wuhan, China, the M.S. degree in pure mathematics from Shandong University, Jinan, China, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2008, 2011, and 2014, respectively.

He is currently an Assistant Professor with the Institute of Automation, Chinese Academy of Sciences. His current research interests include adaptive dynamic programming, reinforcement learning, adaptive control, and neural networks.